

Professor Mauricio Begnini

## Trabalho Final

Desenvolver um sistema comercial para algum estabelecimento fictício. É necessário usar no mínimo **três estruturas** (structs) e **três arquivos** para armazenar os dados necessários.

Uma das estruturas deve ser de **ligação/transação** entre as duas outras estruturas (vendas, compras, empréstimos ...).

Deve conter funcionalidades de **inclusão, listagem, busca, edição** para as **duas estruturas bases**, e funcionalidade de **inserção, listagem e listagem filtrada** para a **estrutura de transações**.

**Ex: Sistema de biblioteca:**

**Estrutura livro**, que armazena número de identificação, título, edição, autor, número de páginas;

**Estrutura usuário**, que armazena número de identificação, nome, cpf, telefone;

**Estrutura empréstimo**, que armazena número de identificação de um usuário, número de identificação de um livro, e data de empréstimo.

O programa deve ter as funcionalidades de **incluir livros, listar todos os livros existentes, buscar por um livro por número de identificação, alterar informações de um livro buscado por número de identificação**;

O programa também deve ter funcionalidades de **incluir usuário, listar todos os usuários existentes, buscar por um usuário por número de identificação, alterar informações de um usuário buscado por número de identificação**;

O programa também deve fornecer uma opção de **empréstimo de livro**, que deve ler o **número de identificação do livro, número de identificação do usuário** e a **data**. Para que um empréstimo ocorra tanto o **usuário quanto o livro devem existir**. O programa ainda deve ser capaz de **mostrar todos os empréstimos que já ocorreram, todos os empréstimos que um usuário específico já fez, e todos os empréstimos para um livro específico**.

O programa deve contar com **três arquivos**, um de **usuários**, um de **livros** e um de **empréstimos**. O programador pode escolher uma entre duas alternativas:

(i) Toda vez que o programa é **iniciado todas as informações dos arquivos devem ser carregadas para três vetores**, um **vetor de usuários**, um **vetor de livros** e um **vetor de empréstimos**. Quando o programa for **encerrado o conteúdo dos três vetores deve ser novamente salvo nos arquivos** (Os vetores podem ser de 50 posições cada).

(ii) **Toda operação que o usuário tentar fazer deve-se acessar o arquivo e modificar as informações diretamente no arquivo.**

\*Não é necessário que uma struct armazene uma referência para outra struct.