

# Classes, Atributos, Métodos e Objetos



Prof. Wanderson Timóteo

[www.wandersontimoteo.com.br](http://www.wandersontimoteo.com.br)

<https://github.com/professor-wanderson-timoteo/csharp>

# O que vamos aprender

Vamos analisar os conceitos de:

- Classes;
- Atributos;
- Métodos;
- Objetos;

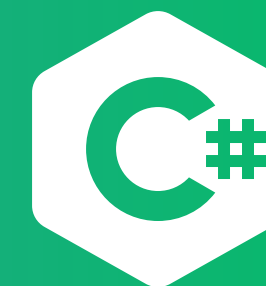




# O que vamos aprender

Vamos analisar o conceito de:

- **Classes;**
- ~~Atributos;~~
- ~~Métodos;~~
- ~~Objetos;~~



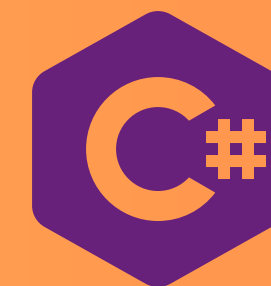
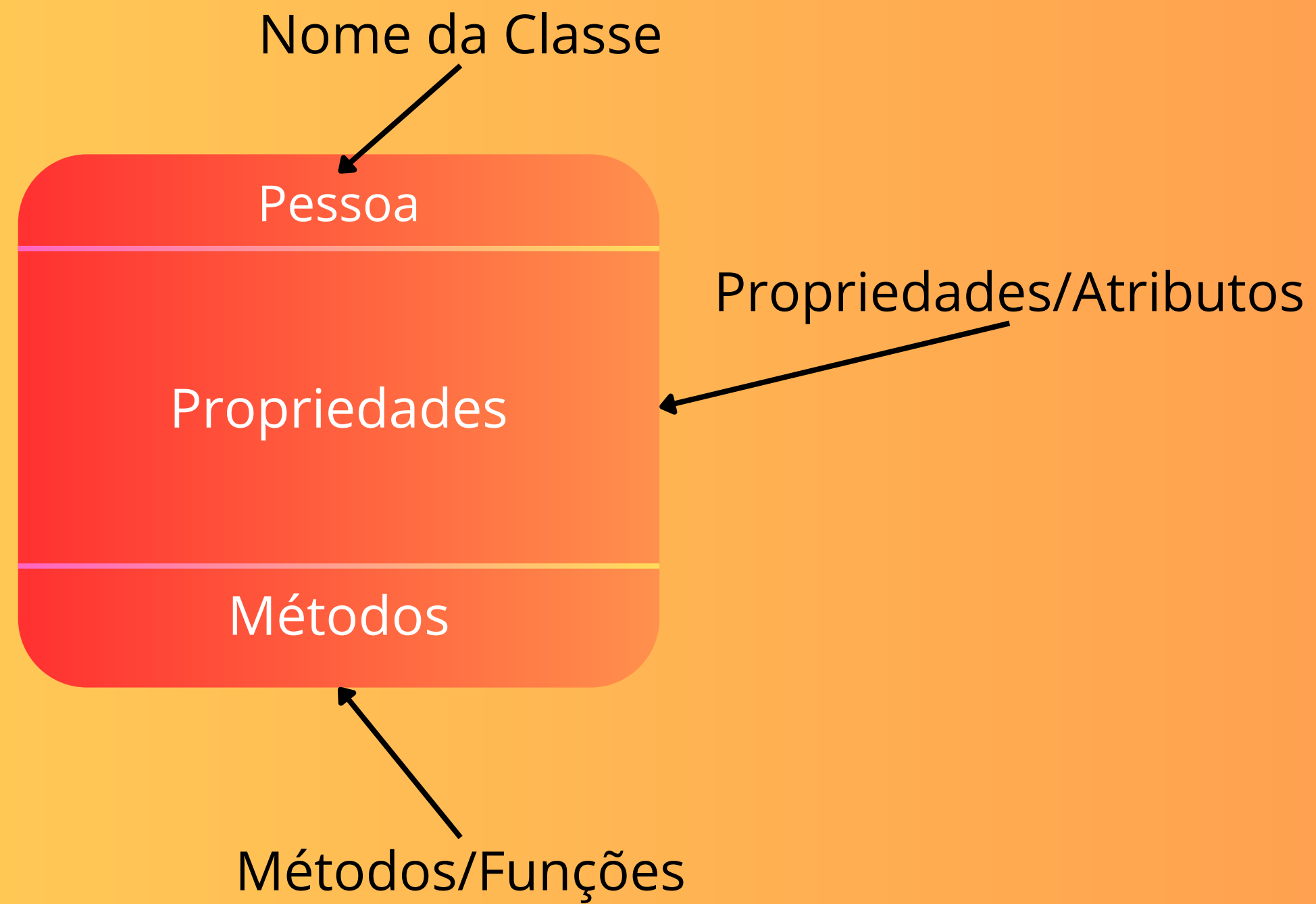
# Classes

## Classe

Uma classe é um molde para criar objetos. Ela define um conjunto de **propriedades** e **métodos** que os objetos criados a partir dessa classe irão possuir. Uma classe pode ser vista como uma definição abstrata das características e comportamentos que os objetos daquele tipo terão. No C#, uma classe é definida usando a palavra-chave `class`.



# Diagrama de Classes





# Classes

## Classes

O nome da classe **DEVE** ser o mesmo nome do arquivo '.cs' e a letra inicial **DEVE** ser maiúscula.

Exemplo: Pessoa.cs

```
Pessoa.cs
1 using System;
2
3 namespace Classe.Modelo
4 {
5
6     public class Pessoa
7     {
8         // Definir as propriedades e métodos da classe Pessoa
9     }
10 }
```

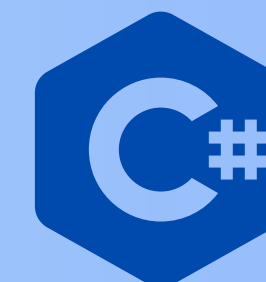




# O que vamos aprender

Vamos analisar o conceito de:

- ~~Classes;~~
- **Atributos;**
- ~~Métodos;~~
- ~~Objetos;~~

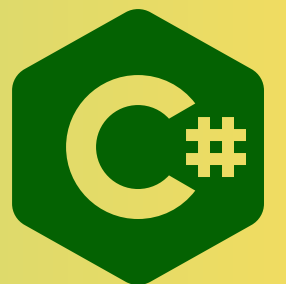




# Atributos

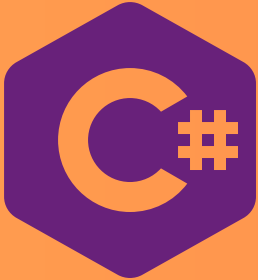
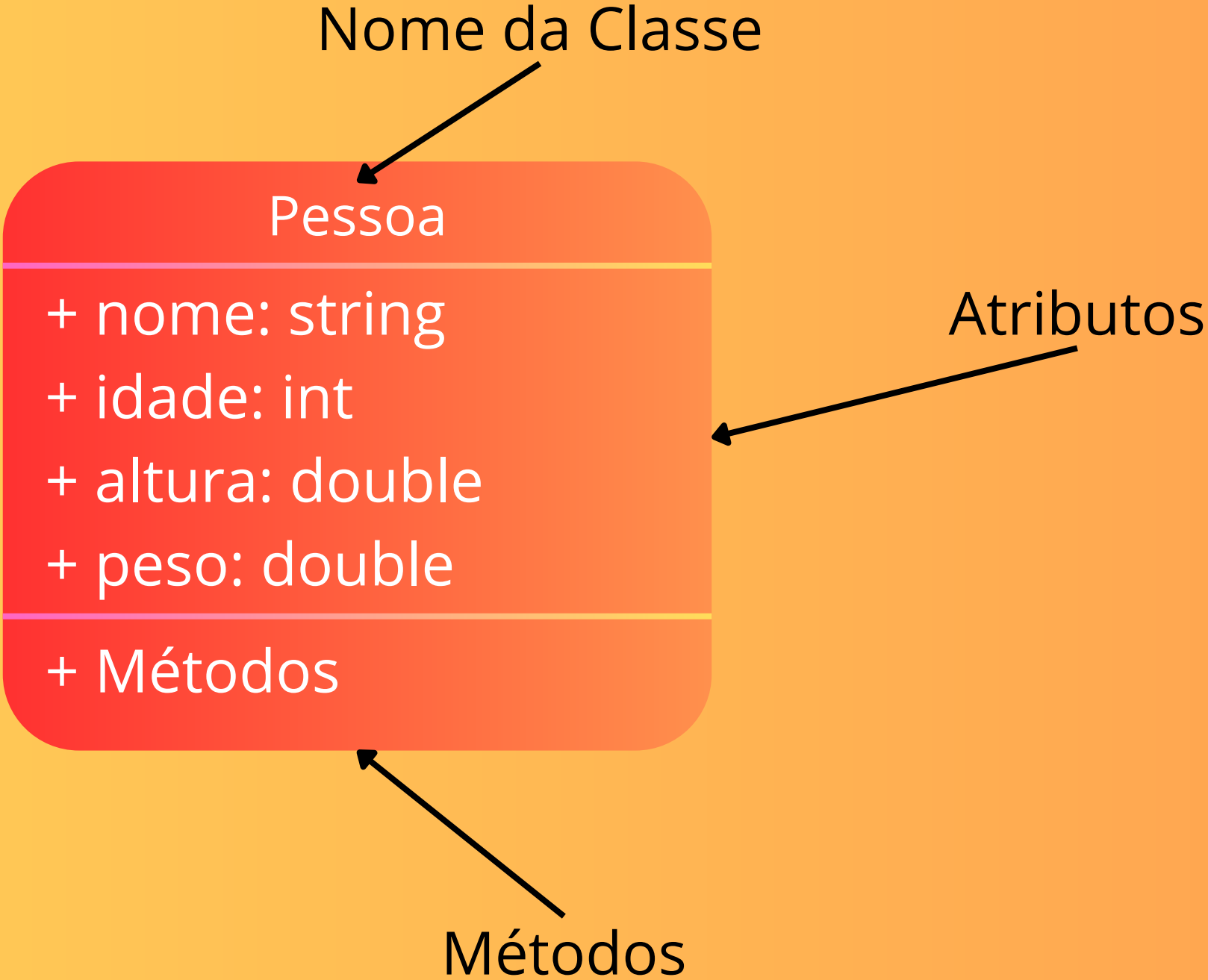
## Atributos

Atributos são variáveis dentro de uma classe que armazenam dados sobre os objetos criados a partir da classe. Eles representam as características ou propriedades do objeto. Atributos também são conhecidos como campos ou propriedades.





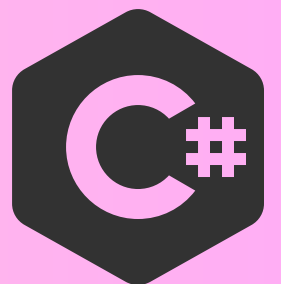
# Diagrama de Classes



# Atributos

Vamos adicionar os atributos “nome”, “idade”, “altura” e “peso” na classe Pessoa.

```
Pessoa.cs
1 public class Pessoa
2 {
3     // Atributos
4     string nome;
5     int idade;
6     double altura;
7     double peso;
8
9     // Construtor
10    public Pessoa(string nome, int idade, double altura, double peso)
11    {
12        this.nome = nome;
13        this.idade = idade;
14        this.altura = altura;
15        this.peso = peso;
16    }
17 }
```



# O que vamos aprender

Vamos analisar o conceito de:

- ~~Classes;~~
- ~~Atributos;~~
- **Métodos;**
- ~~Objetos;~~

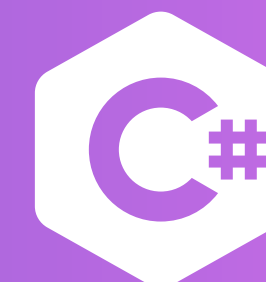




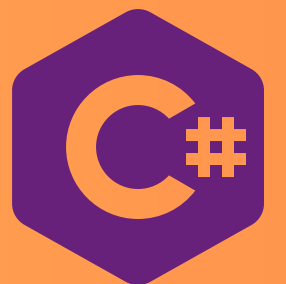
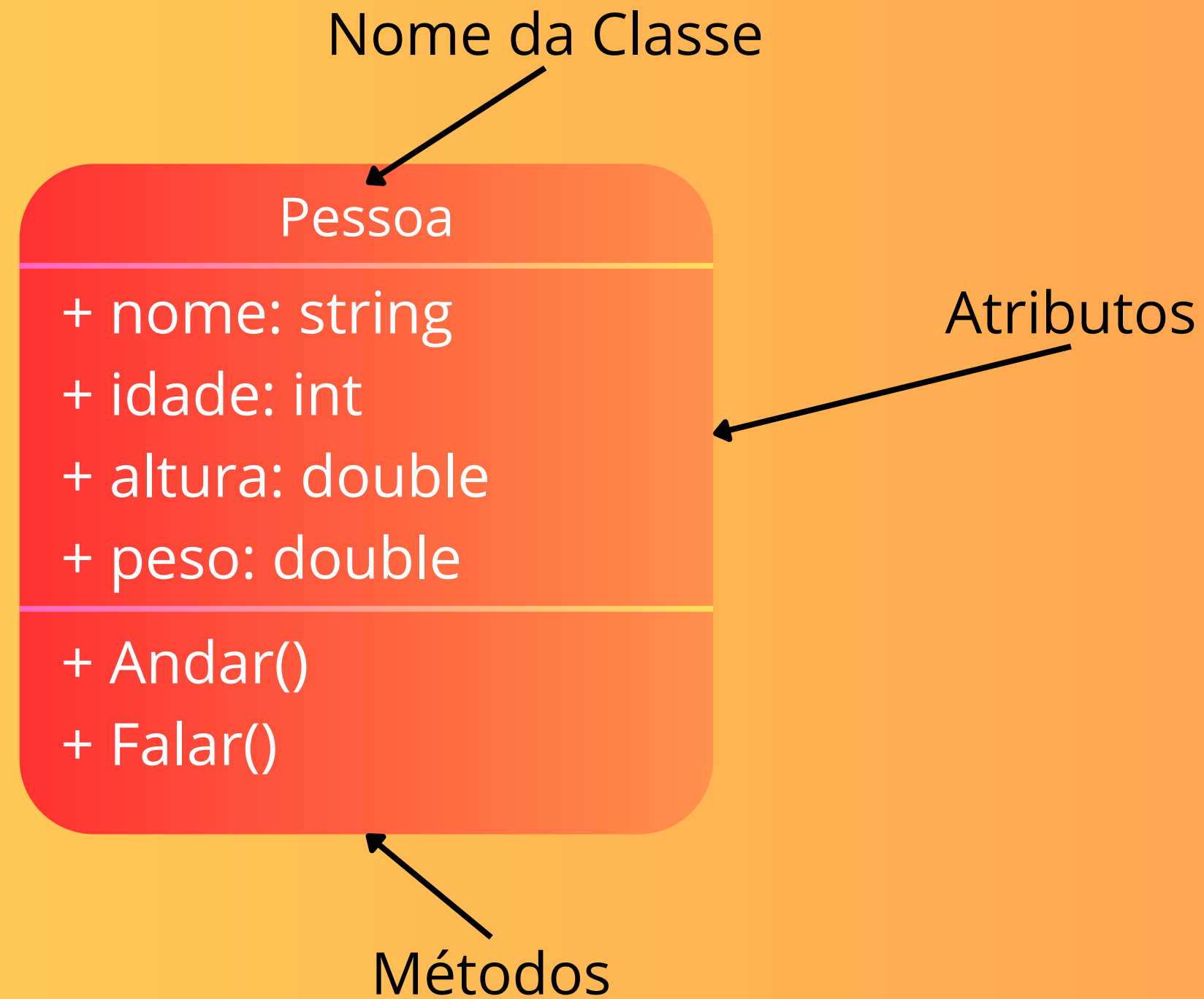
# Métodos

## Métodos

Métodos são funções definidas dentro de uma classe que descrevem os comportamentos dos objetos da classe. Eles podem manipular os atributos da classe e realizar operações relacionadas ao objeto. Um método é chamado por um objeto para executar uma tarefa específica.



# Diagrama de Classes



# Métodos

## Métodos

Vamos adicionar os métodos “Andar” e “Falar” na classe Pessoa. Por convenção a letra inicial dos métodos **DEVE** ser maiúscula.



```
Pessoa.cs
1  using System;
2
3  namespace Classe.Modelo
4  {
5      public class Pessoa
6      {
7          // Atributos
8
9          // Construtor
10
11         // Métodos Andar e Falar
12         public void Andar()
13         {
14             Console.WriteLine($"{nome} está andando.");
15         }
16         public void Falar()
17         {
18             Console.WriteLine($"{nome} Vou ficar rico programando em C#!");
19         }
20     }
21 }
```



# O que vamos aprender

Vamos analisar o conceito de:

- ~~Classes;~~
- ~~Atributos;~~
- ~~Métodos;~~
- **Objetos;**





# Objetos

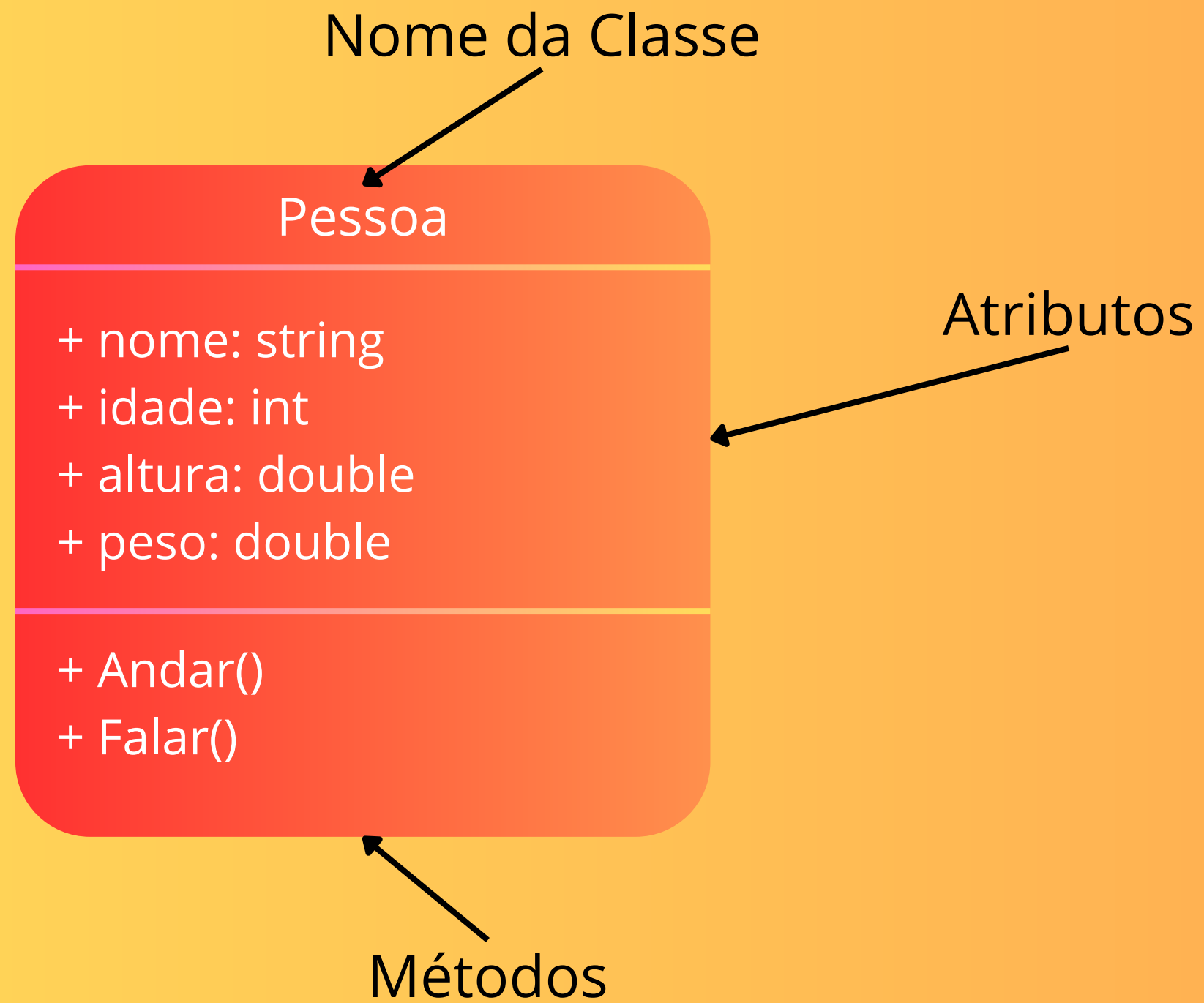
## Objetos

Um objeto é uma instância de uma classe. Quando você cria um objeto, está instanciando uma classe e alocando memória para ela. Objetos representam entidades concretas ou abstratas que possuem estados e comportamentos definidos pela classe.





# Diagrama de Classes





# Objetos

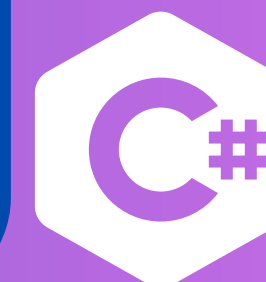
## Objetos

Vamos criar a instância/objeto a partir da classe Pessoa.

- **Classe:** Molde que define atributos e métodos.
- **Atributos:** Características ou propriedades da classe.
- **Métodos:** Funções que definem comportamentos da classe.
- **Objetos:** Instâncias concretas de uma classe.



```
Program.cs
1 using Classe.Modelo
2
3 // Criando um objeto a partir da classe Pessoa
4 Pessoa obj = new Pessoa("Wanderson", 25, 1.90, 60.9);
5 obj.Andar();
6 obj.Falar();
7
8
9
10
```



# Exercício

Crie um novo projeto chamado exercicio-classe e implemente o Diagrama de Classes abaixo:

