

## **ETAPA 0**

### **Slice of Life - aplicação para pedidos de pizza**

#### **Resumo**

Este projeto consistirá em uma aplicação em Java que permitirá aos usuários realizar pedidos de pizza de forma interativa e intuitiva, utilizando JavaFX e SceneBuilder. A aplicação apresentará um menu com opções de tamanhos de pizza e ingredientes, permitindo que o cliente personalize sua pizza conforme suas preferências. Os usuários poderão facilmente adicionar e remover itens do carrinho de compras, visualizando o total do pedido antes da finalização. Durante a etapa de finalização, serão coletadas informações essenciais do usuário, como nome e endereço, para processar o pagamento de forma segura. O objetivo é desenvolver uma solução eficaz que simplifique o processo de pedidos de pizza, aumentando a eficiência e a satisfação do usuário.

#### **Motivação**

A motivação deste projeto é a oportunidade de utilizar o JavaFX e o SceneBuilder, uma ferramenta NoCode, para desenvolver uma aplicação que permita aos usuários fazer pedidos de pizza de forma interativa e personalizada, além do Java e POO. Queremos explorar essas tecnologias enquanto criamos uma experiência divertida e intuitiva, que não apenas atenda à demanda por soluções práticas no setor de alimentação, mas também nos ajude a aprimorar nossas habilidades técnicas e criativas.

#### **Proposta**

##### **Funcionamento**

A tela inicial da aplicação exibirá o nome do sistema e um botão que irá direcionar o cliente para a tela de pedidos. Na tela de pedidos, o cliente poderá selecionar o tamanho da pizza e a quantidade de sabores desejada (2, 3 ou 4). Após escolher o número de sabores, o cliente será levado à etapa de personalização, onde poderá selecionar os ingredientes para cada divisão da pizza. Concluída a escolha da quantidade de sabores e dos ingredientes, o cliente avançará para o preenchimento de dados essenciais, como nome, endereço e forma de pagamento. A tela final apresentará uma revisão do pedido, oferecendo a opção de adicionar outra pizza, caso deseje. Se o cliente optar por finalizar a compra, uma nota fiscal será gerada automaticamente.

##### **Escopo**

O projeto consiste em uma implementação similar aos sistemas de totens de pedidos em praças de alimentação, mas com a conveniência de o cliente poder fazer isso de casa. A aplicação não exigirá cadastro ou login, focando exclusivamente na perspectiva do cliente, sem incluir funcionalidades administrativas. O cliente poderá montar sua pizza escolhendo ingredientes individualmente, em vez de optar por sabores predefinidos, como pepperoni ou calabresa. Cada pizza poderá ter até 5 ingredientes por sabor. A aplicação reunirá

todas as informações sobre os pedidos feitos pelos clientes, permitindo uma experiência personalizada e prática.

### **Apps semelhantes**

- [Pratika Pizzaria](#)
- [Nono Ludovico](#)

### **Dificuldades esperadas**

Durante o desenvolvimento da aplicação de pedidos de pizza, podemos enfrentar dificuldades relacionadas ao uso do JavaFX e do SceneBuilder, especialmente no aprendizado de como linkar os componentes criados no SceneBuilder com o código JavaFX para manipulá-los adequadamente. Essa integração é fundamental para garantir que a interface gráfica funcione de maneira eficiente e responda corretamente às interações do usuário. Além disso, será desafiador dividir corretamente as classes e interligá-las, já que a estruturação adequada do código é essencial para a organização e manutenção da aplicação. Encontrar o equilíbrio entre as responsabilidades de cada classe e a comunicação entre elas pode ser complexo.

### **Relação com a disciplina**

A relação do trabalho com a disciplina de Técnica de Construção de Programa é relevante, pois os conceitos aprendidos serão essenciais para o desenvolvimento da aplicação de pedidos de pizza. A refatoração nos ajudará a organizar o código, enquanto a depuração será crucial para identificar e corrigir erros. Além disso, os fundamentos de teste de software garantirão que cada funcionalidade funcione corretamente. A adoção de boas práticas de programação manterá o código limpo e facilitará a colaboração. Conceitos como herança e polimorfismo serão úteis para criar hierarquias eficientes e reutilizar código, enquanto a decomposição permitirá dividir a aplicação em partes gerenciáveis. No geral, o que aprendemos na disciplina será fundamental para construirmos um software de qualidade.