

## Sumário

1. Introdução:	2
2. Implementação:	2
3. Testes	2
4. Conclusão	4
Referências	4
Anexos	4
tp01.c	4
DadosEntrada.csv	4

# 1. Introdução:

O objetivo deste trabalho é ler os dados de um arquivo, processá-los e imprimi-los em um arquivo distinto. Os dados fornecidos serão: nome, telefone, curso e 2 notas de um aluno de uma universidade. Os dados que o programa devolverá serão: nome, média das notas e situação final do aluno.

GitHub:

<https://github.com/arthursleite/Estrutura-Dados-UCB/tree/main/tp01>

# 2. Implementação:

Foi utilizada a linguagem C com assistência da IDE Visual Studio Code e compilador GCC. Os arquivos .csv foram manipulados através de métodos fornecidos pela biblioteca stdio.h e a manipulação de algumas strings foi feita por métodos fornecidos pela biblioteca string.h.

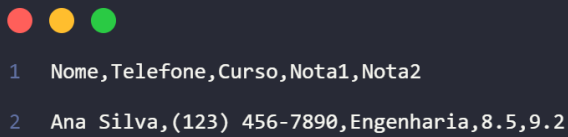
A implementação foi feita utilizando somente um arquivo .c contendo toda a lógica, funções e variáveis do programa.

# 3. Testes

Na imagem abaixo, nas linhas 1 e 2, é possível ver quais são as bibliotecas que serão utilizadas. Nas linhas 4 e 5, a função main é chamada. Da 6 a 12, é possível ver a declaração das variáveis do tipo FILE, char e double. Nas linhas 14 e 15, os arquivos são abertos para leitura e escrita, respectivamente. Da linha 16 a 20, é checado se algum dos arquivos é nulo, caso sim, uma mensagem de erro será enviada e a função main() retornará 1.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      FILE *arquivoIn;
7      FILE *arquivoOut;
8      char nome[75];
9      char telefone[15];
10     char curso[40];
11     double nota1;
12     double nota2;
13
14     arquivoIn = fopen("../DadosEntrada.csv", "r");
15     arquivoOut = fopen("../SituacaoFinal.csv", "w");
16     if (arquivoIn == NULL || arquivoOut == NULL)
17     {
18         printf("Erro ao abrir o arquivo.\n");
19         return 1;
20     }
```

A primeira linha do arquivo CSV (primeira imagem abaixo) contém o a descrição de cada dado e os dados começam a partir da segunda linha.

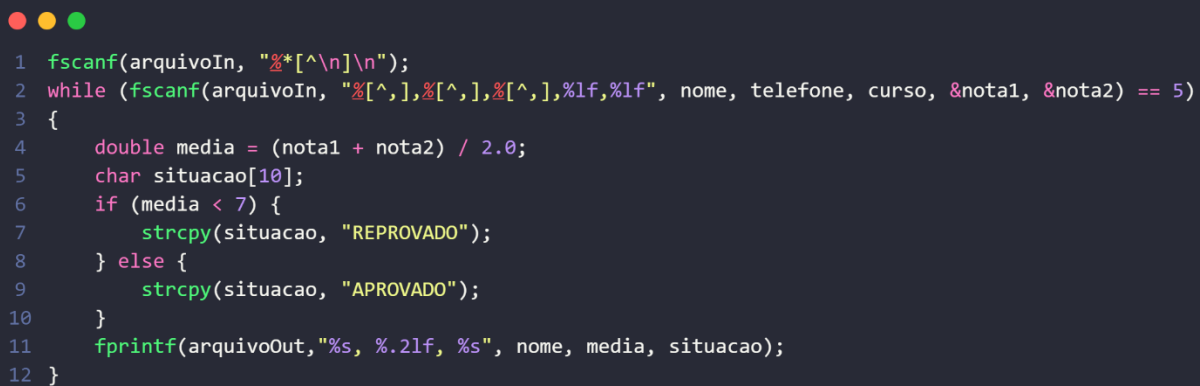


```
1 Nome,Telefone,Curso,Nota1,Nota2
2 Ana Silva,(123) 456-7890,Engenharia,8.5,9.2
```

Para pular essa linha de descrições mencionada acima, foi preciso usar um `fscanf()` que leia os dados da primeira linha do arquivo `DadosEntrada`, não os armazene em nenhuma variável e pule para a próxima linha assim que encontrar um caractere `\n`.

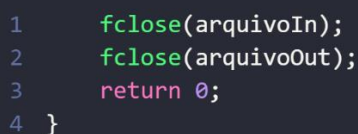
Na linha 2 é utilizado uma função `while()` que verificará se os 5 dados foram verificados e armazenado corretamente, caso resulte em falso, o loop será interrompido na linha 12.

Dentro do `while()`, na linha 4 é declarada uma variável `media` que armazena a média entre as duas notas do aluno. Nas linhas 6 a 10 o programa verifica a média do aluno, e caso seja menor que 7, ele será reprovado, caso não, será aprovado. Esse valor em formato de texto será armazenado na variável declarada na linha 5. Na linha 11 é feita a impressão do nome, média das notas e situação final (aprovado ou reprovado) do aluno no arquivo `SituacaoFinal`.



```
1 fscanf(arquivoIn, "%*[^\\n]\\n");
2 while (fscanf(arquivoIn, "%[^,],%[^,],%[^,],%lf,%lf", nome, telefone, curso, &nota1, &nota2) == 5)
3 {
4     double media = (nota1 + nota2) / 2.0;
5     char situacao[10];
6     if (media < 7) {
7         strcpy(situacao, "REPROVADO");
8     } else {
9         strcpy(situacao, "APROVADO");
10    }
11    fprintf(arquivoOut,"%s, %.2lf, %s", nome, media, situacao);
12 }
```

Na imagem abaixo, nas linhas 1 e 2, os arquivos são fechados. Na linha 4, o programa retornará caso o resto do programa rode com sucesso. E na linha 4 é fechada a função `main()`.



```
1 fclose(arquivoIn);
2 fclose(arquivoOut);
3 return 0;
4 }
```

## 4. Conclusão

O programa funcionou perfeitamente de acordo com o que foi pedido no enunciado. O que poderia ser melhorado em próximas versões seria uma forma de formatar a leitura dos dados igualmente efetiva à atual porém com legibilidade e entendimento melhor para qualquer um que leia o código.

## Referências

<https://en.cppreference.com/w/c/io>

<https://stackoverflow.com/questions/41125193/using-fscanf-to-read-from-a-csv-file-in-c>

<https://stackoverflow.com/questions/30432856/best-way-to-get-number-of-lines-in-a-file-c>

<https://stackoverflow.com/questions/2799612/how-to-skip-a-line-when-fscanning-a-text-file>

## Anexos

**tp01.c**

***DadosEntrada.csv***

***SituacaoFinal.csv***