Centro Universitário de Excelência
Sistemas de Informação

OttawaDelivery – Sistema de Delivery em Java

Autores: Alex Oliveira
Alice Aragão
Ana Clara Ribeiro
Monique Prado

Vitória da Conquista, 2025

Agenda

Objetivo da apresentação

1. Descrição do sistema: Como o sistema foi estruturado em Java.

2. Tecnologias utilizadas: Ferramentas e linguagens aplicadas.

3. Funcionalidades implementadas : Principais recursos do sistema.

4. Diagrama de Classes: Estrutura do sistema para simular todo o processo .

5. Experiência do usuário: Base sólida e escalável para uma futura interface intuitiva

6. Conclusão e próximos passos: Resultados e melhorias futuras.

Descrição do sistema

- O sistema foi desenvolvido em **Java**, com foco em **orientação a objetos**.
- A arquitetura inclui classes específicas para **Cliente, Pedido, Item de Cardápio, Fluxo de Entrega e Relatórios**.
- O fluxo simula desde o **cadastro do cliente** até a **entrega do pedido**, com geração de relatórios sobre o funcionamento.
- Estrutura modular que facilita manutenção, expansão e reutilização de código.

Tecnologias utilizadas

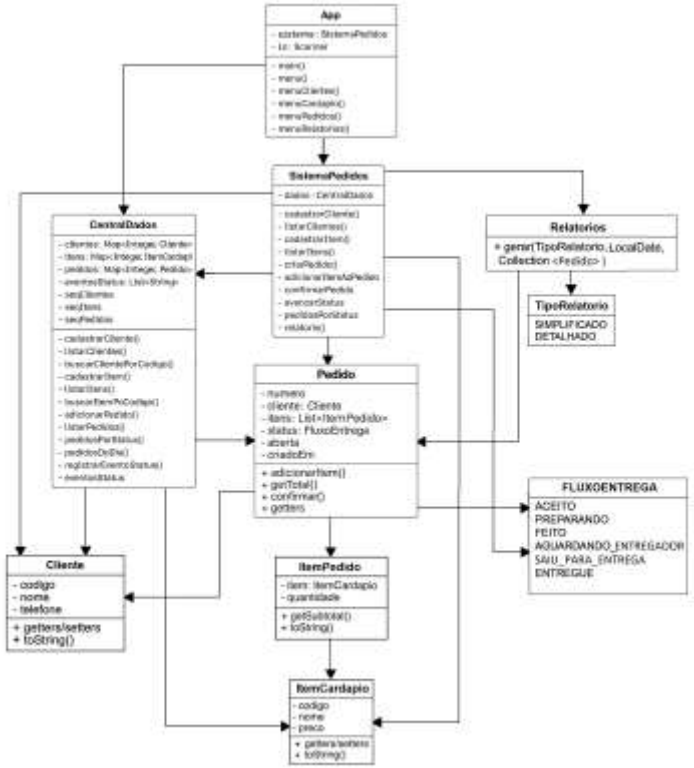
- Java: linguagem principal de desenvolvimento.
- Programação Orientada a Objetos (POO): base da modelagem do sistema.
- Estrutura modular com classes: permite organização e clareza no código.
- Console (linha de comando): interface simples para interação e testes.
- Git/GitHub: versionamento e controle de código em equipe.

Funcionalidades implementadas

- Cadastro de clientes: inclusão de novos usuários no sistema.
- Gerenciamento de cardápio: itens de comida/bebida disponíveis para pedidos.
- Criação de pedidos: associação de clientes aos itens do cardápio.
- Controle de entregas: acompanhamento do fluxo até a finalização.
- Relatórios: visão geral de desempenho e histórico de pedidos.

Diagrama de Classe

- O diagrama de classes mostra a organização modular do sistema, com classes que representam clientes, pedidos, itens do cardápio, controle de fluxo de entrega, centralização de dados e geração de relatórios, interagindo de forma coesa para simular o processo completo de delivery. Imagem completa



Experiência do usuário

Embora ainda não haja interface gráfica, o sistema já foi estruturado para garantir uma experiência clara e escalável.

- **Fluxo Lógico:** pedidos seguem etapas coerentes (cadastro → pedido → entrega → relatório).
- **Prevenção de Erros:** regras de negócio validam operações antes de avançar.
- **Centralização de Dados:** uso de Singleton mantém consistência e reduz falhas.
- **Clareza Técnica:** comandos, logs e mensagens facilitam testes e depuração.
- **Escalabilidade:** arquitetura modular permite evolução sem comprometer funções existentes.
- **Base para Usabilidade:** back-end sólido prepara terreno para futura interface intuitiva.

Conclusão e Próximos Passos

Conclusão

- O OttawaDelivery é um sistema funcional de delivery, desenvolvido em Java e orientado a objetos.
- A arquitetura modular e o uso de POO facilitaram a manutenção e a expansão do sistema.

Próximos Passos

- Implementar interface gráfica para melhorar a experiência do usuário.
- Integrar persistência de dados com banco de dados real.
- Expandir funcionalidades como promoções e avaliações de clientes.

Referências

- Repositório do projeto OttawaDelivery: https://github.com/HeyAlexOliveira/tia-lu-dev-web-oo-ottawa
- DevMedia - Principais conceitos da Programação Orientada a Objetos: https://www.devmedia.com.br/principais-conceitos-da-programacao-orientada-a-objetos/32285.
- Alura - Programação Orientada a Objetos (POO): https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos?srsltid=AfmBOorUixodtqmF044417xR1-FlYUBbpwP5QPt4Hp_pr74_5qJcTBV4.
- DevMedia - Diagrama de Classes (UML): https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224.
- DevMedia - Padrão de Projeto Singleton em Java: https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392.