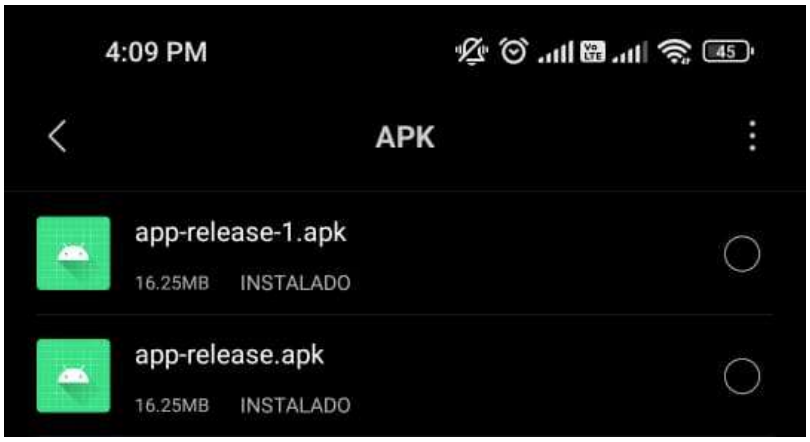
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
INGENIERIA EN CIENCIAS Y SISTEMAS
LABORATORIO INTELIGENCIA ARTIFICIAL 1
SERGIO ALFONSO FERRER GARCÍA - 200915305
ANDREA VIRGINIA CHAVARRÍA GUZMÁN – 200920081

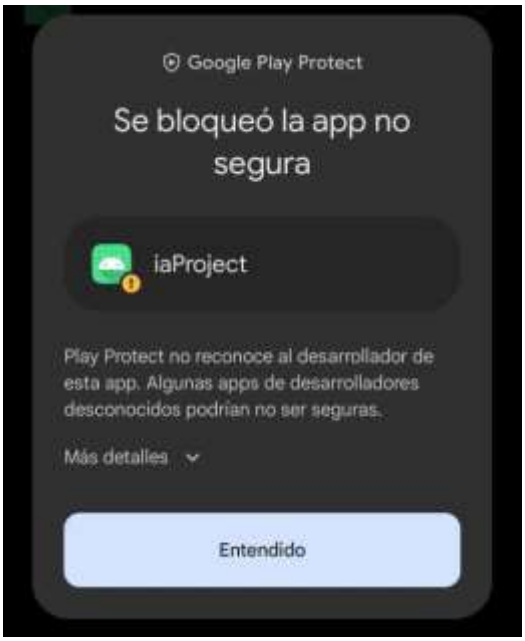
MANUAL DE USUARIO

1. **Descargar de la APP app-release-1.apk, y proceder a instalar con los permisos correspondientes.**

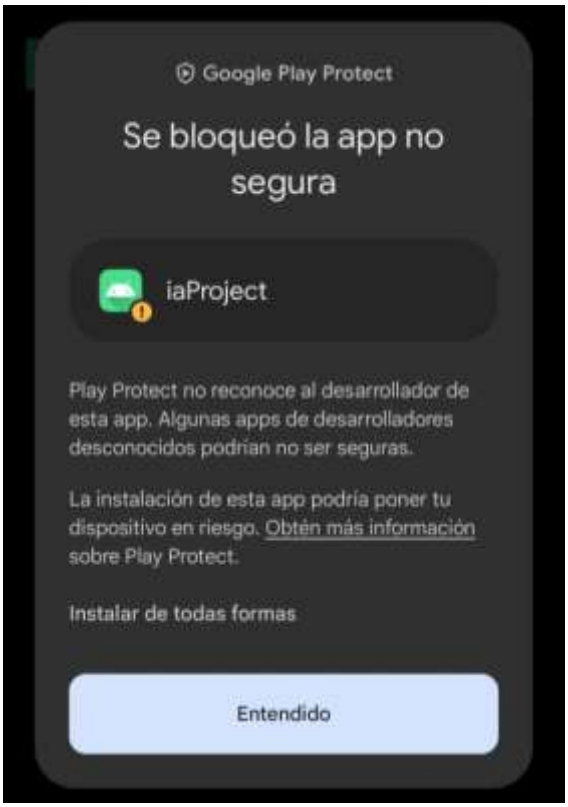


2. **Si la aplicación no se ha instalado nunca, instalar, en este caso se actualizará con el nuevo reléase.**

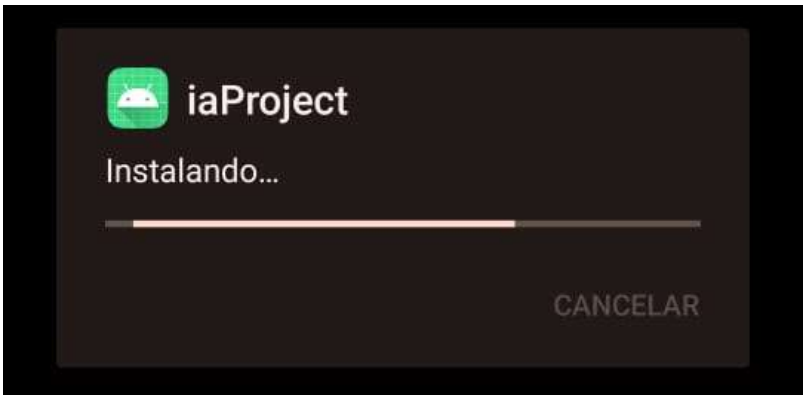
3. **El siguiente mensaje de seguridad en normal en los dispositivos Android, procederemos a dar click en “Más detalle”, para continuar con la instalación.**



4. **En la siguiente pantalla emergente, presionaremos donde dice “instalar de todas formas”, sabiendo que nuestra app no implica ningún riesgo para nuestro móvil.**



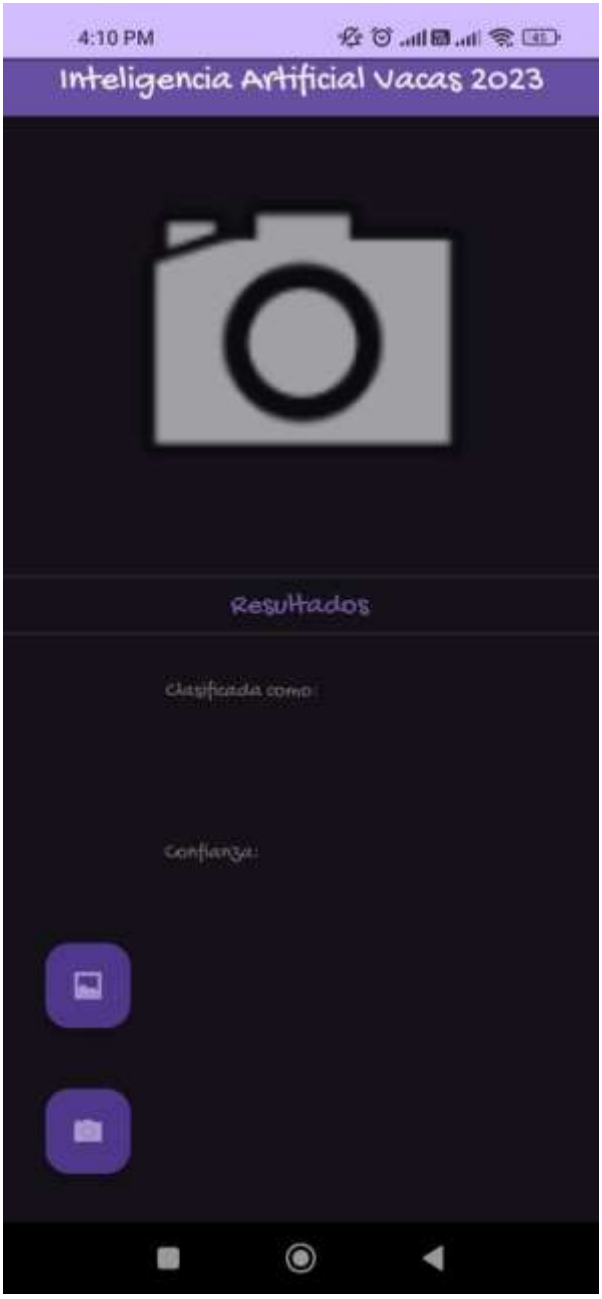
5. **Esperamos a que se realice la instalación, esto solo tomara un par de minutos.**



6. **Se realiza un análisis de seguridad, es un proceso normal en los dispositivos Android.**

7. **El análisis termino y la app no implica ningún riesgo para continuar con la instalación.**

8. **A continuación, se muestra la aplicación, en el cual procederemos a realizar la comparación de imágenes de dos tipos de animales, PANDAS Y GATOS, tenemos dos opciones, cargar imágenes del dispositivo y tomar una foto con el dispositivo.**



9. **Carga de imágenes desde el dispositivo, primero se compara la imagen de un PANDA y posteriormente la comparación de la imagen de un GATO.**

10. Y la comparación de una imagen utilizando la cámara del móvil.

MANUAL TECNICO.

Aplicación desarrollada para comparar imágenes de dos tipos diferentes de animales, PANDAS Y GATOS, el desarrollo de la aplicación fue realizado utilizando Android Estudio y las librerías de TensorFlow 2.0 además de TensorFlow Lite, a continuación de describirá más detenidamente las tecnologías utilizadas.

1. TENSORFLOW

TensorFlow es una biblioteca de código abierto desarrollada por el equipo de Google Brain que se utiliza para realizar tareas de aprendizaje automático y aprendizaje profundo. Fue diseñada para ser flexible y escalable, lo que la hace adecuada para una amplia gama de aplicaciones de inteligencia artificial.

2. ANDROID ESTUDIO

La integración de TensorFlow con Android Studio permite desarrollar aplicaciones de Android que aprovechan modelos de aprendizaje automático para realizar tareas específicas, como reconocimiento de imágenes, procesamiento de lenguaje natural, entre otras.

3. TENSORFLOW LITE

TensorFlow Lite es una versión ligera de TensorFlow diseñada específicamente para aplicaciones en dispositivos móviles y sistemas integrados. Esta versión está optimizada para ejecutar modelos de aprendizaje automático en dispositivos con recursos limitados, como teléfonos inteligentes, tabletas, dispositivos IoT (Internet de las cosas) y otros dispositivos embebidos.

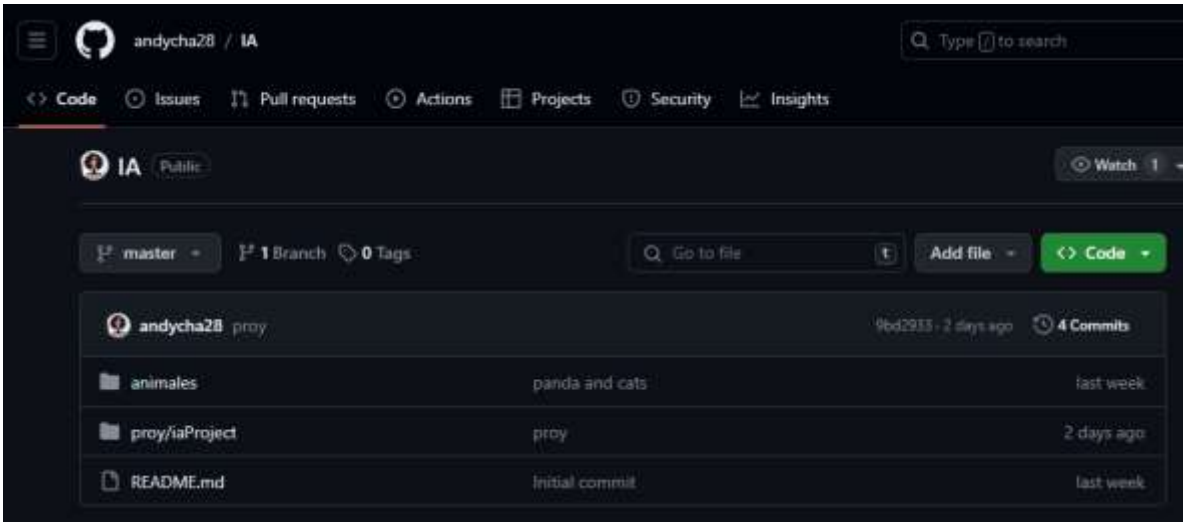
4. DATASET

En el contexto de TensorFlow, un "dataset" (conjunto de datos) se refiere a una colección de datos que se utiliza para entrenar, validar o probar modelos de aprendizaje automático. En TensorFlow, la manipulación eficiente y la carga de datos se gestionan mediante el uso de la API tf.data. Esta API permite crear y manipular flujos de datos que se pueden utilizar directamente para entrenar modelos.

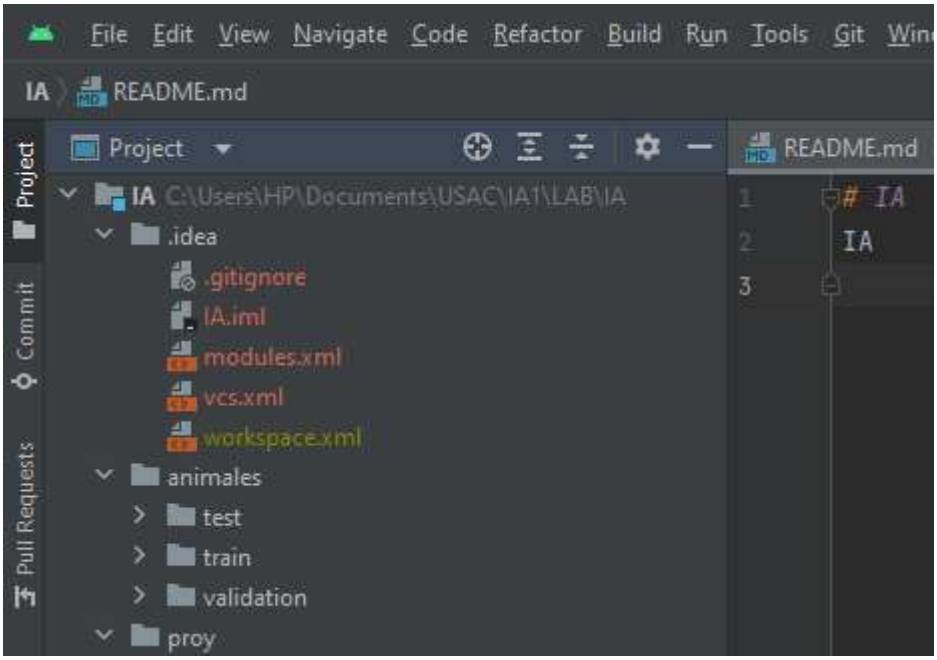
Un conjunto de datos en TensorFlow generalmente consta de ejemplos de entrada y sus correspondientes etiquetas. Por ejemplo, en un problema de clasificación de imágenes, cada ejemplo podría ser una imagen y su etiqueta sería la clase a la que pertenece la imagen.

5. PROYECTO Y REPOSITORIO

El proyecto tiene el nombre de “IA” y se encuentra alojado en el siguiente repositorio.
https://github.com/andycha28/IA

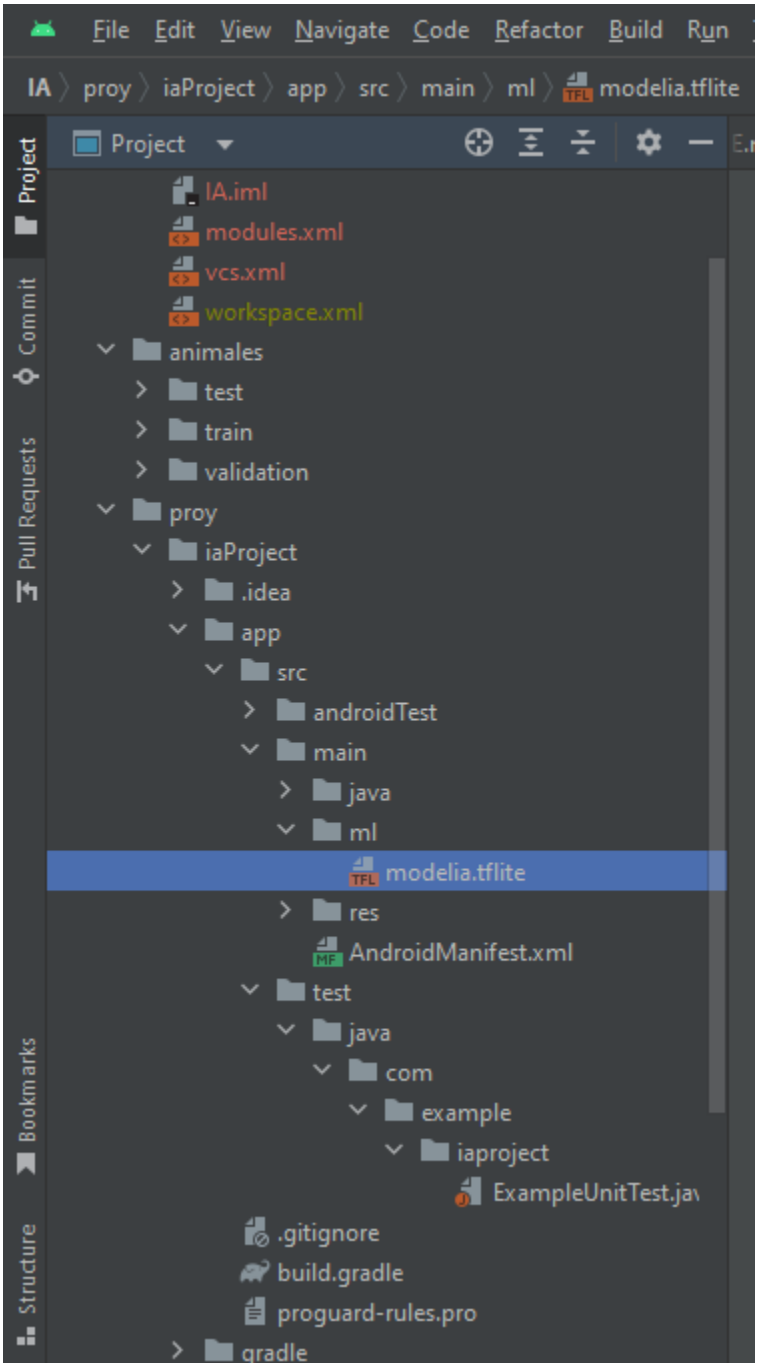


Para la clasificación y aprendizaje se utilizaron tres diferentes carpetas (test, train, validation), en estas carpetas se encuentran alojadas aproximadamente 4000 imágenes de GATOS Y 1000 imágenes de GATOS.



6. MODELO

El modelo de aprendizaje es un archivo llamado “modelia.tflite”, los podemos identificar por que tienen la extensión .tflite, en este caso la tenemos agregada en la carpeta src, en donde se encuentra el código fuente principal de la appp.



7. COLAB

https://colab.research.google.com/drive/1jrtFpCvWDtQoKviC_QDvPKSLBOWHqNx6#scrollTo=pfwpNxfllzQi

REPOSITORIO

```
[ ] !git clone https://github.com/andycha28/IA.git

Cloning into 'IA'...
remote: Enumerating objects: 6660, done.
remote: Counting objects: 100% (495/495), done.
remote: Compressing objects: 100% (471/471), done.
remote: Total 6660 (delta 1), reused 495 (delta 1), pack-reused 6165
Receiving objects: 100% (6660/6660), 375.12 MiB | 15.59 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Updating files: 100% (6199/6199), done.
```

LIBRERIAS TENSORFLOW

```
[ ] import tensorflow as tf
    import matplotlib.pyplot as plt
    print("TensorFlow version:", tf.__version__)

    TensorFlow version: 2.15.0
```

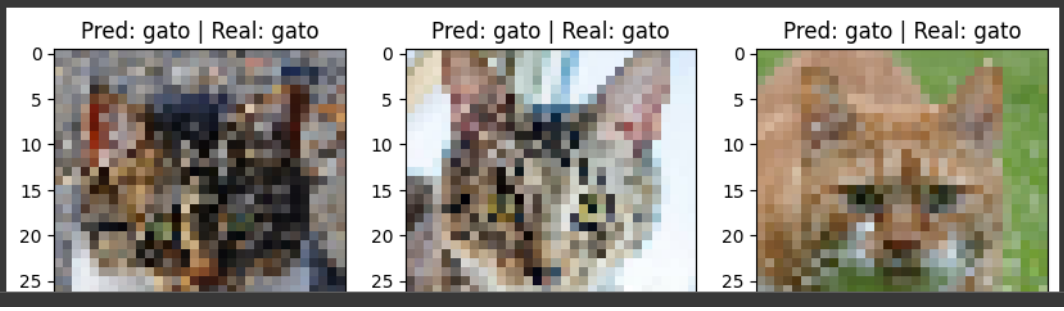
MODELOS

CLASIFICACION

```
[ ]  import numpy

     plt.figure(figsize=(10,10))
     for images, labels in test_ds.take(1):
       classifications = model(images)
       # print(classifications)

       for i in range(6):
         ax = plt.subplot(3, 3, i + 1)
         plt.imshow(images[i].numpy().astype("uint8"))
         index = numpy.argmax(classifications[i])
         plt.title("Pred: " + class_names[index] + " | Real: " + class_names[labels[i]])
```



METADATOS

```
#METADATOS

!pip install tflite-support-nightly
```

```
[ ]  #METADATOS

     from tflite_support.metadata_writers import image_classifier
     from tflite_support.metadata_writers import metadata_info
     from tflite_support.metadata_writers import writer_utils
```

```
[ ]  TF_MODEL_FILE_PATH = 'modelia.tflite' # The default path to the saved TensorFlow Lite model

     interpreter = tf.lite.Interpreter(model_path=TF_MODEL_FILE_PATH)
```

8. **Finalmente tenemos una aplicación funcional con la cual podemos comparar imágenes de PANDAS y GATOS, mostrando el porcentaje de confianza y la clasificación del tipo de imágenes.**