

# **Actividad 1 - Pantalla de Inicio**

## **Desarrollo de aplicaciones móviles I**

### **Ingeniería en Desarrollo de Software**

**Tutor: Humberto Jesús Ortega Vázquez**

**Alumno: Jusi Ismael Linares Gutiérrez**

**Fecha: 05/11/2023**

# Índice

|                               |   |
|-------------------------------|---|
| Introducción .....            | 3 |
| Descripción .....             | 3 |
| Justificación .....           | 3 |
| Desarrollo: .....             | 4 |
| Interfaz .....                | 4 |
| Codificación.....             | 5 |
| Prueba de la aplicación ..... | 6 |
| Conclusión .....              | 7 |
| Referencias.....              | 7 |
| GitHub .....                  | 7 |

## Introducción

El primer paso es instalar Android Studio, una plataforma ampliamente utilizada en el desarrollo de aplicaciones Android. Luego, usaremos esta herramienta para construir la aplicación según los requisitos del cliente. Entre los detalles más interesantes, se nos pide crear una pantalla de inicio atractiva con un fondo de colores suaves y un logo animado que se desplace desde abajo hacia arriba (Splash Screen).

Además, hay un toque de automatización: después de dos segundos de cargar la pantalla de inicio, la aplicación deberá redirigir al usuario a la pantalla de autenticación.

En un mundo donde las aplicaciones móviles son parte de nuestra vida diaria, esta actividad ofrece una visión práctica de cómo se desarrollan y cómo pueden mejorar nuestro trabajo y experiencias cotidianas. Android Studio es nuestra herramienta principal,

## Descripción

En este escenario, una unidad de servicios bancarios necesita una aplicación móvil, y para lograrlo, están buscando la ayuda de un ingeniero en desarrollo de software. La aplicación que desean debe tener un aspecto atractivo y funcionar de manera fluida.

En lo que se refiere al aspecto visual, quieren que la pantalla de inicio tenga un fondo con una mezcla de colores y un logo animado que se mueva hacia arriba en la pantalla cuando arranque la aplicación. Esto dará a los usuarios una primera impresión agradable.

En cuanto a la funcionalidad, desean que después de dos segundos en la pantalla de inicio, la aplicación los redirija automáticamente a la pantalla de autenticación, agilizando así el proceso de acceso.

La actividad en sí implica instalar y configurar Android Studio, que es una herramienta para crear aplicaciones Android. Luego, el desarrollo real de la aplicación implica cumplir con estos requisitos de diseño y funcionalidad.

## Justificación

La elección de Android Studio como la solución para el desarrollo de la aplicación móvil en este caso es altamente justificada debido a varias razones fundamentales. Android Studio es el entorno de desarrollo oficial de Android, respaldado por Google, lo que garantiza compatibilidad y acceso a las últimas características y actualizaciones de la plataforma Android.

Además, Android Studio es una herramienta altamente eficiente y versátil que facilita la creación de aplicaciones móviles atractivas y funcionales. Permite un rápido desarrollo y pruebas en tiempo real, lo que es crucial para cumplir con las expectativas del cliente y asegurarse de que la aplicación se ajuste a sus requisitos.

Android Studio no solo es la elección lógica para el desarrollo de aplicaciones Android debido a su respaldo oficial y su capacidad para proporcionar las características necesarias, sino que también simplifica significativamente el proceso de desarrollo, lo que es esencial para entregar una aplicación de alta calidad de manera eficiente.

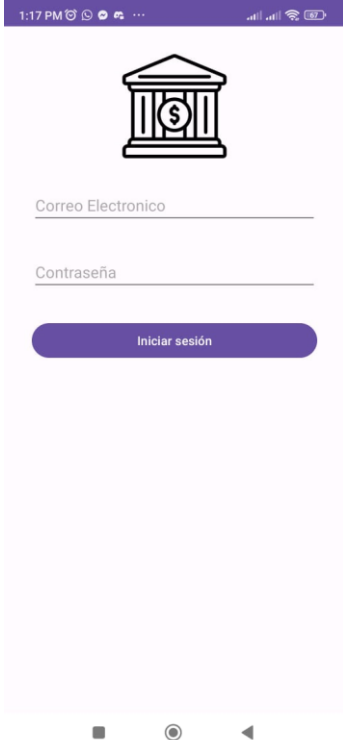
## Desarrollo:

### Interfaz

La primera interfaz que tenemos es la splash screen de nuestra app (Mas adelante la veremos en movimiento)

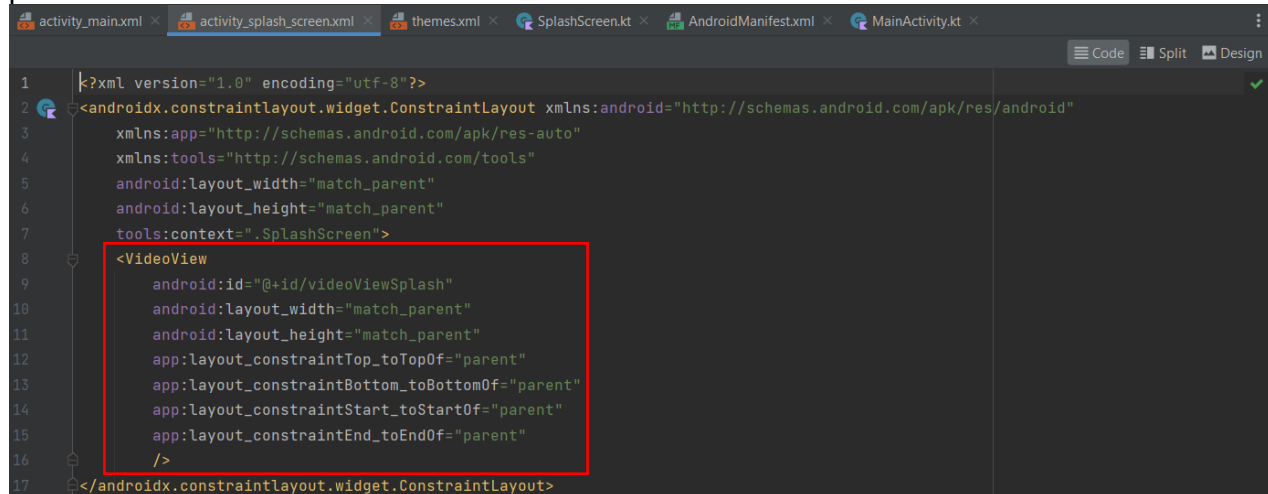


Esta es la pantalla del Login (Que haremos funcionar en actividades posteriores)



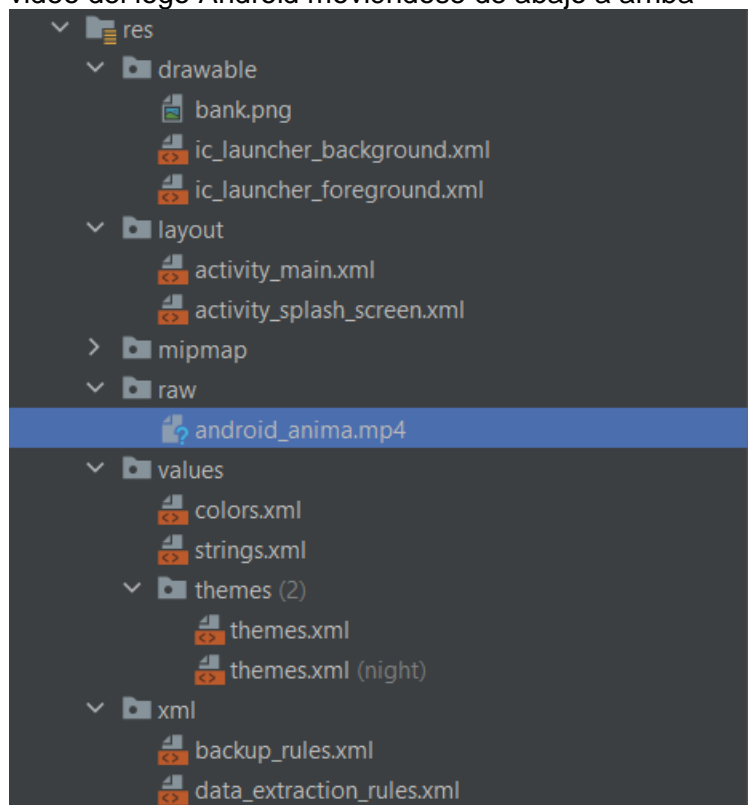
## Codificación

Con este código creamos una instancia videoview, posteriormente le damos un identificador "videoViewSplash". Con el match parent en los width y height nos aseguramos de que se ocupe toda la pantalla y por ultimo se ancla a los bordes superior, inferior, izquierdo y derecho de la pantalla.



```
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:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".SplashScreen">
8     <VideoView
9         android:id="@+id/videoViewSplash"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        app:layout_constraintTop_toTop0f="parent"
13        app:layout_constraintBottom_toBottom0f="parent"
14        app:layout_constraintStart_toStart0f="parent"
15        app:layout_constraintEnd_toEnd0f="parent"
16    />
17 </androidx.constraintlayout.widget.ConstraintLayout>
```

En res vamos a crear un directorio llamado raw y dentro de este nuevo directorio metemos nuestro video del logo Android moviéndose de abajo a arriba



Ya que tenemos el video, podemos escribir este código.

Este código crea una referencia al VideoView definido en el archivo de diseño XML y luego especificamos la ubicación del video que deseamos reproducir en ese VideoView.

```
package com.example.banco_movil

import ...

class SplashScreen : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash_screen)

        val videoView = findViewById<VideoView>(R.id.videoViewSplash)
        val videoPath = "android.resource://" + packageName + "/" + R.raw.android_anima
```

este código carga un video en nuestro VideoView, lo inicia y, cuando se completa la reproducción, inicia una nueva actividad (el cual será nuestro login) y finaliza la actividad actual que contiene el video.

```
val videoUri = Uri.parse(videoPath)
videoView.setVideoURI(videoUri)
videoView.start()

videoView.setOnCompletionListener { it: MediaPlayer!
    val intent = Intent( packageContext: this, MainActivity::class.java)
    startActivity(intent)
    finish()
}
```

En el Android manifest simplemente cortaremos el intent filter de la .MainActivity a la .SplashScreen para que esta sea la primera en mostrarse al abrir la app

```
<activity
    android:name=".SplashScreen"
    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=".MainActivity"
    android:exported="false">

</activity>
```

Prueba de la aplicación

Video: <https://drive.google.com/file/d/1qp87QKLu6qVs07RDvj-aqimvfS-oFEIv/view?usp=sharing>

Proyecto:

<https://drive.google.com/file/d/1SXXOERzzwt0l5HsnEV4xozF6hn4b9xno/view?usp=sharing>

## Conclusión

La actividad de crear una aplicación móvil para la unidad de negocio enfocada en servicios bancarios es un paso crucial en el mundo laboral de la ingeniería en desarrollo de software. Para llevar a cabo este proyecto, primero debemos instalar y configurar Android Studio, una herramienta fundamental en el desarrollo de aplicaciones Android.

Luego, al diseñar la pantalla de inicio con un fondo degradado y un logotipo animado que se desplaza de abajo hacia arriba, estamos cumpliendo con los requisitos de interfaz que el cliente solicitó. Además, al configurar la redirección automática a la pantalla de autenticación después de dos segundos, estamos abordando los requisitos funcionales de la aplicación.

En mi campo laboral, esta actividad demuestra la importancia de la adaptabilidad y la capacidad para satisfacer las necesidades específicas del cliente.

## Referencias

*video splash screen in android studio tutorial | kotlin android tutorial for beginners | Android.*

(2023, July 27).

## GitHub

[https://github.com/JusiLinGu/Practicas\\_UMI](https://github.com/JusiLinGu/Practicas_UMI)