Informe Técnico Detallado del Proyecto: FinanzAPPs

# 1. Introducción

FinanzAPPs es una aplicación móvil desarrollada por el Grupo 19 de la Escuela Superior de Innovación y Tecnología (ESIT), como proyecto práctico del curso de Desarrollo de Aplicaciones Móviles. La app tiene como objetivo proporcionar una herramienta sencilla y eficiente para registrar ingresos, administrar gastos, y visualizar el ahorro mensual mediante gráficos.

# 2. Estructura y Navegación General de la Aplicación

La aplicación está organizada en pantallas (screens) que guían al usuario desde un inicio visual atractivo hasta la funcionalidad de gestión de finanzas personales. Cada pantalla está encapsulada como un widget, y las transiciones entre pantallas se hacen mediante Navigator. A continuación, se describe el flujo principal:  
  
Inicio → SplashScreen → SalaryInputScreen → HomeScreen → (Botón +) → AddExpenseScreen → HomeScreen → (Botón gráfico) → SummaryScreen

# 3. Funcionalidad de Archivos y Códigos Clave

## 3.1 main.dart

Este archivo contiene la función `main()` que ejecuta la aplicación mediante `runApp(FinanzApps())`. Define el widget principal de la app (`FinanzApps`) que establece el `MaterialApp`, define el tema general, y especifica la pantalla inicial (`SplashScreen`). Es el punto de arranque obligatorio de toda app Flutter.

## 3.2 splash\_screen.dart

Contiene una interfaz simple con fondo degradado, ícono y texto de bienvenida. Se muestra durante 2 segundos y luego navega automáticamente a la pantalla de ingreso de salario (`SalaryInputScreen`). Esto se hace mediante `Future.delayed(Duration(seconds: 2), () { Navigator.push(...); })` dentro de `initState()`.

## 3.3 salary\_input\_screen.dart

Esta pantalla solicita al usuario ingresar su salario mensual. El valor es validado y almacenado localmente con `SharedPreferences`. Si ya hay un salario almacenado, se suma al nuevo, permitiendo un control acumulativo. El valor se guarda con:  
`prefs.setDouble('salario', salarioActual + nuevoSalario);`  
Una vez ingresado, se navega a `HomeScreen`.

## 3.4 home\_screen.dart

Pantalla principal tras el ingreso del salario. Aquí se muestra:  
- El salario actual recuperado desde `SharedPreferences`  
- Una lista de los gastos recientes  
- Un botón flotante para agregar un nuevo gasto (abre `AddExpenseScreen`)  
- Un botón superior para ver el resumen gráfico (`SummaryScreen`)  
También permite editar y eliminar gastos mediante `showDialog()` con TextFields, y actualiza los cambios localmente en SharedPreferences.

## 3.5 expenses\_input\_screen.dart

Permite al usuario registrar un nuevo gasto. Los campos incluyen:  
- Monto  
- Categoría (Arriendo, Mercado, Servicios, Transporte, Estudio, Otros)  
- Fecha  
- Descripción (opcional)  
Los datos se validan y se guardan como cadena en `SharedPreferences`. Cada gasto incluye la fecha, monto, categoría y descripción opcional.  
Los campos vacíos o montos inválidos se notifican con `SnackBar`.

## 3.6 summary\_screen.dart

Muestra un gráfico circular generado con la librería `pie\_chart`. Los datos provienen de los gastos agrupados por categoría. También se muestra el ahorro restante del usuario en función del salario menos los gastos. Usa `PieChart()` y `ChartValuesOptions` para mostrar porcentajes. En caso de que no haya ahorro (ahorro negativo), se muestra en rojo.

## 3.7 pubspec.yaml

Este archivo define las dependencias del proyecto. Algunas incluidas fueron:  
- flutter  
- shared\_preferences (para almacenamiento local)  
- pie\_chart (para gráficos de resumen)  
Además, se personalizó la salida del archivo APK con:  
`outputFileName = "FinanzAPPs-Grupo19-v1.apk"` en build.gradle.

## 3.8 android/app/build.gradle

Se modificó el archivo `build.gradle` para personalizar el nombre de la APK generada con la línea:  
`outputFileName = 'FinanzAPPs-Grupo19-v1.apk'` dentro del bloque `applicationVariants.all {}`.  
Esto permite generar un archivo APK con nombre entendible para compartir fácilmente.

# 4. Problemas Frecuentes y Soluciones

1. Botón no guarda gasto → Revisar que todos los campos estén completos.  
2. Gastos no visibles → Verificar que se use `SharedPreferences.setStringList()` correctamente.  
3. Formulario no abre → Confirmar que `Navigator.push()` use correctamente el `BuildContext`.  
4. APK no genera → Ejecutar `flutter clean` y luego `flutter build apk --release`.

# 5. Conclusiones

FinanzAPPs cumple con su objetivo de facilitar el registro de gastos y administración del ingreso mensual. Con funciones intuitivas, almacenamiento local y un diseño gráfico atractivo, representa un proyecto completo y funcional para el aprendizaje y aplicación de Flutter. Se implementaron mejoras significativas como el Splash automático, resumen gráfico y edición de gastos, lo que añade valor a la experiencia del usuario.