

# Documentação - Trabalho Prático

## Algoritmos e Estruturas de Dados (2)

### Introdução

“Estruturas de Dados”, segundo seu significado, é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento. Dessa forma, se torna evidente a importância do conhecimento desse assunto para todos aqueles que são, ou pretendem ser, desenvolvedores, visto que, essa matéria tem larga aplicação em atividades como gerenciamento de memória, execução de processos, armazenamento e gerenciamento de dados, tópicos que possuem extrema relevância em inúmeros projetos.

O objetivo do trabalho proposto é implementar um simulador de um servidor de *emails*, de modo que, o sistema dará suporte ao gerenciamento de contas, assim como a entrega de mensagens entre os usuários. É importante ressaltar a necessidade de um bom aproveitamento de memória e de uma implementação eficiente.

Com isso, tendo em vista as regras apresentadas pelo trabalho, a aplicação se torna uma forma intuitiva de aprendizado de assuntos como Pilhas, Listas e Filas, e também uma forma de primeiro contato com algo que poderá ser visto em trabalhos futuros, facilitando e exercitando as maneiras de pensar em um sistema eficaz.

### Implementação

Para a implementação do trabalho, de início, foi notada a necessidade da utilização de estruturas (structs), visto que essas nos permitem trabalhar com diversos tipos de informações de maneira rápida e organizada, e são muito utilizadas em programas que necessitam e fazem o uso de vários tipos de variáveis e características, além de possuírem auto referência (uso de ponteiros).

Dessa forma, o código conta com uma struct inicial que tem como objetivo criar um tipo email para identificar o usuário, esta também define a prioridade, ou seja a ordem em que os emails serão apresentados e a definição de um ID (***struct Email***).

Outra struct implementada também foi a (***struct Usuario***), que tem como objetivo criar um usuário para adição, consulta e remoção.

## Funções e Procedimentos

O código possui as seguintes funções:

***user \*NovoUsuario()***{: Função que tem como objetivo alocar dinamicamente a lista de usuário, criando a “Cabeça da lista” e retornando-a.

***email \*constroimg()***{: Função que tem como objetivo alocar dinamicamente o espaço de memória para a lista de emails, criando também, a já mencionada cabeça da lista.

***void Cadastrar(user \*CriaUsuario, FILE \*LISTA)***{: Função que tem como objetivo adicionar um usuário, essa possui dois ponteiros auxiliares e uma alocação dinâmica “NEWUSER”.

Além disso, a partir dessa função é possível verificar se o usuário cadastrado já é existente. Caso o usuário seja novo, ele é adicionado a lista apontando para o próximo elemento.

Complexidade: laço de repetição  $O(n)$ .ii  $O(1)$  + laço de repetição  $O(n)$ .  
(if  $O(1)$  else  $O(1)$ )/2 =  $O(n)$ .

***void Remover(user \*CriaUsuario, FILE \*LISTA)***{: Função que tem como objetivo remover usuário, possui um ponteiro auxiliar que irá percorrer a lista e outro que irá realizar a remoção

Complexidade: laço de repetição  $O(n)$ .( $O(1)$ . $O(1)$ )/2 =  $O(n)$

***int Enviar(FILE \*LISTA, user \*CriaUsuario, email \*CriaEmail)***{: Função que tem o objetivo de cadastrar a mensagem no sistema. Possui um vetor de char que salva o conteúdo de cada email, realiza a leitura do ID, a prioridade e o conteúdo do email (fscanf, fgets), um ponteiro para saber onde ocorre o término da mensagem. Caso a lista esteja vazia ela é inteiramente percorrida por uma auxiliar.

Complexidade : laço de repetição  $O(n)$ . $O(1)$ ( $O(1)$ . $O(n)$ )/2 =  $O(n^2)$

***void Consultar(FILE \*LISTA, user \*CriaUsuario, email \*CriaEmail)***{: Função que tem como objetivo consultar o usuário. Verifica a existência do usuário, caso seja existente +1 é adicionado ao contador.

Além disso, nessa função também ocorre uma alocação para guardar o email e para ter a garantia de que há espaço para a mensagem. A lista é percorrida para ver a disponibilidade de mensagens na caixa de entrada e garantir que o email mostrado seja o de maior prioridade.

Complexidade:  $O(n)$ . $O(1)$ + $O(1)$ + $O(n \log n)$ +  $O(1)$ + $O(1)$ . $O(n)$ = $O(n \log n)$

## Programa Principal

O programa principal declara duas variáveis, uma do tipo usuário para a utilização nas listas de email e outra do tipo email para a lista de emails.

Após a declaração chama duas funções, a que irá alocar a cabeça da lista de usuários e a que irá alocar a cabeça da lista de emails. É importante ressaltar que o vetor Char é utilizado para armazenar a opção do usuário contida no arquivo. Além disso, há também uma confirmação de que o arquivo exista, caso contrário exibe uma mensagem de que o arquivo não poderá ser lido.

Seguido de todos esses detalhes superiores, em uma sequência de (if's) são chamadas as funções CADASTRAR, ENVIAR, REMOVER E CONSULTAR.

Logo em seguida a lista é fechada e a função retorna zero.

Organização do Código

O código possui 4 bibliotecas

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

```
#include<locale.h> (para a implementação de caracteres (acentuação))
```

Não possui nenhum tipo de constante implementada dessa forma, pois faz parte das regras do trabalho.

O compilador utilizado para a execução do código final foi o gcc.exe(TDM-2 mingw32) presente no aplicativo Falcon C++, no sistema operacional Windows 10, para executá-lo basta digitar f9 do teclado caso o aplicativo usado seja o mesmo.

## Testes

Para a criação do código foram utilizados dois aplicativos:

Falcon (versão para Windows)

Atom (versão para Linux)

Os testes foram realizados em dois computadores

– AMD Quad-Core A12-9720P RADEON R7 12 COMPUTE CORES 4C+8G 2.70GHz

– Intel(R)Core(TM)i3-4005 CPU @ 1.70GHz k1.70GHz

A figura abaixo mostra uma saída com erros na execução

```
C:\Users\marce\Desktop\Milagres Existem\Jesus Me Ajuda.exe
#####
Trabalho Organizado Por: Marcela Netto e Marcelo Oliveira
Orientador: Rafael Sachetto
*****Nao Aceite Copias*****
Usuario Não se esqueça de Colocar os Comandos no Arquivo de texto
Comandos Contidos No Arquivo
-----
CONTA 2 CADASTRADA COM SUCESSO

CONTA 5 CADASTRADA COM SUCESSO

OK: MENSAGEM PARA 5 ENTREGUE
ERRO: CONTA 5 NAO EXISTE

OK: MENSAGEM PARA 5 ENTREGUE
ERRO: CONTA 5 NAO EXISTE
```

Já, esta figura abaixo apresenta a saída de uma execução típica com o programa e código em seu devido funcionamento

```
C:\Users\marce\Desktop\Milagres Existem\Jesus Me Ajuda.exe

*****Nao Aceite Copias*****

Usuario Não se esqueça de Colocar os Comandos no Arquivo de texto
Comandos Contidos No Arquivo
-----

CONTA 2 CADASTRADA COM SUCESSO

CONTA 5 CADASTRADA COM SUCESSO

MENSAGEM PARA 5 ENTREGUE

MENSAGEM PARA 5 ENTREGUE

CONSULTA 5: O SACHETTO FOI NO CALDEIRAO DO HUCK

CONSULTA 5: MAS O MADEIRA FEZ ELE PERDER

O Processo retornou 0   tempo de execução : 4.618 s
Pressione uma tecla para continuar...
```

## **Conclusão**

Tendo em vista todas as recomendações, estudos em sala e externos, discussões sobre o tema e implementações realizadas, foi possível finalizar o trabalho dentro do prazo descrito e com aproveitamento total de tudo aquilo que foi proposto.

Em relação as dificuldades encontradas pelo caminho, a que mais exigiu esforço e estudo da dupla foi a criação da “caixa de entrada”, essa que tinha como objetivo armazenar as mensagens enviadas para o usuário e exibir aquela de maior prioridade.

Para resolver os problemas encontrados houve estudos paralelos a partir de leituras (obra declarada em referências) e também o comparecimento da dupla na monitoria disponibilizada pelo orientador e pela universidade.

Dessa forma, concluímos que a realização do trabalho foi de suma importância para o aprendizado sobre estrutura de dados, visto que trabalhamos com diversos tipos de listas até chegarmos no código correto e entregamos o trabalho com maior facilidade em enxergar soluções para os problemas propostos pelo assunto.

## **Referências**

Obra *C completo e total - terceira edição*

Autor Herbert Schildt - Osborne

# **Algoritmos e Estruturas de Dados | |**

**Marcella Netto de Oliveira**

**Marcelo Ari de Oliveira**

**Universidade Federal de São João Del Rei - MG**

**25 de Abril de 2019**

