

## **API para entrenar un modelo**

**Endpoint:** /train

- Método: POST
- Descripción: Entrena un modelo de clasificación utilizando un conjunto de datos CSV.
- Parámetros:
  - dataset: Nombre del archivo CSV con los datos de entrenamiento.
  - model\_name: Nombre del modelo a guardar.
- Respuesta:
  - Si el entrenamiento es exitoso, devuelve un mensaje: {"message": "Modelo entrenado correctamente"}.
  - Si no se encuentra el archivo CSV, devuelve un error 404 con detalles.
  - Si ocurre un error durante el entrenamiento, devuelve un error 500 con detalles.

## **API para hacer predicciones con el modelo IRIS**

**Endpoint:** /predictIRIS

- Método: POST
- Descripción: Realiza una predicción utilizando el modelo IRIS preentrenado.
- Parámetros:
  - model\_name: Nombre del modelo previamente guardado.
  - sepal\_length, sepal\_width, petal\_length, petal\_width: Características de la flor IRIS.
- Respuesta:
  - Devuelve un diccionario con la predicción: {"prediction": resultado}.
  - Si no se encuentra el modelo, devuelve un error 404 con detalles.
  - Si ocurre un error durante la predicción, devuelve un error 500 con detalles.

## **API para hacer predicciones con un modelo de estudiantes**

**Endpoint:** /predictStudents

- Método: POST
- Descripción: Realiza una predicción utilizando un modelo de estudiantes (asumiendo que existe un modelo diferente).
- Parámetros:
  - Lista de características relacionadas con los estudiantes (coincidiendo con las columnas del conjunto de datos).
- Respuesta:
  - Devuelve un resultado de predicción basado en las características proporcionadas.
  - Si no se encuentra el modelo, devuelve un error 404 con detalles.
  - Si ocurre un error durante la predicción, devuelve un error 500 con detalles.