

# Documentación del Proyecto - Teachable Machine en Web

---

## 1. Descripción General del Proyecto

Este proyecto implementa una interfaz web que utiliza modelos de aprendizaje automático creados con **Teachable Machine**, una herramienta de Google, para reconocer patrones visuales en tiempo real desde la webcam.

El modelo cargado reconoce distintas clases visuales y muestra los resultados en una interfaz atractiva diseñada en tonos rosados. La predicción se presenta mediante **barras de porcentaje**, lo que permite una visualización clara e intuitiva de los resultados. Una vez iniciado el sistema, el botón de "Start" desaparece para evitar una interacción innecesaria y para brindar una mejor experiencia al usuario.

---

## 2. Tecnologías Utilizadas

Este sistema se apoya en varias tecnologías modernas que permiten una implementación rápida y eficiente en el navegador. Las principales tecnologías usadas son:

- **HTML:** Define la estructura de la página web.
  - **CSS:** Estiliza los elementos con una temática rosada, centrando el contenido y utilizando un contenedor distintivo para mostrar los resultados.
  - **JavaScript:** Controla la lógica de activación de la cámara, carga del modelo y predicciones.
  - **TensorFlow.js:** Biblioteca que permite cargar y usar modelos de IA directamente en el navegador.
  - **Teachable Machine:** Plataforma visual para entrenar modelos de aprendizaje automático sin código.
  - **Webcam API:** Captura video en tiempo real desde la cámara del usuario.
-

### 3. Funcionamiento del Sistema

El flujo general del sistema es el siguiente:

1. El usuario ingresa a la página web y ve un botón llamado "Start".
  2. Al hacer clic en el botón, se solicita acceso a la cámara web.
  3. El botón desaparece y se inicia la captura en vivo desde la webcam.
  4. El modelo previamente entrenado (y alojado en la nube por Teachable Machine) se carga usando `tmlImage.load(...)`.
  5. El video en tiempo real se procesa y cada frame es analizado por el modelo para determinar a qué clase corresponde.
  6. Los resultados de predicción se actualizan continuamente y se muestran como **barras de progreso animadas**, cada una correspondiente a una clase reconocida.
- 

### 4. Instrucciones de Uso

**Para probar el proyecto:**

- Asegúrate de tener una cámara web funcional.
- Abre el archivo HTML del proyecto en un navegador moderno (como Google Chrome o Firefox).
- Haz clic en el botón "Start".
- Permite el acceso a la cámara si el navegador lo solicita.
- Observa cómo se inicia la predicción automática.
- Las clases entrenadas serán visibles debajo del video como barras de progreso con porcentajes.

**Para reemplazar el modelo:**

- Entrena un nuevo modelo en Teachable Machine.
  - Exporta el modelo como **TensorFlow.js**.
  - Copia la nueva URL del modelo (`model.json` y `metadata.json`).
  - Sustituye la constante URL en el script JavaScript con tu nuevo enlace.
-

## 5. Investigación Documentada (Parte 1)

### ¿Qué es Teachable Machine y cómo se entrena un modelo ahí?

Teachable Machine es una herramienta visual gratuita de Google que permite a cualquier persona crear modelos de aprendizaje automático. Se puede usar para entrenar modelos de reconocimiento de imagen, sonido o postura.

#### Entrenamiento:

1. Selecciona el tipo de modelo (imagen, sonido o postura).
  2. Crea distintas clases (por ejemplo: Gato, Perro, Persona).
  3. Sube ejemplos o captura datos con tu webcam o micrófono.
  4. Haz clic en "Entrenar modelo" y Teachable Machine generará un modelo entrenado en el navegador.
- 

### ¿Cómo se exporta un modelo para uso en web (TensorflowJS)?

Después de entrenar un modelo en Teachable Machine:

1. Haz clic en **Exportar modelo**.
2. Elige la opción **TensorFlow.js**.
3. Puedes descargar los archivos o usar el **enlace del modelo en la nube**.
4. Los archivos que se generan incluyen:
  - model.json: arquitectura y pesos del modelo.
  - metadata.json: información sobre las clases.
  - Archivos binarios .bin que contienen los pesos del modelo.

Estos archivos se pueden integrar fácilmente a una web con TensorFlow.js.

---

## 6. Investigación Documentada (Parte 2)

### ¿Qué es TensorflowJS y cómo se utiliza para cargar un modelo?

**TensorFlow.js** es una biblioteca de JavaScript que permite ejecutar modelos de aprendizaje automático directamente en el navegador, sin necesidad de servidores.

Para usarlo:

1. Incluye el script de TensorFlow.js y teachablemachine-image en tu HTML.

2. Usa `tmlImage.load(modelURL, metadataURL)` para cargar el modelo.
3. Usa `model.predict()` para obtener predicciones desde una imagen o un video.

Es ideal para proyectos educativos, interactivos o prototipos rápidos.

---

### ¿Qué es .h5 y cómo se exporta un modelo en Python (Keras)?

.h5 es la extensión de los archivos en formato **HDF5**, usados por Keras para guardar modelos completos (estructura + pesos + configuraciones).

En Python:

```
from keras.models import load_model
```

```
# Guardar
```

```
model.save('modelo.h5')
```

```
# Cargar
```

```
modelo_cargado = load_model('modelo.h5')
```

Este formato es común cuando trabajas con Python y quieres reutilizar el modelo entrenado en otro entorno.

---

### ¿Qué es GitHub y cómo se puede versionar un proyecto allí?

**GitHub** es una plataforma online para alojar código fuente utilizando el sistema de control de versiones **Git**. Permite:

- Hacer seguimiento de los cambios (commits).
- Colaborar con otras personas.
- Crear ramas para trabajar en nuevas funciones sin afectar el proyecto principal.
- Subir proyectos públicos o privados.
- Ver el historial completo de modificaciones.

#### **Pasos básicos para usar Git y GitHub:**

```
git init
```

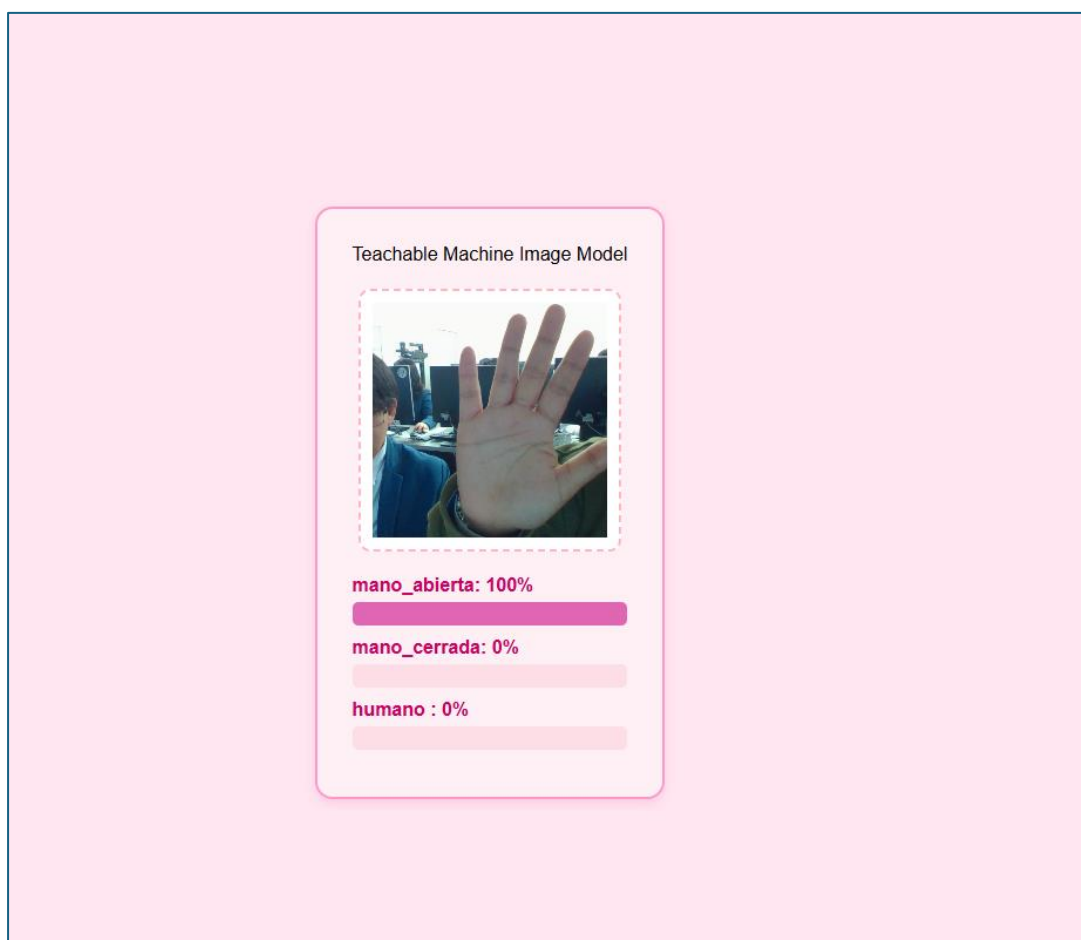
```
git add .
```

```
git commit -m "Primer commit"
```

```
git remote add origin https://github.com/usuario/repositorio.git
```

```
git push -u origin main
```

fácilmente antes de convertirlo a PDF?



*Demostración de entrenamiento.*