**CENTRO PAULA SOUZA**

**ETEC DA ZONA LESTE**

Técnico de Desenvolvimento de Sistemas

**Gabriel Rodrigues Vieira Brandão, João Paulo Marque Manaresi, Felipe Silva do Nascimento**

**Projeto Integrado – Aplicação de Banco de Dados e Programação e Algorítmo: Estudo de Caso**

**São Paulo**

**2022**

**Gabriel Rodrigues Vieira Brandão, João Paulo Marques Manaresi, Felipe Silva do Nascimento**

**Projeto Integrado – Aplicação de Banco de Dados e Programação e Algorítmo: Estudo de Caso**

Trabalho de conclusão de semestre apresentado ao Curso Técnico em Desenvolvimento de Sistemas da Etec da Zona Leste orientado pelo Prof. José Eduardo Pimenta como requisito parcial para menção do 1º Módulo.

**São Paulo**

**2022**

**Sumário**

[**ÍNDICE DE FIGURAS** 4](#_Toc107164454)

[**1.** **INTRODUÇÃO** 6](#_Toc107164455)

[**2.** **Modelagem de Dados** 7](#_Toc107164456)

[**2.1.** **Lista Não Normalizada** 7](#_Toc107164457)

[**2.2.** **Primeira Forma Normal** 8](#_Toc107164458)

[**2.3.** **Segunda Forma Normal** 8](#_Toc107164459)

[**2.4.** **Terceira Forma Normal** 9](#_Toc107164460)

[**3.** **Diagrama Entidade Relacionamento** 10](#_Toc107164461)

[**3.1.** **Definição de DER** 10](#_Toc107164462)

[**3.2.** **Relações** 11](#_Toc107164463)

[**4.** **Modelo Entidade Relacionamento** 13](#_Toc107164464)

[**4.1.** **Definição de MER** 13](#_Toc107164465)

[**5.** **Desenvolvimento do Banco de Dados** 14](#_Toc107164466)

[**5.1.** **Tabelas** 14](#_Toc107164467)

[**5.2.** **Relações** 16](#_Toc107164468)

[**5.3.** **Relatórios** 17](#_Toc107164469)

[**5.4.** **Consultas** 22](#_Toc107164470)

[**5.5.** **Formulários** 26](#_Toc107164471)

[**6.** **Criação da Aplicação de Controle dos Empréstimos** 35](#_Toc107164472)

[**6.1.** **Descrição do Aplicativo** 38](#_Toc107164473)

[**6.1.1.** **Bibliotecas** 38](#_Toc107164474)

[**6.1.2.** **Declaração de Variáveis** 38](#_Toc107164475)

[**6.1.3.** **Vetores e Matrizes** 39](#_Toc107164476)

[**6.1.4.** **Algoritmo** 40](#_Toc107164477)

[**7.** **Conclusão** 42](#_Toc107164478)

[**8.** **Referências Bibliográficas** 43](#_Toc107164479)

# **ÍNDICE DE FIGURAS**

[Figura 1 - LISTA NÃO NORMALIZADA 7](#_Toc107164480)

[Figura 2 - PRIMEIRA FORMA NORMAL 8](#_Toc107164481)

[Figura 3 - DER: RELAÇÕES FUNCIONÁRIO, CARGO, DEPARTAMENTO, TELEFONE E EMPRÉSTIMO 10](#_Toc107164482)

[Figura 4 - DER: RELAÇÕES SEÇÃO, EXEMPLAR E LIVRO 11](#_Toc107164483)

[Figura 5 - DER: RELAÇÕES LIVRO, EXEMPLAR E AUTOR 11](#_Toc107164484)

[Figura 6 - MER (MODELO ENTIDADE E RELACIONAMENTO) 13](#_Toc107164485)

[Figura 7 - TABELA CADASTRO DE FUNCIONÁRIO 14](#_Toc107164486)

[Figura 8 - TABELA CARGO 14](#_Toc107164487)

[Figura 9 - TABELA DEPARTAMENTO 14](#_Toc107164488)

[Figura 10 - TABELA TELEFONE 15](#_Toc107164489)

[Figura 11 - TABELA EMPRÉSTIMO 15](#_Toc107164490)

[Figura 12 - TABELA LIVRO 15](#_Toc107164491)

[Figura 13 - TABELA AUTOR 15](#_Toc107164492)

[Figura 14 - TABELA SEÇÃO 16](#_Toc107164493)

[Figura 15 - TABELA EXEMPLAR 16](#_Toc107164494)

[Figura 16 - TABELA ESCREVER 16](#_Toc107164495)

[Figura 17 - RELAÇÕES CRIADAS NO MICROSOFT ACCESS A PARTIR DO MER 17](#_Toc107164496)

[Figura 18 - RELATÓRIO CADASTRO DE FUNCIONÁRIO 17](#_Toc107164497)

[Figura 19 - RELATÓRIO CARGO 18](#_Toc107164498)

[Figura 20 - RELATÓRIO DEPARTAMENTO 18](#_Toc107164499)

[Figura 21 - RELATÓRIO TELEFONE 19](#_Toc107164500)

[Figura 22 - RELATÓRIO EMPRÉSTIMO 19](#_Toc107164501)

[Figura 23 - RELATÓRIO LIVRO 20](#_Toc107164502)

[Figura 24 - RELATÓRIO AUTOR 20](#_Toc107164503)

[Figura 25 - RELATÓRIO SEÇÃO 21](#_Toc107164504)

[Figura 26 – RELATÓRIO EXEMPLAR 21](#_Toc107164505)

[Figura 27 - RELATÓRIO ESCREVER 22](#_Toc107164506)

[Figura 28 - CONSULTA CADASTRO DE FUNCIONÁRIO 22](#_Toc107164507)

[Figura 29 - CONSULTA CARGO 23](#_Toc107164508)

[Figura 30 - CONSULTA DEPARTAMENTO 23](#_Toc107164509)

[Figura 31 - CONSULTA TELEFONE 23](#_Toc107164510)

[Figura 32 - CONSULTA EMPRÉSTIMO 24](#_Toc107164511)

[Figura 33 - CONSULTA LIVRO 24](#_Toc107164512)

[Figura 34 - CONSULTA AUTOR 24](#_Toc107164513)

[Figura 35 - CONSULTA SEÇÃO 25](#_Toc107164514)

[Figura 36 - CONSULTA EXEMPLAR 25](#_Toc107164515)

[Figura 37 - CONSULTA ESCREVER 25](#_Toc107164516)

[Figura 38 – MENU PRINCIPAL 26](#_Toc107164517)

[Figura 39 - MENU DOS FORMULÁRIOS 27](#_Toc107164518)

[Figura 40 - FORMULÁRIO CADASTRO DE FUNCIONÁRIO 28](#_Toc107164519)

[Figura 41 - FORMULÁRIO CARGO 29](#_Toc107164520)

[Figura 42 - FORMULÁRIO DEPARTAMENTO 29](#_Toc107164521)

[Figura 43 - FORMULÁRIO TELEFONE 29](#_Toc107164522)

[Figura 44 - FORMULÁRIO EMPRÉSTIMO 30](#_Toc107164523)

[Figura 45 - FORMULÁRIO LIVRO 30](#_Toc107164524)

[Figura 46 - FORMULÁRIO AUTOR 31](#_Toc107164525)

[Figura 47 - FORMULÁRIO SEÇÃO 31](#_Toc107164526)

[Figura 48 - FORMULÁRIO EXEMPLAR 32](#_Toc107164527)

[Figura 49 - FORMULÁRIO ESCREVER 32](#_Toc107164528)

[Figura 50 - MENU DOS RELATÓRIOS 33](#_Toc107164529)

[Figura 51 - MENU DAS CONSULTAS 34](#_Toc107164530)

# **INTRODUÇÃO**

O material apresentado tem como finalidade estudar a aplicação dos conceitos de Banco de Dados e Programação por meio de um estudo de caso realizado em uma Biblioteca de um Escritório de Advocacia.

O trabalho descreve toda a estrutura de um Banco de Dados, desde sua modelagem, normalização, relação, Modelo Entidade e Relacionamento, Diagrama Entidade e Relacionamento, criação do Banco de Dados no Microsoft Access e as descrições de cada etapa no desenvolvimento do trabalho.

Também foi desenvolvido um aplicativo na linguagem C para o controle dos empréstimos realizados durante o dia. Todo o código, suas variáveis, funções e elementos são descritos e explicados no corpo deste artigo.

# **Modelagem de Dados**

## **Lista Não Normalizada**

Para começar o desenvolvimento do trabalho de Banco de Dados, foi feito uma observação a partir de um estudo de caso relacionado a uma biblioteca de um escritório de advocacia. A partir deste estudo, foram obtidas as informações necessárias para a realização da Modelagem de Dados.

O trabalho iniciou-se por observar e criar os atributos e entidades necessários para enfim começar a normalização dos dados, resultando na tabela a seguir:

**Tabela

Descrição gerada automaticamente**

Figura 1 - LISTA NÃO NORMALIZADA

## **Primeira Forma Normal**

Feita então a lista Não Normalizada, o próximo objetivo foi começar a primeira forma normal dos atributos e entidades criadas. Uma tabela está na Primeira Forma Normal quando seus atributos não contêm grupos de Repetição. Os procedimentos para a realização da Primeira Forma Normal são respectivamente:

1. Identificar a chave primária da entidade.
2. Identificar o grupo repetitivo e excluí-lo da entidade;
3. Criar uma entidade com a chave primária da entidade anterior e o grupo repetitivo.

Segue agora a Primeira Forma Normal feita a partir dos atributos e entidades criadas na Lista Não Normalizada.

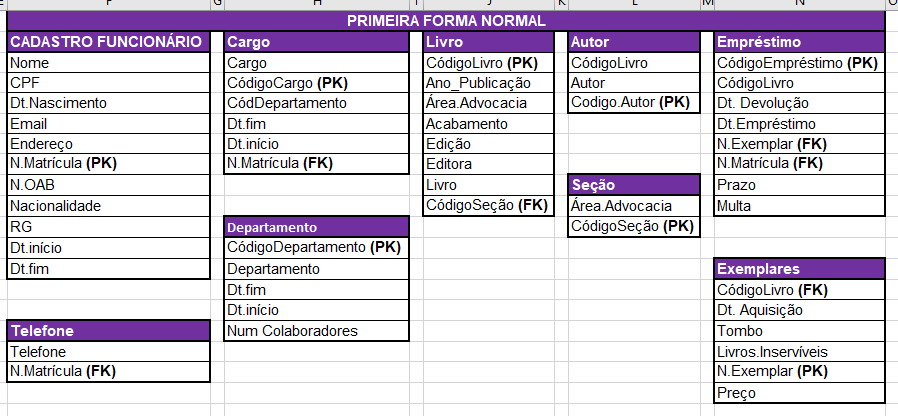
****

Figura 2 - PRIMEIRA FORMA NORMAL

## **Segunda Forma Normal**

Na teoria, o próximo passo a se seguir, é fazer a Segunda Forma Normal a partir das tabelas criadas na realização da Primeira Forma Normal.

A Segunda Forma Normal ocorre quando a chave Primária é composta por mais de um campo. Neste caso, não foi observado a necessidade de aplicação desta forma.

## **Terceira Forma Normal**

Na teoria, o próximo passo a se seguir, é fazer a Terceira Forma Normal a partir das tabelas criadas na realização da Primeira ou Segunda Forma Normal.

A Terceira Forma Normal acontece, pois, na definição dos campos de uma entidade, podem ocorres casos em que um campo não seja dependente diretamente da chave primária ou de parte dela, mas sim dependente de um outro caminho da tabela, campo este que não a chave Primária. Neste caso, não foi observado a necessidade de aplicação desta forma.

A partir então destas regras, chegou-se ao total de nove tabelas, sendo elas:

1. Cadastro de Funcionário;
2. Cargo;
3. Departamento;
4. Telefone;
5. Empréstimo;
6. Livro;
7. Autor;
8. Exemplar;
9. Seção.

# **Diagrama Entidade Relacionamento**

A partir das tabelas criadas pelo processo de normalização, o próximo passo foi definir como estas tabelas se relacionam entre si. Para isto foi criado um Diagrama conhecido como **DER** (Diagrama Entidade Relacionamento).

## **Definição de DER**

Um diagrama entidade relacionamento (DER) é um tipo de fluxograma que ilustra como “entidades”, como por exemplo: pessoas, objetos ou conceitos, se relacionam entre si dentro de um sistema. Diagramas DER são mais utilizados para projetar ou depurar bancos de dados relacionais nas áreas de engenharia de software, sistemas de informações empresariais, educação e pesquisa. Também conhecidos como DERs, ou modelos ER, usam um conjunto definido de símbolos, tais como retângulos, diamantes, ovais e linhas de conexão para representar a interconectividade de entidades, relacionamentos e seus atributos. Eles espelham estruturas gramaticais, onde entidades são substantivos e relacionamentos são verbos.

As figuras a seguir representam o Diagrama de Entidade e Relacionamento criado:

**Gráfico, Gráfico de dispersão

Descrição gerada automaticamente**

Figura 3 - DER: RELAÇÕES FUNCIONÁRIO, CARGO, DEPARTAMENTO, TELEFONE E EMPRÉSTIMO

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Figura 4 - DER: RELAÇÕES SEÇÃO, EXEMPLAR E LIVRO

Interface gráfica do usuário, Tabela, Excel

Descrição gerada automaticamente

Figura 5 - DER: RELAÇÕES LIVRO, EXEMPLAR E AUTOR

## **Relações**

Ao ser criado o DER, foram identificadas seguintes relações:

* Um **funcionário** exerce um **cargo** e um **cargo** pode ser exercido por apenas um **funcionário** (Relação 1 – 1);
* Um **funcionário** possui mais de um **telefone** e um **telefone** é de posse de apenas um **funcionário** (Relação N – 1);
* Um **funcionário** pode realizar vários **empréstimos** e cada **empréstimo** pode ser realizado por apenas um **funcionário** (Relação N – 1);
* Um **cargo** é disponibilizado em um único **departamento** e um **departamento** pode disponibilizar vários **cargos** (Relação 1 – N);
* Um **empréstimo** dispõe de um **exemplar** e um **exemplar** dispõe de apenas um **empréstimo** (Relação 1 – 1);
* Uma **seção** pode conter vários **livros** e um **livro** apenas se contêm em uma só **seção** (Relação N – 1);
* Um **livro** pode ser escrito por vários **autores** e um **autor** pode escrever vários **livros** (Relação N – N);
* Um **livro** pode originar vários **exemplares** e um **exemplar** só pode ser originado por um único **livro** (Relação N – 1).

Como pôde ser observado, a relação criada pelas entidades “autor” e “livro” originou uma relação N – N. Quando uma relação é composta por N – N, cabe ao programador criar uma entidade para o verbo utilizado na relação.

Portanto para este modelo foi criada a entidade “**escrever**” como observado na Figura 5.

# **Modelo Entidade Relacionamento**

A partir das tabelas criadas pelo processo de normalização e das relações desenvolvidas durante o DER, foi realizado o processo de MER (Modelo Entidade Relacionamento).

Neste processo, foram definidas as chaves primárias e as chaves estrangeiras das entidades que compõem o Banco de Dados, realizando-se a relação entre elas.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 6 - MER (MODELO ENTIDADE E RELACIONAMENTO)

## **Definição de MER**

Um **modelo entidade relacionamento** (MER) é um modelo de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um banco de dados, como um banco de dados relacional. Os principais componentes dos Modelos Entidade-Relacionamento (MER) são as entidades (coisas, objetos) suas relações e armazenamento em bancos de dados.

# **Desenvolvimento do Banco de Dados**

Inicia-se então o processo de Desenvolvimento do Banco de Dados.

## **Tabelas**

Para começar, foram criadas no aplicativo **Microsoft Access** as tabelas que compõem ele baseando-se no modelo criado a partir dos processos de modelagem dos Dados. Estas tabelas foram preenchidas com alguns dados para teste.

Uma imagem contendo Texto

Descrição gerada automaticamente

Figura 7 - TABELA CADASTRO DE FUNCIONÁRIO

Tabela

Descrição gerada automaticamente

Figura 8 - TABELA CARGO

Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Figura 9 - TABELA DEPARTAMENTO

Tabela

Descrição gerada automaticamente

Figura 10 - TABELA TELEFONE

Tabela

Descrição gerada automaticamente

Figura 11 - TABELA EMPRÉSTIMO

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 12 - TABELA LIVRO

Tabela

Descrição gerada automaticamente

Figura 13 - TABELA AUTOR

Tabela

Descrição gerada automaticamente

Figura 14 - TABELA SEÇÃO

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 15 - TABELA EXEMPLAR

Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

Figura 16 - TABELA ESCREVER

## **Relações**

Criadas as tabelas, foram definidas as relações entre elas baseando-se no MER (Figura 6).

Diagrama

Descrição gerada automaticamente

Figura 17 - RELAÇÕES CRIADAS NO MICROSOFT ACCESS A PARTIR DO MER

## **Relatórios**

Para o controle dos dados, as secretarias necessitam emitir relatórios das tabelas. Para tanto, foram adicionados na interface formas para a emissão de tais relatórios conforme os exemplos abaixo:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 18 - RELATÓRIO CADASTRO DE FUNCIONÁRIO

Tabela

Descrição gerada automaticamente

Figura 19 - RELATÓRIO CARGO

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente

Figura 20 - RELATÓRIO DEPARTAMENTO

Tabela

Descrição gerada automaticamente

Figura 21 - RELATÓRIO TELEFONE

Tabela

Descrição gerada automaticamente

Figura 22 - RELATÓRIO EMPRÉSTIMO

Tabela

Descrição gerada automaticamente

Figura 23 - RELATÓRIO LIVRO

Tabela

Descrição gerada automaticamente

Figura 24 - RELATÓRIO AUTOR

Tabela

Descrição gerada automaticamente

Figura 25 - RELATÓRIO SEÇÃO

Tabela

Descrição gerada automaticamente

Figura 26 – RELATÓRIO EXEMPLAR

Tabela

Descrição gerada automaticamente

Figura 27 - RELATÓRIO ESCREVER

## **Consultas**

Tanto os funcionários quanto as secretarias precisam de forma para consulta de informações.

As secretarias necessitam controlar os empréstimos e a disponibilidade dos livros. Já os funcionários precisam saber se determinado livro está disponível na biblioteca e sua localização.

Para tanto foram criadas as consultas no Microsoft Access.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 28 - CONSULTA CADASTRO DE FUNCIONÁRIO

Tabela

Descrição gerada automaticamente

Figura 29 - CONSULTA CARGO

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Figura 30 - CONSULTA DEPARTAMENTO

Tabela

Descrição gerada automaticamente

Figura 31 - CONSULTA TELEFONE

Tabela

Descrição gerada automaticamente

Figura 32 - CONSULTA EMPRÉSTIMO

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 33 - CONSULTA LIVRO

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 34 - CONSULTA AUTOR

Interface gráfica do usuário, Texto, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Figura 35 - CONSULTA SEÇÃO

Tabela

Descrição gerada automaticamente

Figura 36 - CONSULTA EXEMPLAR

Tabela

Descrição gerada automaticamente

Figura 37 - CONSULTA ESCREVER

## **Formulários**

Por fim, existe a necessidade de se criar uma interface para os usuários do Banco de Dados. Nesta interface os funcionários terão acesso às consultas para facilitar a localização dos livros desejados. Enquanto as secretarias utilizarão para realizar os cadastros dos livros e funcionários no Banco de Dados da biblioteca.

Esta interface foi desenvolvida a partir da função “formulários” do aplicativo Microsoft Access. A tela inicial deve conter botões para acesso aos relatórios, consultas e formulários.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 38 – MENU PRINCIPAL

O menu “ABRIR FORMULÁRIOS” dá acesso aos formulários de cadastro do Banco de Dados. É por meio deste que a secretaria realizará o cadastro das informações dentro das tabelas do Banco de Dados.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 39 - MENU DOS FORMULÁRIOS

Interface gráfica do usuário, Aplicativo, Email

Descrição gerada automaticamente

Figura 40 - FORMULÁRIO CADASTRO DE FUNCIONÁRIO

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 41 - FORMULÁRIO CARGO

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 42 - FORMULÁRIO DEPARTAMENTO

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 43 - FORMULÁRIO TELEFONE

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 44 - FORMULÁRIO EMPRÉSTIMO

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 45 - FORMULÁRIO LIVRO

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 46 - FORMULÁRIO AUTOR

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 47 - FORMULÁRIO SEÇÃO

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Figura 48 - FORMULÁRIO EXEMPLAR

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Figura 49 - FORMULÁRIO ESCREVER

O menu “ABRIR MENU RELATÓRIOS” permite o acesso aos relatórios do Banco de Dados (Figuras 18 – 27).

Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

Figura 50 - MENU DOS RELATÓRIOS

O menu “ABRIR MENU CONSULTAS” permite o acesso às consultas do Banco de Dados (Figuras 28 – 37).

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Figura 51 - MENU DAS CONSULTAS

# **Criação da Aplicação de Controle dos Empréstimos**

Por fim, foi solicitada a criação de uma aplicação que controlará os empréstimos realizados na Biblioteca. Esta aplicação deve solicitar ao usuário sua matrícula e função (Advogado, estagiário ou Outros) e realizar o controle dos empréstimos, não podendo permitir ao usuário o empréstimo de mais de um exemplar do mesmo livro, bem como não poderá permitir a adição de mais de quatro livros por empréstimo. Ao final do expediente, o programa deve emitir um relatório demonstrando o total de livros emprestado e a função que mais emprestou, mostrando a quantidade total emprestada por ela.

O sistema solicitado foi desenvolvido na linguagem C, resultando no programa abaixo:

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <locale.h>*

*#include <string.h>*

*/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/*

*int main(int argc, char \*argv[]) {*

*setlocale (LC\_ALL, "Portuguese");*

*int i, j, cargo, liv, livros, matric, verif, totLiv;*

*int contLivroFunc[3];*

*int livEsc[4];*

*char ocup [3][20] = {"Advogado", "Estagiário", "Outras Funções"};*

*char confirma, encerrar;*

*totLiv=0;*

*do{*

*livros=0;*

*i=0;*

*for (j=0; j<4; j++){*

*livEsc[j]=0;*

*}*

*printf ("Digite sua matricula\n");*

*scanf ("%d", &matric);*

*do{*

*printf ("Digite sua função (1- Advogado, 2- Estagiário, 3- Outra Função): \n");*

*scanf ("%d", &cargo);*

*if (cargo>3||cargo<1)*

*printf ("Função inválida.\n");*

*} while (cargo<1||cargo>3);*

*do {*

*do{*

*verif=0;*

*liv=0;*

*printf ("Digite o código do livro: (numérico)\n");*

*scanf ("%d", &liv);*

*for (j=0; j<4; j++){*

*if (liv==livEsc[j])*

*verif=1;*

*}*

*if (verif==1)*

*printf ("Não é permitido o empréstimo de mais de um exemplar do mesmo livro, por favor escolha outro.\n");*

*}while (verif==1);*

*livros++;*

*livEsc[i]=liv;*

*i++;*

*printf ("Deseja incluir outro livro no empréstimo? (S/N)\n");*

*scanf ("%s", &confirma);*

*if (livros>=4)*

*printf ("Limite máximo de livros por empréstimo atingido.\n");*

*} while (i<=3&&(confirma=='s'||confirma=='S'));*

*totLiv=totLiv+livros;*

*switch (cargo){*

*case 1: contLivroFunc[0]=contLivroFunc[0]+livros;*

*break;*

*case 2: contLivroFunc[1]=contLivroFunc[1]+livros;*

*break;*

*case 3: contLivroFunc[2]=contLivroFunc[2]+livros;*

*break;*

*default: break;*

*}*

*printf ("Deseja encerrar o programa? (S/N)\n");*

*scanf ("%s", &encerrar);*

*system ("cls");*

*} while (encerrar=='n'||encerrar=='n');*

*printf ("Relatório:\n");*

*printf ("Total de Livros emprestados: %d", livros);*

*if (contLivroFunc[0]>contLivroFunc[1]&&contLivroFunc[0]>contLivroFunc[2])*

*printf ("Maior quantidade de livros emprestada para a função %s. Total: %d livros. \n", ocup[0], contLivroFunc[0]);*

*if (contLivroFunc[1]>contLivroFunc[0]&&contLivroFunc[1]>contLivroFunc[2])*

*printf ("Maior quantidade de livros emprestada para a função %s. Total: %d livros. \n", ocup[1], contLivroFunc[1]);*

*if (contLivroFunc[2]>contLivroFunc[1]&&contLivroFunc[2]>contLivroFunc[0])*

*printf ("Maior quantidade de livros emprestada para a função %s. Total: %d livros. \n", ocup[2], contLivroFunc[2]);*

*system ("PAUSE");*

*return 0;*

*}*

## **Descrição do Aplicativo**

### **Bibliotecas**

A linguagem C trabalha com diversas bibliotecas. Estas podendo ser incluídas no código quando necessário para a execução por meio do comando **#include <biblioteca>**, neste programa utilizamos além das bibliotecas padrão **<stdio.h>** e **<stdlib.h>** as bibliotecas **<locale.h>**, para que o programa exiba e reconheça as palavras na língua portuguesa, e **<string.h>** para trabalhar com palavras ou frases inteiras.

### **Declaração de Variáveis**

Todo e qualquer algoritmo trabalha com variáveis, estas são espaços na memória que o programa reserva para que informações de determinados tipos sejam incluídas. Neste caso, foram incluídas variáveis do tipo número inteiro **"int"** e caractere **"char"**. Estas variáveis são definidas por:

* **int** i e j: Estas variáveis representam índices utilizados principalmente na elaboração de loops dentro do programa;
* **int** cargo, liv e matric: Representam valores de índice para registro das informações, a variável "cargo" por exemplo, serve para que o programa reconheça, por meio de um input numérico, o cargo do usuário;
* **int** livros e totLiv: Contadores que armazenam a quantidade de livros total de um empréstimo e de toda a operação;
* **char** confirma e encerrar: variáveis do tipo caractere que reconhecem o input para que o programa saiba se deve continuar a execução ou repetir o loop atual.

### **Vetores e Matrizes**

Na aplicação também foram utilizados vetores (Arrays), os vetores são conjuntos de variáveis que podem armazenar informações diferentes de um mesmo tipo, estas informações podem ser acessadas pelo programa por meio da utilização de um índice ou apontadas diretamente no código. Todos os vetores são indicados pela utilização de colchetes **[ ]** em sua declaração, dentro destes deve estar declarado a quantidade máxima de variáveis inclusa no vetor.

Além disso também foram utilizadas Matrizes, estas nada mais são do que vetores que armazenam outros vetores as matrizes possuem linhas e colunas, similar a uma tabela e são identificadas por possuir um número entre colchetes que representa a quantidade de linhas e outro que representa a quantidade de colunas. Portanto os vetores e matrizes utilizados se definem por:

* **int** contLivroFunc[3]: Vetor do tipo inteiro utilizado para armazenar a quantidade de livros emprestada para cada classe de funcionários;
* **int** livEsc[4]: Vetor que armazena o código dos livros para que o programa verifique se há repetição do livro escolhido;
* **char** ocup[3][20]: Matriz do tipo caractere que armazena os nomes das funções. Estas funções já foram pré-definidas para os valores definidos entre chaves {"Advogado", "Estagiário", "Outras Funções"}.

### **Algoritmo**

Após a definição das variáveis começa a programação do algoritmo. Primeiro, é importante notar o comando **"setlocale (LC\_ALL, "Portuguese");"** que tem como função definir a língua do programa para "Português", permitindo assim que o programa leia e imprima caracteres acentuados.

No começo do programa foi utilizada a operação **"totLiv=0;"** para que o programa defina o valor da variável **totLiv** como zero, assim, garantindo que não haja nenhuma informação divergente na contagem do total de livros emprestados durante a execução do programa. Logo após, inicia-se um loop utilizando o comando **"do{} while"**, este tipo de loop serve para definir, ao final da operação do empréstimo, se o funcionário deseja encerrar o programa e exibir os relatórios. Para tal, foi utilizada a condição **"(encerrar=='n'||encerrar=='n')"** após o comando **while**, o que fará com que o programa só repita o loop quando a variável encerrar for igual a 'n' ou 'N'.

Depois, foi definido os valores dos índices "i" e "livros" como 0, este passo é necessário para que a cada empréstimos estas variáveis se iniciem com os valores zerados. Além disso o loop **"for (j=0; j<4; j++){livEsc[j]=0;}"** é utilizado para limpar os dados da variável **LivEsc**, isso faz com que, a cada empréstimo, a variável se inicie com os valores zerados . Com isto, o programa apresentará a primeira mensagem ao usuário, definida pelo comando **"printf ("Digite sua matrícula\n")"** e logo em seguida o programa solicitará o input do usuário e salvará na variável **"matric"**, por meio do comando **"scanf ("%d", &matric);".**

O loop **"do{printf ("Digite sua função (1- Advogado, 2- Estagiário, 3- Outra Função): \n");scanf ("%d", &cargo);if (cargo>3||cargo<1)printf ("Função inválida.\n");} while (cargo<1||cargo>3);"** Serve para solicitar ao usuário que informe sua função, e caso o mesmo digite um valor que não seja válido, o programa exibirá a mensagem "Função inválida." e reiniciará o loop. A condição para a exibição da mensagem está definida pelo comando **"if (cargo>3||cargo<1)".**

Após o input da função, o sistema realizará a função de armazenar os códigos dos livros emprestados para isto, foram definidos os valores das variáveis **"verif"** e **"liv"** para zero, depois, utilizando um loop do tipo **"for"**, o programa verificará se o código digitado pelo usuário é igual a algum que já fora incluído anteriormente, caso isso aconteça, o sistema definirá o valor da variável **"verif"** para 1, ativando assim a condição para que a mensagem **"Não é permitido o empréstimo de mais de um exemplar do mesmo livro, por favor escolha outro.\n"** apareça e para que a repetição do loop **"do{} {while(verif==1)"** seja executada. Após isso, a aplicação adicionará 1 aos contadores "livros" e "i" e atualizará o valor da linha "i" do vetor **"livEsc"** com o código do livro emprestado. Após este passo, a aplicação solicitará ao usuário se ele deseja incluir outro livro no empréstimo, porém, caso ele já tenha adicionado quatro livros, o sistema apresentará a mensagem "Limite máximo de livros por empréstimo atingido." e encerrará o loop.

Finalizado o empréstimo, o sistema adicionará a quantidade de livros contida no empréstimo ao total de livros emprestados durante a execução da aplicação, e então adicionará a quantidade de livros à posição do vetor que corresponde à função indicada pelo usuário no começo da aplicação. Para isto foi utilizada uma função de decisão múltipla escolha denominada **"switch"**, esta função verifica o valor da variável "cargo" e realiza a operação de soma na posição indicada dentro do vetor. Após isso, o programa solicitará ao usuário se ele deseja encerrar a operação, repetindo o loop caso o usuário digite 'N' ou 'n'.

Caso a operação seja encerrada, o sistema exibirá o relatório demonstrando a quantidade de livros emprestada no dia e a quantidade emprestada pela função que mais solicitou empréstimos durante o dia. Nesta tela, o sistema só se encerrará quando o usuário pressionar alguma tecla.

# **Conclusão**

O trabalho descreve toda a estrutura de um Banco de Dados, desde sua modelagem, normalização, relação, Modelo Entidade e Relacionamento, Diagrama Entidade e Relacionamento, criação do Banco de Dados no Microsoft Access e as descrições de cada etapa no desenvolvimento do trabalho.

Também foi desenvolvido um aplicativo na linguagem C para o controle dos empréstimos realizados durante o dia. Todo o código, suas variáveis, funções e elementos são descritos e explicados no corpo deste artigo.

O trabalho apresentado pode ser de grande ajuda a todas a pessoas que desejam iniciar o curso de Desenvolvimento de Sistemas, dado o fato que os recursos utilizados no desenvolvimento do trabalho é ensinado no início do curso (1º Módulo), nas disciplinas de Operação de Software e Aplicativo, Programação e Algoritmo e Banco de Dados.

# **Referências Bibliográficas**