



Benemérita Universidad Autónoma de Puebla

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Administración de Proyectos

Otoño 2025

Plataforma de Aprendizaje de LSM para Niñez Intermedia

FECHA DE ENTREGA: 00/11/2025

Profesor:

Ángel Omar Rojas Mendoza

Alumno(s):

Carmona Velasco Andry Sebastian

Pestaña Márquez Valeria

Romero Lagunes Josette Yeraldin

Ruiz Santiago María Fernanda

Visca Cocotzin Blanca Flor



Índice

1. Introducción	3
2. Justificación	3
3. Descripción General del Producto	5
4. Objetivos	5
4.1. Objetivos Específicos:	5
5. Equipo de Desarrollo	5
6. Plan de Desarrollo	8
6.1. Objetivo Específico 1: Diseño de la Interfaz Gráfica del Usuario	8
6.2. Objetivo Específico 2: Entrenamiento de la IA para el Reconocimiento de Imágenes. .	14
6.3. Objetivo Específico 3: Ejecución del Periodo de Prueba.	16
7. Análisis de Riesgos	18
7.1. Problemas de interacción entre el usuario y la plataforma.	18
7.2. Exceso en la petición o solicitud del análisis de la imagen.	18
7.3. Conectividad con la red.	19
8. Conclusión	21



1. Introducción

Actualmente es indispensable que personas con diversidad comunicativa puedan expresarse y comprender a otros sin impedimentos, por tal razón, es de suma importancia desarrollar herramientas que permitan disminuir la brecha de comunicación con la sociedad en general. La Lengua de Señas es un lenguaje con gramática que permite la comunicación a través de movimientos con la mano, en México se reconoce la Lengua de Señas Mexicana (LSM) como una lengua oficial nacional, desarrollada para establecer la comunicación entre individuos con alguna discapacidad auditiva, mediante un intercambio viso-gestual.

Este proyecto plantea la creación de un aplicación de aprendizaje enfocada a la niñez intermedia que es una etapa clave para motivar el aprendizaje, por lo que los niños tienen gran capacidad de generar nuevas conexiones cerebrales. Gracias a ello, es más sencillo introducir y practicar las letras básicas de la LSM, mediante el uso de Inteligencia (IA) que permita el reconocimiento de señas realizadas para puntuar el progreso con el análisis de imágenes, aportando así una herramienta más de aprendizaje.

Los motivos para el desarrollo de este proyecto se rigen con fines educativos, conforme la necesidad planteada en el objetivo de desarrollo sostenible número 4 “Educación de calidad” mismo que busca garantizar una educación inclusiva, promoviendo oportunidades de aprendizaje sin importar sus diferencias, pretende que las personas tengan igualdad de oportunidades fomentando el respeto, la empatía y la colaboración en un entorno de aprendizaje variado. Su importancia radica en el desarrollo integral de los alumnos y la preparación para un mundo diverso.

2. Justificación

El desarrollo de herramientas que faciliten la comunicación y promuevan la inclusión social es una necesidad crítica en la sociedad contemporánea. En México, la Lengua de Señas Mexicana (LSM) es reconocida como una lengua oficial, sin embargo, existe una marcada brecha de comunicación entre



la comunidad sorda y la población oyente, lo que limita la plena participación de las personas con discapacidad auditiva en la vida social y educativa.

El presente proyecto se justifica bajo tres pilares fundamentales:

- 1. Inclusión y Pertinencia Social:** La aplicación responde directamente a la necesidad de fomentar la accesibilidad y la equidad en el aprendizaje. Al enfocarse en la niñez intermedia (6 a 12 años), se actúa en una etapa crítica donde el cerebro tiene una alta capacidad para generar nuevas conexiones, haciendo del aprendizaje de un segundo lenguaje (como la LSM) una tarea más natural y efectiva. Este enfoque no solo dota a los niños oyentes de una habilidad comunicativa crucial, sino que promueve desde temprana edad el respeto, la empatía y la colaboración hacia la comunidad sorda, alineándose perfectamente con el Objetivo de Desarrollo Sostenible (ODS) número 4: Educación de Calidad.
- 2. Innovación Tecnológica y Metodológica:** La solución propuesta utiliza la Inteligencia Artificial con el motor MediaPipe/Landmark y TensorFlow.js para el reconocimiento de señas en tiempo real. Esta aproximación tecnológica convierte el aprendizaje de la LSM, que tradicionalmente requiere la presencia de un instructor, en una experiencia interactiva, autodirigida y con retroalimentación inmediata. El uso de la IA no es un fin en sí mismo, sino un medio para ofrecer una herramienta pedagógica de alta precisión y escalabilidad, capaz de analizar la ejecución viso-gestual del niño.
- 3. Soporte al Proceso de Aprendizaje:** El diseño de la plataforma como un entorno gamificado y visualmente atractivo (como se detalla en la sección de Diseño de la Interfaz) garantiza la motivación y la retención del público objetivo. Esto resuelve el desafío de mantener el interés de los niños en un tema complejo, transformándolo en un juego que facilita la introducción y práctica del alfabeto de la LSM.

En resumen, la creación de esta aplicación se justifica como una solución tecnológica innovadora y socialmente responsable que busca democratizar el acceso al aprendizaje de la LSM, contribuyendo



de manera significativa a la construcción de una sociedad más inclusiva y con una educación de mayor calidad para todos.

3. Descripción General del Producto

Descripción Funcional Una plataforma web interactiva donde los niños podrán aprender el alfabeto de la Lengua de Señas Mexicana (LSM). La aplicación mostrará una letra y una imagen correspondiente a la seña, y el niño podrá practicar haciéndola frente a la cámara web. La IA analizará la imagen en tiempo real, proporcionando retroalimentación inmediata sobre si la seña fue ejecutada correctamente.

Público Objetivo: Niñez intermedia que se encuentra entre los 6 a 12 años y sus educadores.

4. Objetivos

Objetivo General: Desarrollo de un app web para el reconocimiento de señas mediante el procesamiento de imágenes con el uso de la Inteligencia Artificial.

4.1. Objetivos Específicos:

1. Diseño de la interfaz gráfica del usuario.
2. Entrenamiento de la IA para el reconocimiento de imágenes.
3. Ejecución del periodo de pruebas durante 3 semanas

5. Equipo de Desarrollo

- **Project Management:** Responsable de la planificación, ejecución y cierre exitoso del proyecto, asegurando que se cumpla el alcance, presupuesto y cronograma establecido.



Habilidades clave: liderazgo, organización, comunicación, pensamiento crítico y adaptabilidad.

- **Líder Técnico:** Supervisar aspectos técnicos, tomar decisiones arquitectónicas, hacer mentoría a los miembros del equipo, gestionar y comunicar el proceso a las partes interesadas y alinear los objetivos técnicos con la estrategia de la empresa.

Habilidades clave: experiencia técnica profunda, comunicación, capacidad de mentoría, liderazgo y empatía.

- **Tester:** Encargado de planificar y ejecutar las pruebas para identificar y reportar errores, verificar el cumplimiento de requisitos, asegurar la calidad del producto y colaborar con el equipo de desarrollo.

Habilidades clave: técnicas de prueba, conocimiento de lenguajes y sistemas, pensamiento analítico y lógico, resolución de problemas, comunicación y trabajo en equipo.

- **Desarrollador de Software:** Su labor abarca desde el análisis de las necesidades del usuario y el diseño de soluciones, hasta la escritura del código en diferentes lenguajes de programación.

Habilidades clave: resolución de problemas, pensamiento crítico y analítico, trabajo en equipo, adaptabilidad, atención al detalle.

- **Control de Calidad de la IA:** Automatiza y optimiza la detección de defectos en productos y servicios, mejorando la precisión y eficiencia mediante el uso de algoritmos de aprendizaje automático y visión artificial.

Habilidades clave: Comunicación y colaboración, conocimiento en lenguaje de programación, pensamiento crítico, resolución de problemas y arquitectura de redes neuronales.

- **Diseñador UX:** Se encarga de hacer que un producto o servicio sea útil, agradable y accesible para los usuarios, centrándose en la experiencia general de información, más allá de lo puramente visual.

Habilidades clave: comunicación, metodologías ágiles, pensamiento analítico, investigación de usuario y diseño de interacción.



- **Diseñador UI:** Se encarga de diseñar los elementos visuales e interactivos de un producto digital para que la interacción del usuario sea estéticamente agradable, intuitiva y funcional.

Habilidades clave: comunicación, colaboración, creatividad, proactividad y principios de diseño virtual.

- **Diseñador de Software:** Diseñar la estructura fundamental del sistema de software para garantizar que sea escalable, robusto, seguro y el cumplimiento de los objetivos. Esto implica definir los estándares y tecnología a utilizar.

Habilidades clave: análisis de requisitos, diseño de soluciones, pruebas y validación, documentación.

- **Entrenador de IA:** Supervisión del entrenamiento de redes neuronales, trabajo con grandes volúmenes de datos supervisando el proceso de entrenamiento y ajustando los parámetros para que la IA pueda aprender de los datos de manera más eficaz.

Habilidades clave: entrenamiento de modelos, evaluación y ajuste del modelo y anotación de datos.

- **Ingeniero de Software:** Encargado del diseño, desarrollo, implementación y mantenimiento de sistemas informativos y programas de software.

Habilidades clave: gestión de proyectos, pensamiento crítico, adaptabilidad, trabajo en equipo y compresión del negocio.

- **Auxiliar de Desarrollador:**

- **Auxiliar de Entrenador:**

Para ver la jerarquía completa en mejor resolución, véase *Anexo 1: Organigrama*

6. Plan de Desarrollo

6.1. Objetivo Específico 1: Diseño de la Interfaz Gráfica del Usuario

Etapa 1 — Diseño y Maquetado Visual Se realizó el ticket "*Diseño de la Interfaz*" (CV-001-RSMF), donde se llevó a cabo el maquetado de la interfaz, haciendo una selección de colores adecuada para los infantes. Así como el acomodo estratégico de los botones que tendrá nuestra plataforma. Podrás ver los mockups en el apartado

1. Inicio de la plataforma

La elección de la paleta de colores y los elementos de diseño se realizó considerando principios de psicología del color y las necesidades específicas del público objetivo: niños y niñas en la etapa de niñez intermedia (6 a 12 años). Véase en la fig. 1

- **Azul:** Transmite confianza, calma y seguridad. Es ideal para crear un entorno de aprendizaje sereno pero confiable, lo que reduce la ansiedad inicial al enfrentarse a un nuevo lenguaje.
- **Amarillo:** Asociado con la energía, la alegría y la estimulación mental. Atrae la atención de manera natural y promueve el optimismo y la creatividad, clave para mantener el interés de los niños.
- **Rosa:** Evoca calidez, amabilidad y diversión. Psicológicamente, se relaciona con la empatía y el cuidado, reforzando un ambiente acogedor e inclusivo. Es un color que suele resonar muy bien con los niños, transmitiendo cercanía y entusiasmo.
- **Lila:** Simboliza imaginación, tranquilidad y sensibilidad. Ayuda a fomentar la creatividad y la concentración, además de aportar un toque lúdico y mágico que enriquece la experiencia visual.
- **Contraste y armonía:** La combinación de colores cálidos (amarillo, rosa) con tonos fríos

(azul, lila) crea un balance perfecto: estimula sin saturar, y guía la mirada de manera intuitiva hacia los elementos clave.

- **Enfoque inclusivo y amigable:** Las ilustraciones diversas y los colores vibrantes reforzan un mensaje de inclusión y alegría, esencial para que los niños se identifiquen y se motiven a aprender la lengua de señas mexicanas en un entorno que se siente como un juego.
- **Botón prominente ("Comenzar ahora"):** El uso del lila y el rosa para el botón principal no solo contrasta con el fondo blanco, sino que aprovecha su asociación con la acción y el entusiasmo.



Figura 1: Inicio de la plataforma

2. Sección de temas de la plataforma

El diseño de esta pantalla prioriza la claridad, la simplicidad y la accesibilidad cognitiva, fundamentales para el público infantil. Cada elección de diseño está orientada a reducir la carga cognitiva y guiar al niño de manera intuitiva hacia el contenido de aprendizaje. Cabe mencionar que aquí se llevó a cabo una actualización de los temas y que el cambio se aplicó directamente en el ticket de "*Desarrollo de la Interfaz*"(RS-002-RJLY). Observa la fig. 2 de esta sección.

- **Distribución y Layout:** La disposición de los tres botones de navegación principal ("CONSONANTES", "VOCALES", "NOMBRE") en una columna vertical central es altamente efectiva. Elimina la confusión y presenta las opciones de forma clara y ordenada, permitiendo al niño enfocarse en una sola decisión a la vez. El espacio generoso entre ellos evita errores táctiles.
- **Tipografía y Iconografía:** El uso de una tipografía **sans-serif redondeada** y de peso bold es amigable y fácil de leer para los niños. Los íconos son simples y reconocibles, actuando como apoyos visuales para predecir el contenido de cada sección, lo que es crucial para niños que aún están desarrollando habilidades de lectura.
- **Consistencia y Familiaridad:** La barra de navegación superior con el logo de la aplicación y el ícono de la flecha mantienen la coherencia con la pantalla de inicio. Esto proporciona una sensación de familiaridad y seguridad, permitiendo a los niños orientarse fácilmente dentro de la aplicación.



Figura 2: Sección de temas de la plataforma

3. Vista general de las letras que puedes seleccionar

La estructuración del abecedario en vocales y consonantes refleja una comprensión profunda



de la pedagogía infantil. Esta división no solo facilita el aprendizaje al dosificar la información, sino que también aprovecha los principios de la psicología del color para guiar y enfocar la atención de los niños.

- **Segmentación Pedagógica:** Separar las vocales (el núcleo de toda sílaba y sonido) de las consonantes es un método de enseñanza efectivo. Permite a los niños dominar primero los sonidos fundamentales antes de combinarlos, reduciendo la sobrecarga cognitiva y construyendo el conocimiento de manera escalonada y firme.
- **Diseño de Tarjetas (Cards):** Cada letra está contenida en una tarjeta o botón individual con bordes redondeados y un color de fondo sólido. Este diseño es crucial por varias razones:
 - **Delimitación Clara:** Ayuda a los niños a distinguir y enfocarse en una letra a la vez.
 - **Amigable para Toque:** El tamaño generoso y el espacio entre ellas previenen errores táctiles y son ideales para dedos pequeños.
 - **Refuerzo Visual:** El color de fondo actúa como un código cromático constante para cada sección, reforzando la asociación subconsciente (vocales=amarillo, consonantes=azul).
- **Jerarquía Visual:** El uso de tipografía grande, en negrita y en mayúsculas para las letras maximiza la legibilidad. El contraste alto entre el texto (negro) y el fondo (color sólido) asegura que cada grafía sea perfectamente clara, lo que es esencial para el reconocimiento de formas. Véase en la fig. 3

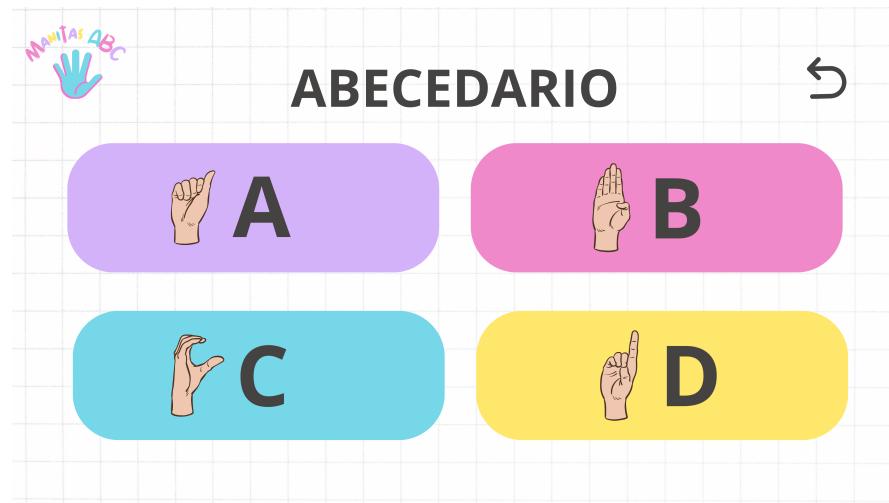


Figura 3: Vista general de las letras que puedes seleccionar

4. Sección donde puedes comprobar si tu seña es correcta

El diseño de esta interfaz es crítico, ya que no es solo una cámara, sino una herramienta de aprendizaje interactiva y de retroalimentación. Cada elemento está pensado para ser intuitivo, motivador y efectivo para un niño.

- **Vista Previa Central y Dominante:** La cámara ocupa el espacio principal, comunicando de inmediato que es el núcleo de la actividad. Esto le da importancia y deja claro al niño que sus acciones (sus señas) son el centro de atención. El marco rectangular con bordes redondeados suaviza la interfaz, haciéndola más amigable y menos técnica.

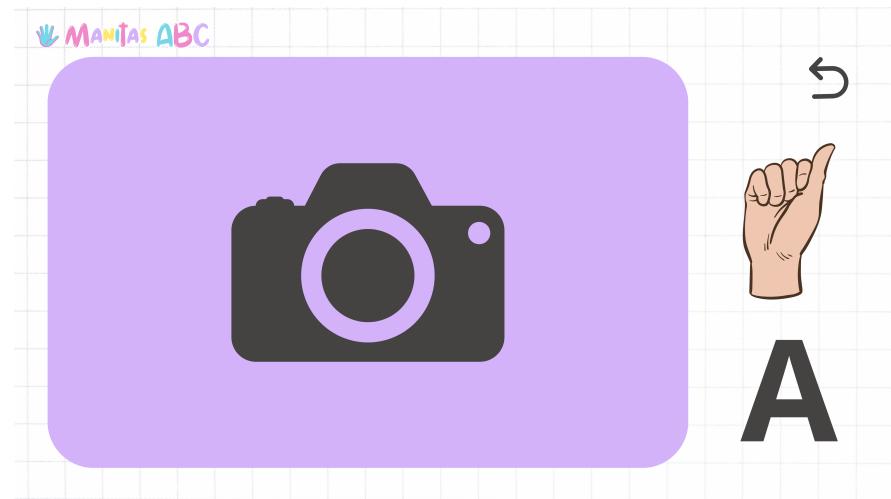


Figura 4: Sección donde puedes comprobar si tu seña es correcta

Etapa 2 — Implementación del Desarrollo. Se comenzó con el ticket *"Desarrollo de la interfaz"* (RS-002-RJLY), donde se implementó la herramienta HTML para la creación del menú principal de la plataforma, así como, la estructura interna para las secciones de nombres y letras (vocales y algunas consonantes).

Al terminar de desarrollar los esqueletos de estas vistas, se llegó a la conclusión de implementar una mejora, la migración a REACT (ticket *"Migración de HTML a React"* (RS-003-RJLY)), la cuál nos ayudaría a reutilizar código para componentes como las tarjetas de letras y la navegación en cámara.

Así mismo, se continúo con el ticket *"Implementación de CSS y JS"* (RS-004-PMCV) donde se aplicaron los estilos conforme al maquetado de la etapa 1, además de la redirección para el flujo de la aplicación.

Se desarrolló el ticket *"Redimensión de la cámara"* (RS-009-CVAS), cuyo objetivo era ajustar la vista de la cámara para proteger la privacidad del usuario. En un inicio se consideraron opciones como recortar la imagen, aplicar desenfoque o incorporar filtros sobre el rostro del infante. Sin embargo, se optó finalmente por una solución más práctica: mostrar mensajes de advertencia que



indiquen a los menores colocar únicamente la mano dentro del área visible de la cámara, evitando así comprometer su privacidad sin afectar el análisis de las señas realizadas, registramos dichos mensajes de alerta en el ticket “*Mensaje de alerta para el uso de la cámara*”(RS-028-CVAS).

Se generó el ticket “*Ajuste de la imagen representativa de la seña*”(RS-010-PMV), con el propósito de modificar la posición y el tamaño de la imagen de la seña, ya que se mostraba algo pequeña y ligeramente descentrada. Sin embargo, el ajuste resultó complicado, pues la imagen estaba vinculada tanto a la sección de cámara como a la de tarjetas; al corregirla en una, se desalineaba en la otra. Finalmente, se optó por una configuración intermedia que permitiera que la imagen se visualizara adecuadamente en ambas secciones.

6.2. Objetivo Específico 2: Entrenamiento de la IA para el Reconocimiento de Imágenes.

Etapa 1 — Elección de IA y material base. Para iniciar este proceso, se llevó a cabo la elección de la inteligencia artificial, una decisión clave que se registró en el ticket “*Elección de la IA*” (VC-006-RJLY). Se optó por implementar la API MediaPipe/Landmark de Google, ya que facilita la detección de la mano a través de una serie de nodos y líneas que permiten identificar con precisión su estructura.

Posteriormente, comenzó el ticket “*Imágenes de reconocimiento*” (RL-007-VCBF), en el que se recopiló al menos 50 imágenes de cada letra para después alimentar con ellas el entrenamiento de la IA. Al mismo tiempo, con el ticket “*Creación de la base de datos*” (RL-005-VCBF), se buscaba almacenar dichas imágenes en una base de datos, sin embargo, se decidió prescindir de este paso, ya que estas podían implementarse directamente en TensorFlow, registro que se ve en el ticket “*Sustitución de la BD por TensorFlow.js*” (RL-008-VCBF).

Etapa 2 — Implementación y entrenamiento. Después de definir las herramientas que se utilizarán para el entrenamiento y de recopilar los materiales necesarios, se implementó el primer

ticket respecto al entrenamiento "*Configuración de MediaPipe para entrenamiento*" (VC-011-RLJY), en este punto, el sistema comenzó a detectar la mano del usuario junto con las líneas y nodos que servirían para identificar las señas. Después con el ticket "*Construcción de dataset de landmarks de manos*" (RL-012-VCBF) se procesaron las imágenes recolectadas con ayuda de OpenCV y MediPipe para la creación de los datasets que se implementarán en el ticket "*Instalación de TensorFlow.js y conversión de modelos*" (VC-013-RLJY) donde se realizó el proceso de instalación de **TensorFlow** y la conversión de los datasets al formato bin y json.

Siguiendo con el entrenamiento básico, se abrió el ticket "*Entrenamiento para análisis*" (RL-014-RLJY) en el que se establecieron las métricas de predicción correspondientes para poder evaluar los modelos, además de verificar estas predicciones en base a los datos procesados y así mismo, la implementación con la cámara en tiempo real, siendo funcional en la sección de letras (vocales y algunas consonantes). Mientras que en la sección de "Mi Nombre" se implementó un ticket diferente "*Implementación de reconocimiento para sección Mi nombre*" (PM-021-RLJY), en este se modificó la modalidad de análisis, ya que debía realizarse en tiempo real con la cámara, analizando letra por letra en un periodo corto de tiempo y añadiendo métricas específicas para validar el avance y la precisión entre letras.

Como últimas implementaciones se realizaron dos tickets, el primero: "*Temporizador de preparación para realizar la señal*" (RS-027-CVAS), se colocó un temporizador sobre la cámara con el propósito de guiar al usuario y otorgarle un tiempo de preparación para realizar la señal. Y el segundo: "*Evaluación de eficacia de la señal*" (RS-029-CVAS) donde modificamos la cuadricula de visualización de 100 unidades por un indicador de 5 unidades representadas visualmente por cinco manos animadas de color amarillo, con el objetivo de buscar una unidad de visualización más amigable para nuestro público objetivo.



6.3. Objetivo Específico 3: Ejecución del Periodo de Prueba.

Durante este objetivo se realizaron algunas pruebas sobre el sistema, en el ticket "*Primer prueba de reconocimiento*" (RL-015-PMV) se observó una mala visualización de porcentajes respecto a la señal mostrada y errores en el análisis básico. El ticket "*Modificación de entrenamiento*" (PM-016-RLJY) da solución a esta primera prueba, brindando mejoría pues se modificaron las métricas de análisis y se precisó la lógica de entrenamiento. Sin embargo, al realizar estas modificaciones se desarrolló un problema más, ahora con la detección de la mano durante el análisis, se dió solución y se añadieron mejorías, como mensajes para mostrar umbrales en pruebas y ajustes en los tiempos de espera del análisis.

Siguiendo con el ticket "*Segunda prueba de reconocimiento*" (RL-017-PMV) se pudieron reconocer errores en el dataset respecto a la letra "m", la cual no era reconocida. Además, los análisis de landmark solían ciclarse provocando a su vez el no mostrar los porcentajes de una señal realizada. Buscando solucionar los problemas del ticket anterior se abrió el siguiente: "*Corrección en lógica de entrenamiento*" (PM-018-RLJY), en el cual se modificaron los parámetros y helpers para la detección de la letra "m", se realizaron limpiezas en el tiempo de captura y análisis y además, se implementaron mensajes del porcentaje obtenido de acuerdo al modelo analizado.

Se realizó el ticket "*Tercera prueba de reconocimiento*" (RL-019-PMV), donde se comienzan a observar confusiones en las letras "u" y "r". Esto pasaba porque ambas señales son muy parecidas entre sí, y el modelo no lograba distinguir los pequeños detalles que las diferencian. En el ticket "*Problemas de reconocimiento*" (PM-020-RLJY) se realiza una aprobación respecto a al buen funcionamiento en los parámetros de la mayoría de letras, exceptuando la "u" y "r", así que se toma la decisión de realizar un nuevo dataset para que el sistema pueda reconocerlas de manera más precisa.

Se implementa el ticket "*Primera prueba para deletreo de nombre*" (RL-023-PMV), en la cual analizamos problemas específicos de la sección "Mi nombre", como el deletreo no fluido, pues al realizar los cambios de letras el sistema se pausaba y, en algunas ocasiones, no permitía continuar

con el análisis. Por estos problemas se trabajó el ticket "*Corrección de sección Mi nombre*" (PM-024-RLJY). Ahí se revisó la lógica de las señas, se ajustaron funciones para que el sistema no guardara información de intentos anteriores y se agregó una función de reset para limpiar todo antes de volver a analizar. También se mejoró el flujo entre cada letra y el modal que muestra el resumen. Con estos cambios, la sección empezó a funcionar de manera más estable.

Después de eso se trabajó en el ticket "*Estados de análisis para Cámara*" (RL-025-RLJY), donde revisamos cómo y cuándo debía detenerse el análisis desde la cámara. Se ajustaron funciones y callbacks para que el sistema reconociera correctamente el final de cada intento y no siguiera analizando de más. También se actualizaron los elementos ref y se modificaron las funciones de inicio y detención de la cámara para evitar que el análisis siguiera activo hasta que el usuario interactuara con los modales. Con estos cambios, el control del análisis desde la cámara quedó más claro y estable.

Como últimas pruebas el ticket "*Segunda prueba de análisis en sección Mi nombre*" (RL-026-PMV) se verificó y restauró la funcionalidad general del sistema de análisis de deletreo. Tras las revisiones, todo el sistema quedó funcionando correctamente, excepto un problema puntual con el reconocimiento de las letras "U" y "R", que se atendió por separado en el ticket RL-022-VCBF. Una vez confirmado que el resto de la funcionalidad operaba sin fallas, el ticket se cerró.

En el ticket "*Corrección de datasets correspondientes a las letras "U" y "R"*" (RL-022-VCBF) se rehizo todo el proceso de actualización del modelo usando un dataset nuevo y más completo. Primero se regeneraron los landmarks con MediaPipe en un entorno separado para evitar conflictos entre librerías, obteniendo los archivos base con las características numéricas y etiquetas. Con ese dataset ya limpio y actualizado, se volvió a entrenar el modelo en TensorFlow/Keras aplicando mejoras para diferenciar mejor clases parecidas (como r y u), usando aumento de datos, pesos de clase y focal loss. Después del entrenamiento se guardó un nuevo modelo en formato Keras y se actualizaron las etiquetas usadas por el frontend. Finalmente, el modelo se convirtió a TensorFlow.js dentro de un entorno aislado para evitar incompatibilidades, generando correctamente el model.json y los shards que se colocaron en public/model, dejando el modelo listo para usarse directamente en



la app.

A continuación se muestra el cronograma de trabajo del sistemas correspondiente y sus etapas de trabajo. Véase en :

- *Anexo 2: Cronograma Ideal*
- *Anexo 3: Cronograma Real*

7. Análisis de Riesgos

7.1. Problemas de interacción entre el usuario y la plataforma.

Dado que el proyecto está dirigido a la niñez intermedia, un riesgo importante es que la interfaz no sea lo suficientemente clara o atractiva para este rango de edad. La complejidad en la navegación o en las instrucciones puede provocar frustración, desinterés o un uso incorrecto de la aplicación. Este riesgo es crítico porque afecta directamente la motivación y la efectividad pedagógica de la herramienta, reduciendo su impacto como recurso inclusivo.

7.2. Exceso en la petición o solicitud del análisis de la imagen.

El proceso de reconocimiento implica que el usuario coloque la mano frente a la cámara durante aproximadamente 5 segundos para capturar la señal, la cual será procesada posteriormente. El riesgo radica en que el tiempo de análisis de la imagen puede ser mayor al esperado, generando retrasos en la retroalimentación. Esto afecta la experiencia del usuario, ya que los niños requieren respuestas inmediatas para mantener la motivación y la continuidad en el aprendizaje. Si el sistema tarda demasiado en mostrar los resultados, puede provocar desinterés, pérdida de concentración o una percepción negativa de la herramienta. Por esta razón, se considera un riesgo relevante en el diseño de la plataforma.



7.3. Conectividad con la red.

La plataforma se encuentra alojada en un repositorio de GitHub esto implica que una conexión a internet estable es necesaria tanto para acceder al repositorio como para instalar o actualizar las dependencias del proyecto. En caso de que la conectividad sea deficiente, no será posible ejecutar la aplicación de manera correcta ni visualizar la interfaz de usuario, lo cual interrumpe el flujo de uso. Este riesgo es relevante porque afecta directamente la disponibilidad de la plataforma durante pruebas, demostraciones o sesiones de aprendizaje, limitando la continuidad del proyecto.

Glosario

Accesibilidad Digital Conjunto de prácticas que garantizan que una aplicación pueda ser utilizada por todas las personas, incluidas aquellas con alguna discapacidad.

API Conjunto de funciones y protocolos que permiten la comunicación entre distintos sistemas o herramientas.

Branch (Rama) Línea paralela de desarrollo utilizada para trabajar nuevas funciones sin afectar la versión principal del proyecto.

Clasificación Tarea de aprendizaje automático en la que un modelo identifica a qué categoría pertenece un dato.

Commit Registro de cambios realizado en el código dentro de un repositorio.

Componentes Reutilizables Elementos de interfaz o funcionalidad que pueden utilizarse en diferentes partes del proyecto.

Dataset (Conjunto de Datos) Colección de información o imágenes utilizada para entrenar un modelo de IA.



Entrenamiento del Modelo Proceso mediante el cual un modelo ajusta sus parámetros a partir de los datos para mejorar su capacidad de predicción.

Flujo de Usuario Ruta o secuencia de pasos que sigue un usuario dentro de la aplicación.

Gamificación Uso de elementos de juego (niveles, puntos, recompensas) para motivar a los usuarios.

Inferencia Etapa en la que un modelo ya entrenado realiza predicciones en tiempo real.

Landmarks (Puntos de Referencia) Coordenadas detectadas en la mano mediante algoritmos como MediaPipe, útiles para reconocer señales o posiciones.

Lengua de Señas Mexicana (LSM) Lengua viso-gestual empleada por la comunidad sorda en México.

Machine Learning (Aprendizaje Automático) Subcampo de la IA que permite que los sistemas aprendan patrones a partir de datos sin programación explícita.

MediaPipe Framework de visión computacional que permite detectar manos, rostros y poses en tiempo real.

Modelo de IA Estructura matemática que aprende patrones a partir de datos para hacer predicciones o clasificaciones.

Mockup Representación visual del diseño de una interfaz.

Modal Ventana emergente que aparece dentro de la aplicación para mostrar información o solicitar acciones específicas.

Overfitting Situación en la que un modelo aprende demasiado de su dataset y tiene bajo rendimiento con datos nuevos.

Preprocesamiento Transformaciones realizadas a los datos antes del entrenamiento, como norma-



lización, filtrado o extracción de landmarks.

Prototipo Versión interactiva de una interfaz que simula el funcionamiento de la aplicación antes de su implementación final.

Repositorios Espacios donde se almacena y gestiona el código del proyecto (GitHub, GitLab, etc.).

TensorFlow.js Biblioteca de JavaScript que permite ejecutar modelos de Machine Learning directamente en el navegador.

UI (User Interface) Diseño visual de los elementos con los que interactúa el usuario, como botones, íconos o tipografías.

Umbral de Confianza Porcentaje mínimo que debe alcanzar la predicción del modelo para ser considerada válida.

UX (User Experience) Experiencia general del usuario al interactuar con un sistema o aplicación.

Validación del Modelo Proceso donde se evalúa el rendimiento del modelo usando datos distintos a los del entrenamiento.

Wireframe Esquema básico de la interfaz que muestra la distribución de elementos sin detalles visuales.

8. Conclusión

El desarrollo del proyecto permitió integrar de forma coherente todos los elementos necesarios para que el sistema funcionara como una herramienta práctica y accesible. A lo largo del proceso se revisaron y ajustaron los componentes clave: desde la creación y corrección del dataset, el entrenamiento del modelo y la conversión a TensorFlow.js, hasta las mejoras en la interfaz y el comportamiento de la cámara y el temporizador. Cada ajuste, por pequeño que fuera, ayudó a que la experiencia final



fuera más estable y clara para el usuario.

El trabajo también dejó en evidencia la importancia de mantener un control detallado del flujo de versiones, los ambientes y las dependencias, especialmente en la parte del modelo. Resolver conflictos entre librerías, corregir el reconocimiento de ciertas letras y asegurar compatibilidad entre las herramientas fue tan importante como la implementación misma. Estas acciones permitieron entregar un sistema que no solo funciona, sino que es confiable y más fácil de mantener en el futuro.

En conjunto, el proyecto demuestra que, con una base bien estructurada y un enfoque iterativo, es posible construir soluciones que combinan visión por computadora, aprendizaje automático y una interfaz accesible para apoyar el aprendizaje de la Lengua de Señas Mexicana. Los resultados obtenidos dejan un camino claro para continuar ampliando el modelo, mejorar su precisión e incorporar nuevas funciones que enriquezcan aún más la aplicación.