



USCS – Universidade Municipal de São Caetano do Sul

Curso: Ciência da computação

Projeto de Desenvolvimento de Sistema em C

Algoritmos e Estruturas de Dados II

Sistema de Cadastro de Cliente em C

Nome dos Integrantes do Grupo:

- Eduardo Antonio Braga Junior
- Tiago Felix da Silva
- Enzo Martins Guinzani
- Diogo Custodio Bastos
- Carlos Eduardo da Silva Lima

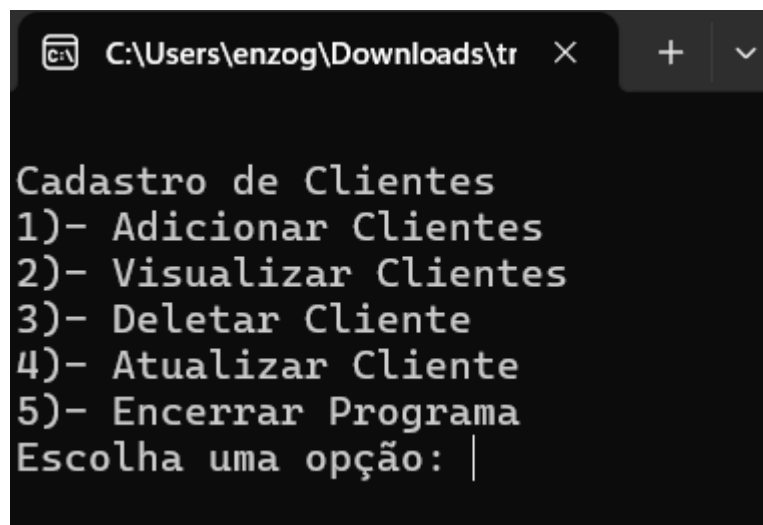
São Caetano do Sul

2024

INTRODUÇÃO

Este projeto tem como finalidade trazer um sistema feito na linguagem C onde possui o foco no tema “Cadastro de clientes”, dessa maneira, o projeto traz para o usuário a possibilidade de gerenciar os seus clientes dentro de um programa em C com um “arquivo.csv” (onde no qual poderá salvar as informações).

Com isso em mente, o programa permite ao seu usuário: cadastrar clientes, atualizar clientes, deletar clientes e visualizar quais os dados dos clientes no qual foram inseridos.

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Users\enzog\Downloads\tr' and standard window controls. The command prompt displays the following text:

```
Cadastro de Clientes  
1)- Adicionar Clientes  
2)- Visualizar Clientes  
3)- Deletar Cliente  
4)- Atualizar Cliente  
5)- Encerrar Programa  
Escolha uma opção: |
```

Imagem básica mostrando a tela inicial do programa

¹ Algoritmos e Estruturas de Dados II do curso de Ciência da computação, Universidade Municipal de São Caetano do Sul, 2024

Modelagem do Banco de Dados

Quando tratamos da parte de modelagem de dados, o programa faz a criação da tabela de cadastro de clientes com base na struct definida dentro do código fonte com os dados “Código, Nome, Telefone, Endereço, Estado, CPF e RG” onde todas as informações digitadas serão inseridas.

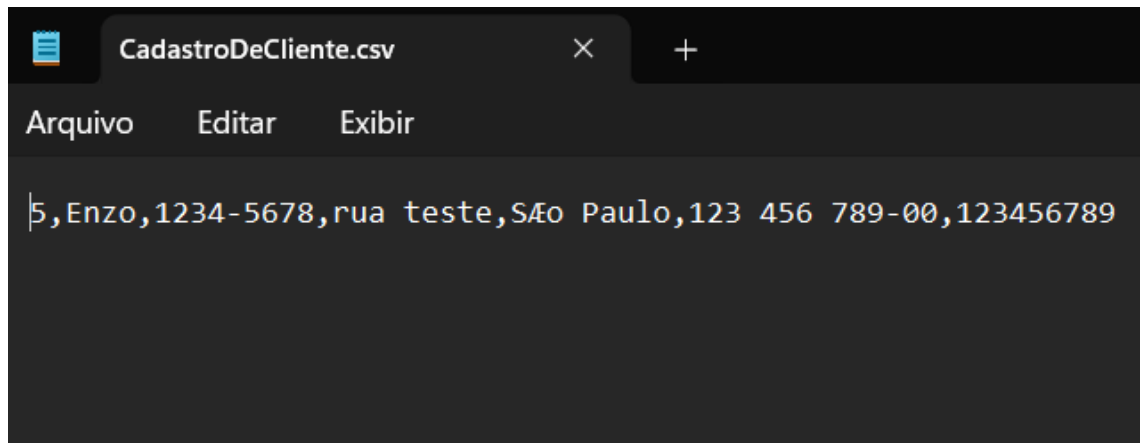
```
Escolha uma opção: 2

Lista de clientes cadastrados:
Código | Nome | Telefone | Endereço | Estado | CPF | RG
5 | Enzo | 1234-5678 | rua teste | São Paulo | 123 456 789-00 | 123456789

Cadastro de Clientes
1)- Adicionar Clientes
2)- Visualizar Clientes
3)- Deletar Cliente
4)- Atualizar Cliente
5)- Encerrar Programa
Escolha uma opção: |
```

Na imagem acima é possível analisar que ao escolher a opção de visualizar os clientes, o programa faz a geração da tabela com os dados de clientes cadastrados, mostrando assim desta maneira, as colunas com os tipos de dados (código, nome, telefone....) e logo após na próxima fileira estão os dados cadastrados de clientes.

Um outro detalhe importante é que o programa também irá salvar e transferir esses mesmos dados em um arquivo de texto (CSV) na máquina do usuário, simulando um Banco de Dados para ele, além disso, as outras funcionalidades, como Deletar e Atualizar clientes também estarão sendo aplicadas dentro do arquivo csv do usuário.

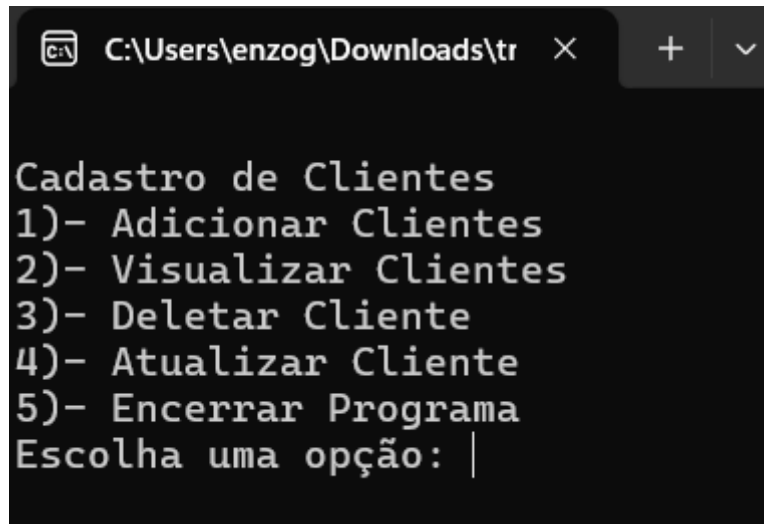


Demonstração do arquivo CadastroDeCliente.csv

Na imagem está o arquivo.csv que recebeu toda a informação digitada dentro do programa em C, observa se que os dados estão sendo separados por vírgula, isso é para destacar cada coluna com a respectivas informações cadastradas.

Desenvolvimento

Após mostrar os fundamentos básicos do código, agora será mostrado todo o funcionamento do programa e todas as suas funcionalidades.



```
C:\Users\enzog\Downloads\tr >
Cadastro de Clientes
1)- Adicionar Clientes
2)- Visualizar Clientes
3)- Deletar Cliente
4)- Atualizar Cliente
5)- Encerrar Programa
Escolha uma opção: |
```

Ao inicializar o programa, o usuário haverá acesso a opções iniciais:

- 1- Adicionar Clientes
- 2- Visualizar Clientes
- 3- Deletar Clientes
- 4- Atualizar Clientes
- 5- Encerrar Programa

Para acessar algumas dessas opções, basta o usuário apenas digitar em seu teclado o número da opção no qual ele quer escolher.

Caso o usuário digitar um número inválido, como um número menor que 1, ou maior que 5, ou o usuário digitar até mesmo uma letra, o programa irá mostrar uma mensagem de erro “Erro, entrada inválida. Tente novamente.” e permitirá a ele digitar até que a entrada seja válida (ou seja, um número de 1 a 5).

```

int main() {
    setlocale(LC_ALL, "portuguese");
    int opcao;
    do {
        printf("\nCadastro de Clientes\n");
        printf("1)- Adicionar Clientes\n");
        printf("2)- Visualizar Clientes\n");
        printf("3)- Deletar Cliente\n");
        printf("4)- Atualizar Cliente\n");
        printf("5)- Encerrar Programa\n");
        printf("Escolha uma opção: ");

        if (scanf("%d", &opcao) != 1) {
            printf("Erro, entrada inválida. Tente novamente.\n");
            while (getchar() != '\n');
            continue;
        }
    } while (opcao != 5);
}

```

Para realizar essa funcionalidade do programa, o código acima foi implementado, utilizando um loop IF para fazer o usuário digitar um número da lista e poder assim, continuar usando o programa.

•Adicionar Clientes

Por meio da opção “adicionar clientes”, é possível inserir dados e informações de um cliente e transferir a tabela do cadastramento e ao arquivo.csv.

```

Escolha uma opção: 1

Código do cliente: 3
Nome: Pessoa Teste
Telefone: 1111-1111
Endereço: endereço teste
Estado: teste estado
CPF: 111 111 111-11
RG: 1234567809
Cliente cadastrado com sucesso!

```

Após digitar 1 e selecionar a opção, o sistema vai abrir campos para o usuário digitar os dados, e com isso, ao preencher todos os campos necessários, o programa entregará uma mensagem de sucesso “Cliente cadastrado com sucesso!” e enviará as informações digitadas para o CadastroDeCliente.csv e irá deixar salvo em sua tabela de clientes cadastrados.

```

printf("Nome: ");
scanf(" %[^\\n]", cliente.Nome);
printf("Telefone: ");
scanf(" %[^\\n]", cliente.Telefone);
printf("Endereço: ");
scanf(" %[^\\n]", cliente.endereco);
printf("Estado: ");
scanf(" %[^\\n]", cliente.estado);
printf("CPF: ");
scanf(" %[^\\n]", cliente.CPF);
printf("RG: ");
scanf(" %[^\\n]", cliente.RG);

fprintf(arquivo, "%d,%s,%s,%s,%s,%s,%s\\n", c
fclose(arquivo);

printf("Cliente cadastrado com sucesso!\\n");

```

Nesta imagem está a parte principal de cadastramento dos clientes, onde estão os códigos de printf e scanf que vão receber a entrada do usuário, sendo apenas o nome como uma entrada do tipo INT (número inteiro), enquanto todas as outras entradas são de variáveis do tipo CHAR (caractere).

Se o usuário digitar por exemplo, uma letra no campo do código onde é necessário incluir apenas números, o programa retornará "0" (erro) e enviará uma mensagem de erro e após isso ele encerra o cadastro do cliente para evitar problemas na inserção de dados.

```

Cliente cliente;
printf("\\nCódigo do cliente: ");
if (scanf("%d", &cliente.codCliente) != 1) {
    printf("Erro ao inserir o código do cliente.\\n");
    fclose(arquivo);
    while (getchar() != '\\n');
    return;
}

```

Ilustração do código da entrada do "Código do cliente"

•Visualizar Clientes

Na segunda opção, a “visualização de clientes”, como previamente já demonstrado, o usuário pode ver quais os clientes já foram cadastrados, nessa opção o código pega toda a informação já inserida e salva no documento csv e mostra para o próprio usuário.

```

Lista de clientes cadastrados:
Código | Nome | Telefone | Endereço | Estado | CPF | RG
5 | Enzo | 1234-5678 | rua teste | São Paulo | 123 456 789-00 | 123456789
3 | Pessoa Teste | 1111-1111 | endereço teste | teste estado | 111 111 111-11 | 1234567809

Cadastro de Clientes
1)- Adicionar Clientes
2)- Visualizar Clientes
3)- Deletar Cliente
4)- Atualizar Cliente
5)- Encerrar Programa
Escolha uma opção: |

```

Se caso não houver nenhum cliente cadastrado, o programa mostra uma mensagem de “Nenhum cliente cadastrado.” ao usuário e continuará com a possibilidade de ele usar alguma as opções da lista.

```

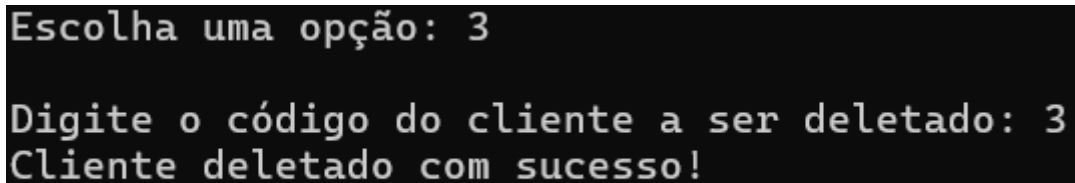
Cliente cliente;
int hasData = 0;
printf("\nLista de clientes cadastrados:\n");
printf("Código | Nome | Telefone | Endereço | Estado | CPF | RG\n");
while (fscanf(arquivo, "%d,%29[^,],%14[^,],%49[^,],%49[^,],%19[^,],%19[^,],%19[^,],%19[^,],%19[^,]\n",
    &cliente.codCliente, &cliente.nome, &cliente.telefone, &cliente.endereco, &cliente.estado, &cliente.cpf, &cliente.rg) == 1)
{
    printf("%d | %s | %s | %s | %s | %s | %s\n", cliente.codCliente, cliente.nome, cliente.telefone, cliente.endereco, cliente.estado, cliente.cpf, cliente.rg);
    hasData = 1;
}
if (!hasData) {
    printf("Nenhum cliente cadastrado.\n");
}
fclose(arquivo);

```

Código principal para a ilustração da tabela de clientes cadastrados

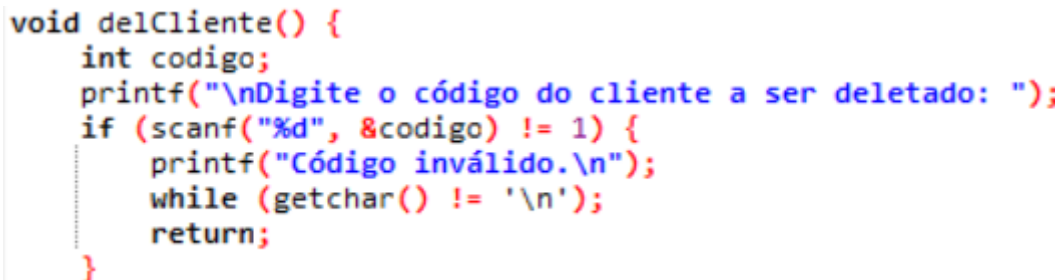
•Deletar Clientes

Assim como é possível adicionar clientes, também é possível deletar clientes, para utilizar essa terceira opção, só é necessário incluir o código do cliente no qual vai ser retirado da tabela, é importante mencionar que o cliente deletado também será removido em completo do arquivo.csv.



```
Escolha uma opção: 3
Digite o código do cliente a ser deletado: 3
Cliente deletado com sucesso!
```

Imagem do programa usando a opção de deletar



```
void delCliente() {
    int codigo;
    printf("\nDigite o código do cliente a ser deletado: ");
    if (scanf("%d", &codigo) != 1) {
        printf("Código inválido.\n");
        while (getchar() != '\n');
        return;
    }
}
```

Imagem do código que exerce a função de deletar

Se a entrada do usuário for um código de um cliente que não existe ou uma letra, será enviado uma mensagem de erro a ele e será interrompido a opção de deletar.

•Atualizar Clientes

O sistema de cadastramento de clientes também permite atualizar os dados do cliente, essa opção funciona de forma semelhante ao cadastramento, onde ao escolher ela o programa mostrará os campos onde o usuário vai digitar as informações, porém antes disso é necessário digitar o código do cliente onde vai sobrescrever os dados e atualizar o próprio.

Adicionalmente, ao decorrer da atualização de dados, o próprio programa faz um arquivo temp (temporário), onde ele passa as informações para ele a fim de evitar com que o usuário perca os dados caso o programa seja encerrado no meio do procedimento de atualizar o cliente.

```
Escolha uma opção: 4

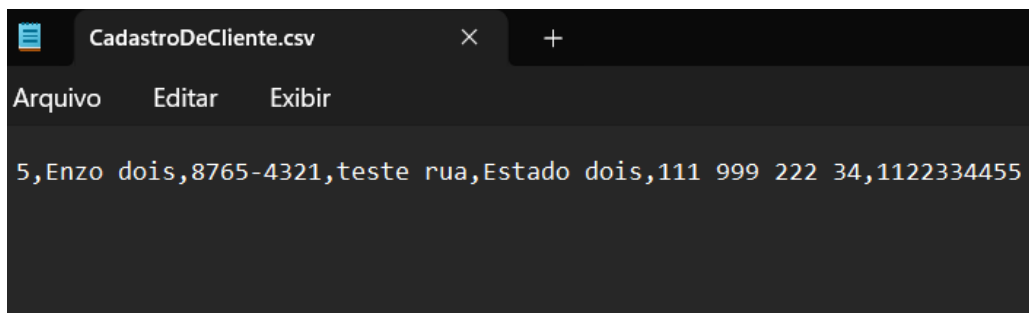
Digite o código do cliente a ser atualizado: 5
Novo nome: Enzo dois
Novo telefone: 8765-4321
Novo Endereço: teste rua
Novo Estado: Estado dois
Novo CPF: 111 999 222 34
Novo RG: 1122334455
Cliente atualizado com sucesso!
```

Imagem do sistema atualizando o cliente

```
Cadastro de Clientes
1)- Adicionar Clientes
2)- Visualizar Clientes
3)- Deletar Cliente
4)- Atualizar Cliente
5)- Encerrar Programa
Escolha uma opção: 2

Lista de clientes cadastrados:
Código | Nome | Telefone | Endereço | Estado | CPF | RG
5 | Enzo dois | 8765-4321 | teste rua | Estado dois | 111 999 222 34 | 1122334455
```

Conferindo os dados atualizados dentro do sistema



Conferindo os dados atualizados no documento csv

```
if (cliente.codCliente == codigo) {
    found = 1;
    printf("Novo nome: ");
    scanf(" %[^\\n]", cliente.Nome);
    printf("Novo telefone: ");
    scanf(" %[^\\n]", cliente.Telefone);
    printf("Novo Endereço: ");
    scanf(" %[^\\n]", cliente.endereco);
    printf("Novo Estado: ");
    scanf(" %[^\\n]", cliente.estado);
    printf("Novo CPF: ");
    scanf(" %[^\\n]", cliente.CPF);
    printf("Novo RG: ");
    scanf(" %[^\\n]", cliente.RG);
}
```

Código principal para atualizar o cliente

```
void updateCliente() {
    int codigo;
    printf("\nDigite o código do cliente a ser atualizado: ");
    if (scanf("%d", &codigo) != 1) {
        printf("Código inválido.\n");
        while (getchar() != '\\n');
        return;
    }
}
```

Código para impedir a inserção de uma entrada inválida

Assim como a opção de deletar clientes, a função de atualizar também possui um código para casos em que o usuário digite uma entrada inválida.

•Encerrar Programa

Por fim, a última funcionalidade desse sistema é o encerramento do programa, uma opção simples que permite apenas fechar o programa ao digitar “5” na tela de opções.

```
Cadastro de Clientes
1)- Adicionar Clientes
2)- Visualizar Clientes
3)- Deletar Cliente
4)- Atualizar Cliente
5)- Encerrar Programa
Escolha uma opção: 5
Programa fechado!

-----
Process exited after 1312 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Conclusão

O projeto teve como objetivo fazer um sistema CRUD (create, read, update, delete) para um cadastramento de clientes, e foi possível realizar cada uma dessas opções dentro dele.

No desenvolvimento desse sistema também houve algumas certas dificuldades, como a inserção de códigos que impeçam que erros ocorram durante o uso do próprio sistema, sendo por exemplo, problemas que aconteçam se o usuário digitar uma entrada inválida ou problemas em relação a perda de dados que foram possam ter sido cadastrados ou editados no sistema.

Ao fim disso, foi possível aprender como funciona um banco de dados em formato da linguagem C, como um código com uso de arquivos mais avançado funciona e interage com documentos externos fora do programa e por fim também permitiu aprender como o uso da struct pode ser mais valorizado em Sistemas mais complexos feitos em C como esse.