

DSM

# MANUAL TÉCNICO

DESARROLLO DE  
SOFTWARE PARA MÓVILES



# ÍNDICE

01	DESCRIPCIÓN DEL PROYECTO
02	ESTRUCTURA DEL PROYECTO
03	CONFIGURACIÓN
04	EJECUTAR EL PROYECTO

05	MAINACTIVITY.KT
06	APPTHEME.KT
07	PBUILD.GRADLE.KTS
08	ANDROIDMANIFEST.XML
09	OTROS ARCHIVOS
10	UI COMPOSABLES
11	VIEWMODEL
12	MODELOS DE DATOS
13	REPOSITORIOS
14	PRUEBAS UNITARIAS
15	NAVEGACIÓN
16	DEPENDENCIAS
17	NOTAS DE DESARROLLO

# DESCRIPCIÓN GENERAL DEL PROYECTO

---

LA APLICACIÓN DE CONTROL DE GASTOS ES UNA HERRAMIENTA DISEÑADA PARA AYUDAR A LOS USUARIOS A GESTIONAR Y SUPERVISAR SUS FINANZAS PERSONALES. SU OBJETIVO ES FACILITAR EL SEGUIMIENTO DE INGRESOS Y GASTOS, PROPORCIONANDO UNA INTERFAZ CLARA Y FÁCIL DE USAR. LA APLICACIÓN UTILIZA JETPACK COMPOSE PARA CREAR INTERFACES MODERNAS Y REACTIVAS, LO QUE PERMITE UN DESARROLLO MÁS EFICIENTE Y UN RENDIMIENTO OPTIMIZADO. LOS USUARIOS PUEDEN AGREGAR, EDITAR Y ELIMINAR GASTOS, Y LA APLICACIÓN CATEGORIZA AUTOMÁTICAMENTE LOS GASTOS PARA FACILITAR SU ANÁLISIS.

# ESTRUCTURA DEL PROYECTO

---

LA ESTRUCTURA DE ARCHIVOS DEL PROYECTO SIGUE LA CONVENCION DE ANDROID Y KOTLIN MULTIPLATFORM, LO QUE PERMITE LA REUTILIZACION DEL CODIGO Y LA SEPARACION DE LAS PREOCUPACIONES. AQUÍ SE DESCRIBE LA JERARQUÍA BÁSICA DE CARPETAS Y ARCHIVOS:

ORG.EXAMPLE.PROJECT

```
|—— /SRC
|   |—— /ANDROIDMAIN
|   |   |—— MAINACTIVITY.KT
|   |   |—— ANDROIDMANIFEST.XML
|   |   |—— RES/ (ARCHIVOS DE RECURSOS)
|   |—— /COMMONMAIN
|       |—— APPTHEME.KT
|       |—— (UI COMPOSABLES)
|       |—— VIEWMODEL
|       |—— MODELOS DE DATOS
|       |—— REPOSITORIOS
|—— BUILD.GRADLE.KTS
```

# CONFIGURACIÓN DEL ENTORNO DE DESARROLLO

---

1. **INSTALAR ANDROID STUDIO: ASEGÚRATE DE TENER LA ÚLTIMA VERSIÓN DE ANDROID STUDIO INSTALADA.**
2. **CONFIGURAR KOTLIN MULTIPLATFORM: ASEGÚRATE DE TENER CONFIGURADAS LAS HERRAMIENTAS NECESARIAS PARA TRABAJAR CON KOTLIN MULTIPLATFORM.**
3. **CLONAR EL REPOSITORIO: CLONA EL REPOSITORIO DONDE SE ENCUENTRA EL PROYECTO.**
4. **SINCRONIZAR EL PROYECTO: ABRE EL PROYECTO EN ANDROID STUDIO Y SINCRONIZA LAS DEPENDENCIAS CON GRADLE.**

# EJECUTAR EL PROYECTO

---

1. SELECCIONAR EL DISPOSITIVO: ASEGÚRATE DE QUE UN DISPOSITIVO FÍSICO O UN EMULADOR ESTÉ CONFIGURADO Y LISTO PARA USAR.
2. EJECUTAR LA APLICACIÓN: HAZ CLIC EN EL BOTÓN "RUN" EN ANDROID STUDIO PARA COMPILAR Y EJECUTAR LA APLICACIÓN EN EL DISPOSITIVO SELECCIONADO.

# MAINACTIVITY.KT

---

UBICACIÓN: `SRC/ANDROIDMAIN/MAINACTIVITY.KT`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** ESTE ARCHIVO DEFINE LA CLASE `MAINACTIVITY`, QUE ES LA ENTRADA PRINCIPAL DE LA APLICACIÓN ANDROID. EXTIENDE DE `COMPONENTACTIVITY`, LO QUE LE PERMITE UTILIZAR LAS CARACTERÍSTICAS DE JETPACK COMPOSE. DENTRO DE ESTA CLASE, SE CONFIGURA EL `SETCONTENT`, DONDE SE ESTABLECE EL CONTENIDO DE LA INTERFAZ DE USUARIO DE LA APLICACIÓN.

- **COMPONENTES CLAVE:**

- **`SETCONTENT`:** ESTE MÉTODO ESTABLECE LA INTERFAZ DE USUARIO UTILIZANDO COMPOSABLES DEFINIDOS EN EL PROYECTO.

- **INTEGRACIÓN CON VIEWMODEL:** LA ACTIVIDAD PUEDE OBSERVAR LOS CAMBIOS EN EL VIEWMODEL Y ACTUALIZAR LA INTERFAZ DE USUARIO EN CONSECUENCIA.

# APPTHEME.KT

---

UBICACIÓN: `SRC/COMMONMAIN/APPTHEME.KT`

DESCRIPCIÓN:

- FUNCIONALIDAD: ESTE ARCHIVO DEFINE EL TEMA DE LA APLICACIÓN UTILIZANDO `MATERIALTHEME`, LO QUE PERMITE PERSONALIZAR LA APARIENCIA DE LA INTERFAZ DE USUARIO. AQUÍ SE PUEDEN MODIFICAR LOS COLORES, LAS FORMAS Y OTROS ATRIBUTOS ESTÉTICOS.

- COMPONENTES CLAVE:

- `MATERIALTHEME`: SE UTILIZA PARA APLICAR EL TEMA A LOS COMPONENTES DE LA INTERFAZ DE USUARIO.

- COLORES PERSONALIZADOS: SE DEFINEN DIFERENTES COLORES PARA EL MODO OSCURO Y CLARO, PERMITIENDO QUE LA APLICACIÓN SE ADAPTE A LAS PREFERENCIAS DEL USUARIO.



# BUILD.GRADLE.KTS

---

UBICACIÓN: `BUILD.GRADLE.KTS`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** ESTE ARCHIVO ES UN SCRIPT DE CONFIGURACIÓN DE GRADLE QUE DEFINE CÓMO SE CONSTRUYE EL PROYECTO. CONTIENE INFORMACIÓN SOBRE LAS DEPENDENCIAS, LOS PLUGINS UTILIZADOS Y LAS CONFIGURACIONES ESPECÍFICAS PARA EL ENTORNO ANDROID.

- **COMPONENTES CLAVE:**

- **PLUGINS:** SE INCLUYEN PLUGINS COMO KOTLIN MULTIPLATFORM Y JETPACK COMPOSE.

- **DEPENDENCIAS:** SE DECLARAN LAS BIBLIOTECAS NECESARIAS PARA EL FUNCIONAMIENTO DE LA APLICACIÓN, COMO COMPOSE, ROOM, Y OTRAS.

- **CONFIGURACIÓN DEL ANDROID SDK:** SE ESPECIFICAN LAS VERSIONES DEL SDK A UTILIZAR EN LA COMPILACIÓN.

# ANDROIDMANIFEST.XML

---

UBICACIÓN: `SRC/ANDROIDMAIN/ANDROIDMANIFEST.XML`

DESCRIPCIÓN:

– FUNCIONALIDAD: ESTE ARCHIVO ES ESENCIAL PARA CUALQUIER APLICACIÓN DE ANDROID. DECLARA TODOS LOS COMPONENTES DE LA APLICACIÓN, COMO ACTIVIDADES, SERVICIOS Y RECEPTORES DE CONTENIDO. TAMBIÉN GESTIONA LOS PERMISOS NECESARIOS PARA ACCEDER A FUNCIONES ESPECÍFICAS DEL DISPOSITIVO, COMO LA RED O EL ALMACENAMIENTO.

– COMPONENTES CLAVE:

– DECLARACIÓN DE LA ACTIVIDAD PRINCIPAL: INDICA CUÁL ES LA ACTIVIDAD DE INICIO.

– PERMISOS: SE ESPECIFICAN LOS PERMISOS NECESARIOS PARA ACCEDER A LA RED, ALMACENAMIENTO, ETC.

# OTROS ARCHIVOS DE RECURSOS

---

UBICACIÓN: `SRC/ANDROIDMAIN/RES/`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** ESTA CARPETA CONTIENE TODOS LOS RECURSOS NECESARIOS PARA LA INTERFAZ DE USUARIO DE LA APLICACIÓN. INCLUYE IMÁGENES, ARCHIVOS DE DISEÑO, CADENAS DE TEXTO Y OTROS RECURSOS.

- **COMPONENTES CLAVE:**

- **ARCHIVOS DE DISEÑO XML:** DEFINEN LA ESTRUCTURA DE LAS PANTALLAS Y SE UTILIZAN JUNTO CON JETPACK COMPOSE PARA CREAR LA INTERFAZ.

- **IMÁGENES Y OTROS RECURSOS MULTIMEDIA:** UTILIZADOS PARA ENRIQUECER LA EXPERIENCIA VISUAL DEL USUARIO.

# UI COMPOSABLES

---

UBICACIÓN: `SRC/COMMONMAIN/`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** ESTA CARPETA INCLUYE LAS FUNCIONES COMPOSABLES QUE REPRESENTAN LOS DIFERENTES COMPONENTES DE LA INTERFAZ DE USUARIO. ESTOS COMPONENTES SE UTILIZAN PARA CONSTRUIR LA UI DE MANERA DECLARATIVA, PERMITIENDO UNA SEPARACIÓN CLARA ENTRE LA LÓGICA DE PRESENTACIÓN Y LA LÓGICA DE NEGOCIO.
- **COMPONENTES CLAVE:**
  - **COMPONENTES REUTILIZABLES:** SE CREAN ELEMENTOS COMO BOTONES, LISTAS Y FORMULARIOS QUE SE PUEDEN UTILIZAR EN DIFERENTES PANTALLAS.
  - **INTERACTIVIDAD:** LAS COMPOSABLES PUEDEN MANEJAR EVENTOS, COMO CLICS Y CAMBIOS DE ESTADO.

# VIEWMODEL

---

UBICACIÓN: `SRC/COMMONMAIN/`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** LOS ARCHIVOS RELACIONADOS CON EL VIEWMODEL GESTIONAN LA LÓGICA DE LA APLICACIÓN Y MANTIENEN EL ESTADO DE LA INTERFAZ DE USUARIO. EL VIEWMODEL PERMITE QUE LOS DATOS SOBREVIVAN A CAMBIOS DE CONFIGURACIÓN, COMO ROTACIONES DE PANTALLA.

- **COMPONENTES CLAVE:**

- **LIVEDATA Y STATEFLOW:** SE UTILIZAN PARA OBSERVAR LOS CAMBIOS EN LOS DATOS Y NOTIFICAR A LA INTERFAZ DE USUARIO CUANDO SE PRODUCEN CAMBIOS.

- **MÉTODOS PARA GESTIONAR LA LÓGICA DE NEGOCIO:** INCLUYEN OPERACIONES COMO AGREGAR, EDITAR Y ELIMINAR GASTOS.

# MODELOS DE DATOS

---

UBICACIÓN: `SRC/COMMONMAIN/`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** ESTE ARCHIVO DEFINE LAS CLASES QUE REPRESENTAN LOS DATOS DE LA APLICACIÓN, COMO LOS GASTOS Y CATEGORÍAS. LOS MODELOS DE DATOS SON ESENCIALES PARA LA GESTIÓN Y MANIPULACIÓN DE LA INFORMACIÓN EN LA APLICACIÓN.

- **COMPONENTES CLAVE:**

- **CLASES DE DATOS:** DEFINEN LAS PROPIEDADES Y MÉTODOS RELACIONADOS CON LOS GASTOS, INCLUYENDO SUS ATRIBUTOS COMO MONTO, CATEGORÍA Y FECHA.

- **MÉTODOS DE SERIALIZACIÓN:** FACILITAN LA CONVERSIÓN DE LOS DATOS A FORMATOS QUE SE PUEDEN ALMACENAR O ENVIAR A TRAVÉS DE LA RED.

# REPOSITARIOS

---

UBICACIÓN: `SRC/COMMONMAIN/`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** LOS REPOSITARIOS GESTIONAN EL ACCESO A LOS DATOS, PROPORCIONANDO UNA INTERFAZ PARA QUE EL VIEWMODEL PUEDA OBTENER Y ALMACENAR INFORMACIÓN. SE ENCARGAN DE LA LÓGICA DE ACCESO A DATOS, YA SEA DESDE UNA BASE DE DATOS LOCAL, UNA API O CUALQUIER OTRA FUENTE.

- **COMPONENTES CLAVE:**

- **MÉTODOS PARA OBTENER DATOS:** PROPORCIONAN FUNCIONES PARA ACCEDER A LOS GASTOS Y OTROS DATOS ALMACENADOS.

- **GESTIÓN DE FUENTES DE DATOS:** LOS REPOSITARIOS PUEDEN DECIDIR DE DÓNDE OBTENER LOS DATOS, YA SEA DE UNA BASE DE DATOS LOCAL (COMO ROOM) O DE UN SERVICIO WEB.

# PRUEBAS UNITARIAS

---

UBICACIÓN: `SRC/TEST/`

DESCRIPCIÓN:

- FUNCIONALIDAD: ESTA CARPETA CONTIENE PRUEBAS UNITARIAS QUE VERIFICAN QUE LA LÓGICA DE NEGOCIO Y LAS FUNCIONALIDADES DE LA APLICACIÓN FUNCIONEN CORRECTAMENTE. LAS PRUEBAS AYUDAN A IDENTIFICAR ERRORES Y A ASEGURAR QUE LAS FUTURAS MODIFICACIONES NO ROMPAN EL FUNCIONAMIENTO EXISTENTE.

- COMPONENTES CLAVE:

- PRUEBAS PARA VIEWMODEL: VERIFICAN QUE LOS MÉTODOS DEL VIEWMODEL DEVUELVAN LOS RESULTADOS ESPERADOS.

- PRUEBAS PARA REPOSITORIOS: ASEGURAN QUE LOS MÉTODOS DE ACCESO A DATOS FUNCIONEN CORRECTAMENTE.



# NAVEGACIÓN

---

SRC/CommonMain/`

DESCRIPCIÓN:

- **FUNCIONALIDAD:** LA NAVEGACIÓN EN LA APLICACIÓN PERMITE A LOS USUARIOS MOVERSE ENTRE DIFERENTES PANTALLAS Y SE IMPLEMENTA UTILIZANDO LA API DE NAVEGACIÓN DE JETPACK COMPOSE.

- **COMPONENTES CLAVE:**

- **GRÁFICOS DE NAVEGACIÓN:** DEFINEN CÓMO SE CONECTAN LAS DIFERENTES PANTALLAS.

- **MANEJO DE PARÁMETROS:** PERMITEN PASAR DATOS ENTRE LAS PANTALLAS DURANTE LA NAVEGACIÓN.

# DEPENDENCIAS

---

UBICACIÓN: ``BUILD.GRADLE.KTS``

DESCRIPCIÓN:

- FUNCIONALIDAD: LAS DEPENDENCIAS SON BIBLIOTECAS DE TERCEROS QUE SE UTILIZAN PARA EXTENDER LA FUNCIONALIDAD DE LA APLICACIÓN. EN EL ARCHIVO ``BUILD.GRADLE.KTS``, SE DECLARAN LAS DEPENDENCIAS QUE SE UTILIZAN EN LA APLICACIÓN.

- COMPONENTES CLAVE:

- JETPACK COMPOSE: SE UTILIZA PARA CONSTRUIR LA INTERFAZ DE USUARIO DE MANERA DECLARATIVA.

- ROOM: SE USA PARA LA GESTIÓN DE LA BASE DE DATOS LOCAL, FACILITANDO EL ALMACENAMIENTO Y RECUPERACIÓN DE DATOS.

# NOTAS DE DESARROLLO

---

UBICACIÓN: `SRC/COMMONMAIN/`

DESCRIPCIÓN:

- FUNCIONALIDAD: ESTA SECCIÓN PUEDE INCLUIR CUALQUIER NOTA IMPORTANTE RELACIONADA CON EL DESARROLLO DEL PROYECTO, COMO DECISIONES ARQUITECTÓNICAS, PROBLEMAS ENCONTRADOS DURANTE EL DESARROLLO O CARACTERÍSTICAS PLANIFICADAS PARA FUTURAS VERSIONES.

- COMPONENTES CLAVE:

- COMENTARIOS SOBRE DECISIONES DE DISEÑO: EXPLICACIONES SOBRE POR QUÉ SE ELIGIERON CIERTAS BIBLIOTECAS O PATRONES DE DISEÑO.

- TAREAS PENDIENTES: LISTAS DE CARACTERÍSTICAS QUE SE DEBEN IMPLEMENTAR O PROBLEMAS QUE DEBEN RESOLVERSE.