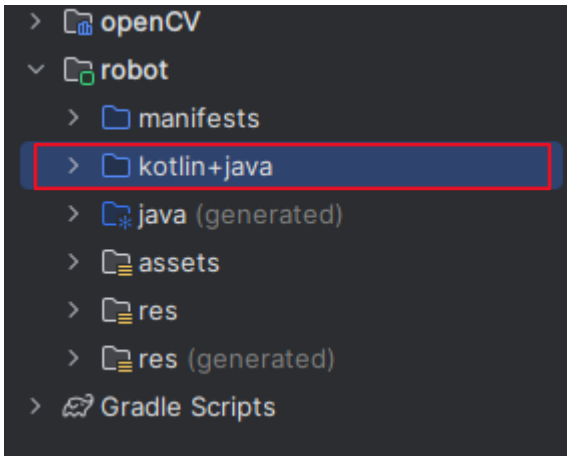
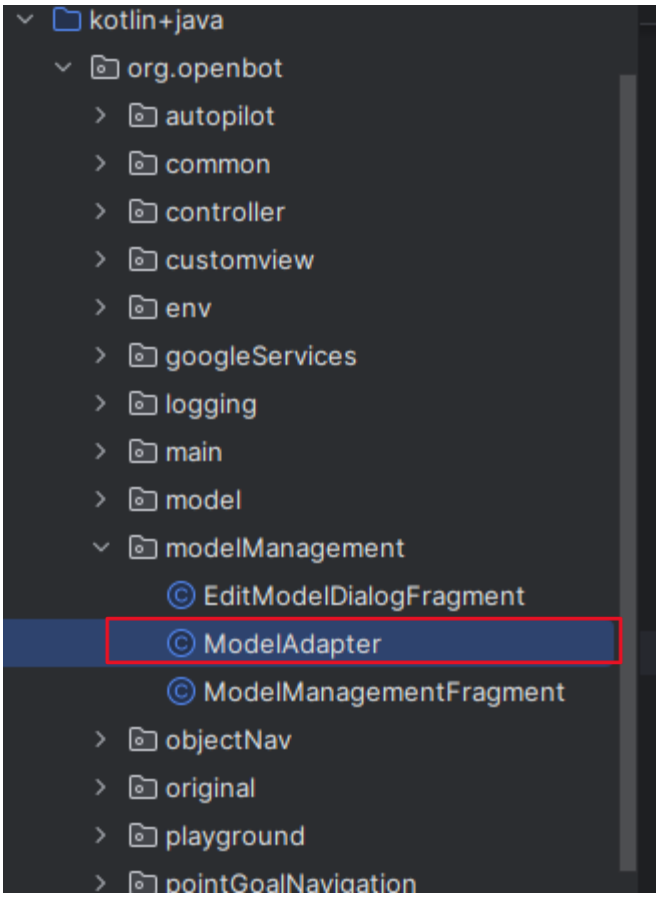
Documentación Modelos en proyecto Android Studio

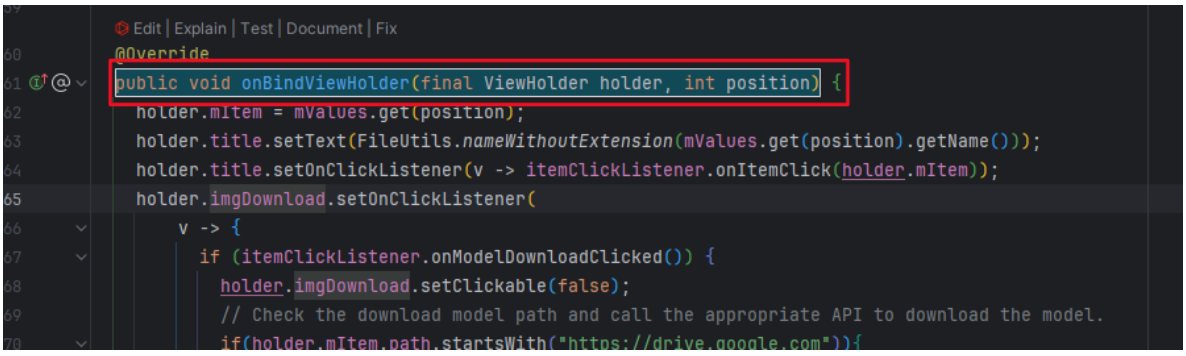
Para reconocer en el proyecto donde se encuentran los modelos de inteligencia artificial, nos debemos posicionar en la carpeta "*kotlin+java*" posteriormente ingresar a la carpeta "*modelManagement"* y dar click en la clase *"ModelAdapter"*.





En esta clase encontraremos el código donde proporciona la manera en como se recuperan los modelos de inteligencia artificial que son utilizados en el proyecto. Para dar una idea, los modelos son recuperados desde una carpeta que se encuentra almacenada en una carpeta de drive.

Las líneas importantes de esta clase para poderse dar una idea y poder realizar cambios en un futuro son los métodos los cuales hacen el redireccionamiento al llenado y descarga de los modelos disponibles.



Dentro de este método se configura el botón de descarga (imgDownload), así que **aquí se decide cómo y cuándo se recupera el modelo**.

```
holder.imgDownload.setOnClickListener(
    v -> {
      if (itemClickListener.onModelDownloadClicked()) {
        holder.imgDownload.setClickable(false);
        // Check the download model path and call the appropriate API to download the model.
        if(holder.mItem.path.startsWith("https://drive.google.com")){
          downloadFromDrive(holder);
        } else {
          Ion.with(holder.itemView.getContext()) LoadBuilder<B>
                  .load(holder.mItem.path) B
                  .progress(
                          (downloaded, total) -> {
                            System.out.println("" + downloaded + " / " + total);
                            holder.progressBar.setProgress((int) (downloaded * 100 / total));
                          })
                  //              .write(new File("/sdcard/openbot/tf.tflite"))
                  .write(
                          new File(
                                  pathname: holder.itemView.getContext().getFilesDir()
                                          + File.separator
                                          + holder.mItem.name)) ResponseFuture<File>
                  .setCallback(
                          (e, file) -> {
```

Dentro de este método se encuentran este fragmento de código donde se detecta si el modelo se descarga desde Google Drive o desde una URL directa.

```
private void downloadFromDrive(ViewHolder holder) { 1 usage
    new DownloadFileTask(mContext, progress -> {
      holder.progressBar.setProgress(progress);
      // Update your progress bar or do any other UI updates based on progress.
    }, file -> {
      // Handle the downloaded file here, for example, write its content to a new file.
      try {
        File newFile = new File(
                pathname: holder.itemView.getContext().getFilesDir()
                + File.separator
                + holder.mItem.name);
        InputStream inputStream = new FileInputStream(file);
        OutputStream outputStream = new FileOutputStream(newFile);
        byte[] buffer = new byte[4096];
        int bytesRead;
```

Este método descarga el archivo desde Drive a un archivo temporal, posteriormente lo copia a la ruta interna de la app.

Ahora hace uso de otra clase la cual se encuentra en la carpeta "*projects*" donde esta hace el proceso de descarga el archivo de los modelos.