

# Módulo 1 - Estruturas básicas - Lista #1

Data de entrega: 26 de março de 2017

Acompanham esta lista os arquivos **mod1\_lista1.h** e **mod1\_lista1.cpp**, contendo as declarações iniciais. Modifique estes arquivos para implementar as questões pedidas e envie-os de volta zipados com nome no padrão <numero\_matricula>.zip por email para **profs-eda@tecgraf.puc-rio.br**, com o assunto **[EDA] Lista 1**. Atenção: Crie um arquivo contendo a função main do seu programa para testar suas implementações, mas envie SOMENTE os arquivos e as classes solicitadas.

---

1. Implemente as funções da classe List, que representa uma lista de inteiros duplamente encadeada. Cada nó é representado por uma struct que contém o valor do inteiro e os ponteiros para o elemento anterior e o próximo.

```
#pragma once

struct Node
{
    int val;
    Node* next;
    Node* prev;
};

class List
{
public:
    //Cria uma lista vazia
    List();

    //Cria uma lista a partir de outra lista
    List(const List& list);

    //Destroi a lista
    ~List();

    //Insere um elemento no fim da lista
```

```

void push_back(const int& element);

//Insere um elemento no inicio da lista
void push_front(const int& element);

//Remove o elemento do final da lista
void pop_back();

//Remove o elemento do inicio da lista
void pop_front();

//Retorna o tamanho da lista
unsigned int size();

//Remove a primeira ocorrencia de element
void remove(const int& element);

//Imprime os elementos da lista separados
//por espaco e por fim salta uma linha
void print();

private:
    Node* first;
    Node* last;
};

```

2. Na classe List implementada na questão anterior, inclua uma função **void sort()**, que ordena os elementos da lista. Implemente um algoritmo baseado em trocas, de complexidade  $O(n^2)$ .