

PRÁCTICA 3

"Aplicaciones nativas"

Salvador Costilla Caballero

ESCOM | IPN

Introducción

Durante esta práctica se desarrollaron dos aplicaciones nativas para Android utilizando el lenguaje Kotlin. La primera fue un **Gestor de Archivos**, y la segunda una **Aplicación de Cámara y Micrófono**. La elección de estos proyectos se basó en el interés por profundizar en el desarrollo exclusivo para Android, dejando de lado soluciones multiplataforma como Flutter, con la cual ya se tenía experiencia previa. También se contaba con conocimientos sobre desarrollo en iOS, por lo que se optó por enfocarse completamente en Android para conocer sus APIs, restricciones de seguridad, permisos, temas visuales y almacenamiento local.

Desarrollo

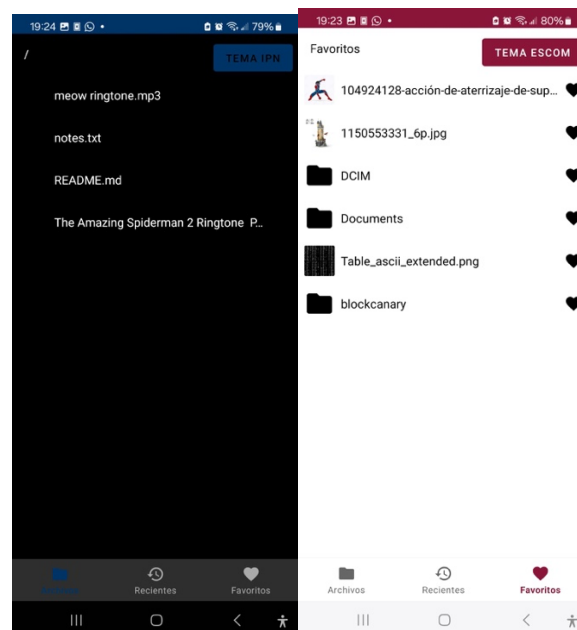
Ejercicio 1: Gestor de Archivos para Android (Kotlin)

1. Funcionalidades principales:

- Exploración de almacenamiento interno y externo
- Visualización jerárquica de carpetas y archivos
- Apertura de archivos de texto (.txt, .md, etc.)
- Visualización de imágenes con zoom y rotación
- Diálogo para archivos no compatibles, ofreciendo abrir con otras apps

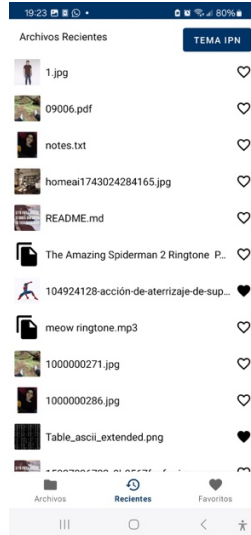
2. Interfaz de Usuario:

- Dos temas personalizables:
 - Tema **Guinda** (IPN)
 - Tema **Azul** (ESCOM)
- Adaptación automática a modo claro/oscuro del sistema
- Interfaz responsiva para diferentes tamaños de pantalla



3. Almacenamiento local:

- Historial de archivos recientes con `SharedPreferences`
- Sistema de favoritos utilizando `Room`
 - Cache de miniaturas de imágenes para mejor rendimiento

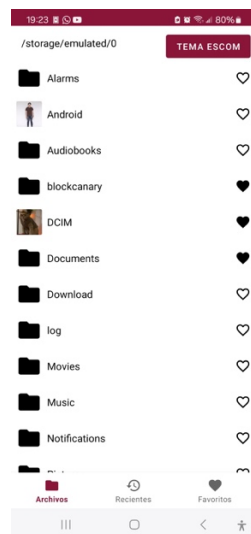


4. Permisos y seguridad:

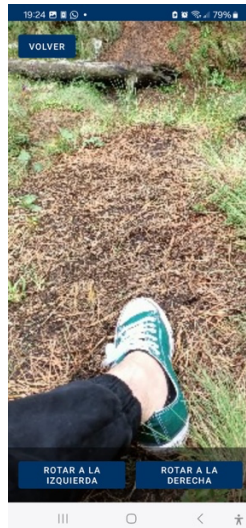
- Solicitud y gestión de permisos con `ActivityResultContracts`
- Manejo de rutas inaccesibles con excepciones
- Cumplimiento de restricciones de seguridad modernas (scoped storage)

Capturas de pantalla:

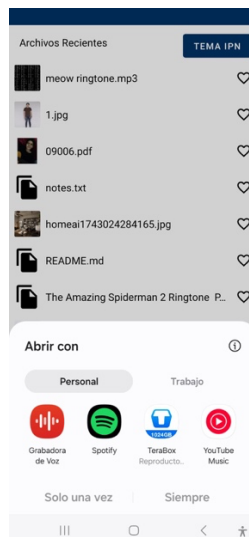
- Explorador de archivos



- Vista previa de imagen con zoom



- Diálogo "Abrir con"



Ejercicio 2: Aplicación de Cámara y Micrófono para Android (Kotlin)

1. Funcionalidades principales:

- Captura de fotografías
- Grabación de audio
- Almacenamiento organizado de archivos capturados

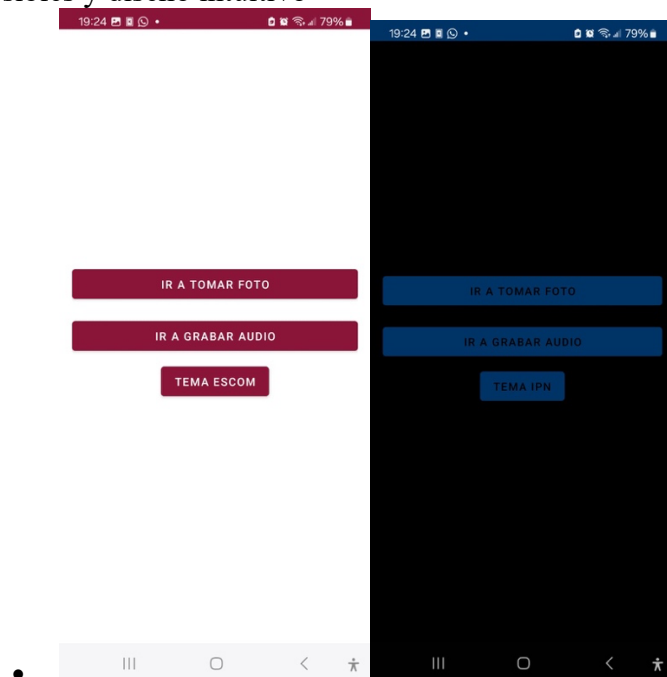
2. Galería integrada:

- Visualizador de imágenes con opciones de edición básica
- Reproductor de audio integrado
- Organización del contenido por fecha

3. Interfaz de Usuario:

- Misma paleta de temas: **Guinda** y **Azul**

- Modo claro/oscuro automático
- Controles accesibles y diseño intuitivo



Capturas de pantalla:

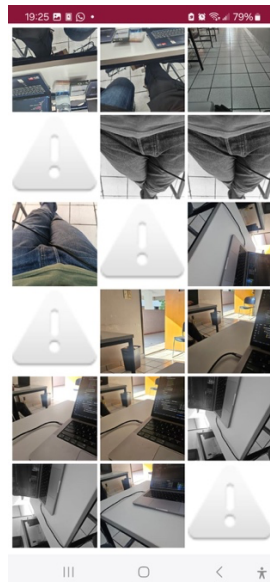
- Interfaz de cámara



- Reproductor de audio



- Vista de galería organizada



Pruebas Realizadas

Las aplicaciones fueron probadas en:

- **Dispositivo físico:** Samsung Galaxy s23 plus (Android 14)

Resultados:

- Correcta adaptación de UI en distintos tamaños
 - Funcionamiento estable con cambios de tema y modo del sistema
 - Permisos correctamente gestionados en todos los casos
-

Conclusiones

Esta práctica permitió profundizar en el desarrollo nativo en Android utilizando Kotlin, abarcando aspectos clave como:

- Gestión moderna de permisos
- Scoped storage y restricciones de seguridad
- Interfaz adaptable con temas y modo oscuro
- Integración de cámara, micrófono y almacenamiento

El desarrollo de estas aplicaciones fue desafiante pero enriquecedor. Se mejoró la comprensión del ciclo de vida de actividades, el uso de contratos de actividad (`ActivityResultContract`), y la implementación de interfaces modernas responsivas.

Bibliografía

- Android Developers. (2024). *Storage use in Android*. <https://developer.android.com/guide/topics/data>
- CanHub. (2024). *Android Image Cropper*. <https://github.com/CanHub/Android-Image-Cropper>
- Google Developers. (2024). *CameraX API*. <https://developer.android.com/training/camerax>
- Room Persistence Library. (2024). <https://developer.android.com/jetpack/androidx/releases/room>
- Kotlin Programming Language. (2024). <https://kotlinlang.org/docs/home.html>