
CURSO: ENGENHARIA DE SOFTWARE

DISCIPLINA: Estruturas de Dados 1

TURMA: B

SEMESTRE: 2019.2

CRÉDITOS: 4

PROFESSOR: Rafael Fazzolino

LISTA DE EXERCÍCIOS 3

1. Listas Encadeadas

- 1) Crie um algoritmo que seja capaz de registrar idades em uma lista duplamente encadeada. O programa deve permitir que o usuário insira quantas idades desejar, até o momento em que o usuário entrar com uma idade negativa (idade < 0). Então o programa deve permitir que o usuário escolha a opção de listagem (do primeiro para o último, ou do último para o primeiro).
- 2) Crie a função `void *push_back(List *list, Node *node)`. Esta função insere um novo elemento no final da lista.
- 3) Crie a função `void erase(List *list, Node *node)`. Esta função remove o elemento node da lista.
- 4) Continuando o código das questões anteriores, crie uma função `pop(List *list)`. Esta função remove o primeiro elemento da lista.
- 5) Continuando o código das questões anteriores, crie uma função que remove o último elemento da lista, podemos chamá-la de `pop_back(List *list)`.
- 6) Continuando o código das questões anteriores, crie uma função `void insert(List *list, Node *node, int index)`. Esta função insere determinado elemento na posição desejada, se esta posição for válida para minha lista. Ou seja, uma lista vazia só deve permitir a inserção na posição 0. Uma lista de tamanho 10 só deve permitir a inserção até a posição 10 (começando de 0, ou seja, 11º elemento da lista). Lembre-se de cuidar do **Head** e do **Tail**. Para verificar o comportamento adequado, imprima a lista na ordem normal e na ordem inversa (funções da primeira questão da lista).