



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Programa académico / Plan de estudios

Ingeniería en Sistemas Computacionales / 2020

Unidad de aprendizaje

Desarrollo de aplicaciones móviles nativas

Práctica 3: Aplicaciones Nativas

Objetivo: Desarrollar aplicaciones móviles nativas que interactúen con recursos del dispositivo, implementando almacenamiento local y funcionamiento sin conexión. El estudiante deberá elegir y completar 2 de los 3 ejercicios principales propuestos, con un ejercicio opcional para quienes deseen expandir sus conocimientos en desarrollo multiplataforma.

Instrucciones Generales:

- Debe seleccionar y completar 2 de los 3 ejercicios principales (Ejercicios 1, 2 y 3)
- El Ejercicio 4 es completamente opcional
- Todas las aplicaciones deben funcionar sin conexión a Internet, gestionando y almacenando datos localmente
- Cada aplicación debe implementar los temas personalizables especificados

Ejercicio 1: Gestor de Archivos para Android (Kotlin)

1. Funcionalidades principales:

- Explorar directorios del almacenamiento interno y externo (si está disponible)
- Visualizar la estructura jerárquica de carpetas y archivos
- Abrir y visualizar archivos de texto (.txt, .md, etc.)
- Visualizar imágenes con opciones de zoom y rotación
- Implementar diálogo para gestionar archivos que la aplicación no puede abrir directamente, ofreciendo abrir con otras aplicaciones

2. Interfaz de Usuario:

- Implementar temas personalizables:
 - Tema Guinda (color representativo del IPN)
 - Tema Azul (color representativo de la ESCOM)
- Ambos temas deben adaptarse automáticamente al modo del sistema:
 - Versión clara cuando el dispositivo esté en modo claro
 - Versión oscura cuando el dispositivo esté en modo oscuro
- Diseñar una interfaz responsiva que funcione correctamente en diferentes tamaños de pantalla
- Implementar navegación intuitiva entre directorios mediante breadcrumbs o indicadores visuales

3. Almacenamiento local:

- Mantener historial de archivos recientes utilizando almacenamiento persistente
 - Implementar sistema de favoritos utilizando SharedPreferences o Room
 - Gestionar cache para miniaturas de imágenes para mejorar rendimiento
4. **Permisos y seguridad:**
- Solicitar y gestionar correctamente los permisos de acceso al almacenamiento
 - Implementar manejo adecuado de excepciones para rutas inaccesibles
 - Respetar las restricciones de seguridad de Android para el acceso a archivos
-

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

1. **Funcionalidades principales:**
- Capturar fotos utilizando la cámara del dispositivo
 - Grabar audio utilizando el micrófono
 - Implementar opciones básicas de personalización de captura:
 - Para fotos: filtros, flash, temporizador
 - Para audio: niveles de sensibilidad, temporizador de grabación
 - Gestionar adecuadamente el almacenamiento de los archivos capturados
2. **Galería integrada:**
- Implementar visualizador de imágenes capturadas con opciones básicas de edición
 - Crear reproductor de audio para las grabaciones realizadas
 - Permitir organizar el contenido en categorías o álbumes
3. **Interfaz de Usuario:**
- Aplicar los mismos temas personalizables:
 - Tema Guinda (color representativo del IPN)
 - Tema Azul (color representativo de la ESCOM)
 - Adaptación automática al modo del sistema (claro/oscuro)
 - Diseñar una interfaz intuitiva con controles accesibles y claros
 - Implementar gestos y animaciones para mejorar la experiencia de usuario
4. **Almacenamiento local:**
- Guardar todos los archivos capturados en el almacenamiento del dispositivo
 - Implementar base de datos para metadatos (fecha, ubicación, etiquetas)
 - Mantener miniaturas y cache para acceso rápido a la galería
 - Permitir exportar e importar contenido
-

Ejercicio 3: Desarrollo con Flutter (Gestor de Archivos o Cámara y Micrófono)

1. **Opciones de desarrollo:** El estudiante debe elegir una de estas opciones:
- **Opción A: Gestor de archivos** con funcionalidades similares al Ejercicio 1
 - **Opción B: Aplicación de cámara y micrófono** con funcionalidades similares al Ejercicio 2
2. **Requisitos técnicos:**
- Desarrollar la aplicación utilizando el framework Flutter
 - Implementar arquitectura limpia (Clean Architecture) o similar
 - Utilizar paquetes y plugins adecuados para acceso a recursos del dispositivo
 - Implementar gestión de estado (Provider, Bloc, Riverpod o similar)
3. **Interfaz de Usuario:**
- Aplicar los mismos temas personalizables:
 - Tema Guinda (color representativo del IPN)
 - Tema Azul (color representativo de la ESCOM)
 - Adaptación automática al modo del sistema (claro/oscuro)
 - Diseñar utilizando Material Design o Cupertino según preferencia
 - Garantizar que la interfaz sea consistente en ambas plataformas (iOS y Android)
4. **Almacenamiento local:**
- Implementar persistencia utilizando soluciones multiplataforma como Hive, SQLite o similar

- Gestionar permisos de acceso a recursos de manera adecuada en ambas plataformas
 - Optimizar el acceso a recursos para mejorar rendimiento y experiencia de usuario
-

Ejercicio 4: Desarrollo para iOS con Swift (Opcional)

1. **Opciones de desarrollo:** El estudiante puede elegir entre desarrollar:
 - Un gestor de archivos
 - Una aplicación de cámara y micrófono
 2. **Requisitos técnicos:**
 - Desarrollar utilizando Swift y UIKit o SwiftUI
 - Aplicar patrones de diseño recomendados para iOS (MVC, MVVM, o similar)
 - Utilizar Core Data para persistencia local
 - Implementar permisos según directrices de Apple
 3. **Interfaz de Usuario:**
 - Implementar temas IPN y ESCOM adaptados al estilo de diseño de iOS
 - Compatibilidad con modo claro/oscuro
 - Seguir las Human Interface Guidelines de Apple
 - Adaptabilidad a diferentes tamaños de dispositivos iOS
 4. **Características específicas de iOS:**
 - Implementar al menos dos características específicas de iOS:
 - Widgets para pantalla de inicio
 - Soporte para Apple Pencil (en dispositivos compatibles)
 - Integración con Siri Shortcuts
 - Soporte para funciones de accesibilidad
-

Consideraciones importantes:

- **Funcionamiento sin conexión:** Todas las aplicaciones deben ser completamente funcionales sin requerir conexión a Internet, utilizando almacenamiento local para persistencia de datos.
 - **Permisos:** Gestionar correctamente los permisos de acceso a archivos, cámara y micrófono, con mensajes claros para el usuario.
-

Entrega de la Práctica:

1. **Código fuente:**
 - Repositorio GitHub con el código de las aplicaciones seleccionadas
 - README detallado con instrucciones de instalación y uso
 - Estructura clara y comentarios explicativos
2. **APK/IPA instalables:**
 - Versiones compiladas de las aplicaciones
3. **Informe de la práctica:** Siguiendo esta estructura:
 - **Portada:** Nombre completo, número de boleta, asignatura, profesor y fecha.
 - **Introducción:** Explicación general de las aplicaciones desarrolladas y justificación de elección.
 - **Desarrollo:** Descripción técnica y capturas de pantalla de cada aplicación.
 - **Pruebas realizadas:** Documentación de pruebas en diferentes dispositivos/tamaños de pantalla.
 - **Conclusiones:** Reflexiones sobre la experiencia y aprendizajes obtenidos.
 - **Bibliografía:** Fuentes consultadas en formato APA.

Fecha de Entrega: La fecha límite para la entrega de esta práctica es el jueves 10 de abril de 2025. No se aceptarán entregas fuera de tiempo y forma.