

Actividad 3 - Pantalla de Autenticación

Desarrollo de Aplicaciones Móviles I

Ingeniería en Desarrollo de Software

Tutor: Humberto Jesús Ortega Vazquez

Alumno: Jose Eduardo Canche Alfaro

Fecha: 08 de abril del 2024

Índice

2

1.	Introducción.....	3
2.	Descripción.....	3
3.	Justificación.....	4
4.	Desarrollo:	5
4.1.	Interfaz	5
4.2.	Codificación	6
4.3.	Prueba de la aplicación.....	8
4.4.	Conclusión.....	11

1. Introducción

3

En esta etapa, presentaremos la fase final de nuestra aplicación bancaria, la cual hemos ido desarrollando a lo largo de actividades anteriores. En este proceso, hemos empleado el software Android Studio para diseñar cada etapa de la aplicación, aprovechando sus herramientas para codificar y crear mensajes interactivos al interactuar con los botones. Android Studio se destaca por su facilidad de uso y funcionalidades avanzadas. Su entorno integrado de desarrollo ofrece características como la finalización automática de código, facilitando la escritura de código eficiente. Además, integra todas las herramientas necesarias, desde el editor de código hasta el emulador de dispositivos, en una interfaz unificada. Una de las ventajas principales de Android Studio es su compatibilidad con Kotlin y Java, los dos principales lenguajes de programación para aplicaciones Android. Esto brinda a los desarrolladores la flexibilidad de elegir el lenguaje que mejor se adapte a sus necesidades. Además, Google proporciona actualizaciones regulares para Android Studio, lo que asegura que los desarrolladores puedan acceder a las últimas características, mejoras de rendimiento y correcciones de errores. Esta constante evolución del software garantiza un entorno de desarrollo actualizado y optimizado para la creación de aplicaciones Android de alta calidad.

2. Descripción

En esta unidad de negocio centrada en los servicios bancarios, se nos encomienda como desarrolladores de software la creación de una aplicación móvil que satisfaga las necesidades del usuario. Nuestra tarea consiste en diseñar y desarrollar esta aplicación de acuerdo con los requisitos establecidos. La aplicación debe presentar un saludo personalizado al iniciar sesión, adaptado según la hora del día en que se accede. Además, debe incluir una imagen distintiva y

solicitar la dirección de correo electrónico y la contraseña del usuario. También se nos solicita añadir dos botones: uno para registrarse y otro para iniciar sesión. Al pulsar estos botones, la aplicación debe mostrar mensajes correspondientes a las acciones realizadas, confirmando el registro o el inicio de sesión del usuario. Nuestro objetivo es cumplir con todos estos requisitos para ofrecer una experiencia óptima al usuario final.

4

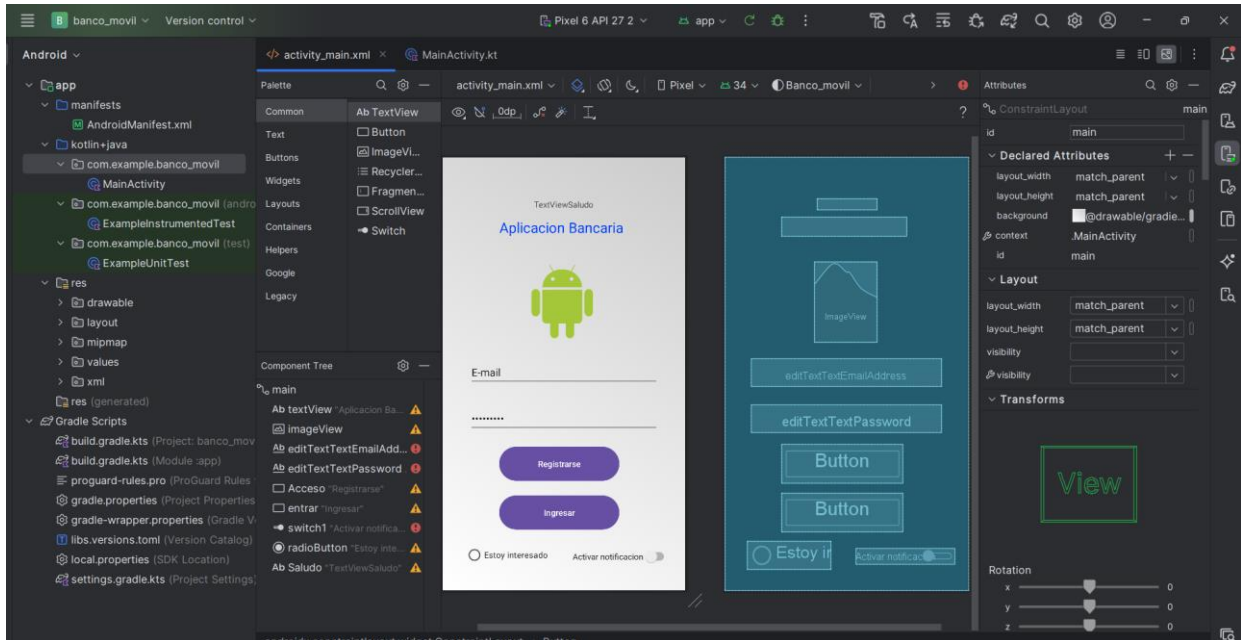
3. Justificación

En esta actividad, concluimos el desarrollo de nuestra aplicación bancaria, la cual hemos estado construyendo desde la primera actividad. Hemos aplicado los conocimientos adquiridos en las tutorías para completar todas las etapas de este proyecto, aprovechando las herramientas proporcionadas por Android Studio. En esta fase, nos hemos centrado en la codificación para dotar a nuestra aplicación de funcionalidad. Al hacer clic en los botones que hemos añadido, como el de registro, mostramos un mensaje de bienvenida confirmando el registro exitoso del usuario. De manera similar, al presionar el botón de entrar, la aplicación indica que el acceso ha sido correcto. Además, al iniciar sesión, la aplicación muestra un saludo personalizado en función de la hora del día ("Buenos días", "Buenas tardes" o "Buenas noches"). Con esta codificación, hemos logrado integrar de manera efectiva todas las actividades previas en nuestra aplicación, asegurando su funcionalidad y usabilidad para el usuario final.

4. Desarrollo:

5

4.1. Interfaz



En esta captura tenemos la interfaz de nuestro diseño de la aplicación bancaria que se nos esta pidiendo tenemos el diseño ahora lo que nos toca hacer es darle funcionamiento a los botones.

```

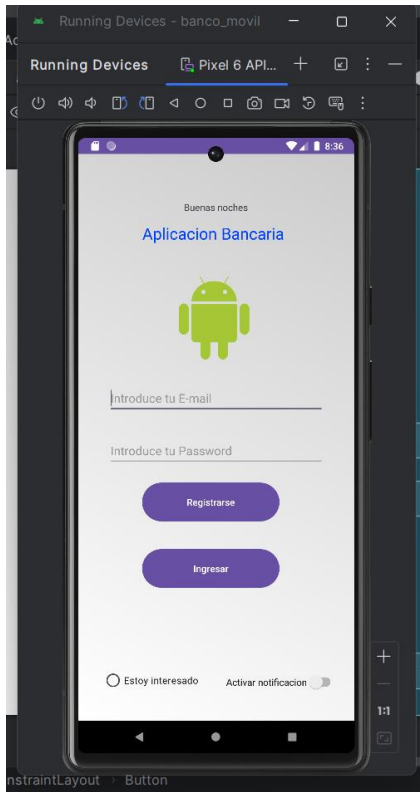
1 package com.example.banco_movil
2
3 import android.os.Bundle
4 import android.widget.Button
5 import android.widget.TextView
6 import androidx.appcompat.app.AlertDialog
7 import androidx.appcompat.app.AppCompatActivity
8 import java.util.*
9
10 class MainActivity : AppCompatActivity() {
11
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15
16         // Obtener la hora actual
17         val cal = Calendar.getInstance()
18         val hora = cal.get(Calendar.HOUR_OF_DAY)
19
20         // Determinar si es de día o de tarde
21         val textViewSaludo = findViewById<TextView>(R.id.Saludo)
22         textViewSaludo.text = when (hora) {
23             in 0 ..< 11 -> "Buenos días"
24             in 12 ..< 17 -> "Buenas tardes"
25             else -> "Buenas noches"
26         }
27
28         // Asignar el listener al botón Acceso
29         val btnAcceso = findViewById<Button>(R.id.Acceso)
30         btnAcceso.setOnClickListener { it: View!
31             // Mostrar el diálogo de alerta utilizando la función de extensión
32             showAlertDialog(title="Registro Correcto", message="¡Bienvenido! Te has registrado correctamente.")
33         }
34
35         // Obtener la instancia del botón "entrar"
36         val buttonEntrar = findViewById<Button>(R.id.entrar)
37         // Configurar el OnClickListener para el botón "entrar"
38         buttonEntrar.setOnClickListener { it: View!
39             // Crear una alerta de diálogo
40             val alertDialogBuilder = AlertDialog.Builder(context=this)
41             alertDialogBuilder.apply { this: AlertDialog.Builder
42                 setTitle("Acceso Correcto")
43                 setMessage("¡Has accedido correctamente!")
44                 setPositiveButton(text="Aceptar") { dialog, _ ->
45                     dialog.dismiss() // Cerrar el diálogo cuando se hace clic en Aceptar
46                 }
47             }
48             // Mostrar la alerta de diálogo
49             val alertDialog = alertDialogBuilder.create()
50             alertDialog.show()
51         }
52     }
53
54     // Función de extensión para mostrar el diálogo de alerta
55     private fun AppCompatActivity.showAlertDialog(title: String, message: String) {
56         val builder = AlertDialog.Builder(context=this)
57         builder.setTitle(title)
58         builder.setMessage(message)
59         builder.setPositiveButton(text="Aceptar", listener=null)
60         // Mostrar el diálogo
61         val alertDialog = builder.create()
62         alertDialog.show()
63     }
64 }
65
66
67
68

```

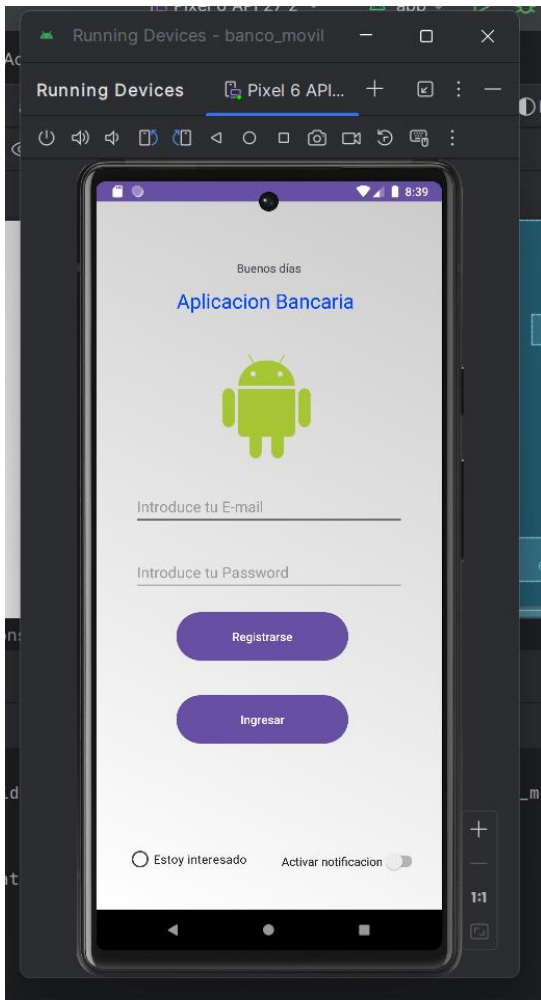
Este código representa nuestra aplicación. Comenzamos importando las bibliotecas necesarias y definimos la clase principal, llamada 'MainActivity'. Dentro del método 'onCreate()', configuramos la interfaz de usuario con el diseño definido en 'activity_main.xml'. Obtenemos la hora actual del dispositivo y, según esta hora, mostramos un saludo correspondiente ("Buenos días", "Buenas tardes" o "Buenas noches") en un TextView identificado como 'Saludo'. Asociamos un OnClickListener al botón 'Acceso'. Cuando este botón es presionado, mostramos un cuadro de diálogo de alerta con el mensaje "¡Bienvenido! Has ingresado correctamente.". También configuramos un OnClickListener para el botón 'entrar'. Al hacer clic en este botón, se muestra otro cuadro de diálogo de alerta con el mensaje "¡Has accedido correctamente!". Además, definimos una función de extensión llamada 'showAlertDialog()' que simplifica la creación y visualización de cuadros de diálogo de alerta con el título y mensaje proporcionados. Esta función facilita la presentación de alertas dentro de la actividad 'MainActivity'.

4.3.Prueba de la aplicación

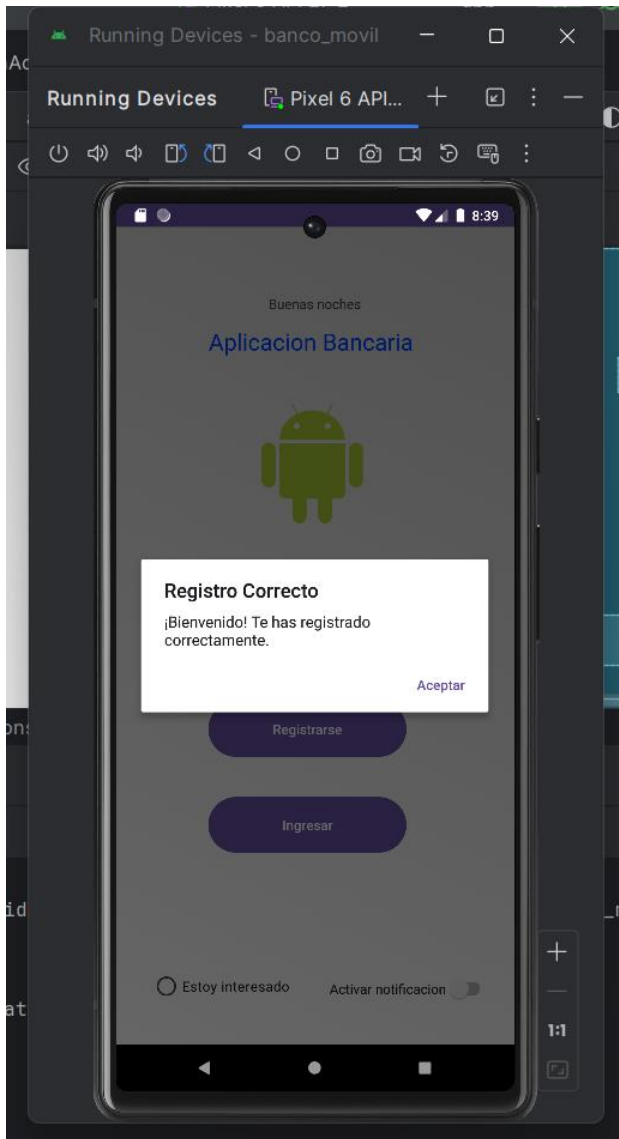
8



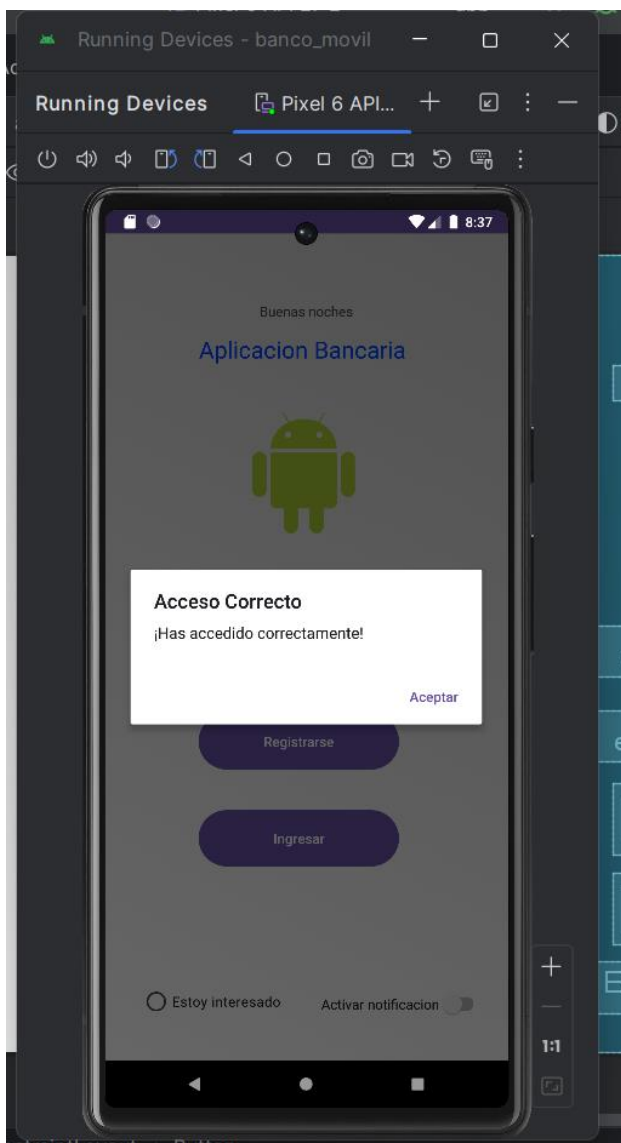
En esta imagen nos muestra el mensaje de buenas noches con respecto a la hora que tiene el dispositivo virtual.



En esta imagen el dispositivo virtual tiene otra hora y nos muestra el mensaje de buenos días.



En esta imagen nos muestra un mensaje al apretar el botón de registro que nos dice que se hizo el registro correctamente.



En esta imagen al darle click al botón de ingresar nos muestra un mensaje que se a accedido correctamente.

4.4.Conclusión

A lo largo de este proyecto de desarrollo, hemos adquirido un profundo entendimiento sobre la creación de aplicaciones para dispositivos Android. Este conocimiento no solo nos permite poner en práctica lo aprendido, sino también identificar las herramientas necesarias para llevar a cabo

estas tareas de manera efectiva. El conjunto de herramientas proporcionadas por Android Studio facilita enormemente el desarrollo de proyectos como este. Esta experiencia es invaluable en el mundo laboral, donde la demanda de aplicaciones para dispositivos inteligentes es cada vez mayor. Android, como uno de los principales sistemas operativos móviles, busca ofrecer aplicaciones de alta calidad, lo que subraya la importancia de que los desarrolladores estén bien preparados y cuenten con las habilidades necesarias. El conocimiento adquirido a lo largo de este proyecto puede abrirnos muchas puertas en el campo laboral, brindándonos oportunidades para trabajar como desarrolladores de aplicaciones. Es crucial comprender la relevancia y el impacto que estas habilidades tienen en un mercado cada vez más competitivo y orientado hacia la tecnología.