



Centro Universitário de Excelência
Sistemas de Informação

FoodDelivery – Peru

Sistema de Pedidos para Restaurantes

Autores:

Franklin Ferreira
Giovanna Rodrigues
Kailla Costa
Lucas Barreto
Lucas Oliveira

Feira de Santana, 2025

Contéudo

Apresentar o desenvolvimento do sistema de pedidos de restaurante em Java utilizando Programação Orientada a Objetos.

1. Introdução: Objetivos do projeto

2. Divisão de tarefas:
Funções de cada integrante

3. Funcionalidades: Funções implementadas

4. MenuCLI:
Código em execução

5. Questão para reflexão:
Criação de uma nova classe

6. Pedido → Cliente: Regras de associação

7. Diagrama de Classes:
Construção final do diagrama

8. Referências: Fontes usadas para construção do projeto

Introdução



- Desenvolvimento do sistema **FoodDelivery – Peru** em Java.
- Aplicação de conceitos de **POO** (encapsulamento, composição, coesão).
- Divisão em **5 módulos** (P1–P5), cada integrante responsável por uma função.

Divisão de Tarefas



- **P1 – Infraestrutura (Lucas Oliveira).**

Criar a base estável para o sistema

- **P2 – Clientes (Kailla Costa).**

Permitir o cadastro e listagem de clientes

- **P3 – Cardápio (Giovanna Rodrigues).**

Gerenciar os itens disponíveis pra venda e operações de CRUD

- **P4 – Pedidos (Lucas Barreto).**

Criar pedidos, gerenciar seus itens e controlar seu ciclo de vida através dos status

- **P5 – Relatórios (Franklin Ferreira)**

Consolidar todos os dados em uma base única (CentralDeDados) e gerar relatórios

Funcionalidades



O sistema permite:

Cadastro de Clientes: O sistema permite registrar clientes com nome e telefone (obrigatórios). IDs são gerados automaticamente.

Gerenciamento de Cardápio: Itens podem ser cadastrados com nome (não vazio) e preço (> 0). Códigos são sequenciais e gerados automaticamente.

Registro de Pedidos: Pedidos são criados com número sequencial, data/hora de criação, status inicial ACEITO e associados a um cliente. Itens do cardápio são adicionados com quantidades.

Controle de Status: Transições de status seguem fluxo rígido (ACEITO \rightarrow PREPARANDO \rightarrow ... \rightarrow ENTREGUE), sem que seja possível fazer saltos.

Relatórios:

Simplificado: Total de pedidos e valor arrecadado.

Detalhado: Lista completa de pedidos com clientes, itens e valores.

MenuCLI



Quando executamos o programa:

```
===== FOOD DELIVERY =====  
1) Cadastrar Cliente  
2) Listar Clientes  
3) Cadastrar Item no Cardápio  
4) Listar Itens do Cardápio  
5) Criar Pedido  
6) Avançar Status de Pedido  
7) Listar Pedidos por Status  
8) Relatório de Vendas (Simplificado)  
9) Relatório de Vendas (Detalhado)  
0) Sair  
=====
```

Escolha uma opção:

Questão para reflexão



Uma associação direta entre as classes Pedido e ItemCardapio se mostrou insuficiente, pois apenas indicaria que um item estava no pedido, mas não quantas unidades desse item foram solicitadas

Como conseguimos representar a quantidade de cada item dentro de um pedido se a associação é direta?

Foi criada a classe **PedidoItem**, que atua como uma classe de associação. Esta decisão permite que a quantidade seja armazenada como um atributo específico dessa relação. Dessa forma, o sistema pode representar com precisão que um Pedido contém *n* unidades de um determinado ItemCardapio

Pedido → Cliente

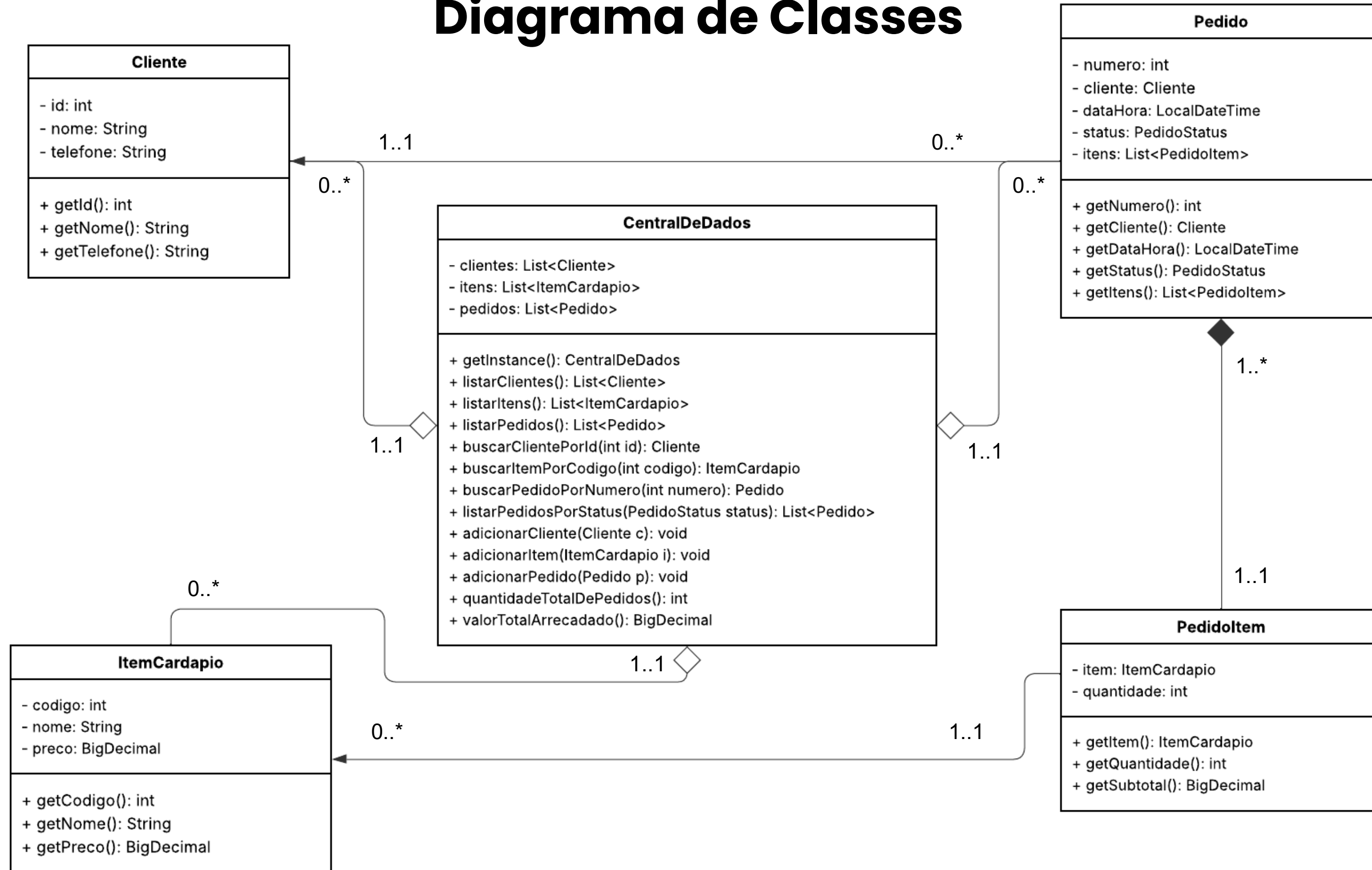


Optou-se pela seta simples de associação (\rightarrow) em vez do diamante preto de composição ($\blacklozenge \rightarrow$) entre Pedido e Cliente (no exemplo mostrado pelo professor), porque os objetos possuem ciclos de vida independentes.

Enquanto um pedido depende da existência de um cliente para ser criado, a exclusão de um pedido não implica na exclusão do cliente associado.

O cliente é uma entidade independente que pode existir e realizar múltiplos pedidos no sistema, caracterizando uma relação de associação e não de composição.

Diagrama de Classes



Referências



Guanabara, G. (2020). Curso de Java POO. [s.l.]: Curso em Vídeo. Disponível em:
https://www.youtube.com/watch?v=KIIL63MeyMY&list=PLHz_AreHm4dkqe2aR0tQK74m8SFe-aGsY.
Acesso em: 19 ago. 2025.

DEVMEDIA. (2010). Padrão de Projeto Singleton em Java. [s.l.]: DevMedia. Disponível em:
<https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392>. Acesso em: 22 ago. 2025.

DEVMEDIA. Encapsulamento em Java. São Paulo: DevMedia, 2018. Disponível em:
<https://www.devmedia.com.br/encapsulamento-em-java/29106>. Acesso em: 23 ago. 2025.