INVESTIGACIÓN & DESARROLLO, Vol. **25**, No. **1**: 97 – 105 (2025)
ISSN 2518-4431

LENGUAJE FORMAL DE DESCRIPCIÓN PARA LA LENGUA DE SEÑAS BOLIVIANA (LSB): UNA HERRAMIENTA ABIERTA PARA SU USO EN OTRAS LENGUAS DE SEÑAS

A FORMAL DESCRIPTION LANGUAGE FOR BOLIVIAN SIGN LANGUAGE (BSL): AN OPEN TOOL FOR USE IN OTHER SIGN LANGUAGES

Mayra Oropeza-Condori[1], Miguel Frade-Flores[1], Marcel Barrero Mendizábal[1] y Cecilia Tapia-Siles[2,]
[1]Facultad de Ingenierías y Arquitectura (FIA)
[2]Laboratorio de Innovación Tecnológica Industrial y Robótica (LITIR)
Universidad Privada Boliviana, Cochabamba, Bolivia
ceciliatapia@upb.edu
(Recibido el 15 de junio 2025, aceptado para publicación el 31 de julio 2025)

RESUMEN

En el contexto de la comunicación para personas sordas, se presenta el desarrollo de un lenguaje formalizado y un compilador para facilitar la realización de animaciones de señas de la Lengua de Señas Boliviana (LSB), con potencial de adaptación a otras Lenguas de Señas. El sistema emplea un lenguaje específico para el dominio (DSL - del inglés: Domain Specific Language) que define cada seña como una secuencia de poses basadas en vectores y variables finitas, integrando funciones centrales de repetición y velocidad para facilitar su escritura. Los resultados destacan el potencial de su utilización para mejorar la accesibilidad y escalabilidad de herramientas enfocadas en la comunidad Sorda, para que no se vea limitada por la lengua de su región, estableciendo una base para futuros avances en las tecnologías de asistentes para usuarios Sordos.

Palabras Clave: Sordera, lengua de señas, animación, lenguaje, compilador, Python, LSB

ABSTRACT

Within the context of communication for deaf individuals, the development of a formalized language and a compiler is presented, serving to facilitate the creation of animations for Bolivian Sign Language (LSB – from Spanish: Lengua de Señas Boliviana) signs, and is adaptable for use in other sign languages. The system uses domain-specific language (DSL) that defines each sign as a sequence of poses based on vectors and finite variables, integrating central repetition and speed functions to facilitate writing. The results highlight the potential of its use to improve the accessibility and scalability of tools focused on the Deaf community, so as not to be limited by the language of their region, establishing a basis for future advances in assistive technologies for Deaf users.

Keywords: Deafness, sign language, animation, language, compiler, Python

1. INTRODUCCIÓN

Las personas Sordas prelocutivas en Bolivia enfrentan importantes barreras en la comprensión lectora del español escrito. A diferencia de las personas oyentes, quienes adquieren el lenguaje de manera natural a través de la audición, los Sordos prelocutivos deben desarrollar primero la Lengua de Señas Boliviana (LSB) como base e intermediario para luego aprender el español escrito como segunda lengua. Debido a ello, la falta de un dominio sólido de la LSB puede ocasionar una mayor dificultad para comprender el español escrito, ya que este proceso requiere estrategias pedagógicas especializadas y una enseñanza bilingüe adecuada [1].

El acceso a una educación de calidad para esta comunidad es fundamental. Sin embargo, se ve limitado por la escasez de materiales adaptados y poca oferta de personal capacitado. Además, sin contar los aspectos ya mencionados, el español escrito sigue siendo una barrera significativa para muchas personas Sordas, ya que su estructura gramatical y morfosintáctica difiere de la LSB, lo que complica su aprendizaje sin la mediación de herramientas y apoyo adecuados [2].

Ante esta problemática, se hace evidente la necesidad de desarrollar soluciones tecnológicas que faciliten la interpretación del español escrito en un formato accesible para los Sordos prelocutivos.

1.1 Estado del arte – soluciones actuales

El desarrollo de herramientas tecnológicas dirigidas a la comunidad Sorda ha avanzado en distintas regiones del mundo [3], [4], [5], sin embargo, muchas de estas soluciones presentan una limitación fundamental: están diseñadas para una lengua de señas específica y su adaptación a otra lengua requiere repetir el mismo proceso de creación de las animaciones señantes ya que el material es poco compatible y reutilizable, aún en Lenguas de Señas similares procesos costosos y

DOI: 10.23881/idupbo.025.1-8i

complejos (ver Figura 1). A diferencia del español o el inglés, que pueden compartirse en múltiples países con variaciones mínimas, cada lengua de señas es única, con gramáticas, léxicos y estructuras propias que dependen del contexto sociocultural de la comunidad que la utiliza [6].

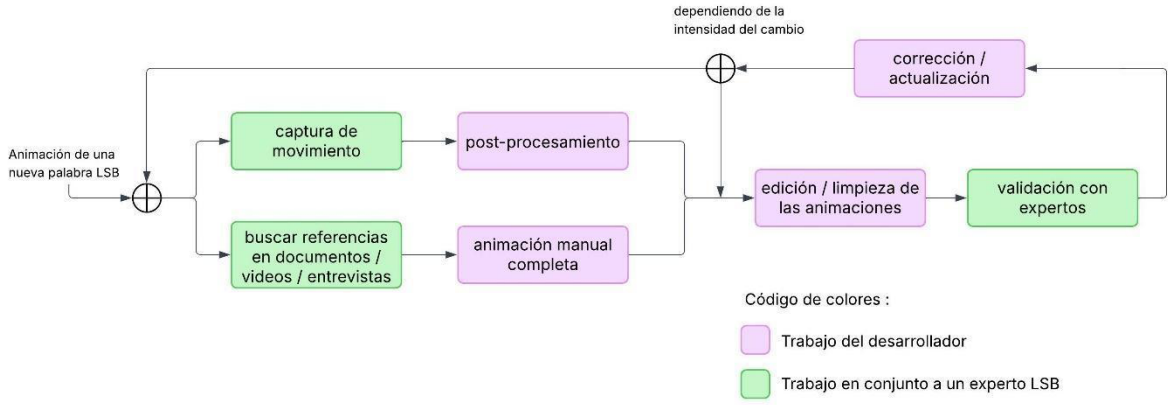

Figura 1: Proceso de animación convencional

Debido a estas dificultades, muchas herramientas tecnológicas para la comunidad Sorda no son escalables y su adaptación a nuevos entornos lingüísticos es limitada. La necesidad de capturar nuevas animaciones, integrarlas manualmente y ajustar la programación del software genera una barrera que restringe su implementación en diferentes países o comunidades.

Frente a esta problemática, una solución innovadora es la formalización de un lenguaje de animación para la representación de las señas, lo que permitiría sistematizar la creación de animaciones sin necesidad de recurrir constantemente a la captura de movimiento. Con un sistema basado en parámetros predefinidos, sería posible generar nuevas animaciones mediante reglas programadas, facilitando la inclusión de diferentes lenguas de señas sin depender de procesos manuales extensos (ver Figura 1). Esta propuesta reduciría costos, agilizaría la expansión de las herramientas existentes y garantizaría una mayor accesibilidad para la comunidad Sorda a nivel global [2], sin tener que replicar el proceso de animación convencional para cada Lengua de Señas, como se observa en la Figura 2.

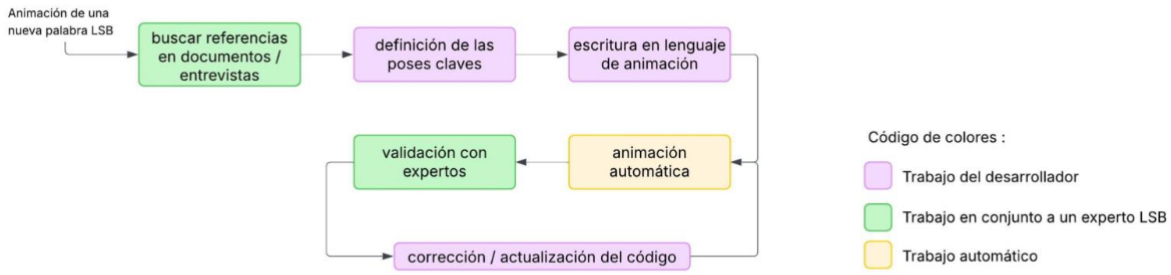

Figura 2: Proceso de animación propuesto

Una vez definidas las poses clave, se realiza la escritura manual de las animaciones con la asistencia de las herramientas desarrolladas para capturar los vectores de posición de los brazos, el plugin de Blender, y el DSL (Domain Specific Language) desarrollado con su respectivo compilador. Ambas herramientas son presentadas formalmente en la sección 2.2 y la sección 3.2. Una vez obtenidas las animaciones basadas en el DSL de manera automática con el compilador (ver Figura 3), se utiliza el archivo “database.json” para la generación de dichas animaciones en un formato 3D apto para el modelo articulado del intérprete utilizado, resultando en el archivo “animations.fbx”.

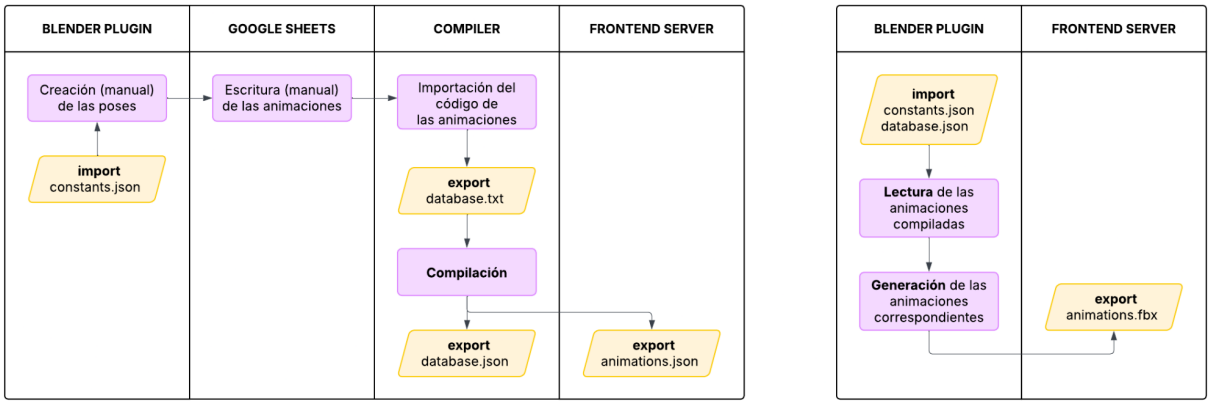

Figura 3: Trabajo "automático" de la generación de las animaciones

1.2 Solución propuesta

Se trabajó en la creación de un intérprete virtual que traduce texto en español a la Lengua de Señas Boliviana (LSB), con el objetivo de mejorar la comprensión lectora de la comunidad Sorda en Bolivia. Este sistema integró múltiples tecnologías para abordar la problemática desde un enfoque integral, combinando procesamiento de lenguaje natural, modelado y animación 3D, y la formalización estructurada de las señas.

De este modo, el prototipo final se construyó a partir de tres componentes fundamentales:

- Un asistente inteligente, diseñado para procesar texto en español y reformularlo en una estructura compatible con la LSB. Este módulo utilizó modelos de lenguaje para adaptar frases y seleccionar términos adecuados dentro del glosario de la LSB.
- Un lenguaje de animación especializado, cuya función fue estructurar las señas en parámetros programables, permitiendo la generación de animaciones sin depender de la captura de movimiento manual. Este lenguaje hizo posible estandarizar la representación de las señas y optimizar la creación de nuevos gestos.
- Un avatar 3D animado, desarrollado para representar las señas de manera visual e intuitiva. La animación del avatar respetó las características gramaticales y expresivas de la LSB, asegurando que la interpretación fuera comprensible y fluida.

Estos componentes se integraron en una aplicación web, accesible desde cualquier dispositivo con conexión a internet, lo que garantizaría su disponibilidad para la comunidad Sorda sin requerir equipamiento especializado. El sistema IVILSB implementado, incluyendo el DSL, se puede acceder en el repositorio de GitHub [7]. En este repositorio se encuentran estructurados el sistema del portal web (front-end) junto a la integración del avatar 3D y el sistema backend que permite conectarse con la API de OpenAI para realizar las adaptaciones de texto a LSB.

Si bien la formalización del lenguaje de animación representó un aspecto innovador, su implementación formó parte de un desarrollo más amplio, centrado en la creación de un intérprete virtual funcional y accesible. Más allá de la animación de las señas, el sistema abordó el desafío de la comprensión lectora desde una perspectiva completa, combinando inteligencia artificial, lingüística computacional y tecnología 3D para ofrecer una solución integral a las necesidades de las personas Sordas en Bolivia.

2. FORMALIZACIÓN DE UNA LENGUA SEÑANTE

Con el objetivo de estructurar y automatizar la representación digital de la Lengua de Señas Boliviana (LSB), se propuso una clasificación exhaustiva de sus componentes gestuales.

2.1 Clasificación de la LSB

Esta clasificación propuesta permite identificar relaciones internas entre las señas y establecer características formalizables computacionalmente. La clasificación parte desde la concepción de que una seña está compuesta de un conjunto de poses clave, donde se suele mostrar el cambio de movimiento durante la seña, como se puede observar en la seña “Buenos días” de la Figura 4. Para la realización de las animaciones, las poses clave pueden ser tantas como sea necesario para preservar el movimiento natural del intérprete, por lo tanto, estas suelen aumentar cuando existe un movimiento curvo pronunciado o que implica un cambio en el gesto de las manos.

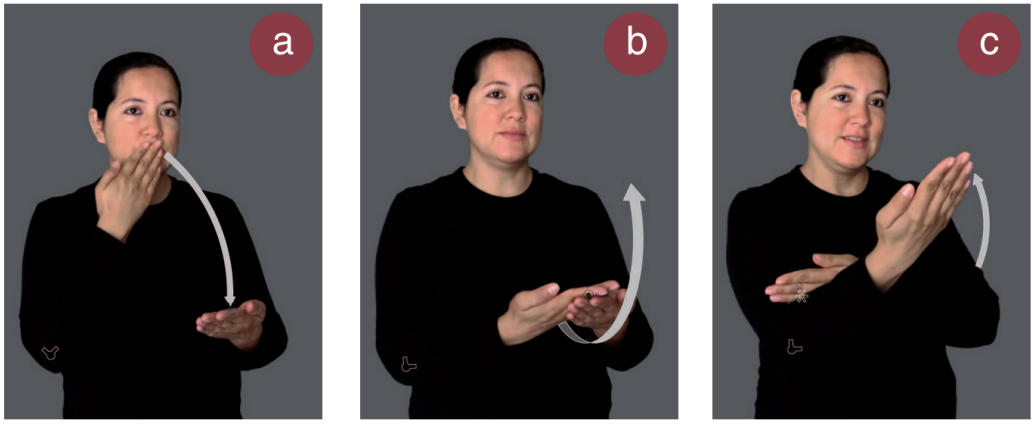

Figura 4: Composición de la seña "Buenos días" por medio de imágenes de las poses clave (Extraído de documentos publicados por el Ministerio de Educación de Bolivia [6]). La secuencia a,b,c muestra los movimientos necesarios para expresar "Buenos días" en LSB.

En base a las poses clave, se categorizaron los movimientos señantes en dos grupos principales:

Variables finitas, correspondientes a configuraciones discretas de la posición de los dedos y expresiones faciales codificables. Estas comprenden un conjunto limitado de combinaciones que pueden ser representadas como símbolos en un lenguaje formal.

Variables infinitas, asociadas al movimiento continuo de los brazos (muñeca, codo, hombro). Estas requieren una representación basada en vectores de posición y orientación en el espacio tridimensional para capturar su naturaleza dinámica.

Esta clasificación dio pie a una formalización lingüística y computacional de la LSB, facilitando la automatización de su interpretación mediante algoritmos y sistemas gráficos como el usado en el prototipo desarrollado, como se puede ver en la Figura 5, siendo la Pose el nodo/elemento básico que compone este lenguaje y a su vez un conjunto o lista de Poses son los que crean una animación.

Estructura de una Pose	Pose : {
	Mano_Der : String,
	Rot_H_Der : [Rx: float, Ry: float, Rz: float],
	Rot_C_Der : [Rx: float, Ry: float, Rz: float],
	Rot_M_Der : [Rx: float, Ry: float, Rz: float],
	Mano_Izq : String,
	Rot_H_Izq : [Rx: float, Ry: float, Rz: float],
	Rot_C_Izq : [Rx: float, Ry: float, Rz: float],
	Rot_M_Izq : [Rx: float, Ry: float, Rz: float]
	Velocidad : int
	}

Estructura de una animación	Animación : Lista<Pose>
Ejemplo de animación	Animacion1 = [Pose1, Pose2, Pose3, Pose4]

Figura 5: Estructura de las instrucciones de animación.

2.2 Lenguaje de animación

Si bien la clasificación previa de los movimientos señantes permitió establecer una estructura precisa para definir animaciones en la LSB, el uso directo de dichos parámetros resultaba poco práctico al generar secuencias complejas de señas. La manipulación manual de cada variable, aunque precisa, hacía el proceso tedioso, propenso a errores y poco escalable.

A causa de ello, se propuso optimizar su flujo, diseñando un lenguaje de animación específico, capaz de abstraer instrucciones complejas en bloques funcionales reutilizables y más comprensibles. Este lenguaje incluyó la incorporación de funciones clave que simplificaron la definición de patrones temporales y espaciales frecuentes en las animaciones señantes. Entre estas, destacan especialmente las funciones REPEAT y SPEED [2].

La función REPEAT permite repetir una o varias instrucciones de animación un número determinado de veces, lo que es útil para representar movimientos cíclicos o énfasis propios de la lengua de señas. Por otra parte, la función SPEED permite ajustar la velocidad de ejecución de un grupo de instrucciones, adaptando el ritmo del gesto a las necesidades expresivas o gramaticales del signo.

Ambas funciones pueden anidar otras instrucciones o funciones, lo que da lugar a estructuras jerárquicas potentes y flexibles. Por ejemplo, es posible definir una secuencia de gestos que se repita tres veces, con una velocidad distinta en cada repetición, mediante el uso combinado de estas funciones. Esta capacidad de anidamiento permite modelar de manera más realista las variaciones expresivas y temporales presentes en la LSB.

La incorporación de estas funciones permitió abstraer la complejidad técnica de los parámetros de bajo nivel, ofreciendo un mecanismo más claro y reutilizable para construir animaciones. Esto no solo mejora la usabilidad del sistema por parte de desarrolladores, sino que también abre la posibilidad de escalar el enfoque a nuevos contextos de traducción y animación señante. Esto se puede observar en el ejemplo de la estructura de instrucciones final (MUJER) que se presenta en la Figura 6, donde los valores de los vectores vectoriales se obtienen del plugin de Blender desarrollado (ver Sección 3.2).

Formalización de seña Mujer — 51 líneas de código

```
{
"MUJER": {
"name": "MUJER",
"poses": [
{
"RH": "R_10",
"R1": "[0.13, 0.73, 0.93]",
"R2": "[-0.03, 0.86, 85.86]",
...
```

Seña Mujer en el lenguaje de animación — ~6 líneas de código

```
(MUJER)
REPEAT(2, {R_10, [0.13, 0.73, 0.93], [-0.03, 0.86, 85.86], [0.21, 0.12,
0.06]} - {L_P3, [0.28, -0.11, 0.90], [0.00, 0.20, 0.00], [0.22, 0.33,
0.00]},
{R_10, [0.15, 0.60, 0.95], [-0.05, 0.81, 2055.85], [0.21, 0.12, 0.06]} -
{L_P3, [0.28, -0.11, 0.90], [0.00, 0.20, 0.00], [0.22, 0.33, 0.00]})
```

(valores vectoriales obtenidos del plugin de Blender, de elaboración propia)

Figura 6: Estructura final de las instrucciones (ahora lenguaje). En el formato original son 51 líneas, pero en el lenguaje se reducen a 6.

3. IMPLEMENTACIÓN

Con la intención de garantizar que las animaciones definidas mediante el lenguaje de animación sean comprensibles por el sistema y ejecutables por el avatar diseñado, se desarrolló un compilador específico para dicho lenguaje.

3.1 Compilador del DSL

Este compilador tiene la función de validar la sintaxis del código escrito en el DSL (Domain-Specific Language) diseñado para representar las señas, y transformarlo en una estructura de datos interpretable por el motor de animaciones.

El compilador verifica que cada instrucción escrita cumpla con las reglas gramaticales del lenguaje, incluyendo la correcta escritura de poses, vectores y funciones. En caso de que se detecten errores, el compilador impide su procesamiento y señala el punto conflictivo, lo cual permite una depuración inmediata del script.

Una vez validado, el compilador transforma el código en un formato estructurado, específicamente un archivo JSON, que contiene la secuencia de animaciones desglosadas y listas para ser interpretadas por el sistema. Las funciones

repetitivas o parametrizadas son descompuestas durante este proceso, permitiendo una ejecución más directa por parte del motor gráfico.

Este enfoque no solo asegura la integridad de los datos antes de ser enviados al sistema de animación, sino que también permite mantener una base de datos consistente, donde cada seña tiene asociada una descripción formal y reproducible de su animación.

3.2 Plugin de Blender

Como parte del sistema desarrollado, se creó un plugin personalizado para Blender con el propósito de facilitar la generación, visualización y prueba de las animaciones señantes del intérprete virtual. Este complemento fue diseñado en Python, adaptándose a la arquitectura interna de Blender v4.1, y se integra directamente en su interfaz gráfica.

El plugin se organiza en cinco secciones principales:

- Configuraciones globales, la cual permite cargar la base de datos de animaciones, las posiciones clave de las manos, las escalas de rotación para cada articulación, y otros parámetros generales necesarios para el funcionamiento del intérprete. Desde esta sección también es posible cargar una pose completa a partir de una instrucción escrita o copiar la pose actual adoptada por el avatar.
- Control de brazos (izquierdo y derecho), que ofrece al usuario la posibilidad de manipular directamente los ángulos de rotación de hombro, codo y muñeca, ya sea en grados reales o mediante una escala normalizada definida previamente. Las posiciones pueden sincronizarse, restaurarse o modificarse de forma individual para cada brazo.
- Configuración de manos, la cual permite seleccionar directamente la pose de cada mano a partir de un listado de posiciones codificadas y previamente cargadas. Estas poses incluyen tanto las extraídas del alfabeto dactilológico como las identificadas en el glosario propio de la LSB, las cuales fueron previamente creadas en Blender.
- Controlador de la animación, permite importar y ejecutar una animación completa compuesta por varias poses clave. El usuario puede especificar la cantidad de fotogramas por transición, verificar la fluidez del movimiento y realizar ajustes en tiempo real. Esta sección se usa también para validar visualmente los scripts generados y exportar las animaciones en formato FBX.
- Logs y seguimiento, donde el plugin incluye un sistema de registro en consola que informa sobre las operaciones realizadas, errores de carga o ejecución y cualquier conflicto en la interpretación de instrucciones.

Este plugin fue esencial durante el proceso de desarrollo de la herramienta, ya que permitió un entorno de prueba y depuración directa de las animaciones antes de su integración en la aplicación web. Además, al estar conectado con la base de datos generada por el compilador, el plugin facilitó la edición rápida y escalable del repertorio de señas sin necesidad de recurrir a procesos manuales o a la captura de movimiento (ver Figura 7).

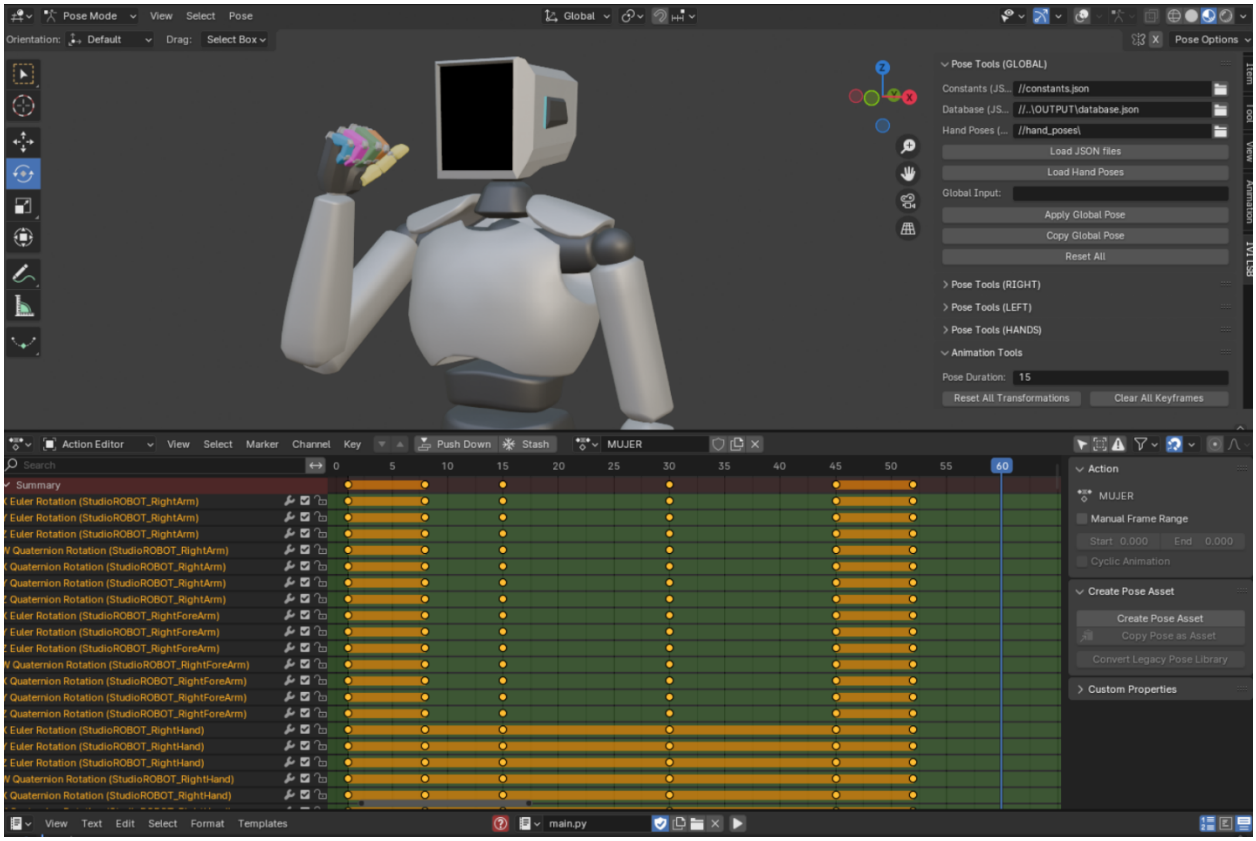

Figura 7: Creación de una animación con captura de señas.

3.3 Integración en el prototipo

Una vez desarrollados los distintos componentes mencionados en las secciones previas, se procedió a su integración en un prototipo funcional accesible desde una aplicación web. Este prototipo fue diseñado para que cualquier persona pudiera ingresar texto en español y obtener, como resultado, su interpretación en LSB a través de un avatar 3D animado.

Para ello, se construyó una interfaz gráfica simple e intuitiva, en la que el usuario puede introducir una oración en español, la cual es procesada por una API basada en modelos de lenguaje de OpenAI, del cual se optó por trabajar con el modelo GPT-4o luego de evaluar opciones similares disponibles a la fecha de realización de la herramienta, debido a su capacidad de formular respuestas acordes a los requisitos de interpretación con pocas iteraciones de corrección. Cabe destacar que se consideró de primera instancia el uso de modelos pre-entrenados comerciales por su capacidad de manejo del lenguaje humano y la ausencia de una base de datos que permita la construcción de un LLM (Large Language Model) especializado. Esta API tiene la función de adaptar el texto ingresado a una estructura compatible con la gramática de la LSB. Dado que esta lengua posee un orden sintáctico y morfosintáctico diferente al español, por lo que el texto original debe ser reformulado para preservar su significado al ser traducido a señas.

El texto adaptado se transforma posteriormente en un script en el lenguaje de animación previamente definido. Este script es validado por el compilador, que genera un archivo estructurado con las instrucciones necesarias para el motor de animación. El motor interpreta ese archivo y transmite las instrucciones al avatar animado, que reproduce la secuencia correspondiente mediante una serie de poses clave sincronizadas.

La aplicación fue construida en ReactJS, mientras que el backend, encargado del procesamiento lingüístico, la compilación y la gestión de archivos de animación, fue desarrollado en Python 3.11.3. La comunicación entre la interfaz, el sistema de interpretación y el avatar se realiza de forma transparente para el usuario final.

Esta integración permite conectar todas las partes del sistema en un flujo coherente y automatizado: desde la entrada del texto, pasando por la adaptación gramatical, la generación del script de señas, la validación estructural, y finalmente la visualización en tiempo real de la interpretación en la LSB. La interfaz gráfica final se puede observar en la Figura 8.

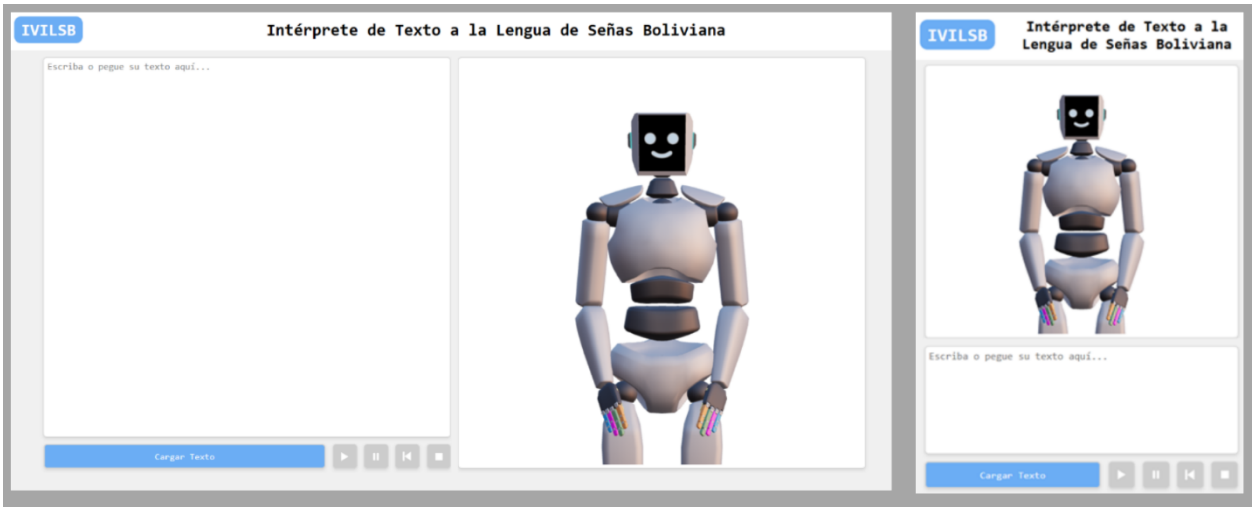

Figura 8: Aplicativo IVILSB con su interfaz de usuario gráfica.

4. RESULTADOS

El sistema desarrollado logró formalizar exitosamente los gestos de la LSB dentro de un marco de animación programable y reutilizable. A través de la combinación del lenguaje de animación, el compilador y el motor de ejecución con un avatar 3D, fue posible representar señas completas de forma fluida, precisa y sin depender de herramientas como la captura de movimiento o la constante participación de intérpretes expertos en la LSB.

Estos expertos estuvieron presentes solamente en las entrevistas cualitativas realizadas para la investigación inicial de la problemática y, posteriormente, para validar el correcto uso de la LSB dentro del aplicativo. Como experto se incluyó a intérpretes certificados y usuarios nativos, a quienes se les presentó una serie de demostraciones del intérprete virtual en funcionamiento. Específicamente se tomaron en cuenta a 7 personas para su validación, la cual consistió en una reunión con la directora del Instituto Fe y Alegría de la ciudad de Sucre, 2 profesores, 3 estudiantes para docente del mismo instituto y una reunión privada por videoconferencia con la psicóloga de la Unidad Educativa Lucy Argandoña de la ciudad de Cochabamba. En términos generales, las animaciones fueron aprobadas como representaciones válidas y comprensibles de las señas mostradas. Se destacó la fidelidad de las poses manuales, la claridad en las transiciones y la correcta sincronización entre gestos y expresiones faciales.

Además, se observó que el sistema es flexible y escalable, permitiendo la incorporación de nuevas señas o modificaciones gramaticales sin necesidad de rehacer las animaciones, donde los expertos estuvieron de acuerdo en que sería posible utilizar la herramienta en otras Lenguas de Señas similares, tomando en cuenta sus respectivos cambios gramaticales y expertos aptos para su validación. Esta capacidad de adaptación fue valorada como una ventaja significativa en comparación con soluciones convencionales basadas en videos o capturas de movimiento estáticas.

Los resultados obtenidos confirmaron la viabilidad técnica y lingüística del enfoque propuesto, abriendo la puerta a futuras mejoras y ampliaciones del sistema en contextos educativos, institucionales o de acceso a información pública.

5. DISCUSIÓN

Si bien el lenguaje de animación desarrollado permitió estructurar y automatizar una gran parte de los gestos de la LSB, aún existen limitaciones inherentes a su naturaleza programática y a la complejidad del lenguaje señado. Una de las principales restricciones está relacionada con la expresividad no manual, especialmente aquellas sutilezas del rostro, mirada y postura corporal que pueden variar según el contexto comunicativo o emocional, y que no siempre pueden codificarse mediante instrucciones discretas.

Otro límite importante es que el lenguaje, tal como fue concebido, se orienta a la representación gramatical y mecánica de la seña, pero no incorpora mecanismos de interpretación semántica profunda. Es decir, la transformación del texto en estructuras gramaticales compatibles con la LSB aún depende de un modelo externo (como las APIs de OpenAI), lo que introduce una capa de dependencia y posibles imprecisiones si no se ajusta adecuadamente el prompt o los parámetros de control.

Además, si bien el lenguaje admite funciones como REPEAT y SPEED, que aportan flexibilidad y simplificación, la curva de aprendizaje para su uso sigue siendo moderadamente técnica. Usuarios sin conocimientos en programación o lógica formal podrían encontrar dificultades para definir nuevas señas complejas sin una interfaz que los asista.

Se recomienda, en trabajos futuros, ampliar el vocabulario estructural del lenguaje con nuevas funciones de control expresivo (por ejemplo, niveles de tensión muscular o direcciones de mirada), así como desarrollar interfaces gráficas que generen automáticamente scripts del lenguaje a partir de opciones visuales, lo que facilitaría su adopción por parte de usuarios no técnicos. También sería valioso explorar la posibilidad de integrar modelos lingüísticos más específicos para la LSB que automaticen con mayor precisión la interpretación gramatical del texto fuente antes de su conversión a señas.

6. CONCLUSIÓN

Se logró demostrar la viabilidad de formalizar la LSB dentro de un marco técnico que permite su animación automática y precisa, sin necesidad del uso de captura de movimiento. La propuesta combinó el desarrollo de un lenguaje de animación especializado, un compilador capaz de validar y traducir estas instrucciones, un plugin para la manipulación directa en Blender, y una aplicación web con un avatar 3D como interfaz visual del intérprete virtual.

A nivel funcional, el sistema fue capaz de traducir texto en español a secuencias animadas en LSB, manteniendo la coherencia gramatical de la lengua de señas y generando resultados comprensibles para usuarios expertos. Las entrevistas de validación confirmaron la fidelidad de las animaciones producidas, y se destacó la utilidad del enfoque formal para simplificar y escalar la producción de señas.

Más allá del desarrollo técnico, este trabajo evidenció el potencial de las herramientas computacionales para atender necesidades reales de accesibilidad en la comunidad Sorda. Al proponer un sistema modular, reutilizable y adaptable, se abre un camino hacia soluciones más inclusivas y sostenibles que pueden extenderse a otras lenguas de señas o aplicaciones educativas. De esta forma, la herramienta presentada busca posicionarse como una de las bases para futuras investigaciones centradas en la integración de tecnología, lenguaje y accesibilidad social.

REFERENCIAS

[1] G. R. R. B. de Bolivia, Guía para la atención a personas con sordera, hipoacusia y sordoceguera en el sistema educativo plurinacional de Bolivia, La Paz, Bolivia: Ministerio de Educación, 2023. Disponible en: https://www.minedu.gob.bo/files/publicaciones/veaye/dgee/GUIA-PERSONAS-SORDAS-cT.pdf

[2] M. Oropeza Condori, “Prototipo de Intérprete Virtual de Texto a Lengua de Señas Boliviana como Herramienta de Comprensión Lectora”, Proyecto de Grado, Universidad Privada Boliviana, Cochabamba, Bolivia, 2025.

[3] V. Setiawan et al., “An Interactive Sign Language Based Mobile Application for Deaf People”, en *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2023, pp. 1635–1641, doi: 10.1109/ICOEI56765.2023.10125821.

[4] G. G. Nath y V. S. Anu, “Embedded sign language interpreter system for deaf and dumb people”, en *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, IEEE, 2017, pp. 1–5, doi: 10.1109/ICIIECS.2017.8275907.

[5] N. Tiangtae, S. Ramingwong, L. Ramingwong, D. Potikanond, N. Homkong, y N. Maneerat, “Developing software for the deaf community: Conquering an extreme case scenario”, en *2017 21st International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2017, pp. 1–5, doi: 10.1109/ICSEC.2017.8443794.

[6] Ministerio de Educación de Bolivia, Federación Boliviana de Sordos, Fundación Amazónica para el Desarrollo de los Sordos, y Proyecto “arca” de Riberalta, “Curso de enseñanza de la Lengua de Señas Boliviana LSB”. Ministerio de Educación, Diciembre de 2012. Consultado: el 12 de junio de 2025. [En línea]. Disponible en: https://www.minedu.gob.bo/files/publicaciones/veaye/dgee/CURSO-DE-ENSENANZA-DE-LA-LENGUA-DE-SENAS-BOLIVIANA-Modulo-1.pdf

[7] M. Oropeza Condori, “IVILSB”, repositorio de Github, Github. Consultado: el 25 de julio de 2025 [En línea]. Disponible en: https://github.com/MayOnesita/IVILSB.git