



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

INSTITUTO METRÓPOLE DIGITAL

Pedro Ruan Cavalcante Nunes

**RELATÓRIO TÉCNICO SOBRE INTRODUÇÃO ÀS PRÁTICAS DE
PROGRAMAÇÃO DA UNIDADE 2**

Natal, RN

29/11/2025

Pedro Ruan Cavalcante Nunes

**RELATÓRIO TÉCNICO SOBRE INTRODUÇÃO ÀS PRÁTICAS DE
PROGRAMAÇÃO DA UNIDADE 2**

Relatório técnico científico apresentado ao curso Bacharelado de Tecnologia da Informação Universidade Federal do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial para a aprovação da disciplina: Introdução às práticas de programação, com foco na 2º unidade da disciplina.

Orientador: Fernando Marques Figueira Filho.

Natal, RN

29/11/2025

RESUMO

Desenvolveu-se um programa em linguagem C com foco na aplicação de conceitos fundamentais da programação estruturada, como o uso de variáveis com tipos definidos, vetores, operações aritméticas, comandos condicionais e de repetição, além da implementação de funções. O sistema tem como objetivo registrar produtos, validar os dados inseridos e calcular o valor total com base no preço unitário e na quantidade. A estrutura modular do código promove organização e reutilização, demonstrando a aplicabilidade da programação na resolução de problemas cotidianos.

Palavras-chave: Programação, Linguagem C.

SUMÁRIO

1. INTRODUÇÃO.....	5
2. DESENVOLVIMENTO.....	5
2.1. STRINGS.....	6
2.2. ESTRUTURAS DE REPETIÇÕES ANINHADAS.....	7
2.3. MATRIZES.....	7
2.4. PONTEIROS BÁSICOS.....	7
2.5. ALOCAÇÃO DINÂMICA BÁSICA.....	8
3. CONCLUSÃO.....	8

1. INTRODUÇÃO

Dando continuidade ao projeto da primeira unidade, mantive o foco na melhoria do Sistema de Gerenciamento de Compras (itens no geral), que tem como finalidade simular um sistema focado em CRUD (Create, Read, Update e Delete). Inicialmente retratando uma lista com os comandos necessários para realizar cada operação, deixando a livre escolha do usuário o manuseio do sistema. O principal objetivo para manter esse projeto, tendo em vista que é um aprimoramento da unidade 1, é feito com o intuito de auxiliar o manuseio da informação dos produtos e, conseqüentemente facilitando o dia-a-dia de um estabelecimento.

2. DESENVOLVIMENTO

Durante o desenvolvimento desse projeto o principal sistema operacional utilizado foi o Subsistema de *Linux* para *Windows* (*Ubuntu*), devido à facilidade de compilação e instalação de futuros componentes que serão integrados ao projeto. Além disso, foi utilizado como principal ambiente de programação o *Visual Studio Code*, ligado diretamente ao *GitHub*, mantendo atualizações constantes de versões via *commits* e um repositório público online para o acompanhamento das atualizações.

Como mencionado anteriormente, mantive o desenvolvimento desse projeto no sistema de Subsistema de *Linux* para *Windows* (*Ubuntu*). Mantive também o principal ambiente de programação do *Visual Studio Code*, ligado diretamente ao *GitHub*, mantendo atualizações constantemente. Além disso, realizei a criação de uma nova biblioteca com *GTK* (*Gimp Toolkit*), uma biblioteca *Open Source* focado na parte de interface gráfica que foi um dos pontos de atualização dessa unidade. Como última atualização, realizei a criação de um *Makefile* que realiza a configuração de ambas as bibliotecas necessárias para

configuração do ambiente além de rodar tanto o sistema via linha de comando, quanto pela interface gráfica, de forma prática, simples e otimizada tendo como o comando *“make help”* o conjunto de todos os comandos e suas respectivas funções.

Com relação a principal dificuldade encontrada, obtive diversos bugs e problemas com o código, tendo em vista que utilizei typedef struct para cada criação de novo produto. Dessa forma toda vez que um produto novo era adicionado esse array de produtos crescia também e com isso, muitas vezes ao excluir produtos, ele acabou por não listar certo. Além disso, com a integração da interface gráfica, foi necessário destrinchar o código e separar cada função em arquivos diferentes, dessa forma tomando bastante tempo e problemas durante a compilação e testes de funcionalidade. Pelo fato de não haver experiências com interface gráfica em C, apresentei dificuldades na criação dos componentes e na parte de integração do backend com o frontend, assim fiz diversas buscas na web e com IA , para ajudar a cumprir o objetivo final.

2.1. STRINGS

O programa anteriormente já utilizava strings principalmente para criação de nomes de produtos, porém com o avanço do desenvolvimento do código, vi a necessidade de editar um produto, por exemplo, atualizar o espaço de memória onde ele estava localizado para passar os novos parâmetros de produtos, e com isso, utilizei a função *strncpy()*, passando como parâmetro o identificador do produto e os dados que serão sobrescritos posteriormente, dessa forma sendo necessário também a biblioteca de *<string.h>* para implementar essas funções.

2.2. ESTRUTURAS DE REPETIÇÕES ANINHADAS

Com relação às estruturas de repetição aninhadas, no código em si não foi necessário a implementação de estruturas aninhadas, sendo necessário apenas a utilização de um `for` simples, por exemplo, para a deleção de um determinado produto que é identificado pelo seu respectivo ID, assim realocando todos os outros elementos dentro do array de produtos para um index anterior. Por se tratar de um sistema de CRUD não seria necessário processamentos entre múltiplos elementos simultaneamente.

2.3. MATRIZES

O sistema de gerenciamento de produtos em si utiliza apenas um estrutura de dados unidimensional, assim armazenando uma lista linear de produtos, assim cada produto é um entidade independente, além das operações serem realizadas sobre produtos individuais ou sobre a lista de produtos como um todo.

2.4. PONTEIROS BÁSICOS

O código em si utiliza de ponteiros para uma das principais funções que seria a de edição de produtos, com a variável `*char`, ligada diretamente ao ponteiro da memória alocada. Ao se tratar de uma edição, é necessário modificar diretamente o espaço de memória alocado no Struct de produtos, para modificá-lo. Além de ser utilizada uma variável *temp*, onde é alocado um ponteiro específico para os produtos, sendo um armazenamento temporário, e com isso diversas outras funções foram feitas e implementadas.

2.5. ALOCAÇÃO DINÂMICA BÁSICA

O código faz uso de alocação dinâmica, iniciando sempre com pelo menos 10 espaços de capacidade para cadastro de produtos, e sempre que esse limite é encontrado, a capacidade é dobrada, assim crescendo dinamicamente. Além disso, ao fim do programa, é utilizada a função *free()* para liberar a memória alocada, assim evitando que os bytes alocados fiquem salvos, assim limpando completamente e reiniciando o programa.

3. CONCLUSÃO

O programa *produto.c* integra os principais conceitos abordados na Unidade 2, abrangendo desde o uso de Strings, até a implementação de memória dinâmica para que assim, não haja um limite para a quantidade de produtos a serem adicionados. Ele representa um exemplo prático de como os fundamentos da programação e a introdução às práticas de programação podem ser aplicados para resolver problemas cotidianos de forma eficiente e organizada. O projeto utiliza uma situação comum — como o ato de ir ao supermercado — para ilustrar como a programação pode ser usada para automatizar e simplificar tarefas do dia a dia, demonstrando sua utilidade na solução de problemas reais, principalmente com a principal adição da interface gráfica que facilita mais ainda para o usuário o que deve ser feito.