

MAQUETACIÓN DE LA INTERFAZ GRÁFICA EN XML – ANDROID

Realizado por:

ANTONIO MUÑOZ

Ficha:

2977449

Presentado a:

ADALBERTO CARCAMO ALVARADO

Servicio de Aprendizaje Sena

13 de marzo de 2025

Introducción

El presente documento detalla el proceso de desarrollo, análisis y diseño de "MiGestorDeGastos", una aplicación móvil orientada a la gestión financiera personal, creada utilizando la plataforma MIT App Inventor. Este trabajo pone especial énfasis en la **maquetación de la interfaz gráfica mediante XML** como componente fundamental del proceso de desarrollo para aplicaciones Android.

La separación entre presentación y lógica representa uno de los principios fundamentales en el desarrollo de software moderno, particularmente en entornos móviles donde la experiencia de usuario constituye un factor crítico para el éxito de una aplicación. Esta separación, implementada a través de la maqueta XML, no solo facilita el mantenimiento y la escalabilidad del código, sino que también permite asegurar el cumplimiento de estándares de usabilidad y accesibilidad ampliamente reconocidos en la industria.

Este documento abordará minuciosamente el diseño de cada componente de la interfaz de usuario, explicará la arquitectura de navegación implementada, y proporcionará evidencia visual del funcionamiento de la aplicación en diferentes estados y contextos de uso.

Tipo de Aplicación y Perspectivas del Producto

Tipo de Aplicación

"MiGestorDeGastos" es una **APLICACIÓN NATIVA** desarrollada específicamente para la plataforma Android utilizando MIT App Inventor como entorno de desarrollo. Esta elección responde a la necesidad de garantizar un rendimiento óptimo y una integración fluida con las funcionalidades nativas del sistema operativo Android, proporcionando así una experiencia de usuario consistente y eficiente.

Perspectivas del Producto

"MiGestorDeGastos" se posiciona como una solución integral para el seguimiento y administración de finanzas personales, orientada a usuarios con diferentes niveles de alfabetización tecnológica. El producto destaca por:

- **Interfaz minimalista y funcional:** Diseñada bajo principios de simplicidad visual que elimina distracciones y facilita la comprensión inmediata de la información financiera.
- **Curva de aprendizaje reducida:** La aplicación prioriza la intuitividad en cada interacción, permitiendo que incluso usuarios sin experiencia previa en aplicaciones financieras puedan utilizarla eficazmente desde el primer contacto.
- **Enfoque en datos esenciales:** Centrada en proporcionar exactamente la información necesaria para la toma de decisiones financieras, evitando la sobrecarga

cognitiva mediante una cuidadosa selección y presentación de los datos más relevantes.

- **Transparencia informativa:** Todos los procesos y cálculos se presentan de manera clara, permitiendo al usuario comprender fácilmente cómo se derivan los resultados financieros mostrados.

3. Funciones del Producto

"MiGestorDeGastos" implementa un conjunto cuidadosamente seleccionado de funcionalidades esenciales para la gestión financiera personal. Estas funciones han sido diseñadas siguiendo un enfoque centrado en el usuario, priorizando aquellas capacidades que proporcionan mayor valor práctico en el contexto del control de gastos diarios:

1. **Registro de Gastos con Validación Integrada:**
 - Sistema de entrada de datos con validación en tiempo real
 - Campos optimizados para captura de montos, descripciones y categorización
 - Retroalimentación visual inmediata sobre la validez de los datos ingresados
2. **Sistema de Categorización Jerárquica:**
 - Categorías predefinidas basadas en patrones comunes de gasto (Alimentación, Transporte, Entretenimiento, Servicios, etc.)
 - Estructura jerárquica que facilita el análisis detallado de patrones de gasto
3. **Visualización de Datos Financieros:**
 - Interfaz de resumen que ofrece una visión consolidada por categorías
 - Representación clara del total acumulado y distribución porcentual
 - Codificación cromática para facilitar la identificación visual de categorías
4. **Persistencia de Datos mediante TinyDB:**
 - Implementación optimizada del sistema de almacenamiento local TinyDB
 - Estructura de datos normalizada que garantiza la integridad de la información
 - Mecanismos de respaldo para prevenir pérdida accidental de datos
5. **Sistema de Consulta Detallada:**
 - Interfaz específica para acceder al historial completo de transacciones
 - Capacidades de filtrado y ordenamiento para facilitar el análisis
 - Visualización cronológica de la evolución de gastos
6. **Arquitectura de Navegación Intuitiva:**
 - Diseño de flujos de interacción optimizados para minimizar la carga cognitiva
 - Estructura jerárquica que refleja modelos mentales comunes sobre gestión financiera
 - Mecanismos consistentes de retorno a estados anteriores

Cada una de estas funcionalidades ha sido implementada considerando principios de diseño centrado en el usuario, garantizando que la aplicación no solo cumpla con su propósito funcional, sino que lo haga de manera que resulte natural e intuitiva para el usuario final.

4. Requerimientos Funcionales y No Funcionales

Requerimientos Funcionales

Los siguientes requerimientos funcionales han sido documentados siguiendo la metodología de identificación basada en prefijos "APP-", facilitando su trazabilidad a lo largo del ciclo de desarrollo:

- **APP-2501: Inicialización y Carga de Datos**
 - *Descripción:* Al iniciar, la aplicación debe realizar la carga automática de la interfaz principal y ejecutar una consulta a TinyDB para recuperar el historial de gastos previos.
 - *Comportamiento esperado:* En caso de existir datos almacenados, estos deben visualizarse inmediatamente. En ausencia de datos previos, la aplicación presentará una interfaz inicial con campos vacíos, listos para el ingreso de nueva información.
 - *Criterios de aceptación:* Tiempo de carga inferior a 2 segundos, visualización correcta de datos históricos cuando existan, presentación apropiada de la interfaz inicial para nuevos usuarios.
- **APP-2502: Registro y Validación de Gastos**
 - *Descripción:* Proporcionar mecanismos para el ingreso estructurado de gastos, incluyendo cantidad, descripción detallada y categorización.
 - *Comportamiento esperado:* Todos los datos ingresados deben someterse a un proceso de validación que verifique: valores numéricos positivos para cantidades, presencia de descripción informativa, y selección obligatoria de categoría.
 - *Criterios de aceptación:* Retroalimentación visual inmediata sobre errores de validación, persistencia exitosa de datos validados, actualización automática de visualizaciones tras registro exitoso.
- **APP-2503: Visualización Analítica de Gastos**
 - *Descripción:* Implementar una visualización consolidada que presente datos agregados por categoría y calcule automáticamente totales acumulados.
 - *Comportamiento esperado:* La información debe organizarse jerárquicamente por categorías, mostrando subtotales por cada una y un cálculo global del gasto total.
 - *Criterios de aceptación:* Precisión absoluta en cálculos agregados, actualización automática tras nuevos registros, diferenciación visual clara entre categorías.
- **APP-2504: Sistema de Categorización Dinámica**
 - *Descripción:* Proporcionar un mecanismo de selección de categorías predefinidas a través de una interfaz de lista desplegable.
 - *Comportamiento esperado:* El usuario debe poder seleccionar de un conjunto de categorías predefinidas que representen los rubros más comunes de gasto personal.
 - *Criterios de aceptación:* Lista desplegable funcional con todas las categorías establecidas, selección por defecto informativa, persistencia de la categorización en el almacenamiento.
- **APP-2505: Visualización Detallada del Historial**

- *Descripción:* Implementar una vista especializada que permita acceder al detalle completo de cada transacción registrada.
- *Comportamiento esperado:* Presentación cronológica del historial de gastos con toda la información asociada a cada registro: fecha, descripción, monto y categoría.
- *Criterios de aceptación:* Visualización correcta de todos los campos asociados a cada gasto, navegación fluida entre la vista resumida y la detallada, capacidad de retorno a la pantalla principal.
- **APP-2506: Persistencia y Gestión de Datos**
 - *Descripción:* Implementar un sistema robusto de almacenamiento persistente utilizando TinyDB como motor de base de datos local.
 - *Comportamiento esperado:* Cada operación de registro o modificación debe persistirse inmediatamente, garantizando la conservación de datos incluso tras cierres no programados de la aplicación.
 - *Criterios de aceptación:* Recuperación exitosa de datos tras reinicio, integridad de la información almacenada, optimización del almacenamiento para evitar duplicidades o inconsistencias.

Requerimientos No Funcionales

Los requerimientos no funcionales se han estructurado en cuatro categorías principales que establecen los atributos de calidad y restricciones técnicas del sistema:

1. **Usabilidad y Experiencia de Usuario**
 - Implementación de una interfaz intuitiva con disposición lógica de elementos y flujos de navegación naturales.
 - Sistema integral de retroalimentación visual que confirma cada acción del usuario (confirmaciones, alertas, indicadores de progreso).
 - Diseño accesible con elementos táctiles de tamaño adecuado (mínimo 48dp para elementos interactivos) y tipografía legible (14sp mínimo para texto informativo).
 - Consistencia visual en toda la aplicación, manteniendo patrones de color, tipografía y disposición espacial.
2. **Rendimiento y Eficiencia**
 - Tiempo de respuesta inferior a 1 segundo para operaciones regulares y máximo 2 segundos para operaciones complejas con feedback visual durante la espera.
 - Arranque completo de la aplicación en menos de 3 segundos en dispositivos de gama media.
 - Uso eficiente de recursos del sistema, con un consumo de memoria RAM inferior a 50MB en operación normal.
 - Optimización del almacenamiento local, manteniendo un tamaño de base de datos inferior a 5MB incluso con históricos extensos.
3. **Fiabilidad, Seguridad y Protección de Datos**
 - Implementación de validación exhaustiva de entradas para prevenir errores de formato o inconsistencias.

- Mecanismos de respaldo automático que protejan contra pérdida accidental de datos.
- Encriptación básica de la información financiera almacenada localmente.
- Tolerancia a interrupciones, con capacidad de recuperación y preservación de datos en caso de cierres no programados.

4. **Mantenibilidad, Compatibilidad y Escalabilidad**

- Arquitectura modular con clara separación entre presentación (XML) y lógica de negocio.
- Comentarios explicativos en componentes clave para facilitar el mantenimiento.
- Compatibilidad garantizada con Android 5.0 (API nivel 21) y versiones superiores.
- Diseño responsivo que adapta la interfaz a diferentes tamaños y orientaciones de pantalla.
- Estructura flexible que permita la incorporación futura de nuevas funcionalidades sin necesidad de reescritura significativa.

5. Maquetación de la Interfaz Gráfica en XML

La maquetación de la interfaz gráfica mediante XML representa un componente fundamental en el desarrollo de aplicaciones Android, permitiendo una clara separación entre la presentación visual y la lógica funcional. Esta metodología ofrece múltiples ventajas:

- **Separación de responsabilidades:** La definición declarativa de la interfaz facilita el trabajo paralelo entre diseñadores y desarrolladores.
- **Mantenibilidad mejorada:** Las modificaciones visuales pueden realizarse sin afectar la lógica subyacente.
- **Consistencia visual:** Los elementos definidos de manera declarativa mantienen su apariencia en diferentes contextos.
- **Rendimiento optimizado:** El sistema Android puede optimizar la renderización de interfaces definidas en XML.

A continuación, se presenta el código XML fundamental utilizado para la maquetación de la pantalla principal de "MiGestorDeGastos":

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- Encabezado de la App con elevación para efecto material design -->

    <TextView
        android:id="@+id/tvTitulo"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="MiGestorDeGastos"
    android:textSize="24sp"
    android:textStyle="bold"
    android:padding="12dp"
    android:background="#f57c00"
    android:textColor="#ffffff"
    android:elevation="4dp" />
```

<!-- Sección de Formulario con CardView para efecto material -->

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:elevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Registrar Nuevo Gasto"
            android:textSize="18sp"
            android:textStyle="bold"
            android:layout_marginBottom="12dp" />

        <!-- Campo para ingresar la cantidad con validación -->
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Cantidad">

            <EditText
                android:id="@+id/etCantidad"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="numberDecimal"
                android:padding="12dp" />
        </com.google.android.material.textfield.TextInputLayout>

        <!-- Campo para ingresar la descripción -->
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Descripción"
            android:layout_marginTop="8dp">

            <EditText
                android:id="@+id/etDescripcion"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="text"
```

```

        android:padding="12dp" />
</com.google.android.material.textfield.TextInputLayout>

<!-- Lista desplegable para seleccionar la categoría -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Categoría"
    android:layout_marginTop="12dp"
    android:layout_marginBottom="4dp" />

<Spinner
    android:id="@+id/spnCategoria"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/btn_dropdown"
    android:padding="12dp" />

<!-- Botón para guardar con estilo material design -->
<Button
    android:id="@+id/btnGuardar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="GUARDAR"
    android:textColor="#ffffff"
    android:background="#4caf50"
    android:padding="12dp"
    android:layout_marginTop="16dp"
    android:elevation="2dp" />
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>

```

Este fragmento XML ilustra la implementación de una interfaz siguiendo los principios de Material Design, incorporando elementos visuales como elevación, tarjetas (CardView) y campos de entrada con validación mediante TextInputLayout. La estructura jerárquica define claramente la disposición de los elementos y sus propiedades visuales.

Maquetación Avanzada y Compatibilidad

Para garantizar que la interfaz se visualice correctamente en diferentes dispositivos y versiones de Android, se han implementado las siguientes estrategias de maquetación avanzada:

1. **Uso de dimensiones relativas:** Empleando "match_parent" y "wrap_content" en lugar de valores fijos en píxeles para permitir la adaptabilidad a diferentes tamaños de pantalla.
2. **Sistema de padding y márgenes consistente:** Aplicación de espaciado uniforme (múltiplos de 8dp según las guías de Material Design) para mantener la coherencia visual.

3. **Compatibilidad con versiones anteriores:** Utilización de la biblioteca de soporte (AndroidX) para garantizar que los componentes modernos funcionen correctamente en versiones antiguas del sistema operativo.
4. **Optimización para diferentes densidades de pantalla:** Definición de recursos gráficos alternativos para diferentes resoluciones (mdpi, hdpi, xhdpi, xxhdpi) cuando sea necesario.

6. Diseño de la Interfaz y Mapa de Navegación

Arquitectura Visual

La aplicación "MiGestorDeGastos" implementa una arquitectura visual estructurada en torno a tres pantallas principales, cada una con un propósito específico dentro del flujo de interacción:

1. Pantalla Principal (Home)

Esta pantalla constituye el punto de entrada y el centro operativo de la aplicación, dividida en tres secciones funcionales claramente diferenciadas:

- **Sección Superior (Header):**
 - Encabezado prominente con el nombre de la aplicación
 - Mensaje de bienvenida personalizado
 - Indicador visual del periodo activo de registro
- **Sección de Registro (Input Section):**
 - Formulario estructurado para entrada de datos financieros
 - Campos optimizados para diferentes tipos de información (numérica, textual, categórica)
 - Sistema de validación visual en tiempo real
 - Botón de acción prominente para confirmar el registro
- **Sección de Resumen (Summary Section):**
 - Tabla interactiva con categorización jerárquica de gastos
 - Visualización de montos acumulados por categoría
 - Cálculo y presentación del total global
 - Control de navegación hacia la vista detallada

2. Pantalla de Detalle de Gastos (Detail View)

Esta pantalla ofrece una visualización exhaustiva del historial financiero:

- Lista cronológica invertida (del más reciente al más antiguo)
- Presentación completa de cada registro con todos sus atributos:
 - Fecha y hora de registro
 - Descripción textual completa
 - Valor monetario con formato localizado
 - Categoría asociada con indicador visual

- Mecanismo de retorno directo a la pantalla principal

3. Pantalla de Configuración (Settings)

Aunque no implementada en la versión actual, esta pantalla está contemplada en la arquitectura para expansiones futuras:

- Personalización de categorías
- Ajustes de visualización y preferencias de usuario
- Configuración de alertas y recordatorios
- Opciones de exportación e importación de datos

Paleta Cromática y Elementos Visuales

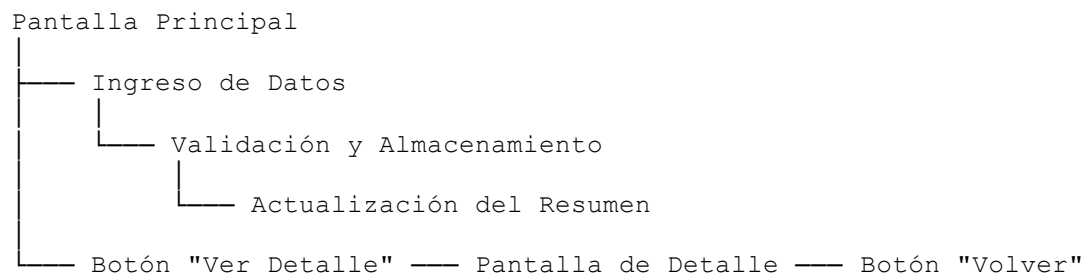
La aplicación implementa una paleta de colores cuidadosamente seleccionada para optimizar la usabilidad y crear una identidad visual coherente:

- **Color Primario:** Naranja (#f57c00) aplicado en encabezados y elementos destacados
- **Color de Acción:** Verde (#4caf50) utilizado en botones de confirmación y valores positivos
- **Color de Navegación:** Azul (#2196f3) para elementos de navegación y enlaces
- **Colores de Fondo:** Combinación de blanco (#ffffff) y gris claro (#eeeeee) para áreas de contenido
- **Colores de Texto:** Negro para texto principal (#212121) y gris oscuro (#757575) para información secundaria

Esta paleta cumple con los requisitos de contraste establecidos por las directrices WCAG 2.1 AA, garantizando la legibilidad para usuarios con diferentes capacidades visuales.

Mapa de Navegación

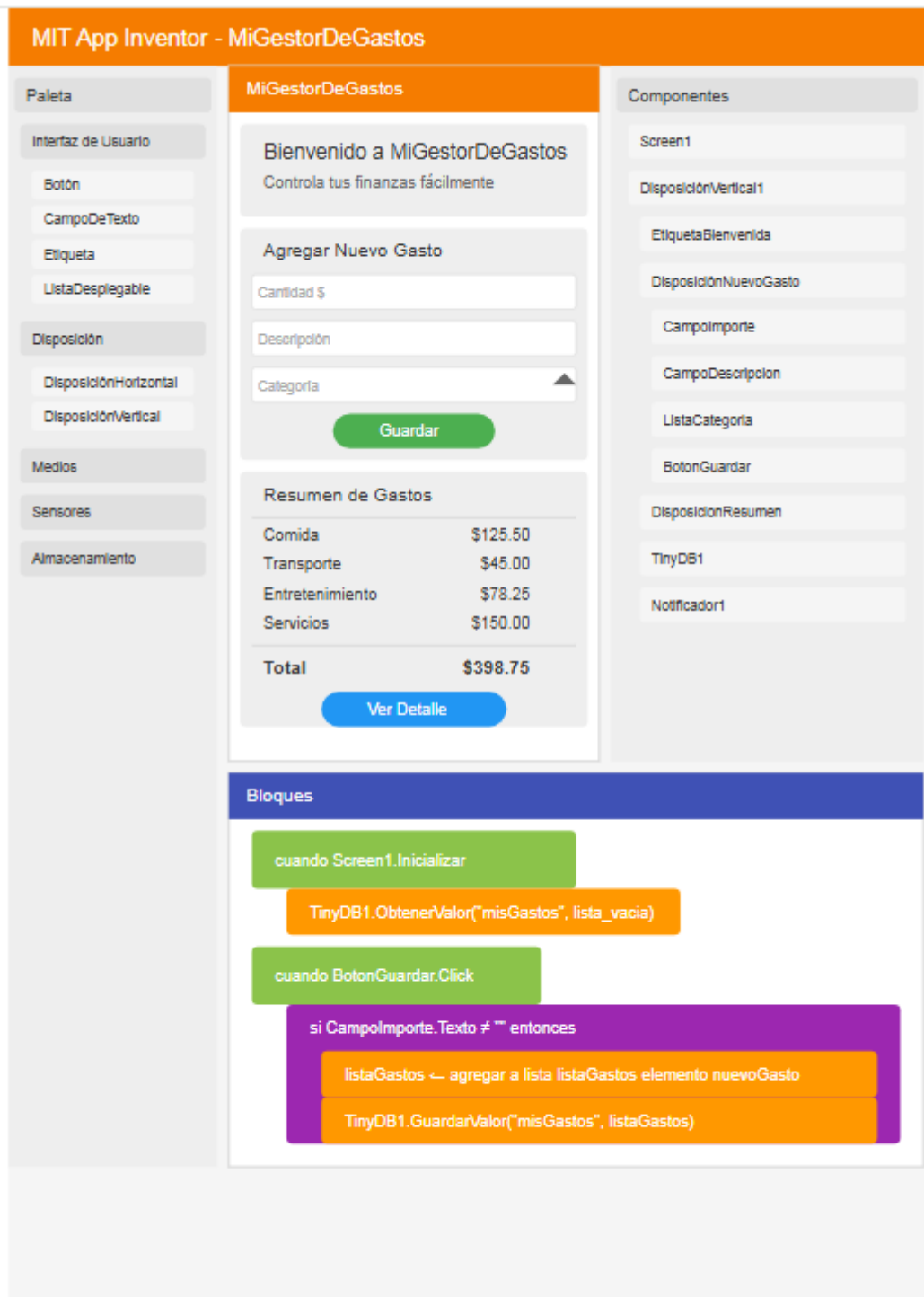
El flujo de navegación de la aplicación se ha diseñado siguiendo principios de simplicidad y previsibilidad, minimizando la profundidad jerárquica y ofreciendo rutas claras de retorno:



Este esquema de navegación implementa:

- **Navegación Plana:** Arquitectura de poca profundidad que evita que el usuario se "pierda" en múltiples niveles.
- **Consistencia en los Mecanismos de Retorno:** Botones de "Volver" ubicados consistentemente en la misma posición.
- **Persistencia de Estado:** Mantenimiento del contexto y los datos entre transiciones de pantalla.
- **Retroalimentación Visual durante las Transiciones:** Indicadores de carga o progreso cuando sea necesario.

Implementación y Visualización



Implementación de Pantallas

La implementación visual de la aplicación "MiGestorDeGastos" se materializa a través de múltiples pantallas XML cuidadosamente diseñadas. A continuación, se presentan fragmentos de código XML adicionales para las pantallas de Resumen y Detalle, junto con su descripción visual correspondiente:

Pantalla de Resumen de Gastos

Resumen de Gastos

Categoría	Monto
Alimentación	\$1,250.00
Transporte	\$850.00
Entretenimiento	\$500.00

Ver Detalle

Pantalla de Detalle de Gastos

Detalle de Gastos

Compra supermercado Alimentación	\$450.00
Cine fin de semana Entretenimiento	\$200.00
Gasolina Transporte	\$350.00
Comida restaurante Alimentación	\$350.00

Volver