



**Universidad Nacional Autónoma de
México**

Facultad de Contaduría y Administración



Desarrollo de Aplicaciones Móviles

Profesor: Cristian Cardoso Arellano

Alumno: Ortega Maldonado Diego Daniel

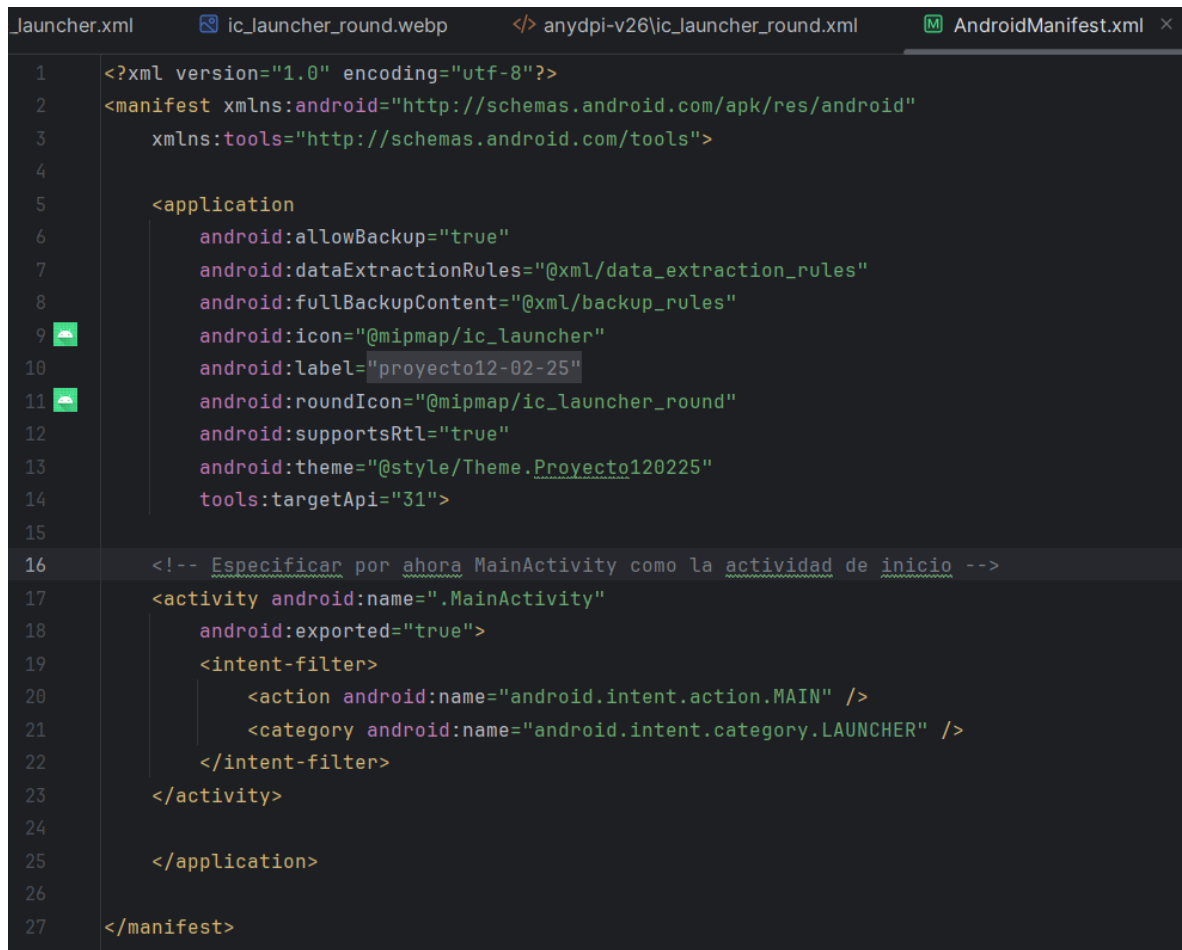
Actividad M2-01

Fecha de entrega: 17 de Febrero, 2025

1. Ingresar el código de programación para imprimir en el log “Hello World”.

Para poder hacer la impresión del hello world se requiere hacer algunas modificaciones:

La primera modificación es especificar en el archivo manifest qué actividad será la ejecutada por default. Para motivos de la actividad, especificaremos que será la impresión del mensaje, contenida en una clase Java llamada MainActivity.



```
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="proyecto12-02-25"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportRtl="true"
13         android:theme="@style/Theme.Proyecto120225"
14         tools:targetApi="31">
15
16         <!-- Especificar por ahora MainActivity como la actividad de inicio -->
17         <activity android:name=".MainActivity"
18             android:exported="true">
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21                 <category android:name="android.intent.category.LAUNCHER" />
22             </intent-filter>
23         </activity>
24
25     </application>
26
27 </manifest>
```

Lo siguiente es crear un nuevo layout llamado activity_main con extensión .xml. Su contenido es el siguiente:

```

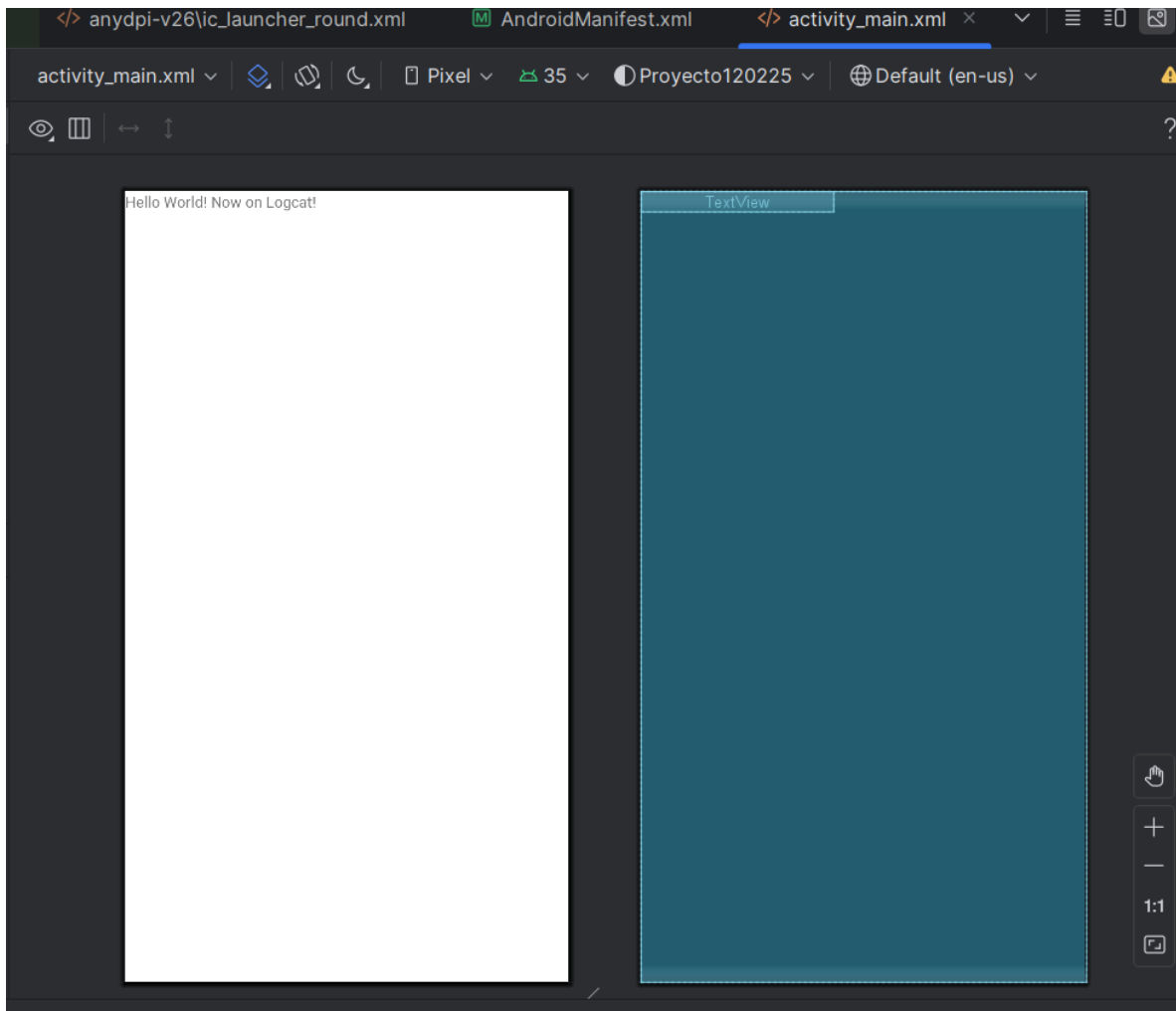
MINGW64:/d:/GitHub/Moviles2025-2/12-02-25/app/src/main/res/layout
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World! Now on Logcat!" />

</LinearLayout>
~

```

Su visualización en Android Studio es la siguiente:



Finalmente, creamos la clase que permitirá que al ejecutar la aplicación se muestre el hello world tanto en el dispositivo como en el logcat, que es lo que nos interesa.

```
© MainActivity.java ×
1 package com.example.proyecto12_02_25;
2
3 import android.os.Bundle;
4 import android.util.Log;
5 import androidx.appcompat.app.AppCompatActivity;
6
7 ▶ </> public class MainActivity extends AppCompatActivity {
8
9     //Le ponemos un nombre a nuestro log.
10    1 usage
11    private static final String TAG = "MiApp";
12
13    @Override
14    protected void onCreate(Bundle savedInstanceState) {
15        super.onCreate(savedInstanceState);
16        setContentView(R.layout.activity_main);
17
18        //Este mensaje aparecerá en logcat.
19        Log.d(TAG, msg: "Hello, World! Now on logcat!");
20    }
21 }
22
```

2. Buscar la impresion del “hello world” en la ventana “logcat” dentro de Android Studio.

Al haber ejecutado la clase, que recordemos será la actividad principal por ahora, entramos a la pestaña logcat para buscar nuestro hello world. El mensaje se verá así:

```
Logcat
Medium Phone API 35 (emulator-5554) Android 15, API level 35 package:mine

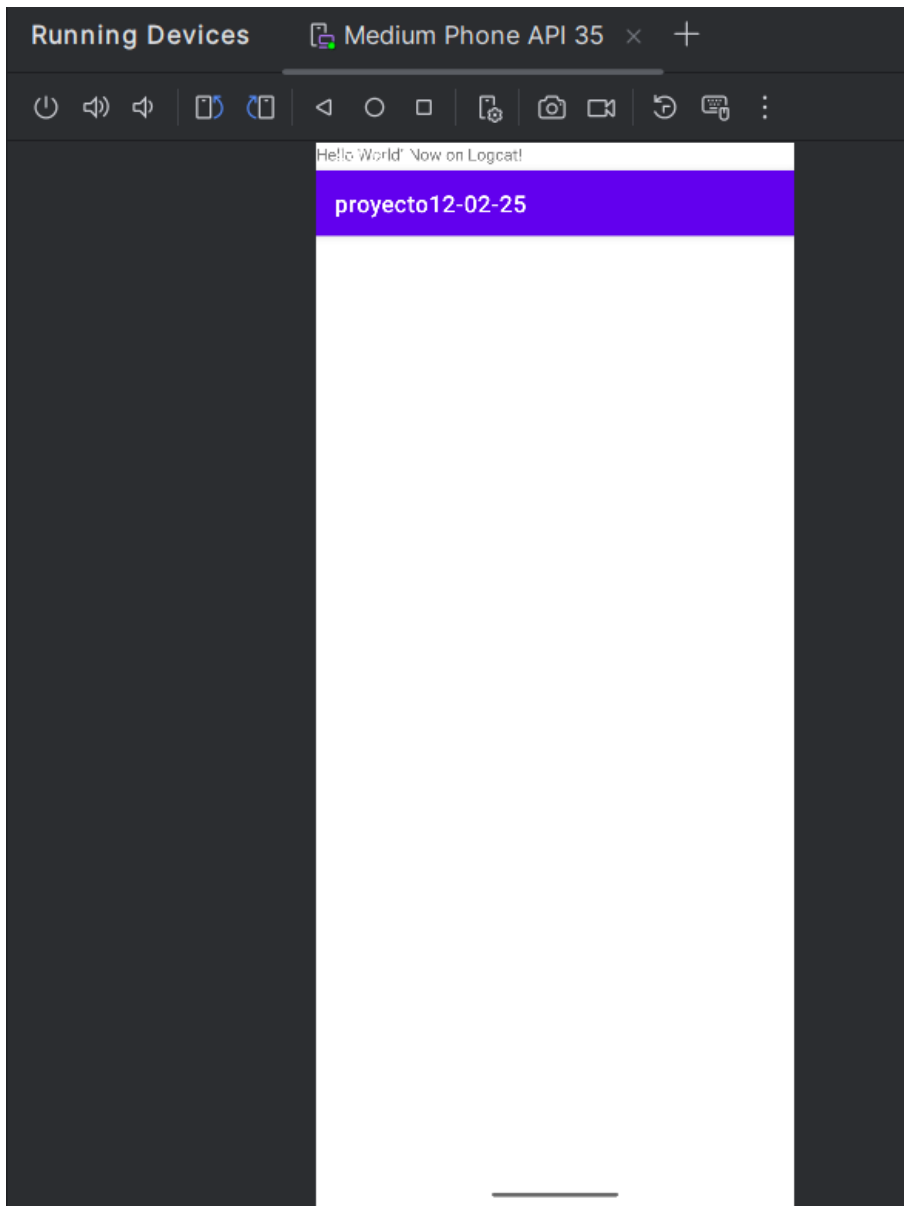
2025-02-16 22:38:13.340 1658-1658 GraphicsEnvironment com.example.projeto12_02_25
2025-02-16 22:38:13.341 1658-1658 GraphicsEnvironment com.example.projeto12_02_25
2025-02-16 22:38:13.491 1658-1658 CompatChangeReporter com.example.projeto12_02_25
2025-02-16 22:38:14.035 1658-1658 AppCompatDelegate com.example.projeto12_02_25
2025-02-16 22:38:14.369 1658-1658 CompatChangeReporter com.example.projeto12_02_25
2025-02-16 22:38:14.687 1658-1658 proyecto12_02_25 com.example.projeto12_02_25
2025-02-16 22:38:14.649 1658-1658 CompatChangeReporter com.example.projeto12_02_25
2025-02-16 22:38:14.712 1658-1658 MiApp com.example.projeto12_02_25
2025-02-16 22:38:15.010 1658-1658 HWUI com.example.projeto12_02_25
2025-02-16 22:38:15.117 1658-1658 Choreographer com.example.projeto12_02_25
2025-02-16 22:38:15.255 1658-1832 EGL_emulation com.example.projeto12_02_25
2025-02-16 22:38:15.255 1658-1832 EGL_emulation com.example.projeto12_02_25
2025-02-16 22:38:15.308 1658-1832 HWUI com.example.projeto12_02_25
2025-02-16 22:38:15.308 1658-1832 HWUI com.example.projeto12_02_25
2025-02-16 22:38:15.461 1658-1832 Gralloc4 com.example.projeto12_02_25
2025-02-16 22:38:15.902 1658-1680 HWUI com.example.projeto12_02_25
2025-02-16 22:38:15.960 1658-1658 Choreographer com.example.projeto12_02_25
2025-02-16 22:38:16.029 1658-1658 HWUI com.example.projeto12_02_25

EGLSettings values are invalid. Number of packages: 0, number of vo
Neither updatable production driver nor prerelease driver is supported.
D Compat change id reported: 279446685; UID 10209; state: ENABLED
D Checking for metadata for AppLocalesMetadataHolderService : Service not
D Compat change id reported: 309578419; UID 10209; state: ENABLED
W Accessing hidden method Landroid/view/ViewGroup;->makeOptionalFitsSystem
D Compat change id reported: 63938206; UID 10209; state: ENABLED
D Hello, World! Now on logcat!
W Unknown dataspace 0
I Skipped 99 frames! The application may be doing too much work on its m
I Opening libGLESv1_CM.emulation.so
I Opening libGLESv2_emulation.so
W Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying with
W Failed to initialize 101010-2 format, error = EGL_SUCCESS
I mapper 4.x is not supported
I Davey! duration=2392ms; Flags=1, FrameTimelineVsyncId=10321, IntendedV
I Skipped 45 frames! The application may be doing too much work on its m
I Davey! duration=782ms; Flags=0, FrameTimelineVsyncId=10501, IntendedV
```

Haciendo un poco más de zoom:

```
2025-02-16 22:38:14.712 1658-1658 MiApp com.example.projeto12_02_25 D Hello, World! Now on logcat!
2025-02-16 22:38:15.010 1658-1658 HWUI com.example.projeto12_02_25 W Unknown dataspace 0
2025-02-16 22:38:15.117 1658-1658 Choreographer com.example.projeto12_02_25 I Skipped 99 frames! The applicatio
```

Como extra, la aplicación se ve así:



Conclusión.

La impresión de un Hello World clásico cambia demasiado cuando lo queremos pasar al entrono móvil. En otros casos hubiese bastado hacerlo con un `System.out.println("Hello, world!");` y ya, pero en este caso al tratarse de algo que deberá aparecer en logcat, que recordemos es una bitácora de procesos relativos a la aplicación que desarrollemos, forzosamente se requiere que esté dentro de la ejecución de la misma. Esta actividad ha servido para familiarizarnos un poco con los archivos xml que se necesitan para poder imprimir el mensaje, y también sirve como introducción a la lectura de mensajes en logcat.