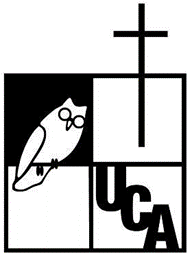
**Documentación técnica.**

**TREE C.C**

**Aplicaciones de Código Abierto**

**Ciclo 01/22**



**Grupo 9**

**Integrantes:**

-Oscar Isaí Aldana Cruz 00153818

-Miguel Ernesto Rivas Serrano 00087518

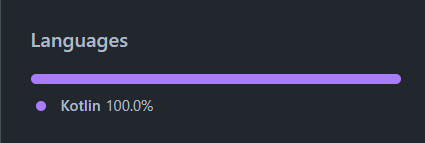
-Bryan Alexis Orellana Cabrera 00049318

-Jesús Antonio Alvarado González 00144318

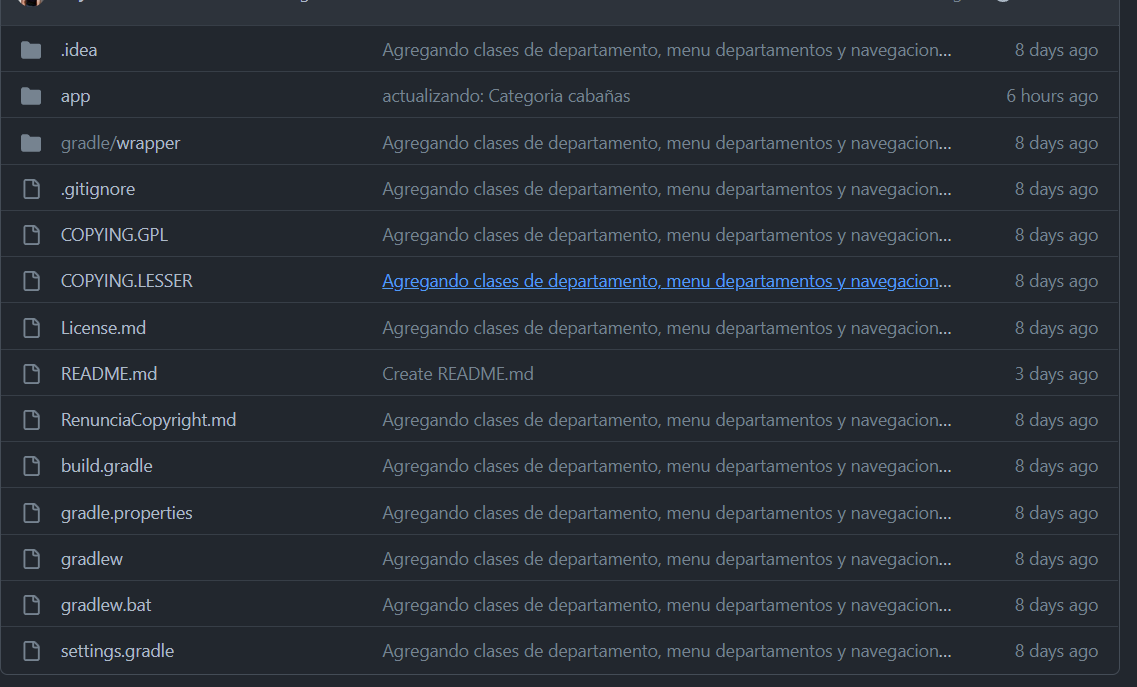
-Gabriel Enrique Gonzalez Rodriguez 00199518

Este documento se ha elaborado con la finalidad de llevar a cabo un registro detallado de todos los documentos técnicos de la aplicación y que se han modificado a lo largo de su desarrollo.

La aplicación utiliza en su totalidad el lenguaje de programación Kotlin.

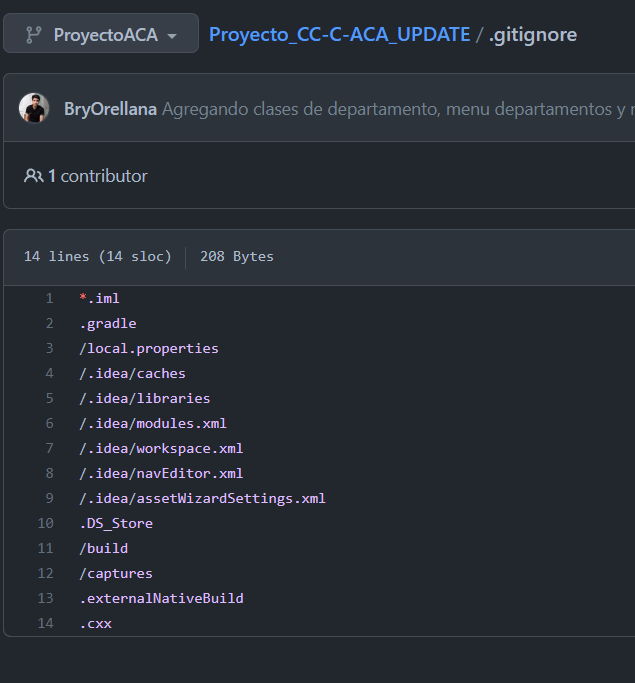


En el repositorio se encuentran alojados todos los archivos necesarios para la correcta ejecución del programa.



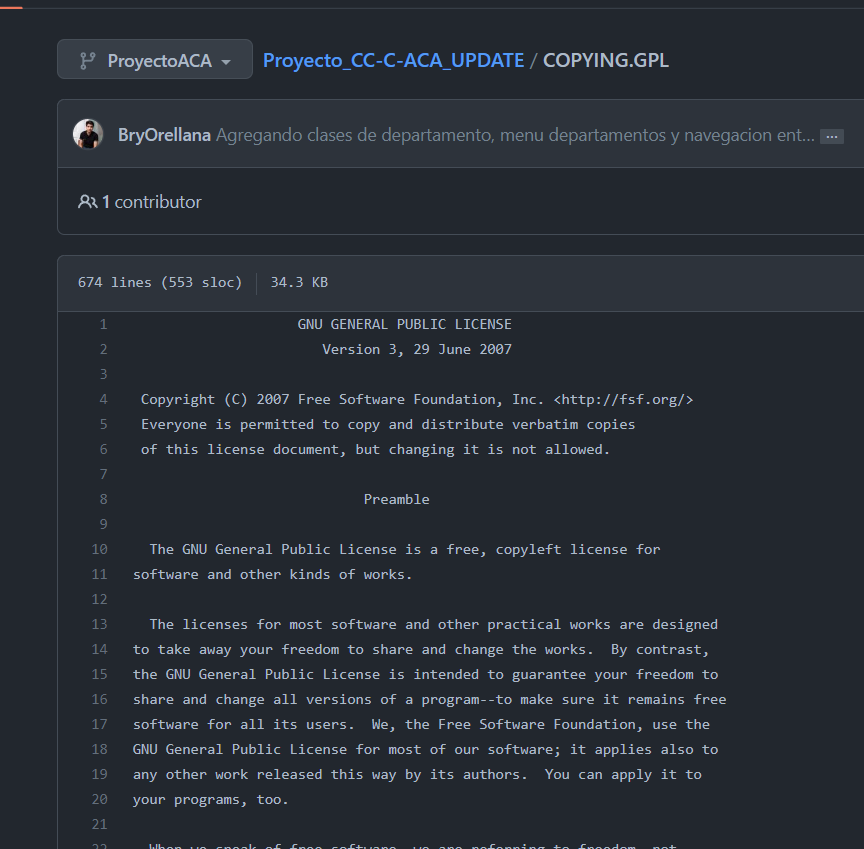
**.gitignore**

Archivo utilizado para agregar carpetas o archivos innecesarios para el repositorio.



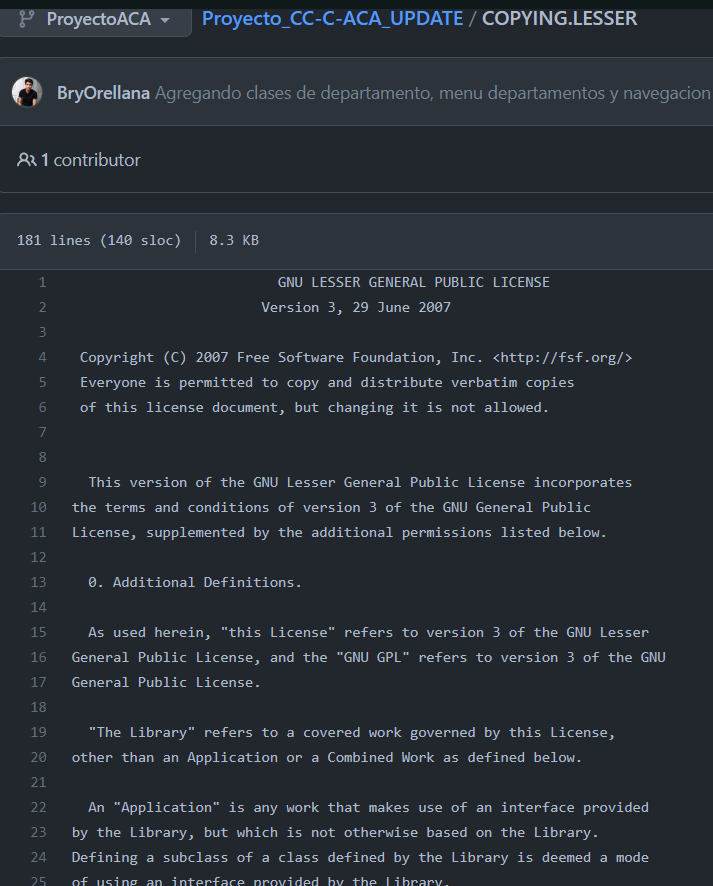
**COPYING.GPL**

Archivo donde se manifiesta la licencia que se ha utilizado para el proyecto.



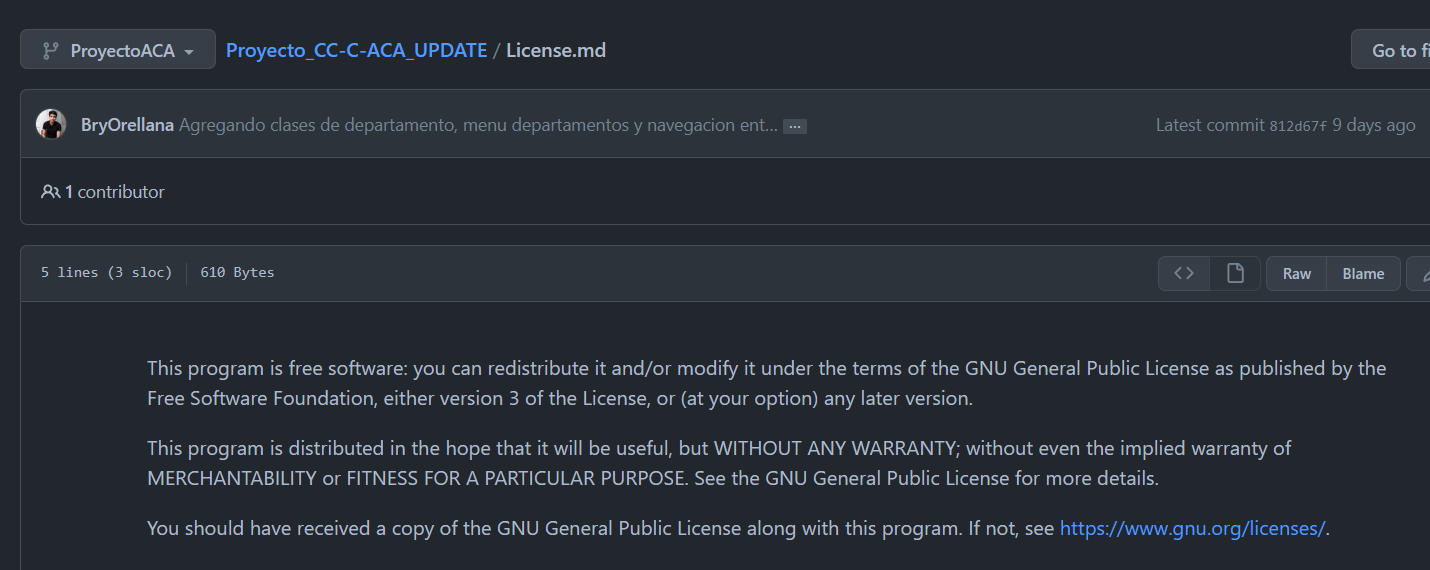
**COPYING.LESSER**

Similar al archivo anterior, se encuentran ciertas definiciones adicionales relacionados con la licencia de uso.



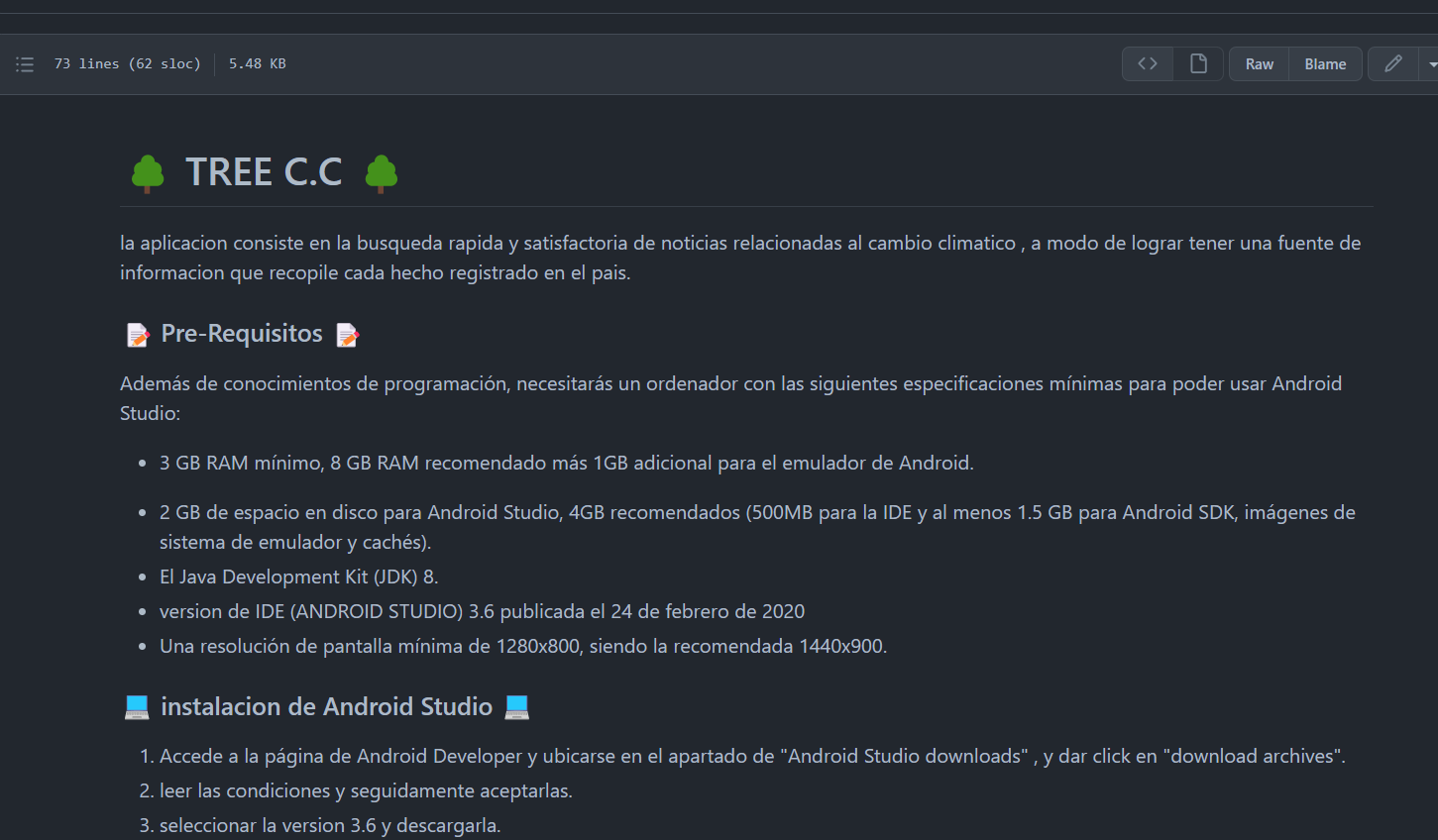
**License.md**

Archivo de licencia donde se refleja que el software creado es un software libre y que por lo tanto se puede copiar y redistribuir al gusto.



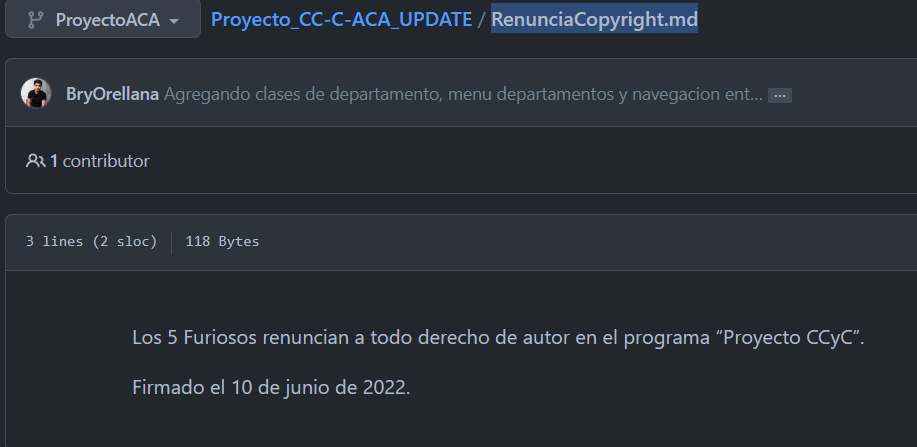
**Readme.md**

Archivo readme donde se manifiestan muchos de los requerimientos y manera de correr correctamente el código fuente.



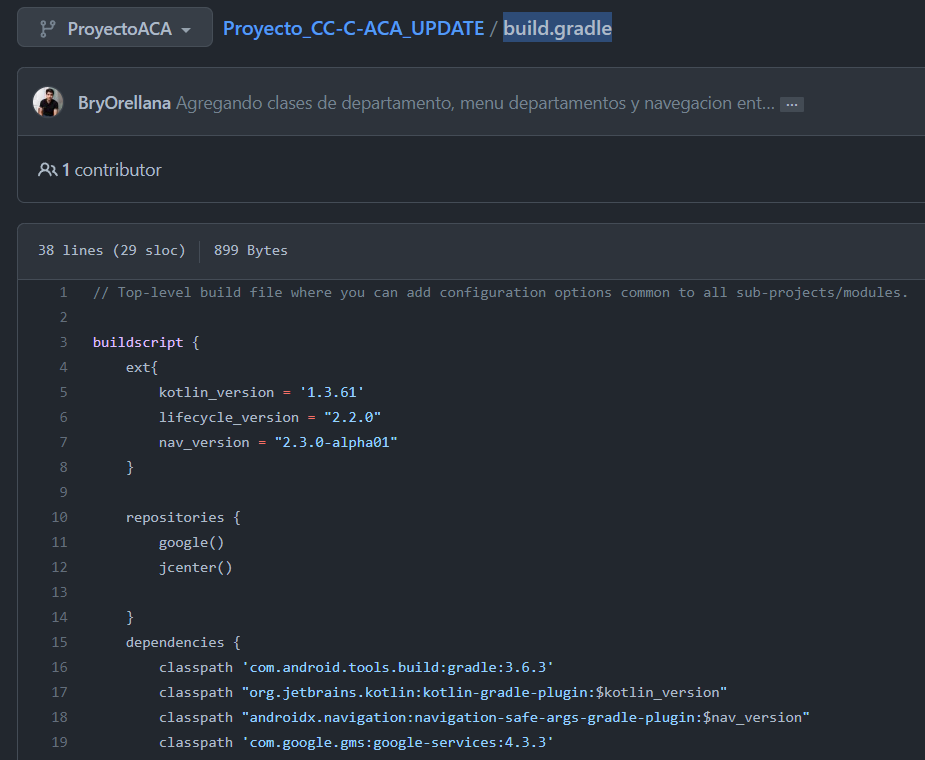
**RenunciaCopyright.md**

Archivo mediante el cual se hace oficial la renuncia por parte de nuestro equipo a cualquier derecho de autor en este código.



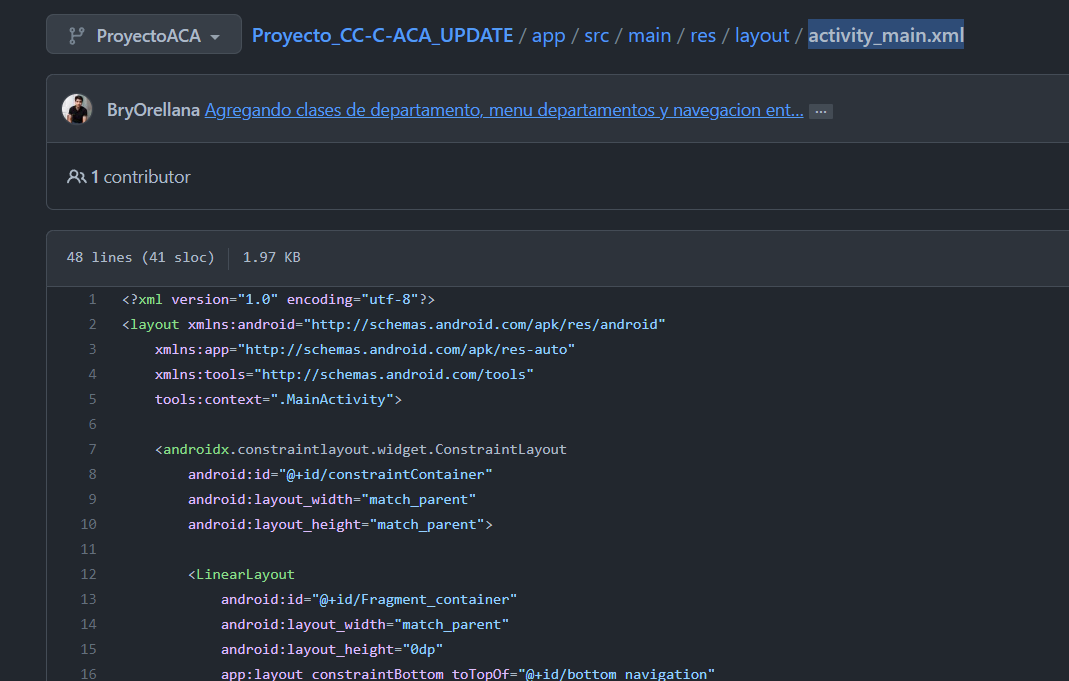
**build.gradle**

Archivo donde se muestran todos los modulos utilizados para el proyecto.



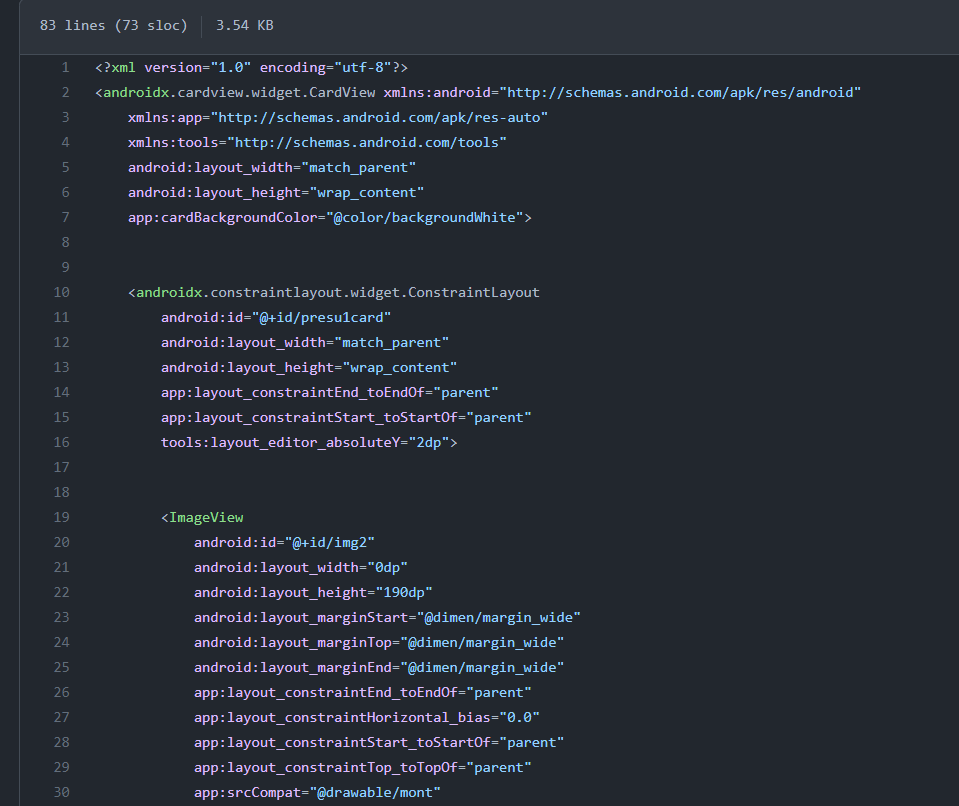
**activity\_main.xml**

Pantalla principal de la aplicación, en ella se ejecutan los demás fragmentos para evitar recargar la aplicación.



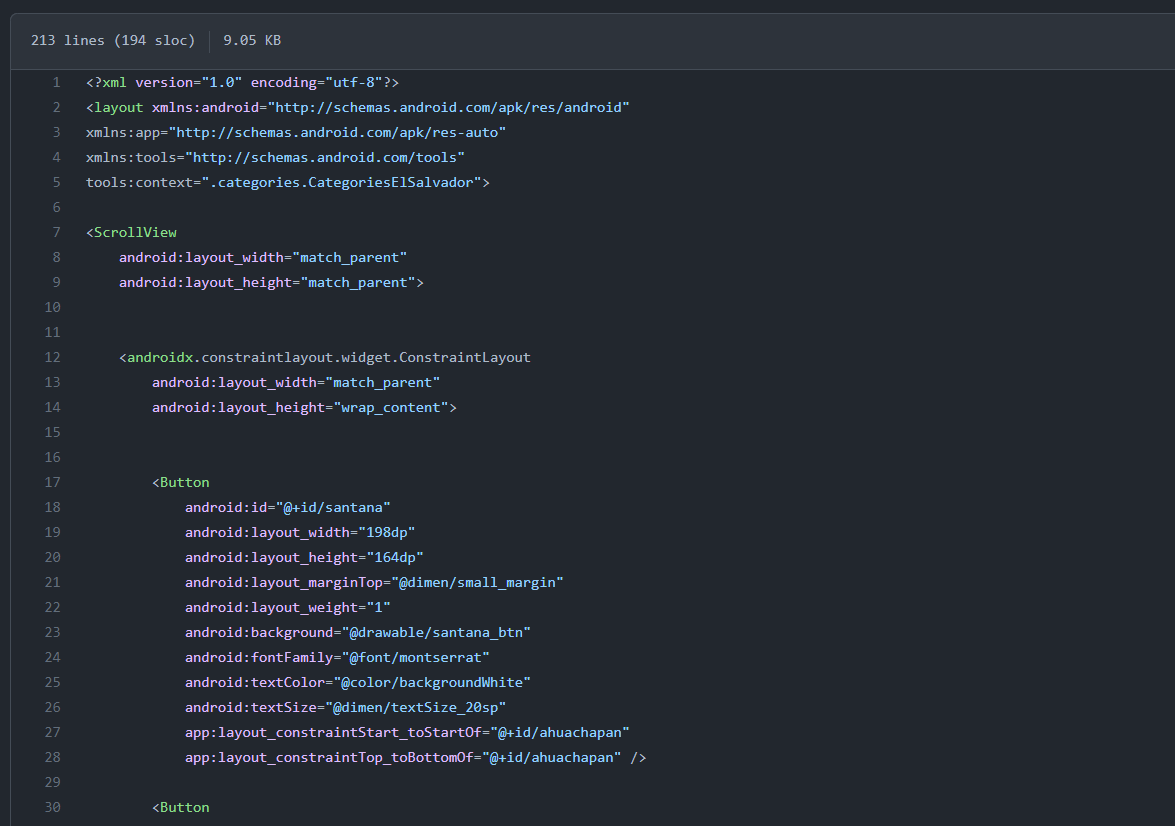
**budget\_list.xml**

Esta vista controla la manera en que cada tarjeta se muestra.



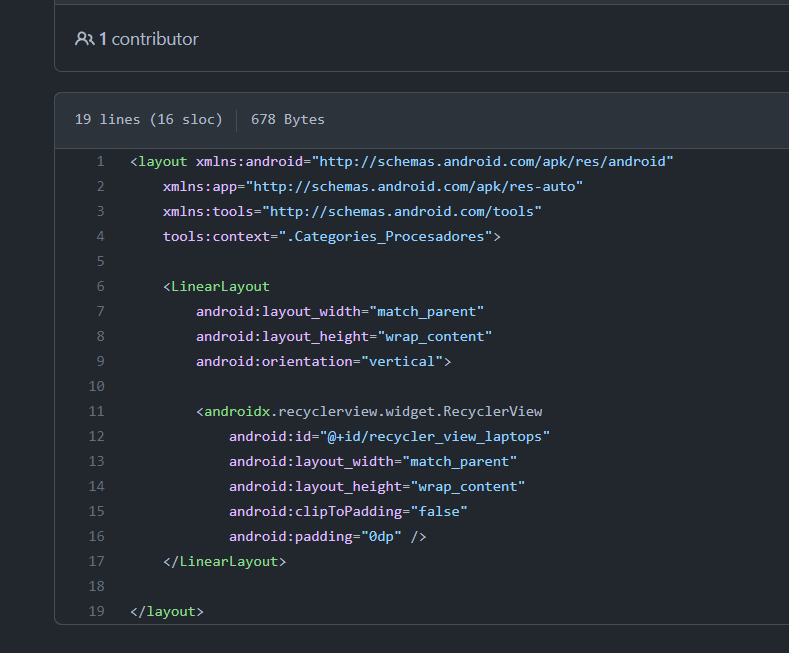
**fragment\_categories\_el\_salvador.xml**

Fragmento que controla la manera en que las categorías se muestran.



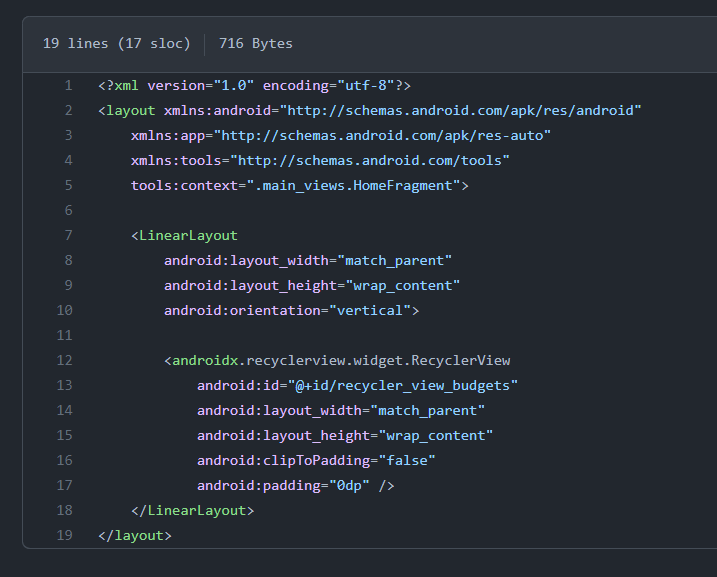
**fragment\_categories\_inside.xml**

Archivo donde se manejan las diferentes categorias con las que cuenta la aplicación.



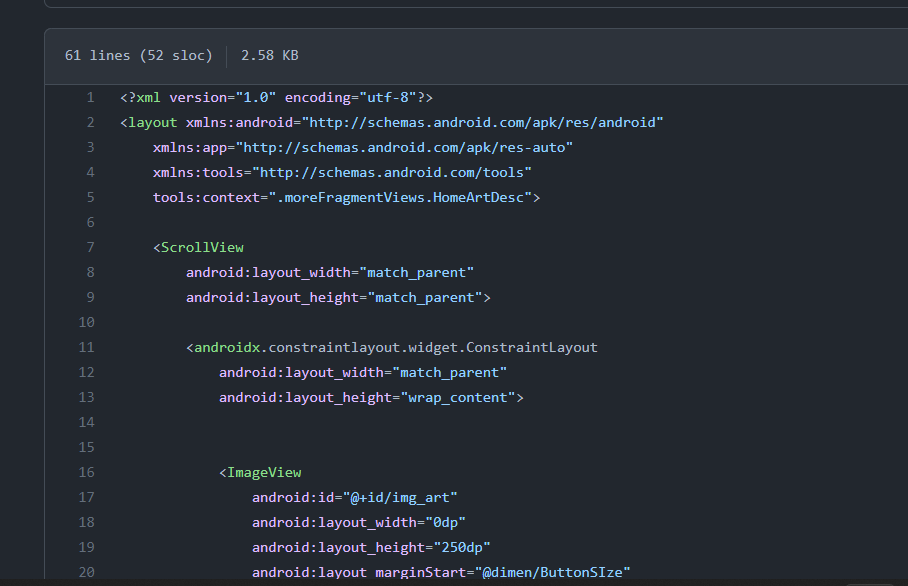
**fragment\_home.xml**

Fragmento principal, donde se van a desplegar las demás vistas de la aplicación.



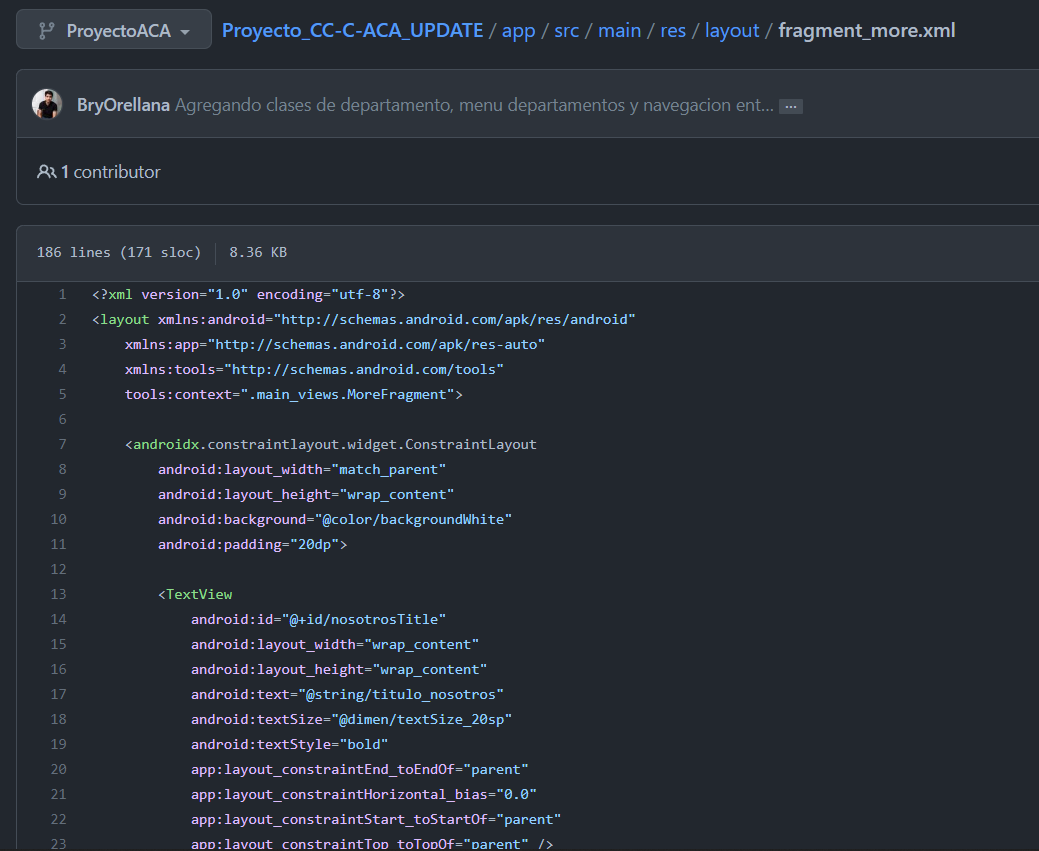
**fragment\_home\_art\_desc.xml**

Fragmento principal que se muestra al abrir la aplicación

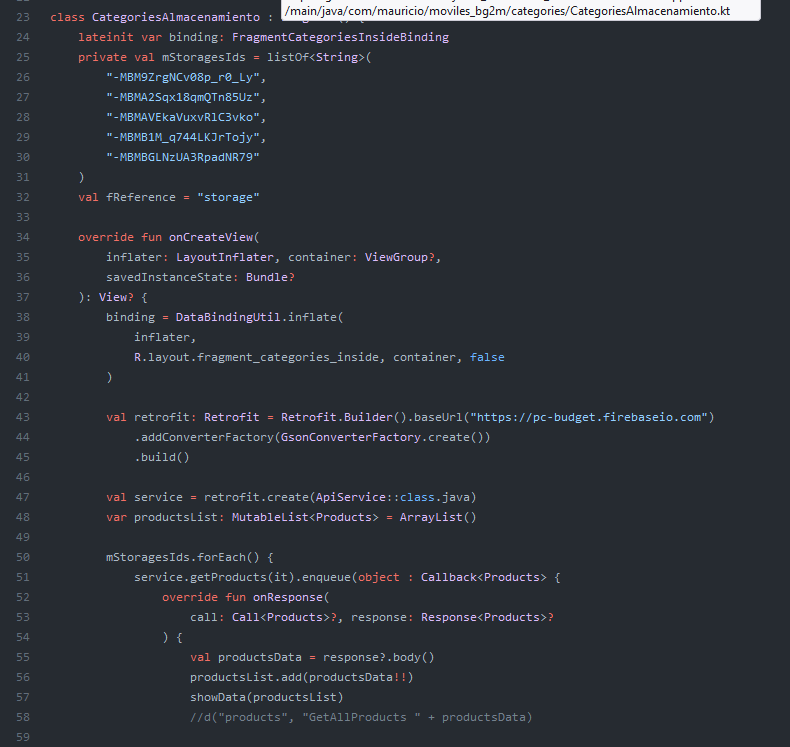


**fragment\_more.xml**

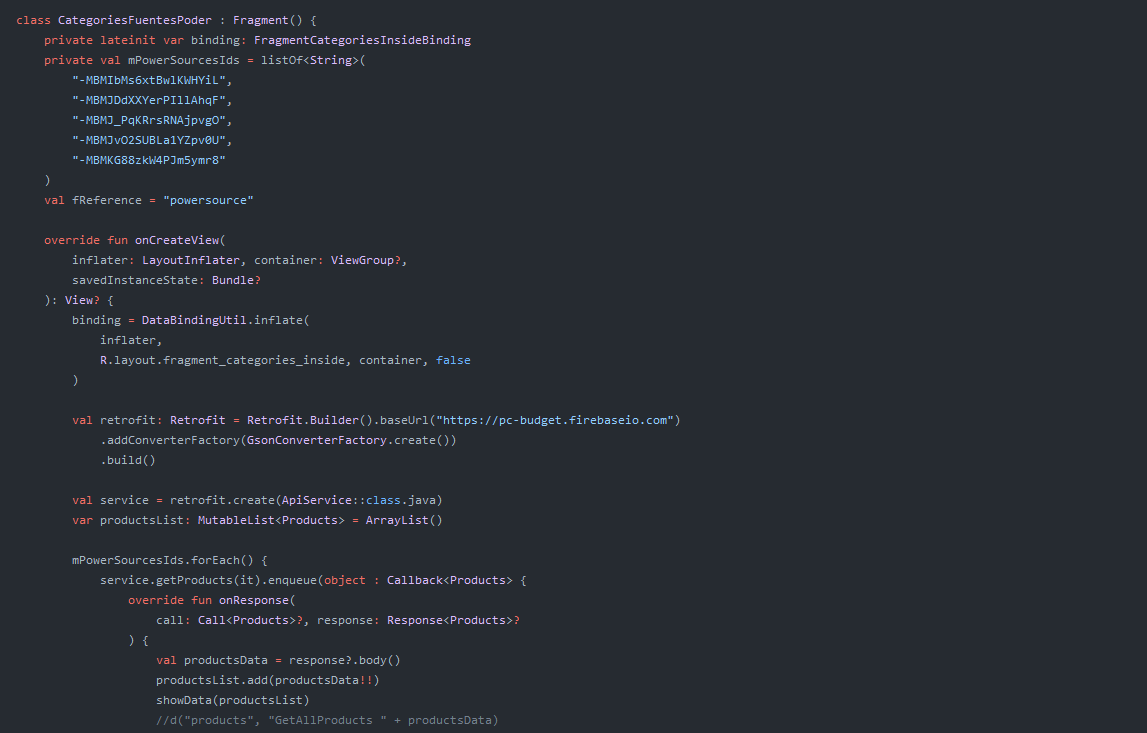
Fragmento donde se muestran todos los detalles extra de la aplicación como el acerca de, o las redes sociales de la aplicación.



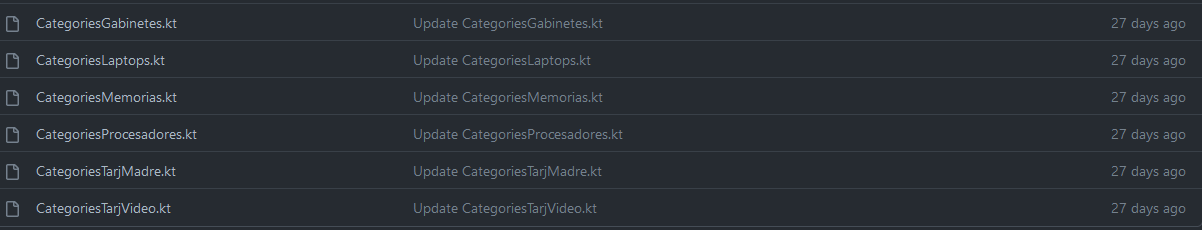
**CategoriesAlmacenamiento**: Esta clase controla la vista de clasificaciones de la app. En ella se traen los objetos almacenados en la base de datos y se muestran en pantalla.



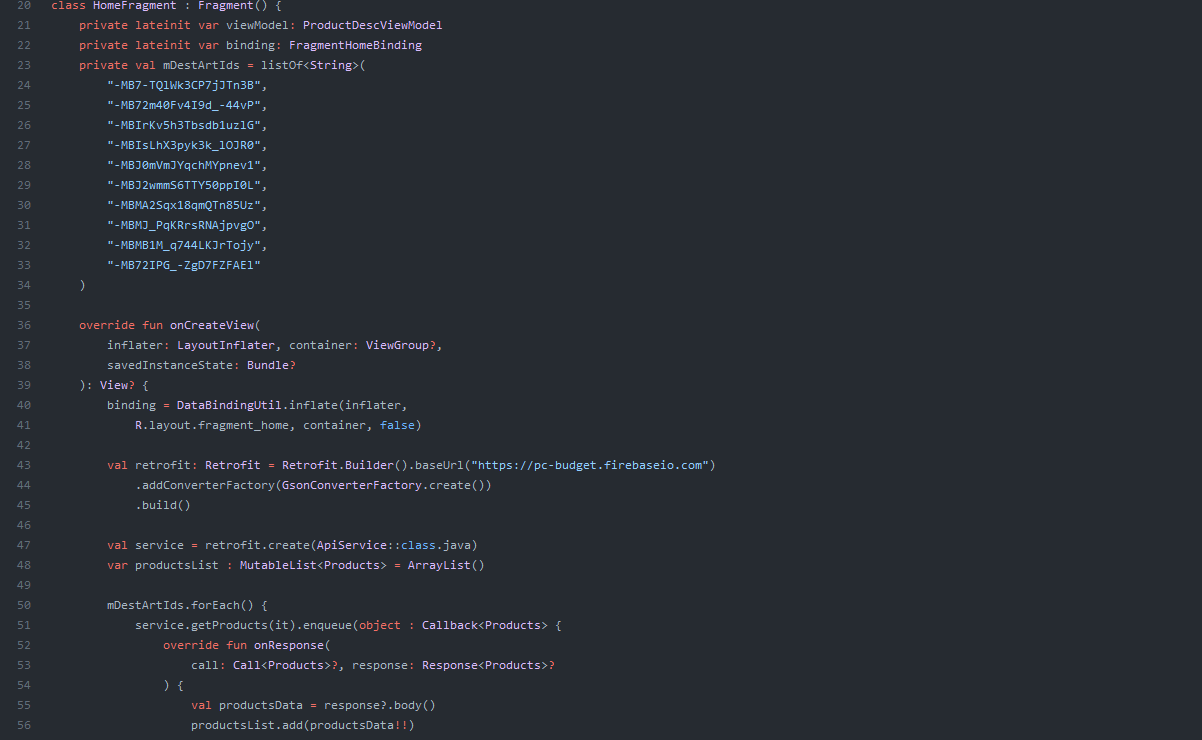
Se utiliza la librería de retrofit para mostrar las imágenes en la base de datos.



El resto de las clases de la carpeta de categorías hacen la misma funcionalidad para diferentes categorías.



**HomeFragment:** Contiene el código para desplegar la pantalla de inicio de la aplicación. Toma de la base de datos la información necesaria para construir las tarjetas donde se muestran los resumenes.



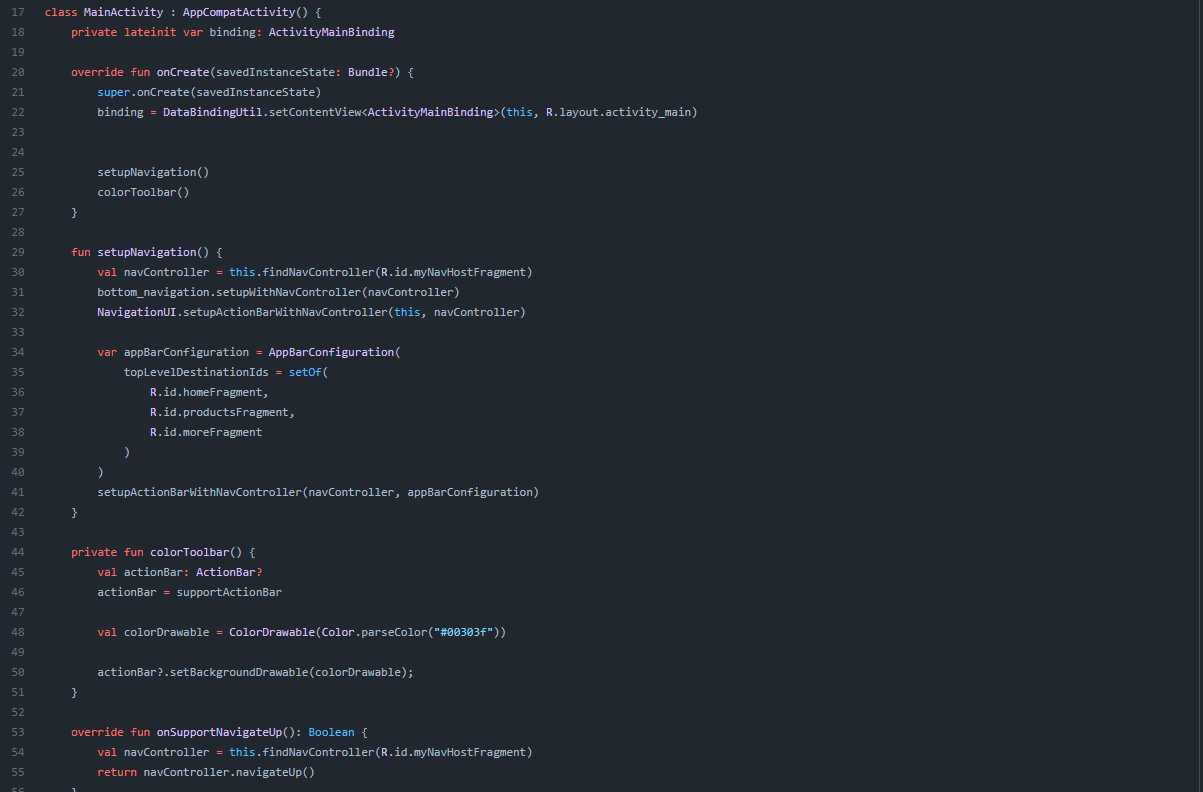
MoreFragment.kt hace lo mismo para la pantalla de Mas y ProductsFragment.kt para la de categorías.

**ApiService:** Ejecuta los métodos que llaman a la API para obtener un articulo en especifico.

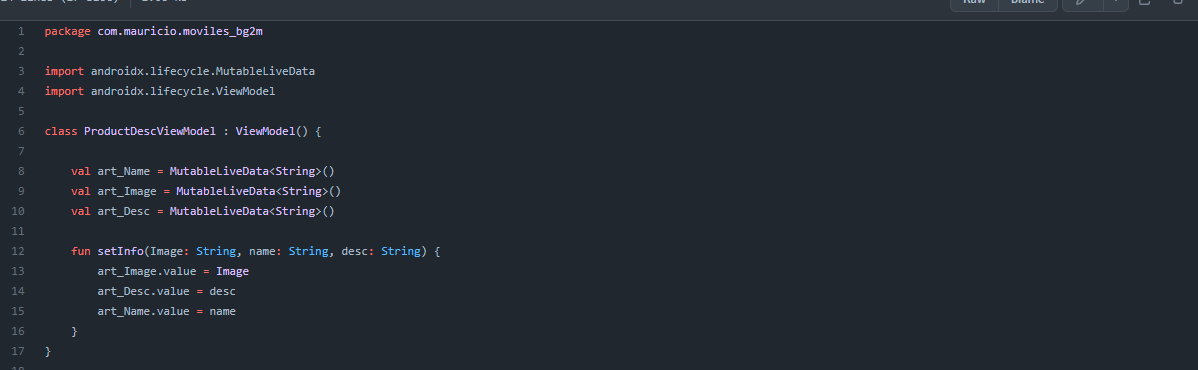


**MainActivity:**

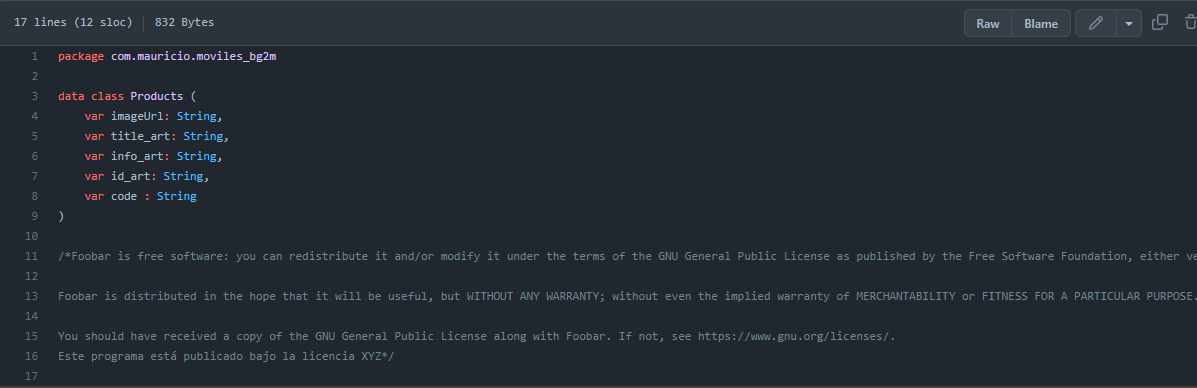
Crea la actividad principal al inicio de la aplicación

****

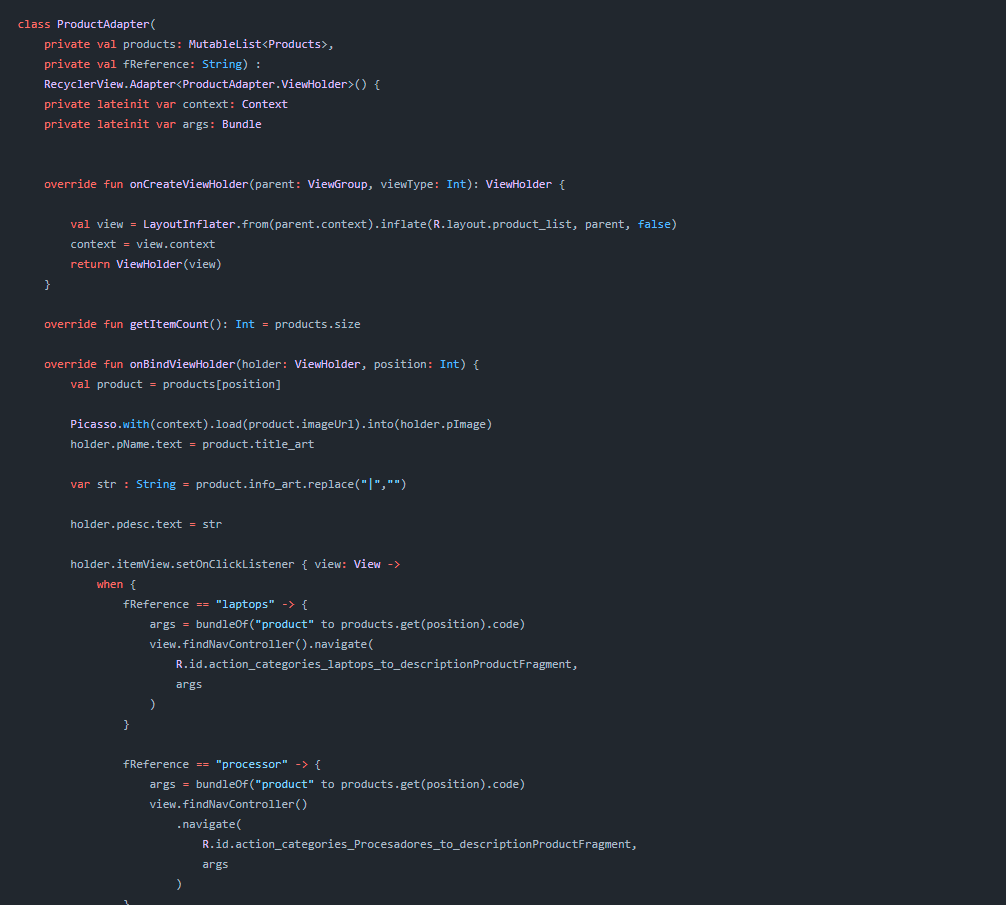
**ProductoDescViewer:** Contiene el modelo básico para resumir un artículo en una tarjeta compacta.

****

**Products:** Objeto instanciable que contiene un artículo.



**ProductoAdapter:** Clase que instancia un artículo y lo popula.

****