



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – CAMPUS NATAL  
CURSO TECNOLOGIA DA INFORMAÇÃO**

**RELATÓRIO TÉCNICO : Sistema de Gerenciamento de Biblioteca**

**Aluno: Nelson Ryan Silva de Vasconcelos**

**Matrícula:20250062976**

**Disciplina: Introdução a Técnicas de Programação (2025.2)**

**Projeto da Unidade 1: Sistema de Gerenciamento de Biblioteca Pt.1**

**Professor: Fernando Marques Figueira Filho**

**Data: 07/10/2025**

**Natal - RN  
2025**

# 1. Introdução e Contexto

O presente relatório detalha o processo de desenvolvimento do projeto da Unidade 1 da disciplina de Introdução a Técnicas de Programação. O projeto consiste em um Sistema de Gerenciamento de Biblioteca, desenvolvido inteiramente na linguagem C. O objetivo principal do sistema é permitir o cadastro, a listagem, o empréstimo e a devolução de livros através da implementação de um código simples que irá ser complementado com outras funcionalidades ao longo das unidades.

O sistema foi concebido para auxiliar na organização de bibliotecas por meio de um programa que possa digitalizar processos, para o controle de pequenos acervos, automatizando o registro e o status de cada obra específica.. A escolha deste tema foi motivada pela possibilidade de aplicar grande parte dos conteúdos da primeira Unidade, (e futuramente das outras duas unidades) na implementação do código, como funções, vetores, laços e estruturas condicionais.

## 2. Análise Técnica

### 2.1. Metodologia e Ferramentas

Para a construção do projeto, foram utilizadas as seguintes ferramentas e tecnologias:

- **Linguagem de Programação:** C (Padrão C99).
- **Compilador:** GCC (versão 15.2.0) no sistema operacional Windows 11 Home.
- **Editor de Código:** Visual Studio Code.
- **Controle de Versão:** Git e GitHub.

### 2.2. Aplicação dos Conceitos da Unidade 1

Os conceitos fundamentais da unidade foram aplicados da seguinte forma:

- **Vetores:** A estrutura de dados principal do sistema foi implementada com vetores paralelos para armazenar as informações dos livros, como os títulos, seus autores, o ano de publicação e o status dos livros. Este método permitiu associar várias strings a esses vetores por meio de índices que indicam tanto a posição da string, quanto o número de caracteres que cada string pode assumir em cada posição. Dessa forma, as informações dos livros ficam armazenadas e podem ser acessadas através da numeração que cada livro possui.
- **Estruturas de Repetição:** Laços foram cruciais para a funcionalidade do sistema. Um laço do-while na função main garante que o menu de opções seja exibido repetidamente. O laço for, por sua vez, foi essencial na função listaLivros para exibir o acervo dos livros disponíveis, de acordo com suas numerações, representadas pelos índices 'i', além de contribuir para as substituições das quebras de linhas, para evitar bugs. Já o laço while, junto com a função getchar(), contribuiu para limpeza do buffer, também eliminando quebras de linhas.
- **Estruturas Condicionais:** A lógica do programa é controlada por if/else. Nas funções emprestimoLivros e devolucaoLivros, condicionais foram usadas para verificar se os livros do acervo estavam disponíveis para empréstimo, ou pendentes

de devolução, por meio da identificação do livro fornecida pelo usuário, sendo feitas certas validações, dessa forma verificando o status do livro (disponível ou emprestado).

- **Funções:** O código foi modularizado em funções com responsabilidades claras: `cadastroLivros`, `listaLivros`, `emprestimoLivros` e `devolucaoLivros`. Essa separação é benéfica pois deixou o código mais organizado e a função `main` mais limpa, dando mais clareza sobre cada parte do código.

### 3. Desafios, Soluções e Organização

#### 3.1. Dificuldades Encontradas

Uma das principais dificuldades encontradas durante o desenvolvimento foi gerenciar o buffer de entrada do teclado, visto que estava acontecendo certos bugs no código. Inicialmente, a tentativa de usar `scanf` para leitura de strings acarretou em erros, visto que para ler palavras, essa função não é otimizada e gerava falhas quando eram usados espaços. Posteriormente, mesmo usando `fgets` (uma função mais otimizada para strings), ao se alternar o uso de `fgets` para leitura de texto e `scanf` para leitura de números, aconteciam leituras puladas.

#### 3.2. Soluções Implementadas

Para solucionar os desafios de entrada de dados, o método adotado foi padronizar o uso exclusivo da função `fgets` para leitura de todas as strings, visto que é uma função mais segura e otimizada. Para resolver o problema com as quebras de linhas '\n', deixadas pela função `scanf`, foi feito o uso da seguinte estrutura: `while (getchar() != '\n')`. Dessa forma, a função `getchar()` dentro desse laço, limpava o buffer de entrada, antes de novas leituras acontecerem, evitando falhas nas leituras posteriores.

#### 3.3. Organização do Código

O projeto foi estruturado em um único arquivo (`main.c`) seguindo uma ordem lógica: a biblioteca principal `stdio.h`, o uso de constantes - para padronização do código - o uso de variáveis globais, o uso de protótipo das funções, a função principal `main` e a implementação de cada função. A inserção dos protótipos das funções de início é de grande relevância, pois permite que a função `main`, bem como qualquer uma das outras funções, pudessem chamar umas às outras, evitando erros.

### 4. Respostas às Perguntas Orientadoras

- **Quais conceitos da Unidade 1 foram aplicados e onde?**
  - Estruturas de condições, como `if/else` foram usadas principalmente nas funções `emprestimoLivros` e `devolucaoLivros`, para verificação da disponibilidade de livros no acervo, além do `switch-case` que foi usado para determinar as escolhas do menu.

- Laços de repetições, como o for, foram implementados na listagem dos livros e para substituição de quebra de linhas, o do-while no menu de escolha, e o while junto com getchar(), para limpeza do buffer de entrada.
- Funções, que estruturam todo o código, cada uma com sua utilidade específica no sistema de biblioteca.
- Vetores, amplamente usados para armazenar informações sobre os livros, mais especificamente na função cadastroLivros.
- **Como a organização em funções facilita a manutenção do código?**
  - A organização do código em funções, favorece a manutenção, pois os possíveis erros que podem aparecer no código ficam mais individualizados. Exemplo: foi feito um cadastro de livros corretamente no programa, através da função cadastroLivro e uma listagem por meio da função ListaLivros(), mas na hora de usar a função emprestimoLivros(), acontece um erro que afeta outras partes do código. Muito provavelmente, a função de empréstimo está afetando o funcionamento do sistema e precisa ser corrigida.
- **Quais foram os principais desafios na implementação das estruturas de repetição?**
  - Um dos desafios foi estabelecer certos limites para armazenamento de livros nos vetores. Para isso foi criada a constante MAX\_LIVROS, e a variável global total\_livros, que definiam os limites dentro dos laços de repetição. A variável total\_livros foi o limite que garantiu que não fosse exibidos lixo de memória dentro dos vetores. Além disso, outro desafio foi usar os laços de repetição para substituição de '\n' por '\0', que não foi tão intuitivo no começo.
- **Como os vetores foram utilizados para resolver o problema proposto?**
  - Os vetores foram de extrema importância no código, pois permitiram armazenar os dados mais relevantes do sistema de biblioteca, como os títulos dos livros, seus autores, ano de publicação das obras e a sua disponibilidade.
- **Que melhorias poderiam ser implementadas nas próximas unidades?**
  - Uma grande melhoria seria a melhor manipulação de strings, para evitar problemas com o buffer de entrada. Outra melhoria seria a implementação de novas funcionalidades dentro das funções, para otimizar ainda mais o código. Ademais, a utilização de arquivos dentro desse sistema e uma maneira de deixar os dados salvos, é de extrema importância, visto que todos os dados são perdidos ao término do programa.

## 5. Conclusão

O desenvolvimento do Sistema de Gerenciamento de Biblioteca permitiu atingir os objetivos propostos para a Unidade 1, resultando em uma aplicação funcional que cumpre os requisitos básicos de controle de um acervo. O principal aprendizado obtido foi a importância de validar corretamente as entradas no programa, evitando bugs corriqueiros na linguagem C, e também implementar os conceitos aprendidos em aula em um projeto mais longo que, diferentemente das listas, vai amadurecendo ao longo das unidades. O projeto serviu como uma excelente base prática para os fundamentos da programação.