



**UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES**

**CENTRO DE CIENCIAS BÁSICAS**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**TESINA**

**ANÁLISIS COMPARATIVO ENTRE UNA MÁQUINA DE SOPORTE VECTORIAL, EL  
MODELO RANDOM FOREST Y EL MODELO DE BAYES INGENUO EN LA  
PREDICCIÓN DE INTENCIÓN DE MOVIMIENTO ORIENTADO AL USO DE UNA  
COMPUTADORA.**

**Tiempo de redacción:** Once meses.

**Presenta:**

Joel Alejandro Espinoza Sánchez

Para Obtener el Grado de Ingeniería en Computación Inteligente

**Directores:**

Dra. Aurora Torres Soto

Dra. María Dolores Torres Soto

**Comité Evaluador:**

Dr. Francisco Javier Álvarez Rodríguez

Dr. Alejandro Padilla Díaz

Aguascalientes, Ags., Mayo 2023.



Aguascalientes, Ags., XX de Mayo de 2023

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Joel Alejandro Espinoza Sánchez** de la carrera de Ingeniería en Computación Inteligente con ID: **211800**, ha terminado satisfactoriamente su tesina titulada: **“Análisis Comparativo entre una Máquina de Vectores de Soporte, el modelo Random Forest y el modelo de Bayes Ingenuo en la predicción de intención de movimiento orientado al uso de una computadora.”**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

**ATENTAMENTE**

---

**Dra. Aurora Torres Soto**  
Codirector



Aguascalientes, Ags., XX de Mayo de 2023

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Joel Alejandro Espinoza Sánchez** de la carrera de Ingeniería en Computación Inteligente con **ID: 211800**, ha terminado satisfactoriamente su tesina titulada: **“Análisis Comparativo entre una Máquina de Vectores de Soporte, el modelo Random Forest y el modelo de Bayes Ingenuo en la predicción de intención de movimiento orientado al uso de una computadora.”**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

**ATENTAMENTE**

---

**Dra. María Dolores Torres Soto**  
Codirector



Aguascalientes, Ags., XX de Mayo de 2023

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Joel Alejandro Espinoza Sánchez** de la carrera de Ingeniería en Computación Inteligente con ID: **211800**, ha terminado satisfactoriamente su tesina titulada: **“Análisis Comparativo entre una Máquina de Vectores de Soporte, el modelo Random Forest y el modelo de Bayes Ingenuo en la predicción de intención de movimiento orientado al uso de una computadora.”**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

**ATENTAMENTE**

---

**Dr. Francisco Javier Álvarez Rodríguez**  
**Miembro del Comité Tutorial**



Aguascalientes, Ags., XX de Mayo de 2023

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Joel Alejandro Espinoza Sánchez** de la carrera de Ingeniería en Computación Inteligente con **ID: 211800**, ha terminado satisfactoriamente su tesina titulada: **“Análisis Comparativo entre una Máquina de Vectores de Soporte, el modelo Random Forest y el modelo de Bayes Ingenuo en la predicción de intención de movimiento orientado al uso de una computadora.”**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

**ATENTAMENTE**

---

**Dr. Alejandro Padilla Díaz**  
**Miembro del Comité Tutorial**

## Resumen

Las interfaces cerebro – computadora permiten un nuevo canal de comunicación directo entre el cerebro y la máquina. Sin embargo para interpretar los pensamientos como el habla imaginada o la intención de movimiento se necesitan herramientas como electroencefalogramas que permitan la recolección de la actividad cerebral. Además, la interpretación de estos datos necesita de algoritmos computacionales de inteligencia artificial que sean efectivos, pero entre toda la gama de algoritmos existentes en este campo se esperaría idealmente comenzar a construir una interfaz cerebro – computadora con algún algoritmo con alta precisión de predicción de intención del movimiento con base en los datos de actividad cerebral de una persona.

Es así que en la presente investigación se planea implementar una interfaz usando una diadema Emotiv Epoc+, el programa elaborado en Python CyKit, OpenViBE y código personal desarrollado en Python para probar experimentalmente cuál de los algoritmos de aprendizaje supervisado entre la máquina de soporte vectorial, el modelo de Naive Bayes y el modelo Random Forest es capaz de predecir con mayor eficiencia intenciones del movimiento con datos obtenidos mediante experimentación.

**Palabras clave:** Interfaz Cerebro – Computadora. Intención del movimiento. Aprendizaje automático. Máquina de soporte vectorial. Bayes ingenuo. Bosque aleatorio.

## **Abstract**

Brain – computer interfaces provide a new direct communication channel between the brain and the machine. However, to interpret thoughts such as imagined speech or intention of movement, tools such as electroencephalograms are needed to collect brain activity. In addition, the interpretation of this data requires computational algorithms of artificial intelligence that are effective. However, ideally, among the entire range of existing algorithms in this field, it would be expected to start building a brain-computer interface with an algorithm that has high prediction accuracy of movement intention based on a person's brain activity data.

Therefore, the present research aims to implement an interface using an Emotiv Epoc+ headset, the Python CyKit program, OpenViBE, and personal code developed in Python to experimentally test which supervised machine learning algorithm between Support Vector Machine, Naive Bayes, and Random Forest models can predict intention of movement with greater efficiency using data obtained through experimentation.

**Key words:** Brain – computer interfaces. Intention of movement. Machine learning. Support Vector Machine. Naive Bayes. Random Forest.

## **Agradecimientos**

Sección de agradecimientos.



## Tabla de Contenido

Tabla de Contenido .....	1
Índice de Figuras .....	4
Índice de Tablas .....	7
Índice de Ecuaciones .....	8
1. Introducción.....	9
2. Planteamiento del Problema .....	13
3. Justificación.....	14
4. Objetivos .....	15
5. Hipótesis.....	16
6. Pregunta de Investigación .....	17
7. Marco Teórico.....	18
7.1. Introducción a la Neuroanatomía .....	18
7.1.1. El cerebro y el sistema nervioso .....	18
7.1.2. El cerebro a nivel macroestructura .....	19
7.1.3. El cerebro a nivel microestructura .....	24
7.1.3.1. Las neuronas.....	24
7.1.3.2. Morfología de las neuronas.....	25
7.1.3.3. Funciones de las neuronas .....	28
7.1.3.4. Interacción entre neuronas .....	33
7.2. Electroencefalografía y Estudios del Cerebro .....	35
7.2.1. El Electroencefalograma .....	35
7.2.2. Breve Historia del Electroencefalograma .....	36
7.2.3. Ondas Encefálicas.....	36
7.3. Fundamentos Computacionales, Inteligencia Artificial y Aprendizaje Automático.....	38

7.3.1. Introducción a la Computación -----	38
7.3.1.1. Computación -----	38
7.3.1.2. Programación -----	39
7.3.1.3. El Algoritmo-----	40
7.3.2. Inteligencia Artificial -----	41
7.3.3. Aprendizaje Automático -----	42
7.3.4. Aprendizaje Supervisado -----	44
7.3.4.1. La técnica Support Vector Machine -----	47
7.3.4.2. La técnica Random Forest-----	51
7.3.4.3. La técnica Naive Bayes-----	54
7.4. Habla Imaginada e Intención de Movimiento -----	58
8. Metodología-----	64
9. Desarrollo -----	65
10. Experimentación y pruebas -----	67
11. Análisis e Interpretación de Resultados-----	74
12. Conclusiones -----	78
13. Referencias -----	79
14. Anexos -----	83
14.1. Anexo 1: Configuración para recibir datos desde la diadema hacia el equipo de cómputo usando la paquetería de software de Emotiv, OpenViBE y la librería de Python: Cykit -----	83
14.2. Anexo 2: Código del archivo de Python: functions.py.-----	100
14.3. Anexo 3: Pseudocódigo de la función RunUpdateStream() encontrada en el archivo de Python: functions.py.-----	104
14.4. Anexo 4: Pseudocódigo de la función createDataset() encontrada en el archivo de Python: functions.py.-----	105
14.5. Anexo 5: Acuerdo de privacidad entregado a los usuarios que colaboraron con el experimento.-----	106

14.6. Anexo 6: Protocolo de experimentación. -----	108
14.7. Anexo 7: Evidencias de la experimentación de campo. -----	111

## Índice de Figuras

### 7. Marco Teórico

#### 7.1. Introducción a la Neuroanatomía

Figura 7.1.1. Los cuatro lóbulos cerebrales que pueden observarse desde la corteza cerebral (Sabater, 2020). -----	20
Figura 7.1.2. Representación gráfica de la ubicación del lóbulo insular, señalada en color verde. (Laguna, 2022).-----	22
Figura 7.1.3. Estructuras que conforman el lóbulo límbico. (Triglia, 2016).-----	23
Figura 7.1.4. Morfología de una neurona. (Merck & Co, 2022). -----	25
Figura 7.1.5. Gráfica del potencial de acción de una neurona en función del tiempo. (MDurance, 2021).-----	30
Figura 7.1.6. Gráfica del potencial de acción de una neurona mostrando la interacción iónica. (Olmo, Nave, & Nave, 2022).-----	31

#### 7.2. Electroencefalografía y Estudios del Cerebro

Figura 7.2.1. Ejemplo de electroencefalograma (Sosa Romano, 2022). -----	35
Figura 7.2.2. Primer registro de un electroencefalograma en un humano (Wikipedia, s.f.). --	36

#### 7.3. Fundamentos Computacionales, Inteligencia Artificial y Aprendizaje Automático

Figura 7.3.1. Clasificación del Machine Learning (o Aprendizaje Automático) en sus tres grandes tipos de aprendizaje (González Barrio, Calleja Ochoa, Gómez-Escudero, Rodríguez Ezquerro, & López de Lacalle Marcaide, 2021).-----	43
Figura 7.3.2. Procedimiento general del Aprendizaje Supervisado (TIBCO Data Science, 2022). -----	45
Figura 7.3.3. Principales algoritmos de Machine Learning (Betanzos Gómez, 2020).-----	47
Figura 7.3.4. Definición del “margen” entre clases: el criterio que los SVM intentan optimizar (The MathWorks Inc., s.f.).-----	48
Figura 7.3.5. Gráficas de separación del hiperplano de las distintos kernels (Marius, 2020). 50	
Figura 7.3.6. Árbol de Decisión aplicado en Economía (Sanabria Castro, 2020). -----	51
Figura 7.3.7. Visualización del “bagging” como separación, usado en Random Forest (Orellana Alvear, 2018). -----	53
Figura 7.3.8. Comparación visual entre el “bagging” y el “boosting” (Sruthi, 2022). -----	54
Figura 7.3.9. Visualización del teorema de Bayes por superposición de dos árboles de decisión (Parrás & Tedesco). -----	55
Figura 7.3.10. El uso del Teorema de Bayes en la técnica de Naive Bayes (Roman, 2019). 56	

#### 7.4. Habla Imaginada e Intención de Movimiento

Figura 7.4.1. Ubicación de los nodos en la diadema Emotiv usado en su experimentación (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).-----	62
----------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

### 8. Metodología

Figura 8.1. Diagrama de pasos a seguir según la metodología planteada. ....	64
9. Desarrollo	
Figura 9.1. Escenario de OpenViBE Designer para experimentación de esta investigación. 65	
Figura 9.2. Botón para ciclar la ejecución. ....	65
10. Experimentación y pruebas	
Figura 10.1. Ejemplo de uso de la diadema en los voluntarios 1, 8, 7, 9 y 23 de izquierda a derecha, de arriba abajo. ....	70
Figura 10.2. Fragmento del segundo archivo obtenido de la experimentación de campo del voluntario 14. ....	71
Figura 10.3. Fragmento del segundo archivo obtenido tras la Normalización Z del voluntario 26. ....	71
Figura 10.4. Fragmento del segundo archivo obtenido tras la Transformación de Fourier del voluntario 19. ....	72
Figura 10.5. Resultado de ejecución del modelo de Random Forest sobre el segundo archivo del voluntario 10. ....	72
Figura 10.6. Resultados de ejecución de los tres modelos sobre el primer archivo de los primeros siete voluntarios. ....	73
11. Análisis e Interpretación de Resultados	
Figura 11.1. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos recopilados en la experimentación de campo usando el archivo uno de los voluntarios. ....	74
Figura 11.2. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos recopilados en la experimentación de campo usando el archivo dos de los voluntarios. ....	75
Figura 11.3. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos normalizados usando el archivo dos de los voluntarios. ....	75
Figura 11.4. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos normalizados usando el archivo dos de los voluntarios. ....	76
Figura 11.5. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos transformados mediante una transformada de Fourier usando el archivo dos de los voluntarios. ....	76
Figura 11.6. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos transformados mediante una transformada de Fourier usando el archivo dos de los voluntarios. ....	77
14. Anexos	
14.1. Anexo 1: Configuración para recibir datos desde la diadema hacia el equipo de cómputo usando la paquetería de software de Emotiv, OpenViBE y la librería de Python: Cykit	
Figura 14.1.1. Sitio web de Emotiv. ....	83

Figura 14.1.2. Registro para la cuenta Emotiv ID. -----	84
Figura 14.1.3. Emotiv Launcher sin la detección de una diadema conectada al equipo. ----	85
Figura 14.1.4. Diadema Emotiv encendida y dispositivo USB conectado al equipo. -----	85
Figura 14.1.5. Emotiv Launcher detectando una diadema conectada al equipo. -----	86
Figura 14.1.6. Emotiv Launcher guiando al usuario a acomodar correctamente la diadema.	87
Figura 14.1.7. Sitio web de GitHub donde se encuentra el repositorio del proyecto de CyKit por CymatiCorp. -----	87
Figura 14.1.8. Ejecución correcta de CyKIT.py usando Python. -----	89
Figura 14.1.9. Explicación de algunas variables de configuración para ejecutar CyKit. -----	89
Figura 14.1.10. Explicación de algunas variables de configuración para ejecutar CyKit. ----	90
Figura 14.1.11. CyKit ejecutándose bajo las configuraciones indicadas. -----	91
Figura 14.1.12. Sitio web de OpenViBE. -----	92
Figura 14.1.13. Configuración general de OpenViBE Acquisition Server. -----	92
Figura 14.1.14. Configuración de propiedades del driver de OpenViBE Acquisition Server.-	93
Figura 14.1.15. Conexión exitosa entre CyKit y OpenViBE Acquisition Server. -----	94
Figura 14.1.16. Ilustración del escenario creado en OpenViBE Designer. -----	95
Figura 14.1.17. Electroencefalograma de OpenViBE Designer en funcionamiento correcto.	95
Figura 14.1.18. Tabla de datos de OpenViBE Designer en funcionamiento correcto. -----	96
Figura 14.1.19. Escenario de OpenViBE Designer modificado personalmente.-----	96
Figura 14.1.20. Configuraciones para “CSV File Writer”.-----	97
Figura 14.1.21. Archivo CSV producido tras una ejecución de transmisión de datos desde la diadema. -----	97
Figura 14.1.22. Botón para ciclar la ejecución. -----	98
14.7. Anexo 7: Evidencias de la experimentación de campo.	
Figura 14.7.1. Evidencia de los conjuntos de datos recopilados en el sistema de archivos.	112

## Índice de Tablas

### 7. Marco Teórico

#### 7.2. Electroencefalografía y Estudios del Cerebro

Tabla 7.2.1. Clasificación de las ondas encefálicas (Hermann, 1997). ----- 37

#### 7.3. Fundamentos Computacionales, Inteligencia Artificial y Aprendizaje Automático

Tabla 7.3.1. Cuadro de definiciones de la Inteligencia Artificial presentado por Stuart Russell. (Russell & Norvig, 2004). ----- 41

### 14. Anexos

#### 14.7. Anexo 7: Evidencias de la experimentación de campo.

Tabla 14.7.1. Usuarios voluntarios en la experimentación de campo junto a los términos cedidos en el acuerdo. ----- 111

## Índice de Ecuaciones

### 7. Marco Teórico

#### 7.3. Fundamentos Computacionales, Inteligencia Artificial y Aprendizaje Automático

Ecuación 7.3.1. Función de base radial o gaussiana (The MathWorks Inc., s.f.).----- 49

Ecuación 7.3.2. Función lineal (The MathWorks Inc., s.f.). ----- 49

Ecuación 7.3.3. Función polinómica (The MathWorks Inc., s.f.). ----- 49

Ecuación 7.3.4. Función sigmoide (The MathWorks Inc., s.f.). ----- 49

Ecuación 7.3.5. El Teorema de Bayes (Parzen, 1987). ----- 55

Ecuación 7.3.6. La Regla de Bayes (Walpole, Myers, Myers, & Ye, 2012). ----- 56

#### 7.4. Habla Imaginada e Intención de Movimiento

Ecuación 7.4.1. Fórmula del cálculo del CAR (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013). ----- 60

Ecuación 7.4.2. Fórmula del cálculo de la DWT (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013). ----- 61

Ecuación 7.4.3. Definición de la base de información disponible D (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013). ----- 61



## 1. Introducción

La interacción entre el ser humano y un equipo de cómputo ha mejorado para que esta relación y comunicación sea cada vez más cómoda y deseable para el individuo. La importancia de dar una buena experiencia al usuario es crucial en el mundo actual, donde la tecnología juega un papel fundamental en la vida de las personas. La interacción entre el usuario y la computadora es cada vez más relevante, y una experiencia positiva puede marcar la diferencia entre una persona satisfecha y una insatisfecha.

La mejora de las interfaces humano – computadora es fundamental para garantizar una buena experiencia al usuario. Las interfaces deben ser intuitivas, fáciles de usar y eficientes para garantizar que el usuario no tenga problemas para realizar sus tareas. Una buena interfaz permite que el usuario se sienta cómodo y confiado mientras interactúa con la computadora, lo que a su vez mejora su experiencia y aumenta la satisfacción.

Las interfaces humano – computadora son la clave para una experiencia positiva. Deben ser diseñadas teniendo en cuenta las necesidades y preferencias del usuario, y deben ser continuamente mejoradas y optimizadas para garantizar que el usuario tenga la mejor experiencia posible. La mejora de las interfaces humano – computadora no solo mejora la experiencia del usuario, sino que también aumenta la eficiencia y productividad de la computadora, lo que a su vez aumenta la satisfacción del usuario.

La evolución de la interacción entre humano y computadora ha sido constante desde el surgimiento de las computadoras. Al principio, las interfaces eran limitadas y se basaban en la entrada de comandos a través de teclados y pantallas. Con el tiempo, estas interfaces evolucionaron para incluir gráficos y diseños más atractivos, así como la introducción de dispositivos como el mouse y la pantalla táctil, lo que permitió una interacción más intuitiva y natural.

Sin embargo, con la emergente tecnología de las interfaces cerebro – computadora, estamos presenciando un cambio fundamental en la forma en que interactuamos con las computadoras. Esta tecnología permite a los usuarios controlar las computadoras y otros dispositivos mediante la lectura de señales cerebrales, lo que significa que los usuarios pueden controlar sus dispositivos sin tener que usar dispositivos externos como lo son los tradicionales dispositivos de entrada, teclados o el mouse. La tecnología de las interfaces cerebro – computadora abre nuevas posibilidades para la interacción humano – computadora, lo que puede revolucionar la forma en que las personas interactúan con la tecnología.

Las interfaces cerebro – computadora pueden ser de gran ayuda para personas con ciertas discapacidades que pueden limitar su capacidad para interactuar con la tecnología. Por ejemplo, para personas con discapacidades motoras o de habla, las interfaces cerebro – computadora pueden ofrecer una forma alternativa y más accesible de controlar las computadoras y otros dispositivos.

Esto significa que las personas con discapacidades pueden tener una experiencia más inclusiva y equitativa con la tecnología.

En contraste con la tradicional interfaz humano – computadora, las interfaces cerebro – computadora pueden ser mucho más accesibles para personas con discapacidades. Por ejemplo, las interfaces cerebro – computadora no requieren la capacidad de mover los brazos o las manos, o de hablar, lo que significa que están disponibles para una amplia gama de personas con discapacidades. Igualmente, las interfaces cerebro – computadora son más intuitivas y naturales, lo que significa que son más fáciles de usar y pueden ofrecer una experiencia más satisfactoria para los usuarios con discapacidades.

La tecnología de las interfaces cerebro – computadora representa un salto tecnológico en la interacción humano – computadora al permitir la traducción directa de pensamientos abstractos a acciones en el mundo real. Esto significa que los usuarios pueden controlar las computadoras y otros dispositivos simplemente pensando en lo que quieren que suceda. Esta capacidad es una característica clave de las interfaces cerebro – computadora y es una de las formas más avanzadas y naturales de interactuar con la tecnología.

El poder operar una computadora mediante la tecnología de las interfaces cerebro – computadora es solo un ejemplo de cómo esta tecnología puede transformar la forma en que interactuamos con el mundo. Al permitir la traducción directa de pensamientos abstractos a acciones concretas, las interfaces cerebro – computadora pueden revolucionar la forma en la que trabajamos, nos comunicamos y experimentamos el mundo que nos rodea. Además, la tecnología de las interfaces cerebro – computadora puede tener un impacto positivo en la vida de las personas con discapacidades y ofrecer nuevas posibilidades para la interacción humano – computadora en el futuro.

La comprensión de los procesos cerebrales es fundamental para el desarrollo de las interfaces cerebro – computadora eficaces y avanzadas. La neurociencia y la neuroanatomía son disciplinas clave que han contribuido en gran medida a nuestra comprensión del cerebro y cómo funciona. Al entender los procesos cerebrales, los investigadores pueden desarrollar tecnologías que permitan la comunicación directa entre el cerebro y la computadora.

Los estudios de neurociencia y neuroanatomía han permitido avances en la comprensión de la estructura y función del cerebro y cómo se relacionan con los procesos cognitivos y el comportamiento. Esta comprensión es crucial para el desarrollo de interfaces cerebro – computadora avanzadas que puedan interpretar los patrones cerebrales y traducirlos en acciones concretas en el mundo real. Asimismo, la neurociencia y la neuroanatomía también han contribuido a la comprensión de las diferencias individuales en el cerebro y cómo éstos pueden afectar la interacción con la tecnología.

Al comprender cómo funciona el cerebro, los investigadores pueden desarrollar tecnologías que permitan la comunicación directa entre el cerebro y la computadora y así mejorar la interacción humano – computadora. Además, la neurociencia y la neuroanatomía continúan evolucionando y desarrollándose, lo que significa que las interfaces cerebro – computadora seguirán mejorando y evolucionando a medida que los investigadores aprendan más sobre el cerebro y cómo funciona. Es decir, tanto la neurociencia como la neuroanatomía son esenciales para el desarrollo de interfaces cerebro – computadora avanzadas y eficaces y para la mejora continua de tecnologías que permiten la interacción humano – computadora.

La invención de los electroencefalogramas (EEG) ha tenido un impacto significativo en la forma en que se diagnostican y tratan las enfermedades del cerebro. Un EEG es una técnica no invasiva que registra la actividad eléctrica del cerebro y permite a los médicos ver los patrones de actividad cerebral. Esto les permite diagnosticar enfermedades del cerebro como el trastorno del espectro autista, la epilepsia y otras enfermedades neurológicas.

La capacidad de leer la actividad cerebral con un EEG ha permitido una mejor comprensión de las enfermedades del cerebro y ha ayudado a desarrollar tratamientos más eficaces. Por ejemplo, los EEG pueden ayudar a identificar la presencia de anomalías en la actividad cerebral que están asociadas con ciertas enfermedades. Además, los EEG también pueden ayudar a los médicos a monitorear la respuesta al tratamiento y ajustarlo en consecuencia. Es así que la invención de los EEG ha sido fundamental para el diagnóstico de enfermedades del cerebro y ha ayudado a mejorar el tratamiento de estas enfermedades.

De hecho, un EEG puede ser utilizado como una pieza clave de una interfaz cerebro – computadora, ya que permite a los investigadores y desarrolladores capturar la actividad cerebral en tiempo real y convertirla en señales electrónicas que pueden ser procesadas y utilizadas por una computadora. Al registrar la actividad cerebral, los EEG pueden ayudar a identificar patrones específicos de actividad que pueden ser utilizados para controlar dispositivos externos, como robots o dispositivos de asistencia.

Por otra parte, los EEG también pueden ser utilizados para crear una experiencia de interacción más intuitiva y natural para el usuario, permitiéndole controlar dispositivos y aplicaciones simplemente con sus pensamientos. La capacidad de leer la actividad cerebral a través de un EEG permite a los desarrolladores crear interfaces cerebro – computadora más precisas y confiables, lo que puede tener un impacto significativo en la forma en que las personas interactúan con la tecnología.

La implementación de interfaces cerebro – computadora ha sido un desafío debido a la complejidad de predecir lo que el usuario realmente desea con sólo un impulso cerebral. Debido a la naturaleza subjetiva y variada de las percepciones humanas, es difícil traducir la actividad cerebral en una acción específica y predecible para la computadora.

Sin embargo, esta dificultad ha sido abordada a través del uso del aprendizaje automático, una tecnología que permite a la computadora aprender a partir de la experiencia y mejorar con el tiempo. Al usar técnicas de aprendizaje automático, las interfaces cerebro – computadora pueden aprender a identificar patrones específicos de actividad cerebral y traducirlos en acciones en tiempo real, haciendo la interacción más efectiva y eficiente.

El aprendizaje automático se ha utilizado para mejorar la precisión y confiabilidad de las interfaces cerebro – computadora, al mismo tiempo que permite a los desarrolladores tener un mejor entendimiento de las preferencias y comportamientos de los usuarios. Además, al ser una tecnología en constante evolución, el aprendizaje automático puede ser una herramienta clave en el desarrollo de nuevas aplicaciones y tecnologías basadas en interfaces cerebro – computadora.

Es así que la implementación de interfaces cerebro – computadora ha sido un desafío debido a la complejidad de predecir lo que el usuario realmente desea con sólo un impulso cerebral. Sin embargo, el uso del aprendizaje automático ha permitido abordar esta problemática y mejorar la precisión y confiabilidad de las interfaces cerebro – computadora. Con el tiempo, se espera que esta tecnología continúe evolucionando y permita nuevas aplicaciones y tecnologías en el futuro.

El habla imaginada es un fenómeno neurológico en el que el cerebro genera señales eléctricas similares a las producidas durante la producción de habla. Un fenómeno similar es la intención de movimiento, el cual consiste en la voluntad de una persona de realizar alguna acción en específico, labor que comienza en el cerebro y que se encuentra estrechamente relacionada con el habla imaginada, incluso implicando las mismas zonas cerebrales para estas operaciones.

Gracias al aprendizaje automático, se está haciendo un acercamiento cada vez más preciso en la interpretación de estas ondas cerebrales como habla imaginada o intención de movimiento. Esto significa que cada vez es más posible comprender lo que el usuario realmente desea con sólo su impulso cerebral, lo que sería un gran avance en el desarrollo e implementación de las interfaces cerebro – computadora.

Esto quiere decir que mediante modelos de aprendizaje automático es posible realizar estas interpretaciones, gracias a investigaciones que más adelante se abordarán y concluyen en resultados muy esperanzadores, sin embargo, entre la gran variedad de modelos de aprendizaje automático ¿cuál puede ser el más efectivo?

En la presente investigación se tratará de resolver esta pregunta comparando tres modelos de aprendizaje automático, abordando a mayor profundidad este campo y explorando tres interesantes técnicas dentro de esta área, así como el comportamiento cerebral a mayor detalle, igualmente se explorará el EEG y finalmente se hablará de la intención de movimiento junto con los estudios recientes del habla imaginada dentro de este tema del conocimiento.

## 2. Planteamiento del Problema

Si bien se ha concluido en múltiples investigaciones que vale la pena indagar en la actividad cerebral para encontrar mayor información de habla imaginada e intención de movimiento, la cantidad de técnicas de machine learning para analizar estos datos es muy extensa. Pueden reducirse los candidatos iniciales a una selecta muestra de algoritmos como lo son los de aprendizaje supervisado, pero dentro de estos algoritmos, la variedad sigue siendo vasta.

Pensando en tres técnicas elegidas: Support Vector Machine, Random Forest y Naive Bayes, ahora la idea será descubrir cuál de estos tres algoritmos realizaría una predicción de intención de movimiento con mayor eficiencia para tener un mejor acercamiento a la interpretación de los datos que se leen desde un EEG y transformarlos en un concepto pensado por medio de habla imaginada o una intención deseada de accionar un método de entrada de un equipo de cómputo, donde, para la presente investigación se tomará un conjunto de acciones reducido como las distintas intenciones posibles para la experimentación aquí planteada.

Así la problemática descrita es del tipo comparativa, buscando de entre los tres modelos previamente mencionados, aquel que tenga una mejor adaptabilidad para realizar una relación entre los valores que un EEG proporcione al algoritmo y un selecto grupo de palabras y acciones definidas que permitan hacer una evaluación a pequeña escala de intención de movimiento.

### 3. Justificación

Esta investigación cimenta su justificación – desde un punto de vista social – en la mejora de la interacción que existe entre el ser humano y la computadora, tratando de dar un paso más cerca en la dirección del uso de las interfaces cerebro – computadora como una vía más directa en la comunicación de los individuos y la operación por medio del cerebro, en este caso, de un equipo de cómputo.

Hay personas que por dificultades motrices les cuesta hacer un manejo óptimo de un equipo de cómputo. Desarrollando una interfaz cerebro – computadora que funcione adecuadamente podría ayudar a este sector de la población a tener una mejoría en el control de una computadora.

Es así también que, se espera que sea un paso más para permitir una mayor inclusión en el uso de los dispositivos computacionales para personas con ciertas discapacidades, que no les permitan usar de forma óptima métodos de entrada tradicionales como el teclado y el mouse; así, este tipo de usuarios podría operar el equipo por vías que les sean accesibles.

Por otra parte, desde un enfoque computacional, la justificación de esta investigación yace en analizar algunos algoritmos de aprendizaje supervisado con el objetivo de ponerlos a prueba y verificar cuál de ellos es el que mejor realiza la predicción de intención de movimiento dentro de un conjunto de acciones reducidas enfocadas al manejo de un sistema computacional al completo.

Cabe aclarar que aunque se intentó, esta investigación no avanza este agigantado paso, sino que trata de ser una piedra más en el camino para que el futuro de la investigación por esta vía siga acomodando el camino y que la justificación de la presente algún día sea concluida con mayor exploración en esta área y mayores descubrimientos que guíen por la senda que se matiza en este apartado de la presente investigación.

## 4. Objetivos

Los objetivos de la presente investigación se dividen en dos tipos, siendo los objetivos generales aquellos que dan una guía universal del proyecto al que se decidió embarcarse. Por otra parte, los objetivos particulares se fijaron más como metas que desglosan el objetivo general en pequeños puntos de control que idealmente se desean conseguir.

El objetivo general que se destaca es:

- Encontrar el modelo de aprendizaje supervisado, entre el Support Vector Machine, el modelo Random Forest y el modelo Naive Bayes, cuál se adapta con mayor eficiencia a las predicciones de los datos dados por una diadema Emotiv Epoc+ para interpretarlos como intención de movimiento de un conjunto de conceptos dados a los usuarios de prueba.

Así también, se establecieron los siguientes objetivos específicos logrados:

- I. Realizar una conexión eficiente entre el dispositivo Emotiv Epoc+ y un equipo computacional que permita extraer los datos de la actividad neuronal de un usuario.
- II. Adaptar las herramientas necesarias, tales como el programa desarrollado en Python por CymatiCorp: CyKit, el programa OpenViBE y el software de Emotiv, para que la recepción de datos del dispositivo que actúa como EEG sea realizada correctamente.
- III. Implementar el modelo Support Vector Machine, Random Forest y Naive Bayes para hacer las pruebas de aprendizaje sobre los datos de actividad neuronal.
- IV. Analizar cuál de los modelos previamente mencionados aprende con mayor eficiencia a convertir la actividad neuronal del usuario en una interfaz de comunicación eficiente entre éste y la computadora.

Se añaden también los objetivos específicos que en un principio eran deseables pero no se lograron:

- i. Comprender a nivel matemático qué diferencias entre cada modelo hacen notables en ciertos aspectos a cada tipo de técnica de aprendizaje supervisado previamente mencionado para construir un híbrido compuesto de las mejores características de las tres técnicas base.
- ii. Mejorar la interfaz realizada para que la transmisión de datos se realice en vivo en vez de realizar recopilación de datos y posterior entrenamiento y predicción del conjunto de datos.
- iii. Ampliar el conjunto de intenciones de prueba a todo el conjunto de teclas y funciones posibles con un teclado y un mouse, para la posterior implementación de la librería de Python: PyAutoGUI como método de control y operación de un equipo computacional a través de una interfaz cerebro – computadora.

Estos son los objetivos que fungen como brújula del proyecto en cuestión.

## 5. Hipótesis

Como hipótesis casi segura, siendo esta parte en la que más personal redacte, es un hecho que la conexión se hará correctamente. Por otra parte cuento con que investigaciones que menciono a detalle en el capítulo 7.4 dedicado al habla imaginada sean verídicas, dando a entender que vale la pena realizar mayor investigación en este campo pues parece que sí existe una correlación de la actividad cerebral con el habla imaginada y la intención de movimiento que algún individuo pueda estar llevando a cabo.

Dentro de mi hipótesis no creo que los modelos de machine learning no arrojen resultados relevantes. La interfaz será exitosamente desarrollada. Creo que el modelo que mejor funcionará será el Support Vector Machine debido a la experiencia previa en las clases que tengo y por las ventajas que creo que posee con las cuestiones que conozco entre los tres modelos, pues, pese a que no he visto mucho de las propuestas matemáticas de cada uno, este modelo fue desarrollado bajo vectores de soporte y pensando en sus ventajas vistas en clase puede ser el que mejor se adapte al funcionamiento del cerebro.



## 6. Pregunta de Investigación

En principio se esperaba que dos preguntas de investigación fueran las que marcaran el rumbo de la presente investigación. Al final, dado que una pregunta de investigación dependía de un objetivo específico que no consiguió concretarse, se descartó esta pregunta de investigación.

La pregunta de investigación que es el pilar principal de esta investigación es la siguiente:

- ¿Cuál modelo de aprendizaje supervisado entre el Support Vector Machine, Random Forest y Naive Bayes es el más eficiente para la predicción de intención de movimiento y posterior la integración a una interfaz cerebro – computadora orientado al control de un equipo de cómputo?

Por otra parte, se contemplaba la siguiente pregunta como parte continua a la investigación posterior a responder esta previa pregunta:

- Matemáticamente ¿qué ventajas poseen cada uno de los modelos anteriores y cómo podrían combinarse en un modelo híbrido computacional de aprendizaje supervisado?

Como se menciona anteriormente, esta pregunta fue descartada debido a la falta de tiempo para avanzar hasta este punto en la exploración de este campo, por lo que toda la investigación se sustenta con únicamente la primera pregunta mencionada en este apartado.

## 7. Marco Teórico

### 7.1. Introducción a la Neuroanatomía

*Soy un cerebro, Watson. El resto de mí es un mero apéndice.*  
Arthur Conan Doyle: La piedra de Mazarino.

#### 7.1.1. El cerebro y el sistema nervioso

El cerebro es un órgano muy interesante, que ha cautivado la atención de muchos investigadores y la curiosidad de las personas, pues como lo señala la corporación Caldaria en su blog HDOSO Magazine, “el cerebro es el órgano más complejo del cuerpo, al menos en los vertebrados, y lo es porque es este órgano el que controla el resto de órganos del cuerpo y el que, en definitiva, nos define como seres humanos” (Caldaria, 2020). Nazareno, estudiante de la Universidad de Harvard mientras escribía un artículo expresando su fascinación por el cerebro declara cómo a su percepción, este órgano es fascinante y los hallazgos sobre éste siguen asombrando a la comunidad científica enfocada en esta parte del cuerpo (Nazareno, 2020).

Hill define al cerebro desde una perspectiva etimológica “del latín cerebrum, con su raíz indoeuropea ker, cabeza, en lo alto de la cabeza y brum, ‘llevar’; teniendo el significado arcaico de «lo que lleva la cabeza». Es un órgano que centraliza la actividad del sistema nervioso y existe en la mayor parte de los animales” (Hill, 2006). Este órgano “es el más complejo del cuerpo. El telencéfalo adquiere su máximo desarrollo y está formado por los hemisferios cerebrales. El cerebro humano contiene en la corteza cerebral, un número estimado de 20,000,000,000 (20 mil millones,  $2 \times 10^{10}$ ) de neuronas” (von Bartheld, Bahney, & Herculano-Houzel, 2016), (Pelvig, Pakkenberg, Stark, & Pakkenberg, 2008), (Herculano-Houzel, 2009).

Puede deducirse gracias a las definiciones de los autores dadas previamente que el cerebro humano es el órgano central que recibe todo impulso nervioso que el cuerpo detecta y que tanto por su estructura como por su funcionamiento, es necesario dividirlo en algunas clasificaciones, como lo realizan Latarjet y Ruiz Liard, pues ellos señalan que está formado “por las estructuras derivadas del Telencéfalo y el Diencefalo, los dos sectores anteriores del Prosencéfalo embrionario. Ocupa el sector anterior y superior del cráneo llamados fosa craneal anterior y fosa craneal media” (Latarjet & Ruiz Liard, 2004).

Al ser la piedra angular del sistema nervioso, cabe definir a este grupo como “el conjunto de órganos que regulan, coordinan e integran todas las actividades del organismo. Asimismo, constituye una unidad funcional compleja que se puede dividir, desde el punto de vista didáctico, en dos componentes morfológicos fundamentales: el sistema nervioso central (SNC) y el sistema nervioso periférico (SNP)” (Ojeda Sahagún & Icardo de la Escalera, 2004).

Ojeda Sahagún e Icardo de la Escalera (2004) también señalan en su obra algunas otras consideraciones entre esta división, pues “el SNC agrupa todas las estructuras del sistema nervioso que se encuentran alojadas dentro del estuche osteofibroso formado por la cavidad craneal y el conducto vertebral. Por situarse en la línea media, a veces se denomina neuroeje” (Ojeda Sahagún, 2004). Por otra parte, el SNP, comentan Ojeda Sahagún e Icardo de la Escalera (2004), “comprende el resto de estructuras nerviosas que, aunque en su origen siguen un breve trayecto dentro de la cavidad craneal o del conducto vertebral, se sitúan fuera del estuche osteofibroso”.

El SNC se divide en encéfalo y médula espinal según Ojeda Sahagún (2004). Para efectos de la presente investigación, se centrará la atención en esta parte del sistema nervioso, específicamente en el encéfalo, que a su vez está constituido por el tronco del encéfalo, el cerebelo, el diencefalo y los hemisferios cerebrales. “El conjunto del diencefalo y los hemisferios cerebrales se denomina cerebro” (Ojeda Sahagún & Icardo de la Escalera, 2004) y se enfocará la atención en este grupo.

Con lo anterior mencionado, puede realizarse la siguiente definición funcional para efectos de esta investigación, pues el cerebro abarca muchos conceptos gracias a su alta complejidad, pero también los estudios analizados coinciden con que éste es el punto central del sistema nervioso y es nuestro principal procesador de información, el que recibe los estímulos nerviosos del exterior y los convierte en conocimiento a través de la percepción, la razón, las emociones y las otras formas de comprensión de nuestro entorno. Es capaz también de controlar los movimientos voluntarios, el habla y nuestro productor de inteligencia.

Aunque se sabe que el cerebro es capaz de realizar estas acciones, es también importante saber el cómo es capaz de realizar estas actividades. Para ello requerimos de analizar la estructura del órgano tanto a nivel general como a un nivel específico.

### **7.1.2. El cerebro a nivel macroestructura**

A un nivel de análisis amplio, Kandel, Schwartz y Jessel (2000) mencionan en su obra que el cerebro se divide en dos hemisferios cerebrales denominados hemisferio izquierdo y derecho, los cuales “son aproximadamente simétricos, sin embargo el izquierdo es ligeramente mayor. Están separados por la profunda cisura medial. Están cubiertos por una capa cortical sinuosa, la corteza cerebral, formada por sustancia gris” (Kandel, Schwartz, & Jessel, 2000).

Anatomistas como los anteriormente mencionados clasifican cada hemisferio en seis lóbulos: lóbulo frontal, lóbulo parietal, lóbulo occipital, lóbulo temporal, lóbulo insular y lóbulo límbico. En la figura 7.1.1 puede observarse el cerebro y la división de los lóbulos que son visibles desde la corteza cerebral. La división de esta forma del cerebro permite centrarse en las diferentes secciones sabiendo el hecho de que “cada lóbulo cerebral presente una serie de características

propias no significa que cada estructura controle casi en “exclusiva” una determinada tarea. Muchas actividades y procesos se superponen a través de las distintas regiones cerebrales” (Sabater, 2020).

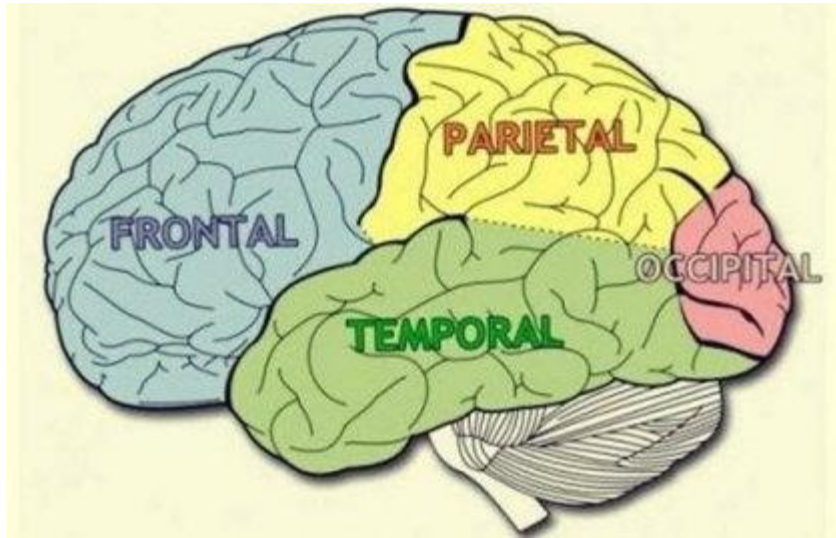


Figura 7.1.1. Los cuatro lóbulos cerebrales que pueden observarse desde la corteza cerebral. (Sabater, 2020).

Comenzando por analizar los lóbulos frontales, éstos poseen grandes muestras de la evolución humana, pues “situados en la parte frontal de la cabeza, y justo debajo de los huesos frontales del cráneo y cerca de la frente, conforman la región más afinada de nuestro cerebro, la que más tiempo tardó en evolucionar y aparecer” (Sabater, 2020). Sabater también menciona que las tareas más notorias que este lóbulo realiza son las siguientes:

- La producción de habla y lenguaje.
- Procesos cognitivos que nos permiten planificar, fijar la atención, memorizar datos a largo plazo, comprender lo que vemos, regular nuestras emociones, etc.
- Comprender y reaccionar ante los sentimientos de los demás, como por ejemplo, la empatía.
- Regulación de la motivación y búsqueda de recompensas.

El siguiente lóbulo se trata del lóbulo parietal. Esta parte del cerebro se caracteriza por su importante papel en la percepción de los sentidos, en el razonamiento espacial, el movimiento de nuestro cuerpo y nuestra orientación. Tal como lo dice Sabater (2020), pues es en esta área donde “se capta la información sensorial relativa a la mayoría de nuestros órganos sensoriales. Es aquí donde se procesa y regula la sensación del dolor, la presión física y la temperatura, etc.” (Sabater, 2020). Gracias al área parietal, también es posible comprender la naturaleza de los números y la relación que encontramos con las matemáticas.

El siguiente lóbulo es el lóbulo occipital. Este es el lóbulo más pequeño y aunque se considera un camino de paso de los demás procesos, Sabater (2020) señala que este lóbulo sí realiza algunos procesos importantes como los siguientes:

- Participa en los procesos de percepción y reconocimiento visual.
- Tiene alta importancia en todo lo relativo al sentido de visión, ya que su corteza integra diversas áreas visuales como la detección de patrones, procesamiento de información y envío de esta información a otras áreas del encéfalo.
- Ayuda en la diferenciación de colores.
- Participa también en la elaboración de las emociones y pensamientos.

Dentro de este conjunto tenemos también al lóbulo temporal. Esta parte del cerebro, según Sabater (2020) se encarga especialmente de las siguientes tareas:

- Nos ayuda a reconocer los rostros.
- Tiene estrecha relación con la articulación del lenguaje, la comprensión de sonidos, voces y la música.
- Facilita el equilibrio.
- Participa en la regulación de emociones como la motivación, la rabia, la ansiedad, el placer, entre otras.

Aunque también destaca la autora que es “muy complicado asociar a cada una de estas estructuras a una única función especializada. Todas dependen unas de otras, todas se hallan conectadas y favorecen esa armonía perfecta” (Sabater, 2020), de modo que todos los lóbulos se desempeñen juntos en consonancia.

Sabater menciona que, gracias a los estudios del cerebro realizados, se ha hecho el hallazgo de la quinta región: el lóbulo insular o la ínsula lobular; descrita por esta autora como “un lóbulo oculto justo debajo de los lóbulos temporal, frontal y parietal. Es un área muy recóndita y de complejo acceso localizada entre los vasos venosos y arterias” (Sabater, 2020). Puede observarse una muestra gráfica de su ubicación en la figura 7.1.2.

Sabater menciona que, pese a que no se sabe a ciencia cierta cuáles son sus funciones, gracias a otros procesos y correlaciones, el lóbulo insular participa en el sentido del gusto, el control visceral, la somatopercepción y otros procesos emocionales de los cuales forma parte junto con el sistema límbico. Otros autores comparten este hecho de la quinta región, pues como menciona Laguna, “el lóbulo de la ínsula es el que menos se ha estudiado. Por esta razón, es considerado un lóbulo nuevo para los estudiantes de ciencias de la salud ya que hay muy poca información sobre esta estructura” (Laguna, 2022).

Sin embargo, Laguna menciona otras funciones de esta área del cerebro, encontradas gracias a que “diversos estudios de neuroimagen han relacionado al lóbulo de la ínsula con los deseos, los antojos y las adicciones. Asimismo, se ha evidenciado que este lóbulo juega un papel de suma importancia en cuanto a los trastornos psiquiátricos, tales como la esquizofrenia, los trastornos de pánico, el estrés postraumático y el trastorno obsesivo-compulsivo” (Laguna, 2022).

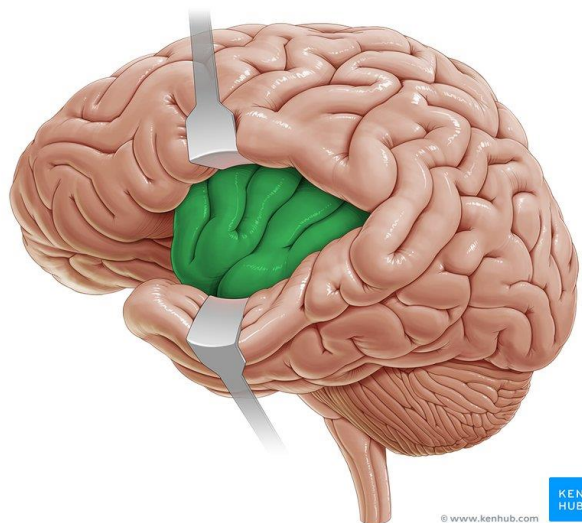


Figura 7.1.2. Representación gráfica de la ubicación del lóbulo insular, señalada en color verde. (Laguna, 2022).

Finalmente se tiene el sexto lóbulo: el lóbulo límbico. Éste “comprende una serie de estructuras situadas en la zona medial de los hemisferios cerebrales. Interviene principalmente en la expresión de afectos y en la memoria” (Interpsiquis, 2022). Estos autores mencionan las estructuras que forman parte de este lóbulo, mismas que pueden observarse en la figura 7.1.3, siendo las siguientes:

- Estructuras corticales: cíngulo y giro parahipocámpico.
- Formaciones hipocámpicas: giro dentado, hipocampo y complejo subicular.
- La amígdala.
- El núcleo *accumbens*.
- El hipotálamo.
- El tálamo (núcleo anterior y núcleo dorsomedial).
- Otras estructuras corticales como el orbitofrontal y el polo temporal.

## Sistema Límbico

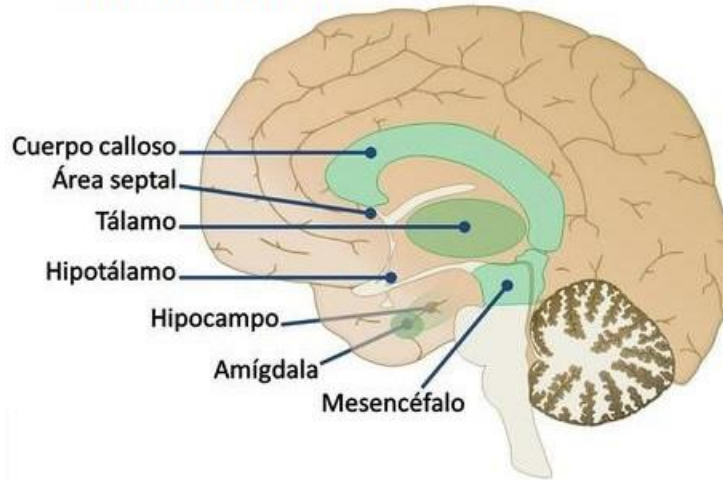


Figura 7.1.3. Estructuras que conforman el lóbulo límbico. (Triglia, 2016).

Con base en las definiciones y proposiciones de los autores previos, es posible enumerar las principales actividades de cada uno de los lóbulos cerebrales; así, serían las siguientes:

- **Lóbulo frontal:** Trabaja principalmente en el habla y el lenguaje, ayuda en fijar la atención y la memoria a largo plazo. Regula la motivación y búsqueda de recompensas, así como la empatía.
- **Lóbulo parietal:** Tiene alta importancia en la percepción de los sentidos, razonamiento espacial, percepción de nuestro cuerpo – también conocida como la percepción somatosensorial – y orientación.
- **Lóbulo occipital:** Participa activamente en la percepción y reconocimiento visual. Detecta patrones y procesa la información para enviarla a otras áreas del encéfalo.
- **Lóbulo temporal:** Actúa en el reconocimiento de rostros, está altamente relacionada con la articulación del lenguaje, comprensión de sonidos, voces y música. Actúa también como soporte en la regulación de emociones como la motivación, placer y ansiedad.
- **Lóbulo insular:** Es parte importante del sentido del gusto, control visceral. También participa en la somatopercepción.
- **Lóbulo límbico:** Actúa en la expresión de afectos y la memoria.

Es claro entonces, que el cerebro posee ciertas divisiones funcionales que se encargan en mayor medida de alguna tarea. Éstas son descritas por Gray (2002) en su obra donde reconoce como divisiones funcionales al área sensorial primaria, el área motora primaria y las áreas de asociación.

El área sensorial primaria incluye “el área visual del lóbulo occipital, el área auditiva primaria en el lóbulo temporal y la corteza insular, y el área somatosensorial en el lóbulo parietal” (Gray, 2002). Esta área recibe señales de los nervios sensoriales y las envía a través del tálamo. El área de asociación la define Gray (2002) como la receptora de información entrante de áreas sensoriales y está implicada en el proceso de percepción, pensamiento y toma de decisiones.

Finalmente, se encuentra el área motora primaria, la cual, según Gray (2002), ocupa la parte posterior del lóbulo frontal, delante del área somatosensorial. Esta área envía los axones hasta las neuronas del tronco encefálico y la médula espinal para realizar sus tareas. Sin embargo, hablar de axones y neuronas implica entrar en el terreno de la microarquitectura cerebral, pero de esta forma será posible entender qué “señales” envía el cerebro para la intercomunicación corporal e incluso entre el mismo órgano. Comprender este aspecto será de mayor utilidad en capítulos posteriores de la presente investigación.

### **7.1.3. El cerebro a nivel microestructura**

Es momento de hablar sobre la neurona; componente del sistema nervioso a un nivel de escala menor al de los anteriores constituyentes del sistema nervioso. Las neuronas son parte importante del sistema nervioso, pues incluso “se pueden distinguir dos grupos celulares básicos: células propias del SNC y células comunes con otros sistemas del organismo, como las células endoteliales de los capilares sanguíneos. Las células propias del SNC son de dos tipos diferentes: a) células excitables, denominadas neuronas y b) células no excitables, que incluyen la neuroglia y las células ependimarias” (Ojeda Sahagún & Icardo de la Escalera, 2004).

#### **7.1.3.1. Las neuronas**

Las neuronas son un componente principal del sistema nervioso, cuya función principal es recibir, procesar y transmitir información a través de señales químicas y eléctricas gracias a la excitabilidad eléctrica de su membrana plasmática. Las neuronas “son las unidades estructurales y funcionales del sistema nervioso. Son células excitables especializadas en la recepción, integración, transformación y transmisión en una sola dirección (conducción ortodrómica) de información codificada por cambios electroquímicos en su membrana plasmática” (Ojeda Sahagún & Icardo de la Escalera, 2004).

Estas células “están especializadas en la recepción de estímulos y conducción del impulso nervioso (en forma de potencial de acción) entre ellas mediante conexiones llamadas sinapsis, o con otros tipos de células como, por ejemplo, las fibras musculares de la placa motora. Altamente diferenciadas, la mayoría de las neuronas no se dividen una vez alcanzada su madurez; no obstante, una minoría sí lo hace” (Cayre, Malaterre, Scotto-Lomassese, Strambi, & Strambi, 2010).



Se han realizado estimaciones que señalan que, “el encéfalo humano contiene 86,000,000,000 (ochenta y seis mil millones.  $8.6 \times 10^{10}$ ) de neuronas, de las cuales cerca de 10,000,000,000 (diez mil millones.  $1 \times 10^{10}$ ) son células piramidales corticales. Estas células transmiten las señales a través de mil billones ( $10^{15}$ ) de conexiones sinápticas” (Murre & Sturdy, 1995).

### 7.1.3.2. Morfología de las neuronas

Respecto a la morfología de las neuronas, diferentes autores hacen señalizaciones distintas a la composición de estas células. Ojeda Sahagún e Icardo de la Escalera (2004) mencionan que las neuronas “se caracterizan por poseer una gran superficie celular, lo que les permite llevar a cabo sus funciones específicas. El aumento de la superficie celular se consigue mediante la presencia de expansiones ramificadas que parten del cuerpo celular o soma neuronal y que se denominan neuritas (axón y dendritas). Ciertos tipos de neuronas se caracterizan por la presencia en su citoplasma de pigmentos de diferentes tipos, por lo que sus agrupaciones presentan una coloración característica”. Ellos también destacan como los elementos morfológicos más importantes al soma neuronal, las dendritas y al axón o también denominado cilindroeje.

Aunque otros autores destacan de la morfología de las neuronas “un núcleo voluminoso central, situado en el soma; un pericarion que alberga los orgánulos celulares típicos de cualquier célula eucariota; y neuritas (esto es, generalmente un axón y varias dendritas) que emergen del pericarion” (Paniagua, y otros, 2002).

Incluyendo la morfología que propone Merck & Co Inc. (2022), en la cual propone como componentes principales el cuerpo celular o también llamado soma neuronal, las dendritas y axones, así como las vainas de mielina de las neuronas. Estos componentes pueden observarse en la figura 7.1.4 con una observación gráfica de la morfología de las neuronas.

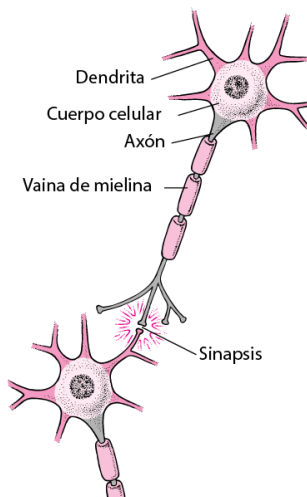


Figura 7.1.4. Morfología de una neurona. (Merck & Co, 2022).

A destacar entre estos componentes morfológicos de la neurona, el primero será el soma neuronal. Esta estructura está localizada en cada una de las neuronas del cerebro, la médula espinal y también en los ganglios espinales del sistema nervioso periférico. La función del soma neuronal es “mantener al núcleo junto con la información genética (ADN), contener a las proteínas del retículo endoplasmático y además es el que genera la energía necesaria para que la neurona trabaje correctamente” (Fisio, 2022). Gracias a esta definición, dado que otros autores definen al cuerpo celular como el encargado de estas funciones, esta parte de la célula recibe ambos nombres, tanto soma neuronal como cuerpo celular.

Es, de hecho, en el soma neuronal donde se asienta el núcleo celular. Tal como lo señalan Ojeda Sahagún e Icardo de la Escalera (2004), pues este elemento “constituye el centro para el crecimiento y mantenimiento de las neuritas y el lugar de la síntesis proteica. El tamaño del soma y del núcleo neuronal se encuentra en relación directa con el número y longitud de neuritas”.

Para comprender la estructura anterior hay que entender qué es una neurita. Kevin Flynn lo define de la manera más concisa, pero adecuada para los términos necesarios en esta investigación; señalando que éstas son “cualquier proyección del soma de una neurona, ya sea una dendrita o un axón. El término se usa con frecuencia al hablar de células nerviosas inmaduras o en desarrollo” (Flynn, 2013). Es por ello que bajo este término, Ojeda Sahagún e Icardo de la Escalera engloban a los axones y las dendritas.

Las neuronas son muy similares a células de otros tejidos en el aspecto de su estructura y organización de su núcleo celular encontrado en el cuerpo de la neurona, pero incluso así, Ojeda Sahagún e Icardo de la Escalera (2004) clasifican en tres tipos de estructuras características al soma neuronal, las cuales son las siguientes:

- Cuerpos de Nissl.
- Neurofilamentos.
- Neurotúbulos.

Otro elemento importante de las neuronas son las dendritas. Las dendritas son “ramificaciones que proceden del soma neuronal que consiste en proyecciones citoplasmáticas envueltas por una membrana plasmática sin envoltura de mielina. En ocasiones, poseen un contorno irregular, desarrollando espinas. Sus orgánulos y componentes característicos son: muchos microtúbulos y pocos neurofilamentos, ambos dispuestos en haces paralelos; además muchas mitocondrias; grumos de Nissl, más abundantes en la zona adyacente al soma; retículo endoplasmático liso, especialmente en forma de vesículas relacionadas con la sinapsis” (Wikipedia, 2022).

Asimismo, éstas también se definen como “prolongaciones celulares que se originan a partir del soma neuronal. Constituyen la porción receptiva de las neuronas y suelen ser muy numerosas. Las dendritas que se originan directamente del soma neuronal se denominan dendritas primarias o principales. Éstas se ramifican repetidamente, por lo general de forma dicotómica, dando origen a las dendritas secundarias. El conjunto de dendritas primarias y sus ramificaciones se conoce como árbol dendrítico” (Ojeda Sahagún & Icardo de la Escalera, 2004).

Estas prolongaciones de las neuronas están “dedicadas principalmente a la recepción de estímulos y, secundariamente, a la alimentación celular” (Roche, 2003). Éstas tienen quimiorreceptores que son capaces de reaccionar con los neurotransmisores que son enviados desde los extremos de los axones y siendo altamente importantes para la correcta transmisión de los impulsos quimioeléctricos a través de la vía nerviosa.

Las dendritas contienen la mayor parte de organelos que están presentes en el soma neuronal, entre ellos el retículo endoplásmico; otra característica es que son más cortas, gruesas y rugosas que el axón. “Las rugosidades se deben a la presencia de numerosas expansiones cortas, las espinas dendríticas que constituyen puntos donde una neurona entra en contacto con otras. Las dendritas no poseen envolturas especializadas, como ocurre en el caso de los axones” (Ojeda Sahagún & Icardo de la Escalera, 2004).

Otra parte importante de la neurona es el axón. El axón es definido en algunas ocasiones como “una delgada y extensa prolongación del soma neuronal, que está rodeado por su membrana, el axolema. El axolema puede estar recubierto por células de Schwann en el sistema nervioso periférico de vertebrados, con producción o no de mielina. Puede dividirse, de forma centrífuga al pericarion, en tres sectores: el cono axónico, el segmento inicial y el resto del axón” (Paniagua, y otros, 2002). Es, de hecho, destacado por Ojeda Sahagún e Icardo de la Escalera (2004) como la prolongación más importante del soma neuronal.

Algunas características notables de los axones, descritas en la obra de Ojeda Sahagún e Icardo de la Escalera (2004), son las siguientes:

- a) El axón carece de la maquinaria precisa para realizar la síntesis proteica, de modo que los neurotransmisores y otros materiales deben ser transportados continuamente desde el soma neuronal a las terminaciones axonales, lo que constituye el transporte o flujo axonal anterógrado; también existe un movimiento en sentido contrario de menor intensidad llamado flujo axonal retrógrado.
- b) A excepción del segmento inicial, el cual es la zona de origen del soma neuronal, la superficie del axón se encuentra cubierta por una vaina de mielina o por una envoltura

celular, recibiendo el nombre de axones mielinizados o amielínicos respectivamente. La envoltura celular (para el caso del sistema nervioso central) es de oligodendroglia.

Cabe resaltar que la vaina de mielina la cual rodea al axón “no es continua, sino que se encuentra interrumpida de manera periódica. En estas zonas desnudas, denominadas nódulos de Ranvier, el axón puede poseer ramas colaterales. Los axones suelen terminar ramificándose en varias prolongaciones, las terminaciones axonales o telodendrias, que finalmente establecen contactos o sinapsis con otras neuronas o con células efectoras (músculo o glándulas)” (Ojeda Sahagún & Icardo de la Escalera, 2004).

Este mismo conjunto de autores, junto con otros especialistas en el área, describen también los tres sectores anteriormente mencionados:

- **Cono axónico.** “Adyacente al pericarion, es muy visible en las neuronas de gran tamaño. En él se observa la progresiva desaparición de los grumos de Nissl y la abundancia de microtúbulos y neurofilamentos que, en esta zona, se organizan en haces paralelos que se proyectarán a lo largo del axón” (Paniagua, y otros, 2002).
- **Segmento inicial del axón.** “En él comienza la mielinización externa. En el citoplasma, a esa altura se detecta una zona rica en material electrodense en continuidad con la membrana plasmática, constituido por material filamentoso y partículas densas. La membrana se continúa con el axolema y se asume que este sector interviene en la generación del potencial de acción que transmitirá la señal sináptica” (Kole & Stuart, 2012).
- **Resto del axón.** Según Paniagua y los autores que apoyaron su obra (2002), es en este sector donde aparecen los nódulos de Ranvier y las sinapsis.

Con base en los autores destacados previamente es posible obtener una definición propia útil para el objetivo de la presente investigación, pues la importancia de estas células en la presente, radica en su composición y comunicación. Para este caso, cabe resaltar que una neurona está compuesta del soma neuronal, las dendritas y los axones, principales componentes de estas células, donde todos estos son importantes para la comunicación entre estas células a través de actividades como la transmisión de potenciales eléctricos y síntesis proteica.

### 7.1.3.3. Funciones de las neuronas

Sin embargo, hablar de la comunicación entre neuronas requiere también profundizar en las funciones que las neuronas cumplen. No sólo es necesaria la comprensión de estos tópicos, sino que, aunque podría parecer de menor importancia mencionar a algunos investigadores en esta área, para esta investigación tendrán alta relevancia, pues no sólo estarán

involucrados en la investigación de estos procesos neuronales, también lo están al hablar de modelos matemáticos de redes que formaron para explicar el complejo sistema de neuronas humano; pero previo a los temas de modelado, será primordial entender las funciones neuronales, explicadas a continuación.

La principal función de las neuronas es comunicarse entre sí. Estas células poseen una capacidad de realizar esta tarea con precisión, rapidez y a larga distancia, sin importar si éstas son de carácter nervioso, muscular o glandular. Es a través de estas células por donde se transmiten señales eléctricas a las cuales se les denomina impulsos nerviosos, los cuales “viajan por toda la neurona comenzando por las dendritas hasta llegar a los botones terminales, que se pueden conectar con otra neurona, fibras musculares o glándulas” (Wikipedia, 2022). Esta conexión entre una neurona y otra es lo que recibe el nombre de sinapsis.

Es de hecho la sinapsis la unidad más simple de funcionamiento nervioso segmentario, la cual requiere dos neuronas: una neurona sensitiva receptora y una neurona motora o efectora; así lo explican Afifi y Bergman (1998), quienes también mencionan que “el acoplamiento anatomofuncional de estas dos neuronas ocurre a través de lo que se llama sinapsis. Las arborizaciones terminales de la neurona sensitiva (axones) se dilatan en pequeños brotes o botones (llamados *boutons terminaux*, un término acuñado por un investigador francés), que yacen en contacto con las dendritas, cuerpos celulares o axones de las neuronas efectoras” (Afifi & Bergman, 1998).

Las neuronas “conforman e interconectan los tres componentes del sistema nervioso: sensitivo, motor e integrador o mixto; de esta manera, un estímulo que es captado en alguna región sensorial entrega cierta información que es conducida a través de las neuronas y es analizada por el componente integrador, el cual puede elaborar una respuesta, cuya señal es conducida a través de las neuronas. Dicha respuesta es ejecutada mediante una acción motora, como la contracción muscular o secreción glandular” (Wikipedia, 2022).

La primera función destacable de las neuronas es el impulso nervioso, en el cual las neuronas transmiten ondas eléctricas causadas por un cambio transitorio en la membrana plasmática. La propagación de estas ondas se debe a “la existencia de una diferencia de potencial o potencial de membrana entre la parte interna y externa de la célula, que surge gracias a las concentraciones distintas de iones a ambos lados de la membrana, según describe el potencial de Nernst” (Cromer, 1996). Usualmente esta diferencia de potencial, según Cromer (1996), es de  $-70$  milivolts (mV).

La carga de una célula inactiva se mantiene en valores negativos, pues se trata de la carga del interior respecto al exterior, y varía de unos estrechos márgenes. “Cuando el potencial de

membrana de una célula excitable se despolariza más allá de un cierto umbral (de 65 mV a 55 mV aproximadamente) la célula genera un potencial de acción. Un potencial de acción es un cambio muy rápido en la polaridad de la membrana de negativo a positivo y vuelta a negativo, en un ciclo que dura milisegundos” (Bear, Connors, & Paradiso, 2002).

El proceso del potencial de acción puede verse ilustrado en la figura 7.1.5, que muestra el voltaje en función del tiempo transcurrido, observando el cambio que ocurre cuando esta función es realizada.

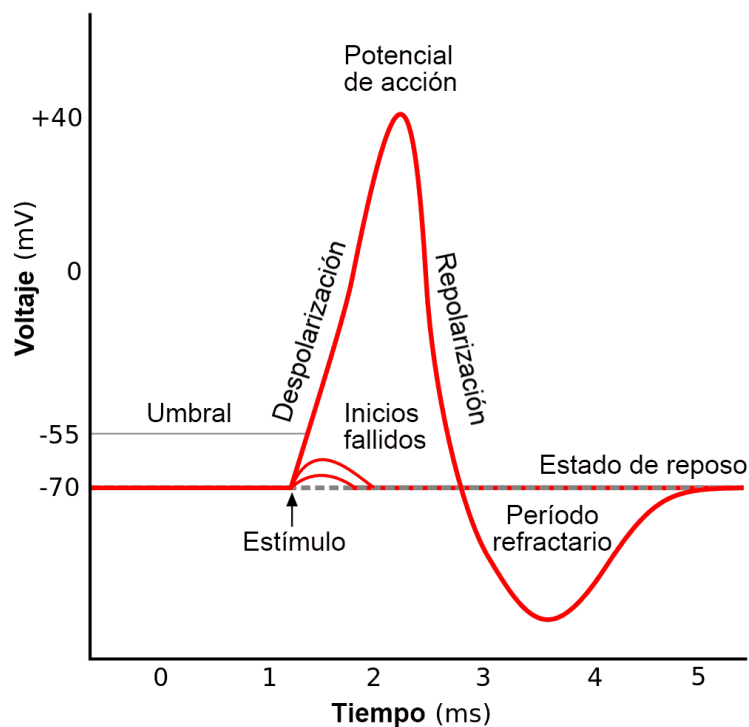


Figura 7.1.5. Gráfica del potencial de acción de una neurona en función del tiempo. (MDurance, 2021).

La siguiente función de las neuronas será en la que se hará mayor énfasis en este documento, pues se trata de las corrientes iónicas que ocurren durante el potencial de acción, siendo de alta importancia para la problemática actual, también se debe a que el primer registro de un potencial de acción fue realizado por los investigadores Alan Lloyd Hodgkin y Andrew Fielding Huxley.

Estos nombres no sólo son notables por el estudio de esta función de las neuronas, sino que también realizaron un conjunto de ecuaciones diferenciales ordinarias no lineales las cuales aproximan las características eléctricas de las neuronas (en general, de las células excitables) proponiendo el modelo en 1952 y posteriormente obteniendo el Premio Nobel en fisiología – medicina por esta investigación en 1963 tras investigar y registrar esta exploración experimentando con el axón de un calamar gigante (Hodgkin & Huxley, 1939).

El potencial de acción comprende tres fases:

- 1) Potencial en reposo o potencial de membrana, llevándose a cabo una permeabilidad al sodio y al potasio.
- 2) Despolarización de la membrana celular al sodio y al potasio.
- 3) Repolarización de la membrana al sodio y al potasio.

Hodgkin y Huxley midieron la conductancia durante la despolarización y repolarización observando cambios de conductancia para el sodio (Na) y el potasio (K) durante el potencial de acción (Hodgkin & Huxley, 1939).

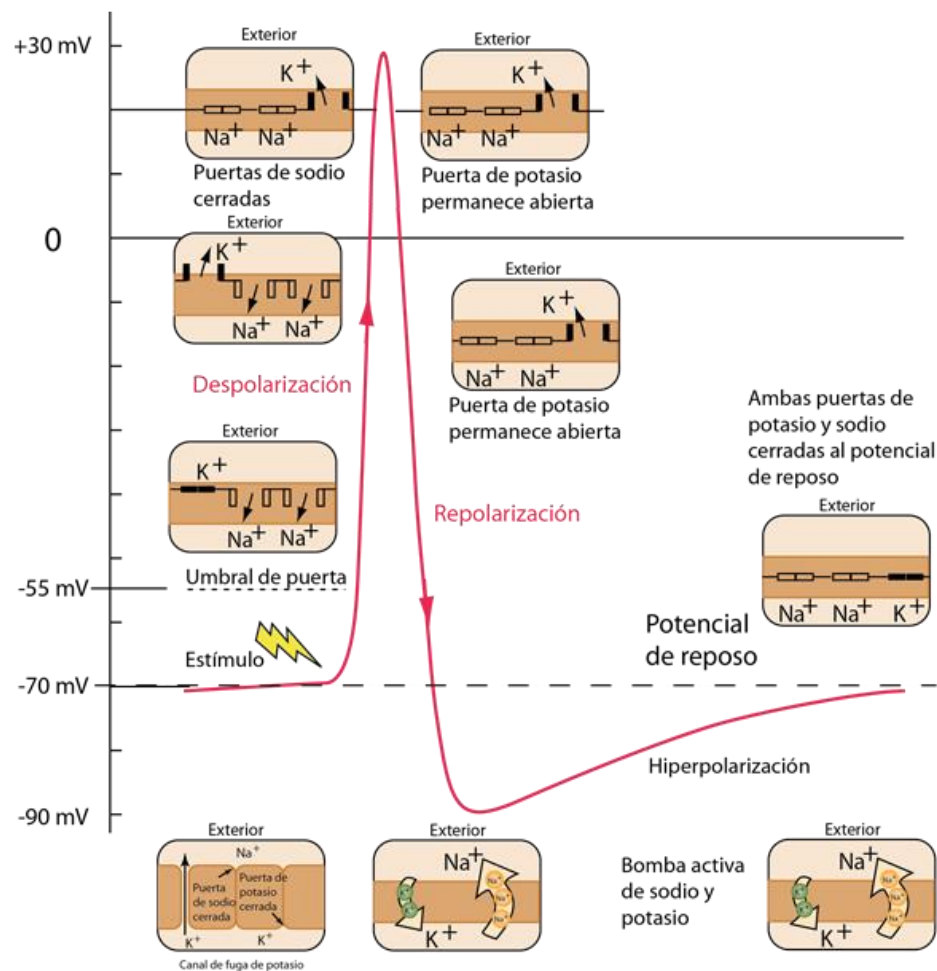


Figura 7.1.6. Gráfica del potencial de acción de una neurona mostrando la interacción iónica. (Olmo, Nave, & Nave, 2022)

Al observar los cambios previos puede determinarse la diferencia de potencial por la diferencia absoluta entre las cargas positivas y negativas entre el interior y el exterior con relación a la membrana. Sin embargo, como a continuación se tratarán las cargas eléctricas de las moléculas que atraviesan la membrana, para entender estos conceptos es importante

destacar que estas cargas se originan gracias a los iones; átomos o moléculas “cargadas eléctricamente de modo que ya no son eléctricamente neutras” (Ebbing & Gammon, 2010). De este modo, puede haber dos tipos de iones los cuales también describen Ebbing y Gammon, que son los aniones y los cationes. Un anión es un ion con una ganancia de electrones o como se conoce, un ion con carga negativa, así entonces, un catión es un ion con pérdida de electrones, es decir, un ion con carga positiva.

Lo anterior es necesario explicarlo y entenderlo porque de hecho, Wikipedia (2022) en su sitio especializado para la neurona, explica que la diferencia durante el potencial de acción se obtiene por la carga aniónica y catiónica – es decir, por la carga negativa y positiva – de los iones involucrados. A destacar, el anión participante en este proceso es el cloro ( $\text{Cl}^-$ ); por otra parte, los cationes de mayor reconocimiento son el potasio ( $\text{K}^+$ ), magnesio ( $\text{Mg}^{2+}$ ), calcio ( $\text{Ca}^{2+}$ ) y sodio ( $\text{Na}^+$ ). Este procedimiento puede observarse en la figura 7.1.6.

Ahora, cuando un canal iónico es abierto, “el tránsito iónico es a favor de su gradiente electroquímico, esto es, pretende equilibrar el número de iones, independientemente del potencial transmembrana actual. Este mecanismo circunstancial de movimiento iónico permite el tránsito entre estados de polarización y despolarización” (Wikipedia, 2022).

Esto anterior con el objetivo de equilibrar la diferencia electroquímica que existe tanto al exterior como al interior de la membrana, siguiendo, claro, la definición de gradiente en matemáticas, pues se trata de “la variación del valor de una magnitud en dos puntos y la distancia que los separa” (Sáenz, 2013), ya que la definición usada de gradiente electroquímico previamente combina tanto este concepto como su aplicación dentro de la electroquímica, siendo ésta la diferencia iónica en los dos puntos de la membrana anteriormente mencionados.

Este procedimiento fue usado para modelar la propuesta de Hodgkin y Huxley, la cual se retomará más adelante desde un punto de vista más analítico. Finalmente, Wikipedia (2022) menciona otro grupo de funciones importantes de las neuronas, a saber:

- Propiedades electrofisiológicas intrínsecas.
- Neurosecreción.
- Neurodegeneración.

Las anteriores sólo se mencionan de manera superficial dado que no contribuyen en gran medida para solucionar el problema de la presente investigación.

Ha sido un largo camino de neuroanatomía que ha habido que recorrer, sin embargo, es necesario revisar dos tópicos más antes de avanzar al siguiente capítulo y acercarse un poco



más al problema que se desea abordar en esta investigación. Estos dos temas son la interacción entre neuronas y, por consiguiente, también las conexiones sinápticas.

Ya son del conocimiento del lector conceptos relacionados al cerebro, como su división a macroescala, así como su funcionamiento a microescala. De esta forma se ha revisado el componente principal de este capítulo: la neurona; observando su morfología y funciones, pero también es importante integrar la neurona con más de este tipo de células y explicar la interacción entre estas células.

#### **7.1.3.4. Interacción entre neuronas**

Para comenzar con la interacción entre neuronas, es necesario comprender que “el sistema nervioso procesa la información siguiendo un circuito más o menos estándar. La señal se inicia cuando una neurona sensorial recibe un estímulo externo. Su axón se denomina fibra aferente. Esta neurona sensorial transmite una señal a otra aleadaña, de modo que acceda un centro de integración del sistema nervioso. Las interneuronas, situadas en dicho sistema, transportan la señal a través de sinapsis. Finalmente, si debe existir respuesta, se excitan neuronas eferentes que controlan músculos, glándulas u otras estructuras anatómicas. Las neuronas aferentes y eferentes, junto con las interneuronas, constituyen el circuito neuronal” (Randall, Burggren, & French, 1998).

La interacción entre neuronas se da gracias a toda la red de neuronas que están interconectadas. Es así que Randall, Burggren y French (1998) realizan definiciones de estos conceptos, como el mencionado circuito neuronal, el cual se trata de “un conjunto de conexiones sinápticas ordenadas que se produce como resultado de la unión de las neuronas a otras en sus regiones correspondientes tras la migración neuronal”.

Mencionando también que “el crecimiento dirigido de los axones y el reconocimiento de las estructuras sinápticas está mediado por el cono de crecimiento, que es una especialización en el extremo de cada axón en crecimiento. El cono de crecimiento detecta y responde a moléculas de señalización que pueden ser de retraimiento, giro o continuación, que identifican las vías correctas, prohíben las incorrectas y facilitan la formación de sinapsis” (Randall, Burggren, & French, 1998).

Cuando se habla de una red neuronal biológica se destaca que “la función de un determinado grupo de neuronas es alcanzar un determinado estado final en función de los estímulos externos. Por ejemplo, en la percepción del color, un grupo de neuronas puede encargarse de acabar en un determinado estado si el estímulo es "rojo" y otro determinado estado si el estímulo es "verde". El número de "estados estables" posibles del circuito neuronal se corresponde con el número de patrones (en este caso colores diferentes) que puede reconocer el circuito neuronal. Los trabajos de Freeman en los años 1990 aclararon que un determinado

grupo de neuronas sigue un patrón de evolución temporal caótico hasta alcanzar un determinado estado” (Solé & Manrubia, 1996).

Solé y Manrubia (1996) profundizan mucho en estos tópicos de circuitos neuronales. De hecho también entran en análisis de modelos matemáticos, pues por ejemplo, estudian algunos temas de estas redes como la teoría de Hopfield y la regla de Hebb pues de estos dos argumentos mencionan que “estiman la relación entre el número de neuronas  $N$  que intervienen en reconocer  $p$  patrones y la probabilidad de error  $P_e$  en el reconocimiento de patrones”.

Sin embargo, la teoría de Hopfield también se trata de un modelo matemático y ciertamente en este capítulo ya se están tocando temas que corresponden a apartados posteriores. Aun así, los conocimientos de neuroanatomía han sido cubiertos en su totalidad y todos y cada uno de los conceptos vistos en este capítulo serán de alta importancia en los posteriores.

Aunque ya se están tocando temas de matemáticas, antes es necesario conocer cómo es posible obtener la actividad eléctrica cerebral, de qué forma se puede clasificar la actividad una vez que se capture y cuáles serían los aparatos con los que se obtendrían estos datos. Estos conceptos y más relacionados serán del conocimiento del lector en el próximo capítulo relacionado a la electroencefalografía.

## 7.2. Electroencefalografía y Estudios del Cerebro

*Todo hombre puede ser, si se lo propone, escultor de su propio cerebro*  
Santiago Ramón y Cajal

Antes de avanzar con algunos conocimientos computacionales, mantendremos la línea teórica centrada en el cerebro, pero hablando de un dispositivo importante para el desarrollo de esta investigación el cual sustenta la comprensión de un área conocida como la electroencefalografía.

### 7.2.1. El Electroencefalograma

La herramienta principal de la electroencefalografía es el electroencefalograma. Éste puede definirse como la herramienta que permite realizar una exploración neurofisiológica basándose en el registro de la actividad bioeléctrica cerebral. Propiamente puede definirse como “un estudio de la función cerebral que recoge la actividad eléctrica del cerebro en situación basal y con métodos de activación, como la hiperventilación y la fotoestimulación” (Urrestarazu, 2022). Este estudio recopila la señal eléctrica la cual posteriormente se amplifica en forma de líneas, interpretándose la actividad de las distintas áreas cerebrales a lo largo del tiempo. Un ejemplo de éste se aprecia en la figura 7.2.1.



Figura 7.2.1. Ejemplo de electroencefalograma (Sosa Romano, 2022).

El estudio con un electroencefalograma, o desde ahora llamado EEG, permite observar patrones normales o anormales del comportamiento cerebral que pueden hacer sospechar lesiones o enfermedades características. Un EEG sirve para “para observar el funcionamiento eléctrico

cerebral. Es de interés conocer su normalidad o no en pacientes con alteración de las funciones cerebrales, bien de forma persistente o bien de modo episódico” (Urrestarazu, 2022).

Debido a esto, el estudio con un EEG es “un medio de diagnóstico funcional de enfermedades cerebrales complementario a otros estudios, especialmente los radiológicos” (Urrestarazu, 2022) como es el caso de la resonancia magnética.

Un EEG puede detectar alteraciones de todo el cerebro o de algunas zonas. Esto es útil para revisar alteraciones en lesiones, como tumores, hemorragias, encefalitis, traumatismos u otro tipo, así como lesiones difusas tóxicas, metabólicas o infecciosas.

Una prueba EEG, según Urrestarazu (2022), normalmente arroja las siguientes indicaciones con mayor frecuencia:

- Cefaleas.
- Epilepsia.
- Trastornos del movimiento
- Trastornos del sueño

### 7.2.2. Breve Historia del Electroencefalograma

Wikipedia (s.f.) en su sitio “Electroencefalografía” plasman una breve historia del electroencefalograma. Ellos denotan como primer hecho importante los hallazgos presentados por Richard Birmick Caton sobre los fenómenos bioeléctricos en los hemisferios cerebrales de ratones y monos en 1875.

El siguiente hecho rescatable en el sitio previamente mencionado data de 1912 donde Vladimir Pravdich-Neminsky publica el primer EEG y potenciales evocados de perros. Posteriormente Hans Berger comenzó a realizar sus estudios sobre electroencefalografía en humanos en 1920.

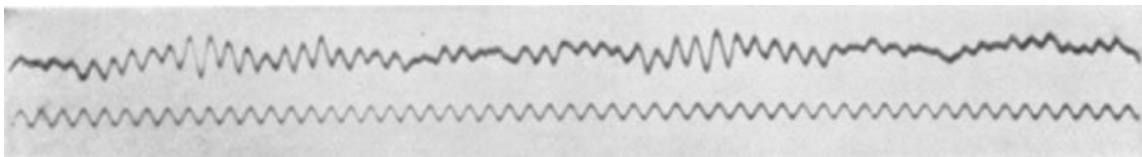


Figura 7.2.2. Primer registro de un electroencefalograma en un humano (Wikipedia, s.f.).

En la figura 7.2.2 se observa el primer electroencefalograma de un humano registrado, perteneciente al año 1929.

### 7.2.3. Ondas Encefálicas

Las ondas cerebrales o también llamadas ondas encefálicas son, según Wikipedia (2022) en su sitio web “Ondas cerebrales”, registros de “actividad eléctrica que generan las diferentes

estructuras del encéfalo. Estas oscilaciones son de muy baja amplitud, del orden de microvoltios en humanos y no siempre siguen una senoide regular. El análisis de los patrones de ondas, es una gran herramienta de investigación en la Neurofisiología del estado de funcionamiento del encéfalo”.

Estas oscilaciones se originan en distintas estructuras tales como el cerebro, cerebelo y núcleos basales. Lo que permite un EEG es recopilar la suma de potenciales de las dendritas postsinápticas. Éstas “forman un agrupamiento de unidades (grupo neuronal o región cortical) con orientación similar y proximidad inmediata, en las capas superficiales de la corteza” (Barrett, Barman, Boitano, & Brooks, 2013).

Según Hermann (1997), las ondas de la actividad eléctrica de fondo del encéfalo que se detectan se pueden clasificar según su frecuencia de oscilación de la siguiente forma:

Tabla 7.2.1. Clasificación de las ondas encefálicas (Hermann, 1997).

Ondas gamma	25 a 100 Hz
Ondas beta	14 a 30 Hz
Ondas alfa	8 a 13 Hz
Ondas theta	3 a 7 Hz
Ondas delta	0 a 4 Hz

Por último, para finalizar esta breve sección teórica cabe destacar que existen múltiples métodos de activación encefálica entre los cuales menciona Wikipedia (s.f.):

- Hiperpnea
- Estimulación luminosa intermitente
- Estimulación visual
- Estimulación auditiva
- Estimulación somestésica
- Estimulación nociceptiva

Estos métodos nos ayudarán posteriormente a la estimulación de los sujetos experimentales y tener el conocimiento de estos elementos los cuales serán utilizados como el pilar fundamental de la experimentación de la presente investigación.

Con esto dicho, es momento de pasar al uso de las ciencias de la computación que complementarán altamente la investigación actual, brindando la parte de automatización y predicción al proyecto para la solución del problema de habla imaginada.

### 7.3. Fundamentos Computacionales, Inteligencia Artificial y Aprendizaje Automático

*Toda tecnología lo suficientemente avanzada es indistinguible de la magia*  
Arthur C. Clarke

Hemos dejado atrás los conceptos de neurociencia – aunque dejar atrás no significa olvidar – y es momento de revisar lo que, para el autor, son los temas, tanto de mayor interés como los que más le atañen en función de obtener el grado profesional como propósito de esta investigación. Se trata de nada más y nada menos que los conceptos matemáticos y computacionales que rodean a esta investigación.

Nos hemos acercado a revisar el cerebro y su estructura, también su estudio a una gran escala y a una pequeña escala y fue aquí donde nos adentramos en nuestro componente más importante: la neurona. Revisamos la morfología de una neurona y las funciones que realiza así como la manera en la que una de estas células interactúa con otra.

Después de todo esto avanzamos de capítulo para entender cómo podía estudiarse esta actividad cerebral y entender lo que era un electroencefalograma, así como la forma en la que se reciben los datos de la actividad cerebral por medio de este aparato; clasificamos las diferentes ondas cerebrales para entender por separado lo que cada tipo de onda representaba. Pero ahora que hablaremos de conceptos de ciencias computacionales, ¿todo esto qué tiene que ver con la computación?

Bien, pues a lo largo del capítulo se observará la importancia de la computadora para poder tratar los datos recibidos por un electroencefalograma, siendo este aspecto la última pieza del presente rompecabezas al cual le he nombrado tesina de grado licenciatura para poder llegar al problema de investigación, el cual tratará justamente de lo que es posible realizar con los datos de un electroencefalograma al manipularlos, entenderlos y aplicar técnicas matemáticas adaptadas a las máquinas para que estos procedimientos los realicen de forma automática, pero para ello, primero será necesario arrancar desde el principio definiendo lo que una computadora es.

#### 7.3.1. Introducción a la Computación

Dado que la presente investigación requiere avanzar a tópicos matemáticos y computacionales considerados algo avanzados, estas primeras definiciones no serán muy discutidas, pero son necesarias de mencionar.

##### 7.3.1.1. Computación

Primeramente, hay que definir lo que es una computadora. Una computadora es “un dispositivo electrónico, utilizado para procesar información y obtener resultados; capaz de ejecutar cálculos y tomar decisiones. En el sentido más simple una computadora es un dispositivo para realizar cálculos o computar” (Joyanes Aguilar, 2008).

Indiscutiblemente, para los efectos de esta investigación, la definición de Joyanes Aguilar es completa y concisa para abarcar los términos necesarios de lo que es una computadora, pues se trata de una máquina a la que se le ingresan datos y ésta responde procesándolos y regresando un resultado después de su procesamiento. Joyanes Aguilar (2008) también hace la clarificación de dividir una computadora en dos partes distintas: “hardware y software. El hardware es la computadora en sí misma. El software es el conjunto de programas que indican a la computadora las tareas que debe realizar” donde puede interpretarse esta mención al hardware como la parte tangible de la computadora mientras que el software, siendo los programas del equipo, se vuelve la parte intangible de la computadora. Así también se conoce con el término programador a la persona que escribe programas.

Sin embargo, es justo gracias al software donde en su mayoría puede enfocarse el procesamiento de datos, o más bien, las instrucciones para este procesamiento, pues el procesamiento de los datos que la computadora recibe se realiza gracias a “un conjunto de instrucciones denominadas programas de computadora. Estos programas controlan y dirigen a la computadora para que realice un conjunto de acciones (instrucciones) especificadas por personas especializadas” (Joyanes Aguilar, 2008) a los cuales este autor denomina programas de computadora.

Son justamente estos programas de computadora los que reciben datos, realizan el procesamiento y producen un resultado con los datos recibidos. “Los datos y la información se pueden introducir en la computadora por una entrada (input) y a continuación se procesan para producir una salida (output, resultados)” (Joyanes Aguilar, 2008).

### **7.3.1.2. Programación**

“La computadora se puede considerar como una unidad en la que se colocan ciertos datos (entrada de datos), se procesan y se produce un resultado (datos de salida o información)” (Joyanes Aguilar, 2008). Este autor menciona que los datos, tanto de entrada como de salida pueden ser de cualquier tipo como texto, dibujos, sonido, imágenes, entre muchos otros tipos. Aunque también menciona que el formato más sencillo de comunicación entre una persona con una computadora es mediante un teclado, una pantalla o monitor y un ratón o mouse.

Dado que en esta investigación lo que se producirá es una solución de software, se centrará específicamente en este aspecto de las ciencias computacionales. Para que el hardware realice las operaciones necesarias se especifican por un conjunto de instrucciones llamados programas. “El software de una computadora es un conjunto de instrucciones de programa detalladas que controlan y coordinan los componentes hardware de una computadora y controlan las operaciones de un sistema informático. El auge de las computadoras el siglo

pasado y en el actual siglo XXI, se debe esencialmente al desarrollo de sucesivas generaciones de software potentes y cada vez más amistosas” (Joyanes Aguilar, 2008).

Un programa tiene que escribirse en un lenguaje de programación que será la principal herramienta del programador para indicarle a la computadora qué deberá realizar al escribir el código fuente de un programa. Sin embargo, “la computadora no entiende directamente los lenguajes de programación, sino que se requiere un programa que traduzca el código fuente a otro lenguaje que sí entiende la máquina directamente, pero muy complejo para las personas; este lenguaje se conoce como lenguaje máquina y el código correspondiente código máquina. Los programas que traducen el código fuente escrito en un lenguaje de programación a código máquina se denominan traductores” (Joyanes Aguilar, 2008).

### **7.3.1.3. El algoritmo**

Antes de escribir el código fuente de un programa debe de analizarse el problema y abstraer las partes más importantes que componen a la problemática. Esto es diseñar un algoritmo. Joyanes Aguilar (2008) define a un algoritmo como un método para resolver un problema y él también propone que los pasos para la resolución de un problema son los siguientes:

- 1)Diseño del algoritmo. Describe la secuencia ordenada de pasos —sin ambigüedades— que conducen a la solución de un problema dado. (Análisis del problema y desarrollo del algoritmo).
- 2)Expresar el algoritmo como un programa en un lenguaje de programación adecuado. (Fase de codificación).
- 3)Ejecución y validación del programa por la computadora.

Un algoritmo es independiente del lenguaje de programación y también lo es de la computadora que lo ejecuta y, de hecho, el algoritmo posee una alta importancia en este campo de estudio pues “en la ciencia de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras. Un lenguaje de programación es tan sólo un medio para expresar un algoritmo y una computadora es sólo un procesador para ejecutarlo. Tanto el lenguaje de programación como la computadora son los medios para obtener un fin: conseguir que el algoritmo se ejecute y se efectúe el proceso correspondiente” (Joyanes Aguilar, 2008).

Todo algoritmo debe cumplir con unas características muy específicas para clasificarse como tal. Estas características necesarias dentro de todo algoritmo, según Joyanes Aguilar (2008) se describen a continuación:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.



- Un algoritmo debe estar bien definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos.

### 7.3.2. Inteligencia Artificial

Sin embargo, el desarrollo de software ha evolucionado a lo largo de los años y esto ha traído notables avances en este campo, entre ellos la aparición de la inteligencia artificial o abreviada, IA. Algunos investigadores en el área encuentran que el trabajo en este campo comenzó poco después de la Segunda Guerra Mundial y se le dio su nombre a esta área en 1956. Aun así, la IA ha crecido mucho en los pocos años que ha tenido de desarrollo, con campos que se han creado tales como aprendizaje y percepción, demostración de teoremas matemáticos, diagnóstico de enfermedades, entre muchas otras más. El fundamento de la IA es que “sintetiza y automatiza tareas intelectuales y es, por lo tanto, potencialmente relevante para cualquier ámbito de la actividad intelectual humana. En este sentido, es un campo genuinamente universal” (Russell & Norvig, 2004).

No obstante, la inteligencia artificial no tiene una definición sencilla, pues distintos autores dan definiciones muy dispersas. Russell (2004) realiza un compendio de algunas definiciones en un cuadro que remarca cuatro enfoques que combinan la racionalidad y la forma de actuar de los humanos con el pensamiento y el actuar, lo que resulta en el cuadro presentado en la tabla 7.3.1.

Tabla 7.3.1. Cuadro de definiciones de la inteligencia artificial presentado por Stuart Russell. (Russell & Norvig, 2004).

Sistemas que piensan como humanos	Sistemas que piensan racionalmente
«El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985) «[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...» (Bellman, 1978)	«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985) «El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)
Sistemas que actúan como humanos	Sistemas que actúan racionalmente
«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990) «El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)	«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole <i>et al.</i> , 1998) «IA... está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)

Afortunadamente, en 1950, Alan Turing apareció en esta ecuación para definir a la inteligencia artificial desde un punto de vista operacional. “En vez de proporcionar una lista larga y quizá controvertida de cualidades necesarias para obtener inteligencia artificialmente, él sugirió una prueba basada en la incapacidad de diferenciar entre entidades inteligentes indiscutibles y seres humanos” (Russell & Norvig, 2004). De este modo, una computadora debería poseer las siguientes capacidades para superar la acuñada Prueba de Turing:

- Procesamiento de lenguaje natural que le permita comunicarse satisfactoriamente en algún idioma.
- Representación del conocimiento para almacenar lo que se conoce o siente.
- Razonamiento automático para utilizar la información almacenada para responder a preguntas y extraer nuevas conclusiones.
- Aprendizaje automático para adaptarse a nuevas circunstancias y para detectar y extrapolar patrones.

Los fundamentos de la inteligencia artificial se han apoyado en muchas otras ciencias, como la filosofía, las matemáticas, la economía, la neurociencia, la psicología, la ingeniería computacional, la teoría del control y cibernética y la lengua para entender todos estos aspectos que rodean al campo de la IA pero esto no ha detenido el avance de los modelos de inteligencia artificial. Entre ellos, Stuart Russell en su obra del 2004 señala una basta cantidad de modelos que se estudian comúnmente y que él expuso en su libro, entre ellos se encuentran modelos de aprendizaje estadístico, como los modelos de Bayes Simples (o también conocidos como Naive Bayes), modelos de aprendizaje con variables ocultas, aprendizaje no supervisado, redes bayesianas con variables ocultas, modelos de Markov, aprendizaje basado en instancias, modelos de vecinos más cercanos, redes neuronales, máquinas núcleo y muchos otros.

### **7.3.3. Aprendizaje Automático**

Es tan amplio el campo de la inteligencia artificial que dentro de éste se encuentran ramas notables; la más notable y útil para el caso de la investigación actual es la del aprendizaje automático. Este campo de estudio tiene como objetivo, según Russell y Norvig (2004), “desarrollar técnicas que permitan que las computadoras aprendan. Se dice que un agente aprende cuando su desempeño mejora con la experiencia y mediante el uso de datos; es decir, cuando la habilidad no estaba presente en su genotipo o rasgos de nacimiento”. Estos mismos autores, en ediciones más recientes de la obra mencionan cómo ocurre el aprendizaje automático, explicando que la computadora observa datos después construye un modelo basado en estos datos y lo utiliza como hipótesis acerca del mundo, siendo ésta, una pieza de software que puede resolver problemas.

Así como la mera inteligencia artificial es tan diversa, el campo del aprendizaje automático también mantiene esta tendencia, pues se ha mantenido en constante desarrollo; tan es así que esta rama tiende a clasificar la investigación y aplicación generada en grupos de modelos de aprendizaje, así como algoritmos de aprendizaje. Flach (2012) en su obra menciona algunos modelos destacables por la forma en la que éstos abordan los problemas.

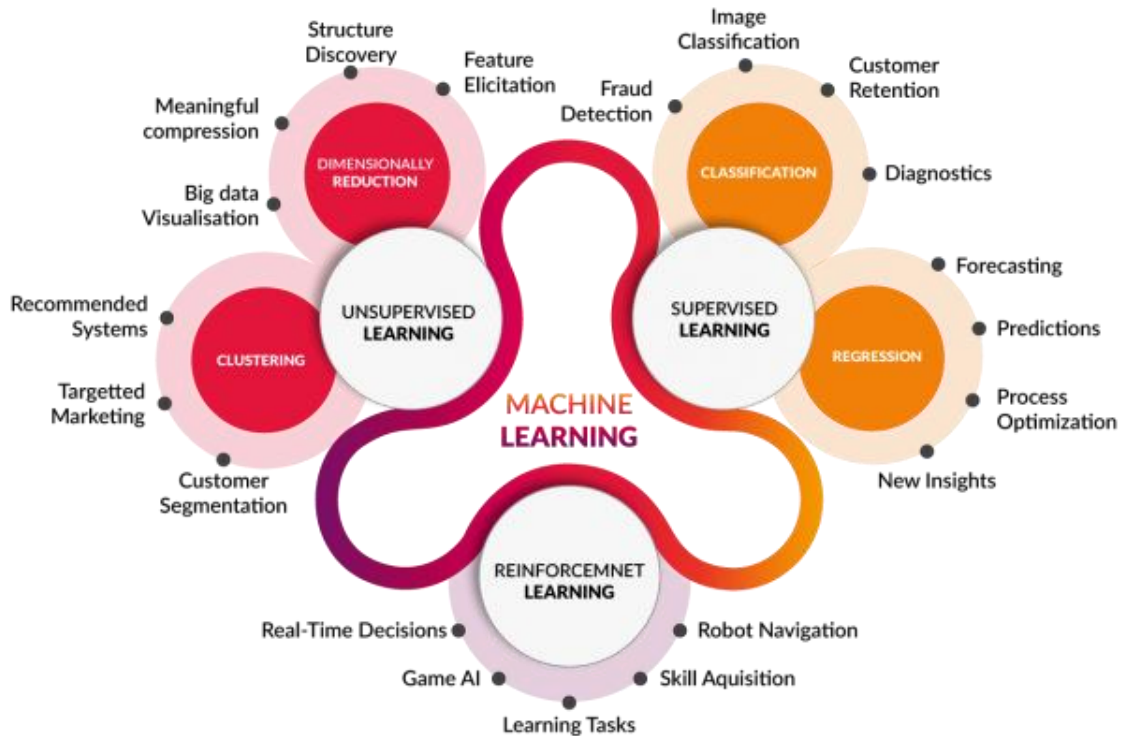


Figura 7.3.1. Clasificación del machine learning (o aprendizaje automático) en sus tres grandes tipos de aprendizaje (González Barrio, Calleja Ochoa, Gómez-Escudero, Rodríguez Ezquerro, & López de Lacalle Marcaide, 2021).

Sin embargo, la riqueza de este campo recae en los distintos tipos de aprendizaje que se han desarrollado bajo el estudio de esta área siendo tres principales:

- 1) **Aprendizaje supervisado:** En este tipo de aprendizaje “el operador proporciona al algoritmo de aprendizaje automático un conjunto de datos conocidos que incluye las entradas y salidas deseadas, y el algoritmo debe encontrar un método para determinar cómo llegar a esas entradas y salidas” (APD, 2019). Esto quiere decir que el algoritmo busca cómo establecer una correspondencia entre las entradas y las salidas deseadas del sistema, donde los ejemplos comunes de estos sistemas son el uso de sistemas con bases de conocimiento las cuales están conformadas por ejemplos anteriormente etiquetados.
- 2) **Aprendizaje no supervisado:** Estos algoritmos están diseñados con el propósito de que el sistema estudie los datos en búsqueda de patrones debido a que “no hay una clave de

respuesta o un operador humano para proporcionar instrucción. En cambio, la máquina determina las correlaciones y las relaciones mediante el análisis de los datos disponibles” (APD, 2019). Entre más datos evalúe el sistema, la toma de decisiones del algoritmo mejorará consiguiendo una mejor segmentación de datos en distintos grupos.

- 3) **Aprendizaje por refuerzo:** El aprendizaje por refuerzo “se centra en los procesos de aprendizajes reglamentados, en los que se proporcionan algoritmos de aprendizaje automáticos con un conjunto de acciones, parámetros y valores finales. Al definir las reglas, el algoritmo de aprendizaje automático intenta explorar diferentes opciones y posibilidades, monitorizando y evaluando cada resultado para determinar cuál es el óptimo” (APD, 2019). Este tipo de sistemas aprenden a través del ensayo y error observando el mundo que les rodea y evaluando la retroalimentación que obtiene al interactuar con el exterior.

Estos tres grandes grupos pueden observarse separados visualmente con mayor claridad en la figura 7.3.1, sin embargo, para efectos de esta investigación se hará énfasis en el aprendizaje supervisado, pues posteriormente se observará con claridad el problema que se trata de resolver y las características del mismo, de modo que pueden percibirse rasgos de clasificación y predicción dentro del problema a resolver; es por esto que se eligió este campo para la solución del problema.

#### **7.3.4. Aprendizaje Supervisado**

El aprendizaje supervisado es un método de análisis de datos muy utilizado por la comunidad científica y que trasciende igualmente al ámbito analítico práctico; dentro de esta rama del machine learning pueden encontrarse algoritmos que “que aprenden iterativamente de los datos para permitir que los ordenadores encuentren información escondida sin tener que programar de manera explícita dónde buscar” (TIBCO Data Science, 2022).

Otros autores definen esta aproximación como “una técnica de aprendizaje automático en la que se proporciona al sistema un conjunto de datos etiquetados, con el objetivo de que el sistema aprenda a generalizar y hacer predicciones sobre nuevos datos” (Alpaydin, 2010).

Asimismo, este concepto tiene atribuciones como englobar “una clase de algoritmos de aprendizaje automático en los que se proporciona al sistema un conjunto de datos con las respuestas correctas, con el objetivo de que el sistema aprenda a hacer predicciones precisas sobre nuevos datos” (Bishop, 2006).

Para todos estos autores, este enfoque para la resolución de problemas converge en una similitud, siendo ésta la utilización de un conjunto de datos etiquetados – como se mencionó en la sección previa – de modo que se entrene a un algoritmo para una o varias tareas específicas. TIBCO Data Science (2022) usan como ejemplo en su artículo la clasificación y predicción de

imágenes en donde se perciban vehículos de dos y cuatro ruedas, donde ellos mencionan que para realizar este procedimiento deberá de entrenarse a un algoritmo con un conjunto de datos que esté correctamente etiquetado, de modo que el algoritmo debe verificar si un vehículo es de dos o de cuatro ruedas.

El aprendizaje supervisado permite “que los algoritmos 'aprendan' de datos históricos/de entrenamiento y los apliquen a entradas desconocidas para obtener la salida correcta. Para funcionar, el aprendizaje supervisado utiliza árboles de decisión, bosques aleatorios y Gradient Boosting Machine” (TIBCO Data Science, 2022).

Un tipo de uso muy común que se le da a nivel práctico al aprendizaje supervisado es al estudio del comportamiento en distintas compañías, creando sistemas de negocio capaces de hacer predicciones variadas, generalmente buscando optimizar las ganancias de este negocio por medio de un proceso de obtención y procesamiento de datos con estos algoritmos. Este proceso se puede ver reflejado en la figura 7.3.2.

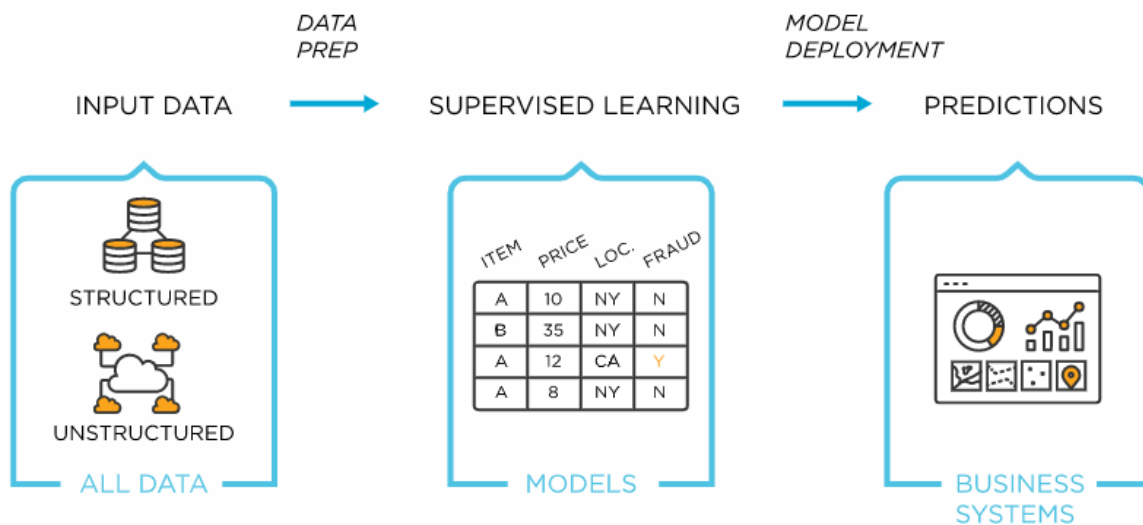


Figura 7.3.2. Procedimiento general del aprendizaje supervisado (TIBCO Data Science, 2022).

Es importante aclarar que, aunque los autores previos no mencionan una característica importante del aprendizaje supervisado, ésta se encuentra intrínseca en las dos subespecializaciones que existen en este campo, tal como se menciona en la definición siguiente, acotando este proceso como uno en el que “un algoritmo de aprendizaje automático se entrena con un conjunto de datos de entrenamiento etiquetados para poder realizar tareas de clasificación o regresión” (Murphy, 2012).

Murphy en su obra otorga clarificaciones de estos dos tipos de tareas que se encuentran como subcampos de este tipo de Aprendizaje: la clasificación y la regresión; entre lo más destacable que él menciona al respecto es la comparación entre ambas, denotando que “la clasificación se

refiere a la tarea de asignar una etiqueta o categoría a una entrada, mientras que la regresión se refiere a la tarea de predecir un valor numérico continuo” (Murphy, 2012).

Aunque como se mencionó anteriormente, los demás autores no se rezagan en hacer notar esta importante distinción, pues Alpaydin (2010) también aporta su propio enunciado, donde define que “la clasificación se refiere a la tarea de asignar una etiqueta o categoría a una entrada, mientras que la regresión se refiere a la tarea de predecir un valor numérico continuo”. Por otra parte, Bishop (2006) menciona que “la clasificación es el proceso de asignar una etiqueta a una instancia, mientras que la regresión es el proceso de predecir un valor numérico continuo”.

Queda claro, entonces, que al tener en cuenta a los autores previos, la diferencia entre clasificación y regresión radica en el tipo de resultado que se desea, lo que causará que se desee inclinarse más por un tipo de tarea u otro.

A través de las definiciones anteriores, puede construirse un concepto más apropiado para los efectos de esta investigación, pues puede observarse que el uso de clasificación recae en situaciones donde se requiere como resultado una clase de entre un número limitado de éstas, mientras que la regresión se utiliza en problemas donde el resultado será un número dentro de un conjunto infinito de posibles resultados.

Martínez Heras (2020) expone en su artículo algunos ejemplos de clasificación, entre los cuales se encuentran:

- 1) ¿Comprará el cliente este producto? [sí, no]
- 2) ¿Tipo de tumor? [maligno, benigno]
- 3) ¿Subirá el índice bursátil? IBEX mañana [sí, no]
- 4) ¿Es este comportamiento una anomalía? [sí, no]
- 5) ¿Nos devolverá este cliente un crédito? [sí, no]
- 6) ¿Qué deporte estás haciendo? Tal y como lo detectan los relojes inteligentes [caminar, correr, bicicleta, nadar]

Este mismo autor también ofrece algunos ejemplos en donde se observa el uso de regresión:

- 1) Predecir por cuánto se va a vender una propiedad inmobiliaria
- 2) Predecir cuánto tiempo va a permanecer un empleado en una empresa
- 3) Estimar cuánto tiempo va a tardar un vehículo en llegar a su destino
- 4) Estimar cuántos productos se van a vender

Finalmente, se ha visto el conocimiento suficiente para poder realizar un acercamiento a las herramientas propias de esta investigación: las técnicas de machine learning. Martínez Heras (2020) menciona como ejemplos de algunas técnicas de aprendizaje automático para la realización de clasificación y regresión son las siguientes:

- Máquinas de vectores de soporte (support vector machines).
- Árboles de decisión (decision trees).
- Bosques aleatorios (random forests).
- Redes neuronales y aprendizaje profundo (neural networks and deep learning).

Cabe aclarar que el autor menciona que, pese a que hay técnicas sumamente especializadas en clasificación y otras en regresión, la mayoría de las técnicas funcionan para ambos enfoques.

En la figura 7.3.3 se observa la separación entre los tres tipos de aprendizaje del machine learning incluyendo algunos ejemplos de técnicas que estos tipos de aprendizaje utilizan para abordar los problemas que se les presentan.

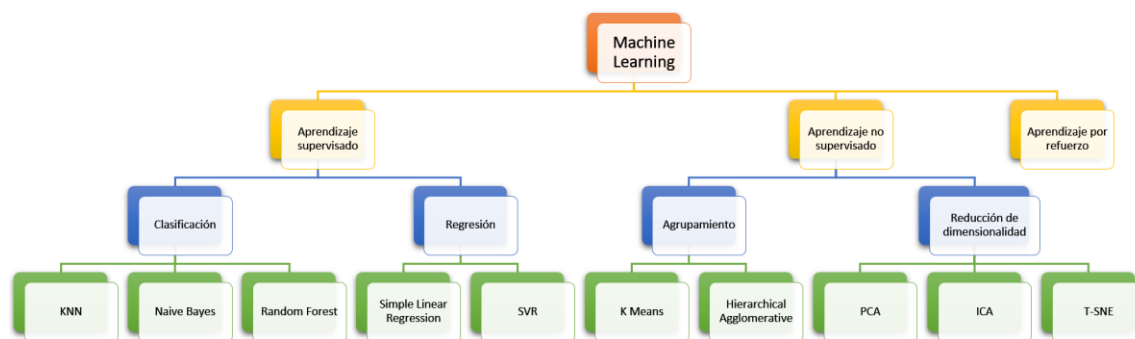


Figura 7.3.3. Principales algoritmos de machine learning (Betanzos Gómez, 2020).

Se le pide al lector dar otro salto de fe en este punto, confiando en que, por el momento se menciona que el problema de la presente investigación se trata de un problema de clasificación. Al finalizar este capítulo todos los cabos sueltos coinciden en un mismo rumbo y en el capítulo siguiente todo se habrá esclarecido – aunque con los tópicos ya revisados de electroencefalografía y aprendizaje supervisado, ya puede ser clara la idea del rumbo de esta investigación – pues todo el conocimiento de base necesario estará puesto sobre la mesa.

Dado que el problema de la presente investigación se trata de un problema de clasificación, se tomó la decisión de abordar esta pregunta de investigación con tres distintas técnicas de machine learning para clasificación: la técnica de máquinas de vectores de soporte, la técnica de bosques aleatorios y la técnica de Bayes ingenuo. A continuación, se aclara cada técnica y se explica por qué se eligieron estas tres técnicas especialmente.

#### 7.3.4.1. La técnica Support Vector Machine

Esta técnica será la primera en uso en esta investigación. La Máquina de Vectores de Soporte, o en inglés “Support Vector Machine” (abreviada de ahora en adelante como SVM) es “un algoritmo de aprendizaje supervisado que se utiliza en muchos problemas de clasificación y

regresión, incluidas aplicaciones médicas de procesamiento de señales, procesamiento del lenguaje natural y reconocimiento de imágenes y voz” (The MathWorks Inc., s.f.).

El equipo de MathWorks (s.f.) menciona que este algoritmo trata de “encontrar un hiperplano que separe de la mejor forma posible dos clases diferentes de puntos de datos. ‘De la mejor forma posible’ implica el hiperplano con el margen más amplio entre las dos clases, representado por los signos más y menos en la siguiente figura”.

El margen que menciona el reconocido equipo dueño de MATLAB es aquel que se define como la anchura máxima de la región paralela al hiperplano que no tiene puntos de datos interiores. El algoritmo solo puede encontrar este hiperplano en problemas que permiten separación lineal; en la mayoría de los problemas prácticos, el algoritmo maximiza el margen flexible permitiendo un pequeño número de clasificaciones erróneas.

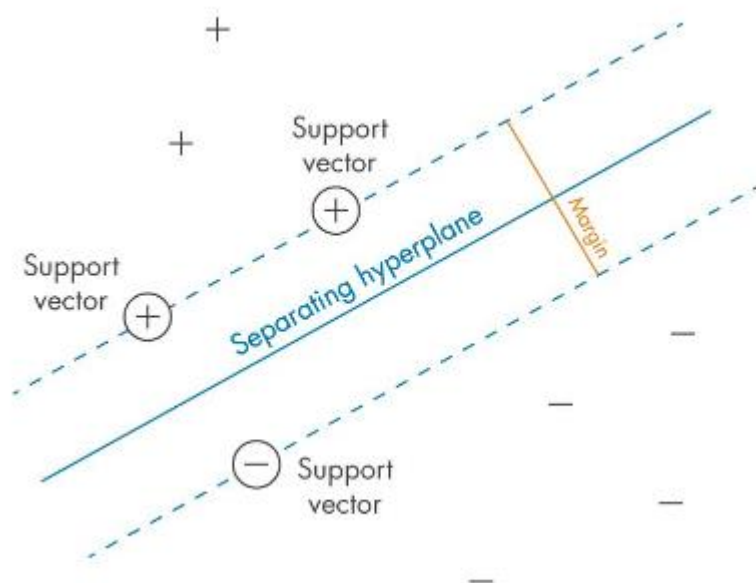


Figura 7.3.4. Definición del “margen” entre clases: el criterio que los SVM intentan optimizar (The MathWorks Inc., s.f.).

Estos vectores de soporte hacen referencia a un subconjunto de observaciones de entrenamiento, las cuales identifican posiciones del hiperplano de separación. Puede observarse una ejemplificación en la figura 7.3.4.

Un algoritmo estándar SVM está formulado para problemas de clasificación binaria, en cuanto a los problemas multiclase normalmente se reducen a una serie de problemas binarios. Existen autores que realizan esta reducción usando una técnica denominada “One vs All” (o “One vs Res”) en la cual se entrena una SVM para cada clase del problema – como lo menciona Band (2020) – comparando cada clase con las demás. Luego, se utiliza la clase con la mayor probabilidad de pertenencia para clasificar una nueva instancia. Otros autores – como



destaca Brownlee (2021) – reducen el problema mediante una técnica “*One vs One*” en la cual se entrena una SVM para cada par de clases, y se elige la clase con la mayor cantidad de victorias.

Cabe destacar que las SVM pertenecen a una clase de algoritmos del machine learning que son denominados “Métodos Kernel”. Estos algoritmos pueden utilizar una función de kernel para transformar las características.

Como lo menciona el equipo de MathWorks (s.f.) “las funciones de kernel asignan los datos a un espacio dimensional diferente, que suele ser superior, con la expectativa de que resulte más fácil separar las clases después de esta transformación, simplificando potencialmente los límites de decisión complejos no lineales para hacerlos lineales en el espacio dimensional de características superior asignado”. Los datos no se tienen que transformar explícitamente durante este procedimiento, lo que supondría una alta carga computacional. A esto se le conoce como “*truco de kernel*”.

Existen algunos kernels muy conocidos con los que se pueden trabajar, el equipo desarrollador de MATLAB destaca dentro de las paqueterías de su software kernels como la función de base radial o gaussiana (véase la ecuación 7.3.1), la función lineal (véase la ecuación 7.3.2), la función polinómica (véase la ecuación 7.3.3) o la función sigmoide (véase la ecuación 7.3.4).

$$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

Ecuación 7.3.1. Función de base radial o gaussiana  
(The MathWorks Inc., s.f.).

$$K(x_1, x_2) = x_1^T x_2$$

Ecuación 7.3.2. Función lineal (The  
MathWorks Inc., s.f.).

$$K(x_1, x_2) = (x_1^T x_2 + 1)^\rho$$

Ecuación 7.3.3. Función polinómica (The MathWorks  
Inc., s.f.).

$$K(x_1, x_2) = \tanh(\beta_0 x_1^T x_2 + \beta_1)$$

Ecuación 7.3.4. Función sigmoide (The MathWorks  
Inc., s.f.).

Un entendimiento visual que es más claro de cómo se realiza la separación del hiperplano puede observarse en la figura 7.3.5.

El entrenamiento de una SVM se muy similar a resolver un problema de optimización cuadrática para ajustar un hiperplano que minimice el margen flexible entre las clases. El número de características está determinado por el número de vectores de soporte.

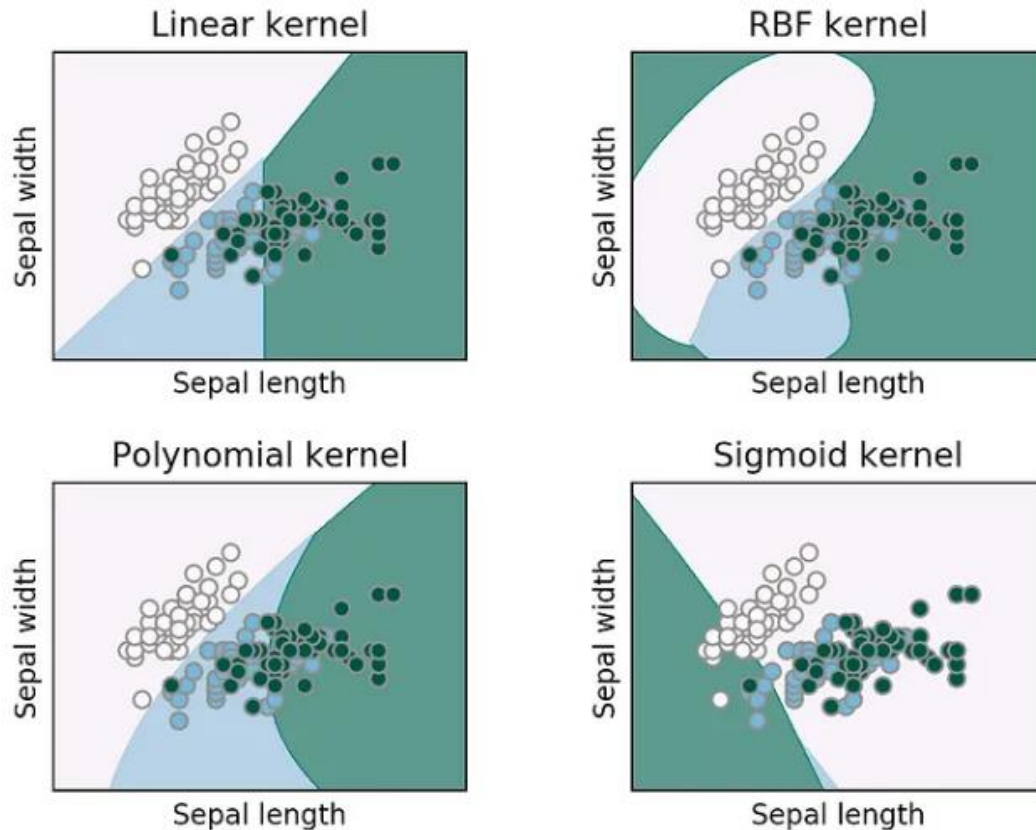


Figura 7.3.5. Gráficas de separación del hiperplano de los distintos kernels (Marius, 2020).

El uso de las SVM trae consigo ciertas ventajas, como las menciona Blog Unipython (2018):

- Es eficaz en espacios de grandes dimensiones.
- Mantiene eficacia en casos donde el número de dimensiones es mayor que el número de muestras.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamada vectores de soporte), por lo que también es eficiente en memoria.
- Es versátil dado que se pueden especificar distintas funciones del núcleo para la función de decisión. Se proporcionan kernels comunes pero también es posible especificar kernels personalizados.

Algunos puntos clave de las SVM son los siguientes según MathWorks (s.f.):

- Son muy populares y logran un buen rendimiento en muchas tareas de clasificación y regresión.
- Aunque los algoritmos SVM están formulados para la clasificación binaria, los algoritmos SVM multiclase se construyen combinando varios clasificadores binarios.

- Los kernels hacen que los SVM sean más flexibles y capaces de gestionar problemas no lineales.
- Para construir la superficie de decisión, sólo se requieren los vectores de soporte seleccionados a partir de los datos de entrenamiento. Una vez terminado el entrenamiento, el resto de datos de entrenamiento es irrelevante, produciendo una representación compacta del modelo que es adecuada para generar código de forma automatizada.

Hay que tomar en cuenta dos factores importantes “si el número de características es mucho mayor que el número de muestras evite el exceso de ajuste al elegir las funciones del Kernel y el término de regularización es crucial” (Blog Unipython, 2018). Estos mismos autores también mencionan que los SVMs no proporcionan directamente estimaciones de probabilidad, éstas se calculan utilizando una validación cruzada quíntuple.

#### 7.3.4.2. La técnica Random Forest

De la siguiente técnica que se hablará será la técnica de Bosques Aleatorios, o en inglés “Random Forest”. Esta técnica es muy popular en el machine learning y el análisis de datos.

La técnica se sustenta en los llamados “Árboles de Decisión”; éstos “ayudan al científico de datos a tomar una decisión gracias a una serie de preguntas (también llamadas tests) cuya respuesta (sí o no) llevará a la decisión final” (Data Scientist, 2022). Pueden encontrarse ejemplos de estos algoritmos basándose en temas de Economía gracias a las facilidades que en este campo han encontrado al utilizar este algoritmo. Un ejemplo es el que brinda Sanabria Castro (2020) el cual se ilustra a continuación en la figura 7.3.6.

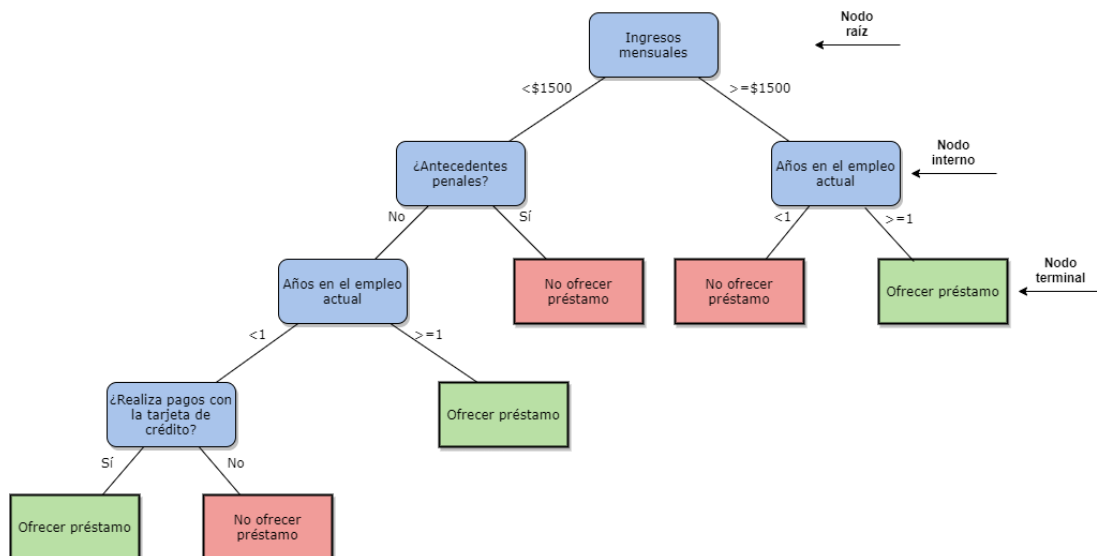


Figura 7.3.6. Árbol de Decisión aplicado en Economía (Sanabria Castro, 2020).

Como breve contextualización de los Árboles de Decisión, Sanabria Castro menciona que este tipo de modelos están formados por “una serie de decisiones lógicas similares a las de un diagrama de flujo. El árbol comienza con un nodo llamado ‘nodo raíz’, en este y en los nodos internos se toman las decisiones basado en diferentes atributos, las ramas indican las decisiones tomadas. Al final del árbol de decisión se encuentran los nodos terminales, que representan el resultado de seguir una combinación de decisiones. Alternativamente, el nodo terminal puede tener asociada una probabilidad de que el valor meta tome cierto valor” (Sanabria Castro, 2020).

Gracias a la construcción del árbol, estos modelos pueden utilizarse para la predicción de variables categóricas y continuas, recibiendo el nombre de árboles clasificación y regresión respectivamente.

La figura 7.3.6 es parte de un ejemplo que Sanabria Castro utiliza para visualizar la idea de un Árbol de Decisión mediante un ejemplo de un banco el cual utiliza un árbol de decisión para decidir si le debería ofrecer un préstamo a una persona.

Sanabria Castro (2020) menciona que “para construir un árbol de decisión es un método heurístico llamado partición recursiva. La idea es que conforme se avance en el árbol de decisión se formen grupos cada vez más uniformes, en donde predomine una clase o un valor. En el ejemplo anterior estos grupos son; personas a las que fue acertado, ofrecerles un préstamo y las que no”. Esta es la forma en la que el algoritmo encuentra patrones que son utilizados para las predicciones que se realizarán con aplicantes nuevos.

La construcción de uno se realiza mediante métodos como el ID3, el C4.5, C5.0 y el CART, sin embargo la explicación en el artículo de Sanabria Castro posee a profundidad los detalles minuciosos de estos algoritmos. La cuestión importante para efectos de esta investigación es conocer que el método está cimentado en encontrar los criterios de separación (o “Splitting Criteria”) necesarios con los que se construye el árbol.

Esta técnica presenta muchas ventajas en comparación con otros algoritmos de datos pues es fácil de interpretar, estable y generalmente regresa buenas coincidencias muy útiles en tareas de regresión o clasificación.

Enlazando la base de los Árboles de Decisión se obtiene el método de Random Forest, pues este método es considerado como un método de conjunto (o Ensemble Method) pues combina resultados con el objetivo de obtener un superresultado final que fue generado de los resultados de los diferentes Árboles de Decisión que componen al algoritmo de Bosque Aleatorio.

Un modelo de Bosque Aleatorio puede “constar de varias decenas, incluso centenas de árboles, el número de árboles es un parámetro que por lo general, se ajusta mediante validación cruzada (o cross-validation, en inglés). Para abreviar, la validación cruzada es una técnica de evaluación de un algoritmo de machine learning que consiste en entrenar y probar el modelo en fragmentos de la serie de datos de partida” (Data Scientistest, 2022).

El sustento del algoritmo está en entrenar a cada árbol en un subconjunto de la serie de datos y así obtener un resultado. “Posteriormente, se combinan los resultados de todos los árboles de decisión para dar una respuesta final. Cada árbol “vota” (sí o no) y la respuesta final es la que tenga la mayoría de votos” (Data Scientistest, 2022).

Esto anteriormente descrito es a lo que en el artículo de Data Scientistest (2022) le denominan como “bagging” y otras fuentes como el artículo en El Estadístico Blogspot (2021) coinciden en el uso de este método, el cual es un procedimiento constituido por los siguientes pasos:

- 1) Se divide la serie de datos en varios subconjuntos compuestos aleatoriamente de muestras, de ahí el “random” de Random Forest.
- 2) Se entrena un modelo en cada subconjunto: habrá tantos modelos como subconjuntos.
- 3) Se combinan todos los resultados de los modelos (con un sistema de voto, por ejemplo) lo que proporciona un resultado final.

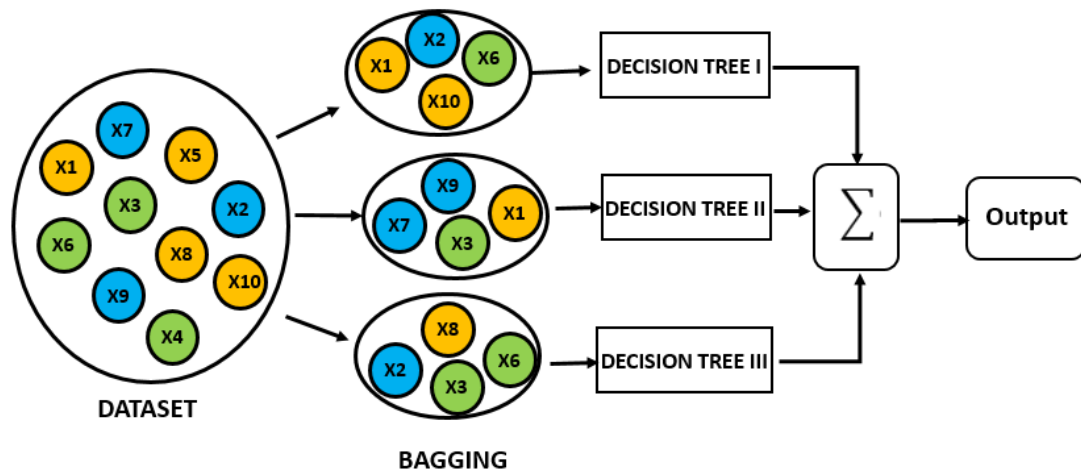


Figura 7.3.7. Visualización del “bagging” como separación, usado en Random Forest (Orellana Alvear, 2018).

La descripción visual del “bagging” puede observarse en la figura 7.3.7. Existe otra técnica llamada “boosting” la cual consiste en una separación y evaluación secuencial, mientras que el “bagging” puede verse como una separación y evaluación en paralelo; esta ejemplificación se observa en la figura 7.3.8 pero no se le profundizará mucho más en este método, pues no es de relevancia para el presente estudio.

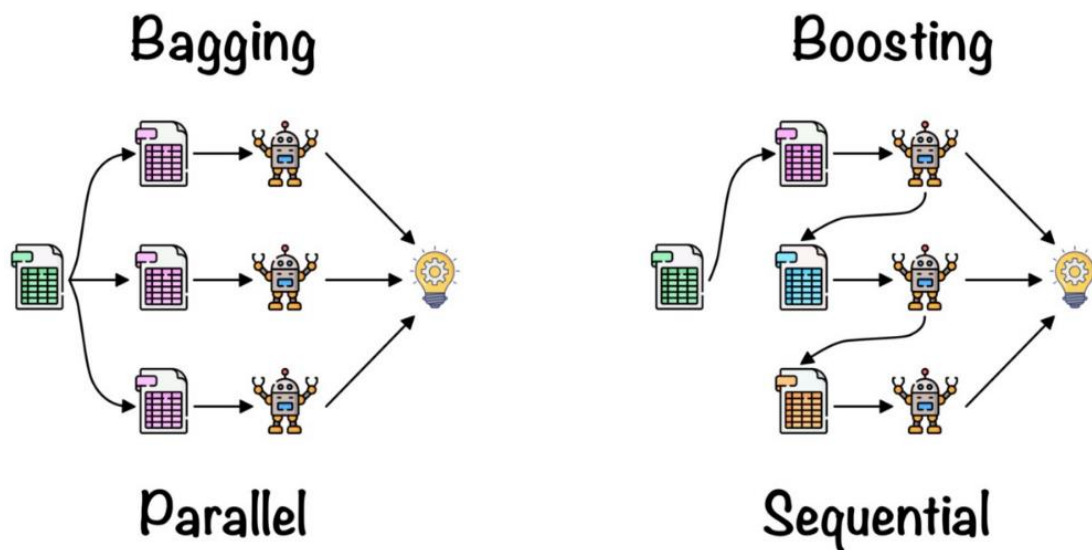


Figura 7.3.8. Comparación visual entre el “bagging” y el “boosting” (Sruthi, 2022).

Matemáticamente, el funcionamiento del Random Forest se puede dividir en cuatro pasos según El Estadístico Blogspot (2021):

- I. Seleccionamos  $k$  *features* o características (columnas o variables) de las  $m$  totales (siendo  $k < m$ ) y se crea un árbol de decisión con esas  $k$  características.
- II. Se crean  $n$  árboles variando siempre la cantidad de  $k$  *features* o características. También podría variarse la cantidad de muestras que se pasan a esos árboles (esto es conocido como “*Bootstrap Sample*”).
- III. Se toma cada uno de los  $n$  árboles y se ejecutan realizando una misma clasificación. Se guarda el resultado de cada árbol obteniendo  $n$  salidas.
- IV. Se calculan los votos obtenidos para cada clase seleccionada y se considerará a la más votada como la clasificación final de nuestro bosque.

#### 7.3.4.3. La técnica Naive Bayes

Como última técnica que se abordará bajo la lupa de esta investigación se tiene a la técnica de Bayes Ingenuo, o en inglés “*Naive Bayes*”. Estos modelos son una clase especial de algoritmos de clasificación dentro del aprendizaje automático y están basados en una técnica de clasificación estadística llamada el teorema de Bayes.

Para brindar superficialmente el conocimiento del teorema de Bayes al lector se resalta que éste mismo es el núcleo de la Inferencia Bayesiana; un campo muy amplio dentro de la teoría de la probabilidad. Se trata de una proposición planteada por el matemático inglés Thomas Bayes la cual fue publicada en 1763 tal como se resalta en el ensayo que el mismo Bayes

(1763) publicó. En términos simples, en este ensayo se encuentra este teorema el cual expresa la probabilidad condicional de un evento aleatorio  $A$  dado  $B$  en términos de la distribución de probabilidad condicional del evento  $B$  dado  $A$  y la distribución de probabilidad marginal de solo  $A$ .

El teorema interpretado por Parzen (1987) está enunciado de la siguiente forma:

Sea  $\{A_1, A_2, \dots, A_i, \dots, A_n\}$  un conjunto de sucesos mutuamente excluyentes y exhaustivos tales que la probabilidad de cada uno de ellos es distinta de cero: ( $P[A_i] \neq 0$  para  $i = 1, 2, \dots, n$ ). Si  $B$  es un suceso cualquiera del que se conocen las probabilidades condicionales  $P(B|A_i)$ , entonces la probabilidad  $P(A_i|B)$  viene dada por la expresión:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \quad \text{Ecuación 7.3.5. El Teorema de Bayes (Parzen, 1987).}$$

Donde:

- $P(A_i)$  son las probabilidades a priori.
- $P(B|A_i)$  es la probabilidad de  $B$  en la hipótesis  $A_i$ .
- $P(A_i|B)$  son las probabilidades resultantes.

El teorema de Bayes puede visualizarse al superponer dos árboles de decisión como se observa en la figura 7.3.9.

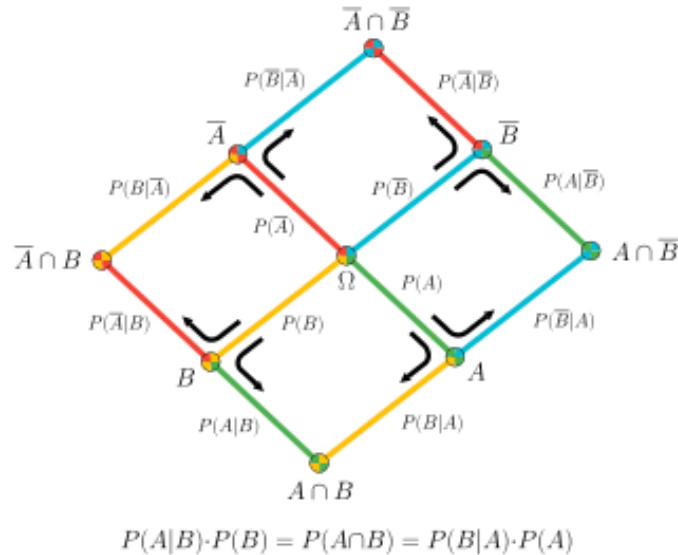


Figura 7.3.9. Visualización del teorema de Bayes por superposición de dos árboles de decisión (Parrás & Tedesco).

Ahora, con base en la definición de probabilidad condicionada se obtiene la fórmula de Bayes. También se le llama Regla de Bayes como Walpole, Myers, Myers y Ye (2012) la denominan.

Ellos la definen matemáticamente en la obra que ellos escribieron titulada “*Probabilidad y Estadística para Ingeniería y Ciencias*” de la siguiente manera:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{k=1}^n P(B|A_k)P(A_k)}$$

Ecuación 7.3.6. La Regla de Bayes (Walpole, Myers, Myers, & Ye, 2012).

La teoría estadística previa es la base de esta técnica, además de la característica de estos modelos de ser “Naive” o ingenuos en español. Se les llama así debido a que este tipo de esquemas asume que las variables predictoras son independientes entre sí.

Debido a la simplicidad de esta técnica, proporcionan una manera muy sencilla de elaborar modelos con un comportamiento óptimo. Esto lo consiguen, según Roman (2019) “proporcionando una forma de calcular la probabilidad ‘posterior’ de que ocurra un cierto evento A, dadas algunas probabilidades de eventos ‘anteriores’”. Matemáticamente se expresa como se observa en la figura 7.3.10.

$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

{

P(A): Probabilidad de A

P(R|A): Probabilidad de que se de R dado A

P(R): Probabilidad de R

P(A|R): Probabilidad posterior de que se de A dado R

Figura 7.3.10. El uso del Teorema de Bayes en la técnica de Naive Bayes (Roman, 2019).

Para poder utilizar esta técnica en problemas de clasificación, Roman (2019) menciona los siguientes pasos necesarios para el trabajo con este algoritmo:

- 1) Convertir el conjunto de datos en una tabla de frecuencias.
- 2) Crear una tabla de probabilidad calculando las correspondientes a que ocurran los diversos eventos.
- 3) La ecuación Naive Bayes se usa para calcular la probabilidad posterior de cada clase.
- 4) La clase con la probabilidad posterior más alta será el resultado de la predicción.

Esta técnica tiene ciertas ventajas y desventajas de su uso. Entre las ventajas que destaca Roman (2019) se encuentran las siguientes:

- I. Un manera fácil y rápida de predecir clases, para problemas de clasificación binarios y multiclase.
- II. En los casos en que sea apropiada una presunción de independencia, el algoritmo se comporta mejor que otros modelos de clasificación, incluso con menos datos de entrenamiento.



- III. El desacoplamiento de las distribuciones de características condicionales de clase significa que cada distribución puede ser estimada independientemente como si tuviera una sola dimensión. Esto ayuda con problemas derivados de la dimensionalidad y mejora el rendimiento.

Por otra parte, Roman (2019) señala las siguientes desventajas:

- a. Aunque son unos clasificadores bastante buenos, los algoritmos Naive Bayes son conocidos por ser pobres estimadores. Por ello, no se deben tomar muy en serio las probabilidades que se obtienen.
- b. La presunción de independencia Naive muy probablemente no reflejará cómo son los datos en el mundo real.
- c. Cuando el conjunto de datos de prueba tiene una característica que no ha sido observada en el conjunto de entrenamiento, el modelo le asignará una probabilidad de cero y será inútil realizar predicciones. Uno de los principales métodos para evitar esto, es la técnica de suavizado, siendo la estimación de Laplace una de las más populares.

Estos serán los algoritmos que se utilizarán posteriormente. Con esto, queda cubierto el conocimiento requerido dentro del campo de las ciencias de la computación; finalmente se cubrirán tópicos del tema de interés de esta investigación, el cual se tratará de la predicción del habla imaginada.

## 7.4. Habla Imaginada e Intención del Movimiento

*El pensamiento humano puede literalmente, transformar el mundo físico*  
Dan Brown: El Símbolo Perdido

Ha llegado el momento, pues tras todo el marco teórico previo, es el momento de abordar el problema de la presente investigación. Los temas previos funcionaron para poner sobre la mesa todo conocimiento necesario para enfocarse ahora en el habla imaginada.

Hay algunos documentos de investigación que profundizan en un estudio de señales electroencefalográficas con machine learning tratando de obtener información de estos datos a partir de una idea primitiva de lo que es el Habla Imaginada.

El Habla Imaginada es un campo novedoso dentro de disciplinas que tratan tópicos muy relacionados, tal como lo es la Psicología. Posee distintas definiciones entre los autores que se han acercado a este ámbito. En la obra de Sheikh y Korm (1994) se da como definición del Habla Imaginada “la habilidad de crear imágenes mentales y asociarlas con el lenguaje” así como también se destaca que es “la capacidad de una persona de hablar en su mente sin pronunciar en voz alta las palabras” (Sheikh & Korm, 1994).

Por otra parte Singer (2006) en su obra “Imagery in Psychotherapy” define este concepto como “una habilidad cognitiva que consiste en el uso de imágenes mentales para generar palabras o frases”.

Puede referirse al Habla Imaginada como un término que se refiere a la capacidad de una persona de hablar en su mente sin pronunciar en voz alta las palabras. Esta habilidad puede estar relacionada con la capacidad de la mente de crear imágenes mentales y de asociarlas con el lenguaje. Sin embargo, aún no se conoce mucho sobre cómo funciona exactamente la Habla Imaginada y más investigación es necesaria para entenderlo mejor.

Un trato muy adecuado es el que le dan Marín, Martínez, Ureña y López (2017) en su artículo, que es al que se apega este capítulo con mayor medida. Estos autores comienzan en su artículo preguntándose sobre si somos capaces de interpretar las señales eléctricas que genera nuestro cerebro mientras estamos pensando. Ellos desde un inicio dan pie a la posibilidad de esto gracias a los avances del aprendizaje automático e incluso mencionan “nos encontramos en el inicio de una nueva era en lo que al conocimiento del cerebro se refiere” (Marín, Martínez, Ureña, & López, 2017).

Estos autores también dan su definición, clara y concisa respecto al habla imaginada: “palabras que se piensan, pero no se dicen”. Mencionan los avances que se están consiguiendo, tales como distintos resultados al proveer un vocabulario reducido consiguiendo que una persona se comuniqué con un equipo de cómputo a través del pensamiento, únicamente con ayuda del hardware de adquisición de la actividad cerebral. Esto fue una obra experimental que realizó el vinculado al Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE) Torres-García (2013) junto a otros investigadores citados a continuación.

Este artículo incluso es citado en el artículo de Marín, Martínez, Ureña y López, siendo una base importante para los estudios del Habla Imaginada y también para la presente investigación. El artículo plantea su uso de las Interfaces Cerebro Computadora “mecanismos neurológicos o procesos empleados por el usuario para generar las señales de control, se les denomina fuentes electrofisiológicas. Las más utilizadas son: los potenciales corticales lentos (SCP, por sus siglas en inglés), los potenciales P300, las imágenes motoras (ritmos sensoriales motrices  $\mu$  y  $\beta$ ) y los potenciales evocados visuales (VEP, por sus siglas en inglés)” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Dentro de su investigación definen que “la fuente electrofisiológica es el habla imaginada, también referida como habla interna o habla no pronunciada (unspoken speech), donde el término habla imaginada se refiere a la pronunciación interna, o imaginada, de palabras pero sin emitir sonidos ni articular gestos para ello” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013) la cual es la brújula de la definición del Habla Imaginada a lo largo de toda la publicación.

Así también, mencionan citando a otra publicación académica realizada por Denby y otros (2010) mencionando que este autor “incluye a estos trabajos dentro de un área de investigación denominada interfaces de habla silente (SSI, por Silent Speech Interfaces) cuya finalidad es desarrollar sistemas capaces de permitir la comunicación “hablada” que toman lugar cuando la emisión de una señal acústica entendible es imposible” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Torres-García y sus colaboradores remarcan la importancia de separar los trabajos de Habla Imaginada en dos enfoques: palabras y sílabas. En el artículo se encuentran referencias a otras publicaciones de ambos enfoques sirviendo como ejemplos principales de cada uno de los enfoques. Posteriormente también explica brevemente los alcances y limitantes de las publicaciones que menciona como referentes de ambas formas de tratamiento de este problema.

Una vez que termina de dar este repaso al estado del arte que tienen Torres-García y su grupo de autores colaboradores, procede a explicar que su propuesta se centra en “interpretar las señales para reconocer la pronunciación imaginada de palabras de un vocabulario reducido compuesto por las siguientes cinco palabras en lenguaje español: “arriba”, “abajo”, “izquierda”, “derecha” y “seleccionar”” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Para descifrar las palabras a partir de un EEG, como primer paso debe contemplarse “el registro de la actividad eléctrica cerebral como consecuencia de la activación de grupos de neuronas, lo que habitualmente se denomina un EEG. Ello requiere el uso de un hardware de uso específico que registre tal actividad. Están proliferando muchas empresas dedicadas al desarrollo de este tipo de hardware (Emotiv – emotiv.com, Neurosky – neurosky.com, GTEC – gtec.at)” (Marín, Martínez, Ureña, & López, 2017).

Estos autores mencionan que es necesario tomar en cuenta la cantidad y la posición de los electrodos pues, “si lo que pretendemos es analizar el habla imaginada, el hardware necesariamente deberá tener electrodos localizados sobre el área del cerebro relacionada con esa tarea” (Marín, Martínez, Ureña, & López, 2017).

El objetivo de la investigación señalada “retoma lo planteado por los trabajos previos con la idea en mente de que las palabras con mayor significado semántico puedan generar mayor actividad cerebral que permita su reconocimiento y correcta clasificación. Además, con ellas sería posible controlar la dirección del cursor de la computadora. El problema es tratado bajo un enfoque de clasificación, y se conoce a priori en qué parte de la señal de EEG la persona imagina la pronunciación de las palabras indicadas” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Como bien se explica en el artículo, al terminar la adquisición de datos, el siguiente paso es el preprocesamiento de las ondas cerebrales, que “consiste básicamente en la eliminación de ciertos artefactos (ruidos, interferencias...) con el fin de evitar cualquier tipo de injerencia no deseada en la señal de interés. Los artefactos más típicos suelen ser de origen biológico (latidos de corazón, parpadeo...) y están localizados en el mismo cuerpo del sujeto a estudiar, aunque debido a la naturaleza de estas ondas y a su rango de amplitudes, no es raro encontrar artefactos procedentes de otros lugares” (Marín, Martínez, Ureña, & López, 2017).

El método usado es el método de referencia promedio común, o CAR, por sus siglas en inglés, como se explica en el artículo de Torres-García y otros (2013). Ellos señalan que el CAR puede ser calculada mediante la siguiente fórmula:

$$V_i^{CAR} = V_i^{ER} - \frac{1}{n} \sum_{j=1}^n V_j^{ER}$$

Ecuación 7.4.1. Fórmula del cálculo del CAR (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Donde:

- $V_i^{ER}$  es el potencial entre el  $i$ -ésimo electrodo y la referencia.
- $n$  es el número de electrodos en el montaje.

También se destaca que en la etapa de preprocesado también “podemos aplicar diferentes técnicas que facilitan la manipulación de los datos obtenidos mediante el EEG. Los EEG son, en definitiva, el registro de señales cuyo valor varía en cada instante temporal. De este modo es posible, por ejemplo, filtrar la señal original quedándonos solamente con las bandas de frecuencia que tienen una carga informativa mayor o que caen dentro de la frecuencia que deseamos estudiar” (Marín, Martínez, Ureña, & López, 2017); esto porque las BCI son no estacionarias puesto que las señales del EEG pueden variar rápidamente con el tiempo.

Torres-García y otros (2013) proponen como técnica que permite modelar dichas variaciones en el dominio tiempo-escala es la Transformada Wavelet Discreta (DWT por sus siglas en inglés). “La DWT provee una representación wavelet altamente eficiente mediante la restricción de la variación en la traslación y la escala, usualmente a potencias de dos. En ese caso, la DWT es algunas veces llamada transformada wavelet diádica” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

La DWT, según Torres-García y otros (2013), se define mediante la siguiente ecuación:

$$W(j, k) = \sum_j \sum_k f(x) 2^{-\frac{j}{2}} \psi(2^{-j}x - k)$$

Ecuación 7.4.2. Fórmula del cálculo de la DWT (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

La explicación del conjunto de funciones  $\psi_{j,k}(n)$  es compleja y se menciona superficialmente “referido como la familia de wavelets derivadas de  $\psi(n)$ , el cual es una función de tiempo con energía finita y rápido decaimiento llamada la wavelet madre. Las bases del espacio wavelet corresponden entonces, a las funciones ortonormales obtenidas de la wavelet madre después de las operaciones de escala y traslación. La definición indica la proyección de la señal de entrada en el espacio wavelet a través del producto interior” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013). No se profundiza tanto en la DWT debido a que en la presente investigación no se hace uso de este recurso.

Saltando entonces, al siguiente paso de este acercamiento al habla imaginada, se menciona que en la investigación analizada se utilizaron técnicas de aprendizaje automático para discriminar ondas cerebrales dado que “la clasificación cubre cualquier contexto en el que alguna decisión o pronóstico es hecho sobre la base de información histórica disponible” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

La base de información disponible  $D$  se define de la siguiente forma:

$$D = \{\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_m, y_m \rangle\} = \langle X, Y \rangle$$

Ecuación 7.4.3. Definición de la base de información disponible  $D$  (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Donde los valores  $X_i \in X$  son típicamente vectores multi-dimensionales de la forma  $X_i = \{z_1, z_2, \dots, z_n\}$  cuyos elementos pueden tomar valores reales o discretos. Los autores denominan a estos componentes como atributos o características y el propósito revelado en esta sección de la investigación de Torres-García y otros (2013) es inferir una función (o relación)  $f$  tal que  $X \rightarrow Y$  de modo que los valores de  $Y$  estén contenidos en un conjunto finito de clases  $C = \{C_1, \dots, C_k\}$  que caracterizan los datos dados.

Las técnicas de clasificación que se usan en dicha investigación son también Naive Bayes, Random Forest y SVM, aunque no con el objetivo de compararlas entre sí, como es el caso de la presente investigación, sino como refuerzo buscando una misma conclusión tratando de demostrar que los distintos canales por los que se transmite la actividad cerebral son relevantes para proceder con investigaciones de Habla Imaginada.

Se recalca que las conclusiones a las que se llegaron son esperanzadoras, pero muy dependientes en gran medida del tipo de sistema que se esté desarrollando, dependiendo de “los tipos de sensores cerebrales usados (internos o superficiales), las condiciones en las que se capturan los EEG y la parte del cerebro que se pretenda estudiar. Incluso entre individuos, se experimentan variaciones de hasta un 20% en la precisión del sistema a la hora de clasificar los datos” (Marín, Martínez, Ureña, & López, 2017).

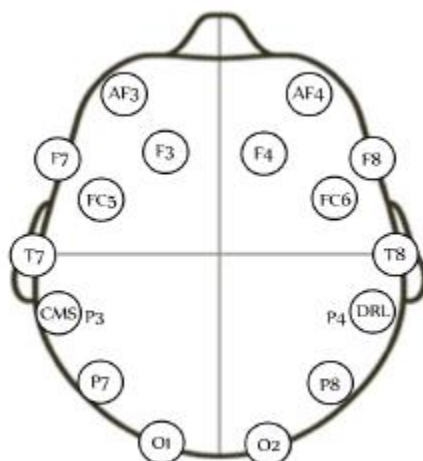


Figura 7.4.1. Ubicación de los nodos en el hardware Emotiv usado en su experimentación (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Ellos mencionan basándose en un diagrama proporcionado dentro de su misma publicación académica – adjunto en este documento como la figura 7.4.1 – los nodos de mayor importancia a los cuales llegaron a las conclusiones mencionadas. Entre los resultados publicados, se destaca que “utilizando habla imaginada registrada únicamente de los canales F7-FC5-T7-P7, en todos los individuos, son superiores al 20%, es decir, están arriba del azar para cinco clases” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Remarcan que esto es curioso por haber usado palabras semánticamente similares, realizando como conjetura que las señales recopiladas podrían contener información para ser utilizada en tareas de clasificación de palabras de un vocabulario reducido. “Actualmente se está realizando un análisis de las características utilizadas por el método con la intención de ampliar nuestro conocimiento sobre

cuáles de ellas son las más apropiadas para la correcta clasificación del habla imaginada” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013).

Claramente también recalcan que “el usar todos los canales no necesariamente implica mayor información para obtener el mejor desempeño posible; por lo que, queda por explorar cuál es la mejor combinación de canales” (Torres-García, Reyes-García, Villaseñor-Pineda, & Ramírez-Cortés, 2013) dentro de todo el espacio de combinaciones que ellos exponen dentro de su artículo para optimizar la interpretación de las señales obtenidas del cerebro.

La experimentación en la presente investigación será similar aunque no igual a la presentada en el artículo de Torres-García y otros (2013) dando por hecho que la información del EEG sí rescata elementos con los que podríamos considerar realizar interpretaciones que lleven a decodificar esta información y usarla como medio para posteriormente desarrollar algoritmos que permitan decodificar estos datos y convertirlos en las interpretaciones del Habla Imaginada.

Con lo anterior previamente expuesto, se han cubierto los tópicos biológicos del cerebro y la forma de acción de las neuronas, se mencionaron los dispositivos que permiten coleccionar esta información, siendo los EEG protagonistas de dicha sección, se destacó el papel de la computación, la inteligencia artificial, el aprendizaje automático y el énfasis en el aprendizaje supervisado, denotando la importancia de las técnicas utilizadas; ahora se habló del problema de investigación: el Habla Imaginada.

El siguiente paso en esta investigación será acomodar todas estas piezas ahora en un esquema práctico y observar: ¿qué técnica de aprendizaje supervisado predice con mayor eficiencia datos provenientes de señales electroencefalográficas interpretables como parte del Habla Imaginada de un individuo que desea operar un equipo de cómputo por este medio? Esta pregunta se responderá a nivel práctico en los próximos capítulos.

## 8. Metodología

La metodología de esta investigación se planteó con base en comparación y contraste de resultados. Primeramente se eligieron los tres modelos de aprendizaje supervisado ya mencionados en capítulos anteriores, posteriormente se dedicó también tiempo a la implementación del enlace entre el dispositivo Emotiv y el equipo de cómputo para simular la Interfaz Cerebro – Computadora y coleccionar los datos mediante código Python desarrollado propiamente y para estos fines.

Se limpió levemente el conjunto de datos, añadiendo la etiqueta del concepto que se le pide al individuo pensar y se separará en distintos perfiles, con el argumento que los distintos individuos tendrán distintas intensidades y peculiaridades en la actividad cerebral, evitando que el modelo se ensucie con una mezcla de todos los conjuntos de datos compuestos como si fuera uno solo.

Después, debido a las carencias de realizar el experimento y la predicción en vivo mientras se transmiten los datos, la predicción se realizó posterior a la recolección de datos, donde se escribió un código Python para evaluar el conjunto de datos, eligiendo una porción del conjunto de datos como conjunto de entrenamiento y el resto del conjunto como conjunto de prueba. Finalmente, la eficiencia fue evaluada utilizando una matriz de confusión y así poder determinar la efectividad de cada modelo propuesto.

Posteriormente se trataron los datos mediante procedimientos de normalización y transformación para mejorar la eficiencia de predicción sobre cuál es el modelo que mejor se adapta a la interpretación de la actividad cerebral como intención de movimiento y a partir de estos resultados, se observaron los comportamientos de las muestras para determinar el modelo que mejor realiza la predicción, dando así por finalizada esta investigación y reportando los resultados en el presente documento. El procedimiento previamente relatado puede observarse a modo de diagrama en la figura 8.1 el cual marca el flujo previamente explicado.

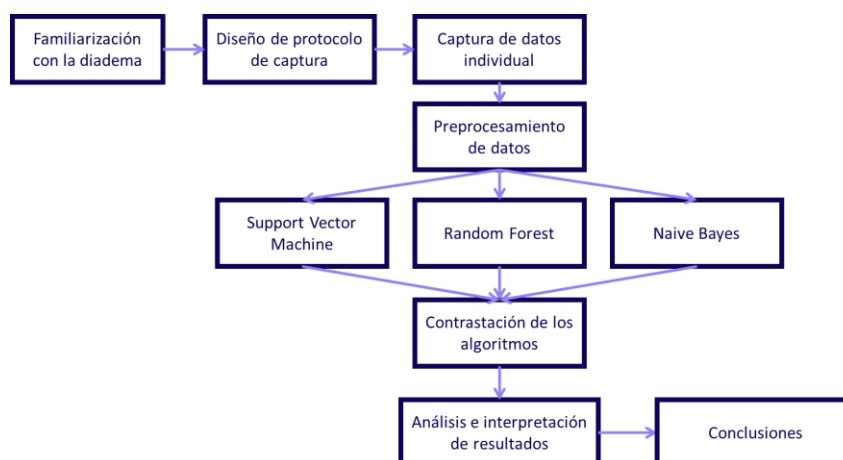


Figura 8.1. Diagrama de pasos a seguir según la metodología planteada.



## 9. Desarrollo

Como parte del desarrollo de esta investigación primero se tuvo el periodo previo a la formalización experimental tratando de adaptarse al hardware y software requerido realizando algunas instalaciones destacables, pues para este proyecto es requerido que la instalación de la diadema y el software se haga de la manera adecuada. Para ello cabe mencionar que la presente investigación se realizó sobre el sistema operativo Windows 11, máquina a la que se le conectaría el modelo de la diadema ya mencionado. También se usará el lenguaje de programación Python en su versión 3.10 y otras instalaciones se cubrirán en el avance de configuración de la diadema con el equipo para recibir y transmitir los datos.

El proceso entero de instalación para tener preparada cada configuración de transmisión y preparación de datos es extenso y no es propio de los objetivos de la investigación, sino sólo es parte de los procedimientos necesarios para poder realizar correctamente la experimentación de esta investigación, por ello el proceso detallado está explicado en el anexo 1.

A grandes rasgos, es necesario tener instalado Python en una versión 3.7.0 o superior y descargar el repositorio de CyKit desde la liga <https://github.com/CymatiCorp/CyKit>. Asimismo debe de seguirse el tutorial detallado por CymatiCorp en la liga <https://github.com/CymatiCorp/CyKit/wiki/How-to-Stream-Data-to-OpenViBE>, en el cual será necesario descargar e instalar OpenViBE en una versión 2.7.2 o superior y crear un escenario como el ilustrado en la figura 9.4.

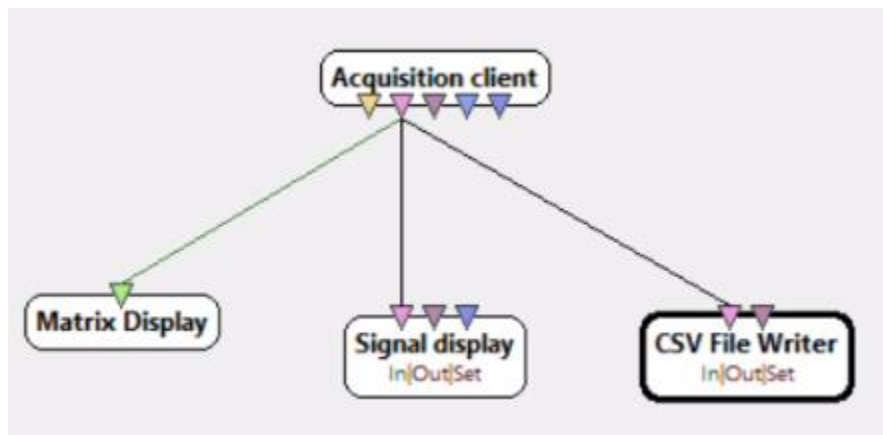


Figura 9.1. Escenario de OpenViBE Designer para experimentación de esta investigación.

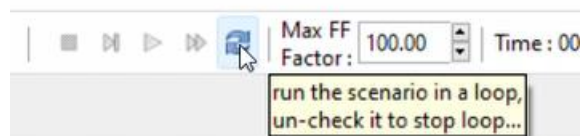


Figura 9.2. Botón para ciclar la ejecución.

Finalmente, se habilita el ciclado del escenario con el botón señalado en la figura 9.5 y usar el siguiente código de Python para que automáticamente este código cicle el procedimiento y simule la transmisión de datos en tiempo real.

```
1 import pyautogui
2 import time
3
4 # Aplicando delay para tener tiempo de acomodar el mouse sobre el botón
5 time.sleep(10)
6
7 x, y = pyautogui.position()
8
9 # Aplicando delay al código para correr el escenario y automáticamente se
  mueva a la posición indicada
10 time.sleep(10)
11 pyautogui.moveTo(x, y)
12
13 # El proceso se mantiene en ejecución mientras el mouse esté en su posición
14 while(pyautogui.position() == (x, y)):
15     pyautogui.click()
16     time.sleep(1) # El CSV se actualiza cada segundo
```

Cabe resaltar que el segmento de código anteriormente expuesto no es parte final de la experimentación de este proyecto, sino que fue usado en una versión primitiva del proyecto antes de abordarlo como formalmente un proyecto de tesina de licenciatura.

Pero gracias a estos inicios, finalmente se formalizó el desarrollo de código de este proyecto como se presenta en la siguiente sección como parte de la experimentación a nivel de código. Esta parte se realizó en una primera instancia preparando el código de Python que próximamente se mencionará para así estar preparado en el momento de realizar la investigación de campo con los usuarios de prueba y posteriormente procesar los datos. Estos pasos se mencionan a profundidad en la siguiente sección.

## 10. Experimentación y Pruebas

*Esas veces en las que estás muy cansado, no quieres exigirte pero igualmente lo haces...  
Ese es el sueño. No es el destino, es el trayecto*  
Kobe Bryant

Todos los conceptos previos revisados han servido únicamente para llegar aquí: la experimentación del documento e investigar de qué manera puede usarse la diadema Emotiv Epoc+ para crear una interfaz cerebro – computadora útil tomando como puntos de comparación distintos modelos de aprendizaje supervisado.

Una vez planteada la estructura anteriormente presentada en el capítulo previo, se procedió a escribir el archivo de Python con el que se hace todo el tratamiento y procedimiento de la actual investigación. Este archivo se encuentra adjunto bajo el nombre `functions.py` así como puede encontrarse en el anexo 2 en el presente documento. A continuación se explican las cuestiones necesarias para el entendimiento del experimento.

La primera función fue una reestructuración al código anteriormente mostrado, formalizándolo en inglés y dándole un formato propio y estándar al resto del código; esto con intenciones de mantener el orden también en el código.

```
1  """ Detect the position of the mouse
2  def MousePosition(secs = 3):
3      # Sleeping the script so we have time to set the mouse on its position
4      time.sleep(secs)
5
6      # Getting the mouse position
7      print(pyautogui.position())
8
9      return
```

El propósito de este código es parcialmente el mismo que el anterior, pues busca identificar la posición del mouse en la pantalla para posteriormente utilizar la librería `pyautogui` que controlará los clicks de la máquina basándose en las coordenadas dadas. Para conocer las coordenadas de estos puntos importantes se ejecuta este código pasando únicamente como parámetro un valor “secs” que será un valor entero, simbolizando los segundos de retraso entre la ejecución del comando y la realización del proceso. Esto con la intención de que el usuario tenga tiempo de establecer el mouse en la posición en la que desea conocer las coordenadas. En una terminal de Python se ejecuta la función sin parámetros y por defecto el usuario dispondrá de tres segundos para posicionar el mouse.

```
MousePosition()
```

En las posteriores funciones se muestran los resultados personales de la ejecución de esta función.

La función que complementa a ésta para la actualización del script sería RunUpdateStream(), la cual fue separado por la forma en la que se planteó (presentada en el anexo 3). Esta función quedó rápidamente depreciada personalmente, pues planteaba una aproximación de evaluación en vivo, sin embargo, se observó que podía ser altamente complicado realizar esto (de modo que en lo sucesivo de este documento, se plantea como propuestas de mejora de la investigación).

```
1  """ Creating the dataset with the input data and the expected output
2  def createDataset(xMaximize, yMaximize, xPlay, yPlay, xWindow, yWindow,
3  xStop, yStop, profileName, expectedResult, rangeTime):
4      # Check if profile path exists
5      if(not(os.path.exists('profiles/' + str(profileName)))):
6          os.makedirs('profiles/' + str(profileName))
7          time.sleep(0.2)
8
9      # Maximizing OpenViBE Designer
10     pyautogui.moveTo(xMaximize, yMaximize)
11     pyautogui.click()
12     time.sleep(1)
13
14     # Playing OpenViBE Designer
15     pyautogui.moveTo(xPlay, yPlay)
16     pyautogui.click()
17     time.sleep(rangeTime)
18
19     # Clicking over OpenViBE Designer
20     pyautogui.moveTo(xWindow, yWindow)
21     pyautogui.click()
22     time.sleep(0.5)
23
24     # Stopping OpenViBE Designer
25     pyautogui.moveTo(xStop, yStop)
26     pyautogui.click()
27     time.sleep(0.5)
28
29     # Loading data.csv
30     dataframe = pandas.read_csv('data.csv')
31
32     # Reworking the dataframe
33     dataframe = dataframe.drop(labels='Time:256Hz', axis=1)
34     dataframe = dataframe.drop(labels='Epoch', axis=1)
35     dataframe = dataframe.drop(labels='Event Id', axis=1)
36     dataframe = dataframe.drop(labels='Event Date', axis=1)
37     dataframe = dataframe.drop(labels='Event Duration', axis=1)
38
39     # Add a column with the expected output
40     dataframe.insert(16, "Expected Output", [expectedResult] *
41     len(dataframe))
42
43     # Append (or create) the dataset of this selected profile
44     if(not(os.path.isfile('profiles/' + str(profileName) +
45     '/dataset.csv'))):
```

```

43     file = open('profiles/' + str(profileName) + '/dataset.csv', "x")
44     file.close()
45     dataframe.to_csv('profiles/' + str(profileName) + '/dataset.csv',
mode='a', index=False, header=True)
46     else:
47         dataframe.to_csv('profiles/' + str(profileName) + '/dataset.csv',
mode='a', index=False, header=False)
48
49     # Remove the data.csv raw file
50     os.remove('data.csv')
51
52     return

```

Así que se optó por un enfoque más estático, donde la transmisión de datos no sería en vivo, sino que, el usuario tendría la orden de mantener concentración en un tema específico mientras la información se registra en un archivo.

Para la realización automática de este proceso, se necesitaban varias posiciones: la posición donde se da click para maximizar OpenViBE, donde se inicia la transmisión de datos con la diadema Emotiv, donde se superpone esta ventana (debido a que, en ocasiones, el electroencefalograma se inicia por encima de los botones importantes) y la posición donde se le da en finalizar a la transmisión (planteamiento presentado en el anexo 4).

Como el algoritmo hace estos clicks automáticamente, también se define el tiempo que sucede entre el inicio y el final de la transmisión con la variable `rangeTime` la cual es definida por defecto para cada instancia de concentración en cinco segundos.

Como se mencionó anteriormente, al usuario se le pedirá que se concentre en cierto aspecto o tópico. Este se señalará en el algoritmo dentro de la variable `expectedResult` el cual se incluirá en el conjunto de datos que se genera como producto de la ejecución de esta función.

Los conjuntos de datos están separados en perfiles, para mantener limpia la información y realizar el entrenamiento con los conjuntos independientes por cada usuario; esto para evitar que el experimento falle por la distinta actividad cerebral entre los usuarios. El perfil se define en `profileName`. Este nombre será el que se le asigne en sistema a una carpeta donde se guardará el conjunto de datos y su respectiva información aislada.

A los usuarios voluntarios se les colocó la diadema, como se muestra en la figura 10.1 con algunos ejemplos de entre todos los voluntarios dando por iniciado el procedimiento de experimentación de campo. Todos los voluntarios están mencionados en el anexo 7.



Figura 10.1. Ejemplo de uso de la diadema en los voluntarios 1, 8, 7, 9 y 23 de izquierda a derecha, de arriba abajo.

El proceso de experimentación de campo se estandarizó mediante un protocolo el cual puede encontrarse a mayor profundidad en el anexo 6 donde se detallan los pasos que se siguieron previo a la experimentación, durante este periodo y después del mismo.

Al finalizar este procedimiento se obtenían dos archivos separados por comas por individuo como el que se aprecia en la figura 10.2; uno en el que sería un primer acercamiento a algo similar a habla imaginada pero no definido como tal, funcionando tanto como calentamiento para el segundo archivo como una cuestión de comparación dentro de los resultados finales. Por otro lado, el segundo archivo será el realmente importante para efectos de la investigación, pues es en éste en el que están los datos de las señales captadas tras pedirle a los usuarios concentrarse en las distintas intenciones de movimiento señaladas.

Para el primer archivo se les pidió a los usuarios pensar y visualizar en los siguientes conceptos:

- Árbol
- Computadora
- Cuaderno
- Perro

Como este procedimiento no fue como tal habla imaginada o intención de movimiento, el proceso del primer archivo se deja como control comparativo frente al segundo archivo, en el que se les pidió



a los usuarios concentrarse en intentar mover el ratón o *mouse* de una computadora en las cuatro direcciones básicas: hacia arriba, hacia abajo, hacia la izquierda y hacia la derecha.

En ambas evaluaciones se pidió una toma de control tratando de no realizar concentración en un pensamiento en específico.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11	Channel 12	Channel 13	Channel 14	Expected Output	
2	4216.02588	4927.30762	4159.87158	4161.6665	4158.71777	4156.6665	4159.87158	4161.15381	4156.53857	4160.38477	4167.17969	4168.3335	4165.38477	4167.30762	MouseUp	
3	4154.4873	4157.94873	4158.58984	4156.41016	4163.20508	4168.46143	4167.56396	4163.07715	4173.07715	4216.92285	4928.84619	4160.25635	4161.6665	4156.6665	MouseUp	
4	4164.87158	4166.79492	4163.58984	4172.69238	4215.5127	4927.69238	4159.4873	4161.53857	4157.43604	4167.82031	4156.41016	4160.76904	4158.58984	4160.25635	MouseUp	
5	4924.74365	4160.76904	4163.07715	4156.28223	4167.43604	4159.4873	4160.5127	4158.20508	4162.43604	4168.07715	4166.92285	4165.5127	4168.97412	4215.64111	MouseUp	
6	4156.79492	4160.5127	4156.02588	4160.38477	4166.79492	4166.15381	4168.20508	4170	4214.61523	4923.97412	4156.6665	4162.05127	4156.53857	4156.53857	MouseUp	
7	4167.69238	4167.69238	4167.43604	4215.5127	4925.12842	4157.56396	4159.61523	4159.10254	4157.69238	4155.25635	4157.94873	4155.76904	4158.71777	4157.69238	MouseUp	
8	4158.97412	4161.02588	4158.20508	4167.05127	4153.58984	4156.6665	4153.58984	4158.58984	4158.71777	4167.56396	4171.15381	4168.58984	4218.58984	4921.6665	MouseUp	
9	4158.97412	4155.5127	4157.43604	4159.61523	4166.6665	4173.46143	4163.97412	4217.69238	4927.82031	4156.15381	4158.84619	4161.02588	4168.20508	4159.10254	MouseUp	
10	4174.87158	4164.61523	4215.12842	4929.61523	4156.15381	4160.38477	4166.53857	4166.92285	4155.25635	4159.87158	4159.4873	4160.89746	4167.56396	4167.94873	MouseUp	
11	4158.71777	4155	4161.28223	4155.5127	4158.3335	4159.10254	4160	4165.38477	4168.20508	4172.05127	4171.6665	4216.15381	4923.07715	4155.76904	MouseUp	
12	4160.5127	4159.23096	4169.35889	4166.92285	4170.25635	4170.38477	4214.10254	4925.38477	4159.87158	4160.76904	4156.79492	4164.87158	4155.76904	4161.53857	MouseUp	
13	4166.15381	4216.28223	4927.17969	4160.25635	4160.64111	4156.02588	4167.05127	4156.79492	4159.4873	4160.89746	4159.4873	4156.6665	4168.46143	4165.25635	MouseUp	
14	4157.56396	4158.97412	4158.3335	4158.58984	4161.6665	4158.97412	4160.89746	4168.20508	4165	4158.97412	4217.94873	4922.94873	4159.4873	4160.76904	MouseUp	
15	4156.15381	4163.71777	4167.56396	4164.10254	4160.12842	4215.76904	4922.05127	4157.43604	4162.30762	4159.87158	4161.53857	4158.58984	4160.38477	4158.46143	MouseUp	
16	4215.89746	4921.79492	4155.64111	4158.58984	4159.23096	4168.58984	4158.84619	4156.79492	4160.76904	4160.89746	4161.41016	4167.30762	4167.05127	4157.56396	MouseUp	
17	4168.20508	4156.56396	4156.53857	4161.41016	4160.5127	4164.74365	4166.02588	4166.92285	4154.23096	4215.76904	4926.41016	4161.28223	4160.25635	4158.07715	MouseUp	
18	4156.56396	4166.28223	4170.25635	4158.46143	4215	4923.84619	4158.3335	4162.43604	4158.84619	4165.38477	4160.12842	4158.71777	4156.92285	4158.84619	MouseUp	
19	4923.46143	4157.17969	4161.28223	4157.69238	4166.15381	4158.46143	4156.53857	4157.56396	4161.28223	4167.30762	4169.74365	4171.53857	4161.41016	4217.30762	MouseUp	
20	4157.56396	4157.94873	4155.76904	4159.61523	4168.97412	4166.41016	4173.3335	4157.56396	418.07715	4927.69238	4161.28223	4159.74365	4159.74365	4166.15381	MouseUp	
21	4166.41016	4177.05127	4159.23096	4214.10254	4925.25635	4159.61523	4159.23096	4158.07715	4163.20508	4158.97412	4159.35889	4157.56396	4157.56396	4154.74365	MouseUp	
22	4158.20508	4160.5127	4156.28223	4162.30762	4155.64111	4159.74365	4160.64111	4158.46143	4161.41016	4167.69238	4176.28223	4162.94873	4216.92285	4923.3335	MouseUp	
23	4160.12842	4156.79492	4159.10254	4162.17969	4168.97412	4179.10254	4166.53857	4217.05127	4926.15381	4157.05127	4158.07715	4161.53857	4157.17969	4158.3335	MouseUp	
24	4177.05127	4162.17969	4216.41016	4925	4155.38477	4156.41016	4159.61523	4162.43604	4157.69238	4160	4157.43604	4158.3335	4160.12842	4166.53857	MouseUp	
25	4157.82031	4159.23096	4161.6665	4157.82031	4158.46143	4157.30762	4158.20508	4160.76904	4167.94873	4170.76904	4161.02588	4215.5127	4924.61523	4158.46143	MouseUp	
26	4154.58984	4160.5127	4165	4168.84619	4171.02588	4162.43604	4216.28223	4923.46143	4162.17969	4160.76904	4162.82031	4153.46143	4157.56396	4161.15381	MouseUp	
27	4158.58984	4218.58984	4925.12842	4162.05127	4163.46143	4160	4153.58984	4159.23096	4161.15381	4160	4157.94873	4159.4873	4167.43604	4172.30762	MouseUp	
28	4158.84619	4148.97412	4159.87158	4160.25635	4160.25635	4160.38477	4162.30762	4168.07715	4168.3335	4161.15381	4177.30762	4923.97412	4155.76904	4162.56396	MouseUp	
29	4161.41016	4168.58984	4167.69238	4169.35889	4163.46143	4218.07715	4922.69238	4157.56396	4161.15381	4157.94873	4149.61523	4157.69238	4159.4873	4158.20508	MouseUp	
30	4216.41016	4928.07715	4156.02588	4160.25635	4162.30762	4155.64111	4156.53857	4159.87158	4155.25635	4160.5127	4166.02588	4167.82031	4179.10254	4156.41016	MouseUp	
31	4158.71777	4155.12842	4160.64111	4157.17969	4162.05127	4168.07715	4167.82031	4174.74365	4158.58984	4216.6665	4924.87158	4161.02588	4163.97412	4157.94873	MouseUp	
32	4163.71777	4167.69238	4174.61523	4162.43604	4215.76904	4923.07715	4159.74365	4159.87158	4157.43604	4152.30762	4156.6665	4158.46143	4159.4873	4160.25635	MouseUp	
33	4925.5127	4157.69238	4159.87158	4158.20508	4153.58984	4159.23096	4158.3335	4157.94873	4159.35889	4162.17969	4168.20508	4170.12842	4163.84619	4176.79492	MouseUp	
34	4159.4873	4158.07715	4158.84619	4159.35889	4158.84619	4167.17969	4168.46143	4162.69238	4216.6665	4925.25635	4163.58984	4157.17969	4163.46143	4151.28223	MouseUp	

Figura 10.2. Fragmento del segundo archivo obtenido de la experimentación de campo del voluntario 14.

Dado que los valores en primeras evaluaciones resultaron bajos, además se realizó un proceso de normalización eligiendo la Normalización Z, buscando un mejor desempeño de los algoritmos tras este tratamiento de los datos. Un ejemplo puede visualizarse en la figura 10.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11	Channel 12	Channel 13	Channel 14	Expected Output
2	-0.31377529	-0.36784798	-0.28292767	-0.30589747	-0.30724003	-0.32002973	-0.31849711	-0.31571988	-0.3115948	-0.30061607	-0.28783556	-0.27020859	-0.28319672	-0.25520886	MouseUp
3	-0.30480198	-0.31621783	-0.32575541	-0.30978692	-0.29074441	-0.28238865	-0.27896412	-0.29303934	-0.25125121	-0.00667662	3.59994926	-0.28967581	-0.31041155	-0.29479331	MouseUp
4	-0.28533003	-0.28186417	-0.30563781	-0.25535444	0.00133421	3.60754965	-0.28933363	-0.31312853	-0.29472588	-0.318135	-0.32477527	-0.32277032	-0.33438616	-0.30906813	MouseUp
5	3.59993401	-0.29612306	-0.29460774	-0.29812352	-0.32643385	-0.3187321	-0.31914629	-0.32608778	-0.31354217	-0.27985234	-0.27487235	-0.29292075	-0.25792651	-0.01576462	MouseUp
6	-0.31648466	-0.32140198	-0.32964794	-0.3020105	-0.29074416	-0.27589804	-0.30034985	-0.25675046	-0.01181925	3.59714974	-0.28977806	-0.30135564	-0.30587574	-0.32074797	MouseUp
7	-0.27105175	-0.29353099	-0.25502254	-0.00652869	3.595544	-0.28758163	-0.31525618	-0.29239026	-0.31786995	-0.32072985	-0.3105286	-0.31563292	-0.30133994	-0.29219587	MouseUp
8	-0.30415432	-0.29806836	-0.29785275	-0.31820993	-0.32124388	-0.317432	-0.2044217	-0.28332051	-0.27331228	-0.27011795	-0.30533293	-0.2617737	-0.02207206	3.6141885	MouseUp
9	-0.31908273	-0.3285339	-0.29590525	-0.2903471	-0.27322431	-0.3187321	-0.25757592	-0.0137527	3.59783384	-0.28309713	-0.29885132	-0.30589807	-0.30069091	-0.32334542	MouseUp
10	-0.29311931	-0.26112262	0.0054139	3.57878657	-0.27906426	0.30510331	-0.28479681	-0.29368841	-0.32327652	-0.32462459	-0.31440618	-0.31044051	-0.29097274	-0.2662412	MouseUp
11	-0.29571491	-0.29418024	-0.30888283	-0.31820993	-0.31994639	-0.32262499	-0.28738857	-0.28202237	-0.28434405	-0.27531012	-0.24635675	-0.0935232	3.59351231	-0.28635594	MouseUp
12	-0.32622817	-0.28640154	-0.28033018	-0.2845154	-0.29853159	-0.2492905	-0.01713207	3.59753842	-0.28888626	-0.30775313	-0.32218115	-0.32212034	-0.32920133	-0.32399292	MouseUp
13	-0.24963186	-0.09962947	3.60601345	-0.29228935	-0.29334162	-0.28628153	-0.33405261	-0.3209026	-0.32003174	-0.31359278	-0.30144298	-0.2961657	-0.27153393	-0.30842062	MouseUp
14	-0.28273443	-0.32010594	-0.31212784	-0.31820993	-0.3303288	-0.29212209	-0.28220258	-0.264527	-0.30251287	-0.25454639	-0.00721865	3.58556091	-0.28967714	-0.30517567	MouseUp
15	-0.28857577	-0.2915857	-0.28800259	-0.28905144	-0.26024692	-0.00657487	3.58435197	-0.29109458	-0.30835249	-0.29737127	-0.3293119	-0.33380265	-0.31559391	-0.32788785	MouseUp
16	-0.01337529	3.58645964	-0.29720276	-0.30136143	-0.29398913	-0.32327504	-0.31590535	-0.3105347	-0.33171098	-0.29282905	-0.30014715	-0.28059094	-0.31041155	-0.26299625	MouseUp
17	-0.3184326	-0.29871761	-0.31861786	-0.33181806	-0.30242654	-0.29342219	-0.28155588	-0.3001668	-0.26228298	-0.02030328	3.60707755	-0.29097329	-0.30069091	-0.29933575	MouseUp
18	-0.30025845	-0.27732681	-0.29395775	-0.2560035	-0.00717416	3.60560445	-0.28868445	-0.28785662	-0.30186292	-0.31164541	-0.31634869	-0.31368546	-0.32444154	-0.29998325	MouseUp
19	3.5785166	-0.29871761	-0.28162769	-0.29941818	-0.30437154	-0.3122445	-0.30748742	-0.31831214	-0.29407935	-0.2804982	-0.2752125	-0.2896781	-0.2656302	-0.00729272	MouseUp
20	-0.31129346	-0.32374727	-0.32112189	-0.2877012	-0.28360672	-0.27654809	-0.30099092	-0.2638804	-0.01506404	3.6016196	-0.2904721	-0.302656	-0.30004435	-0.32788785	MouseUp
21	-0.26910381	-0.30066043	-0.29495246	-0.0181299	3.60008646	-0.29147451	-0.30486666	-0.30146248	-0.31631701	-0.3178495	-0.32283029	-0.33964257	-0.30004435	-0.28960809	MouseUp
22	-0.30008058	-0.30156547	-0.29350255	-0.32015466	-0.31779892	-0.32376317	-0.33721098	-0.29174638	-0.27500443	-0.28244719	-0.28977806	-0.2661631	-0.00384718	-0.30645612	MouseUp
23	-0.31908273	-0.32784864	-0.29395775	-0.28062595	-0.2842567	-0.29082446	-0.25239239	-0.01051227	3.59588895	-0.28309713	-0.29885132	-0.29746319	-0.30267133	-0.318153	MouseUp
24	-0.2739969	-0.3260237	-0.32102872	-0.35910904	-0.29398913	-0.310188	-0.29192539	-0.3150781	-0.31743443	-0.30480308	-0.3234794	-0.29162327	-0.2961157	-0.27597358	MouseUp
25	-0.2905327	-0.3077898	-0.31018034	-0.31108258	-0.33166269	-0.33820146	-0.29775809	-0.29239026	-0.36675274	-0.29607385	-0.25413418	-0.00610737	3.57925834	-0.2908838	MouseUp
26	-0.34005015	-0.30260572	-0.28552669	-0.26507558	-0.28230935	-0.25448348	-0.0048175	3.5839301	-0.28564147	-0.30450834	-0.29314369	-0.7048554	-0.322709	-0.31945049	MouseUp
27	-0.26391261	-0.0070734	3.59433339	-0.28127502	-0.30696652	-0.30570589	-0.30877677	-0.32155167	-0.32587136	-0.33630388	-0.29366801	-0.2802803	-0.2522221	-0.28246349	MouseUp
28	-0.31259126	-0.31232725	-0.32136039	-0.32598636	-0.3044363	-0.29962765	-0.28738857	-0.27295261	-0.26893889	-0.26038604	0.00185461	3.59659077	-0.29868041	-0.28505846	MouseUp
29	-0.2942917	-0.29093891	-0.27708674	-0.29238462	-0.25406697	0.0012109	3.59014864	-0.29435052	-0.31029738	-0.30191349	-0.31051623	-0.32356529	-0.31624924	-0.32139795	MouseUp
30	-0.00290941	3.59488513	-0.28292767	-0.30589747	-0.30566903	-0.31937968	-0.32750774	-0.31701556	-0.33430582	-0.29477642	-0.28842824	-0.27994343	-0.30646694	-0.26296265	MouseUp
31	-0.3210822	-0.32529527	-0.2964794	-0.33959448	-0.30112657	-0.28628153	-0.28609269	-0.29887113	-0.26942002	-0.02095322	3.57985777	-0.29551572	-0.3081719	-0.30633231	MouseUp
32	-0.2942917	-0.2685207	-0.2887677	-0.25729916	-0.00133421	3.58029453	-0.28544332	-0.30794334	-0.2977096	-0.32891429	-0.30922041	-0.31368546	-0.33049697	-0.2908838	MouseUp
33	3.59604601	-0.28993692	-0.30239527	-0.28968003	-0.308914	-0.31335912	-0.32107756	-0.35070161	-0.28034663	-0.29412467	-0.28977806	-0.29551572	-0.26440694	-0.18335997	MouseUp
34	-0.323226	-0.3265886	-0.32770291	-0.29358748	-0.29658412	-0.28693158	-0.30034985	-0.25804861	-0.02739076	3.57963801	-0.29885132	-0.3200562	-0.30480515	-0.32273787	MouseUp

Esto no fue suficiente para tener resultados satisfactorios, por lo que posteriormente se les aplicó a los datos una Transformada de Fourier bajo el enfoque de magnitud y fase, lo cual resultó en dos columnas nuevas por cada canal previo, es decir, un conjunto de datos como el que se visualiza en la figura 10.4.

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	Char
1	Expected Ou	Channel 1 M	Channel 1 F	Channel 2 M	Channel 2 F	Channel 3 M	Channel 3 F	Channel 4 M	Channel 4 F	Channel 5 M	Channel 5 F	Channel 6 M	Channel 6 F	Channel 7 M	Channel 7 F	Channel 8 M	Channel 8 F	Channel 9 M	Channel 9 F	Channel 10 A	Char
2	MouseUp	1.81E-11	3.14089232	2.69E-12	-3.13975537	2.90E-12	0.00061284	2.19E-11	-3.14148136	7.40E-13	-0.01020373	1.05E-11	3.13922088	1.18E-11	-3.14027226	3.57E-12	-0.0073179	4.12E-13	-0.06692692	1.06E-12	0.0
3	MouseUp	2.93900559	-1.25008215	2.50341212	-0.89289976	5.84901209	1.77771061	2.20684177	3.10237807	1.87269001	-3.12849875	2.77383383	-0.88891732	4.48538224	0.01267239	5.4631156	1.81409506	8.05810508	-2.63468713	1.96462983	2.9
4	MouseUp	3.46682979	1.87574368	3.52779372	1.82553404	3.10686589	-0.50067118	0.46768427	-2.53047776	0.50591637	-1.20880212	3.73370605	1.79821064	2.37462762	-2.93955231	3.54829881	-0.65013366	3.05627842	-1.66962159	0.51651136	-1.1
5	MouseUp	3.96459182	-1.59505162	2.68633903	-1.57122212	4.23840348	0.56872161	1.2045269	0.59333102	1.24852047	0.57774349	2.78762294	-1.49885995	1.17080447	2.51934543	3.47258323	0.69700563	2.77702542	1.26689267	1.42785151	0.4
6	MouseUp	3.14327357	0.61942908	2.01488349	0.35952594	7.00078594	-1.50754144	2.24876514	-2.51986775	1.15057737	-2.62164453	3.60956239	0.79427578	5.37974038	-0.27636384	6.45987347	-1.61310323	7.53073932	2.73849567	1.6922576	-2.8
7	MouseUp	0.87713377	0.92093553	0.59835282	0.43697566	3.28507963	2.98206382	3.99052633	-0.07918309	3.49633339	-0.3987833	0.64758937	0.40833287	3.93977053	-3.11097283	3.96610489	3.05999339	3.09589656	-2.88877797	4.15672266	-0.1
8	MouseUp	2.76186739	-0.80731314	2.0099461	-1.18326051	6.83707633	1.69160406	2.56192493	2.92641689	1.46842647	-2.8373902	2.34212065	-0.63483424	4.01699233	-0.0050231	6.17922348	1.70666672	7.92202929	-2.67811951	1.28953495	-2.6
9	MouseUp	3.77950678	1.70265272	3.16357122	1.69416286	3.33155006	-0.67191598	1.2617478	-0.81590387	1.13245549	-1.285801	2.82302298	2.01965178	1.86616622	-2.34302701	3.43441779	-0.7754994	3.21770036	-1.68641381	0.94626277	-0.1
10	MouseUp	3.35054677	-1.43286632	3.37545342	-1.63946995	3.09840663	0.34912297	0.69116146	0.38991181	1.02496114	2.11636042	3.11755981	-1.89041366	0.92438965	-3.06807652	2.92598201	0.22512775	2.68310074	1.69565799	1.65341785	1.4
11	MouseUp	2.02713522	0.95503394	2.56790613	0.56594741	6.71945722	-1.39717034	2.30540823	3.12116293	1.82858651	-2.67459599	2.93480077	0.53818926	5.57946011	-0.25750949	6.06398309	-1.51316817	7.5180235	2.71341679	1.48349379	-2.5
12	MouseUp	0.39058035	-0.1176368	0.60955201	-0.83333782	2.35852961	2.96151014	4.60036354	0.04878768	0.05195472	-0.22109416	0.3184357	-1.52775752	3.74033917	-2.90110794	3.2673666	-3.02836646	3.27407833	3.00503177	3.78074981	0.0
13	MouseUp	2.69140918	-0.70702807	2.54239277	-0.72262743	6.16377329	1.41466797	1.54310277	3.04089417	0.53461582	2.53538252	3.19764224	-0.59244842	5.00271612	0.04339932	5.68082529	1.47087844	7.54489762	-2.61882462	0.95494316	-2.9
14	MouseUp	4.4521443	1.73746033	3.50606638	1.53794048	3.51496522	-0.43432682	1.04369068	0.79362342	0.93934898	-0.65029659	4.16473118	1.88318912	1.16195091	3.0765156	3.37395117	-0.6825528	3.33989321	-1.3685277	1.05540507	-0.8
15	MouseUp	2.66504107	-1.94205061	3.15086441	-1.99468227	3.92518269	0.89875545	1.02648957	-0.06844177	2.05461849	1.29388862	2.19225406	-0.03288991	1.67427336	2.51158125	4.32715928	1.04413173	3.13970436	1.843427	2.0954659	1.4
16	MouseUp	2.53556351	0.82469239	2.32773113	0.48991849	5.7583514	-1.55708886	1.2410657	2.98251907	2.16480088	-2.68276105	1.95225678	0.48082869	4.16261419	-0.28033299	5.54880648	-1.67761429	7.07358137	2.57568828	2.45007748	-2.9
17	MouseUp	0.5848342	0.88021864	0.85634826	-0.16200526	2.36027472	-2.98885656	5.31362393	-0.0805265	4.66664492	0.13361764	1.32416936	0.69100349	2.5382211	2.87544763	3.30829682	2.87653809	2.33625477	3.03661877	4.76152625	-0.0
18	MouseUp	3.10079342	-0.80907455	2.80107432	-0.48305376	6.5067873	1.39126212	1.93653219	2.44408041	1.19664139	-3.12825807	3.99130331	-0.6843746	5.45751515	0.15613095	5.66953424	1.50784115	6.26881081	-7.70340259	0.85852521	-1.9
19	MouseUp	3.9760708	1.76679105	4.50953719	1.66890437	3.9046617	0.48858747	1.32820502	-0.01669054	0.81642666	1.32955499	0.5781091	1.8009849	1.40559911	2.82169014	3.43214515	-0.53778195	3.23281692	1.53261152	0.56113865	-0.2
20	MouseUp	2.95732884	-1.53465355	2.6724002	-1.72519583	3.14980476	0.9992615	0.81274039	0.53370053	0.75826055	-0.02329499	2.40232889	-1.64199933	2.10592386	2.76985694	2.79571293	0.90504812	2.13434425	1.67903408	0.64362661	0.0
21	MouseUp	3.17350108	0.78807506	2.89253824	0.86058122	6.1344066	-1.50347285	0.08822101	-1.79901021	1.35932904	-2.61147382	0.30933363	0.84553677	3.67104673	0.03230848	5.4423493	-1.55481471	6.88272068	2.5903515	1.09283096	2.5
22	MouseUp	0.54340108	0.29504224	1.18632181	-0.6518258	3.35740154	-1.31174513	4.16779357	-0.31156622	4.56055727	-0.12238404	0.86862012	-0.00135421	3.13963444	-2.81579311	3.77848706	3.01582746	3.71510101	-2.96058259	3.95487529	-0.1
23	MouseUp	3.16744051	-0.87616133	2.84412901	-0.78437583	6.206799	1.51909637	1.18482383	-2.97154499	0.7646381	-2.7945657	6.1027521	-0.92418578	5.37828878	0.14011321	6.27996981	1.45522733	7.3483594	-2.62318163	0.98385485	-2.9
24	MouseUp	3.80772828	1.82370001	3.80932369	1.98299701	1.10310621	-0.54947236	1.70815786	-0.84631773	0.70775013	0.00603166	4.25744768	1.82106921	2.83729728	-2.99302532	3.09406853	-0.6725251	3.13832165	-1.53430337	0.74487953	1.1
25	MouseUp	3.85385293	-1.79735014	3.76869498	-1.81420629	2.83029478	0.7811973	1.09310606	0.29587581	1.08934196	0.50949451	2.66502593	-2.0132194	1.7526098	2.80769026	2.86159444	0.83725613	2.84711428	1.34481828	1.39902396	1.3
26	MouseUp	2.65122518	1.07277709	2.50546999	1.47005511	5.86574247	-1.39804046	1.19865295	2.99617063	1.63224481	-2.71946985	2.33190745	0.77514139	3.84940252	-0.09443522	5.13960434	0.9575569	7.57553497	2.60279095	1.50487737	-2.6
27	MouseUp	0.52818857	-1.49036534	0.12517368	-0.5563638	3.35465019	-2.90407878	4.7151522	-0.13159574	4.6267422	-0.03628699	0.5418544	-0.83943881	3.4434178	-3.05131196	4.53842438	-2.86872133	3.07409929	-2.84478077	5.51494938	-0.1
28	MouseUp	3.26117895	-1.10692419	3.0389677	-0.31425603	6.83256579	1.50132699	2.50620922	2.53026160	1.45638498	-0.68324035	2.97621564	-0.85640866	3.93889933	0.38365645	6.81335166	1.57540159	7.802498	-2.75495801	0.96250434	2.4
29	MouseUp	3.94787318	1.93712677	2.98882308	1.78680015	3.4959792	0.83279023	0.93522871	-1.53560603	0.85567172	-0.66683401	3.23430022	2.12817377	3.22172867	-2.73843344	5.18638322	-0.73204378	2.72001791	-1.69767329	0.48359918	-0.4
30	MouseUp	4.42303936	-1.79165145	3.15783121	-1.67717603	2.33655845	0.61973503	1.30582446	0.90440798	0.44670032	1.68174973	2.93511521	-1.49109918	1.93508484	2.69401912	2.19216121	0.77526228	3.66008893	1.70682788	0.6645355	1.8
31	MouseUp	2.16621503	0.96000009	1.69172455	0.72621016	6.18000486	-1.42896409	1.87716869	-0.72560209	1.62387579	-2.77618558	3.34824827	0.79555123	4.73126749	0.05704245	5.44695799	-1.3118822	7.18946279	2.79446039	3.06466833	-2.5
32	MouseUp	1.18658055	-0.11815791	0.66656542	0.06000198	3.42923157	1.03463215	4.53700073	-0.78285082	4.30724114	0.07012981	4.83690818	-1.16250715	3.20061213	-2.96245352	4.00361083	3.12240379	3.70222547	2.96753928	4.29912568	0.1
33	MouseUp	2.71039531	-0.85469883	2.88662761	-0.43593944	7.08827856	1.38294731	2.11500462	2.48472851	1.6295097	-2.70175029	3.05828601	-0.94212635	5.23915942	0.26517578	5.54434783	1.44813933	7.50134721	-2.67463731	1.56674901	-2.7
34	MouseUp	2.91679434	1.70485405	2.7433901	2.04781586	3.54187143	-1.07915251	0.94822791	-0.98612259	1.06651859	0.97805804	3.7839247	2.0503924	2.77809922	-2.73938277	4.31146925	-1.00275992	3.49382732	-1.9196209	0.19914932	1.6

Figura 10.4. Fragmento del segundo archivo obtenido tras la Transformación de Fourier del voluntario 19.

Por último se ejecutaron las funciones de los tres modelos de aprendizaje supervisado únicamente indicando el perfil que se evaluará. Con este proceso, cada función arrojará un porcentaje de efectividad después de haber separado a la población en un conjunto de entrenamiento y un conjunto de prueba, produciendo una matriz de confusión en la que será visible la efectividad del modelo. Al ejecutar estos modelos en Python, el resultado similar al que se visualiza en la figura 10.5.

```
In [4]: RandomForest('Andy2')
Matriz de Confusión
[[139  6  4 11  1]
 [ 8 183  7  2  1]
 [12 13 145  1 10]
 [19  1  2 71 16]
 [ 7  2 11 17 15]]
Precisión del modelo
[0.75135135 0.89268293 0.85798817 0.69607843 0.84864865]
C:\Users\alexe\Anaconda3\envs\ici-thesis\lib\site-
packages\sklearn\metrics\_classification.py:1375: UserWarning: Note that
pos_label (set to 'a') is ignored when average != 'binary' (got None). You
may use labels=[pos_label] to specify a single positive class.
UserWarning,
Out[4]: array([0.75135135, 0.89268293, 0.85798817, 0.69607843,
0.84864865])
```

Figura 10.5. Resultado de ejecución del modelo de Random Forest sobre el segundo archivo del voluntario 10.

Finalmente este procedimiento se repitió varias veces, concretamente por modelo para cada archivo de cada voluntario se ejecutó cinco ocasiones distintas, de modo que se obtuviera un promedio de



estas ejecuciones. Todos estos resultados se almacenaron en una hoja de cálculo donde se realizaría la interpretación de resultados. Un fragmento de la hoja de cálculo puede verse en la figura 10.6 donde las ejecuciones están en celdas cuyo color es grisáceo mientras que los promedios de dichas ejecuciones están coloreados con distintas señalizaciones; naranja para SVM, azul para Naive Bayes y amarillo para Random Forest. Los resultados son presentados a continuación.

Ronda 1	SVM					Naive Bayes					Random Forest				
	Arbol	Cuaderno	Computadora	Perro	Nada	Arbol	Cuaderno	Computadora	Perro	Nada	Arbol	Cuaderno	Computadora	Perro	Nada
Joel Alejandro Espinoza Sánchez	0.28968254	0.34185304	0.38461538	0.31666667	0.30769231	0.2295082	0.21008403	0.20512821	0.4	0.26666667	0.55776892	0.7027027	0.72164948	0.66666667	0.65730337
	0.28384279	0.30116959	0.31531532	0.33333333	0.30681818	0.35185185	0.21052632	0.17391304	0.03571429	0.26923077	0.55072464	0.75342466	0.71590909	0.69180047	0.64705882
	0.33057851	0.31499421	0.264	0.30493274	0.33333333	0.27868852	0.2251497	0.21311475	0.11764706	0.25641026	0.58455882	0.70935961	0.80952381	0.74683175	0.7679558
	0.26859504	0.26842105	0.21052632	0.30392157	0.36986301	0.26315789	0.33289316	0.25737579	0.17543886	0.26315789	0.59160305	0.66515837	0.75	0.65934066	0.72560976
	0.28996283	0.33043478	0.37362637	0.34449761	0.32323232	0.27777778	0.21158392	0.21052632	0.16666667	0.26315789	0.44396552	0.5480427	0.58247423	0.63513514	0.62658228
José Luis Espinoza Sánchez	0.292532342	0.311372634	0.309616678	0.320670384	0.32818783	0.280196848	0.218047426	0.212051616	0.179093324	0.263724696	0.54572419	0.675737608	0.715911322	0.679806938	0.684902006
	0.31952663	0.66666667	0.71428571	0.5	0	0.48648649	0.22244094	0.20155039	0.54	0.23969942	0.47432024	0.52713178	0.52671756	0.58571429	0.50248756
	0.2688935	0.66666667	0	0.5	0.44827586	0.26666667	0.3943602	0.22244489	0.16666667	0.47674419	0.27777778	0.42353211	0.41044776	0.49478261	0.42948718
	0.31692913	0	0.25	0.2775	0	0.4	0.25382932	0.19847328	0.44036697	0.27428571	0.46397695	0.4379085	0.46280992	0.51333333	0.42236025
	0.32443532	0.328125	0	0.42105263	0.31333333	0.34567901	0.24074074	0.18656716	0.5	0.25146199	0.4548105	0.44936709	0.41843972	0.55479452	0.52083333
María Consuelo Sánchez Díaz	0.29003559	0.38888889	0.66666667	0.46153846	0.27672956	0.36363636	0.23766816	0.16666667	0.40322581	0.28481013	0.42105263	0.55714286	0.33027523	0.48780488	0.48734177
	0.311563204	0.410069446	0.476190476	0.41617139	0.226845912	0.398031612	0.23547481	0.183994824	0.472067394	0.265601006	0.447902486	0.476399599	0.414605008	0.31412264	0.478465048
	0.62406367	0	0	0	0	0.53242321	0.10869565	0	1	0.09910913	0.85930309	0.725	0.83333333	0.78461538	0.88148148
	0.61844569	0	0	0	0	0.59009099	0.10909091	0	0.12790698	0.09386471	0.84	0.70526316	0.73880597	0.76821192	0.83229814
	0.61095506	0	0	0	0	0.60669253	0.12328767	1	0	0	0.8652435	0.65829146	0.75167785	0.76595745	0.80480486
José Luis Espinoza González	0.64138577	0	0	0	0	0.65134863	0.12857143	0	0.28481013	0.08158765	0.84418146	0.68869365	0	0.72	0.74193548
	0.61610487	0	0	0	0	0.49315068	0.06779661	0.10578842	0	0	0.85434929	0.73737374	0.83316883	0.80740741	0.80722892
	0.622101012	0	0	0	0	0.574904832	0.107488454	0.221157684	0.282543422	0.054912298	0.852615468	0.686924802	0.774997196	0.773625528	0.809625012
	0.58333333	0	0	0	0	0.65346535	0	0.06451613	0.28571429	0.09025444	0.89965636	0.82816901	0.79640719	0.81764706	0.82872928
	0.56786962	0	0	0	0	0.62658228	0	0.05882353	1	0.09399721	0.90758621	0.8511236	0.82894737	0.89444444	0.80526316
Román Guadalupe de León Vázquez	0.57603093	0	0	0	0	0.67916667	0	0.05555556	0.2	0.08652412	0.89410188	0.83241758	0.86713287	0.79754601	0.8313253
	0.59514352	0	0	0	0	0.68	0.05	0	0.08776207	0.04040404	0.90253672	0.83060109	0.82580645	0.8575641	0.77777778
	0.56072431	0	0	0	0	0.688	0	0.0212766	0.5	0.08888889	0.87991995	0.82608696	0.8	0.85555556	0.86585366
	0.576620302	0	0	0	0	0.66544286	0.01	0.040034364	0.414695272	0.08001374	0.896760224	0.833679648	0.823658776	0.843551434	0.82178936
	0.38738739	0	0.36818182	0.34871795	0.34361233	0.28709625	0	0.23258014	0.22098369	0.30769231	0.85542169	0.84615385	0.9132948	0.9025641	0.93382888
Hiram Efraín Orocio García	0.38589212	0.39726027	0.34803922	0.48603352	0.35950413	0.26052632	0.5	0.27272727	0.24100719	0.2	0.81278539	0.86662871	0.90547264	0.91397849	0.91878173
	0.41666667	0	0.39189189	0.38679245	0	0.21314952	0	0.25	0.5	0.25581395	0.82882883	0.81081081	0.84615385	0.88235294	0.85875706
	0.3699187	0.38926174	0.32888889	0.40609137	0.38974359	0.28744929	0	0.18181818	0.24291498	0.47761194	0.79828326	0.70869655	0.82887701	0.83333333	0.90860215
	0.35080645	0.37067937	0.3632287	0.38541667	0	0.2591687	0	0.14285714	0.24587156	0.1875	0.70130435	0.87595815	0.95336788	0.92890995	0.88235294
	0.382134266	0.231430276	0.360046104	0.402610392	0.21857201	0.261476016	0.1	0.215992146	0.290155884	0.28572364	0.817324704	0.821894214	0.889433236	0.892227762	0.90884552
Andrea Melissa Almeida Ortega	0.81860792	0	0	0	0	0.84209663	0.06586826	0.06266667	0.06584992	0	0.89770911	0.53271028	0.56834532	0.33884298	0.82876712
	0.80946801	0	0	0	0	0.82857143	0.06696935	0.09571788	0.06617361	0	0.91009755	0.49206349	0.61481481	0.4375	0.83892617
	0.81626435	0	0	0	0	0.84061135	0.06744186	0.06727829	0.06064073	0	0.89558372	0.45669291	0.66666667	0.46153846	0.84459459
	0.8073588	0	0	0	0	0.83294664	0.05160269	0.08231707	0.06875934	0	0.88489583	0.48039216	0.56896552	0.37142857	0.94992086
	0.81673307	0	0	0	0	0.83372922	0.05355064	0.10174419	0.06280742	0	0.88823075	0.48101266	0.54954955	0.4375	0.87692308
	0.81368643	0	0	0	0	0.835711054	0.06110256	0.08594482	0.064846204	0	0.894903392	0.4885743	0.593668374	0.409362002	0.847626364
	0.40416667	0.37765957	0	0.45402299	0.52849741	0.24750499	0.29230769	0.32258065	0.28057554	0.33333333	0.78688525	0.93877551	0.91509434	0.88135593	0.90575916
	0.4625	0.36649215	0.40740741	0.49431818	0.49746193	0.23140496	0.14285714	0.33333333	0.43181818	0.2258045	0.74561404	0.86697248	0.9047619	0.87845304	0.90186078
	0.38129496	0.45333333	0	0.52095808	0.46575342	0.29354839	0.38461538	0.27202472	0.6875	0.20580235	0.88444444	0.89047610	0.96858639	0.92610837	0.97905759
	0.40540541	0.36312849	0.42180095	0.54098361	0.53723404	0.22985075	0.66666667	0.23809524	0.31159442	0.21428571	0.89302326	0.92718447	0.9742268	0.92056075	0.96858639
	0.3699187	0.38926174	0.32888889	0.40609137	0.38974359	0.19938335	0.2	0.5	0	0.20588235	0.85294118	0.902134529	0.92349727	0.94370997	0.98901099
	0.404657148	0.389975056	0.231619451	0.483274861	0.483738078	0.240338488	0.337289376	0.324242338	0.342297584	0.237038038	0.832581634	0.904950788	0.93723334	0.909955412	0.948874982

Figura 10.6. Resultados de ejecución de los tres modelos sobre el primer archivo de los primeros siete voluntarios.

## 11. Análisis e Interpretación de Resultados

A continuación se muestran los datos obtenidos; estos se recopilaban de los voluntarios a quienes previamente se les aseguró que la recopilación de información se manejaría con alta seguridad y prudencia (estipulado por escrito bajo un acuerdo de privacidad adjunto en el anexo 5 y entregado a cada uno de los participantes).

Posteriormente, cuando los usuarios aceptaban el acuerdo de privacidad, el procedimiento de experimentación comenzaba y se realizaban los pasos del protocolo mencionado que puede consultarse en el anexo 6 y que posterior a la experimentación se llevaron a cabo las acciones explicadas en el capítulo pasado.

Los resultados iniciales, procesados después de la experimentación de campo, mostraban la tendencia inicialmente expuesta en las figuras 11.1 y 11.2.

Las gráficas expuestas a continuación son gráficas de caja y bigotes donde cada caja representa un concepto a predecir. De izquierda a derecha están cinco cajas de los cinco conceptos a predecir por la SVM, después cinco cajas de los mismos cinco pero bajo Naive Bayes y finalmente las cinco cajas correspondientes a los mismos conceptos bajo el modelo de Random Forest.

El archivo uno posee, de izquierda a derecha los conceptos árbol, computadora, cuaderno, perro y el control de no concentrarse en nada. El archivo dos de izquierda a derecha abarca las intenciones de movimiento de mover el ratón o *mouse* de la computadora hacia arriba, abajo, izquierda, derecha y el control de no concentrarse en una intención específica.

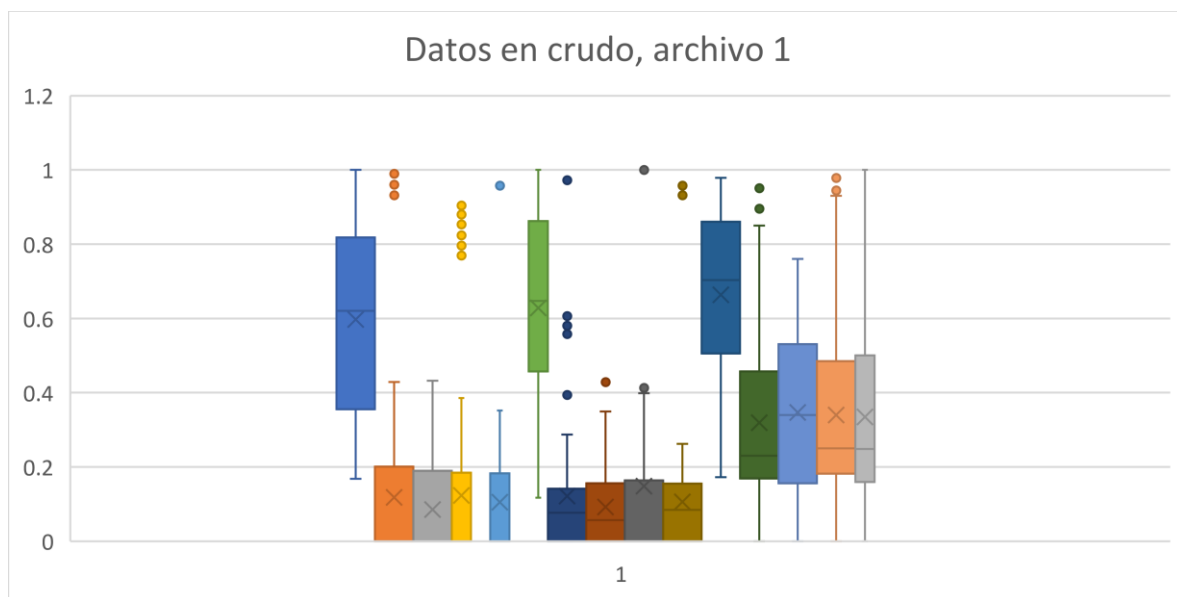


Figura 11.1. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos recopilados en la experimentación de campo usando el archivo uno de los voluntarios.

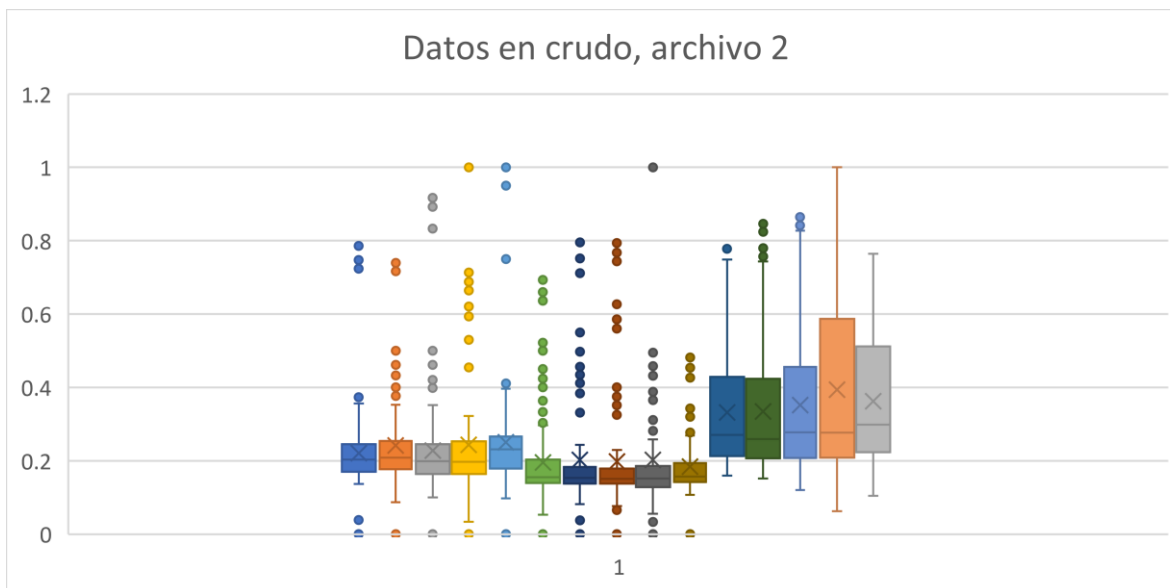


Figura 11.2. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos recopilados en la experimentación de campo usando el archivo dos de los voluntarios.

Se pueden apreciar distintas interpretaciones de estos resultados, sin embargo, la conclusión más destacable es que la precisión de los modelos no es fiable dado su bajo porcentaje de eficiencia alcanzado en general.

Debido a esto se le aplicó un procedimiento de normalización el cual fue la Normalización Z. Nuevamente se pasaron los datos por los modelos para observar si existía algún cambio. Estos cambios se reflejan en las figuras 11.3 y 11.4.

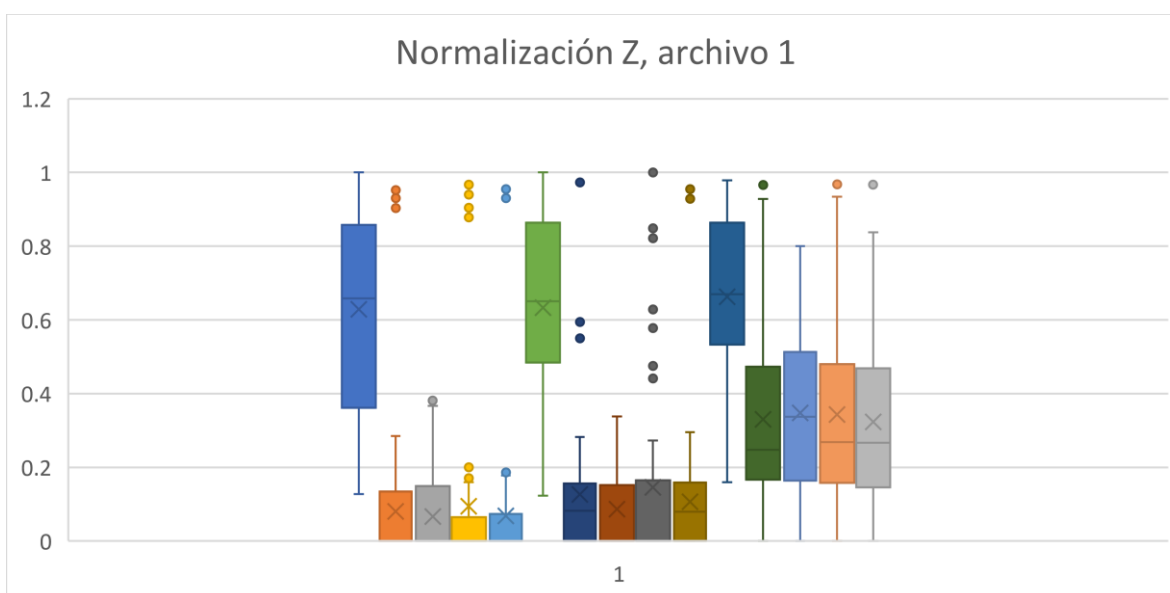


Figura 11.3. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos normalizados usando el archivo dos de los voluntarios.

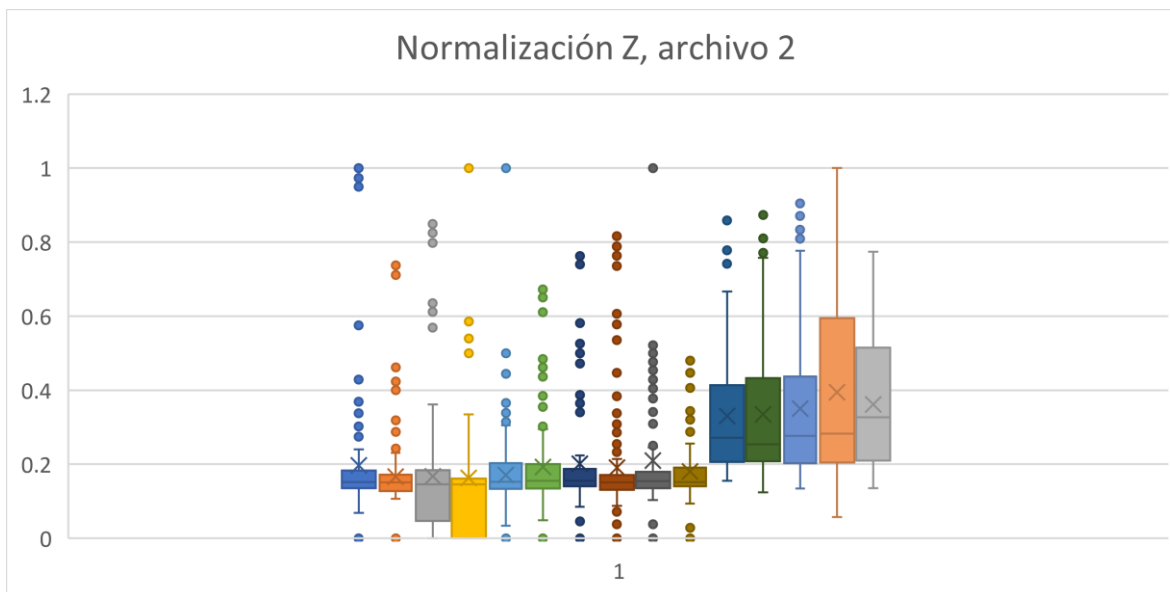


Figura 11.4. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos normalizados usando el archivo dos de los voluntarios.

Pese a que los resultados mejoraron considerablemente y esto haciendo énfasis en la evaluación del archivo dos, el cual contiene los datos de intención de movimiento, la normalización había brindado una mejoría esperanzadora, pero no era suficiente. Es por ello que se optó una última manipulación de datos realizando una transformación de Fourier a los datos, mediante un enfoque de magnitud y fase. Nuevamente los datos transformados serían reingresados a los modelos para su entrenamiento y predicción. Los resultados se presentan en las figuras 11.5 y 11.6.

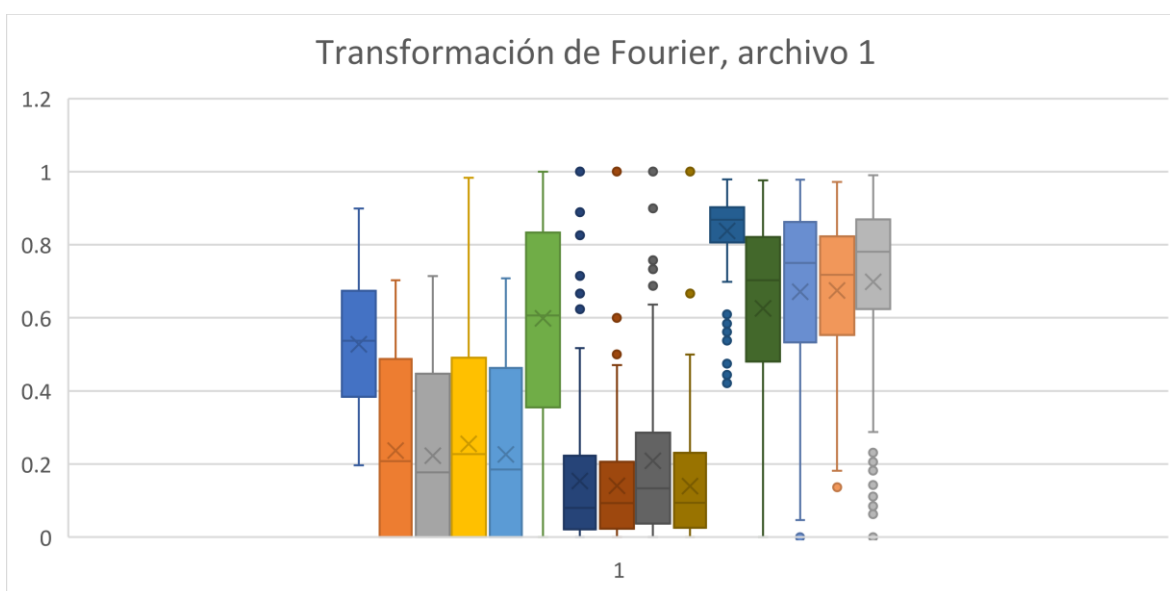


Figura 11.5. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos transformados mediante una transformada de Fourier usando el archivo dos de los voluntarios.

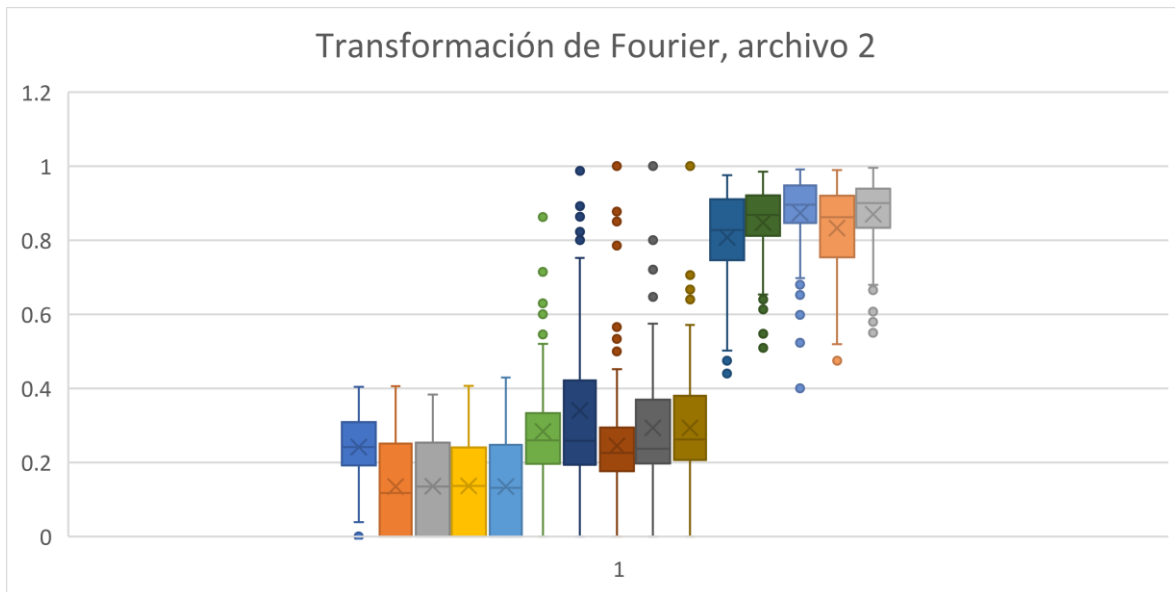


Figura 11.6. Gráfica de los resultados de ejecución de los tres modelos de machine learning usando los datos transformados mediante una transformada de Fourier usando el archivo dos de los voluntarios.

Es aquí donde se percibe el mayor cambio, pues a partir de dicha transformación, los modelos escalan de peor a mejor siendo el que menor alcance tiene el SVM, siguiendo con Naive Bayes que encuentra algunos valores atípicos en predicciones con alta fiabilidad pero estancándose aún en medias bajas y llegando al Random Forest que tiene una distribución muy alentadora dentro de este procedimiento, con una distribución de predicciones cercana a valores que pueden considerarse mayormente fiables, aún con ciertas fallas, pero éstas parecen ser más catalogadas como predicciones atípicas, de modo que con la transformación de Fourier puede observarse que de forma más eficiente el modelo de Random Forest obtiene resultados de predicción elevados posicionándose como el mejor modelo de entre los elegidos para esta investigación.

## 12. Conclusiones

La base de esta investigación yacía en la mejora de interfaces cerebro – computadora buscando con qué algoritmo de machine learning podría trabajarse mejor para realizar una interfaz de este tipo que fuera más efectiva. Con el uso de la diadema Emotiv Epoc+ junto con la paquetería de software mencionada además del código de Python desarrollado personalmente, se buscaba experimentar para conseguir datos de voluntarios quienes se concentrarían en pensamientos de intención de movimiento y así, usando los tres algoritmos de machine learning mencionados, entrenarlos con estos datos y observar la eficiencia que arrojan cada uno de estos.

A través del procedimiento se descubrieron muchas cosas como, primeramente que la hipótesis que señalaba personalmente donde el SVM sería el mejor algoritmo de entre los tres sería una hipótesis que no se cumplió al final del experimento, pues el algoritmo de Random Forest fue quien se llevó los mejores valores de precisión. Así también se descubrieron cuestiones como la importancia de pasos del preprocesamiento que en un principio no estaban contempladas, tales como la normalización y transformación de los datos.

Esta investigación está guiada para arrancar con el modelo vencedor de la presente investigación como objeto de mejora a la propuesta presentada en este documento, buscando perfeccionar aspectos como la transmisión y predicción en vivo, mientras los usuarios siguen usando la diadema o mejorar el porcentaje de efectividad y así realizar una implementación robusta de una interfaz cerebro – computadora tras el trabajo de esta y muchas investigaciones futuras.

También cabe recalcar que pueden existir puntos de mejora en la presente investigación, pues por ejemplo puede mejorarse el espacio y entorno de toma de datos con los voluntarios, perfeccionar las instrucciones de pensamientos deseados que se les dice a los voluntarios, buscar nuevos métodos de normalización o transformación así como nuevos modelos de machine learning con los que explorar nuevas posibilidades o partir con lo presentado y avanzar en las mejoras recientemente propuestas pues el conocimiento presentado en esta investigación tiene distintas vertientes, todas igualmente aceptables de explorar e indagar sobre el conocimiento que en esos campos puedan encontrarse y aportar a esta área del conocimiento.

### 13. Referencias

- Afifi, A., & Bergman, R. (1998). *Neuroanatomía funcional*. México, D.F.: McGraw Hill.
- Alpaydin, E. (2010). *Introduction to Machine Learning*. Cambridge: MA: MIT Press.
- APD, R. (04 de 04 de 2019). *¿Cuáles son los tipos de algoritmos del machine learning?* Obtenido de APD: <https://www.apd.es/algoritmos-del-machine-learning/>
- Band, A. (9 de Mayo de 2020). *Multi-class Classification — One-vs-All & One-vs-One*. Obtenido de Towards Data Science: <https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b>
- Barrett, K., Barman, S., Boitano, S., & Brooks, H. (2013). *Actividad eléctrica del cerebro, estados de sueño-vigilia y ritmos circadianos*. España: Access Medicine.
- Bayes, T. (1763). *An Essay towards solving a Problem in the Doctrine of Chances*. London: Philosophical Transactions of the Royal Society of London.
- Bear, M., Connors, B., & Paradiso, M. (2002). *Neurociencia: explorando el cerebro*. Barcelona: Masson.
- Betanzos Gómez, A. (13 de Marzo de 2020). *Principales algoritmos de Machine Learning*. Obtenido de LinkedIn: <https://www.linkedin.com/pulse/principales-algoritmos-de-machine-learning-alejandro-betanzos-g%C3%B3mez/?originalSubdomain=es>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.
- Blog Unipython. (15 de Enero de 2018). *Support Vector Machines (SVM)*. Obtenido de Unipython: <https://unipython.com/support-vector-machines-svm/>
- Brownlee, J. (2021 de Abril de 27). *One-vs-Rest and One-vs-One for Multi-Class Classification*. Obtenido de Machine Learning Mastery: <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/>
- Caldaria. (22 de Julio de 2020). *Curiosidades sobre el cerebro*. Obtenido de HDOSO Magazine: <https://www.caldaria.es/curiosidades-cerebro/>
- Cayre, M., Malaterre, J., Scotto-Lomassese, S., Strambi, C., & Strambi, A. (2010). The common properties of neurogenesis in the adult brain: from invertebrates to vertebrates. *Comparative Biochemistry and Physiology Part B: Biochemistry and Molecular Biology*, 1-15.
- Cromer, A. (1996). *Física para ciencias de la vida*. New York: Reverté.
- Data Scientist. (25 de Enero de 2022). *Random Forest: Bosque aleatorio. Definición y funcionamiento*. Obtenido de Data Scientist: <https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento>
- Denby, B., Schultz, T., Honda, K., Hueber, T., Gilbert, J., & Brumberg, J. (2010). Silent Speech Interfaces. *Speech Communication*, 270-287.
- Ebbing, D., & Gammon, S. (2010). *Química General*. México, D.F.: Cengage Learning.
- El Estadístico Blogspot. (17 de Febrero de 2021). *Random Forest explicado de forma sencilla*. Obtenido de Blog Estadístico: <https://elestadistico.blogspot.com/2021/02/random-forest-explicado-de-forma.html>

- Fisio, O. (27 de Septiembre de 2022). *¿Qué es soma o cuerpo neuronal?* Obtenido de Fisioterapia Online: <https://www.fisioterapia-online.com/glosario/soma-o-cuerpo-neuronal>
- Flach, P. (2012). *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge: Cambridge University Press.
- Flynn, K. (2013). The cytoskeleton and neurite initiation. *Bioarchitecture*, 86-109.
- Fundy. (15 de Octubre de 2021). *Coding Minecraft to work with Dance Moves...* Obtenido de YouTube: <https://www.youtube.com/watch?v=z2sGFFXuu38>
- González Barrio, H., Calleja Ochoa, A., Gómez-Escudero, G., Rodríguez Ezquerro, A., & López de Lacalle Marcaide, L. (06 de 04 de 2021). *Los conceptos de Machine Learning y Deep Learning en la industria*. Obtenido de Interempresas: <https://www.interempresas.net/MetalMecanica/Articulos/347471-Los-conceptos-de-Machine-Learning-y-Deep-Learning-en-la-industria.html>
- Gray, P. (2002). *Psychology*. New York: Worth Publishers.
- Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. *Hum Neurosci*.
- Hermann, N. (22 de Diciembre de 1997). *What is the function of the various brainwaves?* Obtenido de Scientific American: <https://www.scientificamerican.com/article/what-is-the-function-of-t-1997-12-22/>
- Hill, R. (2006). *Fisiología Animal*. Bogotá: Médica Panamericana.
- Hodgkin, A., & Huxley, A. (1939). Action Potentials Recorded from Inside a Nerve Fibre. *Nature*, 710-711.
- Interpsiquis. (25 de Septiembre de 2022). *Lóbulo Límbico*. Obtenido de Congreso Virtual de Psiquiatría: <https://psiquiatria.com/glosario/lobulo-limbico>
- Joyanes Aguilar, L. (2008). *Fundamentos de Programación*. Madrid: McGraw Hill.
- Kandel, E., Schwartz, J., & Jessel, T. (2000). *Principles of Neural Science*. New York: McGraw Hill.
- Kole, M., & Stuart, G. (2012). Signal processing in the axon initial segment. *Neuron*, 235-247.
- Laguna, M. (17 de Marzo de 2022). *Lóbulo de la ínsula*. Obtenido de Kenhub: <https://www.kenhub.com/es/library/anatomia-es/lobulo-de-la-insula>
- Latarjet, M., & Ruiz Liard, A. (2004). Encéfalo, Generalidades y Definición. *Anatomía Humana*, 168-169.
- Marín, A., Martínez, F., Ureña, L., & López, P. (6 de Septiembre de 2017). *El Lenguaje del Pensamiento*. Obtenido de Ciencia Cognitiva: <https://www.cienciacognitiva.org/?p=1502>
- Marius, H. (9 de Junio de 2020). *Multiclass Classification with Support Vector Machines (SVM), Dual Problem and Kernel Functions*. Obtenido de Towards Data Science: <https://towardsdatascience.com/multiclass-classification-with-support-vector-machines-svm-kernel-trick-kernel-functions-f9d5377d6f02>
- Martínez Heras, J. (29 de Septiembre de 2020). *¿Clasificación o Regresión?* Obtenido de IArtificial: <https://www.iartificial.net/clasificacion-o-regresion/>
- MDurance. (30 de Septiembre de 2021). *Todo lo que debes saber sobre el potencial de acción*. Obtenido de MDurance: <https://blog.mdurance.eu/academia/el-potencial-de-accion/>



- Merck & Co, I. (27 de Septiembre de 2022). *Estructura típica de una neurona*. Obtenido de Manual MSD: <https://www.msmanuals.com/es/hogar/multimedia/figure/estructura-t%C3%ADpica-de-una-neurona>
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge: MA: MIT Press.
- Murre, J., & Sturdy, D. (1995). The connectivity of the brain: multi-level quantitative analysis. *Biological cybernetics*, 73.
- Nazareno, J. (8 de Junio de 2020). *Fascinación con el cerebro y lo neuro*. Obtenido de Conexiones. Plataforma de Ciencias del Aprendizaje: [https://thelearningsciences.com/fascinacion\\_cerebro\\_neuro/](https://thelearningsciences.com/fascinacion_cerebro_neuro/)
- Ojeda Sahagún, J. L., & Icardo de la Escalera, J. M. (2004). *Neuroanatomía humana*. Barcelona: Masson.
- Olmo, M., Nave, A., & Nave, R. (5 de Octubre de 2022). *Action Potentials*. Obtenido de Hyperphysics: <http://hyperphysics.phy-astr.gsu.edu/hbasees/Biology/actpot.html>
- Orellana Alvear, J. (16 de Noviembre de 2018). *Árboles de decisión y Random Forest*. Obtenido de Bookdown: <https://bookdown.org/content/2031/ensambladores-random-forest-parte-i.html>
- Paniagua, R., Nistal, M., Sesma, P., Álvarez-Uría, M., Fraile, B., Anadón, R., & Sáez, F. (2002). *Citología e histología vegetal y animal*. España: McGraw Hill.
- Parrás, D., & Tedesco, A. (s.f.). *Probabilidad*. Buenos Aires: <https://estadisticadeluxemburgo.netlify.app/page4.html>.
- Parzen, E. (1987). *Teoría moderna de probabilidades y sus aplicaciones*. California: Limusa.
- Pelvig, D., Pakkenberg, H., Stark, A., & Pakkenberg, B. (2008). Neocortical glial cell numbers in human brains. *Neurobiology of aging*, 11.
- Randall, D., Burggren, W., & French, K. (1998). *Eckert Fisiología Animal*. Girona: McGraw Hill.
- Roche, H. (2003). *Lexikon Medizin*. Alemania: Urban & Schwarzenberg. Obtenido de Urban & Schwarzenberg
- Roman, V. (25 de Abril de 2019). *Algoritmos Naive Bayes: Fundamentos e Implementación*. Obtenido de Medium: <https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f>
- Russell, S., & Norvig, P. (2004). *Inteligencia Artificial: Un Enfoque Moderno*. Madrid: Pearson.
- Sabater, V. (3 de Agosto de 2020). *Lóbulos cerebrales: características y funciones*. Obtenido de La Mente Es Maravillosa: <https://lamenteesmaravillosa.com/lobulos-cerebrales-caracteristicas-y-funciones/>
- Sáenz, J. (2013). *Cálculo Vectorial*. Venezuela: Hipotenusa.
- Sanabria Castro, A. (19 de Mayo de 2020). *Una introducción a los Árboles de Decisión*. Obtenido de Grupo Dabia: <https://www.grupodabia.com/post/2020-05-19-arbol-de-decision/>
- Sheikh, A., & Korm, E. (1994). *Imagery in Sports and Physical Performance*. New York: Baywood Publishing Company.
- Singer, J. (2006). *Imagery in Psychotherapy*. Washington D.C.: American Psychological Association.

- Solé, R., & Manrubia, S. (1996). *Neurodinámica*. Barcelona: Edicions UPC.
- Sosa Romano, L. (21 de Septiembre de 2022). *Fisiología de la actividad eléctrica del cerebro: electroencefalografía*. Obtenido de Departamento de Fisiología. UNAM.: <https://fisiologia.facmed.unam.mx/index.php/category/unidad-tematica-i/>
- Sruthi, E. R. (30 de Noviembre de 2022). *Understanding Random Forest*. Obtenido de Analytics Vidhya: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>
- The MathWorks Inc. (s.f.). *Support Vector Machine (SVM)*. Obtenido de MathWorks: <https://la.mathworks.com/discovery/support-vector-machine.html>
- TIBCO Data Science. (12 de Septiembre de 2022). *¿Qué es el aprendizaje supervisado?* Obtenido de TIBCO: <https://www.tibco.com/es/reference-center/what-is-supervised-learning>
- Torres-García, A. A., Reyes-García, C. A., Villaseñor-Pineda, L., & Ramírez-Cortés, J. (2013). Análisis de señales electroencefalográficas para la clasificación de habla imaginada. *Revista mexicana de ingeniería biomédica*, 23-39.
- Triglia, A. (23 de Agosto de 2016). *Sistema límbico: la parte emocional del cerebro*. Obtenido de Psicología y Mente: <https://psicologiaymente.com/neurociencias/sistema-limbico-cerebro>
- Urrestarazu, E. (2022). *Electroencefalograma*. Obtenido de Clínica Universidad de Navarra: <https://www.cun.es/enfermedades-tratamientos/pruebas-diagnosticas/electroencefalograma>
- von Bartheld, C., Bahney, J., & Herculano-Houzel, S. (2016). The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *The Journal of Comparative Neurology*, 18.
- Walpole, R., Myers, R., Myers, S., & Ye, K. (2012). *Probabilidad y Estadística para Ingeniería y Ciencias*. San Antonio: Pearson.
- Wikipedia. (1 de Octubre de 2022). *Neurona*. Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/Neurona>
- Wikipedia. (8 de Noviembre de 2022). *Ondas cerebrales*. Obtenido de Wikipedia: [https://es.wikipedia.org/wiki/Ondas\\_cerebrales](https://es.wikipedia.org/wiki/Ondas_cerebrales)
- Wikipedia. (s.f.). *Electroencefalografía*. Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/Electroencefalograf%C3%ADa>

## 14. Anexos

### 14.1. Anexo 1: Configuración para recibir datos desde la diadema hacia el equipo de cómputo usando la paquetería de software de Emotiv, OpenViBE, y la librería de Python: Cykit.

Primeramente, dado que este proyecto está usando la diadema de la marca Emotiv, será necesario instalar el software de esta marca. La paquetería de programas puede encontrarse – a la fecha de redacción de esta investigación – en la siguiente liga la cual es la página de descargas del software. De la misma forma, en la figura 14.1.1.

<https://www.emotiv.com/emotiv-launcher/>

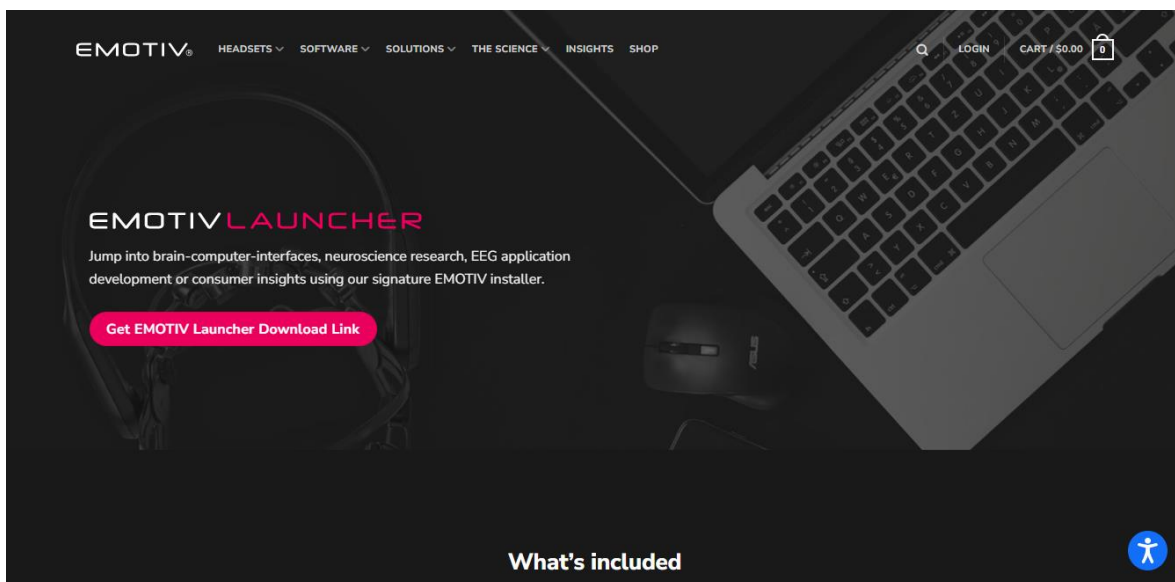


Figura 14.1.1. Sitio web de Emotiv.

En el sitio web anterior se descargará e instalará el paquete cuyo nombre es “Emotiv Launcher”. Durante el proceso de instalación se preguntará por el conjunto de programas a instalar, entre los cuales están “Emotiv Launcher”, “Emotiv BrainViz”, “EmotivPRO” y “EmotivBCI”. Como elección personal, se decidió instalar todo el conjunto de programas ofrecidos durante la instalación. Para poder usar los programas es necesario registrarse en la página de Emotiv y crear lo que la compañía llama un “Emotiv ID” que es una cuenta propia dentro de su sitio web. Esto puede hacerse desde la siguiente liga.

<https://www.emotiv.com/my-account/>

Asimismo, se muestra en la figura 14.1.2 el ejemplo visual del sitio web en el que hay que realizar el registro de la cuenta Emotiv para poder hacer uso del software de esta empresa.

## Create Your EmotivID

---

Fields with (\*) are required.

**EmotivID \***

**Password \***

Password must:

- be at least 8 characters long.
- contain at least one upper and one lowercase.
- not equal to username or email.

**Confirm Your Password \***

**Email \***

Activation email will be sent to this address.

We will also grab your [Gravatar](#) from this address.

Figura 14.1.2. Registro para la cuenta Emotiv ID.

Una vez completada la instalación, el programa Emotiv Launcher será el encargado de detectar si una diadema Emotiv está conectada al equipo. Puede observarse en la figura 14.1.3 este programa antes de conectar una diadema. Pero para conectar la diadema, es necesario – al menos en el caso del modelo Epoc+ – encender la diadema y conectar el dispositivo USB Bluetooth que viene con el paquete de la diadema tal como se aprecia en la figura 14.1.4.

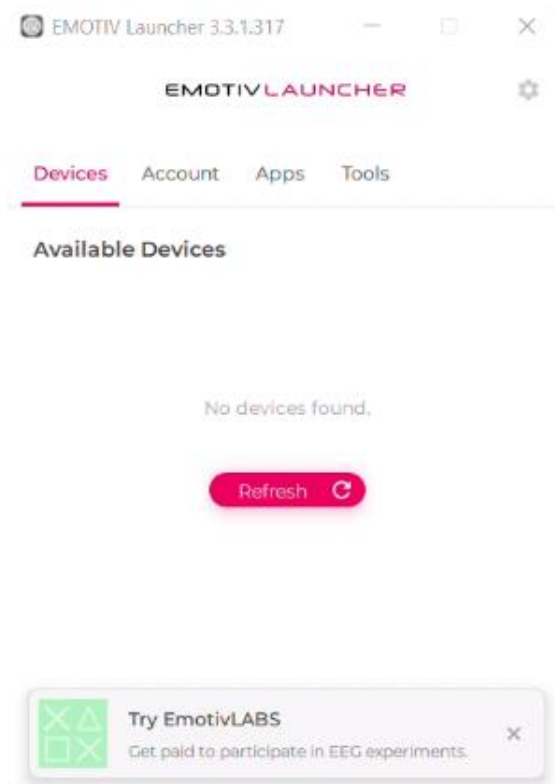


Figura 14.1.3. Emotiv Launcher sin la detección de una diadema conectada al equipo.



Figura 14.1.4. Diadema Emotiv encendida y dispositivo USB conectado al equipo.

Una vez que la conexión se haya establecido correctamente, Emotiv Launcher mostrará la diadema que detecta y permitiéndole al usuario conectarse a ella desde la interfaz de su programa, tal y como se muestra en la figura 14.1.5.

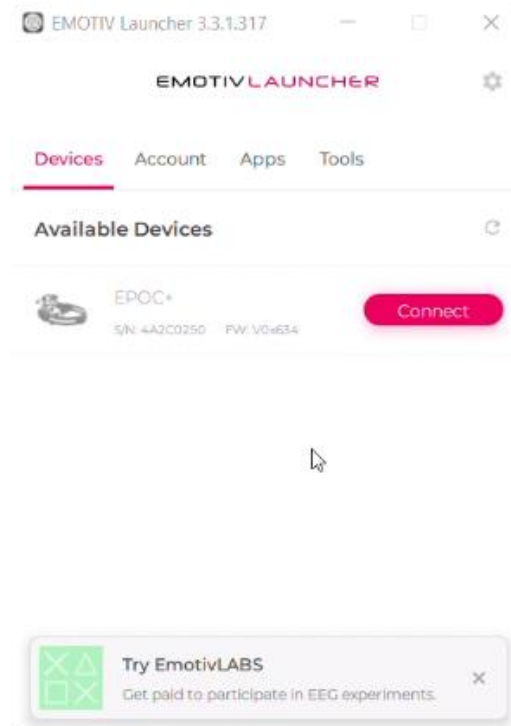


Figura 14.1.5. Emotiv Launcher detectando una diadema conectada al equipo.

A continuación, es necesario que el usuario en cuestión acomode la diadema sobre su cabeza de la manera correcta, realizando un acomodo que permita tanto una gran calidad de contacto y de recepción de las señales EEG. La figura 14.1.6 muestra este procedimiento guiado por la interfaz de Emotiv Launcher.

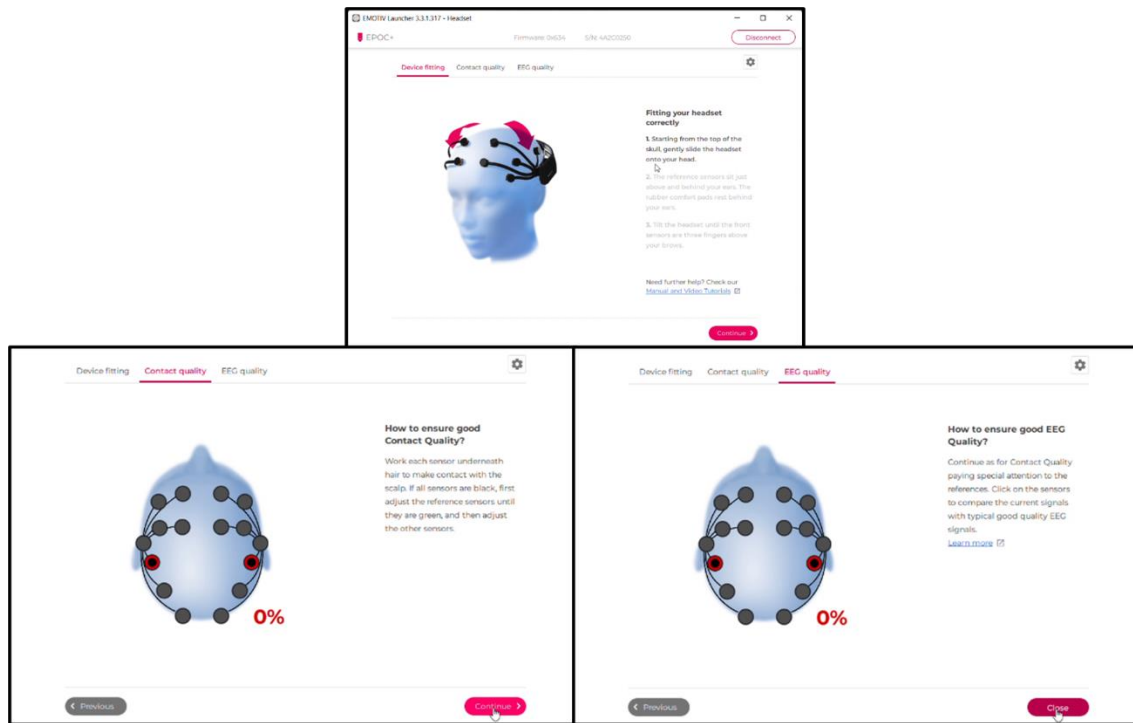


Figura 14.1.6. Emotiv Launcher guiando al usuario a acomodar correctamente la diadema.

Una vez que esté correctamente configurada la diadema con el equipo puede apagarse para un posterior uso. Es necesario también instalar el repositorio de GitHub creado por CymatiCorp: CyKit el cual puede encontrarse en la siguiente liga:

<https://github.com/CymatiCorp/CyKit>

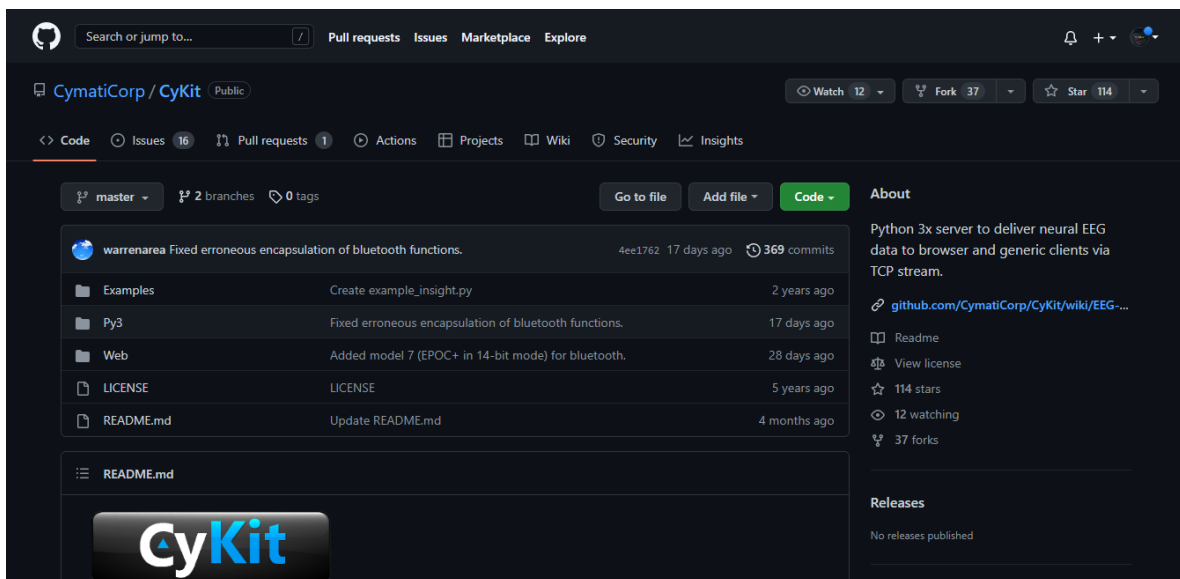


Figura 14.1.7. Sitio web de GitHub donde se encuentra el repositorio del proyecto de CyKit por CymatiCorp.

El sitio web puede observarse en la figura 14.1.7. Desde este sitio web puede accederse a la documentación para instalar este paquete y usarlo junto a una herramienta denominada OpenViBE para producir información a partir de la diadema. Las dos siguientes ligas corresponden a cada apartado de la documentación.

<https://github.com/CymatiCorp/CyKit/wiki/How-to-Install-CyKIT>

<https://github.com/CymatiCorp/CyKit/wiki/How-to-Stream-Data-to-OpenViBE>

Se tomará estrictamente lo que dicen ambas guías a continuación para poder avanzar con las instalaciones pertinentes.

Como, inicialmente se debe hacer, habrá que instalar primeramente la librería CyKit; para ello será necesario contar con una versión de Python, ya sea la 3.7.2 o superior. En el caso propio se trabajó con la versión 3.10.

Posteriormente hay que descargar el repositorio. Esto puede hacerse de varias formas, ya sea descargando el archivo ZIP con todo el contenido del repositorio o usando la tecnología Git – que será el caso propio – clonando el repositorio en el directorio del gusto personal usando el siguiente comando:

```
git clone https://github.com/CymatiCorp/CyKit.git
```

A continuación, puede probarse si la instalación se realizó correctamente probando el programa. Para ello, en una consola que esté ubicada en la raíz del directorio en donde se descargó el repositorio se ejecutarían los siguientes comandos:

```
cd CyKit-master  
cd Py3  
python CyKIT.py
```



```
The pairing name can easily be found in Windows Bluetooth settings.

Join these options (in any order), using a + separator.
(e.g info+confirm )

Example Usage:
Python.exe .\CyKIT.py 127.0.0.1 54123 1 info+confirm

Example Usage:
Python.exe .\CyKIT.py 127.0.0.1 5555 6 openvibe+generic+nocounter+noheader+nobattery+ovdelay:100+integer+ovsamples:004

> USB Device (No Additional Information)
> No USB Device Available. Exiting . . .

C:\Users\Propietario\Desktop\Cykit\CyKit\Py3>
```

Figura 14.1.8. Ejecución correcta de CyKIT.py usando Python.

En la figura 14.1.8 puede observarse la ejecución correcta del archivo CyKIT.py usando Python y por consecuencia, una correcta instalación de la paquetería. Ahora que ya está instalado el paquete de Python, para transmitir datos será necesario ejecutar el archivo similar a como anteriormente se realizó, pero deben de tomarse algunas consideraciones dependiendo de la diadema, frecuencia de transmisión, entre otras cosas, las cuales menciona CymatiCorp en la explicación de instalación y que puede apreciarse en la figura 14.1.9 y en la figura 14.1.10

```
Use +float if you plan to stream 32 bits float
Use +integer if you plan to stream 16 bit unsigned integer

(Tip: Streaming 32 bits float seemed to offer slightly better performance.)

Use the correct Keymodel ( 6 is used for EPOC+ in 16-bit mode, 7 would be used for EPOC+ 14-bit EPOC mode)
Premium models are likely prototypes, that are not often used. If you are in doubt, always try the consumer model.

1 - Epoc (Premium Model)
2 - Epoc (Consumer Model)
3 - Insight (Premium Model)
4 - Insight (Consumer Model)
5 - Epoc+ (Premium Model)
6 - Epoc+ (Consumer Model) [16-bit EPOC+ mode]
7 - EPOC+ (Consumer Model) [14-bit EPOC mode]
```

Figura 14.1.9. Explicación de algunas variables de configuración para ejecutar CyKit.

`+openvibe` This option indicates an OpenViBE data stream will be sent in a format readable by the `Generic Raw Telnet Reader` format. (More information in Step 5.)

`+generic` This option indicates a generic TCP stream has been created.  
Mainly used to differentiate between a generic TCP stream, that anything can connect to and between a TCP stream that may be used by an HTTP web socket. (Requiring a handshake.)

`+nocounter` This option is used to remove the first two bytes of data that is typically sent in the data stream. Where the two bytes are `[COUNTER] [INTERPOLATE]`

`[COUNTER]` being the EEG packet number 0 - 127 (or 0 - 255 for 256hz mode)  
`[INTERPOLATE]` being the number `16` for EEG data, and `32` to indicate data is gyro data.

By default, this option enables `+eegmode` which will send only EEG data, (ignoring Gyro data which could potentially corrupt the stream of data.) It sets this to EEG data by default, because otherwise there would be no way to indicate if the data stream is for EEG data, or Gyro data.

`+noheader` by default, CyKIT will send header information to both "Generic" TCP servers, and a websocket server. This header information includes various useful information about what config settings were set to run the CyKIT server. This header information is passed along, so that other programs can make use of the various settings available to CyKIT.

The header information is not however useful to programs like OpenViBE, so we include this config flag to disable sending this extra info.

`+nobattery` by default, the battery and quality settings are mixed into the data.  
CyKIT sorts this data and sends it via two bytes, which is appended to the EEG data packet.

The battery and quality information are not in any format recognizable by OpenViBE, so this data should be removed to satisfy the OpenViBE format.

`+ovdelay:NNN` is a custom delay counter, designed to work in milliseconds, but results may vary by CPU speed. Where NNN is a number between 001 and 999. This delay will occur before each packet is sent to OpenViBE.

`+float` determines the format the microvoltage data will be sent in. (32-bit big-endian float) "Big Endian"

`+integer` determines the format the microvoltage data will be sent in. (16-bit unsigned integer) "Big Endian"

`+ovsamples:NNN` defines how many samples will be collected in a block of data, before sending to the openViBE server. (This will relate to Step 5, determining `Sample counter per sent block`)

Default for `ovsamples` is 004.

CyKIT can set the sample block to any number between 001 and 999. However OpenViBE only allows specific block sizes.

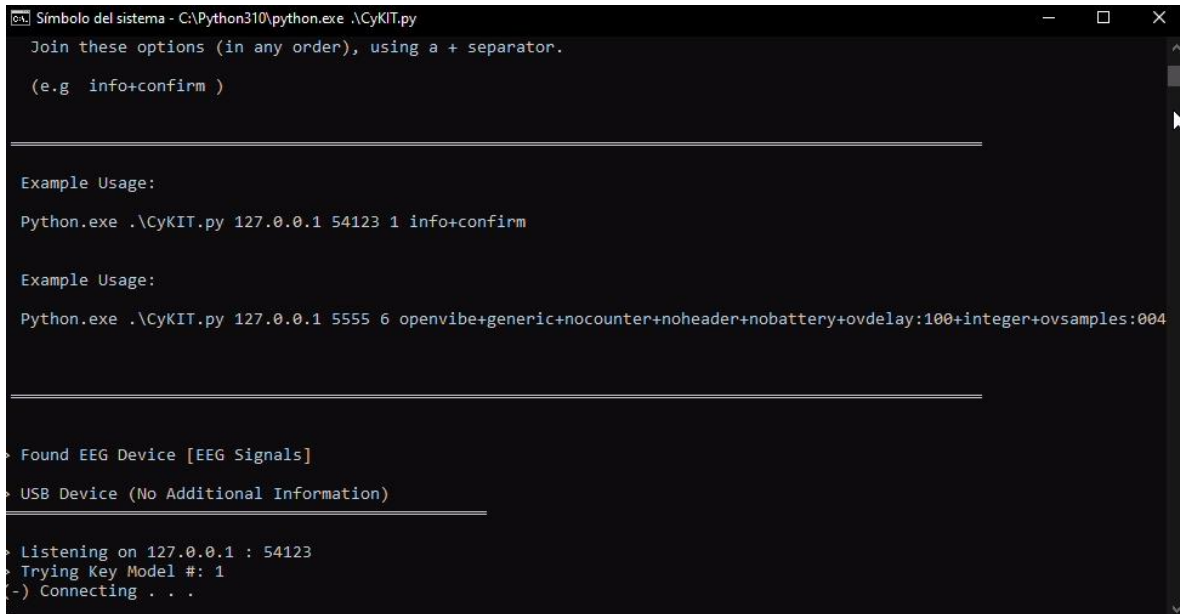
OpenViBE allows for the following sample block sizes: `004, 016, 032, 064, 128, 256, 512`  
Where the larger the packet block is, the longer it will take to be processed by OpenViBE functions.

(See below, for altering OpenViBE to accept 001 packet samples, for faster stream processing.)

Figura 14.1.10. Explicación de algunas variables de configuración para ejecutar CyKit.

En el caso propio, tomando en cuenta las configuraciones recomendadas por CymatiCorp, se eligió por ejecutar el siguiente comando:

```
python CyKIT.py 127.0.0.1 5151 6  
openvibe+generic+nocounter+noheader+nobattery+ovdelay:100+float+ovsamples:004
```



```
Símbolo del sistema - C:\Python310\python.exe .\CyKIT.py  
Join these options (in any order), using a + separator.  
(e.g. info+confirm )  
  
Example Usage:  
Python.exe .\CyKIT.py 127.0.0.1 54123 1 info+confirm  
  
Example Usage:  
Python.exe .\CyKIT.py 127.0.0.1 5555 6 openvibe+generic+nocounter+noheader+nobattery+ovdelay:100+integer+ovsamples:004  
  
Found EEG Device [EEG Signals]  
USB Device (No Additional Information)  
  
Listening on 127.0.0.1 : 54123  
Trying Key Model #: 1  
(-) Connecting . . .
```

Figura 14.1.11. CyKit ejecutándose bajo las configuraciones indicadas.

En la figura 14.1.11 puede observarse el correcto funcionamiento de CyKit al calibrarse con las opciones especificadas. Ahora, el siguiente paso será instalar OpenViBE, el cual se descarga desde la siguiente liga:

<http://openvibe.inria.fr/downloads/>

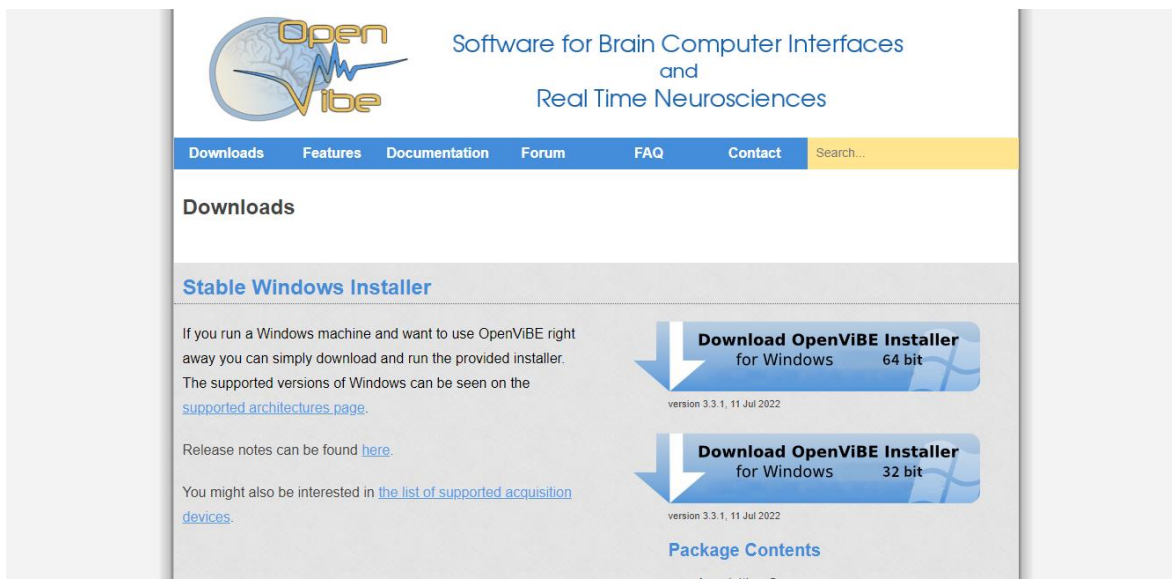


Figura 14.1.12. Sitio web de OpenViBE.

La versión que CymatiCorp recomienda para el trabajo presente es usar de la versión 2.2.0 en adelante. En el caso personal, se usará la versión 3.3.1 de 64 bits. De cumplir con este requisito, lo siguiente es ejecutar los programas “OpenViBE (Acquisition Server)” y “OpenViBE (Designer)”. La instalación no genera un acceso directo a estos programas, por lo que habrá que abrirllos en su mismo directorio o usando atajos que apunten a este directorio, según afirma CymatiCorp como algo necesario.

En OpenViBE (Acquisition Server) deberá seleccionarse “Generic Raw Telnet Reader” como Driver. El puerto de conexión (Connection port) es un puerto local al cual se conectará OpenViBE Designer. Este se deja por facilidad en 1024. Así también, en “Sample count per sent block” se dejará en 4. La configuración anterior puede observarse en la figura 14.1.13.

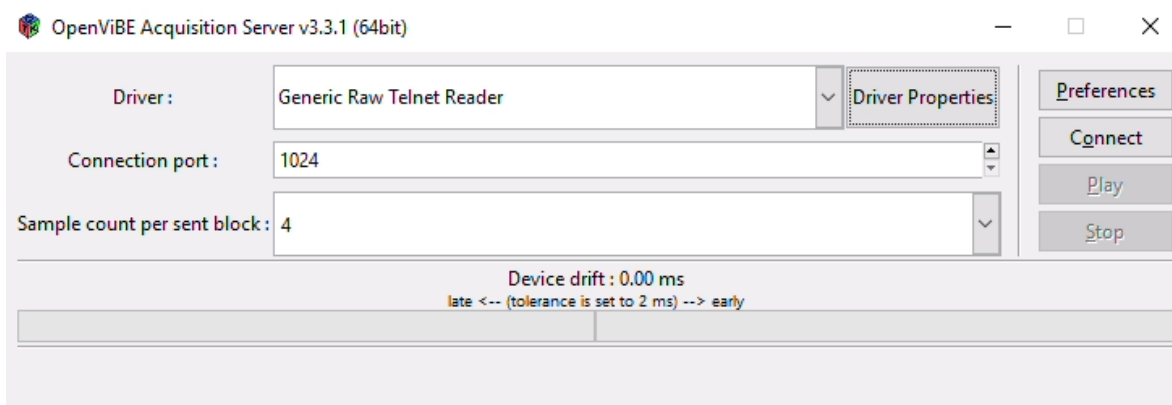


Figura 14.1.13. Configuración general de OpenViBE Acquisition Server.

Lo siguiente es modificar las opciones de propiedades del driver del botón “Driver Properties”. La configuración aparece en la figura 14.1.14 pero también se enumera a continuación la utilizada:

- **Number of channels:** 14.
- **Sampling frequency:** 256.
- **Telnet host name:** localhost.
- **Telnet host port:** 5151.
- **Endianess:** Big Endian.
- **Sample type:** 32 bits float.
- **Skip at start (bytes):** 0.
- **Skip header (bytes):** 0.
- **Skip footer (bytes):** 0.

Después de haber hecho estas configuraciones, se dará click en el botón de “Apply” y en OpenViBE Acquisition Server podrá dársele click en el botón “Connect” con CyKit.py ya ejecutándose para este punto del avance experimental. La conexión exitosa debe aparecer tanto en CyKit como en OpenViBE Acquisition Server como se muestra en la figura 14.1.15.

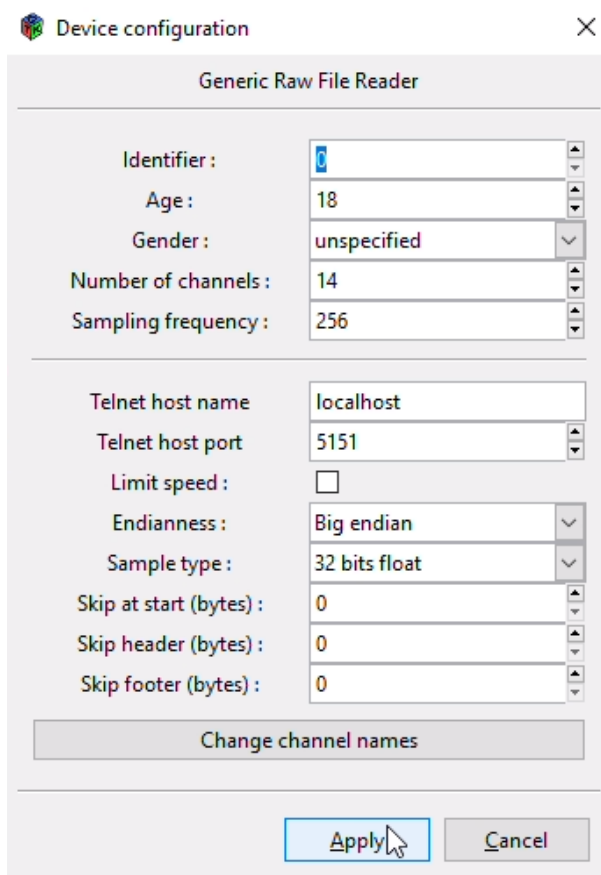


Figura 14.1.14. Configuración de propiedades del driver de OpenViBE Acquisition Server.



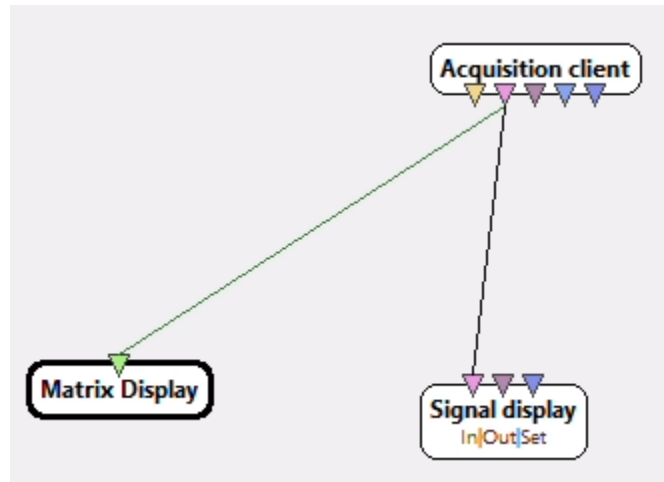


Figura 14.1.16. Ilustración del escenario creado en OpenViBE Designer.

Después de lo anterior, sólo queda darle al botón de ejecutar escenario y si todos los pasos se realizaron correctamente, un electroencefalograma y una tabla de valores deberían aparecer en pantalla, actualizando el estado de recepción de datos de la diadema; lo cual demuestra el funcionamiento correcto de la transmisión de datos. Esto se ilustra en las figuras 14.1.17 y 14.1.18.

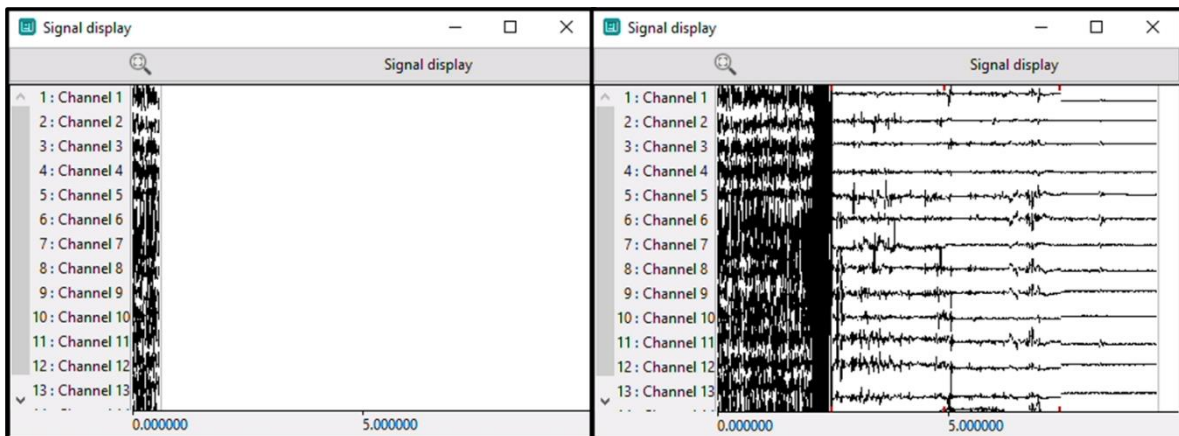


Figura 14.1.17. Electroencefalograma de OpenViBE Designer en funcionamiento correcto.



Channel	Col 1	Col 2	Col 3	Col 4
Channel 1	4157.69	4158.33	4155.26	4162.82
Channel 2	4156.92	4159.10	4157.69	4160.00
Channel 3	4159.62	4162.44	4159.10	4157.69
Channel 4	4150.13	4157.69	4155.90	4159.23
Channel 5	4163.46	4152.69	4158.21	4153.08
Channel 6	4160.90	4159.74	4154.87	4157.05
Channel 7	4152.18	4156.79	4151.92	4160.13
Channel 8	4157.56	4158.59	4159.49	4153.85
Channel 9	4162.69	4162.05	4162.44	4159.36
Channel 10	4166.15	4163.97	4160.77	4160.38
Channel 11	4161.54	4159.62	4158.33	4161.15
Channel 12	4164.74	4166.28	4161.79	4163.97
Channel 13	4217.82	4214.49	4212.69	4211.54
Channel 14	4923.97	4922.95	4922.31	4923.85

Figura 14.1.18. Tabla de datos de OpenViBE Designer en funcionamiento correcto.

Hasta aquí llega la explicación de CymatiCorp para la transmisión de datos. Sin embargo, personalmente todavía no se sentía con mucha facilidad de hacer machine learning de esta forma de recopilar datos, por lo que se tuvieron que hacer algunos añadidos personales al aporte hecho por CymatiCorp.

Estos añadidos fueron dos. El primero está en el escenario realizado en OpenViBE Designer, pues al escenario se le agregará un proceso llamado “CSV File Writer” el cual puede encontrarse dentro del directorio “File reading and writing” → “CSV” en la barra lateral derecha. Después habrá que posicionarse sobre la flecha rosa de “Acquisition Client” y arrastrarla a la flecha rosa de “CSV File Writer”. Finalmente dando doble click sobre este último habrá que modificar el campo “Filename” al nombre y directorio de gusto propio. Así también debe marcarse como verdadera la casilla correspondiente a “Append data”. La figura 14.1.19 muestra el escenario modificado y la figura 14.1.20 muestra las configuraciones del proceso de “CSV File Writer”.

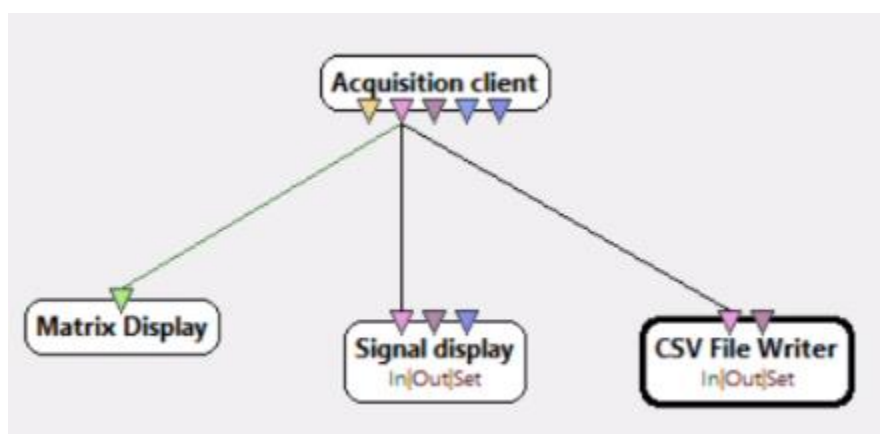


Figura 14.1.19. Escenario de OpenViBE Designer modificado personalmente.



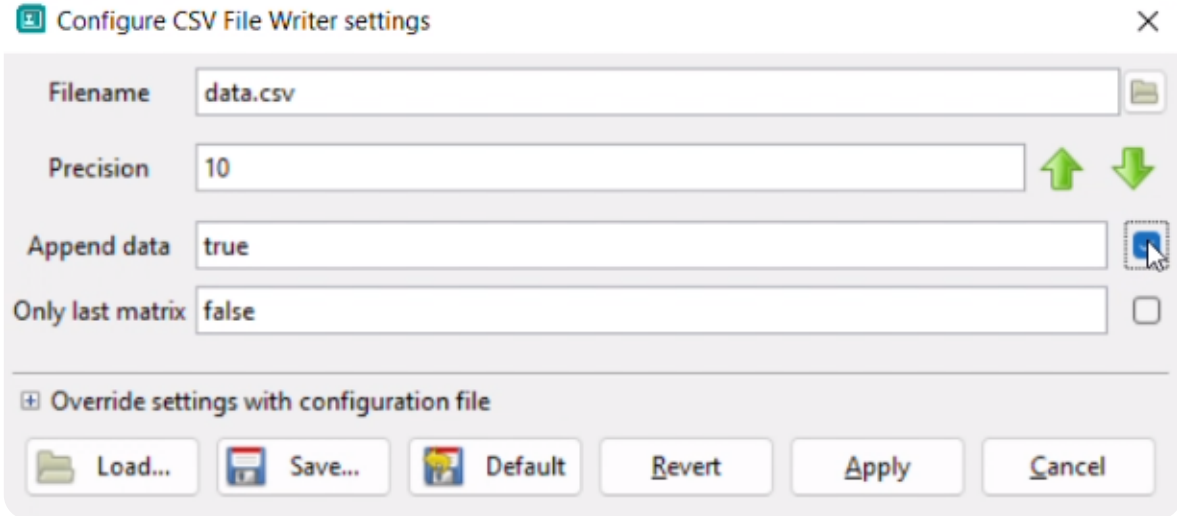


Figura 14.1.20. Configuraciones para "CSV File Writer".

Esto finalmente permite una salida de datos mejor tratable para el machine learning realizado en esta investigación. Puede observarse en la figura 14.1.21 la tabla en formato CSV que produjo este cambio, con la que ya puede hacer más fácil la lectura de datos para los modelos de redes neuronales que se usarían en la presente investigación.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Time:256Hz	Epoch	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11	Channel 12	Channel 13	Channel 14
2	0	0	4164.74365	4159.35889	4160.38477	4153.58984	4189.35889	4155.76904	4149.23096	4158.07715	4161.79492	4166.28223	4162.30762	4165.89746	4218.3335	4636.02588
3	0.00390625	0	4150.89746	4161.28223	4155.64111	4159.10254	4207.30762	4164.23096	4158.97412	4157.69238	4162.17969	4165.89746	4154.74365	4153.58984	4209.4873	4618.58984
4	0.0078125	0	4160.76904	4158.71777	4162.69238	4158.07715	4192.94873	4151.6665	4151.15381	4161.02588	4164.74365	4163.84619	4166.53857	4168.84619	4215.12842	4635.5127
5	0.01171875	0	4154.61523	4161.28223	4154.74365	4158.58984	4206.53857	4160.5127	4159.4873	4153.97412	4160.89746	4162.56396	4160	4161.15381	4213.07715	4621.02588
6	0.015625	1	4161.53857	4158.71777	4158.84619	4156.15381	4190.5127	4157.94873	4153.58984	4162.94873	4161.15381	4166.53857	4161.28223	4163.46143	4214.10254	4630.25635
7	0.01953125	1	4153.20508	4161.02588	4153.97412	4157.17969	4206.15381	4162.43604	4156.79492	4158.71777	4161.79492	4166.41016	4154.23096	4161.6665	4210.64111	4623.20508
8	0.0234375	1	4158.20508	4159.10254	4152.94873	4156.15381	4201.79492	4156.53857	4151.53857	4159.87158	4162.94873	4166.02588	4162.56396	4169.74365	4216.6665	4629.10254
9	0.02734375	1	4162.43604	4160.5127	4156.02588	4159.74365	4200.89746	4159.74365	4158.07715	4160	4160.76904	4166.53857	4166.15381	4166.79492	4218.07715	4623.84619
10	0.03125	2	4161.53857	4162.30762	4161.53857	4160.76904	4198.58984	4156.41016	4156.15381	4156.92285	4160.76904	4164.87158	4161.79492	4162.56396	4214.35889	4620.89746
11	0.03515625	2	4159.10254	4158.97412	4160.64111	4156.92285	4195.38477	4151.02588	4162.43604	4157.94873	4164.4873	4163.3335	4161.6665	4164.87158	4213.97412	4628.97412
12	0.0390625	2	4152.56396	4159.74365	4153.58984	4157.43604	4200	4158.97412	4157.56396	4158.84619	4165.64111	4164.35889	4160	4162.30762	4214.74365	4626.28223
13	0.04296875	2	4157.56396	4161.6665	4158.07715	4159.35889	4189.35889	4156.53857	4152.82031	4163.58984	4166.41016	4167.43604	4162.43604	4166.15381	4216.79492	4638.84619
14	0.046875	3	4150.25635	4163.20508	4154.87158	4160.89746	4203.84619	4158.97412	4157.56396	4154.61523	4161.41016	4165.76904	4156.79492	4162.69238	4211.02588	4622.82031
15	0.05078125	3	4161.41016	4156.28223	4160.25635	4158.97412	4190.38477	4152.17969	4152.56396	4160.64111	4162.43604	4164.87158	4164.74365	4172.94873	4218.58984	4633.20508
16	0.0546875	3	4153.58984	4160.5127	4156.92285	4163.20508	4208.46143	4158.3335	4159.4873	4152.69238	4160.38477	4164.4873	4154.35889	4155.12842	4211.41016	4612.56396
17	0.05859375	3	4163.71777	4160.25635	4165.12842	4157.82031	4192.82031	4148.58984	4150	4157.30762	4162.56396	4166.02588	4164.74365	4166.53857	4215.5127	4634.35889
18	0.0625	4	4150	4163.07715	4154.10254	4156.15381	4210.38477	4155.38477	4156.53857	4152.30762	4161.02588	4164.61523	4157.69238	4162.43604	4212.69238	4613.07715
19	0.06640625	4	4159.10254	4159.10254	4155	4155.76904	4190.25635	4146.15381	4145.25635	4162.17969	4161.53857	4166.92285	4161.02588	4166.92285	4217.82031	4623.58984
20	0.0703125	4	4155.76904	4163.84619	4150.25635	4162.30762	4206.92285	4159.61523	4157.69238	4155.64111	4159.10254	4167.82031	4154.87158	4157.94873	4217.53857	4617.56396
21	0.07421875	4	4162.69238	4161.53857	4157.17969	4159.35889	4192.17969	4155.25635	4152.82031	4160.5127	4162.69238	4165.76904	4165.5127	4169.4873	4217.82031	4636.15381
22	0.078125	5	4150.25635	4160.12842	4153.84619	4158.58984	4198.07715	4159.4873	4154.4873	4159.35889	4162.43604	4164.23096	4156.6665	4159.74365	4213.71777	4616.02588
23	0.08203125	5	4155.76904	4156.53857	4157.43604	4158.58984	4188.58984	4155.5127	4151.53857	4162.69238	4161.15381	4167.30762	4159.61523	4162.30762	4216.02588	4622.69238
24	0.0859375	5	4157.56396	4158.07715	4155.76904	4160.12842	4200.5127	4158.97412	4158.3335	4156.02588	4158.97412	4167.56396	4160.5127	4162.17969	4214.87158	4621.28223
25	0.08984375	5	4159.10254	4161.28223	4158.3335	4156.41016	4200.38477	4156.79492	4152.82031	4156.41016	4161.41016	4165.12842	4161.41016	4164.4873	4213.97412	4619.74365
26	0.09375	6	4154.35889	4162.43604	4160.64111	4153.58984	4198.97412	4154.87158	4148.71777	4162.30762	4162.94873	4163.46143	4157.94873	4161.15381	4213.3335	4624.4873
27	0.09765625	6	4151.41016	4159.87158	4158.71777	4156.53857	4203.20508	4155.5127	4153.3335	4156.28223	4158.84619	4165.38477	4155.89746	4156.79492	4211.41016	4616.79492
28	0.1015625	6	4163.46143	4158.84619	4159.61523	4157.56396	4195.12842	4154.74365	4152.56396	4156.02588	4160.5127	4170	4163.71777	4167.56396	4215.38477	4627.30762
29	0.10546875	6	4153.71777	4163.84619	4156.28223	4158.84619	4208.84619	4159.10254	4154.23096	4154.35889	4163.20508	4166.15381	4154.35889	4160.25635	4210.5127	4617.30762
30	0.109375	7	4157.43604	4158.46143	4161.92285	4156.15381	4191.53857	4148.46143	4148.46143	4160.25635	4162.56396	4162.05127	4158.71777	4162.05127	4214.4873	4628.58984
31	0.11328125	7	4153.58984	4160	4152.82031	4158.3335	4204.87158	4161.02588	4159.4873	4151.53857	4158.20508	4163.97412	4153.71777	4154.10254	4209.23096	4610.76904
32	0.1171875	7	4165.5127	4161.15381	4161.15381	4155.89746	4189.10254	4155.12842	4148.3335	4161.41016	4164.10254	4166.28223	4162.30762	4168.3335	4212.69238	4632.82031
33	0.12109375	7	4150.76904	4164.35889	4157.82031	4158.46143	4208.58984	4159.4873	4157.82031	4154.87158	4161.28223	4161.41016	4153.97412	4158.97412	4208.84619	4617.30762
34	0.125	8	4161.6665	4159.74365	4158.97412	4156.79492	4189.35889	4153.71777	4155.12842	4160.25635	4158.46143	4163.3335	4166.53857	4168.46143	4221.41016	4634.61523
35	0.12890625	8	4157.43604	4163.07715	4151.15381	4158.58984	4203.84619	4164.10254	4159.61523	4156.53857	4157.69238	4166.6665	4160.89746	4160.5127	4215.89746	4615.38477

Figura 14.1.21. Archivo CSV producido tras una ejecución de transmisión de datos desde la diadema.

Sin embargo, aún queda un problema más a resolver y es aquí donde está el segundo cambio notable; pues este archivo CSV se genera sólo hasta el final de la ejecución, es decir, cuando se detiene el proceso. Pero en una investigación y experimentación personal, se encontró una forma

de simular una lectura a tiempo real para poder realizar el análisis a tiempo real de los datos transmitidos desde la diadema al equipo, casi en vivo.

Esta forma de simulación se deberá hacer primeramente activando el ciclado del procedimiento. Esto se hace pulsando el botón de ciclado que está cerca del botón de ejecución, como se puede ver en la figura 14.1.22.

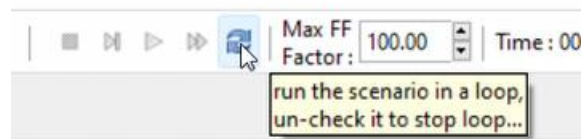


Figura 14.1.22. Botón para ciclar la ejecución.

Ahora, al haber activado esta opción, cada vez que se pulse el botón de detener el proceso, sólo se reiniciará el procedimiento, por lo que de esta manera puede simularse la transmisión en vivo de los datos.

Por último, para poder hacer este proceso automático se proponen dos archivos de Python que faciliten este trabajo. Estos archivos de Python requerirán de la librería “PyAutoGUI”, la cual puede instalarse con el siguiente comando:

```
pip install pyautogui
```

El propósito de estos archivos es que se pulse la detención cíclica del escenario periódicamente, lo cual hará la librería previamente instalada, pues la anterior permite controlar el teclado y el mouse de forma automática con código de Python.

Primeramente hay que encontrar las coordenadas del botón en la pantalla. Para ello se usaría el siguiente archivo Python:

```
1 import pyautogui
2 import time
3
4 # Aplicando delay para tener tiempo de acomodar el mouse sobre el botón
5 time.sleep(10)
6
7 # Obteniendo la posición del mouse
8 print(pyautogui.position())
```

Ahora, cuando se tengan estos valores pueden guardarse en variables “x”, “y” que se pueden usar en el siguiente código, pues ahora, cuando todo el procedimiento de obtención de datos esté en ejecución: la diadema conectada, Emotiv en ejecución y OpenViBE transmitiendo datos, es momento

de ejecutar el código siguiente para que la detención cíclica lo haga automáticamente este código propuesto:

```
1 import pyautogui
2 import time
3
4 x = 0
5 y = 0
6
7 # Aplicando delay al código para correr el escenario y automáticamente se
  mueva a la posición indicada
8 time.sleep(10)
9 pyautogui.moveTo(x, y)
10
11 # El proceso se mantiene en ejecución mientras el mouse esté en su posición
12 while(pyautogui.position() == (x, y)):
13     pyautogui.click()
14     time.sleep(1) # El CSV se actualiza cada segundo
```

Una combinación de ambos en un único código sería el siguiente:

```
1 import pyautogui
2 import time
3
4 # Aplicando delay para tener tiempo de acomodar el mouse sobre el botón
5 time.sleep(10)
6
7 x, y = pyautogui.position()
8
9 # Aplicando delay al código para correr el escenario y automáticamente se
  mueva a la posición indicada
10 time.sleep(10)
11 pyautogui.moveTo(x, y)
12
13 # El proceso se mantiene en ejecución mientras el mouse esté en su posición
14 while(pyautogui.position() == (x, y)):
15     pyautogui.click()
16     time.sleep(1) # El CSV se actualiza cada segundo
```

Y así, al ejecutar este código, se podrá actualizar el archivo CSV cada segundo, o según cuánto quiera personalizarse en la línea 16. Con esto ya podría realizarse machine learning con los datos de forma más efectiva para la presente investigación.

Este mismo procedimiento se documentó a su vez de forma audiovisual y se presenta en el siguiente enlace:

<https://youtu.be/MuBZdcLJXbk>

## 14.2. Anexo 2: Código del archivo de Python: functions.py.

El archivo de Python functions.py está constituido de las distintas funciones que a lo largo del procedimiento de investigación se realizaron y se compilaron en este documento. El archivo en su totalidad es el siguiente:

```
1  """ Libraries
2  import os
3  import time
4  import random
5  import pyautogui
6  import pandas
7  import numpy as np
8  import matplotlib.pyplot as plt
9  from sklearn.model_selection import train_test_split
10 from sklearn.svm import SVC
11 from sklearn.metrics import classification_report
12 from sklearn.metrics import precision_score
13 from sklearn.metrics import confusion_matrix
14 from sklearn.preprocessing import StandardScaler
15 from sklearn.ensemble import RandomForestClassifier
16 from sklearn.naive_bayes import GaussianNB
17
18 """ Detect the position of the mouse
19 def MousePosition(secs = 3):
20     # Sleeping the script so we have time to set the mouse on its position
21     time.sleep(secs)
22
23     # Getting the mouse position
24     print(pyautogui.position())
25
26     return
27
28 """ Run a live stream to update the dataset while streaming
29 def RunUpdateStream(xStop, yStop, xWindow, yWindow, timerRefresher):
30     # Sleeping the script so we have time to run the schema and then
31     # automatically moves to the indicated position
32     time.sleep(10)
33     pyautogui.moveTo(xWindow, yWindow)
34
35     # The process will be okay while the mouse keeps on the indicated
36     # positions
37     while((pyautogui.position() == (xStop, yStop)) or
38           (pyautogui.position() == (xWindow, yWindow))):
39         # Clicking to move the EEG
40         pyautogui.moveTo(xWindow, yWindow)
41         pyautogui.click()
42         time.sleep(timerRefresher/2)
43
44         # Clicking to reload the CSV
45         pyautogui.moveTo(xStop, yStop)
46         pyautogui.click()
47         time.sleep(timerRefresher/2)
48
49     return
```

```

47
48   """ Creating the dataset with the input data and the expected output
49   def createDataset(xMaximize, yMaximize, xPlay, yPlay, xWindow, yWindow,
50                     xStop, yStop, profileName, expectedResult, rangeTime):
51       # Check if profile path exists
52       if(not(os.path.exists('profiles/' + str(profileName)))):
53           os.makedirs('profiles/' + str(profileName))
54       time.sleep(0.2)
55
56       # Maximizing OpenViBE Designer
57       pyautogui.moveTo(xMaximize, yMaximize)
58       pyautogui.click()
59       time.sleep(1)
60
61       # Playing OpenViBE Designer
62       pyautogui.moveTo(xPlay, yPlay)
63       pyautogui.click()
64       time.sleep(rangeTime)
65
66       # Clicking over OpenViBE Designer
67       pyautogui.moveTo(xWindow, yWindow)
68       pyautogui.click()
69       time.sleep(0.5)
70
71       # Stopping OpenViBE Designer
72       pyautogui.moveTo(xStop, yStop)
73       pyautogui.click()
74       time.sleep(0.5)
75
76       # Loading data.csv
77       dataframe = pandas.read_csv('data.csv')
78
79       # Reworking the dataframe
80       dataframe = dataframe.drop(labels='Time:256Hz', axis=1)
81       dataframe = dataframe.drop(labels='Epoch', axis=1)
82       dataframe = dataframe.drop(labels='Event Id', axis=1)
83       dataframe = dataframe.drop(labels='Event Date', axis=1)
84       dataframe = dataframe.drop(labels='Event Duration', axis=1)
85
86       # Add a column with the expected output
87       dataframe.insert(16, "Expected Output", [expectedResult] *
88           len(dataframe))
89
90       # Append (or create) the dataset of this selected profile
91       if(not(os.path.isfile('profiles/' + str(profileName) +
92           '/dataset.csv'))):
93           file = open('profiles/' + str(profileName) + '/dataset.csv', "x")
94           file.close()
95           dataframe.to_csv('profiles/' + str(profileName) + '/dataset.csv',
96               mode='a', index=False, header=True)
97       else:
98           dataframe.to_csv('profiles/' + str(profileName) + '/dataset.csv',
99               mode='a', index=False, header=False)
100
101       # Remove the data.csv raw file

```

```

97     os.remove('data.csv')
98
99     return
100
101     """ First AI model
102     def SVM(profileName):
103         # Condition if profile or dataset doesn't exists
104         if(not(os.path.isfile('profiles/' + str(profileName) +
105                               '/dataset.csv'))):
106             print('User not found or dataset not created yet')
107             return
108
109         # Support Vector Machine process
110         df = pandas.read_csv('profiles/' + str(profileName) + '/dataset.csv')
111
112         X = df.drop(['Expected Output'], axis = 1)
113         Y = df['Expected Output']
114
115         xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size = 0.20)
116
117         classifier = SVC(kernel = 'linear')
118         classifier.fit(xtrain, ytrain)
119
120         ypred = classifier.predict(xtest)
121         print(ypred)
122
123         print(classification_report(ytest, ypred))
124
125         return
126
127     """ Second AI model
128     def NaiveBayes(profileName):
129         # Condition if profile or dataset doesn't exists
130         if(not(os.path.isfile('profiles/' + str(profileName) +
131                               '/dataset.csv'))):
132             print('User not found or dataset not created yet')
133             return
134
135         # Naive Bayes process
136         df = pandas.read_csv('profiles/' + str(profileName) + '/dataset.csv')
137
138         X = df.drop(['Expected Output'], axis = 1)
139         Y = df['Expected Output']
140
141         xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size = 0.20)
142
143         classifier = GaussianNB()
144         classifier.fit(xtrain, ytrain)
145
146         ypred = classifier.predict(xtest)
147
148         matriz = confusion_matrix(ytest, ypred)
149         print('Matriz de Confusión')
150         print(matriz)

```

```

151     precision = precision_score(ytest, ypred, pos_label="a")
152     print('Precisión del modelo')
153     print(precision)
154     return
155
156     """ Third AI model
157 def RandomForest(profileName):
158     # Condition if profile or dataset doesn't exists
159     if(not(os.path.isfile('profiles/' + str(profileName) +
160         '/dataset.csv'))):
161         print('User not found or dataset not created yet')
162         return
163
164     # Random Forest process
165     df = pandas.read_csv('profiles/' + str(profileName) + '/dataset.csv')
166
167     X = df.drop(['Expected Output'], axis = 1)
168     Y = df['Expected Output']
169
170     xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size = 0.20)
171
172     sc = StandardScaler()
173     xtrain = sc.fit_transform(xtrain)
174     xtest = sc.fit_transform(xtest)
175
176     classifier = RandomForestClassifier(n_estimators = 4)
177     classifier.fit(xtrain, ytrain)
178
179     ypred = classifier.predict(xtest)
180
181     matriz = confusion_matrix(ytest, ypred)
182     print('Matriz de Confusión')
183     print(matriz)
184
185     precision = precision_score(ytest, ypred, pos_label="a")
186     print('Precisión del modelo')
187     print(precision)
188     return

```

### 14.3. Anexo 3: Pseudocódigo de la función RunUpdateStream() encontrada en el archivo de Python: functions.py.

El acercamiento para nombrar las variables y revisar el procedimiento a modo de boceto fue el siguiente:

Las variables a utilizar serán las siguientes:

- xStop: La posición en x del botón de Stop.
- yStop: La posición en y del botón de Stop.
- xWindow: La posición en x en otro punto de la pantalla (para evitar que el EEG se interponga).
- yWindow: La posición en y en otro punto de la pantalla (para evitar que el EEG se interponga).
- timerRefresher: El tiempo que reposa el código que ocurre entre click y click para refrescar el data.csv.

El procedimiento será el siguiente:

- 1) Se asignan xStop y yStop a los valores que corresponden.
- 2) Se asignan xWindow y yWindow a los valores que corresponden.
- 3) Mientras la posición del mouse sea (xStop, yStop) o (xWindow, yWindow):
  - a. El mouse hace click sobre la posición (xWindow, yWindow).
  - b. El sistema hace un sleep de:

$$\frac{timerRefresher}{2}$$

- c. El mouse hace click sobre la posición (xStop, yStop).
- d. El sistema hace un sleep de:

$$\frac{timerRefresher}{2}$$



#### 14.4. Anexo 4: Pseudocódigo de la función createDataset() encontrada en el archivo de Python: functions.py.

El acercamiento para nombrar las variables y revisar el procedimiento a modo de boceto fue el siguiente:

Las variables a usar serán:

- xMaximize: La coordenada en x de donde se dará click para maximizar OpenViBE Designer.
- yMaximize: La coordenada en y de donde se dará click para maximizar OpenViBE Designer.
- xPlay: La coordenada en x de donde se dará click para darle play a OpenViBE Designer.
- yPlay: La coordenada en y de donde se dará click para darle play a OpenViBE Designer.
- xWindow: La coordenada en x de donde se da click para superponer a OpenViBE Designer.
- yWindow: La coordenada en y de donde se da click para superponer a OpenViBE Designer.
- xStop: La coordenada en x de donde se dará click para darle stop a OpenViBE Designer.
- yStop: La coordenada en y de donde se dará click para darle stop a OpenViBE Designer.
- profileName: El nombre de perfil que se usará.
- expectedResult: El resultado que se espera en la ejecución de la construcción del dataset.
- rangeTime: El tiempo en segundos que se empleará en la ejecución de la función.

El procedimiento será el siguiente:

- 1) Se checa si existe en la carpeta "profiles" una carpeta con nombre "profileName". Si no existe, se crea esta carpeta y esta se asigna como workspace, si existe directamente se asigna como workspace.
- 2) La función maximiza la ventana de OpenViBE Designer dándole click a (xMaximize, yMaximize).
- 3) Posteriormente la función da click en play en la ejecución del Designer en (xPlay, yPlay).
- 4) La función hará un sleep durante rangeTime recopilando los datos.
- 5) Al terminar este sleep, por precaución se da click en (xWindow, yWindow) y posteriormente a (xStop, yStop).
- 6) Python cargará el "data.csv" que se genera en el mismo directorio donde se encuentra el script para añadirle una columna extra "Expected Result" donde se añadiría la variable "expectedResult".
- 7) Este DataFrame se añade al archivo "dataset.csv" que se ubique en la carpeta del nombre de perfil dentro de "profiles". Si el archivo no existe, se crea, si sí existe, sólo se le hace un append al archivo ya existente.
- 8) Se elimina "data.csv".

#### **14.5. Anexo 5: Acuerdo de privacidad entregado a los usuarios que colaboraron con el experimento.**

En la página siguiente se adjunta el acuerdo de privacidad elaborado para asegurar a los usuarios que colaboraron con el experimento, que sus datos serían tratados éticamente según los fines de esta investigación y los permisos cedidos por ellos mismos.

# ACUERDO DE PRIVACIDAD

## 1. Introducción

Este acuerdo establece los términos y condiciones en los cuales los participantes (en adelante "Usuarios") en el estudio experimental "Implementación de una Interfaz Cerebro – Computadora por medio de la Diadema Emotiv Epoc+ y el Programa Cykit Utilizando Modelos de Aprendizaje Supervisado. Tesina de Joel Alejandro Espinoza Sánchez para Obtener el Grado de Ingeniero en Computación Inteligente" (en adelante "Estudio") ceden el derecho de uso de los datos experimentales generados con su participación por los investigadores en el estudio (en adelante "Investigadores"). Al firmar este acuerdo, los Usuarios aceptan las condiciones establecidas en este documento.

## 2. Objeto del acuerdo

Los Usuarios ceden al equipo de investigación el derecho de usar los datos experimentales generados con su participación en el estudio. Los datos que se generarán para el estudio son generados a través de la diadema Emotiv Epoc+. La información utilizada será únicamente el comportamiento cerebral generado por los Usuarios que registre la diadema. Los datos podrán ser utilizados para la realización de investigaciones, publicaciones y presentaciones en eventos científicos.

## 3. Tratamiento de los datos

Los Investigadores se comprometen a proteger la privacidad de los Usuarios y a tratar los datos de manera confidencial. Los datos se almacenarán de manera segura y solo serán accesibles por el equipo de investigación. Los datos serán anonimizados antes de ser compartidos con terceros, y solo se compartirán con instituciones y organizaciones con fines científicos.

## 4. Derechos de los Usuarios

Los Usuarios tienen derecho a acceder, rectificar y suprimir sus datos en cualquier momento, siempre y cuando no interfiera con el desarrollo del Estudio. También tienen derecho a retirar su consentimiento

en cualquier momento, siempre y cuando no afecte a la validez de los datos ya recolectados.

## 5. Permisos cedidos por el Usuario

El Usuario cede los permisos señalados a continuación como "Acepto" y restringe aquellos indicados como "No Acepto" señalizados y elegidos por el mismo Usuario:

- a) Que se me incluya en el proceso experimental del Estudio proporcionando mis datos relacionados a la actividad cerebral, generados con la diadema Emotiv Epoc+. Acepto: X No Acepto: \_\_\_\_
- b) Que se utilice como dato personal en la redacción, publicación y presentación del Estudio y subsecuentes, mi edad. Acepto: X No Acepto: \_\_\_\_
- c) Que se utilice como dato personal en la redacción, publicación y presentación del Estudio y subsecuentes, mi nombre completo. Acepto: \_\_\_\_ No Acepto: \_\_\_\_
- d) Que se puedan realizar grabaciones de audio y video incluyendo mi imagen únicamente como parte de la evidencia del proceso de realización del experimento del Estudio para el control de avances requerido por docentes y sinodales evaluadores del Estudio. Acepto: \_\_\_\_ No Acepto: \_\_\_\_
- e) Que el material audiovisual generado con mi imagen sea usado en la redacción, publicación y presentación del Estudio y subsecuentes (No aplica si no se acepta el inciso C). Acepto: \_\_\_\_ No Acepto: \_\_\_\_

## 6. Vigencia

Este acuerdo tendrá vigencia hasta la finalización del estudio, y será rescindido una vez que se hayan cumplido todos los objetivos del mismo.

## 7. Aceptación del acuerdo

La firma del presente acuerdo por parte del Usuario implica la aceptación de las condiciones establecidas en este documento.

\_\_\_\_\_  
Firma del Usuario

\_\_\_\_\_  
Fecha

\_\_\_\_\_  
Firma del Investigador

\_\_\_\_\_  
Fecha

## **14.6. Anexo 6: Protocolo de experimentación.**

Se adjunta en el presente anexo el protocolo de experimentación de campo realizado para la toma de muestras de los usuarios de prueba. El protocolo fue realizado para facilitar la toma de la gran cantidad de muestras así como de estandarizar el entorno de hardware y software para todos los usuarios.

Se aclaran los roles “usuario” e “investigador” como el usuario quien se probará la diadema y de quien se recolectarán datos y el supervisor de la prueba el cual está encargado de la coordinación de la experimentación respectivamente

### **Previo a la experimentación**

1. El investigador debe preparar el área de experimentación. Esto implica encontrar y preparar un espacio con poco ruido
2. El investigador debe preparar el equipo de experimentación. Esto implica preparar el equipo de cómputo propio:
  1. Tener el equipo encendido y con batería
  2. Tener abierto el programa de Spyder con el script de uso con el comando de creación del dataset preparado para ejecutarlo
  3. Tener abierto Emotiv Launcher
  4. Tener preparado el comando de ejecución de CyKit
  5. Tener abiertos OpenViBE Designer y OpenViBE Acquisition Server con las configuraciones del esquema apropiadas
3. El investigador debe preparar la diadema para la experimentación.
  1. Sacar la diadema de la caja
  2. Humedecer por aproximadamente cinco minutos las almohadillas en un bote con agua y después retirarlas para reposar del agua en una toalla
  3. Acomodar las almohadillas y los nodos eléctricos en cada uno de los receptores de la diadema
  4. Encender la diadema
  5. Conectar el USB inalámbrico al equipo
4. El investigador debe otorgarle al usuario el acuerdo de privacidad donde éste tiene que aceptar los términos para poder proseguir con el actual usuario
5. Si el usuario acepta los términos del acuerdo de privacidad, puede proseguirse con la experimentación

### Durante la experimentación

1. El investigador acomodará la diadema en la cabeza del usuario de modo que Emotiv Launcher muestre un alto porcentaje de recepción
2. Una vez que el porcentaje sea alto, se realizan las conexiones de software:

1. Se ejecuta CyKit:

```
cd 'CyKit\Py3'  
  
python CyKIT.py 127.0.0.1 5151 6  
openvibe+generic+nocounter+noheader+nobattery+ovdelay:100+float+ovsamples:004
```

2. Se conecta OpenViBE Acquisition Server

3. Al tener las conexiones realizadas, se ejecutará el comando en Python que crea el dataset:

```
createDataset(1211, 1048, 453, 77, 452, 15, 398, 77, 'perfil1', "Arbol", 10)
```

Se nombra la experimentación con el nombre del usuario seguido de un 1 y en este dataset se le pedirá al usuario:

1. Un árbol
2. Un cuaderno
3. Una computadora
4. Un perro
5. Poner la mente en blanco

4. Al tener las conexiones realizadas, se ejecutará el comando en Python que crea el dataset:

```
createDataset(1211, 1048, 453, 77, 452, 15, 398, 77, 'perfil2', "MouseUp", 10)
```

Se nombra la experimentación con el nombre del usuario seguido de un 1 y en este dataset se le pedirá al usuario:

1. Mover el mouse hacia arriba
2. Mover el mouse hacia abajo
3. Mover el mouse hacia izquierda
4. Mover el mouse hacia derecha
5. Poner la mente en blanco

5. Finalmente el investigador retira la diadema del usuario revisando que todas las almohadillas estén en la diadema o de lo contrario buscarlas previo a que el usuario se retire

6. Al verificar que el número de almohadillas sea el correcto, el investigador puede dejar ir al usuario

### Después de la experimentación

Si existen más usuarios después del recientemente finalizado, se repite desde el paso 4 previo a la experimentación hasta el paso 6 durante la experimentación para un nuevo usuario

1. Al terminar la experimentación se cierra Spyder, posteriormente se desconecta y se apaga el Acquisition Server, luego se cierra el Designer y el Acquisition Server y finalmente CyKit
2. Después se desconecta la USB de la diadema y se termina de apagar el equipo de cómputo
3. La diadema se desarma eliminando las almohadillas y los nodos eléctricos dejando todo en su caja

Nota 1: Si se quiere grabar la pantalla de la computadora lo mejor es que se haga por usuario.

Nota 2: El comando de Python funciona con las coordenadas con la siguiente barra de inicio:



## 14.7. Anexo 7: Evidencias de la experimentación de campo.

La experimentación de campo consistió en conseguir datos a partir de personas voluntarias a ser usuarios de prueba dentro de este rubro, ofreciéndose a probarse la diadema y recopilar datos de su actividad cerebral para con ellos poner a prueba los modelos de machine learning mencionados.

Los participantes en el experimento cedieron los permisos indicados en la tabla 14.7.1 donde incluye también el número de participante al que se hace referencia en los resultados del experimento (cabe aclarar que como el investigador también aportó datos, éste es el participante número uno). Antes que nada personalmente he de agradecer a todos y cada uno de los participantes involucrados en la experimentación pues aportaron en una gran medida al desarrollo de esta investigación.

Tabla 14.7.1. Usuarios voluntarios en la experimentación de campo junto a los términos cedidos en el acuerdo.

#	Nombre	Uso del nombre	Grabación del usuario	Uso de las evidencias
2	José Luis Espinoza Sánchez	Sí	Sí	Sí
3	María Consuelo Sánchez Díaz	Sí	Sí	Sí
4	José Luis Espinoza González	Sí	Sí	Sí
5	Román Guadalupe de León Vázquez	Sí	Sí	Sí
6	Hiram Efraín Orocio García	Sí	Sí	Sí
7	Andrea Melissa Almeida Ortega	Sí	Sí	Sí
8	Óscar Alonso Flores Fernández	Sí	Sí	Sí
9	Edith Berenice Orocio García	Sí	Sí	Sí
10	Andrea de Santiago Legaspi	Sí	Sí	Sí
11	Montserrat Alejandra Ulloa Rivera	Sí	Sí	Sí
12	Juan Abraham Ortiz Salas	Sí	Sí	Sí
13	Valeria Macías Soto	Sí	Sí	Sí
14	Andrea Juliett Ordaz de Vierna	Sí	Sí	Sí
15	Gustavo Bejamín II Pedraza Morales	Sí	Sí	Sí
16	Alejandro Ramos Herrera	Sí	Sí	Sí
17	Miguel Ángel Meza de Luna	Sí	Sí	Sí
18	Fernando Ulloa	Sí	Sí	Sí
19	Andrés Eloy Escobedo Esparza	Sí	Sí	Sí
20	Jorge Luis Villalobos Araiza	Sí	Sí	Sí
21	Diego Romo Cruz	Sí	Sí	Sí
22	Raúl	Sí	Sí	Sí
23	Alejandro Padilla Díaz	Sí	Sí	Sí
24	Nancy Yissel Cuéllar Valdivia	Sí	Sí	Sí
25	Daniela del Carmen Fuentes Acata	Sí	Sí	Sí
26	Isabel Cristina Durón Esparza	Sí	Sí	Sí
27	Yolanda Muñoz Gaytán	Sí	Sí	Sí

Cuando se realizó la experimentación de campo se documentó todo en video de la misma forma en la que los datos obtenidos están guardados en archivos de valores separados por comas evidenciado en la figura 14.7.1.

joul1	●	13/02/2023 04:43 p. m.	Carpeta de archivos	Andy1	●	14/02/2023 11:52 a. m.	Carpeta de archivos
joul2	●	13/02/2023 04:47 p. m.	Carpeta de archivos	Andy2	●	14/02/2023 11:56 a. m.	Carpeta de archivos
pep1	●	13/02/2023 05:08 p. m.	Carpeta de archivos	Montse1	●	14/02/2023 12:06 p. m.	Carpeta de archivos
pep2	●	13/02/2023 05:18 p. m.	Carpeta de archivos	Montse2	●	14/02/2023 12:09 p. m.	Carpeta de archivos
mom1	●	13/02/2023 08:52 p. m.	Carpeta de archivos	Abraham1	●	14/02/2023 12:19 p. m.	Carpeta de archivos
mom2	●	13/02/2023 08:55 p. m.	Carpeta de archivos	Abraham2	●	14/02/2023 12:25 p. m.	Carpeta de archivos
dad1	●	13/02/2023 09:04 p. m.	Carpeta de archivos	Valeria1	●	14/02/2023 12:34 p. m.	Carpeta de archivos
dad2	●	13/02/2023 09:08 p. m.	Carpeta de archivos	Valeria2	●	14/02/2023 12:37 p. m.	Carpeta de archivos
Roman1	●	14/02/2023 08:09 a. m.	Carpeta de archivos	Juliett1	●	14/02/2023 12:43 p. m.	Carpeta de archivos
Roman2	●	14/02/2023 08:13 a. m.	Carpeta de archivos	Juliett2	●	14/02/2023 12:46 p. m.	Carpeta de archivos
Hiram1	●	14/02/2023 08:23 a. m.	Carpeta de archivos	Gus1	●	14/02/2023 01:39 p. m.	Carpeta de archivos
Hiram2	●	14/02/2023 08:27 a. m.	Carpeta de archivos	Gus2	●	14/02/2023 01:45 p. m.	Carpeta de archivos
Meli1	●	14/02/2023 08:34 a. m.	Carpeta de archivos	Alex1	●	15/02/2023 10:29 a. m.	Carpeta de archivos
Meli2	●	14/02/2023 08:37 a. m.	Carpeta de archivos	Alex2	●	15/02/2023 10:31 a. m.	Carpeta de archivos
Oscar1	●	14/02/2023 08:45 a. m.	Carpeta de archivos	Meza1	●	15/02/2023 10:38 a. m.	Carpeta de archivos
Oscar2	●	14/02/2023 08:50 a. m.	Carpeta de archivos	Meza2	●	15/02/2023 10:41 a. m.	Carpeta de archivos
Bere1	●	14/02/2023 11:13 a. m.	Carpeta de archivos	Fer1	●	15/02/2023 10:49 a. m.	Carpeta de archivos
Bere2	●	14/02/2023 11:18 a. m.	Carpeta de archivos	Fer2	●	15/02/2023 10:52 a. m.	Carpeta de archivos

Eloy1	●	15/02/2023 01:17 p. m.	Carpeta de archivos
Eloy2	●	15/02/2023 01:19 p. m.	Carpeta de archivos
Jorge2do1	●	16/02/2023 10:29 a. m.	Carpeta de archivos
Jorge2do2	●	16/02/2023 10:36 a. m.	Carpeta de archivos
Diego2do1	●	16/02/2023 10:46 a. m.	Carpeta de archivos
Diego2do2	●	16/02/2023 10:48 a. m.	Carpeta de archivos
Raul2do1	●	16/02/2023 10:58 a. m.	Carpeta de archivos
Raul2do2	●	16/02/2023 11:00 a. m.	Carpeta de archivos
Padilla1	●	16/02/2023 11:47 a. m.	Carpeta de archivos
Padilla2	●	16/02/2023 11:50 a. m.	Carpeta de archivos
Yissel1	●	17/02/2023 09:25 a. m.	Carpeta de archivos
Yissel2	●	17/02/2023 09:28 a. m.	Carpeta de archivos
Daniela1	●	25/02/2023 10:09 a. m.	Carpeta de archivos
Daniela2	●	25/02/2023 10:12 a. m.	Carpeta de archivos
Isabela1	●	25/02/2023 10:24 a. m.	Carpeta de archivos
Isabela2	●	25/02/2023 10:27 a. m.	Carpeta de archivos
Yolanda1	●	25/02/2023 10:36 a. m.	Carpeta de archivos
Yolanda2	●	25/02/2023 10:38 a. m.	Carpeta de archivos

Figura 14.7.1. Evidencia de los conjuntos de datos recopilados en el sistema de archivos.