

Instituto Tecnológico de Culiacán

Carrera: Ing. Tecnologías de la información y comunicaciones

TOPICOS SELECTOS DE INTELIGENCIA ARTIFICIAL

DR. ZURIEL DATHAN MORA FÉLIX

Grupo: 12:00 – 01:00 PM

Tarea 2 Visión artificial

Equipo:

Rodrigo Alonso Páez Gastélum

Adrián Jaret Rendón Ríos

**Índice**

Contents

[**Objetivo de la practica** 2](#_Toc199508623)

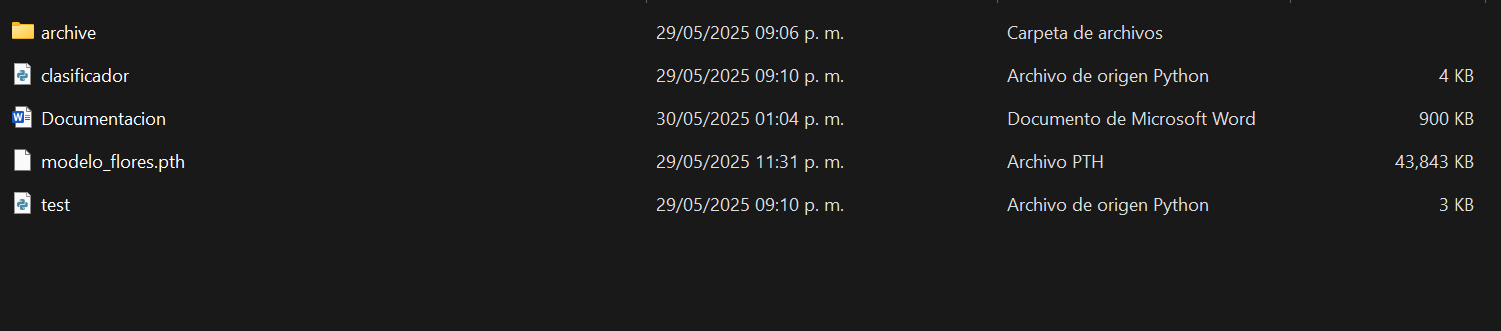
[**Evidencia del código** 2](#_Toc199508624)

[**Evidencia de los resultados** 2](#_Toc199508625)

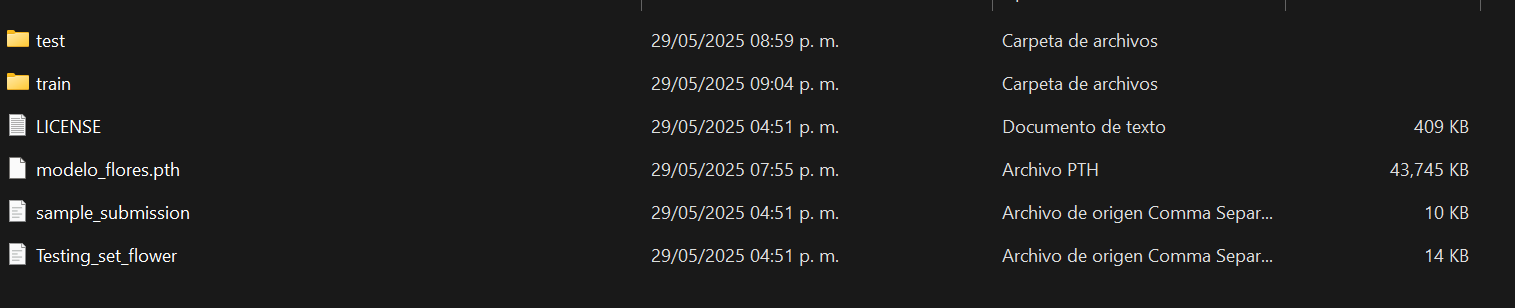
# **Objetivo de la practica**

**La siguiente practica tiene como objetivo desarrollar un sistema de visión artificial que identifique plantas utilizando el Deep learning.**

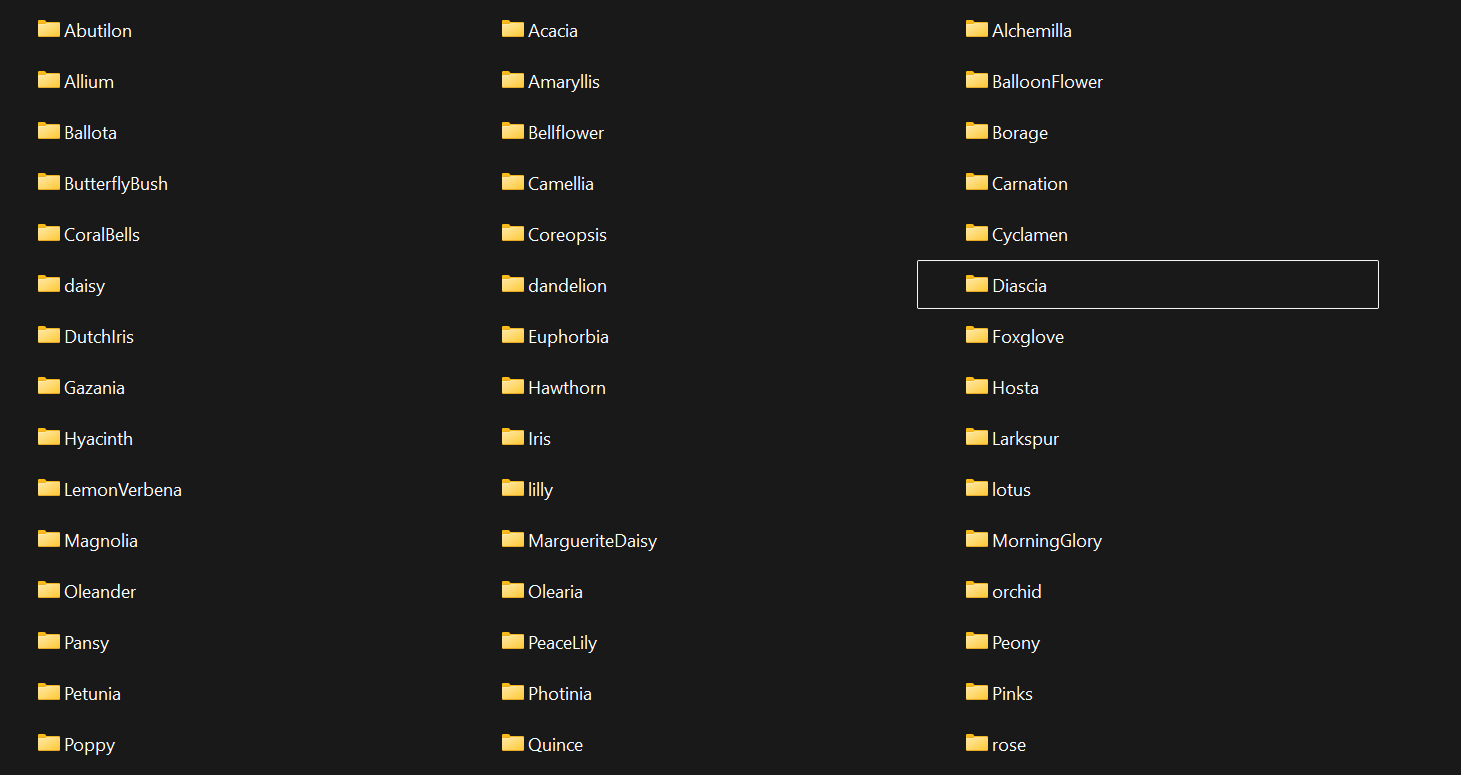
# **Evidencia del proyecto**



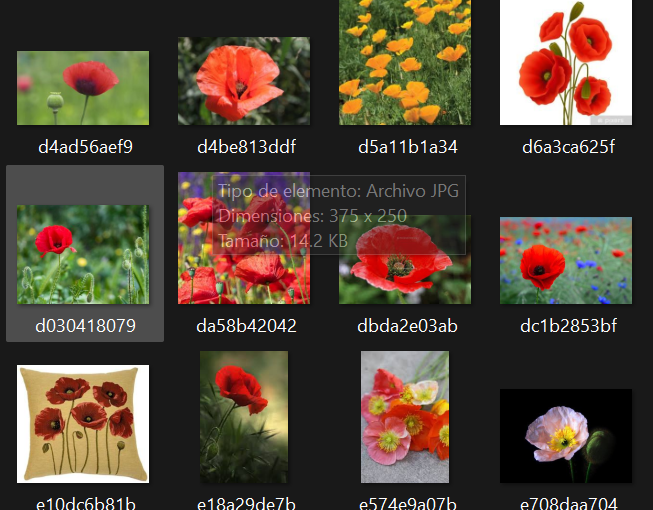
En esta primera imagen encontramos la raíz del proyecto donde se cuenta 5 elementos, este mismo documento, un archivo generado que es el identificador de flores, dos documentos de Python, clasificador, el cual es el entrenador del modelo y test el cual es el que ejecuta el modelo



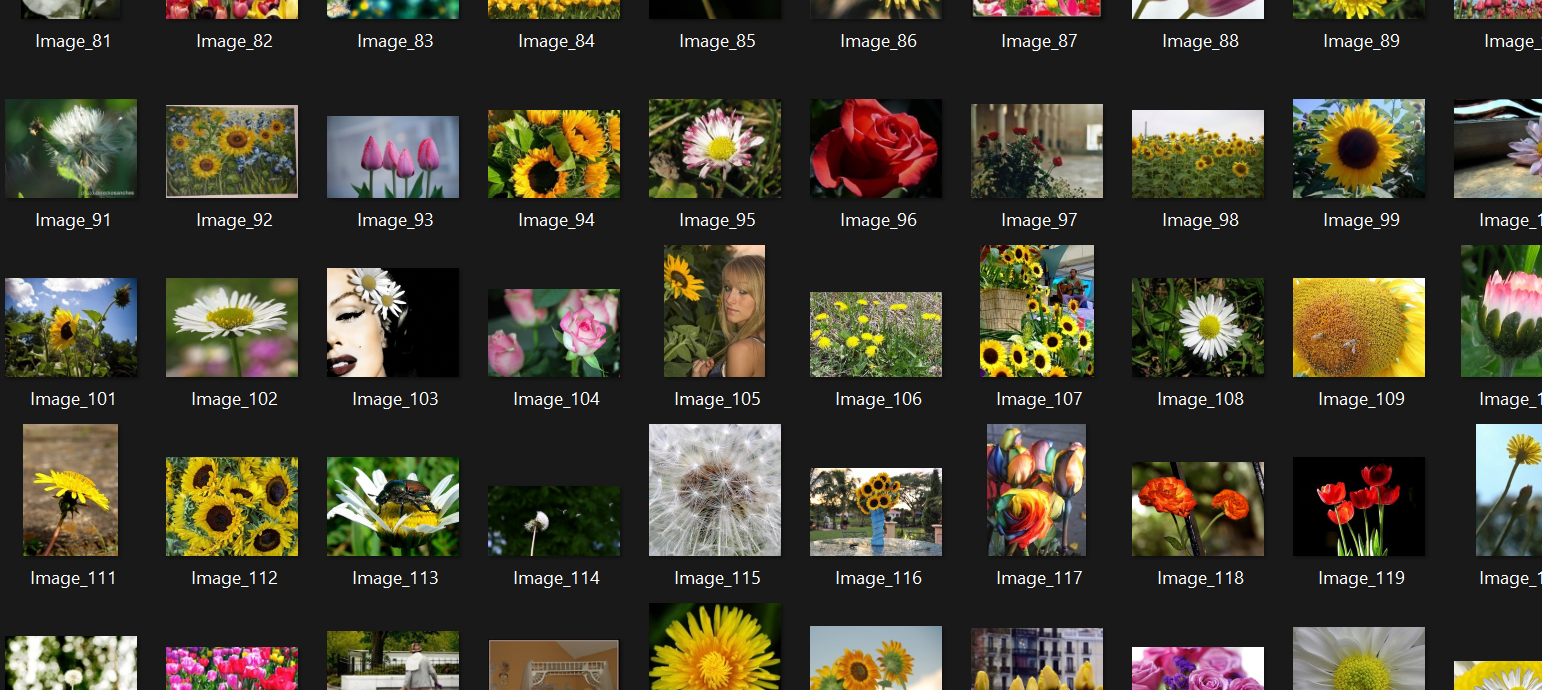
La siguiente imagen contiene los elementos de la carpeta “archive”, los cuales contiene dos dataset.csv; Testing\_set\_flower.csv que contiene las imágenes, sample\_submission.csv que contiene el tipo de flor de las imágenes. LICENSE.TXT es un archivo donde viene los links de propiedad intelectual de las fotos extraídas



Dentro de la carpeta train están las imágenes de cada uno de los tipos de flores identificables para el modelo, teniendo 54 tipos distintos de flores identificables



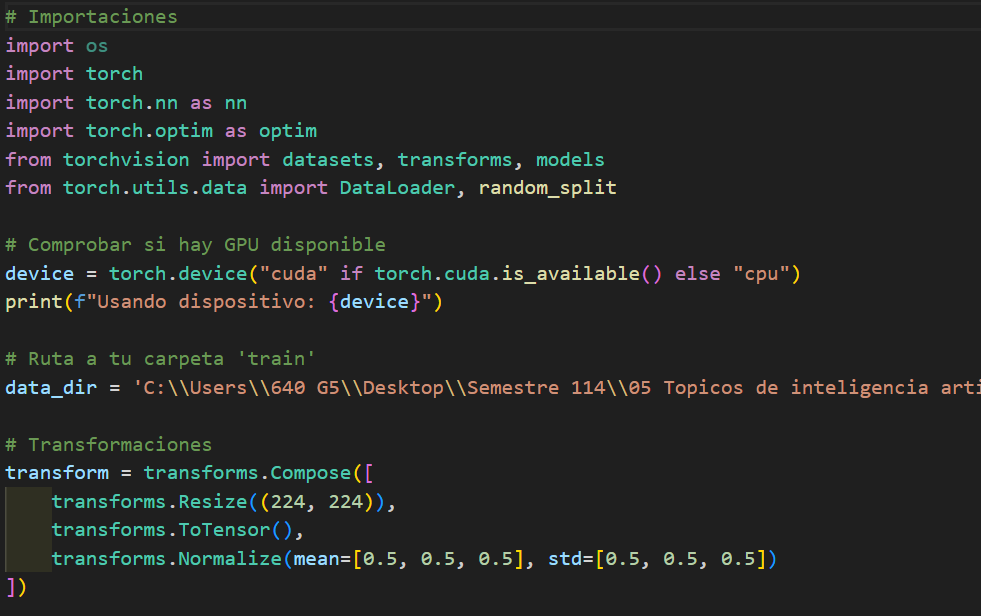
con un número aproximado de 50 imágenes de un tipo determinado de flor cada una de las carpetas.



En la carpeta test hay 924 imágenes para probar que tan bien puede distinguir los diferentes tipos de flores, habiendo diferentes tipos y en diferentes ambientes, donde se pone aprueba la capacidad de distinguir entre un tipo determinado.

Por el lado del código el archivo clasificador.py sirve para entrenar el modelo de Deep Learning, el test.py pone aprueba lo aprendido; donde

**Clasificador.py**

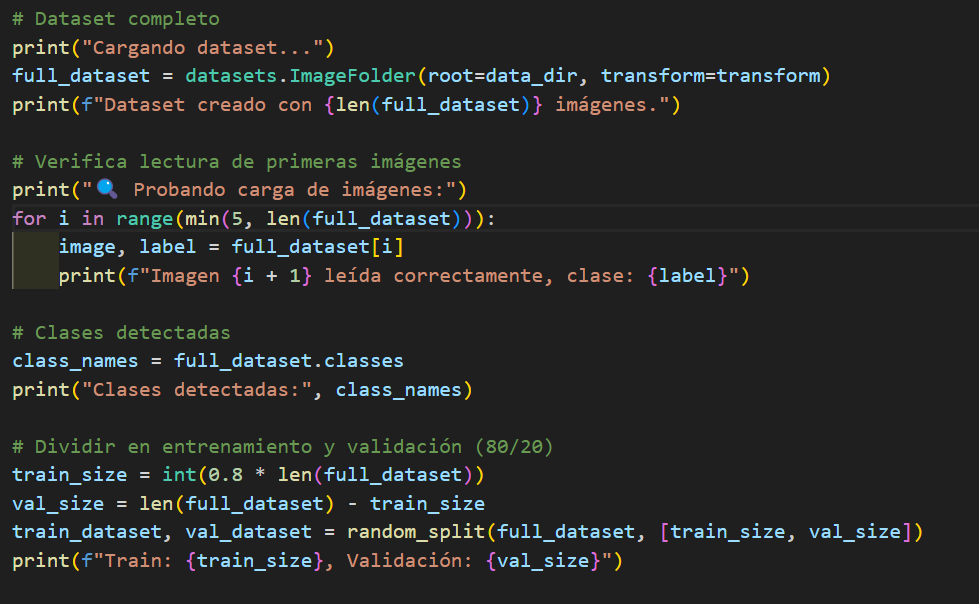


Se importan las bibliotecas necesarias para el proyecto.

Determinar si el entrenamiento usará GPU (para acelerar cálculos) o CPU.

Se especifica la ruta para el entrenamiento donde se buscará las fotos antes mencionadas de los tipos de flores.

Se procesan las imágenes para que sean compatibles con el modelo.

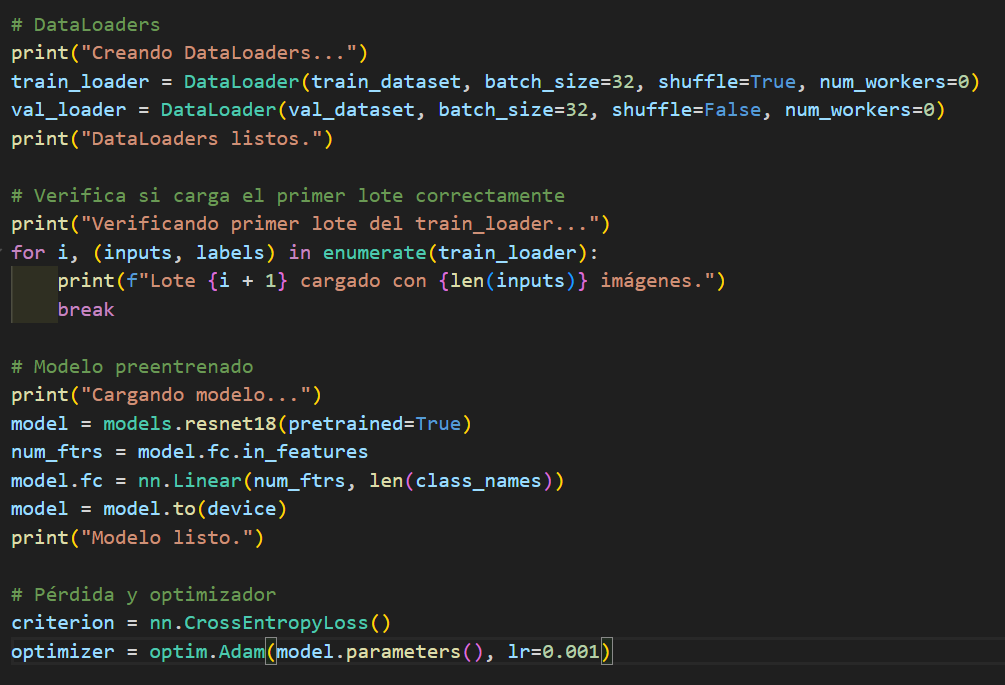


Cargar las imágenes desde la carpeta especificada y aplicarles las transformaciones definidas.

Verificar que las imágenes se cargan correctamente.

Lee las clases de flores y obtiene los nombres del dataset.

Distribuye el data set en 80% de entrenamiento y 20% de validación.



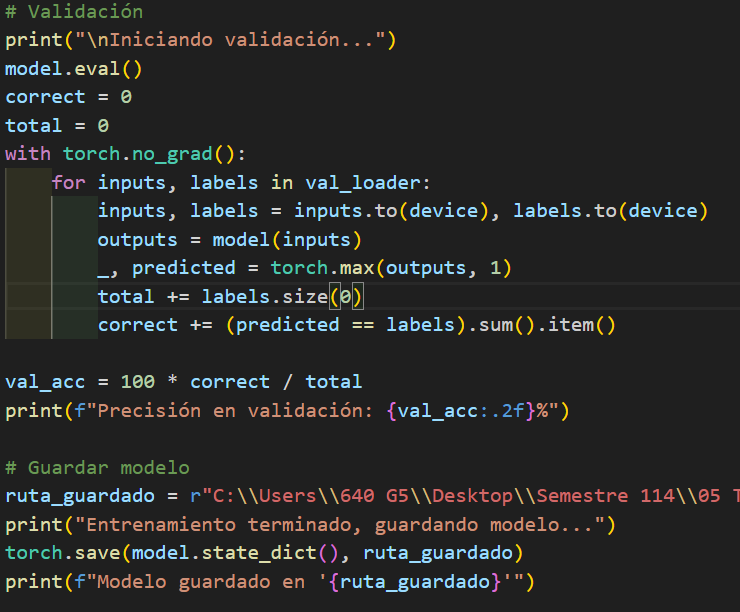
Crea las iteraciones para cargar los datos durante el entrenamiento.

Confirma que los datos se carguen correctamente.

Carga los modelos y setea el peso.

Configura la perdida y optimización.

Bucle principal de entrenamiento donde configura los epoch donde usa los data set procesa los datos.



Cambia el modelo a modo evaluación. Esencial para obtener resultados consistentes en validación.

Se especifica la ruta donde se guarda el modelo

Se guardan los pesos.

**Test.py**

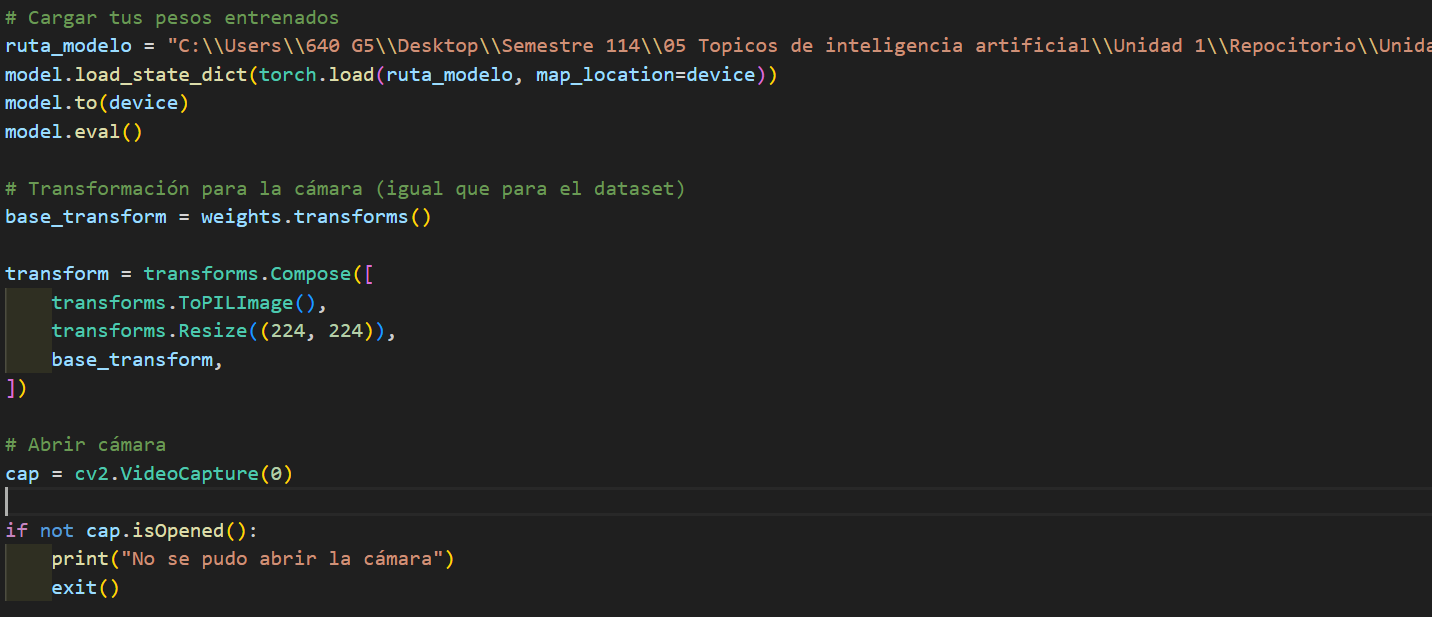


**Detecta y utiliza la GPU si está disponible para acelerar los cálculos, de lo contrario usa CPU.**

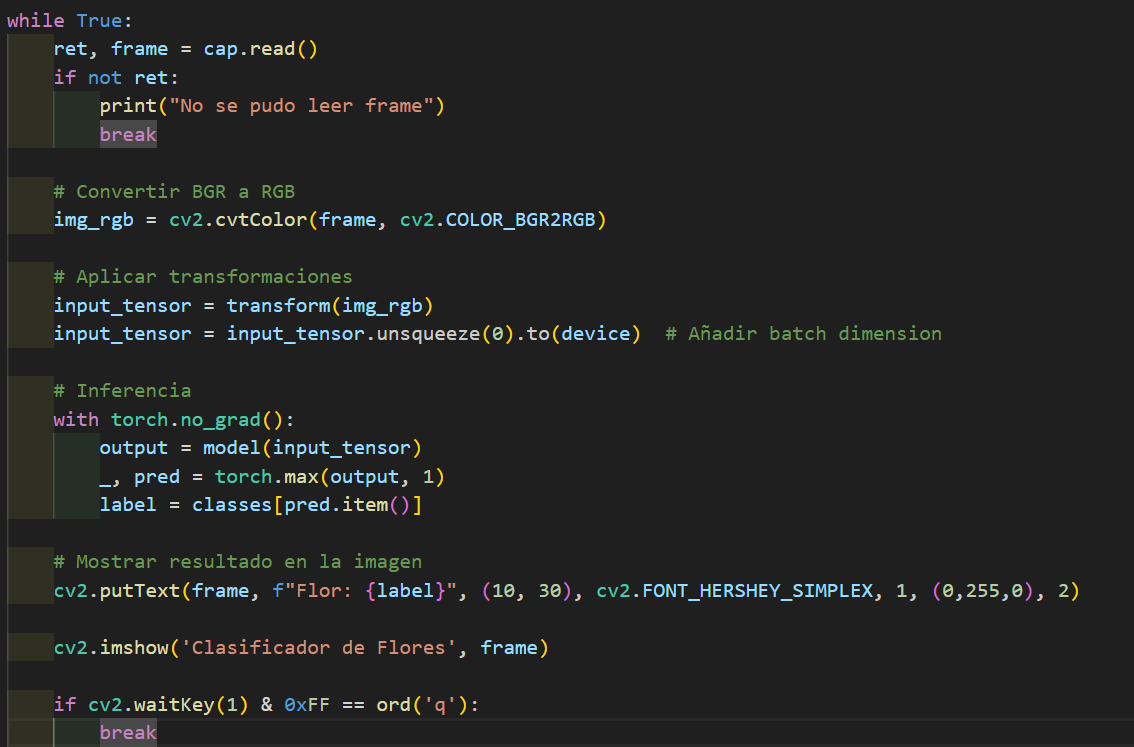
**Lista completa de 54 especies de flores que el modelo debe reconocer.**

**Utiliza ResNet18 preentrenado en ImageNet.**

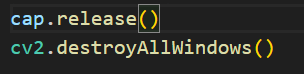
**Personalizar el modelo para nuestra tarea específica.**

****

**Carga los pesos anteriormente entrenados**

****

**Este bloque de código implementa el bucle principal de inferencia en tiempo real para el clasificador de flores usando la cámara web.**

****

**Deja de usar la cámara**

**Cierra la ventana**

# **Evidencia de los resultados**

