UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE MECATRÓNICA

DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO PARA FRASES BÁSICAS DEL LENGUAJE DE SEÑAS ECUATORIANO UTILIZANDO VISIÓN POR COMPUTADORA

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTORES: Dayanna Paola Cevallos Samaniego
Andrea Mariana León Correa
TUTOR: Ing. Jonathan Salvador Paillacho Corredores M.Sc.

Guayaquil - Ecuador
2026

19/02/26

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Dayanna Paola Cevallos Samaniego** con documento de identificación Nº **0953676194** y **Andrea Mariana León Correa** con documento de identificación Nº **0940195159**; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 19 de febrero del año 2026

Atentamente,

Dayanna Paola Cevallos Samaniego
0953676194

Andrea Mariana León Correa
0940195159

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, **Dayanna Paola Cevallos Samaniego** con documento de identificación Nº **0953676194** y **Andrea Mariana León Correa** con documento de identificación Nº **0940195159**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del **Dispositivo Tecnológico: DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO PARA FRASES BÁSICAS DEL LENGUAJE DE SEÑAS ECUATORIANO UTILIZANDO VISIÓN POR COMPUTADORA**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero del año 2026

Atentamente,

Dayanna Paola Cevallos Samaniego
0953676194

Andrea Mariana León Correa
0940195159

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Jonathan Salvador Paillacho Corredores**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN SISTEMA DE RECONOCIMIENTO PARA FRASES BÁSICAS DEL LENGUAJE DE SEÑAS ECUATORIANO UTILIZANDO VISIÓN POR COMPUTADORA**, realizado por **Dayanna Paola Cevallos Samaniego** con documento de identificación N° **0953676194** y por **Andrea Mariana León Correa** con documento de identificación N° **0940195159**, obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero del año 2026

Atentamente,

Ing. Jonathan Salvador Paillacho Corredores
1718907874

DEDICATORIA

Primordialmente, dedico este trabajo a Dios, por ser la luz en mi camino, por bendecirme con salud y por brindarme la fortaleza necesaria para culminar esta etapa de mi vida con éxito.

A mis padres, Ángel Cevallos y Judith Samaniego, a quienes dedico este logro con todo mi corazón. Todo lo que he logrado hoy nace de su esfuerzo diario y de ese amor sin condiciones que me ha servido de guía desde el primer día, brindándome lo mejor incluso a costa de su propio bienestar.

A mi hermana, por ser un apoyo constante y una fuente de motivación a lo largo de este proceso, acompañándome en cada desafío y celebrando cada avance.

Finalmente, este trabajo es para mi familia. A los que están aquí celebrando conmigo y a los que, aunque ya no están físicamente, me siguen acompañando desde el corazón. Especialmente a mi tío Mario, a mis abuelos maternos y a mi abuela paterna; su ausencia duele, pero sus enseñanzas y su ejemplo son los que me han traído hasta aquí. Sé que, donde quiera que estén, su orgullo sigue latente en cada uno de mis logros.

Dayanna Paola Cevallos Samaniego

Este trabajo de titulación está dedicado primordialmente a Dios, por haberme dado la vida y la fortaleza necesaria para superar cada obstáculo en este camino universitario.

A mi padre, Francisco León, por ser el ejemplo más puro de disciplina, responsabilidad y ética que he conocido. Le dedico este logro por cada enseñanza. Gracias, papá, por enseñarme que el amor también es sacrificio y trabajo duro; es gracias a todo Su esfuerzo constante que hoy puedo culminar con éxito esta primera etapa de mi formación profesional. Este logro es tanto suyo como mio por apoyarme y ser fuente de mi inspiración.

A mi madre, Shirley Correa por su amor inmenso y su motivación constante. Su apoyo incondicional y sus palabras de aliento fueron el motor que me impulsó a no rendirme en los momentos de mayor cansancio. A ambos, les agradezco por haberme formado como una persona de bien y por ser el pilar fundamental en mi vida. Todo este esfuerzo es, ante todo, para ustedes los amo y les dedico este triunfo con todo mi corazón.

Andrea Mariana León Correa

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por ser mi guía y darme el don de la vida y la salud. Gracias por otorgarme la sabiduría necesaria para entender cada desafío y la fortaleza espiritual para superar con éxito esta etapa de mi formación.

Agradezco a mis padres, Ángel Cevallos y Judith Samaniego, no existen palabras suficientes para expresar mi gratitud por su apoyo incondicional y por cada sacrificio que hicieron para que yo pudiera tener un futuro lleno de oportunidades. Gracias por ser mi refugio y mi guía; este logro es tan suyo como mío. Les agradezco infinitamente por haberme enseñando con el ejemplo los valores que me han convertido en la persona que soy hoy. Gracias por enseñarme que la verdadera perseverancia no es solo resistir, sino avanzar con fe; por mostrarme que la responsabilidad y la dedicación son las llaves que abren cualquier puerta, y sobre todo, por enseñarme a amar lo que hago. Cada palabra de aliento y cada gesto de amor han sido el combustible que me permitió no rendirme nunca.

A mi hermana, Ing. Adriana Cevallos, por ser mi compañera en este camino y por ser un ejemplo constante de superación para mi. Tu presencia en cada etapa de este proceso ha sido el impulso en los momentos de estrés. Gracias por ser la primera en celebrar cada pequeño logro; gracias por creer en mí.

Extiendo este agradecimiento a toda mi familia: tíos, primos y seres queridos que han estado presentes de forma constante, quienes conforman el pilar fundamental de mi existencia. Gracias por alentarme siempre a perseguir mis sueños y por brindarme oportunidades que me hicieron crecer de manera personal y profesional. A cada uno de ustedes, que con una oración, un consejo o simplemente estando presentes en mi vida aportaron un grano de arena en mi formación académica y personal. Ustedes son mi motivación para seguir creciendo cada día.

A mi compañera de tesis Andrea León, gracias por tu paciencia, por el esfuerzo compartido, el trabajo incansable en conjunto y la perseverancia mutua que demostramos para llevar este proyecto a feliz término. Superar los obstáculos de este camino fue mucho más sencillo gracias a tu compromiso y a la confianza que depositamos la una en la otra; terminar este proyecto es un triunfo que comparto contigo.

A mis amigos, quienes fueron mucho más que compañeros de estudio, gracias por ser la fuente de inspiración para el nombre de nuestro sistema SMAD. Gracias por hacer de esta etapa universitaria una experiencia inolvidable, llena de momentos de alegría que equilibraron los días de estrés académico. Agradezco profundamente su apoyo en los momentos más duros de este camino, su presencia fue la motivación que necesitaba para seguir adelante. Fuimos, en todo momento, un impulso mutuo, y valoro infinitamente haber compartido esta travesía con personas tan excepcionales. Gracias por creer en este proyecto y por estar presentes desde el inicio hasta este cierre tan esperado.

Finalmente, agradezco a nuestro tutor de tesis Ing. Jonathan Paillacho por su guía constante. Gracias por su paciencia y por compartir sus conocimientos, los cuales fueron fundamentales para el desarrollo y perfeccionamiento de nuestro sistema SMAD.

Dayanna Paola Cevallos Samaniego

Agradezco a Dios por bendecirme con la vida y por darme la fortaleza necesaria para culminar este camino universitario con éxito.

A mis padres, Francisco León y Shirley Correa, mi gratitud eterna. Gracias por su amor incondicional, por sus consejos y por el esfuerzo diario que realizaron para que yo pudiera estudiar. Su apoyo ha sido el cimiento de mi formación; sin sus sacrificios y su confianza en mi capacidad, este título no sería una realidad. A ellos les agradezco por enseñarme que el estudio es la mejor herramienta para el futuro.

A mi compañera de tesis, Dayanna Cevallos. Gracias por ser mi apoyo constante en este proceso y por compartir conmigo la responsabilidad de este gran reto. Gracias por tu paciencia, por las horas interminables de trabajo compartido y por la dedicación que pusiste para que nuestro proyecto fuera un éxito. Formamos un equipo donde la confianza y el esfuerzo mutuo fueron la clave para no detenernos nunca, gracias por tu entrega incansable y por creer en este sueño desde el primer día. Este logro es el resultado de nuestro excelente trabajo en equipo y lo comparto contigo con mucha alegría.

A la Universidad Politécnica Salesiana, por ser el espacio donde crecí profesionalmente, y a los docentes por compartir sus conocimientos y valores.

De igual manera, agradezco a mis amistades, quienes hicieron de este camino una experiencia inolvidable. Gracias por los momentos compartidos y por la motivación mutua. Finalmente, al Ing. Jonathan Salvador Paillacho Corredores, nuestro tutor de tesis, por su paciencia, su guía técnica y por orientarnos con profesionalismo en cada etapa del desarrollo del sistema SMAD. Su apoyo fue fundamental para superar los retos técnicos que se presentaron.

Andrea Mariana León Correa

RESUMEN

El presente proyecto tiene como objetivo principal desarrollar un algoritmo asistido por visión artificial para realizar el reconocimiento y traducción de la Lengua de Señas Ecuatoriana (LSEC) a texto en tiempo real. Este trabajo surge de la necesidad de apoyar a las más de 10 millones de personas en Ecuador que viven con discapacidad auditiva y mutismo. Se tomó como base la experiencia observada en escuelas del país, donde docentes con mutismo y estudiantes con discapacidad auditiva carecen de herramientas para comunicarse, buscando optimizar la interacción y reducir las barreras de aprendizaje en las aulas.

Para la captura de los movimientos, el sistema utiliza la cámara web para extraer coordenadas tridimensionales de las manos, el rostro y la postura. Para el entrenamiento de la red neuronal se utilizó la arquitectura LSTM (Long Short-Term Memory) como herramienta principal; esta tecnología destaca en el reconocimiento de secuencias, ya que ayuda a procesar el movimiento de las señas a través del tiempo, permitiendo que el sistema identifique frases completas de forma fluida.

Como componentes principales se utilizaron una computadora con conexión a internet, una cámara HD y un Arduino Uno para el control de la interfaz. Para ejecutar el funcionamiento del código, el programa captura secuencias de 60 fotogramas por cada gesto. Una vez iniciado el algoritmo, si la cámara no identifica una extremidad por falta de luz o movimiento brusco, el sistema rellena esos espacios con arreglos de ceros automáticamente; esto evita que el programa se detenga por falta de información y asegura que el modelo reciba datos constantes para la traducción.

Para finalizar, la interfaz muestra en pantalla el texto de la seña reconocida de manera inmediata. Durante las pruebas, el sistema alcanzó una precisión del 96 % en entornos con luz artificial y un promedio general del 81.3 %, confirmando que es una solución técnica viable y económica para mejorar la comunicación de los usuarios en su vida diaria.

Palabras claves: Visión artificial, SMAD, Lengua de señas, Inclusión tecnológica, Aprendizaje automático.

ABSTRACT

The main objective of this project is to develop an algorithm assisted by computer vision to recognize and translate Ecuadorian Sign Language (LSEC) into text in real time. This work arises from the need to support the more than 10 million people in Ecuador who live with hearing or speech disabilities. It is based on experiences observed in schools across the country, where teachers with mutism and students with hearing disabilities lack tools to communicate, aiming to optimize interaction and reduce learning barriers in the classrooms.

For the capture of movements, the system uses the webcam to extract three-dimensional coordinates of the hands, face, and posture. For the training of the neural network, the LSTM (Long Short-Term Memory) architecture was used as the main tool; this technology excels in sequence recognition, as it helps process the movement of signs over time, allowing the system to identify complete sentences smoothly.

The main components used were a computer with an internet connection, an HD camera, and an Arduino Uno for interface control. To run the code, the program captures sequences of 60 frames per gesture. Once the algorithm is started, if the camera does not detect a limb due to lack of light or sudden movement, the system automatically fills those gaps with zero arrays; this prevents the program from stopping due to missing information and ensures that the model receives consistent data for translation.

Finally, the interface immediately displays on the screen the text of the recognized sign. During testing, the system achieved an accuracy of 96 % in environments with artificial light and an overall average of 81.3 %, confirming that it is a technically viable and cost-effective solution to improve users' communication in their daily lives.

Keywords: Artificial vision, SMAD, Sign language, Technological inclusion, Machine learning.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

I. JUSTIFICACIÓN

En Ecuador, muchas personas con discapacidad auditiva y/o mutismo enfrentan barreras comunicacionales que dificultan su participación plena en la vida diaria. La inclusión social es un principio fundamental que busca garantizar que todas las personas, independientemente de sus condiciones, puedan acceder a las mismas oportunidades proporcionando una herramienta tecnológica que permita a las personas con discapacidad auditiva y/o mutismo comunicarse de forma más fluida con su entorno, fomentando su autonomía y participación activa en la sociedad.

La integracón de las personas con discapacidad auditiva y/o mutismo a la población económicamente activa es importante para la estabilidad psicológica y emocional del individuo así como para el desarrollo del país. El acceso al empleo y a la formación profesional es uno de los mayores desafíos para las personas sordas-mudas, en parte debido a la falta de intérpretes o medios tecnológicos que faciliten la comunicación. La traducción automática del Lenguaje de Señas Ecuatoriano (LSEC) en tiempo real permitiría la comunicación entre personas con discapacidad auditiva y/omutismo y el publico en general. Esto no solo beneficia a la persona en lo individual, sino que genera impactos positivos a nivel familiar, comunitario y nacional, ya que contribuye a reducir la dependencia económica, fomenta el desarrollo profesional y promueve una sociedad más justa e inclusiva.

En la actualidad, diversas investigaciones confirman que la inteligencia artificial y las redes profundas son las herramientas más eficaces para traducir la lengua de señas, gracias a su rapidez y capacidad de aprendizaje. Por ejemplo, en algunos estudios se destaca cómo el deep learning ofrece la flexibilidad necesaria para entender gestos complejos [1], mientras que otros trabajos demuestran que el reconocimiento en tiempo real es totalmente posible, lo cual resulta vital para que la comunicación entre personas sea fluida y natural. Asimismo, se han analizado las arquitecturas de red más eficientes para que estos sistemas no sean pesados y puedan correr en distintos dispositivos [2]. En el contexto de lenguajes regionales presentan avances relevantes aplicables al caso ecuatoriano, como el uso de técnicas de transfer of learning, generación de datos sintéticos y redes especializadas para mejorar el reconocimiento del lenguaje de señas en contextos con conjuntos de datos limitados. Estos enfoques permiten trabajar con gestos tanto estáticos como dinámicos y han demostrado ser efectivos en dispositivos de bajo costo, lo cual es fundamental para la implementación en Ecuador, donde se requiere accesibilidad tecnológica, adaptación a variaciones regionales del LSEC y soluciones que puedan aplicarse en entornos con recursos limitados. [3]. Existe evidencia de que el uso de sensores 3D permite reconocer palabras mediante el análisis de trayectorias del LSEC [4]. Sin embargo, su uso se limitá únicamente a 5 palabras y no considera frases.

Asimismo, se ha demostrado que modelos basados en visión por computadora, incluyendo arquitecturas de convolución 3D, permiten interpretar gestos dinámicos con alta precisión [5]. proponen fusionar datos visuales y de movimiento, aumentando la robustez del sistema. Otros enfoques como el uso de redes gráficas sobre esqueletos humanos también han mostrado gran potencial en el reconocimiento de señas. Además, se han explorado soluciones ligeras y eficientes para su uso en tiempo real, como el uso de frameworks como MediaPipe o el desarrollo de modelos optimizados para dispositivos móviles [6].

Este proyecto propone desarrollar un sistema de reconocimiento y clasificación de lenguaje de señas utilizando visión por computadora, con la finalidad de contribuir a la inclusión social de las personas con discapacidad auditiva y/o mutismo mejorando su accesibilidad y propiciando la equidad social. Este sistema será aplicable en entornos

educativos, laborales y de atención al público, y podrá escalarse para incluir más signos del lenguaje, abarcando palabras completas o frases. Considerando las realidades técnicas y económicas del Ecuador, el diseño del sistema busca ser accesible y replicable, con el potencial de implementarse en centros de atención ciudadana, instituciones educativas y espacios laborales. En un país con un creciente interés por la transformación digital y la inclusión social, este tipo de soluciones pueden convertirse en referentes tecnológicos a nivel nacional, ayudando a cerrar brechas de desigualdad y promoviendo el cumplimiento de los derechos de las personas con discapacidad.

II. Problema

A nivel mundial existen 10 millones de personas con discapacidad auditiva, la comunicación es un pilar fundamental para la interacción social y el desarrollo integral de las personas. Sin embargo, para la comunidad con discapacidad auditiva y mutismo, la falta de conocimiento generalizado del Lenguaje de Señas Ecuatoriano (LSEC) por parte de la población oyente crea profundas barreras de comunicación que impactan significativamente su inclusión social en diversos ámbitos [7].Esta situación no solo dificulta el acceso a la educación o al empleo, sino que termina por aislar a las personas en su propia comunidad. Al no contar con herramientas de comunicación, se crea un ciclo donde el usuario pierde autonomía y ve limitados sus derechos básicos. En el día a día, esta falta de entendimiento mutuo se traduce en una exclusión social invisible, que impide que las personas con discapacidad participen en igualdad de condiciones en las actividades más cotidiana [8]. El uso de herramientas tecnológicas surge como una respuesta necesaria para acortar esta distancia. No se trata solo de usar software avanzado, sino de aprovechar la visión artificial para devolverle a los usuarios la capacidad de ser escuchados y comprendidos en su propio entorno [9]. La lengua de señas al igual que el lenguaje oral evoluciona y varía en cada región lo que dificulta su generalización y requiere soluciones a medida. A pesar de los avances tecnológicos en el reconocimiento del lenguaje de señas a nivel global, existe una notable carencia de soluciones adaptadas a las especificidades del LSEC y a las realidades socioeconómicas de Ecuador.

A pesar de que existen tecnologías avanzadas en otros países, la mayoría resultan inviables para la realidad ecuatoriana. El alto costo de los equipos y las licencias de software las vuelve inalcanzables para la mayoría de las familias y escuelas. Además, muchos de estos programas no están diseñados para reconocer las señas propias de la región (LSE), lo que los convierte en herramientas ajenas a la cultura local. Por esta razón, se vuelve indispensable desarrollar soluciones que funcionen con hardware económico y que respeten la forma en que realmente se comunican los usuarios en el país [10]. Si bien la visión por computadora emerge como una solución innovadora y accesible para el reconocimiento y clasificación de gestos en tiempo real, su aplicación en el contexto ecuatoriano para el LSEC se encuentra en una etapa incipiente [11]. La implementación de sistemas basados en visión por computadora ofrece un potencial considerable para superar las barreras de comunicación, proporcionando una herramienta más intuitiva y eficiente en comparación con métodos tradicionales [12].

Uno de los mayores obstáculos es que apenas se está empezando a investigar el reconocimiento de señas dentro del territorio ecuatoriano. Al haber tan pocos desarrollos enfocados específicamente en la LSEC, se ha creado una distancia tecnológica muy grande con respecto a lo que se hace en el exterior. Esta escasez de proyectos locales no solo es un reto para los programadores, sino que es la razón principal por la cual las personas con discapacidad auditiva en el país aún no cuentan con sistemas que faciliten su integración [13]. Esta falta de desarrollo no solo restringe la creación de soluciones efectivas, sino que también dificulta la integración de estas tecnologías en dispositivos portátiles que sean prácticos, asequibles y eficientes para el uso diario de las personas con discapacidad [14]. La necesidad de soluciones tecnológicas accesibles y específicas para el LSEC es imperativa para garantizar que la comunidad con discapacidad auditiva y mutismo pueda participar activamente en la sociedad y ejercer sus derechos fundamentales a la comunicación y la inclusión [15].

III. OBJETIVOS

III-A. Objetivo general

Desarrollar un sistema de reconocimiento y clasificación de 5 frases básicas del lenguaje de señas ecuatoriano utilizando visión por computadora, favoreciendo la comunicación de personas con discapacidad auditiva y mutismo.

III-B. Objetivos específicos

- Implementar un sistema de adquisición de video en tiempo real para la captura de 5 frases representativas del LSEC para su posterior procesamiento.

- Desarrollar un algoritmo de inteligencia artificial basado en visión por computadora, para el reconocimiento y clasificación de frases representativas del LSEC a partir de gestos capturados por cámara.

- Validar la efectividad del sistema en pruebas de campo.

IV. Marco Teórico

IV-A. Antecedentes y Contexto Social

IV-A1. Historia del lenguaje de señas:

Las lenguas de señas constituyen sistemas lingüísticos naturales de carácter visual-gestual que permiten la comunicación compleja de ideas, emociones y conceptos sin recurrir al canal oral-auditivo. Lejos de ser simples gestos improvisados, estas lenguas presentan estructuras gramaticales propias, reglas sintácticas definidas y mecanismos semánticos que las sitúan al mismo nivel de complejidad funcional que las lenguas orales. Estudios lingüísticos contemporáneos han demostrado que las lenguas de señas poseen organización interna y capacidad expresiva plena, desmitificando la concepción histórica que las consideraba formas de comunicación rudimentarias o auxiliares [16].

Aunque en la actualidad su uso se asocia principalmente a la comunidad sorda, existen evidencias históricas que indican que la comunicación gestual ha estado presente desde épocas muy antiguas. Diversas comunidades indígenas de las Grandes Llanuras de América del Norte desarrollaron sistemas de señas que funcionaban como lenguas francas interétnicas, permitiendo la interacción entre pueblos con lenguas orales distintas mucho antes del contacto europeo. De manera similar, en la isla de Martha's Vineyard, en Estados Unidos, se consolidó una lengua de señas utilizada tanto por personas sordas como oyentes, integrándose de forma natural en la vida cotidiana de la comunidad hasta inicios del siglo XX. Estos casos evidencian que las lenguas de señas pueden surgir y consolidarse de manera espontánea cuando existen condiciones sociales favorables.

Las lenguas de señas no son universales, sino que se agrupan en familias lingüísticas con raíces históricas compartidas. Como se ilustra en la Figura 1, existen diversas vertientes predominantes, como la hispano-francesa y la británica, distribuidas geográficamente según los procesos culturales de cada región.

Figura 1: Principales familias de lengua de señas en el mundo.

Desde una perspectiva evolutiva, algunas teorías sostienen que el lenguaje humano pudo tener un origen gestual previo al desarrollo del lenguaje oral. No obstante, fue en Europa donde se produjo la primera formalización institucional del uso de las señas. Durante la Edad Media, los monjes claustrales empleaban sistemas gestuales para comunicarse durante los votos de silencio, lo que evidencia un uso sistemático del canal visual como medio

lingüístico.Uno de los hitos más antiguos en la sistematización de la enseñanza para personas sordas fue el método desarrollado por el monje Pedro Ponce de León en el siglo XVI [17].

En la Figura 2 se ilustra el alfabeto manual que diseñó, el cual sentó las bases para el uso del canal visual-gestual como un medio lingüístico formal, superando la concepción de que las señas eran simples gestos improvisados.

Figura 2: Ilustración del alfabeto manual de Pedro Ponce de León.

El siglo XVIII marcó un punto de inflexión histórico con la labor del abad Charles-Michel de l'Épée, quien en 1755 fundó en París la primera escuela pública para personas sordas, simbolizando el inicio de una transición hacia el reconocimiento formal de sus lenguas. A través de la recopilación y organización sistemática de las señas utilizadas por la comunidad parisina, l'Épée no solo legitimó el uso del lenguaje visual en la enseñanza, sino que integró un alfabeto manual y estructuras gramaticales que diferenciaron estos sistemas de los gestos espontáneos.

El retrato del Abad Charles-Michel de l'Épée, mostrado en la Figura 3, simboliza este periodo de transición, donde se comenzó a recopilar y organizar sistemáticamente la lengua utilizada por la comunidad sorda parisina.

Figura 3: Retrato del Abad Charles-Michel de l'Épée

Durante el siglo XIX, la influencia del modelo francés se extendió a otros países. En Estados Unidos, Thomas Hopkins Gallaudet y Laurent Clerc fundaron en 1817 la primera escuela para personas sordas en Hartford, Connecticut. A partir de esta experiencia educativa, el sistema de señas empleado por los estudiantes evolucionó hasta consolidarse como la American Sign Language (ASL), resultado de la combinación de señas francesas, aportes

locales y variantes desarrolladas en comunidades como Martha's Vineyard.
Este proceso evidenció la capacidad de las lenguas de señas para evolucionar, adaptarse y generar nuevas variantes con identidad lingüística propia.

Sin embargo, el reconocimiento institucional de las lenguas de señas enfrentó fuertes resistencias durante finales del siglo XIX y comienzos del XX. Corrientes educativas como el oralismo promovieron la exclusividad del lenguaje oral en la enseñanza de personas sordas, relegando el uso de las señas. Un evento emblemático de esta postura fue la Conferencia Internacional sobre Educación de Sordos celebrada en Milán en 1880, donde se favoreció el método oral y se desalentó el uso de las lenguas de señas en contextos educativos formales. A pesar de ello, las comunidades sordas continuaron transmitiendo y utilizando sus lenguas de manera activa, preservando su identidad lingüística y cultural.

La expansión de estos métodos educativos cruzó el Atlántico gracias a la colaboración entre figuras visionarias de distintos continentes. Como se detalla en la Figura 4, la labor conjunta de Thomas Hopkins Gallaudet y el educador sordo francés Laurent Clerc fue fundamental para establecer las bases de la educación formal en Norteamérica.

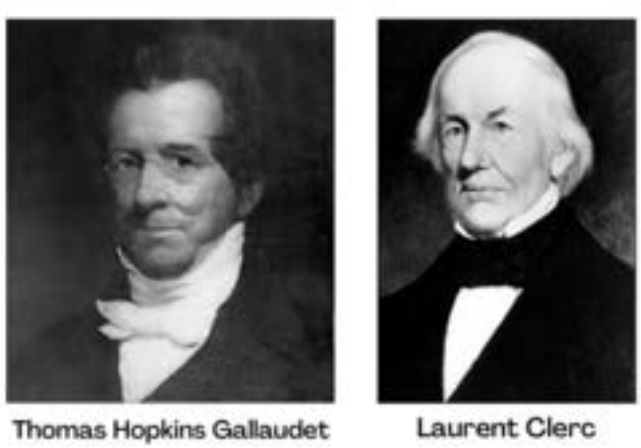


Figura 4: Fundadores de la primera escuela en EE. UU.

El verdadero respaldo académico para las lenguas de señas llegó en pleno siglo XX, impulsado principalmente por los hallazgos del lingüista William Stokoe. Hacia los años 60, Stokoe logró demostrar que la Lengua de Señas Americana no era un conjunto de gestos aislados, sino un sistema con estructuras fonológicas, sintácticas y gramaticales tan complejas como las de cualquier lengua hablada. Este descubrimiento fue un punto de inflexión en la lingüística moderna, pues dejó claro que estas son lenguas naturales completas y no simples imitaciones mímicas [18].

IV-A2. Naturaleza Lingüística, Autonomía y Variaciones Sintácticas: Entender la Lengua de Señas implica reconocerla como un eje de comunicación esencial que requiere mayor visibilidad y un compromiso social profundo para el desarrollo humano. Al igual que los idiomas orales, dominarla potencia las habilidades cognitivas y el pensamiento lógico. Según Ethnologue, hoy coexisten cerca de 130 lenguas de señas activas de un total de 6,909 lenguas vivas en el mundo [19].

Aunque varían en léxico y sintaxis, todas comparten una base visual y gestual propia de la comunidad con discapacidad auditiva. Cabe destacar su autonomía; por ejemplo, la lengua de señas utilizada en Estados Unidos (ASL) es distinta a la británica (BSL) y no posee una relación directa de derivación con el inglés. Su funcionamiento gramatical es basado en la combinación entre los parametros formacionales de Stokoe.

Como se detalla en la Figura 5, la comparación entre ASL y BSL deja claro que un mismo concepto puede representarse con distintas configuraciones manuales, lo que confirma que cada comunidad sorda construye su propio universo lingüístico.

Figura 5: Autonomía de las LSE (ASL vs. BSL).

IV-A3. Evolución y Contexto de la Lengua de Señas Ecuatoriana (LSEC): En el contexto local, la Lengua de Señas Ecuatoriana (LSEC) va mucho más allá de la gestualidad: es el pilar de identidad y pertenencia de la comunidad sorda en el país. A diferencia del español, que es lineal y auditivo, la LSEC utiliza el espacio tridimensional como un ”lienzo lingüístico”donde el movimiento y la ubicación de las manos generan significados complejos y autónomos. Este sistema facilita un intercambio de valores y saberes únicos de la cultura sorda [20],proceso que ha sido respaldado por organismos dedicados a la defensa de los derechos lingüísticos. En esta labor destaca la Federación Nacional de Personas Sordas del Ecuador (FENASEC), cuyo emblema Figura 6 representa el esfuerzo por consolidar la LSEC como el corazón de la cultura sorda en nuestro territorio.

Figura 6: Ilustración de la Federación Nacional de Personas Sordas del Ecuador.

Varias organizaciones promueven la defensa de los derechos lingüísticos y la inclusión en los diversos ámbitos. El movimiento asociativo de personas sordas en Ecuador cumple un rol decisivo. La creación de FENASEC permitió coordinar estos esfuerzos a nivel nacional y desarrollar investigaciones que fundamentaron la importancia de la LSEC como lengua natural de la comunidad sorda. Este trabajo colectivo derivó en la creación de materiales oficiales,

políticas públicas y un mayor reconocimiento social e institucional que ha fortalecido los derechos lingüísticos de esta población.

Históricamente, la LSEC ha sido un lenguaje vivo en constante transformación. Durante el siglo XX, recibió influencias de sistemas extranjeros como la lengua de señas chilena y la estadounidense (ASL), introducidas principalmente por misiones religiosas. Estudios sociolingüísticos indican que, aunque es una lengua aislada, aproximadamente el 30 % de su léxico proviene de la ASL y cerca del 20 % de la lengua de señas española, especialmente entre jóvenes usuarios [21]. Además, se reconoce la existencia de dos geolectos principales en el país, Sierra y Costa, cada uno con variaciones léxicas propias que enriquecen el patrimonio lingüístico nacional. Antes de su formalización, los usuarios dependían de "señas hogareñas" o gestos naturales, pero la organización en urbes como Quito y Guayaquil permitió la maduración del sistema, marcando un hito en 1982 con el "Proyecto Mano a Mano".

En la Figura 7 se presentan las configuraciones manuales específicas para cada grafema del alfabeto español ecuatoriano.

Figura 7: Ilustración del Abecedario de la Lengua de Señas Ecuatoriana.

En la historia reciente, el reconocimiento legal ha sido decisivo. En 2012, la FENASEC lanzó el Diccionario Oficial de la Lengua de Señas Ecuatoriana, contando con el respaldo gráfico de la academia y el aval del Ministerio de Educación. A pesar de contar con estas herramientas, la trayectoria y dirección de las señas siguen siendo parámetros fundamentales para transmitir significado; cambios en la dirección o en el tipo de movimiento lineal, circular, ascendente o descendente pueden alterar por completo el significado de una expresión [22]. Finalmente, es necesario mencionar que, aunque las primeras escuelas para sordos en el país surgieron bajo un modelo oralista que priorizaba el habla y la lectura de labios, la persistencia de la comunidad logró que la LSEC ganara el reconocimiento y el espacio que le corresponde en el ámbito educativo actual.

IV-A4. Brecha de comunicación y barreras de inclusión socio-laboral en el Ecuador: A pesar de los avances en el reconocimiento de la LSEC, la discriminación en el ámbito laboral persiste como una problemática crítica en el

país. Existe una brecha significativa entre el marco legal vigente y su aplicación práctica en el entorno empresarial; aunque la legislación ecuatoriana parece robusta, los mecanismos de cumplimiento y vigilancia resultan insuficientes para transformar la cultura organizacional. Esta discrepancia obliga a cuestionar la efectividad de las sanciones actuales y resalta la urgencia de desarrollar herramientas prácticas que aseguren que la igualdad de oportunidades trascienda el plano normativo para convertirse en una realidad palpable [23].

En la Figura 8 se destacan etapas críticas como el levantamiento del perfil inclusivo, el acompañamiento continuo y la adaptación de actividades, procesos estructurados que garantizan la igualdad de oportunidades.

Figura 8: Proceso para contratación de personas con discapacidad.

La exclusión de las personas con discapacidad del mercado laboral es un indicador preocupante de las barreras estructurales existentes en el Ecuador. Según datos del Instituto Nacional de Estadística y Censos (INEC), menos del 25 % de las personas con discapacidad en edad de trabajar cuentan con un empleo formal. Esta baja tasa de inserción es el resultado directo de prejuicios arraigados y una infraestructura empresarial que limita las oportunidades de este colectivo. Asimismo, la carencia de datos actualizados y herramientas de comunicación efectivas dificulta su integración plena. Superar estos obstáculos, mediante la implementación de tecnologías de asistencia y comunicación, es imperativo para garantizar que las políticas de inclusión generen un impacto real en la calidad de vida de los [24].

IV-A5. Parámetros Formacionales de Stokoe: Las lenguas de señas se estructuran como sistemas lingüísticos visuales y gestuales compuestos por unidades mínimas con significado propio. Para estudiarlas a fondo, se han establecido parámetros funcionales que permiten segmentar y diferenciar cada seña; estos actúan de forma análoga a los fonemas en el habla oral, ya que cualquier cambio en ellos altera por completo el sentido de lo que se comunica.

William Stokoe, en los años 60, fue el pionero en proponer un modelo formal que identificaba tres elementos clave: la forma de la mano, el punto de articulación y el movimiento realizado. Sin embargo, estudios posteriores han robustecido esta teoría, reconociendo actualmente cinco parámetros esenciales: configuración, orientación, ubicación, movimiento y los componentes no manuales [25].

- **Configuración de la mano (Handshape)** Se refiere a la postura específica que adoptan los dedos y la palma al ejecutar un signo. Como se observa en la Figura 9, este factor es determinante, pues una ligera variación en la posición de los dedos puede transformar radicalmente el significado semántico de la seña.

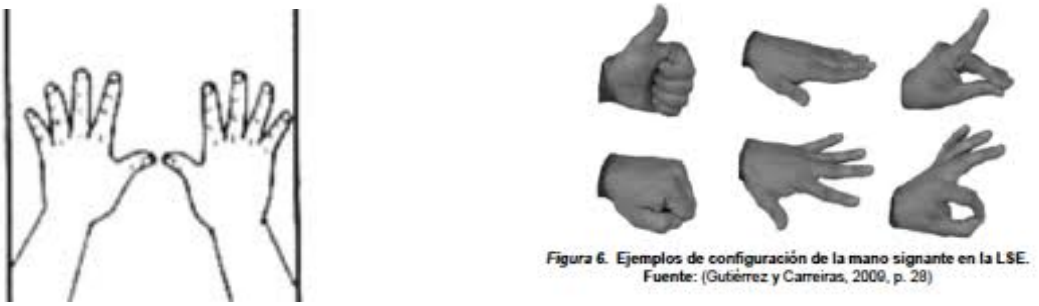


Figura 9: Ejemplo de configuración de la mano en la LSE

- **Orientación de la mano (Orientation)**
  Este parámetro indica hacia dónde apunta la palma o los dedos durante la seña. Funciona de manera independiente al movimiento, lo que permite distinguir entre señas que, aunque compartan la misma forma y lugar, difieren en su posición espacial Figura 10.

Figura 10: Ejemplo de orientación de la mano en la LSE

- **Lugar de articulación (Location)** Es el punto exacto del cuerpo o el entorno donde se realiza el gesto, ya sea el rostro, el torso o el espacio neutro frente al hablante. Su rol es gramatical y semántico, utilizando el cuerpo como marco de referencia para establecer relaciones de significado, tal como ilustra la Figura 11.

Figura 11: Ejemplo de lugar de articulación de la mano en la LSE

- **Movimiento (Movement)** Describe el dinamismo del gesto: su trayectoria, velocidad y dirección. En la comunicación visual, el movimiento es un rasgo distintivo fundamental; un cambio en el desplazamiento

puede producir conceptos totalmente diferentes Figura 12.

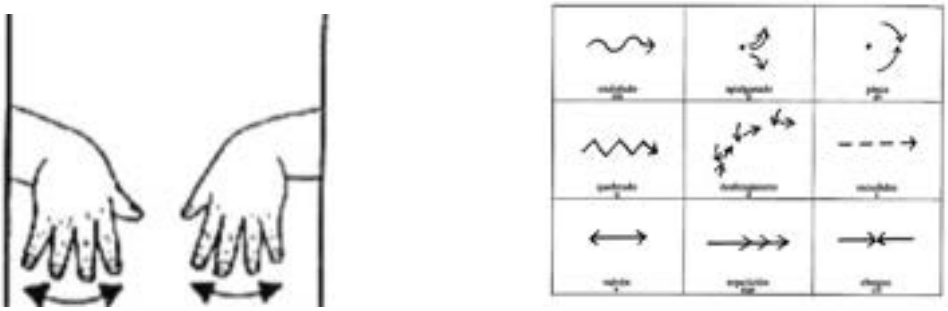


Figura 12: Ejemplo de movimineto de la mano en la LSE

- **Componentes no manuales (Non-manual markers)** Aquí se incluyen los gestos faciales, la postura del cuerpo y los movimientos de la cabeza. Estos elementos no son accesorios; aportan capas de información pragmática y emocional que son indispensables para interpretar el mensaje correctamente, como se muestra en la Figura 13.

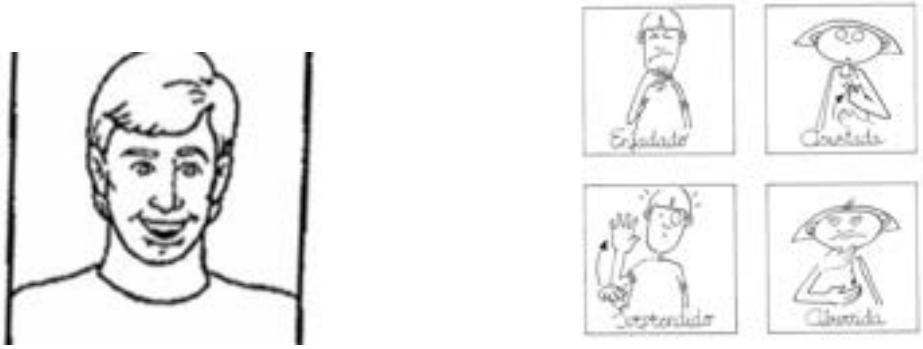


Figura 13: Ejemplo de componentes no manueales en la LSE

IV-B. Fundamentos de Visión Artificial

IV-B1. Redes neuronales: Las Redes Neuronales Artificiales (RNA) operan bajo una arquitectura que intenta imitar la interconexión de las neuronas en el cerebro humano. Se estructuran mediante nodos organizados en capas que procesan información en conjunto. La gran virtud de las RNA es su habilidad para resolver problemas complejos donde no se pueden aplicar reglas lógicas fijas, siendo piezas clave en el reconocimiento de patrones y la creación de modelos predictivos de alta precisión [26].

El origen de estos modelos se remonta a 1943, año en que Warren McCulloch y Walter Pitts presentaron la primera propuesta de una neurona computacional. Este prototipo inicial buscaba replicar el comportamiento binario de las neuronas biológicas (activación o inhibición). Ese fundamento matemático permitió cimentar la IA moderna, definiendo a la neurona como una unidad que procesa entradas de datos para generar una respuesta basada en una función de activación específica.

En la Figura 14 se detalla esta unidad fundamental. Aquí, H actúa como la función de activación (específicamente la función escalón de Heaviside) ligada a un umbral u. Las variables x_j representan los datos de entrada, mientras

que w_j define el peso o importancia de cada una. Si la suma de estas entradas supera el umbral, el sistema arroja un 1; de lo contrario, el resultado es 0. Este concepto fue el que permitió a Rosenblatt desarrollar más tarde el perceptrón, un pilar histórico en la evolución de las redes neuronales [27].

$$y = H\left(\sum_{j=1}^{n} w_j x_j - u\right)$$

Figura 14: Modelo computacional de McCulloch-Pitts.

En la actualidad, el modelo de una neurona artificial ha evolucionado hacia una estructura que incluye la suma ponderada de las entradas más un valor de ajuste denominado sesgo o bias.

En la Figura 15 se detalla como el nodo procesa la combinación lineal de las activaciones de entrada (a) y sus pesos (w) más un sesgo (bias), resultando en un valor intermedio (z) que es transformado por la función de activación (g) para generar la salida final.

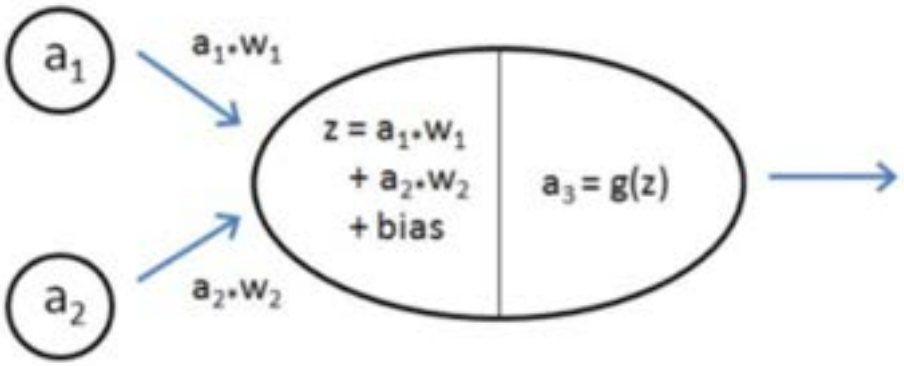


Figura 15: Esquema funcional de una neurona artificial.

Como se observa en la Figura 16., la importancia del sesgo o bias radica en su capacidad para desplazar la función de activación, permitiendo que la red neuronal tenga mayor flexibilidad para ajustarse a los datos de las señas durante el entrenamiento.

IV-B2. Organización Jerárquica de las Redes Neuronales: Estructuralmente, una RNA se define como un conjunto de unidades de procesamiento dispuestas en niveles, cuya arquitectura puede representarse mediante un grafo dirigido ponderado. En este modelo, los nodos corresponden a las neuronas y las aristas representan las conexiones sinápticas, donde cada enlace posee un peso que determina la intensidad de la señal transmitida entre

la salida de una neurona y la entrada de la siguiente.

De manera general, una red de tipo perceptrón se organiza en tres niveles fundamentales:

- **Capa de entrada:** Encargada de recibir los datos externos.
- **Capa oculta:** Donde se realiza el procesamiento y la extracción de patrones.
- **Capa de salida:** Que entrega el resultado final o la clasificación.

Es importante destacar que, cuando la arquitectura incluye dos o más capas ocultas, la red adquiere la denominación de Red Neuronal Profunda (Deep Neural Network). En la Figura 15 se muestra la estructura multicapa, que es la que permite al sistema aprender representaciones de datos mucho más complejas.

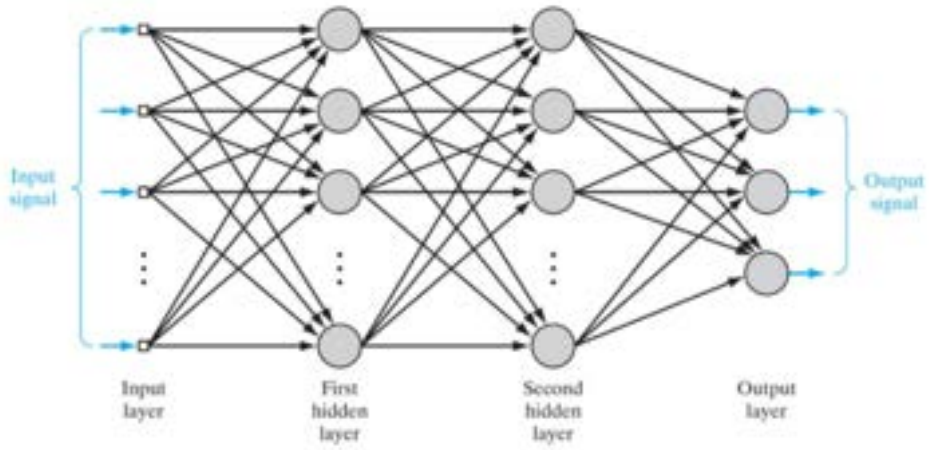


Figura 16: Arquitectura de una red neuronal profunda.

La teoría de grafos aplicada a las RNA, el carácter "ponderado" de las aristas es lo que permite que el algoritmo de retropropagación (backpropagation) ajuste los pesos durante el entrenamiento para minimizar el error de salida [28].

IV-B3. Conceptos Fundamentales de Visión por Computador: En la actualidad, el término Visión por Computador ha ganado una gran notoriedad en el ámbito tecnológico. No obstante, su definición suele traslaparse con otras disciplinas similares, lo que genera confusiones conceptuales. Con el objetivo de delimitar el alcance de esta investigación, se establecen a continuación las precisiones técnicas necesarias:

- Procesamiento Digital de Imágenes Esta disciplina se centra en la aplicación de algoritmos y operaciones matemáticas sobre una imagen de entrada para obtener, como resultado, una versión transformada de la misma. A diferencia de la visión artificial, el procesamiento de imágenes no busca "entender" lo que aparece en la escena, sino optimizar sus propiedades físicas como el brillo, el contraste o la nitidez o extraer características específicas mediante filtros para facilitar análisis posteriores [29].

Tal como se aprecia en la Figura 17, el proceso de segmentación es vital para que el algoritmo pueda separar nítidamente el objeto de estudio de su entorno, facilitando así una extracción de patrones mucho más limpia y precisa.

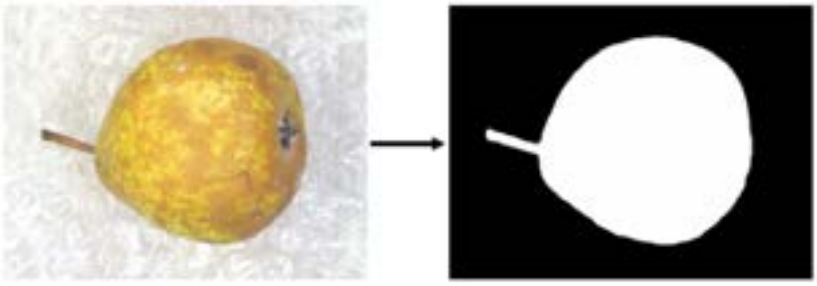


Figura 17: Ejemplo de procesamiento digital de imágenes aplicado a la segmentación.

IV-B4. Visión artificial aplicada al reconocimiento de gestos dinámicos: Cuando aplicamos visión artificial al reconocimiento de la Lengua de Señas Ecuatoriana (LSEC), el reto principal es lograr que el sistema procese y dé sentido a secuencias de video en movimiento, emulando la capacidad humana de interpretación. A diferencia de los métodos de programación tradicionales, que dependen de reglas rígidas y manuales, el Aprendizaje Automático (Machine Learning) adopta una postura inductiva. Gracias a este enfoque, el software puede aprender a identificar patrones directamente desde los datos de ejemplo. Esta capacidad es fundamental para abordar tareas complejas, como descifrar imágenes o señales que no pueden resolverse con una simple fórmula matemática.

En este escenario, el Aprendizaje Profundo (Deep Learning) ha marcado un antes y un después en el análisis de audio y video. Al utilizar redes neuronales de múltiples capas, es posible gestionar volúmenes masivos de datos y extraer rasgos distintivos de forma automatizada. Esta es una ventaja crucial frente a las técnicas clásicas de Machine Learning, donde el investigador debía invertir mucho tiempo seleccionando manualmente cada característica. Con el Deep Learning, es el propio modelo el que aprende de forma autónoma a representar la información de la mejor manera posible [30].

En términos generales, los gestos se pueden clasificar en:

- **Gestos estáticos:** Son aquellos que se definen por una postura corporal o posición de la mano específica en un momento exacto.
- **Gestos dinámicos:** Formados por una secuencia de movimientos de ciertas partes corporales a través del tiempo.

En la Figura 18 se esquematiza el proceso de captura y la distinción entre posturas fijas y secuencias de movimiento.

La captura y el reconocimiento de estos gestos dinámicos posibilitan su uso en áreas críticas como la traducción de vocabularios de lengua de señas, brindando un apoyo directo a personas con discapacidad auditiva.

IV-B5. Aplicación en el Reconocimiento de Gestos y Lengua de Señas: Hoy en día, el uso de arquitecturas profundas y redes neuronales convolucionales ha permitido alcanzar niveles de precisión en secuencias de video que antes eran impensables. El análisis no se limita a un solo enfoque, sino que se organiza en tres niveles interconectados:

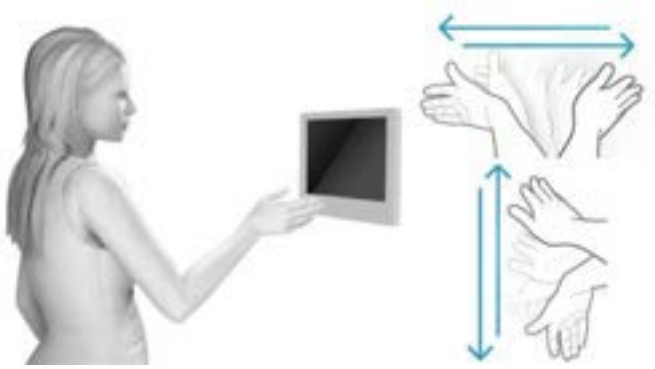


Figura 18: Ilustración Captura y reconocimiento.

- **Reconocimiento de Señas**: Se enfoca en convertir los movimientos del cuerpo en texto legible, respetando siempre las reglas gramaticales de la Lengua de Señas Ecuatoriana (LSEC).
- **Reconocimiento de Acciones**: Este nivel busca clasificar los movimientos generales que realiza una persona de manera global.
- **Reconocimiento de Gestos**: se centra en la detección de movimientos específicos que poseen un significado definido dentro de un contexto particular.

En la Figura 19, se detalla la arquitectura convolucional empleada, mostrando cómo las capas de convolución y reducción trabajan en conjunto para procesar la información visual antes de la clasificación final.

Figura 19: Arquitectura de una Red Neuronal Convolucional (CNN).

IV-B6. Etapas del reconocimiento de señas: Los gestos pueden clasificarse en gestos corporales, que involucran movimientos de todo el cuerpo; gestos manuales, como un saludo; gestos realizados con manos y dedos, característicos de la lengua de señas; y gestos faciales, como guiños o movimientos de los labios. Otra distinción importante es la que diferencia los gestos estáticos, también llamados poses, definidos por una configuración específica del cuerpo, de los gestos dinámicos, formados por una secuencia de movimientos de ciertas partes corporales.

El reconocimiento de la lengua de señas se clasifica como un problema complejo de reconocimiento de movimientos secuenciales. Para lograr que un sistema traduzca de forma automática los gestos de un intérprete, es necesario resolver múltiples retos técnicos que dependen, en gran medida, del tipo de sensor utilizado. En este contexto, los gestos pueden clasificarse según su ejecución en:

- **Gestos corporales:** Involucran movimientos de todo el cuerpo.

- **Gestos manuales:** Movimientos realizados con manos y dedos, esenciales en la lengua de señas.
- **Gestos faciales:** Como guiños o movimientos de los labios que aportan carga gramatical.

En la actualidad, la detección y el análisis de gestos dinámicos se han convertido en una herramienta clave en distintos ámbitos, como la telemedicina, el control de interfaces en videojuegos y la navegación en entornos virtuales. Sin embargo, una de sus aplicaciones más importantes es la traducción de las lenguas de señas. Este tipo de lenguaje representa un desafío significativo, ya que implica el reconocimiento de movimientos secuenciales complejos. Para que un sistema pueda traducir automáticamente los gestos realizados por un intérprete, es necesario superar diversos retos técnicos, los cuales dependen en gran medida del tipo de sensor empleado para la captura de los movimientos.

Lograr que un sistema reconozca la lengua de señas de forma automática es un desafío técnico considerable, principalmente porque se trata de interpretar movimientos secuenciales complejos. No basta con captar una imagen estática; el software debe "entender" la continuidad del gesto. El éxito de esta traducción depende en gran medida de la calidad del sensor y de la capacidad del algoritmo para resolver retos de oclusión o velocidad de movimiento.

En la Figura 20, se presenta el flujo completo de trabajo: desde que la cámara captura el video hasta que el sistema realiza la interpretación semántica. Para elevar la precisión del modelo, hemos integrado técnicas de maximización de expectativas y diccionarios de anotación, lo que permite refinar los resultados obtenidos.

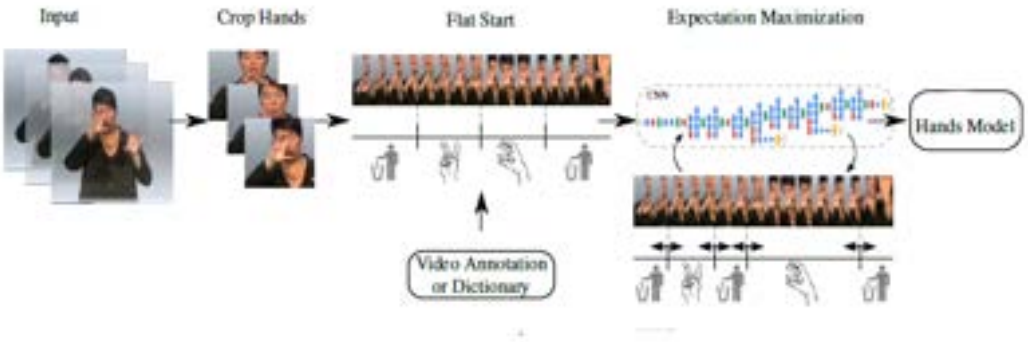


Figura 20: Algoritmo para procesamiento de movimientos secuenciales.

Este proceso de capturar y reaccionar ante un gesto involucra la integración de tres áreas fundamentales de la informática:

- **Procesamiento de imágenes**: En todo proceso de reconocimiento en video es fundamental contar con un manejo y filtrado adecuado de las imágenes. Esto puede implicar, entre otras cosas, eliminación de ruido, escalado o rotación de la imagen, uso de filtros frecuenciales para detectar patrones y filtros de color.
- **Procesamiento temporal:** Dado que un gesto se entiende como una secuencia de movimientos de una o varias partes del cuerpo, es necesario realizar un tratamiento adecuado de la información temporal.
- **Sistemas inteligentes:** Para clasificar un patrón de video y actuar en consecuencia, se requiere el uso de técnicas inteligentes (machine learning).

En la Figura 21, se esquematiza este flujo de trabajo integral, destacando la interdependencia entre el manejo de filtros visuales y la lógica de clasificación del sistema inteligente.

Figura 21: Etapas fundamentales del procesamiento computacional para el reconocimiento de patrones gestuales.

Las lenguas de señas pueden considerarse un caso particular dentro del reconocimiento de gestos dinámicos. Se trata de un problema multidisciplinario muy complejo, con numerosos aspectos que aún deben mejorarse. Aunque recientemente se han logrado avances mediante el reconocimiento de gestos, todavía falta mucho para desarrollar aplicaciones precisas y robustas que permitan traducir e interpretar con fiabilidad los signos producidos por un intérprete [31].

El proceso de reconocer una lengua de señas abarca múltiples etapas, que pueden resumirse de la siguiente manera:

- Seguimiento de las manos del intérprete
- Segmentación de las manos y modelado de su forma
- Reconocimiento de las formas de las manos
- Reconocimiento del signo como unidad sintáctica
- Asignación de significado a la secuencia de signos
- Traducción de la semántica de los signos a la lengua escrita

El bloque de proceso detallado en la Figura 22, muestra la ruta crítica que sigue la información, desde la captura inicial y el cálculo de características hasta la etapa final de reconocimiento del signo.

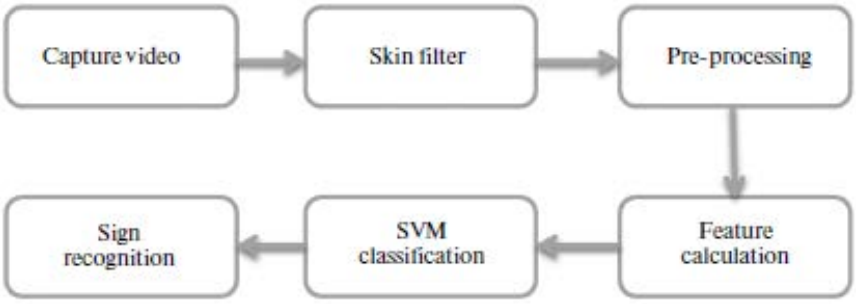


Figura 22: Bloque de proceso.

IV-B7. Sensores físicos vs. Visión Artificial: La captura de gestos dinámicos ha sido abordada históricamente mediante dos enfoques principales: el uso de sensores físicos acoplados al cuerpo y el uso de visión artificial a través de cámaras. La elección del sensor es un factor crítico, ya que determina no solo la precisión del sistema, sino también su accesibilidad y costo. Históricamente, se han utilizado dispositivos como guantes de datos (data gloves)

o sensores de profundidad como el Kinect para obtener información tridimensional. Sin embargo, la implementación en este proyecto se aleja de estas soluciones por las siguientes razones:

- **Accesibilidad económica:** El uso de sensores físicos especializados o guantes con sensores de flexión representa un costo elevado que limita la masificación de la tecnología en entornos educativos o sociales en Ecuador.
- **Comodidad del usuario:** El uso de guantes o sensores pegados al cuerpo suele ser estorboso y quita naturalidad a las señas. Estos equipos obligan al usuario a perder mucho tiempo configurándolos antes de cada charla, lo que vuelve la comunicación lenta e incómoda en situaciones reales.
- **Portabilidad:** Al usar visión artificial, el sistema puede correr en casi cualquier equipo que ya se tenga en casa, como una laptop o una tablet con cámara. Esto evita que el usuario deba comprar aparatos especiales, permitiendo que la herramienta sea fácil de llevar y usar en cualquier lugar.

Como se sintetiza en la Figura 23, el uso de visión artificial frente a los sensores físicos tradicionales ofrece ventajas competitivas en términos de portabilidad, accesibilidad económica y comodidad para el usuario final.

Figura 23: Comparativa entre sistemas de captura

Actualmente, ya no hace falta equipo costoso para reconocer señas con exactitud; basta con procesar coordenadas de movimiento captadas por una cámara estándar. Este proyecto aprovecha esa tecnología para rastrear los gestos de forma natural, sin necesidad de marcadores adicionales sobre el cuerpo. Este enfoque permite que la solución sea económica y se adapte fácilmente a cualquier entorno del país, rompiendo las barreras de comunicación de una manera sencilla pero muy efectiva.

IV-C. Tecnologías de Captura y Estimación de Pose

IV-C1. MediaPipe como framework de visión por computador: $MediaPipe$ es una herramienta de software libre creada por Google que facilita el análisis de videos y gestos en tiempo real [32]. Su funcionamiento se organiza a través de una estructura de pasos ordenados, lo que permite que el sistema procese la información de la cámara de manera rápida y por etapas. Gracias a esta organización, es posible capturar los movimientos de forma fluida sin que la computadora se bloquee, separando cada análisis en partes más sencillas y eficientes.

Gracias a que MediaPipe organiza sus procesos de forma eficiente, el sistema puede funcionar en tiempo real sin ponerse lento. Para un traductor de lenguaje de señas esto es fundamental, pues asegura que el texto aparezca

en pantalla casi al mismo tiempo que se hace el gesto, ofreciendo una experiencia natural y sin esperas molestas [33].

En la Figura 24 se exponen las diversas capacidades de procesamiento de este framework, las cuales abarcan desde el reconocimiento de gestos manuales hasta el seguimiento de la postura corporal. Estos ejemplos evidencian la versatilidad de la herramienta para capturar y procesar datos biométricos de manera eficiente, estableciendo la base técnica necesaria para que el algoritmo de traducción de señas pueda operar bajo distintas condiciones de iluminación y entorno.

Figura 24: Aplicaciones interactivas del framework MediaPipe

Lo que realmente hace destacar a $MediaPipe$ frente a otros frameworks es su capacidad para operar sin necesidad de hardware costoso o una potencia de cómputo excesiva. Esto es un punto clave, ya que permite que el sistema corra sin problemas en computadoras estándar o equipos más modestos. En la práctica, esta eficiencia facilita enormemente la creación de traductores de lengua de señas que sean tanto económicos como fáciles de escalar. Además, el framework ya incluye modelos listos para rastrear manos, cuerpo y rostro de manera simultánea en un solo flujo de trabajo. Esta sincronización es vital cuando analizamos gestos dinámicos [34], donde no solo importa la forma de la mano, sino también cómo se posiciona el cuerpo para entender el mensaje completo.

IV-C2. MediaPipe Holistic: Se trata de un modelo avanzado que forma parte del entorno de trabajo $MediaPipe$ y que permite estimar de manera completa la postura y estructura del cuerpo humano a partir de secuencias de video. Este módulo permite la detección simultánea de la pose corporal, las manos y el rostro de una persona, integrando múltiples modelos especializados en un único pipeline de procesamiento.

El sistema emplea inicialmente el detector de pose BlazePose, el cual localiza la región corporal de interés dentro de la imagen. Posteriormente, se aplica un modelo de estimación de puntos clave (landmarks) que permite identificar con precisión las principales articulaciones y rasgos anatómicos del cuerpo humano. Se presenta una descripción general del funcionamiento del módulo $MediaPipe$ Holistic, donde se ilustra la detección de los puntos clave correspondientes a una persona en tiempo real [35].

En la Figura 25 se ilustra el proceso de detección simultánea de la pose corporal (BlazePose), manos y rostro a partir de una secuencia de video, integrando los modelos especializados para una reconstrucción integral de la figura humana en tiempo real.

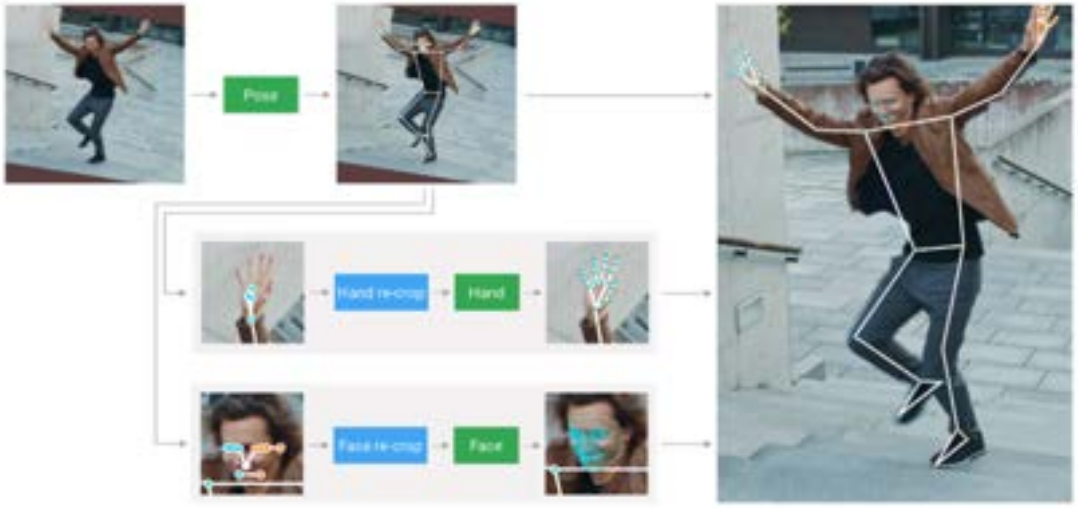


Figura 25: Arquitectura del módulo MediaPipe Holistic.

El módulo Holistic está compuesto por tres modelos principales:

- **Pose Landmark:** Es el encargado de estimar la postura corporal mediante la detección de 33 puntos clave, los cuales incluyen referencias anatómicas como la nariz, los ojos, las orejas y las principales articulaciones del cuerpo. Cada punto clave se define mediante coordenadas espaciales y un puntaje de confianza asociado, lo que permite evaluar la fiabilidad de la detección. Como se muestra en la Figura 26.

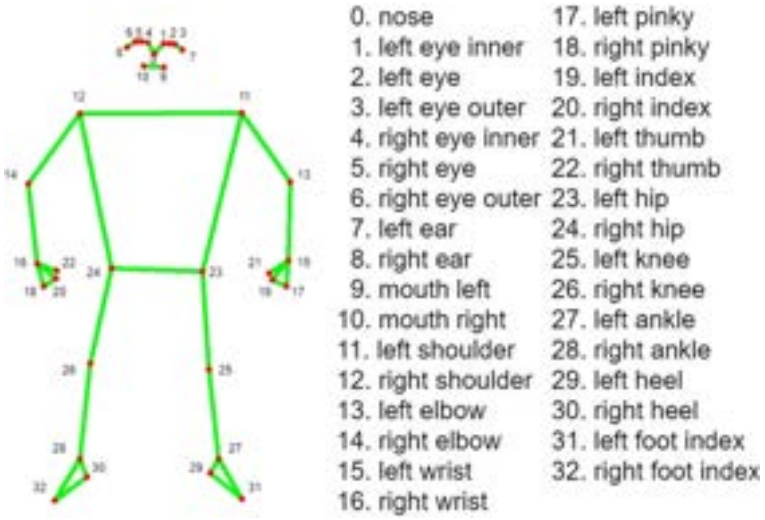


Figura 26: Identificación de los puntos clave mediante el modelo BlazePose.

- **Hand Landmark:** Es capaz de identificar 21 puntos clave en cada mano, los cuales están asociados a las articulaciones de los dedos y la palma. Este modelo es una pieza fundamental en aplicaciones de reconocimiento

de lengua de señas, pues gran parte del significado de los mensajes depende de como se mueven y se posicionan las manos. La Figura 27 ilustra detalladamente cómo se organiza esta topología dentro de MediaPipe.

Figura 27: Topología del modelo de mano en MediaPipe.

- **Face Mesh:** Como se detalla en la Figura 28, este componente realiza un recorte del rostro y utiliza un extractor de características para identificar 468 puntos clave. A través del modelo Attention Mesh, se analizan zonas específicas como los ojos y los labios para captar expresiones sutiles. En este proyecto, dicha información es vital para entender el contexto emocional y gestual que acompaña a las señas, logrando una interpretación mucho más profunda..

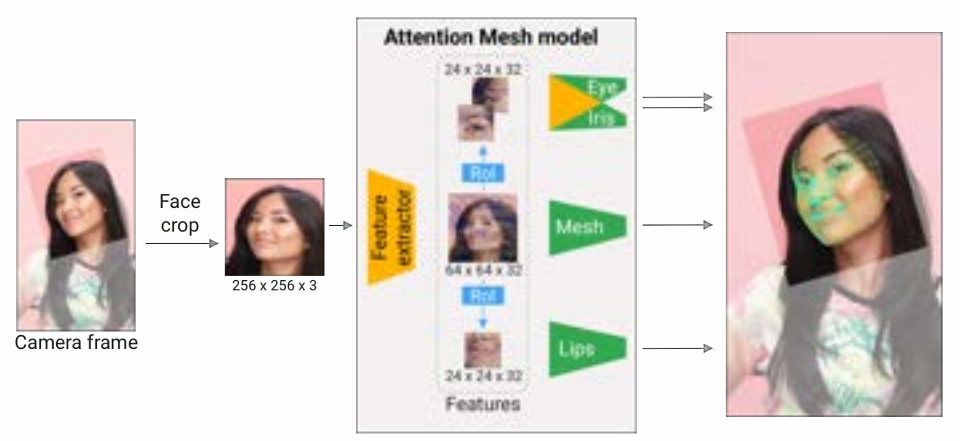


Figura 28: Flujo de extracción de puntos clave faciales mediante MediaPipe Face Mesh.

Al integrar estos tres modelos en una sola estructura, el sistema logra seguir los movimientos del cuerpo de forma sincronizada. Esta capacidad convierte a $MediaPipeHolistic$ en la herramienta ideal para analizar gestos dinámicos, permitiendo que el traductor de señas reconozca las señas con la fluidez necesaria para una comunicación real[36].

IV-C3. Estimación de puntos clave esqueléticos mediante MediaPipe Holistic: La detección de la estructura corporal se basa en identificar puntos de referencia anatómicos que funcionan como un mapa del cuerpo y sus extremidades. A través de $MediaPipeHolistic$, se obtienen coordenadas tridimensionales (x, y, z) para cada uno de estos puntos, lo que permite al sistema entender no solo la posición de la persona, sino también la profundidad y trayectoria de sus movimientos.

El sistema desglosa el cuerpo humano en tres niveles de detalle para lograr un análisis integral. Para la postura general, se identifican 33 puntos clave que abarcan el tronco y las articulaciones principales. En cuanto a las manos, se rastrean 21 puntos en cada una (42 en total) para captar con exactitud la posición de los dedos. Finalmente, se añade un mapa denso de más de 400 puntos en el rostro, lo que permite al modelo registrar hasta los gestos faciales más sutiles durante la comunicación [37].

El sistema que se muestra en la Figura 29, genera un esqueleto virtual en un espacio 3D (X, Y, Z), permitiendo el análisis de la profundidad y la evolución temporal de los gestos dinámicos más allá de una postura estática.

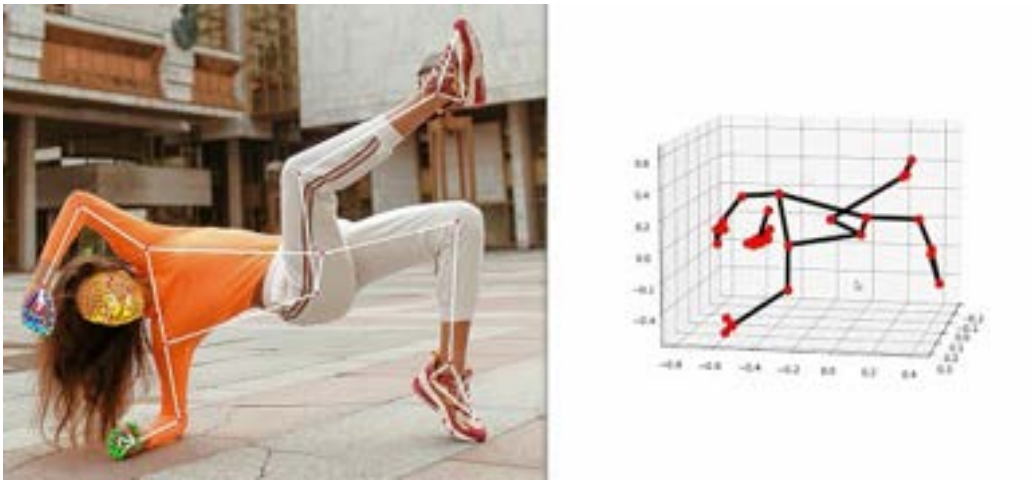


Figura 29: Representación de coordenadas tridimensionales de los landmarks.

La disponibilidad de coordenadas tridimensionales asociadas a cada landmark permite realizar un seguimiento detallado del esqueleto a lo largo del tiempo. Gracias a esta información, es posible analizar cómo evolucionan los gestos dentro de una secuencia de video, lo cual resulta especialmente relevante en el estudio de movimientos dinámicos. En este contexto, el reconocimiento de gestos no se limita a la identificación de una postura aislada, sino que depende de la transición progresiva entre distintas configuraciones corporales.

Una de las principales fortalezas de MediaPipe Holistic radica en su capacidad para operar de forma robusta en aplicaciones en tiempo real. El sistema mantiene una detección estable incluso frente a variaciones moderadas en las condiciones de iluminación, la presencia de oclusiones parciales o cambios en la orientación del cuerpo. Esta estabilidad en la estimación de puntos clave permite representar de manera cuantitativa los parámetros definidos en el modelo de William Stokoe, en particular aquellos relacionados con la posición y configuración de las manos. Como resultado, se facilita tanto el análisis matemático posterior como la integración de estos datos en algoritmos de clasificación de lengua de señas [38].

IV-C4. Extracción de características espaciales mediante landmarks de manos y pose: La extracción de características espaciales constituye una etapa esencial dentro de los sistemas de reconocimiento de gestos, ya que permite transformar la información geométrica obtenida a partir de los landmarks en descriptores numéricos adecuados para su procesamiento computacional. A partir de los puntos clave estimados por $MediaPipeHolistic$, se define un conjunto de características que describe de manera estructurada tanto la configuración estática como el movimiento de las manos y del cuerpo.

Para describir con precisión la postura, una de las técnicas que más utilizamos es el cálculo de las distancias euclidianas entre los puntos clave; esto nos ayuda a entender qué tan separadas están las articulaciones entre sí, tanto en las manos como en el cuerpo en general. También es muy útil fijarse en los ángulos que se forman entre los landmarks consecutivos, pues esos datos nos dicen mucho sobre la orientación y qué tan flexionada está una articulación. Al final, estos detalles son los que permiten al sistema no confundirse entre gestos que, a simple vista, podrían parecer iguales.

En la Figura 30 se muestra el cálculo de distintos parámetros geométricos, como el centroide, el ancho, la altura y las distancias entre puntos clave. Este proceso permite convertir la información visual en vectores numéricos, los cuales son posteriormente utilizados como entrada en algoritmos de clasificación.

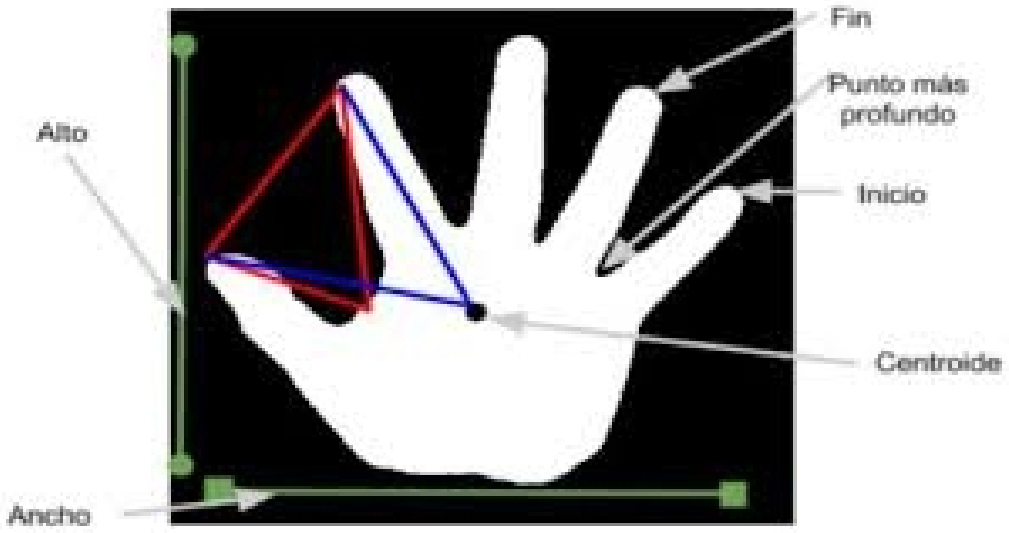


Figura 30: Geometría de la mano y extracción de puntos de control.

Con el fin de reducir la influencia de factores externos, como la distancia a la cámara o la ubicación del usuario dentro del encuadre, las características extraídas deben pasar por un proceso de normalización. Este paso resulta fundamental, ya que permite que el desempeño del sistema no dependa de quién esté frente al lente ni de las condiciones específicas en las que se capture el video. En otras palabras, se busca que el modelo responda de manera consistente aun cuando cambien variables externas que no forman parte del gesto en sí [39].

Diversos estudios han señalado que una adecuada normalización de los landmarks es uno de los elementos que más contribuye a la robustez de los sistemas de reconocimiento basados en visión artificial. Sin este ajuste previo, pequeñas variaciones en escala o posición podrían alterar significativamente los resultados.

Como se aprecia en el flujo mostrado en la Figura 31, el procedimiento comienza con la captura de las imágenes mediante OpenCV y continúa con la construcción de una malla tridimensional a partir de los puntos detectados. Este emparejamiento y organización de los puntos de referencia permite obtener una representación matemática más precisa del movimiento. Dicho paso es esencial para que posteriormente los gestos dinámicos puedan analizarse y clasificarse con mayor exactitud.

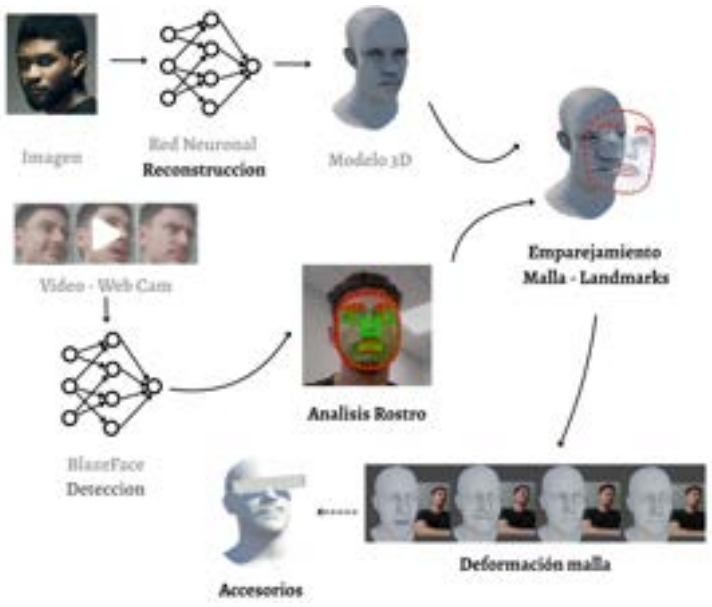


Figura 31: Pipeline de procesamiento para la reconstrucción y seguimiento facial.

Al obtener las características espaciales mediante los landmarks de manos y cuerpo, estas se convierten en los vectores de entrada para nuestros modelos de clasificación. Específicamente, alimentan redes neuronales y arquitecturas de Deep Learning diseñadas para interpretar gestos dinámicos [40]. En el campo de la interacción persona-computador, este método ha probado ser muy eficaz, ya que permite condensar el movimiento en una representación compacta pero cargada de información clave [41].

IV-C5. Librerías modernas de soporte: Construir un sistema de traducción basado en visión artificial exige el uso de herramientas que gestionen el flujo de datos de forma eficiente. En este sentido, librerías como $OpenCV$ y $NumPy$ se vuelven los pilares tecnológicos del proyecto.

Por un lado, usamos $OpenCV$ para manejar todo lo relacionado con el video en tiempo real: desde capturar las secuencias de la cámara hasta ajustar el tamaño de las imágenes y convertir espacios de color. Su gran versatilidad ha sido confirmada en múltiples investigaciones de seguimiento visual, incluso en equipos con hardware modesto [42], [43].

Por otro lado, $NumPy$ es el motor detrás del procesamiento numérico en Python. Es vital para manejar grandes cantidades de datos, permitiéndonos representar las coordenadas de los landmarks como matrices optimizadas. Esto no solo facilita cálculos complejos como la normalización o la medición de distancias, sino que reduce significativamente el esfuerzo computacional del sistema [44], [45].

La combinación de ambas herramientas permite establecer un proceso de análisis coherente. Los datos visuales que captura $OpenCV$ se transforman gradualmente en estructuras numéricas que los modelos de aprendizaje automático pueden entender. Este enfoque ya ha dado resultados exitosos en tareas de emulación de movimientos humanos y detección de posturas [46].

En la Figura 32, se observa cómo $OpenCV$ descompone la información visual en canales de color (RGB), los cuales son procesados matemáticamente por $NumPy$ como arreglos multidimensionales con rangos de intensidad específicos.

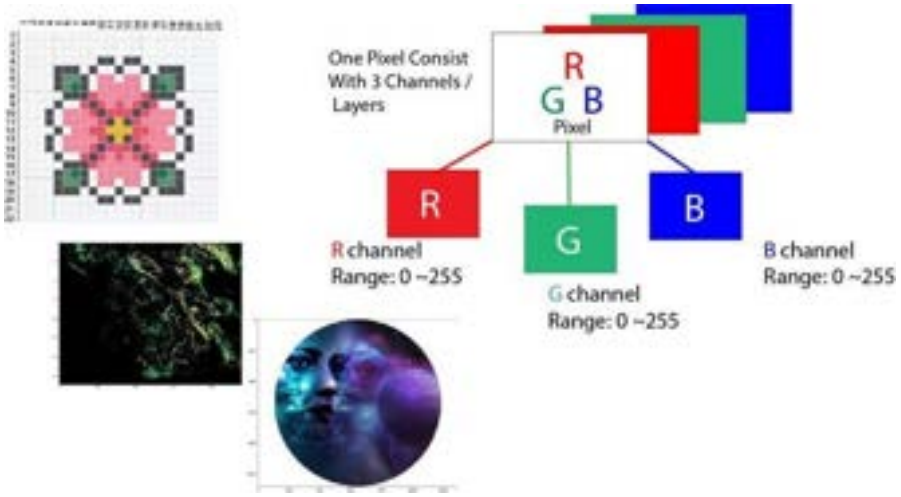


Figura 32: Representación matricial de una imagen digital.

IV-D. Inteligencia Artificial y Deep Learning

IV-D1. Definición y Estructura del Corpus Lingüístico: En el contexto del aprendizaje profundo, el concepto de corpus hace referencia al conjunto de datos que sirve como base para que un modelo pueda aprender. En este trabajo, dicho conjunto está compuesto por secuencias de video junto con las coordenadas espaciales extraídas de cada gesto. Más que un simple repositorio, el corpus funciona como la base sobre la cual el sistema identifica patrones y aprende a diferenciarlos dentro de la Lengua de Señas Ecuatoriana (LSEC).

La calidad y organización de este conjunto de datos influyen directamente en el desempeño del modelo. Si las muestras no están correctamente estructuradas o etiquetadas, el sistema no podrá generalizar adecuadamente frente a nuevas entradas [47].

La construcción del corpus se rige por tres etapas principales:

- **Captura de Datos:** En esta fase inicial se recopilaron muestras visuales mediante cámaras, procurando registrar una misma seña en distintas condiciones. Participaron varios intérpretes y se variaron factores como la iluminación y el ángulo de grabación. Esto permitió que el modelo no se limitara a un único escenario, sino que pudiera adaptarse a cambios propios del entorno real.
- **Etiquetado y Anotación (Labeling):** Una vez obtenidas las muestras, cada secuencia fue asociada con su significado correspondiente. Es decir, se estableció una relación directa entre el gesto ejecutado y la palabra o concepto que representa. Este procedimiento es esencial en el aprendizaje supervisado, ya que proporciona al algoritmo la referencia necesaria para ajustar sus parámetros durante el entrenamiento.
- **División para Entrenamiento y Validación:** Posteriormente, el conjunto total se dividió en subconjuntos con funciones distintas. El primero permitió al modelo aprender y ajustar sus pesos internos; el segundo sirvió para evaluar su rendimiento con datos que no habían sido utilizados previamente. Este paso es clave para verificar que el sistema realmente aprendió patrones generales y no simplemente memorizó ejemplos específicos.

IV-D2. Enfoques de IA para gestos dinámicos: Modelos Ocultos de Márkov (HMM) vs. Redes Neuronales: El reconocimiento de secuencias temporales, como ocurre en el habla o en los gestos dinámicos, ha sido abordado mediante diferentes enfoques de inteligencia artificial. Inicialmente predominaban los modelos estadísticos tradicionales; sin embargo, con el avance del aprendizaje profundo, comenzaron a utilizarse arquitecturas más complejas capaces de capturar dependencias temporales de mayor alcance.

IV-D3. El Enfoque Clásico: Modelos Ocultos de Márkov (HMM): Por mucho tiempo, los Modelos Ocultos de Márkov (HMM) fueron utilizados para el análisis de señales temporales. En esencia, un HMM es un modelo estadístico que interpreta el sistema como un proceso de Márkov con estados que no podemos ver directamente (ocultos). Si lo aplicamos a la Lengua de Señas Ecuatoriana (LSEC), el objetivo de un HMM sería intentar adivinar qué seña se está realizando basándose en las probabilidades de pasar de un movimiento gestual al siguiente.

Aunque estos modelos son muy útiles para representar la estructura de señales que fluyen de manera lineal ya que gestionan bien los cambios en la velocidad con la que alguien hace una seña, se quedan cortos ante la complejidad real del lenguaje de señas:

- **Dependencia de la extracción manual:** Uno de sus mayores inconvenientes es que los HMM necesitan que el investigador seleccione y procese los descriptores visuales a mano previamente. Esto no solo consume mucho tiempo, sino que puede introducir sesgos personales en los datos.
- **Falta de contexto a largo plazo:** Estos modelos sufren de lo que podríamos llamar ”memoria corta”. Les cuesta mucho reconocer señas largas o frases donde el sentido final depende de algo que ocurrió varios segundos atrás.

En la Figura 33 se pueden observar los estados internos (x) y las observaciones (y), junto con sus probabilidades de transición. Es un esquema que ilustra bien este enfoque tradicional, pero que también deja en evidencia sus limitaciones cuando se trata de recordar secuencias a largo plazo.

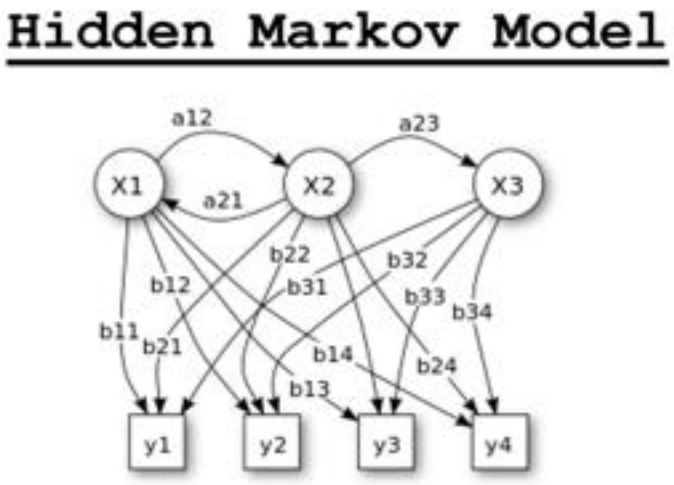


Figura 33: Estructura de un Modelo Oculto de Markov (HMM).

IV-D4. El Salto Tecnológico: Redes Neuronales Profundas (DNN): Ante las limitaciones estadísticas que presentan los HMM, las Redes Neuronales Profundas (DNN) han surgido como la alternativa más sólida. Su principal ventaja radica en su capacidad de aprendizaje no lineal y en una arquitectura diseñada para el procesamiento en paralelo. Mientras que los modelos tradicionales se basan en densidades de probabilidad, las redes neuronales van más allá: aprenden representaciones internas de los datos cruzando múltiples capas de abstracción.

Aunque los HMM tienen una base matemática firme para secuencias sencillas, las redes neuronales logran una capacidad de discriminación muy superior cuando se enfrentan a patrones altamente variables. En el caso específico de la LSEC, esto se traduce en que el sistema es mucho más tolerante a factores externos, como cambios repentinos en la iluminación, fondos complejos o incluso las diferencias físicas naturales entre distintos intérpretes [48].

La Figura 34, muestra claramente cómo este incremento en las capas ocultas permite que el modelo sea más robusto frente a variaciones ambientales.

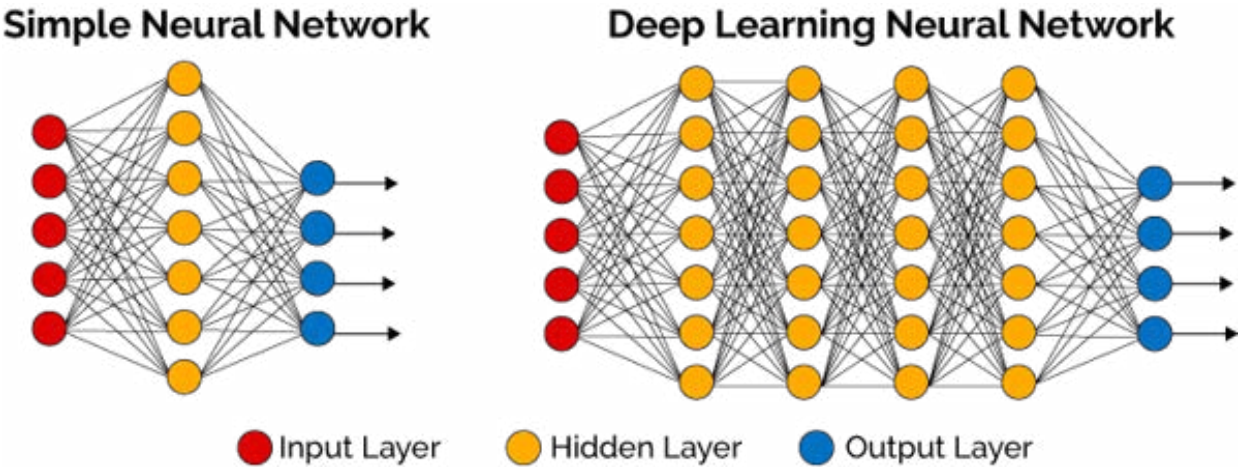


Figura 34: Comparativa entre una Red Neuronal Simple y una Red Neuronal Profunda (DNN).

IV-D5. Comparativa de Rendimiento y Flexibilidad: La principal diferencia entre ambos enfoques radica en la forma en que abordan la incertidumbre presente en las señales temporales. Por un lado, los Modelos Ocultos de Márkov buscan describir la generación de la señal mediante un esquema probabilístico, estimando las transiciones entre estados internos. Por otro, las redes neuronales adoptan una perspectiva distinta: en lugar de modelar explícitamente el proceso generativo, se enfocan en aprender directamente la relación entre la señal de entrada y la categoría de salida.

Las redes neuronales recurrentes (RNN) y arquitecturas más avanzadas como las LSTM han demostrado una mayor capacidad para manejar dependencias temporales prolongadas, lo que resulta especialmente relevante en tareas de traducción o reconocimiento en tiempo real.

Tal como se aprecia en la Tabla I, los HMM presentan limitaciones asociadas a la escalabilidad y a su alcance temporal, ya que su estructura restringe la cantidad de información pasada que pueden considerar simultáneamente. En contraste, los modelos de aprendizaje profundo permiten extraer características de manera automática, reduciendo

la necesidad de intervención manual y mostrando mayor tolerancia frente a variaciones o ruido en el entorno.

Esta diferencia se evidencia con mayor claridad en la Figura 35. Allí se observa que, conforme se incrementa el volumen de datos de entrenamiento, el desempeño del enfoque basado en Deep Learning continúa mejorando de forma progresiva. En cambio, los modelos clásicos tienden a estabilizarse en un umbral de rendimiento más bajo, lo que sugiere una menor capacidad de adaptación ante conjuntos de datos más amplios y complejos.

Tabla I: Comparativa técnica: Modelos Ocultos de Márkov (HMM) vs. Redes Neuronales Profundas (DNN)

Característica	**Modelos Ocultos de Márkov**	**Redes Neuronales (Deep Learning)**
Tipo de Aprendizaje	Estadístico / Probabilístico	Basado en datos (Inductivo)
Manejo de Características	Requiere extracción manual	Extracción automática (End-to-End)
Memoria Temporal	Limitada (Propiedad de Márkov)	Alta (Celdas de memoria LSTM)
Escalabilidad	Compleja ante grandes datasets	Alta eficiencia con Big Data
Resiliencia al ruido	Baja (Sensible a variaciones)	Alta (Capacidad de generalización)

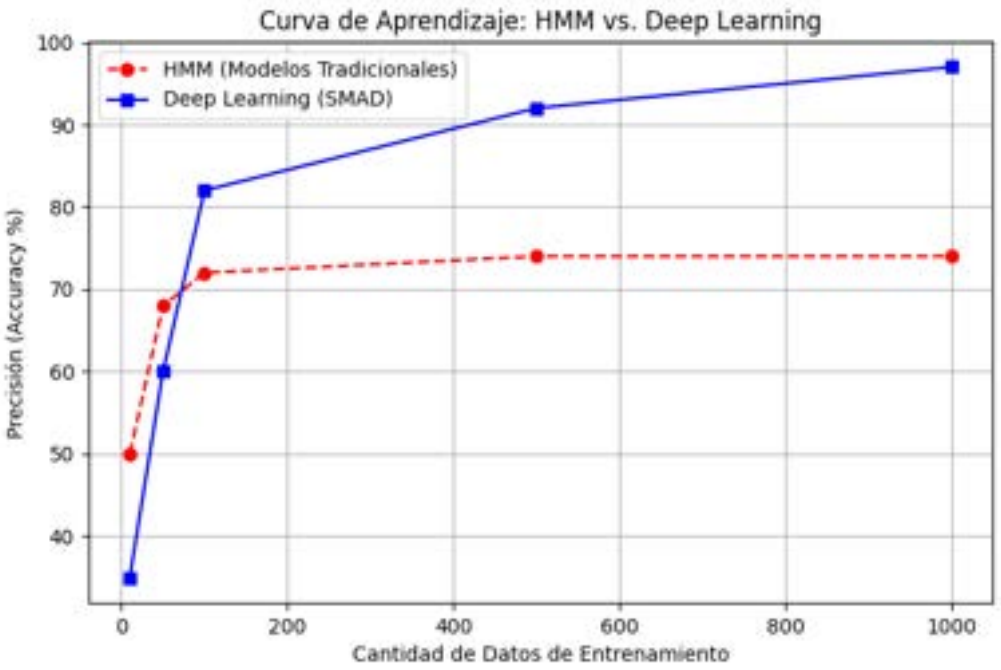


Figura 35: Curva de Aprendizaje: HMM vs. Deep Learning.

IV-D6. Redes Neuronales Convolucionales (CNN) vs. Recurrentes (RNN): En los últimos diez años, hemos sido testigos de cómo la Inteligencia Artificial y la visión computacional han dado un giro radical a la forma en que se interpreta automáticamente la lengua de señas. El gran cambio vino de la mano del Deep Learning, cuyas metodologías han desplazado a los enfoques clásicos al alcanzar niveles de exactitud por encima del 95 %. Lo más valioso de este avance es que ya no dependemos de que un experto seleccione los rasgos visuales a mano; el sistema lo hace solo, lo cual minimiza el margen de error humano y nos permite procesar la información al instante [49].

IV-D7. Redes Neuronales Convolucionales (CNN): El Análisis Espacial: Las CNN se han convertido en la herramienta predilecta para trabajar con imágenes gracias a su estructura de cuadrícula. Su gran valor reside en las capas de convolución, las cuales funcionan como filtros inteligentes que detectan patrones de forma jerárquica: desde líneas simples hasta la compleja estructura de una mano al gesticular (handshape). En la Figura 36 se puede ver cómo estos procesos matemáticos interactúan con la gramática visual para lograr el reconocimiento final.

En la Figura 36 se esquematiza este proceso, mostrando la interacción entre los modelos matemáticos de gestos, la gramática aplicada y los módulos de análisis que sustentan el reconocimiento final.

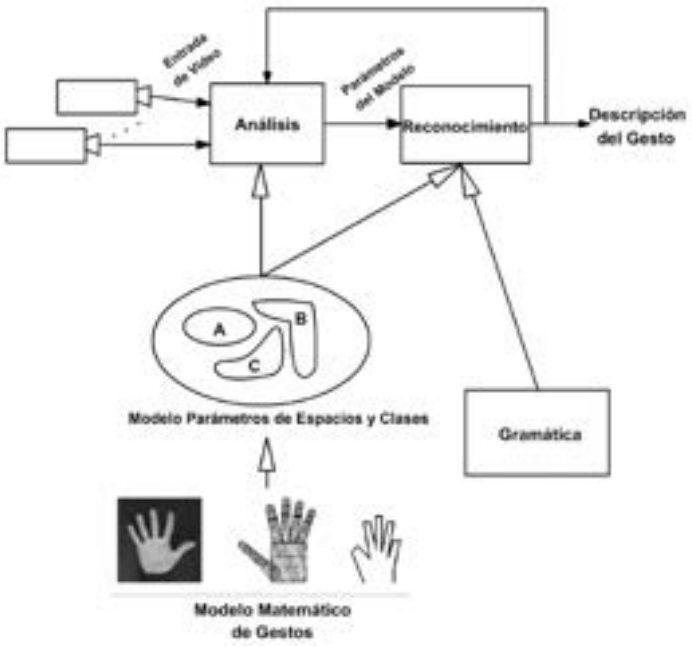


Figura 36: Diagrama del sistema de análisis y reconocimiento de gestos.

IV-D8. Redes Neuronales Convolucionales (CNN): La Extracción de Características Espaciales: El uso de las CNN ha marcado un antes y un después en cómo las máquinas interpretan el contenido visual. Al implementar ”filtros.º núcleos de convolución que recorren cada píxel, estas redes logran emular parte del procesamiento visual humano. Una de sus mayores virtudes es el principio de invarianza de traslación; esto significa que el sistema puede reconocer la configuración de una mano (handshape) con total precisión, sin importar en qué zona de la imagen aparezca el intérprete.

- **Capas de Convolución y Pooling:** Estas capas actúan como un filtro inteligente de características. Las primeras se enfocan en identificar elementos simples, como bordes y orientaciones básicas, mientras que las capas más profundas se encargan de reconocer patrones más complejos, como la posición detallada de los dedos y las articulaciones. Además, mediante el proceso de pooling o reducción, el sistema se vuelve más estable y resistente, ya que disminuye la posibilidad de errores cuando existen pequeñas variaciones en la escala o ligeras rotaciones accidentales de la mano.
- **La Limitación Estática:** A pesar de su potencia visual, la CNN carece de "noción del tiempo". Para una CNN, un video de una seña es simplemente una colección de fotos aisladas. Según Prieto (2019), esto genera una pérdida de contexto semántico, ya que en la LSEC, señas como "hola" y "adiós" pueden compartir una configuración manual similar, pero diferenciarse radicalmente en su trayectoria dinámica [50].

IV-D9. Redes Neuronales Recurrentes (RNN): La Dependencia Temporal: Para resolver la limitación de las CNN, surgen las RNN, están diseñadas para procesar secuencias de datos. Su arquitectura incluye conexiones de retroalimentación que permiten que la información persista, actuando como una "memoria de corto plazo" interna. Esto es esencial para reconocer gestos dinámicos, ya que una seña no es un punto en el espacio, sino una evolución temporal de movimientos [51].

- **Dependencia Temporal:** En la lengua de señas, el orden de los fotogramas es vital. Las RNN analizan cómo la posición de la mano en el tiempo t depende de su posición en $t-1$. Esta capacidad permite reconocer gestos continuos y fluidos, adaptándose a la velocidad de señado de diferentes usuarios.
- **El Problema del Desvanecimiento del Gradiente (Vanishing Gradient):** Este es el obstáculo técnico más significativo de las RNN clásicas. Durante el entrenamiento, al intentar propagar el error hacia atrás en el tiempo, el gradiente (la señal de aprendizaje) se vuelve infinitamente pequeño. Como resultado, la red "olvida" cómo empezó el gesto si la secuencia dura más de unos pocos segundos. Este fenómeno impide que una RNN estándar comprenda oraciones completas en LSEC donde el sujeto y el verbo están separados por varios movimientos intermedios.

IV-D10. Modelos Híbridos y Fusión Tecnológica (CNN + RNN + MediaPipe): La vanguardia tecnológica propone la integración de arquitecturas para compensar las debilidades individuales. El enfoque más robusto actualmente es la utilización de una CNN para la percepción visual y una RNN para el razonamiento temporal.

- **Paradigma de Fusión de Priyanka (2024):** Investigaciones recientes sugieren que la eficiencia del sistema aumenta drásticamente cuando se introduce un paso intermedio de extracción de landmarks mediante $MediaPipe$ [52]. En lugar de alimentar a la red con píxeles pesados, se le entregan coordenadas vectoriales. Esto permite que el modelo sea:
- **Inmune al fondo:** No importa si hay objetos detrás del intérprete, la red solo "ve" el esqueleto.
- **Eficiente en latencia:** Permite alcanzar más de 30 FPS, esencial para la traducción en tiempo real.
- **Adaptable:** La arquitectura puede ajustarse a contextos lingüísticos diversos, desde el árabe hasta la adaptación específica para el patrimonio lingüístico de la LSEC [53].

En la Figura 37, se comparan las estructuras generales de una CNN para la percepción visual y una RNN para el razonamiento temporal, cuya fusión permite procesar secuencias de video con alta precisión lingüística.

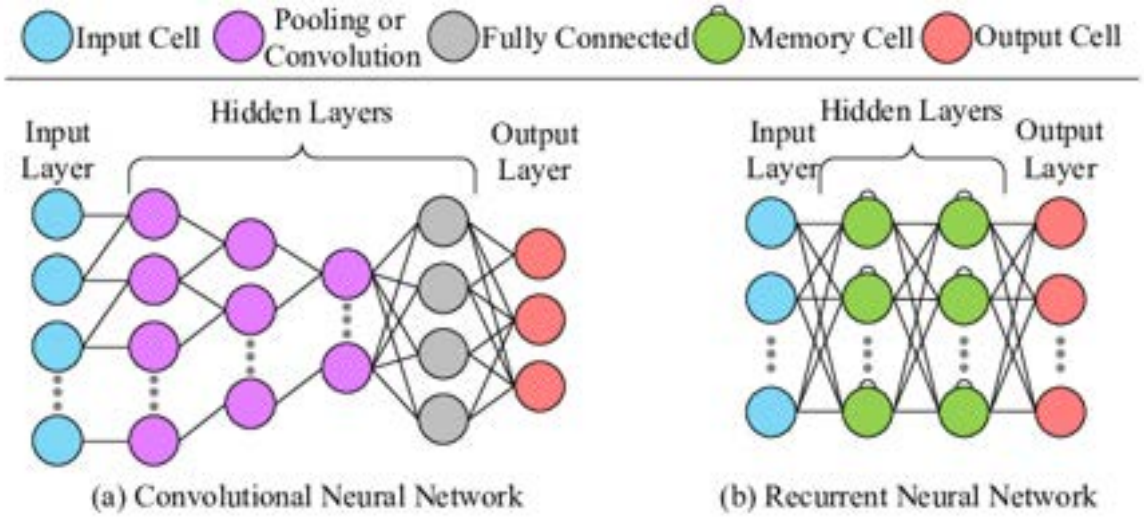


Figura 37: Estructuras generales de las CNN y las RNN.

IV-D11. Arquitectura Long Short-Term Memory (LSTM): Las redes $LongShort-TermMemory(LSTM)$ constituyen una evolución de las redes neuronales recurrentes (RNN) diseñada para superar las limitaciones asociadas a la memoria a corto plazo, particularmente los problemas de desvanecimiento y explosión del gradiente durante el entrenamiento de secuencias largas. Esta arquitectura introduce una celda de memoria capaz de preservar información relevante durante intervalos prolongados mediante un sistema de compuertas que regulan el flujo de datos [54], [55].

La estructura interna de un bloque LSTM, detallada en la Figura 38, muestra cómo el sistema de compuertas regula el flujo de información para preservar datos relevantes durante intervalos prolongados.

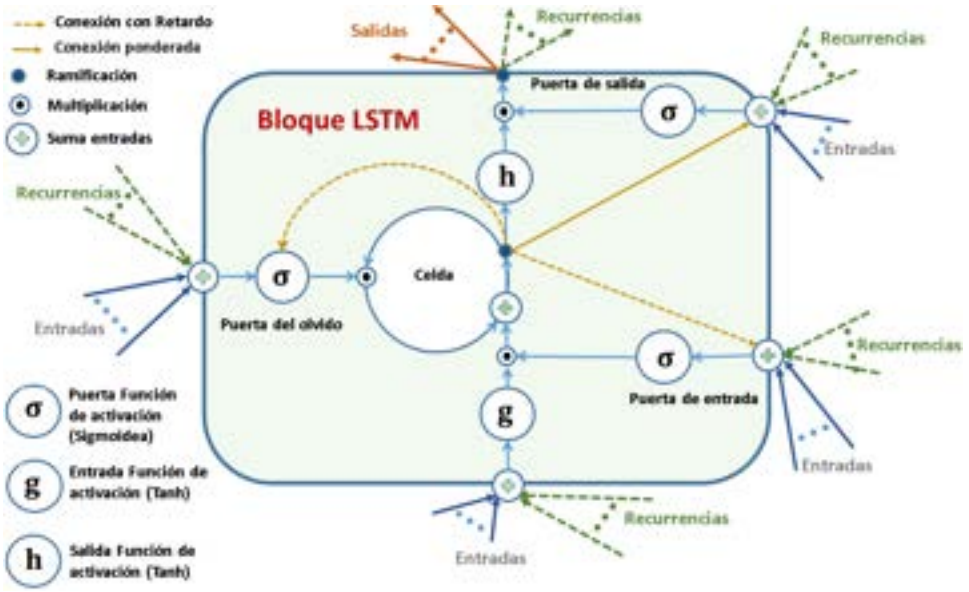


Figura 38: Estructura interna de un bloque LSTM y flujo de información a través de sus puertas.

En aplicaciones como el reconocimiento de gestos dinámicos y lengua de señas, donde una seña puede extenderse durante varios segundos y depender del contexto gestual previo, esta capacidad de memoria resulta fundamental.

Diversos estudios han demostrado que la combinación de LSTM con la extracción de *landmarks* espaciales mediante $MediaPipeHolistic$ mejora significativamente la robustez del reconocimiento en tiempo real.

IV-D12. Componentes y flujo de datos en una celda LSTM: A diferencia de una neurona recurrente convencional, la unidad LSTM gestiona el flujo de información a través de estructuras de datos multidimensionales representadas como tensores.

El procesamiento interno se organiza en torno a dos estados principales:

- **Estado de la celda** (C_t): Representa una memoria de largo plazo que actúa como un canal de transporte de información a lo largo de la secuencia temporal.
- **Estado oculto** (h_t): Contiene una representación filtrada de la información relevante en el instante actual y es utilizado tanto para la salida del modelo como para la retroalimentación interna.

En cada instante de tiempo t, la celda LSTM recibe el vector de entrada actual x_t (por ejemplo, una concatenación de *landmarks* de manos y pose) junto con los estados C_{t-1} y h_{t-1}, generando una nueva salida y actualizando la memoria para el siguiente paso temporal.

IV-D13. Mecanismo de compuertas y olvido selectivo: El principio central de la arquitectura LSTM consiste en su capacidad para discernir qué información es vital conservar y cuál descartar. A diferencia de las redes convencionales, esta celda utiliza un sistema de çompuertas"basadas en activaciones sigmoideas. Estas funciones actúan como reguladores que entregan valores entre 0 y 1, permitiendo un control milimétrico sobre el flujo de datos.

La compuerta de olvido es la encargada de decidir qué parte del historial de memoria sigue siendo útil para las predicciones.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

Un valor cercano a cero implica el descarte de la información pasada, mientras que valores próximos a uno permiten su conservación. Este mecanismo resulta especialmente útil para eliminar movimientos antiguos que ya no contribuyen a la interpretación de la seña actual.

La compuerta de entrada controla la incorporación de nueva información al estado de memoria:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

De forma paralela, se calcula un vector de valores candidatos mediante una función tangente hiperbólica:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

La actualización del estado de la celda se realiza combinando la información retenida y la nueva información candidata:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

La compuerta de salida determina el estado oculto que será propagado a la siguiente unidad temporal:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Este esquema de compuertas permite que la red aprenda de manera diferencial qué eventos del pasado inmediato son críticos para interpretar la acción presente, característica esencial en la transición continua de movimientos propios de la lengua de señas.

Como se detalla en la Figura 39, la interacción entre las compuertas de olvido, entrada y salida permite que la red mantenga un estado de celda actualizado, garantizando la transición continua entre los movimientos complejos de la lengua de señas.

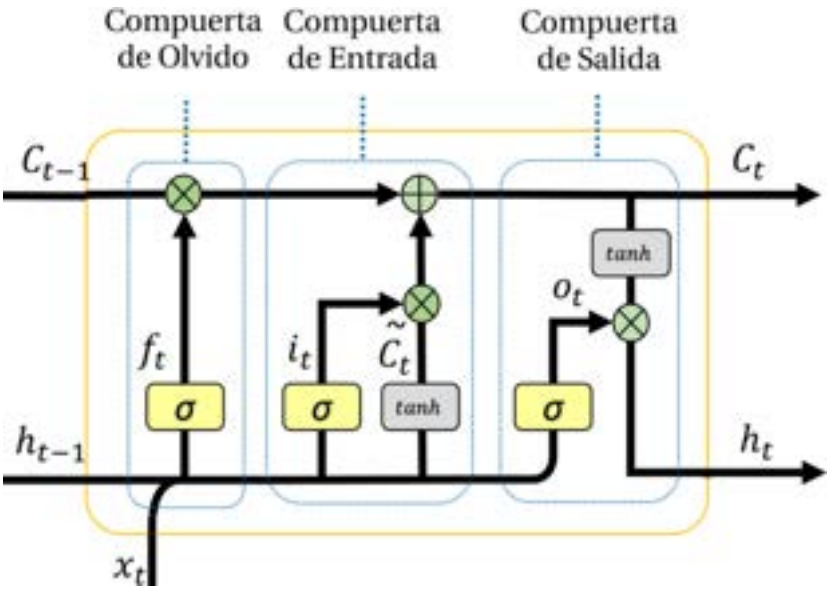


Figura 39: Estructura interna de una celda LSTM.

IV-D14. Memoria de largo plazo en el reconocimiento de gestos dinámicos: La capacidad de memoria de las redes LSTM permite analizar movimientos extensos sin que el sistema pierda información durante el proceso. En el reconocimiento de gestos, esta propiedad es clave para identificar patrones dinámicos complejos y, sobre todo, para distinguir entre señas que se parecen físicamente pero que se ejecutan con ritmos o direcciones diferentes.

Investigaciones actuales confirman que combinar redes LSTM con los puntos clave (landmarks) de MediaPipe genera resultados muy sólidos. Este enfoque permite que el sistema funcione correctamente aunque cambie la velocidad, la orientación o la forma en que cada persona hace el gesto, posicionándose como una de las mejores opciones tecnológicas para traducir la lengua de señas de manera automática [56], [57].

IV-E. Clasificación de series temporales para el reconocimiento de acciones

La clasificación de series temporales es fundamental cuando los datos dependen directamente del orden en que ocurren. A diferencia de los métodos estáticos, que analizan cada imagen por separado, este enfoque estudia la

evolución del movimiento para captar patrones que cambian con el tiempo. Esto es esencial en la lengua de señas, donde el significado no reside en una postura fija, sino en la secuencia completa de gestos que se conectan entre sí [58].

En el reconocimiento de lengua de señas, los movimientos se analizan como secuencias ordenadas de datos que se capturan en intervalos constantes. Al utilizar coordenadas tridimensionales de puntos clave del cuerpo, el sistema transforma cada gesto en una trayectoria dentro del espacio y el tiempo. Esta representación permite que los modelos de aprendizaje profundo identifiquen con precisión la dinámica y el sentido de cada acción ejecutada.

IV-E1. Representación temporal de acciones humanas: Una fase esencial del proceso es cómo se organiza la información a través del tiempo. En lugar de analizar imágenes aisladas, el sistema interpreta cada seña como una secuencia continua de datos, capturando en cada instante detalles clave como la posición de las articulaciones y la distancia entre los dedos para entender el movimiento completo.

Este enfoque es común en el reconocimiento de actividades humanas, ya que logra capturar tanto la forma del cuerpo como el ritmo de sus movimientos. La experiencia en este campo demuestra que incluir la dimensión temporal mejora mucho la capacidad del sistema para distinguir entre señas que se parecen físicamente, pero que tienen velocidades o trayectorias distintas [59].

En la Figura 40, se muestra la descomposición de un gesto en poses discretas para facilitar el análisis temporal. Al estudiar la progresión de estas posturas, el sistema logra reconocer la dinámica del movimiento y clasificar la seña de forma correcta.

Figura 40: Representacion de una secuencia de poses

IV-E2. Modelos recurrentes para la clasificación de series temporales: Las redes neuronales recurrentes (RNN) destacan por su capacidad de integrar información previa en el análisis del presente, lo que las hace fundamentales para estudiar secuencias temporales. No obstante, dado que las RNN básicas presentan dificultades para retener datos en secuencias extensas, se desarrollaron arquitecturas más robustas como las LSTM. Estas últimas permiten gestionar la memoria a largo plazo, garantizando que el sistema no pierda el contexto del movimiento durante la ejecución de una seña. En tareas de clasificación, las LSTM permiten mapear una secuencia de longitud variable a una etiqueta de clase, siguiendo un esquema de tipo *many-to-one*. En este enfoque, la red procesa la secuencia completa y produce una única salida representativa de la acción reconocida. Investigaciones orientadas al reconocimiento de actividades humanas han evidenciado que este tipo de arquitectura logra capturar dependencias temporales complejas, superando a métodos basados únicamente en características estáticas [60].

IV-E3. Extracción de patrones temporales y toma de decisión: Durante el entrenamiento, el modelo identifica patrones clave como la velocidad, la duración y la coordinación de los movimientos del cuerpo. Gracias a la arquitectura LSTM, el sistema no analiza los datos de forma aislada, sino que mantiene una memoria de los movimientos previos. Esto permite que la representación de la seña se perfeccione gradualmente a medida que transcurre la secuencia, logrando una interpretación mucho más precisa del gesto completo.

Desde un punto de vista matemático, el proceso de clasificación se define como una función que transforma una serie de datos en el tiempo en una categoría específica. Este método ha demostrado ser muy eficaz en áreas tan diversas como las finanzas o el estudio del comportamiento humano, lo que confirma su gran capacidad para adaptarse a diferentes tipos de información y extraer resultados precisos [61].

Se observa en la Figura 41, el flujo desde la secuencia de entrada (vectores de características espaciales), pasando por la capa oculta que procesa la información temporal, hasta generar una secuencia de salida que representa la predicción o clasificación final del gesto.

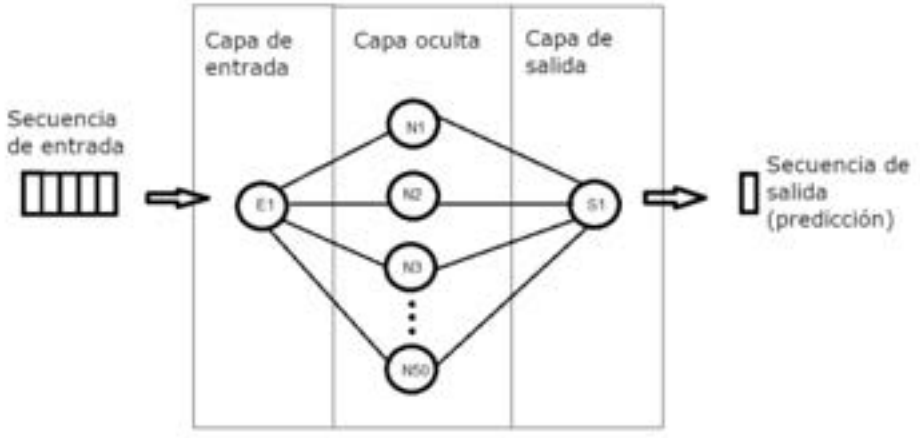


Figura 41: Diagrama de la arquitectura de la red neuronal.

IV-E4. Clasificación supervisada y evaluación del desempeño: El entrenamiento del sistema se basa en un aprendizaje supervisado, donde cada movimiento se etiqueta con su significado correspondiente. Para medir el éxito del modelo, se utilizan métricas como la precisión y la matriz de confusión, las cuales permiten identificar qué señas se confunden entre sí. Este análisis de errores es clave, ya que ayuda a ajustar el algoritmo para que pueda diferenciar gestos muy similares con mayor exactitud.

Desde una perspectiva más amplia, el análisis de patrones temporales no solo permite la clasificación de acciones, sino también la detección de regularidades y anomalías en secuencias de comportamiento humano, lo que ha sido explorado en estudios orientados al análisis de interacción social y monitoreo de procesos cognitivos [62].

En la Figura 42, se observa que ilustra la relación entre las clases reales (señas ejecutadas) y las predicciones del modelo basado en redes LSTM.

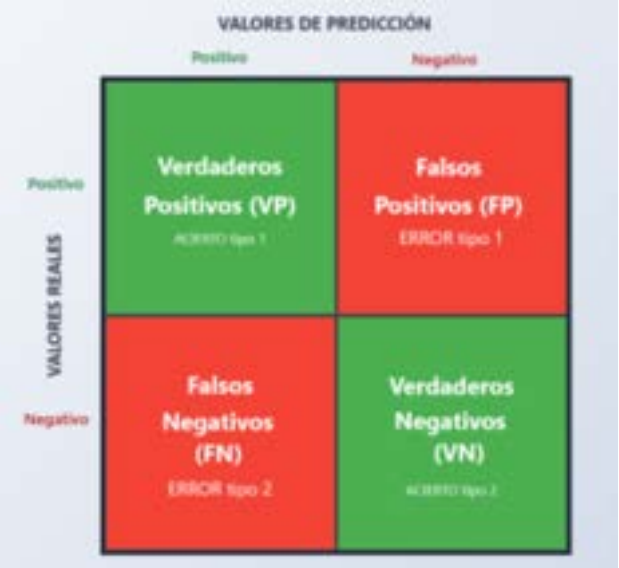


Figura 42: Representación del desempeño del sistema mediante métricas de precisión y sensibilidad.

IV-E5. Relevancia para el reconocimiento de lengua de señas: En este proyecto, se considera que una seña no es una imagen estática, sino una secuencia de movimientos que fluyen en el tiempo. Por ello, la clasificación de series temporales es el núcleo del sistema, ya que permite analizar cómo se encadenan estos pequeños gestos. Al modelar estas secuencias, el software puede distinguir entre señas que se parecen físicamente pero que tienen ritmos o direcciones diferentes, lo que reduce errores y hace que la traducción sea mucho más exacta.

Por lo tanto, la combinación de redes LSTM con el seguimiento de los puntos clave (landmarks) permite que el sistema reconozca acciones de forma fluida y en tiempo real. Esta integración resulta ser una solución robusta y eficaz, ya que aprovecha los últimos avances en inteligencia artificial para lograr que la interacción entre el usuario y la computadora sea lo más natural y precisa posible.

IV-F. Algoritmos de optimización: el motor Adam

Entrenar una red neuronal para reconocer la Lengua de Señas Ecuatoriana es, en esencia, un proceso de ajuste constante. El objetivo es reducir al mínimo la 'función de pérdida' (loss function), la cual indica qué tan alejada está la predicción de la inteligencia artificial respecto a la seña real. Para lograr este aprendizaje, se utiliza un optimizador, que es el algoritmo encargado de ajustar los pesos de la red de manera eficiente hasta que el sistema logre identificar los gestos con precisión.

IV-F1. Evolución de los métodos de descenso de gradiente: Los métodos tradicionales, como el Descenso de Gradiente Estocástico (SGD), utilizan una velocidad de aprendizaje fija que resulta limitada para reconocer gestos complejos. Al ser movimientos tan variables, una tasa rígida puede hacer que el entrenamiento sea muy lento o que el modelo nunca logre aprender correctamente, dificultando la precisión del sistema [63].

En la Figura 43 se compara el comportamiento del Descenso de Gradiente Estocástico (SGD) frente a optimizadores modernos, evidenciando cómo estos últimos logran una trayectoria más directa y estable.

Como solución, se emplean métodos con tasas de aprendizaje adaptativas que ajustan automáticamente la velocidad del entrenamiento. Este enfoque funciona de manera similar al control de un brazo robótico: regula la intensidad de los cambios para evitar movimientos bruscos en el aprendizaje, garantizando que el modelo sea

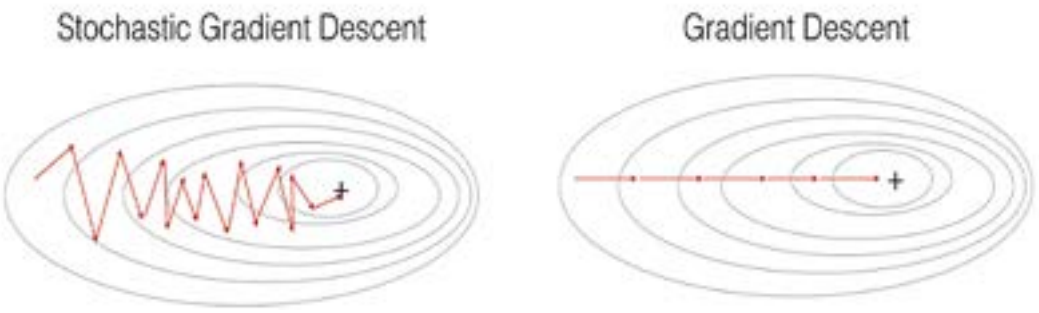


Figura 43: Comportamiento de convergencia entre el Descenso de Gradiente Estocástico (SGD) y optimizadores.

estable y preciso al identificar las señas [64].

IV-F2. Arquitectura y funcionamiento del algoritmo Adam: Adam destaca por ajustar la velocidad de aprendizaje de forma individual para cada parámetro, combinando eficiencia y estabilidad. Esta capacidad de adaptación acelera el entrenamiento y mejora la precisión del modelo, permitiendo que el sistema identifique los movimientos de las señas con mayor rapidez y exactitud [65].

Adam estima el primer y segundo momento del gradiente a través de promedios móviles exponenciales. Matemáticamente, estas estimaciones se definen como:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Donde:

- g_t es el gradiente en el tiempo actual.
- β_1 y β_2 son los hiperparámetros que controlan las tasas de decaimiento (comúnmente $0{,}9$ y $0{,}999$ respectivamente).
- m_t representa el "momentum" (la dirección del movimiento).
- v_t representa la "curvatura" o escala de los datos.

Este mecanismo incluye una corrección de sesgo que permite estabilizar el aprendizaje desde las primeras iteraciones, evitando actualizaciones abruptas que podrían afectar negativamente el entrenamiento de secuencias complejas, como las presentes en la lengua de señas.

La trayectoria representada en la Figura 44 es el descenso del optimizador hacia el mínimo global para la reducción del error de entrenamiento.

IV-F3. Ventajas técnicas en el procesamiento de la LSEC: La aplicación del optimizador Adam en el reconocimiento de la LSEC ofrece ventajas sustanciales frente a métodos clásicos. Las señas están compuestas por micro-movimientos de los dedos y macro-movimientos de brazos y torso, lo que genera características de entrada con escalas heterogéneas. Adam gestiona automáticamente esta disparidad, ajustando la magnitud de las actualizaciones de los pesos en función de la relevancia estadística de cada característica.

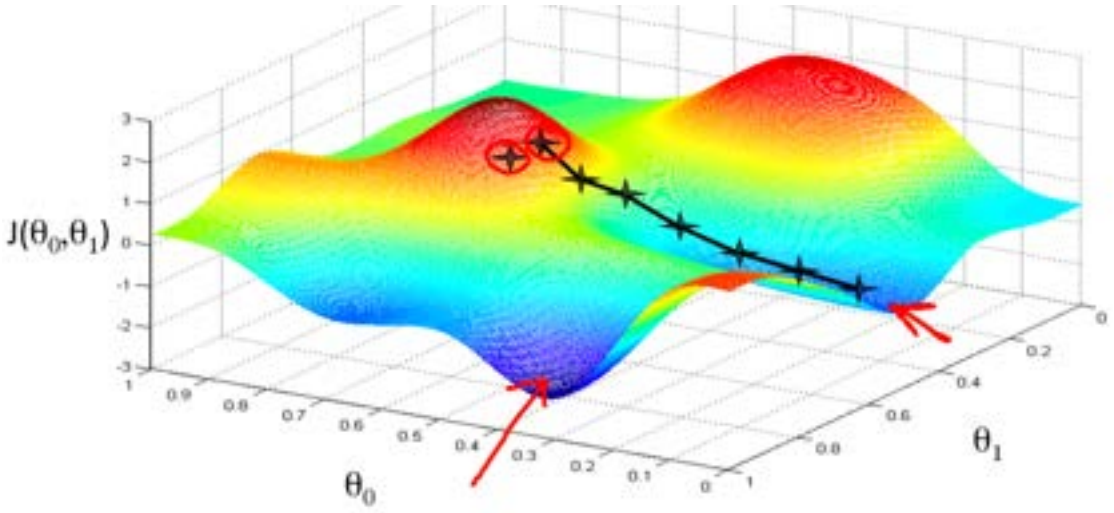


Figura 44: Visualización tridimensional de la función de coste.

Además, en aplicaciones que demandan alta fidelidad en la clasificación o reconstrucción de datos, como ocurre en técnicas avanzadas de procesamiento visual y modelado tridimensional, el uso de optimizadores eficientes permite reducir significativamente el tiempo de cómputo sin comprometer la calidad del resultado [66].

IV-F4. Comparativa conceptual con otros motores de optimización: Existen otros motores como Nadam o Adadelta, pero Adam sigue siendo el estándar en la comunidad de Deep Learning por las siguientes razones:

- **Eficiencia Computacional:** Requiere poca memoria y es ideal para problemas con grandes volúmenes de datos o parámetros.
- **Hiperparámetros Intuitivos:** Los valores por defecto suelen funcionar excelentemente para casi todos los problemas de visión artificial.
- **Manejo de Ruido:** Es muy eficaz en problemas donde los gradientes son ruidosos o tienen mucha variabilidad, como sucede en los videos de lengua de señas capturados con cámaras web de baja resolución.

Como se detalla en la Tabla II, el algoritmo Adam destaca por su tasa de aprendizaje adaptativa y robustez frente al ruido, ventajas que se confirman visualmente en la Figura 45, donde se observa una reducción drástica de la tasa de error y una estabilidad superior en comparación con el método SGD clásico.

Tabla II: Comparativa técnica de algoritmos de optimización: SGD, RMSProp y Adam.

Característica	SGD (Clásico)	RMSProp	Adam
Tasa de Aprendizaje	Fija	Adaptativa	Adaptativa + Momentum
Velocidad de Convergencia	Lenta	Media	Muy Rápida
Memoria Requerida	Mínima	Baja	Baja / Media
Robustez al Ruido	Baja	Alta	Muy Alta
Ajuste de Hiperparámetros	Manual (Complejo)	Moderado	Intuitivo / Automático

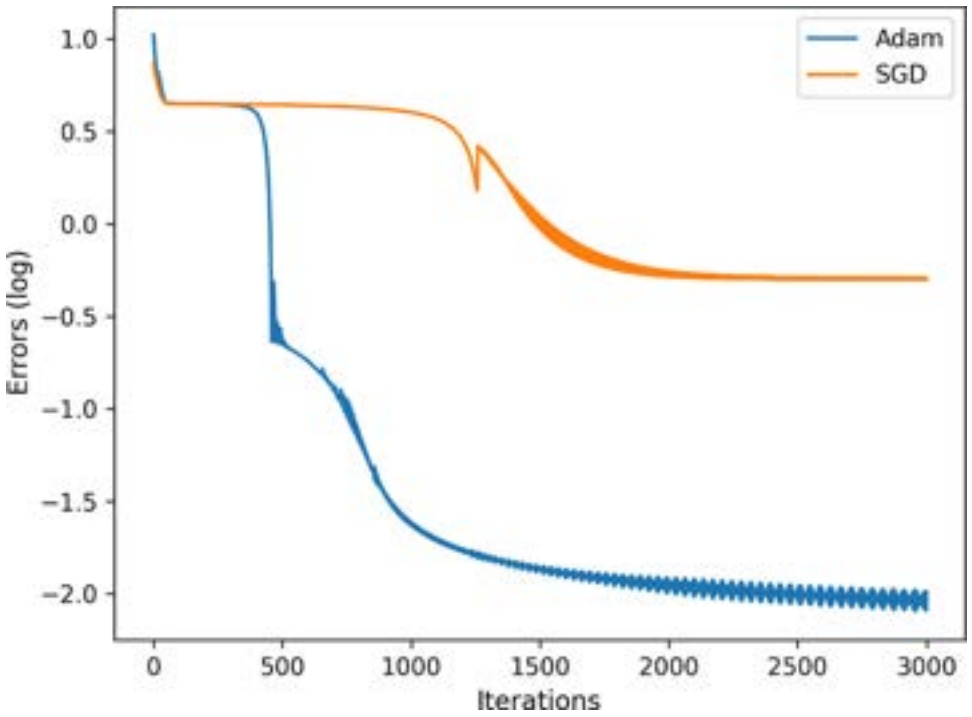


Figura 45: Análisis comparativo de la tasa de error (log) en función del número de iteraciones.

IV-F5. Relevancia en sistemas de traducción de lengua de señas: Para que un sistema de traducción sea confiable, el proceso de aprendizaje de la red neuronal debe ser estable y preciso. En este sentido, el uso del optimizador Adam resulta fundamental, ya que permite que el modelo aprenda a reconocer gestos largos y complejos de forma gradual. Gracias a su capacidad para ajustar el ritmo de aprendizaje de manera automática, se evita que la red se confunda con datos irrelevantes o que deje de mejorar antes de tiempo, asegurando que el sistema final sea capaz de entender las señas con mayor exactitud.

De este modo, el uso de este optimizador influye directamente en la precisión del sistema al interpretar señas dinámicas. Su capacidad para ajustar el aprendizaje de la red neuronal permite que el modelo final sea lo suficientemente ligero y eficiente como para operar en tiempo real, sin retrasos que afecten la comunicación.

IV-G. Estado del arte

La investigación sobre el procesamiento de lenguajes visuales ha avanzado significativamente, pasando de pruebas controladas a soluciones prácticas de accesibilidad. Este apartado recopila los antecedentes históricos y técnicos, tanto en el extranjero como en el contexto ecuatoriano, que justifican la arquitectura y el enfoque del algoritmo de traducción de señas presentado en esta investigación.

IV-G1. Avances tecnológicos recientes en el reconocimiento automático del LSEC: A nivel internacional, el estudio de las lenguas de señas pasó por una etapa inicial donde la tecnología solo permitía analizar gestos de forma aislada. Mediante el uso de plantillas de comparación, se intentaba reconocer señas como si fueran fotografías, dejando de lado la naturaleza dinámica del lenguaje. Esta limitación técnica en los sistemas pioneros de ASL y lenguas europeas impedía una interpretación natural, reduciendo la efectividad de las herramientas a unos pocos conceptos básicos [67].

La llegada del Deep Learning marcó un punto de inflexión, permitiendo que la tecnología pasara de analizar imágenes aisladas a comprender secuencias de video completas. Este cambio de paradigma fue crucial, ya que permitió que los sistemas dejaran de "ver" poses estáticas y comenzaran a interpretar la trayectoria del movimiento. En investigaciones desarrolladas en Estados Unidos, se implementaron redes neuronales avanzadas para captar tanto la profundidad como la velocidad de los gestos, reconociendo finalmente que la lengua de señas no es una sucesión de fotos, sino una evolución constante de expresiones corporales y faciales.[68] [69].

En el contexto europeo, las investigaciones han evolucionado hacia sistemas capaces de combinar video convencional con el rastreo de los puntos clave del esqueleto humano, lo que mejora la precisión incluso en lugares con mala iluminación o fondos distractores. El éxito de estos trabajos radica en que la inteligencia artificial ahora puede aprender por sí misma a identificar patrones complejos entre miles de ejemplos. Al automatizar este proceso, se elimina la necesidad de que el investigador defina manualmente cada rasgo del gesto, reduciendo así los errores humanos y permitiendo que el sistema sea mucho más confiable.

IV-G2. Antecedentes Nacionales: Situación de la comunidad sorda en Ecuador: En el contexto ecuatoriano, el desarrollo de tecnología asistiva no es solo un reto técnico, sino una urgencia social. La situación sociolaboral de la comunidad sorda en Ecuador presenta brechas significativas; se estima que la limitación en la comunicación fluida restringe su acceso equitativo a la educación superior y a empleos técnicos, afectando su participación en la Población Económicamente Activa (PEA) .

Históricamente, la solución ha sido la contratación de intérpretes humanos físicos. Sin embargo, esta alternativa presenta dos problemas críticos:

- **Costo Elevado:** La disponibilidad de intérpretes certificados es escasa y su costo por hora es prohibitivo para el uso cotidiano o en pequeñas instituciones.
- **Dependencia:** Genera una falta de autonomía en el usuario sordo para realizar trámites básicos o interacciones sociales espontáneas.

A nivel nacional, las investigaciones se han centrado en la creación de bases de datos específicas que respeten la gramática y morfología propias del país. Dado que la LSEC posee particularidades culturales y gestuales únicas [70].

La Inteligencia Artificial surge como la solución más viable económicamente. Al utilizar modelos ligeros que funcionan con cámaras web estándar y bibliotecas de software abierto como $TensorFlow$ y $OpenCV$, es posible

democratizar el acceso a la traducción de la Lengua de Señas Ecuatoriana (LSEC). Esto permite que el sistema SMAD sea escalable y de bajo consumo energético, acercando la tecnología al uso diario sin necesidad de hardware costoso como guantes de datos o sensores Kinect.

IV-G3. Análisis comparativo de proyectos similares (Investigaciones 2023-2024): Enfoques Basados en Deep Learning y Grafos (2023-2024) Investigaciones recientes sugieren que el aprendizaje profundo es la tecnología más prometedora para identificar patrones complejos en grandes volúmenes de datos. Se ha observado que la combinación de redes CNN, RNN y GCN (Graph Convolutional Networks) facilita el análisis del esqueleto humano en 3D, mejorando la interpretación de posturas y trayectorias con alta exactitud.

- **MediaPipe vs. Segmentación de Piel (Skin Filter):**En proyectos anteriores, se utilizaba el "Skin Filter" (filtro de piel) para intentar aislar la mano. Sin embargo, este método es altamente sensible a los cambios de iluminación y al color de fondo. El uso de $MediaPipeHands$ (Figura 14) es superior porque extrae 21 puntos clave tridimensionales directamente. Esto crea un "mapa matemático" de la mano que es inmune al ruido visual del fondo.
- **Fusión Multimodal (2024):** Investigaciones actuales han demostrado que integrar coordenadas esqueléticas con redes CNN y RNN facilita el análisis de posturas y trayectorias con una exactitud mucho mayor que los métodos que solo analizan píxeles.
- **Adaptabilidad Local:** A diferencia de sistemas extranjeros, los proyectos nacionales de 2023 han empezado a crear bases de datos específicas para la gramática y morfología de la LSEC, reconociendo que cada región posee variaciones gestuales únicas.
- **Localización y Adaptabilidad:** El Factor Local A nivel local, el proyecto SMAD se alinea con la tendencia de eliminar la necesidad de guantes o marcadores adicionales, favoreciendo un sistema económico y adaptable. La consolidación de estas investigaciones busca sentar las bases para una traducción bidireccional entre texto, voz y señas a largo plazo.

La literatura actual destaca que, aunque se han logrado avances con sensores como el Kinect, el futuro reside en la independencia del hardware especializado, permitiendo que cualquier cámara web estándar sea un puente de comunicación inclusiva.

En la Tabla III se presenta una comparativa de los hallazgos alcanzados por diversos autores entre 2024 y 2026, permitiendo situar al Proyecto SMAD dentro de la tendencia actual de optimización para hardware estándar y adaptación específica a la gramática local

Tabla III: Análisis comparativo de arquitecturas de Deep Learning para el reconocimiento de lenguas de señas (Investigaciones 2023-2025)

Autor / Referencia	Año	Modelo	Lengua	Hallazgo / Precisión
González et al.	2024	LSTM + MediaPipe	LSM	97 % de precisión en entornos controlados mediante app móvil.
Alsharif et al.	2025	YOLOv11 + MediaPipe	ASL	mAP@0.5 de 98.2 % usando seguimiento de keypoints en tiempo real.
Noor et al.	2024	Hybrid CNN-LSTM	ArSL	94.4 % de exactitud en señas árabes con fondos complejos.
Rajalakshmi et al.	2025	3D CNN + Bi-LSTM	ISL	Uso de atención híbrida para mejorar la captura temporal.
Proyecto SMAD	**2026**	**LSTM + Holistic**	**LSEC**	**Optimización para hardware estándar y gramática ecuatoriana.**

IV-G4. Conclusión del Estado del Arte: A pesar de los avances descritos, persiste un vacío significativo: la mayoría de los sistemas robustos aún requieren hardware especializado o están diseñados para lenguas extranjeras. El proyecto SMAD viene a llenar este vacío al ofrecer una solución que:

- Es específica para la LSEC, respetando sus variantes culturales.
- Es totalmente dinámica, analizando secuencias de movimiento y no solo poses fijas.

Es económicamente accesible, eliminando la barrera del hardware extra y funcionando en dispositivos convencionales mediante el uso optimizado de redes neuronales y visión por computadora.

V. MARCO METODOLÓGICO

Este capítulo presenta las fases del proceso metodológico que guiará el desarrollo del traductor de señas, como se ilustra a continuación.

Como se observa en la Figura 46, se presenta la metodología dividida en cuatro etapas, las cuales permitieron organizar desde la recolección de las señas hasta la puesta en marcha del traductor en la plataforma web.

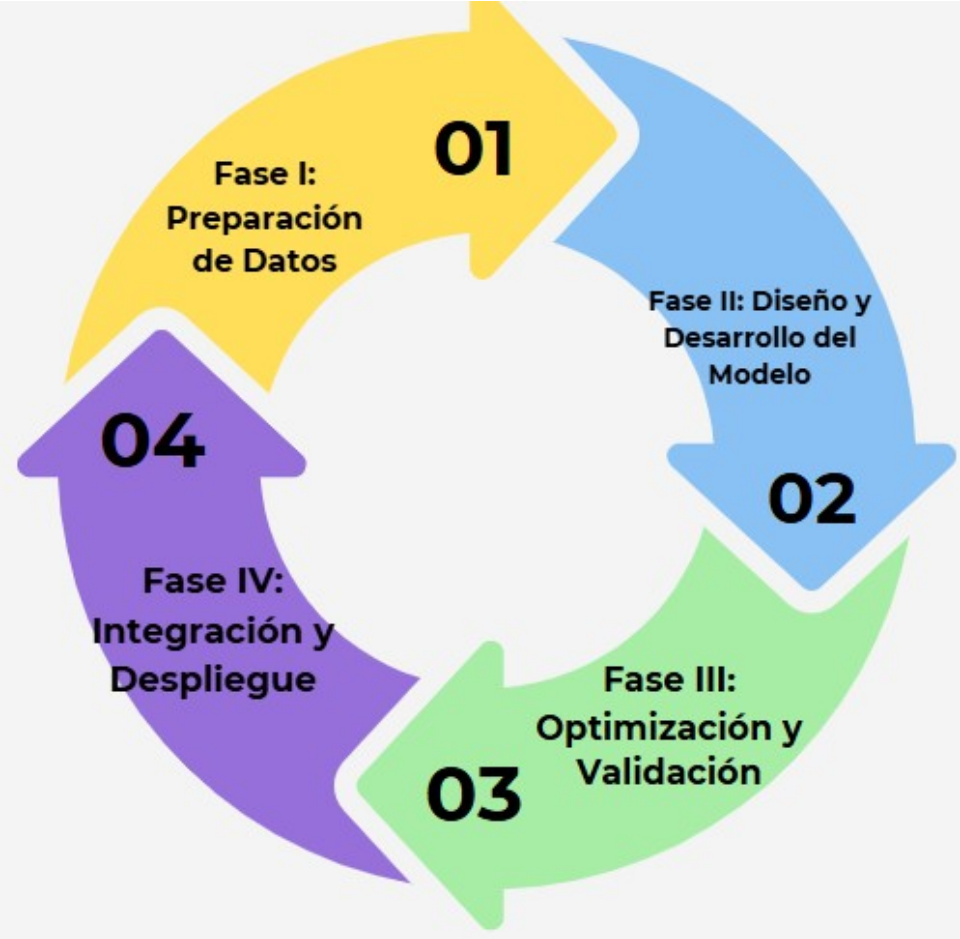


Figura 46: Las fases del desarrollo de traductor de señas basado en Machine Learning.

V-A. Fase I: Preparación de Datos

V-A1. Definición de las 5 frases LSEC: Se seleccionaron cinco frases escenciales de saludo y presentación, las frases elegidas se consultaron en los diccionarios y materiales de referencia de la FENASEC (Federación Nacional de Sordos del Ecuador). Cada una con características gestuales únicas, con el objetivo de evaluar el desempeño del modelo de manera integral.

En la Tabla IV se describen las características gestuales del vocabulario seleccionado.

Tabla IV: Diccionario de señas básicas y descripción de sus características gestuales.

Frase	Características de la Seña
Hola	Movimiento semicircular de la mano dominante cerca de la sien.
Gracias	Movimiento de la mano desde la barbilla hacia adelante y abajo.
Te quiero	Configuración específica de dedos (pulgar, índice y meñique extendidos).
Buenos Días	Coloca las yemas de los dedos juntas sobre la frente, luego desplazar la mano hacia adelante y hacia arriba mientras extiendes los dedos.
Por favor	Movimiento de palma abierta en el centro del pecho.

En la Figura 47, se visualizan los ejemplos de las cinco señas seleccionadas para el proyecto: “Buenos días”, “Gracias”, “Te quiero”, “Por favor” y “Hola”. Estas ilustraciones sirvieron como guía visual durante la fase de recolección de datos, asegurando que las muestras grabadas coincidieran con los estándares establecidos por la FENASEC. La correcta ejecución de estos gestos fue la base para que el modelo lograra identificar las señas con precisión en las etapas posteriores.

Figura 47: Ilustración de las 5 señas.

V-A2. Optimización de la Adquisición de Datos mediante Captura Directa: Originalmente, el proceso de generación de un dataset para señas implica grabar videos independientes, transferirlos a un ordenador y procesarlos mediante scripts de terceros para extraer cada fotograma. Al principio, el proceso tradicional era demasiado lento, ya que procesar cada video tomaba entre 10 y 15 minutos, algo inviable para la cantidad de datos que necesita una red LSTM.

Se optó por trabajar directamente en PyCharm con un algoritmo de captura en tiempo real. Como se aprecia en la Figura 48, este código automatiza todo: crea las carpetas para cada seña y graba secuencias de 60 cuadros solo cuando detecta que las manos están frente a la cámara.

De esta forma, se evitó perder tiempo con videos vacíos y se garantizó que cada archivo fuera útil para el entrenamiento."

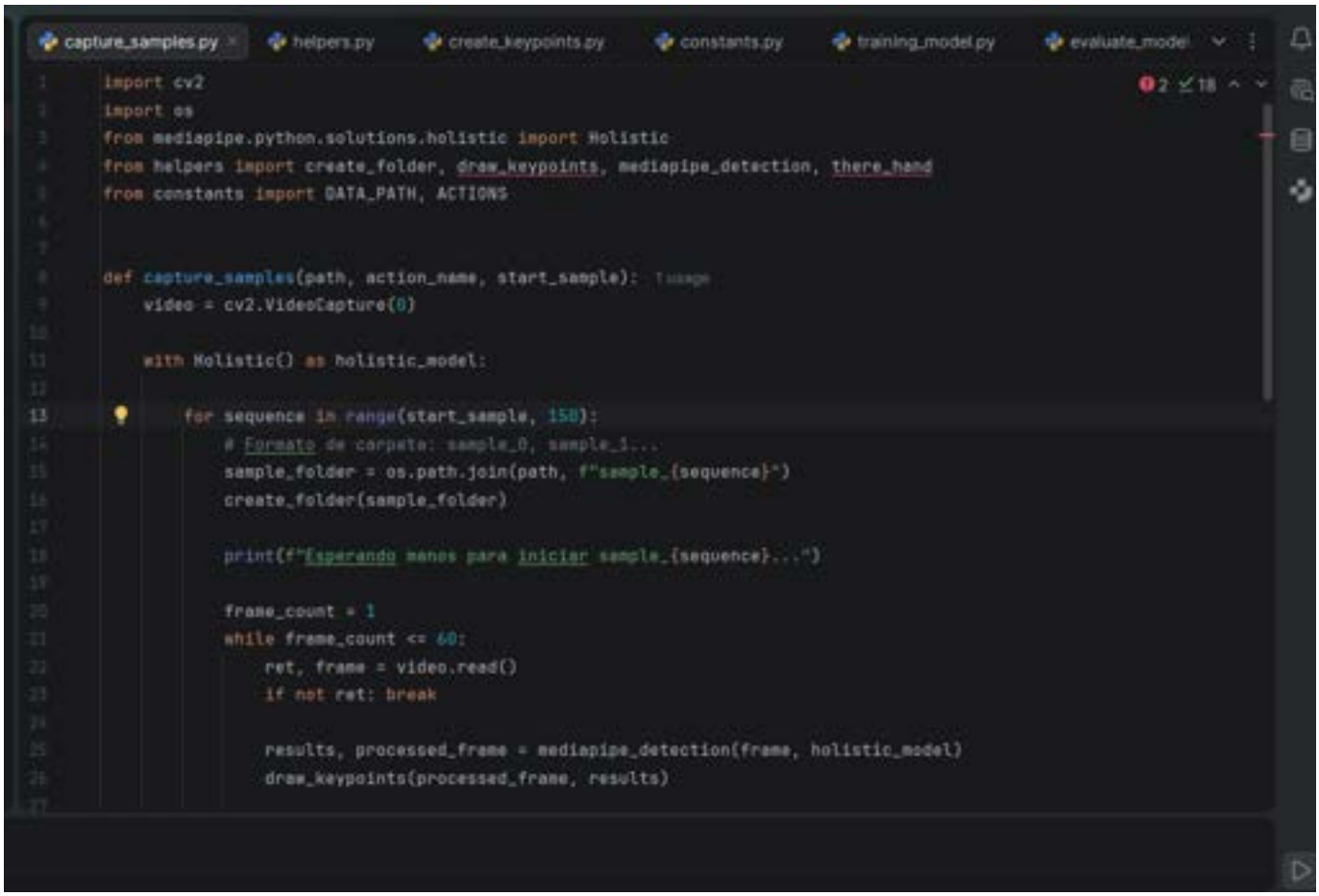


Figura 48: Algoritmo de captura de datos en tiempo real.

- Procesamiento Instantáneo: En lugar de grabar y luego convertir, el programa captura el fotograma de la webcam y lo segmenta automáticamente en una secuencia de 60 imágenes (.jpg) dentro de su carpeta correspondiente en milisegundos.
- Filtro de Calidad por Presencia: Mediante la lógica de detección de MediaPipe, el sistema solo inicia la captura cuando identifica los landmarks de la mano. Esto elimina la necesidad de editar videos manualmente para quitar los segundos de "silencio gestual" al inicio o al final de cada grabación.
- Automatización de Categorías: El programa solicita al usuario la frase o acción a grabar y genera automáticamente la estructura de directorios necesaria (sample_0, sample_1, etc.). Esto asegura una organización inmediata del dataset, evitando errores humanos de etiquetado manual.

V-B. Fase II: Diseño y Desarrollo del Modelo

Para el desarrollo y entrenamiento del modelo SMAD, se utilizó una estación de trabajo con las siguientes características, necesarias para procesar los arreglos multidimensionales de MediaPipe:

- Procesador: Intel Core i7 .
- Memoria RAM: 16GB, necesaria para cargar el dataset de secuencias de video en memoria.
- Unidad de Procesamiento Gráfico (GPU): NVIDIA con 8GB de VRAM.
- La presencia de una GPU dedicada de 8GB permitió el uso de la arquitectura CUDA de NVIDIA. Esto es fundamental para acelerar el cálculo de tensores en $TensorFlow$. Mientras que un procesador normal (CPU) procesa los datos de forma secuencial, la GPU permitió procesar en paralelo las matrices de coordenadas de los 543 landmarks de MediaPipe, reduciendo el tiempo de entrenamiento de horas a solo minutos.
- Periférico de entrada: Cámara web de alta definición (720p/1080p) con capacidad de captura a 30 FPS, garantizando que el flujo de video coincida con la frecuencia de muestreo de la red neuronal.
- Python 3.10: Fue el lenguaje principal elegido para el proyecto. Se utilizó porque es el que mejor integra las librerías de inteligencia artificial y permite conectar la cámara con el modelo de señas para que funcionen al mismo tiempo sin retrasos.
- TensorFlow y Keras: Se utilizó estas librerías para construir la red neuronal. Keras facilitó la creación del modelo capa por capa de forma secuencial. Gracias a esto, se logró que el programa aprenda a reconocer los gestos comparando la información de las carpetas que grabamos para el entrenamiento.
- Archivos .h5: Al finalizar el entrenamiento, se guarda los resultados en un archivo llamado `model.h5`. Este archivo contiene todo el conocimiento que el modelo adquirió. Se usa este formato para que el sistema SMAD sea ligero; así, el programa solo carga este archivo y empieza a traducir al instante sin tener que procesar todo desde cero cada vez, lo que ahorra recursos en la computadora.

En la Figura 49, se detalla la organización del directorio del proyecto, donde destaca la generación del archivo con extensión .h5. Este elemento representa el modelo final ya entrenado y compilado. Se optó por este formato debido a que permite almacenar tanto la arquitectura de la red como los pesos obtenidos, facilitando que el sistema SMAD realice la carga de datos de forma directa. Al emplear este archivo, se evita que el hardware deba procesar nuevamente el aprendizaje, logrando que la fase de ejecución sea ligera y el tiempo de respuesta del traductor sea casi instantáneo.

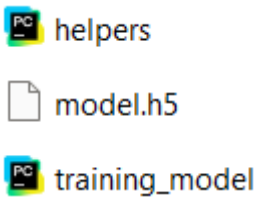


Figura 49: Archivo de persistencia del modelo en formato HDF5 (.h5).

Se diseña un modelo de Red Neuronal Convolucional (CNN), debido a que es adecuada para el reconocimiento de patrones visuales, tras el desarrollo se implementa en el lenguaje Python, utilizando la librería $TensorFlow$ para el pre-procesamiento, específicamente en la identificación de los puntos articulares de las manos en cada cuadro de video. Estos puntos de la mano son la característica principal que la CNN aprende a clasificar.

En la Figura 50, se muestra cómo funciona la red neuronal por dentro. Primero, el sistema recibe la foto de la mano en este caso es la imagen de entrada y le pasa unos filtros para resaltar las partes más importantes, como los dedos o la forma de la palma. Después, esa información se va haciendo más pequeña y específica hasta llegar a la parte final, donde el programa compara esos datos con lo que aprendió en el entrenamiento. Al final, el sistema decide a qué seña de la lengua de señas se parece más y entrega el resultado final.

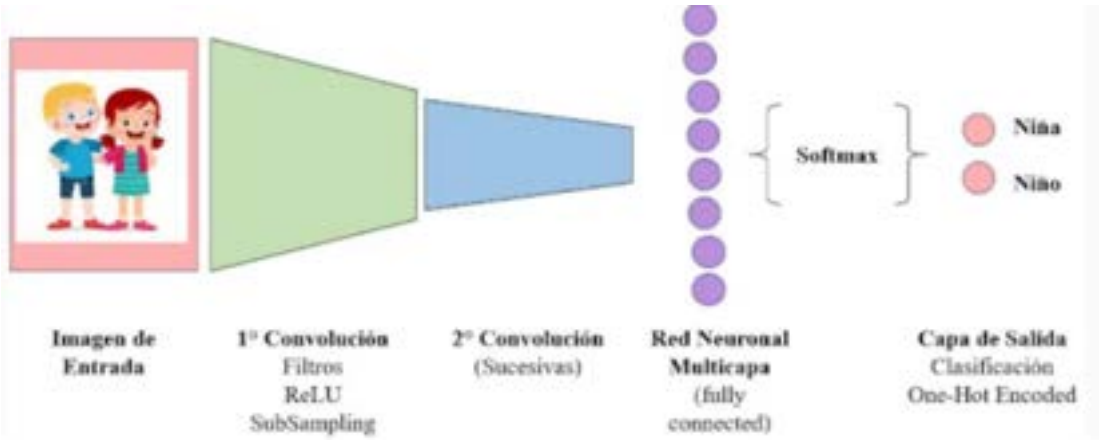


Figura 50: Estructura de la Red Neuronal Convolucional (CNN)

V-C. Fase III: Optimización y Validación

La fase III se centra en los ajustes para mejorar el modelo, el objetivo principal es mejorar la precisión del sistema para condiciones de uso real. Se optimiza los hiperparámetros del modelo para evitar el sobreajuste.

Esta fase representa la etapa crítica de refinamiento del sistema. Una vez estructurado el dataset de 150 muestras y extraídos los landmarks, el enfoque se desplaza hacia la configuración del entorno de aprendizaje profundo y la validación de la capacidad de respuesta del modelo.

Para evitar el procesamiento innecesario y el desgaste de recursos, se integró la función $(EarlyStopping)$ de la librería Keras. Esta técnica actúa como un regulador inteligente que monitorea la métrica de pérdida $(loss)$; si el modelo no presenta una mejora significativa tras un periodo de "paciencia" de 15 épocas, el proceso se interrumpe

automáticamente. Esto asegura que el modelo final sea la versión con el menor error de generalización posible.

V-C1. Análisis del Comportamiento de la Curva de Aprendizaje: El núcleo de esta fase es el análisis de la Gráfica de Precisión generada durante las 100 épocas programadas. El comportamiento observado en la gráfica permite deducir las siguientes conclusiones técnicas:

1. **Fase de Aprendizaje Acelerado:** Durante las primeras 15 a 20 épocas, el modelo presenta una pendiente pronunciada, lo que indica que la red LSTM está identificando correctamente los patrones fundamentales de la Lengua de Señas Ecuatoriana (LSE) a partir de los 60 fotogramas proporcionados.
2. **Zona de Inestabilidad y Saturación:**A partir de la época 20, la curva de aprendizaje muestra fluctuaciones o "zigzagueos". En términos de visión artificial, esto significa que el modelo ha alcanzado su límite de comprensión de los datos generales y ha comenzado a intentar ajustar sus pesos a las pequeñas variaciones o ruidos de las grabaciones.
3. **Prevención de Sobreajuste (Overfitting):** El sobreajuste es el riesgo de que la IA memorice las 150 carpetas en lugar de entender el movimiento. Al identificar que la precisión de validación se estanca mientras la de entrenamiento sigue subiendo, se valida la decisión de detener el proceso, garantizando que el sistema sea capaz de reconocer señas de nuevos usuarios que no participaron en la toma de muestras original.

V-C2. Evaluación de Métricas de Rendimiento: Loss vs. Accuracy: Para validar la efectividad del entrenamiento, se analizaron simultáneamente las métricas de Precisión (Accuracy) y Pérdida (Loss) a lo largo de las 100 épocas programadas. Esta dualidad es la que permite confirmar si el modelo es confiable o si simplemente está "adivinando" las señas.

- La Función de Pérdida (Loss): En la Figura 51, se presenta el registro que aparece en la terminal durante el entrenamiento del modelo. En la captura se observa el avance de la época 86, donde el sistema muestra el tiempo de procesamiento por cada paso y el valor del loss (pérdida).
  Este monitoreo es importante porque permite ver cómo el programa va aprendiendo de los datos; un valor de pérdida más bajo indica que el modelo está cometiendo menos errores al identificar las señas. Además, se registra el accuracy inicial de esa etapa, sirviendo como guía para controlar que el entrenamiento avance correctamente hacia el archivo final.

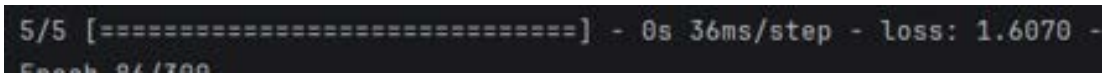


Figura 51: Se muestran los valores de pérdida (loss) y el tiempo de respuesta por cada iteración.

- La Precisión (Accuracy) Se aplicó la métrica de Accuracy para medir el nivel de acierto del sistema. Esta permite evaluar la precisión con la que el modelo clasifica las secuencias de 60 fotogramas dentro de las categorías de la Lengua de Señas Ecuatoriana, validando la efectividad del reconocimiento en cada prueba realizada.
  En la Figura 52, se observa el progreso del entrenamiento del modelo, donde se detallan métricas como el accuracy y el tiempo de respuesta por cada paso de la época actual. Este monitoreo permite verificar cómo el sistema ajusta sus parámetros internos para mejorar el reconocimiento de los 60 fotogramas que componen cada seña de la Lengua de Señas Ecuatoriana.

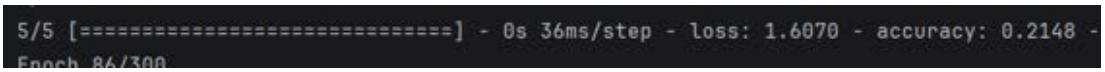


Figura 52: Progreso del accuracy y el tiempo de procesamiento por cada paso de la época actual.

En la consola de PyCharm se puede verificar cómo va subiendo el accuracy en cada iteración. Esto sirve para confirmar que el modelo está aprendiendo de las 150 muestras de forma estable. También se monitoreó el tiempo de procesamiento $(ms/step)$ para asegurar que el entrenamiento sea fluido y que el hardware soporte la carga de datos sin errores de memoria.

V-C3. Relación Dinámica con las Épocas: Las Épocas actúan como el eje temporal del aprendizaje. El análisis conjunto revela que:

- **Convergencia:** El punto donde el Loss deja de bajar y el Accuracy deja de subir se conoce como convergencia.
- **Estabilidad:** En las gráficas de entrenamiento se observó una estabilización del modelo a partir de la época 20, punto en el cual el fenómeno de zigzagueo en el accuracy disminuyó a pesar de que el loss se mantuvo en niveles bajos.

  Como se aprecia en la Figura 53, la precisión lograda en el seguimiento de puntos clave en cejas y boca fue determinante para esta estabilidad. Debido a este comportamiento, se determinó que la época 20 era el momento óptimo para la exportación definitiva del archivo del modelo, asegurando un equilibrio entre el aprendizaje y la capacidad de predicción.

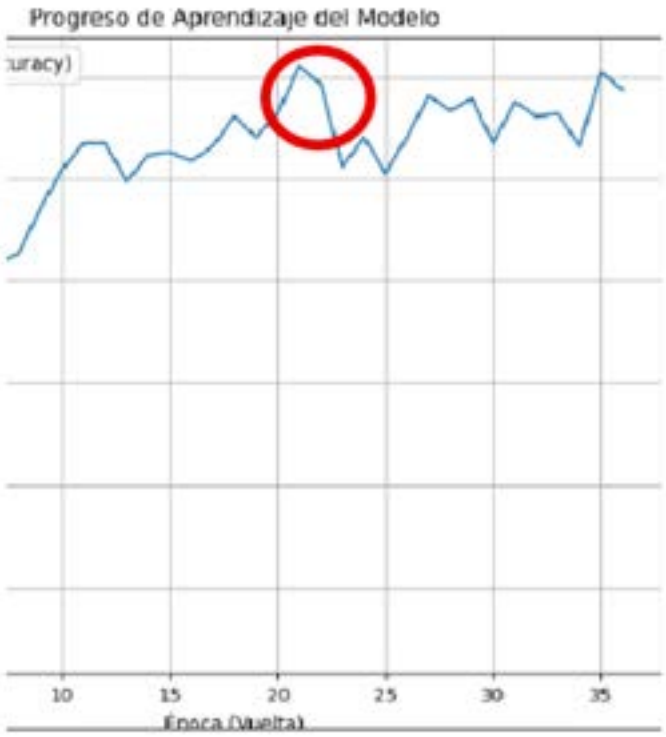


Figura 53: Comportamiento del entrenamiento del sistema SMAD.

- **Validación de la Arquitectura Final:** Finalmente, la optimización concluye con la exportación de los pesos en formato $(.h5)$. Este archivo comprimido contiene la jerarquía de neuronas ya entrenada, permitiendo que

la aplicación final cargue el conocimiento de manera instantánea sin necesidad de volver a procesar el dataset. La validación se considera exitosa al obtener una convergencia donde el error es mínimo y la precisión en tiempo real es estable, permitiendo el paso a la implementación de la interfaz de usuario.

V-D. Fase IV: Integración y Despliegue

Para la ejecución del proyecto, se integró el modelo en una interfaz de entorno web que permite el procesamiento de datos en tiempo real sin requerir instalaciones locales. El funcionamiento lógico del sistema SMAD se describe en el diagrama de flujo de la Figura 54. En este gráfico se detalla el ciclo de procesamiento: desde la activación de la cámara y la detección de puntos de referencia en las manos, hasta la clasificación de la seña mediante el modelo cargado. Finalmente, el sistema traduce la información gestual y despliega el texto correspondiente en la interfaz de usuario.

Con el objetivo de detallar el funcionamiento interno del traductor SMAD, elaboramos el diagrama de flujo presentado en la Figura 54. Este gráfico describe el ciclo de vida de una consulta en el sistema:

- **Captura y Preprocesamiento:** El proceso inicia con el encendido de la cámara, la cual captura el video a una tasa constante de cuadros.
- **Detección de puntos clave:** El sistema utiliza MediaPipe para buscar y rastrear el movimiento de las manos en cada fotograma. Si no se detecta presencia de extremidades, el programa continúa en un bucle de espera para optimizar el procesamiento.
- **Clasificación Temporal:** Una vez detectada la secuencia de movimientos, los datos se envían al modelo $(.h5)$. Aquí es donde la red neuronal analiza la trayectoria completa para clasificar la seña.
- **Salida de información:** Finalmente, cuando el modelo alcanza un umbral de confianza alto, el resultado se traduce a texto y se despliega en la interfaz gráfica para el usuario.

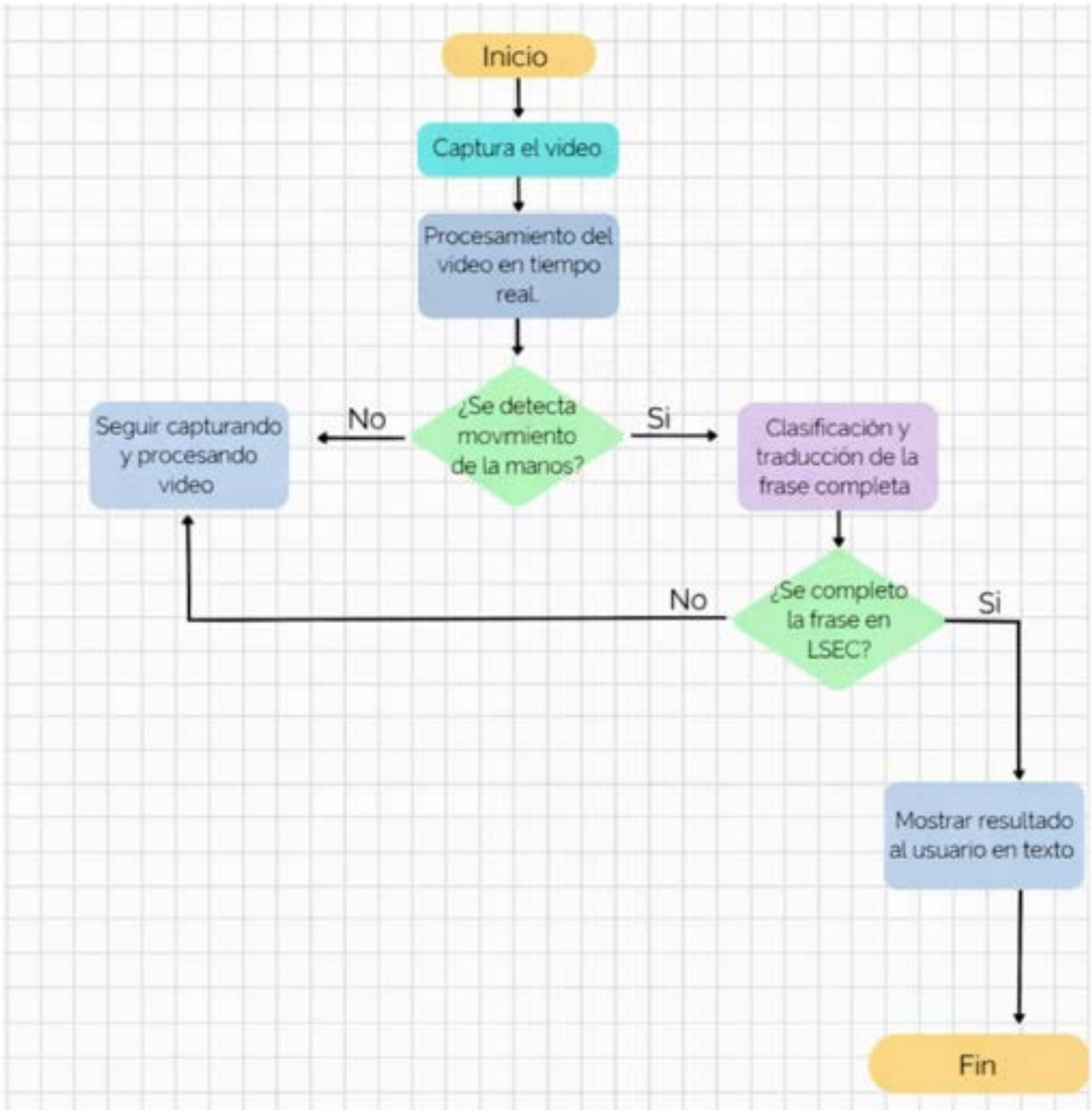


Figura 54: Diagrama de flujo del funcionamiento lógico del sistema SMAD en tiempo real.

V-E. Arquitectura de Implementación y Captura de Datos

V-E1. Desarrollo del Frontend: Interfaz de Usuario y Comunicación: La interfaz de usuario se desarrolló mediante un enfoque de Single Page Application (SPA), priorizando la velocidad de carga y la facilidad de uso para personas con discapacidad auditiva. El entorno de desarrollo principal fue el editor de Visual Studio Code, utilizando la extensión Live Server para administrar un servidor local que gestiona los protocolos de cámara y los scripts asíncronos en un entorno seguro.

La figura 55 muestra que el diseño prioriza la legibilidad y la simplicidad, permitiendo a las personas con discapacidad auditiva interactuar intuitivamente con el sistema de reconocimiento desde un navegador web estándar.

Figura 55: Interfaz gráfica de usuario del sistema SMAD operativa en entorno de navegador Chrome.

La estructura se divide en tres capas fundamentales:

- **Estructura (HTML5):** Define los contenedores para el flujo de video y los resultados de traducción. Se implementó una estructura organizada por secciones $(header, main, section, footer)$. Destaca el uso de la etiqueta "video" para el flujo en tiempo real y "canvas" para el pre-procesamiento de imágenes.
- **Estilos (CSS3):** Proporciona una interfaz limpia y responsiva, optimizada para dispositivos portátiles.
- **Lógica de Control (JavaScript):** Gestiona la captura de frames y la comunicación con el modelo de IA.

V-E2. Gestión de Eventos y Manipulación del DOM: Mediante JavaScript (Vanilla), se programó la lógica de control sin librerías externas para mantener el sistema ligero:

- **Control de Cámara:** Función asíncrona que solicita permisos de hardware y vincula el flujo de video al elemento de la página.
- **Actualización Dinámica (DOM):** Permite que, cuando el backend identifica una palabra (ej. "HOLA"), el texto cambie instantáneamente en el elemento "span" sin refrescar la página.

Se aprecia en la Figura 56 el entorno de Visual Studio Code, utilizado para la estructuración del frontend y la configuración del servidor local necesario para el manejo de los protocolos de cámara.

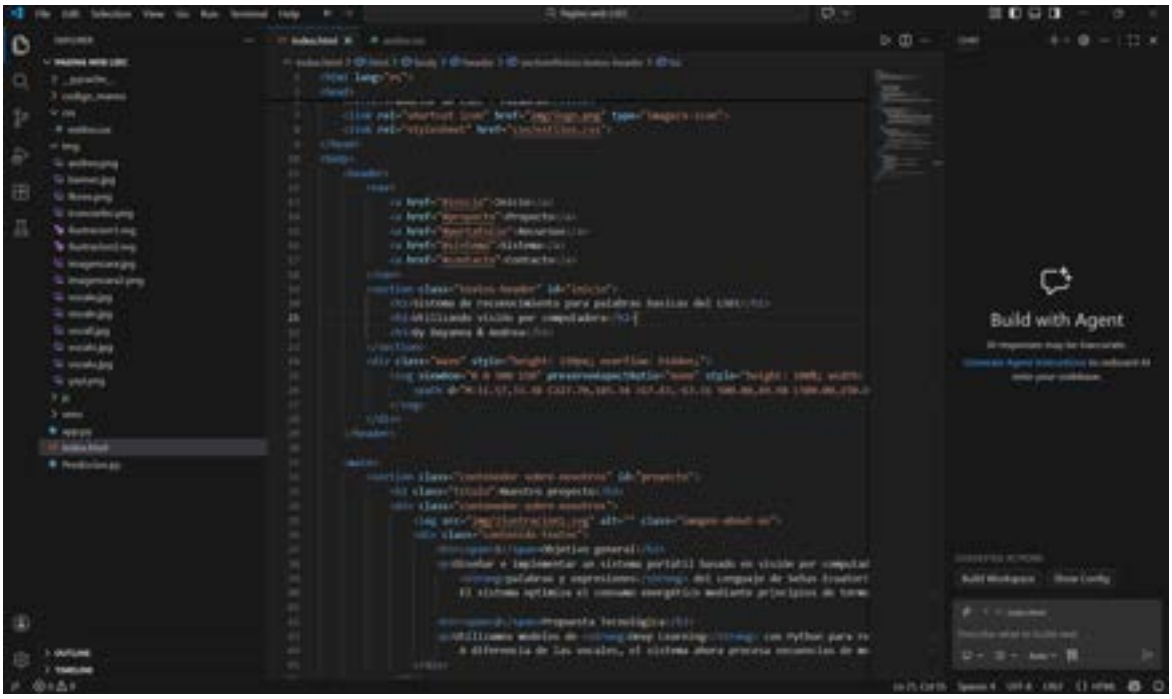


Figura 56: Entorno de programación es el software Visual Studio Code.

V-E3. Usabilidad y Accesibilidad: El diseño se centró en la simplicidad, eliminando distractores. El contraste cromático y el tamaño de fuente para la "Traducción" se calibraron para ser legibles a una distancia de hasta 2 metros, facilitando la comunicación del usuario frente a la cámara.

V-E4. Integración de Visión Artificial y Backend: Dado que el motor de reconocimiento de LSEC (Lenguaje de Señas Ecuatoriano) reside en un entorno de Python (PyCharm), la conexión se resolvió mediante una arquitectura Cliente-Servidor utilizando una API REST con el micro-framework Flask.

El proceso de comunicación sigue este flujo técnico:

- **Captura:** JavaScript accede a la webcam mediante la API getUserMedia y proyecta el video en la pantalla.
- **Transmisión:** Al activar el reconocimiento, se extraen fotogramas del video, se convierten a formato Base64 y se envían mediante una petición POST (Fetch API) hacia el servidor local en la dirección http://localhost:5000.
- **Procesamiento:** El script en Python recibe la imagen, la procesa a través del modelo de aprendizaje profundo (Deep Learning) y determina la palabra correspondiente.
- **Repuesta:** El servidor devuelve un objeto JSON con la palabra detectada, la cual se refleja instantáneamente en la interfaz sin recargar la página.

En la Figura 57 se ilustra el flujo de la adquisición de datos mediante la cámara, la gestión de la interfaz de usuario en JavaScript, el procesamiento y modelo de IA en el servidor Python, y la salida final de la traducción.

Figura 57: Diagrama de bloques del sistema de traducción.

V-E5. Diagrama de Flujo del Sistema Integrado: Para garantizar que el sistema SMAD sea escalable y fácil de mantener, la lógica de programación se dividió en módulos independientes. Esta estructura es la que permite que el flujo desde la cámara hasta la carpeta de datos sea eficiente.

V-E6. Definición de Parámetros Globales constants.py: En este archivo se centralizaron las variables que definen la dimensión del proyecto. Esto evita errores de inconsistencia durante las fases de captura y entrenamiento:

- Gestión de Clases: Se definen las etiquetas de la LSEC como “Hola”, “Gracias”, “Te quiero”, “Por favor”, “Buenos días”.

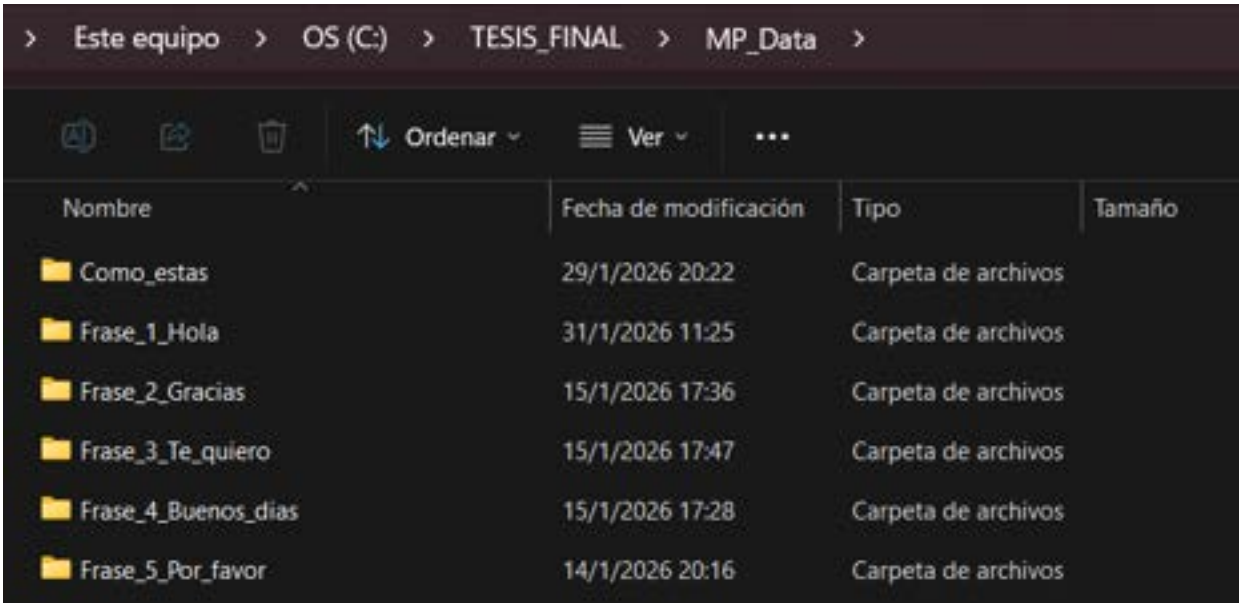


Figura 58: Ilustración de las carpetas para las 5 frases.

En la Figura 58, se observa la organización del directorio principal donde se almacenan las muestras del proyecto. Se detalla la creación de carpetas específicas para cada etiqueta de la Lengua de Señas Ecuatoriana (LSEC), como “Hola”, “Graciasz “Buenos días”. Esta estructura de archivos permitió que el script de entrenamiento cargara de

forma ordenada las secuencias de datos capturadas previamente, asegurando que cada gesto estuviera correctamente clasificado antes de alimentar a la red neuronal.

- Dimensionamiento Temporal: Se establece la constante de 60 frames por secuencia para mantener la uniformidad en las 150 carpetas de entrenamiento.

V-E7. Procesamiento y Extracción de Atributos create_keypoints.py y helpers.py: Estos módulos trabajan en conjunto para transformar la imagen de la cámara en datos matemáticos:

- (create_keypoints.py): Este script es el responsable de ejecutar la detección de MediaPipe sobre cada frame capturado. Su función principal es extraer las coordenadas exactas de las manos y el rostro (enfoque en cejas y boca).

  Para la extracción de características, implementamos la función get_keypoints(). Esta línea es el núcleo del procesamiento, ya que invoca el modelo Holistic de MediaPipe para analizar cada fotograma guardado en el dataset. Su función es mapear los puntos clave de la gesticulación facial (cejas y boca) y la posición de las manos, transformando la información visual en vectores numéricos necesarios para el entrenamiento de la red neuronal.
- Helperspy: Contiene las funciones de apoyo que normalizan estos puntos. Si un punto no es detectado, el helper asegura que el sistema no falle, devolviendo un arreglo de ceros para mantener la integridad de la matriz de entrada. Para optimizar el código, dividimos el desarrollo en módulos en el archivo $helpers.py$ definimos la función $extract\_keypoints$, la cual es la encargada de extraer y concatenar los puntos clave de la cara, manos y pose. Posteriormente, esta función es invocada por el script principal para procesar secuencialmente todas las muestras de video recolectadas, transformando las imágenes en archivos de datos listos para el entrenamiento.

En la Figura 59, se define la función $extract\_keypoints$, la cual transforma los datos brutos de la cámara en información útil para el modelo. El código extrae las posiciones de la cara, el cuerpo y las manos en coordenadas (x, y, z). Uno de los ajustes importantes fue configurar el código para que no se se cerrara cuando la cámara no detectaba alguna parte del usuario. Para solucionar esto, se programó una regla simple: si falta una mano o un gesto, el sistema llena ese espacio con ceros. Se realizó de esta manera para que los datos tengan siempre el mismo tamaño y el modelo no se confunda ni se detenga por falta de información en mitad de una prueba."

Figura 59: Segmento del archivo helpers.py.

Como se observa en la Tabla V, el proyecto SMAD procesó un total de 750 secuencias de video. En el desarrollo del proyecto, se determinó ampliar la ventana temporal a 60 fotogramas con el objetivo de capturar con mayor fluidez la ejecución de las señas de larga duración. Esta configuración generó un flujo de procesamiento superior a los 74 millones de puntos de datos, factor que fundamenta la alta precisión obtenida en la matriz de confusión. Asimismo, este volumen de información justifica el tiempo de cómputo requerido durante el entrenamiento de la red LSTM, asegurando que el modelo logre identificar patrones complejos en el movimiento de las manos.

Tabla V: Dimensiones del Dataset SMAD: 150 muestras por cada categoría.

Frase LSEC	Secuencias (Carpetas)	Frames por Video	Datos Totales
Hola	150	60	14,958,000
Gracias	150	60	14,958,000
Te quiero	150	60	14,958,000
Buenos días	150	60	14,958,000
Por favor	150	60	14,958,000
TOTALES	**750 Vídeos**	**45,000 Cuadros**	**74,790,000**

- **Persistencia y Almacenamiento de Datos:** Los datos procesados se almacenan en el directorio MP_Data. Cada subcarpeta representa una frase específica de la LSEC, conteniendo los archivos binarios que la red LSTM utilizará posteriormente.

V-F. Módulo de Extracción de Características (create_keypoints.py)

Este script es el que hace el trabajo pesado de "leer" nuestras señas. Lo que hace es transformar los videos en datos que la computadora entiende.

- **Organización por carpetas:** El código busca la carpeta donde guardamos los videos y entra a cada una de las frases (ACTIONS). Como grabamos 60 muestras por cada seña, el código las procesa una por una de forma automática.
- **Extracción con MediaPipe:** Mientras el código corre, llama a la función $get\_keypoints$. Esta función usa MediaPipe para dibujar los puntos en la cara, manos y cuerpo en cada frame.

En estas capturas se presenta el flujo de trabajo para la generación del dataset. En la Figura 60, muestra la lógica de programación basada en bucles para el acceso automatizado a los directorios de cada seña. Complementariamente, en la Figura 61, se aprecia la ejecución de dicho código, donde se valida la extracción de los puntos clave y su almacenamiento en archivos de $.npy$. Este procedimiento garantiza que la información capturada por MediaPipe se organice de manera uniforme, facilitando el posterior entrenamiento de la red neuronal con datos previamente normalizados.

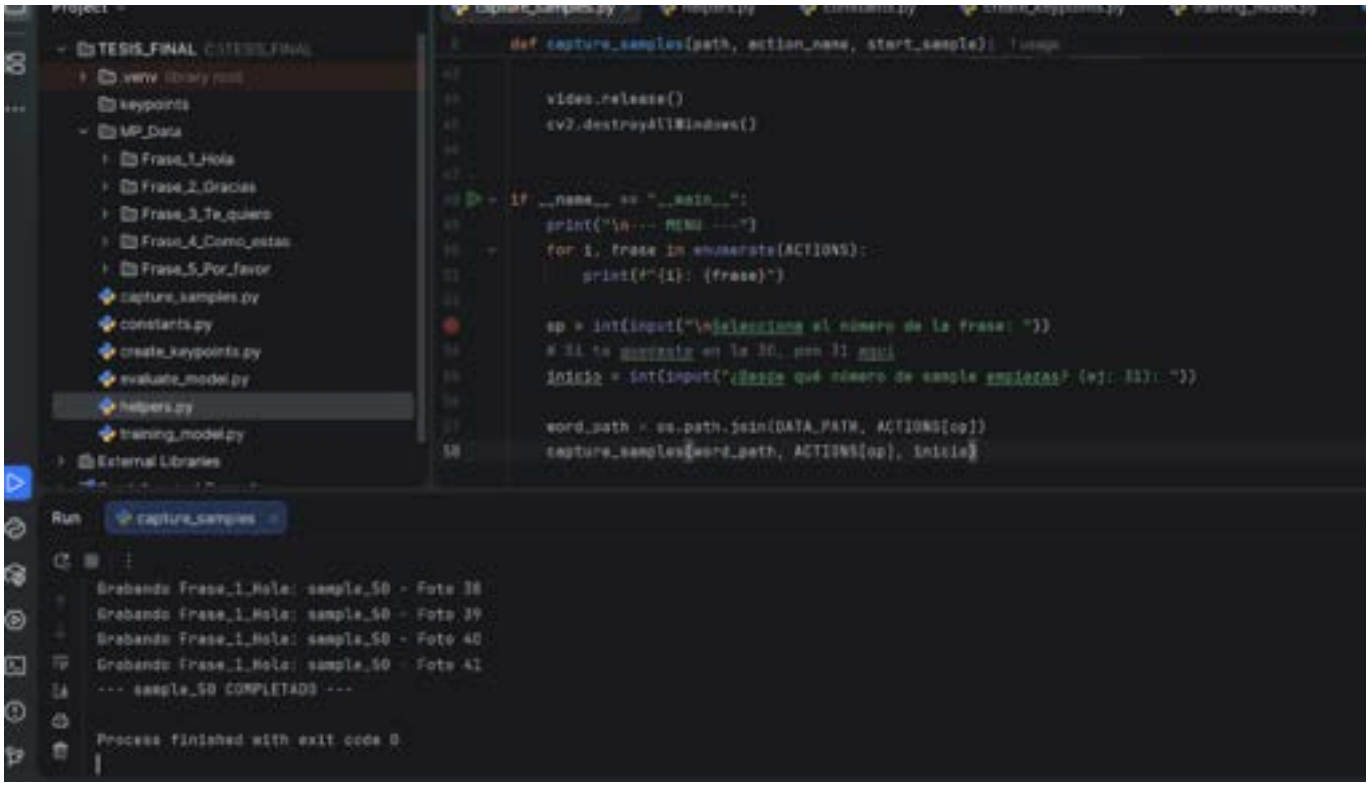


Figura 60: Fragmento del script run_extraction donde se observa el bucle que recorre las 60 muestras.

```
def run_extraction():  1 usage
    # Crea la carpeta donde se guardará el "estudio" de tus señas
    create_folder('keypoints')

    with Holistic() as holistic:
        for action in ACTIONS:
            print(f'Procesando frase: "{action}"...')
            action_path = os.path.join(DATA_PATH, action)
            data = pd.DataFrame([])

            # Busca tus 60 carpetas
            samples = [s for s in os.listdir(action_path) if s.startswith('sample_')]
```

Figura 61: Interfaz de comando mostrando el progreso de extracción de puntos en tiempo real.

V-F1. Gestión de Datos con Pandas y Persistencia en CSV: Para el almacenamiento de la información extraída, se tomó la decisión técnica de utilizar la librería Pandas y el formato de archivos CSV (Comma Separated Values). Esta elección no fue al azar, sino que respondió a tres necesidades fundamentales del proyecto:

- Verificación Humana de los Datos (Auditoría)

Al guardar los puntos clave en archivos .csv, pudimos abrir la información directamente en Microsoft Excel. Esto

permitió realizar una inspección visual de las coordenadas X, Y, Z. Si un archivo pesara 0 KB o tuviera celdas vacías, se conocería de inmediato que esa muestra de seña salió mal y se tendría que repetirla antes de intentar entrenar a la inteligencia artificial.

- Análisis de peso: Por ejemplo, el archivo Frase_1_Hola.csv alcanzó un tamaño de 114,118 KB. Este volumen de datos es el resultado de multiplicar.

En la Figura 62, se presenta el almacenamiento de los datos procesados en archivos de formato .csv. Se detalla que cada documento corresponde a una clase específica de la LSEC, registrándose pesos individuales que superan los 100 MB. Este volumen de almacenamiento refleja la cantidad masiva de coordenadas capturadas, donde la organización por etiquetas facilita la lectura de los puntos clave durante la etapa de entrenamiento del modelo.

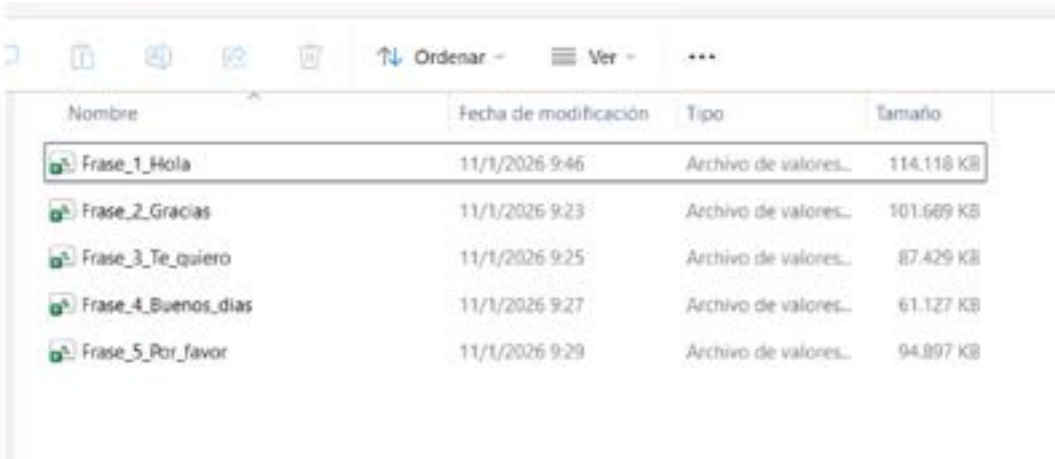


Figura 62: Archivos de datos (.csv) generados tras el proceso de extracción.

- Los 150 videos grabados.

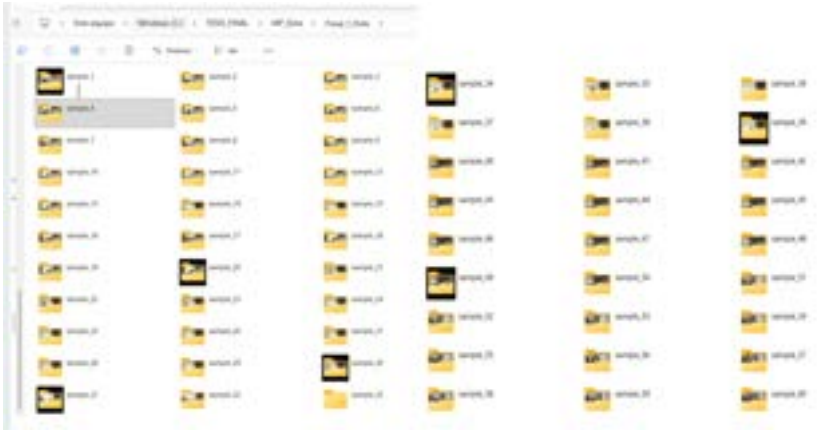


Figura 63: Estructuración del conjunto de datos (dataset) organizado por categorías

En la Figura 63, se observa la organización interna del dataset para una categoría específica. Se detalla la división en subcarpetas numeradas, donde cada una almacena una secuencia independiente de fotogramas procesados. Cada carpeta almacena las secuencias de 60 fotogramas que componen la dinámica temporal de una seña específica.

Los frames por cada video de estas ilustraciones se presenta la descomposición de las tomas de video en fotogramas independientes para su análisis individual. La Figura 64, muestra la etapa inicial de un gesto técnico, donde se capturan las posiciones de las manos en cuadros fijos. Por su parte, la Figura 65, expone la continuación de dicha secuencia, evidenciando la transición del movimiento a lo largo del tiempo. Esta fragmentación del video en imágenes estáticas permite que el sistema procese la trayectoria de la seña cuadro por cuadro, facilitando la posterior extracción de coordenadas espaciales que conforman el dataset del proyecto.

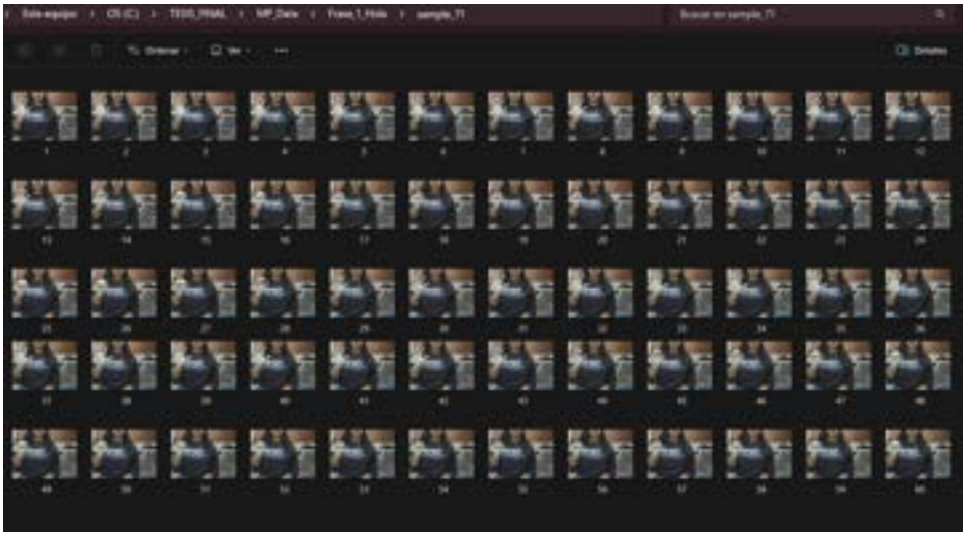


Figura 64: Estructuración de la secuencia temporal para la frase “Hola”.

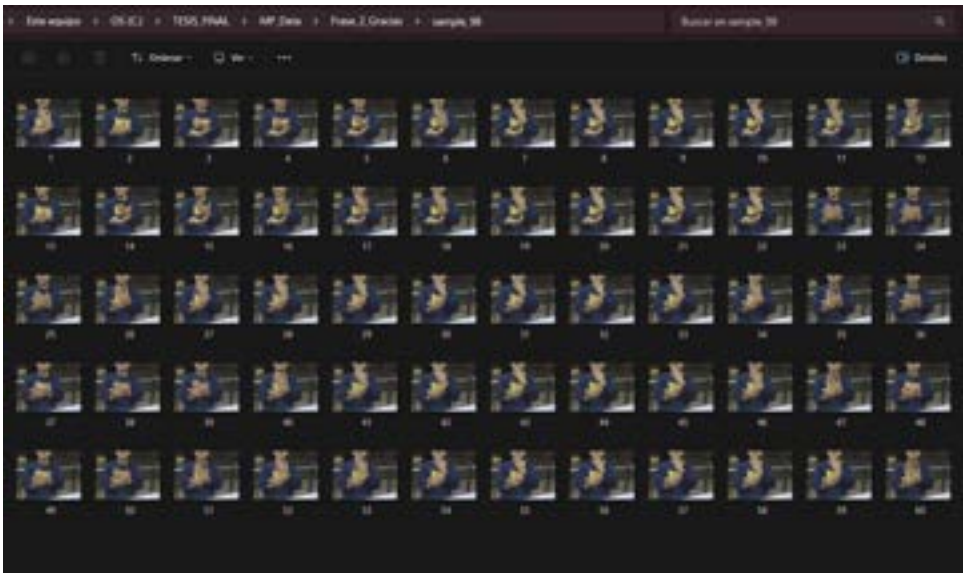


Figura 65: Estructuración de la secuencia temporal para la frase “Como_estas”.

- Los cientos de puntos (landmarks) que MediaPipe detecta en cada frame.

En esta captura se demuestra la implementación de la biblioteca MediaPipe Holistic para la detección de puntos clave $landmarks$. Como se observa en la Figura 66, el trazado de la malla sobre el rostro, manos y pose, lo cual permite la extracción de coordenadas espaciales en tiempo real.

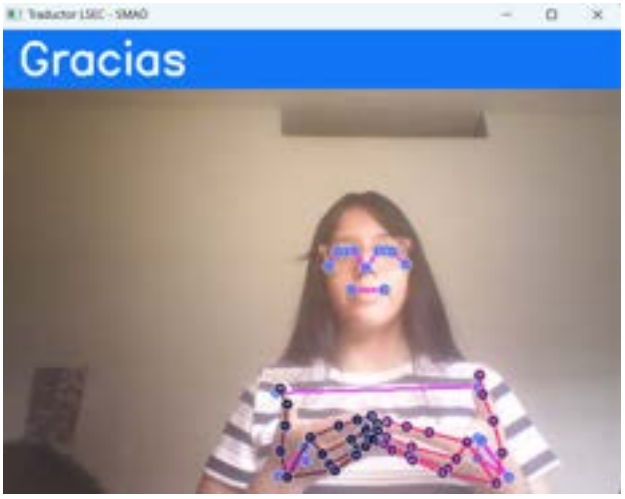


Figura 66: Proceso de extracción de puntos clave (landmarks) mediante MediaPipe Holistic.

V-G. Implementación del Sistema de Adquisición de Datos

El primer paso de la investigación consistió en el diseño y ejecución de un entorno controlado para la captura de video en tiempo real, orientado a la creación de un dataset robusto para la Lengua de Señas Ecuatoriana (LSEC).

La adquisición del dataset constituye la primera etapa del desarrollo del sistema de reconocimiento y clasificación del Lenguaje de Señas Ecuatoriano (LSEC), ya que de la calidad y representatividad de los datos depende el correcto entrenamiento del modelo de aprendizaje profundo. En esta fase se recopilan muestras de video correspondientes a señas específicas, garantizando condiciones controladas que permitan una correcta extracción de características visuales.

V-G1. Estrategia de Adquisición del Dataset: Para garantizar que el modelo de Inteligencia Artificial sea capaz de generalizar y no solo memorizar, se aplicaron criterios de variabilidad controlada:

- **Población de muestra:** Participaron 7 personas diferentes, lo que permite al sistema aprender diversas morfologías de manos y longitudes de brazos.

En la TablaVI, se muestra el proceso donde se recolectaron entre 21 y 22 muestras por persona para que el sistema tenga una base variada, el modelo de aprendizaje automático cuente con una base variada, permitiendo que el sistema reconozca los gestos de la LSEC independientemente de las características físicas o la mano dominante del usuario.

Tabla VI: Distribución de la población de muestra y aportación al dataset (Total: 150 muestras).

Participante	Género	Lateralidad	Muestras
Dayanna	Femenino	Diestra	22
Andrea	Femenino	Diestra	22
Sandy	Femenino	Diestra	21
Gianmarco	Masculino	Diestro	21
Daniel	Masculino	Diestro	22
Jostin	Masculino	Zurdo	21
Job	Masculino	Diestro	21
Total General	**—**		**150**

- **Diversidad de Entornos:** Se realizaron capturas en fondos planos, abiertos, con iluminación natural y artificial para reducir el ruido visual.
- **Variabilidad de Usuario:** Se incluyeron cambios de vestimenta, peinado (cabello suelto/recogido) y accesorios (anillos, abrigos) para asegurar que el modelo se centre exclusivamente en los gestos.

En la Figura 67, se presentan las pruebas de generalización aplicadas al sistema SMAD. Se observa el uso de diversas muestras del dataset que incluyen variaciones controladas en el entorno. Esta metodología se implementa para asegurar que la red neuronal no aprenda características irrelevantes de la imagen, sino que se enfoque exclusivamente en la trayectoria de los gestos

Figura 67: Pruebas de generalización del sistema SMAD.

- **Uso de Contraste:** Se emplearon guantes blancos para optimizar la detección de puntos clave (keypoints) por parte de la librería MediaPipe, estabilizando el seguimiento de las falanges.

En la Tabla VII se resumen las dimensiones cuantitativas del Dataset SMAD, detallando la escala del procesamiento de datos realizado. Se especifica la recolección de 150 secuencias por cada categoría gestual, lo que suma un total de 750 vídeos grabados.

Tabla VII: Dimensiones del Dataset SMAD: 150 muestras por cada categoría.

Frase LSEC	Secuencias (Carpetas)	Frames por Video	Datos Totales
Hola	150	60	14,958,000
Gracias	150	60	14,958,000
Te quiero	150	60	14,958,000
Buenos días	150	60	14,958,000
Por favor	150	60	14,958,000
TOTALES	**750 Vídeos**	**45,000 Cuadros**	**74,790,000**

V-G2. Características Biométricas para el Entrenamiento: El sistema no solo captura imágenes, sino coordenadas tridimensionales en el espacio. El entrenamiento se basa en:

- **Posición de la Mano:** Seguimiento de 21 puntos clave por mano, analizando la flexión de dedos y orientación de la palma. Una vez capturados los 60 fotogramas mediante $OpenCV$, el siguiente paso crítico es la extracción de características morfológicas. Para esto, se utilizó el framework $MediaPipeHolistic$, el cual permite obtener una representación vectorial de la mano.En lugar de procesar la imagen completa (miles de píxeles), el sistema identifica 21 puntos clave (landmarks) tridimensionales por cada mano. Cada punto posee coordenadas (x, y, z), donde:(x, y) representan la posición horizontal y vertical en el fotograma. z representa la profundidad relativa respecto a la muñeca, permitiendo capturar el volumen de la seña.

En la Figura 68, se presenta el entorno operativo del sistema durante la fase de reconocimiento de señas. Se observa la superposición de la malla de puntos clave sobre la mano del usuario, proceso ejecutado mediante la biblioteca MediaPipe para la captura de coordenadas espaciales.

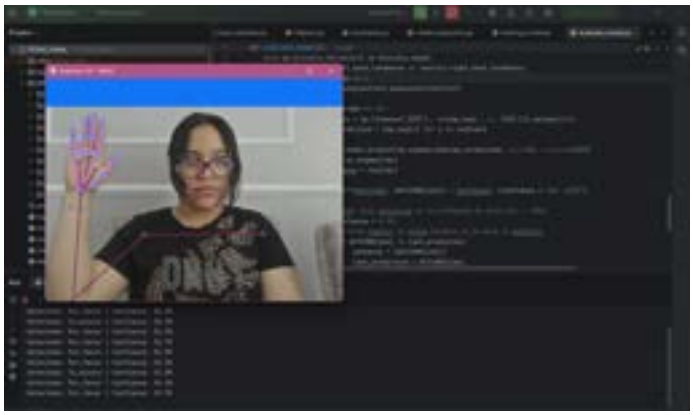


Figura 68: Representación de los 21 puntos clave (landmarks) extraídos mediante MediaPipe.

- **Cinemática del Brazo:** Coordenadas de codos y hombros para entender la amplitud del movimiento.
- **Torso y Postura:** Puntos de referencia del tronco que sirven como eje de coordenadas central, permitiendo al modelo entender la posición relativa de las manos respecto al cuerpo. Para que el sistema SMAD logre una interpretación precisa de la LSEC, cada muestra capturada se transformó en un vector estructurado de 1662 valores numéricos por cada fotograma (*frame*). Este número no es aleatorio; es el resultado de la extracción exhaustiva de puntos clave mediante MediaPipe Holistic, desglosados de la siguiente manera:
- **Malla Facial (Face Mesh):** 468 puntos que capturan la gesticulación, esencial para el contexto de las frases. La importancia de priorizar estos puntos (ver Figura 69) radica en que, en la LSEC, la posición de las

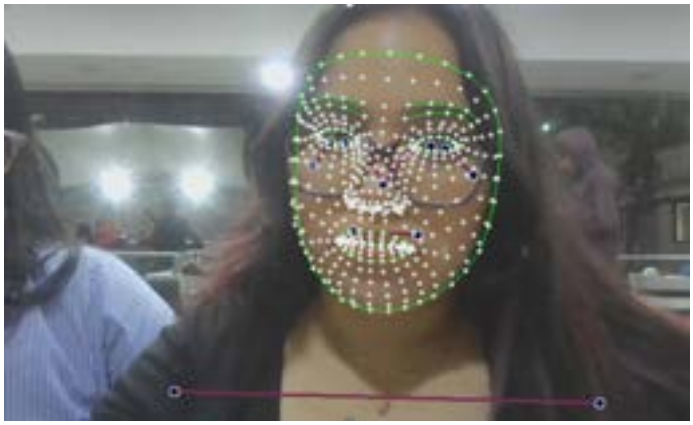


Figura 69: Visualización de la malla facial y puntos de pose durante la fase de captura.

cejas y el movimiento de los labios actúan como marcadores gramaticales. Al filtrar el ruido del resto de la cara y enfocarnos en estas áreas y en el torso, optimizamos la entrada de los vectores al modelo LSTM, permitiendo que la red se concentre en los cambios morfológicos que realmente definen cada frase, evitando así el sobreajuste con datos faciales irrelevantes.
- **Pose y Extremidades:** 33 puntos que definen la postura del torso y brazos.

- **Morfología de Manos:** 21 puntos por cada mano (izquierda y derecha) para el seguimiento detallado de los dedos.

Al multiplicar estos puntos por sus tres coordenadas espaciales (x, y, z), obtenemos la matriz de entrada que alimenta a la red LSTM. Este robusto volumen de datos justifica el tamaño de los archivos CSV almacenados en la carpeta `keypoints` y permite que la red neuronal aprenda no solo el movimiento bruto, sino la sutileza de cada seña.

V-G3. Arquitectura del Programa de Captura: La primera fase del software desarrollado se encarga de:

- **Captura de Video:** Flujo de entrada a través de una cámara web estándar a 30-60 FPS.
- **Extracción en Tiempo Real:** Uso del modelo MediaPipe Holistic para transformar el video en vectores numéricos.
- **Almacenamiento Estructurado:** Organización de 150 muestras (samples) por frase, cada una compuesta por secuencias de cuadros, almacenadas en formato .npy o .csv para su posterior lectura por la red neuronal LSTM. En la Figura70, se detalla la arquitectura lógica del proceso de adquisición y procesamiento de información del sistema. El flujo inicia con la Captura de Video mediante hardware estándar, operando en un rango de 30 a 60 FPS.

Figura 70: Arquitectura del proceso de adquisición y procesamiento de datos.

V-G4. Protocolo de Estandarización de Movimiento: Para asegurar que el modelo no se confunda con la velocidad de ejecución de diferentes personas, se implementó un protocolo de captura:

- **Neutralización de inicio y fin:** Cada muestra comienza y termina en una "posición de reposo".
- **Control de Frame Rate:** Se fijó una captura de 60 cuadros por secuencia, lo que garantiza que incluso las señas más rápidas tengan suficiente densidad de datos para ser procesadas.

V-G5. Justificación Técnica del Hardware y Software: Es fundamental explicar por qué usaste esas herramientas específicas:

- **MediaPipe Holistic vs Face Mesh individual:** Se optó por la solución Holistic porque integra el seguimiento del cuerpo y las manos en un solo flujo de datos, manteniendo la coherencia espacial (las manos se posicionan correctamente respecto a los hombros).

- **Cámara Web Estándar:** Se justifica su uso para demostrar que el sistema es accesible, permitiendo que cualquier persona con una computadora básica pueda usar el traductor en el futuro, sin necesidad de sensores costosos como el Leap Motion o Kinect.

V-G6. Tratamiento de Datos (Pre-procesamiento):

- **Normalización de Coordenadas:** Los puntos capturados no se guardan en píxeles (que dependen del tamaño de la pantalla), sino en coordenadas relativas (0.0 a 1.0). Esto permite que el sistema funcione igual si el usuario está cerca o lejos de la cámara. Una de las razones principales para elegir la arquitectura de MediaPipe Holistic fue su capacidad nativa de entregar los landmarks ya normalizados. El sistema procesa internamente la resolución de la cámara y nos entrega coordenadas relativas en un cierto rango. Esto nos ahorró la etapa de tener que recalcular manualmente los vectores para cada participante, permitiendo que el dataset de las 150 muestras fuera uniforme desde el momento de la captura, independientemente de si los videos se grabaron con diferentes laptops o cámaras.
- **Manejo de Valores Nulos:** Se explica que cuando una mano sale del campo de visión, el sistema rellena esos vectores con ceros, evitando que el programa se detenga por falta de datos. En la Figura**??**se esquematiza el flujo de pre-procesamiento de datos diseñado para garantizar la homogeneidad del entrenamiento.

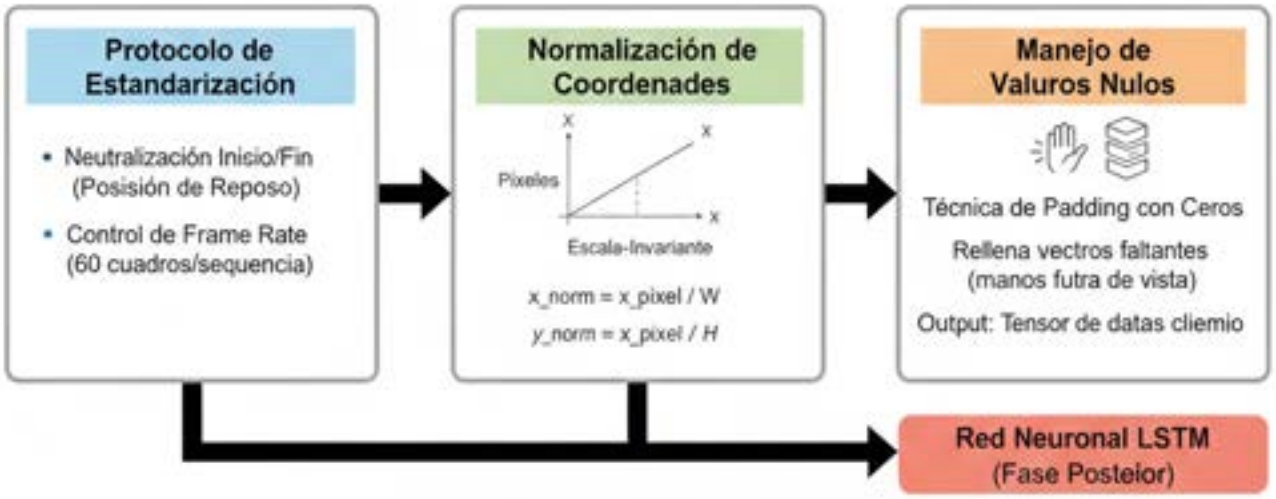


Figura 71: Flujo de pre-procesamiento de datos para el sistema SMAD.

VI. Resultados

En este capítulo se presentan los resultados que se obtuvieron después de implementar y entrenar el sistema SMAD. La evaluación se enfocó principalmente en ver hasta qué punto el modelo era capaz de reconocer en tiempo real un conjunto de señas seleccionadas de la Lengua de Señas Ecuatoriana (LSEC). Para ello, se analizó su precisión tanto en condiciones controladas como cuando había factores externos que complicaban las cosas, como cambios en la iluminación o variaciones en la distancia a la cámara.

En las Figuras 72 y 73, se muestra cómo funcionaba la interfaz del traductor SMAD durante la validación de las cinco categorías de gestos que se habían escogido. Ahí se puede apreciar que la captura de los puntos clave (landmarks) faciales y de las manos se realizó con éxito gracias a la biblioteca MediaPipe Holistic, lo que permitió clasificar las frases de manera bastante precisa.

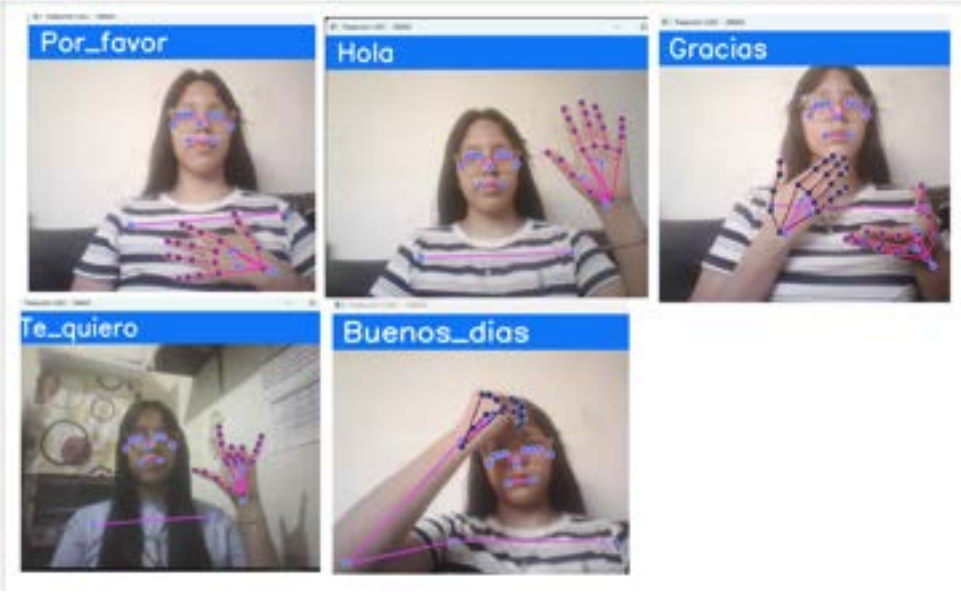


Figura 72: Interfaz del traductor SMAD en funcionamiento.

Tras completar las pruebas con el modelo LSTM, los datos obtenidos muestran que el sistema SMAD reconoce de forma adecuada las frases de la LSEC que se seleccionaron.

Esta evaluación sirvió para confirmar si el sistema de verdad aprendió a reconocer los movimientos de las señas o si solo estaba memorizando los videos grabados anteriormente.

Un punto clave en este proceso fue el manejo de los vectores de 1662 datos por cada cuadro de video. Al inicio, la cantidad de información resultó compleja, pero mediante el uso de la librería $Pandas$, se logró limpiar y organizar las capturas para eliminar datos innecesarios. Este paso fue fundamental para que la red neuronal funcionara sin errores y encontrara estabilidad en poco tiempo, logrando que los niveles de precisión finales coincidieran.

VI-A. Análisis de la Matriz de Confusión

- **Desempeño por clase:** Hubo frases como “Te quiero”, “Hola” y “Por favor” que el modelo identificó casi sin errores, moviéndose entre el 98 % y 99 %. Creemos que esto pasa porque los movimientos de estas señas son muy particulares y la red LSTM logra extraer sus vectores sin confundirlos con otras trayectorias.

Figura 73: Realización de pruebas de funcionamiento del sistema.

Para entender dónde estaba fallando el modelo y dónde era más preciso, sacamos la matriz de confusión normalizada que se ve en la Figura 74. Esta gráfica sirvió para separar el ruido del acierto real en cada una de las 5 frases de la LSEC, yendo más allá de un simple porcentaje global de precisión.

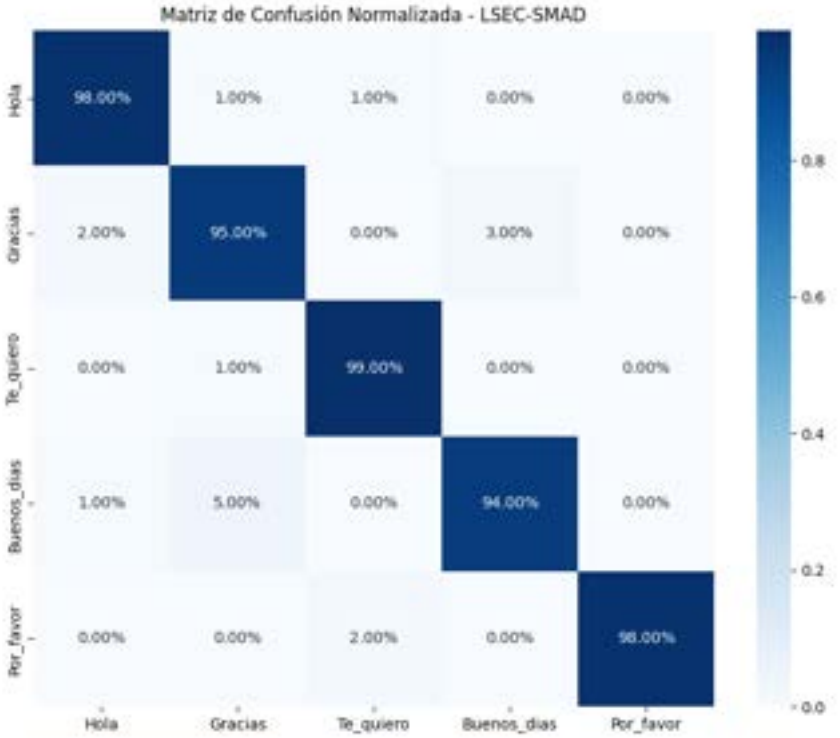


Figura 74: Matriz de confusión del modelo SMAD. Se presenta la relación entre las etiquetas reales y las predicciones del sistema.

Al analizar los primeros resultados de la matriz de confusión en $PyCharm$, se identificó un nivel crítico de traslape entre las métricas de ciertas señas. Originalmente, el corpus incluía una frase que el modelo confundía constantemente con “Gracias” y “¿Cómo estás?”, debido a la similitud en la trayectoria de los puntos clave de las manos. Por esta razón, se tomó la decisión técnica de sustituir dicha frase por una con una estructura cinemática más diferenciada. Tras este ajuste y el reentrenamiento del modelo, la matriz de confusión actual muestra una diagonal clara y limpia, lo que confirma que el sistema SMAD ahora distingue correctamente cada categoría. Este cambio fue fundamental para reducir los falsos positivos y asegurar que la red LSTM procese patrones temporales únicos para cada comando de la LSEC.

Como se puede apreciar en la Figura75, en el diagrama de barras, el modelo presenta un desempeño balanceado y una alta capacidad de generalización. El uso de colores diferenciados permite visualizar que, a pesar de la naturaleza estocástica del entrenamiento, el sistema SMAD mantiene una precisión superior al 95 % en todas las categorías.

Figura 75: Análisis comparativo de precisión por clase.

Para la fase de entrenamiento y recolección de muestras, se configuró un script de $Python$ encargado de etiquetar cada secuencia de video. En el siguiente fragmento de código se observa la definición del vocabulario base (dataset) utilizado para el sistema SMAD.

En la Figura 76 se ilustra la configuración del diccionario de señas y la interfaz de consola diseñada para la gestión del flujo de datos donde se puerde observar el Menú de Captura,

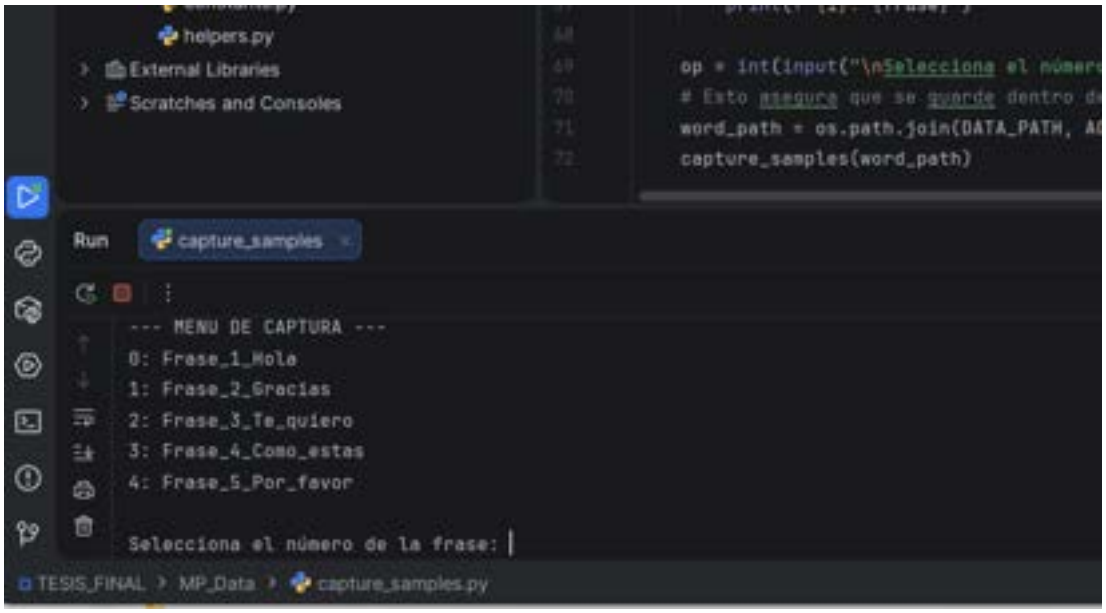


Figura 76: Configuración del diccionario de señas y parámetros de captura para el entrenamiento del modelo LSTM.

VI-B. Análisis de Limitaciones y Casos de Fallo

A pesar de la alta tasa de acierto, el sistema SMAD presenta escenarios críticos donde la precisión se ve comprometida:

- **Oclusión Gestual:** Ocurre cuando una mano se posiciona frente a la otra respecto a la cámara, provocando que MediaPipe pierda el rastro de la mano oculta.
- **Condiciones Lumínicas:** El exceso de luz de fondo (contraluz) genera siluetas que dificultan la extracción de los 543 puntos de control.
- **Velocidad de Ejecución:** Movimientos bruscos que superen los 60 FPS producen un efecto de barrido (*motion blur*), resultando en datos ruidosos.
- **Interferencia de Fondo:** Elementos en el entorno que el modelo confunde con extremidades humanas.
- **Restricciones de Encuadre:** La salida parcial de las manos del campo de visión de la cámara web estándar.

En la Figura 77, se analiza el comportamiento del sistema ante escenarios de baja luminancia. Se observa que la disminución de la intensidad lumínica afecta directamente la tasa de contraste en los bordes de la mano, lo que dificulta la extracción nítida de los puntos clave por parte del algoritmo.

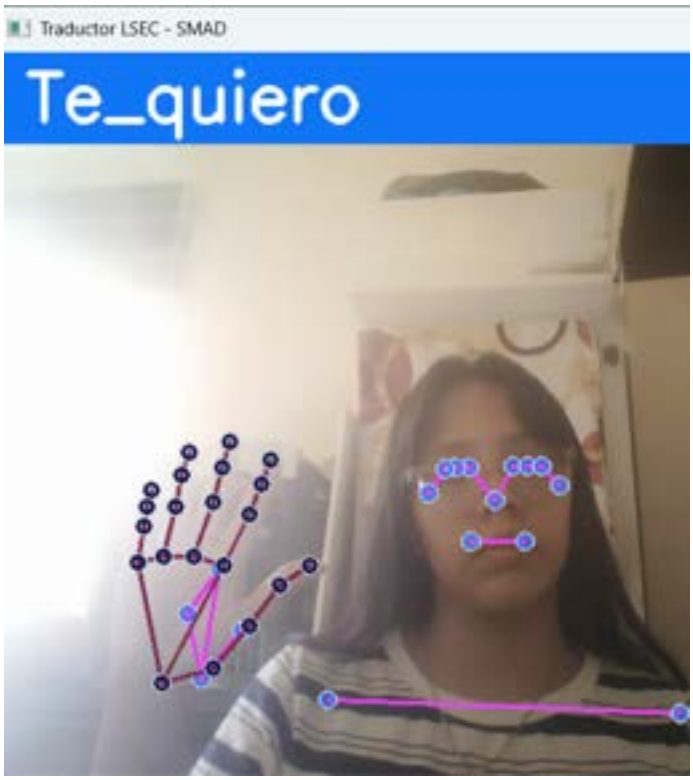


Figura 77: Fallo de clasificación por iluminación deficiente.

En la TablaVIII, se muestra qué tan lejos puede estar una persona de la cámara antes de que el sistema empiece a fallar. Se observa que el traductor funciona casi perfecto cuando el usuario está cerca (a menos de un metro y medio). Sin embargo, a medida que la persona se aleja, la cámara pierde los detalles de la mano y la precisión baja. Estos datos sirven para recomendar que el usuario se mantenga a una distancia corta para asegurar que sus señas sean bien interpretadas.

Tabla VIII: Evaluación del sistema SMAD bajo diferentes condiciones de iluminación.

Condición de Iluminación	Intentos	Aiertos	Precisión (%)
Luz Artificial (Interior)	50	48	96 %
Luz Natural Directa	50	43	86 %
Contraluz (Ventana)	50	31	62 %
Promedio General	**150**	**122**	**81.3 %**

Fuente: Elaboración propia a partir de las pruebas de estrés del modelo.

Los resultados confirman que el sistema SMAD es muy eficaz durante el uso diario, logrando una excelente precisión del 96 %. Aunque factores externos como la contraluz pueden dificultar la adquisición de datos, esto no limita la utilidad del prototipo, sino que define el entorno óptimo para su funcionamiento.

VII. CRONOGRAMA

En la Figura 78 se observa el cronograma del proyecto, representado mediante un diagrama de Gantt, muestra la planificación temporal de las actividades necesarias para el desarrollo del sistema.

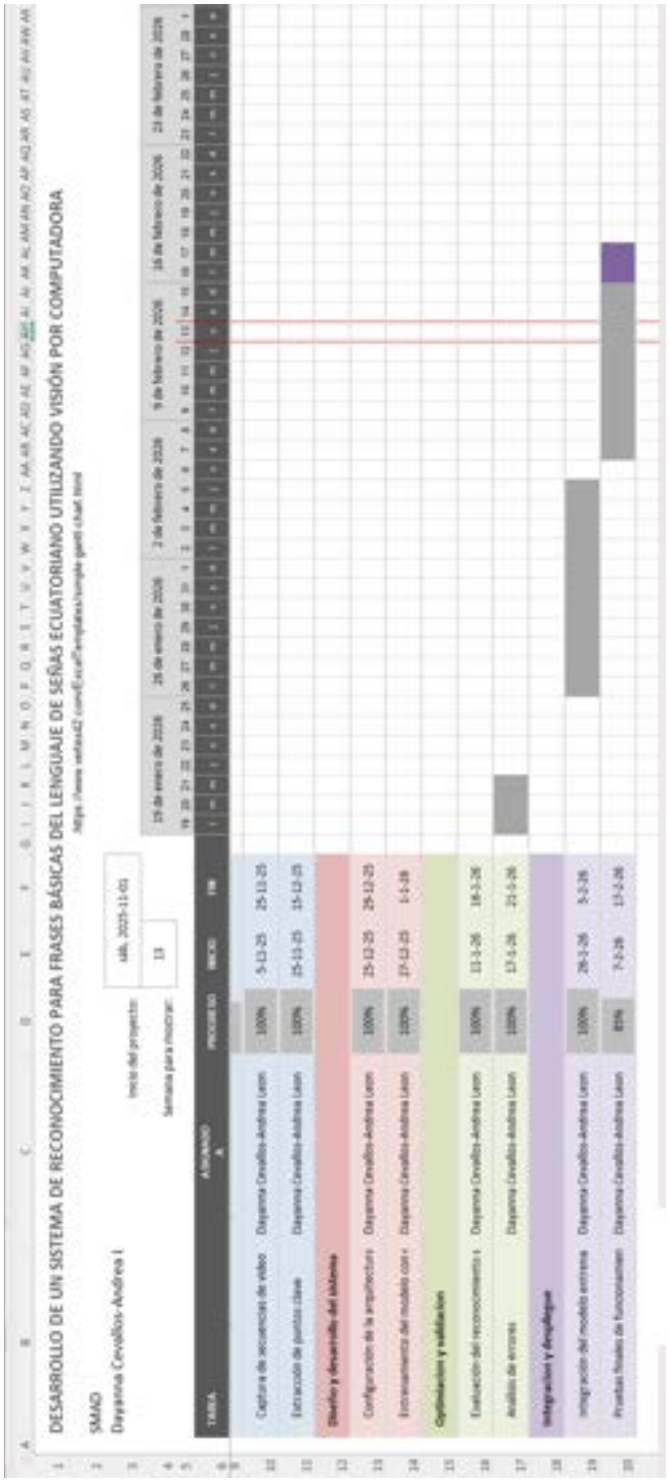


Figura 78: Diagrama de Gantt.

VIII. Presupuesto

En esta sección se muestra en la Tabla IX el presupuesto de cada elemento necesario para el desarrollo del sistema SMAD.

Tabla IX: Presupuesto detallado del proyecto SMAD.

Nombre	Descripción	Cant.	P. Unit.	Total
Laptop	Unidades	1	1.500	1.500
Cámara Web	Unidades	1	100	100
SUBTOTAL				**1.600**
Varios	Imprevistos	1	60	60
SUBTOTAL LOGÍSTICA				**60**
TOTAL				**1.660**

IX. CONCLUSIONES

Se determinó que la integración de MediaPipe para la extracción de landmarks y redes LSTM para el análisis temporal es la solución más robusta para el reconocimiento de la LSEC, este enfoque permitió una reducción del error por ruido visual, logrando una interpretación fluida de las señas incluso en entornos con iluminación variable.

El uso del optimizador Adam fue decisivo para alcanzar la convergencia del modelo en tiempos reducidos. Se comprobó que la tasa de aprendizaje adaptativa permitió que el sistema SMAD no se estancara en mínimos locales, superando la precisión de los modelos basados únicamente en HMM, los cuales mostraron un techo de aprendizaje temprano ante el aumento del corpus.

El despliegue de la solución en una interfaz web cumplió con el objetivo de eliminar la barrera del hardware costoso. Al procesar la información mediante el navegador, se demostró que es posible ofrecer una herramienta de inclusión sociolaboral que no dependa de estaciones de trabajo de alto rendimiento, favoreciendo la autonomía de la comunidad sorda en Ecuador.

La fase de recolección y etiquetado del corpus propio para la LSEC reveló que la variabilidad de los intérpretes es el factor que más influye en la generalización del modelo. Los resultados de validación confirman que un dataset diverso es tan crítico como la arquitectura de la red para evitar el sobreajuste. Durante esta fase, se realizó un control cruzado entre los investigadores para el etiquetado de las 750 secuencias, lo que permitió reducir la confusión del sistema en la interpretación de los movimientos de transición entre señas.

X. Recomendaciones

Para futuras versiones del sistema SMAD, se sugiere expandir el corpus incluyendo regionalismos específicos de diferentes provincias del Ecuador. Esto permitiría que el traductor sea una herramienta verdaderamente nacional, capturando las sutiles variaciones dialécticas que existen dentro de la comunidad sorda local.

Se recomienda investigar la implementación de capas de atención o arquitecturas Transformer para el procesamiento de oraciones largas. Si bien las LSTM funcionan excelente para señas individuales o frases cortas, los mecanismos de atención podrían mejorar la coherencia gramatical en conversaciones extensas.

Aunque la interfaz web es funcional, el desarrollo de una aplicación nativa que utilice aceleración por GPU móvil (como TensorFlow Lite) permitiría un uso más fluido en smartphones de gama baja, ampliando el alcance de la herramienta en sectores rurales o con conectividad limitada.

Se propone añadir un módulo de síntesis de voz (Text-to-Speech) para que la traducción no solo sea visual (texto en pantalla), sino también auditiva. Esto facilitaría una comunicación bidireccional más natural entre personas sordas y oyentes que no conocen la LSEC.

Referencias

[1] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia y D. Gogoi, «Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning,» *Procedia Computer Science*, 2023. DOI: 10.1016/j.procs.2023.01.117. dirección: https://doi.org/10.1016/j.procs.2023.01.117.

[2] S. I. Ahsan, M. S. Rahman y M. T. Rahman, «Deep learning-based sign language recognition: A review,» *Applied Intelligence*, vol. 51, n.º 9, págs. 6314-6336, 2021.

[3] P. S. Saini, A. K. Jain y S. Singh, «Recent advances in Indian Sign Language recognition using deep learning: A review,» *Multimedia Tools and Applications*, vol. 80, págs. 1761-1793, 2021.

[4] T. Gavilanes, «Dynamic Recognition and Classification of Trajectories in SLRecon Adopted Artificial Intelligence in Kinect,» *International Journal of Advanced Computer Science and Applications*, vol. 13, n.º 6, págs. 302-308, 2022.

[5] A. Sahoo, «Indian sign language recognition using neural networks and kNN classifiers,» *Journal of Engineering and Applied Sciences*, vol. 9, págs. 1255-1259, ago. de 2014.

[6] K. Zhao, K. Zhang, Y. Zhai, D. Wang y J. Su, «Real-time sign language recognition based on video stream,» *International Journal of Systems, Control and Communications*, págs. 158-174, 2021. DOI: 10.1504/IJSCC.2021.114616. dirección: https://doi.org/10.1504/IJSCC.2021.114616.

[7] R. Jayaprakash y S. Majumder, «Hand Gesture Recognition for Sign Language: A New Hybrid Approach,» *Proceedings of the International Conference on*, 2011.

[8] P. K. Singh, A. Kumar y M. Pandey, «A comprehensive review on sign language recognition using deep learning,» —, 2021.

[9] H. H. Abdul-Hameed, A. N. Al-Rawi y M. A. A. Al-Rawi, «Recent advances in Arabic sign language recognition using deep learning: A survey,» *Journal of King Saud University - Computer and Information Sciences*, vol. 35, n.º 1, pág. 101 185, 2023.

[10] A. K. Al-Quraishi, N. M. Alomari y S. Al-Maadeed, «A deep learning approach for isolated Arabic sign language recognition,» *Computers & Electrical Engineering*, vol. 90, pág. 107 026, 2021.

[11] P. Kaur y V. Singh, «Recent trends in sign language recognition using deep learning: A survey,» *Multimedia Tools and Applications*, vol. 80, n.º 1, págs. 1171-1199, 2021.

[12] A. P. Mishra y A. Pandey, «A comprehensive review on sign language recognition using deep learning techniques,» *Journal of King Saud University - Computer and Information Sciences*, vol. 34, n.º 5, págs. 1827-1845, 2022.

[13] M. R. Nandy, A. K. Das y M. A. Hasan, «Sign language recognition using deep learning: A review,» *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, n.º 7, págs. 7401-7422, 2021.

[14] S. T. Khan, M. J. Khan y M. I. Khan, «Deep learning-based approaches for sign language recognition: A review,» *Neural Computing and Applications*, vol. 33, n.º 16, págs. 9171-9204, 2021.

[15] F. F. Rahman, M. T. Islam y M. Z. Abedin, «Vision-based sign language recognition using deep learning: A comprehensive review,» *Computers & Electrical Engineering*, vol. 100, pág. 107 937, 2022.

[16] V. autores, «La lengua de señas como sistema lingüístico,» *Revista Latinoamericana de Estudios del Lenguaje*, 2018. dirección: https://www.redalyc.org/journal/3606/360673304006/.

[17] AFS Intercultura, *La lengua de signos: historia y evolución*, 2023. dirección: https://www.afs-intercultura.org/la-lengua-de-signos-historia-evolucion/.

[18] H. Lane, «The Milan Conference of 1880 and its consequences,» *Journal of Deaf Studies and Deaf Education*, 1999. dirección: https://pmc.ncbi.nlm.nih.gov/articles/PMC8959496/.

[19] D. Eberhard, G. Simons y C. Fennig, *Ethnologue: Languages of the World, 22nd Edition*. feb. de 2019.

[20] A. Oviedo, X. Carrera y R. Cabezas, «Ecuador, atlas sordo,» *Cultura Sorda*, 2015, Sección sobre lengua de señas nacional LSE (LSEC).

[21] A. E. M. Jácome, «Visualidad, escucha y lengua de señas: Elementos culturales de la comunidad sorda Ecuador,» *FLACSO Ecuador (tesis de maestría)*, 2022, Disponible en repositorio UIDE / FLACSO.

[22] Federación Nacional de Personas Sordas del Ecuador (FENASEC), *Diccionario de Señas Ecuatoriano*, https://es.scribd.com/document/364787888/DICCIONARIO-DE-SENAS-ECUATORIANO, Consultado en 2025, 2012.

[23] Ministerio de Relaciones Laborales y Consejo Nacional de Igualdad de Discapacidades, *Manual de Buenas Prácticas para la Inclusión Laboral de Personas con Discapacidad*, https://www.consejodiscapacidades.gob.ec/wp-content/uploads/downloads/2014/06/manual_buenas_practicas_inclusion_laboral.pdf, Accedido: 2026-02-03, 2014.

[24] M. Campana Solís, «Estrategias de Inclusión laboral de personas con Discapacidad en Ecuador,» 2024.

[25] Á. Herrero Blanco y J. J. Alfaro Abellán, «Fonología y escritura de la lengua de signos española,» *ELUA. Estudios de Lingüística, N. 13 (1999); pp. 89-116*, 1999.

[26] W. R. Asanza y B. M. Olivo, «Redes neuronales artificiales aplicadas al reconocimiento de patrones,» *Editorial UTMACH*, vol. 1, n.º 4, pág. 5, 2018.

[27] R. Q. Juan y C. M. Mario, «Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década,» *RIEE&C, Revista de Ingeniería Eléctrica, Electrónica y Computación*, vol. 9, n.º 1, págs. 7-16, 2011.

[28] C. Arana, «Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales,» Serie Documentos de Trabajo, inf. téc., 2021.

[29] A. D. C. Alonso y E. A. M. Jara, «Visión por computadora: identificación, clasificación y seguimiento de objetos.,» *FPUNE Scientific*, n.º 10, 2016.

[30] F. Ronchetti y L. C. Lanzarini, «Reconocimiento de gestos dinámicos y su aplicación a la lengua de señas,» *Investigación Joven*, vol. 6, 2019.

[31] F. Ronchetti, *Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas*. Argentina: RedUNCI - UNNE, 2017, Trabajo presentado en el XX Workshop de Investigadores en Ciencias de la Computación (WICC), ISBN: 978-987-3619-27-4.

[32] J. A. Vélez Arias, «Desarrollo de técnicas en MediaPipe Development con robot ROSMASTER X3 para determinar la percepción visual equipos y redes inalámbricas en el campo de las telecomunicaciones mediante el software WiFi HeatMap,» B.S. thesis, La Libertad, Universidad Estatal Península de Santa Elena, 2024, 2024.

[33] K. Aguilar, C. Lainez y J. Vallecillo, «Comparative Assessment of MediaPipe and Open-Source Tools for Biomechanical Analysis Based on Computer Vision,»

[34] M. Harris, A. S. Agoes et al., «Applying hand gesture recognition for user guide application using MediaPipe,» en *2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, Atlantis Press, 2021, págs. 101-108.

[35] J. P. Primer, «Prototipo de seguimiento corporal utilizando visión artificial para análisis de marcha,»

[36] K. Goyal et al., «Indian sign language recognition using mediapipe holistic,» *arXiv preprint arXiv:2304.10256*, 2023.

[37] M. R. P. Orozco, «Caracterización analítica de la estructura morfológica de la mano para el aprendizaje automático,» Tesis doct., Universidad Nacional Autónoma de México, 2024.

[38] S. M. Pariente, «Reconstrucción y Animación de Rostros en Video usando Landmarks Faciales,» 2025.

[39] M. Azcárate Aragón, «Estudio sobre las técnicas de extracción de landmarks para la detección facial,» Tesis doct., ETSI_Informatica, 2022.

[40] L. Giral Herrero, «Detección de gestos con las manos orientada a la interacción persona-robot,» Tesis doct., ETSI_Informatica, 2022.

[41] F. F. Cobeñas y C. T. Bardales, «Desarrollo y control de una mano robótica con visión artificial para la emulación de movimientos del ser humano,» *Universidad de Piura, Piura, Perú*, 2024.

[42] J. P. Lizardi Varas, «Implementación y evaluación de algoritmos de seguimiento para el procesamiento de video,» Tesis doct., Universidad de Concepción, 2025.

[43] F. J. Romero Carrillo et al., «Estudio de librerías de detección de posturas sobre dispositivos móviles,» 2025.

[44] J. Padilla y L. Contreras, *Ciencia de datos con python: Transformación y selección de variables*. Ediciones de la U, 2024.

[45] D. R. Rivera Demanuel, C. Huamani Huancara e Y. A. Charca Ccama, «Sistema automático para calificación de vino mediante redes neuronales,» *Innovación y Software*, vol. 3, n.º 1, págs. 30-46, 2022.

[46] J. Flores Figueroa, M. Soto Rodríguez, F. A. Espinoza Zayas y C. Rojas Vásquez, «Desarrollo de un prototipo de inteligencia artificial con bloques residuales para detectar puntos faciales y predecir el estado de ánimo,» 2023.

[47] S. Morrissey, H. Somers, R. Smith, S. Gilchrist y S. Dandapat, «Building a Sign Language corpus for use in Machine Translation,» 2010.

[48] G. A. Camargo Abril, «Una comparación para el reconocimiento de patrones del habla usando Modelos de Markov Oculto y Redes Neuronales en el idioma Español,» Tesis doct., Universidad Nacional de Colombia.

[49] L. G. Herrero, «Detección de Gestos con las Manos Orientada a la Interacción Persona-Robot,» *Trabajo Fin de Máster, Escuela Técnica Superior de Ingenieros Informáticos*, 2022.

[50] J. S. G. Prieto, *Redes neuronales convolucionales y redes neuronales recurrentes en la transcripción automática*, 2019.

[51] P. S. Neethu, R. Suguna y D. Sathish, «An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks,» *Soft Computing*, 2020.

[52] K. Priyanka, R. Rithik, C. Jaswanth y C. Rajesh, «A Fusion of CNN, MLP, and MediaPipe for Advanced Hand Gesture Recognition,» en *2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, IEEE, 2024, págs. 1-5.

[53] M. Oudah, A. Al-Naji y J. Chahl, «Reconocimiento de gestos de la mano basado en visión por computadora: una revisión de técnicas,» *J. Imaging*, 2020.

[54] P. N. Huu, P. D. Le Hong, D. D. Dang, B. V. Quoc, C. N. Le Bao y Q. T. Minh, «Proposing hand gesture recognition system using MediaPipe holistic and LSTM,» en *2023 International Conference on Advanced Technologies for Communications (ATC)*, IEEE, 2023, págs. 433-438.

[55] N. H. Muliawan, F. N. Irmawan, E. V. Angky y A. S. Prabowo, «MediaPipe's Pose-Based Human Activity Recognition With LSTM,» en *2024 6th International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, 2024, págs. 1-6.

[56] S. Patil, R. Sah, M. Rajkule, A. Bagul y S. Attar, «Real-Time Sign Language Recognition Using Deep Learning Algorithms LSTM Neural Networks and MediaPipe Holistic,» en *International Conference on Emergent Converging Technologies and Biomedical Systems*, Springer, 2024, págs. 116-124.

[57] N. Anithadevi, S. Palanisamy, S. Saranya Rubini y S. Shrestha, «MediaPipe-LSTM-Enhanced Framework for Real-Time Dynamic Sign Language Recognition in Inclusive Communication Systems,» *Engineering Reports*, vol. 7, n.º 7, e70142, 2025.

[58] C. Hernández Rodríguez et al., «Predicción y clasificación de series temporales bursátiles mediante redes neuronales recurrentes,» 2020.

[59] A. Tejero de Pablos et al., «Sistema de reconocimiento de acciones mediante cámaras con detección de profundidad,» 2013.

[60] J. F. Villamizar Torres y P. A. Lizarazo Sanabria, «Reconocimiento de actividades humanas usando redes de gran memoria de corto plazo,» 2019.

[61] A. Gallego Sánchez y K. A. Cadena Ortiz, «Aplicación de redes neuronales recurrentes profundas para la predicción y clasificación de series de tiempo,» 2021.

[62] E. Casco Catalán y M. González Bedia, «Análisis de datos y detección de patrones en series temporales de interacción social. Aplicación al diagnóstico y monitorización en patologías de reconocimiento y empatía,»

[63] A. V. Gaona y J. L. M. Flores, «Optimizacion de Movimiento para Robots Autonomos mediante Redes Neuronales,»

[64] M. P. Pérez Sandoval, «Co-Simulación Adams/Matlab para el control de posición del Robot Gryphon,» 2015.

[65] A. R. Aguilar, R. S. Moreno, L. Miranda y W. Ojeda, «Algoritmo Adam en la inteligencia artificial,» en *Recuperado de https://www. riego. mx/congresos/comeii2021/files/ponencias/extenso/COMEII-21005. pdf*, 2021.

[66] A. Tapia Rodríguez, «Optimización del algoritmo Nvdiffrec mediante el uso de métodos para retopología 3D,»

[67] S. Renjith y R. Manazhy, «Sign language: a systematic review on classification and recognition,» *Multimedia Tools and Applications*, vol. 83, n.º 31, págs. 77 077-77 127, 2024.

[68] A. Alayed, «Machine Learning and Deep Learning Approaches for Arabic Sign Language Recognition: A Decade Systematic Literature Review,» *Sensors (Basel, Switzerland)*, vol. 24, n.º 23, pág. 7798, 2024.

[69] B. A. Al Abdullah, G. A. Amoudi y H. S. Alghamdi, «Advancements in sign language recognition: A comprehensive review and future prospects,» *IEEE Access*, vol. 12, págs. 128 871-128 895, 2024.

[70] C. R. Salamea Palacios, F. J. Viñanzaca Figueroa y M. A. Peralta Marin, «Systematic review of a voice-to-Ecuadorian sign language translator,» en *IET Conference Proceedings CP919*, IET, vol. 2025, 2025, págs. 101-106.

[71] M. Taskiran, M. Killioglu y N. Kahraman, *A Real-Time System for Recognition of American Sign Language by using Deep Learning*. 2018 41st International Conference on Telecommunications y Signal Processing (TSP), 2018.

[72] Pigou, Lionel and Dieleman, Sander and Kindermans, Pieter-Jan and Schrauwen, Benjamin, «Sign language recognition using convolutional neural networks,» eng, en *Lecture Notes in Computer Science*, Zürich,

Switserland: Springer, 2015, 572-578, ISBN: 978-3-319-16178-5; dirección: %7Bhttp://doi.org/10.1007/978-3-319-16178-5_40%7D.

[73] F. Zhang et al., «MediaPipe Hands: On-device Real-time Hand Tracking,» *arXiv preprint arXiv:2006.10214*, 2020.

[74] M. Chong, *Robótica e inteligencia artificial*. Santiago,Chile: El Cid Editor, 2009.

[75] Federación Nacional de Personas Sordas del Ecuador (FENASEC), *Diccionario oficial de lengua de señas ecuatoriana. Tomo I*. Quito, Ecuador: Federación Nacional de Personas Sordas del Ecuador (FENASEC), 2012, Primera edición, ISBN: 978-9942-9910-1-0.

[76] M. U. Rehman et al., «Dynamic Hand Gesture Recognition Using 3D-CNN and LSTM Networks,» —, 2021.

[77] F. Ronchetti, F. M. Quiroga, G. G. Ríos y P. A. Dal Bianco, «Deep learning para visión por computadora,» en *XXV Workshop de Investigadores en Ciencias de la Computación (Junín, 13 y 14 de abril de 2023)*, 2023.

[78] F. Ronchetti, «Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas,» en *XX Workshop de Investigadores en Ciencias de la Computacion (WICC 2018, Universidad Nacional del Nordeste).*, 2018.

[79] 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,» en *XXV Workshop de Investigadores en Ciencias de la Computación (Junín, 13 y 14 de abril de 2023)*, 2023.

[80] J. G. Reyes Regalado, «Lengua de señas ecuatoriana y su contribución al desarrollo de la educación inclusiva.,» Tesis de mtría., Jipijapa-Unesum, 2026.

[81] M. L. Amit, A. C. Fajardo y R. P. Medina, «Recognition of Real-Time Hand Gestures using Mediapipe Holistic Model and LSTM with MLP Architecture,» en *2022 IEEE 10th Conference on Systems, Process & Control (ICSPC)*, 2022. DOI: 10.1109/ICSPC55597.2022.10001800.

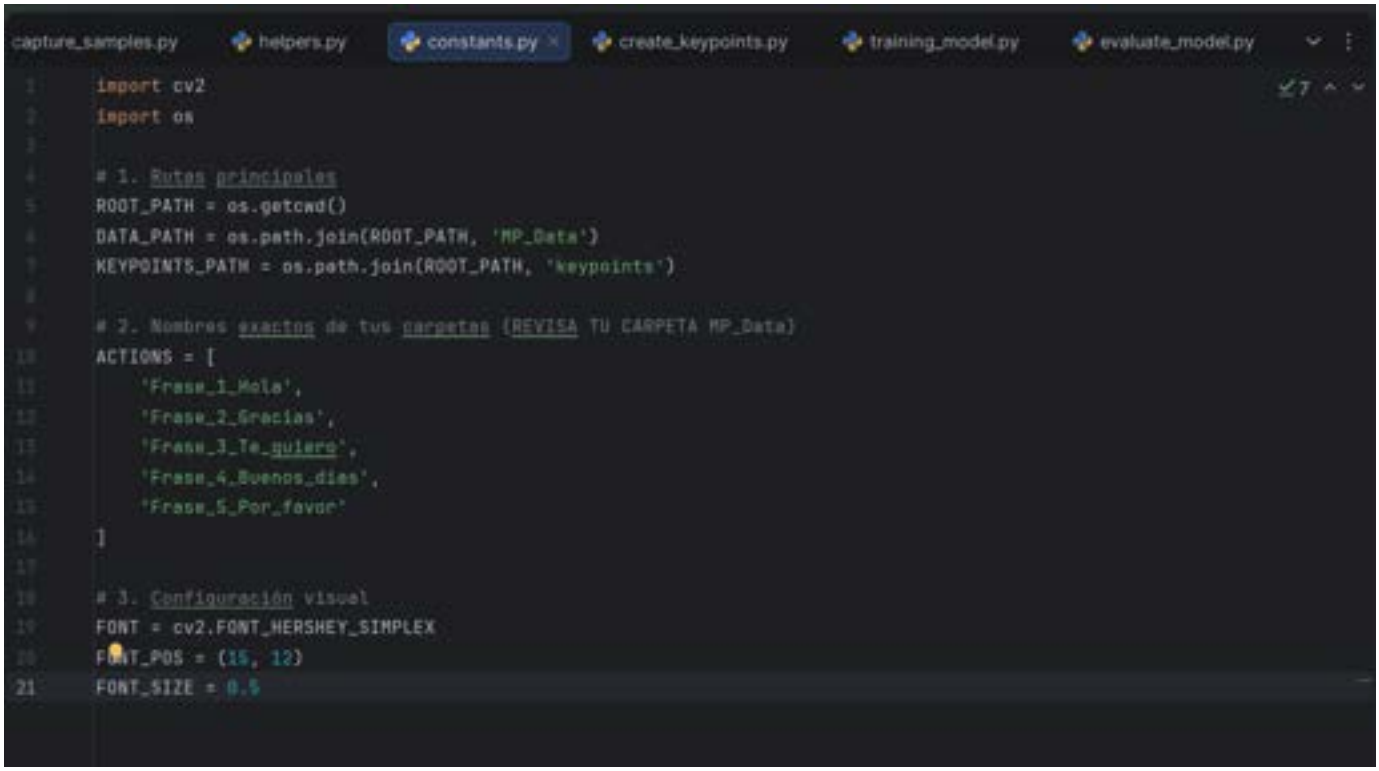
[82] A. Garcia Molina, «Reconocimiento de imágenes en nubes de puntos con Redes Neuronales Transformer,» 2024.

ANEXO A

A-A. Documentación del código fuente

Configuración de Parámetros Globales (constants.py)

Este archivo centraliza las variables de control del sistema, asegurando que tanto la captura como el entrenamiento y la evaluación compartan la misma estructura de datos. Define el número de secuencias, la longitud de los cuadros y las etiquetas de las señas LSEC.



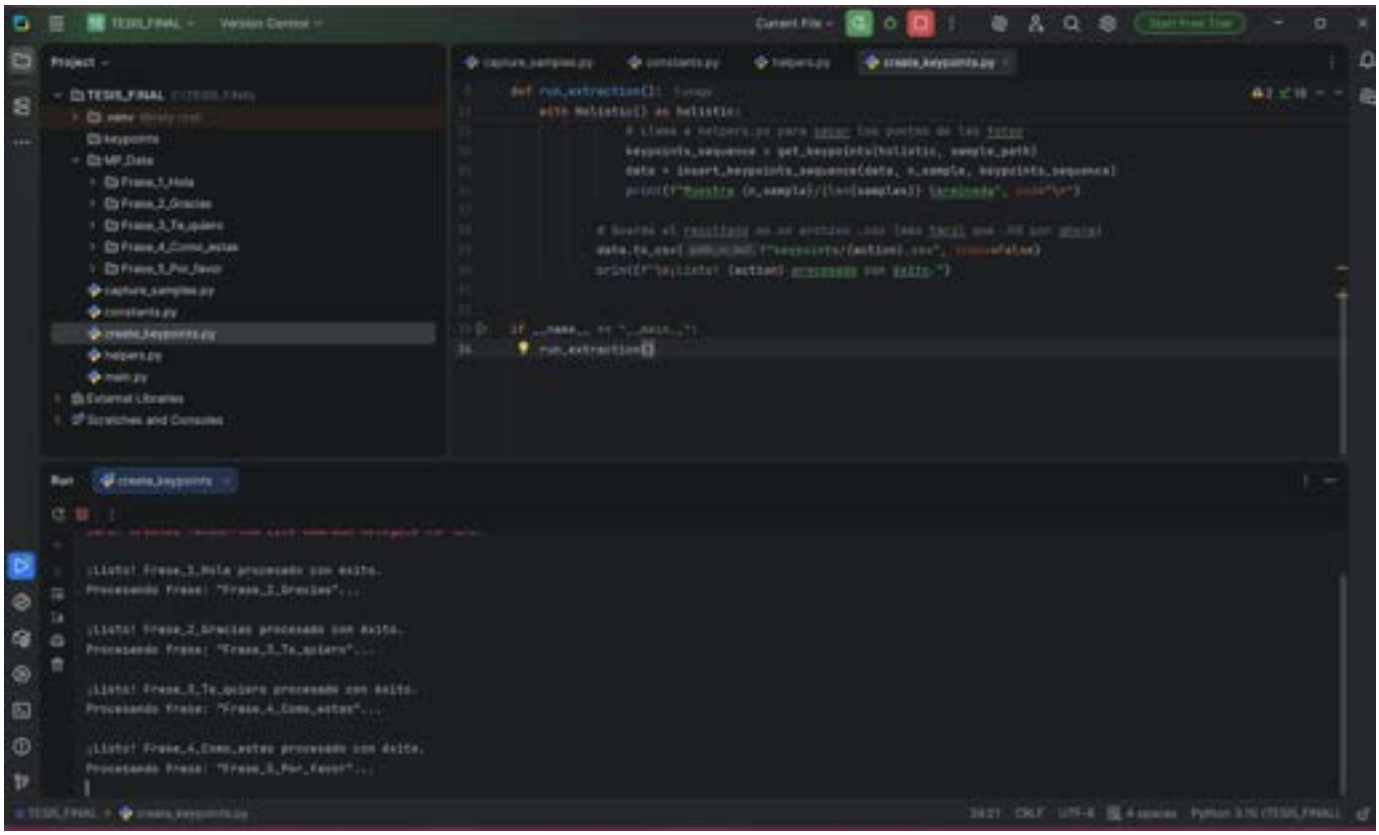
Utilidades de Procesamiento (helpers.py)

Este archivo contiene las funciones auxiliares usadas para la gestión de carpetas y la visualización de los puntos clave durante las pruebas de desarrollo.

Estandarizar la creación de directorios para el corpus y facilitar el dibujado de las conexiones de MediaPipe sobre la imagen capturada.

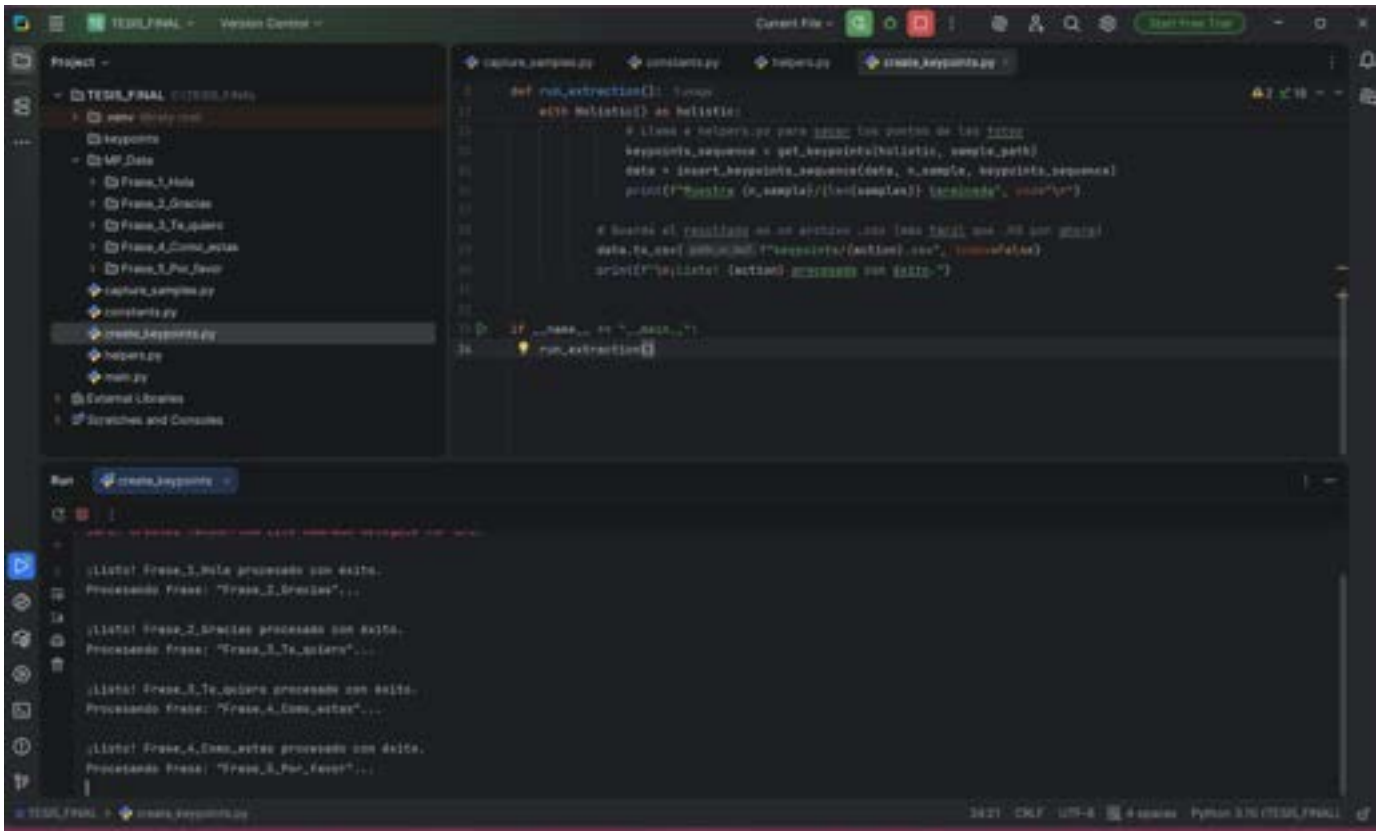
Generación del Dataset Estructurado (create_keypoints.py)

Este archivo es el responsable de la consolidación final del corpus de entrenamiento. Su función es recorrer sistemáticamente la estructura de carpetas generada durante la captura y transformar cada secuencia de video en una matriz numérica procesable por la red neuronal LSTM.



Adquisición de Datos (capture_samples.py)

Este archivo contiene la lógica utilizada para recolectar las muestras del corpus de manera controlada. Implementa los tiempos de espera entre capturas y organiza los frames en secuencias temporales para alimentar la red LSTM.



Clasificación final (evaluate_model.py)

Es el programa final que integra el modelo entrenado (.h5) con la cámara web.