

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

APLICACIÓN DE TÉCNICAS DE APRENDIZAJE DE MÁQUINA PARA FACILITAR LA COMUNICACIÓN EN PACIENTES CON NEUROPATÍA PERIFÉRICA UTILIZANDO SEÑALES EEG Y VISIÓN COMPUTACIONAL

DESARROLLO DE UN MODELO DE CLASIFICACIÓN SUPERVISADO PARA SEÑALES EEG QUE PERMITA LA COMUNICACIÓN EN PACIENTES CON NEUROPATÍA PERIFÉRICA.

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

ANDRÉS WLADIMIR GARCÍA CUVI

andres.garcia02@epn.edu.ec

DIRECTOR: ING. HENRY PATRICIO PAZ ARIAS Msc.

henry.paz@epn.edu.ec

DMQ, septiembre 2022

CERTIFICACIONES

Yo, ANDRÉS WLADIMIR GARCÍA CUVI declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



ANDRÉS WLADIMIR GARCÍA CUVI

Certifico que el presente trabajo de integración curricular fue desarrollado por ANDRÉS WLADIMIR GARCÍA CUVI, bajo mi supervisión.



ING. HENRY PATRICIO PAZ ARIAS Msc
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ANDRÉS WLADIMIR GARCÍA CUVI

ING. HENRY PATRICIO PAZ ARIAS Msc.

DEDICATORIA

El presente trabajo integrador está dedicado a mi familia quienes siempre me apoyaron, especialmente a mi tío Carlos que es como un padre para mí y ahora sufre de las patologías mencionadas en esta investigación, por lo que me inspiró a encontrar una solución para ayudarlo. Y así aportar con un granito de arena a la comunidad científica que busca ayudar a personas con discapacidades.

AGRADECIMIENTOS

Agradezco a Dios por permitir que cumpla con los objetivos que me he propuesto a lo largo de mi vida. Doy gracias a mi familia porque me han motivado a seguir mis sueños. También, a mi director de proyecto Msc. Henry Paz Arias, quien me guió desde el primer momento que mencioné este tema para ayudar a mi tío. A mis compañeros que llegaron a ser como parte de mi familia, con quienes compartí los momentos más difíciles de la vida universitaria.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTOS	IV
ÍNDICE DE CONTENIDO	VI
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	3
1.4.1 Tecnologías de asistencias	3
1.4.2 Interfaz cerebro - computador	4
1.4.3 SSVEP	9
2 METODOLOGÍA	10
2.1 Metodología Action Research	10
2.2 Diseños experimentales	11
2.2.1 Equipo de adquisición de EEG	12
2.2.2 Estructura de la interfaz gráfica	13
2.2.3 Experimento de la grabación de datos para el entrenamiento	18
2.2.4 Introducción a MNE-Python	19
2.2.5 Procesamiento de las señales	21
2.2.6 Clasificación	23
2.2.7 Entrenamiento del modelo	24
2.2.8 Inferencia del modelo	25
3 PRUEBAS, RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	27
3.1 Pruebas	27
3.1.1 Entrenamiento	27
3.1.2 Validación	28

3.1.3 Inferencia	28
3.2 Resultados	28
3.2.1 Entrenamiento y validación	28
3.2.2 Inferencia	31
3.3 Conclusiones	32
3.4 Recomendaciones	32
4 REFERENCIAS BIBLIOGRÁFICAS	34
5 ANEXOS	I
ANEXO I	II
ANEXO II	III
ANEXO III	IV
ANEXO IV	V
ANEXO V	VI
ANEXO VI	VI
ANEXO VII	VII
ANEXO VIII	VIII

RESUMEN

Las neuropatías periféricas son afecciones del sistema nervioso periférico que causa sensación de adormecimiento, dolor, hipersensibilidad, etc. Esto causa que los miembros superiores e inferiores sufran ciertos grados de atrofia muscular. Esto deja al paciente casi completamente paralizado. Además, en ciertos casos puede afectar el lóbulo frontal donde se considera el control del lenguaje expresivo. Para facilitar la comunicación en pacientes con este tipo de afecciones motoras y del habla, existen varias tecnologías de asistencia siendo una de las más prometedoras la interfaz cerebro – computador (BCI) basado en los electroencefalogramas (EEG), la cual permite una interacción directa entre la actividad cerebral y un dispositivo electrónico. Por lo tanto, en este trabajo integrador se aplicó técnicas de aprendizaje de máquina para crear un modelo de clasificación de señales EEG que se basan en potenciales evocados visuales de estado estacionario (SSVEP) que corresponden a señales de las respuestas a estímulos visuales emitidos a cierta frecuencia, logrando así tener un máximo de cinco patrones de comunicación para los pacientes. El trabajo consiste en la evaluación e implementación del modelo, donde el proceso de entrenamiento y validación del modelo se basa en métricas relacionadas con lo que dicta el estado del arte. Dando como resultado un modelo capaz de clasificar los estímulos visuales con frecuencias 13, 15, 17 y 19 Hz. Finalmente, inferir el modelo con cuatro comandos de comunicación básicos para cualquier persona que desee comunicarse.

PALABRAS CLAVE: Aprendizaje de máquina, Interfaz Cerebro - Computador (BCI), electroencefalograma (EEG), potenciales evocados visuales de estado estacionario (SSVEP), neuropatía periférica.

ABSTRACT

Peripheral neuropathies are conditions of the peripheral nervous system that cause a sensation of numbness, pain, hypersensitivity, etc. This causes the upper and lower limbs to suffer certain degrees of muscular atrophy. This leaves the patient almost completely paralyzed. In addition, in certain cases it can affect the frontal lobe where the control of expressive language is considered. To facilitate communication in patients with this type of motor and speech disorders, there are several assistive technologies, one of the most promising being the brain-computer interface (BCI) based on electroencephalograms (EEG), which allows direct interaction between brain activity and an electronic device. Therefore, in this integrative work, machine learning techniques were applied to create a model for classifying EEG signals based on steady-state visual evoked potentials (SSVEP) that correspond to signals from responses to visual stimuli emitted at a certain time. frequency, thus achieving a maximum of five communication patterns for patients. The work consists of the evaluation and implementation of the model, where the process of training and validating the model is based on metrics related to what the state of the art dictates. Resulting in a model capable of classifying visual stimuli with frequencies of 13, 15, 17 and 19 Hz. Finally, inferring the model with four basic communication commands for anyone who wishes to communicate.

KEYWORDS: Machine learning, Brain-Computer Interface (BCI), electroencephalogram (EEG), steady-state visual evoked potentials (SSVEP), peripheral neuropathy.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Las neuropatías periféricas son afecciones procedentes del sistema nervioso periférico conformado por ganglios, raíces, plexos, fibras nerviosas, etc. Esta patología de naturaleza degenerativa o inflamatoria se manifiesta clínicamente debido a la sensación de adormecimiento, quemadura, dolor, hipersensibilidad, sensación de hormigueo, hipotensión arterial postural, frialdad, anhidrosis, entre otros signos y síntomas de con distinta fisiopatología, evolución y pronóstico [1]. En este trabajo integrador se considera para su estudio a la Neuropatía Motora Multifocal (NMM), caracterizada por debilidad progresiva y asimétrica atrofia sin anomalías sensoriales, una presentación similar a la de la enfermedad de la neurona motora [2]. Esto causa que los miembros superiores e inferiores sufran ciertos grados variables de atrofia muscular [3]. Esto puede dejar al paciente casi completamente paralizado y, en el peor de los casos, solo puede mover la cabeza y los ojos. En algunos casos, puede afectar el área de Broca, que se localizada en el lóbulo frontal izquierdo del cerebro y se considera que es el núcleo que rige el lenguaje expresivo.

La aplicación de técnicas de interfaz cerebro-computador (BCI) ha permitido desarrollar soluciones para pacientes con grandes problemas médicos. Entre las diferentes soluciones que se presentan en el campo de estudio basados en electroencefalografía (EEG) no invasivo existe uno en específico que trata de la respuesta en estado estacionario de los cambios en el EEG causados por eventos como estímulos visuales aplicado al sujeto a través de una pantalla de computadora y se denomina potencial evocado visualmente en estado estacionario (SSVEP). Esta característica de la señal EEG se puede utilizar para formar una base de entrada a los dispositivos de asistencia para pacientes que no tengan la posibilidad de moverse o hablar y como resultado, su calidad de vida puede mejorar, así como para dispositivos que mejoran el rendimiento para sujetos sanos.

En este componente se utilizará la aplicación de técnicas de Aprendizaje de Máquina que pueda clasificar las señales EEG obtenidas de un paciente que ha recibido estímulos vi-

suales desde una interfaz gráfica. Para conseguir las señales necesarias para entrenar y validar el modelo de clasificación se utilizarán bases de datos públicas. Para ayudar con la comunicación de un paciente, es necesario que el modelo clasifique hasta 5 estímulos visuales, esto permitirá que el paciente pueda tener un máximo de 5 respuestas para su comunicación en un ambiente controlado.

1.1 OBJETIVO GENERAL

Aplicar técnicas de Aprendizaje de Máquina para facilitar la comunicación en pacientes con Neuropatía periférica utilizando señales EEG.

1.2 OBJETIVOS ESPECÍFICOS

- Revisar la literatura para determinar el estado del arte de la aplicación de técnicas de Aprendizaje de Máquina en pacientes con neuropatía periférica que servirá como insumo para la investigación.
- Analizar y limpiar la base de datos extraída de BCI que permitirá entrenar un modelo supervisado.
- Implementar y Evaluar un modelo de clasificación de señales EEG con aprendizaje supervisado para la clasificación de hasta 5 patrones de comunicación.

1.3 ALCANCE

El alcance de este componente es ayudar a pacientes con Neuropatía periférica a comunicarse con hasta 5 respuestas por medio de una interfaz cerebro-computador (BCI), la cual permite una comunicación directa entre un cerebro y un computador traduciendo la actividad cerebral en comandos computacionales, lo que representa una solución parcial para una persona que no se puede mover, ni hablar. Para la extracción de señales, se envía estímulos visuales al paciente conocidos como Steady state visually evoked potential (SSVEP) que se caracterizan por utilizar estímulos visuales a diferentes frecuencias sensibles a los procesos dinámicos de la parte occipital de la corteza cerebral [4]. Los sistemas BCI que se basan en SSVEP tienen la ventaja sobre otros sistemas BCI porque tienen una mejor

relación señal-ruido (SNR) y una tasa de transferencia de información (ITR) más rápida. Además, no requiere un entrenamiento intensivo y requiere menos canales de EEG para el desarrollo de aplicaciones [5]. Para esto es necesario crear un modelo de aprendizaje de máquina el cual será capaz de clasificar las señales que han sido extraídas como objetivas o no objetivas, esto quiere decir que una señal objetiva será una onda con una frecuencia mayor a lo normal que representa a un estímulo. Una vez se pueda clasificar las señales se podrá crear una interfaz de comunicación donde se mostrarán hasta 5 tipos de respuestas simples que se necesitan para una comunicación básica, estos elementos que serán mostrados tendrán una secuencia de parpadeos con un valor diferente de frecuencia para así poder diferenciarlos entre sí. Con esto se puede identificar la respuesta que el paciente estaría intentando comunicar. La metodología que será utilizada para el desarrollo de este componente será Action-Research (AC), la que es conocida por ser dualista entre la investigación y la acción [6], es importante reconocer que esta metodología es uno de diferentes tipos de indagación-acción, quiere decir, que este es un término genérico para cualquier proceso que sigue un ciclo en el que uno mejora la práctica al oscilar sistemáticamente entre actuar en el campo de la práctica e indagar en él. Primer se planifica, luego se implementa, después se describe y por último se evalúa un cambio de mejora en la propia práctica, donde se aprende más sobre la práctica y la investigación de la acción en el proceso [7]. La razón por la que se escogió esta metodología es porque se crean nuevos conocimientos por medio de la práctica y al ser de manera cíclica se puede agregar mejoras al proyecto en cada iteración y de esta forma acercarse más a la solución del problema que se quiere resolver.

1.4 MARCO TEÓRICO

1.4.1 Tecnologías de asistencias

Las discapacidades físicas y mentales, pueden provocar limitaciones de movimientos. La mayoría de las personas afectadas por este tipo de discapacidad requieren la ayuda de otra persona o tecnología para comunicarse, moverse, aprender, oír o ver. Las tecnologías que ayudan a las personas con discapacidades que les permite realizar actividades de la vida diaria se conocen como dispositivos de tecnologías de asistencia [8]. Las tecnologías de asistencia permiten a las personas vivir vidas saludables, productivas, independientes

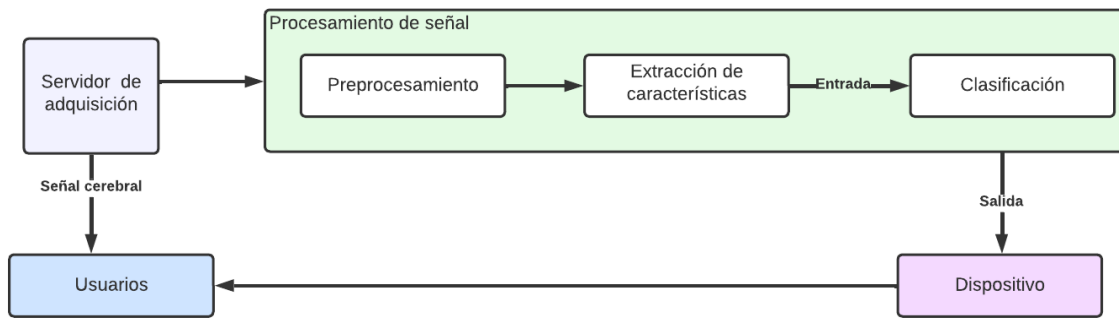


Figura 1.1: BCI, Diagrama de etapas del sistema.

y dignas, pero la mayoría de las personas que las necesitan no tienen acceso a ellas. Esta revisión rápida examina ejemplo de la literatura existente sobre la disponibilidad de las tecnologías de asistencia y los esfuerzos para hacer de estas tecnologías sean más asequibles y accesibles en los países en desarrollo [8]. Un tipo de tecnología de asistencia que se ha mostrado como una tecnología prometedora son las interfaces cerebro - computador, la cual será detallada más adelante. Esta tecnología fue adoptada en este estudio para ayudar a personas que no puedan comunicarse y será estudiada en las siguientes secciones.

1.4.2 Interfaz cerebro - computador

La interfaz cerebro - computador, por sus siglas en inglés BCI (Brain-computer interfaces), es una tecnología que permite la interacción directa entre la actividad cerebral del usuario y dispositivos específicos que no dependen de los músculos y nervios periféricos. Las BCI traducen la intención, pensamiento y las respuestas a los estímulos externos en órdenes de control de entrada permitiendo al usuario manejar dispositivos o aplicaciones externas.

En la Figura 1.1, se presentan las diferentes etapas que implican al usuario, adquisición de la señal cerebral. Como resultado al final de la etapa de procesamiento de señal, se debe generar una señal de salida que corresponde a un intención o comando del usuario para control de BCI.

1.4.2.1 Regiones del cerebro

Como se indica en [9], el encéfalo está compuesto de de tres partes principales como se observan en la Figura 1.2:

1. **El cerebelo:** Se encuentra en la parte posterior e inferior del encéfalo. Entre sus funciones se encuentran la coordinación de la postura y el equilibrio.
2. **El tronco encefálico:** Se encuentra en la parte inferior del encéfalo, y entre sus funciones se encuentran funciones que no dependen del control de la persona como lo son la deglución, digestión, el ciclo circadiano, la frecuencia cardiaca, etc.
3. **El cerebro:** Es la parte mas grande de todas. Consta de dos hemisferios y es la parte encargada de realizar las funciones mas importantes dentro del cuerpo humano, ya que entre otras funciones, esta encargado del control de los sentidos, las emociones, memoria, etc. Los hemisferios del cerebro están a su vez compuestos por cuatro lóbulos que se describen a continuación en la Figura 1.3.
 - ❑ **Lóbulo frontal:** Es el lóbulo más grande, muy estudiado en el campo de la psicología ya que es el encargado de aspectos como la inteligencia, la personalidad, el juicio, la razón, etc.
 - ❑ **Lóbulo parietal:** Este lóbulo se relaciona mucho con aspectos como el procesamiento de información y señales externas, dado que entre sus funciones esta la interpretación del lenguaje, percepción espacial, y la recepción e interpretación de los sentidos.
 - ❑ **Lóbulo occipital:** La principal función del lóbulo occipital yace en el procesamiento de la información visual como son luces, colores y movimiento.
 - ❑ **Lóbulo temporal:** Este lóbulo se relaciona mucho con los otros lóbulos ya que permite entre otras funciones, relacionar estímulos auditivos con palabras. El área de Wernicke es la principal encargada de producir esta comprensión.

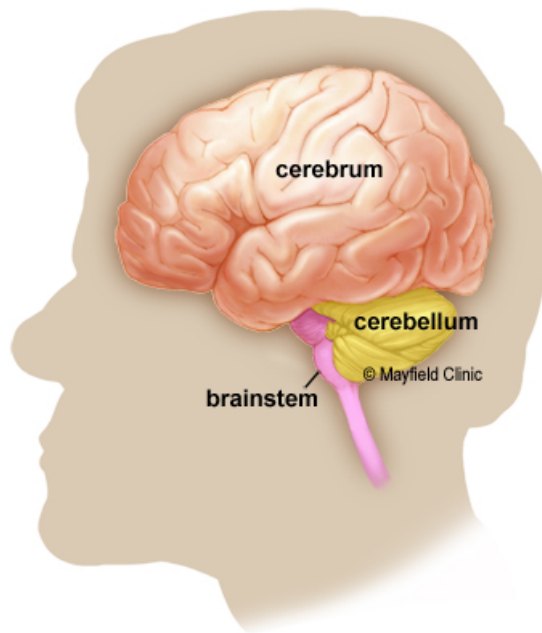


Figura 1.2: Regiones del encéfalo. Fuente [9]

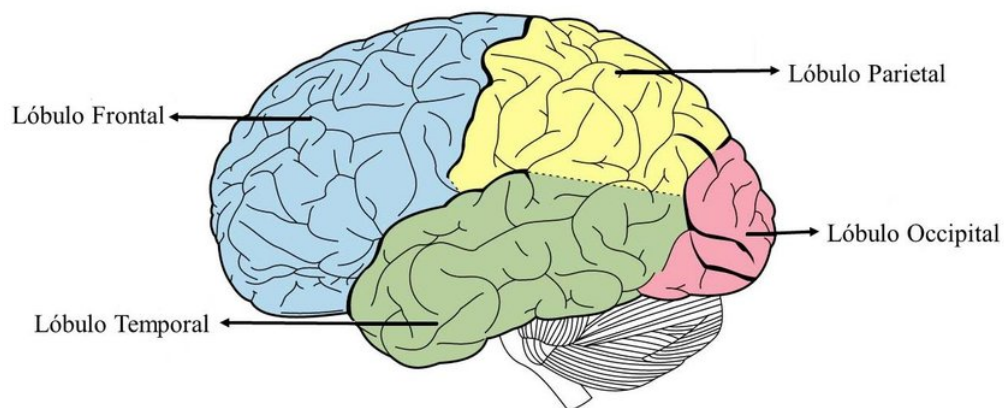


Figura 1.3: Ubicación de los lóbulos del cerebro. Fuente [10]

1.4.2.2 Ondas cerebrales

En el sistema nervioso hay más de 100 mil millones de células neuronales. Recibe información de todos los órganos y nervios sensoriales, los que procesa y establece respuestas a ser ejecutadas por el cuerpo [11]. El funcionamiento del cerebro se manifiesta a través de una actividad eléctrica. Las células neuronales están especializadas en la recepción, integración y transmisión de impulsos eléctricos.

Las ondas cerebrales son registros eléctricos de la superficie del cerebro o de la superficie exterior de la cabeza donde se presentan actividades eléctricas continuas dentro de varias regiones subyacentes de la corteza. Tanto la intensidad como los patrones de estas

actividades eléctricas son determinado en gran medida por el nivel general de excitación regional [12]. Cuando se mantiene un patrón regular comúnmente se define como una de las siguientes formas de onda:

- ❑ Ondas alfa: 8 - 13 Hz, 50mV; típicamente ocurre durante la vigilia tranquila o un estado de reposo (principalmente registrado en las regiones corticales occipitales).
- ❑ Ondas beta: 14–25 Hz; patrones de activación del SNC que típicamente ocurren cuando una persona está bajo tensión (principalmente registrada desde parietal y cortical frontal) regiones).
- ❑ Ondas theta: 4–7 Hz; normalmente provocada durante el estrés emocional, la decepción, frustración (asociada con los lóbulos parietal y temporal).
- ❑ Ondas delta: menos de 3,5 Hz; estos ocurren típicamente durante el sueño profundo, en infancia, y/o en casos de enfermedad cerebral orgánica grave (en toda la corteza).

1.4.2.3 Técnicas de adquisición de ondas cerebrales

Existen varias técnicas para el registro cerebral o la obtención de imágenes del cerebrales. Las más conocidas son la Tomografía por Emisión de Positrones (Positron Emission Tomography o siglas en inglés PET) y la Espectroscopía de Infrarrojo Cercano (Near Infrared Spectroscopy o NIRS), el magnocencefalograma (magnetoencephalogram o MEG), la imagen por resonancia magnética funcional (functional magnetic resonance imaging o fMRI) y la electroencefalografía (electroencephalography o EEG) [13]. La EEG es una de las técnicas más utilizadas y la cual es descrita a continuación.

EEG

La EEG es una técnica en la cual se mide la actividad eléctrica del cerebro mediante la aplicación de electrodos en el cuero cabelludo de un individuo. La EEG utiliza entre 64 y 256 de electrodos ubicados en distintas posiciones. En la Figura 1.4 se muestra el esquema de funcionamiento de la EEG así como un ejemplo de la lectura obtenida en esta técnica. La EEG detecta la actividad eléctrica Sobre el área cortical de la cabeza, reflejando tanto actividades espontáneas como aquellas relacionadas a estímulos.

Entre las principales ventajas de la EEG se encuentran la portabilidad, resolución, facilidad de uso y bajo costo. Por contra parte, esta técnica presenta ciertas limitaciones como es el

hecho de medir solo potenciales postsinápticos en la superficie del cuero cabelludo, y no es sensible a respuestas mas profundas del cerebro. [13], [14]

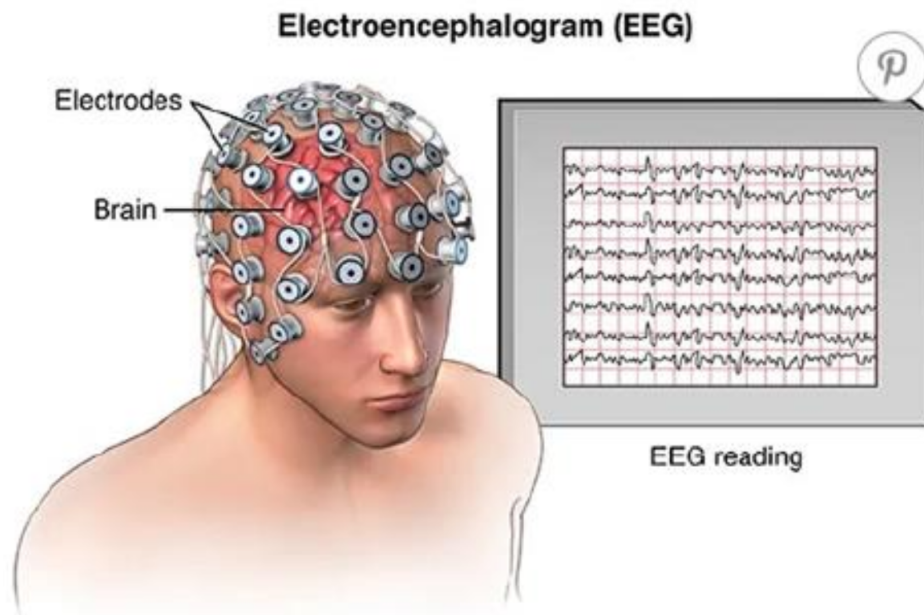


Figura 1.4: Funcionamiento de la EEG con los resultados de su lectura. Fuente: [15]

1.4.2.4 BCI BASADO EN EEG

Debido a la practicidad de las EEG, los dispositivos para realizar esta técnica son unos de los más utilizados en diversas áreas donde se aplican las BCIs. Una ventaja del uso de EEGs para sistemas BCI es su alta resolución temporal. Cuando se usan EEGs con sistemas BCI, la SNR inferior es considerada ya que factores como el ruido ambiental pueden contaminar las señales de la EEG.

Un sistema BCI basado en EEG permite observar la respuesta del cerebro a diversos estímulos, y estos se pueden separar en dos grupos según el potencial del cerebro usado: señales endógenas (provenientes del mismo sujeto) o señales exógenas (correspondientes a estímulos externos). De este último tipo existen varios tipos de potenciales que son evocados por dichas señales como son: El potencial evocado P300, el ritmo sensoriomotor (Sensory Motor Rhythm o SMR por sus siglas en inglés) o los potenciales evocados visuales de estado estacionario (Steady-State visual evoked potentials o SSVEPs por sus siglas en inglés) [14], [16]. De estos últimos potenciales se habla más a profundidad en la siguiente sección.

1.4.3 SSVEP

Los Potenciales evocados visuales de estado estacionario (Steady-State Visual Evoked Potentials o SSVEP por sus siglas en inglés) son señales que corresponden a las respuestas a estímulos visuales emitidos a una frecuencia específica. Cuando la retina es estimulada por estímulos de entre 3.5 Hz y 75 Hz, el cerebro emite actividad eléctrica a frecuencias que son múltiplos de la frecuencia del estímulo emitido[17]. Estas señales son más prominentes en el lóbulo occipital debido a que se encuentran relacionados con los estímulos de la visión. Para estos estímulos se deben tomar muchos aspectos como la fuente del estímulo que por lo general pueden ser producidos por LEDs, con un resultado más confiable que otras fuentes como son LCDs o CRTs, también es importante analizar el tiempo de exposición, la distancia, evidentemente la frecuencia, entre otros factores. Luego de realizar el procesamiento de la señal existen 3 fases importantes: el preprocesamiento, la extracción de características y la clasificación [14].

2 METODOLOGÍA

Cada vez, es más común que la metodología Action Research sea parte de la investigación educativa y de proyectos de desarrollo curricular. Por lo tanto, en este capítulo contiene todas las actividades para aplicar las técnicas de Aprendizaje de Máquina para modelos de clasificación, utilizando esta metodología.

2.1 METODOLOGÍA ACTION RESEARCH

Es una metodología que está basada en una relación amplia entre la experimentación y la generación de conocimiento. Este término fue mencionado por primera vez por Kurt Lewin en 1964 donde menciona que la investigación es el medio para lograr un cambio social [18]. Por otro lado, años después Rapoport [19] nos da una definición más consolidada acerca de esta metodología, y es quizá una de las más citadas acerca de este tema. Argumenta que la metodología Action-Research pretende contribuir tanto a las preocupaciones prácticas de las personas en una situación problemática inmediata y a los objetivos de las ciencias sociales a través del trabajo colaborativo dentro de marcos éticos mutuamente aceptables. Aunque la definición de Rapoport se centra en un objetivo y fue utilizada por varios años, luego Action-Research se fue considerada como un proceso cíclico de cinco fases:

1. Diagnóstico: Identificación o definición del problema.
2. Planificación de acción: Consideración alternativa de cursos de acción para solucionar el problema.
3. Realización de acción: Selección de un curso de acción.
4. Evaluación: Estudio de las consecuencias de una acción.
5. Especificación de aprendizaje: Identificación general de los hallazgos.

La infraestructura del sistema cliente y el investigador en acción mantienen y regulan conjuntamente algunas o todas las cinco fases como se muestra en la Figura 2.1

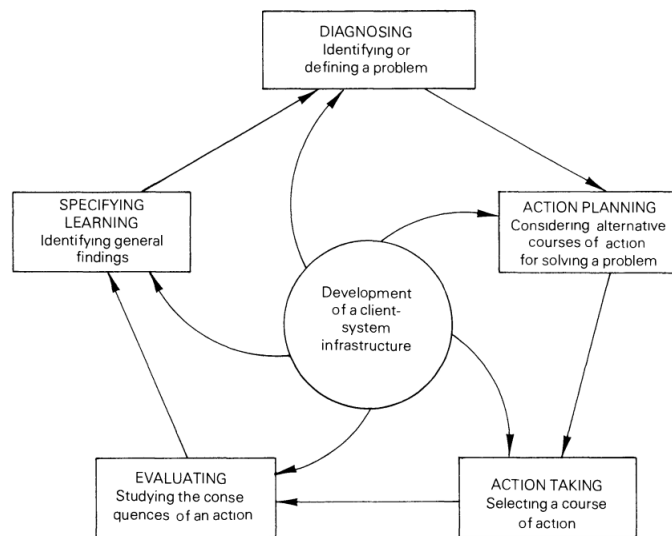


Figura 2.1: Proceso cíclico de Action-Research.

En [20] muestra una visión general acerca de esta metodología y cómo los métodos empíricos tradicionales de la ingeniería de Software deberían complementarse con Action-Research. Además, indica que esta metodología aplicada a la ingeniería de software o ciencias de la computación ayuda a proporcionar el máximo valor a las empresas colaboradoras y cómo ayuda a los investigadores a desarrollar sus carreras.

Con esta información acerca del enfoque empírico y el modelo cíclico se puede cumplir los objetivos propuestos en esta investigación. Para el presente proyecto integrador, se propone un sistema BCI basado en SSVEP, donde se explique la adquisición, procesamiento y clasificación de los datos. Cada ciclo buscará un mejor precisión del clasificador mediante la experimentación.

2.2 DISEÑOS EXPERIMENTALES

Los datos de SSVEP se puede adquirir con varias técnicas como se mencionó en la sección 1.4.2.3. Siendo, el EEG uno de los métodos de adquisición más comunes para acceder a las señales generadas por el cerebro y este método es el que se utiliza en este estudio, porque no es invasivo, con buena resolución temporal. Sin embargo, la resolución espacial de un EEG es limitada, ya que cada electrodo de canal se coloca aproximadamente sobre una gran cantidad de neuronas que se disparan. Además, las señales generadas por el cerebro

se distorsionan y filtran durante el paso por el cuero cabelludo y el cráneo. Para reconstruir o recuperar información útil de estos dispositivos se suelen utilizar métodos avanzados de procesamiento de señales y enfoques de aprendizaje automático. Para lograr cumplir los objetivos de esta investigación, se obtuvo una base de datos pública utilizada para el control de un robot utilizando SSVEP [21]. Se lo realizó con una base de datos existente porque la obtención del Hardware para la adquisición de señales EEG es elevado. Un ejemplo de esto se muestra en la proforma de un kit de adquisición de señales EEG en el Anexo I.

2.2.1 Equipo de adquisición de EEG

En esta sección, se presentará el hardware y software asociados a los experimentos realizados para la adquisición de las señales SSVEP que se encuentran en la base de datos pública.

2.2.1.1 Hardware

El dispositivo amplificador utilizado para registrar las señales EEG fue un gtec g.USBamp con una frecuencia de muestreo de 256 Hz. Este dispositivo de gama alta tiene una tecnología USB 2.0 de g.tec. Además, permite la adquisición de 16 canales de bioseñales de EEG con 24 bits. El amplificador USB permite la conexión directa con una PC o portátil con un conector USB común, sin necesidad de utilizar un dispositivo externo. También, permite ajustar la frecuencia de muestreo entre 64 Hz y 38 400 Hz [22]. Además, se utilizó un gorro EEG con el sistema internacional de posicionamiento de electrodos 10-20 como se muestra en la Figura 2.2. En la Figura 2.3, se muestra un ejemplo del kit para la obtención de señales EEG de la marca gtec. Además, se encuentra la proforma de este kit en el Anexo II.

2.2.1.2 Software

La biblioteca Lab Streaming Layer (LSL) se utilizó para la adquisición de señales EEG. Consiste en un conjunto de herramientas, incluidos programas de grabación, importadores de archivos y aplicaciones que hacen que los datos de varios hardware de adquisición estén disponibles en la red del laboratorio [23]. Dado que el gtec g.USBamp es compatible con las principales plataformas BCI de código abierto, se usó el código publicado por LSL en

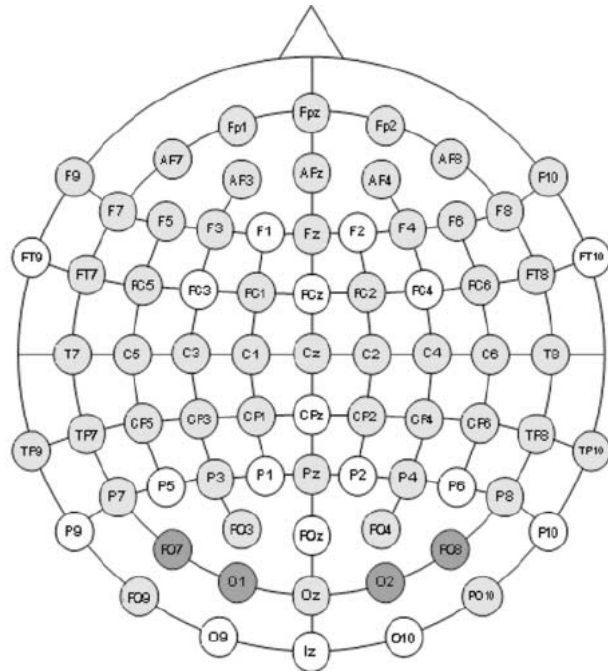


Figura 2.2: Mapa de posiciones de electrodos según el sistema extendido 10 – 20.



Figura 2.3: Amplificador de bioseñales gtec g.USBamp y gorro EEG.

GitHub para la obtención de las señales EEG [24].

2.2.2 Estructura de la interfaz gráfica

En esta sección se describirá el desarrollo, los detalles y la evaluación de la posición de los componentes y el tamaño de la ventana. Esto será utilizado para la generación de los estímulos visuales hacia el paciente.

2.2.2.1 Análisis del estimulador visual

Uno de los principales desafíos de los paradigmas basados BCI basados en SSVEP se relaciona con lograr una presentación precisa y confiable de los estímulos visuales repetitivos. Para garantizar que los estímulos visuales sean fiables, es necesario que el estímulo visual se presente al sujeto con una frecuencia entre 3.5 Hz y 75 Hz según la literatura [25]. Sin embargo, debido al teorema de muestreo de Nyquist-Shannon, la frecuencia más alta que se puede mostrar es igual a la mitad de la frecuencia de actualización máxima del monitor.

Este teorema también se lo conoce como teorema de muestreo [26][27], es un teorema fundamental en el procesamiento de señales biomédicas. Este teorema indica que es posible realizar la reconstrucción de una señal continua, periódica y limitada en banda a partir de una serie de muestras. Para ello, la señal debe cumplir la siguiente condición Dada la señal analógica $x(t)$, donde la frecuencia máxima es $F_{max} = B$. Si la señal es muestreada a una tasa de muestreo $F_m > 2B$, la señal puede ser recuperada sin pérdidas a partir de alguna una función de interpolación Por ejemplo, en la siguiente señal mostrada en la Figura 2.4 con una frecuencia máxima $F_{max} = 3Hz$:

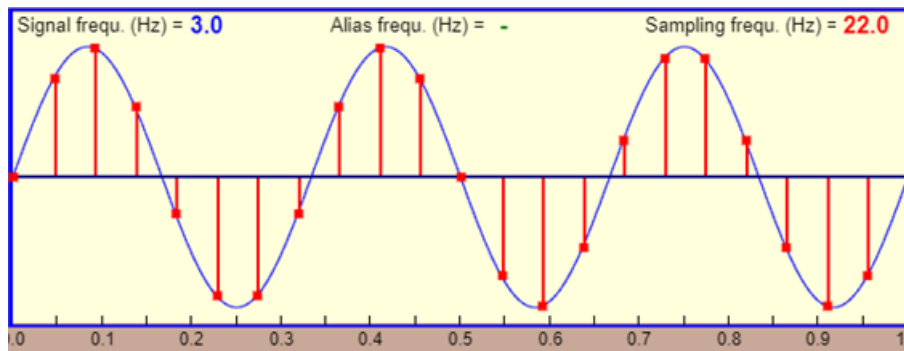


Figura 2.4: Señal sinusoidal con frecuencia máxima de 3Hz sobre la cual se realizará el muestreo.

Si se realiza un muestreo con una frecuencia menor a $F_{max} = 6Hz$, en este caso una frecuencia de muestreo $F_m = 4.1Hz$, la señal reconstruida difiere mucho de la señal original como se observa en la Figura 2.5:

Por el contrario, al realizar muestras con frecuencias mayores a 6Hz, la señal reconstruida toma una forma mucho más parecida a la original, en la Figura 2.6 se indica el muestreo con una frecuencia de 10.1Hz

Por tal razón, la mayoría de los estudios relacionados con BCI basados en SSVEP informan que las respuestas de los estímulo se encuentran alrededor de los 20 Hz [28].

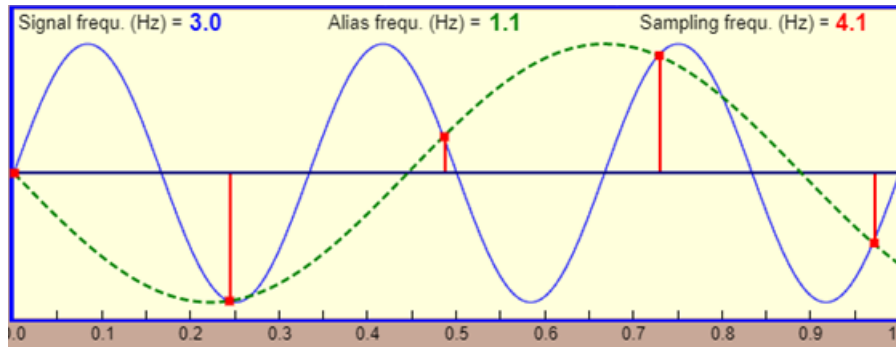


Figura 2.5: Señal sinusoidal realizando el muestreo a una frecuencia de 4.1Hz sobre la señal original. La señal generada por el muestreo difiere de la original debido a que la frecuencia es menor al doble de la frecuencia máxima de la señal original

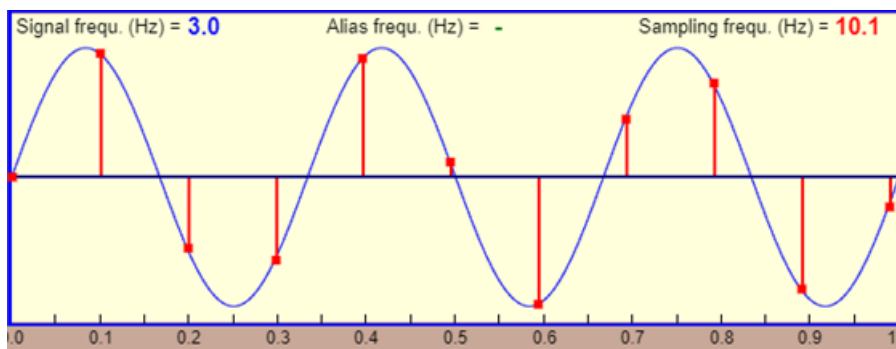


Figura 2.6: Señal sinusoidal realizando el muestreo a una frecuencia de 10.1Hz sobre la señal original. La señal generada por el muestreo puede reconstruir sin pérdidas a la señal original debido a que la frecuencia es mayor al doble de la frecuencia máxima de la señal original.

En [29] dice específicamente que para diseñar un sistema como este se rige por dos restricciones principales:

1. **Resolución de frecuencia:** La diferencia mínima en la frecuencia de parpadeo entre objetivos vecinos que el sujeto puede discriminar: se determina empíricamente en aproximadamente 0,2 Hz.
2. **Ancho de banda efectivo:** El ancho de banda de frecuencia en el que se puede observar efectivamente el SSVEP: se determina empíricamente en aproximadamente 6 a 24 Hz.

Una vez conocido lo mencionado anteriormente, se sugiere utilizar frecuencias de estimulación entre 10 Hz y 20 Hz, porque estas frecuencias evocan la respuesta de mayor amplitud y la mayor relación señal - ruido.

La relación señal-ruido (signal-to-noise ratio o SNR por sus siglas en inglés) se refiere a la proporción que existe entre la potencia promedio P_x de una señal $x(t)$ con respecto a la

potencia promedio P_r del ruido existente en dicha señal dentro de un tiempo T determinado [30]. Cuando la señal es determinística, P_x se define con la Ecuación 2.1

$$P_x = \frac{1}{T} \int_0^T x^2(t) dt \quad (2.1)$$

En caso se tratarse de un proceso estocástico estacionario, P_x se define por el valor de su función de correlación $R_x(\tau)$ definida en la Ecuación 2.2, donde $E[x(t)x(t + \tau)]$ indica el valor esperado.

$$R_x(\tau) \equiv E[x(t)x(t + \tau)] \quad (2.2)$$

Donde el valor P_x esta descrito por la Ecuación 2.3.

$$P_x = R_x(0) \quad (2.3)$$

El valor de P_r , dado que el ruido siempre tiene una descripción estocástica, también esta definido por su función de correlación en el origen, como se muestra en la Ecuación 2.4.

$$P_r = R_r(0) \quad (2.4)$$

Así, la SNR se expresa como en la ecuación 2.5

$$\text{SNR} = \frac{P_x}{P_r} . \quad (2.5)$$

Y en caso de requerirse una expresión en decibelios(dB), se emplea la ecuación 2.6.

$$\text{SNR(dB)} = 10 \log_{10} \frac{P_x}{P_r} . \quad (2.6)$$

Esto debería proporcionar la mejor calidad de datos para entrenar el clasificador de SSVEP. Para la adquisición de la base de datos pública utilizaron un experimento con cuatro frecuencias de estímulo: 13 Hz, 15 Hz, 17 Hz y 19 Hz. Además, se utilizó un monitor de 14 pulgadas LCD con una resolución de 1366 por 768 píxeles, con una frecuencia de refresco de la imagen de 60 fps. Por lo que al presentar los estímulos visuales en la interfaz gráfica el paciente los pueda observar con la frecuencia correcta sin violar el teorema de Nyquist

evitando así efectos de aliasing, que no son más que la alteración de la percepción de un determinado movimiento a través de cualquier óptica. Es decir, es causado por un muestreo incorrecto de la señal que se está digitalizando, de modo que aparecen imperfecciones de la misma.

2.2.2.2 Desarrollo del estimulador visual

Teniendo en cuenta el análisis de la sección anterior, se realizó una interfaz gráfica conocida como cruz, donde los componentes están distribuidos en la parte superior, inferior, derecha, izquierda y en el centro de la ventana. De estos componentes, cuatro son estimuladores visuales y uno es una imagen fija en el centro como se aprecia en la Figura 2.7.

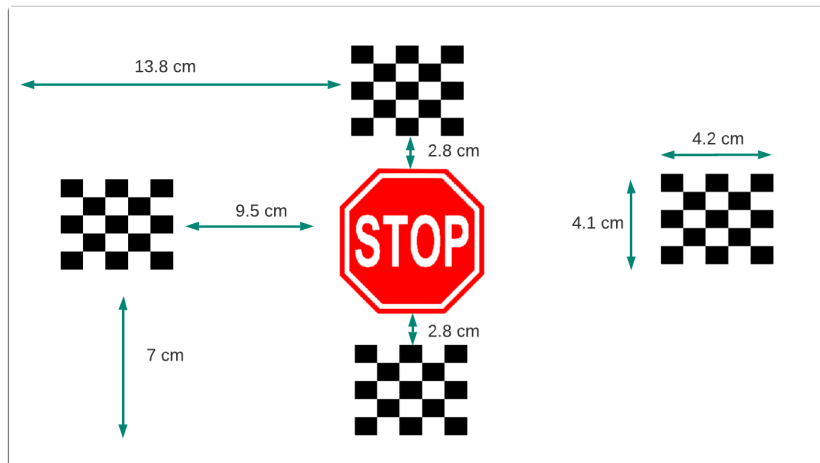


Figura 2.7: Interfaz en cruz representada con sus respectivas medidas entre objetivos y el tamaños del objetivo.

El sistema para producir estímulos visuales al paciente se implementó con la biblioteca de python PyGame, la cual proporciona una API para renderizar interfaces gráficas de usuario para el desarrollo de videojuegos. Los objetos que parpadean en la pantalla se realiza con un tablero similar al ajedrez pero de 5x5 como se muestra en la Figura 2.8. El parpadeo se consigue invirtiendo el color del tablero. El siguiente paso es verificar la frecuencia del parpadeo, ya que la animación del objeto no se proporciona en tiempo real en un sistema sin un núcleo en tiempo real. Para minimizar el parpadeo causado por un programador de procesos de sistemas, la animación debe ser ejecutada en un proceso propio con un controlador de frecuencia. El controlador de frecuencia observa la diferencia entre la frecuencia establecida y la observada y regula el tiempo de espera entre dos animaciones. De este modo,

se puede disminuir el error de inicialización del tiempo de espera y minimizar los errores de frecuencia causados por los cambios de carga de trabajo del sistema. El controlador de frecuencia se implementa con un controlador PID. En [21] afirma que al utilizar un controlador de frecuencia se puede observar una desviación de máximo 2 ms con una sola animación en el ordenador de prueba. También hay que tener en cuenta que esta desviación depende bastante de la carga de trabajo y la configuración del sistema informático.

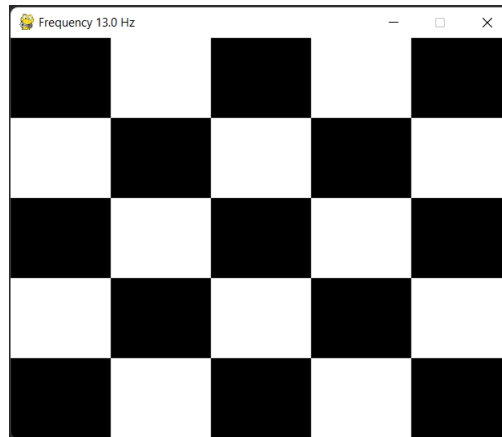


Figura 2.8: Estimulador visual representado como un tablero de 5x5.

2.2.3 Experimento de la grabación de datos para el entrenamiento

En esta sección se describirá cómo se obtuvo los datos para poder entrenar modelos de clasificación con Machine Learning. Como se mencionó anteriormente, los datos de señales EEG recopilados en una base de datos que se utilizó para este estudio pertenece a [21]. Por lo tanto, se va a mencionar el experimento que se llevó a cabo para obtener los datos.

2.2.3.1 Participantes

En este experimento participaron cuatro sujetos. No se especifica el género ni la edad. Sin embargo, para participar en dicho estudio se solicitó que los participantes debían:

- No presentar ningún tipo de déficit cognitivo o físico;
- Tener una visión normal o correcta a la normal;
- Tener como mínimo 18 años de edad;

- ❑ Ser capaces de llegar por sí mismos a las instalaciones de la Universidad donde se realizó el experimento;
- ❑ Estar disponibles durante dos horas consecutivas; y
- ❑ Tener el pelo lavado sin acondicionador y seco.

2.2.3.2 Procedimiento experimental

Los participantes se colocaron en una silla cómoda a 60 centímetros del monitor LCD donde se encontraba la interfaz gráfica presentada en la sección 2.2.2. El gorro EEG se colocó en la cabeza del participante y se aplicó gel conductor para reducir la impedancia electro-piel. Este gorro solo contó con los electrodos O1, O2, Oz dado que están ubicados en el lóbulo occipital del cerebro donde se producen los estímulos visuales. Cada experimento se realizó durante aproximadamente dos horas, desde la colocación del gorro hasta el final de todo el protocolo para cada participante. En cada parte se realizaron 10 intentos de 720 segundos, que consistieron en 10 segundos a un estado de reposo, 15 segundos realizando la tarea (es decir, mirando un estímulo), y así sucesivamente. Todos los participantes recibieron instrucciones de evitar el parpadeo de los ojos y los movimientos corporales mientras realizaban la tarea. Cada 10 segundos, el participante debe mirar a un objetivo visual diferente en la pantalla. La secuencia de objetivos siempre fue en el sentido de las agujas del reloj, donde el primer objetivo en la interfaz en cruz era el superior y el último era el centro, El participante fue orientado a cambiar de objetivo a través de una orden verbal. El intervalo entre intentos fue de un segundo. Utilizando la biblioteca descrita en la sección 2.2.1.2, se la implementó en un programa de python que registra los electrodos seleccionados. Los datos fueron almacenados en varios archivos de extensión *.mat donde al final del nombre del archivo se encuentra la frecuencia del estímulo visual que recibió como se muestra en la Figura 2.9.

2.2.4 Introducción a MNE-Python

MNE–Python es un subproyecto del paquete de software MNE que proporciona un conjunto de algoritmos para permitir a los usuarios un análisis completo de datos del procesamiento M/EEG [31]. Aunque existen varios otros paquetes como EEGLAB, FieldTrip, etc. Estos paquetes se implementan en MATLAB. MNE-Python, por otro lado, proporciona una forma

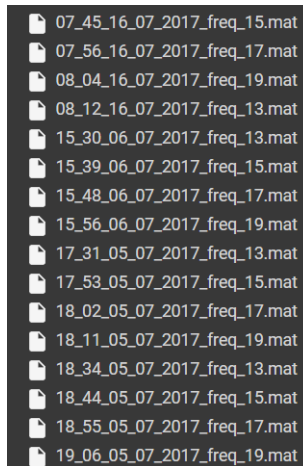


Figura 2.9: Base de datos de SSVEP.

más única y sencilla de análisis de datos M/EEG y también permite implementar paquetes de aprendizaje automático con pocas líneas de códigos. MNE-Python proporciona una gran cantidad de características adicionales como análisis de frecuencia de tiempo, análisis de componentes independientes, análisis de patrones multivariados (MVPA) o aprendizaje supervisado, etc. [32]. MNE se basa en algoritmos puros de Python con módulos de Python incorporados avanzados adicionales como Numpy [33] para almacenar y manipular de manera eficiente matrices numéricas N dimensionales, Scipy para procesamiento de señales y manipulaciones de matrices, Matplotlib [34] para gráficos 2-D, Scikit–learn [35] para tareas de decodificación donde se pueden implementar técnicas de Machine Learning y Python Data Analysis Library (Pandas) para el procesamiento de datos.

2.2.4.1 Flujo de trabajo estándar de MNE-Python

El análisis de datos M/EEG con MNE–Python involucra principalmente tres tipos de contenedores de datos conocidos como objetos Raw, Epochs y Evoked. Los datos sin procesar obtenidos del sistema de adquisición de datos se segmentan en épocas o ensayos, que son segmentos de datos que corresponden a repeticiones de estímulos. Estos segmentos se promedian para formar datos evocados. MNE-Python ha sido diseñado de tal manera que estos pasos los realizan fácilmente los objetos MNE que facilitan la transformación de datos. Los datos sin procesar continuos se almacenan en instancias de la clase Raw y MNE–Python admite la lectura de datos sin procesar de varios formatos de archivo. Los experimentos típicos de EEG involucran presentaciones de estímulos que se repiten en varios intervalos. La aparición de cada respuesta se puede utilizar para definir una época

que captura las señales cerebrales que preceden o siguen a la respuesta. Las épocas de diferentes condiciones experimentales de un usuario se almacenan en una clase Epochs en MNE. El objeto Epochs tiene varios parámetros para preprocesamiento, corrección de línea de base, diezmado temporal, etc. Las épocas se pueden promediar para formar datos evocados y estos datos promediados se almacenan en la clase Evoked. Se puede seguir más a fondo el uso de esta herramienta en su documentación [36].

2.2.5 Procesamiento de las señales

El procesamiento de las señales EEG ayuda a mejorar la precisión de los modelos de clasificación. Estos datos fueron procesados en el entorno virtual de Google Colab en el lenguaje python.

Se seleccionó una muestra del conjunto de datos, específicamente los datos con la frecuencia de 15 Hz, para poder graficar cómo ser muestra los datos recopilados (Figura 2.10).

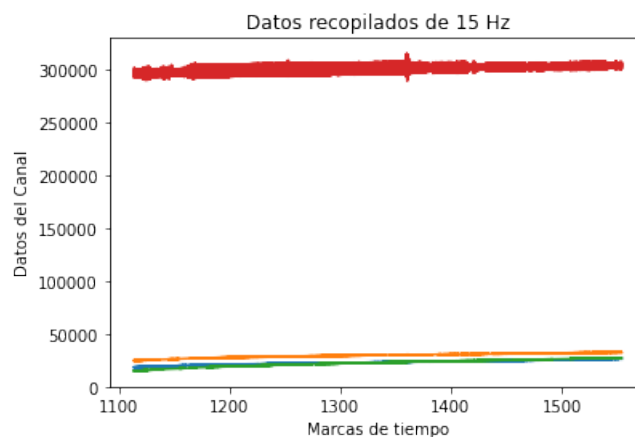


Figura 2.10: Datos de estímulos de 15 Hz.

2.2.5.1 Filtrado y división de épocas

Lo primero que se debe realizar es la división de las épocas. Esto quiere decir, al conjunto de datos se los va a cortar en subconjuntos mucho más pequeños que representan el tiempo que duró el estímulo. Una vez que la data haya sido dividida, se creará un objeto RAW por cada subconjunto. Estos objetos son estructuras de datos MNE-python que se basan en el formato de archivo de Neuromag para el mapeo funcional del cerebro [37] como se mencionó en la sección 2.2.4.1. Gracias a estos nuevos objetos se puede visualizar la

densidad espectral de potencia (power spectral density o PSD por sus siglas en inglés) de uno de los subconjuntos como se muestra en la Figura 2.11. Tomando en cuenta que alrededor de la frecuencia 50 Hz existe un pico, se realiza un filtro de muesca alrededor de esta frecuencia para evitar que la influencia de la frecuencia en la red sea innecesaria.

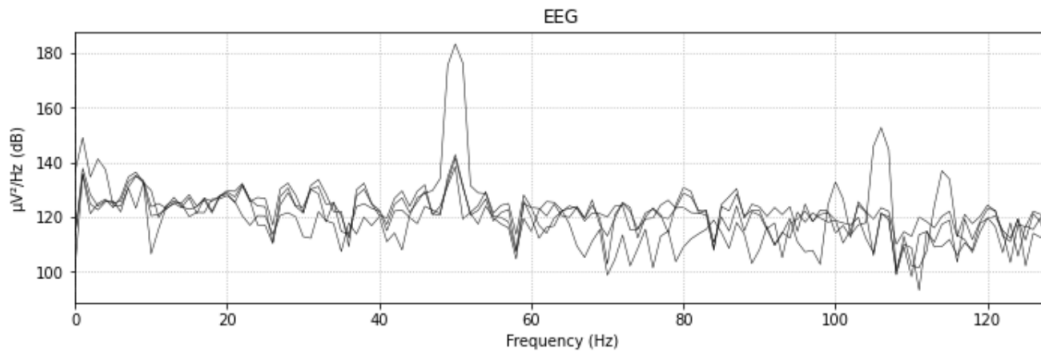


Figura 2.11: PSD de un estímulo visual.

2.2.5.2 Extracción y normalización de las características

Después del filtrado y división de las épocas, lo que se debe hacer a continuación es identificar los canales óptimos de EEG y conseguir resultados precisos mientras se diferencian las frecuencias de estimulación.

Luego se realiza un corte desde 10 Hz hasta 20 Hz. Esto es porque fuera de este rango del espectro no se incluyen las frecuencias con las que los participantes fueron estimulados. Una vez realizado estos cambios, se procede a calcular el PSD con el método de Welch. Este método es utilizado para estimar la potencia de una señal a diferentes frecuencias. Además, se basa en el concepto de utilizar estimaciones de espectro de periodograma, que son el resultado de convertir una señal del dominio del tiempo al dominio de la frecuencia [38]. Dado que la frecuencia con la que se estimula al participante también aparecerá en el EEG, es útil cambiar el dominio del tiempo por el dominio de la frecuencia utilizando este método. Para lograr utilizar este método se importa la función `psd_welch()` desde la librería MNE-Python que se incluye en el paquete de `time_frequency`. En el código del Anexo III, se puede apreciar todos los pasos mencionados en esta sección.

Como resultado se obtiene una mejor representación de las características para el espectro restante. La Figura 2.12 muestra los PDS de cada frecuencia de los estímulos realizados a los participantes. Para mostrar estas gráficas en Python se usa el código presentado en el

Anexo IV.

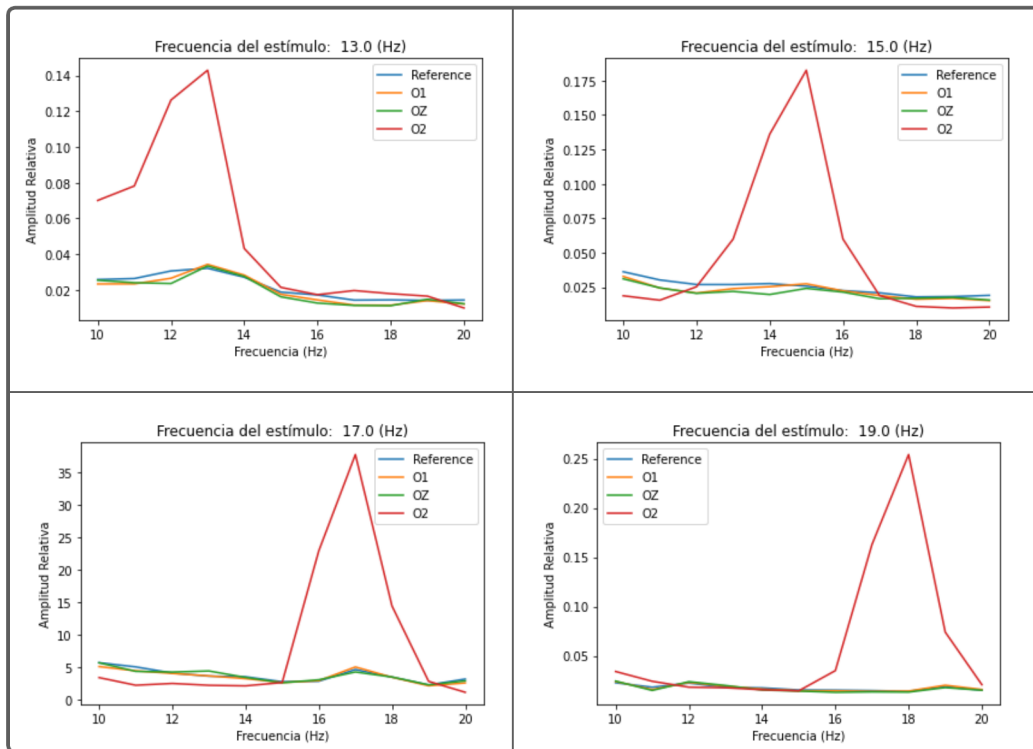


Figura 2.12: PSD de cada frecuencia de estimulación.

2.2.6 Clasificación

Para la creación del modelo de clasificación, se realizó una comparación de modelos existentes donde se utilizó la misma base de datos que se utilizó en esta investigación [21]. La Tabla 2.1 contiene la información sobre los modelos de clasificación seleccionados para la comparación.

Tabla 2.1: Comparativa entre modelos de clasificación de SSVEP

Método	Precisión
Multilayer Perceptron (MLP)	64.06 %
Linear Discriminant Analysis (LDA)	62.71 %
Quadratic Discriminant Analysis (QDA)	60.52 %
Random Forest	66.98 %

Esta comparación entre modelos es muy importante porque nos muestra el resultado obte-

nido al entrenar el mismo conjunto de datos con métodos distintos. Por lo que, se decidió utilizar un método diferente a los mostrados en la Tabla 2.1. El modelo de clasificación elegido es el algoritmo k-NN, ya que este algoritmo es un método de aprendizaje supervisado no paramétrico atractivo para los sistemas SSVEP - BCI [39]. Durante este algoritmo, los datos de entrada se asignan a la clase en función de la distancia euclidiana al K-ésimo vecino más cercano.

2.2.7 Entrenamiento del modelo

Para realizar el entrenamiento del modelo, lo primero que se hizo fue dividir el conjunto de datos entre datos de entrenamiento y datos de validación, tomando el 80% para el entrenamiento y el 20% para la validación. Dando como resultado 3840 datos para entrenar y 960 datos para validar el modelo.

Como siguiente paso se busca el valor óptimo de k para entrenar el modelo, generalmente se encuentra hallando la raíz cuadrada del total de las muestras. Sin embargo, se creó una función iterativa donde se entrenó una y otra vez con un valor creciente de k, dando como resultado que 31 es el mejor valor de K como se muestra en la Figura 2.13.

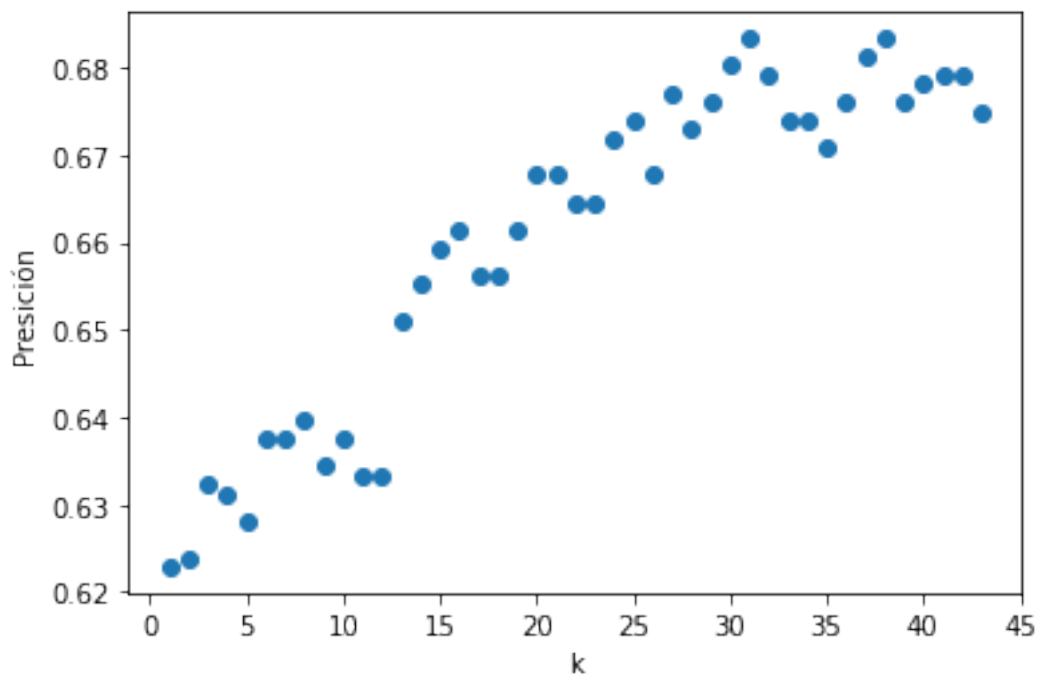


Figura 2.13: Mejor valor de K para el entrenamiento.

Por otro lado, también se puede obtener el mejor valor de K calculando la tasa de error por

cada valor de K en un rango definido, es decir, también se creó una función iterativa donde recorra los valores de K entre el 1 al 70, dando como resultado que 31 es el valor óptimo para este clasificador como se muestra en la Figura 2.14.

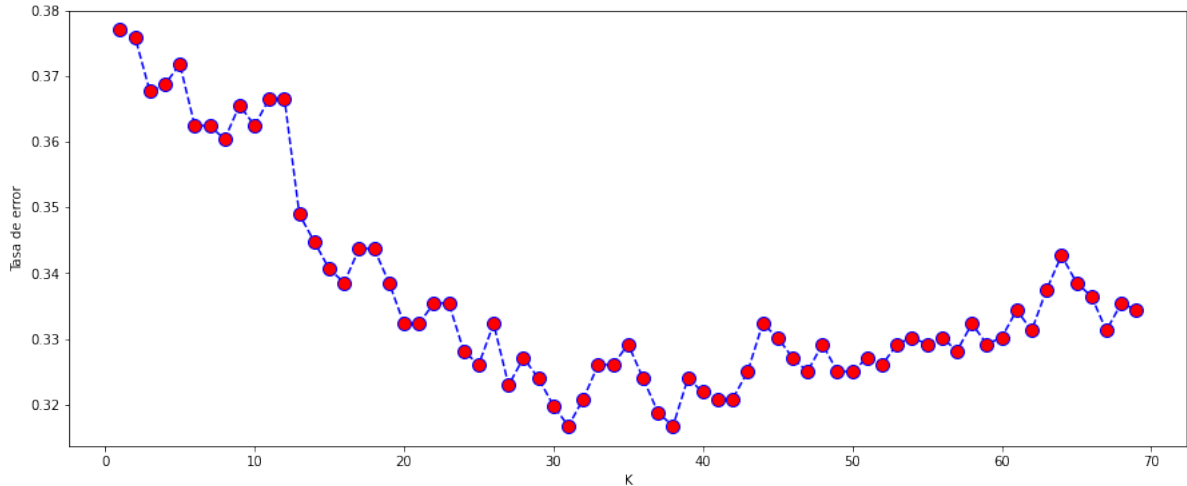


Figura 2.14: Tasa de errores en un rango de valores de K.

Una vez establecido el valor de k, se lo pasa como parámetro para el entrenamiento del modelo, dando como resultado una presión del entrenamiento de 66 %. El tiempo que transcurre durante el entrenamiento es bastante rápido, sin superar los 5 segundos.

Para no perder el modelo recién entrenado, se crea un archivo de tipo Pickle con extensión *.pkl. Esto permite serializar objetos a archivos en el disco, en este caso el modelo, y en cualquier otro momento se los puede deserializarlos en un programa en tiempo de ejecución. Este archivo contiene un flujo de bytes que representa los objetos. Para lograr guardar este archivo se requiere la librería Joblib que se encarga de la canalización ligera en Python, específicamente en el almacenamiento en caché. Se utiliza el método dump para guardar el nuevo archivo serializado del modelo de clasificación. Como se muestra en la Figura 2.15, es el directorio donde se almacenan los modelos entrenados.

2.2.8 Inferencia del modelo

Para la inferencia, se realizó un programa diferente en un ambiente virtual de Google Colab, en el cual se tiene una función para cargar los modelos por medio de la librería Joblib. Además, en este programa se tiene otra función donde procesa los datos ingresados ya que cada muestra tiene 4 lecturas, es decir, una lectura por cada electrodo de 11 valores que corresponden a la amplitud del PDS entre las frecuencias 10 Hz y 20 Hz. Por lo tanto,

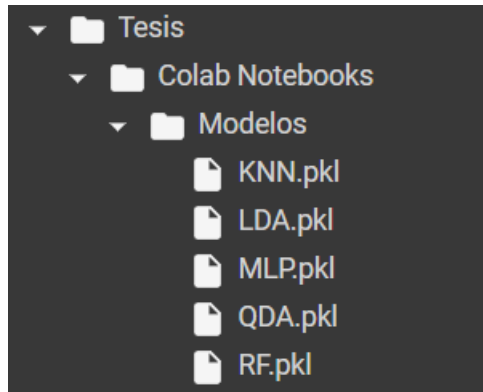


Figura 2.15: Directorio de modelos de clasificación.

con esta función se organiza los datos en una matriz unidimensional de 1x44. Esto permite pasar esa matriz unidimensional por la función de predicción del modelo k-NN. Dando como resultado una clasificación que puede ser 13, 15, 17 o 19 Hz de estímulo visual. Una vez se obtiene ese valor, lo siguiente es crear sentencias condicionales, para que el paciente con neuropatía pueda comunicarse. En este caso, se seleccionó cuatro de las señales más importantes para que un paciente pueda mejorar su estilo de vida. En la tabla 2.2 se muestra la clasificación de la señal junto con su comando.

Tabla 2.2: Comandos de comunicación con respecto a su estímulo

Comando de comunicación	Frecuencia del estímulo
El paciente intenta afirmar	13.0 Hz
El paciente requiere de una bebida	15.0 Hz
El paciente quiere ir al baño	17.0 Hz
El paciente intenta negar	19.0 Hz

3 PRUEBAS, RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En esta sección se presentarán las pruebas y los resultados obtenidos al evaluar el modelo de clasificación. Así como las conclusiones y recomendaciones.

3.1 PRUEBAS

Para las pruebas del modelo de clasificación de estímulos visuales se divide en dos grupos. El primero se realiza en base a los datos que fueron divididos en este caso 960 datos para la validación que representa al 20 % del total del conjunto de datos. Por otra parte, el segundo grupo se lo realizó una inferencia del modelo en otro entorno virtual de Google Colab.

3.1.1 Entrenamiento

Para el entrenamiento se realizó varios procesos de entrenamiento variando el número de muestra que se encuentra como parámetros en la función `train_test_split()`. Tomando en cuenta que mientras más datos de entrenamiento se tenga, mucho mejor será el resultado en su precisión. Sin embargo, al tener pocos datos en general, se opta por elegir la regla del 80/20 ya que así la precisión pudo mejorar un poco. Además, se hizo un análisis para encontrar el mejor valor de K para el modelo de k-NN como se mencionó en la sección 2.2.7. Para obtener el valor de K óptimo y crear las gráficas se utilizó el código que se encuentra en los Anexos V y VI. Una vez se tiene que $K = 31$ se procede a realizar el entrenamiento como se muestra en el código del Anexo VII.

3.1.2 Validación

Para la validación se escogió el 20 % del conjunto total de datos de manera aleatoria. Luego, se utiliza el modelo para clasificarlos, y por último se los compara con las etiquetas de esos datos, y se realiza una matriz de confusión. Esto se encontrará en la sección de Resultados 3.2.1.

3.1.3 Inferencia

En la inferencia, se realiza la limpieza de los datos y el procesamiento de las señales de la misma manera que la Sección 2.2.5, y su código se muestra en el Anexo III. Para lograr que un paciente pueda comunicarse, es necesario tener una función con condicionales donde se compara la clasificación obtenida por el modelo al momento de pasar como parámetro a la matriz unidimensional de 1x44 mencionada en la Sección 2.2.8 obtenida por el procesamiento de la señal mencionado anteriormente. Se puede revisar el código en el Anexo VIII.

3.2 RESULTADOS

Esta sección será dividida en dos partes, la primera los resultados en base al entrenamiento y validación. En cambio, la segunda parte se basa en la inferencia del modelo.

3.2.1 Entrenamiento y validación

Como se mencionó en la Sección 2.2.7 el valor de precisión fue de 66 %. Por otra parte, para la evaluación del modelo existen varias métricas, en este apartado se presentarán los resultados de las cinco métricas más importantes a la hora de evaluar un modelo de clasificación.

3.2.1.1 Matriz de confusión

Esta métrica permite visualizar el rendimiento del modelo de clasificación. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Esto significa que puede visualizar los éxitos y fracasos que recibe su modelo a medida que le pasa datos reales y los que se predijo. En la Figura 3.1, muestra la matriz de confusión del modelo k-NN de clasificación de señales SSVEP, donde se aprecia que para la etiqueta de 13 Hz tuvo 183 aciertos, predijo erróneamente 25 como la etiqueta 15 Hz, 12 como la etiqueta 17 Hz y 20 como la etiqueta 19 Hz. Por otro lado, para la etiqueta 15 Hz acertó 156 datos, pero predijo 25 de ellos como la etiqueta 13 Hz, 33 como la etiqueta 17 Hz y 22 como 19 Hz. Para la etiqueta 17 Hz el modelo acertó con 137 datos y para la etiqueta 19 Hz acertó 180.

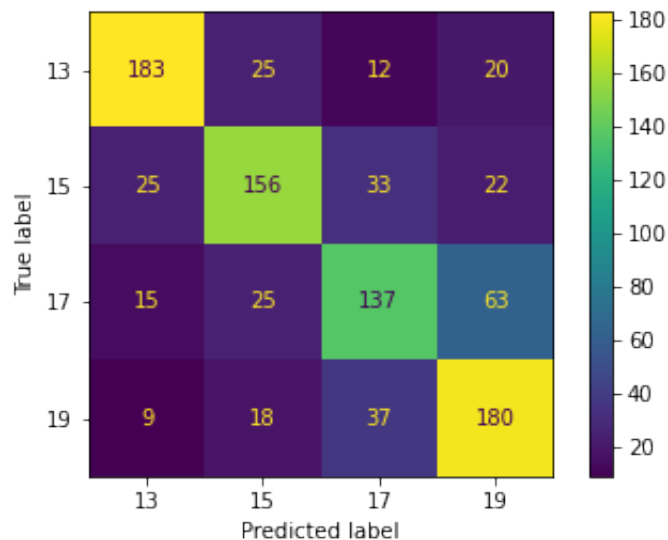


Figura 3.1: Matriz de confusión del modelo de clasificación k-NN.

3.2.1.2 Exactitud

La exactitud de la clasificación es la relación entre las predicciones correctas y el número total de predicciones. Es decir, con qué frecuencia es correcto el clasificador. En este caso, utilizando la función `accuracy_score()` de la librería `sklearn` del paquete `metrics`, dio un resultado de 68.33 %.

3.2.1.3 Precisión

La precisión es la relación entre las predicciones correctas y el número total de predicciones correctas previstas. Esto mide la precisión del clasificador a la hora de predecir casos positivos. En este caso, se pasó como parámetro extra en la función `precision_score()` el average como macro, ya que por defecto está en binary. Esto quiere decir que la función solo vale si existen dos etiquetas, pero en este caso son cuatro. Según la documentación [40], macro calcula las métricas para cada etiqueta y encuentra su media no ponderada. Sin tener en cuenta el desequilibrio de las etiquetas. Como resultado se obtuvo el 68.56 %.

3.2.1.4 Sensibilidad

También conocido como Recall es la relación entre las predicciones positivas correctas y el número total de predicciones positivas. Es decir, qué tan sensible es el clasificador para detectar instancias positivas. Esto también se conoce como la tasa verdadera positiva. El resultado obtenido es 68.30 %.

3.2.1.5 Puntaje de F1

Este puntaje es la medida armónica de la memoria y la precisión, con una puntuación más alta como mejor modelo. El resultado de esta métrica es 68.28 %.

El resumen de las métricas se encuentra en la Tabla 3.1

Tabla 3.1: Resultados del modelo k-NN para clasificación de señales SSVEP.

Métrica de evaluación	Resultado
Exactitud / Accuracy score	68.33 %
Precisión / Precision score	68.56 %
Sensibilidad / Recall	68.30 %
Puntaje de F1 / F1 score	68.28 %

3.2.2 Inferencia

Los resultados de la inferencia se basan en los resultados mencionados anteriormente, es decir, este modelo tiene sus limitaciones ya que para el entrenamiento son pocos datos los que fueron usados. Para demostrar esto se pasó como parámetro una matriz unidimensional de 1x44 en la función creada para ayudar a que un paciente se comuniqué. En este caso, se tomó una característica conocida, para comprobar que el modelo clasifica correctamente. Como se muestra en la Figura 3.2, la respuesta del paciente sería que está negando algo al momento de ver un estímulo visual de 19 Hz.

```
[8] prueba = caracteristicas[4000]

[9] def demostracion(datos):
    resultado = modelo.predict(datos)
    if(resultado==13):
        print('El paciente intenta afirmar')
    elif (resultado==19):
        print('El paciente intenta negar')
    elif (resultado==15):
        print('El paciente requiere de un bebida')
    elif (resultado==17):
        print('El paciente quiere ir al baño')

[10] prueba
array([8.45685986e+00, 6.43113339e-01, 7.53935700e-01, 1.65939823e+00,
       2.31269182e+00, 1.30225867e+00, 1.34832938e+00, 9.98136798e-01,
       8.77536261e-01, 1.37923867e+00, 8.33146022e-02, 8.26118247e+00,
       3.35361468e-01, 5.33381993e-01, 1.64243278e+00, 2.07545673e+00,
       1.54235406e+00, 1.57146348e+00, 1.15372200e+00, 5.73642117e-01,
       1.96511567e+00, 5.70873288e-02, 6.77367006e+00, 1.24774951e+00,
       1.24537171e+00, 2.50324169e+00, 1.42378451e+00, 9.82978156e-02,
       1.29757712e+00, 1.98836043e+00, 4.91575275e-01, 1.28390787e+00,
       3.78949332e-02, 4.60136446e-01, 4.32808322e+00, 5.37774052e+00,
       4.14684832e+00, 6.39280355e-01, 4.43310630e+00, 2.40703818e+00,
       6.36713277e+01, 6.60926614e+01, 2.83189090e+00, 2.30181589e+00])

[11] demostracion([prueba])
El paciente intenta negar

[15] modelo.predict([prueba])
array([19.])

[17] etiquetas[4000]
19.0
```

Figura 3.2: Resultado de la inferencia del modelo.

3.3 CONCLUSIONES

- ❑ En este trabajo integrador, se aplicó técnicas de aprendizaje de máquina aplicado a un BCI de señales EEG basado en SSVEP, donde clasifica cuatro etiquetas según los estímulos visuales que recibe un paciente por medio de una interfaz gráfica. Este BCI es especialmente diseñado para personas con discapacidad motora y del habla, permitiendo así tener hasta 5 comandos que se puede crear por medio de condicionales. En este caso, facilitar la comunicación en pacientes con Neuropatía periférica.
- ❑ Gracias a la revisión de la literatura, se concluye que es posible ayudar a personas a comunicarse por medio de los BCI's y las técnicas de aprendizaje de máquina.
- ❑ En base a lo mencionado en este trabajo integrado se concluye que el buen tratamiento y manejo de la base de datos permitió obtener los resultados con la finalidad de poder analizar el modelo creado.
- ❑ Para evaluar el desempeño del clasificador k-NN desarrollado, se realizó la prueba con 960 datos. En base a las pruebas y resultados obtenidos en la evaluación del modelo se determina que su Exactitud es de 68.33 %.
- ❑ Aplicar técnicas de aprendizaje de máquina para procesar y clasificar señales de EEG basadas en SSVEP no conllevan una complejidad tan alta como aparentan, ya que actualmente existen varios mecanismos de ayuda como librerías, API's, programas de las mismas marcas de los dispositivos de adquisición de ondas cerebrales, etc. Que permiten realizar procesos complejos con pocos pasos.

3.4 RECOMENDACIONES

Como autor de este trabajo integrador, recomiendo para trabajos futuros:

- ❑ Para mejorar las métricas del modelo se necesitan muchos más datos de señales EEG basadas en SSVEP. Esto conlleva un problema, ya que la obtención de estos datos es un proceso largo y cansado, es complicado que varios pacientes estén por horas sentados recibiendo estímulos visuales. Estos pueden causar molestias a los pocos minutos de recibirlos. Mi solución en este caso es utilizar una red generativa

antagónica también conocida como GAN, para generar muchos datos que pueden servir para el entrenamiento del modelo de clasificación.

- ❑ Si se desea realizar el BCI basado en SSVEP en tiempo real, se recomienda tomar en cuenta los costos de los kits de adquisición de ondas cerebrales, ya que son bastante elevados. Existen alternativas más económicas, pero las lecturas serán más imprecisas por lo que el clasificador no será tan óptimo. Además, se tiene que tener las mismas características de la obtención ya que si un parámetro varía, la señal recibida será un tanto diferente lo cual hace que los modelos no funcionen.
- ❑ Se puede aumentar los comandos de comunicación siempre y cuando se tenga datos con más estímulos visuales. Así no queda limitado a cinco opciones como en este caso. También se puede tomar en cuenta, que los comandos son el número de estímulos que se da al paciente más uno. Esto es porque también se puede tomar como comando cuando el paciente no recibe un estímulo visual.
- ❑ Se recomienda crear una interfaz intuitiva, está debe estar conectado en tiempo real al dispositivo de adquisición de señales, al clasificador y a la interfaz de estímulos visuales. Así el paciente podrá comunicarse en el momento que necesite de asistencia.
- ❑ Para una mejor inferencia del modelo es necesario probar este BCI basado en SSVEP en varios pacientes con Neuropatía periférica o en pacientes con discapacidad motora o del habla.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] J. A. Castañeda Fernández y J. Corral García, «Neuropatías periféricas,» *Medisan*, 2003.
- [2] J. R.-P. Dale J Lange, «Multifocal motor neuropathy,» es, *UpToDate*, J. M. Shefner, ed., feb. de 2022.
- [3] A. Traba López y Á. Esteban Garcí´a, «Neuropatí´a motora multifocal,» es, *Rev. Neurol.*, vol. 30, n.º 06, pág. 519, 2000.
- [4] A. Combaz y M. M. Van Hulle, «Simultaneous detection of P300 and steady-state visually evoked potentials for hybrid brain-computer interface,» en, *PLoS One*, vol. 10, n.º 3, e0121481, 2015.
- [5] Y.-H. Lu, «Hemodynamic effects of endoleak formation in abdominal aortic aneurysm patients with stent-graft implants,» en, *J. Med. Biol. Eng.*, vol. 34, n.º 6, págs. 554-558, 2013.
- [6] B. Dick, «Action research: action and research [On line,» ene. de 2002.
- [7] D. Tripp, «Action research: a methodological introduction,» *Educacao e pesquisa*, vol. 31, págs. 443-466, 2005.
- [8] B. Rohwerder, «Assistive technologies in developing countries,» 2018.
- [9] M. Clinic, *Anatomy of the Brain*. dirección: <https://mayfieldclinic.com/pe-anatbrain.htm>.
- [10] D. N. N. Lathulerie, «EVALUACIÓN DE LA TERAPIA DE HIPERTERMIA MAGNÉTICA EMPLEANDO MODELOS DE PERFUSIÓN DE IMÁGENES PARA GLIOBLASTOMAS MULTIFORMES,» Tesis doct., INSTITUTO VENEZOLANO DE INVESTIGACIONES CIENTÍFICAS, 2018.
- [11] J. E. Hall y A. Guyton, «Fisiología médica,» *Guyton y Hall. 12o*, págs. 1003-1017, 2011.
- [12] B. He, ed., *Neural Engineering*, 3.ª ed. Cham, Switzerland: Springer Nature, 2021.

- [13] X. Gui, C. Chuansheng, L. Zhong-Lin y D. Qi, «Brain imaging techniques and their applications in decision-making research,» *Xin li xue bao. Acta psychologica Sinica*, vol. 42, n.º 1, pág. 120, 2010.
- [14] E. P. Zambalde et al., «SSVEP-based BCI with visual stimuli from LCD screen applied for wheelchair control: Offline and online investigations,» 2018.
- [15] Y. Shen, «Machine learning based epileptic seizure detection for responsive neurostimulator system optimization,» en *Journal of Physics: Conference Series*, IOP Publishing, vol. 1453, 2020, pág. 012 089.
- [16] X. Gu, Z. Cao, A. Jolfaei et al., «EEG-Based Brain-Computer Interfaces (BCIs): A Survey of Recent Studies on Signal Sensing Technologies and Computational Intelligence Approaches and Their Applications,» *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, n.º 5, págs. 1645-1666, 2021. DOI: 10.1109/TCBB.2021.3052811.
- [17] F. Beverina, G. Palmas, S. Silvoni, F. Piccione y S. Giove, «User adaptive BCIs: SSVEP and P300 based interfaces,» *PsychNology Journal*, vol. 1, págs. 331-354, ene. de 2003.
- [18] K. Lewin et al., «Action research and minority problems,» *Journal of social issues*, vol. 2, n.º 4, págs. 34-46, 1946.
- [19] R. N. Rapoport, «Three dilemmas in action research: with special reference to the Tavistock experience,» *Human relations*, vol. 23, n.º 6, págs. 499-513, 1970.
- [20] M. Staron, «Action research in software engineering: Metrics' research perspective (invited talk),» en *International Conference on Current Trends in Theory and Practice of Informatics*, Springer, 2019, págs. 39-49.
- [21] B. Specht, D. Stichling, A. Farshad, S. Martinez, «SSVEP Evaluation for Robot Control,» en, Tesis de mtría., Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany.
- [22] g. tec medical engineering GmbH, «g.USBamp, USB Biosignal Amplifier: Instruction for use,» en, inf. téc., 2017.
- [23] *Lab Streaming Layer*, <https://labstreaminglayer.org/#/>, Accessed: 2022-8-25.
- [24] *Labstreaminglayer*, en, feb. de 2018.

- [25] D. Zhu, J. Bieger, G. Garcia-Molina y R. Aarts, «A survey of stimulation methods used in SSVEP-based BCIs,» *Computational Intelligence and Neuroscience*, vol. 2010, pág. 702-357, ene. de 2010. DOI: 10.1155/2010/702357.
- [26] C. Shannon, «Communication in the Presence of Noise,» *Proceedings of the IRE*, vol. 37, n.º 1, págs. 10-21, 1949. DOI: 10.1109/JRPROC.1949.232969.
- [27] H. Nyquist, «Certain topics in telegraph transmission theory,» *Transactions of the American Institute of Electrical Engineers*, vol. 47, n.º 2, págs. 617-644, 1928.
- [28] S. Ladouce, L. Darnet, J. J. Torre Tresols, S. Velut, G. Ferraro y F. Dehais, «Improving user experience of SSVEP BCI through low amplitude depth and high frequency stimuli design,» en, *Sci. Rep.*, vol. 12, n.º 1, pág. 8865, 2022.
- [29] X. Gao, D. Xu, M. Cheng y S. Gao, «A BCI-based environmental controller for the motion-disabled,» *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 11, págs. 137-40, jul. de 2003. DOI: 10.1109/TNSRE.2003.814449.
- [30] D. H. Johnson, «Signal-to-noise ratio,» *Scholarpedia*, vol. 1, n.º 12, pág. 2088, 2006.
- [31] E. Larson y A. K. Lee, «The cortical dynamics underlying effective switching of auditory spatial attention,» *Neuroimage*, vol. 64, págs. 365-370, 2013.
- [32] A. Gramfort, M. Luessi, E. Larson et al., «MEG and EEG data analysis with MNE-Python,» *Frontiers in neuroscience*, pág. 267, 2013.
- [33] S. Van Der Walt, S. C. Colbert y G. Varoquaux, «The NumPy array: a structure for efficient numerical computation,» *Computing in science & engineering*, vol. 13, n.º 2, págs. 22-30, 2011.
- [34] J. D. Hunter, «Matplotlib: A 2D graphics environment,» *Computing in science & engineering*, vol. 9, n.º 03, págs. 90-95, 2007.
- [35] L. Buitinck, G. Louppe, M. Blondel et al., «API design for machine learning software: experiences from the scikit-learn project,» *arXiv preprint arXiv:1309.0238*, 2013.
- [36] *Documentation overview — MNE 1.1.1 documentation*, en, <https://mne.tools/stable/overview/index.html>, Accessed: 2022-8-30.
- [37] *The Raw data structure: continuous data — MNE 1.1.1 documentation*, en, https://mne.tools/stable/auto_tutorials/raw/10_raw_overview.html, Accessed: 2022-8-28.

- [38] O. Solomon Jr, «PSD computations using Welch's method [Power Spectral Density (PSD)]: NASA STI,» Recon Technical Report. 1991. doi: 10.2172/5688766, inf. téc.
- [39] Y. Wu, G. Xu, Y. Wu et al., «An asynchronous detection algorithm for SSVEP-based BCI using gradient boosting decision tree,» en *Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition*, New York, NY, USA: ACM, 2020.
- [40] *Sklearn.Metrics.Precision_score*, en, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html, Accessed: 2022-9-5.

5 ANEXOS

ANEXO I. Proforma de un kit para la adquisición de señales EEG.

ANEXO II. Proforma del kit gtec g.USBamp.

ANEXO III. Script Filtrado y Extracción de características.

ANEXO IV. Script Graficar el PSD de las características.

ANEXO V. Script Encontrar el mejor valor de K.

ANEXO VI. Script Tasa de error para el mejor valor de K.

ANEXO VII. Script Entrenamiento del modelo de clasificación.

ANEXO VIII. Script Inferencia del modelo.

ANEXO I



Brain Support Corporation
8950 NW 27th Street, Suite 303
Doral, FL, 33172
Site: www.brainsupport.co

Importer / Consignee Delivery Address	Quote 08LA03Aug22epn.edu.ec
Escuela Politécnica Nacional Facultad de Sistemas Ecuador	Date: 03/08/2022 13:37 Validity: 05/10/2022 Cod: 1582828000043259070 Contact person: Jackson Cioneck E-mail: jackson@brainlatam.com
Contact Name: ANDRES WLADIMIR GARCIA CUVI E-mail: andres.garcia02@epn.edu.ec	TOTAL PRICE: \$ 35,635.00

Pos.	Art. No.	Description	Quant	Single value	Total value
1.	BP-100-3120	LiveAmp 8 system with actiCAP and Analyzer 18m Incl. BrainVision Recorder, actiCAP slim electrodes, 2 snap caps, Accessories, Analyzer license for 18 month	1	\$ 16,500.00	\$ 16,500.00
2.	BP-110-9040	Triggerbox Kit for LiveAmp Triggerbox + Wireless Trigger Transmitter	1	\$ 4,950.00	\$ 4,950.00
3.	BP-110-2000	Sensor and trigger extension kit for LiveAmp Complete kit incl. all cabling, power extender will need to be purchased locally	1	\$ 9,750.00	\$ 9,750.00
4.	BP-140-1020	GSR Sensor Kit	1	\$ 975.00	\$ 975.00
5.	Install&Check	Check List and first use. Conference of goods and installation. Estimated time up to 4 hours. Does not include training.	1	\$ 3,100.00	\$ 3,100.00
Subtotal:					\$ 35,275.00
Island / Handling cost:					\$ 360.00
Freight cost:					
Insurance cost:					
Total Price:					\$ 35,635.00

Term & Conditions:

Terms of payment: In Advance
Currency: U. S. Dollar
Terms of delivery (Incoterm 2010): Ex-Works: Gilching, Germany

IMPORTANT:

Our solutions are exclusive for neuroscientific studies
CLINICAL USE PROHIBITED

Jackson Cioneck
jackson@brainlatam.com

Figura 5.1: Proforma de un kit para la adquisición de señales EEG.

ANEXO II

Review your selection

BIO SIGNAL AMPLIFIER

g.USBamp RESEARCH	€ 11.900,00	× 1	€ 11.900,00
Includes:			
1 × g.USBamp RESEARCH amplifier			
1 × Carrying case			
1 × USB cable			
1 × Power supply			
1 × Power cable – Type: United States			
2 × Spare fuse			
1 × Digital input cable 1 + 2			
1 × Shortcut cable			
1 × g.USBamp RESEARCH Instructions for Use (PDF)			
1 × Sensors and Utilities Instructions For Use (PDF)			

ELECTRODES

Disposable adhesive mastoid electrodes	€ 29,00 € 27,55	× 1	€ 27,55
g.GAMMAgel	€ 29,00 € 27,55	× 1	€ 27,55
g.GAMMAcap2 EEG cap Type: M	€ 179,00 € 170,05	× 1	€ 170,05

ASSEMBLING

Assembling 64 channels	€ 500,00 € 482,60	× 1	€ 482,60
Includes:			
1 × Assembling 64 channels			
1 × Cable sleeve			

SOFTWARE

g.tec Suite 2020	€ 2.950,00 € 590,00	× 1	€ 590,00
------------------	--------------------------------	-----	----------

TOTAL	€ 15.595,00 € 13.197,75
--------------	------------------------------------

Figura 5.2: Proforma del kit gtec g.USBamp.

ANEXO III

```
1 #Recorrer la data
- for id_archivo, archivo_datos_mat in enumerate(os.listdir(data_path)):
-     if not archivo_datos_mat.endswith(".mat"):
-         continue
5
-
-     archivo_datos_mat = os.path.join(data_path, archivo_datos_mat)
-
-     #Leer archivos tipo mat
10     datos_mat = sio.loadmat(archivo_datos_mat)
-
-
-     Y = datos_mat["Y"][0, :]
15     datos_canal = datos_mat["X"]
-     time_stamp_indexes = datos_mat["trial"][0, :]
-
-     for i, inicio_estimulo in np.ndenumerate(time_stamp_indexes[:,2]): #cada 2
-         ids_archivo.append(id_archivo)
20         i = i[0]
-         fin_estimulo = inicio_estimulo + int(frecuencia_muestreo *
-             duracion_estimulo)
-
-         for j in range(n_divisiones):
-             datos_canal_actual = datos_canal[inicio_estimulo : inicio_estimulo +
25                 offset]
-             inicio_estimulo += offset
-
-             #Creo el objeto Raw
-             mne_raw = mne.io.RawArray(data=datos_canal_actual.T, info=info)
-             psds, frecuencias = psd_welch(mne_raw, fmin=fmin, fmax=fmax) #
30                 Calcula el PSD (power spectral density) usando el método de Welch
-
-             # Solo las etiquetas pares contienen etiquetas con frecuencias de
-                 estimulación
-             # las etiquetas impares son roturas, por lo tanto son cero
```

```

-         etiquetas.append(Y[i * 2])
35         características.append(psds.flatten())
-
-         if Y[i * 2] in frecuencia_a_promedio_pds:
-             frecuencia_a_promedio_pds[Y[i * 2]] += psds
-             factor_normalizacion[Y[i * 2]] += 1
40         else:
-             frecuencia_a_promedio_pds[Y[i * 2]] = psds
-             factor_normalizacion[Y[i * 2]] = 1

```

ANEXO IV

```

1 for frecuencia in frecuencia_a_promedio_pds.keys():
-     frecuencia_a_promedio_pds[frecuencia] /= factor_normalizacion[frecuencia]
-
-     plt.plot(frecuencias, frecuencia_a_promedio_pds[frecuencia][0], label="
-         Reference")
5     plt.plot(frecuencias, frecuencia_a_promedio_pds[frecuencia][1], label="O1")
-     plt.plot(frecuencias, frecuencia_a_promedio_pds[frecuencia][2], label="OZ")
-     plt.plot(frecuencias, frecuencia_a_promedio_pds[frecuencia][3], label="O2")
-     plt.legend()
-
10     plt.title("Frecuencia del estímulo: " + str(frecuencia) + " (Hz)")
-     plt.ylabel("Amplitud Relativa")
-     plt.xlabel("Frecuencia (Hz)")
-     # plt.savefig("pds_" + str(frecuencia) + ".png")
-     plt.show()
15     plt.clf()

```

ANEXO V

```
1 k_range = range(1, 44)
- scores = []
- for k in k_range:
-     knn = KNeighborsClassifier(n_neighbors = k)
5     knn.fit(X_train, y_train)
-     scores.append(knn.score(X_test, y_test))
- plt.figure()
- plt.xlabel('k')
- plt.ylabel('Presición')
10 plt.scatter(k_range, scores)
- plt.xticks([0,5,10,15,20,25,30,35,40,45])
```

ANEXO VI

```
1 import numpy as np
- error_rate = []
- for i in range(1,44):
-     knn = KNeighborsClassifier(n_neighbors = i)
5     knn.fit(X_train, y_train)
-     pred_i = knn.predict(X_test)
-     error_rate.append(np.mean(pred_i != train))
- plt.figure(figsize=(15,6))
- plt.plot(range(1,44), error_rate, color='blue', linestyle='dashed', marker='o',
-         markerfacecolor='red', markersize='10')
10 plt.xlabel('K')
- plt.ylabel('Tasa de error')
```

ANEXO VII

```
1 from sklearn.model_selection import train_test_split
- from sklearn.neighbors import KNeighborsClassifier
- from sklearn.metrics import accuracy_score
-
5 X_train, X_test, y_train, y_test = train_test_split(caracteristicas, etiquetas,
    test_size=0.2, random_state=176)
-
- n_neighbors = 31
-
- knn = KNeighborsClassifier(n_neighbors)
10 knn.fit(X_train, y_train)
- print('Precisión del clasificador K-NN en el conjunto de Train: {:.2f}'
-       .format(knn.score(X_train, y_train)))
- print('Precisión del clasificador K-NN en el conjunto de Test: {:.2f}'
-       .format(knn.score(X_test, y_test)))
15
- y_pred_Train = knn_c.predict(X_train)
- y_pred_Test = knn_c.predict(X_test)
-
- print("Presición del entrenamiento: ", accuracy_score(y_train, y_pred_Train))
20 print("Presición de la validación: ", accuracy_score(y_test, y_pred_Test))
```

ANEXO VIII

```
1 def demostracion(datos):  
- resultado = modelo.predict(datos)  
- if (resultado==13):  
-     print('El paciente intenta afirmar')  
5 elif (resultado==19):  
-     print('El paciente intenta negar')  
- elif (resultado==15):  
-     print('El paciente requiere de un bebida')  
- elif (resultado==17):  
10 print('El paciente quiere ir al baño')
```