

ALUNOS: AFONSO PEREIRA CORREA JUNIOR
KETLYN PRISCILA DE ANDRADE AMORA

Descrição do Sistema

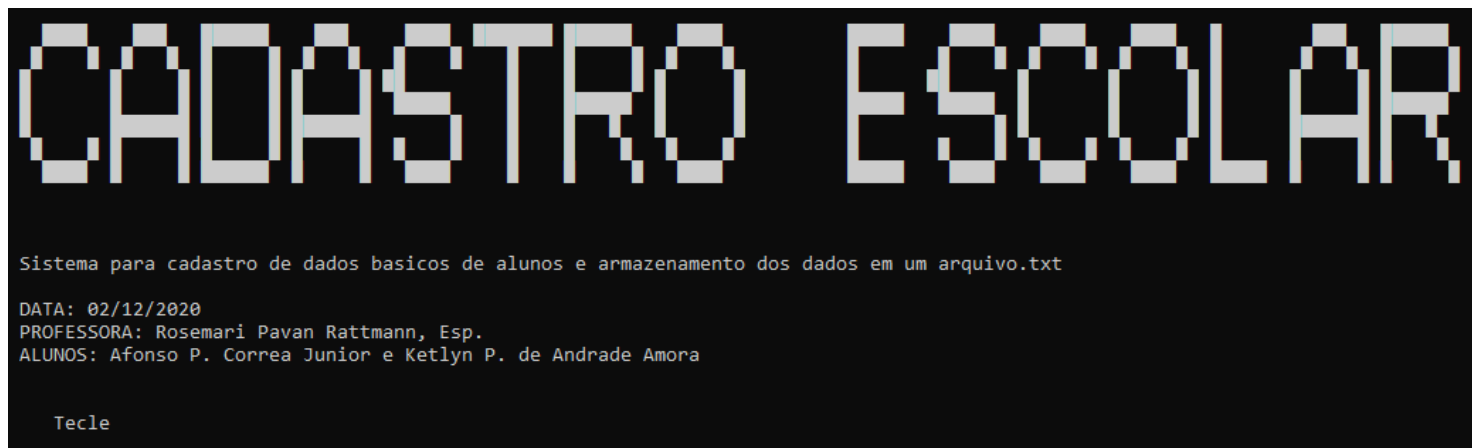
O programa criado pelos alunos: Afonso Pereira Correa Junior e Ketlyn Priscila de Andrade Amora, trata-se de um sistema escolar de controle de notas onde por meio de um menu principal o usuário pode executar o cadastro de um aluno, alterar o cadastro de um aluno, pesquisar os dados de um aluno por nome ou por ID, imprimir o relatório de notas de um aluno pesquisado, imprimir um relatório geral de todos os alunos, gerar um relatório analítico dos dados cadastrados e até mesmo excluir o cadastro de um aluno.

Ademais, no desenvolvimento do código foi-se trabalhado com arquivos, funções parametrizadas, struct(registros), switch case, arrays, ordenação, ponteiros, tela de abertura com apresentação, tela principal onde se é apresentado um menu com as opções que se deseja executar e validações de erros e de controle.

Portanto, apresenta-se um código enxuto, organizado, de fácil entendimento e identificação, totalmente comentado.

TELAS DO PROGRAMA

- ABERTURA



- CRIAÇÃO DO ARQUIVO, CASO NÃO EXISTA NO REPOSITÓRIO DO USUÁRIO

Caso nao tenha um arquivo no seu repositorio, sera criado um com o nome de 'cadastro.txt'
Aperte enter, para prosseguir.

- MENU PRINCIPAL

```
Menu Principal
1 - Cadastrar
2 - Alterar
3 - Pesquisar
4 - Imprimir por nome
5 - Imprimir por ID
6 - Excluir
7 - Sair

Opcao:
```

- CADASTRO caso já exista aluno cadastrado

```
-----
CADASTRO ESCOLAR
-----

CADASTRO DE DADOS

Digite os dados referente ao aluno(a)
NOME: ana julia

Aluno(a) ja cadastrado!
```

- CADASTRO

```
-----
CADASTRO ESCOLAR
-----

Digite os dados referente ao aluno(a)
NOME: guilherme
IDADE: 16
Digite a media referente a materia
Media de Biologia : 8.5
Media de Fisica : 6.5
Media de Geografia : 6.5
Media de Historia : 8.5
Media de Ingles : 9.5
Media de Matematica: 7.5
Media de Portugues : 6.5
Media de Quimica : 6.5

Cadastro salvo com sucesso no arquivo!
```

- INSERIR NOVOS DADOS

```
-----
                        CADASTRO ESCOLAR
-----

Deseja inserir novos dados? [S ou N]
ESCOLHA:
```

- ALTERAR

```
Menu Principal

1 - Cadastrar
2 - Alterar
3 - Pesquisar
4 - Imprimir por nome
5 - Imprimir por ID
6 - Excluir
7 - Sair

Opcao:

Digite o nome do aluno que deseja alterar
NOME: GUILHERME
```

- ALTERAÇÃO DOS DADOS

```
-----
                        CADASTRO ESCOLAR
-----

CADASTRO DE DADOS

Digite os dados referente ao aluno(a)
NOME: GUILHERME SILVA
IDADE: 23
Digite a media referente a materia
Media de Biologia : 10
Media de Fisica : 10
Media de Geografia : 10
Media de Historia : 10
Media de Ingles : 10
Media de Matematica: 10
Media de Portugues : 10
Media de Quimica : 10

Aluno alterado e salvo com sucesso! _
```

- PESQUISAR / MENU PESQUISAR

```
-----
                        CADASTRO ESCOLAR
-----

      DIGITE A OPCAO REFERENTE A PESQUISA DESEJADA

      PESQUISAR POR:
1- NOME
2- ID
3- RELATORIO
4- VOLTAR AO MENU PRINCIPAL
5- SAIR

Opcao:
```

- PESQUISAR POR NOME

```
-----
                        CADASTRO ESCOLAR
-----

      DIGITE A OPCAO REFERENTE A PESQUISA DESEJADA

      PESQUISAR POR:
1- NOME
2- ID
3- RELATORIO
4- VOLTAR AO MENU PRINCIPAL
5- SAIR

Opcao:

Digite o nome do aluno que deseja pesquisar
NOME: ANA JULIA_
```

```
-----
                        CADASTRO ESCOLAR
-----

----- DADOS DO(A) ALUNO(A) -----

ID: 3
Nome: ana julia
Idade: 12
Media das Matérias e Situação Escolar
Biologia   MEDIA: 9.00 - SITUAÇÃO: aprovado
Física     MEDIA: 9.00 - SITUAÇÃO: aprovado
Geografia  MEDIA: 9.00 - SITUAÇÃO: aprovado
História   MEDIA: 9.00 - SITUAÇÃO: aprovado
Inglês     MEDIA: 8.00 - SITUAÇÃO: aprovado
Matemática MEDIA: 8.00 - SITUAÇÃO: aprovado
Português  MEDIA: 8.00 - SITUAÇÃO: aprovado
Química    MEDIA: 5.00 - SITUAÇÃO: reprovado
-----
_
```

- PESQUISAR POR NOME caso o aluno não exista

```
DIGITE A OPCAO REFERENTE A PESQUISA DESEJADA

PESQUISAR POR:
1- NOME
2- ID
3- RELATORIO
4- VOLTAR AO MENU PRINCIPAL
5- SAIR

Opcao:

Digite o nome do aluno que deseja pesquisar
NOME: q

Aluno nao encontrado!
```

- PESQUISAR POR ID

```
DIGITE A OPCAO REFERENTE A PESQUISA DESEJADA

PESQUISAR POR:
1- NOME
2- ID
3- RELATORIO
4- VOLTAR AO MENU PRINCIPAL
5- SAIR

Opcao: 2

Digite o numero do ID do aluno que deseja pesquisar
ID: 1
```

```
----- DADOS DO(A) ALUNO(A) -----

ID: 1
Nome: a
Idade: 1
Media das Matérias e Situação Escolar
Biologia      MEDIA: 1.00 - SITUAÇÃO: reprovado
Física        MEDIA: 1.00 - SITUAÇÃO: reprovado
Geografia     MEDIA: 1.00 - SITUAÇÃO: reprovado
História      MEDIA: 1.00 - SITUAÇÃO: reprovado
Inglês        MEDIA: 1.00 - SITUAÇÃO: reprovado
Matemática    MEDIA: 1.00 - SITUAÇÃO: reprovado
Português     MEDIA: 1.00 - SITUAÇÃO: reprovado
Química       MEDIA: 1.00 - SITUAÇÃO: reprovado
-----
```

- PESQUISAR POR ID caso id digitado não exista

```
DIGITE A OPCAO REFERENTE A PESQUISA DESEJADA

PESQUISAR POR:
1- NOME
2- ID
3- RELATORIO
4- VOLTAR AO MENU PRINCIPAL
5- SAIR

Opcao:

Digite o numero do ID do aluno que deseja pesquisar
ID: 0

Aluno nao encontrado!
```

- RELATÓRIO

```
-----
CADASTRO ESCOLAR
-----

RELATORIO DE CADASTROS

Quantidade de alunos cadastrados: 12
Quantidade de alunos aprovados: 83
Quantidade de alunos reprovados: 13
Faixa etaria dos alunos cadastrados de 1 a 25 anos
Media das idades cadastradas 16.67

Obs: lembrando que o aluno possui 8 disciplinas
Ou seja, ele pode ter sido aprovado em tres e reprovado nas demais
```

- VOLTAR AO MENU PRINCIPAL

```
Menu Principal

1 - Cadastrar
2 - Alterar
3 - Pesquisar
4 - Imprimir por nome
5 - Imprimir por ID
6 - Excluir
7 - Sair

Opcao: _
```

- IMPRIMIR POR NOME

CADASTRO ESCOLAR

DADOS DOS ALUNOS CADASTRADOS

ID: 1

Nome: a

Idade: 1

Media das Matérias e Situação Escolar

Biologia MEDIA: 1.00 - SITUAÇÃO: reprovado

Física MEDIA: 1.00 - SITUAÇÃO: reprovado

Geografia MEDIA: 1.00 - SITUAÇÃO: reprovado

História MEDIA: 1.00 - SITUAÇÃO: reprovado

Inglês MEDIA: 1.00 - SITUAÇÃO: reprovado

Matemática MEDIA: 1.00 - SITUAÇÃO: reprovado

Português MEDIA: 1.00 - SITUAÇÃO: reprovado

Química MEDIA: 1.00 - SITUAÇÃO: reprovado

ID: 2

Nome: ana

Idade: 15

Media das Matérias e Situação Escolar

Biologia MEDIA: 15.00 - SITUAÇÃO: aprovado

Física MEDIA: 15.00 - SITUAÇÃO: aprovado

Geografia MEDIA: 15.00 - SITUAÇÃO: aprovado

História MEDIA: 15.00 - SITUAÇÃO: aprovado

Inglês MEDIA: 15.00 - SITUAÇÃO: aprovado

Matemática MEDIA: 15.00 - SITUAÇÃO: aprovado

Português MEDIA: 15.00 - SITUAÇÃO: aprovado

Química MEDIA: 15.00 - SITUAÇÃO: aprovado

ID: 4

Nome: ana bialli

Idade: 9

Media das Matérias e Situação Escolar

Biologia MEDIA: 7.00 - SITUAÇÃO: aprovado

Física MEDIA: 7.50 - SITUAÇÃO: aprovado

Geografia MEDIA: 6.50 - SITUAÇÃO: aprovado

História MEDIA: 4.50 - SITUAÇÃO: reprovado

Inglês MEDIA: 8.70 - SITUAÇÃO: aprovado

Matemática MEDIA: 9.70 - SITUAÇÃO: aprovado

Português MEDIA: 8.40 - SITUAÇÃO: aprovado

Química MEDIA: 4.50 - SITUAÇÃO: reprovado

ID: 3

Nome: ana julia

Idade: 12

- IMPRIMIR POR ID

CADASTRO ESCOLAR

DADOS DOS ALUNOS CADASTRADOS

ID: 1

Nome: a

Idade: 1

Media das Matérias e Situação Escolar

Biologia	MEDIA: 1.00	- SITUACAO: reprovado
Fisica	MEDIA: 1.00	- SITUACAO: reprovado
Geografia	MEDIA: 1.00	- SITUACAO: reprovado
Historia	MEDIA: 1.00	- SITUACAO: reprovado
Ingles	MEDIA: 1.00	- SITUACAO: reprovado
Matematica	MEDIA: 1.00	- SITUACAO: reprovado
Portugues	MEDIA: 1.00	- SITUACAO: reprovado
Quimica	MEDIA: 1.00	- SITUACAO: reprovado

ID: 2

Nome: ana

Idade: 15

Media das Matérias e Situação Escolar

Biologia	MEDIA: 15.00	- SITUACAO: aprovado
Fisica	MEDIA: 15.00	- SITUACAO: aprovado
Geografia	MEDIA: 15.00	- SITUACAO: aprovado
Historia	MEDIA: 15.00	- SITUACAO: aprovado
Ingles	MEDIA: 15.00	- SITUACAO: aprovado
Matematica	MEDIA: 15.00	- SITUACAO: aprovado
Portugues	MEDIA: 15.00	- SITUACAO: aprovado
Quimica	MEDIA: 15.00	- SITUACAO: aprovado

ID: 3

Nome: ana julia

Idade: 12

Media das Matérias e Situação Escolar

Biologia	MEDIA: 9.00	- SITUACAO: aprovado
Fisica	MEDIA: 9.00	- SITUACAO: aprovado
Geografia	MEDIA: 9.00	- SITUACAO: aprovado
Historia	MEDIA: 9.00	- SITUACAO: aprovado
Ingles	MEDIA: 8.00	- SITUACAO: aprovado
Matematica	MEDIA: 8.00	- SITUACAO: aprovado
Portugues	MEDIA: 8.00	- SITUACAO: aprovado
Quimica	MEDIA: 5.00	- SITUACAO: reprovado

ID: 4

Nome: ana bialli

Idade: 9

- EXCLUIR

```
Menu Principal
1 - Cadastrar
2 - Alterar
3 - Pesquisar
4 - Imprimir por nome
5 - Imprimir por ID
6 - Excluir
7 - Sair

Opcao: 6

Digite o numero do ID do aluno que deseja inativar
ID: 1

Cadastro do aluno(a): a inativado com sucesso!
```

- IMPRIMINDO SEM O ID 1

```
-----
CADASTRO ESCOLAR
-----

DADOS DOS ALUNOS CADASTRADOS

ID: 2

Nome: ana
Idade: 15

Media das Matérias e Situação Escolar
Biologia MEDIA: 15.00 - SITUACAO: aprovado
Fisica MEDIA: 15.00 - SITUACAO: aprovado
Geografia MEDIA: 15.00 - SITUACAO: aprovado
Historia MEDIA: 15.00 - SITUACAO: aprovado
Ingles MEDIA: 15.00 - SITUACAO: aprovado
Matematica MEDIA: 15.00 - SITUACAO: aprovado
Portugues MEDIA: 15.00 - SITUACAO: aprovado
Quimica MEDIA: 15.00 - SITUACAO: aprovado
-----

ID: 3

Nome: ana julia
Idade: 12

Media das Matérias e Situação Escolar
Biologia MEDIA: 9.00 - SITUACAO: aprovado
Fisica MEDIA: 9.00 - SITUACAO: aprovado
Geografia MEDIA: 9.00 - SITUACAO: aprovado
Historia MEDIA: 9.00 - SITUACAO: aprovado
Ingles MEDIA: 8.00 - SITUACAO: aprovado
Matematica MEDIA: 8.00 - SITUACAO: aprovado
Portugues MEDIA: 8.00 - SITUACAO: aprovado
Quimica MEDIA: 5.00 - SITUACAO: reprovado
-----
```

- CASO NÃO HAJA NENHUM CADASTRO é impresso essa mensagem em todas funções

```
Menu Principal

1 - Cadastrar
2 - Alterar
3 - Pesquisar
4 - Imprimir por nome
5 - Imprimir por ID
6 - Excluir
7 - Sair

Opcao: Nenhum aluno cadastrado! █
```

- VALIDAÇÃO DE ENTRADA toda vez que pede um nome

```
-----
CADASTRO ESCOLAR
-----

CADASTRO DE DADOS

Digite os dados referente ao aluno(a)
NOME: askjdbasdhbaksdfb askffdbadjhbfjshfbjhbbsdfkjbasdjfhbsdkhfbjsdhbfjhsbdfjhsdfjhsdhfb

Nome digitado invalido!

Utilize apenas letras e o nome deve conter menos de 49 caracteres.
```

ALGORITMO

SISTEMA ESCOLAR CRUD

```
STRUCT ALUNO {
    - NOME
    - IDADE
    - INATIVO
    - ID
}
STRUCT DISCIPLINA {
```

```
- MATERIA
- MEDIA
- SITUACAO - aprovado/reprovado
}
CADASTRAR {
    - NAO DUPLICAR DADOS (PESQUISAR SE NOME JA EXISTE NA
MEMORIA)
    - ID UNICO PARA CADA ALUNO
    - SOLICITAR DADOS
    - INICIALIZAR INATIVO COM 1
    - VALIDAR NOME
    - CADA MATERIA TERA UMA MEDIA
    - CADA MATERIA TERA UMA SITUACAO
    - GRAVAR NO ARQ
}
ALTERAR {
    - SOLICITAR ID OU NOME
    - PESQUISAR
    - CHAMAR FUNCAO PARA CADASTRAR NO MESMO INDICE ONDE A
PESQUISA RETORNOU
    - SALVAR CADASTRO
    - GRAVAR NO ARQ
    - (NAO DUPLICAR DADOS, EXLCUIR E CRIAR UM NOVO ARQ GRAVANDO
TUDO DE NOVO?)
}
EXCLUIR {
    - SOLICITAR ID OU NOME
    - PESQUISAR
    - INATIVO DO ALUNO CORRESPONDENTE A PESQUISA RECEBE 1
    - DECREMENTA A VARIABEL QUE CONTROLA A QTDE DE ALUNOS
}
PESQUISAR {
    - BUSCAR POR NOME OU ID
    - VALIDAR ENTRADA DO NOME RECEBIDO
    - RETORNAR O INDICE DE ONDE FOI ENCONTRADO
}
IMPRIMIR {
    - VALIDAR SE O ALUNO ESTA INATIVO
    - UMA FUNCAO DE IMPRIMIR APENAS PARA O RETORNO DA FUNCAO
PESQUISAR
    - UMA FUNCAO PARA IMPRIMIR TODOS OS DADOS
    - IMPRIMIR ORDENADO POR NOME
    - IMPRIMIR ORDENADO POR ID
}
```

```
ARQUIVO {
    - LER (FREAD OU FSCAN?)
    - GRAVAR (FWRITE OU FPRINTF?)
    - CONTROLAR QUANTOS CADASTROS HÁ NO ARQ LOGO NO INICIO
    - NAO PERMITIR DUPLICAR DADOS
    - EXCLUIR ARQ E ABRI-LO NOVAMENTE PARA FUNCAO ALTERAR E
EXCLUIR
    - LER OS DADOS E SALVAR NA STRUCT
}
MAIN {
    - ABERTURA
    - MENU
    - SWITCH CASE
        - CADASTRAR
        - EXCLUIR
        - ALTERAR
        - IMPRIMIR
        - MENU PESQUISAR
            - SWTIH CASE
                - PESQUISAR
                - RELATORIO
                - VOLTAR AO MENU
                - SAIR
    - SAIR
}
```

CÓDIGO FONTE

```
/* ===== BIBLIOTECAS ===== */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <locale.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

#define MAT 8 //quantidade maxima de materias cadastradas
#define MAX 100 //quantidade maxima de alunos
```

#define TAM 40

```
typedef struct {  
    float media;  
    char materia[15];  
    char situacao[10];  
} Disciplina;
```

```
typedef struct {  
    int id;  
    int idade;  
    int inativo;  
    char nomeAluno[50];  
    Disciplina disciplinas[MAT];  
} Aluno;
```

```
Aluno alunos[MAX];
```

```
void flush();  
char addMais();  
void montaMenu();  
void abertura();  
void cabecalho();  
int carregarDados();  
void gotoxy(int x,int y);  
void lowerString(char str[]);  
void menuPesquisar(int qtde);  
void criarMenuLinhaSuperior();  
Aluno cadastrarAluno(int cont);  
int pesquisarNome(char nome[]);  
void imprimirCadastro(int qtde);  
int pesquisarId(int id, int qtde);  
void criarMenuLinhaRodape(int tamX);  
void preencherDisciplina(Aluno *aluno);  
void criarMenuLinhaHorizontal(int tamX);  
void criarMenuItem(int tamX, char str[]);  
int validaNome(char nome[], int tamanho);
```

```
void insertionSort(int qtde, int escolha);
void gravarDados(int cont, int mostrarMsg);
void insertionSort(int qtde, int escolha);
void exibirAlunoPesquisado(int indice, int qtde);
void pesquisarDisciplina(char situacao[], int qtde, int *indices);

int main () {

    FILE *arq;

    //variavel para o segundo loop, enquanto o usuario escolher S para
realizar novos cadastros
    char op = 'S', buscarNome[50];
    //cont e qtde irão armazenar o numeros de alunos cadastrados
    int opcao = 0, qtde = 0, numId = 0, buscald = 0;
    //Escolha para armazenar a escolha de como imprimir os cadatros
    //Posicao recebe o retorno do pesquisa nome, onde foi encontrado ou
nao o aluno buscaso
    //tamanho da para o tamanho da string, validar e revalidar para verificar
dados do usuario
    int escolha = 0, posicao = 0, validar, revalidar, tamanho, mostrarMsg;

    system("title CADASTRO ESCOLAR");

    abertura();
    gotoxy(10, 10);
    //Criando um arquivo, caso não exista no repositório e avisando ao
usuario sobre a criação
    printf("\n\n\t Caso nao tenha um arquivo no seu repositório, sera criado
um com o nome de 'cadastro.txt' \n");
    printf("\t Aperte enter, para prosseguir. ");
    getch();
    if (!(arq = fopen("cadastro.txt", "a+"))) {
        printf("Erro! Impossivel abrir o arquivo!\n");
        exit(1);
    }
    fclose(arq);

    //qtde recebe o retorno da função, caso já exista alunos cadastrados,
```

ele é incrementado

```
qtde = carregarDados();
```

```
do{
```

```
    op = 'S';
```

```
    system("cls");
```

```
    gotoxy(2, 5);
```

```
    montaMenu();
```

```
    printf(" Opcao: ");
```

```
    opcao = getch();
```

```
    switch(opcao) {
```

```
        case '1': //funcoes para realizar o cadastro e gravar os
```

dados no arquivo

```
            if(qtde == MAX) {
```

```
                printf("\n Numero de alunos cadastrados
```

excedidos! \n");

```
                getch();
```

```
            }
```

```
            while ((qtde < MAX) && (op == 'S')) {
```

```
                flush();
```

```
                //Chamando funcao para realizar um
```

cadastro e alunos recebe o retorno dela na posicao cont

```
                alunos[qtde] = cadastrarAluno(qtde);
```

```
                mostrarMsg = 1;
```

```
                gravarDados(qtde, mostrarMsg);
```

```
                op = addMais();
```

```
                qtde++;
```

```
            }
```

```
        break;
```

```
        case '2': //funcoes para alterar, cadastrar e gravar na
```

mesma posicao onde aluno x esta

```
            if(qtde == 0) {
```

```
                printf("\n\t Nenhum aluno cadastrado! ");
```

```
                getch();
```

```
                break;
```

```
            }
```

```
        flush();
```

```
        printf("\n\n Digite o nome do aluno que deseja
```

```
alterar\n");

printf(" NOME: ");
gets(buscarNome);
//Verificando se o nome do aluno que será
alterado é um nome válido, se não for, entra no loop
lowerString(buscarNome);
tamanho = strlen (buscarNome);
revalidar = validaNome(buscarNome, tamanho);
while(revalidar != 1){
    printf("\n\t Nome digitado invalido!\n");
    printf("\t NOME: ");
    gets(buscarNome);
    tamanho = strlen (buscarNome);
    flush();
    lowerString(buscarNome);
    revalidar = validaNome(buscarNome,
tamanho);
}
// Posicao recebe o retorno da funcao, que será
o indice do array onde o aluno x foi encontrado
posicao = pesquisarNome(buscarNome);
if(posicao == -1) {
    printf("\n Aluno nao encontrado!\n");
    getch();
    break;
} else {
    //Abrindo arquivo como w+ para ser
criado um novo e os dados alterados serem salvos,
//sem serem duplicados
if (!(arq = fopen("cadastro.txt", "w+"))) {
    printf("Erro! Impossivel abrir o
arquivo!\n");

    exit(1);
}
fclose(arq);

//Chamando funcao para cadastrar
novamente o aluno

numId = alunos[posicao].id;
//variavel é decrementada pois ela possui
```

```
o numero exato do id do aluno, e na funcao para cadastrar
                                //a variavel que controla o id e
incrementada em cada interacao, para que cada aluno tenha um id unico
                                numId -= 1;
                                //Atribuindo um espaço ao nome, para
que o usuario possa utilizar o mesmo nome cadastrado anteriormente
                                strcpy(alunos[posicao].nomeAluno, " ");
                                //E o vetor alunos recebe na mesma
posicao onde o aluno que sera alterado, o novo cadastro
                                alunos[posicao] = cadastrarAluno(numId);
                                //Gravando os novos dados de todos os
alunos novamente no novo arquivo
                                for(int i=0; i<qtde; i++) {
                                    mostrarMsg = 0;
                                    gravarDados(i, mostrarMsg);
                                }
                                printf("\n\n\n\t Aluno alterado e salvo com
sucesso! ");

                                getch();
                            }
                            //funcao alterar
                            break;
                        case '3': //funcao para chamar o menu pesquisar e suas
opcoes de pesquisa
                            //Validando se existe algum cadastro para entao
realizar a pesquisa
                            if(qtde == 0) {
                                printf("\n\t Nenhum aluno cadastrado! ");
                                getch();
                                break;
                            }
                            menuPesquisar(qtde);
                            break;
                        case '4': //funcao para imprimir ordenado por nome
                            //Validando se existe algum cadastro para entao
imprimir os cadastros ordenados por nome
                            if(qtde == 0) {
                                printf("\n\t Nenhum aluno cadastrado! ");
                                getch();
                                break;
```

```
    }
    escolha = 1;
    insertionSort(qtde, escolha);
    imprimirCadastro(qtde);
    break;
case '5': //funcao para imprimir ordenado por ID
    //Validando se existe algum cadastro para entao
imprimir os cadastros ordenados por ID
    if(qtde == 0) {
        printf("\n\t Nenhum aluno cadastrado! ");
        getch();
        break;
    }
    escolha = 2;
    insertionSort(qtde, escolha);
    imprimirCadastro(qtde);
    break;
case '6': //funcao para inativar o cadastro
    //Validando se existe algum cadastro para então
excluir
    if(qtde == 0) {
        printf("\n\t Nenhum aluno cadastrado! ");
        getch();
        break;
    }
    flush();
    printf("\n\n Digite o numero do ID do aluno que
deseja inativar\n");

    printf(" ID: ");
    scanf("%d", &buscald);
    // Posicao recebe o retorno da funcao, que será
o indice do array onde o aluno x foi encontrado
    posicao = pesquisarId(buscald, qtde);
    if(posicao == -1) {
        printf("\n Aluno nao encontrado!\n");
        getch();
        break;
    } else {
        //Abrindo arquivo como w+ para ser
criado um novo e os dados alterados serem salvos,
```

```
//sem serem duplicados
if (!(arq = fopen("cadastro.txt","w+"))) {
    printf("Erro! Impossivel abrir o
arquivo!\n");
    exit(1);
}
fclose(arq);

alunos[posicao].inativo = 1;
qtde -= 1;
printf("\n\t Cadastro do aluno(a): %s
inativado com sucesso!", alunos[posicao].nomeAluno);
//Gravando os novos dados de todos os
alunos novamente no novo arquivo
for(int i=0; i<qtde; i++) {
    mostrarMsg = 0;
    gravarDados(i, mostrarMsg);
}
getch();
}
break;
case '7':
    //SAIR
    exit(1);
    break;
default:
    printf("\n Opcao digitada invalida!\n");
    getch();
}

} while(opcao != 7);
return (0);
}

//Funcao para realizar a abertura
void abertura() {
    //y1=linha inicial y2 para recuperar a linha inicial
    int y1=15, x1=8, y2=y1, x2=x1, a;
    int letras[16][7]={ { 60, 66, 128, 128, 128, 66, 60 }, /*C*/
```

```

{ 56, 68, 130, 130, 254, 130, 130 }, /*A*/
{ 252, 130, 130, 130, 130, 130, 252 }, /*D*/
{ 56, 68, 130, 130, 254, 130, 130 }, /*A*/
{ 60, 66, 192, 124, 2, 66, 60 }, /*S*/
{ 254, 16, 16, 16, 16, 16, 16 }, /*T*/
{ 252, 130, 130, 252, 136, 132, 130 }, /*R*/
{ 60, 66, 129, 129, 129, 66, 60 }, /*O*/
{ 0, 0, 0, 0, 0, 0, 0 }, /*Espaco*/
{ 124, 64, 64, 124, 64, 64, 124 }, /*E*/
{ 60, 66, 192, 124, 2, 66, 60 }, /*S*/
{ 60, 66, 128, 128, 128, 66, 60 }, /*C*/
{ 60, 66, 129, 129, 129, 66, 60 }, /*O*/
{ 64, 64, 64, 64, 64, 64, 124 }, /*L*/
{ 56, 68, 130, 130, 254, 130, 130 }, /*A*/
{ 252, 130, 130, 252, 136, 132, 130 }; /*R*/

```

```

for(int i=0;i<16;i++){          /* sao 16 letras */
    for(int j=0;j<7;j++){      /*o caractere é de 7 linhas */
        a=letras[i][j];       /*pega o numero da 1 linha do caractere
*/
        for(int k=0;k<8;k++){  /*aqui eh um numero de 8 bits*/
            gotoxy(x1, y1);    /*posiciona o cursor no local */
            if (a % 2 != 0)     /*se bit=1 */
                printf("%c",219); /*coloca espaço invertido */
            else
                printf(" ");    /*senão coloca espaço normal*/
            a=a/2;              /*dividindo numero por 2 desloca os bits para
direita*/
            x1--;               /* move para coluna de tras */
            Sleep(10);
        }
        y1++;                  /*muda para linha de baixo */
        x1=x2;                 /*volta para a coluna inicial*/
    }                          /*proxima linha*/
    x2=x2+8;                   /*avança 8 colunas posicionando para o prox
caract*/
    x1=x2;
    y1=y2;                     /*volta para linha inicial*/
}                              /*imprimiu o primeiro caractere*/
gotoxy(0,23);

```

```
printf("\n\n Sistema para cadastro de dados basicos de alunos e
armazenamento dos dados em um arquivo.txt ");
printf("\n\n DATA: 02/12/2020\n");
printf(" PROFESSORA: Rosemari Pavan Rattmann, Esp.\n");
printf(" ALUNOS: Afonso P. Correa Junior e Ketlyn P. de Andrade Amora\n");
printf("\n\n Tecele" ) ;
getch();
system("cls");

}
```

```
//Funcao para ler os dados armazenados no arquivo
int carregarDados() {
    FILE *arq;
    Aluno aux;
    int retorno = 1;
    int indice = 0;
    // abre o arquivo para leitura
    if (!(arq = fopen("cadastro.txt","r"))) {
        printf("Erro! Impossivel abrir o arquivo!\n");
        exit(1);
    }
    while(1)
    {
        // fread ler os dados
        // retorna a quantidade de elementos lidos com sucesso
        retorno = fread(&aux, sizeof(Aluno), 1, arq);

        // se retorno for menor que o count, então sai do loop
        if(retorno < 1)
            break;
        else
            alunos[indice++] = aux;
    }

    fclose(arq); // fecha o arquivo
    return (indice);
}
```

```
//Funcoes graficas para exibir o menu
void criarMenuLinhaSuperior() {
    int i;
    printf("\t\t\t\t\t");
    printf("%c", 201);
    for(i=0; i<TAM; i++) {
        printf("%c", 205);
    }
    printf("%c\n", 187);
}

void criarMenuLinhaRodape(int tamX) {
    int i;
    printf("\t\t\t\t\t");
    printf("%c", 200);
    for(i=0; i<TAM; i++) {
        printf("%c", 205);
    }
    printf("%c\n", 188);
}

void criarMenuItem(int tamX, char str[]) {
    printf("\t\t\t\t\t");
    printf("%c", 186);
    printf("%-*s", tamX, str);
    printf("%c\n", 186);
}

void criarMenuLinhaHorizontal(int tamX) {
    int i;
    printf("\t\t\t\t\t");
    printf("%c", 204);
    for(i=0; i<tamX; i++) {
        printf("%c", 205);
    }
    printf("%c\n", 188);
}

void montaMenu() {
```

```
    criarMenuLinhaSuperior();
    criarMenuLinha(27, "\t\tMenu Principal");
    criarMenuLinhaHorizontal(TAM);
    criarMenuLinha(TAM, " 1 - Cadastrar");
    criarMenuLinha(TAM, " 2 - Alterar");
    criarMenuLinha(TAM, " 3 - Pesquisar");
    criarMenuLinha(TAM, " 4 - Imprimir por nome");
    criarMenuLinha(TAM, " 5 - Imprimir por ID");
    criarMenuLinha(TAM, " 6 - Excluir");
    criarMenuLinha(TAM, " 7 - Sair");
    criarMenuLinhaRodape(TAM);
}

//Funcao para realizar UM cadastro e fazendo validacoes e controle das
informacoes digitadas pelo usuario
Aluno cadastrarAluno(int cont){
    //Variavel temporaria que armazena um unico aluno
    Aluno aluno;

    int validar = 0, revalidar = 0, tamanho = 0;
    cabecalho();
    preencherDisciplina(&aluno);

    printf("\n\t CADASTRO DE DADOS \n");
    printf("\n\t Digite os dados referente ao aluno(a)");
    printf("\n\t NOME: ");
    gets(aluno.nomeAluno);
    lowerString(aluno.nomeAluno);
    tamanho = strlen(aluno.nomeAluno);

    //Validando nome digitado, se contem apenas letras
    validar = validaNome(aluno.nomeAluno, tamanho);

    //Validando se o nome digitado já consta armazenado no sistema
    revalidar = pesquisarNome(aluno.nomeAluno);

    //Loop para enquanto o nome nao ser valido ou constar no sistema,
    pede um novo nome ao usuario
    while((validar != 1) || (revalidar > -1)){
        if (revalidar > -1) {
```

```
printf("\n\n\t\t Aluno(a) ja cadastrado!\n");
}
if (validar != 1) {
    printf("\n\n\t Nome digitado invalido!\n");
    printf("\n\t Utilize apenas letras e o nome deve conter
menos de 49 caracteres.\n");
}
getch();
cabecalho();
printf("\n\t Digite os dados referente ao aluno(a)");
printf("\n\t NOME: ");
gets(aluno.nomeAluno);
lowerString(aluno.nomeAluno);
tamanho = strlen(aluno.nomeAluno);
validar = validaNome(aluno.nomeAluno, tamanho);
revalidar = pesquisarNome(aluno.nomeAluno);
}
aluno.inativo = 0;
//Acrescentando o ID ao aluno e incrementando toda vez que chamar a
funcao
aluno.id = cont+1;
//Controlando a entrada do usuároo se é um numero, se não for entra
no loop
printf("\t IDADE: ");
scanf("%d", &aluno.idade);

printf("\t Digite a media referente a materia\n");
//Loop para armazenar as notas de cada materia
for(int i=0; i<MAT; i++){
    flush();
    printf("\t Media de %s: ", aluno.disciplinas[i].materia);
    scanf("%f", &aluno.disciplinas[i].media);

    if(aluno.disciplinas[i].media >= 6.0) {
        strcpy((aluno.disciplinas[i].situacao), "aprovado");
    } else {
        strcpy((aluno.disciplinas[i].situacao), "reprovado");
    }
}
```



```
        return(aluno);
    }

//Funcao para gravar os dados armazenados no arquivo
void gravarDados(int cont, int mostrarMsg)
{
    int gravado = 0;
    FILE *arq;

    if (!(arq = fopen("cadastro.txt","a"))) {
        printf("Erro! Impossivel abrir o arquivo!\n");
        exit(1);
    }

    //Escreve cada elemento do vetor no arquivo
    gravado = fwrite((float*)&alunos[cont], sizeof(Aluno), 1, arq);
    if((gravado != 0) && (mostrarMsg == 1)) {
        printf("\n\tCadastro salvo com sucesso no arquivo!\n");
        Sleep(1500);
        system("cls");
    }
    fclose(arq);
}

//Funcao para validar se o usuario quer continuar cadastrando
char addMais() {
    char opcao;

    do {
        cabecalho();
        printf("\n\t Deseja inserir novos dados? [S ou N]");
        printf("\n\t ESCOLHA: ");
        opcao = getch();
        opcao = toupper(opcao);
        printf("\n");
    } while (opcao != 'S' && opcao != 'N');
    return opcao;
}
```

```
//Funcao que exhibe o menu com as opcoes de pesquisa
void menuPesquisar(int qtde){

    int opcao, revalidar = 0, indice = 0;
    //Variaveis para armazenar tamanho da string, qtde de aluno
    aprovados e reprovados
    int tamanho = 0, qtdeAp, qtdeRe;
    //Variavel para armazenar o ID que usuario quer pesquisar
    int buscald, maiorIdade, menorIdade;
    float medialdade;
    //Variavel para armazenar o nome que usuario quer pesquisar
    char nomeBuscado[50];

    do {

        cabecalho();
        gotoxy(10,5);
        printf("\n\t DIGITE A OPCAO REFERENTE A PESQUISA
DESEJADA \n\n\t PESQUISAR POR:");
        printf("\n 1- NOME\n 2- ID\n 3- RELATORIO\n 4- VOLTAR AO
MENU PRINCIPAL\n 5- SAIR\n\n Opcao: ");
        opcao = getch();

        switch(opcao) {
            case '1':
                flush();
                printf("\n\n Digite o nome do aluno que deseja
pesquisar\n");

                printf(" NOME: ");
                gets(nomeBuscado);

                //Verificando se o nome do aluno que será
                pesquisado é um nome válido, se não for, entra no loop
                lowerString(nomeBuscado);
                tamanho = strlen (nomeBuscado);
                revalidar = validaNome(nomeBuscado,
tamanho);

                while(revalidar != 1){
```

```
printf("\n\t Nome digitado invalido!\n");
printf("\t NOME: ");
gets(nomeBuscado);
tamanho = strlen (nomeBuscado);
flush();
lowerString(nomeBuscado);
revalidar = validaNome(nomeBuscado,
tamanho);
}
// Indice recebe o retorno da funcao, que será o
indice do array onde o aluno x foi encontrado
indice = pesquisarNome(nomeBuscado);
if(indice == -1) {
    printf("\n Aluno nao encontrado!\n");
    getch();
} else {
    exibirAlunoPesquisado(indice, qtde);
}

break;
case '2':
    flush();
    printf("\n\n Digite o numero do ID do aluno que
deseja pesquisar\n");

    printf(" ID: ");
    scanf("%d", &buscald);
    // Indice recebe o retorno da funcao, que será o
indice do array onde o ID x foi encontrado
    indice = pesquisarId(buscald, qtde);
    if(indice == -1) {
        printf("\n Aluno nao encontrado!\n");
        getch();
    } else {
        exibirAlunoPesquisado(indice, qtde);
    }

    break;
case '3':
    //Armazenando a maior idade, a menor, a media
das idades
```

```
//A qtde de aprovados e repovados nas
disciplinas

maiorIdade = alunos[0].idade;
menorIdade = alunos[0].idade;
qtdeAp = 0;
qtdeRe = 0;
for(int i=0; i<qtde; i++) {
    medialdade += alunos[i].idade;
    if(alunos[i].idade > maiorIdade){
        maiorIdade = alunos[i].idade;
    }
    if(alunos[i].idade < menorIdade) {
        menorIdade = alunos[i].idade;
    }
    for(int j=0; j<MAT; j++) {

if(strcmp(alunos[i].disciplinas[j].situacao, "reprovado") == 0) {
            qtdeRe++;
        } else {
            qtdeAp++;
        }
    }
}

cabecalho();
printf("\n\n\t\t RELATORIO DE CADASTROS");
printf("\n\n\t\t Quantidade de alunos cadastrados:

%d", qtde);

printf("\n\t\t Quantidade de alunos aprovados: %d",
qtdeAp);

printf("\n\t\t Quantidade de alunos reprovados:

%d", qtdeRe);

printf("\n\t\t Faixa etaria dos alunos cadastrados
de %d a %d anos", menorIdade, maiorIdade);
printf("\n\t\t Media das idades cadastradas %.2f",
medialdade/qtde);

printf("\n\n\t\t Obs: lembrando que o aluno possui
8 disciplinas");

printf("\n\t\t Ou seja, ele pode ter sido aprovado
em tres e reprovado nas demais");
```

```
        getch();
        break;
    case '4':
        //VOLTANDO AO MENU PRINCIPAL
        break;
    case '5':
        exit(1);
        break;
    default:

        printf("\n\n\t Digite uma opcao valida\n");
        printf("\t Aperte qualquer tecla para tentar
novamente.");

        getch();

    }
} while(opcao != '4');
}

//Funcao para pesquisar o nome digitado com o nome armazenado
int pesquisarNome(char nome[]){

    int cont = 0, indice = 0;;
    //Loop para comparar o nome digitado com o nome armazenado na
    estrutura alunos
    for(int i=0; i<MAX; i++){
        if(strcmp(nome, alunos[i].nomeAluno) == 0){
            indice = i;
            cont++;
            //Retornando o indice onde o aluno x foi encontrado
            return (indice);
        }
    }

    if(cont == 0) {
        indice = -1;
        //Retornando o indice -1 caso o nome digitado nao tenha sido
    encontrado
        return (indice);
    }
}
```

```
}
```

```
//Funcao para pesquisar o ID digitado com o ID armazenado
```

```
int pesquisarId(int id, int qtde){
```

```
    int cont = 0, indice = 0;
```

```
    //Loop para comparar o ID digitado com o ID armazenado na estrutura
```

```
    alunos
```

```
    for(int i=0; i<qtde; i++) {
```

```
        if(id == alunos[i].id) {
```

```
            indice = i;
```

```
            cont++;
```

```
            //Retornando o indice onde o aluno x foi encontrado
```

```
            return(indice);
```

```
        }
```

```
    }
```

```
    if (cont == 0) {
```

```
        indice = -1;
```

```
        //Retornando o indice -1 caso o nome digitado nao tenha sido
```

```
    encontrado
```

```
        return(indice);
```

```
    }
```

```
}
```

```
//Funcao para imprimir os cadastraos armazenados
```

```
void imprimirCadastro(int qtde) {
```

```
    cabecalho();
```

```
    //Imprimindo todos os cadastros, excetos quais estão inativos no
```

```
    sistema
```

```
    printf("\n\t\t\t\t\tDADOS DOS ALUNOS CADASTRADOS\n\n");
```

```
    for (int i=0; i<qtde; i++) {
```

```
        if(alunos[i].inativo == 1) {
```

```
            continue;
```

```
        }
```

```
        printf("\n\t ID: %d\n", alunos[i].id);
```

```
        printf("\n\t Nome: %s", &alunos[i].nomeAluno);
```

```
printf("\n\t Idade: %d\n", alunos[i].idade);
printf("\n\t Media das Materias e Situacao Escolar ");

for(int j=0; j<MAT; j++) {
    printf("\n\t %s", &alunos[i].disciplinas[j].materia);
    printf(" MEDIA: %.2f -", alunos[i].disciplinas[j].media);
    printf(" SITUACAO: %s ",
&alunos[i].disciplinas[j].situacao);
}
printf("\n\t ----- \n");
printf("\n\t ----- \n");
}
getch();
}
```

```
//Funcao para imprimir o aluno pesquisado, por nome ou por ID,
//recebendo como parametro o indice onde se encontra
void exibirAlunoPesquisado(int indice, int qtde) {
```

```
    cabecalho();

    printf("\n ----- DADOS DO(A) ALUNO(A) ----- \n");
    printf("\n ID: %d ", alunos[indice].id);
    printf("\n Nome: %s", &alunos[indice].nomeAluno);
    printf("\n Idade: %d ", alunos[indice].idade);
    printf("\n Media das Materias e Situacao Escolar ");

    for(int j=0; j<MAT; j++) {
        printf("\n %s", &alunos[indice].disciplinas[j].materia);
        printf(" MEDIA: %.2f -", alunos[indice].disciplinas[j].media);
        printf(" SITUACAO: %s ",
&alunos[indice].disciplinas[j].situacao);
    }
    printf("\n ----- \n");

    getch();
}
```

```
//Funcao para ordenar alunos por nome ou por ID
```

//recebendo como parametro a escolha que é de acordo com a opcao digitada
pelo usuario no menu principal

```
void insertionSort(int qtde, int escolha) {  
  
    int i, j;  
    Aluno aux;  
    for(i=1; i<qtde; i++) {  
        aux = alunos[i];  
        j = i-1;  
  
        if (escolha == 1) {  
            while ((j >= 0) && (strcmp(alunos[j].nomeAluno,  
aux.nomeAluno) > 0)) {  
                alunos[j+1] = alunos[j];  
                j--;  
            }  
        } else {  
            while ((j >= 0) && (alunos[j].id > aux.id)) {  
                alunos[j+1] = alunos[j];  
                j--;  
            }  
        }  
        alunos[j+1] = aux;  
    }  
}
```

//Funcao para preencher a disciplina com os nomes das materias abaixo

```
void preencherDisciplina(Aluno *aluno) {  
    strcpy(aluno->disciplinas[0].materia, "Biologia ");  
    strcpy(aluno->disciplinas[1].materia, "Fisica ");  
    strcpy(aluno->disciplinas[2].materia, "Geografia ");  
    strcpy(aluno->disciplinas[3].materia, "Historia ");  
    strcpy(aluno->disciplinas[4].materia, "Ingles ");  
    strcpy(aluno->disciplinas[5].materia, "Matematica");  
    strcpy(aluno->disciplinas[6].materia, "Portugues ");  
    strcpy(aluno->disciplinas[7].materia, "Quimica ");  
}
```

//Funcao para transformar todas as letras da string, recebida como parametro,
em minusculas


```
void lowerString(char str[]){
    for(int i=0; i<strlen(str); i++){
        if(str[i]>= 65 || str[i]<= 90){
            str[i] = tolower(str[i]);
        }
    }
}
```

//Funcao para validar o nome da string, recebida como parametro, se contem apenas letras ou se há algum espaço

```
int validaNome(char nome[], int tamanho) {
    int i;
    for(i=0; i<strlen(nome); i++) {
        if(nome[i] != ' ' && !isalpha(nome[i])) {
            return 0;
        }
        if(tamanho > 49) {
            return 0;
        }
    }
    return 1;
}
```

```
void cabecalho() {
    system("cls");
    printf("\t-----\n");
    printf("\t\t\t\t\t CADASTRO ESCOLAR\n");
    printf("\t-----\n\n");
}
```

```
void flush() {
    int ch;
    while ( ( ch = fgetc ( stdin ) ) != EOF && ch != '\n' ) {}
}
```

```
void gotoxy(int x,int y){
    COORD c;
    c.X = x;
```

```
c.Y = y;  
SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),c);  
};
```