



Actividad [#1] - [Pantalla de Inicio]

[Desarrollo de Aplicaciones Móviles I]

Ingeniería en Desarrollo de Software

Tutor: Eduardo Israel Castillo García

Alumno: Alan David López Rojas

Fecha: 14/11/2022

Índice

Introducción.....	pág. 3
Descripción.....	pág. 3
Justificación.....	pág. 4
Investigación.....	pág. 4
Desarrollo.....	pág. 7
Interfaz.....	pág. 7
Codificación.....	pág. 16
Prueba de la aplicación.....	pág. 19
Conclusión.....	pág. 23
Bibliografía.....	pág. 24

Introducción

Las aplicaciones móviles se han convertido en un fenómeno mundial de desarrollo de software y tecnología para las telecomunicaciones, el uso de aplicaciones móviles se ha centrado actualmente en el desarrollo de plataformas de distribución que se manejan con dispositivos móviles compatibles con diferentes entornos a través de sus sistemas operativos.

Por tal razón para la actividad 1 de la unidad 2 se plantea el problema: La unidad de negocio enfocada a servicios bancarios solicita una aplicación móvil. Eres el encargado del Departamento de Sistemas, por lo que deberás crearla. La aplicación debe contar con lo siguiente:

Requerimientos de interfaz:

- Pantalla de inicio con los siguientes elementos:
 - o Fondo degradado (colores #FFFFFF y #CCCCCC)
 - o Logo con animación. Este debe moverse de abajo hacia arriba en la pantalla (Splash Screen)

Requerimientos funcionales:

- Después de dos segundos de haber cargado la pantalla de inicio, se debe redireccionar automáticamente a la pantalla de autenticación (segunda pantalla), por lo que deberás configurar esto en esta parte de la aplicación.

Descripción

Este proyecto pretende crear una aplicación que permita a los usuarios controlar los servicios financieros de forma sencilla e intuitiva que llevará el nombre de banco_movil, por ejemplo, el uso que pudiera tener una aplicación de este tipo sería mediante la capacidad de poder consultar su saldo actualizado en todo momento, registrar y categorizar gastos, visualizar los gastos registrados, etc., todo esto una vez se haya identificado el usuario y pueda acceder a su cuenta con sus datos personales, para ello empezaremos con la pantalla de inicio.

El primer paso es crear una pantalla de inicio que tendrá una animación la cual será la que de bienvenida al usuario para posteriormente pasar a la segunda pantalla de autenticación donde el usuario ingresará con sus datos y dará autorización para la cuenta de ese cliente.

Justificación

Debido a la evolución de los mercados y la aparición de las nuevas tecnologías bancarias, que ha llegado para aportar eficiencia a la oferta y demanda de bienes y servicios, la unidad de negocio enfocada a servicios bancarios solicita una aplicación móvil, para el manejo de cuentas para los clientes, por lo que para iniciar este proyecto se tiene que empezar con la pantalla de inicio de la aplicación, la cual se creara con la ayuda de Android Studio, este software fue diseñado para la creación de aplicaciones móviles con sistema operativo Android, es intuitivo y simple de manejar, además permite ejecutar las compilaciones de forma muy rápida, y así poder comprobar en el momento los fallos y mejoras de la aplicación, por eso es la mejor opción para este proyecto.

Investigación

¿Qué es Android Studio?

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android

El sistema operativo de código abierto (open source), android, principalmente es programado en el lenguaje de programación Java y su núcleo se basa en Linux. Android fue comprado en el año 2005 por

Google y no sólo se instala en teléfonos celulares, sino también en tabletas, televisores, relojes...entre otros, por lo que existen muchas posibilidades para su empleo

Por supuesto, Android Studio es un entorno de desarrollo multiplataforma, y es posible utilizarlo en Windows, MacOS, Linux y ChromeOS.

¿Qué se puede crear en Android Studio?

Una de las capacidades más potentes de este software es su emulador. Con él es posible probar las aplicaciones en desarrollo y funciones del sistema operativo, como si en un teléfono estuvieran instaladas. La parte más interesante de usar el emulador es que se puede elegir cualquier versión del sistema operativo de Google y comprobar cómo se comporta la aplicación en determinados terminales.

No es obligatorio ser el desarrollador de las aplicaciones para poder analizarlas. Es posible usar el trabajo de otros creadores, a través de aplicaciones de código abierto. Con ellas podemos aprender cómo se han diseñado y tomar prestado parte de su código.

Además del emulador contamos con la posibilidad de ejecutar nuestras aplicaciones en teléfonos de verdad, en tiempo real. Para conseguirlo se hace uso de la llamada depuración USB. No existe mejor manera de probar que todo está funcionando de forma correcta, con aspectos que no se pueden emular como la pantalla táctil.

- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.

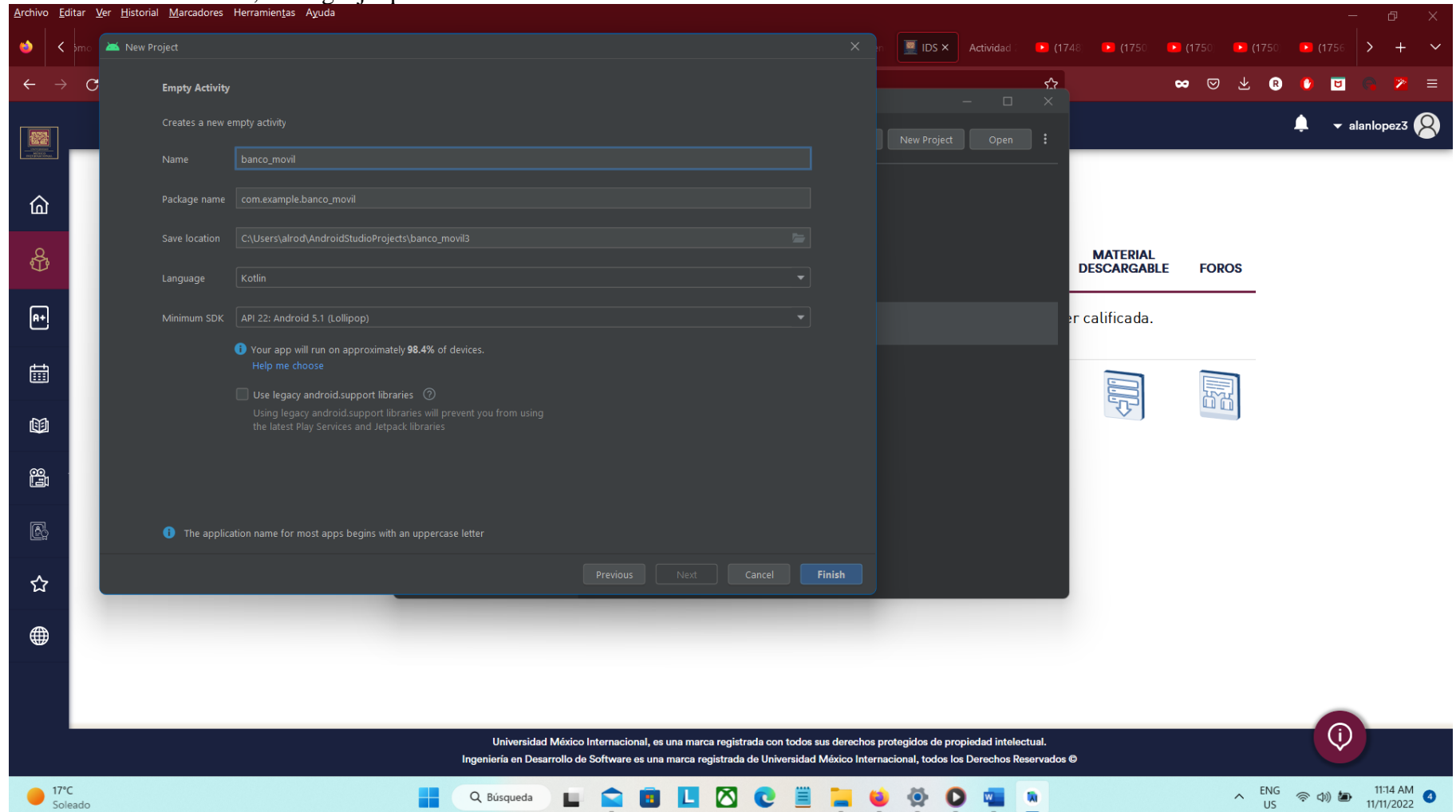
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- Integración de ProGuard y funciones de firma de aplicaciones.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario.
- Soporte para programar aplicaciones para Android Wear.
- Soporte integrado para Google Cloud Platform, que permite la integración con Firebase Cloud Messaging (antes 'Google Cloud Messaging') y Google App Engine.
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.
- Renderizado en tiempo real.
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.

Android Studio admite los mismos lenguajes de programación de IntelliJ (y CLion), como Java, C++ y más con extensiones, como Go; y Android Studio 3.0 o posterior es compatible con Kotlin y "todas las características de lenguaje Java 7 y un subconjunto de características de lenguaje Java 8 que varían según la versión de la plataforma". Los proyectos externos soportan algunas características de Java 9. Si bien IntelliJ en el que se basa Android Studio admite todas las versiones de Java lanzadas, y Java 12, no está claro en qué nivel Android Studio admite versiones de Java hasta Java 12 (la documentación menciona el soporte parcial de Java 8). Al menos algunas características nuevas del lenguaje hasta Java 12 se pueden usar en Android.

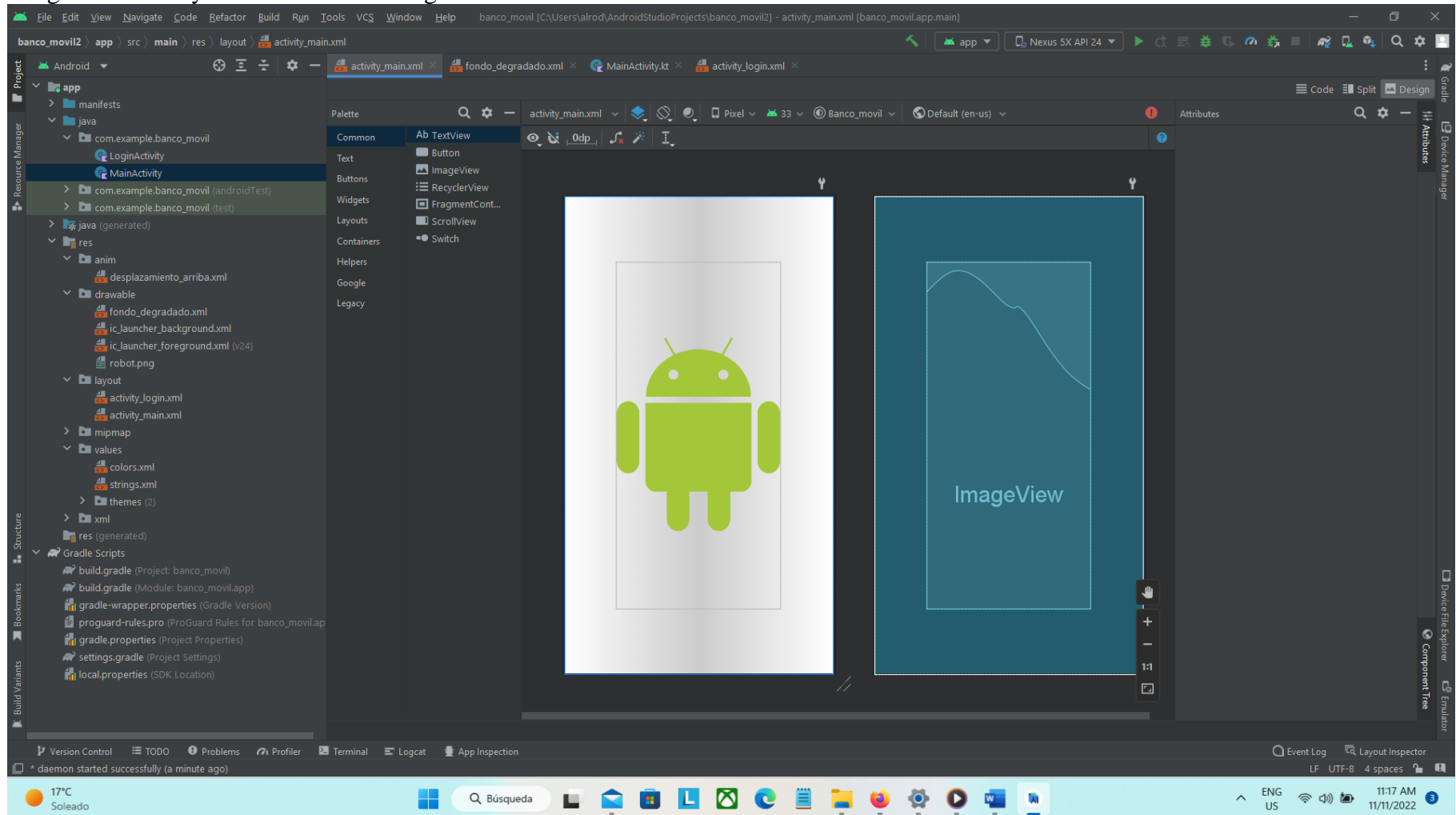
Desarrollo

Interfaz

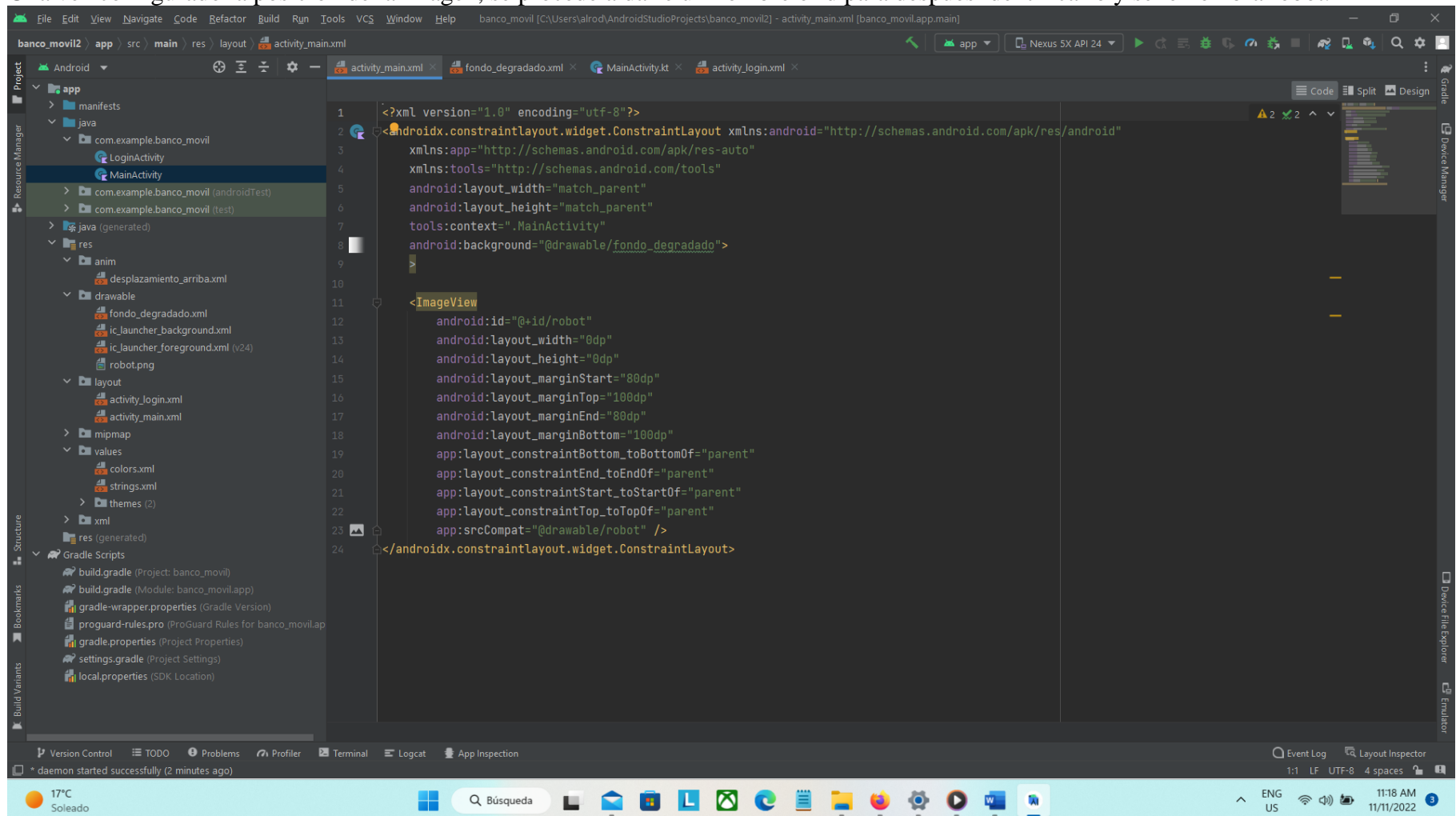
Como primer paso, se hace la creación del proyecto que llevara por nombre `banco_movil`, con la selección de la versión de Android, la localización del archivo, el lenguaje que será kotlin.



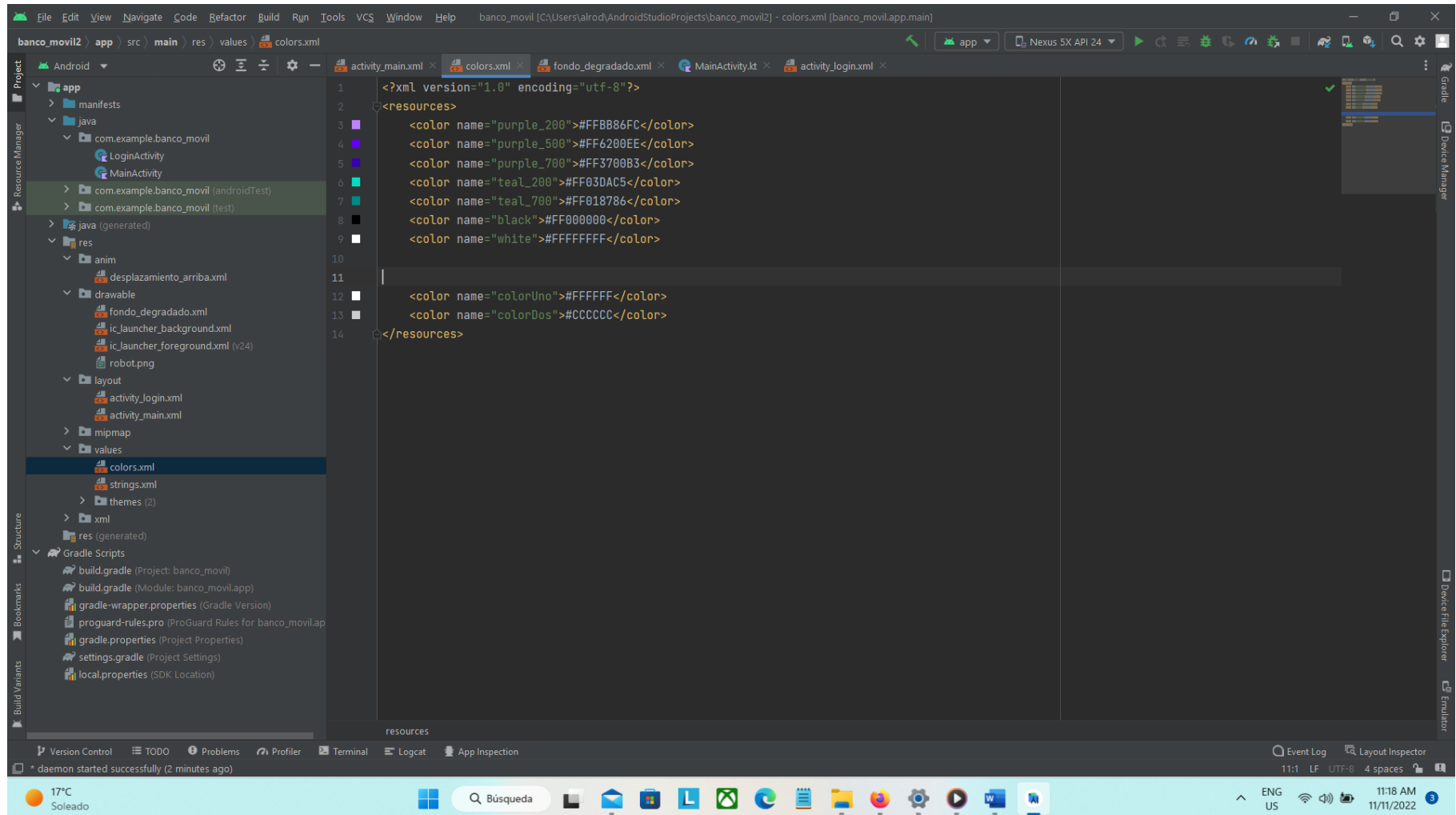
Una vez creado el archivo, se procede a insertar la imagen Android que se proporcionó, se guardará en la carpeta del proyecto “drawable” para poder usarlo posteriormente, en la pestaña activity_main que será la pantalla de inicio, se inserta la imagen y se le ajusta tamaño y márgenes con la ayuda de constraint widget.



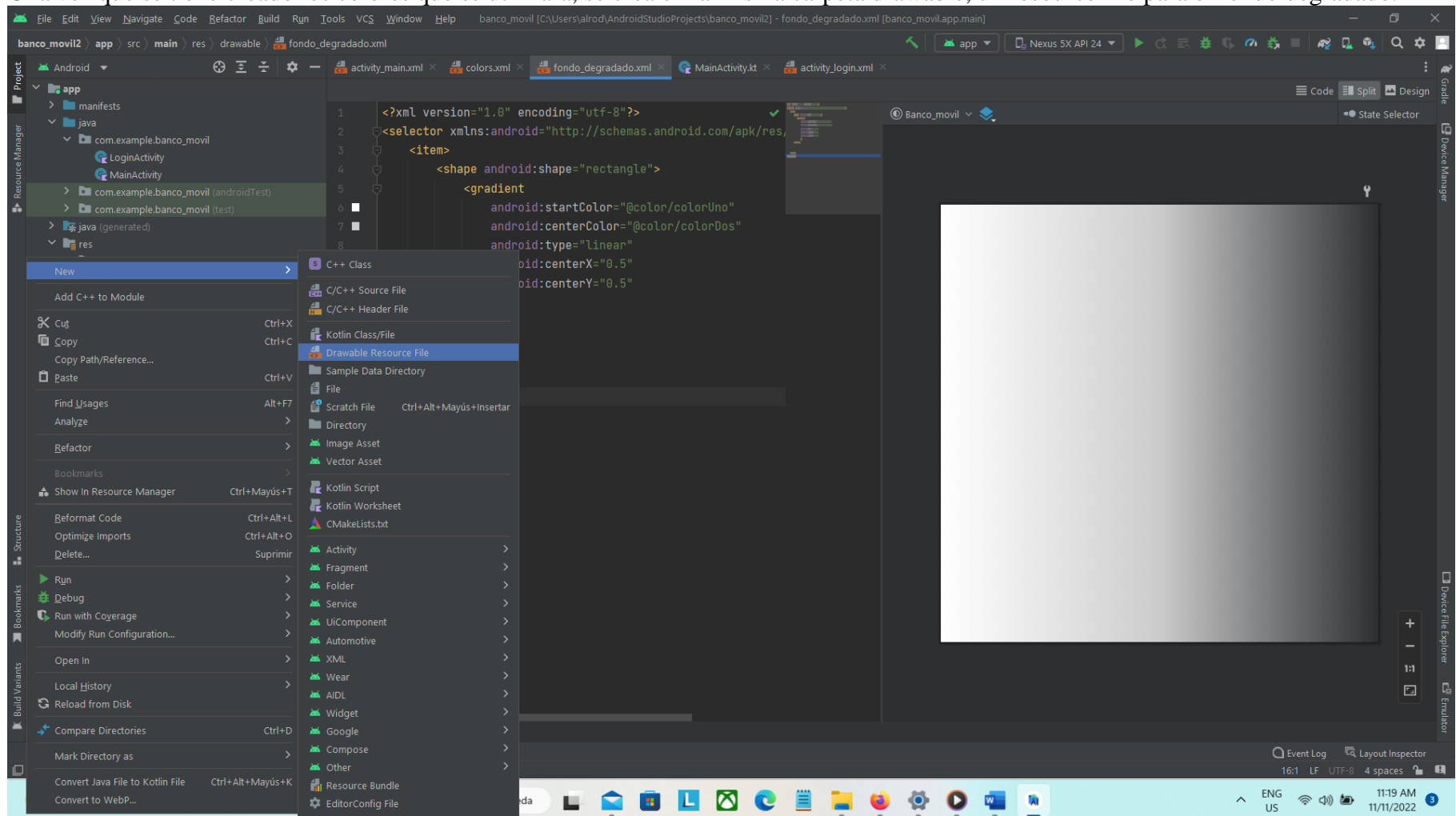
Una vez configurado la posición de la imagen, se procede a darle un nombre o Id para después identificarlo y se le nombra robot.



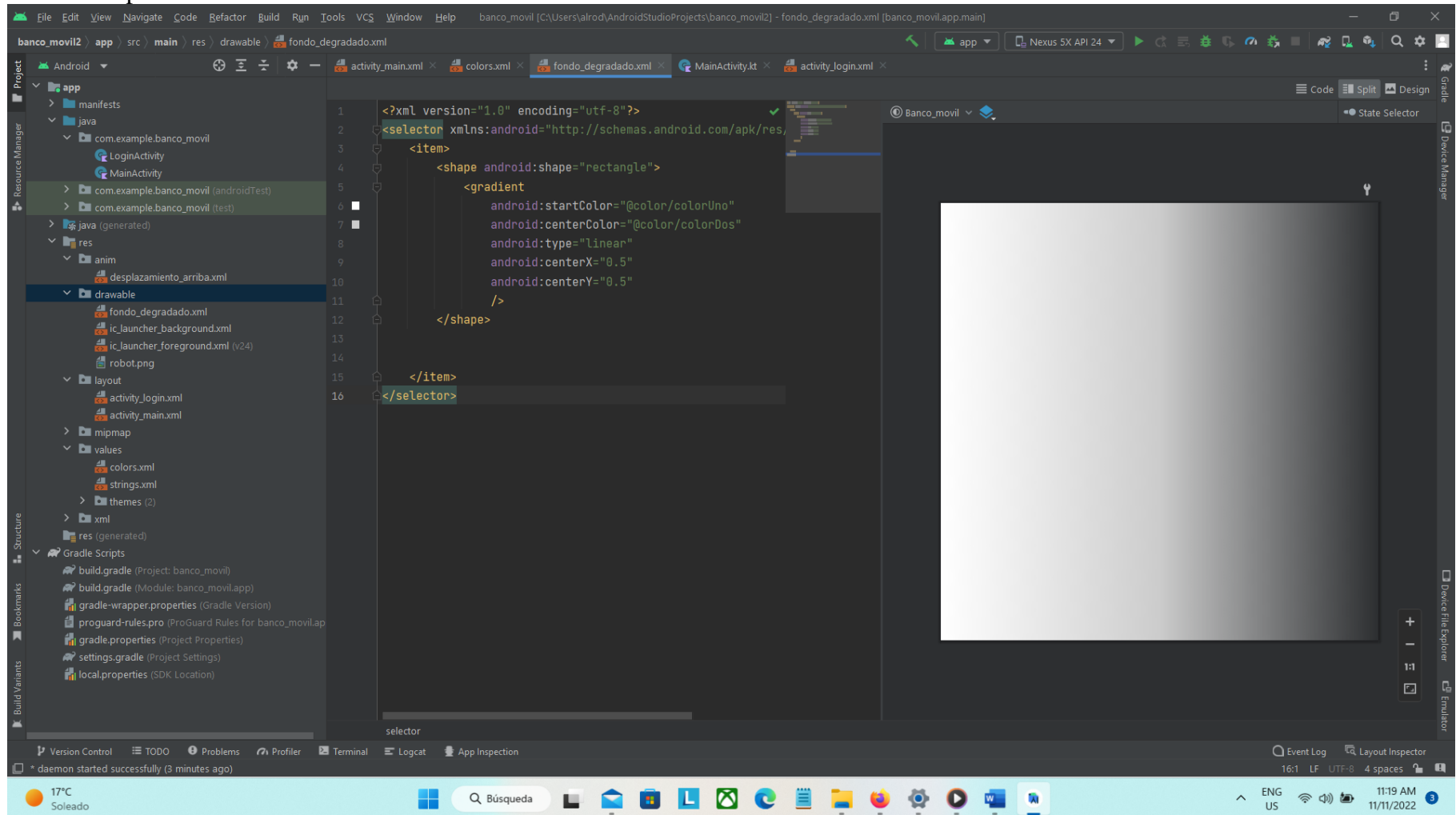
Ahora se crea en la carpeta de drawable, en el archivo colors.xml, los colores que se solicita en la actividad, nombrándolos colorUno y colorDos.



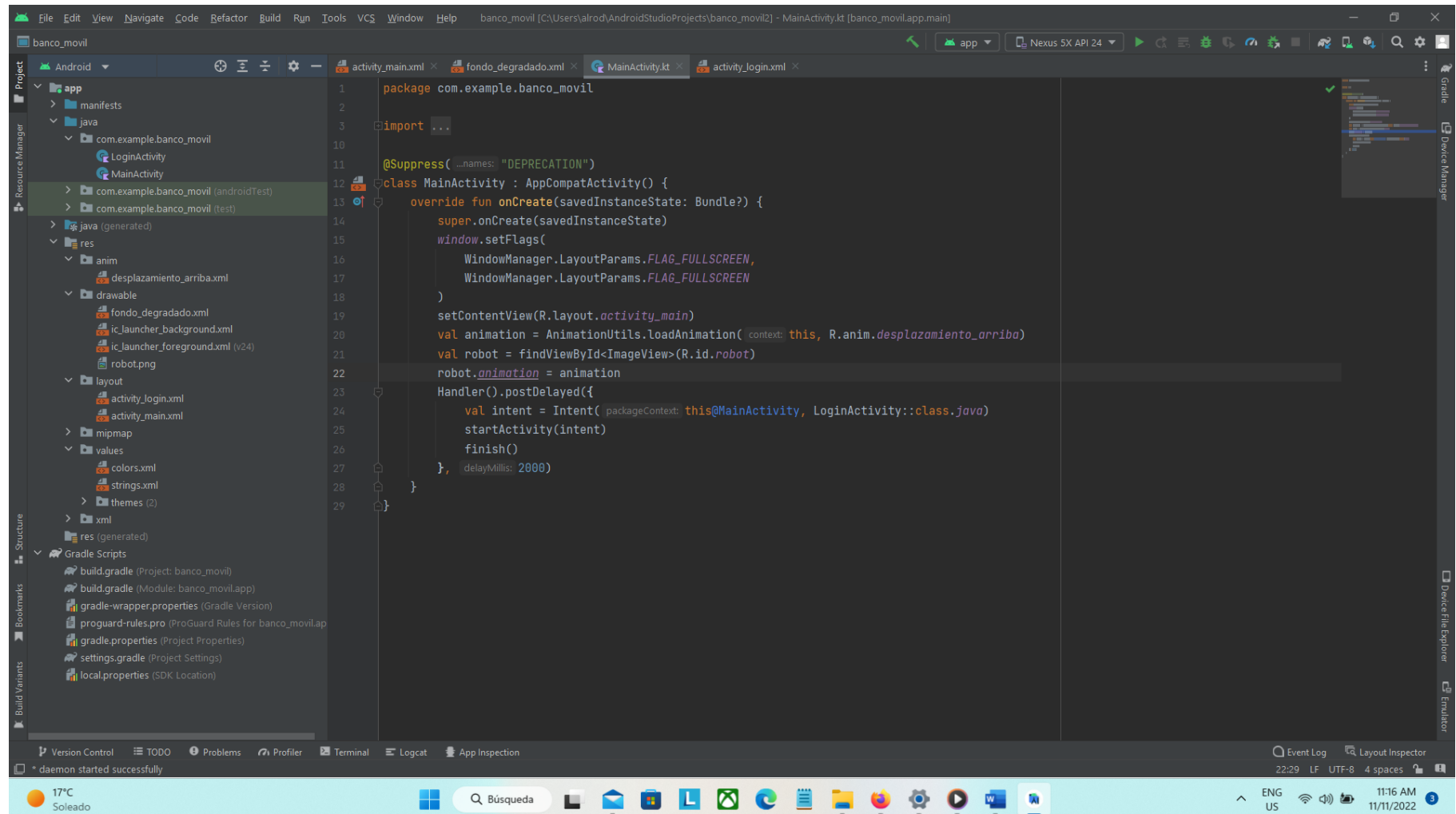
Una vez que se tiene creado los colores que se utilizara, se crea en la misma carpeta drawable, un resource file para el fondo degradado.



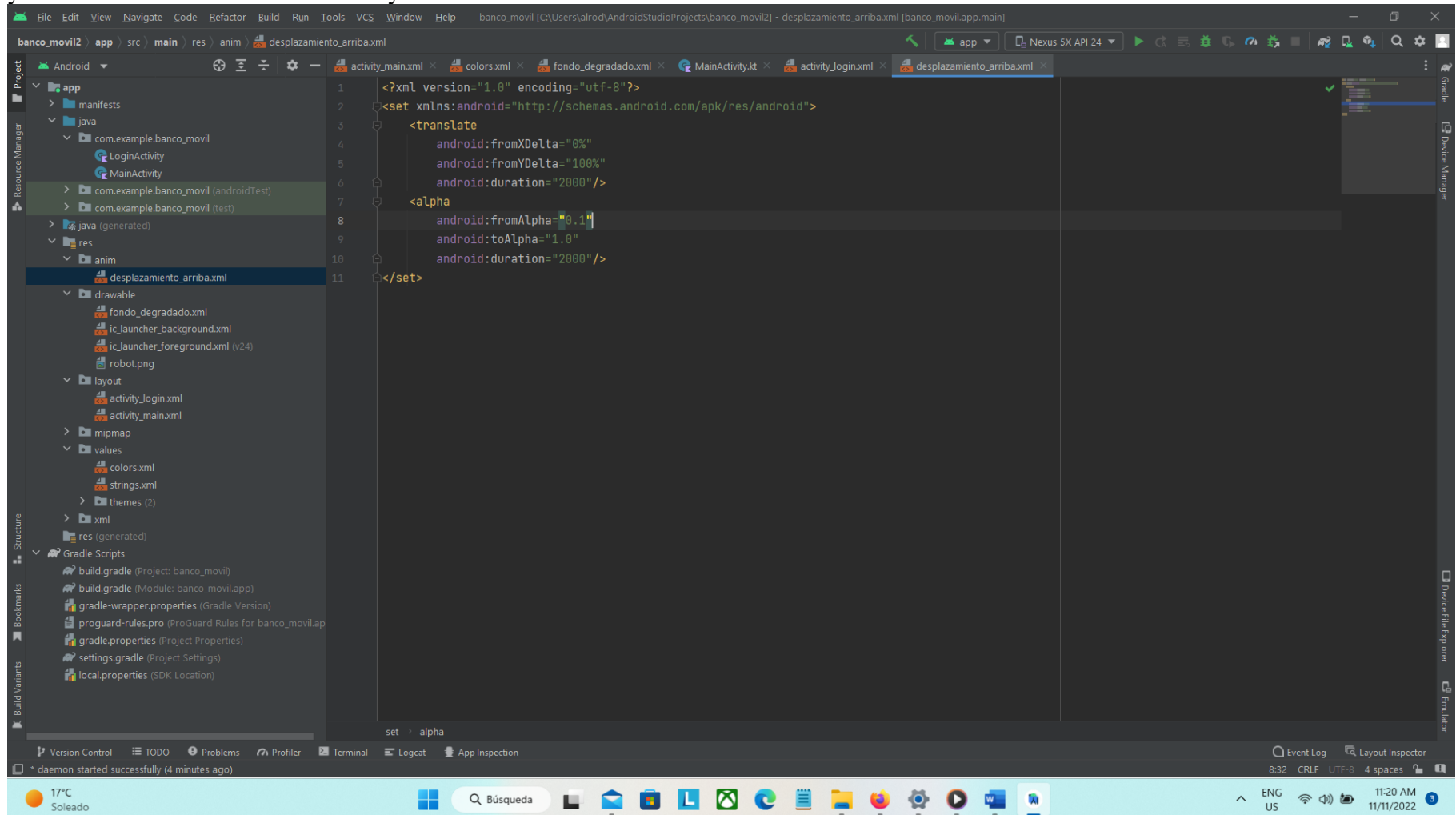
Este archivo se llamará fondo_degradado, aquí se programa como funcionará el fondo degradado y que tipo de degradado se usará, se puede ver la vista previa de como lucirá en el lado derecho.



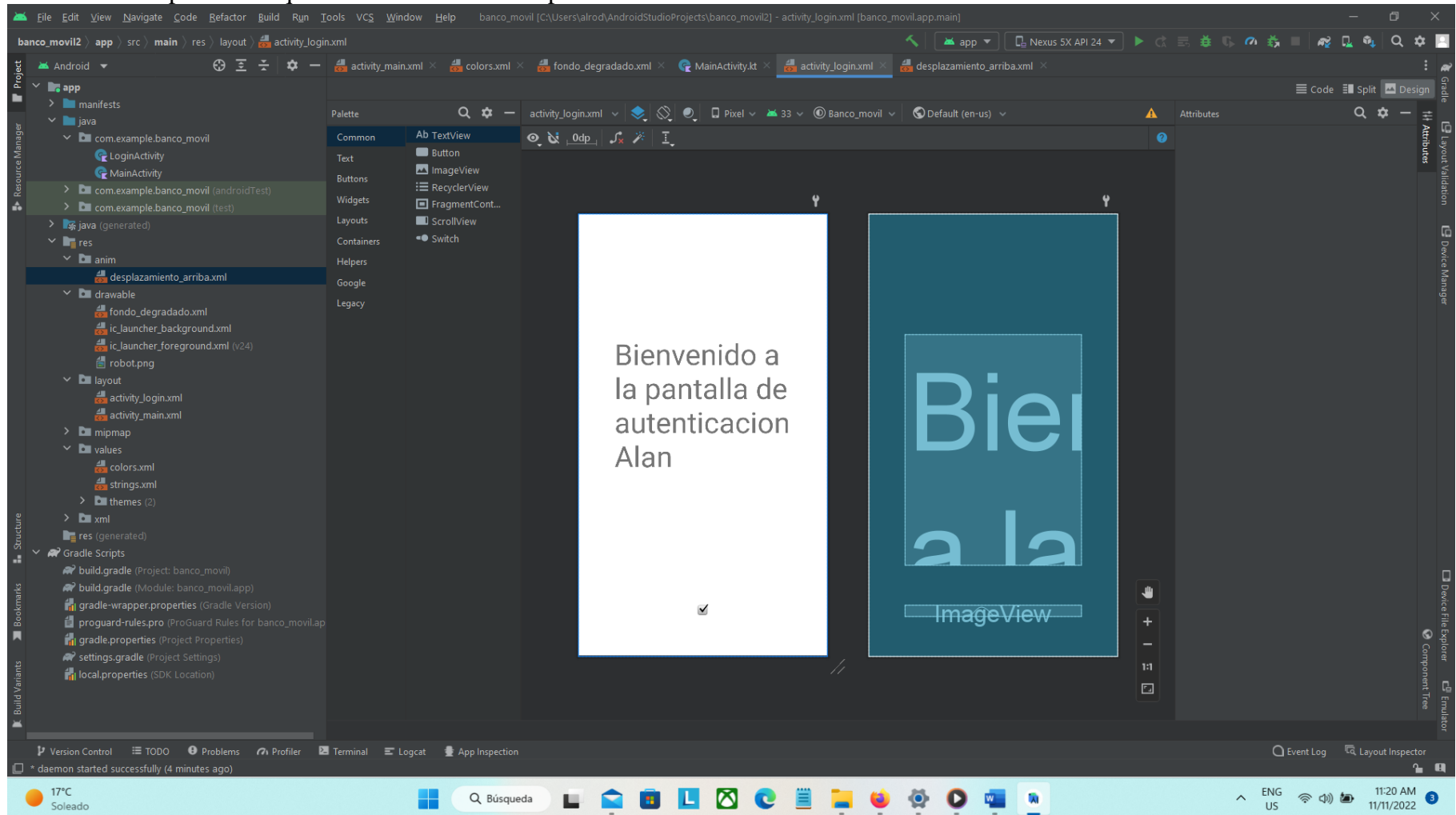
Una vez que se tiene el fondo y la imagen de la pantalla de inicio, se comienza con el código para la creación de la animación que llevara el programa, esta se hizo en el archivo MainActivity.kt, que será la encargada de realizar esa función, aquí se manda a llamar a la animación y que elementos harán esa acción y cuanto tiempo durara, en este caso la imagen de robot, al igual que también que pasara una vez que termine esta acción, que será ir a la pantalla de autenticación o Login pasados 2 segundos.



Ahora se continua con la creación de la animación, en este código se escribirá que hará la animación, que se desplazara de abajo hacia arriba y será invocada desde la MainActivity.



Y se realizara el activity_login que será la pantalla de autenticación ala cual se dirigirá después de la pantalla de inicio, agregue una bienvenida con mi nombre para saber que había cambiado a esa pantalla.



Codificación

MainActivity

```
package com.example.banco_movil

import android.content.Intent
import android.os.Bundle
import android.os.Handler
import android.view.WindowManager
import android.view.animation.AnimationUtils
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity

@Suppress("DEPRECATION")
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        window.setFlags(
            WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN
        )
        setContentView(R.layout.activity_main)
        val animation = AnimationUtils.loadAnimation(this, R.anim.desplazamiento_arriba)
        val robot = findViewById<ImageView>(R.id.robot)
        robot.animation = animation
        Handler().postDelayed({
            val intent = Intent(this@MainActivity, LoginActivity::class.java)
            startActivity(intent)
            finish()
        }, 2000)
    }
}
```

Fondo degradado

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <gradient
                android:startColor="@color/colorUno"
                android:centerColor="@color/colorDos"
                android:type="linear"
                android:centerX="0.5"
```



```

        android:centerY="0.5"
    />
</shape>

</item>
</selector>

```

Animación desplazamiento

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="0%"
        android:fromYDelta="100%"
        android:duration="2000"/>
    <alpha
        android:fromAlpha="0.1"
        android:toAlpha="1.0"
        android:duration="2000"/>
</set>

```

Colores

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>

    <color name="colorUno">#FFFFFF</color>
    <color name="colorDos">#CCCCCC</color>
</resources>

```

Activity Main

```

<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

```

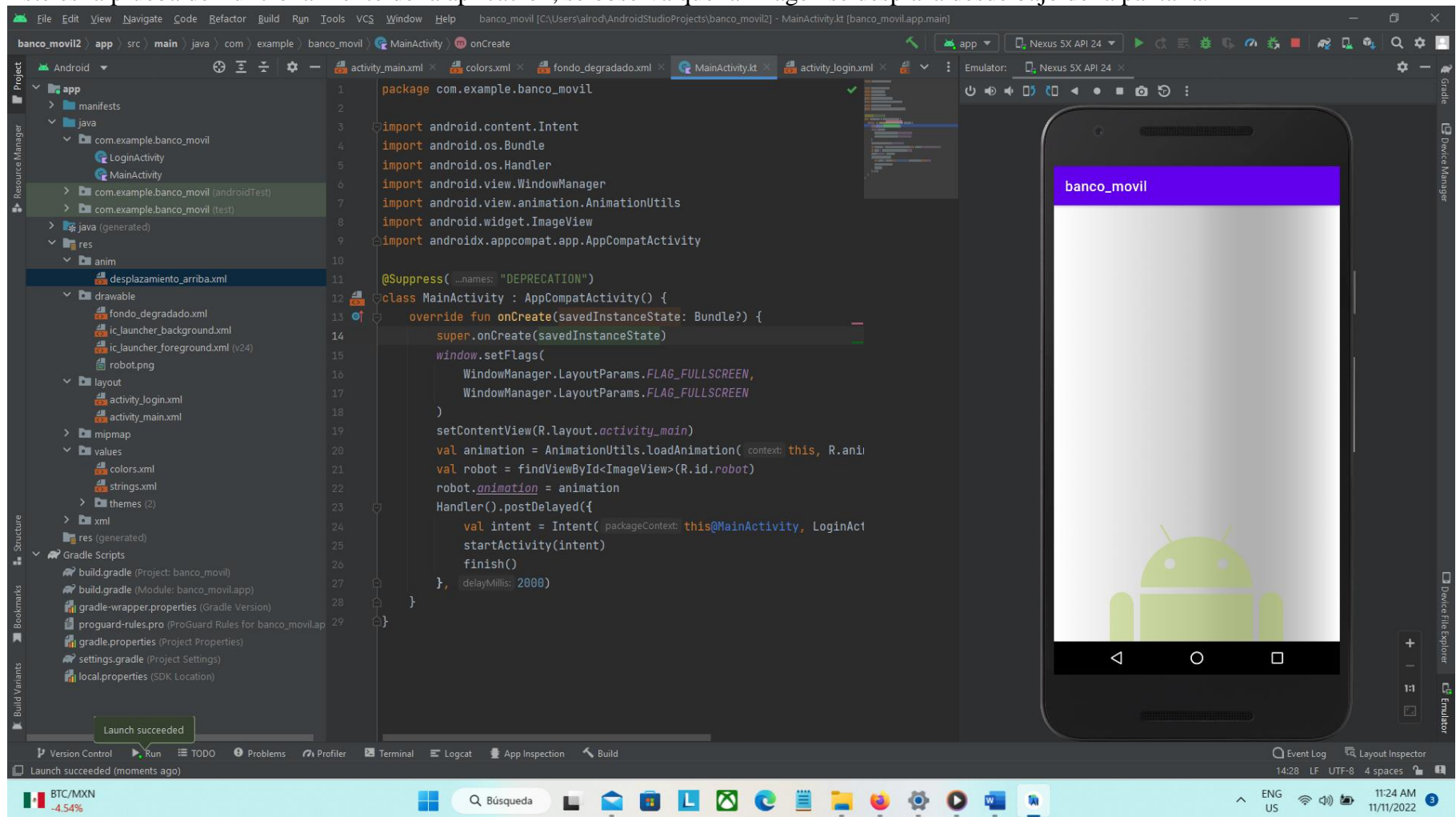
```
android:layout_height="match_parent"
tools:context=".MainActivity"
android:background="@drawable/fondo_degradado">

<ImageView
    android:id="@+id/robot"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="80dp"
    android:layout_marginTop="100dp"
    android:layout_marginEnd="80dp"
    android:layout_marginBottom="100dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/robot" />

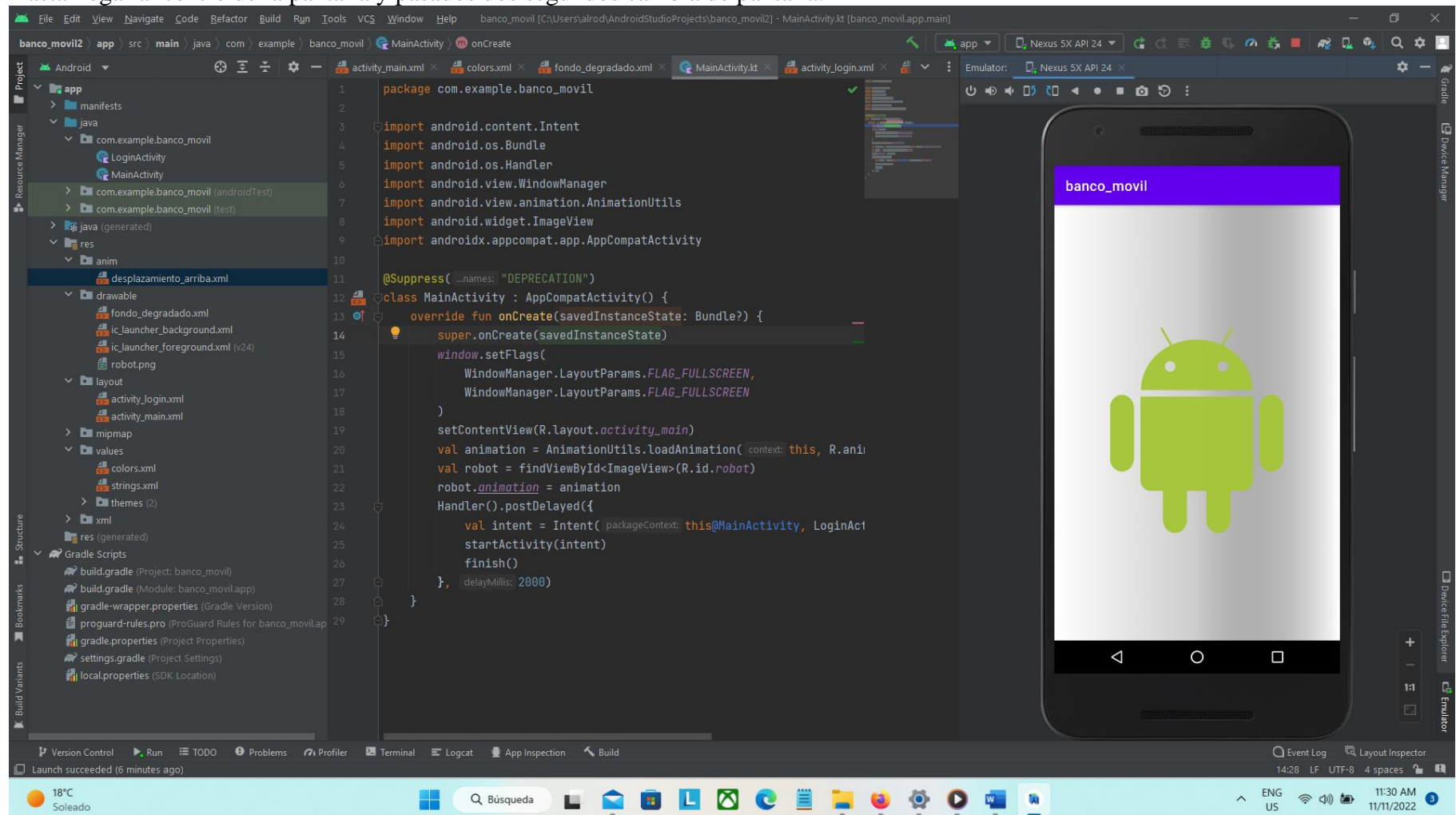
</androidx.constraintlayout.widget.ConstraintLayout>
```

Prueba de aplicación

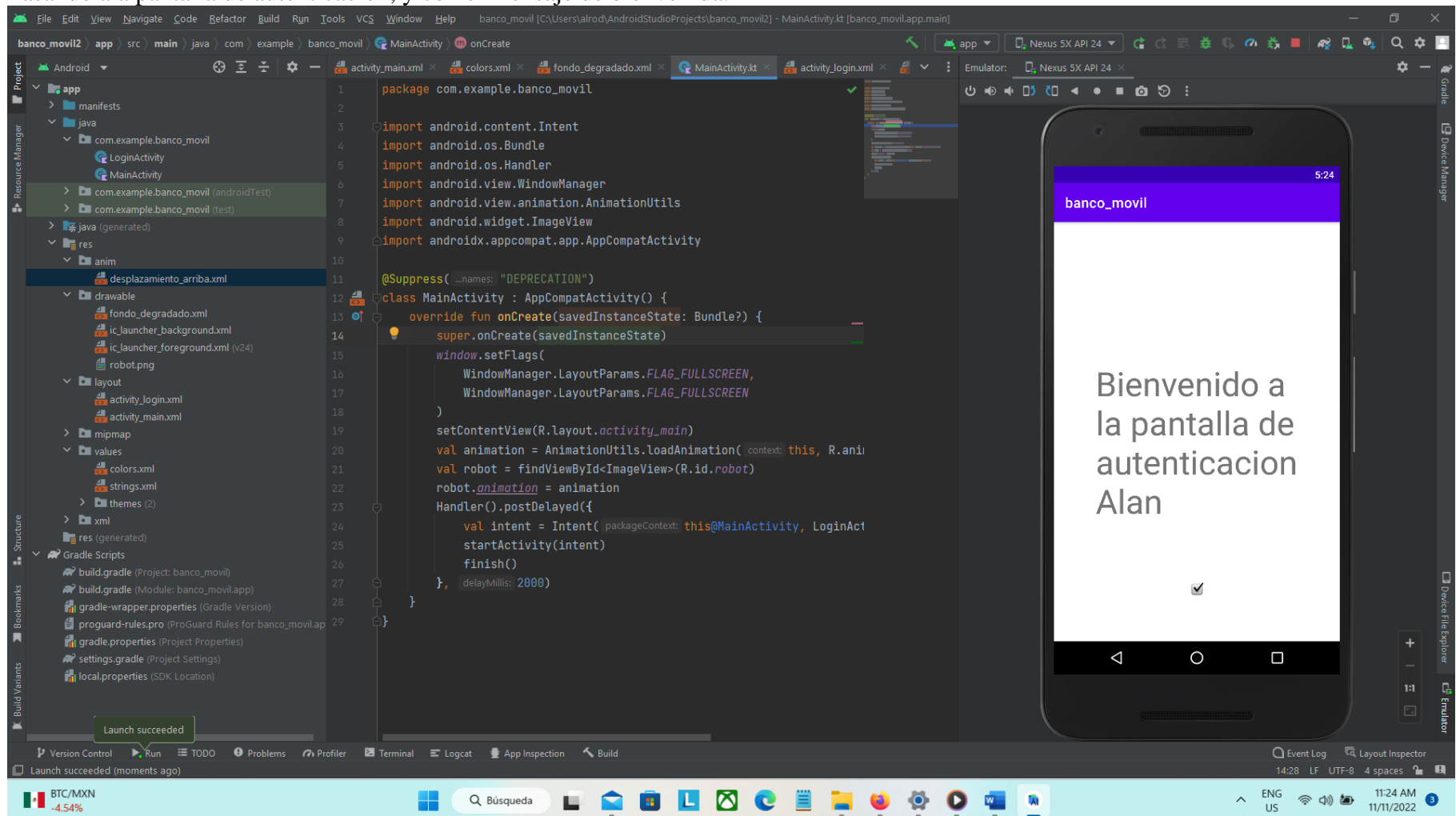
Este es la prueba del funcionamiento de la aplicación, se observa que la imagen se desplaza desde bajo de la pantalla.



Hasta llegar al centro de la pantalla y pasados dos segundos cambia de pantalla.



Pasando ala pantalla de autentificación, y con el mensaje de bienvenida.



Conclusión

El programar aplicaciones para Android es esencial en estos tiempos, ya que es una plataforma que actualmente domina el mercado de los dispositivos móviles, debido a su carácter libre y su desarrollo abierto, no controlado. Android Studio ofrece un entorno de desarrollo que facilita la implementación de aplicaciones de manera ágil y eficiente, aprovechando al máximo las características de cada dispositivo que use Android como sistema operativo (Tv, Móvil, Tablet, etc).

Tiene múltiples funciones, sobre todo en lo referente al desarrollo de la aplicación, para la realización de esta primera actividad, se ha tenido que buscar mucha información y ejemplos de cómo realizar diferentes funciones que requería la aplicación, con ello me di un poco la idea de cómo funciona Android Studio. También hay que decir que, pese a disponer de pequeños conocimientos de programación en java, era la primera vez que se desarrollaba una aplicación real, pero Android Studio facilitó la creación y programación de la pantalla de inicio (Splash Screen), de igual manera el emulador es una de las herramientas más útiles a la hora de probar una aplicación ya que se puede visualizar el resultado como si se estuviera ejecutando en un dispositivo real mientras se puede ver el código en un lado.

ENLACE GOOGLE DRIVE DEL PROYECTO

https://drive.google.com/file/d/1S8juH7-nnJbiZQo-xHlQCQ7NZ2KY2dhF/view?usp=share_link

Bibliografía

Developers, Introducción a Android Studio, sitio web
<https://developer.android.com/studio/intro?hl=es-419>

Marlon Mederos Corrales, Primer acercamiento al Android studio, Universidad tecnológica de la Habana, Monografía CUJAE 2021,
https://www.researchgate.net/publication/354474559_Primer_acercamiento_al_Android_Studio

Salvador Gómez Oliver, Manual Programación Android, Versión online 3.0, junio 2013,
<https://aluzardo.github.io/trabajo-fin-de-grado/Tutoriales/Manual%20Programacion%20Android.pdf>