UNIVERSIDAD NACIONAL DE MOQUEGUA

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING

TESIS

PRESENTADO POR:

BACH. SERGIO ALAIN RAMIREZ AGUILAR

BACH. JHONNY ENRIQUE ANDY VILCA CONDORI

PARA OPTAR POR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS E INFORMÁTICA

ASESOR:

M.SC. HUGO EULER TITO CHURA

ILO - PERÚ

2025

Folio Nº 023

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis : [X]
Trabajo de Suficiencia Profesional : [ ]
Otro: ..... : [ ]

II. Documento de aprobación para la sustentación:

Resolución 275-2025-CF-FAIN-UNAM

Fecha: 12-12-2025

III. El jurado ha dictaminado la (el) Tesis, TITULADO:

Desarrollo de una aplicación móvil didactica para el aprendizaje de Lenguaje de Señas Peruano usando Deep Learning

Bachiller: Sergio Alain Ramirez Aguilar

Asesor Principal: Msc. Hugo Ederlito Tito Chura

Co-Asesor: .....

IV. Se dictamina:

Como: aprobado por unanimidad la sustentación de manera:

Presencial [X] Virtual [ ]

Con nota: Dieciseis (16) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA.

Siendo las 2:47 horas, del 12 de diciembre del año 2025 se procede con la firma en conformidad de lo dictaminado por el Jurado.

José A. Gutiérrez Gallardo
Presidente

Raúl Aragón Encinas
Primer Miembro

Dra. Morales Gonzales Rosa A.
Segundo Miembro

Sergio Ramirez
Bachiller

UNAM
UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS E INFORMÁTICA

Folio Nº **024**

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis	:	[X]
Trabajo de Suficiencia Profesional	:	[ ]
Otro:	:	[ ]

II. Documento de aprobación para la sustentación:

Resolución 275-2025-CF-FAIN-UNAM

Fecha: 12-12-2025

III. El jurado ha dictaminado la (el) Tesis, TITULADO:

Desarrollo de una aplicación móvil didáctica para el aprendizaje del lenguaje de Señas Peruano usando Deep Learning

Bachiller: Jhonny Enrique Andy Vilca Condori

Asesor Principal: M.Sc Hugo Ruber Tilo Chura

Co-Asesor: .....

IV. Se dictamina:

Como: aprobado por unanimidad la sustentación de manera:

Presencial [X] Virtual [ ]

Con nota: dieciseis (16) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA.

Siendo las 2:47 horas, del 12 de diciembre del año 2025 se procede con la firma en conformidad de lo dictaminado por el Jurado.

José A. Gozmán Valdivia
Presidente

Saul Huaquipaco Encinas
Primer Miembro

Dr. Marcelo Gonzales Rojas
Segundo Miembro

Jhonny Vilca Condori
Bachiller

DEDICATORIA

De Sergio Alain Ramírez Aguilar:

Agradecer a Dios siempre por acompañarme en la realización de este proyecto, y a la par por siempre guiarme por el buen camino y darme la fuerza necesaria para nunca rendirme.

A mi mamá que siempre me la da las energías necesarias, el cariño constante y la seguridad para siempre ser una mejor persona. A mi papá que tristemente falleció no hace mucho; pero de igual fue un eslabón importante en mi desarrollo profesional; también agradecerle por todo el esfuerzo, cariño, apoyo y dedicación que nunca me hicieron falta; este logro es para ustedes.

A mi gata Misha que, a pesar de ya no estar en esta vida, su presencia fue parte crucial en mi trayecto universitario y por consiguiente en la realización de este proyecto.

A mis familiares y todos aquellos que siempre estuvieron ahí dándome aliento y el apoyo necesario para llegar donde estoy ahora.

De Jhonny Enrique Andy Vilca Condori:

Escribir una sección como la dedicatoria es el final de la etapa de tesis, donde puedo agradecer y dedicar el trabajo a las personas que estuvieron conmigo y me apoyaron en el camino:

A mí, por darme aliento y el tiempo de poder desarrollar la tesis, por esforzarme y acordarme de avanzar; para de ese modo poder culminarlo en un tiempo no tan largo.

A mi mamá que siempre estuvo conmigo escuchándome cuando le comentaba aspectos de mi tesis que estoy seguro que no me entendía, pero me prestaba atención y eso era todo el apoyo que requería.

A mi mejor amiga Daniela, por ayudarme en la gestión y elaboración del dataset utilizado en esta tesis, por siempre estar ahí ayudándome en lo que requería, por existir y hacer que la vida sea más divertida.

A mi trabajo que me permitió tener el tiempo requerido para poder avanzar con mi tesis, caso contrario, no hubiera terminado de redactar y desarrollar lo planteado en este trabajo.

Y a todos aquellos, que, sin saberlo, sumaron con alguna palabra, apoyo o simplemente estando presentes, gracias también.

RESUMEN

El presente trabajo se enfoca en el desarrollo de una aplicación móvil didáctica de nombre Sígnalo, la cual está orientada al aprendizaje del Lenguaje de Señas Peruano (LSP); mediante el uso y apoyo de técnicas avanzadas de Deep Learning. La propuesta surge como respuesta a la necesidad de reducir las barreras de comunicación existentes que enfrentan las personas con discapacidad auditiva en el Perú, dado que el desconocimiento generalizado del lenguaje de señas y la escasez de intérpretes formales son un limitante para su inclusión en ámbitos clave como la educación, trabajo y salud. Con este propósito, se planteó como objetivo principal diseñar e implementar una herramienta móvil interactiva que, mediante modelos de aprendizaje profundo, permita tanto el reconocimiento automático como la enseñanza efectiva del abecedario en LSP.La investigación adoptó un enfoque cuantitativo, aplicado, no experimental y de corte transversal. Se elaboró un dataset propio de 25,000 imágenes, sobre el cual se entrenaron modelos de redes neuronales convolucionales utilizando técnicas de transferencia de aprendizaje con las arquitecturas EfficientNetB0 y MobileNetV2. Esta última fue seleccionada para su integración final por su alto rendimiento (F1-score de 97.66%) y eficiencia en dispositivos móviles de gama media (modelo final de 8.8 MB). La aplicación fue evaluada en escenarios reales, demostrando tiempos de inferencia óptimos, funcionamiento fluido, estabilidad y precisión en tiempo real. Asimismo, se validó su usabilidad mediante encuestas a usuarios reales, obteniendo una alta aceptación en aspectos como facilidad de uso, claridad de la interfaz y utilidad percibida. Se concluye que el objetivo fue alcanzado exitosamente, y que SignalGo representa una solución tecnológica inclusiva, educativa y viable para el aprendizaje del Lenguaje de Señas Peruano, con gran potencial de escalabilidad y mejora.

Palabras clave: Lenguaje de señas peruano, Deep Learning, aplicación móvil, inclusión, redes neuronales convolucionales, MobileNetV2, reconocimiento de imágenes, aprendizaje automático.

ABSTRACT

This paper focuses on the development of an educational mobile application called SignalGo, which is designed to teach Peruvian Sign Language (LSP) through the use and support of advanced deep learning techniques. The proposal arises in response to the need to reduce existing communication barriers faced by people with hearing disabilities in Peru, given that widespread ignorance of sign language and a shortage of formal interpreters are a limitation to their inclusion in key areas such as education, work, and health. To this end, the main objective was to design and implement an interactive mobile tool that, through deep learning models, would enable both automatic recognition and effective teaching of the LSP alphabet. The research adopted a quantitative, applied, non-experimental, cross-sectional approach. A proprietary dataset of 25,000 images was created, on which convolutional neural network models were trained using transfer learning techniques with the EfficientNetB0 and MobileNetV2 architectures. The latter was selected for final integration due to its high performance (F1-score of 97.66%) and efficiency on mid-range mobile devices (final model of 8.8 MB). The application was evaluated in real-world scenarios, demonstrating optimal inference times, smooth operation, stability, and real-time accuracy. Its usability was also validated through surveys of real users, obtaining high acceptance in areas such as ease of use, interface clarity, and perceived usefulness. It is concluded that the objective was successfully achieved

and that SignalGo represents an inclusive, educational, and viable technological solution for learning Peruvian Sign Language, with great potential for scalability and improvement.

Keywords: Peruvian Sign Language, Deep Learning, mobile application, inclusion, convolutional neural networks, MobileNetV2, image recognition, machine learning.

ÍNDICE

LISTA DE TABLAS....11

LISTA DE FIGURAS....12

LISTA DE SIGLAS....14

INTRODUCCIÓN....15

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA....17

1.1 Descripción de la realidad problemática....17

1.2 Formulación del Problema....19

1.2.1 Problema General....19

1.2.2 Problemas Específicos....19

1.3 Objetivos....19

1.3.1 Objetivo general:....19

1.3.2 Objetivos específicos:....20

1.4 Justificación e importancia....20

1.5 Variables....22

- Variable 1: Modelo de Deep Learning.....22
- Variable 2: Precisión del Modelo.....22
- Variable 3: Usabilidad de la Aplicación Móvil.....22

CAPITULO II: MARCO TEÓRICO....23

2.1. Antecedentes....23

2.2. Bases Teóricas....25

2.2.1. Lenguaje de señas....25

2.2.2. Inteligencia artificial y aprendizaje automático....26

2.2.3. Desarrollo de aplicaciones móviles....26

2.2.4. Procesamiento de imágenes y video 27
2.2.5. Interacción humano computadora (IHC) 27
2.2.6. Deep learning.....28
2.2.6.1 Redes neuronales artificiales (ANN) 28
2.2.6.2 Red neuronal convolucional (CNN) 29
2.2.6.3 Data Augmentation.....29
2.2.6.4 Métricas de evaluación de modelos de IA 30
2.2.6.5 Usabilidad del sistema 31
2.2.6.6 Adquisición de datos 31
2.2.6.7 MobileNet 32
2.2.6.8 EfficientNet.....32
2.2.6.9. Aplicaciones de MobileNetV2 y EfficientNetB0 en reconocimiento de Lenguaje de señas.....33
2.3. Definición de términos.....34
2.3.1 Deep learning.....34
2.3.2 Lenguaje de señas.....34
2.3.3 CNN 35
2.3.4 Data augmentation.....35
2.3.5 Visión por computadora 35
2.3.6 Aplicativo móvil 35
2.3.7 Reconocimiento de gestos 35
2.3.8 Overfitting.....35
2.3.9 Interfaz de usuario 36
CAPITULO III: METODOLOGÍA.....37
3.1 Tipo y nivel de investigación.....37

3.2 Diseño de investigación....37
3.3. Población y muestra....39
3.3.1 Población....39
3.3.2 Muestra....41
3.4. Instrumentos....42
3.4.1. Python....42
3.4.2. Tensorflow/Keras....43
3.4.3. Android studio....43
3.4.4. Redes neuronales convolucionales....43
3.4.5. Métricas de evaluación....44
3.4.6. Jupyter notebooks....44
3.4.7. OpenCV....45
3.4.8. Tensorflow lite....45
3.5 Obtención y Procesamiento de Datos....45
3.5.1 Recolección y organización de datos....45
3.5.2 Preprocesamiento de datos....46
3.5.3 Distribución del dataset....48
3.6 Desarrollo del Modelo e Implementación de la App....51
3.6.1 Selección de modelos base: EfficientNetB0 y MobileNetV2....51
3.6.1.1 EfficientNetB0....51
3.6.1.2. MobileNetV2....51
3.6.2 Configuración del entrenamiento....52
3.6.3 Transfer learning....53
3.6.4 Evaluación de los hiperparametros....54
3.6.5 Evaluación de los modelos....61

3.6.6 Evaluación del funcionamiento de los modelos en tiempo real 62
3.6.6.1 Pruebas en tiempo real 62
3.6.7 Integración del modelo en la aplicación móvil 63
3.7 Evaluación de usabilidad y efectividad didáctica 78
CAPITULO IV: RESULTADOS Y DISCUSIÓN 79
4.1 Resultados 79
4.1.1. Recolección y Preparación del Dataset.....79
4.1.2. Desarrollo e integración del modelo de reconocimiento en la aplicación móvil……..80
4.1.2.1 Resultados generales por arquitectura.....80
4.1.2.2 Matriz de confusión y análisis de errores 82
4.1.2.3 Predicciones en imágenes aleatorias.....84
4.1.2.4 Evaluación de la aplicación móvil en condiciones reales 85
4.1.2.5 Tiempo de inferencia promedio.....86
4.1.2.6 Funcionamiento en tiempo real 86
4.1.2.7 Robustez ante variaciones externas 87
4.1.3 Resultados de usabilidad 88
4.2 Discusión de Resultados 90
CONCLUSIONES 94
RECOMENDACIONES 96
REFERENCIAS.....98
ANEXOS 103

LISTA DE TABLAS

Tabla 1 *Tabla de Resultados con EfficienNetB0* 59

Tabla 2 *Tabla de Resultados con MobileNetV2* 59

Tabla 3 *Resultados Generales por Arquitectura* 81

Tabla 4 *Tamaño del modelo convertido a TensorFlow Lite* 82

Tabla 6 *Variaciones aplicadas y resultados observados* 87

Tabla 7 *Puntaje Global del System Usability Scale (SUS)* 88

Tabla 8 Limitaciones Observadas y sus Detalles 90

LISTA DE FIGURAS

Figura 1 Presentación y análisis de datos 48

Figura 2 Distribución del Dataset.....50

Figura 3 Resultados obtenidos con diferentes valores de Learning Rate y Batch Size para EfficientNetB0 56

Figura 4 Resultados obtenidos con diferentes valores de Learning Rate y Batch Size para MobileNet.....57

Figura 5 Métricas de Entrenamiento y Evaluación con MobileNetV2.....61

Figura 6 Métricas de Entrenamiento y Evaluación con EfficientNetB0 61

Figura 7 Diseño de la Interfaz en Android Studio 65

Figura 8 Interfaz del aplicativo móvil.....65

Figura 9 Diseño de la interfaz del minijuego en Android Studio.....67

Figura 10 Interfaz del minijuego 67

Figura 11 Base de Datos del Activity 1 68

Figura 12 Diseño de la interfaz del activity donde se va a reconocer la seña.....70

Figura 13 Resultado de la interfaz finalizada donde se reconocerá las señas.....70

Figura 14 Diseño de la Interfaz Juego de Cartas.....71

Figura 15 Resultado de la interfaz del juego de cartas.....72

Figura 16 Diseño de la Interfaz Insignias y Progreso 73

Figura 17 Resultado de la interfaz Insignias y Progreso.....74

Figura 18 Base de Datos de la interfaz Insignias y Progresos.....74

Figura 19 Diseño de la Interfaz Diccionario de Señas 75

Figura 20 Resultado de la interfaz Diccionario de Señas 76

Figura 21 Resultado de la interfaz Diccionario de Señas en la sección búsqueda 76

Figura 22 Base de Datos de la interfaz Diccionario de Señas 77

Figura 23 Matriz de confusión del modelo MobileNetV2.....84

Figura 24 Etiquetas vs predicciones de imágenes aleatorias del modelo MobileNetV285

LISTA DE SIGLAS

LSP	: Lenguaje de señas peruano
IA	: Inteligencia artificial
DL	: Aprendizaje Profundo (Deep Learning)
CNN	: Red neuronal convolucional (Convolutional Neural Network)
XML	: Lenguaje de marcado extensible (Extensible Markup Language)
RGB	: Red Green Blue
SGD	: Descenso de gradiente estocástico (Stochastic gradient descent)
IHC	: Interacción humano computador
IOS	: Sistema Operativo de Iphone (iPhone Operating System)
INEI	: Instituto Nacional de Estadística e informática

INTRODUCCIÓN

En una sociedad cada vez más interconectada, la inclusión social y la accesibilidad se han convertido en pilares importantes para el desarrollo de sociedades con igualdad de oportunidades. Sin embargo, las personas con discapacidad auditiva enfrentan barreras significativas en la comunicación, lo que dificulta su participación en diversos ámbitos de la vida. En el Perú, donde la comunidad sorda representa un sector importante de la población, la carencia de herramientas educativas accesibles y eficaces para el aprendizaje del Lenguaje de Señas Peruano (LSP) es un desafío constante. Esta situación afecta a las personas sordas, como también a sus familias, educadores y profesionales del entorno que buscan establecer puentes de comunicación efectivos.

En contexto, la tecnología sobresale como un aliado poderoso para superar estas barreras. El desarrollo de aplicaciones móviles didácticas, integradas con técnicas avanzadas de IA como el deep learning, nos ofrece una oportunidad única para enseñar el aprendizaje del LSP. Estas herramientas además de tener la capacidad de facilitar el aprendizaje de habilidades en el lenguaje de señas, también cierran brechas, promoviendo la inclusión y el entendimiento mutuo entre personas sordas y oyentes.

El presente trabajo de investigación se enfoca en el diseño e implementación de una aplicación móvil didáctica que utiliza deep learning para apoyar el aprendizaje del Lenguaje de Señas Peruano. A través de un enfoque tecnológico y social; se busca crear un aplicación interactiva y accesible que permita a los usuarios, aprender y practicar el LSP de manera autónoma y eficiente. La aplicación integrará modelos de reconocimiento de señas, entrenados con datos específicos del LSP, para proporcionar reconocimiento en tiempo real y garantizar una experiencia de aprendizaje personalizada.

Esta tesis a parte de abordar un problema social relevante, buscar la exploración de las posibilidades que ofrece la inteligencia artificial en el ámbito educativo. Al combinar tecnología como lo es la IA con una perspectiva inclusiva, se espera contribuir al desarrollo de herramientas que fomenten la igualdad de oportunidades y el respeto por la diversidad lingüística y cultural. A través de este proyecto, se aspira a sentar las bases para futuras investigaciones y aplicaciones que impulsen la inclusión de las personas sordas en el Perú y más allá.

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción de la realidad problemática

La comunicación oral es esencial para que las personas puedan comunicar sus pensamientos e ideas a los demás, lo que permite una interacción social efectiva. No obstante, para las personas con discapacidad auditiva, representa una barrera que les impide comprenderse mutuamente. El lenguaje de señas es crucial porque permite la comunicación con las personas sordas. Desafortunadamente, este idioma aún no se usa ampliamente en nuestro país. La exclusión social de las personas sordas es significativa porque la mayoría de las personas desconocen esta forma de comunicación. No pueden interactuar plenamente con la sociedad y participar activamente en ella debido a esta falta de conocimiento; enfrentarán dificultades en temas fundamentales como el acceso al trabajo, la educación y la salud. Es por ello, que el lenguaje de señas es una herramienta fundamental para reducir en cierta manera las barreras de comunicación, logrando en las personas sordas poder expresar sus opiniones e ideas con los demás, generando de esa forma un ambiente social inclusivo.

En el Perú, hay 1,619,880 personas con discapacidad, lo que equivale al 5.2 % de la población total. Según el CENSO del INEI (Instituto Nacional de Estadística e Informática) realizado en 2015, se estima que 560 730 personas de este grupo tienen discapacidad auditiva, lo que representa el 1.8 % de la población total. En el mismo año, solo 24 intérpretes del Perú han sido reconocidos y calificados por ASISEP (Asociación de Intérpretes y Guías de Intérpretes de Lengua de Señas del Perú) a pesar de esta gran cantidad (Perú, s.f.).

Según la presidenta de ASISEP, Isabel Rey Clemente, hay una gran cantidad de intérpretes empíricos que han aprendido el lenguaje de señas a través de amigos o familiares sordos, sin emplear técnicas de interpretación adecuadas. Es alarmante la falta de intérpretes formales. El aprendizaje y la especialización en el lenguaje de señas son esenciales para mejorar la comunicación y la inclusión de las personas con discapacidad auditiva en áreas fundamentales como el trabajo, la educación y la salud; esta situación destaca la urgencia de promoverlo.

En los antecedentes de trabajos similares, se ha logrado identificar la aplicación de diversas técnicas de machine learning, como Support Vector Machine (SVM), redes neuronales convolucionales (CNN) y Oriented Fast and Rotated Brief (ORB). Estas técnicas han demostrado ser efectivas en el reconocimiento y traducción de señas. Además, se utilizará una fuente propia que se basa en los estándares del lenguaje de señas peruano para recopilar la información. Para garantizar la relevancia y la precisión de las señas, se utilizará como referencia imágenes y videos del abecedario y de las señas más utilizadas del lenguaje de señas, que se encuentran en un repositorio de ASISEP.

En este trabajo proponemos un enfoque más amplio, ya que algunos de los trabajos relacionados solo desarrollan el reconocimiento de unas cuantas letras y no lo utilizan para enseñar en un ámbito educativo. Usando algoritmos de deep learning, trabajaremos en la creación de un aplicativo móvil que pueda reconocer y traducir el lenguaje de señas. Nuestro propósito es abarcar todas las letras y señas más comunes y transmitir este conocimiento de manera pedagógica y didáctica. Para promover una mayor comprensión e inclusión del lenguaje de señas peruano, esta aplicación está

destinada a enseñar el lenguaje de señas a los usuarios finales de manera interactiva y accesible.

1.2 Formulación del Problema

1.2.1 Problema General

- ¿Es posible desarrollar un aplicativo móvil basado en deep learning que facilite tanto el aprendizaje como el reconocimiento del lenguaje de señas?

1.2.2 Problemas Específicos

- ¿Cómo se puede recolectar y procesar un conjunto de datos visuales del Lenguaje de Señas Peruano que sea adecuado para entrenar un modelo de deep learning?
- ¿Qué técnicas de deep learning resultan más efectivas para desarrollar un modelo que reconozca señas en tiempo real a través de una cámara móvil?
- ¿Qué nivel de usabilidad y efectividad didáctica presenta la aplicación cuando es utilizada por usuarios en contexto de prueba?

1.3 Objetivos

1.3.1 Objetivo general:

- Desarrollar un aplicativo móvil didáctico usando deep learning para el aprendizaje y reconocimiento de lenguaje de señas peruano.

1.3.2 Objetivos específicos:

- Recolectar y procesar datos visuales del Lenguaje de Señas para entrenar un modelo de deep learning.
- Desarrollar e integrar un modelo de reconocimiento de señas basado en deep learning dentro de la aplicación móvil.
- Evaluar la usabilidad y efectividad didáctica del aplicativo móvil mediante pruebas con usuarios reales.

1.4 Justificación e importancia

En el Perú, la comunidad sorda enfrenta importantes barreras para acceder a servicios fundamentales, debido a una brecha comunicacional que persiste en distintos entornos sociales. Según el INEI 2017 (Instituto Nacional de Estadística e Informática, 2017), existen más de 532,000 personas con discapacidad auditiva, pero solo hay alrededor de 24 intérpretes certificados a nivel nacional (Ministerio de Educación del Perú, 2021). Esta situación limita el acceso equitativo a la educación, la atención médica y oportunidades laborales, especialmente en contextos donde la disponibilidad de intérpretes es reducida o inexistente.

Frente a este panorama, la aplicación móvil propuesta busca contribuir al proceso de aprendizaje de la Lengua de Señas Peruana (LSP) mediante el uso de inteligencia artificial, permitiendo interpretar en tiempo real las señas realizadas por el usuario y ofrecer retroalimentación inmediata. Aunque no pretende reemplazar el rol fundamental de los intérpretes profesionales ni cubrir toda la complejidad lingüística de la LSP, la app se plantea como una herramienta accesible que favorece el aprendizaje autónomo y

continuo, ayudando así a fortalecer la inclusión social y educativa. Esto cobra especial relevancia si se considera que menos del 10 % de instituciones educativas en el país cuenta con recursos adaptados para estudiantes sordos (Ministerio de Educación del Perú, 2021).

Desde el enfoque tecnológico, esta propuesta ofrece un valor agregado respecto a muchas aplicaciones existentes, las cuales suelen limitarse a mostrar vocabulario estático. A través del uso de redes neuronales convolucionales (CNN), la app permite el reconocimiento en tiempo real de las señas, adaptándose a variaciones individuales y promoviendo una experiencia interactiva. Además, se apoya en la infraestructura tecnológica actual, ya que el 72 % de los peruanos utiliza dispositivos móviles (Organismo Supervisor de Inversión Privada en Telecomunicaciones, 2023), lo que facilita su implementación sin necesidad de equipos adicionales.

En el plano económico, el uso de esta herramienta puede contribuir a reducir los costos asociados al aprendizaje básico de la LSP, al evitar la necesidad de talleres presenciales que, según CONADIS, suelen superar los S/500 por persona (CONADIS, 2021). Asimismo, en ciertos contextos básicos podría complementar la comunicación sin sustituir la labor profesional de intérpretes, cuyo trabajo sigue siendo esencial en escenarios más complejos.

Finalmente, el proyecto cuenta con potencial de crecimiento, permitiendo en el futuro la incorporación de nuevas señas o adaptaciones a dialectos regionales, lo que ampliará su impacto a nivel nacional.

1.5 Variables

- **Variable 1:** Modelo de Deep Learning.
- **Variable 2:** Precisión del modelo.
- **Variable 3:** Usabilidad de la aplicación móvil.

CAPITULO II: MARCO TEÓRICO

2.1. Antecedentes

(Kumari & Anand, 2024) desarrollaron un modelo híbrido para reconocimiento de señas en video combinando una CNN, un módulo de atención y una LSTM para modelado temporal. El sistema alcanzó 84.65 % de accuracy en el dataset WLASL100, superando arquitecturas puramente convolucionales y demostrando que la atención mejora la estabilidad del reconocimiento en secuencias dinámicas.

(Hussain et al., 2023) implementaron un sistema de reconocimiento de señas estáticas mediante transfer learning con Inception-V3 y EfficientNet-B0. Tras el ajuste fino, EfficientNet-B0 obtuvo el mejor rendimiento, logrando 99.1 % de accuracy, 98.9 % de precisión y 98.7 % de F1-score, evidenciando que arquitecturas eficientes modernas pueden alcanzar alta exactitud con datasets moderados.

(Baihan et al., 2024) propusieron un modelo híbrido CNNSa-LSTM, que integra convoluciones profundas, self-attention y LSTM, optimizado mediante un algoritmo híbrido. El modelo alcanzó **98.7 % de accuracy** y redujo los errores WER y SER, superando a arquitecturas CNN y CNN-LSTM tradicionales. La attention permitió capturar mejor patrones temporales complejos en secuencias de señas.

(Berru Novoa, 2025) desarrollaron un sistema de reconocimiento del alfabeto de la Lengua de Señas Peruana utilizando un dataset de 2,400 imágenes capturadas con cámaras de baja resolución. Para la clasificación se empleó HOG junto con cuatro algoritmos de aprendizaje automático, obteniéndose el mejor desempeño con SVM,

alcanzando 89.46 % de accuracy y mostrando robustez ante variaciones de traslación, rotación y escala. El estudio demuestra que técnicas clásicas de visión por computadora pueden ofrecer buenos resultados incluso con hardware limitado.

(Vega Calero, 2023) evaluaron una aplicación basada en deep learning para mejorar la comprensión y producción de signos básicos de la LSP en estudiantes del CEBE Manuel Duato. La comprensión aumentó de 41 **%** a 74 **%** (+33 puntos) y la producción de 54 % a 79 **%** (+25 puntos). Estos resultados muestran que las herramientas de DL fortalecen el aprendizaje de la LSP y contribuyen a la inclusión comunicativa en entornos de educación especial.

(Salgado Martínez et al., 2024) implementaron una aplicación móvil para reconocer señas fijas y en movimiento de la LSM. El modelo MLP alcanzó 96.2 **%** de accuracy en señas estáticas, mientras que el Transformer combinado con CNN obtuvo 94.16 **%** en señas dinámicas. En el dispositivo móvil, las señas fijas presentaron inferencias de 0.60 ms y las dinámicas superaron el 93 **%** de precisión con un uso máximo de 1 MB de memoria, confirmando la eficacia de modelos ligeros para traducción de LSM en móviles.

(Llantoy Luján, 2024) desarrollaron un aplicativo móvil basado en deep learning para mejorar la interpretación y comprensión de la Lengua de Señas Peruana. La precisión aumentó de 49.2 % a 88.97 **%** (mejora del 40.38 %) y el tiempo de comprensión se redujo de 16.6 s a 4.6 s (–72.3 %). Además, la percepción de accesibilidad comunicativa mejoró de una mediana de 2 a 4 en la escala de Likert. Estos resultados

evidencian que el uso de aplicaciones basadas en DL potencia el aprendizaje y la inclusión comunicativa en usuarios de LSP.

(Imran, et al., 2021) implementaron el algoritmo Super Vector Machine(SVM) en una aplicación para dispositivos móviles donde a través del reconocimiento de imágenes se traducirá a lenguaje Urdo, que es el lenguaje usado en Arabia; la data fue creada y recolectada por los mismos autores, esta investigación tuvo un accuracy de casi 90 %.

(Wang & Chen, 2020) adaptaron la arquitectura MobileNetV2 para reconocimiento de ASL en dispositivos móviles, implementando técnicas de cuantización que redujeron el tamaño del modelo en un 60 %. Evaluado con el dataset ASL Alphabet (87,000 imágenes), alcanzó 82 % de precisión con tiempos de inferencia menores a 50ms en un Xiaomi Redmi Note 8. Los autores identificaron como principal desafío el rendimiento en condiciones de baja iluminación, donde la precisión caía al 68 %, este estudio es fundamental para entender las limitaciones de CNN en entornos reales.

(Ortiz & Camargo, 2020) implementaron un software para el reconocimiento de lenguaje de señas usando redes neuronales convolucionales(CNN) teniendo como resultados un accuracy del 68 % de las 22 clases que se usaron para el entrenamiento.

2.2. Bases Teóricas

2.2.1. Lenguaje de señas

El Lenguaje de Señas Peruana (LSP) es el sistema lingüístico visual-gestual utilizado por la comunidad sorda del Perú. Al igual que otras lenguas de señas, es una lengua natural con estructura gramatical propia, distinta del español, y se basa en

componentes manuales (configuración, movimiento y ubicación de las manos) y no manuales (expresiones faciales, movimientos corporales y labiales) (Muñoz, 2013).El LSP fue reconocido oficialmente en el Perú mediante la Ley N° 29535 (Congreso de la República del Perú, 2010), que lo declara como la lengua de la comunidad sorda peruana y promueve su inclusión en educación y servicios públicos. Además, el Reglamento de la Ley General de la Persona con Discapacidad (Ministerio de Salud del Perú (MINSA), 2014) exige la accesibilidad en comunicación para sordos, incluyendo intérpretes de LSP en instituciones estatales.

2.2.2. Inteligencia artificial y aprendizaje automático

El reconocimiento de gestos se basa en la inteligencia artificial (IA) y el aprendizaje automático (Machine Learning). De acuerdo con (LeCun, Bengio, & Hinton, 2015), el aprendizaje profundo (Deep Learning) ha transformado el ámbito de la visión computacional, posibilitando la creación de modelos que pueden identificar patrones complejos en imágenes y vídeos. Las redes neuronales convolucionales (CNN) resultan particularmente beneficiosas para el procesamiento de imágenes, dado que pueden obtener automáticamente atributos visuales. Dentro del marco del lenguaje de señas, los modelos de Inteligencia Artificial se capacitan utilizando grandes volúmenes de datos (datasets) que incluyen ejemplos de gestos. Estos modelos aprenden a vincular patrones visuales con interpretaciones concretas, facilitando así la interpretación automática de señas.

2.2.3. Desarrollo de aplicaciones móviles

La creación de aplicaciones para el reconocimiento de lenguaje de señas en dispositivos móviles necesita la incorporación de modelos de Inteligencia Artificial en

plataformas como Android o iOS. De acuerdo con (Deitel & Deitel, 2017), el desarrollo móvil requiere la utilización de lenguajes como Java, Kotlin (en Android) o Swift (en iOS). Adicionalmente, frameworks multiplataforma como Flutter facilitan la creación de aplicaciones que operan en ambos sistemas operativos utilizando un único código base (Greenspan & Loftus, 2020). La incorporación de modelos de Inteligencia Artificial en aplicaciones para móviles se simplifica con herramientas como TensorFlow Lite, que posibilita la ejecución de modelos de aprendizaje profundo en aparatos con recursos escasos (Google, 2021). Es fundamental para asegurar que el aplicativo sea eficaz y fácil de usar.

2.2.4. Procesamiento de imágenes y video

El manejo de imágenes y videos es crucial para la identificación de gestos. De acuerdo con (Forsyth & Ponce, 2012), métodos como la normalización, el filtrado y la extracción de atributos facilitan la mejora de la exactitud de los modelos de Inteligencia Artificial. Instrumentos como OpenCV simplifican la aplicación de estos procedimientos en aplicaciones para móviles.

2.2.5. Interacción humano computadora (IHC)

La interacción humana-computadora (HCI) es una disciplina que analiza la forma en que los individuos interactúan con los ordenadores y cómo crear sistemas que resulten sencillos de manejar. De acuerdo con (Norman, 2013), el diseño de interfaces debe enfocarse en la facilidad de uso y la accesibilidad, particularmente en aplicaciones destinadas a individuos con discapacidad. Para el lenguaje de señas, es esencial crear interfaces fácilmente comprensibles que faciliten una comunicación eficaz entre el usuario y el programa.

2.2.6. Deep learning

El Deep Learning (Aprendizaje Profundo) es un subcampo del Machine Learning (Aprendizaje Automático) que se basa en redes neuronales artificiales con múltiples capas de abstracción. Su principio fundamental es aprender representaciones de datos a través de capas sucesivas, donde cada nivel aprende a transformar la entrada en una representación más abstracta y compuesta (LeCun, Bengio, & Hinton, 2015). La teoría detrás del Deep Learning se sustenta en la idea de que estas arquitecturas profundas pueden modelar funciones complejas y de alta dimensionalidad de manera más eficiente que los modelos superficiales, gracias a su capacidad de composicionalidad y jerarquía.

2.2.6.1 Redes neuronales artificiales (ANN)

Las Redes Neuronales Artificiales son modelos computacionales inspirados en la estructura biológica del cerebro. El modelo conceptual de una neurona artificial, el Perceptrón, fue introducido por Frank Rosenblatt en 1958 (Rosenblatt, 1958). Una ANN típica consta de:

- Capa de entrada: Recibe los datos iniciales.
- Capas ocultas: Procesan la información mediante transformaciones no lineales.
- Capa de salida: Produce el resultado final.

El algoritmo de retropropagación (backpropagation) desarrollado en los años 80 (Rumelhart, 1986) permite entrenar redes con múltiples capas ajustando los pesos sinápticos mediante descenso de gradiente, minimizando el error entre las predicciones y los valores reales.

2.2.6.2. Red neuronal convolucional (CNN)

Las CNNs son arquitecturas especializadas para procesar datos con estructura de grid, como imágenes. Inspiradas en el sistema visual de los mamíferos (Hubel, 1962), su estructura incluye:

- Capas convolucionales: Aplican filtros (kernels) para detectar características locales.
- Capas de pooling: Reducen la dimensionalidad mediante submuestreo.
- Capas fully connected: Realizan la clasificación final basándose en las características extraídas.

Arquitecturas pioneras como LeNet-5 (LeCun, Bottou, Bengio, & Haffner, 1998) y AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) demostraron la superioridad de las CNNs en tareas de visión por computadora.

2.2.6.3. Data Augmentation

Data Augmentation es una técnica que consiste en aplicar distintos tipos de transformaciones a los datos originales con el propósito de generar nuevos datos (Krizhevsky, Sutskever, & Hinton, 2017). Entre las transformaciones más conocidas y ampliamente utilizadas están la rotación, inclinación, traslación, recortado, escalado, modificación de los colores (tono, brillo, saturación), adición de ruido gaussiano, entre otras. Este tipo de transformaciones son muy útiles, ya que no afectan a las clases de los datos.

2.2.6.4. Métricas de evaluación de modelos de IA

La evaluación de modelos de clasificación se basa en la teoría de la teoría de la decisión estadística y el análisis de los posibles resultados, resumidos en la Matriz de Confusión (Provost, 2001). A partir de esta matriz (que cruza las clases reales con las predichas), se derivan métricas fundamentales:

- Exactitud (Accuracy), mide el porcentaje total de predicciones correctas realizadas por el modelo. Es útil para tener una visión general del rendimiento, pero puede ser engañosa en conjuntos de datos desbalanceados.
- Precisión (Precision), indica la proporción de predicciones positivas que fueron correctas. Es especialmente importante en aplicaciones donde los falsos positivos pueden tener consecuencias significativas, como en entornos médicos o educativos.
- Sensibilidad (Recall), mide la capacidad del modelo para identificar correctamente todas las instancias relevantes. Un alto recall es crucial para asegurar que no se pasen por alto señas importantes.
- Puntuación F1 (F1-Score), combina precisión y recall en un solo valor, proporcionando un balance entre ambas métricas. Es particularmente útil cuando se necesita equilibrar la importancia de los falsos positivos y falsos negativos.
- La matriz de confusión es una herramienta fundamental para evaluar el rendimiento de modelos generados o entrenados basados en Deep learning. Proporciona una representación visual completa de los aciertos y errores del modelo, permitiendo un análisis detallado de su comportamiento.

2.2.6.5. Usabilidad del sistema

La usabilidad es un concepto central dentro del campo de la **I**nteracción Humano-Computadora (HCI). Jakob Nielsen, uno de los autores más influyentes, la define a través de cinco componentes de calidad: facilidad de aprendizaje, eficiencia, memorabilidad, baja tasa de errores y satisfacción (Rosenblatt F. , 1958). En el contexto de aplicaciones educativas, estos principios se traducen en que la interfaz debe ser intuitiva, permitiendo al usuario centrarse en el contenido de aprendizaje y no en descifrar cómo funciona la aplicación. El método estándar para evaluarla es mediante escalas de usabilidad, como el **System Usability Scale (SUS)** de Brooke (1996) (Brooke, 1996), que utiliza preguntas con escalas Likert para cuantificar la experiencia subjetiva del usuario.

2.2.6.6. Adquisición de datos

La adquisición de datos para sistemas de reconocimiento de lenguaje de señas se fundamenta en la teoría de visión por computadora aplicada a gestos dinámicos (Wang & Chen, 2020). Esta establece que la calidad del dataset depende críticamente de:

- La variabilidad articulatoria (configuraciones manuales, trayectorias de movimiento).
- Los parámetros espaciales (distancia cámara-sujeto, ángulos de captura).
- Las condiciones ambientales controladas (iluminación, fondos).

Según la teoría de muestreo para sistemas de reconocimiento gestual (Chen, Li, & Wang, 2021), se requiere:

- Mínimo 300-500 muestras por clase para garantizar representatividad estadística.

- Distribución balanceada de variaciones intra-clase (diferentes ejecutantes, velocidades.

2.2.6.7. MobileNet

MobileNet es una familia de arquitecturas de redes neuronales convolucionales (CNN) diseñadas específicamente para aplicaciones de visión por computadora en dispositivos móviles y sistemas embebidos con recursos limitados. Desarrollada por investigadores de Google (Howard, et al., 2017), su principal innovación radica en el uso de convoluciones profundas separables (depthwise separable convolutions), que permiten una reducción significativa en la complejidad computacional y el tamaño del modelo, manteniendo un buen rendimiento en tareas de clasificación y detección de objetos.

2.2.6.8. EfficientNet

EfficientNet es una familia de modelos de redes neuronales convolucionales (CNN) diseñados para maximizar la eficiencia en el balance precisión-velocidad. Desarrollado por Google Research (Tan & Le, 2019), introduce un método de escalado compuesto (compound scaling) que optimiza simultáneamente la profundidad, anchura y resolución de la red, logrando un rendimiento superior con menos parámetros comparado con arquitecturas tradicionales.

2.2.6.9. Aplicaciones de MobileNetV2 y EfficientNetB0 en reconocimiento de Lenguaje de señas

a) Aplicaciones de MobileNetV2 en contexto de bajos recursos

En el contexto peruano, donde los recursos computacionales pueden ser limitados (Organismo Supervisor de Inversión Privada en Telecomunicaciones, 2023), MobileNetV2 ha demostrado ser particularmente efectivo para aplicaciones de reconocimiento de señas en dispositivos móviles. (Wang & Chen, 2020) desarrollaron una CNN ligera basada en principios similares a MobileNetV2 que logró:

- El 95.2% de precisión en reconocimiento de señas estáticas.
- Tiempo de inferencia de 18ms en dispositivos Android de gama media.
- Consumo de memoria reducido en 65% comparado con arquitecturas convencionales.

b) Aplicaciones de EfficientNetB0 en reconocimiento de señas peruano

(Hussain et al., 2023) desarrollaron un sistema de reconocimiento de señas estáticas utilizando transfer learning con dos arquitecturas modernas, Inception-V3 y EfficientNet-B0. Los mejores resultados se obtuvieron con EfficientNet-B0, destacando:

- El 99.1 % de accuracy, superando ampliamente a modelos convolucionales previos
- El 98.9 % de precisión y 98.7 % de F1-score, evidenciando un reconocimiento robusto y estable

- Buena generalización incluso con un dataset moderado, gracias al fine-tuning sobre ImageNet
- Arquitectura altamente adecuada para tareas de reconocimiento de señas estáticas por su capacidad para capturar detalles finos de la mano

2.3. Definición de términos

2.3.1 Deep learning

El Deep Learning es un subconjunto del aprendizaje automático (machine learning), que a su vez forma parte del campo más amplio de la inteligencia artificial. En el Deep Learning, se utilizan redes neuronales, algoritmos inspirados en el funcionamiento del cerebro humano, para aprender y extraer patrones a partir de grandes volúmenes de datos. Estas redes neuronales profundas tienen la capacidad de realizar tareas complejas mediante el procesamiento jerárquico de la información.

2.3.2 Lenguaje de señas

La lengua de señas es una lengua natural que facilita la comunicación mediante la percepción visual y la expresión y configuración gesto-espacial. El uso de este sistema permite a las personas sordas establecer un canal de comunicación efectivo con su entorno social, ya sea con otras personas que dominen el idioma en cuestión o con otros hablantes de la lengua de señas.

2.3.3 CNN

Es una red neuronal artificial (ANN) que utiliza aprendizaje profundo para analizar imágenes, clasificar elementos visuales y realizar tareas de visión artificial.

2.3.4 Data augmentation

Técnica para generar nuevos datos a partir de una cantidad pequeña o grande de datos para poder reducir el overfitting.

2.3.5 Visión por computadora

El campo de la inteligencia artificial conocido como visión computacional se centra en facilitar que las máquinas interpreten y entiendan imágenes y vídeos.

2.3.6 Aplicativo móvil

Un software móvil es un programa creado para funcionar en dispositivos móviles como smartphones o tablets.

2.3.7 Reconocimiento de gestos

El reconocimiento de gestos es un método que facilita la identificación y categorización de movimientos o posturas concretas del cuerpo, principalmente las manos, con el fin de entender su sentido.

2.3.8 Overfitting

El overfitting sucede cuando un modelo de aprendizaje automático se adecúa excesivamente a los datos de entrenamiento, disminuyendo su habilidad para generalizar a nueva información.

2.3.9 Interfaz de usuario

El medio por el cual los usuarios interactúan con una aplicación o sistema es la interfaz de usuario.

CAPITULO III: METODOLOGÍA

3.1 Tipo y nivel de investigación

La presente investigación fue de tipo aplicada (Hernández, Fernández, & Baptista, 2014), ya que busco ofrecer una solución tecnológica a una problemática concreta: la falta de recursos interactivos y accesibles para el aprendizaje del Lenguaje de Señas Peruano (LSP). Se desarrolló una aplicación móvil didáctica que integró un modelo de reconocimiento de señas basado en deep learning, permitiendo a los usuarios aprender y practicar este lenguaje de forma interactiva. El nivel de investigación fue descriptivo, ya que se busco caracterizar el rendimiento del modelo de reconocimiento de señas y la usabilidad del aplicativo en función de métricas objetivas. El enfoque fue cuantitativo (Bernal, 2010), debido a que se utilizaron datos medibles y objetivos para evaluar el rendimiento del modelo de reconocimiento (precisión, exactitud), así como la usabilidad de la aplicación a través de métricas estandarizadas.

3.2 Diseño de investigación

La presente investigación adoptó un enfoque cuantitativo, aplicado, no experimental y de corte transversal (Arias, 2012), orientado al desarrollo e integración de un sistema basado en redes neuronales convolucionales (CNN) para el reconocimiento de señas del abecedario en Lengua de Señas Peruana (LSP), mediante una aplicación móvil con fines didácticos. La estrategia metodológica se estructuró en seis etapas secuenciales, alineadas con los objetivos planteados en el estudio. A continuación, se describen cada una de ellas:

Etapa 1 - Recolección de datos: Se elaboró un conjunto de datos propio, compuesto por imágenes del abecedario en Lengua de Señas Peruana. Las imágenes fueron capturadas en colaboración con usuarios voluntarios que realizaron las señas correspondientes, garantizando diversidad en iluminación, ángulos y características físicas. Estas imágenes fueron organizadas por clase (una carpeta por cada letra del abecedario) y etiquetadas manualmente para su posterior uso en el entrenamiento del modelo.

Etapa 2 - Preprocesamiento de datos: Los datos recolectados fueron sometidos a un proceso de limpieza y normalización. Esto incluyo el redimensionamiento de las imágenes a una escala uniforme y la transformación de los valores de píxeles para asegurar su compatibilidad con los algoritmos de aprendizaje profundo. El objetivo fue mejorar la calidad de los datos de entrada y reducir el impacto de variaciones innecesarias.

Etapa 3 - Implementación del modelo de aprendizaje profundo: Se entreno un modelo de red neuronal convolucional (CNN) para el reconocimiento de señas. Asimismo, se evaluaron dos enfoques de aprendizaje por transferencia, integrando y procesando las arquitecturas (MobileNet y EfficientNet), con el fin de comparar el rendimiento entre ambas. El entrenamiento se realizó utilizando un conjunto de entrenamiento y validación, y se aplicaron métricas estándar de evaluación.

Etapa 4 - Evaluación del rendimiento del modelo: Los modelos fueron evaluados en función de su precisión, exactitud, recuperación (recall) y puntuación F1, además de una matriz de confusión que permitió analizar los aciertos y errores por clase.

El modelo con mejor desempeño fue seleccionado para su integración en la aplicación móvil.

Etapa 5 - Desarrollo e integración de la aplicación móvil: Se desarrollo una aplicación móvil educativa para Android, que integro el modelo de reconocimiento seleccionado mediante TensorFlow Lite. La aplicación tuvo un enfoque didáctico e interactivo, incluyendo actividades complementarias como juegos, desafíos, práctica libre y retroalimentación inmediata, con el propósito de apoyar el aprendizaje del abecedario en LSP de manera lúdica y accesible.

Etapa 6 - Evaluación de usabilidad: La aplicación fue evaluada por un grupo de doce usuarios voluntarios mediante el System Usability Scale (SUS), un cuestionario estándar de diez ítems reconocido internacionalmente. Este instrumento está diseñado para medir aspectos clave de la experiencia del usuario, incluyendo la facilidad de uso, la claridad de la interfaz, la utilidad percibida y la satisfacción general. Los resultados se analizaron cuantitativamente para obtener una puntuación de 0 a 100 y se compararon con los benchmarks internacionales, sirviendo así para validar la experiencia de uso y el aporte educativo de la herramienta.

3.3. Población y muestra

3.3.1 Población

La población que se usó para el reconocimiento de señas peruano estuvo conformada por 25,000 registros, entre estas se encuentran imágenes de las diferentes señas del abecedario en el contexto peruano.

La población objetivo para este proyecto estuvo constituida por la totalidad de señas potenciales del alfabeto en el contexto de la Lengua de Señas Peruana (LSP). En términos de datos, esto se traduce en un universo teóricamente muy amplio de posibles variaciones (iluminación, fondos, ejecutantes, ángulos, etc.) de cada seña.

Población teórica (Población Objetivo)

- La totalidad de señas potenciales del alfabeto perteneciente a la Lengua de Señas Peruana (LSP).
- Todos los usuarios potenciales de la LSP en Perú, con su diversidad inherente (edad, género, estilo personal de seña, variaciones regionales).
- El conjunto infinito de posibles condiciones del mundo real (iluminaciones naturales y artificiales, fondos diversos, ángulos de cámara, resoluciones).
- En esencia, representa la variabilidad completa que el modelo de reconocimiento encontraría en una aplicación real.

Población estadística (Dataset Disponible)

- Está conformada por 25,000 registros específicos (imágenes)
- Incluye las diferentes señas del abecedario capturados bajo condiciones controladas.
- Representa una instancia específica y finita de la población teórica, limitada por los recursos, tiempo y acceso a colaboradores.
- Constituye la base tangible sobre la cual se entrena y evalúa el modelo de reconocimiento.

Población de la evaluación de usabilidad

Para la evaluación de la experiencia de usuario y usabilidad de la aplicación móvil SignalGo, la población objetivo estuvo constituida por personas interesadas en aprender Lengua de Señas Peruana (LSP), tales como estudiantes y usuarios sin experiencia previa, pero con interés en esta forma de comunicación.

Se consideraron personas entre los 18 y 30 años con habilidades tecnológicas básicas, lo que representa el perfil de usuario real esperado para la aplicación.

3.3.2 Muestra

Dado que no existía un dataset público y suficientemente amplio de LSP que satisficiera las necesidades del proyecto, se optó por la creación de un dataset propio. La estrategia de muestreo empleada fue de tipo no probabilístico intencional, específicamente un muestreo por cuotas. Esta elección se basó en la necesidad de capturar de manera controlada y sistemática las variaciones clave que afectan el reconocimiento de señas.

El tamaño del dataset creado, de 25,000 registros entre imágenes y videos, se justifica en base a los siguientes criterios:

Complejidad del problema: El reconocimiento de señas es una tarea de Visión por Computadora compleja que implica desde la clasificación de posturas estáticas (abecedario), hasta la interpretación de movimientos dinámicos (gestos y frases). Los modelos de Deep Learning requieren grandes volúmenes de datos para aprender estas representaciones espaciotemporales de manera efectiva y generalizar correctamente, evitando el sobreajuste (overfitting).

Cobertura de variabilidad: El número de 25,000 registros permite una distribución suficiente de ejemplos por categoría. Por ejemplo, si se considera el alfabeto (28 letras) y un conjunto de frases, este volumen asegura cientos o miles de ejemplos por seña, lo cual es crucial para cubrir las variaciones entre ejecutantes, iluminación y fondos definidas en la estrategia de muestreo.

Recursos y viabilidad: Si bien un dataset más grande podría ser ideal, 25,000 registros representan un equilibrio entre la necesidad de datos suficientes para entrenar un modelo robusto y la viabilidad práctica en términos de costos de almacenamiento, tiempo de captura, etiquetado y capacidad de procesamiento durante el entrenamiento.

Benchmarks con datasets existentes: Este tamaño es comparable o superior a otros datasets pioneros en reconocimiento de señas de otros idiomas, lo que sugiere que es un punto de partida sólido para la tarea propuesta.

Asimismo, para la evaluación de usabilidad de la aplicación SignalGo, se utilizó una muestra conformada por doce usuarios voluntarios.

3.4. Instrumentos

3.4.1. Python

Python es el lenguaje de programación más empleado en el campo del Deep Learning gracias a su estructura simple y a la amplia gama de bibliotecas y frameworks existentes. Será el idioma clave para elaborar el modelo de identificación de lenguaje de señas.

3.4.2. Tensorflow/Keras

TensorFlow es un marco de aprendizaje profundo muy empleado para la creación y entrenamiento de modelos de redes neuronales. Keras, un componente de TensorFlow, es perfecto para la creación rápida y eficaz de arquitecturas de redes neuronales convolucionales (CNN). Será beneficioso para adiestrar el modelo de identificación de gestos.

3.4.3. Android studio

Android Studio es el IDE oficial para desarrollar aplicaciones para móviles en Android. Será el instrumento clave para la creación de la app móvil que incorporará el modelo de aprendizaje profundo para identificar lenguaje de señas.

3.4.4. Redes neuronales convolucionales

Las CNN son esenciales para el procesamiento de imágenes y videos, dado que facilitan la extracción eficaz de atributos visuales. Algunas estructuras que podrían emplear incluyen:

- **ResNet18**: Perfecto para labores de categorización de imágenes, manteniendo un balance adecuado entre exactitud y eficacia.
- **DenseNet**: Facilita vínculos sólidos entre capas, lo que optimiza el intercambio de información y el desempeño en labores de reconocimiento.
- **MobileNet**: Es una alternativa ligera y eficaz para incorporar en aplicaciones móviles.
- **EfficientNet**: Combina escalamiento de profundidad, ancho y resolución para lograr alta precisión con bajo costo computacional.

3.4.5. Métricas de evaluación

Para valorar el desempeño del modelo de reconocimiento de lenguaje de señas, se emplearán indicadores tales como:

- **Accuracy**: Porcentaje de predicciones acertadas sobre el total de predicciones realizadas.
- **Precisión (Precision)**: Proporción de predicciones positivas que realmente corresponden a casos positivos.
- **Recall**: Habilidad del modelo para reconocer de manera adecuada los casos positivos existentes.
- **F1-score**: Media armónica entre precisión y recall, eficaz para medir el balance entre ambas métricas, especialmente útil en contextos con clases desbalanceadas.
- **Matriz de Confusión**: Herramienta visual que permite observar el rendimiento del modelo en cada clase individual, facilitando la identificación de errores específicos y de aquellas señas que el modelo tiende a confundir con mayor frecuencia.

3.4.6. Jupyter notebooks

Jupyter Notebooks es un recurso interactivo que facilita la unión de código, texto y visualizaciones en un único documento. Será beneficioso para probar el modelo, visualizar la información y registrar el proceso de desarrollo.

3.4.7. OpenCV

OpenCV es una librería de visión computacional que ofrece instrumentos para el procesamiento de imágenes y vídeos. Para preprocesar los datos de entrada (gestos de lenguaje de señas) antes de trasladarlos al modelo de Deep Learning, resultará crucial.

3.4.8. Tensorflow lite

TensorFlow Lite es una versión ligera de TensorFlow, creada para aparatos móviles. Facilitará la incorporación eficaz del modelo de Deep Learning entrenado en la aplicación móvil.

3.5 Obtención y Procesamiento de Datos

3.5.1 Recolección y organización de datos

Para asegurar la correcta funcionalidad del sistema de reconocimiento de señas en cuestión de precisión y representación adecuada se elaboró un dataset específico personalizado del Lenguaje de Señas Peruano (LSP).

El proceso de adquisición consistió en grabaciones de varios voluntarios ejecutando cada seña o clase necesaria para el modelo, lo cual corresponden las letras desde la “A” a la Y, todo esto se hizo bajo diversas condiciones como la variación de iluminación, diversos fondos y ángulos. Esto permitió que el modelo aprendiera de diversos entornos reales.

Las condiciones ambientales se mantuvieron uniformes durante toda la captura, con iluminación controlada entre 200-300 lux utilizando luz natural indirecta o artificial

uniforme, fondos neutros en color gris o blanco, distancia constante de 40-60 cm entre la cámara y la mano. En cuanto a los participantes, se contó con 6 voluntarios de género mixto (tres hombres y tres mujeres) en el rango de edad de 18 a 25 años, con diversa experiencia en LSP.

Se tomaron 25000 muestreos o datos, obtenidos de imágenes y videos, los cuales se distribuyen entre las 24 clases (letras) necesarias. Cada registro o dato fue etiquetado cuidadosamente para conservar la integridad y eficiencia del modelo. Finalmente, los datos fueron organizados en carpetas divididos por clases y se tuvo un estándar en resolución de 720X1280 píxeles, de esa manera facilita su posterior procesamiento.

3.5.2 Preprocesamiento de datos

Para pasar a la etapa de entrenamiento es necesario que el conjunto de datos pase por un preprocesamiento para asegurar y optimizar su calidad, así como su compatibilidad con los modelos de deep learning a utilizar. En la **Figura 1** se presenta el flujo general del proceso de datos implementado, que abarca desde la captura y organización inicial hasta el preprocesamiento y la fase de entrenamiento del modelo

Limpieza de datos

- Se hizo el descarte y eliminación de imágenes no nítidas o con baja resolución, gestos no apreciables correctamente o con etiquetas erróneas.
- No se consideraron aquellas imágenes con factores que afectan la correcta visualización de las señas o tenían una iluminación muy alta.

Durante esta etapa de depuración se eliminaron aproximadamente 315 muestras (1.26 % del total), debido a problemas de calidad visual o etiquetado incorrecto.

La cantidad final de datos válidos utilizados fue de 25000 imágenes, manteniendo una representación equilibrada entre todas las clases.

El bajo porcentaje de pérdida evidencia un control adecuado del dataset y garantiza que la calidad y diversidad de las muestras no se vean comprometidas.

Redimensionamiento y normalización

- Las imágenes obtenidas fueron ajustadas al tamaño de 96x96 píxeles, lo cual era adecuado para el modelo a ser entrenado, la eficiencia computacional y el contexto final de uso que era para dispositivos móviles.
- Se hizo la conversión del formato BGR, el cual es usado por OpenCV a RGB usado para Tensor Flow/Keras.
- Se realizó la normalización de píxeles de [0, 255] a [0,1] para de ese modo mejorar la convergencia con el modelo.

Estructuración y balanceo

- Las imágenes fueron almacenadas como arrays Numpy, usando las etiquetas numéricas por letra (0=A, 24=Y).
- La distribución entre clases se mantuvo equilibrada para que no hubiera sesgos en el aprendizaje del modelo.

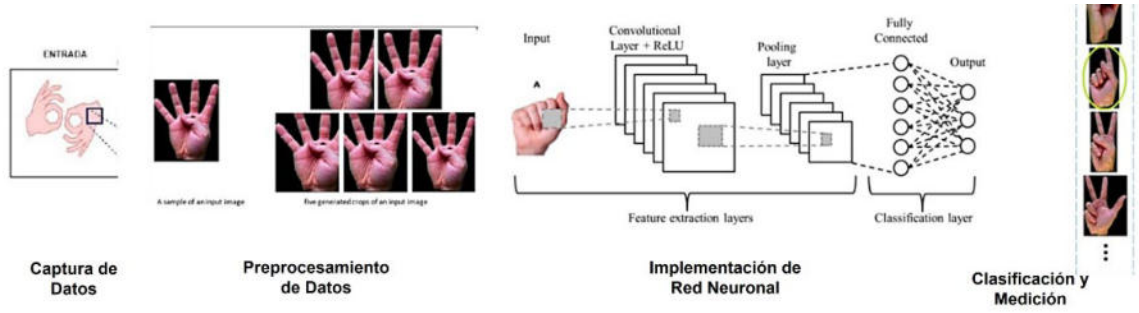
División del dataset

El conjunto de datos fue dividido en subconjuntos, siguiendo proporciones estándar comúnmente utilizadas en aprendizaje automático:

- 70% para entrenamiento, donde el modelo va aprendiendo los patrones.
- 15% para validación, que permite ajustar hiperparámetros y hacer un monitoreo del rendimiento del modelo durante la fase de entrenamiento.
- 15% para prueba, lo cual es de uso exclusivo para la hora de evaluar el modelo.

Figura 1

Presentación y análisis de datos



La separación fue realizada de manera directa, respetando las proporciones mencionadas y asegurando una distribución adecuada entre los subconjuntos. Esta división permite separar los datos usados en la fase de entrenamiento y evaluación, de ese modo se logra medir con mayor precisión el desempeño del modelo frente a datos nuevos. Para futuras iteraciones o entrenamientos se hace la recomendación de usar técnicas de aumento de datos (data augmentation), con el fin de mejorar la robustez que tiene el modelo.

3.5.3 Distribución del dataset

Para la implementación del modelo de clasificación de lenguaje de señas peruano, se ha utilizado un conjunto de datos de adquisición propia, el cual está compuesto de 24

clases, que corresponden a las letras del alfabeto. El dataset estuvo compuesto por 25000 imágenes y se distribuye de la siguiente manera: 1200 para la letra A, 1340 para B, 1150 para C, 1006 para D, 1001 para E, 1005 para F, 996 para G, 1002 para H, 997 para I, 1000 para K, 1089 para L, 1003 para M, 1002 para N, 995 para O, 1070 para P, 997 para Q, 1040 para R, 1001 para S, 1003 para T, 999 para U, 1002 para V, 1039 para W, 1003 para X, y 1060 para la letra Y.

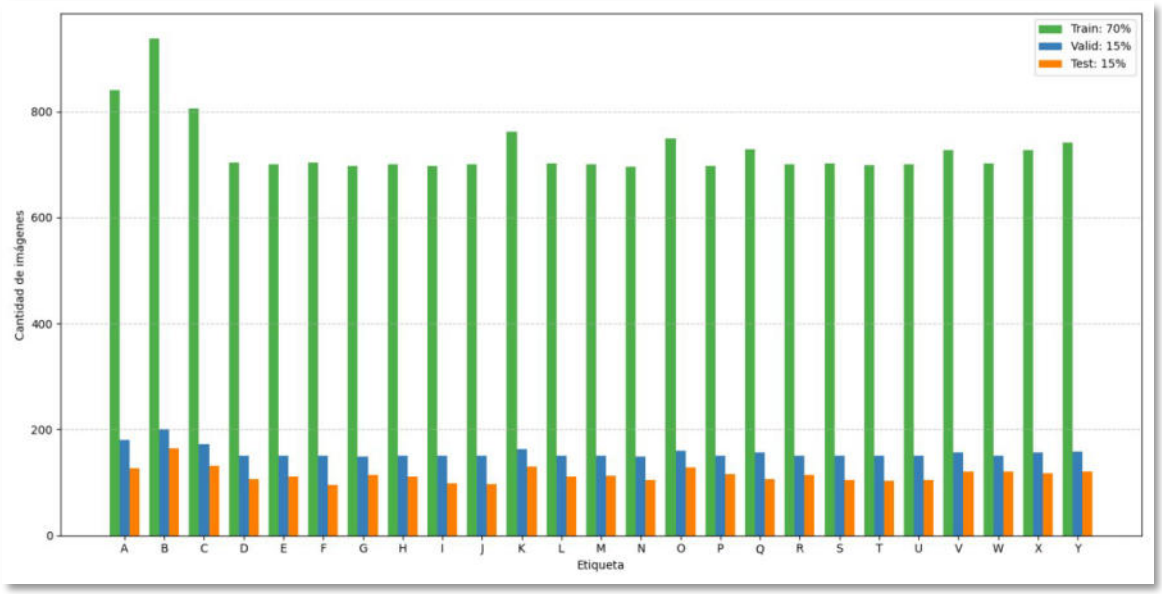
Las imágenes fueron sido redimensionadas a 96x96 pixeles y convertidas a formato de color RGB, para unificar su formato y tamaño, además que de esta manera se pueden adaptar a las arquitecturas de redes neuronales convolucionales propuestas a utilizar.

Posteriormente, los datos se han dividido en tres subconjuntos, dando así la proporción de 70% utilizado para entrenamiento, 15% para la validación y 15% para prueba, asegurando el balance de representación equitativa de todas las clases para cada conjunto de datos.

En la **Figura 2** se aprecia la distribución existente de imágenes por cada clase, esto corresponde las 24 letras del alfabeto en el lenguaje de señas. Se puede observar, que la cantidad de muestras es uniforme y no presenta un desbalance significativo entre las diferentes clases o letras. Debido a esta distribución que se nota equilibrada, no ha sido necesario la implementación de técnicas de pesos por clase (class weights) durante la fase de entrenamiento. Esto permitió utilizar la función de pérdida estándar sin utilizar algunos ajustes adicionales, debido a que se contaba con una representación adecuada por clase en el conjunto de datos usado en el entrenamiento, esto favorece en el aprendizaje homogéneo del modelo.

Figura 2

Distribución del Dataset



El ajuste de las dimensiones de las imágenes del conjunto de datos a 96x96 píxeles fue necesario debido a que el tamaño original de las muestras no era el adecuado para ser procesados por las arquitecturas convoluciones que se tenía como **EfficientNetB0** y **MobileNetV2**. Aunque están diseñadas para ser ligeras, se requieren dimensiones apropiadas en las imágenes de entrada para así garantizar su flujo adecuado cuando pasen por las capas profundas de convolución y polling, en donde sus dimensiones son reducidas progresivamente.

Se optó por el uso del optimizador Adam en la fase de entrenamiento debido a su capacidad de ajuste adaptativo de la tasa de aprendizaje, esto le permite ser eficiente frente a otros algoritmos como SGD, en especial en tareas de clasificación multiclases como es el caso del reconocimiento de letras en el lenguaje de señas.

3.6 Desarrollo del Modelo e Implementación de la App

Después de realizado el preprocesamiento de los datos, se hizo la implementación de una red neuronal convolucional (CNN) para el reconocimiento del Lenguaje de Señas Peruano (LSP). Debido al tamaño del dataset a usar y la complejidad de la tarea, se ha optado por utilizar Transfer Learning, de este modo se aprovechó un modelo pre entrenado para poder mejorar la eficiencia y precisión del sistema a implementar.

3.6.1 Selección de modelos base: EfficientNetB0 y MobileNetV2

Para realizar el reconocimiento de las señas, se ha optado por la implantación y comparación de dos arquitecturas de deep learning muy usadas en visión por computadora: EfficientNetB0 y MobileNetV2. Estas fueron seleccionadas por el equilibrio existente entre precisión, eficiencia computacional y su compatibilidad con dispositivos móviles, debido que este modelo será desplegado finalmente en una aplicación móvil.

3.6.1.1 . EfficientNetB0

Se selecciono esta arquitectura debido al rendimiento destacado que posee para tareas de clasificación de imágenes, su desempeño combina profundidad, ancho y resolución de una manera eficiente.

3.6.1.2. MobileNetV2

La segunda arquitectura incorporada fue MobileNetV2 para hacer una comparación de rendimiento frente a EfficienteNetB0, debido a su diseño ligero y óptimo funcionamiento que tiene para dispositivos con recursos un poco limitados.

3.6.2 Configuración del entrenamiento

Ambas arquitecturas fueron adaptadas para ajustarse a la tarea específica de reconocimiento de señas. En ambos casos, se eliminaron las capas finales originales para incorporar una salida personalizada y se añadieron las siguientes capas:

Ambos modelos fueron entrenados bajo una configuración base unificada para garantizar una comparación justa. La configuración inicial fue la siguiente:

- Optimizador: Adam, por su capacidad adaptativa y eficiencia en clasificación.
- Función de pérdida: sparse_categorical_crossentropy, adecuada para clasificación multiclase con etiquetas enteras.
- Épocas: 20.
- Tamaño de lote (batch size): 16
- Hardware: Se empleó una GPU para acelerar el proceso de entrenamiento.

Como parte de la configuración del entrenamiento, se incluyeron dos callbacks esenciales:

- ModelCheckpoint: encargado de guardar automáticamente los pesos del modelo cada vez que mejoraba el rendimiento sobre el conjunto de validación, asegurando conservar la mejor versión del modelo.
- ReduceLROnPlateau: empleado para disminuir la tasa de aprendizaje cuando no se observaban mejoras, facilitando así una convergencia más estable en las últimas épocas.

3.6.2.1 . Entrenamiento y comparación

Cuando se ha finalizado el entrenamiento, los modelos utilizados fueron evaluados de manera comparativo en términos de:

- Precisión (precision)
- Recall
- F1-score
- Matriz de confusión
- Exactitud (accuracy)

El propósito es poder seleccionar la arquitectura con mejor rendimiento para así integrarlo en la aplicación móvil de reconocimiento de señas en tiempo real.

3.6.3 Transfer learning

La aplicación de la técnica de Transfer Learning permite optimizar el desarrollo del modelo, así se evita entrenar una red neuronal desde cero. La estrategia consiste en usar un modelo pre entrenado para una tarea general (como clasificación de imágenes en el dataset imageNet); aprovechar esto y utilizarlo para una tarea específica como es el caso del reconocimiento de señas del alfabeto del lenguaje de Señas Peruano (LSP). Esta técnica permite aprovechar las características que ya tienen aprendidas los modelos robustos, particularmente en las primeras capas, que es donde suelen identificar patrones visuales básicos como los bordes, las texturas y las formas. De este modo, el modelo no necesita aprender todo ello desde cero, sino que ya parte con una base sólida.

Modelos base utilizados:

- EfficientNetB0
- MobileNetV2

Para el proceso de adaptación mediante el uso de Transfer Learning, se utilizaron los pesos preentrenados con ImageNet, para los modelos EfficientNetB0 y MobileNetV2,

de este modo se aprovecha el conocimiento previo ya adquirido en estas primeras capas internas. Después se fue añadiendo las capas personalizadas, como GlobalAveragePooling2D para reducir la dimensionalidad, Dropout(0.3) para que se sobreajuste y la capa dense con 24 unidades y de activación softmax para de ese modo clasificar las letras del alfabeto del lenguaje de Señas Peruano. Es así que el entrenamiento de estas capas superior le permitió al modelo adaptarse progresivamente al nuevo contexto sin perder o desaprovechar lo ya aprendido en ImageNet.

3.6.4 Evaluación de los hiperparametros

Las métricas de evaluación usadas para la construcción del modelo fueron la función de perdida sparse_categorical_crossentropy y la métrica de exactitud (accuracy); ambos son apropiadas para casos o problemas de clasificación donde hay varias clases balanceadas.

Para determinar los hiperparametros que mejor se ajusten, se han realizado múltiples pruebas en donde se varía la tasa de aprendizaje (learning rate), del mismo modo el tamaño de lote (batch size) también se varió para encontrar el mejor ajuste posible. Estas pruebas se realizaron con ambas arquitecturas propuestas, para analizar el impacto que tiene cada configuración en el rendimiento final del modelo.

En la **Figura 3** se muestran los resultados que se han obtenido para distintos valores de learning rate con EfficientNetB0, y tambien los valores correspondientes al manejo de distintos bath size.

Además, se puede observar el comportamiento que tiene el modelo frente a valores distintos de learning rate como: 0.1, 0.01, 0.001, 0.0001 y 1e-5. Se aprecia que

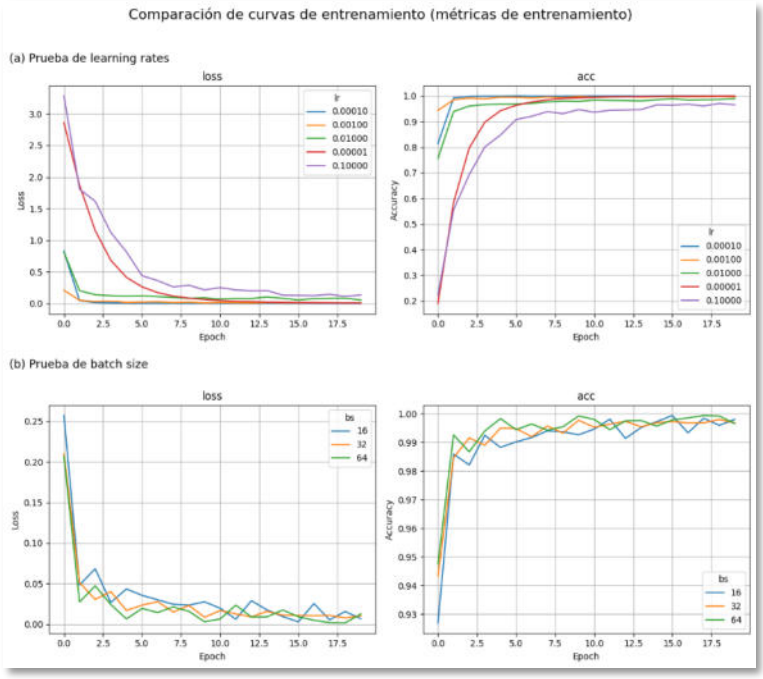
con valores altos de learning rate como 0.1, no converge de buena manera, dado que muestra una alta pérdida (loss) inicial y bajo rendimiento en términos de accuracy.

Contrastando con los valores medios y bajos; como 0.001 y 0.0001, estos muestran un mejor resultado y comportamiento, donde la perdida disminuye rápidamente y el accuracy alcanza valores altos que superan el 98% en pocas épocas. Es así que el valor de 0.0001 se define como el valor más balanceado, dado que alcanza una rápida convergencia estable, lo cual es una de los mejores hiperparametros para el entrenamiento final.

En la **Figura 3**, se presentan los resultados obtenidos tras el entrenamiento del modelo con diferentes tamaños de batch, correspondiente a 16,32,64. Se aprecia que en términos generales todos los abc sizes logran obtener una convergencia adecuada, con perdida (loss) que tienden a 0 y presiones altas cercas a 99%. Sin embargo, el batch size de 16, tiene una curva de loss más suave y muestra la precisión más estable a lo largo de las épocas de entrenamiento, lo cual se interpreta como una mejor capacidad para la generalización. Por otro lado, el batch size de 64, si bien se muestra efectivo, presenta algunas funciones que pueden afectar la estabilidad en contextos de prueba más ruidosos.

Figura 3

Resultados obtenidos con diferentes valores de Learning Rate y Batch Size para EfficientNetB0



Mientras que en la **Figura 4** se muestran los resultados obtenidos para distintos valores de learning rate para MobileNet y también los valores correspondientes a variaciones de batch size.

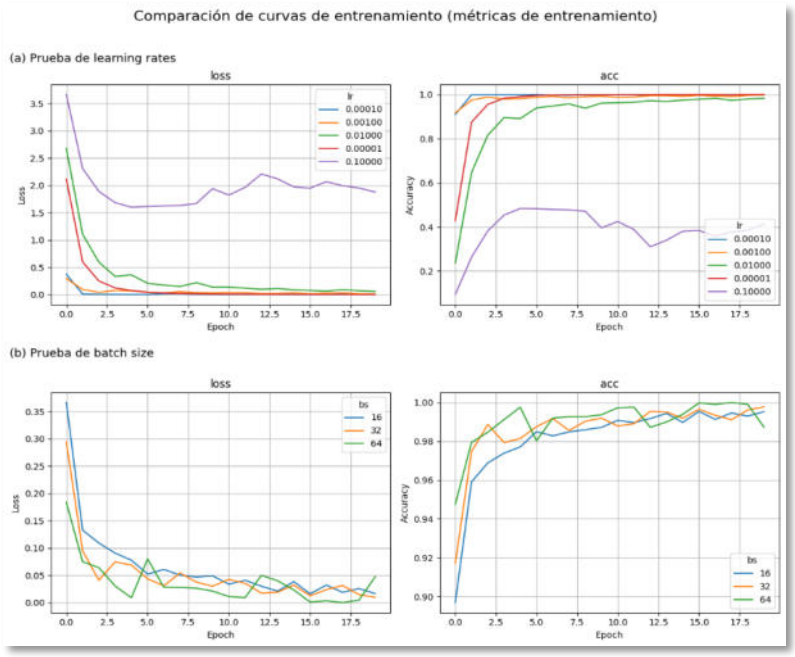
En la **Figura 4**, se observa el rendimiento del modelo MobilenetV2 con distintos valores de Learning Rate como: 0.1, 0.01, 0.001, 0.0001 y 1e-5. La grafica muestra que el valor de 0.1 genera una curva de perdida notable y un rendimiento deficiente en cuantos accuracy. Por el contrario, frente a valores intermedios como 0.001 y 0.0001 presenta una mejor convergencia, evidenciando un aprendizaje más efectivo dado que hay una disminución rápida de Loss, y alcanza valores de accuracy por encima de 98% en las primeras épocas. Por este motivo, el learning rate de 0.0001 al mostrar una mejor

precisión y estabilidad, es considerado el valor más adecuado como configuración para ser usado finalmente en el modelo MobileNetV2.

Además, se presentan los resultados que se obtuvieron tras la prueba con distintos batch sizes, valores de 16,32 y 64. EN general todos los tamaños permiten convergen al modelo de manera adecuada, alcanzando así valores superiores al 98%. Sin embargo, el batch size de 16, provee una mejor curva de loss, siendo más estable y con menor fluctuación en las últimas épocas de la etapa de entrenamiento. Esto sugiere que en el modelo MobilenetV2, un batch size mayor contribuye a la estabilidad del modelo sin comprometer su rendimiento. Es así que se opta por usar un batch size de 16 y learning rate de 0.0001, esta configuración se define dado al buen equilibrio entre rendimiento y comportamiento en la fase de entrenamiento.

Figura 4

Resultados obtenidos con diferentes valores de Learning Rate y Batch Size para MobileNet



Las Tabla 1 y Tabla 2 presentan las configuraciones de hiperparametros que se evaluaron durante el entrenamiento de los modelos basados en las arquitecturas EfficientNetB0 y MobileNetV2, bajo el criterio de evaluación de los valores obtenidos sobre el conjunto de datos de prueba. Para cada modelo se utilizaron diferentes combinaciones de la tasa de aprendizaje (learning rate) y tamaño de lote (batch size). Se uso el optimizado Adam y transferencia de conocimiento mediante pesos pre entrenados en ImageNet. En ambos casos las carpetas del modelo se han mantenido sin congelar para permitir asi, su adaptación completa al nuevo dominio que es la clasificación de señas. Se aprecia que los mejores resultados de cada arquitectura usada, se han obtenido al usar valores de learning rate como 0.001 o 0.0001 juntos con batch sizes pequeños o intermedios, de esta manera se logró valores de accuracy en la etapa de validación superiores al 92%.

Para el caso de EfficientNetB0, la mejor configuración usada para alcanzar un buen rendimiento fue con un learning rate de 0.0001 y batch size de 32, logrando de esta manera una precisión del 92.1% y una perdida (loss) en validación de 0.3821, como se aprecia en la **Tabla 1**. En esta figura se puede observar la evolución de las métricas de *loss* y *accuracy* tanto en entrenamiento como en validación. El modelo mostró una curva de pérdida decreciente y estable, sin indicios de sobreajuste, y una progresión sostenida en la precisión, lo cual indica una adecuada capacidad de generalización.

En el otro caso, MobileNetV2, la mejor configuración se alcanzó con el uso de un learning rate de 0.0001 y un batch size de 16, obteniendo así una precisión de 93.2% y una pérdida de 0.3977, como se muestra en la Tabla 2 correspondiente. Aunque en este modelo se observan ciertas oscilaciones en la curva de validación, especialmente en la

métrica de precisión, el modelo logró mantener un rendimiento efectivo y estable a lo largo de las 20 épocas, con una diferencia leve entre el entrenamiento y la validación.

Tabla 1

Tabla de Resultados con EfficienNetB0

Arquitectura	Épocas	Tasa de aprendizaje	Tamaño de lote	Optimizador	Pesos	Freeze	Pérdida (Loss)	Exactitud (Acc.)
EfficientNetB0	20	0.0001	32	Adam	ImageNet	No	0.3821	0.921
EfficientNetB0	20	0.0001	16	Adam	ImageNet	No	0.3046	0.9154
EfficientNetB0	20	0.0001	64	Adam	ImageNet	No	0.3994	0.913
EfficientNetB0	20	0.01	64	Adam	ImageNet	No	0.625	0.9009
EfficientNetB0	20	1e-05	16	Adam	ImageNet	No	0.4056	0.8961
EfficientNetB0	20	1e-05	32	Adam	ImageNet	No	0.3737	0.8888
EfficientNetB0	20	1e-05	64	Adam	ImageNet	No	0.74	0.8598
EfficientNetB0	20	0.001	16	Adam	ImageNet	No	0.7624	0.8544
EfficientNetB0	20	0.001	16	Adam	ImageNet	No	2.0946	0.8447
EfficientNetB0	20	1e-05	64	Adam	ImageNet	No	0.5172	0.8429
EfficientNetB0	20	0.001	32	Adam	ImageNet	No	0.8121	0.8286
EfficientNetB0	20	0.01	32	Adam	ImageNet	No	2.1856	0.8098
EfficientNetB0	20	0.1	16	Adam	ImageNet	No	2.0652	0.711
EfficientNetB0	20	0.1	64	Adam	ImageNet	No	4.6975	0.6516
EfficientNetB0	20	0.1	32	Adam	ImageNet	No	3.8947	0.6009

Tabla 2

Tabla de Resultados con MobileNetV2

Arquitectura	Épocas	Tasa de aprendizaje	Tamaño de lote	Optimizador	Pesos	Freeze	Pérdida (Loss)	Exactitud (Acc.)
MobileNetV2	20	0.0001	16	Adam	ImageNet	No	0.3977	0.9323
MobileNetV2	20	1e-05	16	Adam	ImageNet	No	0.2914	0.92
MobileNetV2	20	0.0001	32	Adam	ImageNet	No	0.5469	0.8982
MobileNetV2	20	1e-05	32	Adam	ImageNet	No	0.3841	0.8963

MobileNetV2	20	0.0001	64	Adam	ImageNet	No	0.6233	0.8791
MobileNetV2	20	0.0001	32	Adam	ImageNet	No	0.9269	0.8721
MobileNetV2	20	0.0001	16	Adam	ImageNet	No	0.7935	0.8678
MobileNetV2	20	1e-05	64	Adam	ImageNet	No	0.561	0.819
MobileNetV2	20	0.001	64	Adam	ImageNet	No	6.3768	0.5501
MobileNetV2	20	0.01	32	Adam	ImageNet	No	8.9544	0.4217
MobileNetV2	20	0.01	16	Adam	ImageNet	No	10.9399	0.383
MobileNetV2	20	0.01	64	Adam	ImageNet	No	33.3381	0.1923
MobileNetV2	20	0.1	32	Adam	ImageNet	No	4.6239	0.0551
MobileNetV2	20	0.1	16	Adam	ImageNet	No	3.2076	0.0414
MobileNetV2	20	0.1	64	Adam	ImageNet	No	24.6602	0.0301

En las **figuras 5 y 6** se presentan las curvas de pérdida (loss) y precisión (accuracy) correspondientes al proceso de entrenamiento y validación de las arquitecturas MobileNetV2 y EfficientNetB0, respectivamente. Estas gráficas permiten analizar el comportamiento del aprendizaje de cada modelo y evaluar su estabilidad a lo largo de las épocas.

En ambas arquitecturas puestas a prueba, se confirma el impacto positivo de usar Transfer Learning con pesos de ImageNet, dado que esto permite alcanzar niveles altos de precisión con un número de épocas limitado y sin requerir una infraestructura tecnológica de alto costo. Asimismo, se resalta lo visto en las gráficas durante el entrenamiento correspondiente a estabilidad, donde EfficientNetB0 muestra curvas más suaves y regulares, debido a su diseño más optimizado; mientras que MobileNetV2, si bien logra una mayor precisión, presenta ciertas fluctuaciones en sus métricas de validación, lo cual es algo importante a considerar si en caso se desea implementar en contextos más sensibles donde existe una variabilidad de entrada cambiante.

Figura 5

Métricas de Entrenamiento y Evaluación con MobileNetV2

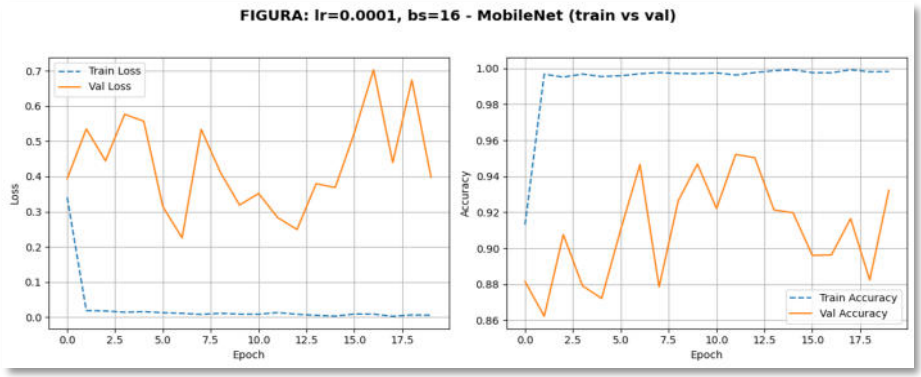
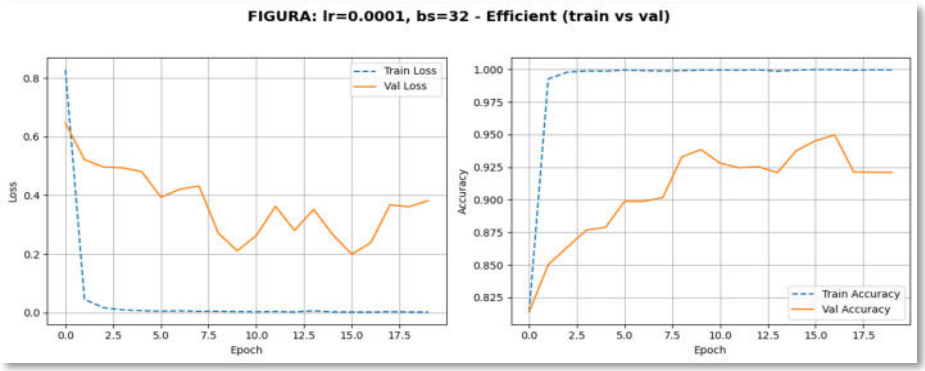


Figura 6

Métricas de Entrenamiento y Evaluación con EfficientNetB0



3.6.5 Evaluación de los modelos

Después del proceso de entrenamiento, se procedió a evaluar ambos modelos utilizados, para esto se necesitó el conjunto de datos de prueba, el cual corresponde al 15% del total, este no fue utilizado durante la fase de aprendizaje. Esta etapa es muy relevante para poder validar la capacidad de generalización de los modelos entrenados y así comprobar o validar su rendimiento frente a datos no vistos previamente durante el entrenamiento.

Para este efecto, se emplearon métricas utilizadas muy comúnmente para tareas de clasificación que son multiclase, las cuales son accuracy, precision, recall, f1-score y la matriz de confusión. Las métricas se calcularon a partir de las predicciones hechas sobre el conjunto de datos de X_Test comparándolas con las etiquetas que son reales (Y_Test).

3.6.6 Evaluación del funcionamiento de los modelos en tiempo real

A parte de evaluar el modelo con los datos de prueba o ya almacenados, se han realizado pruebas en tiempo real para ver su desempeño en el contexto que necesitamos, bajo condiciones reales a los que estará expuesto la app móvil realizada, esto es de suma importancia para garantizar que el modelo funcione correctamente en su entorno final.

3.6.6.1 Pruebas en tiempo real

El modelo entrenado fue integrado en la interfaz donde finalmente iba a desempeñar, la cual utiliza la cámara como dispositivo de captura de gestos en vivo. En estas pruebas se evaluaron aspectos como:

- **Tiempo de respuesta del modelo:** Se midió la rapidez que tenía el modelo para poder realizar una predicción una vez capturado el gesto o imagen en tiempo real.
- **Estabilidad en la predicción:** Se validó si el modelo tenía consistencia en mostrar la predicción (respuesta), cuando un gesto se repite múltiples veces en diferentes formas.
- **Sensibilidad a condiciones externas:** Se ha probado los modelos en distintos escenarios propuestos donde han habido cambios de iluminación,

diferentes fondos, diversas distancias de la cámara, con el objetivo de verificar la robustez del mismo.

3.6.7 Integración del modelo en la aplicación móvil

Después de haberse entrenado y validado el modelo para el reconocimiento de señas, se procedió con la etapa de integración dentro de la aplicación móvil que se desarrolló en Android Studio, con el objetivo de ofrecer una solución tecnológica didáctica en el aprendizaje del lenguaje de señas peruano.

a) Conversión del modelo a TensorFlow Lite

El modelo resultante final, el cual fue desarrollado con Tensorflow y Keras, fue convertido a TensorFlow Lite (tflite), que está optimizado para uso final en dispositivos móviles. Esta conversión hace posible reducir de gran manera el tamaño del archivo del modelo y mejorar su rendimiento, especialmente en el punto que respecta al tiempo de inferencia, el cual es la velocidad de respuesta.

b) Integración del modelo en la actividad

Para utilizar el modelo.tflite dentro de la activity principal en la aplicación realizada, se integró la librería TensorFlow Lite para Android. La cual nos permite cargar el modelo para poder realizar las predicciones desde el dispositivo móvil, sin necesidad de alguna conexión externa de internet.

El flujo de trabajo con el modelo es:

- La cámara captura la imagen
- Se redimensiona la imagen a **96x96 píxeles**, al igual que en el entrenamiento.

- Se hace una normalización de los valores de píxeles.
- El modelo cargado ejecuta las predicciones.
- Se interpreta el resultado y es mostrado al usuario final.

Desarrollo de la aplicación en Android Studio

Se desarrolló una aplicación de nombre SignalGo, la cual su construcción fue realizada a través de Android Studio, se empleó el lenguaje de Java para tal efecto y XML para la construcción de las distintas interfaces gráficas. El diseño fue enfocado principalmente en brindar una experiencia de usuario amigable, didáctica; que pueda adaptarse a personas principiantes que buscan aprender o reforzar sus conocimientos en lenguaje de señas peruano.

De manera estructural, la aplicación se organiza en base a la actividad principal (MainActivity) la cual actúa como menú de navegación y muestra las distintas actividades que ofrece la app. En la pantalla inicial se presentan cinco botones; la cual hace referencia a una funcionalidad específica de este sistema. En la **Figura 7** se muestra el entorno de desarrollo en Android Studio, donde se diseñó la interfaz principal del aplicativo.

Por otro lado, en la **Figura 8** se presenta la interfaz final del aplicativo móvil SignalGo, donde se aprecian los cinco botones principales correspondientes a las funcionalidades: Alfabeto, Ponte a prueba, Juegos, Insignias y Diccionario.

Figura 7

Diseño de la Interfaz en Android Studio

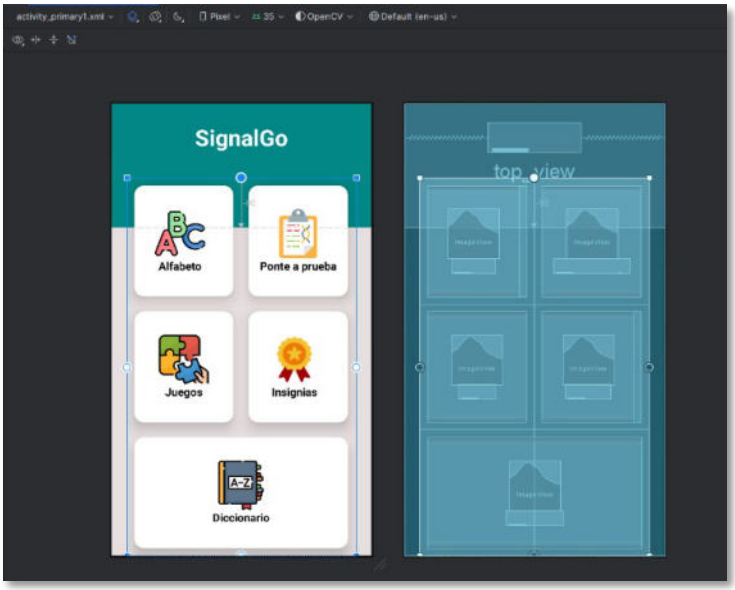


Figura 8

Interfaz del aplicativo móvil



Actividad: Alfabeto interactivo

En esta sección, se presenta un tipo de minijuego de selección múltiple en la que el usuario visualiza un video corto donde se presenta o muestra una seña (puede ser palabra o letra); asimismo, se le da tres opciones o botones para que el elija cual corresponde a lo visto. Las respuestas correctas otorgan puntaje, esto ayuda a promover la repetición y refuerzo de lo aprendido; así como el sistema de puntuación anima al usuario a intentar ganar los mayores puntos posibles.

En la Figura 9 se muestra el diseño de la interfaz del minijuego desarrollado en Android Studio, donde se definieron los elementos visuales y de interacción, tales como los botones de respuesta, las áreas de video y las etiquetas correspondientes.

Por su parte, en la **Figura 10** se presenta la interfaz final del minijuego ejecutándose en el dispositivo móvil, en la cual el usuario puede visualizar el video de la seña y seleccionar la opción que considere correcta según lo observado.

La información que alimenta el funcionamiento de esta actividad se encuentra almacenada en una base de datos interna, implementada en SQLite, donde se registran las señas, rutas de los videos, etiquetas y puntuaciones obtenidas por el usuario. En la **Figura 11** se muestra una vista de esta base de datos, que permite la correcta gestión y vinculación del contenido multimedia con las funcionalidades del aplicativo.

La actividad no se limita únicamente a señas de letras, sino que además promueve palabras completas o frases, lo cual enriquece el contenido presentado y aprendido por el usuario.

Figura 9

Diseño de la interfaz del minijuego en Android Studio

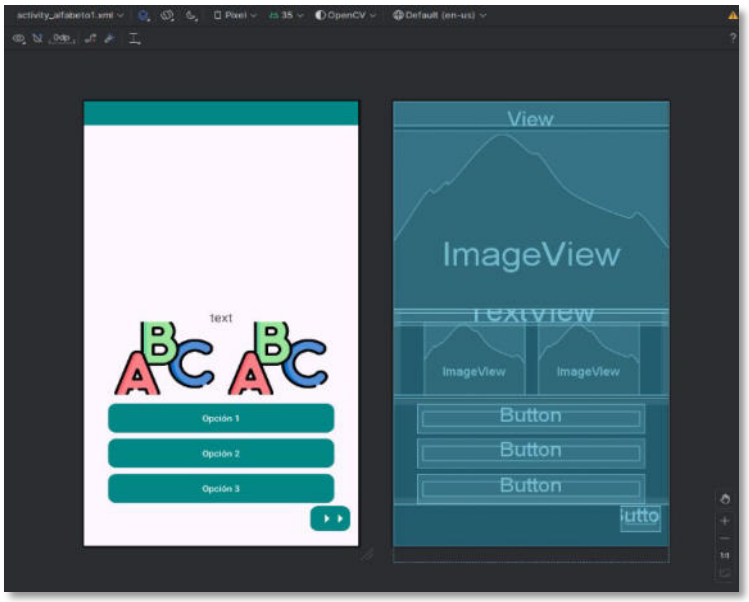


Figura 10

Interfaz del minijuego

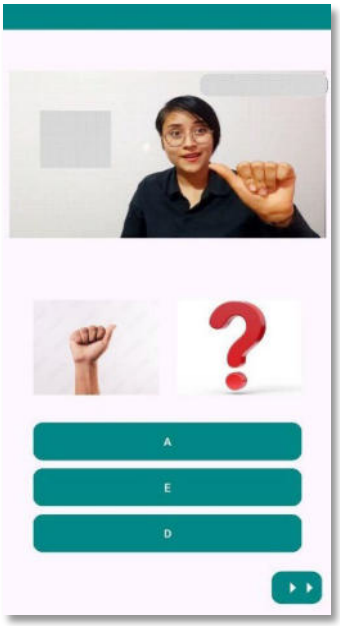
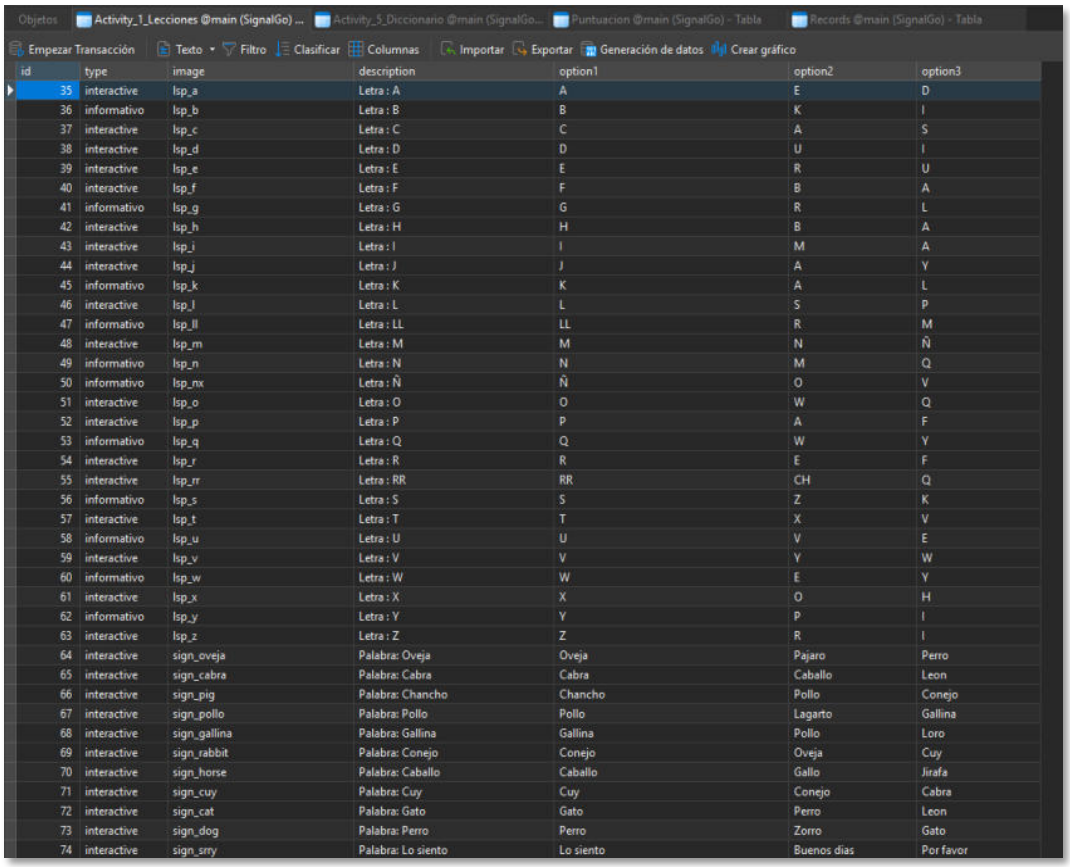


Figura 11

Base de Datos del Activity 1



Objetos | Activity_1_Lecciones @main (SignalGo) ... | Activity_5_Diccionario @main (SignalGo... | Puntuacion @main (SignalGo) - Tabla | Records @main (SignalGo) - Tabla

Empezar Transacción | Texto | Filtro | Clasificar | Columnas | Importar | Exportar | Generación de datos | Crear gráfico

id	type	image	description	option1	option2	option3
35	interactive	lsp_a	Letra : A	A	E	D
36	informativo	lsp_b	Letra : B	B	K	I
37	interactive	lsp_c	Letra : C	C	A	S
38	interactive	lsp_d	Letra : D	D	U	I
39	interactive	lsp_e	Letra : E	E	R	U
40	interactive	lsp_f	Letra : F	F	B	A
41	informativo	lsp_g	Letra : G	G	R	L
42	interactive	lsp_h	Letra : H	H	B	A
43	interactive	lsp_i	Letra : I	I	M	A
44	interactive	lsp_j	Letra : J	J	A	Y
45	informativo	lsp_k	Letra : K	K	A	L
46	interactive	lsp_l	Letra : L	L	S	P
47	informativo	lsp_ll	Letra : LL	LL	R	M
48	interactive	lsp_m	Letra : M	M	N	Ñ
49	informativo	lsp_n	Letra : N	N	M	Q
50	informativo	lsp_nx	Letra : Ñ	Ñ	O	V
51	interactive	lsp_o	Letra : O	O	W	Q
52	interactive	lsp_p	Letra : P	P	A	F
53	informativo	lsp_q	Letra : Q	Q	W	Y
54	interactive	lsp_r	Letra : R	R	E	F
55	interactive	lsp_rr	Letra : RR	RR	CH	Q
56	informativo	lsp_s	Letra : S	S	Z	K
57	interactive	lsp_t	Letra : T	T	X	V
58	informativo	lsp_u	Letra : U	U	V	E
59	interactive	lsp_v	Letra : V	V	Y	W
60	informativo	lsp_w	Letra : W	W	E	Y
61	interactive	lsp_x	Letra : X	X	O	H
62	informativo	lsp_y	Letra : Y	Y	P	I
63	interactive	lsp_z	Letra : Z	Z	R	I
64	interactive	sign_oveja	Palabra: Oveja	Oveja	Pajaro	Perro
65	interactive	sign_cabra	Palabra: Cabra	Cabra	Caballo	Leon
66	interactive	sign_pig	Palabra: Chancho	Chancho	Pollo	Conejo
67	interactive	sign_pollo	Palabra: Pollo	Pollo	Lagarto	Gallina
68	interactive	sign_gallina	Palabra: Gallina	Gallina	Pollo	Loro
69	interactive	sign_rabbit	Palabra: Conejo	Conejo	Oveja	Cuy
70	interactive	sign_horse	Palabra: Caballo	Caballo	Gallo	Jirafa
71	interactive	sign_cuy	Palabra: Cuy	Cuy	Conejo	Cabra
72	interactive	sign_cat	Palabra: Gato	Gato	Perro	Leon
73	interactive	sign_dog	Palabra: Perro	Perro	Zorro	Gato
74	interactive	sign_srry	Palabra: Lo siento	Lo siento	Buenos dias	Por favor

Actividad: Reconocimiento con IA (núcleo de la app)

Esta actividad es el core de toda la aplicación SignalGo, primeramente, porque integra el modelo entrenado con Deep Learning. En esta sección, se pone a prueba al usuario mediante instrucciones como “Haz la letra P”; entonces con la cámara del dispositivo actividad el usuario debe realizar la seña correspondiente a lo leído, mientras tanto las imágenes son capturadas y procesadas por el modelo tflite implementado, este irá determinando si la seña realizada es la correcta.

En la **Figura 13** se muestra el diseño de la interfaz del módulo de reconocimiento desarrollado en Android Studio, donde se definieron los elementos de cámara, superficie de vista y área de retroalimentación visual.

Por su parte, la **Figura 12** presenta la interfaz final en ejecución, donde el sistema muestra al usuario la validación de su seña junto con el puntaje obtenido y mensajes de retroalimentación inmediata.

De ser el caso que acierte, el sistema mostrará un mensaje de éxito y asignara un puntaje; pasando así a la siguiente seña, aquí también se valora la velocidad en la asignación del puntaje de la seña correcta.

Esta dinámica o modo de juego permite al usuario practicar la expresión y uso de las señas de manera interactiva y con retroalimentación en tiempo real, permitiendo así el aprendizaje de manera más didáctica y con un feedback al instante.

Figura 12

Diseño de la interfaz del activity donde se va a reconocer la seña

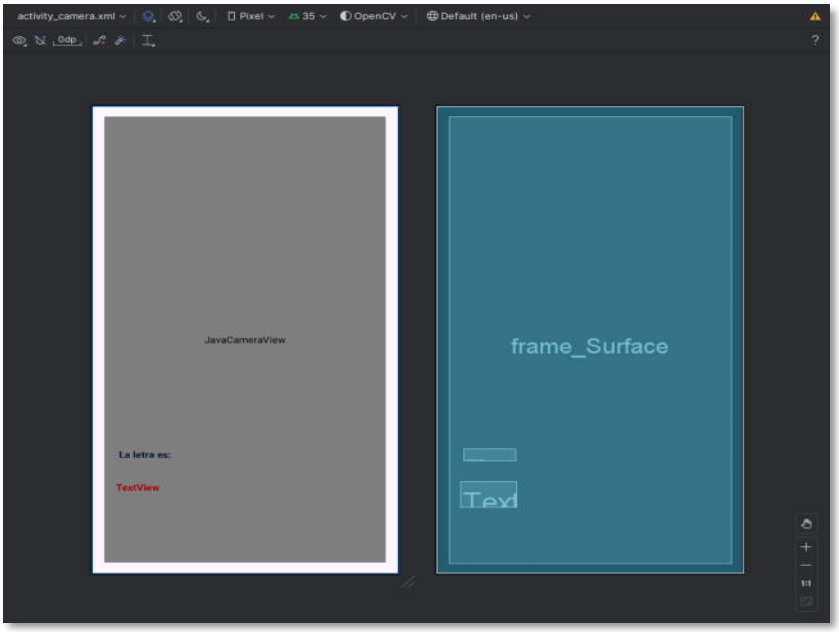
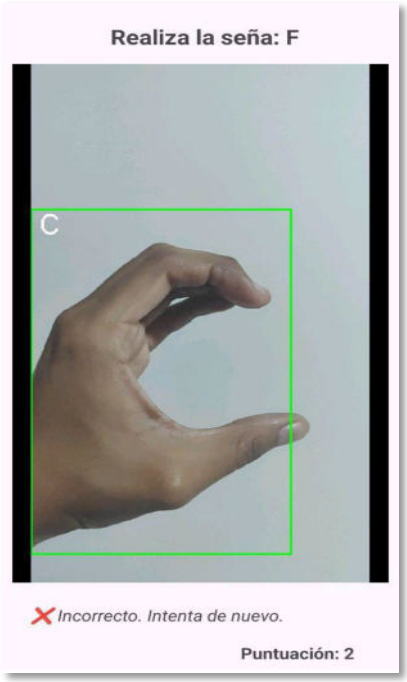


Figura 13

Resultado de la interfaz finalizada donde se reconocerá las señas



Actividad: Juego de cartas

Esta sección nos presenta un juego de cartas con el objetivo de emparejar, el usuario debe encontrar entre todas las cartas los pares correspondientes, esto se compone de una seña y su letra. En la **Figura 14** se muestra el diseño de la interfaz del juego desarrollado en Android Studio, donde se definen los componentes visuales y la disposición de las cartas dentro del tablero de juego.

Por su parte, en la **Figura 15** se presenta la interfaz final ejecutándose en el dispositivo móvil, donde se observa el funcionamiento del juego durante el proceso de emparejamiento entre una seña y su letra correspondiente. La mecánica fortalece tanto la memoria visual como también una experiencia lúdica que va manteniendo y captando el interés del usuario, reforzando así el aprendizaje didáctico del lenguaje de señas peruano.

Figura 14

Diseño de la Interfaz Juego de Cartas

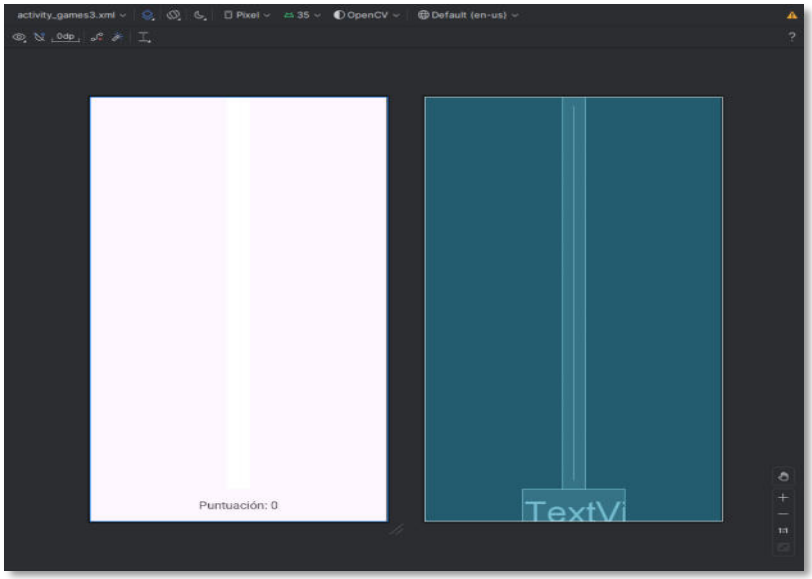
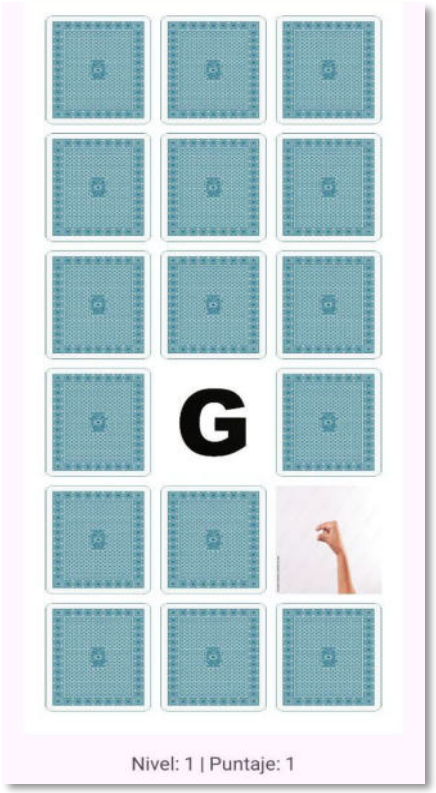


Figura 15

Resultado de la interfaz del juego de cartas



Actividad: Insignias y progreso

Esta pantalla tiene el funcionamiento y objetivo de mostrar los logros del usuario. Mediante esta actividad puede visualizar medallas, puntajes acumulados, los niveles alcanzados y el progreso que está teniendo dentro de la aplicació

En la **Figura 16** se muestra el diseño de la interfaz de la actividad Insignias y Progreso, elaborado en Android Studio, donde se definieron los componentes visuales que permiten presentar los niveles, puntajes, rachas de días consecutivos e insignias obtenidas por el usuario.

Por su parte, en la **Figura 17** se aprecia la interfaz final del aplicativo en ejecución, mostrando el nivel actual, la cantidad de puntos acumulados y las insignias desbloqueadas según el desempeño del usuario.

La información que respalda este funcionamiento se gestiona mediante una base de datos interna, donde se registran las insignias, puntajes y fechas de obtención. En la **Figura 18** se muestra una vista de dicha base de datos, que permite mantener actualizados los logros y el progreso alcanzado.

Esta actividad permite dar una retroalimentación positiva y mantener al usuario siempre motivado al ver su avance, del mismo modo promueve el compromiso por su propio proceso de aprendizaje, dando como resultado la constancia.

Figura 16

Diseño de la Interfaz Insignias y Progreso

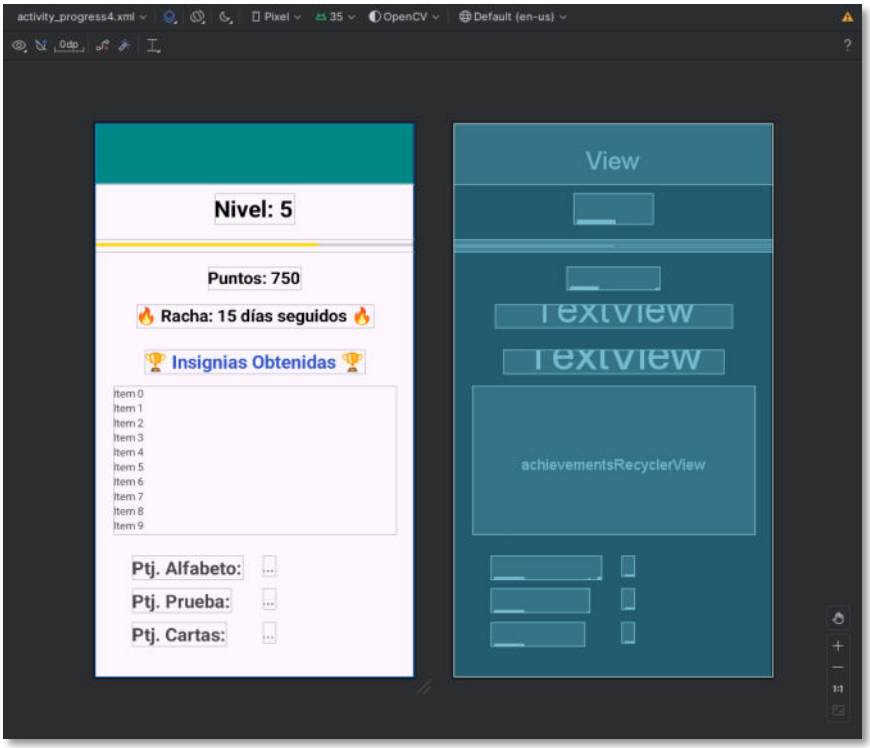


Figura 17

Resultado de la interfaz Insignias y Progreso

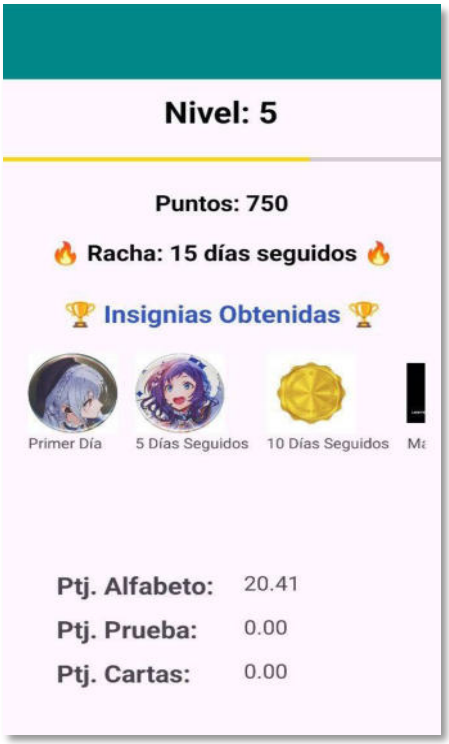
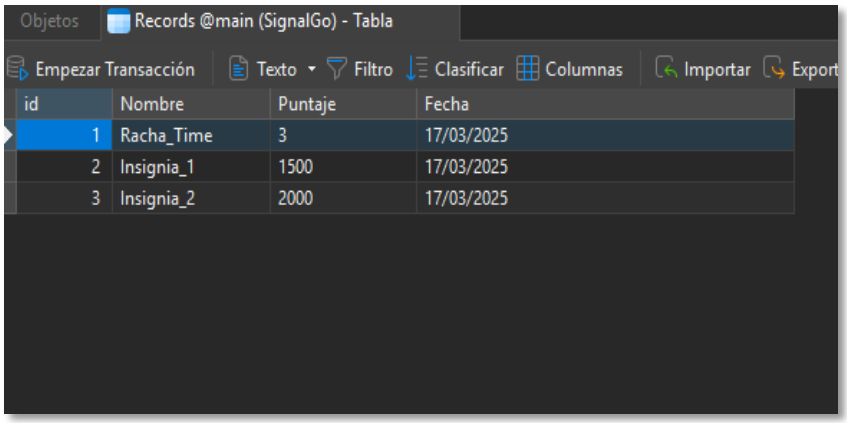


Figura 18

Base de Datos de la interfaz Insignias y Progresos



Actividad: Diccionario de señas

La actividad final incluye un diccionario interactivo que da la oportunidad al usuario de buscar alguna palabra o letra en específico, para así poder visualizar y aprender la seña que corresponde. Cada seña contiene una imagen o video que ilustra la manera de realizarla, así como una breve explicación y contexto del uso de la seña. En la **Figura 19** se muestra el diseño de la interfaz del diccionario desarrollado en Android Studio, mientras que en las **Figuras 20 y 21** se presenta la interfaz final del aplicativo en funcionamiento, donde el usuario puede realizar búsquedas y visualizar el detalle de cada seña o palabra.

Asimismo, en la Figura **22** se muestra la base de datos que almacena las señas, definiciones, descripciones y tipos, garantizando el correcto funcionamiento del diccionario interactivo.

Este es un recurso muy útil para aprender o repasar alguna seña, así como para ir estudiando autónomamente el lenguaje de señas peruano proporcionado por la aplicación.

Figura 19

Diseño de la Interfaz Diccionario de Señas

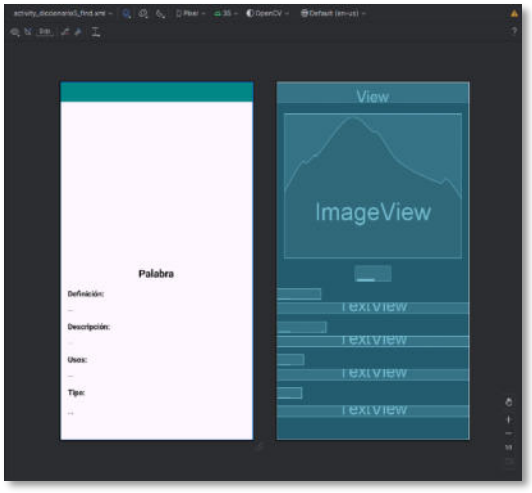


Figura 20

Resultado de la interfaz Diccionario de Señas



Figura 21

Resultado de la interfaz Diccionario de Señas en la sección búsqueda

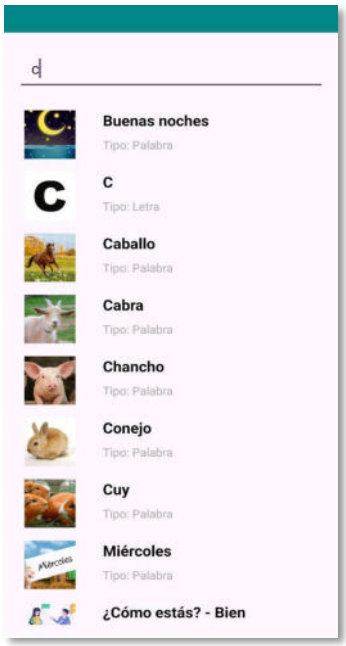
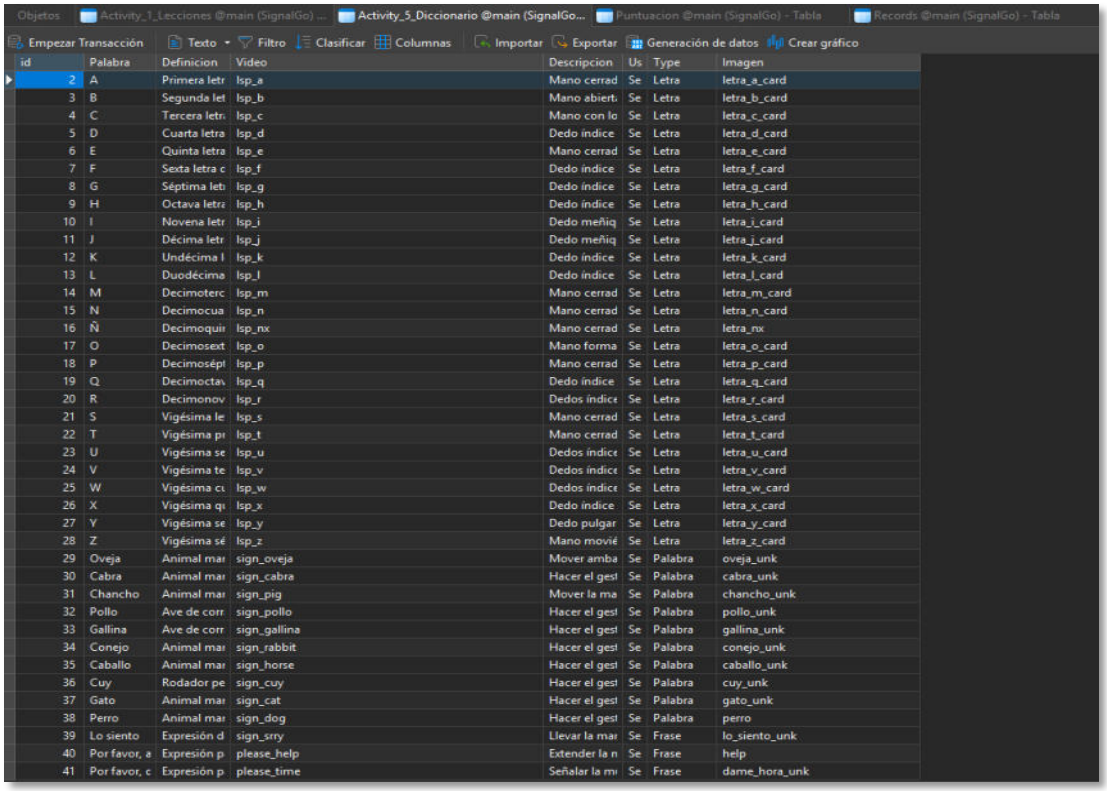


Figura 22

Base de Datos de la interfaz Diccionario de Señas



id	Palabra	Definicion	Video	Descripcion	Us	Type	Imagen
2	A	Primera letr	lsp_a	Mano cerrad	Se	Letra	letra_a_card
3	B	Segunda let	lsp_b	Mano abiert	Se	Letra	letra_b_card
4	C	Tercera letr	lsp_c	Mano con lo	Se	Letra	letra_c_card
5	D	Cuarta letra	lsp_d	Dedo índice	Se	Letra	letra_d_card
6	E	Quinta letra	lsp_e	Mano cerrad	Se	Letra	letra_e_card
7	F	Sexta letra c	lsp_f	Dedo índice	Se	Letra	letra_f_card
8	G	Séptima let	lsp_g	Dedo índice	Se	Letra	letra_g_card
9	H	Octava letra	lsp_h	Dedo índice	Se	Letra	letra_h_card
10	I	Novena letr	lsp_i	Dedo meñiq	Se	Letra	letra_i_card
11	J	Décima letr	lsp_j	Dedo meñiq	Se	Letra	letra_j_card
12	K	Undécima l	lsp_k	Dedo índice	Se	Letra	letra_k_card
13	L	Duodécima	lsp_l	Dedo índice	Se	Letra	letra_l_card
14	M	Decimoterc	lsp_m	Mano cerrad	Se	Letra	letra_m_card
15	N	Decimocua	lsp_n	Mano cerrad	Se	Letra	letra_n_card
16	Ñ	Decimoquir	lsp_nx	Mano cerrad	Se	Letra	letra_nx
17	O	Decimosext	lsp_o	Mano forma	Se	Letra	letra_o_card
18	P	Decimosépt	lsp_p	Mano cerrad	Se	Letra	letra_p_card
19	Q	Decimocta	lsp_q	Dedo índice	Se	Letra	letra_q_card
20	R	Decimonov	lsp_r	Dedos índice	Se	Letra	letra_r_card
21	S	Vigésima le	lsp_s	Mano cerrad	Se	Letra	letra_s_card
22	T	Vigésima pr	lsp_t	Mano cerrad	Se	Letra	letra_t_card
23	U	Vigésima se	lsp_u	Dedos índice	Se	Letra	letra_u_card
24	V	Vigésima te	lsp_v	Dedos índice	Se	Letra	letra_v_card
25	W	Vigésima cu	lsp_w	Dedos índice	Se	Letra	letra_w_card
26	X	Vigésima qu	lsp_x	Dedo índice	Se	Letra	letra_x_card
27	Y	Vigésima se	lsp_y	Dedo pulgar	Se	Letra	letra_y_card
28	Z	Vigésima sé	lsp_z	Mano movié	Se	Letra	letra_z_card
29	Oveja	Animal mar	sign_oveja	Mover amba	Se	Palabra	oveja_unk
30	Cabra	Animal mar	sign_cabra	Hacer el gest	Se	Palabra	cabra_unk
31	Chancho	Animal mar	sign_pig	Mover la ma	Se	Palabra	chancho_unk
32	Pollo	Ave de corr	sign_pollo	Hacer el gest	Se	Palabra	pollo_unk
33	Gallina	Ave de corr	sign_gallina	Hacer el gest	Se	Palabra	gallina_unk
34	Conejo	Animal mar	sign_rabbit	Hacer el gest	Se	Palabra	conejo_unk
35	Caballo	Animal mar	sign_horse	Hacer el gest	Se	Palabra	caballo_unk
36	Cuy	Rodador pe	sign_cuy	Hacer el gest	Se	Palabra	cuy_unk
37	Gato	Animal mar	sign_cat	Hacer el gest	Se	Palabra	gato_unk
38	Perro	Animal mar	sign_dog	Hacer el gest	Se	Palabra	perro
39	Lo siento	Expresión d	sign_srry	Llevar la mar	Se	Frase	lo_siento_unk
40	Por favor, a	Expresión p	please_help	Extender la n	Se	Frase	help
41	Por favor, c	Expresión p	please_time	Señalar la mi	Se	Frase	dame_hora_unk

La aplicación presentada fue construida bajo el enfoque de usabilidad y accesibilidad, de manera cuidadosa se implementa los elementos visuales para que sean representativos y motiven al usuario a seguir aprendiendo, haciendo que la experiencia de uso sea sencilla y amigable. Los componentes interactivos, así como los elementos educativos y la correcta integración del modelo de deep learning generado permite que SignalGo sea una herramienta robusta para el aprendizaje práctico del lenguaje de señas peruano.

3.7 Evaluación de usabilidad y efectividad didáctica

Como parte del proceso de validación integral del sistema, se llevó a cabo una evaluación orientada a medir la usabilidad, funcionalidad y efectividad didáctica de la aplicación SignalGo en condiciones reales de uso. Esta etapa fue esencial para valorar el impacto del sistema desde la perspectiva del usuario final, considerando tanto aspectos técnicos como educativos.

Para ello, se diseñó una prueba de usabilidad que fue aplicada a un grupo de usuarios voluntarios, conformado por personas interesadas en el aprendizaje del Lenguaje de Señas Peruano (LSP). La evaluación tuvo como objetivo recoger impresiones sobre la facilidad de uso, la interfaz gráfica, la precisión del reconocimiento de señas, y el valor percibido como herramienta educativa.

El procedimiento seguido fue el siguiente:

- Se brindó una breve inducción sobre el uso de la aplicación.
- Los usuarios interactuaron libremente con las distintas funcionalidades de la app: reconocimiento de señas en tiempo real, actividades didácticas y el diccionario de señas.
- Se aplicó el System Usability Scale (SUS), un cuestionario estándar de 10 ítems, permitiendo cuantificar el nivel de satisfacción y usabilidad en una escala de 0 a 100.
- Se recolectó además retroalimentación cualitativa mediante preguntas abiertas, observación directa y comentarios espontáneos durante la sesión.

Esta prueba permitió recoger información valiosa sobre la experiencia del usuario, detectar posibles áreas de mejora y verificar que la aplicación cumple su función como herramienta educativa, interactiva e inclusiva para el aprendizaje del LSP.

CAPITULO IV: RESULTADOS Y DISCUSIÓN

4.1 Resultados

En esta sección se presentan los resultados obtenidos tras el entrenamiento y la evaluación de las arquitecturas EfficientNetB0 y MobileNetV2 para el reconocimiento de señas del Lenguaje de Señas Peruano (LSP).

4.1.1. Recolección y Preparación del Dataset

Para entrenar el modelo de reconocimiento se utilizó un conjunto de datos elaborado específicamente para este proyecto, compuesto por 25,000 imágenes que representan las letras del alfabeto en el Lenguaje de Señas Peruano. Las imágenes fueron obtenidas a partir de grabaciones realizadas a varios voluntarios en condiciones controladas pero variadas, lo que permitió capturar distintos escenarios de iluminación, ángulo y fondo. Todas las imágenes fueron correctamente etiquetadas, asegurando que cada clase tuviera su correspondiente representación.

Posterior a la captura, el dataset pasó por una etapa de limpieza donde se descartaron imágenes con mala calidad o errores, manteniendo una base de datos clara y balanceada. Después se redimensionaron las imágenes a 96x96 píxeles, se ajustaron al formato de color requerido por el modelo y se normalizaron sus valores para facilitar el entrenamiento.

Este trabajo de recolección y preparación permitió cumplir con el primer objetivo específico del proyecto, logrando un dataset de calidad, con una buena distribución entre clases y adaptado técnicamente para su uso en modelos de deep learning. Dado este conjunto de datos, fue posible entrenar el modelo de reconocimiento con resultados

favorables en las siguientes etapas. Además, el dataset fue dividido en subconjuntos de entrenamiento, validación y prueba, respetando las proporciones estándar de 70%, 15% y 15% respectivamente, lo que permitió evaluar de manera correcta el desempeño del modelo.

4.1.2. Desarrollo e integración del modelo de reconocimiento en la aplicación móvil

Ambos modelos fueron ajustados mediante Transfer Learning con TensorFlow/Keras y evaluados sobre el subconjunto de prueba (test set) empleando métricas estándar para clasificación multiclase: accuracy, precision, recall, F1-score y loss.

4.1.2.1 Resultados generales por arquitectura

En esta sección, se presentan y analizan las métricas de rendimiento obtenidas. Para evaluar la robustez estadística de los modelos, se calculó el Intervalo de Confianza (IC) del 95% para las métricas clave (Accuracy y F1-Score) mediante el método Bootstrap con 1000 iteraciones. Estos intervalos definen el rango dentro del cual se espera que se encuentre el verdadero rendimiento de los modelos en la población de datos.

En la **Tabla 3** se muestran los rendimientos obtenidos a modo comparativo por cada arquitectura. El modelo MobileNetV2 tuvo un mejor desempeño nominal, alcanzando una Accuracy del 97.57% (IC 95%: [97.04% - 98.03%]) y un F1-score con IC 95%: [97.19% - 98.14%]. Este rendimiento supera ligeramente al de EfficientNetB0, el cual obtuvo una Accuracy del 96.37% (IC 95%: [95.79% - 96.96%]) y un F1-score con IC 95%: [95.96% - 97.02%].

El análisis de solapamiento de los intervalos de confianza resulta clave. Se observa que el IC del 95% para la Accuracy y el F1-score de MobileNetV2 no se solapa con el correspondiente de EfficientNetB0. Esta ausencia de solapamiento indica que la diferencia de rendimiento entre ambos modelos es estadísticamente significativa al nivel de confianza del 95%. En otras palabras, la ventaja de MobileNetV2 no es atribuible al azar ni a la variabilidad del conjunto de prueba.

Además del mejor rendimiento, MobileNetV2 destaca por su eficiencia computacional. Su menor tamaño final en formato TFLite (8.8 MB frente a 15.8 MB) representa una ventaja fundamental para el despliegue en aplicaciones móviles de gama media o de recursos limitados, siendo este un factor decisivo para su selección en contextos reales.

Tabla 3

Resultados Generales por Arquitectura

Modelo	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Loss
MobileNetV2	97.57 (IC: [97.04-98.03])	97.86	97.57	97.66 (IC: [97.19-98.14])	0.2980
EfficientNetB0	96.37 (IC: [95.79-96.96])	96.84	96.37	96.49 (IC: [95.96-97.02])	0.3338

Tamaño y eficiencia del modelo

Para integrarlos en la aplicación móvil, ambos modelos se convirtieron en un formato Tensowflow Lite (.tflite). En la **Tabla 4** se muestra el tamaño final correspondiente a cada modelo. MobileNetV2, con 8.8 MB, el cual resulta mucho más liviano que el modelo EfficientNetB0, el cual tuvo un tamaño de 15.8 MB. La diferencia de tamaño presentada,

combinando con los tiempos de inferencia más rápidos, es clave al momento de optar por elegir el modelo MobileNetV2, debido al rendimiento combinado más optimo, es así que, este modelo será el principal utilizado en la aplicación SignalGo, considerando el despliegue final en dispositivos Android de gama media o recursos limitados.

Tabla 4

Tamaño del modelo convertido a TensorFlow Lite

Modelo	Tamaño. tflite
EfficientNetB0	15.8 MB
MobileNetV2	8.8 MB

En la **Figura 23** se presenta la matriz de confusión correspondiente al modelo **MobileNetV2**. Se observa una alta tasa de aciertos a lo largo de la diagonal principal, lo que indica que el modelo logra **una clasificación precisa en la mayoría de las clases**.

El rendimiento global alcanza una **precisión ponderada del 97.57%**, con un **F1-score promedio ponderado de 97.66%**, lo cual demuestra una alta capacidad del modelo para generalizar en el conjunto de prueba.

4.1.2.2 Matriz de confusión y análisis de errores

En la **Figura 23** se presenta la matriz de confusión generada para el modelo MobileNetv2 sobre el conjunto de prueba que fue evaluado. Se visualiza la existencia de una clara predominancia de valores en la diagonal principal, lo cual nos indica una alta precisión en la clasificación de la mayoría de las clases entrenadas. El modelo logra un alto desempeño, alcanzando de ese modo una precisión del 97.57% y F1-score promedio

de 97.66%, lo que es reflejo de una capacidad de generalización elevada sobre el conjunto de datos utilizado.

Entre las clases que tuvieron un mejor desempeño están las letras A", "C", "E", "F", "G", "H", "M", "O", "P", "Q", "R", "S", "T" e "Y", que presentan una diagonal técnicamente perfecta, sin confusiones significativas con otras clases puestas a prueba.

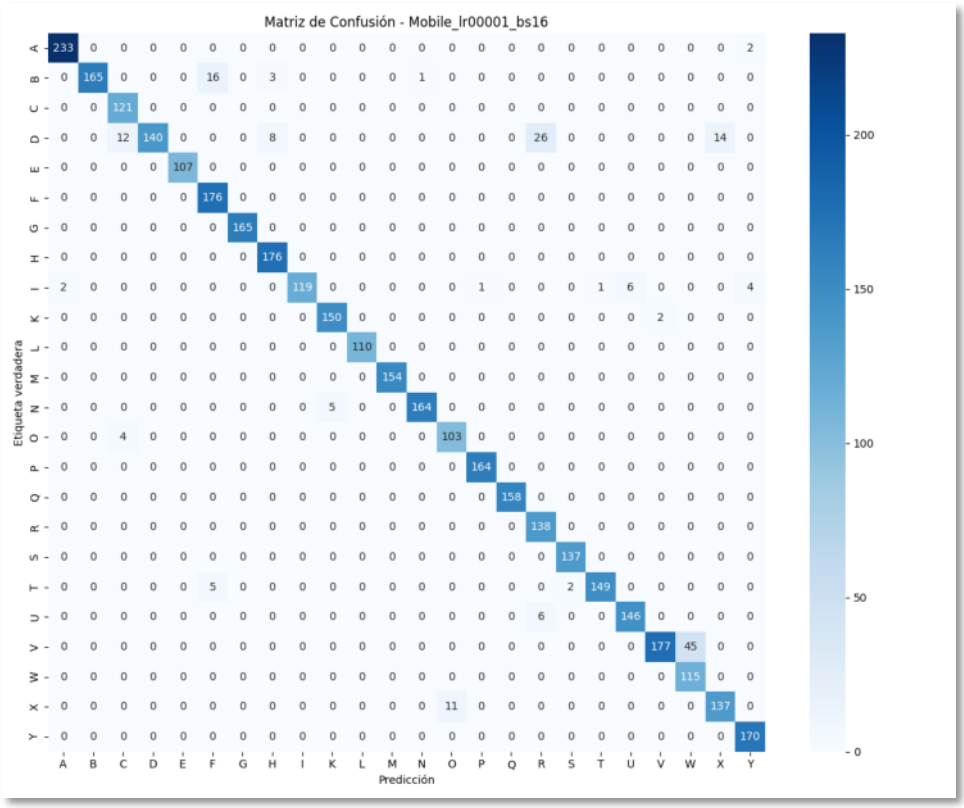
Por ejemplo, la Letra "F" tuvo una correcta clasificación en la totalidad de casos, mientras que la letra "A" solo tuvo 2 errores sobre 235 muestras, lo cual hace notar que el comportamiento del modelo es altamente confiable en dichas categorías.

Sin embargo, en la matriz observada a mayor detalle, se pueden identificar ciertos errores sistemáticos en algunas clases específicas. En la letra "D", esta fue clasificada correctamente en 140 de 200 casos, teniendo confusiones recurrentes con la letra U y Y, lo que sugiere y denota que presenta dificultades al momento de diferenciar estos gestos que tienen configuraciones visuales parecidas. De manera similar, la letra "V" fue errada en 45 ocasiones con la letra "W", debido a su parecido visual entre ambas señas. La clase "I" presento 11 errores hacia "X", probablemente debido a temas de ángulo o postura de mano. Asimismo, la letra K solo tuvo dos errores con la letra R, lo cual no es muy significativo considerando la muestra de esa clase. En los errores encontrados se evidencia la necesidad o importancia de reforzar en la diferenciación de signos visualmente parecidos, mediante técnicas de aumento de datos o estrategias de entrenamiento enfocadas.

En conjunto, la matriz confirma el buen desempeño del modelo en términos generales, además que permite identificar puntos específicos de mejora en ciertas clases que luego pueden ser abordados con estrategias de refinamiento de ciertas categorías que presentan ambigüedad visual.

Figura 23

Matriz de confusión del modelo MobileNetV2



4.1.2.3 Predicciones en imágenes aleatorias

En la **Figura 24** se muestran 12 ejemplos visuales seleccionados aleatoriamente del conjunto sometido a prueba. Cada imagen incluye la etiqueta real y la predicción del modelo MobilenetV2.

El análisis visual refleja que se tuvo un rendimiento muy bueno en la mayoría de los casos. El modelo logra predecir de manera correcta señales claras y centradas, incluso cambiando el fondo. Sin embargo, se identificador errores puntuales cuando estaban bajo ciertas condiciones como imágenes borradas debido al movimiento en la captura, o iluminación variante, y en señas que están parcialmente ocultas por temas de ropa, ángulo.

Las limitaciones pueden abordarse en trabajos futuros mediante mejoras en la etapa de preprocesamiento.

Figura 24

Etiquetas vs predicciones de algunas imágenes aleatorias del modelo MobileNetV2



4.1.2.4 Evaluación de la aplicación móvil en condiciones reales

Al finalizar el proceso de entrenamiento y la generación del modelo MobielNetV2 a formato Tensorflow Lite (tflite), se integró en la aplicación desarrollada SignalGo la cual fue diseñada para el correcto reconocimiento de Lenguaje de señas peruano (LSP). En esta sección se hace la descripción del comportamiento del sistema ejecutándose bajo un entorno real, haciendo uso de un dispositivo Android de gama media. La evaluación realizada se centró en tres aspectos: tiempo de inferencia, funcionamiento en tiempo real y robustez en condiciones externas diferentes.

4.1.2.5 Tiempo de inferencia promedio

Durante las pruebas que se hicieron en el dispositivo Android de gama media utilizado, se midió el tiempo promedio que el modelo usa para procesar una imagen que es captura por la cámara y hasta que devuelve una predicción. Este factor o parámetro es clave para poder asegurar una experiencia fluida y natural en aplicaciones de reconocimiento en tiempo real.

Modelo MobileNetV2

- Tiempo de inferencia promedio: 34 milisegundos por fotograma.

Este resultado obtenido es considerado adecuado para aplicaciones de reconocimiento en tiempo real, lo cual permite tener una tasa de refresco cerca de 30 frames por segundo. A nivel practico, esto es traducido como una respuesta inmediata o rápida tras la ejecución de una seña, lo cual favorece una interacción fluida y continua.

4.1.2.6 Funcionamiento en tiempo real

Asimismo, también se llevaron a cabo pruebas de funcionamiento bajo condiciones reales, los usuarios realizaban diferentes letras del alfabeto frente a la cámara de su dispositivo; de este modo se pudieron evaluar tres aspectos fundamentales:

- **Estabilidad:** La aplicación se mantuvo en operaciones en las sesiones de uso las cuales fueron prolongadas, sin haber cierres inesperados o errores al ejecutarse.
- **Fluidez:** La interfaz de la aplicación y su uso tuvieron respuestas agiles y fluidas, teniendo una baja latencia entre la captura de las imágenes y la visualización del resultado de la predicción del modelo.

- **Precisión:** El modelo mostro un nivel alto y óptimo de aciertos, mientras estaba bajo condiciones ideales de iluminación y postura.

Los resultados obtenidos dan evidencia de que la aplicación ofrece una experiencia interactiva optima y agradable para el usuario final.

4.1.2.7 Robustez ante variaciones externas

El sistema fue sometido a una evaluación ante condiciones no ideales para probar su robustez, es así que se desarrollaron pruebas específicas con variaciones en los entornos y condiciones. Los resultados obtenidos se muestran en la **Tabla 5.**

Tabla 6

Variaciones aplicadas y resultados observados

Variación aplicada	Resultado observado
Iluminación baja	La precisión disminuyo levemente en cambios de luz del entorno.
Fondo complejo	Reducción moderada de precisión si el fondo incluía objetos visualmente parecido a manos.
Cambio de ropa o tono de piel	No se evidenció impacto relevante; el modelo generalizó correctamente para los distintos usuarios.

En conjunto, los resultados demostraron que la aplicación mantiene un desempeño sólido y optimo ante posibles variaciones del entorno, esto es clave para una aplicación que tiene un objetivo practico y uso en contextos reales y distintos. La capacidad de adaptación mostrada refuerza la viabilidad de uso de SignalGo como una

herramienta accesible y útil para el aprendizaje e interpretación del lenguaje de señas para dispositivos móviles.

4.1.3 Resultados de usabilidad

Con el objetivo de evaluar la experiencia de uso y la efectividad didáctica presentada por SignalGo, se realizado una prueba de usabilidad con un grupo de usuarios voluntarios. La finalidad de la evaluación realizada fue el análisis de la satisfacción general, la facilidad de uso y la precision percibida en cuanto al reconocimiento de señas, así como la accesibilidad presentada por la aplicación en condiciones variadas y reales de funcionamiento, usando un estándar internacional.

Nivel de satisfacción de los usuarios usando el System Usability Scale (SUS)

La experiencia de usuario y la satisfacción de la aplicación SignalGo fue evaluada utilizando el System Usability Scale (SUS), un cuestionario de 10 ítems reconocido a nivel mundial para medir la usabilidad de sistemas y aplicaciones. Se contó con la participación de 12 usuarios voluntarios (estudiantes y personas interesadas en el aprendizaje de señas), un tamaño de muestra considerado válido para la aplicación de este instrumento.

Tras la recopilación de las respuestas y la aplicación de la fórmula de cálculo del SUS (que normaliza el resultado a una escala de 0 a 100), el puntaje global obtenido para la aplicación SignalGo fue de 82.3, la cual es mostrada en la **Tabla 6**.

Tabla 7

Puntaje Global del System Usability Scale (SUS)

Métrica	Puntaje (0 a 100)
Puntaje SUS	82.3

El puntaje de 82.3 se sitúa significativamente por encima del benchmark promedio global de 68, el cual representa el umbral de usabilidad aceptable. De acuerdo con la literatura en la ingeniería de usabilidad, un puntaje superior a 80.3 se clasifica como de usabilidad Excelente, lo que confirma que la aplicación SignalGo cumple con los requisitos funcionales, además de ofrecer una experiencia de usuario altamente satisfactoria y bien diseñada.

Este resultado valida el diseño de la interfaz y la experiencia de usuario general como un factor clave en la efectividad didáctica percibida por el usuario.

Comentarios y retroalimentación recibida

En las sesiones de evaluación de usabilidad, se pudo recopilar retroalimentación y comentarios de parte de los usuarios, los cuales se correlacionan con el alto puntaje SUS, a la vez que identifican áreas específicas de mejora:

- “La aplicación me parece bastante útil y nunca había usado una similar para tal propósito de aprender señas”.
- “Me parecería interesante poder practicar palabras completas o frases, además de solo letras.”
- “En general fue bastante bueno sin embargo algunas letras se confundían, cuando movía ligeramente las manos”.
- “Es impresionante que pueda funcionar sin acceso a internet, así permite practicar sin limitaciones.”

Los comentarios representan aspectos positivos del sistema desarrollado, a su vez, que muestra sugerencias y puntos de mejora para futuras versiones.

Limitaciones observadas y oportunidades de mejora

Se pudieron identificar ciertas limitaciones a pesar del nivel de satisfacción **RESULTANTE**, lo cual permite tener oportunidades de mejora para el sistema, se presenta el cuadro con las observaciones encontradas en la **Tabla 7**.

Tabla 8

Limitaciones Observadas y sus Detalles

Observación	Detalle
Cobertura limitada	Actualmente, la aplicación solo reconoce las letras del alfabeto entrenado.
Sensibilidad al movimiento	Los movimientos repentinos o cambios de las manos hacen que pueda reconocer una seña errónea.
Interfaz mejorable	Se puede mejorar la interfaz para que sea más comprensible y de fácil uso para el usuario, como una guía que permita al usuario navegar por los componentes del sistema.

Las observaciones presentadas son tomadas en consideración para futuras mejores del desarrollo presentado, priorizando los puntos de accesibilidad, contenido interactivo y robustez del sistema ante condiciones cambiantes de entorno o variantes de uso.

4.2 Discusión de resultados

El análisis de los resultados obtenidos se orienta a contrastar el rendimiento y la eficiencia de los modelos desarrollados con los antecedentes de la literatura, además de

validar la efectividad de la aplicación móvil frente a la experiencia del usuario final. La recolección y procesamiento de datos permitió construir un dataset robusto de 25,000 imágenes, lo que representa una mejora notable frente a estudios previos como el de Berru Novoa (2025), quien empleó solo 2,400 imágenes y obtuvo 89.46 **%** de accuracy con HOG + SVM. Esto confirma que la cantidad y variabilidad de datos es un factor determinante para mejorar la generalización del modelo.

En cuanto al desempeño de los modelos, MobileNetV2 fue el seleccionado para el despliegue final al alcanzar una Accuracy de 97.57 % (IC 95%: [97.04%–98.03%]) y un F1-score de 97.66 % (IC 95%: [97.19%–98.14%]), superando ampliamente métodos clásicos y también modelos híbridos no optimizados. A modo de comparación, Hussain et al. (2023) reportaron 99.1 % con EfficientNet-B0, mientras que Baihan et al. (2024) alcanzaron 98.7 % con CNNSa-LSTM; sin embargo, estos estudios utilizaron escenarios más controlados.

Asimismo, MobileNetV2 demuestra un rendimiento competitivo frente a sistemas dinámicos como el de Kumari & Anand (2024), quienes lograron 84.65 % con CNN + atención + LSTM, manteniendo una arquitectura más ligera y eficiente para dispositivos móviles. Respecto al equilibrio entre precisión y eficiencia, MobileNetV2 no solo mostró el mejor rendimiento en métricas, sino que también destacó por su tamaño reducido (8.8 MB en TFLite) frente a alternativas como EfficientNet-B0 (15.8 MB).

Esta diferencia resulta determinante para aplicaciones móviles, especialmente considerando las limitaciones de memoria y procesamiento señaladas en estudios previos. Por ejemplo, Wang & Chen (2020) documentaron caídas significativas de

rendimiento en MobileNetV2 bajo iluminación deficiente, lo cual coincide parcialmente con las pruebas de robustez del presente estudio, donde se observó una ligera disminución de precisión en entornos de baja calidad visual.

La evaluación en el dispositivo móvil reafirma estas ventajas. El tiempo de inferencia de 34 ms por fotograma se sitúa por debajo del umbral de tiempo real recomendado (50 ms), reportado en investigaciones como la de Wang & Chen (2020), lo que garantiza una interacción fluida y sin retrasos perceptibles. Este rendimiento es especialmente relevante en aplicaciones educativas y de accesibilidad, donde la rapidez de respuesta condiciona directamente la experiencia del usuario y la efectividad del aprendizaje.

Además, se obtuvo una experiencia de usuario altamente positiva: el puntaje 82.3 en el System Usability Scale (SUS) clasifica la aplicación como “Excelente”, superando ampliamente el umbral estándar de 68. Este resultado coincide con hallazgos de tesis orientadas al aprendizaje asistido mediante DL, como la de Vega Calero (2023), donde el uso de herramientas digitales mejoró significativamente los niveles de comprensión y producción de signos.

En conjunto, estos resultados confirman que MobileNetV2 ofrece una combinación óptima de precisión, eficiencia y usabilidad, superando los desafíos reportados en estudios previos, especialmente aquellos relacionados con la latencia, el consumo de recursos o la falta de datasets amplios.

Finalmente, la robustez del sistema ante variaciones moderadas de fondo, piel y vestimenta demuestra que el modelo está adecuadamente preparado para un uso cotidiano, consolidando su valor como herramienta de apoyo en el aprendizaje e inclusión comunicativa.

.

CONCLUSIONES

La presente investigación tuvo por objetivo principal el desarrollar una aplicación móvil didáctica para al aprendizaje del Lenguaje de Señas Peruano; por lo cual este trabajo integró y ejecuto de forma coherente distintas etapas técnicas, desde la recopilación y procesamiento de datos, el entrenamiento del modelo de reconocimiento de señas hasta el diseño de la aplicación y su posterior validación con usuarios reales, cumpliendo así con los objetivos establecidos en la investigación.

En este sentido, se desarrolló SignalGo, una aplicación funcional que permite al usuario aprender de forma autónoma mediante una experiencia dinámica e intuitiva. La aplicación fue concebida para facilitar la navegación y la interacción, ofreciendo una estructura clara y accesible que se adapta a distintos tipos de usuarios y escenarios de uso, especialmente en dispositivos móviles de gama media.

Asimismo, se llevó a cabo una recolección y procesamiento de datos visuales del Lenguaje de Señas Peruano, creando así un conjunto de datos compuesto por 25,000 imágenes capturadas bajo condiciones controladas. Estas imágenes fueron preprocesadas para asegurar uniformidad y calidad, permitiendo de este modo su uso eficiente en el entrenamiento de los modelos de aprendizaje profundo a entrenar. El dataset resultante representa un aporte técnico valioso, debido a su volumen como también por su enfoque; el cual esta contextualizado en señas peruanas.

Con el propósito de desarrollar e integrar un modelo de reconocimiento de señas basado en Deep Learning dentro de la aplicación, se entrenaron dos arquitecturas: EfficientNetB0 y MobileNetV2. Ambos modelos presentados alcanzaron métricas superiores al 96 % en términos de accuracy y F1-score. MobileNetV2 fue el seleccionado debido a su menor tamaño y velocidad en tiempo de inferencia, lo cual permitió garantizar

un rendimiento fluido y optimo tras integrarlo en el entorno móvil. Este modelo fue exportado en formato .tflite, asegurando una ejecución eficiente en dispositivos móviles con recursos limitados.

Asimismo, como parte de la evaluación de la usabilidad y efectividad didáctica del aplicativo, se realizaron pruebas con usuarios reales. Las cuales demostraron un alto nivel de satisfacción en aspectos como facilidad de uso, claridad visual y precisión del sistema. Además, se comprobó la estabilidad operativa del modelo en diversas condiciones, incluyendo cambios de iluminación, fondos complejos y variabilidad física entre usuarios. A partir de esta validación práctica, se identificaron oportunidades de mejora para la aplicación, como la expansión del vocabulario, ajustes en la interfaz y una tolerancia considerable antes movimientos inesperados o rápidos.

En conjunto, la investigación aporta una solución tecnológica viable con SignalGo, integrando de forma efectiva las etapas de recolección de datos, modelado y despliegue funcional de una aplicación móvil. Académicamente, se propone una metodología replicable que puede servir de base para futuras investigaciones en tecnologías educativas. Socialmente, se contribuye a la inclusión al brindar una herramienta de apoyo accesible para el aprendizaje del Lenguaje de Señas. Desde el plano tecnológico, se demuestra la aplicabilidad de modelos de Deep Learning optimizados para su uso en entornos móviles. Las limitaciones encontradas abren nuevas líneas de desarrollo, como la incorporación del reconocimiento de frases completas, mejoras en la robustez del sistema y la adaptación a otros lenguajes de señas.

RECOMENDACIONES

En referencia a los resultados que se obtuvieron dado el desarrollo y evaluación correspondiente de la Aplicación SignalGo, se han planteado recomendaciones que se orientan a distintos actores que pueden contribuir en la mejora e implementación como herramienta tecnológica que apoye el aprendizaje del Lenguaje de Señas Peruano (LSP).

En el ámbito educativo, se da como sugerencia que las instituciones; que tengan estudiantes con alguna discapacidad auditiva como las de educación regular, tomen en consideración el uso de la aplicación SignalGo como recurso didáctico para completar sus programas académicos. La aplicación puede integrarse como material de apoyo en clases de comunicación, tutorías inclusivas o talleres de sensibilización, promoviendo así el aprendizaje básico del LSP desde edades tempranas. Esta integración favorecería a los estudiantes con discapacidad, como también fomentaría una cultura de inclusión, empatía y accesibilidad comunicativa en toda la comunidad educativa.

Desde una perspectiva técnica, orientada a desarrolladores, es recomendable trabajar en el aspecto de escalabilidad de la aplicación con el fin de permitir su adaptación a otros lenguajes de señas existentes, como el American Sign Language (ASL) o la Lengua de Señas Mexicana (LSM), ampliando de esta forma su aplicabilidad a otros contextos. Además, se considera relevante optimizar el modelo de reconocimiento para hacerlo más robusto frente a variaciones externas como cambios en la iluminación, movimientos imprecisos o fondos variados.

La aplicación de técnicas como data augmentation y el uso de arquitecturas híbridas de redes neuronales pueden mejorar el modelo utilizado pare tener un mejor rendimiento del sistema en condiciones reales variantes. También se sugiere continuar desarrollando la interfaz de usuario, priorizando el diseño accesible, la navegación

intuitiva y la incorporación de tutoriales o guías de uso inicial que faciliten el aprendizaje autónomo, especialmente en personas con bajo nivel de alfabetización digital.

Finalmente, para futuros investigadores, se recomienda ampliar el alcance funcional de la plataforma propuesta, pasando así desde el reconocimiento del abecedario hacia la interpretación de palabras y frases completas en Lengua de Señas Peruana. Este avance puede permitir el desarrollo de una herramienta mucho más integral, con un impacto real en situaciones de comunicación cotidiana. Asimismo, se considera adecuado la incorporación de módulos interactivos que ayuden en el refuerzo del proceso de aprendizaje, tales como ejercicios guiados, pruebas de autoevaluación o juegos didácticos, los cuales podrían mejorar significativamente la retención del conocimiento proporcionado a los usuarios. Por último, se sugiere realizar pruebas de usabilidad con una muestra más diversa y amplia, considerando aspectos como edad, nivel educativo y experiencia previa con tecnología, lo que permitiría obtener retroalimentación más representativa y fortalecer el impacto social y educativo de la solución tecnológica propuesta.

REFERENCIAS

Arias, F. G. (2012). *El proyecto de investigación: Introducción a la metodología científica.* Caracas: Episteme.

Baihan, A., Alutaibi, A.I., Alshehri, M. et al. Sign language recognition using modified deep learning network and hybrid optimization: a hybrid optimizer (HO) based optimized CNNSa-LSTM approach. Sci Rep 14, 26111 (2024). https://doi.org/10.1038/s41598-024-76174-7

Berru Novoa, B. J. (2025). *Reconocimiento de gestos estáticos del abecedario de la lengua de señas peruana utilizando cámaras de baja resolución* (Tesis de licenciatura). Universidad Peruana de Ciencias Aplicadas (UPC). https://hdl.handle.net/10757/628221

Bernal Torres, C. A. (2010). Metodología de la investigación. Bogotá: Pearson Educación.

Brooke, J. (1996). SUS: A quick and dirty usability scale. London: Taylor and Francis.

Chen, L., Li, H., & Wang, Q. (2021). Sampling theory for sign language datasets. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 10234–10243.

CONADIS. (2021). Guía de capacitación en lengua de señas peruana. Obtenido de Ministerio de la Mujer y Poblaciones Vulnerables: https://www.conadisperu.gob.pe/recursos/publicaciones/

Congreso de la República del Perú. (2010). Ley n.º 29535: Ley que reconoce la lengua de señas peruana como lengua de las personas sordas en el Perú. Perú: El Peruano.

Deitel, P. J., & Deitel, H. M. (2017). Java: How to Program, Early Objects. (Pearson, Ed.) Pearson.

Forsyth, D. A., & Ponce, J. (2012). Computer vision: A modern approach. Pearson.

Fu, H., Niu, Z., Zhang, C., Ma, J., & Chen, J. (2016). Visual cortex inspired CNN model for feature construction in text analysis. Frontiers in Computational Neuroscience, 64.

Google. (2021). Recap of TensorFlow at Google I/O 2021. The TensorFlow Blog. Obtenido de https://blog.tensorflow.org/2021/05/recap-of-tensorflow-at-google-io-2021.html

Greenspan, R. L., & Loftus, E. F. (2020). Eyewitness confidence malleability: Misinformation as post-identification feedback. Law and Human Behavior, 44, 194–208. doi:https://doi.org/10.1037/lhb0000369

Hernández Sampieri, R. F. (2014). Metodología de la investigación. Ciudad de México: McGraw Hill.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & ... Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. doi:https://doi.org/10.48550/arXiv.1704.04861

Hubel, D. H. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology, 160, 1.

Hussain, A., Ul Amin, S., & Fayaz, M. (2023). An Efficient and Robust Hand Gesture Recognition System of Sign Language Employing Finetuned Inception-V3 and Efficientnet-B0 Network. Computer Systems Science & Engineering, 46(3).

Imran, A., Razzaq, A., Baig, I. A., Hussain, A., Shahid, S., & ur Rehman, T. (2021). Dataset of Pakistan Sign Language and automatic recognition of hand configuration of Urdu

alphabet through machine learning. Data in Brief, 36, 107021. doi:https://doi.org/10.1016/j.dib.2021.107021

Instituto Nacional de Estadística e Informática (INEI). (2017). Primeros resultados del Censo Nacional 2017: XII de población, VII de vivienda y III de comunidades indígenas. Obtenido de INEI: https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1540/libro.pdf

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25.

Kumari, D., & Anand, R. S. (2024). Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. Electronics, 13(7), 1229. https://doi.org/10.3390/electronics13071229

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436–444. doi:https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. New York: IEEE.

Llantoy Luján, F. (2024). *Implementación de un aplicativo móvil usando deep learning para mejorar el reconocimiento de lengua de señas peruana en Lima, Perú* (Tesis de licenciatura). Universidad Tecnológica del Perú. https://repositorio.utp.edu.pe/item/870f61c6-7c4a-42ba-8421-024d06775e0c

Ministerio de Educación del Perú. (23 de setiembre de 2021). Más de 4,200 estudiantes con discapacidad auditiva se encuentran matriculados en II.EE. del Minedu. Obtenido de Gobierno del Perú: https://www.gob.pe/institucion/minedu/noticias/527771

Ministerio de Salud del Perú (MINSA). (2014). Reglamento de la Ley General de la Persona con Discapacidad. Ministerio de Salud del Perú (MINSA). Peru: Ministerio de Salud.

Muñoz, E. (2013). El lenguaje de señas peruana: Estructura y uso. Universidad Nacional Mayor de San Marcos.

Norman, D. A. (2013). The design of everyday things. Basic Books.

Organismo Supervisor de Inversión Privada en Telecomunicaciones (OSIPTEL). (2023). Reporte de acceso a internet y dispositivos móviles en Perú. Obtenido de OSIPTEL: https://www.osiptel.gob.pe

Ortiz, N., & Camargo, J. E. (2020). Modelo computacional para reconocimiento de lenguaje de señas en un contexto colombiano. TecnoLógicas, 23, 191–226.

Perú, M. d. (s.f.). Más de 4,200 estudiantes con discapacidad auditiva se encuentran matriculados en II.EE. del Minedu. Obtenido de Gobierno del Perú: https://www.gob.pe/institucion/minedu/noticias/527771-mas-de-4200-estudiantes-con-discapacidad-auditiva-se-encuentran-matriculados-en-iiee-del-minedu

Provost, F. &. (2001). Robust classification for imprecise environments. Machine Learning,

Quispe, R., & Huamán, L. (2021). Peruvian sign language recognition using convolutional neural networks. Journal of Peruvian Engineering, 12, 45–60. Obtenido de http://revistas.uni.edu.pe/index.php/ingenieria/article/view/123

Rosenblatt. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Washington: American Psychological Association.

Rumelhart, D. E. (1986). Learning representations by back-propagating errors. London: Nature Publishing Group.

Salgado Martínez, G. A., Cuevas Valencia, R. E., Morales, A. F., & Catalán Villegas, A. (2024). Reconocimiento de la Lengua de Señas Mexicana usando deep learning mediante una aplicación móvil. Ciencia Latina Revista Científica Multidisciplinar, 8(5). https://doi.org/10.37811/cl_rcm.v8i5.14458

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. doi:https://doi.org/10.48550/arXiv.1905.11946

Vega Calero, V. H. (2023). Deep learning para el aprendizaje del lenguaje de señas peruanas (Tesis de licenciatura). Universidad César Vallejo. https://repositorio.ucv.edu.pe/handle/20.500.12692/134452

Wang, J., & Chen, X. (2021). Signnet: A hybrid CNN-RNN architecture for sign language recognition. Neurocomputing, 434, 224–238. doi:https://doi.org/10.1016/j.neucom.2020.12.072

Wang, Y., & Chen, L. (2020). Lightweight CNN for real-time sign language recognition on mobile devices. IEEE Transactions on Human-Machine Systems, 215–224.

Zhang, R., & Li, W. (2023). Automated deep learning design for sign language recognition using AutoML. IEEE Transactions on Neural Networks, 1892–1905.

ANEXOS

ANEXO 1: CUESTIONARIO SUS (SYSTEM USABILITY SCALE) REALIZADA A LOS USUARIOS QUE USARON LA APP

Instrucciones: A continuación, encontrarás 10 enunciados sobre el sistema que acabas de utilizar. Por favor, lee cada uno cuidadosamente y marca con una "X" en el círculo/casilla que mejor represente tu grado de acuerdo o desacuerdo, utilizando la siguiente escala:

- ○ **1 - Totalmente en desacuerdo**
- ○ **2 - En desacuerdo**
- ○ **3 - Neutro**
- ○ **4 - De acuerdo**
- ○ **5 - Totalmente de acuerdo**

Preguntas:

1. Creo que me gustaría utilizar este sistema con frecuencia.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

2. Encontré el sistema innecesariamente complejo.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

3. Pensé que el sistema era fácil de usar.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

5. Encontré que las diversas funciones de este sistema estaban bien integradas.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

6. Pensé que había demasiada inconsistencia en este sistema.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

8. Encontré el sistema muy complicado de usar.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

9. Me sentí muy seguro usando el sistema.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

10. Necesitaba aprender muchas cosas antes de empezar con este sistema.

○ 1 - Totalmente en desacuerdo
○ 2 - En desacuerdo
○ 3 - Neutro
○ 4 - De acuerdo
○ 5 - Totalmente de acuerdo

COMENTARIOS ADICIONALES

¿Qué aspectos considera que podrían mejorar en la aplicación para facilitar aún más su uso y optimizar la experiencia de aprendizaje?

_____

_____

_____

_____

ANEXO 2: MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES

Variable	Dimensión	Indicador	Escala
1	Modelo de Deep Learning	Arquitectura del modelo (CNN, MobileNet, etc.)	Categórica
		Optimización y entrenamiento (Épocas, tasa de aprendizaje, función de pérdida)	Numérica (valores técnicos)
2	Precisión del Modelo	Accuracy (proporción de aciertos)	Rango 0-1
		Recall (Sensibilidad)	Rango 0-1
		F1-score	Rango 0-1
		Precisión	Rango 0-1
3	Usabilidad del Sistema	Facilidad de uso (percepción de navegación)	Escala Likert (1-5)
		Satisfacción general	Escala Likert (1-5)
		Facilidad de aprendizaje	Escala Likert (1-5)
		Percepción de aprendizaje	Escala Likert (1-5)

ANEXO 3: MATRIZ DE CONSISTENCIA

Problema	Objetivos	Variables y Dimensiones	Indicadores
¿Es posible desarrollar un aplicativo móvil basado en deep learning que facilite tanto el aprendizaje como el reconocimiento del lenguaje de señas?	Desarrollar un aplicativo móvil didáctico usando deep learning para el aprendizaje y reconocimiento de lenguaje de señas peruano.	**Variable 1:** Modelo de Deep Learning	• Arquitectura del modelo (CNN, MobileNet, etc.) • Optimización y entrenamiento
¿Cómo se puede recolectar y procesar un conjunto de datos visuales del Lenguaje de Señas de señas peruano que sea adecuado para entrenar un modelo de deep learning?	Recolectar y procesar datos visuales del Lenguaje de Señas de señas peruano para entrenar un modelo de deep learning.	**Variable 2:** Precisión del Modelo	• Accuracy, Recall • F1-score, Precisión
¿Qué técnicas de deep learning resultan más efectivas para desarrollar un modelo que reconozca señas en tiempo real a través de una cámara móvil?	Desarrollar e integrar un modelo de reconocimiento de señas basado en deep learning dentro de la aplicación móvil.	**Variable 3:** Usabilidad del Sistema	• Facilidad de uso • Satisfacción general Facilidad de aprendizaje
¿Cómo debe diseñarse una aplicación móvil interactiva que facilite el aprendizaje del Lenguaje de Señas Peruano?	Diseñar e implementar una aplicación móvil interactiva para el aprendizaje del Lenguaje de Señas Peruano.		
Metodología			
Tipo de Investigación: Aplicativa **Diseño de la Investigación:** Enfoque cuantitativo y aplicativo **Población:** Dataset de imágenes **Muestra:** 25,000 registros			

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos	: Sergio Alain Ramirez Aguilar
DNI/Carnet de Extranjería/Pasaporte N°	: 72130184
Correo electrónico	: 569SER@gmail.com :
Teléfono/Celular	968755157

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
	X			

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (X)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (X)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING
.....
b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : 1.02.01
c) Fecha de publicación : Año 2026

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : HUGO EULER TITO CHURA
b) DNI/Pasaporte/Cédula : 01315850
c) ORCID : 0000-0001-8269-0655

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 72130184
NOMBRES Y APELLIDOS : Sergio Alain Ramirez Aguilar

"UNIVERSIDAD NACIONAL DE MOQUEGUA"

VICERRECTORADO DE INVESTIGACIÓN

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos	: Jhonny Enrique Andy Vilca Condori
DNI/Carnet de Extranjería/Pasaporte N°	: 73048991
Correo electrónico	: sheikegamer@gmail.com
Teléfono/Celular	: 960427771

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
	X			

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (X)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (X)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING
.....
b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : 1.02.01
c) Fecha de publicación : Año 2026

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : HUGO EULER TITO CHURA
b) DNI/Pasaporte/Cédula : 01315850
c) ORCID : 0000-0001-8269-0655

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 73048991
NOMBRES Y APELLIDOS : Jhonny Enrique Andy Vilca Condori

"UNIVERSIDAD NACIONAL DE MOQUEGUA"

VICERRECTORADO DE INVESTIGACIÓN

LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

(X) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 72130184

NOMBRES Y APELLIDOS : Sergio Alain Ramirez Aguilar

FECHA : 26/01/2026



LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

(X) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 73048991
NOMBRES Y APELLIDOS : Jhonny Enrique Andy Vilca Condori
FECHA : 26/01/2026



ANEXO VIII

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN / TESIS / TRABAJO DE SUFICIENCIA PROFESIONAL

Yo, SERGIO ALAIN RAMIREZ AGUILAR ,
bachiller de la Escuela Profesional de:
INGENIERÍA DE SISTEMAS E INFORMÁTICA ,
de la Universidad Nacional de Moquegua, identificado(a) con DNI N°... 72130184 ..., autor(a) del Trabajo de Investigación / Tesis / Trabajo de suficiencia profesional denominado:

DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING

Declaro bajo juramento que:

1. El trabajo de investigación es auténtico, siendo resultado de mi trabajo personal, que no se ha copiado, que no se ha utilizado ideas, formulaciones, citas integrales e ilustraciones diversas, sacadas de cualquier tesis, obra, artículo, memoria, etc., (en versión digital o impresa), sin mencionar de forma clara y exacta su origen o autor, tanto en el cuerpo del texto, figuras, cuadros, tablas u otros que tengan derechos de autor. Por tanto, el trabajo de investigación no ha sido plagiado total ni parcialmente.
2. El trabajo de investigación no ha sido autoplagiada, es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presentan en el trabajo de investigación constituirán aporte a la realidad investigada.

Soy consciente de que, el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales, por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades en el trabajo de investigación, así como de los derechos sobre la obra presentada.

Moquegua, 26 de Enero del 2026



Huella digital

Nombres y Apellidos: SERGIO ALAIN RAMIREZ AGUILAR
DNI: 72130184

ANEXO VIII

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN / TESIS / TRABAJO DE SUFICIENCIA PROFESIONAL

Yo, JHONNY ENRIQUE ANDY VILCA CONDORI ,
bachiller de la Escuela Profesional de:
INGENIERÍA DE SISTEMAS E INFORMÁTICA ,
de la Universidad Nacional de Moquegua, identificado(a) con DNI Nº... 73048991 , autor(a) del Trabajo de Investigación / Tesis / Trabajo de suficiencia profesional denominado:

DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING

Declaro bajo juramento que:

1. El trabajo de investigación es auténtico, siendo resultado de mi trabajo personal, que no se ha copiado, que no se ha utilizado ideas, formulaciones, citas integrales e ilustraciones diversas, sacadas de cualquier tesis, obra, artículo, memoria, etc., (en versión digital o impresa), sin mencionar de forma clara y exacta su origen o autor, tanto en el cuerpo del texto, figuras, cuadros, tablas u otros que tengan derechos de autor. Por tanto, el trabajo de investigación no ha sido plagiado total ni parcialmente.
2. El trabajo de investigación no ha sido autoplagiada, es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presentan en el trabajo de investigación constituirán aporte a la realidad investigada.

Soy consciente de que, el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales, por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades en el trabajo de investigación, así como de los derechos sobre la obra presentada.

Moquegua, 26 de Enero del 2026



Huella digital

Nombres y Apellidos: JHONNY ENRIQUE ANDY VILCA CONDORI
DNI: 73048991

UNAM
Firmado digitalmente por TITO CHURA Hugo Euler FAU 20449347448 soft
Motivo: Soy el autor del documento
Fecha: 19.11.2025 18:31:40 -05:00

Hugo Euler TITO CHURA

Tesis_sign_language.pdf

Tesis correjida de Sergio

Tesis 2025

Universidad Nacional de Moquegua

Detalles del documento

Identificador de la entrega
trn:oid:::1:3417952499

Fecha de entrega
19 nov 2025, 6:13 p.m. GMT-5

Fecha de descarga
19 nov 2025, 6:30 p.m. GMT-5

Nombre del archivo
Tesis_sign_language.pdf

Tamaño del archivo
1.8 MB

107 páginas

18.800 palabras

111.672 caracteres

3% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 15 palabras)

Fuentes principales

2% Fuentes de Internet
0% Publicaciones
2% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.° de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Fuentes principales

2% Fuentes de Internet
0% Publicaciones
2% Trabajos entregados (trabajos del estudiante)

Fuentes principales

Las fuentes con el mayor número de coincidencias dentro de la entrega. Las fuentes superpuestas no se mostrarán.

1 Internet
hdl.handle.net <1%

2 Internet
repositorio.unac.edu.pe <1%

3 Internet
sedici.unlp.edu.ar <1%

4 Internet
repositorio.uwiener.edu.pe <1%

5 Trabajos del estudiante
Universidad Continental <1%

6 Internet
geekflare.com <1%

7 Trabajos del estudiante
Universidad Europea de Madrid <1%

8 Trabajos del estudiante
Universidad de Málaga - Tii <1%

9 Internet
repositorioacademico.upc.edu.pe <1%

10 Internet
repositorio.unap.edu.pe <1%

11 Internet
repositorio.unjfsc.edu.pe <1%

12 Trabajos del estudiante
Universidad Nacional de Moquegua <1%

13 Internet
dspace.unach.edu.ec <1%

14 Trabajos del estudiante
Universidad Politécnica de Madrid <1%

15 Trabajos del estudiante
Escuela Politecnica Nacional <1%

16 Trabajos del estudiante
Universidad Carlos III de Madrid - EUR <1%

17 Trabajos del estudiante
Universidad Internacional de la Rioja <1%

18 Internet
intra.uigv.edu.pe <1%

19 Internet
pingpdf.com <1%

20 Trabajos del estudiante
ucr <1%

21 Trabajos del estudiante
unifranz <1%

Universidad Nacional de Moquegua

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 098-2026-CU

Moquegua, 13 de febrero de 2026

VISTOS:

El Proveído N° 359-2026-R de fecha 10 de febrero de 2026 emitido por el Rectorado, Informe N° D0063-2026-UNAM-R-SG de fecha 10 de febrero de 2026 emitido por el Secretario General, Informe Técnico N° D0018-2026-SEGE-GT de Grados y Títulos, el Acuerdo del Pleno del Consejo Universitario de Sesión Extraordinaria continuada N° 07-2026-SE-CU-UNAM, de fecha 12 de febrero de 2026, y;

CONSIDERANDO:

Que, conforme al 4to párrafo del artículo 18, de la Constitución Política del Perú, dispone que cada Universidad es autónoma en su régimen normativo, de gobierno, académico, administrativo y económico. Y el artículo 8° de la Ley Nº 30220-Ley Universitaria, establece que la referida autonomía inherente a las Universidades, se ejerce de conformidad con la Constitución y demás normativa aplicable. Concordante con el artículo 1, de la Ley Nº 31520, Ley que restablece la autonomía y la institucionalidad de las universidades peruanas, así como la Ley N° 32141, Ley que modifica la Ley N° 30220-Ley Universitaria, a fin de fortalecer la institucionalidad de las Universidades Públicas licenciadas;

Que, el Artículo 58, de la Ley N° 30220-Ley Universitaria, establece que el Consejo Universitario es el máximo órgano de gestión, dirección y ejecución académica y administrativa de la Universidad. Concordante con el artículo 18° del Estatuto de la Universidad Nacional de Moquegua, aprobado bajo Resolución de Asamblea Universitaria N° 015-2025-AU-UNAM;

Que, el artículo 59 de la Ley Universitaria-Ley N° 30220, establece las atribuciones del Consejo Universitario, y específicamente el numeral 59.9, establece; Conferir los grados académicos y los títulos profesionales aprobados por las Facultades y Escuela de Posgrado, así como otorgar distinciones honoríficas y reconocer y revalidar los estudios, grados y títulos de universidades extranjeras, cuando la universidad está autorizada por la Superintendencia Nacional de Educación Superior Universitaria. Concordante con el numeral 14, del artículo 20° del Estatuto de la Universidad Nacional de Moquegua;

Que, el Sr. Rector es el personero y representante legal de la Universidad Nacional de Moquegua, conforme a lo dispuesto por la Ley Nº 30220-Ley Universitaria, tiene a su cargo la Dirección, Conducción y Gestión del Gobierno Universitario en todos sus ámbitos, concordante con el artículo 24° y numeral 2 del artículo 26° del Estatuto de la Universidad Nacional de Moquegua;

Que, mediante Informe Técnico N° D0018-2026-SEGE-GT de fecha 10 de febrero de 2026, e Informe N° D0063-2026-UNAM-R-SG de fecha 10 de febrero de 2026, el Secretario General de la UNAM, remite al Rectorado los expedientes para el trámite de Título Profesional de Ingeniero de Sistemas e Informática a los Bachilleres: RAMIREZ AGUILAR SERGIO ALAIN, VILCA CONDORI JHONNY ENRIQUE ANDY y MAMANI GUTIERREZ ESTEFANY PAOLA de la Escuela Profesional de Ingeniería de Sistemas e Informática, quienes se encuentran aptos para la expedición del Título Profesional de Ingeniero de Sistemas e Informática, al haber cumplido satisfactoriamente con los requisitos legales y reglamentarios, exigidos en el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, conforme lo señala la Dirección de Actividades y Servicios Académicos y la Dirección de la Escuela Profesional de Ingeniería de Sistemas e Informática, razón por la cual, el señor Secretario General incluye dichos expedientes como punto de agenda para ser tratado en Sesión Extraordinaria continuada de Consejo Universitario de fecha 12 de febrero de 2026;

Que, el artículo 44° de la Ley N° 30220, Ley Universitaria señala que las universidades otorgan los grados académicos de Bachiller, Maestro, Doctor y los Títulos Profesionales que correspondan, a nombre de la Nación (...). Asimismo, la Ley N° 31803, modifica el artículo en mención adicionando; La Superintendencia Nacional de Educación Superior Universitaria (SUNEDU) determina los criterios, así como establece y supervisa el procedimiento administrativo para el reconocimiento de grados y títulos otorgados en el extranjero, en el marco de la normativa vigente;

Que, el numeral 45.2 del artículo 45° de la Ley Nº 30220, Ley Universitaria, precisa los requisitos mínimos para la obtención del título profesional, como son: el grado de Bachiller y la aprobación de una tesis o trabajo de suficiencia profesional. Asimismo, establece que el título profesional solo se puede obtener en la universidad en la cual se haya obtenido el grado de bachiller;

Que, mediante Resolución de Comisión Organizadora N° 1153-2021-UNAM, se aprobó el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, el mismo que fue modificado a través de las Resoluciones de Comisión Organizadora N° 410; 453 y 890-2022, N° 880-2023-UNAM y N° 081, 484-2024-UNAM el cual establece los requisitos y norma

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 098-2026-CU

Moquegua, 13 de febrero de 2026

los procedimientos para la obtención de los grados académicos y títulos profesionales conferidos por esta Casa Superior de Estudios;

Que, bajo el amparo de los considerandos precedentes, el Pleno del Consejo Universitario en Sesión Extraordinaria continuada N° 07-2026-SE-CU-UNAM, de fecha 12 de febrero de 2026, emitió, el siguiente, Acuerdo, POR UNANIMIDAD: CONFERIR, a nombre de la Nación el Título Profesional de Ingeniero de Sistemas e Informática por la modalidad de sustentación de tesis a los Bachilleres: RAMIREZ AGUILAR SERGIO ALAIN, VILCA CONDORI JHONNY ENRIQUE ANDY y MAMANI GUTIERREZ ESTEFANY PAOLA, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Facultad de Ingeniería y Arquitectura de la Universidad Nacional de Moquegua;

Que, mediante la Resolución Rectoral Nº 048-2026-R-UNAM de fecha 30 de enero de 2026, se resolvió ENCARGAR, al Dr. Alejandro Manuel Ecos Espino, en su calidad de Vicerrector Académico el Despacho de Rectorado de la Universidad Nacional de Moquegua, para que ejerza las funciones correspondientes en ausencia del Rector, durante los días 02 al 11 de febrero de 2026, por tanto, en aplicación de lo dispuesto por el numeral 5 del artículo 30 del Estatuto de la Universidad Nacional de Moquegua, y conforme a los considerandos del referido acto administrativo, el señor Vicerrector Académico asume el cargo, con todas las prerrogativas de la autoridad;

Que, a través del Memorando N° D0099-2026-UNAM-R de fecha 12 de febrero de 2026, el señor Rector, encarga las funciones de la jefatura de Secretaría General al Abg. Richard Alex Flores Perez, Jefe de la Oficina de Asesoría Jurídica, a partir del 12 de febrero, hasta la designación en el cargo por la autoridad universitaria;

En uso de las facultades y atribuciones conferidas por el Art. 18, de la Constitución Política del Perú, la Ley N° 30220 Ley Universitaria, el Estatuto de la Universidad Nacional de Moquegua, y a lo acordado por el pleno del Consejo Universitario:

SE RESUELVE:

ARTÍCULO 1°: CONFERIR, A NOMBRE DE LA NACIÓN a los Bachilleres

- **RAMIREZ AGUILAR SERGIO ALAIN**
  TESIS : "DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING", cuyo Asesor Principal fue el MSc. Hugo Euler Tito Chura.
- **VILCA CONDORI JHONNY ENRIQUE ANDY**
  TESIS : "DESARROLLO DE UNA APLICACIÓN MÓVIL DIDÁCTICA PARA EL APRENDIZAJE DEL LENGUAJE DE SEÑAS PERUANO USANDO DEEP LEARNING", cuyo Asesor Principal fue el MSc. Hugo Euler Tito Chura.
- **MAMANI GUTIERREZ ESTEFANY PAOLA**
  TESIS : "TÉCNICAS DE BALANCE DE DATOS PARA OPTIMIZAR EL RENDIMIENTO DE MODELOS DE APRENDIZAJE PROFUNDO EN CLASIFICACIÓN DE RADIOGRAFÍAS TORÁCICAS", cuyo Asesor Principal fue el Mgr. JUAN CARLOS CLARES PERCA, y el Coasesor el Ing. Diego Ismael Mamani Padilla.

El **TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS E INFORMÁTICA.**

Por la modalidad de SUSTENTACIÓN DE TESIS, de la Escuela Profesional de Ingeniería de Sistemas e Informática, de la Facultad de Ingeniería y Arquitectura de la Universidad Nacional de Moquegua.

ARTÍCULO 2°: ENCARGAR, al Secretario General a través de quien corresponda, para que realice las acciones pertinentes relacionadas con la inscripción correspondiente en el Registro de Grados y Títulos de la Superintendencia Nacional de Educación Superior Universitaria - SUNEDU.

ARTÍCULO 3°: DISPONER, el cumplimiento de la presente Resolución a todas las dependencias de la Universidad Nacional de Moquegua.

ARTÍCULO 4°: DISPONER la publicación de la presente resolución en el Portal Institucional de la Universidad Nacional de Moquegua www.unam.edu.pe

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

RESOLUCIÓN DE CONSEJO UNIVERSITARIO
N° 098-2026-CU

Moquegua, 13 de febrero de 2026

REGÍSTRESE, PUBLÍQUESE Y CÚMPLASE;

ABG. RICHARD ALEX FLORES PEREZ
SECRETARIO GENERAL (e)

DR. HUGO RUBEN MARCA MAQUERA
RECTOR
PRESIDENTE DEL CONSEJO UNIVERSITARIO

DISTRIBUCION: VRA , VRI, FAIN, EPIS, GyT, DASA, OTI Interesados, c.c. /Arch.

Domicilio Legal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe