# Documentación del Proyecto - Recomendaciones Agrónomas

# Descripción del Proyecto

Este proyecto utiliza TensorFlow.js para proporcionar recomendaciones en tiempo real para el cuidado de plantas. Basado en parámetros como la humedad, luz solar y temperatura, el modelo predictivo se ejecuta directamente en el navegador, lo que permite procesar los datos de manera local y rápida.

# Tecnologías Utilizadas

* TensorFlow.js: Para la creación y ejecución de modelos de aprendizaje automático en el navegador
* Node.js: Backend del proyecto, maneja las peticiones y sirve los archivos
* HTML5/CSS3/JavaScript: Para la construcción de la interfaz gráfica y lógica de cliente.
* npm: Utilizado para gestionar las dependencias del proyecto.

# Dependencias

El proyecto incluye las siguientes dependencias, que están listadas en el archivo `package.json`:

* tensorflow/tfjs
* express
* dotenv
* cors

# Estructura del Proyecto

La estructura del proyecto es la siguiente:

* /public: Archivos estáticos como HTML, CSS y JavaScript para la interfaz gráfica.
* /src: Contiene los scripts para la lógica del backend y configuración del servidor
* /models/: Modelos de TensorFlow.js utilizados para las recomendaciones.
* server.js: Script principal que inicializa el servidor en Node.js.

# Funcionalidades Principales

1. Recomendaciones Basadas en Parámetros de Plantas: El sistema toma datos como humedad, luz y temperatura y genera recomendaciones para el riego y cuidados específicos.
2. Predicciones en Tiempo Real: A través de TensorFlow.js, el modelo se ejecuta en el navegador, ofreciendo resultados instantáneos sin necesidad de conectarse a servidores remotos.
3. Interfaz de Usuario Intuitiva: La interfaz gráfica está diseñada para que el usuario ingrese fácilmente los datos y reciba sugerencias de forma rápida y eficiente.

# Recomendaciones para Mejoras Futuras

1. Ampliación del Modelo\*\*: Se puede ampliar el modelo para incorporar más parámetros, como el tipo de suelo o fertilizantes específicos.
2. API Externa para Datos Climáticos: Integrar una API que proporcione datos climáticos en tiempo real podría mejorar la precisión de las recomendaciones

Aplicación Móvil: Convertir el proyecto en una aplicación móvil para mejorar la accesibilidad y el uso en campo.

# Sistema de Recomendaciones Agrónomas

Este proyecto implementa un sistema de recomendaciones para el cuidado de plantas utilizando TensorFlow.js. El sistema está diseñado para proporcionar recomendaciones basadas en las condiciones específicas de las plantas y los datos proporcionados.

# Descripción

El Sistema de Recomendaciones Agrónomas es una aplicación web que utiliza técnicas de aprendizaje automático para analizar datos sobre las condiciones de las plantas y proporcionar recomendaciones personalizadas para su cuidado. Utiliza TensorFlow.js para ejecutar modelos de aprendizaje automático directamente en el navegador, lo que permite una experiencia de usuario fluida y rápida sin necesidad de servidores backend.

# Características

* Recomendaciones Personalizadas: Ofrece sugerencias específicas para el cuidado de las plantas según sus necesidades.
* Uso de TensorFlow.js: Implementación de modelos de aprendizaje automático directamente en el navegador.
* Interfaz Intuitiva: Fácil de usar para usuarios con diferentes niveles de experiencia en el cuidado de plantas.
* Análisis en Tiempo Real: Procesa y analiza los datos en tiempo real para proporcionar recomendaciones inmediatas.

# Requisitos

- \*\*Node.js\*\*  
- \*\*npm\*\* o \*\*yarn\*\*

# Instalación

Clona el repositorio:  
```bash  
git clone https://github.com/tokien736/recomendaciones-agronomas.git  
```  
Navega al directorio del proyecto:  
```bash  
cd recomendaciones-agronomas  
```  
Instala las dependencias:  
```bash  
npm install  
```  
o  
```bash  
yarn install  
```

# Uso

Inicia el servidor de desarrollo:  
```bash  
npm start  
```  
o  
```bash  
yarn start  
```  
Abre tu navegador y navega a http://localhost:3000 para ver la aplicación en acción.  
  
Introduce los datos de la planta (como la humedad del suelo, la luz recibida, y la temperatura) y obtén recomendaciones personalizadas para su cuidado.

# Ejemplo de Código

A continuación se muestra un ejemplo de cómo se puede utilizar TensorFlow.js para crear un modelo de recomendaciones:

```javascript  
import \* as tf from '@tensorflow/tfjs';  
  
// Definir el modelo  
const model = tf.sequential();  
model.add(tf.layers.dense({ units: 32, activation: 'relu', inputShape: [inputFeatures] }));  
model.add(tf.layers.dense({ units: 1, activation: 'linear' }));  
  
// Compilar el modelo  
model.compile({ optimizer: 'adam', loss: 'meanSquaredError' });  
  
// Entrenar el modelo  
async function trainModel(data) {  
 const { inputs, labels } = data;  
 await model.fit(inputs, labels, {  
 epochs: 50,  
 batchSize: 32,  
 });  
}  
  
// Predecir recomendaciones  
async function predict(inputData) {  
 const prediction = model.predict(tf.tensor2d([inputData], [1, inputFeatures]));  
 return prediction.dataSync()[0];  
}  
```