**FATEC SÃO CAETANO DO SUL – ANTONIO RUSSO**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS – AMS**

**São Caetano do Sul**

**2024**

**FATEC SÃO CAETANO DO SUL – ANTONIO RUSSO**

**Bianca de Freitas Soares**

**Bruno Leonardo Martinho**

**Giovanni de Pita Cicero**

**Jorge Antonio Terence Novaes de Santana**

**Ruan Defavari Godoi**

**Sarah Alves De Melo**

**SISTEMA DE GERENCIAENTO DE CONTATOS**

**São Caetano do Sul**

**2024**

**RESUMO**

O objetivo do Projeto de Gerenciamento de Contatos é criar uma aplicação fácil de usar e fácil de entender para organizar, armazenar e gerenciar informações de contatos pessoais e profissionais. A ferramenta foi desenvolvida para ajudar os usuários a manterem suas listas de contatos atualizadas, o que facilita a comunicação e permite o acesso rápido a informações importantes. Para facilitar o gerenciamento de contatos, o sistema organiza a lista em ordem alfabética automaticamente ao adicionar um novo contato e permite buscar pessoas pelo seu nome.

**Palavras-chave**: Organização. Listagem. Contatos. Agenda.

**LISTA DE FIGURAS**

[Figura 1: Código struct 4](#_Toc168652355)

[Figura 2: exibir lista de contatos 4](#_Toc168652356)

[Figura 3: fila print\_queue 5](#_Toc168652357)

[Figura 4: pilha para consulta 6](#_Toc168652358)

[Figura 5: lista ordenada para armazenamento 7](#_Toc168652359)

[Figura 6: plano de testes 9](#_Toc168652360)

[Figura 7: teste de cadastro 10](#_Toc168652361)

[Figura 8: teste de exibição 10](#_Toc168652362)

[Figura 9: teste de consulta por nome 11](#_Toc168652363)

[Figura 10: teste de edição 11](#_Toc168652364)

[Figura 11: teste de exclusão 12](#_Toc168652365)

[Figura 12: teste de letras no telefone 12](#_Toc168652366)

[Figura 13: teste de consulta pelo mesmo nome 13](#_Toc168652367)

**LISTA DE TABELAS**

[Tabela 1: GitHub dos desenvolvedores 8](#_Toc168652369)

**SUMÁRIO**

[1. INTRODUÇÃO 1](#_Toc168561329)

[2. CONTEXTO DO PROBLEMA 2](#_Toc168561330)

[3. REVISÃO BIBLIOGRÁFICA 3](#_Toc168561331)

[4. IMPLEMENTAÇÃO 4](#_Toc168561332)

[5. TESTES DE SOFTWARE 6](#_Toc168561333)

[6. CONCLUSÃO 8](#_Toc168561334)

# 

# INTRODUÇÃO

A agenda de contatos existe há muito tempo, seja por papel ou por qualquer outro meio digital. A agenda eletrônica foi inventada em 1975, pelo indiano Sam Pitroda, que, na época, ele nomeou de “diário eletrônico”. Mais invenções como essas foram surgindo com o passar do tempo para facilitar nossos meios de comunicação, armazenamento de dados de contatos, telefones, e-mails e outros, de uma forma eficiente, rápida e até intuitiva.

O sistema de gerenciamento de contatos atua como uma agenda de contatos, onde cada contato adicionado é organizado alfabeticamente e cada contato pode ser editado para que a agenda esteja sempre atualizada. Juntamente, o sistema possibilita uma pesquisa rápida através dos nomes dos contatos, facilitando a busca de contatos existentes.

Hoje em dia, muitas pessoas usam agendas de contatos com inúmeras folhas ou até usam o próprio aplicativo “contatos” do seu dispositivo mobile. Porém, ambas alternativas acabam sendo trabalhosas. Enquanto uma tropeça por ser um processo mais lento, a outra acaba sendo trabalhosa na hora de editar um contato já existente.

Com isso em mente, o sistema criado oferece busca, adição, exclusão e alteração de contatos de uma forma rápida e eficiente. Focamos em deixar nosso sistema o mais detalhado para que não haja erros na lista de contatos de quem utiliza.

Para isso, fizemos inúmeros testes para garantir que o sistema atue com excelência e de uma forma que seja intuitiva e agradável de usar. Por exemplo, os números do contato não podem ter letras assim como é obrigatório todos os campos (nome, sobrenome, telefone e e-mail) estejam preenchidos para que seja adicionado à lista de contatos.

Quando o programa é inicializado, o usuário pode escolher entre algumas opções como: cadastrar, exibir, consultar, editar e remover os contatos. Essas opções são representadas por números de 0 a 5, sendo a opção 0 sair do programa. Lembrando que, caso a escolha de opção seja 0 e o usuário queira sair do sistema, o mesmo irá resetar, pois não se tem um banco de dados para o armazém das informações dos contatos.

# CONTEXTO DO PROBLEMA

Muitas pessoas, principalmente idosos ou pessoas não habituadas à tecnologia, ainda têm dificuldade na hora de incluir ou excluir um contato nos dispositivos mobile e acabam por preferir anotar em agendas, papéis ou até guardar cartões de contato. Porém essa opção pode ser um processo demorado na hora de procurar ou até acrescentar um novo contato. E pode ser ainda mais trabalhoso editar os contatos antigos.

Com essa preocupação, criamos um sistema que atua como um aplicativo de contatos que normalmente usamos em nossos dispositivos mobile, porém com um sistema semelhante a de uma agenda de contatos físico.

Assim, podemos facilitar o uso da agenda para essas pessoas, pois a ideia integra tanto o método antigo do papel quanto a tecnologia de usar um sistema ao seu favor para facilitar ações como busca e ordenação alfabética.

# REVISÃO BIBLIOGRÁFICA

Para a realização do sistema, utilizamos conhecimentos abordados anteriormente nas aulas de Estrutura de Dados com o professor Carlos Henrique Veríssimo.

Durante as aulas do primeiro semestre, passamos por inúmeros assuntos destinados à estrutura de dados, tais como: abstração de dados, ponteiros, filas, pilhas, listas ordenadas, struct e outros. Esses conhecimentos foram necessários para a construção do gerenciamento de contatos.

Por exemplo, usamos ponteiros para fazer nó nos contatos; fila para exibir os contatos; pilha para consultar os contatos; lista ordenada para o armazenamento dos contatos; struct para o modelo de um contato. Todos esses tópicos serão adentrados com mais clareza no próximo capítulo.

# IMPLEMENTAÇÃO

Vamos adentrar mais nos assuntos abordados e aplicados no código.

Começando pelo struct, utilizamos ele para criar um modelo de um contato, ou seja, o struct serve para que, quando um usuário adiciona um contato à sua agenda, o contato segue um modelo pré-definido que seria: nome, sobrenome, telefone e e-mail. Confira o código abaixo:

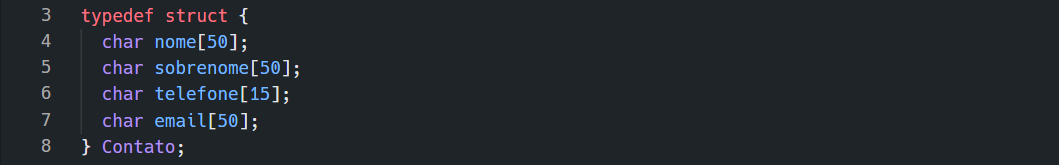


Figura 1: Código struct

Na imagem começamos definindo uma estrutura chamada “Contato” que tem atributos como os citados anteriormente. Definimos todos como char, ou seja, todos eles podem receber caracteres. Nome, sobrenome e e-mail recebem até 50 caracteres como seu máximo, enquanto telefone recebe até 15 caracteres.

A screen shot of a computer program

Description automatically generated

Figura 2: exibir lista de contatos

Usamos fila para exibir os contatos quando solicitado. Podemos ver no código abaixo um exemplo:

No código é criado uma função chamada *ll\_show* que recebe como parâmetro a origem da lista ordenada que armazena os contatos. Esta então é convertida para uma fila, por identificar a origem e o fim da lista. O fim é o elemento cujo atributo *proximo* é nulo; criamos um loop for que guarda uma variável e para assim encontre um valor nulo. A execução é passada então para outra função.

A computer screen shot of code

Description automatically generated

Figura 3: fila print\_queue

Recebendo como argumento uma fila, *print\_queue* primeiro garante que a fila não esteja vazia e então tira da fila todos os itens, mostrando-os um-a-um.

A screenshot of a computer program

Description automatically generated

Figura 4: pilha para consulta

Também utilizamos a pilha para a consulta dos contatos. Então, quando um usuário precisa de alguma função, o sistema irá utilizar da pilha de contatos para realizar aquela função específica.

Como visto no código acima, a função pergunta um nome ao usuário e instancia uma pilha com um limite definido por uma constante global. A lista ordenada é iterada, armazenando contatos com nomes que igualam à busca do usuário.

Caso a pilha ainda esteja vazia, o programa alerta que a busca não encontrou resultados. Depois é mostrado um resumo dos contatos encontrados, dos quais o usuário deve escolher algum para mostrar seus detalhes. Caso um número fora da seleção seja digitado, é alertada a falha e a execução volta para o loop inicial.

A screen shot of a computer program

Description automatically generated

Figura 5: lista ordenada para armazenamento

Para utilizarmos listas ordenadas no código, usamos para o armazenamento dos contatos. Então a toda adição, um novo contato é adicionado à essa lista ordenada para que, mais tarde, possa ser utilizado em outra função.

O código define a criação de um nó (ou *node*) com os dados do contato. Um nó é o componente unitário da lista; vários nós criam a estrutura virtual da lista, que não existe como apenas uma variável, mas uma sequência de objetos conectados na memória.

A partir daí o código se torna mais obtuso, pois é necessário organizar os contatos em ordem alfabética. Isso é feito com o auxílio da função *strcmp*, pertencente à biblioteca nativa *string.h*. Ela responde a relação entre a primeira e segunda strings passadas como argumento. Retorna -1 se a primeira vier antes, 1 caso venha depois ou 0 quando forem idênticas.

Caso a origem for ignorada, itera-se pela lista buscando o primeiro contato com o nome que seja ordenado antes ao novo, parando a busca no fim da lista. Com isso em mãos, é possível alterar o endereço de memória do nó anterior para apontar para o novo contato e o novo contato, por sua vez, para o restante da lista.

Com isso, conseguimos implementar muitas coisas vistas, estudadas e implementadas em aula através de exemplos e exercícios em nosso próprio projeto de gerenciamento de contatos.

Para melhor visualização do código e dos desenvolvedores do código, segue uma tabela com o link do GitHub de cada envolvido no projeto:

Tabela 1: GitHub dos desenvolvedores

|  |  |
| --- | --- |
| Bianca De Freitas Soares | https://github.com/ibellatrxx |
| Bruno Leonardo Martinho | https://github.com/Bruno-Martinho |
| Giovanni De Pita Cicero | https://github.com/Walrus01 |
| Jorge Antonio T. Novaes De Santana | https://github.com/JorgeTerence |
| Ruan Defavari Godoi | https://github.com/ruangodoi |
| Sarah Alves De Melo | https://github.com/sarahamelo |

Para visualizar o projeto do gerenciamento de contatos, você deve ir ao perfil de algum dos integrantes e procurar por um repositório que comece assim: FATEC-AMDS-ED2024-1. Após entrar no repositório procure por uma pasta chamada “Gerenciamento de Contatos”. Lá estará todos os arquivos necessários do sistema.

# TESTES DE SOFTWARE

Foi citado na introdução desse documento que fizemos muitos testes para que o sistema criado seja utilizado com excelência entre seus usuários. Vamos adentrar mais os testes nesse capítulo.

Para que tenhamos ideias de testes a serem feitos, foram necessários testar o mesmo código muitas vezes e fazer muitas perguntas ao longo de seu desenvolvimento e além dele.

“Consigo colocar letras no telefone?”

“Se eu pesquisar pelo nome, irá aparecer todos os contatos com o mesmo nome?”

São perguntas como essa que fazemos durante e depois do desenvolvimento do código para que possamos aperfeiçoar o mesmo de forma que, quando utilizado, não ocorra problemas, bugs, ou outras ações indesejadas.

Para organizar nossos testes, fizemos uma planilha de testes. Essa planilha foi inspirada em um modelo previamente mandado pelo professor. Veremos mais sobre cada teste que foi feito durante o desenvolvimento do código do projeto.

Interface gráfica do usuário, Aplicativo, Tabela, Word, Excel

Descrição gerada automaticamente

Figura 6: plano de testes

Os primeiros testes foram os testes das principais funções do gerenciamento de contatos. As principais funções são: adição, exibição, consulta, edição e exclusão de um contato.

O primeiro teste foi sobre a edição. Queríamos testar se o código estava realmente adicionando o contato à lista ordenada, ou seja, se o código estava armazenando o novo contato. Após a inserção do nome, sobrenome, telefone e e-mail, o contato foi adicionado à lista e, consequentemente, foi armazenado.

Texto

Descrição gerada automaticamente

Figura 7: teste de cadastro

O segundo teste foi para a exibição desse novo contato e, principalmente, em ordem alfabética. Para isso, adicionamos mais um contato a lista, somando dois contatos ao total e pedidos para consultar. Após a solicitação de exibição dos contatos, o código nos mostra os contatos adicionados em ordem alfabética.

Texto

Descrição gerada automaticamente

Figura 8: teste de exibição

O terceiro teste foi para consultar um contato em específico baseado no nome. Então ao chamar a função de consulta, o código pergunta qual o nome do contato que queremos que ele ache para nós. Ao inserir o nome, é feito uma busca e acha todos os contatos com o mesmo nome, diferenciando apenas no sobrenome.

Texto

Descrição gerada automaticamente

Figura 9: teste de consulta por nome

O quarto teste foi para editar um contato. Quando solicitamos a edição de um contato, o programa pede o nome da pessoa que queremos que os dados sejam editados, mas, em muitos casos, já contatos com o mesmo nome, porém outros dados diferentes. Para isso, o sistema lista os contatos com aquele nome e pede qual deles você deseja editar, veja abaixo:

Texto

Descrição gerada automaticamente

Figura 10: teste de edição

O quinto teste foi sobre a exclusão de um contato. Então quando solicitamos a função de exclusão, o sistema nos pergunta qual o nome e o sobrenome do contato que você gostaria de excluir. O sistema pergunta o sobrenome, pois, como em muitas agendas de contatos, há pessoas quem tem o mesmo nome a sobrenome diferentes. Após inserir ambos, nome e sobrenome, o contato é excluído da lista de contatos. Para ter certeza dessa exclusão, pedimos ao sistema para que exibisse todos os contatos.

Texto

Descrição gerada automaticamente

Figura 11: teste de exclusão

Tivemos outros pequenos testes que foram as soluções de algumas perguntas como as citadas anteriormente. Então vamos respondê-las?

Para responder “Consigo colocar letras no telefone?”. Pedimos ao código para adicionar um contato e, na opção “telefone” colocamos uma letra e o sistema impede que um telefone com alguma letra possa ser adicionado à lista de contatos.

Texto

Descrição gerada automaticamente

Figura 12: teste de letras no telefone

Sobre a pergunta “Se eu pesquisar pelo nome, irá aparecer todos os contatos com o mesmo nome?”. Ao solicitar a consulta de um contato baseado em seu nome, ele irá pesquisar todos os contatos que possuem o mesmo nome, mesmo com que sobrenomes diferentes. Isso só é possível pois o nome e o sobrenome são partes separadas no armazenamento.

Texto

Descrição gerada automaticamente

Figura 13: teste de consulta pelo mesmo nome

Com os testes feitos e dados como bem-sucedidos, concluímos que o programa está dentro dos critérios de qualidade. Sendo assim, o sistema de gerenciamento de contatos pode ser usado de forma que não haja corrupções, erros ou outras ações indesejadas.

# CONCLUSÃO

Em conclusão, após estudos feitos durante as aulas de Estrutura de Dados, lecionadas pelo Prof. Carlos Henrique Veríssimo, pudemos fazer um projeto na linguagem C sobre um sistema de gerenciamento de contatos que seja: rápido, eficiente, bem-estruturado, intuitivo, fácil de usar e que tenha passado por todos os testes de qualidade.

O programa desenvolvido ajudou para que pudéssemos integrar todo conhecimento adquirido no primeiro semestre da disciplina. Ou seja, o desenvolvimento do software em conjunto da equipe reforçou e aprimorou nossas habilidades técnicas em Estrutura de Dados.