

Actividad | 3 | Codificación de la aplicación. Desarrollo de Aplicaciones Biométricas.

Ingeniería en Desarrollo de
Software.



TUTOR: Marco Alonso Rodríguez

ALUMNO: Carlos Ariel Nicolini

FECHA: 30/09/2025

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	5
Etapas 1	6
• Diseño de interfaces	6
Etapas 2	9
• Codificación	9
• Ejecución en el teléfono	14
• Enlace del proyecto	16
Conclusión.....	17
Referencias.....	18

Introducción

La codificación en programación se trata del modo en que se escribe un mensaje (comando) para que una computadora lo entienda y realice tareas. La codificación se usa para crear software y aplicaciones que usamos en la vida cotidiana, como:

- Aplicaciones de celulares: la app como WhatsApp, Instagram y los juegos, están creados mediante códigos escritos por programadores.
- Sitios Web: Cuando navegas por internet, visitas páginas que también están construidas con códigos. Así, los diseñadores web y los programadores se encargan de crear su apariencia y funcionalidad.
- Videojuegos: Si sos fanático del gaming o te entretenés con la Play Station, X-Box o la compu, la codificación es esencial. Cada movimiento de tu personaje, enemigo y efecto especial son controlados por código.
- Redes Sociales: Facebook, Twitter y otras RRSS son posibles gracias al mismo procedimiento. Los programadores crean las plataformas y las aplicaciones que te permiten conectarte con amigos y compartir fotos y mensajes.

La codificación para Android es el proceso de creación de aplicaciones para dispositivos que ejecutan un sistema operativo Android. Android Studio es el entorno de desarrollo integrado oficial (IDE) para Android nativo.

Descripción

Contextualización:

Realizar una aplicación que funcione para iniciar sesión con las huellas dactilares previamente registradas en el teléfono. La interfaz de la app deberá contar con dos pantallas:

1. Pantalla de inicio de sesión.
2. Pantalla de bienvenida.

La aplicación deberá realizar las siguientes funciones:

- Cuando se ingrese una huella dactilar que no esté registrada, deberá mostrar un icono que representa un escaneo fallido, acompañado de un mensaje de error que diga lo siguiente: “Escaneo fallido, huella dactilar no registrada”.
- Cuando se ingresa la huella correcta, se deberá mostrar un icono que represente un escaneo exitoso, acompañado del siguiente mensaje: “¡Escaneo de huella dactilar exitoso! Iniciando sesión ...”
- Una vez confirmada la huella dactilar correcta, deberá pasar a la segunda pantalla de bienvenida. Esta deberá contar con un botón que permita regresar al inicio de sesión.

Actividad:

Realizar la codificación de la aplicación que ha realizado desde la actividad 2. Generar los métodos necesarios para la ejecución, así como los mensajes que se mostraran dependiendo de la huella digital que se escanee. Además, programar la aplicación para que permita acceder a la pantalla de bienvenida cuando se escanee la huella digital correcta.

Justificación

En esta última actividad, continuaremos con todo el trabajo que se realizó a lo largo de esta materia de desarrollo de aplicaciones biométricas, contando como base el prototipo realizado en la actividad 1 en Figma y con el desarrollo en Android Studio de las pantallas que fue aprobado por el consejo de supertierra, se agregará la función de la autenticación de la huella digital. Una vez que la aplicación autentique con la huella digital se dirigirá a la pantalla de bienvenida de la aplicación.

Para este trabajo se tomó mucha información proporcionada en la clase y en los documentos compartidos por el grupo de slack.

Este trabajo gracias a la explicación del profesor fue muy rápido y de mucho valor. Si me costó un poco poder realizar la conexión con el celular, el mío no quiso y tuve que utilizar el teléfono de mi hijo, perdón si en el video sale que la pantalla se nota rara, es que está quebrada, pero espero que el video sea claro.

Espero que el trabajo cumpla con lo solicitado, siempre agradecido con el profesor por su guía y conocimiento. Espero que nos podamos encontrar en otra materia, muchísimas gracias.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-Biom-tricas>

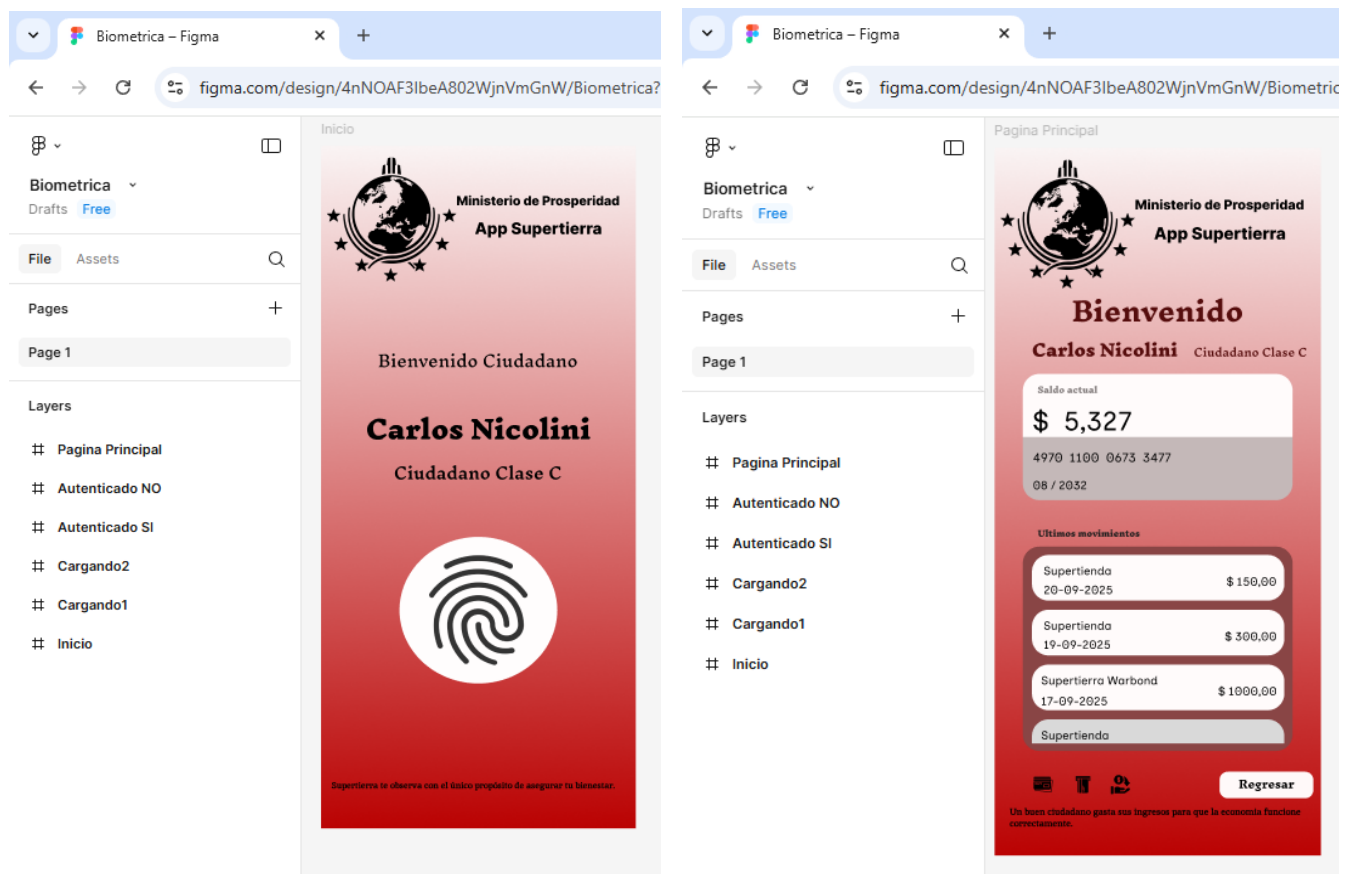
Desarrollo

Etapas 1

Diseño de interfaces

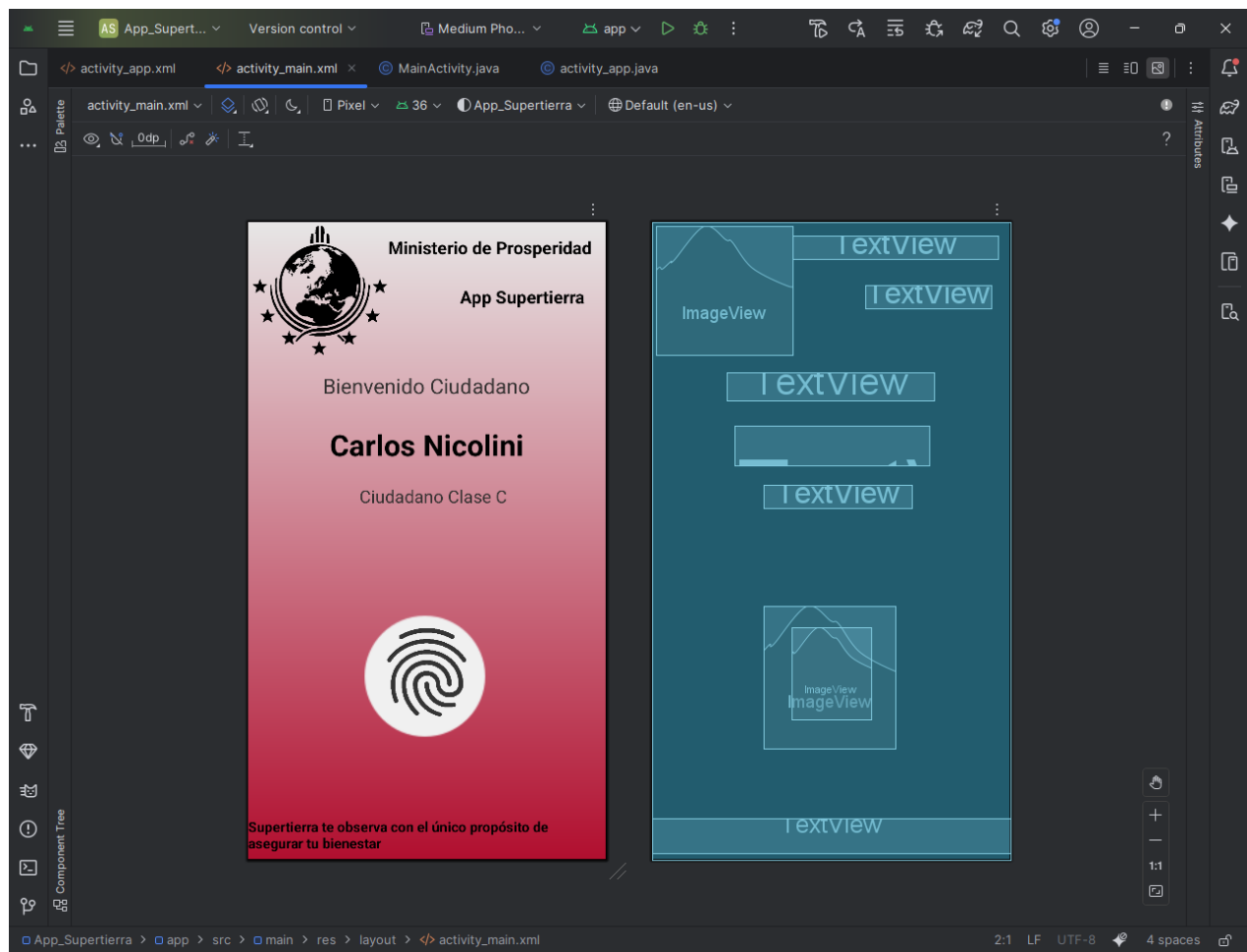
En actividad final de la materia desarrollo de aplicaciones Biométricas continuaremos con lo realizado en las dos actividades anteriores, donde en la actividad número uno se realizó un prototipo de la aplicación en Figma, y en la actividad dos, ese mismo prototipo se realizó en Android Studio. A continuación, se presentarán imágenes de los diseños de las interfaces.

Diseño de prototipo en Figma (actividad 1).

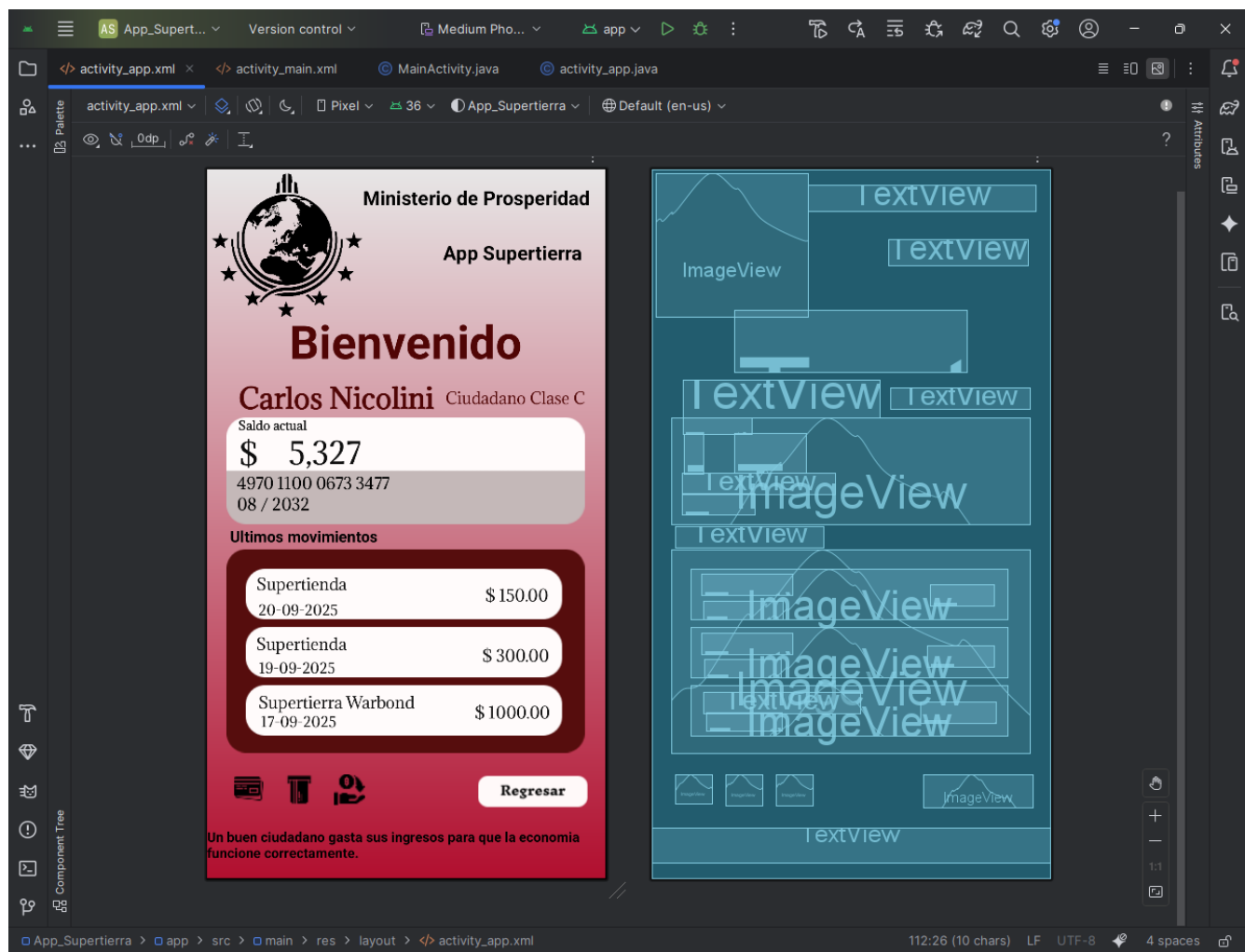


Diseño de interfaces en Android Studio (actividad 2).

Pantalla de inicio de la aplicación (activity_main.xml)



Pagina de bienvenida (activity_app.xml)

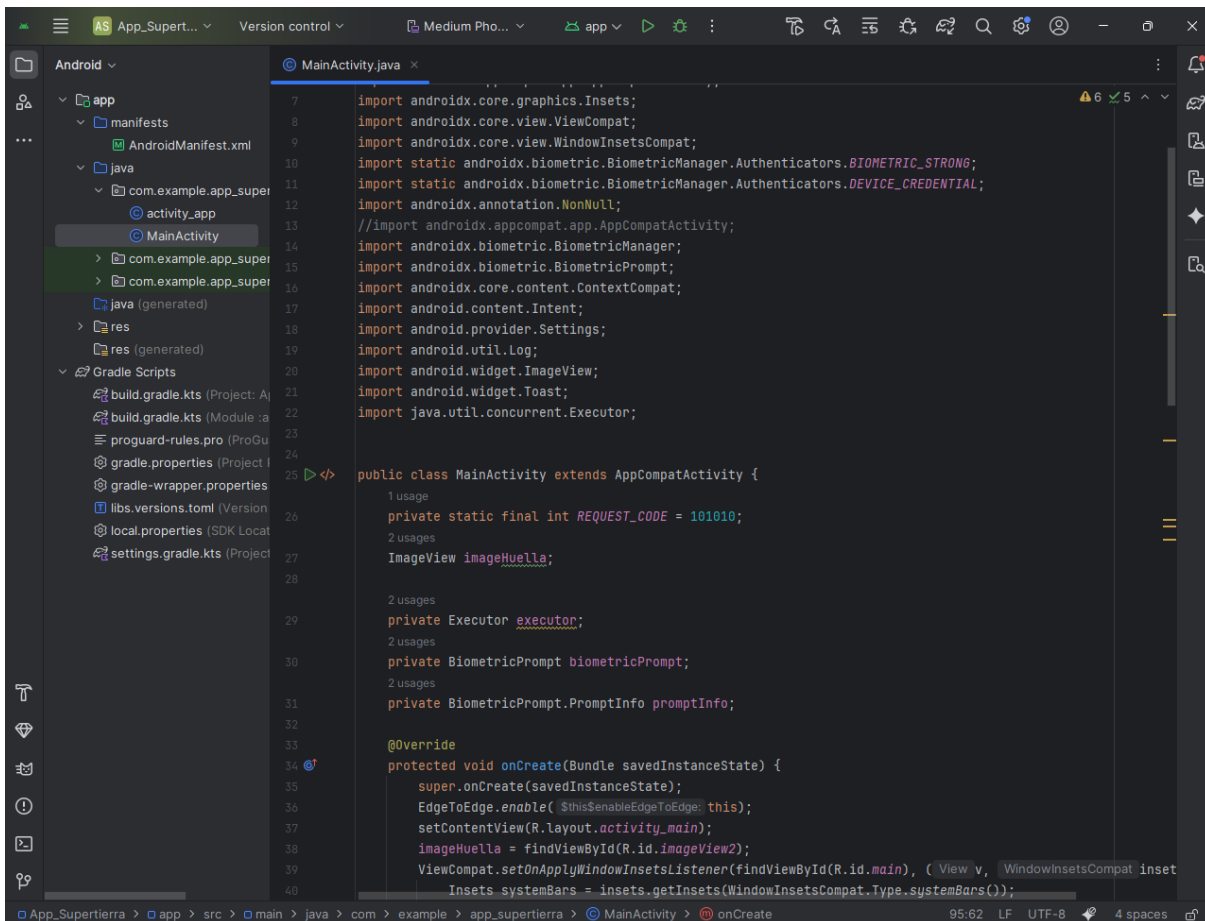


Etapa 2

Diseño de interfaces

En esta parte del ejercicio mostraremos el código realizado con una breve explicación.

MainActivity.java



```
7 import androidx.core.graphics.Insets;
8 import androidx.core.view.ViewCompat;
9 import androidx.core.view.WindowInsetsCompat;
10 import static androidx.biometric.BiometricManager.Authenticators.BIOMETRIC_STRONG;
11 import static androidx.biometric.BiometricManager.Authenticators.DEVICE_CREDENTIAL;
12 import androidx.annotation.NonNull;
13 //import androidx.appcompat.app.AppCompatActivity;
14 import androidx.biometric.BiometricManager;
15 import androidx.biometric.BiometricPrompt;
16 import androidx.core.content.ContextCompat;
17 import android.content.Intent;
18 import android.provider.Settings;
19 import android.util.Log;
20 import android.widget.ImageView;
21 import android.widget.Toast;
22 import java.util.concurrent.Executor;
23
24
25 public class MainActivity extends AppCompatActivity {
26     1 usage
27     private static final int REQUEST_CODE = 101010;
28     2 usages
29     ImageView imageHuella;
30
31     2 usages
32     private Executor executor;
33     2 usages
34     private BiometricPrompt biometricPrompt;
35     2 usages
36     private BiometricPrompt.PromptInfo promptInfo;
37
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
42         setContentView(R.layout.activity_main);
43         imageHuella = findViewById(R.id.imageView2);
44         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), ( v, WindowInsetsCompat insets)
45             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
```

En este MainActivity implementamos la autenticación biométrica (huella digital, reconocimiento facial).

Se importan las librerías necesarias de Android, AppCompatActivity, BiometricManager, BiometricPrompt y utilidades para manejar vistas e insets.

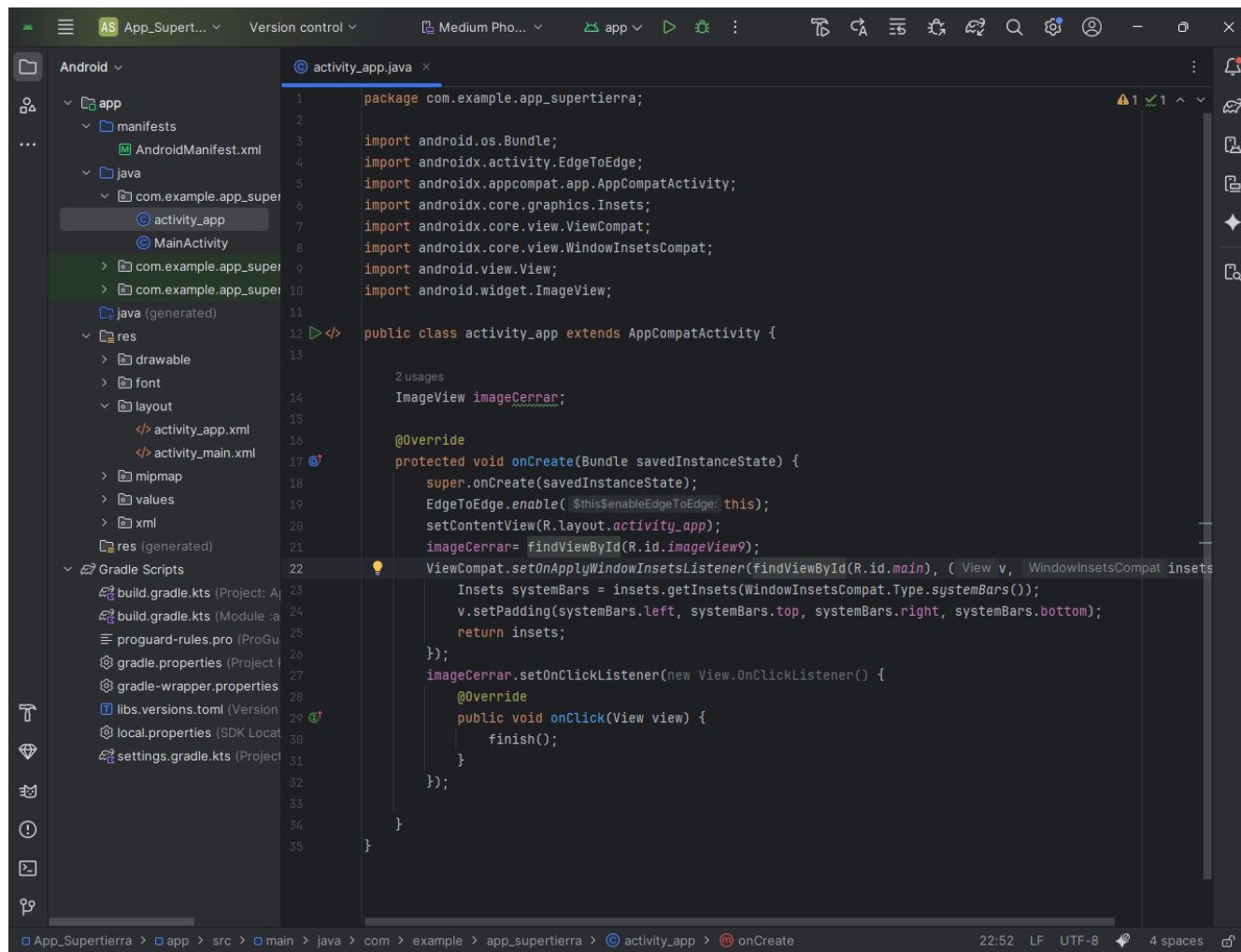
Se agregan las variables de Imagehuella (que servirá como botón para iniciar la autenticación). Executor que define el hilo donde se ejecutarán las respuestas de autenticación, además de biometricPrompt y promptInfo que manejarán el diálogo de autenticación biométrica.

Se crea el onCreate, se configura la interfaz (setContentView) y adapta la vista al diseño del activity_main. Con BiometricManager verifica si el dispositivo soporta autenticación biométrica o con credenciales.

Evento de clic en imagenHuella, cuando el usuario toca la imagen, se lanza el prompt de autenticación biométrica.

El código implementa un login biometrico en Android, cuando el usuario toca la imagen de huella, se muestra un cuadro de autenticación biométrica. Si se valida correctamente, accede a la siguiente actividad (activity_app), de lo contrario muestra un mensaje de error o de denegación.

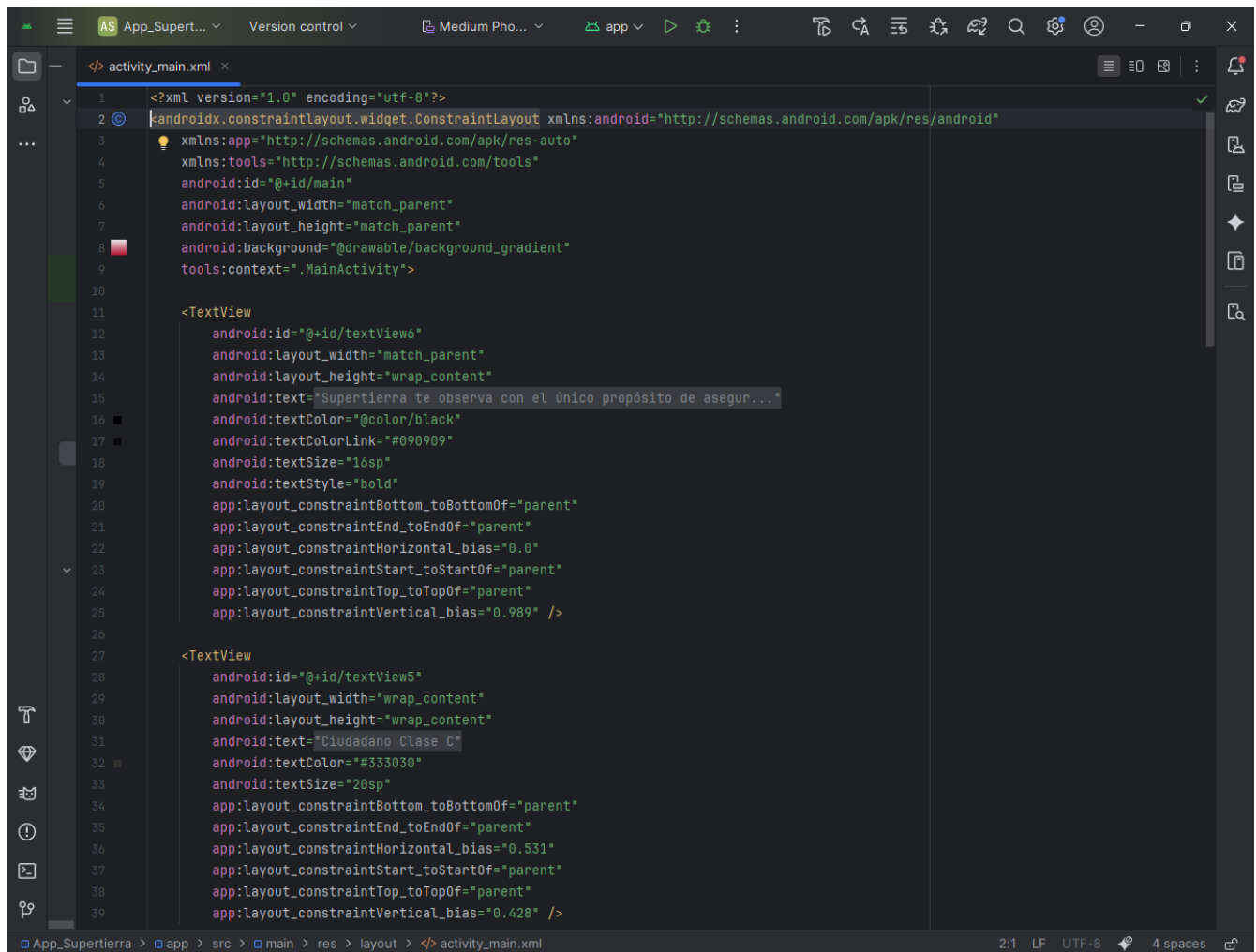
Activity_app.java



En el activity_app es la pantalla secundaria (pantalla de bienvenida) a la que accedemos después del login biométrico.

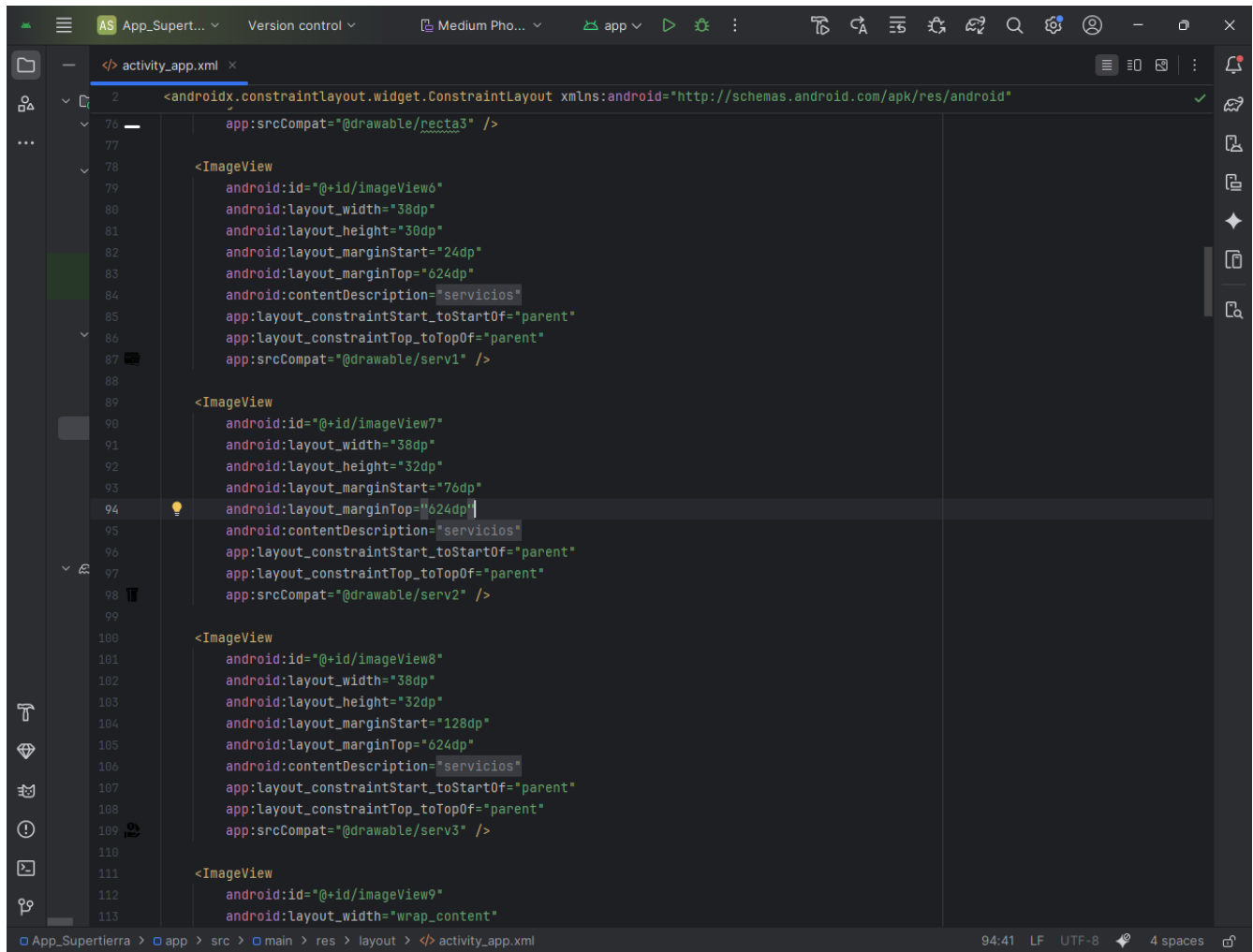
Este tiene un botón con forma de imagen, que, al presionarlo, la pantalla se cierra y se regresa a la vista anterior.

Activity_main.xml



En este archivo se define la interfaz grafica de la actividad principal de la aplicación realizara en Android. Muestra el logo de la app, textos de identificación (nombre, bienvenida, ministerio de la verdad, etc), y en la parte inferior incluye una huella digital de un círculo, que será usada por el usuario para autenticarse.

Activity_app.xml)



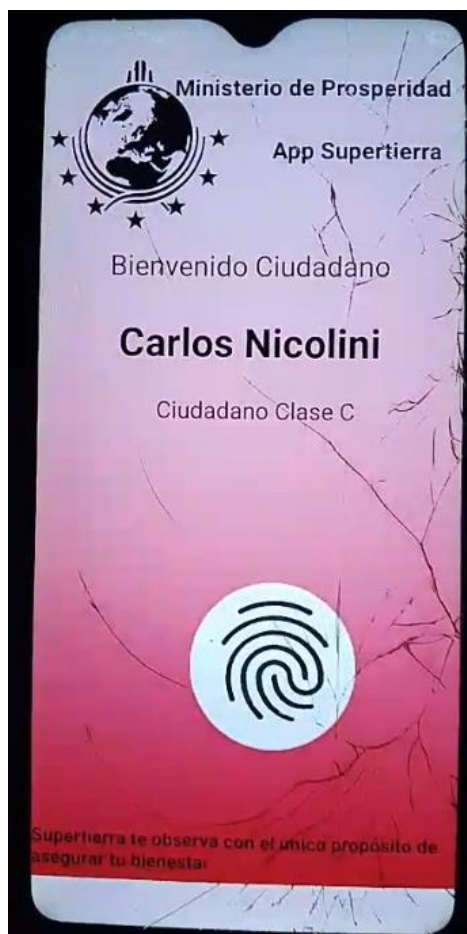
El siguiente archivo es la activity de la segunda pantalla. Este está echo con ConstraintLayout, lo que permite posicionar los elementos (textos e imágenes) en la pantalla de forma flexible y relativa entre sí.

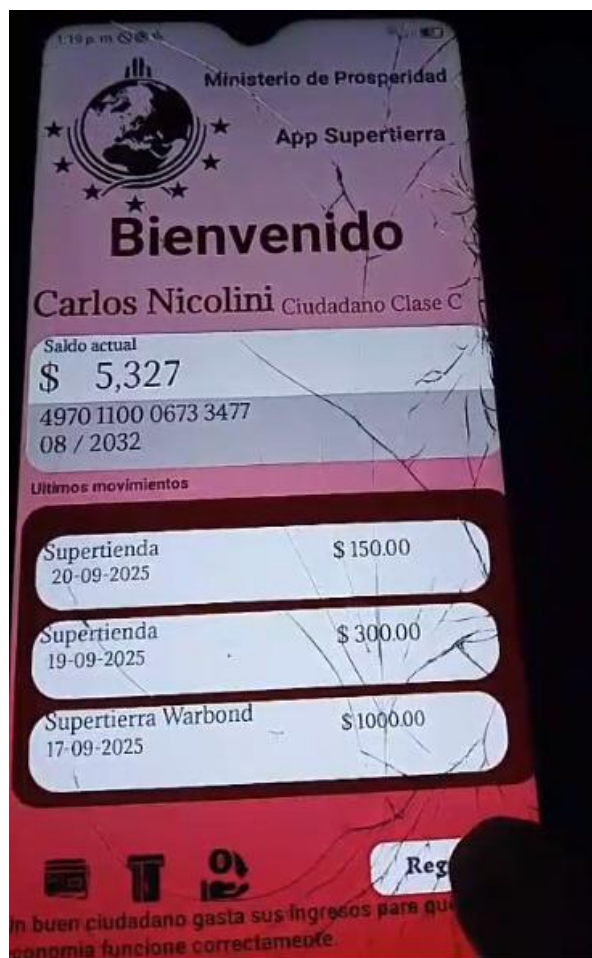
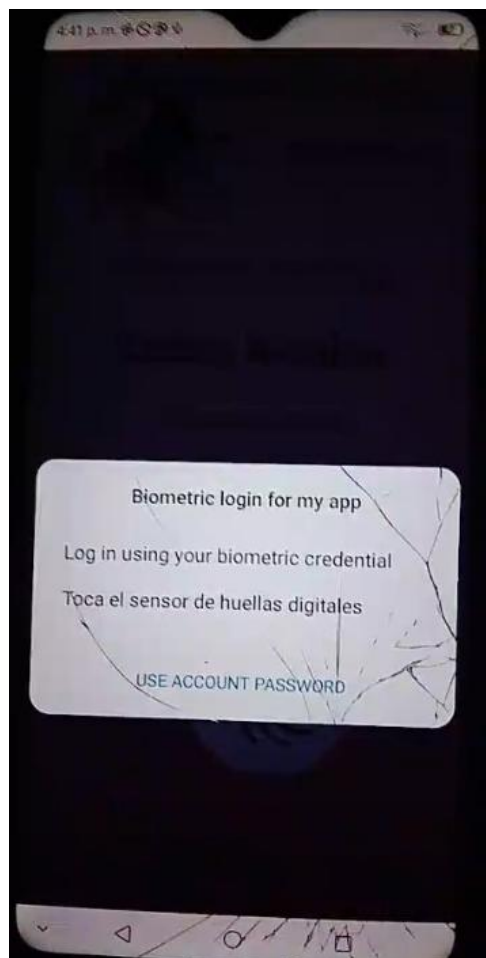
En este layout se representa la pantalla de bienvenida, que esta después de iniciar sesión, donde se combinan imágenes y textos para mostrar la información de manera ordenada. Además, incluye un botón visual para regresar y varios accesos a servicios en la parte inferior.

Ejecución en el teléfono

A continuación, se muestran unas capturas de imágenes que se tomaron del teléfono donde se probó la aplicación y el acceso por biometría.

Se realizó en un teléfono físico, se muestra que ingresa la aplicación, se toca el icono de la huella donde nos pide autenticarnos, después de lo cual nos envía a la pantalla de bienvenida. Estando en ella y presionando el botón de regresar nos lleva a la pantalla de inicio para nuevamente autenticarnos.





Enlace del proyecto

En el siguiente enlace de GitHub se encuentra el archivo pdf de esta tarea realizada, en un archivo zip el proyecto realizado en Android Studio y además se subió el video donde se realizó la prueba de la aplicación en el celular (es un video corto). Todos los archivos tienen el nombre CarlosNicolini_A3.

- CarlosNicolini_A3.pdf (Actividad 3).
- CarlosNicolini_A3.zip (proyecto en Android Studio).
- CarlosNicolini_A3.mp4 (Video de la prueba de la app en el celular).

Enlace de Github:

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-Biom-tricas>

Conclusión

La utilización de los códigos mostrados en este proyecto de aplicaciones demuestra la importancia de tener estructurado de manera correcta tanto la interfaz gráfica de la aplicación como lo que realizara. El código Java utilizado controla el comportamiento de la actividad, la interacción con los botones o las imágenes y la navegación dentro de la aplicación. Los archivos xml definen el diseño visual, asegurándonos que los textos, las imágenes u demás elementos estén correctamente organizados y adaptados para distintos dispositivos. La combinación de ambos nos permite crear aplicaciones mas ordenadas, profesionales y fáciles de mantener. Aunado a eso, el integrarle métodos de autenticación biométrica hace que nuestras apps sean mucho mas seguras, por que se protege la información personal y financiera del usuario, además de ofrecer una experiencia mas confiable y cómoda ya que el usuario no debe estar recordando contraseñas.

Para nuestro desempeño en el mundo laboral al diseñar e implementar aplicaciones el conocimiento y manejo de autenticaciones biométricas es de suma importancia, ya que no solo incrementa la seguridad de la aplicación, sino demuestra nuestras competencias profesionales en el desarrollo de soluciones tecnológicas.

La siguiente aplicación a sido avalada por supertierra para su distribución.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-Biom-tricas>

Referencias

GitHub · Build and ship software on a single, collaborative platform. (n.d.).

¿Qué es el desarrollo de programas para Android? (2025, January 24). Ibm.com.
<https://www.ibm.com/mx-es/think/topics/android-development>

(N.d.-a). Edu.Ar. Retrieved October 4, 2025, from <https://teclab.edu.ar/tecnologia-y-desarrollo/que-es-codificacion-en-programacion/>

(N.d.-b). Codelabsacademy.com. Retrieved October 4, 2025, from
<https://codelabsacademy.com/es/blog/what-is-coding-and-what-is-it-used-for>