

CENTRO FEREDEAL DE DUCAÇÃO TECNOLÓGICA DE MINAS GERAIS –
CEFET/MG
CAMPUS III – LEOPOLDINA

CMA02 - LABORATÓRIO DE PROGRAMAÇÃO I
SISTEMA DE CADASTRO DE CLIENTES

12 de julho de 2022

Nome: Ayres William Raimundo Monteiro, Laura Cruz da Silva, Thiago César Pereira

E-mail: ayresmonteiro52@gmail.com, lauracruzsilva@gmail.com, thiagorasec_15@hotmail.com,

SUMÁRIO

1. INTRODUÇÃO	3
2. CAMPOS A SEREM CADASTRADOS	3
3. GUIA DE UTILIZAÇÃO	4
4. CÓDIGO FONTE	7

SISTEMA DE CADASTRO DE CLIENTES

1. INTRODUÇÃO

Foi desenvolvido, em linguagem C, um sistema de cadastro de clientes, que pode ser usado tanto por um empresário autônomo quanto por uma empresa que deseja obter controle sobre os dados dos seus clientes. O sistema é responsável por permitir ao usuário uma gestão dos dados dos clientes, como: cadastrar cliente, visualizar clientes cadastrados, alterar dados e excluir clientes. Como a gestão de clientes é uma atividade corriqueira, é possível encontrar sistemas de cadastro de clientes a venda no mercado, inclusive no mercado livre. Porém, as funcionalidades e os dados em cada sistema irão variar conforme o desenvolvedor.

- (a) **Implementação:** Para a implementação do programa foi usada uma struct denominada "CLIENTES" com variáveis para armazenar os dados cadastrados. Também foram criadas funções específicas para a gestão dos dados (cadastrarClientes, excluirClientes, atualizarClientes) cujo funcionamento depende da decisão do usuário, de acordo com a opção digitada no menu que é controlada pela função "decisao()".
- (b) **Testes:** Foram feitos alguns testes para prevenir falhas do usuário, no que diz respeito à lógica de programação. Por exemplo, caso o usuário queira excluir um cliente cujo o id(identificador) não exista, o programa filtra esta entrada e explica ao usuário que não existe cliente com aquele id. Ou ainda, caso o usuário digite uma opção diversa da mostrada no menu o programa mostra uma mensagem dizendo a ele que esta é uma opção inválida.

2. CAMPOS A SEREM CADASTRADOS

a) Campos a serem cadastrados no sistema:

Struct CLIENTES

Nome do campo	Tipo	Descrição
id	Inteiro	Campo que identifica o cliente (único, não se repete, é inalterável)
ativo	Inteiro	0 – Cliente inativo

		1 – Cliente ativo
nome	Cadeia de caracteres	Armazena o nome do cliente
cpf	Cadeira de caracteres	Armazena o CPF do cliente (aceita barra, ponto e hífen)
telefone	Cadeia de caracteres	Armazena o telefone do cliente (aceita, parênteses, hífen, sinal)
sexo	Cadeia de caracteres	Armazena o sexo do cliente
dataNascimento	Cadeia de caracteres	Armazena a data de nascimento do cliente (aceita barra)

3. GUIA DE UTILIZAÇÃO

A tela inicial do sistema mostra o nome do mesmo, a versão e o ano (SisCAD 1.0 2022) e uma mensagem de boas-vindas ao usuário. Conforme se observa na imagem abaixo:

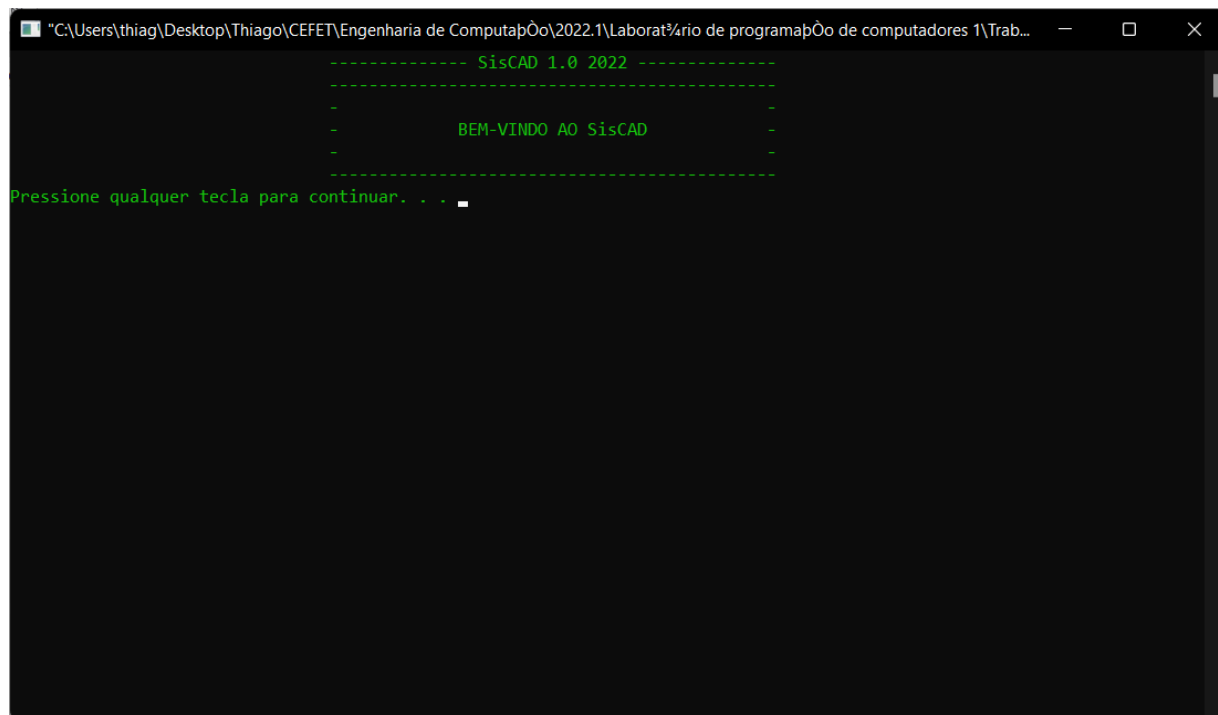
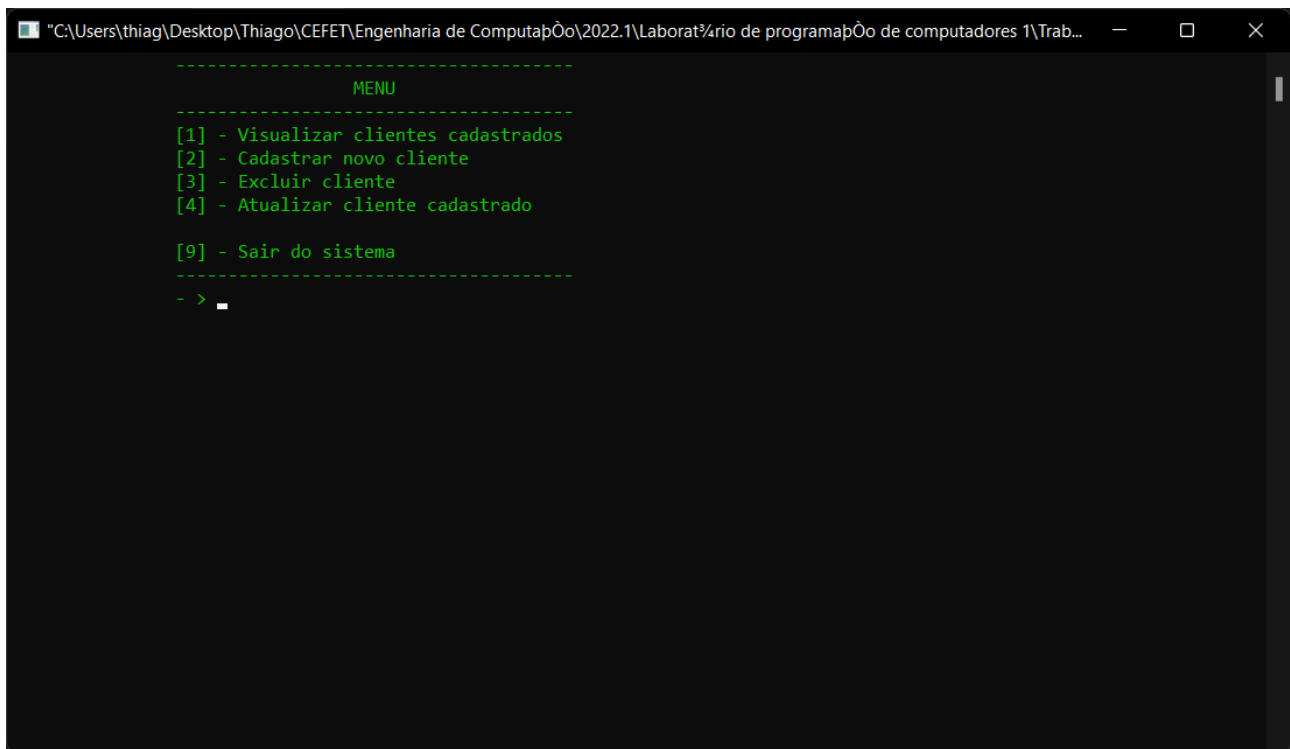


Figura 1 - Tela inicial SisCAD

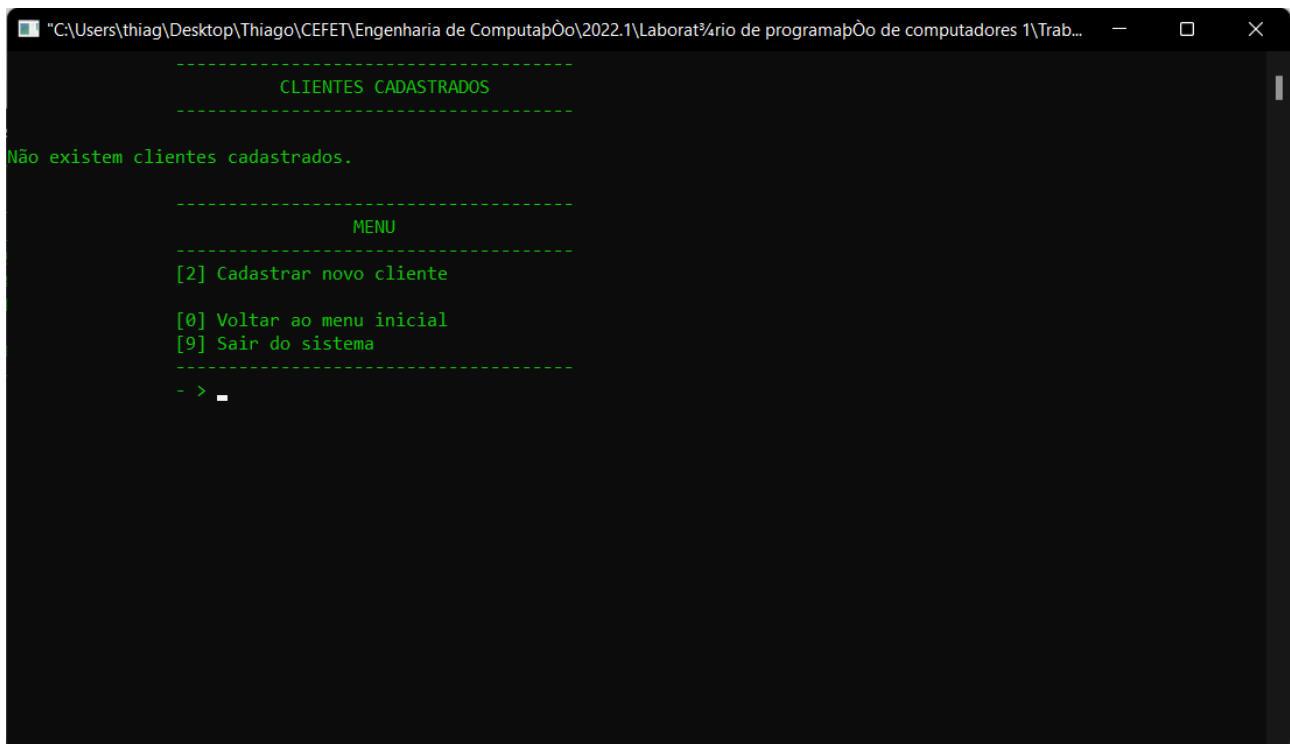
Após clicar em qualquer tecla o usuário é direcionado ao menu inicial do sistema, conforme a figura abaixo:



```
"C:\Users\thiag\Desktop\Thiago\CEFET\Engenharia de Computação\2022.1\Laboratório de programação de computadores 1\Trab...  
-----  
MENU  
-----  
[1] - Visualizar clientes cadastrados  
[2] - Cadastrar novo cliente  
[3] - Excluir cliente  
[4] - Atualizar cliente cadastrado  
  
[9] - Sair do sistema  
-----  
- > _
```

Figura 2 - O menu inicial

São apresentadas as opções que o usuário poderá seguir pelo sistema. Caso escolha a opção 1, esta é a tela que irá aparecer para ele:



```
"C:\Users\thiag\Desktop\Thiago\CEFET\Engenharia de Computação\2022.1\Laboratório de programação de computadores 1\Trab...  
-----  
CLIENTES CADASTRADOS  
-----  
Não existem clientes cadastrados.  
  
-----  
MENU  
-----  
[2] Cadastrar novo cliente  
  
[0] Voltar ao menu inicial  
[9] Sair do sistema  
-----  
- > _
```

Figura 3 - Tela clientes cadastrados

Caso existam clientes cadastrados o sistema mostra os dados destes. Caso contrário, como acima, exibe a mensagem: “Não existem clientes cadastrados”. Por outro lado, se o usuário escolher a opção 2 no menu inicial será direcionado para o cadastro de clientes, cuja tela é:

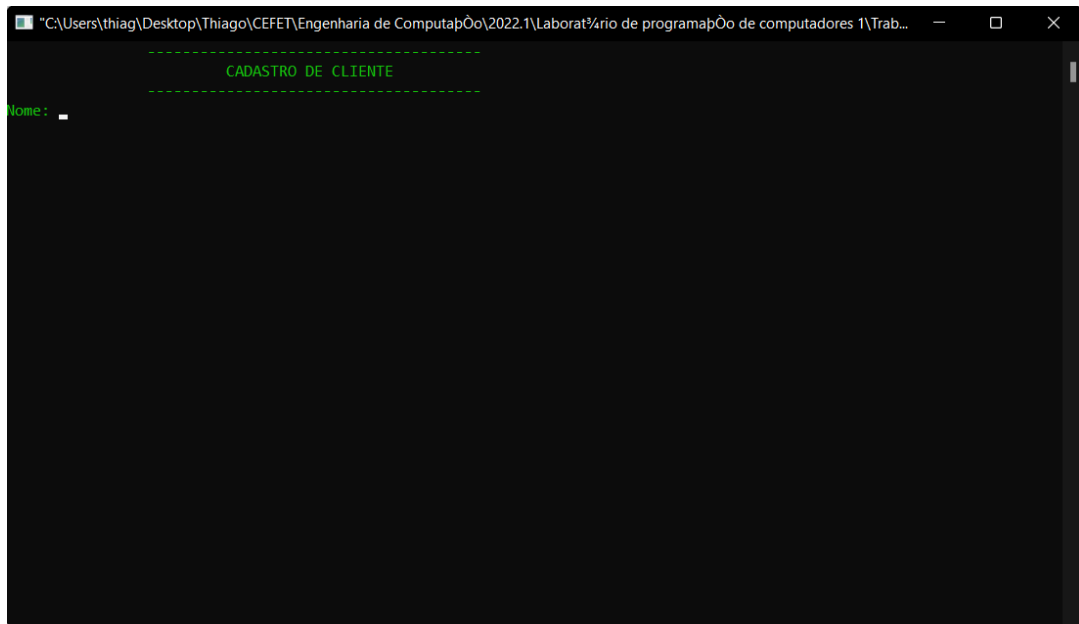


Figura 4 - Cadastrar novo cliente

As outras opções seguem a mesma lógica. Caso digite a opção 3 será direcionado para a tela de exclusão de clientes, na qual deverá digitar o id do usuário que gostaria de excluir. Por fim, temos a opção 9 (Sair do sistema), cuja tela é a mostra abaixo:

```
"C:\Users\thiag\Desktop\Thiago\CEFET\Engenharia de ComputaçÕo\2022.1\Laborat³rio de programaçÕo de computadores 1\Trab...
-      OBRIGADO      -
-      VOLTE SEMPRE  -
-      -      -
-----
Process returned 0 (0x0)   execution time : 256.660 s
Press any key to continue.
```

Figura 5 - Tela de saída do sistema

Caso deseja ver o sistema em funcionamento e como é a navegação dentro de cada opção mostrada no menu inicial pode-se acessar o tutorial do sistema através do link: <https://youtu.be/AoE3b0BXfQU>

4. CÓDIGO FONTE

a) Campos a serem cadastrados no sistema:

Struct CLIENTES

Nome do campo	Tipo	Descrição
id	Inteiro	Campo que identifica o cliente (único, não se repete, é inalterável)
ativo	Inteiro	0 – Cliente inativo 1 – Cliente ativo
nome	Cadeia de caracteres	Armazena o nome do cliente

cpf	Cadeira de caracteres	Armazena o CPF do cliente (aceita barra, ponto e hífen)
telefone	Cadeia de caracteres	Armazena o telefone do cliente (aceita, parênteses, hífen, sinal)
sexo	Cadeia de caracteres	Armazena o sexo do cliente
dataNascimento	Cadeia de caracteres	Armazena a data de nascimento do cliente (aceita barra)

b) Variáveis gerais importantes:

MAX_CLI	Constante	Define o tamanho do vetor de struct. Ou seja, quantos clientes posso cadastrar. Fica a cargo do gestor aumentar ou diminuir
i	Inteiro	Usada dentro do for nas minhas funções para percorrer o vetor de struct
opcao	Inteiro	Usada para armazenar a opção que o usuário quer acessar de acordo com o menu
contador	inteiro	Usada dentro da função listarClientes para verificar se já existe cliente cadastrado ou não
cliente	Vetor de struct	Variável responsável por criar o vetor no qual irei armazenar os dados dos clientes

c) Procedimentos e funções:

tela_inicial ();

Serve para alterar a cor padrão do texto para verde, mantendo o fundo preto. Além disso, mostra uma mensagem inicial de boas vindas ao usuário. Além de apresentar o nome e a versão do sistema, bem como o ano de desenvolvimento. Não possui parâmetros.

menu();

Serve para mostrar ao usuário as opções de navegação e permite que ele escolha para onde quer ir no programa. Tem como parâmetros o vetor de struct e um inteiro com o tamanho máximo do vetor. Ao final, chama a função `decisao()`, passando a opção que o usuário digitou de acordo com o menu.

obrigado();

Assim como a `tela_inicial()` é uma função que mostra apenas texto. No caso desta, mostra uma mensagem de agradecimento ao usuário por usar o sistema. Não recebe parâmetros.

inicializaClientes();

Recebe como parâmetros o vetor de struct com os clientes e o tamanho máximo deste vetor. Sua finalidade é inicializar os id's do vetor com um inteiro fixo (máximo de clientes + 1) e inicializa a variável ativo como 0 (inativo). Desta forma facilita o filtro em outras funções. Por exemplo, se tem uma posição no vetor de struct cujo id é "Máximo de clientes + 1" é porque não tem cliente cadastrado nesta posição. Visto que, após o cadastro o id máximo que um cliente terá é igual ao tamanho máximo do vetor.

decisao();

Recebe como parâmetros o vetor de struct com os clientes, o tamanho máximo deste vetor e um inteiro que é a opção que o usuário digitou para navegar no menu. Essa função direciona o usuário dentro do programa. Dependendo da opção digitada, essa função chama as outras funções, como a `cadastroCliente()`, por exemplo. Esta função também filtra se a opção digitada pelo usuário é válida dentro das disponíveis.

cadastroCliente();

Recebe como parâmetros o vetor de struct com os clientes e o tamanho máximo deste vetor. Como o próprio nome já diz, é responsável por receber do usuário os dados necessários para cadastro de um cliente. Esta função faz um filtro inicial, através de um comando `while` para verificar se já existem clientes cadastrados. Esta verificação é importante, pois caso já existam clientes cadastrados, o novo cadastro deverá começar da primeira posição disponível e não da posição inicial. Ao final, chama a função `decisao()` passando a opção digitada pelo usuário de acordo com o menu mostrado.

listarClientes();

Recebe como parâmetros o vetor de struct com os clientes e o tamanho máximo deste vetor. Esta função é responsável por mostrar, caso existam, clientes cadastrados no sistema. Se não existir imprime na tela a mensagem: "Não existem usuários cadastrados". Ao final, chama a função `decisao()` passando a opção digitada pelo usuário de acordo com o menu mostrado.

excluirCliente()

Recebe como parâmetros o vetor de struct com os clientes e o tamanho máximo deste vetor. É responsável pela exclusão de um registro. Pede ao usuário o id do cliente que ele deseja excluir. Se

for um id inválido a função avisa que não existe aquele id. Caso seja válido, faz a exclusão do registro. É uma exclusão lógica, não física. Ou seja, ao excluir um usuário, a variável ativo dele passa a ser 0. Ao final, chama a função decisao() passando a opção digitada pelo usuário de acordo com o menu mostrado.

atualiarCliente()

Recebe como parâmetros o vetor de struct com os clientes e o tamanho máximo deste vetor. Pede ao usuário o id do cliente que ele deseja cadastrar. Assim como na função excluirCliente() mostra uma mensagem caso a id seja inválida. Caso contrário, mostra os dados do cliente cadastrado com aquele id para facilitar a visualização do usuário de qual dado deseja atualizar. O usuário então escolhe o dado que deseja atualizar e o sistema pede a nova entrada de dados. O dado então é atualizado. Ao final, chama a função decisao() passando a opção digitada pelo usuário de acordo com o menu mostrado.