



**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: 26/11/2025

**Manual de Instalación:**

Manitas ABC

**Alumno(s):**

Carmona Velasco Andry Sebastian

Pestaña Márquez Valeria

Romero Lagunes Jossette Yeraldin

Ruiz Santiago María Fernanda

Visca Cocotzin Blanca Flor



# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivo</b>	<b>3</b>
<b>3. Alcance</b>	<b>4</b>
<b>4. Público Objetivo</b>	<b>4</b>
<b>5. Requisitos del Sistema</b>	<b>4</b>
5.1. Requisitos de Hardware . . . . .	5
5.2. Requisitos de Software . . . . .	5
5.3. Requisitos de Red . . . . .	5
5.4. Requisitos de Seguridad . . . . .	6
5.5. Requisitos Funcionales . . . . .	6
<b>6. Instalación del Entorno</b>	<b>6</b>
6.1. Clonado del Repositorio . . . . .	7
6.2. Instalación de Node.js y npm . . . . .	7
6.3. Compatibilidad de Versiones . . . . .	8
6.4. Instalación de Dependencias . . . . .	9
<b>7. Ejecución de la Aplicación</b>	<b>9</b>
7.1. Iniciar el Servidor de Desarrollo . . . . .	10
7.2. Recarga Automática . . . . .	10
7.3. Posibles Advertencias . . . . .	10
7.4. Detener el Servidor . . . . .	11
<b>8. Conclusión</b>	<b>11</b>



## 1. Introducción

El presente manual describe el proceso de instalación, configuración y preparación del entorno necesario para la ejecución del proyecto **Manitas ABC**, una aplicación desarrollada en **React** donde se incorporan tecnologías de visión por computadora mediante **MediaPipe**, **TensorFlow**, **TensorFlow.js** y modelos personalizados entrenados en **Python**.

Este documento está dirigido tanto a usuarios que requieren ejecutar la aplicación en un entorno local, como a desarrolladores que deseen comprender, mantener o continuar con la evolución del sistema. Por ello, se describe paso a paso el proceso necesario para preparar el entorno de trabajo, instalar todas las dependencias esenciales y ejecutar correctamente la aplicación. En esta guía se incluyen los requisitos previos relacionados con la instalación de Node.js y npm, la verificación de su funcionamiento, así como el procedimiento para clonar el repositorio y descargar automáticamente las librerías especificadas en el archivo `package.json`. De igual manera, se documenta la instalación de los módulos adicionales utilizados por el proyecto, entre ellos `@mediapipe/hands`, `@mediapipe/camera_utils`, `@mediapipe/drawing_utils` y `@tensorflow/tfjs`, necesarios para la detección, análisis y procesamiento de los puntos clave de la mano.

El propósito de este manual es garantizar que cualquier usuario o desarrollador pueda reproducir el entorno sin dificultades y contar con una referencia clara, ordenada y accesible que facilite el proceso de instalación. Con ello se busca asegurar que la aplicación funcione adecuadamente desde el primer inicio, brindando una experiencia óptima al interactuar con las funcionalidades basadas en visión por computadora que caracterizan al proyecto.

## 2. Objetivo

Se busca proporcionar una guía clara, estructurada y detallada para la instalación y configuración del entorno necesario para la ejecución del proyecto en un equipo local. Este documento permite que cualquier usuario o desarrollador pueda preparar correctamente las herramientas, dependencias



y componentes para la puesta en marcha de la aplicación, asegurando su funcionamiento.

### 3. Alcance

Se abarca el proceso completo de instalación del entorno de ejecución incluyendo la verificación e instalación de **Node.js** y **npm**, la obtención del proyecto desde el repositorio, la instalación de dependencias especificadas en el archivo **package.json**, así como las instrucciones necesarias para iniciar la aplicación en modo local.

El manual no cubre aspectos relacionados con el desarrollo avanzado del sistema, modificaciones del código fuente, entrenamiento de nuevos modelos den Python, ni la implementación del proyecto en servidores o entornos de producción.

### 4. Público Objetivo

Este documento está dirigido a dos tipos de usuarios:

- **Usuarios finales:** Niños y niñas en la etapa de niñez intermedia (6 a 12 años) que requieran ejecutar el proyecto en un entorno local sin necesidad de modificar su código o realizar tareas de programación.
- **Desarrolladores:** Integrantes del equipo de desarrollo interesados en comprender la estructura del proyecto, instalar sus dependencias y ejecutar la aplicación dentro de su entorno de trabajo para fines de mantenimiento, depuración o mejora del sistema.

### 5. Requisitos del Sistema

Se establecen las condiciones mínimas necesarias para garantizar el funcionamiento correcto de la aplicación, asegurando su instalación, operación y desempeño adecuado en los dispositivos del



público objetivo. Estos requisitos consideran las necesidades técnicas del procesamiento de imágenes en tiempo real, la ejecución de modelos de **Inteligencia Artificial** en el navegador y la interacción fluida de la interfaz diseñada para niños de 6 a 12 años.

### 5.1. Requisitos de Hardware

- Procesador de doble núcleo (2.0GHz o superior).
- 4GB de memoria RAM mínima; 8GB recomendados para un rendimiento óptimo al procesador de video en tiempo real.
- Cámara web de al menos 720p para capturar las señas con precisión.
- 200MB de espacio libre para almacenamiento temporal y caché del navegador.

### 5.2. Requisitos de Software

- Computadora o laptop con sistema operativo **Windows 10+**.
- **Navegador web** actualizado con soporte para **WebGL** y **TensorFlow.js** como **Chrome**, **Firefox** o **Edge**.
- Librerías internas: **Media Pipe Hands/Landmark**, **TensorFlow.js** y herramientas propias del proyecto.
- Conexión a internet par acceso y carga de dependencias.

### 5.3. Requisitos de Red

- Conexión estable de **10Mbps** recomendados para cargar modelos, recursos gráficos y comunicación con la plataforma.



- Permitir acceso a recursos estáticos y **scripts** alojados en el servidor o **CDN** correspondiente.

## 5.4. Requisitos de Seguridad

- Permisos de cámara otorgados por el usuario, necesarios para el reconocimiento de señas.
- No se requiere conexión **HTTPS**, dado que la aplicación se ejecuta de manera local mediante `localhost`.
- Aislamiento del procesamiento de imágenes, ya que todo el análisis se realiza localmente en el dispositivo, evitando el envío de información biométrica a servidores externos.

## 5.5. Requisitos Funcionales

- El sistema debe permitir la visualización en tiempo real del video capturado por la cámara.
- Debe reconocer y evaluar gestos del alfabeto de la Lengua de Señas Mexicana mediante **IA**.
- Debe registrar el desempeño del usuario y proporcionar retroalimentación inmediata.
- La interfaz debe ser accesible, gamificada y adecuada para niños en etapa de niñez intermedia.

Con estos requisitos se garantiza que la aplicación pueda funcionar de manera eficiente y segura, ofreciendo una experiencia de aprendizaje accesible, fluida y didáctica para el público infantil al que está dirigida.

## 6. Instalación del Entorno

Con el fin de asegurar el correcto funcionamiento de la aplicación, es necesario preparar un entorno local que permita ejecutar la interfaz desarrollada en **React** y las tecnologías de **visión por computadora** integradas mediante las librerías de **MediaPipe** y **TensorFlow.js**. En esta sección



se describe la obtención del proyecto, el proceso de instalación de las herramientas esenciales y la descarga automática de todas las dependencias necesarias.

## 6.1. Clonado del Repositorio

El código fuente de **Manitas ABC** se encuentra alojado en un repositorio de **GitHub**. **GitHub** es una plataforma que permite almacenar proyectos, gestionar versiones y facilitar la colaboración entre desarrolladores. A través de **GitHub** es posible acceder al proyecto, descargarlo y mantenerlo actualizado. El repositorio del proyecto puede visualizarse desde: [https://github.com/Joss-Y8/REACT\\_ManitasABC.git](https://github.com/Joss-Y8/REACT_ManitasABC.git).

Para obtener el proyecto en el equipo local es necesario utilizar **Git**, una herramienta de control de versiones que permite clonar repositorios desde la terminal. En caso de no contar con **Git** instalado, puede descargarse desde: <https://git-scm.com/>. Una vez instalado **Git**, se debe abrir una terminal del **Símbolo del Sistema** y navegar hasta el directorio donde desea almacenar el proyecto. Después, ejecutar el siguiente comando para clonar el repositorio:

```
git clone https://github.com/Joss-Y8/REACT_ManitasABC.git
```

Una vez completada la descarga, es necesario acceder a la carpeta del proyecto:

```
cd manitas_abc
```

A partir de este punto, el usuario dispone de todos los archivos necesarios para continuar con la instalación del entorno.

## 6.2. Instalación de Node.js y npm

Para ejecutar la aplicación, es necesario contar con **Node.js** y su gestor de paquetes **npm**. **Node.js** es un entorno de ejecución que permite utilizar **JavaScript** fuera del navegador, mientras que **npm (Node Package Manager)** es el gestor de paquetes encargado de instalar las dependencias



definidas en el archivo `package.json`. Ambos son indispensables para ejecutar la aplicación. Pueden instalarse mediante la terminal utilizando un gestor de paquetes del sistema operativo.

## Windows (mediante winget)

Para los dispositivos con sistema operativo **Windows**, la forma más sencilla de instalar **Node.js** mediante comandos es utilizando **winget**. Dentro de la terminal del **Símbolo del Sistema**, se debe ejecutar el comando:

```
winget install OpenJs.NodeJs
```

Una vez completada la instalación, verificar las versiones:

```
node -v
```

```
npm -v
```

Si ambos comandos muestran un número de versión, **Node.js** y **npm** se instalaron correctamente. Es importante tomar en cuenta que durante la instalación por comandos, **Node.js** se agrega automáticamente al **PATH**, por lo que no se requiere configuración adicional.

## 6.3. Compatibilidad de Versiones

Es importante considerar que el proyecto fue desarrollado utilizando **Node.js 18**, versión que mantiene compatibilidad completa con las dependencias de **React** y **TensorFlow.js** utilizadas en el sistema. Debido a ello se recomienda instalar **Node.js** en su versión **18.x**. Versiones más recientes, como **Node 20** o **Node 22**, pueden generar errores durante la instalación o ejecución del proyecto, ya que algunas herramientas incluidas en **react-scripts** no son totalmente compatibles con dichas versiones.





## 6.4. Instalación de Dependencias

Entre los archivos que se clonaron del repositorio se encuentra el archivo `package.json`, mismo que contiene la lista completa de las librerías necesarias para el funcionamiento de la aplicación. Para instalarlas, ejecutar dentro de la carpeta del proyecto el comando:

```
npm install
```

Con este comando se descargan automáticamente todas las dependencias, incluyendo:

- **React y React-DOM**: base de la interfaz del sistema.
- **@mediapipe/hands**, **@mediapipe/camera\_utils**, **@mediapipe/drawing\_utils**: herramientas de visión por computadora para detección de manos.
- **@tensorflow/tfjs**: motor de ejecución del modelo de **IA** en el navegador.
- Herramientas de pruebas y scripts incluidos por **Create React App**.

Este proceso creará la carpeta `node_modules`, donde se almacenan todos los paquetes instalados, esta carpeta, así como el archivo `package.json` no deben ser modificados.

## 7. Ejecución de la Aplicación

Una vez instalado el entorno y configuradas todas las dependencias necesarias, el siguiente punto radica en la ejecución de la aplicación **Manitas ABC** en modo de desarrollo. Este modo permite visualizar en tiempo real los cambios realizados en el código y facilita el proceso de revisión y pruebas.

Antes de iniciar la ejecución, es necesario asegurarse de que la terminal se encuentra dentro del directorio raíz del proyecto, es decir, carpeta que contiene los archivos ***package.json*** y las carpetas **src**, **public**.



## 7.1. Iniciar el Servidor de Desarrollo

Para iniciar la aplicación, se utiliza el siguiente comando:

```
npm start
```

Al ejecutar esta instrucción, **Node.js** pone en marcha un servidor de desarrollo local que compila el proyecto e inicia la interfaz de usuario en un entorno interactivo. Durante este proceso, la terminal mostrará mensajes de inicio provenientes de **React Scripts** y una vez completado, abrirá automáticamente el navegador predeterminado.

Por defecto la aplicación se ejecutará en:

```
http://localhost:3000/
```

Si por alguna razón el puerto 3000 se encuentra ocupado, el sistema solicitará permiso para utilizar otro puerto disponible.

## 7.2. Recarga Automática

El servidor de desarrollo cuenta con una función de recarga automática, lo que significa que cualquier cambio realizado en los archivos del proyecto se verá reflejado inmediatamente en el navegador sin necesidad de reiniciar la aplicación manualmente.

## 7.3. Posibles Advertencias

Durante la ejecución inicial, es posible que aparezcan mensajes de **advertencia (warnings)**. Estos no impiden la ejecución de la aplicación y en la mayoría de los casos, se relacionan con configuraciones internas del entorno. No obstante, si se presenta un error y la aplicación no inicia, se recomienda revisar:

- Que las dependencias se hayan instalado correctamente.



- Que la versión de **Node.js** sea compatible.
- Que no existan procesos ejecutándose en el puerto utilizado.

En caso de encontrar advertencias en la terminal, se recomienda:

- Usuarios finales: Continuar con el uso de la plataforma sin realizar modificaciones a los archivos del proyecto.
- Desarrolladores: Notificar a los encargados del área correspondiente a **Desarrollo de Software** las advertencias encontradas para su inmediata solución.

## 7.4. Detener el Servidor

Para detener la ejecución del proyecto, dentro del **Símbolo del Sistema** basta con utilizar la siguiente combinación:

`Ctrl + C`

Esto cerrará el servidor de desarrollo y liberará el puerto utilizado.

## 8. Conclusión

La instalación y configuración del entorno de desarrollo se completaron de manera satisfactoria, garantizando que todas las herramientas necesarias (incluyendo **Node.js**, **npm** y los componentes adicionales) funcionen correctamente y de forma compatible con el sistema. Asimismo, la documentación elaborada proporciona una guía clara sobre los pasos de instalación, configuración y ejecución del sistema, permitiendo reproducir el entorno de forma confiable. Con ello, se asegura que cualquier integrante del equipo pueda continuar con el desarrollo o resolución de problemas sin ambigüedades. En conjunto, el proceso realizado permite contar con un entorno estable, documentado y alineado con los objetivos técnicos del proyecto, sentando las bases para un desarrollo eficiente y sostenible.