

# Exercício 03 - Implementação Funcionario

Neste exercício, você irá criar um novo objeto chamado **Funcionario**, seguindo o modelo do nosso projeto que já possui classes de **Controller**, **Core** e **objetos**.

## Requisitos do Objeto Funcionario

A classe **Funcionario** deve ser **abstrata** e conter os seguintes atributos:

- **nome** (String): O nome do funcionário.
- **salario** (double): O salário do funcionário.

## Métodos:

- O método **abstrato calcularBonus()**: Método sem implementação que deve ser sobrescrito pelas classes filhas.

## Heranças:

### Classe Gerente

Crie a classe **Gerente**, que herda de **Funcionario**, e adicione o seguinte comportamento:

- Sobrescreva o método **calcularBonus()**: Ele deve retornar o valor do salário multiplicado por 0.20, indicando um bônus de 20% para gerentes.

### Classe Desenvolvedor

Crie a classe **Desenvolvedor**, que herda de **Funcionario**, e adicione o seguinte comportamento:

- Sobrescreva o método **calcularBonus()**: Ele deve retornar o valor do salário multiplicado por 0.10, indicando um bônus de 10% para desenvolvedores.

## Validações:

- O atributo **nome** deve ser uma string não vazia. Caso contrário, deve-se lançar uma exceção informando que o nome do funcionário é inválido.
- O atributo **salario** deve ser maior que zero. Se o valor for menor ou igual a zero, deve-se lançar uma exceção informando que o salário deve ser positivo.

## Exemplo de Dados:

Para exemplificar, considere os seguintes valores:

- **nome:** "João"
- **salario:** 5000.00

### Entrega:

- Implemente as classes conforme os requisitos acima.
- Demonstre a funcionalidade criando objetos das classes **Gerente** e **Desenvolvedor** em um método principal e testando os métodos de cálculo de bônus e exibição de detalhes.