

Traductor de lenguaje de señas simplificado por una inteligencia artificial

Juan Pablo Sablich¹, Martin Correa², Marcelo Taruschio³, Rodolfo Bertone⁴

¹ Alumno de Grado FACEI UCALP

² Profesor Lic. en Sistemas FACEI UCALP

³ Director de Carrera FACEI UCALP

⁴ Secretario Academico FACEI UCALP

juampi_sablich@hotmail.com, martin.correa@gmail.com,
marcelo.taruschio@ucalp.edu.ar, rodolfo.bertone@ucalp.edu.ar

Abstract. Este trabajo de investigación se centra en aprovechar la evolución tecnológica para mitigar las barreras de comunicación entre personas oyentes y aquellas con discapacidad auditiva. Estas barreras obstaculizan la interacción en entornos educativos, laborales y sociales. Se propone implementar un sistema inteligente de reconocimiento de lenguaje de señas mediante un traductor impulsado por inteligencia artificial (IA), destacándose como un avance tecnológico que mejora significativamente la calidad de vida.

El sistema utiliza redes neuronales previamente entrenadas para mejorar la precisión en diversas tareas, facilitando la comunicación efectiva entre personas con discapacidad auditiva y oyentes. Además de reducir las barreras comunicativas, el sistema busca promover la inclusión social y laboral de las personas con discapacidad auditiva, anticipando mejoras significativas en las relaciones interpersonales y la inclusión en sectores cruciales como la educación, el ámbito social y el trabajo.

El objetivo general consiste en Proveer una herramienta accesible para que cualquier persona, independientemente de su conocimiento previo del lenguaje de señas, pueda comprender las interpretaciones de manera fluida, mediante la implementación de un sistema traductor de lenguaje de señas mediante inteligencia artificial.

Los objetivos Específicos son: Efectuar el entrenamiento de la red neuronal utilizando imágenes capturadas por una cámara para reconocimiento del lenguaje de señas mediante algoritmos de inteligencia artificial, desarrollar una aplicación web que garantice escenarios de calidad para fomentar la comunicación en lengua de señas, utilizando el modelo de inteligencia artificial implementado y validar la usabilidad y utilidad percibida del aplicativo mediante pruebas que incluyan la detección y depuración de errores, considerando la funcionalidad del sistema en diferentes condiciones lumínicas.

Keywords: Traductor Lenguaje de Señas. Sistemas de identificación automática con IA. Lenguaje de Sordos. Aplicación Inclusiva.

1. Introducción

En el reino animal, la comunicación se manifiesta a través de señales químicas, sonidos y gestos, fundamentales para la interacción y supervivencia. Sin embargo, la habilidad humana única de transmitir significados complejos mediante el lenguaje distingue nuestra comunicación de la de otras especies. La comunicación no es solo un rasgo opcional, sino un proceso esencial en la vida misma [1][2].

Las personas con discapacidad auditiva enfrentan desafíos únicos en un entorno predominantemente audiovisual. ¿Cómo pueden participar plenamente en una sociedad donde la comunicación verbal y auditiva es predominante? La falta de acceso al lenguaje verbal representa una barrera significativa en la integración social, limitando el desarrollo humano, educativo y profesional de las personas afectadas [3][4].

El lenguaje de señas se destaca como una herramienta crucial para la comunicación de personas con discapacidad auditiva o dificultades en el habla, rompiendo barreras diarias y facilitando el acceso a la información. Aunque efectivo entre personas sordas, puede no ser comprensible para quienes tienen pérdida auditiva parcial o no están familiarizados con él. Alternativas como el lenguaje escrito y la lectura de labios, aunque utilizadas históricamente, presentan limitaciones significativas en diversos contextos.

El avance tecnológico reciente ofrece soluciones innovadoras como los sistemas de reconocimiento y traducción del lenguaje de señas impulsados por inteligencia artificial (IA). Este trabajo de investigación se enfoca en reducir las barreras de comunicación para personas con discapacidades auditivas mediante la aplicación de algoritmos de IA en la interpretación del lenguaje de señas. Esta iniciativa no solo busca garantizar los derechos, sino también mejorar significativamente la calidad de vida de las personas afectadas.

La propuesta específica implica implementar un sistema capaz de traducir el lenguaje de señas utilizando IA. Este sistema captura imágenes del lenguaje de señas, aprende patrones y gestos específicos, procesa la información y la interpreta para mostrarla en tiempo real en una pantalla o dispositivo. Esta tecnología no solo facilita la comunicación fluida entre personas sordas y oyentes, sino que también promueve una mayor inclusión social y accesibilidad en diversos entornos [5][6].

La inteligencia artificial (IA) ofrece ventajas significativas en la interpretación del lenguaje de señas, como la automatización de procesos, la mejora en la precisión del reconocimiento y traducción de gestos, la reducción del error humano y la eficiencia en el análisis de datos.

Estas capacidades no solo mejoran la eficiencia y la precisión de la comunicación mediante lenguaje de señas, sino que también promueven una mayor inclusión y accesibilidad para personas con discapacidades auditivas en diversos ámbitos sociales y laborales.

2. Revisión del problema

2.1. Antecedentes

La historia del lenguaje de señas se remonta al siglo XVII y puede dividirse en tres etapas cruciales: la antigüedad, los siglos XVIII y XIX y el siglo XX [7]. Inicialmente, el uso del lenguaje de señas era limitado y no reconocido formalmente. Sin embargo, con la creación de escuelas para sordos y el reconocimiento de esta forma de comunicación, su uso se expandió significativamente durante los siglos XVIII y XIX. En el siglo XX, el desarrollo tecnológico ha permitido avances significativos en el tratamiento, almacenamiento y reproducción de imágenes, facilitando la transmisión y aprendizaje del lenguaje de señas.

Diversas investigaciones y proyectos han contribuido al desarrollo de herramientas tecnológicas que interpretan el lenguaje de señas mediante dispositivos visuales y auditivos. Entre ellos, destaca el diseño de guantes electrónicos que transforman los gestos en caracteres y voz artificial, como el proyecto desarrollado en 2016 que utiliza un Arduino MEGA para procesar señales eléctricas[8]. Otros ejemplos incluyen el traductor de la Fundación Hetah, que convierte texto a lenguaje de señas mediante un avatar, y Signalo, una solución multiplataforma que utiliza un avatar en 3D para traducir texto y habla en tiempo real [9].

Proyectos como GANAS y los guantes que hablan han revolucionado la accesibilidad para personas sordas, proporcionando traductores de texto a lengua de signos y dispositivos que captan y traducen los movimientos de las manos. Estos avances han mejorado significativamente la comunicación y la inclusión de personas con discapacidades auditivas, facilitando su interacción en diversos entornos sociales y educativos [10].

2.2. Lenguaje de señas

La lengua de señas posee una estructura gramatical propia y se basa en gestos que refieren a imágenes de cosas, objetos o conceptos a expresar [11]. Es la lengua natural de expresión y configuración gesto-espacial y percepción visual que permite a las personas sordas comunicarse con su entorno social. Basándose en movimientos y expresiones a través de las manos, los ojos, el rostro, la boca y el cuerpo, la lengua de señas es un sistema completo y estructurado. No se trata de un conjunto de gestos desordenados e incompletos, sino de un lenguaje complejo con igual velocidad y profundidad que la lengua oral. Además, la lengua de señas no es universal; cada país tiene su propia versión que puede variar incluso entre regiones. Además de las manos, toda la expresión corporal está involucrada en la comunicación. La lengua de señas incluye giros idiomáticos propios y etapas de adquisición semejantes a las que atraviesan los niños oyentes en el aprendizaje del lenguaje oral.

3. Metodología

3.1 Tipo de estudio

El diseño utilizado para el desarrollo del presente trabajo es un sistema traductor de lengua de señas mediante inteligencia artificial para personas con discapacidad

auditiva, incluyendo a aquellas con discapacidad del habla. Este estudio se enmarca en una investigación experimental, destinada a evaluar si este sistema mejora la comunicación mediante indicadores como el porcentaje de aciertos del sistema bajo diferentes condiciones de iluminación.

3.2 Métodos de investigación

La metodología de investigación se basó en los siguientes aspectos:

- Estudio de Casos: Parte de la problemática identificada en la literatura respecto a las dificultades en el aprendizaje de la lengua de señas y la necesidad de sistemas que faciliten la interacción.
- Cualitativo: Explora el impacto cualitativo de las tecnologías basadas en inteligencia artificial en el acceso a la educación y la cultura para personas con discapacidad.
- Cuantitativo: Evalúa cuantitativamente si la implementación de este sistema mejora significativamente la comunicación entre personas con discapacidad auditiva y oyentes, basándose en datos recopilados.

3.3 Técnicas y fuentes de información

Para la investigación, se emplean las siguientes técnicas:

- Investigación de campo y experimental: Utiliza observación y técnicas de visión artificial para recopilar y analizar datos sobre el comportamiento de los usuarios del sistema.
- Revisión sistemática: Se realiza una revisión exhaustiva de fuentes científicas y técnicas para respaldar el desarrollo del sistema traductor.

Para obtener las fuentes de información se realizó una revisión sistemática de literatura científica y técnica proveniente de sociedades científicas reconocidas y confiables.

3.4. Población y muestra

- Población: Datos obtenidos de los aciertos del sistema bajo diferentes condiciones de luz (buena, regular, mala).
- Muestra: Se selecciona aleatoriamente una muestra simple de la población mencionada, calculando su tamaño de acuerdo a fórmulas apropiadas.

La operacionalización de las variables se resume en la siguiente tabla:

Variable	Concepto	Indicadores	Técnicas de instrumentación
Calidad de la luz	Variable independiente: variable que se pone a prueba a nivel experimental. Es manipulada por el investigador para probar cierta hipótesis.	Número de Luxes (medida de luminosidad). Condiciones de iluminación: Buena, regular y mala.	Sensores de luz Luxómetro
Porcentaje de acierto del sistema	El comportamiento de la variable dependiente se ve afectado por la variable independiente	Número de aciertos del sistema.	Usuario que usa el sistema. Observación directa

3.5. Definición de Hipótesis

- Definición de Hipótesis Nula (**H0**): El diseño e implementación de un sistema traductor de lengua de señas mediante inteligencia artificial no se verá afectado de acuerdo a las condiciones lumínicas.
- Definición de Hipótesis Alternativa (**H1**): El diseño e implementación de un sistema traductor de lengua de señas mediante inteligencia artificial se verá afectado de acuerdo a las condiciones lumínicas.

3.6. Procesamiento y análisis

El sistema se compone de 2 partes:

- Parte 1 se efectúa la fase en la cual se genera una base de datos que contiene los puntos de referencia de las manos con relación al lenguaje de señas. Es la fase de entrenamiento.
- Parte 2 es el programa que aplica la base de datos creada anteriormente. Por medio de una cámara captura la imagen en tiempo real para que después mediante los algoritmos puestos en práctica, compare y determine la letra correspondiente a la lengua de señas.

Para el desarrollo del sistema se ha definido el siguiente proceso:

Para la ejecución del sistema fue imprescindible usar varias librerías, módulos, algoritmos. Para la parte de interfaz gráfica se emplea el módulo PyQt5 que permite crear interfaces gráficas con Python de manera rápida y sencilla. Posee además una interfaz de diseño para crear nuestras interfaces gráficas. Una de sus herramientas es “Designer” o el diseñador que es el programa de la librería que generará nuestro código para que se materialice la ventana con nuestra interfaz gráfica.

El uso de un entorno de desarrollo integrado (IDE) simplifica enormemente la programación y el proceso de desarrollo. La herramienta interpreta lo que el desarrollador escribe y sugiere palabras clave relevantes para insertar, y asigna diferentes colores a los distintos elementos del código.

Un entorno de desarrollo integrado incluye un editor de texto, un editor de proyectos en el que se pueden almacenar archivos, y muchos módulos y paquetes para añadir funcionalidades fácilmente. Para la parte de programación se usa el IDE Spyder que es empleado para desarrollar programación en Python. Su editor de código inteligente ayuda a escribir código de alta calidad. Sus diferentes códigos de colores para las palabras clave, las clases y las funciones aumentan la legibilidad y la comprensión del código. Esto también simplifica la detección de errores. También está incluida la función de autocompletar.

3.7. Obtención de Información

Para el entrenamiento de la red neuronal es necesario adquirir la información que será almacenada, para ello se utiliza MediaPipe Hands. Esta solución permite la detección de manos, con 21 puntos de referencias 3D, reconociendo de qué mano se trata, izquierda o derecha. Para ello MediaPipe emplea aprendizaje automático (Machine Learning), de donde han obtenido múltiples modelos que trabajan juntos, para obtener los resultados que podemos apreciar a continuación.

El sistema está basado en un rastreo de la palma y los dedos de la mano para luego detectar 21 puntos claves 3D en un solo fotograma, es una imagen cinematográfica

considerada aisladamente, por lo que se divide en tres partes. Es decir, un detector de palmas que recorta la forma de la mano del fotograma, un modelo que detecta puntos de referencia en 3D de la imagen recortada y un detector de gestos que clasifica los puntos claves configurados previamente en un set de gestos.

3.8. Entrenamiento Red Neuronal:

Una red neuronal es un modelo que simula el procesamiento de información del cerebro humano. Está compuesta por capas de neuronas interconectadas que reciben datos, procesan información en capas ocultas y producen una salida final. Durante el entrenamiento, ajusta los pesos de cada neurona para mejorar las predicciones, utilizando el algoritmo de retropropagación (backpropagation) para minimizar errores. El modelo matemático de una neurona incluye una suma ponderada de las entradas, que se activa mediante funciones como Relu (para mantenerse inactiva si el valor de entrada es menor a 0) y Softmax (para calcular probabilidades de diferentes clases).

La arquitectura de la red neuronal para este proyecto incluye una capa de entrada con 21x2 unidades para coordenadas X e Y de puntos de referencia de la mano, seguida por dos capas ocultas activadas por Relu y una capa de salida con 27 neuronas activadas por Softmax, representando gestos de la lengua de señas.

TensorFlow y Keras se utilizan para construir y entrenar la red, facilitando el manejo de grandes conjuntos de datos como Keypoint.csv, que contiene gestos de manos previamente entrenados.

4. Implementación

Para llevar a cabo el desarrollo de este algoritmo se utilizó el lenguaje de programación Python utilizando un entorno de programación creado en Anaconda y utilizando Spyder (entorno de desarrollo integrado).

Anaconda es una distribución de los lenguajes de programación Python y R diseñada para aplicaciones de computación científica, como ciencia de datos, aprendizaje automático y análisis de datos a gran escala.

Desarrollada por Anaconda, Inc., fundada por Peter Wang y Travis Oliphant en 2012, su objetivo principal es simplificar la gestión y despliegue de paquetes en Windows, Linux y macOS. La distribución utiliza el sistema de gestión de paquetes conda, que difiere de pip en la gestión de dependencias. Antes de la versión 20.3, pip instalaba dependencias sin verificar conflictos, lo que podía afectar a la compatibilidad de los paquetes instalados.

Conda, en cambio, analiza el entorno actual y las limitaciones de versión para asegurar la compatibilidad de las dependencias, advirtiéndolo cuando no es posible lograrla. Spyder es un entorno de desarrollo integrado (IDE) multiplataforma de código abierto diseñado específicamente para la programación científica en Python.

Desarrollado inicialmente por Pierre Raybaut en 2009 y mantenido desde 2012 por un equipo de desarrolladores científicos de Python, Spyder se integra con importantes paquetes de la pila científica como NumPy, SciPy, Matplotlib, pandas, IPython, SymPy y Cython.

Este IDE ofrece características avanzadas como un editor con resaltado de sintaxis, introspección y finalización de código, soporte para múltiples consolas IPython,

capacidad de explorar y editar variables desde una interfaz gráfica de usuario (GUI), y un panel de ayuda que muestra documentación enriquecida sobre funciones y métodos automáticamente. Incluye un depurador vinculado a IPdb para la ejecución paso a paso, análisis de código estático con Pylint, y un generador de perfiles en tiempo de ejecución para comparar el rendimiento del código.

Spyder también proporciona soporte para proyectos que permiten trabajar en múltiples esfuerzos de desarrollo simultáneamente, un explorador de archivos integrado, función de búsqueda en archivos con expresiones regulares, y un navegador de ayuda en línea para consultar la documentación de Python y paquetes directamente desde el IDE. Además, registra un historial detallado de los comandos de usuario y ofrece una consola interna para la introspección y control del funcionamiento de Spyder. Está disponible en diversas plataformas, incluyendo Windows, macOS (a través de MacPorts) y las principales distribuciones de Linux.

El dataset utilizado para entrenar el modelo de reconocimiento de letras está compuesto por aproximadamente 1788 imágenes. A través de la utilización del dataset obtenido se han aumentado los datos para obtener: Normalización de datos y Modificación de brillo.

Para esta aplicación se ha utilizado el modelo VGG19, que ha sido entrenado sobre el dataset ImageNet. Este modelo consiste en un encoder convolucional constituido por 16 capas convolucionales y 3 capas densas (dos de ellas fully connected y una softmax). Además, cuenta con cinco capas de max pooling.

Añadir complejidad al modelo con estos cambios permitieron que alcanzamos unos buenos resultados. Como hiper parámetros cabe destacar: SGD (stochastic gradient descent) como optimizador, 0.02 de learning rate y bach size de 32 para el modelo de los números y 64 para las letras.

5. Resultados

Para verificar la funcionalidad del prototipo desarrollado, se evaluaron diferentes condiciones lumínicas que se clasificaron en tres categorías: buena, regular y mala. En condiciones de iluminación regular, se estableció un rango de 100 a 200 luxes. Por sobre este rango las condiciones son buenas y por debajo se la considera categoría mala.

Durante las pruebas experimentales, se utilizó un luxómetro específico, en este caso el Lux Light Meter Pro, una aplicación calibrada para medir la intensidad de la luz con precisión profesional. Esta herramienta permite realizar mediciones en tiempo real, proporcionando valores medios y máximos de iluminación tanto en entornos interiores como exteriores.

Para validar la funcionalidad del sistema desarrollado, se estableció un umbral de precisión entre 0,90 y 0,957. Estos valores fueron determinados tras una revisión bibliográfica sobre la tecnología utilizada en la implementación del sistema. Se consideró que el sistema operaba correctamente si la precisión calculada supera el umbral de 0,90; de lo contrario, se interpretó que el sistema había fallado debido a factores específicos.

Además, se presenta a continuación la base de datos obtenida después de completar las pruebas, la cual proporciona una visión detallada del rendimiento del sistema bajo las distintas condiciones lumínicas mencionadas.

Se realizaron 50 pruebas bajo condiciones de iluminación de las tres categorías mencionadas, los resultados obtenidos están a disposición para quien desee consultarlos.

Los datos muestran variaciones según el tipo de iluminación, por lo que se emplea el análisis de varianza de un factor (Anova) para determinar si existen diferencias significativas entre las medias de los distintos grupos lumínicos. Se formulan dos hipótesis para este análisis, con un nivel de significancia establecido en $\alpha = 5\%$: la hipótesis nula (H_0) postula que las medias de los porcentajes de aciertos para cada condición lumínica son iguales, mientras que la hipótesis alternativa (H_1) sostiene que al menos una de las medias de aciertos es diferente. Posteriormente, se verifica la normalidad de los datos utilizando el software SPSS, requisito fundamental para la aplicación adecuada del método Anova de un factor. En este caso también los resultados están a disposición para quien desee consultarlos.

La prueba de normalidad realizada confirma que los datos cumplen con la normalidad, dado que el valor de significancia calculado ($p > 0,05$) es superior al nivel establecido. Con esta validación, se procede a realizar el análisis de varianza de un factor (Anova). Los resultados revelan que el sistema implementado muestra diferencias significativas en su desempeño según la condición lumínica. Esto se debe a que cada nivel de iluminación afecta la capacidad del sistema para detectar gestos, siendo más efectivo en condiciones de buena iluminación donde detecta la mayoría, e incluso todos, de los gestos realizados. Para una interpretación más clara de estos resultados, se genera un gráfico de las medias que ilustra cómo varía el rendimiento del sistema en cada condición lumínica, evidenciando un mejor desempeño en condiciones de buena iluminación y un rendimiento decreciente en condiciones lumínicas menos favorables.

5.1. Análisis de resultados

Después de realizar los cálculos pertinentes y analizar los resultados obtenidos, se observa que el valor estadístico F calculado es mayor que el valor crítico, lo cual conduce al rechazo de la hipótesis nula y a la aceptación de la hipótesis alternativa. Además, el valor de significancia obtenido es menor que el nivel establecido inicialmente, lo que permite afirmar con un 95% de confianza que al menos un par de medias de los aciertos en cada condición lumínica son significativamente diferentes entre sí.

Para identificar las similitudes entre las condiciones lumínicas, se realizaron pruebas post hoc adicionales. Los resultados indican que todas las medias se encuentran en conjuntos diferentes, lo que implica que ninguna de ellas es igual, evidenciando así que el sistema presenta variaciones significativas en su funcionamiento según la iluminación ambiental.

En condiciones lumínicas favorables, el prototipo logra detectar el 94.46% de los gestos realizados, lo cual representa un desempeño notablemente superior a sistemas previos. En condiciones lumínicas regulares, alcanza una detección del 92.07%, también destacada por encima de tecnologías anteriores. En condiciones lumínicas

desfavorables, el sistema logra detectar el 89.15% de los gestos, un resultado que refleja un nivel medio entre sistemas anteriores y la tecnología actual. Estos hallazgos demuestran que el sistema exhibe un mejor rendimiento y alta confiabilidad especialmente en condiciones lumínicas óptimas.

6. Conclusiones

Las conclusiones del estudio enfatizan el uso del procesamiento de imágenes en tiempo real mediante Python, con especial atención a la implementación de la librería MediaPipe Hands para la detección precisa de manos, palmas y dedos. Se realizaron ajustes exhaustivos en la configuración de la red neuronal, manipulando parámetros como tamaño del lote, tamaño del núcleo, iteraciones, número de capas y neuronas, para optimizar el entrenamiento del modelo y garantizar su eficacia operativa.

El algoritmo desarrollado demostró habilidades significativas en el reconocimiento dinámico de manos bajo diversas condiciones de ángulo, fondo e iluminación, respaldado por la capacidad de MediaPipe para colocar de manera precisa los 21 puntos clave en las manos de los sujetos frente a la cámara del ordenador portátil, incluso en condiciones de baja resolución de píxeles.

No obstante, se identificó que el modelo aún no alcanza una precisión del 100% en la traducción de la lengua de señas, debido a similitudes en las formas de las señas y a errores ocasionales en la predicción. Los resultados muestran porcentajes de efectividad que, si bien son aceptables, revelan áreas para mejorar, particularmente en términos de precisión, exactitud y exhaustividad. El análisis manual y los datos cuantitativos subrayan la idoneidad del modelo para la clasificación de cada señal traducida, pero sugieren oportunidades para la expansión del conjunto de datos, incluyendo mayor diversidad en términos de vestimenta, fondos y tonos de piel.

Las recomendaciones para mejoras futuras incluyen la ampliación del conjunto de datos con imágenes adicionales por clase y la incorporación de un vocabulario más extenso, con el objetivo de fortalecer la robustez del modelo. Además, se aconseja el uso de fondos blancos y ropa oscura durante la implementación para optimizar la precisión del reconocimiento tanto de las manos como de las señas interpretadas.

Bibliografía

- [1] Wilson, E. O. (2006). *Naturalist*. Island Press.
- [2] Bradbury, J. W., & Vehrencamp, S. L. (2011). *Principles of Animal Communication*. Sinauer Associates.
- [3] Glickman, N. S., & Glickman, G. (2009). *The Deaf Way II Anthology: A Literary Collection by Deaf and Hard of Hearing Writers*. Gallaudet University Press.
- [4] Marschark, M., & Hauser, P. C. (2008). *Deaf Cognition: Foundations and Outcomes*. Oxford University Press.
- [5] Marschark, M., & Hauser, P. C. (2008). *Deaf Cognition: Foundations and Outcomes*. Oxford University Press.
- [6] Starner, T., & Pentland, A. (1995). Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5), 526-533.
- [7] Wilson, E. O. (2006). *Naturalist*. Island Press.
- [8] Glickman, N. S., & Glickman, G. (2009). *The Deaf Way II Anthology: A Literary Collection by Deaf and Hard of Hearing Writers*. Gallaudet University Press.
- [9] Marschark, M., & Hauser, P. C. (2008). *Deaf Cognition: Foundations and Outcomes*. Oxford University Press.
- [10] Starner, T., & Pentland, A. (1995). Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5), 526-533.
- [11] Sablich Juan pablo Estudio sobre la lengua de señas: estructura gramatical y características. 2023, Trabajo de Grado UCALP FACEI