Etapa 1 - Escolha e Delimitação do Problema.

Objetivo: Um sistema para organizar estoque e vendas seja de um bar pequeno ou estebelecimentos de bairro

Escreva o nome do sistema — Vendas\_Do\_Bar

Em bares e estabelecimentos menores o controle de vendas e estoque muitas vezes são feitos manualmente com papel e planilhas, podendo ocorrer erros de anotação e erro de pedidos sem falar da falta de controle do estoque.

Criar um sistema simples para gerir e controlar as vendas e o estoque , o programa permitirá cadastrar produtos , registrar cada mesa , calcular o total e atualizar automaticamente o estoque e o total de vendas por dia

Redução de erros e maior controle de vendas e estoque acelerando o atendimento e fornecendo informações imediatas sobre vendas e estoque.

Publico alvo: Donos e proprietários de bares e estabelecimentos

Principais informações que o sistema precisa armazenar:

Cadastro de produtos: código, nome, preço, categoria (bebida/petisco), unidade de medida.

Estoque: quantidade atual por produto, ponto de reposição mínimo.

Mesas/pedidos: número da mesa, itens pedidos (produto, quantidade), status do pedido (aberto/fechado).

Vendas/caixa: registros de vendas por data/hora, total por venda, método de pagamento (opcional).

Relatórios básicos: total de vendas do dia, itens mais vendidos, produtos com estoque abaixo do ponto mínimo.

Etapa 2 - Análise de Requisitos Computacionais.

Passos detalhados:

O sistema deve cadastrar, alterar e remover produtos (código, nome, preço, categoria, unidade).

O sistema deve registrar pedidos por mesa, associando itens e quantidades a uma mesa específica.

O sistema deve calcular o total de cada pedido, aplicando quantidades e preços.

O sistema deve atualizar automaticamente o estoque ao registrar venda/fechamento de pedido.

O sistema deve permitir o fechamento de conta por mesa, gerando um registro de venda no caixa.

O sistema deve gerar relatórios básicos (vendas do dia, itens mais vendidos, produtos abaixo do ponto de reposição).

O sistema deve salvar e carregar dados de persistência em arquivos (produtos, estoque, vendas, pedidos abertos).

O sistema deve listar produtos e consultar estoque por produto ou por categoria.

Usabilidade: a interface textual deve ser simples e navegável por teclado, com mensagens claras e validações de entrada.

Performance/Leveza: o sistema deve rodar responsivamente em máquinas com poucos recursos (baixo uso de memória/CPU).

Persistência e confiabilidade: os dados devem ser gravados em arquivos texto/binary ao finalizar operações críticas (ex.: fechamento de conta) para evitar perda de dados

|  |  |  |
| --- | --- | --- |
| Tipo | Descrição | Exemplo do sistema |
| ENTRADA | Dados digitado pelo usuário | Produto, cerveja,batata frita |
| PROCESSAMENTO | Operações realizadas pelo  Sistema | Produto 1 + produto 2 = valor |
| SAÍDA | Resultados apresentados ao usuário | Valor total = |

Pseudocódigo INÍCIO

// Bloco 1 - Inicialização e carregamento de dados

carregar\_dados\_produtos()

carregar\_dados\_estoque()

carregar\_dados\_vendas()

// Menu principal

REPITA

limpar\_tela()

exibir\_menu\_principal()

ler\_opcao(opcao)

SE opcao == 1 ENTÃO

cadastrar\_produto()

SENÃO SE opcao == 2 ENTÃO

registrar\_pedido()

SENÃO SE opcao == 3 ENTÃO

fechar\_pedido()

SENÃO SE opcao == 4 ENTÃO

gerar\_relatorio()

SENÃO SE opcao == 5 ENTÃO

exibir\_estoque()

SENÃO SE opcao == 0 ENTÃO

salvar\_dados()

exibir\_mensagem("Saindo do sistema...")

SENÃO

exibir\_mensagem("Opção inválida!")

FIMSE

ATÉ opcao == 0

FIM

FUNÇÃO cadastrar\_produto()

EXIBIR "Informe o código do produto: "

LER codigo

EXIBIR "Informe o nome do produto: "

LER nome

EXIBIR "Informe o preço do produto: "

LER preco

EXIBIR "Informe a categoria (1-Bebida / 2-Petisco): "

LER categoria

EXIBIR "Informe o ponto mínimo de estoque: "

LER ponto\_minimo

// Verifica se já existe o produto

SE produto\_existe(codigo) ENTÃO

EXIBIR "Produto já cadastrado!"

SENÃO

adicionar\_produto(codigo, nome, preco, categoria, ponto\_minimo)

EXIBIR "Produto cadastrado com sucesso!"

FIMSE

FIMFUNÇÃO

FUNÇÃO registrar\_pedido()

EXIBIR "Número da mesa: "

LER mesa

criar\_pedido(mesa)

REPITA

EXIBIR "Código do produto (0 para encerrar): "

LER codigo

SE codigo == 0 ENTÃO

SAIR\_DO\_LOOP

SENÃO

EXIBIR "Quantidade: "

LER qtd

SE verificar\_estoque(codigo, qtd) == FALSO ENTÃO

EXIBIR "Estoque insuficiente!"

SENÃO

adicionar\_item\_ao\_pedido(mesa, codigo, qtd)

FIMSE

FIMSE

ATÉ FIM\_DO\_PEDIDO

FIMFUNÇÃO

FUNÇÃO fechar\_pedido()

EXIBIR "Informe o número da mesa a fechar: "

LER mesa

total <- calcular\_total\_pedido(mesa)

EXIBIR "Total da conta: ", total

EXIBIR "Confirmar fechamento? (S/N): "

LER resp

SE resp == 'S' OU resp == 's' ENTÃO

atualizar\_estoque(mesa)

registrar\_venda(mesa, total)

remover\_pedido(mesa)

EXIBIR "Pedido fechado com sucesso!"

SENÃO

EXIBIR "Operação cancelada."

FIMSE

FIMFUNÇÃO

FUNÇÃO gerar\_relatorio()

EXIBIR "1 - Vendas do dia"

EXIBIR "2 - Produtos abaixo do ponto mínimo"

EXIBIR "3 - Itens mais vendidos"

LER opcao

SE opcao == 1 ENTÃO

listar\_vendas\_dia()

SENÃO SE opcao == 2 ENTÃO

listar\_produtos\_baixo\_estoque()

SENÃO SE opcao == 3 ENTÃO

listar\_itens\_mais\_vendidos()

SENÃO

EXIBIR "Opção inválida!"

FIMSE

FIMFUNÇÃO

Fluxograma :

