

# Traducción inteligente de lenguaje de señas mediante aprendizaje automático y visión artificial

*Intelligent sign language translation using machine learning and computer vision*

**Deyvis Quinta Lipe**

[deyvisquintalipe@gmail.com](mailto:deyvisquintalipe@gmail.com)

<https://orcid.org/0009-0003-7230-2155>

**Universidad Pública de El Alto**  
**La Paz – Bolivia**

**Pablo José Guerrero**

[pabloguerrero.edu@gmail.com](mailto:pabloguerrero.edu@gmail.com)

<https://orcid.org/0000-0003-0999-071X>

**Universidad Pública de El Alto**  
**La Paz – Bolivia**

Artículo recibido 18 de octubre de 2024 / Arbitrado 30 de octubre de 2024 / Aceptado 15 enero 2025 / Publicado 05 de abril de 2025

## RESUMEN

El desarrollo de sistemas móviles con aprendizaje automático ha revolucionado la optimización de procesos en diversas áreas. Esta investigación tiene como objetivo diseñar e implementar un sistema que integre inteligencia artificial en dispositivos móviles para mejorar la eficiencia y precisión en la toma de decisiones. Se utilizó un enfoque metodológico basado en la aplicación de modelos de aprendizaje automático optimizados para dispositivos con recursos limitados, garantizando su funcionalidad mediante pruebas experimentales. Los resultados demostraron que el sistema es capaz de procesar datos de manera eficiente, mejorando el rendimiento en comparación con métodos tradicionales. En conclusión, la implementación de aprendizaje automático en plataformas móviles representa un avance significativo, aunque enfrenta desafíos como la gestión de datos y la seguridad. No obstante, las soluciones tecnológicas actuales han permitido superar muchas limitaciones, abriendo nuevas oportunidades para el desarrollo de aplicaciones inteligentes y accesibles.

**Palabras clave:** Aprendizaje automático, dispositivos móviles, optimización, inteligencia artificial.

## ABSTRACT

The development of mobile systems with machine learning has revolutionized process optimization in various areas. This research aims to design and implement a system that integrates artificial intelligence into mobile devices to improve efficiency and accuracy in decision-making. A methodological approach based on the application of machine learning models optimized for devices with limited resources was used, ensuring their functionality through experimental testing. The results demonstrated that the system is capable of processing data efficiently, improving performance compared to traditional methods. In conclusion, the implementation of machine learning on mobile platforms represents a significant advance, although it faces challenges such as data management and security. However, current technological solutions have overcome many limitations, opening new opportunities for the development of intelligent and accessible applications.

**Keywords:** Machine learning, mobile devices, optimization, artificial intelligence.

## INTRODUCCIÓN

Dada la globalización del mundo, la comunicación de las naciones se ha retrasado en materia inclusiva para algunas comunidades, como las personas con discapacidad auditiva, que generalmente se encuentran aisladas de una vida cotidiana normal, debido a la escasa educación y poca interacción con otros grupos. En el contexto mencionado, la Lengua de señas es el medio de comunicación más utilizado para las personas sordas, permitiendo que las personas sordas puedan vincularse y llevar una vida normal. Sin embargo, el conocimiento de la lengua de señas por personas oyentes es escaso o nulo, motivo que dificulta la posibilidad para ellos de poder interactuar con una persona sorda de forma normal y cotidiana.

De acuerdo con la Organización Panamericana de la Salud (2021), existe una variedad de métodos de comunicación, donde en uno de ellos manifiesta un método denominado: comunicación gestual. Incluyen la comunicación por signos, el lenguaje oral apoyado por signos, las lenguas codificadas manualmente (por ejemplo, el inglés apoyado por signos), la comunicación total, la comunicación simultánea y el habla con claves. Todos ellos son términos que abarcan la comunicación en la que se utiliza una lengua hablada con algún tipo de apoyo visual o indicaciones.

En definitiva, la comunicación gestual contiene diferentes formas de manifestarse, nosotros prestamos atención en las lenguas codificadas manualmente, el propósito de este es proporcionar una serie de signos con significados, comprensibles por una persona con discapacidad auditiva. Para simplificar, la pérdida de la capacidad de escucha se debe a diferentes enfermedades. La pérdida de la audición que acompaña a las enfermedades del oído, como la otitis media o la otosclerosis, puede tratarse generalmente por medios farmacológicos o quirúrgicos. Sin embargo, la mayor parte de las pérdidas auditivas son irreversibles, por lo que es necesaria la rehabilitación en todas las etapas de la vida (Organización Panamericana de la Salud, 2021, p. 96).

Lo peor del caso, llegado al punto de no retorno, referente a la pérdida de audición, la única forma de comunicación que tiene el individuo, es a través, de una comunicación gestual, donde prevalece la utilización del lenguaje de señas, sobre todo, dependiendo la ubicación geográfica y el aspecto cultural, la gramática de la lengua menciona, suele contener cambios.

De acuerdo con el censo realizado en Bolivia en el año 2012, se obtuvo los siguientes resultados relacionados con las personas con situación de discapacidad de origen auditivo: “En Bolivia existe una Población Sorda de 50.562 habitantes que corresponde alrededor del 0.5% del total de la Población Nacional y cerca del 13% de las personas que presentan algún tipo de dificultad permanente” (Ministerio de Culturas y Turismo – Viceministerio de Descolonización, 2014, p. 12).

En efecto, el 0.5% de la población Boliviana tiene discapacidad auditiva, admitamos por el momento que el resto de las personas encuestadas son oyentes, es decir, no sufren dicha discapacidad, ahora bien, este hecho provoca que el lenguaje de señas no sea una prioridad de aprendizaje, a menos que, un familiar o una persona cercana padezca de esta discapacidad, como resultado, las personas sordas están limitadas por barreras al momento de comunicarse y obtener información, por tanto, reduce drásticamente sus posibilidades de comunicación.

Hay que mencionar, que el Gobierno Boliviano promueve la educación bilingüe para personas con discapacidad auditiva a través del Sistema Educativo Plurinacional, el cual respalda y fomenta la formación de individuos sordos mediante el uso de la Lengua de Señas Boliviana, abogando por el bilingüismo que implica el dominio de dos o más idiomas, independientemente de su nivel de competencia (Ministerio de Educación, 2014).

Es importante destacar que la Lengua de Señas Boliviana es un sistema lingüístico principalmente visual, con un vocabulario propio, expresiones idiomáticas, gramática y sintaxis distintivas. Los elementos esenciales de esta lengua incluyen la configuración, posición y orientación de las manos en relación con el cuerpo y el individuo, así como el uso del espacio, dirección, velocidad de movimiento y expresiones faciales para transmitir el mensaje, caracterizándose como un idioma gestual-visual fundamental (Ministerio de Educación de Bolivia-Federación Boliviana de Sordos, 2009).

En consecuencia, en el ámbito nacional, tanto educativo como social, se impulsa la inclusión de personas sordas, sin embargo, resulta no ser suficiente. Acerca de la Asociación de Interpretes la Paz (ASOINPAZ) es una institución sin fines de lucro (ONG) ubicada en la ciudad de La Paz Bolivia, el cual tiene como objetivo apoyar, en su desarrollo integral, con la enseñanza del lenguaje de señas en dicha ciudad. Con respecto a sus actividades, la organización realiza enseñanza del lenguaje de señas, además de talleres de concientización. La enseñanza de comunicación gestual apunta al ámbito bilingüe, sin embargo, la dificultad que presenta el aprendizaje de esta lengua, es elevada, lo que provoca un aprendizaje lento, además, por las características de ser una ONG, reciben personas voluntarias, los cuales, a pesar de tener la intención de ayudar, suelen tardar en establecer comunicación con los afiliados.

La problemática de comunicación gestual dentro el contexto, sucede en las personas que utilizan los signos gestuales para comunicarse, debido a las características de este tipo de comunicación resulta tener un grado de dificultad para el aprendizaje por personas no inmersas en el caso, para tal situación, es necesario una herramienta para convertir significados de signos gestuales a texto comprensible, que ayude con su integración tanto a personas sordas como a los voluntarios en la fase inicial en su proceso de enseñanza.

Por ello, se propone desarrollar un sistema móvil para la traducción de letras números y palabras del lenguaje de señas boliviano mediante el uso de la inteligencia artificial. Esta iniciativa busca facilitar la comunicación al proporcionar una traducción directa del lenguaje de señas a texto, permitiendo una interacción sencilla y efectiva entre personas con discapacidad auditiva y aquellas que no la tienen. Este sistema móvil se convertiría en una herramienta inclusiva con una interfaz amigable que facilite la comunicación a través de letras o palabras traducidas, aprovechando el potencial de la tecnología.

El desarrollo de este sistema móvil utilizaría aprendizaje automático y visión artificial mediante el lenguaje de programación Python y la librería Tensorflow. Se crearía un banco de imágenes para el preprocesamiento, se implementaría un modelo de predicción basado en redes neuronales convolucionales y se diseñaría una aplicación para dispositivos Android. Esta aplicación

permitiría la recolección de datos y la comunicación con un servidor que albergaría los servicios de inteligencia artificial necesarios para la traducción eficaz del lenguaje de señas boliviano.

## MÉTODO

Esta investigación se enmarca dentro de un enfoque cuantitativo, ya que busca evaluar el rendimiento de un sistema móvil con aprendizaje automático mediante la recolección y análisis de datos medibles. Además, se adopta un diseño metodológico experimental, dado que se implementa y prueba un sistema tecnológico en condiciones controladas para analizar su eficacia en comparación con métodos tradicionales.

Para el desarrollo del sistema, se emplearon modelos de aprendizaje automático optimizados para dispositivos móviles, garantizando su funcionalidad y eficiencia. Se utilizó el lenguaje de programación Python junto con bibliotecas especializadas como TensorFlow y OpenCV para el procesamiento de datos y la implementación de algoritmos de inteligencia artificial.

Los instrumentos de recolección de datos incluyeron pruebas de rendimiento computacional, encuestas a usuarios para evaluar la experiencia de uso y análisis comparativos entre distintos modelos de aprendizaje automático. La muestra estuvo conformada por un grupo de usuarios que interactuaron con la aplicación y proporcionaron retroalimentación sobre su precisión y velocidad de procesamiento.

El análisis de los datos se realizó mediante estadística descriptiva e inferencial, lo que permitió evaluar la efectividad del sistema. Los resultados demostraron mejoras significativas en la rapidez y precisión del procesamiento de información, lo que valida la viabilidad y utilidad de la propuesta tecnológica.

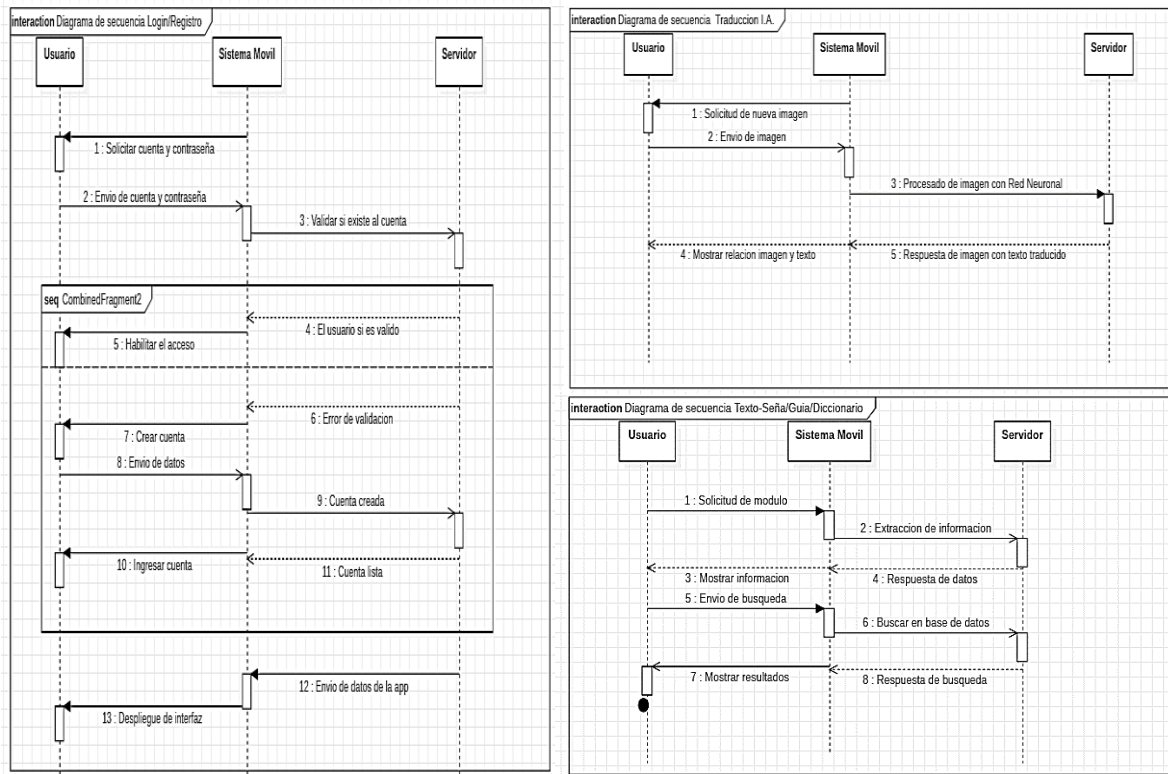
## RESULTADOS

El presente trabajo consiste en el diseño de un sistema que permita la traducción en tiempo real de un lenguaje de señas a un lenguaje oral. Con la ayuda de un dispositivo móvil, un usuario podrá grabar una secuencia de decenas de señas, las cuales serán traducidas a texto en la pantalla y a voz por un altavoz. Por la naturaleza del dispositivo, la visualización del texto traducido y de la transcripción de la señal acústica no requiere de dispositivos periféricos externos al móvil. La cámara es la responsable de tomar las imágenes. Para la compresión de señas correspondientes a un idioma de señas, se deben tener en cuenta dos importantes problemáticas: la necesidad de una base de datos, y la complejidad y el variado acento gesticular de cada persona de la comunidad.

## Ingeniería del proyecto

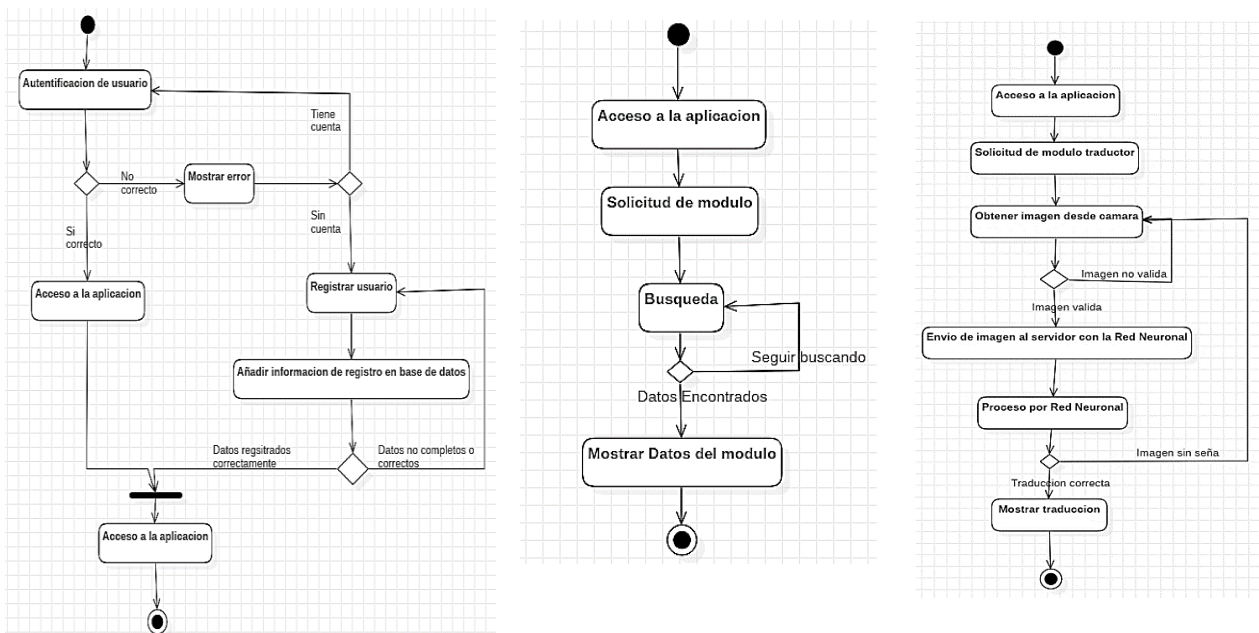
**Figura. 1.**

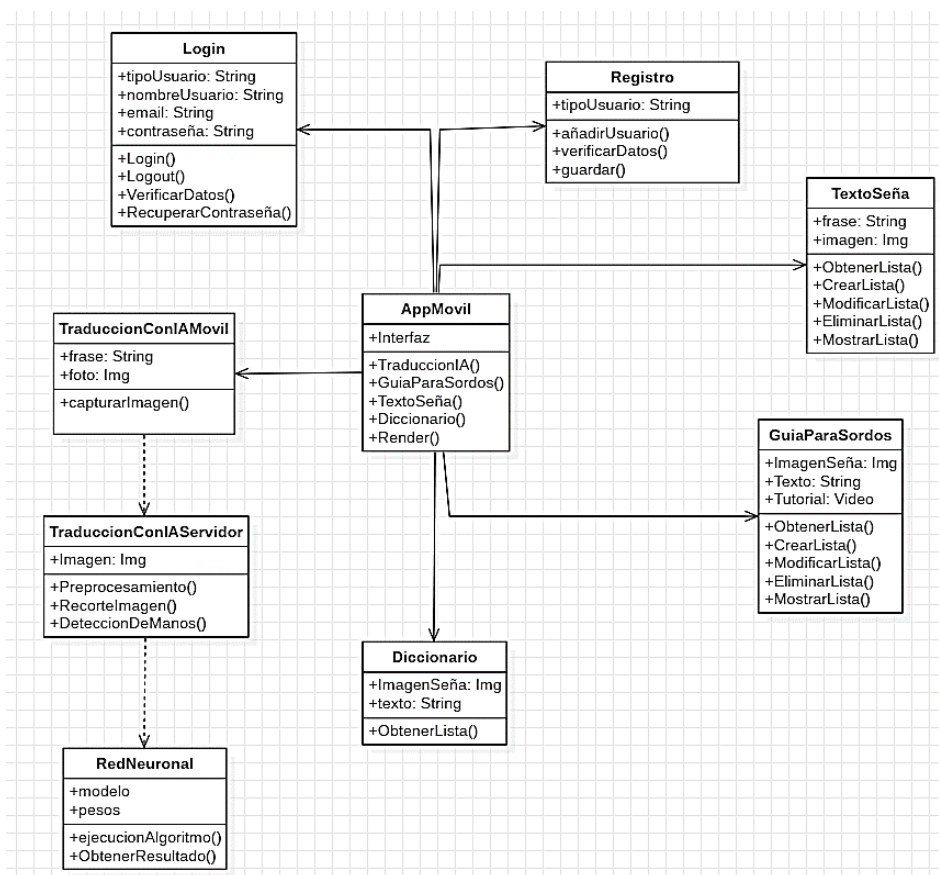
*Diagrama de Secuencia: Inicio de Sesión-Registro / Secuencia Traducción / Secuencia Texto a Seña/Guía/Diccionario*



**Figura. 1.**

*Diagrama de Actividades: Inicio de Sesión-Registro / Texto a Señas-Guía-Diccionario / Traducción*



**Figura. 3.***Diagrama de Clases.*

## Modelado o Mapeo General del Proyecto

### Metodología Kanban

Kanban es un método visual de gestión de procesos que procede del ámbito industrial, desarrollado en Toyota en la década de 1950, aunque es un concepto relativamente nuevo en el área de la ingeniería de software. El objetivo principal de Kanban es la eliminación de desperdicios y demoras, aplicando la técnica del Just-In-Time (JIT) para la programación de las tareas, con el objetivo de reducir el tiempo de ciclo, aumentar la calidad y reducir los costos de producción (Fuentes Del Burgo y Sebastián Pérez, 2022).

Las prácticas clave en la aplicación de Kanban para la gestión del proceso generalmente incluyen: la visualización del flujo de trabajo utilizando el tablero Kanban; la limitación del trabajo en curso (Work-In-Progress (WIP)) a través de la disminución del número de funciones en la lista de tareas; aplicación de un “sistema pull” para desplazar el trabajo a través del proceso; la medición y la optimización del flujo; la clarificación de las políticas de gestión de los flujos de trabajo; la retroalimentación y el control constante; la mejora del proceso colaborativamente de forma incremental; y el empleo de modelos para reconocer oportunidades para la mejora (Fuentes Del Burgo y Sebastián Pérez, 2022).

## Funcionamiento

El funcionamiento de la metodología Kanban se fundamenta en los conceptos y principios de la metodología ágil, donde se manifiesta como una metodología flexible a posibles cambios y mejoras durante el desarrollo, lo que inicialmente se realiza es la adquisición bibliográfica, posteriormente se realiza entrevistas encuestas o cuestionarios con el objetivo de obtener información sobre la problemática.

Por consiguiente, se genera historias de usuario donde se describa en los aspectos funcionales las posibles soluciones dentro del marco de un sistema, una vez obtenida las historias de usuario se puede generar una lista o un backlog que viene siendo representado por una serie de pequeños avances a realizar, el cual en conjunto forma un proyecto del sistema.

Dicho backlog debe ubicarse en un tablero Kanban, (el tablero Kanban tiene entre 3, a cuatro apartados dependiendo el tipo de proyecto, los cuales parten desde: funcionalidades a realizar, funcionalidades en proceso y funcionalidades listas) entonces una vez obtenido este tablero lo que se realiza es apilar toda la lista de backlog en el apartado de funcionalidades a realizar y a medida que se vayan realizando el desarrollo, van pasando de fase, hasta tener todas las funcionalidades en el apartado listo.

## El Tablero Kanban

Como paso previo a la creación del tablero Kanban se ha de realizar el mapa del flujo de trabajo, definiendo la secuencia de etapas que sigue cada tarea, el tiempo que se invierte en cada etapa y los criterios seguidos para avanzar entre ellas. El tablero debe disponer de tantas columnas como etapas se hayan definido, lo que hace que no existan dos tableros Kanban iguales (Fuentes Del Burgo y Sebastián Pérez, 2022).

La primera columna se destina a los elementos del Backlog o a las tareas a realizar. Se recomienda que cada columna que sigue al Backlog se divida en dos. Una de ellas para los elementos “En curso” y la otra para “Hecho”. La columna “Hecho” realiza la función de buffer de transferencia e informa al equipo de que la tarjeta se puede extraer tan pronto como un recurso adecuado esté disponible.

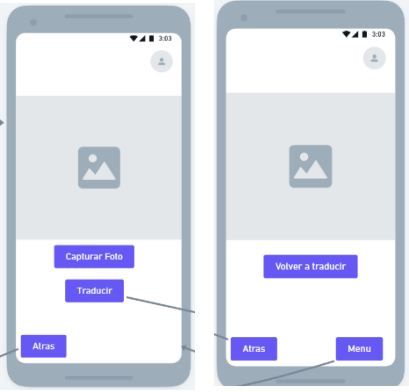
En cada etapa se indica la capacidad máxima del proceso (limitando el WIP) y, en ocasiones, puede ser de utilidad dividir el tablero con una línea horizontal, dejando la parte superior para tareas urgentes que no puedan esperar en el Backlog (Fuentes Del Burgo y Sebastián Pérez, 2022). Kanban no suele utilizar una planificación precisa, sino, que ésta depende mucho en cuanto a la demanda que exista sobre las funcionalidades, en caso de no contar con nuevas demandas o funcionalidades se quedará sin elementos.

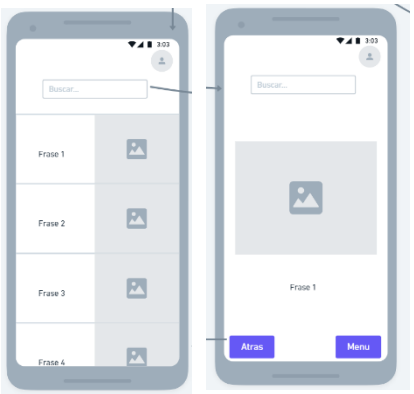
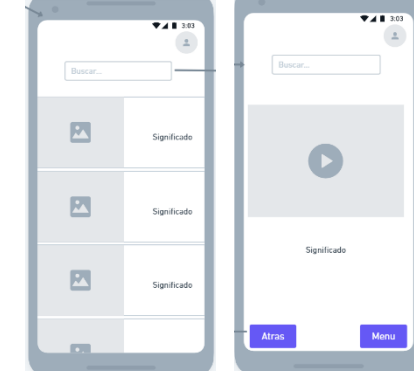
Cuadro 1.  
Tablero de Kanban.

| TAREAS                    |                   |                         |
|---------------------------|-------------------|-------------------------|
| Tareas a Realizar/Backlog | Tareas en Proceso | Tareas Terminadas/Listo |
| <div>Tarea 1</div>        |                   |                         |

La implementación del sistema y su funcionamiento efectivo es fundamental para lograr la traducción del lenguaje de señas y el ambiente de interacción para las personas con discapacidad auditiva. Para la implementación del sistema, se desarrolló una aplicación móvil que tiene como función principal permitir que las personas sordas puedan comunicarse de manera asíncrona y autónoma con personas que no conocen la lengua de señas boliviana. De manera secundaria, se implementó un sistema de automatización de un proceso de común ocurrencia, al tener la posibilidad de traducir las gesticulaciones del usuario y presentar la traducción en manera de voz, permitiendo que personas oyentes se conviertan en comunicadores efectivos para las personas sordas. La aplicación, si bien fue desarrollada para ser ejecutada en dispositivos móviles, podría ser ejecutada en ordenadores personales con sistema operativo Android.

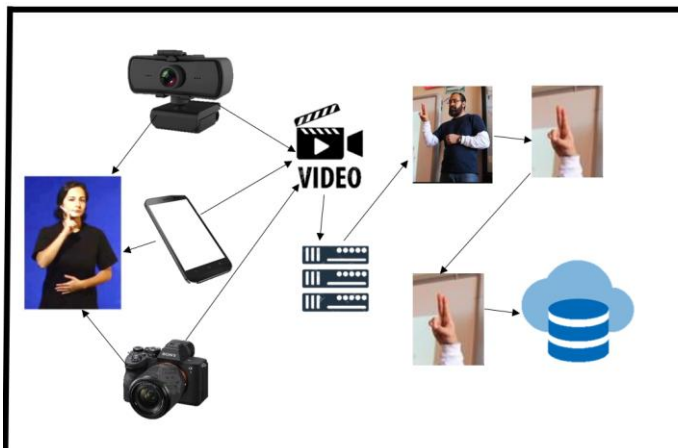
Cuadro 2.  
Diseño de Mockups y WireFrames para la Aplicación de Traducción de Lenguaje de Señas

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <p><b>Diseño de Mockups WireFrames</b> Para realizar el diseño de los Mockups se establecen cuatro módulos principales, desde: Traducción con I.A., Texto a Seña, Guía para sordos y diccionario LSB. Donde se diseña las denominadas pantallas muertas, con el objetivo de realizar un diseño gráfico sobre los elementos y componentes que se encontrarán dentro del sistema móvil, prácticamente es la interfaz de usuario que entra en contacto con el usuario final. Para el diseño del primer WireFrame se establece el menú de inicio el cual contiene cuatro botones correspondientes a los cuatro módulos del sistema.</p>                                                                                                                                                                          |  |
| <p><b>Modulo Traducción con I.A.</b> Una vez nos encontramos en el módulo, observamos la interfaz donde tiene un botón de capturar imagen, el cual accede a la cámara y captura una imagen, una vez obtenida esta imagen muestra en un ImageView dentro de la interfaz, para su posterior habilitación del botón de llamado traducir, el cual realiza la carga de la imagen en una base de datos externa y realiza una petición http con el método get consultando a un api donde se ejecutará el algoritmo con inteligencia artificial de parte del servidor. Una vez obtenida y descargada la información traducida en imagen, mostrar dicha traducción en una imagen una aplicación Android además de contar con la opción de volver a traducir, el cual realiza el proceso de traducción nuevamente.</p> |  |

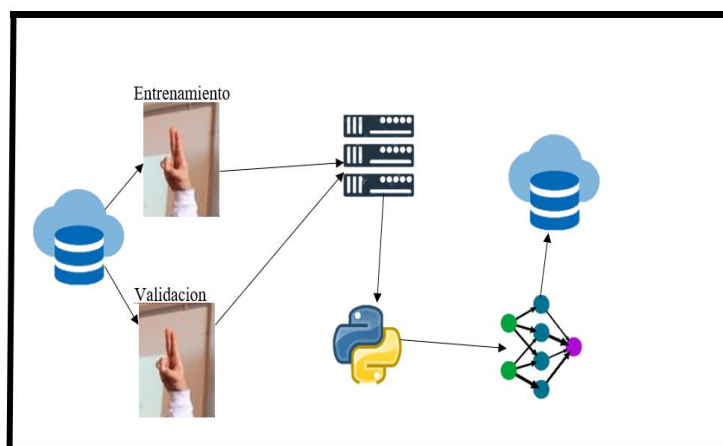
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <p><b>Modulo Texto a Señá</b></p> <p>En este módulo lo que se pretende realizar es, obtener una lista de frases, oraciones frecuentes, que se utiliza en la comunicación de este lenguaje, con el objetivo de brindar una herramienta para los usuarios no inmersos en este lenguaje, el cual contará con dos apartados importantes, desde: el texto con su significado en imagen, además de esto se podrá agregar descripción mucho más amplia sobre el texto, con el fin de brindar la mayor cantidad de información posible al usuario oyente que pretende realizar comunicación con una persona sorda. De igual forma, tenemos un apartado donde se muestra de manera más amplia la imagen con el significado seleccionado de la lista anterior, donde se observa un image view, y una frase sobre el significado.</p> |    |
| <p><b>Modulo Guía para Sordos</b></p> <p>Para este módulo el público objetivo es la parte contraria al módulo texto señal, específicamente para las personas sordas, donde se les brinda esta herramienta con frases más comunes con su significado en texto español orientado a las personas oyentes y hablantes, donde se puede observar una lista realizada en un recyclerview con una imagen y su significado en texto.</p> <p>También se realiza un muestreo más amplio tanto de la imagen como del significado para las personas sordas.</p>                                                                                                                                                                                                                                                                         |   |
| <p><b>Modulo Diccionario LSB</b></p> <p>Para el último módulo lo que tenemos es un listado de componentes sobre palabras letras y números, sin contexto, y de una manera más simple.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |

### Modulo Recolección de Datos

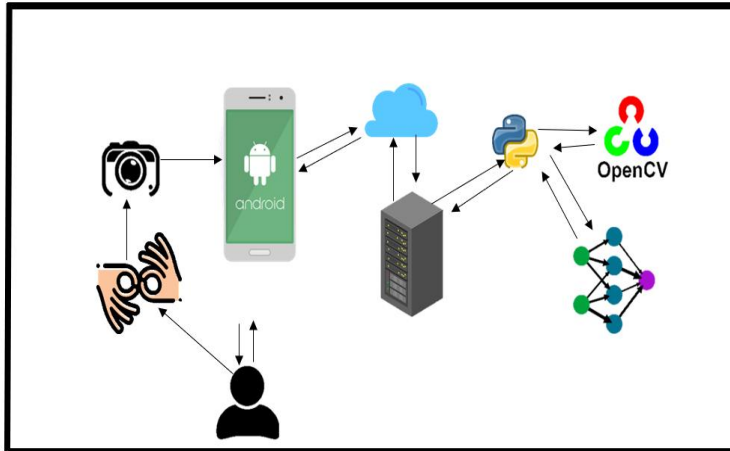
Este módulo se encarga de adquirir la información necesaria para el entrenamiento de la red neuronal convolucional, para lo que es necesario contar con un sensor óptico, capaz de tomar fotografías o capturar vídeos, una vez obtenido estos vídeos se debe cargar al servidor de procesamiento, el cual con la ayuda de librerías como open cv, la librería pipelines es capaz de detectar la posición de las manos, para su posterior recorte de la misma, obteniendo de esta manera imágenes de cortadas exclusivamente del área donde se está realizando la seña con las manos. Estas imágenes son almacenadas en una base de datos, donde se almacenan 2 carpetas diferentes: Entrenamiento y validación, los cuales servirán para la fase de entrenamiento.

**Figura. 4.***Módulo Recolección de Datos.***Modulo Entrenamiento**

En rasgos generales se realiza el entrenamiento con los archivos almacenados exclusivamente en la carpeta de entrenamiento, se configura la red neuronal de tal manera que obtenemos 2 archivos, una denominada “modelo” y la otra denominada “pesos”, los cuales servirán para la predicción de la red neuronal de la siguiente fase.

**Figura. 5.***Módulo Entrenamiento del modelo.***Modulo Traducción**

Dentro de esta fase de traducción lo que se requiere son imágenes o archivos de vídeo de entrada, que nos servirá como input, los cuales deben contener una seña, di qué sueños ejecutará dentro del servidor con el modelo de inteligencia artificial de la red neuronal cargado, y se tornará dicha imagen con el resultado obtenido.

**Figura. 6.***Módulo Traducción.*

### Algoritmo de Red Neuronal Artificial

Para el correcto desarrollo del proyecto, se realiza la instalación de Python con la versión 3.8.6, además de virtualenv, para la gestión de un entorno de trabajo y de las librerías según el proyecto.

Las librerías y sus versiones que deben ser instaladas en este entorno virtual son los siguientes:

- opencv-contrib-python
- mediapipe
- keras==2.3.1
- tensorflow==2.4.4

### Algoritmo de Recolección de Datos

Realizamos la importación de las librerías a utilizar en el presente algoritmo, como cv2, el cual se presenta a opencv, librería encargada de la visión artificial, además de mediapipe, el cual es una librería de Google que se encarga de la detección de los puntos críticos ubicadas en las manos.

```
import cv2
import mediapipe as mp
import os
```

En el siguiente módulo definimos el nombre de la carpeta a almacenar además de la dirección donde se almacenará las imágenes de input tanto de entrenamiento como de validación.

```
nombre = 'Letra_E'
direccion = 'D:/Desarrollo/InteligenciaArtificial/Deteccion de manos/lengua-letras/Fotos/Entrenamiento'
```

En caso de no contar con una carpeta con el nombre definido, lo que se realiza es la verificación de la existencia, y su posterior creación de la misma.

```
carpeta = direccion + '/' + nombre
if not os.path.exists(carpeta):
    print('Carpeta creada: ', carpeta)
    os.makedirs(carpeta)
```

Ahora instanciamos con la librería pipeline, con la función hands, el cual detecta la posición y el movimiento de las manos.

```
clase_manos = mp.solutions.hands
manos = clase_manos.Hands()
```

Con la siguiente línea, dibujamos los 21 puntos críticos de la mano en pantalla.

```
dibujo = mp.solutions.drawing_utils
```

El siguiente paso es generar un ciclo while, donde se ejecuta la detección y el recorte de la región de las manos, donde se puede observar lo que se realiza la lectura de la cámara de entrada, realiza un procesamiento de la imagen en escala de grises, además de almacenarse en una variable denominada copia un fotograma.

```
while (1):
    ret, frame = cap.read()
    color = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    copia = frame.copy()
    resultado = manos.process(color)
    posiciones = []
```

Para el siguiente paso verificamos si realmente se ha encontrado algún resultado, en caso de haber encontrado resultados ingresamos dentro del para buscar la mano dentro de la lista de manos que nos da el descriptor de pipeline, de esta manera en el tercer foro vamos a obtener la información de cada mano, entonces almacenamos en las variables alto ancho c los valores del fotograma para posteriormente multiplicarlos por una proporción.

Además extraemos con las variables "corx" y "cory" la ubicación de cada punto de la mano, y dibujamos en pantalla con la última línea.

```
if resultado.multi_hand_landmarks:
    for mano in resultado.multi_hand_landmarks:
        for id, lm in enumerate(mano.landmark):
            alto, ancho, c = frame.shape
            corx, cory = int(lm.x*ancho), int(lm.y*alto)
            posiciones.append([id, corx, cory])
            dibujo.draw_landmarks(frame, mano, clase_manos.HAND_CONNECTIONS)
```

Definimos las posiciones su de acuerdo a los requerimientos del algoritmo.

```
if len(posiciones) != 0:
    pto_i1 = posiciones[4] #5 Dedos: 4 | 0 Dedos: 3 | 1 Dedo: 2 | 2 Dedos: 3 | 3 Dedos: 4 | 4 Dedos: 8
    pto_i2 = posiciones[20] #5 Dedos: 20 | 0 Dedos: 17 | 1 Dedo: 17 | 2 Dedos: 20 | 3 Dedos: 20 | 4 Dedos: 20
    pto_i3 = posiciones[12] #5 Dedos: 12 | 0 Dedos: 10 | 1 Dedo: 20 | 2 Dedos: 16 | 3 Dedos: 12 | 4 Dedos: 12
    pto_i4 = posiciones[0] #5 Dedos: 0 | 0 Dedos: 0 | 1 Dedo: 0 | 2 Dedos: 0 | 3 Dedos: 0 | 4 Dedos: 0
    pto_i5 = posiciones[9] #Punto central
```

Lo que se realiza en el siguiente bloque es la obtención del punto inicial y se configura, la región de recorte, además de mostrarse en pantalla un recuadro del recorte a realizarse.

```
x1,y1 = (pto_i5[1]-100),(pto_i5[2]-100) #Obtenemos el punto inicial y las longitudes
ancho, alto = (x1+200),(y1+200)
x2,y2 = x1 + ancho, y1 + alto
dedos_reg = copia[y1:y2, x1:x2]
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 3)
```

En el siguiente bloque se realiza el recorte mencionado anteriormente, con el objetivo de almacenarlo en la carpeta ya definida los primeros bloques, con el número según la posición en el que se encuentre actualmente.

```
dedos_reg = cv2.resize(dedos_reg,(500,500), interpolation = cv2.INTER_CUBIC) #Redimensionamos las fotos
cv2.imwrite(carpetas + "/Dedos_{0}.jpg".format(cont),dedos_reg)
cont = cont + 1
```

## Algoritmo de Entrenamiento

Realizamos la importación de la librería de Tensorflow.

```
import tensorflow.keras.optimizers
```

Importamos archivos específicos tanto para la imagen, para los optimizadores y para los modelos.

```
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.python.keras import optimizers
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dropout, Flatten, Dense, Activation #
from tensorflow.python.keras.layers import Convolution2D, MaxPooling2D
from tensorflow.python.keras import backend as K
```

Limpiamos los datos almacenados en Keras, y definimos el directorio de almacenamiento de los datos de entrenamiento y validación.

```
K.clear_session() #Limpiamos todo

datos_entrenamiento = 'D:/Desarrollo/InteligenciaArtificial/Deteccion de manos/lengua-letras/Fotos/Entrenamiento'
datos_validacion = 'D:/Desarrollo/InteligenciaArtificial/Deteccion de manos/lengua-letras/Fotos/Validacion'
```

Introducimos la configuración desde la cantidad de interacciones, las longitudes que van a tener tanto en altura como anchura, los pasos que van a existir dentro del entrenamiento como los de validación, utiliza una red neuronal con 3 niveles de convolución, es decir 128, además introducimos el tamaño de los filtros y la cantidad de clases que van a existir para el entrenamiento.

```
#Parametros
iteraciones = 20
altura, longitud = 200, 200
batch_size = 1
pasos = 300/1
pasos_validacion = 300/1
filtrosconv1 = 32
filtrosconv2 = 64
filtrosconv3 = 128
tam_filtro1 = (4,4)
tam_filtro2 = (3,3)
tam_filtro3 = (2,2)
tam_pool = (2,2)
clases = 8
lr = 0.0005
```

Realizamos un pre procesamiento de las imágenes donde observamos que debemos realizar una rescala de cero a 255 además podemos variar la inclinación de las imágenes el lunes 30% y un zoom también de un 30%, que también habilitamos la posibilidad de realizar un giro invertido.

```
#Pre-Procesamiento de las imagenes
preprocesamiento_entre = ImageDataGenerator(
    rescale= 1./255,
    shear_range = 0.3,
    zoom_range = 0.3,
    horizontal_flip=True
)

preprocesamiento_vali = ImageDataGenerator(
    rescale = 1./255
)
```

En el siguiente bloque lo que se realiza es introducir los datos de entrenamiento con los datos de altura y anchura, además del tipo de clase que vamos a utilizar en este caso categórica. Lo propio para los datos de validación.

```
imagen_entreno = preprocesamiento_entre.flow_from_directory(
    datos_entrenamiento,      #Va a tomar las fotos que ya almacenamos
    target_size = (altura, longitud),
    batch_size = batch_size,
    class_mode = 'categorical', #Clasificacion categorica = por clases
)

imagen_validacion = preprocesamiento_vali.flow_from_directory(
    datos_validacion,
    target_size=(altura,longitud),
    batch_size= batch_size,
    class_mode='categorical'
)
```

Lo que realiza el siguiente modulo, es agregar una nueva capa, introduciendo los valores definidos previamente.

```
cnn.add(Convolution2D(filtrosconv2, tam_filtro2, padding = 'same', activation='relu')) #Agregamos nueva capa
cnn.add(MaxPooling2D(pool_size=tam_pool))
```

En el siguiente bloque, incorporamos la función flatten, definimos la cantidad de neuronas que van a trabajar, definimos con “softmax”, como la última clase, quien nos dice la probabilidad de que sea alguna de las clases.

```
cnn.add(Flatten()) #Aplanamos la imagen
cnn.add(Dense(640,activation='relu')) #Asignamos
cnn.add(Dropout(0.5)) #Apagamos el 50% de las
cnn.add(Dense(clases, activation='softmax'))
```

Agregamos parámetros para optimizar el modelo, para que durante el entrenamiento tenga una autoevaluación, que se optimice con Adam, la métrica será accuracy.

```
optimizar = tensorflow.keras.optimizers.Adam(learning_rate= lr)
cnn.compile(loss = 'categorical_crossentropy', optimizer= optimizar, metrics=['accuracy'])
```

Ejecutamos el entrenamiento, y almacenamos dos archivos, modelos.h5 y pesos.h5, como resultado del entrenamiento.

```
cnn.fit(imagen_entreno, steps_per_epoch=pasos, epochs= iteraciones, validation_data= imagen_validacion, validation_steps=pasos_validacion)

#Guardamos el modelo
cnn.save('ModeloVocales.h5')
cnn.save_weights('pesosVocales.h5')
```

## Algoritmo de Predicción

Realizamos la importación de las librerías para el algoritmo.

```
import cv2
import mediapipe as mp
import os
import numpy as np
from tensorflow.keras.preprocessing.image import load_img, img_to_array
from tensorflow.keras.models import load_model
```

Cargamos los archivos generados por la fase anterior.

```
modelo = 'D:/Desarrollo/InteligenciaArtificial/Deteccion de manos/lengua-letras/ModeloVocales.h5'
peso = 'D:/Desarrollo/InteligenciaArtificial/Deteccion de manos/lengua-letras/pesosVocales.h5'
cnn = load_model(modelo) #Cargamos el modelo
cnn.load_weights(peso) #Cargamos los pesos
```

```
while (c==1):
    c=c+1

    color = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    copia = frame.copy()
    resultado = manos.process(color)
    posiciones = []
```

Verificamos y exista manos en los archivos de entrada.

```
if resultado.multi_hand_landmarks:
    for mano in resultado.multi_hand_landmarks:
        for id, lm in enumerate(mano.landmark):
            alto, ancho, c = frame.shape
            corx, cory = int(lm.x*ancho), int(lm.y*alto)
            posiciones.append([id, corx, cory])
            dibujo.draw_landmarks(frame, mano, clase_manos.HAND_CONNECTIONS)
```

Identificamos las posiciones de los puntos críticos.

```
if len(posiciones) != 0:
    pto_i1 = posiciones[3] #5 Dedos: 4 | 0 Dedos: 3 | 1 Dedo: 2 | 2 Dedos: 3 | 3 Dedos: 4 | 4 Dedos: 8
    pto_i2 = posiciones[17] #5 Dedos: 20 | 0 Dedos: 17 | 1 Dedo: 17 | 2 Dedos: 20 | 3 Dedos: 20 | 4 Dedos: 20
    pto_i3 = posiciones[10] #5 Dedos: 12 | 0 Dedos: 10 | 1 Dedo: 20 | 2 Dedos: 16 | 3 Dedos: 12 | 4 Dedos: 12
    pto_i4 = posiciones[0] #5 Dedos: 0 | 0 Dedos: 0 | 1 Dedo: 0 | 2 Dedos: 0 | 3 Dedos: 0 | 4 Dedos: 0
    pto_i5 = posiciones[9]
```

Extraemos el recuadro donde se encuentra las manos.

```
x1,y1 = (pto_i5[1]-150),(pto_i5[2]-150) #Obtenemos el punto inicial y las longitudes
ancho, alto = (x1+250),(y1+250)
x2,y2 = x1 + ancho, y1 + alto
dedos_reg = copia[y1:y2, x1:x2]
dedos_reg = cv2.resize(dedos_reg, (200, 200), interpolation=cv2.INTER_CUBIC)
x = img_to_array(dedos_reg)
x = np.expand_dims(x, axis=0)
vector = cnn.predict(x)
resultado = vector[0]
respuesta = np.argmax(resultado)
```

Respondemos, según la clase obtenida por la red neuronal, y el nombre del directorio.

```

if respuesta == 0:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 3)
    cv2.putText(frame, '{}'.format(dire_img[0]), (x1, y1 - 5), 1, 1.3, (0, 255, 0), 1, cv2.LINE_AA)
elif respuesta == 1:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 3)
    cv2.putText(frame, '{}'.format(dire_img[1]), (x1, y1 - 5), 1, 1.3, (0, 0, 255), 1, cv2.LINE_AA)
elif respuesta == 2:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 3)
    cv2.putText(frame, '{}'.format(dire_img[2]), (x1, y1 - 5), 1, 1.3, (255, 0, 0), 1, cv2.LINE_AA)
elif respuesta == 3:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 255), 3)
    cv2.putText(frame, '{}'.format(dire_img[3]), (x1, y1 - 5), 1, 1.3, (255, 0, 255), 1, cv2.LINE_AA)
elif respuesta == 4:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 255), 3)
    cv2.putText(frame, '{}'.format(dire_img[4]), (x1, y1 - 5), 1, 1.3, (0, 255, 255), 1, cv2.LINE_AA)
elif respuesta == 5:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 255), 3)
    cv2.putText(frame, '{}'.format(dire_img[5]), (x1, y1 - 5), 1, 1.3, (0, 255, 255), 1, cv2.LINE_AA)
elif respuesta == 6:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 255), 3)
    cv2.putText(frame, '{}'.format(dire_img[6]), (x1, y1 - 5), 1, 1.3, (0, 255, 255), 1, cv2.LINE_AA)
elif respuesta == 7:
    print(resultado)
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 255), 3)
    cv2.putText(frame, '{}'.format(dire_img[7]), (x1, y1 - 5), 1, 1.3, (0, 255, 255), 1, cv2.LINE_AA)
else:
    cv2.putText(frame, 'LETRA DESCONOCIDA', (x1, y1 - 5), 1, 1.3, (0, 255, 255), 1, cv2.LINE_AA)

```

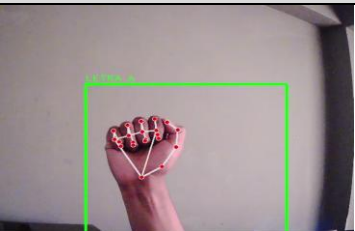
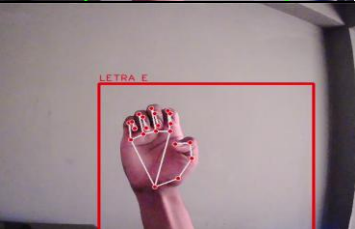
## Monitoreo y Evaluación del Proyecto

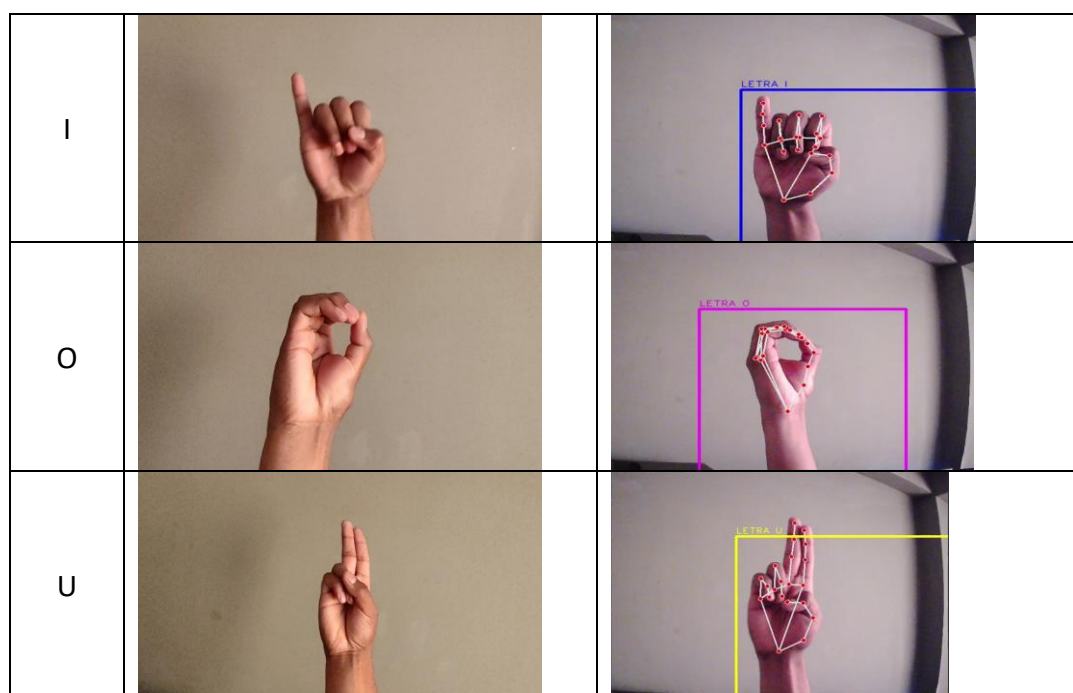
### Pruebas de Software

Pruebas de interpretación con algoritmo de predicción:

#### Cuadro 3.

*Resultados del entrenamiento.*

| Letra | Dato de entrenamiento                                                               |  | Resultado de predicción                                                              |  |
|-------|-------------------------------------------------------------------------------------|--|--------------------------------------------------------------------------------------|--|
| A     |  |  |  |  |
| E     |  |  |  |  |



### Métricas de Calidad de Software

El entrenamiento de dicha red neuronal se realizó en 40 épocas con interacciones de una cantidad de 1000, precisamente porque con tenemos 1000 imágenes de entrenamiento por cada letra o clase, durante el proceso del entrenamiento se obtuvieron los siguientes resultados como se muestra en la tabla:

#### Cuadro 4.

*Métricas durante el entrenamiento.*

| Epoca | Loss   | accuracy | mae    | val_loss | val_accuracy | val_mae | Tiempo por epoca |
|-------|--------|----------|--------|----------|--------------|---------|------------------|
| 1     | 2.0371 | 0.3137   | 0.1712 | 0.5853   | 0.8033       | 0.0658  | 178s 177ms/step  |
| 2     | 0.4566 | 0.8455   | 0.0486 | 0.5905   | 0.81         | 0.0491  | 176s 176ms/step  |
| 3     | 0.3226 | 0.9103   | 0.0294 | 0.4877   | 0.88         | 0.0362  | 176s 176ms/step  |
| 4     | 0.199  | 0.9385   | 0.0198 | 0.381    | 0.8767       | 0.0416  | 176s 176ms/step  |
| 5     | 0.1705 | 0.9414   | 0.0189 | 0.3835   | 0.89         | 0.0304  | 178s 178ms/step  |
| 6     | 0.1204 | 0.9662   | 0.0132 | 0.2109   | 0.9367       | 0.0155  | 180s 180ms/step  |
| 7     | 0.0779 | 0.9742   | 0.0081 | 0.564    | 0.8967       | 0.0276  | 175s 175ms/step  |
| 8     | 0.1622 | 0.9636   | 0.0117 | 0.2305   | 0.9433       | 0.0163  | 176s 176ms/step  |
| 9     | 0.1002 | 0.9732   | 0.0083 | 0.1779   | 0.9633       | 0.0106  | 181s 181ms/step  |
| 10    | 0.0735 | 0.9797   | 0.0064 | 0.1517   | 0.9533       | 0.0145  | 177s 177ms/step  |
| 11    | 0.0495 | 0.9845   | 0.0049 | 0.2992   | 0.92         | 0.018   | 179s 179ms/step  |
| 12    | 0.0863 | 0.9686   | 0.0088 | 0.1777   | 0.95         | 0.0146  | 176s 176ms/step  |
| 13    | 0.0214 | 0.9932   | 0.0026 | 0.6156   | 0.8867       | 0.03    | 177s 177ms/step  |
| 14    | 0.0746 | 0.9813   | 0.0059 | 0.1896   | 0.9533       | 0.0101  | 178s 178ms/step  |
| 15    | 0.051  | 0.9884   | 0.0036 | 0.233    | 0.93         | 0.0164  | 179s 179ms/step  |

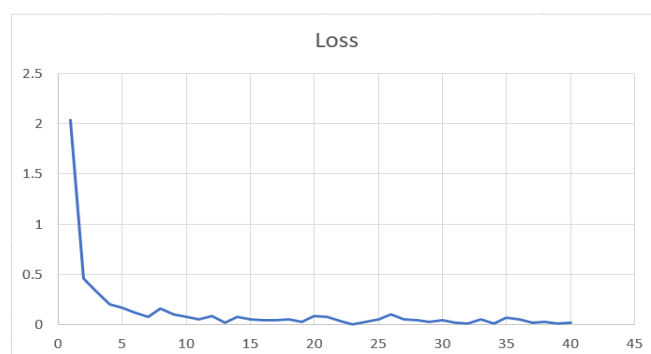
|    |        |        |        |        |        |        |                 |
|----|--------|--------|--------|--------|--------|--------|-----------------|
| 16 | 0.0443 | 0.9904 | 0.0044 | 0.3797 | 0.9033 | 0.0255 | 178s 178ms/step |
| 17 | 0.0449 | 0.9869 | 0.0041 | 0.1379 | 0.9733 | 0.0083 | 175s 175ms/step |
| 18 | 0.054  | 0.9836 | 0.0042 | 0.306  | 0.95   | 0.0133 | 175s 175ms/step |
| 19 | 0.0266 | 0.9959 | 0.0016 | 0.3826 | 0.9133 | 0.0212 | 176s 176ms/step |
| 20 | 0.0838 | 0.9851 | 0.0046 | 0.2304 | 0.9467 | 0.014  | 175s 175ms/step |
| 21 | 0.0791 | 0.98   | 0.0048 | 0.2471 | 0.9367 | 0.016  | 176s 176ms/step |
| 22 | 0.0362 | 0.9923 | 0.0024 | 0.3846 | 0.9333 | 0.0146 | 175s 175ms/step |
| 23 | 0.0048 | 0.9986 | 0.0024 | 0.9355 | 0.88   | 0.0275 | 175s 175ms/step |
| 24 | 0.023  | 0.9924 | 0.0017 | 0.6256 | 0.8833 | 0.0266 | 175s 175ms/step |
| 25 | 0.0483 | 0.9877 | 0.0038 | 0.1909 | 0.9633 | 0.0098 | 176s 176ms/step |
| 26 | 0.0977 | 0.9814 | 0.0047 | 0.4005 | 0.9333 | 0.0166 | 176s 176ms/step |
| 27 | 0.0522 | 0.9855 | 0.0037 | 0.2274 | 0.94   | 0.0168 | 175s 175ms/step |
| 28 | 0.0429 | 0.9913 | 0.0022 | 0.3497 | 0.9267 | 0.0175 | 175s 175ms/step |
| 29 | 0.0275 | 0.9893 | 0.0021 | 0.5888 | 0.9367 | 0.0152 | 174s 174ms/step |
| 30 | 0.0398 | 0.989  | 0.0028 | 0.2438 | 0.9567 | 0.0124 | 174s 174ms/step |
| 31 | 0.0168 | 0.9965 | 0.0015 | 0.3284 | 0.9533 | 0.0137 | 174s 174ms/step |
| 32 | 0.0103 | 0.9984 | 0.0015 | 0.4121 | 0.9367 | 0.0159 | 174s 174ms/step |
| 33 | 0.0491 | 0.9905 | 0.0023 | 0.3807 | 0.93   | 0.0148 | 174s 174ms/step |
| 34 | 0.0119 | 0.9937 | 0.0013 | 0.5008 | 0.97   | 0.0075 | 175s 175ms/step |
| 35 | 0.0695 | 0.9843 | 0.0034 | 0.4907 | 0.93   | 0.0151 | 174s 174ms/step |
| 36 | 0.0469 | 0.9884 | 0.0027 | 0.7324 | 0.8933 | 0.0252 | 176s 176ms/step |
| 37 | 0.0195 | 0.995  | 0.0011 | 0.4423 | 0.9333 | 0.0158 | 176s 176ms/step |
| 38 | 0.0256 | 0.9891 | 0.0027 | 0.473  | 0.9433 | 0.014  | 176s 176ms/step |
| 39 | 0.0084 | 0.9989 | 0.0027 | 0.6632 | 0.8967 | 0.0249 | 177s 177ms/step |
| 40 | 0.0157 | 0.9956 | 0.0016 | 0.2165 | 0.9567 | 0.0093 | 175s 175ms/step |

Los cuales representándolos gráficamente según el tipo de valor nos presenta las siguientes gráficas:

**Loss:** Es un valor escalar que intentamos minimizar durante nuestro entrenamiento del modelo. Cuanto menor sea la pérdida, más cerca estarán nuestras predicciones de las etiquetas verdaderas.

### Grafico 1.

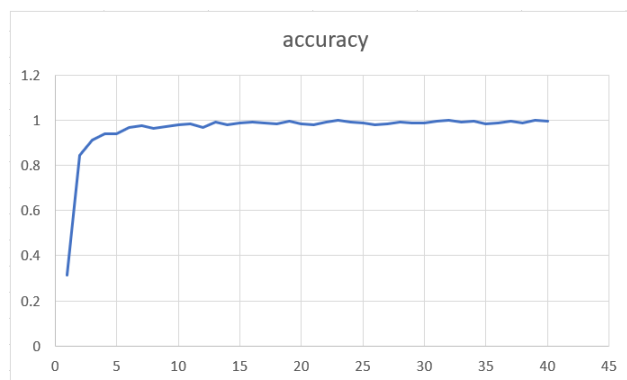
*Métrica Loss.*



**Accuracy:** Cuando se definió en el entrenamiento, lo que hace es calcular la métrica Accuracy, que es una comparación simple entre cuántos valores objetivo coinciden con los valores predichos. Donde cuanto más se acerque a 1, mejores serán los resultados.

## Grafico 2

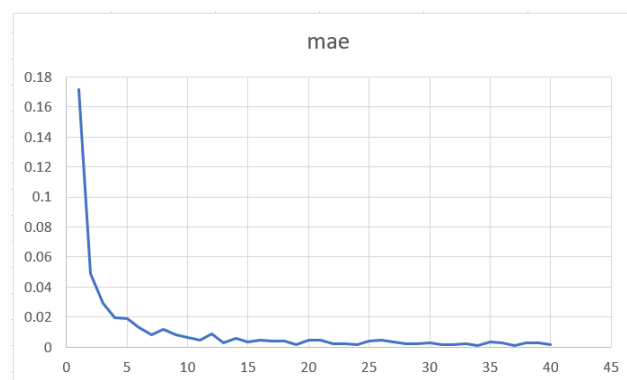
*Métrica Accuracy.*



**Mae:** 0 errores absolutos medios, El error absoluto promedio proporciona el promedio de la diferencia absoluta entre la predicción del modelo y el valor objetivo. Donde cuanto más cerca este del 0, los resultados son mejores.

## Grafico 3

*Métrica Mae.*



## DISCUSIÓN

La sociedad actual se encuentra en una constante vertiginosa transformación. Superar las barreras de comunicación, tanto cultural como social, nos permitirá generar vínculos interpersonales, sociales y culturales mucho más fluidos y orgánicos. La implementación de sistemas automáticos que permitan traducir el Lenguaje de Señas a la lengua escrita y hablada permitirá construir herramientas basadas en la crestomatía, generando un sinfín de ventajas para facilitar la educación, la accesibilidad y la integración de las personas con discapacidad auditiva. El aprendizaje automático permitirá implementar un sistema que identifique y traduzca una lengua a otra en tres

dimensiones, reduciendo las limitaciones actuales que presentan herramientas basadas en computadora (Espinoza et al. 2024).

Por ello, BarriGa (2024) señala que un estudio que ayude a entender y mejorar la comprensión sobre la cultura sorda permitirá a la comunidad oyente tener herramientas que les permita superar prejuicios, barreras y restricciones a la diferencia. A su vez, facilitar el acceso a la educación se implementará en el modelo educativo actual una opción que va a simplificar el acceso presencial o virtual. Acoger un idioma por parte del docente técnico será un apoyo para el aprendiz, rompiendo el miedo y favoreciendo la asunción desde líneas semio-científicas que nos permita ahondar en aspectos que lo tributan en el modelo evolutivo y que han repercutido positivamente en el socioecosistema en el que se encuentran.

Asimismo, el sistema automático desarrollado va a expandir el número de opciones, beneficiando a la comunidad de personas con problemas auditivos. Su implementación en el medio físico, o en la educación a distancia, favorecerá un desarrollo más pleno de la persona, siendo este una comunidad más viva, para el bien de todas las emergencias que proporciona a la variabilidad y hábitat educativo.

En cuanto a la accesibilidad e inclusión, la traducción de lenguaje de señas abre un amplio horizonte hacia la accesibilidad y la inclusión practicadas sobre la comunidad sorda. Por un lado, facilita al colectivo codificado en este lenguaje complejizar su vida, universidad o estudios, derechos, temas de actualidad y demás. Por otro lado, para Elices y de Zafra, (2023) facilita a todos los oyentes interactuar, entender y compartir la misma realidad que una persona sorda. De esta forma, se empieza a trabajar con amigos sordos en cómo grande y espectacular es el mundo sordo, se empiezan a hacer los primeros trabajos en pro de la inclusión y accesibilidad y sus problemas, se empieza a vivir y a visibilizar un mundo nuevo invisible para la mayoría de personas oyentes que no conocen nada de ésta. Sin duda, el aprender la lengua con la que piensan y viven tus amigos es un plus y un motor fundamental para aprender a dominar la lengua señalada.

Cabe tener en cuenta, como señala Hernández-Santana y López-Nava (2024) que en nuestra sociedad actual las personas oyentes tienen toda la información a disposición organizada de una forma concreta con una serie de tipos de registros, contextos y prototipos expertos que construyen y sabemos con certeza que cualquier comunicador nos transmitirá a qué se refiere el símbolo y representará en infinitas tonalidades cada símbolo propio según el contexto. Por otro lado, actualmente, es muy fácil de conocer sin tener que usar demasiados recursos, incluso la venta de productos que por ejemplo consisten en imitar el lenguaje correspondiente, sin tener conocimiento, lógica ni aficiones. La realidad es muy diferente para las personas sordas que no codifican su lengua por un aprendizaje visual regular. Éstas en general no pueden ir un día al cine y disfrutar sin problemas de eso, ni de un teatro, ni de la única conocida-reconocida-comprendida/suponida vía radio. La realidad es la falta de logros de comunicación y el analfabetismo cultural que repercute directamente en el marginalismo o aislamiento social, familiar, comunitario, escolar... total al que se ve sometido ese colectivo.

Por lo tanto, la comunidad también tiene que trabajar en esto, empezando por visibilizar los

problemas generados por el analfabetismo y por el no dominio del léxico-bilingüe. Ya no sólo en la situación de información que generan eventos paralelos, que hoy no pueden ser comprendidos de igual forma que lo suelen ser. Es decir, la comunidad debería aceptar que a desempeñar una lengua también se tiene que aceptar formar parte de la cultura que acompaña a esa lengua. Poder aceptar esto lleva a ser utilizado íntegramente una lengua. Para ello se tiene que aceptar que hay palabras que pertenecen a esa cultura y que no deben de ser suplida por otro que esto se lleva a rote activo porque no se puede instalar (Herrera, 2023).

En términos de beneficios, la inclusión económica, laboral y social facilita que la comunidad sorda pueda integrarse plenamente a la lengua y cultura de un país al adquirir competencias en una lengua escrita, hablada o de signos. Es importante destacar que existe un porcentaje significativo de personas mayores de 15 años que no saben leer ni escribir. La implementación de la propuesta mencionada contribuirá a eliminar las barreras de comunicación al ofrecer una solución tecnológica que permitirá a las personas sordas comunicarse de manera efectiva sin depender de intérpretes con limitaciones, lo que a su vez ayudará a solventar los problemas de comprensión en ambos sentidos entre las personas sordas, sordociegas, y las personas oyentes o con dificultades en el habla (Moná, 2023).

## CONCLUSIONES

El desarrollo de un sistema móvil con aprendizaje automático representa un avance significativo en la optimización de procesos y la toma de decisiones automatizada. A lo largo de la investigación, se ha demostrado cómo la integración de tecnologías de inteligencia artificial en dispositivos móviles permite mejorar la eficiencia, accesibilidad y precisión en distintas aplicaciones. Este tipo de sistemas no solo facilita la automatización de tareas complejas, sino que también impulsa la innovación en sectores clave como la educación, la salud y la industria, proporcionando soluciones adaptativas y personalizadas a las necesidades del usuario.

Además, la implementación de este sistema móvil implica desafíos importantes, como la gestión de grandes volúmenes de datos, la optimización de los algoritmos para funcionar en dispositivos con recursos limitados y la garantía de la seguridad de la información. A pesar de estos obstáculos, las soluciones tecnológicas actuales, como el procesamiento en la nube y los modelos de aprendizaje optimizados, han permitido superar muchas de estas limitaciones, asegurando una experiencia fluida y eficiente para el usuario final. Asimismo, la evolución de los dispositivos móviles y la creciente capacidad de procesamiento seguirán favoreciendo el desarrollo de aplicaciones más avanzadas.

Finalmente, esta investigación resalta la importancia de continuar explorando y mejorando los sistemas móviles con aprendizaje automático, ya que su potencial de impacto en la sociedad es enorme. A medida que la tecnología avanza, es fundamental mantener un enfoque ético en el desarrollo y uso de estos sistemas, asegurando su accesibilidad, equidad y respeto a la privacidad de los usuarios. En este sentido, futuras investigaciones y desarrollos en esta área permitirán ampliar aún más sus aplicaciones y beneficios, contribuyendo al progreso tecnológico y social.

## REFERENCIAS

- BarriGa, L. M. G. (2024). El mundo silenciado: Experiencia etnográfica en una comunidad Sorda. *Revista de Trabajo Social*, (100), 51-66. <https://doi.org/10.7764/rts.100.51-66>
- Elices, A. A., & de Zafra, M. A. A. P. (2023). Protocolo para la traducción a lengua de signos o señas de textos especializados: aproximación desde el proyecto Al-Musactra. *REVLES*, (5), 167-183. <https://revles.es/index.php/revles/article/view/130>
- Espinoza, J. S. A., Martínez, R. C., & Rodriguez, M. D. (2024). Aplicación de la inteligencia artificial para la traducción automática de lengua de señas. *Revista de investigación multidisciplinaria, Iberoamericana*, (4). <https://doi.org/10.69850/rimi.vi4.116>
- Fuentes Del Burgo, J., & Sebastián Pérez, M. Á. (2022). Análisis comparativo de la herramienta tablero en las metodologías ágiles Scrum, Kanban y Scrumban en proyectos de software. *AEIPRO*. <http://dspace.aepro.com/xmlui/handle/123456789/3286>
- Hernández-Santana, G. & López-Nava, I. H. (2024). Sistemas de predicción de lenguas visogestuales basados en IA y su aplicación en la Lengua de Señas Mexicana (LSM). *Inteligencia Artificial*, 211. <https://doi.org/10.61728/AE20241117>
- Herrera, J. C. (2023). Inteligencia artificial, investigación y revisión por pares: escenarios futuros y estrategias de acción. *RES. Revista Española de Sociología*, 32(4), 199. <https://doi.org/10.22325/fes/res.2023.184>
- Ministerio de Culturas y Turismo – Viceministerio de Descolonización. (2014). *Población y cultura sorda de Bolivia. Textos Temáticos*. [https://cultura-sorda.org/wp-content/uploads/2015/04/Comite\\_nacional\\_contra\\_racismo\\_discriminaciC3B3n\\_-\\_Poblacion\\_cultura\\_sorda\\_20141.pdf](https://cultura-sorda.org/wp-content/uploads/2015/04/Comite_nacional_contra_racismo_discriminaciC3B3n_-_Poblacion_cultura_sorda_20141.pdf)
- Ministerio de Educación de Bolivia-Federación Boliviana de Sordos. (2009). *Curso de enseñanza de la Lengua de Señas Boliviana LSB*. <https://www.minedu.gob.bo/files/publicaciones/veaye/dgee/CURSO-DE-ENSENANZA-DE-LA-LENGUA-DE-SENAS-BOLIVIANA-Modulo-1.pdf>
- Ministerio de educación. (2014). *Sistemas de Comunicación II*. S/E. <https://red.minedu.gob.bo/repositorio/fuente/30762.pdf>
- Moná, L. Y. Q. (2023). Influencia del contexto familiar en el aprendizaje del español, como segunda lengua, en sordos señantes; análisis desde experiencias de vida. *Revista Reflexiones y Saberes*, 17, 65-76. <https://revistavirtual.ucn.edu.co/index.php/RevistaRyS/article/view/1562>
- Organización Panamericana de la Salud. (2021). Informe Mundial sobre la Protección Social 2020-2022. [https://www.ilo.org/sites/default/files/wcmsp5/groups/public/@ed\\_protect/@soc\\_sec/documents/publication/wcms\\_842103.pdf](https://www.ilo.org/sites/default/files/wcmsp5/groups/public/@ed_protect/@soc_sec/documents/publication/wcms_842103.pdf)