

## Diseño de la Interfaz

Creación de la idea principal de la interfaz, con los colores #d4b2fa (lila), #76d7e8(azul), #ef89c9(rosa), #ffe76b(amarillo), dividir la plataforma en 4 secciones:

- Inicio: donde se colocará únicamente el logo de la marca y un botón de acceso, que te deberá dirigir a Temas.
- En las siguientes secciones deberá incluir un header con el logo de la marca, el título de la sección y con un botón de regresar, además de colocar en el cuerpo el contenido de los temas en recuadros con la gama de colores establecida.
- Cámara: en esta sección se conservará el header y se colocará la imagen (letra y seña) a recrear en una columna de lado izquierdo, mientras que la cámara en la columna derecha.

Estándares relacionados:

- ISO/IEC 25010 - Usabilidad: Evalúa claridad, facilidad de aprendizaje, accesibilidad y experiencia del usuario.
- ISO 19796-1 - Calidad en software educativo: Aplica porque el sistema está orientado a aprendizaje infantil (LSM).
- Ley General para la Inclusión de las Personas con Discapacidad (LGIPD): Respalda el deber de hacer interfaces inclusivas y accesibles.
- Reconocimiento de la LSM - Ley Federal de Telecomunicaciones y Radiodifusión: Relevante porque la interfaz debe ser adecuada para el aprendizaje de señas.

## Desarrollo Interfaz

Realización del código HTML de la interfaz, inicio, sección de temas, y sección de nombre, apegándose a el maquetado realizado en el ticket #CV-001-RSMF respetando las secciones indicadas:

- Inicio: donde se colocará únicamente el logo de la marca y un botón de acceso, que te deberá dirigir a Temas.
- En las siguientes secciones deberá incluir un header con el logo de la marca, el título de la sección y con un botón de regresar, además de colocar en el cuerpo el contenido de los temas en recuadros con la gama de colores establecida. Los temas a considerar son los siguientes:
  - Consonantes: Se debe mostrar únicamente consonantes con su imagen correspondiente.
  - Vocales: Realizar el etiquetado correspondiente a las vocales y su imagen.
  - Mi nombre: Se debe agregar un input de texto donde el usuario escribirá su nombre y un botón que lo direccione a la cámara.
- Cámara: en esta sección se conservará el header y se colocará la imagen (letra y seña) a recrear en una columna de lado izquierdo, mientras que la cámara deberá tener un espacio en la columna derecha.

Nota: Se ha decidido migrar a REACT, esto con la finalidad de reutilizar código de componentes como los son las tarjetas para cada letra y la navegación a la cámara.

# Implementación de CSS y JS

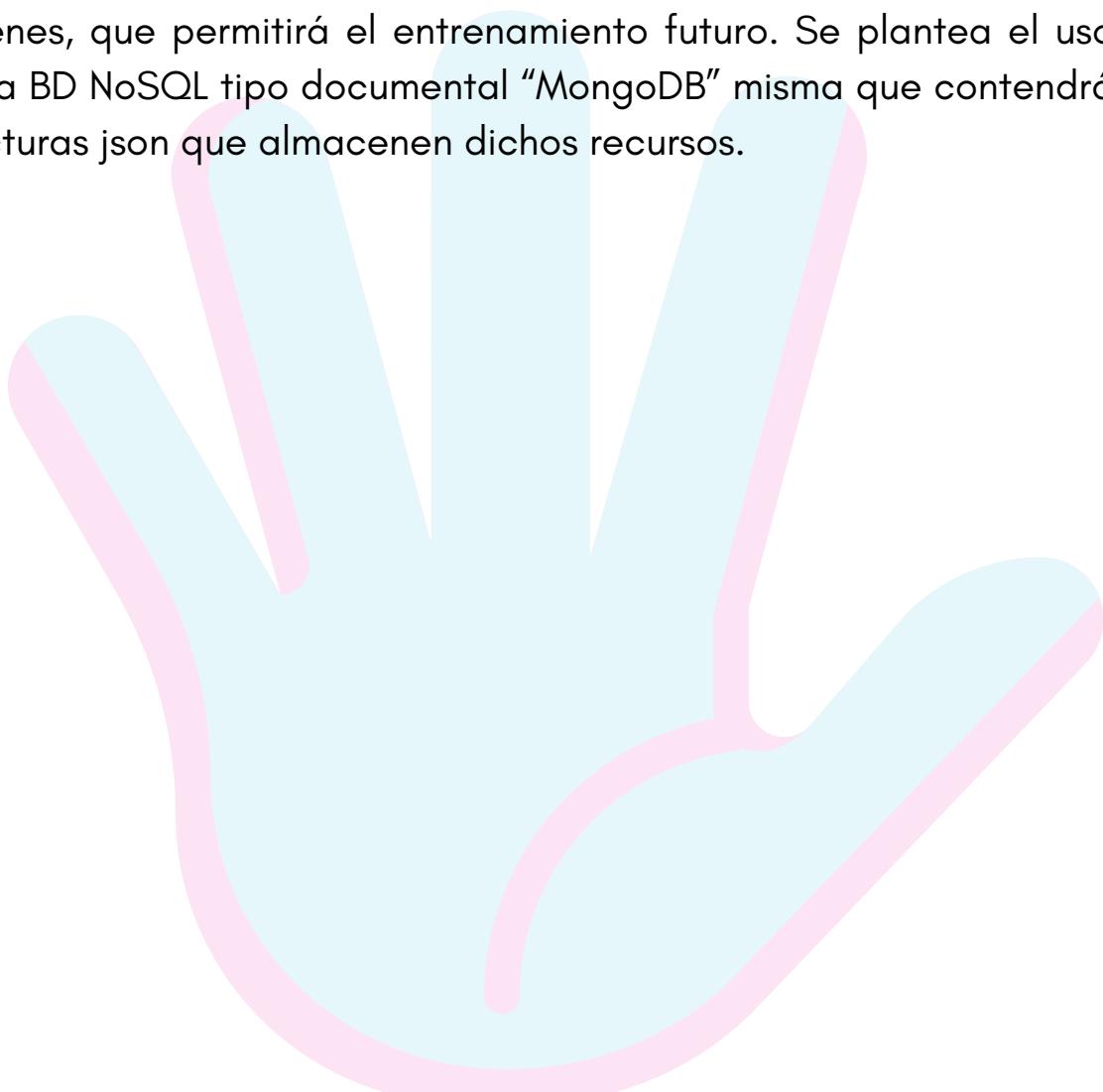
Implementación de estilos con CSS y del flujo de la aplicación web con JavaScript.

- Aplicación del estilos apegándose a los colores y tonalidades definidos en el maquetado previo.
- Adecuación de las imágenes representativas del gesto de la mano y ajuste del diseño de los botones, tomando en cuenta el posicionamiento, color y tamaño.
- Redirecciónamiento a cada uno de los HTML mediante el click en los botones para permitir la navegación dentro de la aplicación web.
- Activación para el uso de la cámara que permitirá en el futuro el reconocimiento de la señal de la mano.

Se sugiere modificar la posición de la imagen de referencia, para ofrecer una mejor visualización

## Creación de la base de datos

Creación de una base de datos para el almacenamiento de las imágenes, que permitirá el entrenamiento futuro. Se plantea el uso de una BD NoSQL tipo documental “MongoDB” misma que contendrá estructuras json que almacenen dichos recursos.



## Elección de la IA

Selección de la IA para el reconocimiento inicial de la mano del usuario a partir de la cámara. Detectando mediante el uso de nodos y líneas la interpretación de señas realizadas con la mano.

Se sugiere utilizar la API de Google Media Pipe Hands para la detección de la mano del usuario mediante la cámara de su navegador. Esta herramienta permitirá identificar en tiempo real los 21 puntos clave (landmarks) de la mano para representarlos mediante una red de nodos y líneas, lo que facilitará la interpretación de las señas realizadas.

## Imágenes de reconocimiento

Recolección de al menos 50 imágenes donde se visualicen las manos izquierda y derecha por separado de las señas para las vocales "a", "e", "i", "o", "u" y las consonantes "m" y "r" tomando en cuenta las siguientes consideraciones:

- Tonalidad de la piel.
- Tamaño de la mano a capturar.
- Género de la persona.
- Iluminación de fondo de la imagen.
- Calidad de la cámara.

## Sustitución de la BD por TensorFlow.js

Durante el proceso de investigación e implementación para realizar la conexión de la base de datos con la API Media Pipe Hands, se determinó que la conectividad de los componentes es incompatible por lo que se ha decidido trabajar con la API TensorFlow.js para continuar con el proceso de entrenamiento de modelos con el uso de python.

TensorFlow.js nos servirá para convertir nuestros entrenamientos a archivos con extensión .json + .bin y que sea compatible con la detección de manos que ya se ha implementado.

## Redimensión de la Cámara.

Con el propósito de salvaguardar la privacidad del usuario, se requiere ajustar la configuración de la cámara para que únicamente se visualice la mano del infante. Algunas consideraciones:

- La imagen deberá permanecer ubicada en la columna derecha de la interfaz.
- El alcance de la cámara deberá restringirse, o en su defecto, aplicarse un enfoque específico a la mano del usuario.

## Construcción de dataset

Se abrió un ticket para documentar el proceso de creación y entrenamiento del modelo de clasificación de letras basado en landmarks. Inicialmente se ejecutó el script de generación del dataset, encargado de capturar y procesar los puntos de referencia mediante la librería MediaPipe y almacenar los resultados en arreglos X\_landmarks.npy y y\_labels\_idx.npy, junto con el archivo classes.json. Durante esta etapa se detectó una falla en la letra "m", ya que 36 imágenes no fueron procesadas debido a errores en la detección de puntos.

Se procedió a revisar la implementación, identificando que el problema estaba asociado a inconsistencias en la lectura de los frames. Se ajustaron los parámetros de captura y se volvió a ejecutar el script, logrando completar el conjunto de datos sin errores.

Posteriormente, se entrenó el modelo con el script train\_landmark\_classifier.py. Se aplicó normalización de los datos, se dividió el dataset en entrenamiento y validación, y se entrenó una red MLP con capas densas y funciones de activación ReLU, empleando Dropout para regularización y EarlyStopping para prevenir sobreajuste. Finalmente, se evaluó el rendimiento, se guardó el modelo entrenado junto con las estadísticas de normalización (mu.npy, sigma.npy) y se cerró el ticket al confirmarse la correcta operación del sistema.

# Instalación de TensorFlow.js y conversión de modelos

Durante el desarrollo del proyecto, fue necesaria la instalación de TensorFlow.js, una herramienta fundamental para el análisis de la seña realizada por el usuario, proporciona las herramientas y arquitecturas de aprendizaje automático (Machine Learning y Deep Learning) necesarias para crear el sistema requerido para la aplicación.

## Problemas de reconocimiento

Durante el desarrollo e implementación de entrenamientos para lograr una buena detección de la señal se encontraron diversos problemas, algunos de ellos estaban relacionados con la codificación de la lógica, los umbrales de reconocimiento y los helpers de detección. Uno de los problemas que no ha tenido solución es que el sistema confunde la señal de la letra "U" reconociéndola como "R" o simplemente la desconoce.

Algunas razones específicas incluyen:

- Similitud de la configuración manual: Ambas señales, la de la 'R' y la 'U' en dactilología (el alfabeto manual), implican configuraciones de mano con los dedos índice y medio extendidos y juntos. La diferencia clave es muy sutil: en la 'R', los dedos se cruzan ligeramente, mientras que en la 'U', permanecen paralelos.
- Dificultades en la visión artificial: Los modelos de IA, que a menudo utilizan el reconocimiento de puntos de referencia de la mano (como los 21 puntos que detecta la librería MediaPipe de Google), pueden tener dificultades para capturar la distinción exacta entre los dedos cruzados (R) y los dedos paralelos (U). La proximidad y la oclusión parcial de los dedos pueden llevar a errores de clasificación.

Al realizar un análisis específico en el proceso de creación de datasets, se detectaron imágenes que no se habían procesado de manera correcta, quedando así con un porcentaje menor de datos por clasificar para las letras "U" y "R". Por ello, se recomienda realizar dos nuevos datasets exclusivos para esas letras.

## Corrección de dataset

En esta etapa se volvió a realizar el proceso completo de actualización del modelo a partir de un dataset nuevo con más imágenes. Para evitar conflictos entre librerías, el flujo se separó por ambientes: primero se generó nuevamente el dataset de landmarks con MediaPipe, obteniendo los archivos base (`X_landmarks.npy`, `y_labels_idx.npy` y `classes.json`) que contienen las características numéricas de cada imagen y sus etiquetas. Luego, con ese dataset ya actualizado, se reentrenó el modelo en TensorFlow/Keras usando un script que incluye estrategias para mejorar la separación entre clases parecidas (especialmente r y u), como aumento de datos en los landmarks, pesos de clase y focal loss. Al finalizar el entrenamiento se guardó un modelo nuevo en formato Keras (`model_landmarks.h5`) y se actualizaron los archivos de etiquetas que consume el frontend.

Finalmente, el modelo entrenado se convirtió a formato TensorFlow.js para poder integrarse en la aplicación React. Debido a incompatibilidades comunes entre versiones de TensorFlow, tensorflowjs y dependencias como JAX y NumPy, la conversión se ejecutó en un entorno aislado con versiones compatibles. Con esto se logró el reconocimiento de las letras U y R sin embargo se detectaron fallas para la r, en donde se logra un reconocimiento del 50% de las ejecuciones.

## Temporizador de preparación para realizar la seña

Con el propósito de guiar al usuario y otorgarle un tiempo de preparación para realizar la seña entre una y otra continuamente, se optó por determinar un intervalo de tiempo para realizar la seña indicada en la pantalla por una imagen ilustrativa.

Esta implementación de un temporizador que va x a x busca indicarle concretamente al usuario cual es el momento adecuado para que realice la seña frente a la cámara, además de agregar cierta interacción con el usuario evitando un mala experiencia, con el fin de evitar una posible frustración para el usuario que pueda deberse a una espera inesperada o que para su percepción sea una espera prolongada, lo cual interrumpira el flujo de la tarea y la generación de molestia por parte del usuario.

Dicho temporizador de tiempo esta creado con tres de los colores mas llamativos del sistema, los cuales son amarillo, azul y morado que se encuentran en un tono desaturado, continuando con la coherencia visual de nuestra identidad de marca.

## Mensaje de alerta para el uso de la cámara

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 esto con ayuda de la API de Google. 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.

El mensaje de advertencia dedicado para nuestro público objetivo será claro y preciso para su respectivo entendimiento sin llegar a ser un punto de perdida de foco o de generación de frustración.

## Evaluación de eficacia de la seña

Durante los tickets de pruebas se implementaron dos formas gráficas para la representación de la evaluación del reconocimiento de la seña, de los cuales son; el primero una ponderación representada porcentualmente del 0% al 100% de la eficacia de las señas o seña realizada y el segundo una cuadricula de visualización de 100 unidades para ilustrar visualmente el concepto de porcentaje, donde cada cuadro representa el 1%.

Con el objetivo de buscar una unidad de visualización más amigable para nuestro público objetivo y con ello poder continuar con una coherencia visual procurando nuestra identidad de marca se ha acordado sustituir la cuadricula de visualización de 100 unidades por un indicador de 5 unidades representadas visualmente por cinco manos animadas de color amarillo, lo cual contribuye a una armonía visual y a la escala de calificación de métricas más recomendada.