**Aplicación:** Mamma Mia Recetas

**Alumno:** Alejandro Díez Redondo

**Fecha:** Mayo de 2023

**Resumen:**

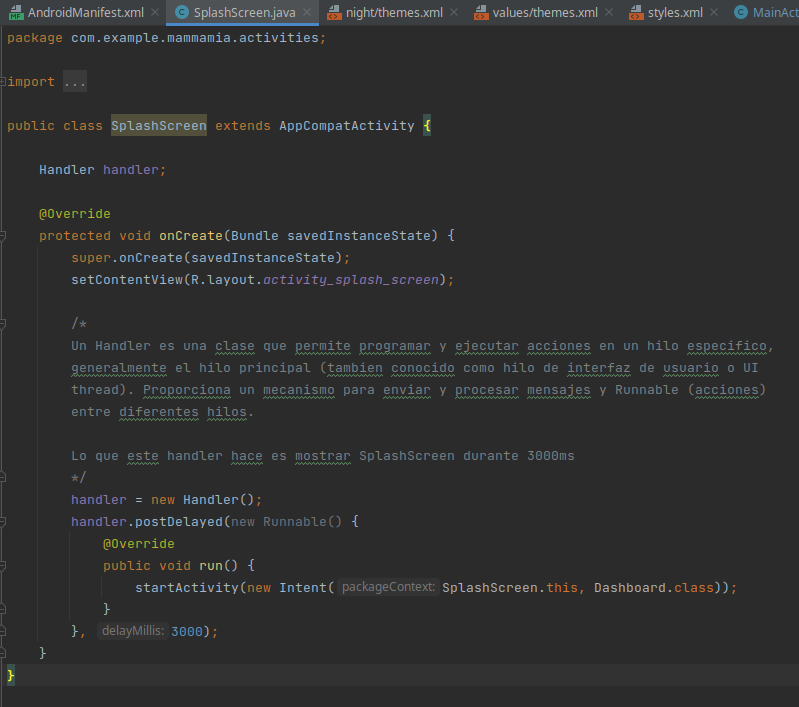
La aplicación Mamma Mia Recetas consiste en una colección de recetas para 4 platos: aperitivos, primeros platos, segundos platos y postres.

**Funcionamiento:**

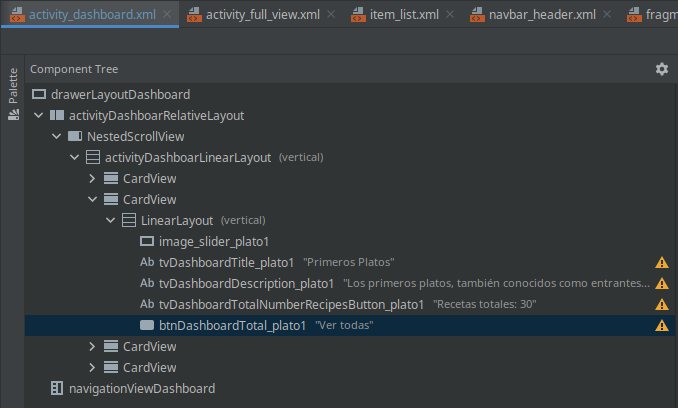
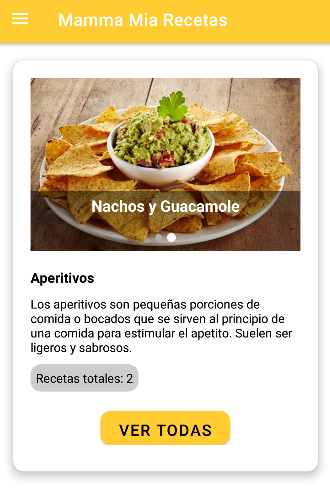
***SplashScreen:***

Comienza con un SplashScreen, una pantalla de 3 segundos en la que se muestra el logo y el nombre de la aplicación a través de un Handler:

****

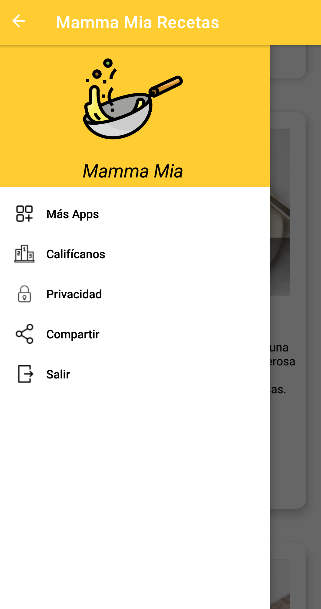
****

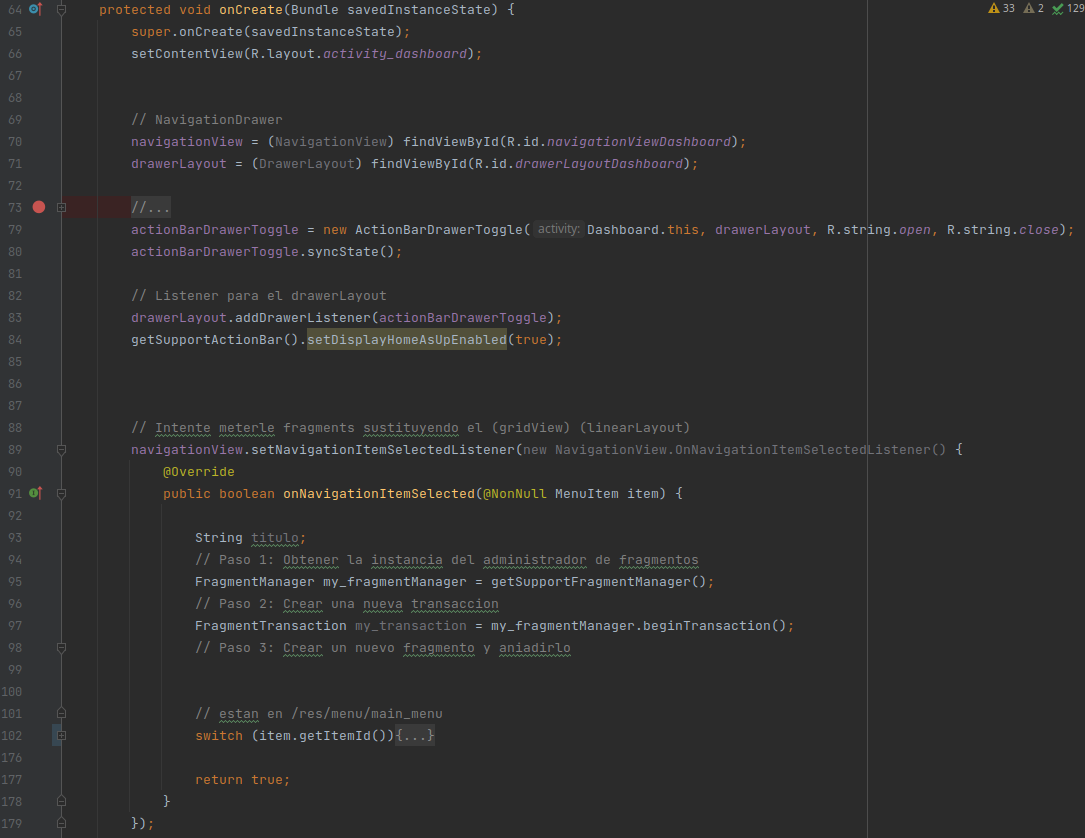
***Dashboard:***  
Después de esos 3 segundos se inicia desde aquí la actividad del Dashboard.java, el panel donde encontramos los 4 platos en un NestedScrollView (activity\_dashboard.xml), cada uno dentro de su correspondiente CardView.

****

Cada CardView cuenta con una breve descripción del tipo de plato, el número de recetas de ese tipo de plato disponibles y un botón que nos lleva al activity\_main.xml.

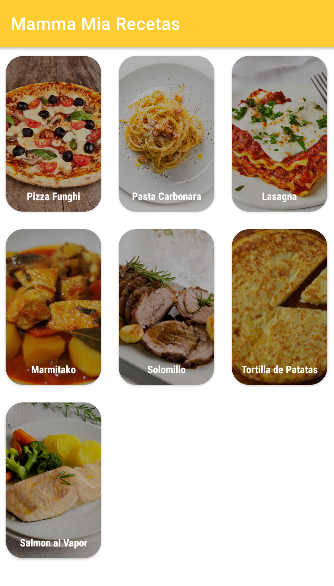
Dashboard cuenta con un NavigationDrawer, el menú lateral de la hamburguesita, donde se pueden realizar distintas acciones gracias a un switch:

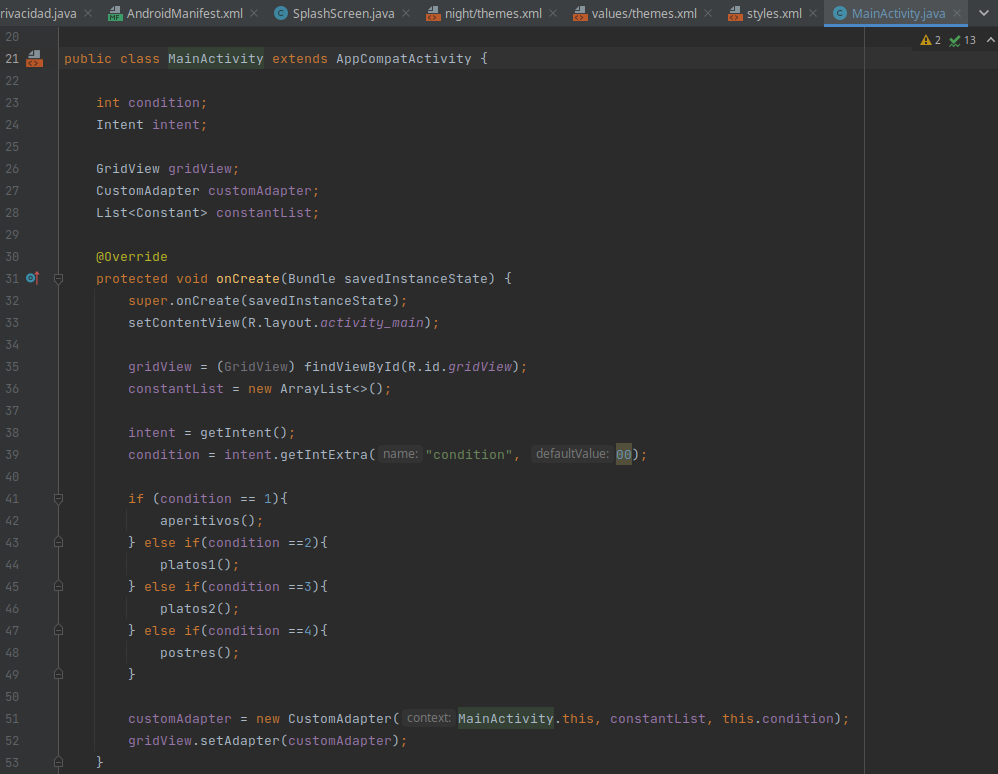
****

****

***MainActivity:***

Al pulsar el botón "Ver Todas", dependiendo de en qué CardView nos encontremos, nos lleva al activity\_main.xml (MainActivity.java), que contiene los items, (las distintas recetas), dentro de un gridView:

****

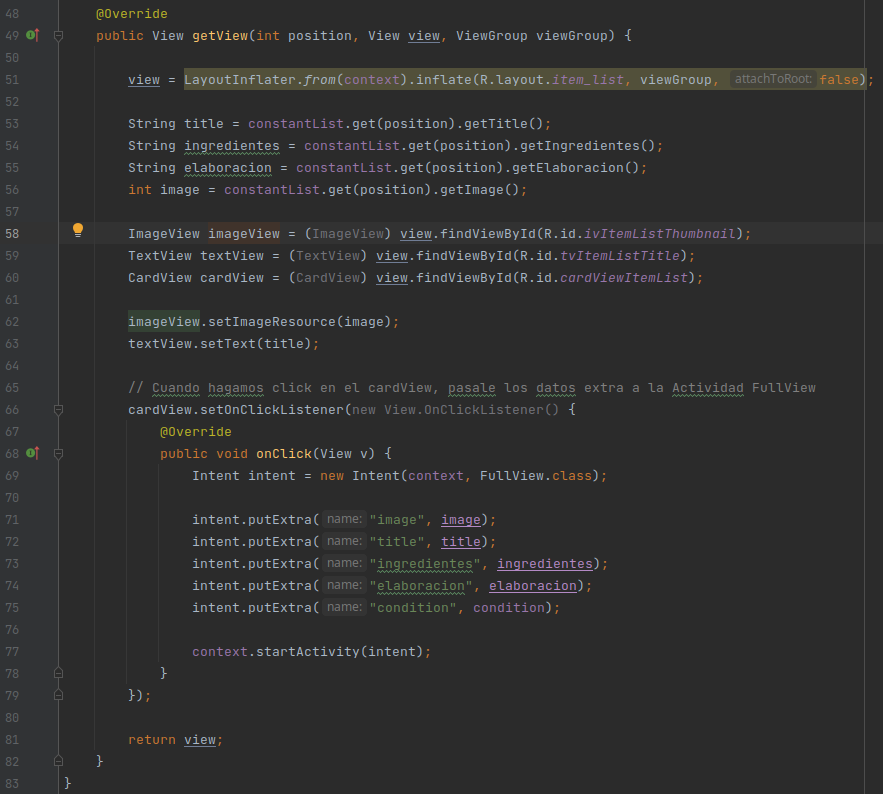
****

Dependiendo de la condición se añaden los platos a la constantList y se pasan al adaptador.

***CustomAdapter:***

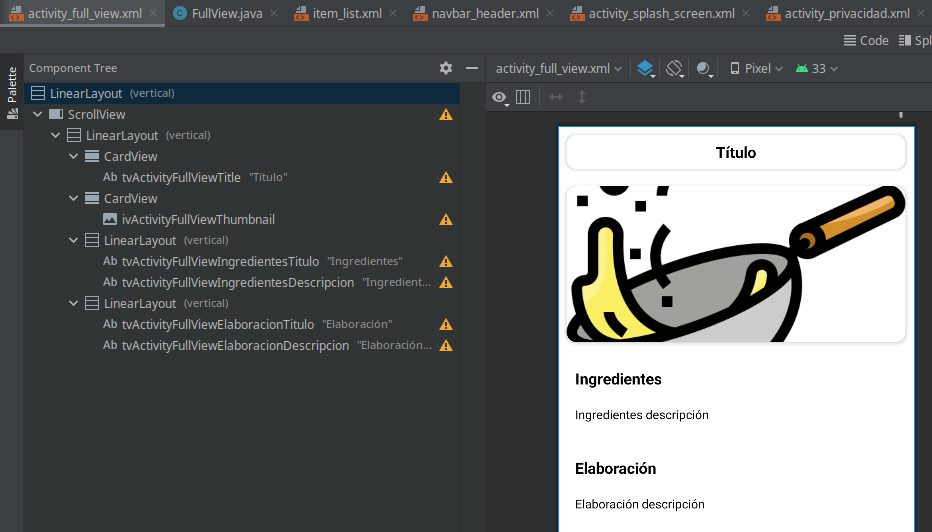
Un adaptador customizado para mostrar una lista de elementos en la interfaz de usuario.

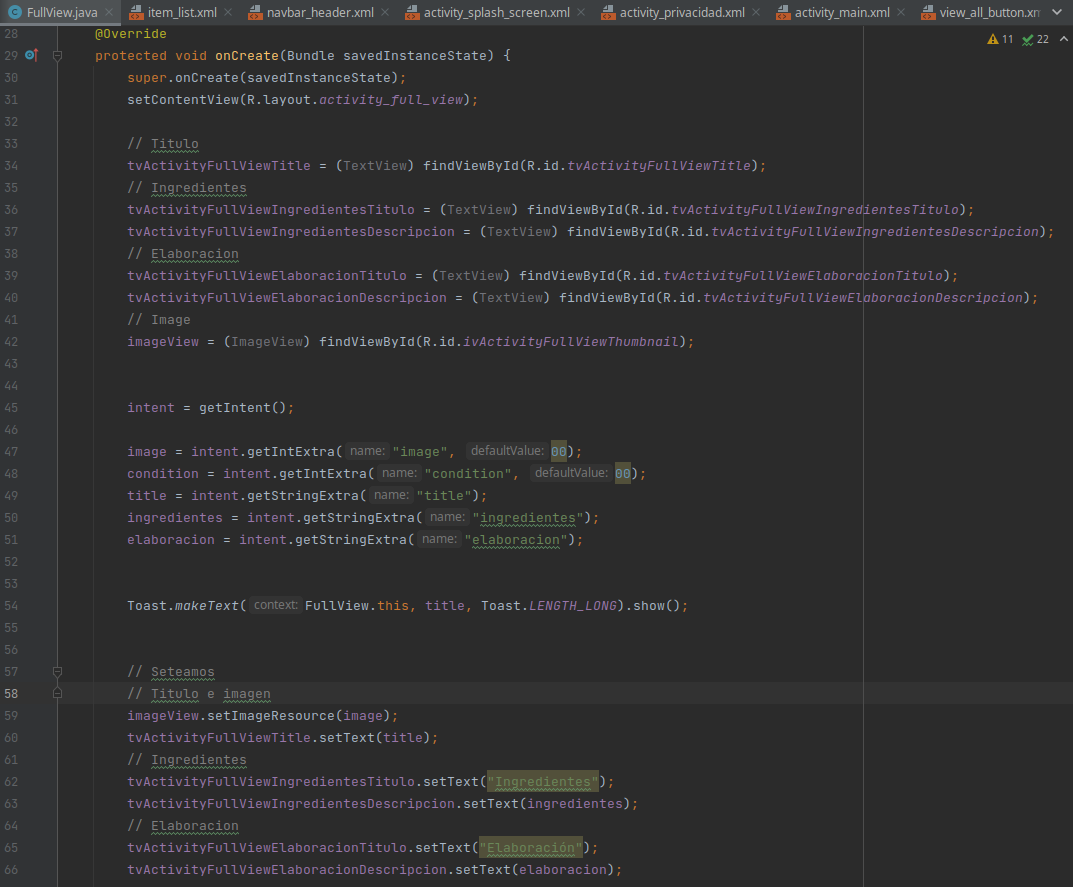
El método getView() se encarga de crear y retornar la vista de cada elemento de la lista. Se utiliza el inflador LayoutInflater para inflar el diseño de la vista de lista definido en el archivo XML item\_list.xml. Luego, se obtienen los datos específicos del elemento de la lista en la posición dada, como el título, los ingredientes, la elaboración y la imagen. Estos datos se establecen en los elementos de la vista, como un ImageView, un TextView y un CardView. Además, se establece un OnClickListener en el CardView para capturar el evento de clic y abrir una nueva actividad FullView cuando se hace clic en un elemento de la lista. Se pasan los datos adicionales a la nueva actividad mediante putExtra(). Por último, se devuelve la vista creada.



***FullView:***

Al pulsar en una de las recetas vamos al activitiy\_full\_view.xml (FullView.java), que nos abre la receta en versión extendida, donde se ven los ingredientes y la elaboración para 2 personas:

****

****

FullView.java coge los datos del intento y los setea.

****