



ANDROID

SESIÓN 1

Presentación de la asignatura
Introducción al entorno de desarrollo

PLANIFICACIÓN PROPUESTA

Sesión	Contenido
1	Presentación Introducción entorno de desarrollo
2	Interfaz de usuario de una actividad Eventos Orientación
3	Ciclo de vida de una aplicación Guardar y recuperar el estado Widgets Scroll
4	Listas Adaptadores personalizados
5	Múltiples actividades Persistencia
6	Tareas asíncronas Conexiones HTTP Clase ViewModel
7	Fragmentos Mapas
8	Estilos y temas Barra de herramientas Menús Permisos Localización
9	Directorio assets Analizar XML

Android

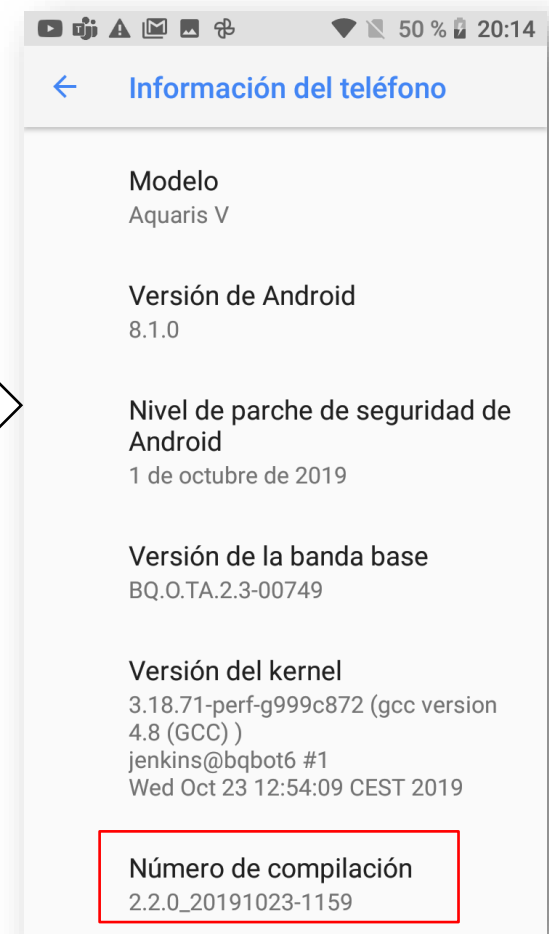
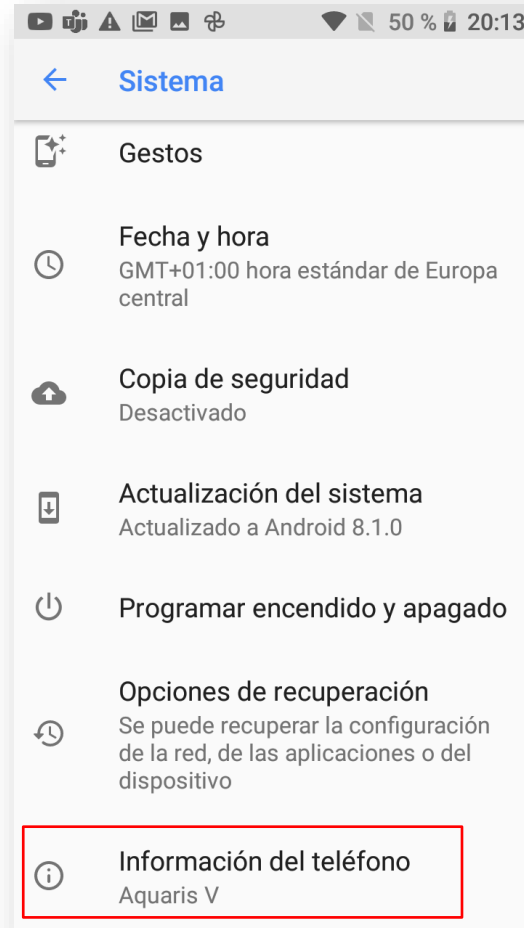
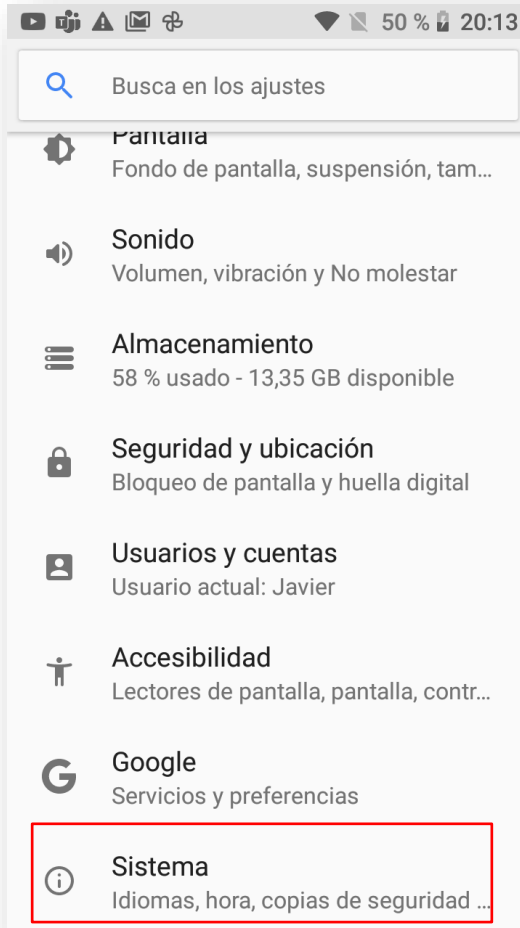
- ❑ Sistema operativo basado en el núcleo de Linux, diseñado para dispositivos móviles con pantalla táctil (teléfonos, tabletas, automóviles, televisiones o relojes inteligentes).
- ❑ La fundación *Open Handset Alliance* desarrolla *Android Open Source Project* (AOSP), software gratuito y de código abierto. Liderada por Google participan Samsung, Sony, LG, ...
- ❑ Android es una marca registrada por Google que impone estándares para restringir su uso en dispositivos no certificados
- ❑ *Google Mobile Services* (GMS) es un conjunto de aplicaciones (Chrome, Play y otros servicios) que se incluye en la mayoría de los dispositivos que usan AOSP
- ❑ A principios de 2025, el 73,5% de los dispositivos del mundo usan Android frente al 26% que usan iOS. En España 79/20.

Fuente <https://gs.statcounter.com/os-market-share/mobile/worldwide>

Lenguajes de programación para Android*

- ❑ **Java.** El primer lenguaje oficial y usado en muchas aplicaciones antiguas (25-30%)
- ❑ **Kotlin.** Aparece en 2017 como alternativa a Java. Sintaxis más concisa y clara, diseñado específicamente para trabajar con Android. Ahora es el dominante (60-70%)
- ❑ **Dart.** Usado en el *framework* Flutter para el desarrollo de aplicaciones multiplataforma (5-10%)
- ❑ **C++.** Usado mediante la plataforma *Android Native Development Kit* (NDK), permite acceder a funciones de bajo nivel y mejorar el rendimiento de las aplicaciones (5-10%)
- ❑ **Python, C#, JavaScript,** Usando los *framework* adecuados (<5%)

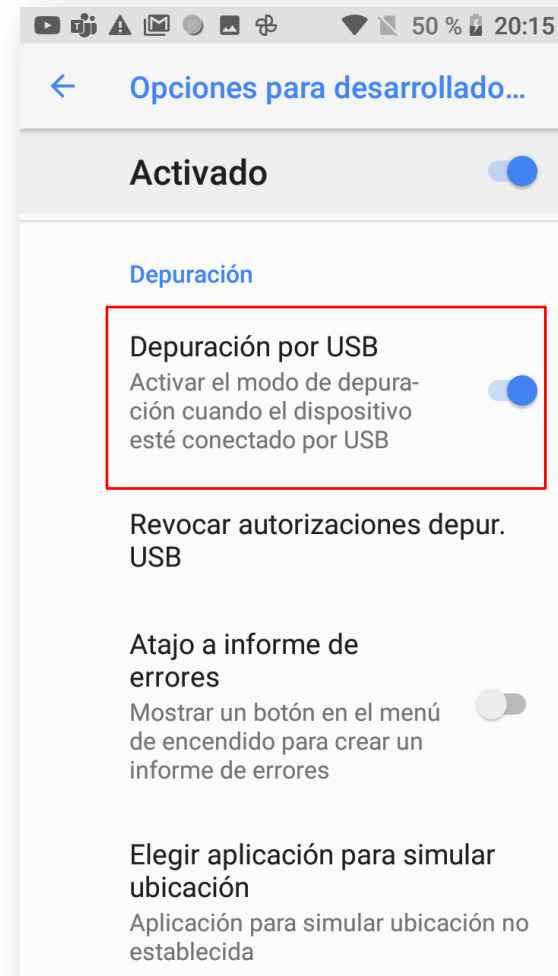
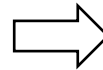
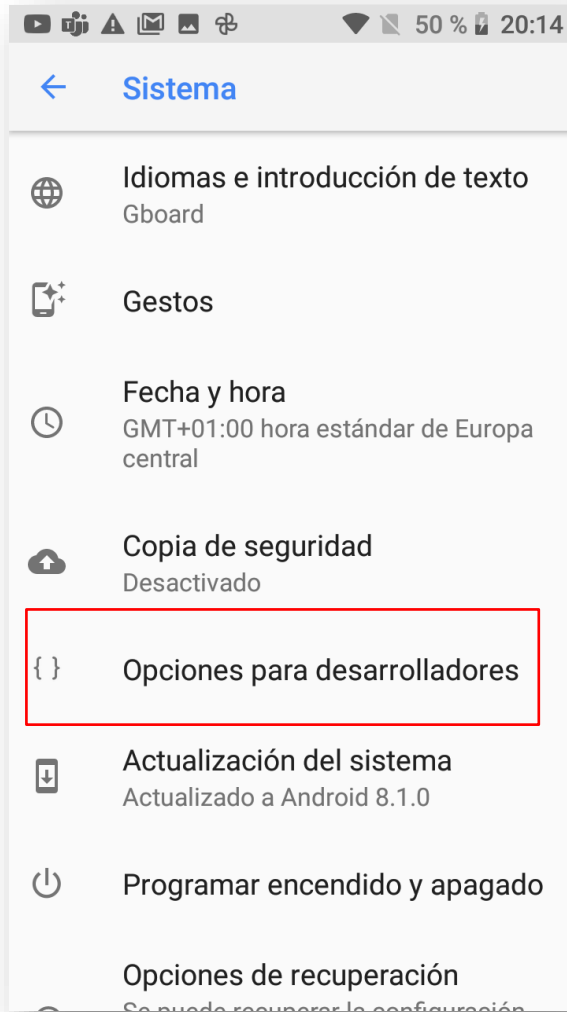
Terminal de prueba: activación de las opciones del desarrollador (1)



Pulsar 7 veces

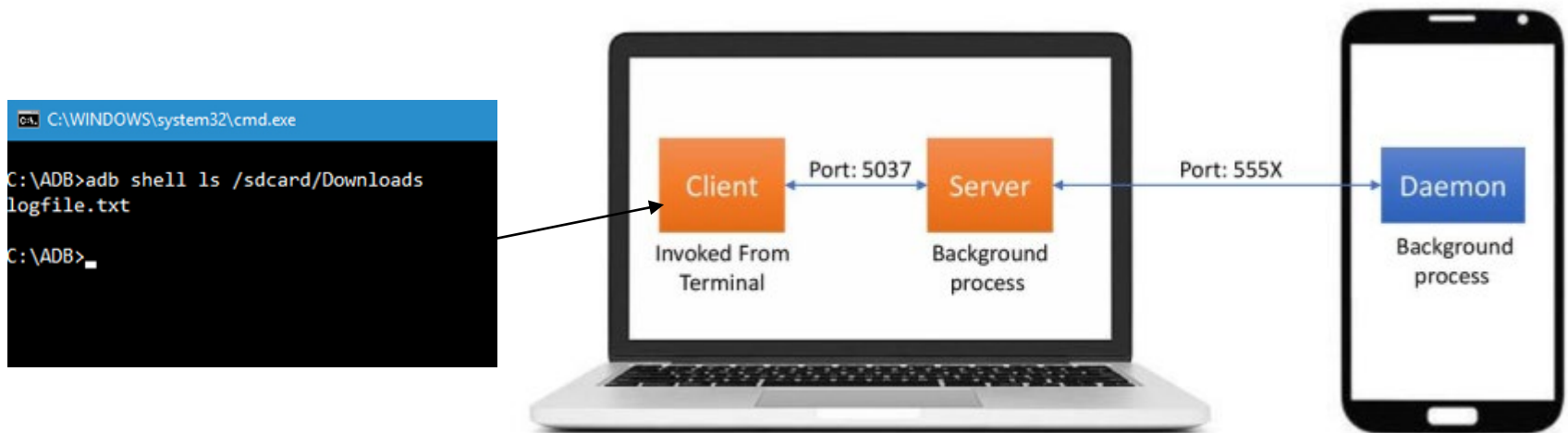
Dependiendo del fabricante del terminal, estas pantallas pueden ser diferentes

Terminal de prueba: activación de las opciones del desarrollador (2)



Interfaz ADB: Android Debug Bridge (1)

- ❑ Android Debug Bridge (ADB) permite la comunicación con un dispositivo Android con fines de monitorización, instalación de aplicaciones y depuración, entre otras.
- ❑ ADB es un programa cliente servidor. Al arrancarlo por primera vez se crea el servidor en el ordenador y se arranca el cliente. El servidor sirve para enviar la información del cliente a un demonio que está corriendo en el terminal, que es el que ejecuta el comando



[Fuente](#)

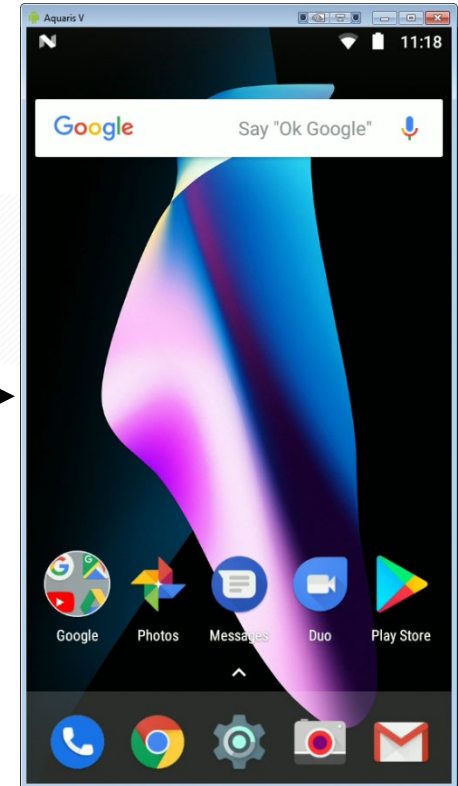
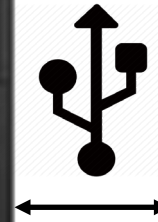
Interfaz ADB: Android Debug Bridge (2)

- ❑ El entorno de desarrollo Android Studio usa ADB para instalar y ejecutar las aplicaciones, si bien puede usarse como una aplicación independiente.
- ❑ Algunas capacidades:
 - Listado de los dispositivos conectados.
 - Instalación, ejecución y depuración de aplicaciones.
 - Copia de ficheros desde/hacia el terminal.
 - Acceso al intérprete de órdenes del terminal
 - Capturas de pantalla
 - Funciona tanto por cable como por wifi
- ❑ El ejecutable ADB está en el directorio del SDK dentro de `platform-tools`
 - Por defecto en `C:\users\usuario\AppData\Local\Android-Sdk\platform-tools`
 - En los PCs del laboratorio en `C:\Android\sdk\platform-tools`
- ❑ Si no se dispone del SDK, ADB se puede descargar de forma aislada desde [aquí](#)
- ❑ Más información: <https://developer.android.com/studio/command-line/adb.html>

Invocación desde el intérprete de órdenes

```
platform-tools> adb devices -l
List of devices attached
WI019834      device usb:1-9 product:nappa
emulator-5554 device product:sdk_gphone_x86_64

platform-tools>
```



Ejercicio 1

Habiendo habilitado las opciones de depuración en el terminal Android que vas a usar, conéctalo por USB al ordenador. Abre un intérprete de órdenes y, mediante adb:

1. Lista los terminales Android conectados al ordenador
2. Lista el contenido del directorio Download (se encuentra en /sdcard/Download)
3. Copia un fichero de texto del ordenador al terminal dejándolo en el directorio Download. Verifica que se ha copiado abriéndolo en el terminal con algún programa del terminal (por ej. *Files*)
4. Abre un intérprete de órdenes (*shell*) en el terminal, y navega por los directorios, ejecutando mandatos como `ls` y `cd`, hasta localizar donde se guardan los ficheros de las fotos que se hacen con la cámara. Cuando termines sal del intérprete de órdenes del terminal para volver al intérprete de órdenes del ordenador
5. Copia una foto, existente en el terminal, al ordenador. Verifica que se ha copiado abriéndola en el ordenador
6. Con una orden de adb haz una captura de pantalla del terminal, guardándola en el directorio Download del terminal, y luego muéstrala en el terminal con alguna aplicación del terminal
7. Con una orden de adb graba un video en el terminal, guardando el fichero de video en el directorio Download del terminal, y luego muéstralo en el terminal
8. Desde el navegador web del ordenador, accede al servicio <https://app.webadb.com> y conéctate al terminal. Examina las opciones

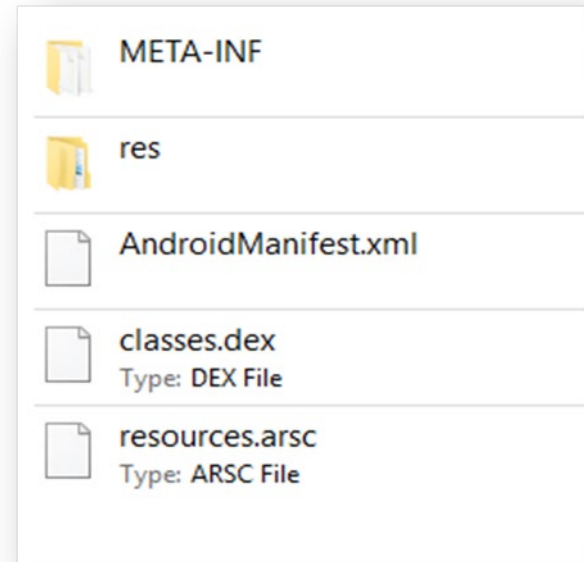
Ficheros APK

Android SDK tools compila el código fuente (Java/Kotlin) en ficheros “.dex”, y genera un fichero ZIP con el bytecode “.dex”, los recursos de la aplicación (imágenes, XML de configuración, etc.) y el fichero “manifest” en un fichero APK.

Mediante un descompresor ZIP se puede descomprimir un fichero android “.apk” y ver su contenido.

Android Studio genera el fichero .apk:

```
<RaizProyecto>/app/build/intermediates/apk/debug
```



Ejercicio 2

1. Descarga en tu ordenador la aplicación *Prueba*, cuyo *apk* está en moodle.
2. Instálala en el terminal Android mediante un mandato *adb*, teniendo en cuenta que es una aplicación de prueba (se necesita el parámetro *-t*) *
3. Ejecútala en el terminal Android
4. Abre el fichero *apk* con un descompresor de archivos zip, por ejemplo [7-zip](#), o bien renombra la extensión por zip para abrirla con el explorador de archivos de tu ordenador. Examina su contenido hasta localizar el icono de la aplicación
5. Desinstálala desde el ordenador, con una orden *adb*, teniendo en cuenta que el paquete empieza por la palabra *dam*.

Pistas: para desinstalar una aplicación tienes que conocer el nombre del paquete, y los paquetes de las aplicaciones instaladas se pueden conocer con el comando *list* de la *shell* del terminal.

* En algunos terminales, además de la depuración por USB hay que autorizar la instalación de aplicaciones apk por usb dentro de las opciones de desarrollador, si no, el comando de instalación da un error de *no permitido*.

Versiónes Android

Name
My Application

Package name
[Empty field]

Save location
[Empty field]

Language
Java

Minimum SDK
API 27: Android 8.1 (Oreo)

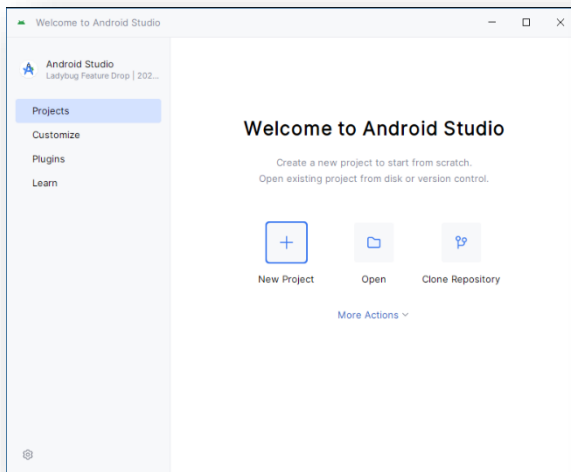
Your app will run on approximately 53,5% of devices.
[Help me choose](#)

Use legacy android.support libraries

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.4 KitKat	19	
5 Lollipop	21	99,7%
5.1 Lollipop	22	99,6%
6 Marshmallow	23	98,8%
7 Nougat	24	97,4%
7.1 Nougat	25	96,4%
8 Oreo	26	95,4%
8.1 Oreo	27	93,9%
9 Pie	28	89,6%
10 Q	29	81,2%
11 R	30	67,6%
12 S	31	48,6%
13 T	33	33,9%
14 U	34	13,0%

Last updated: May 1, 2024

Android SDK



Settings

Search: Q-

Languages & Frameworks > Android SDK Revert changes ← →

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: Edit Optimize disk space

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API ...	Re...	Status
> <input type="checkbox"/> Android 10.0 ("Q")			
> <input type="checkbox"/> Android 9.0 ("Pie")			
<input checked="" type="checkbox"/> Android 8.1 ("Oreo") <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Android SDK Platform 27 27 3 Installed <input type="checkbox"/> Sources for Android 27 27 1 Not insta... <input type="checkbox"/> Android TV Intel x86 Atom System Image 27 9 Not insta... <input type="checkbox"/> ARM 64 v8a System Image 27 2 Not insta... <input type="checkbox"/> Intel x86 Atom System Image 27 1 Not insta... <input type="checkbox"/> Intel x86_64 Atom System Image 27 1 Not insta... <input type="checkbox"/> Google APIs ARM 64 v8a System Image 27 3 Not insta... <input checked="" type="checkbox"/> Google APIs Intel x86 Atom System Image 27 11 Installed <input checked="" type="checkbox"/> Google Play Intel x86 Atom System Image 27 3 Installed 			
> <input type="checkbox"/> Android 8.0 ("Oreo")			
> <input type="checkbox"/> Android 7.1.1 ("Nougat")			

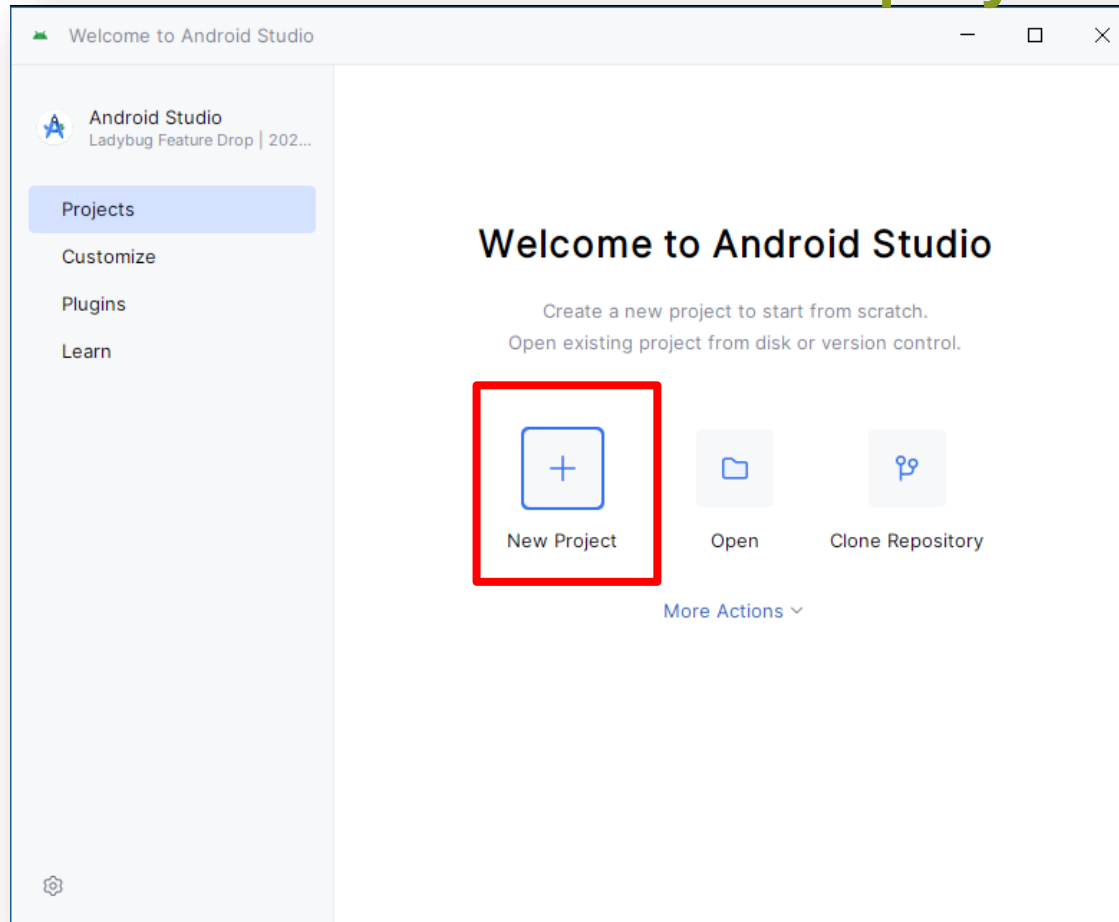
Hide Obsolete Packages Show Package Details

Project-level settings will be applied to new projects OK Cancel Apply

Plataformas de desarrollo de Android

- [Android Studio](#). Plataforma oficial de Google para aplicaciones Android. Lenguajes de programación Java y Kotlin
- [Flutter](#) (de Google). Crea aplicaciones mutiplataforma para Android , iOS , Linux , macOS , Windows y web. Lenguaje de programación Dart
- [React Native](#). Crea aplicaciones nativas mutiplataforma para Android e iOS . Lenguaje de programación JavaScript
- [Xamarin](#) (de Microsoft). Crea aplicaciones nativas mutiplataforma para Android, iOS y Windows. Lenguaje de programación C#

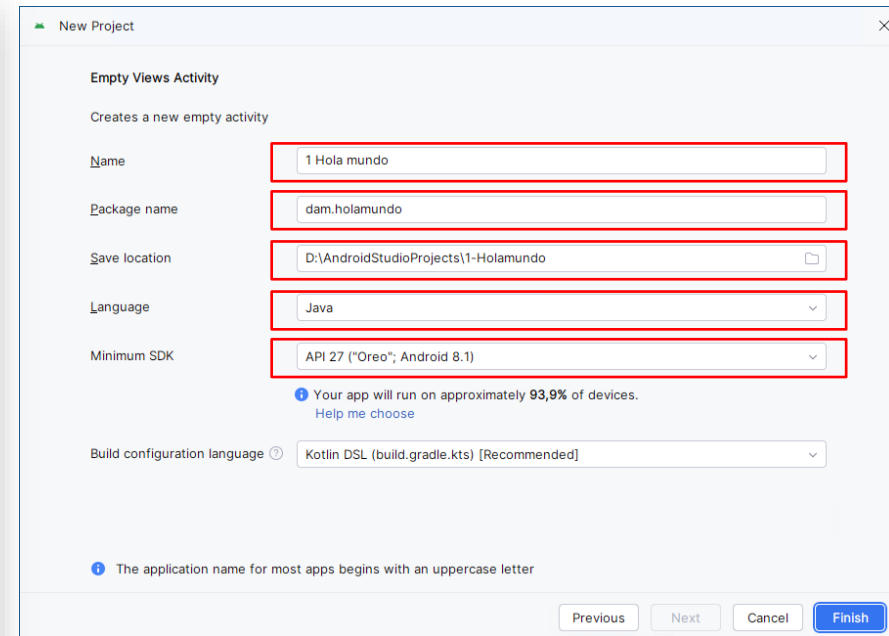
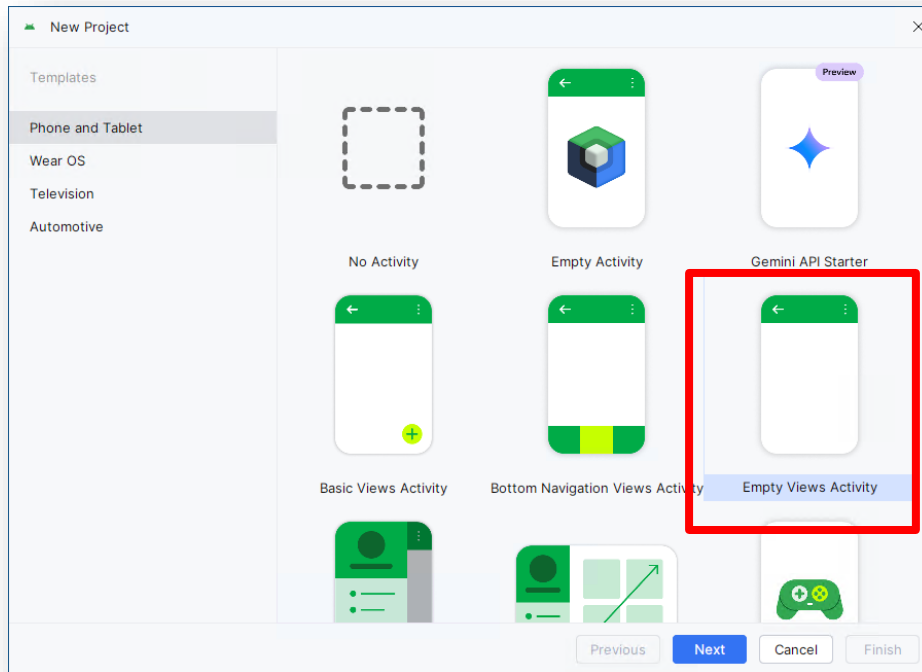
Android Studio : crear nuevo proyecto (1)



- ❑ Más información:

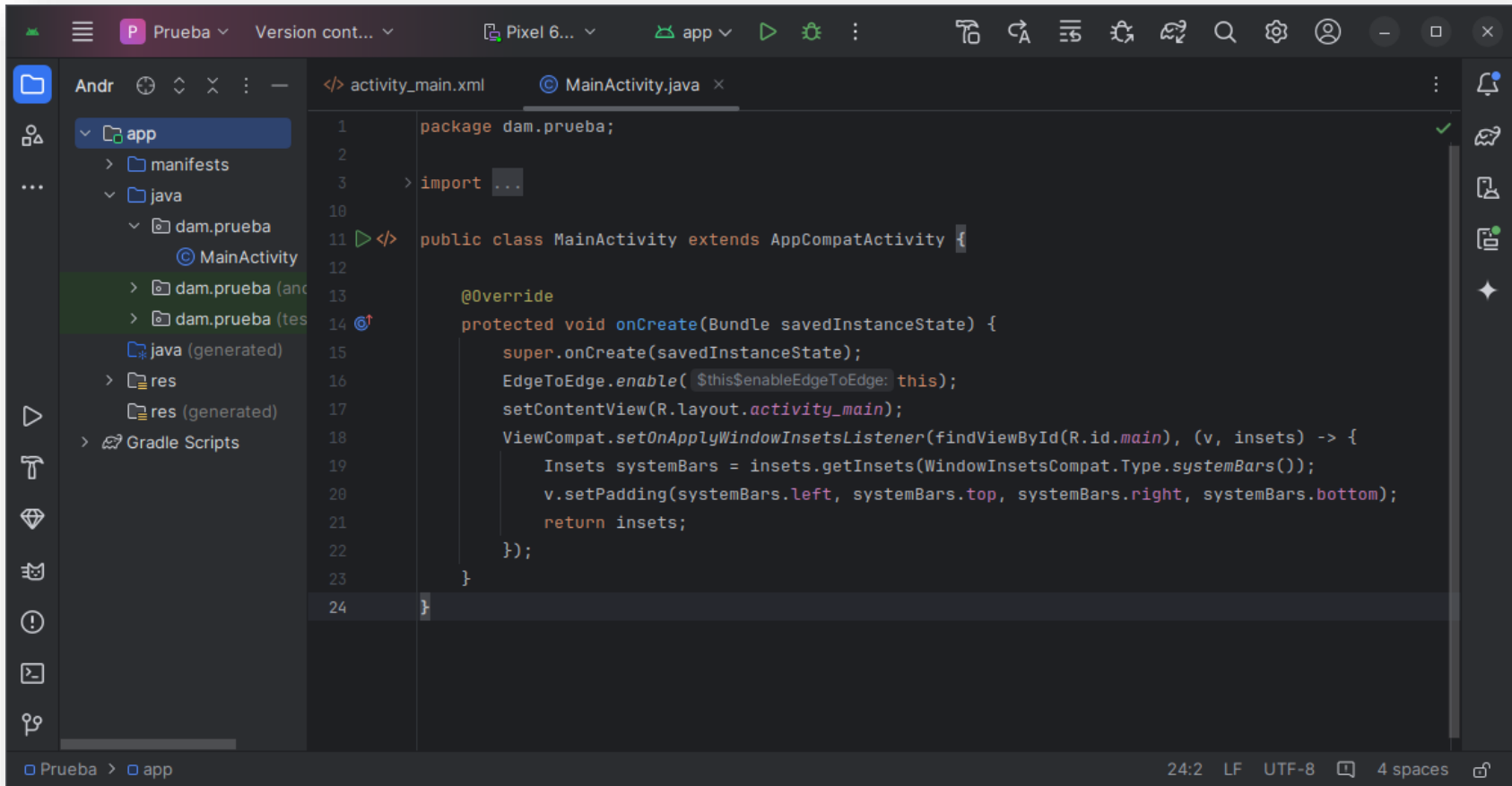
<https://developer.android.com/studio/projects/create-project?hl=es-419>

Android Studio: crear nuevo proyecto (2)



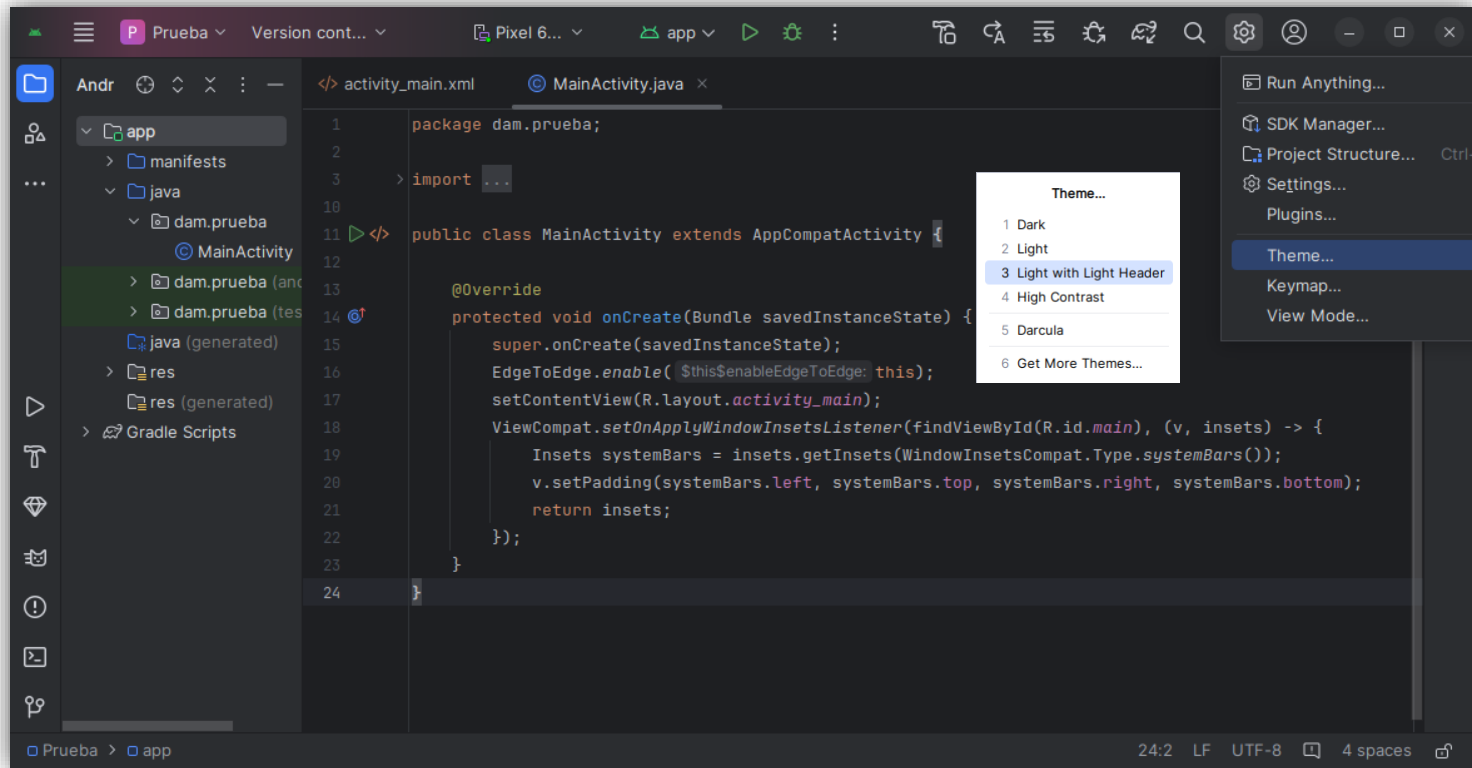
En este ejemplo se elige la API 27 para cubrir un amplio número de terminales, pero podría elegirse otra más actual, asumiendo que no se podrá ejecutar en terminales más antiguos

Android Studio: crear nuevo proyecto (3)

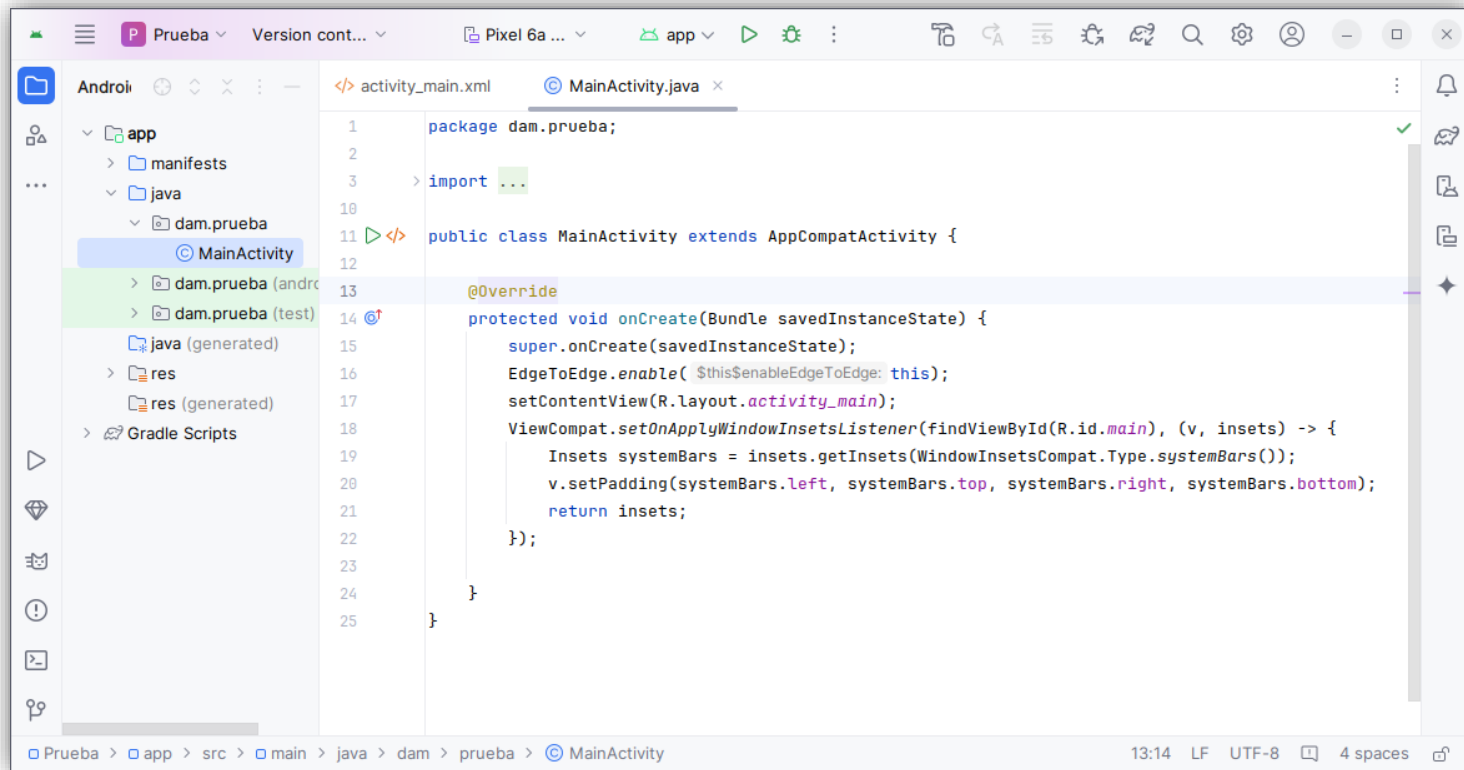


```
1 package dam.prueba;
2
3 import ...
4
10
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         EdgeToEdge.enable(this);
17         setContentView(R.layout.activity_main);
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 }
```

Android Studio: Cambiar tema (1)



Android Studio: Cambiar tema (2)

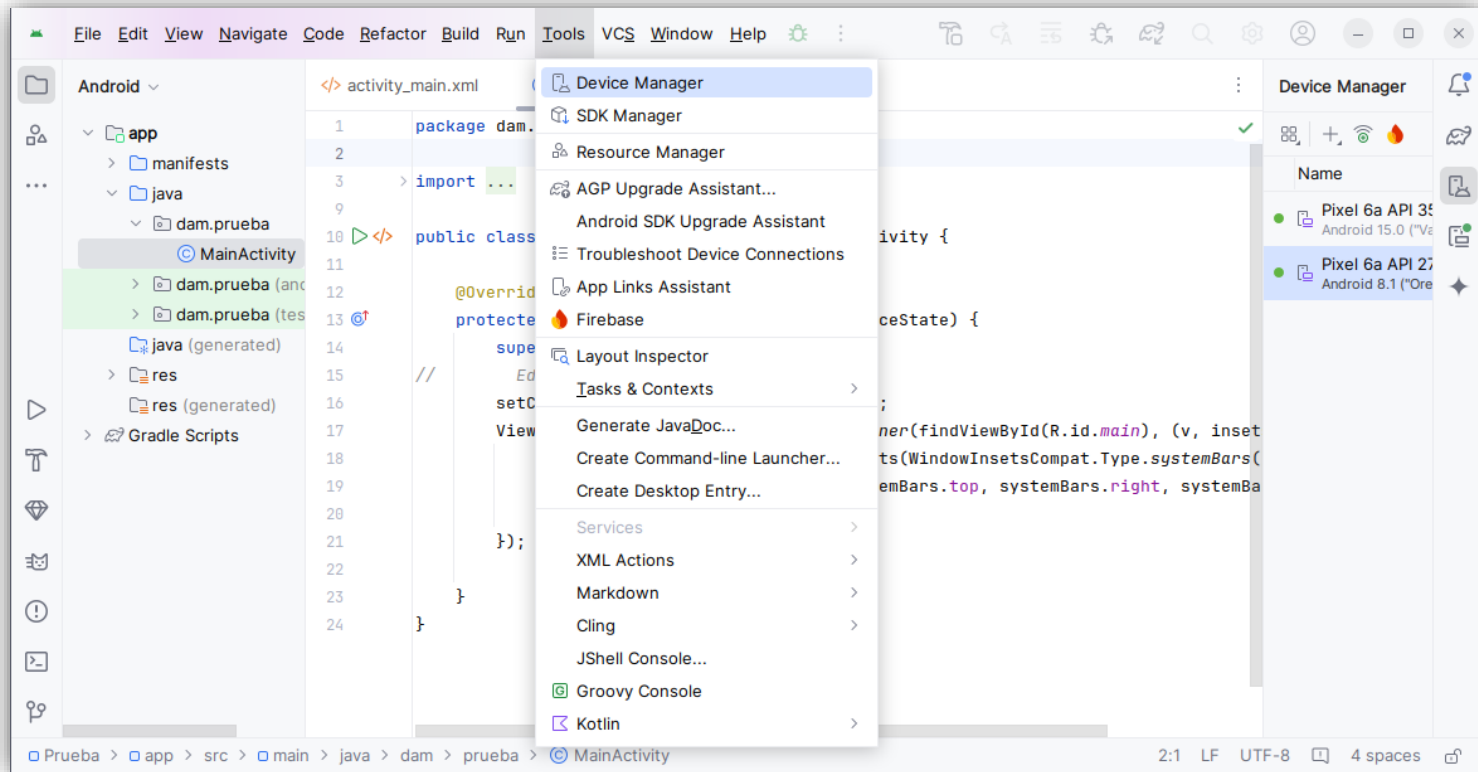


The screenshot shows the Android Studio IDE with the MainActivity.java file open. The code is as follows:

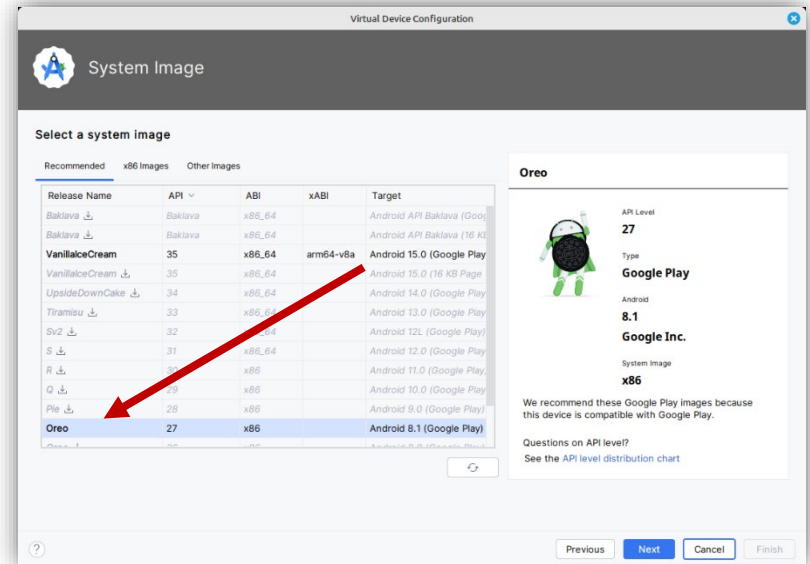
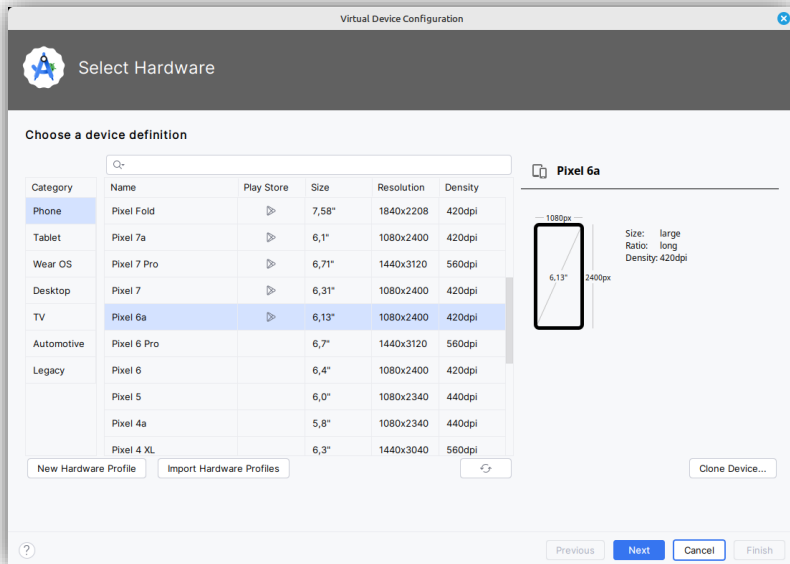
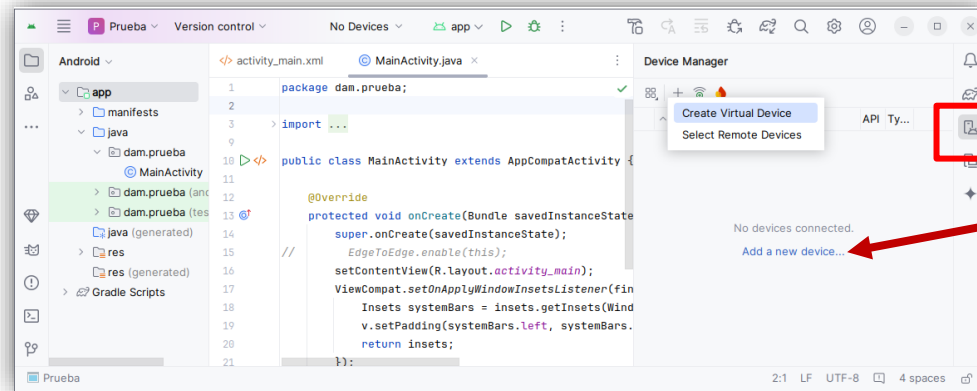
```
1 package dam.prueba;
2
3 import ...
4
5
6
7
8
9
10
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         EdgeToEdge.enable(this);
17         setContentView(R.layout.activity_main);
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 }
25 }
```

The status bar at the bottom shows the file path: Prueba > app > src > main > java > dam > prueba > MainActivity. The system clock is 13:14, the encoding is UTF-8, and there are 4 spaces.

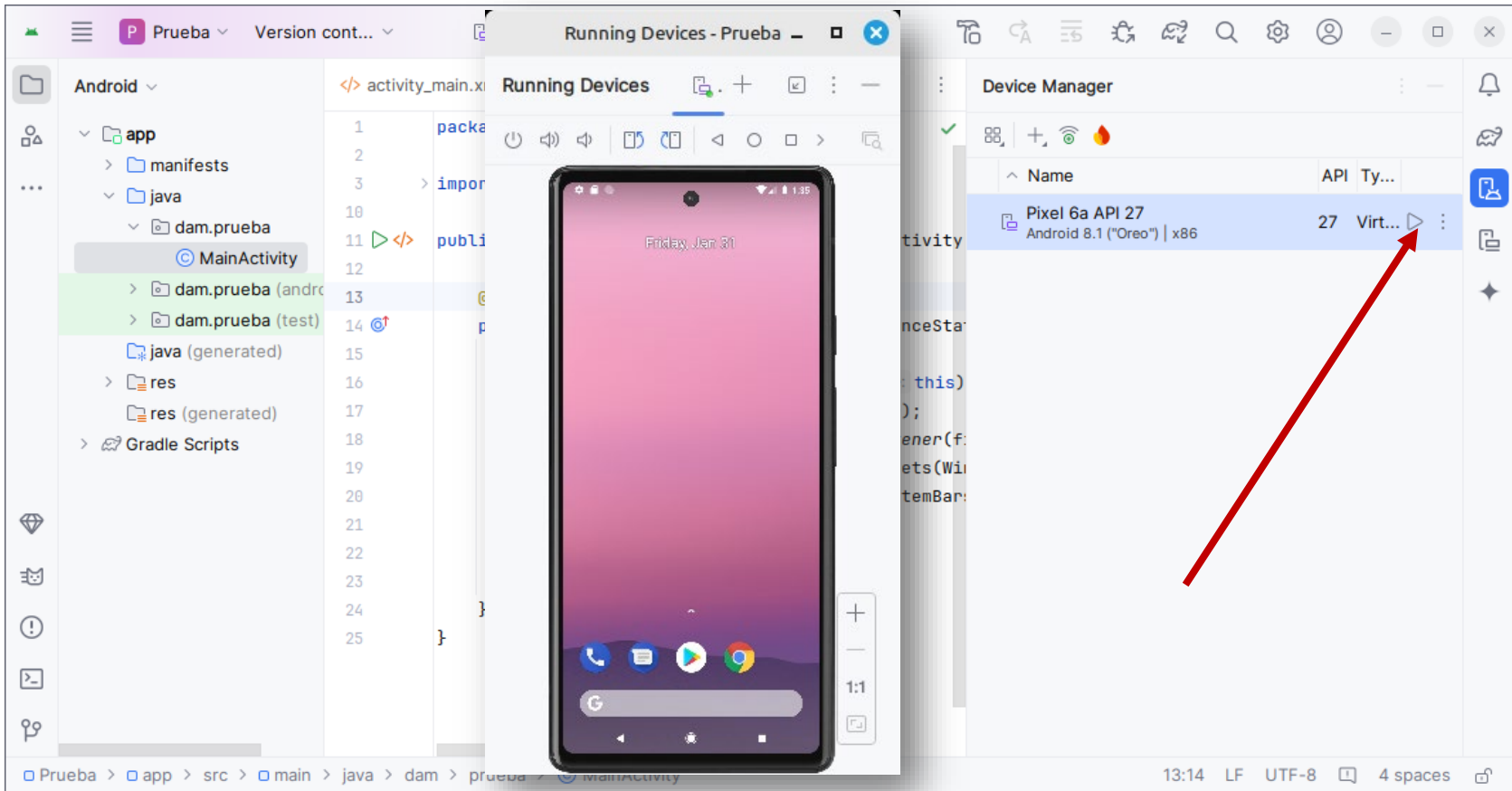
Android Studio: crear un emulador (1)



Android Studio: crear un emulador (2)



Android Studio: ejecutar el emulador

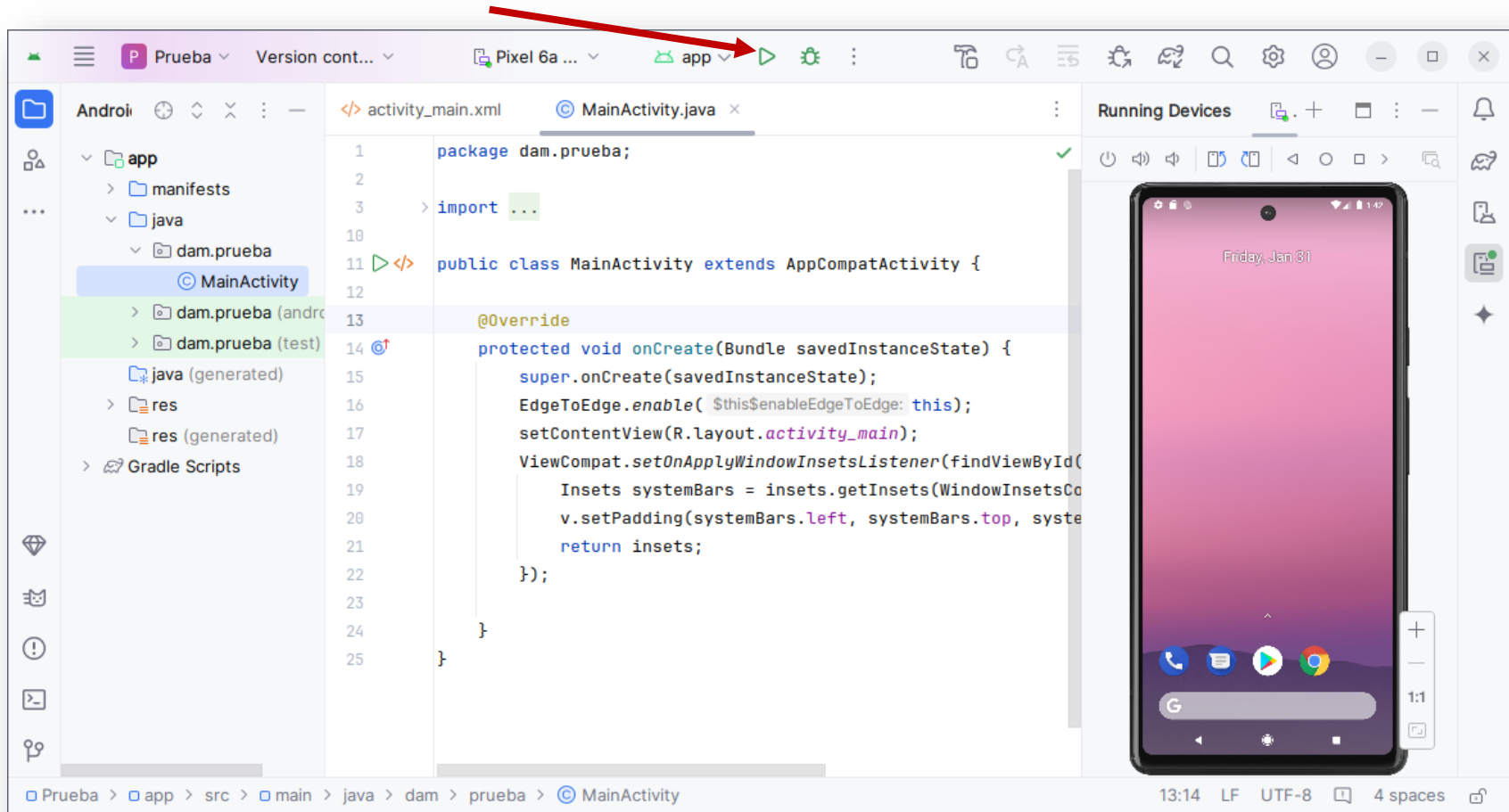


También se puede ejecutar desde el intérprete de órdenes:

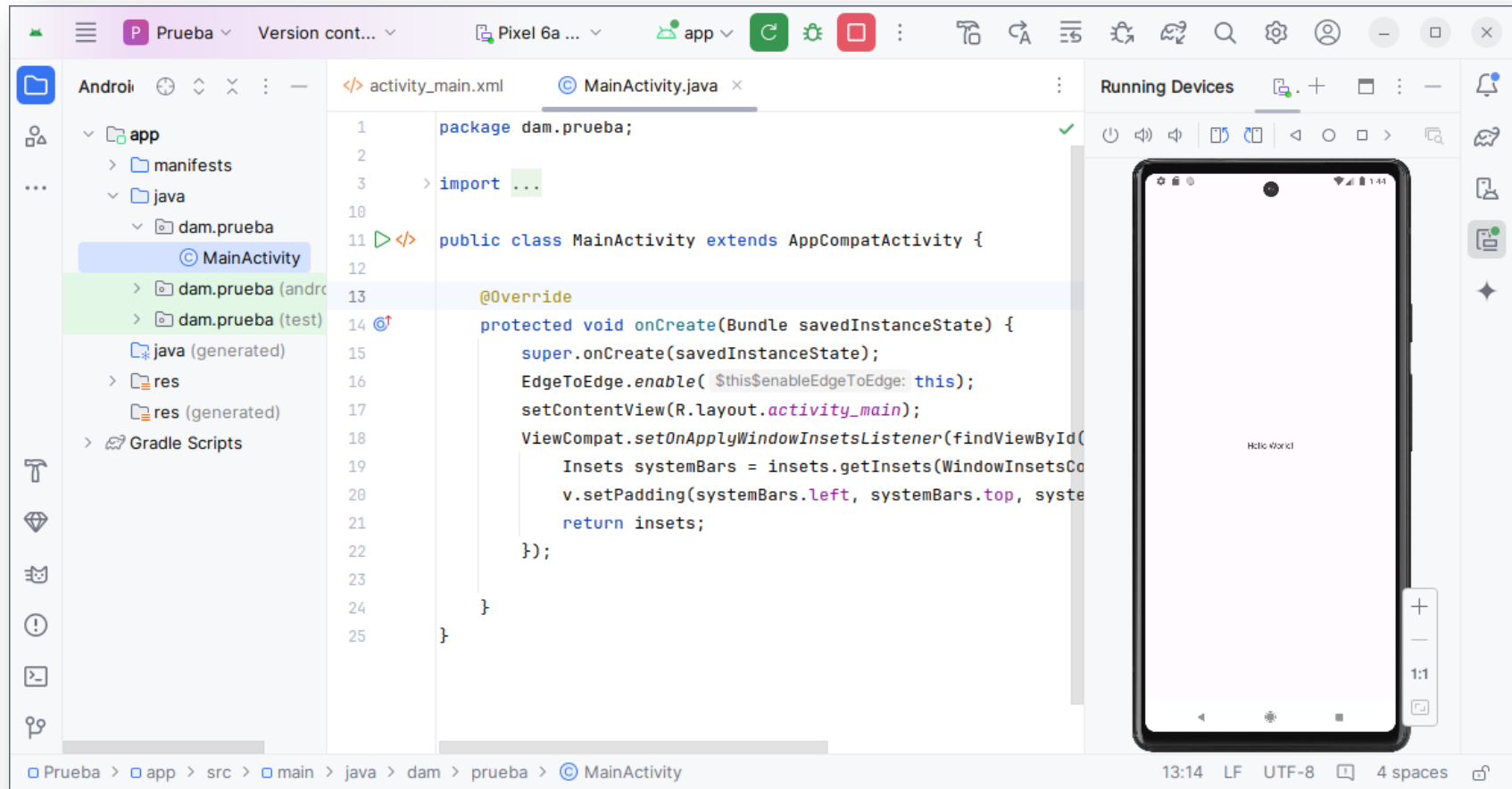
`Directorio_sdk/emulator/emulator @nombre_dispositivo` donde `nombre_dispositivo` se obtiene de:

`Directorio_sdk/emulator/emulator -list-avds`

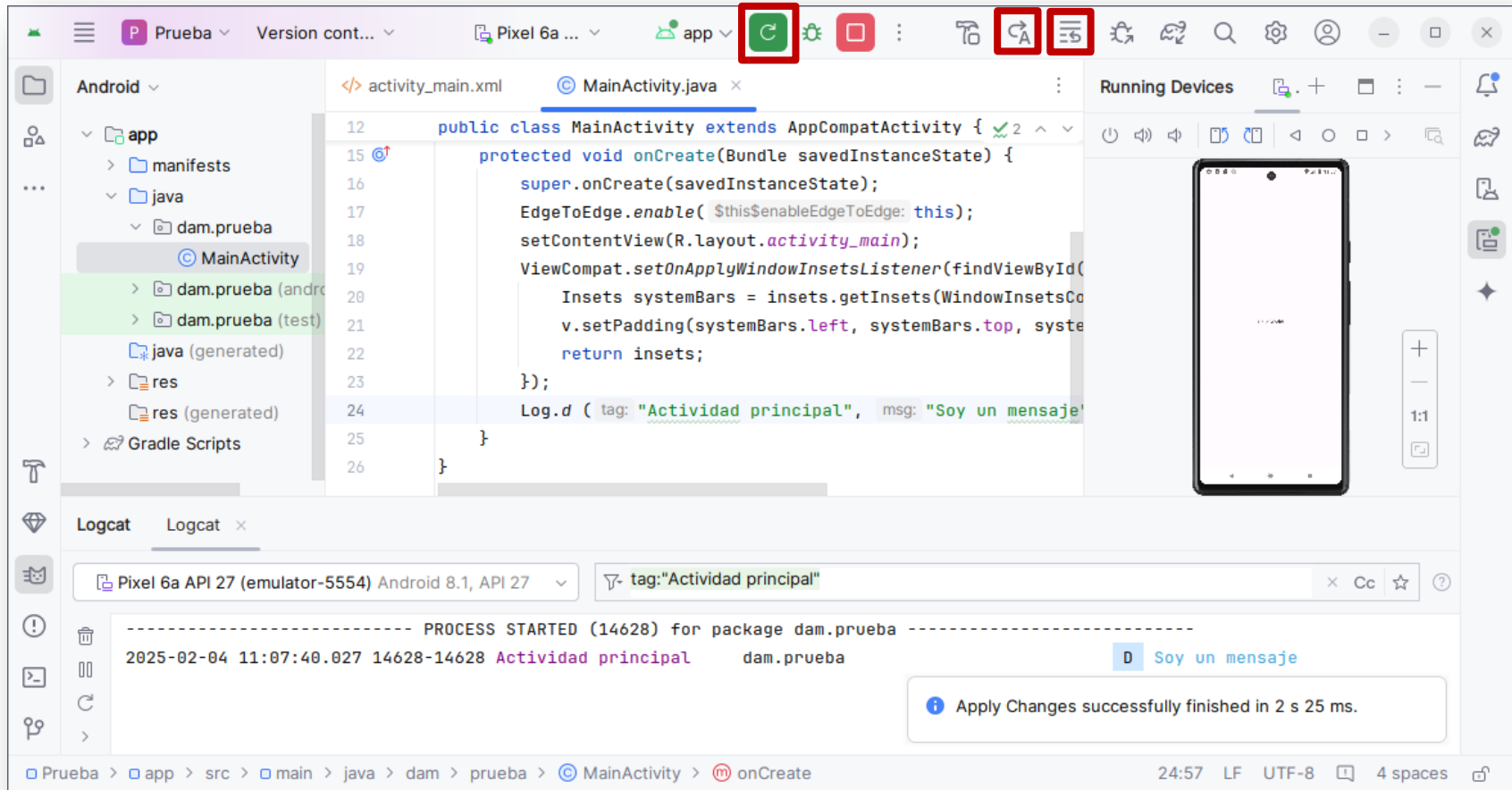
Android Studio: arrancar la aplicación (1)



Android Studio: arrancar la aplicación (2)



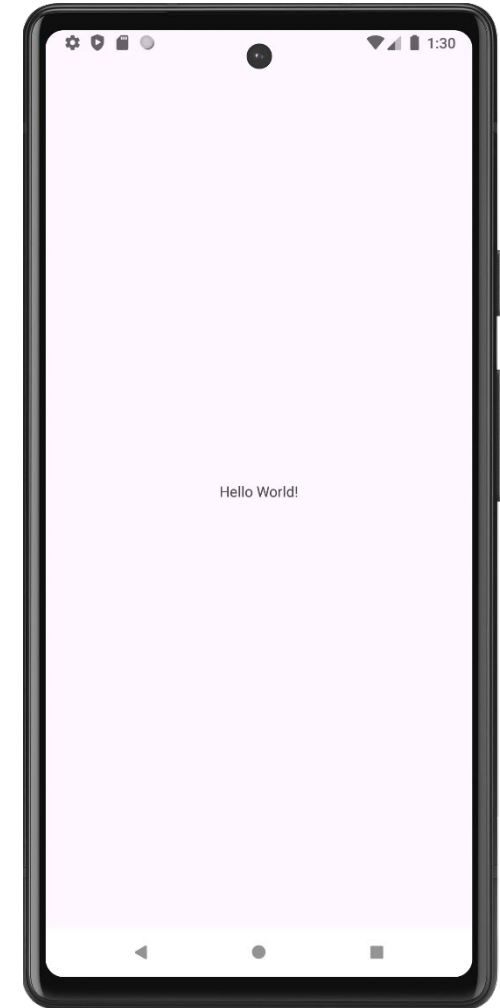
Android Studio: reiniciar la aplicación



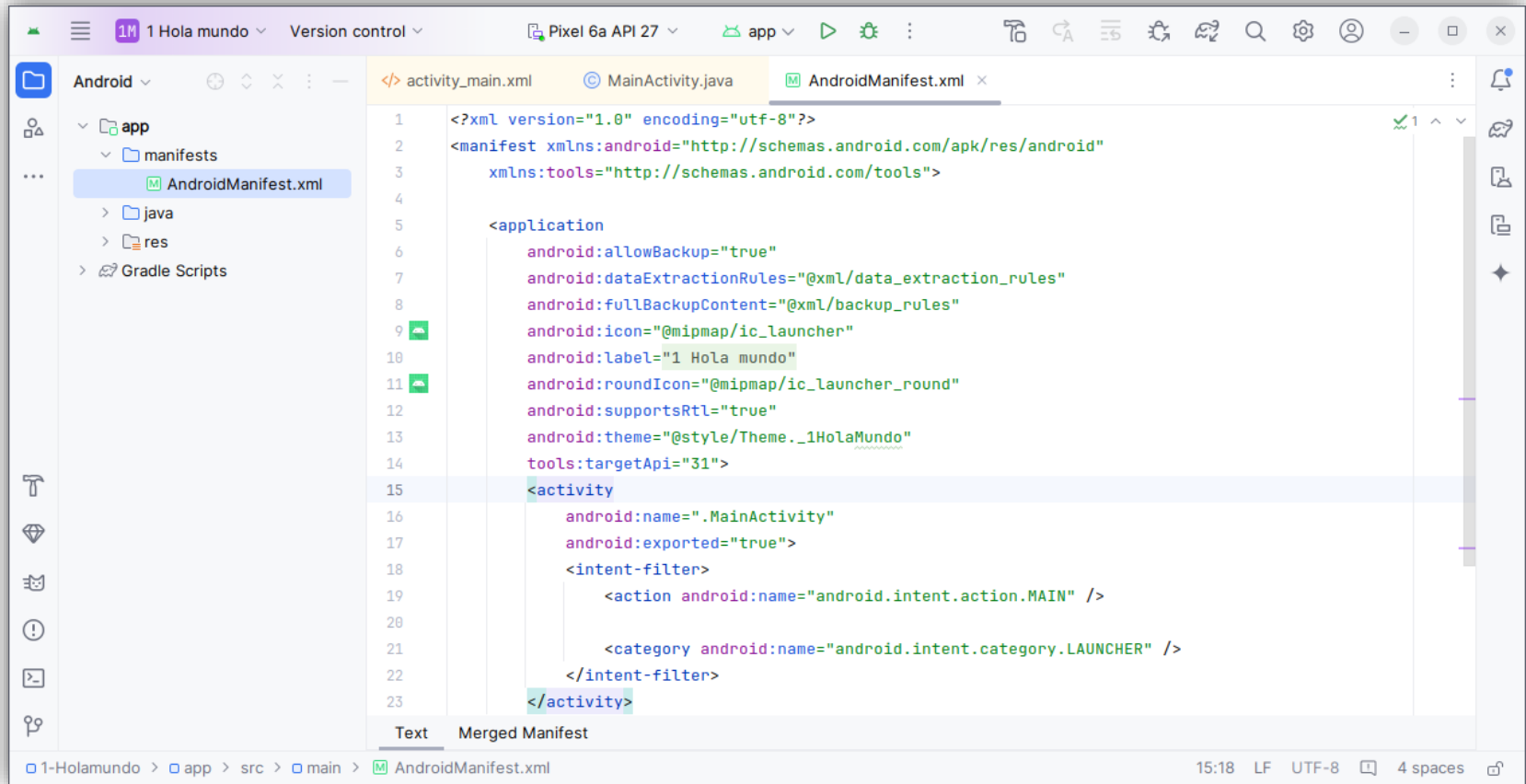
Ejercicio 3

- ❑ En Android Studio crea un proyecto nuevo, con las siguientes características:
 - Plantilla: *Empty Views Activity*
 - Nombre: 1 Hola mundo
 - Paquete: dam.holamundo
 - Directorio: 1-HolaMundo

- ❑ Compíllalo y ejecútalo en un terminal Android o en el emulador



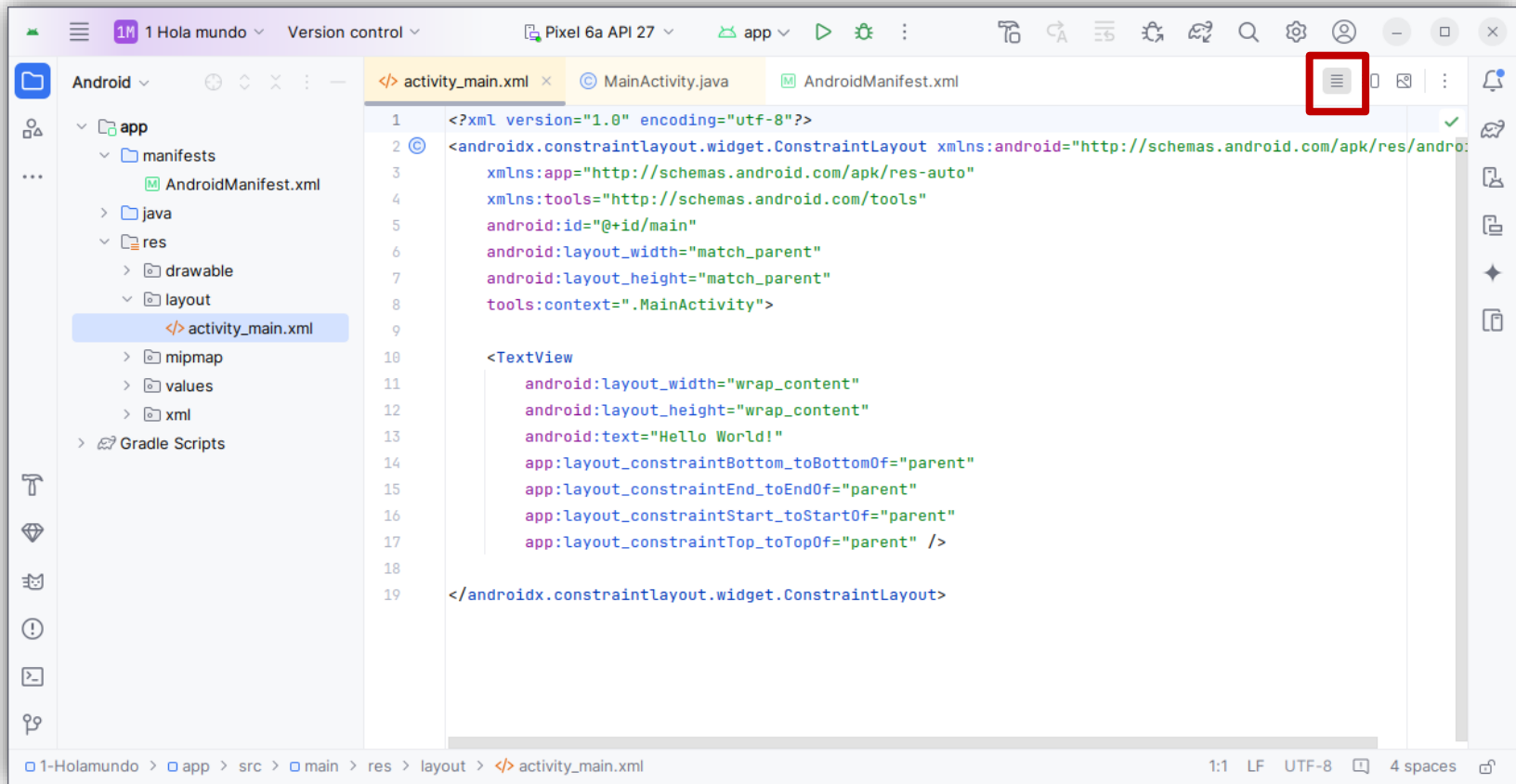
Archivo Manifest



```
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     <application
6         android:allowBackup="true"
7         android:dataExtractionRules="@xml/data_extraction_rules"
8         android:fullBackupContent="@xml/backup_rules"
9         android:icon="@mipmap/ic_launcher"
10        android:label="@string/app_name"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportsRtl="true"
13        android:theme="@style/Theme._1HolaMundo"
14        tools:targetApi="31">
15
16         <activity
17             android:name=".MainActivity"
18             android:exported="true">
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25     </application>
26 </manifest>
```

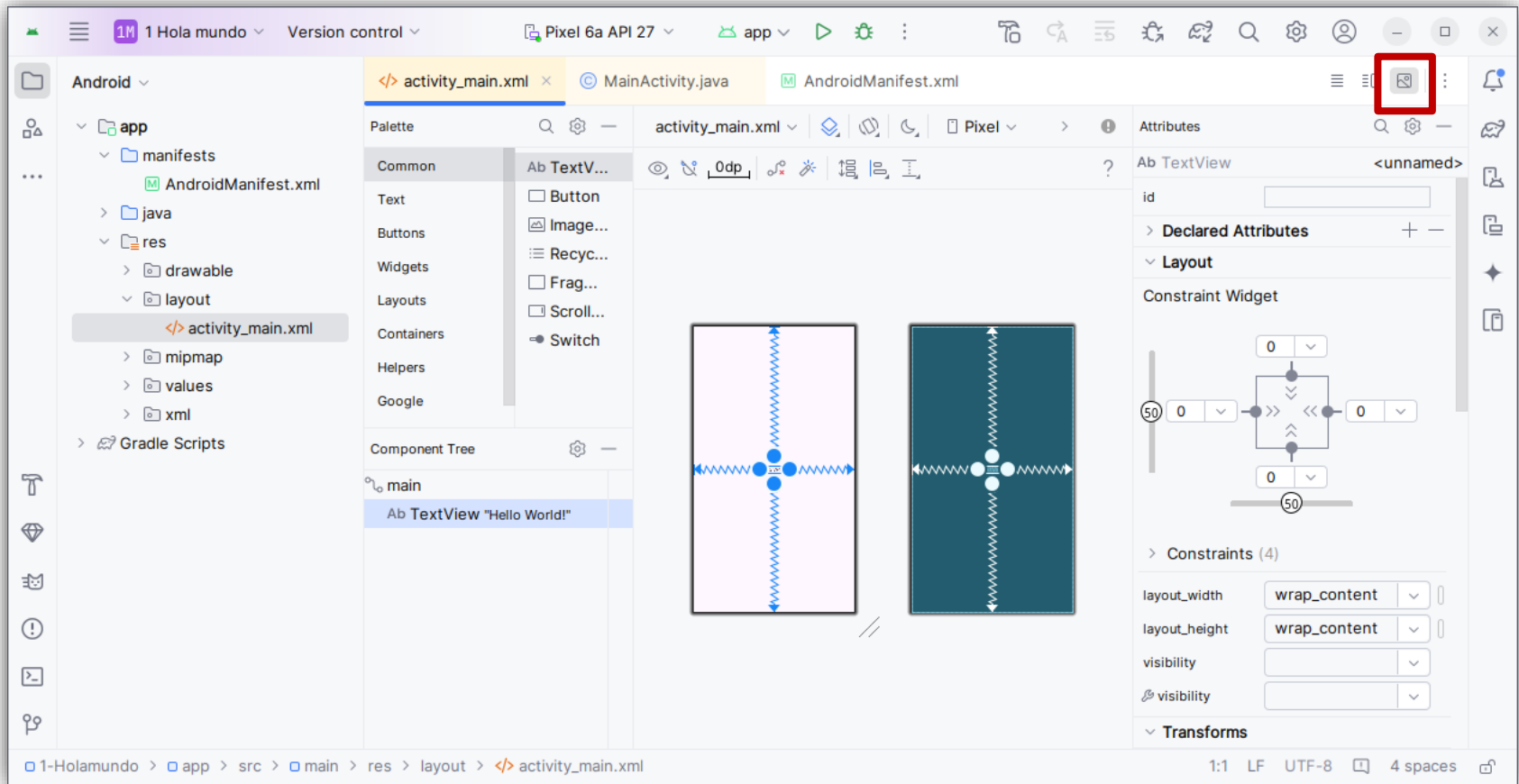
Más información: developer.android.com/guide/topics/manifest/manifest-intro

Layout (diseños): disposición de los elementos gráficos (1)



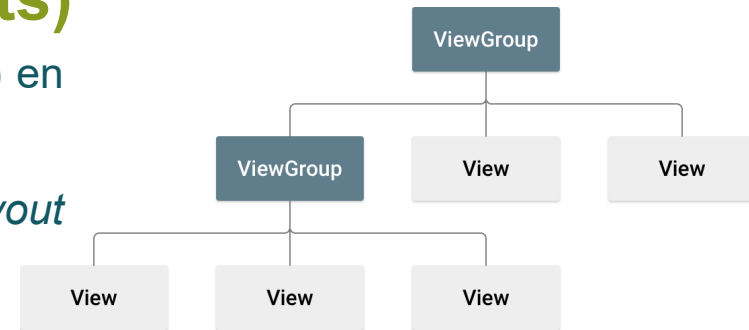
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintEnd_toEndOf="parent"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

Layout (diseños): disposición de los elementos gráficos (2)



Diseños (Layouts)

- ❑ Controla la posición de un elemento gráfico (*View*) en un contenedor (*ViewGroup*)
- ❑ Existen diferentes tipos y se pueden anidar (*layout* dentro de *layout*)
- ❑ *Layouts* básicos
 - *LinearLayout*: Los elementos alineados en una dirección: horizontal o vertical
 - *RelativeLayout*: Cada elemento define una posición relativa a otro elemento
 - *ConstraintLayout*: Similar a *RelativeLayout* pero más potente y flexible.
 - *ListView*: Lista de elementos con desplazamiento
 - *GridView*: Elementos dispuestos en una rejilla



LinearLayout



RelativeLayout



ListView



GridView

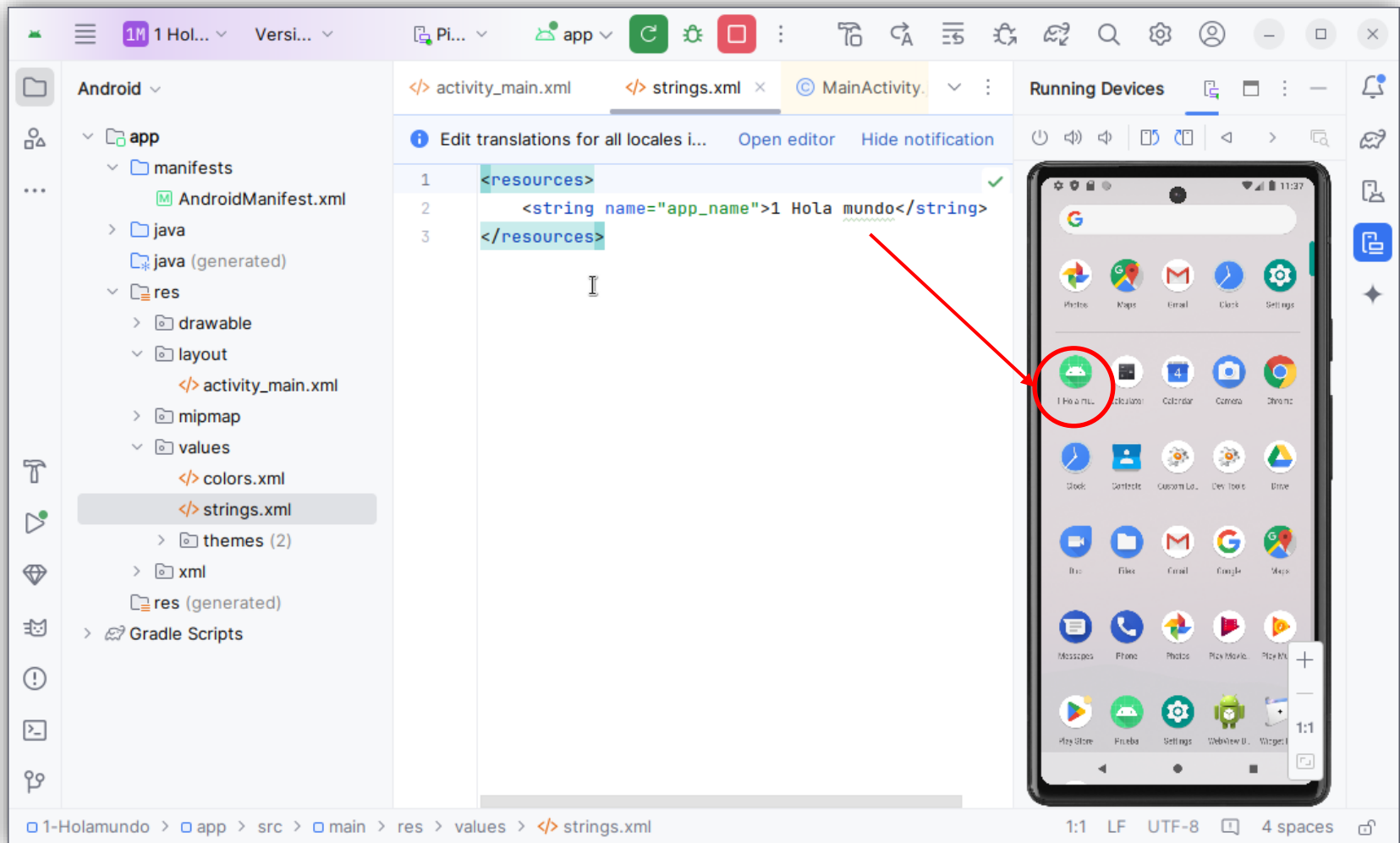
- ❑ Más información: <https://developer.android.com/guide/topics/ui/declaring-layout>

Convertir un ConstraintLayout a un LinearLayout

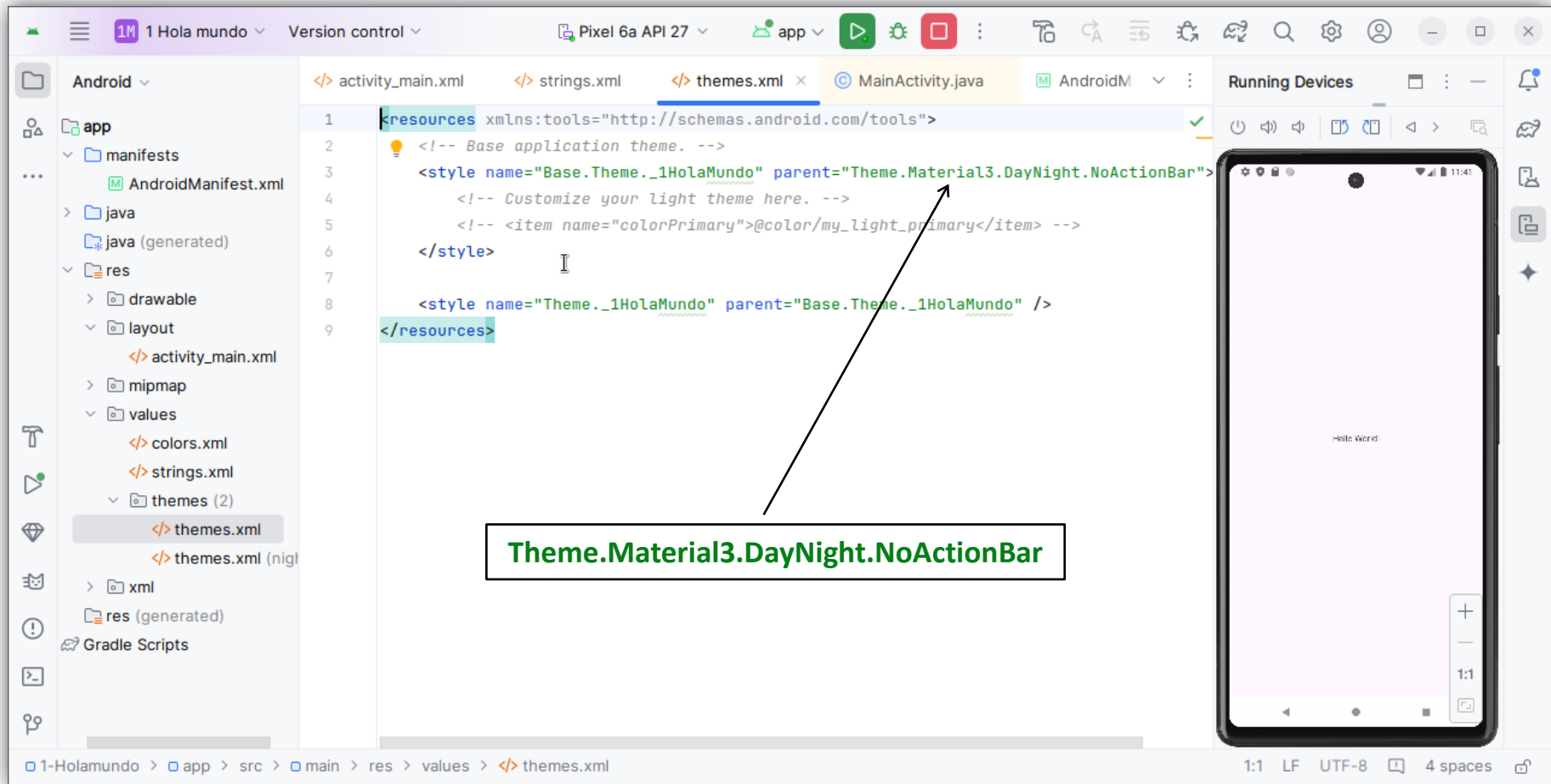
The screenshot shows the Android Studio IDE with the following components:

- Project Explorer:** Shows the project structure for '1 Hola mundo', including 'manifests', 'java', 'res', and 'layout' folders. The 'activity_main.xml' file is selected in the 'layout' folder.
- Component Tree:** Shows the 'main' component with a single child, 'Ab TextView "Hello World!"'.
- Palette:** Displays various widget categories like Common, Text, Buttons, etc. The 'Ab TextView...' widget is selected.
- Attributes:** Shows the properties for the selected 'ConstraintLayout' widget, including 'id', 'Declared Attributes', 'Layout', and 'Transforms'.
- Context Menu:** A right-click context menu is open over the widget, with the 'Convert view...' option highlighted.
- Convert View Dialog:** A dialog box titled 'Convert View to:' is open, showing a list of layout types: 'LinearLayout', 'CoordinatorLayout', 'RelativeLayout', and 'FrameLayout'. 'LinearLayout' is selected.
- Preview:** A visual preview of the widget is shown, with a rotation control at the bottom.

Recursos



Temas (1)



```
1 <resources xmlns:tools="http://schemas.android.com/tools">
2     <!-- Base application theme. -->
3     <style name="Base.Theme._1HolaMundo" parent="Theme.Material3.DayNight.NoActionBar">
4         <!-- Customize your light theme here. -->
5         <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
6     </style>
7
8     <style name="Theme._1HolaMundo" parent="Base.Theme._1HolaMundo" />
9 </resources>
```

Theme.Material3.DayNight.NoActionBar

Running Devices: Pixel 6a API 27

1-Holamundo > app > src > main > res > values > themes.xml

Temas (2)



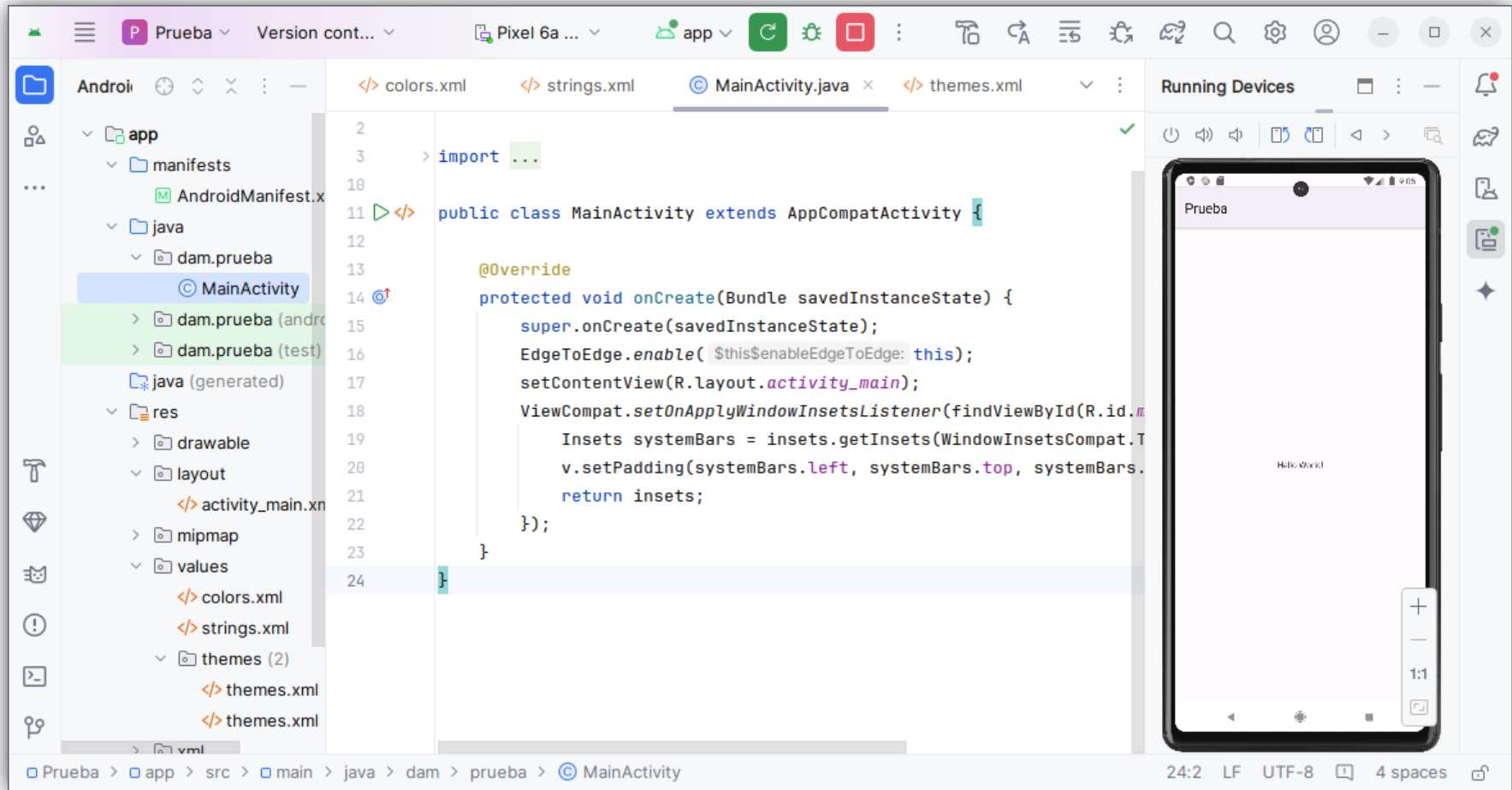
```
1 <resources xmlns:tools="http://schemas.android.com/tools">
2   <!-- Base application theme. -->
3   <style name="Base.Theme._1HolaMundo" parent="Theme.Material3.DayNight">
4     <!-- Customize your light theme here. -->
5     <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
6   </style>
7
8   <style name="Theme._1HolaMundo" parent="Base.Theme._1HolaMundo" />
9 </resources>
```

Theme.Material3.DayNight

Running Devices: 1 Hola mundo

1:1

Código Java

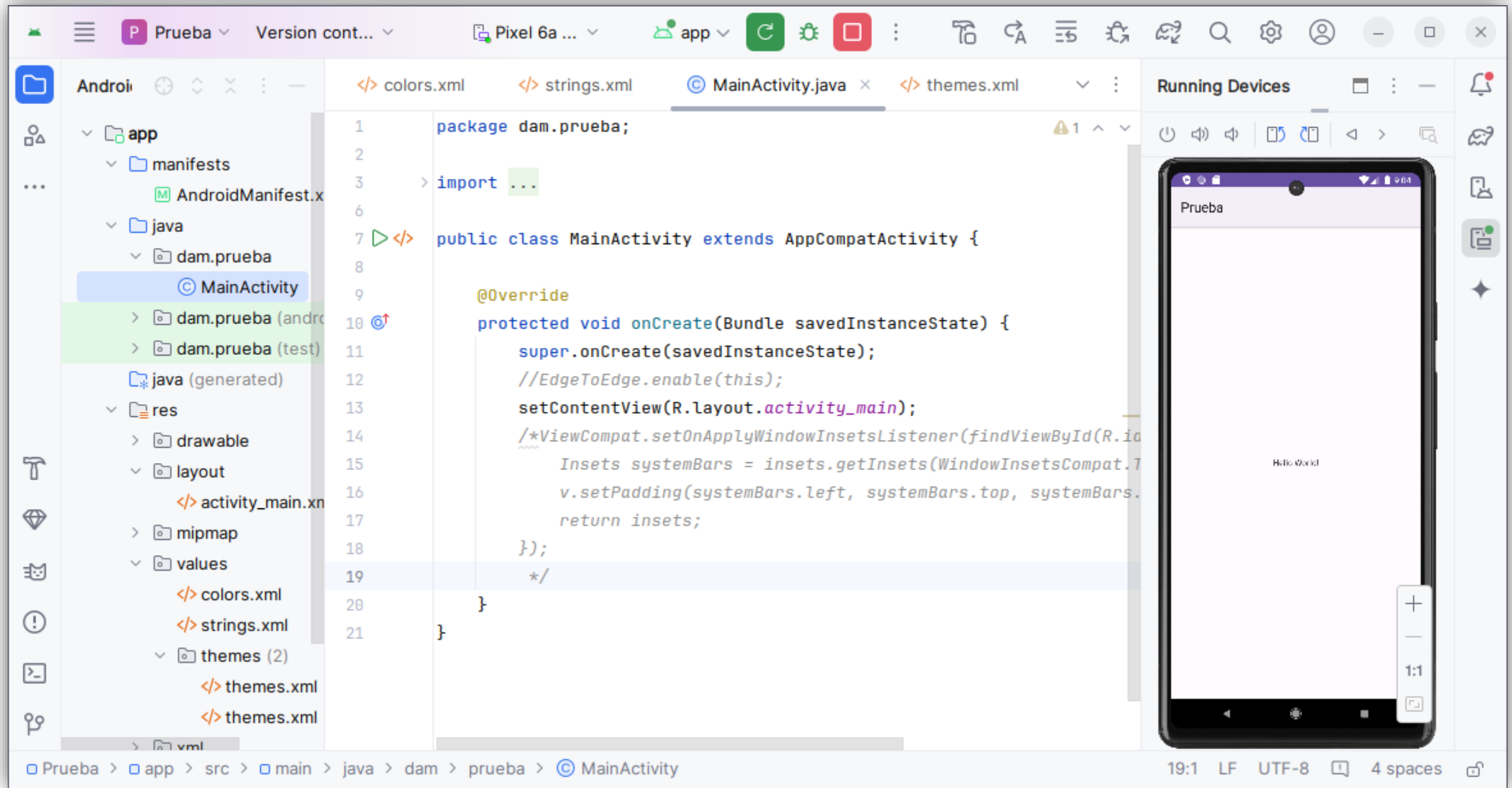


The screenshot displays the Android Studio interface. The central editor shows the `MainActivity.java` file with the following code:

```
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);  
17          setContentView(R.layout.activity_main);  
18          ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.m  
19              Insets systemBars = insets.getInsets(WindowInsetsCompat.T  
20              v.setPadding(systemBars.Left, systemBars.top, systemBars.  
21              return insets;  
22          }  
23      }  
24  }
```

The right-hand side of the interface shows the 'Running Devices' tab with a virtual Pixel 6a smartphone. The app is titled 'Prueba' and the screen displays 'Hello World'. The bottom status bar of the emulator shows the time as 24:2, the layout as LF, the encoding as UTF-8, and 4 spaces.

Barra de estado y de navegación sin heredar el fondo de la APP



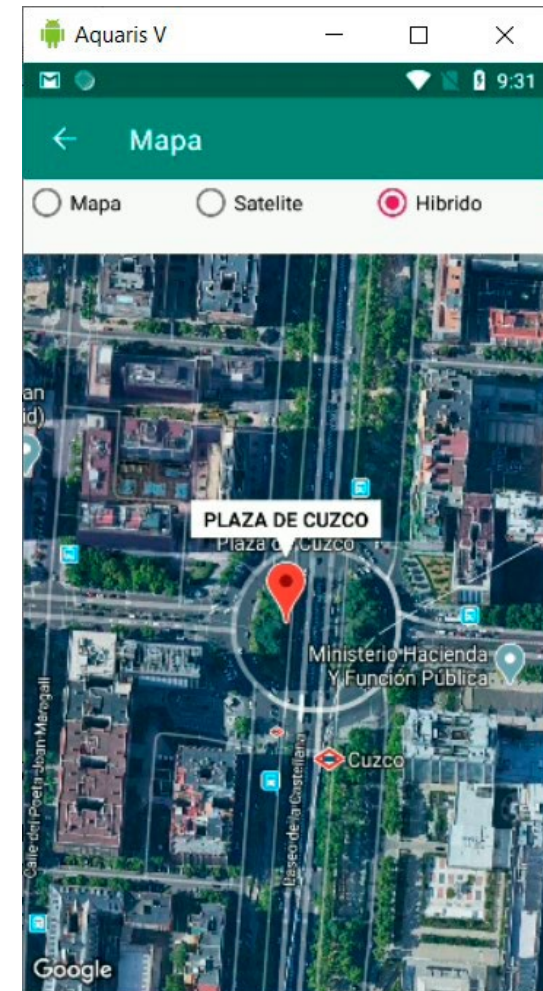
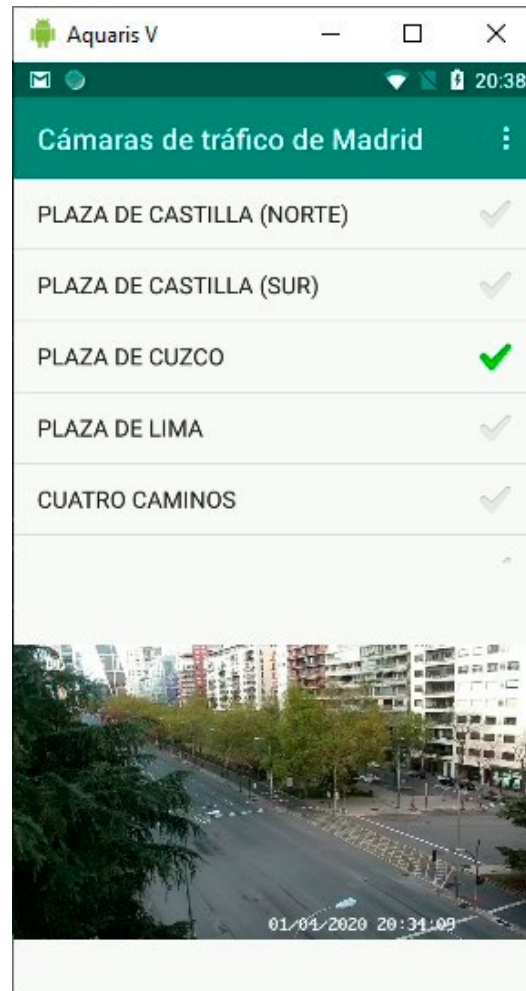
The screenshot displays an IDE interface for an Android application. The main editor shows the `MainActivity.java` file with the following code:

```
1 package dam.prueba;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        //EdgeToEdge.enable(this);
13        setContentView(R.layout.activity_main);
14        /*ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.
15        Insets systemBars = insets.getInsets(WindowInsetsCompat.T
16        v.setPadding(systemBars.left, systemBars.top, systemBars.
17        return insets;
18    });
19    */
20 }
21 }
```

The `Running Devices` panel on the right shows a virtual device named "Prueba" with a white screen displaying "Hello World". The status bar at the top of the device is white, and the navigation bar at the bottom is black. The IDE interface includes a file explorer on the left, a toolbar at the top, and a breadcrumb at the bottom: `Prueba > app > src > main > java > dam > prueba > MainActivity`. The status bar at the bottom of the IDE shows `19:1 LF UTF-8 4 spaces`.

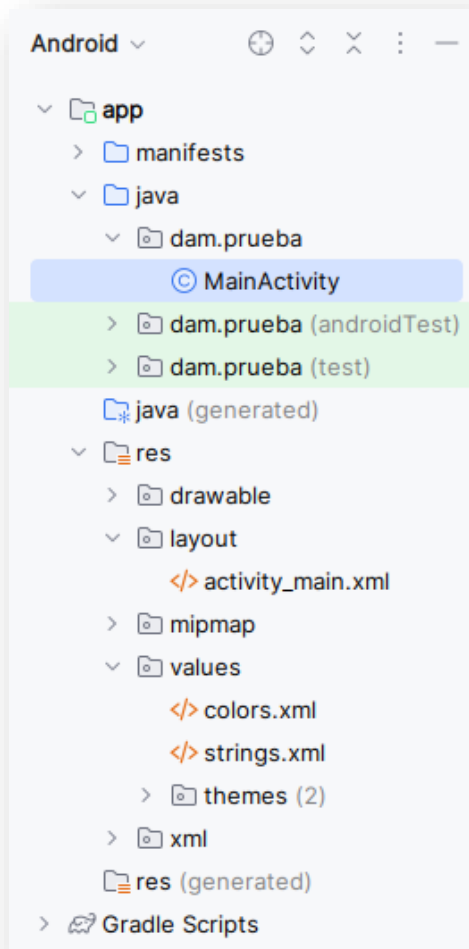
Actividades

- ❑ Una aplicación tiene al menos una actividad.
- ❑ Desde el punto de vista del usuario, una actividad se asocia a una *pantalla* de la aplicación.
- ❑ Por ejemplo:
 - Actividad inicial con un listado de calles.
 - Actividad nueva en respuesta a pulsar una calle.
- ❑ En las sesiones siguientes se estudiará el ciclo de vida de una actividad.

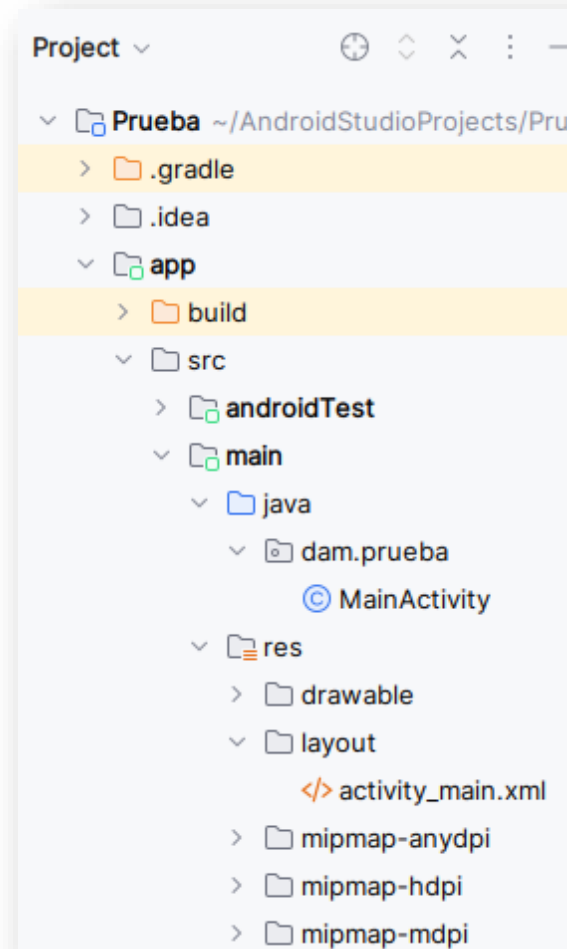


Correspondencia con el sistema de ficheros

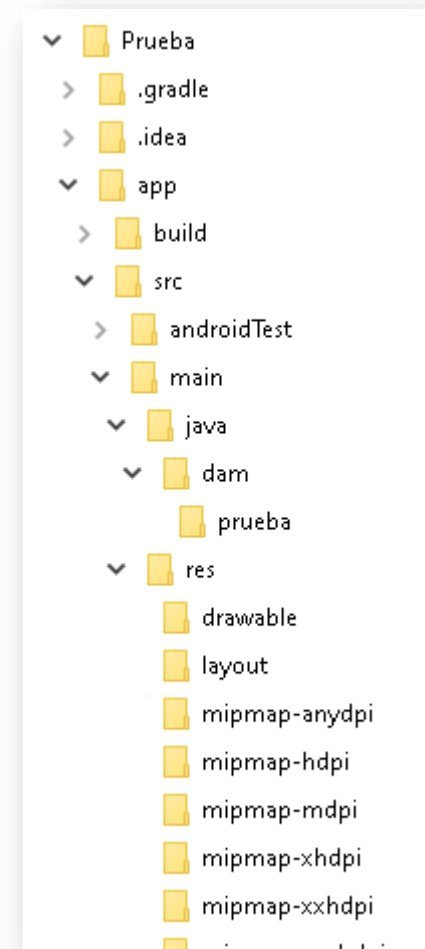
Vista Android



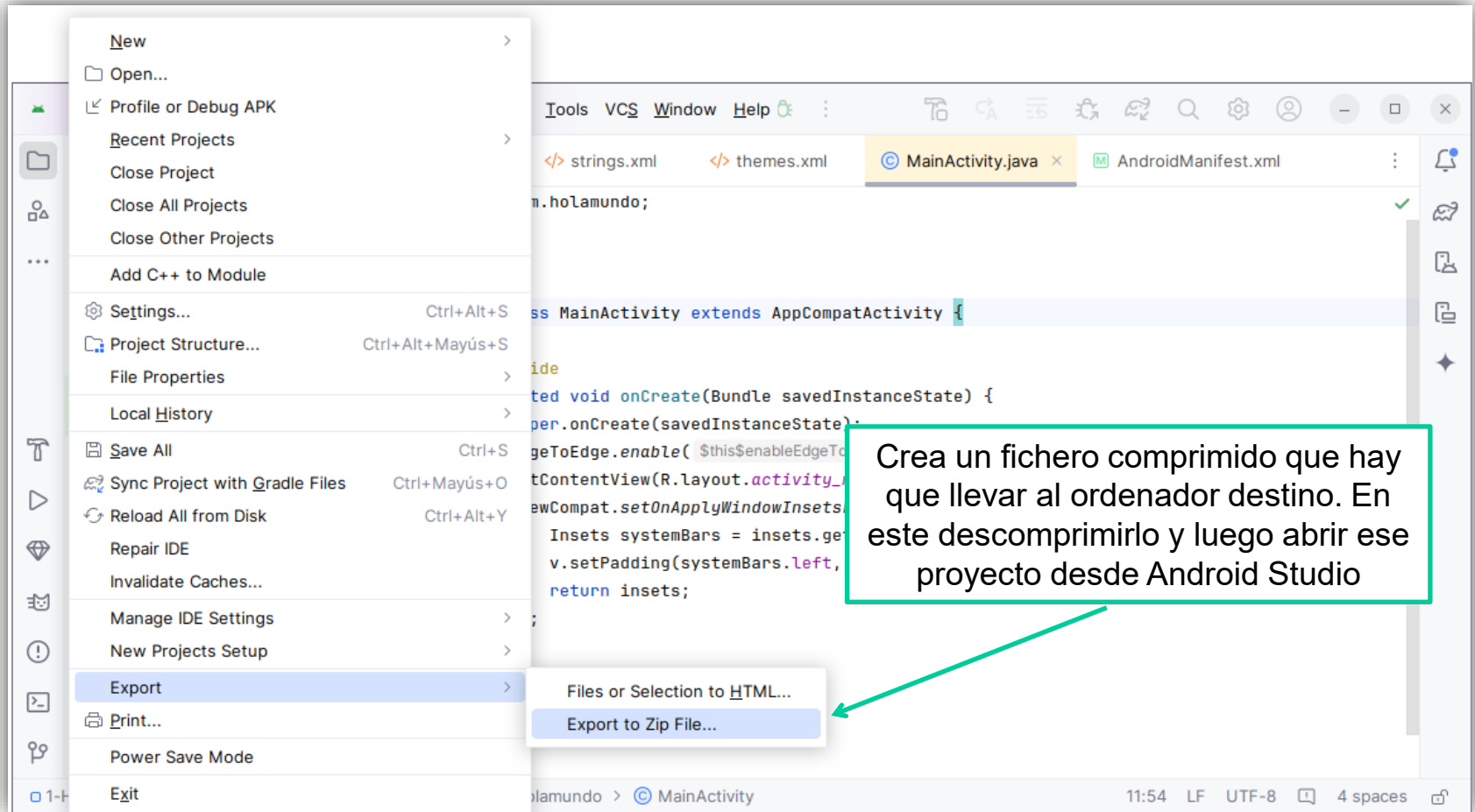
Vista Proyecto



Sistema de ficheros



Como llevar proyectos de un ordenador a otro



The screenshot shows the Android Studio interface with the 'Export' menu option highlighted. A callout box with a green border and arrow points to the 'Export to Zip File...' option in the sub-menu. The callout text reads: 'Crea un fichero comprimido que hay que llevar al ordenador destino. En este descomprimirlo y luego abrir ese proyecto desde Android Studio'.

```
Tools VCS Window Help :
</> strings.xml </> themes.xml MainActivity.java x AndroidManifest.xml
m.ho!amundo;
ss MainActivity extends AppCompatActivity {
ide
ted void onCreate(Bundle savedInstanceState) {
per.onCreate(savedInstanceState):
geToEdge.enable( $this$enableEdgeTo
tContentView(R.layout.activity_
ewCompat.setOnApplyWindowInsets
Insets systemBars = insets.ge
v.setPadding(systemBars.left,
return insets;
;
lamundo > MainActivity 11:54 LF UTF-8 4 spaces
```

Ejercicio 4 (1)

Modifica el proyecto del ejercicio 3:

- Cambia el tema para que se vea la barra de título
- Comenta o borra las dos instrucciones que extienden la interfaz a toda la pantalla
- Cambia el nombre de la aplicación a “1.1 Hola Mundo”. Comprueba que ha cambiado en la barra de título y en el menú de aplicaciones
- Cambia el texto “Hello World!” y su estilo (color, tipo de letra, tamaño, ...) en el fichero `activity_main.xml`
- Cambia el layout *ConstraintLayout* a un *LinearLayout* vertical con atributo *gravity* centrado
- Inserta otros mensajes usando *TextView*

Los atributos de *TextView* que necesitas para cambiar el tipo de letra, tamaño, estilo, alineación, etc. son:

`layout_marginTop/Right/Left`, `gravity`, `textColor`, `textSize`, `textStyle` y `typeface`



Ejercicio 4 (2)

- ❑ Cambia el icono de la APP. Ver como se hace en [Image Asset Studio](#). Google proporciona iconos gratuitos [aquí](#)
- ❑ En el fichero MainActivity.java inserta un [Toast](#) al final del método `onCreate()`.
Un *Toast* proporciona un mecanismo sencillo para imprimir un mensaje en pantalla que desaparece de forma automática

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    : : : :  
    Toast.makeText(this, "Cuando arranco salgo", Toast.LENGTH_SHORT).show();  
}
```

