UNIVERSIDADE FEDERAL DE LAVRAS – GAC109

**PROJETO PRÁTICO:** Sistema de Cadastro em Arquivos com Ordenação

**NOMES:** Ana Clara Gomes – 202210591 – 22A Camila Vitória Lopes – 202210169 – 22A

Mateus Henrique Teixeira – 202210162 – 22A

# INTRODUÇÃO

O trabalho desenvolvido simula um sistema que, a partir de uma planilha de dados (.csv), consegue manipular as informações contidas nela, de modo a buscar, inserir, excluir, ordenar e exibir esses dados. Além disso, vale ressaltar que, mediante as informações contidas na base da dados, a qual será melhor discutida a seguir, simulou-se um sistema que armazena dados sobre medicamentos.

# BASE DE DADOS

A base de dados escolhida foi a de número 10, no entanto, após realizar todo o trabalho notou-se que o google drive de bases continha a “base1\_OK”, a “base1OK” e a “base10\_OK”. Dessa forma, utilizou-se durante todo o desenvolvimento do trabalho a “base1OK”, pensando que era referente a base 10. Todavia, após todo o trabalho estar pronto, notou-se a presença da base “base 10\_OK” e, ao abri-la, notou-se que, na verdade, fez-se o trabalho todo em cima da base 1 e não na base escolhida. Desse modo, considerou-se as seguintes colunas: descrição dos medicamentos, código de barras, preço na farmácia 1, preço na farmácia 2 e status (situação que descreve se os dados estão atualizados ou não), respectivamente.

É importante destacar que algumas modificações foram feitas na base de dados, tais como preenchimento de células vazias das colunas de preço com números aleatórios e a exclusão de 18 linhas aleatórias que, por algum motivo não compreendido, impediam o perfeito funcionamento da função de busca.

# ESTRUTURAS DO CÓDIGO

* **IMPORTAÇÃO**

Primeiramente foi realizada a leitura do arquivo .csv, na qual foi necessário adicionar condições para que fosse possível ler todas as informações contidas nas linhas presentes. Para isso, criou-se um procedimento que considerava o fato do ponto e vírgula “;” separar as colunas e a quebra de linha “\n” mudar a linha. Além disso, foi criada uma condição para que as aspas não atrapalhassem a leitura das células do arquivo .csv. Para a leitura do arquivo, foi usado o registro “dados”, de modo a armazenar as informações no tipo de variável desejada, tal como “descrição” no tipo vetor de caracteres e “preço 1” no tipo float, por exemplo. Os dados do registro foram sendo gravados em um arquivo binário nomeado “arqvsaida.bin”, o qual será utilizado para as funções abaixo. Vale ressaltar que no caso dos preços, os mesmo eram lidos

como *strings,* porém, por meio da função “stof()” e da substituição da vírgula pelo ponto (12,23 passou a ser 12.23, por exemplo), eram passados para o tipo float.

* **ORDENAÇÃO DE DADOS**

A partir do arquivo binário, utilizou-se do algoritmo Shell Sort para ordenar os dados pelas colunas contendo o preço do medicamento na farmácia 1 e na farmácia 2. Para isso foi criado um procedimento que permite o usuário escolher por qual preço deseja ordenar os dados. Além disso, para cada forma de ordenação, foi criado outro procedimento que abre o arquivo para leitura, verifica os dados deletados para excluí-los (utiliza um procedimento auxiliar que informa a quantidade de dados do arquivo), copia os dados não apagados para um vetor do tipo “dados” e o ordena com o algoritmo Shell Sort. Vale ressaltar que foi utilizada a sequência de Knuth para definição do gap de comparação. Por fim, após ordenado, o vetor é copiado novamente para o arquivo binário por meio de um procedimento auxiliar criado que apaga todos os dados do arquivo e escreve novamente, dessa vez ordenados.

* **IMPRESSÃO NA TELA**

Para impressão na tela, o procedimento criado para tal função utiliza o procedimento auxiliar que informa a quantidade de dados do arquivo para exibir ao usuário essa informação e, também quantos dados foram apagados do mesmo. Depois disso, é permitido ao usuário escolher se deseja imprimir todo o arquivo em tela, ou se quer imprimir de um determinado medicamento até outro. Vale ressaltar que apenas os dados não apagados são exibidos na tela.

* **EXCLUSÃO DE DADOS**

Na parte de excluir arquivos, o arquivo binário é aberto para leitura e escrita e é utilizado o procedimento para verificar a quantidade de dados do arquivo. Após isso, uma mensagem é exibida para que o usuário digite o código de barras do medicamento que deseja buscar as informações. A busca é feita de forma sequencial até que o medicamento seja encontrado e, por meio de um marcador booleano criado dentro do registro “dados”, o medicamento que se deseja excluir fica marcado como um dado apagado (o booleano fica igual a 1), mas não é completamente deletado do arquivo binário. Dessa forma, a única maneira de deletar totalmente informações é ordenando o arquivo.

É importante salientar que a comparação entre o código digitado e os códigos do arquivo .bin é feita pela função “strcmp()”, da biblioteca “string.h”, que verifica a igualdade entre vetores de caracteres.

* **BUSCA DE DADOS**

Para a busca de um medicamento específico, é permitido ao usuário buscar tanto pela descrição do medicamento, quanto pelo código de barras. Dessa forma, para cada um dos casos de busca, foi criado um procedimento que abre o arquivo binário para leitura, recebe a informação do tamanho do arquivo pelo procedimento auxiliar criado e verifica por todo o arquivo até encontrar o código de barras ou a descrição buscada (essa busca também utiliza o “strcmp()” para comparar os vetores de caracteres), desde que esses não estejam com o marcador booleano de apagado. Em caso de uma busca bem sucedida, o ponteiro de leitura do arquivo é posicionado na posição do registro do medicamento buscado e são informados todos

os dados do mesmo. Caso contrário, é exibida na tela uma mensagem de medicamento não encontrado.

* **INSERÇÃO DE DADOS**

Foi criado um procedimento responsável por inserir novos medicamentos na base de dados, desde que o código de barras não fosse igual aos dos já cadastrados no sistema. Para isso, o arquivo foi aberto no modo leitura, uma mensagem pede ao usuário que digite o código de barras do medicamento novo, o mesmo é verificado se não já existe na base de dados, e no caso de ser novo, uma série de mensagens pede as informações do medicamento a ser adicionado. Após isso, há uma condição que verifica se o arquivo está sendo ordenado pelo preço da farmácia 1 ou 2, e caso positivo, já ordena o dado inserido no arquivo. Destaca-se que os novos medicamentos são inseridos no final do arquivo por um procedimento auxiliar criado.

* **EXPORTAÇÃO**

Também é possível exportar o arquivo binário criado e modificado. Para isso, foi criado um procedimento que abre o arquivo binário para leitura e abre um arquivo .csv, cujo nome é inserido pelo usuário, limpando todos seus dados e copiando todos os dados do arquivo binário. Para isso, utilizou-se do fato do ponto e vírgula separar as colunas e a quebra de linha separar as linhas.

* **PROCEDIMENTO AUXILIARES**

Além dos procedimentos auxiliares citados, tais como a definição da quantidade de dados no arquivo, a escrita de novos dados no final do arquivo e da reescrita total do arquivo com os dados, foi criado um procedimento que limpava o terminal, de modo a mantê-lo mais organizado e dinâmico.

Também foi criado um procedimento que continha o menu com as funções supracitadas do sistema, tais como ordenação e inserção de medicamentos. Dessa forma, todas as opções eram exibidas para o usuário escolher qual ele gostaria de executar.

Por fim, cabe destacar que, mesmo não sendo um procedimento, no final de cada função da estrutura principal do código, uma mensagem perguntava ao usuário se ele desejava ir ao menu ou repetir a função recém executada. Tal funcionalidade não fora criada em um procedimento, pois para cada função uma mensagem nova era exibida, tal como: “deseja ir ao menu ou ordenar novamente” ou “deseja ir ao menu ou excluir outro medicamento”.

# CONCLUSÃO

A partir das funções acimas, conseguiu-se um sistema capaz de ler a base de dados utilizada (“base1OK”) e modificá-la de acordo com as funções disponibilizadas. Alguns pontos de melhoria para o sistema seriam: a capacidade de importar diferentes bases de dados, importar arquivos .csv com células sem informação alguma e impedir que o sistema trave ao digitar uma letra ou vírgula no procedimento de inserir o preço.

Ademais, o sistema apresenta uma acessibilidade e simplicidade ao usuário e não apresenta um limite na quantidade de dados a ser armazenados, o que é um ponto importante em sistemas de cadastro.