

## Etapa 2

# Herança e Polimorfismo

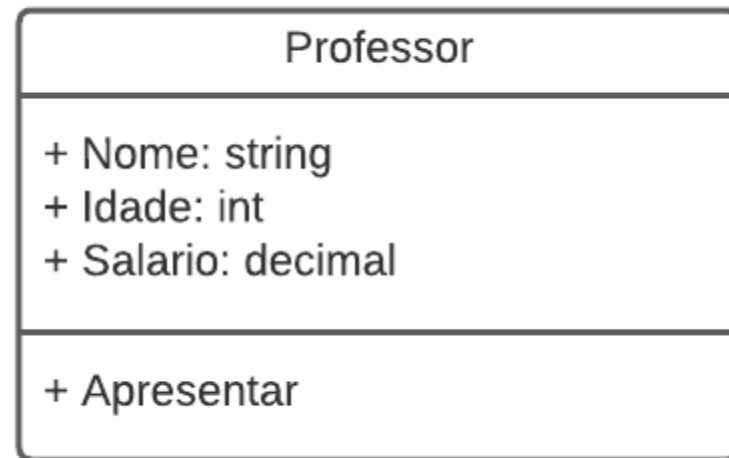
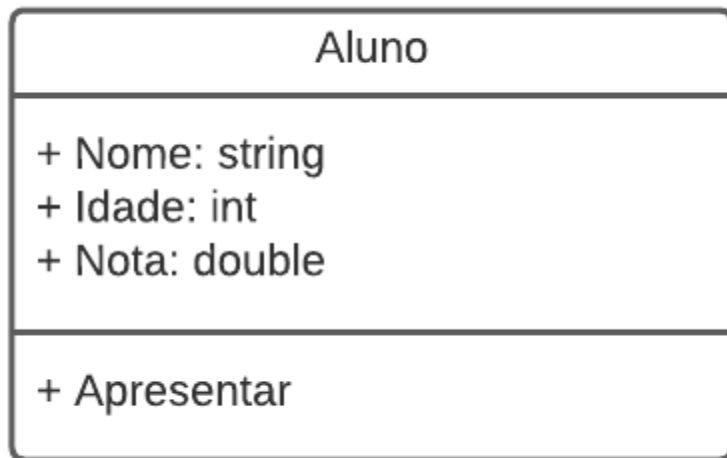
// Programação Orientada a Objetos

# Herança

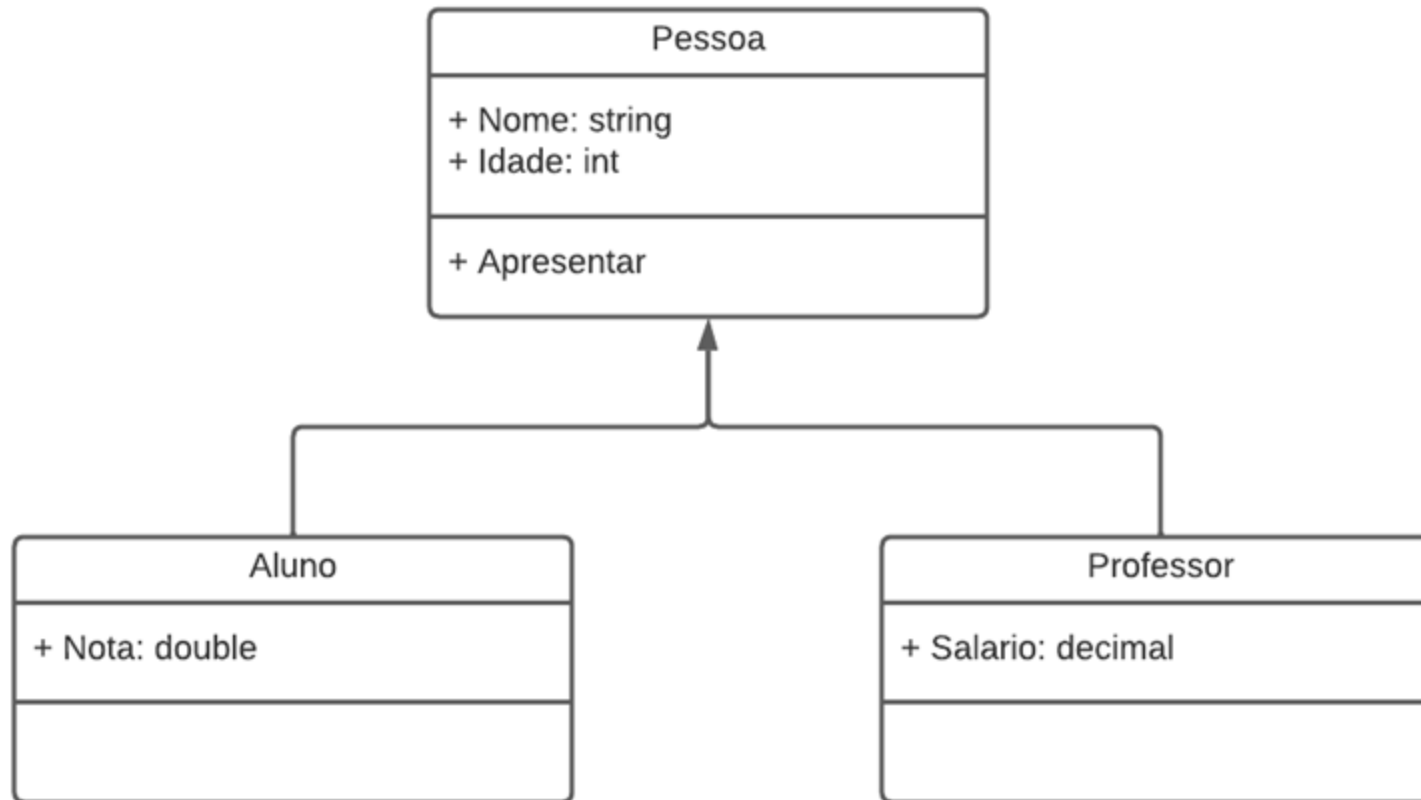
A herança nos permite reutilizar atributos, métodos e comportamentos de uma classe em outras classes.

Serve para agrupar objetos que são do mesmo tipo, porém com características diferentes.

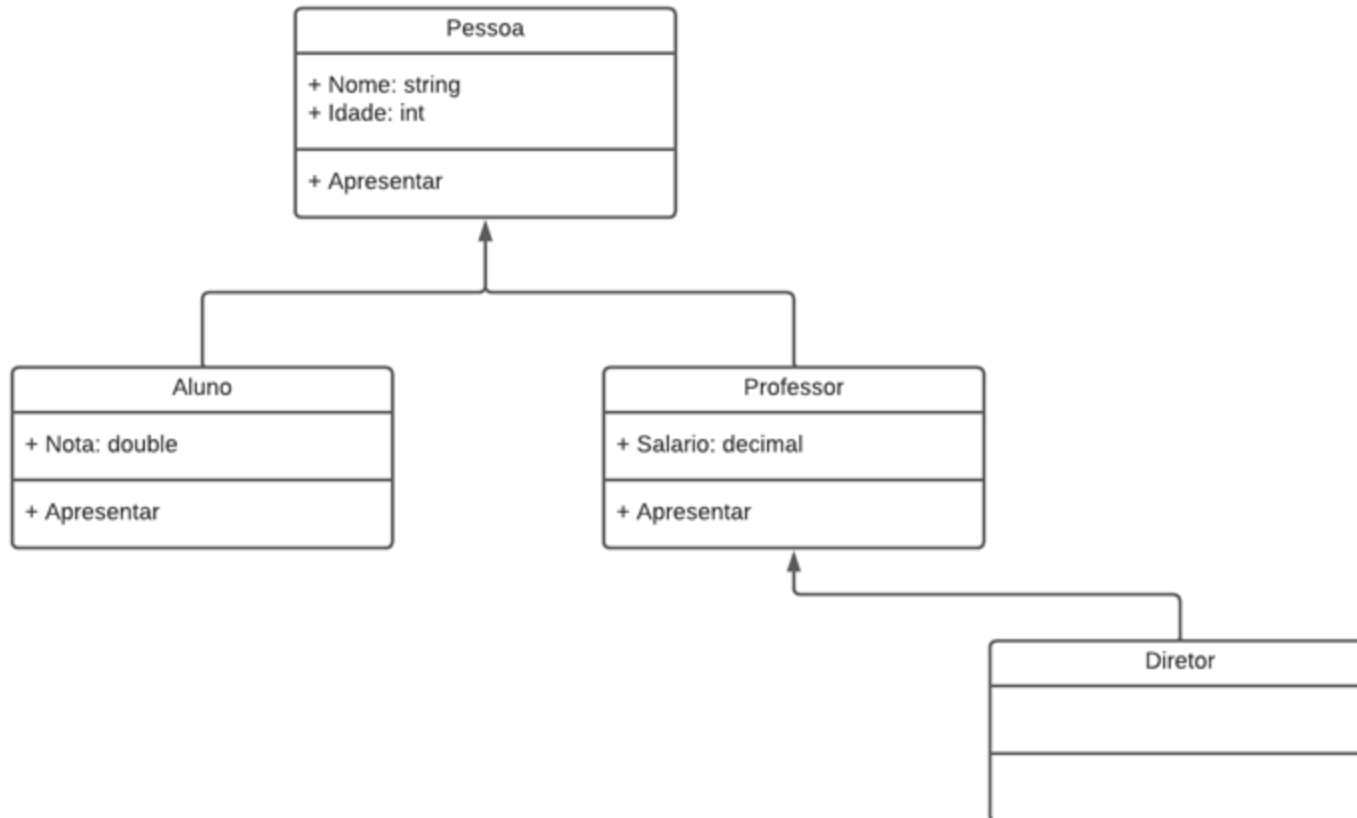
# Herança



# Herança



# Herança

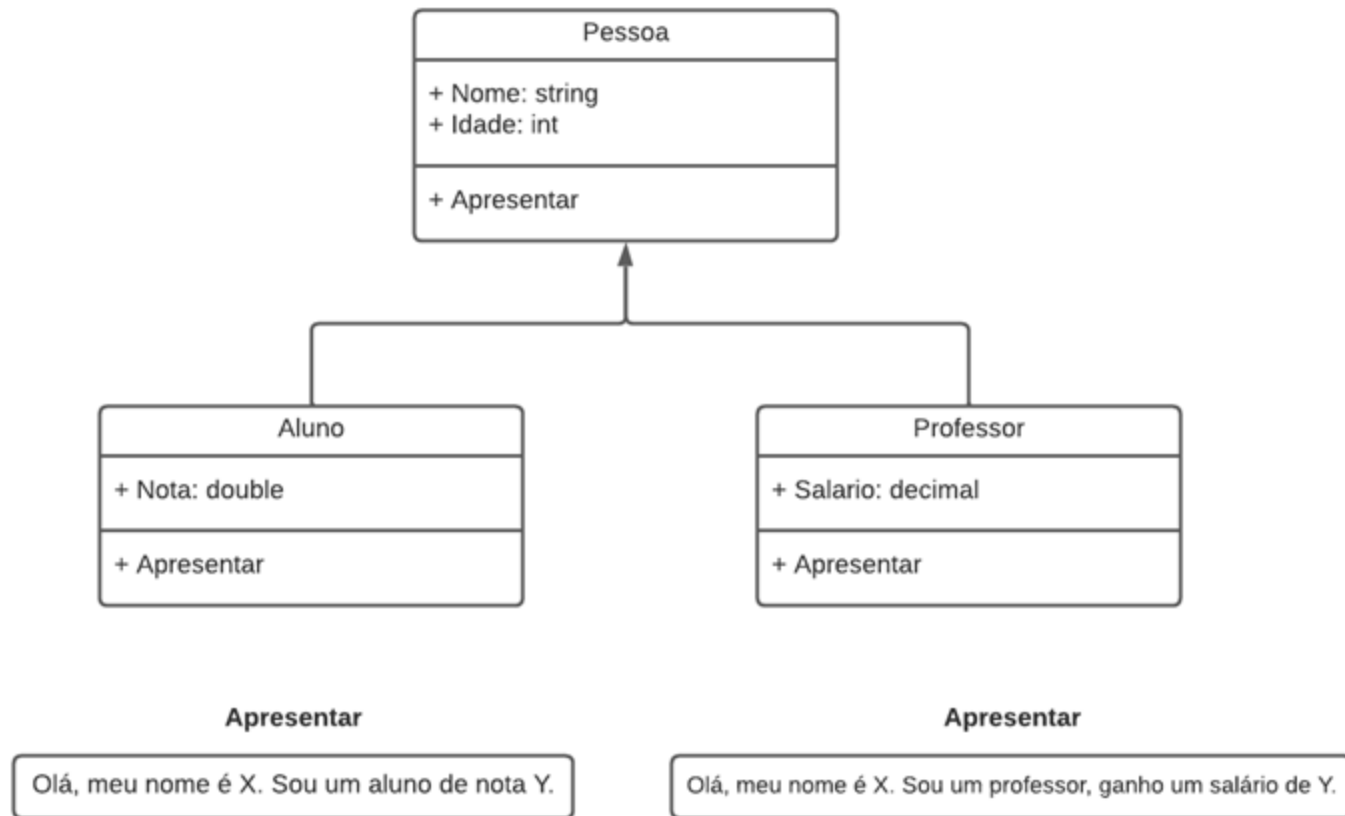


# Polimorfismo

O polimorfismo vem do grego e significa “muitas formas”.

Com o polimorfismo, podemos sobrescrever métodos das classes filhas para que se comportem de maneira diferente e ter sua própria implementação.

# Polimorfismo



# Polimorfismo

Polimorfismo em tempo de compilação (Overload/Early Binding)

```
public class Calculadora
{
    public int Somar(int num1, int num2)
    {
        return num1 + num2;
    }

    public int Somar(int num1, int num2, int num3)
    {
        return num1 + num2 + num3;
    }
}
```



# Polimorfismo

Polimorfismo em tempo de execução (Override/Late Binding)

```
public class Pessoa
{
    public string Nome { get; set; }
    public int Idade { get; set; }
    public string Email { get; set; }

    public virtual void Apresentar()
    {
        Console.WriteLine($"Olá, meu nome é {Nome} e tenho {Idade} anos!");
    }
}

public class Aluno : Pessoa
{
    public double Nota { get; set; }

    public override void Apresentar()
    {
        Console.WriteLine($"Olá, meu nome é {Nome}, tenho {Idade} anos, e sou um aluno nota {Nota}");
    }
}
```

# Percurso

## **Etapa 1**

**~~Introdução POO, Abstração e Encapsulamento~~**

## **Etapa 2**

**~~Herança e Polimorfismo~~**

## **Etapa 3**

**Classes Abstratas e Interfaces**