



Classes Abstratas em UML



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Temas

1

**UML da Classe
Abstrata**

2

**Exemplos e boas
práticas**

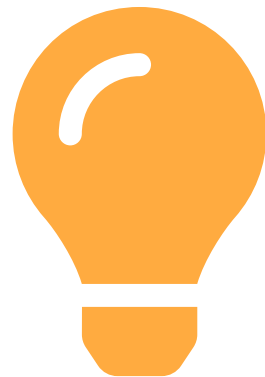


1

UML da Classe Abstrata



Vamos ver **como é representado uma classe abstrata** em nossos diagramas UML.





Classes Abstratas

Classes abstratas são aquelas que por si mesmas não podem ser identificadas com algo "**concreto**" (elas não existem como tal no mundo real), mas têm certas características que são comuns em outras classes que herdarão disso.

Essas classes abstratas nos permitem declarar **métodos, mas eles não são implementados**, ou seja, não fazem nada na classe abstrata, e esses métodos que também chamaremos de abstratos, forçarão as subclasses a sobrescrevê-los para dar-lhes uma implementação.

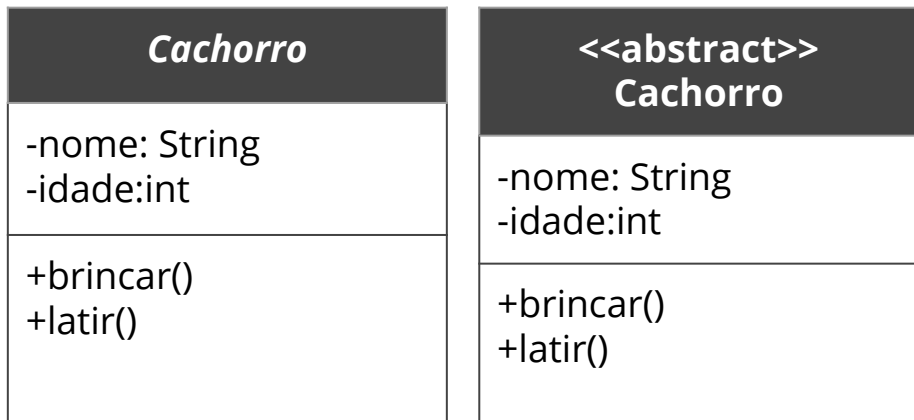


Não vamos esquecer que não poderemos instanciar objetos de uma classe abstrata.



Classe abstrata no diagrama UML

Representaremos as classes abstratas no diagrama UML colocando nome da classe em **itálico** ou deixando explícito acima do nome classe que é **<<abstract>>**, podemos escolher qualquer um dos dois.





Vamos especificar os **métodos abstratos** nos diagramas UML conforme mostrado abaixo, o método latir agora será um método abstrato e nos diagramas UML acrescentamos a palavra abstract para identificá-lo como tal.



Ao tornar o método latir abstrato, indica que ele **não está implementado** na classe Cachorro e deve ser implementado por todas as classes que herdam de Cachorro.

<<abstract>>
Cachorro

-nome: String

-idade:int

+brincar()

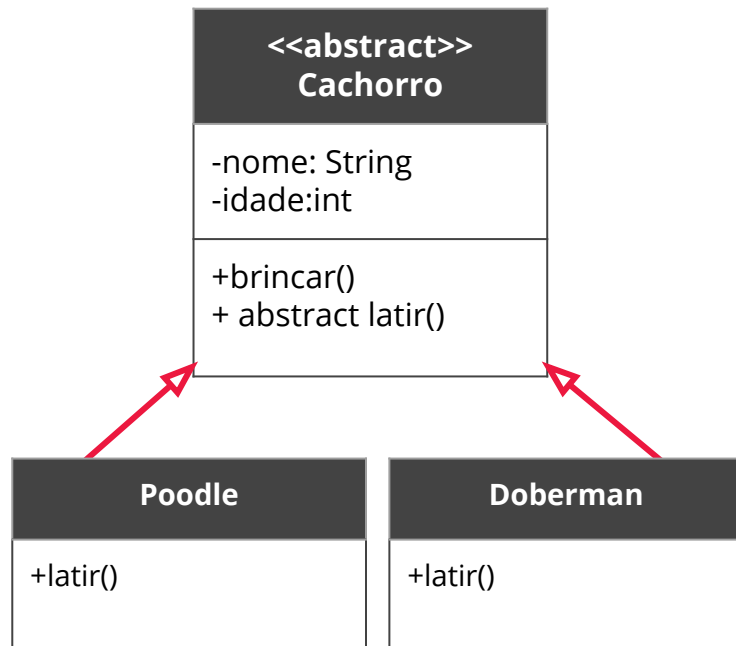
+ abstract latir()



No exemplo abaixo, colocar a classe abstrata Cachorro com um método abstrato **força** as classes Poodle e Doberman a **sobrescrever** esse método implementando-o.



Todo cachorro deve latir, mas apenas cada cachorro específico sabe "como" latir.





Dessa forma, uma classe abstrata pode ter atributos e métodos que serão herdados pelas subclasses, mas também pode conter **métodos abstratos que atuam como um contrato**, forçando essas subclasses a implementar esses métodos.



Embora uma classe abstrata possa ter um ou mais métodos abstratos, não é necessário tê-los.



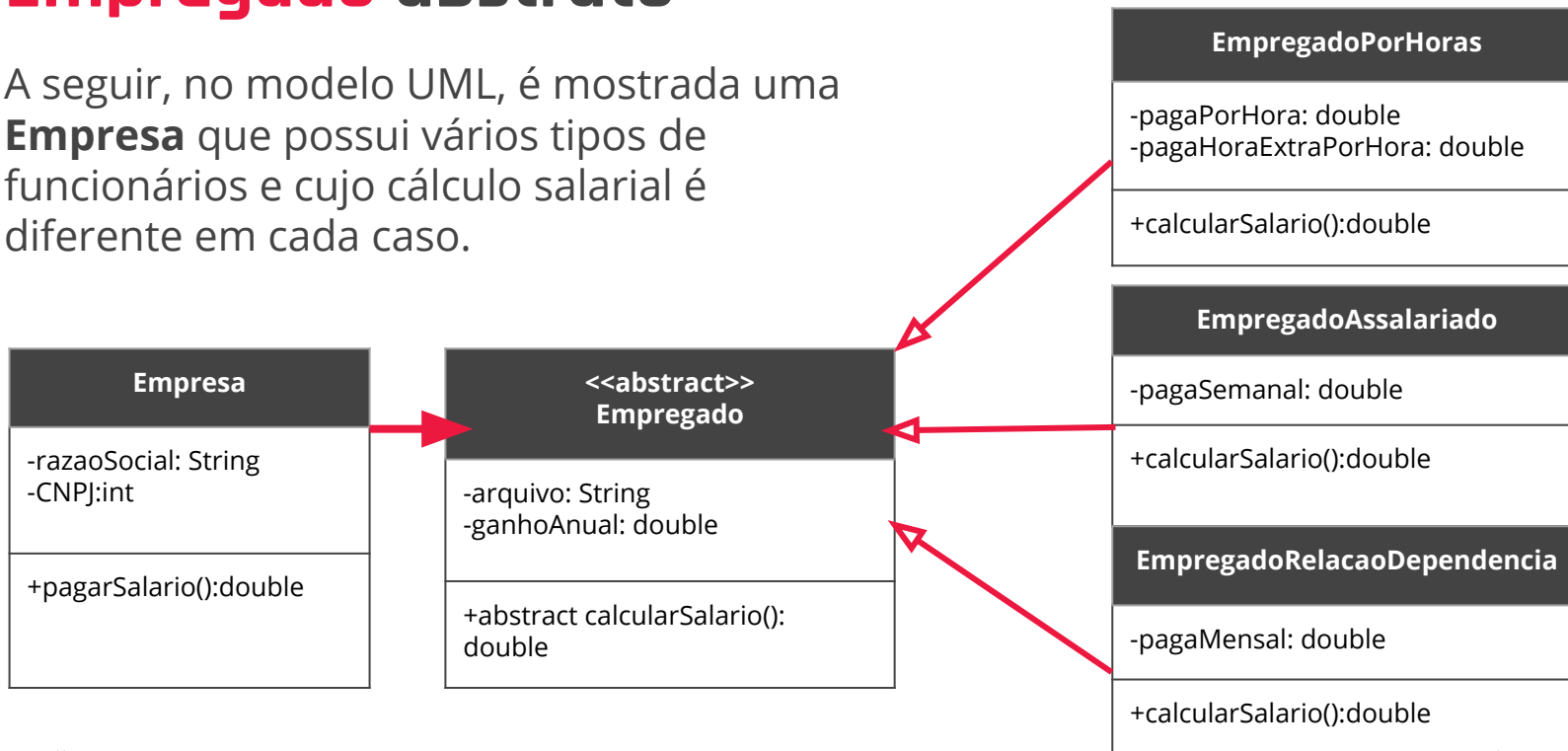
2

Exemplos e boas práticas



Empregado abstrato

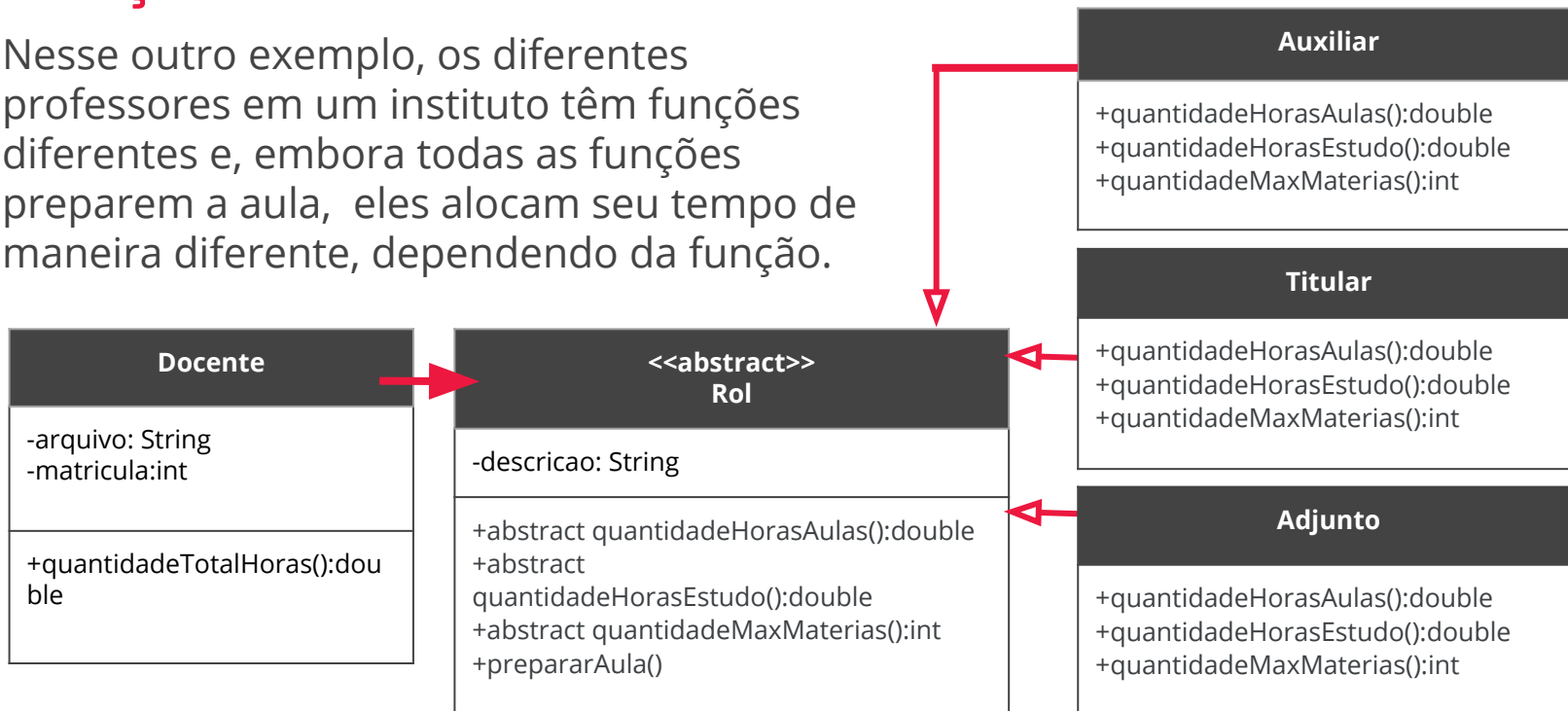
A seguir, no modelo UML, é mostrada uma **Empresa** que possui vários tipos de funcionários e cujo cálculo salarial é diferente em cada caso.





Função abstrata

Nesse outro exemplo, os diferentes professores em um instituto têm funções diferentes e, embora todas as funções preparem a aula, eles alocam seu tempo de maneira diferente, dependendo da função.



DigitalHouse>
Coding School