



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TÍTULO DEL TRABAJO DE TITULACIÓN

**DESARROLLO DE UNA APP QUE DETECTA MOVIMIENTO DE MANOS
Y TRADUCE SEÑAS PARA LA FUNDACIÓN MELVIN JONES**

AUTOR

POZO GONZÁLEZ JESÚS MICHAEL

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

**Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

TUTOR

Ing. Jaime Orozco Iguasnia, Mgt

La Libertad, Ecuador

Año 2024



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgt.
DIRECTOR DE LA CARRERA

Ing. Jaime Orozco Iguasnia, Mgt.
TUTOR

Ing. Walter Orozco Iguasnia, Mgt.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suarez, Mgt.
DOCENTE GUÍA UIC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA**

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por POZO GONZÁLEZ JESÚS MICHAEL, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 18 días del mes de junio del año 2024

TUTOR

Ing. Jaime Orozco Iguasnia, Mgt



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA**

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, POZO GONZÁLEZ JESÚS MICHAEL

DECLARO QUE:

El trabajo de Titulación, Desarrollo de una App que detecta movimiento de manos y traduce señas para la fundación Melvin Jones, previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 18 días del mes de junio del año 2024

EL AUTOR

A handwritten signature in blue ink, appearing to read "Pozo González Jesús Michael", is written over a horizontal line.

Pozo González Jesús Michael



UPSE

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Desarrollo de una App que detecta movimiento de manos y traduce señas para la fundación Melvin Jones, presentado por el estudiante, POZO GONZÁLEZ JESÚS MICHAEL fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al $< 1\%$, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

CERTIFICADO DE ANÁLISIS
magister

JESÚS MICHAEL POZO GONZÁLEZ

< 1%
Textos sospechosos

< 1% Similitudes
0% similitudes entre comillas
0% entre las fuentes mencionadas
0% Idiomas no reconocidos

Nombre del documento: JESÚS MICHAEL POZO GONZÁLEZ.pdf
ID del documento: 3c125facbf2c60a2f004ac1a67f850d263574687
Tamaño del documento original: 3,38 MB

Depositante: JAIME BENJAMÍN OROZCO IGUASNIA
Fecha de depósito: 18/6/2024
Tipo de carga: interface
fecha de fin de análisis: 18/6/2024

Número de palabras: 15.062
Número de caracteres: 95.854

Ubicación de las similitudes en el documento:

Fuente principal detectada

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.upse.edu.ec https://repositorio.upse.edu.ec/bitstream/46000/10191/1/UPSE-TT-2023-0049.pdf	< 1%		Palabras idénticas: < 1% (36 palabras)

TUTOR

Ing. Jaime Orozco Iguasnia, Mgt.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, POZO GONZÁLEZ JESÚS MICHAEL

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de titulación con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 18 días del mes de junio del año 2024

EL AUTOR

A handwritten signature in blue ink, appearing to read "Pozo González Jesús Michael", is written over a horizontal line.

Pozo González Jesús Michael

AGRADECIMIENTO

Agradezco a cada una de las personas que me brindaron su apoyo y creyeron que podía terminar esto que un día empecé. A mis padres, Raúl Pozo Rosales por permitirme enfocarme en mis estudios enseñándome que con constancia y esfuerzo es posible lograr los objetivos, a mi madre por darme fuerzas cada día con su amor y cariño para continuar y no desmayar ni darme por vencido en el camino llenándome de valores y principios.

A mis hermanos, Jefferson Pozo González por haberme forjado, a mi hermano Josthin Pozo González por ser la persona con la que puedo contar siempre y a mi hermana Melissa Pozo González por brindarme su cariño.

A mis mascotas por ser quienes me acompañaban donde no había más que oscuridad y desesperación y que con tan solo una caricia o ronroneo lograban llenar mi alma para que volviera ver una luz de esperanza y seguir adelante.

A mis docentes que me brindaron sus enseñanzas y aprendizaje para formarme académicamente y a mis compañeros de estudio con los que podía contar para realizar algún trabajo, a todos y cada uno de ellos les agradezco.

Pozo González Jesús Michael

DEDICATORIA

Dedico este trabajo a mis padres Rosa González y Raúl Pozo, sin ellos nada sería posible y por ellos estoy aquí el día de hoy, son las personas que más creyeron en mí desde el primer momento dándome su apoyo moral y quienes nunca duraron en que podía cumplir mis objetivos.

A mis hermanos Jefferson Pozo, Josthin Pozo y mi hermana Melissa Pozo por darme las fuerzas que me hacían falta, así como también la motivación para terminar mi carrera, a mis mascotas que, aunque no pudieran hablar siempre estuvieron ahí y con eso es más que suficiente.

A mis compañeros de estudio por haberme acompañado durante este proceso dándome su apoyo y confiar en que podía concluir este proyecto de manera satisfactoria, siempre los tendré presente a cada uno de ellos y es por eso por lo que les dedico este trabajo.

Pozo González Jesús Michael

ÍNDICE GENERAL

TÍTULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XI
ÍNDICE DE FIGURAS	XII
ÍNDICE DE ANEXOS	XVII
RESUMEN	XVIII
ABSTRACT	XIX
INTRODUCCIÓN	2
CAPÍTULO I	2
1. FUNDAMENTACIÓN	2
1.1 ANTECEDENTES	2
1.2 DESCRIPCIÓN DEL PROYECTO	5
1.3 OBJETIVOS DEL PROYECTO	7
1.4 JUSTIFICACIÓN	8
1.5 ALCANCE DEL PROYECTO	10
1.6 METODOLOGÍA DEL PROYECTO	12

1.6.1	METODOLOGÍA DE LA INVESTIGACIÓN	12
1.6.2	BENEFICIARIOS DEL PROYECTO	13
1.6.3	VARIABLE	14
1.6.4	ANÁLISIS DE RECOLECCIÓN DE DATOS	14
1.7	METODOLOGÍA DE DESARROLLO	15
CAPÍTULO II		17
1.	PROPUESTA	17
2.1	MARCO CONTEXTUAL	17
2.2	MARCO CONCEPTUAL	18
2.3	MARCO TEÓRICO	25
2.4	REQUERIMIENTOS	29
2.4.1	REQUERIMIENTOS FUNCIONALES	29
2.4.2	REQUERIMIENTOS NO FUNCIONALES	31
2.5	COMPONENTES DE LA PROPUESTA	32
2.5.1	ARQUITECTURA DEL SISTEMA	32
2.5.2	INSTRUMENTOS DE DESARROLLO	33
2.5.3	DESARROLLO DEL ALGORITMO DE DETECCIÓN Y TRADUCCIÓN DE SEÑAS MEDIANTE EL USO DE LA SOLUCIÓN MEDIAPIPE HANDS.	34
2.5.4	CREACIÓN DE LA API PARA LA CONEXIÓN DE LA APLICACIÓN MÓVIL CON EL SERVIDOR Y BASE DE DATOS.	38
2.5.5	DIAGRAMAS DE CASO DE USO	44
2.5.6	MODELADO DE DATOS	48
2.6	DISEÑO DE LA INTERFAZ	49
2.7	PRUEBAS	55
2.8	RESULTADOS	63
CONCLUSIONES		68
RECOMENDACIONES		69
REFERENCIAS		70

ÍNDICE DE TABLAS

Tabla 1 Beneficiarios del Proyecto	13
Tabla 2 Población del proyecto.	14
Tabla 3 Soluciones de MediaPipe	25
Tabla 4. Requerimientos Funcionales	31
Tabla 5 Requerimientos No Funcionales	32
Tabla 6 Caso de uso para iniciar una captura	46
Tabla 7 Caso de uso actualizar dataset y clasificar los datos	47
Tabla 8 Caso de uso para comprobar patrones	48
Tabla 9 Prueba de funcionalidad - Inicio de sesión	56
Tabla 10. Prueba de funcionalidad - registro de usuarios	58
Tabla 11. Prueba de funcionalidad - detectar y traducir seña	59
Tabla 12 Caso de funcionalidad - Agregar un nuevo patrón e iniciar captura	61
Tabla 13 Caso de funcionalidad - Agregar un nuevo patrón y entrenarlo	62
Tabla 14 Prueba de funcionalidad - Ver las imágenes de los patrones existentes	63
Tabla 15 Resultados de prueba funcional de agregar patrones	64
Tabla 16 Resultados de prueba funcional del módulo de traducción	65

ÍNDICE DE FIGURAS

Fig. 1. Fases de la metodología incremental	16
Fig. 2. Mapa de ubicación: CENTRO DE EDUCACIÓN INTEGRAL MELVIN JONES.	17
Fig. 3 Tipos de aprendizaje automático [1]	19
Fig. 4. Hand Landmark Model	23
Fig. 5. Cuadro delimitador en el reconocimiento de un objeto	24
Fig. 6. Arquitectura cliente – servidor	32
Fig. 7. Importaciones para realizar cada uno de los métodos de la API	34
Fig. 8. Funciones para cargar etiquetas almacenadas y proporcionar nuevos IDs	34
Fig. 9. Método para generar la captura de imágenes.	35
Fig. 10. Método para actualizar el dataset	35
Fig. 11. Método para clasificar las muestras	36
Fig. 12. Clase Traductor	36
Fig. 13. Función de la clase traductor process_hand_coordinates.	37
Fig. 14. Función de la clase traductor process_frame	37
Fig. 15. Método para realizar la traducción	38
Fig. 16. Conexión de la API con cada uno de los métodos	38
Fig. 17. Conexión con la base de datos	39
Fig. 18. Método para iniciar sesión	39
Fig. 19. Método para registro de usuarios	40
Fig. 20. Conexión a la API del servidor desde la aplicación flutter en Android Studio	40
Fig. 21. Ruta para iniciar la captura de imágenes desde la aplicación flutter en Android Studio	41

Fig. 22. Rutas para obtener los nombres de las clases guardadas desde la aplicación flutter en Android Studio	41
Fig. 23. Ruta para entrenar el modelo desde la aplicación flutter en Android Studio	42
Fig. 24. Ruta para obtener una imagen aleatoria desde la aplicación flutter en Android Studio	42
Fig. 25- Ruta que sirve para actualizar el dataset desde la aplicación flutter en Android Studio	43
Fig. 26. Ruta para iniciar sesión desde la aplicación flutter en Android Studio	43
Fig. 27. Diagrama de caso de uso general para los tipos de usuarios de la aplicación	44
Fig. 28. Diagrama de caso de uso para iniciar capturas	45
Fig. 29. Diagrama de caso de uso para actualizar el dataset y clasificar los datos	46
Fig. 30. Diagrama de caso de uso para comprobar los patrones registrados	47
Fig. 31. Modelado de base de datos	48
Fig. 32. Interfaz de inicio de sesión	49
Fig. 33. Interfaz de registro de usuarios	49
Fig. 34. Interfaz de administrador	50
Fig. 35. Traducción realizada con la mano izquierda	51
Fig. 36. Traducción realizada con la mano derecha	51
Fig. 37. Interfaz para el entrenamiento del modelo	51
Fig. 38. Interfaz para agregar nuevos patrones	52
Fig. 39. Interfaz de agregar nuevos patrones al generar nombre de la clase	52
Fig. 40. Interfaz de usuario	53
Fig. 41. Interfaz de traducción para el usuario	53
Fig. 42. Traducción mano der. interfaz de usuario	54
Fig. 43. Traducción mano izq. interfaz usuario	54

Fig. 44. Interfaz para ver patrones grabados	54
Fig. 45. Menú para ver patrones guardados	55
Fig. 46. Interfaz de usuario admin	83
Fig. 47. Inicio de sesión - Usuario administrador	84
Fig. 48. Ver contraseña	85
Fig. 49. Ingresat credenciales de administrador	85
Fig. 50. Interfaz para iniciar sesión	85
Fig. 51. Oprimir el botón iniciar sesión	86
Fig. 52. Interfaz administrador manual de usuario	86
Fig. 53. Botón agregar nuevo patron	87
Fig. 54. Iniciar captura de imágenes	88
Fig. 55. Iniciar captura de imágenes	88
Fig. 56. Iniciar captura con nombre de seña repetida	89
Fig. 57. Iniciar captura sin ingresar datos en los campos	89
Fig. 58. Iniciar captura con datos correctos	90
Fig. 59. Realizar captura con mano derecha	90
Fig. 60. Ingreso de nombre de clase con datos correctos	90
Fig. 61. Realizar capturas con mano izquierda	90
Fig. 62. Mensaje dataset actualizado	91
Fig. 63. Opción 2 crear dataset	91
Fig. 64. Muestras clasificadas	92
Fig. 65. Opcion 3 clasificador	92
Fig. 66. Traducción de seña con mano izquierda	93
Fig. 68. Oprimir el botón iniciar traducción en administrador	93

Fig. 67. Traducción de seña con mano derecha	93
Fig. 69. Inicio se sesión - Usuario no administrador	94
Fig. 70. Ver contraseña usuario no admin	95
Fig. 71. Inicio de sesión con credenciales de usuarios erróneos	95
Fig. 72. Error de inicio de sesión	96
Fig. 73. Oprimir el botón registrase	97
Fig. 74. Navegar a la interfaz de registro de usuarios	97
Fig. 75. Registrar usuario	97
Fig. 76. Registrar usuarios con campos vacios	97
Fig. 77. Login exitoso - interfaz de usuario	98
Fig. 78. Inicio de sesión exitoso - usuario	98
Fig. 79. Botón ver patrones guardados	99
Fig. 80. Interfaz para ver los patrones existentes	99
Fig. 81. Obtener imagen aleatoria del patrón elegido	100
Fig. 82. Menú desplegable con señas guardadas en el servidor	100
Fig. 83. Botón traducir de interfaz de usuario	101
Fig. 84. Botón que inicia la traducción de señas	101
Fig. 86. Traducción de seña con mano izquierda interfaz de usuario	101
Fig. 85. Traducción de seña con mano derecha de interfaz de usuario	101
Fig. 87. Descargar	102
Fig. 88. Seleccionar descargar	102
Fig. 89. Descargar de todos modos	103
Fig. 90. Abrir	103
Fig. 91. Instalar	103

ÍNDICE DE ANEXOS

Anexo 1 Formato de entrevista para la directora del CEI Melvin Jones	77
Anexo 2. Encuesta de disponibilidad y uso de la aplicación móvil para traducir el alfabeto de lengua de señas	79
Anexo 3. Manual de usuario para usuario administrador	83
Anexo 4. Manual de usuario para usuario no administrador	94
Anexo 5. Descarga e instalación	102

RESUMEN

El presente trabajo tiene como finalidad desarrollar una Aplicación móvil que sea capaz de detectar los movimientos de seña de una mano y procesarlas con el propósito de entrenar el modelo para posteriormente poder traducirla a texto, usando señas del alfabeto del lenguaje de señas ecuatoriano y de esta forma poder convertirse en una herramienta en la que se pueda identificar la postura de una seña a fin de que las personas que presentan problemas auditivos puedan ser capaces de mantener una mejor interacción al momento de comunicarse con el entorno social que los rodea. Para el desarrollo de esta aplicación se hizo uso de MediaPipe Solutions para detectar los puntos de referencia de los nudillos de las manos trabajando con soluciones de aprendizaje automático, usando Python como lenguaje de programación, desarrollo de la API y del código para realizar el entrenamiento del modelo de detección de seña, para el desarrollo de la aplicación móvil se usó flutter como framework y como metodología de desarrollo se llevó a cabo la metodología incremental conformado por 4 fases: análisis, diseño, codificación y pruebas. Se consiguió determinar la funcionalidad de la aplicación mediante las pruebas realizadas logrando procesar y captar la postura de las señas efectuadas para posteriormente entrenarlas mediante el algoritmo creado con el que se especificará un porcentaje de clasificación de cada nuevo patrón, permitiendo obtener una aplicación móvil con el que el usuario pueda visualizar como se realiza una seña, aprenderla y traducirla mediante el algoritmo de traducción de la aplicación.

Palabras claves: Aplicación móvil, MediaPipe, lenguaje de señas, aprendizaje automático

ABSTRACT

The purpose of this work is to develop a mobile application that is able to detect the movements of a hand sign and process them in order to train the model and then translate it into text, using signs from the dictionary of Ecuadorian sign language and thus be able to become a tool that can identify the position of a sign so that people with hearing problems may be able to maintain a better interaction when communicating with the social environment that surrounds them. For the development of this application, MediaPipe Solutions was used to detect the reference points of the knuckles of the hands working with machine learning solutions, using Python as programming language, development of the API and code for the training of the sign detection model, for the development of the mobile application flutter was used as a framework and as development methodology the incremental methodology was carried out, consisting of 4 phases: analysis, design, coding and testing. It was possible to determine the functionality of the application by means of the tests carried out, managing to process and capture the posture of the signs made to later train them by means of the algorithm created with which a classification percentage of each new pattern will be specified, allowing to obtain a mobile application with which the user can visualize how a sign is made, learn it and translate it by means of the application's translation algorithm.

Keywords: Mobile application, MediaPipe, sign language, machine learning

INTRODUCCIÓN

La comunicación es una de las partes fundamentales que tienen los seres humanos y que ayuda a relacionarse con las demás personas a lo largo de la vida, a través de ella las personas son capaces de transmitir sus ideas, emociones y sentimientos, el proceso de comunicación comúnmente se da de forma oral mediante el habla, tiene como características el producir ondas sonoras las cuales son percibidas por el canal auditivo hasta llegar al cerebro en donde son procesados.

Aunque la comunicación sea importante y se realice mediante el habla no todos son capaces de realizarla de esa manera lo que provoca que estas personas quienes presentan problemas auditivos tengan dificultades para comunicarse con otros individuos, sin embargo a medida que el ingenio humano avanza se ha buscado una alternativa en que los seres que no pueden comunicarse por medio de la voz puedan expresar sus ideas con otras personas, en estos casos la comunicación es realizada mediante gestos, siendo en el siglo XVII donde se originan los primeros rastros del uso de lenguaje de señas.

Con la evolución de la tecnología se han logrado obtener grandes avances en relación con la comunicación a través del lenguaje de señas, creando soluciones de aprendizaje automático de reconocimiento en donde las personas que son capaces de desarrollar algoritmos y de esta manera poder crear nuevas herramientas tecnológicas que faciliten el proceso de comunicación entre personas que tienen deficiencia auditiva.

Es por este motivo que mediante la realización de este trabajo de investigación y el uso de herramientas de reconocimiento como MediaPipe se pretende desarrollar una aplicación móvil que sea capaz de reconocer el movimiento realizado por una mano y poder captar señas que a su vez se traducirán a texto para que se pueda entender y poder establecer una comunicación entre personas.

En conclusión, por medio de la finalización del proyecto se pretende conseguir una aplicación móvil que sirva como un traductor del alfabeto de lengua de señas a palabras basado en el lenguaje de señas ecuatoriano “Gabriel Román”, dicho proyecto será llevado a cabo en el Centro de Educación Integral (CEI) “Melvin Jones” efectuando el uso de MediaPipe hands siendo este un recurso importante que se encargará de detectar los puntos de referencia de la mano de la seña que una persona realice para posteriormente traducirlas.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1 ANTECEDENTES

Las personas que padecen de problemas auditivos y limitaciones del habla son dependientes del lenguaje de señas para poder manifestar sus ideas, sentimientos y carencias, esta manera de comunicación requiere que el usuario realice movimientos de manos y ademanes con los que logra comunicar mensajes de una forma eficaz, a pesar de ello y debido a la deficiente información que tiene la población oyente acerca de este lenguaje provoca que se restrinjan las posibilidades de interrelación y entendimiento mutuo [1].

En un análisis a la bibliografía de la hipoacusia nos demuestra pequeños rastros de la poca escolarización de estudiantes con capacidades diferentes, esto posiblemente fue causado debido a que en la época antigua primaba el rechazo, desamparo e infanticidio para aquellas personas que para otras eran considerados como “defectuosos”, las personas que nacían con incapacidad en alguna parte de su cuerpo eran juzgadas como una carga para sus familias y castigados a vivir absolutamente sin ninguna manera de poder desarrollarse e incorporarse a la sociedad [2].

Mediante la entrevista realizada a la directora de la Institución (ver anexo 1) los principales problemas que afectan el aprendizaje de los estudiantes es que debido a las limitaciones económicas que tiene la institución no pueden contratar al personal adecuado en cada área, es por ese motivo que no es posible atender a todo tipo de discapacidad que tienen los estudiantes, dejando sin atención a personas con discapacidades consideradas severas como el autismo.

La limitación de medios económicos no solo provoca el deterioro del establecimiento, sino que también se ven afectados el desarrollo de aprendizaje tanto individual como colectivo de los estudiantes, limitando los recursos y herramientas con la que los discapacitados realizan sus tratamientos, ocasionando de esta manera un retraso en el aprendizaje académico.

Otro factor que incide en el progreso de readaptación y formación académica del colegiado es la inadecuada colaboración de los padres dentro de la institución, ya que, se exige que tanto los familiares como instructor a cargo estén presentes en cada terapia que recibe el alumno, una mala relación entre ambas partes en la planificación y seguimientos que estos deben mantener para visualizar el progreso educativo ocasiona una falta de comunicación y comprensión mutua con respecto a las necesidades y desafíos que tiene cada uno de los estudiantes.

La relación de estos problemas causa que la disminución del rendimiento académico del alumnado con discapacidades auditivas sea evidente, al no poder contar con las herramientas suficientes el alumno comienza a sentir frustración, desmotivación en el entorno escolar, baja autoestima y pérdida de confianza en sus habilidades y capacidades para el desarrollo de sus actividades.

También se realizó una encuesta (ver anexo 2) a 9 docentes que imparten sus enseñanzas en el Centro de Educación Integral “Melvin Jones” con la finalidad de tener información acerca de si hacen uso de aplicaciones educativas con las que se pueden visualizar los siguientes puntos fundamentales: el 100% usa un dispositivo móvil, el 78% de ellos usa aplicaciones educativas en su teléfono móvil, el 62% usa una aplicación educativa todos los días.

También se obtuvo que solo el 11% de las personas encuestadas ha usado una aplicación educativa para aprender el alfabeto del lenguaje de señas, el 67% de las concluyó que una ventaja que se tendría con el uso de una aplicación educativa para aprender el alfabeto de lenguas de señas ecuatoriana es la disponibilidad de tiempo y el 68% determinó que prefería hacer uso de la aplicación en la noche.

En la ciudad de Ambato se llevó a cabo una investigación con el tema “sistema automático de decodificación de lenguaje de señas ecuatoriano a comunicación verbal”, basándose en el diccionario de lenguaje de señas ecuatoriano “Gabriel Román” con el que se puede determinar el vocabulario básico que se adaptará al sistema, se elaboró un guante electrónico en la mano derecha que contenían cinco sensores Flex, un sensor inercial MPU6050 y una placa Arduino Nano, estos son usados para efectuar el reconocimiento del movimiento que realizan las manos y los dedos [3].

El proceso de los datos es realizado en Python para posteriormente crear la base de datos, teniendo como objetivo principal el de adaptar algoritmos de machine learning y hacer uso de ellos. Los algoritmos que abarcaron el estudio del proyecto son k-Nearest Neighbors (KNN) y Random Forest, teniendo una precisión del 94,62% y 96,20% correlativamente mejorando de esta manera con los resultados obtenidos en anteriores casos [3].

En Cuba a través de la investigación con el tema “App Android mediante el uso de la realidad aumentada para guía de Lengua de Señas Cubanas” en el año de 2022, propone desarrollar una aplicación móvil Android con el objetivo de ayudar y mejorar el aprendizaje en personas con discapacidades auditivas, a través de la Realidad Aumentada en dispositivos Android y el análisis de las características fundamentales de los elementos que deben ser parte del sistema [4].

Mediante esta investigación y posteriormente desarrollo del tema se dio como resultado la elaboración de una aplicación móvil capaz de aportar al aprendizaje del alumnado que presentaba problemas de discapacidad auditiva, siendo principal soporte el Manual de la Lengua de Señas Cubana LSC, el uso principal de las Tecnologías de la Información y las Comunicaciones haciendo posible la elaboración del proyecto [4].

En Colombia mediante el tema de investigación “Desarrollo de una aplicación en dispositivos móviles para interprete de lenguaje de señas a través del uso de servicios web” en el año 2022, propone como solución y debido al gran cantidad de personas que hacen uso de estos dispositivos, el desarrollo de una aplicación móvil que facilite el procesamiento de comunicación entre las personas a través de la traducción del LSC y por medio de este poder comunicarse con otras personas [5].

El desarrollo de este proyecto llevo a cabo el uso de algoritmos en Python, utilizando la metodología Scrum para el desarrollo de la aplicación, Flutter como framework de desarrollo e implementación de una API para catalogar las señas, transmitiendo los datos con la aplicación a través de microservicios, este proyecto busca mejorar la comunicación que tienen las personas sordomudas con la sociedad colombiana permitiéndoles tener una mejor interacción e inclusión en el entorno social [5].

1.2 DESCRIPCIÓN DEL PROYECTO

Este proyecto tiene como finalidad desarrollar una app en dispositivos Android para el CEI “Melvin Jones” , siendo capaz de poder ingresar patrones de seña de una mano y guardarlas en el servidor, procesar las imágenes guardadas para entrenar el modelo, posteriormente detectar el patrón guardado y traducirlo a texto mostrando su significado en pantalla, de esta forma integrar una herramienta en la que se pueda registrar patrones, poder ver los patrones guardados, practicarlos con la función para traducir que emitirá una respuesta de acuerdo con la postura de mano detectada para una seña y finalmente poder aprenderlos.

La aplicación móvil de traducción de lenguaje de señas constará de los siguientes módulos: módulo de inicio de sesión, módulo de registro, módulo de registro de patrones, módulo de detección de gestos, módulo de procesamiento de imágenes, módulo de traducción de lenguaje de señas y módulo de pruebas.

Módulo de inicio de sesión

- Ingreso de credenciales de usuario
- Identificación de usuario (admin usuario normal)
- Validación de datos
- Iniciar sesión
- Ingresar un nuevo usuario

Módulo de registro

- Ingresar datos del usuario
- Validar datos
- Validar campos
- Registrar un nuevo usuario
- Almacenar datos

Módulo de registro de patrones

- Ingresar como administrador
- Ingresar un nuevo patrón
- Ingresar el número de patrones
- Validar el número de patrones a ingresar

- Ingresar el tamaño del dataset
- Validar el tamaño del dataset
- Generar nombre para el patrón ingresado
- Ingresar el nombre para el nuevo patrón
- Validar el nombre del patrón ingresado
- Iniciar captura

Módulo de detección de gestos.

- Detección de la ubicación del eje de referencia de la mano percibida.
- Detección de palma.
- Detección de los puntos de referencia de las manos
- Reconocimiento de movimientos realizado por la mano.
- Detección del gesto.

Módulo de procesamiento de imágenes

- Captura de muestras para la recolección de datos
- Procesar el movimiento realizado con la mano.
- Extracción de características.
- Guardar y actualizar las imágenes en el Dataset
- Clasificar las imágenes tomadas

Módulo de traducción de lenguaje de señas

- Iniciar traducción
- La aplicación móvil procesa la imagen captada y detecta el de movimiento de mano.
- La aplicación móvil reconoce una seña de mano y la traduce.
- La aplicación móvil detecta la seña y la compara de acuerdo con los patrones guardados en el dataset.
- La aplicación móvil devuelve un frame con los puntos de referencia de las manos.
- La aplicación móvil devuelve un frame con la traducción de acuerdo a la postura de seña realizada.
- La aplicación móvil realiza una interpretación de señas de mano a texto.

Módulo de pruebas

- Realizar pruebas de uso
- Validar el correcto funcionamiento de la aplicación.
- Realizar pruebas de rendimiento y optimización.

Para el desarrollo del proyecto se llevará a cabo el cumplimiento de siete módulos establecidos.

Este proyecto contribuirá a la línea de investigación de Desarrollo de Software, dado que está relacionado con la elaboración de una aplicación móvil y a su vez estará vinculada con la sublínea de Desarrollo de algoritmos y visión artificial haciendo uso de librerías de visión por computador y aprendizaje automático que permitirán el desarrollo del algoritmo de reconocimiento de señas para la elaboración de una herramienta que beneficiarán a las personas con discapacidad auditiva a poder practicar señas y comunicarse de una mejor forma con la sociedad [6].

1.3 OBJETIVOS DEL PROYECTO

1.3.1 OBJETIVO GENERAL

Desarrollar una aplicación móvil que permita ingresar patrones de señas estáticas de una mano mediante el uso de la librería MediaPipe Solutions, de esta forma crear una herramienta donde el usuario sea capaz de visualizar una seña, aprenderla y luego poder traducirla, convirtiéndose en un medio de comunicación y aprendizaje para personas con problemas auditivos en el CEI “Melvin Jones”.

1.3.2 OBJETIVOS ESPECÍFICOS

- Diseñar un algoritmo que permita detectar y traducir lenguaje de señas mediante el uso de la solución MediaPipe Hands.
- Elaborar una API que posibilite la comunicación bidireccional entre la aplicación móvil y la información almacenada en el servidor y la base de datos.
- Desarrollar una aplicación móvil que pueda hacer uso del algoritmo de detección de lenguaje de señas ecuatoriano.

1.4 JUSTIFICACIÓN

Las personas que poseen deficiencia auditiva, como una forma de solventar la dificultad de relacionarse con su entorno se ven obligados a buscar una manera de poder comunicarse con las demás personas de su alrededor, una de estas maneras de comunicación es mediante el lenguaje de señas que se representa a través del movimiento de las manos, de esta forma poder realizar una comunicación con resultados positivos entre una persona con problemas auditivos y un oyente [7].

Es importante dar a conocer que no existe una lengua de signos universal, sin embargo, dentro de nuestro país existe el diccionario de Lengua de Señas Ecuatoriano “Gabriel Román” (LSEC), en su formato web cuenta con señas de palabras y su significado, contiene gráficos y videos multimedia en la que se explica la manera correcta de articular una seña, el LSEC contiene su propio léxico, estructura gramatical y características que se adaptan a las diferentes regiones del país [8].

El uso de recursos tecnológicos para mejorar el modelo de aprendizaje en instituciones es fundamental para el desarrollo de la preparación académica estudiantil [9] uno de los principales usos de la tecnología es que a través de ella se puede facilitar la forma de enseñar y aprender el LSEC mediante la creación de un software o una aplicación móvil y a partir de estos poder mantener una correcta comunicación entre personas con problemas auditivos sin la necesidad de tener un intérprete a su lado [10].

La finalidad de este proyecto es el de desarrollar una app que se usará en dispositivos Android para el CEI “Melvin Jones” que sirva como método de aprendizaje del alfabeto del lenguaje de señas ecuatoriano, de esta forma poder integrar una herramienta de aprendizaje, la aplicación móvil detectará el movimiento de la mano de una persona y hará un reconocimiento de la imagen percibida de la seña con el que está establecido en el diccionario de LSEC y a su vez esta emitirá una imagen con la seña detectada y el texto del significado de la seña captada por el dispositivo.

La implementación de esta app para dispositivos Android permitirá que los docentes de la institución puedan hacer uso de ella como herramienta de enseñanza

del alfabeto del lenguaje de señas ecuatoriano, el convertir la forma de impartir y ofrecer un aprendizaje más dinámico, en este caso un docente a la hora de querer enseñar nuevas palabras del alfabeto de lenguaje de señas podría hacerlo mediante la aplicación.

También a ayudar en la mejora del aprendizaje académico, realizar grupos de estudiantes y mediante la aplicación poder comunicarse y aprender palabras entre sí, durante esta actividad cada estudiante es capaz de aprender y poder relacionarse con otras personas, esta aplicación promueve la traducción del lenguaje de señas, además de que sirve como una estrategia pedagógica que puede ser usada por los tutores con el fin de mejorar y facilitar el aprendizaje de los estudiantes a través de técnicas y herramientas que puedan contribuir con el mejoramiento académico de cada uno de ellos.

Aumentar los instrumentos y recursos que un docente posea para evolucionar sus métodos de enseñanza, mantener una mejor relación de los estudiantes con problemas auditivos y su entorno, así como también evitar la disminución del rendimiento académico e impulsar a que cada estudiante tenga más confianza fortaleciendo las habilidades y capacidades que el alumnado posee.

Este proyecto en relación con su aporte de desarrollo a los objetivos del plan de creación de Oportunidades del año 2021-2025, se encuentra fundamentado en los siguientes ejes:

Eje económico

Objetivo 4: Garantizar la gestión de las finanzas públicas de manera sostenible y transparente [11].

Eje social

Política 5.5: Mejorar la conectividad digital y el acceso a nuevas tecnologías de la población [11].

Lineamientos territoriales

Pol 5.1.

C3. Promover programas de protección social no contributiva en las provincias de mayor incidencia de pobreza y desnutrición [11].

Pol 5.4.

A4. Fortalecer la conectividad y el acceso a las TIC como vía para mejorar el acceso a otros servicios [11].

Con base a todo lo expuesto anteriormente se pretende realizar una aplicación en dispositivos Android para el CEI “Melvin Jones” que sea capaz de reconocer el alfabeto del lenguaje de señas ecuatoriano y poder traducir los gestos estáticos realizados con una mano, además de tener una opción integrada que permite al usuario administrador ingresar más patrones y de esta forma incrementar el aprendizaje de la aplicación con respecto a señas registradas, haciendo que los estudiantes puedan visualizar y aprender nuevos patrones mediante el uso de la aplicación.

1.5 ALCANCE DEL PROYECTO

Con la culminación del proyecto se pretende conseguir una app para dispositivos Android que sirva como un traductor del alfabeto de señas a palabras basado en el lenguaje de señas ecuatoriano “Gabriel Román”, dicho proyecto será implementado en el CEI “Melvin Jones” con el uso de MediaPipe Solutions que se encargará de reconocer los puntos de referencia que una persona realice con la mano para posteriormente traducirlas.

Debido a que el proyecto está basado en el formato web del vocabulario LSEC “Gabriel Román” y al contar con cerca de 5000 palabras a disposición de los usuarios, del sitio web oficial en la sección de “palabras más usadas” se seleccionará y limitará a escoger las palabras que estén más acordes en la comunicación de personas, es necesario que para realizar la traducción las señas sean estáticas y realizadas con una sola mano [12].

Para el desarrollo del proyecto se llevará a cabo el cumplimiento de siete módulos establecidos.

El primer módulo dará acceso al ingreso de las credenciales del usuario, se validarán los datos en caso el usuario ya se haya registrado con anterioridad para

posteriormente iniciar sesión en la aplicación, se validará el tipo de usuario, si el usuario que inició sesión es administrador se redirigirá a la interfaz administrador con todas las funciones disponibles, por el contrario si es un usuario normal se le redirigirá a la interfaz para que pueda hacer uso de la función para ver los patrones disponibles y traducir señas, si los datos ingresados son incorrectos se pedirá que ingrese datos válidos, en la pantalla de inicio de sesión también se incluirá un botón para acceder al menú de registro y agregar un nuevo usuario.

En el módulo de registro se ingresarán los datos de un nuevo usuario para que tenga acceso a la aplicación y obtener las credenciales para que pueda iniciar sesión, los datos ingresados del nuevo usuario serán guardados y almacenados en la base de datos para evitar que el usuario tenga problemas al momento de iniciar sesión.

Para ingresar al módulo donde se agrega un nuevo patrón el usuario que inicio sesión debe ser validado como administrador, este módulo contendrá las funciones para ingresar un nuevo patrón y validar que el número ingresado este permitido así como también se ingresará el tamaño de datos que se capturará para el patrón que se agregará, se generará el nombre para el patrón y se verificará que el nombre puesto no sea igual a uno que ya exista en el servidor, una vez validado se procede a iniciar la captura de datos para guardarlos en el dataset.

En el módulo donde se detecta la gesticulación se clasificarán los ademanes hechos por la palma de acuerdo con las articulaciones realizadas por el movimiento de manos, se detectará el eje de referencia de la seña captada, así como también la posición de la palma y los puntos de referencia, de esta forma reconocer los movimientos frecuentes y clasificarlos acorde a los movimientos que realicen las manos.

En el módulo de tratamiento de capturas se llevará a cabo la recopilación de información y se procesará el movimiento realizado por la mano, se extraerán las características mediante el uso de MediaPipe Solutions, se clasificarán las muestras tomadas y en cada una de las clases se almacenará una cantidad de datos del gesto capturado, de esta forma la aplicación mediante el algoritmo de reconocimiento de seña creado y el uso de aprendizaje automático memorizará las posturas, las guardará en el dataset y lo actualizará, para que finalmente puedan ser clasificadas.

Luego de finalizar con el proceso de detección de gestos y procesamiento de imágenes comienza el módulo de traducción de lenguaje de seña, este módulo llevará a cabo la detección de la imagen de una seña, esto se realizará con el uso de MediaPipe Hands, luego de que la seña sea reconocida por la aplicación móvil se procesará y detectará la imagen con la señal captada para posteriormente emitir un frame con el texto del significado de la seña traducida.

Una vez culminado todo el proceso de detección de gestos, procesamiento de imágenes y traducción de lenguaje de seña, se entra en el último módulo donde se realizan las etapas de pruebas para determinar el correcto funcionamiento de cada una de las características de la aplicación, así como su rendimiento y optimización, estas pruebas serán realizadas por cada elemento de manera individual, así como también el funcionamiento de la aplicación de forma completa.

1.6 METODOLOGÍA DEL PROYECTO

1.6.1 METODOLOGÍA DE LA INVESTIGACIÓN

Debido a la escasa información que se tiene sobre el uso de las aplicaciones móviles para la traducción de señas en personas con problemas auditivos, se utilizará la metodología de investigación de tipo exploratoria [13]. Se realizará una revisión bibliográfica con el objetivo de recolectar información de trabajos de nivel local, así como también internacional y de esta forma poder tener una mejor visión con respecto a los resultados del proyecto.

La investigación es considerado de carácter experimental debido a que el proyecto implica la creación y aplicación de una tecnología específica, en este caso al creación de una app mediante el uso de MediaPipe Hands para observar cómo afecta o mejora la comunicación de los estudiantes con problemas auditivos, la naturaleza experimental se debe a esta intervención directa, donde se manipula una variable que es la introducción de una herramienta de traducción de lengua de señas para observar el efecto que produce [14].

Dado que el tipo de investigación es experimental, donde se desarrolla y prueba una aplicación móvil para traducir el lenguaje de señas a texto, el alcance de la investigación será clasificado como Explicativo, con un enfoque experimental, se está manipulando una variable para observar cómo afecta o mejora la comunicación

de los estudiantes con problemas auditivos y entender si la introducción de esta tecnología causa cambios en la forma en que se comunican los estudiantes y si mejora su experiencia comunicativa, por lo tanto se busca explicar cómo y por qué la aplicación puede tener un impacto en la comunicación de los usuarios [15].

Con el propósito de conocer el estado actual de la institución, así como también los problemas que afecta en la misma se pretende realizar un análisis dentro del CEI “Melvin Jones” aplicando la investigación de tipo cualitativa [16], haciendo uso de la entrevista como método de recolección de información y de esta forma conocer más detalles de las necesidades con respecto a las mejoras del proyecto.

Puesto que se ha optado por elegir el tipo de investigación cualitativa para el desarrollo del proyecto y realizar una entrevista a la fundación para conocer más el estado de la institución, el método de investigación a utilizar será el inductivo ya que es un método que se utiliza comúnmente en la investigación cualitativa, este método se encarga en generar nuevas teorías e hipótesis a partir de los datos obtenidos, en este caso se hace uso de la entrevista para identificar temas, necesidades o problemas sin presuponer resultados específicos [17].

1.6.2 BENEFICIARIOS DEL PROYECTO

Mediante la recolección de datos y posteriormente análisis de la información se determinó los beneficiarios del proyecto divididos en categorías: beneficiarios directos e indirectos.

A continuación, se describirá los detalles mediante una tabla:

BENEFICIARIOS	TIPO	CANTIDAD
B. DIRECTOS	DOCENTES	9
B. INDIRECTOS	Comunidad estudiantil	

Tabla 1 Beneficiarios del Proyecto

Se determinó que el personal docente, compuesto por 9 elementos, constituye la población de interés para este proyecto, esta población representa el grupo de personas con las que se explorará los resultados que la aplicación demuestre, permitiendo comprender cómo la herramienta afecta su comunicación y experiencia educativa. La elección de este grupo específico se basa en su relevancia directa para

el ámbito educativo y su importancia en el proceso de enseñanza-aprendizaje para los estudiantes con problemas auditivos.

Población	
Tipo	Cantidad
Docentes	9

Tabla 2 Población del proyecto.

1.6.3 VARIABLE

Determinar si la introducción y el uso de una aplicación móvil de traducción de lenguaje de señas a textos en un centro de educación integral sirve como herramienta de enseñanza para mejorar significativamente la comunicación y aprendizaje de acuerdo con funciones establecidas con las que el usuario pueda observar patrones y luego traducirlas mediante el traductor de la aplicación, creando una interacción más fluida y efectiva en entornos educativos y sociales.

1.6.4 ANÁLISIS DE RECOLECCIÓN DE DATOS

Se utilizó la entrevista como técnica de recolección de información, dado que es una técnica que permite obtener información de manera sencilla e importante para definir avances significativos en el proyecto, se realizó una serie de preguntas dirigidas a la rectora de la institución con el propósito de conocer las condiciones actuales dentro del CEI “Melvin Jones”.

En la entrevista (anexo 1) se realizó para conocer cuáles son los servicios y que tipo de tratamientos ofrece la institución a personas que presenten problemas auditivos, debido a que la institución cuenta con personal voluntario también se consultó si ellos reciben algún tipo de capacitación para poder tratar con los niños del establecimiento sin presentar problemas.

Una razón por el que se realizó esta entrevista es para conocer cuáles son los problemas que afectan el CEI Melvin Jones, siendo esta una fundación sin fines de lucro que atiende a personas con discapacidades, determinar los problemas y tratar de dar una solución acorde a las necesidades y características de la institución.

Otro detalle relevante en el que se hizo énfasis es el de conocer si la fundación hace uso de la tecnología para mejorar la comunicación de los estudiantes con problemas auditivos, si emplean la tecnología como herramienta de enseñanza, así como también consultar si conocen cuáles son los beneficios que tiene la tecnología para mejorar la comunicación lenguaje de señas entre personas.

También se realizó una encuesta (anexo 2) que constará de 10 preguntas de selección con la que se podrá obtener resultados de si los encuestados hacen uso de aplicaciones educativas, así como también de aplicaciones para aprender lenguaje de señas y qué ventajas tiene con el uso de la app presentando graficas con porcentajes de cada una de las preguntas.

En base a la información recopilada en la entrevista y encuesta se puede concluir que una forma de implementar la tecnología utilizando bajos recursos y sea posible usarlo como herramienta que se pueda usar en cualquier momento del día y en cualquier lugar para realizar la traducción del alfabeto del lenguaje de señas ecuatoriano y mejorar la comunicación de los estudiantes, así como también el aprendizaje de la lengua de señas ecuatoriano es factible realizarlo mediante la aplicación móvil.

1.7 METODOLOGÍA DE DESARROLLO

El presente proyecto tiene como finalidad el de desarrollar una aplicación móvil, para llevar a cabo una correcta planificación y desarrollo es necesario definir una metodología, en este caso el desarrollo de proyecto se hará bajo la metodología incremental, el cual tiene como propósito realizar pequeños ciclos de forma iterativa, dicho en otras palabras a cada nuevo ciclo se le agregaran nuevos incrementos siendo así que por cada incremento se tendrá un prototipo de la aplicación [18], esta metodología se divide en las siguientes fases:

Fase de análisis: En esta fase se define los requerimientos y los recursos necesarios para la elaboración del proyecto en base a la investigación y datos recolectados en el CEI “Melvin Jones”.

Fase de diseño: Luego de determinar los requerimientos obtenidos en la fase previa, se procede a generar la arquitectura del proyecto, se elabora la interfaz de la aplicación y se muestra los resultados mediante diagramas.

Fase de codificación: Se desarrolla las interfaces bajo los criterios establecidos en la fase anterior, se llevan a cabo pruebas para comprobar que cada característica funcione correctamente.

Fase de pruebas: Una vez culminado las fases previas se llega a la fase final en donde se obtiene una versión operativa del sistema y se realizan las pruebas de cada uno de los módulos por separado, luego se integra cada uno de los módulos y se prueba su funcionalidad en conjunto, de esta forma poder garantizar que el sistema entero opera de forma correcta.

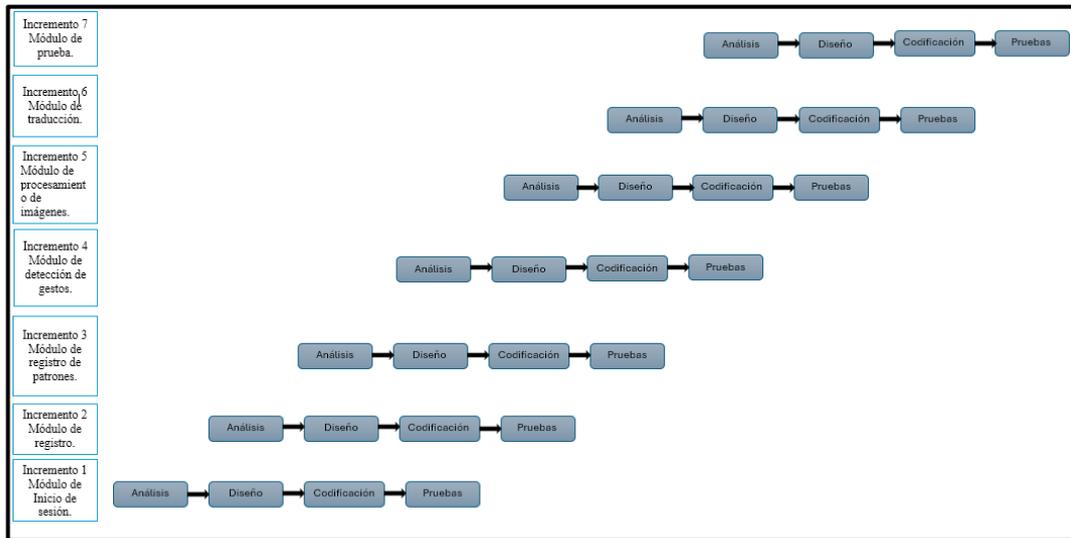


Fig. 1. Fases de la metodología incremental

CAPITULO II

1. PROPUESTA

2.1 MARCO CONTEXTUAL

El CEI "Melvin Jones" está situado en el cantón de La Libertad Barrio 28 de mayo Av. 16 entre las calles 13 y 14 en la provincia de Santa Elena, surge con el único propósito de ayudar a niños, niñas y jóvenes vulnerables de bajos recursos en la provincia de Santa Elena que sufren de discapacidades especiales, ofreciendo una enseñanza integra de acompañamiento escolar de calidad [19].

En el desarrollo de mejorar e impulsar las capacidades cognitivas de los niños, niñas y jóvenes el CEI “Melvin Jones” ofrece diversos métodos y servicios de enseñanza tales como: rehabilitación física, lenguaje, psicología, refuerzo escolar, área educativa y talleres, promoviendo la inclusión y usando métodos multimodal, lectura labial, braille, Abaco y lenguaje de señas estimulando y mejorando las habilidades individuales de los estudiantes [20].

Misión

“Brindar atención integral de calidad a niños, niñas y jóvenes con o sin discapacidad y grupo familiar a fin de lograr una adecuada integración en la comunidad. Contribuir al desarrollo armónico de la persona para mejorar su calidad de vida y su relación con el entorno social y laboral” [20].

Ubicación Sectorial



Fig. 2. Mapa de ubicación: CENTRO DE EDUCACIÓN INTEGRAL MELVIN JONES.

2.2 MARCO CONCEPTUAL

2.2.1 Visión artificial

La visión artificial, a menudo referida por algunos autores como visión por computadora, se caracteriza por ser la disciplina científica que se enfoca en programar a las computadoras, darles la capacidad para procesar, interpretar y entender imágenes o secuencias de video, este campo abarca la conversión de datos visuales capturados por cámaras o sensores en fotogramas, transformándolos en decisiones informadas o en representaciones alternativas más útiles para aplicaciones específicas, tal proceso no solo incluye la identificación de patrones o características en las imágenes, sino también la comprensión de su contexto y significado, lo que permite una amplia gama de aplicaciones, desde la automatización industrial hasta la mejora de interfaces de usuario interactivas [21].

2.2.2 Inteligencia Artificial

La Inteligencia Artificial (IA), se considera un área destacada dentro de las ciencias de la computación, ha captado un gran interés actualmente debido a sus amplias aplicaciones, la exploración de métodos para entender la inteligencia y desarrollar prototipos ha inspirado a muchos investigadores a dedicarse por completo a este ámbito. El desarrollo inicial del concepto de IA se atribuye a la visión del matemático británico Alan Turing, mientras que el término "Inteligencia Artificial" fue atribuido por McCarthy durante una conferencia en el Dartmouth College en Estados Unidos, donde se exploró la posibilidad de producir dispositivos inteligentes [22].

2.2.3 Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial, donde un programa de computadora se entrena con un conjunto de datos establecidos para localizar patrones y realizar predicciones sobre datos futuros [23]. El entrenamiento que se realiza depende de una serie de datos que sirven para que el computador mediante el programa realice el proceso de aprendizaje automático, en este punto es donde la maquina ciertamente aprende y memoriza patrones, el tiempo de entrenamiento va a depender de que tan complejo es el algoritmo además de

consumir recursos informáticos considerables con la finalidad de obtener los resultados deseados [24].

Existen tres tipos de aprendizaje automático:

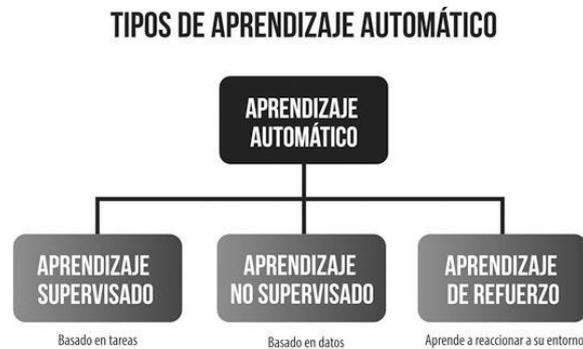


Fig. 3 Tipos de aprendizaje automático [1]

2.2.3.1 Aprendizaje supervisado

En el método de aprendizaje supervisado, los algoritmos se adiestran utilizando datos que previamente han sido clasificados o catalogados para mostrar cómo deberían ser categorizados nuevos datos. Este proceso implica una participación del ser humano, quien debe dar retroalimentación al sistema. Por ejemplo, se le mostrarían al algoritmo imágenes de gatos de antemano para que aprenda a reconocer otras fotos similares en el futuro [25].

2.2.3.2 Aprendizaje no supervisado

En el aprendizaje no supervisado, los algoritmos operan sin el uso de datos previamente etiquetados o clasificados para guiar la categorización de nueva información. En cambio, deben determinar por sí mismos cómo estructurar los datos. Así, este enfoque elimina la necesidad de intervención humana. Por ejemplo, los algoritmos son los que tendrían que identificar y clasificar por cuenta propia las fotos que contienen gatos en una categoría específica [26].

2.2.3.3 Aprendizaje reforzado

Finalmente, en el aprendizaje reforzado, los algoritmos se desarrollan a través de la experiencia acumulada. Esto significa que deben recibir "refuerzos positivos" cada vez que realizan una acción correctamente. Este proceso de aprendizaje es similar

al entrenamiento de perros, donde se les ofrece una "recompensa" cada vez que ejecutan correctamente un comando, como sentarse [27].

2.2.4 Backend

Se conoce como backend a los datos e infraestructura que hacen posible que una aplicación funcione de manera correcta, almacena y procesa los datos mediante solicitudes http, luego de procesar los datos devuelve la respuesta a la aplicación, al momento de realizar una petición usa protocolos así como también tecnologías de comunicación para cumplir con la solicitud, el backend es conocido como servidor, este puede estar desarrollado en diferentes lenguajes de programación de acuerdo con el gusto del programador [28].

2.2.5 Python

Es un lenguaje de programación que da la facilidad de elaborar aceleradamente e incorporar sistemas de una forma más competente, una de sus características es el uso de menos líneas de códigos comparados con otros lenguajes de programación, posee múltiples librerías para el desarrollo de aplicaciones, es de los lenguajes de programación más usados para desarrollar aplicaciones de aprendizaje automático debido a la gran cantidad de librerías que posee [29].

2.2.6 API

La interfaz de programación de aplicación también conocida como API por sus siglas en inglés es una serie de especificaciones y protocolos que son usados para esquematizar e incorporar el software a las aplicaciones, las APIS hacen posible la comunicación entre dos aplicaciones, funciona como un contrato donde si en un lado se realiza una solicitud de una determinada estructura, esa misma estructura decidirá como será la respuesta de la aplicación y que se mostrará del otro lado [30].

2.2.7 Visual Studio Code

Es un editor de código fuente creado por Microsoft, muy popular entre los programadores por su flexibilidad y potencia, admite numerosos lenguajes de programación y cuenta con una gran variedad de extensiones que amplían sus capacidades, su interfaz de fácil manejo y configurable permite a los

desarrolladores escribir y, depurar código de forma efectiva convirtiéndose en una herramienta indispensable para el desarrollo de software. [31].

2.2.8 phpMyAdmin

Es un programa de libre acceso basado en PHP que posibilita la administración de un database en MySQL mediante una interfaz web. Esta herramienta posibilita a los usuarios realizar operaciones como crear, editar y borrar bases de datos, tablas y registros, ejecutar consultas SQL, y administrar los permisos de los usuarios. Gracias a su diseño intuitivo, es una opción accesible y eficaz tanto para desarrolladores novatos como para expertos que necesitan interactuar con MySQL de manera sencilla [32].

2.2.9 XAMPP

Es un paquete de software libre que proporciona una solución completa para desarrolladores que necesitan un entorno de servidor web local. Incluye Apache (servidor web), MySQL (base de datos), y soporte para PHP y Perl, permitiendo a los usuarios instalar y configurar estos componentes de manera sencilla en sus computadoras, es ideal para pruebas y desarrollo, XAMPP facilita la creación y gestión de aplicaciones web sin necesidad de un servidor remoto, ofreciendo un entorno integrado y fácil de usar [33].

2.2.10 Frontend

Es la parte visible con la que el usuario interactuará, esta puede ser la interfaz gráfica de una página web, programa o aplicación móvil, el objetivo que tiene el frontend es que sea intuitivo y llamativo de fácil entendimiento para el manejo del usuario, colores, botones entre otros, el frontend mediante la interfaz gráfica hace peticiones a la API donde envía solicitudes para realizar procesos del backend y mostrar un resultado, para hacer uso de un frontend el necesario utilizar un framework de acuerdo a la plataforma en la que se está desarrollando [34].

2.2.11 Framework

Es un entorno de trabajo para el desarrollo de aplicaciones a una plataforma en específico estas pueden de plataforma web, aplicación móvil u administración de

base de datos, los frameworks de software suelen ser desarrollados, evaluados y mejorados por equipos de ingenieros y programadores experimentados, son herramientas versátiles, robustas y eficientes. Utilizar un framework de software para crear aplicaciones permite enfocarse en las funcionalidades avanzadas de la aplicación, ya que el framework gestiona las tareas de bajo nivel [35].

2.2.12 Flutter

Flutter es un framework de código abierto creado por Google que permite desarrollar aplicaciones multiplataforma de gran calidad visual, compiladas de manera nativa a partir de un único código fuente, flutter usa lenguaje de tipo dart para el desarrollo del código, es usado para crear interfaces únicas y llamativas tanto para dispositivos móviles, web de escritorios e integrados [36].

2.2.13 Lenguaje de Señas

El lenguaje de signos, caracterizada por su naturaleza gesto-espacial y de percepción visual, es fundamental para la comunidad con problemas auditivos, no solo como un medio de comunicación sin barreras, sino también como un elemento clave de su identidad y herencia cultural y, además contribuye significativamente al desarrollo lingüístico y cognitivo [37].

Es el principal medio de comunicación utilizado por la comunidad hipoacúsica por ende se puede determinar al lenguaje de signos como una parte fundamental para los seres humanos debido a que sin ella una gran parte de la población quedaría excluida sin poder comunicarse, esta es adquirido de manera innata y convertido en una lengua esencial, funcional y valiosa con el que se pueden comunicar con las personas de su entorno [38].

2.2.14 OpenCv

Es una biblioteca de software de libre acceso enfocada en la visión por computadora y el aprendizaje automático, ofrece herramientas para procesar imágenes y videos, permitiendo tareas como detección de rostros, reconocimiento de objetos y análisis en tiempo real. Está escrita en C++ y es compatible con varios lenguajes de programación entre ellas Java y Python, facilitando su uso en distintos proyectos y plataformas [39].

2.2.15 MediaPipe

Facilita la implementación de soluciones de aprendizaje automático (AA) en aplicaciones, pone a disposición un framework para poder formar conductos de procesamientos, una de las principales características de esta librería es la detección de objetos [40]. MediaPipe es una plataforma de código abierto desarrollada por Google que simplifica la creación de aplicaciones de visión computarizada en tiempo real.

Ofrece una variedad de algoritmos y modelos ya entrenados que permiten a los desarrolladores utilizar la inteligencia artificial para funciones como el seguimiento de objetos, la identificación de gestos, el reconocimiento facial y el monitoreo de movimientos de las manos, entre otros. Su popularidad se debe a su facilidad de uso, capacidad de escalado y eficacia en el procesamiento en tiempo real.

2.2.15.1 MediaPipe Hands

Es una Solución en el que se realiza la persecución y detección de los movimientos de las manos en tiempo real a partir del uso de un dispositivo, este procedimiento consiste en dos métodos: detector de palma y detección de los puntos de referencia de la mano, para su implementación se utiliza MediaPipe, una plataforma de aprendizaje automático compatible con múltiples sistemas operativos, esta arquitectura asegura una rápida velocidad de inferencia en dispositivos móviles manteniendo al mismo tiempo una predicción precisa y de alta calidad además MediaPipe Hands se ofrece como una solución de código abierto [41].

2.2.15.2 Hand Landmark Model

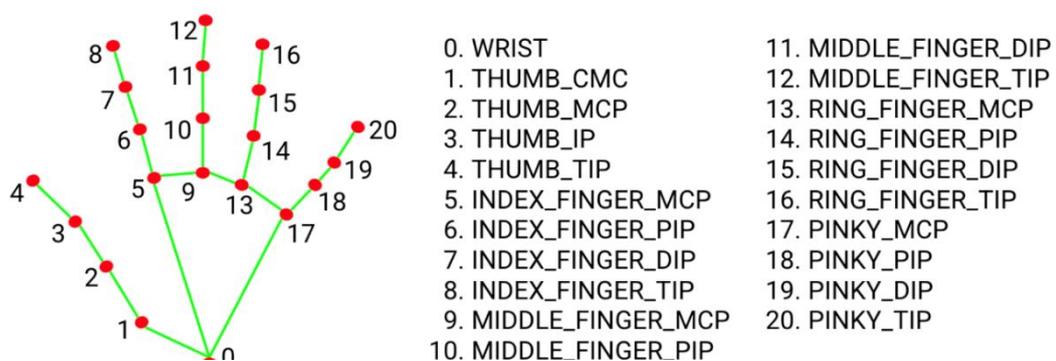


Fig. 4. Hand Landmark Model

Es un modelo de reconocimiento de los puntos de referencia de las manos, este modelo fue entrenado mediante aprendizaje automático con una gran cantidad de imágenes reales logrando que el modelo sea capaz de detectar 21 puntos de referencias de los nudillos de las manos, este modelo puede reconocer los puntos claves de la mano tanto izquierda como derecha [42].

2.2.15.3 Detección de Objetos y cuadros delimitadores

La localización de elementos en una imagen implica tanto ubicar los objetos presentes como clasificarlos. Este proceso realiza la creación de marcos alrededor de cada objeto y les asigna una etiqueta de clase según corresponda. Esto facilita la clasificación de múltiples objetos dentro de una misma imagen, lo que es particularmente útil para buscar imágenes de objetos específicos o para identificar imágenes que incluyen varios objetos específicos en conjunto.

Una de las maneras en las que se realiza la detección de un elemento es mediante la delimitación de cuadros, este se produce debido a que el cuerpo que se desea encontrar se encasilla en una caja ubicado en un plano con las coordenadas (X, Y) encerrando el elemento al que se le asigna ancho y un alto de acuerdo con las dimensiones del elemento para que posteriormente se pueda llevar a cabo el análisis.

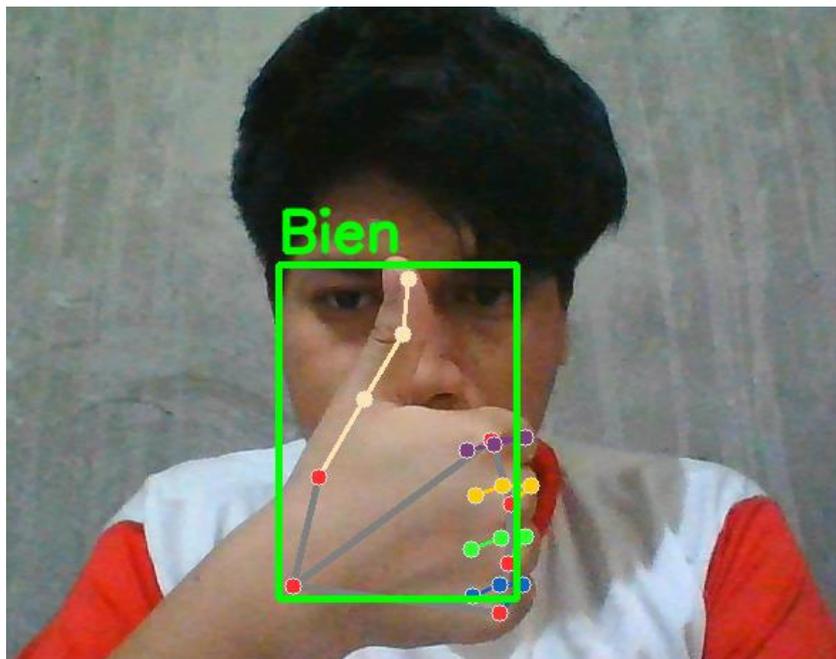


Fig. 5. Cuadro delimitador en el reconocimiento de un objeto

Proyectos de MediaPipe	
Detección de Objetos	Facilita la identificación y localización de diversas clases de objetos en imágenes o vídeos, esta herramienta es capaz de detectar y señalar la posición de perros dentro de una imagen.
Reconocimientos de gestos	Se basa en el reconocimiento de 21 puntos de referencia ubicadas en la palma de una mano, la detección se realiza en tiempo real.
Detección de rostro	Detecta la presencia de rostro con puntos clave de referencia.
Estilización de caras	Estiliza los rostros dentro de una imagen, sirve para crear avatares virtuales.
Malla facial	Realiza una estimación de alrededor de 478 puntos de referencias faciales en 3D.

Tabla 3 Soluciones de MediaPipe

2.3 MARCO TEÓRICO

2.3.1 Uso de procedimientos de aprendizaje computacional y redes profundas para la interpretación de lenguaje de señas

En las últimas dos décadas, el campo del Machine Learning (aprendizaje automático) ha desempeñado un papel crucial en el desarrollo de aplicaciones de software. Esta área de la informática y la Data Science emplea algoritmos y métodos estadísticos para aprender de los datos, extraer inferencias y reconocer patrones sin necesidad de programación explícita. El Deep Learning o aprendizaje profundo es

una subárea del Machine Learning, hace uso de redes neuronales multicapa para simular el complejo tratamiento de información del cerebro humano [43].

La proliferación de datos digitales en diversos campos, como la ciencia, la ingeniería, el internet de las cosas, la biomedicina, la atención sanitaria y muchos sectores comerciales, ha marcado el inicio de la era del big data. Estos datos no pueden ser analizados mediante métodos estadísticos tradicionales y requieren de técnicas avanzadas de ML y DL. Debido a que el aprendizaje automático se basa en el análisis de datos en lugar de en la programación de reglas específicas, se está explorando su uso para crear sistemas capaces de resolver problemas con la misma eficacia que los expertos humanos en diversos campos [43].

2.3.2 Visión computarizada y machine learning para la creación de aplicaciones

La identificación de lenguaje de gestos es un campo de investigación bastante nuevo que tiene como objetivo primordial traducir esta manera de comunicación a texto, este proceso conlleva capturar imágenes de una persona haciendo uso de la lengua de señas e identificar la postura que tiene su cuerpo, cara y manos además de denotar la expresión que forman cada uno de sus sentidos y mediante los datos se debe poder identificar la seña y con esto tener una traducción escrita [44].

Una de las áreas de estudio es la clasificación de forma de las manos, por lo general la lengua de señas utilizan un grupo limitado de formas de mano que, en conjunto con las expresiones faciales y movimientos del cuerpo, son usados para comunicarse a través de señas, según estudios anteriores, una fase esencial para el reconocimiento de la lengua de señas es el proceso de clasificación de la forma de la mano, lo cual es sustancial para aumentar la precisión de la detección [44].

2.3.3 Revisión bibliográfica sobre el uso de la IA en la interpretación de lenguaje de signos

La traducción de lenguaje de gestos debe facilitar una conexión fluida e incluyente, por lo que es crucial hablar de proyectos, programas, métodos y adaptaciones diversas para mejorar el proceso de comunicación entre estos grupos de personas, por un lado, Sultan señalan que los lenguajes de señas más comunes son el Lenguaje

de Señas Americano (ASL), el Lenguaje de Señas Español (SSL), el Lenguaje de Señas Australiano (AUSLAN) y el Lenguaje de Señas Árabe (ASL) [45].

Desde el año 2020, ha habido un auge en el campo de las Ciencias de la Computación enfocado en la traducción del lenguaje de señas. Se concluye que ha habido una evolución exponencial en el uso de redes neuronales para mejorar la eficacia en la traducción de lenguajes de señas, las redes neuronales más utilizadas y eficientes han sido las Redes Neuronales Convolucionales (CNN), que lograron un alto nivel de precisión, con resultados que oscilaron entre el 90% y el 100% de aciertos. Las etapas más aplicadas incluyeron la clasificación de muestras, el preprocesamiento de datos, la aplicación de capas de redes neuronales, el entrenamiento de los modelos de aprendizaje y la evaluación de métricas [45].

2.3.4 Instrucción de la lengua de signos mediante las TI

Hasta ahora, a través de los avances de las ciencias aplicadas destinados a personas con problemas hipoacúsicos han tenido una repercusión significativa, mejorando su calidad de vida, con el continuo progreso de las ciencias cada día surgen más y mejores soluciones que facilitan el desarrollo de nuevos dispositivos de entre ellos están los siguientes ejemplos, la Universidad de East Anglia y la productora británica de software de animación para televisión televirtual, elaboró un software siendo este un intérprete virtual de lenguaje de señas, el software presenta a un personaje animado llamado Guido, encargado de traducir la lengua de signos británica. Hoy en día, el software se halla en períodos de pruebas [46].

Jorge Enrique Leal Rodríguez, un ingeniero colombiano colaborando con el apoyo de la fundación Herramientas Tecnológicas para Ayuda Humanitaria (HETAH) y la empresa Tecnocontrol Supervisa S.A., creó un traductor en línea de lenguas de señas de acceso libre, este software no convierte la voz a lengua de señas, sino que traduce texto (palabras o frases) a lengua de señas utilizando un avatar [46].

2.3.5 Mediapipe: un framework para percibir la existencia y procesarla

El estudio realizado por Camilo Lugaresi establecen las características que ofrece Mediapipe siendo estas el poder desarrollar una aplicación que procese entradas perceptuales implicando más que simplemente ejecutar un modelo de aprendizaje automático, los desarrolladores deben aprovechar las capacidades de una amplia

gama de dispositivos, equilibrar el uso de recursos y la calidad de los resultados, ejecutar múltiples operaciones en paralelo y con canalización, y asegurarse de que los datos en series temporales estén correctamente sincronizados, el marco de trabajo MediaPipe aborda estos desafíos, un desarrollador puede usar MediaPipe para combinar rápida y fácilmente componentes de percepción existentes y nuevos en prototipos, y avanzar hacia aplicaciones multiplataforma pulidas [47].

Con MediaPipe, el desarrollador puede configurar una aplicación para gestionar recursos de manera eficiente tanto CPU como GPU para un rendimiento de baja latencia, manejar la sincronización de datos en series temporales como fotogramas de audio y video, y medir el rendimiento y el consumo de recursos. Estas características permiten al desarrollador centrarse en el desarrollo de algoritmos o modelos, utilizando MediaPipe como un entorno para mejorar la aplicación, con resultados reproducibles en diferentes dispositivos y plataformas [47].

2.3.6 Imaginando un futuro más asequible: uso de la metodología MediaPipe Hands para la elaboración de una aplicación móvil de traducción de lenguaje de señas a voz

En el expediente presentado por Recalde, Melchizedek y otros autores establecen el uso de la solución de mediapipe hands para la elaboración de una aplicación que detecta movimiento de manos denominada SignTech, emplea la solución de MediaPipe para transformar los gestos del alfabeto del lenguaje de señas americano (ASL) en texto y luego en voz. Utilizando el marco MediaPipe Hands Solutions, los programadores mapearon y calcularon los puntos de referencia de las manos para formar cada letra del ASL. Estos puntos de referencia calculados se compararon con los definidos para cada letra del ASL, y se empleó un algoritmo de coincidencia para identificar la mejor correspondencia [48].

Esta solución innovadora rompe barreras de comunicación, permitiendo que las personas sordas se expresen con mayor facilidad y que las personas oyentes comprendan e interactúen con los usuarios de lengua de signos, los desarrolladores llevaron a cabo un estudio de usuarios para evaluar la eficacia y usabilidad de SignTech, cuyos resultados demostraron el éxito en la traducción del lenguaje de señas a la voz, esta investigación ofrece información valiosa sobre el desarrollo de

tecnología para la traducción de lenguaje de señas y destaca el potencial de MediaPipe Hands para mejorar la comunicación entre personas sordas y oyentes a través de una aplicación móvil [48].

2.4 REQUERIMIENTOS

2.4.1 REQUERIMIENTOS FUNCIONALES

A continuación, se describirán los requerimientos funcionales de las actividades que ejecutará la aplicación.

N°	Descripción del Requerimiento
1	La aplicación permitirá el ingreso de un correo y clave para que cada usuario pueda acceder con sus credenciales.
2	Se requiere un correo y una contraseña para iniciar sesión, en el caso de que una nueva persona quiera acceder, la aplicación tendrá un registro de usuario para que obtenga sus credenciales y poder iniciar sesión.
3	La aplicación permitirá al usuario registrado hacer uso de el detector de señas mediante un botón.
4	La aplicación abrirá la cámara y en un cuadro delimitador englobará la seña del usuario mostrando la traducción en un texto arriba del cuadro.
5	La aplicación permitirá la traducción de lenguaje de señas.
6	La aplicación móvil validará el tipo de usuario que ha iniciado sesión.
7	Si un usuario es administrador e inicia sesión la aplicación móvil lo redirigirá a la interfaz de administrador.
8	La interfaz de administrador contará con dos botones principales cuando el usuario admin inicie sesión que son: “iniciar traducción” y “Agregar nuevo patrón”.

N°	Descripción del Requerimiento
9	El usuario administrador podrá comprobar los patrones guardados mediante el botón “comprobar patrones”.
10	La interfaz de administrador contará con una cámara encendida para que el usuario administrador pueda realizar la comprobación de patrones.
11	El usuario administrador podrá ingresar nuevos patrones mediante el boton “Agregar nuevos patrones”.
12	Cuando el administrador oprima el botón “agregar nuevos patrones” se abrirá una nueva interfaz con tres opciones que servirán para agregar nuevos patrones y entrenar el modelo.
13	La nueva interfaz contendrá los botones “Iniciar captura de imágenes”, “Actualizar dataset” y “Clasificador”.
14	Cuando el usuario administrador oprima el botón “Iniciar captura de imágenes” se abrirá una nueva interfaz para capturar nuevos patrones.
15	En la nueva interfaz para agregar nuevos patrones tendrá las siguientes opciones: Una casilla para entrada de datos del número de patrones a ingresar, una casilla para entrada de datos del tamaño del dataset y un botón para generar los nombres, una cámara integrada para capturar las imágenes y un botón para iniciar la captura.
16	La aplicación móvil validará que los campos estén llenos para poder generar el nombre de los patrones, además de que se espera que no ingrese valores incorrectos en las casillas.
17	La aplicación móvil tomará captura de los datos de clases ingresado siempre y cuando el nombre de los patrones no se repita con los guardados en el servidor.

N°	Descripción del Requerimiento
18	La interfaz de usuario deberá tener dos botones.
19	Cuando una persona ingrese de manera normal sin privilegios de administrador la aplicación móvil redirigirá a la interfaz de usuario.
20	Al ingresar a la interfaz de usuario, el usuario podrá utilizar las funciones de los botones en este caso el botón “traducir” que permitirá al usuario traducir una seña con los patrones guardados previamente.
21	El usuario podrá ver los patrones guardados y como es la seña con la que el modelo detecta el patron.
22	Se requiere tener acceso a internet

Tabla 4. Requerimientos Funcionales

2.4.2 REQUERIMIENTOS NO FUNCIONALES

A continuación, se describen los requerimientos no funcionales, estos son los que hacen alusión a las propiedades del sistema.

N°	Descripción del Requerimiento
1	La aplicación deberá reconocer señas de acuerdo con las señas establecidas en el Lenguaje de señas ecuatoriano “Gabriel Román”.
2	La aplicación deberá tener una interfaz sencilla e intuitiva para fácil manejo de los usuarios.
3	El desarrollo de la aplicación contará con un backend, API y frontend para poder mantener una comunicación con el servicio y la aplicación.
4	La aplicación será desarrollada en lenguaje de programación Python.
5	Se hará uso de Visual Studio Code como IDE para el desarrollo del algoritmo de detección de señas

N°	Descripción del Requerimiento
6	Se utilizará Flutter como framework para el desarrollo de la interfaz en dispositivos móviles.
7	Se hará uso de librerías como MediaPipe, OpenCv, scikit-learn para el reconocimiento de señas.
8	Se utilizará XAMPP para realizar la gestión de la base de datos
9	Se utilizará phpMyAdmin para la creación de la base de datos que permitirá al usuario iniciar sesión y registrarse.
10	Se hará uso de Android estudio para desarrollar el código de la aplicación móvil mediante flutter y lenguaje de programación Dart.

Tabla 5 Requerimientos No Funcionales

2.5 COMPONENTES DE LA PROPUESTA

2.5.1 ARQUITECTURA DEL SISTEMA

CLIENTE – SERVIDOR

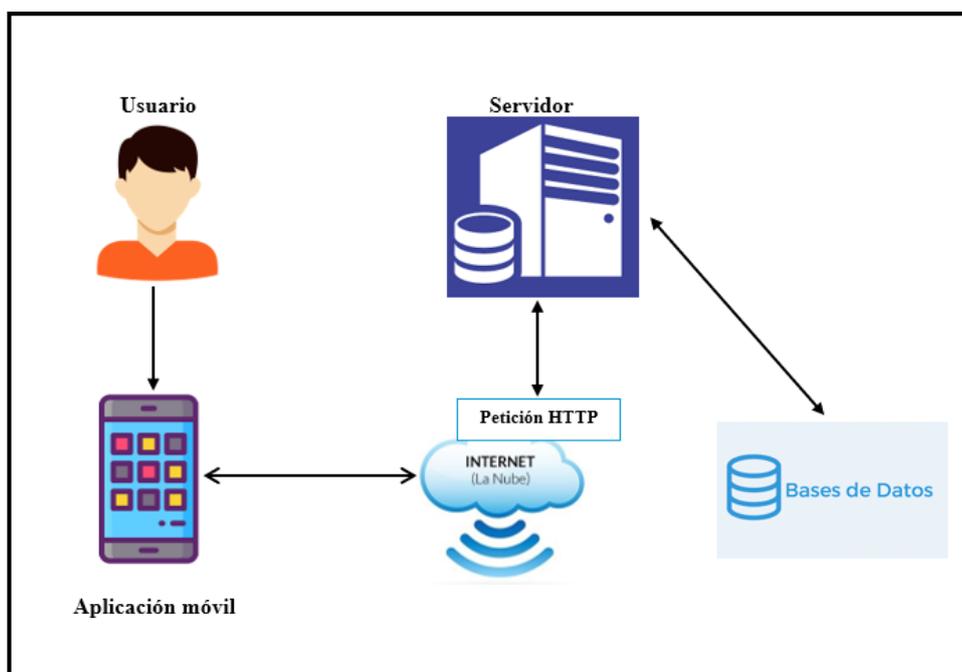


Fig. 6. Arquitectura cliente – servidor

La estructura cliente – servidor es un tipo de arquitectura de procesamiento cooperativo donde clientes y servidores colaboran en la gestión de todas las funcionalidades de una aplicación o conjunto de aplicaciones, en este modelo los clientes realizan solicitudes de servicio (Request) a los servidores, quienes procesan estas solicitudes según una lógica específica y devuelven un resultado (Response) como parte de este proceso, a nivel físico (hardware), clientes y servidores pueden compartir el mismo procesador, estar en procesadores diferentes, ubicarse en el mismo servidor físico o en servidores virtuales, y estar juntos o separados físicamente [49].

En general, los clientes son procesos o aplicaciones que solicitan servicios, y los servidores son procesos o aplicaciones que proporcionan esos servicios, en este esquema comúnmente el cliente empieza la comunicación con uno o varios servidores, un cliente puede solicitar un servicio a varios servidores, el cliente y servidor pueden trabajar en el mismo hardware o en uno diferente, inclusive pueden trabajar en redes diferentes ya sea que estén divididos tanto física como lógica.

2.5.2 INSTRUMENTOS DE DESARROLLO

Para el desarrollo del proyecto fue necesario el uso de los siguientes instrumentos:

Equipo Hardware:

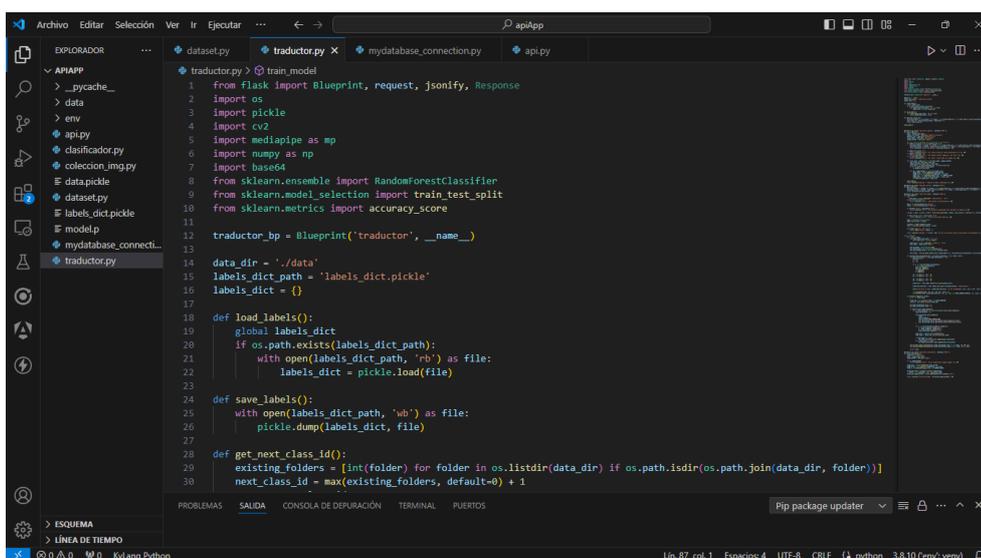
- 1 computador portátil con: Procesador de 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.40 GHz, 8GB de memoria RAM.
- 1 celular Realme C3 modelo RMX2020 con 3 GB de memoria RAM, procesador G70, con triple cámara principal de 12 MP + 2MP + 2 MP y una cámara frontal de 5MP, versión de Android 10.

Equipo Software:

- Visual Studio Code 1.89.1
- Android Studio Giraffe | Patch 3
- Flutter 3.22.0
- Python 3.8
- Librerías

2.5.3 DESARROLLO DEL ALGORITMO DE DETECCIÓN Y TRADUCCIÓN DE SEÑAS MEDIANTE EL USO DE LA SOLUCIÓN MEDIAPIPE HANDS.

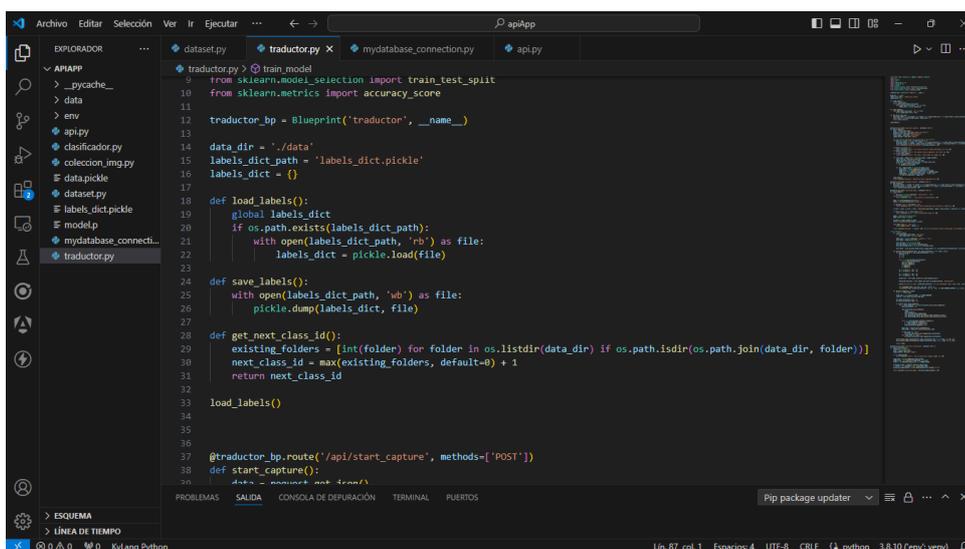
El algoritmo de detección y traducción de señas está dividido en cuatro procesos que lo componen de la siguiente manera: captura de muestra de nuevos patrones, actualización de dataset, clasificador y por último traducción. A continuación, se muestran todas las importaciones necesarias.



```
1 from flask import Blueprint, request, jsonify, Response
2 import os
3 import pickle
4 import cv2
5 import mediapipe as mp
6 import numpy as np
7 import base64
8 from sklearn.ensemble import RandomForestClassifier
9 from sklearn.model_selection import train_test_split
10 from sklearn.metrics import accuracy_score
11
12 traductor_bp = Blueprint('traductor', __name__)
13
14 data_dir = './data'
15 labels_dict_path = 'labels_dict.pickle'
16 labels_dict = {}
17
18 def load_labels():
19     global labels_dict
20     if os.path.exists(labels_dict_path):
21         with open(labels_dict_path, 'rb') as file:
22             labels_dict = pickle.load(file)
23
24 def save_labels():
25     with open(labels_dict_path, 'wb') as file:
26         pickle.dump(labels_dict, file)
27
28 def get_next_class_id():
29     existing_folders = [int(folder) for folder in os.listdir(data_dir) if os.path.isdir(os.path.join(data_dir, folder))]
30     next_class_id = max(existing_folders, default=0) + 1
```

Fig. 7. Importaciones para realizar cada uno de los métodos de la API

Se muestran tres funciones la primera es load_labels(): su función es el de cargar las etiquetas de cada uno de las clases guardadas.



```
9 from sklearn.model_selection import train_test_split
10 from sklearn.metrics import accuracy_score
11
12 traductor_bp = Blueprint('traductor', __name__)
13
14 data_dir = './data'
15 labels_dict_path = 'labels_dict.pickle'
16 labels_dict = {}
17
18 def load_labels():
19     global labels_dict
20     if os.path.exists(labels_dict_path):
21         with open(labels_dict_path, 'rb') as file:
22             labels_dict = pickle.load(file)
23
24 def save_labels():
25     with open(labels_dict_path, 'wb') as file:
26         pickle.dump(labels_dict, file)
27
28 def get_next_class_id():
29     existing_folders = [int(folder) for folder in os.listdir(data_dir) if os.path.isdir(os.path.join(data_dir, folder))]
30     next_class_id = max(existing_folders, default=0) + 1
31     return next_class_id
32
33 load_labels()
34
35
36
37 @traductor_bp.route('/api/start_capture', methods=['POST'])
38 def start_capture():
39     data = request.get_json()
```

Fig. 8. Funciones para cargar etiquetas almacenadas y proporcionar nuevos IDs

Para realizar la captura de imágenes se crea un método POST en la API en donde se pide que ingrese el número de clases, tamaño del dataset y el nombre de la clase luego de ello se iniciará con la captura de datos para almacenarlo en el servidor, se validan que los valores a ingresar no sean negativos ni mayores a 5, no ingresar valores negativos y que es necesario llenar todos los campos.

```

35
36
37 @tradorator_bp.route('/api/start_capture', methods=['POST'])
38 def start_capture():
39     data = request.get_json()
40     number_of_classes = data.get('number_of_classes')
41     dataset_size = data.get('dataset_size')
42     class_names = data.get('class_names')
43     images_base64 = data.get('images')
44
45     #si los valores ingresados son 0 actualiza el diccionario
46     if number_of_classes == 0 and dataset_size == 0:
47         existing_folders = [folder for folder in os.listdir(data_dir) if os.path.isdir(os.path.join(data_dir, folder))]
48         existing_classes = {folder: labels_dict[int(folder)] for folder in existing_folders if int(folder) in labels_dict}
49         return jsonify({"error": "Clases existentes": existing_classes}), 200
50
51     if number_of_classes > 5:
52         return jsonify({"error": "El número máximo de clases permitidas es 5."}), 400
53     if number_of_classes < 0:
54         return jsonify({"error": "No ingrese valores negativos, por favor."}), 400
55     if not all(class_names):
56         return jsonify({"error": "Por favor, llene todos los campos."}), 400
57
58     for class_name, image_list in zip(class_names, images_base64):
59         next_class_id = get_next_class_id()
60         labels_dict[next_class_id] = class_name
61         class_path = os.path.join(data_dir, str(next_class_id))
62         if not os.path.exists(class_path):
63             os.makedirs(class_path)
64

```

Fig. 9. Método para generar la captura de imágenes.

Luego de capturar las imágenes es necesario actualizar el dataset para ello se crea un método POST que permita realizar dicha tarea.

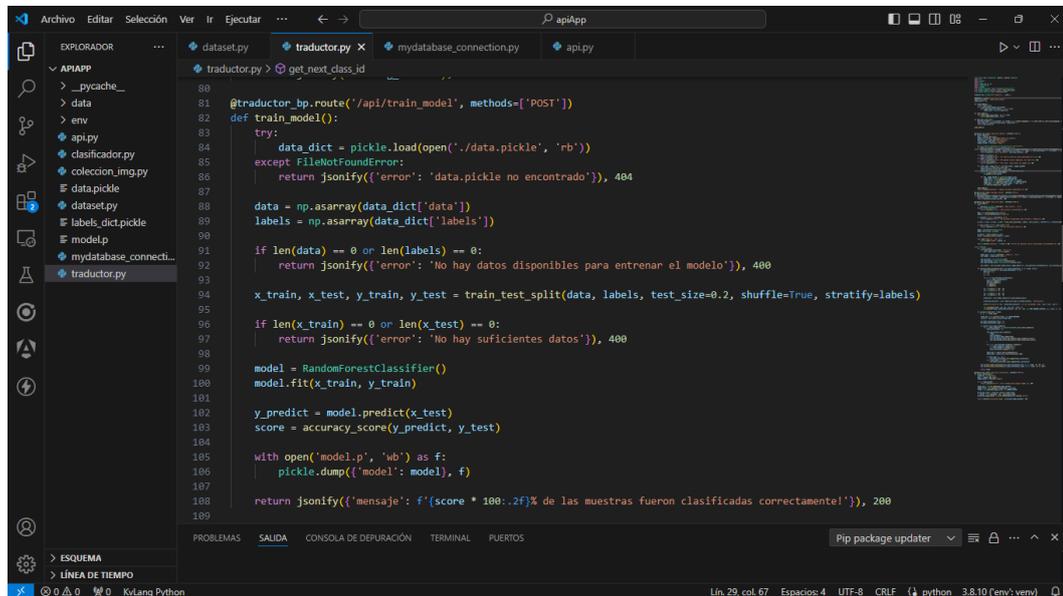
```

14 DATA_DIR = './data'
15
16 @dataset_bp.route('/api/process_data', methods=['POST'])
17 def process_data():
18     mp_hands = mp.solutions.hands
19     hands = mp_hands.Hands(static_image_mode=True, min_detection_confidence=0.3)
20
21     data = []
22     labels = []
23
24     for dir_ in os.listdir(DATA_DIR):
25         for img_path in os.listdir(os.path.join(DATA_DIR, dir_)):
26             data_aux = []
27             img = cv2.imread(os.path.join(DATA_DIR, dir_, img_path))
28             img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
29
30             results = hands.process(img_rgb)
31             if results.multi_hand_landmarks:
32                 for hand_landmarks in results.multi_hand_landmarks:
33                     for i in range(len(hand_landmarks.landmark)):
34                         x = hand_landmarks.landmark[i].x
35                         y = hand_landmarks.landmark[i].y
36                         data_aux.append(x)
37                         data_aux.append(y)
38
39             data.append(data_aux)
40             labels.append(dir_)
41
42     with open('data.pickle', 'wb') as f:
43         pickle.dump({'data': data, 'labels': labels}, f)

```

Fig. 10. Método para actualizar el dataset

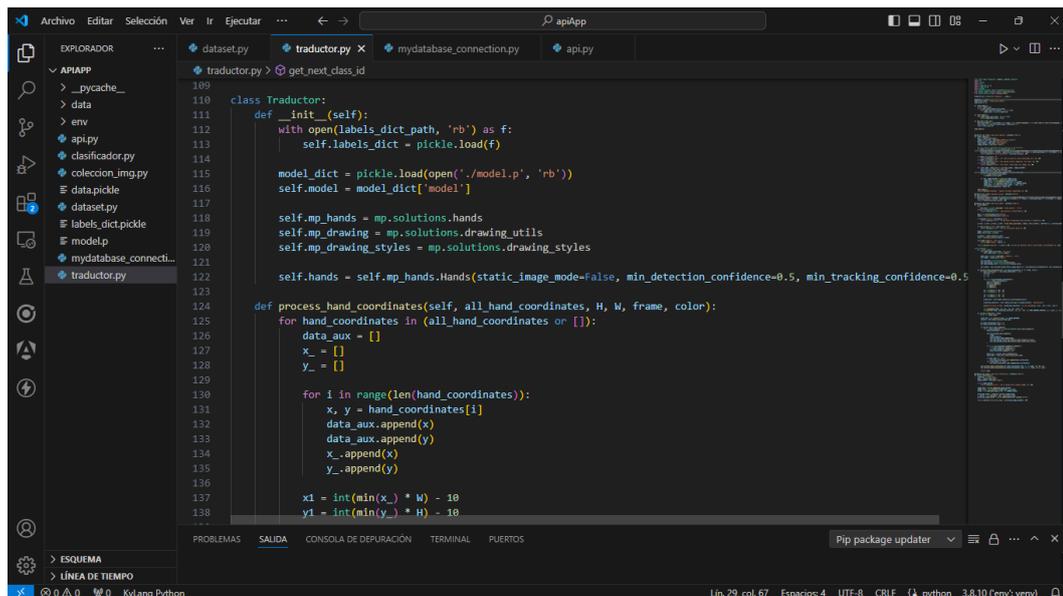
Es necesario en una aplicación que funciona mediante aprendizaje automático el tener que clasificar las muestras que se han tomado en los anteriores puntos, para ello se desarrolla un método POST que realice ese trabajo, una vez clasificados los datos se emitirá un mensaje con el porcentaje que se clasificó de manera correcta.



```
80
81 @tradorador_bp.route('/api/train_model', methods=['POST'])
82 def train_model():
83     try:
84         data_dict = pickle.load(open('./data.pickle', 'rb'))
85     except FileNotFoundError:
86         return jsonify({'error': 'data.pickle no encontrado'}), 404
87
88     data = np.asarray(data_dict['data'])
89     labels = np.asarray(data_dict['labels'])
90
91     if len(data) == 0 or len(labels) == 0:
92         return jsonify({'error': 'No hay datos disponibles para entrenar el modelo'}), 400
93
94     x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, shuffle=True, stratify=labels)
95
96     if len(x_train) == 0 or len(x_test) == 0:
97         return jsonify({'error': 'No hay suficientes datos'}), 400
98
99     model = RandomForestClassifier()
100    model.fit(x_train, y_train)
101
102    y_predict = model.predict(x_test)
103    score = accuracy_score(y_predict, y_test)
104
105    with open('model.p', 'wb') as f:
106        pickle.dump({'model': model}, f)
107
108    return jsonify({'mensaje': f'{score * 100:.2f}% de las muestras fueron clasificadas correctamente'}), 200
109
```

Fig. 11. Método para clasificar las muestras

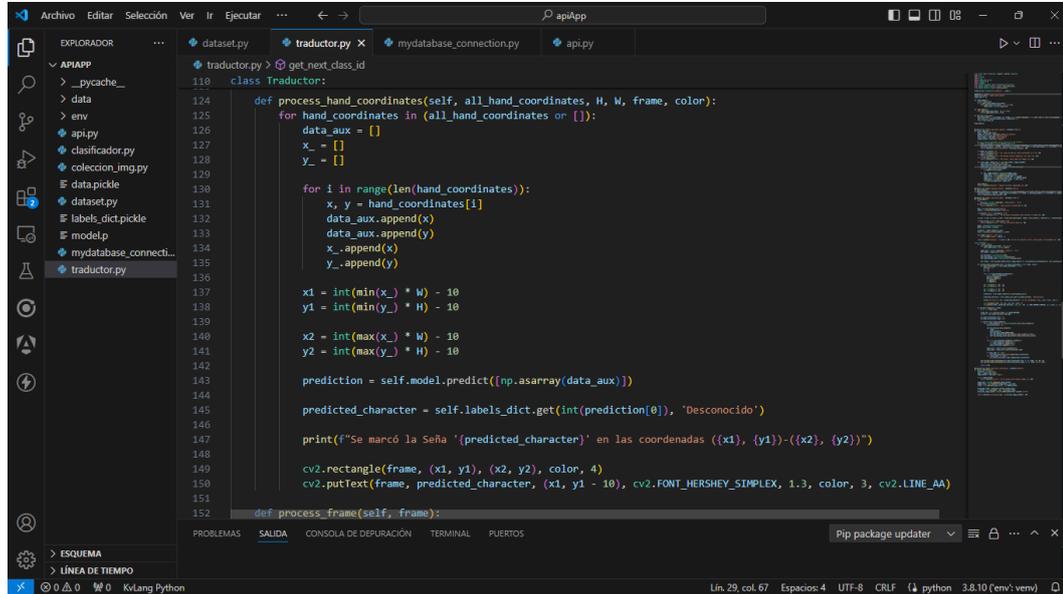
La siguiente imagen muestra el proceso de traducción usando las librerías de MediaPipe en este caso la posición de las manos, los puntos de referencia y el color de se verá para los puntos de referencia.



```
109
110 class Traductor:
111     def __init__(self):
112         with open(labels_dict_path, 'rb') as f:
113             self.labels_dict = pickle.load(f)
114
115         model_dict = pickle.load(open('./model.p', 'rb'))
116         self.model = model_dict['model']
117
118         self.mp_hands = mp.solutions.hands
119         self.mp_drawing = mp.solutions.drawing_utils
120         self.mp_drawing_styles = mp.solutions.drawing_styles
121
122         self.hands = self.mp_hands.Hands(static_image_mode=False, min_detection_confidence=0.5, min_tracking_confidence=0.5)
123
124     def process_hand_coordinates(self, all_hand_coordinates, H, W, frame, color):
125         for hand_coordinates in all_hand_coordinates or []:
126             data_aux = []
127             x_ = []
128             y_ = []
129
130             for i in range(len(hand_coordinates)):
131                 x, y = hand_coordinates[i]
132                 data_aux.append(x)
133                 data_aux.append(y)
134                 x_.append(x)
135                 y_.append(y)
136
137             x1 = int(min(x_) * W) - 10
138             y1 = int(min(y_) * H) - 10
```

Fig. 12. Clase Traductor

La clase traductor presenta dos funciones la primera es `process_hand_coordinates()` en donde se procesa las coordenadas de las manos, realiza la predicción usando el modelo de aprendizaje automático.



```
class Traductor:
    def process_hand_coordinates(self, all_hand_coordinates, H, W, frame, color):
        for hand_coordinates in all_hand_coordinates on ([]):
            data_aux = []
            x_ = []
            y_ = []

            for i in range(len(hand_coordinates)):
                x, y = hand_coordinates[i]
                data_aux.append(x)
                data_aux.append(y)
                x_.append(x)
                y_.append(y)

            x1 = int(min(x_) * W) - 10
            y1 = int(min(y_) * H) - 10

            x2 = int(max(x_) * W) - 10
            y2 = int(max(y_) * H) - 10

            prediction = self.model.predict(np.asarray(data_aux))

            predicted_character = self.labels_dict.get(int(prediction[0]), 'Desconocido')

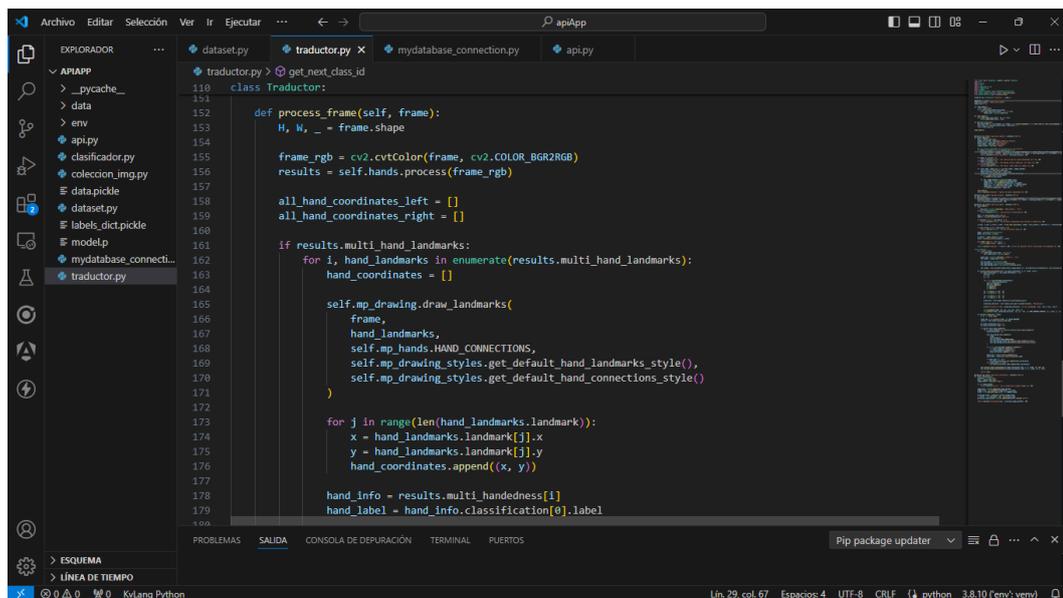
            print(f"Se marcó la Seña '{predicted_character}' en las coordenadas ({x1}, {y1})-({x2}, {y2})")

            cv2.rectangle(frame, (x1, y1), (x2, y2), color, 4)
            cv2.putText(frame, predicted_character, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 1.3, color, 3, cv2.LINE_AA)

    def process_frame(self, frame):
```

Fig. 13. Función de la clase traductor `process_hand_coordinates`.

La segunda función de la clase traductor tiene la finalidad de procesar fotogramas de un video para detectar los gestos realizado de la mano, usa MediaPipe para detectar el gesto y si detecta una mano dibuja los puntos de referencia.



```
class Traductor:
    def process_frame(self, frame):
        H, W, _ = frame.shape

        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        results = self.hands.process(frame_rgb)

        all_hand_coordinates_left = []
        all_hand_coordinates_right = []

        if results.multi_hand_landmarks:
            for i, hand_landmarks in enumerate(results.multi_hand_landmarks):
                hand_coordinates = []

                self.mp_drawing.draw_landmarks(
                    frame,
                    hand_landmarks,
                    self.mp_hands.HAND_CONNECTIONS,
                    self.mp_drawing_styles.get_default_hand_landmarks_style(),
                    self.mp_drawing_styles.get_default_hand_connections_style()
                )

                for j in range(len(hand_landmarks.landmark)):
                    x = hand_landmarks.landmark[j].x
                    y = hand_landmarks.landmark[j].y
                    hand_coordinates.append((x, y))

                hand_info = results.multi_handedness[i]
                hand_label = hand_info.classification[0].label
```

Fig. 14. Función de la clase traductor `process_frame`

Una vez realizado cada uno de los procesos anteriores correctamente se realiza la parte que hará la traducción para ello se crea un método POST que realizará ese trabajo como se muestra a continuación.

```

class Traductor:
    def process_frame(self, frame):
        return frame

@app.route('/api/start_translation', methods=['POST'])
def start_translation():
    traductor = Traductor()
    data = request.get_json()
    image_base64 = data.get('image')

    if not image_base64:
        return jsonify({'error': 'No se proporcionó ninguna imagen.'}), 400

    image_data = base64.b64decode(image_base64)
    image_array = np.frombuffer(image_data, np.uint8)
    frame = cv2.imdecode(image_array, cv2.IMREAD_COLOR)

    processed_frame = traductor.process_frame(frame)
    _, buffer = cv2.imencode('.jpg', processed_frame)
    processed_image_base64 = base64.b64encode(buffer).decode('utf-8')

    return jsonify({'processed_image': processed_image_base64}), 200

```

Fig. 15. Método para realizar la traducción

Se crea un método POST que esté conectado con la base de datos, este método pedirá que se ingrese el correo y contraseña, verificará que las credenciales ingresadas sean correctas para ingresar a la interfaz correspondiente al usuario

2.5.4 CREACIÓN DE LA API PARA LA CONEXIÓN DE LA APLICACIÓN MÓVIL CON EL SERVIDOR Y BASE DE DATOS.

Se realiza la conexión de la API usando flask de python agregando cada uno de los métodos creados y se prueba su funcionamiento.

```

from flask import Flask
from flask_cors import CORS
from traductor import traductor_bp
from dataset import dataset_bp

app = Flask(__name__)
CORS(app, resources={r"/*": {"origins": "*"}})
app.register_blueprint(traductor_bp)
app.register_blueprint(dataset_bp)

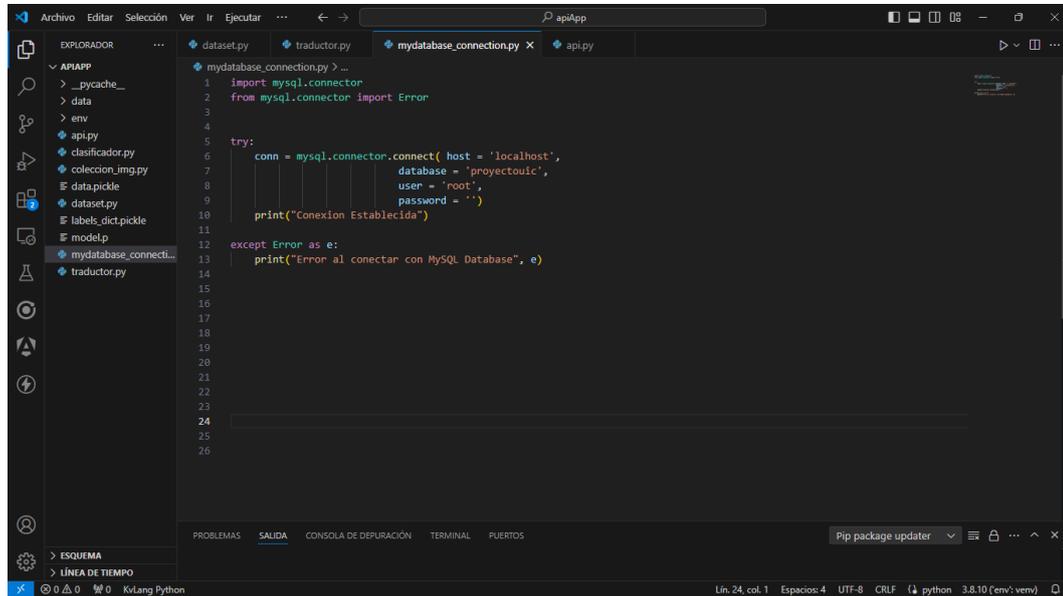
@app.route("/")
def home():
    return "Bienvenido a la API del traductor de lenguaje de señas"

if __name__ == "__main__":
    app.run(host="192.168.100.204", port=5000, debug=True)

```

Fig. 16. Conexión de la API con cada uno de los métodos

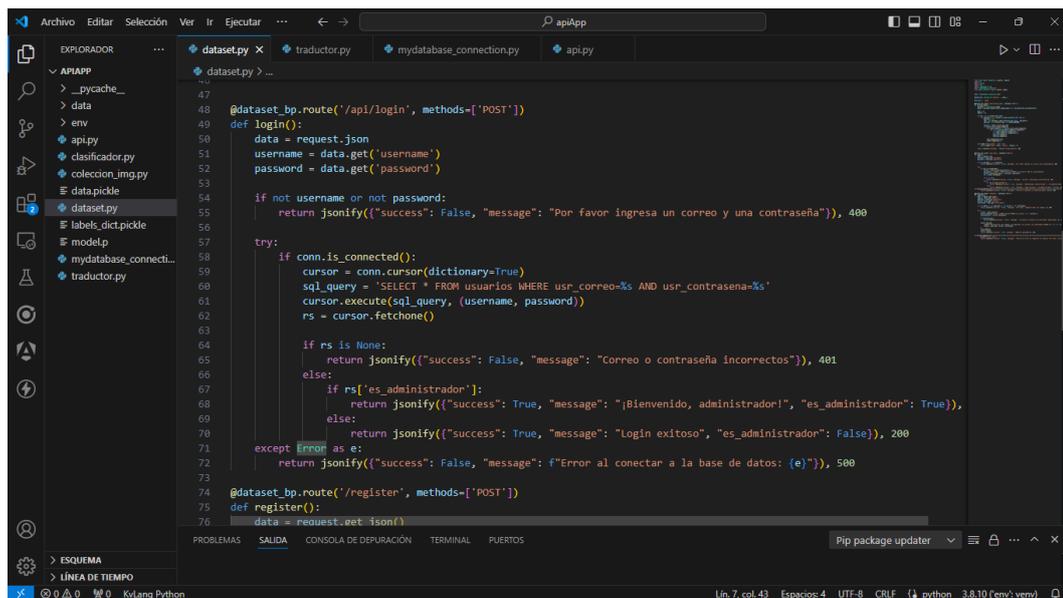
Esta parte del código muestra cómo se realiza la conexión con la base de datos, para eso se importan las librerías de `mysql.connector`, esto servirá para permitir el ingreso de usuarios así como también el registro, está conectado con la API .



```
1 import mysql.connector
2 from mysql.connector import Error
3
4
5
6
7
8
9
10 print("Conexion Establecida")
11
12
13 except Error as e:
14     print("Error al conectar con MySQL Database", e)
15
16
17
18
19
20
21
22
23
24
25
26
```

Fig. 17. Conexión con la base de datos

La siguiente imagen muestra el método que sirve para iniciar sesión, este consiste en realizar la conexión con la base de datos para que las credenciales correo y contraseña que el usuario ingrese sean validadas y verificar que son correctas e ingresar a la interfaz correspondiente.



```
47
48 @dataset_bp.route('/api/login', methods=['POST'])
49 def login():
50     data = request.json
51     username = data.get('username')
52     password = data.get('password')
53
54     if not username or not password:
55         return jsonify({"success": False, "message": "Por favor ingresa un correo y una contraseña"}), 400
56
57     try:
58         if conn.is_connected():
59             cursor = conn.cursor(dictionary=True)
60             sql_query = "SELECT * FROM usuarios WHERE usr_correo=%s AND usr_contrasena=%s"
61             cursor.execute(sql_query, (username, password))
62             rs = cursor.fetchone()
63
64             if rs is None:
65                 return jsonify({"success": False, "message": "Correo o contraseña incorrectos"}), 401
66             else:
67                 if rs['es_administrador']:
68                     return jsonify({"success": True, "message": "¡Bienvenido, administrador!", "es_administrador": True}), 200
69                 else:
70                     return jsonify({"success": True, "message": "Login exitoso", "es_administrador": False}), 200
71     except Error as e:
72         return jsonify({"success": False, "message": f"Error al conectar a la base de datos: {e}"}), 500
73
74 @dataset_bp.route('/api/register', methods=['POST'])
75 def register():
76     data = request.get_json()
```

Fig. 18. Método para iniciar sesión

Al igual que el método para iniciar sesión el método de registro también estará conectado con la base de datos, se pedirá el usuario que ingrese nombre, apellido, correo y contraseña, se validará que los datos estén correctos y se procederá registrar al usuario para que este tenga sus credenciales de inicio de sesión.

```

73
74 @dataset_bp.route('/register', methods=['POST'])
75 def register():
76     data = request.get_json()
77     nombre = data.get('nombre')
78     apellido = data.get('apellido')
79     correo = data.get('correo')
80     contraseña = data.get('contrasena')
81
82     if not nombre or not apellido or not correo or not contraseña:
83         return jsonify({'success': False, 'message': 'Por favor, complete todos los campos.'}), 400
84
85     try:
86         cursor = conn.cursor()
87         cursor.execute("SELECT * FROM usuarios WHERE usr_correo = %s", (correo,))
88         existing_user = cursor.fetchone()
89
90         if existing_user:
91             return jsonify({'success': False, 'message': 'Ya existe un usuario con esos datos registrados.'}), 409
92
93         cursor.execute(
94             "INSERT INTO usuarios (usr_nombre, usr_apellido, usr_correo, usr_contrasena) VALUES (%s, %s, %s, %s)",
95             (nombre, apellido, correo, contraseña)
96         )
97         conn.commit()
98         cursor.close()
99         return jsonify({'success': True, 'message': 'Registro guardado'}), 201
100
101     except Exception as e:
102         print(f"Error: {e}")

```

Fig. 19. Método para registro de usuarios

A continuación, se muestra la conexión desde la aplicación flutter en Android Studio con el servidor de la API, aquí se encuentran todos los métodos que facilitan la comunicación entre el cliente y servidor.

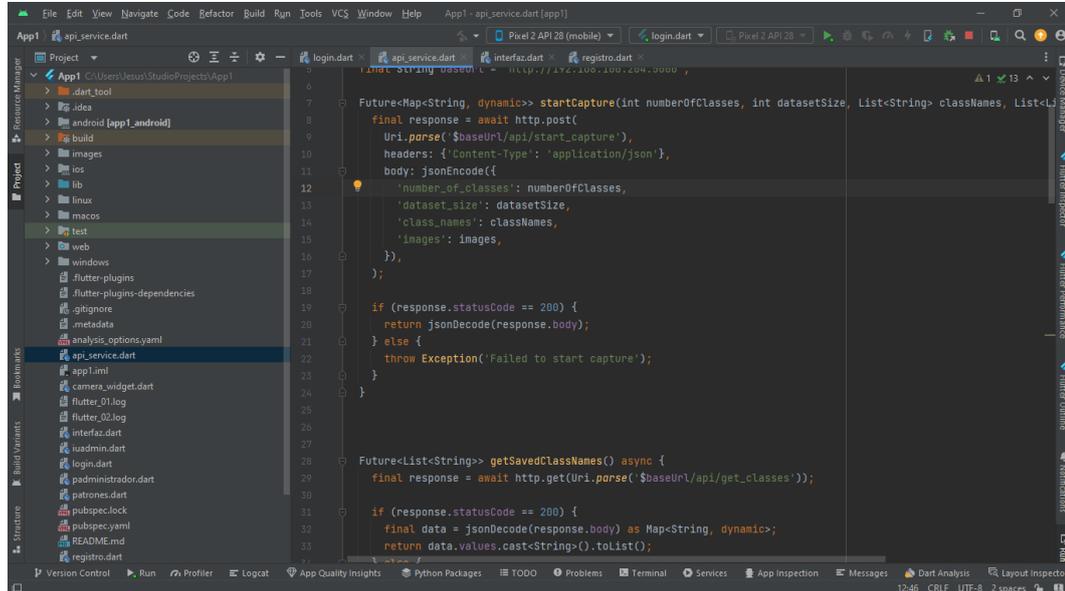
```

1 import 'dart:convert';
2 import 'package:http/http.dart' as http;
3
4 class ApiService {
5     final String baseUrl = 'http://192.168.100.284:5000';
6
7     Future<Map<String, dynamic>> startCapture(int numberOfClasses, int datasetSize, List<String> classNames, List<String> images) async {
8         final response = await http.post(
9             Uri.parse('$baseUrl/api/start_capture'),
10            headers: {'Content-Type': 'application/json'},
11            body: jsonEncode({
12                'number_of_classes': numberOfClasses,
13                'dataset_size': datasetSize,
14                'class_names': classNames,
15                'images': images,
16            }));
17
18         if (response.statusCode == 200) {
19             return jsonDecode(response.body);
20         } else {
21             throw Exception('Failed to start capture');
22         }
23     }
24
25     Future<List<String>> getSavedClassNames() async {
26         final response = await http.get(Uri.parse('$baseUrl/api/get_classes'));
27
28         if (response.statusCode == 200) {
29             return jsonDecode(response.body);
30         } else {
31             throw Exception('Failed to get saved class names');
32         }
33     }
34 }

```

Fig. 20. Conexión a la API del servidor desde la aplicación flutter en Android Studio

Se muestra la ruta desde la aplicación de flutter con la que se podrá iniciar la captura de imágenes se le pedirá que ingrese el número de patrones a grabar, tamaño de dataset y el nombre de la clase.



```
Future<Map<String, dynamic>> startCapture(int numberOfClasses, int datasetSize, List<String> classNames, List<String> images) async {
  final response = await http.post(
    Uri.parse('$baseUrl/api/start_capture'),
    headers: {'Content-Type': 'application/json'},
    body: jsonEncode({
      'number_of_classes': numberOfClasses,
      'dataset_size': datasetSize,
      'class_names': classNames,
      'images': images,
    })),
  );

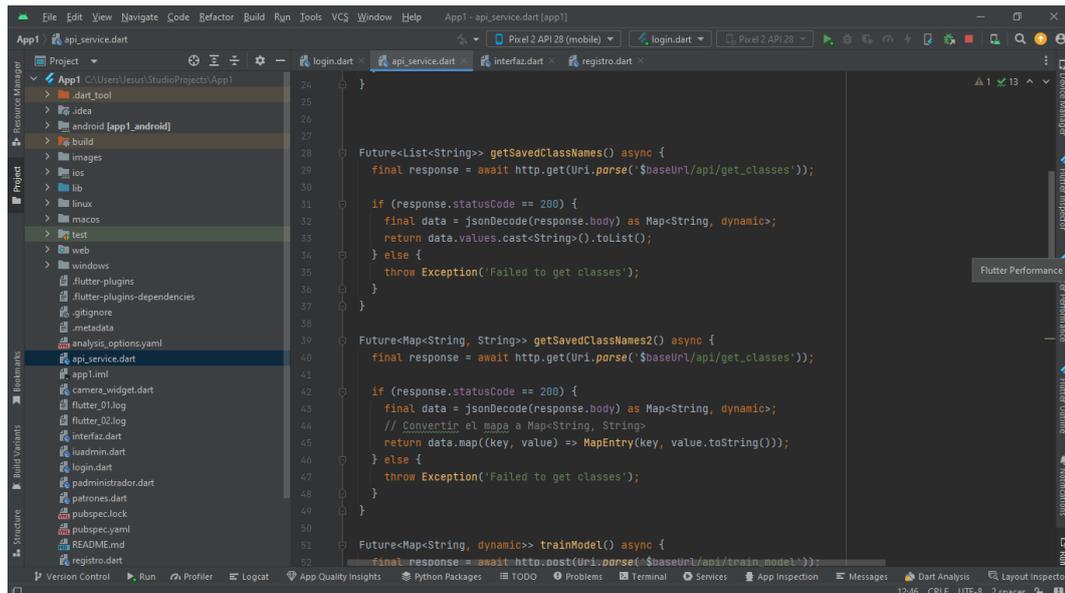
  if (response.statusCode == 200) {
    return jsonDecode(response.body);
  } else {
    throw Exception('Failed to start capture');
  }
}

Future<List<String>> getSavedClassNames() async {
  final response = await http.get(Uri.parse('$baseUrl/api/get_classes'));

  if (response.statusCode == 200) {
    final data = jsonDecode(response.body) as Map<String, dynamic>;
    return data.values.cast<String>().toList();
  }
}
```

Fig. 21. Ruta para iniciar la captura de imágenes desde la aplicación flutter en Android Studio

Se muestra la ruta desde la aplicación de flutter para obtener el nombre de las clases actuales guardadas en el servidor, esto servirá para que en la aplicación el usuario pueda visualizar cuales son los patrones de señas que hay guardados y sea sencillo el comprender como se realiza cada seña.



```
Future<List<String>> getSavedClassNames() async {
  final response = await http.get(Uri.parse('$baseUrl/api/get_classes'));

  if (response.statusCode == 200) {
    final data = jsonDecode(response.body) as Map<String, dynamic>;
    return data.values.cast<String>().toList();
  } else {
    throw Exception('Failed to get classes');
  }
}

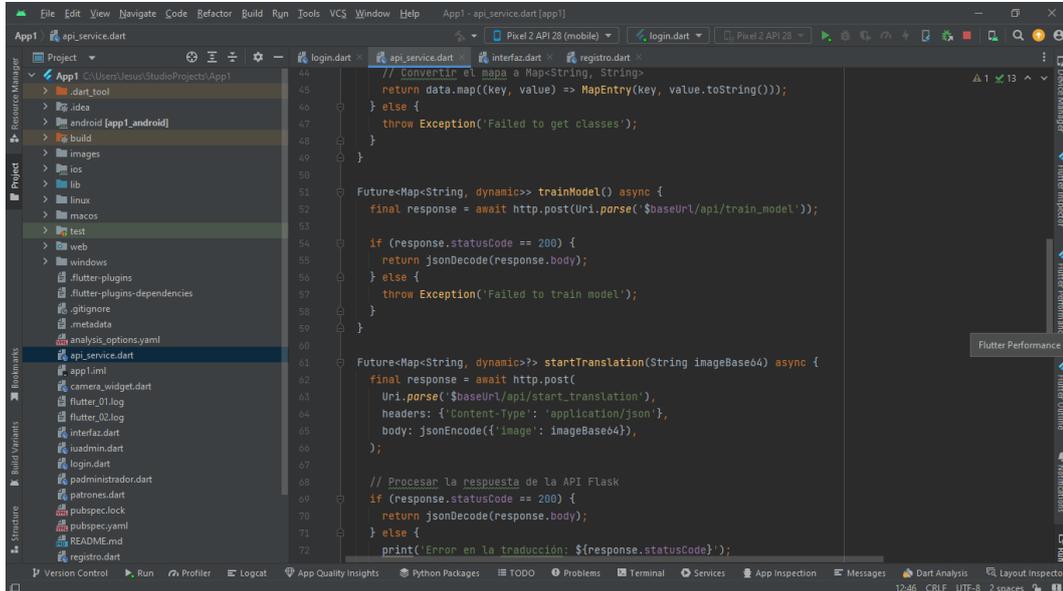
Future<Map<String, String>> getSavedClassNames2() async {
  final response = await http.post(Uri.parse('$baseUrl/api/get_classes'));

  if (response.statusCode == 200) {
    final data = jsonDecode(response.body) as Map<String, dynamic>;
    // Convertir el mapa a Map<String, String>
    return data.map((key, value) => MapEntry(key, value.toString()));
  } else {
    throw Exception('Failed to get classes');
  }
}

Future<Map<String, dynamic>> trainModel() async {
  final response = await http.post(Uri.parse('$baseUrl/api/train_model'));
}
```

Fig. 22. Rutas para obtener los nombres de las clases guardadas desde la aplicación flutter en Android Studio

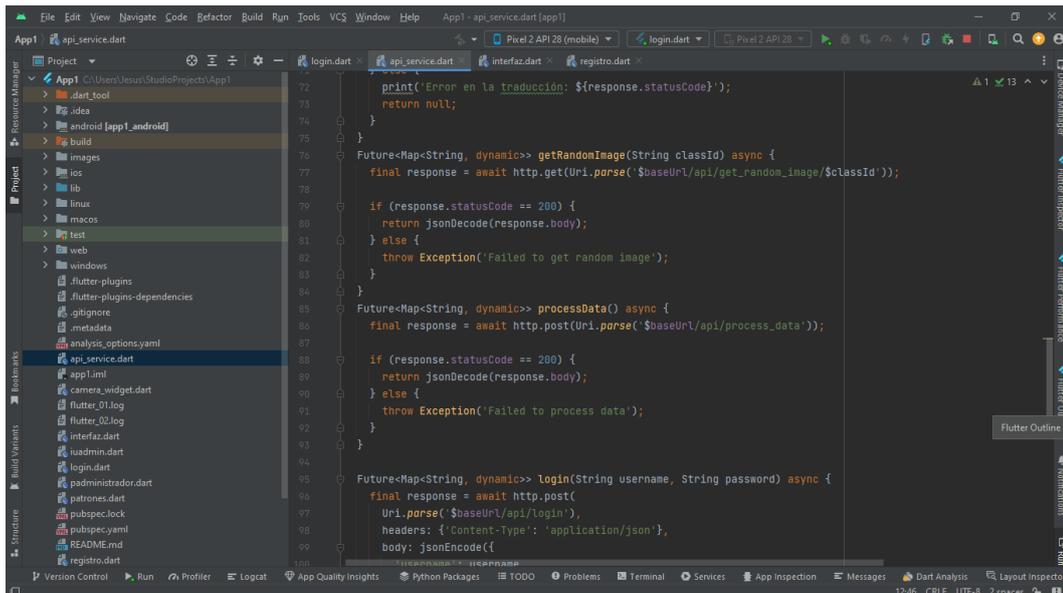
Ruta que sirve para realizar el entrenamiento del modelo, desde la aplicación flutter envía solicitudes http al servidor para que este le devuelva la seña traducida, esta ruta será llamada por el administrador y el usuario.



```
44 // Convertir el mapa a Map<String, String>
45 return data.map((key, value) => MapEntry(key, value.toString()));
46 } else {
47   throw Exception('Failed to get classes');
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 Future<Map<String, dynamic>> trainModel() async {
62   final response = await http.post(Uri.parse('${baseUrl}/api/train_model'));
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
```

Fig. 23. Ruta para entrenar el modelo desde la aplicación flutter en Android Studio

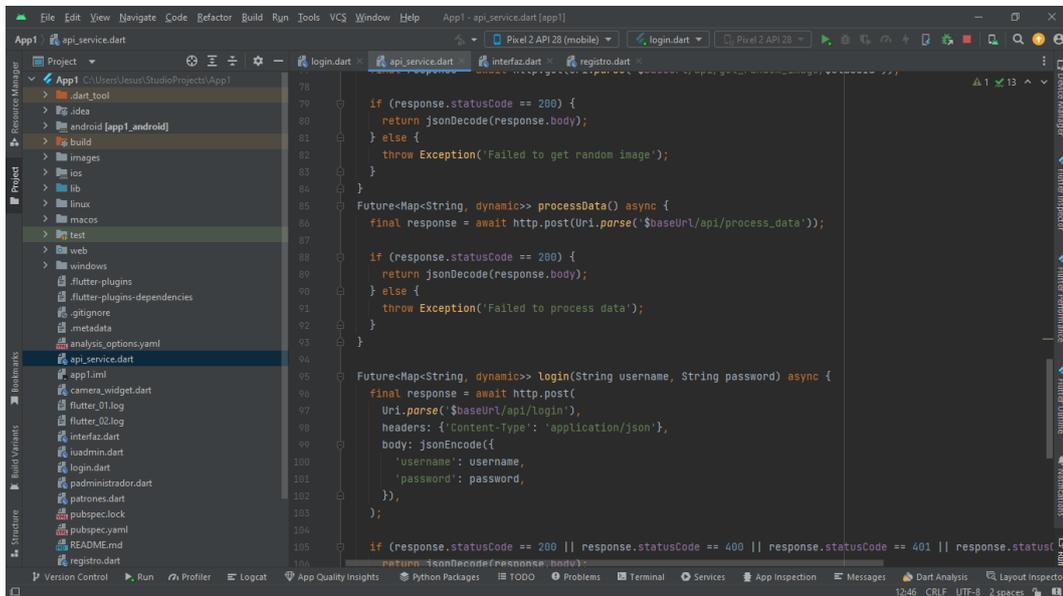
La siguiente ruta sirve para poder mostrar el usuario una imagen de manera random que se encuentre dentro de la clase seleccionada, servirá cuando un usuario quiera ver los patrones que existen guardados y se le haga fácil el poder ver como se realiza la postura de un patrón.



```
72 print('Error en la traducción: ${response.statusCode}');
73 return null;
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 Future<Map<String, dynamic>> getRandomImage(String classId) async {
89   final response = await http.get(Uri.parse('${baseUrl}/api/get_random_image/$classId'));
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
```

Fig. 24. Ruta para obtener una imagen aleatoria desde la aplicación flutter en Android Studio

La ruta que se muestra a continuación sirve para poder actualizar el dataset existente en el servidor cuando se hayan ingresado nuevos patrones, de esta forma poder tener el dataset completo con los últimos patrones agregados y que estos puedan ser clasificados sin presentar algún problema.



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help App1 - api_service.dart [app1]
Project
  App1 C:\Users\Jesus\StudioProjects\App1
    .dart_tool
    .idea
    android [app1_android]
    build
    images
    ios
    lib
    linux
    macos
    test
    web
    windows
    flutter-plugins
    flutter-plugins-dependencies
    gitignore
    imetadata
    analysis_options.yaml
    api_service.dart
    app1.lml
    camera_widget.dart
    flutter_01.log
    flutter_02.log
    interfaz.dart
    luadmin.dart
    login.dart
    padministrador.dart
    patrones.dart
    pubspec.lock
    pubspec.yaml
    README.md
    registro.dart
  Build Variants
  Structure
  Bootstrap

login.dart
api_service.dart
interfaz.dart
registro.dart

if (response.statusCode == 200) {
  return jsonDecode(response.body);
} else {
  throw Exception('Failed to get random image');
}

Future<Map<String, dynamic>> processData() async {
  final response = await http.post(Uri.parse('${baseUrl}/api/process_data'));

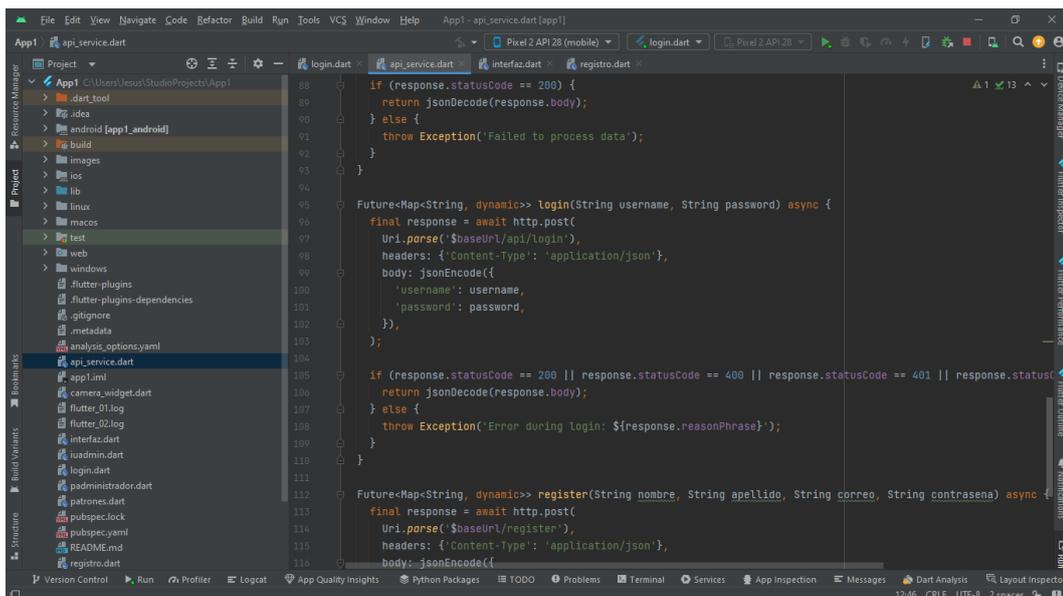
  if (response.statusCode == 200) {
    return jsonDecode(response.body);
  } else {
    throw Exception('Failed to process data');
  }
}

Future<Map<String, dynamic>> login(String username, String password) async {
  final response = await http.post(
    Uri.parse('${baseUrl}/api/login'),
    headers: {'Content-Type': 'application/json'},
    body: jsonEncode({
      'username': username,
      'password': password,
    }));
}

if (response.statusCode == 200 || response.statusCode == 400 || response.statusCode == 401 || response.status
```

Fig. 25- Ruta que sirve para actualizar el dataset desde la aplicación flutter en Android Studio

Esta ruta sirve para ejecutar el proceso de sesión para los usuarios, se pide ingresar un correo y una contraseña, estas son validadas de acuerdo con el tipo de usuario.



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help App1 - api_service.dart [app1]
Project
  App1 C:\Users\Jesus\StudioProjects\App1
    .dart_tool
    .idea
    android [app1_android]
    build
    images
    ios
    lib
    linux
    macos
    test
    web
    windows
    flutter-plugins
    flutter-plugins-dependencies
    gitignore
    imetadata
    analysis_options.yaml
    api_service.dart
    app1.lml
    camera_widget.dart
    flutter_01.log
    flutter_02.log
    interfaz.dart
    luadmin.dart
    login.dart
    padministrador.dart
    patrones.dart
    pubspec.lock
    pubspec.yaml
    README.md
    registro.dart
  Build Variants
  Structure
  Bootstrap

login.dart
api_service.dart
interfaz.dart
registro.dart

if (response.statusCode == 200) {
  return jsonDecode(response.body);
} else {
  throw Exception('Failed to process data');
}

Future<Map<String, dynamic>> login(String username, String password) async {
  final response = await http.post(
    Uri.parse('${baseUrl}/api/login'),
    headers: {'Content-Type': 'application/json'},
    body: jsonEncode({
      'username': username,
      'password': password,
    }));
}

if (response.statusCode == 200 || response.statusCode == 400 || response.statusCode == 401 || response.status
return jsonDecode(response.body);
} else {
  throw Exception('Error during login: ${response.reasonPhrase}');
}

Future<Map<String, dynamic>> register(String nombre, String apellido, String correo, String contraseña) async {
  final response = await http.post(
    Uri.parse('${baseUrl}/register'),
    headers: {'Content-Type': 'application/json'},
    body: jsonEncode({
```

Fig. 26. Ruta para iniciar sesión desde la aplicación flutter en Android Studio

Se realizan las importaciones en la página del login que funcionará como main y de allí se conectará a las demás interfaces de la aplicación.

La ruta que muestra en la siguiente sirve para poder registrar un nuevo usuario, se pide ingresar un nombre, apellido, correo y contraseña y de la misma forma la información es validada con la base de datos y guardada.

2.5.5 DIAGRAMAS DE CASO DE USO

Los diagramas de casos de uso representan el comportamiento esperado del sistema, sin mostrar el orden de los pasos, para lo cual se utiliza un diagrama de secuencia que indica como interactúan los objetos en el tiempo. La definición del límite del sistema establece que es externo o interno al sistema, un actor simboliza un rol desempeñado por un objeto externo, y un mismo objeto puede asumir varios roles, representándose por varios actores, es decir un caso de uso es una serie de eventos que ocurre cuando un actor utiliza un sistema para completar un proceso, cubriendo regularmente un proceso grande y no un paso individual [50].

Diagrama de caso de uso general para los tipos de usuarios de la aplicación

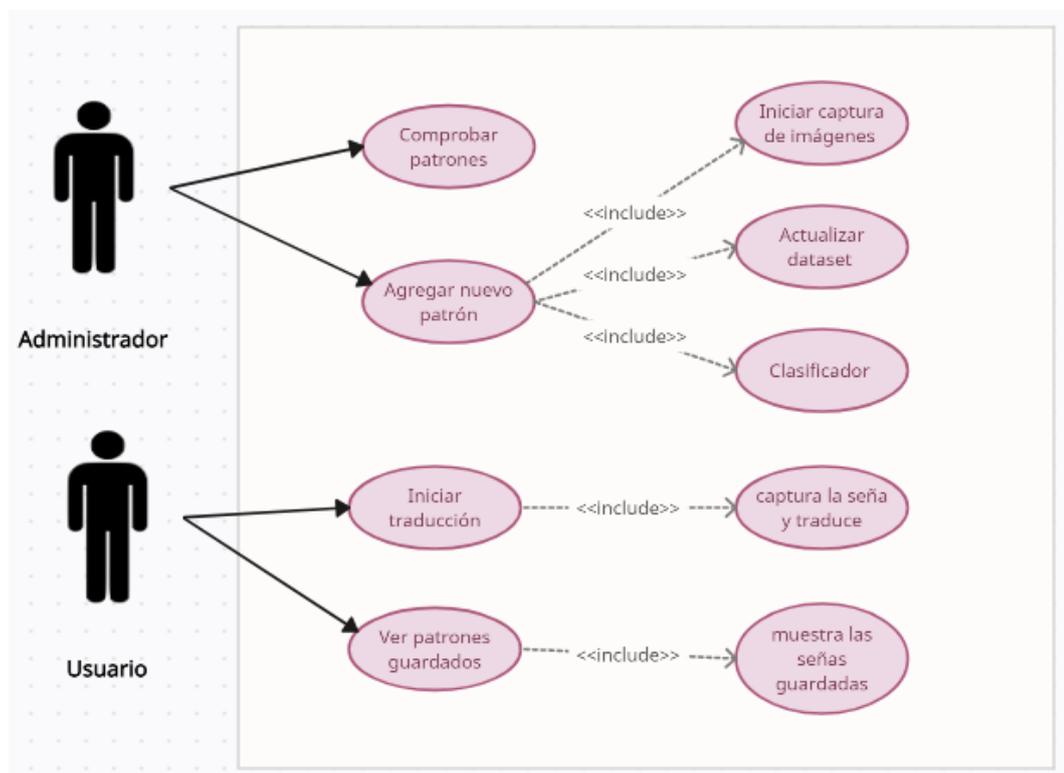


Fig. 27. Diagrama de caso de uso general para los tipos de usuarios de la aplicación

Diagrama de caso de uso para iniciar captura de imágenes

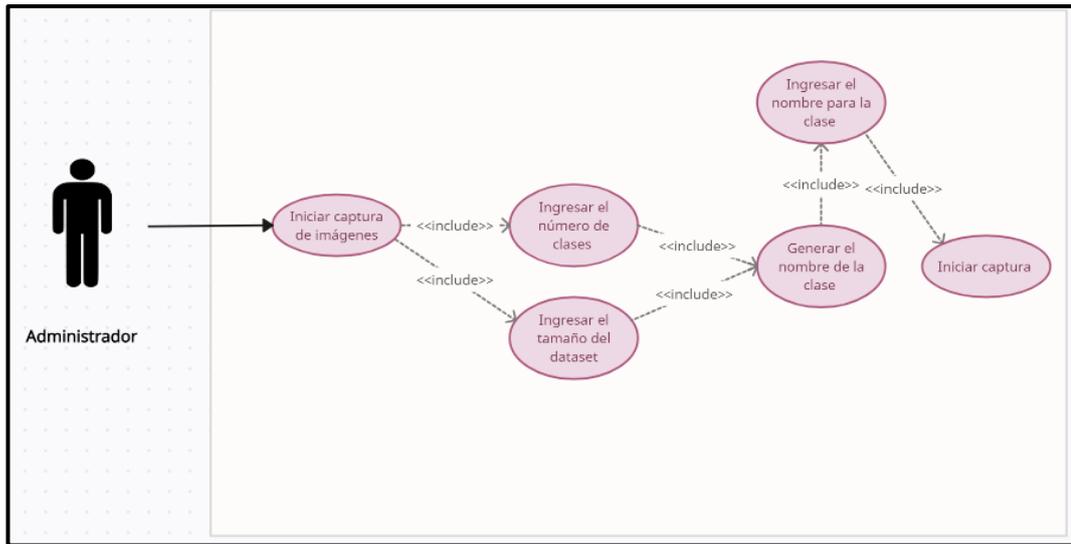


Fig. 28. Diagrama de caso de uso para iniciar capturas

Caso de uso	Capturar imágenes
Actor	Administrador
Descripción	Permite realizar la captura de imágenes para un nuevo patrón de acuerdo con la cantidad del tamaño de dataset y la cantidad de clases creadas.
Evento desencadenador	El administrador debe ingresar la cantidad de patrones y el tamaño del dataset para generar las clases y luego capturarlas.
Pasos realizados	
	<ul style="list-style-type: none"> • Ingresar el número de clases. • Ingresar el tamaño de clases. • Generar el nombre para las clases. • Iniciar captura de las imágenes.
Precondiciones	<ul style="list-style-type: none"> • El número de clases debe ser acorde a lo permitido por la aplicación.

	<ul style="list-style-type: none"> • El tamaño del dataset deber ser acorde al permitido por la aplicación. • El nombre de la clase no debe estar repetido con los guardados en el dataset del servidor.
Postcondiciones	Captura de imágenes guardados en el dataset del servidor.

Tabla 6 Caso de uso para iniciar una captura

Diagrama de caso de uso para actualizar el dataset y clasificar los datos

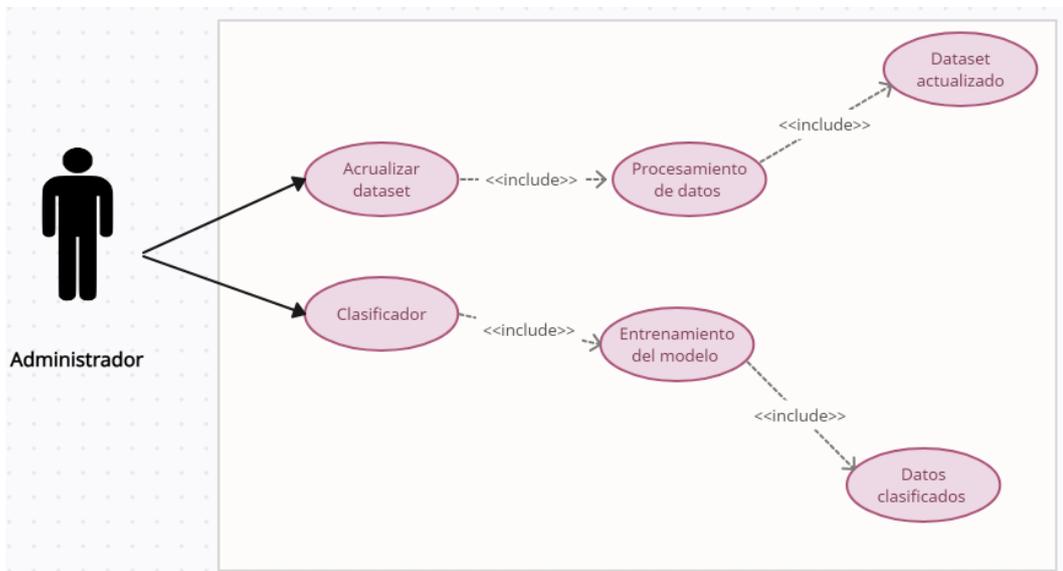


Fig. 29. Diagrama de caso de uso para actualizar el dataset y clasificar los datos

Caso de uso	Actualizar el dataset y clasificar los datos
Actor	Administrador
Descripción	Permite al administrador actualizar el dataset y clasificar los datos cuándo se hayan ingresado nuevos patrones.
Evento desencadenador	Luego de ingresar patrones nuevos se deberá actualizar el dataset y posteriormente clasificar los datos nuevos.
Pasos realizados	

	<ul style="list-style-type: none"> • Oprimir el botón “actualizar dataset”. • Oprimir el botón “clasificador”
Precondiciones	Ninguna
Postcondiciones	<ul style="list-style-type: none"> • El dataset se actualizará. • Los datos serán entrenados con el modelo para la detección y traducción de señas.

Tabla 7 Caso de uso actualizar dataset y clasificar los datos

Diagrama de caso de uso para comprobar los patrones registrados

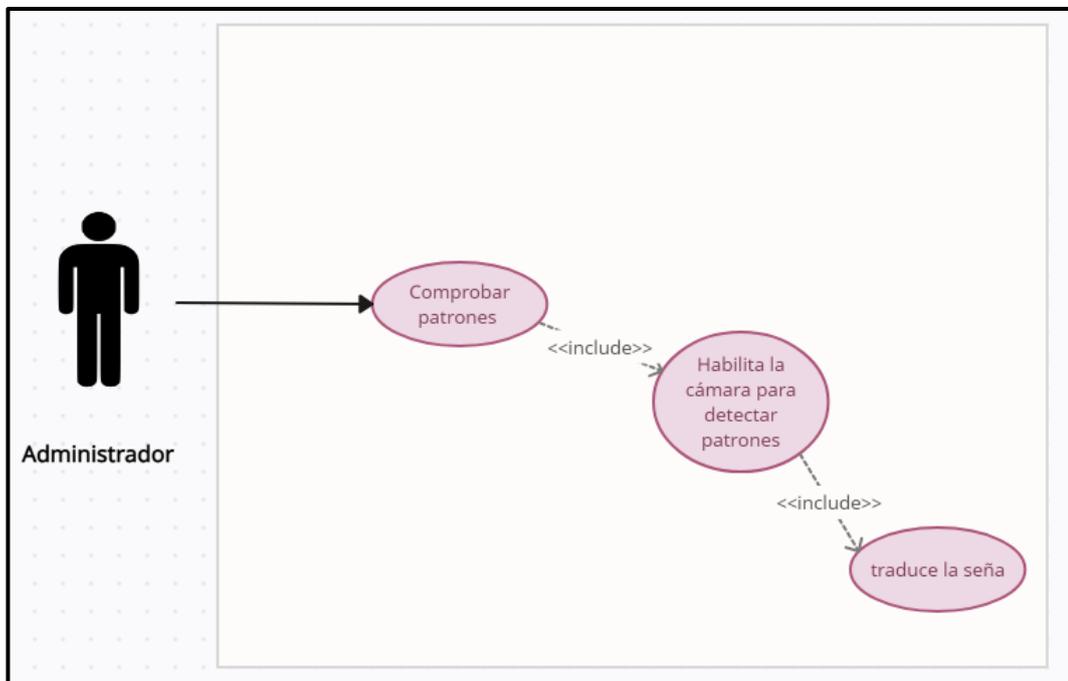


Fig. 30. Diagrama de caso de uso para comprobar los patrones registrados

Caso de uso	Comprobar patrones registrados
Actor	Administrador
Descripción	Permite al administrador verificar los patrones registrados en el servidor.

Pasos realizados	
<ul style="list-style-type: none"> Oprimir el botón “comprobar patrones”. 	
Precondiciones	Ninguna
Postcondiciones	Se podrá verificar si los patrones registrados se guardaron correctamente.

Tabla 8 Caso de uso para comprobar patrones

2.5.6 MODELADO DE DATOS

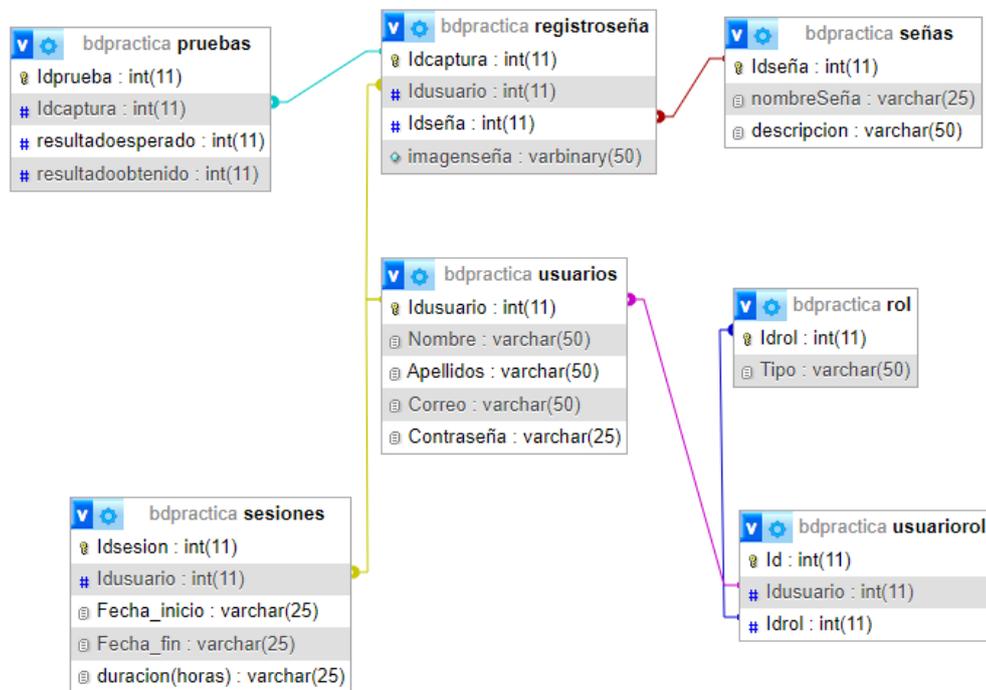


Fig. 31. Modelado de base de datos

2.6 DISEÑO DE LA INTERFAZ



Fig. 32. Interfaz de inicio de sesión

La interfaz de inicio de sesión cuenta con dos cajones para entrada de datos para verificar los datos de los usuarios, un botón para iniciar sesión y un botón de texto para redirigir a la interfaz de registro.

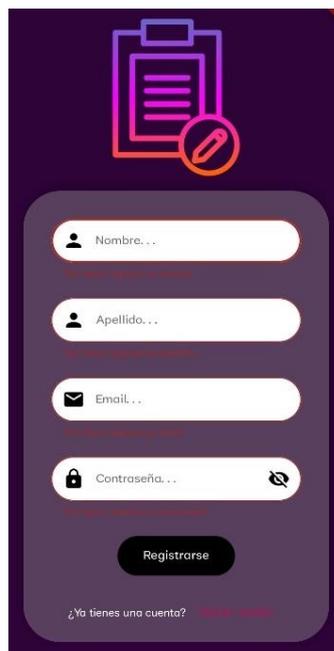


Fig. 33. Interfaz de registro de usuarios

La interfaz de registro de usuarios cuenta con 4 cajones de entrada de texto con el propósito de registrar la información que se ingresen en la base de datos, los cajones están validados en caso una persona no ingrese datos y quiera dar clic en el botón para registrar no envíe datos vacíos a guardar en la base de datos, así como también valores repetidos, el cajón de la contraseña contiene un icono para que al seleccionar se pueda visualizar la contraseña, así como también ocultarla.



Fig. 34. Interfaz de administrador

La interfaz de administrador cuenta con una cámara integrada y dos botones, el propósito de la cámara es que cuando el usuario de clic en el botón “probar patrones” la cámara pueda detectar la seña y devolver una traducción, el segundo botón redirige a la interfaz para entrenar el modelo.

La interfaz para comprobar patrones presenta un frame con el resultado con el nombre de la seña de la traducción realizada por la detección de la mano, además de que muestra los puntos de referencia de la seña que ha sido captada, esto con el fin de verificar si la seña se guardó con éxito.

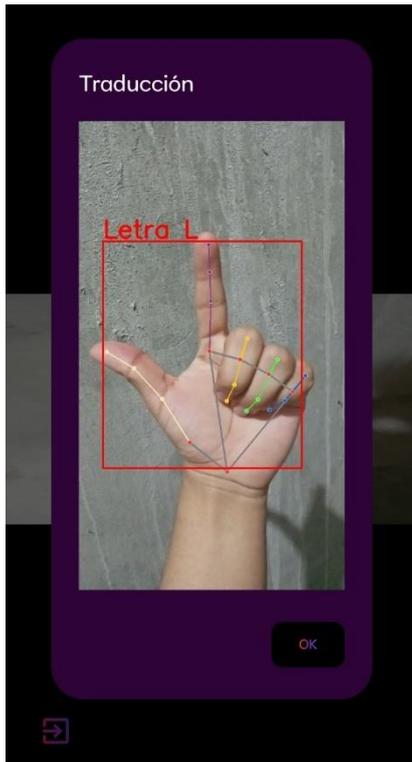


Fig. 36. Traducción realizada con la mano izquierda

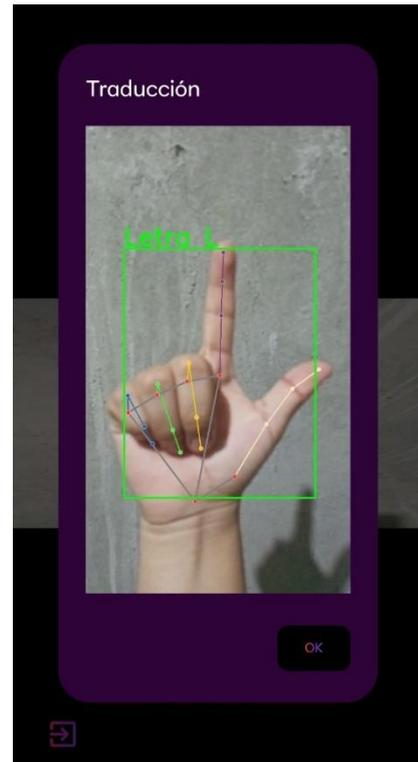


Fig. 35. Traducción realizada con la mano derecha

Aquí se puede visualizar la interfaz donde están las opciones para entrenar el modelo



Fig. 37. Interfaz para el entrenamiento del modelo

La interfaz de entrenamiento del modelo consta de 3 botones, el primero abre la interfaz para capturar datos, el segundo botón es para actualizar el dataset y el tercer botón es para clasificar los datos y terminar el entrenamiento.

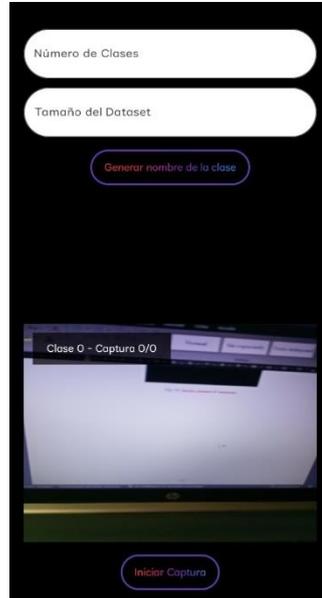


Fig. 38. Interfaz para agregar nuevos patrones

Luego de ello aparecerá una interfaz que contiene dos entradas de textos denominados: Numero de clase y Tamaño del Dataset, un botón para generar nombre para la clase, una cámara integrada para capturar las imágenes y un botón en la parte inferior para iniciar captura.

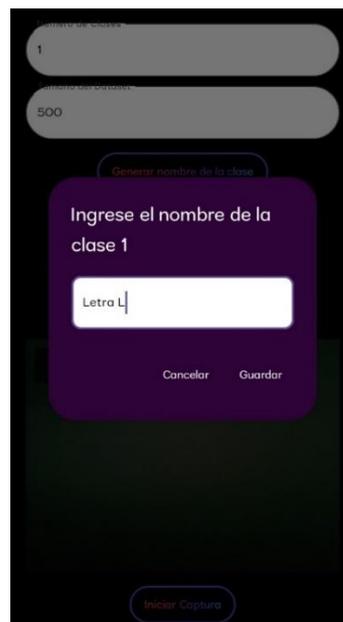


Fig. 39. Interfaz de agregar nuevos patrones al generar nombre de la clase

La interfaz de usuario tiene dos botones, uno es para realizar las traducciones al igual que la que existe en el interfaz de administrador y el otro botón sirve para visualizar una imagen de los patrones guardados.



Fig. 40. Interfaz de usuario

A continuación, se puede observar la interfaz de traducción disponible para que el usuario pueda comenzar a probar la detección de patrones.

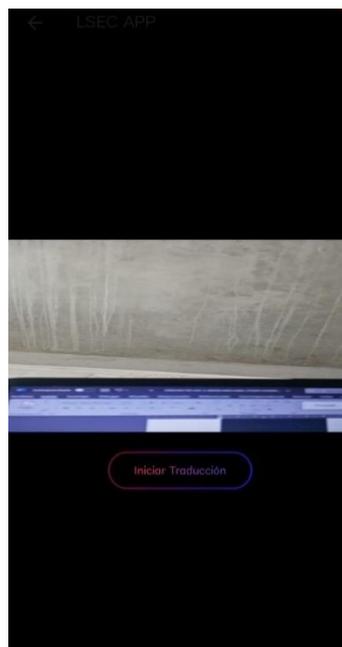


Fig. 41. Interfaz de traducción para el usuario

En la interfaz de traducción se encuentra la cámara lista para ejecutar el proceso de detectar señas, esto hará que la cámara integrada al poner la mano frente a ella pueda detectarla y devolver un frame con la seña traducida mostrando el nombre y los 21 puntos de referencia que posee una mano, mano izquierda color rojo y mano derecha color verde.

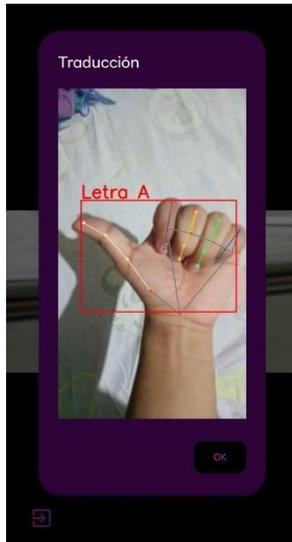


Fig. 43. Traducción mano izq. interfaz usuario

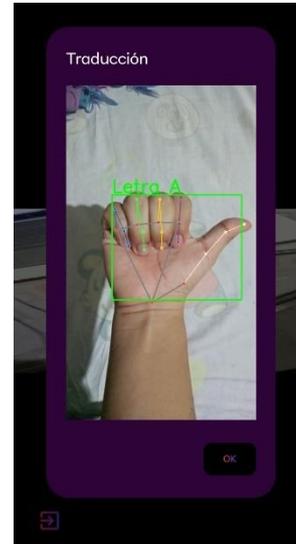


Fig. 42. Traducción mano der. interfaz de usuario

Luego está interfaz para poder visualizar los patrones guardados además de que también se podrá visualizar una imagen de cada patrón.



Fig. 44. Interfaz para ver patrones grabados

En esta interfaz podemos visualizar los diferentes patrones que existen guardados en el servidor y con ello elegir uno para poder ver una imagen de cómo se realiza la seña del patrón seleccionado.



Fig. 45. Menú para ver patrones guardados

2.7 PRUEBAS

Las pruebas de funcionalidad se realizan con el propósito de verificar el correcto funcionamiento de cada una de las partes, así como también opciones del sistema y de esta manera poder encontrar posibles fallos que se encuentren y poder corregirlos para garantizar la ejecución exitosa de cada uno de los procedimientos de la aplicación.

Prueba N° 1: Inicio de sesión en la aplicación móvil	
Objetivo	Verificar que el ingreso de los usuarios a la aplicación funcione de manera correcta.
Descripción	El usuario ingresa las credenciales en los campos correspondientes para acceder a la siguiente interfaz,

Rol	Usuario – Administrador	
Caso N° 1: Usuario con credenciales de inicio de sesión correctos		
Datos de Entrada	Datos de Salida	
<ul style="list-style-type: none"> ✓ Correo electrónico ✓ Contraseña 	<ul style="list-style-type: none"> ✓ La aplicación móvil valida las credenciales del usuario. ✓ La aplicación móvil muestra un mensaje de inicio de sesión exitoso y redirige al usuario a la interfaz correspondiente. ✓ Muestra las funciones disponibles para el usuario. 	
Caso N° 2: Usuario con credenciales de inicio de sesión incorrectas		
Datos de Entrada	Datos de Salida	
<ul style="list-style-type: none"> ✓ Correo electrónico ✓ Contraseña 	<ul style="list-style-type: none"> ✓ La aplicación móvil muestra un mensaje que dice “correo o contraseña incorrectos”. 	
Caso N° 3: Iniciar sesión con los campos vacíos		
Datos de Entrada	Datos de Salida	
Ninguno	<ul style="list-style-type: none"> ✓ La aplicación móvil muestra un mensaje que dice “Por favor ingresa un correo y una contraseña”. 	
RESULTADOS DE LA PRUEBA		
RESULTADOS OBTENIDOS	EVALUACIÓN	
La aplicación móvil permite el inicio de sesión con credenciales correctas.	Exitoso: <input checked="" type="checkbox"/> Fallido: <input type="checkbox"/>	

Tabla 9 Prueba de funcionalidad - Inicio de sesión

Prueba N° 2: Registro de usuarios en la aplicación móvil	
Objetivo	Verificar que los datos de registro de usuarios de la aplicación móvil funcionen y se guarden de manera correcta.
Descripción	El usuario ingresa las credenciales en los campos correspondientes para acceder a la siguiente interfaz.
Rol	Usuario
Caso N° 1: Registro de usuarios correctos	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Nombre ✓ Apellido ✓ Correo ✓ Contraseña 	<ul style="list-style-type: none"> ✓ La aplicación móvil valida los datos ingresados por el usuario. ✓ La base de datos se actualiza y ahora es posible iniciar sesión con los nuevos datos registrados.
Caso N° 2: Registro de usuarios con campos faltantes	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Nombre ✓ Apellido ✓ Contraseña 	<ul style="list-style-type: none"> ✓ En el caso de que en un campo no se hayan ingresado datos y el usuario quiera registrar la información, la aplicación móvil emitirá un mensaje pidiendo que ingrese datos para dicho campo.
Caso N° 3: Registro de usuarios con los campos vacíos	
Datos de Entrada	Datos de Salida

Ninguno	✓ La aplicación móvil pedirá en cada uno de los campos que ingrese datos para que estos puedan ser guardados.
RESULTADOS DE LA PRUEBA	
RESULTADOS OBTENIDOS	EVALUACIÓN
La aplicación móvil permite realizar correctamente el registro de información de los usuarios y de esta forma conseguir las credenciales para iniciar sesión.	Exitoso: <input checked="" type="checkbox"/> Fallido: <input type="checkbox"/>

Tabla 10. Prueba de funcionalidad - registro de usuarios

Prueba N° 3: Detectar y traducir señas	
Objetivo	Verificar que la traducción de los patrones mediante la cámara integrada en la aplicación funcione de manera correcta.
Descripción	Debido a que esta función la tendrán el usuario y el administrador se espera que al oprimir el botón que realiza la opción para traducir señas funcione de manera correcta de acuerdo con cada una de las interfaces correspondiente al tipo de usuario.
Rol	Usuario – Administrador
Caso N° 1: Iniciar traducción	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Oprimir el botón para iniciar la traducción. ✓ Realizar una seña de mano frente a la cámara. 	<ul style="list-style-type: none"> ✓ Detecta la seña ✓ Emite un frame con los puntos de referencia de la mano marcados (21 puntos de referencia).

	✓ Muestra el nombre de la seña traducida.
RESULTADOS DE LA PRUEBA	
RESULTADOS OBTENIDOS	EVALUACIÓN
La aplicación móvil permite la detección de la seña de una mano y devuelve la traducción.	Exitoso: <input checked="" type="checkbox"/> Fallido: <input type="checkbox"/>

Tabla 11. Prueba de funcionalidad - detectar y traducir seña

Prueba N° 4: Agregar un nuevo patrón – Iniciar captura de imágenes	
Objetivo	Verificar que se pueda agregar un nuevo patrón de manera correcta y se pueda realizar la captura de imágenes de este con la cámara del dispositivo móvil y guardarlas en el dataset del servidor.
Descripción	El administrador deberá ingresar todos los campos que permiten registrar un nuevo patrón.
Rol	Administrador
Caso N° 1: Registro correcto de un nuevo patrón	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Numero de patrón ✓ Tamaño del dataset ✓ Nombre del patrón ✓ Captura de imágenes 	<ul style="list-style-type: none"> ✓ La aplicación móvil validará que los datos que se han ingresado para generar un nuevo patrón sean correctos. ✓ La aplicación móvil generará un nuevo input para añadir el nombre del nuevo patrón. ✓ La aplicación móvil comenzará a capturar las imágenes de acuerdo con el tamaño del dataset ingresado para

	que estos sean guardados en el servidor.
Caso N° 2: Registro incorrecto de un nuevo patrón	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Ninguno, datos faltantes o datos repetidos. 	<ul style="list-style-type: none"> ✓ En el caso de que en un campo no se hayan ingresado datos y el administrador quiera generar el nombre para el patrón, la aplicación móvil emitirá un mensaje pidiendo que ingrese datos para dicho campo. ✓ Si en un campo se ingresó valores no permitidos para generar el nombre del patrón, la aplicación móvil emitirá un mensaje pidiendo que ingrese valores correctos. ✓ Si el administrador quiere iniciar la captura de imágenes de un patrón sin tener los campos correctos, la aplicación móvil pedirá que ingrese valores validos antes de querer iniciar con la captura de imágenes. ✓ Si para el nombre del nuevo patrón se ingresó un nombre que ya existe en el dataset, la aplicación móvil mostrará un mensaje diciendo que ya existe un nombre igual y no permitirá iniciar la captura de imágenes.
RESULTADOS DE LA PRUEBA	
RESULTADOS OBTENIDOS	EVALUACIÓN

La aplicación móvil permite agregar correctamente un nuevo patrón e iniciar la captura de imágenes con la cámara del teléfono y guardar las imágenes en el dataset del servidor.	Exitoso: <input checked="" type="checkbox"/> Fallido: <input type="checkbox"/>
--	---

Tabla 12 Caso de funcionalidad - Agregar un nuevo patrón e iniciar captura

Prueba N° 5: Agregar un nuevo patrón – Entrenamiento del modelo	
Objetivo	Verificar que luego de agregar un nuevo patrón de manera correcta, este pueda ser entrenado mediante el modelo de detección de puntos de referencia y traducción.
Descripción	El administrador deberá entrenar el modelo cada que se ingrese un nuevo patrón y este pueda ser traducido, esto se realizará mediante los botones “actualizar dataset” y “clasificador”.
Rol	Administrador
Caso N° 1: Entrenamiento correcto del modelo	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Botón “actualizar dataset” ✓ Botón “clasificador” 	<ul style="list-style-type: none"> ✓ La aplicación móvil mostrará el mensaje “el dataset ha sido actualizado con éxito” cuando se haya oprimido el botón actualizar dataset. ✓ La aplicación móvil mostrará el mensaje “X% de las muestras fueron clasificadas correctamente!” cuando se haya oprimido el botón clasificador.

Caso N° 2: Entrenamiento incorrecto del modelo	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ✓ Botón “actualizar dataset” ✓ Botón “clasificador” 	<ul style="list-style-type: none"> ✓ La aplicación móvil mostrará el mensaje “el dataset no pudo ser actualizado” cuando se haya oprimido el botón actualizar dataset. ✓ La aplicación móvil mostrará el mensaje “no se pudo entrenar el modelo” cuando se haya oprimido el botón clasificador.
RESULTADOS DE LA PRUEBA	
RESULTADOS OBTENIDOS	EVALUACIÓN
La aplicación móvil permite agregar correctamente un nuevo patrón e iniciar con el proceso de entrenamiento con el modelo de detección de puntos de referencia y traducción de señas.	Exitoso: <input checked="" type="checkbox"/> Fallido: <input type="checkbox"/>

Tabla 13 Caso de funcionalidad - Agregar un nuevo patrón y entrenarlo

Prueba N° 6: Ver las imágenes de los patrones existentes	
Objetivo	Verificar que el usuario pueda observar cómo se realiza la seña de los patrones que se han guardado para que utilice con más facilidad la función de traducir.
Descripción	El usuario podrá ver como se realiza la seña de los patrones que se han guardado en el dataset haciendo clic en el botón “ver patrones grabados”.
Rol	Usuario

Caso N° 1: Ver patrones grabados	
Datos de Entrada	Datos de Salida
✓ Botón “ver patrones grabados	✓ La aplicación móvil mostrará los patrones grabados con su nombre cuando el usuario haya oprimido el botón ver patrones grabados.
RESULTADOS DE LA PRUEBA	
RESULTADOS OBTENIDOS	EVALUACIÓN
La aplicación móvil permite observar al usuario una imagen de los patrones grabados y el nombre que tiene.	Exitoso: <input checked="" type="checkbox"/> Fallido: <input type="checkbox"/>

Tabla 14 Prueba de funcionalidad - Ver las imágenes de los patrones existentes

2.8 RESULTADOS

Las pruebas se hicieron con el objetivo de comprobar el funcionamiento general de la aplicación en relación con el módulo de registro de nuevos patrones (detección de una nueva seña), donde se toma la muestra de una cantidad de 500 imágenes por seña para que el reconocimiento del nuevo patrón sea más preciso, esto ayudaría a cubrir el ángulo, iluminación entre otras condiciones que permitirán obtener un mejor resultado.

También era necesario comprobar el funcionamiento del módulo de traducción de la aplicación con el objetivo de determinar si el modelo cumple la función de traducir un gesto de acuerdo con el entrenamiento de la seña, para ello se hizo la prueba con la detección de un patrón de cada uno de los datos existentes por un total de 10 veces, usando la mano derecha para las primeros 5 detecciones y luego la mano izquierda para las últimos 5 y de esta forma poder observar cuáles son los resultados que presenta.

A continuación, se muestra la tabla con los resultados obtenidos:

Resultados de las pruebas de funcionalidad al ingresar un nuevo patrón

Seña	Exitoso	
	Cantidad de imágenes	% del entrenamiento
Letra L	500	96.77
Bien	500	97.59
Letra A	500	97.44
Letra B	500	95.60

Tabla 15 Resultados de prueba funcional de agregar patrones

Se realizaron pruebas por medio de la aplicación móvil, donde se realizó la captura de 500 imágenes de cada nueva seña o patrón que se guardará en el dataset del servidor y se realizó la validación con respecto a las respuestas mostradas en la aplicación con el porcentaje del entrenamiento del modelo de acuerdo con cada proceso de seña guardada obteniendo los siguientes resultados:

Para la primera seña “Letra L” guardada se tomó un total de 500 muestras (captura de imágenes), se guardó en el dataset y se realizó el proceso de entrenamiento del modelo, teniendo un total de 96.77% de muestras clasificadas de manera correcta.

Para la segunda seña guardada “Bien” se tomó un total de 500 muestras (captura de imágenes), se guardó en el dataset y se realizó el proceso de entrenamiento del modelo, teniendo un total de 97.59% de muestras clasificadas de manera correcta.

Para la tercera seña guardada “Letra A” se tomó un total de 500 muestras (captura de imágenes), se guardó en el dataset y se realizó el proceso de entrenamiento del modelo, teniendo un total de 97.44% de muestras clasificadas de manera correcta.

Para la cuarta seña guardada se tomó un total de 500 muestras (captura de imágenes), se guardó en el dataset y se realizó el proceso de entrenamiento del modelo, teniendo un total de 95.60% de muestras clasificadas de manera correcta.

A continuación, se muestra una tabla con los resultados de las pruebas obtenidas al momento de traducir una seña:

Resultados de las pruebas de funcionalidad al traducir una seña

Seña	Exitoso		Fallido		No ejecutado	
	Cantidad de imágenes	%	Cantidad de imágenes	%	Cantidad de imágenes	%
Letra L	10	100	0	0	0	0
Bien	9	90	1	10	0	0
Letra A	10	100	0	0	0	0
Letra B	10	100	0	0	0	0

Tabla 16 Resultados de prueba funcional del módulo de traducción

Para el módulo de traducción de señas también se realizaron pruebas que verifiquen el correcto funcionamiento, para este caso se realizó la detección de una seña en concreto y en conjunto con el servidor validar la respuesta que este envía a la aplicación obteniendo los siguientes resultados:

Para la primera seña “Letra L” se realizó la detección por un total de 10 veces en donde se obtuvo un total de 100% exitoso, un total de 0% fallido y un total de 0% de traducciones no ejecutadas.

Para la segunda seña se realizó la detección por un total de 10 veces en donde se obtuvo un total de 90% exitoso, un total de 10 % fallido y un total de 0% de traducciones no ejecutadas.

Para la tercera seña se realizó la detección por un total de 10 veces en donde se obtuvo un total de 100% exitoso, un total de 0% fallido y un total de 0% de traducciones no ejecutadas.

Para la cuarta seña se realizó la detección por un total de 10 veces en donde se obtuvo un total de 100% exitoso, un total de 0% fallido y un total de 0% de traducciones no ejecutadas.

Se realizó una encuesta en el Centro de Educación Integral “Melvin Jones” para un

total de 9 docentes que son quienes se encuentran trabajando actualmente en la institución, la encuesta fue efectuada con la finalidad de obtener información de si las personas hacen uso de aplicaciones educativas así como también aplicaciones educativas para aprender el alfabeto de lengua de señas ecuatoriano, que ventajas podría presentar el uso de una aplicación con respecto a la enseñanza del lenguaje de señas tradicional y también si la aplicación desarrollado funciona de manera correcta.

La encuesta podrá ser vista en el [Anexo 2].

Mediante el siguiente análisis de la encuesta se pudo obtener los siguientes resultados: la cantidad de docentes que poseen un teléfono móvil fue del 100%, siendo este el porcentaje completo con el que las personas podrán hacer uso de la aplicación final que servirá como herramienta para aprender el alfabeto del lenguaje de señas ecuatoriano, que los docentes que usan aplicaciones educativas en su teléfono móvil como métodos de enseñanza es de alrededor del 78% demostrando que el uso de aplicaciones es una herramienta para el docente, mientras que el porcentaje restantes no usa aplicaciones educativas en su teléfono móvil.

La frecuencia en el que el docente hace uso de aplicaciones educativas fue de un 62% empleando este tipo de herramienta todos los días ya sea de aprendizaje autónomo o como herramienta de aprendizaje para sus estudiantes, a pesar de que los docentes hacen uso de aplicaciones educativas solo el 11% usa una aplicación para aprender el alfabeto de la lengua de señas ecuatoriano, se conoció que una ventaja considerable para el 67% de personas encuestadas es la disponibilidad de horarios esto debido a que la aplicación móvil podrá ser usada las 24 horas del día, mientras que las clases para aprender palabras del lenguaje de señas es de 4 horas por jornada.

Dado que la aplicación móvil “LSECAPP” está disponible las 24 horas del día y con el objetivo de conocer cuál sería el momento adecuado en que una persona le gustaría hacer uso de la aplicación y aprender palabras se obtuvo como resultado que el 60% de los encuestados prefiere aprender en la noche, el 30% en la tarde y solo el 10% de día demostrando que la noche sería el mejor momento para usar la app y aprender nuevas señas.

El 100% de los encuestados pudo hacer uso de la aplicación de manera correcta sin problema alguno desde el lugar donde se encontraban, se conoció que para el 67% fue fácil descargar e instalar la aplicación, mientras que para el 22% presento una dificultad normal y para el 11% restante fue difícil, se determinó que la dificultad que tienen los usuarios en usar la aplicación para el 78% fue fácil, mientras que para el 22% restante tuvieron una dificultad considerada como normal al momento de hacer uso de ella.

Con el propósito de conocer la satisfacción del usuario con cada una de las funciones mostradas en la aplicación así como también cada uno de los aspectos de la misma se elaboró una pregunta que servirá como puntuación abarcando cada uno de los puntos tanto disponibilidad, sencillez y funcionamiento teniendo en cuenta una escala de 1 a 5 (1 = muy insatisfecho, 2 = satisfecho, 3 = está bien, 4 = satisfecho, 5 = muy satisfecho), en la que se obtuvo que para el 67% de los encuestados estuvo “muy satisfecho” y el 33% restante quedó “satisfecho” luego de haber usado la aplicación móvil “LSECAPP”.

CONCLUSIONES

- ❖ Se pudo cumplir con el objetivo de crear un modelo de detección y traducción de señas acorde a señas existentes en el lenguaje de señas ecuatoriano usando MediaPipe Solutions bajo las siguientes condiciones: la captura de datos deben ser señas estáticas de una sola mano para que posteriormente se pueda realizar el entrenamiento del modelo, permitiendo clasificar las muestras y así poder determinar un porcentaje de efectividad.
- ❖ Luego de realizar la captura de datos y actualizar el dataset de las muestras tomadas para cada nuevo patrón, estas fueron clasificadas usando la librería de scikit-learn mostrando de esta manera que cantidad de porcentaje fue clasificada de forma correcta.
- ❖ Se creó correctamente la API que permite la comunicación bidireccional entre la aplicación móvil, el servidor y la base de datos, enviando solicitudes http desde la aplicación móvil desarrollada en flutter hacia la API del servidor.
- ❖ Se desarrollaron correctamente cada uno de los módulos establecidos para el funcionamiento de la aplicación móvil, con una interfaz sencilla e intuitiva para que pueda ser usado de manera correcta, teniendo un registro para nuevos usuarios y un inicio de sesión para los usuarios ya establecidos, con funciones para administrador y funciones para usuarios no administradores.
- ❖ Para realizar las capturas de las imágenes y detección usando la librería OpenCv fue necesario decodificar la imagen en formato base64, luego de esto convertirlo en un array de Numpy para posteriormente decodificar el array de Numpy en una imagen usando opencv y finalmente guardar la imagen captada en la ruta especificada.
- ❖ Fue posible la creación de una aplicación móvil para dispositivos Android que se use como una herramienta de aprendizaje donde el usuario pueda visualizar patrones guardados, aprenderlos y luego con el uso del traductor de la aplicación verificar que la postura realizada sea detectada obteniendo la traducción en texto del significado de la seña.

RECOMENDACIONES

- ❖ Para usar la aplicación de manera correcta se recomienda revisar el manual de usuario donde se detallan cada una de las funciones para el administrador y para el usuario.
- ❖ Cuando se comienza a realizar las capturas si bien la cámara detecta cuando un lugar carece de luz y enciende al flash para tomar fotos no es lo más recomendable hacer ya que las imágenes captadas suelen ser opacadas por una luz producto del flash, lo recomendable es realizar el proceso para guardar un nuevo patrón en un ambiente claro para evitar el ruido, oscuridad y opacidad en las imágenes.
- ❖ Se recomienda al administrador que al momento de realizar el proceso de captura para el reconocimiento de las imágenes la cantidad de muestra que tome (capturas tomadas) sea de 500 imágenes como mínimo, esto debido a que la cantidad de muestras con las que trabaja una aplicación de aprendizaje automático pequeño (5 a 10 palabras) el mínimo es de 10, mientras que uno que trabaja con (20 a 30 palabras) es de 500 a 1000 capturas.
- ❖ Con el propósito de que las funciones de la aplicación móvil se ejecuten de manera correcta es necesario darle permiso a la cámara del dispositivo además de tener acceso a conexión de internet.

REFERENCIAS

- [1] P. Cobeñas, «Exclusión educativa de personas con discapacidad: Un problema pedagógico,» *REICE: Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, vol. 18, nº 1, pp. 65-81, 2020.
- [2] C. P. Dussan, «Educación inclusiva: Un modelo de educación para todos.,» *ISEES: Inclusión Social y Equidad en la Educación Superior*, nº 8, pp. 73-84, 2010.
- [3] W. P. Gavilanes Proaño, «Sistema automático de decodificación de lengua de señas ecuatoriano a comunicación verbal,» Bachelor's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Telecomunicaciones, 2023.
- [4] A. A. Ochoa Leyva, «Aplicación Android con Realidad Aumentada para el Manual de Lengua de Señas Cubana,» (Bachelor's thesis, Universidad de las Ciencias Informáticas. Facultad 4), 2022.
- [5] G. Pinzón-Bayona y Y. G. Sandra-Orjuela, «Desarrollo de una aplicación móvil para traductor de lenguaje de señas mediante el uso de servicios web,» 2022.
- [6] U. E. P. d. S. Elena, «Resolución RCF-FST-SO-09 No. 03-2021,» La Libertad, 2021.
- [7] C. A. Montenegro Cachay y D. R. Villa Rodríguez, «Sistema inteligente de reconocimiento de lenguaje de señas peruano para mejorar la comunicación entre las personas sordomudas de la Institución Educativa Bautista para sordos Harvest en Chiclayo.,» 2020.

- [8] C. N. p. I. I. d. Discapacidades, «Diccionario de Lengua de Señas Ecuatoriano “Gabriel Román”,» [En línea]. Available: <https://www.consejodiscapacidades.gob.ec/diccionario-de-lengua-de-senas-ecuadoriano-gabriel-roman/>. [Último acceso: 17 Octubre 2023].
- [9] D. Martínez-Acosta, E. Suárez-Brieva y Y. Gordon-Hernández, «Aplicación móvil como estrategia de enseñanza para iniciar el proceso de lectura a estudiantes en condición de discapacidad auditiva.,» *Información tecnológica*, vol. 33, n° 4, pp. 1-12, 2022.
- [10] D. D. I. Pérez, L. A. Huamán, M. R. A. Yllanes y J. R. D. Dumont, «Aplicación Móvil educativa para facilitar el aprendizaje de la Lengua de Señas del Perú en personas sordomudas.,» *TAYACAJA*, vol. 3, n° 2, 2020.
- [11] Secretaría Nacional de Planificación, Plan de Creación de Oportunidades 2021-2025, Quito, 2020, pp. 1-120.
- [12] CONADIS, «Diccionario de Lengua de Señas Ecuatoriana Gabriel Román,» [En línea]. Available: <http://www.plataformaconadis.gob.ec/~platafor/diccionario/>.
- [13] N. Morales, «Investigación exploratoria: tipos, metodología y ejemplos.,» 2015. [En línea]. Available: <https://www.lifeder.com/investigacion-exploratoria..>
- [14] R. Hernández-Sampieri y C. Mendoza, Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta., 2020.
- [15] C. A. R. Galarza, «Los alcances de una investigación,» *CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica*, vol. 9, n° 3, pp. 1-6, 2020.

- [16] G. González, «Investigación diagnóstica: características, técnicas, tipos, ejemplos,» 2020. [En línea]. Available: <https://www.lifeder.com/investigacion-diagnostica/>.
- [17] C. L. Escudero Sánchez y L. A. Cortez Suárez, Técnicas y métodos cualitativos para la investigación científica, Ecuador: UTMACH, 2018.
- [18] D. Ramos, R. Noriega, J. R. Laínez y A. Durango, Curso de Ingeniería de Software: 2ª Edición., IT Campus academy., 2017.
- [19] C. MELVIN JONES, «U.E.E. MELVIN JONES "QUIENES SOMOS",» [En línea]. Available: <https://ceimelvinjones.wordpress.com/quienes-somos/>. [Último acceso: 19 Septiembre 2023].
- [20] Infoescuelas, «UNIDAD DE EDUCACION ESPECILIZADA MELVIN JONES en LA LIBERTAD,» [En línea]. Available: <https://www.infoescuelas.com/ecuador/santa-elena/unidad-de-educacion-especilizada-melvin-jones-en-la-libertad/>.
- [21] V. Alvear-Puertas, P. Rosero-Montalvo, D. Peluffo-Ordóñez y J. Pijal-Rojas, «Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura.,» *Enfoque UTE*, vol. 8, pp. 244-256, 2017.
- [22] J. C. Ponce Gallegos, A. Torres Soto, F. S. Quezada Aguilera, A. Silva Sprock, E. U. Martínez Flor, A. Casali, ... y O. Pedreño, Inteligencia artificial, Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.
- [23] C. M. P. Pertuz, Aprendizaje automático y profundo en python, Ra-Ma Editorial., 2022.
- [24] A. T. Norman, Aprendizaje automático en acción, Litres, 2019.

- [25] B. Nguyen Cong, J. L. Rivero Pérez y C. Morell, «Aprendizaje supervisado de funciones de distancia: estado del arte.,» *Revista Cubana de Ciencias Informáticas*, vol. 9, nº 2, pp. 14-28, 2015.
- [26] F. A. González, «Modelos de aprendizaje computacional en reumatología,» *Revista Colombiana de Reumatología*, vol. 22, nº 2, pp. 77-78, 2015.
- [27] Y. C. Fonseca-Reyna, Y. Martínez-Jiménez y A. Nowé, «Aprendizaje reforzado aplicado a la programación de tareas bajo condiciones reales,» *Ingeniería Industrial*, vol. 39, nº 1, pp. 36-45, 2018.
- [28] A. re:Invent, «¿Cuál es la diferencia entre el front end y back end en el desarrollo de aplicaciones?,» [En línea]. Available: <https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/>.
- [29] Python, «¿Qué es Python?,» [En línea]. Available: <https://www.python.org>.
- [30] R. Hat, «¿Qué es una API y cómo funciona?,» [En línea]. Available: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>.
- [31] Microsoft, «Visual Studio Code,» [En línea]. Available: <https://visualstudio.microsoft.com/es/#vscode-section>.
- [32] phpMyAdmin, «Bringing MySQL to the web,» [En línea]. Available: <https://www.phpmyadmin.net>.
- [33] A. friends, «XAMPP Apache + MariaDB + PHP + Perl,» [En línea]. Available: <https://www.apachefriends.org/es/index.html>.
- [34] a. i. o. technology, «¿Qué es Frontend?,» [En línea]. Available: <https://assemblerinstitute.com/blog/backend-vs-frontend/>.

- [35] A. C. & Services, «¿QUÉ ES UN FRAMEWORK? TIPOS DE,» [En línea]. Available: <https://admcloudservices.com/blog/que-es-un-framework-tipos-de-frameworks/>.
- [36] Flutter, «Build for any screen,» [En línea]. Available: <https://flutter.dev/>.
- [37] A. Marzo Peña, X. Rodríguez Fleitas y M. M. Fresquet Pedroso, « La lengua de señas. Su importancia en la educación de sordos,» *Varona. Revista Científico Metodológica*, nº 75, 2022.
- [38] A. R. Villa, «El lenguaje de señas en un mundo globalizado,» *Blog Digit. Univ*, pp. 1-12, 2021.
- [39] O. team, «OpenCV - Open Computer Vision Library,» [En línea]. Available: <https://opencv.org>.
- [40] Google, «Mediapipe,» [En línea]. Available: <https://developers.google.com/mediapipe>.
- [41] A. Vakunov, C. L. Chang, F. Zhang, G. Sung, M. Grundmann y V. Bazarevsky, «Mediapipe hands: On-device real-time hand tracking,» de *CVPR Workshop*, 2020.
- [42] G. Developers, «Hand landmarks detection guide,» [En línea]. Available: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker.
- [43] R. O. Klenzi, M. I. Masanet, F. Recabarren, S. Saez y G. Conturso, «Machine learning y deep learning en la interpretación del lenguaje de señas,» In *XXV Workshop de Investigadores en Ciencias de la Computación*, 2023.
- [44] L. C. Lanzarini, C. A. Estrebou, F. Ronchetti, F. Quiroga, C. Luna, R. .. Antonio y A. Rosete, «Aprendizaje automático: aplicaciones en visión por

computadora.» In XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste, 2018.

- [45] C. Ortiz-Leon, F. Yupanqui-Allcca y B. Meneses-Claudio, «Uso de la Inteligencia Artificial para la traducción de lenguajes de señas: una revisión sistemática de literatura,» *Salud, Ciencia y Tecnología-Serie de Conferencias*, vol. 2, pp. 446-446, 2023.
- [46] C. Hernández, J. L. Pulido y J. E. Arias, «Las tecnologías de la información en el aprendizaje de la lengua de señas,» *Revista de Salud Pública*, vol. 17, pp. 61-73, 2015.
- [47] C. Lugaresi, J. N. H. Tang, C. McClanahan, E. H. M. Uboweja, ... y M. Grundmann, «Mediapipe: A framework for perceiving and processing reality,» *In Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, vol. 2019, 2019.
- [48] M. Recalde, A. Losito, M. J. Morillo, C. Pancho, J. B. Reyes y J. Aranas, «Creating an Accessible Future: Developing a Sign Language to Speech Translation Mobile Application with MediaPipe Hands Technology,» *International Conference on ICT for Smart Society (ICISS)*, pp. 1-6, 2023.
- [49] C. Bircher, «Arquitectura Cliente-Servidor/Cliente-Servidor en Aplicaciones de Misión Crítica,» 2017.
- [50] Microsoft, «Crear un diagrama de casos de uso UML,» [En línea]. Available: <https://support.microsoft.com/es-es/topic/crear-un-diagrama-de-casos-de-uso-uml-92cc948d-fc74-466c-9457-e82d62ee1298#:~:text=Los%20diagramas%20de%20casos%20de,externo%20o%20interno%20del%20sistema..>

ANEXOS

Anexo 1 Formato de entrevista para la directora del CEI Melvin Jones

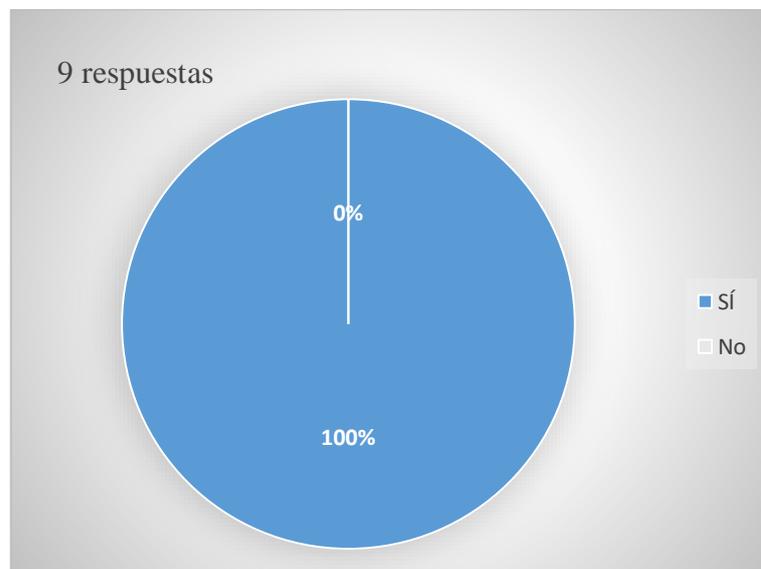
Objetivo	Conocer el estado actual de la institución y los métodos de aprendizaje con los que cuenta el Centro de Educación Integral Melvin Jones.
N°	Pregunta.
1	<p>¿Qué tipo de servicios y que áreas de tratamiento ofrecen a los padres que desean inscribir a sus hijos al CEI Melvin Jones?</p> <hr/> <hr/> <hr/> <hr/>
2	<p>¿Cómo se encuentra conformado su equipo de trabajo y personal docente?</p> <hr/> <hr/> <hr/> <hr/>
3	<p>Siendo una institución sin fines de lucros ¿Cuáles son los problemas que enfrenta al brindarle tratamientos y educación a una gran cantidad de estudiantes?</p> <hr/> <hr/> <hr/> <hr/>

4	<p>¿Qué tipo de capacitación reciben los voluntarios que desean prestar sus servicios para ayudar con el tratamiento de niños, niñas y jóvenes de la institución?</p> <hr/> <hr/> <hr/> <hr/>
5	<p>¿Qué tipo de rol cumplen los padres de los estudiantes dentro y fuera del centro educativo y como esto ayuda en el proceso de enseñanza que tienen sus hijos?</p> <hr/> <hr/> <hr/> <hr/>
6	<p>¿Cuáles son los métodos y enfoques educativos que se han implementado dentro del centro para fomentar la inclusión y obtener éxito académico en los estudiantes con discapacidad auditiva?</p> <hr/> <hr/> <hr/> <hr/>
7	<p>¿Cómo es evaluado el aprendizaje y progreso individual que ha tenido el estudiante en el transcurso de su estancia en centro integral?</p> <hr/> <hr/>

8	<p>¿La institución cuenta con tecnología que ayude a mejorar el aprendizaje y tratamiento de los estudiantes?</p> <hr/> <hr/> <hr/> <hr/>
9	<p>¿Conoce cuáles son los beneficios que la tecnología puede brindar a un centro que se especialice en atender a niños con discapacidades?</p> <hr/> <hr/> <hr/> <hr/>

Anexo 2. Encuesta de disponibilidad y uso de la aplicación móvil para traducir el alfabeto de lengua de señas

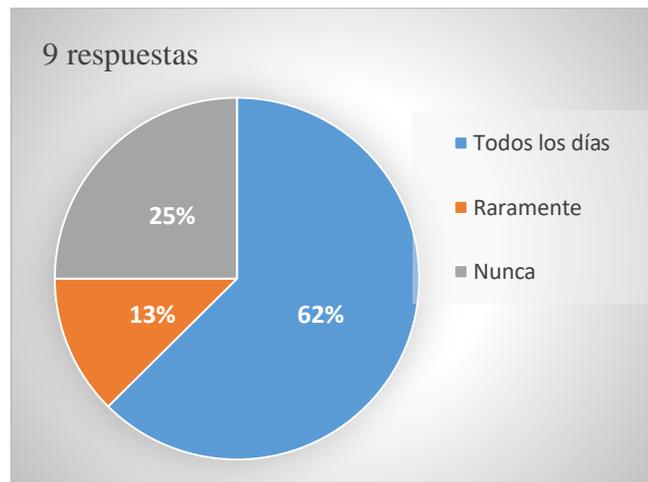
1.- ¿Usted tiene un dispositivo móvil?



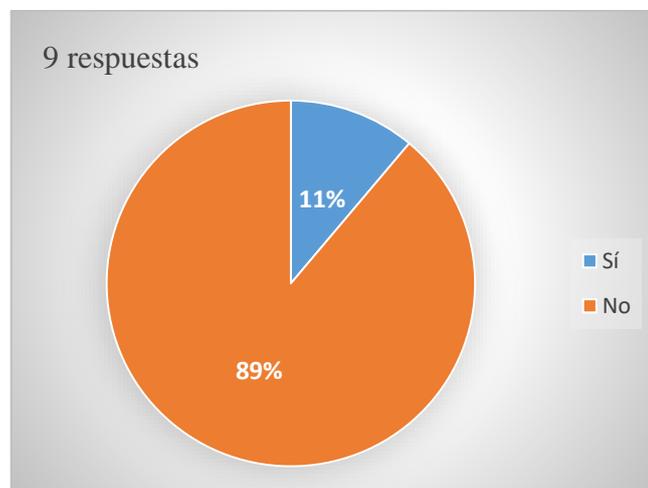
2.- ¿Ha utilizado aplicaciones educativas en su dispositivo móvil?



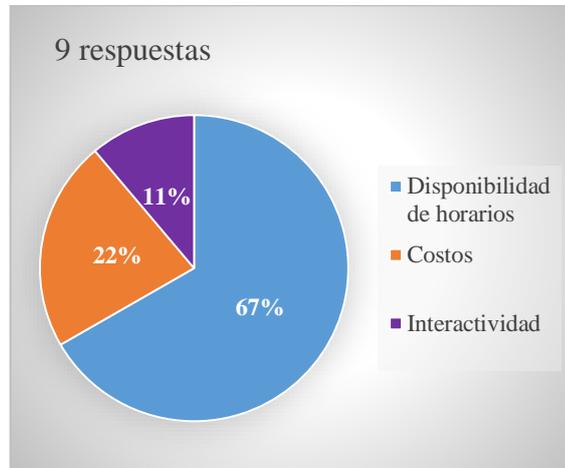
3.- ¿Con qué frecuencia usa aplicaciones educativas en su dispositivo móvil?



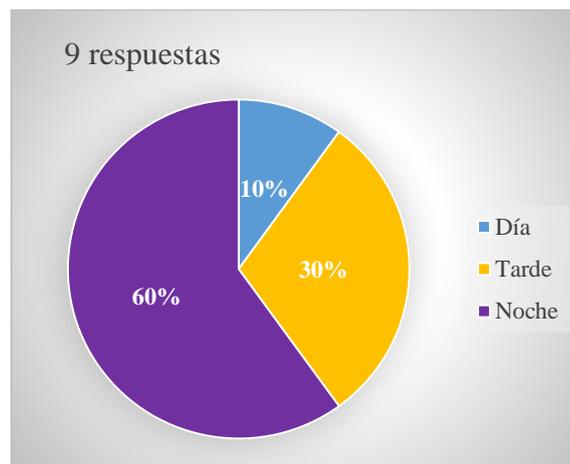
4.- ¿Ha utilizado una aplicación educativa para aprender el alfabeto de lengua de señas ecuatoriano en su dispositivo móvil?



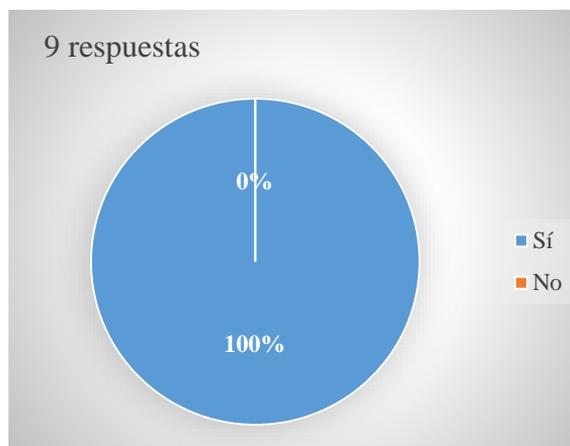
5.- ¿Qué ventajas ve en el uso de una aplicación educativa para aprender el lenguaje de señas en comparación con las clases tradicionales?



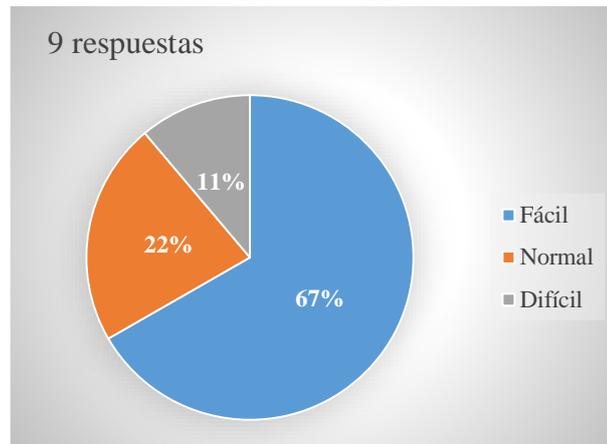
6.- Fuera de las horas regulares que imparte clase ¿Preferiría usar la aplicación para aprender lenguaje de señas durante la mañana, tarde o noche?



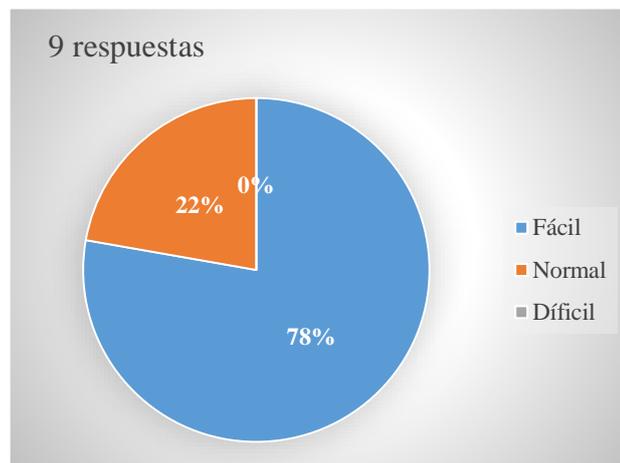
7.- ¿Fue posible usar la aplicación “LSECAPP” en cualquier hora del día?



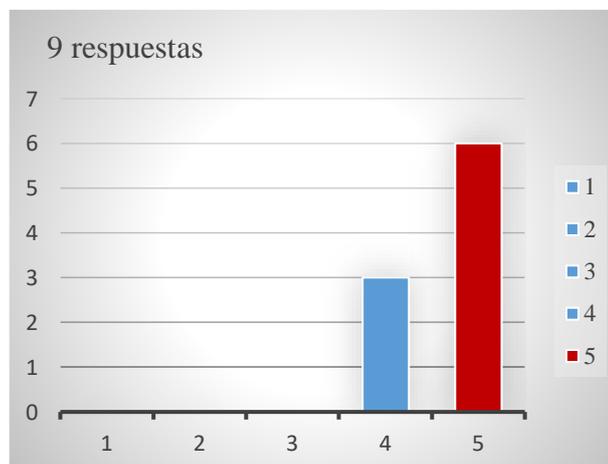
8.- ¿Cuál fue la dificultad que tuvo para descargar la aplicación “LSECAPP”



9.- ¿Cuál fue la dificultad que tuvo para usar las funciones que ofrece la aplicación “LSECAPP”



10.- En una escala del 1 al 5, ¿cuán satisfecho estás con el aprendizaje del alfabeto de lenguaje de señas a través de aplicación móvil “LSECAAP”? (1 = muy insatisfecho, 2 = satisfecho, 3 = está bien, 4 = satisfecho, 5 = muy satisfecho)



Anexo 3. Manual de usuario para usuario administrador

Aplicación móvil en Android para detectar y traducir señas estáticas de una mano

El siguiente manual describe los pasos para el uso correcto de la aplicación.

1. Objetivo.

Este manual tiene como propósito el guiar de manera correcta cada uno de los pasos y funcionalidades que puede emplear el usuario administrador.

2. Uso de la aplicación móvil Inicio de sesión.

Paso 1: El usuario administrador deberá ingresar las credenciales en los campos de entrada de texto que se muestra en la figura:



Fig. 46. Interfaz de usuario admin



Fig. 47. Inicio de sesión - Usuario administrador

Paso 2: El usuario administrador deberá ingresar las siguientes credenciales para ingresar la interfaz de administrador.

Credenciales de Usuario:

Correo: admin

Contraseña: admin

El proceso debería ser el siguiente como se muestra a continuación en la figura Fig. 49.

Con el símbolo del “ojito” en la parte derecha del cajón para ingresar la contraseña se puede visualizar la contraseña dándole clic como se muestra en la figura Fig. 48, de igual forma si se vuelve a dar clic sobre el símbolo se esconde la contraseña nuevamente.



Fig. 49. Ingresar credenciales de administrador



Fig. 48. Ver contraseña

Paso 3: Dar clic en el botón **“Iniciar sesión”** esto redirigirá al usuario a la interfaz de administrador



Fig. 50. Interfaz para iniciar sesión

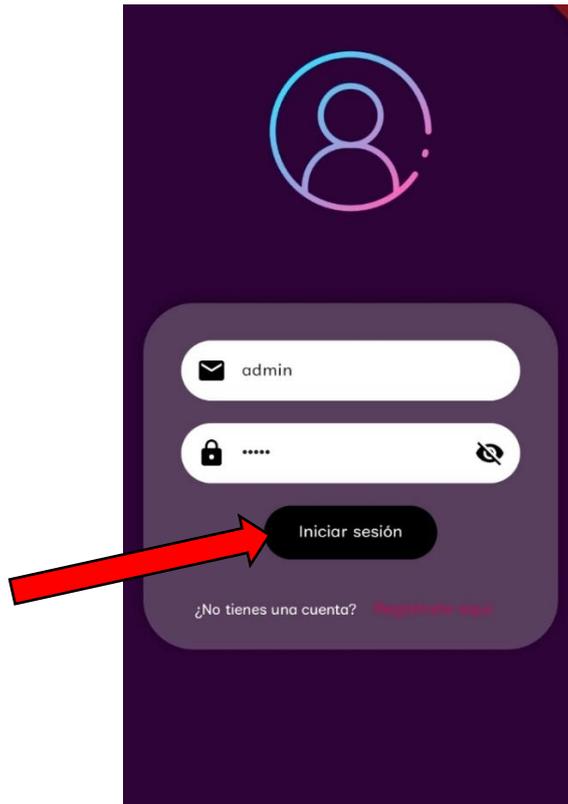


Fig. 51. Oprimir el botón iniciar sesión

Interfaz administrador

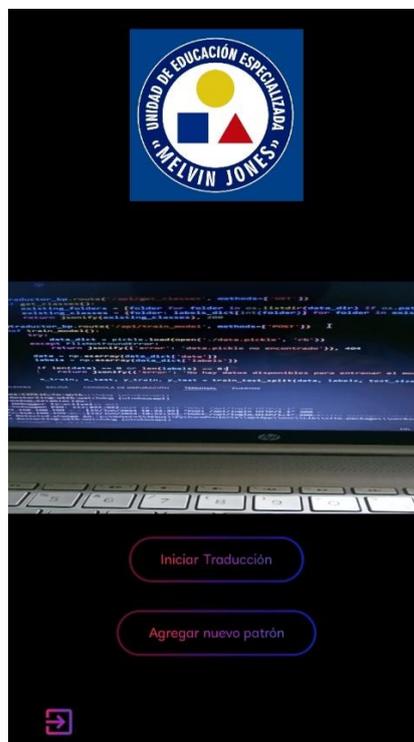


Fig. 52. Interfaz administrador manual de usuario

Paso 4 Grabar un nuevo patrón: Una vez adentro en el interfaz administrador muestra una cámara integrada y dos opciones mediante botones, empecemos primero por el botón “**agregar nuevo patrón**”, da clic como se muestra en la siguiente figura:

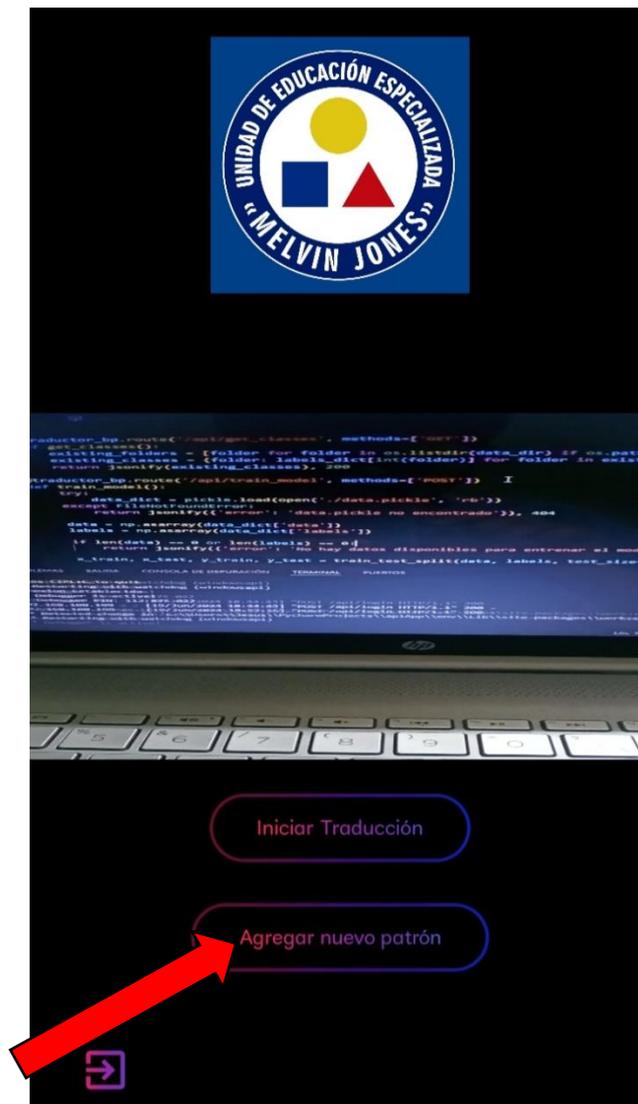


Fig. 53. Botón agregar nuevo patron

Paso 5: Luego de dar clic en el botón agregar nuevo patrón se abre una interfaz que con tres botones ordenados del 1 al 3, esta interfaz es la que se encargará de entrenar el modelo para que pueda detectar una señal y luego traducirlas, para ello deberá realizarse los pasos de acuerdo con el numero establecido de cada botón por ende daremos clic al botón 1 que es “**Iniciar captura de imágenes**”:

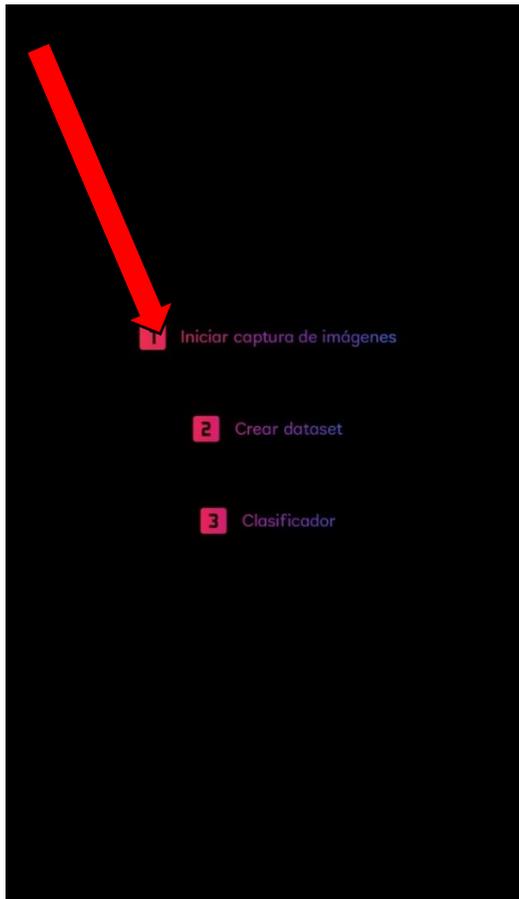


Fig. 55. Iniciar captura de imágenes

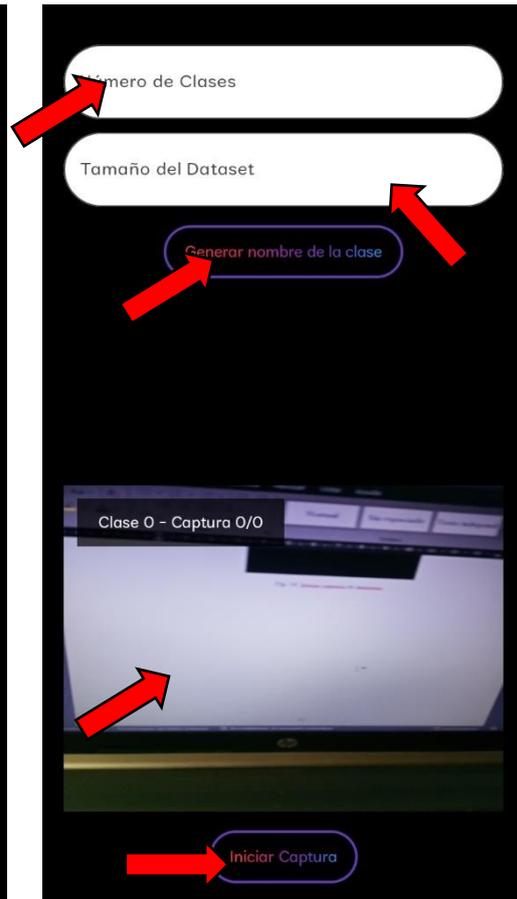


Fig. 54. Iniciar captura de imágenes

Luego de ello aparecerá una interfaz como se muestra en la figura Fig. 54 que contiene dos entradas de textos denominados: Numero de clase y Tamaño del Dataset, un botón para generar nombre para la clase, una cámara integrada para capturar las imágenes y un botón en la parte inferior para iniciar captura.

Número de Clases: Solo valores positivos de 1-5

Tamaño de Dataset: Valores positivos de 10 – 1000.

Se recomienda que para el tamaño del dataset se use un total de 500 muestras por patrón esto para mejorar la precisión.

Luego de ingresar los valores correctos en los campos se deberá dar clic en el botón generar nombre de la clase:

Recordatorio 1: Los valores ingresados no pueden ser iguales a valores ya guardados anteriormente en el servidor, en caso de ingresar un valor repetido no se

podrá iniciar con la captura de datos y emitirá un mensaje diciendo que ya existe una clase con ese nombre como se muestra en la figura Fig. 56.

Recordatorio 2: Si no se ingresa ningún valor en los campos y se trata de generar el nombre para las clases o iniciar captura la aplicación emitirá una alerta para que complete los campos correspondientes como se muestra en la figura Fig. 57.

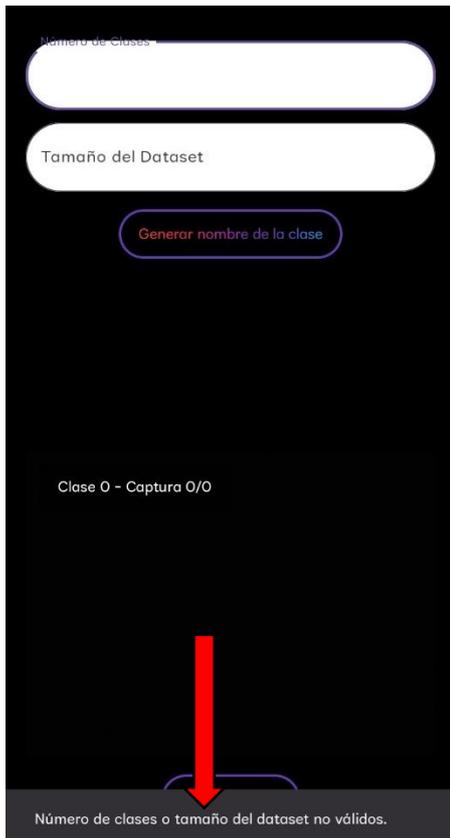


Fig. 57. Iniciar captura sin ingresar datos en los campos

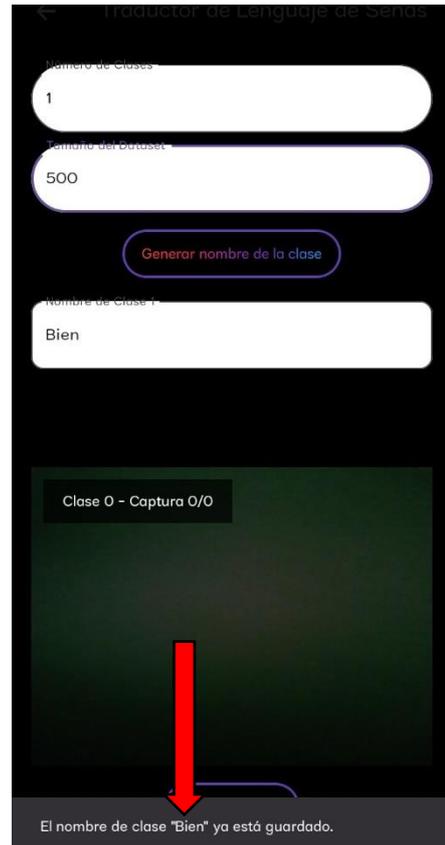


Fig. 56. Iniciar captura con nombre de seña repetida

Paso 6: Iniciar el proceso de captura de forma correcta para ello se deberá ingresar en los campos valores permitidos, al generar el nombre de la clase y guardarlo como en la figura Fig. 60 se procederá a realizar la captura de datos, como se muestra en la figura Fig. 59.

Recomendación 3: Se recomienda que cuando se realice la captura de señas se intercambie la mano llegando a la mitad de las capturas tomadas, es decir:

Si el tamaño del dataset es de 500

250 —————> con la mano izquierda como muestra la figura Fig. 61.

250 —————> con la mano derecha como muestra la figura Fig. 59.

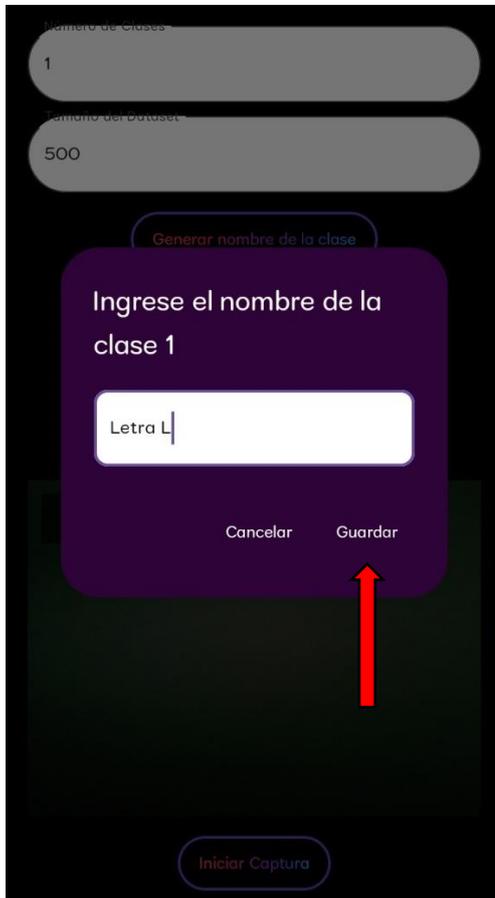


Fig. 60. Ingreso de nombre de clase con datos correctos

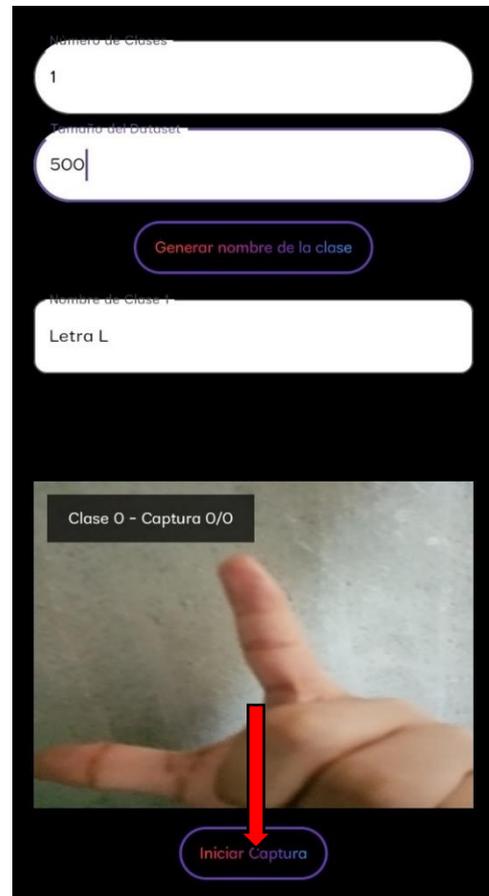


Fig. 58. Iniciar captura con datos correctos

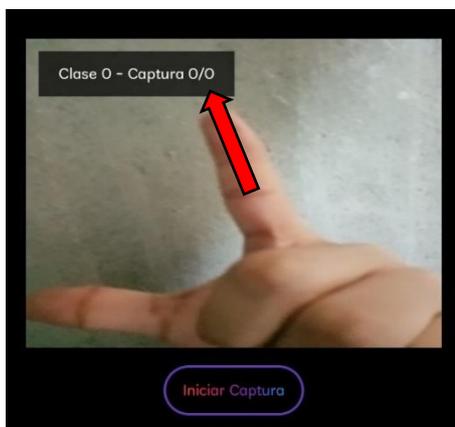


Fig. 61. Realizar capturas con mano izquierda

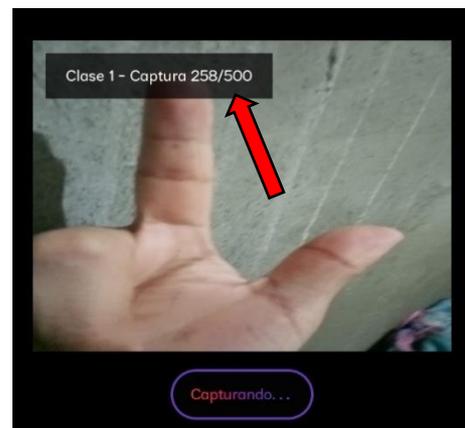


Fig. 59. Realizar captura con mano derecha

Paso 7: Regresar a la interfaz donde se encuentran las opciones del entrenamiento del modelo y oprimirá la segunda opción “**2 Crear dataset**” figura Fig. 63.

Y luego de dar clic se deberá esperar un momento hasta que la aplicación emita un mensaje que diga “**Dataset Actualizado**” figura Fig. 62.

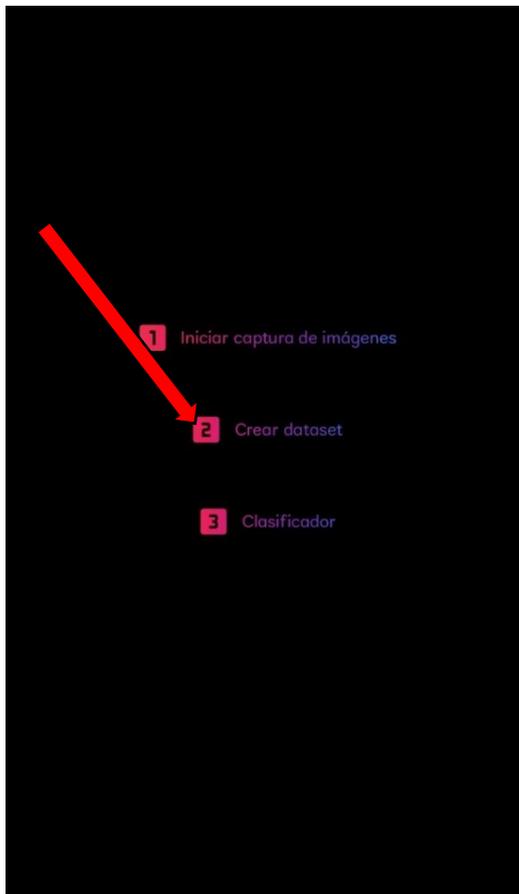


Fig. 63. Opción 2 crear dataset



Fig. 62. Mensaje dataset actualizado

Paso 8:

Luego de haber actualizado el dataset mediante el botón 2, se oprimirá el botón “**3 Clasificador**” Fig. 65

Se esperará por unos instantes hasta que la aplicación emita un mensaje que diga entrenamiento del modelo “**% de las muestras fueron clasificadas correctamente!**” Fig. 64.

Recomendación 4: Es importante que luego de cada registro de nuevo patrón

se realice el pase 2 y 3 de la interfaz de administrador donde se entrena el modelo este tiene el propósito de actualizar los nombres existentes de cada clase.

Nota: Si los datos no son actualizados al ingresar un nuevo patrón con un nombre repetido será posible y dañaría el registro de datos.

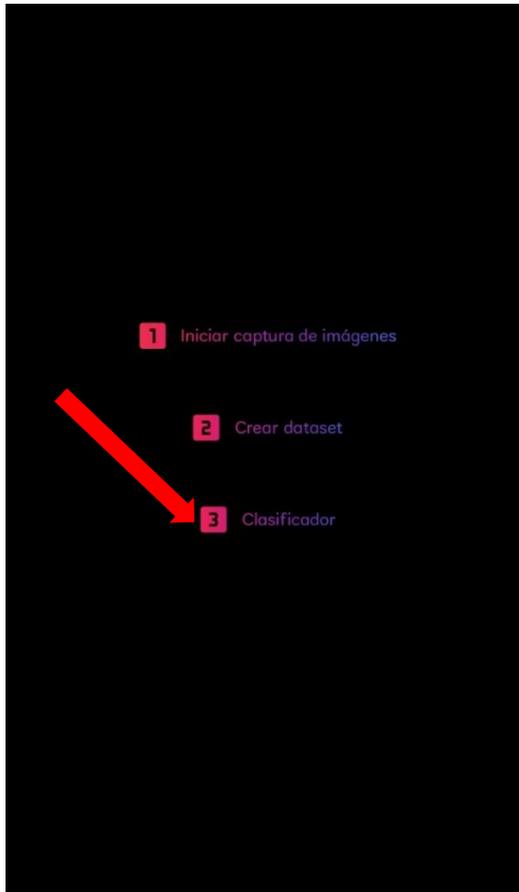


Fig. 65. Opcion 3 clasificador



Fig. 64. Muestras clasificadas

Traducción

Paso 9:

Y como parte final se encuentra el módulo de traducción, en este se regresa a la parte inicial de la interfaz y se selecciona la opción “**Iniciar traducción**” Fig. 68 este botón tiene el objetivo de comprobar si el patrón ingresado fue registrado de manera correcta.

Esto hará que la cámara integrada al poner la mano frente a ella pueda detectarla y devolver un frame con la seña traducida mostrando el nombre y los 21 puntos de

referencia que posee una mano, mano izquierda color rojo Fig. 66 mano derecha color verde Fig. 67

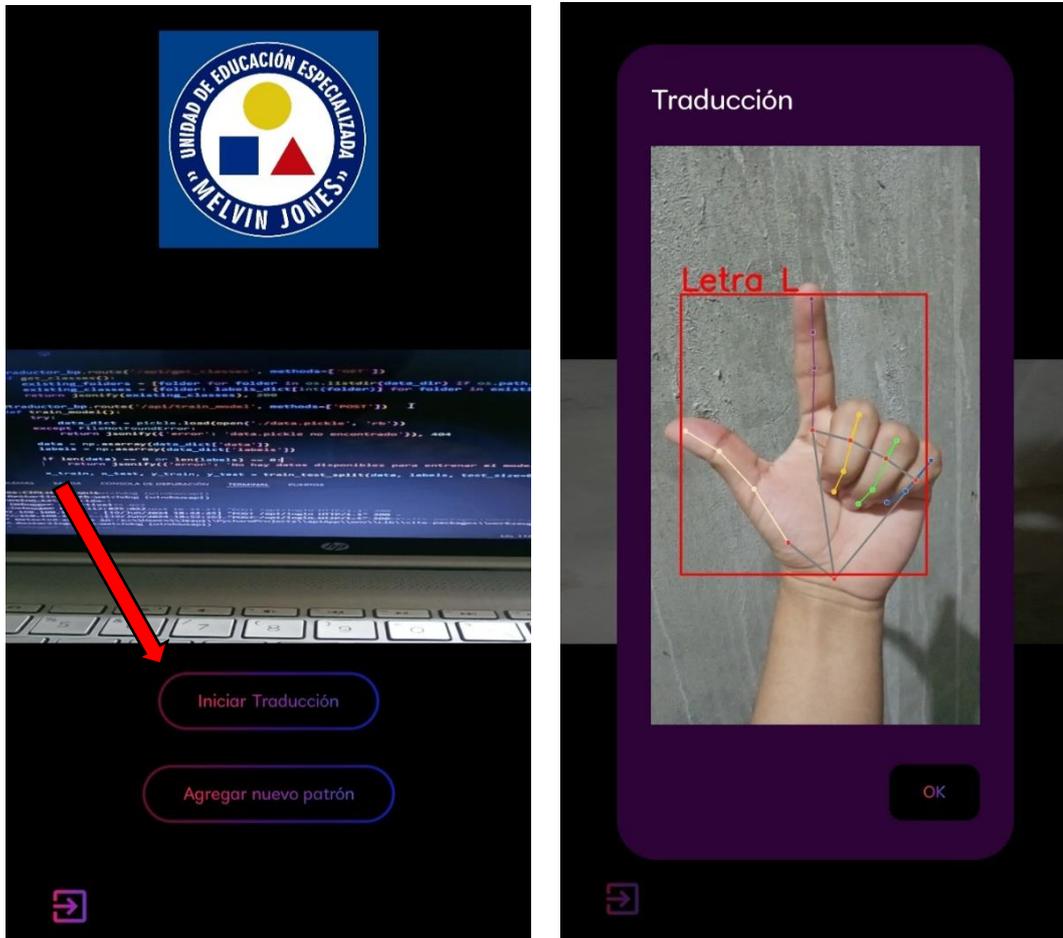


Fig. 67. Oprimir el botón iniciar traducción en Fig. 66. Traducción de seña con mano izquierda administrador

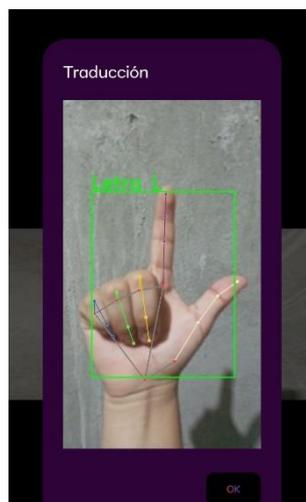


Fig. 68. Traducción de seña con mano derecha

Anexo 4. Manual de usuario para usuario no administrador

Aplicación móvil en Android para detectar y traducir señas estáticas de una mano

El siguiente manual describe los pasos para el uso correcto de la aplicación.

1. Objetivo.

Este manual tiene como propósito el guiar de manera correcta cada uno de los pasos y funcionalidades que puede emplear un usuario no administrador.

2. Uso de la aplicación móvil.

Inicio de sesión fallido

Paso 1: El usuario no administrador deberá ingresar las credenciales en los campos de entrada de texto que se muestra en la figura:

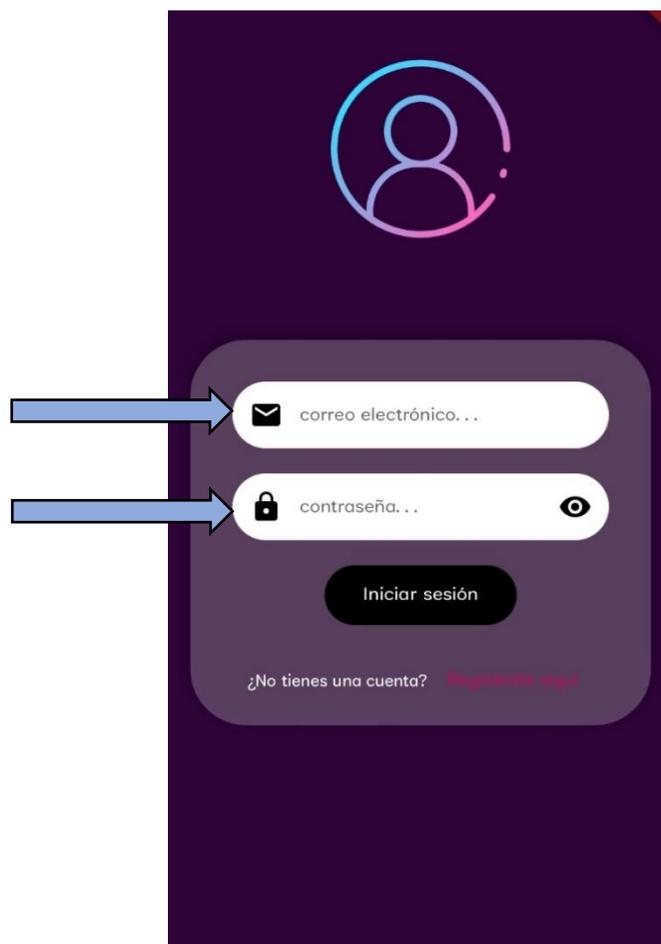


Fig. 69. Inicio de sesión - Usuario no administrador

Paso 2: El usuario deberá ingresar las siguientes credenciales para ingresar la interfaz de administrador, por ejemplo:

Credenciales de Usuario:

Correo: ejemplo1@gmail.com

Contraseña: ejemplo123

El proceso debería ser el siguiente como se muestra a continuación en la figura Fig. 71.

Con el símbolo del “ojito” en la parte derecha del cajón para ingresar la contraseña se puede visualizar la contraseña dándole clic como se muestra en la figura Fig. 70, de igual forma si se vuelve a dar clic sobre el símbolo se esconde la contraseña nuevamente.



Fig. 71. Inicio de sesión con credenciales de usuarios erróneos

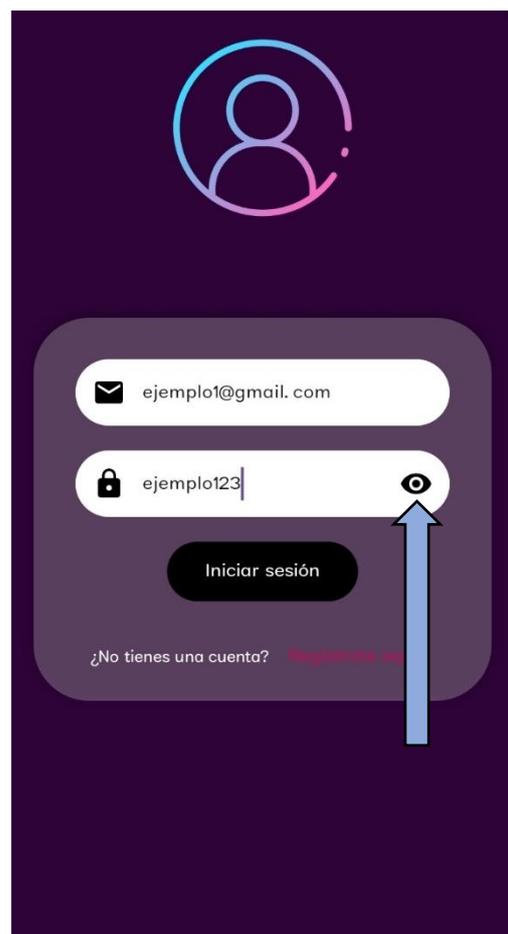


Fig. 70. Ver contraseña usuario no admin

En el caso de que las credenciales ingresadas sean incorrectas mostrara el mensaje “correo o contraseña incorrectos” como se muestra a continuación.



Fig. 72. Error de inicio de sesión

Registro de un nuevo usuario

Paso 3: de la interfaz de login da clic en donde dice “**Regístrate aquí**” eso lo llevará a la interfaz de registro de usuarios. Fig. 74

Paso 4: Llenar los campos que se requieren para iniciar el registro de usuarios y dar clic en el botón “**registrarse**” Fig. 73 eso lo redirigirá a la interfaz del login y mostrara un mensaje que dice: “**registro exitoso**” Fig. 75

en caso de que no se llenen ningún campo o los campos queden incompletos no se podrá iniciar sesión y marcara las casillas de color rojo Fig76. esto hasta que cada uno de los campos se llenen y se inicie sesión.

A continuación, las figuras que muestran el procedimiento:



Fig. 74. Navegar a la interfaz de registro de usuarios



Fig. 73. Oprimir el botón registrarse

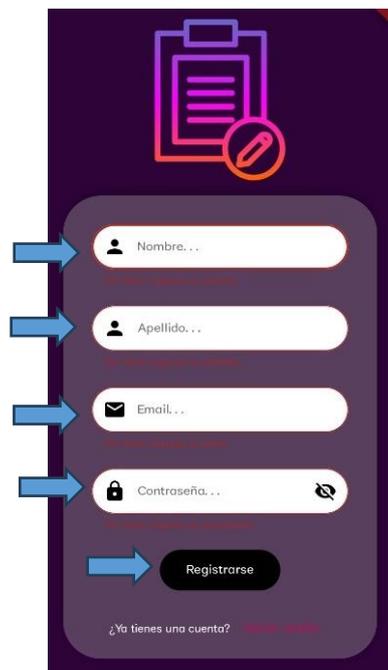


Fig. 76. Registrar usuarios con campos vacíos



Fig. 75. Registrar usuario

Inicio sesión exitoso

Paso 5: Iniciar sesión con las credenciales de usuario que fueron registradas anteriormente.

Credenciales de Usuario:

Correo: ejemplo1@gmail.com

Contraseña: ejemplo123

El proceso debería ser el siguiente como se muestra a continuación en la figura Fig. 78.

Luego de ello le aparecerá un mensaje que dice “login exitoso” Fig. 77 y se abrirá la interfaz del usuario con dos funciones, cada función es realizada mediante un botón.

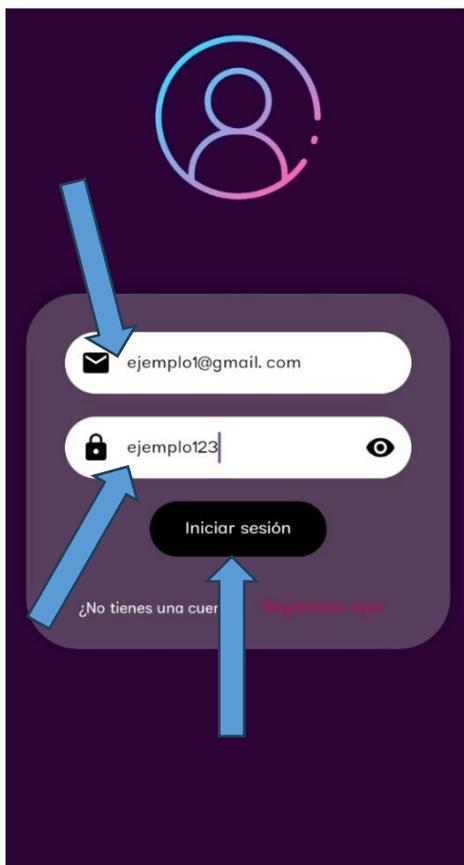


Fig. 78. Inicio de sesión exitoso - usuario



Fig. 77. Login exitoso - interfaz de usuario

Ver patrones guardado

Paso 6: De las dos opciones que existen en la interfaz de usuario escogeremos la opción que dice “**ver patrones guardados**” ubicado en el segundo botón Fig. 79.

Al dar clic nos abre la interfaz donde podemos ver que existen el nombre de una clase y un botón más abajo llamado “**Obtener imagen aleatoria**”



Fig. 79. Botón ver patrones guardados



Fig. 80. Interfaz para ver los patrones existentes

Paso 7

Si presionamos el nombre de la seña podemos ver que nos muestra un menú desplegable donde están todos los nombres de las señas guardadas en el servidor Fig. 82.

Si presionamos el botón “**Obtener imagen aleatoria**” nos muestra una imagen aleatoria del nombre de la seña seleccionada Fig. 81, este tiene el objetivo de informar al usuario como se realiza la seña de los nombres guardados, con la

finalidad de que el usuario tenga en conocimiento como hacer la seña de un nombre determinado.

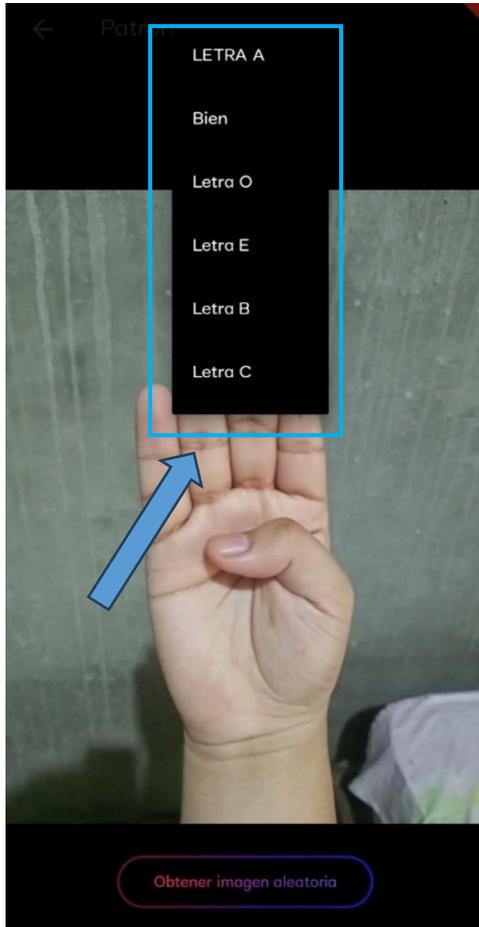


Fig. 82. Menú desplegable con señas guardadas en el servidor



Fig. 81. Obtener imagen aleatoria del patrón elegido

Traducción

Paso 8: Para realizar la traducción se regresa a la interfaz de inicio del usuario y se selecciona la opción del botón “Traducir” Fig.83 este llevará a la interfaz de traducción donde se encuentra la cámara lista para ejecutar el proceso Fig. 84 de detectar señas.

Esto hará que la cámara integrada al poner la mano frente a ella pueda detectarla y devolver un frame con la seña traducida mostrando el nombre y los 21 puntos de referencia que posee una mano, mano izquierda color rojo Fig. 86 mano derecha color verde Fig. 85.



Fig. 83. Botón traducir de interfaz de usuario

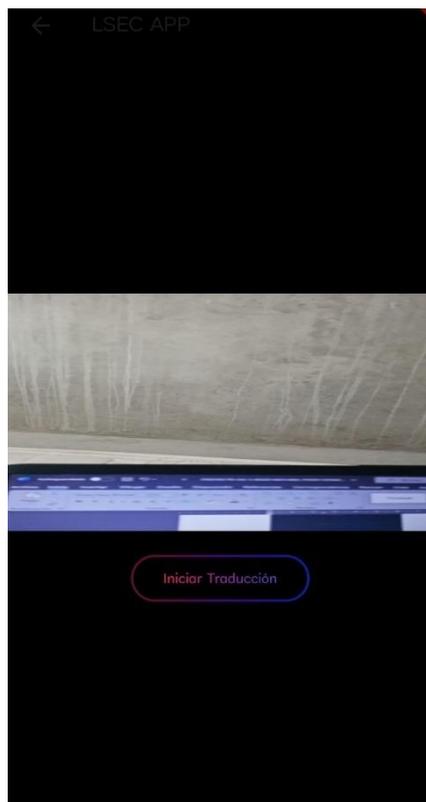


Fig. 84. Botón que inicia la traducción de señas

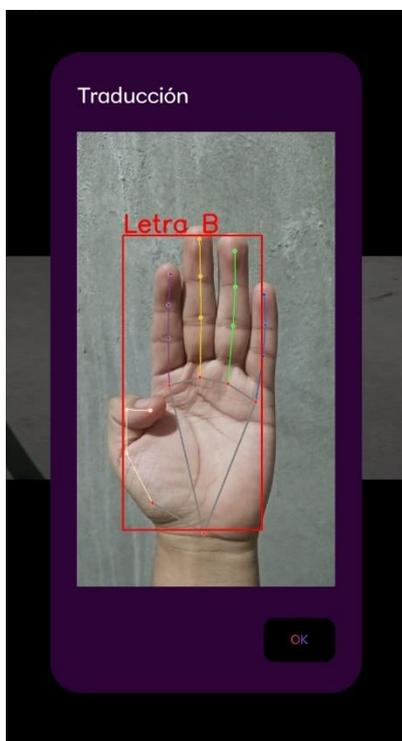


Fig. 86. Traducción de seña con mano izquierda interfaz de usuario

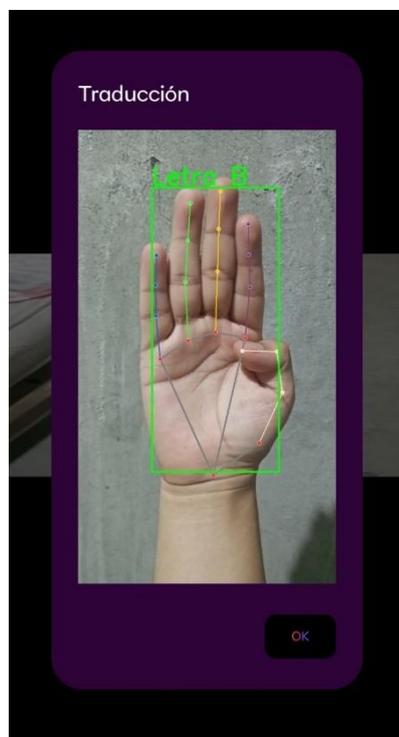


Fig. 85. Traducción de seña con mano derecha de interfaz de usuario

Anexo 5. Descarga e instalación

La aplicación móvil estará alojada en un drive con acceso público para poder ser instalada por los usuarios.

Pasos para la descarga e instalación.

Paso 1: Abrir el siguiente enlace desde un dispositivo móvil:

https://drive.google.com/file/d/1UMjdPWqIA9w-zR4F5SZoxMQn_rfh6Onk/view?usp=sharing

Paso 2: Luego de abrir el enlace le aparecerá la opción para descargar la aplicación como se muestra a continuación.

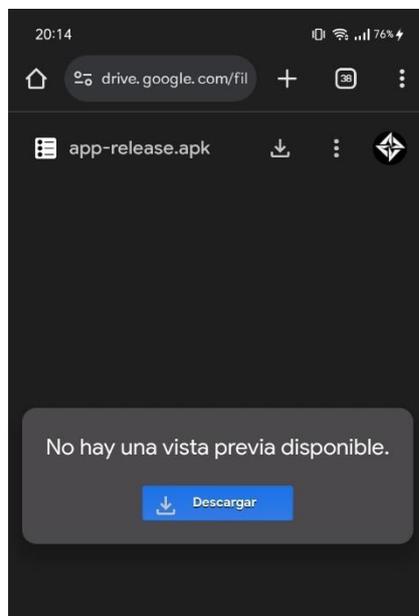


Fig. 87. Descargar

Paso 3: Seleccionar “Descargar de todos modos”



Fig. 88. Seleccionar descargar

Paso 4: Confirmar descargar de todos

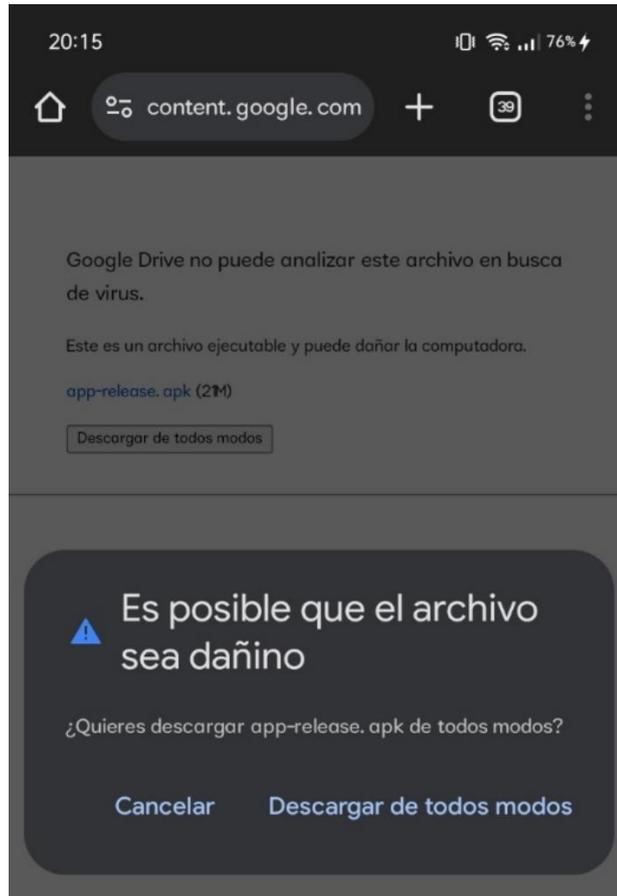


Fig. 89. Descargar de todos modos

Paso 5: Dar clic en instalar

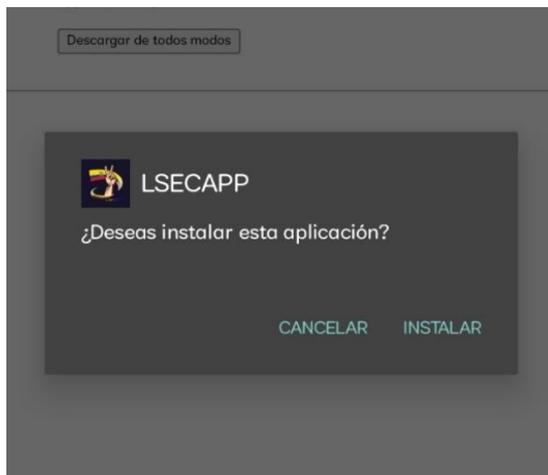


Fig. 91. Instalar

Paso 6: Dar clic en abrir

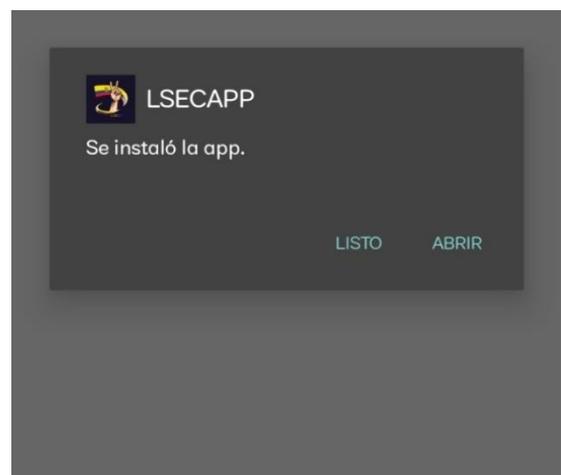


Fig. 90. Abrir

Paso 7: Dar permiso a la cámara del dispositivo

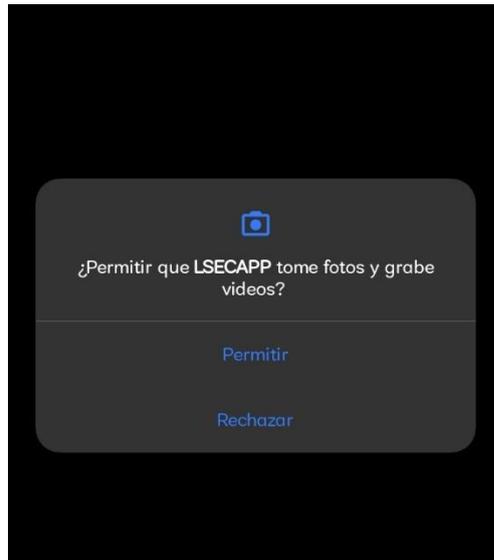


Fig. 92. Permisos de cámara