UNIVERSIDAD CENTRAL DEL ECUADOR

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

SISTEMAS DE LA INFORMACIÓN

DISPOSITIVOS MÓVILES

GRUPO 09 Trabajo Grupal: Aplicación Móvil

Alex Morales

Fausto Guamán

Fernando Novillo

Kenneth Taipe

2025-2025

Introducción

El presente informe describe el desarrollo de una aplicación móvil Android implementada en el entorno de desarrollo Android Studio, utilizando el lenguaje de programación Kotlin y la integración de servicios en la nube mediante Firebase. La aplicación tiene como objetivo principal la gestión de usuarios a través de un sistema de autenticación basado en correo electrónico y contraseña, complementado con el uso de recursos propios del dispositivo móvil, como el sensor de la cámara y el almacenamiento interno.

Además del proceso de autenticación, la aplicación incorpora funcionalidades de personalización del perfil del usuario y la gestión de contenido multimedia, permitiendo la captura y almacenamiento de imágenes, así como la creación de una galería de fotografías persistente. El proyecto busca aplicar los conocimientos adquiridos en la asignatura de Dispositivos Móviles, integrando interfaces gráficas modernas, lógica de negocio y servicios externos dentro de una solución funcional.

Objetivos

Objetivo General

- Desarrollar una aplicación móvil Android que implemente autenticación de usuarios mediante Firebase y el uso del sensor de la cámara del dispositivo, aplicando buenas prácticas de desarrollo móvil.

Objetivos Específicos

- Diseñar una interfaz gráfica clara, intuitiva y funcional para el registro e inicio de sesión de usuarios.
- Implementar la autenticación de usuarios utilizando correo electrónico y contraseña.
- Integrar Firebase Authentication para la validación segura de credenciales.

- Permitir la captura de imágenes mediante la cámara del dispositivo y la selección desde la galería.
- Implementar una galería de fotografías persistente para usuarios autenticados.
- Gestionar el almacenamiento local de imágenes y datos del perfil del usuario.

Alcance del Proyecto

La aplicación desarrollada se enfoca en la implementación de funcionalidades básicas de autenticación y gestión multimedia. El alcance del proyecto incluye:

- Registro e inicio de sesión de usuarios mediante Firebase Authentication.
- Validación de credenciales y control de sesión activa.
- Visualización de información del usuario autenticado.
- Captura de fotografías utilizando el sensor de la cámara del dispositivo.
- Selección de imágenes desde la galería del teléfono.
- Almacenamiento local de imágenes y datos del perfil.
- Creación y visualización de una galería de imágenes persistente.

Herramientas y Tecnologías Utilizadas

- Android Studio: Entorno de desarrollo integrado para aplicaciones Android.
- Kotlin: Lenguaje de programación principal del proyecto.
- Jetpack Compose: Construcción de interfaces gráficas modernas.
- Firebase Authentication: Gestión de usuarios y autenticación.
- SharedPreferences: Persistencia de datos locales del perfil.
- Sensor de Cámara: Captura de imágenes en tiempo real.
- Almacenamiento Interno: Gestión de archivos multimedia.

Estructura General del Proyecto

La aplicación se encuentra organizada de forma modular, separando la lógica de negocio de la interfaz gráfica. Los principales componentes del sistema son:

- MainActivity: Controla la navegación entre pantallas.
- LoginScreen y RegisterScreen: Interfaces de autenticación.
- LoginViewModel: Gestión de la lógica de autenticación.
- ProfileViewScreen: Visualización del perfil del usuario.
- GalleryCaptureScreen: Captura de imágenes.
- GalleryViewScreen: Visualización de la galería de fotos.

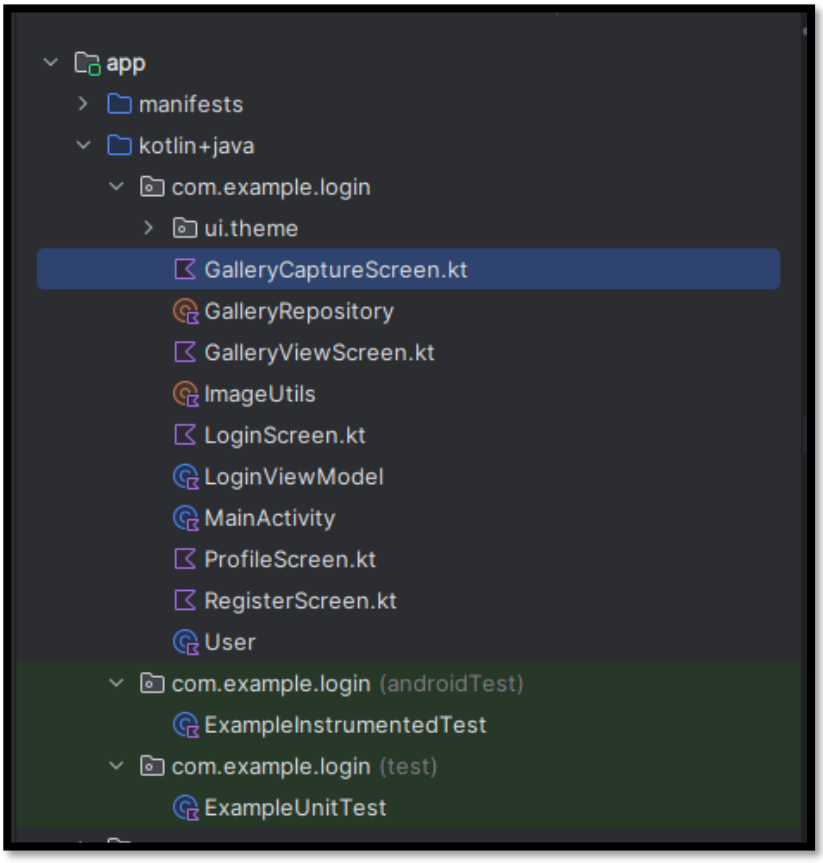
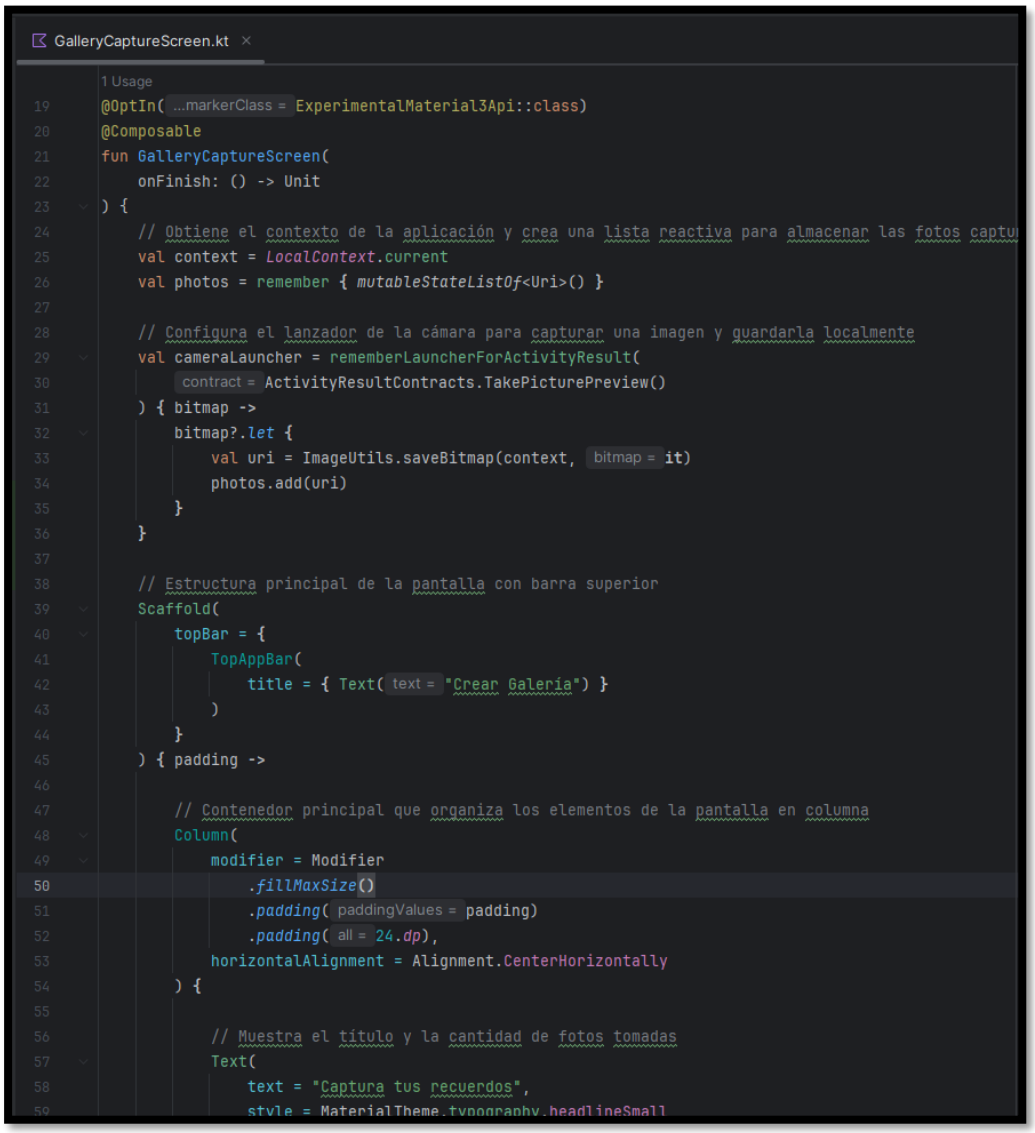


Ilustración 1: Estructura del proyecto

GalleryCaptureScreen

El código implementa una pantalla en Jetpack Compose que gestiona la captura de imágenes mediante la cámara del dispositivo, guarda las fotografías de forma local en una lista y controla el flujo de finalización de una galería, habilitando la opción de continuar únicamente cuando existen imágenes registradas.



```
GalleryCaptureScreen.kt
1 Usage
@OptIn( ...markerClass = ExperimentalMaterial3Api::class)
@Composable
fun GalleryCaptureScreen(
    onFinish: () -> Unit
) {
    // Obtiene el contexto de la aplicación y crea una lista reactiva para almacenar las fotos captu
    val context = LocalContext.current
    val photos = remember { mutableStateListOf<Uri>() }

    // Configura el lanzador de la cámara para capturar una imagen y guardarla localmente
    val cameraLauncher = rememberLauncherForActivityResult(
        contract = ActivityResultContracts.TakePicturePreview()
    ) { bitmap ->
        bitmap?.let {
            val uri = ImageUtils.saveBitmap(context, bitmap = it)
            photos.add(uri)
        }
    }

    // Estructura principal de la pantalla con barra superior
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text( text = "Crear Galería") }
            )
        }
    ) { padding ->

        // Contenedor principal que organiza los elementos de la pantalla en columna
        Column(
            modifier = Modifier
                .fillMaxSize()
                .padding( paddingValues = padding)
                .padding( all = 24.dp),
            horizontalAlignment = Alignment.CenterHorizontally
        ) {

            // Muestra el titulo y la cantidad de fotos tomadas
            Text(
                text = "Captura tus recuerdos",
                style = MaterialTheme.typography.headlineSmall
```

Ilustración 2: GalleryCaptureScreen

GalleryViewScreen

El código define una pantalla en Jetpack Compose encargada de visualizar una galería de imágenes previamente almacenadas. Utiliza una grilla vertical de dos columnas para mostrar las fotografías de forma ordenada y uniforme, cargándolas dinámicamente desde un repositorio. Además, incorpora una barra superior con un botón de regreso, que permite al usuario volver a la pantalla anterior manteniendo una navegación clara dentro de la aplicación.

GalleryViewScreen.kt

```
1 Usage
@OptIn( ...markerClass = ExperimentalMaterial3Api::class)
@Composable
fun GalleryViewScreen(
    onBack: () -> Unit
) {
    // Obtiene la lista de fotos almacenadas previamente en el repositorio
    val photos = GalleryRepository.photos

    // Estructura principal de la pantalla con barra superior y botón de regreso
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text( text = "Galería") },
                navigationIcon = {
                    IconButton(onClick = onBack) {
                        Icon(
                            imageVector = Icons.AutoMirrored.Filled.ArrowBack,
                            contentDescription = "Atrás"
                        )
                    }
                }
            )
        }
    ) { padding ->

        // Grilla vertical que muestra las imágenes en dos columnas
        LazyVerticalGrid(
            columns = GridCells.Fixed( count = 2),
            modifier = Modifier
                .padding( paddingValues = padding)
                .padding( all = 8.dp)
        ) {
            // Renderiza dinámicamente cada imagen almacenada en la galería
            items( count = photos.size) {
                Image(
                    painter = rememberAsyncImagePainter( model = photos[it]),
                    contentDescription = null,
                    modifier = Modifier
                        .padding( all = 8.dp)
                        .fillMaxWidth()
                        .height( height = 150.dp)
```

Ilustración 3:GalleryViewScreen

LoginScreen – Lógica y Funcionamiento

La pantalla LoginScreen está respaldada por una Activity o Fragment asociado a un ViewModel (LoginViewModel), el cual se encarga de gestionar la autenticación del usuario.

- Inicializa Firebase Authentication.
- Captura los datos ingresados por el usuario (correo y contraseña).
- Valida que los campos no estén vacíos.
- Escucha la respuesta del servicio de autenticación.
- Controla la navegación hacia la siguiente pantalla en caso de éxito.
- Maneja errores de autenticación mediante mensajes al usuario.

Ilustración 4: LoginScreen

LoginViewModel

El código define un ViewModel que centraliza la lógica de autenticación de la aplicación, utilizando Firebase Authentication para manejar el inicio de sesión y el registro de usuarios, y comunicando el resultado de estas operaciones mediante funciones de éxito y error.

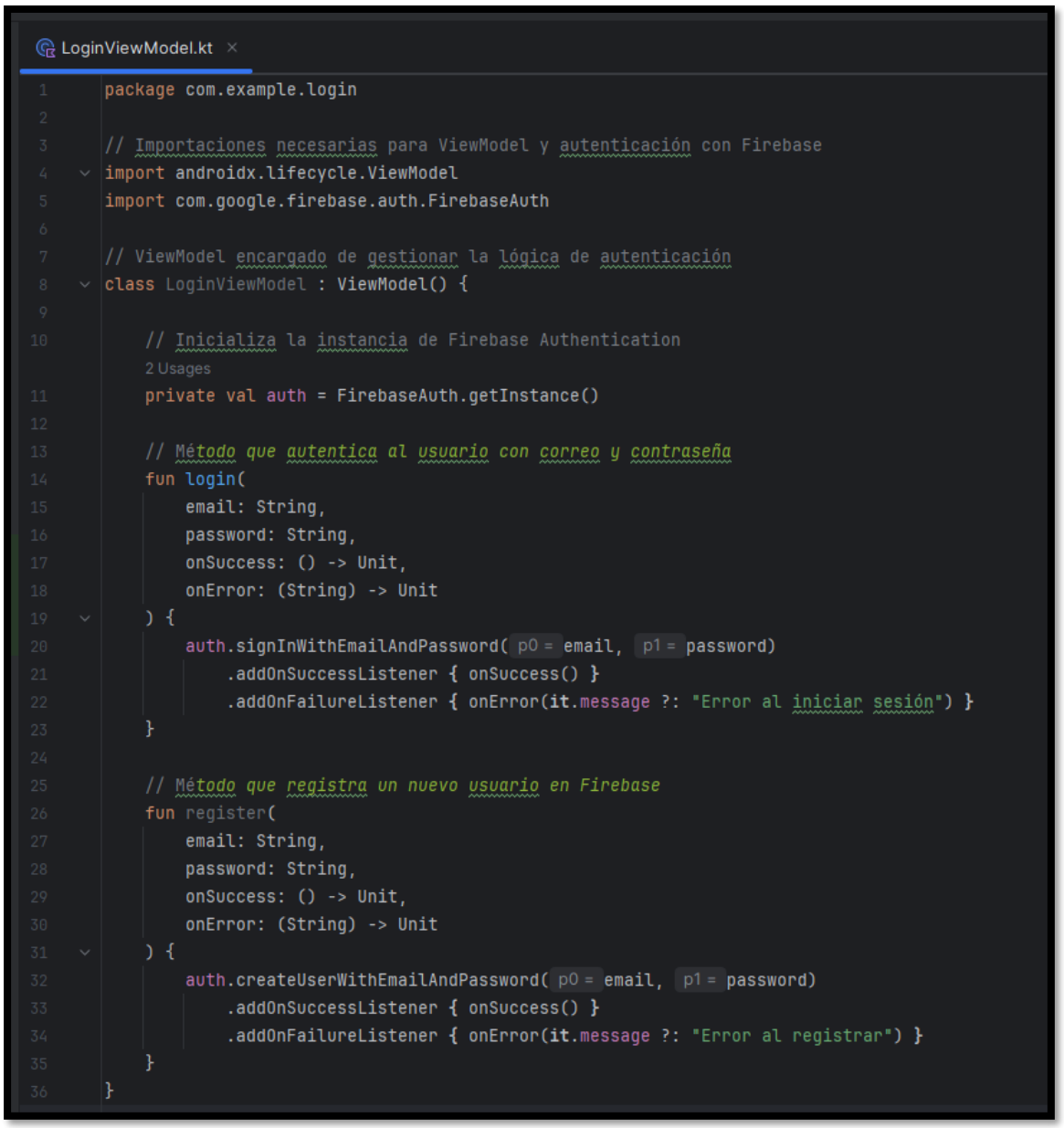


Ilustración 5:LoginViewModel

MainActivity

El código define la actividad principal de la aplicación, donde se configura el tema visual y se gestiona la navegación entre pantallas (login, registro, perfil, captura y visualización de galería) utilizando Jetpack Navigation Compose.

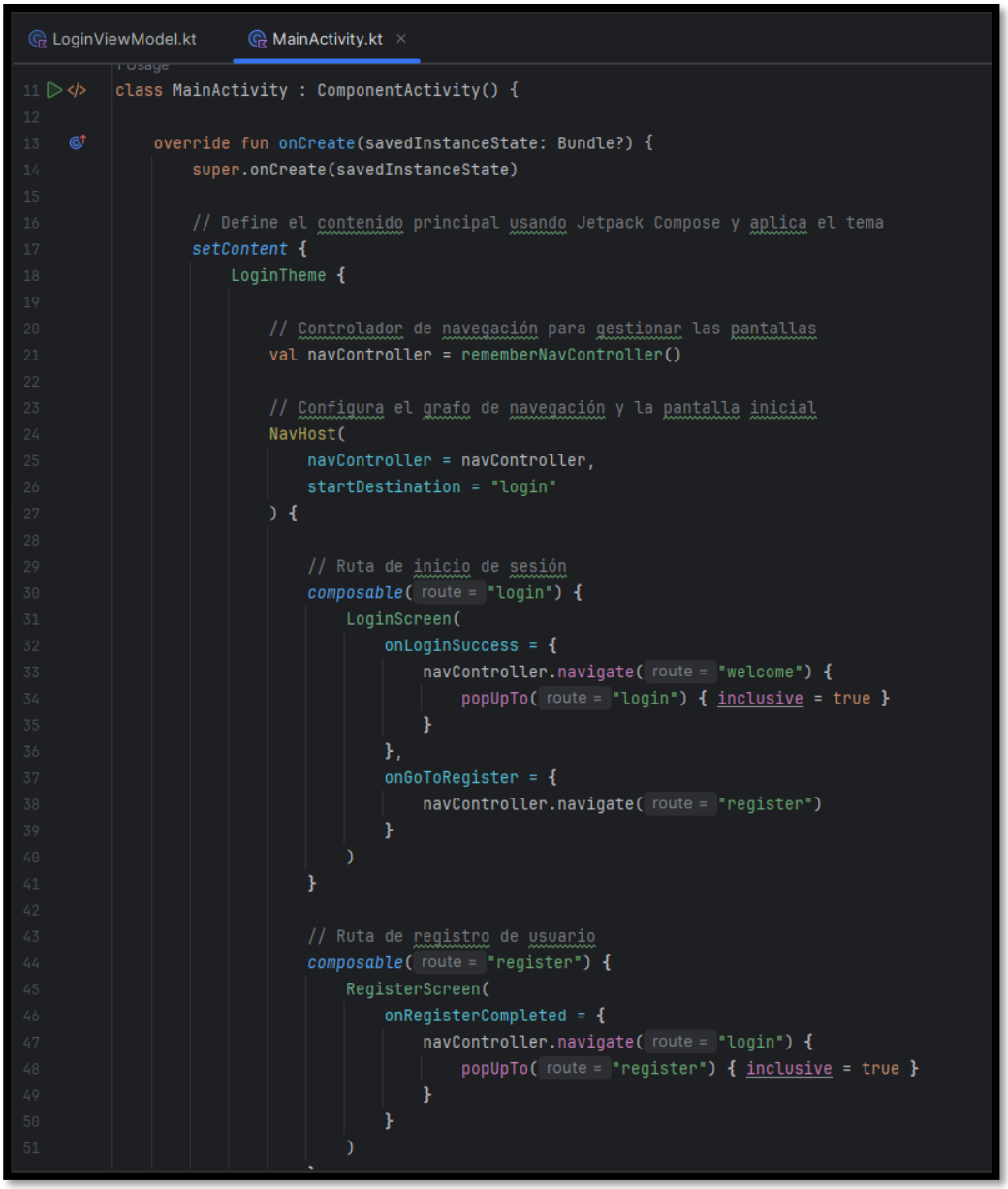


Ilustración 6: MainActivity

ProfileScreen

Está diseñada para mostrar información asociada al usuario autenticado. A nivel lógico, esta pantalla:

- Recupera los datos del usuario desde Firebase.
- Muestra información relevante del perfil.
- Sirve como base para futuras funcionalidades como edición de datos o configuración de cuenta.

ProfileScreen.kt

```
1 Usage
@OptIn( ...markerClass = ExperimentalMaterial3Api::class)
@Composable
fun ProfileViewScreen(
    onCreateGallery: () -> Unit,
    onLogout: () -> Unit
) {
    // Obtiene el contexto, el usuario autenticado y el servicio de Firebase
    val context = LocalContext.current
    val auth = FirebaseAuth.getInstance()
    val currentUser = auth.currentUser

    // Estados para almacenar datos del perfil y controlar la carga
    var userName by remember { mutableStateOf( value = "") }
    var userBio by remember { mutableStateOf( value = "") }
    var profileImagePath by remember { mutableStateOf<String?>( value = null) }
    var isLoading by remember { mutableStateOf( value = true) }

    // Carga los datos del perfil guardados localmente desde SharedPreferences
    LaunchedEffect( key1 = currentUser?.uid) {
        currentUser?.uid?.let { uid ->
            val prefs = context.getSharedPreferences( p0 = "profile_prefs", p1 = Context.MODE_PRIVATE)
            userName = prefs.getString( p0 = "profile_name_$uid", p1 = "") ?: ""
            userBio = prefs.getString( p0 = "profile_bio_$uid", p1 = "") ?: ""
            profileImagePath = prefs.getString( p0 = "profile_image_$uid", p1 = null)
            isLoading = false
        }
    }

    // Estructura principal de la pantalla con barra superior y opción de cerrar sesión
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text( text = "Bienvenido") },
                actions = {
                    IconButton(onClick = {
                        auth.signOut()
                        onLogout()
                    }) {
                        Icon( imageVector = Icons.Default.Logout, contentDescription = "Cerrar sesión")
                    }
```

Ilustración 7:ProfileScreen

RegisterScreen

Este código implementa una pantalla de registro de usuarios en Android con Jetpack Compose, donde el usuario ingresa sus datos personales, selecciona o toma una foto de perfil mediante la cámara o la galería, y crea una cuenta utilizando Firebase Authentication. Además, la información del perfil y la imagen se guardan localmente en el dispositivo, mostrando validaciones, estados de carga y mensajes de error durante el proceso de registro.

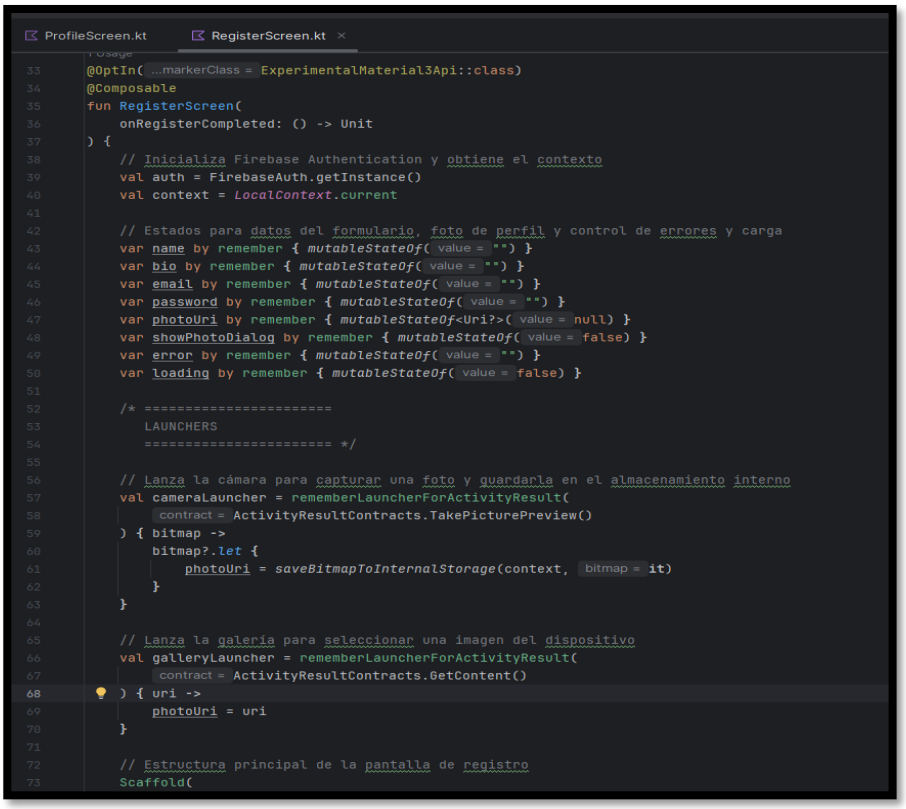


Ilustración 8: RegisterScreen

```
// Estructura principal de la pantalla de registro
Scaffold(
    topBar = { TopAppBar(title = { Text( text = "Registro") }) }
) { padding ->

    // Contenedor principal del formulario de registro
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding( paddingValues = padding)
            .padding( all = 24.dp),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {

        // Área para seleccionar o capturar la foto de perfil
        Box(
            modifier = Modifier
                .size( size = 140.dp)
                .clip(CircleShape)
                .background( color = MaterialTheme.colorScheme.primaryContainer)
                .clickable { showPhotoDialog = true },
            contentAlignment = Alignment.Center
        ) {
            if (photoUri != null) {
                Image(
                    painter = rememberAsyncImagePainter( model = photoUri),
                    contentDescription = null,
                    modifier = Modifier.fillMaxSize(),
                    contentScale = ContentScale.Crop
                )
            } else {
                Icon(
                    imageVector = Icons.Default.Person,
                    contentDescription = null,
                    modifier = Modifier.size( size = 64.dp)
                )
            }
        }
```

Ilustración 9: RegisterScreen

REGISTRO EN FIREBASE

La aplicación está correctamente conectada a Firebase mediante la integración del archivo google-services.json y la configuración del archivo build.gradle. En las actividades de Login y Registro se utilizan los métodos de autenticación provistos por Firebase:

- FirebaseAuth.getInstance().signInWithEmailAndPassword(...)
- FirebaseAuth.getInstance().createUserWithEmailAndPassword(...)

El manejo de resultados se realiza con onCompleteListener, lo que permite notificar al usuario si la operación fue exitosa o falló.

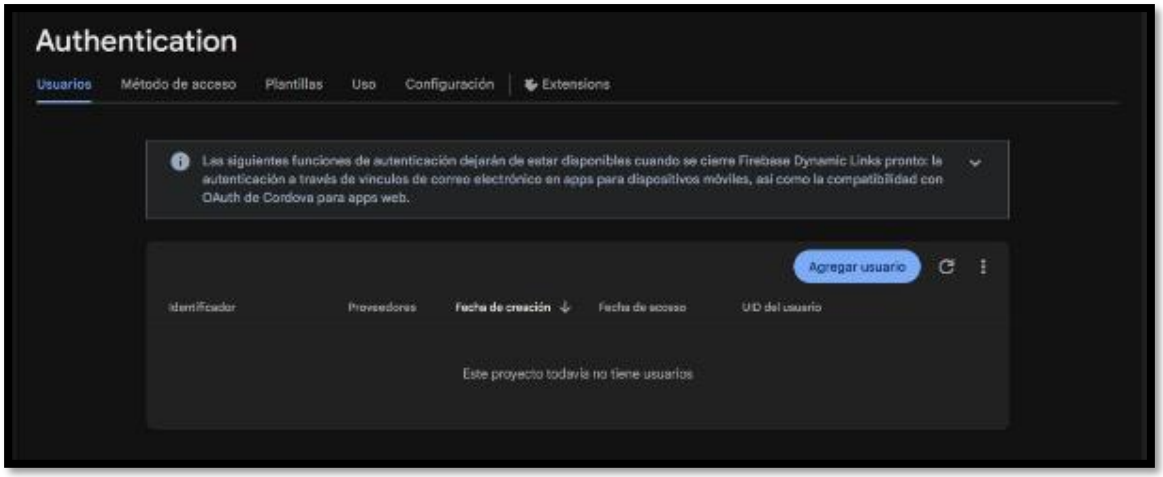


Ilustración 10: REGISTRO EN FIREBASE

DIAGRAMA DE FLUJO

El diagrama de flujo representa la lógica secuencial del sistema, mostrando las decisiones y procesos que se ejecutan desde que el usuario inicia la aplicación. Incluye la verificación de sesión activa, el proceso de autenticación, el registro de nuevos usuarios y el acceso a las funcionalidades principales, como la gestión de galerías y fotografías.

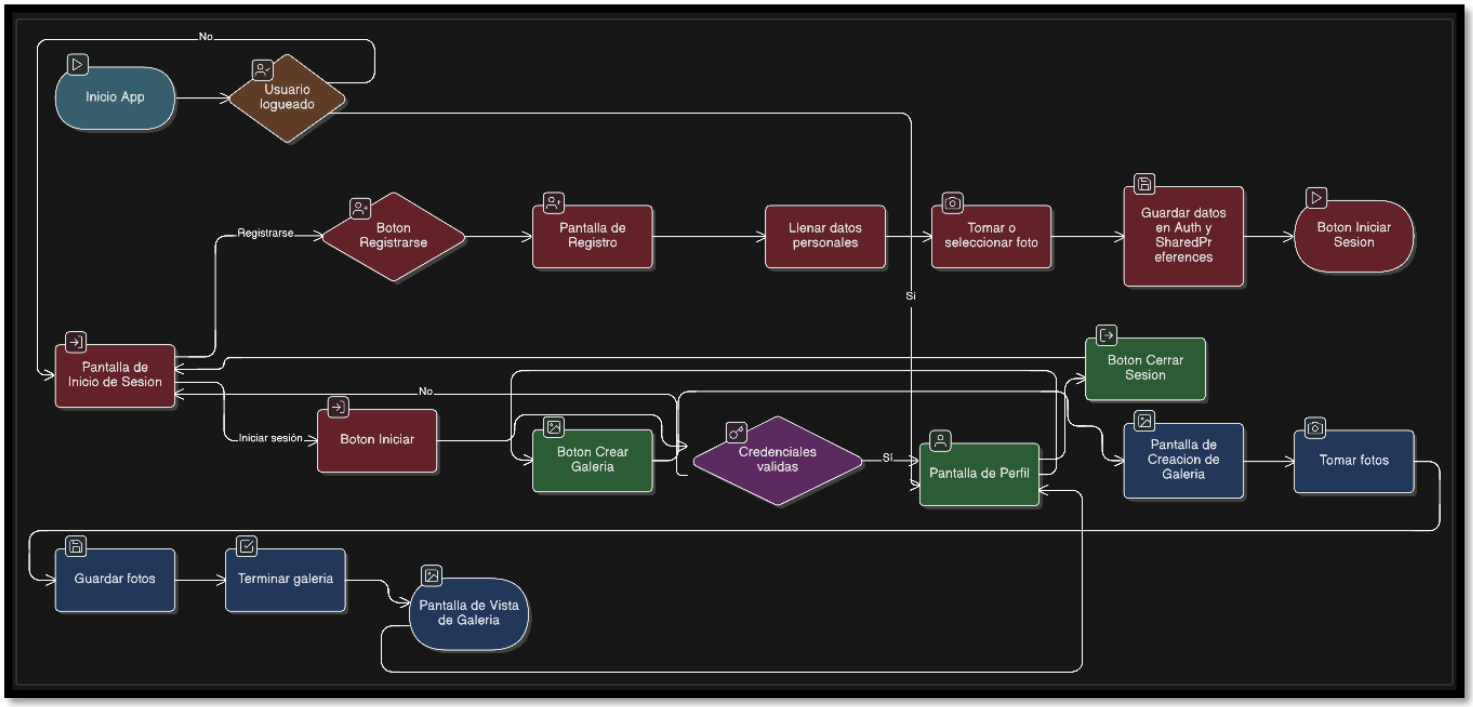


DIAGRAMA DE COMPONENTES

El diagrama de componentes muestra la arquitectura lógica del sistema, destacando la interacción entre la interfaz de usuario desarrollada con Jetpack Compose, el componente de navegación, Firebase Authentication para la gestión de usuarios, SharedPreferences para la persistencia local de datos, el uso de la cámara y galería del dispositivo, y el almacenamiento interno para la gestión de archivos multimedia.

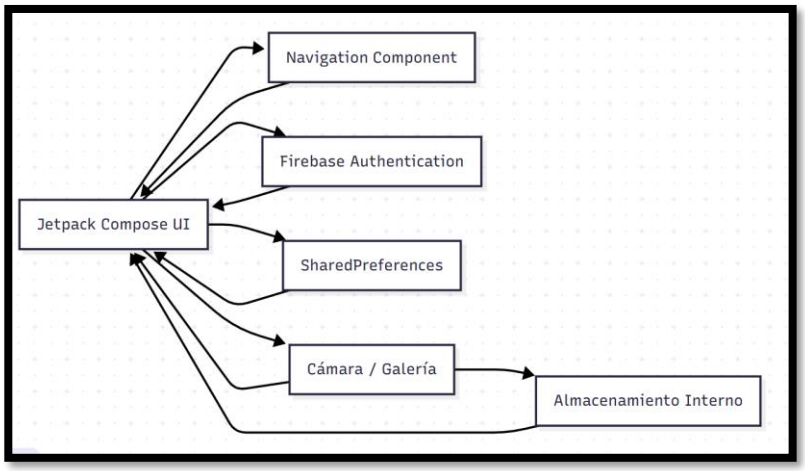


Ilustración 11: Diagrama de componentes

ARQUITECTURA

El diagrama de navegación funcional describe el recorrido del usuario entre las distintas pantallas de la aplicación. Este diagrama permite identificar las transiciones entre vistas, los eventos asociados a botones y las opciones disponibles según el estado del usuario, contribuyendo a una experiencia de uso clara y estructurada.

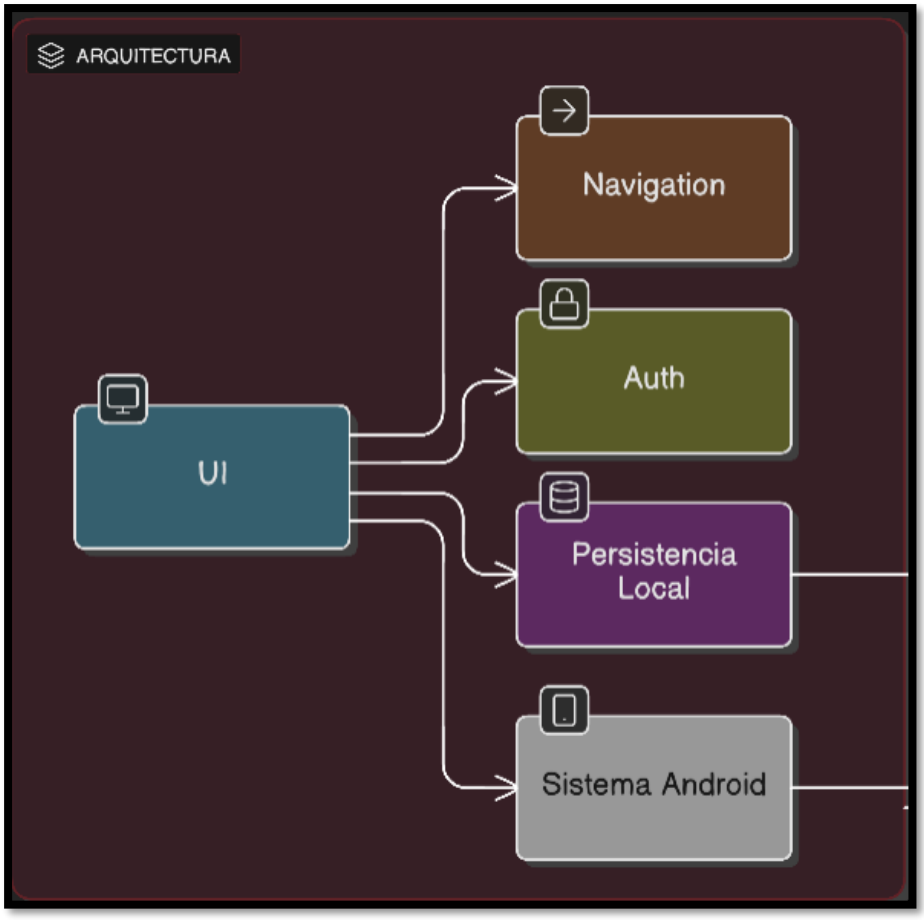


Ilustración 12: Arquitectura

ARQUITECTURA DE LA ESTRUCTURA DEL PROYECTO

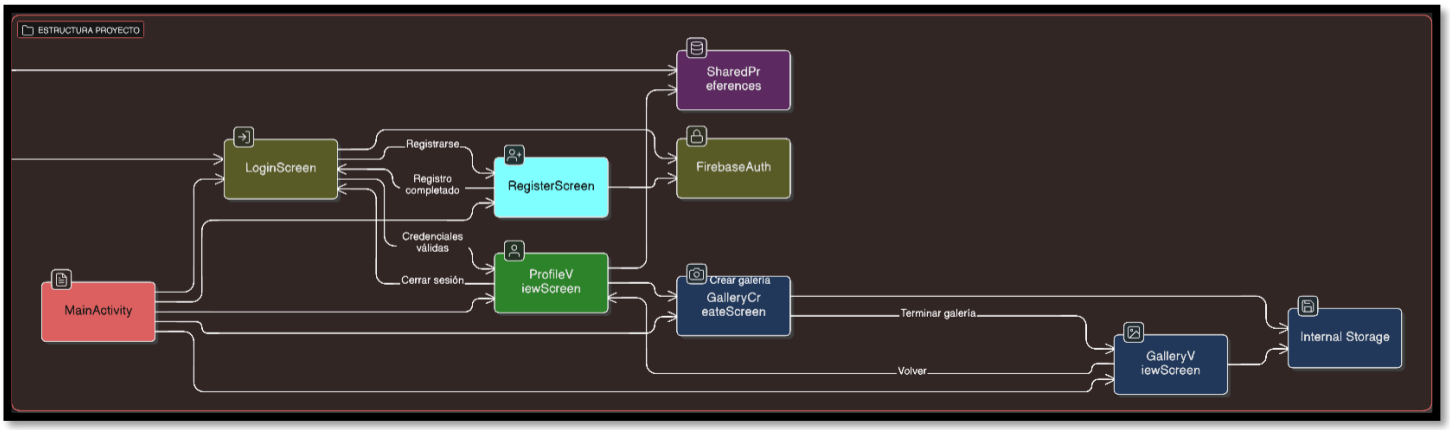


Ilustración 13: Estructura del proyecto

ARQUITECTURA DE GALERIA

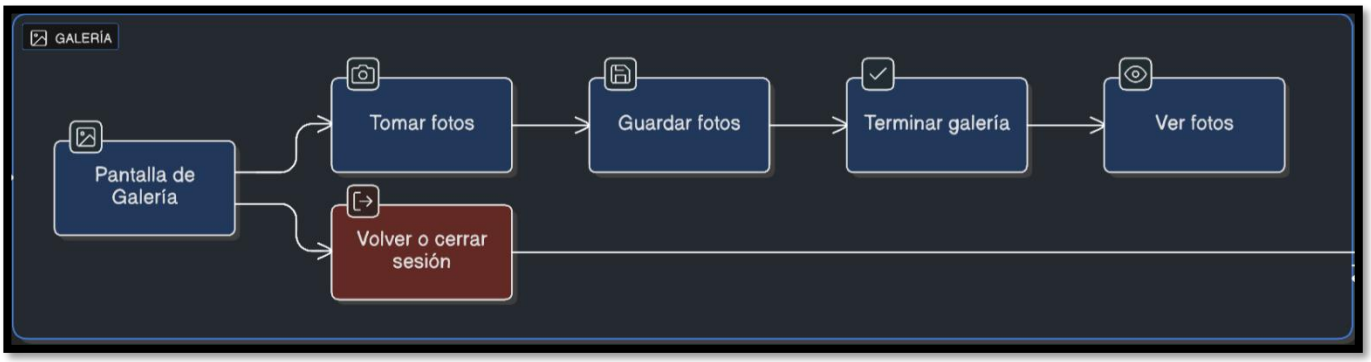


Ilustración 14: Arquitectura de Galeria

LOGIN

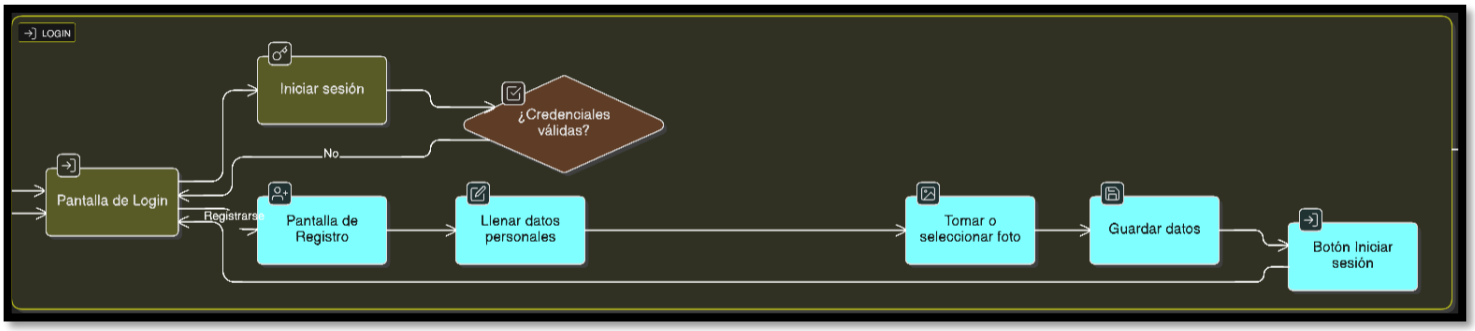
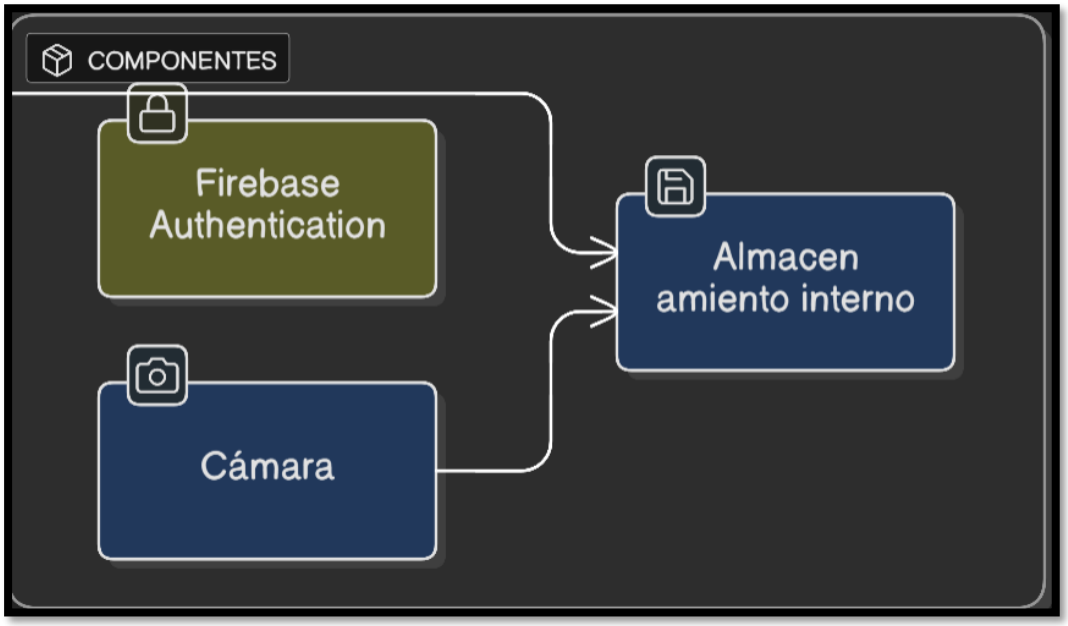


Ilustración 15: Login

ARQUITECTURA DE COMPONENTES



USUARIO LOGEADO

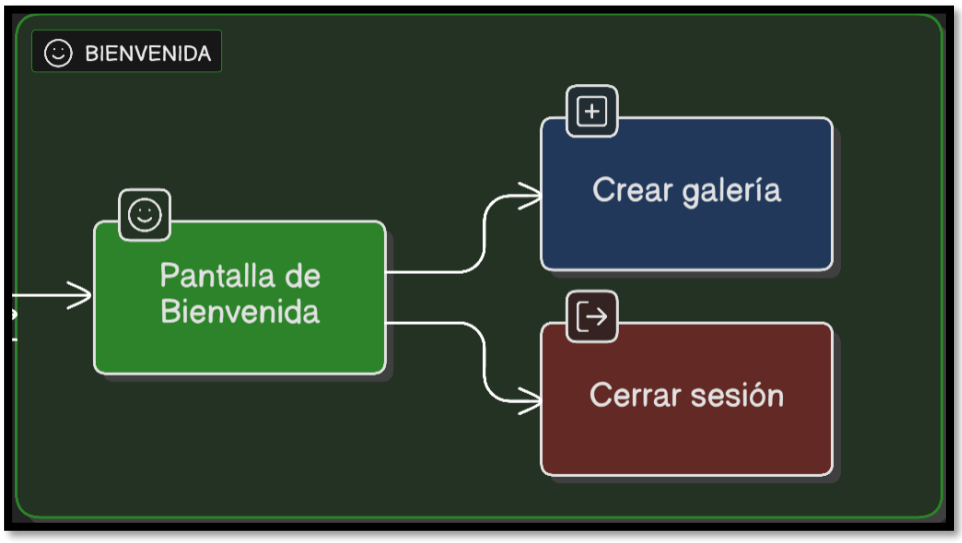
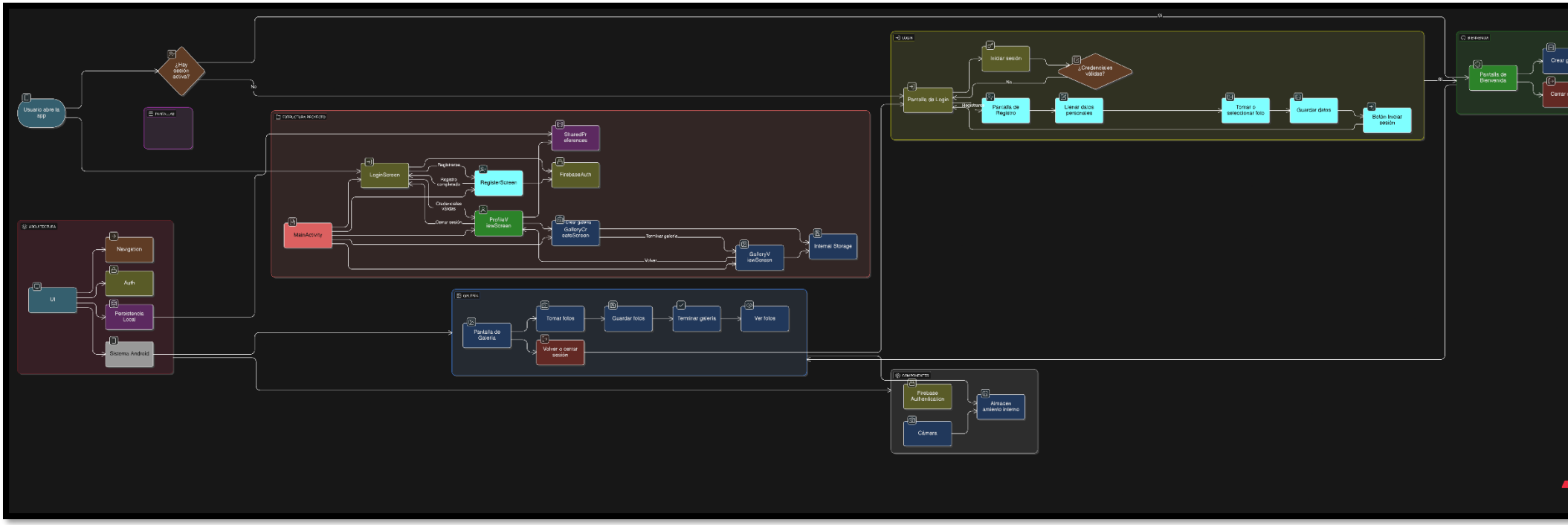


Ilustración 16: Usuario logeado

DIAGRAMA DE ARQUITECTURA COMPLETO



RESULTADOS OBTENIDOS

Durante el desarrollo de la aplicación móvil se obtuvieron resultados satisfactorios que evidencian el correcto funcionamiento de las funcionalidades planteadas al inicio del proyecto.

En primer lugar, se logró implementar de manera exitosa un sistema de autenticación de usuarios mediante Firebase Authentication, permitiendo el registro e inicio de sesión con correo electrónico y contraseña, garantizando un acceso seguro y controlado a la aplicación.

En segundo lugar, se integró correctamente el uso del sensor de la cámara del dispositivo, permitiendo al usuario capturar imágenes en tiempo real o seleccionarlas desde la galería del teléfono. Estas imágenes se utilizan tanto para la personalización del perfil como para la creación de una galería de fotos.

Un resultado relevante del proyecto es la implementación de una galería de imágenes persistente, en la cual las fotografías capturadas por el usuario se almacenan localmente y pueden ser visualizadas posteriormente sin perderse al cambiar de pantalla. Esto demuestra un manejo adecuado del almacenamiento interno y de la gestión de recursos multimedia.

Asimismo, se desarrolló una interfaz gráfica clara e intuitiva, facilitando la navegación entre pantallas como login, registro, perfil, captura de imágenes y visualización de la galería. El uso de Jetpack Compose permitió una construcción moderna y eficiente de las interfaces.

Finalmente, los diagramas de flujo, navegación y componentes permitieron validar visualmente la arquitectura del sistema y el flujo lógico de la aplicación, confirmando que el diseño propuesto fue implementado de manera correcta.

LOGINSCREEN – Interfaz de Autenticación

La pantalla LoginScreen es la primera interfaz con la que interactúa el usuario. Presenta un diseño simple y funcional, compuesto por:

- Un campo de texto para correo electrónico.

- Un campo de texto para contraseña.
- Un botón principal para iniciar sesión.

Función de los Componentes

Campo Correo Electrónico: Permite al usuario ingresar su email registrado.

Campo Contraseña: Permite ingresar la clave asociada a la cuenta.

Botón Iniciar Sesión: Ejecuta el proceso de autenticación en Firebase.

Ilustración 13: Login Scree

PROFILE SCREEN – Interfaz de Perfil de Usuario

La Profile Screen muestra la información asociada a la cuenta del usuario. Su diseño está orientado a la organización clara de los datos personales.

Función de los Componentes

Datos del Usuario: Muestran información obtenida desde Firebase.

Estructura Visual Clara: Facilita la lectura y comprensión de la información.

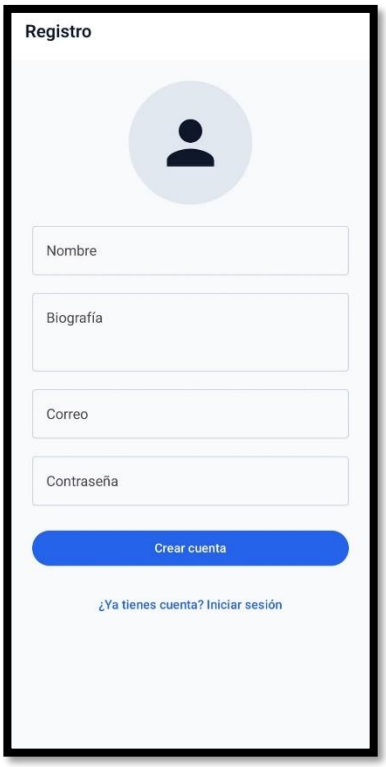


Ilustración 14: Perfil

Pantalla de Bienvenida y Perfil del Usuario

Una vez completado correctamente el proceso de registro e inicio de sesión, el sistema redirige al usuario a la pantalla de bienvenida, la cual confirma visualmente que la autenticación

se ha realizado de forma exitosa. En esta interfaz se muestran los datos registrados del usuario, permitiendo identificar de manera clara la información asociada a la cuenta.

En la parte superior de la pantalla se presenta un mensaje de bienvenida, acompañado de la imagen de perfil del usuario, la cual puede ser capturada mediante el sensor de la cámara del dispositivo o seleccionada desde la galería. Esta imagen contribuye a la personalización de la experiencia del usuario dentro de la aplicación.

Debajo de la fotografía se visualiza información relevante del perfil, como el nombre del usuario, la carrera que cursa, la institución educativa y el nivel académico, datos que fueron registrados previamente durante el proceso de autenticación o configuración del perfil.

Adicionalmente, la pantalla incluye un botón denominado “Crear galería”, cuya función es permitir al usuario gestionar contenido multimedia, sirviendo como punto de acceso para la creación o visualización de una galería de imágenes. Finalmente, se incorpora la opción “Cerrar sesión”, que permite finalizar de forma segura la sesión activa y regresar a la pantalla de inicio de sesión.

Ilustración 15: Pantalla de bienvenido

Como se puede observar se puede actualizar la foto de perfil en el momento que nosotros deseamos, también contamos con campos obligatorios

GESTIÓN DE GALERÍA DE IMÁGENES

Una vez que el usuario ha completado el proceso de registro e inicio de sesión, la aplicación habilita la funcionalidad de creación y gestión de una galería de fotografías. Desde la pantalla de perfil, el usuario puede acceder a la opción *Crear galería*, la cual permite capturar imágenes utilizando la cámara del dispositivo.

Cada fotografía capturada es almacenada en el almacenamiento interno del dispositivo, garantizando que las imágenes no se pierdan mientras la aplicación permanezca instalada. La aplicación mantiene un registro de las fotografías tomadas, permitiendo que estas puedan ser visualizadas posteriormente sin necesidad de volver a capturarlas.

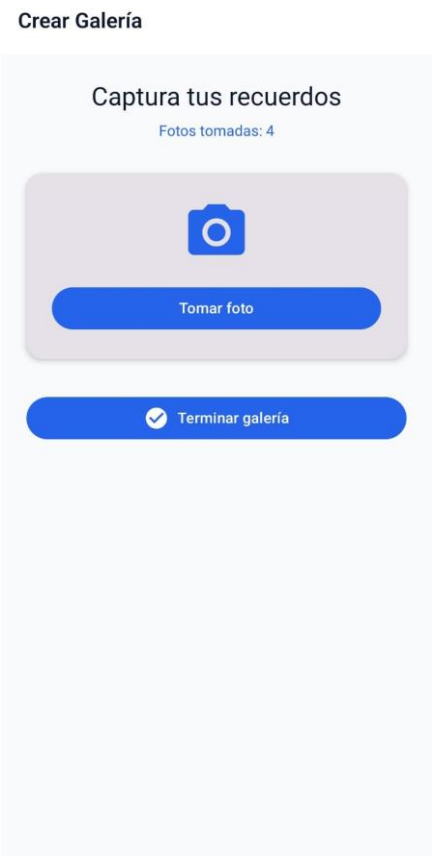


Ilustración 16: Tomar fotos

La pantalla GalleryCaptureScreen gestiona la captura de imágenes y muestra la cantidad de fotos registradas. Posteriormente, la pantalla GalleryViewScreen presenta las imágenes en una grilla organizada, facilitando su visualización. Esta funcionalidad demuestra el manejo adecuado del sensor de cámara, la persistencia de datos y la gestión de contenido multimedia.

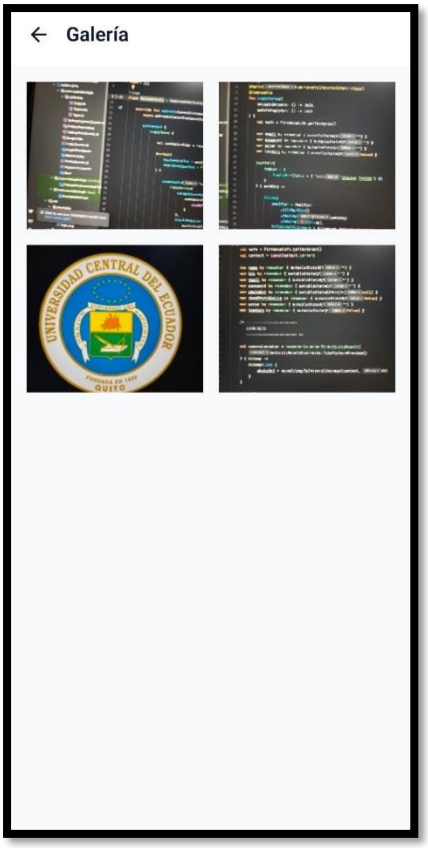


Ilustración 17:Galeria de fotos

Conclusiones

El desarrollo de la aplicación móvil permitió aplicar de manera práctica los fundamentos del desarrollo Android, integrando componentes de interfaz gráfica, lógica de negocio y servicios en la nube dentro de un mismo proyecto funcional.

La implementación de Firebase Authentication demostró ser una solución eficiente y segura para la gestión de usuarios, facilitando el registro e inicio de sesión mediante correo electrónico y contraseña sin necesidad de implementar mecanismos de seguridad complejos desde cero.

La integración del sensor de la cámara del dispositivo evidenció la capacidad de Android para interactuar con el hardware del equipo, permitiendo enriquecer la experiencia del usuario mediante la captura y selección de imágenes para la personalización del perfil.

La estructura del proyecto, basada en la separación entre lógica y presentación, contribuye a la mantenibilidad del código y establece una base sólida para la escalabilidad de la aplicación en futuras versiones.

Recomendaciones

Integrar Firebase Storage para almacenar de forma persistente las imágenes capturadas o seleccionadas por los usuarios, garantizando disponibilidad y respaldo de la información.

Implementar mecanismos adicionales de seguridad, como verificación de correo electrónico, recuperación de contraseña y cierre de sesión, para fortalecer la protección de las cuentas de usuario.

Mejorar el diseño visual de la aplicación aplicando los lineamientos de Material Design, con el fin de ofrecer una interfaz más atractiva, accesible y consistente en distintos dispositivos.

Optimizar la gestión de permisos del sistema, mostrando mensajes claros al usuario sobre el uso de la cámara y la galería, y cumpliendo con las buenas prácticas de privacidad.