

Sistema de Gerenciamento de Restaurante

Descrição Geral

Desenvolver um sistema de gerenciamento para restaurante que integre controle de estoque, gestão da cozinha, atendimento aos clientes e pagamentos.

Requisitos Funcionais

1. Gestão de Estoque

- Cadastrar, consultar e atualizar produtos no estoque
- Cada produto deve ter: código, nome, quantidade, unidade de medida, preço unitário e data de validade
- Alertar sobre produtos com estoque baixo ou próximos ao vencimento

2. Gestão da Cozinha

- **Cardápio**
 - Cadastrar itens no cardápio (nome, descrição, preço, ingredientes necessários)
 - Consultar e atualizar itens do cardápio
- **Receitas**
 - Vincular cada item do cardápio aos ingredientes necessários e suas quantidades
 - Ao preparar um prato, verificar se há ingredientes suficientes no estoque
 - Baixar automaticamente os ingredientes do estoque quando um prato é preparado
- **Gestão de Preparo**
 - Registrar pedidos recebidos dos clientes
 - Mostrar fila de pratos a serem preparados
 - Atualizar status de preparo (recebido, em preparo, pronto para servir)

3. Gestão de Pedidos

- **Mesas**
 - Cadastrar mesas do restaurante (número, capacidade, status)

- Atribuir clientes às mesas
- Visualizar status das mesas (livre, ocupada, reservada)
- **Pedidos**
 - Registrar pedidos por mesa
 - Adicionar, modificar ou cancelar itens de um pedido
 - Enviar pedidos para a cozinha
 - Consultar status dos pedidos

4. Gestão de Pagamento

- **Contas**
 - Calcular conta total por mesa
 - Permitir divisão da conta entre clientes
 - Aplicar descontos ou taxas (serviço)
- **Formas de Pagamento**
 - Registrar pagamentos (dinheiro, cartão)
 - Calcular troco quando necessário
 - Fechar mesa após pagamento
- **Relatórios Financeiros**
 - Total de vendas por dia
 - Itens mais vendidos
 - Valor médio gasto por mesa

Requisitos Técnicos

1. Estrutura de Dados

- Utilizar listas e dicionários para armazenar dados
- Criar estruturas para relacionar:
 - Itens do cardápio com ingredientes
 - Mesas com pedidos
 - Pedidos com itens consumidos

2. Interface de Usuário

- Menu principal com submenus para cada área do sistema
- Interface baseada em texto usando print e input
- Exibir informações de forma organizada (tabelas simples usando formatação de strings)

3. Funções e Organização

- Implementar funções específicas para cada funcionalidade
- Organizar o código em seções lógicas (módulos conceituais)
- Documentar as funções com comentários explicativos

4. Validações e Tratamento de Erros

- Verificar disponibilidade de ingredientes antes de confirmar pedidos
- Validar entradas do usuário
- Fornecer mensagens de erro claras

5. Persistência de Dados (opcional/bônus)

- Salvar e carregar dados em arquivos de texto
- Implementar histórico de pedidos e vendas

Fluxo Básico do Sistema

1. Inicializar o sistema carregando dados (se disponíveis)
2. Exibir menu principal com opções para:
 - Gestão de Estoque
 - Gestão da Cozinha
 - Gestão de Mesas e Pedidos
 - Gestão de Pagamentos
 - Relatórios
 - Sair
3. Para cada área, exibir submenu com funcionalidades específicas
4. Executar operações selecionadas e retornar ao menu correspondente
5. Ao sair, salvar dados (se implementado)

Sugestões de Implementação Progressiva

Para alunos que estão aprendendo, sugiro implementar o sistema em fases:

1. **Fase 1:** Sistema básico de estoque e cardápio
2. **Fase 2:** Adicionar gestão de mesas e pedidos
3. **Fase 3:** Implementar cálculo de contas e pagamentos
4. **Fase 4 (avançada):** Integrar todos os módulos para controle automático do estoque baseado nos pedidos

Este desafio expandido permite que os alunos apliquem todos os conceitos já estudados (variáveis, condicionais, loops e funções) em um contexto mais próximo de um sistema real, desenvolvendo habilidades importantes de organização de código e modelagem de dados.