

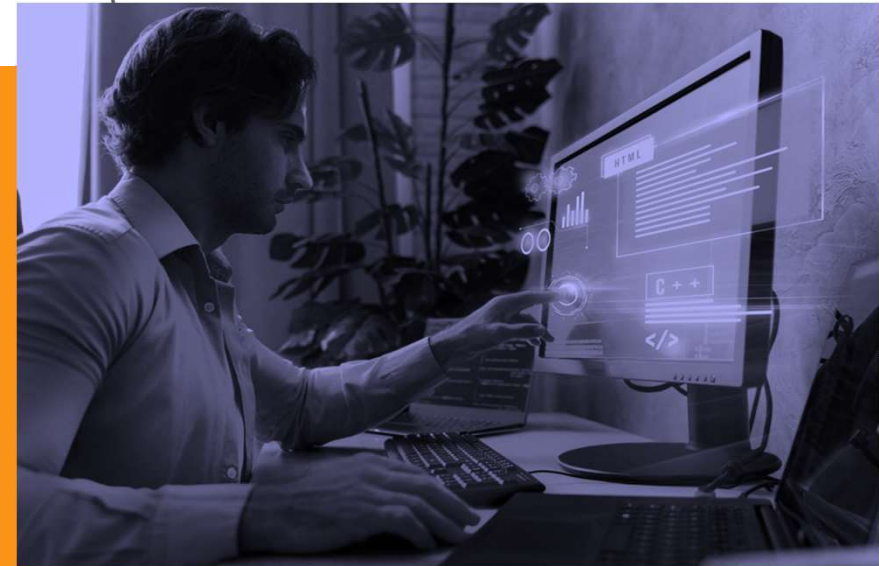
# Análise e Modelagem de Sistemas

Ma. Vanessa Matias Leite

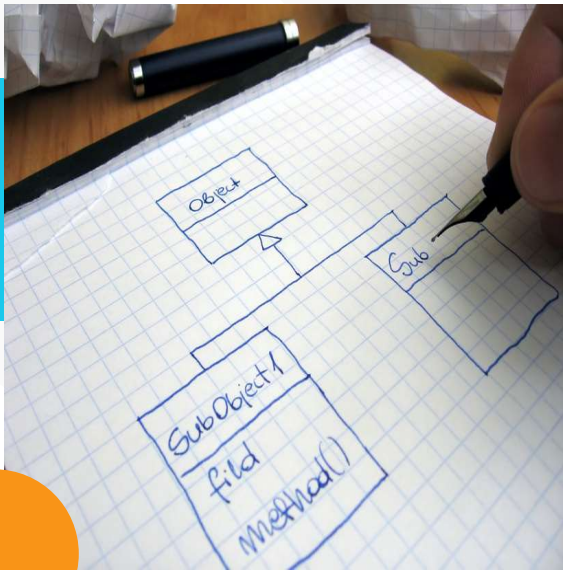


# Diagrama de classes

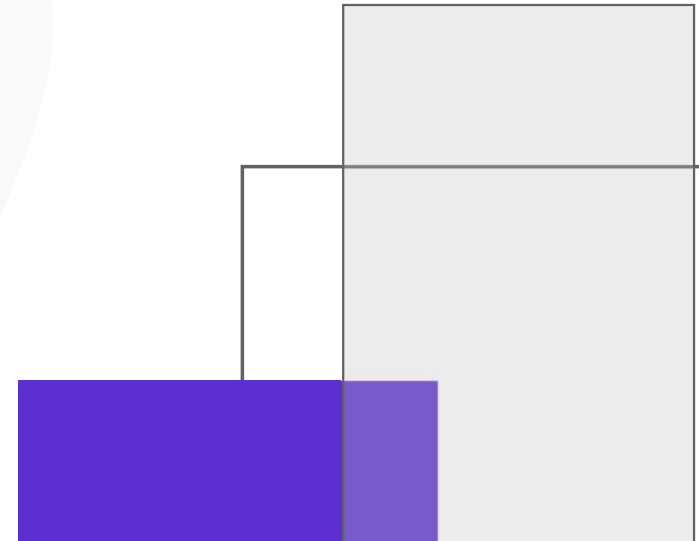
---



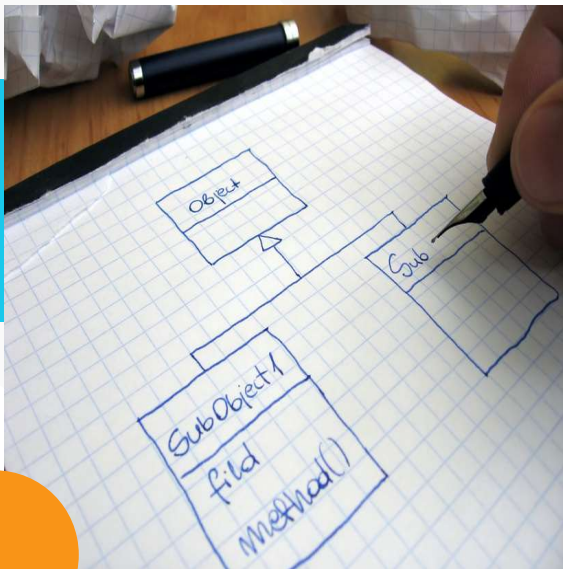
# Conceito de Orientação a Objetos



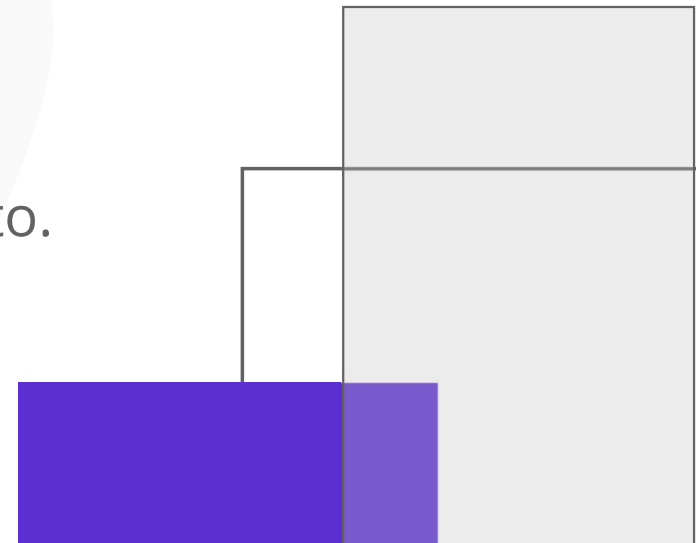
- Padrão de desenvolvimento.
- Como modelar os problemas do mundo real.
- Smaltalk 80.
- C++.

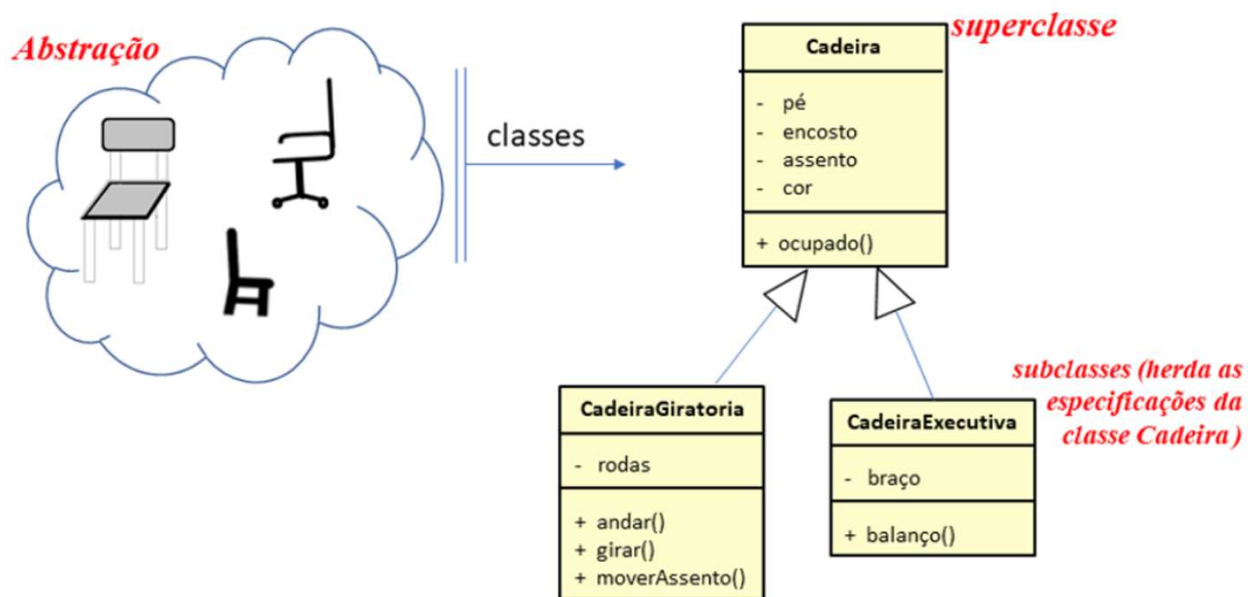


# Conceito de **Orientação a Objetos**



- Abstração.
- Classe.
- Atributos e métodos.
- Objeto.
- Herança.
- Encapsulamento.
- Polimorfismo.





*Exemplos de objetos da classe Cadeira Giratória (são instâncias da classe Cadeira Giratória)*

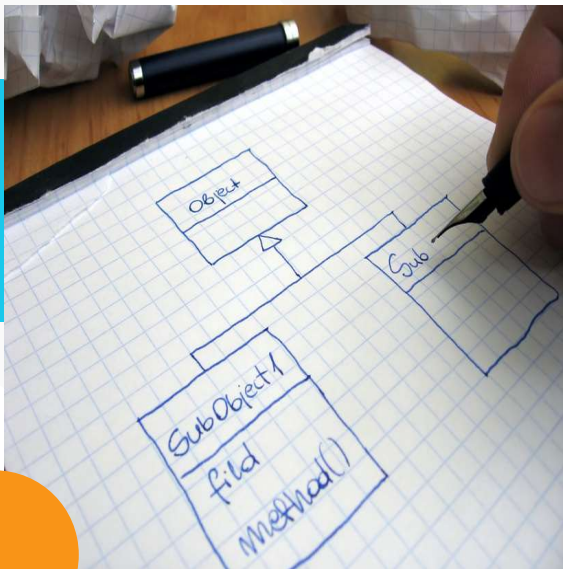
<b>CadeiraGiratória: cadeiraCinza</b>
pé=4 encosto= 1 assento=1 cor=azul rodas = 4
ocupada() andar() girar() moverAssento()

<b>CadeiraGiratória: cadeiraMarrom</b>
pé=3 encosto= 1 assento=1 cor=preta rodas = 3
ocupada() andar() girar() moverAssento()

Fonte: Werlich (2020, p. 172).



# Vantagens de Orientação a Objetos



- Reutilização de código.
- Utilização de um único padrão conceitual para a análise, o projeto e a implementação.
- O tempo de desenvolvimento do software é mais rápido.
- Simplificação.

# Diagrama de classes

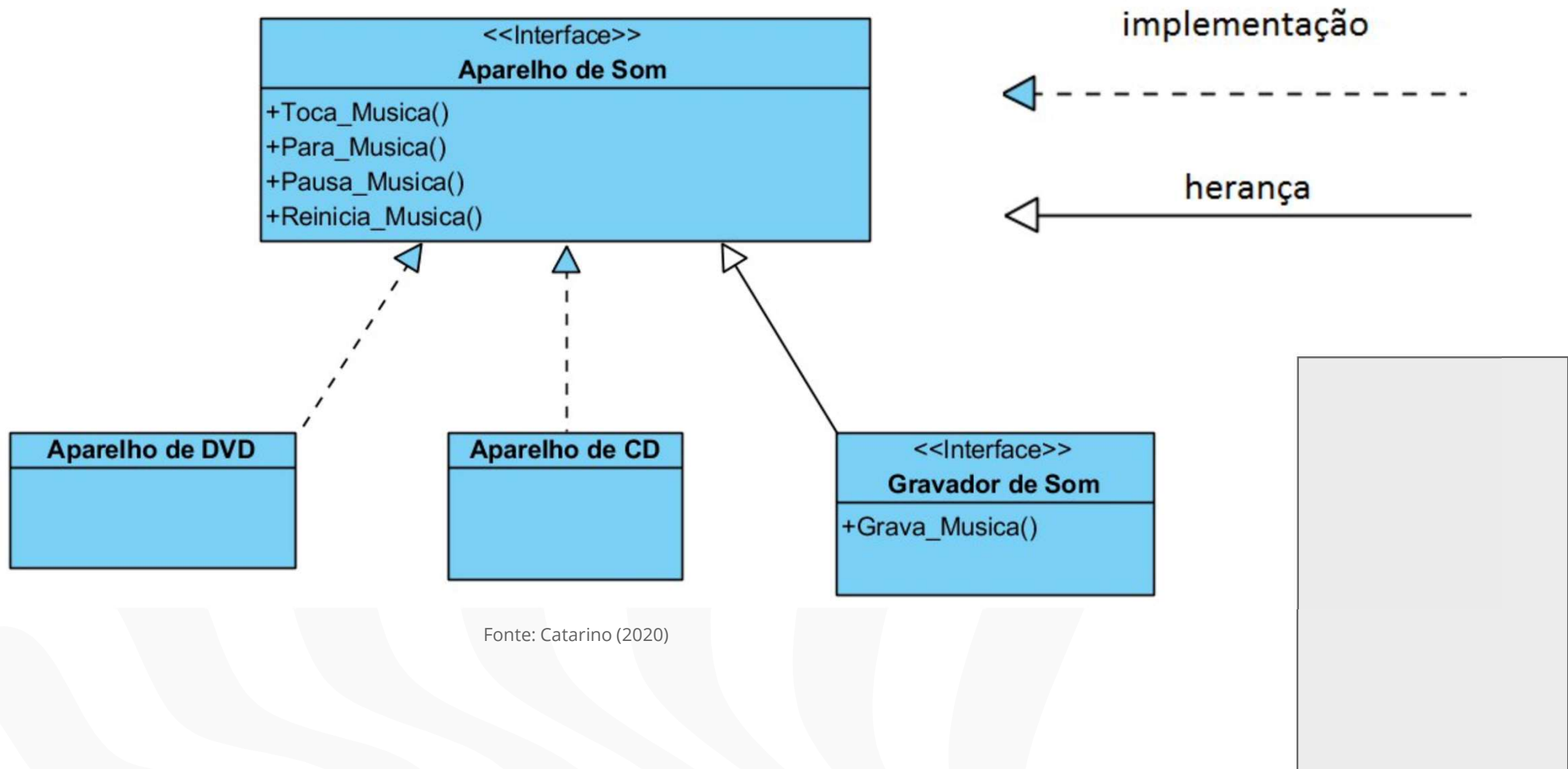
Nome da classe
-atributo +atributo 2
+método() +método2() -método3()

Exemplo de Classe
-Atributo 1 -Atributo 2 -Atributo 3
+getAtributo 3() +setAtributo 3(Atributo 3) : void

<<Interface>> Exemplo de Interface
-Atributo da Interface
+getAtributo da Interface() +setAtributo da Interface(Atributo da Interface) : void

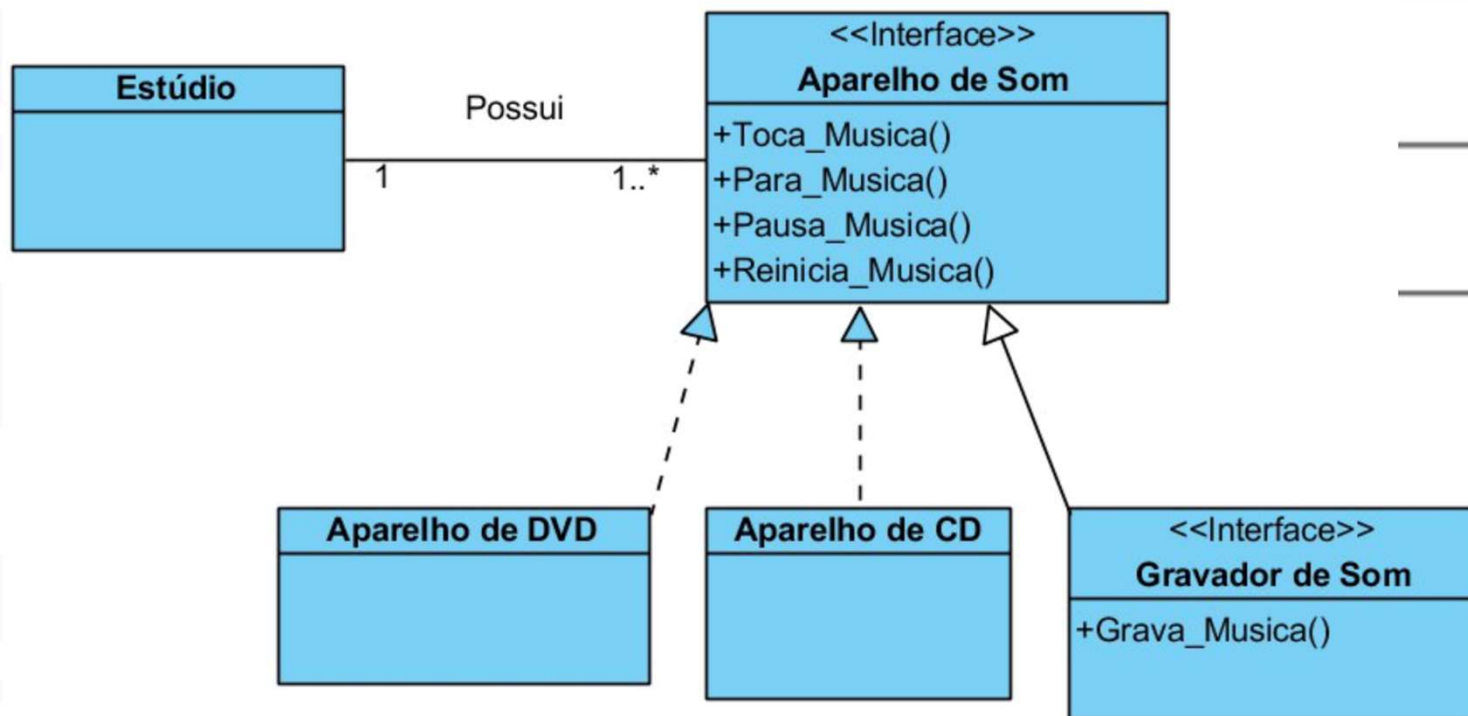
Fonte: Catarino (2020)

# Diagrama de classes

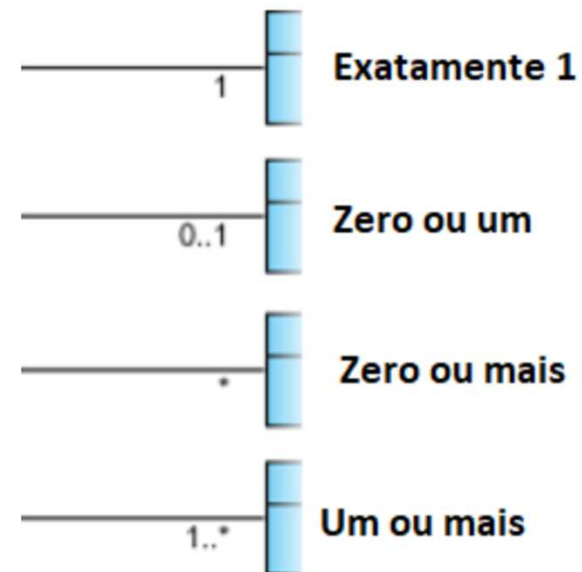




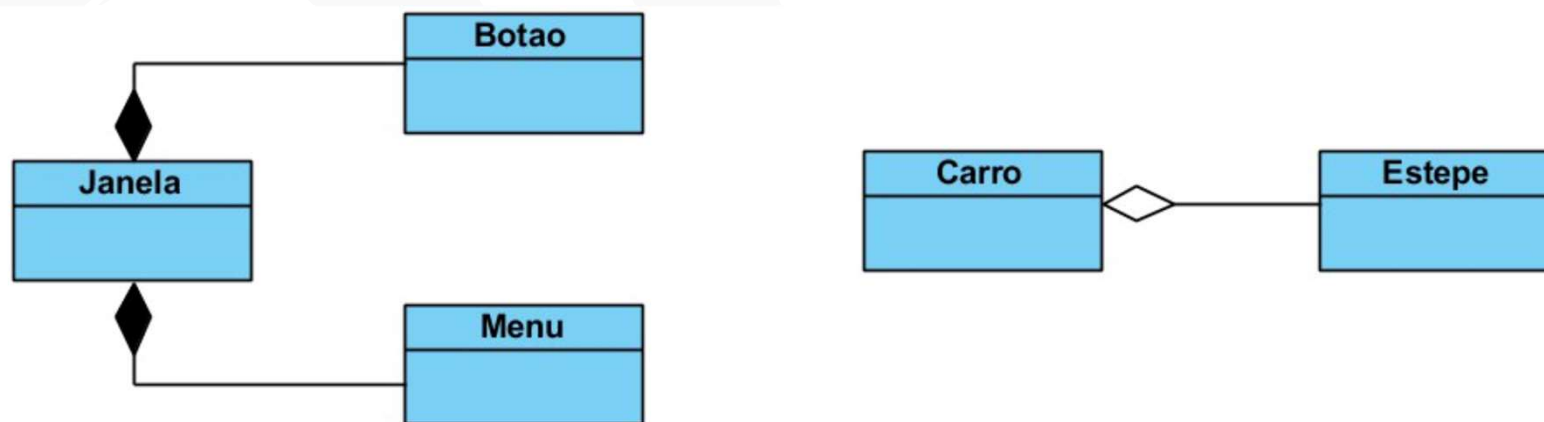
# Diagrama de classes



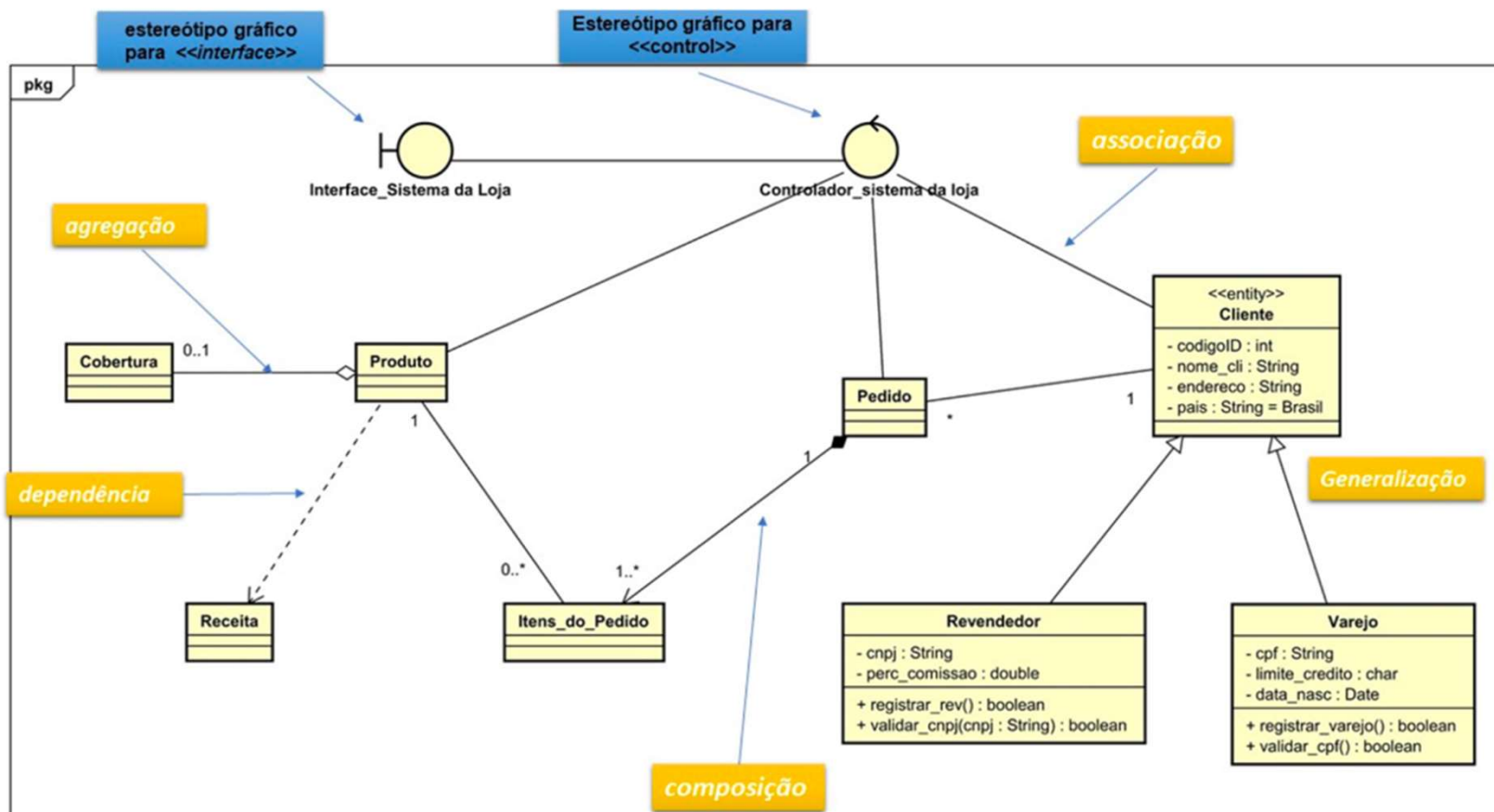
Fonte: Catarino (2020)



# Diagrama de Classes



Fonte: Catarino (2020)



Fonte: Werlich (2020, p. 208).