



# Módulo 13






# BackEnd Java

---

Rodrigo Pires



A decorative graphic on the left side of the slide. It features a large, central cyan hexagon. Surrounding it are several smaller hexagons in various shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, a gear, and a speech bubble. There is also a small network diagram icon with a central node and several connecting lines.

# Interfaces, heranças, composição e agregação

A cluster of hexagonal icons in various shades of blue and cyan. The icons include a lightbulb, a thumbs-up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble. The central hexagon is the largest and contains the number '1'.

1

# Interfaces

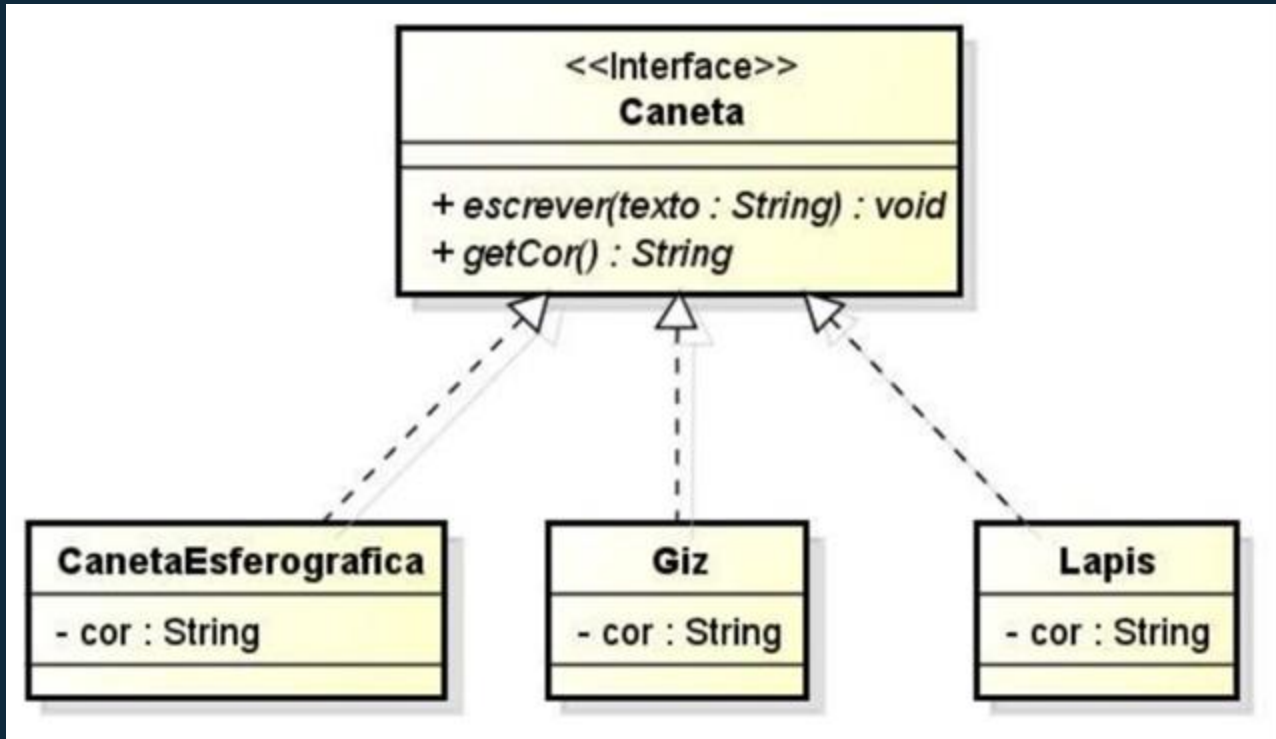


# Interfaces

→ **Interface:** Interface é um contrato que quando assumido por uma classe deve ser implementado.



# Interfaces




A decorative graphic on the left side of the slide. It features a large central hexagon with a blue-to-teal gradient, containing the white number '2'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and teal. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines, and a speech bubble icon.

2

## Classes abstratas

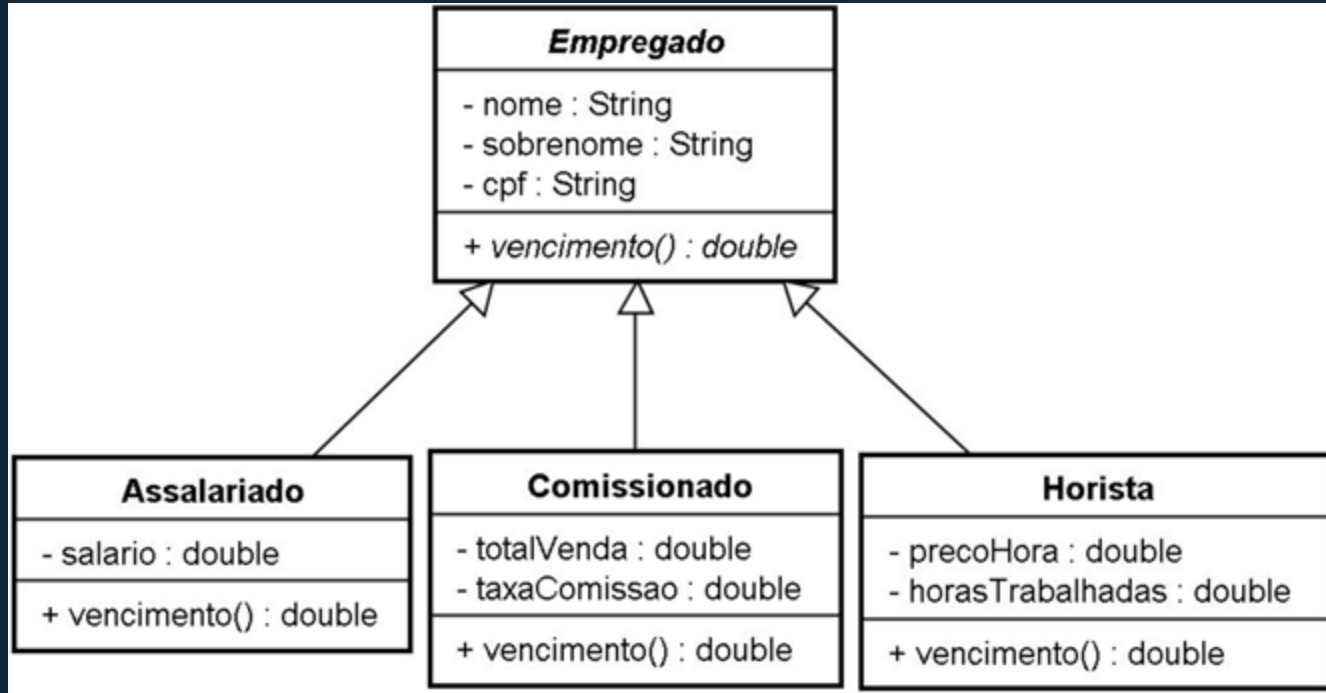


# Classe abstrata

- **Classe abstrata:** Uma classe abstrata é uma classe que serve de modelo para outras classes. Ela sempre será uma superclasse genérica, e suas subclasses serão mais específicas. Além disso, ela não pode ser instanciada e pode conter ou não métodos abstratos, podendo ser implementados nas classes descendentes.
  - **Herança:** A herança é um mecanismo da Orientação a Objeto que permite criar novas classes a partir de classes já existentes, aproveitando-se das características existentes na classe a ser estendida. A linguagem Java permite o uso de herança simples, mas não permite a implementação de herança múltipla.
- 



# Classe abstrata



A decorative graphic on the left side of the slide. It features a large central hexagon with a blue-to-teal gradient, containing the white number '3'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and teal. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines, and a speech bubble icon. The entire graphic is set against a dark blue background.

3

## Agregação e composição

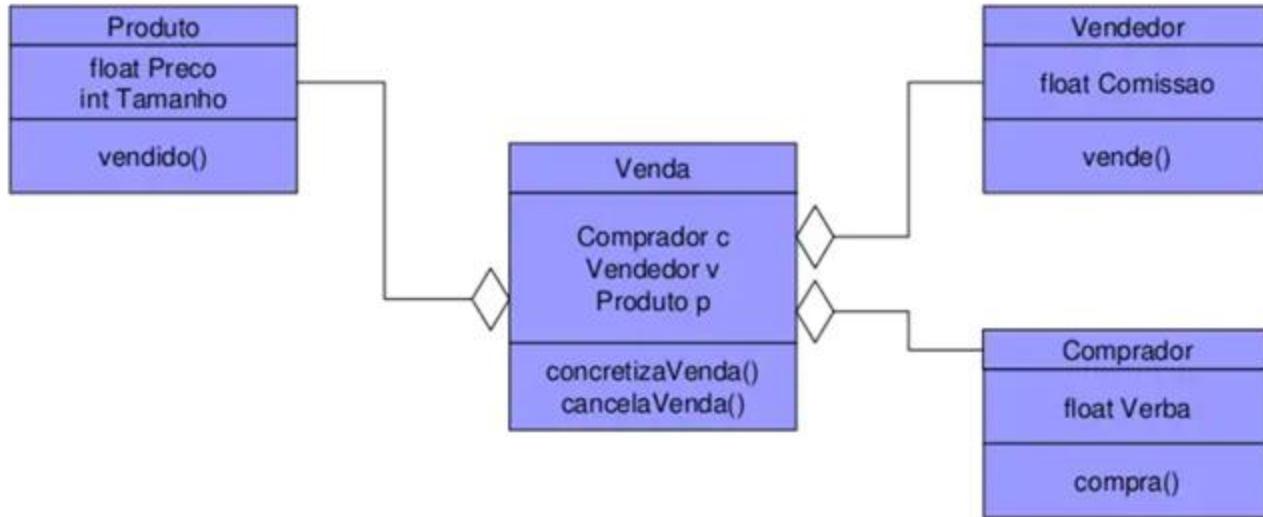


# Agregação

→ **Agregação:** É quando um objeto necessita de outro objeto já criado para existir.



# Agregação



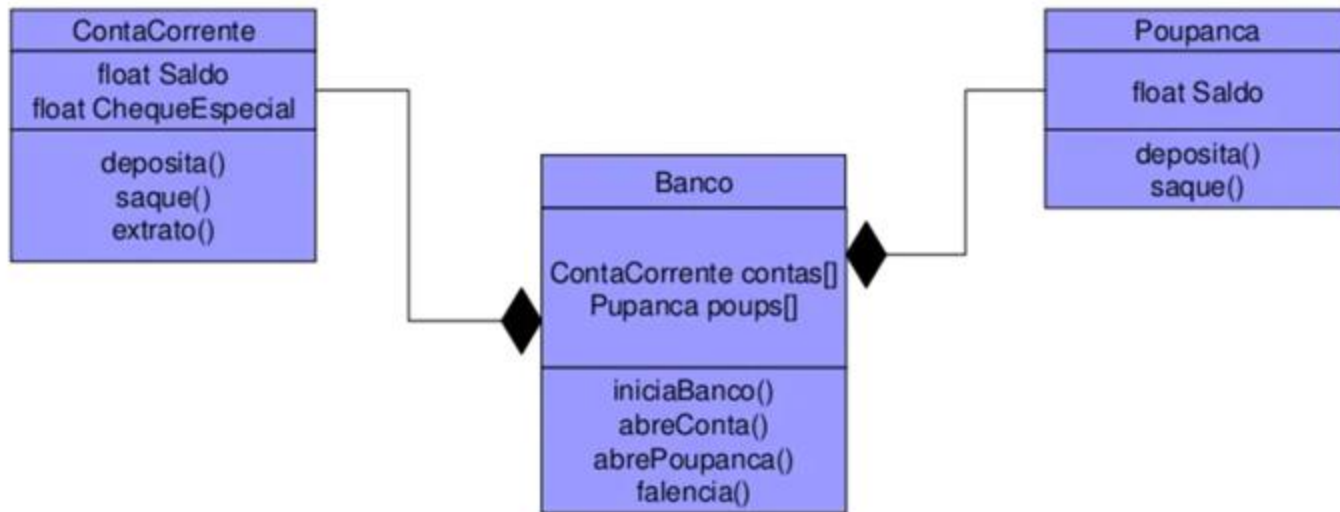


# Composição

→ **Composição:** É quando o objeto não depende de outro, mas quando um objeto morre ou é excluído, ele exclui todos os seus relacionamentos,

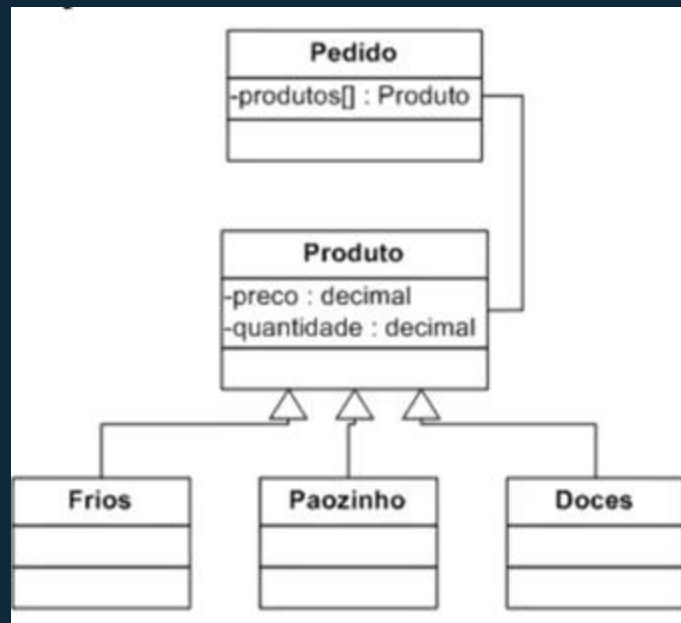


# Composição



# Exemplo Bônus

Olhar no projeto ExemplosMod13 no pacote `br.com.rpires.exemplo.abstracao.bonus`





# Referências

[Exemplos disponíveis no meu github:](#)

<https://github.com/digaomilleniun/backend-java-ebac>

