

# MODELAGEM DE SISTEMAS

## Unidade 4 – Modelo de Classes de Projeto

Luiz Leão – [luizleao@gmail.com](mailto:luizleao@gmail.com)

<http://www.luizleao.com>



**Estácio**

# Conteúdo Programático

- Definição da Visibilidade entre Objetos
- Adição de Operações às Classes de Projeto
- Adição de Interfaces ao Modelo de Classes de Projeto
- Relacionamentos de Dependência
- Adição de Classes Utilitárias e de Coleções ao Modelo de Classes de Projeto
- Diagramas de Classes de Projeto

# Introdução

- Representa o modelo físico do banco de dados, com a transformação das classes associativas em classes de objeto, transporte das chaves primárias para as classes associadas (definição das chaves estrangeiras), estabelecendo a ligação, a indicação dos métodos, a visibilidade e dependência entre as classes.

# Visibilidade Entre Objetos

- Habilidade de um objeto poder “ver” ou ter referência para outro objeto.
- Quando se cria um projeto de objetos que interagem entre si, é preciso assegurar que a **visibilidade** necessária esteja presente.
  - Emissor da mensagem deve ter algum tipo de referência ou ponteiro para o objeto receptor.
  - Visibilidade está relacionada com **escopo**: um recurso (uma instância) está dentro do escopo de outro?

# Visibilidade Entre Objetos

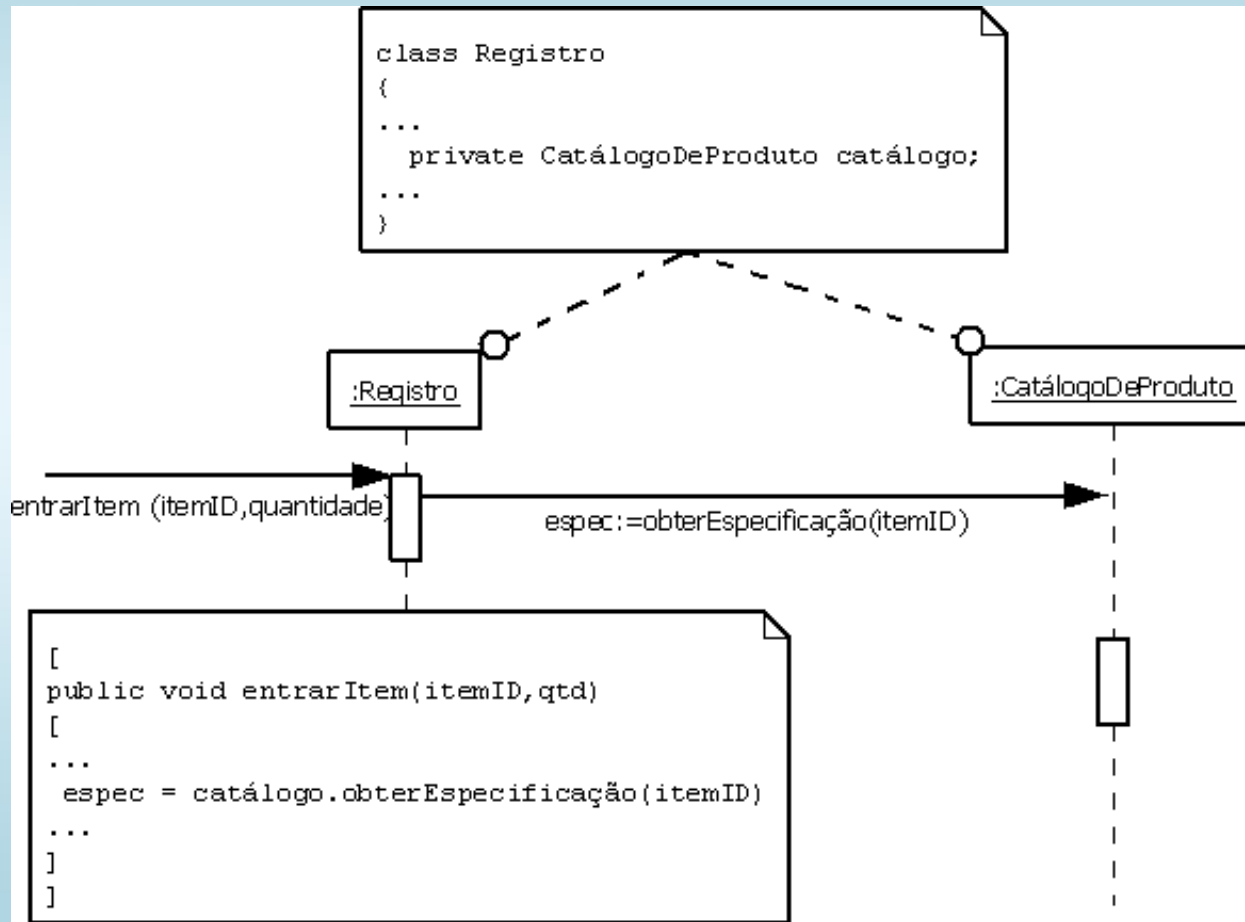


Figura 18.1 – A Visibilidade do Registro para o CatálogoDeProduto é exigida.

# Visibilidade

- Para um objeto A enviar uma mensagem a um objeto B, B deve ser visível para A.
- **Tipos de visibilidade:**
  - **Por Atributo:** B é um atributo de A
  - **Por Parâmetro:** B é um parâmetro de um método de A
  - **Localmente Declarada:** B é declarado como um objeto local em um método de A
  - **Global:** B é, de alguma forma, globalmente visível.

# Tipos de Visibilidade

- **Visibilidade Por Atributo:**

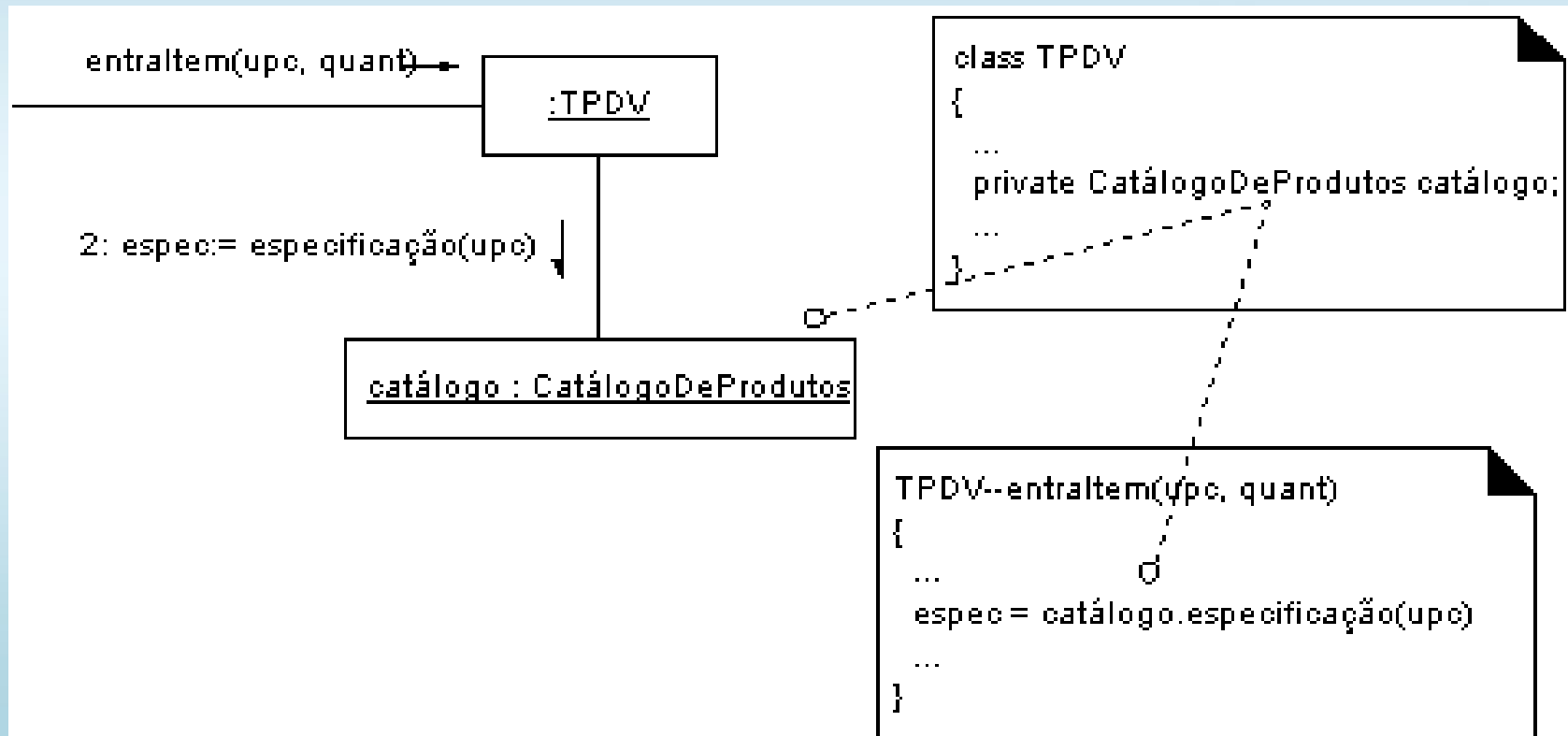


Figura 18.2 – Visibilidade por atributo

# Tipos de Visibilidade

- **Visibilidade Por Parâmetro:**

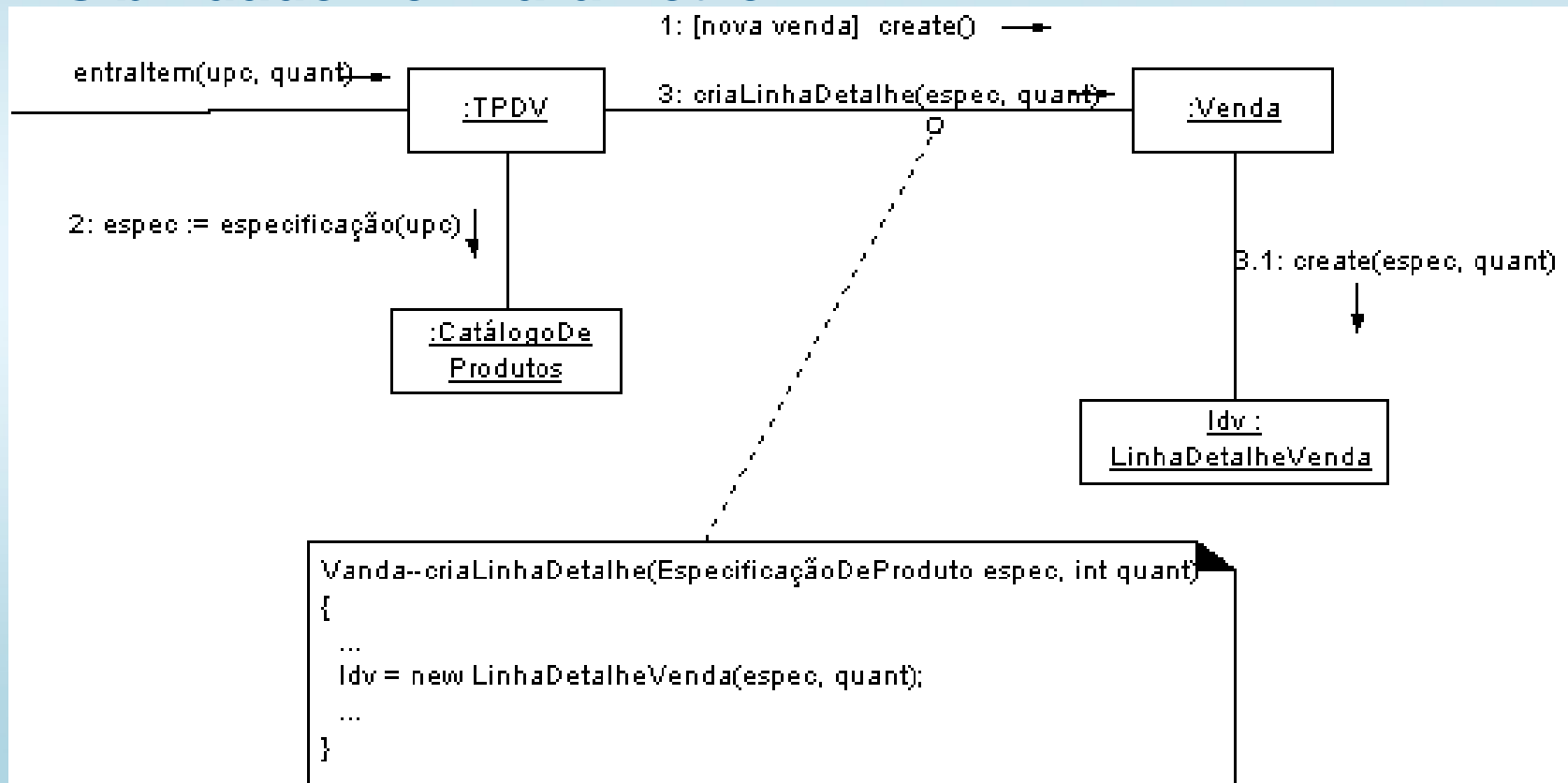


Figura 18.3 – Visibilidade por parâmetro



# Tipos de Visibilidade

- **Visibilidade Por Parâmetro:**

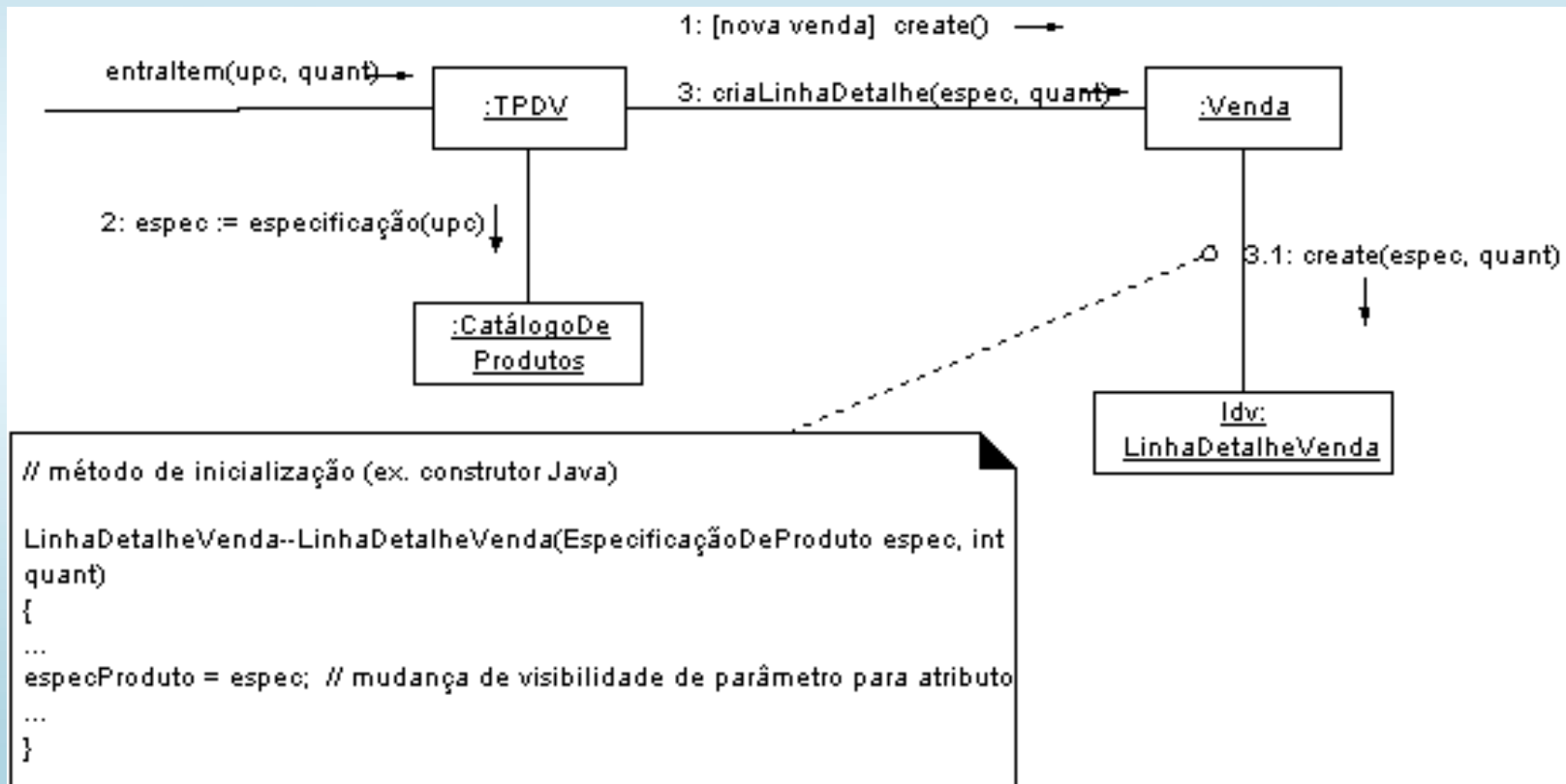


Figura 18.1 – A Visibilidade de parâmetro para atributo.

# Tipos de Visibilidade

- **Visibilidade Local:**

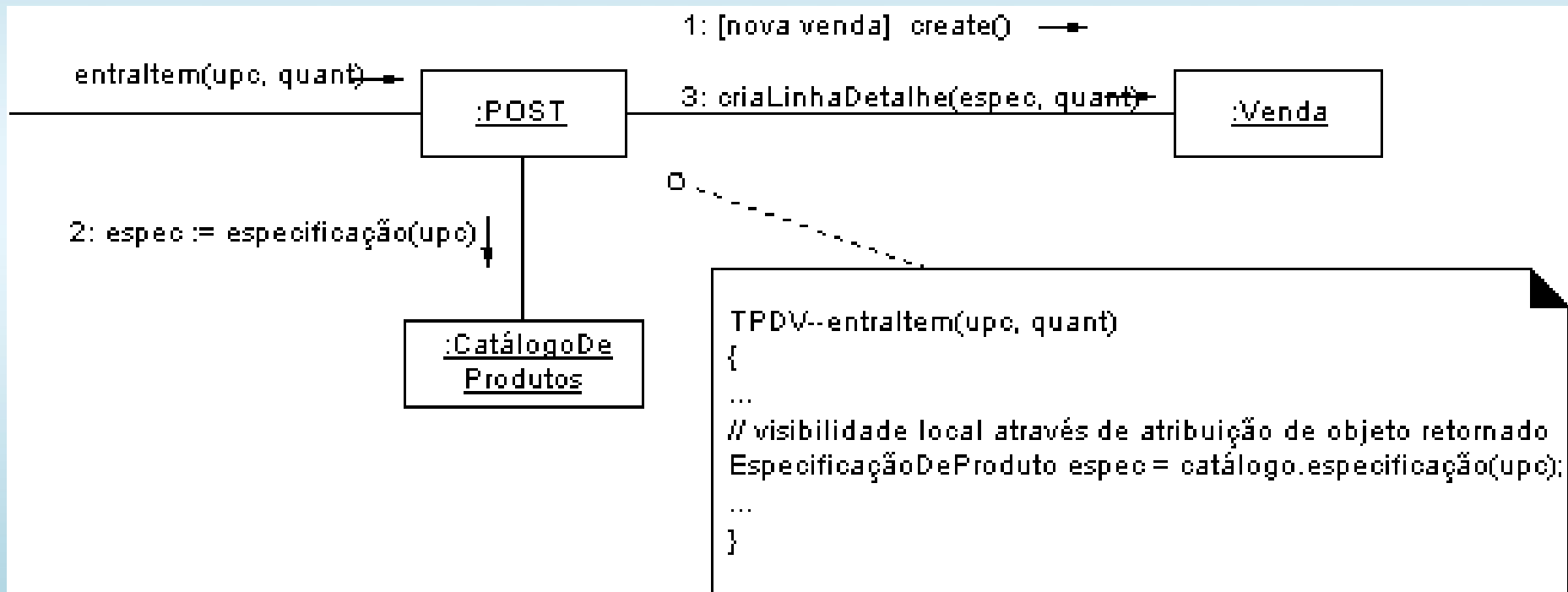


Figura 18.1 – Visibilidade local

# Tipos de Visibilidade

- **Visibilidade Global:**
  - Um objeto global é visível a todos
  - Não uma boa forma de ter visibilidade

# Visibilidade na UML

- Como Ilustrar Visibilidade na UML

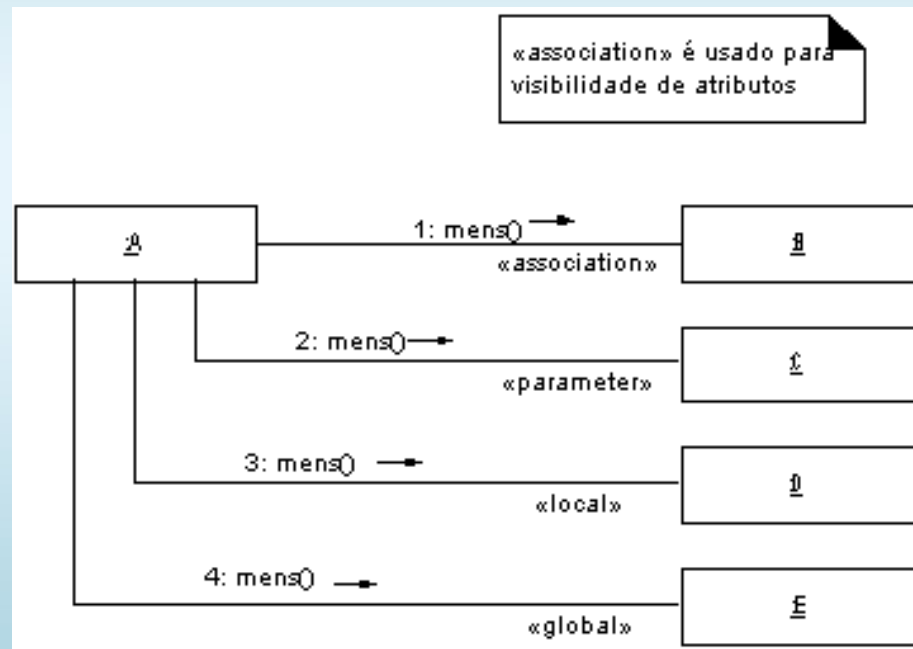


Figura 18.6 – Implementação de estereótipos para visibilidade

# Como Criar Diagramas de Classe de Projeto

- **Objetivos:**
  - Criar Diagramas de Classes de Projeto (DCPs)
  - Identificar as classes, os métodos e as associações para mostrar em um DCP.

## O que é e Quando Criar DCPs?

- Especifica as classes de software e as interfaces da aplicação;
  - Não somente das entidades conceituais.
- Normalmente criado em paralelo com os diagramas de interação.

# O que é e Quando criar DCPs

- Informação tipicamente incluída:
  - Classes, associações e atributos
  - Interfaces, incluindo métodos e constantes
  - Métodos
  - Informação de tipo de atributos
  - Navegabilidade
  - Dependências

# Exemplo de DCPs

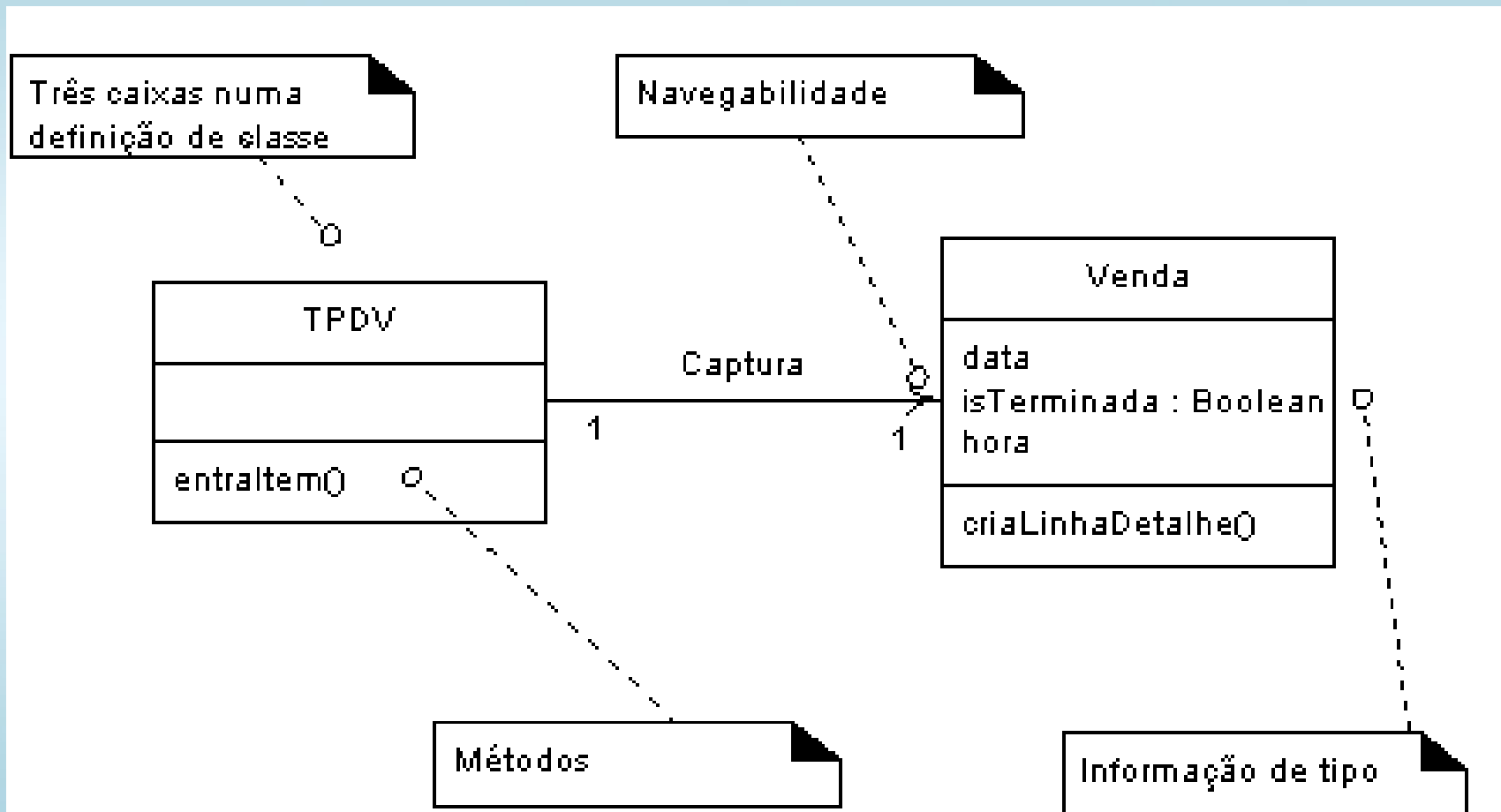


Figura 19.1 – Amostra de diagrama de classe de projeto



## Modelo de Domínio *Versus* Classes de Modelo de Projeto

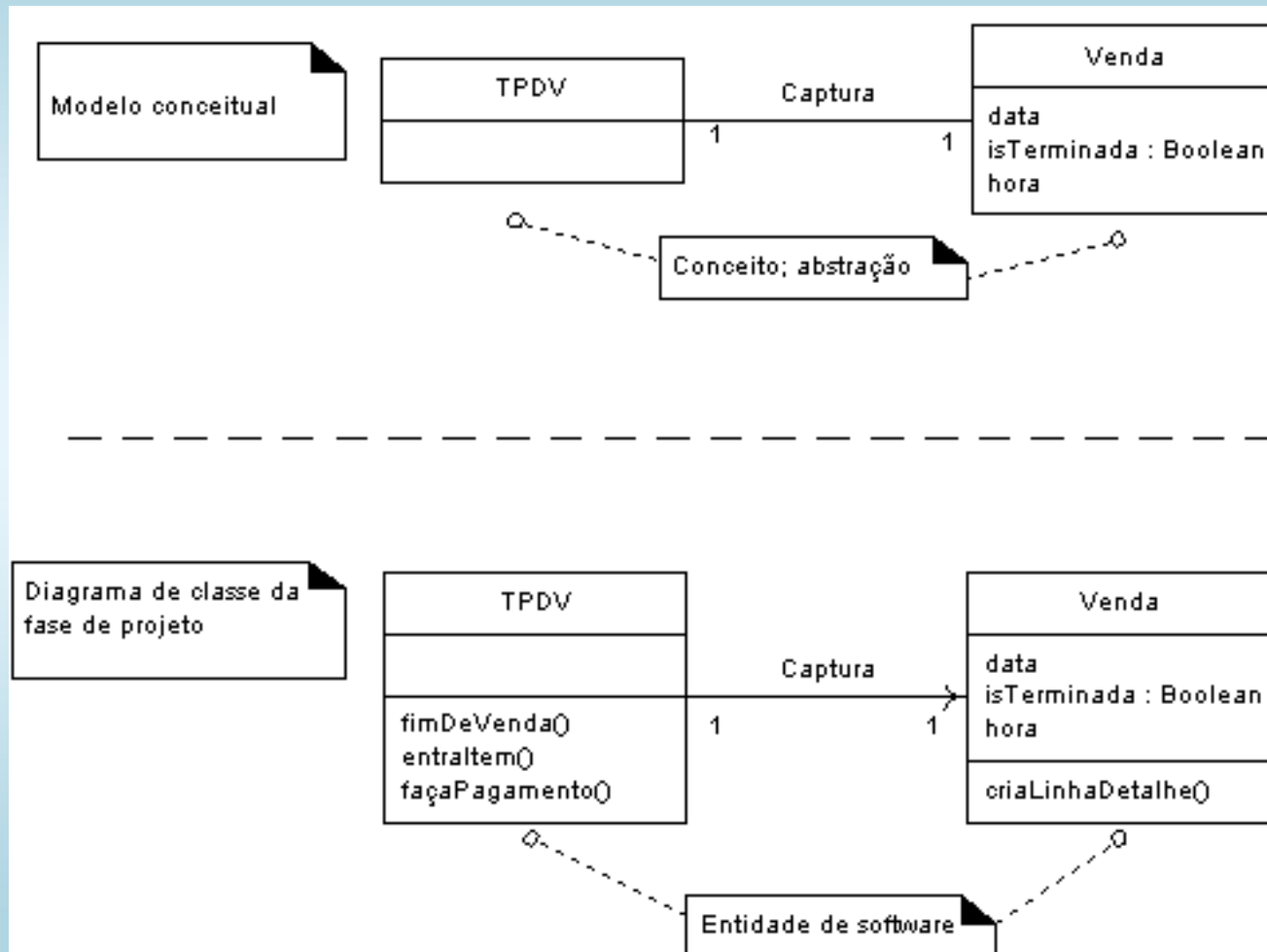


Figura 19.2 – Modelo do domínio vs classes do modelo de projeto

# Criação De Um DCP Para o Estudo de Caso

- Identificar as classes que participam da solução de software.
  - Verificar os diagramas de interação para identificar as classes.
- Desenhar um diagrama de classe para estas classes e incluir os atributos identificados.

# Criação De Um DCP Para o Estudo de Caso

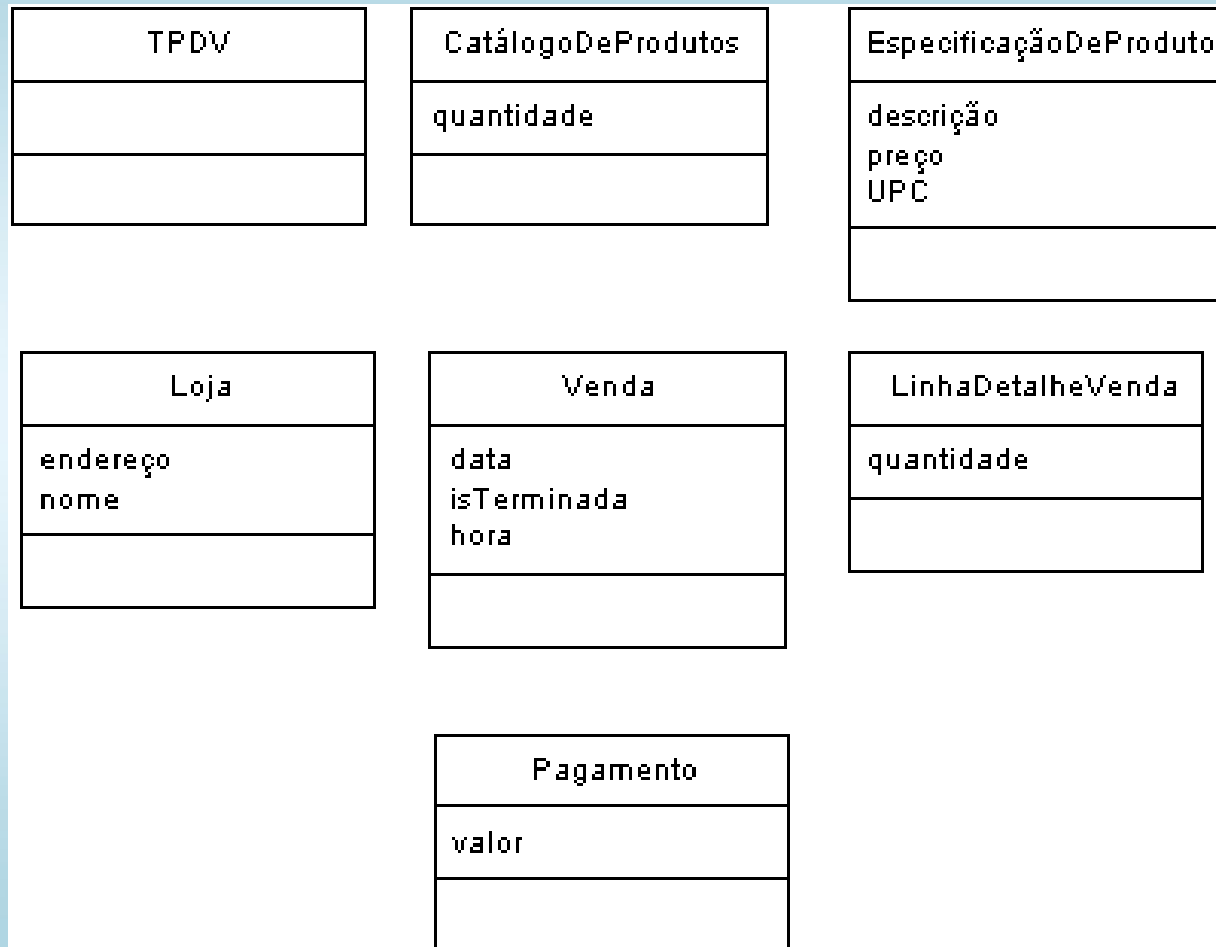


Figura 19.3 – Classes de software na aplicação

# Criação De Um DCP Para o Estudo de Caso

- Acrescentar nomes de métodos

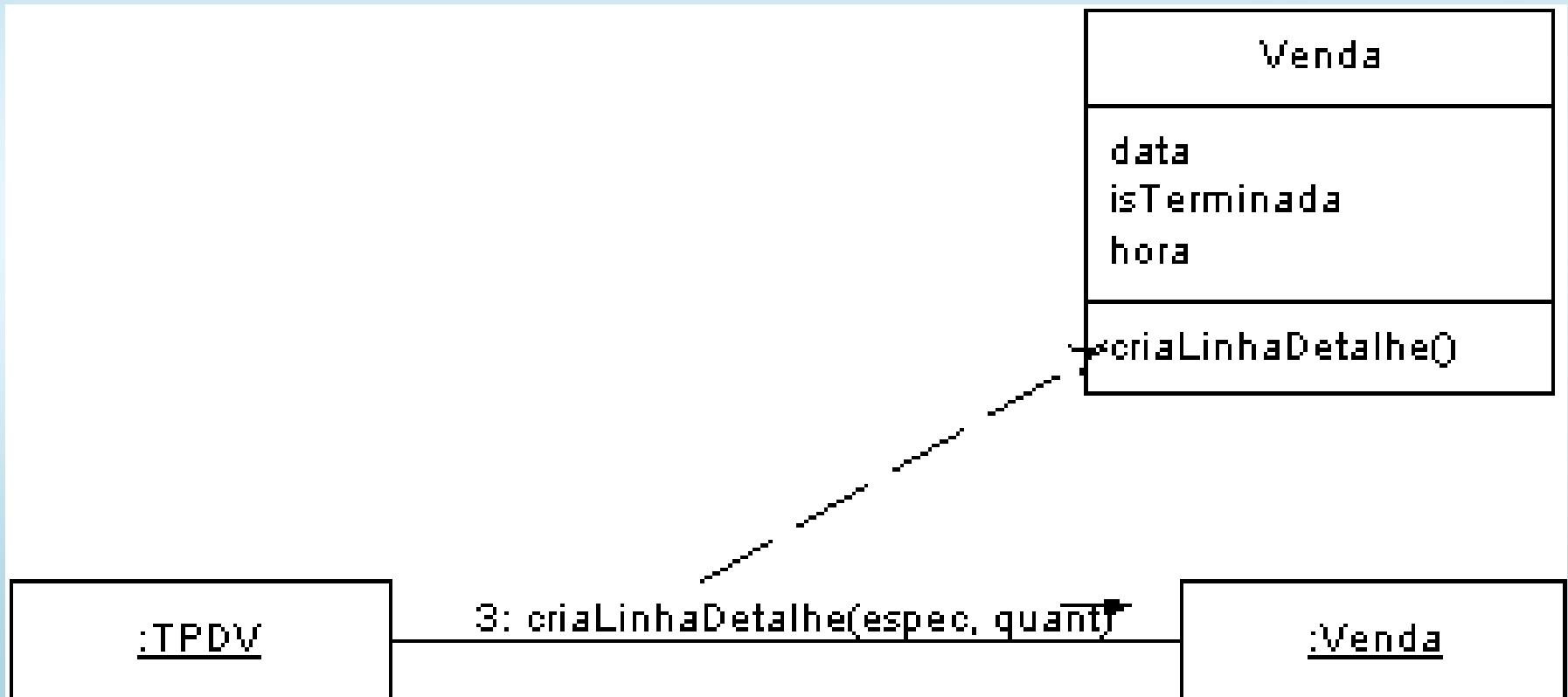


Figura 19.4 – Nomes de métodos a partir dos diagramas de interação

# Criação De Um DCP Para o Estudo de Caso

- Acrescentar nomes de métodos:

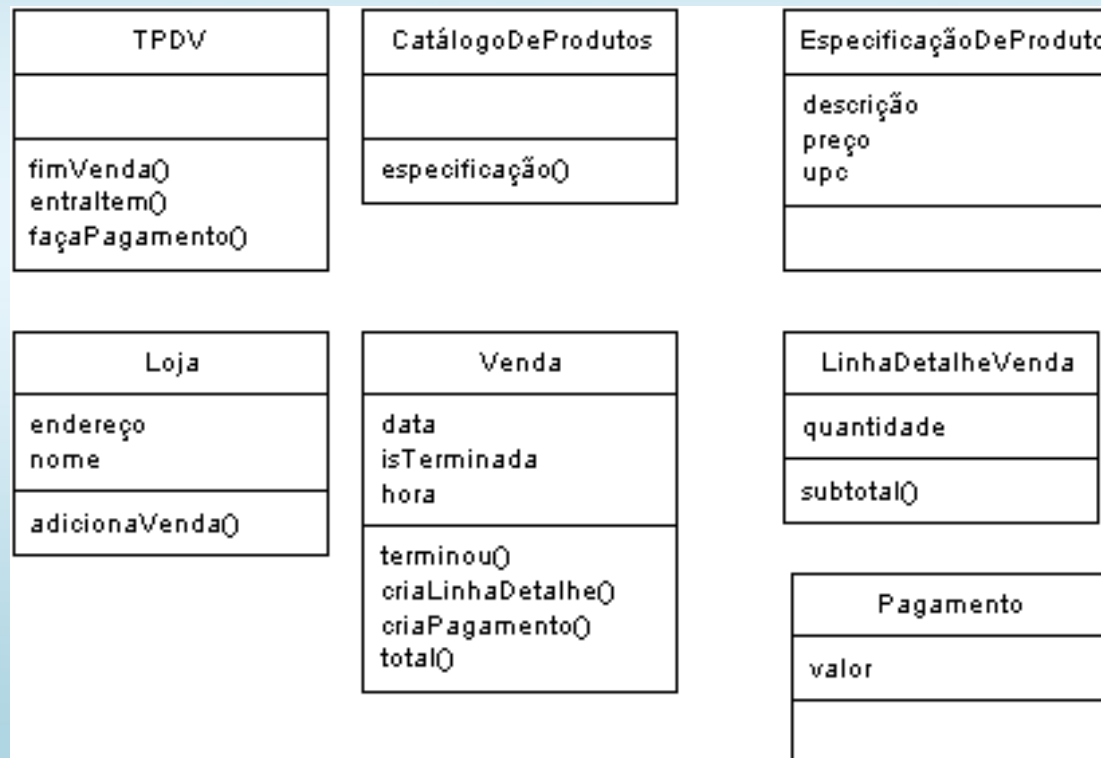


Figura 19.5 – Métodos na aplicação

# Criação De Um DCP Para o Estudo de Caso

- Acrescentar informação de tipo

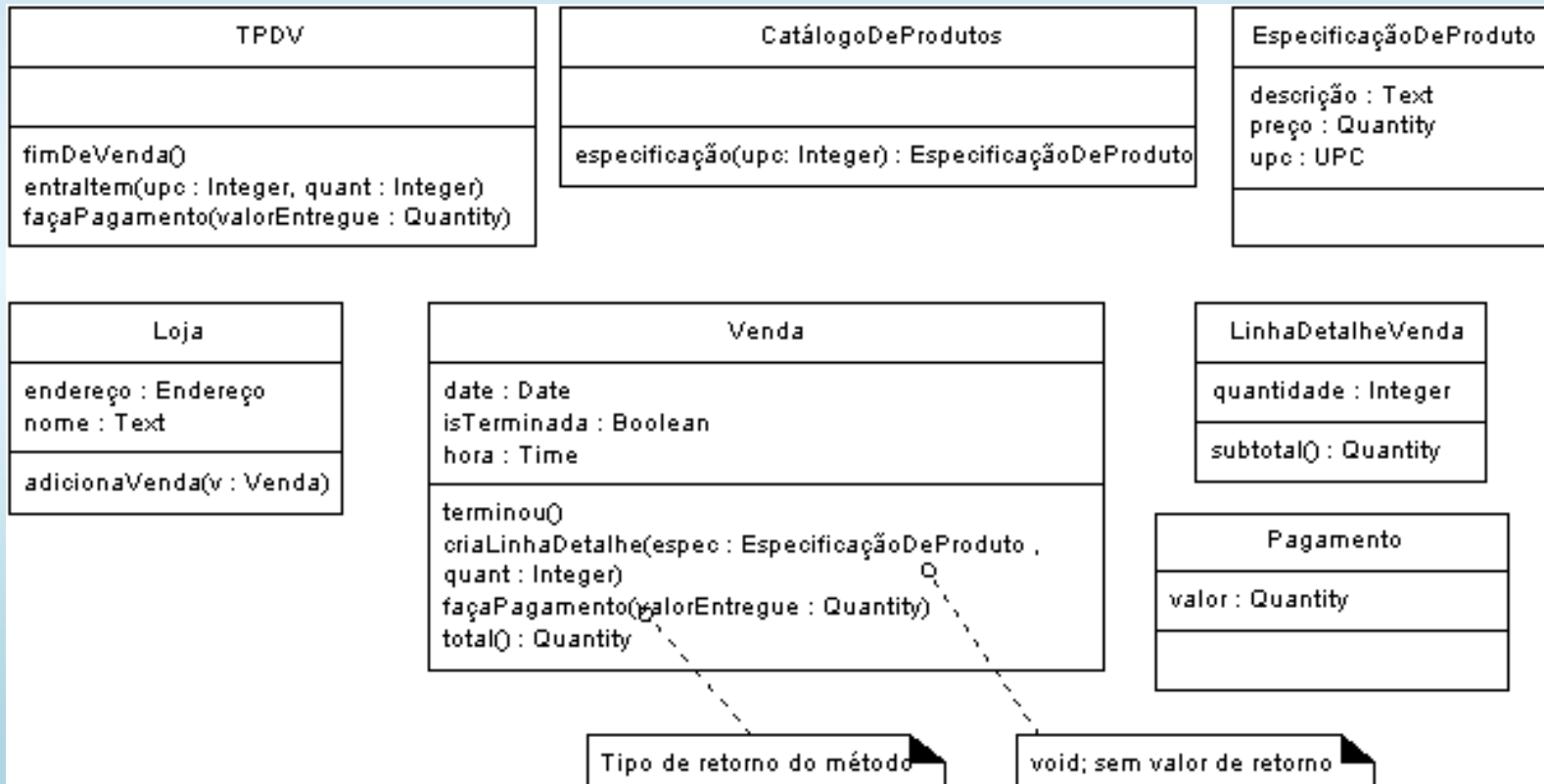


Figura 19.7 – Informação de tipo

# Criação De Um DCP Para o Estudo de Caso

- Acrescentar associações e navegabilidade

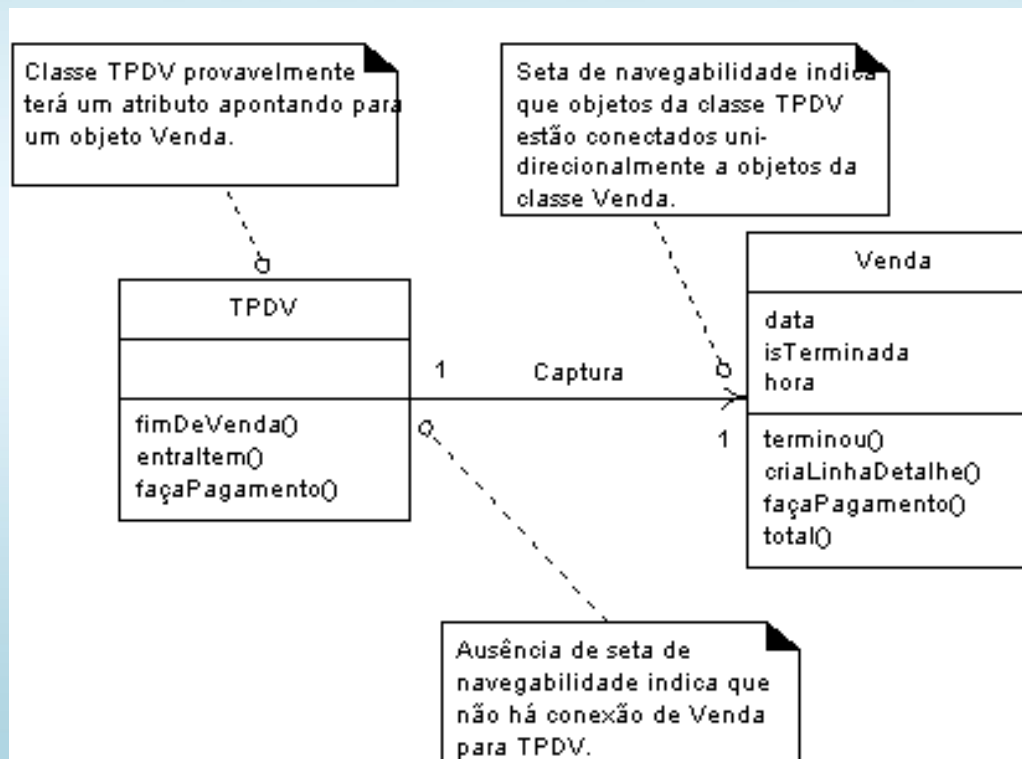


Figura 19.8 – Mostrar navegabilidade ou visibilidade do atributo

# Criação De Um DCP Para o Estudo de Caso

- Acrescentar associações e navegabilidade

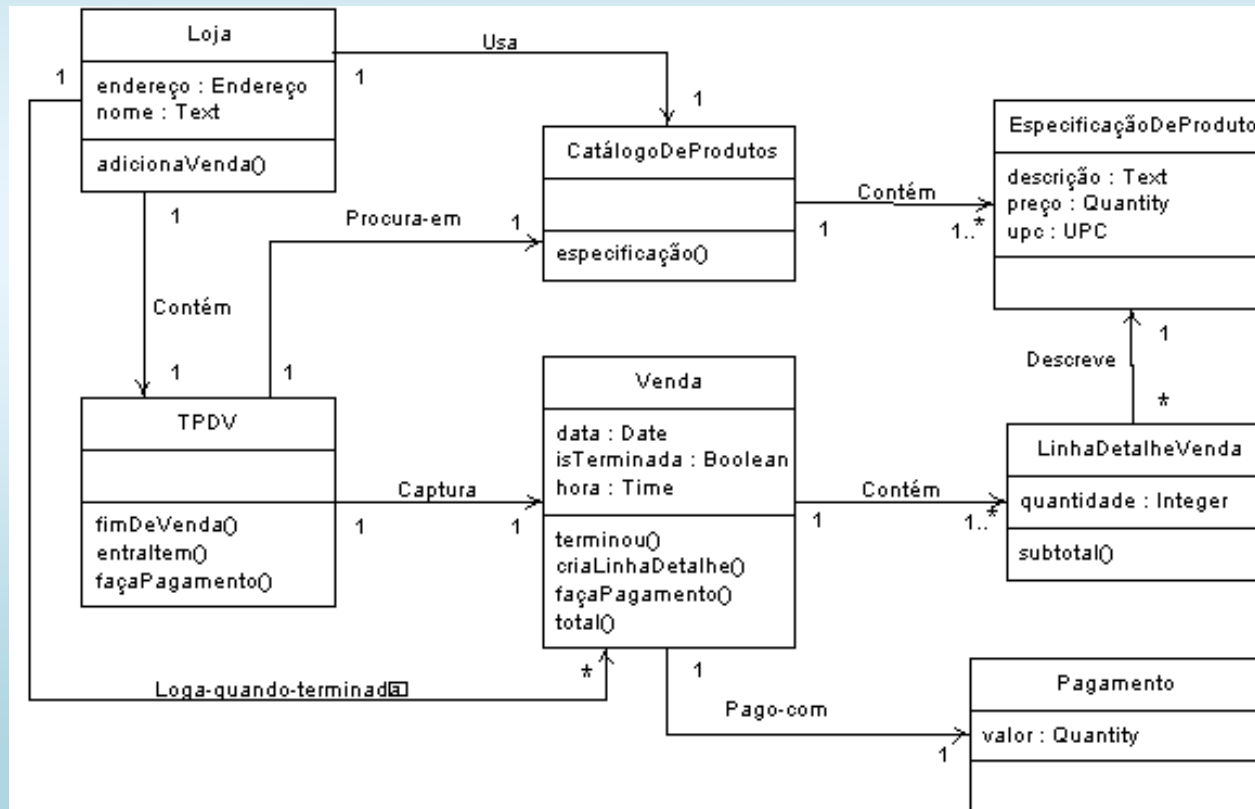


Figura 19.10 – Associações com adorno de navegabilidade



# Criação De Um DCP Para o Estudo de Caso

- Acrescentar relações de dependência

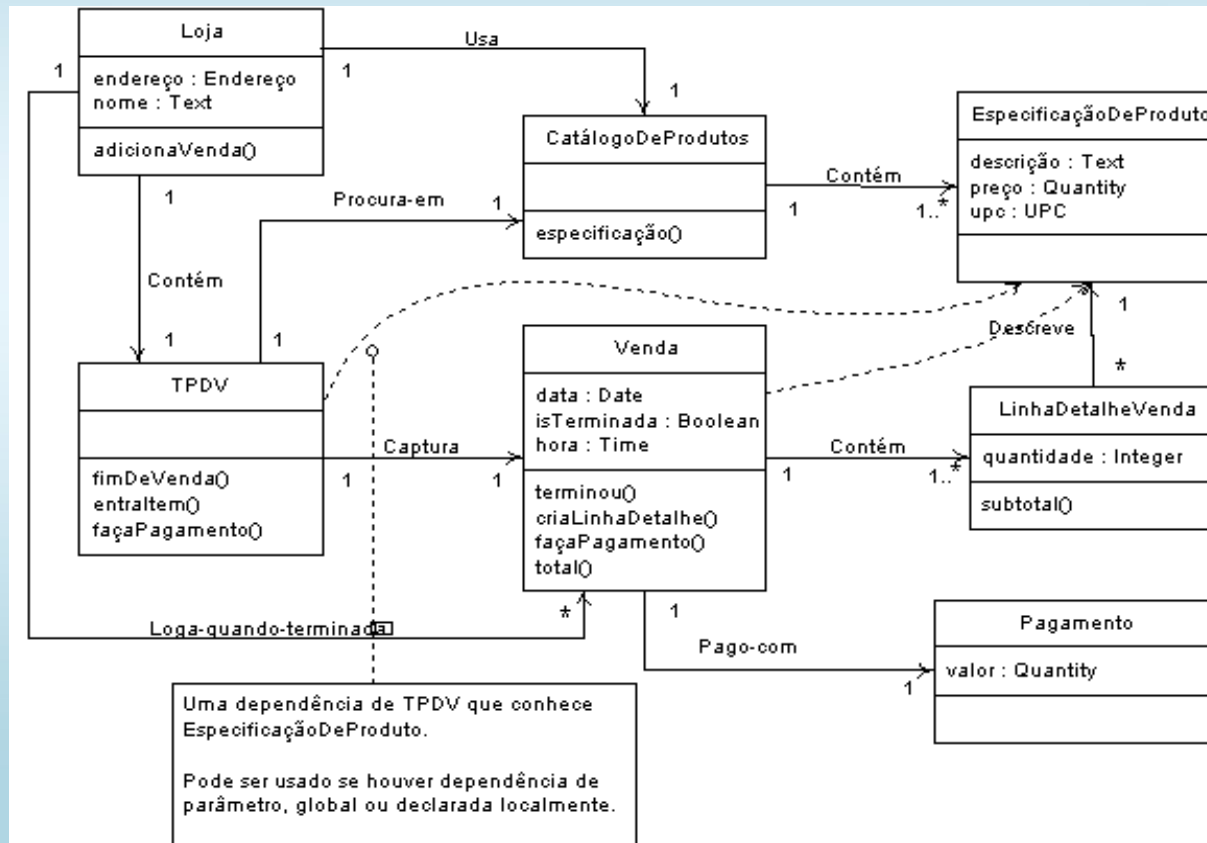


Figura 19.10 – Relacionamentos de dependência que indicam visibilidade que não é implementada por atributo

# Criação De Um DCP Para o Estudo de Caso

- Incluir detalhes de membro de classes
  - Exemplo:

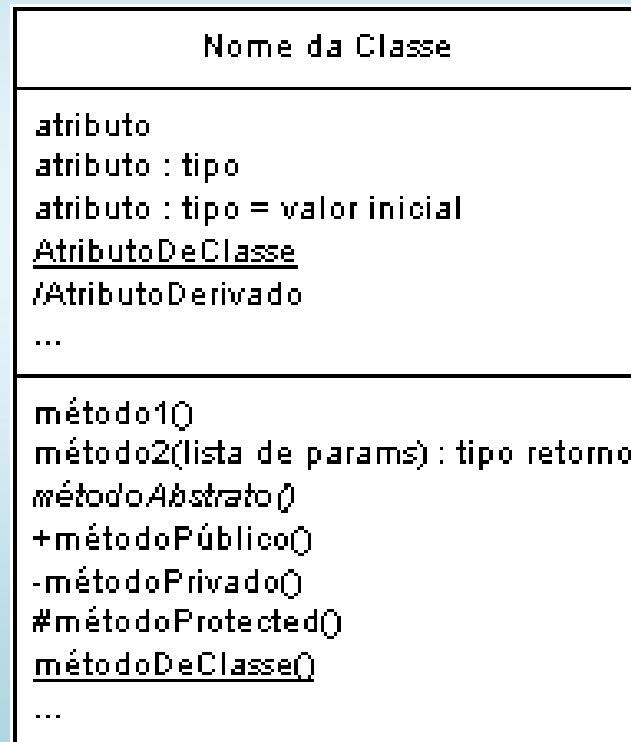


Figura 19.12 – Detalhes da notação de membro do diagrama de classes UML