

# Aprendizaje del Alfabeto en Lengua de Señas con un Avatar 3D Interactivo y Retroalimentación de Gestos en Tiempo Real

1<sup>st</sup> Juan Camilo D'Aleman Rodriguez  
*dept. de sistemas e industrial*  
*Universidad Nacional de Colombia*  
Bogotá, Colombia  
jdalemanr@unal.edu.co

2<sup>nd</sup> Juan Esteban Medina Cardenas  
*dept. de sistemas e industrial*  
*Universidad Nacional de Colombia*  
Bogotá, Colombia  
jumedinaca@unal.edu.co

3<sup>rd</sup> Juan Esteban Muñoz Muñoz  
*dept. de sistemas e industrial*  
*Universidad Nacional de Colombia*  
Bogotá, Colombia  
juamunozmu@unal.edu.co

4<sup>th</sup> Manuel Felipe Espinosa Español  
*dept. de sistemas e industrial*  
*Universidad Nacional de Colombia*  
Bogotá, Colombia  
mespinosae@unal.edu.co

**Abstract**—Este artículo presenta el diseño e implementación de una aplicación educativa MVP que enseña el alfabeto en lengua de señas utilizando un avatar 3D y evaluación de gestos en tiempo real. El sistema combina Unity para contenido interactivo, Blender para modelado y animación de avatares, y MediaPipe Hands para el seguimiento de manos basado en puntos de referencia. Un modelo de coincidencia ligero evalúa los gestos del usuario y proporciona retroalimentación inmediata (Correcto, Casi, Incorrecto). El MVP se centra en las vocales (A, E, I, O, U) y consonantes, admite la persistencia local del progreso del usuario y funciona sin conexión para preservar la privacidad. Los resultados preliminares indican una interacción receptiva (objetivo de retroalimentación menor a 150 ms) y un rendimiento adecuado en PC, con una hoja de ruta para la implementación en Android y futuros módulos.

**Index Terms**—Lengua de señas, visión por computadora, MediaPipe, Unity, Blender, interacción humano-computadora, accesibilidad, tecnología educativa.

## I. INTRODUCCIÓN

El proyecto tiene como objetivo reducir las barreras para el aprendizaje de la lengua de señas proporcionando una herramienta accesible, atractiva y moderna que beneficie tanto a los usuarios oyentes que comienzan su viaje como a los usuarios sordos en contextos de alfabetización. El aprendizaje tradicional sufre de materiales interactivos limitados y escasas oportunidades de retroalimentación inmediata. Nuestro enfoque aprovecha las animaciones 3D interactivas y la visión por computadora en tiempo real para apoyar la práctica y la retención.

Esta primera versión (MVP) se dirige al alfabeto, comenzando con las vocales para el control del alcance y la viabilidad. Los usuarios ven a un avatar realizar la seña, la imitan frente a la cámara y reciben retroalimentación instantánea. El sistema almacena el progreso local y está diseñado para escalar hacia consonantes y saludos básicos.

## II. TRABAJO RELACIONADO Y CONTEXTO

Las herramientas educativas anteriores a menudo dependen de videos pregrabados o ilustraciones estáticas, lo que limita la interactividad y la personalización. La estimación de la pose de la mano ha madurado con soluciones como MediaPipe Hands, que permite una extracción robusta de puntos de referencia en tiempo real (21 puntos clave por mano). Los clasificadores de ML se entrenaron para el reconocimiento completo de gestos.

## III. DESCRIPCIÓN GENERAL DEL SISTEMA

### A. Objetivos

- Entregar un MVP funcional para el aprendizaje del alfabeto en lengua de señas (vocales).
- Combinar la instrucción de avatar 3D con detección de gestos en tiempo real y evaluación.
- Proporcionar retroalimentación inmediata y rastrear el progreso del usuario localmente.
- Operar sin conexión para respetar la privacidad.

### B. Alcance

Incluye un inicio de sesión/registro básico, un módulo completo de vocales, interfaz de usuario principal (Menú, Práctica, Lecciones, Hoja de ruta) y persistencia local. Excluye backends en la nube, gamificación avanzada y soporte multilingüe en el MVP.

### C. Arquitectura

- Presentación: Escenas de Unity para Menú, Hoja de ruta, Práctica, Lecciones.
- Dominio/Lógica: Controladores de letras, máquina de estados de evaluación de gestos.
- Infraestructura: Persistencia basada en JSON (UserData, ProgressData, umbrales).

- Integración de CV: Envoltorio de MediaPipe Hands para transmitir puntos de referencia.
- Animación: Animaciones FBX creadas en Blender importadas a Unity, controladas a través de Animator.

#### IV. MÉTODOS

##### A. Modelado y Animación de Avatares

Se importó un avatar humano riggeado de Mixamo a Blender. Las poses de las manos para las vocales se crearon como animaciones de fotogramas clave (por ejemplo, Sign\_A, Sign\_E). Las animaciones se exportaron como FBX con la configuración adecuada de la armadura (sin huesos de hoja, animación horneada). En Unity, los clips se configuraron bajo un rig humanoide, gestionado a través de Animator Controller, lo que permite la reproducción, el control de bucle y la retención de pose.

##### B. Pipeline de Visión por Computadora

Se integró MediaPipe Hands para capturar puntos de referencia de la mano en tiempo real. Los puntos de referencia se normalizaron en relación con la orientación y escala de la muñeca para proporcionar comparaciones estables independientes de la posición de la cámara. El sistema no almacena imágenes/video, solo vectores numéricos normalizados, preservando la privacidad.

##### C. Clasificación mediante Support Vector Machines (SVM)

A partir de los puntos de referencia normalizados de la mano, cada señal se representó como un vector de características de 63 dimensiones, correspondiente a las coordenadas tridimensionales de los 21 landmarks detectados. Estos vectores describen la geometría relativa de la mano y constituyen la entrada del modelo de clasificación.

Para el reconocimiento de las señales estáticas se empleó un clasificador Support Vector Machine (SVM) con kernel de función de base radial (RBF). Este modelo es adecuado para datos continuos y de baja dimensionalidad, y permite aprender fronteras de decisión no lineales que separan eficazmente las distintas configuraciones geométricas de la mano.

Dado que el problema es multiclase, el SVM se extendió mediante una estrategia one-vs-one, en la cual se entrenan clasificadores binarios para cada par de clases. La decisión final se obtiene a partir de un esquema de votación entre dichos clasificadores. Cada clasificador aprende un hiperplano en el espacio de características, permitiendo una discriminación robusta entre señales con variaciones naturales en la postura de la mano.

El uso de SVM proporciona una solución computacionalmente eficiente y con buena capacidad de generalización, adecuada para la inferencia en tiempo real dentro de la aplicación desarrollada en Unity.

##### D. Evaluación

El MVP utiliza un modelo entrenado para comparar y decidir si el gesto es correcto o incorrecto.

##### E. Retroalimentación y UX

La aplicación proporciona indicadores visuales directos. Las lecciones utilizan la parte superior del cuerpo completa del avatar; la práctica utiliza visuales de manos enfocados. La interfaz de usuario busca un alto contraste, visualización de cámara reflejada para una interacción intuitiva y tiempos claros (marcos de estabilidad ~5).

#### V. DETALLES DE IMPLEMENTACIÓN

##### A. Pila Tecnológica

- Motor: Unity (se recomienda 2022 LTS), C#.
- CV: MediaPipe Hands a través de la integración de Unity.
- 3D: Blender (rigging, animación), exportación FBX a Unity.
- UI: Unity UI (Canvas) y TextMeshPro.
- Datos: JSON para persistencia.

##### B. Organización del Proyecto

Assets/

- Scenes
- Scripts
- Animations
- Models
- UI
- Config

##### C. Prácticas de Exportación e Importación

Exportación FBX desde Blender con Objetos Seleccionados (Malla + Armadura), Aplicar Escalas: FBX Todo, Adelante - Z, Arriba Y, Agregar Huesos de Hoja deshabilitado, Hornear Animación habilitado. En Unity, establecer Rig en Humanoide, extraer materiales/texturas, configurar opciones de clip de animación (Bucle desactivado para poses fijas).

#### VI. RESULTADOS

##### A. Resultados Funcionales

- Lecciones de vocales implementadas con animaciones de avatar.
- Seguimiento de manos en tiempo real y evaluación operativa en PC.
- Retroalimentación inmediata con pistas básicas.
- Persistencia local de finalización por letra.
- Navegación de interfaz de usuario a través de Menú, Lecciones, Práctica.

##### B. Indicadores de Rendimiento

- Latencia objetivo para retroalimentación:  $< 150$  ms (dependiendo del hardware).
- Velocidad de fotogramas objetivo:  $\geq 30$  FPS en Android de gama media; verificado estable en PC.

##### C. Observaciones de Usabilidad

Los usuarios se benefician de la vista de cámara reflejada y la comparación clara lado a lado del avatar frente a la mano del usuario.

## VII. DISCUSIÓN

### A. Riesgos y Mitigaciones

- Riesgo de integración de MediaPipe Android: prototipar temprano, recurrir al modo de demostración de PC si es necesario.
- Riesgo de cobertura de animación: priorizar la fidelidad de la mano, estandarizar nombres, permitir marcadores de posición.
- Riesgo de rendimiento: perfilar y optimizar la resolución de la cámara, reducir las asignaciones por fotograma.

### B. Privacidad

No se persisten imágenes ni videos. Solo se almacenan características numéricas normalizadas para el progreso y la evaluación, manteniendo los datos del usuario locales y mínimos.

## VIII. CONCLUSIÓN Y TRABAJO FUTURO

Entregamos una arquitectura e implementación MVP para el aprendizaje del alfabeto en lengua de señas, combinando un avatar 3D en Unity, animaciones de Blender y seguimiento de manos basado en MediaPipe. El sistema proporciona retroalimentación inmediata y seguimiento del progreso local, operando sin conexión para apoyar la accesibilidad.

El trabajo futuro incluye expandir el contenido a saludos básicos, mejorar las pistas con retroalimentación adaptativa, agregar clasificación ML opcional para gestos ambiguos, mejorar las características de accesibilidad y completar la optimización y empaquetado de Android.

## AGRADECIMIENTOS

Agradecemos a los colaboradores del proyecto en todos los roles, así mismo a la profesora Aura Maria Forero Pachon y el Ing Arbey Aragon Bohorquez por brindarnos el conocimiento necesario en los contenidos de las clases. También reconocemos los recursos de la comunidad (Mixamo, MediaPipe) que permitieron la creación rápida de prototipos.

## REFERENCES

- [1] MediaPipe Hands, Google Research.
- [2] Unity Technologies, Unity Engine Documentation.
- [3] Blender Foundation, Blender Manual.
- [4] Adobe Mixamo, Rigged Character Library.
- [5] TextMeshPro, Unity Asset integration.
- [6] Unity Test Framework Documentation.