

# C206 - POO

Arthur Openheimer

# Informações Gerais

- Atendimento → Terça-feira 17:30-19:10, prédio 1, sala 19
- Email → arthur.openheimer@ges.inatel.br
- Github → <https://github.com/ArthurOpenheimer/C206-Monitoria>

# Classes abstratas

Classes Abstratas são classes modelos para classes filhas, essas classes não podem ser instanciadas mas podem ser usadas como referência

Podemos também criar métodos abstratos, que não tem corpo, que serão implementadas pelas classes filhas

```
public abstract class Usuario {
```



Estrutura

# Vantagens x Desvantagens

Classes abstratas oferecem um modelo genérico, que nos permite utilizar herança e, conseqüentemente, suas vantagens. Além disso, nos fornece uma camada de segurança, não permitindo a existência de um objeto genérico da classe mãe

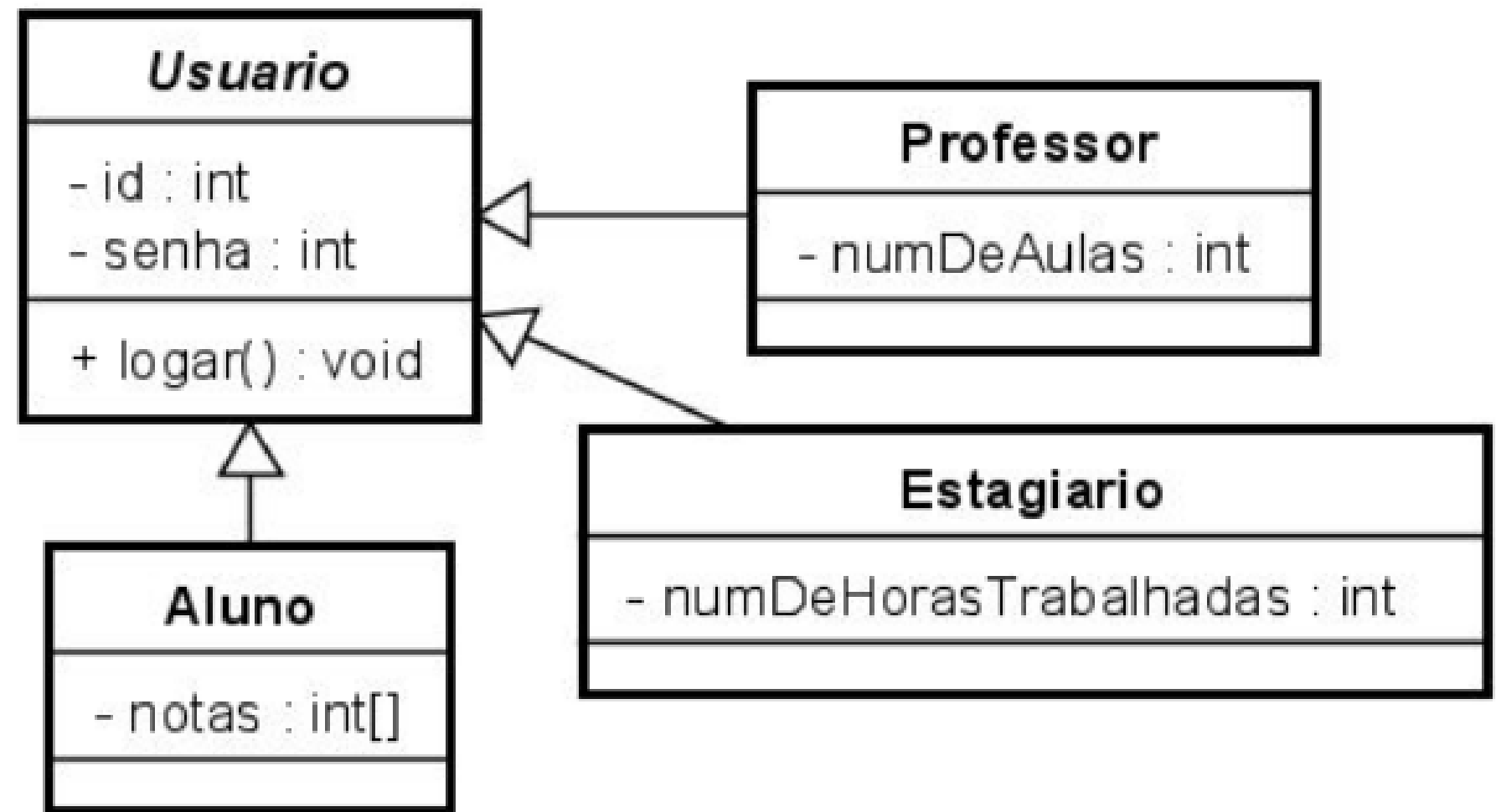
Já métodos abstratos nos garantem que cada classe filha terá sua implementação específica

No Java, uma classe só pode herdar uma única outra classe, portanto, se uma classe tiver uma classe abstrata como classe mãe, ela não pode herdar outra classe

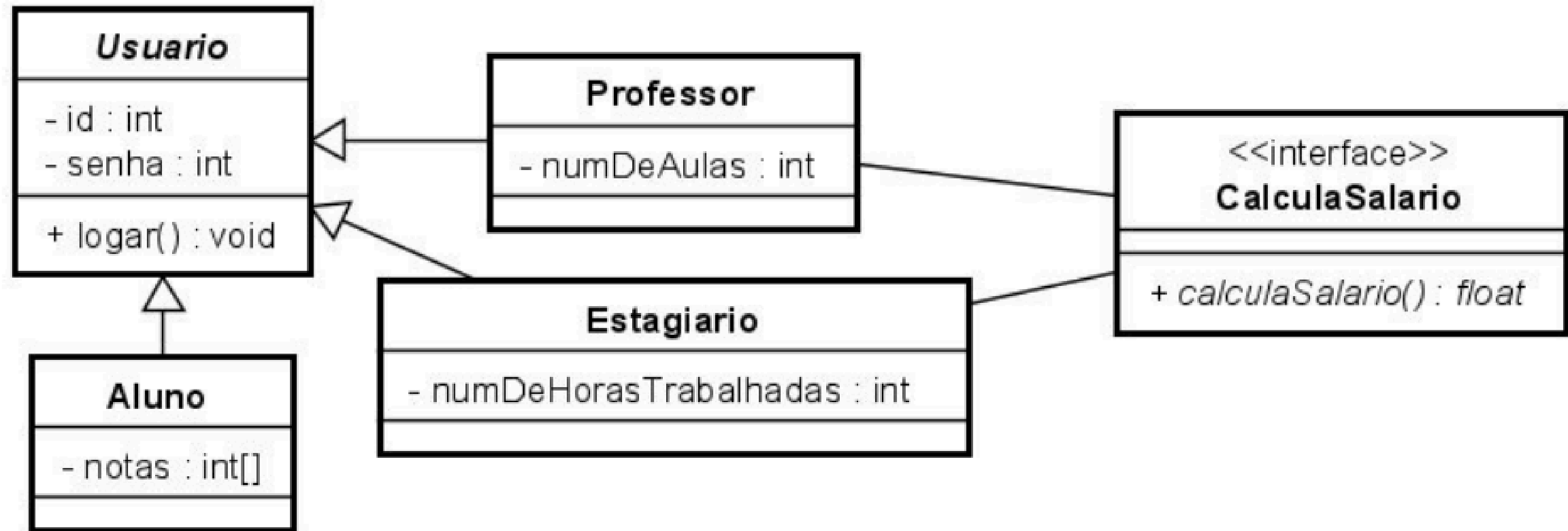
# Problema

Imagine que temos uma classe abstrata *Usuario* (nome em itálico na UML) e as classes Professor, Estagiario e Aluno são classes filhas

Professor e Estagiario precisam de um método `calcularSalario()`, implementar o mesmo método nas duas classes acarretaria em um código repetitivo, então onde devemos implementá-lo?



# Solução



# Interfaces

Interfaces definem um conjunto de métodos que as classes devem seguir. Ao usar uma interface, garantimos que diferentes classes terão a mesma estrutura básica, mesmo que o funcionamento de cada uma seja diferente

Toda interface é abstrata e podemos implementar N interfaces

Classes que implementarem uma interface devem implementar todos os métodos da interface, ou então se transformar em uma classe abstrata

# Implementando uma interface

```
public interface CalculaSalario {  
    public abstract float calcularSalario();  
}
```

Estrutura da interface

Observe que o método é apenas declarado

```
public class Professor extends Usuario implements CalculaSalario {  
    private int numDeAulas;  
  
    @Override  
    public float calcularSalario() {  
        return numDeAulas * 50;  
    }  
}
```

Classe implementando Interface



# Exercício

Crie as instâncias necessárias e chame todos os seus métodos

