

Desarrollo de sistema informático para comunicación de personas sordas y oyentes implementando inteligencia artificial (Interprete lenguaje de señas colombiano)

Development of a computer system for communication between deaf and hearing people implementing artificial intelligence (Colombian sign language interpreter)

Recibido: Mayo 6 de 2024

Aprobado: Agosto 6 de 2024

publicación: Septiembre 1 de 2024

Forma de citar: D. F. Rivera Vásquez, J. A. Muñoz Chavez, J. E. Mejía Manzano, "Desarrollo de sistema informático para comunicación de personas sordas y oyentes implementando inteligencia artificial (Interprete lenguaje de señas colombiano)"; Mundo Fesc, vol. 14, no. 30, pp. 449–469, Sep. 2024, doi: 10.61799/2216-0388.1744.

Ing. Diego Fernando Rivera Vásquez 

Ingeniero Mecatrónico,
diegorivera@unicomfaucauca.edu.co,
<https://orcid.org/0009-0001-5657-6899>,
Corporación Universitaria Comfaucauca
Popayán, Colombia.

PhD. Javier Andrés Muñoz Chaves 

Doctor en Física,
jmunoz@unicomfaucauca.edu.co,
<https://orcid.org/0000-0002-9614-2112>,
Corporación Universitaria Comfaucauca,
Popayán, Colombia.

Ms. Julio Eduardo Mejía Manzano 

Magister en Ingeniería Computacional y Matemáticas,
julioe.mejia@unad.edu.co,
<https://orcid.org/0000-0002-5666-9235>,
Corporación Universitaria Comfaucauca,
Popayán, Colombia.

***Autor para correspondencia:**

diegorivera@unicomfaucauca.edu.co



Desarrollo de sistema informático para comunicación de personas sordas y oyentes implementando inteligencia artificial (Interprete lenguaje de señas colombiano)

Resumen

El lenguaje de signos es el principal mecanismo de comunicación utilizado por las personas sordas para transmitir sus ideas, e interactuar en la sociedad, debido a esto solo es posible comunicarse con quien lo dominan. Generando que las personas sordas no pueden tener un libre desarrollo social y cultural dentro de su entorno. Por eso este trabajo está enfocado en proponer una metodología para crear una herramienta virtual que permite comunicar directamente personas sordas, con personas oyentes que no manejen el lenguaje de signos, incorporando metodologías de desarrollo (Ingeniería de software), arquitecturas de software, algoritmos de procesamiento de señales (audio, imagines, video), e inteligencia artificial, para obtener un prototipo de software que apoye procesos de comunicación entre personas sorda y oyente.

Palabras clave: Redes neuronales, Lenguaje de Señas, Ingeniería de Software, Lenguaje de signos, Intérprete de lenguaje de signos.

**Development of a
 computer system
 for communication
 between deaf and
 hearing people
 implementing
 artificial intelligence
 (Colombian
 sign language
 interpreter)**

Abstract

Sign language is the main communication mechanism used by deaf people to convey their ideas and interact in society; therefore, it is only possible to communicate with those who dominate it. Deaf people cannot have free social and cultural development in their environment. Therefore, this work focuses on proposing a methodology to create a virtual tool that allows deaf people to communicate directly with hearing people who do not know sign language, incorporating development methodologies (software engineering), software architectures, signal processing algorithms (audio, image, video) and artificial intelligence to obtain a software prototype that supports communication processes between deaf and hearing people.

Keywords: INeural Networks, Sign Language, Software Engineering, Sign Language, Sign Language Interpreter.

Introducción

Las discapacidades sensoriales relacionadas con problemas de audición se consideran una de las más prevalentes a nivel mundial en la actualidad, afectando a 278 millones de personas, correspondiendo al 4.2% de la población mundial [1].

Aquí se encuentran personas con pérdida de audición desde nacimiento, las cuales no pueden aprender de forma natural el habla, es decir no pueden hacer uso de ningún lenguaje tradicional para comunicarse, esto implica que deban usar el lenguaje de signos, en donde a través de movimientos y posición de las manos indican palabras, letras o números para comunicarse con los demás. Este método les permite expresarse de manera autónoma, pero solo pueden hacerlo directamente con personas que dominen este mecanismo de comunicación, limitándose a las personas sordas muchas veces el acceso a la educación, trabajo o espacios recreativos [2].

Todo esto se debe a la falta de un sistema que apoye procesos de comunicación entre personas sordas con los demás, el cual tengan en cuenta tanto el lenguaje de señas como el lenguaje natural al mismo tiempo, para disminuir la brecha de interacción y comunicación que presentan las personas sordas en su entorno.

En la actualidad, gracias a los avances de la tecnología se ve la posibilidad de crear este tipo de sistemas donde por medio de técnicas de inteligencia artificial se puede obtener algoritmos para la interpretación del lenguaje de signos, a través del procesamiento de imágenes y redes neuronales convolucionales, como se muestra en [3], donde se logra una eficiencia del 94% de efectividad en detección de signos utilizando redes neuronales convolucionales y visión por computadora. También se encuentra tecnología que permite el procesamiento del lenguaje natural, para analizar información escrita, control por voz, transformar la voz a texto o viceversa [4].

Además se tienen las diferentes metodologías de desarrollo de software como son prototipado, cascada y espiral, que permiten crear sistemas inteligentes de forma rápida y precisa, teniendo en cuenta las diferentes etapas del ciclo de vida del software, desde la extracción de requisitos, hasta el desarrollo del sistema [5].

Los sistemas de reconocimiento automático del lenguaje de señas han alcanzado precisiones superiores al 90 % en condiciones controladas, gracias al uso de redes neuronales profundas y sensores como Kinect, lo que representa un avance significativo en la comunicación asistida para personas sordas [6].

Como principal referente en intérpretes de lenguajes de signos e inteligencia artificial se tiene la aplicación "Sign Language Processing" desarrollada por Google, la cual busca crear un traductor de texto a lenguaje de signos en diferentes idiomas, permitiendo a las personas interactuar con el lenguaje de signos [6].

Esta aplicación funciona similar a la herramienta Google Translate, es decir, traducción de palabras u oraciones a lenguajes de signos, sin buscar una interacción directa entre dos personas sordas con personas oyentes. Ahora bien, la presente investigación busca justamente desarrollar en Colombia, una herramienta digital que facilite la interacción, y comunicación entre personas sordas que utilizan el lenguaje de signos colombiano y personas que hablan español, pero se espera que la metodología propuesta pueda replicarse o sirva como base para crear sistemas personalizados para desarrollar lenguaje de signos. El modelo SB-SLR, que logró una precisión de hasta 96,64% en el dataset Include-50, mostrando un desempeño estable y consistente. Este sistema mejora la clasificación de señas en lenguaje de señas mediante aprendizaje profundo, lo que lo convierte en una base sólida para aplicaciones prácticas en traducción automatizada [8].

Metodología

El desarrollo de esta investigación tiene el enfoque de investigación aplicada, es decir, conforme se investiga y encuentra soluciones acordes al problema o requisito en estudio, se implementan. Ahora bien, al ser un proyecto de software se parte por el estudio de las etapas que tiene el ciclo de vida del software [7]. Cabe aclarar que solo se tuvieron en cuenta las etapas que se acoplan a las necesidades del proyecto para obtener un prototipo del sistema, ver Tabla I.

Tabla I. Etapas de la metodología de diseño [7]

<i>Etapa</i>	<i>Descripción</i>
<i>Requisitos Funcionales</i>	<i>Son el punto de partida de un proyecto de software, normalmente es un documento escrito, en donde se declaran las funcionalidades que debe tener el sistema para suplir una necesidad en específico.</i>
<i>Planificación y análisis</i>	<i>Conforme se obtiene los requisitos del sistema, estos se estudian y analizan para determinar las partes del sistema, crear un plan de desarrollo, estudia la viabilidad financiera y tecnológica del proyecto.</i>
<i>Diseño del Software</i>	<i>Esta etapa busca expresar el sistema en estudio con enfoque técnico, por medio de diagramas lógicos y físicos que hacen referencia al comportamiento del sistema, tecnologías necesarias para cada parte del sistema.</i>
<i>Codificación del Software</i>	<i>Teniendo el diseño del sistema definido se realiza la fase de codificación o desarrollo, en donde se programa cada parte del sistema, para obtener un PMV (Producto mínimo viable)</i>

Para la implementación de estas etapas se tiene en cuenta el conjunto de tareas que propone la metodología "ICONIX" [8], que permite incluir las etapas del ciclo de vida del software mencionadas anteriormente.

El levantamiento de requisitos se realizó tomando en cuenta, las percepciones de personas que podrían llegar hacer uso del sistema, es decir, se compartió la idea de crear la herramienta a personas sordas, sus familias, personas con interés en aprender

lenguaje de signos, o personas dispuestas a interactuar por medio de una herramienta digital con personas sordas. Conforme se declaran los requerimientos generales que podría tener el sistema se realiza un análisis que lleva las percepciones de los posibles usuarios, a un plan de investigación y desarrollo que permita diseñar y codificar el sistema en estudio.

Resultados y Discusión

A continuación, se muestran los resultados de las etapas de desarrollo anteriormente descritas, enfocadas al desarrollo de una herramienta digital que disminuya la brecha de interacción de las personas sordas con su entorno.

Levantamiento de requisitos

El levantamiento de requisitos se realizó en dos etapas, primero se interactuó con las personas que presentan discapacidad auditiva completa y sus familias, para determinar las necesidades al comunicarse. Por otro lado, está la percepción de personas oyentes que buscan interactuar con personas sordas y no manejan el lenguaje de señas. Para esta actividad se implementó la técnica de “entrevista” con el cliente o usuario final, una práctica frecuentemente utilizada en la ingeniería de requerimientos, ya que permite identificar las necesidades de las personas por medio de una conversación, integrando preguntas estructuradas o semiestructuradas de manera organizada o sistemática [9].

Se diseñaron dos tipos de entrevistas enfocados en las posibles necesidades de los usuarios. Para el primer grupo, correspondiente a 10 personas sordas entre los 15 a 20 años y sus familias se realizaron las siguientes preguntas, ver Tabla II.

Tabla II. Pregunta entrevista a personas sordas

Pregunta	Respuesta	Resultado
¿Qué es más importante para usted?	1. Transmitir sus ideas y pensamientos. 2. Relacionarse con personas que no utilizan el lenguaje de signos.	1. 10 2. 10
¿Cómo le gustaría interactuar con la herramienta digital?	1. Dispositivo Móvil 2. Portal Web	1. 10 2. 10
¿Qué mecanismo prefiere para recibir información dentro de una conversación con una persona oyente a través de la herramienta?	1. Texto digital o Manuscritos 2. Medios visuales referentes al lenguaje de signos (imágenes, o videos)	1. 2 2. 10
¿En qué escenarios utilizaría continuamente la herramienta? (Marque una sola respuesta)	1. Actividades académicas 2. Actividades laborales 3. Actividades sociales y culturales	1. 6 2. 3 3. 1

Para el segundo tipo de usuario, correspondiente a personas sin dominio del lenguaje de signos, interesados en entablar conversaciones con personas sordas, se realizó un proceso similar al anterior aplicando las preguntas de la Tabla 3 a 20 personas oyentes

entre los 20 a 40 años.

Tabla III. Preguntas entrevista a personas oyentes

Pregunta	Respuesta	Resultado
¿Qué es más importante para usted?	1. Etablir una conversación con personas sordas. 2. Relacionarse con personas sordas. 3. Aprender lenguaje de señas	1. 20 2. 10 3. 20
¿Cómo le gustaría interactuar con la herramienta digital?	1. Dispositivo Móvil 2. Portal Web	1. 20 2. 10
¿Qué mecanismo prefiere para ingresar y recibir información dentro de una conversación con una persona sorda a través de la herramienta?	1. Captura de voz y audio 2. Texto digital	1. 10 2. 10
¿En qué escenarios utilizaría continuamente la herramienta? (Marque una sola respuesta)	1. Actividades académicas 2. Actividades laborales 3. Actividades sociales y culturales	1. 5 2. 5 3. 10

Con base a estas preguntas y comentarios que se dieron en el momento de interacción con los entrevistados, se declara los siguientes requisitos funcionales mínimos generales, que abarcan lo necesario para que el sistema satisfaga la necesidad estudiada, tomando en cuenta los dos perfiles de usuario, las personas sordas (US), y personas oyentes (UO), ver Tabla IV.

Tabla IV Requisitos funcionales generales

Requisitos		
Código	Usuario	Definición
R1	US	Sistema interactivo para recibir información de una conversación mediante implementación de medios audiovisuales referentes al lenguaje de signos, pasar de texto a lenguaje de signos.
R2	UO	Sistema interactivo para interpretar el lenguaje de signos, pasar de signos a texto o voz.
R3	US-UO	Arquitectura para operatividad en dispositivos móviles y páginas web.
R4	US-UO	Interfaz de usuario minimalista

Planificación y análisis

Tomando en cuenta los requerimientos funcionales generales declarados, se realiza un estudio de viabilidad del proyecto mediante investigación y análisis de la literatura. Para esto se realizó un mapeo sistemático de literatura con el método de Petersen, que permite buscar de forma rápida y eficiente información de un tema en específico ingresando cadenas o fórmulas de búsqueda en bases de datos bibliográficas [10].

El mapeo se realizó en temáticas enfocadas a los requerimientos R1 y R2 del sistema, para determinar qué tecnologías o técnicas de desarrollo de sistemas inteligentes permiten cumplirlos, ver Tabla V.

Tabla V. Metodología de búsqueda

Mapeo	Descripción	Cadena de búsqueda
R1	Consulta de proyectos, técnicas, y metodología de procesamiento de imágenes/video, relacionadas al desarrollo de intérpretes de lenguaje de señas.	("Deaf person" OR "deafened" OR "hard of hearing" OR "hearing impaired" OR "Sign Language Recognition" OR "sign language interpreter" OR "sign language translator" OR "Software development approach") AND ("Computer Vision" OR "artificial vision" OR "Machine Learning" OR "artificial intelligence")
R2	Consulta de proyectos, técnicas y metodología de procesamiento de audio para recepción y transmisión de palabras, frases y diálogos (Procesamiento de lenguaje natural)	("dictated words" OR "word recognition" OR "Automatic dictation" OR "word detection" OR "natural language processing") AND ("Software development approach" OR "artificial intelligence" OR "digital audio processing")

La búsqueda de información se realizó en diferentes bases de datos relacionadas a la ingeniería de software y tecnologías, tales como, ScienceDirect, IEEE Xplore, incluso el motor de búsqueda Google Scholar. De esta forma, se lograron obtener alrededor de 147 documentos científicos, 82 para el mapeo R1 y 65 para el mapeo R2, de donde al final se eligieron 33 documentos para el mapeo de R1 y 24 para el mapeo de R2, con la finalidad de extraer información clave que aporta al desarrollo técnico del sistema en estudio.

El mapeo R1 entregó información sobre posibles tecnologías y técnicas de desarrollo de sistemas inteligentes que permiten diseñar, programar y ejecutar sistemas de inteligencia artificial para la interpretación de lenguaje de signos, ver Tabla VI.

Tabla VI. Herramientas para desarrollar algoritmos de intérprete de lenguaje de signo

Mapeo	Técnicas/Ciencias	Herramientas/Tecnologías
R1	-Machine Learning -Redes neuronales artificiales CNN- RNN -Modelos Markov -Deep Learning -Memoria bidireccional (Bi-LSTM) -Procesamiento de imágenes y video -Sistemas expertos	-Microsoft Kinect -TensorFlow -ML.NET -MatLab -Python -OpenCV -Mediapipe -PyTorch

En cuanto al mapeo R2 se obtuvieron las metodologías y tecnología que apoyan el procesamiento de lenguaje natural por medio de algoritmos para procesar información escrita o de voz, ver Tabla VII.

Tabla VII. Herramientas para desarrollar algoritmos de procesamiento de lenguaje natural

Mapeo	Técnicas/Ciencias	Herramientas/Tecnologías
R2	-Segmentación de palabras -Extracción de raíz de palabras -Reconocimiento de palabras aisladas -Redes Neuronales RNN -Derivación de audio -Estudio de espectrogramas -Machine Learning	-Banana PI -Python -PyTorch -C++ -SpeechRecognition -Watson -PyAudio

En cuanto al requisito R3 se toma como punto de partida las arquitecturas de sistemas multiplataforma más comunes, como son, "Arquitectura de 3 capas" y "Arquitectura de microservicios", que manejan los sistemas informáticos en la actualidad [11].

El requisito R4 está enlazado a la arquitectura, por tanto es necesario desarrollar mínimo 2 interfaz de usuario, uno para dispositivos móviles y otra para el entorno web.

Así, con base en los datos recopilados, se puede concluir que en la actualidad se cuentan con los recursos necesarios para llevar a cabo el desarrollo del sistema.

Diseño del Software

El diseño del sistema en estudio está enfocado en desarrollar un producto mínimo viable (PMV), buscando obtener un prototipo del sistema. Para comprender el diseño del sistema se lo expresa mediante diagramas de flujo basado en el modelo C4 para arquitecturas de software [12], donde se presenta el comportamiento del sistema por medio de diagramas jerárquicos que contienen información detallada conforme se avanza en cada nivel del modelo, ver Tabla VIII.

Tabla VIII. Componentes de modelos C4 para arquitecturas de software

Nivel C4	Nombre	Descripción
1	Contexto	Diagrama donde se expresa el contexto del sistema, muestra la interacción de las personas que lo utilizan junto a los sistemas del software.
2	Contenedor	Este diagrama muestra los contenedores (aplicaciones, bases de datos, microservicios, etc.) que componen el sistema.
3	Componentes	El diagrama de componentes, especifica un contenedor individual para mostrar sus componentes.
4	Código	Son diagramas de código, para ampliar cada componente y mostrar cómo se podría implementar.

Partiendo por el primer nivel del modelo, en la Ilustración 1 se indica el diagrama de contexto del sistema y la interacción que tendría los dos tipos de usuario con el sistema, en donde las personas sordas ingresan su lenguaje de signos, y esta los convierte en texto o mensaje de audio a la persona oyente. Vale destacar que también se puede interactuar de manera inversa, es decir, la persona oyente ingresa notas de voz o texto y estas se reflejan en lenguaje de signos para la persona sorda.

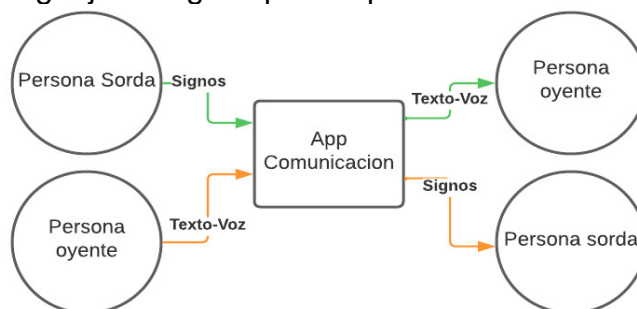


Ilustración 1. Diagrama de contexto del sistema

En la Ilustración 2 se presenta el nivel 2 del modelo, donde se indica a mayor detalle los contenedores de "App Comunicación". Aquí, los contenedores "Aplicación web" y "Aplicación Móvil" hacen referencia a la interfaz de usuario del sistema; el contenedor "Api Rest", hace referencia a la interfaz de comunicación por donde se enlaza la información de los módulos de un sistema a sus diferentes interfaces [13].

Este último, se utiliza para conectar y dirigir el flujo de información que ingresa al sistema, por ejemplo, cuando una persona oyente envía un texto o nota de voz a través de cualquier interfaz, el "Api Rest" lleva esa información al contenedor "Modulo Texto-Voz a Signos"; el cual se encarga de obtener la traducción del mensaje en lenguaje de signos y el resultado se envía nuevamente por "Api Rest" a la interfaz de usuario para ser visualizada por la persona sorda.

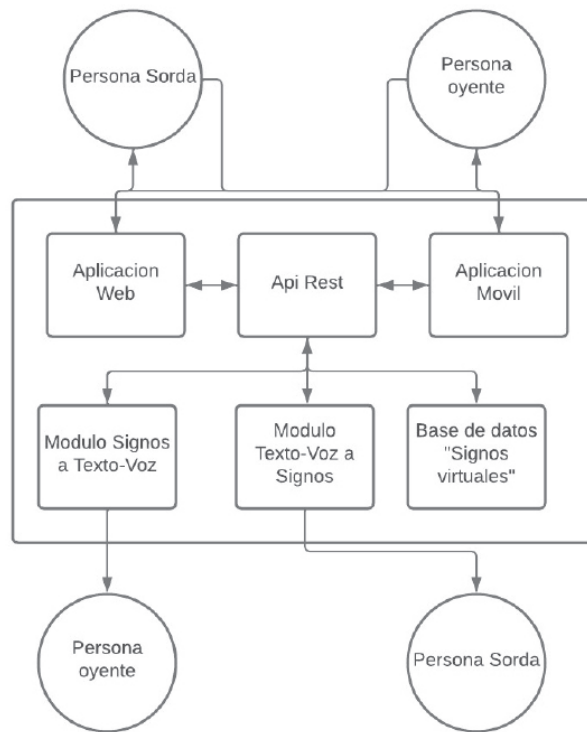


Ilustración 2. Diagrama de contenedores del sistema

El contenedor Base de datos "Signos Virtuales", hace referencia a una colección de videos e imágenes que representan los signos del lenguaje de señas colombiano y están etiquetados directamente a las palabras, números y letras que representan.

Para los diagramas de tercer nivel solo se indican los contenedores de "Modulo Signos a Texto-Voz"; y "Modulo Texto a Signos" ver Ilustración 3, que son parte funcional del sistema. La configuración y desarrollo de los contenedores "Aplicación web", "Aplicación Móvil", y "Api Rest" que hacen referencia a la parte operativa del sistema se implementan por medio del patrón "Modelo Vista Controlador (MVC)" [14].

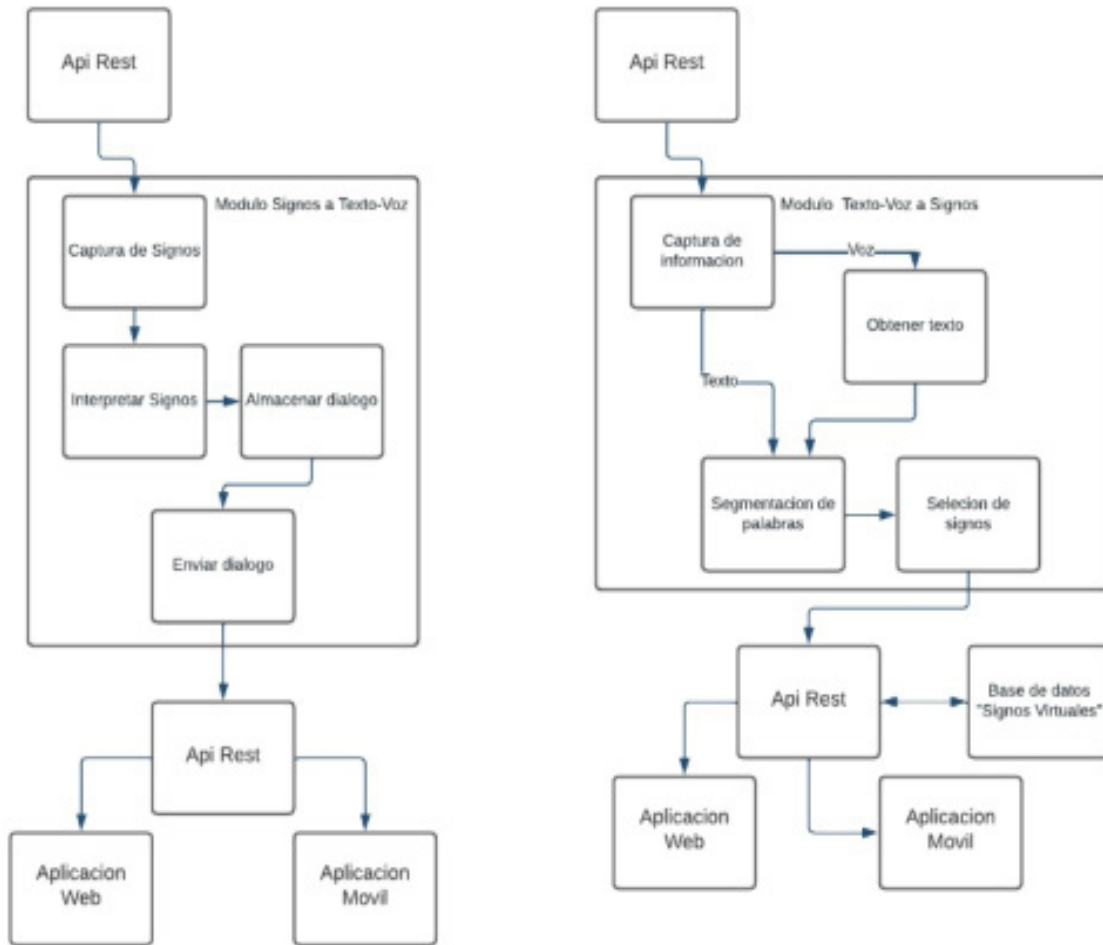


Ilustración 3. Diagramas componentes del sistema

En la Ilustración 4, se muestra el diagrama de flujo del componente principal de “Modulo Signos a Texto-Voz”, haciendo referencia al nivel 4 del modelo, los demás componentes tienen funcionalidades básicas definidas, ver Tabla IX.

Tabla IX. Funcionalidad definida de componentes estáticos del módulo Signos a Texto-Voz

Nombre	Descripción
Captura de Signos	Acceder a la cámara del dispositivo (Computador, Celular)
Almacenar diálogo	Guardar en un vector el diálogo (Texto)
Enviar diálogo	Enlazar el módulo con la Api Rest para enviar la información

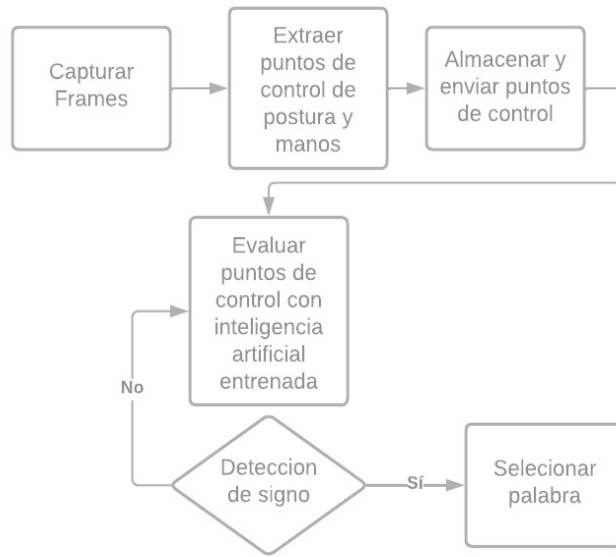


Ilustración 4. Diagrama de código intérprete Signo [Autor]

En el “Módulo Texto-Voz a Signos”, se tiene dos componentes principales que son, “Segmentación de palabras”, y “Obtener texto” los diagramas de código se indican en la Ilustración 5 y los demás cumplen funciones estáticas definidas en la Tabla X.

Tabla X. Funcionalidades definidas de componentes estáticos del Módulo Texto-Voz a Signos [Autor]

Nombre	Descripción
Captura de información	Acceder al micrófono o teclado del dispositivo (Computador, Celular)
Selección de signos	Acceder a la ruta donde esta almacenado la ayuda audiovisual del signo que representa el dialogo ingresado.

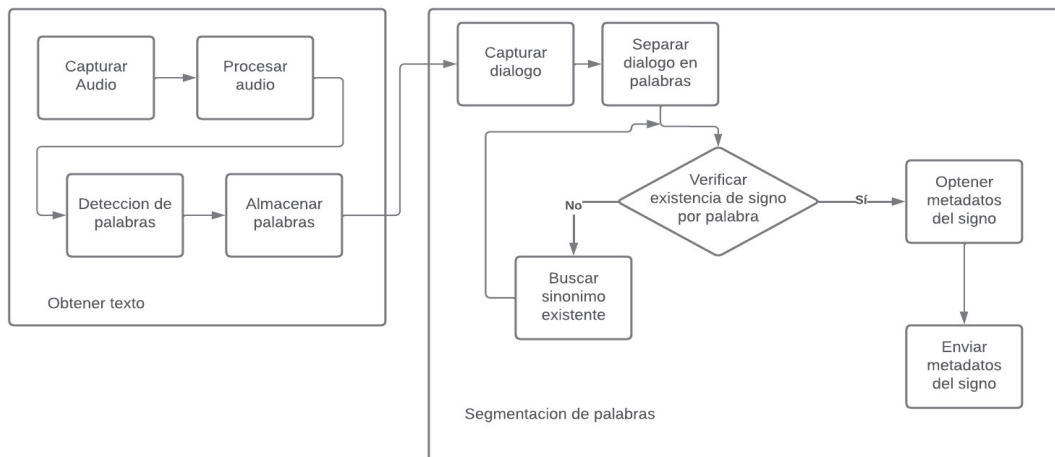


Ilustración 5. Diagrama código componentes modulo Texto-Voz a Signos [Autor]

Codificación del Software

A medida que se planifica y evalúa el sistema, se decide abordar su desarrollo dividiéndolo en cuatro módulos de programación claramente establecidos, como se detalla en la Tabla XI

Tabla XI. Módulos del sistema en estudio [Autor].

Módulo	Descripción
Intérprete LSC	Algoritmo encargado de detectar el lenguaje de señas por medio de un dispositivo móvil y su cámara (Interpretar Signos).
Transcribir Voz a Texto	Algoritmo encargado de transcribir una conversación dictada a un archivo de texto, por medio del micrófono del dispositivo móvil (Obtener Texto).
Convertir Texto a LSC	Algoritmo encargado de convertir el Texto a LSC apoyado en ayudas audiovisuales que representan el lenguaje de señas colombiano (Segmentación de palabras).
Interfaz de usuario	Módulo de desarrollo de la interfaz de usuario (Aplicación web, Aplicación móvil, Api Rest)

Intérprete LSC

El lenguaje de señas colombiano se compone por dos tipos de signos, los estáticos que representan el abecedario o los números, y los signos compuestos que hacen referencia a una serie de movimientos corporales que indican una palabra [15]. Estos últimos son los comúnmente más utilizados ya que permiten a personas sordas comunicarse de manera rápida. Tomando esto en cuenta, junto con la información recolectada en el mapeo sistemático aplicado, este módulo se desarrolla en el lenguaje de programación Python, implementando redes neuronales recurrentes modelo "LSTM", debido a que pueden ser programas para el procesamiento de video y detección de objetos, patrones, o signos en tiempo real [16]. Vale destacar que esto permitirá entrenar la red neuronal con videos realizando los diferentes signos del lenguaje de señas colombiano, sean estáticos o compuestos.

Para el entrenamiento de la red neuronal LSTM se utilizaron 30 videos de 30 segundos por cada signo, con el fin de extraer puntos de control que indican la postura y posición de manos de la persona durante los 30 fotogramas que capturan todo el movimiento que lleva cada signo. De esta forma, la red puede hacer un seguimiento continuo de la posición y ubicación de las manos, para identificar el signo. Los signos de estudio fueron los pertenecientes al abecedario colombiano y a signos básicos que entrega el Instituto Nacional para Sordos-INSOR que permiten saludar, conocer pronombres, verbos, números y días de la semana, para entablar una comunicación básica entre personas sordas y oyentes [17].

Este algoritmo se compone por tecnologías que permiten realizar el procesamiento digital de imágenes y video, junto a herramientas para entrenar redes neuronales. Estas herramientas se describen en la Tabla XII.

Tabla XII. Tecnologías para el algoritmo Intérprete LSC [Autor]

Tecnología	Funcionalidad
OpenCV	Captura y procesamiento de los fotogramas del video
TensorFlow y Keras	Programas y entrenar modelo LSTM
MediaPipe	Extracción de puntos de control (postura y posición de manos) de los fotogramas
Numpy	Almacenar los metadatos de los puntos de control
Matplotlib	Imprimir los puntos de control

Transcribir Voz a Texto

Este módulo se desarrolló gracias a la herramienta "Speech Recognition" de Google, que permite acceder al micrófono del dispositivo para capturar audios y luego guardarlo en un archivo de texto [18]. El funcionamiento es sencillo, la persona graba una nota de voz, esta es recibida y procesada por "Speech Recognition", la cual obtiene el diálogo y se guarda en una variable o en un archivo ".txt" y objeto ".json".

Convertir Texto a LSC

Este módulo permite hacer una visualización en LSC de la nota de voz o texto que ingresa la persona oyente, es decir, toma de entrada el archivo de texto o la variable con el mensaje y realiza una petición por medio de la "Api Rest" a la base de datos con las ayudas audiovisuales, que corresponden a los signos necesarios para traducir el contenido de texto a LSC. Para realizar este módulo se empieza por construir la estructura de datos llamada "diccionario" de Python ya que permite almacenar un carácter o cadenas de caracteres como clave o identificador único de un elemento asociando variables del sistema al contenido de las bases de datos [19].

Es decir, se almacenan las palabras, y el abecedario que representa cada una de las ayudas visuales LSC con las cuales se entrenó la red LSTM para hacer identificación de signos.

Para hacer la traducción de texto a LSC, se implementaron técnicas de procesamiento de lenguaje natural para realizar, extracción de características de texto, descomposición de textos y análisis semántico por medio de algoritmos de Deep learning, que permiten hacer simplificación de texto[20].

Este proceso se realiza con el fin de poder llevar a la mínima expresión una oración o párrafo digital, de esta forma se obtienen las palabras clave mínimas necesarias para transmitir la información y hacer la búsqueda de estas dentro del diccionario. A continuación se realiza la petición de signos a la base de datos y se imprimen en pantalla. Este módulo se realizó implementando las tecnologías presentes en la Tabla XIII.

Tabla XIII. Tecnologías del algoritmo Convertir Texto a LSC [Autor]

Tecnología	Funcionalidad
PyTorch-NLP	Entrenamiento de redes neuronales recurrentes para etiquetado de metadatos de signos a palabras.
Base64	Codificación y decodificación de ayudas audiovisuales.
PyMongo	Conexión a la base de datos.
BytesIO	Leer la información codificada imágenes o videos.
Numpy	Apoyar la consulta de bases de datos y manejo de vectores.
Matplotlib	Visualizar resultado de las consultas.
OpenCV	Visualizar las ayudas audiovisuales

Interfaz de Usuario

La interfaz de usuario se desarrolló utilizando cuatro tecnologías que integran el desarrollo web, móvil y bases de datos del sistema, ver Tabla XIV.

Tabla XIV. Tecnologías para el desarrollo de la interfaz de usuario [Autor]

Tecnología	Funcionalidad
NodeJS	Desarrollar la aplicación web del sistema por medio de HTML y CSS en Visual Studio Code
Kotlin	Desarrollar la aplicación móvil por medio de Android Studio
Mongo DB	Desarrollo de la base de datos de ayudas audiovisuales de signos
FastAPI	Desarrollo de APIs del sistema

En Ilustración 6 se puede observar el resultado del desarrollo de la aplicación web del sistema indicando cada una de las pantallas que la componen.

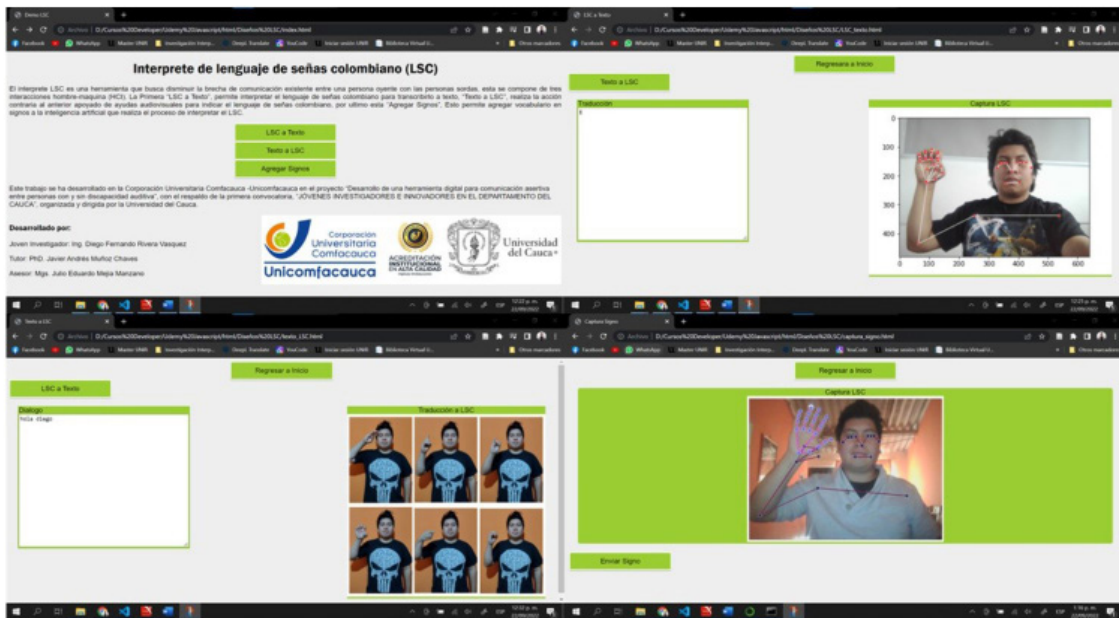


Ilustración 6. Pantallas de la interfaz de usuario de la aplicación web

Por último, la Ilustración 7 hace referencia a las pantallas que componen la aplicación móvil del sistema.



Ilustración 7. Pantallas de la interfaz de usuario de la aplicación móvil

Pruebas de concepto

Con el fin de probar y validar el prototipo desarrollado se realizaron pruebas de concepto (PoC) las cuales permiten identificar índices de usabilidad del sistema, fallas y eficiencia. Para estas pruebas se puede vincular al cliente o mercado objetivo de la aplicación en ejercicios de entrevistas y pruebas directas del producto en desarrollo [21], esto ayuda a obtener una retroalimentación del grupo focal de interés, en donde se pueden identificar mejoras funcionales, visuales y de rendimiento del sistema.

Protocolo de pruebas

En primera instancia, se define el protocolo de las pruebas de concepto que describe los pasos de implementación de la prueba.

Primero se establece que tipo de conversación tendrán los usuarios por medio de la herramienta para determinar su eficiencia. Para esto se definen dos escenarios que se describen a continuación.

1. Diálogo definido (DD): Se entrega un diálogo definido a los usuarios para que interactúen a través de la herramienta, estos diálogos están diseñados teniendo en cuenta los signos con los que cuenta el sistema.

- Diálogo aleatorio (DA): Se solicita a los usuarios que generen oraciones breves utilizando los signos y palabras de su elección.

En cada caso, se llevaron a cabo tres conversaciones para cada usuario, comenzando con la interpretación del lenguaje de signos de las personas sordas. Al finalizar la interacción con el sistema, se les formularon cuatro preguntas relacionadas con la satisfacción y eficiencia. Este proceso se realizó con 10 personas sordas y sus familias que participaron en la definición de requerimientos del sistema, es decir, se realizaron 10 pruebas por cada tipo de diálogo. Al finalizar cada ejercicio respondieron una encuesta y dieron sus observaciones pertinentes. En la Tabla XV se puede ver el resultado de esta actividad.

Tabla XV. Resultado pruebas de concepto

Preguntas	R/US		R/UO		Observaciones/Conclusiones
	DD	DA	DD	DA	
Del 1 al 5, donde 5 es muy eficiente y 1 es poco eficiente, en cuanto califica la eficiencia del prototipo	1- 0 2- 0 3- 7 4- 3 5- 0	1- 0 2- 8 3- 2 4- 0 5- 0	1- 0 2- 2 3- 6 4- 2 5- 0	1- 0 2- 4 3- 6 4- 0 5- 0	Los usuarios manifestaron que el prototipo es funcional, pero que hace falta agregar más vocabulario al sistema, esto porque en los diálogos aleatorios el sistema no logró hacer eficientemente la comunicación.
Del 1 al 5, donde 5 es excelente y 1 mala, en cuanto califica la experiencia al interactuar con la interfaz del prototipo.	1- 0 2- 0 3- 2 4- 8 5- 0	1- 0 2- 0 3- 2 4- 8 5- 0	1- 0 2- 0 3- 0 4- 10 5- 0	1- 0 2- 0 3- 0 4- 10 5- 0	-Interfaz simple de utilizar -Buena paleta de colores -Debería agregarse un manual de usuario gráfico
En cuanto a rapidez del sistema para recibir y entregar la información, en cuanto calificaría el prototipo del 1 al 5 donde 5 es excelente y 1 mala.	1- 0 2- 0 3- 4 4- 6 5- 0	1- 0 2- 0 3- 4 4- 6 5- 0	1- 0 2- 0 3- 5 4- 5 5- 0	1- 0 2- 0 3- 5 4- 5 5- 0	En algunos casos, el sistema requiere que los movimientos del signo se realicen de forma precisa y clara, para identificarlo lo cual hace que la transmisión de información sea lenta -En cuestión de pasar texto a lenguaje de signos, el sistema es funcional, pero tiene un periodo de espera que se debería mejorar
Utilizará la versión actual del prototipo en casos reales de comunicación.	Si- 2 No-8	Si- 2 No-8	Si-5 No-5	Si-5 No-5	-En su mayoría los usuarios no están satisfechos para utilizar el sistema de forma frecuentemente debido a la falta de vocabulario del sistema. -Las personas que dijeron que sí, solo la utilizarían en espacios sociales familiares en donde el sistema sería un apoyo al proceso de comunicación mas no el medio directo de este.

Trabajos futuros

A continuación, se describen los posibles trabajos futuros que se pueden abordar después de esta investigación.

- Implementar requerimientos no funcionales al diseño del sistema para declarar requisitos referentes a acceso de usuarios, seguridad, mantenimiento y escalabilidad

para pasar de un PMV a un sistema SaaS (Software as a Service).

2. Realizar ajustes de funcionalidad que sugirieron las personas en los casos de estudio para mejorar el prototipo, como aumentar vocabulario al sistema, optimización de código para mejorar tiempos de búsqueda y entrega de información.
3. Diseñar pruebas de estudios de casos con enfoque educativos, laboral y social de uso de la herramienta, es decir, pruebas con mayor complejidad en entornos controlados como un salón de clase o una reunión laboral o social.
4. Diseñar la infraestructura del sistema para calcular el presupuesto de despliegue y mantenimiento del sistema teniendo en cuenta los requerimientos no funcionales y funcionales de este.

Conclusiones

Al terminar el desarrollo del sistema para comunicación de personas sordas y oyente implementando inteligencia artificial, se puede concluir que:

Para el desarrollo de proyectos que involucren el desarrollo de software, es importante realizar una investigación previa para relacionar las tecnologías disponibles con los requerimientos del sistema, esto con el fin de cumplir con cada uno de ellos y/o indicar al usuario cuales no se podrán abordar, buscando delimitar el alcance de la solución.

Las técnicas de inteligencia artificial, tal como las redes neuronales recurrentes, permiten desarrollar algoritmos funcionales para la interpretación de lenguaje de signos, y procesamiento de lenguaje natural que permite crear una interacción directa entre personas sordas y personas oyentes sin manejo del lenguaje de signos. Sin embargo, la eficiencia de este proceso está limitada por el vocabulario que tiene el sistema, esto se evidenció en base a los resultados obtenidos por los usuarios en las pruebas de diálogos aleatorios, donde se identificó carencia de algunas palabras que el sistema no pudo interpretar ni transmitir.

El desarrollo de proyectos de software en modalidad de investigación también requiere el estudio de técnicas de ingeniería de software, como el ciclo de vida del software, patrones de diseño del modelo C4, y modelo de vista controlador, para diseñar un plan de trabajo que acople el tiempo de investigación a obtener un PMV (Producto mínimo viable).

Al momento de interactuar las personas sordas con el sistema, se nota que el sistema requiere de mayor tiempo, que le toma a ellos realizar los signos. Es decir, mientras el sistema requiere que el movimiento del signo sea de alrededor de 15 a 30 segundos para identificarlo, las personas sordas puede hacer de 3 a 5 signos en el mismo tiempo, lo cual genera una interacción pausada con la persona oyente, pero conforme avanza

la tecnología y las diferentes estrategias de inteligencia artificial, los tiempos de interpretación de signos podrían disminuir dando una mejor experiencia al usuario.

Referencias

- [1] Gobierno de Colombia "Recursos recomendados para población sorda": <http://www.colombiaprende.edu.co/agenda/actualidad/diamundialdelaaudicion> (accedido 13 de febrero de 2025).
- [2] M. Á. S. Durango, "Estrategias para la inclusión de estudiantes sordos en la educación superior latinoamericana", *Ratio Juris UNAULA*, vol. 13, n.º 26, Art. n.º 26, sep. 2018, doi: 10.24142/raju.v13n26a9.
- [3] V. J. Schmalz, "Real-time Italian Sign Language Recognition with Deep Learning", en *CEUR Workshop Proceedings*, CEUR Workshop Proceedings, ene. 2022. Febrero de 2023. [En línea]. Disponible en: <https://lirias.kuleuven.be/3677145>
- [4] G. DIŞKEN, L. SARIBULUT, Z. TÜFEKÇİ, y U. ÇEVİK, "Real-Time Speaker Independent Isolated Word Recognition on Banana Pi", en *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, jun. 2018, pp. 1-4. doi: 10.1109/ECAI.2018.8679024.
- [5] E. G. Maida y J. Pacienza, "Metodologías de desarrollo de software", 2015, Accedido: 13 de febrero de 2025. [En línea]. Disponible en: <https://repositorio.uca.edu.ar/handle/123456789/522>
- [6] Y. Zhang, Y. Han, Z. Zhu, X. Jiang, Y. Zhang, "Artificial intelligence in sign language recognition: A comprehensive bibliometric and visual análisis", *Computers and Electrical Engineering*, Volume 120, Part C, 2024, 109854, ISSN 0045-7906.
- [7] M. E. Navarro, M. P. Moreno, J. Aranda, L. Parra, J. R. Rueda, y J. C. Pantano, "Integración de arquitectura de software en el ciclo de vida de las metodologías ágiles", presentado en *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*, sep. 2017. Accedido: 13 de febrero de 2023. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/62077>
- [8] S. Renjith, M.S. Sumi, M. Rashmi, "An effective skeleton-based approach for multilingual sign language recognition", *Engineering Applications of Artificial Intelligence*, Volume 143, 2025, 109995, ISSN 0952-1976
- [9] R. A. Paldês, E. D. Canedo, F. de A. Guimarães, y A. T. S. Calazans, "Functional Requirements Elicitation in IoT Systems: a follow-up study", en *Proceedings of the XIX Brazilian Symposium on Software Quality*, en SBQS '20. New York, NY, USA: Association

- for Computing Machinery, mar. 2021, pp. 1-10. doi: 10.1145/3439961.3439975.
- [10] A. Solis y J. Hurtado, "Reutilización de software en la robótica industrial: un mapeo sistemático", *Revista Iberoamericana de Automática e Informática industrial*, vol. 17, n.º 4, Art. n.º 4, sep. 2020, doi: 10.4995/riai.2020.13335.
- [11] M. E. Navarro, M. P. Moreno, J. Aranda, L. Parra, J. R. Rueda, y J. C. Pantano, "Selección de metodologías ágiles e integración de arquitecturas de software en el desarrollo de sistemas de información", en *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*, sep. 2017. Accedido: 13 de febrero de 2023. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/62179>
- [12] A. Vázquez-Ingelmo, A. García-Holgado, y F. J. García-Peñalvo, "C4 model in a Software Engineering subject to ease the comprehension of UML and the software", en *2020 IEEE Global Engineering Education Conference (EDUCON)*, abr. 2020, pp. 919-924. doi: 10.1109/EDUCON45650.2020.9125335.
- [13] V. Ranga y A. Soni, "API Features Individualizing of Web Services: REST and SOAP", *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, ago. 2019, doi: 10.35940/ijitee.I1107.0789S19.
- [14] C. Ávila Garzón, "Modelo Vista Controlador, MVC (Model, View, Controller)", dic. 2019, Accedido: 13 de febrero de 2023. [En línea]. Disponible en: <https://repositorio.konradlorenz.edu.co/handle/001/1528>
- [15] L. A. Tovar, "La definición en la lengua de señas colombiana (LSC)", *Lenguaje*, vol. 45, n.º 2, pp. 383-417, dic. 2017, doi: 10.25100/lenguaje.v45i2.5277.
- [16] Y. Yu, X. Si, C. Hu, y J. Zhang, «A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures», *Neural Computation*, vol. 31, n.º 7, pp. 1235-1270, jul. 2019, doi: 10.1162/neco_a_01199.
- [17] Ministerio de Educación de Colombia, "INSOR | Instituto Nacional para Sordos – Trabajando por la Población Sorda Colombiana" (accedido 9 de junio de 2023). [En línea]. Disponible en: <https://www.insor.gov.co/home/>
- [18] J. Li, "Recent Advances in End-to-End Automatic Speech Recognition", *SIP*, vol. 11, n.º 1, 2022, doi: 10.1561/116.00000050.
- [19] Simplilearn, "What Is A Dictionary In Python?", *Simplilearn.com*, 1 de agosto de 2022. (accedido 9 de junio de 2023). [En línea]. Disponible en: <https://www.simplilearn.com/dictionary-in-python-article>
- [20] Ayisha Tabassum y Dr. Rajendra R. Patil, "A Survey on Text Pre-Processing & Feature

Extraction Techniques in Natural Language Processing", *International Research Journal of Engineering and Technology (IRJET)*, vol. 07, n.º 6, pp. 4864- 4867, jun. 2020, e-ISSN: 2395-0056

[21] K. Iannace, "Proof of Concept: 5 Steps for Successful Software Development", Designli Blog, (accedido 10 de marzo de 2025). [En Línea]. Disponible en: <https://designli.co/blog/5-steps-proof-concept-successful-software-development/>