

Documento de Arquitectura

Aplicación de Cálculos Financieros Empresariales

1. Introducción

Propósito:

Este documento describe la arquitectura de la aplicación móvil de cálculos financieros, incluyendo la estructura, componentes principales, patrones de diseño aplicados y las decisiones técnicas tomadas.

Objetivo:

Desarrollar una aplicación móvil de una sola pantalla que permita realizar cálculos financieros específicos para el contexto colombiano. La aplicación debe ser intuitiva y permitir al usuario seleccionar entre diferentes categorías de cálculos.

2. Descripción General del Sistema

La aplicación permite a los usuarios realizar cálculos financieros categorizados en:

1. **Cálculos de Productos:** Precio con IVA, margen de ganancia, punto de equilibrio, ROI.
2. **Cálculos de Empleador:** Costos de nómina, aportes parafiscales, seguridad social, prestaciones sociales.
3. **Cálculos de Empleado:** Salario neto, deducciones, horas extras, bonificaciones.

Plataforma: Android

Lenguaje: Kotlin (compatible con Java 1.8)

Framework UI: Jetpack Compose

Arquitectura: MVVM (Model-View-ViewModel)

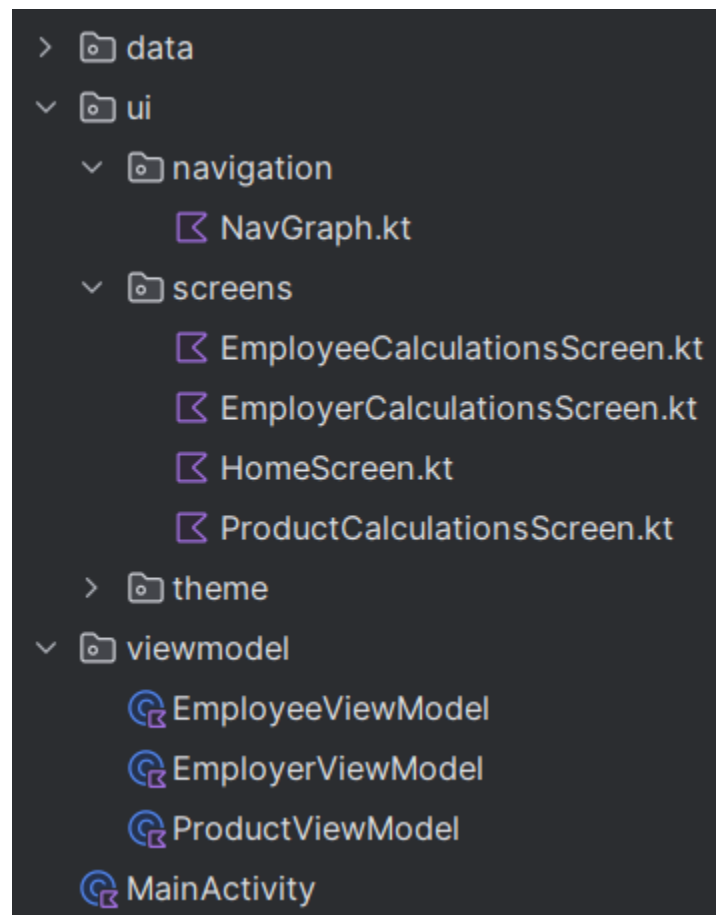
Gestión de Estados: State y MutableState

3. Arquitectura del Sistema

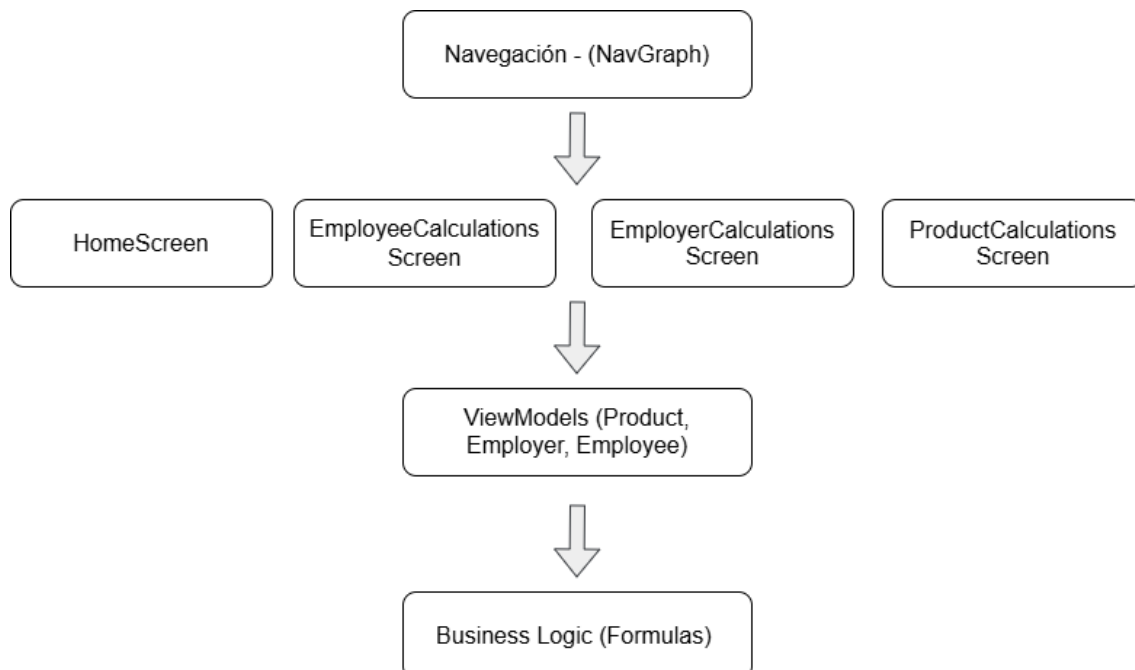
3.1 Patrones de Diseño

- **MVVM (Model-View-ViewModel):**
Se utiliza para separar la lógica de negocio (ViewModel) de la interfaz de usuario (View), asegurando una mejor mantenibilidad y testabilidad.
- **Navegación declarativa (Jetpack Navigation):**
Permite transiciones de pantallas entre categorías mediante un flujo centralizado.

3.2 Estructura del Proyecto



3.3 Diagrama de Componentes



4. Componentes Principales

4.1 Navegación

La navegación se implementa con **Jetpack Navigation Compose**, permitiendo la transición entre pantallas basadas en rutas.

Rutas:

- home: Pantalla principal.
- product_calculations: Cálculos de productos.
- employer_calculations: Cálculos de empleador.
- employee_calculations: Cálculos de empleado.

4.2 Pantallas

Cada pantalla tiene un único propósito y utiliza componentes reutilizables.

HomeScreen

- Botones para acceder a cada categoría.
- Uso de NavController para navegación.

4.3 ViewModels

Cada categoría tiene su propio ViewModel para gestionar cálculos y estados.

ProductViewModel

- Lógica para calcular precio con IVA, margen de ganancia, etc.
- Estados manejados con MutableState para reactividad.

5. Lógica de Negocio

Fórmulas

1. Cálculos de Productos:

- **Precio con IVA:** $\text{Precio base} * 1.19$
- **Margen de ganancia:** $((\text{Precio venta} - \text{Costo}) / \text{Precio venta}) * 100$
- **Punto de equilibrio:** $\text{Costos fijos} / (\text{Precio venta unitario} - \text{Costo variable unitario})$

- **ROI:** $((\text{Ingresos} - \text{Inversión}) / \text{Inversión}) * 100$

2. Cálculos de Empleador:

- **Aportes parafiscales:** Salario base * 9%
- **Seguridad social:** Salario base * 20.5%
- **Prestaciones sociales:** Salario base * 21.83%

3. Cálculos de Empleado:

- **Salario neto:** Salario base - (Salud + Pensión)
- **Horas extra diurna:** $(\text{Salario base} / 240) * 1.25$

6. Gestión de Estados

- **State y MutableState:**

Utilizados para actualizar la interfaz de usuario en tiempo real.

- **Validación de datos:**

Validamos entradas con funciones utilitarias en el ViewModel para evitar errores de cálculo.

7. Decisiones de Diseño

1. Jetpack Compose:

- Simplicidad para crear interfaces declarativas.
- Integración nativa con la arquitectura MVVM.

2. Arquitectura Modular:

- Pantallas independientes para facilitar la escalabilidad.
- Cada categoría tiene su propia lógica de negocio separada en ViewModels.

3. UX/UI:

- Diseño minimalista.
- Botones grandes y texto claro para una experiencia accesible.

8. Futuras Mejoras

1. Persistencia de Datos:

- Integrar una base de datos local como **Room** para guardar cálculos.

2. Soporte Multiplataforma:

- Adaptar el proyecto para **Jetpack Compose Multiplatform**.

3. Generación de Reportes:

- Exportar cálculos como PDF.