



# PROGRAMAÇÃO I

## RELACIONAMENTO DE OBJETOS

Prof. Dr. Daniel Caetano

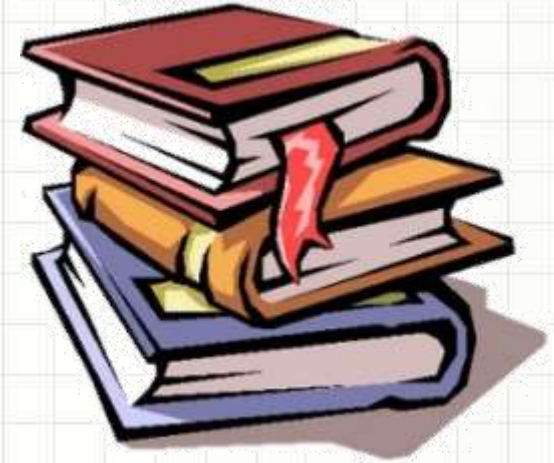
2017 - 1

# Objetivos

- Compreender os diferentes tipos de relacionamento entre objetos
- Implementar os diferentes tipos de relacionamentos



# Material de Estudo



---

## Material

## Acesso ao Material

Apresentação

<http://www.caetano.eng.br/>  
(Programação I – Aula 8)

Material Didático

Programação I – Págs 70 a 74

---



# **RELACIONAMENTOS DE OBJETOS**

# Relacionamentos

- O que existe entre:
  - Motor e carro
  - Turma e aluno
  - Funcionário e computador
- Relações
  - O motor é parte do carro
  - O aluno é parte da turma
  - O funcionário usa o computador

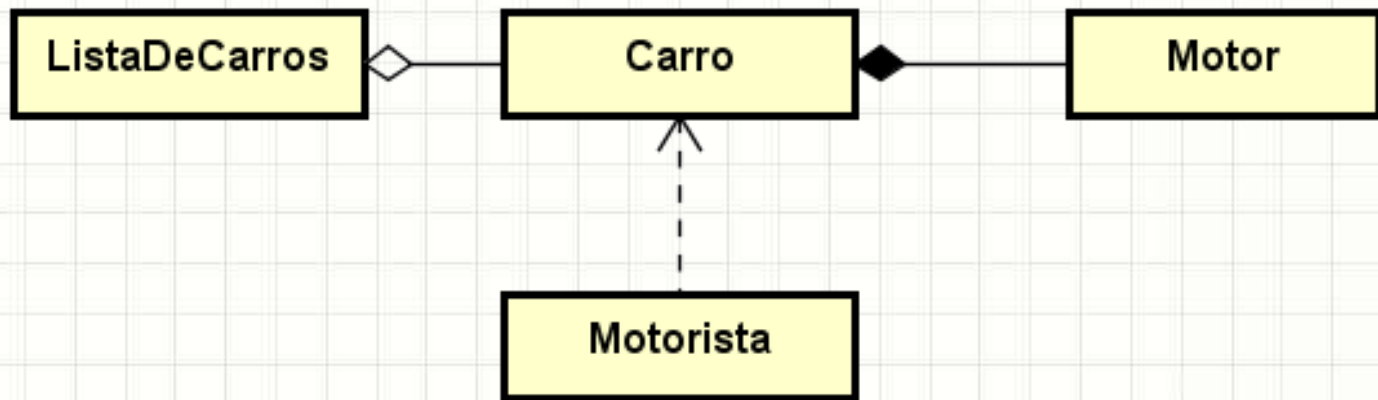


# Relacionamento de Objetos

- Existem três tipos de relações
  - Agregação: Carro → Lista de Carros em Oferta
    - Parte existe sem o todo (posso ter carros sem lista)
    - O mesmo carro pode estar em várias listas
  - Composição: Motor → Carro (venda de veículos)
    - Parte não existe sem o todo (motor sem carro?)
    - Um motor não pode estar em mais de um carro
  - Uso: Motorista → Carro
    - Um objeto não é parte de outro, mas o usa

# Relacionamento de Objetos

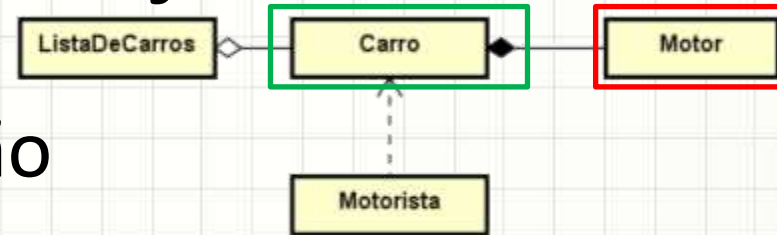
- Representação: Agregação, Composição e Uso



**Nota:** o que é composição em um sistema, pode ser agregação em outro!

Ex.: Concessionária / Venda de Autopeças

# Relacionamento de Objetos



- Programando Composição

```
public class Motor {  
    private float cilindrada;  
}
```

```
public class Carro {  
    private Motor motor;  
}
```



# Relacionamento de Objetos



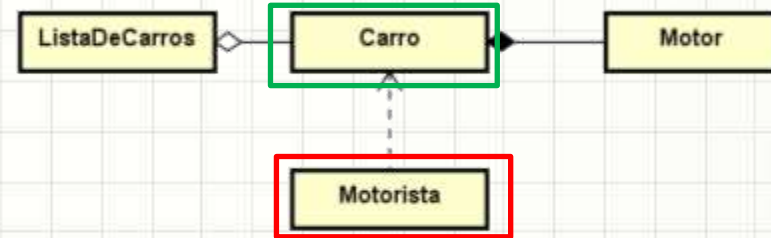
- Programando Agregação

```
public class Carro {  
    private Motor motor;  
}
```

```
public class ListaDeCarros {  
    private Carro[] carros;  
}
```

# Relacionamento de Objetos

- Programando Uso



```
public class Carro {
    private Motor motor;
}
```

```
public class Motorista {
    public void dirige(Carro carro) {
        carro.acelera();
    }
}
```



# **PROGRAMANDO ASSOCIAÇÕES**

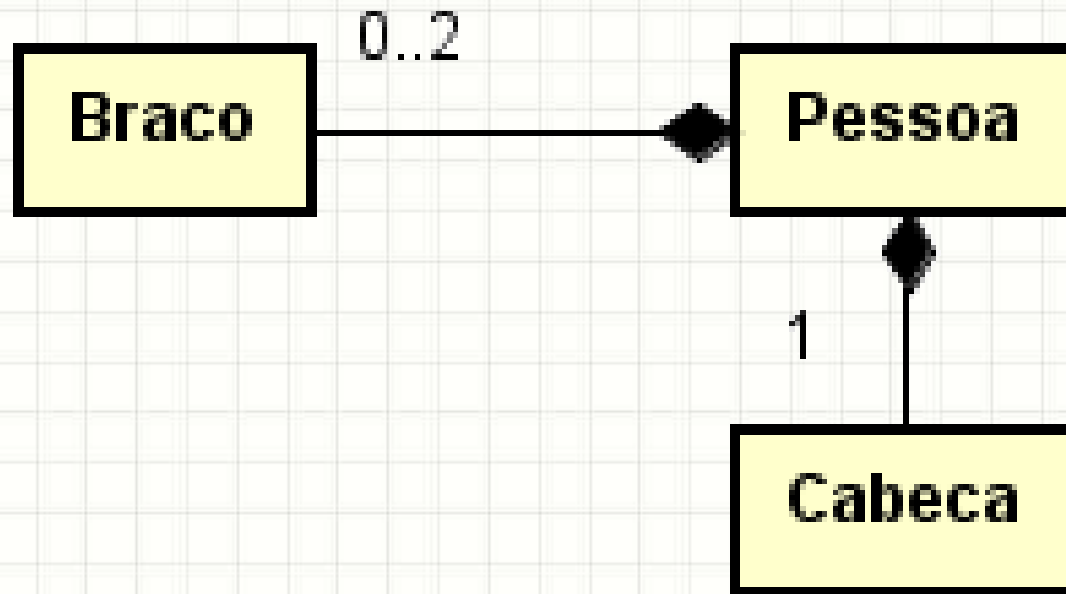
# Exemplo de Associação Simples

- Projeto: **Humano**
- Classe Pessoa
- Classe Cabeça



# Exercício 1

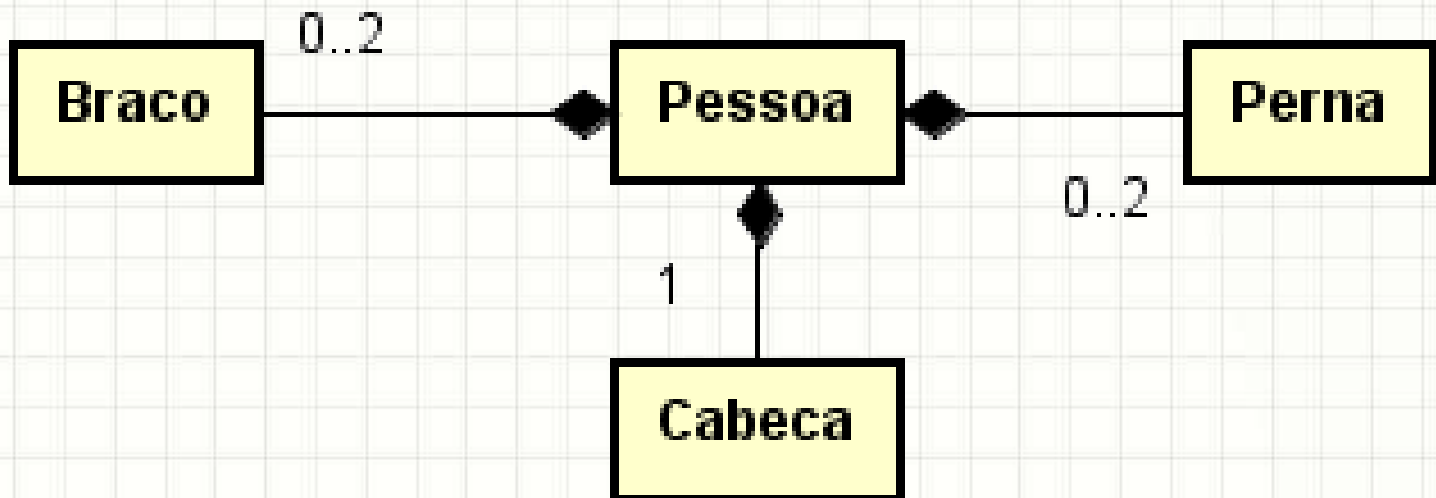
- Projeto: **Humano**
- Classe **Braco**





# Exercício 2

- Projeto: **Humano**
- Classe Perna



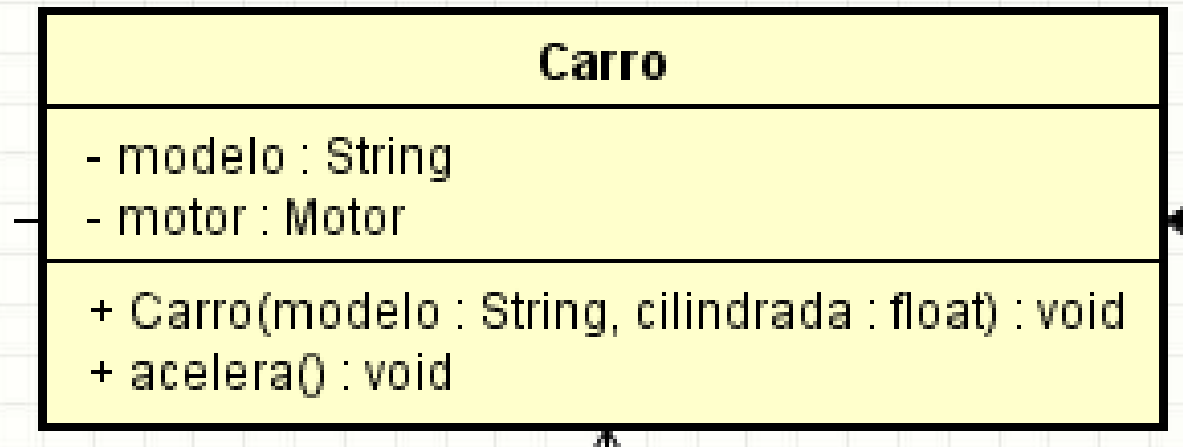
# Exemplo de Associação

- Projeto: **Dirigindo**
- Classe Motor
  - float cilindrada
  - Método: `getVelocidade()`
    - Cilindrada até 1.0 → 140km/h
    - Cilindrada acima de 1.0 a 1.6 → 180km/h
    - Cilindrada acima de 1.6 a 2.0 → 220km/h
    - Cilindrada Acima de 2.0 → 260km/h

Motor
- cilindrada : float
+ <code>getVelocidade()</code> : int

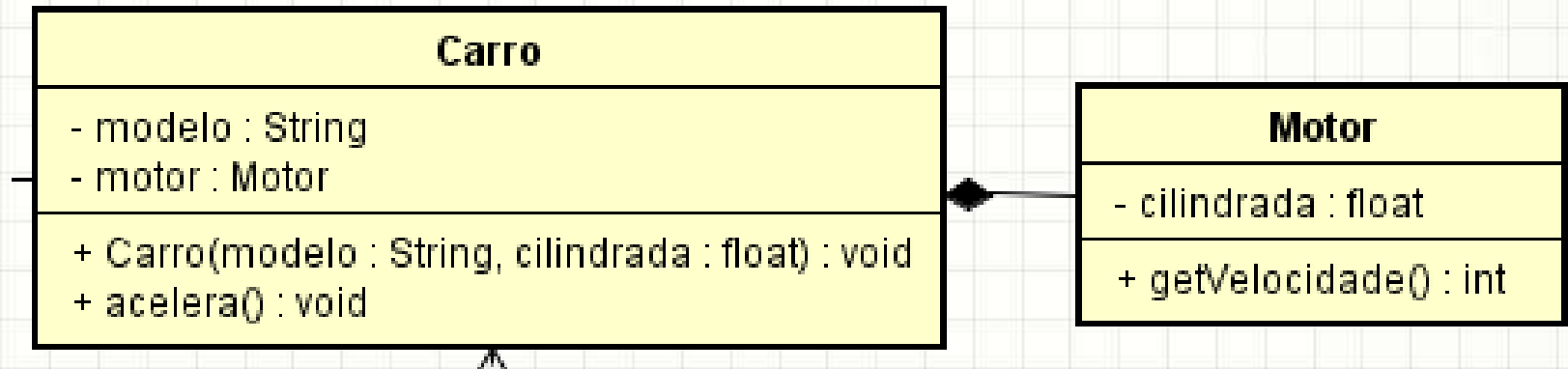
# Exemplo de Associação

- Projeto: **Dirigindo**
- Classe Carro
  - String modelo
  - Motor motor
  - Método: acelera()



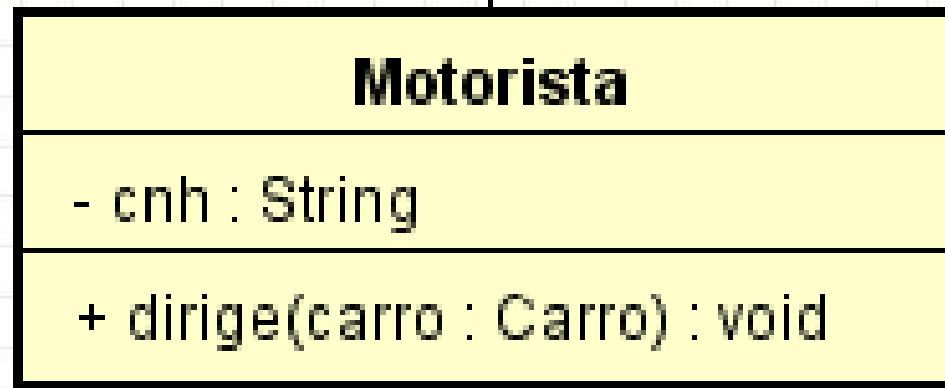
# Exercício 3

- Projeto: **Dirigindo**
- Classe Dirigindo, main:
  - Criar um novo carro
  - Imprimir esse carro



# Exercício 4

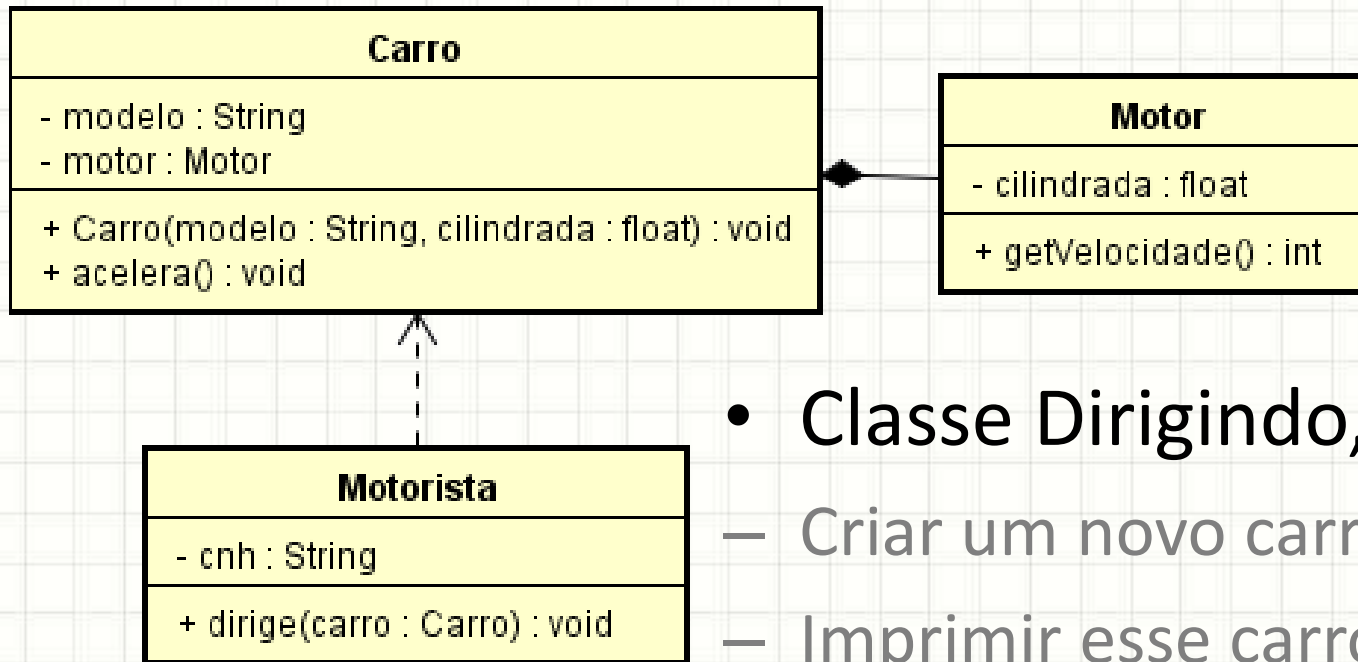
- Projeto: **Dirigindo**
- Classe Motorista
  - String cnh
  - Método: dirige(Carro) → executa acelera()





# Exercício 5

- Projeto: **Dirigindo**



- Classe Dirigindo, main:
  - Criar um novo carro
  - Imprimir esse carro
  - Criar um motorista
  - Faça-o dirigir o carro

# Exemplo de Associação

- Projeto: **Familia**

- Classe Pessoa

- String nome

- Pessoa pai

- Pessoa mae

- toString()

- Imprimir nome da pessoa
- Nome do pai (se não for null)
- Nome da mãe (se não for null)

<b>Pessoa</b>
- nome : String - pai : Pessoa - mae : Pessoa
+ Pessoa(nome : String) : void



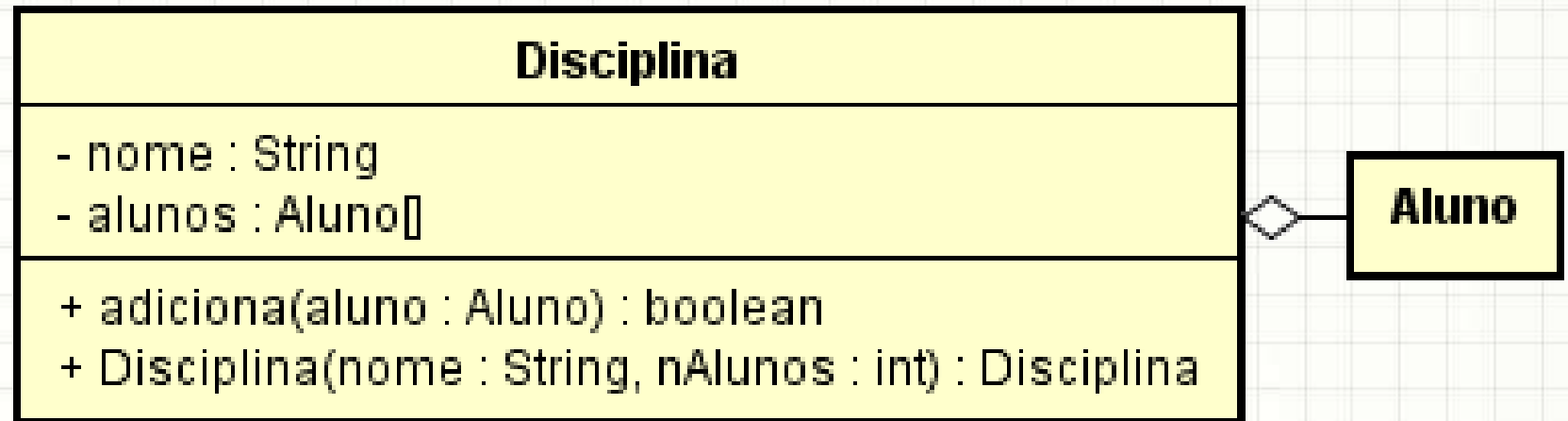
**PERGUNTAS?**



# PARTE PRÁTICA

# Experimentando

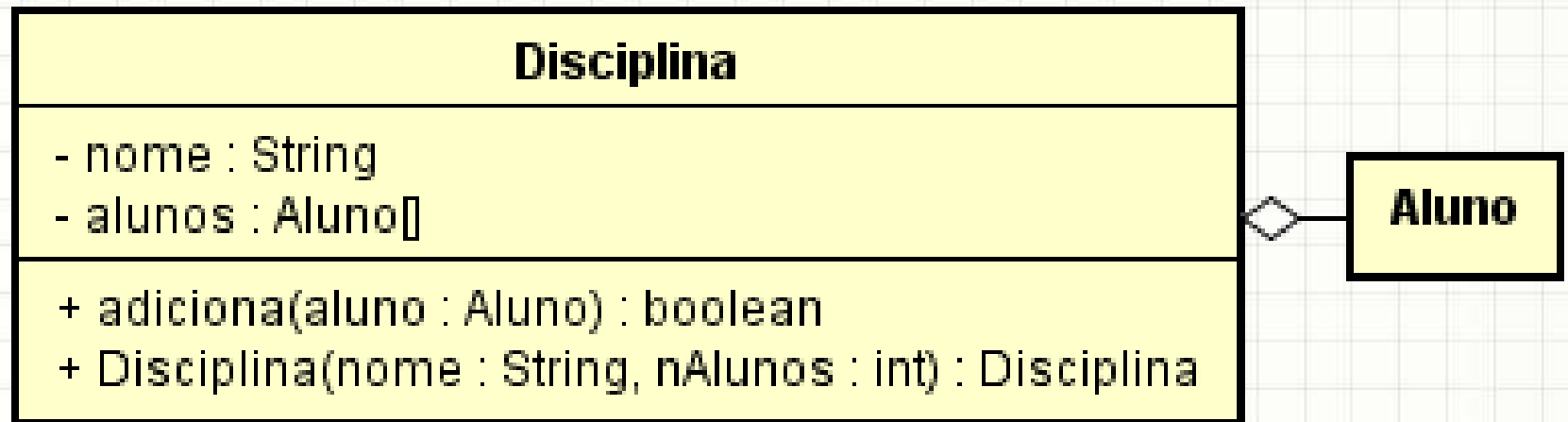
- Abra o projeto **Turma**
- Crie a classe **Disciplina**





# Exercício

- No projeto **Turma**, classe **Disciplina**



- Faça o `toString` imprimir nome, média e alunos acima da média

# Exercício

- No projeto **Turma**, classe Turma
  - Altere o código para usar a classe Disciplina
  - (O nome do curso deve ser substituído pelo nome da disciplina)

# Avançando no CadProd

- Abra o projeto CadProd.
- Crie uma classe chamada ListaDeProdutos
  - Construtor: número máximo de produtos
  - adiciona(Produto p): adiciona um produto à lista
    - Retorna true se ok, false caso contrário
  - listarProdutos: imprime a lista de produtos
- Faça com que o programa principal use essa lista de produtos



**PROJETO / ATIVIDADE**

# Projeto / Atividade

- Objetivo: criar cadastro de clientes simples
- Quem: Duplas especificadas
- A segunda entrega será 05/06
  - Listagem de clientes
  - Busca de clientes
  - Edição de clientes



# Projeto / Atividade

- Hoje: Implementar a classe ListaDeClientes
  - Nos mesmos moldes da CadProd