



Centro Universitário de Excelência
Sistemas de Informação

Food Delivery Rio Grande do Norte

Autor:

<Breno Matos Bastos>
<Bruno Sampaio Silva>
<Claudio dos Santos Junior>
<Gabriel Gomes Cruz Uzeda>
<Herick Marcio Matos Brito>
<Levi Falcão de Queiroz>

Feira de Santana, 2025

Agenda

O objetivo dessa apresentação é criar um sistema para simular um restaurante. Nosso projeto busca de maneira simples e funcional representar o processo de atendimento, desde o cadastro de itens até processar pedidos.

1. Introdução

Objetivo do trabalho e o uso de filas e listas

2. Criar itens

Objetivo do trabalho e o uso de filas e listas

3. Criar Pedidos

Objetivo do trabalho e o uso de filas e listas

4. Processar Pedidos

Objetivo do trabalho e o uso de filas e listas

5. Consultar pedidos:

Objetivo do trabalho e o uso de filas e listas

Introdução



Filas: Organização de ordens no atendimento

- Movimentação em estágios como:
aguardar-aceito-fazendo-feito-entregue

Listas: Estoque ordenado dos Itens

- Estrutura Completa e simples para cadastros
e gerenciamento

Criar Item



Funcionamento da função

- cadastrar_Item()
- ID, Nome, Estoque, Preço, desc
- Inputs para o usuário
- Tratamento de dados
- Colocar na lista 'itens'

```
def cadastrar_item():
    nome = input("Nome do item: ")
    descricao = input("Descrição do item: ")
    preco = None
    quantidade_estoque = None

    # Tratar preco e quantidade para ser float e int
    while type(preco) != float:
        try:
            preco = float(input("Preço do item: "))
        except:
            print('\n-----Insira um número válido!-----\n')

    while type(quantidade_estoque) != int:
        try:
            quantidade_estoque = int(input("Quantidade em estoque: "))
        except:
            print('\n-----Insira um número válido!-----\n')

    global sequencia_cod

    produto = [nome, sequencia_cod, preco, descricao, quantidade_estoque]

    itens.append(produto) # adiciona o produto na lista de itens (final)
    sequencia_cod += 1 # incrementa +1 na variável global 'sequencia_cod'
    print(f"Novo item: {produto[0]}. Cadastrado com sucesso!")
    return produto
```

Criar Pedido



Funcionamento da função

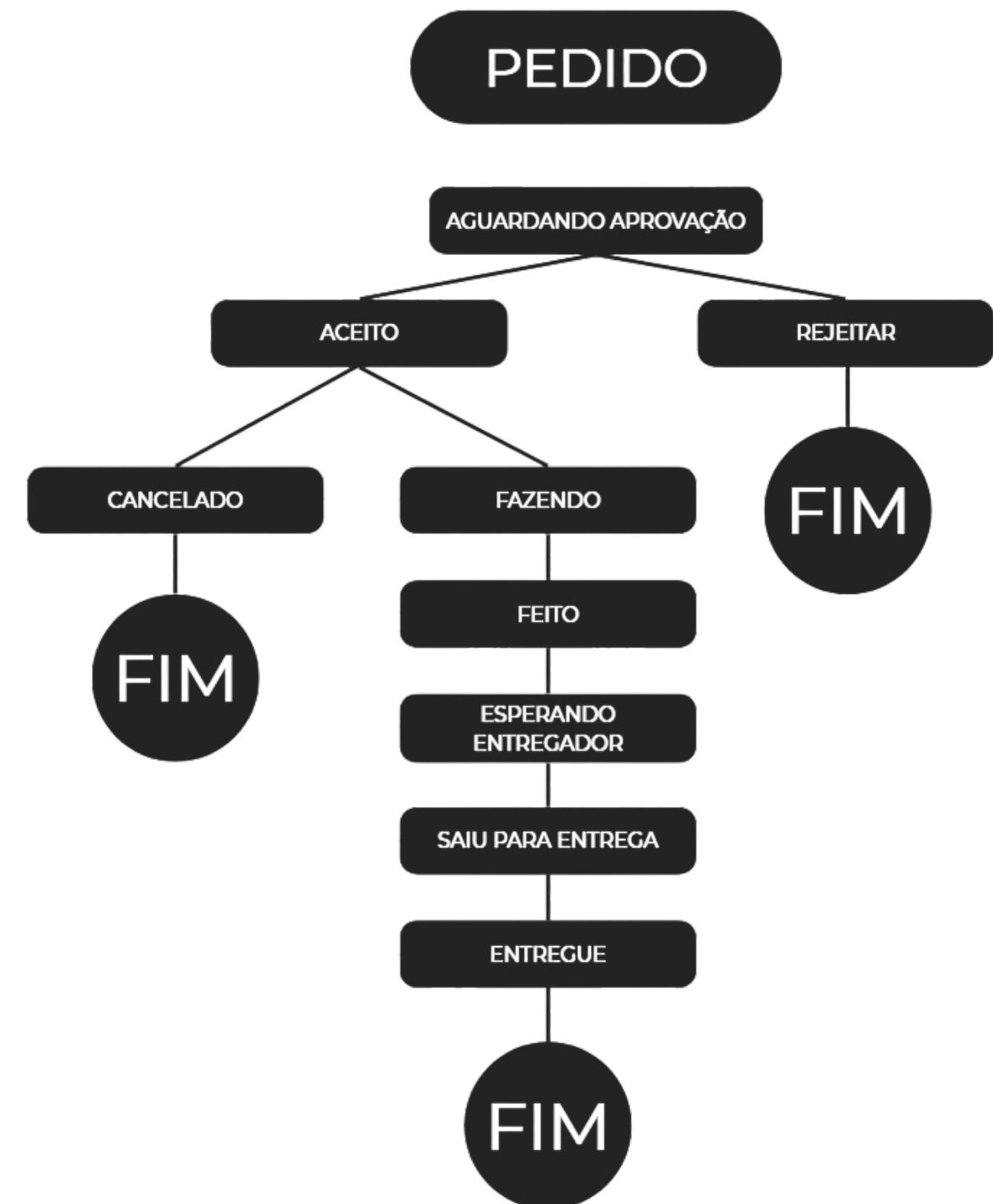
- Variável global para código
- Verificar Estoque do item
- Recebe 2 cupons
- Finalizada apenas com 'fim
- Variável global para código

```
def criar_pedido():  
    if len(itens) == 0:  
        print("Itens não cadastrados.")  
        return  
  
    cupom = ''  
    status = 'AGUARDANDO APROVACAO'  
    pedido_itens = []  
    valor_total = 0.0  
  
    global sequencia_cod_pedido  
    pedido = [sequencia_cod_pedido]  
  
    print("\nItens disponíveis:")  
    for item in itens:  
        print(f"Código: {item[1]}, Nome: {item[0]}, Preço: R$ {item[2]}")
```

Processar Pedido

Funcionamento da função

- Utiliza o modelo FIFO
- Ordem com base no ID
- Filtragem com base no 'status'
- Finalizada apenas com 'fim



Consultar Pedidos



Funcionamento da função

- Imprime todos os pedidos
- Consulta com base nos dados
- Função externa para formatar
- Filtragem por 'status'

```
def imprimir_detalhes_pedido(pedido):
    """Função auxiliar para imprimir um único pedido de forma formatada."""
    print("\n-----")
    print(f"Pedido Código: #{pedido[0]:02d}")
    print(f"Status: {pedido[4]}")
    print("Itens:")

    # Itera sobre a lista de itens dentro do pedido
    for item_comprado in pedido[1]:
        # item_comprado = ['código', 'nome', quantidade, subtotal]
        codigo_prod = item_comprado[0]
        nome_prod = item_comprado[1]
        quantidade = item_comprado[2]
        subtotal = item_comprado[3]
        print(f"  - {quantidade}x {nome_prod} (Cód: {codigo_prod}) - Subtotal: R$ {subtotal:.2f}")

    if pedido[3]: # Se houver um cupom
        print(f"Cupom Aplicado: {pedido[3]}")

    print(f"Valor Total: R$ {pedido[2]:.2f}")
    print("-----")

def consultar_pedido():
    if not todos_os_pedidos:
        print("\n--- Não há pedidos registrados. ---")
        return

    # Passo 1: Imprimir todos os pedidos de forma formatada
    print("\n===== Histórico de Todos os Pedidos =====")
    for pedido in todos_os_pedidos:
        imprimir_detalhes_pedido(pedido)
```

Considerações Finais

- O projeto LuFood possibilitou a aplicação de conceitos de estruturas de dados em um cenário realista.
- Gerenciar a consistência dos dados utilizando apenas listas.
- Garantir os dados após o termino do programa