

# SISTEMA DE RESTAURANTE

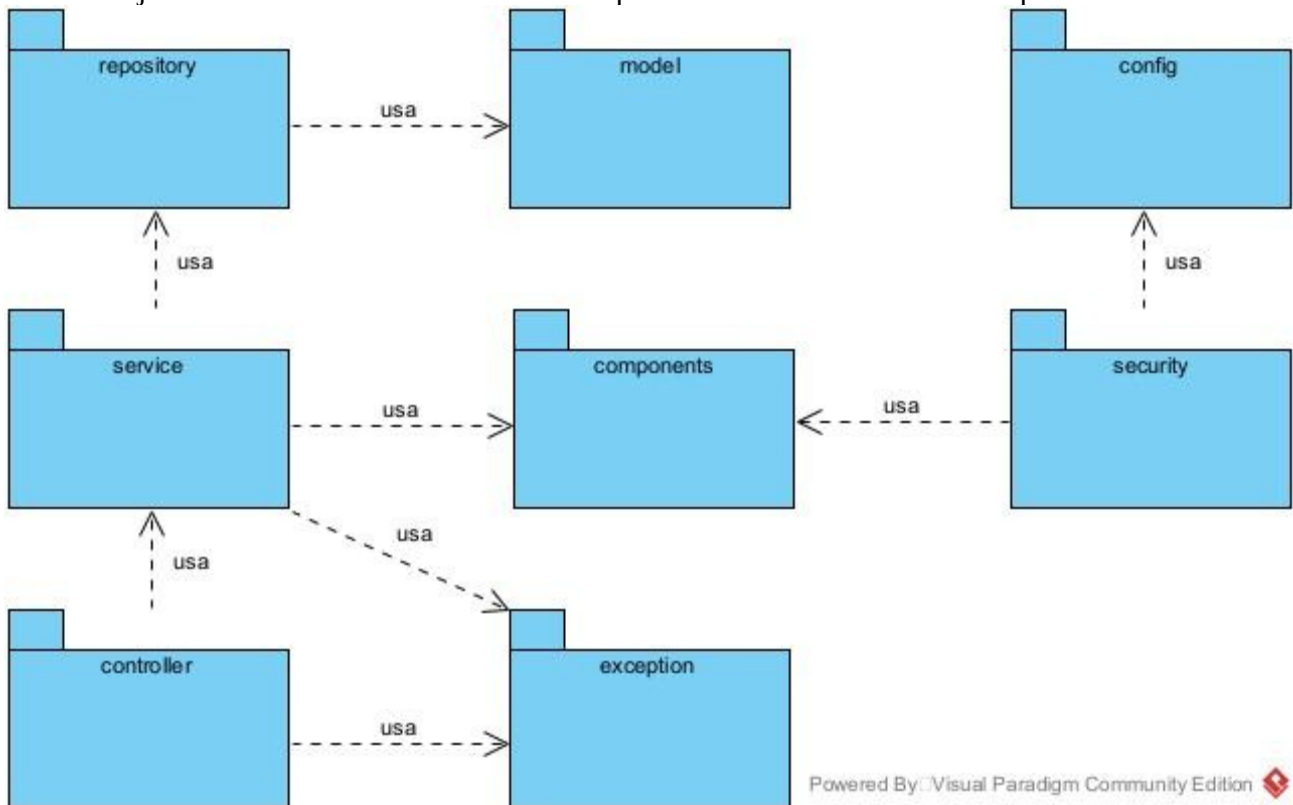
## Tecnologias

O sistema a ser desenvolvido funcionará com **interface web**. Ele utiliza banco de dados e dependerá de um plano de hospedagem para funcionar em produção. As seguintes tecnologias estão sendo utilizadas:

- **Java 22:** Linguagem de programação utilizada para desenvolvimento da aplicação
- **Spring Boot 3,3,0:** Framework para desenvolvimento de web services
  - ***Spring Data JPA:*** Implementação da especificação JPA pelo Spring
  - ***Spring Security:*** Utilizado na autenticação e autorização
    - ***Token JWT:*** O token JSON Web Token (JWT) é utilizado para autenticação stateless e autorização baseada nos papéis embutidos no token
- **PostgreSQL17:** banco de dados utilizado.
- **JasperReports:** Framework utilizado para geração de relatórios em PDF. Há também o **JasperSoft**, o software utilizado para, de modo visual, criar e configurar o relatório.
- **Maven:** Gerenciador de pacotes Java utilizado.
- **Junit5:** Framework utilizado para testes de unidade
- **Angular 18:** Framework para desenvolvimento frontend
- **Angular Material:** Extensão do Angular com vários componentes de interface gráfica prontos.

## Estrutura de pacotes

Veja abaixo como estão estruturados os pacotes do sistema com suas dependências:



## Descrição dos pacotes

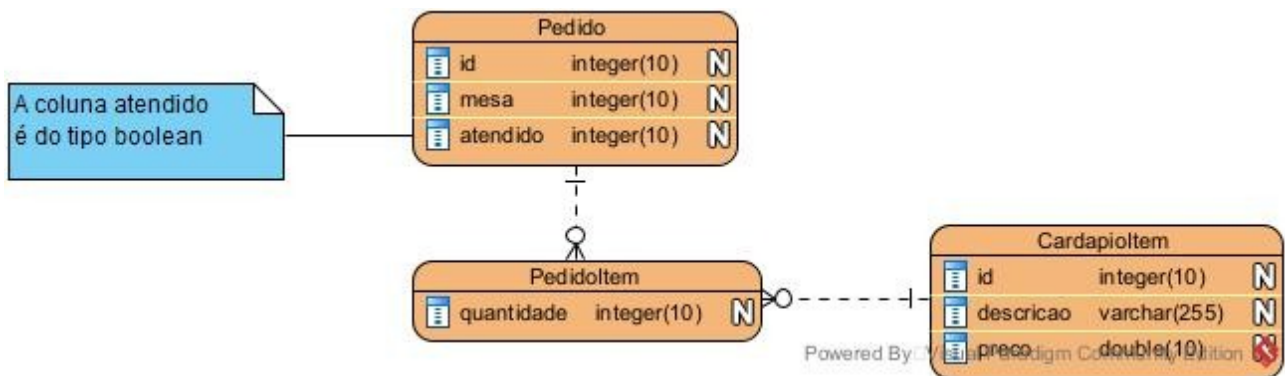
Abaixo, uma breve descrição dos principais pacotes do sistema e os relacionamentos entre eles e seus componentes:

- **model**: Pacote onde estão os objetos de mapeamento das tabelas no banco de dados e também os objetos que mapeiam os JSONs de requisição e resposta.
  - **model/request**: Onde ficam os objetos de mapeamento de JSONs de requisição
  - **model/response**: Onde ficam os objetos de mapeamento de JSONs de resposta
- **repository**: Pacote onde ficam os objetos que são utilizados como objetos da camada de acesso a dados.
  - É nesses objetos onde ficam, também, as queries de consulta ao banco de dados da aplicação que já não vêm por padrão nos objetos JpaRepository
- **service**: Pacote onde fica a lógica de negócio da aplicação. Os objetos desse pacote utilizam um ou vários objetos de repositórios e conhecem tanto os objetos do modelo quanto os de requisição e resposta.
- **controller**: Pacote onde ficam as definições e implementações dos endpoints da aplicação
  - Na comunicação entre os Controllers e os Services, os métodos do service recebem como parâmetro objetos de requisição e, quando retornam algo, deve ser objetos de resposta ou lista de objetos de resposta.
  - Os services também podem receber em alguns casos, além de objetos de requisição, tipos primitivos como inteiro, long, double e string.
  - Os controllers não conhecem os objetos de mapeamento de tabelas do banco de dados.

- **config:** Onde são definidos os beans que podem englobar configurações do sistema ou objetos injetados em outras partes do sistema.
- **security:** Onde fica as classes de configuração e implementação da segurança da aplicação
- **components:** É nesse pacote onde são definidos objetos injetados em outras partes da aplicação, mas que não são necessariamente services, mas sim, outros componentes.

## Persistência no banco de dados

Veja abaixo o modelo de entidade e relacionamento (MER) inicial do banco de dados:



A classe *pedido* tem uma relação de muitos para muitos com a entidade **item de cardápio**. Isto é, o **item de pedido** tem uma quantidade e o mesmo **item de cardápio** pode estar em vários **itens de pedido**. O *pedido* tem vários itens.

A conexão com o banco de dados tem as seguintes propriedades:

**driver:** `org.postgresql.Driver`  
**url:** `jdbc:postgresql://localhost:5432/sisrest`  
**username:** `postgres`  
**password:** `postgres`