Sumário

[1. Introdução 2](#_Toc163456866)

[2. Implementação 2](#_Toc163456867)

[3. Testes 5](#_Toc163456868)

[4. Conclusão 6](#_Toc163456869)

[Referências 6](#_Toc163456870)

[Anexos 7](#_Toc163456871)

[main.c 7](#_Toc163456872)

[main.h](#_main.h). 7

[funcoes](#_funcoes.c).c 7

## Introdução:

Este trabalho prático em grupo tem por finalidade a implementação, em C, de um algoritmo para leitura de um arquivo chamado DadosEntrada.csv, onde em cada linha contém o nome, o telefone (com prefixo), o curso, e as notas que cada um dos alunos matriculados em uma universidade obteve durante um semestre. A partir deste arquivo, o código será responsável por realizar a manipulação de dados e gerar um novo arquivo chamado SituacaoFinal.csv com a situação final de cada aluno. Sendo assim, o programa irá ler os dados de cada aluno, identificando o nome e as notas para determinar a nota média (com duas casas decimais) e a situação final do aluno (APROVADO, caso a nota média seja maior ou igual a 7.0 ou REPROVADO, caso a nota média seja menor do que 7.0).

#### GitHub:

* <https://github.com/fiorellizz/Atividades-UCB/tree/main/Estrutura%20de%20dados/Projetos/Manipulando%20arquivos>
* <https://github.com/AnaBeatrizChaves/EstruturaDados/tree/main/Projetos/ManipulandoArquivos>

## Implementação:

Criado a partir da ferramenta VSCODE, o projeto contém um arquivo principal chamado **main.c** , onde estão organizados os comandos que servem como ponto de partida e controle do programa para direcionar as chamadas das funções, além de outros dois arquivos auxiliares chamado **main.h** e **funcoes.c**.

**EXPLICAÇÃO MAIN.C**

Este arquivo começa com as 3 importações clássicas da biblioteca C necessárias para o correto funcionamento do programa: **<stdio.h>**, **<stdlib.h>**, **<string.h>**. Logo abaixo, há os **#include "main.h"** e **#include "funcoes.c"**, que indicam a importação de outros dois arquivos criados na mesma pasta, chamados **main.h** e **funcoes.c**, respectivamente. Ao iniciar a função principal **int main()**, temos a declaração do ponteiro **\*arquivo** do tipo **FILE** usado para a manipulação de arquivos e a declaração da variável **i** do tipo **INT**, que será utilizada posteriormente na estrutura de repetição FOR. Em seguida, a linha **arquivo = fopen("DadosEntrada.csv", "r");** significa que irá abrir o arquivo **DadosEntrada.csv** em modo leitura (“**r**”) para identificar os dados dos alunos. Nas linhas **int linhas = contarLinhas(arquivo);** e **struct Aluno X[linhas];** temos a declaração de uma função **contarLinhas** presente em um arquivo chamado **funcoes.c** e a declaração da **struct aluno** que se encontra no arquivo **main.h**. A estrutura **IF** realiza a conferência se o arquivo realmente foi aberto, senão irá aparecer na tela “**Erro ao abrir o arquivo!**”. O **fscanf(arquivo, “%\*[^\n]\n”);** será responsável por pular/ignorar a primeira linha do arquivo DadosEntrada.csv referente a **Nome, Telefone, Curso, Nota1, Nota2** dos alunos. Em seguida, há a chamada da função **lerArquivo** com os parâmetros **arquivo, X, &i** que se encontra no arquivo **funcoes.c**. Logo em seguida, o arquivo é fechado e de acordo com os dados coletados, é criado um novo arquivo a partir da linha de código **arquivo = fopen("SituacaoFinal.csv", "w");** sendo o “**w**” correspondente ao modo de escrita, pois irá inserir o nome, a média e a situação final de cada aluno encontrado. Novamente, temos a estrutura condicional **IF** para verificar se o arquivo foi criado. Em seguida, é chamada a função **escreveArquivo**, também presente no arquivo **funcoes.c**, na qual calculará a média de cada um dos alunos e dirá se foram ou não aprovados. Aparece uma mensagem de sucesso na tela e, por fim, o arquivo é fechado.

**CÓDIGO MAIN.C**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "main.h"

#include "funcoes.c"

int main() {

FILE \*arquivo;

int i;

arquivo = fopen("DadosEntrada.csv", "r");

int linhas = contarLinhas(arquivo);

struct Aluno X[linhas];

if(arquivo == NULL) {

printf("Erro ao abrir o arquivo!");

exit(ERRO);

}

fscanf(arquivo, "%\*[^\n]\n");

system("cls");

lerArquivo(arquivo, X, &i);

fclose(arquivo);

arquivo = fopen("SituacaoFinal.csv", "w");

if(arquivo == NULL) {

printf("Erro ao criar o arquivo!");

exit(ERRO);

}

escreveArquivo(arquivo, X, &i);

printf("Executado com sucesso");

fclose(arquivo);

return 0;

}

**EXPLICAÇÃO MAIN.H**

Este arquivo é utilizado como cabeçalho da **main.c**, onde contém a implementação de uma **struct** chamada Aluno, com os campos referentes ao arquivo DadosEntrada.csv, sendo eles: **char nome, char telefone, char curso, float nota1, float nota2** dos alunos. A seguir, tem-se a definição de ERRO = 1 e MAX = 1024 para ser usado nas funções e por fim 3 funções e seus respectivos parâmetros, sendo elas: **contarLinhas, lerArquivo, escreveArquivo**, implementadas no arquivo **funcoes.c**.

**CÓDIGO MAIN.H**

struct Aluno{

char nome[100];

char telefone[100];

char curso[20];

float nota1;

float nota2;

};

#define ERRO 1

#define MAX 1024

int contarLinhas(FILE \*arquivo);

void lerArquivo(FILE \*arquivo, struct Aluno \*X, int \*i);

void escreveArquivo(FILE \*arquivo, struct Aluno \*X, int \*i);

**EXPLICAÇÃO FUNCOES.C**

Este arquivo contém especificadamente algumas das funções utilizadas no programa. Nele está a implementação da função **contarLinhas**, responsável pela contagem das linhas do arquivo DadosEntrada.csv para a construção da struct. Em seguida, a função **lerArquivo** percorrerá linha por linha, separando os dados de cada aluno pela vírgula até o final do arquivo (**EOF - End of File**), através do código: **for ((\*i) = 0; fscanf(arquivo, "%[^,],%[^,],%[^,],%f,%f", X[\*i].nome, X[\*i].telefone, X[\*i].curso, &X[\*i].nota1, &X[\*i].nota2) != EOF; (\*i)++);**. Por fim, a função **escreveArquivo** será responsável pelo preenchimento das informações do novo arquivo SituacaoFinal.csv, de acordo com os dados do arquivo DadosEntrada.csv para calcular a média e determinar a situação final (aprovado ou reprovado), utilizando um laço FOR para percorrer os campos e um **IF** para verificar se está aprovado ou reprovado.

**CÓDIGO FUNCOES.C**

int contarLinhas(FILE \*arquivo) {

int quantidade = 0;

char linha[MAX];

while (fgets(linha, MAX, arquivo) != NULL){

quantidade++;

}

rewind(arquivo);

return quantidade;

}

void lerArquivo(FILE \*arquivo, struct Aluno \*X, int \*i) {

for ((\*i) = 0; fscanf(arquivo, "%[^,],%[^,],%[^,],%f,%f", X[\*i].nome, X[\*i].telefone, X[\*i].curso, &X[\*i].nota1, &X[\*i].nota2) != EOF; (\*i)++);

}

void escreveArquivo(FILE \*arquivo, struct Aluno \*X, int \*i) {

for (int j = 0; j < (\*i); j++) {

fprintf(arquivo, "%s, %.2f, ", X[j].nome, ((X[j].nota1 + X[j].nota2) / 2));

if ((X[j].nota1 + X[j].nota2) / 2 >= 7.0){

fprintf(arquivo, "APROVADO");

} else {

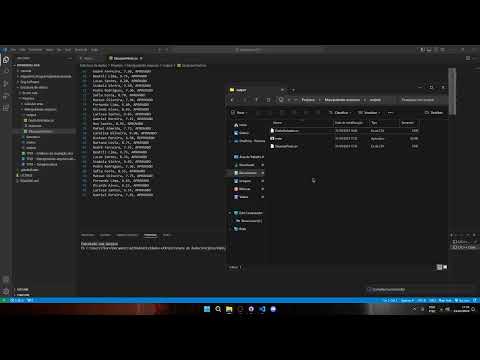
fprintf(arquivo, "REPROVADO");

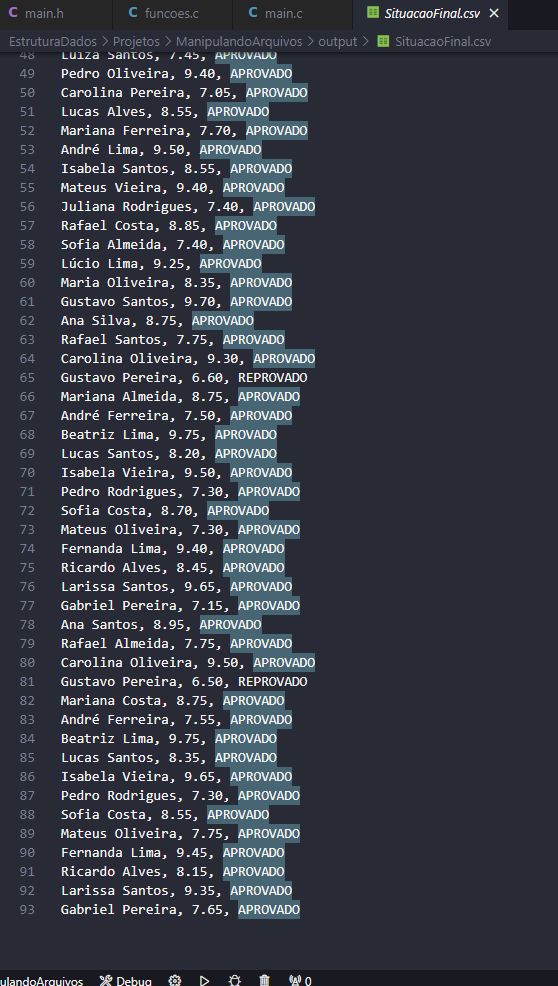
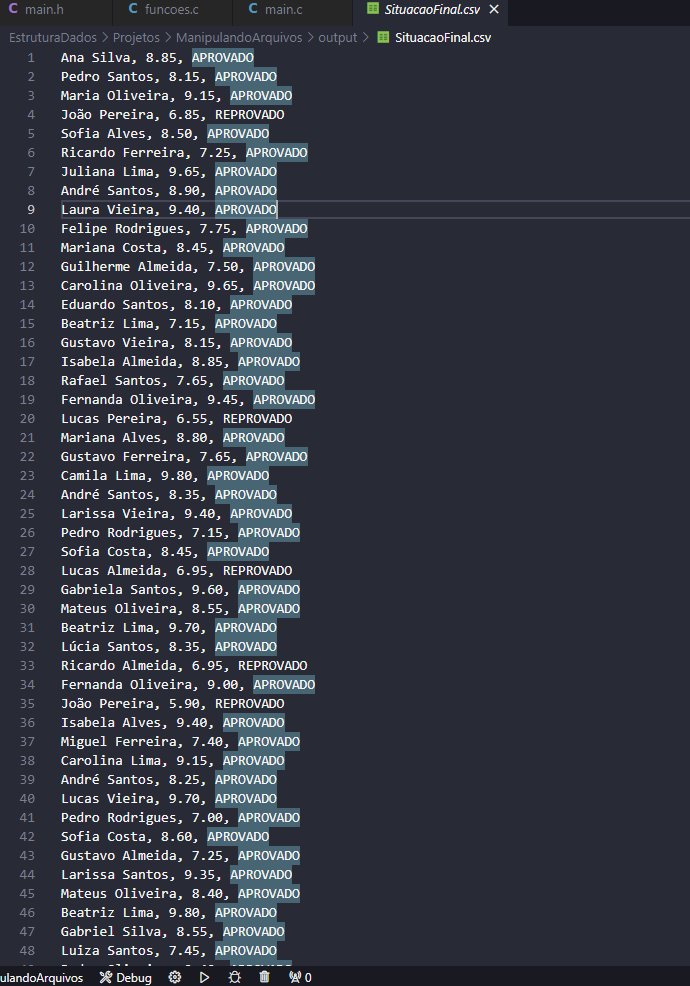
}

}

}

## Testes

[](https://www.youtube.com/embed/OEFGvp7X91k?feature=oembed)Segue a evidência de teste do programa em funcionamento no formato de vídeo e print da saída do terminal.



## Conclusão

De forma geral, conclui-se que a implementação deste programa foi fundamental para a fixação dos assuntos trabalhados em sala de aula, assim como as interações entre a manipulação de dados e a geração de um novo arquivo texto. No que diz respeito às dificuldades, pode-se destacar algumas em relação à forma de pensar na estrutura do código com base em um arquivo .csv, levando em conta apenas determinados tipos de dados do arquivo, assim como a chamada de funções em arquivos .C separados.

## Referências

Projeto implementado entre os membros do grupo e com pesquisa em slides disponibilizado no AVA pelo professor Marcelo.

## Anexos

Na pasta Output deverá constar o arquivo **DadosEntrada.csv** junto ao executável **main.exe** para o correto funcionamento do programa.



### main.c

### 

### main.h

### 

### funcoes.c

**