

**Traductor de lengua de señas ecuatoriana mediante inteligencia artificial para la inclusión educativa .
Ecuadorian sign language translator using artificial intelligence for educational inclusion***Wilson Oswaldo Sánchez Macías, Daniela Elizabeth Sánchez Orozco, Jessica Marlene Ramos Anchundia.***CIENCIA E INNOVACIÓN EN
DIVERSAS DISCIPLINAS
CIENTÍFICAS.****Enero - Junio, V°6-N°1; 2025**

- ✓ **Recibido:** 27/02/2025
- ✓ **Aceptado:** 17/03/2025
- ✓ **Publicado:** 30/03/2025

PAIS

- Ecuador, Guayaquil
- Ecuador, Guayaquil
- Ecuador, Guayaquil

INSTITUCION

- Universidad Estatal de Milagro
- Escuela Superior Politécnica del Litoral
- Universidad Estatal de Milagro

CORREO:

- wsanchezm2@unemi.edu.ec
- dansoroz@espol.edu.ec
- jramosa7@unemi.edu.ec

ORCID:

- <https://orcid.org/0009-0000-3003-1279>
- <https://orcid.org/0000-0001-8242-2239>
- <https://orcid.org/0000-0002-5303-3549>

FORMATO DE CITA APA.

Sánchez, W. Sánchez, D. Ramos, J. (2025). Traductor de lengua de señas ecuatoriana mediante inteligencia artificial para la inclusión educativa. *Revista G-ner@ndo*, V°6 (N°1.). 2981 - 2998.

Resumen

El presente trabajo tiene como objetivo diseñar un sistema embebido para la traducción de frases de la lengua de señas ecuatoriana, mediante la implementación de un circuito compuesto por un sensor giroscopio-acelerómetro MPU6050 y un microcontrolador ESP32. Se incluyeron cuatro señales básicas, escogidas de manera que fue posible realizar el reconocimiento únicamente con el movimiento del antebrazo. Para la interpretación de estas señales se implementó un modelo de inteligencia artificial basado en redes neuronales, el cual fue entrenado con datos recopilados por el dispositivo diseñado. Posteriormente, el modelo fue comprimido e implementado en el microcontrolador ESP32. La inferencia en tiempo real se realizó obteniendo datos desde el sensor MPU6050, los cuales son enviados al microcontrolador para la interpretación de las señales y la presentación de las palabras correspondientes escritas en español. Palabras clave: Traductor de lengua de señas; microcontrolador; inteligencia artificial.

Abstract

This work aims to design an embedded system for translating phrases from Ecuadorian Sign Language by implementing a circuit composed of an MPU6050 gyroscope-accelerometer sensor and an ESP32 microcontroller. Four basic signs were included, selected so that recognition was possible using only forearm movement. To interpret these signs, an artificial intelligence model based on neural networks was implemented and trained with data collected by the designed device. The model was then compressed and deployed on the ESP32 microcontroller. Real-time inference was performed by obtaining data from the MPU6050 sensor, which was sent to the microcontroller for signal interpretation and the presentation of the corresponding words in written Spanish.

Keywords: Sign language translator; Microcontroller; Artificial intelligence.

Introducción

En Ecuador la gran mayoría de personas no conocen lengua de señas, por lo tanto, las personas en situación de discapacidad de origen auditivo enfrentan muchas dificultades para comunicarse mediante este lenguaje. La contratación de intérpretes no es una opción viable para la mayoría de los casos debido a su alto costo (CONADIS, S.f.).

Otro obstáculo percibido es la falta de estandarización de las señas. La ausencia de un diccionario o entidad regulatoria ha provocado que existan múltiples variaciones de este lenguaje, de manera que en cada país se encuentren versiones distintas de este (Villa, 2021). Según el Consejo Nacional para la Igualdad de Discapacidades – CONADIS al corte de noviembre de 2024, en Ecuador existe 68157 niños menores de 12 años con discapacidades auditivas y de lenguaje. Siendo la comunicación un elemento vital para que estas personas consigan integrarse eficientemente a la sociedad, resulta de gran importancia realizar investigaciones basadas en la lengua de señas ecuatoriana (Estadística de discapacidad CONADIS, 2024).

En el caso de que las personas con esta condición son niños pequeños quienes aún no tienen la capacidad cognitiva para poder comunicarse con el lenguaje especializado de señas es más fácil que ellos aprendan signos simples para representar frases sencillas. (La Valle y otros, 2024). La presente investigación propone el diseño de un sistema embebido capaz de reconocer frases en lengua de señas ecuatoriana y traducirlas al lenguaje español. De esta forma, se espera aportar en el desarrollo de estrategias para la inclusión de niños con este tipo de discapacidad en el sistema educativo, mejorando su comunicación y por ende su calidad de vida.

La estrategia propuesta consiste en la adquisición de señales de movimiento del antebrazo de un usuario. Inicialmente, se creará una base de datos con registros de cuatro frases indispensables del lenguaje español: “Buenos días”, “Buenas tardes”, “¿Cómo estás?” y “Por favor”. Estos datos se emplearán para entrenar un modelo de inteligencia artificial, el cual posteriormente será embebido en un microcontrolador que será el encargado de realizar las

futuras identificaciones. Una vez que se posea el modelo entrenado, el microcontrolador será capaz de receptar las señales de movimiento de la seña realizada, y asociarlas a la frase en español correspondiente, para presentarla de manera escrita en la pantalla de un computador.

Para llevar a cabo este trabajo será necesario programar el microcontrolador ESP32 mediante lenguaje C para la recepción de las señales adquiridas por el sensor MPU6050 correspondientes a las frases de lengua de señas escogidas, Implementar un modelo de inteligencia artificial entrenado en el microcontrolador ESP32 para la clasificación de las señales receptadas por el sensor MPU6050. Y finalmente transmitir los datos obtenidos desde el sensor MPU6050 hasta un microcontrolador ESP32.

Existen varias alternativas o técnicas de reconocimiento de lengua de señas. A continuación, se describe el uso de algunas tecnologías para el diseño e implementación de sistemas embebidos dedicados a esta tarea. Una de las técnicas más investigadas es el reconocimiento visual, la cual consiste en la captación de señas en tiempo real por medio de una webcam o cámara integrada en una laptop permitiendo identificar tanto movimientos manuales como expresiones faciales (Farid y otros, 2024). Para la extracción de características manuales se requiere procesar las imágenes para eliminar el fondo, analizar contraste, sombras e iluminación para identificar situaciones donde una mano se sobreponga a otra, extracción de características, y finalmente identificación del movimiento. Además, para el procesamiento de expresiones faciales es necesario involucrar consideraciones más complejas como la postura del usuario, la posición de la cabeza, la línea de vista, la posición de los labios, etc. (Tian y otros, 2024).

Otro tipo de aproximación a la resolución de esta problemática es mediante el uso de sensores. En España, se propuso el diseño y construcción de un guante capaz de detectar la seña ejecutada por un usuario y el reconocimiento de esta mediante un sistema inteligente desarrollado en C++. Para esto, se incorporó un arreglo de Flex resistivo y piezoeléctrico (Yang

y otros, 2024), que envían señales analógicas en función de la posición de la mano y dedos del usuario. Además, se utilizó un sistema inercial con un acelerómetro para adquirir información sobre la posición, velocidad y orientación de la mano, así como identificar la rotación de la muñeca. (Jiménez-Montanés & Holgado-Terriza, 2018), En este caso la limitante es el costo y complejidad del dispositivo.

Una técnica adicional aplicada al reconocimiento de lenguaje de señas mediante sensores es la electromiografía, representada por las siglas EMG. Esta consiste en la captación de señales musculares (Lee y otros, 2025) Se desarrolló un proyecto utilizando un sistema de adquisición de datos Biopac MP100, para receptar señales EMG. Posteriormente, se realizó el procesamiento, filtro y extracción de características de las señales. Esta información permitió realizar la identificación de frases representadas mediante lengua de señas a través de una red neuronal artificial (ANN) de tres capas. A pesar de que este proyecto logró resultados con un 79.4% de éxito, se resaltaron algunas dificultades en el manejo de señales EMG, entre estas su alta sensibilidad al ruido, lo que implica la necesidad de procesos de filtrado y el análisis de múltiples características para evitar información errónea. Además, el dispositivo Biopac MP100 es un equipo especializado con un costo elevado, superior a los \$800, lo que convierte a esta implementación en una alternativa poco comercial (Pomboza-Junez & Terriza, 2016).

En Guayaquil, Ecuador se desarrolló un sistema embebido para la identificación de lengua de señas, basado en EMG y FPGA. Para obtener información que permita desarrollar aplicaciones de reconocimiento de gestos, pueden identificar gestos con precisión utilizando algoritmos de aprendizaje automático, se utiliza una sola característica que es el valor RMS con el objetivo de minimizar el recurso computacional. (Cedeño y otros, 2019). Otro trabajo sobre lengua de señas ecuatoriana utiliza tecnología de visión por computadora para la traducción en tiempo real de los gestos de señas al lenguaje escrito. El estudio utilizó un conjunto de datos en letras del alfabeto, números y frases comunes en Ecuador (Carlosama y otros, 2024).

El presente proyecto propone el diseño de un sistema embebido capaz de identificar las frases en lenguaje de seña realizadas. El sistema de reconocimiento se realiza mediante una tarjeta ESP32, la cual es capaz de ejecutar un modelo de inteligencia artificial previamente entrenado. El entrenamiento de dicho modelo se realizará utilizando el software de programación Python, el cual presenta la ventaja de ser de código abierto. Para la recepción de datos de movimiento de la persona se utilizará un sensor MPU6050, el cual permite obtener datos de acelerómetro y giroscopio en 3 ejes cartesianos. Esto presenta la ventaja de obtener varios datos mediante un único dispositivo, a diferencia de otras técnicas que requieren el uso de varios equipos simultáneamente.

Este tipo de acercamiento evita las complejidades de los sistemas basados en visión artificial, los cuales requieren un procesamiento de datos mucho más robusto al involucrar múltiples parámetros, como se describió anteriormente. Además, estas señales son menos susceptibles al ruido que las señales EMG (Zhang y otros, 2024). Se resalta además que el hardware propuesto en esta investigación posee precios económicos, en contraparte a los dispositivos empleados en los trabajos mencionados. Además, la ESP32 posee ventajas como la adaptación de módulos para comunicación vía Wifi, lo cual permite que el diseño propuesto sea expandible para generar mejoras en el futuro. Finalmente, es importante resaltar que esta investigación se basa en la lengua de señas ecuatoriana, por lo cual brindará beneficios a la comunidad nacional, siendo un área poco estudiada localmente.

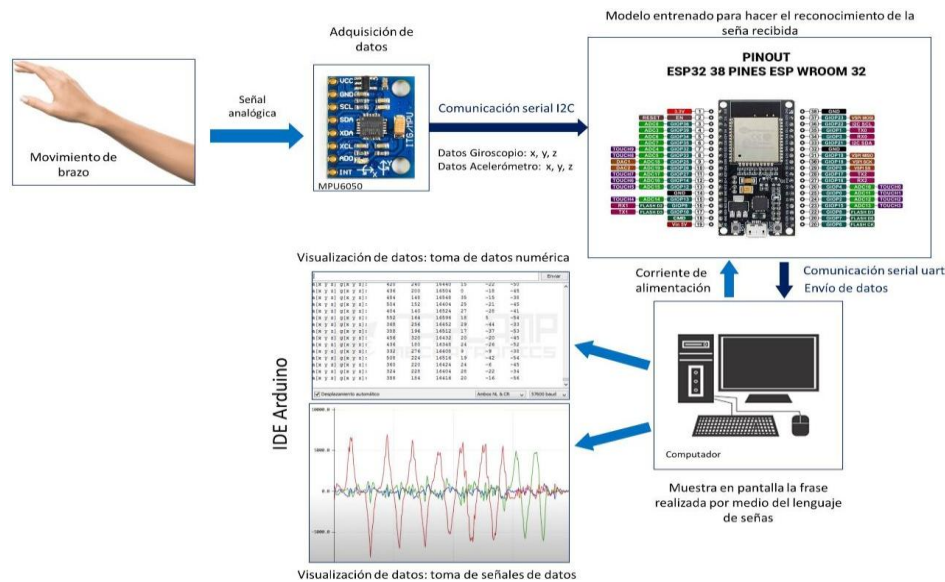
Métodos y Materiales.

Para la captación de señales se utilizó un sensor MPU6050 el cual tiene incorporados un giroscopio y un acelerómetro, que brindan señales en los tres ejes cartesianos. Se determinó que las frases escogidas posean una duración aproximadamente igual, de 2.5 segundos Para la

creación de la base de datos se utilizó una frecuencia de muestreo de 100Hz. Luego, el modelo de inteligencia artificial se entrenó empleando el lenguaje de programación de código abierto Python. Finalmente, este fue embebido en un microcontrolador ESP32, para el reconocimiento de las señas realizadas y la comunicación con el computador.

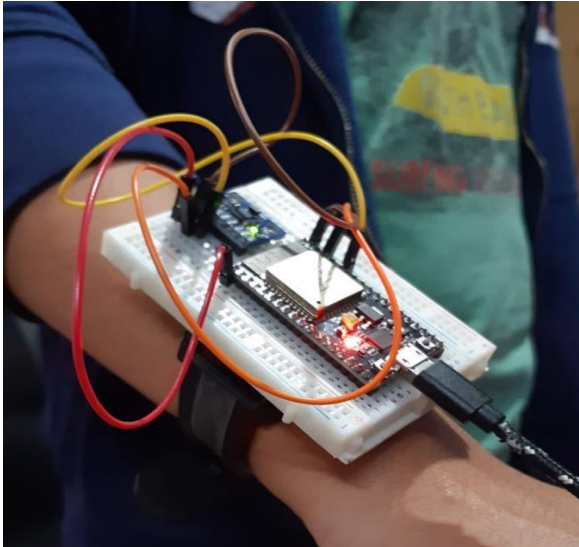
Figura 1: Diagrama de Bloques

Diagrama de bloques



En el diagrama de bloques se pueden apreciar los pasos realizados para llevar a cabo el correcto funcionamiento del proyecto. El sensor en el antebrazo del usuario envía una señal analógica del movimiento al sensor MPU6050, esta señal es convertida a digital y enviada por medio de comunicación serial I2C al ESP32 al modelo entrenado para hacer el reconocimiento de la señal recibida. El microcontrolador envía la señal por medio de comunicación serial UART a un computador donde se mostrará por pantalla la frase realizada por medio del lenguaje de señas en español.

Figura 2: Usuario Portando el Dispositivo



Para realizar la recolección de datos del sensor MPU6050, se creó un código en lenguaje C, a través de la plataforma ESP-IDF en Visual Studio Code, el cual fue embebido en la ESP32. Mediante este programa se estableció la comunicación serial I2C entre ambos dispositivos ESP32 maestro, y el sensor como el esclavo en la comunicación. Además, se estableció la comunicación entre la ESP32 y un computador, mediante el puerto UART, para transmitir las lecturas del sensor a una base de datos.

Para esto, se escogieron 4 frases de lenguaje de señas que serán nombradas como “tareas”. Las frases elegidas fueron “Buenos días”, “Buenas tardes”, “¿Cómo estás?” y “Por favor”.

Dado que se trabajó únicamente con un sensor en el antebrazo del usuario y no en los dedos, se seleccionó frases cuyas acciones predominantes fueran realizadas por el movimiento del antebrazo brazo.

A continuación, se representan los movimientos correspondientes a cada tarea mencionada (CONADIS, s.f.).

Figura 3: Tarea 1 “Buenos días”

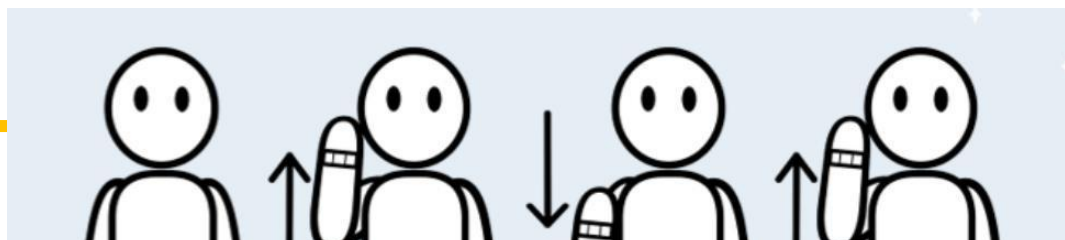


Figura 4: Tarea 2 “Buenas tardes”

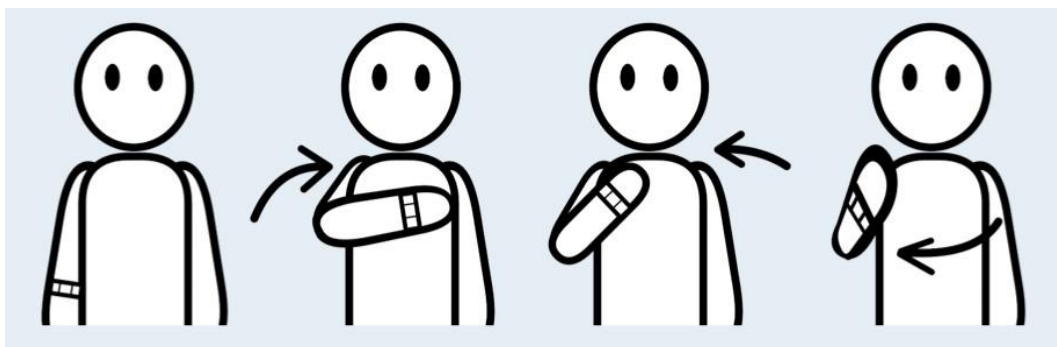


Figura 5: Tarea 3 “¿Cómo estás?”

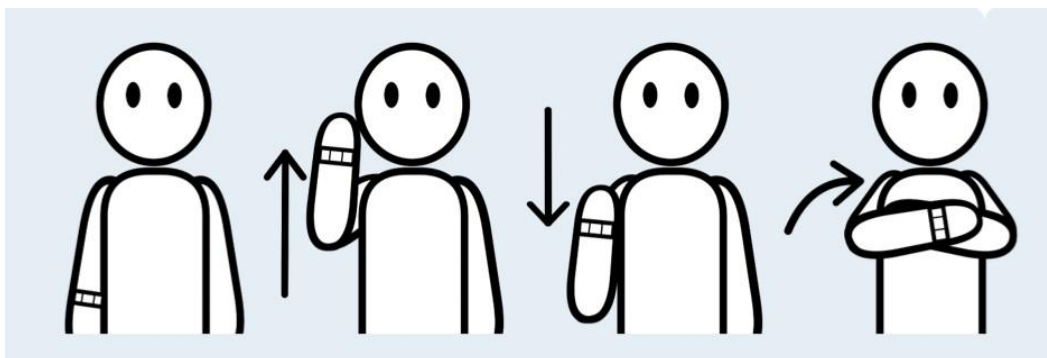
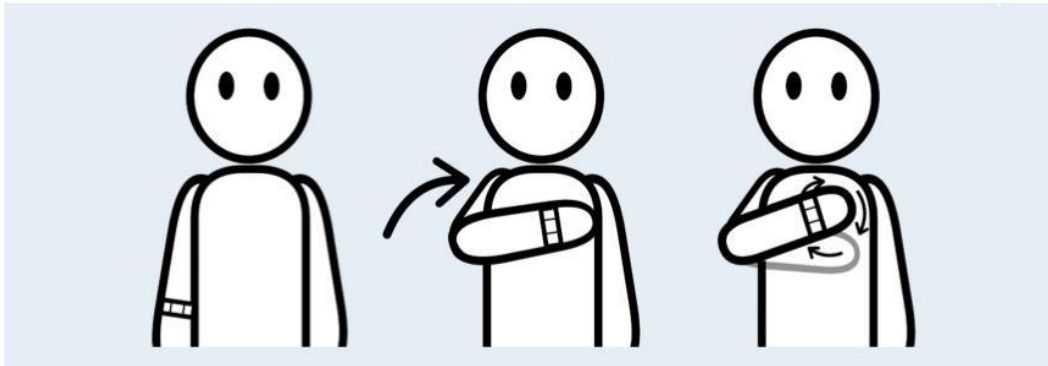


Figura 6: Tarea 4 “Por Favor”

Figura 6: Tarea 4 "Por Favor"

A través del software Coolterm fue posible visualizar los datos obtenidos y grabarlos en archivos .csv. En base a la frecuencia y tiempo de muestreo mencionados, cada tarea fue representada por un archivo .csv de 250 filas y 6 columnas (una por cada variable del sensor). Se realizaron 100 repeticiones por cada frase o tarea escogida, obteniéndose un dataset final de 400 archivos .csv.

Mediante un programa realizado en Matlab se calculó el RMS de cada una de las curvas de las señales para utilizarlas como entradas del sistema. La figura 7 muestra un ejemplo d ellos datos adquiridos de una de las muestras.

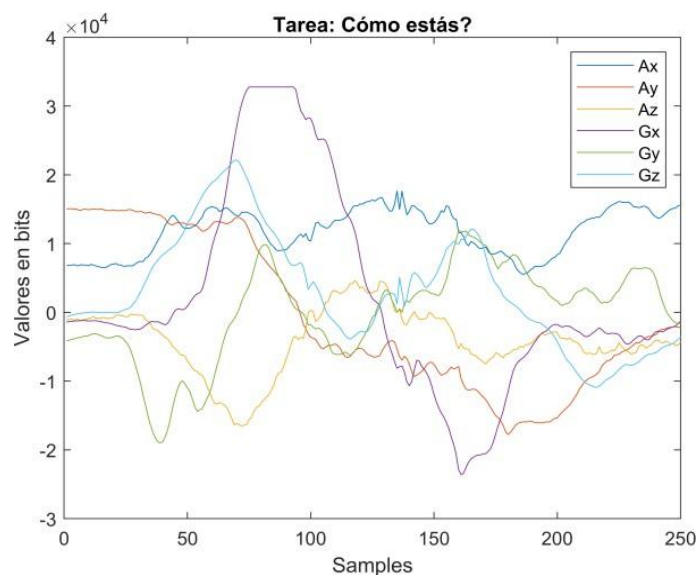


Figura 7: Ejemplo de gráfica de las señales correspondientes a la tarea “¿Cómo estás?”

Para el entrenamiento del modelo de clasificación por medio de inteligencia artificial, se empleó el lenguaje de código abierto Python. Se adaptaron los archivos .csv del dataset. utilizando las librerías Pandas y Numpy, Los datos fueron filtrados aplicando un promedio de ventana móvil. Luego se normalizaron los datos, mediante normalización de mínimo-máximo, de manera que todos los valores se conviertan al rango de 0 a 1. Debido a los excelentes resultados obtenidos en Matlab, se decidió trabajar con la característica RMS. Para obtener una mayor resolución de los datos, se trabajó con una ventana de 50 valores. Es decir, se halló el RMS de cada intervalo de 50 valores en cada una de las variables de una muestra o archivo .csv. Dado que cada archivo cuenta con 250 filas y 6 columnas, se obtuvieron en total 5 valores RMS para cada variable, de manera que cada archivo .csv se representó mediante un vector de 30 valores. Cada uno de estos vectores se asoció a la tarea a la cual corresponde mediante un identificador o label.

Los datos se dividieron en dos grupos, 70% para el entrenamiento, y el 30% restante como datos de validación. Se configuró la arquitectura del modelo en cuatro capas: capa de entrada, capa de salida, y dos capas intermedias. El entrenamiento se realizó en 200 épocas. Posteriormente, los datos de validación permitieron probar el funcionamiento del modelo. La matriz de confusión permitió observar que el modelo logró adquirir una exactitud del 100%, ya que, en todos los escenarios de los datos de validación, el modelo identificó la tarea correspondiente de manera acertada.

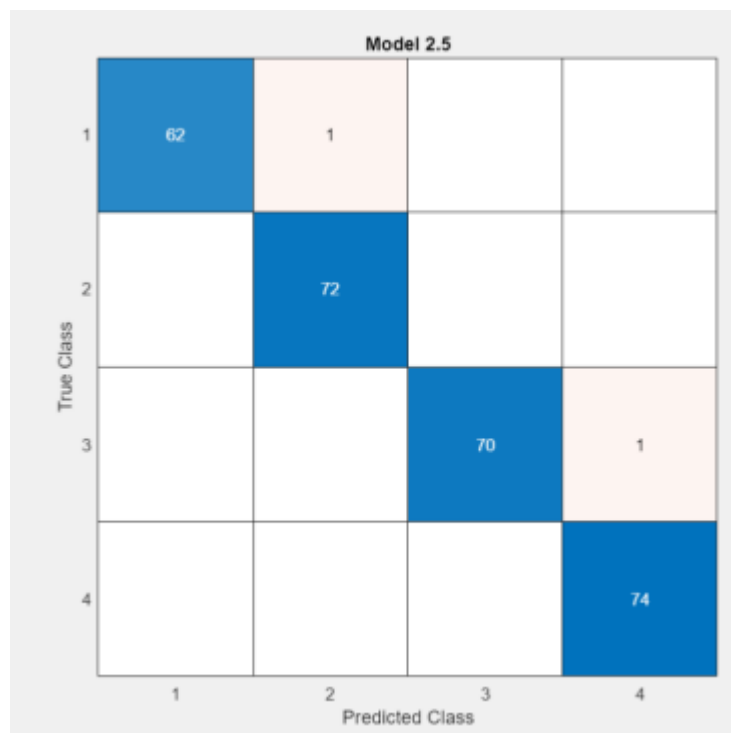
Para implementar el sistema en tiempo real, se desarrolló un código a través de Platform IO que realice la preparación necesaria para integrar el modelo previamente entrenado en la ESP32. El programa toma datos de las 6 variables del sensor MPU6050 durante 2.5 segundos (tiempo promedio de una tarea) con una frecuencia de muestreo de 100Hz. Luego, estos datos son normalizados con la misma técnica que en el procesamiento de datos del modelo para

transformarlos a un rango de 0 a 1. Finalmente, se extrae la característica RMS con una ventana de 50 valores por variable, obteniéndose un vector de 30 valores RMS que represente la lectura del movimiento realizado. De esta manera, los datos se organizan en el mismo formato con el cual fue entrenado el modelo. Para invocar al modelo entrenado, se creó una clase `NeuronalNetwork`, Esta clase requiere el uso de la librería `TensorFlowLite`. De esta manera el modelo clasifica los datos determinando a cuál tarea corresponden.

Análisis de Resultados

Con la utilización de la característica RMS de cada uno de los archivos .csv de la base de datos., se observó que el algoritmo de clasificación lograba identificar la tarea correctamente con una exactitud del 99%, lo cual indica que la característica escogida (RMS) y la calidad del dataset fueron adecuadas.

La figura 8: muestra en azul las veces en que el sistema logró identificar correctamente la tarea.



En las pruebas de funcionamiento del modelo, la matriz de confusión (Figura 9) permitió observar que el modelo logró adquirir una exactitud del 100%, en todos los escenarios de los datos de validación, el modelo identificó la tarea correspondiente de manera acertada. Es decir, en todas las tareas con identificador 0 evaluadas, el modelo reconoció que estas pertenecían a la tarea 0 (23 aciertos de 23 casos), manteniéndose este comportamiento para las demás tareas.

Figura 9: Matriz de confusión del Modelo Entrenado

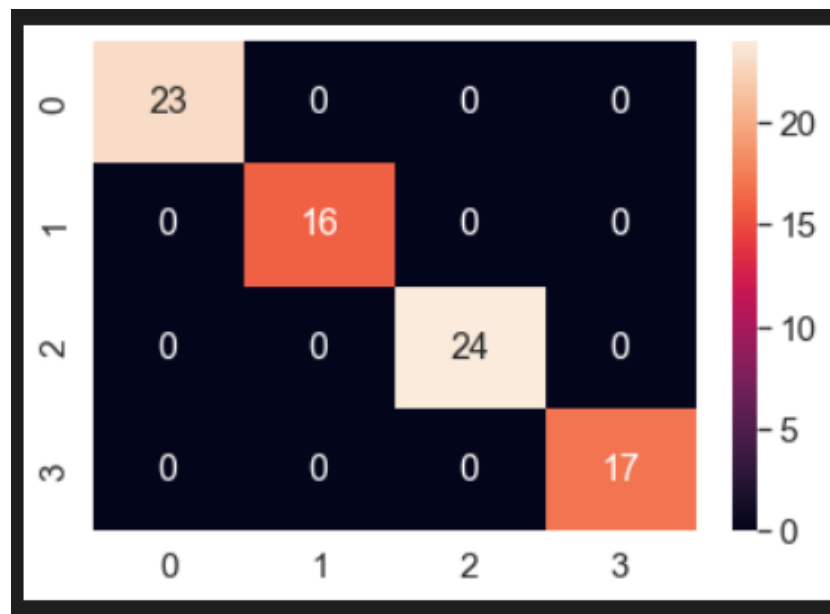
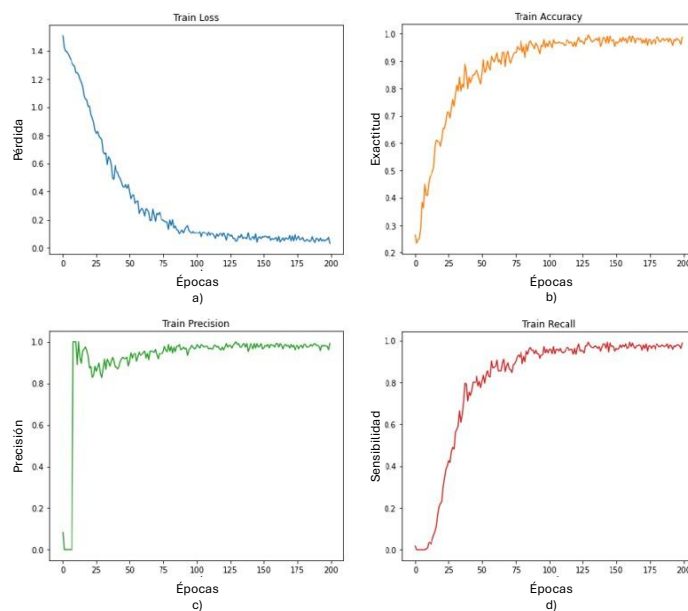


Figura 8: Resultados de Prueba de Clasificación

Se evaluaron además algunas métricas de desempeño para validar el algoritmo. La problemática presentada consiste en una clasificación multiclase, es decir, con una cantidad N de categorías. Para la evaluación de las métricas, se analizó cada categoría individualmente, como si se tratase de una clasificación binaria. Estas métricas se basan en cuatro valores principales: cantidad de Verdaderos Positivos (TP), cantidad de Falsos Positivos (FP), cantidad de Verdaderos Negativos (TN), y cantidad de Falsos Negativos (FN). En la Figura 10 se muestran los resultados correspondientes a la primera categoría: “Buenos días”.

La función de pérdida (Loss) representa la desviación entre las predicciones del modelo y las categorías reales (Elliot-Kerr y otros, 2024). Como puede observarse en la subfigura 10a, la función decrece rápidamente. Se analizó también la exactitud o accuracy del modelo, la cual indica la proporción de valores clasificados correctamente en relación con el valor total de datos. Luego de 75 épocas de entrenamiento, aproximadamente, el modelo ya es capaz de clasificar correctamente el 90% de los datos. La precisión aumenta abruptamente desde el inicio del entrenamiento, lo cual indica que la proporción entre valores positivos clasificados correctamente y todos los valores que el modelo asumió positivos, es muy alta incluso con pocas épocas de entrenamiento. Finalmente, la sensibilidad o recall brinda información sobre los resultados positivos clasificados correctamente, y el total de valores positivos reales. A partir de 75 épocas esta métrica ya presenta resultados superiores al 90%.

Figura 10: Métricas de desempeño del modelo. a) Función de pérdida. b) Exactitud. c) Precisión. d) Sensibilidad



$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precisión = \frac{TP}{TP + FP}$$

$$Sensibilidad = \frac{TP}{TP + FN}$$

Discusión

Los resultados obtenidos permiten demostrar el excelente desempeño del modelo generado. Se logró clasificar todas las frases/señas correctamente, indicando que la selección de frases fue adecuada. Además, esto demuestra la buena calidad del dataset recopilado. (Gong y otros, 2023).

Como se mencionó anteriormente, las métricas presentaron un comportamiento altamente satisfactorio. En cuanto a la función de pérdida, esta mostró cómo el modelo aprende a una velocidad adecuada, pudiendo disminuir significativamente sus falsos positivos y falsos negativos en menos de la mitad del tiempo de entrenamiento. Evaluando todas las métricas consideradas, el modelo alcanzó valores aceptables en menos de la mitad de las épocas utilizadas. Esto es ventajoso ya que podría permitir la implementación de un modelo dinámico, capaz de reentrenarse parcialmente para ajustarse a nuevas circunstancias, disminuyendo el tiempo de entrenamiento, pero manteniendo un rendimiento satisfactorio. (Huang y otros, 2025).

En trabajos futuros, se pretende evaluar el algoritmo utilizando un dataset más amplio que incluya otras palabras o expresiones. El poseer una variedad más amplia de gestos recopilados permitirá asegurar que el algoritmo desarrollado no sufre de sobreajuste. Esto se considera negativo pues condiciona al modelo a aprenderse todo el dataset, en lugar de identificar patrones. Además, resulta importante evaluar cómo se adaptaría el modelo a un escenario menos ideal con señas que posean más similitudes entre sí. Otra mejora consistiría en aumentar el número de sujetos empleados para la recopilación de la base de datos, para evitar sesgos según el usuario.

El uso de sensores de movimiento se considera una decisión acertada ya que logra evitar inconvenientes como la sobreposición de dedos que dificulta el reconocimiento por medio de

imagen. Además, el uso de un dispositivo ligero y fácil de adherir a la muñeca, presenta una experiencia de usuario más ágil y cómoda. En contraste, la traducción de lenguaje de señas por imagen o video requiere el posicionamiento de una cámara en el ángulo y distancia apropiados. Esto puede ser viable en un ambiente como conferencias o reuniones virtuales, sin embargo no es una estrategia práctica para conversaciones del día a día. Por lo tanto, dentro del ámbito educativo, se considera que el uso de sensores de movimiento brinda a los alumnos una herramienta más versátil.

Adicionalmente, es importante destacar los beneficios de comprimir el modelo en un dispositivo de bajos recursos como lo es la ESP32. Esto permite la reducción de costos al no requerirse procesadores de alta gama. Además, los sensores compatibles con esta tecnología también son económicos y de alta accesibilidad local. Con un costo inferior a los \$100, nuestro sistema se posiciona como una de las opciones más económicas en el mercado. Además, este diseño posee una alta escalabilidad al poder incorporarse sensores de otra naturaleza para identificar una mayor variedad de movimientos.

Conclusiones

EL sensor MPU6050 permitió obtener datos del acelerómetro y giroscopio en los tres ejes cartesianos, recopilando información de 6 variables simultáneamente. Estos datos fueron representados mediante valores de 16 bits con signo por cada variable los cuales se enviaron al microprocesador ESP32. Y finalmente se enviaron mediante el protocolo UART al computador para visualizar los datos leídos en pantalla.

Las tareas o frases que conforman el dataset fueron escogidas correctamente puesto que todas poseían una duración aproximada de 2.5 segundos y presentaban movimientos diferentes donde predominaban las acciones del antebrazo, en el cual se coloca el sensor. Esto se pudo concluir gracias a un procesamiento previo de los datos mediante Matlab, en el cual se utilizó la

característica RMS para implementar un algoritmo de clasificación que presentó una exactitud superior al 99%. Por lo tanto, se demostró que el dataset es clasificable y que los datos fueron tomados adecuadamente con una frecuencia de muestreo de 100Hz y un tiempo promedio de 2.5 segundos.

Se utilizó la librería Tensorflow para entrenar un modelo inteligente para la clasificación de datos. Para esto, se filtraron y normalizaron los datos del dataset, para posteriormente extraer la característica RMS de cada archivo .csv. Esta información permitió entrenar un modelo mediante aprendizaje supervisado. Los datos de validación demostraron que el modelo posee un 100% de exactitud, identificando la tarea asociada correctamente en todos los casos de validación. La función de pérdida logró demostrar cómo el modelo disminuye las desviaciones entre las predicciones y valores reales, a la vez aumentar rápidamente la exactitud, precisión y sensibilidad en su clasificación. De esta forma, se concluyó que el modelo entrenado obtenido posee una excelente calidad y desempeño.

Referencias bibliográficas

- Carlosama, J., Criollo, S., Játiva, C., Mina, V., & Velastegui, S. (2024). Ecuadorian Sign Language Detection in Real Time. *ResearchGate*, 298-311.
- Cedeño, C., Cordova-Garcia, J., Asanza, V., Ponguillo, R., & Muñoz, L. (2019). k-NN-Based EMG Recognition for Gestures Communication with Limited Hardware Resources. *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing*, 812-817. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00170>
- CONADIS. (s.f.). *Consejo Nacional de Igualdad de Discapacidades (CONADIS)*. Diccionario de Lengua de señas ecuatoriana Gabriel román: <http://www.plataformaconadis.gob.ec/~platafor/diccionario/>
- Elliot-Kerr, S., Hilton, J., Parkins, K., Ujjwal, K., Huston, C., Swedosh, W., & Penman, T. (2024). Loss functions for spatial wildfire applications. *Environmental Modelling & Software*, 173. <https://doi.org/https://doi.org/10.1016/j.envsoft.2024.105944>
- Estadística de discapacidad CONADIS*. (Noviembre de 2024). consejo nacional para la igualdad de discapacidades: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>
- Farid, F., Hashim, N., Abdullah, J., Bhuiyan, M., Kairanbay, M., Yusoff, Z., Karim, H., Mansor, S., Sarker, M., & Ramasamy, G. (2024). Single Shot Detector CNN and Deep Dilated Masks for Vision-Based Hand Gesture Recognition From Video Sequences. *IEEE Access*, 12, 28564-28574. <https://doi.org/https://doi.org/10.1109/ACCESS.2024.3360857>
- Gong, Y., Liu, G., Xue, Y., Li, R., & Meng, L. (2023). A survey on dataset quality in machine learning. *Information and Software Technology*, 162. <https://doi.org/https://doi.org/10.1016/j.infsof.2023.107268>
-

- Huang, W., Zhang, Y., Guo, S., Shang, Y., & Fu, X. (2025). DynImpt: A Dynamic Data Selection Method for Improving Model Training Efficiency. *EEE Transactions on Knowledge and Data Engineering*, 37(1(<https://doi.org/10.1109/TKDE.2024.3482466>), 239-252.
- Jiménez-Montanés, J., & Holgado-Terriza, J. (2018). Diseño y construcción de un guante de datos para el . *Enseñanza y Aprendizaje de Ingeniería de Computadores.*, 8, 197-213.
- La Valle, C., Shen, L., Shih, W., Kasari, C., Shire, S., Lord, C., & Tager-Flusberg, H. (2024). Does Gestural Communication Influence Later Spoken Language Ability in Minimally Verbal Autistic Children? *Journal of speech, language, and hearing research : JSLHR*, 67(7), 2283-2296. https://doi.org/https://doi.org/10.1044/2024_JSLHR-23-00433
- Lee, H., Jiang, M., Yang, J. Y., & Zhao, Q. (2025). Decoding Gestures in Electromyography: Spatiotemporal Graph Neural Networks for Generalizable and Interpretable Classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, (33), 404-419. <https://doi.org/https://doi.org/10.1109/TNSRE.2024.3523943>
- Tian, Y., Zhu, J., Yao, H., & Chen, D. (2024). Facial Expression Recognition Based on Vision Transformer with Hybrid Local Attention. *Applied Sciences*. <https://doi.org/https://doi.org/10.3390/app14156471>
- Villa, A. R. (2021). El lenguaje de señas en un mundo globalizado. *Blog Digit. Univ*, 1-12.
- Yang, B., Cheng, J., Qu, X., Song, Y., Yang, L. S., Bai, Z., & Ji, L. (2024). riboelectric-Inertial Sensing Glove Enhanced by Charge-Retained Strategy for Human-Machine Interaction. *Advanced Science*,, 12. <https://doi.org/https://doi.org/10.1002/advs.202408689>
- Zhang, W., Teng, F., & Zhang, Z. (2024). Noise Reduction of High-G Accelerometer Signals Based on Frequency-Domain Segmentation and Time-Domain Zeroing. *IEEE Transactions on Instrumentation and Measurement*, 73, 1-10. <https://ieeexplore.ieee.org/document/10571973>
-