



Control de movimientos de una mano robótica utilizando neuroseñales

Trabajo de grado para optar al título de
Ingeniero en Mecatrónica

Trabajo de Grado

Autor:

Iván Fernando Vargas Ochoa

Tutor:

Ph.D. César Augusto Peña Cortés

20 de Junio de 2020



Control de movimientos de una mano robótica utilizando neuroseñales

Trabajo de grado Ingeniería Mecatrónica

Autor

Iván Fernando Vargas Ochoa

Tutor

Ph.D. César Augusto Peña Cortés
Universidad de Pamplona



Trabajo de grado para optar al título de Ingeniero en Mecatrónica

Universidad de Pamplona
Facultad de Ingenierías y Arquitectura
Pamplona, Colombia
20 de Junio de 2020

Dedico este trabajo a...

Dios, por ser mi inspiración y darme las fuerzas necesarias para cumplir uno de mis grandes anhelos.

A mi padre, por ser uno de los pilares más importantes en mi vida, por el apoyo incondicional durante toda mi carrera universitaria y por el trabajo y sacrificio que realiza día a día para que pueda desempeñarme como una mejor persona.

A mi madre, por todo el cariño y amor que me brinda, por sus consejos y confianza que me permitieron siempre seguir adelante y no decaer durante este proceso.

A mis hermanos, por siempre creer en mí y darme ese apoyo moral. A mis amigos y a todas aquellas personas, que nunca dudaron de mis habilidades y estuvieron para ayudarme en los momentos más difíciles.

Agradecimientos

Agradezco a mis padres por todo su amor, trabajo y sacrificio que han realizado durante el transcurso de mi carrera, por ser los promotores de mi sueño, por su apoyo incondicional y todas sus enseñanzas. Gracias a ustedes hoy puedo cumplir uno de mis grandes sueños.

Asimismo, agradezco a mi director de tesis César Augusto Peña Cortes, por guiarme durante el desarrollo de mi trabajo con sus valiosos conocimientos. Por todo el tiempo dedicado en asesoramiento, por estar siempre presente, por facilitarme el uso de herramientas y espacios de trabajo.

Resumen

En este trabajo se presenta la construcción de una mano robótica impresa en 3D que permite emular la naturaleza y agarre de la mano humana mediante el uso de neuroseñales. Actualmente uno de los principales problemas que presentan las prótesis de mano robóticas más sofisticadas, es la forma en que se le indican los movimientos a realizar. Las que han obtenido mejores resultados utilizan sistemas invasivos, lo cual hace necesario que el usuario sea sometido a una costosa y delicada cirugía. El objetivo principal del sistema desarrollado, es permitir que una persona pueda controlar los movimientos y/o gestos de una mano robótica mediante el uso de sus pensamientos, de tal forma que el sistema permita realizar movimientos con mayor fluidez y de manera natural a los usuarios. Para esto, se hace uso de una interfaz cerebro computadora (BCI) de tipo no invasiva y de bajo costo, que permite generar las referencias del sistema de control de la mano robótica mediante el uso de las ondas cerebrales y sus derivaciones. Se presentan las herramientas utilizadas para la creación de los algoritmos, así como los protocolos de comunicación empleados para la comunicación con la placa de control, la cuál es la encargada de variar la posición de cada uno de los actuadores de la mano robótica. Finalmente, se presentan algunas pruebas experimentales realizadas, en donde se comparan los resultados obtenidos y se determina el porcentaje de acierto por cada usuario.

Palabras clave: Interfaz Cerebro-Computadora, Mano robótica, Neuroseñales, Procesamiento de señales, Impresión 3D.

Abstract

This work presents the construction of a 3D printed robotic hand that allows to emulate nature and the grip of the human hand through the use of neurosignals. Currently one of the main problems presented by the most sophisticated robotic hand prostheses is the way in which the movements to be performed are indicated. Those that have obtained better results use invasive systems, which makes it necessary for the user to undergo sometimes an expensive and delicate surgery. The main objective of the developed system is to allow a person to control the movements and / or gestures of a robotic hand through the use of their thoughts, in such a way that the system allows users to move more fluidly and naturally. . For this, a non-invasive and low-cost brain computer interface (BCI) is used, which allows to generate the references of the robotic hand control system through the use of brain waves and their derivations. The tools used to create the algorithms are presented, as well as the communication protocols used for communication with the control board. The possibility is in charge of varying the position of each of the actuators of the robotic hand. Finally, some experimental tests performed are presented, where the results obtained are compared and the percentage of success for each user is determined.

Keywords: Brain-Computer Interface, Robotic Hand, Neurosignals, Signal Processing, 3D Printing.

Índice general

| | |
|------------------------|------------|
| Agradecimientos | vii |
| Resumen | ix |
| Abstract | xi |

| Capítulos | Página |
|--|---------------|
| 1 Introducción | 1 |
| 1.1 Justificación | 3 |
| 1.2 Objetivos | 4 |
| 1.2.1 Objetivo General | 4 |
| 1.2.2 Objetivos específicos | 4 |
| 2 Marco Teórico | 5 |
| 2.1 La Electroencefalografía | 5 |
| 2.1.1 Historia de la Electroencefalografía | 6 |
| 2.1.2 El Cerebro | 7 |
| 2.1.3 Tipos de Electroodos | 8 |
| 2.1.4 Topologías de Medición de EEG | 8 |
| 2.1.5 Posicionamiento de los Electroodos | 9 |
| 2.1.6 Tipos de potenciales EEG | 10 |
| 2.1.7 Ritmos Cerebrales | 11 |
| 2.2 Interfaz Cerebro-Computador (BCI) | 13 |
| 2.2.1 Tipos de BCI | 13 |
| 2.2.2 Clasificación de los sistemas BCI basados en la respuesta Cerebral | 14 |
| 2.2.3 Tipos de respuestas de EEG para el control con BCI | 15 |
| 2.2.4 Dispositivos BCI más relevantes del mercado | 16 |
| 2.2.5 Comparativa de dispositivos BCI comerciales | 19 |
| 2.3 La mano humana | 20 |
| 2.3.1 La mano y la mente Humana | 21 |
| 2.3.2 Anatomía de la mano humana | 22 |
| 2.3.3 Huesos de la mano | 22 |
| 2.3.4 Músculos de la mano | 23 |
| 2.3.5 Tipos de agarres de presión | 24 |

| | | |
|----------|--|-----------|
| 2.3.6 | Amputación de miembro superior | 26 |
| 2.3.7 | Prótesis de mano | 26 |
| 2.3.8 | Evolución en el desarrollo de prótesis de mano | 27 |
| 2.3.9 | Clasificación de las prótesis de miembro superior | 27 |
| 2.3.10 | Prótesis Mecánicas | 28 |
| 2.3.11 | Prótesis eléctricas | 29 |
| 2.3.12 | Prótesis neumáticas | 29 |
| 2.3.13 | Prótesis mioeléctricas | 30 |
| 3 | Metodología | 31 |
| 3.1 | Componentes | 31 |
| 3.1.1 | Servomotor | 31 |
| 3.1.2 | Micro Servo | 32 |
| 3.1.3 | Placas de desarrollo WiFi | 33 |
| 3.2 | Construcción de la mano robótica | 36 |
| 3.2.1 | Diseño de piezas 3D | 36 |
| 3.2.2 | Ensamble | 38 |
| 3.2.3 | Modelo cinemático para un dedo de la mano robótica | 40 |
| 3.3 | Diseño y construcción de la placa de control | 43 |
| 3.3.1 | Esquema eléctrico | 43 |
| 3.3.2 | Diseño de PCB | 44 |
| 3.4 | Emotiv Epoc | 46 |
| 3.4.1 | Introducción al SDK de de Emotiv | 46 |
| 3.4.2 | Desarrollo de la Interfaz 1 | 50 |
| 3.4.3 | Desarrollo de la Interfaz 2 | 62 |
| 3.4.4 | Arquitectura del sistema | 68 |
| 4 | Resultados | 71 |
| 4.1 | Interfaz 1 | 71 |
| 4.2 | Interfaz 2 | 77 |
| 5 | Conclusiones | 85 |
| | Bibliografía | 87 |

Índice de figuras

| | | |
|------|--|----|
| 2.1 | Adquisición de biopotenciales mediante electroencefalograma. | 5 |
| 2.2 | Evolución de la tecnología electroencefalográfica humana y sus aplicaciones. | 7 |
| 2.3 | Áreas principales del cerebro. | 8 |
| 2.4 | Sistema internacional 10-20 para el posicionamiento de los electrodos. . | 10 |
| 2.5 | Procesamiento de la señal EEG para clasificarla por bandas de frecuencia. | 11 |
| 2.6 | Esquema general de una Interfaz Cerebro-Computador (BCI). | 13 |
| 2.7 | Neurosky MindWave Mobile 2 de Neurosky. | 16 |
| 2.8 | Emotiv Epoc de Emotiv Inc. | 17 |
| 2.9 | Emotiv Insight de Emotiv Inc. | 18 |
| 2.10 | Partes de la mano humana. | 20 |
| 2.11 | Predominio de los hemisferios del cerebro en las manos. | 21 |
| 2.12 | Huesos que constituyen la mano humana. | 23 |
| 2.13 | Músculos de la mano humana. | 24 |
| 2.14 | Clasificación de agarres definidos por Schlesinger. | 24 |
| 2.15 | Esquema de clasificación de agarres definidos por Kapandji. | 25 |
| 2.16 | Prótesis activa y pasiva. | 28 |
| 2.17 | Prótesis mecánica. | 28 |
| 2.18 | Prótesis de mano eléctrica MyoHand. | 29 |
| 2.19 | Prótesis de mano neumática. | 30 |
| 2.20 | Prótesis mioeléctrica. | 30 |
| 3.1 | Servomotor Tower Pro MG995. | 31 |
| 3.2 | Micro Servo Tower Pro MG90S. | 32 |
| 3.3 | Wemos D1 mini. | 34 |
| 3.4 | NodeMCU basada en el chip ESP8266. | 34 |
| 3.5 | NodeMCU basada en el chip ESP32. | 35 |
| 3.6 | Modelo 3D del soporte para Servomotores. | 36 |
| 3.7 | Modelo 3D de la cubierta exterior para los Servomotores. | 37 |
| 3.8 | Modelo 3D de la base de la mano robótica. | 37 |
| 3.9 | Ensamble 3D de la mano robótica. | 38 |
| 3.10 | Mano robótica ensamblada. | 39 |
| 3.11 | Asignación de los sistemas de coordenadas al dedo de la mano. | 40 |
| 3.12 | Esquema eléctrico de la placa de control. | 43 |
| 3.13 | Diseño PCB de la placa de control. | 44 |

| | | |
|------|--|----|
| 3.14 | Vista superior de la PCB con todos sus componentes. | 45 |
| 3.15 | PCB ubicada dentro de una de las piezas de la mano robótica. | 45 |
| 3.16 | Dispositivo Emotiv Epoc. | 46 |
| 3.17 | Emotiv Control Panel. | 47 |
| 3.18 | Análisis de emociones utilizando la suite afectiva. | 48 |
| 3.19 | Reconocimiento de gestos mediante la suite expresiva. | 49 |
| 3.20 | Procesamiento de comandos mentales. | 50 |
| 3.21 | Configuración de espacio de trabajo en Eclipse. | 51 |
| 3.22 | Diseño final de la interfaz gráfica 1. | 61 |
| 3.23 | Menú de movimientos. | 63 |
| 3.24 | Diseño final de la interfaz gráfica 2. | 67 |
| 3.25 | Arquitectura del sistema para el control de la mano robótica. | 68 |
| 3.26 | Diagrama de flujo para la interfaz gráfica 1. | 69 |
| 3.27 | Diagrama de flujo para la interfaz la interfaz gráfica 2. | 70 |
| 4.1 | Usuario ejecutando el comando mental agarre de pinza fina. | 73 |
| 4.2 | Usuario ejecutando el comando mental cerrar mano. | 73 |
| 4.3 | Diagrama de barras del porcentaje de acierto de cada secuencia. | 75 |
| 4.4 | Diagrama de barras de los indicadores estadísticos basados en el porcentaje de acierto. | 76 |
| 4.5 | Nivel medio de concentración y frustración de los usuarios de la interfaz 1. | 77 |
| 4.6 | Usuario ejecutando el movimiento pinza fina con cierre del 0%. | 80 |
| 4.7 | Usuario ejecutando el comando mental pinza fina con cierre del 70%. | 80 |
| 4.8 | Nivel medio del porcentaje de acierto de cada comando ejecutado. | 82 |
| 4.9 | Nivel medio de concentración y frustración de los usuarios que ejecutaron la acción p-f-7. | 83 |

Índice de tablas

| | | |
|------|--|----|
| 2.1 | Tabla de la distribución de los electrodos según la normativa 10-20. . . | 10 |
| 2.2 | Amplitud y rango de frecuencia de cada banda espectral. | 12 |
| 2.3 | Comparativa de las características físicas de cada uno de los dispositivos BCI. | 19 |
| 3.1 | Tabla comparativa de servomotores. | 32 |
| 3.2 | Tabla comparativa de micro servos. | 33 |
| 3.3 | Tabla comparativa de algunas placas de desarrollo con conexión WiFi. | 35 |
| 3.4 | Parámetros de Denavit-Hantenberg calculados a partir de la base del dedo índice. | 41 |
| 3.5 | Relación de gestos faciales con las acciones que se pueden ejecutar. . . | 62 |
| 3.6 | Relación de cada uno de los comandos mentales con su intensidad de cierre. | 63 |
| 4.1 | Tiempo de entrenamiento empleado por cada usuario. | 71 |
| 4.2 | Relación de cada comando mental con su imagen de referencia. | 72 |
| 4.3 | Resultados de las pruebas realizadas a cada usuario. | 74 |
| 4.4 | Índices estadísticos de las pruebas realizadas. | 75 |
| 4.5 | Comandos mentales para variar la intensidad de cierre de la mano robótica. | 77 |
| 4.6 | Tiempo de entrenamiento empleado por cada usuario en la interfaz 2. | 78 |
| 4.7 | Tipos de movimientos disponibles en el menú de la interfaz 2. | 78 |
| 4.8 | Porcentaje de reconocimiento para los eventos oculares y el gesto apretar dientes. | 79 |
| 4.9 | Abreviatura de cada uno de los movimientos de la mano robótica. | 79 |
| 4.10 | Resultados de las pruebas realizadas a cada usuario utilizando la interfaz 2. | 81 |
| 4.11 | Índices estadísticos de las métricas afectivas capturadas. | 83 |

Índice de Códigos

| | | |
|------|--|----|
| 3.1 | Algoritmo para establecer la comunicación entre la interfaz y el casco . | 52 |
| 3.2 | Instancia de objetos para la comunicación UDP | 53 |
| 3.3 | Eventos relacionados al entrenamiento de comandos mentales | 53 |
| 3.4 | Lectura de emociones | 54 |
| 3.5 | Identificación de comandos mentales | 56 |
| 3.6 | Algoritmo para crear y guardar perfiles de usuarios | 58 |
| 3.7 | Algoritmo para cargar perfiles previamente entrenados | 59 |
| 3.8 | Entrenamiento cognitivo | 60 |
| 3.9 | Reconocimiento de eventos oculares | 64 |
| 3.10 | Identificación de comandos mentales y gestos faciales en la interfaz 2 . | 65 |

1 Introducción

La mano humana es la extremidad final del miembro superior que permite realizar acciones como agarrar y manipular objetos con gran precisión. Es una herramienta sensitiva, que brinda al ser humano la posibilidad de sentir e interactuar su nuestro entorno y poder relacionarse con diferentes personas mediante el uso de gestos.

La pérdida de la mano puede deberse a diversos factores, como una amputación del miembro debido a un accidente sufrido o que no exista parcialmente debido a malformaciones. Generalmente, esto causa graves consecuencias en el amputado, debido que pierde la habilidad de sujetar y manipular objetos, además de que también se afecta de manera psicológica ya que existe un cambio en apariencia.

Hoy en día las prótesis de mano robótica son fundamentales para ayudar a las personas con deficiencias a integrarse a los entornos sociales, laborales y familiares. Una de sus funciones principales consiste en sustituir y emular la naturaleza de los movimientos de la parte faltante, de tal forma que permita al usuario llevar una mejor calidad de vida. En el presente trabajo se realizó la construcción de una mano robótica, que, aunque no cumple con los estándares para llegar a ser una prótesis, puede evolucionar para convertirse en una. Actualmente en el mercado existen diferentes tipos de prótesis, como por ejemplo las controladas por señales mioeléctricas (EMG) o las que utilizan una interfaz cerebro computadora (BCI) de tipo invasiva. A pesar de que cumplen con su función, tienden a tener un elevado costo, lo cual limita el acceso para ciertos usuarios. Otras desventajas que presentan, es que generalmente los movimientos que permiten realizar son limitados, bruscos y poco precisos, lo cual, en muchas ocasiones generan daños en los objetos que el usuario desea manipular.

Originalmente la adquisición y uso de las señales electroencefalográficas (EEG), estaba limitada únicamente en el campo de la medicina. Pero debido a los grandes avances de los últimos años en el campo tecnológico, este tipo de tecnología ha sido llevada a diferentes áreas, lo cual ha permitido que se realicen una gran multitud de aplicaciones.

Las empresas que más se han visto interesadas en el uso de la electroencefalografía (EEG), son las dedicadas al campo de los video juegos. Esto ha permitido que hoy en día sea mucho más sencillo adquirir un dispositivo captador de ondas cerebrales portátil.

En este trabajo se presenta el desarrollo de un sistema que mediante el uso de una interfaz cerebro computadora (BCI) de bajo costo y de tipo no invasiva, permite adquirir y relacionar las ondas cerebrales y sus derivaciones con algunos movimientos de una mano robótica impresa en 3D. De tal forma que permita a los usuarios mover la mano robótica sin necesidad de realizar un gran esfuerzo. También se presenta la construcción de la mano robótica, el diseño de la parte electrónica y los algoritmos utilizados. Así como el análisis cinemático para uno de los de la mano robótica. Por último, se realizan algunas pruebas experimentales con el fin de poder validar el correcto funcionamiento del sistema. De tal manera que, en un futuro, este pueda ser implementado en aquellas personas que, mediante el uso de una prótesis de miembro superior, sustituyen una parte de su cuerpo.

1.1 Justificación

Actualmente existen diversas formas de controlar una prótesis de mano robótica, entre estas se encuentran las prótesis controladas por señales mioeléctricas (EMG), que mediante el uso de potenciales eléctricos de los músculos del cuerpo permiten realizar movimientos como abrir y cerrar la mano. Pero presentan la desventaja que dependen directamente del estado del músculo y del tipo de amputación. A parte las que han obtenido los mejores resultados, utilizan agujas y sistemas invasivos para la obtención de las señales. Otras prótesis son controladas mediante el uso de una interfaz cerebro computadora (BCI) de tipo invasiva, la cual obtiene y analiza las ondas cerebrales que están relacionadas a la intención de movimiento. Pero requieren que el usuario se someta a una cirugía para el implante de los electrodos, lo cual en muchas ocasiones genera un problema de incompatibilidad por parte del organismo.

Teniendo en cuenta que uno de los principales problemas que presentan las prótesis de mano es la forma en que se le indican los movimientos a realizar y que en ocasiones estos movimientos resultan ser bruscos y poco precisos. Se procede a desarrollar un sistema que, mediante un dispositivo captador de ondas cerebrales de tipo no invasivo, de bajo costo y de fácil adquisición, permita a los usuarios controlar los movimientos de una mano robótica. Brindando una nueva alternativa de control ante los sistemas de manipulación de prótesis actuales.

1.2 Objetivos

1.2.1 Objetivo General

- Controlar los movimientos o gestos de una mano robótica por medio de neuroseñales capturadas con una interfaz cerebro computadora (BCI).

1.2.2 Objetivos específicos

- Construir una mano robótica basada en impresión 3D.
- Identificar posibles neuroseñales que pueda generar el usuario a voluntad para la generación de las referencias del sistema de control.
- Desarrollar un sistema que mediante el uso de una interfaz cerebro computadora (BCI) permita seleccionar o controlar los gestos de una mano robótica.
- Validar el funcionamiento del sistema de manera experimental.

2 Marco Teórico

2.1 La Electroencefalografía

La electroencefalografía es la parte de la medicina encargada de analizar y estudiar los registros de la actividad eléctrica del cerebro, con el fin de comprender el comportamiento del sistema nervioso. Esta actividad eléctrica normalmente es generada por la suma de potenciales postsinápticos de las neuronas en la corteza cerebral y el registro gráfico de dichas señales, es conocido como electroencefalograma (EEG). Estos potenciales son captados con la ayuda de electrodos conductores situados sobre el cuero cabelludo (figura 2.1), los cuales transmiten las señales obtenidas hacia un sistema electrónico (electroencefalógrafo), el cual es el encargado de filtrar y amplificar dichas señales, para finalmente obtener un resultado que es grabado por un dispositivo analógico (los resultados obtenidos son grabados en tiras de papel) o digital (las señales son procesadas e interpretadas por un ordenador con un software específico) [1]. Dependiendo de lo que se desee analizar la prueba se podrá realizar en diferentes estados, como vigilia o sueño profundo. De tal manera que esto permita analizar de manera efectiva el estado actual de la persona y se pueda identificar enfermedades, trastornos, etc.



Figura 2.1: Adquisición de biopotenciales mediante electroencefalograma.

2.1.1 Historia de la Electroencefalografía

Hans Berger, acreditado como el descubridor del EEG humano a finales de 1920, ideó la técnica para proporcionar una ventana en el cerebro. Sin embargo, el estudio de la electroencefalografía viene desde mucho antes [2]. En el año 1770, el médico, fisiólogo y físico italiano Luigi Galvani publicó algunas de sus investigaciones relacionadas a la electricidad animal. Pero ninguna de sus investigaciones generó un gran impacto. Debido en parte, a la influencia de Volta, importante científico quien de manera incorrecta consideraba que todos los hallazgos se debían al efecto ocasionado por un estímulo eléctrico proveniente de la batería sobre dos metales disímiles [3]. Años más tarde, Richard Caton, fisiólogo inglés, registró en 1875 la actividad eléctrica en la corteza cerebral de monos y conejos utilizando un galvanómetro conectado al córtex [4]. Sus investigaciones llamaron la atención de diferentes científicos, ya que esto demostraba la presencia de corrientes eléctricas en el cerebro, siendo más específicos en el encéfalo. Durante los experimentos realizados, pudo observar mediante el uso del galvanómetro, como la corriente eléctrica presentaba variaciones y como aumentaba a medida que se sumergía en un sueño. Años más tarde, Hans Berger decide apoyarse de las investigaciones realizadas por Richard Caton y descubre la “Ola Alfa”. Las cuales eran señales ocasionadas por los movimientos de los ojos.

La comunidad científica originalmente despidió a Berger y su EEG. Fue solo cuando los británicos fisiólogos Edgar Adrian y Bryan Matthews replicaron las observaciones de Berger en el año 1934, y posteriormente su estudio sobre el EEG fue aceptado como una medida no invasiva de los campos eléctricos del cerebro. Desafortunadamente, la investigación de Berger fue detenida por el régimen nazi en Alemania, y él finalmente se suicidó en el año 1941. No obstante, el legado de Berger ha sido la introducción de una técnica que rápidamente tomó una posición firme no solo en las aplicaciones clínicas, sino también en investigación de dominios de neurofisiología en el campo de las ciencias de la computación [2].

En el año 1950 el neurólogo norteamericano y experto en el área de la robótica Gray Walter descubre las ondas Theta (señales comprendidas entre los 4 a 8 Hz) y Delta (comprendidas entre 0 a 4 Hz), las cuales se encontraban relacionadas directamente a la luz y al sueño profundo, así como «el Efecto Photique», el cual demuestra que el cerebro reacciona y responde a los estímulos luminosos (y también sonoros) para alinear sus ondas cerebrales en consecuencia. Tiempo después Gray sería el encargado de desarrollar la primera máquina de amplificación de la topografía cerebral [5]. Gracias a él hoy en día es posible identificar las diferentes ondas cerebrales y clasificarlas por bandas de frecuencia, las cuales van desde las ondas Delta hasta las gammas.

En el año 1998 el investigador Philip Kennedy marcó un desarrollo significativo en el campo del mapeo cerebral cuando implantó el primer objeto de interfaz de computadora

cerebral en un ser humano [6]. En el año 2000 Miguel Nicolelis decide capturar las señales de unas 100 neuronas de forma invasiva en un mono, mientras mueve un joystick en un videojuego. Después, las señales se usan directamente para controlar el videojuego y el mono descubre que no necesita mover el brazo para jugar [7]. Estos fueron los primeros usos de la electroencefalografía como herramienta de control.

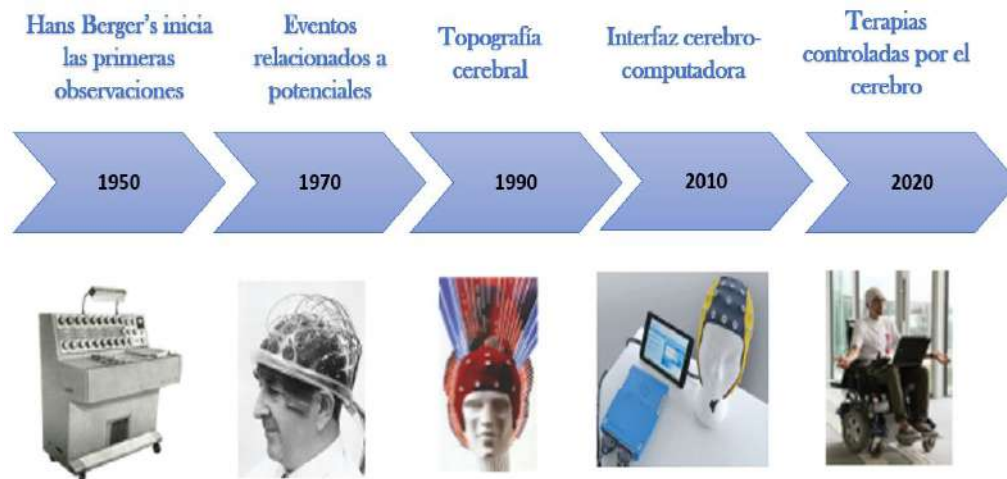


Figura 2.2: Evolución de la tecnología electroencefalográfica humana y sus aplicaciones.

Fuente: Biasiucci et al.

2.1.2 El Cerebro

El cerebro se puede dividir en tres partes principales; cerebro, cerebelo y tronco encefálico como se observa en la figura 2.3. El cerebro es la parte más grande e importante del cerebro humano y está asociado con funciones cerebrales relacionadas con pensamientos, movimientos, emociones y funciones motoras. El cerebro consta de dos hemisferios (hemisferio derecho y hemisferio izquierdo). Cada hemisferio está dividido en cuatro lóbulos denominados: frontal, parietal, occipital y temporal.

El lóbulo frontal se encuentra relacionado directamente con la personalidad, las emociones, la resolución de problemas, la motricidad, el razonamiento, y el movimiento. El lóbulo parietal es el lóbulo responsable de la sensación, comprensión sensorial, reconocimiento, percepción de estímulos y orientación. El lóbulo occipital es responsable del procesamiento visual. El lóbulo temporal está involucrado en el tratamiento de reconocimiento de estímulos auditivos, habla y memoria. El cerebelo se encuentra ubicado en la parte inferior de la cabeza y está dividido en dos hemisferios. Además, es el área encargada de la parte sensorial y es responsable de controlar la parte motora, la percepción sensorial y la coordinación [8].

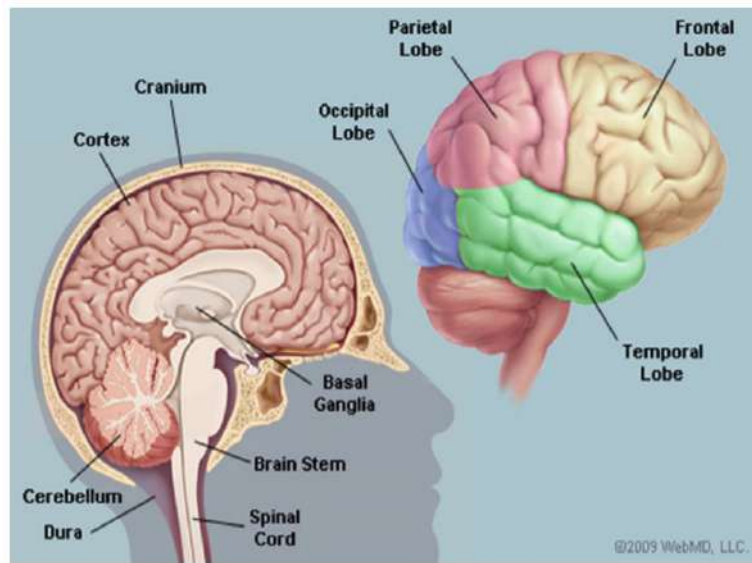


Figura 2.3: Áreas principales del cerebro.

Fuente: Siuly et al.

2.1.3 Tipos de Electrodo

Los electrodos utilizados para el registro electroencefalográfico poseen normalmente una superficie metálica la cual es ubicada sobre el lugar donde se desea obtener la señal electroencefalográfica y posteriormente es conectado al equipo de registro mediante un cable metálico aislado. El tipo de electrodo más común es el de disco de 8mm de diámetro el cuál se adhiere mediante una pasta conductora al cuero cabelludo. Aunque también están los electrodos que contactan con el cuero cabelludo mediante el uso de una esponja embebida en solución salina. Otro tipo de electrodos, son los de tipo invasivos (nasofaríngeos, etmoidales, esfenoidales o colocados en foramen oval), que sirven para registrar la actividad electroencefalográfica en regiones como la superficie medial del lóbulo frontal o temporal [9]. Una de las grandes desventajas que presenta el uso de electrodos de tipo invasivos para realizar el registro de EEG, es el costo de las mallas, cintillas y especialmente de un costo económico mayor. Los registros EEG invasivos, también pueden provocar hemorragias e infecciones que afectan de forma considerable la salud del usuario. Y en la mayoría de los casos el organismo del usuario genera rechazo y problemas de compactibilidad con los sensores implantados [10].

2.1.4 Topologías de Medición de EEG

Dependiendo de la configuración del montaje de los electrodos, se puede distinguir en tres topologías diferentes: montaje monopolar o referencial, bipolar y laplaciano.

- **Monopolar:** Esta configuración consiste en registrar la diferencia de potencial que se genera entre un electrodo ubicado en una zona cerebral activa y otro colocado sobre un área neutral (el lóbulo de la oreja) [11]. El electrodo encargado de realizar el registro de las señales electroencefalográficas es conocido como “electrodo activo”, mientras que el electrodo utilizado como referencia se le conoce como “referencial”. Generalmente los electrodos que son tomados como referencia poseen un voltaje de cero voltios. Los electrodos de referencia pueden ser ubicados en zonas como los lóbulos de las orejas o el mentón, que, aunque no sean cero, reduce mucho la detección de actividad eléctrica del cerebro [12].
- **Bipolar:** La configuración bipolar consiste en medir las diferencias de potencial entre cada par de electrodos activos [13]. Debido a esto, la señal que se obtiene no posee ningún tipo de contaminación y por lo tanto permite localizar la zona en la que se origina una señal en específico.
- **Laplaciano:** Al igual que en la configuración monopolar, se miden las diferencias de potencial entre un electrodo activo y uno de referencia. Sin embargo, en esta configuración el valor de referencia es el promedio de todos los electrodos que se encuentran situados alrededor del electrodo activo. Dicho promedio es calculado de forma digital o mediante métodos análogos [14].

2.1.5 Posicionamiento de los Electrodo

Según la normativa 10-20 para el posicionamiento de los electrodos, estos deben ser ubicados haciendo planos transversal y mediano a intervalos de 10 y 20 por ciento [15]. El 10 hace referencia al diez por ciento de la distancia real entre electrodos adyacentes, mientras que el 20 se refiere al veinte por ciento de la distancia total delantera/trasera o derecha/izquierda, de tal forma que la zona en donde cada electrodo registra las señales electroencefalográficas pueda ser comparada con los registros realizados a otros individuos [12].

Cada uno de los electrodos para el registro de EEG se encuentran marcados con números y letras que indican las diferentes regiones del cerebro en donde deben ser ubicados. Las letras representan el lóbulo cerebral (frontal, parietal, temporal y occipital), mientras que los números representan el hemisferio cerebral. Los números pares representan los electrodos que son ubicados sobre el hemisferio derecho y los números impares se encuentran relacionados a los electrodos que van ubicados en el hemisferio izquierdo. La letra "F" representa el lóbulo frontal, "P" representa el lóbulo parietal, "T" representa el lóbulo temporal, "O" representa el lóbulo occipital, "C" representa la región central, "A" representa la región auricular (lóbulo de la oreja), y una letra minúscula "z" representa los electrodos de la línea media [15]. En la tabla 2.1, se observa la distribución de los electrodos según lo establecido en la normativa 10-20.

| Área Cerebral | Hemisferio Izquierdo | Central | Hemisferio Derecho |
|--------------------------------|----------------------|---------|--------------------|
| Frontopolar | FP1 | - | FP2 |
| Frontal | F3 | FZ | F2 |
| Fronto Temporal | F7 C3 | CZ | F8 C4 |
| Medial temporal y parietal | T3 P3 | PZ | T4 P4 |
| Posterior temporal y occipital | T5 O1 | - | T6 O2 |

Tabla 2.1: Tabla de la distribución de los electrodos según la normativa 10-20.

Fuente: García, 2011

En la figura 2.4 se puede ver gráficamente el posicionamiento de cada uno de los electrodos de acuerdo con la normativa 10-20 para el posicionamiento de los electrodos.

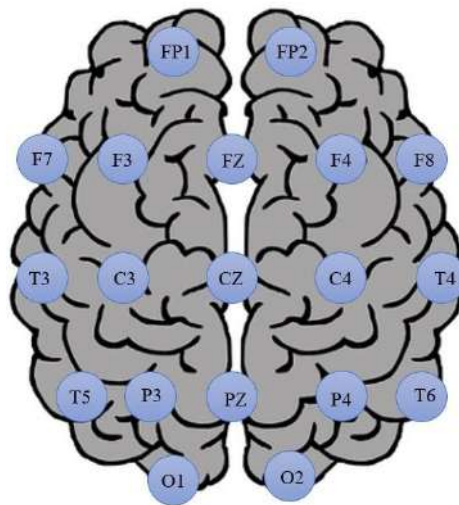


Figura 2.4: Sistema internacional 10-20 para el posicionamiento de los electrodos.

Fuente: Autor.

2.1.6 Tipos de potenciales EEG

El EEG representa la actividad eléctrica del cerebro según lo registrado por los electrodos colocados en el cuero cabelludo siguiendo la normativa anteriormente mencionada [17]. Esta actividad eléctrica proporciona información sobre dos tipos de actividad cerebral:

- **La actividad cerebral Espontanea:** La cuál se encuentra relacionada directamente con los procesos cognitivos (memorización) o estados mentales (cansancio, fatiga, etc.).
- **Potenciales relacionados a eventos:** Los Potenciales Relacionados con Acontecimientos Discretos (PRADs) o también denominados Potenciales Relacionados con Eventos (PREs) [18], se utilizan en la práctica clínica y en la investigación sobre el análisis de la actividad visual, auditiva, somatosensorial y otras funciones cognitivas de más alto nivel [19].

2.1.7 Ritmos Cerebrales

Mediante técnicas avanzadas de procesamiento de señales se pueden extraer características importantes del EEG que pueden estar relacionadas a los diferentes estados psicológicos o fisiológicos de una persona. Los patrones del EEG son muy importantes para comprender las actividades cerebrales al identificar las características morfológicas o examinar las bandas de frecuencia asociadas con diferentes actividades mentales. En la figura 2.5 se puede observar como las señales son adquiridas mediante un electroencefalograma y luego mediante procesamiento de señales se pasan del dominio del tiempo al dominio de la frecuencia, en donde son clasificadas en distintas bandas de frecuencia.

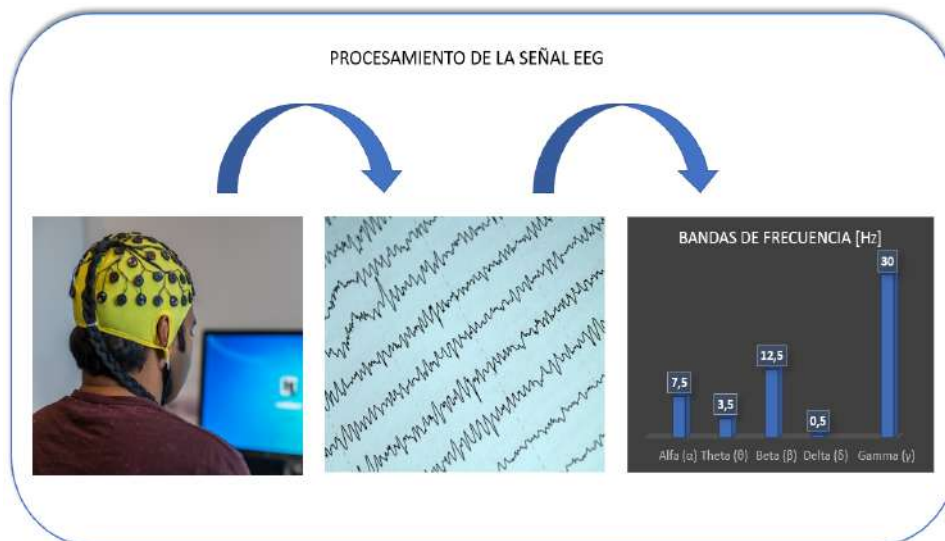


Figura 2.5: Procesamiento de la señal EEG para clasificarla por bandas de frecuencia.

Fuente: Autor.

Las bandas de frecuencia se pueden dividir de la siguiente manera:

- **Alfa:** Aparecen espontáneamente en adultos normales durante la vigilia, en condiciones de relajación e inactividad mental. Se puede visualizar de forma más clara en la zona occipital del cerebro cuando la persona mantiene los ojos cerrados [20].
- **Theta:** Las ondas theta se han asociado con el acceso a material inconsciente, inspiración creativa y meditación profunda. Esta suele ir acompañada de otras frecuencias y en ocasiones se encuentra relacionada con el nivel de excitación.
- **Beta:** Es el despertar habitual del cerebro asociado con el pensamiento activo, atención activa y enfoque en el mundo exterior, y se encuentra en adultos normales. Una onda beta de alto nivel puede aparecer cuando una persona se encuentra en estado de pánico [21].
- **Delta:** Estas ondas están directamente relacionadas con la etapa de sueño profundo. Están presentes en niños menores de 1 año y en adultos en un estado de sueño profundo. Debido a su baja frecuencia, las ondas delta se confunden fácilmente con señales de artefactos causadas por los músculos grandes del cuello o la mandíbula. Su frecuencia varía de 0.1 Hz a 3.5 Hz.
- **Gamma:** Son las ondas más rápidas del cerebro, se asocian con una mayor actividad mental (no son los pensamientos normales o resolución de problemas) y pueden incluir destellos de brillantez y explosiones repentinas de percepciones/intuiciones y momentos de extrema atención y concentración [22].

| Banda Espectral | Rango [Hz] | Amplitud [μ V] |
|-----------------|------------|---------------------|
| Alfa | 7.5 - 12.5 | 50 |
| Theta | 3.5 - 7.5 | 20 - 100 |
| Beta | 12.5 - 30 | 15 -50 |
| Delta | 0.5 - 3.5 | > 50 |
| Gamma | 30 - 60 | - |

Tabla 2.2: Amplitud y rango de frecuencia de cada banda espectral.

Fuente: Autor.

2.2 Interfaz Cerebro-Computador (BCI)

Las interfaces cerebro-computador vinculan las respuestas neuronales cognitivas humanas con el mundo exterior para traducir las intenciones y pensamientos del usuario en señales de control de dispositivos en tiempo real [15]. En la figura 2.6 se puede observar un esquema general del funcionamiento de una BCI. Los sistemas BCI se componen de varios bloques de procesamiento de señal que trabajan de forma consecutiva. Sobre el usuario se ubican unos electrodos de forma estratégica (normativa 10 – 20) que captan la actividad eléctrica generada por el cerebro en cada momento. Esta señal es amplificada y preprocesada con el fin de eliminar el ruido generado por los diferentes artefactos. Cuando la señal ya se encuentra libre de ruidos e interferencias, pasa a la etapa de aprendizaje automático, en donde se extraen las características de la señal EEG obteniendo vectores de datos y seguidamente, se clasifican estos vectores de datos obteniendo una predicción. La predicción del clasificador puede ser utilizada como señal de realimentación mostrándola al usuario a través del control de diversas aplicaciones [23].

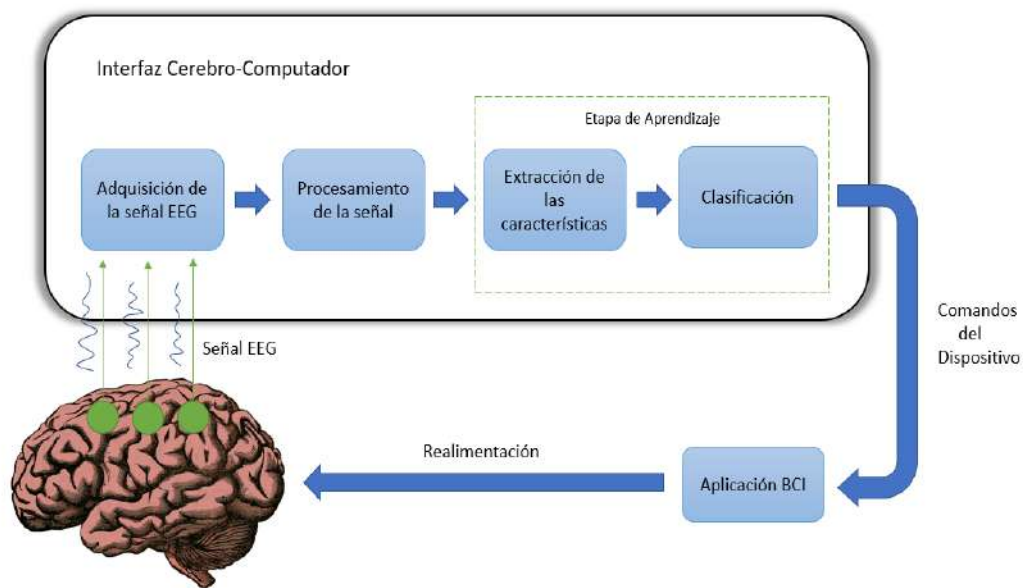


Figura 2.6: Esquema general de una Interfaz Cerebro-Computador (BCI).

Fuente: Autor.

2.2.1 Tipos de BCI

Una interfaz Cerebro-Computador puede monitorizar la actividad cerebral de tres formas diferentes, estas se clasifican de la siguiente manera:

- **Invasivas:** En los sistemas BCI de tipo invasivas, los sensores son implantados dentro del cuerpo humano mediante una intervención quirúrgica. Esto permite que las señales electroencefalográficas obtenidas sean de mayor calidad y sin ningún tipo de ruido o interferencia. Pero presentan la desventaja de que con el tiempo la adquisición de la señal se va debilitando, ya que el cuerpo va generando un rechazo ante los objetos extraños que se encuentra dentro del organismo.
- **Parcialmente Invasivas:** En los sistemas BCI parcialmente invasivos, los sensores se implantan dentro del cráneo mediante cirugía, pero sin invadir la materia gris, por lo cual se sigue obteniendo una señal de mejor calidad que en los sistemas BCI de tipo no invasivos, con la ventaja de que el riesgo de rechazo por parte del organismo del usuario es mínimo comparado con el caso anteriormente mencionado.
- **No Invasivas:** La mayoría de investigaciones publicadas sobre BCI de los últimos años se encuentran relacionadas con las interfaces BCI de tipo no invasivas, debido a su fácil adquisición y bajo costo, además de que no representan ningún tipo de riesgo para los usuarios, ya que no se requiere de ningún tipo de intervención quirúrgica para el implante de los sensores. Sin embargo, este tipo de tecnologías no cuenta con una alta resolución y en la mayoría de ocasiones se ven afectas por ruidos e interferencias. Aunque cabe resaltar, que en los últimos años se ha evidenciado avances significativos utilizando este tipo de dispositivos, como, por ejemplo, en donde un usuario es capaz de controlar mentalmente brazos robóticos o sillas de ruedas utilizando sus pensamientos [24].

2.2.2 Clasificación de los sistemas BCI basados en la respuesta Cerebral

Los BCI se pueden agrupar en tres categorías: reactivo, activo y pasivo; basado en cómo se genera la respuesta cerebral [25].

- **Reactivo:** El uso de estímulos sensoriales para inducir cambios predecibles en el EEG crea un paradigma de control denominado "reactivo" el cuál depende de los parámetros del estímulo sensorial, los cuales pueden estar relacionados a estímulos visuales, auditivos o táctiles [26]. En general, los paradigmas reactivos son sistemas síncronos, donde el desempeño del sistema está sujeto directamente a los estímulos que se presentan en el momento. Una de las desventajas que presentan este tipo de sistemas, es que no permiten realizar un control completamente instantáneo.
-

- **Activo:** A diferencia de los paradigmas anteriormente mencionado, el recrear imágenes o movimientos en la mente del usuario de forma intencional, crea un nuevo paradigma de control denominado “activo”, el cuál consiste en que los cambios en las señales EEG se encuentran relacionados directamente a acciones cognitivas endógenas, es decir, acciones cognitivas que se generan a voluntad. Un ejemplo de este tipo de paradigma, es cuando un usuario imagina los movimientos de la mano izquierda o derecha para mover el cursor de un ordenador hacia la izquierda o derecha de manera respectiva [27].
- **Pasivo:** El concepto de BCI pasivos se basa en que el usuario no puede modular a voluntad las señales EEG para ejercer un control directo y preciso de los dispositivos. Sin embargo, en la mayoría de los escenarios en donde se utiliza este paradigma, el análisis de BCI se encuentra estrechamente relacionada con paradigmas activos / reactivos, por lo cual algunas de las técnicas descritas anteriormente pueden ser aplicables [28].

2.2.3 Tipos de respuestas de EEG para el control con BCI

Para lograr diferentes modos de control, las respuestas EEG pueden clasificarse en respuestas transitorias y continuas.

- **Respuesta Transitoria:** Las respuestas transitorias son cambios instantáneos en el EEG que pueden asociarse con una transición de un estímulo sensorial (reactivo), o una transición en imágenes mentales o estado cognitivo, como un agarre con una sola mano (activo). Tales respuestas transitorias son muy adecuadas para realizar selecciones discretas en aplicaciones como la tipificación BCI.
 - **Respuesta Continua** Las respuestas continuas representan la actividad EEG en curso. En paradigmas reactivos, esto se encuentra asociado con estímulos sensoriales repetitivos que, producen patrones repetitivos de “estado estacionario” en el EEG. En paradigmas activos, las imágenes mentales continuas o el arrastre del estado cognitivo producen cambios sostenidos en el EEG que, en algunos casos, también puede producir modulación gradual con entrenamiento para permitir un control más preciso. Dichas respuestas continuas son adecuadas para aplicaciones de control dimensional continuo como el control de un cursor o de una silla de ruedas [29].
-

2.2.4 Dispositivos BCI más relevantes del mercado

Actualmente el desarrollo de BCI se está convirtiendo en uno de los caminos más innovadores en el campo de la automatización moderna, robótica e informática. Vale la pena mencionar que numerosos inversores privados que contribuyen al progreso en este campo como empresas de videojuegos o de entrenamiento están involucrados en el desarrollo de tecnologías relacionadas con BCI, lo que ha permitido que al día de hoy este tipo de dispositivos sean fáciles de adquirir a un bajo costo. A continuación, se presentan algunos de los sistemas BCI más relevantes del mercado [30].

- **Neurosky MindWave Mobile 2:** Es un auricular EEG inalámbrico de bajo costo para el consumidor que cuenta con un electrodo seco ubicado en la posición FP1 según el sistema Internacional 10-20 para grabar la señal EEG del cuero cabelludo de las personas y cuenta con un electrodo de referencia para voltaje de tierra. Este dispositivo permite obtener diferentes valores de datos como datos EEG sin procesar, bandas de frecuencia (alfa, beta, delta y gamma) y calidad de la señal etc. El auricular se emplea con el chip ThinkGear ASIC Module (TGAM) que procesa previamente los datos dentro del auricular [31]. Además, también cuenta con un kit de desarrollo de software (SDK) que brinda la posibilidad de desarrollar aplicaciones de Interfaz Cerebro-Computador (BCI) para diferentes plataformas (Windows, Mac y Android).



Figura 2.7: Neurosky MindWave Mobile 2 de Neurosky.

Fuente: Paszkiel.

- **Emotiv Epoc:** El Emotiv EPOC es un dispositivo captador de ondas cerebrales de bajo costo que cuenta con 14 canales para la adquisición de señales EEG y adicionalmente posee un giroscopio para control bidimensional. Los electrodos se encuentran ubicados en las posiciones AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 según el sistema internacional 10-20. Además, cuenta con dos electrodos de referencia ubicados justo por encima de las orejas (uno para el hemisferio izquierdo y el otro para el hemisferio derecho de la cabeza). El Emotiv EPOC cuenta con una frecuencia de muestreo internamente de 2048 Hz y una frecuencia de muestreo de 128 Hz por canal. Los datos adquiridos mediante la diadema son enviados a una computadora a través de Bluetooth, utilizando un dispositivo USB con banda de frecuencia de 2.4 GHz. Se recomienda que antes de utilizar el dispositivo, todas las almohadillas de los electrodos en la parte superior sean humedecidos mediante una solución salina. Al igual que el Neurosky MindWave Mobile 2, el Emotiv Epoc también incluye un kit de desarrollo de software (SDK) de Emotiv que proporciona una funcionalidad de conteo de paquetes para garantizar que no se pierdan datos y la posibilidad de poder observar en tiempo real la calidad de contacto de cada uno de los electrodos. Además, también brinda la posibilidad de que los programadores puedan desarrollar sus propias aplicaciones para el control de diversos dispositivos [32].



Figura 2.8: Emotiv Epoc de Emotiv Inc.
Fuente: Stach et al.

- **Emotiv Insight:** Emotiv Insight fue el segundo invento lanzado al mercado por Emotiv para satisfacer la necesidad diaria de controlar la salud cognitiva a un precio asequible. Cuenta con un diseño sofisticado, liviano, instintivo y ergonómico, que brinda una gran comodidad a los usuarios. Este dispositivo cuenta con 5 sensores ubicados en las posiciones AF3, AF4, T7, T8, PZ siguiendo la normativa 10-20 para el posicionamiento de electrodos EEG y cuenta con 2 sensores de referencia. Los datos adquiridos mediante la diadema son enviados a una computadora a través de Bluetooth 4.1 de baja energía denominado bluetooth Smart. El Emotiv Insight ofrece una alta resolución espacial que garantiza una visión más profunda de la señal cerebral, siendo esta de 128 Hz por cada canal. Cuenta con sensores de movimiento para registrar la ubicación de la cabeza con gran precisión permitiendo múltiples grados de libertad para aplicaciones de control [34]. Además, permite medir hasta 6 métricas emocionales: enfoque, estrés, emoción, relajación, interés y compromiso. Al igual que el Neurosky MindWave Mobile 2 y el Emotiv Epoc, el Emotiv Insight es capaz de proporcionar un flujo de datos EEG sin procesar y cuenta con un kit de desarrollo de software (SDK) que brinda la posibilidad de desarrollar software en diversos lenguajes de programación (Python, C++, Matlab, C y Java) para diversas plataformas (Windows, Android y Mac) [35].



Figura 2.9: Emotiv Insight de Emotiv Inc.

Fuente: Paszkiel.

2.2.5 Comparativa de dispositivos BCI comerciales

Con el fin de poder seleccionar el dispositivo BCI más adecuado para el desarrollo de este trabajo, es necesario estudiar las propiedades de cada uno de los dispositivos comerciales anteriormente mencionados. En la tabla 2.3, se pueden observar algunas de las características más relevantes de cada uno de ellos, como: usos comunes, cantidad y tipo de electrodos, posicionamiento de los electrodos de acuerdo a la normativa 10-20, conectividad, compactibilidad y tasa de muestreo.




| Dispositivo | Usos Comunes | Electrodos | Conectividad | Compatibilidad | Tasa de muestreo |
|---|--|--|---|----------------------------|--|
|  Neurosky MindWave | Salud, educación, investigación de mercado, entretenimiento. | 1 de referencia. 1 electrodo situado en la parte frontal (FP1). Electrodo seco. | USB e inalámbrica. | Android/iOS, Windows, Mac. | 512 Hz |
|  Emotiv Epoc | Investigación, entretenimiento, neurorehabilitación. | 2 de referencia. 14 electrodos (AF3, AF4, F3, F4, FC5, FC6, F7, F8, T7, T8, P7, P8, O1, O2). Electrodo húmedo. | Inalámbrica, USB solo con un dispositivo extra. | Android/iOS, Windows, Mac. | 128 o 256 muestras por segundo (2048Hz interno). |
|  Emotiv Insight | Investigación, bienestar cognitivo, neurorehabilitación. | 2 de referencia. 5 electrodos (AF3, AF4, T7, T8, PZ). Electrodo semiseco. | Inalámbrica, USB solo con un dispositivo extra. | Android/iOS, Windows, Mac. | 128 muestras por segundo en cada canal. |

Tabla 2.3: Comparativa de las características físicas de cada uno de los dispositivos BCI.

Fuente: Villegas Méndez et al.

2.3 La mano humana

La mano humana es uno de los sistemas mecánicos más complejos y versátiles que existen [36]. Su destreza le permite realizar movimientos de gran precisión, como por ejemplo el agarre de pinza fina, el cual consiste en sujetar objetos utilizando el dedo pulgar y el índice, a modo de pinza.

Debido a que la mano cuenta con un gran número de grados de libertad, una alta relación de fuerza/peso y un sistema sensorial, el ser humano es capaz de relacionarse y desenvolverse dentro de su entorno [37].

La mano se encuentra conformada por la palma y el dorso, junto con cinco dedos denominados pulgar, índice, corazón, anular y meñique (ver figura 2.10). Cada uno de los dedos posee dos grados de libertad en la base a excepción del pulgar que cuenta con cinco grados de libertad y dos articulaciones que le permiten realizar los movimientos de flexión y extensión. Además, la palma cuenta con otros grados de libertad que le permiten curvar la superficie de la palma, permitiéndole así adaptarse a la forma de ciertos objetos (esféricos, cilíndricos, etc.) [37].



Figura 2.10: Partes de la mano humana.

Fuente: Autor.

2.3.1 La mano y la mente Humana

El ser humano utiliza sus manos para desarrollar un sin número de actividades, en pocas palabras todo lo que la mente es capaz de imaginarse, por eso, en ocasiones se menciona que la mano humana es donde la mente encuentra al mundo.

El cerebro humano, es una de las principales razones por las cuáles es posible diferenciar la mano humana de las otras especies que también poseen manos de cinco dedos. Esto se debe a la sincronización que existe entre el cerebro y la mano, permitiéndole al ser humano comunicarse y relacionarse con su entorno.

Una de las principales funciones que proporciona la mano, es su sistema sensorial, el cuál le permite controlar e interpretar información que recibe de su entorno, como por ejemplo las nociones de espesor, distancias, peso, superficie, temperatura, etc.

La mano le permite al ser humano realizar actividades como escribir o sujetar objetos con una determinada fuerza. Esta habilidad es controlada por el hemisferio cerebral del lado contrario, es decir, una persona que realiza actividades a diario utilizando su mano derecha, predomina su hemisferio cerebral izquierdo y una persona que realiza la mayor parte de sus actividades con la mano izquierda, predomina el hemisferio cerebral derecho (figura 2.11), siendo esto un rasgo personal de cada ser [38].

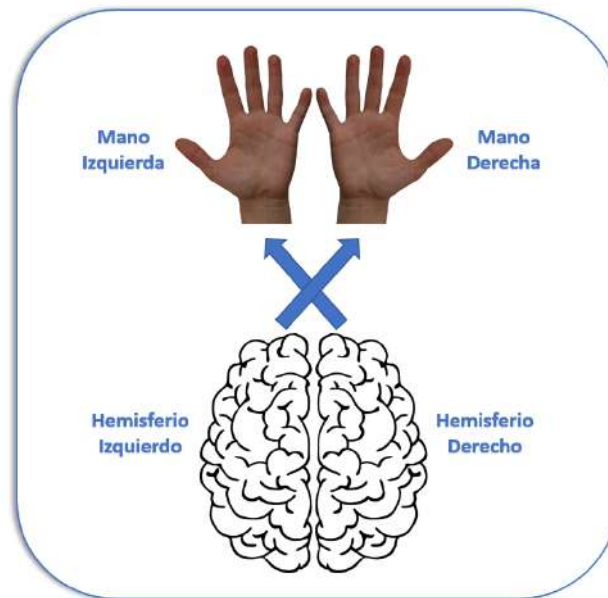


Figura 2.11: Predominio de los hemisferios del cerebro en las manos.

Fuente: Autor.

2.3.2 Anatomía de la mano humana

Para comprender el funcionamiento de la mano humana, es necesario analizar primero su anatomía, es decir cada uno de sus músculos y huesos [39].

2.3.3 Huesos de la mano

La mano humana está compuesta por 14 huesos largos conocidos como falanges: dos en el pulgar y tres en cada uno de los cuatro dedos restantes. En el caso del pulgar, este no cuenta con la falange media. El orden de los falanges para los cuatro dedos a excepción del pulgar, es el siguiente: los falanges ubicados en las puntas de los dedos son conocidos como falanges distales, los falanges ubicados en la zona intermedia de los dedos se conocen como falanges media y los falanges ubicados en el extremo inferior se conocen como falanges proximales. Estos son unidos a la epífisis inferior de los huesos metacarpianos, los cuáles componen la palma de las mano. La unión entre las epífisis inferiores de los metacarpianos con las falanges proximales forman los nudillos de las manos. Así mismo, cada epífisis superior de los huesos metacarpianos se articula con la fila inferior de los huesos carpianos, los cuales son 8 huesos que conforman la muñeca; entre ellos existen una serie de canales por donde se deslizan los tendones; encontrándose organizados en dos hileras, una proximal y otra distal [39]. La organización de cada hueso que compone el carpo es la siguiente:

Hilera Proximal:

- Hueso escafoides
- Hueso semilunar
- Hueso piramidal
- Hueso Pisiforme

Hilera Distal:

- Hueso Trapecio
 - Hueso Trapezoide
 - Hueso grande
 - Hueso Ganchoso
-

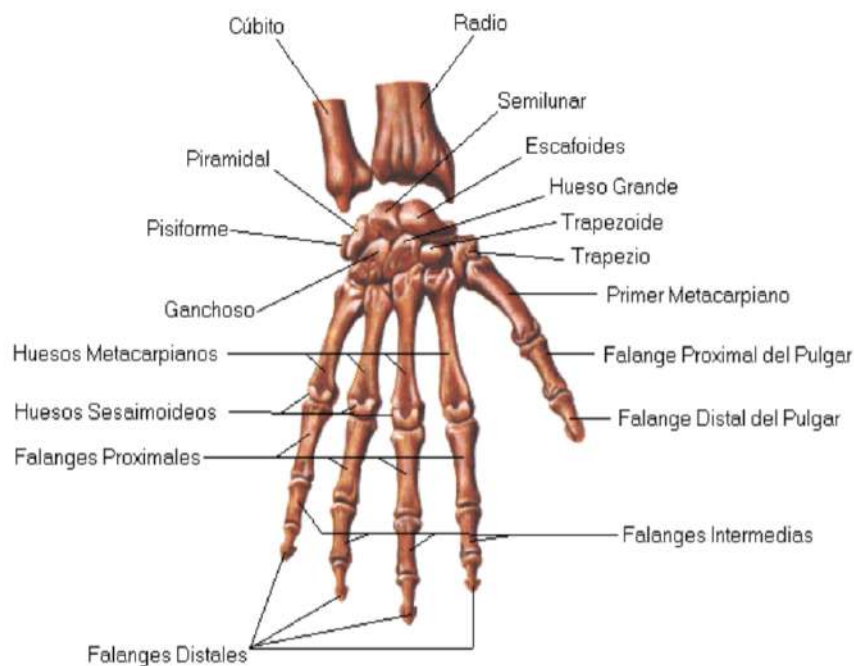


Figura 2.12: Huesos que constituyen la mano humana.

Fuente: Burgos and Albán

2.3.4 Músculos de la mano

El músculo tiene la característica de ser contráctil, teniendo como función generar fuerza en movimiento en colaboración de los huesos y tendones. Los músculos de la mano están formados por los extrínsecos e intrínsecos.

- **Músculos extrínsecos:** Estos se originan en el antebrazo y se insertan en la mano, son los que desempeñan una función de agarre potente, y están formados por los flexores extrínsecos y extensores extrínsecos.
- **Músculos intrínsecos:** Se localizan en la mano y proporcionan esencialmente movimientos de precisión (agarre preciso) entre los dedos y el pulgar, y están formados por el aductor del pulgar, los interóseos, los músculos tenar, los músculos de eminencia hipotenar, el palmar corto y los lumbruales.

Los músculos y tendones existentes en la mano humana son los encargados de realizar los movimientos de los dedos y de que vuelvan a su posición inicial y lo más importante que otorgan estabilidad a los dedos [38].



Figura 2.13: Músculos de la mano humana.

Fuente: Guaricela et al.

2.3.5 Tipos de agarres de presión

Debido a la compleja organización anatómica de la mano humana, esta es capaz de lograr diferentes tipos de agarres. En el año 1919, Schlesinger fue el primero en desarrollar una clasificación de agarres para la mano humana, la cual dividió en 6 principales categorías: agarre cilíndrico, de precisión, de gancho, con las yemas, esféricos y laterales [41]. Los cuales se pueden observar en la figura 2.14.



Figura 2.14: Clasificación de agarres definidos por Schlesinger.

Fuente: Ceres et al.

Tiempo después Kapandji clasificó los agarres de la mano humana en tres categorías diferentes denominadas: presión digital, presión palmar y presión central [42]. En la figura 2.15 se presentan las categorías anteriormente mencionadas y cada una de las subcategorías que la componen.

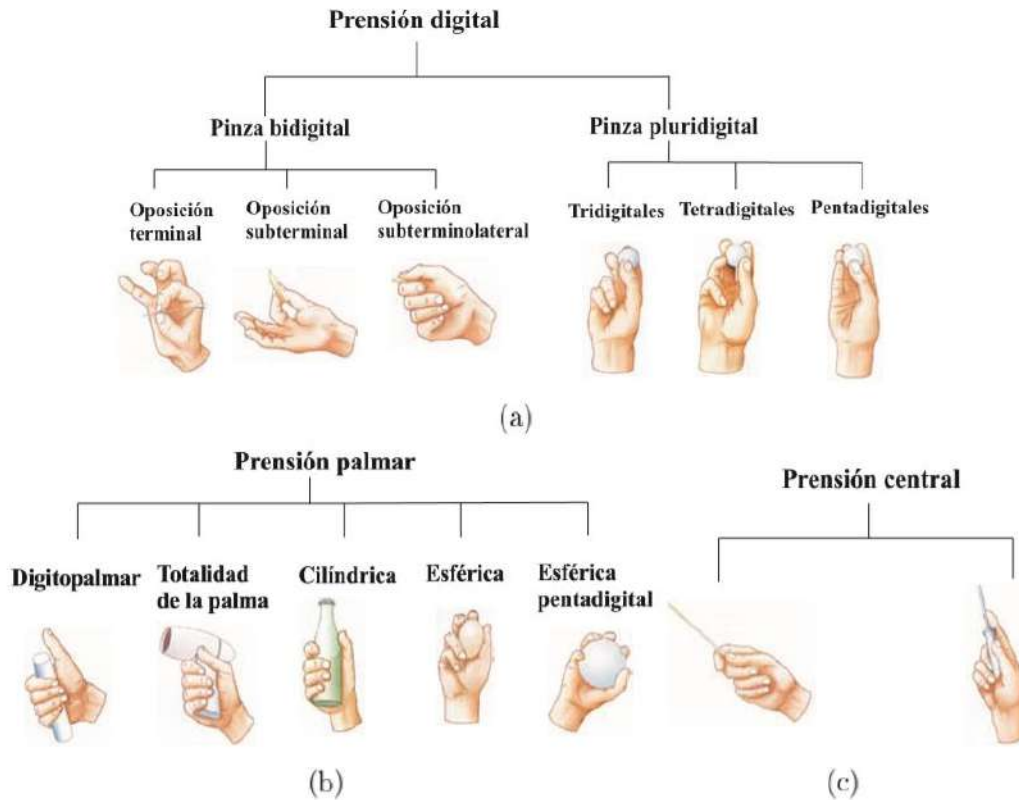


Figura 2.15: Esquema de clasificación de agarres definidos por Kapandji.

Fuente: Guzman

A continuación se describen algunos de estos agarres.

- **Pinza Bidigital:** Se refiere a la clásica pinza pulgar-índice y se divide en tres tipos: oposición terminal, subterminal y subterminolateral. Los cuáles son agarres que no requieren de la intervención de la gravedad para ser ejecutados y se utilizan para sujetar elementos con gran precisión como un lápiz o una hoja de papel.
- **Pinza Pluridigital:** Son presas de menor precisión, pero con mayor firmeza. Se pueden subdividir en tridigital (cuando se utilizan tres dedos), tetradigital (para cuatro dedos) y pentadigital (para cinco dedos). Estos tipos de agarres permiten la sujeción con gran firmeza de objetos muy gruesos.

- **Prensión Palmar:** En este tipo de agarre intervienen tanto los dedos como la palma de la mano. Se puede subdividir en diferentes categorías dependiendo de la forma y del diámetro del objeto que se desea sujetar. Dígito palmar (objetos con un diámetro de 3 a 4 cm), Totalidad de la palma (Objetos pesados y voluminosos), Cilíndrica (a mayor diámetro disminuye su firmeza).
- **Prensión Central:** Este tipo de prensa requiere la flexión de los tres últimos dedos, extensión del índice y mínima oposición del pulgar. Permite el agarre de objetos alargados y el dedo índice es el encargado de dar la orientación para dirigir el utensilio [43].

2.3.6 Amputación de miembro superior

Una amputación es una condición adquirida cuyo resultado es la pérdida de una extremidad, la causa puede estar relacionada a una malformación genética o un accidente sufrido. En la mayoría de los casos la pérdida de un miembro del cuerpo puede generar una afectación emocional o traumática en el paciente, ya que esto altera su imagen estéticamente. Además el amputado debe someterse a un proceso de recuperación, que le permita aprender a adaptar a los diferentes entornos sociales, laborales y familiares [40].

Dependiendo del nivel de amputación, se puede distinguir los siguientes tipos:

- Amputación transhumeral (por encima del codo).
- Desarticulación del codo.
- Desarticulación de la muñeca.
- Amputación parcial de la mano.

2.3.7 Prótesis de mano

Una prótesis es un dispositivo capaz de sustituir o reemplazar una parte del cuerpo faltante, ya sea por malformaciones genéticas o por un accidente sufrido.

El diseño de las prótesis de mano se encuentra centrado actualmente en imitar la anatomía de la mano humana, con el fin de que los agarres ejecutados por las prótesis puedan coincidir con la geometría de los objetos y así poder ejecutar tareas complejas [44].

2.3.8 Evolución en el desarrollo de prótesis de mano

Desde siglos atrás el ser humano ha buscado la manera de sustituir los miembros que ha perdido, debido a accidentes o malformaciones genéticas, por un dispositivo artificial, que le permita realizar los movimientos de la parte faltante y así poder suplir las funciones de las que carece.

Durante el siglo XX uno de los objetivos principales era lograr que los amputados regresaran a la vida laboral. Este objetivo fue alcanzado por el médico francés Gri-poulleau, el cual diseñó diferentes dispositivos capaces de ser utilizados para sustituir el miembro faltante. En el año 1912, Dorrance, desarrolló un dispositivo denominado Hook el cuál permitía a los usuarios ejecutar funciones como el de abrir y cerrar un gancho, mediante movimientos realizados por la cintura escapular. Años después aparecen las prótesis controladas por señales mioeléctricas, las cuales tienen su origen en Rusia en la década de los 60. Su funcionamiento se basaba en la interpretación de las señales eléctricas que eran extraídas de los músculos del muñón. Dichas señales eran amplificadas y procesadas por un dispositivo electrónico, que permitía determinar qué acciones debía realizar el dispositivo protésico. Con el transcurso de los años este tipo de prótesis ha mejorado de forma considerable, llegando al punto de permitirle a usuario controlar de forma independiente cada uno de los dedos de una prótesis de mano.

Hoy en día se pueden encontrar diferentes tipos de prótesis en el mercado, las cuales varían según el tipo de control que utilizan. Los países que actualmente cuentan con un gran avance tecnológico en el desarrollo de prótesis son: Estados Unidos, Alemania, Francia y Japón [45].

2.3.9 Clasificación de las prótesis de miembro superior

Las prótesis de miembro superior pueden ser clasificadas en dos grupos: pasivas y activas. Las primeras son conocidas también como prótesis cosméticas debido a que están diseñadas para imitar únicamente la estética de la mano humana [46].

- **Prótesis pasivas:** Conocidas también como prótesis cosméticas o estéticas, debido a que no permiten realizar ningún tipo de movimiento, sino que están diseñadas especialmente para imitar el aspecto estético de la mano humana. Este tipo de prótesis son fabricadas en materiales como: PVC rígido, látex flexible o silicona, debido a que deben ser lo más livianas posibles. Una de las principales ventajas que presentan este tipo de prótesis es que, al contar con pocas piezas, su mantenimiento es sencillo de realizar en comparación con otros tipos de prótesis.
 - **Prótesis activas:** A diferencia de las prótesis anteriormente mencionadas, estas sí permiten realizar movimientos y su objetivo principal consiste en simular la naturaleza de la mano humana, con el fin de poder suplir funciones de las que
-

carece el amputado. Este tipo de prótesis se clasifican en mecánicas, eléctricas, neumáticas y mioeléctricas, siendo las últimas las que ofrecen un mejor resultado.

En la siguiente imagen se muestra una prótesis de tipo activa y pasiva.



Figura 2.16: Prótesis activa y pasiva.

2.3.10 Prótesis Mecánicas

Las prótesis mecánicas, son prótesis con dispositivos de apertura y cierre mediante cables y arneses unidos al cuerpo del usuario, las cuales se abren o cierran a voluntad por la tracción ejercida por el tensor. Una de las principales desventajas de este tipo de prótesis, es que están limitadas al número de movimientos que permite realizar, ya que requieren de energía propia y movimientos de tensión para su funcionamiento. El sistema suele estar constituido por una mano o un gancho, dependiendo de las necesidades del usuario.

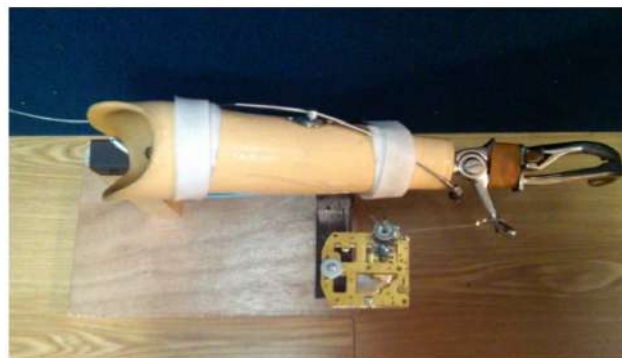


Figura 2.17: Prótesis mecánica.

Fuente: Castañeda et al.

2.3.11 Prótesis eléctricas

Este tipo de prótesis utilizan como fuente principal de energía actuadores como motores, los cuáles son ubicados en la muñeca o codo junto con un sistema de batería recargable. Las prótesis eléctricas pueden ser controladas mediante un sistema de servo control, un pulsador o botón con interruptor de arnés. En algunas ocasiones se utiliza un socket que es un dispositivo intermedio entre la prótesis y el muñón para lograr un mejor desempeño. Generalmente suelen ser costosas, pesadas y delicadas. En la figura 2.18 se puede observar la *MyoHand* el cuál es un dispositivo desarrollado por *Ottobock* que permite realizar agarres con gran precisión y fuerza.



Figura 2.18: Prótesis de mano eléctrica MyoHand.

Fuente: Rivera

2.3.12 Prótesis neumáticas

Las prótesis neumáticas utilizan aire comprimido como fuente de energía para su funcionamiento. Algunas de sus principales ventajas es que permiten realizar movimientos con gran velocidad y fuerza. Actualmente no son tan comercializadas debido que cuentan con elementos grandes y bastante complejos que dificultan su mantenimiento. En la figura 2.19 se puede observar la prótesis neumática *Shadow* [48].



Figura 2.19: Prótesis de mano neumática.
Fuente: Brito et al.

2.3.13 Prótesis mioeléctricas

Hoy en día las prótesis controladas por señales mioeléctricas son las más utilizadas y comercializadas, debido a las características que estas proporcionan (aspecto más natural, fuerza de agarre, velocidad de respuesta).

Tal como lo indica su nombre, las prótesis mioeléctricas son controladas por medio de contracciones musculares que emite el usuario a voluntad. Mediante el uso de electrodos de tipo superficial se puede leer la señal eléctrica generada por la flexión o contracción de los músculos, dicha señal eléctrica se encuentra en el rango de los 5 a 20 μV [46]. Posteriormente esta señal es amplificada, filtrada y procesada por un microcontrolador, el cuál es el encargado de ejecutar unas determinadas acciones como cerrar o abrir la mano. Algunas de las desventajas que presentan este tipo de prótesis es su elevado costo y que requieren de un sistema de batería [49].

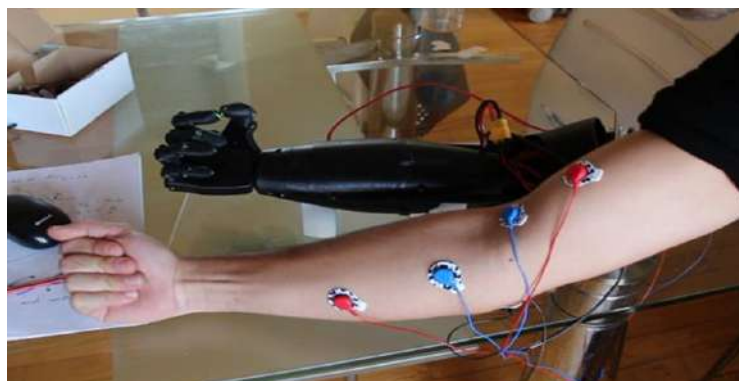


Figura 2.20: Prótesis mioeléctrica.
Fuente: Mahdi

3 Metodología

3.1 Componentes

En esta sección se describe el proceso de selección de los componentes electrónicos utilizados para el desarrollo del trabajo. Teniendo en cuenta sus características, precio y accesibilidad.

3.1.1 Servomotor

Un servomotor es un tipo especial de motor, compuesto por componentes electromecánicos y electrónicos. Está diseñado especialmente para desplazarse una determinada cantidad de grados y luego mantenerse fijo.

Los servomotores poseen en su interior un motor de corriente directa (DC), el cual tiene acoplado en su eje, una caja reductora, que le permite reducir su velocidad para aumentar su torque. Es importante resaltar que existen servomotores que utilizan engranajes y piñonerías de tipo metálicas o de plástico, lo cual varía según el fabricante. Otra parte fundamental de los servomotores, es su circuito de control, que permite al usuario realizar el control de posición del motor.

En la figura 3.1 se observa un servomotor Tower Pro MG995 de piñonería metálica con sus tres cables para su funcionamiento. Donde el cable de color rojo se encuentra asociado al voltaje de funcionamiento (VCC), el de color café, a tierra (GND) y el de color amarillo, a la modulación por ancho de pulso (PWM) [50].



Figura 3.1: Servomotor Tower Pro MG995.
Fuente: SigmaElectrónica.

En la tabla 3.1 se pueden observar las características técnicas de algunos servomotores de bajo costo disponibles en el mercado.

| Servomotor | Alimentación | Torque máximo | Tipo de piñonería | Precio (COP) |
|------------------------|--------------|---------------|-------------------|--------------|
| Tower Pro MG995 | 3.0V - 7.2V | 15 kg*cm | Metálica | \$20.000 |
| Hitec HS-485HB | 4.8V - 6.0V | 6 kg*cm | Metálica | \$80.000 |
| Hitec HS-422 | 4.8V - 6.0V | 4.1 kg*cm | Plástico | \$22.000 |

Tabla 3.1: Tabla comparativa de servomotores.
Fuente: SigmaElectrónica, Servocity, SuperRobótica.

Teniendo en cuenta las características expuestas para cada servomotor en la tabla 3.1, se determina que la mejor opción es el Tower Pro MG995, debido a que cuenta con un rango de alimentación mayor, un torque máximo de 15 kg*cm y con piñonería de tipo metálica (brinda mayor resistencia y durabilidad). Además de que su precio es bastante accesible.

3.1.2 Micro Servo

Los micro servos son servos miniaturas de reducidas dimensiones y bajo consumo. Al igual que los servo motores, el tipo de material empleado para la piñonería (metal o plástico) dependen directamente del modelo y fabricante. En la figura 3.2 se puede observar el micro servo Tower Pro MG90S de piñonería metálica junto con sus cables para el funcionamiento (amarillo alimentación positiva, café asociado al tierra y naranja asociado a la señal PWM) [54].



Figura 3.2: Micro Servo Tower Pro MG90S.
Fuente: geekfactory.mx.

En la tabla 3.2 se presentan las especificaciones técnicas, de algunos micro servos de bajo costo disponibles en el mercado.

| Micro servo | Alimentación | Torque máximo | Tipo de piñonería | Precio (COP) |
|-----------------|--------------|---------------|-------------------|--------------|
| Tower Pro MG90 | 3.0V - 7.2V | 1.6 kg*cm | Plástico | \$7.000 |
| Tower Pro SG92R | 3.0V - 6.0V | 1.6 kg*cm | Plástico | \$15.000 |
| Tower Pro MG90S | 4.0V - 7.2V | 10 kg*cm | Metálica | \$12.000 |

Tabla 3.2: Tabla comparativa de micro servos.

Fuente: geekfactory.mx, taloselectronics.

El micro servo Tower Pro MG90S es el más adecuado para el desarrollo del trabajo. Debido a que cuenta con piñonería de tipo metálica y un alto torque. Además, su precio no difiere mucho en relación a los otros servos anteriormente mencionados.

3.1.3 Placas de desarrollo WiFi

Para realizar el control de la mano robótica, es indispensable el uso de una placa de desarrollo con conexión Wifi, que cuente con dimensiones relativamente pequeñas. Esto con el objetivo de que el circuito electrónico diseñado, pueda estar dentro de la mano robótica, logrando así un aislamiento de la parte electrónica. A continuación, se describen algunas placas de desarrollo Wifi de bajo costo disponibles en el mercado.

- **Wemos D1 mini:** La tarjeta Wemos D1 mini (ver figura 3.3) es una tarjeta Wifi basada en el chip ESP8266, de tamaño compacta (25.6mm de ancho x 34.2mm de largo) y funciona con una tensión de alimentación de 3.3V. Posee 11 entradas y salidas digitales, 10 salidas PWM, 1 entrada analógica que soporta un máximo de 3.2V de entrada y su velocidad puede configurarse a 80 MHz o 160 MHz según se requiera. Algunos de los protocolos de comunicación que soporta esta placa son: I2C, TCP/IP y UDP. Además, puede ser programada en lenguajes como LUA o Arduino [56].



Figura 3.3: Wemos D1 mini.

Fuente: Llamas.

- **NodeMCU basada en el chip ESP8266:** La tarjeta NodeMCU actualmente posee diferentes variantes con respecto al chip o microcontrolador que utiliza. En este caso se abordará la versión con el chip ESP8266 (ver figura 3.4).

La NodeMCU ESP8266 es ampliamente utilizada para el desarrollo de aplicaciones IoT. Debido a que cuenta con características como: un módulo Wifi de 2.4 GHz, 1 entrada analógica (ADC) con resolución de 10 bits y 17 pines de entrada y salida GPIO, que pueden ser utilizadas para propósito general. Además, también soporta protocolos de comunicación como I2C, TCP/IP, MQTT y UDP; y puede ser programada en lenguajes como LUA o Arduino. Cabe resaltar, que a pesar de que esta placa es bastante compacta (24.5 mm de ancho x 49 mm de largo), es más grande en comparación con la placa Wemos D1 mini [57].



Figura 3.4: NodeMCU basada en el chip ESP8266.

Fuente: del Valle.

- **NodeMCU basada en el chip ESP32:** La placa de desarrollo NodeMCU ESP32 (ver figura 3.5) proporciona conexiones de tipo inalámbrica como Wifi, bluetooth y BLE. Además, cuenta con una unidad central de procesos (CPU) compuesta por dos núcleos, que puede llegar a funcionar a una frecuencia máxima de 240 MHz. Una de las grandes ventajas que proporciona esta placa de desarrollo, es la cantidad de periféricos con los que cuenta (sensor de efecto Hall, interfaz para tarjeta SD, sensores táctiles capacitivos, etc.). En relación a su tamaño, es un poco más grande a comparación con las placas anteriormente mencionadas (28 mm de ancho x 54.4 mm de largo) [58].

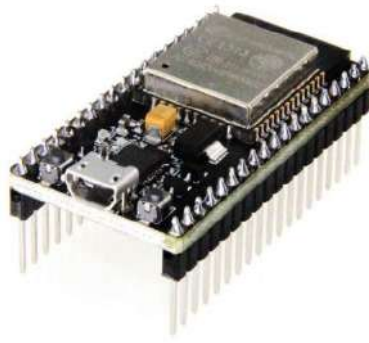


Figura 3.5: NodeMCU basada en el chip ESP32.

Fuente: Prometec.

| Característica | Wemos D1 mini | NodeMCU ESP8266 | NodeMCU ESP32 |
|---------------------|----------------|-----------------|---------------|
| Salidas PWM | 9 | 9 | 25 |
| Alimentación | 5V DC | 5V DC | 5V DC |
| Frecuencia de reloj | 80/160 Mhz | 80/160 Mhz | 160/240 Mhz |
| Tamaño | 25.6 x 34.2 mm | 24.5 x 49 mm | 28 x 54.4 mm |
| Precio (COP) | \$15.500 | \$18.000 | \$38.000 |

Tabla 3.3: Tabla comparativa de algunas placas de desarrollo con conexión WiFi.

Fuente: Llamas, del Valle, Prometec

Teniendo en cuenta la tabla comparativa 3.3, se determina que placa Wemos D1 mini es la más adecuada para el desarrollo del proyecto. Debido a que ofrece prestaciones similares a las demás placas, pero a un menor precio. Además, en cuestión de dimensiones resulta siendo las más pequeña y compacta.

3.2 Construcción de la mano robótica

La mano robótica con la cual se trabajó, fue tomada del proyecto *Humanoid Robotic Hand*, desarrollado por Ryan Gross [59]. Debido a que esta permite realizar una gran variedad de movimientos y aparte es de fácil replicación.

3.2.1 Diseño de piezas 3D

Se realizaron algunos ajustes al diseño de la mano robótica utilizando el programa Solid Works, el cual es un Software de diseño asistido por computadora que permite diseñar y modelar objetos 3D. Esto debido a que no se contaban con los componentes del diseño original. A continuación, se presentan algunos modelos 3D de las piezas que fueron rediseñadas.

- **Soporte para Servomotores:** Esta pieza fue adaptada para los servo motores Tower Pro MG995. En la figura 3.6, se puede observar el ensamble realizado en Solid Works 2019, que permite obtener una vista previa de como quedaría la pieza una vez sea impresa y ensamblada con los componentes.

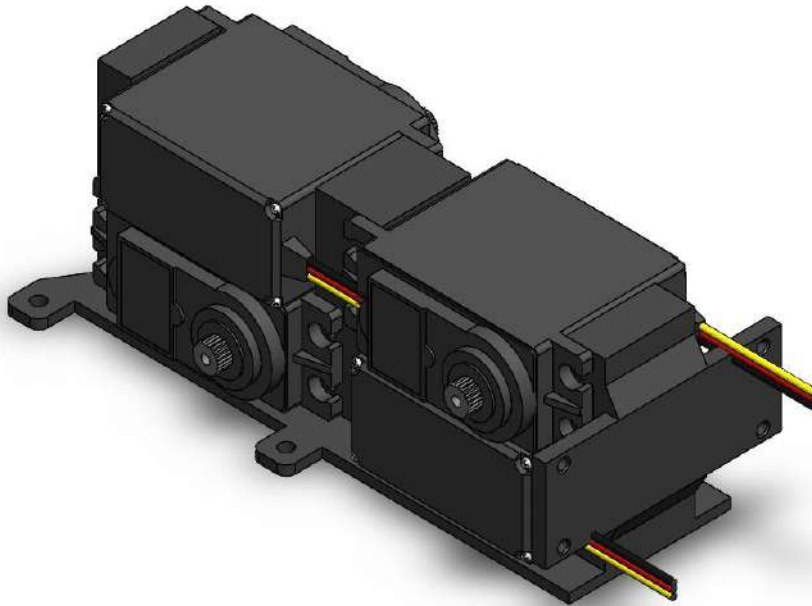


Figura 3.6: Modelo 3D del soporte para Servomotores.

Fuente: Autor.

- **Cubierta exterior para Servomotores:** Debido a que el diseño original con el que se trabajó, no contaba con una cubierta para los servo motores, fue necesario diseñarla, ya que su función principal consiste en proteger cada uno de los servo motores, de agentes externos que puedan llegar a dañarlos. En la figura 3.7 se puede observar el diseño final de esta cubierta.

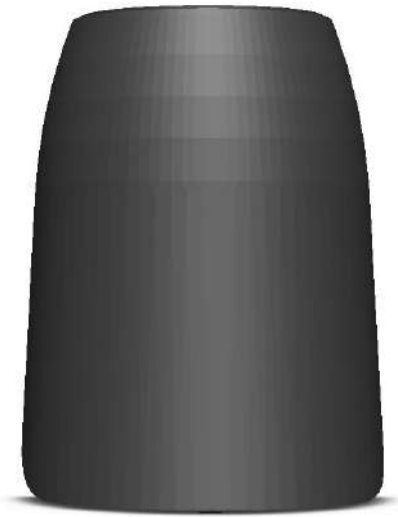


Figura 3.7: Modelo 3D de la cubierta exterior para los Servomotores.

Fuente: Autor.

- **Base para soporte de la mano robótica:** En la figura 3.8 se puede visualizar el diseño 3D para la base de la mano. La cuál cuenta con un pequeño corte, que permite el fácil acceso del cable de alimentación externa.



Figura 3.8: Modelo 3D de la base de la mano robótica.

Fuente: Autor.

3.2.2 Ensamble

El ensamble de la mano robótica fue realizado en el Software Solid Works 2019, con el fin de poder comprobar que todas las piezas encajaran sin problemas, además esto permitió obtener una mejor proyección sobre cómo se vería la mano robótica una vez está estuviera terminada. En la figura 3.9 se puede observar el ensamble.



Figura 3.9: Ensamble 3D de la mano robótica.

Fuente: Autor.

Una vez que se ha verificado el diseño 3D de cada una de las piezas, estas son exportadas en formato STL el cuál elimina aquellos detalles como colores y texturas y mediante varios triángulos aproxima cada pieza a la superficie de un modelo sólido. Esto con el fin de poderlas imprimir en una impresora 3D. El material utilizado para la impresión fue PLA debido a su bajo costo. En la figura 3.10 se puede visualizar la mano robótica impresa y ensamblada.



Figura 3.10: Mano robótica ensamblada.

Fuente: Autor.

3.2.3 Modelo cinemático para un dedo de la mano robótica

El análisis cinemático directo consiste en encontrar la posición y orientación del efector final. Actualmente existen diferentes métodos para obtener el modelo cinemático de un robot, pero en el presente trabajo se hizo uso del método de Denavit-Hantenberg. El cuál nos ayuda a establecer los sistemas de referencia para cada uno de los eslabones con los que cuenta el robot y mediante una serie de reglas poder representar mediante una tabla unos parámetros específicos, que se encuentran relacionados directamente con cada uno de los eslabones [60]. Estos parámetros son los siguientes:

- θ_i : Representa el ángulo entre los ejes X_{i-1} y X_i calculado alrededor del eje Z_{i-1} en sentido antihorario.
- α_i : representa el ángulo entre los ejes Z_{i-1} y Z_i calculado alrededor del eje X_{i-1} en sentido horario.
- l_i representa la distancia entre los ejes Z_{i-1} y Z_i calculado alrededor del eje X_{i-1} .
- d_i representa la distancia entre los ejes X_{i-1} y X_i calculado alrededor del eje Z_{i-1} .

Los ejes de referencia son ubicados sobre cada uno de los dedos de la mano, siguiendo las reglas propuestas por Denavit-Hantenberg, tal como se observa en la figura 3.11.

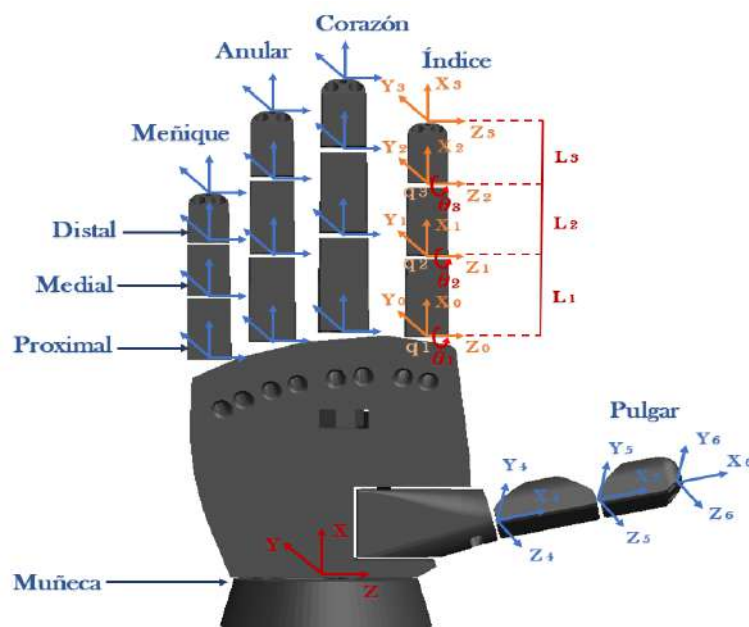


Figura 3.11: Asignación de los sistemas de coordenadas al dedo de la mano.

Fuente: Autor.

En la figura 3.11, L_1 , L_2 y L_3 representan las longitudes de los eslabones del dedo índice (falange proximal, medial y distal respectivamente). Y q_1 , q_2 y q_3 representan la articulación de la falange proximal, medial y distal. Aplicando cada uno de las reglas establecidas por Denavit-Hantenberg, se obtiene la siguiente tabla para el dedo índice.

| Articulación | l_i | α_i | d_i | θ_i |
|--------------|-------|------------|-------|------------|
| 1 | L1 | 0 | 0 | q_1 |
| 2 | L2 | 0 | 0 | q_2 |
| 3 | L3 | 0 | 0 | q_3 |

Tabla 3.4: Parámetros de Denavit-Hantenberg calculados a partir de la base del dedo índice.

Fuente: Autor.

Una vez se han obtenido los parámetros de Denavit-Hantenberg, se procede a calcular las matrices de transformación homogénea asociadas a los tres eslabones del dedo índice respecto a cada eje de referencia.

$$A_1^0 = \begin{vmatrix} C(q_1) & -S(q_1) & 0 & L_1 C(q_1) \\ C(q_1) & C(q_1) & 0 & L_1 S(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.1)$$

donde: *Abreviatura* $\rightarrow \cos(\theta_i) = C(\theta_i)$, $\text{sen}(\theta_i) = S(\theta_i)$
 q_1 \rightarrow Articulación 1.
 L_1 \rightarrow Longitud del eslabón 1 (proximal).

$$A_2^1 = \begin{vmatrix} C(q_2) & -S(q_2) & 0 & L_2 C(q_2) \\ C(q_2) & C(q_2) & 0 & L_2 S(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.2)$$

donde: *Abreviatura* $\rightarrow \cos(\theta_i) = C(\theta_i)$, $\text{sen}(\theta_i) = S(\theta_i)$
 q_2 \rightarrow Articulación 2.
 L_2 \rightarrow Longitud del eslabón 2 (medial).

$$\mathbf{A}_3^2 = \begin{vmatrix} C(q_3) & -S(q_3) & 0 & L_3C(q_3) \\ S(q_3) & C(q_3) & 0 & L_3S(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.3)$$

donde: *Abreviatura* $\rightarrow \cos(\theta_i) = C(\theta_i)$, $\text{sen}(\theta_i) = S(\theta_i)$
 q_3 \rightarrow Articulación 3.
 L_3 \rightarrow Longitud del eslabón 3 (distal).

A continuación se obtienen las matrices de transformación homogénea (MTH) de cada eslabón. Cabe resaltar que las variables L_1 , L_2 y L_3 correspondientes a la longitud y q_1 , q_2 y q_3 correspondientes a las articulaciones del dedo índice.

Para la primera articulación (proximal) se obtiene:

$$\mathbf{A}_0^1 = \begin{vmatrix} C(q_1) & -S(q_1) & 0 & L_1C(q_1) \\ S(q_1) & C(q_1) & 0 & L_1S(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.4)$$

La segunda articulación (medial) se obtiene de la siguiente forma: $\mathbf{A}_2^0 = \mathbf{A}_2^1 * \mathbf{A}_1^0$

$$\mathbf{A}_2^0 = \begin{vmatrix} C(q_1 + q_2 + q_3) & -S(q_1 + q_2 + q_3) & 0 & L_1C(q_1) + L_2C(q_1 + q_2) \\ S(q_1 + q_2 + q_3) & C(q_1 + q_2 + q_3) & 0 & L_1S(q_1) + L_2S(q_1 + q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.5)$$

La tercera articulación (distal) se obtiene de la siguiente forma: $\mathbf{A}_3^0 = \mathbf{A}_3^2 * \mathbf{A}_2^1 * \mathbf{A}_1^0$

$$= \begin{vmatrix} C(q_1 + q_2 + q_3) & -S(q_1 + q_2 + q_3) & 0 & L_1C(q_1) + L_2C(q_1 + q_2) + L_3C(q_1 + q_2 + q_3) \\ S(q_1 + q_2 + q_3) & C(q_1 + q_2 + q_3) & 0 & L_1S(q_1) + L_2S(q_1 + q_2) + L_3S(q_1 + q_2 + q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.6)$$

3.3 Diseño y construcción de la placa de control

Luego de haber terminado la construcción mecánica de la mano se procede a realizar el diseño de la placa de control, utilizando el software Eagle en su versión Educación 9.5.1. Este software se encuentra enfocado especialmente al diseño de placas de circuito impreso (PCBs), por lo cual brinda múltiples herramientas que facilitan el trabajo. El diseño de la placa de control se encuentra dividida en dos partes: esquema eléctrico y diseño de PCB.

3.3.1 Esquema eléctrico

Un esquema eléctrico es una representación gráfica de un circuito eléctrico, cuyo objetivo es brindar una idea más clara sobre cada uno de los dispositivos que lo componen. La placa de control está compuesta por elementos como: borneras, las cuáles son las encargadas de brindar una alimentación de tipo independiente para los servos, regletas de tipo macho de tres pines, para garantizar una buena conexión para cada uno de los servomotores, dos regletas hembra de ocho pines, para la ubicación de la tarjeta programable Wemos D1 mini y un diodo emisor de luz (LED) junto con un resistencia, que sirve para indicar el momento en que la placa programable es conectada a una red wifi. En la figura 3.12 se puede observar el esquemático del circuito.

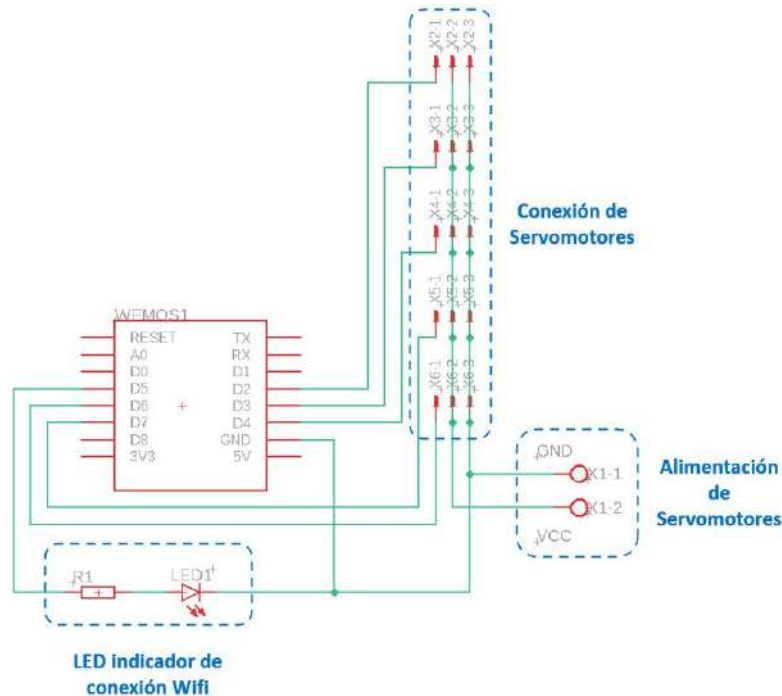


Figura 3.12: Esquema eléctrico de la placa de control.

Fuente: Autor.

3.3.2 Diseño de PCB

Basado en el esquema eléctrico de la figura 3.10, se realiza el diseño impreso del circuito (PCB). Cada uno de los componentes son ubicados de forma que la PCB sea lo más compacta posible, logrando así una dimensión de 57.91 mm de ancho y 40.31 mm de largo, esto debido a que la placa de control se encuentra posicionada en la parte inferior de la mano robótica. El tamaño definido para el ancho de las pistas, fue de aproximadamente 1.14 mm, garantizando de esta forma una tolerancia de corriente superior a los 10A. En la figura 3.13 se observa el diseño de la PCB en Eagle.

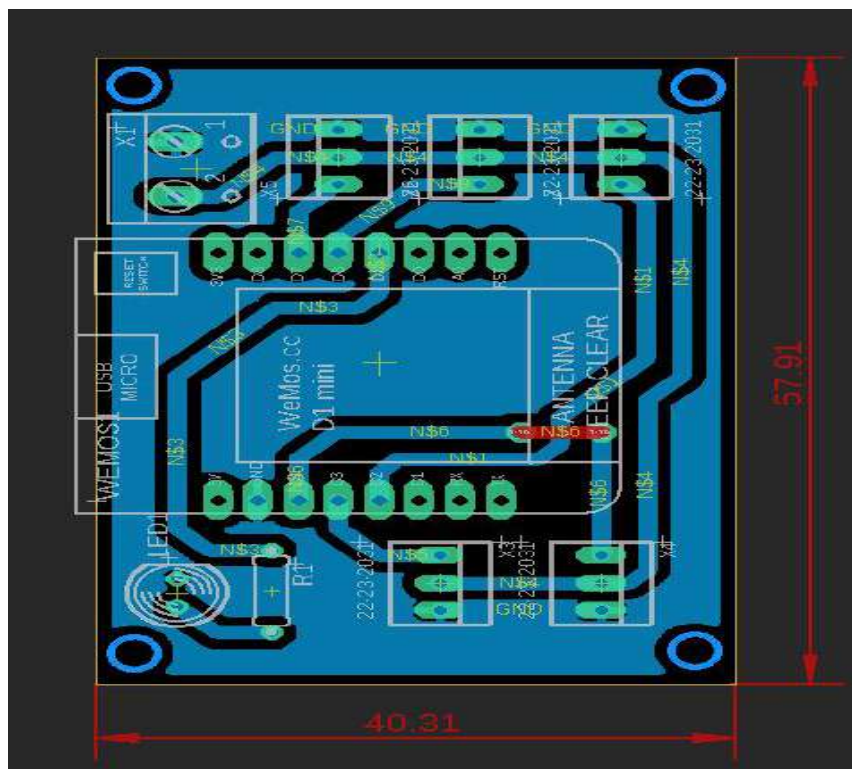


Figura 3.13: Diseño PCB de la placa de control.

Fuente: Autor.

Finalizado el diseño de la PCB en el software Eagle, se da inicio al proceso de fabricación del circuito mediante el método de planchado. Este método consiste en la transferencia térmica del circuito impreso en un papel especial, a una placa de cobre o baquelita, donde luego este es sumergido en ácido férrico, con el fin de que se pueda eliminar el cobre que es innecesario, logrando así la forma del circuito deseado. Posteriormente se realizan los huecos en la baquelita, utilizando una broca de diámetro adecuado, y en cada uno de ellos son posicionados los componentes electrónicos a la placa de control, para finalmente soldarlos utilizando un caudín y estaño.

En la figura 3.14 se observa la placa terminada y en la figura 3.15 el circuito ubicado dentro de una de las piezas de la mano robótica.

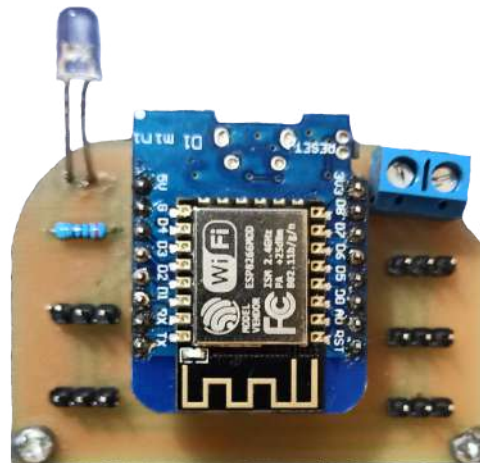


Figura 3.14: Vista superior de la PCB con todos sus componentes.

Fuente: Autor.



Figura 3.15: PCB ubicada dentro de una de las piezas de la mano robótica.

Fuente: Autor.

3.4 Emotiv Epoc

Con el fin de poder seleccionar el dispositivo captador de ondas cerebrales más adecuado para el desarrollo de este proyecto, fue necesario analizar y comparar las características de cada uno de los dispositivos BCI más relevantes del mercado, los cuáles son mencionados en el capítulo 2, sección 2.2.5.

Se determina que el dispositivo Emotiv Epoc (figura 3.16) es la mejor opción debido a su relación calidad/precio, conectividad, accesibilidad, portabilidad y número de electrodos (14 electrodos ubicados según la normativa 10-20 en AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4). Este dispositivo cuenta con un kit de desarrollo de software (SDK), que permite a los usuarios crear sus propias aplicaciones de una manera sencilla utilizando lenguajes de programación como C++, Java, Python, Matlab, entre otros. También incluye diferentes Suites que permiten realizar la identificación de las neuroseñales.



Figura 3.16: Dispositivo Emotiv Epoc.

Fuente: Autor.

3.4.1 Introducción al SDK de de Emotiv

Una de las principales características que proporciona el Kit de Desarrollo de Software (SDK) de Emotiv, son las herramientas que incluye en su paquete de instalación, ya que estas nos permiten procesar las neuroseñales de un usuario para así obtener su estado emocional o identificar una acción cognitiva.

Estas herramientas son conocidas como suite afectiva, expresiva y cognitiva. Para acceder a cada una de ellas es necesario iniciar el entorno gráfico de Emotiv Control Panel. El cuál proporciona información del dispositivo como: ubicación y calidad de contacto de cada uno de los electrodos, nivel de batería y estado de la conexión. En la figura 3.17 se puede observar está interfaz gráfica, la cual divide en pequeñas pestañas cada una de las suites anteriormente mencionadas.

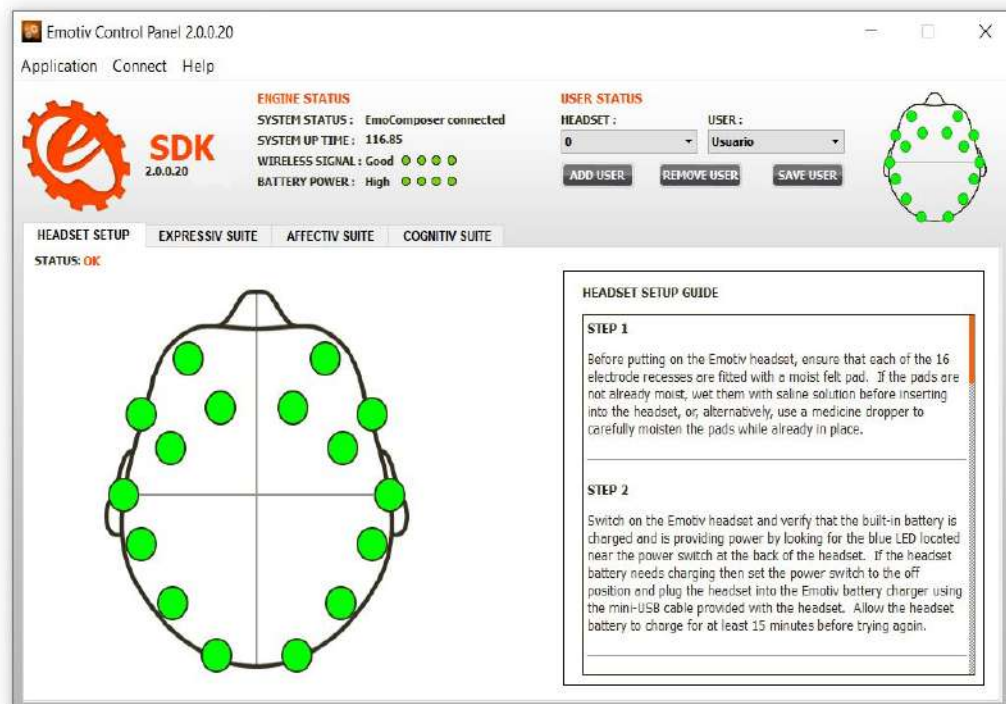


Figura 3.17: Emotiv Control Panel.

Fuente: Autor.

A continuación se describe cada una de las Suite que proporciona el SDK de Emotiv.

- **Suite Afectiva:**

La suite afectiva brinda información relacionada al estado emocional de una persona. Esta herramienta permite identificar estados como: meditación, frustración, excitación y aburrimiento. En la figura 3.18 se observa cómo se van graficando los estados emocionales dentro de la suite expresiva. El primer gráfico (de arriba hacia abajo) proporciona información a corto plazo (30 segundos), mientras que el segundo gráfico puede proporcionar la información capturada durante 600 segundos.



Figura 3.18: Análisis de emociones utilizando la suite afectiva.

Fuente: Autor.

Una aplicación bastante interesante para el uso de esta suite se basa en leer el nivel de atención de un usuario mientras observa un comercial, de tal forma que se pueda determinar y evaluar el impacto que este genera. Logrando así que una empresa pueda mejorar sus estrategias de marketing.

Para el presente trabajo se capturaron algunas métricas afectivas de los usuarios mediante el uso de un programa desarrollado en Java utilizando el SDK de Emotiv, los cuales fueron almacenados en un archivo de Excel. Esto con el fin de poder evaluar la experiencia de los usuarios y así poder determinar qué factores influyeron en el desempeño del sistema.

- **Suite Expresiva:**

Esta suite permite detectar las expresiones o gestos faciales que realiza una persona. Algunos de los gestos que se pueden reconocer son: mirar a la izquierda, mirar a la derecha, mirar al frente, guiñar el ojo izquierdo, guiñar el ojo derecho, sonreír, parpadear y apretar los dientes. Además, cuenta con unas configuraciones adicionales que nos permite ir variando la sensibilidad de detección de cada uno de los gestos fáciles, con el fin de poder mejorar la experiencia del usuario. En la figura 3.19 se puede observar como un avatar dentro de la suite imita cada gesto que ejecuta el usuario.

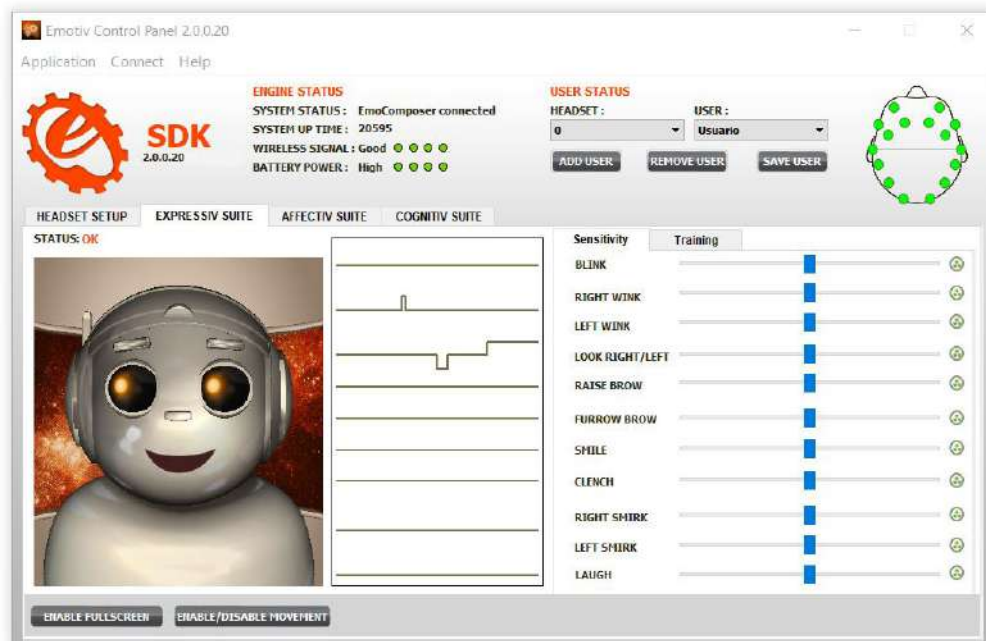


Figura 3.19: Reconocimiento de gestos mediante la suite expresiva.

Fuente: Autor.

- **Suite Cognitiva:**

La suite cognitiva, permite detectar hasta 14 comandos mentales relacionados al movimiento de una caja, como lo son: empujar, jalar, empujar a la izquierda, empujar a la derecha, rotar en sentido horario, rotar en sentido antihorario, entre otros. A diferencia de las otras suite, esta cuenta con una mayor cantidad de herramientas que permiten mejorar la experiencia del usuario.

Debido a que cada persona piensa de una forma diferente se hace necesario realizar un entrenamiento de 8 segundos por cada comando mental a utilizar. Cada vez que un comando sea entrenado, la suite nos mostrara un porcentaje de entrenamiento. Entre más definidos sean nuestros pensamientos, mejor serán los resultados obtenidos. Una vez que han sido entrenados los comandos mentales a utilizar, podemos realizar varias pruebas con el simulador de esta suite, el cual nos proporciona a la vez una retroalimentación de tipo visual, ya que si pensamos en empujar la caja hacia atrás está inmediatamente recreara esa acción que le hemos dado. Por otro lado, en esta suite podemos variar la sensibilidad de cada uno de los 14 comandos mentales que permite detectar, de tal forma que se puedan reducir la complejidad de uso del sistema y los tiempos de entrenamiento. En la figura 3.20, se observa como mediante un pensamiento es posible empujar la caja hacia atrás.

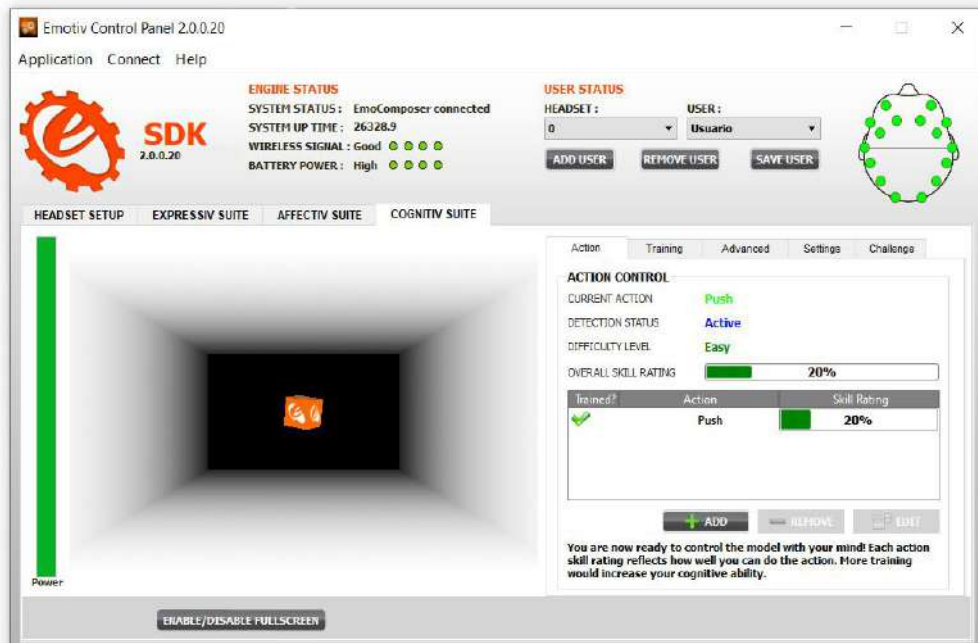


Figura 3.20: Procesamiento de comandos mentales.

Fuente: Autor.

3.4.2 Desarrollo de la Interfaz 1

El Kit de Desarrollo de Software (SDK) de emotiv proporciona algunos ejemplos básicos en lenguajes de programación como Java, C++, Matlab y Python, cuyos ejemplos permiten realizar un procesamiento de las neuroseñales. De esta forma cualquier desarrollador puede crear un programa capaz de interactuar con el casco para realizar una tarea en específico.

Para el desarrollo del presente trabajo se crearon dos interfaces gráficas utilizando el lenguaje de programación Java junto al entorno de desarrollo integrado (IDE) Eclipse. Debido a que este lenguaje de programación proporciona algunas ventajas como el paradigma programación orientado a objetos (POO), el cual permite reutilizar código y facilita el desarrollo. El objetivo de las interfaces gráficas desarrolladas es permitir que un usuario pueda controlar los movimientos de una mano robótica mediante el uso de comandos mentales y/o gestos faciales. En esta sección se explicará el procedimiento realizado para el desarrollo de la primera interfaz gráfica. Es importante resaltar que esta interfaz permite a los usuarios controlar tres movimientos de la mano robótica (abrir mano, cerrar mano, agarre de pinza fina) mediante el uso de comandos mentales. Además, también brinda la posibilidad de controlar el grado de apertura de cada movimiento mediante la intensidad del pensamiento. Es decir, entre más fuerte y definido sea el pensamiento, mayor será la fuerza del movimiento ejecutado.

Se comienza creando un espacio de trabajo dentro del entorno de desarrollo integrado Eclipse. Este espacio de trabajo contiene a su vez una carpeta principal denominada *CognitivExample* que está compuesta por diferentes archivos y librerías necesarias para el proyecto. Generalmente en los proyectos de java se pueden encontrar archivos conocidos como paquetes, los cuales son contenedores de clases que permiten agrupar las distintas partes de un programa, en este caso el paquete se llama *default package*. En el lado izquierdo de la figura 3.21 se puede visualizar la estructura del proyecto.

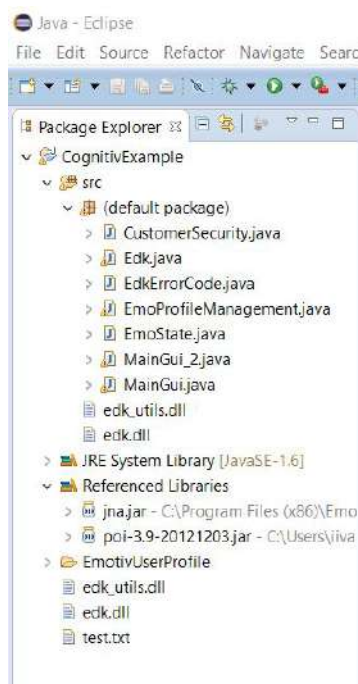


Figura 3.21: Configuración de espacio de trabajo en Eclipse.

Fuente: Autor.

Finalizadas las configuraciones necesarias, se procede a trabajar en el archivo principal *MainGui.java*. Debido a que este archivo es bastante extenso, solo se explicaran los fragmentos de código más relevantes.

Al igual que en la mayoría de algoritmos se comienza importando todas aquellas librerías que se van a utilizar en el proyecto. Luego son declaradas todas aquellas variables de tipo global relacionadas a los elementos de la interfaz, como lo son los *JTextField* (encargados de capturar el texto), los *JButton* (botones) y los *JLabel* (encargados de visualizar las imágenes). Posteriormente se inicia el proceso de comunicación de la interfaz con el dispositivo captador de ondas cerebrales, para ello se utiliza el fragmento de código 3.1. Mediante la variable de tipo entera *option* se define con que se va a comunicar nuestra interfaz. Ya que si esta tiene un valor equivalente a 1 se conectará

con el *EmotivEngine*, es decir directamente con el casco, pero en dado caso que esta variable tenga un valor de 2 la interfaz se conectara a una herramienta denominada *Composer* utilizando el puerto 1726, la cuál es una herramienta que brinda la posibilidad de simular algunas acciones del casco de neuroseñales, para así poder probar nuestras aplicaciones en dado caso de que no contemos con el dispositivo.

Código 3.1: Algoritmo para establecer la comunicación entre la interfaz y el casco

```

1
2   Pointer eEvent = Edk.INSTANCE.EE_EmoEngineEventCreate();
3   Pointer eState = Edk.INSTANCE.EE_EmoStateCreate();
4   IntByReference userID = null;
5   short composerPort = 1726;
6   int option = 1;
7   int state = 0;
8
9   userID = new IntByReference(0);
10
11  switch (option) {
12
13  case 1:
14  {
15      if (Edk.INSTANCE.EE_EngineConnect("Emotiv Systems-5") != EdkErrorCode.EDK_OK.ToInt() ←
16          ↪ ()) {
17          System.out.println("Emotiv Engine start up failed.");
18          return;
19      }
20      break;
21  }
22
23  case 2:
24  {
25      System.out.println("Target IP of EmoComposer: [127.0.0.1] ");
26
27      if (Edk.INSTANCE.EE_EngineRemoteConnect("127.0.0.1", composerPort, "Emotiv Systems-5") ←
28          ↪ != EdkErrorCode.EDK_OK.ToInt()) {
29          System.out.println("Cannot connect to EmoComposer on [127.0.0.1]");
30          return;
31      }
32      System.out.println("Connected to EmoComposer on [127.0.0.1]");
33      break;
34  }
35
36  default:
37      System.out.println("Invalid option...");
38      return;
39  }

```

Cuando la conexión de la interfaz con el casco o el simulador ha sido exitosa, se inicia un ciclo repetitivo infinito, el cual es el encargado de garantizar el envío de datos desde la interfaz desarrollada a la placa de control mediante el protocolo de comunicación UDP (protocolo de datagrama de usuarios), el cuál esta basado en protocolo TCP/IP, pero brinda la ventaja de ser más rápido. Una vez que los datos son recibidos por la placa de control, esta procede a variar la posición de cada uno de los actuadores de la mano robótica con el fin de lograr el movimiento y/o gesto asociado al comando mental ejecutado por el usuario. En el código 3.2 se puede visualizar como son instanciados

los objetos necesarios para establecer este tipo de comunicación.

Código 3.2: Instancia de objetos para la comunicación UDP

```

1
2     while (true) {
3
4         DatagramSocket socket;
5         InetAddress address;
6         byte[] mensaje_bytes = new byte[256];
7         String mensaje = "";
8         mensaje_bytes = mensaje.getBytes();
9
10        DatagramPacket paquete;
11        String cadenaMensaje = "";
12        DatagramPacket servPaquete;
13        byte[] RecogerServidor_bytes = new byte[256];

```

Se hace el llamado de la variable *state* en el código 3.1, con el objetivo de conocer si los archivos *edk.dll* del SDK de Emotiv han sido cargados correctamente y de esta forma poder continuar con la ejecución del programa principal. Después se crea una estructura de decisión basada en condicionales *if* como se observa en el código 3.3, el cual permite evaluar diferentes eventos del programa y respecto a ellos ejecutar unas determinadas acciones, las cuales están relacionadas al entrenamiento de los comandos mentales. Mediante la instrucción *JOptionPane* se muestra un mensaje en la interfaz que nos indica el momento en el que el entrenamiento cognitivo inicia, termina o falla.

Código 3.3: Eventos relacionados al entrenamiento de comandos mentales

```

1
2     state = Edk.INSTANCE.EE_EngineGetNextEvent(eEvent);
3
4     if (state == Edk.ErrorCode.EDK_OK.ToInt()) {
5
6         int eventType = Edk.INSTANCE.EE_EmoEngineEventGetType(eEvent);
7
8         Edk.INSTANCE.EE_EmoEngineEventGetUserId(eEvent, userID);
9
10
11        if(eventType == Edk.EE_Event_t.EE_CognitivEvent.ToInt())
12        {
13            int cogType = Edk.INSTANCE.EE_CognitivEventGetType(eEvent);
14
15            if(cogType == Edk.EE_CognitivEvent_t.EE_CognitivTrainingStarted.getType())
16            {
17                JOptionPane.showMessageDialog(new JFrame(),
18                    "Iniciando Entrenamiento Cognitivo", "Dialog",
19                    JOptionPane.ERROR_MESSAGE);
20            }
21            if(cogType == Edk.EE_CognitivEvent_t.EE_CognitivTrainingCompleted.getType())
22            {
23                JOptionPane.showMessageDialog(new JFrame(),
24                    "Entrenamiento Cognitivo Completado", "Dialog",
25                    JOptionPane.ERROR_MESSAGE);
26            }
27            if(cogType == Edk.EE_CognitivEvent_t.EE_CognitivTrainingSucceeded.getType())
28            {

```

```

29         Edk.INSTANCE.EE_CognitivSetTrainingControl
30         (0,Edk.EE_CognitivTrainingControl_t.COG_ACCEPT.getType());
31         JOptionPane.showMessageDialog(new JFrame(),
32             "Entrenamiento cognitivo exitoso", "Dialog",
33             JOptionPane.ERROR_MESSAGE);
34         label_1.setIcon(new ImageIcon(currentPath + "\\imagenes\\mano_3.png"));
35     }
36     if(cogType == Edk.EE_CognitivEvent_t.EE_CognitivTrainingFailed.getType())
37     {
38         JOptionPane.showMessageDialog(new JFrame(),
39             "Entrenamiento cognitivo fallido", "Dialog",
40             JOptionPane.ERROR_MESSAGE);
41     }
42     if(cogType == Edk.EE_CognitivEvent_t.EE_CognitivTrainingRejected.getType())
43     {
44         JOptionPane.showMessageDialog(new JFrame(),
45             "Entrenamiento cognitivo rechazado", "Dialog",
46             JOptionPane.ERROR_MESSAGE);
47     }
48 }

```

Para el presente trabajo se capturaron algunas métricas relacionadas a la parte afectiva del usuario (meditación, frustración, excitación y aburrimiento) utilizando un botón dentro de la interfaz gráfica 1. Cuando el botón es presionado los datos son guardados dentro de un archivo de Excel. El objetivo de almacenar estos datos es poder estimar la experiencia de los usuarios y así conocer su evolución a medida que iban utilizando y entrenando el sistema.

La captura de las emociones se hace utilizando el código 3.4, el cual se encuentra dentro del ciclo repetitivo infinito mencionado en el código 3.2, ya que esto permite que las métricas estén actualizadas en todo momento. Cuando los campos de dirección ip, puerto y perfil son llenados desde la interfaz, se procede a inicializar el socket para la comunicación UDP y a su vez se inicia la lectura de las emociones. Estas son visualizadas en la interfaz gráfica mediante la instrucción *setText()* y guardadas en un arreglo de variables flotantes, ya que más adelante serán insertadas dentro de un archivo de Excel el cuál se creara y guardara en una ruta específico. Una vez que han transcurrido los 10 minutos desde que se han presionado el botón para guardar las métricas afectivas, la escritura en el archivo de Excel es interrumpida utilizando la instrucción *archivo.close()*, y al instante este se abre de forma automática con el objetivo de validar que la información ha sido guardada de manera correcta.

Código 3.4: Lectura de emociones

```

1
2     if (eventType == Edk.EE_Event_t.EE_EmoStateUpdated.ToInt()) {
3         Edk.INSTANCE.EE_EmoEngineEventGetEmoState(eEvent, eState);
4
5         if (direccion_ip != "" && puerto != "" && p != "") {
6
7             try {
8
9                 // Se crea el socket y se asigna la dirección ip

```

```

10 socket = new DatagramSocket();
11 address = InetAddress.getByName(direccion_ip);
12
13 // Se leen las emociones utilizando los métodos de la clase Emostate
14 float f = EmoState.INSTANCE.ES_AffectivGetFrustrationScore(eState);
15 float m = EmoState.INSTANCE.ES_AffectivGetMeditationScore(eState);
16 float e = EmoState.INSTANCE.ES_AffectivGetExcitementShortTermScore(eState);
17 float a = EmoState.INSTANCE.ES_AffectivGetEngagementBoredomScore(eState);
18
19 // Las emociones son almacenadas en un vector de tipo float
20 emociones[0] = f * 10;
21 emociones[1] = m * 10;
22 emociones[2] = e * 10;
23 emociones[3] = a * 10;
24
25 String fr, me, ex, ab;
26
27 // Conversión Float a String
28 fr = Float.toString(emociones[0]);
29 me = Float.toString(emociones[1]);
30 ex = Float.toString(emociones[2]);
31 ab = Float.toString(emociones[3]);
32
33 // Se muestra el valor de cada emoción en la interfaz
34 frustracion.setText(fr);
35 meditacion.setText(me);
36 excitacion.setText(ex);
37 Aburrimiento.setText(ab);
38
39 // Condición para iniciar la captura de las emociones
40 if (estado == 1) {
41     cont++;
42
43     TInicio = System.currentTimeMillis();
44     if (documento == 1) {
45
46         // Se define la ruta del archivo
47         String rutaArchivo = System.getProperty("user.home") + "/" + p + ".xls";
48         System.out.println(rutaArchivo);
49         archivoXLS = new File(rutaArchivo);
50
51         // En dado caso de que ya existe, se elimina y crea uno nuevo
52         if (archivoXLS.exists())
53             archivoXLS.delete();
54         archivoXLS.createNewFile();
55         libro = new HSSFWorkbook();
56         archivo = new FileOutputStream(archivoXLS);
57         hoja = libro.createSheet("Emociones Emotiv EPOC");
58         documento = 0;
59
60     }
61
62     TFin = System.currentTimeMillis();
63     int fi = cont - 1;
64
65     // Se inicia el llenado del archivo de excel
66     Row fila = hoja.createRow(fi);
67     for (int c = 0; c < 4; c++) {
68         Cell celda = fila.createCell(c);
69         if (f == 0) {
70             celda.setCellValue("Encabezado #" + c);
71         } else {
72             celda.setCellValue(emociones[c]);
73         }
74     }

```

```

74     }
75
76     // Cuando se cumplen los 10 minutos, finaliza la escritura del archivo .xls
77     // y se abre al instante
78     if (tiempo >= 600000) {
79         libro.write(archivo);
80         archivo.close();
81         cont = 0;
82         estado = 0;
83         Desktop.getDesktop().open(archivoXLS);
84     }
85 }

```

Mediante el código 3.5 se realiza la identificación de los comandos mentales. En este código se utiliza dos métodos propios del SDK de emotiv los cuáles son: el *CognitivGetCurrentAction()* el cuál retorna un valor de tipo entero que es almacenado en la variable caso y el *CognitivGetCurrentActionPower()* que retorna la intensidad del comando mental (valor entre 0 y 1, con incrementos de 0.1). La variable caso es evaluada mediante una estructura condicional case, con el fin de poder determinar el tipo de comando mental que el usuario está ejecutando. Una vez que el comando mental ha sido identificado, se procede a multiplicar el valor de la intensidad del comando por 10, con el fin de convertirlo a un valor entero y así facilitar el mapeo de los datos en la placa de control. Posteriormente los datos son enviados a la placa mediante la instrucción *socket.send()*, donde esta se encarga de variar las posiciones de cada uno de los servomotores con el fin de lograr los movimientos que el usuario quiere realizar (abrir mano, cerrar mano, posición de pinza fina).

Código 3.5: Identificación de comandos mentales

```

1
2     double power;
3
4     // Se lee la acción cognitiva del usuario
5     caso = EmoState.INSTANCE.ES_CognitivGetCurrentAction(eState);
6
7     // Se lee la intensidad con la que se ejecuta la acción cognitiva
8     power = EmoState.INSTANCE.ES_CognitivGetCurrentActionPower(eState);
9
10    switch (caso) {
11
12    // Abrir mano
13    case 1:
14
15        // Se muestra el comando mental en la interfaz gráfica
16        comando.setText("Abrir Mano");
17        mensaje = "1," + "0";
18
19        // Se muestra el dato que se va a enviar a la placa de control
20        dato_e.setText(mensaje);
21
22        mensaje_bytes = mensaje.getBytes();
23
24        paquete = new DatagramPacket(mensaje_bytes,
25        mensaje.length(), address,
26        Integer.parseInt(puerto));

```

```
27
28 // Se envía el dato correspondiente al estado neutral
29 socket.send(paquete);
30
31 if (iniciar == 1) {
32     // Se muestra a imagen de una mano abierta
33     // Retroalimentación visual
34     label_1.setIcon(new ImageIcon(
35         currentPath + "\\imagenes\\abrir.jpg"));
36 }
37
38 break;
39
40 // Cerrar mano
41 case 2:
42
43     // Se muestra el comando mental en la interfaz gráfica
44     comando.setText("Cerrar Mano");
45
46     // Se define el mensaje a enviar
47     mensaje = "2," + (power * 10);
48
49     // Se muestra el dato que se va a enviar a la placa de control
50     dato_e.setText(mensaje);
51
52     mensaje_bytes = mensaje.getBytes();
53
54     paquete = new DatagramPacket(mensaje_bytes,
55         mensaje.length(), address,
56         Integer.parseInt(puerto));
57
58     // Se envía el dato correspondiente al comando mental cerrar mano
59     socket.send(paquete);
60
61     if (iniciar == 1) {
62         // Se muestra la imagen de una mano cerrada
63         // Retroalimentación visual
64         label_1.setIcon(new ImageIcon(
65             currentPath + "\\imagenes\\cerrada.jpg"));
66     }
67
68     break;
69
70 // Agarre de pinza fina
71 case 8:
72
73     // Se muestra el comando mental en la interfaz gráfica
74     comando.setText("Pinza Fina");
75
76     // Se define el mensaje a enviar
77     mensaje = "3," + (power * 10);
78
79     // Se muestra el dato que se va a enviar a la placa de control
80     dato_e.setText(mensaje);
81
82     mensaje_bytes = mensaje.getBytes();
83     paquete = new DatagramPacket(mensaje_bytes,
84         mensaje.length(), address,
85         Integer.parseInt(puerto));
86
87     // Se envía el dato correspondiente al comando mental agarre de pinza fina
88     socket.send(paquete);
89
90     if (iniciar == 1) {
```

```

91         // Se muestra la imagen de una mano en posición de pinza fina
92
93         // Retroalimentación visual
94         label_1.setIcon(new ImageIcon(
95             currentPath + "\\imagenes\\pinza.jpg"));
96     }
97
98     break;
99
100    // Comando mental diferente a los establecidos
101    default:
102
103        // Se muestra que no hay un comando mental detectado
104        comando.setText("Comando no");
105
106    }

```

Debido a que cada persona piensa de forma diferente se hace necesario crear un perfil personal por cada usuario que utiliza el sistema, de tal forma que este permita guardar los entrenamientos cognitivos realizados. La creación del perfil se hace utilizando el código 3.6.

Código 3.6: Algoritmo para crear y guardar perfiles de usuarios

```

1
2    // Se instancia el botón crear perfil de tipo JButton
3    JButton btnCrearPerfil = new JButton("Crear");
4    btnCrearPerfil.setBounds(127, 498, 87, 27);
5
6    // Se ejecuta el siguiente evento cuando el botón crear perfil es presionado
7    btnCrearPerfil.addActionListener(new ActionListener() {
8
9        @Override
10       public void actionPerformed(ActionEvent e) {
11
12           // Se obtiene el texto de la interfaz
13           p = perfil.getText();
14           EmoProfileManagement.AddNewProfile(p);
15           JOptionPane.showMessageDialog(new JFrame(), "Usuario Agregado",
16               "Dialog", JOptionPane.ERROR_MESSAGE);
17       }
18   });
19
20   // Se instancia el botón guardar perfil de tipo JButton
21   saveBtt = new JButton("Guardar Perfil");
22   saveBtt.setBounds(765, 498, 139, 27);
23   saveBtt.setFont(new Font("Times New Roman", Font.PLAIN, 16));
24
25   // Se activa el siguiente evento cuando el botón es presiondo en la interfaz
26   saveBtt.addActionListener(new ActionListener() {
27
28       @Override
29       public void actionPerformed(ActionEvent e) {
30
31           // Se guarda el perfil
32           EmoProfileManagement.SaveCurrentProfile();
33           EmoProfileManagement.SaveProfilesToFile();
34       }
35   });

```

El código 3.7 permite cargar un perfil que haya sido previamente entrenado. La instrucción *EmoProfileManagement.CheckCurrentProfile()* retorna un valor de tipo *String* el cual permite conocer el tipo y número de comandos mentales que fueron entrenados con el perfil cargado.

Código 3.7: Algoritmo para cargar perfiles previamente entrenados

```
1
2 // Se ejecuta el siguiente evento cuando el botón cargar perfil es presionado
3 loadBtt.addActionListener(new ActionListener() {
4
5     @Override
6     public void actionPerformed(ActionEvent e) {
7
8         // Se obtiene el nombre del perfil a cargar
9         p = perfil.getText();
10
11        // Se carga el perfil
12        EmoProfileManagement.LoadProfilesFromFile();
13        EmoProfileManagement.SetUserProfile(p);
14
15        // Se leen las acciones cognitivas entrenadas con el perfil
16        String actionList = EmoProfileManagement.CheckCurrentProfile();
17        long cognitivActions = Long.valueOf(actionList);
18
19        Edk.INSTANCE.EE_CognitivSetActiveActions(0, cognitivActions);
20
21        JOptionPane.showMessageDialog(new JFrame(), "Usuario Cargado",
22            "Dialog", JOptionPane.ERROR_MESSAGE);
23
24    }
25
26 });
```

Esta interfaz permite realizar el entrenamiento de tres comandos mentales diferentes, los cuales son: abrir mano, cerrar mano y agarre de pinza fina. Cabe resaltar que el entrenamiento de cada comando mental dura aproximadamente 8 segundos. Dentro de la interfaz se puede visualizar un botón desplegable que permite seleccionar el comando mental a entrenar, posteriormente a esto se ejecuta el fragmento de código 3.8, el cual permite identificar la opción marcada y basado en ello iniciar el entrenamiento correspondiente. Además, con el fin de lograr que los usuarios obtengan un entrenamiento más exitoso, en el área de entrenamiento se visualiza una imagen asociada al comando mental que se está ejecutando. De tal forma que esto le permita al usuario tener una retroalimentación de tipo visual. El cambio de la imagen se realiza mediante el uso de la instrucción *label1.setIcon(new ImageIcon())* el cuál es un método propio de Java.

Código 3.8: Entrenamiento cognitivo

```

1 // Se instancia el objeto del botón entrenar perfil
2 trainBtt = new JButton("Entrenar");
3 trainBtt.setBounds(657, 498, 96, 27);
4 trainBtt.setFont(new Font("Times New Roman", Font.PLAIN, 16));
5 // Se ejecuta el siguiente código cuando el botón es presionado
6 trainBtt.addActionListener(new ActionListener() {
7
8     @Override
9     public void actionPerformed(ActionEvent e) {
10
11         // Se lee el tipo de comando mental que será entrenado
12         int index = comboBox.getSelectedIndex();
13
14         // Entrenamiento del comando mental abrir mano
15         if (index == 0) {
16             Edk.INSTANCE.EE_CognitivSetTrainingAction(0,
17                 EmoState.EE_CognitivAction_t.COG_NEUTRAL.ToInt());
18             Edk.INSTANCE.EE_CognitivSetTrainingControl(0,
19                 Edk.EE_CognitivTrainingControl_t.COG_START
20                 .getType());
21             label_1.setIcon(new ImageIcon(
22                 // Se muestra la imagen de una mano abierta
23                 currentPath + "\\imagenes\\abrir.jpg"));
24         }
25
26         // Entrenamiento del comando cerrar mano
27         if (index == 1) {
28             try {
29                 EnableCognitivAction(
30                     EmoState.EE_CognitivAction_t.COG_PUSH, true);
31                 EnableCognitivActionsList();
32                 StartTrainingCognitiv(EmoState.EE_CognitivAction_t.COG_PUSH);
33
34                 label_1.setIcon(new ImageIcon(
35                     // Se muestra la imagen de una mano cerrada
36                     currentPath + "\\imagenes\\cerrada.jpg"));
37
38             } catch (Exception ex) {
39                 ex.printStackTrace();
40             }
41         }
42
43         // Entrenamiento del comando mental agarre de pinza fina
44         if (index == 2) {
45             try {
46                 EnableCognitivAction(
47                     EmoState.EE_CognitivAction_t.COG_LIFT, true);
48                 EnableCognitivActionsList();
49                 StartTrainingCognitiv(EmoState.EE_CognitivAction_t.COG_LIFT);
50                 label_1.setIcon(new ImageIcon(
51                     // Se muestra la imagen de una mano en posición de pinza fina
52                     currentPath + "\\imagenes\\pinza.jpg"));
53             } catch (Exception ex) {
54                 ex.printStackTrace();
55             }
56         }
57     }
58 });


```

En la figura 3.22 se puede observar el diseño final de la interfaz gráfica 1. La cuál

se divide en diferentes secciones: perfil de usuario, área de entrenamiento, configuración para la comunicación UDP (protocolo de datagrama de usuarios) y visualización y captura de emociones. Cabe resaltar que dentro del código se ha configurado la opción de conexión, de tal forma que la interfaz se conecte directamente con el casco de neuroseñales y no con el *Composser* (simulador del casco). La imagen que se encuentra en el área de entrenamiento cambiará conforme se vayan seleccionando los comandos mentales a entrenar, proporcionando así una retroalimentación visual y un mejor entrenamiento para los usuarios.


CONTROL DE MANO ROBÓTICA

Perfil:



Nombre:

Entrenamiento:



Comandos Mentales

Configuración UDP:

Puerto:

Dirección IP:

Lectura de Emociones:

Meditación:

Excitación:

Frustración:

Aburrimiento:

Figura 3.22: Diseño final de la interfaz gráfica 1.

Fuente: Autor.

3.4.3 Desarrollo de la Interfaz 2

La interfaz 1 está limitada a un uso máximo de tres comandos mentales, por lo cual los usuarios podrán realizar solo tres tipos de movimientos con la mano robótica (mano abierta, mano cerrada y posición de pinza fina). Esto se debe a que un cuarto comando mental exigiría a los usuarios una mayor destreza y un tiempo considerable de entrenamiento (más de una hora y media), parece mucho tiempo, pero en realidad es poco teniendo en cuenta cuanto tarde un bebe en dominar los movimientos de su mano. Es importante resaltar, que los agarres precisos también son más difíciles de ejecutar, debido a que el grado de apertura de la mano robótica se encuentra relacionado directamente con la intensidad del comando mental. Es decir, entre más fuerte sea el pensamiento, mayor será el agarre de la mano robótica.

Debido a esto se procede a crear la interfaz gráfica 2, la cual combina el uso de la parte cognitiva con la parte expresiva, permitiendo lograr a los usuarios la ejecución de un mayor número de movimientos con la mano (presión digito palmar, mano cerrada, agarre palmar esférica penta digital, agarre presa de destornillador, agarre de pinza fina). La interfaz está conformada por un menú el cuál es controlado mediante el uso de eventos oculares, como mirar hacia la derecha para avanzar, mirar hacia la izquierda para retroceder y mirar hacia el frente para seleccionar el movimiento a realizar. Mediante el uso de comandos mentales previamente entrenados y que se encuentran relacionados directamente con colores en específicos, se realiza el control de la intensidad de agarre de cada movimiento. Sin embargo, los comandos mentales solo permiten variar la intensidad de cierre de 0% a 75%, por lo cual se incluyó el gesto facial apretar dientes, de tal forma que, cuando el usuario haga presión con sus dientes, el movimiento de la mano seleccionado en el menú, se ejecute realizando un cierre del 100%. Además, como complemento, en la interfaz se visualiza un color rojo, cada vez que este gesto es activado. En la tabla 3.5 y 3.6 se puede observar los gestos faciales y comandos mentales, relacionados cada uno con la acción que estos permiten ejecutar dentro de la interfaz.

| Tipo de expresión facial | Tipo de acción |
|--------------------------|--------------------------|
| Mirar a la izquierda | Retrocede en el menú |
| Mirar a la derecha | Avanza en el menú |
| Mirar al frente | Selecciona el movimiento |
| Apretar los dientes | Cierre del 100 % |

Tabla 3.5: Relación de gestos faciales con las acciones que se pueden ejecutar.

Fuente: Autor.

| Tipo de comando mental | Tipo de acción |
|------------------------|-----------------|
| Azul | Cierre del 0 % |
| Verde | Cierre del 35 % |
| Amarillo | Cierre del 70 % |

Tabla 3.6: Relación de cada uno de los comandos mentales con su intensidad de cierre.

Fuente: Autor.

En la figura 3.23 se puede visualizar el menú de movimientos, en donde la flecha azul representa la acción de retroceder y la flecha verde la acción de avanzar. En la parte inferior del menú se observa un círculo azul (activo) y tres de color gris (inactivos). Según se vaya ejecutando los comandos mentales de la tabla 3.3, estos se irán activando. Brindando una retroalimentación de tipo visual que permita a los usuarios obtener un mejor desempeño en el uso del sistema. El botón blanco que se visualiza en la parte inferior derecha de la imagen permite activar o desactivar la detección de los gestos faciales, ya que en algunas ocasiones se generan ruidos que afectan el funcionamiento del sistema.

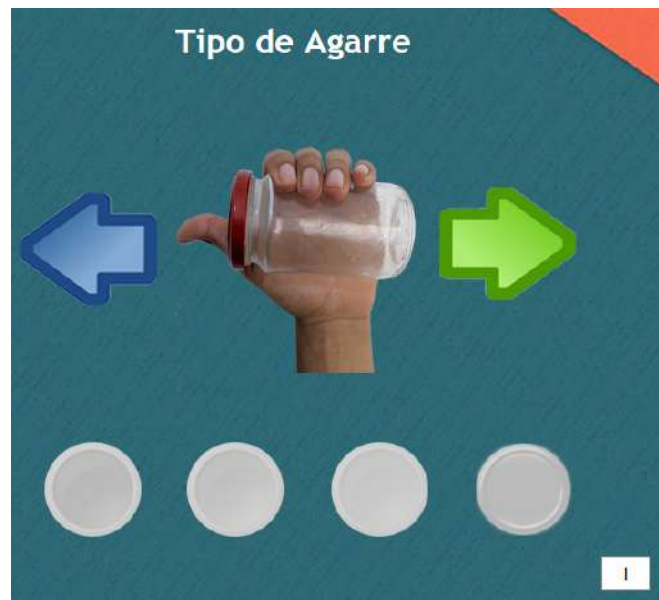


Figura 3.23: Menú de movimientos.

Fuente: Autor.

La configuración para establecer la conexión de la interfaz con el casco de neuroseñales, configuración del protocolo de comunicación UDP, instancia de eventos relacionados al entrenamiento de comandos mentales y la lectura de emociones. Se hacen utilizando los algoritmos mencionados en el desarrollo de la interfaz 1.

Para el reconocimiento de los eventos oculares se utilizaron algunos métodos propios del SDK de emotiv, los cuales retornan un valor booleano (1 para indicar que el evento fue detectado, 0 para lo contrario). La instrucción *ExpressivIsLookingLeft(eState)* es la encargada de detectar el movimiento de los ojos hacia el lado izquierdo, mientras que la instrucción *ExpressivIsLookingRight(eState)* permite detectar el movimiento de los ojos hacia el lado derecho. En el código 3.9 se puede visualizar de forma más detallada el reconocimiento de estos gestos y como permiten realizar la navegación dentro del menú de movimientos.

Código 3.9: Reconocimiento de eventos oculares

```

1
2 // Captura de eventos oculares
3 if (activar_gestos == true) {
4
5 // Mirar hacia la izquierda
6 if (EmoState.INSTANCE
7     .ES_ExpressivIsLookingLeft(eState) == 1) {
8
9 // Retrocede en el menú
10 seleccion--;
11
12 // Vuelve al movimiento inicial
13 if (seleccion == -1) {
14     seleccion = 4;
15 }
16
17 // Se muestra la imagen de presión
18 // digitopalmar
19 lblNewLabel_1
20     .setIcon(new ImageIcon(
21         currentPath + "\\imagenes\\f4.png"));
22 }
23 // Mirar hacia la derecha
24 else if (EmoState.INSTANCE
25     .ES_ExpressivIsLookingRight(eState) == 1) {
26     seleccion++;
27     label_2.setIcon(new ImageIcon(
28         currentPath + "\\imagenes\\f3.png"));
29 }
30
31 // Primera opción del menú
32 if (seleccion == 0) {
33     agarre.setText("Presión digitopalmar");
34     label_1.setIcon(new ImageIcon(
35         currentPath + "\\imagenes\\mano_1.png"));
36 }
37 // Segunda opción del menú
38 else if (seleccion == 1) {
39     agarre.setText("Cerrar Mano");
40     label_1.setIcon(new ImageIcon(
41         currentPath + "\\imagenes\\mano_2.png"));

```

```

42     }
43     // Tercera opción del menú
44     else if (seleccion == 2) {
45         agarre.setText("Pinza Fina");
46         label_1.setIcon(new ImageIcon(
47             currentPath + "\\imagenes\\mano_3.png"));
48     }
49     // Cuarta opción del menú
50     else if (seleccion == 3) {
51         agarre.setText("Presa del destornillador");
52         label_1.setIcon(new ImageIcon(
53             currentPath + "\\imagenes\\mano_4.png"));
54     }
55     // Quinta opción del menú
56     else if (seleccion == 4) {
57         agarre.setText("Palmar esférica pentadigital");
58         label_1.setIcon(new ImageIcon(
59             currentPath + "\\imagenes\\mano_5.png"));
60     } else {
61         seleccion = 0;
62     }
63 }
64 }

```

Para variar el grado de apertura de cada uno de los movimientos de la mano, se utilizaron tres comandos mentales relacionados a colores específicos como: azul para un cierre del 0 %, verde para un cierre del 35 % y amarillo para un cierre del 70 %. Como complemento se utilizó el gesto facial apretar dientes para ejecutar un cierre completo. Esto debido a que el entrenamiento de un cuarto comando mental hubiera aumentado la dificultad del sistema. En el código 3.10 se explica mediante el uso de comentarios, como se realiza la identificación de cada uno de los comandos mentales anteriormente mencionados y del gesto facial apretar dientes.

Código 3.10: Identificación de comandos mentales y gestos faciales en la interfaz 2

```

1     //Se lee la acción cognitiva del usuario
2     caso = EmoState.INSTANCE.ES_CognitivGetCurrentAction(eState);
3     // Se determina el comando mental que se esta ejecutando
4     switch (caso) {
5
6         // Cierre del 0 %
7         case 1:
8             // Se crea el mensaje para la placa de control indicando un cierre del 0 %
9             mensaje = (seleccion + 1) + "," + "0";
10            mensaje_bytes = mensaje.getBytes();
11            paquete = new DatagramPacket(mensaje_bytes,
12                mensaje.length(), address,
13                Integer.parseInt(puerto));
14            // Se envía el mensaje a la placa de control
15            socket.send(paquete);
16
17            // Se cambia el estado de los botones en el menú
18            // Retroalimentación visual para los comandos mentales
19            label_3.setEnabled(true);
20            label_6.setEnabled(false);
21            label_4.setEnabled(false);
22            label_5.setEnabled(false);
23            break;

```

```
24
25 // Cierre del 35 %
26 case 2:
27 // Se crea el mensaje para la placa de control indicando un cierre del 35 %
28 mensaje = (seleccion + 1) + "," + "3.5";
29 mensaje_bytes = mensaje.getBytes();
30 paquete = new DatagramPacket(mensaje_bytes,
31     mensaje.length(), address,
32     Integer.parseInt(puerto));
33 // Se envía el mensaje a la placa de control
34 socket.send(paquete);
35
36 // Se cambia el estado de los botones en el menú
37 // Retroalimentación visual para los comandos mentales
38 label_3.setEnabled(false);
39 label_6.setEnabled(true);
40 label_4.setEnabled(false);
41 label_5.setEnabled(false);
42 break;
43
44 // Cierre del 70 %
45 case 8:
46 // Se crea el mensaje para la placa de control indicando un cierre del 70 %
47 mensaje = (seleccion + 1) + "," + "7";
48 mensaje_bytes = mensaje.getBytes();
49 paquete = new DatagramPacket(mensaje_bytes,
50     mensaje.length(), address,
51     Integer.parseInt(puerto));
52 // Se envía el mensaje a la placa de control
53 socket.send(paquete);
54
55 // Se cambia el estado de los botones en el menú
56 // Retroalimentación visual para los comandos mentales
57 label_3.setEnabled(false);
58 label_6.setEnabled(false);
59 label_4.setEnabled(true);
60 label_5.setEnabled(false);
61 break;
62
63 default:
64 // No se ejecuta ningun tipo de acción
65 System.out.println("Comando no");
66 }
67
68 // Detección del gesto apretar dientes
69 if (EmoState.INSTANCE.ES_ExpressivGetClenchExtent(eState) >= 0.12) {
70
71 // Se crea el mensaje para la placa de control indicando un cierre del 100 %
72 mensaje = (seleccion + 1) + "," + "10";
73 mensaje_bytes = mensaje.getBytes();
74 paquete = new DatagramPacket(mensaje_bytes,
75     mensaje.length(), address,
76     Integer.parseInt(puerto));
77
78 // Se envía el mensaje a la placa de control
79 socket.send(paquete);
80
81 if (activar_gestos == true) {
82 // Se cambia el estado de los botones en el menú
83 // Retroalimentación visual para los comandos mentales
84 label_3.setEnabled(false);
85 label_6.setEnabled(false);
86 label_4.setEnabled(false);
87 label_5.setEnabled(true);
```

```
88     }  
89 }
```

En la figura 3.24 se puede apreciar el diseño final de la interfaz gráfica 2. La cuál se encuentra dividida diferentes secciones: perfil de usuario, área de entrenamiento, menú para la selección del movimiento a ejecutar y visualización de las métricas afectivas. Al igual que en la interfaz gráfica 1, la conexión de la interfaz con el casco ha sido establecida dentro del mismo código. Es decir, una vez que la interfaz gráfica es iniciada, está se conectara de forma automática con el dispositivo captador de ondas cerebrales (emotiv epoc).



Figura 3.24: Diseño final de la interfaz gráfica 2.

Fuente: Autor.

3.4.4 Arquitectura del sistema

El funcionamiento general del sistema se encuentra estructurado de la siguiente forma: se comienza leyendo las ondas cerebrales de cada uno de los usuarios utilizando el dispositivo emotiv epoc, luego se realiza la respectiva calibración del equipo, garantizando así un buen contacto de cada uno de los electrodos del dispositivo. Posteriormente, se realiza un previo entrenamiento de los comandos mentales a utilizar. Dicho entrenamiento puede durar entre 45 minutos a 1 hora dependiendo del nivel de destreza de cada uno de los usuarios. En el caso de la interfaz gráfica 1, el usuario recrea los movimientos de la mano robótica en su mente, y para la interfaz gráfica 2, el usuario relaciona el grado de apertura de la mano con colores específicos, ya que mediante el uso de eventos oculares podrá seleccionar los movimientos de la mano a realizar. Finalizado el entrenamiento para cada uno de los comandos mentales, se envían las señales electroencefalográficas adquiridas por el dispositivo Emotiv Epoc mediante una conexión Bluetooth a un ordenador o Raspberry Pi. De tal forma que los datos sean procesados mediante un programa desarrollado en Java utilizando el kit de desarrollo de software (SDK). En el momento en el que son identificadas las acciones cognitivas que el usuario desea ejecutar, los datos son enviados desde el ordenador a la placa de control mediante una red wifi utilizando el protocolo de comunicación UDP (protocolo de datagrama de usuarios). Con el fin de que la placa de control pueda variar la posición de cada uno de los actuadores de la mano robótica, logrando así el movimiento o agarre deseado por el usuario. La arquitectura del sistema, puede ser visualizada en la figura 3.25.

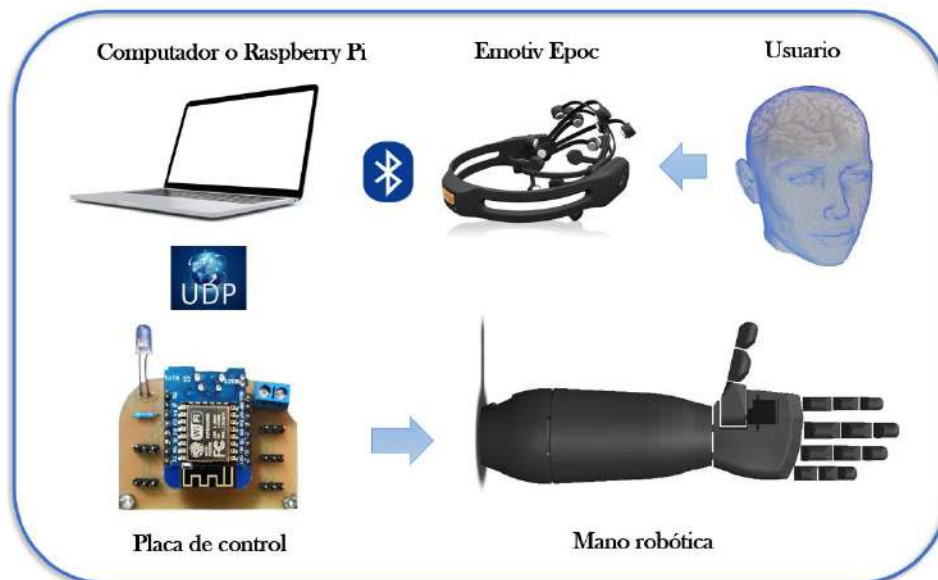


Figura 3.25: Arquitectura del sistema para el control de la mano robótica.

Fuente: Autor.

- **Flujo de información de la Interfaz 1:**

Una vez es iniciada la interfaz gráfica 1, el usuario debe realizar un entrenamiento previo de los comandos mentales a utilizar. Los cuales se encuentran relacionados con los movimientos de la mano robótica (abrir mano, cerrar mano y realizar agarre de pinza fina). Finalizado el entrenamiento cognitivo, la interfaz se conecta mediante una conexión bluetooth con el casco de neuroseñales y se procede a realizar la detección de los comandos mentales del usuario. En el momento en el que un comando mental es detectado, la interfaz determina la intensidad de dicho comando (valor comprendido entre 0 a 10) y envía los datos mediante el protocolo de comunicación UDP a la placa de control, de tal forma que esta pueda ejecutar el movimiento deseado por el usuario, teniendo en cuenta la intensidad de cierre. Es importante resaltar, que entre más fuerte sea el pensamiento ejecutado, mayor será el cierre del movimiento de la mano. Este flujo de información se puede visualizar en la figura 3.26.

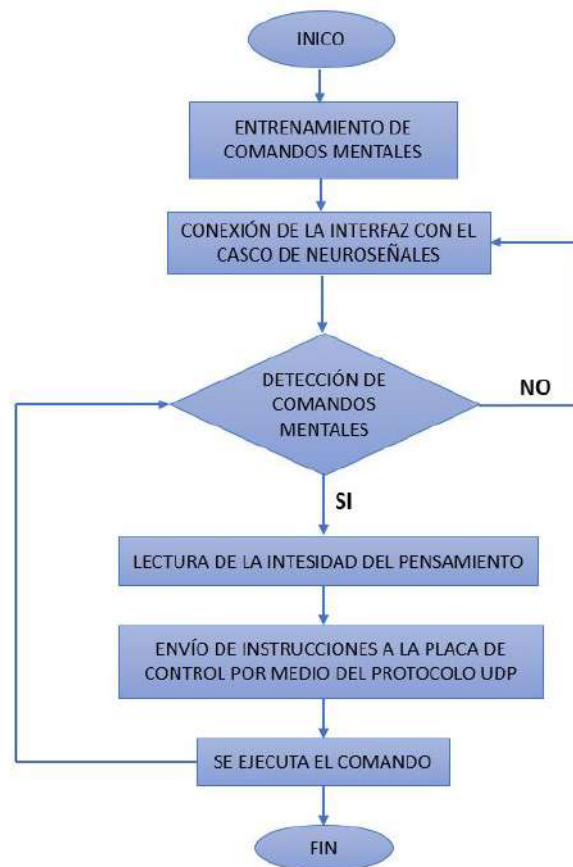


Figura 3.26: Diagrama de flujo para la interfaz gráfica 1.

Fuente: Autor.

- **Flujo de información interfaz 2:**

En el momento en el que se inicia la interfaz gráfica 2, el usuario debe someterse a un entrenamiento cognitivo, el cual se encuentra relacionado con colores específicos que permiten variar la intensidad de cierre de cada movimiento de la mano. El color azul permite realizar un cierre del 0%, el verde un cierre del 35% y el amarillo un cierre del 70%. Adicional a esto, si el usuario desea realizar un cierre completo, deberá ejecutar el gesto facial "apretar dientes". Una vez completado el entrenamiento cognitivo, se establece la conexión entre la interfaz 2 y el casco de neuroseñales, y la interfaz inicia el proceso de detección de comandos y gestos faciales. En el momento en el que alguno de estos es detectado, se envían unas instrucciones específicas a la placa de control, de manera que esta pueda ejecutar el movimiento deseado por el usuario junto con la intensidad de cierre requerida. En la figura 3.27, se puede apreciar este flujo de información.



Figura 3.27: Diagrama de flujo para la interfaz la interfaz gráfica 2.

Fuente: Autor.

4 Resultados

4.1 Interfaz 1

Se seleccionaron cinco estudiantes de la Universidad de Pamplona (entre 20 y 24 años de edad), para poder validar el correcto funcionamiento del sistema utilizando la interfaz gráfica 1. El tiempo empleado por los usuarios para el entrenamiento de los comandos mentales estuvo comprendido entre 40 minutos y 1 hora. Algunos de los usuarios se adaptaron rápidamente al uso del sistema, mientras que otros presentaron algunas dificultades durante el entrenamiento. Debido a factores como el estrés y la frustración por no poder ejecutar ningún tipo de comando mental en el inicio de los entrenamientos. Es importante señalar que entre más definido este el pensamiento, más fácil será distinguirlo y procesarlo a través del sistema desarrollado. Los resultados mostrados en la tabla 4.1 indican el tiempo empleado por cada usuario para realizar el entrenamiento cognitivo.

| Usuario | Tiempo de entrenamiento |
|---------|-------------------------|
| 1 | 43 minutos |
| 2 | 44 minutos |
| 3 | 56 minutos |
| 4 | 41 minutos |
| 5 | 52 minutos |

Tabla 4.1: Tiempo de entrenamiento empleado por cada usuario.

Fuente: Autor.

Para facilitar el entrenamiento cognitivo, a medida que se entrenaba un comando en específico este se visualizaba en el área de entrenamiento dentro de la interfaz, de tal forma que el usuario pudiera tener un tipo de retroalimentación visual, permitiéndole lograr a futuro un mejor desempeño en la ejecución de los comandos mentales. En la tabla 4.2 se muestra cada uno de los comandos mentales relacionados a su imagen de referencia.




| Comando Mental | Imagen de referencia |
|----------------------|--|
| Abrir mano |  |
| Cerrar mano |  |
| Agarre de pinza fina |  |

Tabla 4.2: Relación de cada comando mental con su imagen de referencia.

Fuente: Autor.

Cada uno de los usuarios, realizo 7 pruebas por cada secuencia de comandos mentales, esto con el objetivo de poder determinar el porcentaje de acierto de cada una de las acciones cognitivas. Las secuencias establecidas fueron las siguientes:

- **Secuencia 1:**

Cerrar y abrir la mano robótica.

- **Secuencia 2:**

Cerrar y abrir la mano robótica y posteriormente realizar el agarre de pinza fina.

- **Secuencia 3:**

Ejecutar el agarre de pinza fina y posteriormente abrir y cerrar la mano robótica.

Las pruebas se realizaron a lo largo de tres semanas, ajustando el tiempo de disposición de cada usuario. Cabe resaltar que uno de los usuarios durante el entrenamiento de los comandos mentales, decidió relacionar sus pensamientos con movimientos que recreaban utilizando su mano izquierda. Esto se debe a que cada persona piensa de una forma diferente y puede que la estrategia utilizada por un usuario no funcione en otra persona. En la figura 4.1 y 4.2 se observa como uno de los usuarios recrea con su mano derecha, algunas de las acciones cognitivas.



Figura 4.1: Usuario ejecutando el comando mental agarre de pinza fina.
Fuente: Autor.



Figura 4.2: Usuario ejecutando el comando mental cerrar mano.
Fuente: Autor.

En la tabla 4.3 se ilustran los resultados obtenidos por los 5 usuarios que realizaron el experimento. Estos resultados fueron clasificados en aciertos y fallos por cada secuencia realizada, con el objetivo de poder determinar el porcentaje de acierto para cada secuencia y validar el correcto funcionamiento del sistema.

Se puede observar una semejanza en los datos de los usuarios 1 y 2, los cuáles a simple vista demuestran un buen desempeño. Esto se debe a que son usuarios experimentados en el uso del sistema, los cuáles han tenido la oportunidad de interactuar por más de una semana con el dispositivo captador de ondas cerebrales. El objetivo principal de esto, era observar que tanto variaba el porcentaje de acierto entre una personas que jamás ha tenido contacto con este tipo de sistemas, con respecto a otras que ya han tenido una interacción previa.

| Usuario | Tipo de secuencia | Aciertos | Fallos |
|---------|-------------------|----------|--------|
| 1 | Secuencia 1 | 6 | 1 |
| | Secuencia 2 | 6 | 1 |
| | Secuencia 3 | 5 | 2 |
| 2 | Secuencia 1 | 7 | 0 |
| | Secuencia 2 | 6 | 1 |
| | Secuencia 3 | 6 | 1 |
| 3 | Secuencia 1 | 4 | 3 |
| | Secuencia 2 | 2 | 5 |
| | Secuencia 3 | 1 | 6 |
| 4 | Secuencia 1 | 5 | 2 |
| | Secuencia 2 | 3 | 4 |
| | Secuencia 3 | 4 | 3 |
| 5 | Secuencia 1 | 5 | 2 |
| | Secuencia 2 | 4 | 3 |
| | Secuencia 3 | 2 | 4 |

Tabla 4.3: Resultados de las pruebas realizadas a cada usuario.

Fuente: Autor.

Se evidencia en la figura 4.3, que la mayoría de las tasas de éxito son superiores al 50%, lo que demuestra una buena destreza en la mayoría de los usuarios para adaptarse a este tipo de control. El usuario 3 obtuvo un porcentaje del 14,29% de acierto (muy bajo) en la secuencia 3. Esto se debe a que la mayoría de los usuarios no están acostumbrados a controlar objetos físicos utilizando sus pensamientos, lo cual representa para muchos un nuevo desafío. Sin embargo, existe otros factores adicionales que influyen en el desempeño, como el nivel de habilidad de la persona para ejecutar los comandos mentales, la intensidad con que se realiza la acción cognitiva y el nivel de conexión del casco con respecto el cuero cabelludo.

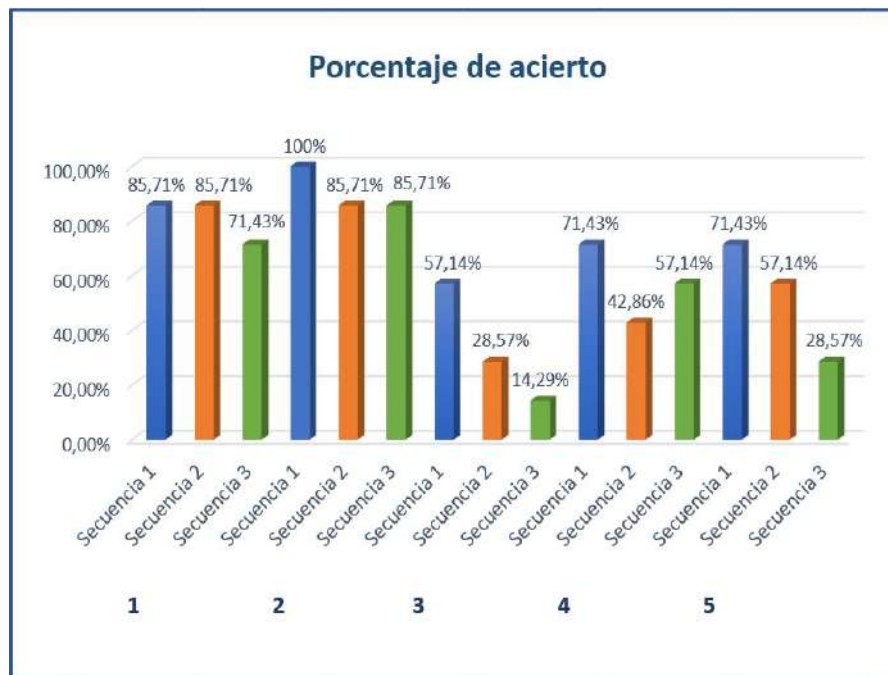


Figura 4.3: Diagrama de barras del porcentaje de acierto de cada secuencia.
Fuente: Autor.

En la tabla 4.4 se presentan algunos indicadores estadísticos de las tasas de éxito obtenidas por cada usuario, como la media, la mediana y la moda. Se aprecia un nivel de éxito promedio de más del 70% para la secuencia 1, del 60% para la secuencia 2 y superior al 50 % para la secuencia 3. La razón por la cual los usuarios obtuvieron un mejor desempeño realizando la secuencia 1, fue debido a que esta involucraba un menor número de comandos mentales. Mientras que las secuencias 2 y 3, exigían un nivel mayor de concentración y destreza. A medida que el número de comandos mentales incrementa, el sistema se vuelve más complejo de utilizar. En la figura 4.4 se pueden observar gráficamente los indicadores estadísticos.

| Índice estadístico | Secuencia 1 | Secuencia 2 | Secuencia 3 |
|--------------------|-------------|-------------|-------------|
| Media | 77,14% | 60,00% | 51,43% |
| Mediana | 71,43% | 57,14% | 57,14% |
| Moda | 71,43% | 85,71% | - |

Tabla 4.4: Índices estadísticos de las pruebas realizadas.
Fuente: Autor.

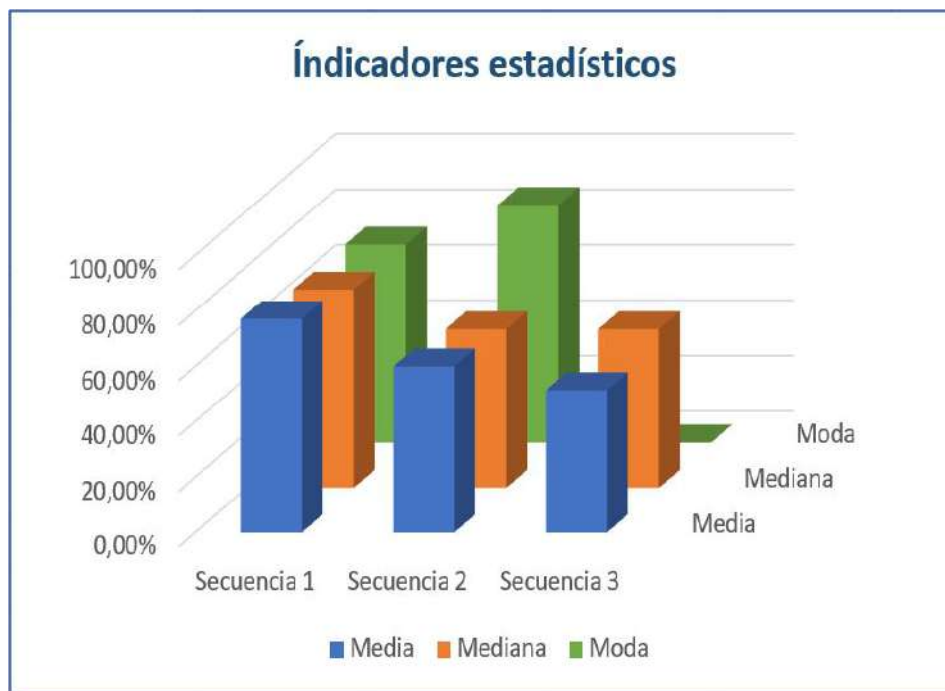


Figura 4.4: Diagrama de barras de los indicadores estadísticos basados en el porcentaje de acierto.

Fuente: Autor.

El desempeño de los usuarios se puede ver afectado por algunos factores como: ruidos externos que afectan la concentración de los usuarios o el estrés provocado por no poder ejecutar una acción cognitiva.

Teniendo en cuenta lo anteriormente mencionado, se procede a realizar la captura de algunas métricas afectivas (estrés y concentración) de los usuarios, mientras estos ejecutan las acciones cognitivas de la secuencia 3 (la más difícil de realizar). Permitiendo de esta forma obtener un análisis más completo sobre el desempeño de los usuarios.

En la figura 4.5 se ilustran los resultados correspondientes al promedio de estrés y concentración para cada usuario. Se observa que tanto el usuario 1 como el 2, obtuvieron un valor promedio de estrés comprendido entre 3 y 5 en una escala del 0 al 10, lo cual es bastante bajo. Esto se debe a que son usuarios experimentados en el uso del sistema, por lo cual el esfuerzo que deben realizar es menor comparado con el de los demás usuarios. En el caso del usuario 3, se observa que su valor medio de concentración es el menor de todos, lo cual puede ser ocasionado por algún tipo de ruido externo. Es posible que el bajo nivel de concentración influya en los resultados obtenidos para este usuario. Pues su desempeño para la ejecución de la secuencia 3, fue el más bajo de todos.

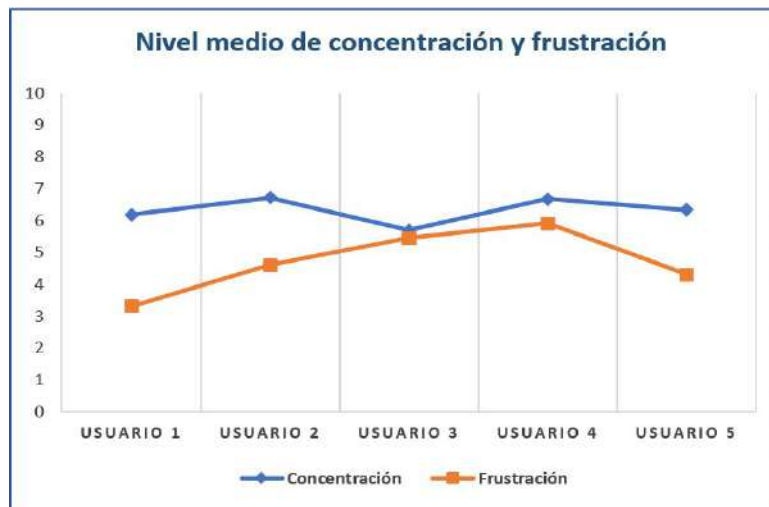


Figura 4.5: Nivel medio de concentración y frustración de los usuarios de la interfaz 1.

Fuente: Autor.

4.2 Interfaz 2

La interfaz 2 combina la parte cognitiva con la parte expresiva, brindando a los usuarios la posibilidad de ejecutar un mayor número de movimientos con la mano robótica. En este caso los comandos mentales están relacionados a colores en específicos (ver tala 4.5), los cuáles permiten variar la intensidad de cierre de cada movimiento.




| Comando mental | Imagen de referencia | Intensidad de cierre |
|----------------|---|----------------------|
| Azul |  | Cierre del 0% |
| Verde |  | Cierre del 35% |
| Amarillo |  | Cierre del 70% |

Tabla 4.5: Comandos mentales para variar la intensidad de cierre de la mano robótica.

Fuente: Autor.

Para ejecutar el cierre del 100% de cada uno de los movimientos de la mano, se utiliza el gesto facial apretar dientes. Debido a que la inclusión de un cuarto comando mental aumenta de forma considerable los tiempos de entrenamiento (más de hora y media) y vuelve más complejo el uso del sistema.

Para las realizar las pruebas del sistema, se utilizan los mismos usuarios que realizaron las pruebas de la interfaz 1. En la tabla 4.6 se presenta el tiempo empleado por cada uno de los usuarios para realizar el entrenamiento de los comandos mentales. En el caso de los gestos faciales, no se requiere de un entrenamiento previo.

| Usuario | Tiempo de entrenamiento |
|---------|-------------------------|
| 1 | 39 minutos |
| 2 | 42 minutos |
| 3 | 50 minutos |
| 4 | 40 minutos |
| 5 | 47 minutos |

Tabla 4.6: Tiempo de entrenamiento empleado por cada usuario en la interfaz 2.

Fuente: Autor.

Mediante el uso de eventos oculares el usuario puede desplazarse dentro del menú de la interfaz 2, de tal forma que pueda escoger el movimiento a realizar con la mano robótica. Para avanzar dentro del menú, el usuario debe mirar hacia el lado derecho, en dado caso que requiera devolverse a la opción anterior, solo debe mover sus ojos hacia el lado izquierdo, y para seleccionar el movimiento a ejecutar, basta solo con mirar al frente. En la tabla 4.7 se ilustran los movimientos de la mano robótica que están disponibles en el menú.

| Tipo de movimiento | Imagen |
|-------------------------------|---|
| Prensión digito palmar |  |
| Cerrar mano |  |
| Agarre de pinza fina |  |
| Presión del destornillador |  |
| Palmar esférica penta digital |  |

Tabla 4.7: Tipos de movimientos disponibles en el menú de la interfaz 2.

Fuente: Autor.

Con el fin de determinar qué tan preciso es el reconocimiento de los gestos faciales por parte del sistema, se procede a realizar una secuencia compuesta de diez intentos. En donde el usuario deberá observar diez veces hacia el lado izquierdo, diez veces hacia el lado derecho y apretar los dientes; tomando un descanso de tres segundos entre gesto facial ejecutado. La expresión facial apretar dientes, es la encargada de realizar el cierre del 100% de los movimientos de la mano. Este gesto facial sustituye el uso de un cuarto comando mental. En la tabla 4.8 se pueden observar los resultados obtenidos.

| Usuario | Mirar derecha | Mirar izquierda | Apretar dientes |
|---------|---------------|-----------------|-----------------|
| 1 | 100 % | 90% | 100% |
| 2 | 80 % | 80% | 90% |
| 3 | 100% | 70% | 70% |
| 4 | 70% | 60% | 80% |
| 5 | 70% | 70% | 90% |

Tabla 4.8: Porcentaje de reconocimiento para los eventos oculares y el gesto apretar dientes.
Fuente: Autor.

Para evaluar el desempeño general del sistema, cada uno de los usuarios realizó 5 pruebas por cada tipo de movimiento. A continuación, se presentan en la tabla 4.9 el significado de cada abreviación, la cual se encuentra relacionada al tipo de movimiento y la acción que se desea ejecutar. Esto con el fin de brindar una mejor interpretación al momento de observar los resultados expuestos en la tabla 4.10.

| Abreviatura | Tipo de movimiento | Tipo de acción ejecutada |
|----------------|-------------------------------|--------------------------|
| p-dp-30 | Prensión digito palmar | Cierre del 30% |
| c-m-30 | Cerrar mano | Cierre del 30% |
| p-f-70 | Agarre de pinza fina | Cierre del 70% |
| p-d-70 | Presa del destornillador | Cierre del 70% |
| p-p-100 | Palmar esférica penta digital | Cierre del 100% |

Tabla 4.9: Abreviatura de cada uno de los movimientos de la mano robótica.
Fuente: Autor.



Figura 4.6: Usuario ejecutando el movimiento pinza fina con cierre del 0%.

Fuente: Autor.



Figura 4.7: Usuario ejecutando el comando mental pinza fina con cierre del 70%.

Fuente: Autor.

Las pruebas fueron realizadas en el transcurso de 3 semanas ajustándose al tiempo de disponibilidad de cada usuario. En la figura 4.6 se visualiza como dentro de la interfaz 2, se encuentra seleccionada la acción de pinza fina, pero debido a que el usuario está pensando en el color azul, el tipo de acción ejecutada es un cierre del 0%, razón por la cual la mano permanece abierta. Para la figura 4.7 el usuario realiza el movimiento p-f-70 (agarre de pinza fina con cierre del 70%), por lo cual en la interfaz se activa el color amarillo (intensidad de cierre del 70%).

La tabla 4.10 se muestra los resultados obtenidos de las pruebas realizadas a los usuarios. En donde se obtiene el porcentaje de acierto de acción ejecutada por usuario.

| Usuario | Movimiento | Aciertos | Fallos | Porcentaje acierto (%) |
|---------|------------|----------|--------|------------------------|
| 1 | p-dp-30 | 4 | 1 | 80 |
| | c-m-30 | 5 | 0 | 100 |
| | p-f-70 | 4 | 1 | 80 |
| | p-d-70 | 3 | 2 | 60 |
| | p-p-100 | 5 | 0 | 100 |
| 2 | p-dp-30 | 5 | 0 | 100 |
| | c-m-30 | 4 | 1 | 80 |
| | p-f-70 | 3 | 2 | 60 |
| | p-d-70 | 4 | 1 | 80 |
| | p-p-100 | 4 | 1 | 80 |
| 3 | p-dp-30 | 3 | 2 | 60 |
| | c-m-30 | 4 | 1 | 80 |
| | p-f-70 | 3 | 2 | 60 |
| | p-d-70 | 2 | 3 | 40 |
| | p-p-100 | 4 | 1 | 80 |
| 4 | p-dp-30 | 3 | 2 | 60 |
| | c-m-30 | 4 | 1 | 80 |
| | p-f-70 | 3 | 2 | 60 |
| | p-d-70 | 3 | 2 | 60 |
| | p-p-100 | 4 | 1 | 80 |
| 5 | p-dp-30 | 4 | 1 | 80 |
| | c-m-30 | 3 | 2 | 60 |
| | p-f-70 | 2 | 3 | 40 |
| | p-d-70 | 3 | 2 | 60 |
| | p-p-100 | 3 | 2 | 60 |

Tabla 4.10: Resultados de las pruebas realizadas a cada usuario utilizando la interfaz 2.

Fuente: Autor.

Para obtener una mejor representación de los datos expuestos en la tabla 4.10, se calcula la media y mediana del porcentaje de acierto de cada uno de los movimientos ejecutados por los usuarios. Su representación gráfica se puede observar en la figura 4.8

Se evidencia en la figura 4.8, que el nivel medio de la mayoría de las tasas de éxito es del 60% o superiores, lo que demuestra un buen rendimiento del sistema y buenas habilidades en la mayoría de los usuarios para controlar la intensidad de cierre de los diferentes movimientos de la mano robótica. Se observa también que el gesto facial apretar dientes, tiene un nivel medio de acierto del 80%. Siendo reconocido sin problema por la mayoría de los usuarios.

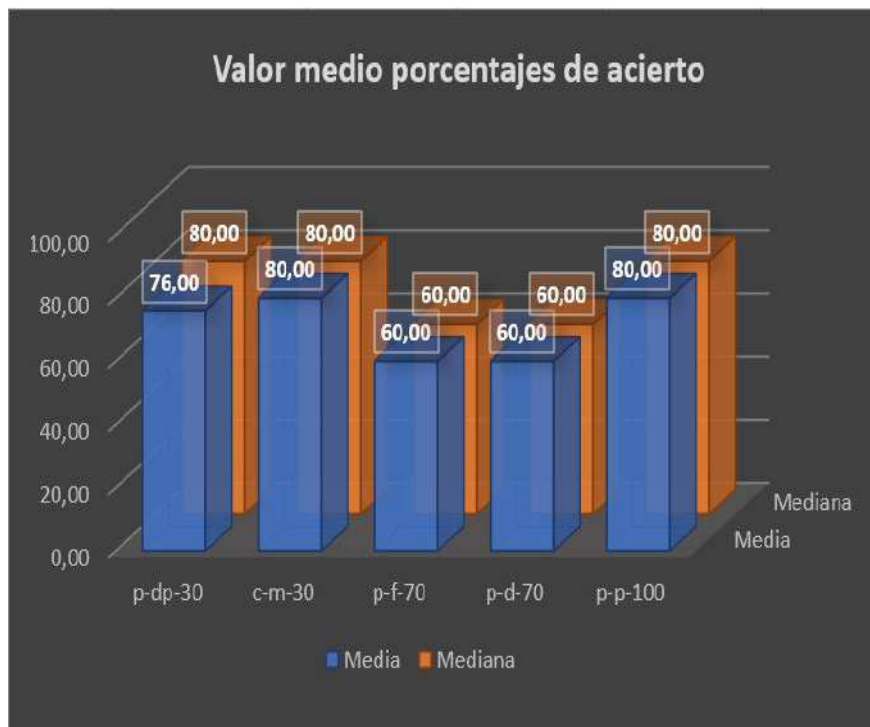


Figura 4.8: Nivel medio del porcentaje de acierto de cada comando ejecutado.

Fuente: Autor.

Con el fin de estimar como se sintieron cada uno de los usuarios que realizaron el experimento, se decidió capturar algunas métricas expresivas relacionadas con la concentración y la frustración. Cabe aclarar que esta prueba solo se realizó para la acción cognitiva p-f-70 (agarre de pinza fina con cierre del 70%). Debido que es considerado el movimiento más difícil de realizar por parte de los usuarios.

En la tabla 4.11 se puede observar algunos indicadores estadísticos relacionados a las métricas recolectadas, como la media, la mediana, la varianza y la desviación estándar. Se evidencia que la desviación estándar obtenida para la concentración es inferior a 0.8 en la mayoría de los usuarios. Lo cual indica que durante la prueba mantuvieron un nivel de concentración muy similar al valor medio calculado para cada uno de ellos. En el caso de la frustración se evidencia que el valor medio en la mayoría de los usuarios es inferior a 3.6, por lo que se deduce que no hubo algún tipo de afectación emocional en el desempeño de los usuarios.

Con el fin de obtener una mejor representación de los datos expuestos en la tabla 4.11, se procede a realizar un gráfico de líneas (ver figura 4.9), en el cual se ilustra el valor medio de atención y frustración de cada uno de los usuarios que realizaron la prueba.

| Usuario | Estado | Media | Varianza | Desviación estándar |
|---------|---------------|-------|----------|---------------------|
| 1 | Concentración | 6,71 | 0,39 | 0,63 |
| | Frustración | 3,52 | 0,02 | 0,14 |
| 2 | Concentración | 7,11 | 0,73 | 0,85 |
| | Frustración | 3,41 | 0,16 | 0,39 |
| 3 | Concentración | 6,21 | 0,24 | 0,49 |
| | Frustración | 3,50 | 0,03 | 0,17 |
| 4 | Concentración | 5,80 | 0,11 | 0,33 |
| | Frustración | 3,26 | 0,01 | 0,11 |
| 5 | Concentración | 5,71 | 0,56 | 0,75 |
| | Frustración | 3,29 | 0,01 | 0,08 |

Tabla 4.11: Índices estadísticos de las métricas afectivas capturadas.
Fuente: Autor.

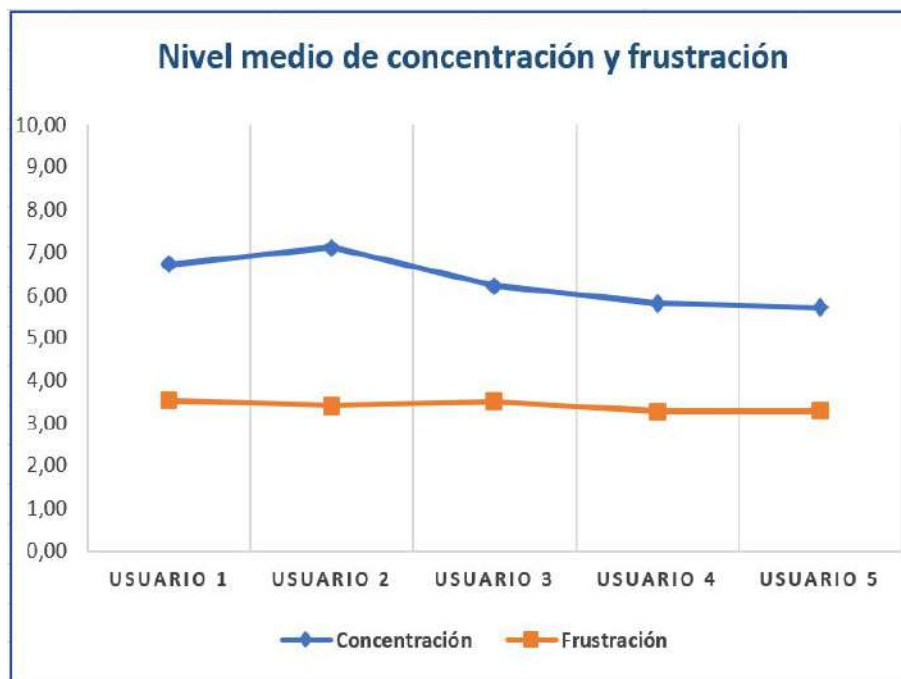


Figura 4.9: Nivel medio de concentración y frustración de los usuarios que ejecutaron la acción p-f-7.

Fuente: Autor.

5 Conclusiones

Teniendo en cuenta los resultados obtenidos, se determina que el control mental funciona de manera efectiva. Y que el buen desempeño del sistema, está sujeto directamente a los pensamientos que utiliza cada uno de los usuarios para relacionar los movimientos de la mano robótica. En el caso de la interfaz gráfica 1, uno de los usuarios decidió utilizar una estrategia adicional para el entrenamiento de sus pensamientos, que consistía en imitar con su mano izquierda el movimiento a ejecutar, lo cual le permitió adaptarse de forma rápida al uso del sistema. El tiempo empleado para el entrenamiento de los comandos mentales, el buen contacto y posicionamiento de los electrodos, también representan una parte fundamental, ya que un mal contacto puede ocasionar ruido en las señales y generar errores en el control.

En el caso de la interfaz gráfica 1, se evidencia un valor medio superior al 70% de acierto para la secuencia 1, de 60% para la secuencia 2 y superior al 50% para la secuencia 3. Lo que demuestra que a medida que aumenta el número de comandos mentales a utilizar, el sistema se va volviendo más complejo. Además de que los tiempos de entrenamiento aumentan de forma considerable, llegando a durar hasta 1 hora y media para el entrenamiento de un cuarto comando mental. Lo cual cansa y fatiga mentalmente al usuario.

Mediante la captura de algunas métricas como la concentración y el estrés, se puede evidenciar como el usuario 3, obtuvo un valor de aproximadamente 5, en una escala de 0 a 10, mientras realizaba las pruebas para la secuencia 3. Esto puede deberse a diversos factores, como el no poder ejecutar de forma correcta los comandos mentales de la mano. Lo cual se refleja en su porcentaje de acierto obtenido, siendo este de apenas un 28,57%. Aunque es importante resaltar, que puede haber más factores involucrados.

Para la interfaz gráfica 2, los usuarios pudieron ejecutar un mayor número de movimientos con la mano robótica. Debido a que la selección de estos, se realizaba mediante un menú controlado por medio de eventos oculares. Los cuáles tuvieron un porcentaje de reconocimiento muy bueno, superando en la mayoría de los casos el 70%. Cabe destacar, que la detección de los gestos faciales se encuentra sujeto a la habilidad de los usuarios para ejecutarlos. En el caso de la parte cognitiva, los usuarios obtienen excelentes resultados (tasas de acierto superiores al 60% en la mayoría de las acciones ejecutadas), permitiéndoles así controlar diferentes intensidades de cierre para cada movimiento de la mano robótica.

Las métricas de concentración y frustración en la interfaz 2, fueron capturadas para cada usuario mientras estos ejecutaban la acción p-f-7 (agarre de pinza fina con cierre del 70%). De lo cual se obtuvo un valor de concentración medio superior a 5, para cada uno de los usuarios, lo que indica que estuvieron concentrados la mayor parte de la prueba. Además, la variabilidad de estos datos fue muy poca, ya que la desviación estándar que se obtuvo para cada uno de los usuarios fue inferior a 0,7.

Algunos de los estudiantes que realizaron las pruebas del sistema tenían el cabello largo, pero esto no represento algún tipo de problema con el contacto de los electrodos. Esto se debe a las propiedades físicas del cabello, ya que algunos pueden convertirse en buenos conductores y facilitar la adquisición de las señales electroencefalográficas.

Finalmente, se puede decir que el control mental representa un nuevo reto para los usuarios, pero a medida que se va utilizando y entrenando, este se vuelve un control más natural y preciso, que brinda grandes posibilidades de control a las personas que a diario hacen uso de prótesis de mano robóticas para suplir funciones de las que carecen. Además, el dispositivo utilizado posee la ventaja de ser de bajo costo y de fácil adquisición y no requiere de ningún tipo de cirugía para poder utilizarlo, lo cual representa una gran alternativa a los sistemas invasivos actualmente empleados para el control de prótesis de mano.

Bibliografía

- [1] J Tejeiro Martínez. Indicaciones de la electroencefalografía. *Medicine-Programa de Formación Médica Continuada Acreditado*, 9(75):4851–4853, 2007. ISSN 0304-5412. doi: [https://doi.org/10.1016/S0211-3449\(07\)75460-7](https://doi.org/10.1016/S0211-3449(07)75460-7). URL <http://www.sciencedirect.com/science/article/pii/S0211344907754607>.
- [2] Andrea Biasiucci, Benedetta Franceschiello, and Micah M. Murray. Electroencephalography. *Current Biology*, 29(3):R80 – R85, 2019. ISSN 0960-9822. doi: <https://doi.org/10.1016/j.cub.2018.11.052>. URL <http://www.sciencedirect.com/science/article/pii/S0960982218315513>.
- [3] ROBERTO ALEJANDRO ALAGIA GIMENO. Procesamiento de artefactos en eeg para aplicaciones de comunicación y control. 2018.
- [4] M Balcells and V Cisteré. Historia de la electroencefalografía en españa: introducción y evolución. pages 38 – 42, 2014.
- [5] Ruby Villar. La neuroestimulación o la neuroestimulación dinámica, 2016. URL <https://www.rvd-psychologue.com/es/la-neuroestimulacion.html>. [Web; accedido el 07-04-2020].
- [6] Carlos Antonio Bulnes Domínguez. Desarrollo de un software libre para la lectura, graficación y registro en tiempo real de señales provenientes de ondas cerebrales. 2015.
- [7] Alejandro Mol Gómez-Vázquez et al. La divulgación oral de la ingeniería en inglés: un ejemplo de las conferencias ted. *Escuela Técnica Superior de Ingeniería Industrial*, 2018. URL <http://hdl.handle.net/10317/7583>.
- [8] Siuly Siuly, Yan Li, and Yanchun Zhang. Eeg signal analysis and classification. *IEEE Trans Neural Syst Rehabil Eng*, 11:141–4, 2016.
- [9] Antonio Martínez Milagros Merino. Electroencefalografía convencional en pediatría: técnica e interpretación. *Hospital Universitario La Paz*, 2007.
- [10] JW Cornejo Ochoa and ME Toro Pérez. Las epilepsias del lóbulo temporal. *Liga Cubana contra la Epilepsia*, 2011.
- [11] F Ramos-Argüelles, G Morales, S Egozcue, RM Pabón, and MT Alonso. Técnicas básicas de electroencefalografía: principios y aplicaciones clínicas. 32:69–82, 2009.

-
- [12] Jonathan González Sánchez. Técnicas de toma de datos y análisis de electroencefalografía. *Obtenido de <https://opera.eii.us.es/archivos/sinergia/entregables/2013-2014/Grupo9/Grupo9Memoria1.pdf> (Recuperado 04/07/2015)*, 2014.
- [13] Mario E Avila Perona and Bioing Pablo F Diez. Bioinstrumentación ii. *Universidad Nacional de San Juan*, 2006.
- [14] Andrea Noelia Bermúdez Cicchino. Técnicas de procesamiento de eeg para detección de eventos. 2013.
- [15] Dipali Bansal and Rashima Mahajan. Chapter 2 - eeg-based brain-computer interfacing (bci). pages 21 – 71, 2019. doi: <https://doi.org/10.1016/B978-0-12-814687-3.00002-8>. URL <http://www.sciencedirect.com/science/article/pii/B9780128146873000028>.
- [16] T. Talamillo García. Manual básico para enfermeros en electroencefalografía. pages 29 – 33, 2011. URL <http://www.sspa.juntadeandalucia.es/servicioandaluzdesalud/huvvsites/default/files/revistas/ED-094-07.pdf>.
- [17] Josefina Ricardo Garcell and Walfred Rueda Medina. Electroencefalograma y potenciales relacionados con eventos en el trastorno obsesivo compulsivo. *Salud mental*, 32(2):173–181, 2009.
- [18] Elena Isabel Rodríguez Martínez. Indicadores de maduración cerebral y su relación con la memoria de trabajo. *Universidad de Sevilla*, 2014.
- [19] Elena María Mosquera Guerrero. Análisis e identificación de potenciales evocados en el electroencefalograma. *Universidad de Sevilla*, 2019.
- [20] Walter Freeman and Rodrigo Quian Quiroga. *Imaging brain function with EEG: advanced temporal and spatial analysis of electroencephalographic signals*. Springer Science & Business Media, 2012.
- [21] Saeid Sanei and Jonathon A Chambers. *EEG signal processing*. John Wiley & Sons, 2013.
- [22] Henríquez Muñoz and Claudia Nureibis. Estudio de técnicas de análisis y clasificación de señales eeg en el contexto de sistemas bci (brain computer interface). 2014.
- [23] Germán Rodríguez Bermúdez, Pedro José García Laencina, Domitien Brizion, Joaquín Roca Dorda, et al. Adquisición, procesamiento y clasificación de señales eeg para diseño de sistemas bci basados en imaginación de movimiento. 2013.
-

-
- [24] Leonardo José Gómez Figueroa. Análisis de señales eeg para detección de eventos oculares, musculares y cognitivos. 2016.
- [25] S. Jatzev M. Gaertner T.O. Zander, C. Kothe. Enhancing human-computer interaction with input from active and passive brain-computer interfaces. pages 181 – 199, 2010.
- [26] F.Schettini A.Brouwer D.Mattila F. Babiloni F.Cincotti F. Aloise, I. Lasorsa. Multimodal stimulation for a p300-based bci. pages 128 – 130, 2007.
- [27] D.J.McFarland J.R.Wolpaw. Multichanneleeg-basedbrain-computercommunication. pages 444 – 449, 1994.
- [28] C. Kothe T.O. Zander. Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. 2011.
- [29] Chang-Hwan Im. Computational eeg analysis, methods and applications. 2018. doi: <https://doi.org/10.1007/978-981-13-0908-3>.
- [30] Szczepan Paszkiel. *Analysis and Classification of EEG Signals for Brain-computer Interfaces*. Springer, 2020.
- [31] Rahul Raj, Suman Deb, and Paritosh Bhattacharya. Brain computer interfaced single key omni directional pointing and command system: a screen pointing interface for differently-abled person. *Procedia computer science*, 133:161–168, 2018.
- [32] Rami N. Khushaba, Luke Greenacre, Sarath Kodagoda, Jordan Louviere, Sandra Burke, and Gamini Dissanayake. Choice modeling and the brain: A study on the electroencephalogram (eeg) of preferences. *Expert Systems with Applications*, 39(16):12378 – 12388, 2012. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2012.04.084>. URL <http://www.sciencedirect.com/science/article/pii/S0957417412006811>.
- [33] Tomasz Stach, Natalia Browarska, and Aleksandra Kawala-Janik. Initial study on using emotiv epoc+ neuroheadset as a control device for picture script-based communicators. *IFAC-PapersOnLine*, 51(6):180 – 184, 2018. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2018.07.150>. URL <http://www.sciencedirect.com/science/article/pii/S2405896318308966>. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018.
- [34] Dipali Bansal and Rashima Mahajan. Chapter 3 - real-time eeg acquisition. pages 73 – 105, 2019. doi: <https://doi.org/10.1016/B978-0-12-814687-3.00003-X>. URL <http://www.sciencedirect.com/science/article/pii/B978012814687300003X>.
-

-
- [35] Benjamín Miguel Villegas Méndez, Rojas Fernández, and Marcelo Greby. Interfaz cerebro ordenador bci mediante el uso de emotiv insight. *Acta Nova*, 9(1):3–31, 2019.
- [36] A Pérez González, M Jurado Tovar, and JL Sancho Bru. Fuerzas de contacto entre mano y objeto en el agarre cilíndrico: comparación de dos técnicas de medición.
- [37] Christian Augusto Silva Castellanos, Jhon Edison Muñoz Riaños, Diego Alexander Garzón Alvarado, Nancy Stella Landínez Parra, and Octavio Silva Caicedo. Diseño mecánico y cosmético de una prótesis parcial de mano. *Revista Cubana de Investigaciones Biomédicas*, 30(1):15–41, 2011.
- [38] Brito Guaricela, Joffre Luis, Marlon Xavier Quinde Abril, and José David Cuzco Patiño. Diseño, construcción e implementación de una prótesis biomecánica de mano derecha. 2013.
- [39] Alexander Rincón Jiménez. Prototipo de exoesqueleto para la rehabilitación motora de los dedos de la mano humana. 2015.
- [40] CA Quinayés Burgos and Oscar Andrés Vivas Albán. Diseño y construcción de una prótesis robotica de mano funcional adaptada a varios agarres. *Popayán, enero del*, 2010.
- [41] R. Ceres, J.L. Pons, L. Calderón, and J. Moreno. La robótica en la discapacidad. desarrollo de la prótesis diestra de extremidad inferior manus-hand. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(2):60 – 68, 2008. ISSN 1697-7912. doi: [https://doi.org/10.1016/S1697-7912\(08\)70145-6](https://doi.org/10.1016/S1697-7912(08)70145-6). URL <http://www.sciencedirect.com/science/article/pii/S1697791208701456>.
- [42] Salvador Cobos Guzman. Análisis de tareas de manipulación virtual basadas en modelos reducidos de la mano humana. 2010.
- [43] ÁLVARO ALFREDO BASOALTO CALVANESE. Desarrollo de prótesis de mano con sistema de agarre adaptativo para uso general en amputaciones a nivel de desarticulación de muñeca. 2018.
- [44] Johanna Carolina Chico Moreno et al. Desarrollo de prototipo de prótesis de mano emulada en un ambiente virtual. 2016.
- [45] Jair L Loaiza and Nelson Arzola. Evolución y tendencias en el desarrollo de prótesis de mano. *Dyna*, 78(169):191–200, 2011.
- [46] Daniel Lopesino Rivera. Análisis y mejora del diseño de una mano protésica. *Escuela Técnica de Ingenieros Industriales, Universidad Politécnica de Madrid*, 2018.
-

-
- [47] E Garibay Castañeda, RI Flores Luna, F Cuenca Jimenez, and JM Dorador González. Diseño de un sistema amplificador de fuerza para prótesis mecánica. 2012.
- [48] Jofre Brito, Marlon Quinde, David Cusco, and John Calle. Estudio del estado del arte de las prótesis de mano. *Ingenius*, 06 2013. doi: 10.17163/ings.n9.2013.08.
- [49] Hussein Mahdi. 3d printed myoelectric prosthetic arm. 2014. doi: https://www.academia.edu/39599212/3D_Printed_Myoelectric_Prosthetic_Arm_3D_Printed_Myoelectric_Prosthetic_Arm_i.
- [50] Varios Autores. Como funciona un servomotor. url <https://como-funciona.co/un-servomotor/>.
- [51] SigmaElectrónica. Servomotor mg995. url <https://www.sigmaelectronica.net/producto/mg995/>.
- [52] Servocity. Hs-485hb servo. url <https://www.servocity.com/hs-485hb-servo>.
- [53] SuperRobótica. Servo motor hitec hs422 s330165. url <http://www.superrobotica.com/S330165.htm>.
- [54] geekfactory.mx. Mg90s tower pro micro servo motor metálico. url <https://www.geekfactory.mx/tienda/motores-y-controladores/mg90s-tower-pro-micro-servo-motor-metalico/>.
- [55] taloselectronics. Servomotor tower pro sg92r. url <https://www.taloselectronics.com/products/servomotor-tower-pro-sg92r>.
- [56] Luis Llamas. Wemos d1 mini, placa de desarrollo con esp8266. url <https://www.luisllamas.es/wemos-d1-mini-una-genial-placa-de-desarrollo-con-esp8266/>.
- [57] Luis del Valle. Nodemcu. url <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>.
- [58] Prometec. Especificaciones del esp32. url <https://www.prometec.net/instalando-esp32/>.
- [59] Ryan Gross. Humanoid robotic hand. url <https://www.myminifactory.com/object/3d-print-humanoid-robotic-hand-34508>.
- [60] Fernando Herrera Arroyo. Desarrollo de una mano mecatrónica de bajo costo para un robot humanoide. 2017. doi: <http://estadisticas.repositorioinstitucional.buap.mx/viewer/index.php?code=710617T>.
-