

SISTEMA DE PEDIDOS

Autor: Bruna Floro
Davi Goes
Emily Oliveira
Gabriel Macedo
Lizandra Soares

Agenda

O objetivo desta apresentação é explicar como foi feito o sistema de pedidos do Food Delivery.

1. Introdução:

Protótipo de aplicativo de Food Delivery
Desenvolvido em Python

2. Estrutura de dados

Estruturas Utilizadas

3. Funcionalidades do Sistema

4. Fluxo de Pedidos

Filas

5. Considerações Finais

Introdução



Imagine uma rotina de vendas fluída, eficiente e sem complicações, onde sua equipe de vendas tem todas as ferramentas necessárias para receber e enviar pedidos de forma ágil, precisa e sem erros.

Nesse cenário, a gestão de entrada e pedidos, faturamento e relacionamento com o cliente se tornam tarefas simples e mais fáceis, graças à automação. O protótipo de aplicativo de Food Delivery que vamos apresentar torna isso uma realidade.

Estrutura de dados



- **Listas**

- Gerenciam o cardápio de itens (inserção, remoção, atualização)

- **Filas (FIFO)**

- Controlam o fluxo dos pedidos (pendentes, aceitos, prontos)

Funcionalidades do Sistema



- Cadastro e atualização de itens do cardápio
- Criação de pedidos com cálculo de total e desconto (cupom DESCONTO10)
- Atualização de status: AGUARDANDO → ACEITO → FAZENDO → ENTREGUE
- Geração de relatório diário: vendas, cancelamentos

```
# Submenu de Gerenciamento de Itens
while True:
    print("\n--- GERENCIAR MENU DE ITENS ---")
    print("1. Cadastrar Novo Item")
    print("2. Atualizar Item")
    print("3. Consultar Todos os Itens")
    print("4. Voltar ao Menu Principal")

    sub_opcao = input("Escolha uma opção: ")

    if sub_opcao == "1":
        # Cadastrar novo item
        print("\n--- CADASTRAR NOVO ITEM ---")
        nome = input("Nome do item: ")
        descricao = input("Descrição: ")
```

```
# Atualizar status do pedido
if not pedidos:
    print("Nenhum pedido cadastrado!")
    continue

print("\n--- ATUALIZAR STATUS DO PEDIDO ---")
numero_pedido = int(input("Número do pedido: "))

pedido_encontrado = None
for pedido in pedidos:
    if pedido['numero'] == numero_pedido:
        pedido_encontrado = pedido
        break

if not pedido_encontrado:
    print("Pedido não encontrado!")
    continue
```

Fluxo de Pedidos

- **Sistema de Pedidos**

- (*fila_pendentes*): pedidos aguardando aprovação
- (*fila_aceitos*): pedidos aceitos em preparo
- (*fila_prontos*): pedidos prontos para entrega
- Permite processamento sequencial e controle de estoque

```
# Atualizar filas
if novo_status == 'ACEITO':
    if pedido_encontrado in fila_pendentes:
        fila_pendentes.remove(pedido_encontrado)
        fila_aceitos.append(pedido_encontrado)
elif novo_status == 'FEITO':
    if pedido_encontrado in fila_aceitos:
        fila_aceitos.remove(pedido_encontrado)
        fila_prontos.append(pedido_encontrado)
elif novo_status in ['REJEITADO', 'CANCELADO']:
    # Devolver itens ao estoque se cancelado
    if novo_status == 'CANCELADO':
        for item_pedido in pedido_encontrado['itens']:
            for item_menu in itens_menu:
                if item_menu['codigo'] == item_pedido['codigo']:
                    item_menu['estoque'] += item_pedido['quantidade']
                    break
```

Considerações finais



O desenvolvimento do sistema de pedidos apresentado neste trabalho evidenciou a importância da tecnologia como ferramenta estratégica para otimizar processos de compra e venda.

- Agilidade e transparência no atendimento
- Redução de falhas operacionais
- Maior controle de vendas e estoque

Referências



- <https://weduu.com/blog/weduu/sistema-de-pedidos/>
- <https://blog.consumer.com.br/conheca-5-sistemas-de-pedidos-para-restaurantes-da-consumer/>
- Código Sistema_de_pedidos.py