



**Centro Universitário de Excelência**  
**Sistemas de Informação**

# **FoodDelivery: Implementação de Sistema de Gerenciamento de Pedidos para Restaurantes**

**Autores:** Clara Gabryellen P. Aderno, Igor dos S. Vieira, Jhon Luiz S. Santos, Natan C. Da Silva, Rayan S. Silva

[gabryellenpaixao@gmail.com](mailto:gabryellenpaixao@gmail.com), [igorvieira1445@gmail.com](mailto:igorvieira1445@gmail.com), [devjhon01010@gmail.com](mailto:devjhon01010@gmail.com),  
[natancorreiatr@gmail.com](mailto:natancorreiatr@gmail.com), [rayansantos725@gmail.com](mailto:rayansantos725@gmail.com)

Feira de Santana, 2025

# Agenda

Objetivo da apresentação:

a criação de um protótipo de sistema de gerenciamento de pedidos para um restaurante, operando através de uma interface de linha de comando (CLI) e utilizando a linguagem de programação Java.

**1. Item 1:** Primeiro desafio do programa de trainee FoodDelivery, focado no desenvolvimento back-end orientado a objetos. O objetivo foi criar um protótipo de sistema de gerenciamento de pedidos para um restaurante, operando via interface de linha de comando (CLI) em Java.

**2. Item:** A solução emprega conceitos fundamentais de Programação Orientada a Objetos, como classes, objetos, atributos, métodos e relacionamentos de Associação, Agregação e Composição.

**3. Item:** Foram desenvolvidas funcionalidades para cadastro de clientes e itens de cardápio, gerenciamento de pedidos com ciclo de vida definido e geração de relatórios de vendas, mantendo os dados exclusivamente em memória

**4. Item:** Este sistema foi projetado para gerenciar clientes, itens de cardápio e pedidos, com todos os dados mantidos exclusivamente em memória sem persistência de banco de dados.

**Apresenta a implementação do primeiro entregável do programa trainee da empresa FoodDelivery, que tem como objetivo capacitar talentos em desenvolvimento web orientado a objetos. O desafio inicial consistiu na criação de um protótipo de sistema para gerenciamento de pedidos de restaurante, utilizando a linguagem Java e operando em uma interface de linha de comando (CLI). A solução desenvolvida aplicou conceitos fundamentais da Programação Orientada a Objetos, como classes, objetos, atributos, métodos e relacionamentos. O sistema permite o cadastro de clientes, gerenciamento de cardápio, acompanhamento completo do ciclo de pedidos e a geração de relatórios de vendas, com todos os dados mantidos em memória.**

**A Programação Orientada a Objetos (POO) fundamenta-se em conceitos como classes, atributos, construtores, métodos e diagramas de classes, essenciais para o desenvolvimento de sistemas robustos e organizados. No projeto FoodDelivery, esses elementos foram aplicados na linguagem Java para modelar clientes, itens de cardápio e pedidos. As classes representam entidades do domínio; os atributos armazenam informações específicas, como nome, preço e códigos automáticos; os construtores garantem a criação de objetos válidos; e os métodos encapsulam a lógica de negócios. O diagrama de classes em UML documenta a estrutura do sistema, evidenciando os relacionamentos entre entidades e a necessidade de uma classe intermediária para gerenciar itens em pedidos.**

# **Conceitos Fundamentais de POO**

## **Aplicados ao Sistema FoodDelivery**



**O sistema FoodDelivery foi modelado a partir de conceitos fundamentais da programação orientada a objetos. As classes representam os elementos do domínio, como Cliente, ItemCardapio e Pedido, definindo atributos e comportamentos. Os atributos armazenam as características de cada objeto, como nome, telefone e preço, além de códigos únicos e sequenciais gerados automaticamente. Os construtores garantem a criação de objetos em estado válido, como no caso de um pedido que deve ser iniciado com cliente, status “ACEITO” e data/hora de registro. Já os métodos encapsulam a lógica de negócios e possibilitam operações como calcular valores e atualizar status. Por fim, o diagrama de classes em UML documenta a estrutura estática do sistema, mostrando relações importantes, como a associação entre cliente e pedido, bem como a necessidade de uma classe intermediária para gerenciar itens em um pedido.**

# Metodologia



A metodologia empregada no desenvolvimento do sistema FoodDelivery foi estruturada em quatro etapas principais. A primeira consistiu na análise e modelagem inicial, com levantamento de requisitos funcionais, regras de negócio e definição do ciclo de vida dos pedidos. Em seguida, ocorreu a implementação das classes de domínio em Java, contemplando atributos, métodos e construtores para garantir consistência na criação dos objetos. A terceira etapa envolveu o desenvolvimento de uma interface de linha de comando (CLI), permitindo a interação do usuário de forma simplificada. Por fim, utilizou-se o Git para versionamento do projeto e colaboração entre os membros da equipe, assegurando integração e organização no processo de desenvolvimento.



# Resultados



A implementação do primeiro entregável da FoodDelivery gerou um protótipo funcional em Java, com interface de linha de comando (CLI) e dados mantidos em memória. O sistema contempla o gerenciamento de clientes, itens de cardápio e pedidos, incluindo cadastro, listagem, atualização de status e consulta. Foram desenvolvidos relatórios de vendas em duas modalidades (simplificado e detalhado), além de regras de negócio como geração automática de códigos únicos, numeração sequencial de pedidos e ciclo de vida rígido com transições de status pré-definidas. O projeto foi desenvolvido em grupo, versionado com Git e documentado por meio de um diagrama de classes que evidencia toda a estrutura estática, incluindo a resolução da relação muitos-para-muitos entre pedidos e itens.

# Considerações Finais



O desenvolvimento do programa proporcionou aprendizado prático em Programação Orientada a Objetos (POO), aplicando conceitos como classes, atributos, construtores e métodos no gerenciamento de pedidos via CLI. O sistema implementou funcionalidades essenciais, regras de negócio como geração automática de códigos e ciclo de vida rígido de pedidos, e utilizou uma central de dados para controle. Os principais desafios envolveram restrições da POO sem herança e polimorfismo e a manipulação de dados em memória. A experiência consolidou habilidades em Java e Git, e futuras melhorias incluem persistência de dados, herança, polimorfismo, tratamento de exceções, testes automatizados e notificações.



# Referências



<https://ramonrdm.wordpress.com/wp-content/uploads/2011/09/java-orientado-a-objetos.pdf>

[https://s3.amazonaws.com/cm-cls-content/201801/INTERATIVAS\\_2\\_0/PROGRAMACAO\\_ORIENTADA\\_A\\_OBJETOS/U1/LIVRO\\_UNICO.pdf](https://s3.amazonaws.com/cm-cls-content/201801/INTERATIVAS_2_0/PROGRAMACAO_ORIENTADA_A_OBJETOS/U1/LIVRO_UNICO.pdf)

LIVRO Programação orientada a objetos: Conceitos e técnicas