



Centro Universitário de Excelência
Sistemas de Informação

Sistema de Gerenciamentos de Pedidos

Tia Lu Food Delivery

Grupo Pernambuco

Autor: Jefte Martins
Pedro Silveira
Paulo Soares
David Cairo
Ian Neves

Vitória da Conquista, 2025

Agenda

O Objetivo dessa apresentação é apresentar o desenvolvimento de um sistema de pedidos em Python, demonstrando a aplicação prática e as diferenças fundamentais entre as estruturas de dados de lista e fila no gerenciamento de informações e processos.

1. Introdução

Contextualização do projeto e a importância das estruturas de dados.

4. Resultados e Discussão

Demonstração das funcionalidades e da aplicação prática das estruturas no sistema.

2. Fundamentação Teórica

Apresentação dos conceitos de Listas e Filas, suas operações e diferenças

3. Metodologia

Detalhamento da arquitetura do sistema e de como as estruturas foram implementadas

5. Conclusão

Análise dos desafios, aprendizados do projeto e sugestões para trabalhos futuros.

Introdução



A Importância das Estruturas de Dados

- **Conteúdo para discussão:**
- A Ciência da Computação dedica-se a resolver problemas e manipular informações de forma eficiente.
- Nesse cenário, as estruturas de dados são um pilar essencial, oferecendo modelos para organizar, armazenar e acessar dados de maneira otimizada.
- A escolha da estrutura de dados correta é um fator determinante na eficiência de um software. Uma seleção inadequada pode tornar uma solução inviável, enquanto a escolha certa é a base para sistemas robustos.



Fundamentação Teórica

Listas e Filas



Listas (Flexibilidade Total):

- **Conteúdo para discussão:**
- O que são: Uma estrutura de acesso restrito que segue a política FIFO (First-In, First-Out): o primeiro a entrar é o primeiro a sair.
- Uso ideal: Essencial para garantir a ordem cronológica em processos, como em uma fila de atendimento ou no gerenciamento de tarefas de um sistema. Suas operações se resumem a adicionar no fim (Enqueue) e remover do início (Dequeue).

Filas (Disciplina e Ordem):

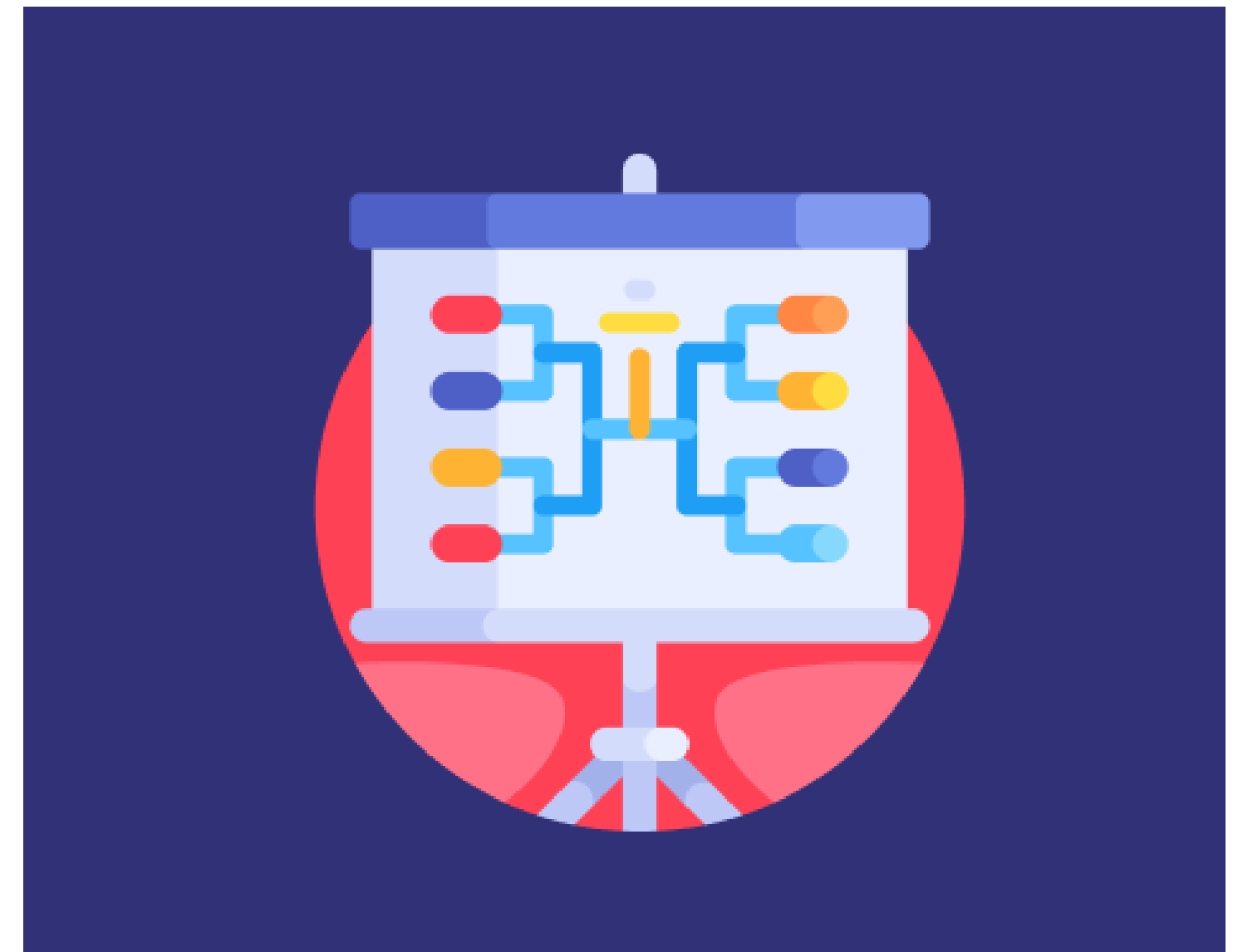
- **Conteúdo para discussão:**
- O que são: Uma estrutura linear e flexível que permite acesso, inserção e remoção de elementos em qualquer posição por meio de índices.
- Uso ideal: Perfeita para coleções que exigem manipulação constante, como o gerenciamento de um cardápio ou uma lista de tarefas.

Metodologia

Estrutura do Sistema



- **Conteúdo para discussão:**
- O sistema foi implementado em um ambiente de console, com o objetivo de focar na aplicação das estruturas de dados. A arquitetura foi organizada de forma modular para facilitar a implementação e a manutenção do código. O módulo de
- Gerenciamento de Itens utiliza listas para armazenar os dados do cardápio, simulando um vetor dinâmico que permite cadastrar, consultar e atualizar produtos com flexibilidade. Já o módulo de
- Gerenciamento de Pedidos organiza as solicitações em três listas que funcionam como filas (pendentes, aceitos e prontos), garantindo que os pedidos sejam processados na ordem em que chegam, respeitando a disciplina FIFO. Por fim, um módulo de consultas e Relatórios permite a visualização e filtragem do histórico de pedidos por status



Sistema de Pedidos em Python com Listas e Filas

Resultados e Discussão

O Sistema em Operação



- **Conteúdo para discussão:**

- Durante a execução, observou-se o funcionamento adequado das listas (cardápio) e das filas (pedidos).
- Cadastro e Edição de Itens: O código implementado permitiu adicionar e gerenciar com sucesso novos itens à lista cardápio, exemplificando a flexibilidade da estrutura.
- Criação de Pedidos (FIFO): Foi implementada a criação de pedidos associados a clientes, respeitando a disciplina FIFO. O bloco de código que utiliza: `append` (adicionar no final) e `pop(0)` (remover do início) foi o mais importante para garantir essa disciplina.
- Evolução de Status: O sistema demonstrou uma evolução consistente dos status dos pedidos (pendente, aceito, em preparo, finalizado). O código ilustra como um pedido transita entre diferentes filas, o que garante a consistência do fluxo.
- Consultas e Filtros: O sistema permitiu a realização de consultas gerais e filtradas, possibilitando gerar relatórios de pedidos por qualquer status definido no sistema (ACEITO, FAZENDO, etc.).

```
# === MÓDULO 1: GERENCIAR MENU DE ITENS =====
if opcao_principal == '1':
    print("\n" * 3)
    print("--- Gerenciar Menu de Itens ---")
    print("1. Cadastrar Item")
    print("2. Consultar Itens")
    print("3. Atualizar Item")
    print("4. Voltar ao Menu Principal")
    opcao_itens = input("Escolha uma opção: ")

# --- Cadastrar Item ---
if opcao_itens == '1':
    print("\n" * 3)
    print("-- Cadastro de Novo Item --")
    nome = input("Nome do item: ")
    descricao = input("Descrição do item: ")
    try:
        preco = float(input("Preço (ex: 45.50): "))
        estoque = int(input("Quantidade em estoque: "))

        # Cria o item
        novo_item = {
            'id': proximo_codigo_item,
            'nome': nome,
            'descricao': descricao,
            'preço': preco,
            'estoque': estoque
        }
        cardapio.append(novo_item)
        proximo_codigo_item += 1
        print(f"\nItem '{nome}' cadastrado com sucesso! id: {novo_item['id']}")
    except ValueError:
        print("\nERRO: Preço e estoque devem ser números. Operação cancelada.")
    input("Pressione Enter para continuar...")
```

O código de Cadastro e Edição de Itens

Considerações Finais & Aprendizados



- O projeto proporcionou uma experiência enriquecedora, tanto no aspecto técnico quanto no de aprendizado coletivo.
- Ficou evidente na prática a versatilidade das listas, que podem assumir diferentes papéis dentro de um mesmo sistema, como gerenciar itens e controlar o fluxo de pedidos.
- A importância da disciplina FIFO no uso das filas foi reforçada pelo desafio de manter a consistência no fluxo dos pedidos.
- O projeto cumpriu seu propósito de consolidar o entendimento sobre listas e filas e abriu caminhos para reflexões sobre como sistemas simples podem evoluir para soluções mais complexas e robustas.

Referências



- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Introduction to Algorithms. 3. ed. Cambridge, MA: MIT Press, 2009.
- LAFORE, R. Data Structures & Algorithms in Java. 2. ed. Indianapolis, IN: Sams Publishing, 2012.
- TANENBAUM, A. S.; BOS, H. Modern Operating Systems. 4. ed. Upper Saddle River, NJ: Pearson Education, 2015.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: Teoria e Prática. 3. Ed. Rio de Janeiro: Elsevier, 2012.
- WIRTH, N. Algoritmos + Estruturas de Dados = Programas. 1. Ed. Rio de Janeiro: LTC, 1985.
- SEDGEWICK, R.; WAYNE, K. Algorithms. 4. Ed. Boston: Addison-Wesley, 2011.
- TANENBAUM, A. S. Estruturas de Dados Usando C. São Paulo: Pearson, 2009.
- LAUDON, K. C.; LAUDON, J. P. Sistemas de Informação Gerenciais. 12. Ed. São Paulo: Pearson, 201.