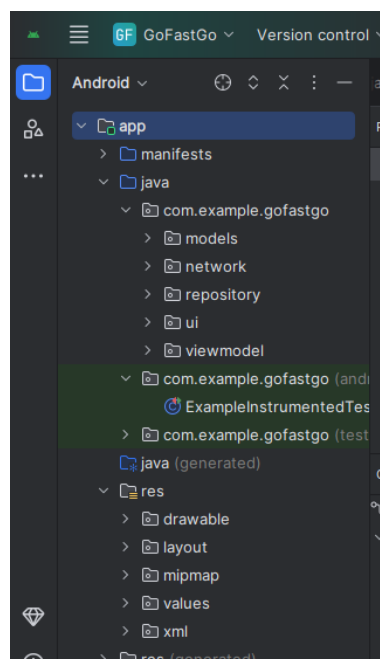
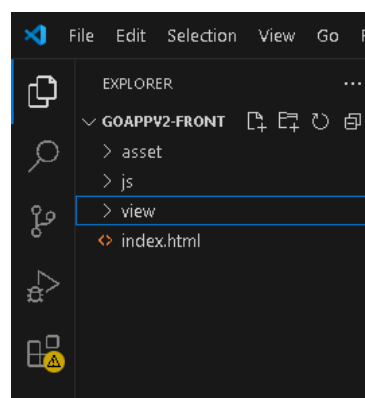
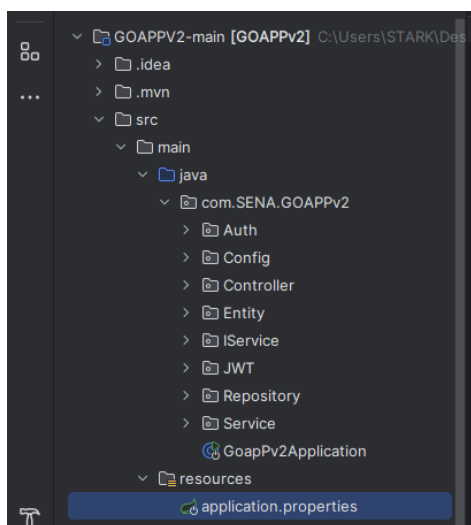


Artefactos de Código Go- Fast Go

Introducción:

Este documento describe los artefactos de código generados durante el desarrollo de Go Fast Go, una aplicación para la gestión de domicilios y pagos por horas de trabajo. Se incluyen detalles sobre los archivos fuente Scripts y componentes desarrollados para las plataformas de escritorio, y móvil.



Descripción de los Artefactos

Backend:

Lenguaje Framework: Java con Spring Boot

1. **Controller** contiene la lógica de negocio, los endpoints para la gestión de los usuarios, validaciones de turnos, autenticaciones
2. **Modelos (Entity)**: Define los modelos de datos utilizados en la base de datos
3. **Config**: contiene archivos de configuración para spring Security, JWT, Cors
4. **Repository**, gestiona el JPA para el manejo de la base de datos
5. **Service, IService** contiene las clases de lógica de negocio o servicios de la aplicación:

Service

- Encapsula la lógica de negocio (procesamiento de datos, reglas del negocio, validaciones)
- Interactuar con la capa de persistencia (repositorios o DAOS)
- Proveer métodos reutilizables para los controladores
- Ejecuta transacciones y manejar errores a nivel de negocio

IService

- Facilita la inyección de dependencias
- Permite intercambiar implementaciones sin afectar el código que las usa
- Mejorar la estabilidad al permitir el uso de mocks en pruebas unitarias

6. JWT implementación Json web token: es el mecanismo de autenticación que permite a los usuarios identificarse mediante token firmando digitalmente, en lugar de utilizar sesiones de usuario tradicionales, requiere autenticación sin estado stateless

App Móvil con Android Studio:

1. Modelos: define las clases de modelo de datos, representan los objetos con los que trabaja la aplicación y pueden ser utilizadas para manejar información en diferentes partes del código como la interfaz de usuario, bases de datos o APIS
 - Modelos de apis
 - Modelos de bases de datos
 - Modelos para viewmodel
2. Network: contiene clases relacionadas con solicitudes HTTP, conexión con APIS y manejo de datos remotos
 - Api Service
 - Retrofit: solicitudes http a la api
 - Network : intercepta y modifica solicitudes http
 - Api response clase genérica para manejar respuestas de la api

3. Repository : sigue el patrón de diseño repository pattern, que actúa como intermediario entre la capa de datos (api, base de datos)
Maneja la lógica de obtención de datos, combinando múltiples fuentes:
 - Api remotas(retrofit, volley)
 - Bases de datos locales (Room SQLite)
 - Archivos locales o sharedPreferences
4. Ui : organiza todo lo relacionado con la interfaz gráfica y la lógica de presentación de una app de Android , sigue la arquitectura (mvvm) model view controller, para separar la UI de la lógica de negocio
5. Viewmodel se encarga de manejar la lógica de presentación de la UI, separándola de la Activity y fragment

Frontend, Visual Studio Code

1. Assets : se usa para almacenar los recursos estáticos como imágenes iconos fuentes y archivos css o json
2. Contiene la lógica del negocio con la parte frontend, como funciones reutilizables , controladores, servicios, validaciones
3. View
4. Almacena las vistas de la aplicación, paginas principales que se muestran al usuario.

Dependencias principales:

Backend: Java Spring Boot, Spring security, JWT

Frontend Escritorio: bootstrap

App movil: Java Movil

Android Studio: Retrofit, Room ShareReference

Conclusion:

Esta documentación proporciona una visión general de los artefactos de código, las tecnologías utilizadas y las dependencias clave. Esto facilita la colaboración y el mantenimiento del sistema.