

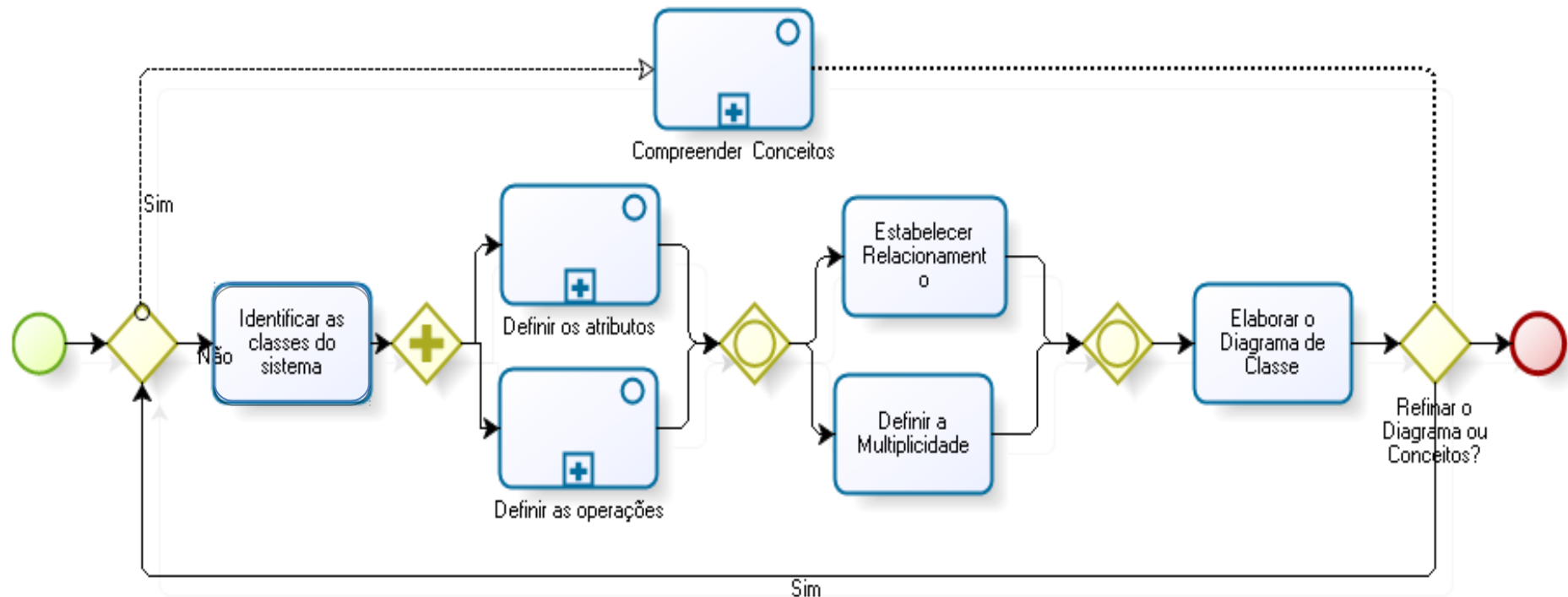
Profa Simone Sawasaki Tanaka

simone.tanaka@unifil.br

Diagrama de Classe

- Mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos.
- Os diagramas de classes abrangem a visão estática do projeto de um sistema.

Workflow do Diagrama de Classe



Identificar as Classes do Sistema

- Todas as classes devem fazer sentido no domínio da aplicação. Nem todas as classes surgem explicitamente na definição do problema; algumas estão implícitas no domínio da aplicação ou no conhecimento geral [30].
- A partir da definição do requisitos, comece a listar as classes de objetos candidatas encontradas na descrição do problema. Não seja muito seletivo; registre todas as classes que vierem à cabeça. As classes muitas vezes correspondem a substantivos. Por exemplo, na frase “um sistema de reservas para vender entradas para peças em diversos teatros”, as classes possíveis seriam “Reserva, Sistema, Entrada, Peça e Teatro” [30].



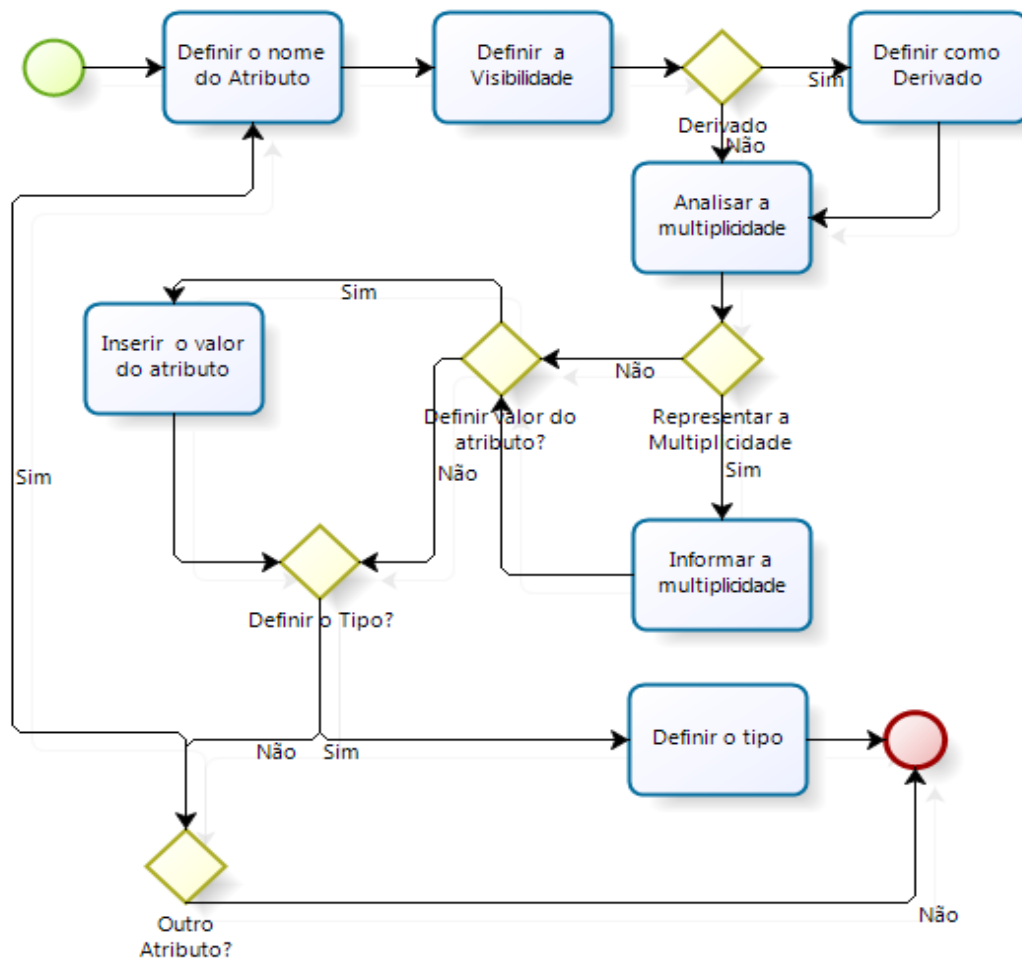
Definir Atributos

- Um atributo é uma propriedade nomeada de uma classe que descreve um intervalo de valores que as instâncias da propriedade podem apresentar.
- Conforme Rup, durante a definição de **métodos** e a identificação de **estados**, são identificados os atributos que a classe precisa para executar suas operações. Os atributos fornecem armazenamento de informações para a instância da classe e muitas vezes são utilizados para representar o estado da instância da classe. Qualquer informação que a própria classe mantém, ela o faz através de seus **atributos**. Para cada atributo, defina:

Definir Atributos

- seu **nome**
- seu **tipo**, que será um tipo de dado elementar suportado pela linguagem de implementação;
- seu **valor padrão ou inicial**, com o qual é inicializado quando novas instâncias da classe são criadas;
- a **visibilidade**
- a **multiplicidade** especifica a quantidade valores que podem estar associados a um elemento do modelo.
- **String_propriedade** O elemento propriedades permite acrescentar praticamente qualquer informação adicional sobre o atributo que não se encaixa em um dos elementos predefinidos

Passo a passo para Definir o Atributo

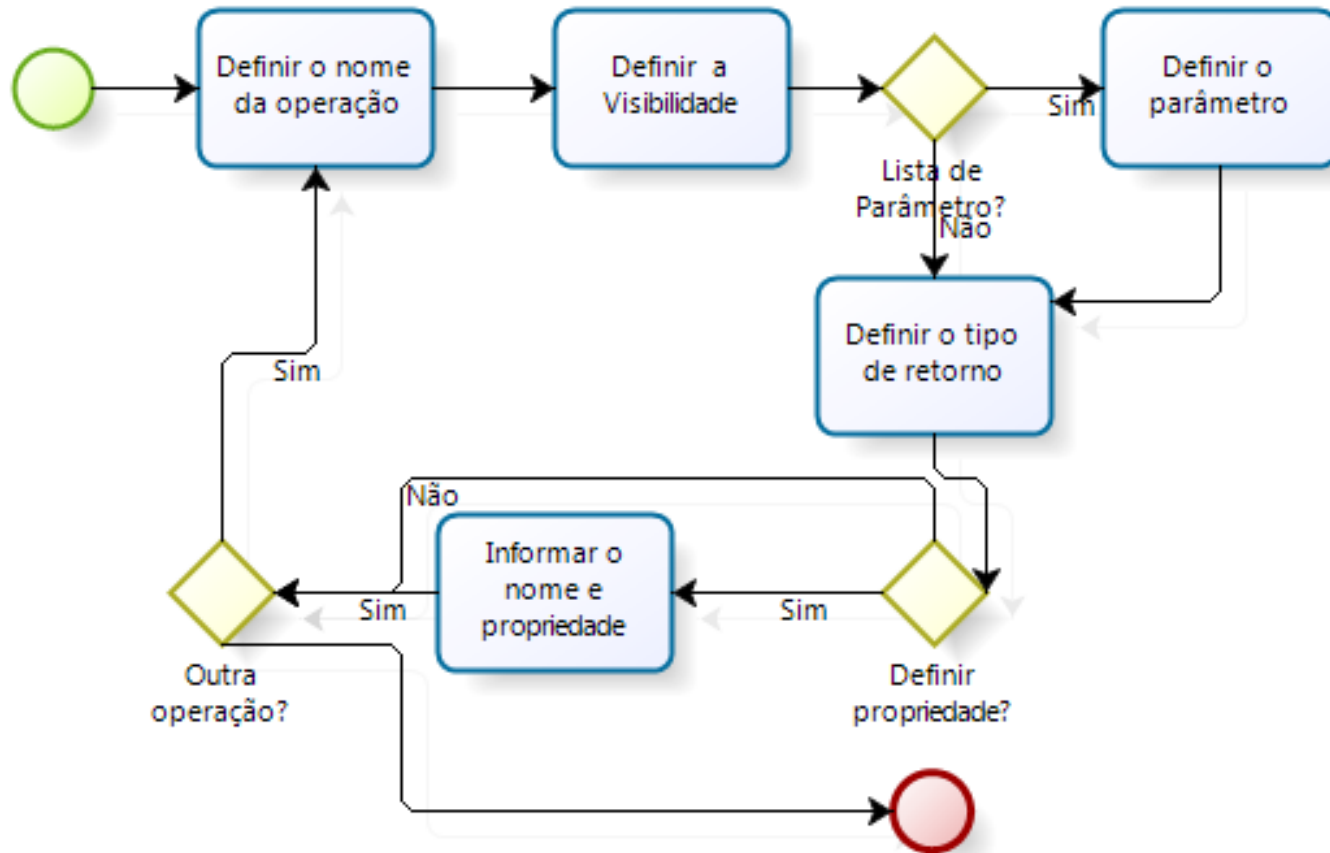


```
[visibilidade] [/] nome-do-atributo : [tipo] [multiplicidade] [=default]
[{string propriedade}]
Nome: String=No default, {Required, 1..30 carac, espaços e pontuação
```

Definir Operação

- Uma operação é uma abstração de algo que pode ser feito com um objeto e que é compartilhado por todos os objetos dessa classe. Uma classe pode ter qualquer número de operações ou até nenhuma operação.
- Para identificar operações em classes :
- Estude as responsabilidades de cada classe de análise correspondente, criando uma operação para cada responsabilidade. Use a descrição da responsabilidade como a descrição inicial da operação.
- Estude as realizações de casos de uso nas participações de classe para ver como as operações são utilizadas pelas realizações de casos de uso. Amplie as operações, uma realização de Caso de Uso por vez, refinando as operações, suas descrições, tipos de retorno e parâmetros. Os requisitos de cada realização de Caso de Uso pertencentes às classes são descritos textualmente no Fluxo de Eventos da realização de Caso de Uso.

Passo a Passo para Definir a Operação



[visibilidade] nome ([lista-parâmetro]) ":" [resultado-retorno] [{propriedades + definirDuracao(nova_duracao: int) : void {a nova duracao não pode causar sobreposição com outro evento planejado}}

Estabelecer a Relacionamento

- As associações proporcionam o mecanismo para os objetos se comunicarem entre si. Elas fornecem aos objetos um canal pelo qual as mensagens podem fluir. Além disso, documentam as dependências entre as classes, destacando que as mudanças em uma classe podem ser percebidas entre muitas outras classes.



Definir a Multiplicidade

- A multiplicidade procura determinar o número mínimo e o máximo de objetos envolvidos em cada extremidade da associação, além de permitir especificar o nível de dependência de um objeto para com os outros envolvidos na associação.

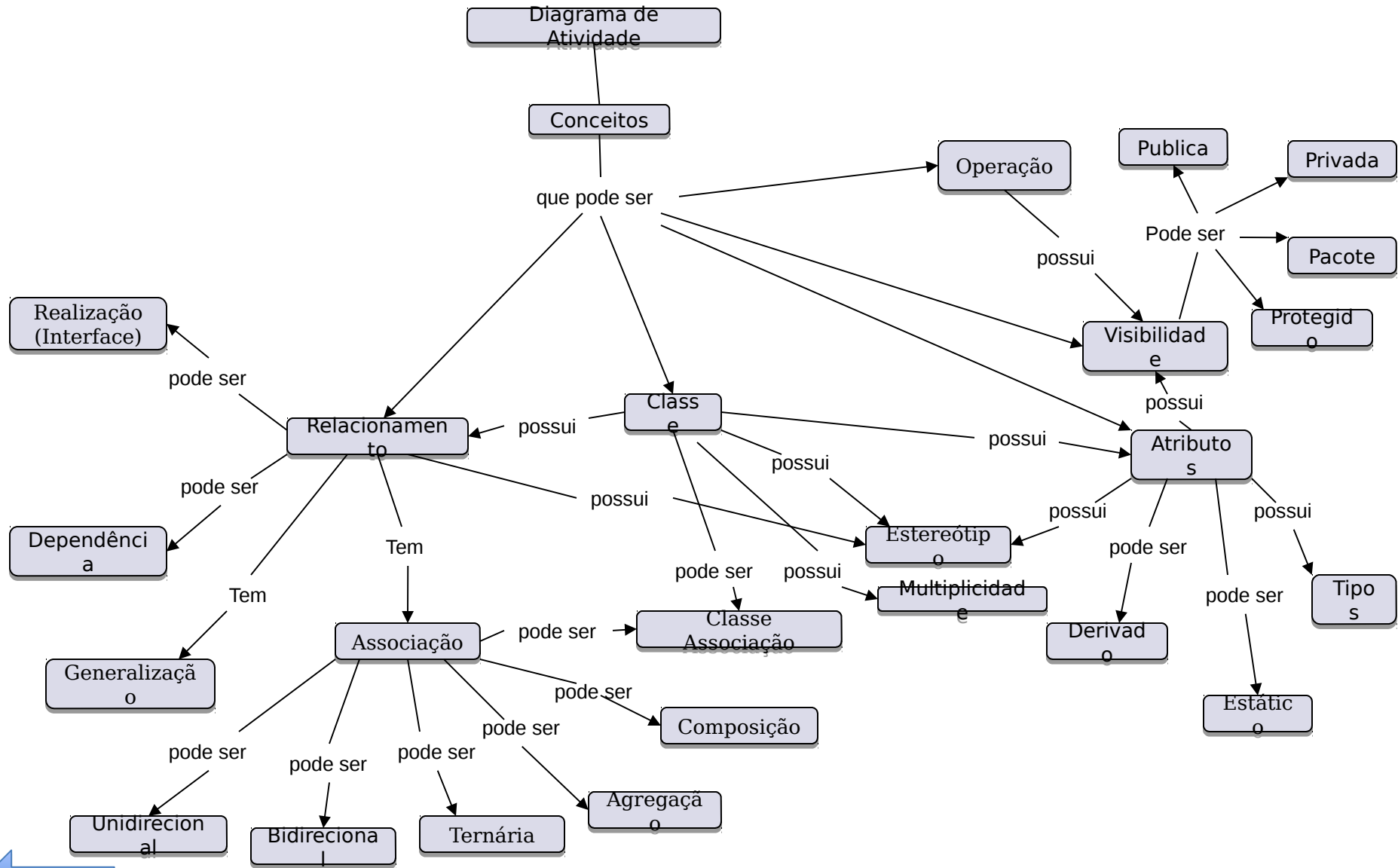


Elaborar o Diagrama de Classe

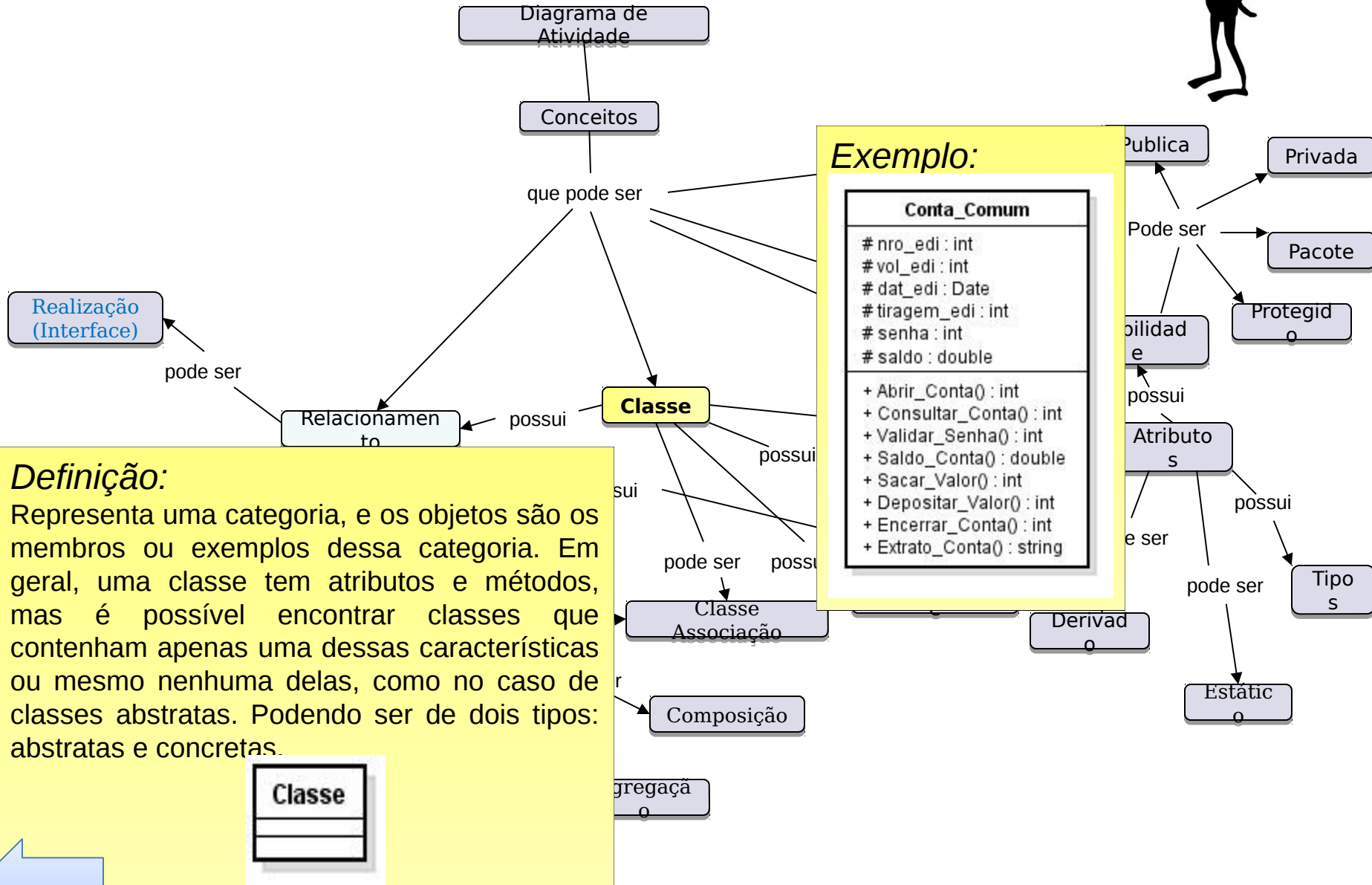
- Os diagramas de classes mostram a estrutura estática do modelo, principalmente os elementos existentes, como classes, sua estrutura interna e seus relacionamentos com outras classes. Eles não mostram informações temporárias.
- Um diagrama de classes é apresentado como um conjunto de elementos do modelo declarativos (estáticos) - como classes, pacotes e seus relacionamentos - que são conectados entre si e a seu conteúdo como um gráfico. Os diagramas de classe podem ser organizados em (e pertencentes a) pacotes, mostrando apenas o que é relevante em um determinado pacote.



Mapa Conceitual do Diagrama de Classe



Classe



Classe

- As **classes** são os blocos de construção mais importante de qualquer sistema orientado a objetos.
- É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos.

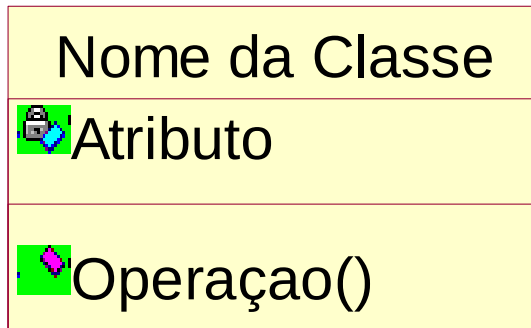
Classe

- Representa a abstração de um conjunto de OBJETOS do Mundo Real que possuem tipos de características e de comportamento em comum.

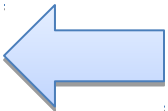


Classe Veículo

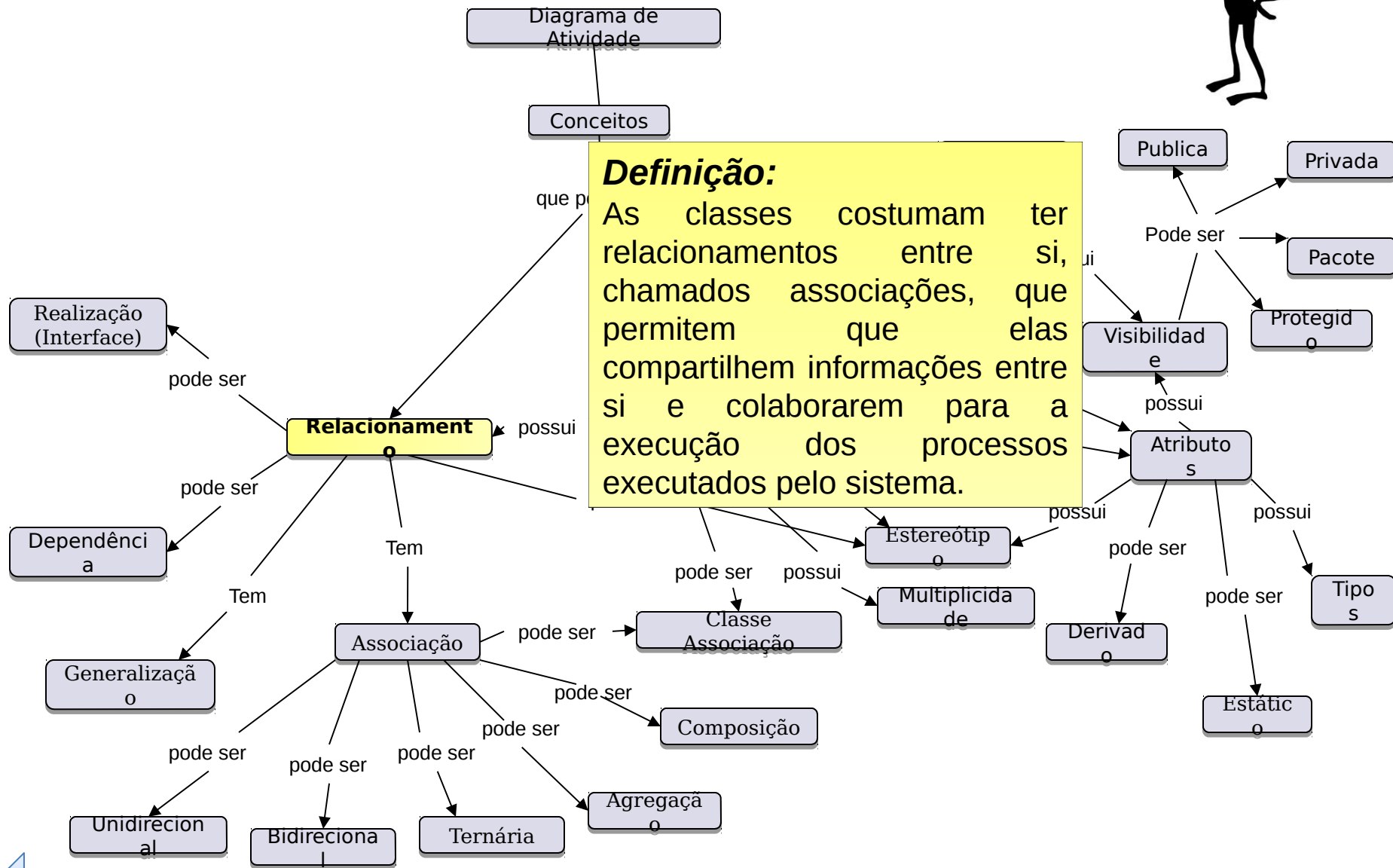
Notação da Classe



- A notação da classe é representada por um retângulo subdividido em 3 partes.
- Nome da Classe
- Atributo
- Operação

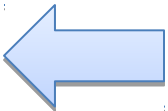


Relacionamento



Relacionamento

- Na UML, os modos pelos quais os itens podem estar conectados a outros, isto é, logicamente ou fisicamente, são modelados como relacionamentos, que permitem compartilhar informações e colaboram para a execução dos processos pelo sistema (GUEDES, 2005).
- Existem 4 tipos de relacionamentos:
 - Associação
 - Generalização
 - Dependência
 - Realização



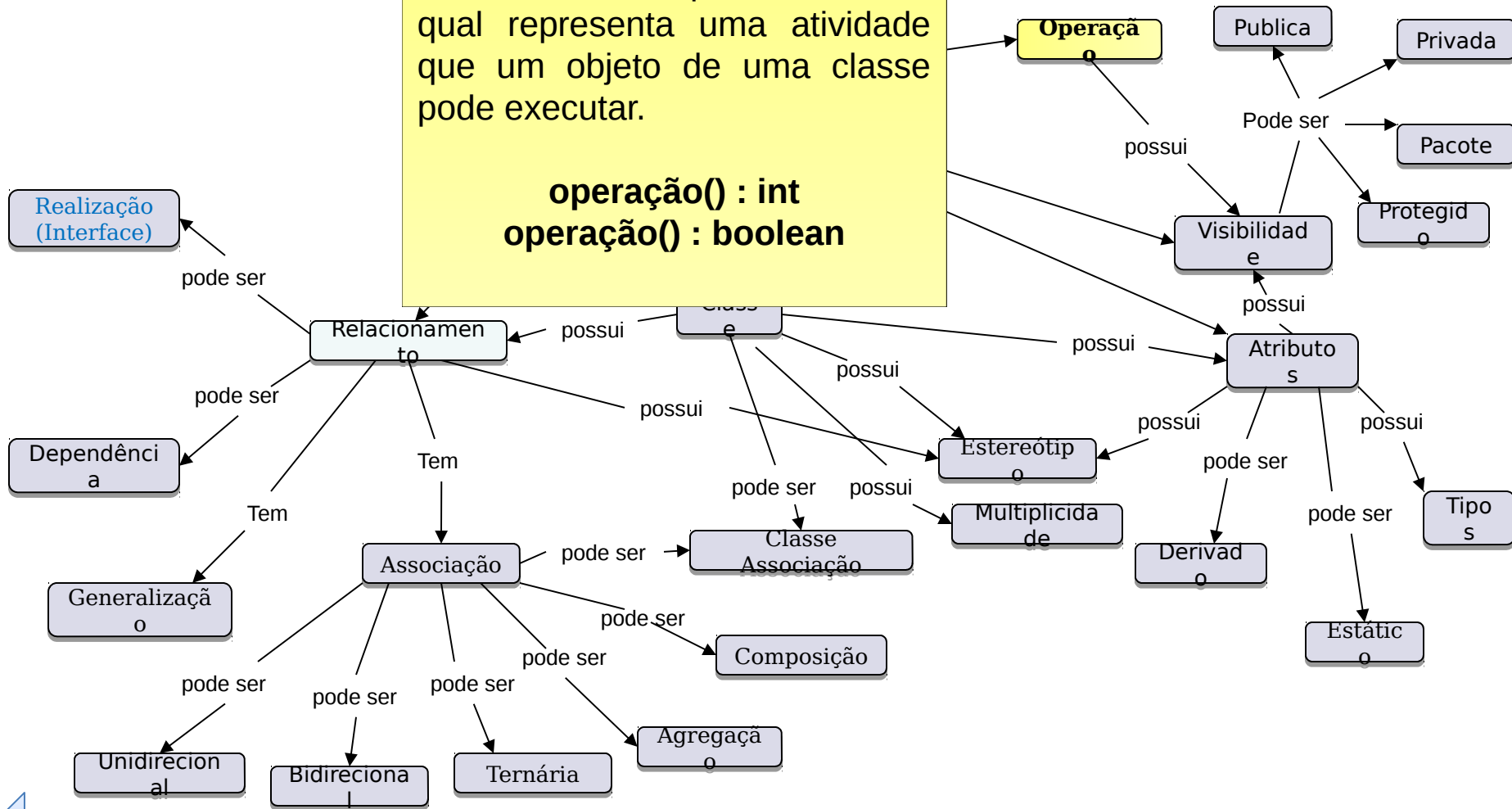
Operação



Definição:

Também conhecido como método ou comportamento da qual representa uma atividade que um objeto de uma classe pode executar.

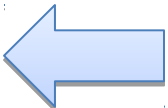
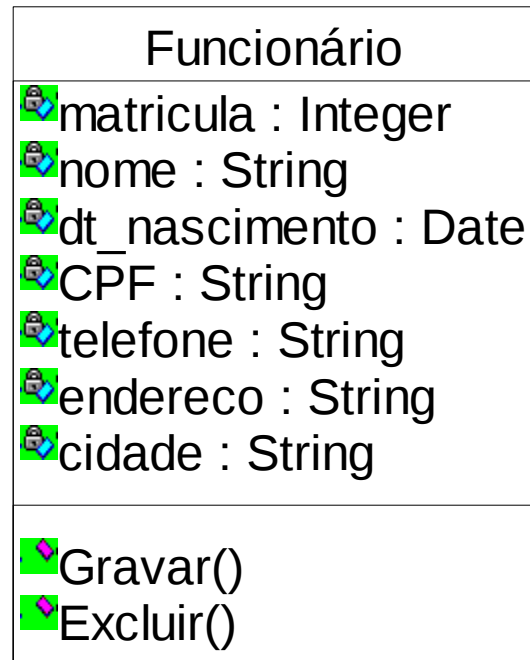
operação() : int
operação() : boolean



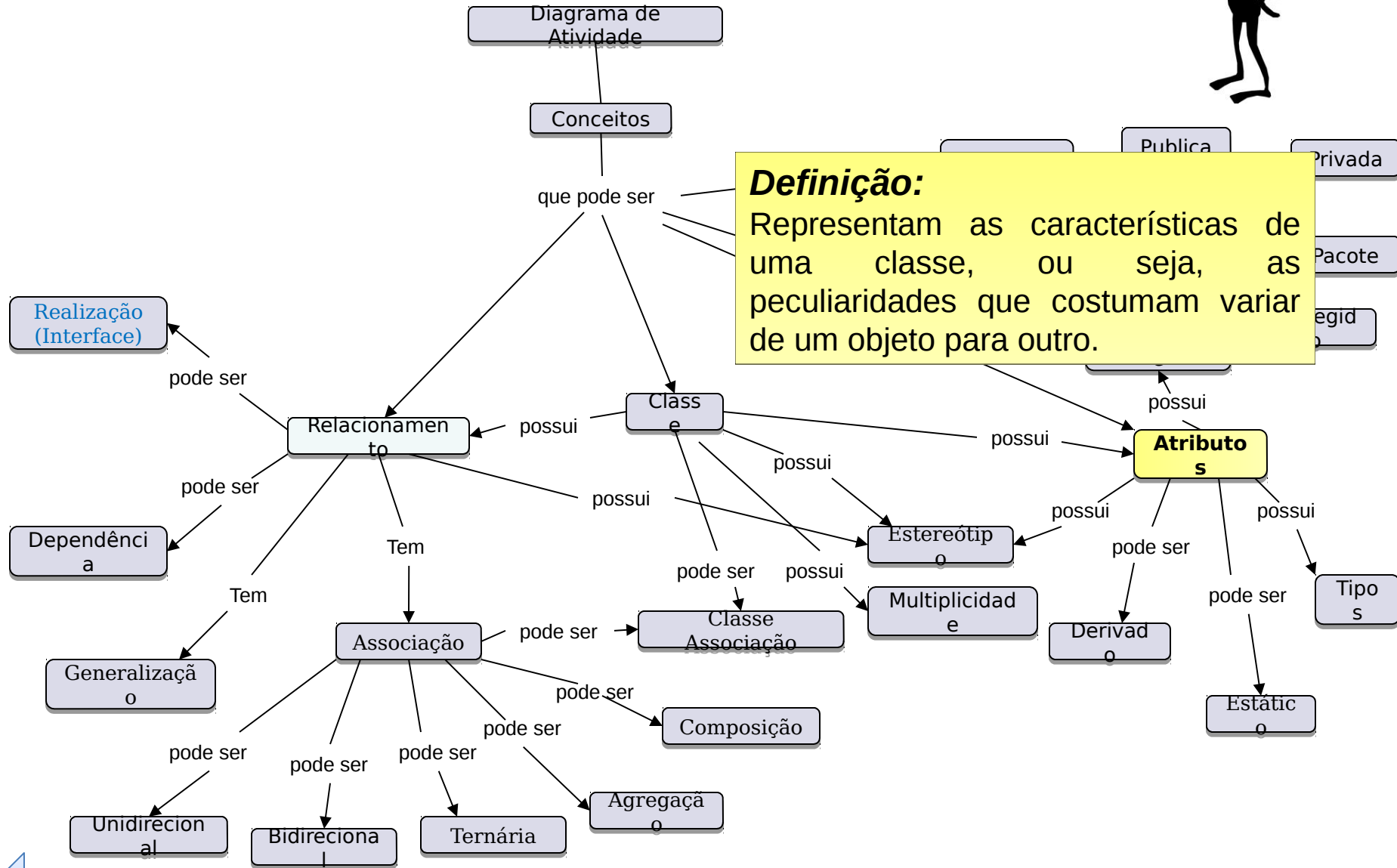
Operação

- Uma ação que o objeto executa, é uma ordem que faz o objeto a agir.
- A implementação de uma operação é chamada de **método**.

Operação



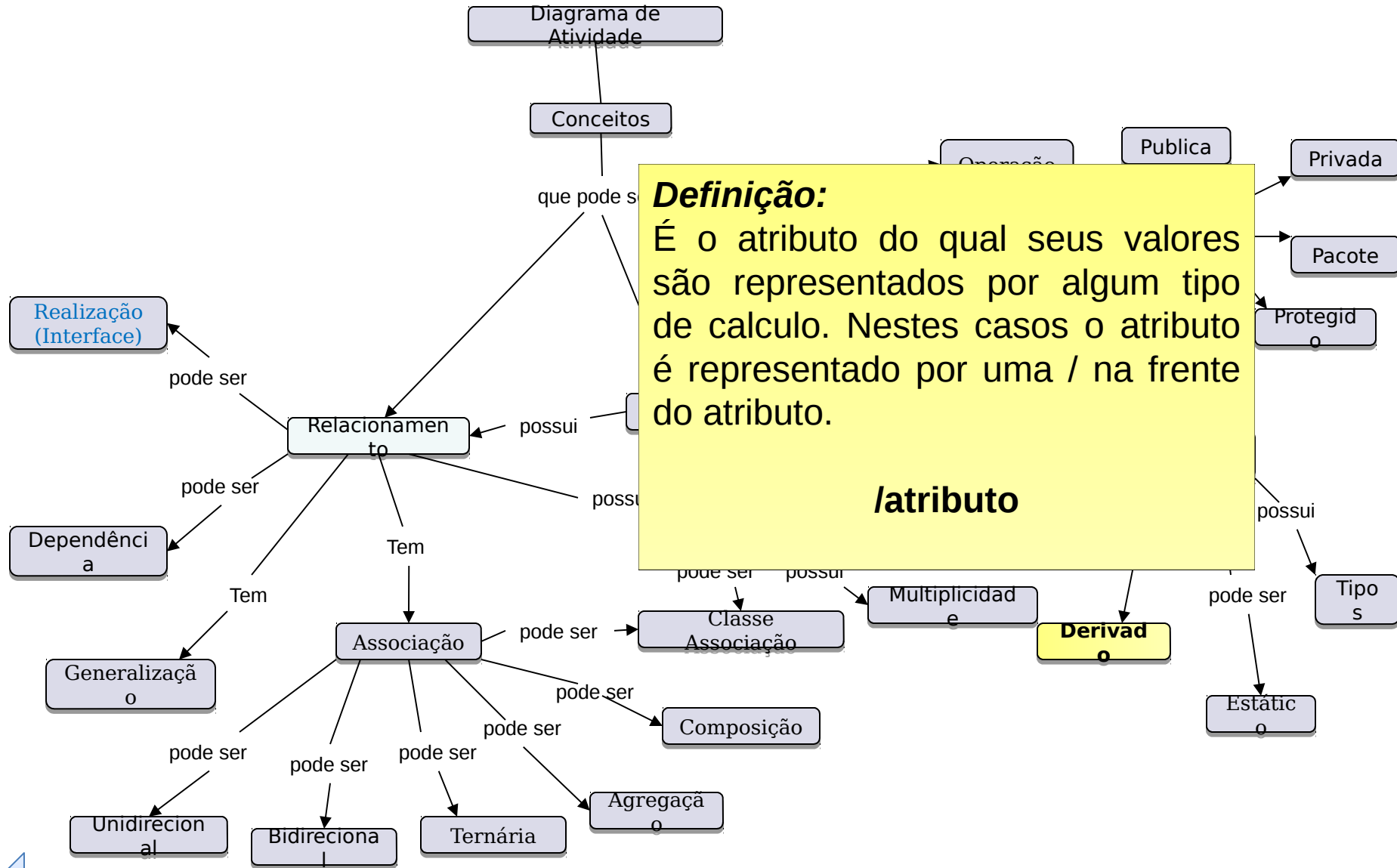
Atributos



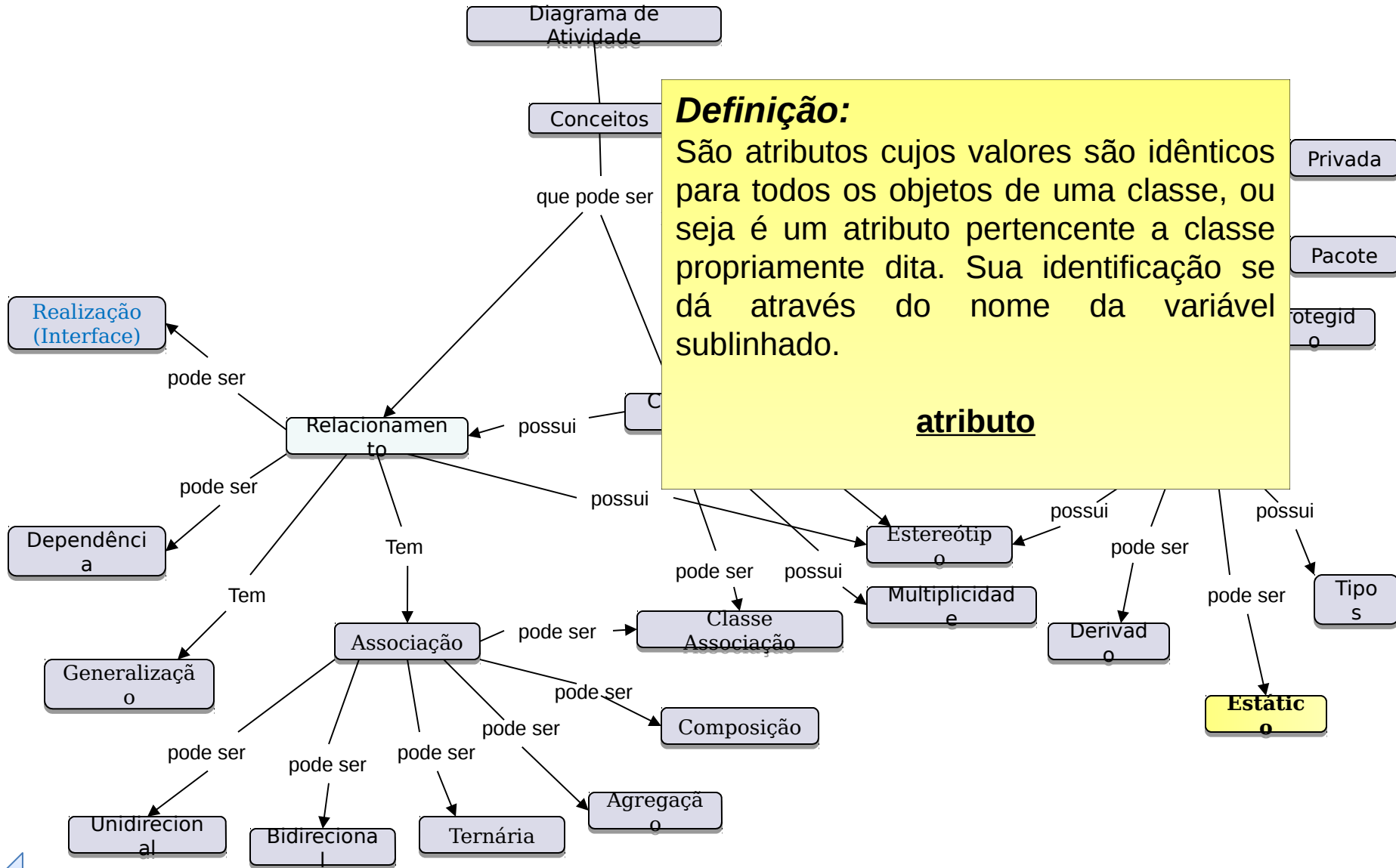
Notação do Atributo

- Esses elementos normalmente são expressos em uma única string de texto, utilizando a sintaxe:
- [visibilidade] [/] nome [: tipo]
- A maioria das ferramentas trazem essas informações em campos separados, tornando assim mais fácil a visualização.

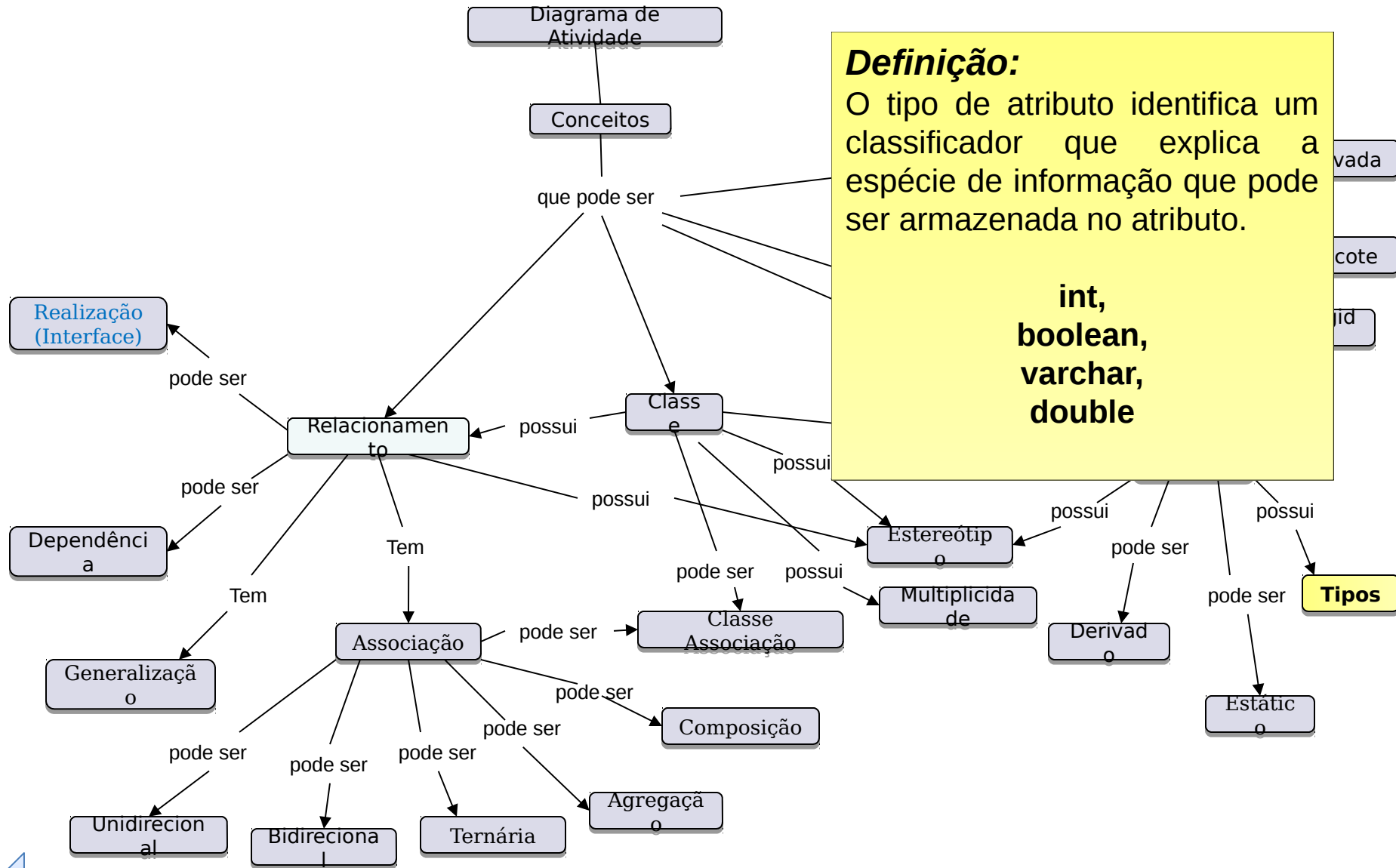
Derivado



Estático



Tipos



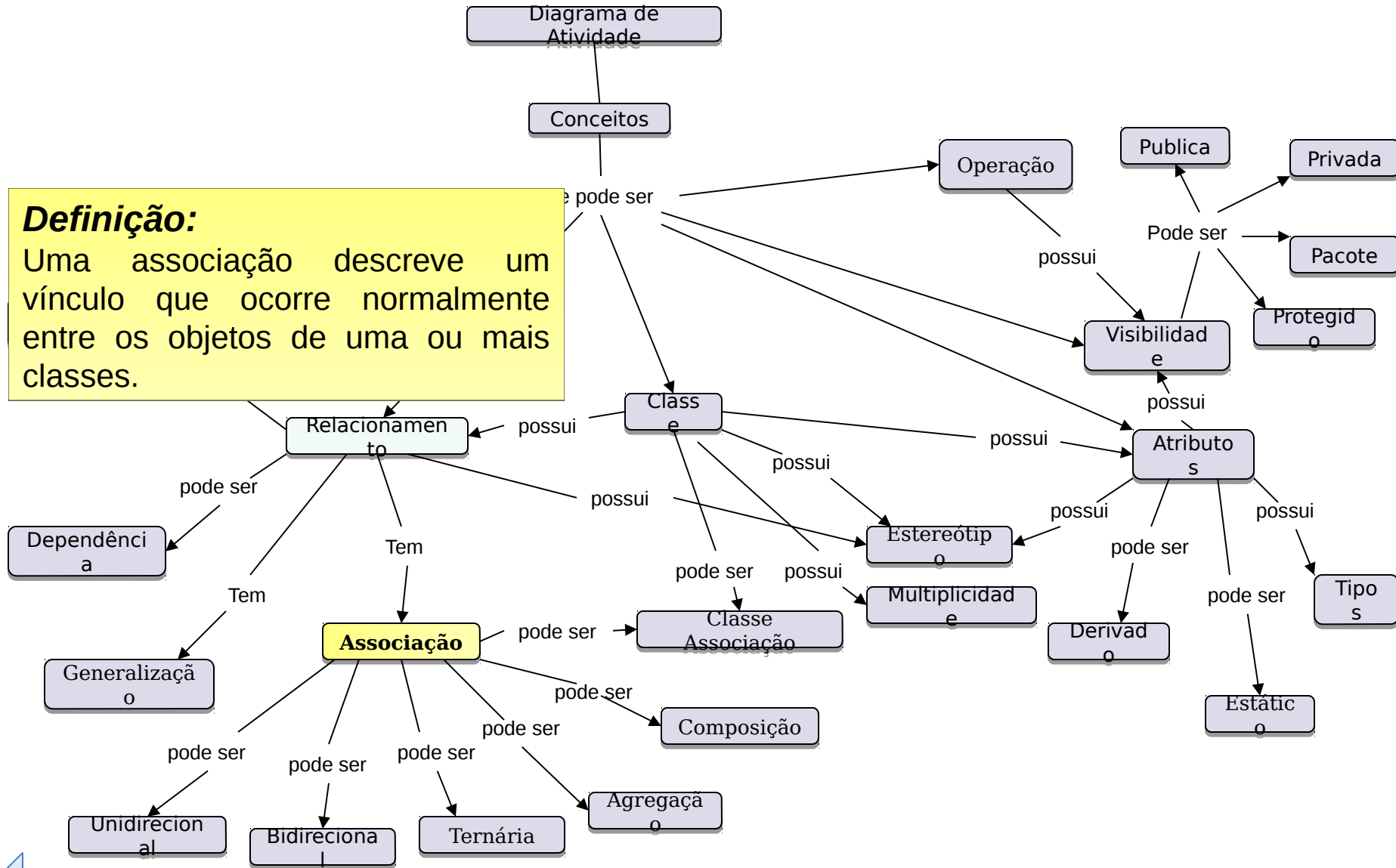
vada

cote

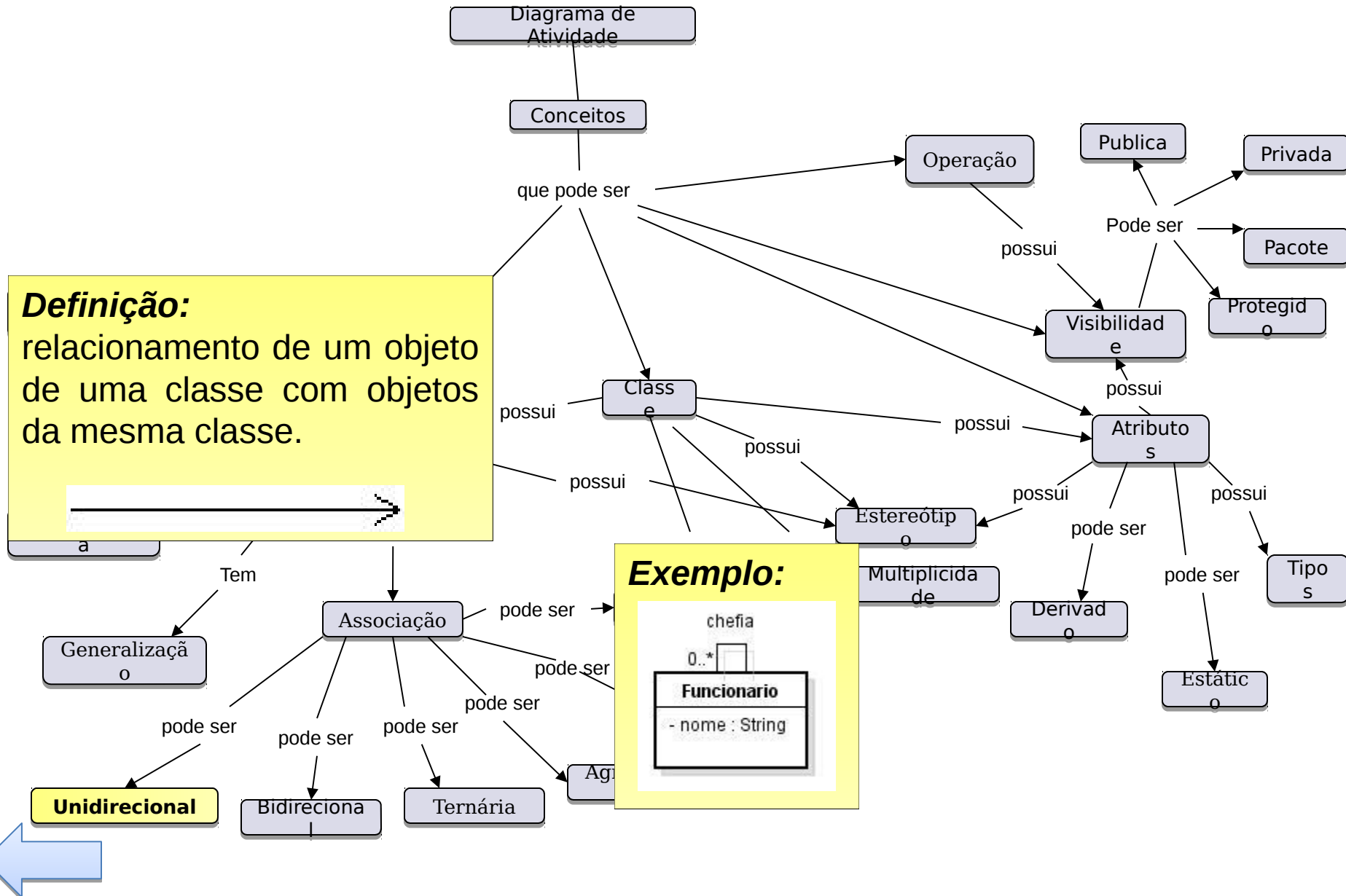
id



Associação



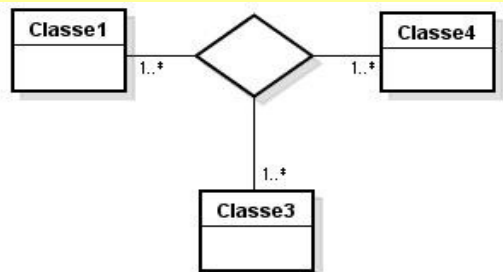
Unidirecional



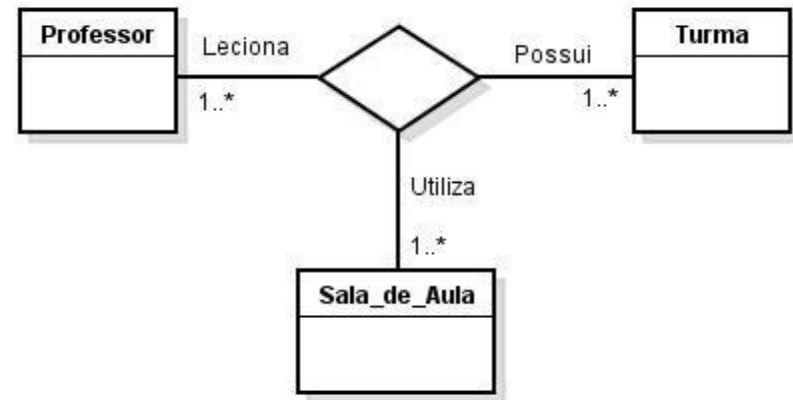
Ternária

Definição:

São associações que conectam objetos de mais de duas classes. São representadas por um losango para onde convergem todas as ligações da associação.



Exemplo:



Generalização

Tem

Associação

pode ser

pode ser

pode ser

pode ser

pode ser

pode ser

Unidirecional

Bidirecional

Ternária

Agregação

Operação

Publica

Privada

Pode ser

Pacote

Protegido

possui

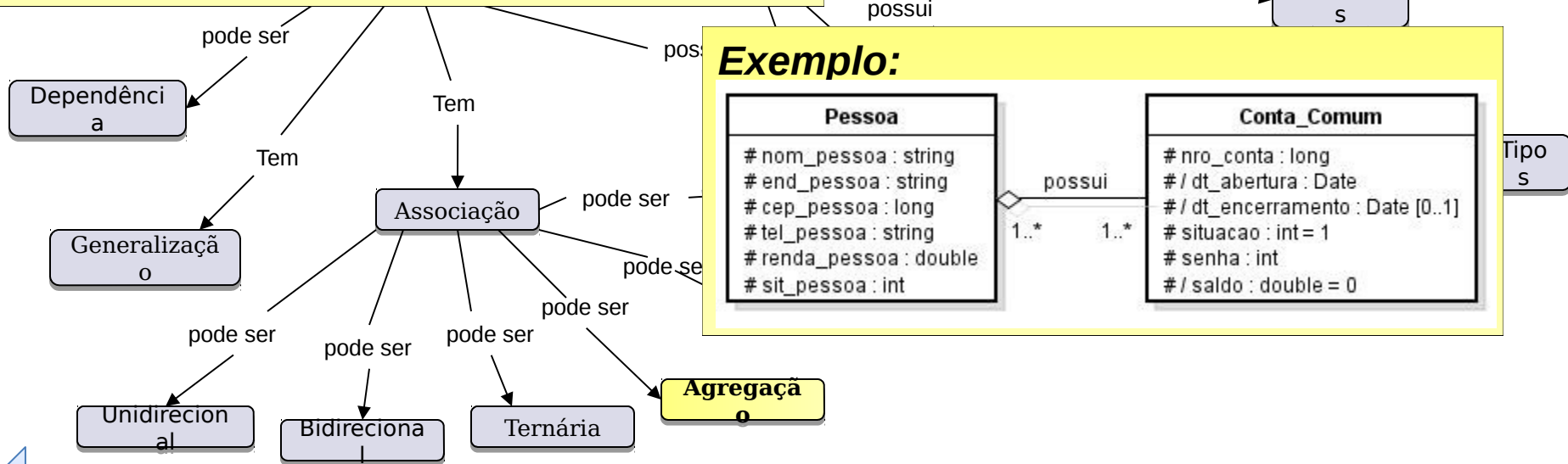
Visibilidade

possui

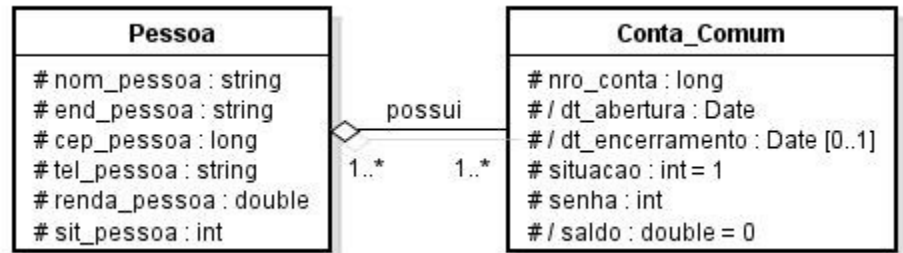
Agregação

Definição:

É um tipo especial de associação onde se tenta demonstrar que as informações de um objeto (chamado objeto-todo) precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe (chamados objetos-parte). Esse tipo de associação tenta demonstrar uma relação todo/parte entre os objetos associados.



Exemplo:



Tipo
s

Composição

Diagrama de Atividade

Conceitos

que pode ser

Definição:

É um tipo de composição da qual constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de estar associados a um único objeto-todo.



Exemplo:

Revista_Cientifica

- tit_rev : String
- issn_rev : long
- periodicidade : String

publica

1..*

Edicao

- nro_edi : int
- vol_edi : int
- dat_edi : Date
- tiragem_edi : int

Contém

6..10

Artigo

- tit_art : String

Estático

Generalizaçã
o

Associação

pode ser

pode ser

pode ser

pode ser

pode ser

pode ser

Composição

Unidirecion
al

Bidireciona

Ternária

Agregaçã
o

ivada

acote

gid

Tipo
s



Classe Associação

Diagrama de
Atividade

Conceitos

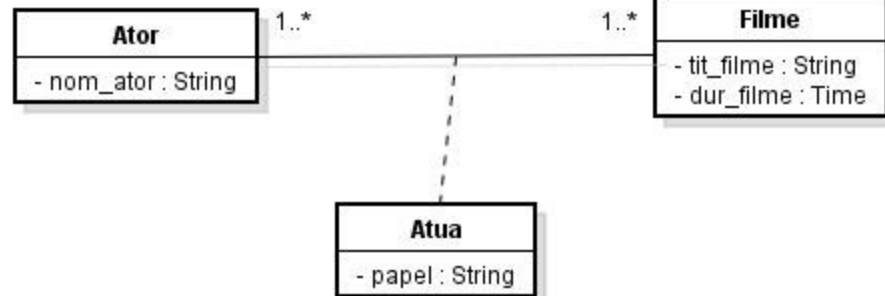
Operação

Publica

Privada

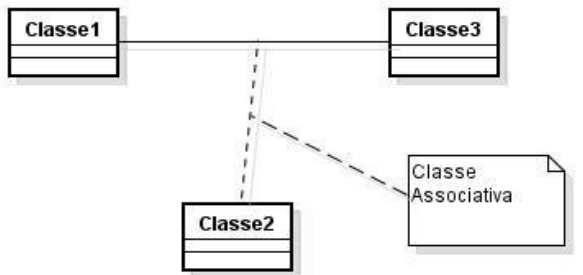
que pode ser

Exemplo:



Definição:

são aquelas produzidas quando da ocorrência de associações que tenham multiplicidade muitos (*) em todas as suas extremidades.



pode ser

possui

ser

pode ser

Multiplicidade

Derivada

pode ser

Estática

Tipos

Classe Associação

Composição

pode ser

pode ser

pode ser

Agregação

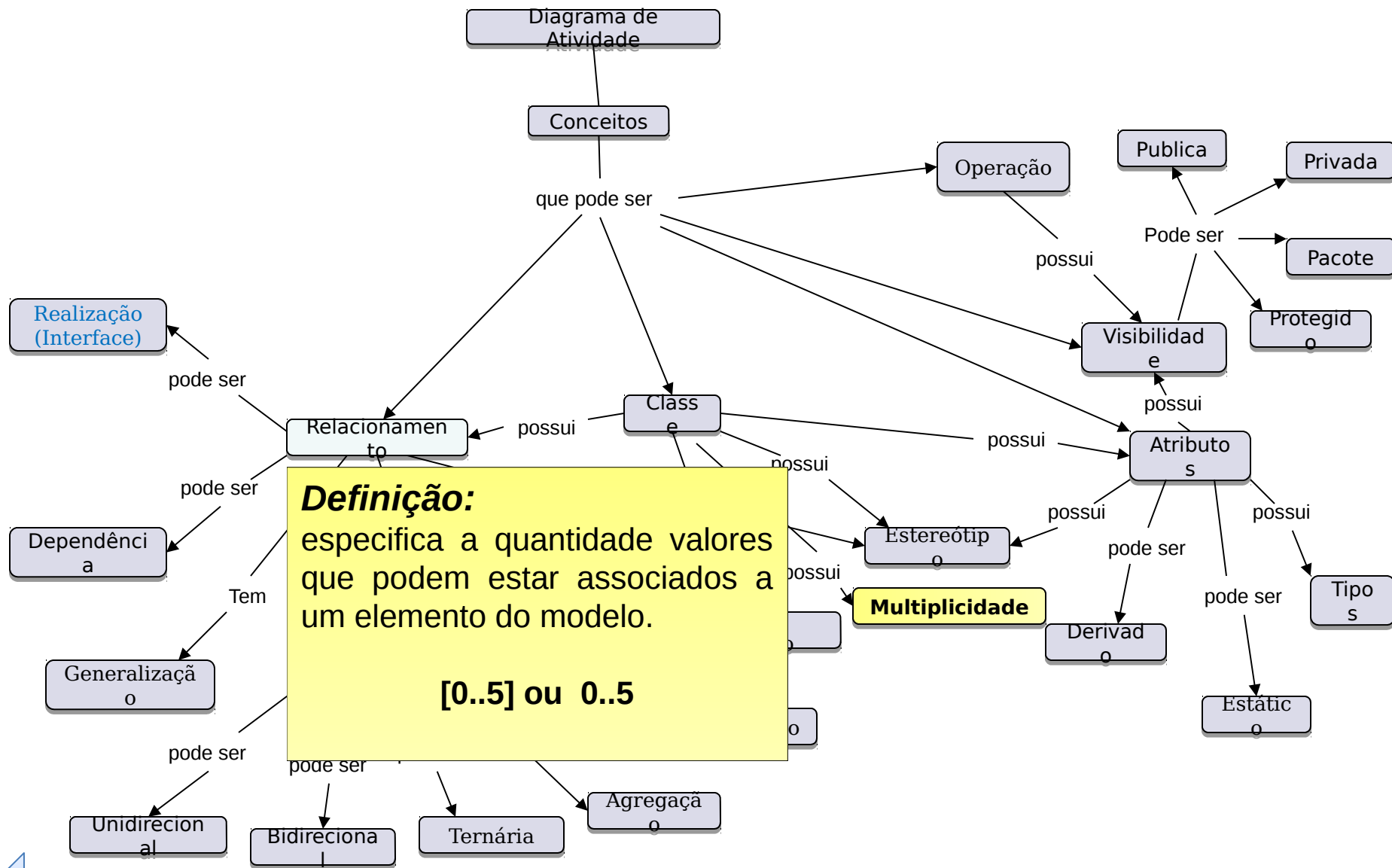
Unidirecional

Bidirecional

Ternária



Multiplicidade



Realização (Interface)

Diagrama de Atividade

Definição:

É um tipo de relacionamento especial que mistura características dos relacionamentos de generalização e dependência, sendo usada para identificar classes responsáveis por executar funções para outras classes.

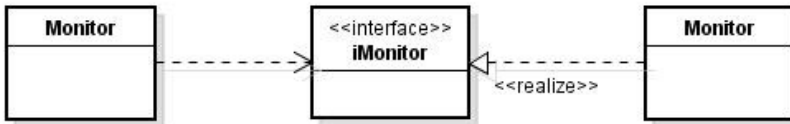
Realização (Interface)

pode ser

pode ser

possui

Exemplo:



pode ser

pode ser

pode ser

Unidirecional

Bidirecional

Ternária

Agregação

pode ser

possui

Classe Associação

Composição

Estereótipo

Multiplicidade

Derivada

Estático

possui

pode ser

pode ser

possui

possui

pode ser

possui

pode ser

Publica

Privada

Pacote

Protegido

possui

pode ser

possui

pode ser

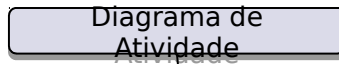
pode ser

pode ser

pode ser

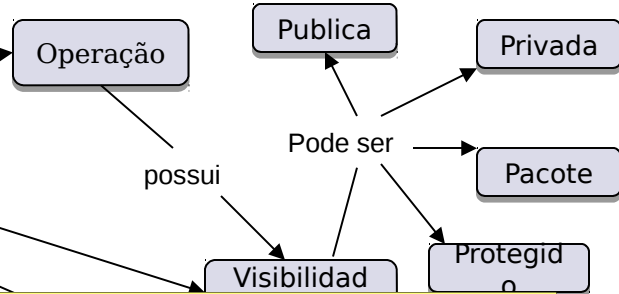
pode ser

Generalização

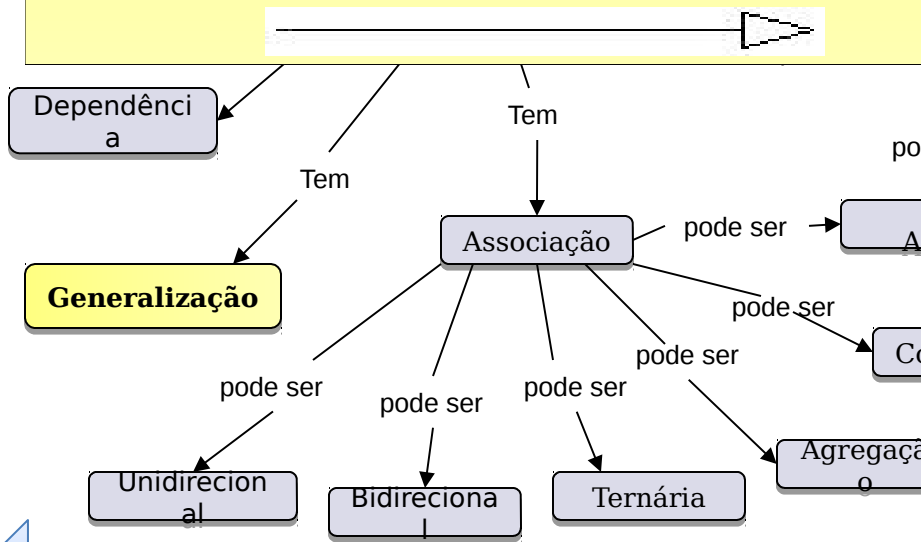
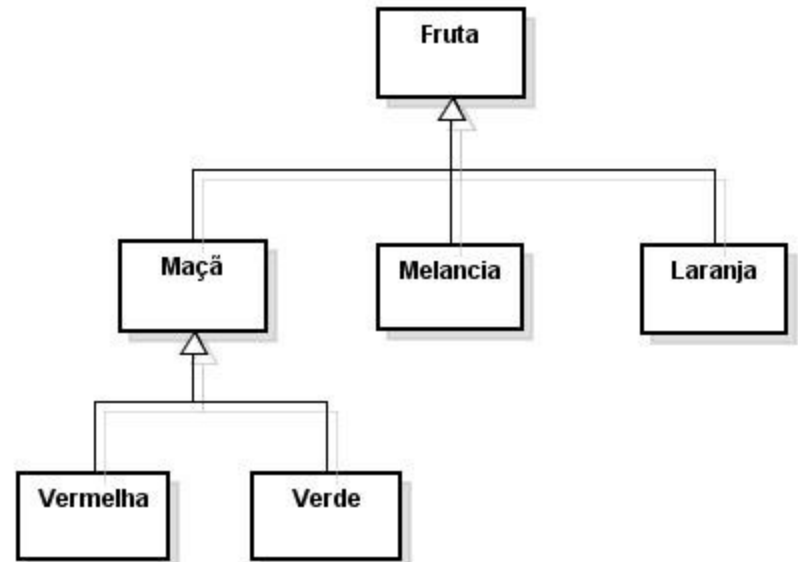


Definição:

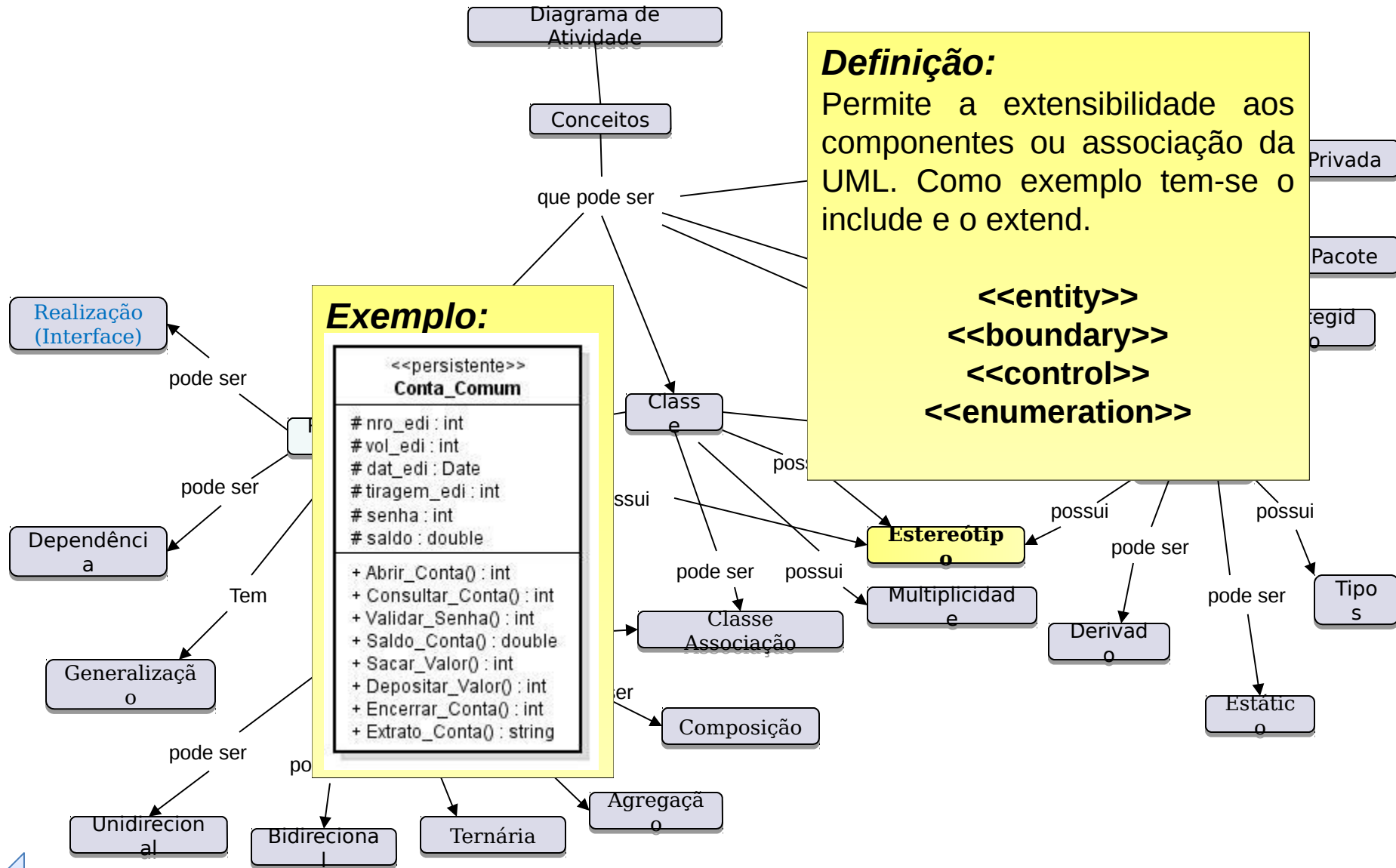
O objetivo dessa associação é representar a ocorrência de herança entre as classes, identificando as superclasses, chamadas gerais e subclasses, chamadas especializadas, demonstrando a hierarquia entre as classes e possivelmente métodos polimórficos nas classes especializadas.



Exemplo:



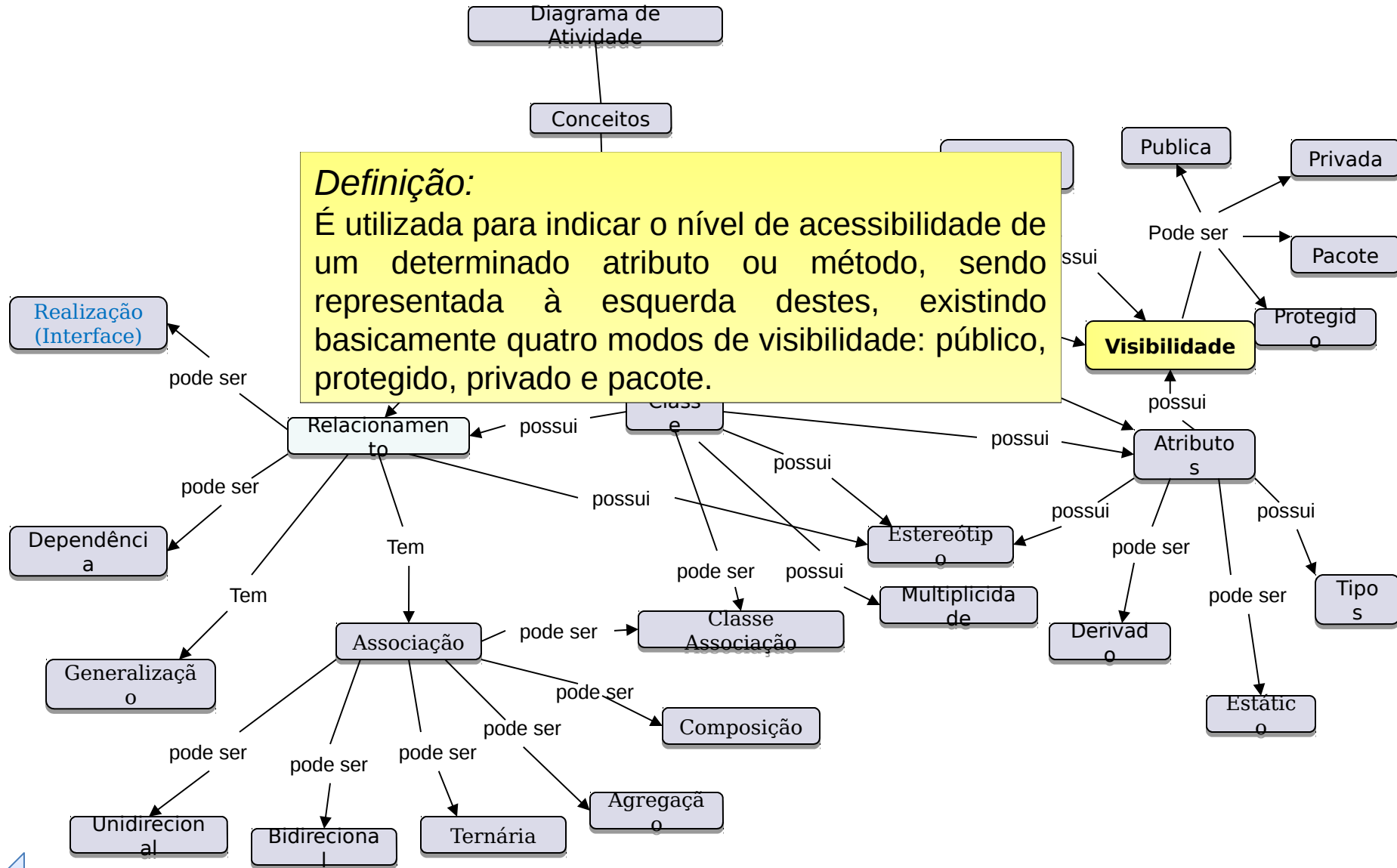
Estereótipo



Visibilidade

Definição:

É utilizada para indicar o nível de acessibilidade de um determinado atributo ou método, sendo representada à esquerda destes, existindo basicamente quatro modos de visibilidade: público, protegido, privado e pacote.

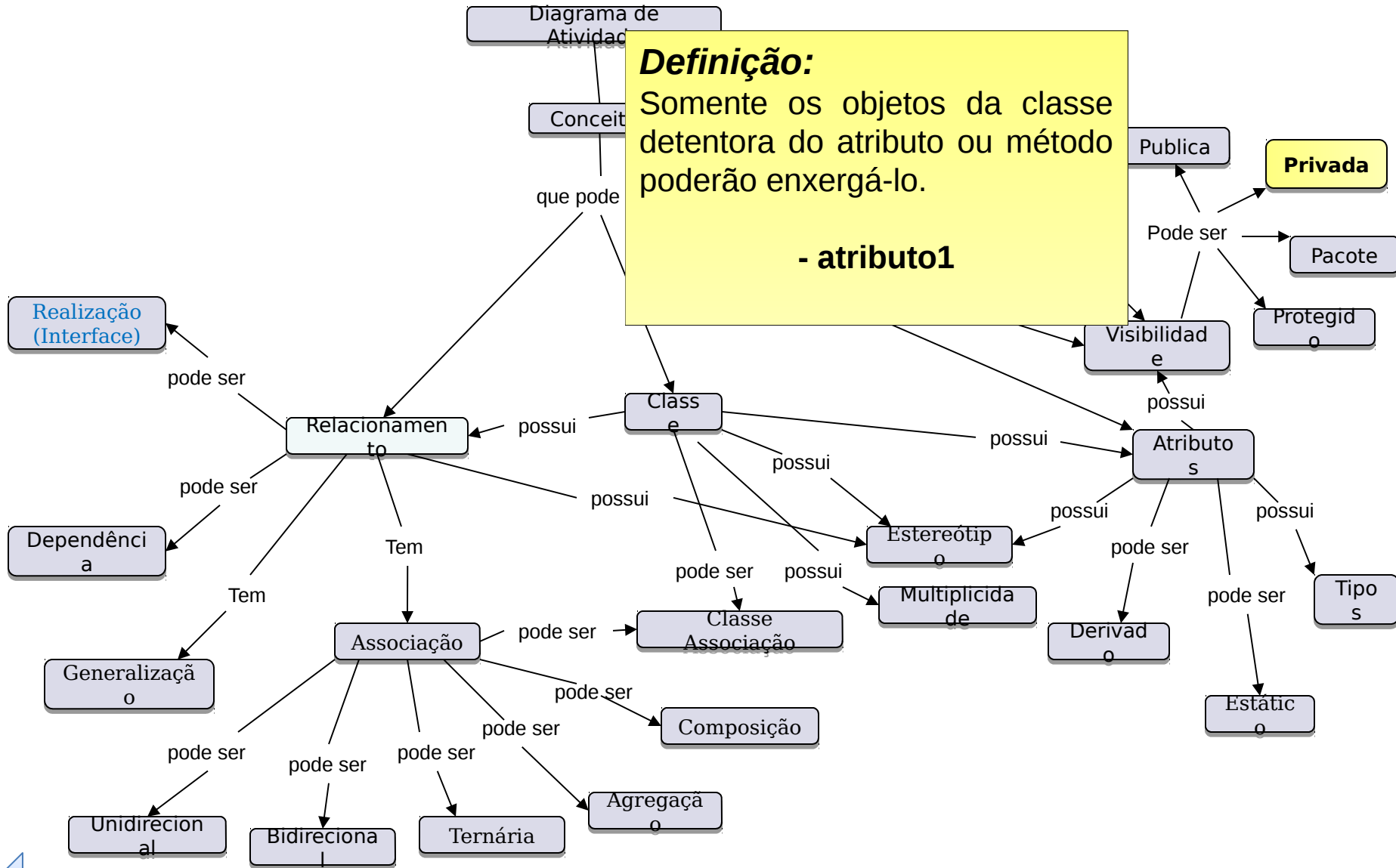


Definição:

+ atributo3



Privada



Determina que o atributo ou método é visível por qualquer objeto dentro do pacote. Somente elementos que fazem parte de um pacote podem ter essa visibilidade. Nenhum elemento fora do pacote poderá ter acesso a um atributo ou método com essa visibilidade.

```

classDiagram
    class Publica
    class Privada
    class Pacote
    class Protegido
    class Atributos
    class Tipos
    class Estatico

    Publica --|> Privada
    Publica --|> Pacote
    Publica --|> Protegido
    Atributos --|> Tipos
    Atributos --|> Estatico
    Atributos --> Pacote : possui
    Atributos --> Tipos : possui
    Atributos --> Estatico : pode ser
    
```



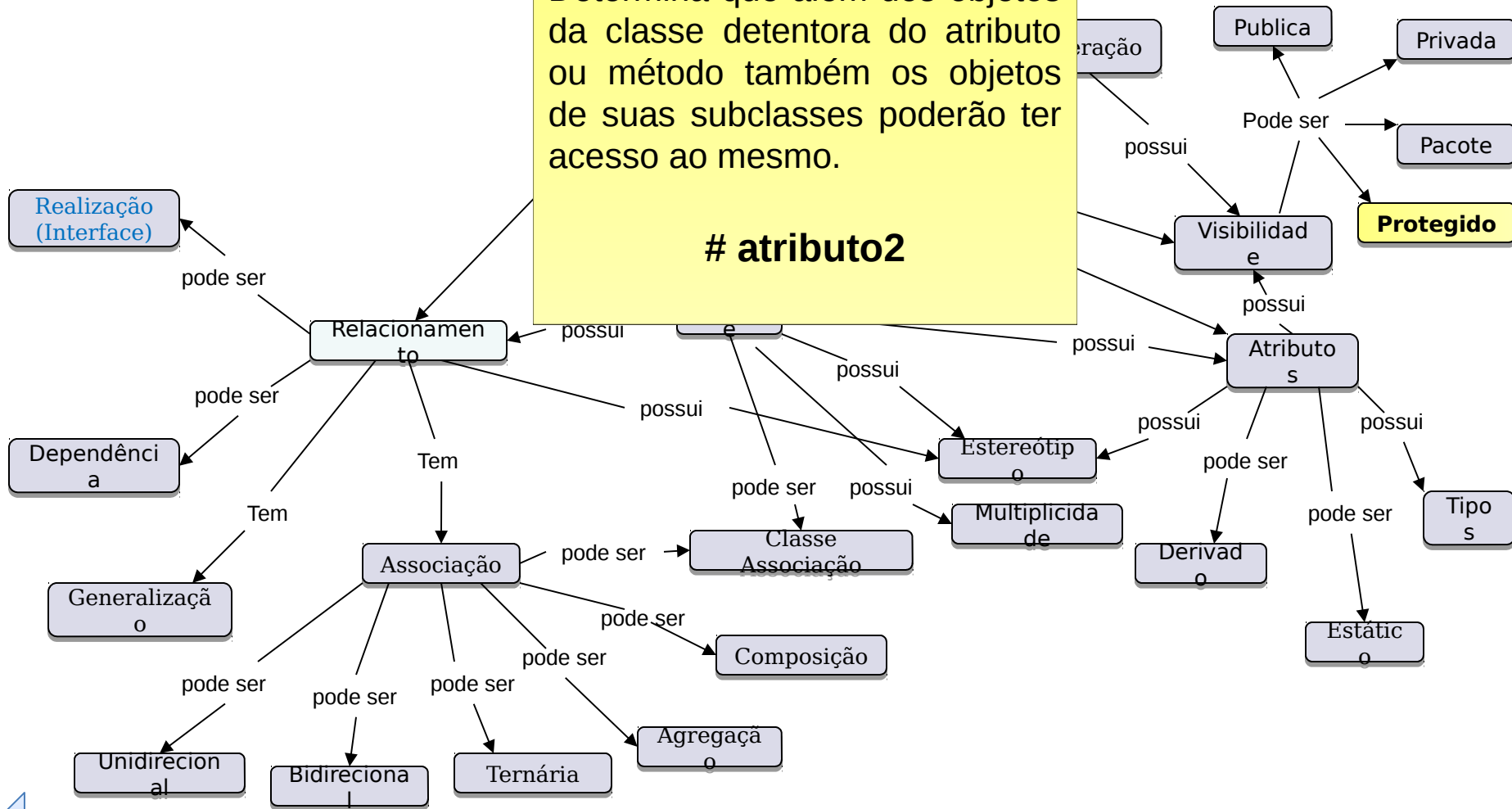
Protegido

Diagrama de
Atividade

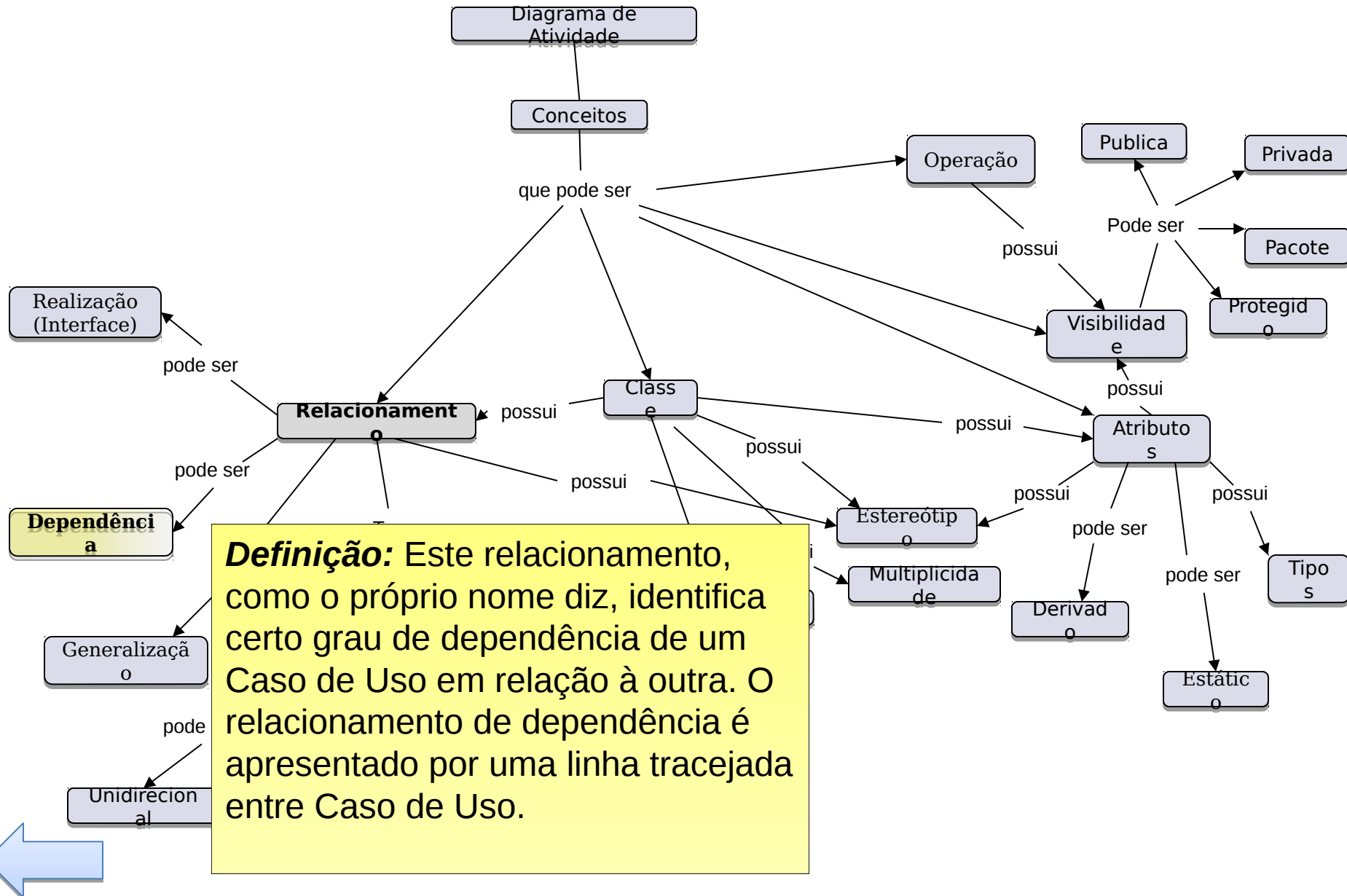
Definição:

Determina que além dos objetos da classe detentora do atributo ou método também os objetos de suas subclasses poderão ter acesso ao mesmo.

atributo2

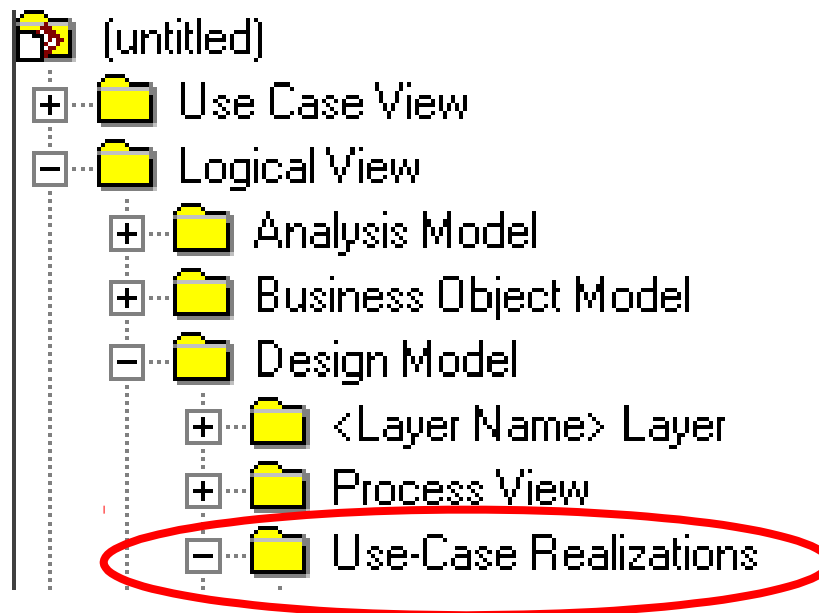


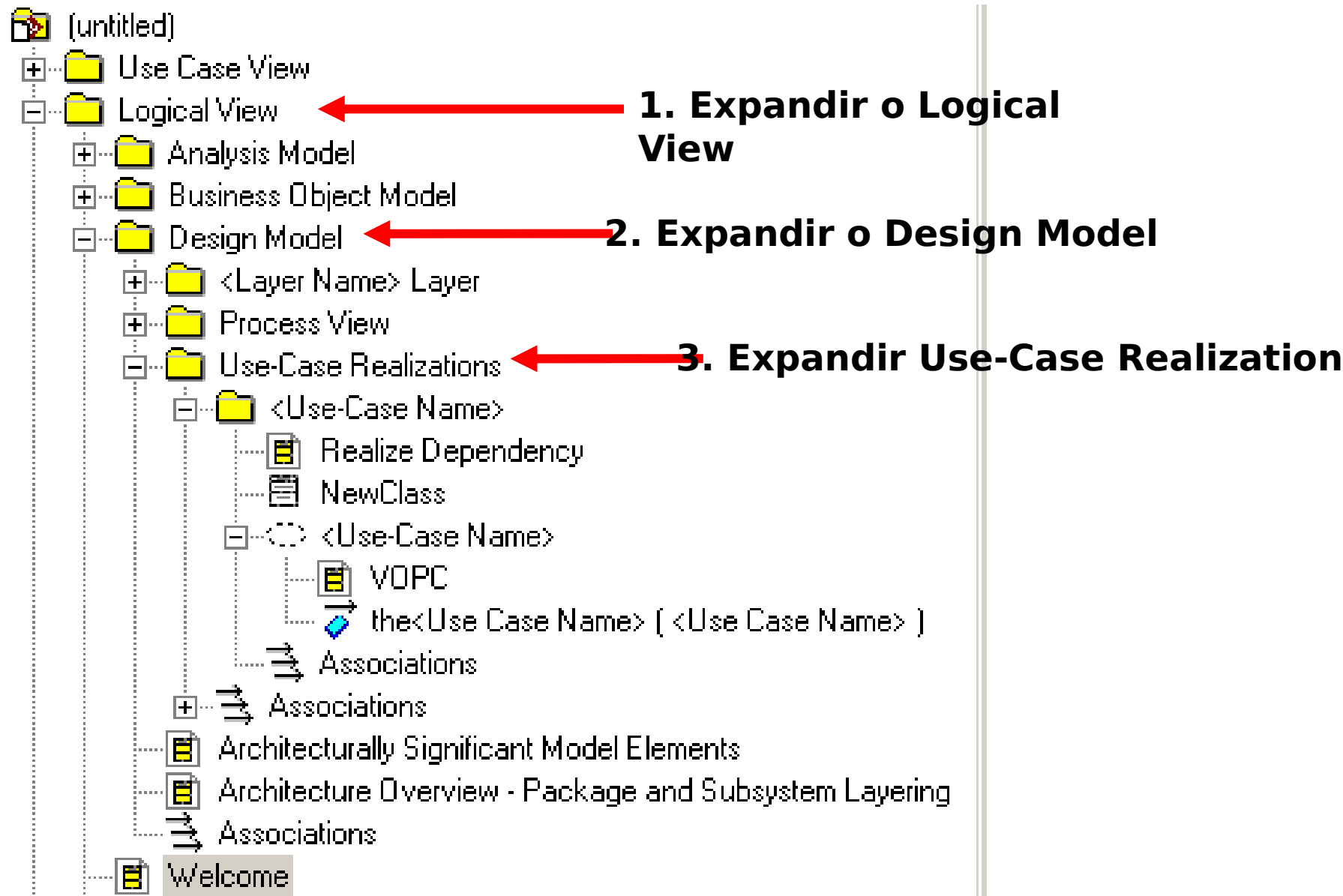
Dependência



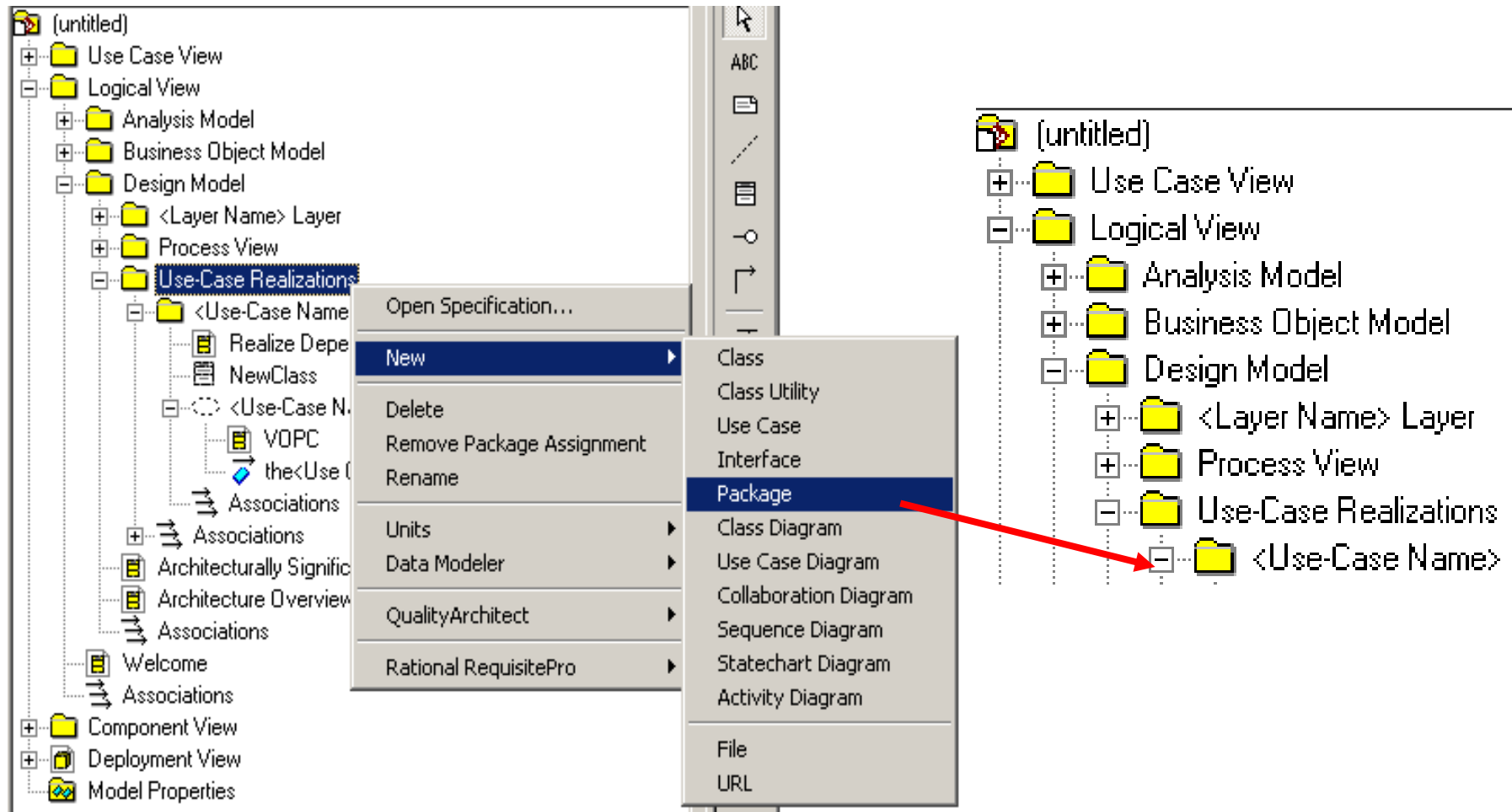
Onde fazer o Diagrama de Classe no Rational Rose?

- O Diagrama de Classe deve ser feito no Logical View □ Design Model
- Dentro do pacote Use-Case Realization

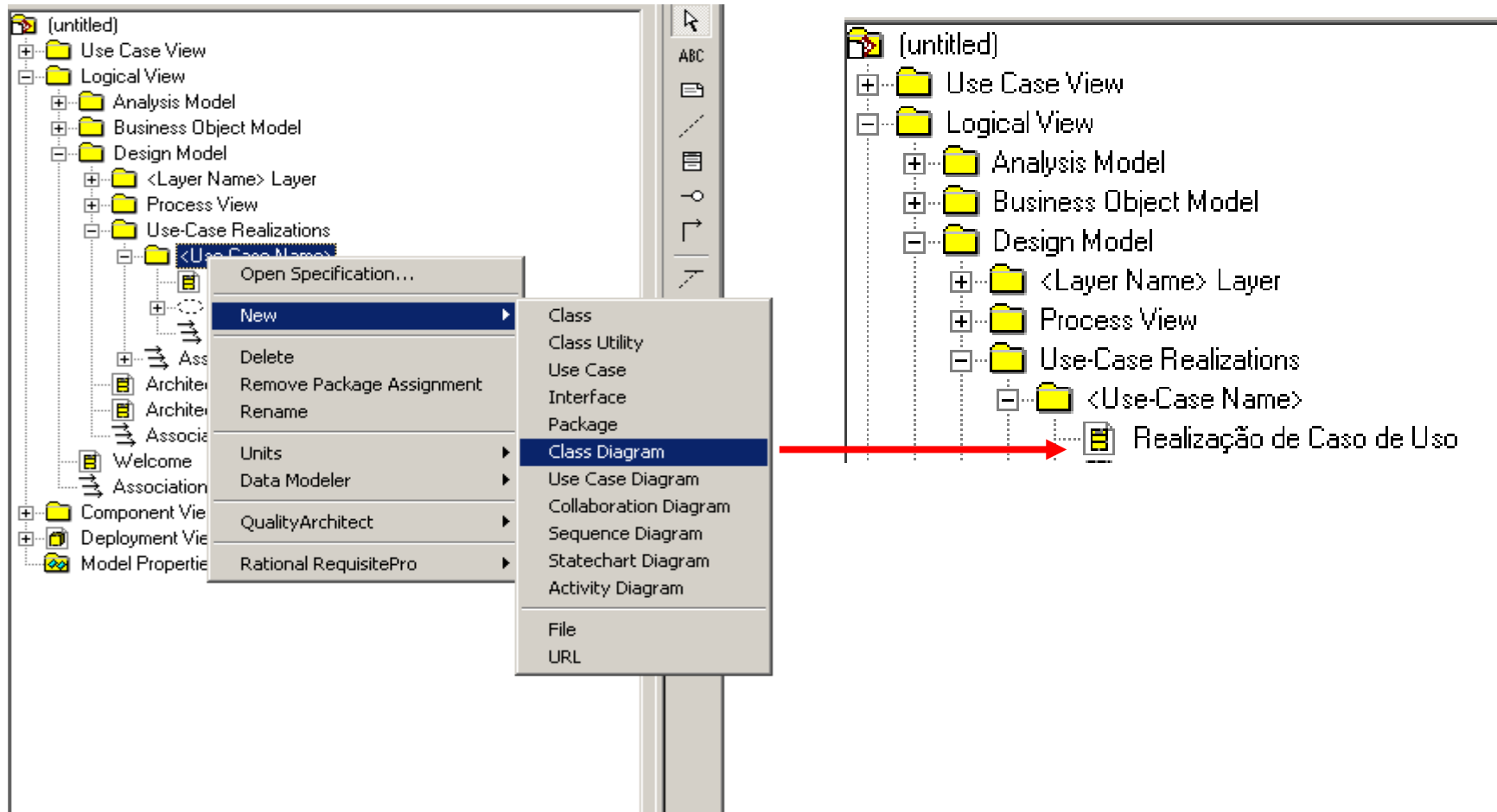




4. Criar o pacote com o nome do Caso de Uso



5. Fazer a realização de caso de uso



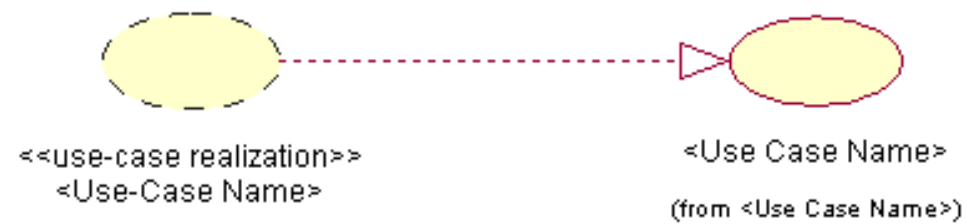
6. Clique duplo na Realização do Caso de uso, para fazer a realização

6.1. Arrastar o caso de uso do Use Case View

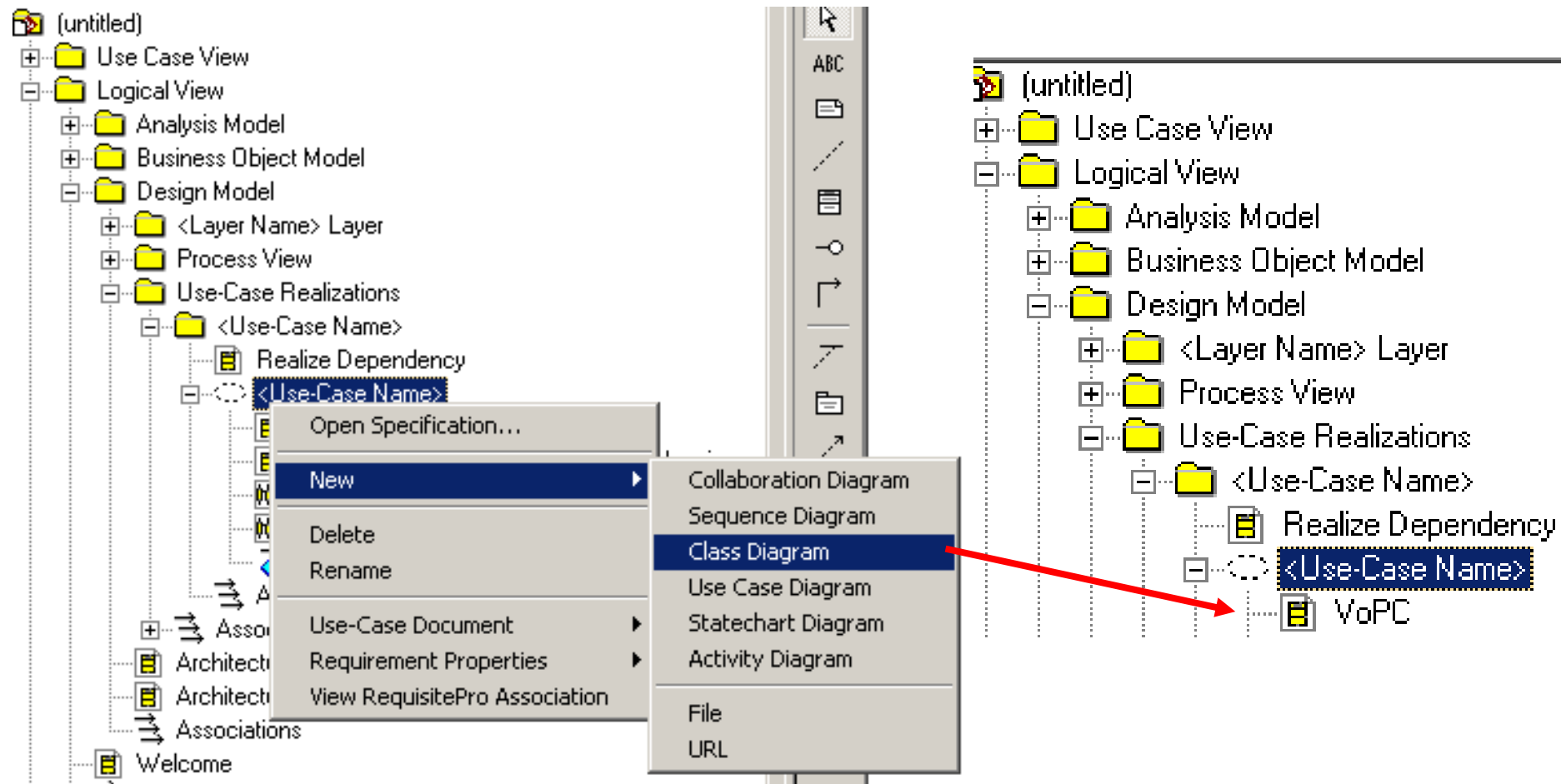
6.2. Arrastar o elemento do use case realization

6.3. Fazer a associação entre eles utilizando o Realize





7. VoPC – View of Participating Class



8. Clique duplo no VoPC, para a criação da classe

8.1. Arrastar o elemento que representa a classe



8.2. Inserir o nome da classe, atributos e operações

8.3. Caso necessário, efetuar a associação entre as classes.



- Após realizado todos os VoPC's para as classes, deverá arrastá-las para o Welcome e efetuar as associações entre elas.

