



ESCOLA  
**POLITÉCNICA**

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

Algoritmos e Estruturas de Dados I - Turma 033 - 2024/1

Prof. Leonardo Rodriguez Heredia

Marcelo Dos Santos Benfato  
Marco Antônio Laimer de Oliveira  
Mateus Alves Bittencourt  
Vitor Kaefer Guttler

## **T2 – Simulador de Pedidos de Pizza**

PORTO ALEGRE - RS

2024

## **Resumo**

O trabalho consiste em implementar um programa que simula o recebimento e processamento de pedidos em uma pizzeria. Utiliza-se uma unidade de tempo fictícia que inicia em 0 e avança de 1 em 1. Cada pedido é feito em um instante específico e requer um tempo determinado para ser preparado. Como a pizzeria tem apenas um pizzaiolo, ele processa um pedido de cada vez, com os pedidos adicionais aguardando em uma fila. Os dados de simulação são fornecidos em um arquivo CSV contendo informações sobre o código do pedido, o sabor da pizza, o instante de tempo do pedido e o tempo de preparo necessário.

## **Desenvolvimento**

Aqui está um resumo de como vamos desenvolver o programa para simular o recebimento e processamento de pedidos em uma pizzeria:

### **1. Criar a Classe `Pedido`**

Primeiro, vamos criar uma classe chamada `Pedido`. Essa classe vai representar cada pedido feito na pizzeria e terá algumas informações importantes:

- Código do pedido: um número único para identificar cada pedido.
- Sabor da pizza: para saber qual pizza foi pedida.
- Instante do pedido: a hora em que o pedido foi feito.
- Tempo de preparo: quanto tempo o pizzaiolo precisa para fazer a pizza.

A ideia aqui é ter uma forma organizada de armazenar essas informações para cada pedido.

### **2. Ler os Dados do Arquivo CSV**

Depois, no arquivo `SimuladorPizzeria.java`, vamos implementar a leitura dos dados dos pedidos que estão em um arquivo CSV. Vamos usar uma lista para armazenar todos os pedidos que foram lidos do arquivo. Cada linha do CSV vai virar um objeto `Pedido`, assim a gente mantém tudo bem estruturado.

### **3. Implementar a Fila de Pedidos**

Como só temos um pizzaiolo, precisamos de uma fila para gerenciar os pedidos que chegam. Vamos usar uma estrutura de fila para armazenar os

*pedidos enquanto o pizzaiolo está ocupado. Cada novo pedido que chega vai para o final dessa fila e espera a sua vez.*

#### **4. Simular o Processamento dos Pedidos**

*A simulação vai rolar baseada em uma unidade de tempo fictícia, que começa em 0 e vai aumentando de 1 em 1. A cada unidade de tempo, vamos checar se o pizzaiolo está disponível. Se ele estiver, ele começa a fazer o próximo pedido na fila. Se não, o pedido atual continua sendo processado até ficar pronto. Quando o pedido termina, o pizzaiolo pega o próximo da fila.*

#### **5. Gerar um Relatório de Resultados**

*No final da simulação, o programa vai gerar um relatório mostrando quando cada pedido foi concluído. Isso vai ajudar a entender quanto tempo cada pedido levou e se teve algum gargalo no processo.*

#### **Resumindo**

- Classe `Pedido*`: Guarda as informações de cada pedido.
- Leitura do CSV: Lê os pedidos do arquivo CSV e armazena em uma lista.
- Fila de Pedidos: Organiza os pedidos enquanto esperam para ser feitos.
- Simulação do Tempo: Processa os pedidos um por vez, conforme o tempo passa.
- Relatório de Resultados: Mostra quando cada pedido foi concluído.

*Com essa abordagem, a gente vai conseguir simular o funcionamento da pizzeria de forma bem realista, mostrando como os pedidos são recebidos, processados e entregues.*

## **Conclusão**

*Concluindo, o desenvolvimento do programa de simulação para uma pizzeria mostrou como gerenciar eficientemente o recebimento e processamento de pedidos utilizando uma abordagem estruturada. A implementação das classes `Pedido` e `SimuladorPizzaria`, a leitura de dados de um arquivo CSV, a gestão de uma fila de pedidos e a simulação do tempo permitiram uma representação realista do fluxo de trabalho da pizzeria. Este programa pode ser expandido para incluir mais funcionalidades no futuro, como a adição de mais pizzaiolos ou a otimização do tempo de preparo. No geral, a simulação fornece uma ferramenta útil para analisar e melhorar o processo operacional de uma pizzeria.*