

## Exercício 01 - Agenda

---

David Castello Branco - 2017074157

Leonardo Fernandes Lima Dias de Magalhães - 2017098129

Marcus Vinicius Mascarenhas Honorato - 2017074246

25 de Abril de 2019

### 1 INTRODUÇÃO

Esse trabalho tem como objetivo realizar a implementação de um agenda eletrônica que armazena os compromissos em seus respectivos dias e horários. Para realizar tais ações, será usado o tipo de dados lista encadeada para fazer as conexões entre as estruturas criadas.

### 2 DESENVOLVIMENTO

#### 2.1 ESTRUTURAS DE DADOS

Para este trabalho, foi feito apenas um arquivo para o TAD usado. As estruturas de dados usadas foram um tipo Agenda que guarda uma lista de meses; um tipo Mes que guarda uma lista de dias e um ponteiro para o próximo elemento da lista além do nome do mês e o total de dias que ele possui; um tipo Dia que armazena o total de compromissos daquele dia e uma listas de compromisso; e um tipo Compromisso que armazena a hora e a descrição do compromisso, além de um ponteiro para o próximo elemento da lista.

## 2.2 FLUXO DE EXECUÇÃO

O fluxo de execução do trabalho pode ser dividido em 3 partes:

1. Inicialização das estruturas Essa etapa consiste em inicializar todas as estruturas de cada mês e realizar a leitura do arquivo "historico.txt" para armazenar os dados previamente salvos de outras interações.
2. Preenchimento das estruturas Essa etapa consiste na interação do usuário com o sistema. Após escolher o mês e o dia ao qual quer se referir, ele terá a opção de inserir ou deletar um compromisso naquele dia e listar os compromissos do dia. Essas ações sempre acontecerão para o dia selecionado até que o usuário solicite para trocar.
3. Encerramento das estruturas Após realizar suas funções, quando o usuário selecionar para fechar a agenda, todos os dados da agenda serão impressos no arquivo "historico.txt" para ficar salvo para as interações futuras.

## 2.3 FUNÇÕES E PROCEDIMENTOS

A seguir, serão explicadas as funções de maior importância do trabalho.

VOID INSERE\_LISTA\_COMP(COMPROMISSO \*\*C, INT HORA, INT MIN, STRING DESC);  
Função que irá inserir um item na lista de compromissos. Essa função irá percorrer sempre que o horario do compromisso a ser inserido for menor que o horário do compromisso da posição atual da célula. No momento em que achar um horário maior, o compromisso será inserido antes deste em questão. Caso encontre um compromisso no horário exato do que está para ser inserido, irá mostrar uma mensagem informando que já existe compromisso nesse horário e não será realizada a inserção.

VOID DELETA\_LISTA\_COMP(COMPROMISSO \*\*C, INT HORA, INT MIN); Função que irá deletar um item na lista de compromissos. Essa função irá percorrer até encontrar um elemento no horário exato do que foi passado para a função e nesse momento o compromisso será deletado da lista, ou seja, será reorganizado os ponteiros e dado free naquela célula. Caso percorra a lista toda até achar um NULL, irá mostrar uma mensagem informando que não há compromisso nesse horário para ser deletado.

VOID FECHA\_agenda(Agenda\*calendario); Função que irá dar free em todas as estruturas alocadas da agenda.

VOID INICIA\_AGENDA(AGENDA\*\*); Função que irá iniciar as estruturas da agenda. Irá criar a lista encadeada de meses assim como inicializar como vazia a lista de compromisso de todos os dias de todos os meses.

VOID CARREGA\_AGENDA(AGENDA\*\*); Função que irá ler todo o arquivo e verificar se há informações para serem escritas na agenda ou não. O arquivo tem o nome padrao de "histórico.txt" e caso não exista, este será criado. Seu texto também contém um padrao para facilitar a leitura. Inicialmente é escrito o nome do mês precedido do caractér '-' para identificar, por exemplo "-Janeiro". Na linha de baixo, caso existam compromissos naquele mês, virá qual o dia referente precedido do caracter ", por exemplo "01". Abaixo do dia, virão os compromissos, um por linha, no seguinte formato "HH:MM Descrição" por exemplo "00:00 Ano Novo".

VOID REALIZA\_OPERACOES\_AGENDA(MES\*\*, INT); Essa função irá controlar toda a interação do usuário com os compromissos do dia. É mostrado um menu com 4 opções e cada uma delas realizará uma ação.

1. Inserir Compromisso Irá solicitar a hora e a descrição do compromisso para inseri-lo na lista de compromissos do dia selecionado. A lista sempre se manterá ordenada crescentemente de acordo com a hora dos compromissos. Caso já exista um compromisso naquela hora e minuto exatos que o usuário está tentando imprimir, um aviso informando que já existe um compromisso naquele horário será informado e a inserção não irá ser realizada.
2. Remover Compromisso Irá solicitar a hora para buscar o compromisso em questão e remove-lo da lista. Caso não exista compromisso naquela hora, será exibida uma mensagem de que o compromisso não foi encontrado.
3. Listar Compromissos Irá mostrar todos os compromissos, ordenados pela hora, daquele dia.
4. Trocar o dia Irá permitir o usuário voltar para tela em que ele pode escolher o dia ou trocar de mês.

Essa função irá rodar infinitamente até que a opção 4 seja selecionada.

VOID SALVA\_AGENDA\_ARQUIVO(AGENDA \*CALENDARIO); Função que irá pegar toda a agenda e imprimir seu conteúdo no arquivo historico.txt, seguindo o padrão de escrita explicado anteriormente. Se o mês não tiver compromisso, apenas seu nome aparecerá no arquivo, dias sem compromissos não serão escritos.

## 2.4 PROGRAMA PRINCIPAL

A main do programa realiza o controle de fluxo do programa. Inicialmente é feita a inicialização de todas as estruturas e logo após vem a parte de preenchimento das estruturas. Primeiramente, o usuário ter duas opções, escolher um mês ou encerrar a agenda (essa opção irá leva-lo para a etapa de Encerramento das estruturas). Ao escolher o mês, ele terá a opção de escolher o dia ou retornar para escolher o mês. Ao escolher o dia, será mostrado o menu de opções mostrado na função realiza\_operacoes\_agenda(). Quando selecionar para fechar a agenda, o programa irá chamar a função de imprimir no arquivo de saída e encerrará o programa.

## 2.5 DESCISÕES DE IMPLEMENTAÇÃO E DETALHES TÉCNICOS

O trabalho foi desenvolvido no ambiente linux e compilado com o g++. O código está dividido em 3 arquivos principais: agenda.hpp e agenda.cpp, que implementam o TAD agenda com suas funções usadas no programa, e main.cPP, que implementa o programa principal. Para rodar o programa basta abrir o terminal e digitar o comando `g++ -Wall -std=c++11 *.cpp -o main`, depois, digitar `./main`. Para execução do programa, basta seguir as instruções em tela e manter o padrão de entrada que o programa espera nas instruções.

## 3 CONCLUSÃO

Várias foram as dificuldades encontradas durante a execução do trabalho. As principais foram a criação das funções da lista e a manipulação dos ponteiros. A leitura e tratamento das palavras do arquivo foram tranquilas por seguirem um padrão previamente estabelecido pelos programadores. Entretanto, as dificuldades foram superadas e o trabalho correu sem problemas.

## 4 REFERÊNCIAS

- [1] <http://www.cplusplus.com/doc/tutorial/files/>
- [2] <http://www.cplusplus.com/doc/tutorial/structures/>
- [3] <https://www.geeksforgeeks.org/linked-list-set-1-introduction/>