

HÉRICLES VITÓRIO GOMES DA COSTA

SISTEMA DE GERENCIAMENTO DE BIBLIOTECA EM C

Relatório técnico apresentado como requisito parcial para unidade I da disciplina de Introdução às Técnicas de Programação, do curso de Bacharelado em Tecnologia da Informação da Universidade Federal do Rio Grande do Norte.

Professor: Fernando Marques Figueira Filho.

Natal, Rio Grande do Norte
2025

1 INTRODUÇÃO E CONTEXTO

Este relatório técnico descreve o processo de criação e o funcionamento do projeto "Sistema de Gerenciamento de Biblioteca". O programa foi desenvolvido na linguagem C e tem como objetivo implementar as funcionalidades básicas para o controle de um acervo de livros.

1.1 Nome do projeto e objetivo

- **Nome do projeto:** Sistema de Gerenciamento de Biblioteca em C.
- **Objetivo Geral:** Desenvolver um sistema simples para gerenciar o acervo de uma pequena biblioteca, permitindo que o usuário realize as operações essenciais de controle dos livros.
- **Objetivos Específicos:**
 - Criar uma função para cadastrar novos livros no sistema.
 - Desenvolver uma função para remover livros existentes do acervo.
 - Implementar uma função para listar todos os livros cadastrados.
 - Construir um menu interativo para que o usuário possa escolher facilmente as operações.

1.2 Problema que o projeto resolve

O controle de livros em pequenas bibliotecas, coleções pessoais ou salas de aula muitas vezes é feito de forma manual, com cadernos ou planilhas. Esse método pode levar a erros, perda de informações e torna difícil a consulta rápida de títulos. O projeto busca resolver isso com uma ferramenta digital simples, que automatiza o cadastro, a remoção e a listagem de livros, trazendo mais organização e agilidade para a gestão do acervo.

1.3 Justificativa da escolha do projeto

A escolha de um sistema de biblioteca foi ideal para aplicar na prática os conceitos de programação em C vistos em aula. O desenvolvimento do projeto exigiu o uso de vetores para guardar a lista de livros, laços de repetição para fazer o menu funcionar e para listar os itens, estruturas condicionais para processar as opções do usuário e a separação do código em diferentes funções. Assim, o projeto serviu como um ótimo exercício para consolidar o aprendizado da disciplina.

2 ANÁLISE TÉCNICA

Esta seção apresenta os detalhes técnicos do projeto, desde as ferramentas usadas até a forma como a lógica e os dados foram estruturados.

2.1 Metodologia

O projeto foi desenvolvido no sistema operacional Windows 10, utilizando o Windows Subsystem for Linux (WSL) com a distribuição Ubuntu para compilação e execução. O código foi escrito no editor Visual Studio Code (VS Code), e o compilador utilizado foi o GCC (GNU Compiler Collection), acessado pelo terminal do Ubuntu.

2.2 Aplicação dos Conceitos

- **Uso de Estruturas Condicionais:** As estruturas condicionais foram essenciais para controlar o que o programa faz. A estrutura `switch-case` foi usada para criar o menu principal, de modo que, quando o usuário digita uma opção, o programa chama a função correta (cadastrar, remover, etc.). Além disso, a estrutura `if` foi usada para fazer verificações importantes, como:
 - Verificar se a biblioteca está cheia antes de tentar cadastrar um novo livro.
 - Verificar se existe algum livro na lista antes de tentar remover ou listar, avisando o usuário caso o acervo esteja vazio.
- **Lógica das Estruturas de Repetição:** Os laços de repetição foram usados para duas coisas principais. Um laço `do-while` mantém o menu principal sempre ativo, voltando a ser exibido após cada operação, até que o usuário escolha a opção "Sair". Outro laço, o `for`, foi usado dentro da função de listar, para passar por cada posição do vetor e mostrar na tela os livros cadastrados. Um laço `for` também foi a solução para reorganizar o vetor na hora de remover um livro.
- **Aplicação de Vetores:** O vetor foi a principal forma de armazenar os dados do programa. Foi criado um vetor chamado `biblioteca`, que funciona como uma lista temporária para guardar os livros. Ao cadastrar um livro, seu título é salvo em uma posição vazia desse vetor. O sistema foi configurado para gerenciar uma coleção de até 5 livros, mas essa capacidade pode ser facilmente alterada no arquivo "biblioteca.h".
- **Organização das Funções:** O código foi dividido em funções para ficar mais organizado e fácil de entender. Além da função `main`, que inicia o programa e controla o menu, foram criadas três outras funções:
 - `cadastarLivros()`: Pede os dados do livro ao usuário e o salva no vetor.

- `listarLivros()`: Mostra na tela todos os livros que estão no vetor.
- `removerLivros()`: Pergunta qual livro o usuário quer remover e o retira do vetor.

Essa separação ajuda muito, pois se um erro acontecer, é mais fácil encontrar o problema em uma função específica sem afetar o resto do programa.

2.3 Estruturas de Dados

- **Vetor `biblioteca`**: A estrutura principal do projeto. É um vetor onde cada espaço guarda as informações de um livro.
- **Variável `totalLivros`**: Uma variável inteira que funciona como um contador. Ela sabe quantos livros estão cadastrados no momento, ajudando a saber onde adicionar o próximo livro e até onde o laço de listagem deve ir.
- **Variáveis Auxiliares**:
 - `opcao`: Guarda o número que o usuário digita no menu.
 - `indice`: Guarda o número do livro que o usuário escolhe para remover.

3 IMPLEMENTAÇÃO E REFLEXÃO

Esta parte final comenta sobre a experiência de desenvolver o projeto, os desafios encontrados, as soluções e os aprendizados.

3.1 Dificuldades Encontradas

O maior desafio técnico foi, sem dúvida, a função de remover um livro. Acertar a lógica do laço `for` para que ele reorganizasse o vetor corretamente, sem pular livros ou tentar acessar uma posição que não existe, exigiu bastante teste e raciocínio.

Outra dificuldade foi se adaptar à forma como a linguagem C lida com vetores. Em outras linguagens, muitas vezes existem funções prontas para adicionar ou remover itens de uma lista. Em C, foi preciso criar essa lógica do zero, controlando manualmente os índices e o espaço disponível, o que foi um aprendizado importante.

3.2 Soluções Implementadas

Para resolver o problema da remoção, a solução foi pensar no processo passo a passo. Quando um livro é removido, um "buraco" aparece no vetor. Para fechar

esse buraco, criei um laço `for` que começa na posição do livro removido e copia cada livro seguinte para a posição anterior (`vetor[i] = vetor[i+1]`). Depois que o laço termina de "empurrar" todo mundo, eu simplesmente diminuo o valor da variável `totalLivros`. Assim, o programa passa a ignorar a última posição, que agora é um dado repetido, e o livro é efetivamente removido.

Para a dificuldade geral com vetores, a solução foi realmente estudar e praticar a lógica de programação. Controlar cada etapa do processo manualmente foi desafiador, mas me deu uma compreensão muito maior sobre como os dados são organizados na memória.

3.3 Organização do Código

A decisão de separar o código em funções foi para deixar tudo mais organizado. A ideia era que cada função tivesse um único trabalho: uma para cadastrar, uma para listar e outra para remover. Essa escolha se justificou por deixar o código:

- **Mais fácil de ler:** A função `main` funciona como um resumo do que o programa faz.
- **Mais simples de corrigir:** Se o cadastro desse erro, eu saberia que o problema estava na função `cadastrarLivros()`, sem precisar mexer no resto.
- **Mais fácil de reutilizar:** No futuro, essas mesmas funções podem ser aproveitadas para outras versões do projeto.

3.4 Conclusão

Trabalhar com C pela primeira vez foi uma experiência de grande aprendizado. O projeto me forçou a pensar sobre gerenciamento de memória e manipulação de dados de uma forma muito mais direta do que em outras linguagens. No final, o resultado não foi só um programa funcionando, mas um entendimento mais profundo sobre os fundamentos da programação.

Assim, vendo o que é proposto atualmente e pensando no futuro, há várias melhorias que podem ser implementadas:

- **Melhorar os Dados dos Livros:** Em vez de salvar só o título, expandir o programa para salvar também o autor, ano de lançamento e gênero.
- **Criar um Sistema de Empréstimos:** Adicionar funções de "emprestar" e "devolver", com um status para cada livro (ex: "disponível" ou "emprestado").
- **Salvar os Dados em Arquivo:** Fazer o programa salvar a lista de livros em um arquivo, para que os dados não se percam quando ele for fechado.