

### **Actividad 3 – Pantalla de Autenticación**

**Desarrollo de Aplicaciones Móviles I**

**Ingeniería en Desarrollo de Software**

**Tutor: Humberto Jesús Ortega Vázquez**

**Alumno: Kathya Viridiana Chávez Domínguez**

**Fecha: 08/04/2024**

## Índice

Introducción .....	3
Descripción .....	4
Justificación.....	5
Desarrollo .....	6
Interfaz .....	6
Codificación .....	11
Prueba de la aplicación.....	13
Conclusión.....	14
Referencias .....	15

## Introducción

Una pantalla de autenticación es una interfaz utilizada para verificar la identidad de un usuario antes de permitirle acceder a un sistema. Este aspecto es importante para cualquier organización que recopile, almacene y comparta información derivada de su actividad y del contacto que mantienen con sus usuarios.

A través de este proyecto continuaremos trabajando con el entorno de desarrollo Android Studio, para mejorar la interfaz que desarrollamos en la actividad anterior. Agregaremos validaciones que le permitirán al sistema indicar al usuario por medio de mensajes de alerta, cuando ingrese a la aplicación, cuando la contraseña sea correcta o cuando se introduzca un usuario y contraseña válidos. Al finalizar, seremos capaces de comprender la importancia de desarrollar una aplicación Android funcional y práctica, así como conocer a fondo las diversas funciones y herramientas que tiene para ofrecer Android Studio. Este conocimiento lo podremos aplicar en futuros proyectos y en nuestra vida laboral.

## Descripción

En esta ocasión, la unidad de negocio enfocada a servicios bancarios solicita una aplicación móvil. Por lo que, solicitan que un ingeniero en desarrollo de software atienda a su necesidad, teniendo en cuenta que la aplicación contenga los siguientes requerimientos:

### **Requerimientos de interfaz:**

- Pantalla de autenticación con dos cajas de texto:
- La primera caja servirá para ingresar el usuario.
- La segunda caja para ingresar la contraseña.

### **Requerimientos funcionales:**

- El sistema validará que el usuario haya ingresado correctamente su correo.  
Si esto no sucede, aparecerá una pantalla modal para indicarle al usuario.
- El sistema validará que la contraseña que el usuario haya ingresado sea la correcta. Si esto no sucede, aparecerá una pantalla modal para indicarle al usuario.
- Si se ingresa un usuario y contraseña válidos, el sistema mostrará una alerta con el texto “Entró”.

Para esta actividades se solicita continuar trabajando en las actividades 1 y 2 agregando los requerimientos funcionales anteriormente mencionados.

## Justificación

Tal como se mencionó en la introducción, la autenticación móvil es un factor importante para cualquier aplicación, ya que es esencial poder identificar a los usuarios que solicitan el acceso a sistemas que almacenan datos confidenciales. Además, se vuelve una necesidad para mejorar la experiencia del público en general durante el proceso de registro o inicio de sesión. Al proporcionar un nombre de usuario y una contraseña, el sistema valida los datos para confirmar la identidad del usuario o denegar el acceso.

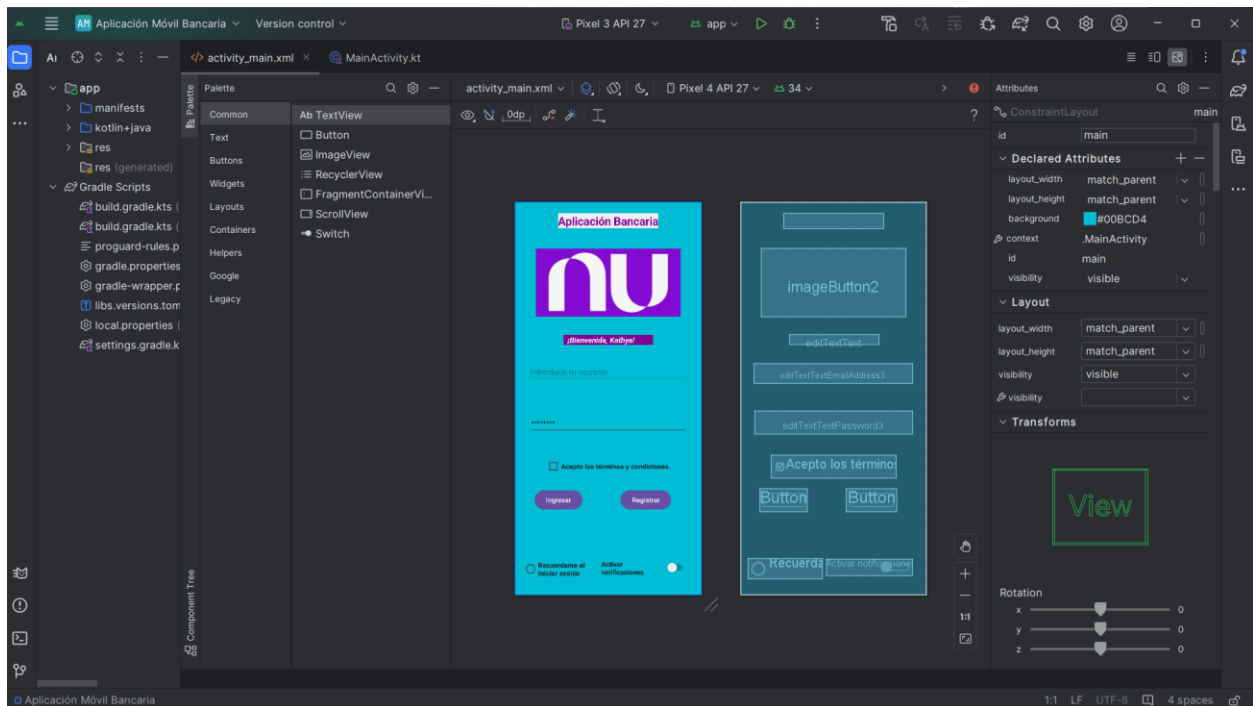
Una buena pantalla de autenticación ofrece seguridad y practicidad a los usuarios al permitirles iniciar sesión de manera más rápida y evitar la necesidad de recordar contraseñas. En resumen, el control de acceso y autenticación son acciones claves para proteger cualquier aplicación móvil. Su objetivo principal es proporcionar a los negocios procesos que faciliten la gestión de usuarios y datos de identificación, al mismo tiempo que controlan el acceso a los recursos.

## Desarrollo

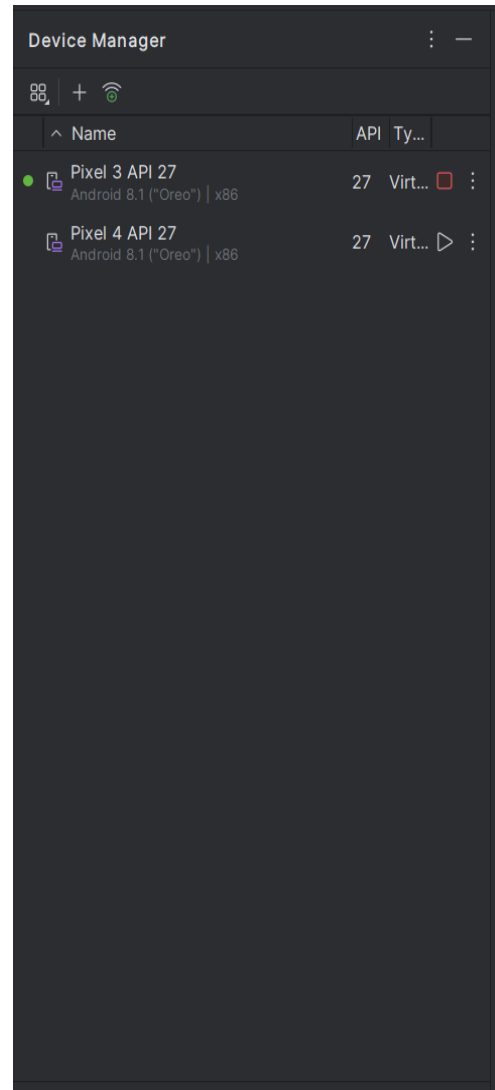
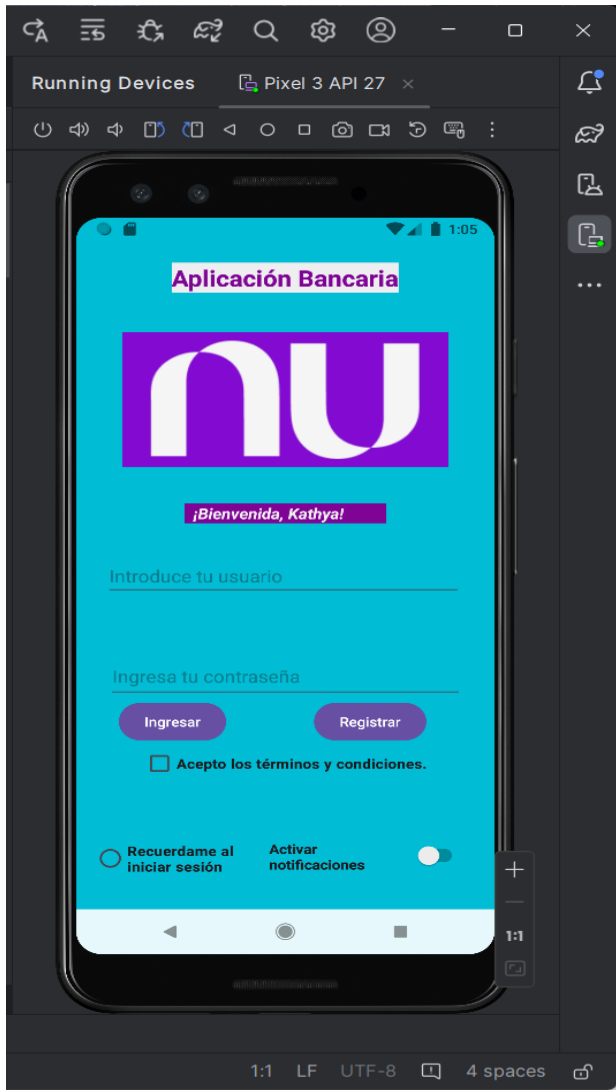
### Interfaz

Comenzamos nuestro proyecto utilizando el diseño creado en la actividad anterior, con el fin de incorporar los nuevos requisitos y funcionalidades.

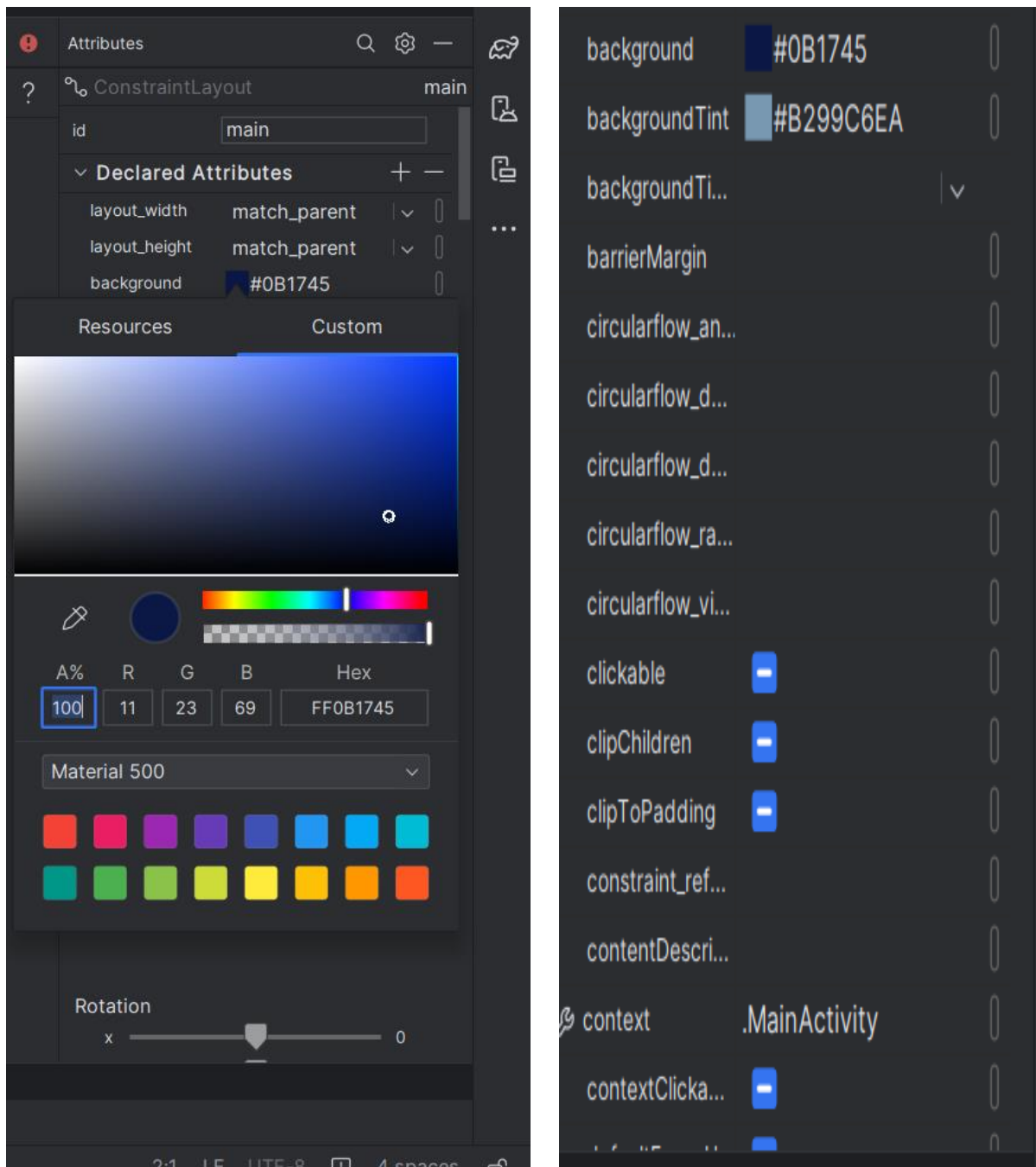
Además, realizaremos mejoras en el diseño aplicando los conocimientos adquiridos durante las últimas dos actividades. Esta práctica tiene como objetivo reforzar nuestras habilidades y adquirir mayor destreza en el entorno de desarrollo.



Al igual que antes, es crucial probar el emulador para prevenir errores durante la ejecución de la aplicación. Como podemos ver, el emulador funciona correctamente. Para esta práctica, estamos utilizando el dispositivo virtual Pixel 3 con API 27 y la versión de Android seleccionada es "Oreo". Hemos elegido este dispositivo y versión específicos debido a que son ideales para asegurar la compatibilidad y el rendimiento óptimo de la aplicación.

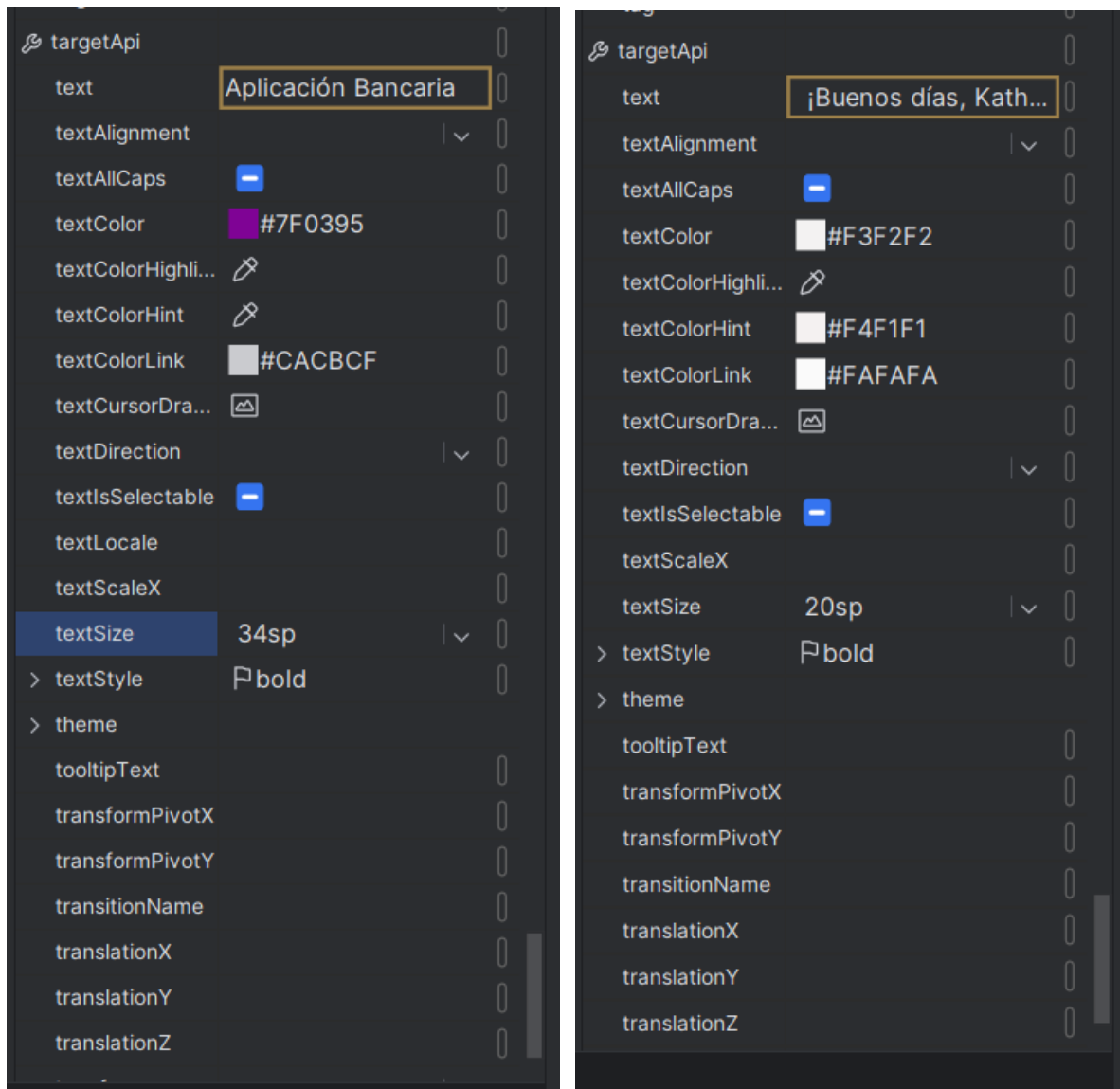


Recordemos que la barra de "atributos" nos permite modificar nuestra aplicación, incluyendo el tipo de letra, color de fondo y otros elementos que contribuyen a desarrollar una interfaz más atractiva para el usuario. Además, como aprendimos en la primera actividad, podemos combinar dos colores para mejorar la apariencia visual de la interfaz.

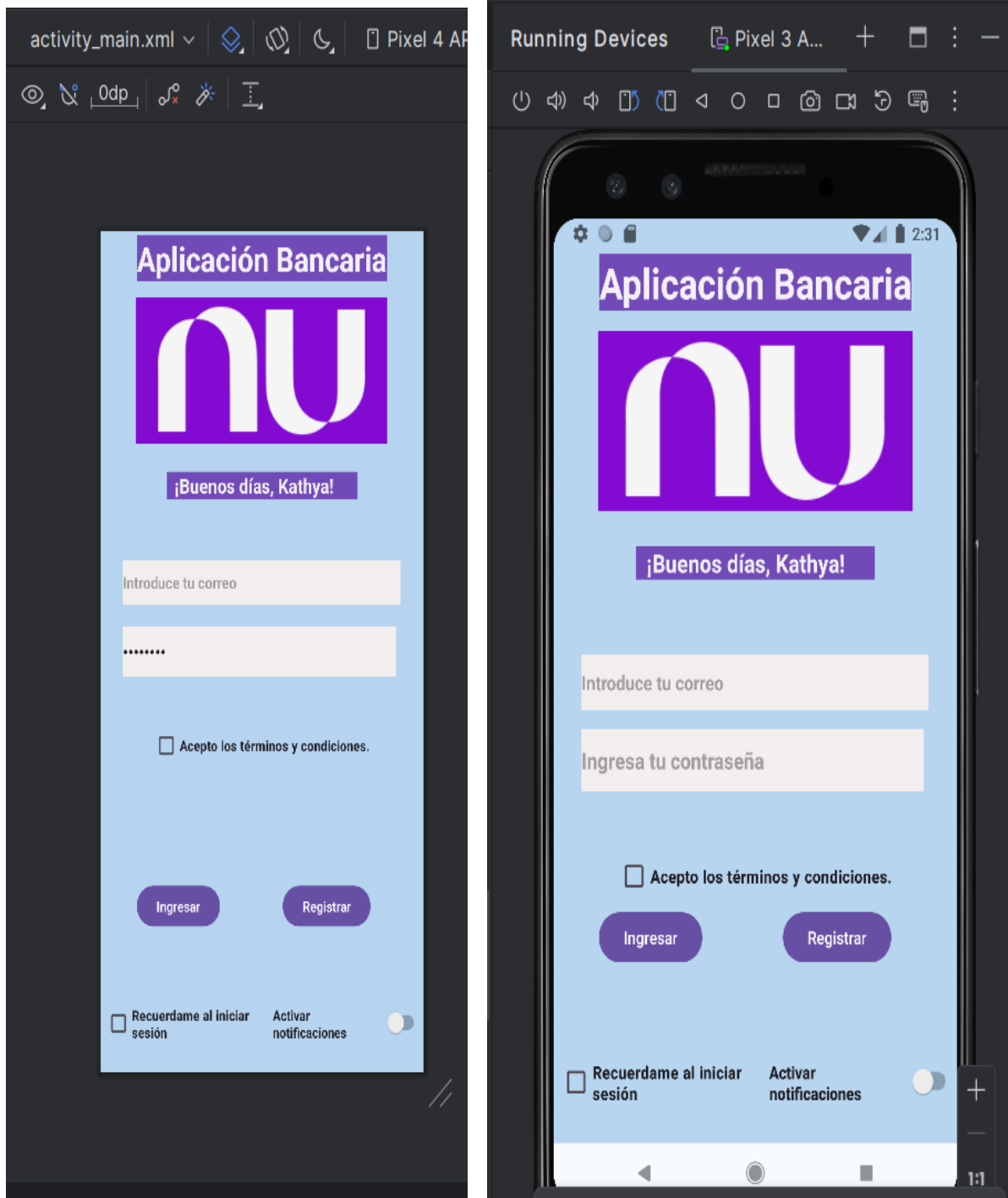




Como podemos observar, con Android Studio tenemos la capacidad de modificar todos los elementos disponibles con el objetivo de ofrecer una mejor experiencia al usuario.



Gracias a los elementos que hemos modificado e incorporado, podemos obtener una interfaz aún más atractiva. Es fundamental probar el emulador para asegurarnos de que los "constraints" estén correctamente colocados y que la pantalla muestre la visualización deseada.

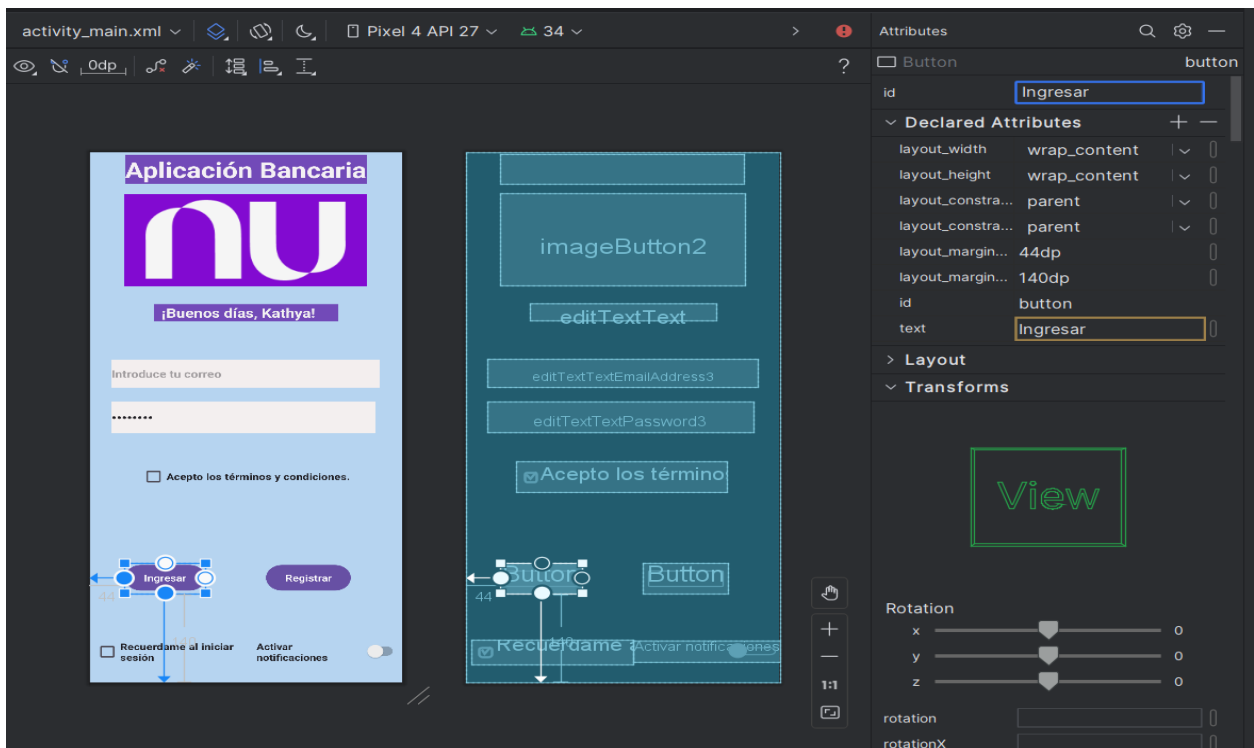


## Codificación

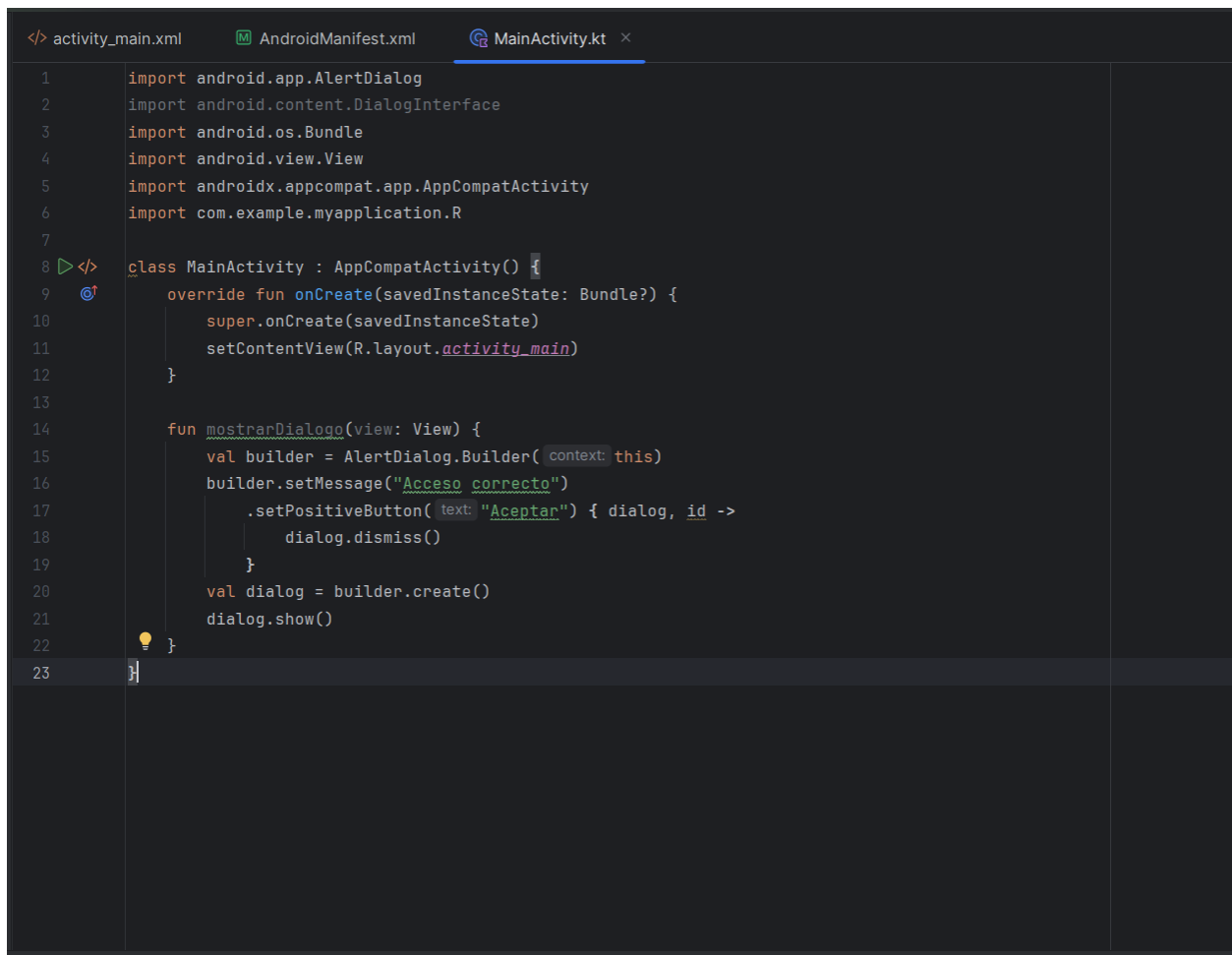
Ahora vamos a crear diálogos de alerta para que cuando seleccionemos un botón u opción, se muestre el mensaje correspondiente. Esto lo vamos a ingresar en la sección de "MainActivity.kt".

```
</> activity_main.xml      MainActivity.kt  x
1  package com.example.aplicacionmvilbancaria
2
3  > import ...
4
5
6
7
8
9  class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         enableEdgeToEdge()
13         setContentView(R.layout.activity_main)
14         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
15             val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
16             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
17             insets
18         }
19     }
20 }
```

Sin embargo, para que el código funcione correctamente, es necesario asignar nombres a los botones y revisar minuciosamente la lógica de este código.



A continuación se muestra el código actualizado que permite que la aplicación muestre mensajes de alerta. Anteriormente, se producía un error, pero después de importar la clase "R", el código pudo ejecutarse sin problemas.



```
<?xml version="1.0" encoding="utf-8"?>
<activity_main.xml>
</activity_main.xml>
AndroidManifest.xml
MainActivity.kt x
1  import android.app.AlertDialog
2  import android.content.DialogInterface
3  import android.os.Bundle
4  import android.view.View
5  import androidx.appcompat.app.AppCompatActivity
6  import com.example.myapplication.R
7
8  class MainActivity : AppCompatActivity() {
9      override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12     }
13
14     fun mostrarDialogo(view: View) {
15         val builder = AlertDialog.Builder(context = this)
16         builder.setMessage("Acceso correcto")
17         builder.setPositiveButton(text = "Aceptar") { dialog, id ->
18             dialog.dismiss()
19         }
20         val dialog = builder.create()
21         dialog.show()
22     }
23 }
```

### Prueba de la aplicación

Gracias a la codificación y los requisitos previamente establecidos, podemos ver que nuestra aplicación ahora funciona correctamente. Esto permite al usuario ingresar datos y habilitar los campos necesarios de forma adecuada.



## Conclusión

A través de esta actividad hemos tenido la oportunidad de trabajar nuevamente con el entorno de desarrollo integrado, Android Studio, para finalizar la pantalla de autenticación desarrollada en la última actividad. Gracias a esta experiencia pudimos revisar a detalle las diferentes herramientas que ofrece Android Studio, además de entender la importancia de incluir elementos que faciliten una autenticación exitosa y una experiencia intuitiva y práctica para el usuario.

Como hemos visto, la pantalla de autenticación desempeña un papel fundamental en la seguridad y la experiencia del usuario en cualquier aplicación o sistema. En conclusión, la implementación efectiva de una pantalla de autenticación no solo garantiza la seguridad de una aplicación, sino que también contribuye de manera significativa a una experiencia de usuario positiva y sobre todo confiable. No cabe duda de que las habilidades y el conocimiento adquiridos durante este proyecto serán de gran utilidad en futuros proyectos y sobre todo en nuestra vida laboral.

**Link GitHub:** <https://github.com/KathyaCh/Aplicaci-nBancariaAndroidStudio.git>

## Referencias

- I. Admin. (2023, octubre 5). *5 características clave de la autenticación móvil para el compromiso del cliente actual - Route Mobile*. Route Mobile.  
<https://routemobile.com/es/blog/5-key-features-of-mobile-authentication-for-todays-consumers-and-businesses-2/>
- II. Grupo Cibernos. (s.f.). *¿Cómo funcionan los datos de autenticación y por qué son tan importantes?* Recuperado de <https://www.grupocibernos.com/blog/como-funcionan-los-datos-de-autenticacion-y-por-que-son-tan-importantes>
- III. Mato, M. (2021, 15 diciembre). *Cómo crear un AlertDialog custom en Android con Kotlin y el patrón Delegation*. *Medium*.  
<https://medium.com/@manuelmato/c%C3%B3mo-crear-un-alertdialog-custom-en-android-con-kotlin-y-el-patr%C3%B3n-delegation-353196eb2f58>