

## **Algoritmo e Estruturas de Dados I**

### **Prof. Bruno Matheus**

### **Trabalho I – Sistema de Biblioteca**

#### **I) Integrantes:**

- Victor Hugo de Lima Grecca, Número USP: 10392185
- Pedro Pastorello Fernandes, Número USP: 10262502

#### **II) Descrição:**

O primeiro trabalho da disciplina de Algoritmos e Estruturas de Dados I envolve o desenvolvimento de um Sistema de Biblioteca, utilizando as estruturas de Pilhas, Filas, Listas e Bancos de Memória. O trabalho foi desenvolvido em duas versões paralelas, como requerido: completamente estático e completamente dinâmico.

Seu funcionamento é o de um sistema de biblioteca, que executa as principais funções que uma biblioteca precisa para manter uma organização sobre seus dados. A interface é por linha de comando, contudo as mensagens foram pensadas para tornar a interação mais fácil possível. Foram contempladas todas as operações requeridas, além de adicionadas algumas poucas e simples, necessárias para o funcionamento do código (como interface e manipulação de buffer de entrada).

As funções de Adicionar Aluno ou Livro servem para inserir os dados desses elementos na lista de dados do Sistema, da mesma forma que Remover Livro ou Aluno serve para apagar os dados desses elementos da lista. A Função de Retirar Livro coloca o livro requisitado na fila do Aluno, e a função de Devolver Livro retira esse livro da fila do aluno e o coloca de volta a sua fila. A função de Mostrar Mensagens funciona como um painel de mensagens e avisos ao aluno, é através dessas mensagens que o aluno sabe que o livro foi retirado ou devolvido com sucesso, pois cada ação gera uma nova mensagem em sua caixa de mensagens pessoal. Finalmente, temos uma função que mostra o Menu quantas vezes precisar e uma última função para sair do programa, caso as ações do dia ou do momento estejam finalizadas.

O desenvolvimento inicial do trabalho foi feito na versão dinâmica. Então, uma cópia da mesma foi feita e as alterações necessárias foram feitas para tornar esta uma versão estática.

Para alterar a versão que será utilizada, deve-se seguir as instruções em comentários no arquivo principal (Main.c) e comentar o que for necessário no próprio arquivo.

O código está completamente comentado e indentado e, apesar do tamanho dos arquivos “Biblioteca\_Estatico.c” e “Biblioteca\_Dinamico.c”, o código é facilmente legível e compreensível.

### III) Apresentação das estruturas de dados e notas sobre as versões:

- **Versão Dinâmico:**

- Na versão dinâmica, as funções de busca retornam ponteiros para o tipo de elemento buscado (livro ou aluno).
- Não existem limites estipulados para o tamanho das estruturas de dados (exceto para strings, que é de 100 caracteres). Como o sistema é dinâmico, teoricamente, deve ser possível adicionar infinita quantidade de elementos.
- Não existem variáveis de erro na versão dinâmica, os casos de erro são conferidos quando o retorno de uma função é nulo.

- **Versão Estática:**

- Na versão estática, as funções de busca retornam inteiros equivalentes ao índice do elemento buscado no vetor da estrutura de dados que está armazenado.
- Os limites para tamanho de todas as estruturas de dados pode ser alterado pelo arquivo “Defines.h” e, por padrão, a string está em 50 e todo o resto em 10.
- Variáveis de erro foram utilizadas.

- **Arquivos:**

O sistema é composto por diversos arquivos .c e .h, organizando diversos TADs. No total, o Sistema de Biblioteca Dinâmico possui 18 arquivos na versão dinâmica e 19 na versão estática. Essa diferença se dá pelo arquivo “Defines.h”, que só é necessário na versão estática.

### Estruturas de dados utilizadas durante o projeto:

**Fila:** Uma fila foi implementada, como requerido, nas filas de espera dos livros.

**Pilha:** Uma pilha foi implementada, como requerido, para manipular a ‘caixa de mensagens’ dos alunos.

**Listas:** Foram utilizadas duas listas encadeadas, uma para armazenar os dados de livros, e outra para armazenar os dados dos alunos. A lista encadeada foi utilizada nestes casos pois nem sempre sabe-se qual a posição do elemento a ser adicionado ou removido à estrutura de dados (ao contrario de filas e pilhas).

**Banco de Memória:** Foi necessário a implementação de bancos de memória na versão estática do trabalho. Foram utilizados dois bancos de memória, um para a lista de livros e outro para a lista de alunos.

**Strings:** Dado a ausência do tipo de dado nas bibliotecas padrão da linguagem C , foi criado, por questões de praticidade e abstração de código, um TAD de manipulação de strings.

#### **IV) Complexidade do sistema:**

O sistema de biblioteca feito por nós possui complexidade  $O(n)$ , pois as rotinas mais complexas (rotinas de busca) necessitam de diversas comparações encadeadas em laços para encontrar o elemento buscado. Algumas seriam possíveis de se implementar em complexidade  $O(1)$ , porém não todas, o que torna esse esforço desnecessário pois não apresentará mudança na complexidade total do programa.