



Iniciando com Ruby e Orientação a Objetos



4.

Classes






O que é uma Classe?




"Uma classe é um gabarito para a definição de objetos."







Através da definição de uma classe,
descreve-se que **propriedades/atributos**
e **métodos/ações** o objeto terá.



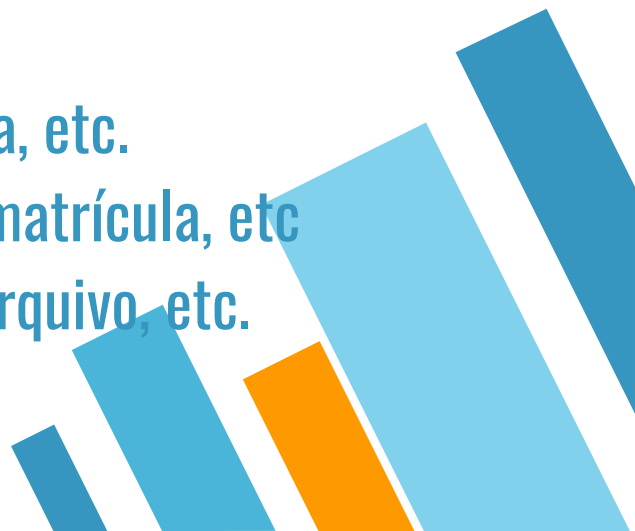


Em resumo...

- Uma classe representa um conjunto de objetos com características afins.
 - Uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos.
- 

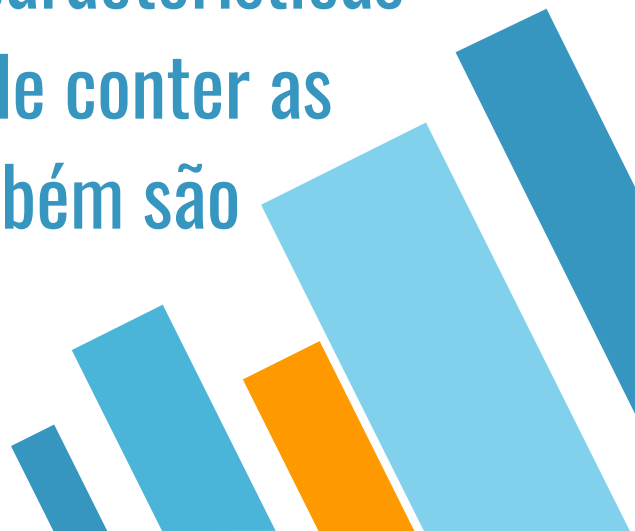


É possível especificar classes para entidades físicas, conceituais ou de software. Ex:

- **Entidade Física:** caminhão, carro, bicicleta, etc.
 - **Entidade Conceitual:** processo químico, matrícula, etc
 - **Entidade de Software:** lista encadeada, arquivo, etc.
- 



Exemplo:

- A classe "Lâmpada", pode conter os atributos "potência" e "voltagem", que são características comuns à todas as lâmpadas, e pode conter as ações "ligar" e "desligar", que também são características comuns.
- 



**Como representar uma Classe
graficamente?**



- Em notação UML uma classe pode ser representada assim:


Nome da Classe
Atributos da Classe
Métodos da Classe



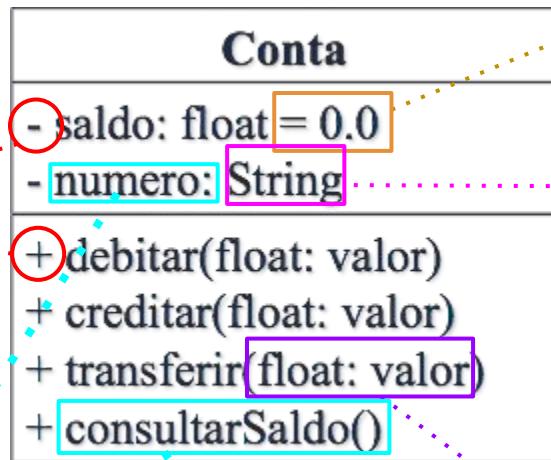
Conta
- saldo: float - numero: String
+ debitar() + creditar() + transferir() + consultarSaldo()

- 
- Ou ainda, com mais riqueza de detalhes
- detalhes:

Conta
- saldo: float = 0.0 - numero: String
+ debitar(float: valor) + creditar(float: valor) + transferir(float: valor) + consultarSaldo()



Visibilidade



Valor inicial do atributo

Tipo do atributo

Parâmetros do método

Nome do
atributo/método



Visibilidade

- Para entendermos o que é a visibilidade, falaremos sobre o conceito de Encapsulamento.
- 