**Registro de pruebas**

*Sergio Groppa y Marcos Vázquez*

10/05/2024

Índice

[Introducción 3](#_Toc167464097)

[Desarrollo 3](#_Toc167464098)

[Prueba 1: 3](#_Toc167464099)

[Prueba 2: 6](#_Toc167464100)

[Prueba 3: 8](#_Toc167464101)

[Prueba 4: 11](#_Toc167464102)

[Prueba 5: 15](#_Toc167464103)

[Prueba 6: 21](#_Toc167464104)

[Prueba 7: 24](#_Toc167464105)

[Bibliografía 27](#_Toc167464106)

## Introducción

En este documento redactamos todas las pruebas realizadas en entorno de desarrollo.

## Desarrollo

#### Prueba 1:

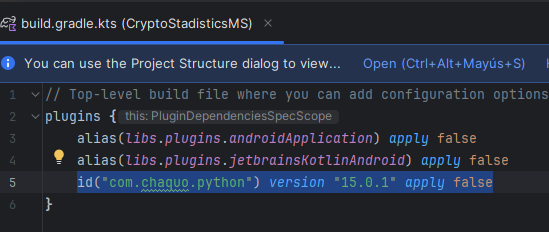
(02/04/2024)

Escenario inicial:

Creado el proyecto por defecto de Android studio, añadirle la capacidad de ejecutar código Python y realizar una prueba de compatibilidad entre Kotlin y Pyhon.

Proceso:

En el archivo build.grandle.kts (Module) añadirle la librería chaquo.

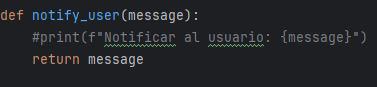


Después en build.grandle.kts(: app) añadir la librería chaquo y su librería y configuración.



En un archivo

Para realizar una prueba en el archivo Bot.py creamos un método.



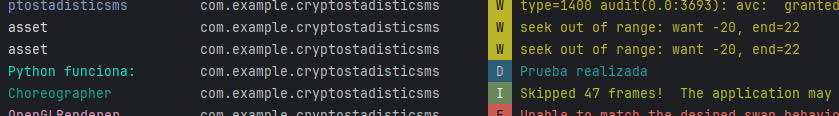
Y en la MainActivity ponemos el código para que realice el método de Python definido.



Resultado esperado:

Que se guarde el contenido correctamente y lo muestre en la consola.

Resultado final:

Que se guarde el contenido correctamente y lo muestre en la consola

#### Prueba 2:

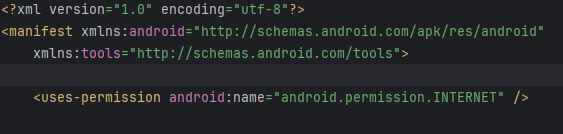
(05/04/2024)

Escenario inicial:

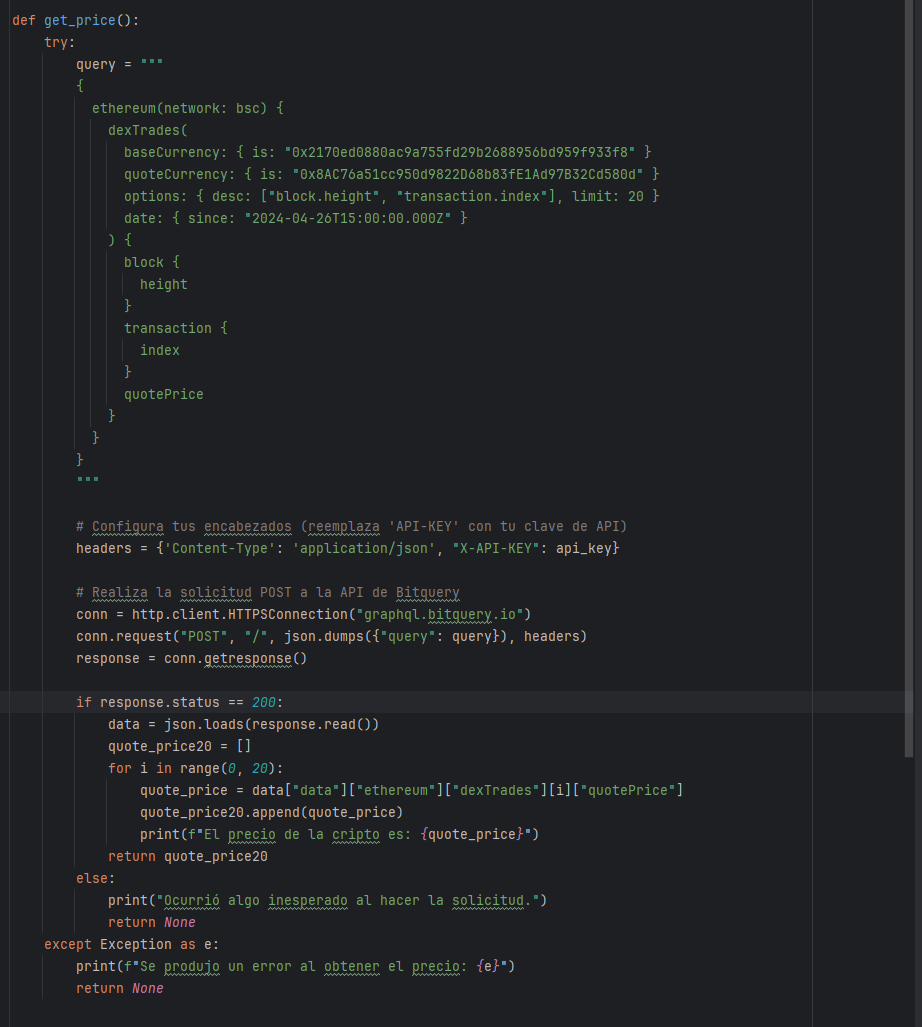
Creado el proyecto, realizar una conexión a la API de terceros (BitQuery).

Proceso:

Activar el permiso para acceder a internet, en AndroidManifest.xml:



Crear una cuenta en “Bitquery”, guardar la API key y la ruta inicial de la página web en el archivo Bot.py. Crear un archivo para las peticiones de la API.



Resultado esperado:

La conexión se realice correctamente.

Resultado final:

La conexión se realiza correctamente.

#### Prueba 3:

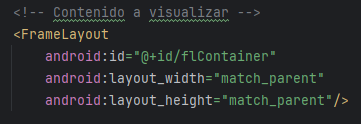
(12/04/2024)

Escenario inicial:

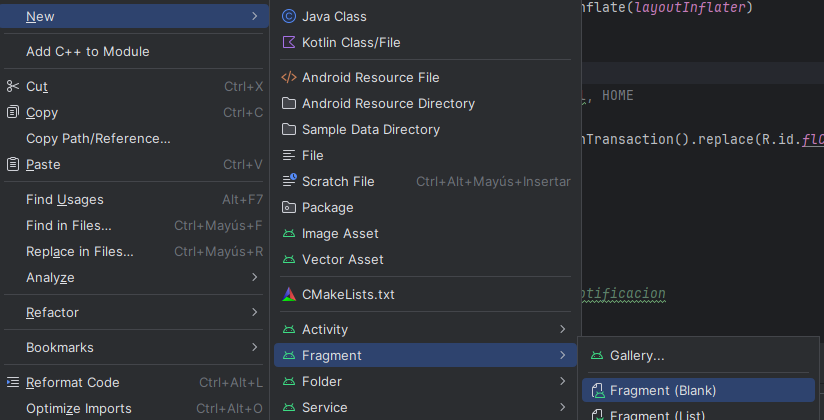
Crear Fragment y que se visualice para realizar cada apartada de la aplicación

Proceso:

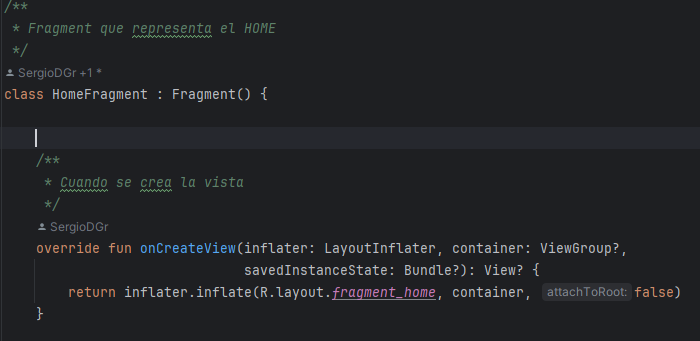
En layout main\_activity, añadirle un FrameLayout:



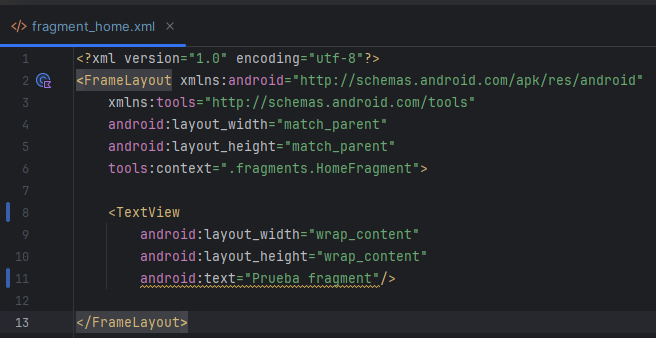
Creamos el Fragment Home.



En HomeFragment:



En fragment\_home:



Y para finalizar en MainActivy ponemos esto:

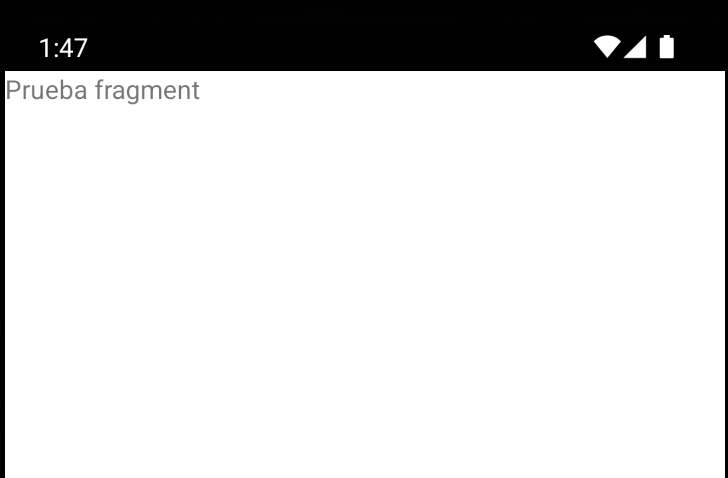


Resultado esperado:

Que se visualice el mensaje del Fragment.

Resultado final:

Que se visualice el mensaje del Fragment.



#### Prueba 4:

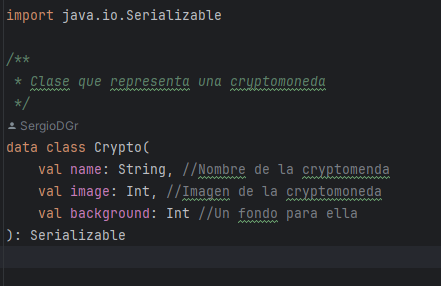
(15/04/2024)

Escenario inicial:

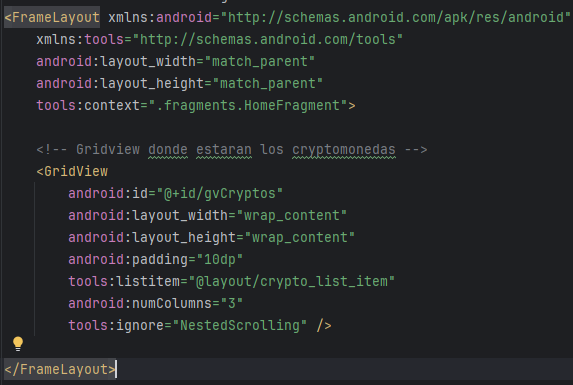
Realizar un adapter para un GridView que se visualice los elementos que defina.

Proceso:

Definimos los elementos que queremos visualizar:



En el layout se define GridView:



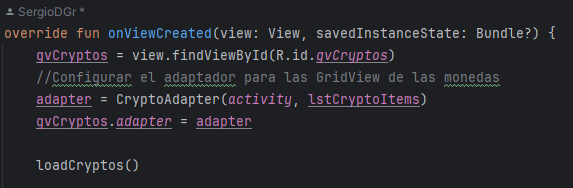
Creamos el layout para cada item que visualizara el contenido:



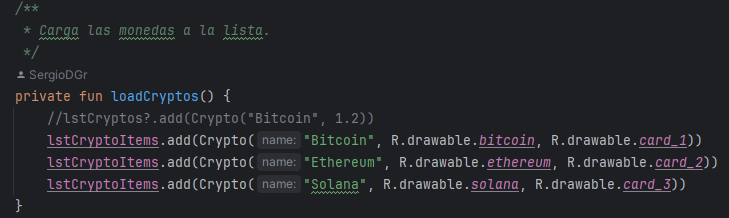
Creamos el adapter:



En el GridView le definimos el adapter:



y le cargamos el contenido:

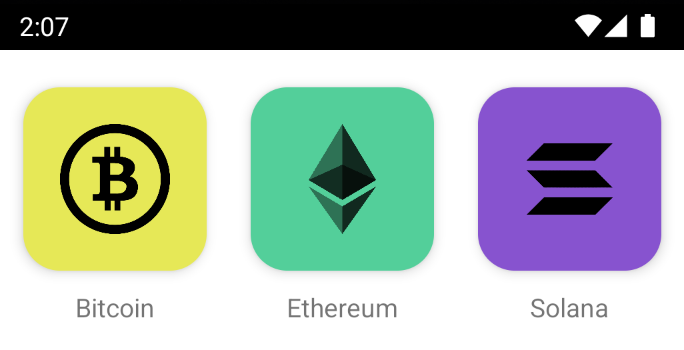


Resultado esperado:

Que visualice los elementos.

Resultado final:

Que visualice los elementos.



#### Prueba 5:

(23/04/2024)

Escenario inicial:

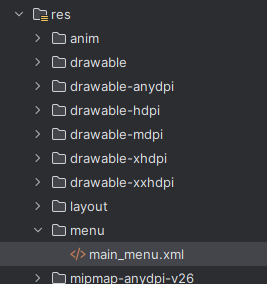
En la actividad principal al tener varios Fragments o contenido diferente para visualizarlo, es necesario crear un navegador

Proceso:

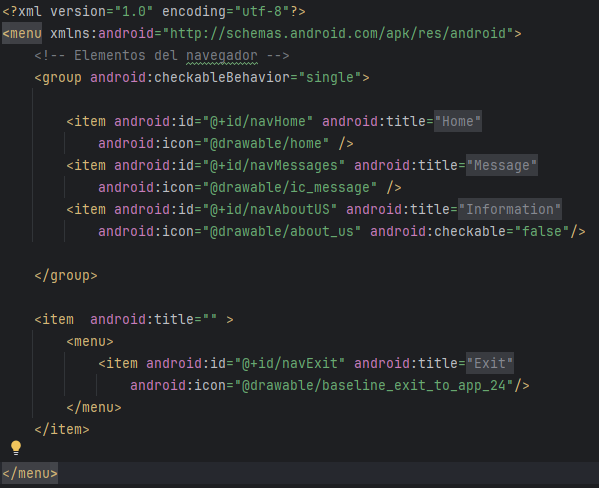
En MainActivity cambiamos la etiqueta principal DrawerLayout. Añadimos la etiqueta NavigationView y un icono para el para acceder al Navegador.



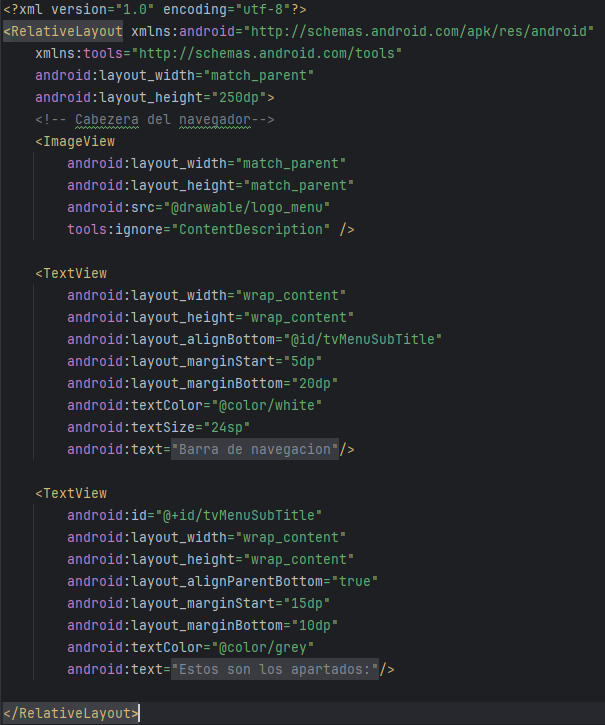
Creamos el layout que se vería el navegador pare ello primero creamos el archivo main\_menu.



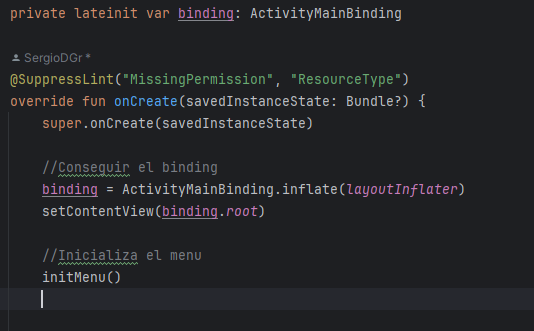
Y su contenido:



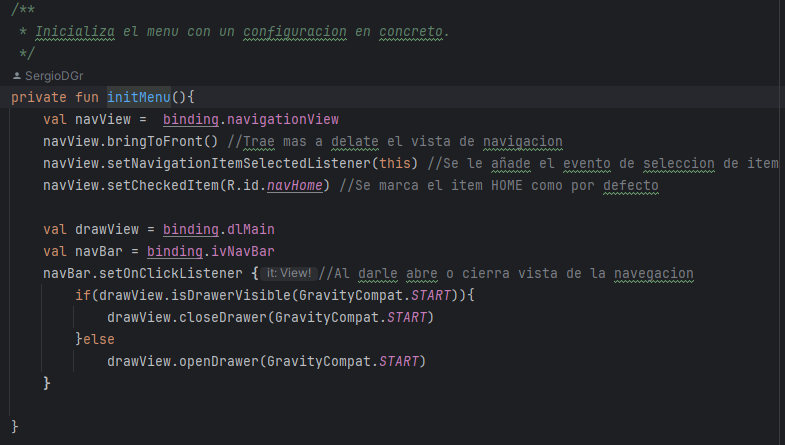
Y para añadirle una cabecera al navegador le creamos un layout:



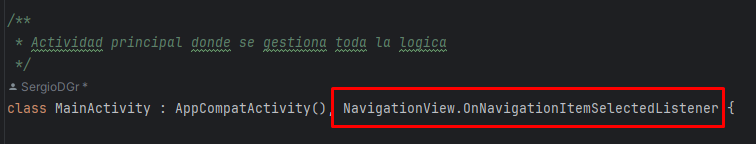
En MainActivity configuramos los componentes:



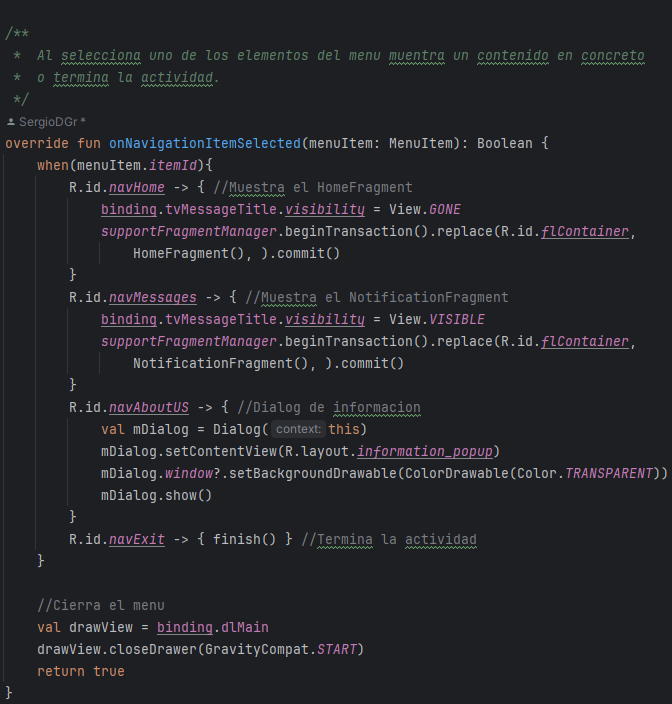
Para que visulize o cierra el menu.



Implementamos esta interfaz para la selección del menu.



Redefinimos el método:

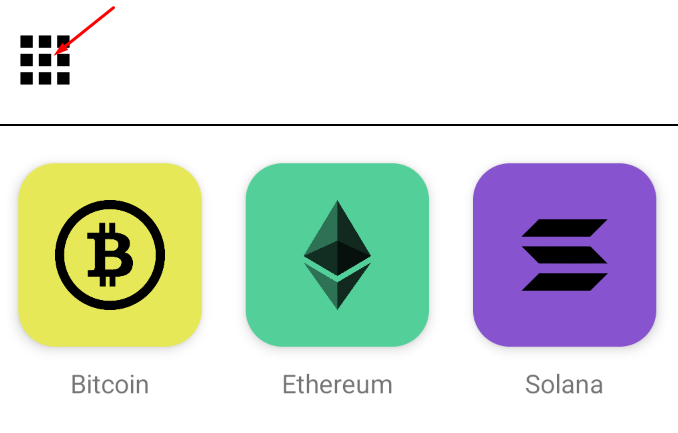


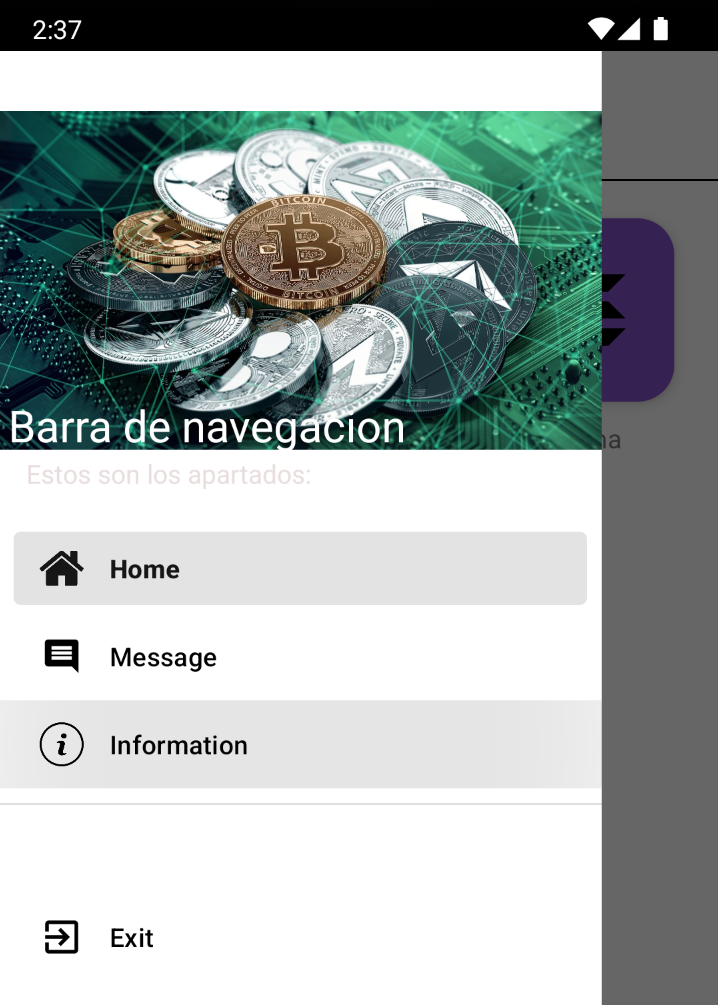
Resultado esperado:

Que visualice el menú y al selecciona un elemento haga su acción.

Resultado final:

Que visualice el menú y al selecciona un elemento haga su acción.







#### Prueba 6:

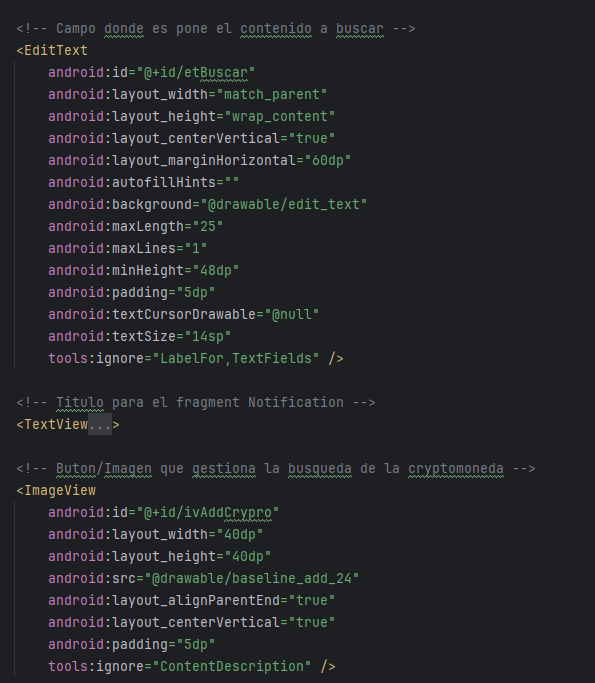
(05/05/2024)

Escenario inicial:

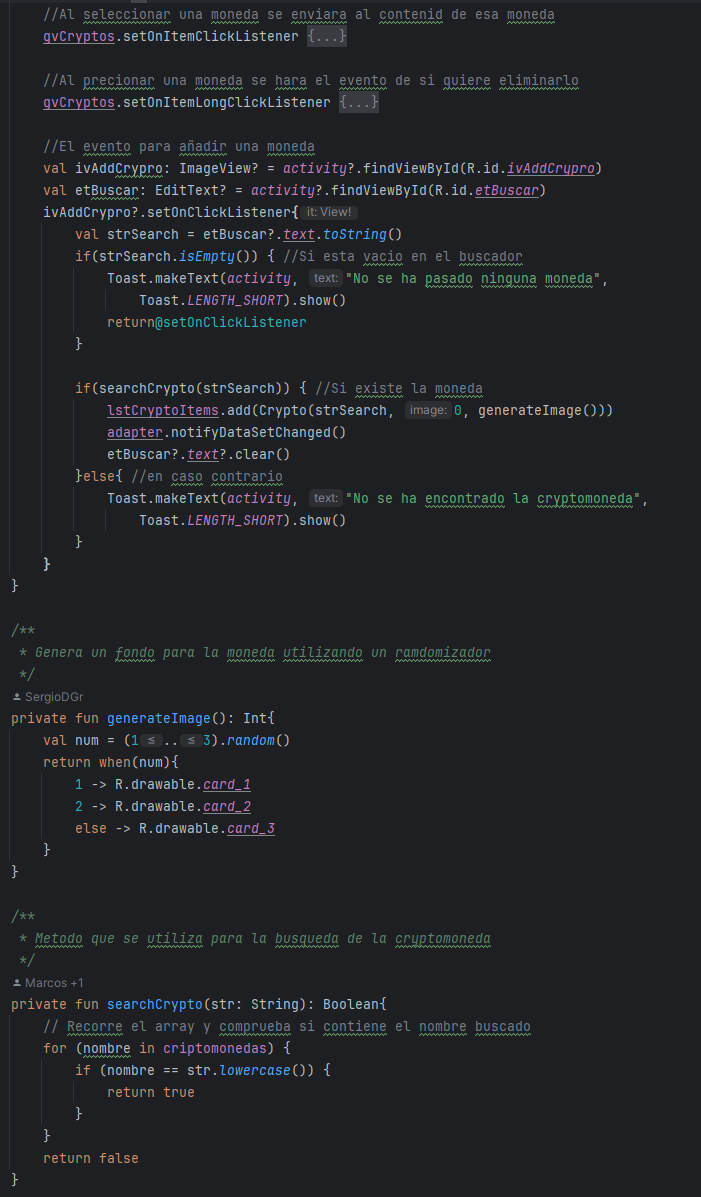
Al buscar un elemento EditText añada un elemento Moneda a la lista y se visualice el cambio en el GridView.

Proceso:

En layout añadimos EditText y ImageView como si fuera un boton para añadir el elemento moneda:



En el HomeFragment le definimos el evento del click para el añadir el elemento Moneda:



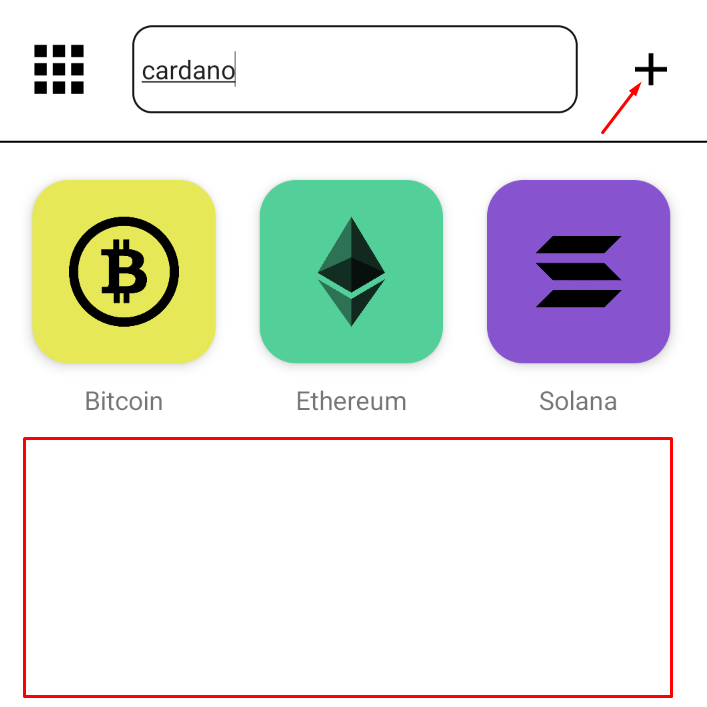
Resultado esperado:

Que, al buscar la moneda, se añada y se visualice el elemento.

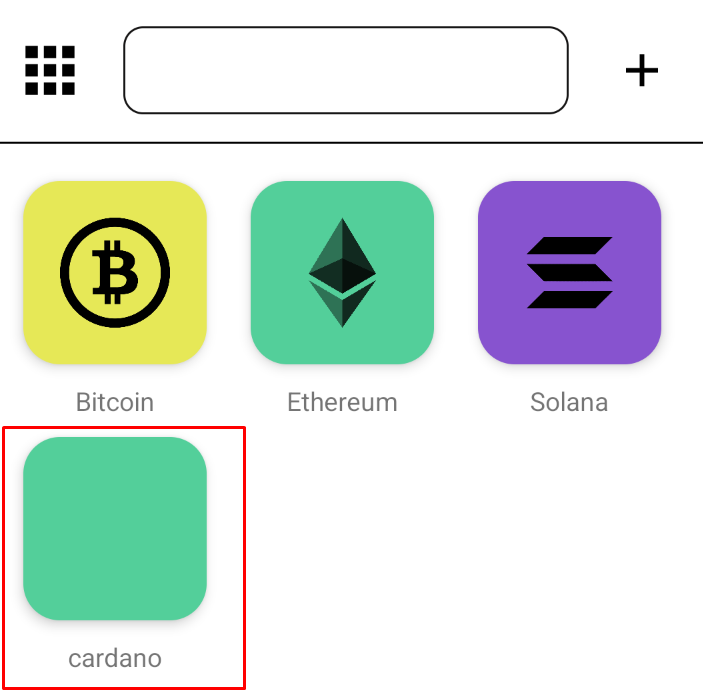
Resultado final:

Que, al buscar la moneda, se añada y se visualice el elemento.

Se busca:



Y lo guarda:



#### Prueba 7:

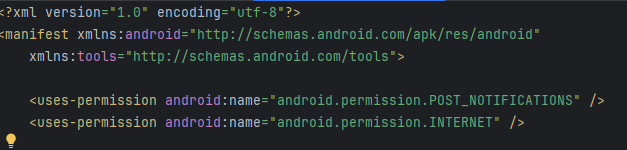
(18/05/2024)

Escenario inicial:

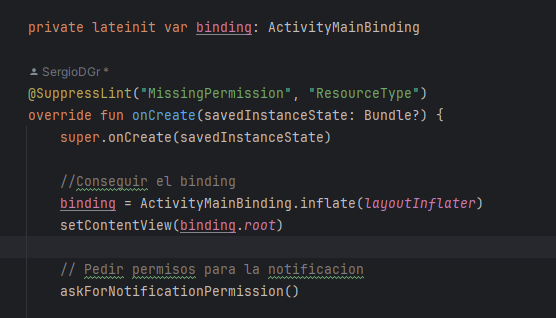
En la actividad de la Moneda poder realizar una notificación.

Proceso:

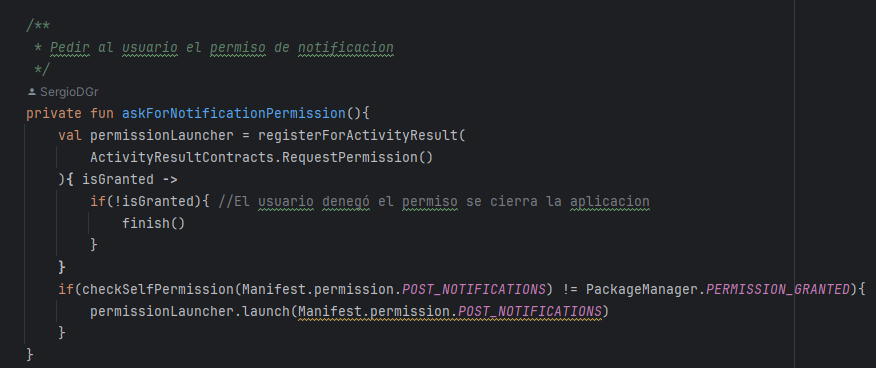
Primero tenemos que activar el los permisos de notificación, en AndroidManifest.xml:



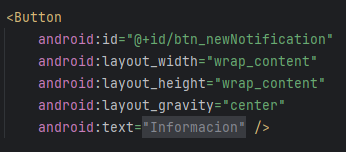
En la MainActivity pedimos permisos de notificación al usuario:



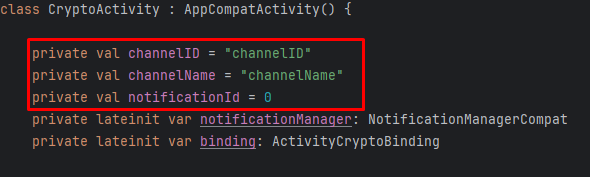
En el metodo:



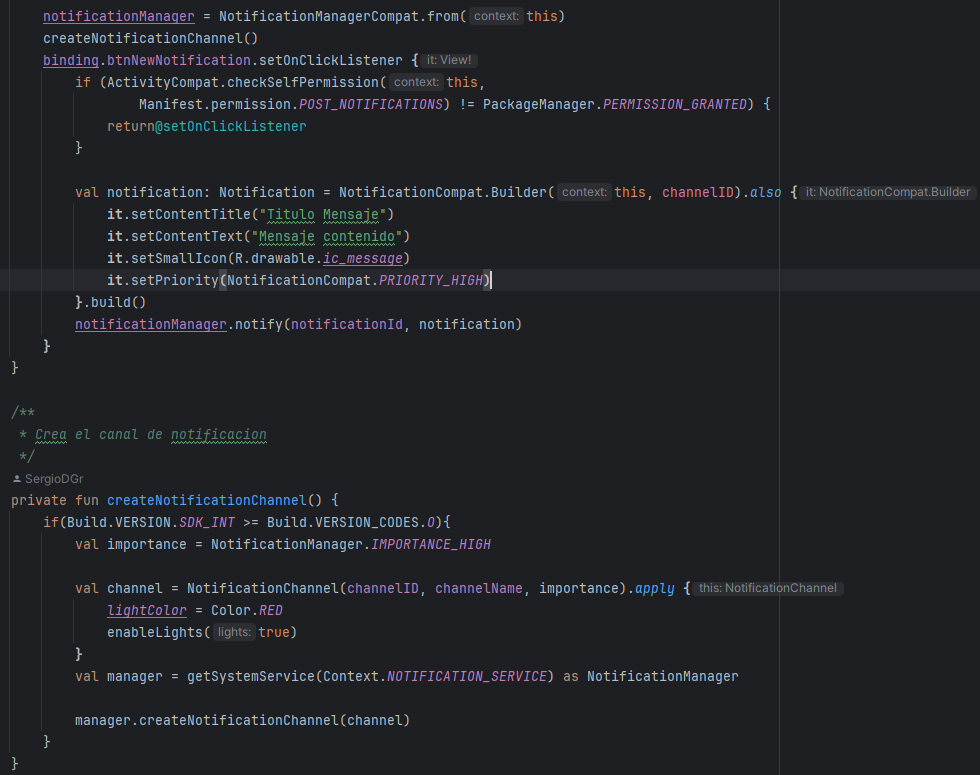
En el layout activity\_crypto añadimos el botón:



En CryptoActivity en el código. Se define los siguiente variables.



Cuando se crea la vista, al botón le definimos su evento del Click:



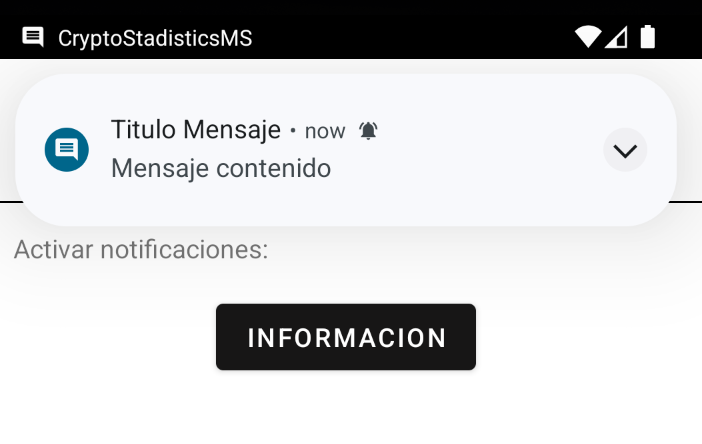
Resultado esperado:

Que visualice el mensaje al darle al botón.

Resultado final:

Que visualice el mensaje al darle al botón.





## Bibliografía

Documentación oficial para peticiones:

[Desarrolla para Android  |  Android Developers](https://developer.android.com/develop?hl=es-419)

<https://community.bitquery.io/t/how-to-get-started-with-bitquerys-blockchain-graphql-apis/13>

[Gradle plugin - Chaquopy 15.0](https://chaquo.com/chaquopy/doc/current/android.html)

[Developer David González - YouTube](https://www.youtube.com/@developergb)

[Android Knowledge - YouTube](https://www.youtube.com/@android_knowledge)

[Coding With T - YouTube](https://www.youtube.com/@CodingwithT)