

# UNIVERSIDADE FEDERAL DE MINAS GERAIS

## Exercício 1 - Agenda

Aline Fontoura Silva - 2018108578  
Bruna Fontes Almeida - 2017088930  
Bruno Antônio Manzolli - 2017111362

28 de Abril de 2019

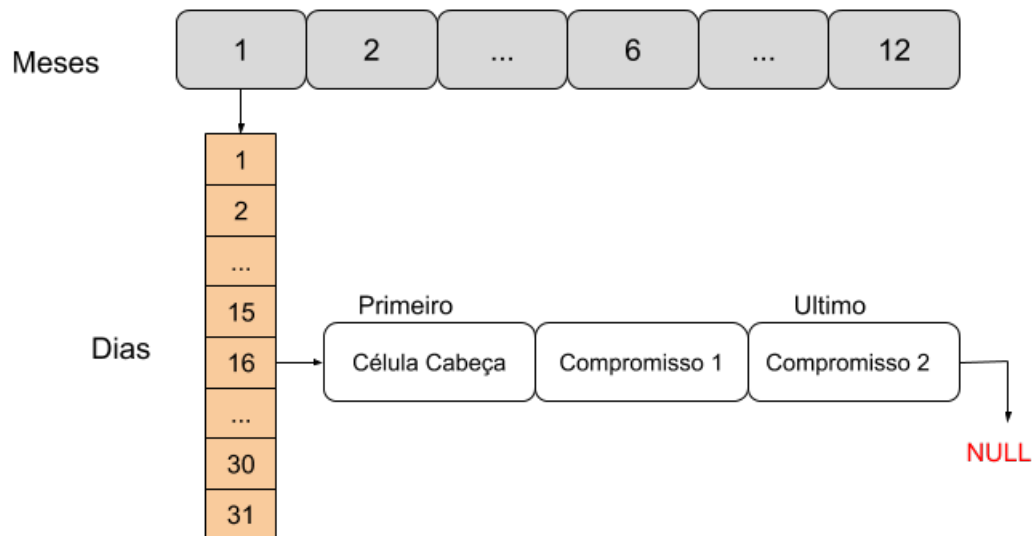
## 1 Introdução

Uma lista encadeada é uma estrutura composta por células a qual uma célula possui um apontador(ponteiro) para a célula seguinte. Com isso, este trabalho tem como objetivo utilizar diferentes listas para realizar a implementação de uma agenda anual.

## 2 Implementação

### 2.1 Estruturas de Dados

Um TAD com nome lista foi criado para representar as estruturas utilizadas. Existem duas listas, uma que irá representar os meses onde cada mês possui um conjunto de dias e cada dia possui uma outra lista de compromissos, o qual contém o horário e a descrição do compromisso. Abaixo encontra-se uma imagem de como os dados formam estururados.



## 2.2 Funções e Procedimentos

As funções a seguir são as que estão presentes no TAD lista.

- **void preencheAgendaArquivo(Agenda \*lista);** Função responsável por ler o arquivo, caso este exista, e preencher a agenda com os compromissos do arquivo.
- **void imprimeAgendaArquivo(Agenda lista);** Função responsável por imprimir todos os compromissos da agenda no arquivo "agenda.txt".
- **void iniciaAgenda(Agenda \*lista);** Função que cria os meses e os insere em forma de lista além de alocar e inicializar as estruturas de cada mês.
- **void insereMes(ItemMes item, Agenda \*lista);** Função que irá inserir os meses na lista. Essa possui célula cabeça e a inserção é sempre feita no final da lista.
- **Mes \*buscaMes(int valorBuscado, Agenda \*lista);** Função que percorre a lista de meses e retorna o ponteiro pro mês solicitado.
- **void liberaAgenda(Agenda \*lista);** Função que irá dar free na agenda.
- **bool listaVazia(Compromisso \*lista);** Verifica se a lista está vazia.
- **void iniciaListaCompromisso(Compromisso \*lista);** Função que aloca as estruturas da lista.
- **void insereCompromisso(ItemCompromisso item, Compromisso \*lista);** Função que irá inserir os compromissos na lista. Essa possui célula cabeça e a inserção é sempre feita no final da lista.
- **void removeCompromisso(int hora, int minutos, Compromisso \*lista);**

- **void liberaListaCompromisso(Compromisso \*lista);** Função que da free nos elementos da lista.
- **void imprimeListaCompromisso(Compromisso \*lista);** Função que percorre a lista e imprime as informações do compromisso.
- **bool verificaCompromisso(int hora, int minutos, Compromisso \*lista);** Função que percorre a lista de compromisso para verificar se já existe algo naquele horário.

## 2.3 Programa Principal

A main do trabalho irá organizar todo o fluxo de execução do programa. Primeiro será realizado a leitura do arquivo com os compromissos anteriores para inserção na agenda, feito isso, será mostrada as opções de manipulação da agenda para o usuário.

O usuário poderá realizar 4 operações, sendo elas inserir ou deletar um novo compromisso, listar todos os compromissos do dia e verificar a existência de um compromisso naquele dia. Ao fim do programa, todos os compromissos agendados serão impressos no arquivo.

## 2.4 Decisões de Implementação e Detalhes Técnicos

O trabalho contém três arquivos. Dois compõem o TAD lista, lista .cpp e .hpp, e o main.c que é o arquivo principal. Apenas um TAD foi criado pois todas as funções implementadas usam estruturas definidas no mesmo arquivo e por questões de dependência, foi preferido unir tudo. O trabalho foi feito no linux, na linguagem C++ e compilado com o g++. Para compilar o trabalho, basta abrir o terminal e digitar g++ \*.cpp -o main. Para executar o programa, basta digitar o comando ./main. Caso o arquivo agenda.txt não exista, ele será criado ao fim da execução com a agenda criada. Caso já exista, ele será importado para o programa no início da execução.

## 3 Conclusão

Inicialmente, a principal dificuldade encontrada no trabalho foi a implementação das listas e seu funcionamento. A leitura do arquivo e inserção na lista também teve um grau de dificuldade, mas ao decorrer do tempo as implementações foram concluídas. Mesmo com as dificuldades apresentadas, o trabalho foi finalizado e seu objetivo foi concluído.

## Referências

- [1] <https://web.inf.ufpr.br/menotti/ci056-2015-2-1/slides/aulaEDListaEnc.pdf>
- [2] <http://www.ic.unicamp.br/ra069320/PED/MC102/1s2008/Apostilas/Cap10.pdf>