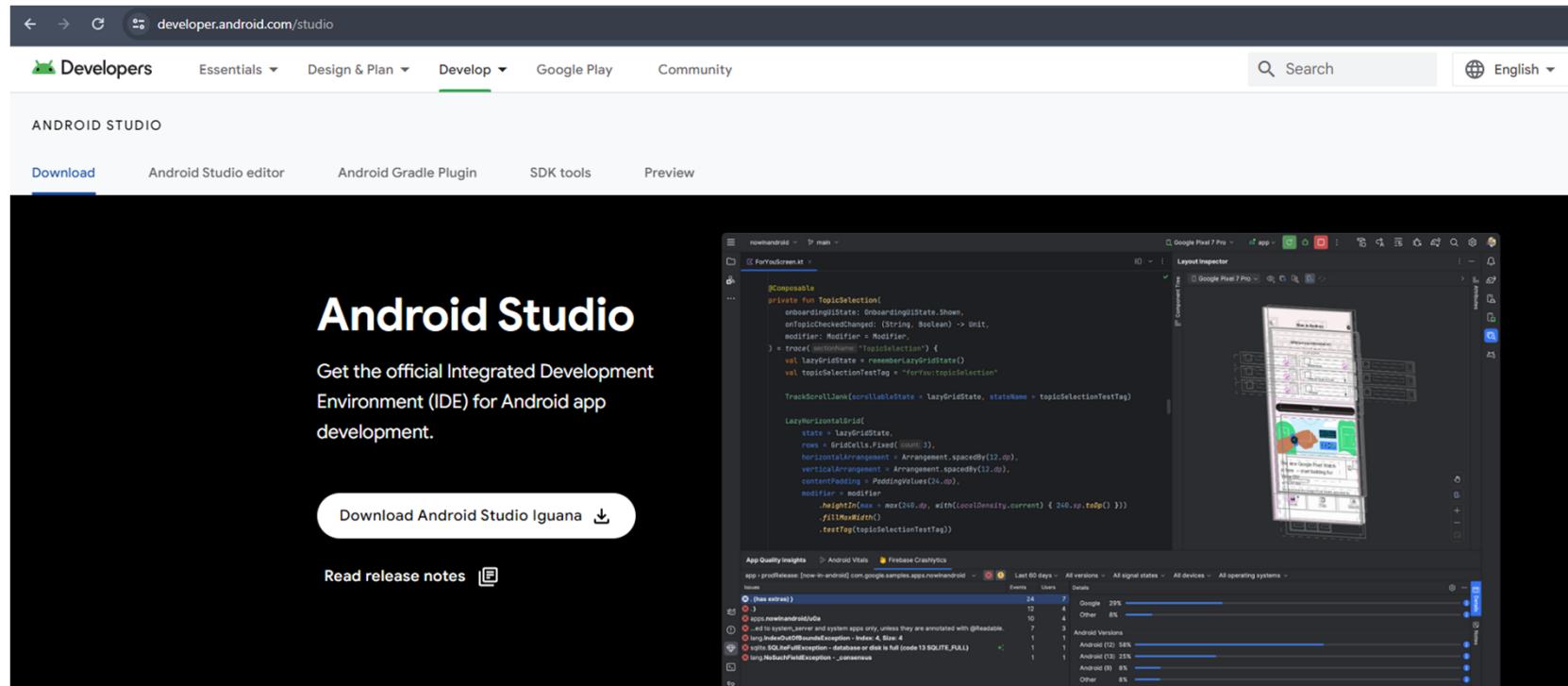


Android Studio

IDE for Android app development

Descarga



Si a todo

10.1 YOU EXPRESSLY UNDERSTAND AND AGREE THAT YOUR USE OF THE SDK IS AT YOUR SOLE RISK AND THAT THE SDK IS PROVIDED "AS IS" AND "AS AVAILABLE" WITHOUT WARRANTY OF ANY KIND FROM GOOGLE. 10.2 YOUR USE OF THE SDK AND ANY MATERIAL DOWNLOADED OR OTHERWISE OBTAINED THROUGH THE USE OF THE SDK IS AT YOUR OWN DISCRETION AND RISK AND YOU ARE SOLELY RESPONSIBLE FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR OTHER DEVICE OR LOSS OF DATA THAT RESULTS FROM SUCH USE. 10.3 GOOGLE FURTHER EXPRESSLY DISCLAIMS ALL WARRANTIES AND CONDITIONS OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

11. LIMITATION OF LIABILITY

11.1 YOU EXPRESSLY UNDERSTAND AND AGREE THAT GOOGLE, ITS SUBSIDIARIES AND AFFILIATES, AND ITS LICENSORS SHALL NOT BE LIABLE TO YOU UNDER ANY THEORY OF LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES THAT MAY BE INCURRED BY YOU, INCLUDING ANY LOSS OF DATA, WHETHER OR NOT GOOGLE OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF OR SHOULD HAVE BEEN AWARE OF THE POSSIBILITY OF ANY SUCH LOSSES ARISING.

12. Indemnification

12.1 To the maximum extent permitted by law, you agree to defend, indemnify and hold harmless Google, its affiliates and their respective directors, officers, employees and agents from and against any and all claims, actions, suits or proceedings, as well as any and all losses, liabilities, damages, costs and expenses (including reasonable attorneys fees) arising out of or accruing from (a) your use of the SDK, (b) any application you develop on the SDK that infringes any copyright, trademark, trade secret, trade dress, patent or other intellectual property right of any person or defames any person or violates their rights of publicity or privacy, and (c) any non-compliance by you with the License Agreement.

13. Changes to the License Agreement

13.1 Google may make changes to the License Agreement as it distributes new versions of the SDK. When these changes are made, Google will make a new version of the License Agreement available on the website where the SDK is made available.

14. General Legal Terms

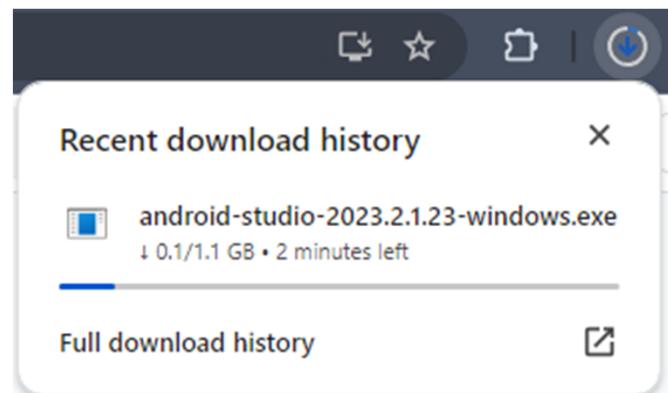
14.1 The License Agreement constitutes the whole legal agreement between you and Google and governs your use of the SDK (excluding any services which Google may provide to you under a separate written agreement), and completely replaces any prior agreements between you and Google in relation to the SDK. 14.2 You agree that if Google does not exercise or enforce any legal right or remedy which is contained in the License Agreement (or which Google has the benefit of under any applicable law), this will not be taken to be a formal waiver of Google's rights and that those rights or remedies will still be available to Google. 14.3 If any court of law, having the jurisdiction to decide on this matter, rules that any provision of the License Agreement is invalid, then that provision will be removed from the License Agreement without affecting the rest of the License Agreement. The remaining provisions of the License Agreement will continue to be valid and enforceable. 14.4 You acknowledge and agree that each member of the group of companies of which Google is the parent shall be third party beneficiaries to the License Agreement and that such other companies shall be entitled to directly enforce, and rely upon, any provision of the License Agreement that confers a benefit on (or rights in favor of) them. Other than this, no other person or company shall be third party beneficiaries to the License Agreement. 14.5 EXPORT RESTRICTIONS. THE SDK IS SUBJECT TO UNITED STATES EXPORT LAWS AND REGULATIONS. YOU MUST COMPLY WITH ALL DOMESTIC AND INTERNATIONAL EXPORT LAWS AND REGULATIONS THAT APPLY TO THE SDK. THESE LAWS INCLUDE RESTRICTIONS ON DESTINATIONS, END USERS AND END USE. 14.6 The rights granted in the License Agreement may not be assigned or transferred by either you or Google without the prior written approval of the other party. Neither you nor Google shall be permitted to delegate their responsibilities or obligations under the License Agreement without the prior written approval of the other party. 14.7 The License Agreement, and your relationship with Google under the License Agreement, shall be governed by the laws of the State of California without regard to its conflict of laws provisions. You and Google agree to submit to the exclusive jurisdiction of the courts located within the county of Santa Clara, California to resolve any legal matter arising from the License Agreement. Notwithstanding this, you agree that Google shall still be allowed to apply for injunctive remedies (or an equivalent type of urgent legal relief) in any jurisdiction. July 27, 2021

I have read and agree with the above terms and conditions

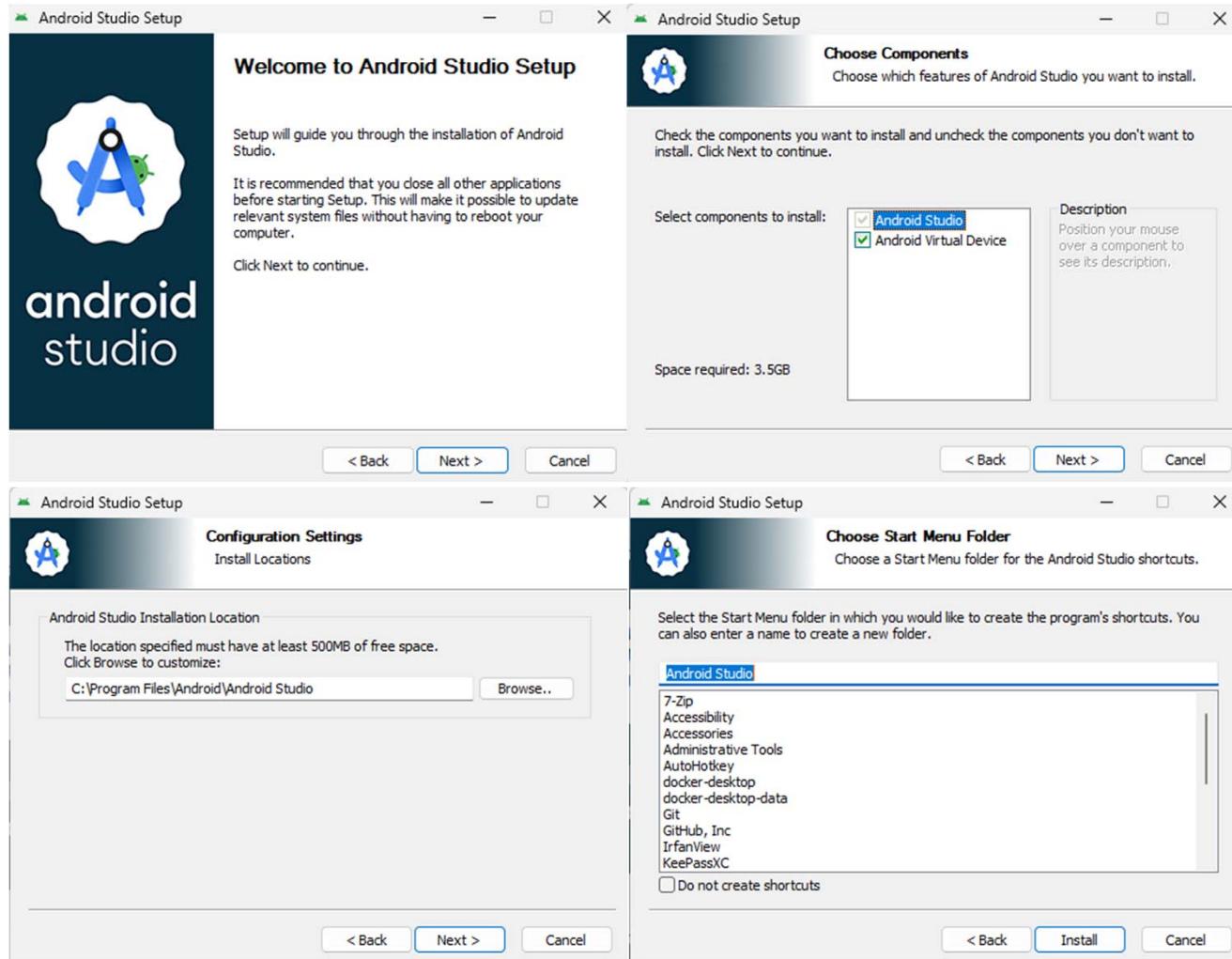
[Download Android Studio Iguana | 2023.2.1 for Windows](#)

android-studio-2023.2.1.23-windows.exe

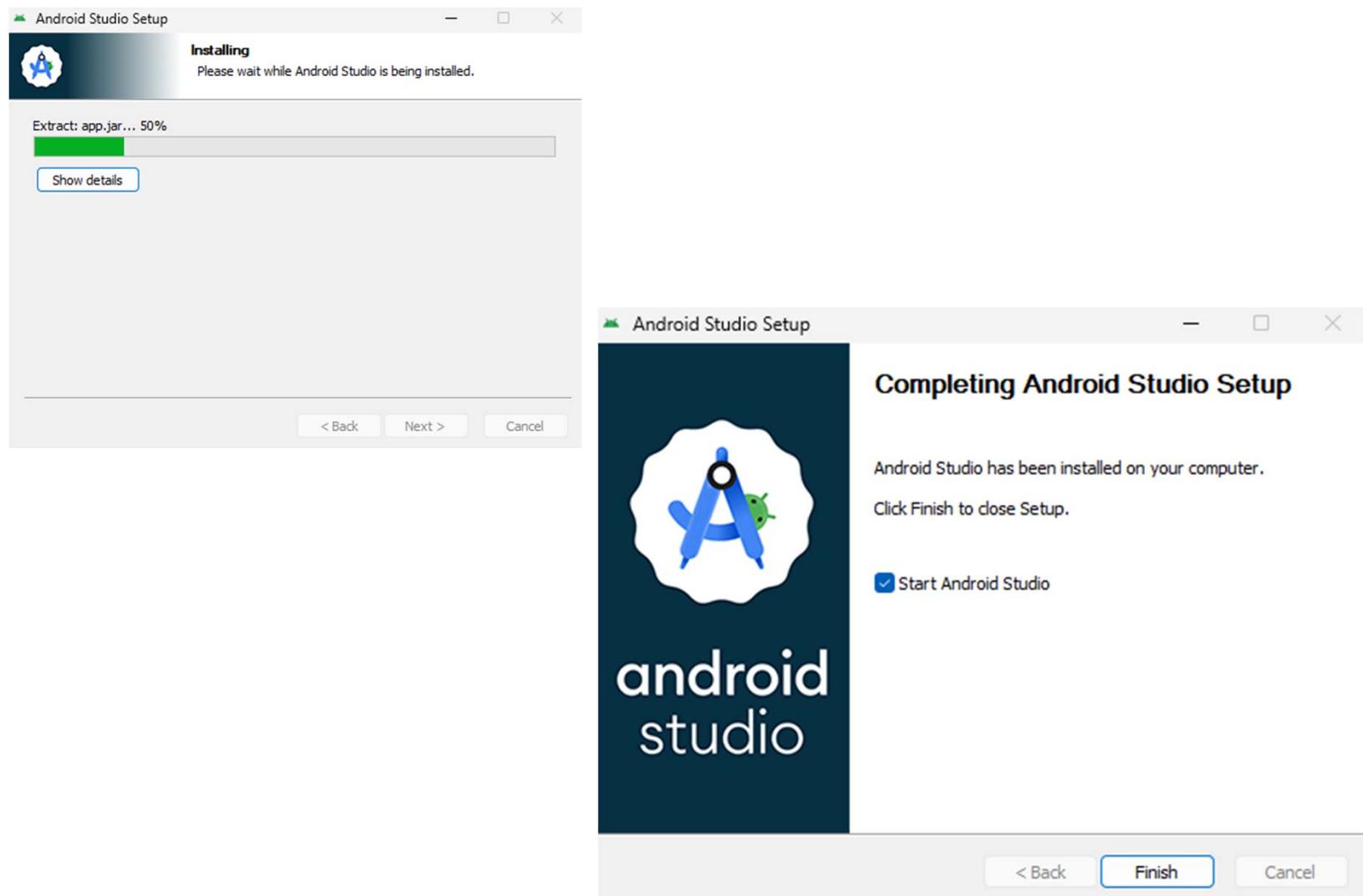
Esperando la descarga ...



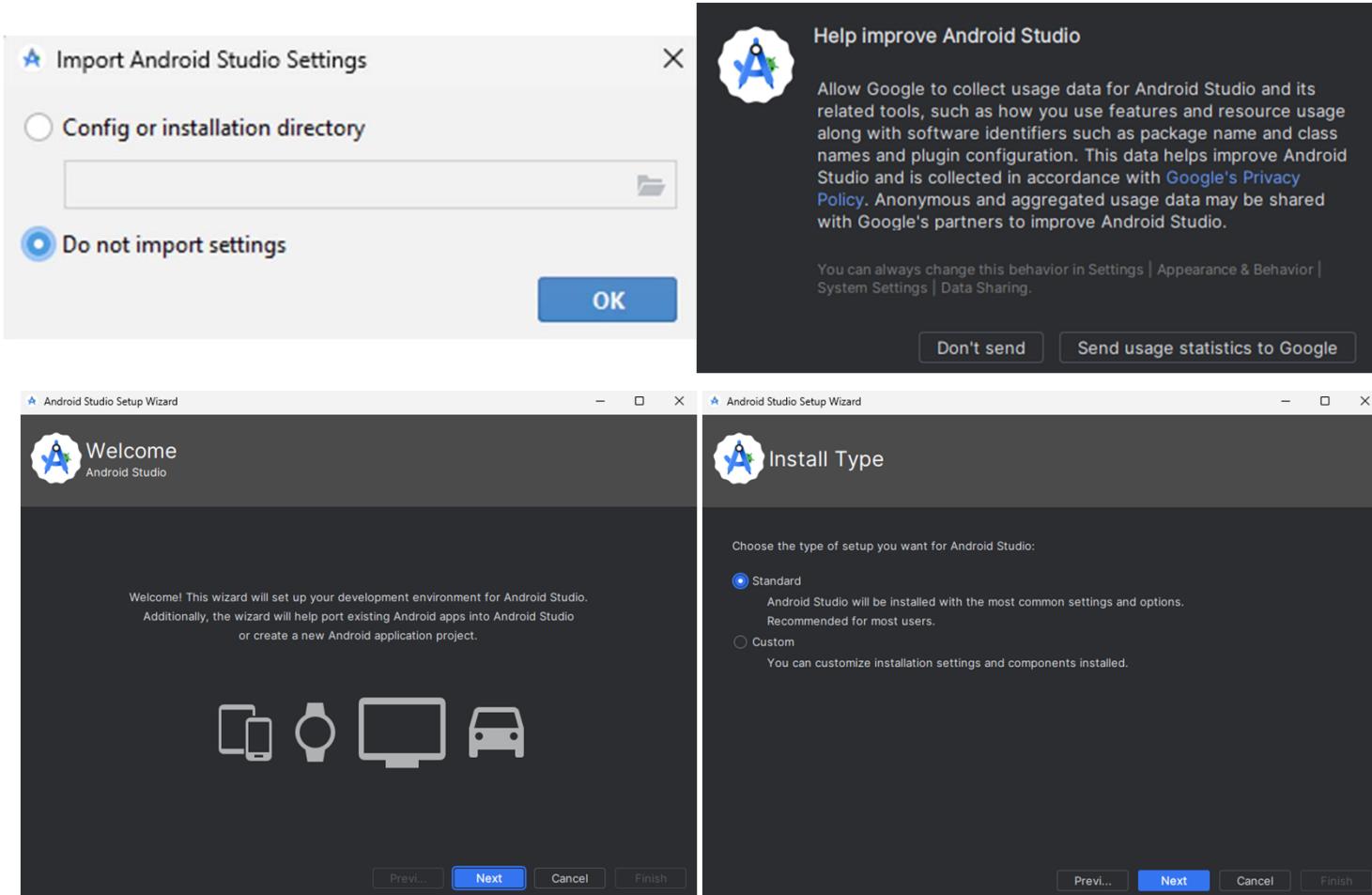
Instalando ...



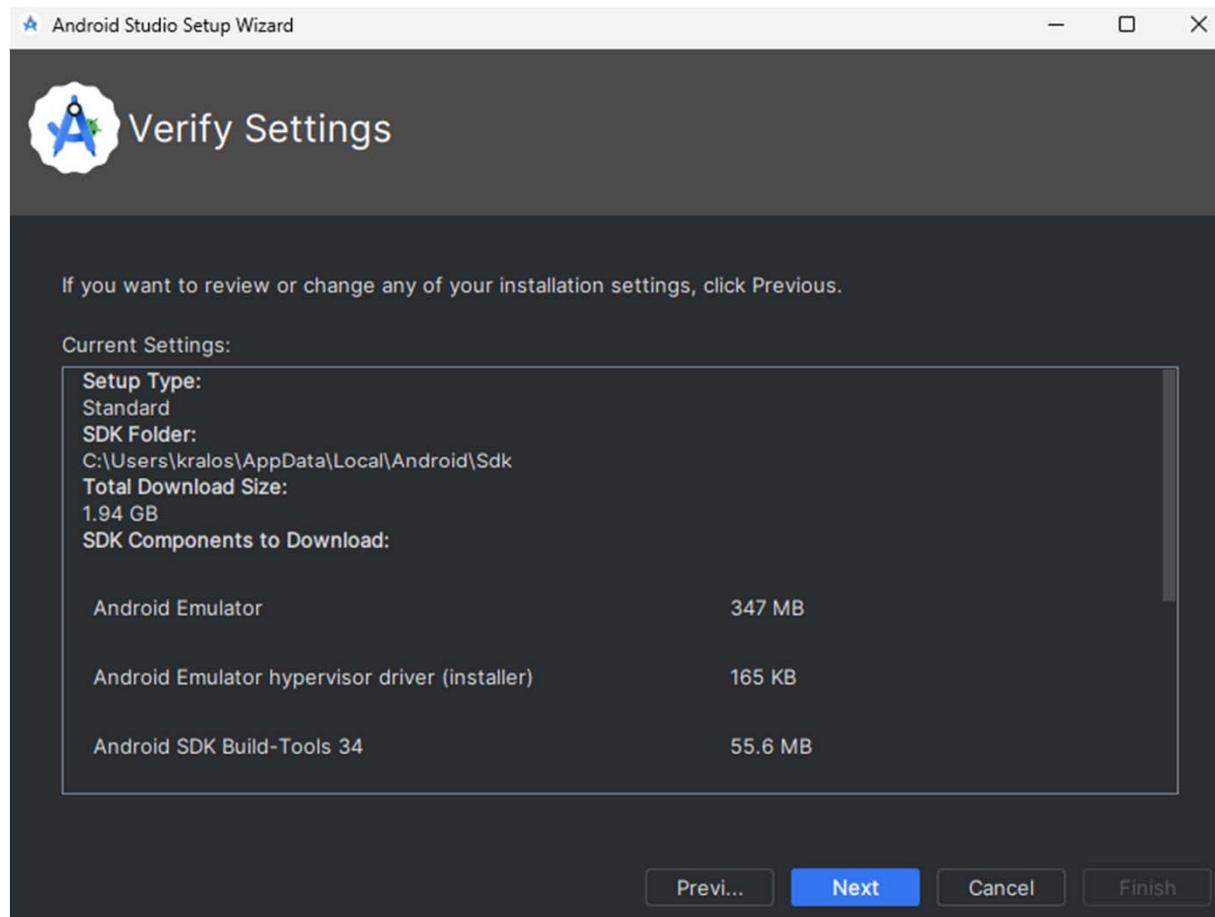
Instalando ...



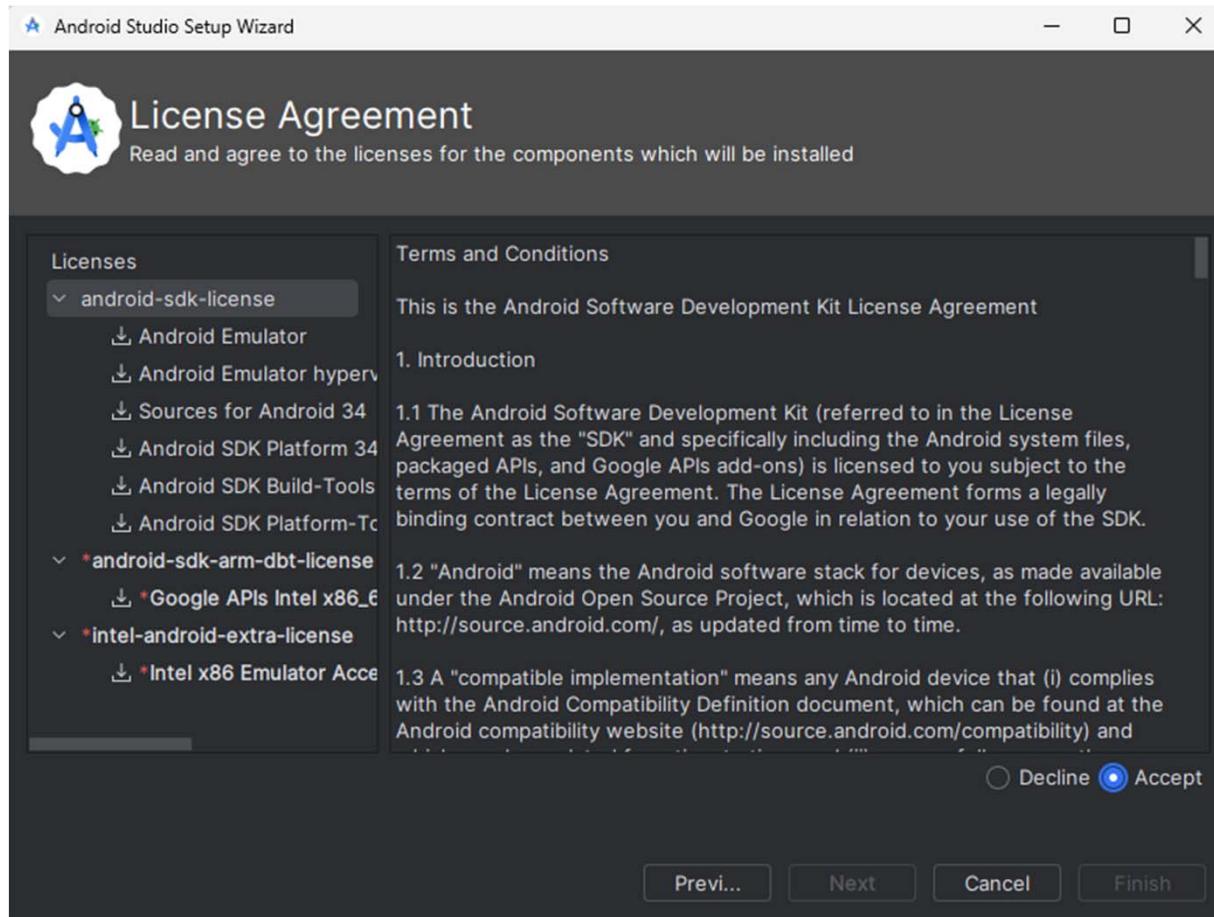
Configurando ...



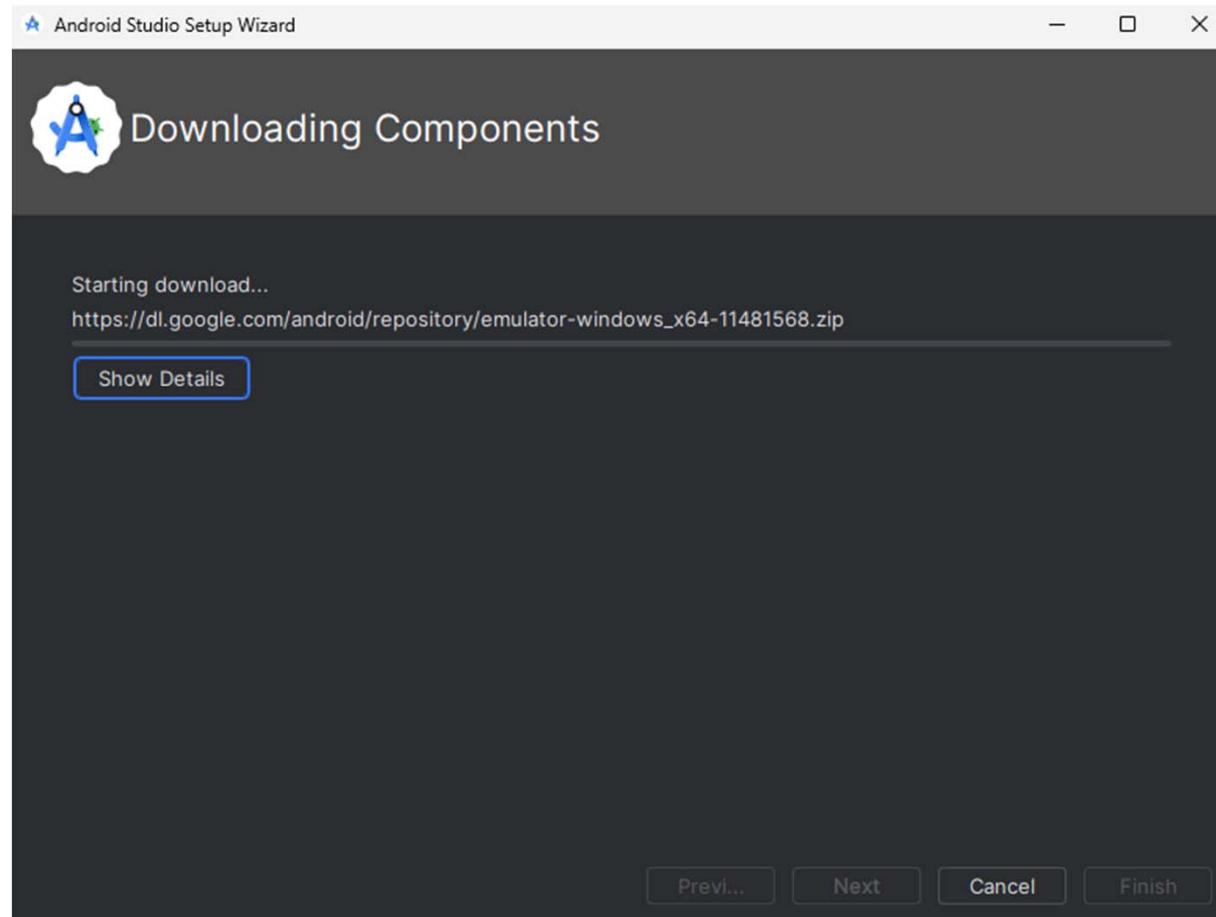
Configurando ...



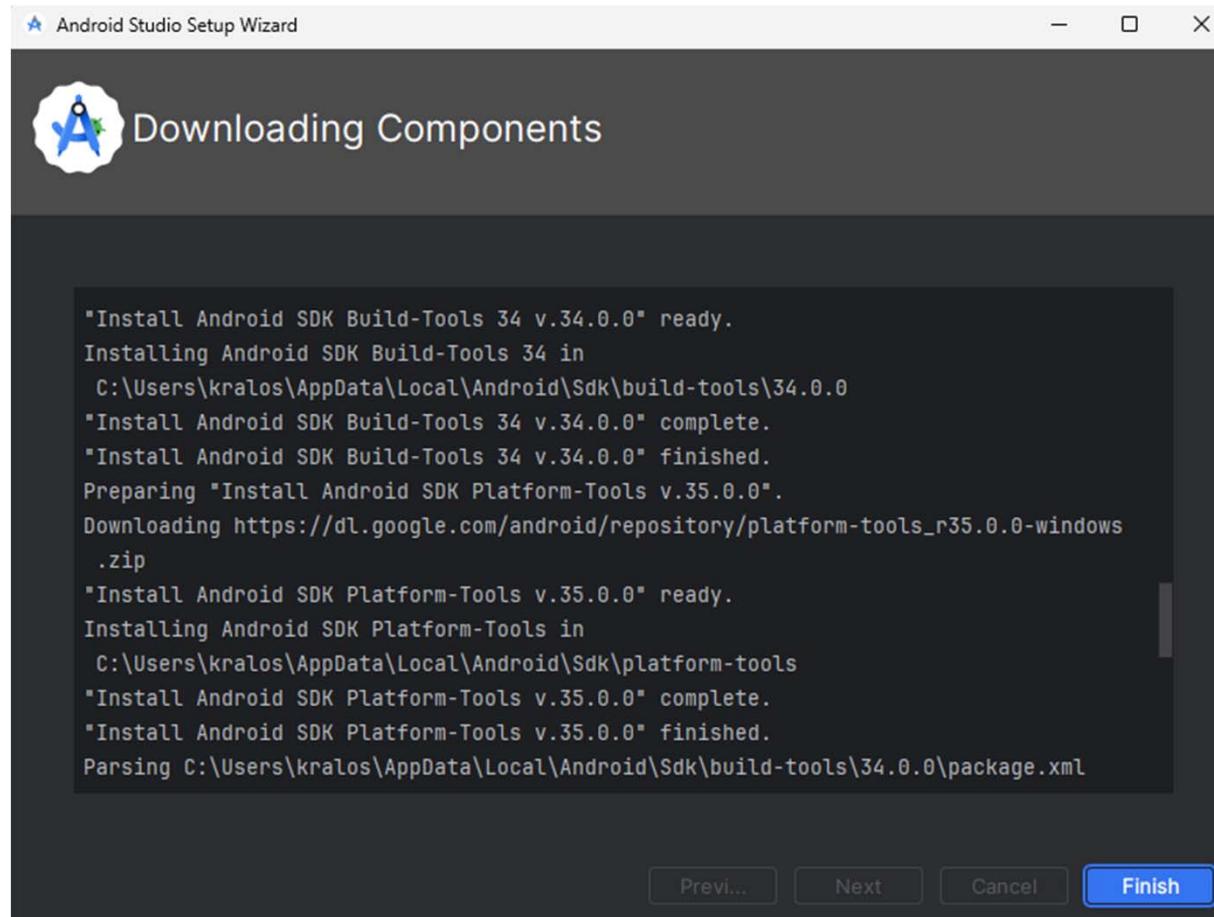
Configurando ...



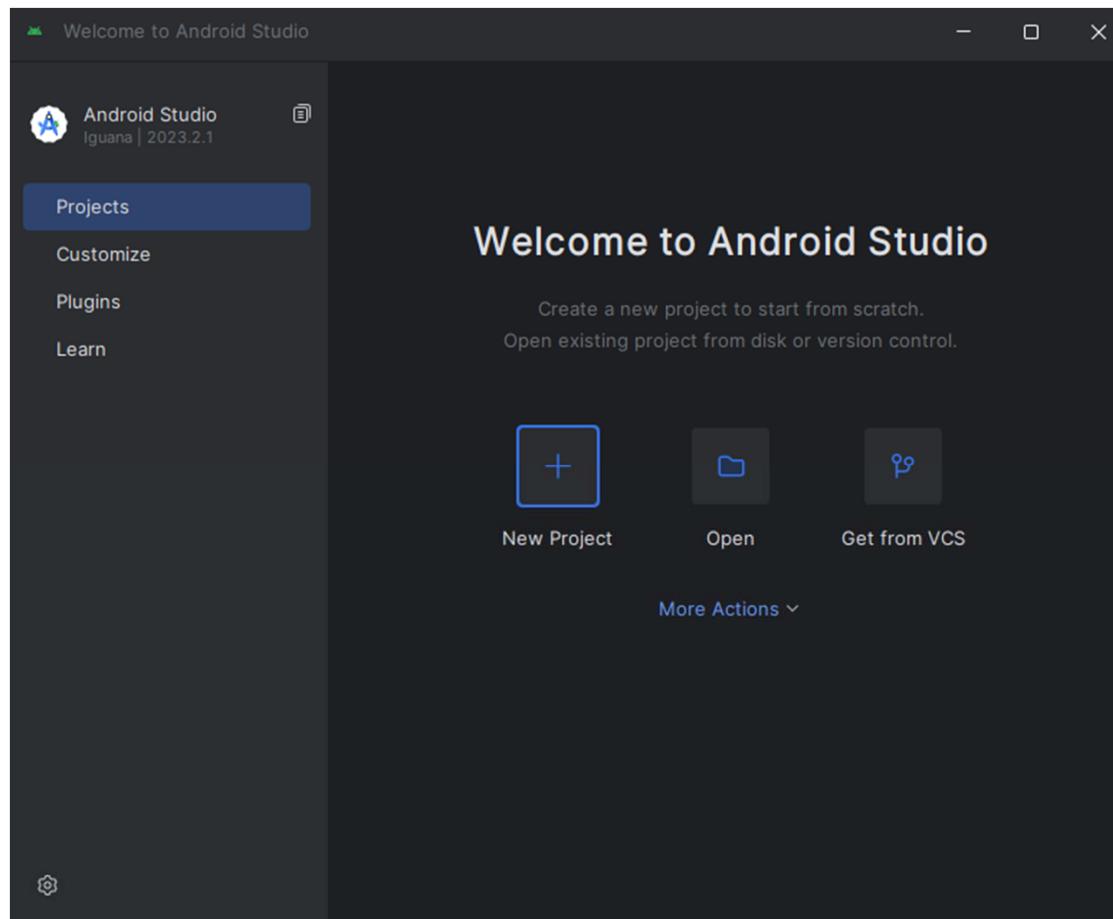
Configurando ... descarga extra



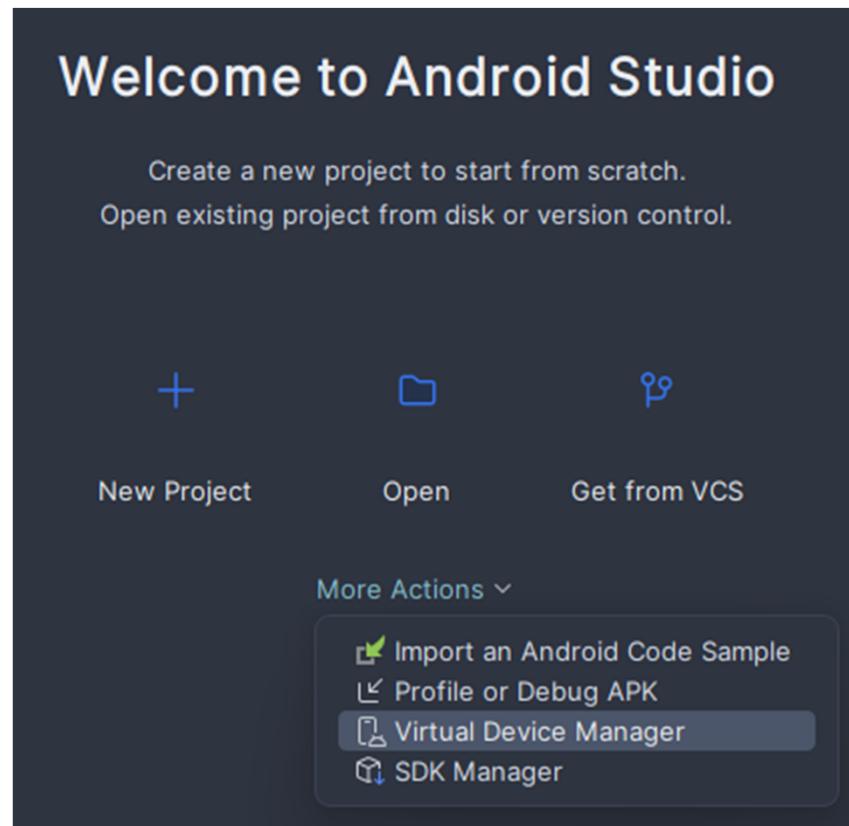
Configurando ... descarga extra



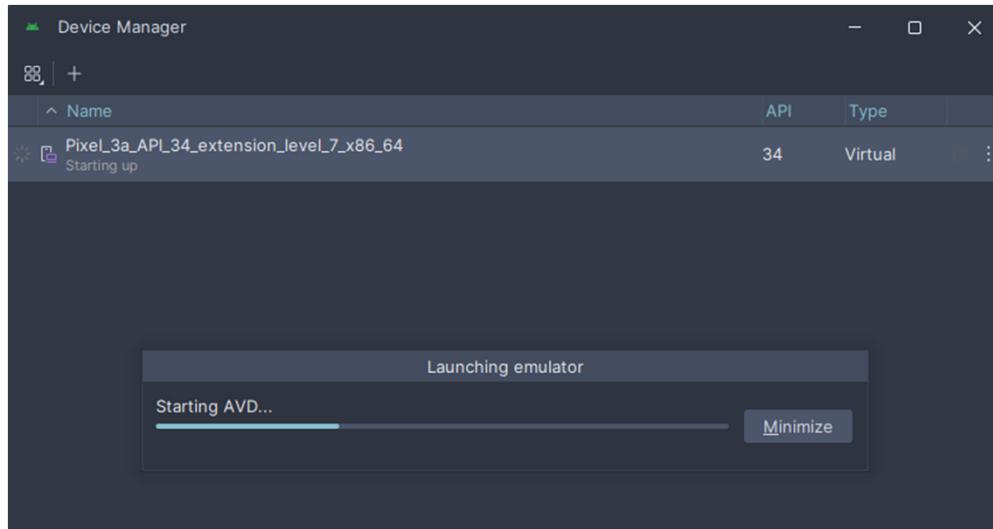
Casi todo listo ...



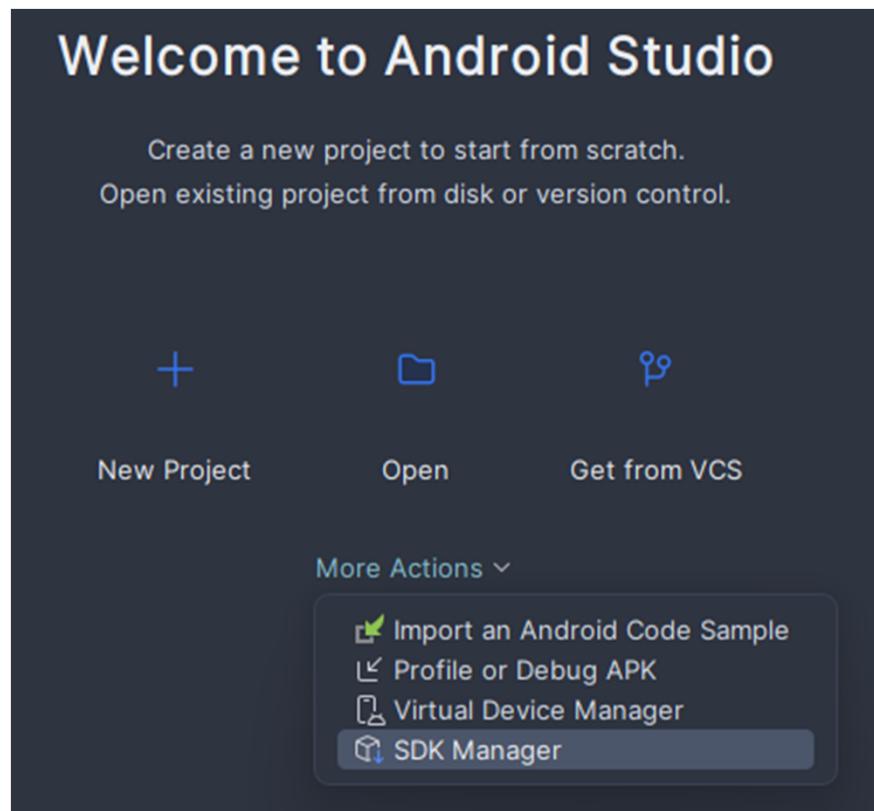
Casi todo listo ...



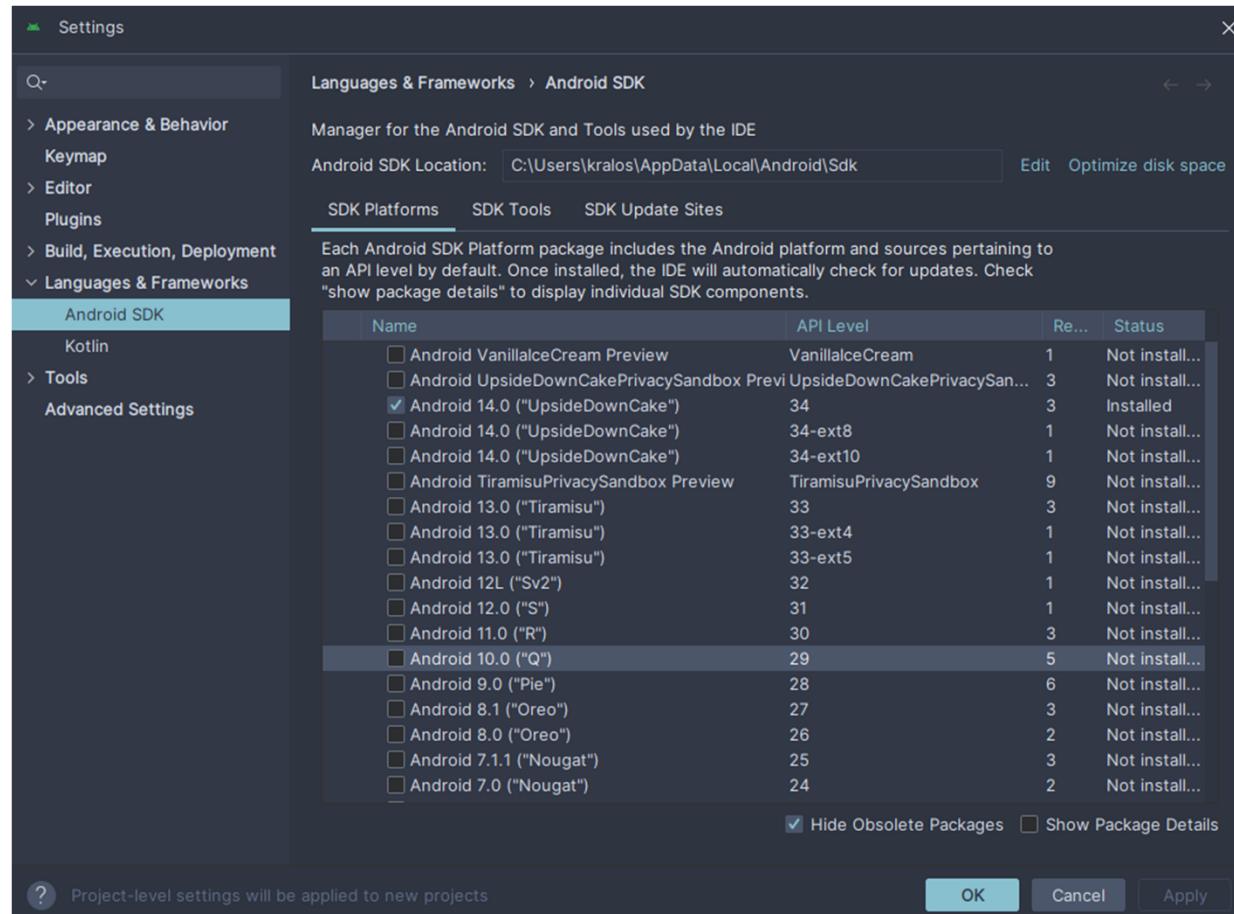
Casi todo listo ...



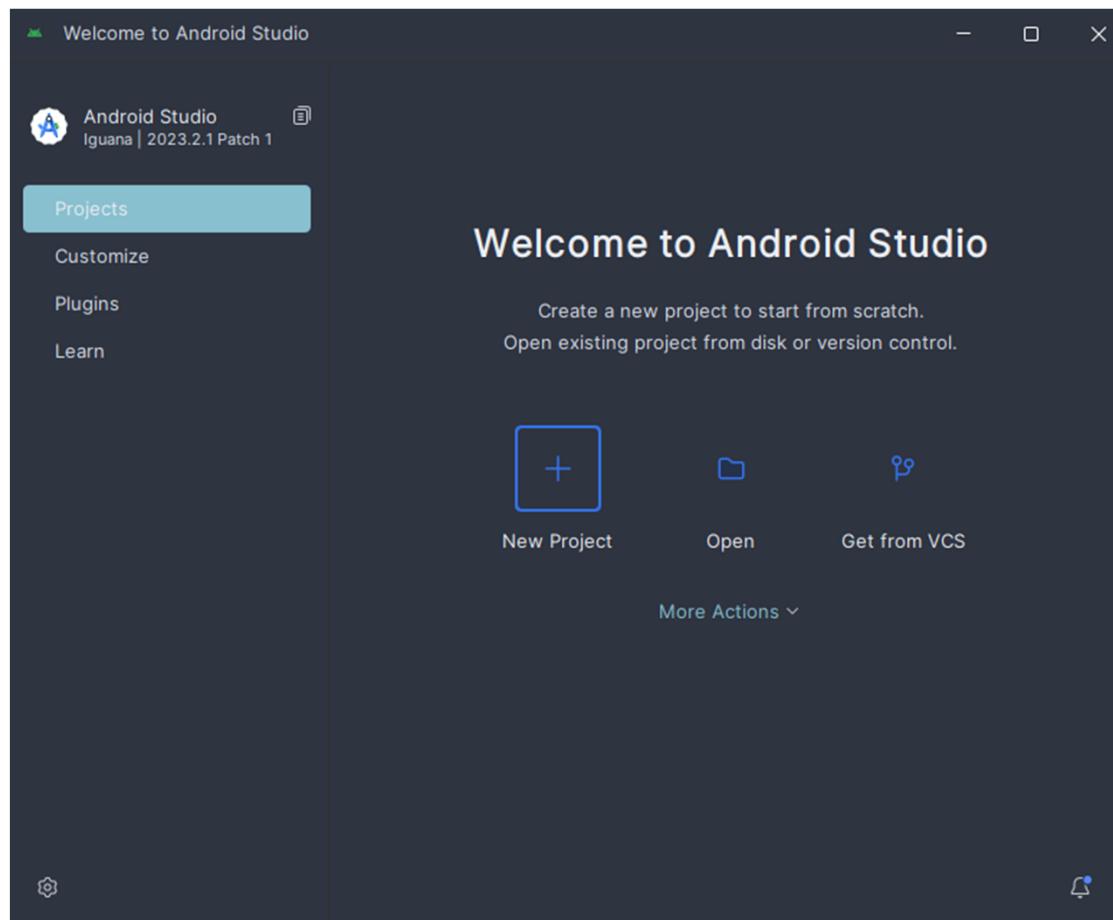
Casi todo listo ...



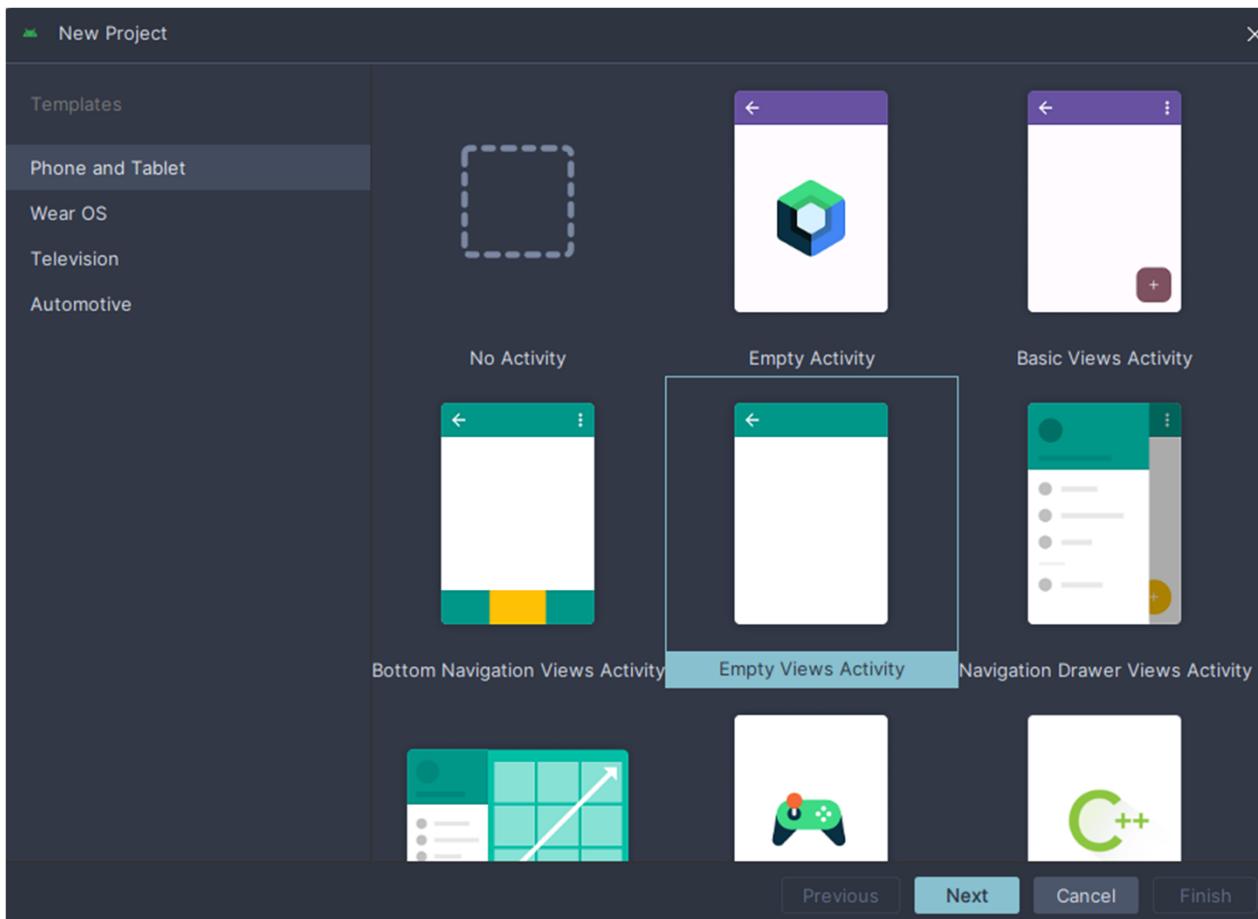
Casi todo listo ...



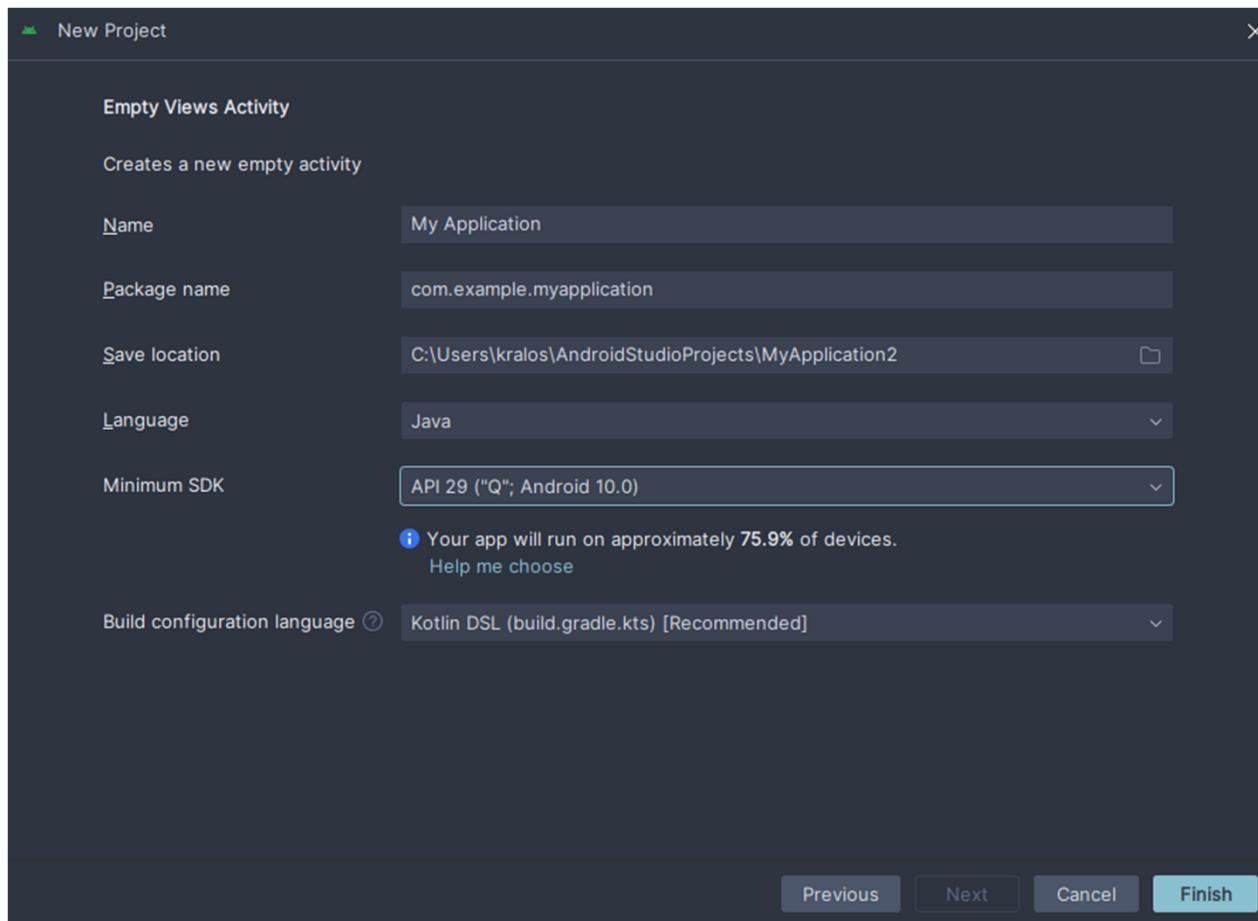
Primer App



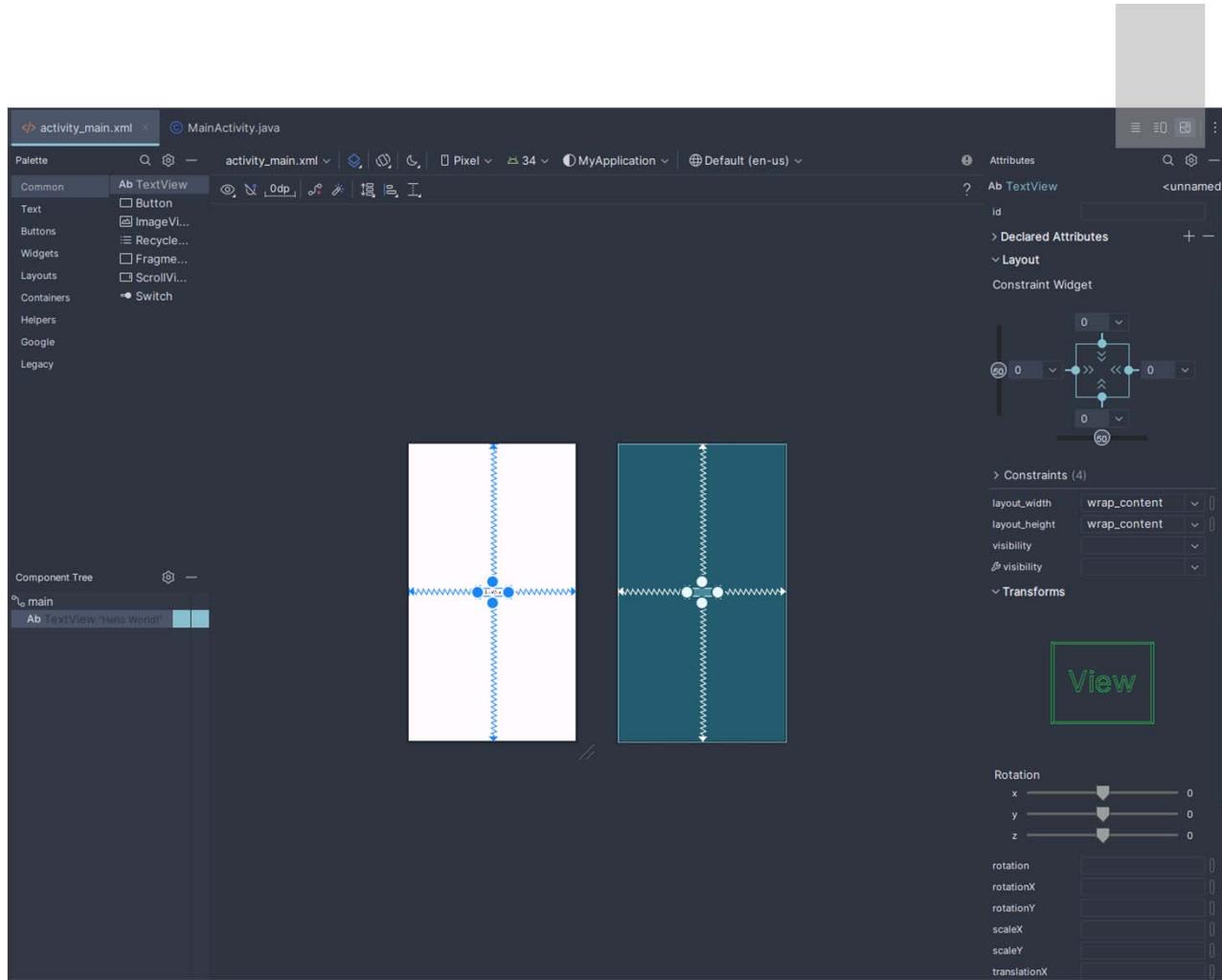
Primer App



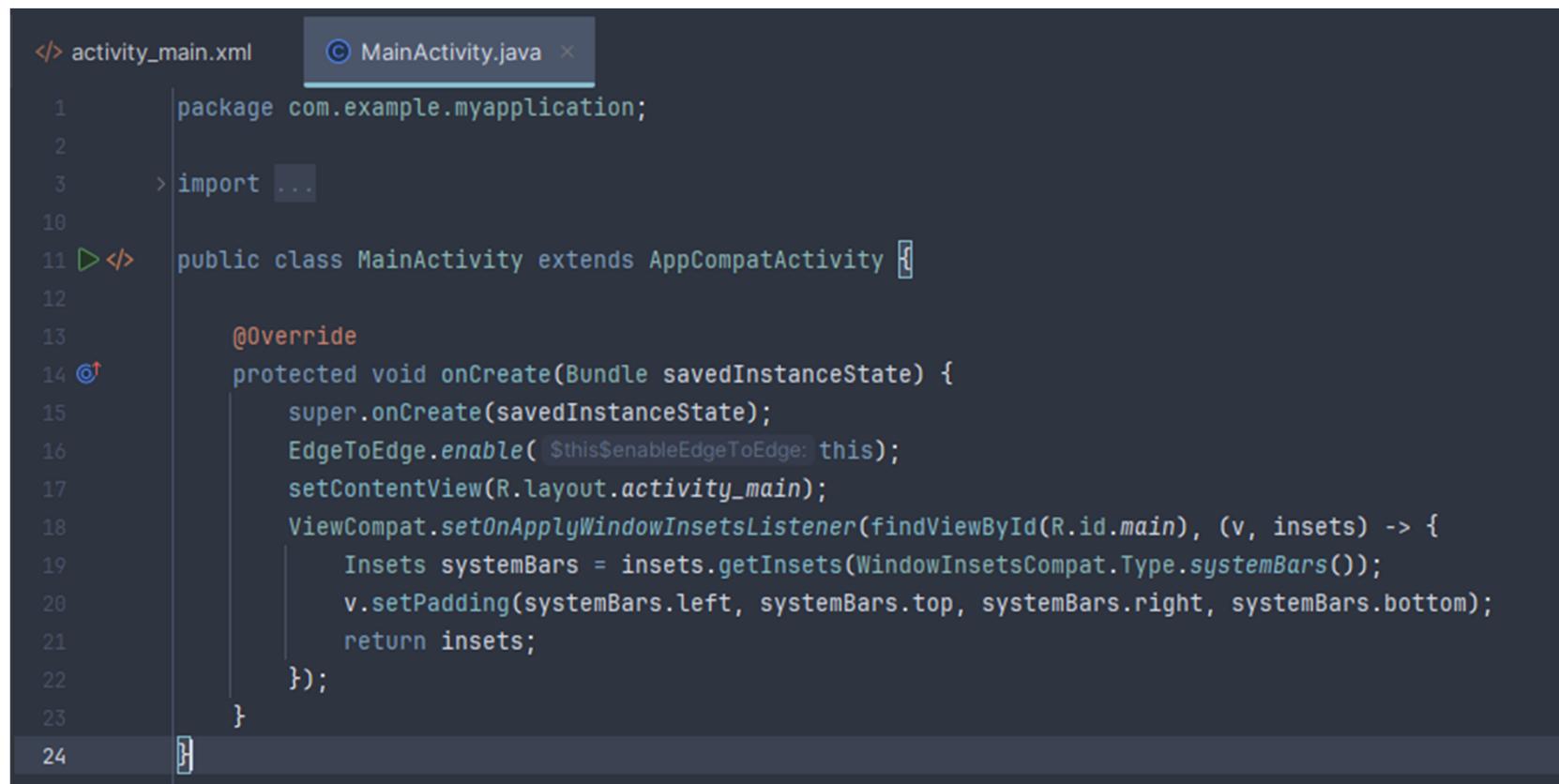
Primer App



Primer App



Primer App



The screenshot shows a code editor with two tabs: 'activity_main.xml' and 'MainActivity.java'. The 'MainActivity.java' tab is active, highlighted with a dark grey background and white text. The code in the editor is as follows:

```
</> activity_main.xml      ⚡ MainActivity.java ×  
1     package com.example.myapplication;  
2  
3     > import ...  
10  
11 <></> public class MainActivity extends AppCompatActivity {  
12  
13     @Override  
14     protected void onCreate(Bundle savedInstanceState) {  
15         super.onCreate(savedInstanceState);  
16         EdgeToEdge.enable( $this$enableEdgeToEdge: this);  
17         setContentView(R.layout.activity_main);  
18         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
19             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
20             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
21             return insets;  
22         });  
23     }  
24 }
```

Primer App

The screenshot shows the Android Studio interface with the XML layout editor on the right and the code editor on the left.

Code Editor (activity_main.xml):

```
<?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">

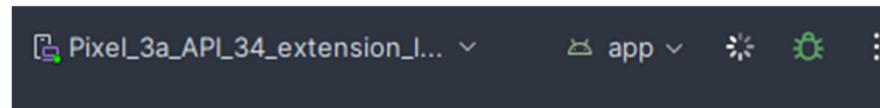
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello Umarino!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

XML Layout Editor:

The layout editor displays two views side-by-side. Both views show a blue constraint layout with a single text view centered both horizontally and vertically. The text view contains the text "Hello Umarino!". The layout uses constraints relative to the parent's edges.

Primer App



The screenshot displays the 'Build Output' tab in the Android Studio interface. The main pane shows the build log for 'MyApplication2'. The log indicates a successful build completed at 4/15/21 1 min, 33 sec, 796 ms. The log lists the following tasks:

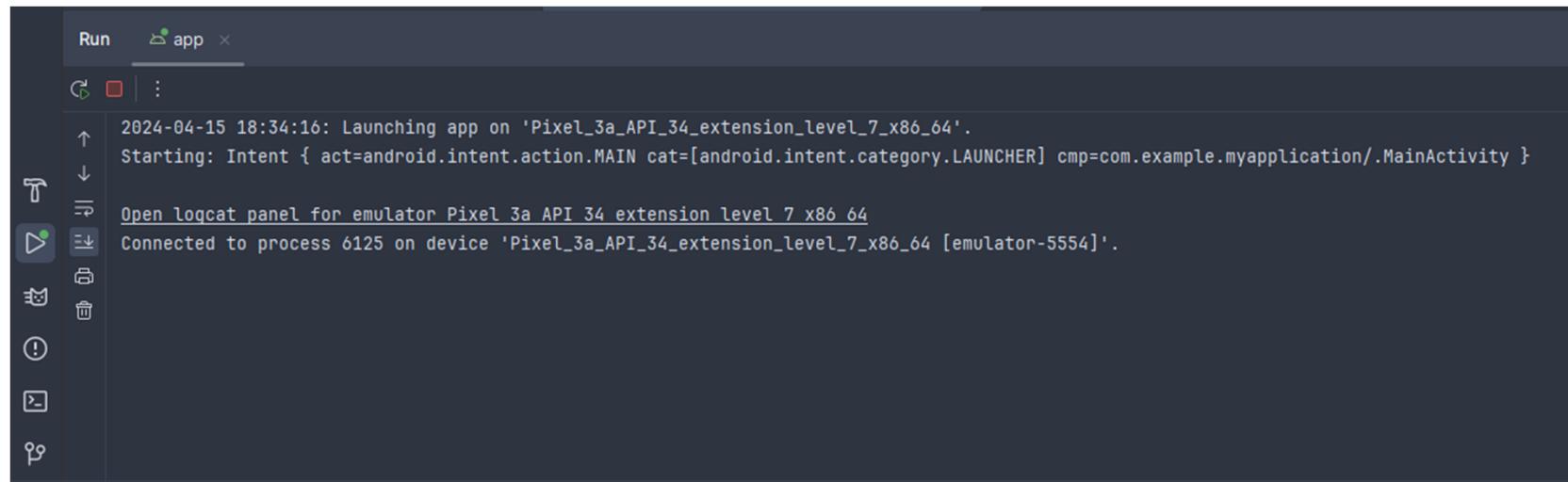
```
> Task :app:writeDebugSigningConfigVersions  
> Task :app:processDebugResources  
> Task :app:compileDebugJavaWithJavac  
> Task :app:dexBuilderDebug  
> Task :app:mergeProjectDexDebug  
> Task :app:packageDebug  
> Task :app:createDebugApkListingFileRedirect  
> Task :app:assembleDebug
```

The log concludes with:

```
BUILD SUCCESSFUL in 1m 33s  
31 actionable tasks: 31 executed
```

At the bottom of the log, there is a link: [Build Analyzer results available](#).

Primer App



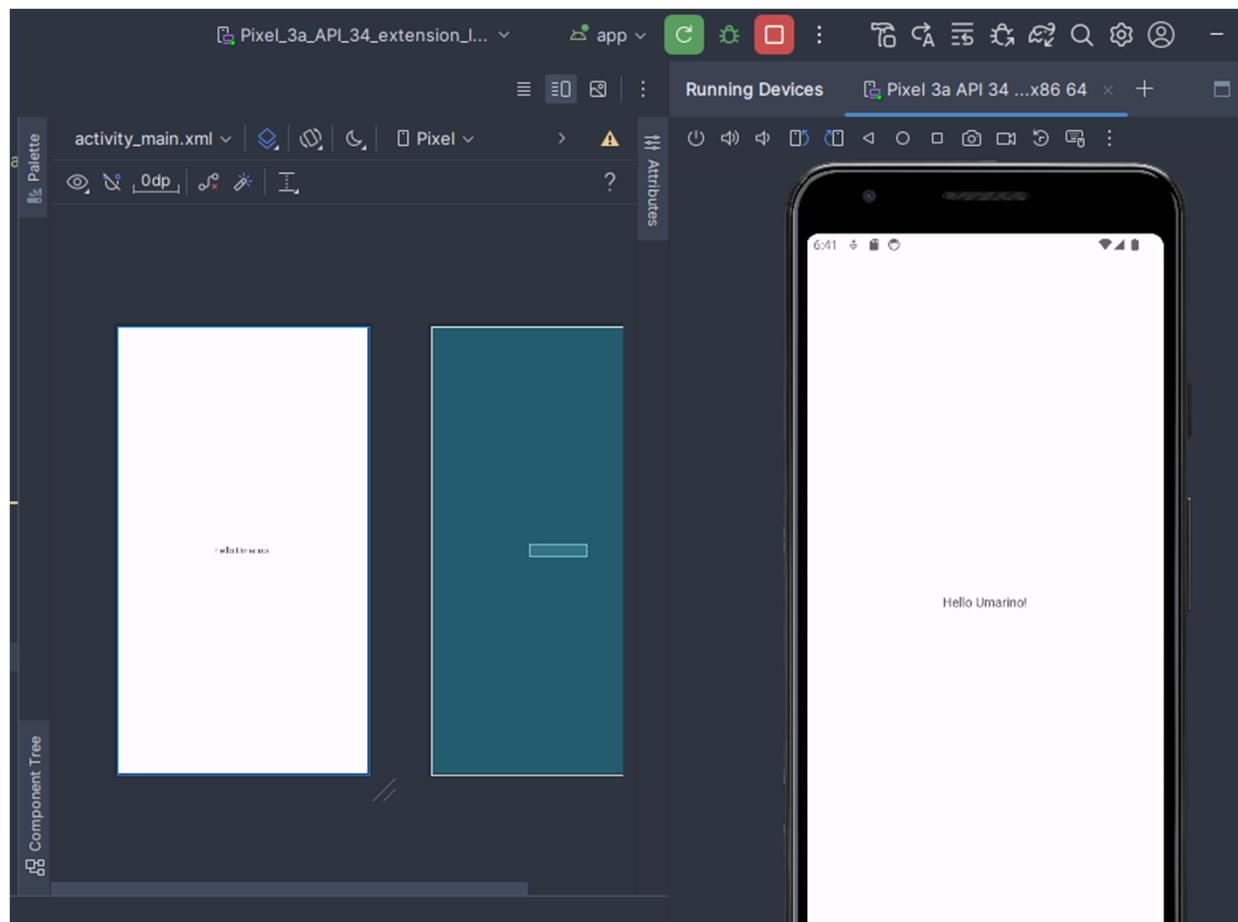
The screenshot shows the Android Studio interface with the 'Run' tab selected. The 'app' configuration is active. The log output window displays the following text:

```
2024-04-15 18:34:16: Launching app on 'Pixel_3a_API_34_extension_level_7_x86_64'.
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.myapplication/.MainActivity }

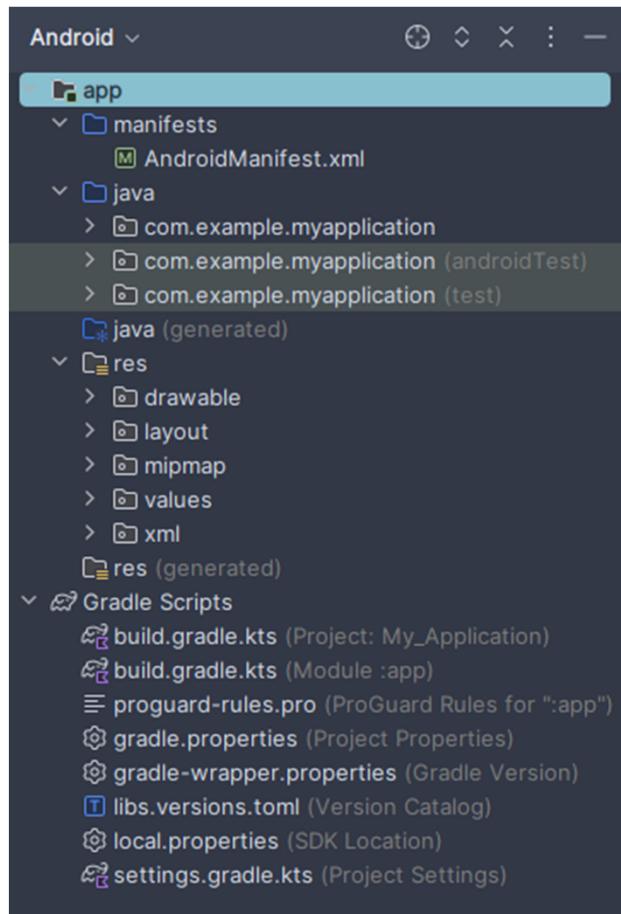
Open logcat panel for emulator Pixel_3a API 34 extension level 7 x86_64
Connected to process 6125 on device 'Pixel_3a_API_34_extension_level_7_x86_64 [emulator-5554]'.
```

The left sidebar contains various run configurations and tool icons.

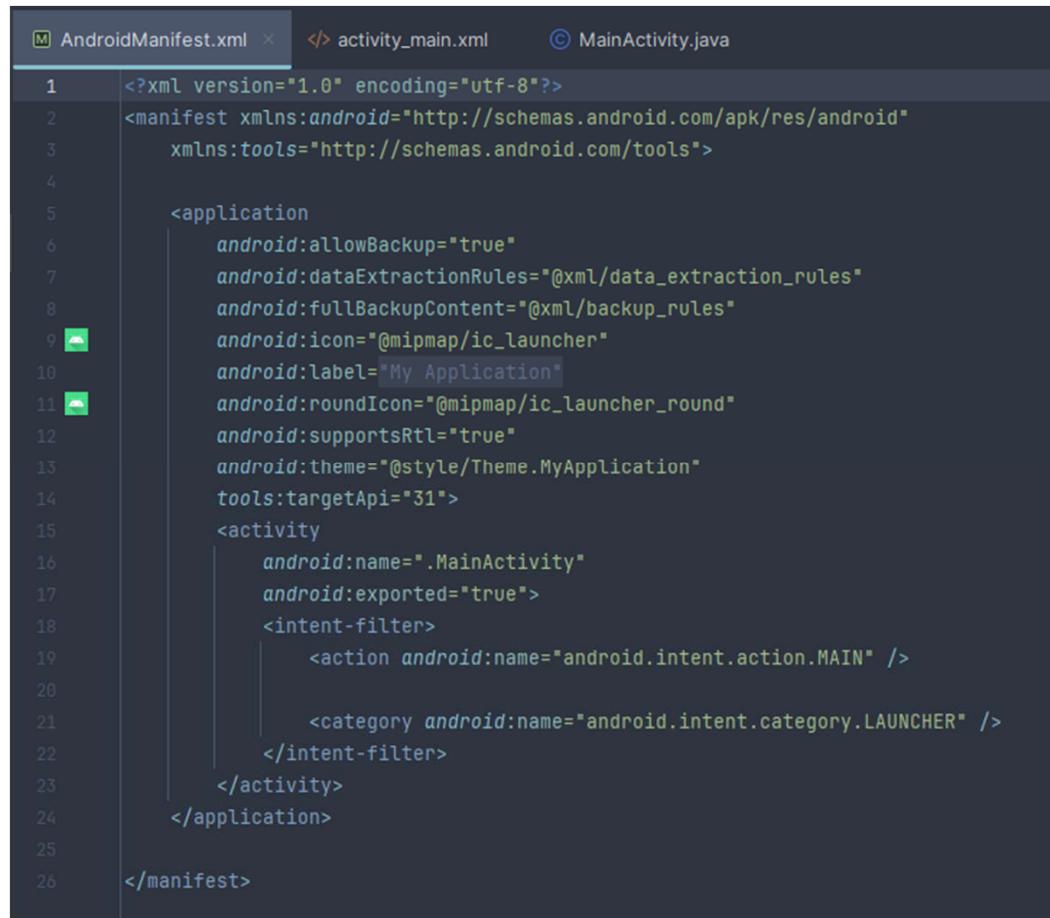
Primer App



Primer App



Primer App

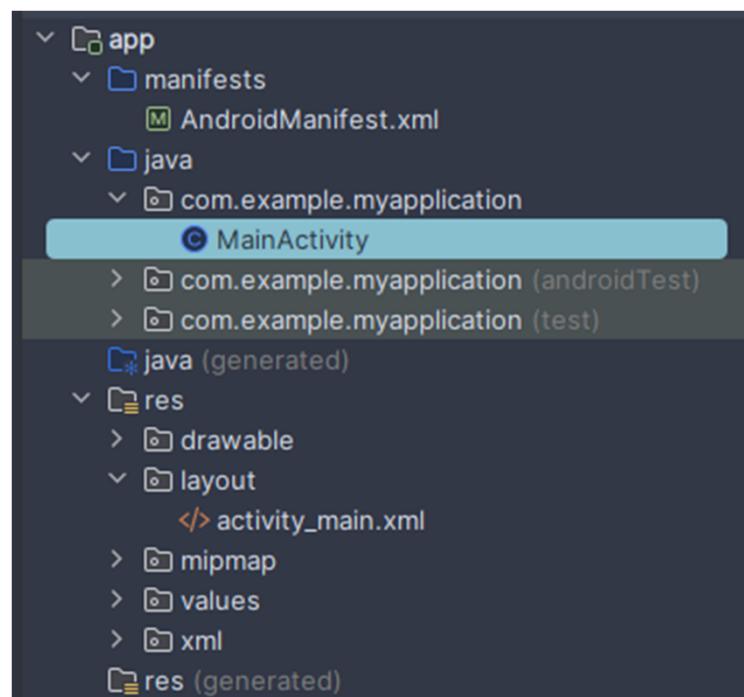


The screenshot shows a code editor with three tabs at the top: 'AndroidManifest.xml' (selected), 'activity_main.xml', and 'MainActivity.java'. The 'AndroidManifest.xml' tab contains the following XML code:

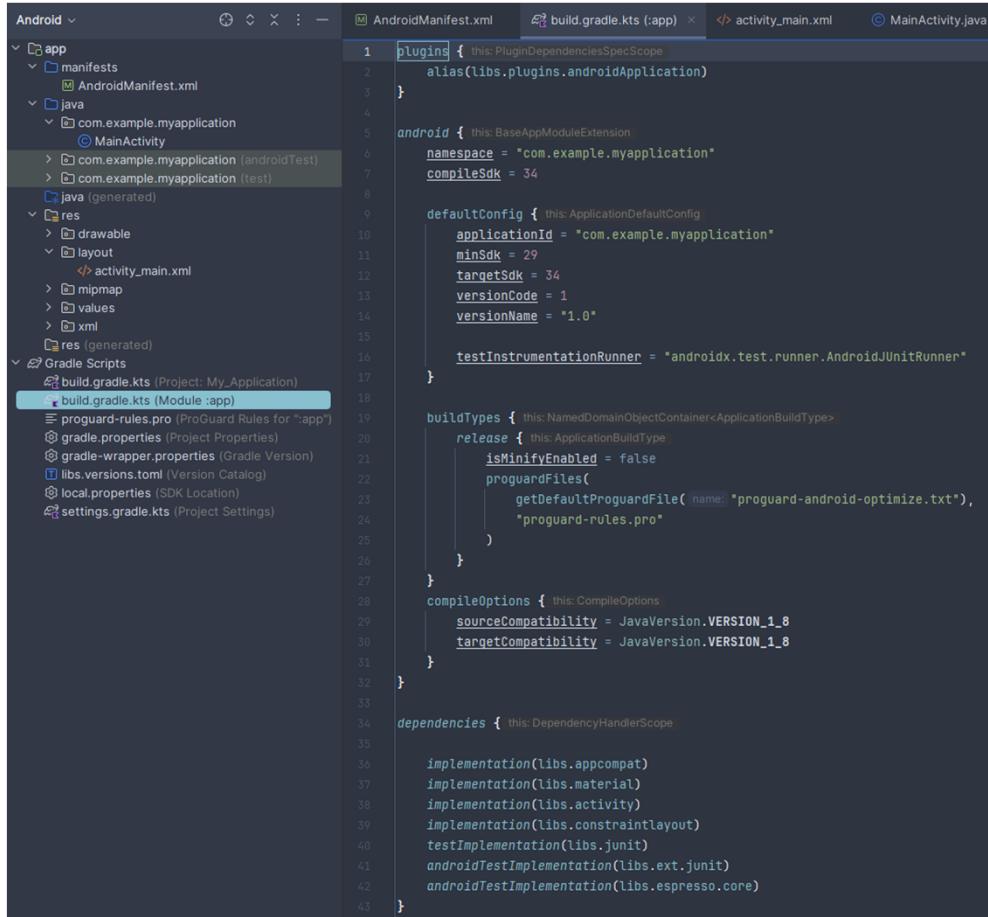
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Primer App



Primer App



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left pane displays the project structure under "Android". It includes the "app" module with sub-directories "manifests", "java", "res", and "Gradle Scripts". The "build.gradle.kts" file is selected in the "Gradle Scripts" section.
- Code Editor:** The right pane shows the content of the "build.gradle.kts" file. The code is a Kotlin script defining the app's configuration, dependencies, and build types.

```
plugins {
    alias(libs.plugins.androidApplication)
}

android {
    namespace = "com.example.myapplication"
    compileSdk = 34

    defaultConfig {
        applicationId = "com.example.myapplication"
        minSdk = 29
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

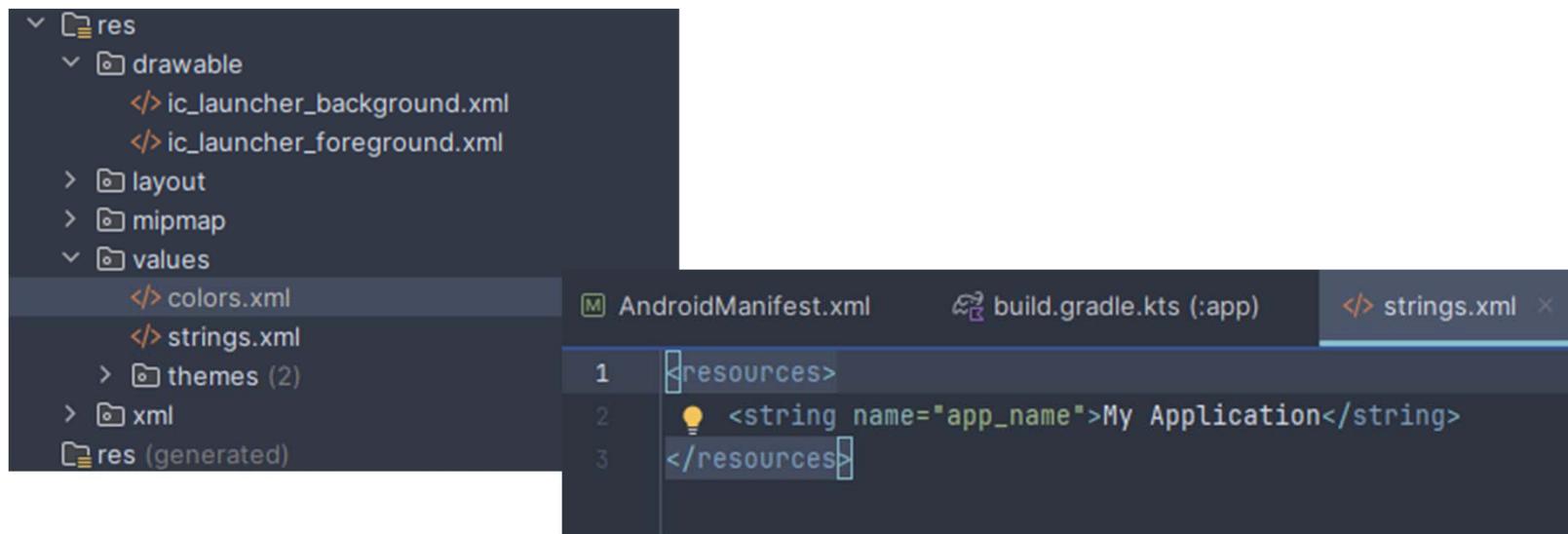
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}

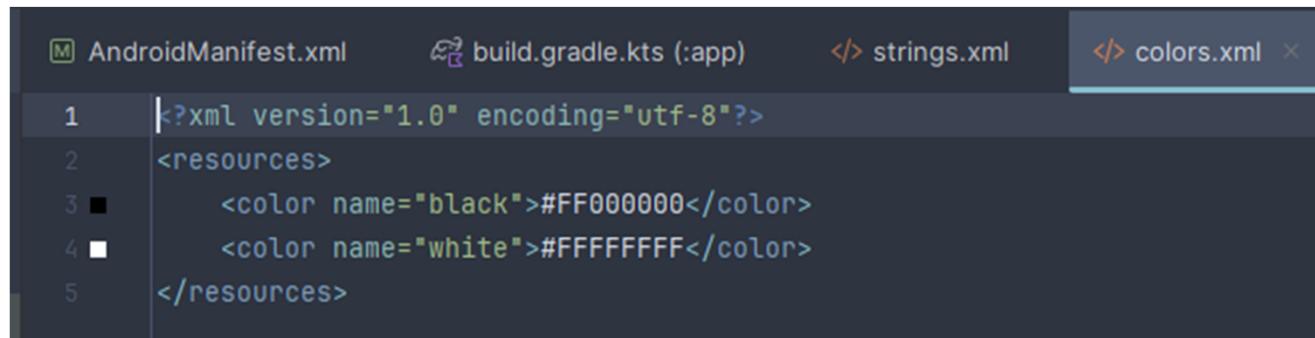
dependencies {
    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

Primer App



The screenshot shows the Android Studio interface. On the left, the project structure is displayed under the 'res' folder, which contains 'drawable', 'layout', 'mipmap', and 'values'. Inside 'values', there are files for 'colors.xml' and 'strings.xml', along with a 'themes' folder containing two items. On the right, the code editor shows the 'AndroidManifest.xml' file with the following content:

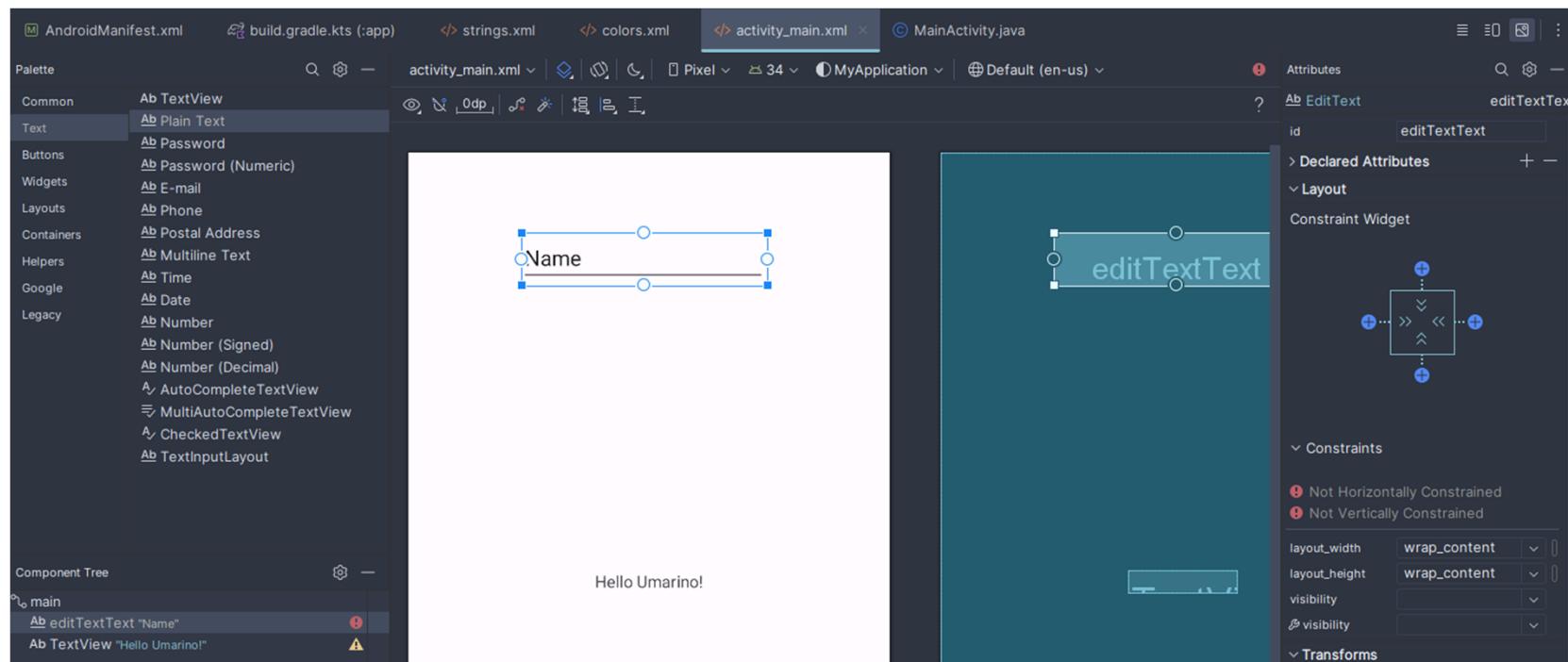
```
<resources>
    <string name="app_name">My Application</string>
</resources>
```



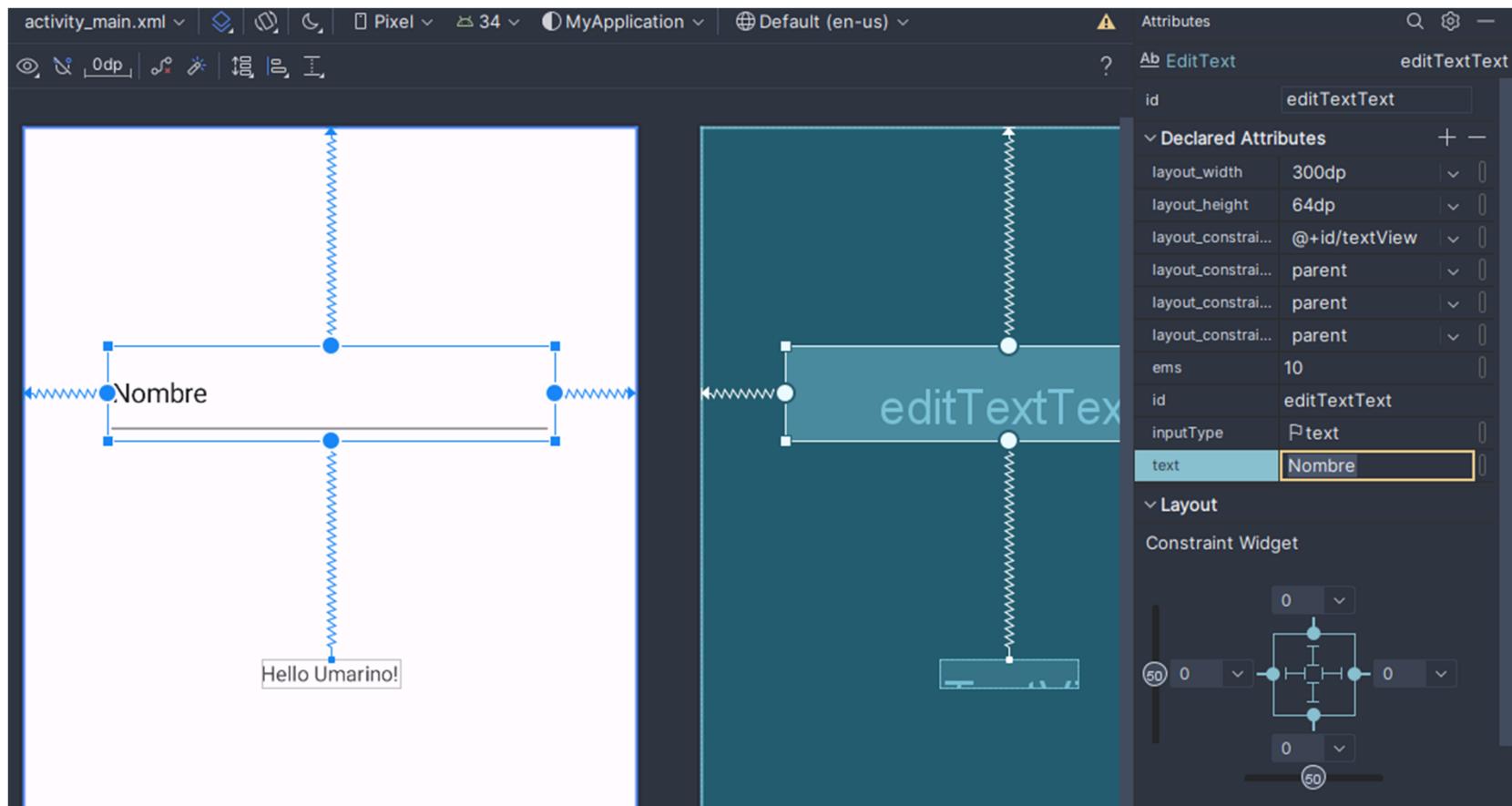
The code editor is focused on the 'colors.xml' file, which contains the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>
</resources>
```

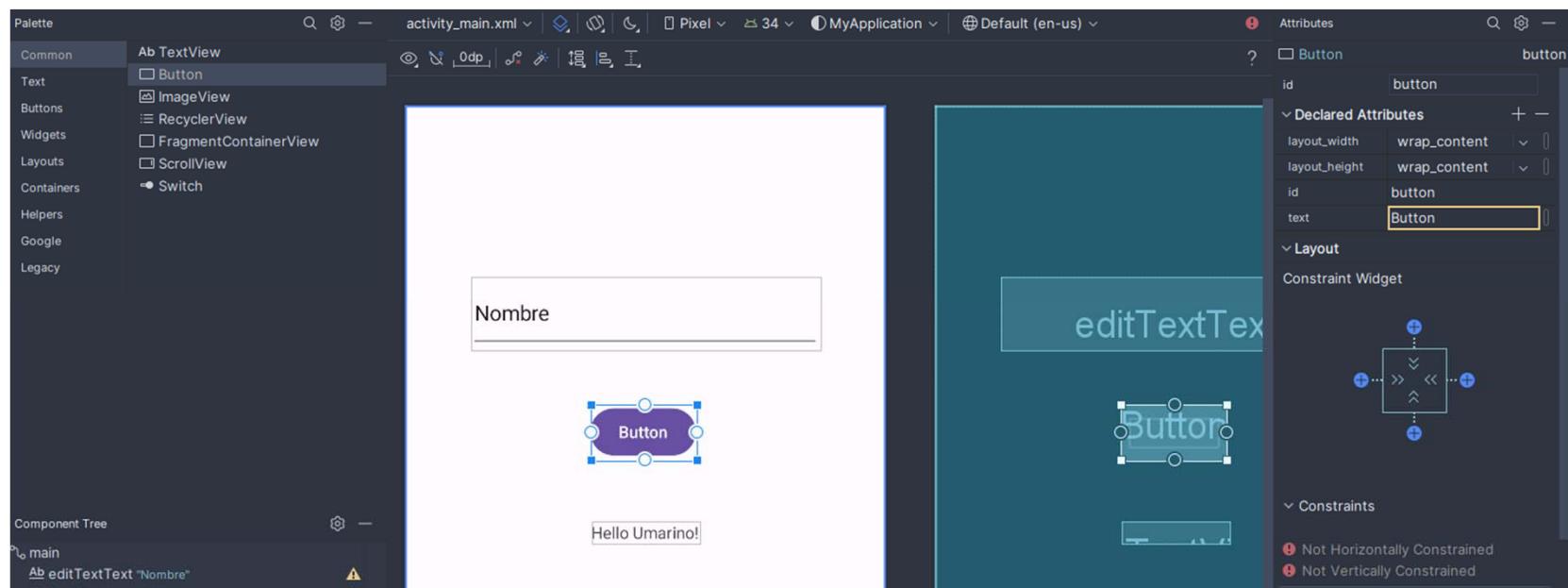
Segunda APP



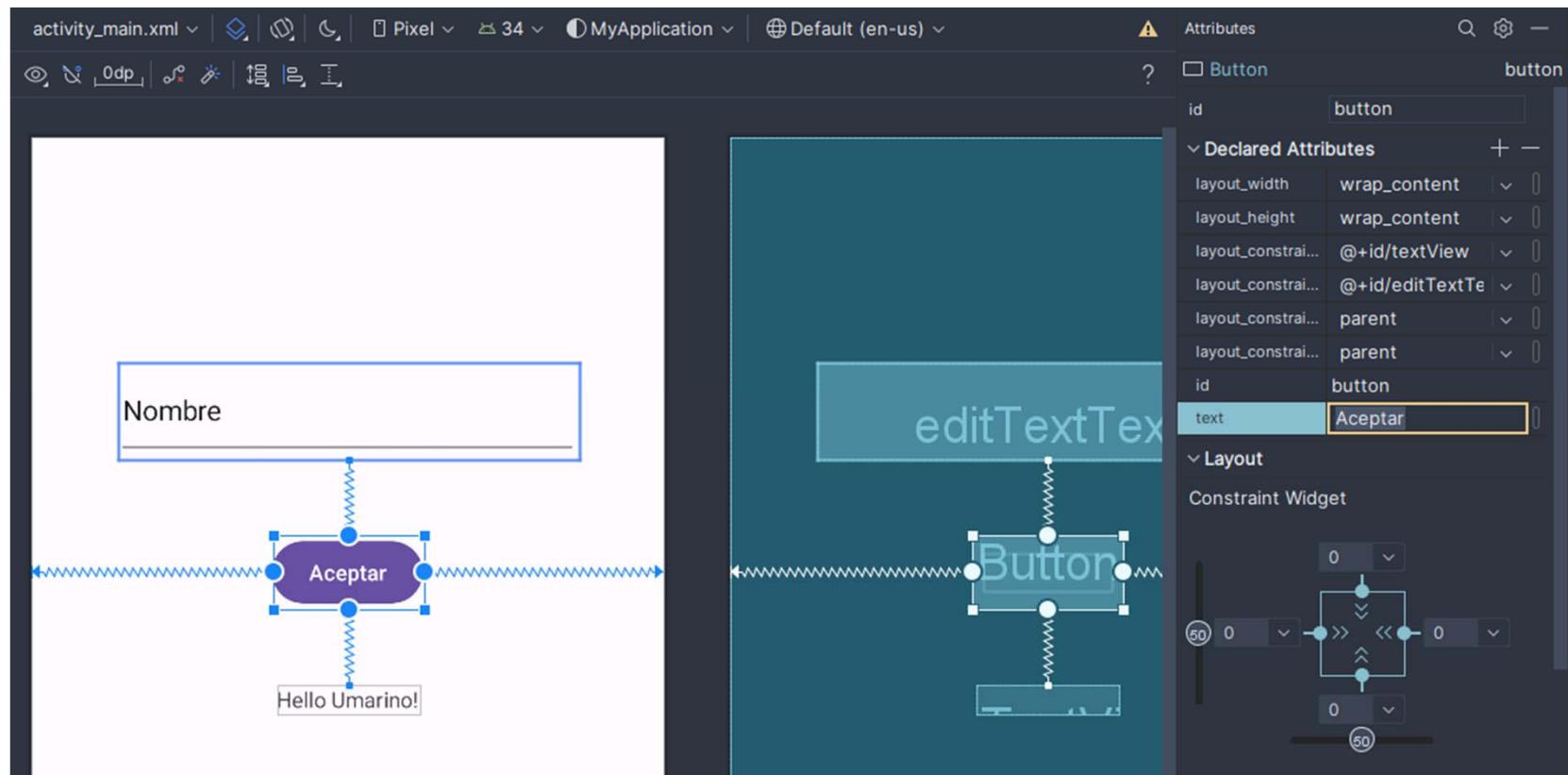
Segunda APP



Segunda APP



Segunda APP



Segunda APP

```
public class MainActivity extends AppCompatActivity {

    2 usages
    public TextInputEditText texto_nombre;
    2 usages
    public Button aceptar;
    2 usages
    public TextView mensaje;

    @SuppressLint("WrongViewCast")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);

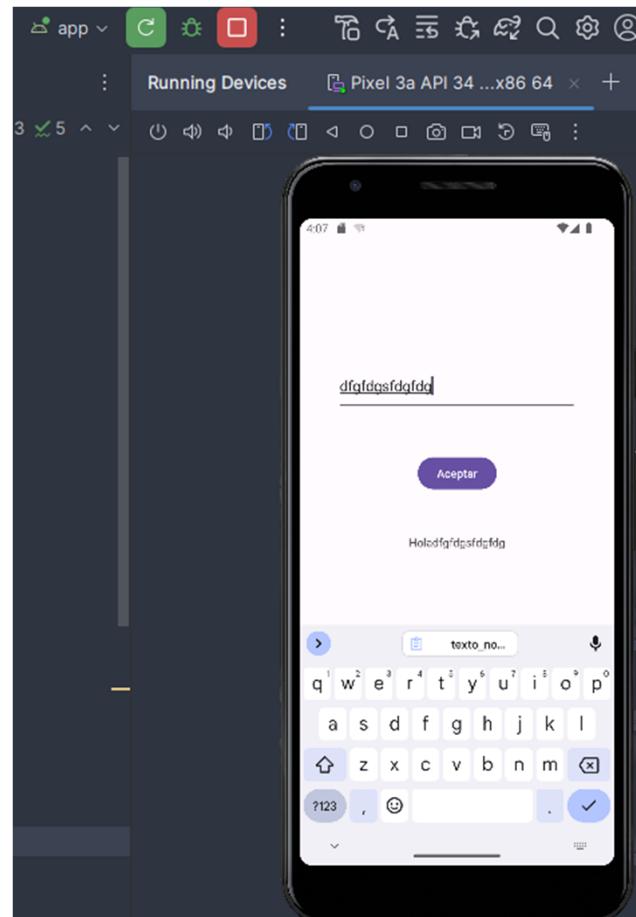
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);

            texto_nombre = v.findViewById(R.id.editTextText);
            aceptar = v.findViewById(R.id.button);
            mensaje = v.findViewById(R.id.textView);

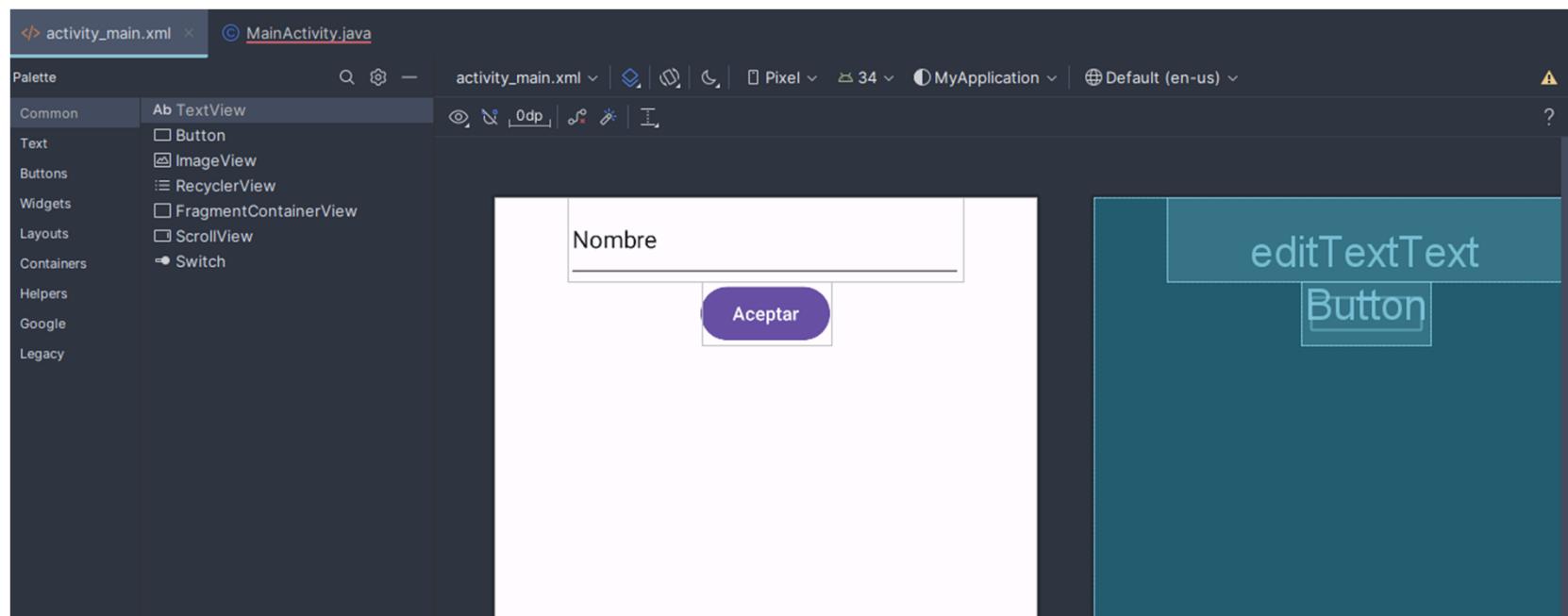
            aceptar.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    mensaje.setText("Hola"+ texto_nombre.getText());
                }
            });
        });

        return insets;
    }
}
```

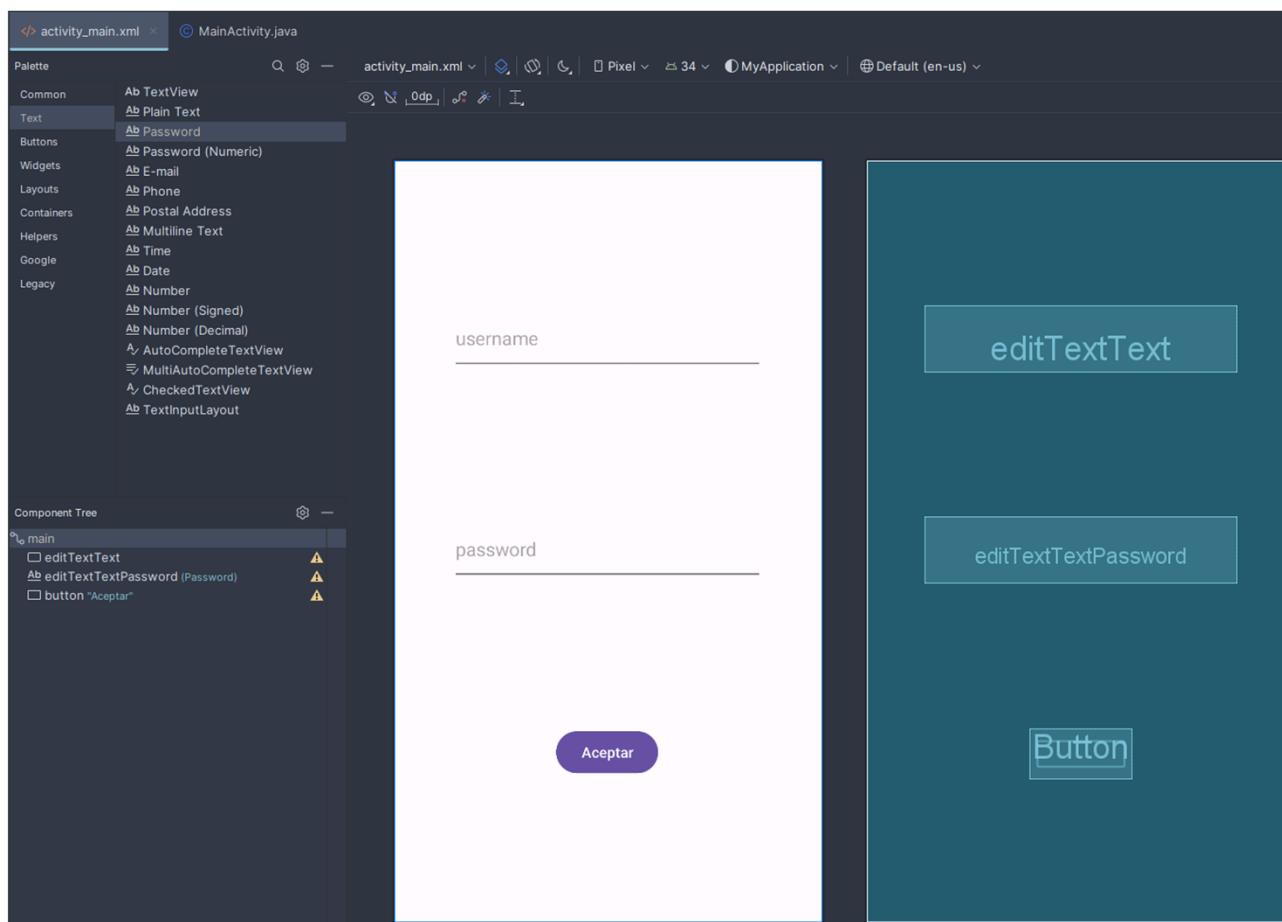
Segunda APP



Tercera APP



Tercera APP



Tercera APP

```
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/editTextText"
    android:layout_width="300dp"
    android:layout_height="64dp"
    android:ems="10"
    android:hint="username"
    android:inputType="text"
    app:layout_constraintBottom_toTopOf="@+id/editTextTextPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.275" />

<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="300dp"
    android:layout_height="64dp"
    android:ems="10"
    android:hint="password"
    android:inputType="textPassword"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextText" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Aceptar"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPassword" />
```

Tercera APP

```
public class MainActivity extends AppCompatActivity {

    2 usages
    public TextInputEditText username;
    2 usages
    public EditText clave;
    ! 2 usages
    public Button aceptar;

    @SuppressLint("WrongViewCast")
    @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);

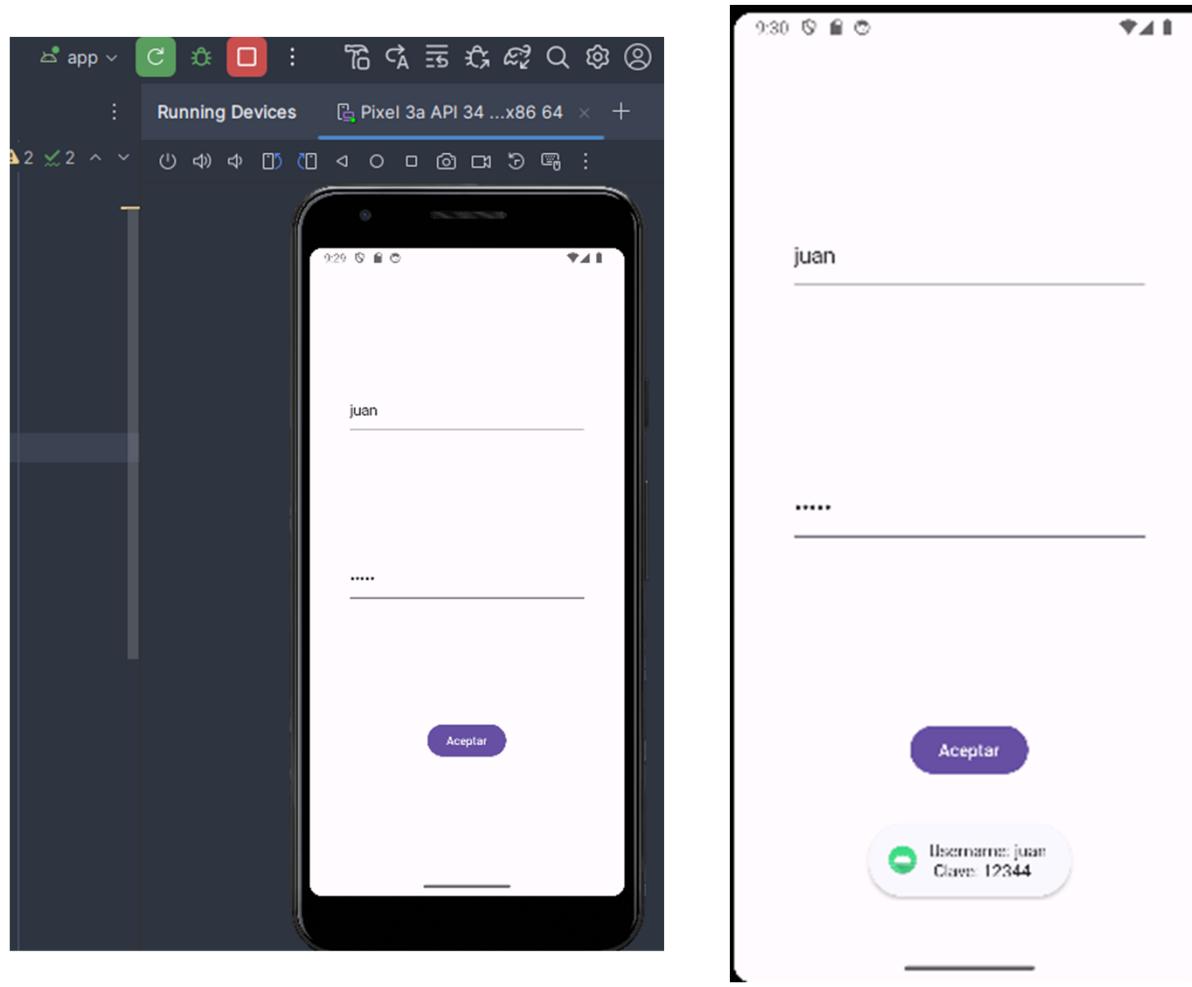
            username = v.findViewById(R.id.editTextText);
            clave = v.findViewById(R.id.editTextTextPassword);
            aceptar = v.findViewById(R.id.button);

            aceptar.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    String datos = "Username: " + username.getText() + "\n Clave: " + clave.getText();
                    Toast.makeText(getApplicationContext(),datos, Toast.LENGTH_SHORT).show();
                }
            });
        });

        return insets;
    }

}
```

Tercera APP



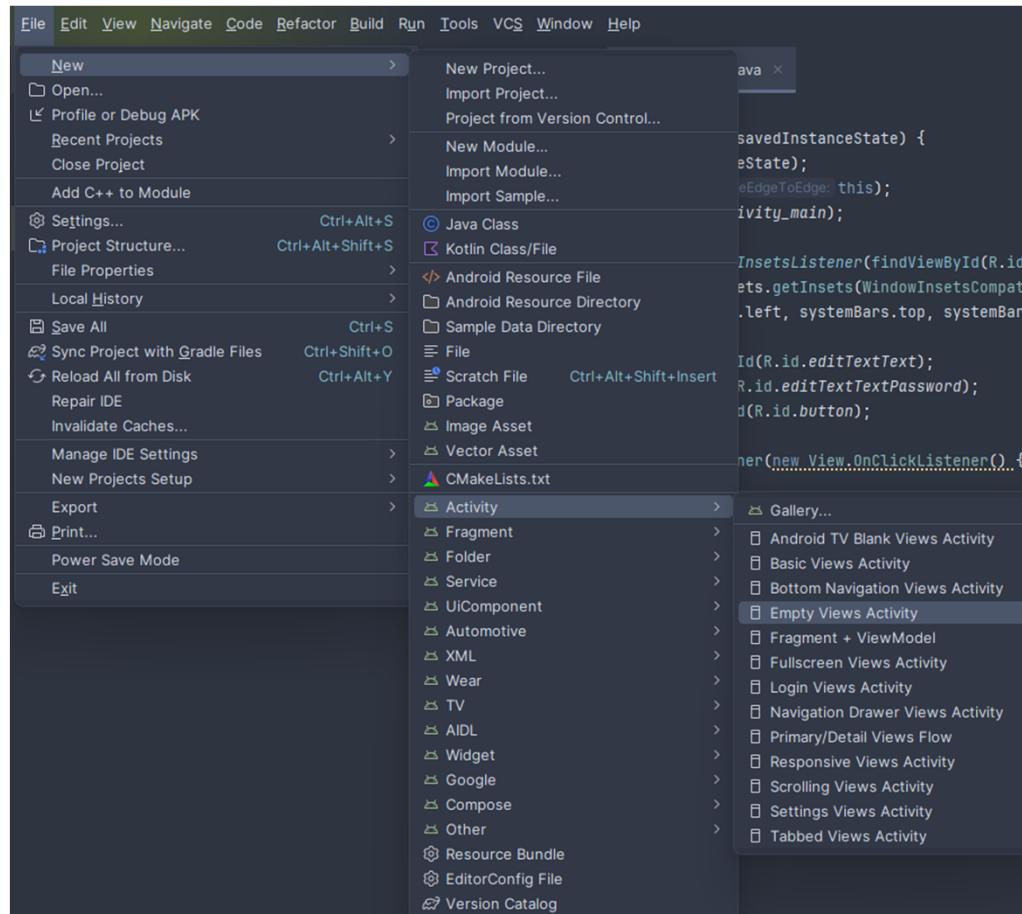
Tercera APP

```
aceptar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String datos = "Username: " + username.getText() + "\n Clave: " + clave.getText();
        Toast.makeText(getApplicationContext(), datos, Toast.LENGTH_SHORT).show();

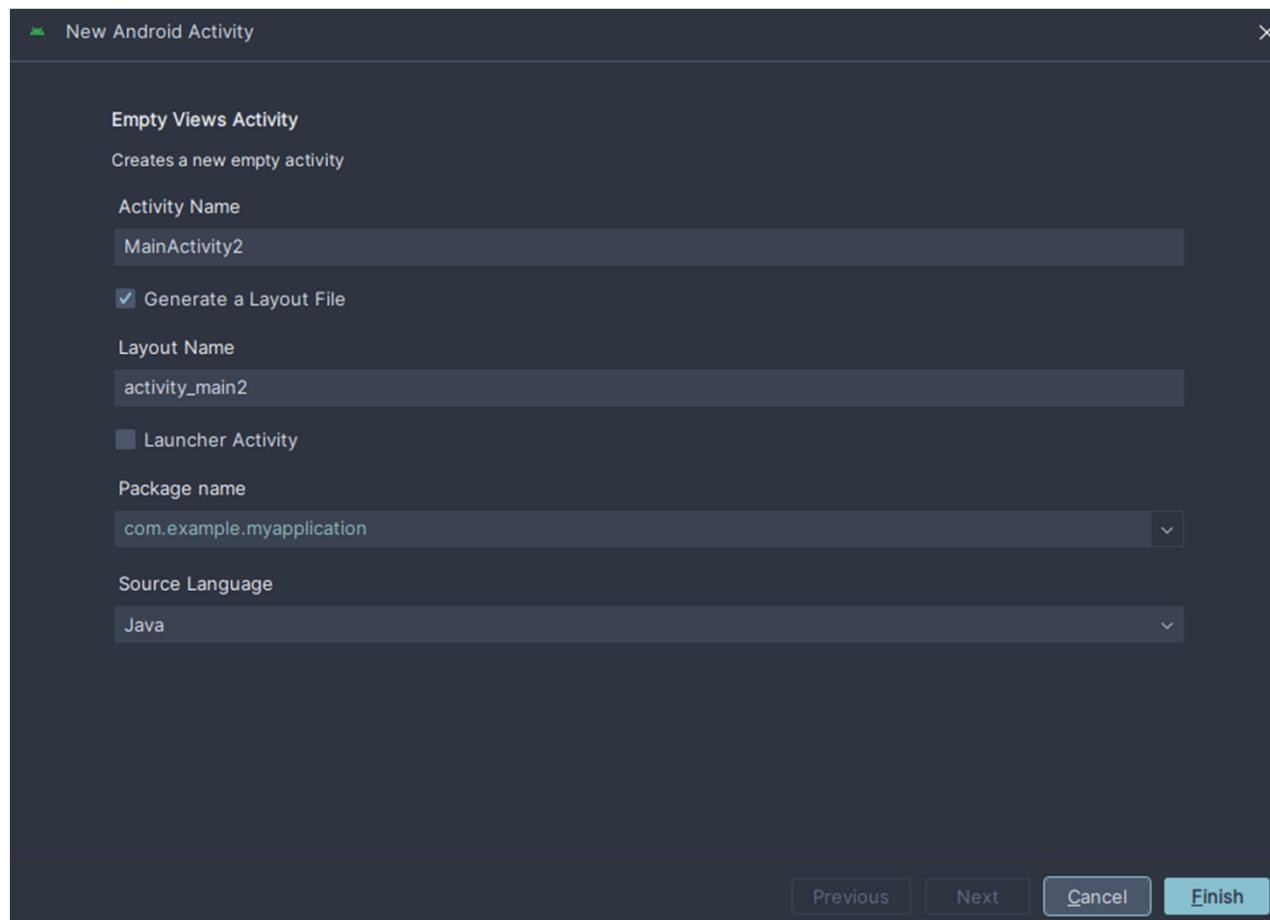
        String name = String.valueOf(username.getText());
        String pass = String.valueOf(clave.getText());

        if(name.equals("umarino") && pass.equals("1234")){
            Toast.makeText(getApplicationContext(), text: "Acceso Autorizado :)", Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(getApplicationContext(), text: "Datos incorrectos :( ", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Tercera APP



Tercera APP

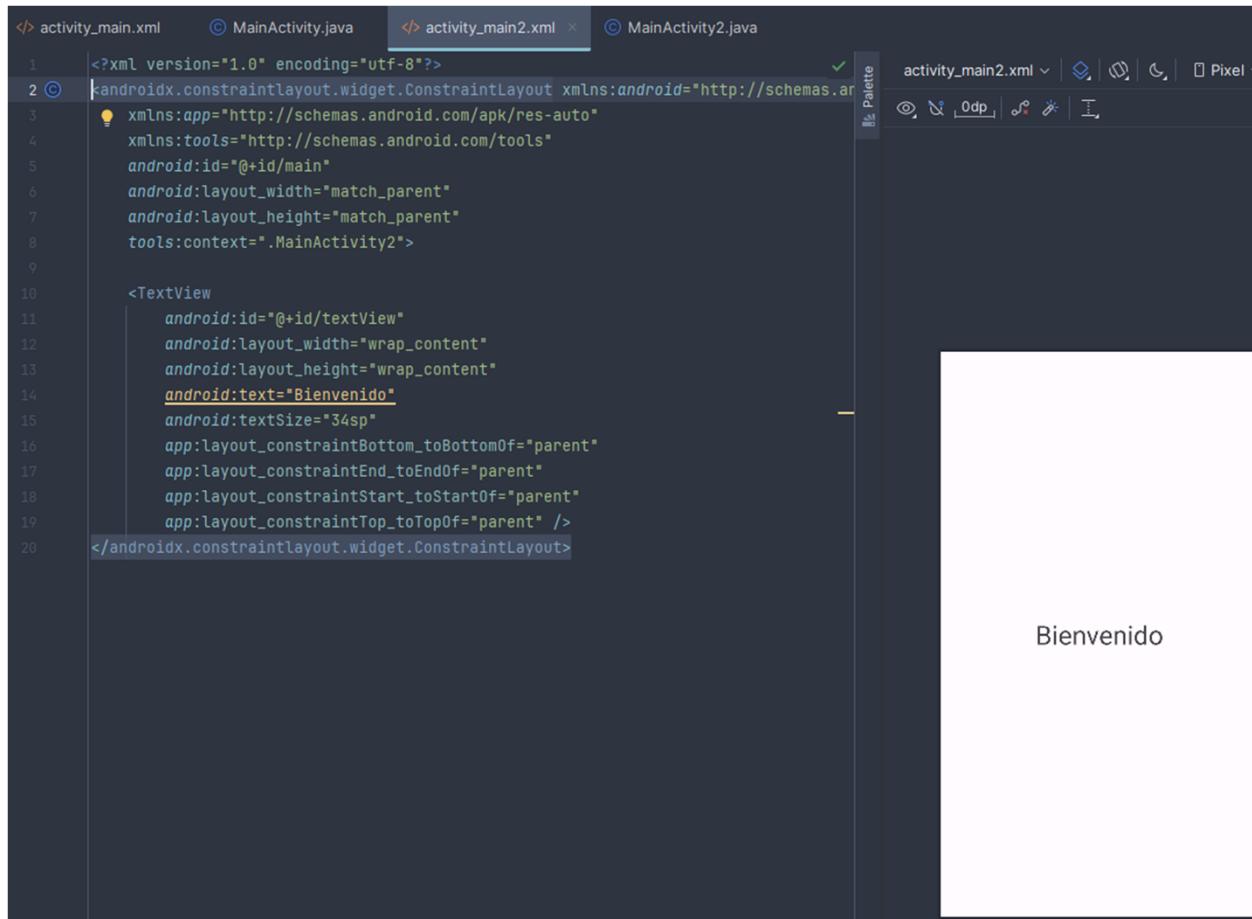


Tercera APP

The screenshot shows the Android Studio interface. On the left is the Project Navigational Bar, which includes sections for app, manifests, java, res, and Gradle Scripts. The Java section is expanded, showing packages com.example.myapplication and com.example.myapplication.generated. Inside com.example.myapplication, there are files MainActivity and MainActivity2, with MainActivity2 currently selected and highlighted with a blue background. The right side of the screen is the code editor, displaying the MainActivity2.java file. The code is as follows:

```
1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity2 extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
11        setContentView(R.layout.activity_main2);
12        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
13            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
14            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
15            return insets;
16        });
17    }
18}
19
20}
21
22}
23}
24}
```

Tercera APP



The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Shows tabs for activity_main.xml, MainActivity.java, activity_main2.xml (selected), and MainActivity2.java.
- Left Panel (Code View):** Displays the XML code for activity_main2.xml. The code defines a ConstraintLayout with a central TextView containing the text "Bienvenido".

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

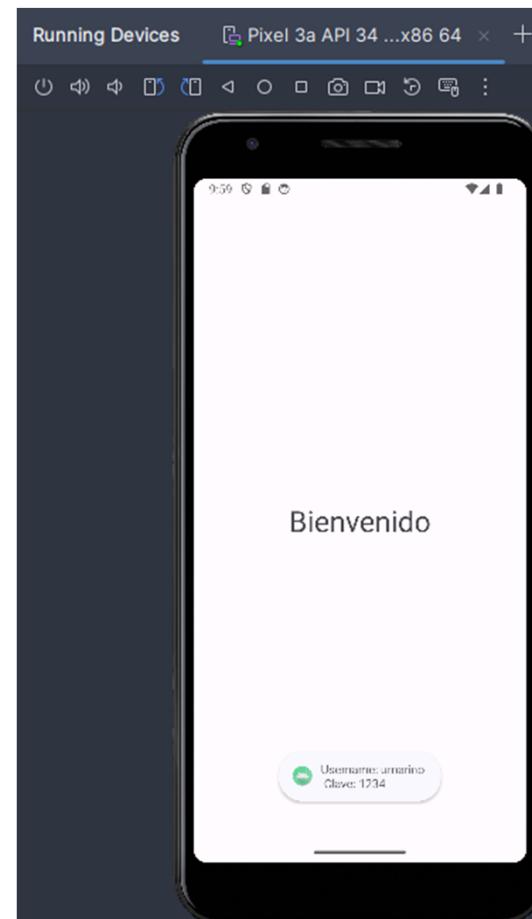
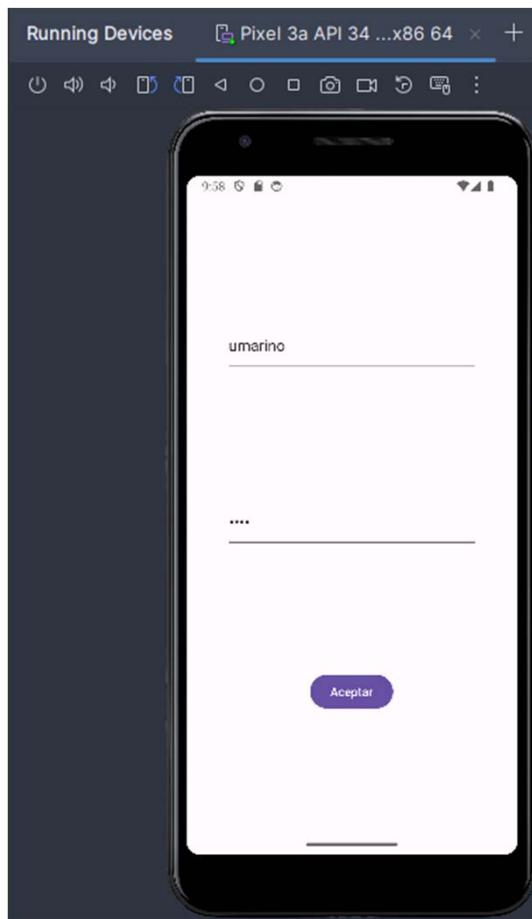
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bienvenido"
        android:textSize="34sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Right Panel (Preview View):** Shows a preview of the Android application's user interface. It features a single TextView centered on the screen with the text "Bienvenido".

Tercera APP

```
if(name.equals("umarino") && pass.equals("1234")){
    Toast.makeText(getApplicationContext(), text: "Acceso Autorizado :)", Toast.LENGTH_SHORT).show();
    Intent intent = new Intent( packageContext: MainActivity.this,MainActivity2.class);
    startActivity(intent);
}else{
    Toast.makeText(getApplicationContext(), text: "Datos incorrectos :( ", Toast.LENGTH_SHORT).show();
}
```

Tercera APP



Tercera APP

```
3 usages
public TextInputEditText username;
3 usages
public EditText clave;
2 usages
public Button aceptar;
2 usages
public static final String dato_name = "username";
```

```
if(name.equals("umarino") && pass.equals("1234")){
    Toast.makeText(getApplicationContext(), text: "Acceso Autorizado :)", Toast.LENGTH_SHORT).show();

    Intent intent = new Intent( packageContext: MainActivity.this,MainActivity2.class);
    intent.putExtra(dato_name, name);
    startActivity(intent);

} else{
    Toast.makeText(getApplicationContext(), text: "Datos incorrectos :( ", Toast.LENGTH_SHORT).show();
}
```

Tercera APP

```
public class MainActivity2 extends AppCompatActivity {

    2 usages
    public TextView saludo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main2);
        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);

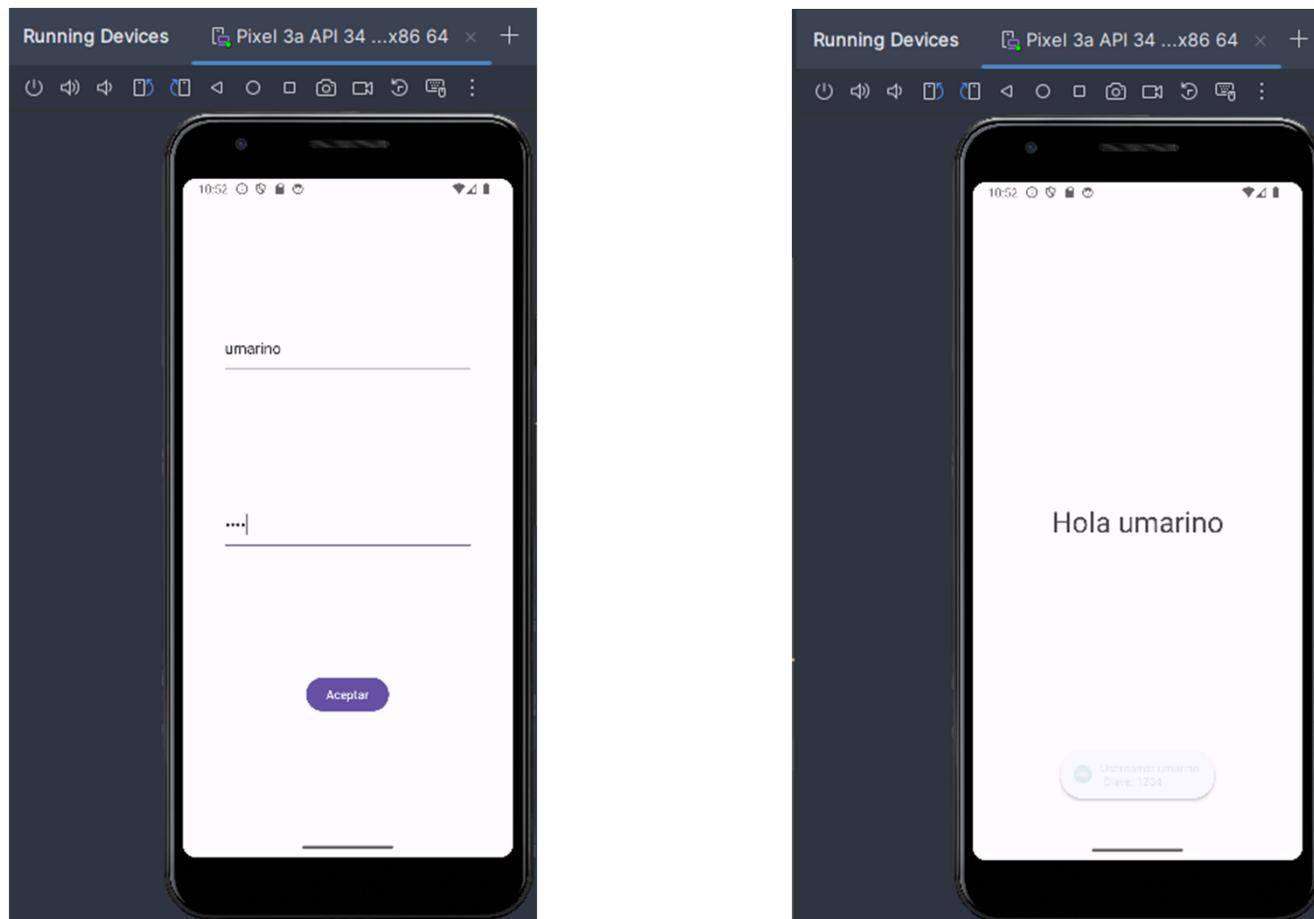
            saludo = v.findViewById(R.id.textView);

            Intent intent = getIntent();
            String dato = intent.getStringExtra(MainActivity.dato_name);

            saludo.setText("Hola "+dato);

            return insets;
        });
    }
}
```

Tercera APP



Cuarta APP

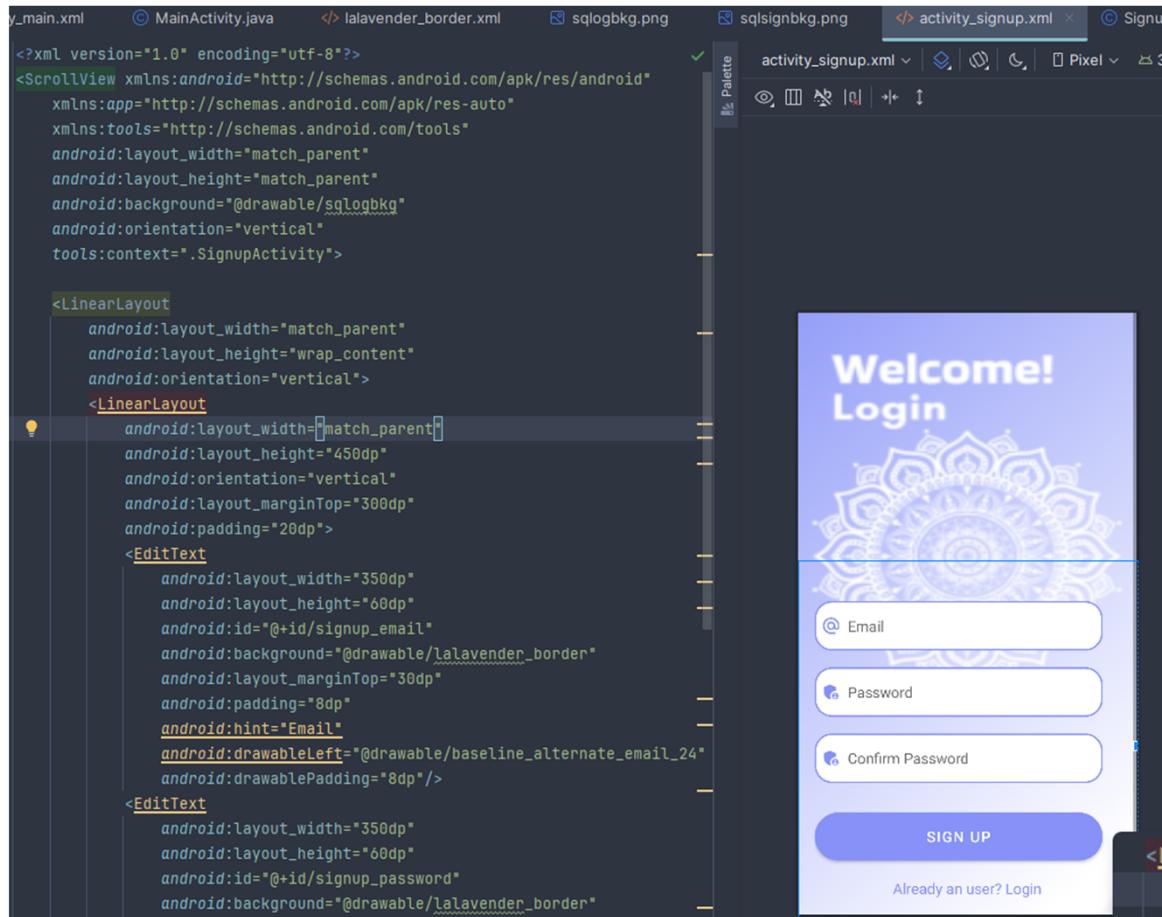
```
buildTypes { this: NamedDomainObjectContainer<ApplicationBuildType>
    release { this: ApplicationBuildType
        isMinifyEnabled = false
        proguardFiles(
            getDefaultProguardFile( name: "proguard-android-optimize.txt"),
            "proguard-rules.pro"
        )
    }
}

buildFeatures{ this: ApplicationBuildFeatures
    viewBinding = true
}

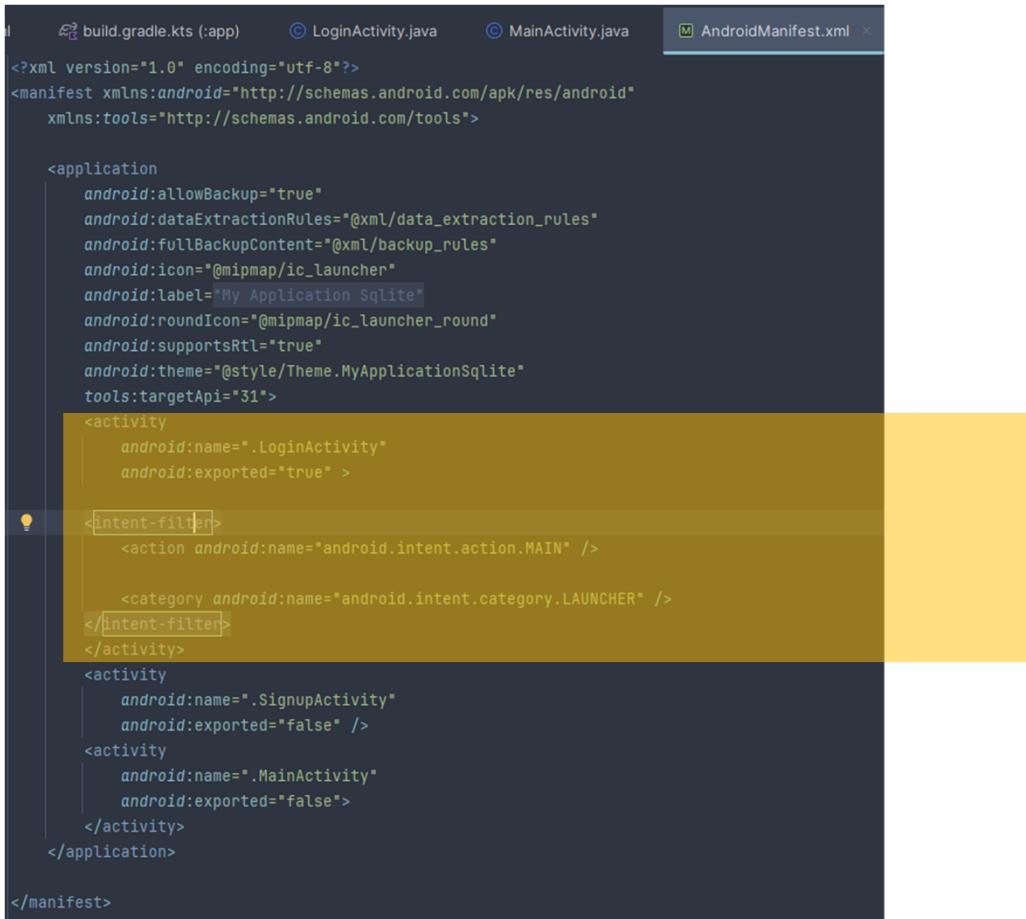
compileOptions { this: CompileOptions
    sourceCompatibility = JavaVersion.VERSION_1_8
    targetCompatibility = JavaVersion.VERSION_1_8
}
```

Agregar '='

Cuarta APP



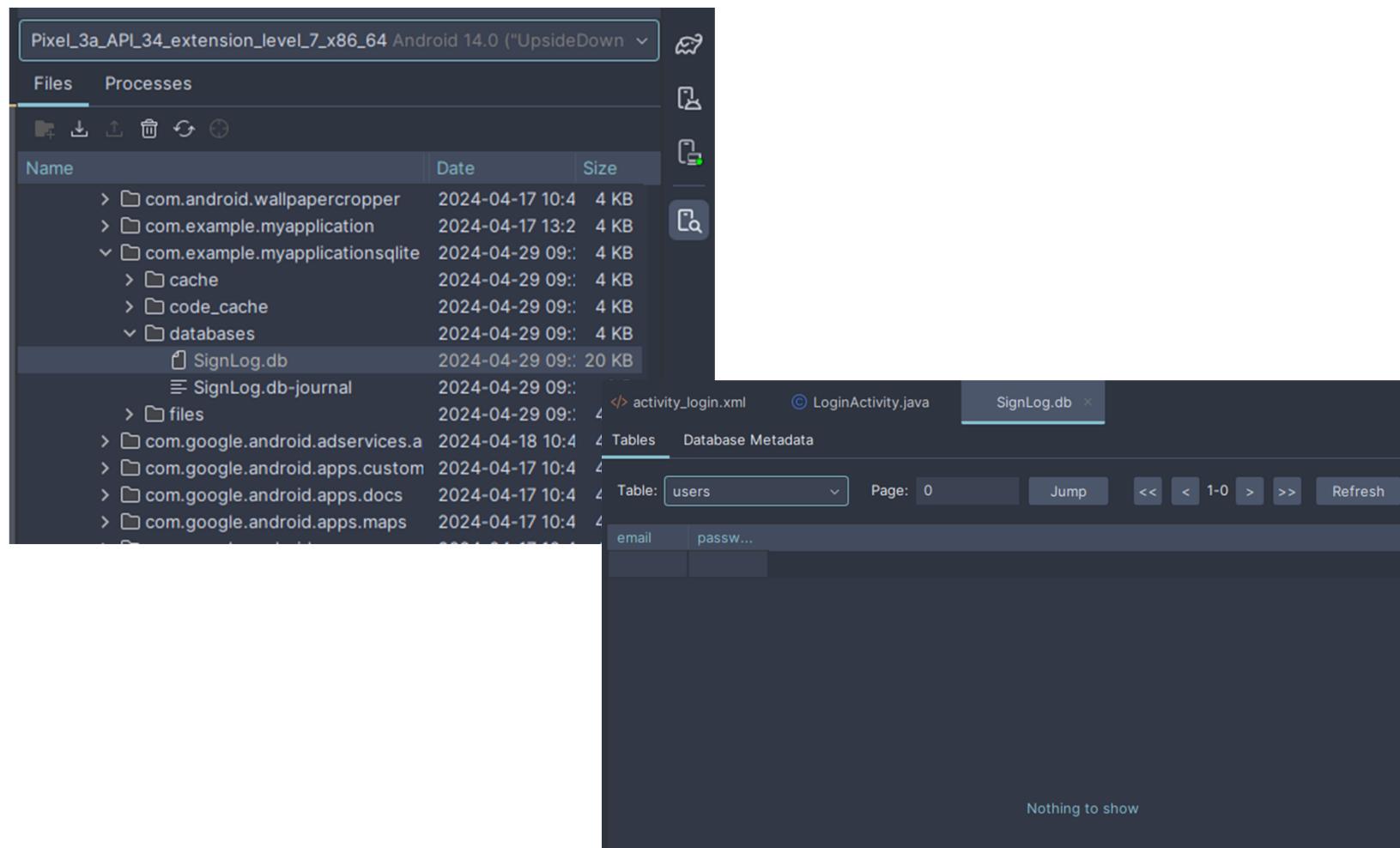
Cuarta APP



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application Sqlite"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplicationSqlite"
        tools:targetApi="31">
        <activity
            android:name=".LoginActivity"
            android:exported="true" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SignupActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="false">
        </activity>
    </application>
</manifest>
```

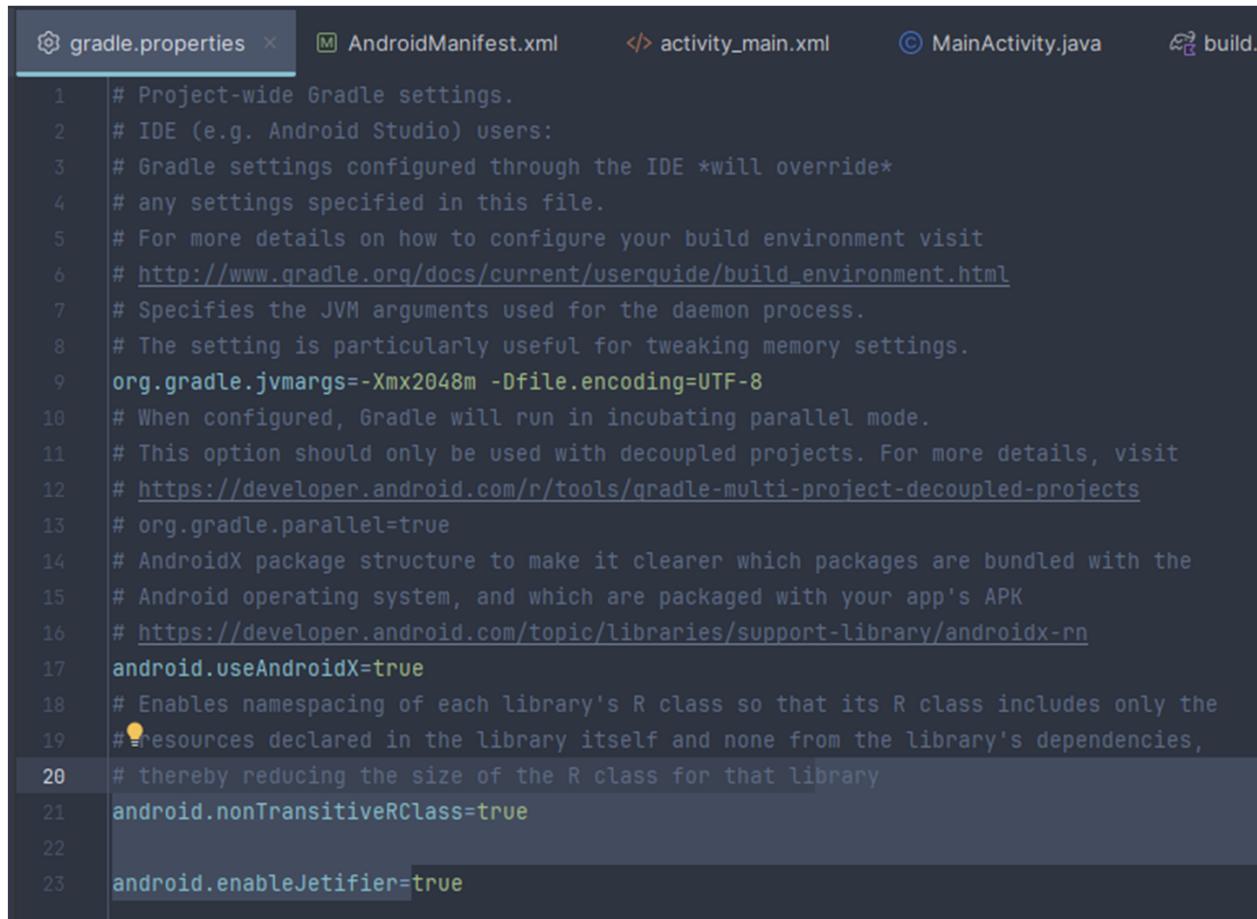
Cuarta APP



APP QR

```
dependencies { this: DependencyHandlerScope  
  
    implementation(libs.appcompat)  
    implementation(libs.material)  
    implementation(libs.activity)  
    implementation(libs.constraintlayout)  
    testImplementation(libs.junit)  
    androidTestImplementation(libs.ext.junit)  
    androidTestImplementation(libs.espresso.core)  
    implementation(libs.zxing.android.embedded)  
}
```

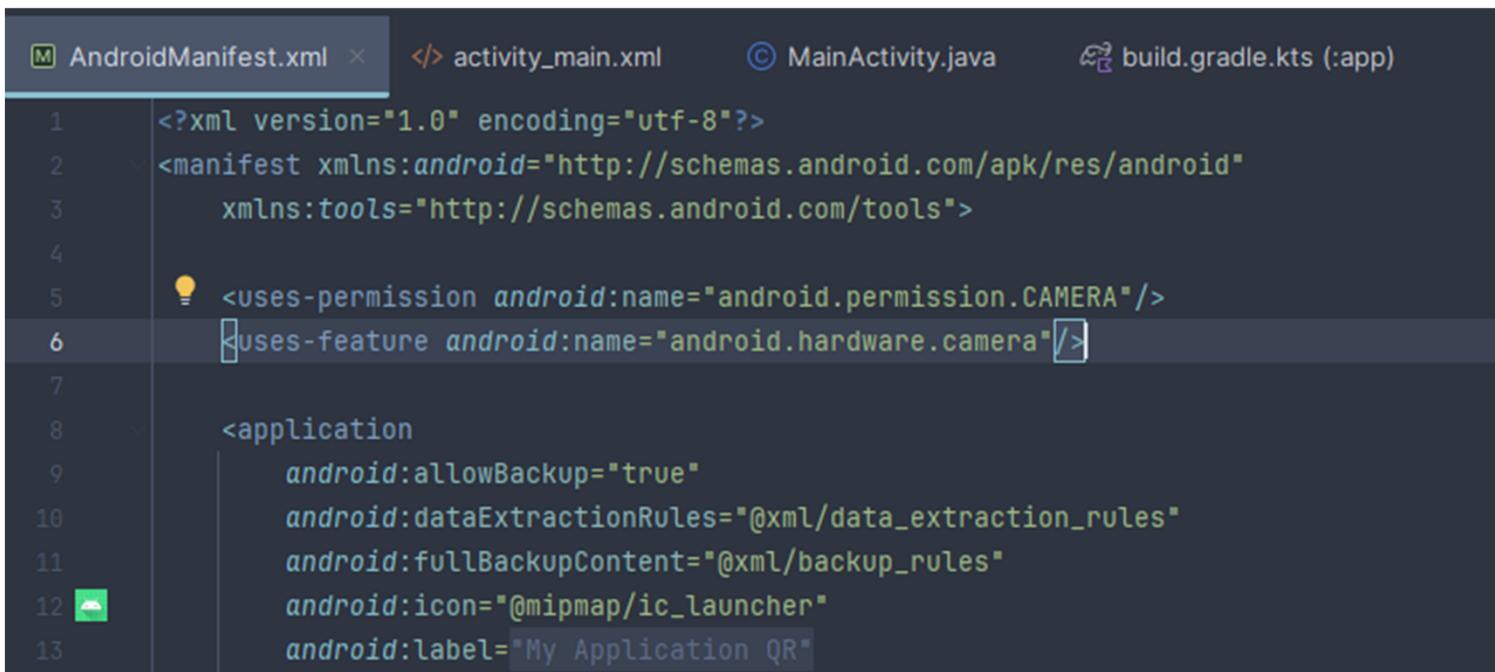
APP QR



The screenshot shows the Android Studio interface with the 'gradle.properties' file open. The file contains configuration settings for the Gradle build system. The code is numbered from 1 to 23. Lines 1 through 19 are comments explaining various Gradle settings like JVM arguments, parallel mode, and AndroidX usage. Lines 20 through 23 are specific configurations for the R class generation process.

```
1 # Project-wide Gradle settings.
2 # IDE (e.g. Android Studio) users:
3 # Gradle settings configured through the IDE *will override*
4 # any settings specified in this file.
5 # For more details on how to configure your build environment visit
6 # http://www.gradle.org/docs/current/userguide/build\_environment.html
7 # Specifies the JVM arguments used for the daemon process.
8 # The setting is particularly useful for tweaking memory settings.
9 org.gradle.jvmargs=-Xmx2048m -Dfile.encoding=UTF-8
10 # When configured, Gradle will run in incubating parallel mode.
11 # This option should only be used with decoupled projects. For more details, visit
12 # https://developer.android.com/r/tools/gradle-multi-project-decoupled-projects
13 # org.gradle.parallel=true
14 # AndroidX package structure to make it clearer which packages are bundled with the
15 # Android operating system, and which are packaged with your app's APK
16 # https://developer.android.com/topic/libraries/support-library/androidx-rn
17 android.useAndroidX=true
18 # Enables namespace of each library's R class so that its R class includes only the
19 # resources declared in the library itself and none from the library's dependencies,
20 # thereby reducing the size of the R class for that library
21 android.nonTransitiveRClass=true
22
23 android.enableJetifier=true
```

APP QR



The screenshot shows the AndroidManifest.xml file in an Android Studio code editor. The manifest file includes permissions for camera and features, and defines an application with various attributes.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-feature android:name="android.hardware.camera"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application QR"
        android:supportRtl="true"
        tools:targetApi="lollipop">
```

APP QR

The screenshot shows the Android Studio interface with the activity_main.xml layout file open. The layout consists of a central purple button labeled "Leer Código QR" and a text view below it containing the placeholder text "Contenido ... :D". The layout is defined using ConstraintLayout.

```
<?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">

    <Button
        android:id="@+id/scanner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="Leer Código QR"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.248" />

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Contenido ... :D"
        android:textSize="34sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/scanner" />

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

APP QR

```
public class MainActivity extends AppCompatActivity {

    2 usages
    Button scan_btn;
    2 usages
    TextView texto;

    @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);

            scan_btn = findViewById(R.id.scanner);
            texto = findViewById(R.id.text);

            scan_btn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    IntentIntegrator intentIntegrator = new IntentIntegrator( activity: MainActivity.this);
                    intentIntegrator.setOrientationLocked(true);
                    intentIntegrator.setPrompt("Leyendo QR");
                    intentIntegrator.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE);
                    intentIntegrator.initiateScan();
                }
            });
            return insets;
        });
    }
}
```

APP QR

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  
    IntentResult intentResult = IntentIntegrator.parseActivityResult(requestCode,resultCode,data);  
    if(intentResult != null){  
        String contenido = intentResult.getContents();  
        texto.setText(contenido);  
    }else{  
        super.onActivityResult(requestCode,resultCode,data);  
    }  
}
```