



UNIVERSIDAD SANTO TOMÁS  
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

## **Herramienta para la generación de texto basada en una interfaz cerebro-computador**

Andrés Felipe Reyes Fernández

Universidad Santo Tomás  
Facultad de Ingeniería Electrónica  
Programa de Ingeniería Electrónica

Bogotá D.C

2020

# **Herramienta para la generación de texto basada en una interfaz cerebro-computador**

Andrés Felipe Reyes Fernández

Trabajo de grado presentado como requisito para optar al título de:

**Pregrado en Ingeniería Electrónica**

Director:

Edgar Camilo Camacho Poveda

Codirector:

Armando Mateus Rojas

Universidad Santo Tomás

Facultad de Ingeniería Electrónica

Programa de Ingeniería Electrónica

Bogotá D.C

2020

## **DECLARACIÓN DE AUTORÍA**

Declaro que este trabajo de grado es absolutamente original, auténtico, personal, que se han citado todas las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

## **DEDICATORIA**

A Dios por ser mi mayor inspiración.

A mis padres porque todos mis logros son logros de ellos.

## **AGRADECIMIENTOS**

A Dios por brindarme la sabiduría necesaria para solucionar los distintos retos presentados a lo largo del proyecto.

A mis directores del proyecto, Camilo Camacho y Armando Mateus por todo su tiempo, orientación, apoyo y consejos, que permitieron la realización exitosa de este proyecto.

A mis padres (Jorge Reyes y Carolina Fernández) por colaborar con el registro de las señales EEG, con las pruebas de la aplicación móvil, y por todo el apoyo emocional que me brindaron.

A toda mi familia por ser una gran fuente de apoyo y motivación.

A mi novia (Carolina Gaitán) por colaborar con el diseño gráfico de la aplicación móvil, y por todo su apoyo emocional.

A todas las personas que me colaboraron para tomar registro de las señales EEG, y para hacer las pruebas de la aplicación móvil.

## RESUMEN

En este trabajo se presenta el desarrollo de una herramienta que permite a las personas comunicarse, haciendo uso únicamente de sus parpadeos voluntarios. Esta herramienta brinda un medio de comunicación principalmente a las personas que tienen alguna discapacidad motora para comunicarse de forma verbal o escrita.

Para resolver el problema de la detección de los parpadeos voluntarios, en el presente trabajo se tomó como referencia el electroencefalograma (EEG), que en este caso fue registrado por el dispositivo *Mindwave Mobile 2* de la empresa *Neurosky*, el cual cuenta con un canal de medición de EEG, que se ubica en la frente de la persona.

Para el procesamiento digital del electroencefalograma (EEG) capturado por el dispositivo mencionado, se implementó una red neuronal artificial recurrente (RNN) del tipo *Long-Short Term Memory* (LSTM), ya que este tipo de redes son efectivas para el tratamiento de series de tiempo, como por ejemplo las señales electroencefalográficas (EEG).

La red neuronal implementada en este trabajo clasifica la señal EEG en una de cinco clases posibles que son, sin parpadeos, un parpadeo, dos parpadeos, tres parpadeos, o acción diferente. El modelo implementado entregó como resultado en su entrenamiento un porcentaje de exactitud promedio de 92%.

Finalmente, la red neuronal artificial se embebió en una aplicación móvil nativa de Android que se conecta vía bluetooth al dispositivo *Mindwave Mobile 2*, y que presenta un teclado virtual conformado por las 27 letras del abecedario de la lengua española, más los comandos “borrar”, “espacio”, y “enter”. Cada carácter del teclado puede ser seleccionado por el usuario únicamente mediante una serie determinada de parpadeos voluntarios. Cuando el usuario escribe una palabra y selecciona el comando “enter”, la palabra es presentada de forma audiovisual por la aplicación.

La aplicación móvil fue desarrollada en los lenguajes Java y XML en el entorno integrado de desarrollo (IDE) Android Studio. Para verificar su funcionamiento, se realizó un experimento con ocho personas, que entregó como resultado una efectividad en la selección correcta de letras del 91,26% en promedio. Por otra parte, el modelo de la red neuronal fue diseñado e implementado con el lenguaje Python, mediante el uso de las librerías TensorFlow y Keras (librerías para aprendizaje de máquina), y su entrenamiento se llevó a cabo en el entorno de desarrollo Google Colab.

## ABSTRACT

The purpose of the present work is about the development of a tool that allows people to communicate, only through their voluntary blinks. This tool provides a communication link mainly for people with motor disabilities, who cannot communicate through voice or text.

This work takes the electroencephalogram (EEG) as the source of information to solve the problem of detecting the voluntary blinks. In this case, the EEG is recorded by the Mindwave Mobile 2 headset (from Neurosky company), which counts on one EEG channel, located in the frontal lobe of the scalp.

In order to perform the digital processing of the EEG signal, a recurrent neural network (RNN) was implemented, more specifically a *Long-Short Term Memory* (LSTM), as these types of networks are effective for time series applications, for instance, EEG signals.

The neural network implemented in this work, classifies the EEG signal in one of 5 possible classes, named: No blink, One blink, Two blinks, Three blinks, Other. The results of the trained model were an average accuracy percentage of 92%.

Finally, the neural network was embedded in a native Android mobile application, that connects via Bluetooth to the Mindwave Mobile 2, and shows a virtual keyboard consisting of the 27 letters of the spanish alphabet, plus three characters are the commands “delete”, “space”, and “enter”. Each character can be selected by the user only through a determined number of voluntary blinks executed at certain times. When the user types a word and selects the “enter” command, the word is presented audio visually by the application.

The mobile application was developed in Java and XML languages in the Android Studio IDE (integrated development environment). In order to verify its performance, an experiment with eight people was executed, that achieved an average spelling precision of 91,26%. On the other hand, the neural network model was designed and implemented in Python language using the TensorFlow and Keras libraries (machine learning libraries), and it was trained in the Google Colab software development environment.

## ÍNDICE GENERAL

|                                        |    |
|----------------------------------------|----|
| 1. Introducción                        | 7  |
| 2. Planteamiento del problema          | 8  |
| 3. Justificación                       | 9  |
| 4. Impacto social                      | 10 |
| 5. Objetivos                           | 11 |
| 5.1. Objetivo general                  | 11 |
| 5.2. Objetivos específicos             | 11 |
| 6. Marco teórico                       | 12 |
| 7. Antecedentes                        | 19 |
| 8. Procedimiento Metodológico          | 22 |
| 9. Modelo de predicción basado en LSTM | 25 |
| 10. Aplicación móvil                   | 27 |
| 11. Resultados                         | 34 |
| 12. Conclusiones                       | 43 |
| 13. Trabajos futuros                   | 44 |
| 14. Referencias                        | 45 |



## 1. INTRODUCCIÓN

La comunicación es una herramienta fundamental para el ser humano, que ha sido clave en el desarrollo de la sociedad desde largo tiempo atrás. La forma de comunicación más fácil y rápida empleada por las personas es el lenguaje oral [1]. Sin embargo, por diversas circunstancias, algunas personas se encuentran limitadas para emplear esta forma de comunicación. Lo cual ha impulsado el desarrollo de nuevas tecnologías que brindan medios de comunicación alternativos, como lo es el caso de las interfaces cerebro-computador, que según [2], son sistemas que proveen un enlace de comunicación directa entre el cerebro y un computador, o algún otro dispositivo externo.

La investigación en las interfaces cerebro-computador comenzó alrededor de 1970 en la Universidad de California Los Ángeles (UCLA), bajo una subvención de la Fundación Nacional de Ciencia (NSF) [3]. Por otra parte, en años recientes, distintos programas de investigación en interfaces cerebro-computador han surgido debido a una mayor comprensión de las funciones del cerebro y al poderoso equipo computacional de bajo costo que se ha desarrollado [4].

Una fuente de información ampliamente usada por las interfaces cerebro-computador, para registrar la actividad cerebral, son las señales de electroencefalografía (EEG), ya que este método es no invasivo [5], lo cual implica que no hay necesidad de romper la piel o penetrar físicamente el cuerpo de la persona. Además, estas señales son las más adecuadas para las aplicaciones en tiempo real [6].

Distintas aplicaciones de interfaces cerebro-computador se basan en el parpadeo de la persona, que se caracteriza por tener dos picos en la señal EEG, un pico positivo que se produce con el evento de cerrar los ojos, y un pico negativo que se produce con el evento de abrir los ojos [7]. Los parpadeos se pueden clasificar en tres categorías: el parpadeo espontáneo, que según [8], se produce en promedio 14 veces por minuto, con la finalidad de mantener la humedad en la córnea. El parpadeo reflexivo, que se produce cuando un estímulo externo aparece de repente cerca al ojo. Y el parpadeo voluntario que se produce cuando la persona lo desea. Este tipo de parpadeo tiende a tener la mayor duración, que oscila generalmente entre 400 y 500 milisegundos [9].

En el presente documento se describe primero, la problemática que se va a solucionar mediante este proyecto, y la importancia de ejecutarlo. Luego, se presenta su impacto social y sus objetivos a cumplir. Después, se presentan los diseños y la implementación de la red neuronal LSTM propuesta, y de la aplicación móvil con el teclado virtual. Finalmente, se exponen los resultados obtenidos con el experimento realizado y las conclusiones.

## 2. PLANTEAMIENTO DEL PROBLEMA

Actualmente, un elevado número de personas en el mundo se encuentran limitadas por varias discapacidades motoras causadas por accidentes o enfermedades [10]. Por tal motivo, estas personas no tienen la posibilidad de interactuar con su entorno con la misma facilidad que lo hacen las personas que no se encuentran en esa condición. Esto se evidencia en la población colombiana que sufre en la actualidad una gran variedad de discapacidades, que la ponen en una situación crítica por causa de la vulnerabilidad presentada debido al limitado acceso a los servicios de salud y oportunidades laborales [11].

Por otra parte, un caso puntual de discapacidad es causada por la esclerosis lateral amiotrófica (ELA), que consiste en una enfermedad del sistema nervioso caracterizada por la degeneración de las neuronas motoras del cerebro y la médula espinal, reduciendo así la funcionalidad de los músculos. Esta enfermedad afecta entre seis y 26 personas por 1'000.000 de habitantes en el mundo [12].

Sin embargo, la necesidad de comunicación de las personas que presentan debilidad o disfuncionalidad en los músculos relacionados con el habla y con las manos se ve afectado en gran medida, ya que sus discapacidades les generan dificultades en la articulación de palabras, lentitud al hablar y pérdida de la movilidad de las manos, lo cual les impide escribir o digitar en un teclado, debido a la discapacidad motora.

A partir del problema previamente planteado, se plantea la siguiente pregunta de Investigación:

*¿Cómo seleccionar palabras deseadas por el usuario a partir de parpadeos voluntarios detectados de una señal electroencefalográfica (EEG), utilizando una herramienta de software?*

### 3. JUSTIFICACIÓN

Las consecuencias más comunes relacionadas con las discapacidades motoras consisten en la pérdida de control parcial o total de los músculos, que dificultan o imposibilitan el uso de habilidades fundamentales como el habla. La población colombiana no es ajena a esta situación, ya que, se ve afectada por la presencia de diversos casos de discapacidades motoras, las cuales generan elevados costos en el tratamiento para recuperar la movilidad [11]. Por estas razones se hace necesario la existencia de herramientas que faciliten la interacción de los discapacitados con el entorno y la sociedad, para lograr una mejoría en su calidad de vida.

Los sistemas de comunicación para personas que presentan dificultades relacionadas con su capacidad de expresarse verbal y no verbalmente, les brinda la oportunidad de satisfacer nuevamente la necesidad de comunicarse, la cual es fundamental porque permite expresar emociones, pensamientos, y afecto. En pocas palabras, se trata de una necesidad que impulsa el desarrollo personal, social y afectivo.

En la actualidad, los precios de los dispositivos que ofrecen asistencia a personas con discapacidad motora, oscilan entre USD \$1.000 y USD \$16.000 [13]. Por tal motivo, el acceso a estos dispositivos se encuentra limitado para las personas que cuentan con elevados recursos, ya que, los sistemas de seguridad social normalmente no cubren estos gastos. Por lo tanto, la herramienta que se propone en este proyecto, será de bajo costo, pues se estima que tendrá un precio de venta menor a USD \$500.

#### 4. IMPACTO SOCIAL

Según la Organización Mundial de la Salud, el 15% de la población mundial se encuentra afectada por una discapacidad física, psíquica o sensorial [14], porcentaje que equivale aproximadamente a 900 millones de personas y evidencia la necesidad latente a nivel mundial de investigar y desarrollar alternativas que garanticen igualdad de oportunidades. Esta necesidad ha sido reconocida y resaltada por las Naciones Unidas en la definición de los Ocho Objetivos de Desarrollo del Milenio, en especial a través de las metas 1a, 2a y 2b [15] que promueven la universalidad de acceso a la educación y trabajo entre otras.

Soportado por el Documento Marco de Proyección Social de la Universidad Santo Tomás [16] en donde se contextualiza que a través del desarrollo comunitario se apuesta por la preservación de la vida en todas sus multiplicidades, para satisfacer la necesidad anteriormente expuesta se están desarrollando nuevas tecnologías que favorecen la interacción de la comunidad discapacitada con su entorno. Casos específicos de estas tecnologías se encuentran en los trabajos [17], [18], [19], [20], [21], [22], [23] y [24].

El producto desarrollado con la ejecución de este proyecto, se encuentra basado en una interfaz cerebro-computador, que presenta un teclado virtual mediante el cual la persona puede comunicarse haciendo uso únicamente de sus parpadeos voluntarios. Este producto constituye una primera fase en la cual se obtiene un prototipo funcional que satisface la necesidad anteriormente expuesta y ofrece un beneficio principalmente a la comunidad con discapacidad motora. Con esto, en proyectos subsecuentes pueden desarrollarse soluciones económicamente viables que puedan ser eficazmente empleado por personas que se encuentran inhabilitadas para hacer uso del habla o para escribir, ya que, esta aplicación les permite volver a tener la posibilidad de comunicarse de forma oral y escrita.

## **5. OJETIVOS**

### **5.1. Objetivo general**

Diseñar e implementar una herramienta software que presente un teclado virtual y construya las palabras deseadas por el usuario a partir de sus parpadeos voluntarios detectados mediante una señal electroencefalográfica (EEG) registrada por el dispositivo *Neurosky Mindwave Mobile 2*.

### **5.2. Objetivos específicos**

- Realizar una revisión del estado del arte acerca de las distintas formas y metodologías empleadas para la detección de parpadeos voluntarios mediante un canal de las señales electroencefalográficas.
- Implementar los algoritmos que permitan procesar digitalmente la señal EEG adquirida por el dispositivo *Neurosky Mindwave Mobile 2*, con el fin de detectar los parpadeos voluntarios registrados en un lapso, para la selección de las letras deseadas por el usuario.
- Implementar un software que presente de forma audiovisual las palabras construidas por la herramienta, a través de un dispositivo móvil.
- Evaluar el funcionamiento de la herramienta software comparando cuantitativamente el número de letras seleccionadas correctamente por la herramienta, con las letras deseadas por distintos usuarios que hagan uso de la aplicación.

## 6. MARCO TEÓRICO

### 6.1. Electroencefalograma (EEG)

Según [25], el electroencefalograma es un método no invasivo ampliamente usado para monitorear el cerebro, basado en la medición de la actividad eléctrica en el cerebro a través del uso de electrodos metálicos ubicados en el cuero cabelludo, en donde registran pequeños potenciales eléctricos, los cuales son producidos por la suma de los impulsos eléctricos emitidos por grandes grupos de neuronas y células gliales operando en sincronía para comunicarse entre sí. Por otra parte, el primer electroencefalograma en la historia fue realizado en 1929 por el psiquiatra alemán vez Hans Berger.

### 6.2. Interfaces cerebro-computador

Son sistemas de comunicación conformados por hardware y software, que permiten el control de computadores y dispositivos externos a partir de señales relacionadas con la actividad cerebral [26]. Estas señales son registradas por un dispositivo que posteriormente las envía a una unidad de cómputo, en donde son procesadas y traducidas a comandos para ejecutar una determinada acción de tipo visual, auditiva o audiovisual. Para registrar la actividad cerebral, el dispositivo debe hacer uso de algún método o técnica tal como, el electroencefalograma, la espectroscopia de infrarrojo cercano, las imágenes por resonancia magnética funcional, entre otras.

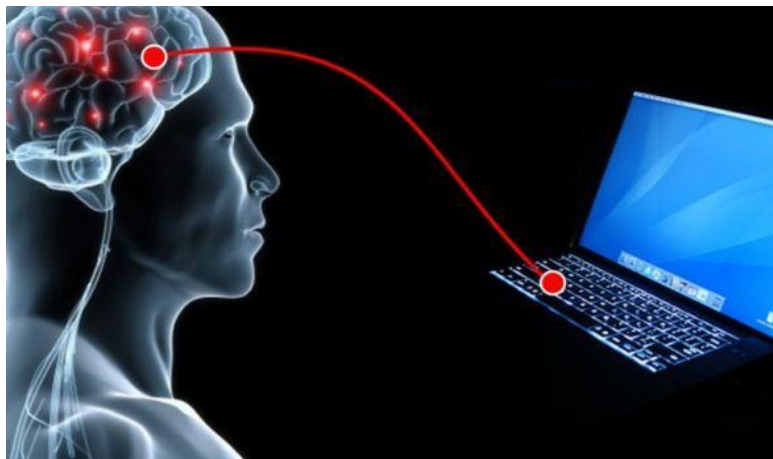


Figura 1. Interfaz cerebro-computador.

Fuente: <https://www.100cia.site/index.php/programacion-y-computacion/item/16787-que-es-una-interfaz-cerebro-computadora>

### 6.3. Aprendizaje de máquina

Se trata de una técnica que involucra el desarrollo de algoritmos que automáticamente mejoran con la práctica, permitiéndole a los sistemas computacionales “aprender” a partir de cierta información de entrada que se les presente [27]. En este contexto, aprender se define como el proceso de convertir la experiencia en experticia o conocimiento. En este proceso los algoritmos de aprendizaje reciben como entrada datos de entrenamiento, los cuales representan la experiencia. Y la salida arrojada por el algoritmo es la experticia. Uno de los objetivos principales del aprendizaje de máquina es lograr la detección automatizada de ciertos patrones significativos en distintos grupos de datos.

#### 6.3.1. Redes neuronales artificiales (ANN)

Son un modelo matemático inspirado en la arquitectura y el funcionamiento de las neuronas de los seres vivos, con el propósito de clasificar un grupo de datos de entrada. Este modelo consiste en una red que está compuesta por nodos (neuronas) interconectados que sirven como unidades de procesamiento de información [28]. Estos nodos se comportan de acuerdo con una función de activación que está determinada por una ecuación.

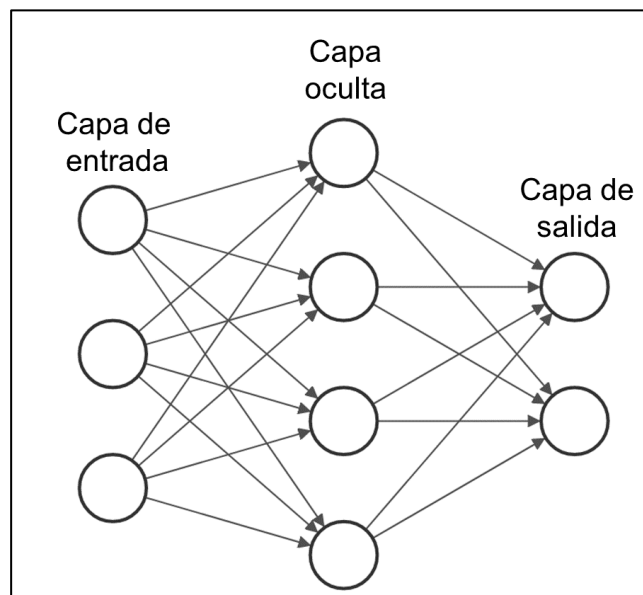


Figura 2. Red neuronal artificial

Las redes neuronales se componen por tres capas principalmente (Figura 2), que son la capa de entrada, en la cual se encuentran los datos que se van a procesar, la capa oculta o capas ocultas, en las cuales se encuentran los algoritmos de procesamiento, y finalmente la capa de salida, en donde se encuentra las clases posibles en las cuales se puede clasificar el grupo de datos de la capa de entrada.

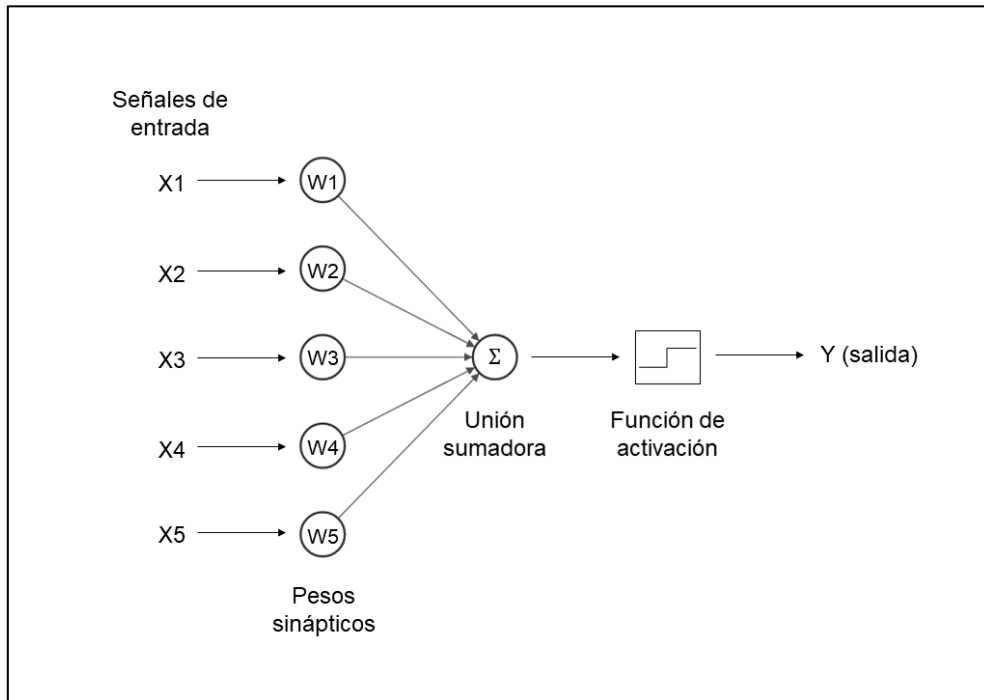


Figura 3. Esquema de un perceptrón o neurona artificial

El modelo matemático de una neurona artificial está determinado por la ecuación (1).

$$X * W^T = (x_1, x_2, \dots, x_n) * (w_1, w_2, \dots, w_n)^T = \sum_{i=1}^n x_i * w_i \quad (1)$$

Como se observa en la Figura 3, la neurona recibe cierta cantidad de datos o señales de entrada, los cuales en la ecuación (1) se representan con el vector X. Cada entrada se encuentra asociada a un peso o parámetro, y estos pesos se representan con el vector W. Por lo tanto, la función de ponderación de cada neurona se determina con la sumatoria de la multiplicación de todas las entradas con todos sus respectivos pesos.

Al resultado de la función de ponderación de cada neurona se le aplica una función de activación, tal como se muestra en la ecuación (2) que arrojará como resultado el valor de salida (Y) de la neurona. Dicho valor puede ser la nueva entrada de una neurona, formando así las redes neuronales, o puede ser el resultado final.

$$f\left(\sum_{i=1}^n x_i * w_i\right) \quad (2)$$

Existen diferentes funciones de activación. Entre las más utilizadas se encuentran:



- Función sigmoideal:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

- Función Tangente Hiperbólica:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (4)$$

### 6.3.2. Redes neuronales recurrentes (RNN)

De acuerdo con [29], las redes neuronales recurrentes son sistemas dinámicos que tienen un estado interno en cada paso de tiempo de la clasificación, y pueden ser utilizados, en distintos modelos y aplicaciones de procesamiento temporal de datos. Además, estos sistemas al ser modelados por ecuaciones diferenciales ordinarias, se pueden implementar digitalmente usando software estándar para la integración de las ecuaciones.

Por otra parte, las redes neuronales recurrentes pueden ser clasificadas como globalmente recurrentes, cuando las conexiones de retroalimentación entre todas las neuronas están permitidas, y se clasifican como localmente recurrentes cuando las neuronas de la red tienen conexiones de retroalimentación local.

En general, las redes globalmente recurrentes presentan problemas de estabilidad durante el entrenamiento, y requieren de complicados algoritmos de entrenamiento que consumen bastante tiempo. Mientras que las redes localmente recurrentes están diseñadas con modelos de neurona dinámicos que contienen lazos de retroalimentación internos, pero las interconexiones entre las neuronas son estrictas.

La ecuación (5) representa la fórmula básica que rige a las redes neuronales recurrentes.

$$h_t = g(h_{t-1}, x_t; \theta) \quad (5)$$

En la ecuación (5) la variable  $t$  representa un paso de tiempo discreto, y  $h_t$  es el estado interno actual de la neurona, que se define por una función en la cual se incluye el estado interno anterior  $h_{t-1}$ , la entrada actual a la neurona  $x_t$ , en donde  $\theta$  representa los parámetros o pesos de la neurona.

En la ecuación (6) se muestra el cálculo de la salida  $O_t$  de la RNN en el tiempo  $t$ , basado en la función que procesa el estado interno actual de la RNN  $h_t$ .

$$O_t = f(h_t; \theta) \quad (6)$$

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

Son un tipo especial de redes neuronales recurrentes basadas en el gradiente, capaces de aprender a disminuir los retrasos mínimos en más de 1000 pasos de tiempo discreto [30]. Estas redes usan carruseles de error constante (CEC), los cuales se encargan de imponer un flujo de error constante en las celdas especiales, con el fin de controlar el acceso a las celdas mediante compuertas multiplicativas, que aprenden cuando deben otorgar ese acceso. Las redes LSTM usan tres tipos de compuertas que son: la compuerta de entrada o actualización, la compuerta de olvido, y la compuerta de salida. Estas compuertas se pueden apreciar en la Figura 4.

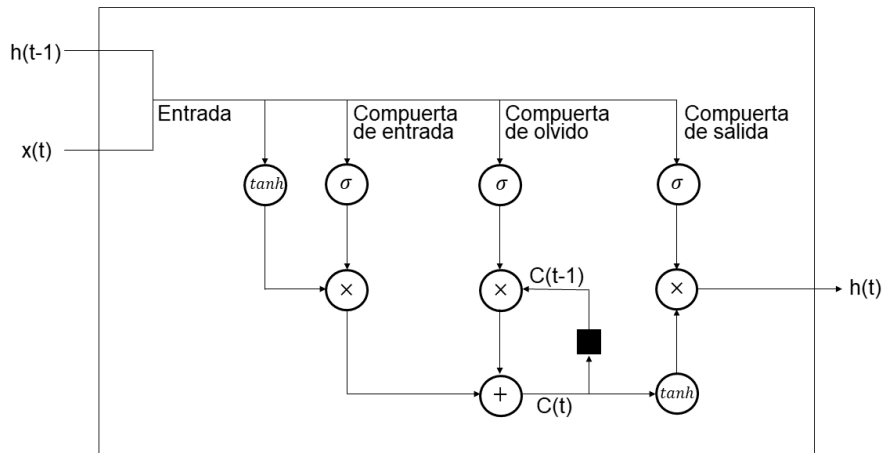


Figura 4. Estructura de una neurona LSTM. [30]

Además, las redes LSTM se encargan de solucionar el problema de desvanecimiento de gradiente. Este problema consiste en que el gradiente se va disminuyendo a valores muy pequeños, impidiendo que el peso pueda cambiar su valor en las siguientes iteraciones. Por otra parte, el modelo matemático que representa a las redes LSTM está determinado por las siguientes ecuaciones:

- Compuerta de entrada: Esta compuerta determina si se almacena nueva información en la memoria de la celda, a través de la ecuación (7):

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (7)$$

En la ecuación (7) se introduce un nuevo término  $b_i$ , conocido como sesgo (o bias en inglés), el cual es un valor de compensación que se le adiciona a cada neurona de una capa de la red neuronal, y este valor es independiente de las

entradas de la capa. Por otra parte, sigma  $\sigma$  hace referencia a la función sigmoideal.

En la ecuación (8) se introduce un nuevo término  $C_t$ , el cual es un vector con los valores almacenados en la celda, representando así la memoria de la neurona LSTM.

$$C_t = \tanh (W_c * [h_{t-1}, x_t] + b_c) \quad (8)$$

- Compuerta de olvido: Esta compuerta determina si se elimina información previamente almacenada en la memoria.

$$f_t = \sigma (W_f * [h_{t-1}, x_t] + b_f) \quad (9)$$

- Compuerta de salida: Esta compuerta determina qué información almacenada en la memoria se coloca en la salida, de acuerdo con la ecuación (10).

$$O_t = \sigma (W_o * [h_{t-1}, x_t] + b_o) \quad (10)$$

Finalmente, el cálculo del estado actual  $h_t$  de la celda LSTM, se realiza mediante la ecuación (11).

$$h_t = O_t * \tanh (C_t) \quad (11)$$

#### 6.4. Métricas para analizar los resultados del experimento

Dado que en el experimento cada persona debe escribir un cierto número de frases en el teclado virtual de la aplicación móvil, se definió un parámetro cuantitativo para medir los resultados del experimento. El cual consiste en un porcentaje de efectividad que se le asigna a cada frase escrita, y se obtiene con la ecuación (12):

$$E = \frac{P - F}{P} * 100 \quad (12)$$

*P: Número de predicciones*

*F: Número de fallas*

*E: Porcentaje de efectividad*

- **P:** Es la mínima cantidad total de predicciones que debe realizar la red neuronal para construir cierta frase, y se determina multiplicando el número de letras de la frase por 3, ya que, para escoger una sola letra, la aplicación debe realizar mínimo 3 selecciones. En cada selección debe identificar una de 4 posibles acciones del usuario (1 parpadeo, 2 parpadeos, 3 parpadeos, o nada).

Esto significa que el número de predicciones de la frase es igual al número mínimo de selecciones que debe realizar la aplicación para escribir la frase.

$$P = 3 * \text{Número de letras de la frase} \quad (13)$$

- **F:** Es el número de predicciones erróneas que hace la aplicación en el proceso de escoger todas las letras de la frase deseada por la persona.
- **E:** Es el porcentaje de efectividad de la aplicación móvil obtenido en la construcción de una frase.

## 7. ANTECEDENTES

De acuerdo con la investigación realizada sobre las distintas metodologías empleadas en trabajos relacionados, para la detección de parpadeos basados en señales EEG, se encontró que muchos enfoques, se basan en el uso de algoritmos de aprendizaje de máquina [31], [32], [33]. Por ejemplo, en el trabajo realizado por [34], se implementó una máquina de vectores de soporte (SVM), capaz de clasificar parpadeos voluntarios e involuntarios, a partir de una señal EEG registrada en el lóbulo frontal de la persona, con el uso del dispositivo *Emotiv*. Algunas de las características de la señal EEG tomadas en cuenta fueron: el valor máximo y mínimo, la varianza, y la curtosis. En los experimentos realizados por este trabajo se lograron exactitudes de clasificación promedio cercanas al 90%.

En el trabajo desarrollado por [35], se implementó un sistema de control para un dron, basado en parpadeos voluntarios, los cuales fueron registrados con un dispositivo EEG de la empresa *Cogionics Inc.* En este trabajo se implementó una red neuronal convolucional (CNN) para la clasificación de las señales EEG. Y como resultado de los experimentos realizados, se obtuvo un 91,27% de exactitud en el desempeño del sistema. Además, este trabajo propone en sus conclusiones la implementación de redes neuronales recurrentes (RNN) para mejorar el desempeño del sistema.

Otros enfoques han aplicado algoritmos que involucran distintos métodos algebraicos y técnicas de filtrado, basándose en características de tipo espacial, temporal y frecuencial de la señal EEG, para la detección de los parpadeos [36], [37], [38] y [39]. En el caso de [40]. Se implementa la transformada discreta de *Wavelet*, para descomponer la señal EEG en sus diferentes bandas de frecuencias, las cuales contienen la información que permite la identificación de los parpadeos. En este trabajo, la señal EEG fue registrada en el lóbulo frontal por el dispositivo *Mindwave* de *Neurosky*.

Por otra parte, en el trabajo realizado por [41], se implementó una red neuronal con una capa LSTM para la clasificación de las señales EEG asociadas a distintas imágenes mentales motoras (movimiento de una mano, movimiento del pie). las señales fueron capturadas por los 4 canales de medición del dispositivo *Muse* de la empresa *Interaxon*. En este trabajo la exactitud de clasificación registrada fue de 80%.

Finalmente, cabe mencionar que de acuerdo con [42], para las aplicaciones de interfaces cerebro-computador, las redes neuronales han sido ampliamente usadas en mayor proporción, que los otros tipos de algoritmos de aprendizaje de máquina, principalmente porque la mayoría de la investigación en estos

algoritmos se enfoca en datos estáticos, lo cual no es óptimo para categorizar con exactitud las señales cerebrales, que varían a gran velocidad.

Por otra parte, con respecto a los proyectos relacionados con teclados virtuales, se encuentran el trabajo realizado por dos universidades de Pakistán, sobre la implementación de un sistema deletreador de texto, basado en la clasificación del movimiento del ojo y en el parpadeo, usando las señales electroencefalográficas (EEG) registradas por un dispositivo llamado *Emotiv* [43]. Por otra parte, en un trabajo realizado por la Universidad Nacional *Pukyong* de Corea del Sur, se desarrolló una interfaz cerebro-computador para deletrear, utilizando señales electroencefalográficas relacionadas con tener los ojos cerrados y con el doble parpadeo [44].

Otros trabajos se han enfocado en el uso de cámaras para la detección de parpadeos, así sucede con el trabajo de investigación realizado por [45], en el cual se desarrolló un sistema de asistencia para personas con parálisis, basado en el monitoreo de los ojos y en el conteo de parpadeos mediante algoritmos de procesamiento de imágenes.

Debido al elevado costo de diversos sistemas para asistencia a personas con discapacidad motora, ciertos trabajos se han enfocado en utilizar dispositivos de bajo costo para el desarrollo de este tipo de sistemas, por ejemplo, en el caso de los trabajos realizados por [46], [47] y [48], en los que se implementó un sistema compuesto por un dispositivo para la medición de un electroencefalograma de la empresa *Neurosky*.

Un problema relevante que han presentado distintos trabajos relacionados con los sistemas de deletreo para discapacitados, es el largo tiempo requerido para la selección de cada carácter o letra. Esto se evidencia en el trabajo realizado por [49], en el cual se desarrolló un teclado virtual controlado únicamente por parpadeos, que son detectados mediante la señal EEG adquirida por el sistema *Biopac MP36*. La tasa de deletreo obtenida en los experimentos de este trabajo fue de un carácter por minuto, lo cual resulta lento para el usuario.

Por otra parte, el procedimiento común para evaluar los sistemas de deletreo para discapacitados, consiste en acudir a personas con y sin discapacidad motora de diferentes edades para que hagan uso del sistema con el fin de verificar su funcionamiento. Por ejemplo, para el teclado virtual desarrollado por [50], cuatro personas participaron en el proceso de validación del sistema, en el cual, a cada persona se le pidió que construyera palabras de distinta longitud (entre tres y ocho caracteres), con el propósito de medir el tiempo que a cada uno le tomaba construir una palabra, para poder determinar la tasa de deletreo. Por otra parte, en el trabajo realizado por [51]

se implementó un sistema de comunicación para discapacitados basado en EEG, y para verificar su precisión, se desarrolló un experimento en el cual participaron cinco personas. A cada una se le asignaron tres tareas diferentes, que consisten en digitar una frase de 10 caracteres.

## 8. PROCEDIMIENTO METODOLÓGICO

En esta sección se presenta la metodología que se llevó a cabo para la ejecución de este proyecto. Los pasos que se siguieron se ilustran en la Figura 5.

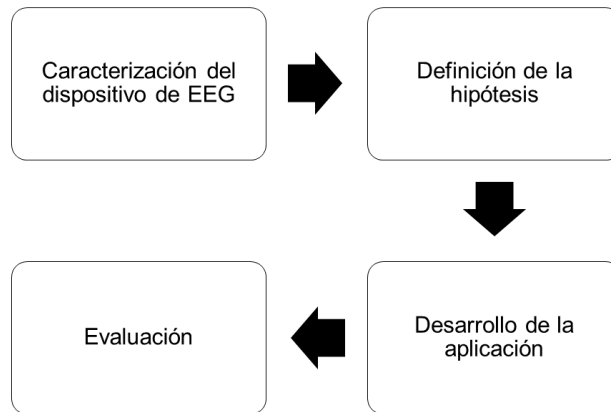


Figura 5. Esquema del procedimiento metodológico

### 8.1. Caracterización del dispositivo de EEG

Para la adquisición del electroencefalograma en este proyecto, se hizo uso del dispositivo Mindwave Mobile 2 de la empresa *Neurosky*. Que consiste en una diadema que cuenta con un electrodo seco de medición de EEG, el cual se ubica en el lóbulo frontal de la persona. Este dispositivo cuenta con un chip llamado *ThinkGear* (TGAM1), que muestrea la señal EEG a una frecuencia de 512 Hz, la procesa, y la transmite vía bluetooth [52].

El procesamiento en este dispositivo se realiza con un filtro *notch* (rechaza-banda) para eliminar el ruido proporcionado en la frecuencia de 60hz. Además, este chip cuenta con algoritmos que extraen los niveles de energía relacionados con cada una de las bandas de frecuencia de la señal EEG (*Delta*, *Theta*, *Alpha*, *Beta* y *Gamma*), y con base en esos niveles de energía, determina el nivel de concentración y de relajación de la persona en un rango de 0 a 100.

En este proyecto se realizó la conexión entre el dispositivo Mindwave Mobile 2 y un PC que cuenta con el sistema operativo Ubuntu 18. Para esto fue necesario descargar el programa *BlueMan*, mediante el cual se pudo emparejar el dispositivo vía bluetooth con el PC. Además, se instaló la librería de Python *NeuroPy*, con el fin de leer los datos transmitidos por el dispositivo.

También se estableció la conexión del dispositivo con un celular (Xiaomi) con sistema operativo Android. La conexión fue realizada en Java, en el entorno de



desarrollo Android Studio, con el uso del Kit de Desarrollo de Software (SDK) para Android, ofrecido por la empresa *Neurosky*.

## **8.2. Definición de la hipótesis**

Teniendo en cuenta que en el trabajo realizado por [35], se recomienda en sus conclusiones el uso de redes neuronales recurrentes para mejorar el desempeño del sistema en la detección de parpadeos a partir de señales EEG. Y que en el trabajo realizado por [41], se obtuvieron buenos resultados con el uso de una red LSTM para la clasificación de señales EEG asociadas a distintas imágenes motoras. Se decidió en este trabajo implementar una red neuronal recurrente de tipo LSTM para la detección y clasificación de los parpadeos.

Haciendo uso de la librería *NeuroPy* y *NumPy* se desarrolló un script en Python para la construcción del conjunto de datos usado en el entrenamiento de la red neuronal, el cual está compuesto por las señales Electroencefalográficas asociadas a la realización de distintas actividades por parte de 3 personas. El experimento para la toma del conjunto de datos consistió en que cada persona debía realizar las siguientes actividades repitiéndolas cierto número de veces cada una: 1) No hacer nada 2) Parpadear 1 vez, 3) Parpadear 2 veces, 4) Parpadear 3 veces, 5) alzar las cejas 1, 2 y 3 veces.

Cada actividad se realizó por un periodo de dos segundos, tiempo durante el cual se tomó registro de la señal EEG de la persona, logrando así obtener muestras de 1024 datos cada una. Esta ventana de tiempo (2 segundos) se estableció de acuerdo a la información encontrada en el estado del arte sobre la duración de un parpadeo voluntario, que es aproximadamente de 500 milisegundos. De forma que, para alcanzar a capturar tres parpadeos voluntarios se necesita un tiempo mínimo de 1500 milisegundos = 1,5 segundos, y teniendo en cuenta el tiempo de reacción de la persona, junto con los tiempos muertos entre un parpadeo y otro, se agregaron 500 milisegundos, obteniendo así los dos segundos de duración de la ventana.

Para el modelo implementado en la aplicación final, se registraron 1500 muestras por persona (300 muestras por clase), por lo cual el tamaño total del conjunto de datos fue de 4500 muestras, cada una con 1024 datos. De este conjunto de datos, el 80% fue destinado para entrenamiento, un 10% para validación y el 10% restante para testeo.

Después, se diseñaron distintos modelos de redes neuronales de tipo LSTM en el lenguaje Python, mediante el uso de las librerías *TensorFlow* y *Keras* (librerías para aprendizaje de máquina), y se implementaron en el entorno de desarrollo *Google Colab*. Estos modelos fueron comparados según su rendimiento basado en las curvas de aprendizaje, en los porcentajes de exactitud obtenidos con el entrenamiento de la red, y en las matrices de confusión obtenidas con los

resultados del conjunto de datos de testeo. Esto se hizo con el fin de elegir el mejor modelo, que fue embebido en la aplicación móvil final. Los modelos diseñados clasifican los datos en 5 categorías posibles: 0) cero parpadeos 1) un parpadeo 2) dos parpadeos 3) tres parpadeos y 4) otra acción.

### **8.3. Desarrollo de la aplicación**

En primer lugar, se realizó la estructura lógica de la aplicación móvil, específicamente del proceso de selección de cada carácter del teclado por parte del usuario. Con base en ese proceso, se diseñó gráficamente el teclado virtual que se presenta al usuario.

Después, se diseñó la interfaz gráfica de toda la aplicación, con el uso de la herramienta online Marvel App, a través de la cual se simuló la secuencia que debía seguir la aplicación móvil durante su ejecución

Luego, se implementó el *backend* de la aplicación móvil en Java. Comenzando por la adquisición de los datos del dispositivo EEG, seguido de la implementación del modelo de la red neuronal almacenada en un archivo con extensión protobuf (pb), y en las funciones para adecuar los datos de entrada al modelo y realizar la clasificación de éstos.

Finalmente, se implementó el *frontend* de la aplicación móvil en Java y XML, tomando como base la interfaz gráfica diseñada anteriormente en la herramienta online Marvel App

### **8.4. Evaluación**

Para evaluar el funcionamiento de la aplicación, se realizó un experimento, en el que participaron 8 personas de distintas edades, en un rango entre 20 y 75 años. El experimento consistió en que cada persona debía escribir 6 frases en el teclado virtual de la aplicación móvil. 3 de las frases fueron iguales para todos los participantes, y las otras 3 frases fueron de libre elección por cada persona, pero con la restricción de que cada una de las frases elegidas debía contener 12 letras. Las frases predeterminadas para todos fueron: 1) Quiero ir al baño (14 letras) 2) Tengo hambre (11 letras) 3) Me siento cansado (15 letras).

El experimento se realizó de esta forma, con el fin de comparar los resultados obtenidos cuando los usuarios escribieron frases predeterminadas con distinta cantidad de letras, a cuando escribieron frases escogidas por ellos con la misma cantidad de letras cada una.

## 9. MODELO DE PREDICCIÓN BASADO EN LSTM

En la Figura 6 se muestra la estructura de la red neuronal implementada. En esta red, la capa *dropout* cumple la función de desactivar un porcentaje de la cantidad total de neuronas en la capa anterior. Esto con el fin de evitar el problema de sobreentrenar la red [53].

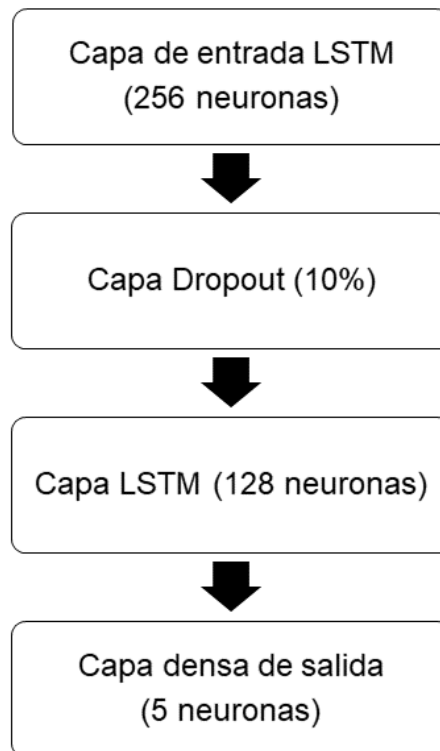


Figura 6. Estructura de la red neuronal LSTM

En la Tabla 1 se muestran los parámetros del modelo de la red neuronal implementada.

Este modelo fue entrenado con 90 épocas y un tamaño de lote de 64 muestras. Se utilizó el optimizador Adam en el proceso de entrenamiento, con una tasa de aprendizaje de 0,001, y el porcentaje de neuronas desactivadas por la capa *dropout* fue de 10%. Por otra parte, Las capas de tipo LSTM implementadas usan como función de activación, la función de la tangente hiperbólica (*tanh*). También cabe resaltar que las capas LSTM y la capa *dropout* se utilizaron para aprender las características de la señal EEG, y la capa de salida densa es utilizada para la clasificación.

| <b>Parámetros del modelo LSTM</b> |              |
|-----------------------------------|--------------|
| <b>Capa 1 LSTM</b>                | 256 neuronas |
| <b>Capa dropout</b>               | 10%          |
| <b>Capa 2 LSTM</b>                | 128 neuronas |
| <b>Capa de salida</b>             | 5 neuronas   |
| <b>Pasos de tiempo</b>            | 1024         |
| <b>Dimensión de entrada</b>       | 1            |
| <b>Muestras</b>                   | 4050         |
| <b>Tamaño de lote</b>             | 64           |
| <b>Épocas</b>                     | 90           |
| <b>Tasa de aprendizaje</b>        | 0,001        |

Tabla 1. Parámetros de la red neuronal LSTM

## 10. APLICACIÓN MÓVIL

En la Figura 7 se presenta el diagrama de clases de la aplicación móvil desarrollada en Java/XML.

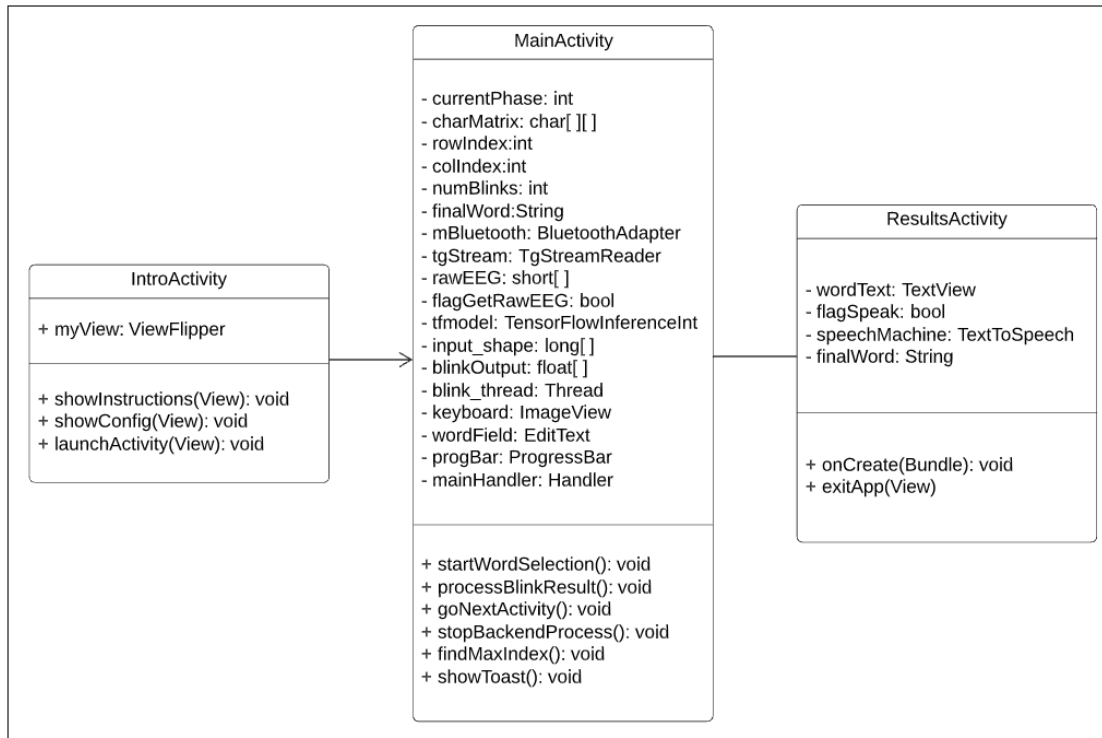


Figura 7. Diagrama de clases de la aplicación móvil

En la Figura 8 se muestra el diagrama de flujo correspondiente al funcionamiento de la aplicación móvil.

La aplicación móvil implementada en este trabajo cuenta con tres secciones: Inicio, Teclado virtual, y Final.

- Inicio

En esta sección se muestra primero la portada de la aplicación, que se puede apreciar en la Figura 9.

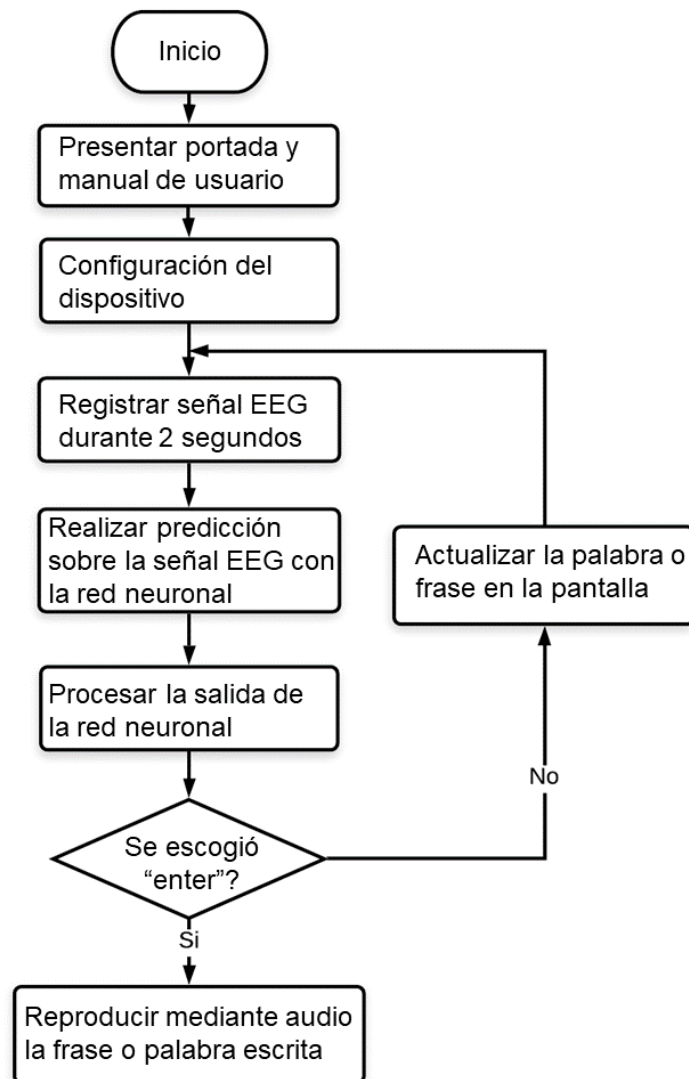


Figura 8. Diagrama de flujo de la aplicación móvil

Luego, al presionar el botón “COMENZAR” se presenta el manual de usuario, en el cual se indica cómo funciona el teclado virtual, y se muestra en la Figura 10.

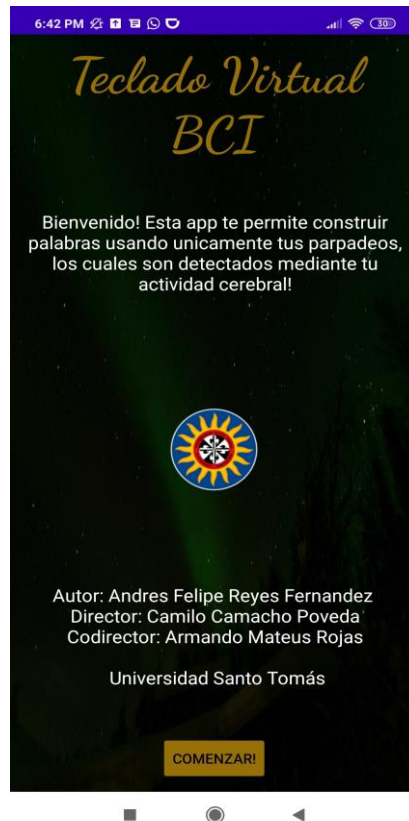


Figura 9. Portada de la aplicación móvil

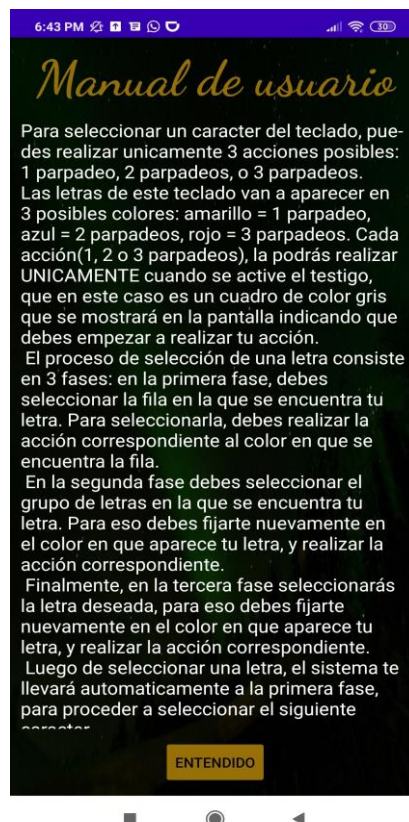


Figura 10. Manual de usuario de la aplicación móvil

Después, al presionar el botón “ENTENDIDO” se presentan las instrucciones para configurar la diadema Mindwave Mobile 2 correctamente para la medición del electroencefalograma (EEG). Esto se muestra en la Figura 11.



Figura 11. Configuración del dispositivo

- Teclado virtual

En este teclado las letras aparecen en tres colores distintos, tal y como se muestra en la Figura 12. Para seleccionar un carácter del teclado, se pueden realizar únicamente tres acciones posibles: un parpadeo, dos parpadeos, o tres parpadeos. Cada acción se encuentra asociada a un color. En el caso de los caracteres “Borrar”, “[ ]”, y “Enter”, la persona debe esperar un momento sin hacer nada hasta que estos tres caracteres cambien de color como se muestra en la Figura 13, para poder seleccionar alguno de ellos con la acción correspondiente.

Cada acción (uno, dos o tres parpadeos), se podrá realizar únicamente cuando se active el testigo, que como se presenta en la Figura 13, es un cuadro de color gris que aparece en la pantalla indicando que el usuario debe empezar a realizar su acción.



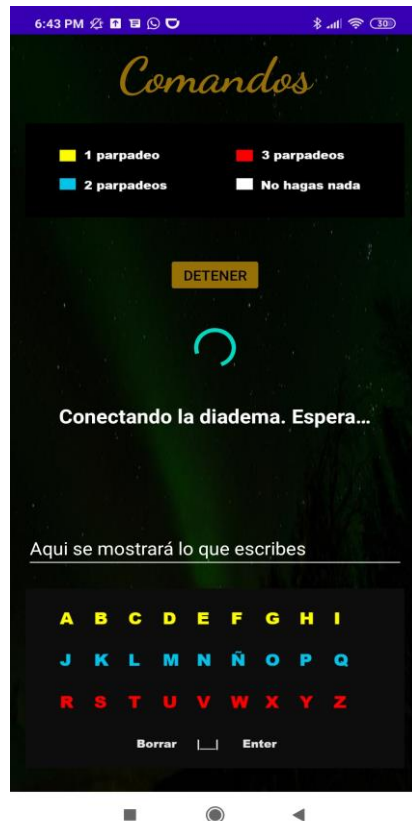


Figura 12. Presentación del teclado virtual



Figura 13. Testigo del teclado virtual

El proceso de selección de una letra en este teclado, fue invención del autor de este trabajo, y consiste en tres fases. La primera fase se muestra la Figura 14, y consiste en escoger la fila de la letra deseada.

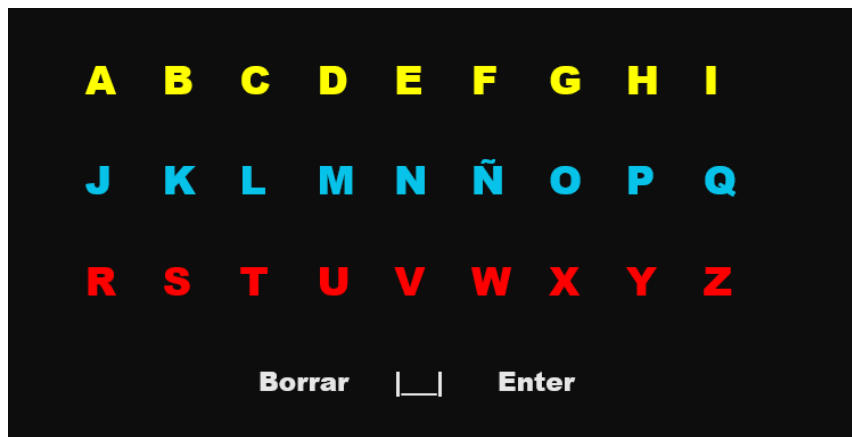


Figura 14. Fase 1 del teclado virtual

En la segunda fase se muestra iluminada únicamente la fila escogida en la primera fase, y el resto de los caracteres se desactivan. Esta fase consiste en escoger el grupo de letras (tres letras) en el que se encuentra la letra deseada. En la Figura 15 se muestra el teclado en la segunda fase, asumiendo que se escogió la primera fila.



Figura 15. Fase 2 del teclado virtual

En la tercera fase se muestra iluminado únicamente el grupo de tres letras escogido en la segunda fase, y el resto de los caracteres se desactivan. Esta fase consiste finalmente en escoger la letra deseada. En la Figura 16 se muestra el teclado en la tercera fase, asumiendo que se escogió la primera fila y el grupo amarillo en la segunda fase.

Cuando la persona realiza la selección en la tercera fase, la aplicación muestra la letra seleccionada por la persona y automáticamente vuelve a la primera fase para comenzar el proceso de selección del siguiente carácter.

- Final

Finalmente, en esta sección se reproduce de manera audible la palabra o la frase escrita por el usuario, utilizando la librería *TextToSpeech*, desarrollada por Android. Además, esta frase/palabra se presenta en el centro de la pantalla.

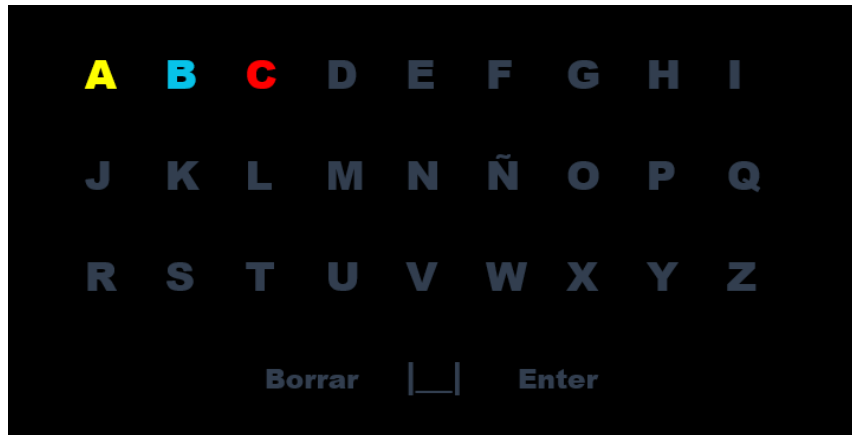


Figura 16. Fase 3 del teclado virtual

## 11. RESULTADOS

### 11.1. Red neuronal LSTM:

En la Tabla 2 se muestran los 5 modelos diseñados que presentaron los mejores resultados en el entrenamiento, en los cuales se presenta el valor de exactitud y de pérdida registrados en el set de validación.

| Modelo             | Capas LSTM                | Exactitud | Pérdida |
|--------------------|---------------------------|-----------|---------|
| 1                  | 2(128 y 64 neuronas)      | 0,88      | 0,37    |
| 2                  | 2(512 y 256 neuronas)     | 0,89      | 0,48    |
| 3                  | 3(256, 128 y 64 neuronas) | 0,84      | 0,55    |
| 4                  | 1(256 neuronas)           | 0,81      | 0,63    |
| 5(Modelo escogido) | 2(256 y 128 neuronas)     | 0,92      | 0,25    |

Tabla 2. Modelos LSTM

De acuerdo con los datos registrados en la Tabla 2 se puede observar que el modelo que presenta la mayor exactitud y la menor pérdida es el modelo 5. Estos datos fueron tomados en cuenta en la toma de decisión del modelo a implementar en la aplicación móvil.

Por otra parte, luego del entrenamiento de cada modelo, se graficaron sus curvas de aprendizaje resultantes, correspondientes a los valores de pérdidas y al porcentaje de exactitud del conjunto de datos de entrenamiento (curva verde) y del conjunto de datos de validación (curva amarilla). En el eje X se encuentra el número de épocas del entrenamiento, y en el eje Y se encuentran los valores de la pérdida registrados en el primer caso y los porcentajes de exactitud registrados en el segundo caso.

En las Figura 17 y Figura 18 se presentan y analizan las curvas de aprendizaje obtenidas con el entrenamiento del modelo escogido a continuación.

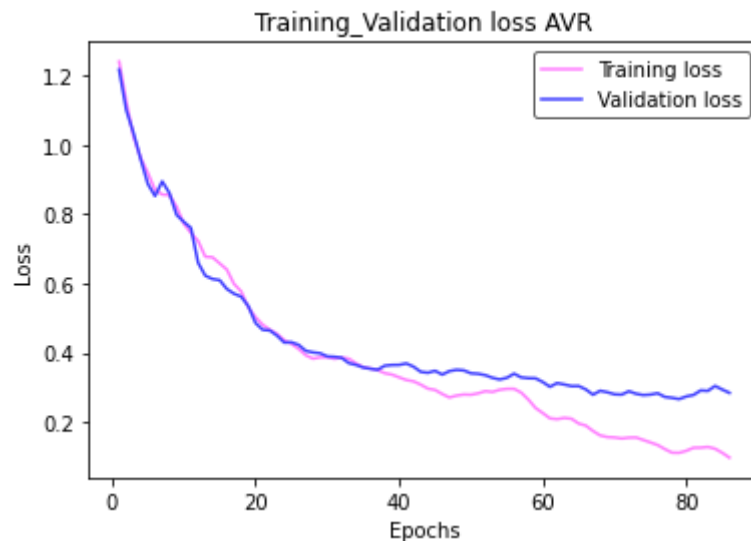


Figura 17. Curvas de aprendizaje de perdida

De acuerdo con las curvas presentadas en la Figura 17, se puede apreciar que la pérdida en el conjunto de datos de validación se logra minimizar desde 1.2 hasta 0.2 aproximadamente. También se observa que el valor de pérdida del conjunto de datos de validación se comienza a estabilizar luego de pasar 60 épocas. por otra parte, la brecha entre los valores de pérdida en el conjunto de datos de entrenamiento y de validación, comienza a incrementar significativamente a partir de 80 épocas, lo cual permitió definir el número de épocas necesario, para evitar el problema de sobre aprendizaje en la red.

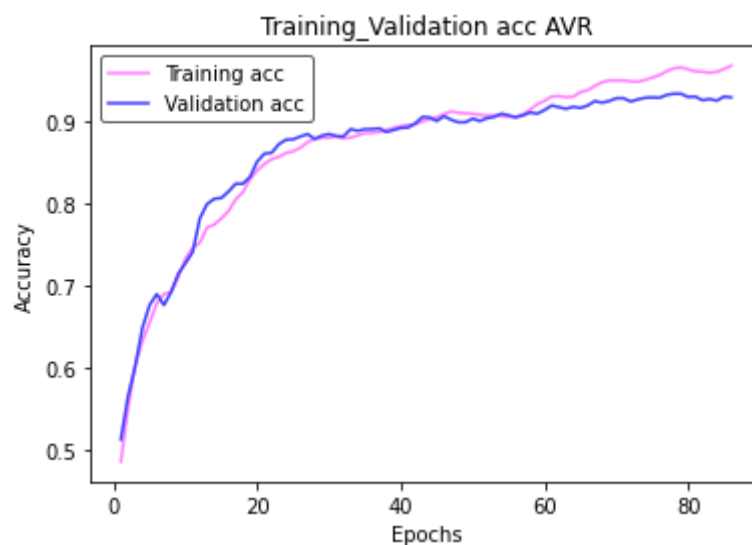


Figura 18. Curvas de aprendizaje de exactitud

De acuerdo con las curvas presentadas en la Figura 18, se puede apreciar que la exactitud en el conjunto de datos de validación alcanza un valor superior al 90% aproximadamente, y se comienza a estabilizar luego de pasar 30 épocas. Por otra parte, la brecha entre la exactitud registrada en el conjunto de datos de entrenamiento y de validación, comienza a incrementar significativamente a partir de 80 épocas, lo cual refleja indicios de sobre aprendizaje en la red. Esto sirvió como referencia para establecer el número de épocas adecuado para el entrenamiento.

Para analizar los resultados de las predicciones realizadas por estos modelos con el conjunto de datos de prueba, se graficaron las matrices de confusión. Ya que es una herramienta acertada para evaluar el rendimiento de modelos de clasificación con múltiples categorías. El conjunto de datos de prueba utilizado fue el mismo para todos los modelos y se encuentra compuesto por 450 muestras, distribuidas de la siguiente manera: 90 muestras de la categoría 0: cero parpadeos, 90 muestras de la categoría 1: un parpadeo, 90 muestras de la categoría 2: dos parpadeos, 90 muestras de la categoría 3: tres parpadeos, y 90 muestras de la categoría 4: otra acción.

También cabe resaltar que los valores de las matrices de confusión se encuentran normalizados entre 0 y 1. En el eje X, se muestra la cantidad predicciones realizadas por modelo en cada categoría, y en el eje Y, se muestra la cantidad de muestras que realmente pertenecen a cada categoría.

Además, se presentan las curvas de aprendizaje registradas en cada modelo.

## Modelo 1

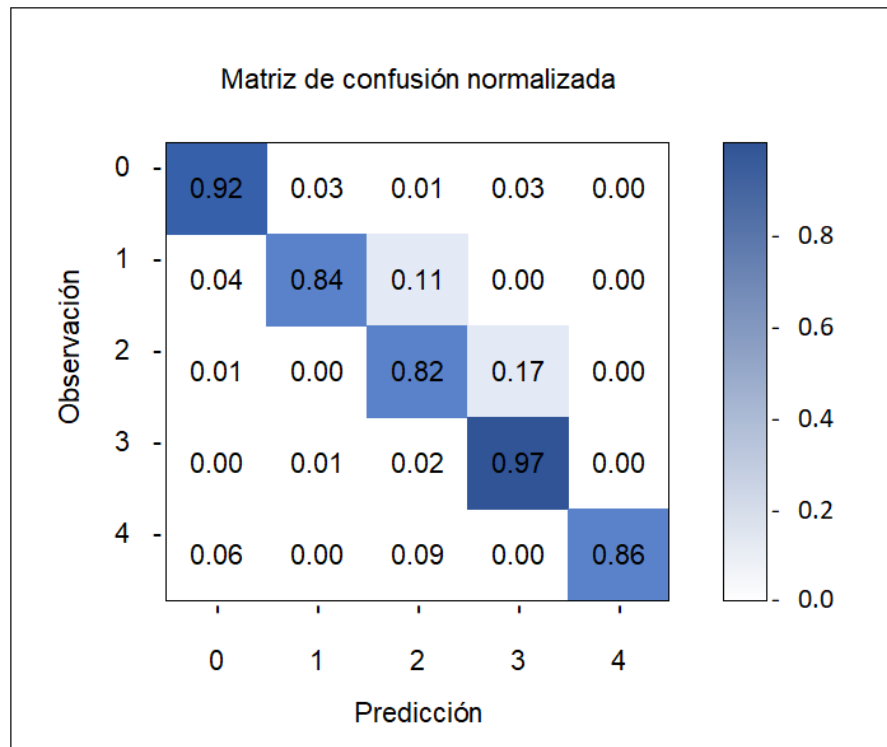


Figura 19. Matriz de confusión del modelo 1

En la Figura 19 se puede apreciar que el modelo presenta un excelente desempeño en la clasificación de las categorías 0 y 3, ya que muestran una precisión superior al 90% en la predicción. Sin embargo, el modelo presenta cierta tendencia a clasificar erróneamente muestras de la categoría 2 (2 parpadeos) en la categoría 3 (3 parpadeos).

## Modelo 2

En la Figura 20 se puede apreciar que el modelo presenta un excelente desempeño en la clasificación de las categorías 0 y 3, ya que muestran una precisión superior al 90% en la predicción. Sin embargo, no es tan preciso clasificando muestras pertenecientes a la categoría 1 (1 parpadeo), y presenta cierta tendencia a clasificar erróneamente muestras de la categoría 1 (1 parpadeos) en la categoría 0 (0 parpadeos).

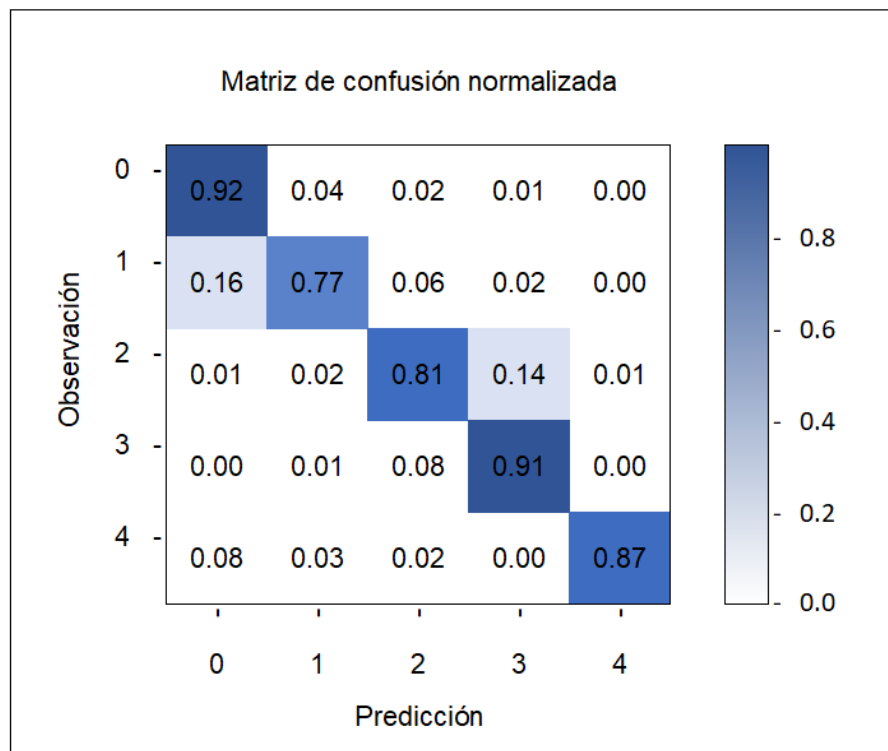


Figura 20. Matriz de confusión del modelo 2

### Modelo 3

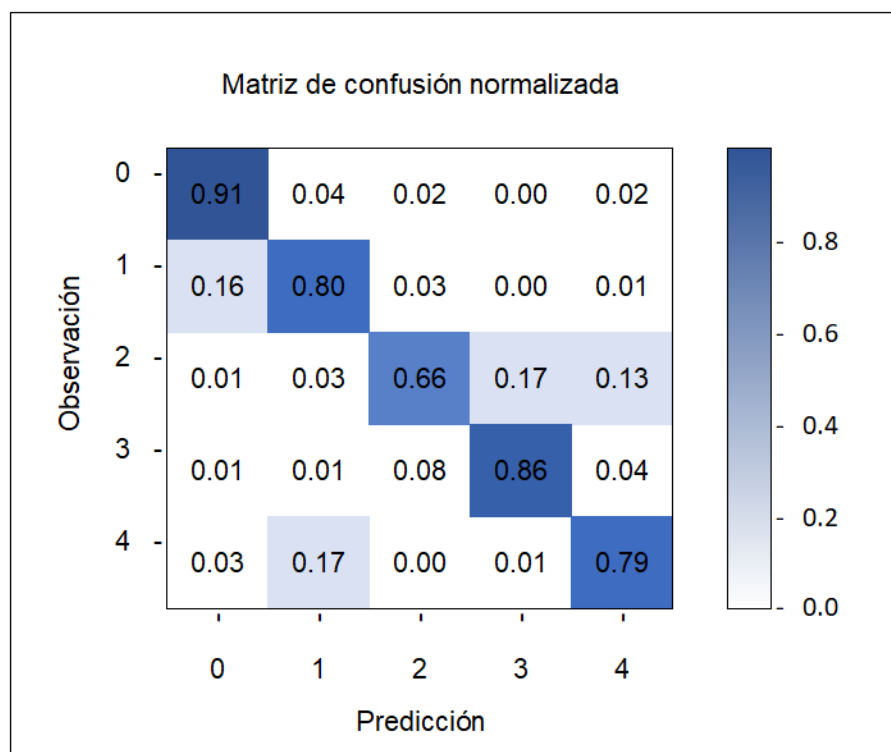


Figura 21. Matriz de confusión del modelo 3



En la Figura 21 se puede apreciar que el modelo presenta un desempeño deficiente en la clasificación de la categoría 2, ya que registra una precisión del 66% en la predicción.

#### Modelo 4

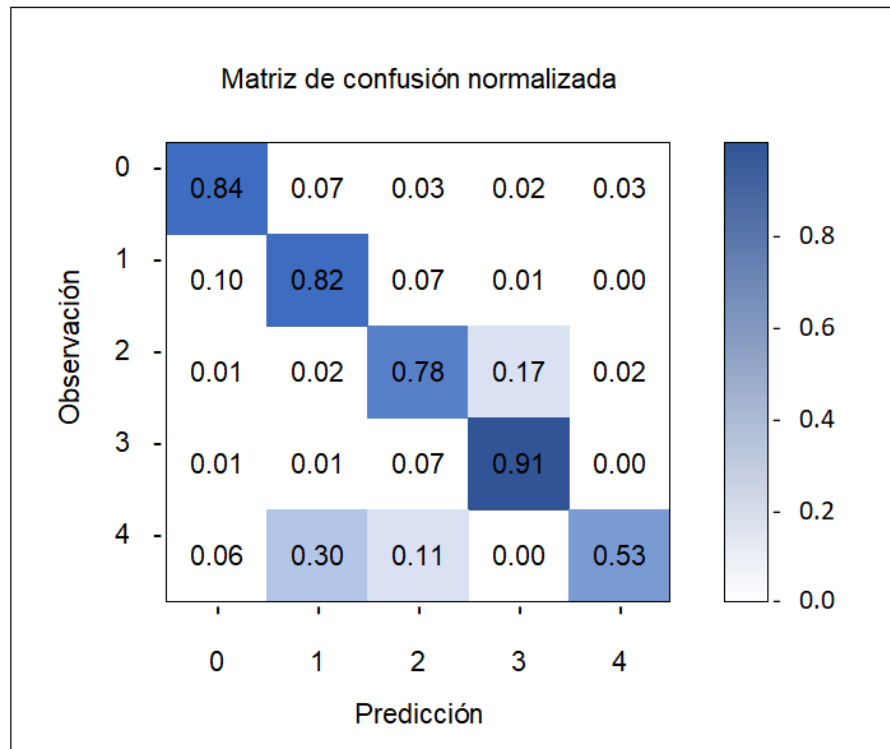


Figura 22. Matriz de confusión del modelo 4

En la Figura 22 se puede apreciar que el modelo presenta un desempeño deficiente en la clasificación de la categoría 4, ya que registra una precisión del 53% en la predicción.

#### Modelo 5 (Modelo implementado en la aplicación móvil)

En la Figura 23 se puede apreciar que el modelo presenta un excelente desempeño en la clasificación de las 5 categorías, ya que registra una precisión superior al 90% en la predicción de cada una de ellas. Además, el modelo no muestra tendencia de confundir alguna categoría con otra. Con base en estos datos, se tomó la decisión de escoger este modelo para ser implementado en la aplicación móvil.

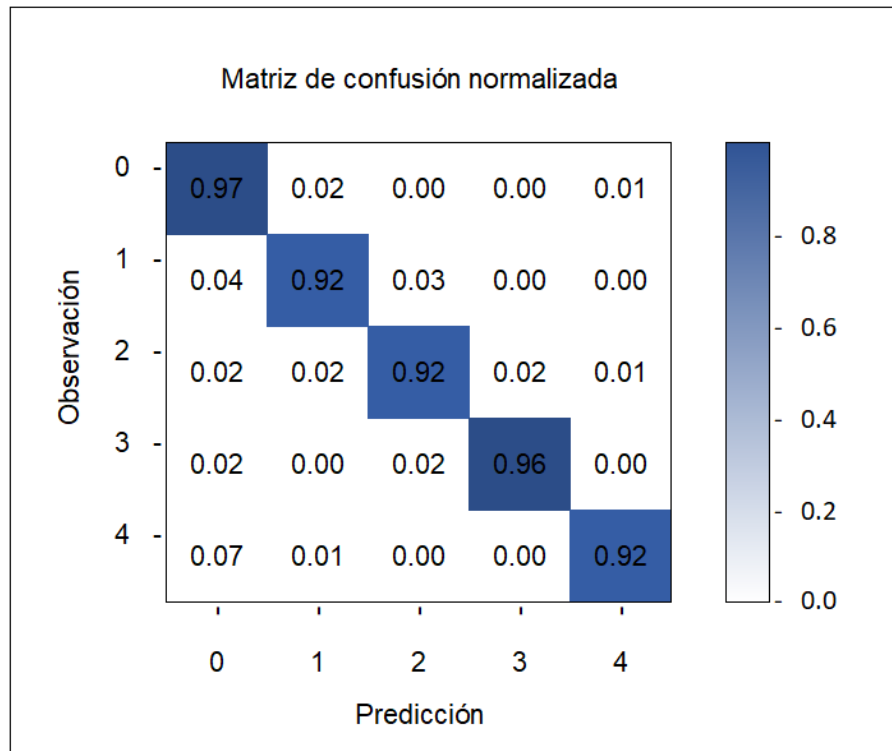


Figura 23. Matriz de confusión del modelo 5

## 11.2. Experimento con aplicación móvil:

En la Tabla 3 se muestran los resultados obtenidos con el primer experimento.

| <b>Experimento con las mismas 3 frases para todas las personas, pero cada frase con diferente cantidad de letras</b> |                            |                      |                            |                      |                            |                      |
|----------------------------------------------------------------------------------------------------------------------|----------------------------|----------------------|----------------------------|----------------------|----------------------------|----------------------|
| <b>Personas</b>                                                                                                      | <b>Frase 1 (14 letras)</b> |                      | <b>Frase 2 (11 letras)</b> |                      | <b>Frase 3 (15 letras)</b> |                      |
|                                                                                                                      | <b># fallas</b>            | <b>% efectividad</b> | <b># fallas</b>            | <b>% efectividad</b> | <b># fallas</b>            | <b>% efectividad</b> |
| Claudia                                                                                                              | 5                          | 88,10                | 5                          | 84,85                | 2                          | 95,56                |
| Juan David                                                                                                           | 5                          | 88,10                | 2                          | 93,94                | 3                          | 93,33                |
| Marga                                                                                                                | 6                          | 85,71                | 4                          | 87,88                | 7                          | 84,44                |
| Rosalba                                                                                                              | 8                          | 80,95                | 2                          | 93,94                | 3                          | 93,33                |
| Carolina                                                                                                             | 2                          | 95,24                | 0                          | 100,00               | 1                          | 97,78                |
| Andrés                                                                                                               | 3                          | 92,86                | 2                          | 93,94                | 2                          | 95,56                |
| Jorge                                                                                                                | 3                          | 92,86                | 7                          | 78,79                | 4                          | 91,11                |
| Camilo                                                                                                               | 6                          | 85,71                | 3                          | 90,91                | 7                          | 84,44                |
|                                                                                                                      |                            | <b>88,69</b>         |                            | <b>90,53</b>         |                            | <b>91,94</b>         |

Tabla 3. Resultados del primer experimento

De acuerdo con la información registrada en la Tabla 3, se puede obtuvo que el porcentaje de efectividad promedio en el primer experimento fue de: 90,39%.

En la Tabla 4 se muestran los resultados obtenidos en el segundo experimento.

| <b>Experimento con 3 frases diferentes por persona, pero con la misma cantidad de letras</b> |                            |                      |                            |                      |                            |                      |
|----------------------------------------------------------------------------------------------|----------------------------|----------------------|----------------------------|----------------------|----------------------------|----------------------|
| <b>Personas</b>                                                                              | <b>Frase 4 (12 letras)</b> |                      | <b>Frase 5 (12 letras)</b> |                      | <b>Frase 6 (12 letras)</b> |                      |
|                                                                                              | <b># fallas</b>            | <b>% efectividad</b> | <b># fallas</b>            | <b>% efectividad</b> | <b># fallas</b>            | <b>% efectividad</b> |
| Claudia                                                                                      | 1                          | 97,22                | 2                          | 94,44                | 0                          | 100,00               |
| Juan David                                                                                   | 5                          | 86,11                | 3                          | 91,67                | 2                          | 94,44                |
| Marga                                                                                        | 3                          | 91,67                | 3                          | 91,67                | 10                         | 72,22                |
| Rosalba                                                                                      | 4                          | 88,89                | 1                          | 97,22                | 3                          | 91,67                |
| Carolina                                                                                     | 2                          | 94,44                | 0                          | 100,00               | 2                          | 94,44                |
| Andrés                                                                                       | 1                          | 97,22                | 1                          | 97,22                | 0                          | 100,00               |
| Jorge                                                                                        | 4                          | 88,89                | 4                          | 88,89                | 2                          | 94,44                |
| Camilo                                                                                       | 2                          | 94,44                | 9                          | 75,00                | 4                          | 88,89                |
|                                                                                              |                            | <b>92,36</b>         |                            | <b>92,01</b>         |                            | <b>92,01</b>         |

Tabla 4. Resultados del segundo experimento

De acuerdo con la información registrada en la Tabla 4, se obtuvo que el porcentaje de efectividad promedio en el primer experimento fue de: 92,13%.

Promediando los porcentajes obtenidos de cada experimento, se obtiene como resultado que el sistema tiene una efectividad del 91,26%.

## 12. CONCLUSIONES

En este trabajo se presentó el diseño y la implementación de una herramienta que presenta mediante una aplicación móvil, un teclado virtual, que puede ser controlado únicamente con 3 posibles comandos: 1 parpadeo, 2 parpadeos o 3 parpadeos. Los parpadeos fueron identificados a partir de la señal electroencefalográfica registrada por el dispositivo *Mindwave Mobile 2* con un canal de medición EEG en el lóbulo frontal de la persona. Esta señal fue procesada por una red neuronal recurrente con dos capas ocultas de tipo LSTM, con el fin de clasificarla en 5 categorías posibles: No parpadeo, 1 parpadeo, 2 parpadeos, 3 parpadeos, Otra acción.

Las curvas de aprendizaje obtenidas del entrenamiento de las redes neuronales implementadas, se tomaron como referencia para definir correctamente el número de épocas en el entrenamiento, basado en la técnica usada para evitar el sobre aprendizaje, conocida como *early-stopping*. Esto sirvió para alcanzar el rendimiento deseado de la red neuronal implementada en la aplicación móvil.

Por otra parte, el proceso de selección de caracteres diseñado en este trabajo para el teclado virtual, resultó ser altamente eficiente, ya que, las personas que participaron en el experimento no presentaron dificultad y no les tomó mucho tiempo familiarizarse con el sistema. Además, la tasa de deletreo alcanzada por este teclado virtual de 5 caracteres por minuto, es superior a la tasa alcanzada por otros trabajos, ya que, según [54], los teclados virtuales controlados por interfaces cerebro-computador hasta el momento han logrado alcanzar en promedio una tasa de 3 letras por minuto únicamente.

Con el porcentaje de exactitud promedio obtenido en este trabajo de 92% se muestra evidencia experimental de la viabilidad y el alto desempeño del método propuesto en este trabajo para la detección de los parpadeos voluntarios de una persona, basado en el uso de redes neuronales recurrentes de tipo LSTM, para la clasificación de una señal EEG. La efectividad de este tipo de redes neuronales también se demostró con su capacidad de generalizar las características de las señales EEG asociadas a los parpadeos de distintas personas, a pesar de haber sido entrenada con datos tomados únicamente de 3 personas.

Por otra parte, según los resultados obtenidos en el experimento realizado, que arrojaron un 91,26% de efectividad en el funcionamiento de la aplicación, se demuestra la viabilidad de comercializar este producto, ya que, cuenta con tecnología de bajo costo. Además, al contar con un solo canal de medición EEG, este dispositivo resulta ser más sencillo, que los dispositivos que cuentan con varios canales de medición EEG.

### **13. TRABAJOS FUTUROS**

Los resultados de este trabajo pueden ser mejorados ampliando el conjunto de datos de entrenamiento de la red neuronal, de forma que se encuentra conformado por señales EEG de un mayor número de personas. En cuanto a la aplicación móvil, en la selección de los caracteres se puede implementar un sistema inteligente de auto completar la palabra que está escribiendo la persona, con el fin de agilizar su proceso de escritura.

Por otra parte, como trabajo futuro también se pueden implementar estos algoritmos de aprendizaje de máquina para el procesamiento de más canales de EEG, con el fin de extraer patrones característicos diferentes al parpadeo, que permitan a la persona ejecutar comandos únicamente con base en ciertos procesos mentales.

## 14. REFERENCIAS

- [1] Alejandro Antonio Torres García, Dr. Carlos Alberto Reyes, Dr. Luis Villaseñor Pineda. Análisis y clasificación de electroencefalogramas (EEG) registrados durante el habla imaginada. Tesis doctoral del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). 2016.
- [2] Saha, S., Mamun, K.A., Ahmed, K.I., Mostafa, R., Naik, G.R., Khandoker, A.H., Darvishi, S., & Baumert, M. (2019). Progress in Brain Computer Interfaces: Challenges and Trends. ArXiv, abs/1901.03442.
- [3] Shurkhay, Vsevolod & Alexandrova, Evgenia & Goryaynov, Sergey & Potapov, Alexander. (2015). The Current State of the Brain-Computer Interface Problem.
- [4] Avinash Kumar Singh, Yu-Kai Wang, Iun-Tai King, Chin-Teng Lin, and Li-Wei Ko. A Simple Communication System based on Brain Computer Interface. Tainan, Taiwan nov. 20-22, 2015.
- [5] Padfield, N., Zabalza, J., Zhao, H., Masero, V., & Ren, J. (2019). EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges. *Sensors* (Basel, Switzerland), 19(6), 1423. <https://doi.org/10.3390/s19061423>
- [6] Chin-Teng Lin, Chih-Sheng Huang, Wen-Yu Yang, Avinash Kumar Singh, Chun-Hsiang Chuang, and Yu-Kai Wang. Real-Time EEG Signal Enhancement Using Canonical Correlation Analysis and Gaussian Mixture Clustering. 2018.
- [7] M. Abo-Zahhad, Sabah M. Ahmed, Sherif N. Abbas. A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals. *I.J. Intelligent Systems and Applications*, 2015, 06, 48-54.
- [8] Kaminer, Jaime & Powers, Alice & Horn, Kyle & Hui, Channing & Evinger, Craig. (2011). Characterizing The Spontaneous Blink Generator: An Animal Model. *The Journal of neuroscience: the official journal of the Society for Neuroscience*. 31. 11256-67. 10.1523/JNEUROSCI.6218-10.2011.
- [9] Mansor, Wahidah & Rani, Mohd & Wahy, Nurfatehah. (2011). Integrating Neural Signal and Embedded System for Controlling Small Motor. 10.5772/22210.
- [10] Michael Varela, Department of Computer Engineering, ITCR, Cartago, Costa Rica. 2015. Raw EEG Signal Processing for BCI Control Based on Voluntary Eye Blinks. *Proceedings of the 2015 IEEE thirty fifth central American and Panama convention (Concapan XXXV)*.

- [11] Merquil Stiven Rodriguez Alvarez, Edwin Eduardo Millan Rojas. Diseño de una interfaz neuronal para personas con discapacidad motora. Universidad Distrital Francisco José de Caldas.
- [12] Paula Brandão Furlan, Almir Kimura Junior, Charles Luiz Silva de Melo, Giovanni Ribeiro Caldeira, Rayza Araújo Bezerra. Portable communication system for disabled speech people controlled by Electroencephalographic signals. 2016.
- [13] Alex Larson, Joshua Herrera, Kiran George and Aaron Matthews. Electrooculography based Electronic Communication Device for Individuals with ALS. 2017.
- [14] Lic. Ricardo A. Koon, Lic. María Eugenia de la Vega. El Impacto Tecnológico En Las Personas Con Discapacidad.
- [15] Naciones Unidas. Objetivos de Desarrollo del Milenio. <https://www.un.org/es>
- [16] Unidad Proyeccion Social. Universidad Santo Tomás. Documento Marco Proyección Social. 2015.
- [17] Zeki Oralhan. A New Paradigm for Region-Based P300 Speller in Brain Computer Interface. Received July 13, 2019, accepted July 31, 2019.
- [18] J Utama and G Palada. Prosthetic arm Controller Based on Brainwaves Spectrum EEG Sensor. 2019 IOP Conf. Ser.: Mater. Sci. Eng. 662 052017.
- [19] Salma Alhagry, Aly Aly Fahmy, Reda A. El-Khoribi. Emotion Recognition based on EEG using LSTM Recurrent Neural Network. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 10, 2017.
- [20] Alexander Craik. Deep learning for electroencephalogram (EEG) classification tasks: a review. et al 2019 J. Neural Eng. 16 031001.
- [21] Sarah N. AbdulkaderAyman AtiaMostafa-Sami M. Mostafa. Brain computer interfacing: Applicationsand challenges. HCI-LAB, Department of Computer Science, Faculty of Computers and Information, Helwan University, Cairo, Egypt. 2015.
- [22] Rahib H. Abiyev, Nurullah Akkaya, Ersin Aytac Irfan Günsel, and Ahmet Çagman. Brain-Computer Interface for Control of Wheelchair Using Fuzzy Neural Networks. 2016.
- [23] Ewelina Sobotnicka, Aleksander Sobotnicki. BCI interface – new opportunities and hopes for the disabled. An overview of available solutions. Institute of Medical Technology and Equipment ITAM Zabrze 118 Roosevelt St., 41-800 Zabrze, Poland.



- [24] Junhua Li, Member, IEEE, Gong Chen, Pavithra Thangavel, Haoyong Yu, Nitish Thakor, Fellow IEEE, Anastasios Bezerianos, Senior Member, IEEE, and Yu SUN, Member, IEEE. A robotic knee exoskeleton for walking assistance and connectivity topology exploration in EEG signal. 6th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob) June 26-29, 2016. UTown, Singapore.
- [25] Electroencephalogram. Alexander J. Casson, Mohammed Abdulaal, Meera Dulabh, Siddharth Kohli, Sammy Krachunov, and Eleanor Trimble. Springer International Publishing AG 2018 EEG.
- [26] Luis, Luis & Gómez-Gil, Jaime. (2012). Brain Computer Interfaces, a Review. Sensors (Basel, Switzerland). 12. 1211-79. 10.3390/s120201211.
- [27] Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Published 2014 by Cambridge University Press.
- [28] Juri Fedjaev. Decoding EEG Brain Signals using Recurrent Neural Networks. MASTER THESIS. Technische Universitat Munchen. 2017.
- [29] Du, K.-L & Swamy, M.N.s. (2014). Recurrent Neural Networks. 10.1007/978-1-4471-5571-3\_11.
- [30] Staudemeyer, Ralf & Morris, Eric. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks.
- [31] Miguel A. Sovierzoski, Fernanda I. M. Argoud, and Fernando M. de Azevedo. Identifying Eye Blinks in EEG Signal Analysis. Proceedings of the 5th International Conference on Information Technology and Application in Biomedicine. Shenzhen, China, May 30-31, 2008.
- [32] Brijil Chambayil, Rajesh Singla, R. Jha. EEG Eye Blink Classification Using Neural Network. WCE 2010, June 30 - July 2, 2010, London, U.K.
- [33] Mohit Agarwal, Raghupathy Sivakumar. Blink: A Fully Automated Unsupervised Algorithm for Eye-Blink Detection in EEG Signals. 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton) Allerton Park and Retreat Center Monticello, IL, USA, September 24-27, 2019.
- [34] Diego V. Escamilla, Cristywelina Escamilla, Ciro A. Rodríguez. Eye Blink Detection Using a Support Vector Machine Classifier. Centro de Innovación en Diseño y Tecnología. Tecnológico de Monterrey. Monterrey N.L., México.

- [35] Dalin Yang and Keum-Shik Hong. Quadcopter Control via Onset Eye Blink Signals: A BCI Study. 2019 19th International Conference on Control, Automation and Systems (ICCAS 2019) Oct. 15~18, 2019; ICC Jeju, Jeju, Korea.
- [36] Mohd Shaifulrizal b Abd Rani, Wahidah bt. Mansor. DETECTION OF EYE BLINKS FROM EEG SIGNALS FOR HOME LIGHTING SYSTEM ACTIVATION. Proceeding of the 6th International Symposium on Mechatronics and its Applications (ISMA09), Sharjah, UAE, March 24-26,2009.
- [37] Won-Du Chang a, Ho-Seung Chaa, Kiwoong Kimb, Chang-Hwan Ima. Detection of eye blink artifacts from single prefrontal channel electroencephalogram. Computer methods and programs in biomedicine 124 (2016) 19-30.
- [38] Roy, Raphaëlle N. and Charbonnier, Sylvie and Bonnet, Stéphane Eye blink characterization from frontal EEG electrodes using source separation and pattern recognition algorithms. (2014) Biomedical Signal Processing and Control, 14. 256-264. ISSN 1746-8094.
- [39] Zoran Tiganj, Mamadou Mboup, Christophe Pouzat, Bouchra Lotfi. An algebraic method for eye blink artifacts detection in single channel EEG recordings. 17TH INTERNATIONAL CONFERENCE ON BIOMAGNETISM ADVANCES IN BIOMAGNETISM – BIOMAG2010, Mar 2010, Dubrovnik, Croatia. pp.175-178, ff10.1007/978-3-642-12197-5\_38ff. fhal-00518627ff.
- [40] L. Ramya Stephygraph, N. Arunkumar, and V. Venkatraman. Wireless Mobile Robot Control through Human Machine Interface using Brain Signals. 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. pp.596-603.
- [41] Paula Ivone Rodriguez, Jose Mejia, Boris Mederos, Nayeli Edith Moreno, and Victor Manuel Mendoza. Acquisition, analysis and classification of EEG signals for control design. Universidad Autónoma de Ciudad Juárez.
- [42] Xiaotong Gu, Zehong Cao, Alireza Jolfaei, Peng Xu, Dongrui Wu, Tzyy-Ping Jung, and Chin-Teng Lin. EEG-based Brain-Computer Interfaces (BCIs): A Survey of Recent Studies on Signal Sensing Technologies and Computational Intelligence Approaches and Their Applications. 2020.
- [43] Aasim Raheel, Syed M. Anwar, Muhammad Majid, Bilal Khan, Ehatisham-ul-Haq. Real Time Text Speller based on Eye Movement Classification Using Wearable EEG Sensors. SAI Computing Conference 2016.

- [44] Trung-Hau Nguyen, Da-Lin Yang, and Wan-Young Chung. A High-Rate BCI Speller Based on Eye-Closed EEG Signal. Received May 10, 2018, accepted June 11, 2018.
- [45] Atish Udayashankar, Amit R Kowshik, Chandramouli S, H S Prashanth. ASSISTANCE FOR THE PARALYZED USING EYE BLINK DETECTION. 2012 Fourth International Conference on Digital Home.
- [46] Mohammad H. Alomari, Ayman AbuBaker, Aiman Turani, Ali M. Baniyounes, Adnan Manasreh. EEG Mouse: A Machine Learning-Based Brain Computer Interface. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 4, 2014.
- [47] Mr. Ramesh C R, Prof. Lyla B Das. Brain Computer Interface Device for Speech Impediments. 2015 International Conference on Control, Communication & Computing India (ICCC) 19-21 November 2015 Trivandrum.
- [48] Brain Wave Signal (EEG) of NeuroSky, Inc. December 15, 2009.
- [49] Brijil Chambayil, Rajesh Singla, R. Jha. Virtual Keyboard BCI using Eye blinks in EEG. 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications.
- [50] Suneth Pathirana, David Asirvatham, Md Gapar Md Johar. Designing Virtual Keyboards for Brain-Computer Interfaces. 2018.
- [51] Khalil Ullah, Mohsin Ali, Muhammad Rizwan, Muhammad Imran. Low-Cost Single-Channel EEG Based Communication System for People with Lock-in Syndrome. 2011.
- [52] Lucas B. Coffey. Assessing Ratio-Based Fatigue Indexes Using a Single Channel EEG. University of North Florida. 2018.
- [53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. 15(56): 1929-1958, 2014.
- [54] Swagata Das, Devashree Tripathy, Jagdish Lal Raheja. Real-Time BCI System Design to Control Arduino Based Speed Controllable Robot Using EEG. Springer Singapore, 2018.