



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

**“CONSTRUCCIÓN DE UN PROTOTIPO ELECTRÓNICO
TRADUCTOR DE LENGUAJE DE SEÑAS A VOZ CON BASE EN
PROCESAMIENTO DE IMÁGENES”.**

Trabajo de Integración Curricular

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES: JAVIER MOISES BUÑAY CUJILEMA

JOSE EMILIO MULLO YAUTIBUG

DIRECTOR: ING. JORGE LUIS PAUCAR SAMANIEGO

Riobamba - Ecuador

2023

©2023, Javier Moises Buñay Cujilema, Jose Emilio Mullo Yautibug

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Nosotros, Javier Moises Buñay Cujilema y Jose Emilio Mullo Yautibug, declaramos que el presente Trabajo de Integración Curricular es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.


Como autores asumimos la responsabilidad legal y académica de los contenidos de este Trabajo de Integración Curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 21 de noviembre de 2023.



Javier Moises Buñay Cujilema

060480094-6

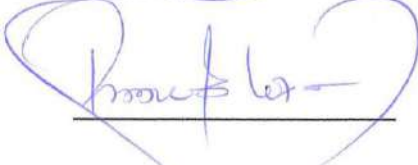


Jose Emilio Mullo Yautibug

060534044-7

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Integración Curricular certifica que: El Trabajo de Integración Curricular; Tipo: Dispositivo Tecnológico, “**CONSTRUCCIÓN DE UN PROTOTIPO ELECTRÓNICO TRADUCTOR DE LENGUAJE DE SEÑAS A VOZ CON BASE EN PROCESAMIENTO DE IMÁGENES**”, realizado por los señores **JAVIER MOISES BUÑAY CUJILEMA** y **JOSE EMILIO MULLO YAUTIBUG**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Nancy del Rocío Velasco Erazo PRESIDENTE DEL TRIBUNAL		2023-11-21
Ing. Jorge Luis Paucar Samaniego DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-11-21
Ing. Pablo Eduardo Lozada Yáñez ASESOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		2023-11-21

DEDICATORIA

Dedico este trabajo al esfuerzo de toda mi familia, ya que sin ellos no habría logrado mis metas profesionales. Agradezco de manera especial a mis padres, Carlos Buñay y Josefa Cujilema, por el apoyo inquebrantable que me proporcionó la fortaleza necesaria para avanzar en mi camino universitario.

Mi reconocimiento también se extiende a mis hermanos, cuyo respaldo y amistad han sido un regalo invaluable. Les agradezco por compartir este viaje conmigo y por ser una fuente constante de alegría y motivación.

Javier

El presente Trabajo de Integración Curricular está dedicado a mi Dios, por haberme guiado e iluminado en todas las decisiones de mi vida, a mis padres Gerardo Mullo y Rosa Yautibug, pero de manera especial a mi madre ya que, gracias a su esfuerzo, sus consejos y palabras de aliento me inspiró a culminar esta meta importante en mi vida, les agradezco infinitamente por brindarme ese amor y apoyo incondicional hoy, mañana y siempre. A mis hermanos Mariano, Remigio y Manuel y a mis hermanas Sara, Olga y Rebeca por darme fortaleza en todo momento. Y finalmente, a todos aquellos que creen en el poder del conocimiento y la perseverancia para transformar realidades.

Jose

AGRADECIMIENTO

Agradezco principalmente a Dios por otorgarme la bendición de la vida y por brindarme la fuerza necesaria para enfrentar las adversidades que han surgido a lo largo de mi vida, tanto en mi desarrollo universitario como en lo personal. También, expreso mi gratitud hacia mis profesores, cuya dedicación y valiosos conocimientos han influido significativamente en mi progreso diario como profesional.

Javier

Agradezco a Dios por otorgarme salud, vida, fortaleza y perseverancia, permitiéndome superar los momentos desafiantes de mi trayectoria académica y persistir hasta la culminación de mis estudios. Deseo expresar un sincero y cálido agradecimiento a mi círculo familiar y social, quienes confiaron en mí y me brindaron el apoyo y aliento necesarios para alcanzar esta meta. También, quiero expresar mi gratitud a los docentes de la ESPOCH por su valiosa contribución en mi formación académica, la cual ha culminado en la realización de este trabajo de Integración Curricular. Es relevante resaltar a mi tutor en particular, cuya dedicación y pasión por la enseñanza han sido evidentes a lo largo de la elaboración de este documento.

Jose

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLA	xi
ÍNDICE DE ILUSTRACIÓN	xii
ÍNDICE DE ECUACIONES	xiv
ÍNDICE DE ABREVIATURAS.....	xv
ÍNDICE DE ANEXOS	xvi
RESUMEN.....	xvii
SUMMARY	xviii
INTRODUCCIÓN	1

CAPITULO I

1.	DIAGNÓSTICO DEL PROBLEMA	3
1.1.	Planteamiento del problema	3
1.1.1.	<i>Sistematización del problema</i>	5
1.2.	Justificación.....	5
1.2.1.	<i>Justificación Teórica</i>	5
1.2.2.	<i>Justificación Aplicativa</i>	6
1.3.	Objetivos.....	6
1.3.1.	<i>Objetivo general</i>	6
1.3.2.	<i>Objetivos específicos</i>	6

CAPÍTULO II

2.	MARCO TEÓRICO.....	7
2.1.	Discapacidad	7
2.1.1.	<i>Discapacidad Auditiva</i>	9
2.1.2.	<i>Discapacidad en Ecuador</i>	10
2.2.	Lengua de Señas.....	12
2.2.1.	<i>Lengua de Señas Ecuatoriana (LSEC)</i>	12
2.3.	Inteligencia Artificial (IA).....	13

2.3.1.	Aplicaciones de la Inteligencia Artificial	13
2.3.2.	Visión Artificial	15
2.3.2.1.	<i>Etapas de la visión artificial</i>	15
2.3.3.	Red Neuronal Artificial (RNA)	15
2.3.3.1.	<i>Elementos de las Redes Neuronales Artificiales</i>	16
2.3.4.	Aprendizaje Automático (Machine Learning, ML)	16
2.3.4.1.	<i>Recopilar de datos</i>	17
2.3.4.2.	<i>Preparar los datos para el entrenamiento</i>	17
2.3.4.3.	<i>Seleccionar un modelo</i>	17
2.3.4.4.	<i>Entrenar el modelo</i>	17
2.3.4.5.	<i>Evaluación</i>	17
2.3.5.	El Aprendizaje Profundo (Deep Learning, DL)	17
2.4.	Investigación de traductores de lenguaje de señas a nivel nacional	18
2.4.1.	<i>Arquitectura de los dispositivos traductores de la LSEC</i>	20
2.5.	Tarjetas de desarrollo	21
2.5.1.	<i>Análisis comparativo de las tarjetas de desarrollo</i>	24
2.6.	Hardware	24
2.6.1.	<i>Hardware de adquisición de datos</i>	25
2.6.1.1.	<i>Análisis comparativo de cámaras de video</i>	25
2.6.2.	<i>Hardware de emisión de datos</i>	26
2.6.2.1.	<i>Tipos de pantalla</i>	26
2.7.	Software	27
2.7.1.	<i>Open Source</i>	27
2.7.2.	<i>Python</i>	27
2.7.3.	<i>OpenCV (Open-Source Computer Vision Library)</i>	27
2.7.4.	<i>TensorFlow</i>	28
2.7.5.	<i>MediaPipe</i>	28
2.7.5.1.	<i>Mediapipe Hands</i>	28

CAPÍTULO III

3.	MARCO METODOLÓGICO	30
3.1.	Requerimientos del prototipo del traductor de lenguaje de señas a voz	30
3.2.	Concepción del diseño general del prototipo traductor de lenguaje de señas a voz	30
3.3.	Diseño de bloques del prototipo traductor de lenguaje de señas a voz	31
3.3.1.	<i>Bloque de adquisición de datos</i>	32

3.3.2.	<i>Bloque de procesamiento de datos</i>	32
3.3.3.	<i>Bloque de visualización y reproducción de audio de la información</i>	32
3.4.	Selección de hardware	32
3.4.1.	<i>Cámara ELP USB</i>	32
3.4.2.	<i>Tarjeta de desarrollo Jetson Nano 4GB B01</i>	33
3.4.3.	<i>Pantalla ELECROW IPS 800 X 480 USB</i>	34
3.5.	Esquema de conexión electrónica	35
3.6.	Selección de software de desarrollo	36
3.6.1.	<i>Adquisición de datos de aprendizaje</i>	36
3.6.2.	Desarrollo de la arquitectura y entrenamiento de la red Neuronal	41
3.6.2.1.	<i>Red Neuronal Feedforward (FNN)</i>	41
3.6.2.2.	<i>Perceptrón Multicapa (MLP)</i>	42
3.6.2.3.	<i>Aprendizaje supervisado</i>	42
3.6.2.4.	<i>Funciones de activación</i>	42
3.6.2.5.	<i>Función ReLU</i>	43
3.6.2.6.	<i>Función SOFTMAX</i>	43
3.6.2.7.	<i>Algoritmo de aprendizaje Backpropagation</i>	44
3.6.3.	Proceso de traducción y visualización	48
3.7.	Diseño estructural	51

CAPÍTULO IV

4.	VALIDACIÓN DE PROTOTIPO	53
4.1.	Pruebas del desempeño de la red neuronal	53
4.1.1.	<i>Métricas de evaluación de la matriz de confusión</i>	54
4.1.2.	<i>Evaluación de matriz de confusión del prototipo traductor de lenguaje de señas</i> ..	55
4.2.	Pruebas de aciertos y porcentaje de precisión de detección con relación a la distancia utilizando la mano derecha	56
4.2.1.	Pruebas a una distancia de 30 cm	57
4.2.1.1.	<i>Pruebas de la letra A</i>	57
4.2.1.2.	<i>Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)</i>	58
4.2.2.	Pruebas a una distancia de 60 cm	60
4.2.2.1.	<i>Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)</i>	60
4.2.3.	Pruebas a una distancia de 100 cm	62
4.2.3.1.	<i>Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)</i>	62
4.3.	Pruebas de aciertos y porcentaje de precisión de detección con relación a la distancia utilizando la mano Izquierda	64

4.3.1.	<i>Pruebas a una distancia de 30 cm</i>	65
4.3.1.1.	<i>Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)</i>	65
4.3.2.	<i>Pruebas a una distancia de 60 cm</i>	67
4.3.2.1.	<i>Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)</i>	67
4.3.3.	<i>Pruebas a una distancia de 100 cm</i>	68
4.3.3.1.	<i>Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)</i>	68
4.4.	Análisis estadísticos de las pruebas	70
4.4.1.1.	<i>Pruebas de Normalidad</i>	71
4.4.1.2.	<i>Prueba de homogeneidad y varianza.</i>	71
4.4.1.3.	<i>ANOVA</i>	72
4.5.	Pruebas de tiempo de procesamiento de datos	73
4.6.	Pruebas de consumo energético del prototipo traductor	75
4.7.	Análisis de Costos	76
	CONCLUSIONES	78
	RECOMENDACIONES	80
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLA

Tabla 1-2:	Aspectos generales en la evaluación de la discapacidad auditiva	7
Tabla 2-2:	Aspectos generales en la evaluación de la discapacidad visual	8
Tabla 3-2:	Aspectos generales en la evaluación de la discapacidad física	8
Tabla 4-2:	Aspectos generales en la evaluación de la discapacidad intelectual	9
Tabla 5-2:	Datos de personas con discapacidad auditiva	12
Tabla 6-2:	Aplicaciones de la inteligencia artificial	14
Tabla 7-2:	Características de los diferentes modelos de la Rasberry Pi	21
Tabla 8-2:	Características de los diferentes modelos de la Jetson Nano	22
Tabla 9-2:	Características de la Google Coral Dev Board	23
Tabla 10-2:	Comparativa de las diferentes tarjetas de desarrollo	24
Tabla 11-2:	Características de los diferentes tipos de cámaras	25
Tabla 12-2:	Comparativa de los diferentes modelos de pantallas	26
Tabla 1-3:	Especificaciones técnicas de la cámara ELP USB.....	33
Tabla 2-3:	Especificaciones técnicas de la Jetson Nano 4GB B01	34
Tabla 3-3:	Especificaciones técnicas de la pantalla ELECROW	35
Tabla 1-4:	Resultados de las pruebas a una distancia de 30 cm	59
Tabla 2-4:	Resultados de las pruebas a una distancia de 60 cm	61
Tabla 3-4:	Resultados de las pruebas a una distancia de 100 cm	63
Tabla 4-4:	Resultados de las pruebas a una distancia de 30 cm	65
Tabla 5-4:	Resultados de las pruebas a una distancia de 60 cm	67
Tabla 6-4:	Resultados de las pruebas a una distancia de 100 cm	69
Tabla 7-4:	Pruebas de normalidad	71
Tabla 8-4:	Pruebas de homogeneidad de varianzas.	72
Tabla 9-4:	Pruebas de ANOVA	72
Tabla 10-4:	Pruebas de muestras independientes	73
Tabla 11-4:	Pruebas de tiempo de reproducción	74
Tabla 12-4:	Resultados de mediciones de consumo energético	75
Tabla 13-4:	Consumo eléctrico individual	76
Tabla 14-4:	Costos del prototipo	77

ÍNDICE DE ILUSTRACIÓN

Ilustración 1-2:	Registro de Discapacidad en Ecuador	10
Ilustración 2-2:	Registro de Discapacidad en la Provincia de Chimborazo	11
Ilustración 3-2:	Registro de Discapacidad en el Cantón Riobamba	11
Ilustración 4-2:	Abecedario de la Lengua de Señas Ecuatoriana	13
Ilustración 5-2:	Esquema de la red neuronal	16
Ilustración 6-2:	Reconocimiento Esquelético	18
Ilustración 7-2:	Guante traductor de lenguaje de señas	19
Ilustración 8-2:	Sistema traductor de lenguaje de señas con Visión Artificial	20
Ilustración 9-2:	Arquitectura de los dispositivos estudiados	21
Ilustración 10-2:	Puntos de referencia de la mano que utiliza Medipipe Hands	29
Ilustración 1-3:	Concepción general del prototipo traductor de lenguaje de señas	31
Ilustración 2-3:	Concepción del prototipo por bloques	31
Ilustración 3-3:	ELP Cámara USB	33
Ilustración 4-3:	Jetson Nano 4GB	33
Ilustración 5-3:	Pantalla ELECROW de 5 pulgadas	34
Ilustración 6-3:	Esquema de conexión electrónica	36
Ilustración 7-3:	Diagrama de flujo de la adquisición de los datos de aprendizaje	37
Ilustración 8-3:	Parámetros de configuración de la función cv.VideoCapture()	39
Ilustración 9-3:	Parámetros de configuración de la función mp_hands.Hands()	40
Ilustración 10-3:	Detección de los puntos de las dos manos	40
Ilustración 11-3:	Seguimiento de la palma de la mano	40
Ilustración 12-3:	Esquema de una neurona artificial	41
Ilustración 13-3:	Función de activación ReLU	43
Ilustración 14-3:	Función de activación SOFTMAX	44
Ilustración 15-3:	Algoritmo Backpropagation	44
Ilustración 16-3:	Diagrama de flujo del entrenamiento de la red neuronal	45
Ilustración 17-3:	Configuración de la función train test	46
Ilustración 18-3:	Configuración de la función tf.keras.models.Sequential	46
Ilustración 19-3:	Esquema de la red neuronal	47
Ilustración 20-3:	Configuración de la función model.fit	47
Ilustración 21-3:	Función tf.lite.TFLiteConverter.from_keras_()	47
Ilustración 22-3:	Diagrama de flujo del proceso de traducción	48
Ilustración 23-3:	Bibliotecas gtts y pygame para la traducción de texto a voz	50
Ilustración 24-3:	Concatenación de la palabra HOLA	50

Ilustración 25-3:	Interfaz final del prototipo	50
Ilustración 26-3:	Vista externa del prototipo en SolidWorks	51
Ilustración 27-3:	Vista interna del prototipo en SolidWorks	51
Ilustración 28-3:	Implementación final del prototipo.....	52
Ilustración 1-4:	Matriz de confusión	53
Ilustración 2-4:	Matriz de confusión del prototipo traductor	55
Ilustración 3-4:	Reporte del entrenamiento de la red neuronal	56
Ilustración 4-1:	Pruebas de aciertos y precisión de detección a una distancia de 30 cm para la letra “A” con la mano derecha	57
Ilustración 5-4:	Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm.....	60
Ilustración 6-4:	Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm	60
Ilustración 7-4:	Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm	62
Ilustración 8-4:	Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm	62
Ilustración 9-4:	Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm	64
Ilustración 10-4:	Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm	64
Ilustración 11-4:	Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm	66
Ilustración 12-4:	Porcentajes de precisión de detección del alfabeto de la Lengua de Señas a 30 cm	66
Ilustración 13-4:	Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm	68
Ilustración 14-4:	Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm	68
Ilustración 15-4:	Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm.....	70
Ilustración 16-4:	Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm	70

ÍNDICE DE ECUACIONES

Ecuación 1:	Salida (S) de la Red Neuronal	16
Ecuación 2:	Función de ponderación	42
Ecuación 3:	Función de activación	42
Ecuación 4:	Función de activación ReLU.	43
Ecuación 5:	Función de activación SOFTMAX	43
Ecuación 6:	Métrica Exactitud (Accuracy).....	54
Ecuación 7:	Métrica Precisión (Precision)	54
Ecuación 8:	Métrica Recuperación o Sensibilidad (Recall o Sensitivity)	54
Ecuación 9:	Métrica Puntuación (F1 Score)	55
Ecuación 10:	Métrica de Especificidad (Specificity)	55
Ecuación 11:	Media de porcentaje de precisión	58
Ecuación 12:	Consumo eléctrico en kilovatios por hora (kWh)	76

ÍNDICE DE ABREVIATURAS

OMS	Organización Mundial de la Salud
LSEC	Lengua de Señas Ecuatoriana
IA	Inteligencia Artificial
RNA	Redes Neuronales Artificiales
GPU	Graphics Processing Unit
GPIO	General-Purpose Input/Output
UPHY	Universal PHY
USB	Universal Serial Bus
CSV	Comma-Separated Values
UVC	Universal Video Class controller
CONADIS	Consejo Nacional para la Igualdad de Discapacidades
FPS	Fotogramas Por Segundo
MP	Megapíxeles
cm	Centímetros
ML	Machine Learning
DL	Deep Learning
PLA	Filamento Bioplástico

ÍNDICE DE ANEXOS

- ANEXO A:** Datos de las pruebas de acierto y porcentajes de precisión en la detección del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm, 60 cm y 100 cm.
- ANEXO B:** Datos estadísticos en el software IBM SPSS.
- ANEXO C:** Mediciones del consumo energético.
- ANEXO D:** Pruebas de funcionamiento del prototipo.
- ANEXO E:** Solicitud de acceso a la Unidad Educativa Especializada Sordos de Chimborazo.
- ANEXO F:** Autorización de acceso a la Unidad Educativa Especializada Sordos de Chimborazo.
- ANEXO G:** Recibido por parte de la Unidad Educativa Especializada Sordos de Chimborazo.
- ANEXO H:** Cédula de identidad del docente que evaluó el prototipo.
- ANEXO I:** Pruebas realizadas al prototipo por parte del docente de la Unidad Educativa Especializada Sordos de Chimborazo.
- ANEXO J:** Código de programación.

RESUMEN

El objetivo de la presente investigación fue la construcción de un prototipo electrónico traductor de lenguaje de señas a voz con base en procesamiento de imágenes. Este prototipo se compone de tres bloques fundamentales: uno de adquisición de datos, que incorpora una cámara USB; un bloque de procesamiento, compuesto por una tarjeta de desarrollo Jetson Nano, donde se lleva a cabo el procesamiento de las imágenes capturadas; y finalmente, un bloque de visualización y reproducción de la información, compuesto por una pantalla y un altavoz, respectivamente. El bloque central extrae datos de las imágenes, específicamente de los 21 nudillos de la mano. Estos datos permiten la traducción de las letras del alfabeto de la Lengua de Señas Ecuatoriana al alfabeto común, lo que posibilita la generación de una amplia variedad de palabras mediante la concatenación de estas letras. Finalmente, estas palabras se presentan al usuario en formato de texto y sonido. Como resultados de las pruebas realizadas a distancias de 30 cm, 60 cm y 100 cm indican que su mejor rendimiento se alcanzó a una distancia de 30 cm, logrando un porcentaje de precisión en la detección del 93.21%. Con lo cual se concluye que el prototipo desarrollado logra una buena traducción, constituyéndose en una herramienta de ayuda para mejorar la comunicación de personas con discapacidad auditiva con aquellas que no la tienen. Se recomienda contar con iluminación adecuada en el entorno de captura y evitar situaciones de contraluz debido a que algunas condiciones de iluminación pueden hacer que sea difícil para el prototipo identificar las señas y gestos realizados por el usuario.

Palabras claves: <VISIÓN ARTIFICIAL>, <COMUNICACIÓN>, <DISCAPACIDAD AUDITIVA>, <LENGUAJE DE SEÑAS>, <LENGUAJE ORAL>.



SUMMARY

The objective of the present research was to construct an electronic prototype for translating sign language into speech based on image processing. This prototype is made up of three fundamental blocks: one for data acquisition, which includes a USB camera; a processing block, composed of a Jetson Nano development board, where the processing of the captured images is carried out; and finally, a block for displaying and reproducing information, composed of a screen and a speaker, respectively. The central block extracts data from the images, specifically from the 21 knuckles of the hand. These data allow the translation of the letters of the Ecuadorian Sign Language alphabet into the standard alphabet, enabling the generation of a wide variety of words by the concatenation of these letters. Finally, these words are presented to the user in text and sound formats. The results of the tests conducted at distances of 30 cm, 60 cm, and 100 cm indicate that the best performance was achieved at a distance of 30 cm, with a detection accuracy of 93.21%. It is concluded that the developed prototype achieves good translation, making it a valuable tool to improve communication for individuals with hearing impairment with those who do not have such disabilities. It is recommended to have adequate lighting in the capture environment and avoid backlight situations because some lighting conditions can make it difficult for the prototype to identify the signs and gestures made by the user.

Keywords: <ARTIFICIAL VISION>, <COMMUNICATION>, <HEARING IMPAIRMENT>, <SIGN LANGUAGE>, <ORAL LANGUAGE>.



Lenin Iván Lara Olivo

0602546103

INTRODUCCIÓN

La comunicación principalmente se lleva de forma verbal, pero no todas las personas pueden hacerlo debido a limitaciones como la sordera o la mudez. La sordera puede deberse a factores genéticos, complicaciones en el nacimiento, enfermedades infecciosas, infecciones crónicas del oído, uso de medicamentos, exposición prolongada a ruidos fuertes y el proceso de envejecimiento (OMS, 2020). En cierto modo, la mudez puede ser resultado de situaciones como intubación endotraqueal, traqueotomía o daños en las cuerdas vocales causados por enfermedades o lesiones (Tietze, 2011). Es importante notar que en ciertos casos la mudez puede ser una consecuencia directa de la sordera.

En la actualidad, se estima que alrededor de 466 millones de personas en todo el mundo sufren de pérdida auditiva, de las cuales 34 millones son niños. Proyecciones sugieren que para el año 2050, la cifra de personas con pérdida auditiva podría ascender a 900 millones (OMS, 2020). En Ecuador, se encuentran registradas 54,244 personas con discapacidad auditiva (CONADIS, 2022). En vista de la notable cantidad de individuos afectados por la pérdida de audición, se prevé que una cifra similar de personas enfrentará dificultades para hablar. A pesar de los diversos métodos y medidas de prevención desarrollados, aquellos que lamentablemente viven con estas discapacidades dependen en gran medida del lenguaje de señas como medio de comunicación. La lengua de señas tiene su origen desde el siglo V a.C., como ilustró Sócrates al expresar: *"Si careciéramos de voz y lengua, pero quisiéramos transmitirnos pensamientos, ¿no intentaríamos comunicarnos mediante gestos con las manos, la cabeza y el resto del cuerpo, como hacen los mudos en la actualidad?"* (Ohna, 2008). A lo largo de la historia, han emergido diversas versiones de la Lengua de Señas, Sin embargo, el enfoque de este trabajo se centra en la Lengua de Señas Ecuatoriana (LSEC).

Mediante la Lengua de Señas, los sordos y los mudos pueden comunicarse de algún modo. Solo algunas personas entienden la Lengua de Señas. Si los sordomudos necesitan hablar en público, suelen recurrir a la ayuda de un traductor.

Bajo estos preliminares, en el presente trabajo se lleva a cabo la construcción de un dispositivo que traduce la LSEC a voz mediante el procesamiento de imágenes. El objetivo principal de este trabajo es eliminar las barreras de comunicación y promover la inclusión de las personas sordas y aquellas con dificultades en el habla. Esto se logrará permitiendo que las personas con discapacidad auditiva se comuniquen de mejor manera con aquellos que no están familiarizados con la LSEC, mediante la utilización de dispositivos tecnológicos, visión artificial y redes

neuronales. El presente documento está descrito en cuatro capítulos, lo cuales comprenden el presente trabajo, en donde:

En el capítulo I, denominado ‘Diagnóstico del Problema’, se abordan varios puntos que sirven como base para el desarrollo del tema propuesto. Estos incluyen antecedentes, planteamiento del problema, justificación teórica, justificación aplicativa y los objetivos, tanto generales como específicos.

En el capítulo II, se presentan las bases teóricas necesarias para el desarrollo de este trabajo. Se revisan los diferentes tipos de traductores de lenguaje de señas, los distintos tipos de discapacidades, la lengua de señas, las etapas de la visión artificial, las redes neuronales, la tecnología y otros componentes adecuados. Esta revisión orienta la elección de la tecnología para el prototipo y proporciona al lector una comprensión más profunda de los conceptos relacionados

En el Capítulo III, titulado ‘Marco Metodológico’, se aborda el desarrollo del prototipo, incluyendo sus etapas, concepción y una descripción de los elementos de *hardware* y *software* empleados. Además, se detalla el diseño del prototipo y la visualización y reproducción de la información.

En el Capítulo IV, denominado ‘Pruebas y Resultados’, se realiza la validación del prototipo, lo cual incluye pruebas de desempeño de la red neuronal, funcionamiento, detección, tiempo de procesamiento, estadísticas y consumo energético.

CAPITULO I

1. DIAGNÓSTICO DEL PROBLEMA

En el siguiente capítulo se analiza el planteamiento del problema, la justificación teórica y aplicativa, los objetivos: general y específicos.

1.1. Planteamiento del problema

En la sociedad actual, resulta complicado detenerse a reflexionar sobre la situación de las personas con discapacidades, tanto físicas como motoras. La globalización continúa en aumento, tal como lo indica la OMS, y según las estadísticas mundiales, alrededor del 5% de la población mundial, aproximadamente 360 millones de personas, enfrentan algún grado de pérdida auditiva discapacitante. De esta cifra, aproximadamente 328 millones corresponden a adultos, mientras que 32 millones son niños. Es importante destacar que la mayoría de las personas con pérdida auditiva viven en países con bajos y medianos ingresos económicos (El Telégrafo, 2022).

Las lenguas de señas no son universales; cada una es una lengua distinta. Por ejemplo, la Lengua de Señas Estadounidense (ASL) tiene su pariente más cercano en la Lengua de Señas Francés (LSF). Además, existen otras lenguas de señas como la Lenguaje de Señas Brasileña (LSB), la Lengua de Señas Quebequense (LSQ), la Lengua de Señas Mexicana (LSM) y la Lengua de Señas Ecuatoriana (LSEC).

En 2020, en la Universidad de California en Los Ángeles (UCLA), un equipo de bioingenieros investigadores buscó facilitar la comunicación de las personas sordas con su entorno mediante la creación de un guante que traduce el lenguaje de señas en tiempo real. De acuerdo con la publicación realizada en la revista Nature Electronic, este dispositivo trabaja en conjunto con una aplicación para Smartphone con la capacidad de traducir las señas en tiempo real. El guante se encarga de rastrear las posiciones de los dedos, y en menos de un segundo, envía esta información como señales al procesador. El procesador es capaz de traducir las señas y enviar los resultados mediante bluetooth a la aplicación del Smartphone, la cual reproduce la palabra traducida a través de un altavoz. En la conclusión, se presentó que el dispositivo puede comunicar 600 signos norteamericanos con una precisión del 98,63%. No obstante, es importante mencionar que este dispositivo no es universal y está diseñado exclusivamente para su uso en el país establecido (Futuro360, 2020).

A nivel latinoamericano, se han desarrollado trabajos de investigaciones que presentan similitudes directas. En México, en el año 2020, se realizó un guante traductor de señas para personas sordomudas. Según el autor, la implementación de este guante tuvo un costo de \$6,687.00 dólares americanos, lo cual se consideró como un costo muy elevado. Sin embargo, en las conclusiones se determinó que, para el diseño del guante en 3 dimensiones (3D), se optaría por realizarlo bajo pedido personalizado, ya que no es posible estandarizar las medidas de las manos de las personas (Ayala Aburto, 2020).

En Colombia, en el año 2021, se desarrolló un traductor de la Lengua de Señas Colombiana utilizando visión por computadora e inteligencia artificial, el cual se materializó en forma de una aplicación web para computadoras. En el desarrollo del modelo se implementó redes neuronales, React y el lenguaje de programación JavaScript, haciendo posible traducir algunas palabras de la Lengua de Señas Colombiano en tiempo real. Sin embargo, la precisión obtenida fue del 44% (Vásquez Parada & Corredor Nieto, 2021).

En Ecuador, existen investigaciones similares desarrolladas en diferentes universidades. En Ambato, en el año 2017, se llevó a cabo la creación de un guante electrónico con la capacidad de traducir el lenguaje de señas a caracteres y posteriormente reproducirlo mediante un dispositivo móvil a través de una conexión inalámbrica. Esta investigación fue desarrollada específicamente para personas con discapacidad auditiva. No obstante, se determinó que el sistema se basa en la detección de la configuración manual para realizar la traducción. Por lo tanto, se optó por controlar el sistema de la misma manera, es decir, mediante gestos específicos. De esta forma, el usuario puede seleccionar el modo de traducción o salir de la interfaz, lo que representa una mayor facilidad en el manejo y acoplamiento a su uso. Sin embargo, se recomienda a los usuarios evitar realizar movimientos bruscos, ya que estos podrían ocasionar daños en las conexiones de los sensores o en los elementos electrónicos del guante (Brito Moncayo & Guzmán Arellano, 2017).

En 2015, en la Escuela Superior Politécnica de Chimborazo, se desarrolló un sistema de traducción del lenguaje de señas para personas sordomudas utilizando el sensor Kinect. El sistema se basa en el seguimiento y reconocimiento de los movimientos gestuales de la persona que utiliza el lenguaje de señas. El sensor Kinect captura los puntos clave del esqueleto mientras realiza las señas, lo que permite comprender la posición y movimiento de las manos y otros gestos relevantes para la comunicación. Una vez detectada la señal gestual, el sistema busca en una base de datos para determinar el significado y lo transforma en voz para que las personas oyentes puedan entender el mensaje (Guallo Criollo & Duicela Cargua, 2015).

En 2017, en la Escuela Superior Politécnica de Chimborazo, se desarrolló un dispositivo electrónico basado en sistemas embebidos con el propósito de traducir el lenguaje de señas a voz.

El dispositivo adoptó la forma de un guante electrónico. Sin embargo, los investigadores notaron que no siempre detectaba correctamente las señas o presentaba deficiencias en su funcionamiento debido a las variaciones en el tamaño de las manos entre los usuarios. Para abordar esta cuestión, se sugiere recalibrar los rangos de los sensores a través de la programación, con el fin de ajustar el dispositivo de manera más precisa a las características individuales de la mano de cada usuario (Alulema Aimara & Loza Martínez, 2017).

1.1.1. Sistematización del problema

¿Qué sistemas y dispositivos de traductor de lenguaje de señas a voz existen y cuáles son sus componentes y funcionamiento?

¿Cuáles son los requerimientos que debe cumplir el prototipo traductor de lenguaje de señas a voz?

¿Cuál es el diseño que permite cumplir con los requerimientos planteados para el desarrollo del dispositivo traductor de lenguaje de señas a voz?

¿Qué elementos de *software* y *hardware* permiten implementar el prototipo propuesto?

¿Cómo validar que el prototipo traductor de lenguaje de señas a voz cumple con los requerimientos del diseño?

1.2. Justificación

1.2.1. Justificación Teórica

En Ecuador, alrededor del 4.6% de la población, lo que equivale a unas 805,000 personas, presenta discapacidad auditiva. Las causas abarcan desde factores genéticos hasta infecciones durante el embarazo, complicaciones en el parto, enfermedades en la infancia y exposición a ruidos fuertes. Esta discapacidad puede generar dificultades en la comunicación, aislamiento social, problemas educativos y laborales, así como efectos emocionales negativos (CONADIS, 2022).

A pesar de los avances en tecnología y enfoques de inclusión, la comunicación efectiva entre las personas con discapacidad auditiva y aquellas que no lo son sigue siendo una barrera significativa. Existe una falta de herramientas que faciliten la comunicación entre estas dos poblaciones, lo que puede limitar las oportunidades educativas, laborales y sociales de las personas con discapacidad auditiva. El proyecto tiene como objetivo desarrollar un dispositivo que traduzca la LSEC a voz para mejorar la comunicación de personas con discapacidad auditiva. Esta investigación se centra en la implementación de Visión Artificial y RNA.

1.2.2. Justificación Aplicativa

El tema de investigación busca construir un prototipo electrónico traductor del lenguaje de señas a voz con base en el procesamiento de imágenes, con el fin de mejorar la comunicación entre personas con discapacidad auditiva que utilizan La Lengua de Señas Ecuatoriana y personas que no están familiarizadas con este lenguaje. Principalmente, consta de tres etapas fundamentales: etapa de adquisición de datos, etapa de procesamiento de datos y, etapa de visualización y reproducción de audio de la información.

La etapa de adquisición de datos será la encargada de recoger, mediante una cámara, toda la información referente a la LSEC. Esta información está conformada por las coordenadas de los 21 nudillos de la mano. La segunda etapa será la encargada de recibir y procesar la información adquirida en la etapa de adquisición de datos. Dicha información servirá para interpretar el alfabeto de la LSEC en el alfabeto común. Finalmente, en la etapa de visualización y reproducción de audio de la información, donde se utiliza una pantalla para visualizar la información, el prototipo necesitará estar conectado a una red de internet para reproducir el mensaje generado a través de un altavoz.

1.3. Objetivos

1.3.1. Objetivo general

Construir un prototipo electrónico traductor de lenguaje de señas a voz con base en procesamiento de imágenes.

1.3.2. Objetivos específicos

- Investigar los diferentes tipos de sistemas y dispositivos de traducción de lenguaje de señas a voz y conocer cuáles son sus componentes y su funcionamiento.
- Determinar los requerimientos que debe cumplir el prototipo para la traducción del lenguaje de señas a voz.
- Establecer el diseño que permite cumplir con los requerimientos planteados.
- Elegir los elementos de software y hardware que permitan implementar el diseño propuesto.
- Validar el cumplimiento de los requerimientos de diseño del prototipo traductor de lenguaje de señas a voz.

CAPÍTULO II

2. MARCO TEÓRICO

En este capítulo se lleva a cabo la investigación que proporcionará la información necesaria para el desarrollo del prototipo. Se explicarán definiciones relacionadas con los tipos de sistemas de traducción de lenguaje de señas a voz, discapacidad auditiva, la lengua de señas, IA, tarjetas de desarrollo, *hardware* y *software*.

2.1. Discapacidad

Es una condición temporal o permanente de las personas con deficiencias físicas, psíquicas, intelectuales o sensoriales que, al tener este tipo de condición, no tienen una participación plena y equitativa en la sociedad, en igualdad de condiciones con las demás personas. A continuación, se describe las generalidades de los tipos de discapacidad, los grados de discapacidad que puede tener una persona cuyos rangos son establecidos por la OMS, y también los diferentes métodos de comunicación alternativos que existen para las personas que tienen una discapacidad en particular (Padilla Muñoz, 2010).

Discapacidad Auditiva. - Es la pérdida total o parcial de la capacidad auditiva de la persona (Sanz Rivas & Reina Vaíllo, 2013).

En la Tabla 1-2 se presenta el método de evaluación empleado para determinar el grado de discapacidad auditiva, también se presentan los métodos de comunicación alternativos utilizados para las personas que tienen esta discapacidad.

Tabla 1-2: Aspectos generales en la evaluación de la discapacidad auditiva.

Evaluación	Grado de discapacidad	Métodos de comunicación alternativa.
Para evaluar el grado de discapacidad auditiva, se considera el Umbral Tonal Medio (UTM)	Normal: 0 - 30 dB Leve: 31 – 40 dB Moderada: 41 – 70 dB Moderada/Sev: 70–90 dB Sev/profunda: >90 dB	- Lenguaje de señas. - Mensaje escrito.

Fuente: (Sanz Rivas & Reina Vaíllo, 2013).

Realizado por: Buñay Javier, Mullo Jose, 2023

Discapacidad Visual. - La discapacidad visual se refiere a la pérdida parcial o total de la capacidad de ver debido a diversas causas. Puede estar relacionado con daños en el ojo, el nervio óptico o el sistema cerebral. El déficit puede ser congénito o adquirido (Sanz Rivas & Reina Vaíllo, 2013). En la Tabla 2-2 se muestra el método de evaluación empleado para determinar el grado de discapacidad visual, también se presentan los métodos de comunicación alternativos utilizados para las personas que tienen esta discapacidad.

Tabla 2-2: Aspectos generales en la evaluación de la discapacidad visual.

Evaluación	Grado de discapacidad	Métodos de comunicación alternativa.
Para determinar el grado de discapacidad visual se toma en cuenta la medición de agudeza visual.	<p>Leve: agudeza visual inferior a 6/12 o igual o superior a 6/18</p> <p>Moderado: agudeza visual inferior a 6/18 o igual o superior a 6/60</p> <p>Grave: agudeza visual inferior a 6/60 o igual o superior a 3/60</p> <p>Ceguera: agudeza visual inferior a 3/60</p>	<p>- Sistema Braille</p> <p>-La audiodescripción</p>

Fuente: (Sanz Rivas & Reina Vaíllo, 2013).

Realizado por: Buñay Javier, Mullo Jose, 2023

Discapacidad Física. - La discapacidad física se refiere a limitaciones físicas que pueden ser causadas por lesiones, enfermedades o condiciones congénitas. Puede variar en gravedad y afectar la movilidad y la capacidad para realizar actividades cotidianas (Sanz Rivas & Reina Vaíllo, 2013). En la Tabla 3-2 se presenta el método de evaluación empleado para determinar el grado de discapacidad física, también se presentan los métodos de comunicación alternativos utilizados para las personas que tienen esta discapacidad.

Tabla 3-2: Aspectos generales en la evaluación de la discapacidad física.

Evaluación	Grado de discapacidad	Métodos de comunicación alternativa.
Para determinar el grado de discapacidad física se toma en	<p>Normal: 0%</p> <p>Leve: 1 - 24%</p> <p>Moderada: 25 - 49%</p>	<p>- Oral</p> <p>- Escrito</p> <p>- Gestual, etc.</p>

cuenta el porcentaje de daños físicos. **Grave:** 50–75%
Muy Grave: > 75%

Fuente: (Sanz Rivas & Reina Vaíllo, 2013).

Realizado por: Buñay Javier, Mullo Jose, 2023

Discapacidad Intelectual. - La discapacidad intelectual se manifiesta cuando el coeficiente intelectual es inferior a la media, lo que resulta en dificultades en el aprendizaje, la comunicación y la interacción social (Sanz Rivas & Reina Vaíllo, 2013).

En la Tabla 4-2 se presenta el método de evaluación empleado para determinar el grado de discapacidad intelectual, también se presenta el método de comunicación alternativo utilizados para las personas que tienen esta discapacidad.

Tabla 4-2: Aspectos generales en la evaluación de la discapacidad intelectual.

Evaluación	Grado de discapacidad	Métodos de comunicación alternativa.
La evaluación de la discapacidad intelectual se realiza mediante pruebas estandarizadas del coeficiente intelectual.	Limítrofe: 70 – 79 Leve: 50 -69 Moderada: 35 -39 Grave o Severa: 20 - 34 Intelectual profunda: por debajo de 20	- Lectura fácil

Fuente: (Sanz Rivas & Reina Vaíllo, 2013).

Realizado por: Buñay Javier, Mullo Jose, 2023

Discapacidad Psicosocial. -La discapacidad psicosocial es el término empleado para describir las discapacidades que pueden originarse a partir de problemas de salud mental (Sanz Rivas & Reina Vaíllo, 2013).

2.1.1. Discapacidad Auditiva

La discapacidad auditiva se refiere a la incapacidad para escuchar los sonidos de manera adecuada o para no percibir ningún sonido en absoluto. Las deficiencias auditivas pueden ser congénitas o adquiridas más adelante en la vida. Estas deficiencias pueden variar desde leves hasta graves y pueden afectar la habilidad de una persona para comunicarse de manera efectiva, tanto en términos de comprensión auditiva como expresión verbal. Este tipo de discapacidad tiene un

impacto negativo en el reconocimiento de los sonidos, la conversación y la comprensión, especialmente a nivel social, emocional y educativo (Sanz Rivas & Reina Vaíllo, 2013).

2.1.2. Discapacidad en Ecuador

De acuerdo a los datos suministrados por el Consejo Nacional para las Personas con Discapacidad (CONADIS), presentados en la Ilustración 1-2, a nivel nacional existe un total de 471,205 personas con discapacidad. De este grupo, 264,463 son hombres (56.12%), mientras que 206,714 son mujeres (43.87%), y 28 son individuos pertenecientes al colectivo LGBT (0.01%). En lo que respecta a las diversas categorías de discapacidad, se observa que 215,156 personas presentan discapacidad física (45.66%), 108,957 personas tienen discapacidad intelectual (23.12%), 66,538 personas padecen discapacidad auditiva (14.12%), 54,397 personas enfrentan discapacidad visual (11.54%), y 26,157 personas experimentan discapacidad psicosocial (5.55%) (CONADIS, 2022).

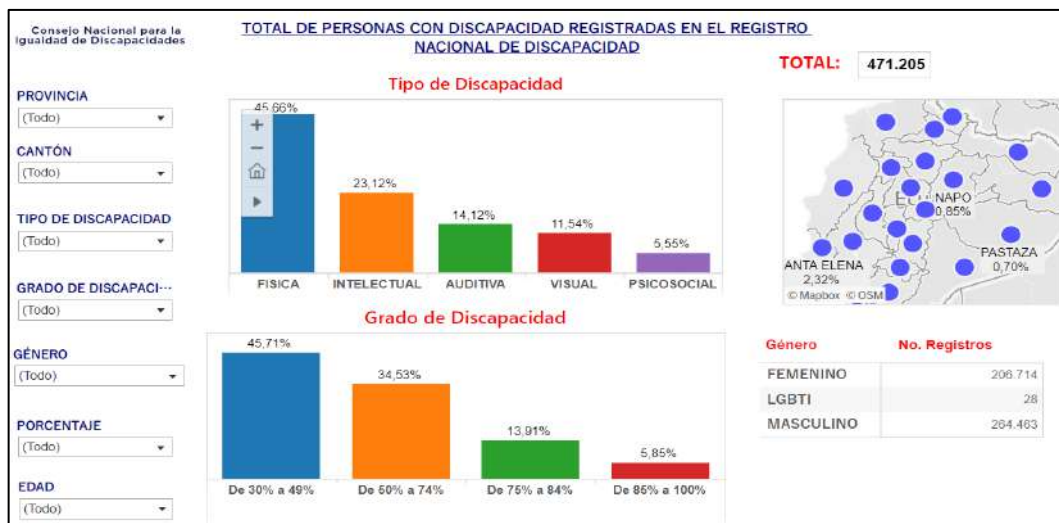


Ilustración 1-2: Registro de personas con discapacidad en Ecuador.

Fuente: (CONADIS, 2022)

De acuerdo a la Ilustración 2-2, en la provincia de Chimborazo se ha registrado un total de 14,712 personas con discapacidad. De estas, 5,470 (37.18%) presentan discapacidad física, 3,717 (25.27%) tienen discapacidad intelectual, 3,360 (22.84%) tienen discapacidad auditiva, 1,693 (11.51%) enfrentan discapacidad visual, y 472 (3.21%) experimentan discapacidad psicosocial.

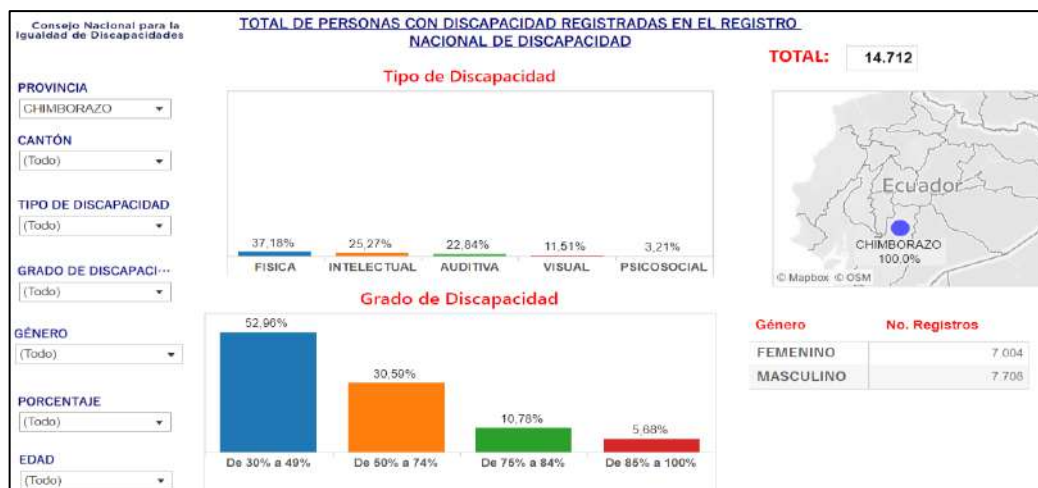


Ilustración 2-2: Registro de personas con discapacidad en la provincia de Chimborazo.
Fuente: (CONADIS, 2022)

En la Ilustración 3-2 se evidencia que en el cantón Riobamba se han registrado un total de 7,111 personas con discapacidad, incluyendo hombres, mujeres y niños. De este grupo, 2,942 (41.37%) presentan discapacidad física, 1,409 (19.81%) tienen discapacidad intelectual, 1,512 (21.26%) padecen discapacidad auditiva, 943 (13.26%) enfrentan discapacidad visual y 305 (4.29%) experimentan discapacidad psicosocial.

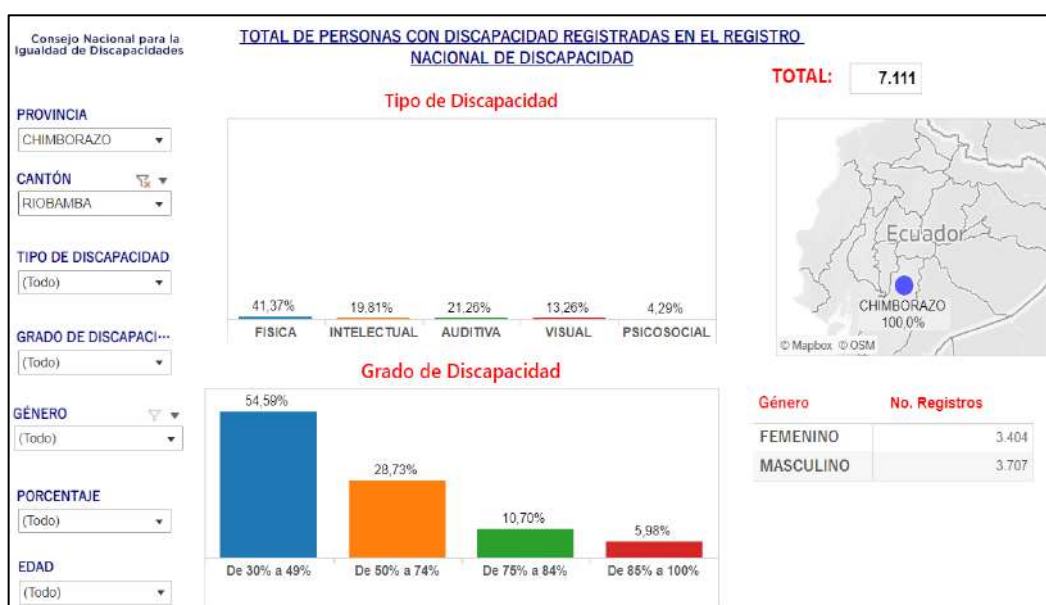


Ilustración 3-2: Registro de personas con discapacidad en el cantón Riobamba.
Fuente: (CONADIS, 2022)

En la Tabla 5-2 se puede apreciar toda la información recopilada, la cual contiene los datos más significativos para el desarrollo del Trabajo de Integración Curricular. Estos datos se concentran de manera particular en personas con discapacidad auditiva, dado que el desarrollo del prototipo está dirigido hacia este grupo.

Tabla 5-2: Datos de personas con discapacidad auditiva.

Discapacidad auditiva por:	Total, de personas discapacitadas.	Número de personas con discapacidad auditiva	Porcentaje
País: Ecuador	471,205	66,538	14,12%
Provincia: Chimborazo	14,712	3,360	22,84%
Cantón: Riobamba	7,111	1,512	21,26%

Fuente: (CONADIS, 2022)

Realizado por: Buñay Javier, Mullo Jose, 2023

2.2. Lengua de Señas

La lengua de señas es un método visual de comunicación que emplea gestos, expresiones faciales y movimientos corporales. Se utiliza principalmente entre personas sordas o con deficiencias auditivas. La lengua de señas tiene una estructura gramatical distinta de la de las lenguas orales. Su rasgo principal radica en la configuración de las manos, su orientación y posición, así como en los movimientos que se complementan con expresiones faciales y labiales. Es importante destacar que la lengua de señas no es universal (FENASEC, 2019).

2.2.1. Lengua de Señas Ecuatoriana (LSEC)

La LSEC es el sistema de comunicación gestual utilizado por la comunidad sorda y personas con discapacidad auditiva en Ecuador para comunicarse entre sí y con personas que conocen esta lengua. Al igual que otras lenguas de señas, la LSEC tiene su propio conjunto de señas y reglas gramaticales que permiten la comunicación efectiva en la comunidad sorda ecuatoriana (FENASEC, 2019).

En la Ilustración 4-2 se presenta el alfabeto de la LSEC. El Trabajo de Integración Curricular se enfoca en desarrollar un dispositivo traductor de lenguaje de señas a voz basado en este alfabeto.

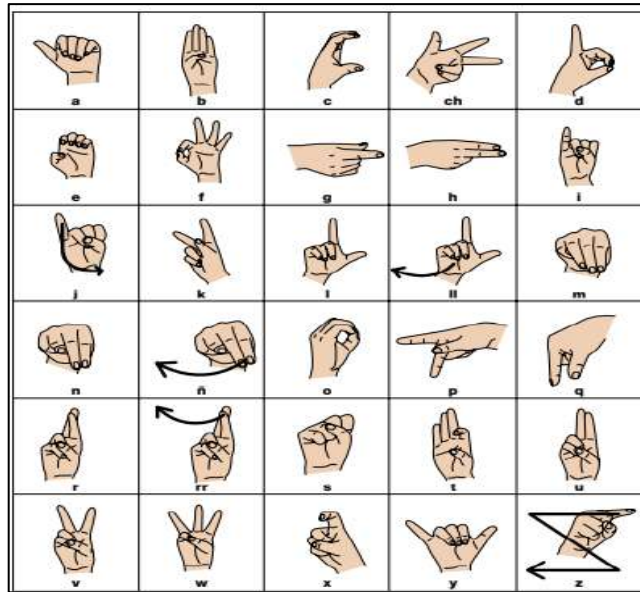


Ilustración 4-2: Abecedario de la Lengua de Señas Ecuatoriana.

Fuente: (FENASEC, 2019)

2.3. Inteligencia Artificial (IA)

La IA es la disciplina de la informática que se enfoca en el desarrollo de sistemas de hardware y software con capacidades específicas propias a las del ser humano, como la interacción con el entorno, el aprendizaje y adaptación, el razonamiento y la planificación. Estos sistemas son capaces de perseguir autónomamente objetivos predefinidos, tomando decisiones que previamente eran encomendadas a las personas (Pino Díez, et al., 2001).

Dentro del ámbito de la IA, se distinguen tres ramas principales:

- Lógica Difusa
- RNA
- Algoritmos Genéticos

Cada una presentan características particulares y cumplen funciones específicas. Para el desarrollo del prototipo, el enfoque se centra en el estudio de los RNA.

2.3.1. Aplicaciones de la Inteligencia Artificial

En la última década, el uso de la IA ha experimentado un notable crecimiento. Las empresas de consumo en Internet han recopilado volúmenes masivos de datos, los cuales se emplean para

entrenar programas de aprendizaje automático de alta potencia. Los algoritmos de aprendizaje automático están ampliamente disponibles para diversas aplicaciones, y algunos de ellos incluso son de código abierto (Pino Díez, et al., 2001).

Tabla 6-2: Aplicaciones de la inteligencia artificial.

Aplicaciones	Descripción	Ingresos globales del 2016 a 2025
Reconocimiento estático de imagen, clasificación y etiquetado	Identifica, analiza y compara las matrices de bits que componen una imagen digital, para poder clasificar o buscar a posteriori.	7.288,2 MUSD
Mejora del funcionamiento de trading algorítmico	Generación de análisis más precisos, sistematización de actividades, pronósticos más precisos, mejora las decisiones de inversión, y aproveché las oportunidades que el big data ofrece.	6.786,5 MUSD
Mejora del procesamiento de datos de pacientes	Analiza grandes cantidades de datos como imágenes médicas, pruebas de resultados clínicos de una manera más rápida y precisa para la detección temprana de enfermedades o anomalías.	6.629,7 MUSD
Mantenimiento predictivo	Analiza datos en tiempo real y el uso de los conocimientos resultantes para evitar las fallas de los equipos y el tiempo de inactividad que se produce durante el mantenimiento.	4.212,2 MUSD
Identificación, clasificación y seguimiento de objetos	Utiliza imágenes y videos para detectar, clasificar, rastrear objetos o eventos para interpretar escenarios del mundo real.	3.780,9 MUSD
Distribución de contenido en redes sociales	Estos algoritmos analizan datos de búsqueda, temas de tendencias, hashtag, patrones de publicación en las redes sociales y agrupan el tipo de contenido para el usuario objetivo.	3.209,9 MUSD

Fuente: (Pino Díez, et al., 2001)

Realizado por: Buñay Javier, Mullo Jose, 2023

La Tabla 6-2 muestra algunas de las aplicaciones más destacadas a nivel mundial, acompañadas de los ingresos económicos generados por cada aplicación. Estos ingresos se expresan en millones de dólares (MUSD).

2.3.2. *Visión Artificial*

La tecnología de visión por computadora permite a los equipos industriales “ver” sus actividades, permitiendo tomar decisiones rápidas basadas en lo que captan. Las aplicaciones más comunes de la visión artificial incluyen inspección visual, detección de defectos, posicionamiento y medición de piezas, así como la identificación, clasificación y seguimiento de productos (Domínguez Mínguez, 2021).

La Visión Artificial se encuentra entre las tecnologías pioneras de la automatización industrial. A lo largo de décadas, ha contribuido a mejorar la calidad del producto, acelerar la producción y optimizar procesos de fabricación y logística. En la actualidad, esta tecnología probada se fusiona con la IA para impulsar la transición hacia la Industria 4.0 (Domínguez Mínguez, 2021).

2.3.2.1. *Etapas de la visión artificial*

Según (Domínguez Mínguez, 2021) define las siguientes etapas:

- **Captura:** proceso de adquirir imágenes o vídeos mediante cámaras.
- **Preprocesamiento:** aplicación de técnicas como reducción de ruido y realce de detalles.
- **Segmentación:** división de la imagen en objetos de interés.
- **Extracción de características:** obtención de atributos para distinguir tipos de objetos.
- **Reconocimiento y clasificación:** identificación de objetos en una escena.
- **Interpretación:** asignación de significado a objetos reconocidos.

2.3.3. *Red Neuronal Artificial (RNA)*

Una RNA es un modelo matemático inspirado en el sistema nervioso de los seres vivos, como el cerebro humano. Consiste en nodos conectados por enlaces, organizados en capas de entrada, ocultas y de salida. La RNA aprende ajustando los pesos de las conexiones a partir de datos de entrenamiento, mejorando su rendimiento en tareas como clasificación y reconocimiento de patrones. Se utilizan en problemas complejos y no lineales, con arquitecturas avanzadas como redes neuronales convolucionales y recurrentes para aplicaciones específicas (Flórez López & Fernández Fernández, 2008).

2.3.3.1. Elementos de las Redes Neuronales Artificiales

Las RNA están conformadas por neuronas artificiales interconectadas que se organizan en tres capas: la capa de entrada, la capa oculta y la capa de salida.

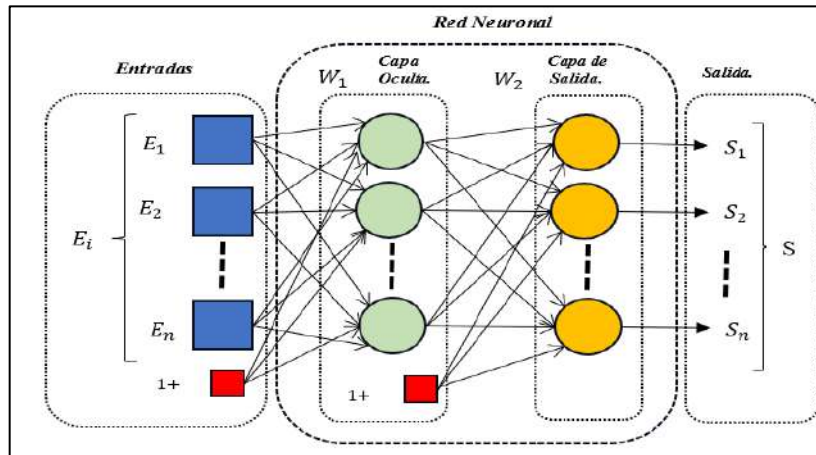


Ilustración 5-2: Esquema de la red neuronal.

Realizado por: Buñay Javier, Mullo Jose, 2023

Ecuación 1: Salida (S) de la Red Neuronal.

$$S = g(\sum W_i E_i) \quad (1)$$

Donde:

S: Salida de la neurona.

g: Función de activación.

W_i : Pesos asociados a las conexiones entre la entrada (E_i) y la neurona.

E_i : Entradas de la neurona.

La Ecuación 1 se emplea para representar la estructura de la red neuronal.

2.3.4. Aprendizaje Automático (Machine Learning, ML)

El ML es un sistema capaz de aprender a partir de ejemplos, razonamiento y experiencia. Los algoritmos permiten que las máquinas inteligentes mejoren sus capacidades y desempeño con el tiempo. De esta manera, la máquina puede aprender a realizar tareas específicas y mejorar sus respuestas y funcionalidades a través de la experiencia (Norris, 2019).

En la base del ML se encuentran diversos algoritmos que, partiendo de conceptos fundamentales, pueden tomar decisiones concretas y llevar a cabo acciones aprendidas con el tiempo (Norris, 2019). Los procesos necesarios para crear un modelo de ML se deben considerar de la siguiente manera:

2.3.4.1. Recopilar de datos

Este paso es sumamente importante, ya que la calidad y cantidad de datos recopilados determinan directamente la capacidad de un modelo predictivo para desempeñarse de manera efectiva (Bobadilla, 2021).

2.3.4.2. Preparar los datos para el entrenamiento

Dividir los datos en dos partes. La primera parte se utiliza para entrenar el modelo y constituirá la mayor proporción del conjunto de datos. La segunda parte se emplea para evaluar el rendimiento del modelo entrenado (Bobadilla, 2021).

2.3.4.3. Seleccionar un modelo

El tipo de algoritmo a utilizar está condicionado por la naturaleza de los datos disponibles, dado que ciertos métodos son adecuados para imágenes, mientras que otros son más apropiados para secuencias (como texto o música); además, hay algunos diseñados para datos numéricos y otros para información basada en texto (Bobadilla, 2021).

2.3.4.4. Entrenar el modelo

Este proceso consiste en ingresar variables al algoritmo, comparar la salida con los resultados esperados, ajustar valores y repetirlo hasta que el algoritmo devuelva el resultado correcto en la mayoría de las ocasiones (Bobadilla, 2021).

2.3.4.5. Evaluación

La evaluación implica poner a prueba el modelo con datos que no fueron utilizados durante el entrenamiento. Esta métrica proporciona una idea de cómo el modelo podría desempeñarse ante información que aún no ha sido vista (Bobadilla, 2021).

2.3.5. El Aprendizaje Profundo (Deep Learning, DL)

El DL es un enfoque de aprendizaje automático y jerárquico que pertenece a la rama del ML y la IA. Este método imita la manera en que los seres humanos adquieren ciertos tipos de

conocimiento. Consiste en la utilización de RNA dispuestos en capas sucesivas para aprender de los datos de forma iterativa (Bobadilla, 2021).

2.4. Investigación de traductores de lenguaje de señas a nivel nacional

En Ecuador, a través de una revisión bibliográfica, se identifican varias investigaciones en diferentes universidades locales sobre sistemas y dispositivos traductores de la LSEC. Estos utilizan varios componentes y recursos de *hardware* y *software*. En esta sección, se describirán varios dispositivos y sistemas, explicando su funcionamiento y los componentes que cada uno de ellos emplea.

Traductor de Lenguaje de Señas utilizando el sensor Kinect. - En 2015, en la Escuela Superior Politécnica de Chimborazo, se desarrolló un sistema de traducción de lenguaje de señas para personas sordomudas utilizando el sensor Kinect. Este sistema utiliza la Kinect para registrar los gestos del lenguaje de señas a través de sus cámaras y sensores de profundidad. El dispositivo es capaz de identificar y monitorear los movimientos y la posición de diversas partes del cuerpo humano, dividiendo el esqueleto en 20 puntos cuando la persona está de pie y en 10 puntos cuando está sentada (Guallo Criollo & Duicela Cargua, 2015).

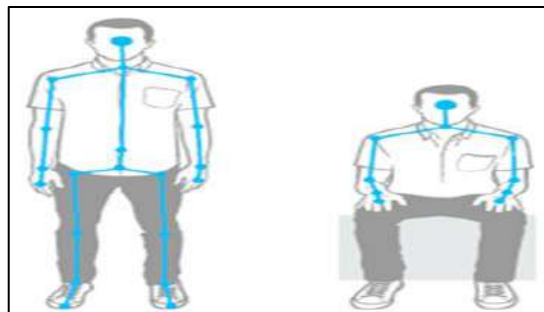


Ilustración 6-2: Reconocimiento esquelético.

Fuente: (Guallo Criollo & Duicela Cargua, 2015)

Como se muestra en la Ilustración 6-2, estos puntos abarcan desde la cabeza hasta los brazos, pasando por las manos, las piernas y los pies. A través de esta habilidad de seguimiento, la Kinect logra comprender la postura y los gestos de un individuo, lo cual resulta fundamental para la interpretación y traducción efectiva del lenguaje de señas. Mediante algoritmos, estos gestos se convierten en formato de texto y posteriormente se transforman en voz sintetizada. Esto facilita la comunicación entre personas con discapacidad auditiva y aquellas que escuchan, utilizando la tecnología de la Kinect como puente de interacción.

- **Desventajas del uso del sensor Kinect en el traductor:** Los factores que tienen un impacto negativo son ciertas condiciones de iluminación y el uso de ropa llamativa, lo que dificulta que el sensor Kinect pueda identificar o seguir los movimientos del usuario. Por lo tanto, se recomienda evitar la exposición directa a la luz solar y optar por vestimenta de tonos oscuros.

Dispositivo electrónico mediante Sistemas Embebidos para la traducción del lenguaje de señas a palabras. - En 2017, en la Escuela Superior Politécnica de Chimborazo, Se diseñó e implementó un dispositivo electrónico que traduce el lenguaje de señas en palabras para permitir una comunicación efectiva de personas con discapacidad auditiva. Se utilizaron Sensores Flex Resistivos para captar la forma en que se doblaban los dedos al realizar señas, y un microcontrolador ATMega para procesar estas señales y convertirlas en palabras o sonidos. Se empleó una Raspberry Pi para gestionar la comunicación y la visualización. El dispositivo puede traducir las 28 letras del alfabeto y combinarlas para formar palabras (Alulema Aimara & Loza Martínez, 2017).



Ilustración 7-2: Guante traductor de lenguaje de señas.

Fuente: (Alulema Aimara & Loza Martínez, 2017)

- **Desventaja del sistema embebido traductor de lenguaje de señas:** La versatilidad de este dispositivo se ve limitada debido a las variaciones en el tamaño de las manos entre las personas. Por lo tanto, se requiere recalibrar los rangos de los sensores cuando otra persona con un tamaño de mano diferente lo utilice.

Traductor de lenguaje de señas mediante visión artificial. - En el año 2012, en la Escuela Politécnica Nacional, se desarrolló un sistema que emplea la visión artificial para traducir el lenguaje de señas en texto. Esto facilita la comunicación entre personas con discapacidad auditiva o visual que utilizan lenguaje de señas y aquellos que no lo comprenden. El sistema incluye una función de entrenamiento para aquellos que no están familiarizados con el lenguaje de señas. Su funcionamiento consiste en capturar una imagen y procesarla digitalmente mediante LabVIEW

2009 junto con su toolkit de visión artificial. Se aplican filtros y operaciones morfológicas para mejorar la calidad de la imagen y eliminar elementos no deseados. Para identificar las letras en las imágenes, se utiliza el Vision Assistant de LabVIEW, tal como se muestra en la Ilustración 8-2. El texto resultante puede ser mostrado en pantalla, exportado a un documento de Word y también puede ser convertido en audio utilizando la función de "texto a voz" de Windows (Chiguano Rodríguez & Moreno Díaz, 2011).



Ilustración 8-2: Sistema traductor de lenguaje de señas con Visión Artificial.

Fuente: (Chiguano Rodríguez & Moreno Díaz, 2011)

- **Desventaja del sistema traductor de lenguaje de señas mediante Visión Artificial.** - Este sistema no es capaz de reconocer letras de la Lengua de Señas que involucran movimiento. Por ello, se sugiere la utilización de un módulo RIO de National Instruments que pueda realizar el procesamiento en tiempo real de imágenes y trabajar con cámaras de mayor velocidad, como las cámaras Smart. Esto habilitaría la aplicación de un algoritmo para detectar trayectorias en tiempo real.

2.4.1. *Arquitectura de los dispositivos traductores de la LSEC*

Basándose en la investigación de dispositivos y sistemas traductores de la LSEC a nivel nacional, se identifican los componentes de interés necesarios para el desarrollo del prototipo. La Ilustración 9-2 muestra la arquitectura de los dispositivos investigados, con el propósito de analizar y determinar los componentes de *hardware* y *software* esenciales.

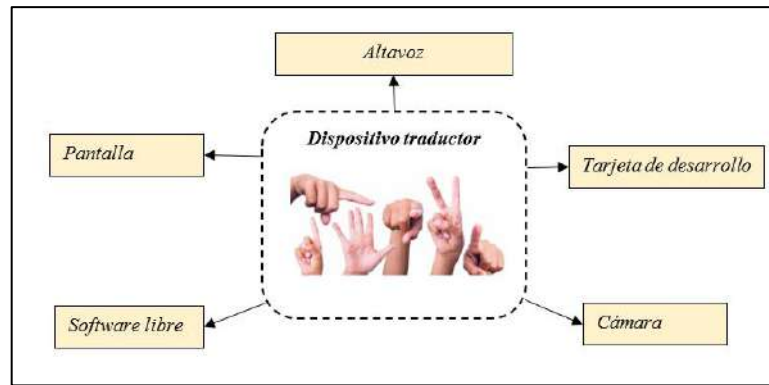


Ilustración 9-2: Arquitectura de los dispositivos estudiados.

Realizado por: Buñay Javier, Mullo Jose, 2023.

2.5. Tarjetas de desarrollo

Las tarjetas de desarrollo son placas electrónicas diseñadas para ayudar en la creación y prototipado de proyectos electrónicos. Estas tarjetas suelen incluir microcontroladores o microprocesadores, puertos de entrada y salida, y a menudo se programan para realizar diversas tareas. Son ampliamente utilizadas por ingenieros, desarrolladores y entusiastas de la electrónica para crear y probar circuitos y sistemas electrónicos (Scotts, 2013).

Raspberry Pi: Según (Scotts, 2013), la Raspberry Pi es una nano computadora capaz de procesar información y ejecutar lenguajes de programación de alto nivel. Tiene el tamaño de una tarjeta de crédito y está equipado con elementos esenciales, como un microprocesador ARM, memoria RAM, tarjeta de vídeo, tarjeta ethernet, conectividad wifi y bluetooth.

En la Tabla 7-2 se detallan las características y tipos de la Raspberry Pi diseñados para funcionar en diferentes sistemas operativos, como Raspbian, Arch Linux, Pidora, Ubuntu, Windows 10 y aplicaciones de IoT.

Tabla 7-2: Características de los diferentes modelos de la Raspberry Pi.

Modelo	Tamaño(mm)	Alimentación	CPU	RAM	USB	Red
Raspberry Pi Zero	65X30	160 mA 5V/ 0.8W	1GHz ARM1176JZF-S	512 MB	1micro	No
Raspberry Pi Zero W	65x30	160 mA 5vV 0.8W	1GHz ARM1176JZF-S	512 MB	1micro	Wifi, BT
Raspberry Pi 2 B	85X56	800 mA 5V/4W	900 MHz ARM Cortex-A7	1 GB	4	ETH

Raspberry Pi 3 B+	85x56	2.5 A 5V/12.5W	1.4 GHz ARM Cortex-A53	1GB	4	ETH Wifi
Raspberry Pi 4B	85x56	2.5 A 5V/12.5W	1.5 GHz ARM Cortex-A72	1,2 y 4GB	2 (2.0) 2(3.0)	ETH Wifi

Fuente: (Scotts, 2013)

Realizado por: Buñay Javier, Mullo Jose, 2023.

Jetson Nano: Es una placa de desarrollo diseñada por NVIDIA para aplicaciones de IA y computación de alto rendimiento en sistemas embebidos. Esta placa ofrece un conjunto de características que la hacen adecuada para tareas como el procesamiento de imágenes, la inferencia de RNA y la robótica, entre otros. Está equipado con los componentes esenciales para su funcionamiento, tales como GPU, CPU, memoria, codificación de video, decodificación de video, conectividad, salidas de video, GPIO y UPHY (Arroyo Ruiz, 2022).

La Tabla 8-2 proporciona detalles sobre los modelos de la línea Jetson desarrollados por NVIDIA. Esta compañía ofrece una serie de productos con capacidades sobresalientes, entre los que se destacan la Jetson AGX XAVIER y la Jetson TX2.

Tabla 8-2: Características de los diferentes modelos de la Jetson Nano.

Modelo	Tamaño(mm)	Alimentación	CPU	RAM	USB	Red
Jetson AGX Xavier 64 GB	100x87mm	10 W 15 W 30 W	NVIDIA Carmel Armv8.2 de 64 bits de 8 núcleos 6 MB L2 + 4 MB L3	LPDDR4x de 64, 32 GB y 256 bits 136,5 GB/s	4	ETH, Wifi, BT
Jetson TX2 NX	69,6x45 mm	7.5W 15W	64 bits Denver 2	LPDDR4 de 4 GB y 51,2GB/s	4	ETH, Wifi, BT
Jetson TX2 4GB	87x 50 mm	7.5W	Quad-core Arm®	4 GB	4	ETH, Wifi, BT
Jetson TX2	50 x 87 mm	7.5W	Cortex®-A57 MPCore processor	LPDDR4 de 8 GB y 59,7GB/s	4	ETH, Wifi, BT

TX2i	-	10W 20W	-	LPDDR4 de 8 GB y (ECC Support) 4 GB y (ECC Support) 51,2GB/s	ETH, Wifi, BT
-------------	---	------------	---	--	---------------------

Fuente: (Arroyo Ruiz, 2022)

Realizado por: Buñay Javier, Mullo Jose, 2023.

Google Coral Dev Board: La Google Coral Dev Board es una placa de desarrollo diseñada por Google que se utiliza para crear aplicaciones de IA y ML en sistemas embebidos. Esta placa está especialmente diseñada para aplicaciones de visión por computadora, reconocimiento de voz y otras tareas de IA. Google Coral se enfoca en TensorFlow Lite. Posee la capacidad de ejecutar 4 billones de operaciones por segundo con un consumo de energía de tan solo 2 vatios (Kurniawan, 2019).

En la Tabla 9-2 se presentan las características más importantes de la tarjeta de desarrollo Google Coral Dev Board.

Tabla 9-2: Características de la Google Coral Dev Board.

UPC	Soc. NXP i.MX 8M (cortex-A53 cuádruple, Cortex-M4F)			
GPU	Gráficos GC7000 Lite integrados			
Acelerador de aprendizaje automático	Coprocessor	Google	Edge	TPU:
	4 TOPS (int8); 2 TOPS por vatio			
RAM	1 o 4 GB LPDDR4			
Memoria flash	MMC de 8 GB, ranura MicroSD			
Inalámbrico	Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5GHz) y Bluetooth 4.2			
USB	tipo C OTG; potencia tipo C; Anfitrión tipo A 3.0; Consola serie Micro-B			
LAN	Puerto Ethernet Gigabit			
Audio	Conector de audio de 3,5 mm (compatible con CTIA); Micrófono digital PDM (x2); Terminal de 4 pines de 2,54 mm para altavoces estéreo			
Video	HDMI 2.0a (tamaño completo); Conector FFC de 39 pines para pantalla MIPI-DSI (4 carriles); Conector FFC de 24 pines para cámara MIPI-CSI2 (4 carriles)			
Fuerza	5 V CC (USB tipo C)			

Fuente: (Kurniawan, 2019)

Realizado por: Buñay Javier, Mullo Jose, 2023.

2.5.1. Análisis comparativo de las tarjetas de desarrollo

La Tabla 10-2 presenta una comparación entre las tarjetas de desarrollo, enfocándose especialmente en aquellas que ofrecen las características de funcionalidad más idóneas para la implementación del prototipo.

Tabla 10-2: Comparativa de las diferentes tarjetas de desarrollo.

Parámetros	Raspberry Pi 4 GB	Jetson Nano 4GB	Google Coral Dev Board 4GB.
Dimensiones	85.6x56.5 mm	50x87 mm	64 x 48 mm
Consumo de Energía	5V – 3.6V	5V – 4A	3.3V- 82mA
Voltaje de entrada	5V	5V	5V
Procesador	ARM Cortex	Quad-core Arm® ARM Cortex	Gráficos GC7000 Lite integrados
RAM	4 GB	4G	4G
Almacenamiento	MicroSD	MicroSD	Flash eMMC
GPU	-		
Proyectos (IA)	-		
Precio (dólares)	\$250	\$300	\$350

Realizado por: Buñay Javier, Mullo Jose, 2023.

Después de analizar las diferentes características que se muestran en la Tabla 10-2, se optó por seleccionar la tarjeta de desarrollo Jetson Nano 4GB B01. Esta elección se basa en su capacidad para cumplir con los requisitos para el desarrollo del prototipo, ya que está especialmente diseñada para aplicaciones relacionadas con la IA, el procesamiento de imágenes y la inferencia de RNA. Además, esta tarjeta tiene un costo más económico en comparación con otras alternativas de desarrollo destinadas a esta misma área.

2.6. Hardware

El *hardware* se compone de todas las partes físicas que conforman una computadora o dispositivo electrónico, es decir, aquellas que se pueden tocar principalmente (Blyler, 2013). A continuación, se presenta una descripción del *hardware* para la adquisición de datos y el *hardware* de emisión de datos que conforma el prototipo desarrollado.

2.6.1. Hardware de adquisición de datos

Son dispositivos que se conectan a una unidad central de procesamiento para su buen funcionamiento. Estos periféricos se utilizan para establecer comunicación con el entorno o para añadir nuevas funciones y capacidades a nuestros dispositivos (Blyler, 2013).

Cámara: es un dispositivo electrónico que permite captar, registrar y reproducir imágenes según los servicios que necesite el usuario y la tecnología que emplee cada dispositivo.

En el mercado existen diferentes tipos de cámaras establecidos para distintos fines. La única diferencia está en el tamaño y la calidad de procesamiento de imágenes (Blyler, 2013).

2.6.1.1. Análisis comparativo de cámaras de video

En la Tabla 11-2 se presentan los tipos de cámaras de video con sus respectivas características. Se ha considerado la compatibilidad de las cámaras con la tarjeta de desarrollo previamente seleccionada.

Tabla 11-2: Características de los diferentes tipos de cámaras.

Parámetros	Cámara IMX219	ELP cámara USB	Cámara de seguridad Dahua.
Resolución	2160P / UHK / 4K	1080p (Full HD)	720p(HD)
Resolución en Mpx	8 Mpx	2 Mpx	720 Px
Resolución (H)x(V)	3840 x 2160	1920 x 1080	1280 x 720
Infrarrojo	-	100 fps	-
Tipo de conexión	Conexión tipo E	USB	Coaxial RJ45
Precio (dólares)	\$ 40	\$ 50	\$ 15

Realizado por: Buñay Javier, Mullo Jose, 2023.

Después de analizar las diferentes características, se optó por seleccionar la cámara de video ELP USB debido a que se ajusta al desarrollo del prototipo. Ofrece una buena resolución y funciona de manera eficaz incluso en condiciones de baja luminosidad. Además, su costo de adquisición resulta accesible y sus puertos de conexión son compatibles con la tarjeta de desarrollo.

2.6.2. Hardware de emisión de datos

Son dispositivos que permiten difundir la información hacia los receptores, ya sea de forma visual o sonora, una vez que ha sido procesada en el centro de procesamiento de información (Blyler, 2013).

Pantalla. - Es un dispositivo visual que muestra información, imágenes o contenido multimedia. Generalmente, consta de una superficie plana que puede ser de diversos tamaños y tecnologías, diseñada para proyectar imágenes visibles para el observador (López Aldea, 2017).

2.6.2.1. Tipos de pantalla

La Tabla 12-2 muestra varios tipos de pantallas junto con sus correspondientes características.

Tabla 12-2: Comparativa de los diferentes modelos de pantallas

Parámetros	XYGSTUDY	ELECROW	HOSYOND
Pulgadas	7 plg	5 plg	5 plg
Resolución	1024 x 600 LCD	800 x 480 IPS	800 x 480 IPC LCD
Tecnología de conectividad.	USB – HDMI	USB – HDMI	USB - HDMI
Altavoz			
Compatibilidad	Raspberry Pi, Raspbian, 5 puntos de contacto, sin conductor, retropié, controlador libre y Jetson Nano.	Raspberry Pi, soporta Raspbian, Ubuntu, Kali, Jetson Nano, un solo toque y controlador libre.	Raspberry Pi, BB Black, Banana Pi, Jetson Nano y otros mini PC principales
Soporte de sistemas operativos.	Ubuntu, Windows	Ubuntu, Windows	Ubuntu, Windows
Precio (dólares)	\$ 77	\$ 62	\$ 43

Realizado por: Buñay Javier, Mullo Jose, 2023.

Después de realizar el análisis de las diferentes pantallas, se ha seleccionado la pantalla HDMI ELECROW Raspberry Pi de 5 pulgadas, que cuenta con pantalla táctil, tecnología IPS y una resolución de 800 x 480 píxeles. Este monitor se alimenta mediante USB y es compatible con la tarjeta Jetson Nano, además de cumplir con los requerimientos del prototipo.

2.7. Software

El *software* es un conjunto de programas, instrucciones y datos que permiten que una computadora o un dispositivo electrónico ejecuten tareas específicas. Es la parte lógica o intangible de un sistema informático, que contrasta con el *hardware*, que es la parte física o tangible (Blyler, 2013).

A continuación, se presenta una descripción detallada del lenguaje de programación Python y las bibliotecas que simplifican la aplicación de técnicas de Visión Artificial e IA.

2.7.1. Open Source

Es un tipo de *software* que se distribuye al público sin costo alguno. Depende de los derechos del desarrollador, quien pone a disposición su código fuente para que pueda ser reutilizado o no en el *software* que se esté desarrollando. Los códigos libres restrictivos son programas que carecen de derechos de autor, lo que significa que están en el dominio público. Muchos programas populares comenzaron como *software* libre antes de evolucionar y convertirse en *software* de código abierto (Moreno, 2006).

2.7.2. Python

Python es un lenguaje de programación de alto nivel que se ha convertido en una herramienta poderosa y versátil en diversas áreas de la informática. Esto se debe a su facilidad de uso, sintaxis clara y a una amplia comunidad de desarrolladores. Sus características clave incluyen legibilidad, versatilidad, interpretación, amplia biblioteca y orientación a objetos. Python se emplea en proyectos tanto pequeños como grandes, abarcando desde scripts simples hasta aplicaciones empresariales y proyectos científicos complejos (Buttu, 2016).

2.7.3. OpenCV (Open-Source Computer Vision Library)

Es una biblioteca de código abierto que se utiliza para procesar imágenes y videos en aplicaciones de visión por computadora. Fue creada con varios propósitos, incluyendo el ML, la visión artificial, algoritmos, operaciones matemáticas, captura de video, procesamiento de imágenes, entre otros. Está disponible en varios lenguajes de programación y se utiliza en una variedad de campos, incluyendo robótica, medicina y seguridad (Gevorgyan, et al., 2020).

2.7.4. TensorFlow

TensorFlow es una biblioteca de código abierto usada principalmente en proyectos de ML. Sirve para crear y entrenar RNA que reconocen patrones, como lo haría la mente humana. Simplifica cálculos complejos al definir la arquitectura del modelo, el número de capas, funciones de activación y optimizadores. Permite entrenar el modelo con datos y destaca por su escalabilidad en diversos *hardwares*, además de ofrecer una interfaz visual para depurar y rastrear el progreso del entrenamiento, ayudando a solucionar problemas en los modelos (Buttu, 2016).

2.7.5. MediaPipe

Es una herramienta de código abierto creada por Google que simplifica el desarrollo de aplicaciones en tiempo real de visión por computadora. Ofrece componentes ya entrenados para tareas como seguimiento de manos y reconocimiento facial. Además, su interfaz es amigable y se utiliza en diversas aplicaciones, como realidad aumentada y análisis de postura (Dolinina, et al., 2023).

2.7.5.1. Mediapipe Hands

Es una biblioteca de código abierto desarrollada por Google para detectar y rastrear las manos en tiempo real mediante cámaras y videos. Utiliza visión por computadora y ML para reconocer la posición y movimiento de las manos en aplicaciones como control de gestos y realidad aumentada (Dolinina, et al., 2023).

Esta tecnología es útil en diversas aplicaciones, como la interacción con dispositivos mediante gestos de las manos, la realidad aumentada, la realidad virtual, la edición de video y muchas otras áreas que requieren una comprensión precisa de la posición y los movimientos de las manos en tiempo real.

En la Ilustración 10-2 se pueden apreciar los 21 puntos clave que utiliza Mediapipe Hands para realizar un seguimiento preciso de las manos en imágenes o videos. Estos puntos desempeñan un papel fundamental al permitir a Mediapipe Hands detectar gestos y movimientos en tiempo real, lo que posibilita una amplia variedad de aplicaciones, incluyendo el control de gestos y la interacción con dispositivos

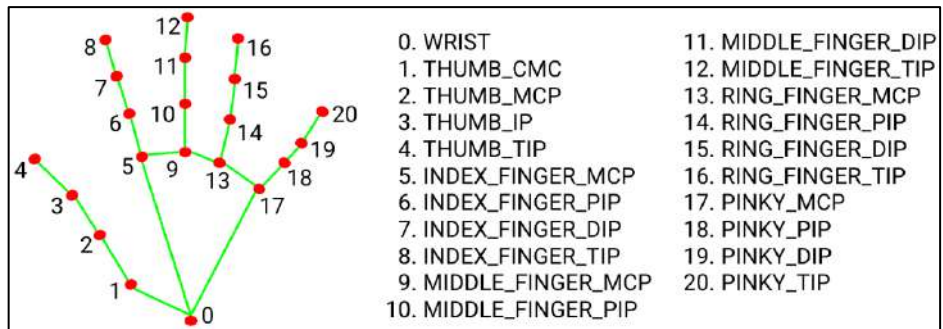


Ilustración 10-2: Puntos de referencia de la mano que utiliza Medipipe Hands.

Fuente: (Martínez Delgado, 2011)

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo se describen los requerimientos que debe cumplir el prototipo electrónico, detallando el concepto arquitectónico, los componentes de *hardware* y *software*, los algoritmos y las conexiones para la implementación.

3.1. Requerimientos del prototipo del traductor de lenguaje de señas a voz

Con base en la investigación realizada en el capítulo anterior, se planteó los requerimientos necesarios para la construcción del prototipo electrónico traductor de lenguaje de señas a voz con base en procesamiento de imágenes:

- La cámara debe capturar imágenes en un entorno con niveles de iluminación entre 100 y 300 lux, a una distancia que oscile entre 20 cm y 60 cm.
- Utilizar un sistema operativo y bibliotecas de software gratuitas.
- Proporcionar al prototipo suficiente capacidad de procesamiento para evitar tiempos de latencia.
- Localizar las 21 coordenadas de los nudillos de la mano dentro de la región detectada y almacenarlas en un archivo CSV.
- Crear y entrenar una red neuronal para que interprete la información de las 21 coordenadas detectadas en la mano y las relacione con el alfabeto de la Lengua de Señas Ecuatoriana.
- El entrenamiento de la RNA debe llevarse a cabo de manera offline.
- Desarrollar un algoritmo que posibilite la concatenación de letras para formar palabras a partir de los datos ingresados.
- Para formar frases, el prototipo contará con opciones de 'space' y 'backspace', las cuales serán configuradas mediante gestos de la mano
- El prototipo contará con una pantalla para mostrar la información generada, así como un altavoz para reproducir esa información.

3.2. Concepción del diseño general del prototipo traductor de lenguaje de señas a voz

En la Ilustración 1-3 se muestran los elementos que conforman el prototipo. Estos elementos incluyen una cámara USB que se encargará de capturar imágenes y transmitir las a través de la conexión USB a la tarjeta de desarrollo. La tarjeta será responsable de procesar la información

recibida. Posteriormente, la información procesada se mostrará en una pantalla y se reproducirá a través de un altavoz.

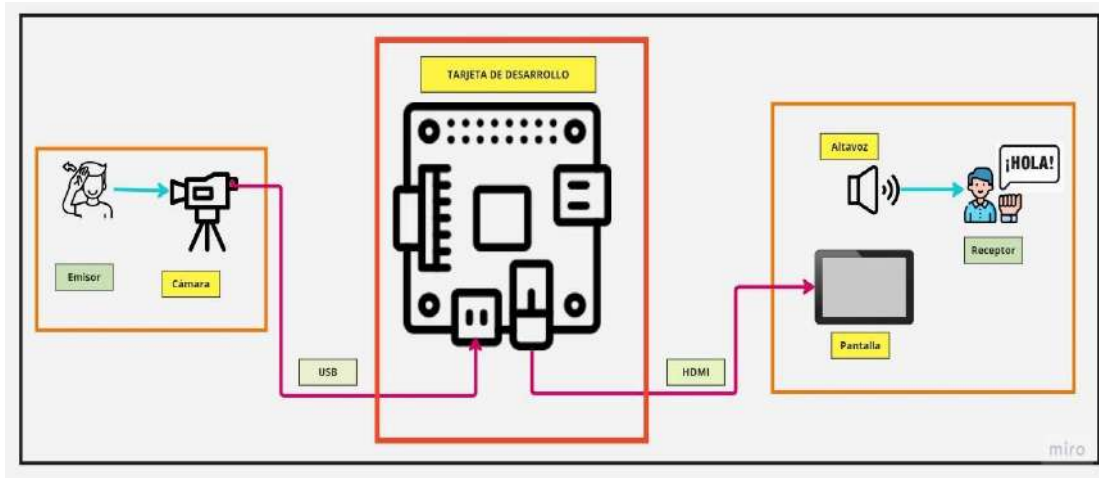


Ilustración 1-3: Concepción general del prototipo traductor de lenguaje de señas.

Realizado por: Buñay Javier & Mullo Jose, 2023.

3.3. Diseño de bloques del prototipo traductor de lenguaje de señas a voz

Tras completar la etapa de concepción general del prototipo, que se muestra en la Ilustración 2-3, se puede observar que consta de tres bloques: adquisición de datos, procesamiento de datos y visualización junto con la reproducción de audio de la información. A continuación, se ofrece una breve descripción de cada bloque.

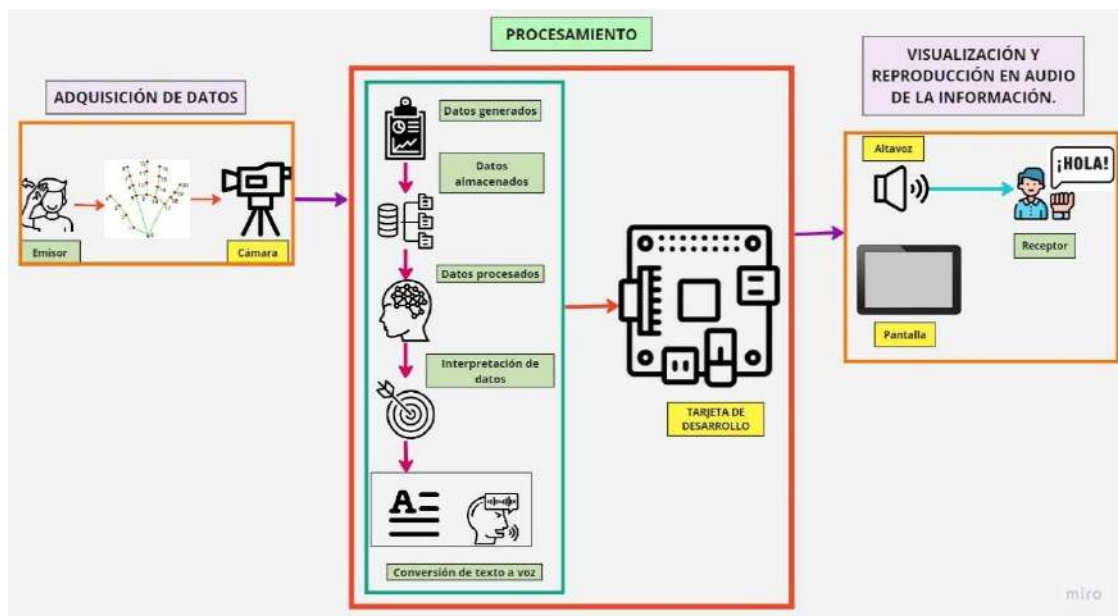


Ilustración 2-3: Concepción del prototipo por bloques.

Realizado por: Buñay Javier, Mullo Jose, 2023.

3.3.1. Bloque de adquisición de datos

Está compuesta por una cámara USB infrarroja que captura imágenes a una resolución de 1280 x 720 píxeles, estas imágenes serán transferidas mediante conexión USB a la tarjeta de desarrollo para su procesamiento.

3.3.2. Bloque de procesamiento de datos

En la tarjeta de desarrollo se ejecutará un programa desarrollado en Python que extraerá datos de las imágenes. Estos datos consistirán en coordenadas, tanto en el eje 'x' como en el eje 'y', correspondientes a los 21 nudillos de la mano. Estas coordenadas serán utilizadas para interpretar la información recibida del alfabeto de Lengua de Señas Ecuatoriana y convertirla al alfabeto común.

3.3.3. Bloque de visualización y reproducción de audio de la información

En esta etapa, se dispone de una pantalla de 5 pulgadas que muestra las coordenadas de los 21 nudillos de la mano, además de la letra correspondiente a la seña realizada por el usuario. También, cuenta con un altavoz que reproduce la información con el objetivo de facilitar la comunicación.

3.4. Selección de hardware

En la siguiente sección, se proporcionan detalles de los componentes utilizados en la construcción del prototipo traductor de lenguaje de señas a voz, incluyendo una descripción y las principales características técnicas.

3.4.1. Cámara ELP USB

En la Ilustración 3-3 se presenta una cámara USB 2.0 con una resolución de 2MP y visión nocturna. Los controladores UVC nativos de Windows, Linux y Mac son compatibles con esta cámara, lo que significa que no se necesitan controladores adicionales.



Ilustración 3-3: ELP Cámara USB.

Fuente: <https://www.amazon.com/-/es>

En la Tabla 1-3 se detallan las especificaciones técnicas de la cámara ELP USB.

Tabla 1-3: Especificaciones técnicas de la cámara ELP USB.

Especificación Técnica	Valor
Voltaje	5V
Corriente	150mA
Resolución Máxima	2.0 megapíxeles:1920 (H) x 1080 (V) píxeles
Temperatura de trabajo	-10 ~ 70 °C
Sensor	1/2.7" CMOS OV2710
Tamaño de lente	lente gran angular de 0.142 in.
Compatibilidad	Compatible con el controlador libre

Realizado por: Buñay Javier & Mullo Jose, 2023.

3.4.2. Tarjeta de desarrollo Jetson Nano 4GB B01

La tarjeta de desarrollo Jetson Nano 4GB que se muestra en la Ilustración 4-3, es compatible con el pack SDK NVIDIA JetPack, que está basado en un sistema operativo Linux y contiene todos los controladores, bibliotecas, API's de IA y visión por computador.



Ilustración 4-3: Jetson Nano 4GB.

Fuente: <https://www.amazon.com>

En la Tabla 2-3 se describen las especificaciones técnicas de la Jetson Nano 4GB B01.

Tabla 2-3: Especificaciones técnicas de la Jetson Nano 4GB B01.

Especificación técnica	Valor
Voltaje de Operación	5v
Consumo de corriente	1 A
GPU	GPU con arquitectura NVIDIA Maxwell de 128 núcleos
CPU	Procesador ARM Cortex-A57 MPCore de cuatro núcleos
Pantalla	1 HDMI 2.0, 1x DP 1.2
Redes	Gigabit Ethernet, M.2 Key E, Wi-Fi, Bluetooth
Almacenamiento	4 GB.
USB	4 conectores x USB 3.0 Type-A 1 conector USB Micro-B: USB 2.0
Otras E/S	Cabezales de 40 clavijas (UART, SPI, I2S, I2C, PWM, GPIO) Cabezal de automatización de 12 clavijas Cabezal del ventilador de 4 clavijas Cabezal POE de 4 clavijas Toma de corriente continua Botones de energía, recuperación forzada y reinicio
Dimensiones	100x79x30.21mm
Peso	340g

Realizado por: Buñay Javier & Mullo Jose, 2023.

3.4.3. Pantalla ELECROW IPS 800 X 480 USB

En la Ilustración 5-3 se muestra una Pantalla LCD de 5 pulgadas que cuenta con una resolución de 800 x 480. Tiene una interfaz HDMI para visualización, es plug and play, dispone de una interfaz micro USB para la función táctil y es compatible con la Jetson Nano.



Ilustración 5-3: Pantalla ELECROW de 5 pulgadas.

Fuente: <https://www.amazon.com/-/es>

En la Tabla 3-3 se detallan las especificaciones técnicas de la pantalla ELECROW.

Tabla 3-3: Especificaciones técnicas de la pantalla ELECROW.

Especificación técnica	Valor
Voltaje de operación	12v
Consumo de corriente	150mA
Resolución Máxima	800 X 480
Tamaño	Diagonal de 5 pulg.
Conectividad	Usb, HDMI
Temperatura de Funcionamiento	20 a 50 °C
Dimensiones	100x7
Peso	480g

Realizado por: Buñay Javier & Mullo Jose, 2023.

3.5. Esquema de conexión electrónica

En la Ilustración 6-3 se presenta el esquema de conexión de los elementos que conforman el prototipo de traductor de lenguaje de señas.

- La tarjeta de desarrollo Jetson Nano es el núcleo del suministro de energía para los demás componentes electrónicos. Será alimentada con 5V/4A y cuenta con un regulador de voltaje que convierte 120V a 5V, conectado a la interfaz de alimentación de la tarjeta.
- La pantalla está conectada al puerto HDMI de la tarjeta de desarrollo. Además, la pantalla cuenta con altavoces incorporados y está conectada al puerto USB 2 de la tarjeta.
- La cámara está conectada al puerto USB 3.0 de la tarjeta de desarrollo.

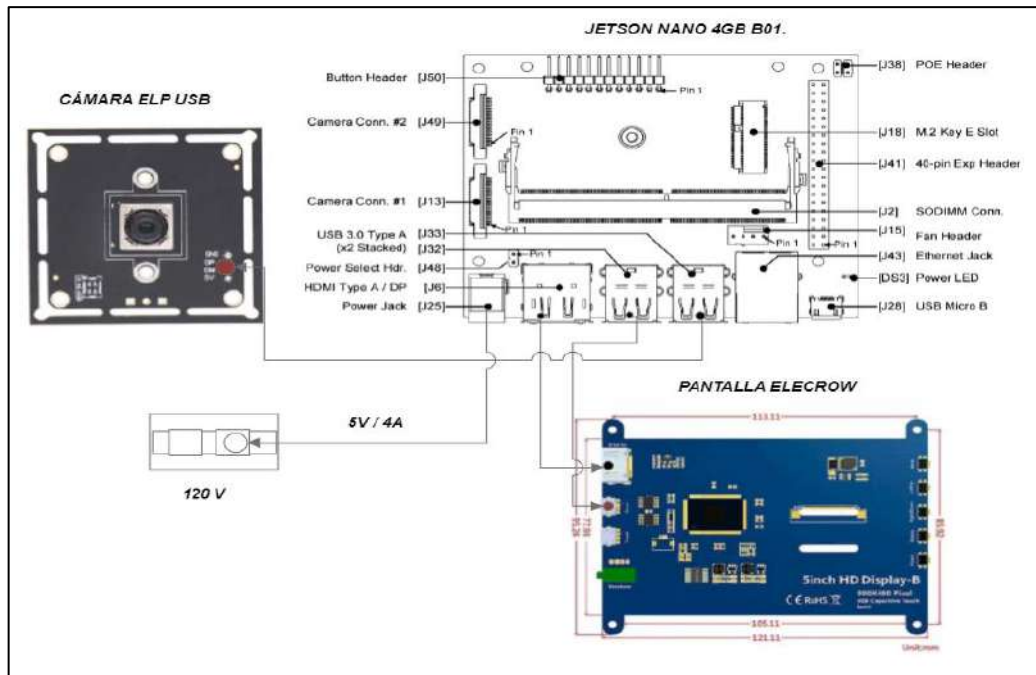


Ilustración 6-3: Esquema de conexión electrónica.

Realizado por: Buñay Javier, Mullo Jose, 2023.

3.6. Selección de *software* de desarrollo

Después de llevar a cabo las investigaciones con el fin de que el prototipo traductor de lenguaje de señas sea económico y eficiente, se trabajó con software no licenciado. Se utilizó el sistema operativo Ubuntu, junto con el paquete de desarrollo de software Jetpack diseñado para trabajar con las plataformas de hardware Jetson. Se optó por el uso de un lenguaje de programación de alto nivel, como Python. Además, se utilizaron bibliotecas como OpenCV, MediaPipe Hand y TensorFlow. A continuación, se detalla la programación realizada en cada fase.

3.6.1. *Adquisición de datos de aprendizaje*

En la Ilustración 7-3, se muestra el diagrama de flujo utilizado para la adquisición de los datos de aprendizaje.

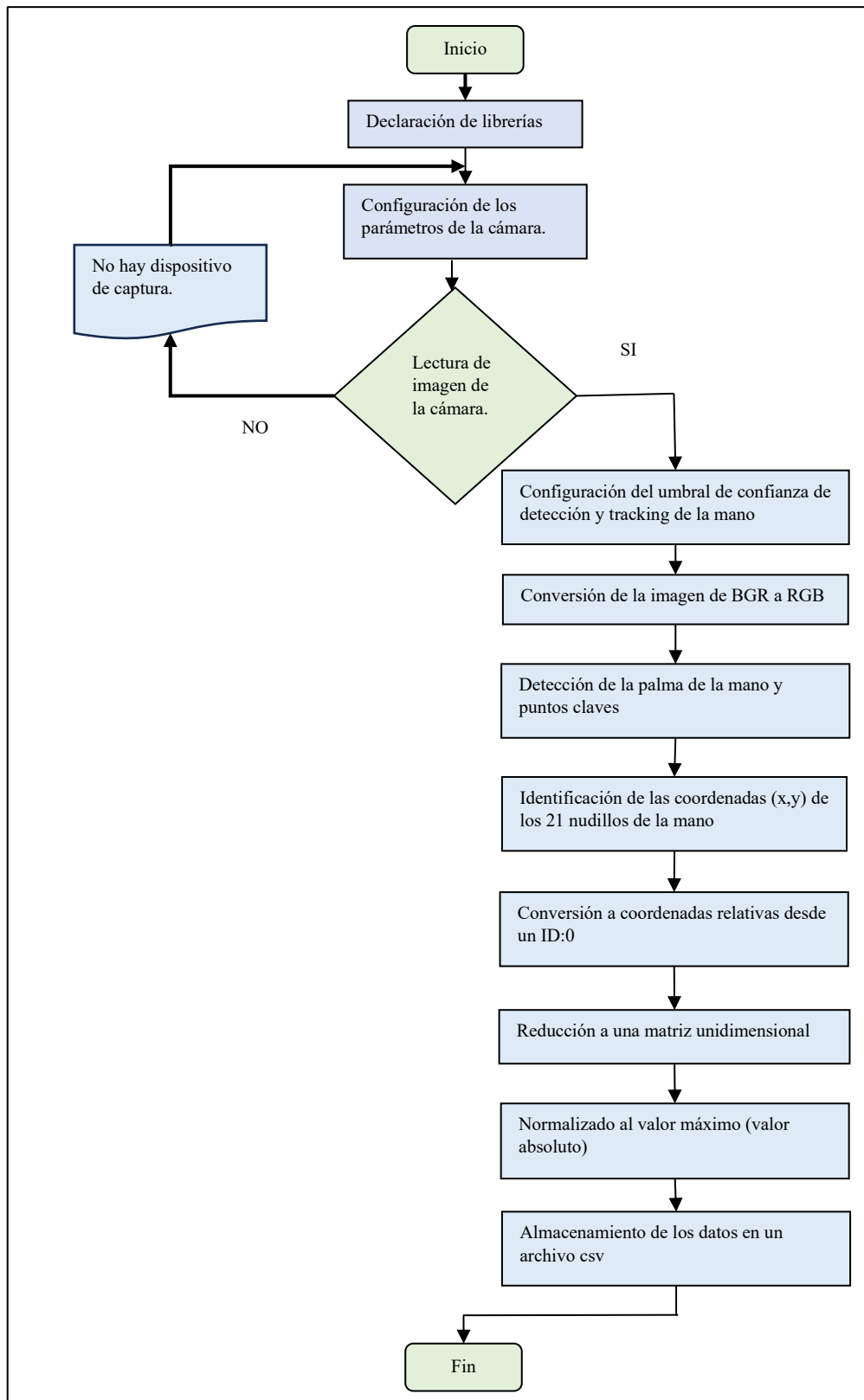


Ilustración 7-3: Diagrama de flujo de la adquisición de los datos de aprendizaje.

Realizado por: Buñay Javier, Mullo Jose, 2023.

A continuación, se describe el algoritmo correspondiente al diagrama de flujo presentado en la Ilustración 7-3.

- Se configuran los parámetros de funcionamiento de la cámara para la captura de imagen.
- Se establecen la configuración de inferencia de MediaPipe, para el número de manos a detectar, el umbral de confianza de detección y el umbral de confianza de seguimiento.
- Se realiza la conversión del formato BGR a RGB.
- El proceso de normalización de los datos se desarrolla de la siguiente manera:
- Se detectan las 21 coordenadas (x, y) de los nudillos de las manos.
- Se convierten a coordenadas relativas, asignando un ID a los datos de cada letra ingresada. Estos números van del 0 al 25 para las 26 letras del alfabeto, ingresados en orden ascendente.
 - Se reducen a una matriz unidimensional.
 - Se normalizan al valor máximo absoluto.
 - Por último, todos los datos se guardan en un archivo CSV que contendrá las coordenadas (x, y) de los 21 nudillos de la mano junto con su respectivo ID.

Las funciones utilizadas en el proceso son:

- **cv.VideoCapture():** Es una función de la biblioteca OpenCV que se emplea para capturar secuencias de video desde cámaras en tiempo real o archivos de video. Permite abrir una conexión con la fuente de video y luego leer cuadros individuales para su procesamiento.
- **mp.hands.Hands():** Es una clase de la biblioteca MediaPipe de Google que se usa para detectar y rastrear manos en imágenes o videos. Permite identificar posiciones, movimientos y gestos de manos, siendo útil para aplicaciones que involucren interacción y seguimiento de manos. Se puede usar para reconocer la orientación de las manos, los puntos de las articulaciones y los gestos específicos en tiempo real.
- **cv.cvtColor(image, cv.COLOR_BGR2RGB):** Es una función de la biblioteca OpenCV en Python que convierte una imagen del espacio de color BGR al espacio de color RGB. Esta conversión es útil cuando se trabaja con imágenes, ya que muchas bibliotecas y aplicaciones utilizan el formato RGB.
- **np.empty():** Es una función de la biblioteca NumPy en Python que crea un nuevo arreglo sin valores iniciales. En lugar de asignar valores predeterminados, esta función simplemente reserva memoria para el arreglo, lo que puede resultar en valores aleatorios o no definidos. Se utiliza cuando se planea asignar valores conocidos al arreglo en una etapa posterior.
- **np.array():** Es una función de la biblioteca NumPy en Python que se utiliza para crear arreglos NumPy a partir de listas u otras secuencias. Permite construir arreglos

multidimensionales que son esenciales para realizar operaciones numéricas y matriciales eficientes en NumPy.

A continuación, se describe configuración de los parámetros de las funciones utilizadas. En la Ilustración 8-3, se muestra los parámetros de configuración de la función `cv.VideoCapture()`.

En el parámetro `cap_device`, se ingresa el número cero para utilizar la cámara por defecto del dispositivo. En cuanto a los parámetros `cap_width` y `cap_height`, se configura en 960 y 540 respectivamente; estos valores hacen referencia a la resolución de 960x540 píxeles.

```
cap = cv.VideoCapture(cap_device)
cap.set(cv.CAP_PROP_FRAME_WIDTH, cap_width)
cap.set(cv.CAP_PROP_FRAME_HEIGHT, cap_height)
```

Ilustración 8-3: Parámetros de configuración de la función `cv.VideoCapture()`.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la Ilustración 9-3 se presenta la configuración de la función `mp_hands.Hands()`.

- En el parámetro `use_static_image`, se establece en “false” para considerar las capturas como una secuencia de imágenes en tiempo real.
- El parámetro `max_num_hands` controla la cantidad máxima de manos que detecta y sigue en este caso, se configura en 1, ya que se necesita detectar una mano. En la Ilustración 10-3 se observa la detección de la mano.
- El parámetro `min_detection_confidence` controla el nivel mínimo de confianza necesario para considerar válida una detección; en este caso, se configura en 0.7.
- El parámetro `min_tracking_confidence` controla el nivel mínimo de confianza necesario para iniciar o mantener el seguimiento de la mano a lo largo del tiempo. Un valor alto garantiza el seguimiento de detecciones confiables, mientras que un valor bajo permite el seguimiento en condiciones menos ideales. En este caso, también se establece en 0.7. Los valores establecidos para estos dos parámetros proporcionaron los mejores resultados. Además, se realizaron configuraciones con valores extremos de 0 y 1; sin embargo, con estos valores se obtuvieron resultados muy pobres.

La Ilustración 11-3 muestra el seguimiento realizado en la mano, la cual está delimitada por un cuadrado.

```
hands = mp_hands.Hands(  
    static_image_mode=use_static_image_mode,  
    max_num_hands=1,  
    min_detection_confidence=min_detection_confidence,  
    min_tracking_confidence=min_tracking_confidence,  
)
```

Ilustración 9-3: Parámetros de configuración de la función `mp_hands.Hands()`.

Realizado por: Buñay Javier, Mullo Jose, 2023.

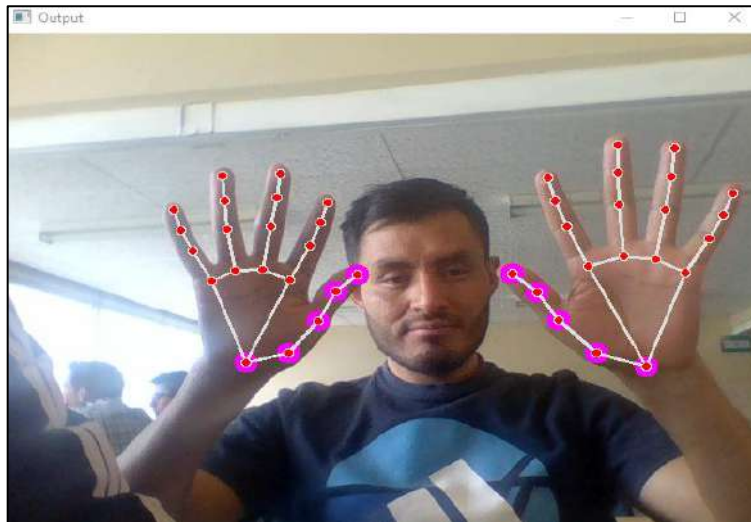


Ilustración 10-3: Detección de los puntos de las dos manos.

Realizado por: Buñay Javier, Mullo Jose, 2023.

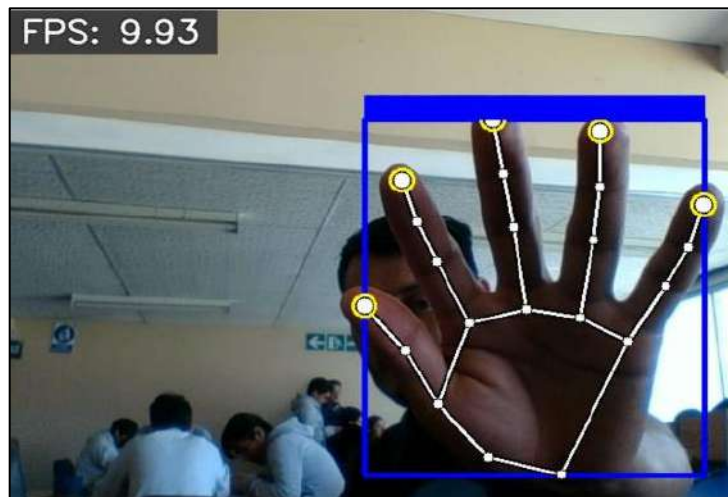


Ilustración 11-3: Seguimiento de la palma de la mano.

Realizado por: Buñay Javier, Mullo Jose, 2023.

3.6.2. Desarrollo de la arquitectura y entrenamiento de la red Neuronal

Para el desarrollo de este prototipo, se incorporaron RNA de tipo Feedforward, implementando la arquitectura del Perceptrón Multicapa. Se adoptó un enfoque basado en aprendizaje supervisado y se aplicaron las funciones de activación ReLU y Softmax. Además, se empleó el algoritmo de aprendizaje Backpropagation. A continuación, se brinda una descripción de cada uno de estos elementos.

En la Ilustración 18-3, se presenta la configuración y la arquitectura de la red neuronal.

3.6.2.1. Red Neuronal Feedforward (FNN)

Las redes neuronales feedforward son un tipo básico de red neuronal en el que la información fluye en una sola dirección, desde la entrada hasta la salida. Están compuestas por capas de neuronas, incluyendo una capa de entrada, capas ocultas y una capa de salida. Cada neurona aplica una función de activación a la suma ponderada de sus entradas y pesos para generar una salida, como se muestra en la Ilustración 12-3.

Modelo matemático:

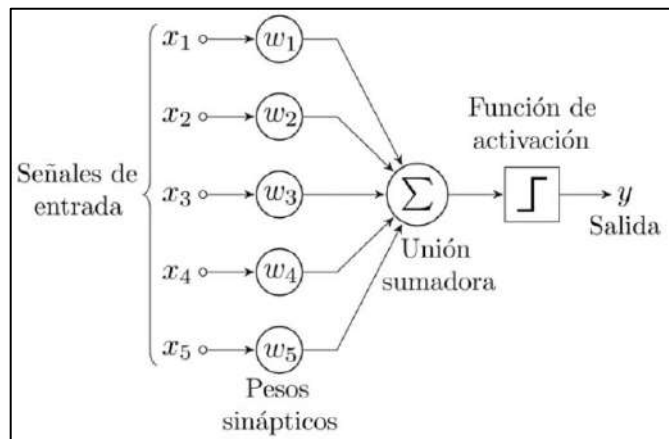


Ilustración 12-3: Esquema de una neurona artificial.

Fuente: <https://telefonicatech.com/blog>

Los valores de entrada son multiplicados por los pesos respectivos, es decir, el vector de entrada se multiplica por el vector de pesos en la sinapsis. Esto da lugar a una combinación lineal de las entradas y los pesos, lo que se conoce como función de ponderación, como lo detalla en la Ecuación 2.

Ecuación 2: Función de ponderación.

$$X * W^t = (x_1, x_2 \dots x_n) * \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ x_n \end{pmatrix} = \sum_{i=1} x_i * w_i \quad (2)$$

A continuación, se empleará la ecuación 3.

Ecuación 3: Función de activación.

$$\Phi \left(\sum_{i=1} x_i * w_i \right) \quad (3)$$

Finalmente, el resultado se propaga hacia la salida, pudiendo ser utilizado como entrada para otra neurona o como respuesta final.

3.6.2.2. *Perceptrón Multicapa (MLP)*

Es un tipo de red neuronal artificial con capas de neuronas organizadas en múltiples niveles. Este diseño permite la detección de relaciones complejas en los datos. La configuración de la red neuronal implementada para el prototipo se puede observar en la Ilustración 18-3 y la arquitectura en la Ilustración 19-3.

3.6.2.3. *Aprendizaje supervisado*

El aprendizaje supervisado es un enfoque en el aprendizaje automático donde un modelo se entrena usando ejemplos de entrada y las respuestas correctas correspondientes. El modelo ajusta sus parámetros para predecir de manera precisa las salidas deseadas a partir de nuevos datos.

3.6.2.4. *Funciones de activación*

Una función de activación en redes neuronales es una fórmula matemática aplicada a las salidas de las neuronas para introducir no linealidad y permitir la captura de patrones complejos en los datos. Determina si una neurona se activa o no según su entrada. La configuración de las funciones

de activación en la red neuronal implementada para el prototipo puede observarse en la Ilustración 18-3.

3.6.2.5. Función ReLU

Introduce no linealidad en las redes, es eficiente en cálculos y evita el desvanecimiento de gradientes en capas profundas. Esta función genera el valor x si es positivo y cero si es negativo, se puede observar su comportamiento en la Ilustración 13-3.

Ecuación 4: Función de activación ReLU.

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x > 0 \end{cases} \quad (4)$$

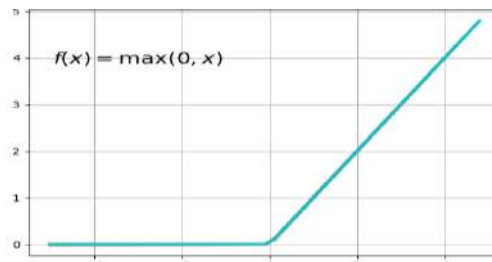


Ilustración 13-3: Función de activación ReLU.

Fuente: <https://www.codificandobits.com/blog>

3.6.2.6. Función SOFTMAX

Convierte salidas en una distribución de probabilidades. Para un vector de valores de entrada, calcula la probabilidad de cada clase dividiendo la exponencial de cada valor por la suma de todas las exponenciales. Esto permite asignar una entrada a una de varias clases posibles. Se puede observar su comportamiento en la Ilustración 14-3.

Ecuación 5: Función de activación SOFTMAX.

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (5)$$

Donde:

- e representa la constante del logaritmo natural, también conocida como el número de Euler.

- e^{z_i} denota el valor correspondiente a la entrada relacionada con la clase i .
- El denominador es la suma de todas las exponenciales de las entradas, lo cual se hace para normalizar y garantizar que las probabilidades sumen 1.

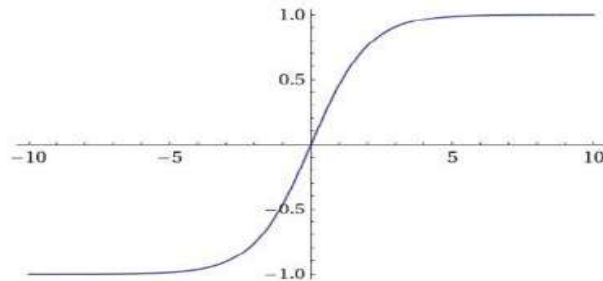


Ilustración 14-3: Función de activación SOFTMAX.

Fuente: <https://www.codificandobits.com/blog>

3.6.2.7. Algoritmo de aprendizaje Backpropagation

Es un algoritmo clave en el entrenamiento de redes neuronales. Se utiliza para ajustar los pesos y sesgos de la red con el fin de mejorar la precisión de las predicciones. Funciona propagando los datos de entrada a través de la red y calculando el error entre las salidas reales y las predicciones. Luego, el error se propaga hacia atrás, ajustando los pesos y sesgos en función de cómo contribuyeron al error. Este proceso se repite en varias iteraciones hasta que la red se ajusta adecuadamente. El esquema del algoritmo se muestra en la Ilustración 15-3.

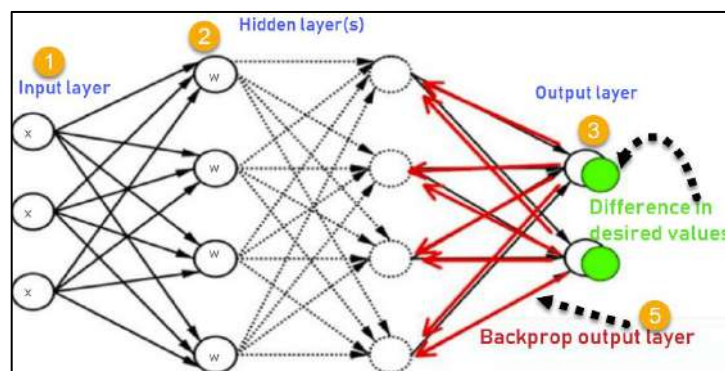


Ilustración 15-3: Algoritmo Backpropagation.

Fuente: <https://www.codificandobits.com/blog>

En la Ilustración 16-3, se presenta el diagrama de flujo utilizado para el desarrollo de la arquitectura y entrenamiento de la red neuronal.

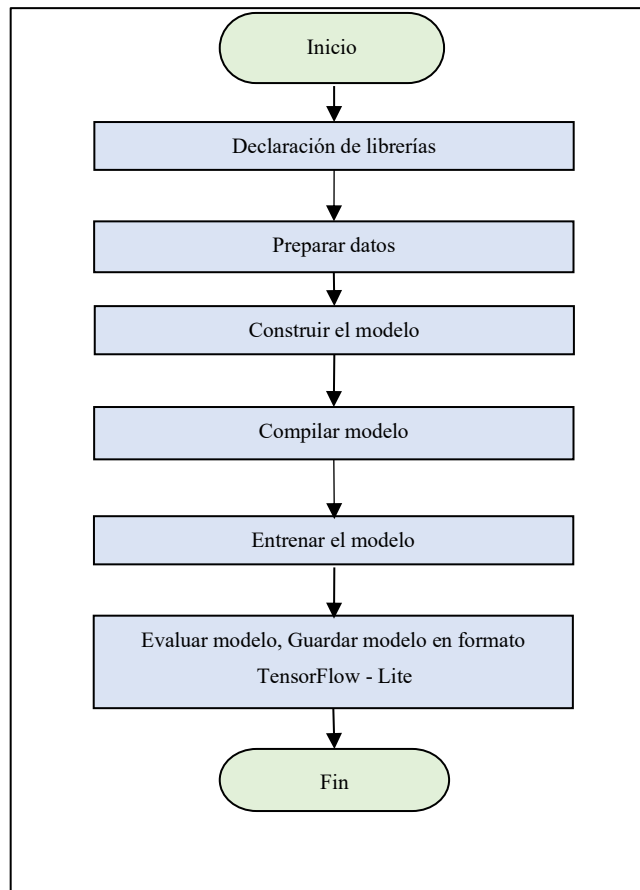


Ilustración 16-3: Diagrama de flujo del entrenamiento de la red neuronal.

Realizado por: Buñay Javier, Mullo Jose, 2023.

A continuación, se describe en detalle el algoritmo correspondiente al diagrama de flujo presentado en la Ilustración 16-3.

- Importación de las librerías necesarias para el desarrollo de la arquitectura y el entrenamiento de la red neuronal.
- Importación de los archivos CSV que contienen las coordenadas (x, y) de los 21 nudillos de la mano.
- División de los datos de entrenamiento en conjuntos de entrenamiento y prueba.
- Diseño de la arquitectura de la red neuronal, la cual consta de una capa de entrada, dos capas ocultas y una capa de salida, junto con las funciones de activación.
- Conversión de un modelo de TensorFlow a un modelo de TensorFlow Lite.

Las funciones utilizadas en este proceso son:

- **train_test_split:** Se utiliza para dividir los datos en un conjunto de entrenamiento y pruebas.
- **tf.keras.models.Sequential:** Permite la construcción de modelos de redes neuronales secuenciales a permitir que las capas se agregan una tras otra.
- **model.fit:** Se utiliza para entrenar modelos de redes neuronales. Permite configurar y realizar el proceso de entrenamiento al proporcionar los datos de entrada y las etiquetas correspondientes. Durante el entrenamiento, ajusta los pesos del modelo para minimizar la función de pérdida definida.
- **tf.lite.TFLiteConverter.from_keras_model():** Convierte modelos de Keras a un formato compatible con TensorFlow Lite (TF Lite), permitiendo ejecutar modelos de aprendizaje profundo en dispositivos con limitaciones de recursos.

A continuación, se describe configuración de los parámetros de las funciones utilizadas.

La Ilustración 17-3 contiene los parámetros de configuración de la función **train_test_split**. Se configura utilizando el 75% de los datos para el entrenamiento y el 25% para las pruebas.

```
X_train, X_test, y_train, y_test = train_test_split(X_dataset, y_dataset, train_size=0.75, random_state=RANDOM_SEED)
```

Ilustración 17-3: Configuración de la función train test split.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la Ilustración 18-3, se presenta la configuración de la función **tf.keras.models.Sequential**. Se establece la configuración de la red neuronal con una capa de entrada de 42 neuronas, seguida por dos capas ocultas: una con 20 neuronas y otra con 25 neuronas, ambas empleando la función de activación ReLu. Finalmente, se incluye una capa de salida con 26 neuronas, donde se aplica la función de activación Softmax, se presenta el esquema de la red neuronal en la Ilustración 19-3.

```
# Construcción de modelos
model = tf.keras.models.Sequential([
    tf.keras.layers.Input((42, )),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(20, activation='relu'),
    tf.keras.layers.Dropout(0.1),
    tf.keras.layers.Dense(25, activation='relu'),
    tf.keras.layers.Dense(26, activation='softmax')
])
```

Ilustración 18-3: Configuración de la función tf.keras.models.Sequential.

Realizado por: Buñay Javier, Mullo Jose, 2023.

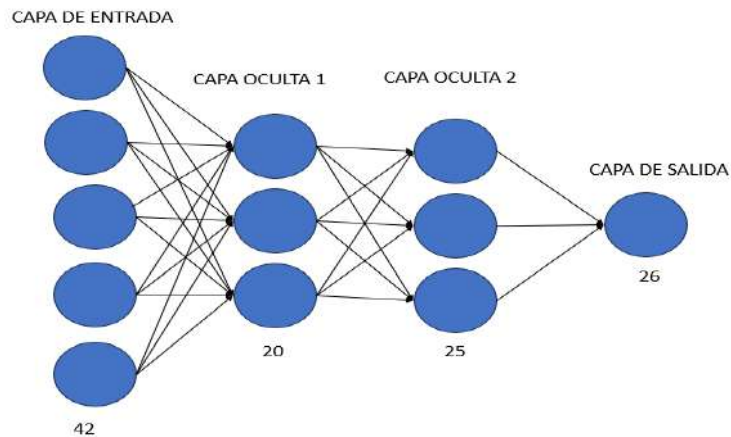


Ilustración 19-3: Esquema de la red neuronal.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la Ilustración 20-3, se detallan los parámetros de configuración de la función `model.fit`. Dicha función se encarga de entrenar el modelo durante un número de épocas predefinido. En este caso particular, el entrenamiento se extiende a lo largo de 1000 épocas, utilizando 128 muestras para cada actualización del gradiente. Los datos utilizados para evaluar la pérdida y las métricas del modelo corresponden a `X_test` e `Y_test`. Por último, los `callbacks` consisten en una lista de instancias que controlan el proceso de entrenamiento.

```

model.fit(
    X_train,
    y_train,
    epochs=1000,
    batch_size=128,
    validation_data=(X_test, y_test),
    callbacks=[cp_callback, es_callback]
)

```

Ilustración 20-3: Configuración de la función `model.fit`.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la Ilustración 21-3, se presentan los parámetros de configuración de la función `tf.lite.TFLiteConverter.from_keras_model()`.

```

converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_quantized_model = converter.convert()

```

Ilustración 21-3: Función `tf.lite.TFLiteConverter.from_keras_model()`.

Realizado por: Buñay Javier, Mullo Jose, 2023.

3.6.3. Proceso de traducción y visualización

En la Ilustración 22-3 se muestra el diagrama de flujo que se utiliza para este proceso.

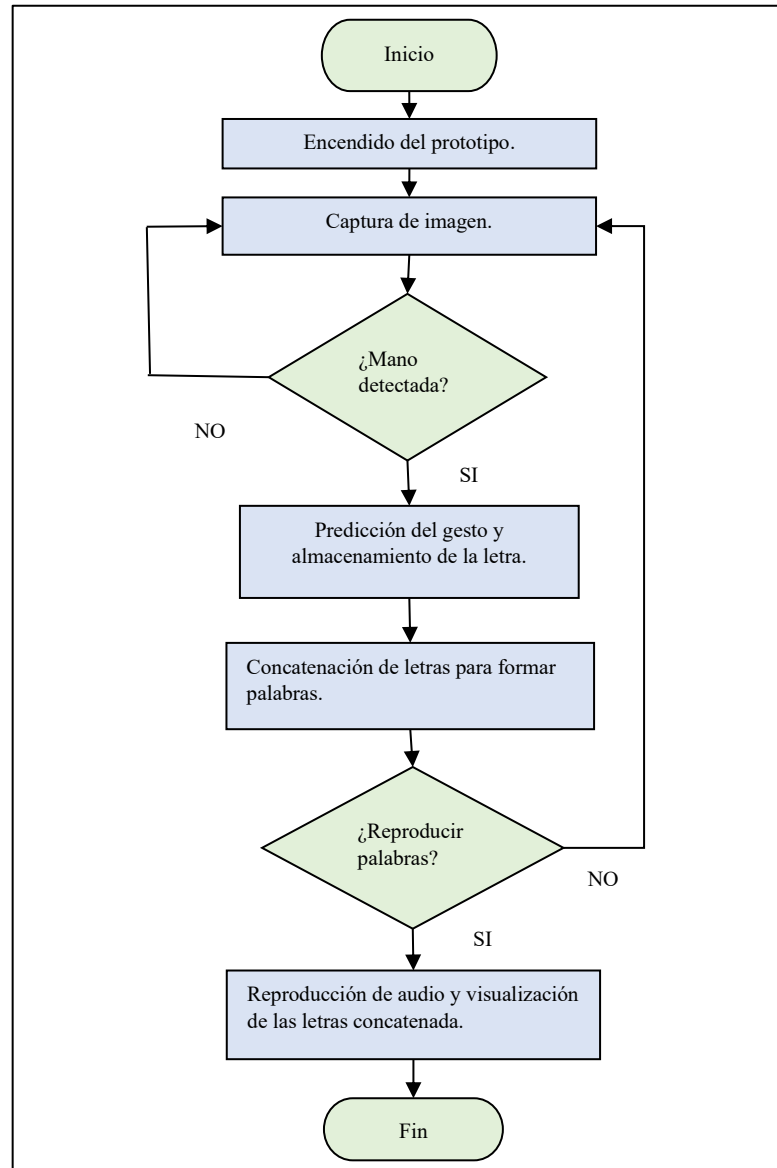


Ilustración 22-3: Diagrama de flujo del proceso de traducción.

Realizado por: Buñay Javier, Mullo Jose, 2023.

A continuación, se proporciona un detalle del algoritmo correspondiente al diagrama de flujo presentado en la Ilustración 22-3.

- Se configuran los parámetros de funcionamiento de la cámara para la captura de imágenes.

- Se establecen las configuraciones de inferencia de MediaPipe, incluyendo el número de manos a detectar, el umbral de confianza de detección y el umbral de confianza de seguimiento.
- Se realiza la conversión del formato BGR a RGB.
- El proceso de normalización de los datos ingresados se desarrolla de la siguiente manera:
- Se detectan las 21 coordenadas (x, y) de los nudillos de las manos.
- Estas coordenadas se reducen a una matriz unidimensional.
- Luego, se normalizan para ajustarse al valor máximo absoluto.
- Se procede a realizar la inferencia de los datos ingresados. Luego, se asigna una seña a una letra del alfabeto común, de modo que cada seña corresponde a una letra específica. Posteriormente, estas letras se concatenan para formar palabras o mensajes completos.
- Finalmente, se procede a reproducir y visualizar la información generada.

Las funciones utilizadas en este proceso son:

- Se utilizan las funciones **cv.VideoCapture()**, **mp.hands.Hands()**, **cv.cvtColor(image, cv.COLOR_BGR2RGB)** y **np.empty()**, las cuales se describen detalladamente en el subcapítulo 3.6.1
- **gtts**: Es una biblioteca de Python permite la conversión de texto en voz. Esta herramienta genera archivos de audio a partir de texto y posteriormente reproducirlos o almacenarlos. Esta biblioteca hace uso de la tecnología de síntesis de voz de Google para llevar a cabo la conversión de texto a voz de manera sencilla y eficiente.
- **pygame**: Es una biblioteca de Python utilizada para el desarrollo de videojuegos, aplicaciones multimedia y gráficos interactivos. Ofrece soporte para la reproducción de sonidos y música, lo que facilita la incorporación de efectos de sonido a las aplicaciones.

A continuación, se describe la configuración de los parámetros de las funciones utilizadas.

En la Ilustración 23-3, se presentan los parámetros de configuración de la biblioteca **gtts()** y **pygame.init()**, la instancia "tts = gTTS (text=texto, lang='es')" se encarga de interpretar el texto en español. El resultado se almacena en un archivo cuyo nombre de variable es "nombre_archivo". Para reproducir la oración o palabra interpretada, se utiliza la clase "pygame.mixer.music.load(nombre_archivo)" de la librería pygame. Es importante tener en cuenta que la biblioteca GTTS requiere una conexión a Internet para comunicarse con la API de Google Text-to-Speech para generar la salida de voz.

```
def texto_a_voz(texto, nombre_archivo):
    tts = gTTS(text=texto, lang='es')
    tts.save(nombre_archivo)

    # Inicializar Pygame
    pygame.init()

    # Reproducir el archivo de audio con Pygame
    pygame.mixer.music.load(nombre_archivo)
    pygame.mixer.music.play()

    # Esperar hasta que finalice la reproducción
    while pygame.mixer.music.get_busy():
        continue

    # Cerrar Pygame
    pygame.quit()
```

Ilustración 23-3: Bibliotecas gTTS y pygame para la traducción de texto a voz.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la Ilustración 24-3, se presenta la concatenación de la palabra "Hola", la cual se construye al ingresar individualmente cada una de las letras que la componen.

```
H
HO
HOL
HOLA
Exception in Tkinter callback
Traceback (most recent call last):
```

Ilustración 24-3: Concatenación de la palabra HOLA.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la Ilustración 25-3, el prototipo dispone de una interfaz más amigable para el usuario final. En esta interfaz, las letras introducidas se presentan de manera clara, y mediante un proceso de concatenación, se genera la palabra completa. Adicionalmente, se ha implementado la funcionalidad de reproducir el mensaje generado a través de un altavoz, enriqueciendo así la experiencia del usuario.

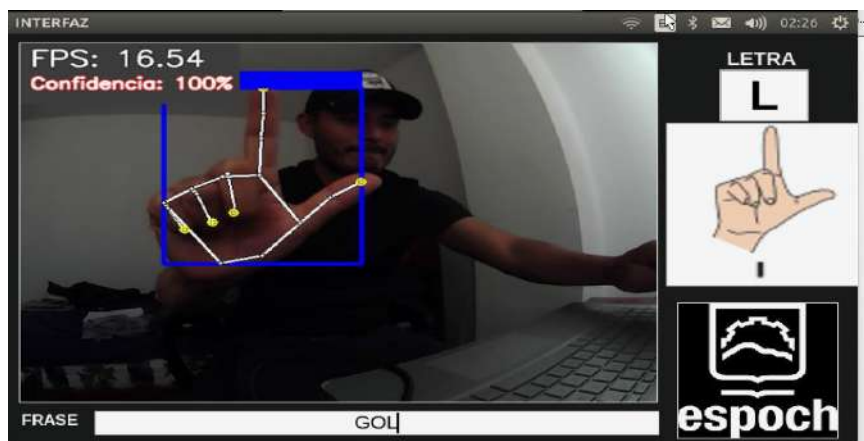


Ilustración 25-3: Interfaz final del prototipo.

Realizado por: Buñay Javier, Mullo Jose, 2023.

3.7. Diseño estructural

El diseño estructural se simuló en SolidWorks, teniendo en cuenta cuidadosamente aspectos importantes que incluyeron la forma, la disposición del hardware en el interior de la estructura, la funcionalidad y la estabilidad del prototipo.

En la Ilustración 26-3, se muestran las siguientes vistas del diseño estructural del prototipo: a) Vista Frontal, b) Vista Lateral y c) Vista Posterior.

Cada una de estas perspectivas ofrece una comprensión detallada de la estructura y su disposición.

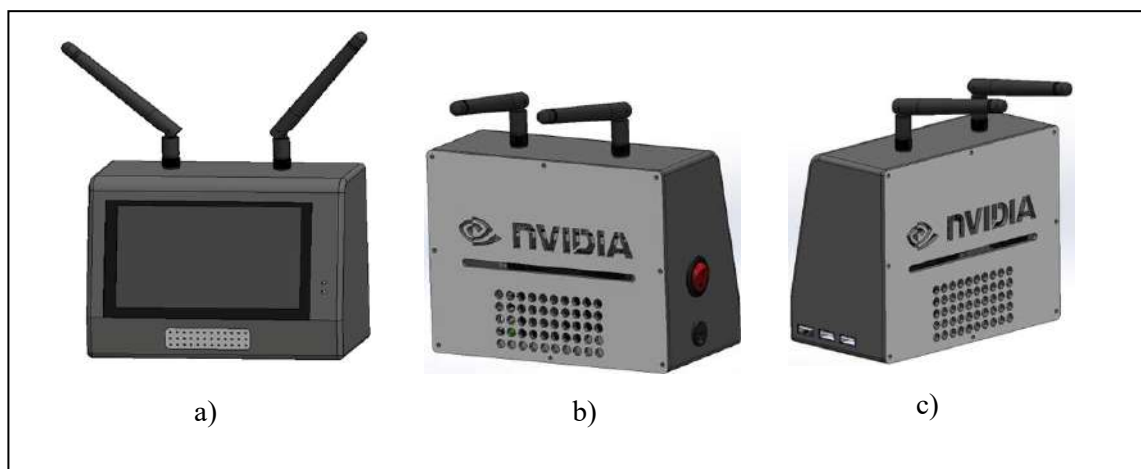


Ilustración 26-3: Vista externa del prototipo en SolidWorks

Realizado por: Buñay Javier, Mullo Jose, 2023

En la Ilustración 27-3 se pueden observar los componentes electrónicos ensamblados en el interior de la estructura del prototipo. En la Ilustración 28-3 se observa el resultado final, donde la estructura está fabricada con plástico impreso mediante una impresora 3D utilizando filamento PLA.



Ilustración 27-3: Vista interna del prototipo en SolidWorks.

Realizado por: Buñay Javier, Mullo Jose, 2023.



Ilustración 28-3: Implementación final del prototipo.

Realizado por: Buñay Javier, Mullo Jose, 2023.

CAPÍTULO IV

4. VALIDACIÓN DE PROTOTIPO

En el presente capítulo se proporcionan detalles sobre las pruebas y resultados realizados con el Prototipo electrónico traductor de lenguaje de señas a voz. Estas pruebas abarcan la evaluación del desempeño de la red neuronal, el funcionamiento en diferentes distancias, la medición del consumo total de corriente y, por último, el análisis de costos del prototipo.

4.1. Pruebas del desempeño de la red neuronal

Para evaluar el rendimiento de la red neuronal, utilizaremos la matriz de confusión, que tiene aplicación en el ámbito de la IA y RNA. Esta matriz proporciona una evaluación más exhaustiva y mayor información acerca del desempeño del modelo. La matriz de confusión no solamente permite calcular la precisión de un clasificador, ya sea en términos de precisión global o por clases, sino que también facilita el cálculo de otras métricas cruciales para evaluar el modelo de la red neuronal.

En la Ilustración 1-4 se muestra una matriz de confusión que se presenta en forma de tabla, donde cada columna representa el número de predicciones para cada clase, y cada fila muestra el número real de instancias correspondientes a cada clase.

		Valores	
		+ve	-ve
Valores de predicción	+ve	45	8
	-ve	15	32

Las flechas en el diagrama indican: TP (True Positive) apunta a la celda (45), FP (False Positive) apunta a la celda (8), FN (False Negative) apunta a la celda (15) y TN (True Negative) apunta a la celda (32).

Ilustración 1-4: Matriz de confusión.

Realizado por: Buñay Javier, Mullo Jose, 2023.

A continuación, se ofrece una explicación de la matriz de confusión que se muestra en la Ilustración 1-4, junto con las métricas asociadas:

- **Verdadero positivo (TP).** Representa una muestra que pertenece a la clase positiva que se clasifica correctamente.

- **Verdadero Negativo (TN).** Representa una muestra que pertenece a la clase negativa que se clasifica correctamente.
- **Falso positivo (FP).** Representa una muestra que pertenece a la clase negativa, pero que se clasifica incorrectamente como perteneciente a la clase positiva.
- **Falso negativo (FN).** Representa una muestra que pertenece a la clase positiva, pero que se clasifica incorrectamente como perteneciente a la clase negativa.

4.1.1. Métricas de evaluación de la matriz de confusión.

En términos generales, podemos obtener las siguientes métricas cuantitativas de evaluación a partir de la matriz de confusión entrenada:

Ecuación 6: Métrica Exactitud (Accuracy).

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

La Ecuación 6 permite calcular el número de muestras clasificadas correctamente de entre todas las muestras presentes en el conjunto de prueba.

Ecuación 7: Métrica Precisión (Precision).

$$Precisión = \frac{TP}{TP + FP} \quad (7)$$

La Ecuación 7 permite calcular el número de muestras que realmente pertenecen a la clase positiva de todas las muestras que el modelo *predijo* que serían de la clase positiva.

Ecuación 8: Métrica Recuperación o Sensibilidad (Recall o Sensitivity).

$$Sensibilidad = \frac{TP}{TP + FN} \quad (8)$$

La Ecuación 8 permite calcular el número de muestras predichas correctamente como pertenecientes a la clase positiva de todas las muestras que *realmente pertenecen* a la clase positiva.

Ecuación 9: Métrica Puntuación (F1 Score)

$$F1 - Score = \frac{2x \text{ Precisión } x \text{ Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}} \quad (9)$$

La Ecuación 9 permite calcular la media armónica de las puntuaciones de precisión y recuperación obtenidas para la clase positiva.

Ecuación 10: Métrica de Especificidad (Specificity).

$$\text{Especificidad (Specificity)} = \frac{TN}{FP + TN} \quad (10)$$

La Ecuación 10 permite calcular el número de muestras predichas correctamente para estar en la clase negativa de todas las muestras en el conjunto de datos que realmente pertenecen a la clase negativa.

Las métricas calculadas por clase incluyen precisión y recuperación, mientras que las métricas restantes se calculan de manera global.

4.1.2. Evaluación de matriz de confusión del prototipo traductor de lenguaje de señas

Para analizar el rendimiento del modelo de la red neuronal en el reconocimiento de letras de la lengua de señas, se empleó la matriz de confusión. El modelo fue entrenado para clasificar las letras del alfabeto de la lengua de señas en 26 clases que representan las letras de dicho alfabeto. En la Ilustración 2-4, se muestra la matriz de confusión resultante:

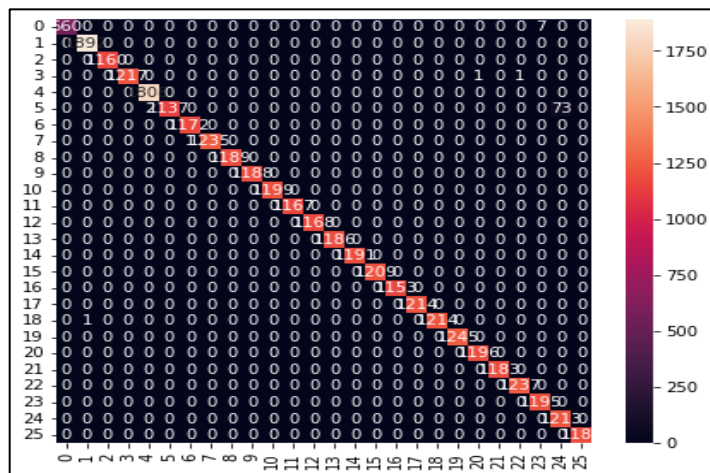


Ilustración 2-4: Matriz de confusión del prototipo traductor.

Realizado por: Buñay Javier, Mullo Jose, 2023.

En la matriz de confusión, cada fila representa una letra real de la lengua de señas, y cada columna muestra la letra predicha por el modelo. Los valores en la diagonal principal indican las predicciones correctas, es decir, los Verdaderos Positivos, para cada letra de la lengua de señas. Mientras que los valores fuera de la diagonal principal reflejan las predicciones incorrectas, que comprenden los Falsos Positivos y los Falsos Negativos.

Se utilizaron métricas adicionales, como la precisión, la recuperación y el F1-Score, para realizar una evaluación detallada del desempeño del modelo de la red neuronal.

	PRECISION	RECALL	F1-SCORE
0	0.99	0.92	0.96
1	1.00	0.98	0.99
2	1.00	0.99	0.99
3	1.00	1.00	1.00
4	0.99	0.99	0.99
5	0.99	0.99	0.99
6	0.96	1.00	0.98
7	1.00	1.00	1.00
8	1.00	0.99	0.99
9	0.99	1.00	1.00
10	0.99	1.00	0.99
11	1.00	1.00	1.00
12	1.00	0.97	0.99
13	0.86	0.92	0.89
14	0.91	0.86	0.89
15	1.00	1.00	1.00
16	1.00	1.00	1.00
17	0.76	0.99	0.86
18	0.90	0.99	0.94
19	1.00	0.95	0.98
20	0.93	0.85	0.89
21	0.98	0.98	0.98
22	1.00	1.00	1.00
23	1.00	0.99	0.99
24	1.00	0.70	0.82
25	0.67	1.00	0.80
26	0.88	1.00	0.94
Accuracy			0.95
Macro Avg	0.96	0.96	0.96
Weighted Avg	0.96	0.95	0.95

Ilustración 3-4: Reporte del entrenamiento.

Realizado por: Buñay Javier, Mullo Jose, 2023.

Como se aprecia en la Ilustración 3-4, se observaron métricas de evaluación que demostraron una precisión destacada en el rendimiento del modelo para cada clase. Estas métricas superaron el umbral del 70%, lo que reflejó un rendimiento bueno del modelo. En términos de precisión, se evidenció una alta exactitud en las predicciones positivas. La recuperación promedio indicó una identificación efectiva de casos positivos, mientras que el F1-Score promedio reflejó un equilibrio efectivo entre precisión y recuperación.

4.2. Pruebas de aciertos y porcentaje de precisión de detección con relación a la distancia utilizando la mano derecha

Para llevar a cabo dicha prueba, se llevaron a cabo mediciones de distancias mediante un flexómetro, con unidades de medida expresadas en centímetros. Se realizaron pruebas para

evaluar la funcionalidad del prototipo a distancias de 30 cm, 60 cm y 100 cm, analizando su impacto en los aciertos y la precisión en la detección del alfabeto de la LSEC.

Para obtener los datos, se llevaron a cabo 30 pruebas para cada letra del alfabeto de la LSEC. Estas pruebas se realizaron en un ambiente controlado que mantiene una condición lumínica óptima. De acuerdo con la Norma Técnica Ecuatoriana INEN 1-153 establecida por el Instituto Ecuatoriano de Normalización, una condición lumínica óptima se define en un rango de 200 a 400 Luxes.

En las pruebas, se procedió a colocar la mano a distancias predefinidas de 30 cm, 60 cm y 100 cm con respecto a la cámara. Durante estos posicionamientos, se ingresaron las señas correspondientes a las letras del alfabeto de la LSEC con el objetivo de evaluar los aciertos y determinar el porcentaje de precisión en la detección de cada letra. A continuación, se muestra los datos resultantes tras la ejecución de las pruebas.

4.2.1. Pruebas a una distancia de 30 cm

4.2.1.1. Pruebas de la letra A

Se procede a realizar el análisis de los datos presentados en el *Anexo A*. Dado que el número de pruebas es extenso, se optó por analizar exclusivamente la letra "A", ya que todas las demás letras cuentan con la misma cantidad de pruebas y siguen un análisis similar

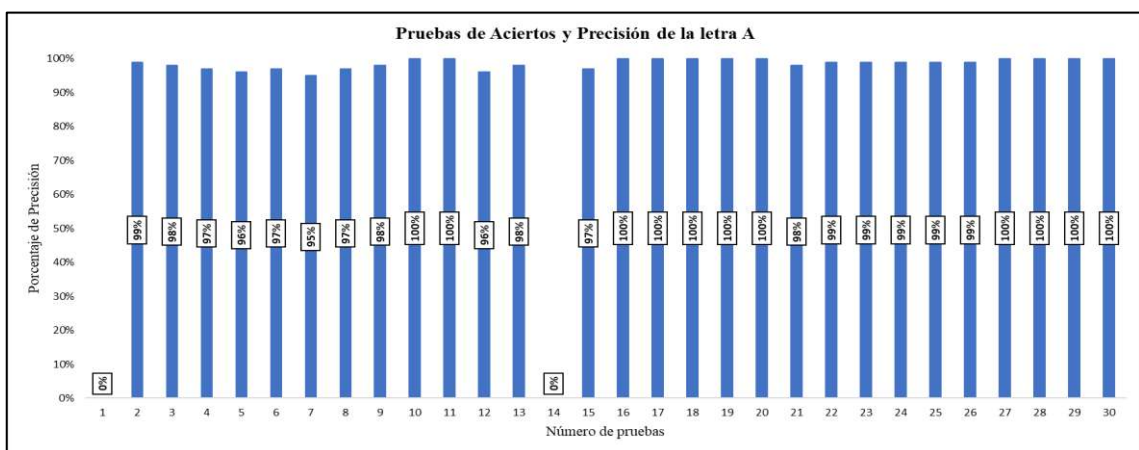


Ilustración 4-4: Pruebas de aciertos y precisión de detección a una distancia de 30 cm para la letra “A” con la mano derecha.

Realizado por: Buñay Javier, Mullo Jose, 2023.

Con base en la Ilustración 4-4, se pueden observar las 30 pruebas realizadas, en las cuales se obtuvieron 28 aciertos y 2 desaciertos en las pruebas 1 y 14. Estas dos últimas presentaron un porcentaje de precisión de detección del 0%. En cambio, los aciertos tuvieron porcentajes de precisión de detección que superaron el 90%.

Para evaluar la efectividad del prototipo, es fundamental determinar el promedio del porcentaje de precisión de detección obtenido en cada acierto de cada letra. A continuación, se calcula el promedio del porcentaje de detección de la letra "A" utilizando la Ecuación 11, que suma los porcentajes de precisión de las 30 pruebas y los divide por el número total de pruebas, que en este caso también son 30. Cabe destacar que se llevará a cabo el mismo análisis para las demás letras.

Se llevó a cabo la evaluación de la efectividad del prototipo, siendo fundamental la determinación del promedio del porcentaje de precisión de detección logrado en cada acierto de todas las letras. Posteriormente, se calcula el promedio del porcentaje de detección de la letra "A" mediante la Ecuación 11. Dicha ecuación suma los porcentajes de precisión de las 30 pruebas y los divide por el número total de pruebas, que en este caso también fue de 30. Cabe destacar que se llevó a cabo el mismo análisis para las demás letras.

Ecuación 11: Media de porcentaje de precisión.

$$\text{Media de porcentaje de precisión} = \frac{\sum_{i=1,2,\dots}^{30} \text{precisión}}{\text{Total}} = 92,07\% \quad (11)$$

El resultado obtenido en ese análisis permite determinar con qué nivel de precisión el prototipo detectó la letra "A" con relación a la mano derecha. El valor medio del porcentaje de precisión obtenido es del 92,07 %. Por lo tanto, se concluye que el resultado obtenido es excelente.

4.2.1.2. Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)

Para realizar esta prueba se llevó a cabo el mismo análisis descrito en el sub capítulo anterior (4.2.1.1.) aplicada para cada letra del alfabeto. El objetivo de esta prueba fue evaluar la eficiencia del prototipo en la predicción de las letras del alfabeto dentro de la distancia establecida. Los resultados obtenidos se encuentran detallados en la Tabla 1-4.

Con base en la Tabla 1-4, se puede observar que la mayoría de las letras tiene una precisión que supera el 90%, con un número de aciertos que sobrepasa los 28 de las 30 pruebas realizadas a cada una de las letras. Sin embargo, es importante señalar que la letra "R" mostró el menor número

de aciertos y el porcentaje de detección más bajo. En las Ilustraciones 5-4 y 6-4 se presentan los resultados, donde la línea horizontal roja representa las medias de aciertos y precisión.

Tabla 1-4: Resultados de las pruebas a una distancia de 30 cm

<i>Letras</i>	<i>Total, de aciertos</i>	<i>Medias de precisión</i>
A	28	92,07%
B	27	88,77%
C	30	99,7%
D	29	95,43%
E	27	88,6%
F	29	91,53%
G	28	91,17%
H	28	91,8%
I	30	97,17%
J	30	99,6%
K	29	94,63%
L	29	96,27%
M	30	99,17%
N	29	96,03%
O	30	99,4%
P	29	96,23%
Q	30	99,3%
R	22	75,43%
S	27	89,07%
T	30	99,1%
U	26	85,97%
V	27	86,3%
W	30	98,9%
X	29	95,87%
Y	30	99,4%
Z	28	92,67%

Realizado por: Buñay Javier, Mullo Jose, 2023.

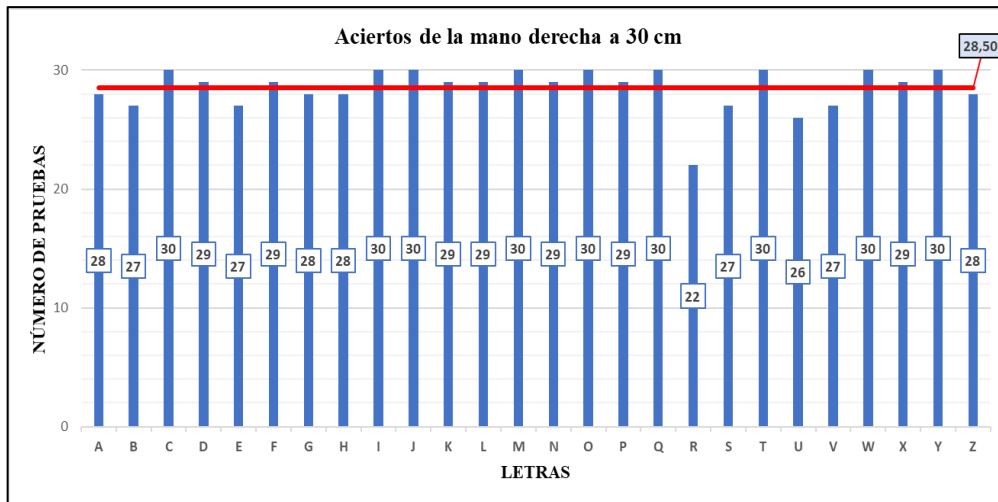


Ilustración 5-4: Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm.

Realizado por: Buñay Javier, Mullo Jose, 2023.

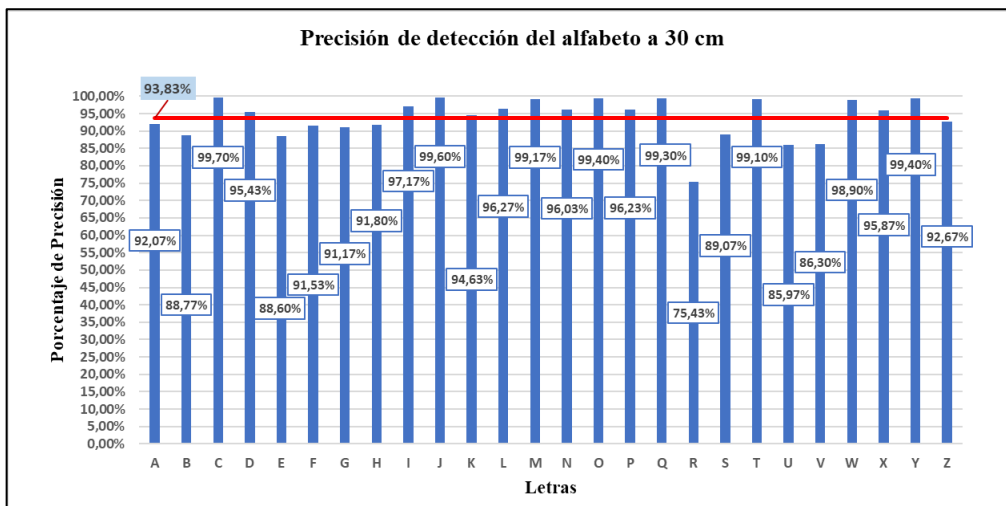


Ilustración 6-4: Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 30cm.

Realizado por: Buñay Javier, Mullo Jose, 2023.

4.2.2. Pruebas a una distancia de 60 cm.

4.2.2.1. Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC).

Para realizar estas pruebas, se llevó a cabo el mismo análisis descrito en el sub capítulo (4.2.1), con la única diferencia de que se realizó a una distancia de 60 cm.

Con base en la Tabla 2-4, se puede observar que la mayoría de las letras tuvieron una precisión que supera el 90%, con un número de aciertos que sobrepasa los 28 de las 30 pruebas realizadas

a cada una de las letras. No obstante, es importante señalar que la letra "R" presentó el menor número de aciertos y el porcentaje de detección más bajo. En las Ilustraciones 7-4 y 8-4 se presentan los resultados, donde la línea horizontal roja representa las medias de aciertos y precisión.

Tabla 2-4: Resultados de las pruebas a una distancia de 60 cm.

<i>Letras</i>	<i>Total, de aciertos</i>	<i>Medias de precisión</i>
A	28	92,40%
B	28	92,03%
C	30	99,33%
D	28	92,07%
E	28	92,17%
F	29	95,23%
G	30	98,53%
H	27	87,90%
I	30	97,87%
J	30	98,33%
K	29	95,57%
L	29	95,47%
M	30	98,23%
N	28	92,50%
O	30	98,73%
P	24	81,83%
Q	30	98,83%
R	20	68,97%
S	28	89,63%
T	29	95,80%
U	27	87,20%
V	29	95,13%
W	30	98%
X	28	87,53%
Y	30	99,10%
Z	29	95,70%

Realizado por: Buñay Javier, Mullo Jose, 2023.

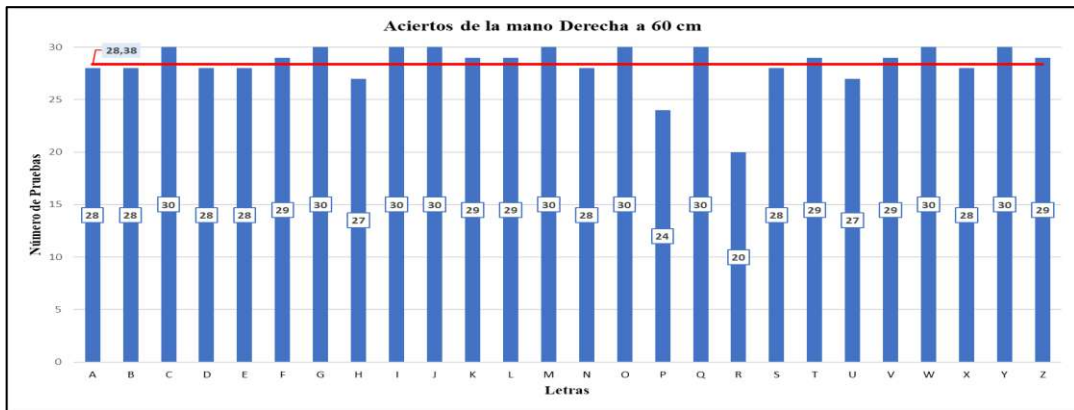


Ilustración 7-4: Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm.

Realizado por: Buñay Javier, Mullo Jose, 2023.

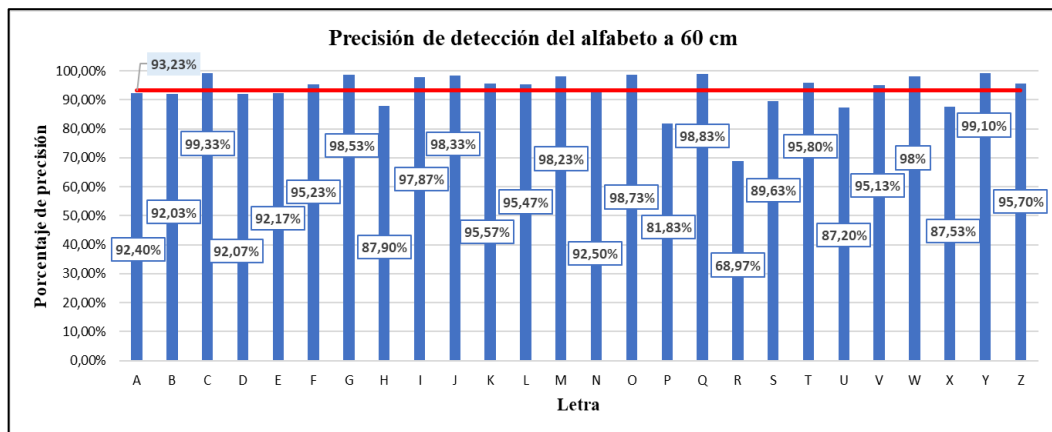


Ilustración 8-4: Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm.

Realizado por: Buñay Javier, Mullo Jose, 2023.

4.2.3. Pruebas a una distancia de 100 cm

4.2.3.1. Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)

Para realizar estas pruebas se llevó a cabo el mismo análisis descrito en el sub capítulo (4.2.1.), con la única diferencia de que se realizó a una distancia de 100 cm.

Con base en la Tabla 3-4, se puede observar que la mayoría de las letras tuvieron una precisión que supera el 90%, con un número de aciertos que sobrepasa los 27 de las 30 pruebas realizadas a cada una de las letras. Es importante señalar que la letra "R" presentó el menor número de aciertos y el porcentaje de detección más bajo. En las Ilustraciones 9-4 y 10-4 se presentan los resultados, donde la línea horizontal roja representa las medias de aciertos y precisión.

Tabla 3-4: Resultados de las pruebas a una distancia de 100 cm.

<i>Letras</i>	<i>Total de aciertos</i>	<i>Medias de precisión</i>
A	30	98,83%
B	28	91,63%
C	30	99,53%
D	28	90,80%
E	29	95,80%
F	30	98,60%
G	30	98,20%
H	29	94,77%
I	30	98,07%
J	29	93,53%
K	30	97,73%
L	30	95,60%
M	30	98,10%
N	30	98,37%
O	30	97,93%
P	27	88,17%
Q	30	95,90%
R	16	49,57%
S	27	83,33%
T	26	83,53%
U	27	83,80%
V	27	84,83%
W	30	97,90%
X	27	83,50%
Y	30	94,70%
Z	28	89%

Realizado por: Buñay Javier, Mullo Jose, 2023.

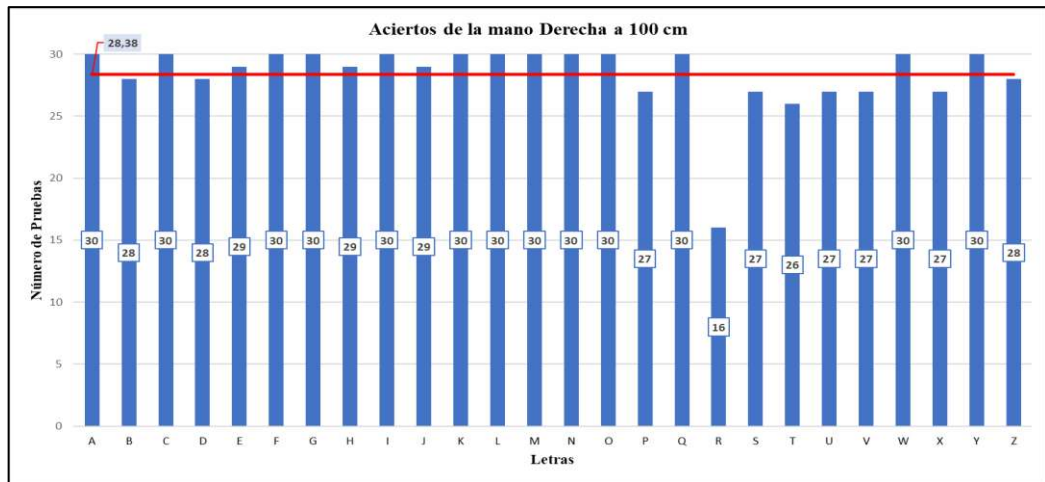


Ilustración 9-4: Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm.

Realizado por: Buñay Javier, Mullo Jose, 2023.

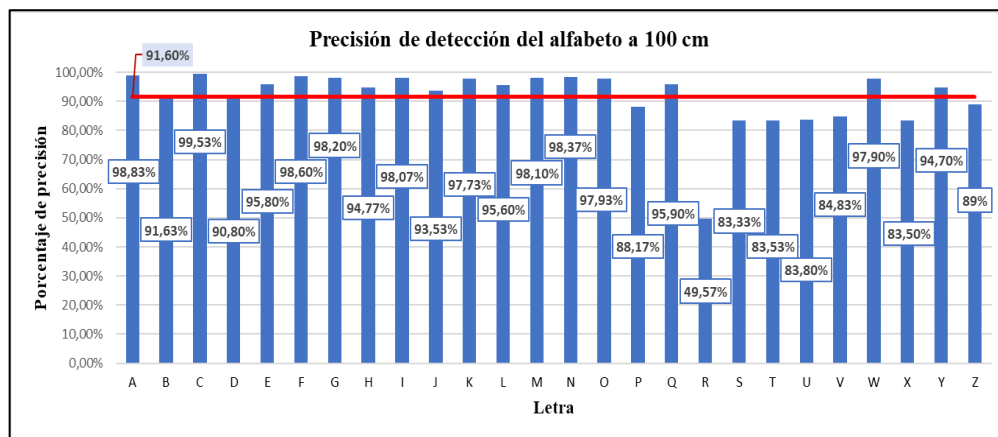


Ilustración 10-4: Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm.

Realizado por: Buñay Javier, Mullo Jose, 2023.

Por lo tanto, se concluye que los promedios totales de los porcentajes de precisión en la detección del alfabeto de la lengua de señas a través de la mano derecha, a distancias de 30 cm, 60 cm y 100 cm, superaron el 90%, lo que refleja un funcionamiento excelente.

4.3. Pruebas de aciertos y porcentaje de precisión de detección con relación a la distancia utilizando la mano Izquierda

Para llevar a cabo estas pruebas, se realizó el mismo análisis que se describió en el sub capítulo (4.2.), con la única distinción de que en este caso se empleó la mano izquierda.

A continuación, se muestra los datos resultantes tras la ejecución de las pruebas.

4.3.1. Pruebas a una distancia de 30 cm

4.3.1.1. Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)

Con base en la Tabla 4-4, se puede observar que la mayoría de las letras tuvieron una precisión que superaron el 90%, con un número de aciertos que sobrepasa los 28 de las 30 pruebas realizadas a cada una de las letras. Sin embargo, es importante señalar que la letra "R" presentó el menor número de aciertos y el porcentaje de detección más bajo. En las Ilustraciones 11-4 y 12-4 se presentan los resultados, donde la línea horizontal roja representa las medias de aciertos y precisión.

Tabla 4-4: Resultados de las pruebas a una distancia de 30 cm

<i>Letras</i>	<i>Total de aciertos</i>	<i>Medias de precisión</i>
A	28	90,50%
B	26	80,53%
C	29	95,53%
D	27	85,83%
E	28	90,83%
F	30	96,63%
G	30	99,67%
H	29	94,30%
I	30	96,87%
J	30	99,20%
K	26	85,03%
L	29	96,13%
M	30	98,57%
N	28	91,77%
O	28	92,40%
P	28	92,57%
Q	30	99%
R	23	74,97%
S	26	84,53%
T	30	98,77%
U	26	82,90%
V	27	87,27%

W	30	98,77%
X	30	99,70%
Y	30	99,07%
Z	29	96,03%

Realizado por: Buñay Javier, Mullo Jose, 2023.

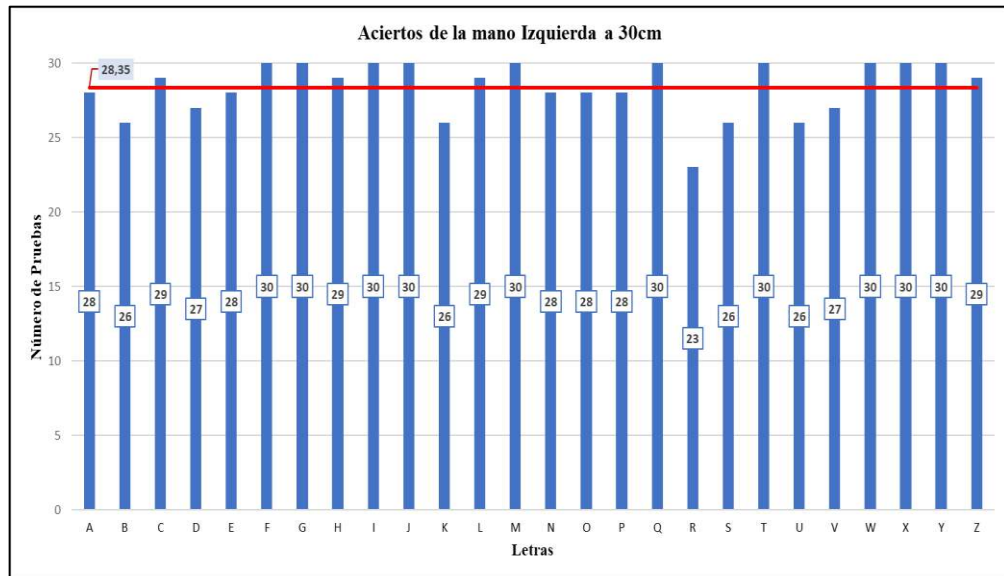


Ilustración 11-4: Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm

Realizado por: Buñay Javier, Mullo Jose, 2023.

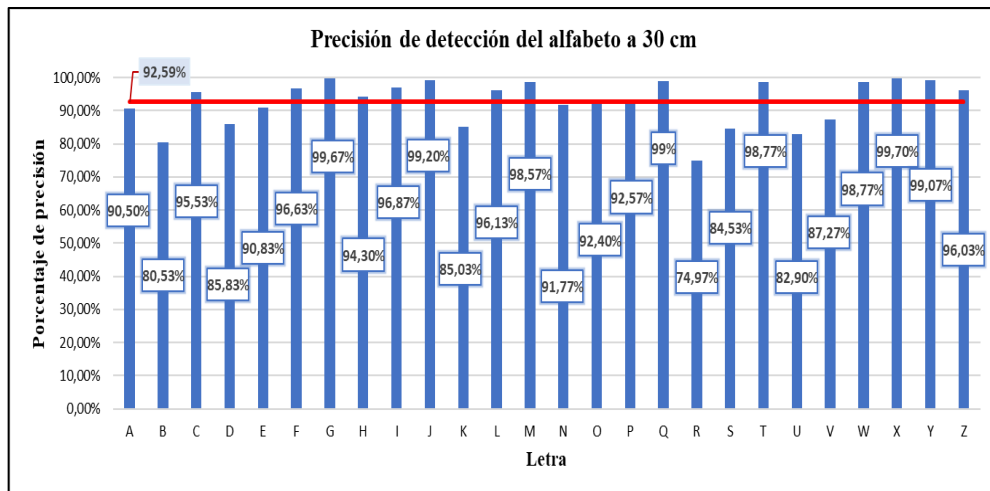


Ilustración 12-4: Porcentajes de precisión de detección del alfabeto de la Lengua de Señas a 30 cm

Realizado por: Buñay Javier, Mullo Jose, 2023.

4.3.2. Pruebas a una distancia de 60 cm

4.3.2.1. Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)

Con base en la Tabla 5-4, se puede observar que la mayoría de las letras tuvieron una precisión que supero el 90%, con un número de aciertos que sobrepasa los 28 de las 30 pruebas realizadas a cada una de las letras. Es importante señalar que la letra "R" presentó el menor número de aciertos y el porcentaje de detección más bajo. En las Ilustraciones 13-4 y 14-4 se presentan los resultados, donde la línea horizontal roja representa las medias de aciertos y precisión.

Tabla 5-4: Resultados de las pruebas a una distancia de 60 cm.

<i>Letras</i>	<i>Total, de aciertos</i>	<i>Medias de precisión</i>
A	28	92,47%
B	28	91,20%
C	29	95,40%
D	27	87,87%
E	29	95,20%
F	30	98,70%
G	30	99,77%
H	27	90,77%
I	30	97,80%
J	29	95,63%
K	29	95,63%
L	30	99,30%
M	30	97,57%
N	30	98,63%
O	29	95,27%
P	28	91,53%
Q	30	99,40%
R	19	62,07%
S	28	89,70%
T	28	89,17%
U	27	87,20%
V	28	90,63%
W	30	98,30%

X	27	84,63%
Y	30	98,57%
Z	28	91,33%

Realizado por: Buñay Javier, Mullo Jose, 2023.

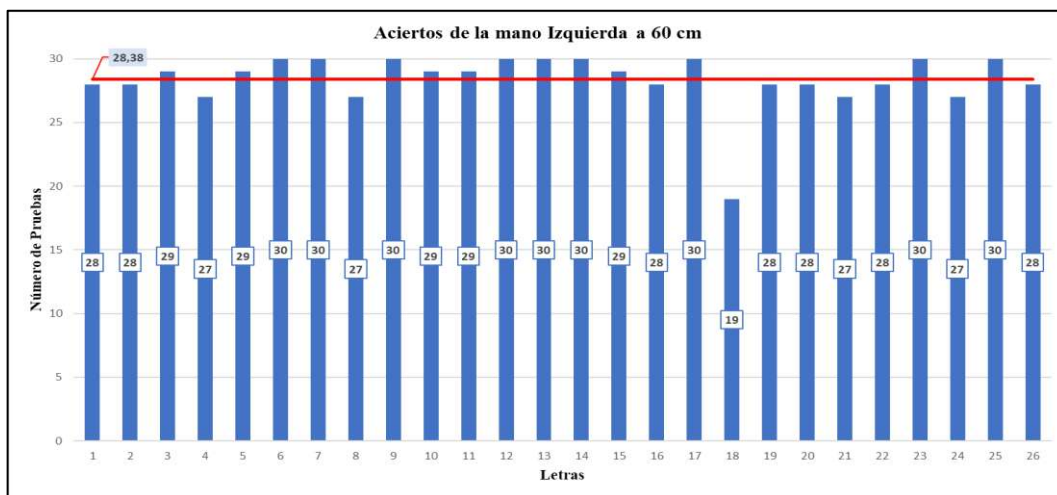


Ilustración 13-4: Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm

Realizado por: Buñay Javier, Mullo Jose, 2023.

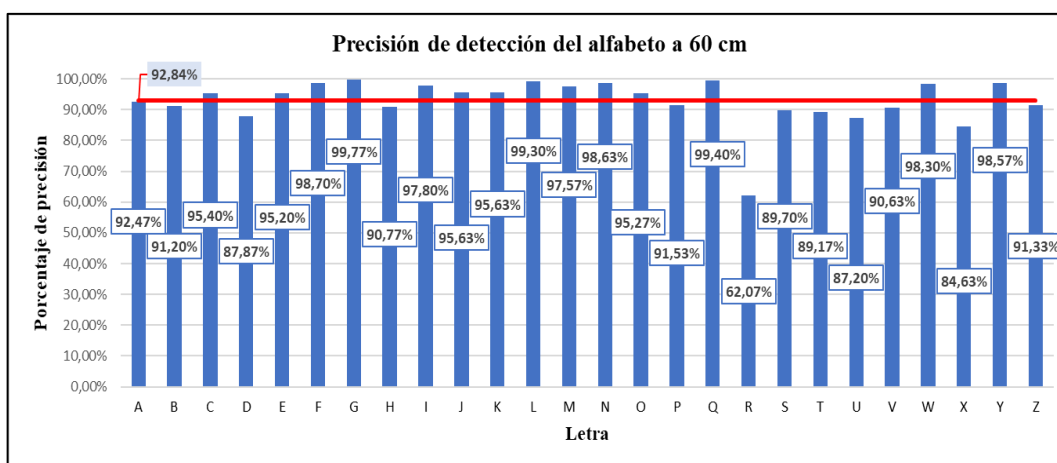


Ilustración 14-4: Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana a 60 cm

Realizado por: Buñay Javier, Mullo Jose, 2023.

4.3.3. Pruebas a una distancia de 100 cm

4.3.3.1. Pruebas del alfabeto de la Lengua de Señas Ecuatoriana (LSEC)

Con base en la Tabla 6-4, se puede observar que la mayoría de las letras tuvieron una precisión que supera el 90%, con un número de aciertos que sobrepasa los 27 de las 30 pruebas realizadas

a cada una de las letras. Sin embargo, es importante señalar que las letras "R" y "U" se destacaron al mostrar el menor número de aciertos y el porcentaje de detección más bajo. En las Ilustraciones 15-4 y 16-4 se presentan los resultados, donde la línea horizontal roja representa las medias de aciertos y precisión.

Tabla 6-4: Resultados de las pruebas a una distancia de 100 cm.

<i>Letras</i>	<i>Total de aciertos</i>	<i>Medias de precisión</i>
A	28	91,77%
B	26	81,87%
C	30	99,27%
D	27	88,40%
E	29	95,70%
F	30	98,83%
G	30	98,23%
H	5	15,83%
I	30	94,53%
J	24	77,07%
K	26	82,80%
L	30	94,93%
M	30	97,70%
N	30	98,50%
O	30	97,80%
P	27	87,57%
Q	30	96,03%
R	18	55,87%
S	26	79,37%
T	27	86,30%
U	27	84,13%
V	26	84%
W	30	97,57%
X	27	84,07%
Y	30	94,40%
Z	28	88,67%

Realizado por: Buñay Javier, Mullo Jose, 2023.

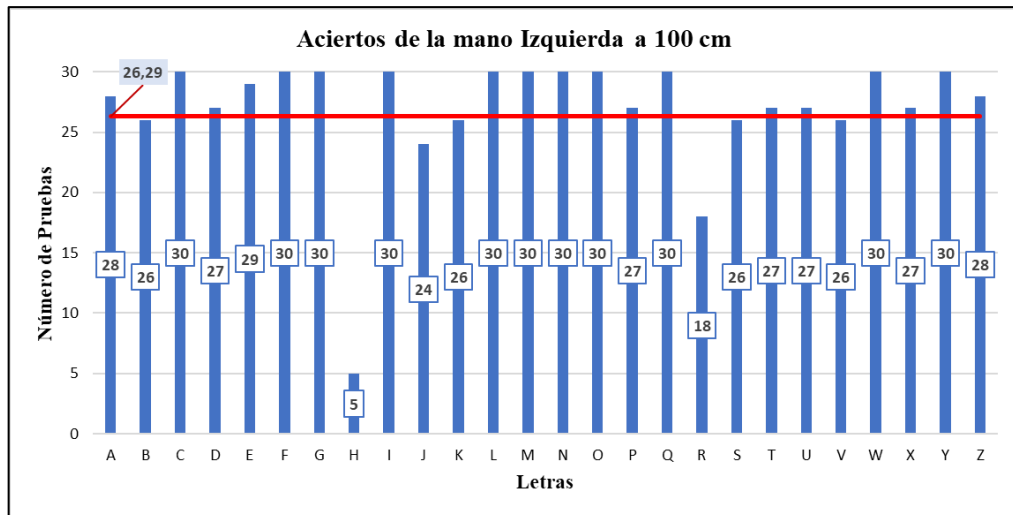


Ilustración 15-4: Aciertos del alfabeto de la Lengua de Señas Ecuatoriana a 100 cm
 Realizado por: Buñay Javier, Mullo Jose, 2023.

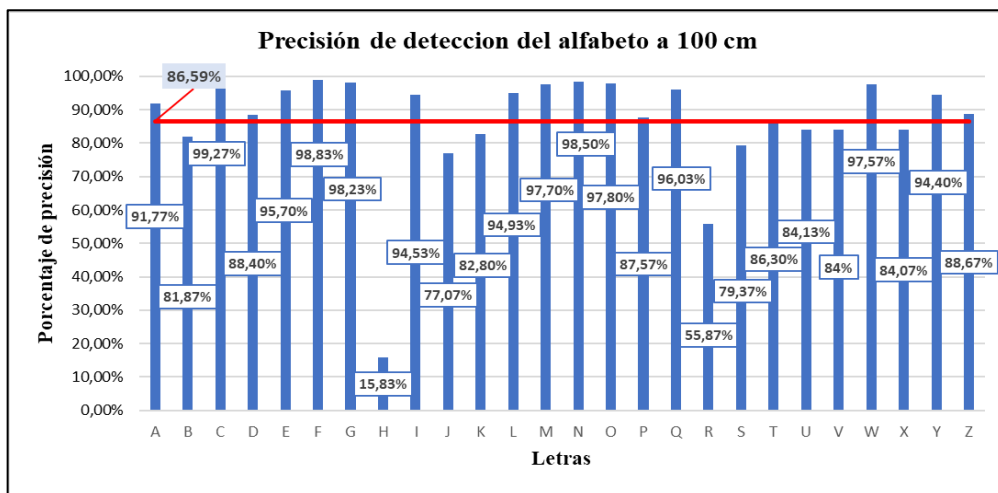


Ilustración 16-4: Porcentajes de precisión de detección del alfabeto de la Lengua de Señas Ecuatoriana.
 Realizado por: Buñay Javier, Mullo Jose, 2023.

Por lo tanto, se concluye que los promedios totales de los porcentajes de precisión en la detección del alfabeto de la lengua de señas a través de la mano izquierda, a distancias de 30 cm, 60 cm y 100 cm, superan el 90%, lo que refleja un funcionamiento excelente.

4.4. Análisis estadísticos de las pruebas

Para el análisis de los datos se utilizó, un nivel de significancia del 5% y el software estadístico IBM SPSS Statistics 25. En primera instancia se procedió a realizar la prueba de normalidad, prueba de homogeneidad de varianza e independencia de las pruebas, esta última se verifica en la

forma como se realizó la experimentación; las pruebas antes mencionadas son requisitos para realizar la ANOVA de un factor.

Se procede a realizar el análisis ANOVA (análisis de varianza) de los datos mostrados en el ANEXO B.

4.4.1.1. Pruebas de Normalidad

H₀: Las medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda siguen una distribución normal

H_a: Las medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda no siguen una distribución normal

Tabla 7-4: Pruebas de normalidad

Pruebas de normalidad								
	Distancias de prueba, 30, 60, 100	Kolmogorov-Smirnov ^a			Shapiro-Wilk			
		Estadístico	gl	Sig.	Estadístico	gl	Sig.	
Medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha	30 cm	,154	26	,114	,865	26	,003	
	60 cm	,198	26	,010	,783	26	,000	
	100 cm	,220	26	,002	,667	26	,000	
Medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano izquierda	30 cm	,167	26	,061	,886	26	,008	
	60 cm	,182	26	,027	,725	26	,000	
	100 cm	,239	26	,001	,647	26	,000	

a. Corrección de significación de Lilliefors

Realizado por: Buñay Javier, Mullo Jose, 2023.

Con base en la Tabla 7-4, según el estadístico de Kolmogorov-Smorrov, las probabilidades de 0.114 y 0.061 son mayores que el nivel de significancia, por lo tanto, no se rechaza la hipótesis nula, en la distancia de prueba de 30 cm. En los otros casos, debido a que las probabilidades son menores al nivel de significancia, se rechaza la hipótesis nula y se acepta la alternativa.

4.4.1.2. Prueba de homogeneidad y varianza.

H₀: $\sigma_{md}^2 = \sigma_{mi}^2$ (Las varianzas de las medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, son iguales).

H_a: $\sigma_{md}^2 \neq \sigma_{mi}^2$ (Las varianzas de las medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, no son iguales).

Con base en la Tabla 8-4, según el estadístico de Levene, las probabilidades de 0.273, 0.064 son mayores que el nivel de significancia, por lo tanto, no se rechaza la hipótesis nula.

Tabla 8-4: Prueba de homogeneidad de varianzas.

		Prueba de homogeneidad de varianzas			
		Estadístico de Levene	gl1	gl2	Sig.
Medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha	Se basa en la media	1,321	2	75	,273
	Se basa en la mediana	,639	2	75	,531
	Se basa en la mediana y con gl ajustado	,639	2	53,117	,532
	Se basa en la media recortada	,944	2	75	,394
Medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano izquierda	Se basa en la media	2,852	2	75	,064
	Se basa en la mediana	2,264	2	75	,111
	Se basa en la mediana y con gl ajustado	2,264	2	39,877	,117
	Se basa en la media recortada	2,332	2	75	,104

Realizado por: Buñay Javier, Mullo Jose, 2023.

4.4.1.3. ANOVA

H_0 : $\mu_1 = \mu_2 = \mu_3$ (Las medias de las medias de porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, a las distancias de 30, 60, y 100 cm, son iguales)

H_a : $\mu_i \neq \mu_j$, $i, j = 1, 2, 3$ (Hay al menos una pareja de medias que son diferentes)

Con base en la Tabla 9-4, según la prueba ANOVA, las probabilidades tanto para la mano derecha e izquierda son mayores que el nivel de significancia, por lo tanto, no se rechaza la hipótesis nula.

Tabla 9-4: Pruebas de ANOVA

		ANOVA				
		Suma de cuadrados	gl	Media cuadrática	F	Sig.
Medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha	Entre grupos	69,005	2	34,503	,566	,570
	Dentro de grupos	4573,693	75	60,983		
	Total	4642,699	77			
Medias de los porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano izquierda	Entre grupos	651,827	2	325,914	2,414	,096
	Dentro de grupos	10126,141	75	135,015		
	Total	10777,968	77			

Realizado por: Buñay Javier, Mullo Jose, 2023.

Por lo tanto, se concluye que las medias de las medias de porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, a las distancias de 30, 60, y 100 cm, son iguales.

Para realizar la prueba de hipótesis, en la que se quiere demostrar, si hay diferencias en los resultados obtenidos tanto por la mano derecha e izquierda, se utiliza una prueba de muestras independientes.

$H_0: \mu_1 = \mu_2$ (Las medias de las medias de porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, son iguales)

$H_a: \mu_1 \neq \mu_2$ (Las medias de las medias de porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, no son iguales)

Tabla 10-4: Prueba de muestras independientes

		Prueba de Levene de igualdad de varianzas		prueba t para la igualdad de medias						
		F	Sig.	t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
									Inferior	Superior
Porcentaje de dos manos	Se asumen varianzas iguales	2,304	,131	1,385	164	,168	2,21872	1,60236	-,94672	5,38415
	No se asumen varianzas iguales			1,385	132,964	,168	2,21872	1,60236	-,95069	5,38813

Realizado por: Buñay Javier, Mullo Jose, 2023.

Con base en la Tabla 10-4, según la prueba de muestras independientes, y tomando encuenta que hay igualdad de varianzas, la probabilidad de 0.168 es mayor que el nivel de significancia, por lo que no se rechaza la hipótesis nula.

Por lo tanto, Las medias de las medias de porcentajes de aciertos de precisión de detección del alfabeto de la lengua de señas a través de la mano derecha e izquierda, son iguales, esto significa que se puede utilizar cualquier mano para realizar las pruebas del alfabeto de lengua de señas.

4.5. Pruebas de tiempo de procesamiento de datos

Las pruebas correspondientes fueron realizadas por un docente de la Unidad Educativa Especializada de Sordos de Chimborazo. La persona encargada de evaluar el prototipo es de género femenino y posee un conocimiento avanzado en el manejo de la Lengua de Señas Ecuatoriana (LSEC). Se llevaron a cabo 28 pruebas en total, abarcando diferentes tipos de

contenido que incluyen vocales, consonantes, palabras de una hasta diez sílabas, así como frases completas.

El objetivo de estas pruebas es medir los tiempos de encendido del dispositivo, escritura después de detectar la letra y de reproducción en audio del mensaje ingresado. Se utiliza un dispositivo de medición para registrar estos tiempos en segundos, y los resultados se detallan en la Tabla 11-4.

Tabla 11-4: Pruebas de tiempo de reproducción.

Tiempo de Reproducción de audio del dispositivo traductor		
Numero de pruebas	Muestras	Tiempo (s)
1	A	2,7
2	E	2,7
3	I	2,7
4	O	2,7
5	U	2,7
6	F	2,7
7	No	2,7
8	¡Hola!	2,7
9	Calidad	2,7
10	Biscochito	2,7
11	Bactericida	2,7
12	Acondicionador	2,7
13	Afortunadamente	2,7
14	Inconstitucionalidad	2,7
15	Anticonstitucionalidad	2,7
16	Latinoamericanamente	2,7
17	¡Hola, buenos días! ¿Cómo estás?	2,7
18	Muy bien, ¿cómo te llamas?	2,7
19	Mi nombre es Lesly	2,7
20	Un gusto conocerte	2,7
21	Que tengas un excelente día	2,7
22	Gracias, igualmente	2,7
23	Por favor, ¿me puedes dar un vaso de agua?	2,7

24	Tengo dolor de estómago, ¿me puedes ayudar?	2,7
25	Muchas gracias por su atención	2,7
26	El dispositivo es genial	2,7
27	Nos vemos y gracias por tu ayuda	2,7
28	Los investigadores exploran lo que ya existe, mientras que los ingenieros crean lo que nunca ha sido.	2,7

Realizado por: Buñay Javier, Mullo Jose, 2023.

De acuerdo con los datos de la Tabla 11-4, el dispositivo presentó un tiempo de encendido de 1 minuto y 55 segundos, un tiempo de escritura de 1.5 segundos, y un tiempo de reproducción de audio de 2.7 segundos, ya sea para la reproducción de una letra, una palabra o para oraciones más extensas.

4.6. Pruebas de consumo energético del prototipo traductor

El propósito de esta prueba es determinar el consumo energético del prototipo cuando está en funcionamiento. Para lograrlo, se empleó una pinza amperimétrica que mide la corriente consumida por el prototipo al activar cada una de sus etapas, permitiendo así el registro de los valores que se suman para calcular el consumo total de corriente.

La evidencia correspondiente a estas pruebas se encuentra adjunta en el *ANEXO C*. Los resultados obtenidos tras efectuar las mediciones del consumo energético se presentan en la Tabla 12-4.

Tabla 12-4: Resultados de mediciones de consumo energético.

Etapa	Estado activo (Wats)	Estado teórico
Adquisición de datos	0,75	Corriente: 4A
Procesamiento	5	Voltaje: 5V
Visualización y reproducción en audio de la información	4,25	
Total	10	Potencia: 20 W

Realizado por: Buñay Javier, Mullo Jose, 2023.

Con base en la Tabla 12-4, los resultados de la prueba de consumo energético mostraron que la etapa de adquisición de datos presenta un consumo menor, representando un 3,75% del total. La

etapa de visualización y reproducción de audio consume un 21,25%, siendo la etapa de procesamiento que registra el mayor consumo, alcanzando un 25%. Esto se debió principalmente a la conexión de todos sus componentes del dispositivo, lo cual tiene un impacto significativo en el consumo total de esta etapa en comparación con el consumo teórico calculado.

Con los datos de consumo en cada elemento del prototipo traductor se calcula el consumo eléctrico con la siguiente fórmula:

Ecuación 12: Consumo eléctrico en kilovatios por hora (kWh)

$$kWh = \frac{W_{elementos} * Horas * días}{1000} \quad (12)$$

En la Tabla 13-4 se presenta el consumo eléctrico individual y total del prototipo usado durante un tiempo de 1 hora, para el cálculo se utiliza la ecuación (2).

Tabla 13-4: Consumo eléctrico individual

Parámetros	Cámara LP	Jetson Nano 4GB	Pantalla ELECROW	Parlantes
Watts (W)	0,75	4	1,8	3
Tiempo activo (Horas)	4	8	4	4
Total, de consumo mensual (kWh)	0,09	0,96	0,216	0,36
Consumo total (kWh)	1,63			

Realizado por: Buñay Javier, Mullo Jose, 2023.

Con base en la tabla 13-4, se concluye que el prototipo tiene un consumo de potencia de 1.63 vatios con un valor de 1.63 centavos de dólar por cada kilovatio-hora (¢USD/kWh).

4.7. Análisis de Costos

En la Tabla 14-4 se muestra el análisis económico dividido en etapas que detallan sus componentes, la cantidad y sus respectivos precios de la implementación del sistema. Donde se observa que el sistema implementado tiene un costo total de 600 dólares americanos (USD).

Tabla 14-4: Costos del prototipo.

Etapas	COMPONENTES	CANTIDAD	PRECIO UNITARIO (\$)	PRECIO TOTAL (\$)
Adquisición de fatos	Cámara ELP	1	70	70
Procesamiento	Jetson Nano B01 4GB	1	350	350
Visualización y reproducción en audio de la información	Pantalla ELECROW	1	70	70
	Parlantes	2	15	30
Accesorios	Varios	Varios	80	80
COSTO TOTAL DEL PROTOTIPO				600

Realizado por: Buñay Javier, Mullo Jose, 2023.

CONCLUSIONES

- Los distintos traductores de lenguaje de señas a voz se fundamentan principalmente en enfoques y tecnologías tales como sensores de mano, guantes inteligentes y visión artificial. Los traductores basados en visión artificial suelen ser versátiles y eficaces. No obstante, los guantes inteligentes enfrentan desafíos debido a la necesidad de recalibración causada por las diferencias en el tamaño de las manos de las personas, además de su elevado costo, lo que los descarta como una preferencia viable.
- Se analizaron 3 tarjetas de desarrollo como son: Raspberry Pi 4, Google Coral Dev y la Jetson Nano, la cual fue seleccionada, debido a su enfoque en aplicaciones de inteligencia artificial y su capacidad para gestionar tareas de alta complejidad gracias a su GPU dedicada. Aunque Raspberry Pi también es una opción viable para tareas más livianas de inteligencia artificial, Jetson Nano se destacó por su potencia, lo cual se evidenció en un aumento de los FPS en comparación con Raspberry Pi.
- En la etapa de diseño de la red neuronal, se llevó a cabo una evaluación de la cantidad de capas y neuronas necesarias para construir el modelo de aprendizaje. Por lo tanto, se introduce una capa inicial que consta de 42 neuronas. Estas neuronas tienen la función de representar los 21 puntos de referencia de la mano, los cuales capturan las coordenadas en los ejes “x” e “y” de cada uno de estos 21 puntos. Adicionalmente, se incluyen dos capas ocultas: una con 20 neuronas y otra con 25 neuronas. Finalmente, se incorpora una capa de salida compuesta por 26 neuronas.
- La implementación de MediaPipe en la traducción de la LSEC presentó un gran potencial. A través de su tecnología de detección de manos, se logró capturar, en la mayoría de los casos, de manera precisa las señas. La combinación de MediaPipe con un modelo de aprendizaje automático potencia la capacidad del dispositivo para reconocer y asignar gestos a sus equivalentes lingüísticos, alcanzando en ciertos casos una precisión de detección de hasta el 100%.
- El entrenamiento de la red neuronal se llevó a cabo empleando TensorFlow y Keras, dos plataformas esenciales en el desarrollo de modelos de aprendizaje automático. Este proceso engloba la preparación de datos, la construcción y compilación del modelo, el ajuste de pesos durante el proceso de entrenamiento, así como el ajuste y validación de hiperparámetros. Finalmente, el modelo es evaluado con nuevos conjuntos de datos.

- Las pruebas realizadas a distancias de 30 cm, 60 cm y 100 cm indicaron que su mejor rendimiento se alcanzó a una distancia de 30 cm, logrando un porcentaje de precisión en la detección del 93.21%. A 60 cm, se obtuvo un nivel de precisión del 93.04%, mientras que, a una distancia de 100 cm, la precisión descendió a un 89.09%. Estos resultados apuntan hacia una disminución progresiva en la precisión de detección a medida que se incrementa la distancia.
- Las letras R y U obtuvieron los resultados menos favorables debido a su marcada similitud visual, lo que dificultó la inferencia de estas dos letras por parte del prototipo.
- De acuerdo con la prueba de muestras independientes, las medias de los porcentajes de aciertos en la precisión de detección del alfabeto de la LSEC, tanto con la mano derecha como con la izquierda, son equivalentes. Esto indica que es posible emplear cualquier mano para llevar a cabo las pruebas del alfabeto de la LSEC, sin necesidad de utilizar ambas manos de manera obligatoria.

RECOMENDACIONES

- Para el correcto funcionamiento del dispositivo, es importante disponer de una iluminación que este entre 100 y 400 lux, y evitar situaciones de contraluz.
- Es importante trabajar directamente con las personas para las cuales se está desarrollando el prototipo. Esto permite obtener una comprensión profunda de sus requerimientos y así garantizar la creación de un producto final funcional y útil, alineado con las necesidades reales de los usuarios.
- Para el entrenamiento, es recomendable enfocarse en la diagonal principal de la matriz de confusión. Esta herramienta permite evaluar el desempeño de una red neuronal. Maximizar los valores en esta diagonal es importante para lograr una clasificación precisa en todas las clases, lo que a su vez contribuye a la construcción de un modelo más confiable y eficiente.
- Para el entrenamiento de una red neuronal es importante contar con un hardware potente, ya que, si no se hace, pueden surgir dificultades y el proceso puede volverse lento o incluso detenerse debido a restricciones en memoria o capacidad de cómputo.

BIBLIOGRAFÍA

ALULEMA AIMARA, Fanny Alexandra & LOZA MARTÍNEZ, Klever Alejandro. Diseño e implementación de un dispositivo electrónico mediante sistemas embebidos para la traducción del lenguaje de señas a palabras [En línea] (Trabajo de Titulación). (Posgrado) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Riobamba, Ecuador. 2017. pp. 15-18. [Consulta: 2023-04-15]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/6504>.

ARROYO RUIZ, José Marcos. *Fundamentos básicos de NVIDIA Jetson Nano*. 1ª ed. España: Marcombo, 2022. ISBN: 978-84-267-3507-2. pp. 25-50.

AYALA ABURTO, José Guillermo. Guante traductor de señas para sordomudos [En línea] (Trabajo de Titulación). (Posgrado) Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Azcapotzalco, México, 2020. pp. 18-20. [Consulta: 2023-04-15]. Disponible en: <https://tesis.ipn.mx/xmlui/handle/123456789/28152>.

BLYLER, John E. *Software-Hardware Integration in Automotive Product Development*. 3ª ed. Estados Unidos: SAE International, 2013. ISBN: 978-0-7680-8078-0. pp. 13 - 99.

BOBADILLA, Jesús. 2021. *Machine Learning y Deep Learning: Usando Python, Scikit y Keras*. 1ª ed. España: RA-MA S.A. Editorial y Publicaciones, 2021. ISBN: 978-958-792-146-5. pp. 14-27.

BRITO MONCAYO, Geovanni Danilo & GUZMÁN ARELLANO, Diego Alejandro. Guante Electrónico para traducir de lenguaje de señas a caracteres con voz artificial y conexión inalámbrica a dispositivos móviles para personas con discapacidad auditiva y de lenguaje en la Universidad Técnica de Ambato [En línea] (Trabajo de Titulación). (Posgrado). Facultad de Ingeniería en Sistemas Electrónica e Industrial, Carrera de Ingeniería en Electrónica y Comunicaciones, Ambato, Ecuador. 2017. pp. 15-30. [Consulta: 2023-04-15]. Disponible en: <https://repositorio.uta.edu.ec/handle/123456789/25193>.

BUTTU, Marco. *El gran libro de Python*. 1ª ed. España: Marcombo, 2016. ISBN: 978-84-267-2905-7. pp. 22-50.

CHIGUANO RODRÍGUEZ, Edwin Fabián & MORENO DÍAZ, Nathaly Verónica. Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado [En línea] (Trabajo de Titulación).

(Posgrado) Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica (FIEE), Quito, Ecuador. 2011. pp. 12-14. [Consulta: 2023-05-02]. Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/3915>.

CONADIS. *Estadísticas de Discapacidad*. [blog] Consejo Nacional para la Igualdad de Discapacidades Quito, Ecuador, 1 de enero, 2022. [Consulta: 12 de abril de 2023]. Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>.

DOLININA, O; et al. *Artificial Intelligence in Models, Methods and Applications*. 1ª ed. Alemania: Springer International Publishing, 2023. ISBN: 978-3-031-22938-1. pp. 299-315.

DOMÍNGUEZ MÍNGUEZ, Tomás. *Visión artificial: Aplicaciones prácticas con OpenCV - Python*. 1ª ed. España: Marcombo, 2021. ISBN: 978-84-267-3347-4. pp. 10-255.

El Telégrafo. "Clasificados". *La discapacidad auditiva afecta a 360 millones de personas en el mundo*. [En línea], 2022, (Ecuador), 16(2), pp. 15. [Consulta: 15 de abril de 2023.] Disponible en: <https://www.eltelegrafo.com.ec/noticias/sociedad/6/la-discapacidad-auditiva-afecta-a-360-millones-de-personas-en-el-mundo>.

FENASEC. 2019. *Diccionario de Lengua de Señas Ecuatoriana "Gabriel Román"*. [En línea] Federación Nacional de Personas Sordas del Ecuador, 23 de septiembre de 2019. [Consulta: 25 de abril de 2023.] Disponible en: <https://www.fenasec.ec/>.

FLÓREZ LÓPEZ, R & FERNÁNDEZ FERNÁNDEZ, J. *Las Redes Neuronales Artificiales*. 1ª ed. España: Netbiblo, 2008. ISBN: 978-84-9745-246-5. pp. 16-28.

Futuro360. 2020. Tecnología. *Crean guante que traduce lenguaje de señas en tiempo real*. [En línea] 29 de Julio de 2020, (United State of America) pp 9-10. [Citado el: 15 de abril de 2023.] https://www.futuro360.com/data/crean-guante-que-traduce-lenguaje-de-senas-en-tiempo-real_20200729/.

GEVORGYAN Menua; et al. *OpenCV 4 with Python Blueprints: Build creative computer vision projects*. 2ª ed. Reino Unido: Packt Publishing, 2020. ISBN: 978-1-78961-763-4. pp. 8-37.

GUALLO CRIOLLO, Cristian Wilfrido & DUICELA CARGUA, Soledad Elizabeth. Desarrollar un Sistema de Traducción del Lenguaje de Señas Utilizando el Sensor Kinect para las Personas Sordo Mudas [En línea] (Trabajo de Titulación). (Posgrado) Escuela Superior

Politécnica de Chimborazo, Facultad de Informática y Electrónica, Ingeniería en Sistemas Informáticos Riobamba, Ecuador. 2015. pp. 15-19. [Consulta: 2023-04-15]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/4565>.

KURNIAWAN, Agus. *Getting Started with Coral Dev Board*. 1ª ed. Estados Unidos: PE Press, 2019. pp. 5-25.

LOPÉZ ALDEA, Eugenio. *Raspberry Pi Fundamentos y Aplicaciones*. 3ª ed. España: RA-MA S.A. Editorial y Publicaciones, 2017. ISBN: 978-84-9964-701-2. pp. 66-86.

MORENO, Muffatto. *Open Source: A Multidisciplinary Approach*. 3ª ed. Singapur: Imperial College Press, 2006. ISBN: 978-1-86094-665-3. pp. 1-18.

NORRIS, Donald J. *Machine Learning with the Raspberry Pi*. 3ª ed. USA: Apress, 2019. ISBN: 978-1-4842-5174-4. pp. 1-106.

OHNA, Stein Erik. *Open Your Eyes Deaf Studies Talking*. 3ª ed. Minneapolis: Routledge, 2008. ISBN: 978-0-8166-4619-7. pp. 1-6.

OMS. 2020. *Temas de Salud*. [En línea] Organización Mundial de la Salud, 27 de febrero de 2020. [Citado el: 13 de abril de 2023.] Disponible en: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>.

PADILLA MUÑOZ, Andrea. "Discapacidad: Contexto, Concepto y Modelos". [En línea], 2010, (Bogotá) 16(2), pp. 1-20. [Consulta: 16 de abril de 2023]. ISSN 1692-8156. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-81562010000100012.

PINO DÍEZ, R; et al. *Introducción a la inteligencia artificial*. 1ª ed. Oviedo: Servicio de Publicaciones, Universidad de Oviedo, 2001. ISBN: 84-8317-249-6. pp. 2-100.

SANZ RIVAS, David & REINA VAILLO, Raúl. *Actividades físicas y deportes adaptados para personas con discapacidad*. 1ª ed. España: Paidotribo, 2021. ISBN 978-84-9910-157-6. pp. 60-89.

SCOTTS, Jason. *Raspberry Pi*. 1ª ed. Estados Unidos: Speedy Publishing LLC, 2013. ISBN: 978-1-63022-248-2. pp. 5-10.

TIETZE, Karen J. *Clinical Skills for Pharmacists*. 3^a ed. Philadelphia: Mosby, 2011. pp. 99-120.

VÁSQUEZ PARADA, Daniel Stevens & CORREDOR NIETO, William Antonio. Traductor de lenguaje de señas colombiano empleando visión por computador e inteligencia artificial [En línea] (Trabajo de Titulación). (Posgrado) Universidad Piloto de Colombia, Facultad de Ingeniería, Programa de Ingeniería Sistemas Bogotá, Colombia. 2021. pp. 17-21. [Consulta: 2023-04-15]. Disponible en: <http://repository.unipiloto.edu.co/handle/20.500.12277/11395>.

ANEXOS

ANEXO A: Datos de las pruebas de acierto y porcentajes de precisión en la detección del alfabeto de la Lengua de Señas Ecuatoriana a 30 cm, 60 cm y 100cm.

En las tablas presentadas a continuación, en la columna de distancias, los números 1, 2 y 3 representan respectivamente las distancias de 30, 60 y 100 cm.

Distancias	Aciertos_Der_A	Aciertos_Izq_A	Precicion_Der_A	Precicion_Izq_A	Aciertos_Der_B	Aciertos_Izq_B	Precicion_Der_B	Precicion_Izq_B	Aciertos_Der_C	Aciertos_Izq_C	Precicion_Der_C	Precicion_Izq_C	Aciertos_Der_D	Aciertos_Izq_D	Precicion_Der_D	Precicion_Izq_D
1	0	1	0	100	1	1	99	93	1	1	100	100	1	1	100	99
1	1	1	99	100	1	1	100	93	1	1	100	100	1	1	100	99
1	1	1	98	100	1	1	100	93	1	1	100	100	1	1	100	99
1	1	1	97	100	1	0	100	0	1	1	100	100	1	1	100	99
1	1	1	96	100	1	1	100	95	1	1	100	100	1	1	100	99
1	1	1	97	95	1	1	100	95	1	1	100	100	1	1	100	99
1	1	1	95	98	1	1	100	95	1	1	100	100	1	1	100	99
1	1	1	97	98	1	1	100	95	1	1	100	100	1	1	100	99
1	1	1	98	98	1	1	100	95	1	1	100	100	1	1	100	99
1	1	1	100	99	1	1	98	95	1	1	100	100	1	1	100	99
1	1	1	100	99	1	1	98	95	1	1	100	100	1	1	100	99
1	1	0	96	0	1	0	98	0	1	1	100	100	1	1	100	99
1	1	1	98	100	1	1	98	95	1	1	100	100	1	1	100	90
1	0	1	0	48	1	1	98	87	1	1	100	99	1	1	100	90
1	1	1	97	100	1	1	98	87	1	1	100	99	1	1	100	90
1	1	1	100	100	1	1	98	94	1	1	100	99	1	1	100	90
1	1	1	100	96	0	1	0	94	1	1	100	99	1	1	100	90
1	1	1	100	96	1	1	97	94	1	1	100	99	1	0	98	0
1	1	0	100	0	0	0	0	0	1	1	100	99	1	1	98	0
1	1	1	100	96	1	1	97	92	1	1	99	99	1	1	98	0
1	1	1	98	98	1	1	97	92	1	1	99	99	1	1	98	0
1	1	1	99	98	1	1	97	92	1	1	99	98	1	1	98	0
1	1	1	99	98	1	1	98	93	1	1	99	98	1	1	98	0
1	1	1	99	100	1	1	98	93	1	1	99	98	1	0	98	0
1	1	1	99	100	0	0	0	0	1	1	99	98	1	1	97	0
1	1	1	99	100	1	1	98	93	1	1	99	98	1	1	97	0
1	1	1	100	100	1	1	99	93	1	1	99	100	1	1	97	0
1	1	1	100	100	1	1	99	93	1	0	99	0	1	0	0	0
1	1	1	100	99	1	1	99	90	1	1	100	92	1	1	93	0
1	1	1	100	99	1	1	99	90	1	1	100	92	1	1	93	0
2	1	1	100	100	1	1	98	94	1	1	100	100	1	1	100	98
2	1	1	98	98	1	1	100	96	1	1	99	99	1	1	98	0
2	1	1	99	99	1	1	99	98	1	1	100	98	1	1	99	0
2	1	1	97	100	1	1	100	99	1	1	99	100	1	1	100	100
2	1	1	100	99	1	1	97	91	1	1	99	99	1	1	97	0
2	1	1	100	100	1	1	99	99	1	1	100	98	1	1	99	100
2	1	1	99	100	1	1	100	97	1	1	100	97	1	1	100	96
2	1	1	100	98	1	1	98	98	1	1	99	100	1	1	97	95
2	1	1	99	100	1	1	96	99	1	1	99	98	1	1	99	98
2	1	1	100	99	1	1	98	99	1	1	99	97	1	1	98	99
2	1	1	98	100	1	1	99	98	1	1	100	99	1	1	99	98
2	1	1	100	97	1	1	100	97	1	1	99	100	0	1	0	99
2	1	1	99	99	1	1	99	99	1	0	99	0	1	1	100	97
2	1	1	98	98	0	1	0	97	1	1	100	100	1	1	97	96
2	1	1	97	99	1	0	100	0	1	1	98	99	1	0	99	0
2	1	1	99	100	1	1	97	97	1	1	99	98	1	1	100	95
2	1	1	100	99	1	1	99	98	1	1	100	100	1	1	97	98
2	1	1	99	100	1	1	100	99	1	1	99	99	1	1	99	100
2	0	1	0	100	1	1	98	99	1	1	99	98	1	1	98	92
2	1	1	98	100	1	1	96	98	1	1	100	97	0	1	0	96
2	1	0	97	0	1	1	98	97	1	1	100	100	1	1	100	98
2	1	1	99	97	1	1	99	99	1	1	99	98	1	1	97	99
2	0	1	0	99	1	1	100	97	1	1	99	97	1	1	99	98
2	1	1	99	98	0	0	0	0	1	1	99	99	1	1	100	99
2	1	1	100	99	1	1	100	99	1	1	100	98	1	1	97	97
2	1	1	99	100	1	1	98	99	1	1	99	100	1	0	99	0
2	1	1	100	99	1	1	96	98	1	1	99	97	1	1	98	96
2	1	1	99	99	1	1	98	97	1	1	100	99	1	1	100	98
2	1	1	99	98	1	1	99	99	1	1	98	98	1	1	97	99
2	1	0	100	0	1	1	100	99	1	1	100	100	1	0	99	0

Aciertos_Der E	Aciertos_Izq E	Precicion_Der E	Precicion_Izq E	Aciertos_Der F	Aciertos_Izq F	Precicion_Der F	Precicion_Izq F	Aciertos_Der G	Aciertos_Izq G	Precicion_Der G	Precicion_Izq G	Aciertos_Der H	Aciertos_Izq H	Precicion_Der H	Precicion_Izq H
1	1	100	90	1	1	91	100	1	1	100	100	1	1	99	100
1	1	100	90	1	1	91	100	1	1	100	100	1	1	99	100
1	1	100	90	1	1	91	100	1	1	100	100	1	1	99	100
1	1	100	99	1	1	91	100	1	1	100	100	1	1	99	100
1	1	100	99	0	1	0	100	1	1	100	100	1	1	99	100
1	1	100	99	1	1	97	100	1	1	97	100	1	1	99	100
1	1	100	99	1	1	97	100	1	1	97	100	1	1	99	100
1	1	99	99	1	1	97	100	0	1	0	100	1	1	99	100
1	1	99	99	1	1	98	95	1	1	98	100	1	1	98	100
0	1	0	99	1	1	98	95	1	1	98	100	1	1	98	97
1	1	99	100	1	1	98	95	1	1	98	100	1	1	98	97
1	1	99	100	1	1	98	95	1	1	100	100	1	1	98	97
1	1	99	100	1	1	98	95	1	1	100	100	1	1	98	97
0	1	0	100	1	1	98	95	1	1	100	100	0	1	0	97
1	1	98	100	1	1	98	95	1	1	98	100	1	1	99	95
1	1	98	100	1	1	98	99	1	1	96	100	1	1	99	95
1	1	98	100	1	1	93	99	1	1	96	100	1	1	99	95
1	1	98	100	1	1	93	99	0	1	0	100	1	1	99	95
1	1	98	100	1	1	93	99	1	1	99	100	1	1	99	95
0	0	0	0	1	1	93	99	1	1	99	100	1	0	0	99
1	1	98	98	1	1	93	99	1	1	99	99	1	1	99	89
1	0	97	0	1	1	93	99	1	1	99	99	1	1	99	89
1	1	97	91	1	1	91	99	1	1	91	99	1	1	99	97
1	1	97	91	1	1	91	99	1	1	91	99	1	1	97	97
1	1	97	91	1	1	91	92	1	1	97	99	1	1	97	97
1	1	97	91	1	1	91	92	1	1	97	99	1	1	97	100
1	1	97	100	1	1	91	89	1	1	91	99	1	1	97	100
1	1	97	100	1	1	98	89	1	1	98	99	1	1	97	100
1	1	97	100	1	1	98	89	1	1	98	99	0	1	0	100
1	1	99	100	1	1	98	92	1	1	98	99	1	1	96	100
1	1	100	99	1	1	100	99	1	1	99	100	1	1	98	97
1	1	99	100	1	1	98	98	1	1	98	100	1	1	97	98
1	1	100	97	1	1	100	100	1	1	99	100	1	1	99	97
1	0	98	0	1	1	98	97	1	1	98	100	1	1	98	99
1	1	100	99	1	1	96	99	1	1	97	100	1	1	96	98
1	1	97	98	1	1	99	98	1	1	96	100	1	1	98	97
1	1	100	100	1	1	97	97	1	1	99	100	1	1	99	96
1	1	99	99	1	1	95	99	1	1	100	99	1	1	97	99
1	1	97	98	1	1	100	100	1	1	99	98	1	1	93	100
1	1	99	100	1	1	100	100	1	1	100	100	1	1	97	98
0	1	0	99	1	1	99	100	1	1	98	100	1	1	98	99
1	1	99	100	1	1	99	99	1	1	100	100	0	1	0	91
1	1	98	100	1	1	100	98	1	1	99	100	1	1	98	95
1	1	100	100	1	1	98	100	1	1	100	100	1	1	99	96
1	1	97	98	1	1	99	97	1	1	100	100	0	1	0	98
1	1	100	97	0	1	0	99	1	1	97	100	1	1	98	97
1	1	99	99	1	1	99	98	1	1	98	99	1	1	99	98
1	1	97	100	1	1	97	98	1	1	95	100	1	1	96	99
1	1	99	98	1	1	99	99	1	1	98	100	1	0	98	0
1	1	99	99	1	1	98	100	1	1	97	100	1	1	97	97
1	1	97	100	1	1	100	100	1	1	100	100	1	1	97	98
0	1	0	99	1	1	98	100	1	1	99	100	1	0	99	94
1	1	100	92	1	1	96	99	1	1	100	100	0	1	0	99
1	1	98	100	1	1	97	98	1	1	100	100	1	0	97	0
1	1	100	97	1	1	99	97	1	1	97	99	1	1	99	95
1	1	97	98	1	1	100	99	1	1	98	98	1	1	98	96
1	1	100	97	1	1	99	98	1	1	95	100	1	1	96	98
1	1	99	95	1	1	98	97	1	1	100	100	1	1	98	97
1	1	97	100	1	1	100	99	1	1	100	100	1	1	99	98
1	1	99	98	1	1	99	100	1	1	100	100	1	1	97	99

Aciertos_Der_I	Aciertos_Izq_I	Precision_Der_I	Precision_Izq_I	Aciertos_Der_J	Aciertos_Izq_J	Precision_Der_J	Precision_Izq_J	Aciertos_Der_K	Aciertos_Izq_K	Precision_Der_K	Precision_Izq_K	Aciertos_Der_L	Aciertos_Izq_L	Precision_Der_L	Precision_Izq_L
1	1	95	99	1	1	100	100	1	1	100	99	1	1	100	99
1	1	95	99	1	1	100	100	1	1	100	99	1	1	100	99
1	1	95	99	1	1	100	100	1	1	100	99	1	1	100	99
1	1	95	99	1	1	100	100	1	1	98	99	1	1	100	100
1	1	95	99	1	1	100	100	1	0	98	0	1	1	100	98
1	1	95	99	1	1	100	100	1	1	100	99	1	1	100	99
1	1	97	99	1	1	100	100	1	1	100	99	1	1	100	100
1	1	97	99	1	1	100	100	1	1	100	99	1	1	100	100
1	1	97	99	1	1	100	100	1	1	92	99	0	1	0	100
1	1	97	99	1	1	100	100	1	1	92	96	1	1	100	100
1	1	97	99	1	1	100	100	1	0	92	0	1	1	100	100
1	1	99	99	1	1	100	100	1	1	100	96	1	1	100	100
1	1	99	98	1	1	100	100	1	1	100	100	1	1	98	100
1	1	99	98	1	1	100	97	1	1	100	100	1	1	98	99
1	1	99	98	1	1	100	97	1	1	100	100	1	1	98	99
1	1	99	98	1	1	100	97	1	1	100	100	1	1	98	99
1	1	99	98	1	1	100	97	1	0	98	0	1	1	100	99
1	1	99	98	1	1	100	97	1	1	98	100	1	0	100	0
1	1	99	98	1	1	96	100	0	1	0	100	1	1	100	100
1	1	96	96	1	1	100	100	1	1	98	100	1	1	100	100
1	1	96	96	1	1	96	100	1	1	99	99	1	1	100	100
1	1	96	96	1	1	100	100	1	0	99	0	1	1	100	100
1	1	96	90	1	1	100	100	1	1	100	99	1	1	100	100
1	1	98	90	1	1	100	100	1	1	100	99	1	1	100	99
1	1	98	90	1	1	100	100	1	1	100	97	1	1	100	99
1	1	98	98	1	1	100	100	1	1	100	97	1	1	99	99
1	1	98	90	1	1	100	97	1	1	92	97	1	1	99	99
1	1	98	90	1	1	100	97	1	1	92	90	1	1	99	99
1	1	100	98	1	1	100	100	1	1	99	99	1	1	99	100
1	1	97	99	1	1	99	98	1	1	100	100	1	1	98	100
1	1	99	97	1	1	99	99	1	1	98	98	1	1	99	100
1	1	98	98	1	1	98	100	1	1	99	100	1	1	96	99
1	1	100	99	1	1	100	99	1	1	97	99	1	1	100	99
1	1	99	99	1	1	99	100	1	1	100	99	1	1	100	100
1	1	98	99	1	1	97	98	1	1	98	98	1	1	99	98
1	1	98	98	1	1	99	97	1	1	99	100	1	1	98	100
1	1	97	96	1	0	100	0	1	1	100	99	1	1	100	98
1	1	99	97	1	1	97	100	1	1	99	100	1	1	97	99
1	1	99	96	1	1	96	98	1	1	99	98	1	1	99	100
1	1	95	100	1	1	98	99	1	1	100	99	1	1	100	100
1	1	98	98	1	1	100	100	1	1	99	100	1	1	99	100
1	1	98	98	1	1	98	97	1	1	98	99	1	1	98	99
1	1	97	99	1	1	98	100	0	1	0	98	0	1	0	99
1	1	96	96	1	1	99	98	1	1	100	97	1	1	99	100
1	1	99	97	1	1	99	99	1	0	97	0	1	1	98	98
1	1	98	96	1	1	97	100	1	1	98	99	1	1	99	100
1	1	98	97	1	1	98	100	1	1	100	99	1	1	96	98
1	1	97	96	1	1	96	98	1	1	100	98	1	1	100	99
1	1	99	100	1	1	98	99	1	1	99	100	1	1	100	100
1	1	99	98	1	1	98	97	1	1	98	99	1	1	99	100
1	1	95	98	1	1	100	99	1	1	98	100	1	1	98	100
1	1	97	99	1	1	98	100	1	1	100	98	1	1	100	99
1	1	99	96	1	1	98	98	1	1	97	99	1	1	97	99
1	1	99	97	1	1	99	99	1	1	98	100	1	1	99	100
1	1	95	99	1	1	99	100	1	1	100	99	1	1	100	98
1	1	98	97	1	1	97	99	1	1	100	98	1	1	99	100
1	1	98	98	1	1	99	100	1	1	99	97	1	1	98	98
1	1	97	99	1	1	97	98	1	1	98	100	1	1	100	99

Aciertos_Der_M	Aciertos_Izg_M	Precicion_Der_M	Precicion_Izg_M	Aciertos_Der_N	Aciertos_Izg_N	Precicion_Der_N	Precicion_Izg_N	Aciertos_Der_O	Aciertos_Izg_O	Precicion_Der_O	Precicion_Izg_O	Aciertos_Der_P	Aciertos_Izg_P	Precicion_Der_P	Precicion_Izg_P
1	1	100	100	1	1	100	97	1	1	100	99	1	1	100	100
1	1	100	100	1	1	100	99	1	1	100	99	1	1	100	100
1	1	100	100	1	1	100	99	1	1	100	100	1	1	100	95
1	1	100	100	1	1	100	100	1	1	100	100	0	1	1	0
1	1	100	100	1	1	100	100	1	1	100	100	1	1	99	100
1	1	97	100	1	1	100	98	1	1	100	100	1	0	99	0
1	1	97	100	1	0	100	0	1	1	100	100	1	1	99	100
1	1	97	97	1	1	97	100	1	1	99	93	1	1	99	99
1	1	99	100	1	1	97	100	1	1	99	93	1	1	100	99
1	1	99	100	1	1	97	91	1	1	99	98	1	1	100	99
1	1	99	100	1	1	100	97	1	1	99	98	1	0	100	0
1	1	99	97	1	1	100	97	1	1	99	100	1	1	100	98
1	1	99	100	1	1	100	97	1	1	100	100	1	1	100	98
1	1	98	100	1	1	98	97	1	1	100	100	1	1	100	98
1	1	98	100	1	1	99	99	1	1	100	100	1	1	100	98
1	1	98	100	1	0	100	0	1	0	99	0	1	1	100	100
1	1	100	99	1	1	100	100	1	1	99	99	1	1	100	100
1	1	99	99	1	1	100	100	1	1	99	99	1	1	99	100
1	1	100	87	1	1	100	96	1	1	98	99	1	1	99	100
1	1	100	97	0	1	0	99	1	1	100	99	1	1	99	99
1	1	100	87	1	1	99	98	1	0	100	0	1	1	99	99
1	1	100	100	1	1	99	98	1	1	100	100	1	1	99	99
1	1	100	97	1	1	99	98	1	1	99	100	1	1	99	99
1	1	100	100	1	1	99	98	1	1	99	100	1	1	100	99
1	1	100	100	1	1	100	95	1	1	97	100	1	1	100	100
1	1	100	100	1	1	100	100	1	1	100	100	1	1	100	100
1	1	99	100	1	1	97	100	1	1	100	98	1	1	98	100
1	1	99	99	1	1	100	100	1	1	99	98	1	1	99	99
1	1	99	99	1	1	100	100	1	1	99	100	1	1	100	99
1	1	99	99	1	1	100	100	1	1	99	100	1	1	100	100
1	1	99	99	1	1	100	99	1	1	100	99	1	1	100	99
1	1	98	98	1	1	99	98	1	1	97	100	1	1	97	98
1	1	99	99	1	1	100	97	1	1	98	99	1	1	95	100
1	1	100	99	1	1	98	100	1	1	100	99	1	1	93	99
1	1	100	100	1	1	100	100	1	1	99	100	0	1	0	100
1	1	99	97	1	1	100	98	1	1	98	99	1	1	98	97
1	1	98	98	1	1	98	96	1	1	100	98	1	1	100	98
1	1	97	99	1	1	99	98	1	1	99	100	1	1	99	100
1	1	96	96	1	1	98	97	1	0	98	0	0	0	0	0
1	1	99	97	1	1	100	99	1	1	97	96	1	1	99	100
1	1	98	98	0	1	0	98	1	1	99	98	1	1	100	98
1	1	100	99	1	1	100	99	1	1	98	97	1	1	98	97
1	1	100	99	1	1	100	100	1	1	98	99	0	1	0	99
1	1	98	100	1	1	98	100	1	1	99	96	1	1	100	98
1	1	98	99	1	1	99	99	1	1	99	97	1	1	97	100
1	1	99	99	1	1	98	97	1	1	100	99	0	1	0	89
1	1	97	100	1	1	100	98	1	1	100	98	1	0	100	0
1	1	96	96	1	1	99	99	1	1	99	94	1	1	100	97
1	1	98	97	1	1	98	99	1	1	100	100	0	1	94	100
1	1	97	95	1	1	100	97	1	1	99	100	1	1	98	99
1	1	96	94	1	1	99	98	1	1	98	99	1	1	99	97
1	1	99	95	0	1	0	99	1	1	100	98	1	1	100	90
1	1	98	100	1	1	100	99	1	1	99	96	1	1	98	98
1	1	100	99	1	1	100	99	1	1	98	99	1	1	97	97
1	1	100	94	1	1	98	98	1	1	97	100	1	1	99	100
1	1	98	97	1	1	99	100	1	1	99	99	0	1	0	99
1	1	98	95	1	1	98	98	1	1	98	99	1	1	100	98
1	1	99	98	1	1	100	100	1	1	98	100	1	1	98	100
1	1	97	97	1	1	99	99	1	1	99	100	1	1	97	99
1	1	96	96	1	1	98	98	1	1	99	100	1	1	99	100

Aciertos_Der_Q	Aciertos_Izq_Q	Precicion_Der_Q	Precicion_Izq_Q	Aciertos_Der_R	Aciertos_Izq_R	Precicion_Der_R	Precicion_Izq_R	Aciertos_Der_S	Aciertos_Izq_S	Precicion_Der_S	Precicion_Izq_S	Aciertos_Der_T	Aciertos_Izq_T	Precicion_Der_T	Precicion_Izq_T
1	1	99	100	1	1	96	95	1	1	98	98	1	1	100	100
1	1	99	100	1	1	96	97	1	1	98	93	1	1	100	98
1	1	100	100	0	1	0	99	1	1	100	96	1	1	100	99
1	1	99	99	1	1	98	99	1	1	100	100	1	1	100	99
1	1	100	99	0	0	98	0	1	1	97	100	1	1	100	100
1	1	99	100	1	1	100	90	1	1	97	99	1	1	100	100
1	1	100	100	0	1	0	90	1	1	99	95	1	1	100	100
1	1	100	100	1	1	100	99	1	1	99	95	1	1	97	100
1	1	100	99	1	0	100	0	1	1	100	100	1	1	99	99
1	1	100	99	0	1	0	100	1	1	96	100	1	1	99	99
1	1	99	99	1	1	98	100	1	0	100	0	1	1	99	99
1	1	99	100	1	1	98	97	0	1	0	98	1	1	99	99
1	1	99	100	1	0	98	0	1	1	96	98	1	1	100	100
1	1	99	100	1	1	99	95	1	0	96	0	1	1	100	90
1	1	99	98	1	1	99	98	1	1	100	99	1	1	100	100
1	1	100	98	0	1	0	97	1	1	100	99	1	1	100	100
1	1	100	99	1	1	98	100	1	1	100	100	1	1	100	100
1	1	100	98	1	0	98	0	1	1	100	100	1	1	90	100
1	1	100	95	1	1	98	97	1	1	99	97	1	1	98	98
1	1	100	99	0	0	0	0	0	0	0	0	1	1	98	98
1	1	99	99	1	1	99	99	1	1	99	94	1	1	98	98
1	1	99	99	1	1	99	99	1	1	98	94	1	1	99	99
1	1	98	97	1	1	100	100	1	1	100	92	1	1	99	99
1	1	99	97	0	0	0	0	1	1	100	96	1	1	99	99
1	1	100	100	1	1	100	100	1	1	100	96	1	1	99	97
1	1	98	100	1	1	100	100	0	1	0	97	1	1	100	98
1	1	98	99	1	1	98	100	1	0	100	0	1	1	100	95
1	1	99	100	0	1	0	99	1	1	100	100	1	1	100	100
1	1	99	99	1	1	97	99	1	1	100	100	1	1	100	100
1	1	99	98	1	0	96	0	1	1	100	100	1	1	100	100
1	1	100	100	0	1	0	95	1	1	91	97	1	1	100	95
1	1	98	100	1	1	96	97	1	1	97	95	1	1	100	97
1	1	100	100	0	1	0	99	1	1	98	99	1	1	100	93
1	1	100	99	1	1	98	99	1	1	90	98	1	1	100	97
1	1	99	100	0	0	98	0	1	1	97	97	1	1	100	98
1	1	100	100	1	1	100	90	1	1	96	94	1	1	100	99
1	1	100	98	0	0	0	0	1	1	98	92	1	1	99	92
1	1	100	100	1	1	100	99	1	1	94	98	1	1	100	97
1	1	97	100	1	0	100	0	1	1	96	91	1	1	98	98
1	1	100	99	0	1	0	100	0	1	0	96	1	1	100	96
1	1	99	98	1	1	98	100	1	1	98	95	1	1	97	89
1	1	99	99	1	1	98	97	1	1	99	94	1	1	99	94
1	1	99	100	0	0	0	0	1	1	97	97	1	1	98	97
1	1	99	100	1	1	99	95	1	0	95	0	1	1	97	99
1	1	98	100	1	0	99	0	1	1	98	97	1	1	100	98
1	1	98	99	0	1	0	97	1	1	97	95	0	1	0	92
1	1	98	100	1	1	98	100	1	1	98	99	1	1	100	89
1	1	98	100	1	0	98	0	0	1	0	98	1	1	100	93
1	1	99	98	1	1	98	97	1	1	91	97	1	0	100	0
1	1	99	100	0	0	0	0	1	1	97	94	1	1	100	95
1	1	99	100	1	1	99	99	1	1	98	92	1	1	99	97
1	1	99	99	1	0	99	0	1	1	90	98	1	1	100	93
1	1	99	98	1	1	100	100	1	1	97	97	1	1	98	97
1	1	97	99	0	0	0	0	1	1	96	96	1	1	100	98
1	1	97	100	1	1	100	100	1	0	98	0	1	1	97	99
1	1	97	100	1	1	100	100	1	1	94	99	1	1	99	92
1	1	99	99	1	0	98	0	1	1	96	100	1	1	98	97
1	1	99	98	0	1	0	99	1	1	98	97	1	1	97	98
1	1	99	99	1	1	97	99	1	1	98	94	1	1	100	96
1	1	100	100	1	0	96	0	1	1	97	95	1	0	98	0

Aciertos_Der_U =	Aciertos_Izq_U =	Precicion_Der_U =	Precicion_Izq_U =	Aciertos_Der_V =	Aciertos_Izq_V =	Precicion_Der_V =	Precicion_Izq_V =	Aciertos_Der_W =	Aciertos_Izq_W =	Precicion_Der_W =	Precicion_Izq_W =	Aciertos_Der_X =	Aciertos_Izq_X =	Precicion_Der_X =	Precicion_Izq_X =
1	1	99	97	1	1	98	95	1	1	100	100	1	1	100	100
1	1	97	97	1	1	98	98	1	1	100	100	1	1	100	100
1	1	94	98	0	1	0	98	1	1	100	99	0	1	0	100
1	1	94	94	1	1	90	97	1	1	100	99	1	1	100	100
1	0	97	0	1	0	98	0	1	1	100	99	1	1	99	100
1	1	90	98	1	1	97	97	1	1	100	98	1	1	100	100
1	1	92	99	1	1	95	95	1	1	100	98	1	1	100	100
1	1	99	95	1	1	93	96	1	1	98	98	1	1	100	100
0	1	92	0	98	1	1	98	1	1	98	96	1	1	100	100
1	1	98	94	1	1	95	98	1	1	98	96	1	1	95	99
1	1	98	90	1	1	98	98	1	1	98	96	1	1	100	100
1	1	94	92	1	0	90	0	1	1	99	96	1	1	100	100
1	0	95	0	0	1	0	98	1	1	99	100	1	1	100	100
1	1	98	98	1	0	98	0	1	1	99	100	1	1	94	99
1	1	98	98	1	1	98	98	1	1	99	100	1	1	100	100
1	1	94	97	1	1	96	94	1	1	99	100	1	1	100	100
1	1	100	99	1	1	98	97	1	1	100	100	1	1	99	97
1	1	98	97	0	1	0	95	1	1	100	100	1	1	100	100
1	0	94	0	1	1	96	97	1	1	100	98	1	1	100	100
0	1	0	92	1	1	96	95	1	1	100	98	1	1	97	100
1	1	97	93	1	1	98	99	1	1	100	98	1	1	97	98
1	1	96	95	1	1	98	98	1	1	99	100	1	1	100	100
1	0	95	0	1	1	98	95	1	1	98	100	1	1	98	100
0	1	0	98	1	1	97	96	1	1	97	100	1	1	100	100
1	1	94	94	1	1	95	97	1	1	97	99	1	1	100	100
1	1	96	95	1	1	87	98	1	1	97	99	1	1	98	99
1	1	98	95	1	1	98	99	1	1	97	99	1	1	100	100
1	1	93	90	1	1	97	96	1	1	97	99	1	1	100	100
0	1	84	96	1	1	97	98	1	1	99	99	1	1	99	99
1	1	99	100	1	1	98	98	1	1	99	99	1	1	100	100
1	1	99	98	1	1	100	98	1	1	98	100	1	1	100	99
1	1	98	99	1	1	97	96	1	1	100	99	1	1	98	100
1	1	96	97	1	1	99	97	1	1	99	97	1	1	89	92
0	1	0	94	1	1	98	98	1	1	97	100	1	1	87	93
1	1	98	96	1	1	100	97	1	1	100	96	1	1	97	94
1	0	97	0	1	1	98	95	1	1	96	97	1	1	96	96
1	1	96	97	1	1	97	98	1	1	97	98	1	1	98	98
0	1	0	95	1	1	100	97	1	1	98	100	0	1	0	87
1	1	98	96	1	1	99	95	1	1	100	100	1	1	99	89
1	1	99	98	1	1	96	98	1	1	96	96	1	0	98	0
1	1	96	99	0	1	0	99	1	1	99	97	1	1	89	94
1	1	94	97	1	1	100	97	1	1	97	100	1	1	87	96
1	0	95	0	1	1	97	96	1	1	100	99	1	1	97	98
1	1	94	96	1	0	99	0	1	1	96	97	1	1	96	87
1	1	99	95	1	1	98	98	1	1	97	100	1	1	98	89
1	1	98	98	1	1	100	96	1	1	100	97	1	1	96	99
1	1	95	96	1	1	98	97	1	1	99	100	1	1	94	100
1	1	96	97	1	1	97	98	1	1	97	96	1	0	97	0
1	1	98	98	1	1	100	97	1	1	100	97	1	1	89	96
1	0	99	0	1	1	99	95	0	1	96	98	0	1	0	98
1	1	100	96	1	1	96	98	1	1	97	100	1	1	87	96
1	1	98	95	1	0	99	0	1	1	98	99	1	1	92	94
1	1	100	98	1	1	98	95	1	1	100	100	1	1	95	97
1	1	92	96	1	1	100	98	1	1	96	99	1	1	95	87
1	1	93	97	1	1	99	99	1	1	97	97	1	1	94	89
1	1	95	98	1	1	97	97	1	1	98	100	1	1	89	99
1	1	97	99	1	1	96	96	1	1	100	97	1	1	87	99
1	1	98	96	1	1	100	98	1	1	96	100	1	0	86	0
0	1	0	97	1	1	99	99	1	1	99	96	1	1	98	87
1	1	98	98	1	1	98	97	1	1	97	97	1	1	98	86

Aciertos_Der_Y	Aciertos_Izq_Y	Precicion_Der_Y	Precicion_Izq_Y	Aciertos_Der_Z	Aciertos_Izq_Z	Precicion_Der_Z	Precicion_Izq_Z
1	1	100	100	1	1	100	100
1	1	100	100	1	1	100	100
1	1	99	98	1	1	99	100
1	1	99	98	1	1	99	98
1	1	100	98	1	1	99	98
1	1	98	100	1	1	100	98
1	1	100	100	1	1	100	98
1	1	100	100	1	1	100	98
1	1	99	99	1	1	98	100
1	1	100	99	1	1	100	100
1	1	100	100	1	1	97	100
1	1	100	100	1	1	100	100
1	1	98	100	1	1	100	97
1	1	100	97	1	1	100	100
1	1	98	100	1	1	100	100
1	1	98	100	1	1	98	100
1	1	100	98	1	1	98	99
1	1	99	100	1	1	100	100
1	1	100	100	1	1	100	100
1	1	100	98	0	1	0	100
1	1	100	100	1	1	100	100
1	1	100	100	1	1	100	99
1	1	100	97	1	1	98	99
1	1	97	100	1	1	96	99
1	1	100	94	1	1	100	100
1	1	98	100	1	1	100	100
1	1	100	96	0	1	0	100
1	1	99	100	1	0	100	0
1	1	100	100	1	1	98	100
1	1	100	100	1	1	100	98
1	1	100	98	1	1	100	99
1	1	99	99	1	1	100	98
1	1	97	100	1	1	99	100
1	1	98	100	1	0	98	0
1	1	100	99	1	1	99	98
1	1	100	97	1	1	100	97
1	1	99	98	1	1	99	96
1	1	100	100	1	1	100	99
1	1	98	100	1	1	99	98
1	1	100	99	1	1	98	99
1	1	100	100	1	1	99	96
1	1	100	99	1	1	100	96
1	1	99	100	1	1	99	95
1	1	100	98	1	1	100	98
1	1	98	99	1	1	98	97
1	1	100	99	1	1	99	100
1	1	100	95	1	1	100	100
1	1	97	98	1	1	99	100
1	1	98	99	1	1	100	99
1	1	100	100	1	1	99	96
1	1	99	97	1	1	98	95
1	1	99	97	1	1	98	98
1	1	98	98	1	1	99	97
1	1	98	98	1	1	98	95
1	1	99	97	1	1	98	98
1	1	98	98	1	1	99	97

Distancias	Aciertos_Der_A	Aciertos_Izq_A	Precision_Der_A	Precision_Izq_A	Aciertos_Der_B	Aciertos_Izq_B	Precision_Der_B	Precision_Izq_B	Aciertos_Der_C	Aciertos_Izq_C	Precision_Der_C	Precision_Izq_C	Aciertos_Der_D	Aciertos_Izq_D	Precision_Der_D	Precision_Izq_D
3	1	1	100	100	1	1	99	99	1	1	100	99	1	1	97	98
3	1	1	99	100	1	1	98	98	1	1	100	99	1	1	98	100
3	1	1	98	100	1	1	97	97	1	1	98	100	1	1	99	97
3	1	1	100	99	1	1	99	96	1	1	100	99	1	1	97	100
3	1	1	100	98	1	1	98	94	1	1	100	98	1	1	100	98
3	1	1	100	98	1	1	99	95	1	1	99	100	1	1	96	99
3	1	1	99	99	1	0	96	0	1	1	100	100	1	1	95	98
3	1	1	98	96	1	1	100	92	1	1	100	100	1	0	94	0
3	1	1	98	97	1	1	99	90	1	1	100	98	1	1	97	98
3	1	1	99	95	1	1	97	85	1	1	98	99	1	1	96	100
3	1	0	96	0	1	1	98	96	1	1	100	100	1	1	99	97
3	1	1	98	98	1	1	97	95	1	1	100	100	1	1	98	100
3	1	1	100	99	1	1	96	92	1	1	100	99	1	1	99	98
3	1	1	99	97	0	1	0	94	1	1	99	98	1	1	97	99
3	1	1	98	100	1	1	97	90	1	1	100	100	1	1	99	96
3	1	1	99	99	1	1	99	91	1	1	100	100	1	0	96	0
3	1	1	100	97	1	0	98	0	1	1	98	100	1	1	100	100
3	1	0	100	0	1	1	99	97	1	1	100	98	1	1	94	98
3	1	1	100	98	1	1	100	95	1	1	100	100	1	1	98	99
3	1	1	100	99	0	1	0	97	1	1	100	100	1	1	96	97
3	1	1	99	97	1	0	96	0	1	1	98	98	1	0	99	0
3	1	1	98	100	1	1	100	99	1	1	100	99	0	1	0	98
3	1	1	98	99	1	1	99	94	1	1	100	100	1	1	99	99
3	1	1	99	97	1	1	97	92	1	1	99	98	1	1	96	97
3	1	1	96	98	1	1	98	99	1	1	100	99	1	1	100	96
3	1	1	98	99	1	0	97	0	1	1	100	100	1	1	94	98
3	1	1	100	96	1	1	98	93	1	1	99	100	1	1	98	99
3	1	1	99	98	1	1	99	96	1	1	100	99	1	1	96	97
3	1	1	98	100	1	1	100	98	1	1	98	98	1	1	97	99
3	1	1	99	100	1	1	99	92	1	1	100	100	0	1	0	97

Aciertos_Der_E	Aciertos_Izq_E	Precision_Der_E	Precision_Izq_E	Aciertos_Der_F	Aciertos_Izq_F	Precision_Der_F	Precision_Izq_F	Aciertos_Der_G	Aciertos_Izq_G	Precision_Der_G	Precision_Izq_G	Aciertos_Der_H	Aciertos_Izq_H	Precision_Der_H	Precision_Izq_H
1	1	100	99	1	1	98	99	1	1	100	96	1	1	99	90
1	1	100	100	1	1	98	97	1	1	98	97	1	1	98	97
1	1	99	98	1	1	100	98	1	1	99	99	1	1	98	90
1	1	98	97	1	1	99	99	1	1	97	95	1	1	99	98
1	1	97	100	1	1	99	99	1	1	100	97	1	1	99	100
1	1	100	99	1	1	100	97	1	1	100	100	1	0	98	0
1	1	99	100	1	1	98	98	1	1	99	99	1	0	97	0
1	1	100	97	1	1	97	99	1	1	98	98	1	0	99	0
1	1	100	100	1	1	98	99	1	1	97	100	1	0	98	0
1	1	99	99	1	1	97	100	1	1	96	99	1	0	96	0
1	1	99	100	1	1	98	100	1	1	97	98	1	0	97	0
1	1	100	100	1	1	99	100	1	1	99	97	1	0	99	0
1	1	98	99	1	1	99	99	1	1	95	96	1	0	98	0
1	1	97	98	1	1	100	99	1	1	97	100	1	0	98	0
1	1	100	97	1	1	100	100	1	1	100	99	1	0	99	0
1	1	99	100	1	1	99	98	1	1	99	98	1	0	97	0
1	1	100	99	1	1	99	97	1	1	98	97	0	0	0	0
1	1	97	100	1	1	100	99	1	1	97	96	1	0	97	0
1	0	100	0	1	1	98	99	1	1	96	100	1	0	99	0
1	1	99	97	1	1	97	100	1	1	100	98	1	0	98	0
1	1	100	100	1	1	98	100	1	1	98	99	1	0	96	0
1	1	100	99	1	1	97	100	1	1	99	100	1	0	97	0
0	1	0	100	1	1	98	99	1	1	97	100	1	0	99	0
1	1	99	97	1	1	99	99	1	1	100	99	1	0	98	0
1	1	100	100	1	1	99	100	1	1	100	98	1	0	98	0
1	1	98	99	1	1	97	99	1	1	99	97	1	0	99	0
1	1	100	100	1	1	98	99	1	1	98	96	1	0	99	0
1	1	99	100	1	1	99	97	1	1	97	100	1	0	98	0
1	1	100	99	1	1	99	98	1	1	96	100	1	0	97	0
1	1	97	98	1	1	100	99	1	1	100	99	1	0	99	0

Aciertos_Der I	Aciertos_Izq I	Precicion_Der I	Precicion_Izq I	Aciertos_Der J	Aciertos_Izq J	Precicion_Der J	Precicion_Izq J	Aciertos_Der K	Aciertos_Izq K	Precicion_Der K	Precicion_Izq K	Aciertos_Der L	Aciertos_Izq L	Precicion_Der L	Precicion_Izq L
1	1	99	99	1	1	99	97	1	1	98	98	1	1	100	99
1	1	98	98	1	1	98	96	1	1	99	97	1	1	99	96
1	1	98	95	1	1	97	98	1	1	97	96	1	1	96	98
1	1	97	97	1	0	98	0	1	0	98	0	1	1	98	99
1	1	99	98	1	1	96	97	1	1	97	97	1	1	99	100
1	1	99	95	1	1	94	93	1	1	97	96	1	1	100	99
1	1	100	89	1	1	93	95	1	1	96	94	1	1	99	94
1	1	99	87	1	1	95	96	1	1	99	96	1	1	94	93
1	1	98	90	1	1	96	97	1	1	100	95	1	1	93	96
1	1	95	99	1	0	97	0	1	1	99	97	1	1	96	98
1	1	97	98	1	1	96	98	1	1	98	99	1	1	98	89
1	1	98	95	1	1	98	97	1	1	96	96	1	1	89	94
1	1	96	97	1	1	99	98	1	0	97	0	1	1	87	93
1	1	100	98	1	1	97	96	1	1	98	96	1	1	90	96
1	1	96	95	1	0	93	0	1	1	97	95	1	1	97	98
1	1	100	89	1	1	95	93	1	1	97	97	1	1	96	89
1	1	100	87	1	1	96	95	1	1	96	99	1	1	95	87
1	1	99	90	1	1	97	96	1	1	99	98	1	1	98	90
1	1	98	99	1	0	96	0	1	0	100	0	1	1	99	97
1	1	99	98	1	1	98	99	1	1	99	96	1	1	100	98
1	1	98	95	1	1	99	97	1	1	98	94	1	1	99	89
1	1	95	97	1	1	99	93	1	1	96	96	1	1	94	87
1	1	97	98	0	1	0	95	1	1	95	95	1	1	93	90
1	1	98	98	1	0	95	0	1	1	97	97	1	1	96	97
1	1	96	95	1	1	96	97	1	1	97	99	1	1	98	96
1	1	100	89	1	1	96	96	1	1	96	95	1	1	87	95
1	1	96	87	1	1	97	98	1	0	99	0	1	1	90	98
1	1	100	90	1	0	99	0	1	1	100	87	1	1	97	100
1	1	99	95	1	1	98	98	1	1	99	89	1	1	96	99
1	1	98	99	1	1	99	97	1	1	98	90	1	1	95	94

Aciertos_Der M	Aciertos_Izq M	Precicion_Der M	Precicion_Izq M	Aciertos_Der N	Aciertos_Izq N	Precicion_Der N	Precicion_Izq N	Aciertos_Der O	Aciertos_Izq O	Precicion_Der O	Precicion_Izq O	Aciertos_Der P	Aciertos_Izq P	Precicion_Der P	Precicion_Izq P
1	1	98	96	1	1	98	98	1	1	100	100	1	1	98	97
1	1	97	99	1	1	99	100	1	1	98	98	1	1	97	96
1	1	96	98	1	1	97	98	1	1	99	97	1	1	96	94
1	1	99	100	1	1	100	98	1	1	100	96	1	1	94	98
1	1	98	100	1	1	100	99	1	1	98	97	0	1	0	100
1	1	100	98	1	1	98	97	1	1	97	98	1	1	98	99
1	1	100	98	1	1	98	96	1	1	96	99	1	1	100	96
1	1	98	99	1	1	99	98	1	1	97	100	1	1	99	94
1	1	98	96	1	1	97	97	1	1	98	96	0	1	0	98
1	1	99	98	1	1	96	100	1	1	99	97	1	0	99	0
1	1	97	97	1	1	98	100	1	1	100	98	1	1	100	99
1	1	96	96	1	1	97	98	1	1	98	98	1	1	98	98
1	1	98	98	1	1	100	98	1	1	97	100	0	1	0	100
1	1	97	100	98	1	100	98	1	1	96	98	1	1	100	99
1	1	96	99	1	1	98	98	1	1	97	97	1	1	97	96
1	1	99	97	1	1	98	98	1	1	98	96	1	1	97	94
1	1	98	96	1	1	99	99	1	1	97	97	1	1	99	99
1	1	100	98	1	1	97	97	1	1	96	96	1	1	100	96
1	1	100	99	1	1	96	96	1	1	97	97	1	1	94	94
1	1	98	96	1	1	100	100	1	1	98	98	1	1	98	98
1	1	98	98	1	1	100	100	1	1	99	99	1	0	99	0
1	1	99	97	1	1	98	98	1	1	100	100	1	1	100	99
1	1	96	96	1	1	100	98	1	1	98	96	1	1	98	98
1	1	98	98	1	1	98	100	1	1	97	97	1	1	97	100
1	1	97	98	1	1	98	98	1	1	96	98	1	1	97	100
1	1	96	99	1	1	99	100	1	1	97	99	1	1	99	99
1	1	99	97	1	1	97	100	1	1	98	100	1	1	100	96
1	1	98	96	1	1	96	98	1	1	99	98	1	1	94	94
1	1	100	98	1	1	100	98	1	1	100	97	1	0	98	0
1	1	100	98	1	1	100	100	1	1	98	96	1	1	99	96

Aciertos_Der_Q	Aciertos_Izq_Q	Precicion_Der_Q	Precicion_Izq_Q	Aciertos_Der_R	Aciertos_Izq_R	Precicion_Der_R	Precicion_Izq_R	Aciertos_Der_S	Aciertos_Izq_S	Precicion_Der_S	Precicion_Izq_S	Aciertos_Der_T	Aciertos_Izq_T	Precicion_Der_T	Precicion_Izq_T
1	1	99	99	1	1	96	96	1	1	92	93	1	1	99	98
1	1	98	97	1	1	92	92	1	1	94	92	1	1	98	94
1	1	99	96	0	1	0	96	1	1	92	94	1	1	97	96
1	1	97	93	1	1	94	94	1	1	95	90	1	1	99	94
1	1	96	98	1	1	95	95	1	1	93	89	1	1	96	98
1	1	93	95	0	0	0	0	1	0	95	0	0	1	0	93
1	1	98	92	1	1	94	94	1	1	92	88	1	1	97	97
1	1	95	99	1	1	89	89	1	1	95	87	1	1	98	93
1	1	92	97	0	0	0	0	0	1	0	90	1	1	94	94
1	1	93	96	0	0	0	0	1	1	94	93	1	0	96	0
1	1	94	98	1	1	90	90	1	1	92	90	1	1	94	96
1	1	98	97	1	1	92	92	1	1	94	99	1	1	98	94
1	1	99	96	0	1	0	96	1	1	92	92	1	1	93	98
1	1	93	93	1	1	94	94	1	1	95	95	1	1	97	93
1	1	96	98	0	0	0	0	1	0	93	0	1	1	95	97
1	1	95	95	1	1	97	97	1	1	90	88	0	0	0	0
1	1	97	92	0	0	0	0	1	1	87	89	1	1	97	99
1	1	96	95	1	1	93	93	1	1	89	87	1	1	99	96
1	1	92	92	1	1	94	94	1	1	92	92	1	1	96	99
1	1	98	93	0	0	0	0	1	1	95	96	1	1	99	97
1	1	95	94	1	0	98	0	0	0	0	0	1	1	97	98
1	1	92	98	0	0	0	0	1	1	94	90	1	1	98	96
1	1	93	99	0	1	0	95	1	1	92	93	1	1	94	94
1	1	94	97	1	1	90	90	1	1	94	90	0	1	0	98
1	1	98	96	1	1	90	90	1	1	92	87	1	1	94	93
1	1	97	98	0	0	0	0	1	1	95	92	1	1	96	96
1	1	96	94	0	0	0	0	1	0	93	0	1	1	94	94
1	1	93	98	1	1	89	89	0	1	0	93	1	0	98	0
1	1	98	99	0	0	0	0	1	1	89	94	1	1	93	98
1	1	95	97	0	0	0	0	1	1	90	98	0	1	0	96

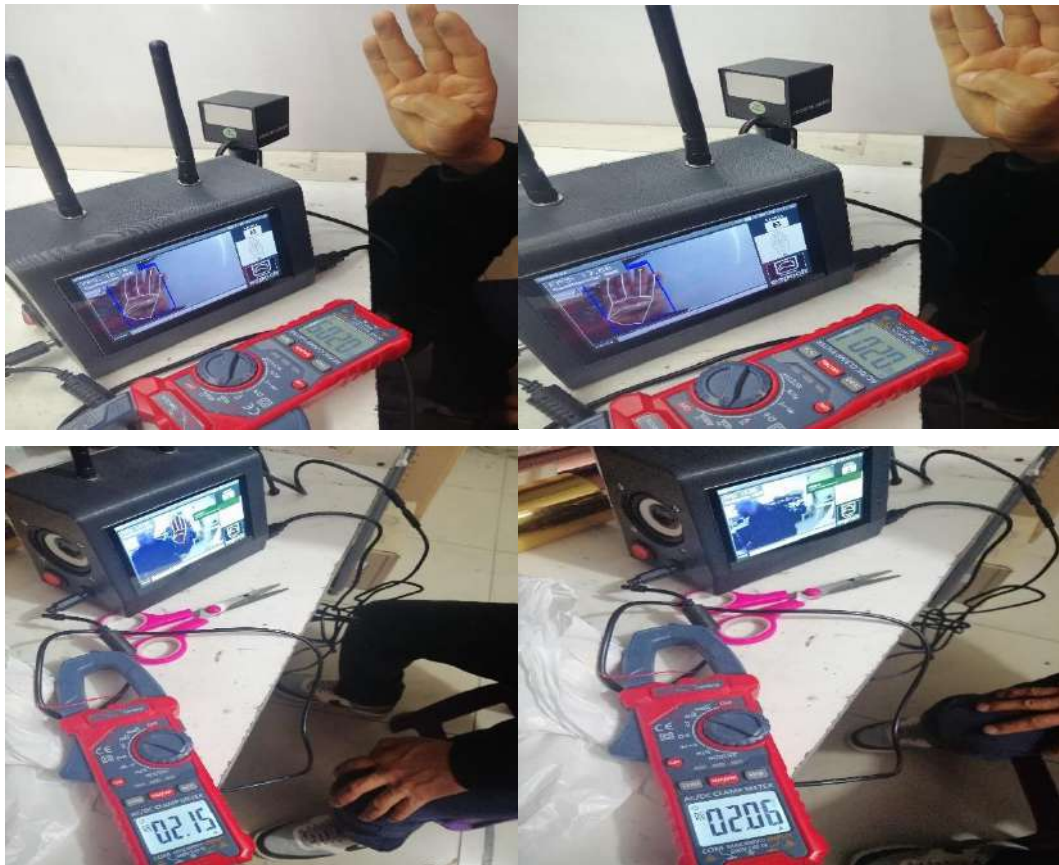
Aciertos_Der_U	Aciertos_Izq_U	Precicion_Der_U	Precicion_Izq_U	Aciertos_Der_V	Aciertos_Izq_V	Precicion_Der_V	Precicion_Izq_V	Aciertos_Der_W	Aciertos_Izq_W	Precicion_Der_W	Precicion_Izq_W	Aciertos_Der_X	Aciertos_Izq_X	Precicion_Der_X	Precicion_Izq_X
1	1	97	97	1	1	96	97	1	98	100	98	1	1	98	98
1	1	94	95	1	1	98	95	1	1	99	99	1	1	94	94
1	1	93	89	0	1	0	98	1	1	98	95	1	1	93	93
1	1	92	88	1	1	93	97	1	1	99	94	1	1	96	95
0	1	0	89	1	1	94	95	1	1	95	97	0	1	0	94
1	1	95	94	1	0	95	0	1	1	94	98	1	1	97	93
1	1	94	93	1	1	96	98	1	1	97	99	1	1	95	91
1	1	97	92	1	1	98	96	1	1	98	100	1	1	94	89
1	0	95	0	1	1	96	97	1	1	99	96	1	1	93	87
1	1	89	95	0	1	0	98	1	1	100	100	1	0	91	0
1	1	88	89	1	0	93	0	1	1	96	98	1	1	89	95
1	1	89	94	1	1	94	95	1	1	100	99	1	1	87	94
1	1	94	95	1	1	95	98	1	1	99	100	1	1	96	93
1	1	95	94	1	1	96	98	1	1	97	96	0	1	0	91
0	0	0	0	1	1	93	96	1	1	99	95	1	1	95	89
1	1	94	89	1	1	92	98	1	1	95	94	1	1	94	98
1	1	97	94	1	0	91	0	1	1	98	97	1	1	93	94
1	1	95	95	1	1	93	95	1	1	98	98	1	0	91	0
1	1	89	97	1	1	94	98	1	1	100	100	1	1	89	95
1	1	94	95	0	1	0	98	1	1	97	96	1	1	94	94
1	1	89	89	1	1	94	96	1	1	98	100	1	1	93	93
1	1	94	94	1	1	95	97	1	1	99	98	1	1	96	91
1	1	95	95	1	1	96	98	1	1	100	99	1	1	91	89
1	1	89	97	1	1	93	95	1	1	96	100	1	1	90	98
1	1	94	95	1	1	92	98	1	1	100	96	0	1	0	94
1	1	89	97	1	1	93	98	1	1	99	94	1	0	89	0
1	1	94	95	1	0	94	0	1	1	97	97	1	1	90	98
0	0	0	0	1	1	95	98	1	1	100	98	1	1	92	94
1	1	95	95	1	1	96	97	1	1	99	100	1	1	89	93
1	1	94	93	1	1	90	96	1	1	97	96	1	1	96	95

Aciertos_Der_Y	Aciertos_Izq_Y	Precicion_Der_Y	Precicion_Izq_Y	Aciertos_Der_Z	Aciertos_Izq_Z	Precicion_Der_Z	Precicion_Izq_Z
1	1	96	94	1	1	93	95
1	1	98	96	1	1	95	98
1	1	96	95	1	1	98	97
1	1	97	92	1	1	97	99
1	1	94	92	1	0	99	0
1	1	98	93	1	1	100	93
1	1	94	91	1	1	87	96
1	1	96	95	1	1	93	93
1	1	95	98	1	1	96	96
1	1	92	94	1	1	93	98
1	1	93	96	1	1	96	97
1	1	91	95	0	1	0	93
1	1	95	92	1	1	97	96
1	1	96	93	1	1	98	93
1	1	97	94	1	1	97	95
1	1	94	96	1	1	92	92
1	1	98	95	1	0	99	0
1	1	94	92	1	1	98	97
1	1	96	92	1	1	91	93
1	1	95	93	1	1	93	96
1	1	92	96	1	1	96	93
1	1	92	97	1	1	93	93
1	1	93	94	1	1	92	96
1	1	91	98	1	1	99	93
1	1	95	94	1	1	98	97
1	1	96	96	1	1	91	96
1	1	94	95	0	1	0	93
1	1	96	92	1	1	92	93
1	1	95	98	1	1	99	96
1	1	92	94	1	1	98	93

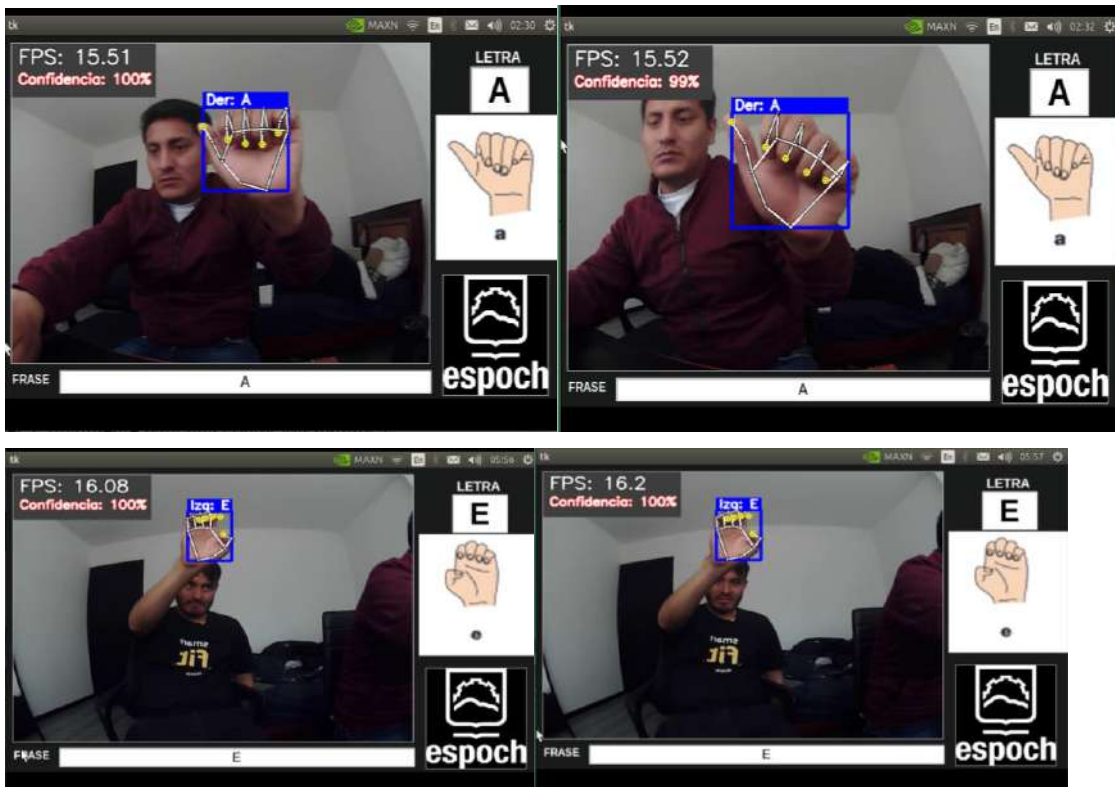
ANEXO B: Datos estadísticos en el software IBM SPSS.

DISTANCIAS	PORCENTAJE_MANO_DERECHA	PORCENTAJE_MANO_IZQUIERDA
30cm	92,07	90,5
30cm	88,77	80,53
30cm	99,7	95,53
30cm	95,43	85,83
30cm	88,6	90,83
30cm	91,53	96,63
30cm	91,17	99,67
30cm	91,8	94,3
30cm	97,17	96,87
30cm	99,6	99,2
30cm	94,63	85,03
30cm	96,27	96,13
30cm	99,17	98,57
30cm	96,03	91,77
30cm	99,4	92,4
30cm	96,23	92,57
30cm	99,3	99
30cm	75,43	74,97
30cm	89,07	84,53
30cm	99,1	98,77
30cm	85,97	82,9
30cm	86,3	87,27
30cm	98,9	98,77
30cm	95,87	99,7
30cm	99,4	99,07
30cm	92,67	96,03
60cm	92,4	92,47
60cm	92,03	91,2
60cm	99,33	95,4
60cm	92,07	87,87
60cm	92,17	95,2
60cm	95,23	98,7
60cm	98,53	99,77
60cm	87,9	90,77
60cm	97,87	97,8
60cm	98,33	95,63
60cm	95,57	95,63
60cm	95,47	99,3
60cm	98,23	97,57
60cm	92,5	98,63
60cm	98,73	95,27
60cm	81,83	91,53
60cm	98,83	99,4
60cm	68,97	62,07
60cm	89,63	89,7
60cm	95,8	89,17
60cm	87,2	87,2
60cm	95,13	90,63
60cm	98	98,3
60cm	87,53	84,63
60cm	99,1	98,57
60cm	95,7	91,33
100cm	98,83	91,77
100cm	91,63	81,87
100cm	99,53	99,27
100cm	90,8	88,4
100cm	95,8	95,7
100cm	98,6	98,83
100cm	98,2	98,23
100cm	94,77	15,83
100cm	98,07	94,53
100cm	93,53	77,07
100cm	97,73	82,8
100cm	95,6	94,93
100cm	98,1	97,7
100cm	98,37	98,5
100cm	97,93	97,8
100cm	88,17	87,57
100cm	95,9	96,03
100cm	49,57	55,87
100cm	83,33	79,37
100cm	83,53	86,3
100cm	83,8	84,13
100cm	84,83	84
100cm	97,9	97,57
100cm	83,5	84,07
100cm	94,7	94,4
100cm	89	88,67

ANEXO C: Mediciones del consumo energético.



ANEXO D: Pruebas de funcionamiento del entrenamiento de la Red Neuronal.



ANEXO E: Solicitud de acceso a la Unidad Educativa Especializada Sordos de Chimborazo



Dirección de
Desarrollo Académico
Espoch

Trámite # 6564-E
(Signature)

Riobamba, 2 de agosto de 2023

Magister

Hugo Patricio Chávez Chávez

DIRECTOR DISTRITAL 06D01 CHAMBO-RIOBAMBA

Presente



De mi consideración:

Reciba un cordial saludo, a la vez solicito comedidamente autorice el acceso al Instituto de Sordos de Chimborazo para realizar las pruebas del prototipo desarrollado por los Sres. JAVIER MOISES BUÑAY CUJILEMA con C.I 060480094-6 y JOSE EMILIO MULLO YAUTIBUG con C.I 060534044-7, que se encuentran realizando el trabajo de titulación denominado "CONSTRUCCIÓN DE UN PROTOTIPO ELECTRÓNICO TRADUCTOR DE LENGUAJE DE SEÑAS A VOZ CON BASE EN PROCESAMIENTO DE IMÁGENES" para titularse de Ingeniero en Electrónica y Automatización. Dicho proyecto tiene como finalidad que las personas que pertenecen a este grupo vulnerable puedan interactuar de mejor manera con su entorno.

Agradezco de antemano su atención y esperando una respuesta favorable a mi petición, quedo a sus órdenes.

Atentamente,

(Signature)
Ing. José Luis Morales
C.I 180321954-0
Coordinador de carrera
Ingeniería en Electrónica y Automatización

*Entregado
8/10/2023
RSP*

ANEXO F: Autorización de acceso a la Unidad Educativa Especializada Sordos de Chimborazo



GUILLERMO LASSO
PRESIDENTE

Oficio Nro. MINEDUC-CZ3-06D01-2023-5802-O

Riobamba, 08 de agosto de 2023

Asunto: COMPROMISO CIUDADANO OFICIO S/N DE FECHA 2 DE AGOSTO DEL 2023 EL ING. JOSÉ LUIS MORALES COORDINADOR DE CARRERA DE INGENIERIA ELECTRÓNICA Y AUTOMATIZACIÓN ESPOCH SOLICITA AUTORIZACIÓN PARA INGRESAR A LA U.E. SORDOS DE CHIMBORAZO A LOS SEÑORES JAVIER

Señor
Jose Luis Morales Gordon
En su Despacho

De mi consideración:

En respuesta al documento MINEDUC-CZ3-06D01-UDAC-2023-6564-E y oficio S/N de fecha 02 de agosto de 2023 el Ing. José Luis Morales COORDINADOR DE CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN ESPOCH SOLICITA AUTORIZACIÓN PARA INGRESAR A LA U.E. SORDOS DE CHIMBORAZO.

En atención a lo solicitado mediante oficio S/N de fecha 2 de agosto del 2023 firmado por el Ing. José Luis Morales Coordinador de Carrera de Ingeniería Electrónica. El Distrito Educativo 06D01-Chambo Riobamba **AUTORIZA** el acceso a la institución educativa a los señores Javier Moisés Buñay Cujilema con C.I 0604800946 y Mullo Yautibug C.I 0605340447, para que realicen las prácticas de titulación y puedan interactuar de mejor manera con su entorno siendo un grupo vulnerable.

Con sentimientos de distinguida consideración.

Atentamente,

Sr. Hugo Patricio Chavez Chavez
DIRECTOR DISTRITAL 06D01 - CHAMBO-RIOBAMBA - EDUCACIÓN

Oficio Nro. MINEDUC-CZ3-06D01-2023-5802-O

Riobamba, 08 de agosto de 2023

Referencias:

- MINEDUC-CZ3-06D01-UDAC-2023-6564-E

Anexos:

- jose_luis_morales_espoch_acceso_ue_sordos_de_chimborazo20230802_13352137.pdf

ip/im

ANEXO G: Recibido por parte de la Unidad Educativa Especializada Sordos de Chimborazo



Dirección de
Desarrollo Académico
Espech
Molongoche

Riobamba, 9 de agosto de 2023

Lic. Susana Romero Z.

RECTORA DE LA "UNIDAD EDUCATIVA ESPECIALIZADA SORDOS DE CHIMBORAZO"

Presente

De nuestra consideración:

Reciba un cordial saludo. Nos dirigimos a usted de manera conjunta para solicitar comedidamente su autorización para acceder a la Unidad Educativa Especializada Sordos de Chimborazo, con el propósito de realizar pruebas del prototipo desarrollado por nosotros, los Sres. JAVIER MOISÉS BUÑAY CUJILEMA, con C.I 060480094-6, y JOSÉ EMILIO MULLO YAUTIBUG, con C.I 060534044-7. Como parte de nuestro proceso de titulación en la carrera de Ingeniería en Electrónica y Automatización, hemos elaborado un proyecto titulado "CONSTRUCCIÓN DE UN PROTOTIPO ELECTRÓNICO TRADUCTOR DE LENGUAJE DE SEÑAS A VOZ CON BASE EN PROCESAMIENTO DE IMÁGENES". Dicho proyecto tiene como finalidad que las personas que pertenecen a este grupo vulnerable puedan interactuar de mejor manera con su entorno.

Agradecemos sinceramente su atención a esta solicitud y esperamos con optimismo una respuesta favorable a nuestra petición. Quedamos a sus órdenes para proporcionar cualquier información adicional que pueda requerir.

Atentamente,

JAVIER MOISÉS BUÑAY CUJILEMA

C.I 060480094-6

JOSÉ EMILIO MULLO YAUTIBUG

C.I 060534044-7



recibido
09.08.2023

ANEXO H: Cédula de identidad del docente que evaluó el prototipo.



ANEXO I: Pruebas realizadas al prototipo por parte del docente de la Unidad Educativa Especializada Sordos de Chimborazo.



ANEXO J: Código de programación.

- a) Programa principal.


```

1  #!/usr/bin/env python
2  import time
3
4  #import matplotlib
5  #matplotlib.use('Agg')
6  import cv2
7  import csv
8  import copy
9  import itertools
10 from collections import Counter
11 from collections import deque
12 import cv2 as cv
13 import numpy as np
14 import mediapipe as mp
15 from utils import CvFpsCalc
16 from model import KeyPointClassifier
17 from gtts import gTTS
18
19 from tkinter import *
20 import tkinter as tk
21 from PIL import Image, ImageTk
22 #import pygame
23 from playsound import playsound
24 # import pygame
25
26 # pygame.init()
27 # pygame.mixer.init()
28
29 global confidencia, limite, contborrar, contletras
30 confidencia = 0.00

```

```

31 limite = 0.75
32 contborrar = 0
33 contletras = 0
34
35 cntCamara = 0
36 siCamaraConectada = False
37
38 # Model load #####
39 mp_hands = mp.solutions.hands
40 hands = mp_hands.Hands(
41     static_image_mode=False,
42     max_num_hands=1,
43     min_detection_confidence=0.9,
44     min_tracking_confidence=0.9,
45 )
46
47 keypoint_classifier = KeyPointClassifier()
48
49 # Read labels #####
50 with open(
51     '/home/nano/Desktop/TESIS/proyecto/SIGNOS_VIS
52     encoding='utf-8-sig') as f:
53     keypoint_classifier_labels = csv.reader(f)
54     keypoint_classifier_labels = [
55         row[0] for row in keypoint_classifier_labels
56     ]
57
58 # FPS Measurement #####
59 cvFpsCalc = CvFpsCalc(buffer_len=10)
60 # Coordinate history #####
61 history_length = 16

```

```

62 point_history = deque(maxlen=history_length)
63 # Finger gesture history #####
64 finger_gesture_history = deque(maxlen=history_length)
65 # #####
66 mode = 0
67 use_brect = True
68 cntLetra = 0
69 cntPalabra = 0
70 letraActual = ""
71 letraSiguiente = ""
72 palabra = ""
73
74
75 def texto_a_voz(texto, nombre_archivo):
76
77     tts = gTTS(text=texto, lang='es')
78     tts.save(nombre_archivo)
79     playsound(nombre_archivo)
80
81
82 def calc_bounding_rect(image, landmarks):
83     image_width, image_height = image.shape[1], image.shape[0]
84     landmark_array = np.empty((0, 2), int)
85
86     for _, landmark in enumerate(landmarks.landmark):
87         landmark_x = min(int(landmark.x * image_width), image_width - 1)
88         landmark_y = min(int(landmark.y * image_height), image_height - 1)
89         landmark_point = [np.array((landmark_x, landmark_y))]
90         landmark_array = np.append(landmark_array, landmark_point, axis=0)
91     x, y, w, h = cv.boundingRect(landmark_array)
92

```

```

93     return [x, y, x+w, y+h]
94
95
96 def calc_landmark_list(image, landmarks):
97     image_width, image_height = image.shape[1], image.shape[0]
98     landmark_point = []
99     # Keypoint
100    for _, landmark in enumerate(landmarks.landmark):
101        landmark_x = min(int(landmark.x * image_width), image_width - 1)
102        landmark_y = min(int(landmark.y * image_height), image_height - 1)
103        # landmark_z = landmark.z
104        landmark_point.append([landmark_x, landmark_y])
105    return landmark_point
106
107
108 def pre_process_landmark(landmark_list):
109     temp_landmark_list = copy.deepcopy(landmark_list)
110     # Convert to relative coordinates
111     base_x, base_y = 0, 0
112     for index, landmark_point in enumerate(temp_landmark_list):
113         if index == 0:
114             base_x, base_y = landmark_point[0], landmark_point[1]
115             temp_landmark_list[index][0] = temp_landmark_list[index][0] - base_x
116             temp_landmark_list[index][1] = temp_landmark_list[index][1] - base_y
117     # Convert to a one-dimensional list
118     temp_landmark_list = list(
119         itertools.chain.from_iterable(temp_landmark_list))
120     # Normalization
121     max_value = max(list(map(abs, temp_landmark_list)))
122     def normalize(n):
123         return n / max_value

```

```

124     temp_landmark_list = list(map(normalize_, temp_landmark_list))
125     return temp_landmark_list
126
127
128 def pre_process_point_history(image, point_history):
129     image_width, image_height = image.shape[1], image.shape[0]
130     temp_point_history = copy.deepcopy(point_history)
131     # Convert to relative coordinates
132     base_x, base_y = 0, 0
133     for index, point in enumerate(temp_point_history):
134         if index == 0:
135             base_x, base_y = point[0], point[1]
136             temp_point_history[index][0] = (temp_point_history[index][0] - base_x) / image_width
137             temp_point_history[index][1] = (temp_point_history[index][1] - base_y) / image_height
138
139     # Convert to a one-dimensional list
140     temp_point_history = list(
141         itertools.chain.from_iterable(temp_point_history))
142     return temp_point_history
143
144
145 def draw_info(image, fps, mode):
146     cv.rectangle(image, (0, 0), (220, 40), (62, 62, 62), -1)
147     cv.putText(image, "FPS: " + str(fps), (10, 30), cv.FONT_HERSHEY_SIMPLEX, 1.0, (0, 0, 0), 4, cv
148     cv.putText(image, "FPS: " + str(fps), (10, 30), cv.FONT_HERSHEY_SIMPLEX, 1.0, (255, 255, 255),
149
150     return image
151
152
153 def draw_landmarks(image, landmark_point):
154     if len(landmark_point) > 0:

```

```

155         # Thumb
156         cv.line(image, tuple(landmark_point[2]), tuple(landmark_point[3]),
157                (0, 0, 0), 4)
158         cv.line(image, tuple(landmark_point[2]), tuple(landmark_point[3]),
159                (255, 255, 255), 2)
160         cv.line(image, tuple(landmark_point[3]), tuple(landmark_point[4]),
161                (0, 0, 0), 4)
162         cv.line(image, tuple(landmark_point[3]), tuple(landmark_point[4]),
163                (255, 255, 255), 2)
164
165         # Index finger
166         cv.line(image, tuple(landmark_point[5]), tuple(landmark_point[6]),
167                (0, 0, 0), 4)
168         cv.line(image, tuple(landmark_point[5]), tuple(landmark_point[6]),
169                (255, 255, 255), 2)
170         cv.line(image, tuple(landmark_point[6]), tuple(landmark_point[7]),
171                (0, 0, 0), 4)
172         cv.line(image, tuple(landmark_point[6]), tuple(landmark_point[7]),
173                (255, 255, 255), 2)
174         cv.line(image, tuple(landmark_point[7]), tuple(landmark_point[8]),
175                (0, 0, 0), 4)
176         cv.line(image, tuple(landmark_point[7]), tuple(landmark_point[8]),
177                (255, 255, 255), 2)
178
179         # Middle finger
180         cv.line(image, tuple(landmark_point[9]), tuple(landmark_point[10]),
181                (0, 0, 0), 4)
182         cv.line(image, tuple(landmark_point[9]), tuple(landmark_point[10]),
183                (255, 255, 255), 2)
184         cv.line(image, tuple(landmark_point[10]), tuple(landmark_point[11]),
185                (0, 0, 0), 4)

```

```

186         cv.line(image, tuple(landmark_point[10]), tuple(landmark_point[11]),
187                (255, 255, 255), 2)
188         cv.line(image, tuple(landmark_point[11]), tuple(landmark_point[12]),
189                (0, 0, 0), 4)
190         cv.line(image, tuple(landmark_point[11]), tuple(landmark_point[12]),
191                (255, 255, 255), 2)
192
193         # Ring finger
194         cv.line(image, tuple(landmark_point[13]), tuple(landmark_point[14]),
195                (0, 0, 0), 4)
196         cv.line(image, tuple(landmark_point[13]), tuple(landmark_point[14]),
197                (255, 255, 255), 2)
198         cv.line(image, tuple(landmark_point[14]), tuple(landmark_point[15]),
199                (0, 0, 0), 4)
200         cv.line(image, tuple(landmark_point[14]), tuple(landmark_point[15]),
201                (255, 255, 255), 2)
202         cv.line(image, tuple(landmark_point[15]), tuple(landmark_point[16]),
203                (0, 0, 0), 4)
204         cv.line(image, tuple(landmark_point[15]), tuple(landmark_point[16]),
205                (255, 255, 255), 2)
206
207         # Little finger
208         cv.line(image, tuple(landmark_point[17]), tuple(landmark_point[18]),
209                (0, 0, 0), 4)
210         cv.line(image, tuple(landmark_point[17]), tuple(landmark_point[18]),
211                (255, 255, 255), 2)
212         cv.line(image, tuple(landmark_point[18]), tuple(landmark_point[19]),
213                (0, 0, 0), 4)
214         cv.line(image, tuple(landmark_point[18]), tuple(landmark_point[19]),
215                (255, 255, 255), 2)

```

```

216 cv.line(image, tuple(landmark_point[19]), tuple(landmark_point[20]),
217         (0, 0, 0), 4)
218 cv.line(image, tuple(landmark_point[19]), tuple(landmark_point[20]),
219         (255, 255, 255), 2)
220
221 # Palm
222 cv.line(image, tuple(landmark_point[0]), tuple(landmark_point[1]),
223         (0, 0, 0), 4)
224 cv.line(image, tuple(landmark_point[0]), tuple(landmark_point[1]),
225         (255, 255, 255), 2)
226 cv.line(image, tuple(landmark_point[1]), tuple(landmark_point[2]),
227         (0, 0, 0), 4)
228 cv.line(image, tuple(landmark_point[1]), tuple(landmark_point[2]),
229         (255, 255, 255), 2)
230 cv.line(image, tuple(landmark_point[2]), tuple(landmark_point[5]),
231         (0, 0, 0), 4)
232 cv.line(image, tuple(landmark_point[2]), tuple(landmark_point[5]),
233         (255, 255, 255), 2)
234 cv.line(image, tuple(landmark_point[5]), tuple(landmark_point[9]),
235         (0, 0, 0), 4)
236 cv.line(image, tuple(landmark_point[5]), tuple(landmark_point[9]),
237         (255, 255, 255), 2)
238 cv.line(image, tuple(landmark_point[9]), tuple(landmark_point[13]),
239         (0, 0, 0), 4)
240 cv.line(image, tuple(landmark_point[9]), tuple(landmark_point[13]),
241         (255, 255, 255), 2)
242 cv.line(image, tuple(landmark_point[13]), tuple(landmark_point[17]),
243         (0, 0, 0), 4)
244 cv.line(image, tuple(landmark_point[13]), tuple(landmark_point[17]),
245         (255, 255, 255), 2)

```

```

246 cv.line(image, tuple(landmark_point[17]), tuple(landmark_point[0]),
247         (0, 0, 0), 4)
248 cv.line(image, tuple(landmark_point[17]), tuple(landmark_point[0]),
249         (255, 255, 255), 2)
250
251 # Key Points
252 for index, landmark in enumerate(landmark_point):
253     if index == 0: # ???
254         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
255
256         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
257
258     if index == 1: # ???
259         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
260         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
261
262     if index == 2: # ???
263         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
264         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
265
266     if index == 3: # ???
267         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
268         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
269
270     if index == 4: # ???
271         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
272         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
273         cv.circle(image, (landmark[0], landmark[1]), 4, (255, 255, 0), 2)
274
275     if index == 5: # ???
276         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
277         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)

```

```

278
279     if index == 6: # ???
280         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
281         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
282
283     if index == 7: # ???
284         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
285         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
286
287     if index == 8: # ???
288         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
289         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
290
291     if index == 9: # ???
292         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
293         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
294
295     if index == 10: # ???
296         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
297         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
298
299     if index == 11: # ???
300         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
301         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
302
303     if index == 12: # ???
304         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
305         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
306
307     if index == 13: # ???
308         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
309         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
310
311     if index == 14: # ???
312         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)

```

```

307     cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
308     if index == 15: # ??:??
309         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
310         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
311     if index == 16: # ??:??
312         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
313         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
314         cv.circle(image, (landmark[0], landmark[1]), 4, (255, 255, 0), 2)
315
316     if index == 17: # ??:??
317         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
318         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
319     if index == 18: # ??:??
320         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
321         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
322     if index == 19: # ??:??
323         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
324         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
325     if index == 20: # ??:??
326         cv.circle(image, (landmark[0], landmark[1]), 3, (255, 255, 255), -1)
327         cv.circle(image, (landmark[0], landmark[1]), 3, (0, 0, 0), 1)
328         cv.circle(image, (landmark[0], landmark[1]), 4, (255, 255, 0), 2)
329
330     return image
331
332     def draw_bounding_rect(use_brect, image, brect):
333         if use_brect:
334             # Outer rectangle
335             cv.rectangle(image, (brect[0], brect[1]), (brect[2], brect[3]), (0, 0, 255), 3)
336
337         return image

```

```

340     def draw_info_text(image, brect, handedness, hand_sign_text):
341         global confidencia, limite
342         cv.rectangle(image, (brect[0], brect[1]), (brect[2], brect[1] - 22), (0, 0, 255), -1)
343         info_text = handedness.classification[0].label[0:]
344         if(info_text=="Right"):
345             info_text = "Der"
346         if (info_text == "Left"):
347             info_text = "Izq"
348
349         if hand_sign_text != "":
350             info_text = info_text + ' ' + hand_sign_text
351         if(confidencia >= limite):
352             cv.putText(image, info_text, (brect[0] + 5, brect[1] - 4), cv.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 255))
353             confidencia2 = "{:.0f}".format(confidencia*100)
354             cv.rectangle(image, (0, 40), (220, 80), (62, 62, 62), -1)
355             cv.putText(image, "Confidencia: " + str(confidencia2)+"%", (10, 60), cv.FONT_HERSHEY_SIMPLEX, (0.7, 0.7), (255, 255, 255))
356             cv.putText(image, "Confidencia: " + str(confidencia2)+"%", (10, 60), cv.FONT_HERSHEY_SIMPLEX, (0.7, 0.7), (255, 255, 255))
357         return image
358     def update_video():
359         global cap, siCamaraConectada, cntCamara, cntLetra, cntPalabra, letraActual, letraSiguiente, palabraActual
360
361         try:
362             if(siCamaraConectada==False):
363
364                 siCamaraConectada = True
365                 # Iniciar la captura de video con OpenCV
366                 cap = cv2.VideoCapture(0)
367                 cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
368                 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

```

```

370         fps = cvFpsCalc.get()
371         ret, frame = cap.read()
372         if ret:
373             frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
374             # Process Key (ESC: end) #####
375             image = cv.flip(frame, 1) # Mirror display
376             debug_image = copy.deepcopy(image)
377             # Detection implementation #####
378             image.flags.writeable = False
379             results = hands.process(image)
380             image.flags.writeable = True
381
382             # #####
383             if results.multi_hand_landmarks is not None:
384
385                 for hand_landmarks, handedness in zip(results.multi_hand_landmarks, results.multi_hand_landmarks):
386                     # Bounding box calculation
387                     brect = calc_bounding_rect(debug_image, hand_landmarks)
388                     # Landmark calculation
389                     landmark_list = calc_landmark_list(debug_image, hand_landmarks)
390                     # Conversion to relative coordinates / normalized coordinates
391                     pre_processed_landmark_list = pre_process_landmark(landmark_list)
392                     # Hand sign classification
393                     hand_sign_id, confidencia = keypoint_classifier(pre_processed_landmark_list)
394                     confidencia = round(confidencia, 2)
395                     # print(confidencia)
396                     if hand_sign_id == 2: # Point gesture
397                         point_history.append(landmark_list[8])
398                     else:
399                         point_history.append([0, 0])
400                     # Finger gesture classification

```

```

401     finger_gesture_id = 0
402     # Calculates the gesture IDs in the latest detection
403     finger_gesture_history.append(finger_gesture_id)
404     most_common_fg_id = Counter(
405         finger_gesture_history).most_common()
406     # Drawing part
407     debug_image = draw_bounding_rect(use_brect, debug_image, brect)
408     debug_image = draw_landmarks(debug_image, landmark_list)
409     debug_image = draw_info_text(
410         debug_image,
411         brect,
412         handedness,
413         keypoint_classifier_labels[hand_sign_id]
414     )
415
416     letraActual = keypoint_classifier_labels[hand_sign_id]
417     cntPalabra = 0
418     if letraActual != "":
419         # print(confide)
420         if (confidencia >= limite):
421             palabra_entry.delete(0, tk.END)
422             palabra_entry.insert(0, palabra)
423             cntLetra += 1
424             if (cntLetra == 25):
425                 if (letraActual == "BORRAR"):
426                     print("borrar letra")
427                     letraActual = ""
428                     letraSiguiete = ""
429             # palabra_entry.delete(cntletras)
430             cntletras -= 1
431             print(palabra)

```

```

432         letra_entry.delete(0, tk.END)
433         letra_entry.insert(0, " ")
434         palabra = palabra[:-1]
435         print(palabra)
436         try:
437             image = Image.open(
438                 "/home/nano/Desktop/TESIS/proyecto/SI
439             image = image.resize((180, 180))
440             photo = ImageTk.PhotoImage(image)
441             imagen_label.config(image=photo)
442             imagen_label.image = photo
443         except:
444             print("error imagen")
445         else:
446             if (letraActual != letraSiguiete):
447
448                 if (letraActual == "SPACE"):
449                     letraSiguiete = " "
450                     palabra += letraSiguiete
451                     cntLetra = 0
452                     letra_entry.delete(0, tk.END)
453                     letra_entry.insert(0, letraSiguiete)
454                     image = Image.open(
455                         "/home/nano/Desktop/TESIS/proyect
456                     image = image.resize((180, 180))
457                     photo = ImageTk.PhotoImage(image)
458                     imagen_label.config(image=photo)
459                     imagen_label.image = photo
460                     palabra_entry.delete(0, tk.END)
461                     palabra_entry.insert(0, palabra)

```

```

462         print(palabra)
463         else:
464             letraSiguiete = letraActual
465             palabra += letraSiguiete
466             cntLetra = 0
467
468             letra_entry.delete(0, tk.END)
469             letra_entry.insert(0, letraSiguiete)
470             image = Image.open(
471                 "/home/nano/Desktop/TESIS/proyect
472             image = image.resize((180, 180))
473             photo = ImageTk.PhotoImage(image)
474             imagen_label.config(image=photo)
475             imagen_label.image = photo
476             palabra_entry.delete(0, tk.END)
477             palabra_entry.insert(0, palabra)
478             print(palabra)
479
480         else:
481             cntborrar = 0
482             cntLetra = 0
483             # print(keypoint_classifier_labels[hand_sign_id])
484
485     else:
486         cntborrar = 0
487         cntLetra = 0
488         cntPalabra += 1
489         if (palabra != ""):
490             if (cntPalabra >= 20):
491                 cntPalabra = 0
492                 cntletras = 0
493

```

```

494         try:
495             texto = "h " + palabra
496             nombre_archivo = "/home/nano/Desktop/TESIS/pr
497
498             # print("antes de gtts: " + texto)
499             texto_a_voz(texto, nombre_archivo)
500         except Exception as e:
501             print("ERROR DE GTTS PARCE: " + str(e))
502         palabra = ""
503         letraActual = ""
504         letraSiguiente = ""
505         palabra_entry.delete(0, tk.END)
506         cntletras = 0
507
508         root.update() # Actualiza la interfaz para mostrar la imagen
509         debug_image = draw_info(debug_image, fps, mode)
510         image = Image.fromarray(debug_image)
511         image = image.resize((600, 400))
512         photo = ImageTk.PhotoImage(image)
513         video_label.config(image=photo)
514         video_label.image = photo
515         cntCamara = 0
516
517     else:
518
519         siCamaraConectada = False
520
521         cntCamara = cntCamara + 1
522
523         if cntCamara == 1:
524             video_label.config(image=None)

```

```

525             video_label.image = None
526         elif cntCamara == 200:
527             nombre_archivo = "/home/nano/Desktop/TESIS/proyecto/SIGNO
528             playsound(nombre_archivo)
529             cntCamara = 0
530
531             video_label.after(1, update_video)
532
533         except Exception as error:
534             siCamaraConectada = False
535
536             cntCamara = cntCamara+1
537             print(cntCamara)
538
539             if cntCamara == 1:
540                 print("Errorrr:", str(error)) # An error occurred: NameError
541                 video_label.config(image=None)
542                 video_label.image = None
543
544             elif cntCamara == 200:
545
546                 nombre_archivo = "/home/nano/Desktop/TESIS/proyecto/SIGNOS_VI
547                 playsound(nombre_archivo)
548                 cntCamara = 0
549
550 root = Tk()
551 root.title("INTERFAZ")
552
553 root.configure(bg="#191B1B")
554 root.geometry("700x400")
555 # Obtener la resolución de la pantalla
556

```

```

556 screen_width = root.winfo_screenwidth()
557 screen_height = root.winfo_screenheight()
558 # Calcular las coordenadas x e y para centrar la ventana
559 x = (screen_width // 2) - (800 // 2)
560 y = (screen_height // 2) - (500 // 2)
561 # Establecer las coordenadas para centrar la ventana
562 # root.geometry(f"800x500+(x)+(y)")
563
564 # Crear el marco principal para la interfaz
565 main_frame = tk.Frame(root)
566 # main_frame.pack()
567 # Crear el label para el video
568 video_label = tk.Label()
569 video_label.place(x=10, y=10)
570
571 # Crear el marco para la sección derecha
572 right_frame = tk.Frame()
573 # right_frame.pack(side=tk.LEFT, fill=tk.BOTH, expand=False)
574
575 # Crear el texto "Letra"
576 letra_label = tk.Label(font=("Arial", 15, "bold"), text="LETRA", fg="white", bg="#191B1B")
577 letra_label.place(x=675, y=15)
578
579 # Crear el text entry para la letra generada
580 letra_entry = tk.Entry(width=3, justify="center", font=("Arial", 35, "bold"))
581 letra_entry.place(x=670, y=40)
582
583 # Cargar y mostrar la imagen estática
584 imagen_label = tk.Label()
585 imagen_label.place(x=620, y=100)

```

```

587 # Cargar y mostrar la imagen estática
588 imagen_esPOCH = tk.Label()
589 imagen_esPOCH.place(x=630, y=300)
590 image = Image.open("/home/nano/Desktop/TESIS/proyecto/SIGNOS_VISION/esPOCH.png")
591 image = image.resize((150, 150))
592 photo = ImageTk.PhotoImage(image)
593 imagen_esPOCH.config(image=photo)
594 imagen_esPOCH.image = photo
595
596 # Crear el texto "Palabra"
597 palabra_label = tk.Label(font=("Arial", 12, "bold"), text="FRASE", fg="white", bg="#19181B")
598 palabra_label.place(x=10, y=420)
599
600 # Crear el text entry para la palabra de la aplicación
601 palabra_entry = tk.Entry(width=48, justify="center", font=("Arial", 15"))
602 palabra_entry.place(x=80, y=420)
603
604 nombre_archivo = "/home/nano/Desktop/TESIS/proyecto/SIGNOS_VISION/intro.mp3"
605 playsound(nombre_archivo)
606
607 # Iniciar la captura de video con OpenCV
608 #cap = cv2.VideoCapture(0)
609 #cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
610 #cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
611 # Actualizar el video en el label
612 update_video()
613 root.mainloop()
614 # Liberar los recursos de la cámara web al cerrar la ventana
615 cap.release()
616 cv2.destroyAllWindows()

```

b) Entrenador

```

1 import csv
2 import numpy as np
3 import tensorflow as tf
4 from sklearn.model_selection import train_test_split
5 # matriz de confusión
6 import pandas as pd
7 import seaborn as sns
8 import matplotlib.pyplot as plt
9 from sklearn.metrics import confusion_matrix, classification_report
10
11 RANDOM_SEED = 30
12 # Designación de cada ruta
13 dataset = "model/keypoint_classifier/keypoint5.csv"
14 model_save_path = "model/keypoint_classifier/keypoint_classifier.hdf5"
15 # Ajuste del número de clasificación
16 NUM_CLASSES = 26
17 # Carga de datos de aprendizaje
18 X_dataset = np.loadtxt(dataset, delimiter=',', dtype="float32", usecols=list(range(1, (21 * 2) + 1))
19
20 y_dataset = np.loadtxt(dataset, delimiter=',', dtype="int32", usecols=(0))
21
22 X_train, X_test, y_train, y_test = train_test_split(X_dataset, y_dataset, train_size=0.70, random_s
23 # Construcción de modelos
24 model = tf.keras.models.Sequential([
25     tf.keras.layers.Input((21 * 2, )),
26     tf.keras.layers.Dropout(0.1),
27     tf.keras.layers.Dense(20, activation="relu"),
28     tf.keras.layers.Dropout(0.1),
29     tf.keras.layers.Dense(25, activation="relu"),
30     tf.keras.layers.Dense(NUM_CLASSES, activation="softmax")
31 ])
32
33 model.summary() # tf.keras.utils.plot_model(model, show_shapes=True)
34
35 # Llamadas al punto de control del modelo
36 cp_callback = tf.keras.callbacks.ModelCheckpoint(
37     model_save_path, verbose=1, save_weights_only=False)
38 # Devolución de llamada por cancelación anticipada
39 es_callback = tf.keras.callbacks.EarlyStopping(patience=20, verbose=1)
40
41 # compilación de modelos
42 model.compile(
43     optimizer="adam",
44     loss="sparse_categorical_crossentropy",
45     metrics=["accuracy"]
46 )
47
48 model.fit(
49     X_train,
50     y_train,
51     epochs=1000,
52     batch_size=128,
53     validation_data=(X_test, y_test),
54     callbacks=[cp_callback, es_callback]
55 )

```

```

58 # Evaluación de modelos
59 val_loss, val_acc = model.evaluate(X_test, y_test, batch_size=128)
60
61 # Cargar modelos guardados.
62 model = tf.keras.models.load_model(model_save_path)
63
64 # Prueba de razonabilidad
65 predict_result = model.predict(np.array([X_test[0]]))
66 print(np.squeeze(predict_result))
67 print(np.argmax(np.squeeze(predict_result)))
68
69
70 # Conversión a modelos para Tensorflow-Lite.
71 # Almacenar como modelos sólo de inferencia.
72 model.save(model_save_path, include_optimizer=False)
73
74 # Transformar (cuantificar) el modelo
75 tflite_save_path = 'model/keypoint_classifier/keypoint_classifier7.tflite'
76
77 converter = tf.lite.TFLiteConverter.from_keras_model(model)
78 converter.optimizations = [tf.lite.Optimize.DEFAULT]
79 tflite_quantized_model = converter.convert()
80
81 open(tflite_save_path, 'wb').write(tflite_quantized_model)
82
83
84 interpreter = tf.lite.Interpreter(model_path=tflite_save_path)
85 interpreter.allocate_tensors()
86
87 # Obtener tensor de entrada/salida.
88 input_details = interpreter.get_input_details()
89 output_details = interpreter.get_output_details()
90 interpreter.set_tensor(input_details[0]['index'], np.array([X_test[0]]))
91 # Aplicación del razonamiento
92 interpreter.invoke()
93 tflite_results = interpreter.get_tensor(output_details[0]['index'])
94 print(np.squeeze(tflite_results))
95 print(np.argmax(np.squeeze(tflite_results)))
96 ##### matriz de confusión #####
97 def print_confusion_matrix(y_true, y_pred, report=True):
98     labels = sorted(list(set(y_true)))
99     cmx_data = confusion_matrix(y_true, y_pred, labels=labels)
100
101     df_cmx = pd.DataFrame(cmx_data, index=labels, columns=labels)
102
103     fig, ax = plt.subplots(figsize=(7, 6))
104     sns.heatmap(df_cmx, annot=True, fmt='g', square=False)
105     ax.set_ylim(len(set(y_true)), 0)
106     plt.show()
107
108     if report:
109         print('Classification Report')
110         print(classification_report(y_test, y_pred))
111 Y_pred = model.predict(X_test)
112 y_pred = np.argmax(Y_pred, axis=1)
113 print_confusion_matrix(y_test, y_pred)

```





ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 25/11/2023

INFORMACIÓN DE LOS AUTORES	
Nombres – Apellidos: Javier Moises Buñay Cujilema Jose Emilio Mullo Yautibug	
INFORMACIÓN INSTITUCIONAL	
Facultad: Informática y Electrónica.	
Carrera: Electrónica y Automatización.	
Título a optar: Ingeniero en Electrónica y Automatización.	
f. Analista de Biblioteca responsable:	 Ing. Fernanda Arévalo M.



1856-DBRA-UPT-2023