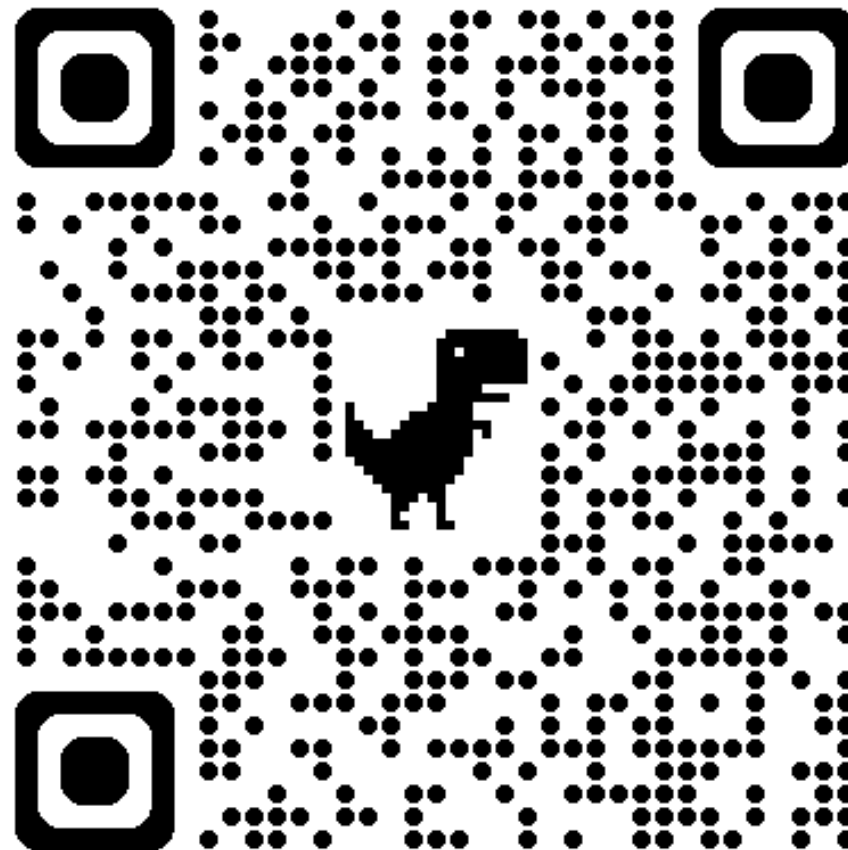


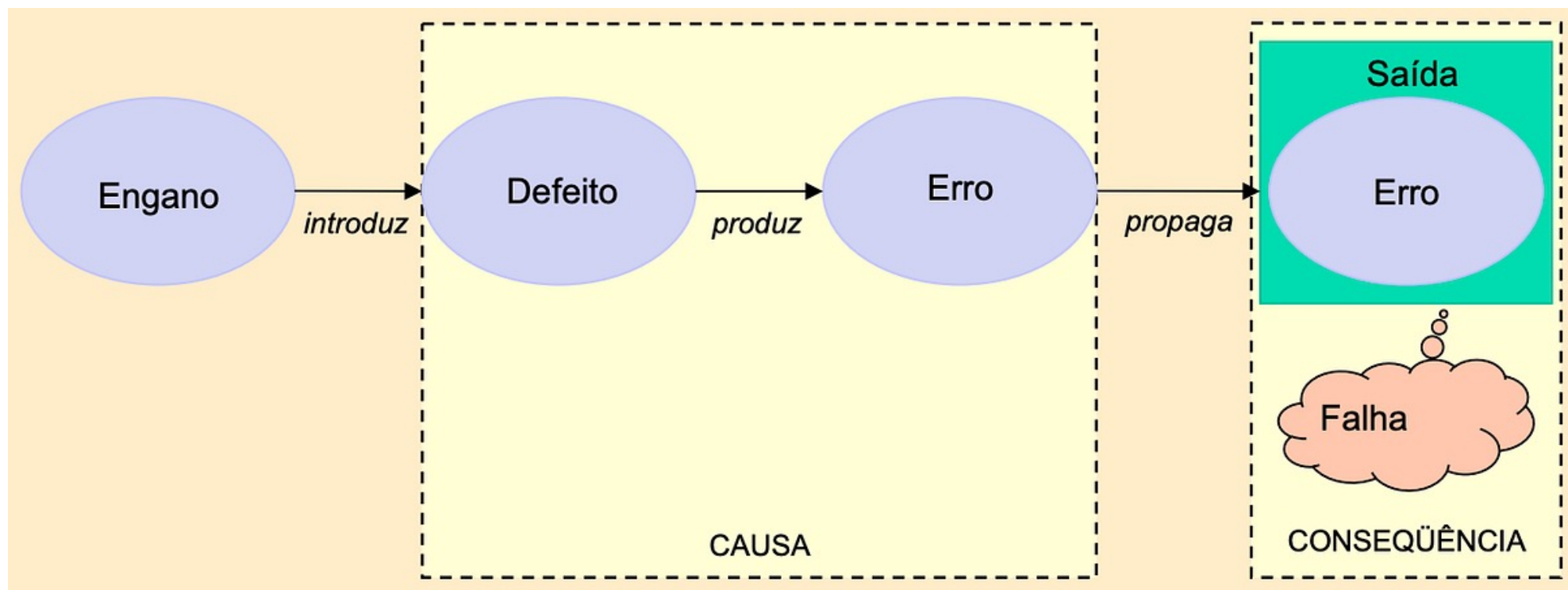
# PROGRAMAÇÃO ORIENTADA A OBJETOS

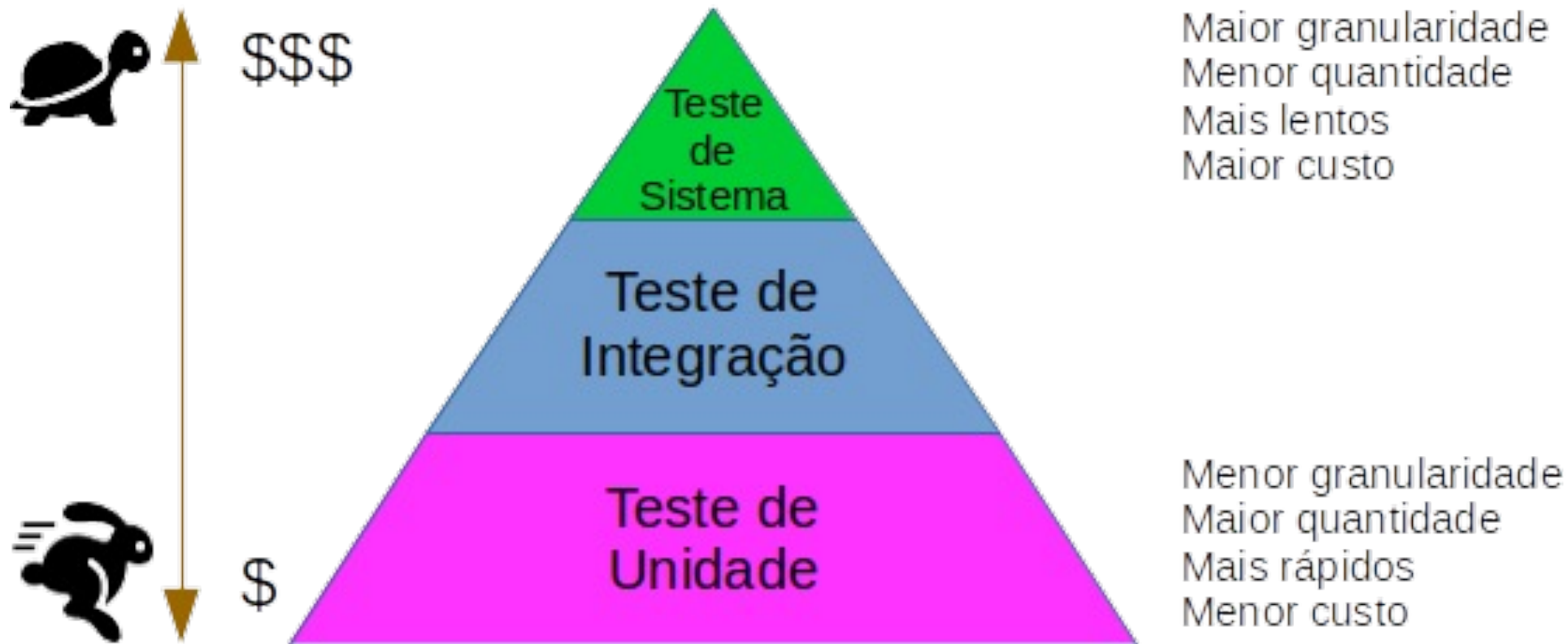
João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

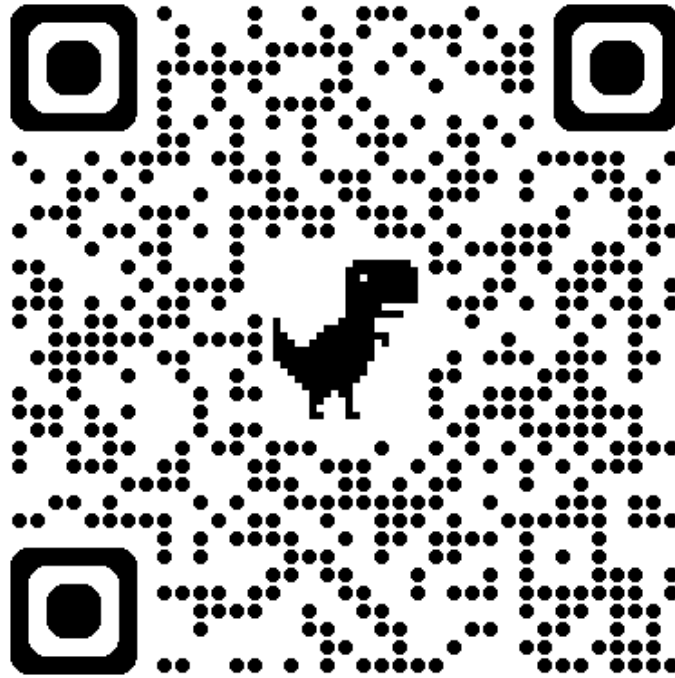
Unicesumar – Maringá



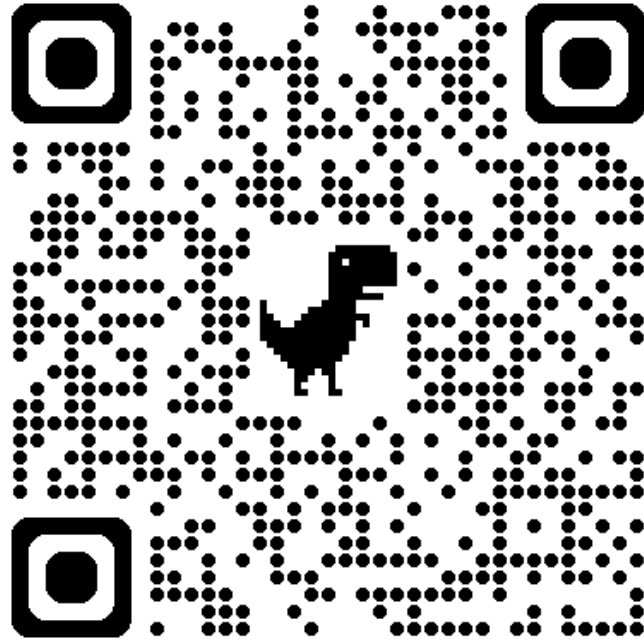




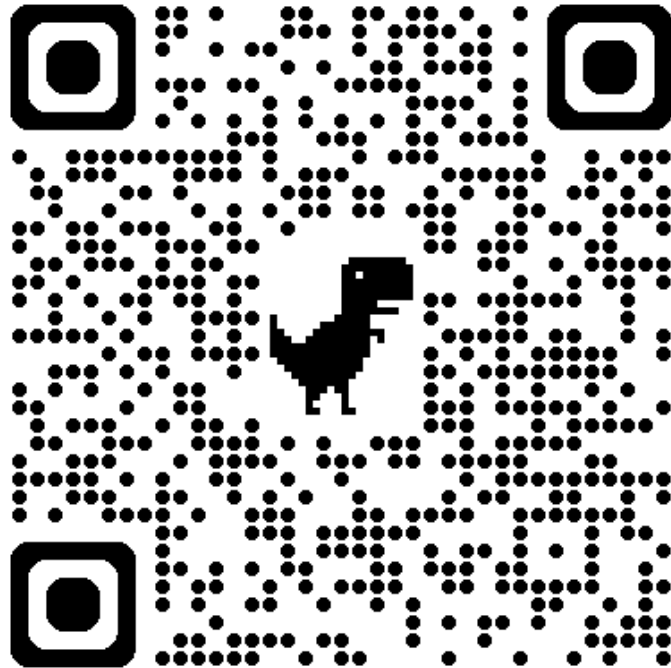
# IDE JAVA



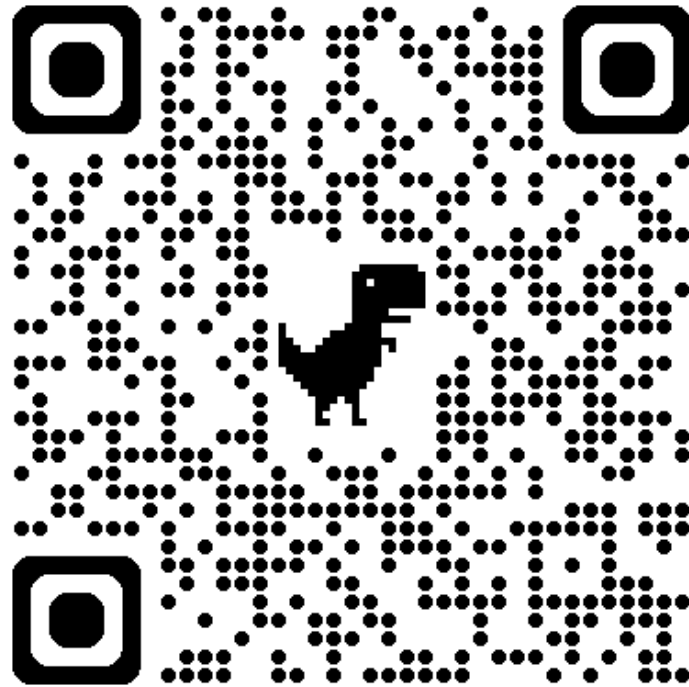
[https://www.jetbrains.com/idea/  
download/](https://www.jetbrains.com/idea/download/)



<https://www.eclipse.org/downloads/packages/installer>



<https://netbeans.apache.org/front/main/download/nb122/nb122/>



<https://www.oracle.com/br/java/technologies/downloads/>



# Relembrando

Orientação a objeto (OO) é um **paradigma** de programação que permite modelar o comportamento de **entidades** do mundo real como **objetos** em software.

# Relembrando

A programação orientada a objeto se concentra em objetos e suas relações com outros objetos.

Os objetos são entidades que possuem valores/pesos/dados e comportamentos.

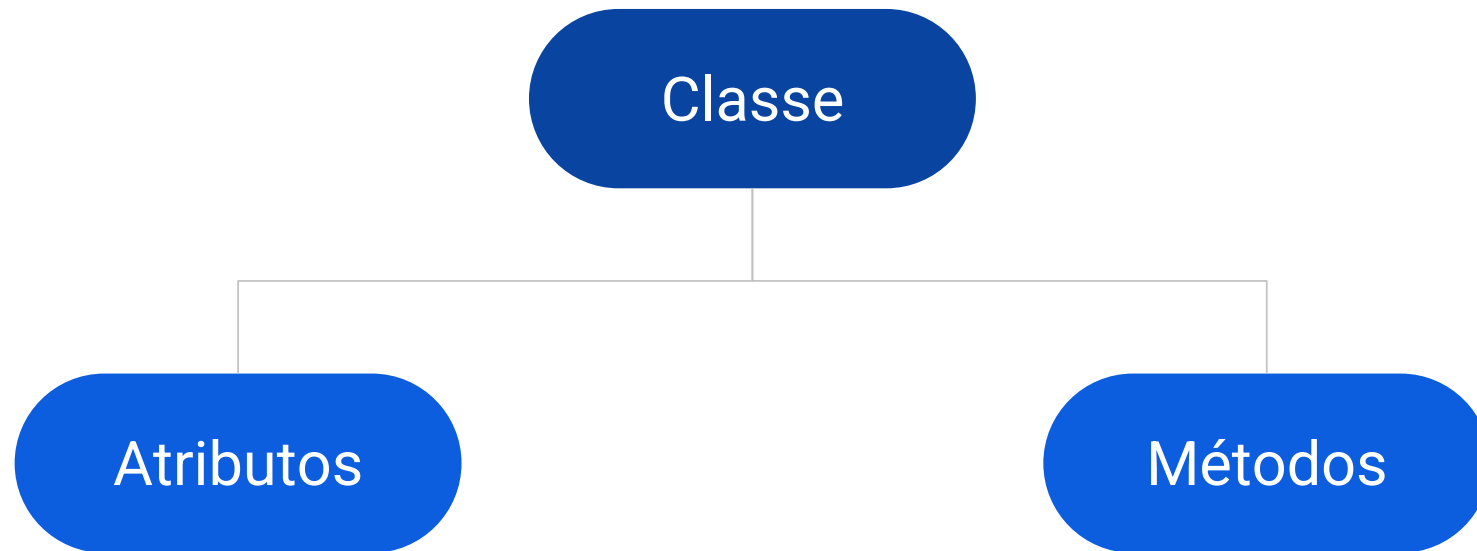


# Classe

Classe é um **modelo** para criar objetos.

A classe possui **atributos** que descrevem características do objeto e **métodos** que descrevem comportamentos.

# Classe



# Classe Pessoa

- Atributos
- Métodos

```
public class Pessoa {  
    private String nome;  
    private int idade;  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public int getIdade() {  
        return idade;  
    }  
  
    public void setIdade(int idade) {  
        this.idade = idade;  
    }  
}
```

# Atributos

Um **atributo** é uma variável que representa uma **característica** ou **propriedade** de um objeto

Cada objeto instanciado a partir de uma classe tem seus **próprios** valores para os atributos da classe

Eles **descrevem** as características dos objetos e podem ser usados para armazenar informações

# Classe Carro

## Modificadores de acesso

- private
- public

```
public class Carro {  
    private String modelo;  
    private int ano;  
    private double velocidade;  
  
    public Carro(String modelo, int ano) {  
        this.modelo = modelo;  
        this.ano = ano;  
        this.velocidade = 0.0;  
    }  
}
```

# Atributos

Um **atributo** é semelhante a uma **variável** em programação estruturada

Diferenças:

- Encapsulamento: Atributos em OO estão **encapsulados** dentro de objetos, enquanto as variáveis geralmente visíveis e acessíveis em todo o programa.
- Acesso: Atributos em OO podem ser acessados e modificados apenas por meio de **métodos** da classe, as variáveis em podem ser acessadas e modificadas em qualquer parte do programa.
- Tipos: Atributos em OO são **tipados**, o que significa que cada atributo tem um tipo de dados específico, enquanto as variáveis em programação estruturada podem ser tipadas ou não, dependendo da linguagem de programação.



# Classe Pessoa

## Atributos

- Nome (privado – acessado pelo método público)
- Idade (privado – acessado pelo método público)

## Métodos

- gets
- sets

```
public class Pessoa {  
    private String nome;  
    private int idade;  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public int getIdade() {  
        return idade;  
    }  
  
    public void setIdade(int idade) {  
        this.idade = idade;  
    }  
}
```

# Métodos

Os **métodos** representam o **comportamento** do objeto

São modificadores de **estado** do objetos

São modificadores dos **atributos** definidos na classe

São responsáveis por **acessar** e modificar os atributos da classe

# Classe Carro

## Métodos

- Gets (públicos)
- Sets (públicos)

```
public String getModelo() {  
    return modelo;  
}  
  
public int getAno() {  
    return ano;  
}  
  
public double getVelocidade() {  
    return velocidade;  
}
```

# Classe Carro

## Métodos

- acelerar (públicos)
- frear (públicos)

```
public void acelerar(double kmPorHora) {  
    this.velocidade += kmPorHora;  
}  
  
public void frear(double kmPorHora) {  
    this.velocidade -= kmPorHora;  
    if (this.velocidade < 0) {  
        this.velocidade = 0;  
    }  
}
```

# Atributos e Métodos

Os atributos e métodos oferecem mais segurança e organização ao seu código, já que estão encapsulados dentro de objetos e estão associados a objetos específicos

# Objetos

Você pode criar objetos a partir da classe **Carro** com o construtor

```
public class Carro {  
    private String modelo;  
    private String cor;  
    private int ano;  
    private int velocidade;  
  
    public Carro(String modelo, String cor, int ano) {  
        this.modelo = modelo;  
        this.cor = cor;  
        this.ano = ano;  
        this.velocidade = 0;  
    }  
}
```

# Objetos

Objeto: **meuCarro**

Definimos que ele será uma Ferrari da cor vermelho do ano de 2020

Chamamos seus métodos **acelerar** e **frear**

Como resultado o atributo velocidade deixa de ser 0

```
public class Main {  
    public static void main(String[] args) {  
        Carro meuCarro = new Carro("Ferrari", "Vermelho", 2020);  
        System.out.println("Modelo: " + meuCarro.getModelo());  
        System.out.println("Cor: " + meuCarro.getCor());  
        System.out.println("Ano: " + meuCarro.getAno());  
  
        meuCarro.acelerar(100);  
        System.out.println("Velocidade após acelerar: " + meuCarro.getVelocidade());  
  
        meuCarro.frear(50);  
        System.out.println("Velocidade após frear: " + meuCarro.getVelocidade());  
    }  
}
```

# Atividade

Meu problema tem as seguintes entidades

- Pessoa
- Carro
- Venda

Defina as classes, métodos e atributos que você achar necessário



# PROGRAMAÇÃO ORIENTADA A OBJETOS

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

Unicesumar – Maringá

