

Manual de usuario – AhorrApp

Elaborado por:

Santiago Peralta Castellanos
José Luis Ramírez Isaza
Sebastián Triana Guanay

Presentado a

Néstor German Bolívar Pulgarín

Asignatura:

Programación Orientada a Objetos

Universiada Nacional de Colombia

2025 – 2s
Bogotá, DC
2025

MANUAL DE USUARIO – AHORRAPP

Introducción

AhorrApp es una aplicación diseñada para proporcionar una estructura base orientada a la gestión de información, lógica de negocio y ejecución modular en Java utilizando Gradle como sistema de construcción. Su código fuente está disponible públicamente y puede ser extendido, modificado y adaptado según las necesidades del usuario o desarrollador. Este manual tiene como objetivo explicar de forma detallada todo lo relacionado con la instalación, estructura, funcionamiento y personalización de la aplicación.

Objetivos del manual

- Guiar al usuario en la instalación y ejecución del proyecto.
- Describir de manera detallada la estructura interna del repositorio.
- Explicar cómo modificar y extender AhorrApp.
- Presentar buenas prácticas y recomendaciones para desarrollo y mantenimiento.

Requisitos Previos

Antes de ejecutar AhorrApp es necesario asegurarse de contar con las siguientes herramientas:

- Java JDK: versión 17 o superior.
- Gradle: aunque el proyecto incluye wrapper, es recomendable tener Gradle configurado.
- Git: para clonar y mantener actualizado el repositorio.
- IDE recomendado: IntelliJ IDEA o Android Studio.

Instalación del Proyecto

Clonar el repositorio

Ejecuta en consola: `git clone https://github.com/JosehhR/AhorrApp.git cd AhorrApp` o desde Android studio con la opción clonar repositorio

Construir el proyecto

El repositorio incluye Gradle Wrapper, por lo que puedes compilar sin instalar Gradle: `./gradlew build` o en Windows: `gradlew.bat build`

Ejecución de la Aplicación

Dependiendo de la configuración del archivo build.gradle.kts, AhorrApp puede ejecutarse mediante:
./gradlew run Si se genera un archivo .jar, puedes ejecutarlo manualmente: java -jar
build/libs/AhorrApp.jar

Estructura del Proyecto

El repositorio está dividido en varios directorios clave que permiten separar correctamente las responsabilidades del sistema. A continuación se presenta una descripción detallada de cada carpeta:

Carpeta raíz

Contiene los archivos principales de configuración:

- settings.gradle.kts: define el nombre del proyecto y módulos incluidos.
- build.gradle.kts: archivo central de construcción, dependencias y tareas.
- gradle.properties: parametrización del proyecto.
- .gitignore: archivos excluidos del control de versiones.
- gradlew / gradlew.bat: wrappers de Gradle para ejecutar tareas sin instalación previa.

Carpeta /app

Aquí se encuentra el código fuente principal. Su estructura es: app/ src/ main/ java/ (clases, paquetes y lógica de la app) main/resources/ test/java/ build.gradle.kts

Detalles Técnicos del Funcionamiento

Se presenta la forma en la que AhorrApp está diseñado para operar dentro del ecosistema Java + Gradle:

Sistema de construcción Gradle

AhorrApp utiliza Gradle en su versión Kotlin DSL. Esto significa que la configuración se hace mediante scripts .kts. Algunas características:

- Gestión de dependencias.
- Configuración de tareas personalizadas.
- Optimización del build.
- Integración con herramientas externas.

Arquitectura sugerida

Aunque el repositorio aún no define módulos avanzados, una arquitectura recomendada es:

- Model: entidades y estructuras de datos.
- Controller/Service: lógica de negocio.
- Repository: acceso a datos o persistencia.
- UI: interfaz gráfica, web o móvil.

Uso de AhorrApp

Actualmente AhorrApp funciona como base para futuras expansiones. Un uso típico incluye:

1. Ejecutar la aplicación.
2. Extender funcionalidades editando las clases dentro de /app/src/main/java.
3. Añadir dependencias necesarias en build.gradle.kts.
4. Construir y probar nuevas características

Buenas Prácticas de Desarrollo

- Mantener el código modular.
- Documentar cada clase y método relevante.
- Utilizar control de versiones con commits pequeños y claros.
- Escribir pruebas unitarias en /test.
- Mantener actualizado Gradle Wrapper.

Solución de Problemas

Error: No se reconoce 'gradlew'. **Solución:** Dar permisos de ejecución: chmod +x gradlew **Error:** Fallos en dependencias. **Solución:** Ejecutar: ./gradlew --refresh-dependencies **Error:** Java no reconocido. **Solución:** Instalar JDK y configurar JAVA_HOME.

Cómo Personalizar AhorrApp

AhorrApp está construido para ser extendido. Puedes modificar:

- Fluxos de ejecución.
- Clases internas.
- Sistema de persistencia.
- Integración de APIs externas.
- Nuevos módulos completos.

Conclusiones

Este manual proporciona una guía completa para trabajar con AhorrApp, desde su instalación hasta su modificación. El repositorio está preparado para servir como base robusta para nuevas aplicaciones de gestión, finanzas personales, o cualquier tipo de sistema modular escrito en Java con Gradle.