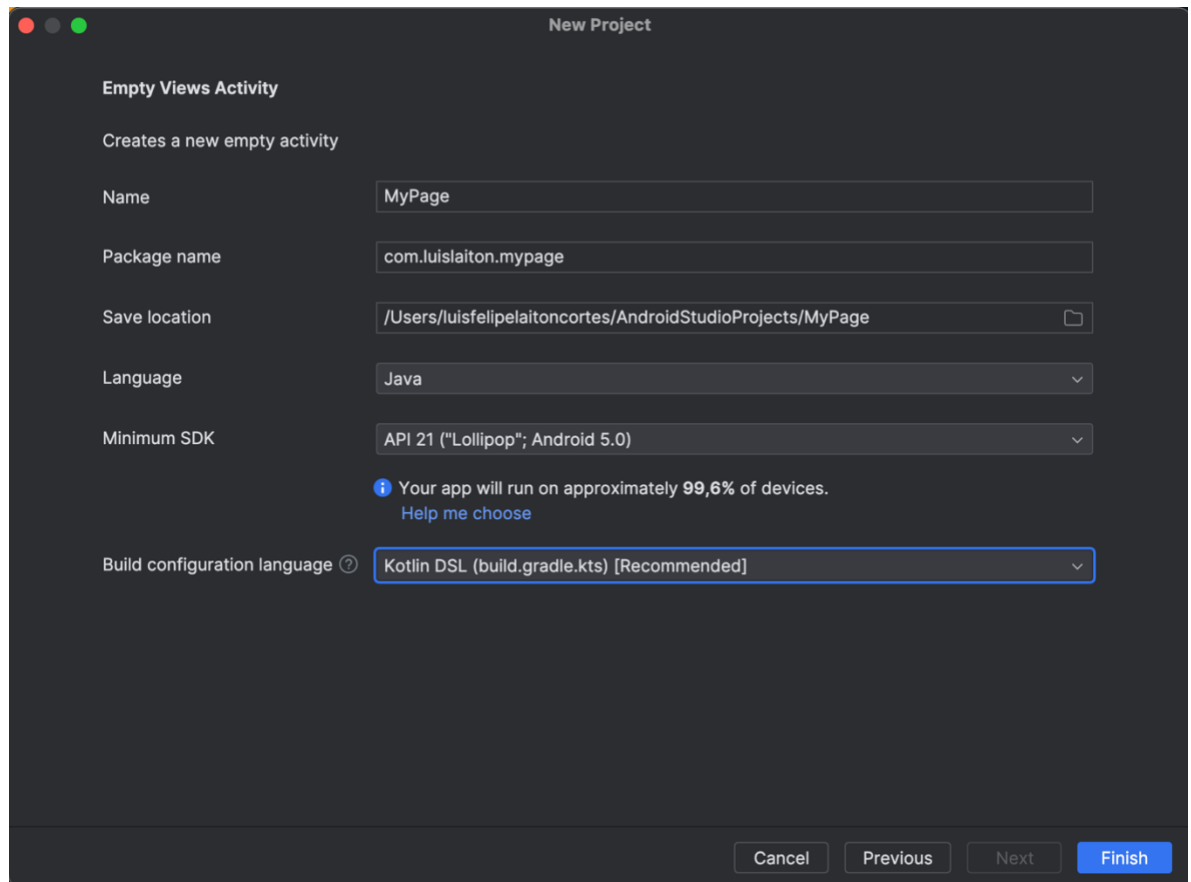


Convirtiendo una página web en una aplicación Android

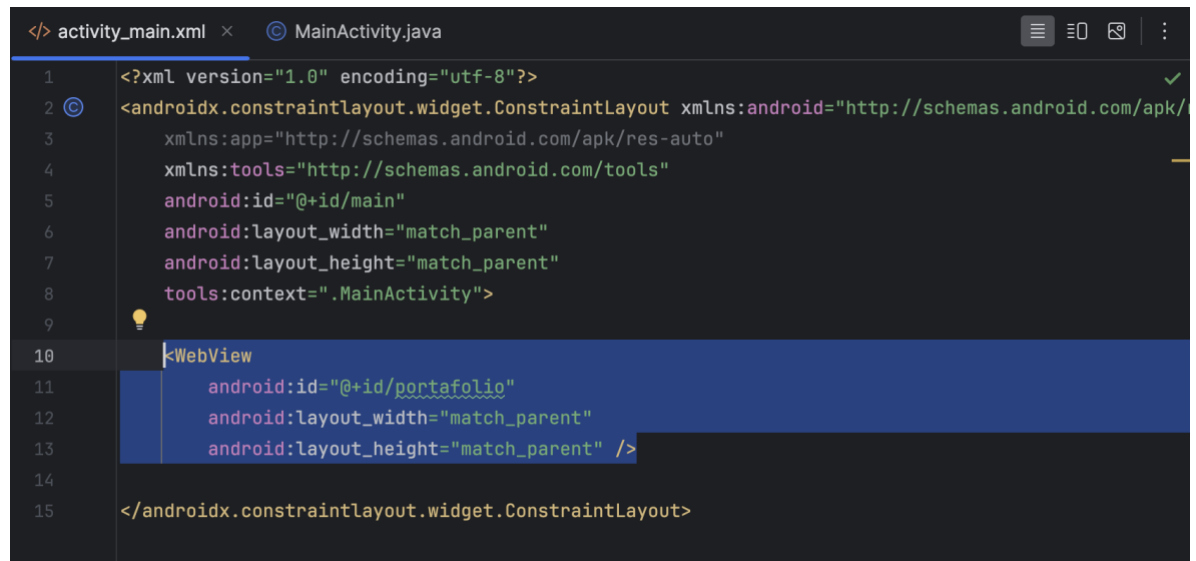
Luis F. Laiton

1. Tener una página web responsiva y disponible en internet a través de un enlace.
2. Instalar Android Studio y crear un nuevo proyecto con la opción Empty Views Activity.
3. Cambiar el nombre, asegurarse de usar Java como lenguaje, y definir la versión mínima en la cual funcionara la aplicación.



4. Luego de que carguen los paquetes, dirigirse al archivo `activity_main.xml`, eliminar la etiqueta `<TextView .../>` y cambiarla por la etiqueta `<WebView.../>`

como se ve en la imagen (el id puede ser diferente).

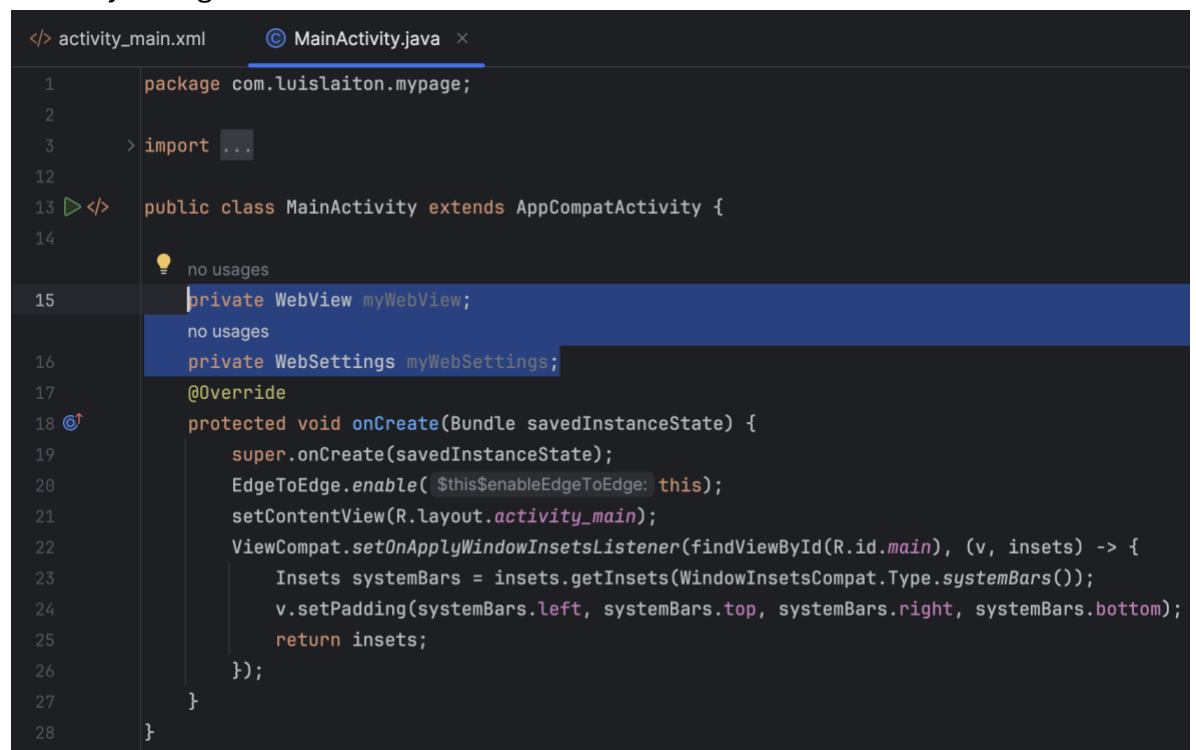


```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <WebView
        android:id="@+id/portafolio"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

5. En el archivo MainActivity.java agregar dos variables para configurar la vista de la web y configurarla adecuadamente.



```
package com.luislaiton.mypage;

import ...

public class MainActivity extends AppCompatActivity {

    private WebView myWebView;

    private WebSettings myWebSettings;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}
```

6. Agregar las siguientes líneas para configurar el acceso de la app a la configuración de la página web.

```
public class MainActivity extends AppCompatActivity {

    2 usages
    private WebView myWebView;
    3 usages
    private WebSettings myWebSettings;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

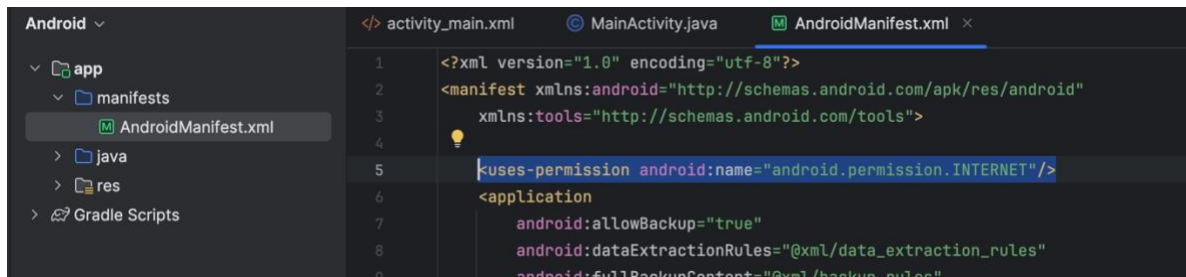
        myWebView = findViewById(R.id.portafolio);
        myWebSettings = myWebView.getSettings();
        myWebSettings.setJavaScriptEnabled(true);
        myWebSettings.setDomStorageEnabled(true);

    }
}
```

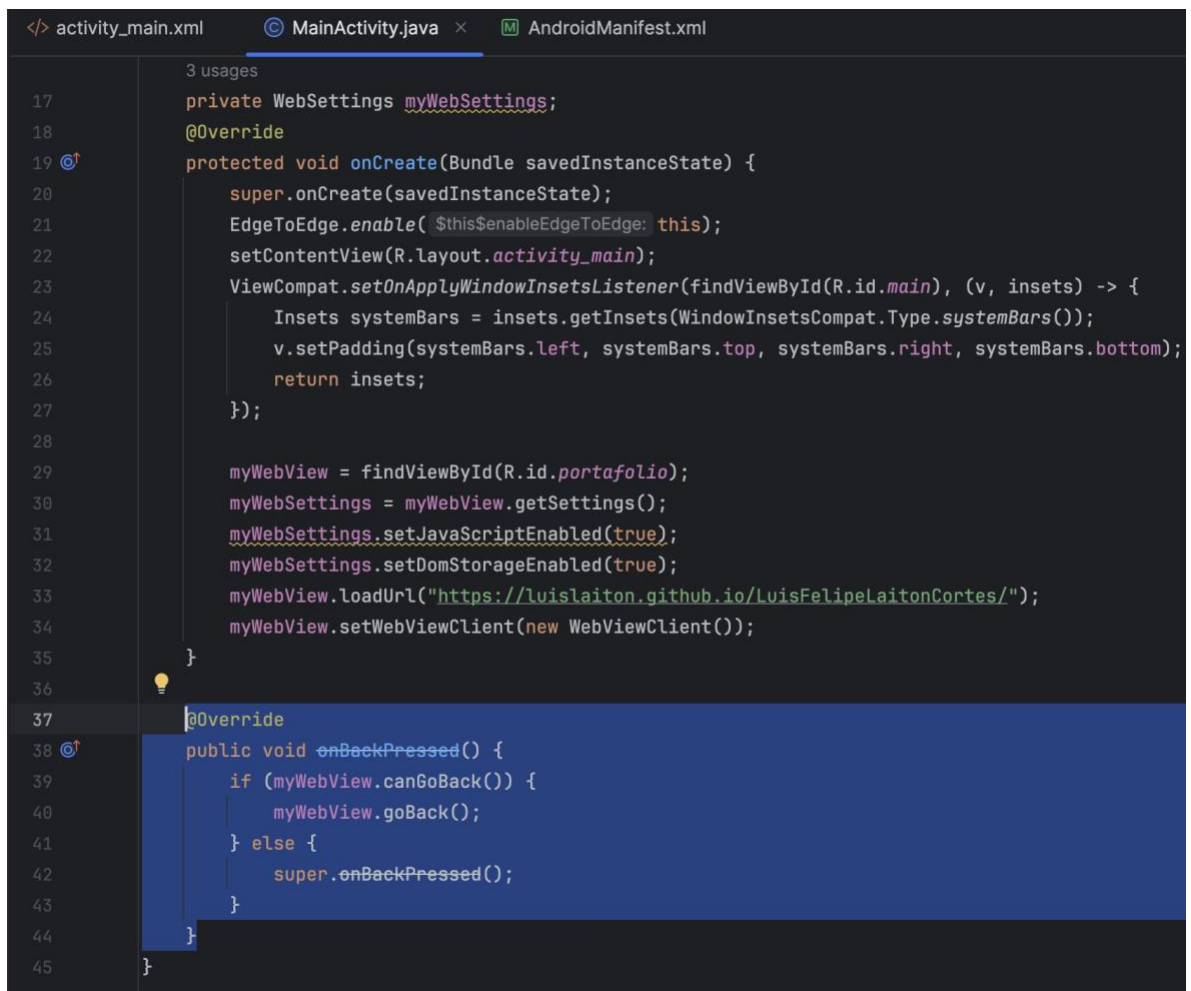
7. Agregar el comando para usar la página web mediante un enlace y debajo el comando para definir una nueva vista en donde el celular actuara como cliente ante la página web.

```
myWebView = findViewById(R.id.portafolio);
myWebSettings = myWebView.getSettings();
myWebSettings.setJavaScriptEnabled(true);
myWebSettings.setDomStorageEnabled(true);
myWebView.loadUrl("https://luislaiton.github.io/LuisFelipeLaitonCortes/");
myWebView.setWebViewClient(new WebViewClient());
```

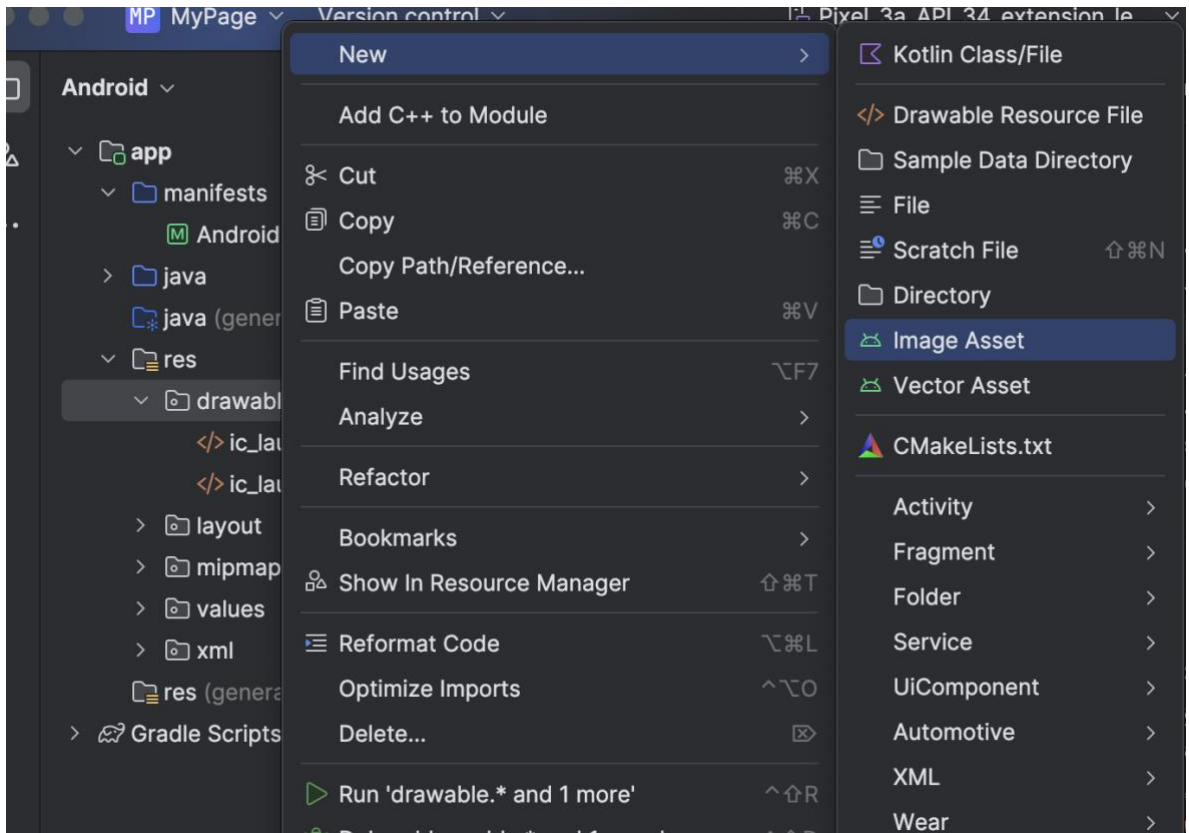
8. Acceder al archivo AndroidManifest.xml para dar los accesos de conexión a internet.



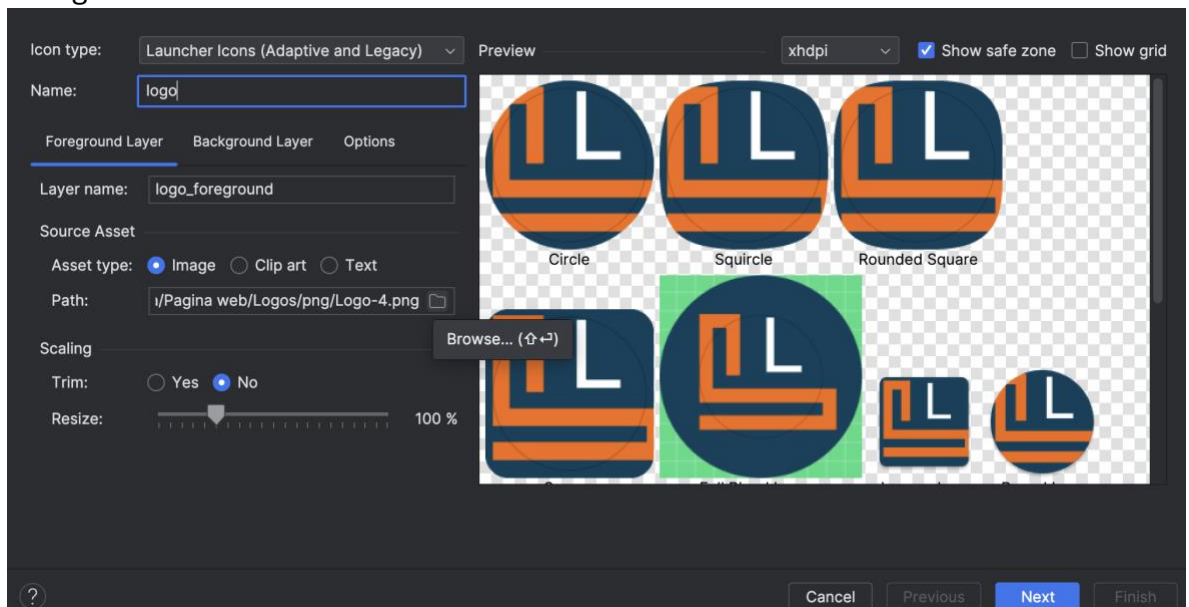
9. En el archivo MainActivity.java agregamos otra función llamada `onBackPressed()` para asegurar que podremos usar el retroceso sin que se salga de la aplicación.



10. Para cambiar el icono de la aplicación vamos a la carpeta `res > drawable` y desplegamos el menú de opciones.



11. Clic en la opción de Path para buscar la imagen y definimos un nombre para la imagen.



12. En el archivo AndroidManifest.xml buscar la línea donde se llama el icono y cambiarlo por el nombre que se definió en la imagen, del mismo modo cambiar roundIcon como se ve en la imagen.

```
activity_main.xml MainActivity.java AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools">
4
5     <uses-permission android:name="android.permission.INTERNET"/>
6     <application
7         android:allowBackup="true"
8         android:dataExtractionRules="@xml/data_extraction_rules"
9         android:fullBackupContent="@xml/backup_rules"
10        android:icon="@mipmap/logo"
11        android:label="MyPage"
12        android:roundIcon="@mipmap/logo_round"
13        android:supportsRtl="true"
14        android:theme="@style/Theme.MyPage"
15        tools:targetApi="31">
16     <activity
```

13. Para cambiar el nombre que muestra la aplicación, ir al archivo en la carpeta res > values > strings.xml

```
Android v
v app
  v manifests
    AndroidManifest.xml
  > java
    java (generated)
  v res
    > drawable
    > layout
    > mipmap
    v values
      </> colors.xml
      </> strings.xml
    > themes (2)

ml MainActivity.java AndroidManifest.xml strings.xml x
1 <resources>
2 <string name="app_name">luisf.laiton</string>
3 </resources>
```