

Relatório Técnico – Unidade 1

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

Projeto: Sistema de Gerenciamento de Biblioteca em C

Aluno: Elton Gustavo da Silva Freitas

Matrícula: 20250024590

1. Introdução e Contexto

Este projeto implementa um **Sistema de Biblioteca** simples em linguagem C.

O objetivo é permitir o **cadastro, listagem e aluguel de livros** utilizando uma interface de linha de comando (CLI).

O problema que o projeto resolve é o **gerenciamento básico de empréstimos**, controlando quais livros estão disponíveis e quais já foram alugados.

A escolha desse projeto se justifica porque:

- É um projeto que consegue englobar todos os conceitos abordados na unidade 1 de forma clara e direta.
- A escolha do projeto foi feita porque ele possui uma complexidade intermediária, não sendo simples demais e nem excessivamente complicado.
- Tal dificuldade permitiu que os conceitos da linguagem C fossem aplicados de forma gradual e estruturada. Assim, mesmo com a falta de familiaridade inicial, foi possível desenvolver aprendizado consistente e progressivo ao longo da implementação.

2. Análise Técnica

2.1 Metodologia e Ferramentas

- **Compilador:** GCC (via WSL no sistema Windows 10).
- **Editor:** VS Code com extensão que permite acessar o sistema do WSL diretamente.
- **Execução:** Terminal do WSL (Ubuntu).

2.2 Conceitos Aplicados

- **Variáveis:** armazenamento dos títulos dos livros, anos de lançamento e nomes dos usuários.
- **Vetores (arrays):** `biblioteca[100]` armazena até 100 livros.
- **Condicionais (if/else):** usados para validar entradas, verificar a disponibilidade dos livros e nas escolhas do que os `printf()` iriam mostrar.
- **Laços de repetição:**
 - `while(1)` na main para manter o menu ativo até o usuário decidir fechar.
 - `for` para percorrer os livros cadastrados.
- **Funções criadas:**
 - `cadastrarLivro()` – adiciona um novo livro no array biblioteca a partir do modelo: struct livro, que possui os atributos **título**, **anoLancamento** e **nomePessoa**.
 - `listarLivros()` – exibe todos os livros cadastrados a partir de um for usando `printf()`.
 - `alugarLivro()` – permite que um usuário alugue um livro usando como entrada o nome da pessoa que irá alugar o livro e qual livro ela quer alugar baseado no índice dele no array biblioteca.
 - `main()` – controla o fluxo geral a partir de um menu infinito por `while(1)`, permite chamar todas as outras funções a partir de uma entrada numérica.

2.3 Estruturas de Dados

- **Struct livro:** guarda informações do livro e de quem alugou.
- **Strings (char[]):** armazena títulos e nomes.
- **Inteiro (int):** guarda o ano de lançamento.

3. Implementação e Reflexão

3.1 Dificuldades Encontradas

- A principal dificuldade encontrada esteve relacionada ao tratamento das entradas do usuário, em especial a combinação de `fgets` para leitura de strings e `scanf` para valores inteiros. A interação entre essas funções frequentemente resultava em comportamentos indesejados, como a captura indevida do caractere de nova linha (`\n`) ou a quebra no fluxo de leitura. Foi necessário, portanto, estudar e aplicar estratégias de organização das entradas, utilizando `getchar()` para limpar o buffer e garantir que os dados fossem lidos corretamente, evitando inconsistências durante a execução do programa.

3.2 Soluções Implementadas

- Uso de `strcspn` para remover o `\n` das entradas.
- Validação de entradas (não permitir título vazio ou ano inválido) a partir do Do-While.

3.3 Organização do Código

O código foi dividido em funções para facilitar a manutenção e melhorar a clareza. Cada função tem uma responsabilidade clara (cadastrar, listar, alugar).

3.4 Conclusão

O projeto permitiu aplicar todos os conceitos da **Unidade 1**: variáveis, vetores, condicionais, laços e funções.

Aprendizados:

- Manipulação de strings em C é mais trabalhosa que em outras linguagens.
- A organização em funções facilita muito a manutenção.
Possíveis melhorias:

3.5 Possíveis Melhorias

- **Função de devolução de livros** – permitir que um usuário devolva um livro já alugado, liberando-o novamente para empréstimo.
- **Persistência em arquivo ou banco de dados** – salvar e carregar os dados da biblioteca em um arquivo ou diretamente no banco de dados SQL, evitando que os cadastros sejam perdidos ao encerrar o programa.

Perguntas Orientadoras (responda no relatório):

Quais conceitos da Unidade 1 foram aplicados e onde?

Usei variáveis para guardar informações como título e ano dos livros, vetores para armazenar a lista de livros, condicionais para verificar se um livro já estava alugado ou não e repetições para manter o menu rodando. Também criei funções separadas para organizar melhor o código.

Como a organização em funções facilita a manutenção do código?

Facilita porque cada parte do programa fica responsável por uma coisa só. Se der problema em cadastrar livros, por exemplo, eu só preciso olhar a função `cadastrarLivro()`, sem mexer no resto do código.

Quais foram os principais desafios na implementação das estruturas de repetição?

O mais complicado foi organizar o menu principal usando o `while(1)` e cuidar das quebras de loop, principalmente para evitar que o programa travasse ou não aceitasse certas entradas.

Como os vetores foram utilizados para resolver o problema proposto?

Os vetores serviram para guardar todos os livros cadastrados na biblioteca. Assim, consigo percorrer o array e mostrar os livros, verificar quais estão alugados e até atualizar quando alguém pega um.

Que melhorias poderiam ser implementadas nas próximas unidades?

É possível adicionar a função de devolver livros, salvar os dados em arquivo ou diretamente no banco de dados, fazer busca dos livros por palavras chave entre outras pequenas funcionalidades que deixariam o projeto mais completo.