



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
CURSO DE BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

Relatório da Unidade 1

Pedro Lucas Barbosa Nascimento

Natal/RN, 2025

Introdução

Este projeto tem como objetivo realizar vendas e gestão de estoque, simulando operações básicas de um varejo, facilitando o trabalho do vendedor, permitindo que ele adicione os produtos e some o valor total, aumentando a eficiência, controle e precisão. Intitulei o projeto de “Sistema de Vendas Super Conveniência 24” como exemplo, já que o ele poderia ser utilizado tanto em mercados e mercearias quanto em lojas de conveniência.

O sistema de vendas resolve um problema fundamental que é o controle manual de estoque e de vendas, que é propenso a erros como um cálculo feito de forma incorreta, por exemplo. Em uma loja de conveniência, a precisão é essencial, já que vários clientes podem ser atendidos em um pequeno intervalo de tempo, e esta estrutura asseguraria isso.

Escolhi o tema do projeto pensando em questões pessoais e de aprendizado. Não tive qualquer experiência com programação antes do curso, e achei que um sistema de vendas iria me ajudar a praticar funções, vetores, cálculos e outras aplicações importantes para um profissional de tecnologia da informação. Além disso, quis ter uma base para praticar e talvez até utilizar o sistema de alguma forma na conveniência de meu pai. Atualmente, o controle por lá é realizado de forma manual, por essa razão, sei o quão difícil é fazer a gestão de vendas dessa forma. Nesse caso, um sistema facilitaria muito, e entender como ele funciona é fundamental para nós.

Neste relatório, iremos refletir sobre como os conceitos da primeira unidade foram aplicados no projeto e como seu processo foi desafiador, mas, ao mesmo tempo, frutífero para o aprendizado.

Análise Técnica

Metodologia: O projeto foi desenvolvido na linguagem C, sendo o código-fonte escrito no editor Visual Studio Code. A compilação e execução foram realizadas através do sistema operacional Windows 11, utilizando o compilador GCC (GNU Compiler Collection).

Aplicações dos Conceitos da U1: Os conceitos da primeira unidade foram aplicados intensivamente no projeto, garantindo a funcionalidade do código.

Condicionais:

1. **Menu Principal (switch):** O switch-case foi usado no main para direcionar a execução do programa para a função apropriada com base na Opção escolhida pelo usuário (Listar Estoque, Realizar Venda, Sair).
2. **Validação de Venda (if/else):** Dentro da função realizar_desconto e no loop principal, condicionais foram usadas para:
Busca de Produto: if (indice_produto != -1) – Garante que a transação só prossiga se o ID digitado realmente existir no vetor estoque.
Limpeza de Buffer: Condicionais robustas foram usadas para tratar entradas inválidas (scanf falho), prevenindo loops infinitos.

Repetições:

3. **Iteração de Vetor (for):** O loop for foi implementado nas funções que precisam percorrer o vetor de produtos, por exemplo, “!listar_estoque”, que itera de $i = 0$ até $\text{num_produtos} - 1$ para exibir cada item do estoque.
4. **Controle do Sistema (do-while):** O loop do-while principal no main mantém o programa em execução e exibe o menu até que o usuário escolha a opção de sair ($\text{opcao} \neq 0$).
5. **Carrinho de Compras (while):** O loop “while (continuar == 1)” dentro do case 2 é a implementação mais dinâmica, permitindo ao vendedor adicionar múltiplos produtos a uma única transação até decidir finalizar a compra.

Vetores:

6. **Estoque como Vetor de structs:** O vetor “estoque[CAPACIDADE_MAXIMA]” armazena o conjunto de produtos. Cada elemento do vetor é uma estrutura (Produto) completa, contendo ID, nome, preço e quantidade.
7. **Acesso e Manipulação:** O acesso aos dados é feito pelo índice, encontrado através da função “buscar_indice_por_id.” A manipulação (desconto da quantidade) é feita diretamente na posição correta do vetor.

Funções:

main(): Gerencia o loop principal e o menu. Age como um controlador que delega tarefas.

menu_principal(): Responsável apenas por exibir as opções na tela.

inicializar_estoque(): Isola o código de setup inicial, pré-carregando o vetor estoque.

listar_estoque(): Isola a lógica de impressão formatada dos dados do estoque.

buscar_indice_por_id(): Isola a lógica de busca sequencial. Essencial para o dinamismo do PDV.

realizar_desconto(): Isola a lógica crítica de validação de estoque, desconto da quantidade e cálculo do subtotal.

Estrutura de Dados:

A principal estrutura de dados utilizada foi a “struct Produto” e o vetor de “structs”.

struct Produto: Define o tipo de dado complexo necessário para representar um item da loja, agrupando id, nome, preço e quantidade em uma única entidade.

Variáveis de Controle: Variáveis simples como “num_produtos” (inteiro que rastreia o número real de itens no vetor) e “total_compra” (flutuante que acumula o valor da transação) foram cruciais para o controle do sistema.

Implementação e Reflexão

Encontrei várias dificuldades ao longo do caminho. Além de não ter experiência nenhuma com programação antes do curso, eu tive muitos problemas com digitação e execução do código, que certamente é comum. Por exemplo, o problema do loop infinito que aconteceu quando o `scanf` falhou ao ler um número (se o usuário digitava um comando inválido) e começava a repetir o menu sem parar, então implementei um mecanismo de validação de entrada que também “joga fora” a entrada inválida, só aceitando os números que o usuário é instruído a digitar. Outra “pedra no meu sapato” foi o sistema de carrinho, que pergunta se o usuário quer adicionar mais algum item e soma o total dos valores do produto antes de concluir a venda. Criei a função “`buscar_indice_por_id`” para que o programa pudesse encontrar o produto e exibir o nome dele envolvi toda a lógica de venda dentro de um “`while (continuar == 1)`” no case 2 do main. Por fim, usei a variável “`total_compra`” para somar o valor de cada item vendido (o `subtotal_item`), o que era o ponto central do requisito do carrinho.

Desenvolver um sistema de vendas foi um pouco complicado, tendo em vista que ele envolve várias operações e funções em sua estrutura. Entretanto, é um dos mais eficazes para quem tem interesse em praticar programação. No começo eu achei que não ia dar certo, vi várias coisas dando errado e pensei em mudar de tema várias vezes, mas continuei trabalhando no código aos poucos, buscando na internet, tirando algumas dúvidas em sala e, principalmente, tendo as listas como base. As listas me fizeram ter um dos meus primeiros contatos com a programação além do “Olá, Mundo”, que eu nunca nem tinha executado com êxito, a propósito. Mas essa experiência do projeto me fez aprender bastante sobre programação, mais ainda do que eu esperava.

Para este projeto, vejo como melhorias futuras a implementação de uma função que cadastre novas mercadorias, para que o estoque não fique tão limitado. Além disso, uma opção que carregue e salve o estoque no computador para que o sistema se lembre do estoque atual quando o usuário abrir também seria uma função muito útil sendo integrada ao código.