Documentación del Proyecto: AppNotas

- Programación Multimedia y Dispositivos Móviles (PMDM)

Francisco Javier Soler Aguado
2º DAM

Indice

1. Introducción

El presente proyecto consiste en el desarrollo de una aplicación móvil nativa para el sistema operativo Android, denominada AppNotas. El objetivo principal de la aplicación es permitir al usuario gestionar información personal mediante la creación, edición y organización de notas de texto.

La aplicación ha sido desarrollada utilizando el entorno de desarrollo Android Studio y el lenguaje de programación Kotlin. Para garantizar la persistencia de los datos entre sesiones, se ha implementado una base de datos relacional local utilizando SQLite, lo que permite que la información se conserve incluso si la aplicación se cierra o el dispositivo se reinicia.

---

2. Análisis de Requisitos

Para cumplir con las especificaciones del boletín y las necesidades del usuario, se han definido los siguientes requisitos funcionales y no funcionales:

2.1. Funcionalidades

- **Gestión de Notas (CRUD):**
    - **Crear:** El usuario debe poder crear nuevas notas con un título y un contenido.
    - **Leer:** Visualización de la lista de notas existentes en la pantalla principal.
    - **Actualizar:** Posibilidad de editar el título y contenido de una nota existente.
    - **Borrar:** Eliminación de notas, tanto individualmente como mediante selección múltiple.
- **Gestión de Categorías:**
    - Creación de nuevas categorías personalizadas.
    - Eliminación de categorías existentes.
    - Asignación de notas a categorías específicas.
- **Filtrado y Organización:**
    - Filtrar la lista de notas visibles seleccionando una categoría en la barra superior.
    - Opción para "Mover a categoría".
- **Persistencia:** Todos los datos (notas y categorías) se guardan en una base de datos local.

2.2. Requisitos No Funcionales

- **Lenguaje:** Kotlin.
- **Base de Datos:** SQLite

---

3. Diseño

3.1. Arquitectura de la Aplicación

La aplicación sigue una arquitectura clásica de Android donde las Activities actúan como controladores que gestionan la interfaz de usuario y coordinan la comunicación entre la vista (XML) y el modelo de datos (Base de datos).

- **Modelo**: Clases de datos (Nota, Categoria) y gestor de base de datos (DBHelper).
- **Vista**: Archivos XML de layout (activity_main.xml, item_nota.xml, etc.).
- **Controlador**: Activities (MainActivity, NotaActivity, etc.) y Adaptadores.

3.2. Estructura de Base de Datos (Modelo E-R)

Se utiliza una base de datos SQLite llamada `db_notas`. El esquema consta de dos tablas relacionadas:

1. **Tabla categorias**:
    - id (INTEGER PK AUTOINCREMENT): Identificador único.
    - name (TEXT UNIQUE): Nombre de la categoría.
2. **Tabla notas**:
    - id (INTEGER PK AUTOINCREMENT): Identificador único.

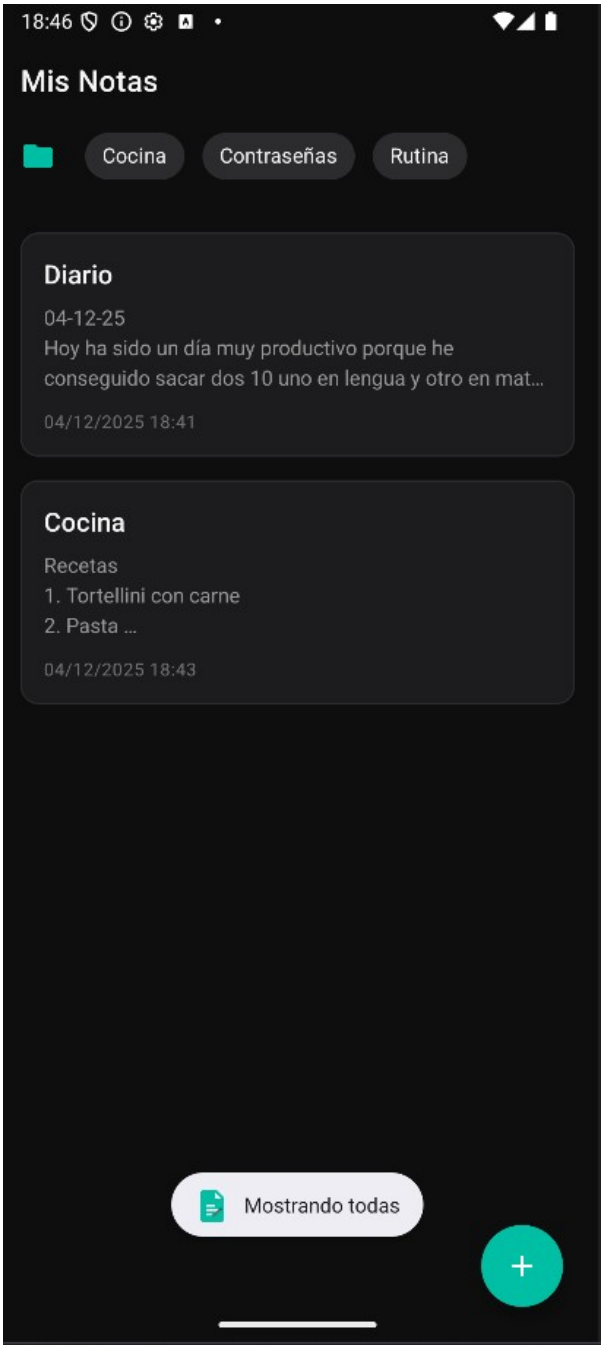
- titulo (TEXT): Título de la nota.
- contenido (TEXT): Cuerpo de la nota.
- fecha (TEXT): Fecha de última modificación.
- color (TEXT): Código de color para la nota.
- id_categoria (INTEGER FK): Clave foránea que referencia a categorias(id). *ON DELETE SET NULL*.

3.3. Estructura de Clases (Diagrama Lógico)

- **DBHelper**: Hereda de SQLiteOpenHelper. Encapsula todas las operaciones SQL (INSERT, UPDATE, DELETE, SELECT) y la gestión de versiones de la BBDD.
- **Nota y Categoria**: Data classes que representan los objetos del dominio. Nota puede contener una referencia a un objeto Categoria.
- **MainActivity**: Pantalla principal.
  - Contiene dos RecyclerView: uno horizontal para filtrar por categorías y uno vertical para listar las notas.
  - Gestiona el ActionMode (menú contextual) para borrar o mover múltiples notas seleccionadas.
- **NotaActivity**: Pantalla de detalle.
  - Permite crear una nota nueva o editar una existente.
  - Guarda automáticamente los cambios al pausar la actividad (onPause).
- **NuevaCategoriaActivity**: Pantalla de gestión de categorías.
  - Permite añadir categorías mediante un diálogo y listarlas/eliminarlas.

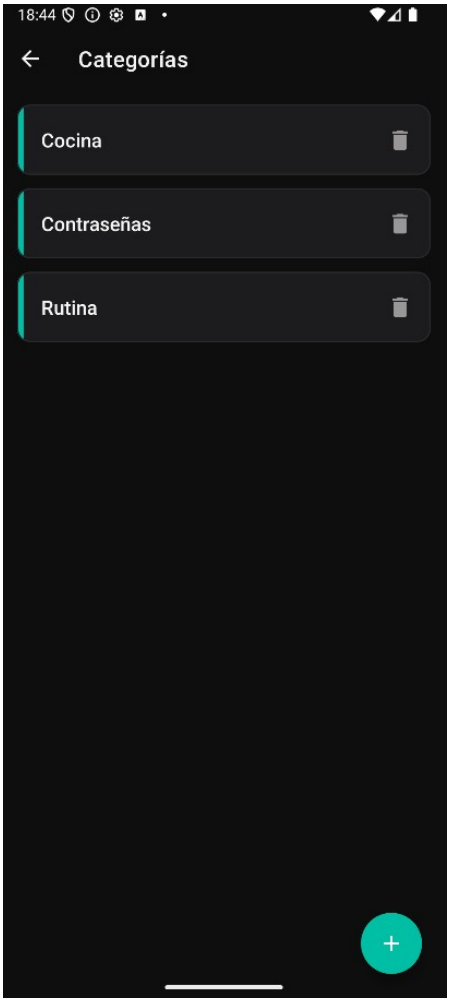
- **Adaptadores (NotaAdapter, CategoriaAdapterMain)**: Gestionan cómo se muestran los datos en las listas y detectan los clics del usuario.

3.4. Interfaz de Usuario (Mockups descriptivos)



1. **Pantalla Principal (MainActivity):** Muestra una barra superior con las categorías (scroll horizontal) y debajo la lista de notas. Incluye un botón flotante (FAB) para añadir notas y un icono de carpeta para gestionar categorías.

2. **Pantalla de Edición (NotaActivity):**
   Formulario simple con campos para Título y Contenido. Incluye una flecha de retorno en la barra superior.



3. **Categorias:** Pantalla que gestiona todas las categorías, mostrandolas, pudiendo borrar y añadir.

4. Pruebas Realizadas

Se han realizado las siguientes pruebas funcionales en el emulador de Android Studio para validar el correcto funcionamiento:

1. **Persistencia de Datos:**
    - *Prueba:* Crear una nota, cerrar la app completamente y volver a abrirla.
    - *Resultado:* La nota aparece correctamente listada (verificado mediante DBHelper.obtenerNotas()).
2. **Filtrado por Categoría:**
    - *Prueba:* Crear las categorías "Trabajo" y "Personal". Asignar notas a cada una y pulsar sobre los filtros.
    - *Resultado:* Al pulsar "Trabajo", solo se muestran las notas asociadas. Al volver a pulsar, se quita el filtro.
3. **Edición y Guardado Automático:**
    - *Prueba:* Modificar una nota existente y pulsar "Atrás".
    - *Resultado:* Los cambios se reflejan en la lista principal inmediatamente (onResume recarga la lista).

5. Despliegue en Terminal Real

Para el despliegue de la aplicación se han seguido los siguientes pasos:

1. **Configuración del Dispositivo:**
    - Activación de las "Opciones de desarrollador" en el terminal Android físico.
    - Habilitación de la "Depuración por USB".
2. **Generación del APK:**
    - En Android Studio, menú *Build > Build Bundle(s) / APK(s) > Build APK(s)*.
    - Se genera el archivo con la apk.
3. **Instalación:**
    - Lo compartimos por la nube con google drive y lo instalamos y comprobamos que funciona