



# Projeto prático

Matheus de Paula Megale

Eduardo Ruan Fonseca

Lucas de Oliveira Pereira

Sistemas de Informação  
Universidade Federal de Lavras

Lavras / 2023

---

## Visão geral

### Breve introdução sobre o trabalho em si:

Este trabalho é um programa em C++ que tem por objetivo criar um sistema de cadastro em arquivos com ordenação. Primeiramente, um arquivo CSV, chamado “base9\_OK.csv” (Imagem 1) foi convertido para binário (Imagem 2). Depois, no programa principal, foram utilizadas cinco bibliotecas (iostream, fstream, sstream, string, cstring), um registro (nomeado “registro”) que possui cinco campos: “campo1”, do tipo long long int, “campo2”, que é um vetor de caracteres de tamanho 10, “campo3”, que também é um vetor de caracteres, porém de tamanho 200, “campo4”, do tipo float e “campo5”, também do tipo float. Ademais, também foram utilizadas catorze funções, cada uma exercendo um papel específico, sendo que algumas delas dependem de outras, além da função principal (int main).

A Imagem 1 mostra um pequeno trecho da base 9 no formato CSV, enquanto a Imagem 2 mostra um pequeno trecho da mesma base convertida para o formato binário.

```

1 1039201260019;Negativa;10 MG COM CT BL AL/PLAS TRANS X 12;7,79;8,73
2 1039201270073;Positiva;15 MG/ML SUS OR CT FR PLAS OPC GOT X 20 ML;10,85;10,85
3 1039201270081;Positiva;15 MG/ML SUS OR CT 50 FR PLAS OPC GOT X 20 ML (EMB HOSP);560,94;560,94
4 1039201320038;Negativa;50 MG/G CREM DERM CT BG AL X 10 G;13,68;15,32
5 1039201350018;Positiva;5 MG + 50 MG COM CT BL AL PLAS INC X 30;13,48;13,48
6 1039201400023;Negativa;5500UI/G + 990UI/G + 150MG/G POM CT BG AL X 45G;11,95;13,38
7 1039201400031;Negativa;5500UI/G + 990UI/G + 150MG/G POM CX 50 BG AL X 45G;458,57;513,6
8 1039201470048;Positiva;40 MG COM CT BL AL PLAS INC X 500 (EMB HOSP);79,07;79,07
9 1039201470056;Positiva;80 MG COM CT BL AL PLAS INC X 30;7,06;7,06
10 1039201470080;Positiva;40 MG COM CT BL AL PLAS INC X 30;4,39;4,39

```

Imagem 1

```

1 8NULNULNegativaNULc110 MG COM CT BL AL/PLAS TRANS X 12NULSTXOSONNULNULSTXOXbNUL:OOTEOTbNULSTXONETE
2 ;?NULNULPositivaNULc1100 MG COM CT BL AL PLAS TRANS X 30NULbNULMB HOSP)NULNS X 100NULGNULNULbNULSTXONETEONSONNULNUL
3 @U7Oo?NULNULPositivaNULc1100 MG COM REV CT ENV AL X 12NULbGNULT AMB X 100 MLNUL X 100NULGNULNULbNULSTXONETE
4 A=
5 AU7Oo$@NULNULPositivaNULc1100 MG COM CT BL AL PLAS TRANS X 12NULAMB X 100 MLNUL X 100NULGNULNULbNULSTXONETEONSi
6 A
7 AU7Oo1@@NULNULPositivaNULc110 MG COM CT ENV AL X 30NULRANS X 20NULX 15 MLNULbMLNUL X 100NULGNULNULbNULSTXONETEONC
8 ETEdC2C
9 ETEdC2CU7OoB@NULNULPositivaNULc110 MG COM CT BL AL PLAS AMB X 500NUL 500NULX 490NULMLNUL X 100NULGNULNULbNULSTXONETE
10 B=

```

Imagem 2

**Descrição em alto nível (isto é, sem código) de todas as estruturas utilizadas explicando a lógica do programa:**

O arquivo “ConverteDeCSVParaBinario.cpp”, é um programa em C++ que lê dados de um arquivo CSV chamado "base9\_OK.csv" e os grava em um arquivo binário chamado "base9\_binario.dat". Os dados são armazenados em uma estrutura "registro", que possui cinco campos: campo1 é um long long int, o campo2 e campo3 são vetores de caracteres de tamanho 10 e 200, respectivamente. O campo4 e campo5 são floats.

O programa lê cada linha do arquivo CSV, divide-o em cinco campos usando o separador de ponto e vírgula e armazena nos campos correspondentes da estrutura do registro. Ele converte o separador decimal de uma vírgula para um ponto para o campo4 e campo5 usando a função “replace” da biblioteca de “algorithm”. Em seguida, ele grava a estrutura de registro no arquivo binário usando a função ofstream. Ele utiliza fluxos de arquivos de entrada e saída da biblioteca “fstream” e fluxos de strings da biblioteca “sstream” para ler e manipular os dados. Ele também usa a função “strcpy” da biblioteca “cstring” para copiar os vetores de caracteres. Por fim, imprime uma mensagem na tela quando a conversão é concluída e fecha os arquivos de entrada e saída.

Já o arquivo “TrabalhoConcluido.cpp”, é um programa em C++ que fornece um menu para manipular um arquivo binário contendo registros com cinco campos, com mesmo formato do programa “ConverteDeCSVParaBinario.cpp”, o programa logo de início entra em um loop que exibe o menu e lê a escolha do usuário na entrada padrão. Se o usuário escolher a opção 0, o programa, define a variável “fecha” que é uma variável booleana como verdadeiro, encerrando o loop e imprimindo uma mensagem de fechamento. Caso contrário, o programa se ramifica com base na escolha do usuário.

Se o usuário escolher a opção 1, irá imprimir o arquivo todo, é chamada à função “imprime\_arquivo”, que lê todos os registros do arquivo binário e os imprime na saída padrão. A função verifica se o valor de campo1 é maior que 0 para cada registro lido do arquivo. Se campo1 for maior que 0, a função imprime os valores de todos os cinco campos na saída padrão, separados por ponto e vírgula e com um caractere de nova linha no final. Depois que todos os registros do arquivo forem lidos e processados, a função fecha o arquivo.

Se a opção escolhida for 2, irá ordenar o arquivo, é solicitado ao usuário que escolha qual campo ele deseja ordenar, através da variável “escolha”. Se “escolha” for igual a 1, o programa abre o arquivo "base9\_binario.dat" com algumas permissões e chama a função ordenaCampo1(), passando como parâmetro o objeto arquivo e “elementos\_excluidos”, que é quantos elementos foram excluídos do arquivo, através

da função “excluir”. A função lê os dados do arquivo e passa para um vetor exceto aqueles marcados como excluídos, faz o ordenamento, e depois escreve no arquivo binário novamente os dados ordenados e sem os marcados como excluídos. Se “escolha” for igual 4, o programa realiza o mesmo procedimento, porém é chamada a função `ordenaCampo4()`, e ordena o arquivo através do `campo4`. Em seguida, o arquivo é fechado.

Se a opção escolhida for 3, será feita a remoção de algum elemento do arquivo, é pedido ao usuário qual a posição do arquivo ele deseja remover um elemento, mas antes imprime na tela qual o intervalo de posições que pode ser feita remoção. Logo em seguida, chama a função “excluir”, passando no fluxo de arquivo, o número de registros, a posição a ser removida e uma referência a uma variável inteira “`elementos_excluidos`”. No corpo da função, primeiro é feita uma verificação para garantir que a posição a ser excluída é válida (ou seja, está dentro do intervalo de posições existentes no arquivo). Caso seja válida, a função lê o registro daquela posição, multiplica o valor do `campo1` por -1 (para indicar que o registro foi excluído) e escreve o registro de volta na mesma posição no arquivo. Por fim, a função retorna a posição inicial do arquivo e incrementa o contador de elementos excluídos em 1. É importante notar que a função não realiza nenhuma remoção física do registro do arquivo, ela apenas “marca” o registro como excluído. Em seguida, o arquivo é fechado.

Se a opção escolhida for 4, será inserido algum elemento no arquivo, é chamada a função “`inserir_elemento`”, ela começa verificando se há registros excluídos no arquivo, ou seja, com o `campo 1` negativo. Se houver, a função percorre o arquivo procurando pelo primeiro registro excluído e, quando encontra, solicita ao usuário que digite os novos valores para cada campo do registro. Em seguida, a função sobrescreve o registro excluído com os novos valores digitados pelo usuário. Caso não haja registros excluídos no arquivo, a função insere o novo registro no final do arquivo. Para isso, a função posiciona o ponteiro de escrita no final do arquivo e solicita ao usuário que digite os novos valores para cada campo do registro. Em seguida, a função escreve o novo registro no final do arquivo. Por fim, a função fecha o arquivo.

Se a opção escolhida for 5, feita a busca de um elemento no arquivo, é solicitado ao usuário por qual campo do registro ele deseja realizar a busca, se a opção escolhida por 1, é pedido para ele escrever o ID do elemento do `campo1` que deseja buscar, logo em seguida é chamada a função “`buscarElementoCampo1`”, que lê o arquivo registro por registro e verifica se o valor a ser encontrado corresponde ao valor do primeiro

campo do registro. Se houver correspondência, a posição do registro no arquivo é armazenada e o conteúdo do registro é impresso na tela. Caso o valor não seja encontrado, é exibida uma mensagem informando que o elemento não foi encontrado. Já se a opção escolhida for 4, é pedido o ID do elemento do campo4, logo em seguida é chamada à função “buscarElementoCampo4”, que realiza o mesmo processo da função da busca pelo campo1, porém compara o valor do registro do quarto campo do registro.

Se a opção escolhida for 6, feita a impressão de um trecho do arquivo, é solicitado ao usuário a posição inicial da impressão logo em seguida a posição final, é chamada a função “imprimeTrecho”, que faz a impressão do trecho solicitado, caso haja algum elemento marcado como excluído é imprimido na tela “Este elemento foi removido!!”. Depois, o arquivo é fechado.

Se a opção escolhida for 7, é feita a exportação dos dados do arquivo binário para um arquivo CSV, é chamada a função “exportaParaArquivoCSV”, que lê o arquivo binário e escreve em um arquivo que foi criado através da função “ofstream” chamado “ArquivoImportadoDeBinarioParaCSV.csv”, todos os dados são escritos exceto os marcados como excluídos.

### **Uma descrição da ordem dos dados armazenados no arquivo:**

Cada linha do arquivo contém um total de cinco dados, sendo elas, respectivamente, dos tipos long long int, vetor de char, vetor de char, float e float.

### **Abordar de forma sucinta e objetiva os acertos e erros durante o desenvolvimento do trabalho:**

Tivemos dificuldades da parte de inserção de novos dados. Sempre que era solicitado para inserir o campo 3, que é um vetor de caracteres, e era digitado o caractere espaço, o programa inseria todos os elementos digitados depois do espaço no campo 4 e ele, por ser do tipo float, ocasionava em um looping infinito. Com o auxílio dos monitores, esse erro foi corrigido.

Também erramos por acharmos, em primeiro plano, que o programa deveria realizar as manipulações em memória primária. Depois de terminado o trabalho, tivemos que refazê-lo.

### **Fazer uma conclusão, apresentando os resultados obtidos:**

Ordenação de elementos:

Para a ordenação, optamos por ordenar os campos 1 e 4, sendo que a Imagem 3 representa a ordenação do campo 1 e a Imagem 4, o campo 4.

```

1  1039201260019;Negativa;10 MG COM CT BL AL/PLAS TRANS X 12;7.79;8.73
2  1039201270073;Positiva;15 MG/ML SUS OR CT FR PLAS OPC GOT X 20 ML;10.85;10.85
3  1039201270081;Positiva;15 MG/ML SUS OR CT 50 FR PLAS OPC GOT X 20 ML (EMB HOSP);560.94;560.94
4  1039201320038;Negativa;50 MG/G CREM DERM CT BG AL X 10 G;13.68;15.32
5  1039201350018;Positiva;5 MG + 50 MG COM CT BL AL PLAS INC X 30;13.48;13.48
6  1039201400023;Negativa;5500UI/G + 990UI/G + 150MG/G POM CT BG AL X 45G;11.95;13.38
7  1039201400031;Negativa;5500UI/G + 990UI/G + 150MG/G POM CX 50 BG AL X 45G;458.57;513.6
8  1039201470048;Positiva;40 MG COM CT BL AL PLAS INC X 500 (EMB HOSP);79.07;79.07
9  1039201470056;Positiva;80 MG COM CT BL AL PLAS INC X 30;7.06;7.06
10 1039201470080;Positiva;40 MG COM CT BL AL PLAS INC X 30;4.39;4.39

```

Imagem 3

```

1  1049711490048;Negativa;40 MG/G GRAN CT 15 ENV POLIET X 5 G&nbsp;1;25.17
2  1048100960022;Positiva;25 MG COM CT BL AL PLAS PVC TRANS X 30;3.05;3.05
3  1046502500011;Positiva;0,4 MG/ML XPE CT FR VD AMB X 120 ML + CP MED;3.38;3.38
4  1039201650011;Positiva;100 MG COM CT BL AL PLAS INC X 20;3.68;3.68
5  1048100100023;Negativa;500 MG/ML SOL OR CT FR PLAS OPC GOT X 20 ML;3.71;4.16
6  1047300390019;Negativa;0,08ML/ML XPE CT FR PLAS AMB X 100ML + COP;3.93;4.4
7  1039201650038;Positiva;200 MG COM CT BL AL PLAS INC X 20;4.11;4.11
8  1049701910036;Positiva;2 MG/ML SOL OR CT FR PLAS OPC GOT X 20 ML;4.35;4.35
9  1049712080013;Positiva;2 MG/ML SOL OR CT FR GOT PLAS PEBD OPC X 20 ML;4.35;4.35
10 1049711680017;Negativa;5 MG COM REV LIB RETARD CT BL AL PLAS TRANS X 20;4.35;4.87

```

Imagem 4

#### Remoção de elementos:

A Imagem 5 mostra uma linha do arquivo que foi removida. A mensagem de remoção apenas aparece quando solicitada a impressão de um trecho específico do arquivo (opção 6).

```

1  Este elemento foi removido!!
2  1039201270073;Positiva;15 MG/ML SUS OR CT FR PLAS OPC GOT X 20 ML;10.85;10.85
3  1039201270073;Positiva;15 MG/ML SUS OR CT FR PLAS OPC GOT X 20 ML;10.85;10.85
4  1039201270081;Positiva;15 MG/ML SUS OR CT 50 FR PLAS OPC GOT X 20 ML (EMB HOSP);560.94;560.94
5  1039201320038;Negativa;50 MG/G CREM DERM CT BG AL X 10 G;13.68;15.32
6  1039201350018;Positiva;5 MG + 50 MG COM CT BL AL PLAS INC X 30;13.48;13.48
7  1039201400023;Negativa;5500UI/G + 990UI/G + 150MG/G POM CT BG AL X 45G;11.95;13.38
8  1039201400031;Negativa;5500UI/G + 990UI/G + 150MG/G POM CX 50 BG AL X 45G;458.57;513.6
9  1039201470048;Positiva;40 MG COM CT BL AL PLAS INC X 500 (EMB HOSP);79.07;79.07
10 1039201470056;Positiva;80 MG COM CT BL AL PLAS INC X 30;7.06;7.06

```

Imagem 5

#### Inserção de um novo elemento:

A Imagem 6 mostra a inserção de novos elementos no arquivo, sendo todos eles dos mesmos tipos dos outros elementos do arquivo.



```

1 1049711490048;Negativa;"40 MG/G GRAN CT 15 ENV POLIET X 5 G&nbsp;1;25.17
2 1048100960022;Positiva;25 MG COM CT BL AL PLAS PVC TRANS X 30;3.05;3.05
3 1046502500011;Positiva;0,4 MG/ML XPE CT FR VD AMB X 120 ML + CP MED;3.38;3.38
4 1039201650011;Positiva;100 MG COM CT BL AL PLAS INC X 20;3.68;3.68
5 111111111111;Qualquer;LAVRAS/MG - 17/02/2023;20;23
6 1047300390019;Negativa;0,08ML/ML XPE CT FR PLAS AMB X 100ML + COP;3.93;4.4
7 1039201650038;Positiva;200 MG COM CT BL AL PLAS INC X 20;4.11;4.11
8 1049701910036;Positiva;2 MG/ML SOL OR CT FR PLAS OPC GOT X 20 ML;4.35;4.35
9 1049712080013;Positiva;2 MG/ML SOL OR CT FR GOT PLAS PEBD OPC X 20 ML;4.35;4.35
10 1049711680017;Negativa;5 MG COM REV LIB RETARD CT BL AL PLAS TRANS X 20;4.35;4.87

```

Imagem 6

Busca de um elemento:

A Imagem 7 mostra a realização da busca de um elemento do arquivo.

```

Por qual campo deseja realizar a busca, digite [1] para o Campo 1 ou [4] para o Campo 4
4
Digite o ID (Campo 4) do elemento que deseja realizar a busca:

15.59

Elemento encontrado na posicao 14 do arquivo!
1039201500095;Positiva;5 MG COM CT BL AL PLAS AMB X 30;15.59;15.59

```

Imagem 7

Impressão de um trecho específico:

A Imagem 8 mostra a impressão da linha 10 até a linha 20, ordenadas pelo campo 1

```

Digite a posição inicial
10
Digite a posição final
20
1039201500060;Positiva;5 MG COM CT BL AL PLAS AMB X 500;83.81;83.81
1039201500079;Positiva;10 MG COM CT BL AL PLAS AMB X 20;30.62;30.62
1039201500087;Positiva;10 MG COM CT BL AL PLAS AMB X 500;142.51;142.51
1039201500095;Positiva;5 MG COM CT BL AL PLAS AMB X 30;15.59;15.59
1039201500109;Positiva;10 MG COM CT BL AL PLAS AMB X 30;19.71;19.71
1039201550033;Positiva;200 MG COM REV CT BL AL PLAS TRANS X 20;8.22;8.22
1039201550041;Positiva;200 MG COM REV CX BL AL PLAS TRANS X 200;101.94;101.94
1039201570069;Positiva;150 MG CAP DURA CT BL AL PLAS TRANS X 1;7.46;7.46
1039201570077;Positiva;150 MG CAP DURA CT BL AL PLAS TRANS X 500;847.94;847.94
1039201570085;Positiva;150 MG CAP DURA CT BL AL PLAS TRANS X 2;10.56;10.56

```