

## = Roteiro 05 =

Algoritmos e Estrutura de dados II - Entrega até 17/Nov às 23:59

Prof. José Rui

[jose.castro@ifsudestemg.edu.br](mailto:jose.castro@ifsudestemg.edu.br)

---

### Atividade 1

**Descrição do problema** - Lista contígua:

Fizemos na sala de aula a classe ListaContigua de Produto; Desta forma, para construir a classe lista contigua de produto, foi necessário construir a classe Produto, e então construir a classe ListaProduto.

1. Pegue o código c++ disponibilizado no DACC
2. Estude minuciosamente cada função que nós fizemos. (lembrando que fizemos juntos, pelo menos, 3 vezes em sala de aula)
3. Você deverá construir agora uma ListaPessoa.
  - a) Para isso será necessário criar a classe Pessoa.
    - considere pessoa ( id, idade, sexo, usaOculos)
  - b) Agora sim, após criar a classe pessoa, com todos os seus atributos e métodos, você poderá iniciar o processo de criação da classe ListaPessoa;
    - Não seja inocente, pegue o código que o professor disponibilizou e faça as devidas alterações nele.
    - Este é um ótimo caminho para você aprender e entender os detalhes da orientação a objetos.

### Atividade 2

**Descrição do problema**

Outro ponto importantíssimo que fizemos na sala de aula é a chamada dos métodos das classes; Fazer corretamente a chamada dos métodos de uma classe é algo essencial na programação. Desta forma faça um programa piloto para chamar as seguintes funções:

- a) Um piloto (uma função main) que chame os métodos: *<está pronta como exemplo...>*
  - `ListaProduto();`
  - `void insert(int posicao);`
  - `void imprime();`

```
int main(){
    ListaProduto prateleira;
    prateleira.insert(3);
    prateleira.insert(0);
    prateleira.imprime();
    return 0;
}
```

b) Um piloto(uma função main) que chame os métodos:

- `ListaProduto(const ListaProduto& outra);`
- `void insert();`
- `void remove(int posicao);`
- `void imprime();`

```
int main(){  
  
    ????  
  
    return 0;  
}
```

c) Um piloto(uma função main) que chame os métodos:

- `ListaProduto(const ListaProduto& outra, int novoTamanho);`
- `void insert();`
- `Produto buscar(int idProduto);`
- `void imprime();`

```
int main(){  
  
    ????  
  
    return 0;  
}
```

d) Um piloto(uma função main) que chame os métodos:

- `ListaProduto(const ListaProduto& outra, int novoTamanho);`
- `void insert();`
- `Produto* buscar(double preco);`
- função para imprimir este vetor que o buscar retornou

```
int imprimirVetor(????){  
    ????  
  
}  
  
int main(){  
  
    ????  
  
    return 0;  
}
```

### Instruções para entrega

Compacte seu projeto dentro de uma pasta como **R05-nomedadupla** e entregue.

OBS: Retire o executável antes de compactar. O gerenciador de e-mail bloqueia executáveis. Se você esquecer de retirar, seu trabalho não chegará no meu e-mail, logo não terá como corrigir. E por favor, sem chorumelllllasssss!!!