

Alunos:

Sérgio Luiz Teixeira Nunes Júnior

Lucas Andrei Conceição do Sacramento

Documentação do Projeto: Sistema de Roteamento Dinâmico com Command e Injeção de Dependências

1. Descrição Geral

O projeto consiste em um sistema de roteamento dinâmico implementado em Java. Ele utiliza o padrão de projeto **Command**, combinado com **injeção de dependências** baseada em **anotações** e **reflexão**. A aplicação permite gerenciar rotas HTTP, alternar entre diferentes implementações de repositórios (memória ou banco de dados HSQLDB) e garantir que determinadas classes sigam o padrão Singleton.

2. Arquitetura do Sistema

Componentes Principais

1. Servlet Centralizado:

Um único servlet (CentralServlet) gerencia todas as requisições HTTP e encaminha para os métodos apropriados em controladores.

2. Repositórios de Dados:

- Implementações em memória (Singletons) e persistência no HSQLDB.
- O **Factory Method** permite alternar dinamicamente entre essas implementações.

3. Anotações Personalizadas:

- **@Rota**: Define métodos que respondem a rotas HTTP específicas.
- **@Inject**: Identifica campos que precisam de dependências injetadas.
- **@Singleton**: Garante que uma classe seja instanciada apenas uma vez.

4. Reflexão:

- Usada para:
 - Mapear dinamicamente rotas para métodos anotados com @Rota.
 - Realizar injeção de dependências em campos anotados com @Inject.
 - Controlar a criação de instâncias únicas para classes anotadas com @Singleton.

3. Implementação

3.1. Padrão Command

O padrão **Command** centraliza o processamento de requisições no servlet único. Métodos anotados com `@Rota` em controladores são mapeados dinamicamente para rotas HTTP específicas.

3.2. Injeção de Dependências

O sistema injeta automaticamente dependências em campos anotados com `@Inject`, utilizando um mecanismo de reflexão. Isso simplifica a configuração de controladores e outras classes.

3.3. Padrão Singleton

As implementações de repositórios em memória usam o padrão Singleton para garantir que apenas uma instância de cada repositório seja criada durante a execução da aplicação.

3.4. Factory Method

O **Factory Method** permite alternar dinamicamente entre repositórios em memória e persistentes (HSQLDB), com base na configuração.

4. Reflexão e Carregamento Dinâmico

4.1. Processamento de Rotas

Durante a inicialização do sistema, métodos anotados com `@Rota` são registrados automaticamente em um mapeamento de rotas. Isso elimina a necessidade de configuração manual de endpoints HTTP.

4.2. Injeção de Dependências

Campos anotados com `@Inject` são identificados e preenchidos automaticamente com as instâncias corretas, utilizando um mecanismo de fábrica.

4.3. Garantia de Singleton

Classes anotadas com `@Singleton` são monitoradas para garantir que apenas uma instância seja criada e reutilizada.