



UFOP

Universidade Federal
de Ouro Preto

Universidade Federal de Ouro Preto
Departamento de Ciências Exatas e Aplicadas

Curso: Engenharia de Computação
Disciplina: Algoritmo e Estrutura de Dados I
Professor: Rafael Alexandre
Aluno: Marcelo Vinícios Melo Martins

Sistema Biblioteca

O programa foi escrito usando IDE Dev-C++ em ambiente Windows.

Para criação do Sistema Bibliotecário, foram utilizadas estruturas¹ e funções², pois esse tipo de dados permite o agrupamento tornando mais legível sua manipulação, e portanto foi útil na elaboração do algoritmo.

```
//Estrutura Aluno
struct Aluno
{
    int matricula;
    char nome[50];
    char logradouro[100];
    int numero;
    char CEP[10];
    char cidade[50];
    int historico;
};

//Estrutura Exemplares
struct Exemplares
{
    int codigolivro;
    int emprestado; // 1 = não disponível . 0 = disponível
    int alunoresponsavel; //Guarda a matrícula do aluno
};

//Estrutura Livro
struct Livro
{
    codigo;
    char nome[50];
    char editora[50];
    int quant_livro;
    Exemplares listaDeExemplares[10];
};
```

¹Uma estrutura agrupa várias variáveis numa só.

²Uma função é um bloco de código de programa que pode ser usado diversas vezes em sua execução.

Essas estruturas, são simplesmente, as variáveis de armazenamento dos dados, ou seja, as listas usadas no programa.

Função criadas:

```
//Protótipo das Funções
int valida_matricula(struct Aluno *aluno, int mat);
int valida_livro(struct Livro *livro, int cod);
int valida_livro2(struct Livro *livro, int cod);
void exibe_aluno(struct Aluno *aluno, int mat);
void cadastro_aluno(struct Aluno *aluno);
void cadastro_livro(struct Livro *livro);
void exibe_livro(struct Livro *livro, int l1, int l2);
void zerolivro(struct Livro *livro);
void zero(struct Aluno *aluno );
int pesquisal(struct Livro *livro, int mat);
int pesquisae(struct Livro *livro, int mat);
void biblioteca();
void continuar(int opcao);
void emprestimo_livro(struct Aluno *a, struct Livro *l);
```

Sentiu-se a necessidade, para o perfeito funcionamento do sistema, que fossem criadas as funções acima que serão explicadas a seguir:

```
int valida_matricula(struct Aluno *aluno, int mat);
```

Essa função, realiza a validação, retornando o índice onde se encontra gravado a matrícula do aluno, caso não encontre será retornado um valor negativo. Os argumentos passados para função é a instância da estrutura aluno, e a matrícula digitada pelo usuário que será validada.

```
int valida_livro(struct Livro *livro, int cod);
```

Essa função retorna o índice aonde se encontra salvo o código do livro, caso não encontre nenhuma ocorrência será retornado um valor negativo que indica que não há nenhum livro com o código informado.

```
int valida_livro2(struct Livro *livro, int cod);
```

A função apresentada tem como princípio a mesma lógica da anterior, o que as distingue é o retorno, retornando a posição aonde se encontra o exemplar.

```
exibe_aluno(struct Aluno *aluno, int mat);
```

Essa função imprime os dados do aluno consultado, como matrícula do aluno, nome do aluno e seu endereço. A função recebe como argumento a instância da estrutura aluno, e a matrícula já validada.

```
void cadastro_aluno(struct Aluno *aluno);
```

Essa função realiza o cadastro do aluno, gravando as informações na estrutura Aluno para que posterior sejam resgatadas.

```
void cadastro_livro(struct Livro *livro);
```

Essa função realiza o cadastro dos livros como também de seus exemplares, o usuário digita a quantidade de exemplares e é gerado automaticamente seus respectivos códigos.

```
void exhibe_livro(struct Livro *livro, int l1, int l2);
```

Essa função imprime as informações do livro consultado como código, nome do livro e editora. A função recebe como parâmetro a instância da estrutura Livro, o código do livro e o código de seu respectivo exemplar.

```
void zerolivro(struct Livro *livro);
```

Tendo em vista alguns problemas com lixo de memória foi necessário a criação de uma função com proposito de inicializar código do livro, código do exemplar e situação que o livro se encontra, ou seja, emprestado ou disponível.

```
void zero(struct Aluno *aluno );
```

Pelo mesmos motivos já apresentados na função anterior foi necessário a inicialização da matrícula do aluno.

```
int pesquisal(struct Livro *livro, int mat);
```

Essa função tem como objetivo retornar o índice do livro aonde se encontra a matrícula do aluno responsável.

```
int pesquisae(struct Livro *livro, int mat);
```

A função apresentada retorna o índice do exemplar aonde encontra salvo a matricula do aluno que realizou o empréstimo.

```
void biblioteca();
```

Para tornar a implementação mais atraente do ponto de vista do usuário foi desenvolvida uma representação gráfica da palavra Biblioteca constituída apenas de símbolos.

```
void continuar(int opcao);
```

Um das prioridades ao escrever um *software* é a portabilidade do código, ou seja, a compilação para varias plataformas e no trabalho apresentado foi criado uma uma função que descarta a utilização do `system("PAUSE")` que é exclusivo da plataforma *Windows*. A função consiste que o usuário pressione a tecla enter para retornar ao menu principal, deixando na tela as infomação solicitadas.

```
void emprestimo_livro(struct Aluno *a, struct Livro *l);
```

Essa função, uma das mais importantes, ela altera a situação do livro, associa a matricula do aluno ao exemplar, incrementa o histórico que é a quantidade de empréstimos realizado por cada aluno.

Como Funciona

Na tela de apresentação, serão pedidas para o usuário, o cadastro de Alunos e Livros aonde serão gravadas as informações que o usuário informar, essas entradas são o requisito mínimo para o funcionamento do Sistema Bibliotecário.

Foram criadas 2 estruturas. A estrutura Aluno, armazenará todos os dados do aluno, a estrutura Livro, onde ficarão todos os dados do livro, existe ainda uma sub estrutura chamada de exemplares.

Depois que o usuário realiza o cadastro tanto do aluno quanto do livro pode-se efetuar o empréstimo aonde será requisitado a matrícula do aluno, será feita a validação, se a matrícula estiver correta será questionado o código do livro que deseja, se passar em todas as validações será impresso na tela uma mensagem de sucesso sendo que matrícula foi vinculada ao livro, a situação do livro mudará para emprestado, não sendo mais possível seu empréstimo até sua devolução.

Se o usuário preferir pode efetuar a devolução, desvinculando a matrícula do aluno do exemplar, basta informar o código do livro, será feita a validação e apagará a matrícula e a situação ficará disponível.

Existe também um menu para consulta de informações, o usuário pode pesquisar quais são os livros que estão com determinado aluno, o sistema pesquisará em qual exemplar tem gravado a matrícula do aluno e será impresso na tela.

Ainda no menu pode ser consultada informação de um livro, como qual aluno é responsável por aquele determinado exemplar.

Também permitida a consulta de quantos exemplares estão disponíveis por livro, quantos empréstimos cada aluno realizou.