Herramientas de aprendizaje automático con redes neuronales para el reconocimiento de Lengua de Señas Colombiana

Andrés Quintero Bedoya

Tesis de grado para optar al título de Ingeniero de Sistemas

Asesores

Danny Alexandro Múnera Ramírez, Ph.D.

Ana Lucía Pérez Patiño, Ph.D.

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Sistemas

Medellín, Antioquia, Colombia

2024

Cita	Quintero-Bedoya, 2024 [1]
Referencia Estilo IEEE (2020)	[1] A. Quintero-Bedoya "Herramientas de aprendizaje automático con redes neuronales para el reconocimiento de Lengua de Señas Colombiana", Tesis de grado, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, 2024

Dedicatoria

Con profundo cariño y gratitud, dedico este trabajo a mis queridos padres César y Stella. Su amor incondicional, su honestidad y su incansable enseñanza sobre el valor del estudio y del conocimiento han sido la brújula de mi vida. A través de sus ejemplos de respeto y honestidad, me han guiado por un camino de integridad y perseverancia.

A mi hermano David, cuyo apoyo inquebrantable ha sido un faro de esperanza y fortaleza.

A mi maravillosa novia Karen, cuya ternura, amor y motivación han sido el aliento en los momentos más difíciles. Gracias por estar a mi lado, por levantarme cuando he estado a punto de rendirme, y por compartir conmigo la luz de tu amor.

A mis abuelos Aura y Francisco, gracias por sus amorosas enseñanzas y por ser ejemplos de vida. Aunque el camino no siempre ha sido fácil, sus lecciones han sido faros de sabiduría en mi travesía.

Desde el cielo, sé que Mariela e Iván me observan con orgullo. A ellos, les dedico también este logro, llevando siempre en mi corazón la certeza de su amor y su presencia.

Dedico también este trabajo a la comunidad sorda y a todas las minorías de mi país, demostrando con este trabajo que la tecnología tiene el poder de cambiar la sociedad logrando más equidad y dando la esperanza por un mejor futuro.

Agradecimientos

Agradezco profundamente a mis asesores Ana Lucía Pérez Patiño y Danny Alexandro Múnera Ramírez por su esencial contribución y humanismo en este proyecto. Su guía ha sido un pilar en mi formación y desarrollo del trabajo. Agradezco también a la comunidad sorda, intérpretes, filólogos y filólogas, me quedaría corto mencionando a todas las personas, pero hago mención especial al Proyecto de accesibilidad física y tecnológica UdeA.

Agradecer también a mis compañeros de trabajo Santiago Bedoya Díaz y José Fernando Waldo Rojas por el compromiso y gran trabajo en diversos componentes del proyecto.

Un especial agradecimiento al profesor Diego Botía, cuyo apoyo en la universidad y en la construcción de la comunidad Web3ForU ha impulsado mi crecimiento profesional y personal.

Agradezco también a Santiago Sánchez, por ser más que un compañero, inspirándome con su profesionalismo y gran corazón.

Agradezco con especial cariño a Martín Elías Quintero y el equipo de Guane Enterprises, cuya mentoría ha sido fundamental en mi desarrollo profesional.

Por último y no menos importante, agradecer a la comunidad Web3ForU, que deja una huella en mi corazón, queriendo que siga creciendo, aportando así a la comunidad universitaria la universalidad del conocimiento.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

Siglas, acrónimos y abreviaturas

ASL	American Sign Language
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
HCI	Human-Computer Interaction
LFG	Lexical Functional Grammar
LSC	Lengua de Señas Colombiana
LSTM	Long Short Term Memory
Ph.D.	Philosophiae Doctor
RNAs	Redes Neuronales Artificiales
RNN	Recurrent Neural Network
SLR	Sign Language Recognizer
SMT	Statistical Machine Translation
STAG	Synchronous Tree Adjoining Grammar
TEAM	Translation from English to ASL by Machine
TLS	Traducción de Lengua de Señas
UdeA	Universidad de Antioquia

RESUMEN

Este estudio aborda la mejora de la inclusión social mediante la creación de un sistema de traducción para la Lengua de Señas Colombiana (LSC), aprovechando avances en inteligencia artificial. Se utiliza tecnología de redes neuronales, específicamente redes neuronales recurrentes con unidades de procesamiento GRU y LSTM. Estas se optimizaron buscando los mejores hiperparámetros en plataformas como TensorFlow y Keras. Además, se propone el desarrollo de una plataforma colaborativa para unir a organizaciones interesadas en el proyecto. Esto permitirá crear un amplio corpus de LSC, esencial para el crecimiento del proyecto. Se utilizó FastAPI para mejorar la interfaz de usuario, complementada con React, y Kubernetes para sostener la infraestructura, asegurando su escalabilidad y eficiencia.

El proyecto incluyó la creación de un corpus con 1500 videos de LSC. Esto no solo facilitó el entrenamiento de los modelos, sino que también generó una base de datos valiosa para futuras investigaciones. Los resultados preliminares indican un progreso notable en la interpretación automática de LSC, lo que podría tener un impacto significativo en la comunicación y la integración de la comunidad sorda en Colombia.

Se desarrollaron experimentos comparando el desempeño de dos tipos de redes neuronales recurrentes con capas GRU y LSTM, obteniendo un desempeño promedio en la métrica en el F1 de $84{,}62\,\% \pm 11{,}04\,\%$ con la red LSTM, mientras que con la red GRU se obtuvo un rendimiento promedio en el F1 del $91{,}17\,\% \pm 3{,}76\,\%$, lo que demostró una estabilidad mayor con las capas GRU.

Palabras clave **— Discapacidad auditiva, Machine Learning, Reconocimiento de Lengua de Señas, Lengua de Señas Colombiana, Redes Neuronales Recurrentes**

ABSTRACT

This study addresses the improvement of social inclusion by creating a translation system for Colombian Sign Language (LSC), leveraging advances in artificial intelligence. Neural network technology is used, specifically recurrent neural networks with GRU and LSTM processing units. These were optimized by seeking the best hyperparameters on platforms such as TensorFlow and Keras. In addition, the development of a collaborative platform is proposed to bring together organizations interested in the project. This will allow the creation of a broad LSC corpus, essential for the project's growth. FastAPI was used to improve the user interface, complemented with React, and Kubernetes to sustain the infrastructure, ensuring its scalability and efficiency.

The project included the creation of a corpus with 1500 LSC videos. This not only facilitated the training of the models but also generated a valuable database for future research. Preliminary results indicate significant progress in the automatic interpretation of LSC, which could have a significant impact on communication and the integration of the deaf community in Colombia.

Experiments were conducted comparing the performance of two types of recurrent neural networks with GRU and LSTM layers, achieving an average performance on the F1 metric of 84,62 %±11,04 % with the LSTM network, while with the GRU network, an average performance on the F1 of 91,17 %±3,76 % was obtained, demonstrating greater stability with GRU layers.

Keywords **— Hearing impairment, Machine Learning, Sign Language recognition, Colombian Sign Language, Recurrent Neural Network**

I. INTRODUCCIÓN

En Colombia, la Lengua de Señas Colombiana (LSC) se utiliza como medio de comunicación principal para las personas con discapacidad auditiva. La LSC es un lenguaje visual-gestual que se basa en movimientos de las manos, expresiones faciales y posturas corporales para transmitir significados y establecer un canal de comunicación efectivo.

A pesar de la importancia de la LSC, existe una limitada disponibilidad de recursos visuales de alta calidad que faciliten su aprendizaje y práctica. La falta de imágenes y videos precisos y claros de las señas básicas del LSC dificulta la creación de herramientas automáticas que permitan la comunicación fluida y eficiente entre las personas con discapacidad auditiva y las personas oyentes [1] [2]. Para abordar esta problemática, se propone una plataforma que apoye la generación de un conjunto de datos con imágenes de alta calidad de la LSC y que permita la creación de modelos matemáticos predictivos, con el objetivo de mejorar la comunicación de las personas con discapacidad auditiva en Colombia.

El presente trabajo se enfoca en el desarrollo de un sistema integral que permita la captura, organización, clasificación y análisis de las señas básicas del LSC. Con este fin, se establecen objetivos específicos que abarcan desde el desarrollo de un módulo de captura de imágenes y videos de señas básicas del LSC, hasta la implementación de un módulo de análisis de datos que permita identificar patrones y tendencias en el uso de las señas. Asimismo, se creó una base de datos estructurada y accesible, que facilite el aprendizaje y la práctica de la LSC, así como la búsqueda y selección de señas específicas según su significado y contexto de uso.

La traducción automática de lengua de señas (TLS) representa un avance significativo y un campo de estudio emergente que ha capturado el interés de investigadores en todo el mundo. Este campo se centra en convertir la lengua de señas, utilizado por personas sordas o con dificultades auditivas, en texto o voz mediante el uso de tecnología, permitiendo una comunicación más accesible. Recientes investigaciones en este ámbito han aplicado técnicas avanzadas de inteligencia artificial, como el análisis de imágenes y el aprendizaje automático, para interpretar y traducir los gestos y movimientos característicos de la lengua de señas

[3]. En el presente estudio, se introduce una metodología que primero procesa los videos capturados de señas mediante la extracción de sus aspectos más destacados como lo son las poses del cuerpo humano y el cambio en el tiempo que tienen las mismas para identificar una seña. Este proceso nos permite identificar con precisión los elementos clave de cada seña. Posteriormente, se aplica un modelamiento matemático basado en redes neuronales recurrentes, para analizar secuencias de movimientos y gestos en el tiempo, realizando así una traducción de Lengua de Señas Colombiana a texto.

II. PLANTEAMIENTO DEL PROBLEMA

En la Universidad de Antioquia, desde el semestre 2012-1 hasta el semestre 2022-2, se inscribieron 410 estudiantes sordos [4]. De estos, 322 estaban en la sede de Medellín y 69 en las sedes regionales (Andes, Apartadó, Carepa, Carmen de Viboral, Caucasia, Turbo y Yarumal). La distribución por estratos socioeconómicos muestra que el 28,13 % de los estudiantes sordos proviene del estrato 1, el 42,20 % del estrato 2, el 24,20 % del estrato 3, el 3,58 % del estrato 4, el 0,26 % del estrato 5, el 0,26 % del estrato 6 y el 0,77 % no tiene estrato socioeconómico asignado [4]. La Ilustración 1 muestra que los estudiantes del estrato 2 son mayoría, seguidos por los de los estratos 1 y 3.

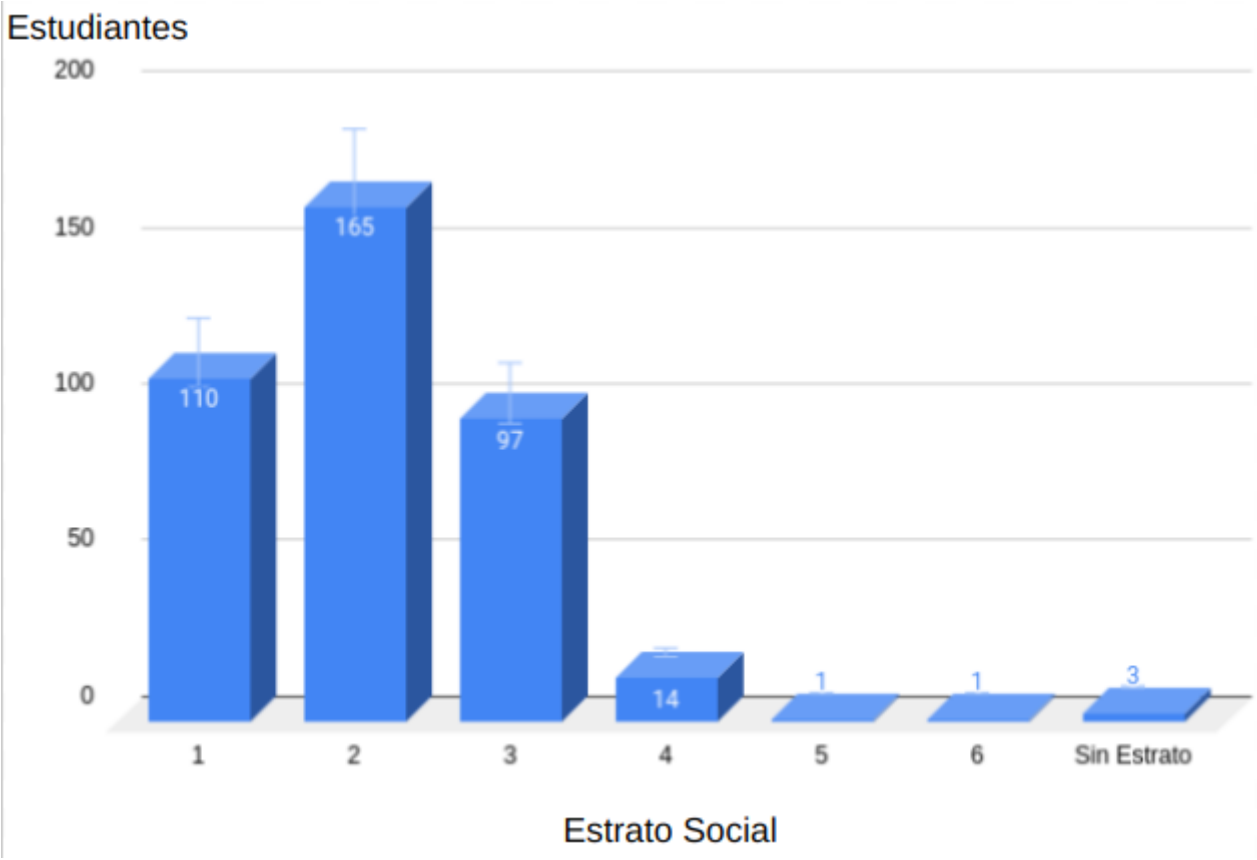

Fig. 1. Ilustración de cantidad de estudiantes sordos por estrato socioeconómico en la Universidad de Antioquia en 2023

Desde el semestre 2012-1 hasta el 2022-1, se admitieron 27 estudiantes sordos en total

en la Universidad de Antioquia, con el semestre 2022-1 registrando la mayor admisión, con 4 estudiantes. Durante este mismo período, 239 estudiantes sordos estuvieron matriculados, alcanzando el punto más alto en el semestre 2022-1 con 21 estudiantes. Además, 8 estudiantes sordos se han graduado y 10 abandonaron sus estudios. Estos datos demuestran que la población de estudiantes con discapacidad en la UdeA es significativa, y lo es aún más la cantidad de personas sordas que desean acceder a la educación superior. Es por esto que la Universidad de Antioquia se ha comprometido con ser un campus inclusivo y accesible, donde se puedan crear tecnologías para toda la población en situación de discapacidad. En este contexto se ha llevado a cabo el "Proyecto de accesibilidad física y tecnológica UdeA" [5], enfatizando la importancia de la integración educativa y tecnológica para los estudiantes sordos.

En el marco de este proyecto se contempla investigar en el desarrollo de TLS, sin embargo, actualmente existen varias limitaciones en el campo de la investigación de Traducción de Lengua de Señas; por ejemplo, hay una necesidad de estandarizar repositorios de lengua de señas, crear corpus robustos con la inclusión de la comunidad sorda y expertos en lengua de señas, y enfrentar el reto de la traducción directa de gestos faciales, que son de gran importancia gramatical y sintáctica en la lengua de señas [6].

Según el DANE en Colombia hay 1.784.372 personas con discapacidad [7] y de acuerdo con el Registro de Localización y Caracterización de Personas con Discapacidad, las personas con discapacidad registradas en Colombia se concentran principalmente en Bogotá (18.3 %) y Antioquia (13.8 %)[8]. Es claro que las familias, la sociedad, las instituciones privadas y, sobre todo, las instituciones públicas como las universidades, tienen un compromiso pendiente con esta población. Adicionalmente, más de la mitad de los colombianos con alguna discapacidad son mujeres (54.1 %); el 58,5 % se encuentra entre los 15 y 64 años y en este rango de edad [7].

En el contexto de la Universidad de Antioquia, a pesar de las apuestas pioneras por la inclusión y los reconocimientos como la "Primera universidad inclusiva del país" y "Compromiso con la inclusión", la igualdad de oportunidades que representan la accesibilidad universal y el diseño para todos siguen siendo retos para abordar en nuestra Alma Mater; entre otros

aspectos porque es necesario conocer con precisión la caracterización de la población con discapacidad, definir las necesidades de accesibilidad que tiene la comunidad universitaria, entre otros.

A. Antecedentes

Una Universidad Inclusiva es aquella que da la bienvenida a la diversidad. Su principal foco es la atención a la diversidad de toda la comunidad educativa (alumnado, profesorado y personal de administración y servicios), entendiendo diversidad como un espectro amplio que contribuye a la equidad a partir de la aceptación de la amplia gama en que las personas son diferentes (raza, género, etnia, edad, nacionalidad, cultura, religión, discapacidad, orientación sexual, estatus socioeconómico, idioma, estilos de aprendizaje, etc.). El concepto de Educación Inclusiva, como otros muchos vinculados al desarrollo de derechos sociales y educativos para las personas con discapacidad, surge de una evolución tanto conceptual como social y política, no surge de manera espontánea o uniforme [9].

Si bien el Instituto Nacional para Ciegos (INCI) hizo un reconocimiento a 22 universidades inclusivas del país que se puede observar en INCI (2021), en la Tabla I se presenta una síntesis de algunos esfuerzos por la educación inclusiva en las Universidades del país.

El compromiso de la alta dirección universitaria con las personas con discapacidad, la gestión de los recursos humanos, el cumplimiento de la normativa y las políticas inclusivas y de igualdad de oportunidades en todos los procedimientos de formación, selección y acceso al empleo, es una iniciativa con la que se vienen comprometiendo universidades del mundo [10].

En el enfoque de Traducción Automática Estadística (Statistical Machine Translation, SMT) se asume que con cierta puntuación de probabilidad, cada palabra en el idioma objetivo es una traducción de la palabra del idioma fuente. La mejor traducción se determina seleccionando la secuencia de palabras que tiene la mayor probabilidad [11, 6]. La decodificación es una fase más compleja en la traducción automática, donde los principales elementos recaen en la complejidad del cifrado y la reordenación del lenguaje objetivo [12].

TABLA I
TABLA DE PROYECTOS RELACIONADOS A EDUCACIÓN INCLUSIVA EN UNIVERSIDADES DE COLOMBIA

Universidad	**Acceso**
Universidad Nacional de Colombia	Observatorio de inclusión educativa para personas con discapacidad
Universidad del Rosario	IncluSer
Universidad del Norte	Universidad Incluyente
Universidad del Valle	Discapacidad e Inclusión
Universidad del Atlántico	DiverSer
Politécnico Grancolombiano	Programa de Inclusión
Universidad de Cartagena	UdeC Inclusiva
Universidad de Medellín	Inclusión
Universidad de Nariño	Aula de apoyos Tecnológicos

Desde el punto de vista investigativo se observa el proyecto TEAM [13] que consiste en una metodología conocida como Gramática de Árboles Adjuntos Sincrónicos (STAG) diseñada para transformar textos en la estructura sintáctica correspondiente Lengua de Señas Americana (ASL). También con proyecto ASL workbench [14] se observa la conversión de una entrada de texto a ASL apoyándose en la Gramática Funcional Léxica (LFG) para mapear las estructuras gramaticales del inglés en ASL. Además se implementa un modelo fonológico basado en los principios de movimiento y pausa que son fundamentales en la fonología del ASL.

En esta breve revisión del estado del arte se encuentra que a nivel mundial se ha investigado sobre le desarrollo de sistemas automáticos para la traducción de Lengua de Señas a un idioma escrito. Sin embargo, la gran mayoría de trabajos se centra en lenguas de señas de otros países, principalmente la Lengua de Señas Americana (ASL).

III. JUSTIFICACIÓN

En el conjunto de los diecisiete Objetivos de Desarrollo Sostenible se hacen once menciones expresas a las personas con discapacidad [15]. La Convención sobre los derechos de las personas con discapacidad de la Organización de Naciones Unidas (ONU), reafirma que las personas, con todos los tipos de discapacidad, deben poder gozar de los derechos humanos y libertades fundamentales [16]. La convención de la ONU destaca la accesibilidad como la "puerta de acceso a los derechos". Con respecto a las mujeres con discapacidad, la convención hace mención en siete de sus artículos, a saber: 3, 6, 8, 16, 25, 28 y 34 y señala factores de exclusión como el analfabetismo, los bajos niveles educativos, la dependencia económica, la pobreza, el irrespeto social hacia los derechos reproductivos y las violencias. Colombia ratificó el acuerdo en el 2011 y se comprometió con su cumplimiento. Sin embargo, en agosto de 2016, la ONU presentó los resultados de los avances del país con resultados no favorables [17].

Si bien en enero del año 2021 se firmó una alianza entre la Vicepresidencia de la República y la ONG "Humanity and Inclusion", con el objetivo de impulsar el Sistema Nacional de Discapacidad [18] los retos del país para cumplir con los acuerdos pactados con la Convención de la ONU, siguen siendo un desafío.

Los derechos de las personas con discapacidad no sólo se definen en la Constitución Política de Colombia de 1991, en la Ley 30 de Educación Superior de 1990 y en la Ley 115 de 1994 (Ley General de Educación), sino en las ley 361 de 1997 que establece mecanismos de inclusión para la persona con discapacidad en todos los ámbitos humanos, la ley 1346 que aprueba la Convención sobre los Derechos de las Personas con Discapacidad y la ley 324 de 1996, por la cual se crean normas a favor de la población sorda. Adicionalmente, el decreto 2082 de 1996 reglamenta también la atención educativa a personas con discapacidad y en el contexto propio de la Universidad de Antioquia, el Acuerdo Académico 577 de 2021 establece el proceso de admisión para las personas sordas señantes usuarias de la Lengua de Señas Colombiana.

La accesibilidad, entendida como la facultad de brindar condiciones de igualdad para

todas las personas, sin distinción de sus características físicas, es fundamental para la inclusión social efectiva. La ciudad de Medellín se está transformando con el objetivo de crear oportunidades equitativas de acceso y participación en todos los ámbitos sociales, lo cual es un derecho inherente para el desarrollo pleno de cualquier individuo. Este proyecto de grado se justifica ante la urgencia de superar los obstáculos comunicacionales que enfrenta la comunidad sorda, obstáculos que se materializan en la falta de recursos accesibles para la traducción eficiente de la Lengua de Señas.

Las tecnologías actuales presentan desafíos significativos en la interpretación automática de la Lengua de Señas, tales como la variabilidad individual en el uso de los signos y la contextualización de los gestos, lo que implica una complejidad adicional en el procesamiento de la información por parte de los sistemas de inteligencia artificial [19] [20]. La adopción de técnicas de machine learning, y en particular del deep learning, promete superar estas barreras, permitiendo el desarrollo de modelos más sofisticados y precisos [20].

Este trabajo persigue la creación de modelos de machine learning que, fundamentados en los principios de la accesibilidad universal, faciliten la comunicación entre la comunidad sorda y oyente. El proyecto se propone como una iniciativa prometedora sin las barreras invisibles que tradicionalmente han separado a la comunidad sorda de la sociedad oyente. Resulta imperativo, por tanto, construir puentes que nos unan y promover una participación activa y equitativa en la sociedad a través de la tecnología. La tecnología actúa como un catalizador para este proyecto, buscando ser un referente de innovación, inclusión y progreso social.

IV. OBJETIVOS

A. Objetivo general

Desarrollar una plataforma para el reconocimiento de lengua de señas colombiana, que facilite la comunicación entre personas sordas y oyentes de la comunidad universitaria de la UdeA, mediante técnicas de machine learning.

B. Objetivos específicos

- Desarrollar un sistema de captura, etiquetado y análisis de datos sobre imágenes y videos de lengua de señas colombiana, mediante una plataforma web para almacenamiento de señas de manera estructurada y etiquetada.
- Diseñar un modelo de inteligencia artificial para identificar patrones en la Lengua de señas Colombiana.
- Producir corpus de videos de Lengua de Señas Colombiana para el entrenamiento de los modelos de inteligencia artificial, mediante un ambiente controlado.
- Implementar un módulo de despliegue para modelos de IA con versionamiento de datos y versionamiento de modelos.
- Validar en ambiente controlado el sistema de inteligencia artificial con personas de la comunidad sorda y la comunidad oyente de la Universidad de Antioquia.

V. MARCO TEÓRICO

Según amplios conocimientos en el campo de la lingüística, se reconoce que William Stokoe desempeñó un papel fundamental en el estudio de la lengua de señas norteamericana y su reconocimiento como un sistema lingüístico completo. En 1960, Stokoe se convirtió en el primer estudioso en acercar el análisis de la lengua de señas a la lingüística al demostrar que la lengua de señas cumple con los criterios necesarios para ser considerada una lengua. Sus investigaciones revelaron que la lengua de señas contiene rasgos convencionales, posee una gramática de combinación y una semántica propia, lo que proporciona una base sólida para analizar las señas como unidades mínimas y reconocer su doble articulación.

El trabajo de Stokoe se encuentra documentado en sus publicaciones y estudios relevantes. Por ejemplo, en su obra seminal "Sign Language Structure" [21], Stokoe examina en detalle los elementos fundamentales de la lengua de señas norteamericana y presenta su argumento sobre su naturaleza lingüística

Las contribuciones de Stokoe han sido ampliamente reconocidas y han sentado las bases para futuras investigaciones en el campo de la lingüística de la lengua de señas. Sus ideas han influido en el desarrollo de la disciplina y han contribuido a cambiar la percepción de la lengua de señas como un sistema completo y autónomo de comunicación.

La lengua de señas es la forma estándar de comunicación entre las personas con discapacidad del habla y la audición. La división regional de la lengua de señas ayuda a los usuarios a tener un método fácil para transmitir información. Dado que la mayoría de la población no comprende la lengua de señas, las personas con discapacidad del habla y la audición suelen depender de un intérprete humano. Sin embargo, la disponibilidad y la asequibilidad de un intérprete humano pueden no ser siempre posibles. La mejor alternativa sería un sistema de traducción automatizada que pueda leer e interpretar las señas y convertirlas en una forma comprensible para las personas oyentes. Este traductor reduciría la brecha de comunicación que existe entre las personas en la sociedad.

El proyecto se divide en 2 grandes secciones donde se pueden evidenciar diferentes etapas de desarrollo. La primer sección corresponde al modelamiento matemático con técni-

cas de aprendizaje automático, donde se estudia lo referente a la predicción de Lengua de Señas Colombiana. La segunda sección corresponde al desarrollo de una plataforma que tiene como objetivo crear un marco de trabajo donde se puedan crear diferentes corpus de forma colaborativa con el apoyo de diferentes entidades y profesionales con saberes relacionados a la Lengua de Señas Colombiana, así como también se realiza tratamiento y versionamiento de modelos de aprendizaje automático relacionados a distintos casos de uso en el marco de la Lengua de Señas Colombiana.

A. Modelos de aprendizaje de máquina para traducción de Lengua de Señas

El SLR (Sign Language Recognizer) debe ser entrenado con muchos datos de lengua de señas para una conversión fluida e ininterrumpida de la lengua de señas. Cada gesto creado hasta ahora tiene un significado y una aplicación específicos. Cada conjunto de señas utilizado en todo el mundo es rico en gramática y vocabulario. El SLR se puede considerar como un modelo de Interacción Persona-Computadora (HCI) modificado, donde el sistema puede leer y procesar el movimiento de las manos, las poses y los gestos. Dichos modelos abrirán el camino hacia una comunicación sin barreras.

En el vasto panorama tecnológico que delinea nuestro proyecto, la aplicación de Redes Neuronales Artificiales (RNAs) se presenta como un pilar conceptual de trascendental importancia. Inspiradas por la asombrosa complejidad y adaptabilidad del sistema nervioso humano, las RNAs constituyen una rama del aprendizaje automático que emula la capacidad del cerebro para procesar información, reconocer patrones y adaptarse a diversas circunstancias [22].

1) Redes Neuronales Recurrentes Las redes neuronales recurrentes, también conocidas por sus siglas en inglés como RNNs, constituyen un grupo de redes neuronales diseñadas para el procesamiento de datos secuenciales [23]. Una RNN es una red que está especializada en procesar secuencias de valores $x^{(1)}, \ldots, x^{(\tau)}$.

La representación matemática de las RNN de Elman [24], se da de la siguiente forma de acuerdo a la Ilustración 2 está dada por:

$$a^{(t)} = b + Vh^{(t-1)} + Ux^{(t)}, \tag{1}$$

$$h^{(t)} = \tanh(a^{(t)}), \tag{2}$$

$$o^{(t)} = c + Wh^{(t)} \tag{3}$$

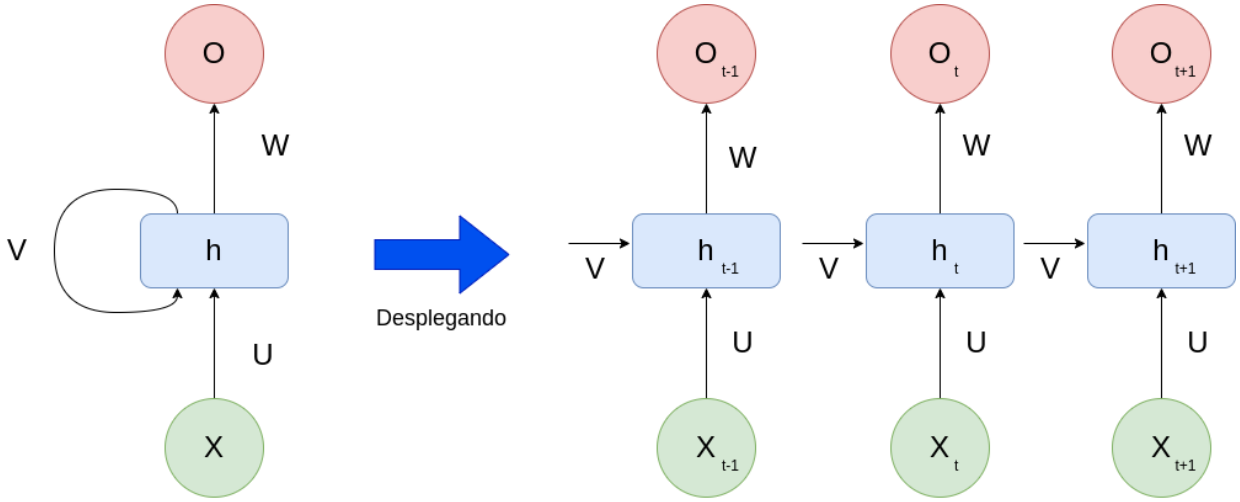

Fig. 2. Ilustración de la representación de las Redes Neuronales Recurrentes de Elman

Donde V son los pesos de la matriz en el bucle de retroalimentación. U son los pesos de la matriz en las entradas y W son los pesos de la matriz que conecta los estados de la red con la capa de salida. b y c son los vectores de sesgos correspondientes para las capas ocultas y la salida respectivamente. La capa de salida corresponde a la aplicación de la función de activación para los valores de $o^{(t)}$.

2) Propagación hacia atrás En redes neuronales que procesan la información de entrada a salida, se introduce un dato x que pasa por distintas capas de la red hasta generar una salida $\hat{y}$. A este proceso se le llama propagación hacia adelante. Durante el aprendizaje, este flujo sigue hasta que se evalúa el desempeño del modelo con una medida de error. Para mejorar el modelo, se utiliza un método llamado propagación hacia atrás [23] que ajusta la red basándose en este error, enviando la señal de ajuste en sentido inverso para afinar las predicciones del modelo.

3) Propagación hacia atrás através del tiempo Similarmente como se realiza la propagación hacia atrás se tiene el algoritmo de propagación hacia atrás que realiza una optimización de los gradientes teniendo en cuenta la estructura de secuencia [25].

4) MediaPipe MediaPipe es un marco de trabajo desarrollado por Google Research que facilita la creación, el procesamiento y la evaluación de aplicaciones que procesan entradas perceptuales, como el audio y el vídeo. Permite a los desarrolladores combinar componentes de percepción ya existentes o nuevos en prototipos y refinarlos en aplicaciones pulidas y multiplataforma. Es configurable para gestionar recursos de manera eficiente, manejar sincronización de datos de series temporales como fotogramas de audio y vídeo, y medir el rendimiento y consumo de recursos [26].

5) Long Short Term Memory (LSTM) RNN Uno de los inconvenientes de las Redes Neuronales Convolucionales es la inhabilidad de comprender las dependencias a largo plazo. Uno de estos problemas se puede ver con el problema del desvanecimiento del gradiente [27]. Con las redes neuronales recurrentes LSTM se plantea una forma de almacenar información durante grandes intervalos de tiempo utilizando un método basado en el gradiente que puede dar una solución al desvanecimiento del gradiente. [28]

La Ilustración 3 representa lo siguiente:

1. Estado interno de la celda de memoria C_{t-1}: Este es el estado de la memoria de la celda en el tiempo anterior $t-1$.

2. Estado oculto H_{t-1}: Representa la salida de la celda de memoria en el tiempo anterior, que también se utiliza en cálculos actuales.

3. Entrada X_t: Es el dato de entrada en el momento actual t.

4. Puerta de olvido F_t: Decide qué parte del estado de la memoria previo será conservado. Se aplica una función de activación sigmoidea (σ) para obtener valores entre 0 y 1.

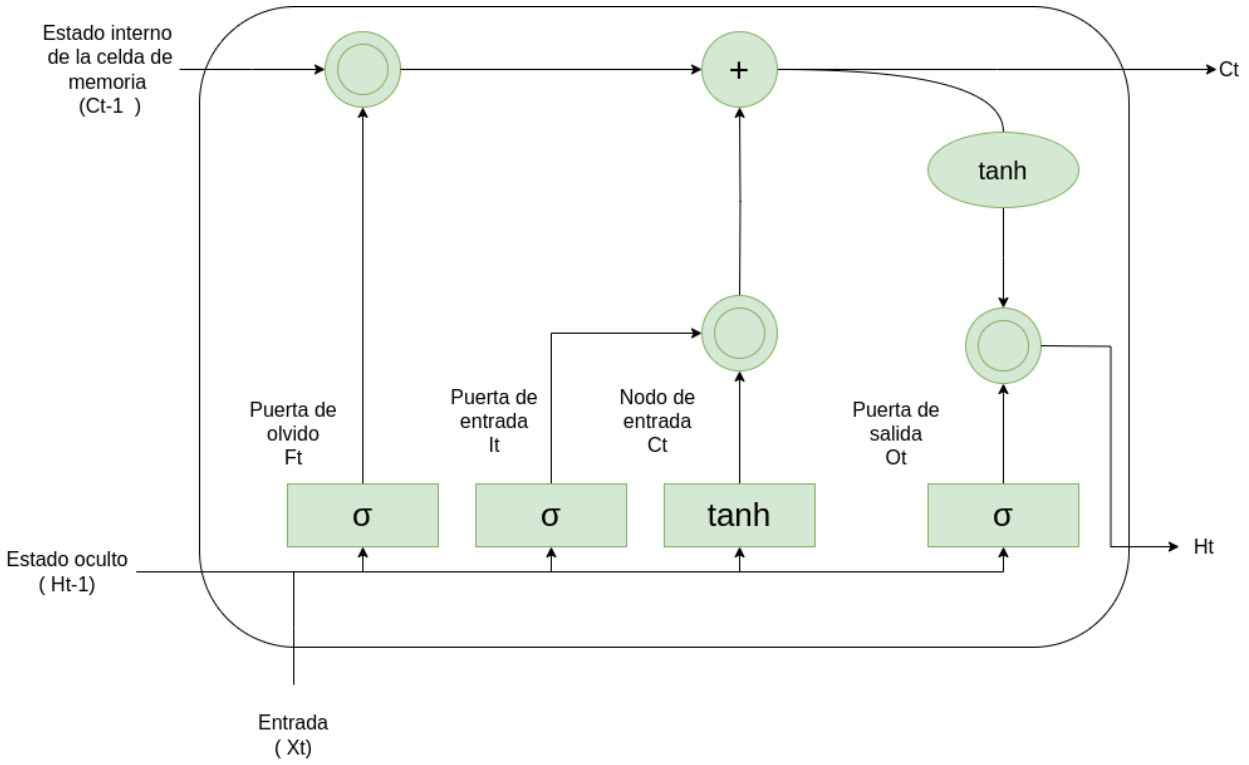

Fig. 3. Ilustración de la arquitectura y el flujo de información de una célula individual de una red neuronal recurrente del tipo LSTM

5. Puerta de entrada I_t y Nodo de entrada $\tilde{C}_t$: La puerta de entrada decide qué información nueva se añadirá, mientras que el nodo de entrada crea un vector de nuevos valores candidatos a ser añadidos al estado de la celda, usando la función de activación tangente hiperbólica (tanh).

6. Actualización del estado de la celda C_t: El estado interno de la celda se actualiza combinando la información de la puerta de olvido y la puerta de entrada de manera element-wise, lo cual significa que se realiza una operación elemento a elemento.

7. Puerta de salida O_t y Salida del estado oculto H_t: La puerta de salida decide qué parte del estado de la celda actualizado será utilizado en el estado oculto. La función de activación tanh se aplica al estado de la celda y luego se multiplica por la activación de la puerta de salida para determinar la salida final H_t.

Las operaciones element-wise se representan con círculos y las funciones de activación se muestran como cajas con σ o tanh, indicando qué tipo de función se aplica. El proceso completo es el siguiente:

- Los estados C_{t-1} y H_{t-1}, junto con la entrada actual X_t, son procesados por las puertas de olvido y entrada para decidir qué información se descarta y qué información nueva se añade.
- La puerta de olvido multiplica su salida por el estado de la celda anterior para olvidar selectivamente los componentes antiguos.
- Paralelamente, la puerta de entrada y el nodo de entrada trabajan juntos para crear y añadir nuevos candidatos de información al estado de la celda.
- Estos dos resultados se suman para formar el nuevo estado de la celda C_t.
- Finalmente, la puerta de salida utiliza el estado de la celda actualizado (pasando por una función tanh para escalar los valores entre -1 y 1) para determinar qué información se enviará como salida H_t .

6) Gated Recurrent Unit (GRU) RNN En 2014 se propone un modelo de procesamiento de lenguaje que utiliza dos redes neuronales recurrentes, una para transformar secuencias de texto en vectores numéricos y otra para realizar el proceso inverso. El objetivo del modelo es mejorar la forma en que las máquinas entienden y traducen el lenguaje al optimizar la precisión con la que se puede predecir una secuencia de texto a partir de otra. Al entrenar las dos redes juntas, el modelo se vuelve eficaz en capturar las sutilezas de significado y estructura del lenguaje. Este enfoque mejora los sistemas de traducción automática al proporcionar una nueva característica basada en la probabilidad de que ciertas frases aparezcan juntas, lo que resulta en traducciones más precisas y coherentes. Además, el modelo no solo traduce sino que también aprende representaciones ricas y detalladas de las frases, captando aspectos semánticos y sintácticos importantes, lo cual es un gran avance en el campo del procesamiento del lenguaje natural [29]. En la Ilustración 4 se observa la arquitectura de una GRU con sus respectivas componentes.

- Entrada: X_t

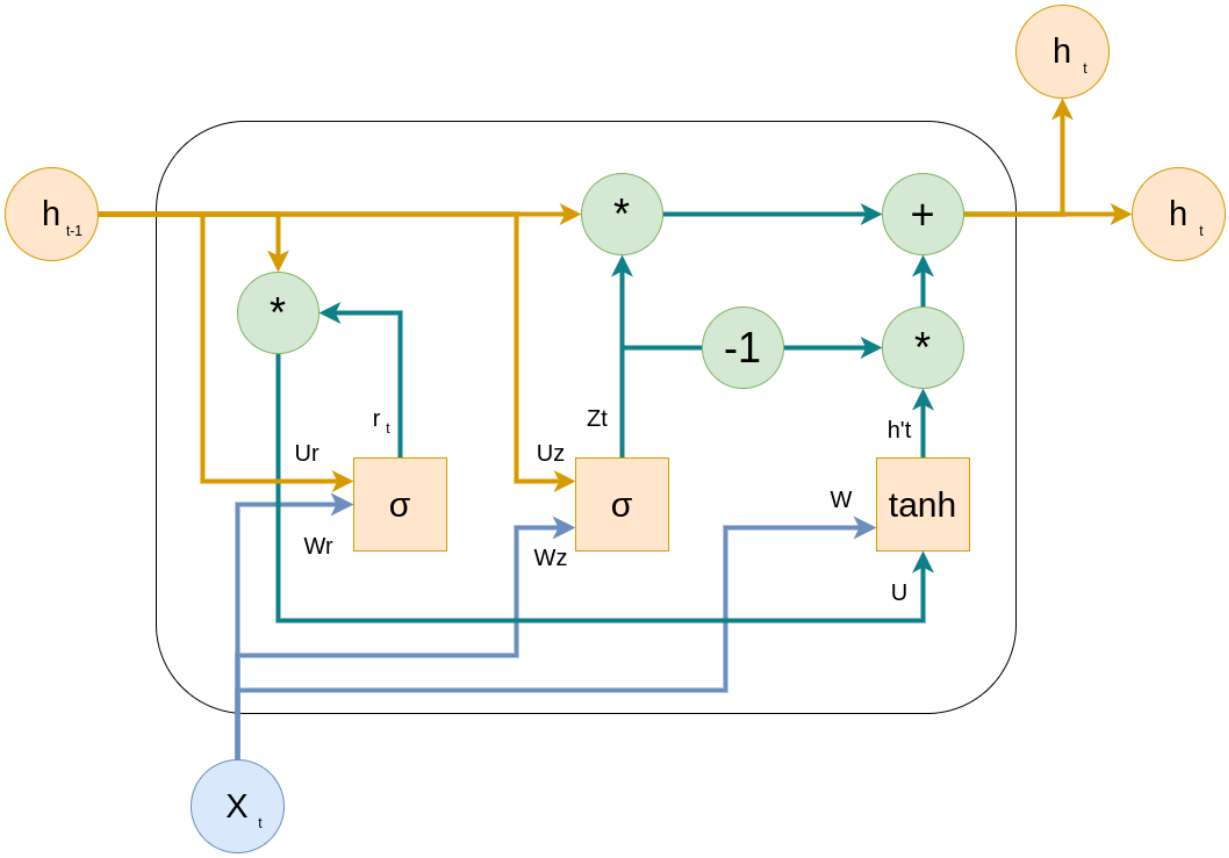

Fig. 4. Ilustración de la arquitectura de una unidad de compuerta recurrente, conocida como GRU

- Estado anterior: h_{t-1}
- Puertas de actualización: u_r, u_z
- Puerta de restablecimiento: $r_t = \sigma(W_r X_t + U_r h_{t-1})$
- Puerta de actualización: $z_t = \sigma(W_z X_t + U_z h_{t-1})$
- Candidato a estado oculto: $\tilde{h}'_t = \tanh(W X_t + U(r_t * h_{t-1}))$
- Estado oculto actualizado: $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}'_t$

Las funciones de activación utilizadas son la sigmoide (σ) y la tangente hiperbólica (tanh). Las GRUs son eficientes en la captura de dependencias temporales de diferentes longitudes gracias a sus puertas de actualización y restablecimiento, lo que les permite recordar o descartar información de manera adaptativa, mejorando así el rendimiento en tareas de modelado secuencial.

7) Validación Cruzada K Fold estratificado El método "Stratified K-Fold" es una técnica de validación cruzada que se caracteriza por dividir los datos en "K" grupos o "folds", asegurando que cada fold tenga aproximadamente la misma distribución de clases que los datos originales. Esto es particularmente útil en conjuntos de datos desequilibrados [30].

8) Métricas de evaluación

- Precisión: es una medida de rendimiento que se utiliza en clasificación y se refiere a la proporción de resultados positivos identificados correctamente (verdaderos positivos) entre el total de resultados identificados como positivos (la suma de verdaderos positivos y falsos positivos) [31]. En la Ecuación 4 podemos observar la métrica de precisión.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Donde TP representa los verdaderos positivos y FP representa los falsos positivos.

- Sensibilidad: también conocida como *recall* o tasa de verdaderos positivos, es una medida de cuántos de los casos positivos reales fueron identificados correctamente por el modelo. Se define como la cantidad de verdaderos positivos (TP) dividida por la cantidad total de casos positivos reales, que es la suma de verdaderos positivos (TP) y falsos negativos (FN) [31]. La sensibilidad se formula de la siguiente manera:

$$Sensibilidad = \frac{TP}{TP + FN} \tag{5}$$

Donde TP son los verdaderos positivos y FN son los falsos negativos

- F1: es una medida de la precisión de un test. Es el promedio armónico de la precisión y la sensibilidad, donde la precisión es la proporción de verdaderos positivos entre la suma

de verdaderos positivos y falsos positivos, y la sensibilidad es la proporción de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos [31]. La métrica F1 se formula de la siguiente manera:

$$F1 = 2 \times \frac{\text{Precisión} \times \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}} \tag{6}$$

B. Herramientas para el desarrollo de sistemas web

1) FastAPI FastAPI, un marco de aplicación web moderno y de alto rendimiento, se ha beneficiado significativamente del reciente aumento en la popularidad de Python. Este auge se debe principalmente a las bibliotecas de Python en el ámbito de las aplicaciones de ciencia de datos. Sin embargo, Python también se utiliza extensamente en el desarrollo de aplicaciones web, gracias a la abundancia de sus marcos para aplicaciones web. FastAPI se destaca en este campo por su facilidad de uso y su capacidad para facilitar el desarrollo rápido, manteniendo un alto rendimiento y ofreciendo características avanzadas para aplicaciones web modernas [32].

2) Kubernetes Kubernetes, al igual que el sistema Borg de Google [33], es un sistema de gestión de clústeres diseñado para ejecutar y coordinar aplicaciones distribuidas en una gran cantidad de máquinas. Su objetivo principal es optimizar la utilización y el rendimiento, a través de métodos como el control de admisión, el empaquetamiento eficiente de tareas y el aislamiento de rendimiento a nivel de proceso. Además, Kubernetes ofrece soporte para aplicaciones de alta disponibilidad, incorporando características que minimizan el tiempo de recuperación ante fallos y políticas para reducir la probabilidad de fallos correlacionados. Al igual que Borg, Kubernetes simplifica la gestión de aplicaciones en entornos distribuidos, proporcionando un lenguaje de especificación de trabajos declarativo, integración con servicios de nombres y herramientas para el monitoreo y análisis del sistema.

3) Shape Up La metodología Shape Up, desarrollada por la compañía de software Basecamp [34], es un enfoque para la gestión de proyectos y el desarrollo de software que se

distingue de las metodologías tradicionales como Scrum o Agile. Su enfoque se basa en ciclos de trabajo fijos de seis semanas, seguidos de dos semanas de enfriamiento.

- Ciclos de Trabajo Definidos: Shape Up propone ciclos de trabajo de seis semanas seguidos de dos semanas de enfriamiento. Durante las seis semanas, los equipos se enfocan intensamente en un conjunto limitado de tareas o proyectos. Las dos semanas de enfriamiento permiten tiempo para la exploración, la planificación y la recuperación.
- Formación de Equipos y Autonomía: Los equipos se forman para cada ciclo y se les otorga autonomía para tomar decisiones sobre cómo ejecutar el proyecto. Esto contrasta con enfoques como Scrum, donde las tareas se asignan y se supervisan más frecuentemente.
- Especificación de Proyectos ("Shaping"): Antes de un ciclo, se realiza el "shaping" de los proyectos. Esto involucra definir el alcance y los límites del proyecto, pero sin detallar excesivamente las tareas específicas. Se establece lo suficiente para que el equipo tenga claridad sobre qué debe entregarse y qué se considera fuera del alcance.
- Apuestas en lugar de Planificación: En lugar de planificar todas las tareas de un proyecto, Shape Up utiliza un enfoque de .ªpuestas". Se eligen proyectos para un ciclo basándose en su valor percibido y viabilidad, no mediante una planificación detallada y a largo plazo.
- Sin Estimaciones de Tiempo Detalladas: Shape Up evita las estimaciones detalladas de tiempo para tareas individuales, argumentando que son inherentemente inexactas y llevan a expectativas erróneas.
- Revisiones y Ajustes: Después de cada ciclo, los equipos revisan su progreso y ajustan sus enfoques para el próximo ciclo basándose en lo aprendido.

VI. METODOLOGÍA

Para alcanzar los objetivos planteados en este estudio se adoptaron dos metodologías complementarias: una basada en la preparación y el modelado de datos para la inteligencia artificial, y otra centrada en el desarrollo de una plataforma iterativa para la Lengua de Señas Colombiana (LSC), utilizando la metodología Shape Up.

Para el desarrollo de la plataforma, se aplica la metodología Shape Up. Este enfoque se caracteriza por ciclos de desarrollo de seis semanas, denominados 'ciclos de modelado', seguidos de períodos de enfriamiento de dos semanas. Durante los ciclos de modelado, se define un conjunto de tareas clave enfocadas en la creación y mejora de la plataforma para la subida y etiquetado de videos. Estas tareas son delimitadas antes de cada ciclo y permiten soluciones creativas dentro de los 'ámbitos de trabajo'definidos.

Shape Up promueve la flexibilidad y la adaptabilidad, cualidades esenciales para responder a los aprendizajes y ajustes que surgen durante el desarrollo del modelo de IA. Además, esta metodología facilita un progreso significativo en componentes clave del proyecto en cada ciclo, con una planificación y ejecución enfocadas que permiten adaptarse a los desafíos únicos de un proyecto iterativo como este.

La combinación de estas metodologías garantiza un enfoque riguroso y sistemático para el manejo y análisis de datos y el modelado de IA, mientras que Shape Up proporciona un marco flexible y adaptable para el desarrollo continuo de la plataforma. Esta integración metodológica es esencial para el éxito del proyecto, permitiendo la colaboración efectiva entre un equipo multidisciplinario que incluye expertos en LSC, desarrolladores de software, y especialistas en inteligencia artificial.

A. Creación del corpus

Para la creación del corpus utilizado en el entrenamiento de los datos de este estudio, se implementó una metodología detallada y sistemática, centrada en la captura de señas específicas de la Lengua de Señas Colombiana (LSC), orientadas al contexto del Programa Domo de la Universidad de Antioquia, el cual es un modelo de emprendimiento social im-

plementado desde la Dirección de Bienestar Universitario que permite la venta productos al interior de las sedes de la universidad vinculando estudiantes con necesidades socioeconómicas difíciles brindándoles así un trabajo que favorezca la permanencia y evitando así la deserción. El proceso de recolección de los videos se llevó a cabo con la participación de 4 personas sordas, cada una aportando muestras de cinco señas diferentes: "Tinto", "Aromática", "¿Qué necesitas?", "Con gusto" y "Hola".

La metodología se diseñó para capturar la riqueza y variabilidad de la LSC en un entorno controlado. Cada seña fue grabada en tres ángulos diferentes (grabación a 3 cámaras) para facilitar el reconocimiento preciso por parte del modelo de IA. Estos ángulos fueron cuidadosamente elegidos para capturar los aspectos clave de cada seña, como la orientación de las manos y la expresión facial.

Para aumentar la robustez y el poder de generalización del modelo, se introdujeron variables en el entorno de grabación. Esto incluyó 4 cambios en las vestimentas de los participantes y la adición de elementos considerados distractores en el fondo. La intención era simular condiciones reales y diversas, donde elementos inesperados en el entorno podrían estar presentes. Esto es especialmente importante en aplicaciones de IA, donde la capacidad del modelo para funcionar de manera fiable en diversos entornos es crítica.

El proceso de grabación se realizó en sesiones estructuradas, donde cada participante realizaba las señas en secuencia, asegurando la consistencia y calidad de las muestras. Tras la grabación, se procedió a la catalogación y etiquetado de los videos, lo cual fue crucial para la fase de entrenamiento del modelo de IA. Este etiquetado incluyó no solo la seña específica, sino también detalles sobre el ángulo de grabación y las variaciones en el entorno. En total se grabaron 300 videos por seña para un total de 1500 videos (Ver Apéndice A).

B. Arquitectura de la plataforma para la captura de datos

Para que este proyecto sea sostenible en el tiempo se diseñó una aplicación[1] donde se pueden almacenar, etiquetar, manipular y descargar señas para su posterior uso en diversos

[1]Repositorio del proyecto de Plataforma de Captura de Datos: https://github.com/SignAIUdeA/SignAI/

modelos. Asimismo, la aplicación también cuenta con un módulo de carga de modelos para que se pueda tener acceso a diversos modelos que se realicen con los datos capturados. El objetivo principal de la plataforma es promover la investigación en la traducción de Lengua de Señas Colombiana integrando diversos entes que enriquezcan la tecnología orientada a la inclusión.

En la Ilustración 5 se detalla la estructura y tecnologías utilizadas para crear una solución eficaz y escalable. La aplicación se compone de un backend desarrollado en FastAPI, un frontend construido con React, y se utiliza Kubernetes para su despliegue y gestión.

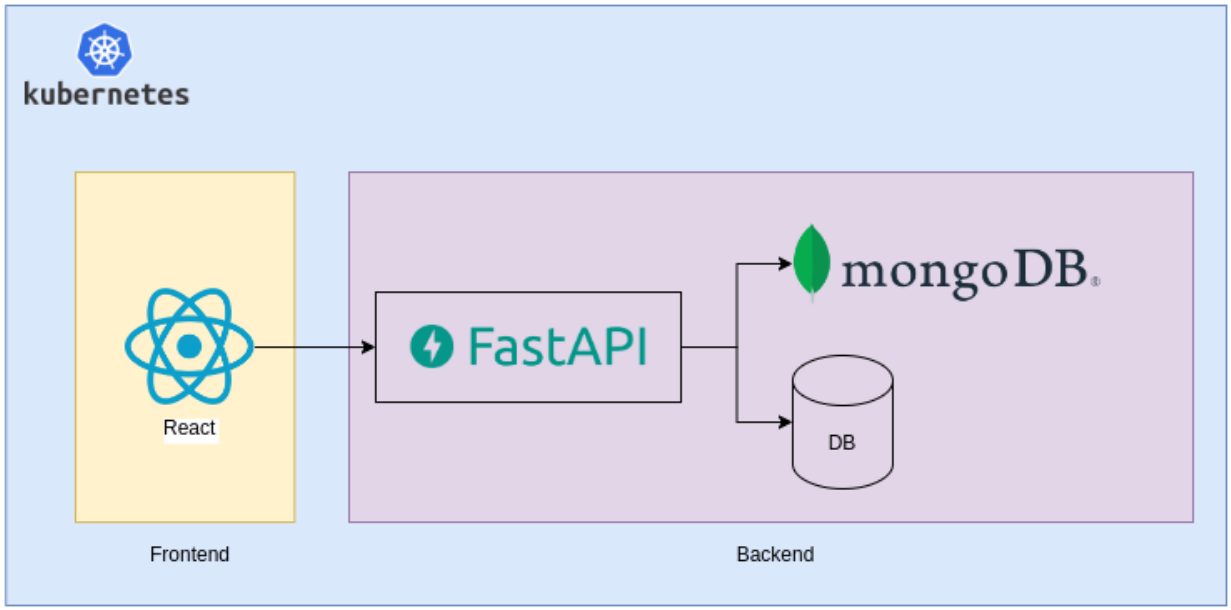

Fig. 5. Arquitectura de la plataforma para captura y etiquetamiento de datos

El backend de la aplicación, implementado con FastAPI, proporciona una base sólida y de alto rendimiento. FastAPI es conocido por su velocidad y eficiencia, así como por su capacidad para manejar solicitudes asincrónicas, lo que es crucial para las operaciones intensivas en datos de esta aplicación. Este backend gestiona todas las operaciones de procesamiento de datos, como almacenar, recuperar y etiquetar las señas recopiladas. Además, está diseñado para facilitar la integración con sistemas de aprendizaje automático, permitiendo así una expansión y mejora continua de los modelos de IA.

En cuanto al frontend, se seleccionó React por su flexibilidad y su eficiente manejo del estado de la interfaz de usuario. Esto es esencial para proporcionar una experiencia de usuario fluida y dinámica, permitiendo a los usuarios interactuar sin problemas con la aplicación para realizar tareas como la carga de videos, el etiquetado de señas y la administración de datos. La interfaz de usuario de React está diseñada para ser intuitiva y accesible, facilitando la colaboración entre equipos interdisciplinarios, incluyendo expertos en LSC, lingüistas, y desarrolladores de software.

Para el despliegue de la aplicación, se utiliza Kubernetes. Esta elección se debe a la capacidad de Kubernetes para gestionar y escalar automáticamente la infraestructura de la aplicación, lo que es vital para manejar cargas de trabajo variables y garantizar la alta disponibilidad del sistema. Kubernetes ofrece una solución robusta y flexible para el despliegue, permitiendo la distribución eficiente de recursos y facilitando la actualización y el mantenimiento continuo de la aplicación sin interrumpir su funcionamiento.

La arquitectura de esta aplicación está diseñada para ser escalable y adaptable, permitiendo su uso en distintos casos de uso y facilitando la colaboración de equipos en diferentes lugares de Colombia. La combinación de FastAPI, React y Kubernetes crea un ecosistema robusto y flexible, adecuado para las necesidades de un proyecto ambicioso y de largo alcance como este, destinado a mejorar la comunicación y la inclusión de la comunidad sorda en Colombia.

Se tienen 4 módulos principales con sus respectivos roles para la gestión y los procesos referentes al proyecto, uno de los módulos es para la gestión de usuarios (Ver Anexo B), donde se pueden ver las personas que participan en el proceso que a su vez se dividen en 3 roles “Administrador” que puede acceder a la plataforma y editar usuarios y manipular procesos administrativos, “Auxiliar de LSC” que se encarga etiquetar videos de señas, para posteriores procesos de investigación y “Profesional de LSC” que se encarga de etiquetar videos etiquetados, y también de revisar y corregir videos que han sido etiquetados por los auxiliares.

Otro módulo consta de subir señas, esto permitirá almacenar los videos en una base de datos y guardar metadata de los mismos para mantener controles de calidad.

El tercer módulo consta del análisis de los datos almacenados, la principal ventaja de éste módulo es el hecho de que se puedan realizar decisiones que permitan priorizar la producción de unas señas sobre otras para mejorar el desempeño de los modelos de inteligencia artificial que surjan a partir de los datos de la plataforma.

El último módulo tiene como objetivo realizar gestión de modelos de investigación enfocados en Lengua de Señas Colombiana; así, se podrán tener modelos realizados con cierto conjunto de datos para diversos casos de uso que se podrán utilizar para futuras investigaciones o simplemente replicar para usos cotidianos en ciertos contextos.

VII. IMPLEMENTACIÓN

La implementación del proyecto[2] se estructura en cuatro fases fundamentales:

- Preprocesamiento de datos
- Selección de conjuntos de datos para entrenamiento, validación y prueba
- Optimización de hiperparámetros
- Selección del modelo óptimo

Esta estructura metodológica se inicia con la recolección y preparación de un conjunto diverso de videos en Lengua de Señas Colombiana (LSC), abarcando procesos de curación de datos que garantizan la calidad y pertinencia del conjunto de datos. La fase de preprocesamiento se enfoca en la normalización y extracción de características relevantes para el aprendizaje, estableciendo una base sólida para el modelamiento. En la etapa subsiguiente, se procede a la selección de los conjuntos de datos destinados al entrenamiento, validación y prueba, configurando una arquitectura que facilita la evaluación objetiva del rendimiento de los modelos. Esta selección es crucial para evitar el sobreajuste y garantizar que el modelo generalice bien a datos no vistos. La búsqueda de hiperparámetros se realiza mediante técnicas avanzadas de optimización, como la validación cruzada, que permiten identificar la configuración más efectiva para maximizar la precisión del modelo. Finalmente, se selecciona el modelo más adecuado basado en su rendimiento, teniendo en cuenta métricas de evaluación rigurosas.

[2]Repositorio del proyecto de IA: https://github.com/SignAIUdeA/AI/

A. Preprocesamiento de datos

Para complementar el enfoque metodológico descrito, se procedió a la implementación de un riguroso proceso de preprocesamiento de los datos, utilizando el marco de trabajo MediaPipe. Esta herramienta, desarrollada por especialistas en visión por computadora, se empleó para la extracción sistemática de puntos clave (keypoints) de la cara, las manos y la postura de los sujetos en los videos.

En primer lugar, se realizó una normalización y estandarización de los videos. Este proceso fue esencial para mitigar posibles sesgos asociados a variables como el ángulo de la cámara, el género, la edad, el color de piel y los elementos de vestuario de los participantes. La aplicación de MediaPipe permitió una detección y segmentación eficiente y precisa de puntos clave corporales. Específicamente, se extrajeron keypoints representativos de las manos, el rostro y la postura general del cuerpo, elementos críticos en la comunicación a través de la Lengua de Señas Colombiana.

La normalización mediante MediaPipe facilitó la reducción de variaciones no esenciales en los datos, mejorando así la homogeneidad y la calidad del conjunto de datos procesado. Este paso fue crucial para asegurar la validez y confiabilidad de los modelos de aprendizaje automático posteriormente entrenados. La exactitud en la extracción de características a través de MediaPipe proporcionó una base sólida para el análisis detallado y la interpretación de los gestos y movimientos en la LSC, garantizando una interpretación precisa y efectiva de estos elementos vitales para la comunicación en lengua de señas.

B. Selección de conjuntos de entrenamiento, validación y test

La selección de conjuntos de entrenamiento, validación y test es un aspecto crucial en el diseño de estudios de aprendizaje automático, especialmente en contextos donde la cantidad de datos y participantes es limitada. En este caso, el estudio contó con la participación de solo cuatro individuos, lo que planteó desafíos significativos en términos de evitar sesgos y asegurar la generalización del modelo.

Para abordar estas limitaciones, se adoptó un enfoque meticuloso en la división de los

conjuntos de datos. Se decidió asignar a uno de los participantes, junto con todos sus videos y señas correspondientes, exclusivamente al conjunto de test. Esta estrategia garantiza que el modelo sea evaluado en un conjunto de datos completamente independiente, no visto durante la fase de entrenamiento o validación, mitigando así el riesgo de sobreajuste y asegurando una evaluación más realista de la capacidad generalizadora del modelo.

Para los datos restantes, correspondientes a los otros tres participantes, se implementó un método de validación cruzada K-fold estratificado, utilizando un k de 5. Este enfoque permite que cada uno de los subconjuntos de datos sea utilizado tanto para entrenamiento como para validación en distintas iteraciones. La estratificación en este contexto es fundamental para mantener la proporción representativa de las diferentes señas realizadas por los participantes en cada fold, considerando que cada individuo podría tener un estilo único o preferencias en las señas, lo cual podría introducir sesgos si no se maneja adecuadamente.

La aplicación del K-fold estratificado asegura que el modelo no solo aprenda las características específicas de un solo individuo, sino que también adquiera una capacidad general para interpretar la LSC a partir de un espectro más amplio de manifestaciones. Este enfoque fortalece la robustez del modelo y su capacidad para generalizar a partir de un conjunto limitado de datos, un desafío común en estudios con muestras pequeñas.

C. Búsqueda de mejores hiperparámetros para los modelos

La implementación de una validación cruzada utilizando Keras Tuner en el estudio presenta una metodología avanzada para la optimización de hiperparámetros en modelos de aprendizaje automático. Este enfoque se centró en realizar una serie de experimentos para identificar el conjunto de hiperparámetros que mejor generaliza los datos de la Lengua de Señas Colombiana (LSC), un aspecto crucial para el éxito del modelo en un contexto de clasificación multiclase.

En el marco de este estudio, se llevaron a cabo cinco experimentos aleatorios utilizando Keras Tuner. Cada experimento varió los hiperparámetros del modelo de forma aleatoria dentro de un rango predefinido. Esta metodología permite explorar un amplio espacio de po-

sibles configuraciones, aumentando las probabilidades de identificar una combinación óptima que mejore la capacidad predictiva del modelo. Los hiperparámetros ajustados incluyeron, entre otros, la tasa de aprendizaje, el número y tamaño de las capas en la red neuronal, y los parámetros de regularización.

La métrica objetivo seleccionada para la optimización mediante Keras Tuner fue el F1 Score. Esta métrica es especialmente adecuada para problemas de clasificación con múltiples clases, como es el caso en este estudio, que cuenta con cinco categorías distintas de señas. El F1 Score proporciona un balance entre la precisión y la sensibilidad del modelo, lo cual es crucial en un contexto donde las clases pueden estar desbalanceadas o donde la precisión en la clasificación de cada clase es igualmente importante.

Para la optimización de los hiperparámetros, se utilizó un enfoque de optimización bayesiana, el cual busca maximizar el F1 Score. La optimización bayesiana es particularmente útil en este contexto, ya que permite una búsqueda más eficiente en el espacio de hiperparámetros, basándose en la probabilidad de mejora dadas las observaciones anteriores. Esto contrasta con métodos de búsqueda más tradicionales que podrían requerir un número mayor de iteraciones para alcanzar resultados óptimos [35].

D. Definición de modelos

En la definición de modelos para este estudio, se optó por utilizar redes neuronales que son particularmente adecuadas para el procesamiento de datos secuenciales, como son las Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU) . La elección de estas redes se fundamenta en las características inherentes de los datos y los objetivos del estudio.

Las redes LSTM y GRU pertenecen a la categoría de redes neuronales recurrentes (RNN). Estas redes son idóneas para trabajar con datos secuenciales, como los videos de Lengua de Señas Colombiana, donde cada señal o gesto se puede considerar como una secuencia de movimientos a lo largo del tiempo. Lo que hace especiales a las LSTM y GRU es su capacidad para recordar información a lo largo de estas secuencias, lo cual es crucial para entender contextos y patrones a lo largo del tiempo en los datos de señas.

Estas redes recurrentes están relacionadas con los procesos Markovianos, un concepto en teoría de probabilidades que se refiere a procesos estocásticos donde el futuro es independiente del pasado, dado el presente. Esta propiedad se alinea bien con la naturaleza de las señas en LSC, donde el significado de un gesto puede depender fuertemente de los gestos anteriores y posteriores en una secuencia.

La combinación de estas arquitecturas de red (LSTM y GRU) proporciona un enfoque robusto y versátil para el análisis de señas en LSC. Mientras las redes LSTM y GRU permiten modelar la dependencia temporal y contextual en los datos. Juntas, estas redes ofrecen una solución integral para capturar y analizar la rica estructura secuencial y temporal de la comunicación a través de la Lengua de Señas Colombiana.

E. Selección del mejor modelo

En la fase de selección del modelo óptimo para la traducción de la Lengua de Señas Colombiana (LSC), se implementó un enfoque riguroso basado en criterios de precisión, eficiencia y capacidad de generalización. Este proceso involucró la evaluación comparativa de diferentes arquitecturas de redes neuronales, incluyendo Long Short-Term Memory (LSTM) y Gated Recurrent Unit (GRU), todas previamente identificadas por su pertinencia en el procesamiento de datos secuenciales.

La evaluación se centró inicialmente en las características intrínsecas de cada arquitectura. Las redes LSTM y GRU, ambas variantes de redes neuronales recurrentes (RNN), fueron examinadas por su habilidad para manejar dependencias temporales largas, una característica esencial para interpretar secuencias de gestos en la LSC. Estas redes tienen la particularidad de preservar información relevante a lo largo del tiempo, lo cual es crucial para comprender contextos y patrones en secuencias de señas.

La selección del modelo más adecuado se realizó mediante una serie de experimentos, donde cada arquitectura fue entrenada y validada con un conjunto de datos representativo de la LSC. Los criterios de evaluación incluyeron la precisión del modelo en la clasificación correcta de gestos y secuencias de señas, su eficiencia computacional y la capacidad de

generalizar a nuevos ejemplos no vistos durante el entrenamiento.

Se utilizó una metodología de validación cruzada para garantizar que la evaluación fuera robusta y confiable. Además, se implementaron métricas de rendimiento como la precisión, la sensibilidad, la especificidad y la métrica F1 para una evaluación integral de cada modelo.

La elección del modelo más apropiado se basó en un equilibrio entre precisión y eficiencia, así como en la capacidad del modelo para capturar las características esenciales de la LSC. Este enfoque metódico asegura que el modelo seleccionado sea no solo técnicamente sólido, sino también efectivo y eficiente en la traducción de la Lengua de Señas Colombiana.

VIII. RESULTADOS

En el marco del desarrollo de un modelo de aprendizaje automático enfocado en la traducción de la Lengua de Señas Colombiana, se llevaron a cabo 25 experimentos para cada una de las arquitecturas de redes neuronales recurrentes exploradas: LSTM y GRU. A partir de los resultados obtenidos en dichos experimentos, se procedió a realizar un análisis comparativo profundo. Este análisis tuvo como objetivo examinar detalladamente el desempeño de cada uno de los modelos frente a la tarea de clasificación de cinco señas específicas pertenecientes a la Lengua de Señas Colombiana.

Se tiene la siguiente codificación para cada una de las clases:

- Café: 1
- Con gusto: 2
- Qué necesitas: 3
- Aromática: 4
- Hola: 5

A. Arquitectura basada en GRU

En la Tabla II se presentan los resultados obtenidos de la optimización de hiperparámetros en modelos que utilizan unidades recurrentes cerradas (Gated Recurrent Unit o

GRU) para la clasificación de de señas de Lengua de Señas Colombiana (LSC).

La optimización Bayesiana ha permitido una búsqueda eficiente en el espacio de hiperparámetros, como lo demuestra la mejora progresiva y la convergencia hacia configuraciones de alto rendimiento. Los resultados de la Tabla II sugieren que los modelos GRU son capaces de capturar las dependencias temporales largas y las secuencias de gestos de la LSC con un alto grado de precisión.

En la Tabla II se varían los hiperparámetros de las dos capa GRU, y la capa Densa por medio de la optimización bayesiana. Luego de observar los resultados de la variación de hiperparámetros en la Tabla II se define la arquitectura con mejores hiperparámetros en la Ilustración 6 donde se tienen 70 neuronas recurrentes en la primer capa, 30 neuronas recurrentes en la segunda capa, y 25 neuronas densas en la tercera capa. Teniendo en cuenta las dimensiones de la capa de entrada y la capa de salida, se tiene un total de 374.251 parámetros entrenables con un peso total de 1.43 MB.

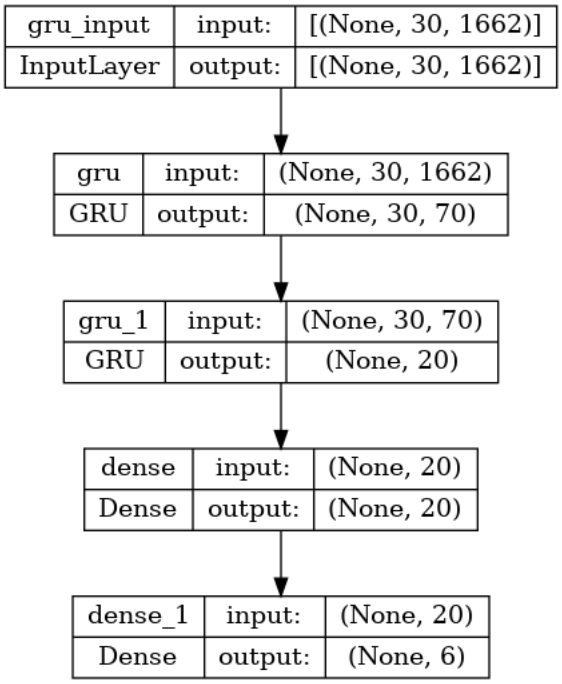

Fig. 6. Arquitectura GRU con mejores hiperparámetros

TABLA II
RESULTADOS DE EXPERIMENTOS DE MODELOS GRU CON DIFERENTES HIPERPARÁMETROS

Intento	Hiperparámetros	F1
1	GRU1: 70, GRU2: 20, Dense: 20	0.904231608
2	GRU1: 70, GRU2: 20, Dense: 20	0.95652169
3	GRU1: 70, GRU2: 20, Dense: 20	0.935122967
4	GRU1: 70, GRU2: 20, Dense: 20	0.951002121
5	GRU1: 70, GRU2: 20, Dense: 20	0.935267806
6	GRU1: 60, GRU2: 10, Dense: 30	0.928251088
7	GRU1: 60, GRU2: 10, Dense: 30	0.943946123
8	GRU1: 60, GRU2: 10, Dense: 30	0.92600894
9	GRU1: 60, GRU2: 10, Dense: 30	0.852643371
10	GRU1: 60, GRU2: 10, Dense: 30	0.948774993
11	GRU1: 30, GRU2: 40, Dense: 20	0.931843519
12	GRU1: 30, GRU2: 40, Dense: 20	0.927374244
13	GRU1: 30, GRU2: 40, Dense: 20	0.924971938
14	GRU1: 30, GRU2: 40, Dense: 20	0.952062368
15	GRU1: 30, GRU2: 40, Dense: 20	0.925306499
16	GRU1: 60, GRU2: 10, Dense: 20	0.868008852
17	GRU1: 60, GRU2: 10, Dense: 20	0.940914154
18	GRU1: 60, GRU2: 10, Dense: 20	0.924276114
19	GRU1: 60, GRU2: 10, Dense: 20	0.877828002
20	GRU1: 60, GRU2: 10, Dense: 20	0.855203509
21	GRU1: 20, GRU2: 20, Dense: 30	0.863945544
22	GRU1: 20, GRU2: 20, Dense: 30	0.83878237
23	GRU1: 20, GRU2: 20, Dense: 30	0.833712935
24	GRU1: 20, GRU2: 20, Dense: 30	0.93749994
25	GRU1: 20, GRU2: 20, Dense: 30	0.909497142

En la Ilustración 7 se muestra una gráfica de líneas con dos curvas suavizadas para resaltar la tendencia general minimizando el ruido representando así la precisión del modelo a lo largo de varias épocas de entrenamiento en dos conjuntos de datos diferentes: entrenamiento y validación (en el fondo se observan las líneas sin suavizar). Las curvas de precisión suavizadas, una en color naranja (F1 en entrenamiento) y la otra en azul (F1 en validación), muestran un comportamiento casi ideal de aprendizaje durante el entrenamiento del modelo GRU. La curva naranja, muestra un rápido aumento inicial y luego se estabiliza, alcanzando un valor suavizado (smoothed) de aproximadamente 97.75 %. La curva azul sigue un patrón similar, aumentando rápidamente y luego estabilizándose, con un valor suavizado ligeramente inferior de aproximadamente 97.14 %. Utilizando la técnica del early stopping, se tiene una convergencia ideal en 29 épocas y un tiempo de entrenamiento de 3,04 minutos.

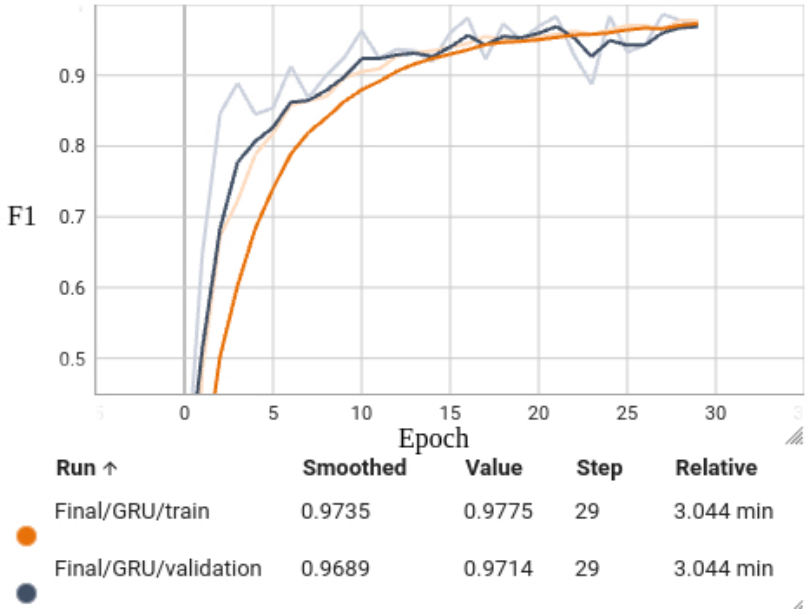

Fig. 7. Monitoreo de la métrica F1 (suavizada) durante el entrenamiento del modelo GRU

Teniendo en cuenta la Ilustración 7 se observa que ambas curvas convergen a medida que avanza el número de pasos (o épocas), lo que sugiere que el modelo está aprendiendo correctamente y generalizando bien a nuevos datos, evidenciado por la proximidad de las curvas de entrenamiento y validación. Este patrón es indicativo de un buen ajuste, ya que no

hay signos evidentes de sobreajuste o subajuste, que serían sugeridos por una gran divergencia entre las curvas de entrenamiento y validación.

En las Ilustraciones 8 y 9 se observa la matriz de confusión para la arquitectura planteada con mejores hiperparámetros en la Ilustración 6. En la Ilustración 8 se observa que la seña 1 ("Café") tiene un desempeño del 100 %, la seña 2 ("Con gusto") tiene un desempeño del 97 %, la seña 3 ("Qué necesitas") tiene un desempeño del 67 % que es el desempeño más bajo en comparación con las demás señas, la seña 4 ("Aromática") tiene un desempeño del 72 % y la seña 5 ("Hola") tiene un desempeño del 85 %.

Observando las métricas evaluadas con los datos de entrenamiento en la Ilustración 9 se tiene que la métrica correspondiente a la seña 3 es del 98 % lo que nos da a entender que puede haber un sobre ajuste de los datos de entrenamiento comparando con los resultados de test.

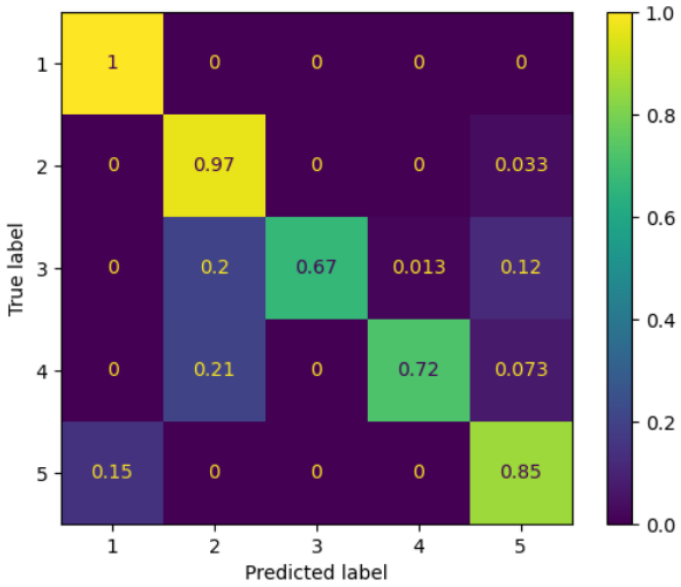

Fig. 8. Matriz de confusión GRU con datos de test

El histograma presentado en la Ilustración 10 se observa la distribución de los valores de los pesos del kernel de una capa densa (totalmente conectada) en una red neuronal a lo largo del entrenamiento. Las capas densas son fundamentales en muchas arquitecturas de redes neuronales, ya que proporcionan la transformación lineal de las entradas mediante la

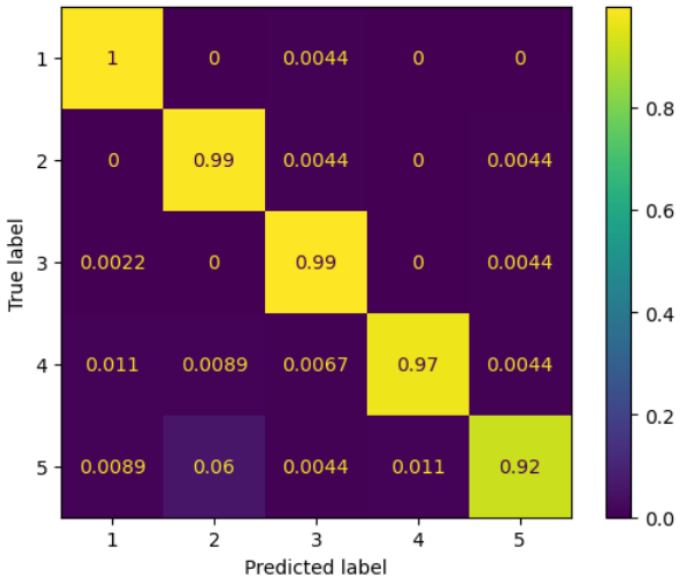

Fig. 9. Matriz de confusión GRU con datos de entrenamiento

multiplicación de estas entradas por los pesos del kernel seguida de una suma de los sesgos, antes de aplicar una función de activación no lineal.

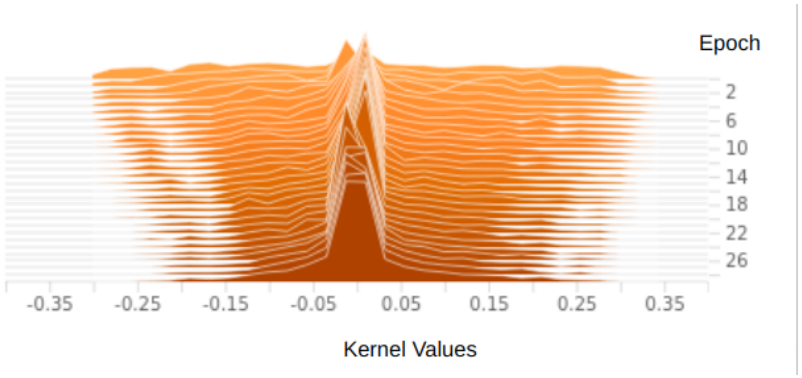

Fig. 10. Histograma por épocas de la distribución de los pesos de la primer capa densa en la arquitectura GRU

Los valores de los pesos en la Ilustración 10 se centran alrededor de cero, lo que es deseable ya que indica que hay una distribución equilibrada de pesos positivos y negativos. Esto permite que la capa capture tanto características positivas como negativas de los datos [36]. Los pesos están dentro de un rango aproximado entre -0.35 y 0.35, lo que sugiere que

no hay pesos extremadamente grandes que podrían causar una activación excesiva de las neuronas y posiblemente llevar al problema del desvanecimiento de los gradientes durante el entrenamiento [37]. En la Ilustración 10 se observa una distribución gaussiana en los pesos que tiene el modelo, donde se tienen picos más pronunciados cerca del cero. Esto indica que una gran cantidad de pesos son pequeños, lo que a veces es una señal de que la red podría no estar aprendiendo tantas características distintivas como podría, o que podría estar demasiado regularizada [36].

El histograma presentado en la Ilustración 11 muestra la distribución de los valores de sesgo (bias) de una capa densa en una red neuronal a lo largo del tiempo de entrenamiento. La mayoría de los valores de sesgo están agrupados cerca de cero, con una distribución asimétrica que tiene una cola más larga hacia la derecha. Los sesgos están típicamente centrados alrededor de cero para permitir ajustes positivos y negativos a la activación de las neuronas .

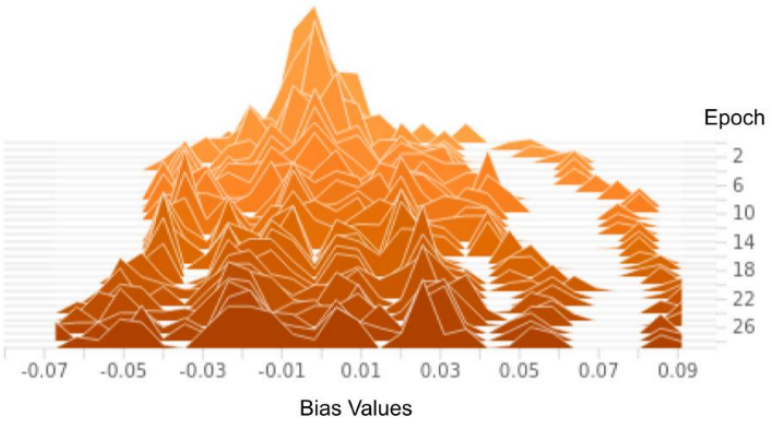

Fig. 11. Histograma por épocas de la distribución de los sesgos de la primer capa densa en la arquitectura GRU

Observando la Ilustración 11 a lo largo del eje horizontal, que representa diferentes etapas del entrenamiento, se observa que los valores de sesgo cambian con el tiempo. Esto indica que el proceso de aprendizaje está ajustando activamente los sesgos, lo que es un signo saludable de que la red está intentando optimizar estos parámetros.

El histograma de la Ilustración 12 muestra la distribución de los valores del kernel recurrente de una capa GRU (Gated Recurrent Unit) a lo largo del tiempo de entrenamiento. Las capas GRU son una variante de las redes neuronales recurrentes que tienen como objetivo resolver el problema de la dependencia a largo plazo, similar a las LSTM, pero con una estructura más simple que omite la célula de memoria [36].

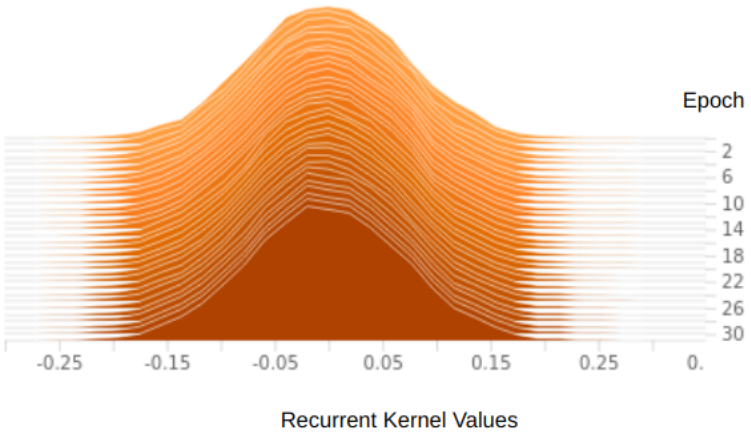

Fig. 12. Histograma por épocas de la distribución de los pesos del kernel recurrente de la primer capa GRU en la arquitectura GRU

En la Ilustración 12 la distribución de los valores de los pesos tiene una forma gaussiana, centrada alrededor de cero. Los valores se extienden aproximadamente desde -0.25 a 0.25. A lo largo del eje horizontal, que representa diferentes puntos en el tiempo durante el entrenamiento. Esto indica que el proceso de aprendizaje ha llegado a una fase de convergencia donde los pesos del kernel recurrente.[36]. La célula GRU utiliza este kernel recurrente para combinar la información del estado anterior y la entrada actual para generar el nuevo estado. La distribución simétrica de los pesos sugiere que la capa tiene la capacidad de ponderar y transformar la información entrante de manera equilibrada, lo cual es crucial para aprender dependencias temporales en los datos.

El histograma de la Ilustración 13 representa la distribución de los valores del kernel

de la primer capa de la neuronal GRU a lo largo del tiempo de entrenamiento. Los valores se extienden más allá del rango que se observa en los kernels recurrentes, desde aproximadamente -0.45 hasta 0.75. Este rango más amplio sugiere que la capa puede estar aprendiendo características con una mayor variabilidad y que el modelo podría estar en una fase de aprendizaje donde se exploran ajustes más significativos en los pesos. La forma de la distribución se mantiene consistente a lo largo del tiempo (indicado por la superposición de las líneas en el histograma), lo que sugiere que los pesos han alcanzado un estado de estabilidad en términos de su magnitud media, aunque la propagación de valores sigue siendo amplia [36].

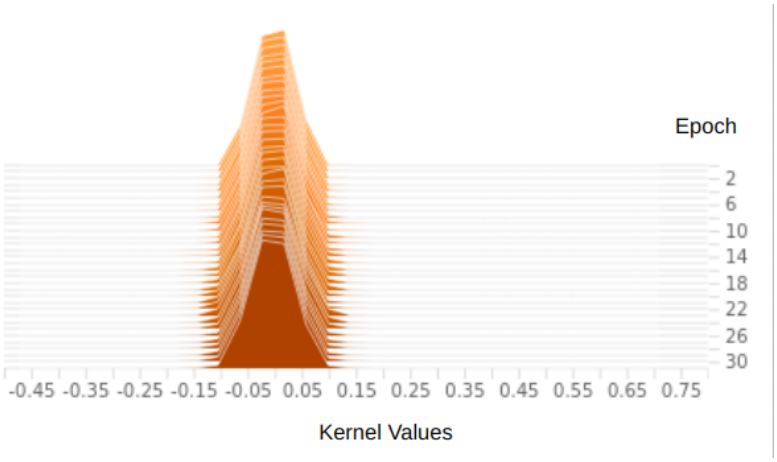

Fig. 13. Histograma por épocas de la distribución de los pesos del kernel de la primer capa GRU en la arquitectura GRU

Observando la Ilustración 13 muchos valores están cerca de cero teniendo una distribución simétrica de los pesos que van desde -0.15 a 0.15, lo sugiere que la red está en capacidad de ajustar su aprendizaje en respuesta valores positivos y negativos.

B. Arquitectura basada en LSTM

En la tabla III se presentan los resultados obtenidos de la optimización de hiperparámetros en modelos que utilizan unidades Long Short-Term Memory networks para la clasificación de de señas de Lengua de Señas Colombiana (LSC).

Teniendo en cuenta la Tabla III se define la arquitectura con mejores hiperparámetros

en la Ilustración 14 donde se tienen 50 neuronas recurrentes en la primer capa, 40 neuronas recurrentes en la segunda capa, y 50 neuronas densas en la tercera capa. Teniendo en cuenta las dimensiones de la capa de entrada y la capa de salida, se tiene un total de 359.516 parámetros entrenables con un peso total de 1.37 MB.

En la Ilustración 15 se incluyen dos líneas suavizadas (en el fondo se observan las líneas sin suavizar) que representan la evolución la métrica F1 para dos conjuntos de datos diferentes: entrenamiento (en azul) y validación (en rosa). Ambas líneas muestran una tendencia al alza, lo que sugiere una mejora en la métrica a medida que avanzan las iteraciones del entrenamiento. Las líneas están bastante cercanas entre sí, lo que indica que el modelo LSTM está generalizando bien y no hay una gran brecha de desempeño entre el entrenamiento y la validación. Utilizando la técnica del early stopping, se tiene una convergencia ideal en 35 épocas y un tiempo de entrenamiento de 2,04 minutos.

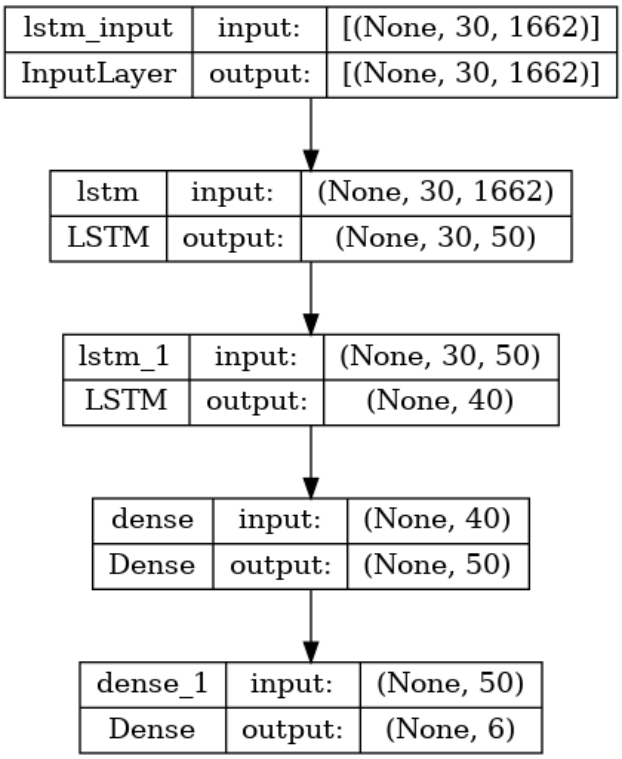

Fig. 14. Arquitectura LSTM con mejores hiperparámetros

TABLA III
RESULTADOS DE EXPERIMENTOS DE MODELOS LSTM CON DIFERENTES HIPERPARÁMETROS

Intento	Hiperparámetros	F1
1	LSTM1: 40, LSTM2: 60, Dense: 50	0.966592371
2	LSTM1: 40, LSTM2: 60, Dense: 50	0.955555499
3	LSTM1: 40, LSTM2: 60, Dense: 50	0.972191215
4	LSTM1: 40, LSTM2: 60, Dense: 50	0.956618428
5	LSTM1: 40, LSTM2: 60, Dense: 50	0.95991087
6	LSTM1: 50, LSTM2: 40, Dense: 10	0.875699878
7	LSTM1: 50, LSTM2: 40, Dense: 10	0.947719574
8	LSTM1: 50, LSTM2: 40, Dense: 10	0.955555499
9	LSTM1: 50, LSTM2: 40, Dense: 10	0.973333299
10	LSTM1: 50, LSTM2: 40, Dense: 10	0.924276114
11	LSTM1: 30, LSTM2: 10, Dense: 10	0.699208379
12	LSTM1: 30, LSTM2: 10, Dense: 10	0.802325547
13	LSTM1: 30, LSTM2: 10, Dense: 10	0.810871899
14	LSTM1: 30, LSTM2: 10, Dense: 10	0.78071177
15	LSTM1: 30, LSTM2: 10, Dense: 10	0.754176557
16	LSTM1: 60, LSTM2: 10, Dense: 30	0.886644125
17	LSTM1: 60, LSTM2: 10, Dense: 30	0.791237116
18	LSTM1: 60, LSTM2: 10, Dense: 30	0.796274662
19	LSTM1: 60, LSTM2: 10, Dense: 30	0.779376447
20	LSTM1: 60, LSTM2: 10, Dense: 30	0.87766552
21	LSTM1: 40, LSTM2: 10, Dense: 40	0.876957476
22	LSTM1: 40, LSTM2: 10, Dense: 40	0.722151041
23	LSTM1: 40, LSTM2: 10, Dense: 40	0.53474319
24	LSTM1: 40, LSTM2: 10, Dense: 40	0.680465698
25	LSTM1: 40, LSTM2: 10, Dense: 40	0.876957476

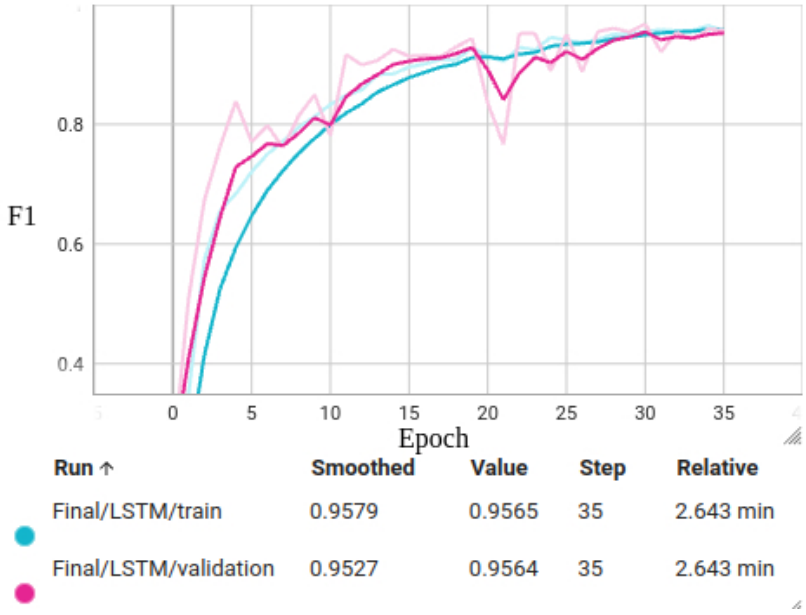

Fig. 15. Monitoreo de la métrica F1 (suavizada) durante el entrenamiento del modelo LSTM

En las Ilustraciones 16 y 17 se observa la matriz de confusión para la arquitectura planteada con mejores hiperparámetros en la Ilustración 14. En la Ilustración 16 se observa que las señas 1 y 5 (“Café” y “Hola”) tienen un desempeño del 100 %, mientras que las señas 2 y 4 (“Con gusto” y “Aromática”) presentan un desempeño del 94 % y 86 % respectivamente. La seña 3 (“Qué necesitas”) es la que peor desempeño demuestra con el 67 % de las muestras de test bien clasificadas.

Observando las métricas evaluadas con los datos de entrenamiento en la Ilustración 9 se tiene que la métrica correspondiente a la seña 3 (“Qué necesitas”) es del 98 % lo que nos da a entender que puede haber un sobre ajuste de los datos de entrenamiento comparando con los resultados de test.

La Ilustración 18 muestra un histograma que representa la distribución de los pesos (también conocidos como kernel) de una capa densa de una red neuronal a lo largo del tiempo de entrenamiento. Cada línea horizontal del histograma corresponde a un paso específico en el tiempo de entrenamiento, con los pasos más recientes en la parte superior y los más antiguos en la parte inferior. Los valores de los pesos se distribuyen a lo largo del eje horizontal.

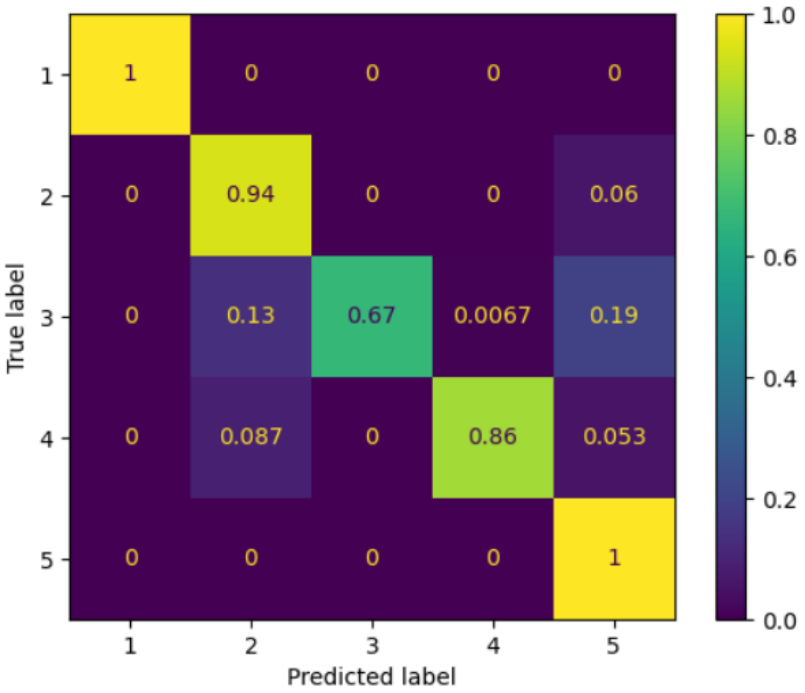

Fig. 16. Matriz de confusión LSTM con datos de test

El histograma de la Ilustración 18 tiene una forma puntiaguda en el centro, lo que sugiere que la mayoría de los pesos están concentrados alrededor de cero y que la distribución de estos es bastante simétrica en ambos lados del eje vertical. Esta simetría y concentración central pueden indicar un buen comportamiento de inicialización y actualización de los pesos durante el entrenamiento. Los picos muy estrechos en el centro pueden ser resultado de una inicialización de pesos cercana a cero o de una fuerte regularización durante el entrenamiento, lo que a menudo se emplea para prevenir el sobreajuste y promover la generalización [36].

En la Ilustración 19 se observa que los valores de sesgo están distribuidos en un rango más estrecho que los pesos (kernels) del histograma anterior, centrados aproximadamente entre -0.15 y 0.09. La distribución general de los sesgos tiene una forma de campana, lo que es indicativo de una distribución normal o gaussiana. Esto sugiere que los sesgos están siendo ajustados de manera que se mantienen alrededor de un valor central, lo que es común en redes neuronales que no están sesgadas hacia activaciones positivas o negativas excesivas [36].

Se observa en la Ilustración 19 que a lo largo del entrenamiento, los sesgos han mostra-

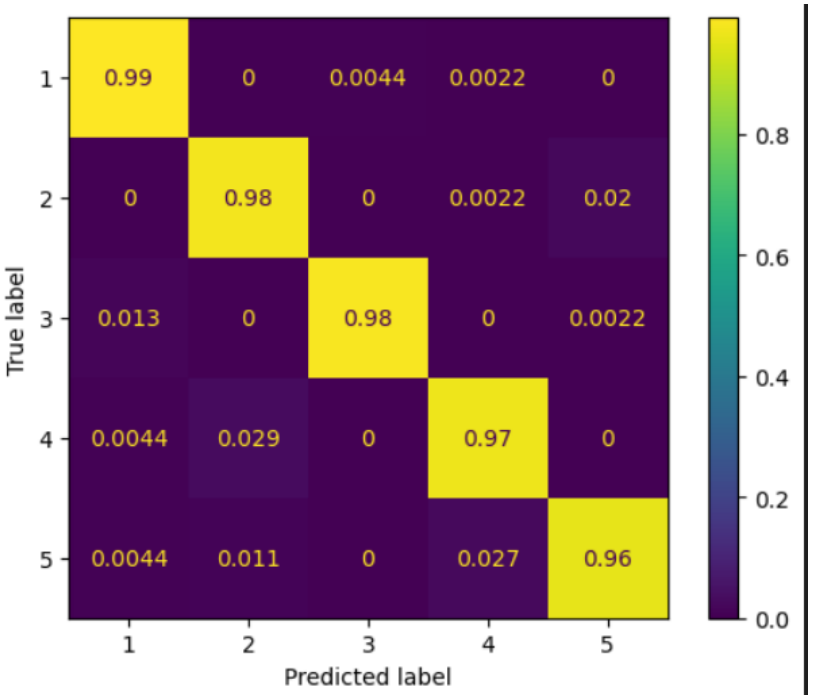

Fig. 17. Matriz de confusión LSTM con datos de entrenamiento

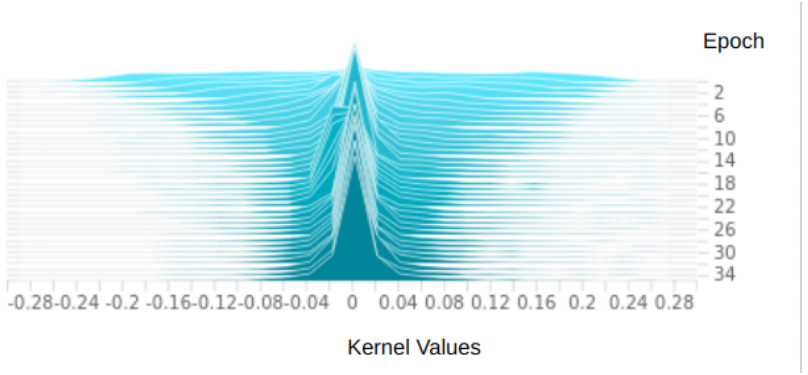

Fig. 18. Histograma por épocas de la distribución de los pesos de la primer capa densa en la arquitectura LSTM

do una variación en sus valores, pero permanecen dentro de un rango relativamente estrecho. Esto puede indicar que la red ha aprendido a ajustar las activaciones de las neuronas de manera efectiva sin depender excesivamente de los sesgos [36].

El histograma de la Ilustración 20 muestra la distribución de los valores de sesgos de

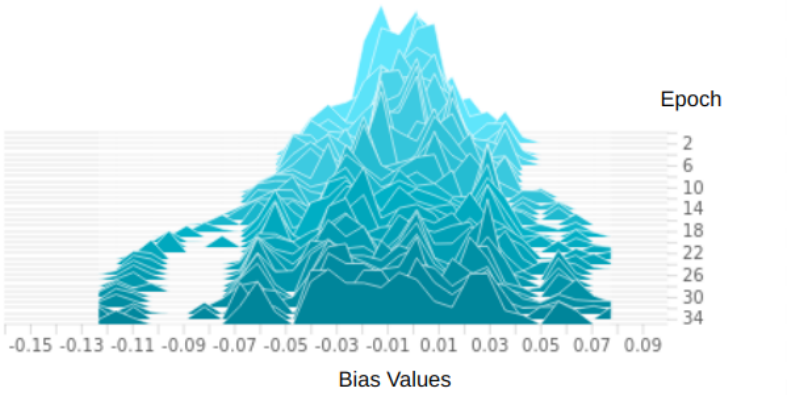

Fig. 19. Histograma por épocas de la distribución de los sesgos de la primer capa densa en la arquitectura LSTM

una celda LSTM durante el entrenamiento. Los valores de los sesgos son fundamentales en una red neuronal, ya que permiten que el modelo ajuste sus predicciones independientemente del peso de las entradas. Analizar la distribución de estos valores puede ofrecer una gran cantidad de información sobre el comportamiento y la configuración del modelo [36].

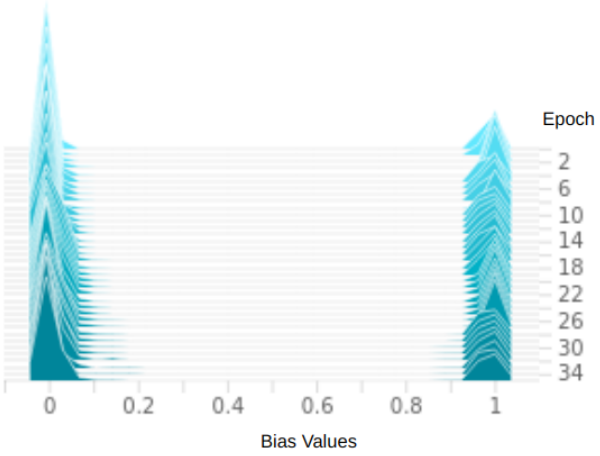

Fig. 20. Histograma por épocas de la distribución de los Sesgos de la primer capa LSTM en la arquitectura LSTM

En la ilustración 20 se observa que los valores de sesgo se encuentran principalmente agrupados cerca del 0 y del 1. Esto es inusual para los parámetros de sesgo, ya que típicamente

se distribuyen alrededor de un valor central como 0 para permitir ajustes simétricos tanto en la dirección positiva como negativa. Con los sesgos inclinados hacia valores mayores que 0, las puertas de la LSTM (entrada, olvido, y salida) tienen más probabilidades de estar abiertas en las primeras etapas del entrenamiento, lo que es una estrategia para permitir que la red pase más información a través del tiempo y capture dependencias a largo plazo [36].

El histograma de la Ilustración 21 muestra la distribución de los valores de los pesos del kernel (también conocidos como matriz de pesos) de una celda LSTM en diferentes puntos durante el entrenamiento. El kernel conecta la entrada con el estado oculto y es fundamental para determinar cómo la información es procesada y transmitida a través de la red.

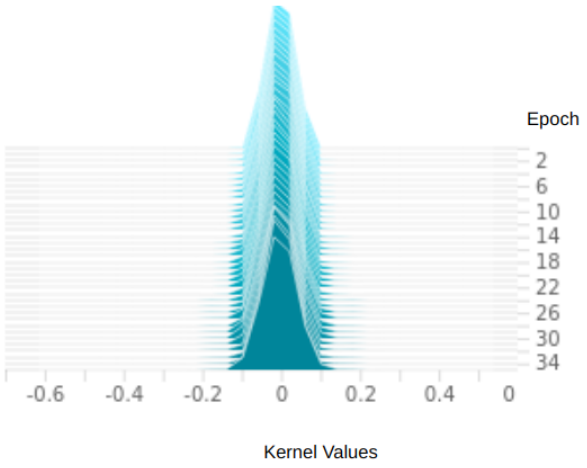

Fig. 21. Histograma por épocas de la distribución de los pesos del kernel de la primer capa LSTM en la arquitectura LSTM

En la figura 21 se observa que los valores del kernel están centrados alrededor de cero, con una distribución simétrica. Esto es un indicativo de que los pesos han sido inicializados correctamente [36]. La mayoría de los valores están dentro de un rango pequeño, entre -0.5 y 0.5. Esto ayuda a prevenir el problema de desvanecimiento del gradiente, un problema común en las RNN debido a su naturaleza recurrente y a la propagación del gradiente a través de muchas etapas temporales. También se observa un pico claro alrededor de cero, lo que indica que muchos pesos son pequeños o nulos.

En la Ilustración 22 se observa la distribución de los pesos del kernel recurrente en

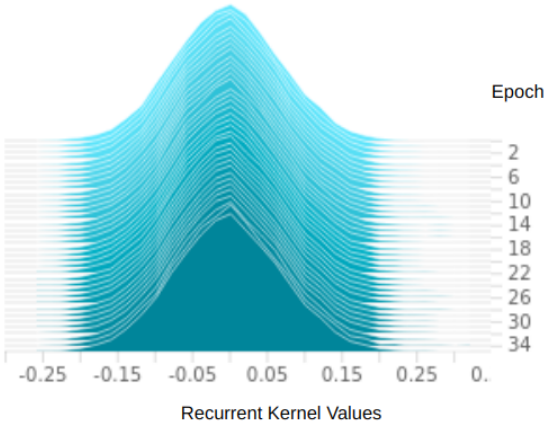

Fig. 22. Histograma por épocas de la distribución de los pesos del kernel recurrente de la primer capa LSTM en la arquitectura LSTM

una red neuronal LSTM a lo largo del entrenamiento. El kernel recurrente es una matriz de pesos que conecta el estado oculto de un paso de tiempo con el siguiente. Es crucial para la capacidad de la LSTM de retener información a lo largo del tiempo, lo que le permite capturar dependencias temporales en secuencias de datos. La distribución es simétrica alrededor de cero y presenta una forma de campana, lo que sugiere una inicialización adecuada y una adaptación uniforme de los pesos durante el entrenamiento. La simetría alrededor de cero es importante para permitir que la red ajuste los estados ocultos en cualquier dirección, positiva o negativa, y para evitar sesgos en el procesamiento de la información. Los valores están mayoritariamente concentrados entre -0.25 y 0.25, lo que indica que no hay valores extremos que son una de las causales del problema de explosión o desvanecimiento del gradiente [37]. Una distribución estrecha también puede indicar que la red está aprendiendo una representación más precisa y controlada de los datos. A lo largo del entrenamiento, los pesos evolucionan pero mantienen una distribución consistente. Esto puede indicar que el modelo está aprendiendo de manera estable, con cambios graduales en los pesos a medida que la red se ajusta a los datos de entrenamiento.

En la Tabla IV se observa un resumen de los mejores hiperparámetros para cada una de las arquitecturas. Teniendo en cuenta las Tablas II y III la arquitectura con mejor desempeño

es la LSTM, sin embargo, analizando la Tabla V donde están los promedios de los resultados de los 25 experimentos por cada una de las arquitecturas, se observa una mayor variabilidad en la arquitectura LSTM, mientras que la arquitectura GRU mantiene una variabilidad baja.

TABLA IV
TABLA DE COMPARACIÓN DE MEJORES RESULTADOS CON GRU Y LSTM

Hiperparámetros	**Arquitectura**	**F1**
LSTM1: 50, LSTM2: 40, Dense: 10	LSTM	97.33 %
GRU1: 70, GRU2: 20, Dense: 20	GRU	95.65 %

TABLA V
TABLA DE COMPARACIÓN DE PROMEDIO DE RESULTADOS CON GRU Y LSTM

Arquitectura	**Promedio F1**	**Desviación estándar (F1)**
LSTM	84.62 %	11.04 %
GRU	91.17 %	3.76 %

IX. DISCUSIÓN

La evaluación exhaustiva de los modelos LSTM y GRU mediante búsqueda de hiperparámetros nos proporciona una visión clara de su rendimiento. El modelo LSTM registró una precisión promedio del 84.62 % con una desviación estándar del 11.04 % (Ilustración 17), mientras que el modelo GRU demostró ser más consistente, alcanzando una precisión promedio del 91.17 % y una desviación estándar significativamente reducida del 3.76 % (Ilustración 9). Este contraste pone de manifiesto la mayor estabilidad del modelo GRU frente a las variaciones en los hiperparámetros dentro de la metodología de este estudio.

En cuanto a la eficiencia temporal, el entrenamiento del modelo GRU convergió en 29 épocas, tomando 3,04 minutos (Ilustración 7), en comparación con la convergencia del

modelo LSTM en 35 épocas, pero con un tiempo menor de 2,64 minutos (Ilustración 15). Este hallazgo sugiere una posible mayor eficiencia computacional de la arquitectura LSTM, a pesar de requerir más épocas para alcanzar la convergencia.

Al analizar las matrices de confusión (Ilustraciones 8, 9, 16, 17), encontramos un rendimiento comparable en ambas arquitecturas, con una precisión del 67 % en la categorización de la seña "Qué necesitas". La consistencia en este resultado puede indicar un sesgo inherente en la señal que afecta de manera similar a ambos modelos.

La inspección de los sesgos y los pesos en las capas recurrentes reveló patrones de distribución similares entre los dos modelos. Sin embargo, se observó que el modelo GRU se adaptaba más rápidamente a los valores óptimos necesarios para una generalización efectiva de los datos.

Respecto a las capas densas, los pesos de la arquitectura GRU exhibieron un rango más amplio entre -0.15 y 0.15 (Ilustración 10), proporcionando un espectro más extenso para la activación de las neuronas. Por otro lado, los pesos de la capa densa del modelo LSTM mostraron un rango más estrecho, aproximadamente entre -0.09 y 0.09 (Ilustración 18). Estos resultados pueden reflejar diferencias significativas en la capacidad de cada modelo para procesar y aprender de las características de los datos.

X. CONCLUSIONES

Este informe resalta los avances en interpretación automática de Lengua de Señas Colombiana mediante redes neuronales recurrentes, LSTM y GRU, apoyado en la creación de un corpus significativo. La creación del corpus fue fundamental para el entrenamiento de modelos, junto con la adecuada selección de metodologías donde se evidencia la precisión y eficacia promedio de GRU sobre LSTM. La colaboración con la comunidad sorda ha sido vital, asegurando que las herramientas desarrolladas satisfagan necesidades reales, promoviendo así la inclusión. Este esfuerzo interdisciplinario subraya el compromiso con la diversidad, accesibilidad y la tecnología como medio para la equidad social.

Se confirmado la utilidad de aplicar modelos de redes neuronales recurrentes, específicamente LSTM y GRU, para la traducción automática de la Lengua de Señas Colombiana. Se observó que el modelo GRU exhibe una precisión promedio superior y una menor desviación estándar que su contraparte LSTM, lo cual indica una estabilidad y consistencia destacables bajo variadas configuraciones de hiperparámetros. La notable rapidez de convergencia del modelo GRU apunta a una capacidad de aprendizaje eficiente, un atributo valioso para su implementación en entornos prácticos.

El avance técnico logrado en la traducción automática de lengua de señas a texto es evidente. La habilidad para procesar y aprender de las características específicas de los datos se refleja en la adaptabilidad y precisión de los modelos desarrollados. La implementación de estas tecnologías podría, por tanto, representar un paso significativo en el campo de la asistencia comunicativa automatizada.

Es imprescindible señalar la importancia de la precisión y eficiencia en el reconocimiento de patrones dentro de la Lengua de Señas Colombiana, como se evidencia en los resultados técnicos de este proyecto. La colaboración con la comunidad sorda en el desarrollo y la evaluación de estas herramientas asegura su alineación con las necesidades específicas y mejora la funcionalidad del sistema. Este enfoque garantiza que la tecnología no solo sea avanzada sino también aplicable y relevante para los usuarios finales.

Es importante destacar que, a pesar de los avances tecnológicos, la colaboración conti-

nua con la comunidad sorda es esencial para garantizar que estas herramientas satisfagan sus necesidades y preferencias. La tecnología de traducción de la lengua de señas no es solo un logro técnico, sino también un paso hacia la equidad y la accesibilidad, reforzando el compromiso de la sociedad con la diversidad y la inclusión. Este estudio subraya la importancia de la investigación interdisciplinaria y la colaboración comunitaria para el desarrollo de soluciones tecnológicas inclusivas.

Algunas limitaciones que se tuvieron fue la poca cantidad de datos (1500 videos) para 5 señas, teniendo 300 videos por cada una de las señas. Esto es un limitante debido a que al momento de aplicar división de datos se reduce mucho la cantidad de datos con la que se pueden realizar pruebas y validaciones. Otra limitante relacionada a los datos es la cantidad de personas que participaron en la creación de los videos (4) lo cual al momento de hacer la división de datos, para evitar sesgos, se debió separar uno de los participantes al azar para realizar pruebas y validaciones, lo cual al tener solo 3 participantes para entrenamiento, puede crear sesgos y poca generalización de los modelos.

XI. RECOMENDACIONES

En futuras investigaciones sobre la traducción automática de la Lengua de Señas Colombiana mediante el uso de inteligencia artificial, sería provechoso expandir el conjunto de datos más allá de las 1500 muestras actuales. Al incluir una diversidad mayor de señas y contextos, así como las variaciones individuales en la señalización, se podría enriquecer la base de datos, mejorando la generalización y la utilidad del modelo. Un conjunto de datos más extenso y variado permitiría también una evaluación más rigurosa de la capacidad de generalización del modelo.

La experimentación con arquitecturas neuronales más avanzadas, como los Transformers y los modelos basados en mecanismos de atención, podría ofrecer mejoras significativas en la interpretación de las secuencias complejas que caracterizan a la lengua de señas. La inclusión de elementos no manuales, como las expresiones faciales y movimientos corporales, también podría proporcionar una dimensión adicional al proceso de traducción, al incorporar componentes críticos para el significado completo en la lengua de señas.

La colaboración interdisciplinaria entre lingüistas, expertos en lengua de señas, ingenieros en IA y miembros de la comunidad sorda es esencial para asegurar la precisión y relevancia de los modelos desarrollados. Las pruebas de usabilidad en situaciones reales del día a día permitirán evaluar la eficacia de la tecnología y su capacidad para funcionar como una herramienta de apoyo en la educación, el trabajo y la interacción social. Además, es importante continuar el diálogo sobre las implicaciones éticas del desarrollo tecnológico, garantizando que el empoderamiento y el respeto hacia la comunidad sorda se mantengan en el centro de la investigación.

El desarrollo de herramientas de anotación mejoradas es también crucial para permitir una expansión eficiente y precisa del conjunto de datos. Investigar la transferibilidad de los modelos entre diferentes lenguas de señas podría abrir la puerta a sistemas de traducción más versátiles y de amplio alcance. Por último, optimizar los modelos para que sean más compactos y rápidos facilitará su implementación en dispositivos móviles, ampliando su accesibilidad y utilidad.

Estos esfuerzos colectivos no sólo avanzarán en el campo técnico, sino que también tendrán un impacto social significativo, promoviendo la inclusión y el acceso a la información para la comunidad sorda, y reafirmando el papel vital de la tecnología en la construcción de una sociedad más equitativa.

REFERENCIAS

[1] A. C. Carneiro, L. B. Silva, and D. P. Salvadeo, "Efficient sign language recognition system and dataset creation method based on deep learning and image processing," in *Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*, vol. 11878. SPIE, 2021, pp. 11–19.

[2] D. Bragg, O. Koller, N. Caselli, and W. Thies, "Exploring collection of sign language datasets: Privacy, participation, and model performance," in *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, 2020, pp. 1–14.

[3] J. Zheng, Z. Zhao, M. Chen, J. Chen, C. Wu, Y. Chen, X. Shi, Y. Tong *et al.*, "An improved sign language translation model with explainable adaptations for processing long sign sentences," *Computational Intelligence and Neuroscience*, vol. 2020, 2020.

[4] J. Cadavid Ramirez, "Análisis exploratorio de datos de los estudiantes con discapacidad en la Universidad de Antioquia," Repositorio Institucional Universidad de Antioquia, 2023, retrieved November 21, 2023. [Online]. Available: https://bibliotecadigital.udea.edu.co/handle/10495/35452

[5] J. Ospina Sánchez, "Accesibilidad física y tecnológica, una apuesta colectiva que avanza en la UdeA," UdeA Noticias, 2023, accessed: Octuber 11, 2023. [Online]. Available: https://www.udea.edu.co/wps/portal/udea/web/inicio/udea-noticias

[6] U. Farooq, M. S. M. Rahim, N. Sabir, A. Hussain, and A. Abid, "Advances in machine translation for sign language: approaches, limitations, and challenges," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14 357–14 399, 2021.

[7] Departamento Administrativo Nacional de Estadística (DANE), "PERSONAS CON DISCAPACIDAD, retos diferenciales en el marco del COVID-19," https://www.dane.gov.co/files/investigaciones/discapacidad/

2020-Boletin-personas-con-discapacidad-marco-COVID-19.pdf, 2020, Último acceso: 10 de Enero del 2024.

[8] Ministerio de Salud y Protección Social, "Boletines Poblacionales: Personas con Discapacidad-Oficina de Promoción Social II-2020," https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/PS/boletines-poblacionales-personas-discapacidadI-2020.pdf, 2020, Último acceso: 10 de Enero del 2024.

[9] M. García-Cano and et al., "¿Qué es una Universidad Inclusiva?" https://blogs.comillas.edu/fei/wp-content/uploads/sites/7/2018/02/ApuntesInclusion-QUE-ES-UNIVERSIDAD-INCLUSIVA.pdf, 2017, Último acceso: 10 de Enero del 2024.

[10] Universidad Carlos III de Madrid, "La UC3M obtiene el sello BEQUAL por su compromiso con la inclusión de las personas con discapacidad," https://www.uc3m.es/ss/Satellite/UC3MInstitucional/es/Detalle/Comunicacion_C/1371262281111/1371215537949/La_UC3M_obtiene_el_sello_BEQUAL_por_su_compromiso_con_la_inclusion_de_las_personas_con_discapacidad, 2022, Último acceso: 10 de Enero del 2024.

[11] Y. Liu, C. M. Vong, and P. K. Wong, "Extreme learning machine for huge hypotheses re-ranking in statistical machine translation," *Cognitive Computation*, vol. 9, no. 2, pp. 285–294, 2017.

[12] S. Sreelekha, "Statistical vs rule based machine translation; a case study on Indian language perspective," in *International Conference on Intelligent Computing and Applications*, 2017.

[13] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, and M. Palmer, "A machine translation system from English to American Sign Language," in *Envisioning Machine Translation in the Information Future: 4th Conference of the Association for Machine*

Translation in the Americas, AMTA 2000 Cuernavaca, Mexico, October 10–14, 2000 Proceedings 4. Springer, 2000, pp. 54–67.

[14] L. S. dÁrmond and M. Speers, "Representation of American sign language for machine translation," Ph.D. dissertation, Georgetown University, 2002.

[15] P. García, "Las personas con discapacidad en los ODS," https://www.gndiario.com/ods-personas-discapacidad-cermi, 2020, Último acceso: 10 de Enero del 2024.

[16] Organización de las Naciones Unidas, "NACIONES UNIDAS- Personas con Discapacidad," https://www.un.org/development/desa/disabilities-es/convencion-sobre-los-derechos-de-las-personas-con-discapacidad-2.html, 2006, Último acceso: 10 de Enero del 2024.

[17] N. E. Gómez Rúa and G. Montenegro Martínez, "Discapacidad, empleo y pobreza," *Revista CES Derecho*, vol. 8, no. 2, pp. 205–207, 2017.

[18] Vicepresidencia de la República, "Alianza por la inclusión de las personas con discapacidad," https://www.pactoglobal-colombia.org/news/importante-alianza-por-la-inclusion-de-las-personas-con-discapacidad.html, 2021, Último acceso: 10 de Enero del 2024.

[19] C. Vogler and S. Goldenstein, "Toward computational understanding of sign language," *Technology and Disability*, vol. 20, no. 2, pp. 109–119, 2008.

[20] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef *et al.*, "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 2019, pp. 16–31.

[21] W. C. Stokoe, "Sign language structure," *Annual review of anthropology*, vol. 9, no. 1, pp. 365–390, 1980.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[24] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] M. Mozer, "A Focused Backpropagation Algorithm," *Complex systems*, vol. 3, pp. 349–381, 1989.

[26] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[27] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen [Ph. D. dissertation]," *Technische Universitt München, München, Germany*, 1991.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[30] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2009, vol. 2.

[31] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[32] M. Lathkar, "Introduction to FastAPI," in *High-Performance Web Apps with FastAPI: The Asynchronous Web Framework Based on Modern Python.* Springer, 2023, pp. 1–28.

[33] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in *Proceedings of the tenth european conference on computer systems*, 2015, pp. 1–17.

[34] R. Singer, "Shape Up-Stop Running in Circles and Ship Work that Matters. Basecamp (2019)."

[35] J. Mockus and J. Mockus, *The Bayesian approach to local optimization*. Springer, 1989.

[36] G. Zaccone and M. R. Karim, *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python*. Packt Publishing Ltd, 2018.

[37] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

ANEXOS

Anexo A. Captura de los videos y procesamiento con Mediapipe

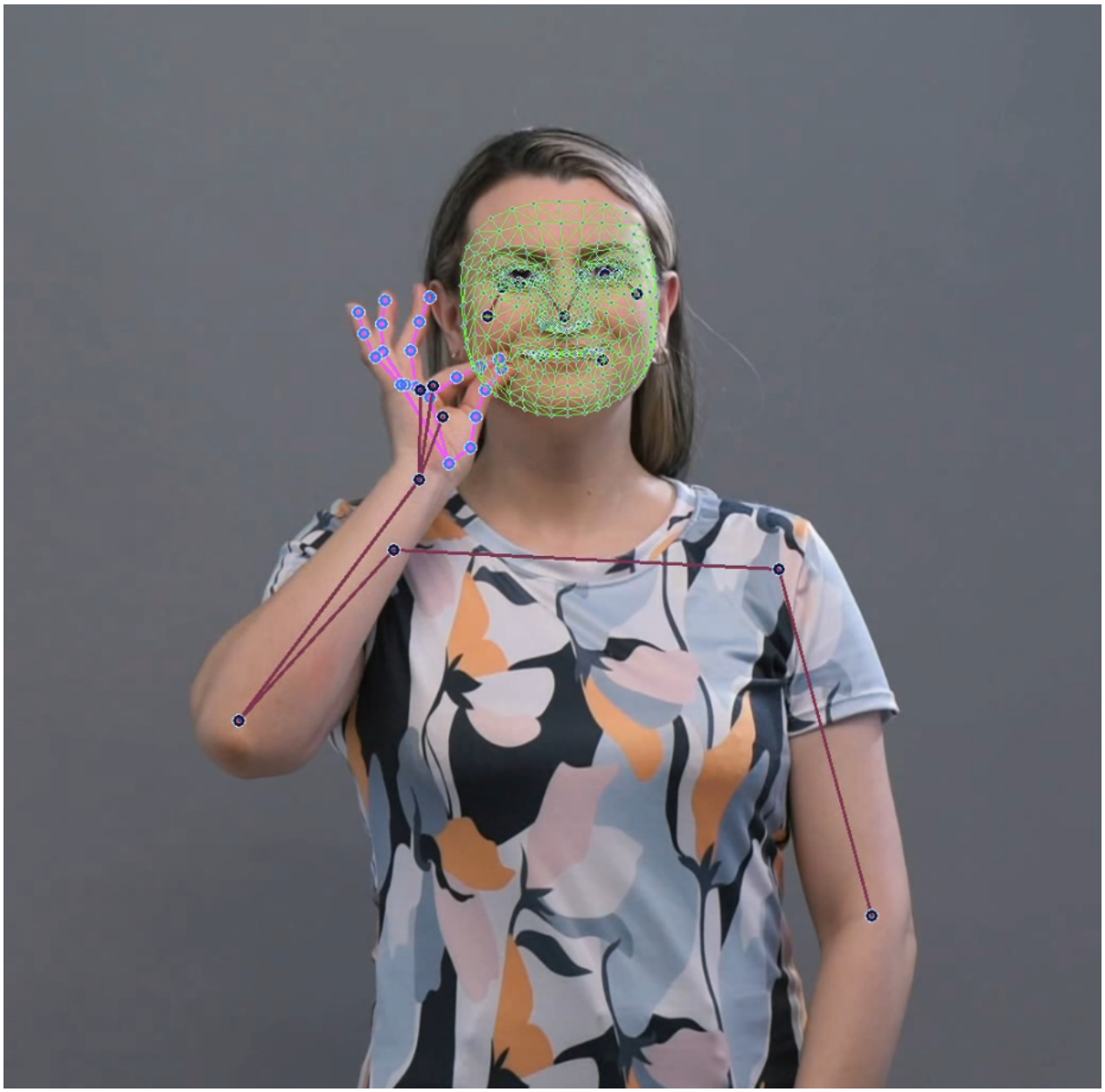

Fig. 23. Captura del video de la seña “Café” por la participante 1, con cámara número 1

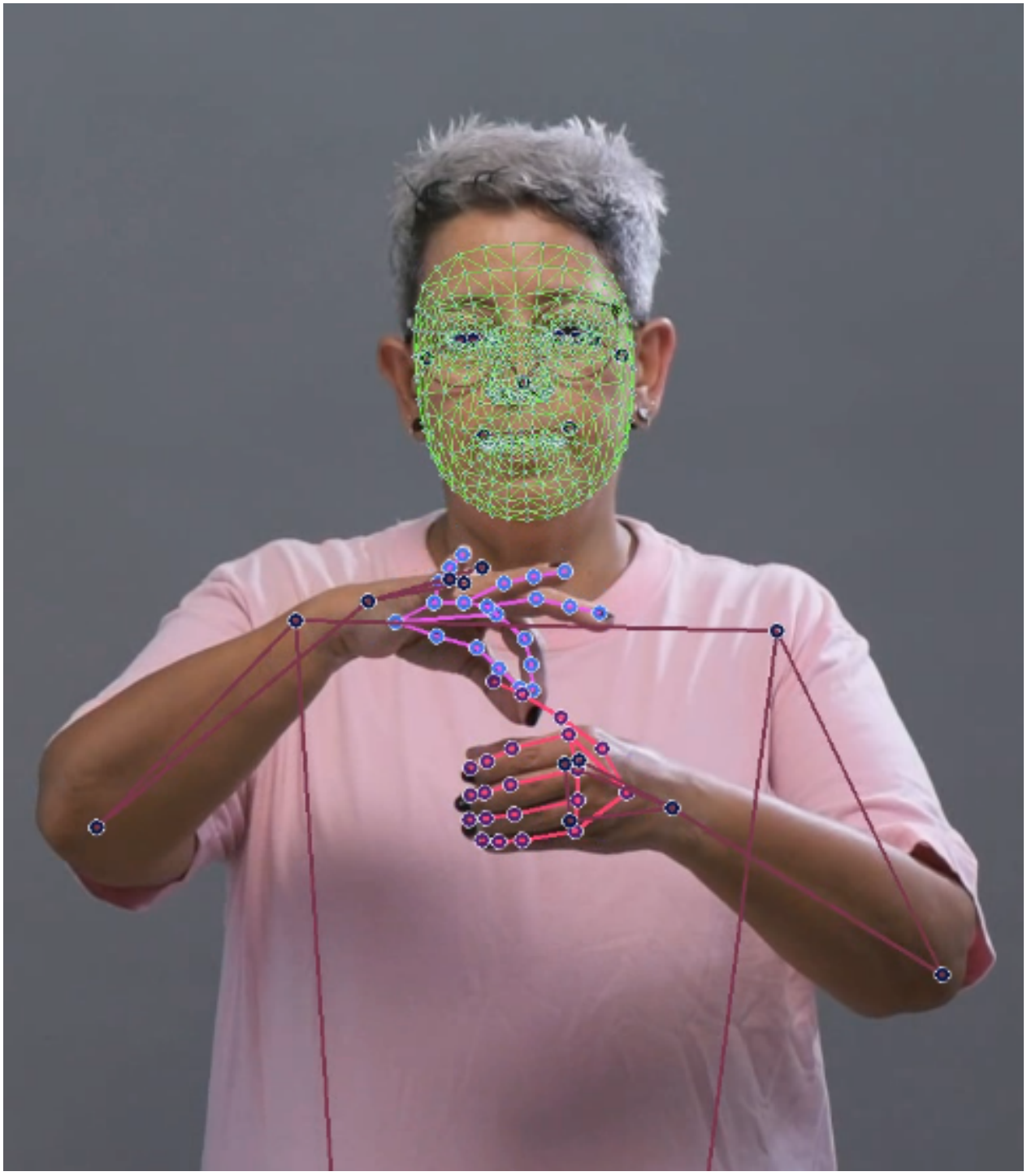

Fig. 24. Captura del video de la seña “Aromática” por la participante 2, con cámara número 1

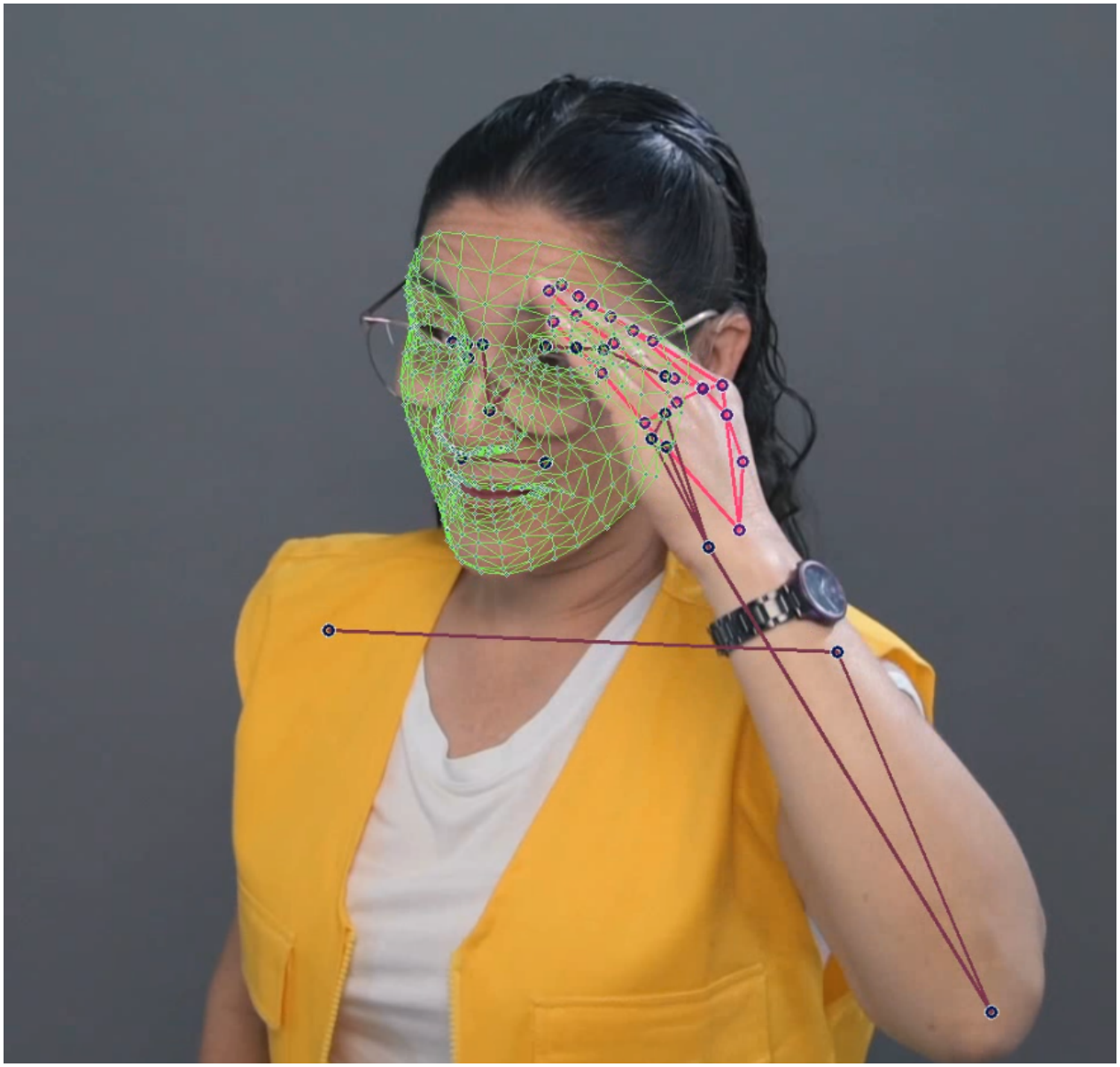

Fig. 25. Captura del video de la seña “Hola” por la participante 3, con cámara número 2

Anexo B. Capturas de la Plataforma SignAI para la gestión de datos y modelos de Lengua de Señas Colombiana

Fig. 26. Captura del video de la seña “Qué necesita” por el participante 4, con cámara número 3

Fig. 27. Captura del video de la seña “Con gusto” por el participante 4, con cámara número 3

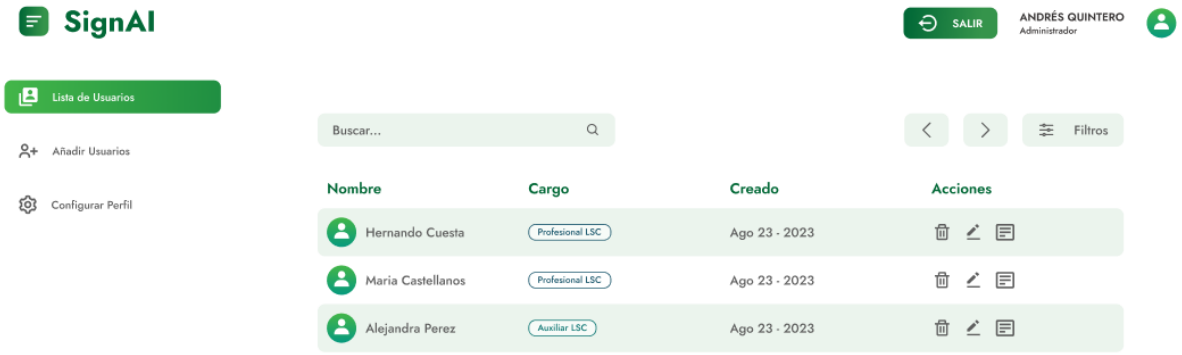

Fig. 28. Módulo de Administrador para la gestión de usuarios

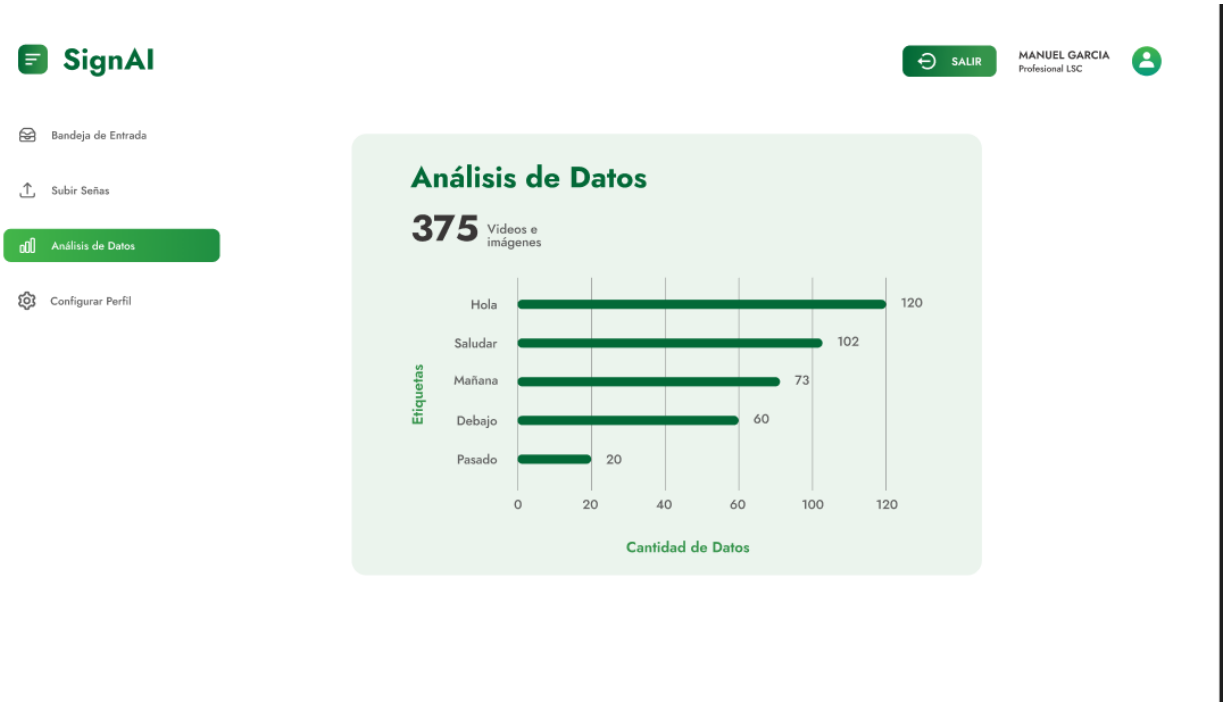

Fig. 29. Módulo de Análisis de datos

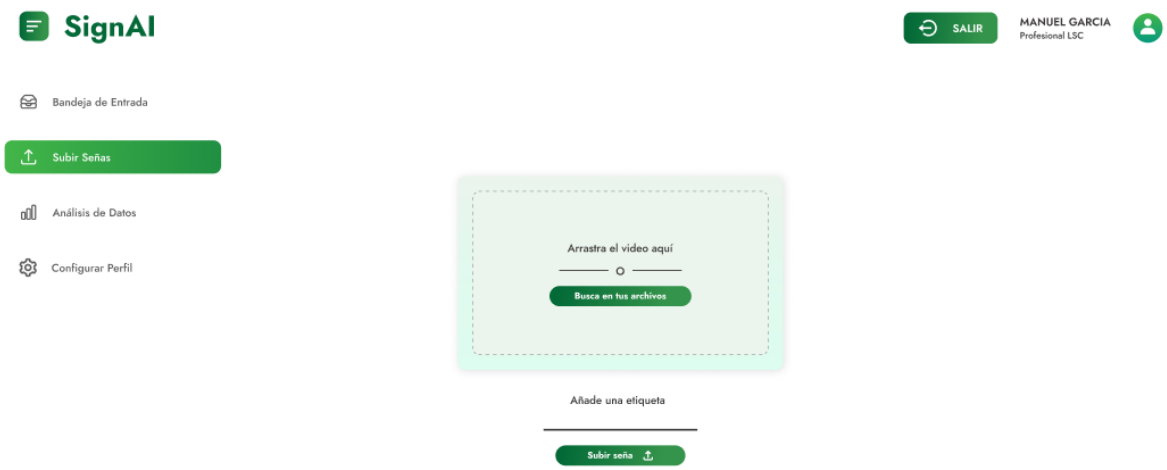

Fig. 30. Módulo para subir señas a la plataforma

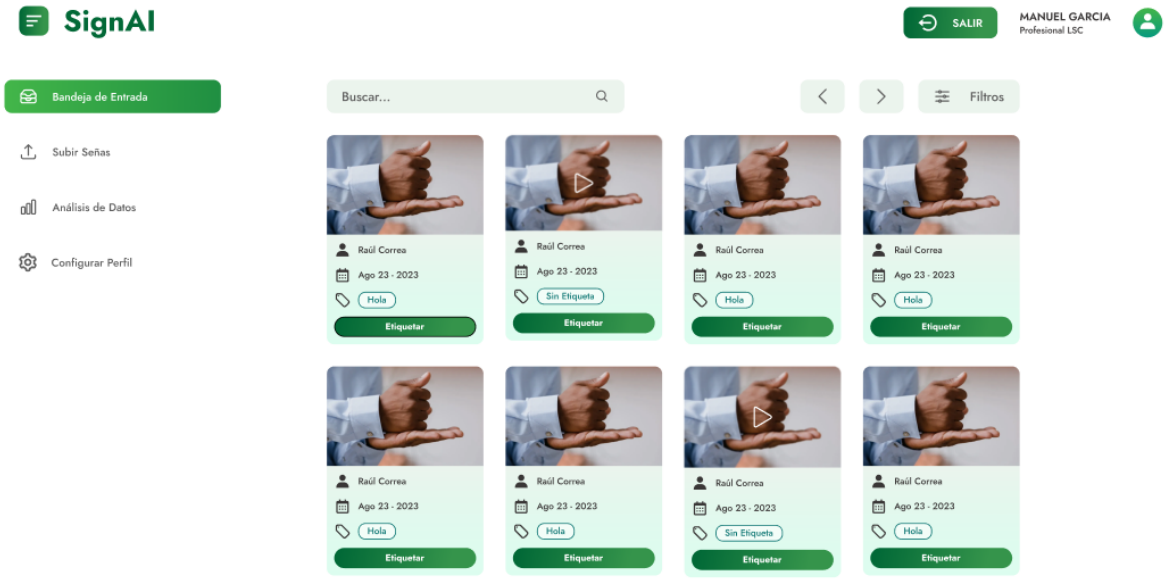

Fig. 31. Módulo para etiquetar señas

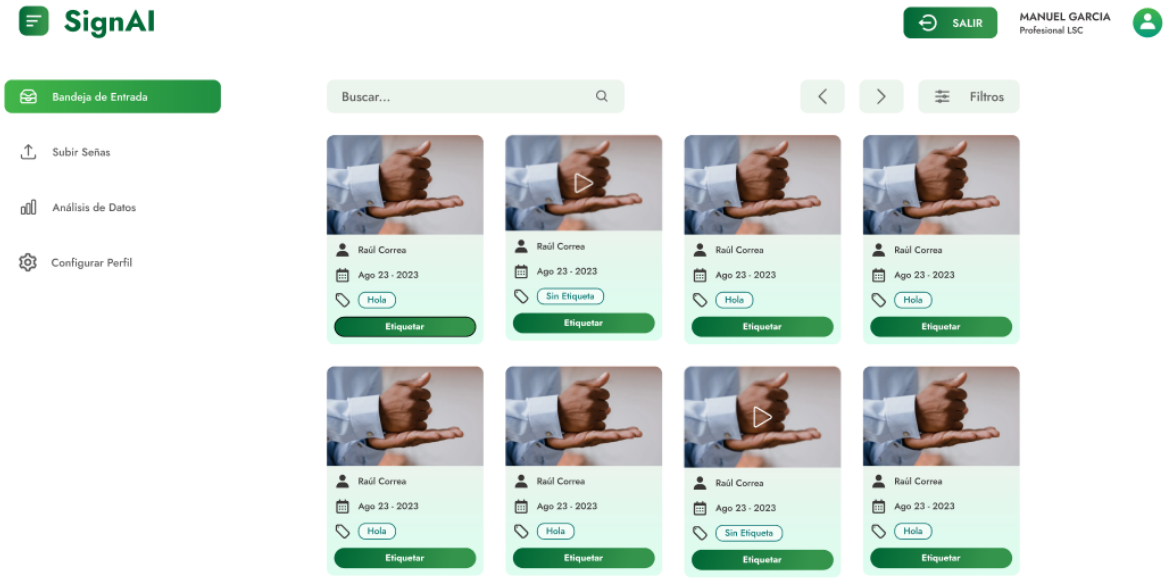

Fig. 32. Módulo para la gestión de modelos e investigación