

Diagrama de Classes

- Mostra a existência de classes e suas relações com a visão lógica do sistema
- Elementos de UML presentes nos Diagramas de Classes:
 - Classes, suas estruturas internas (**atributos**) e comportamento (**métodos**)
 - Relacionamentos (associações, agregações, dependências e generalização/especificação)
 - Adornos (multiplicidade, indicadores de navegação, visibilidade etc)

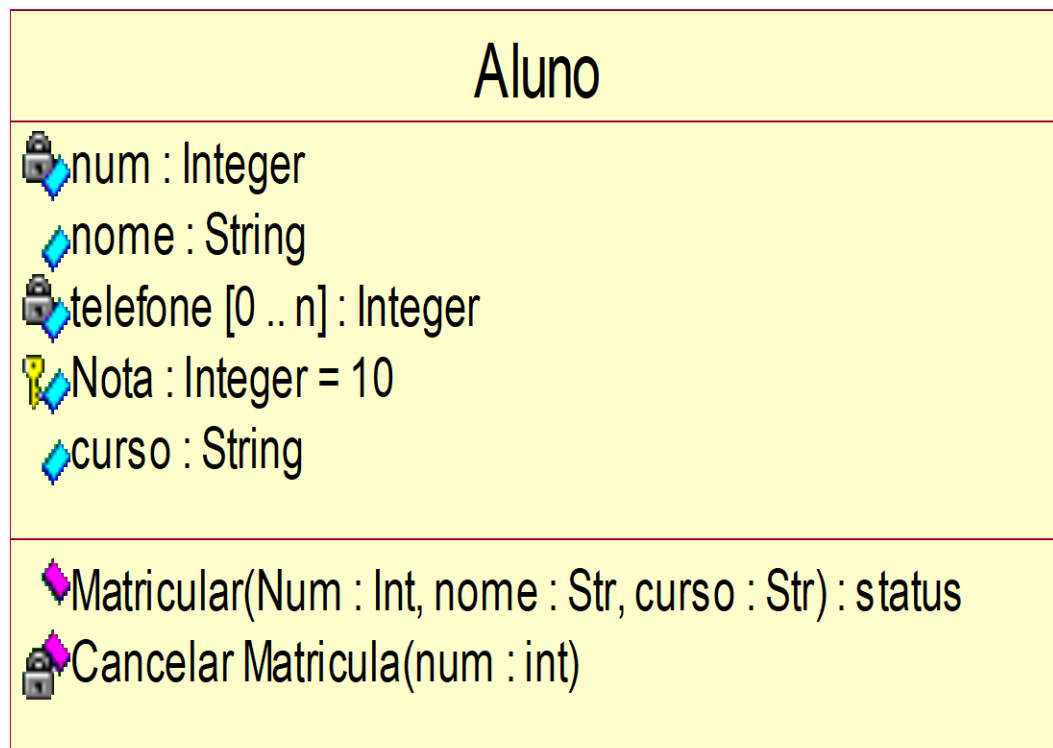
Atributos e Métodos

■ Partes do Atributo:

- ☐ Visibilidade
- ☐ Nome
- ☐ Valor inicial
- ☐ Descrição
- ☐ Tipo de dado
- ☐ Multiplicidade

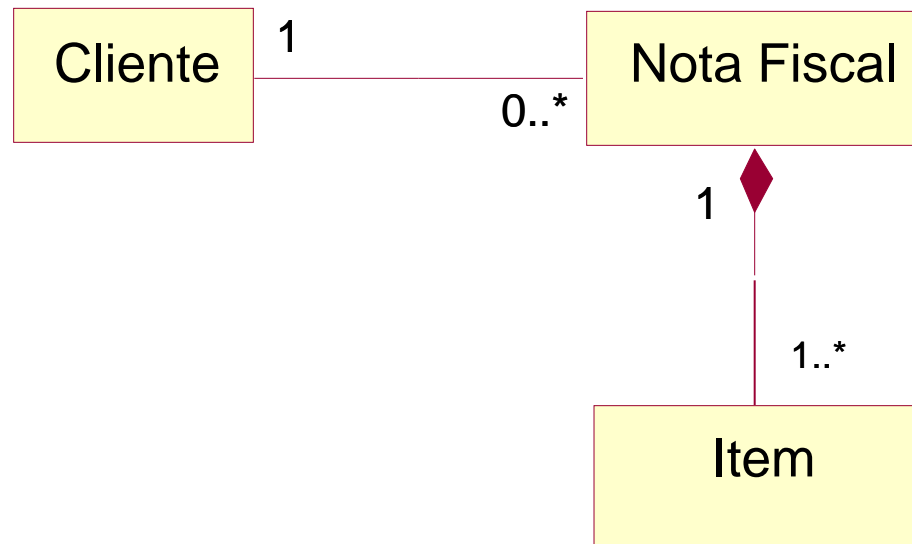
■ Partes do Método:

- ☐ Visibilidade
- ☐ Nome
- ☐ Assinatura



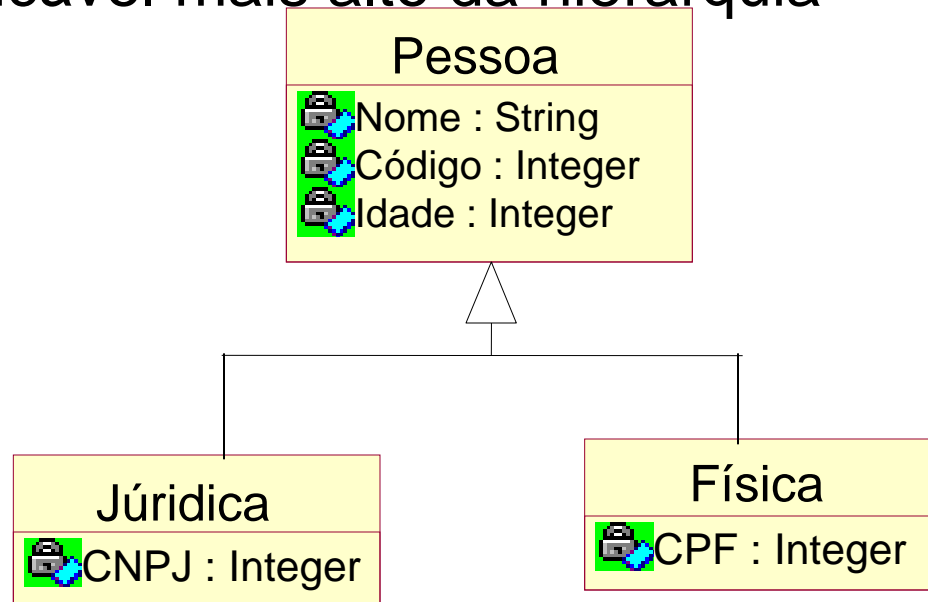
Relacionamento

- **Associação** - é uma conexão bi-direcional entre classes
- **Agregação** - é um tipo mais forte de conexão, aonde a relação é entre o todo e suas partes
 - Simples - estrutura todo-parte
 - Composição - vincula o tempo de vida das partes. O objeto todo é responsável pela criação e destruição de suas partes.



Relacionamento

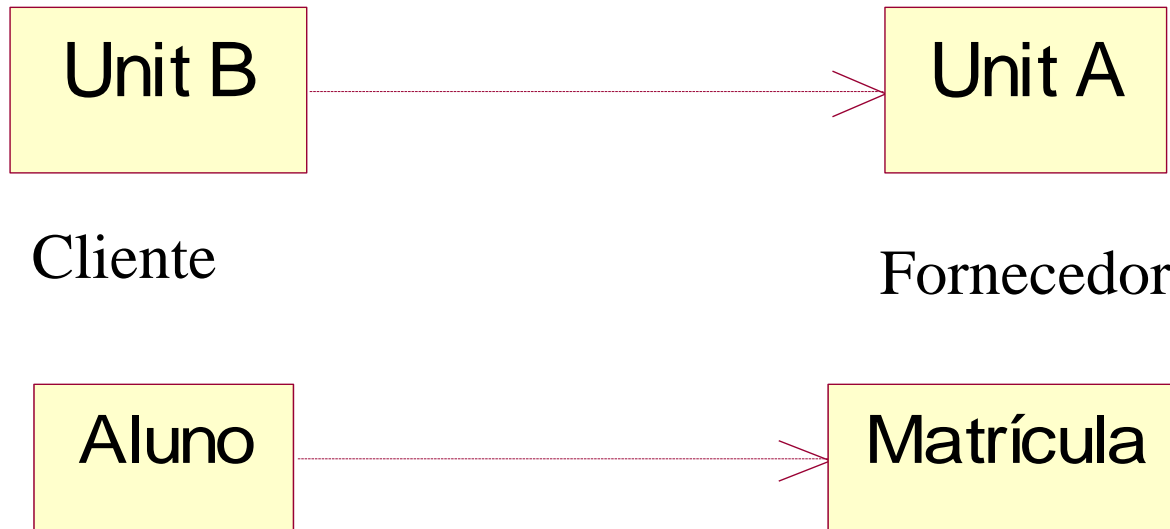
- Herança - é uma relação entre uma super-classe e suas sub-classes
- Há duas formas de se descobrir heranças:
 - Generalização
 - Especialização
- Atributos comuns, operações, relações e/ou, são mostradas no nível aplicável mais alto da hierarquia



Relacionamento

■ Dependência

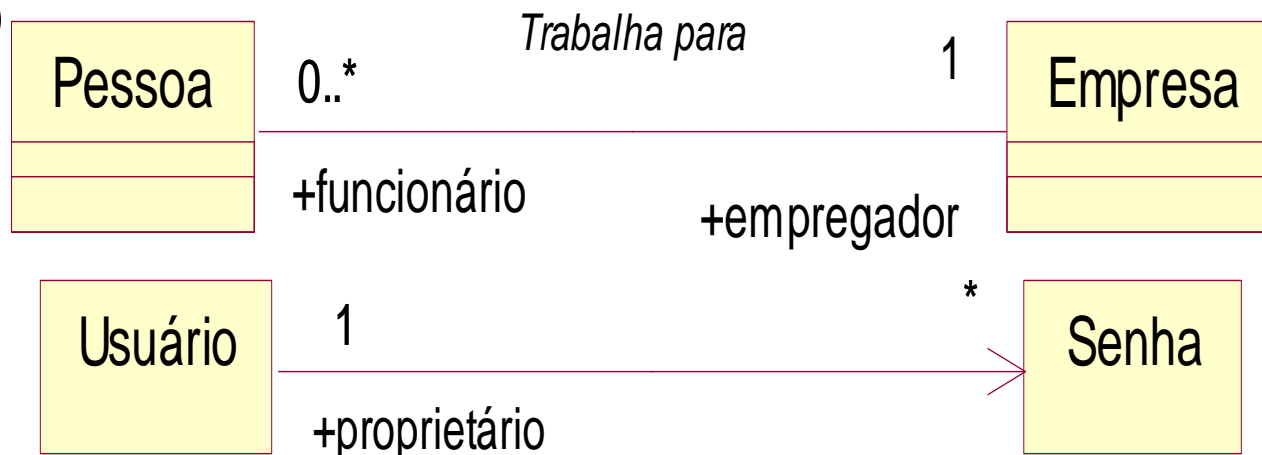
- Indica uma situação na qual uma mudança em um elemento **independente** pode afetar outro elemento **dependente**.
- Ligação entre um cliente e um fornecedor de serviço.
- Relacionamento de utilização.
- EX:



Relacionamento

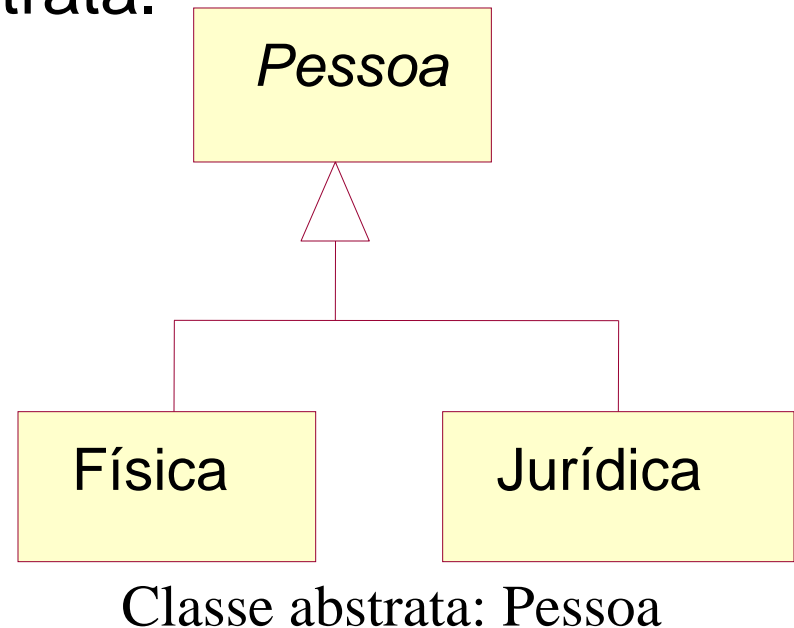
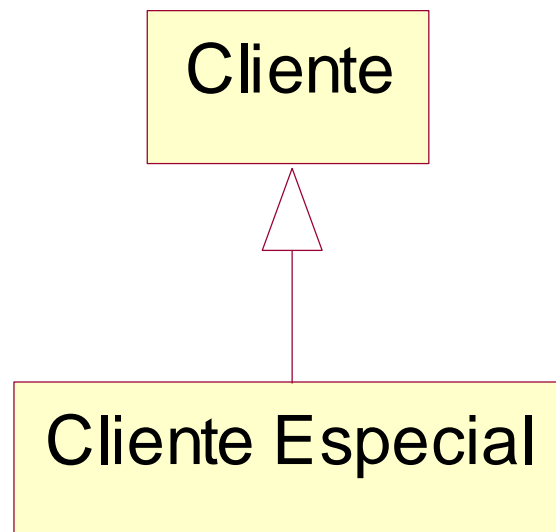
■ Adornos

- Nome
- Multiplicidade
- Navegação - Apesar de associações e agregações serem bi-direcionais por definição, frequentemente é desejável restringir a navegação em uma única direção.
- Visibilidade
- Nomes de papéis (O que uma classe representa para outra)



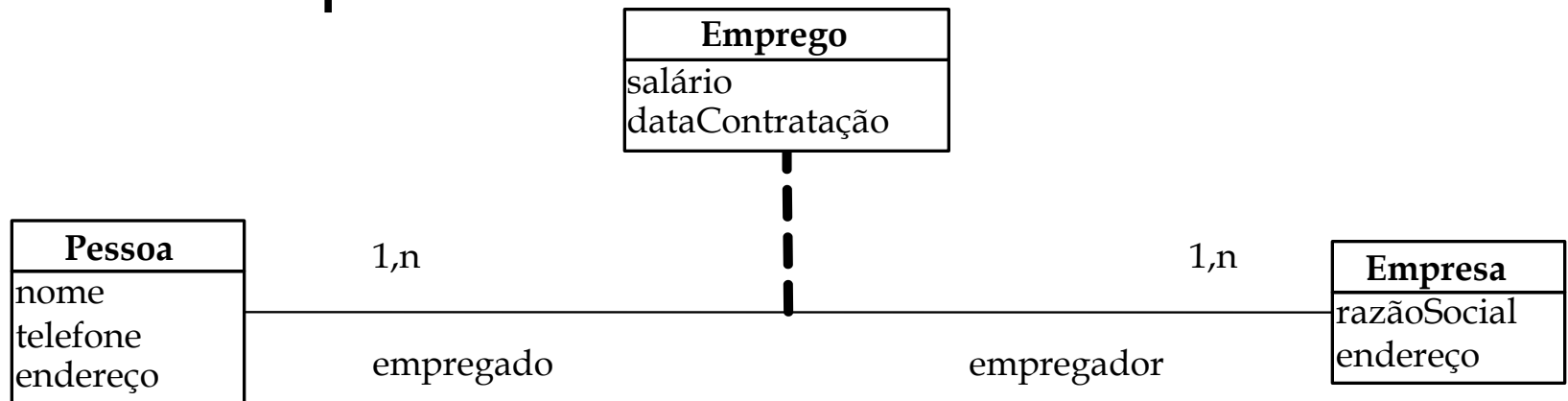
Classe Abstrata

- Uma classe que não possui objeto diretamente instanciado.
- Criada para facilitar o compartilhamento de atributos e operações entre subclasses mais especializadas.
- Toda classe abstrata possui subclasse, mas nem toda super-classe é uma classe abstrata.
- Notação : escrever em *itálico*



Notação para uma classe associativa

- *Representada pela notação utilizada para uma classe. A diferença é que esta classe é ligada a uma associação.*
- Exemplo:





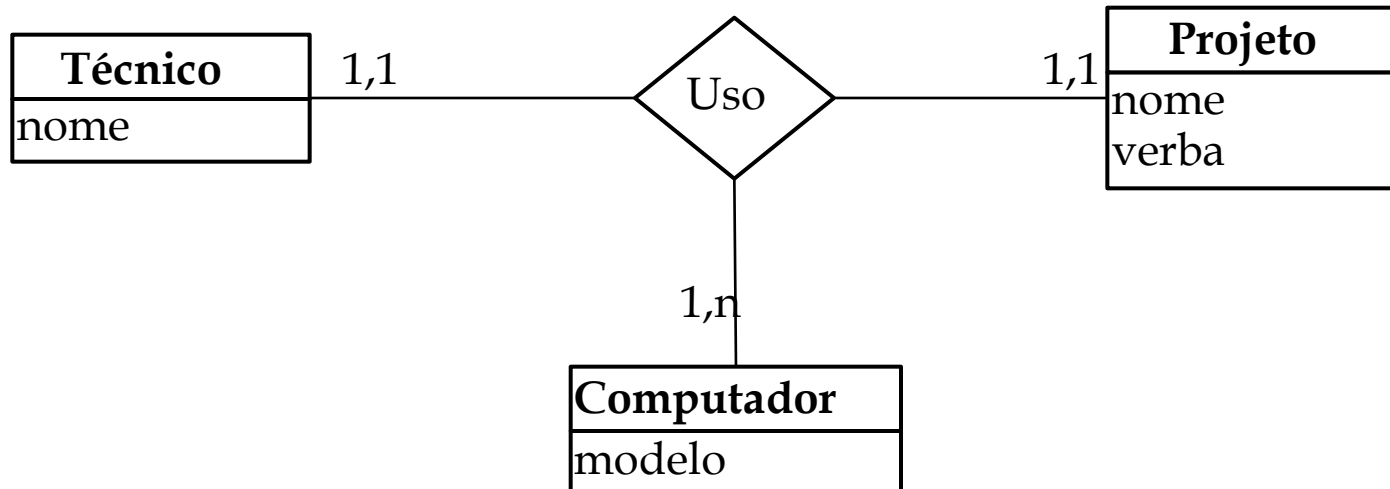
Classe associativa

- É uma classe que está ligada a uma associação, ao invés de estar ligada a outras classes.
- É normalmente necessária quando duas ou mais classes estão associadas, e é necessário manter informações sobre esta associação.
- Uma classe associativa pode estar ligada a associações de qualquer tipo de conectividade.

Associações n-árias

- São utilizadas para representar a associação existente entre objetos de n classes.
- Uma **associação ternária** são uma caso mais comum (menos raro) de associação n -ária ($n = 3$).
- Na notação da UML, as linhas de associação se interceptam em um losango.

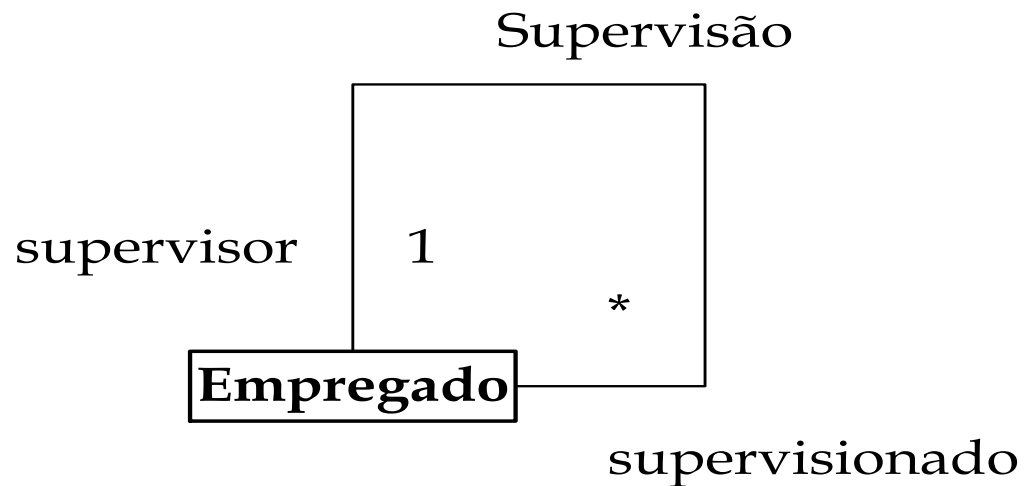
Exemplo (associação ternária)



Associações reflexivas

- Associa objetos da mesma classe.
 - Cada objeto tem um papel distinto na associação.
- A utilização de papéis é bastante importante para evitar ambigüidades na leitura da associação.
- Uma associação reflexiva *não* indica que um objeto se associa com ele próprio.
 - Ao contrário, indica que objetos de uma mesma classe se associam

Exemplo (associação reflexiva)



Modelo de classes no processo de desenvolvimento

- Detalhes são adicionados aos modelos, à medida que o problema é entendido.

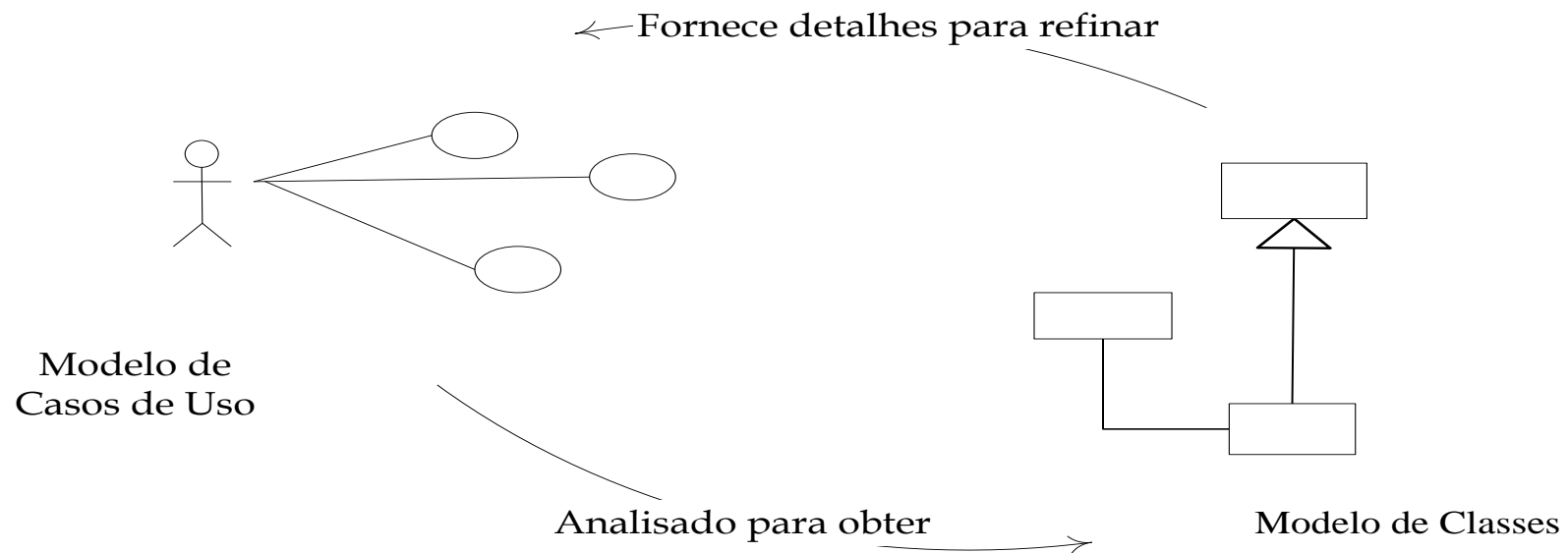
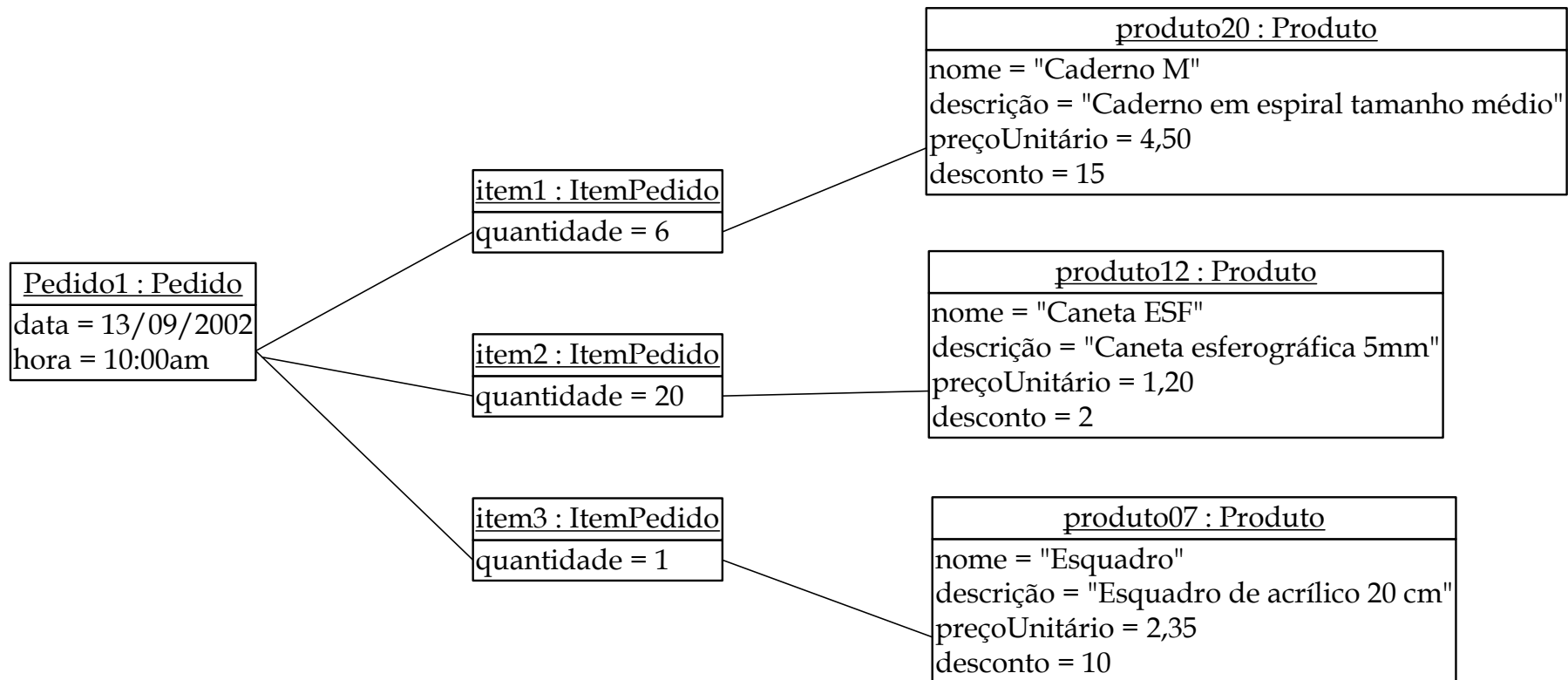


Diagrama de objetos

- Além do diagrama de classes, A UML define um segundo tipo de diagrama estrutural, o diagrama de objetos.
- Pode ser visto com uma instância de diagramas de classes
- Representa uma “fotografia” do sistema em um certo momento.
 - exibe as ligações formadas entre objetos conforme estes interagem e os valores dos seus atributos.

Exemplo (Diagrama de objetos)



Notações Genéricas

■ Estereótipo

- mecanismos de extensão da própria UML
- fornece um grau de liberdade para atender as necessidades específicas do sistema
- permitem adicionar novos blocos de construção semelhantes aos existentes, mas específicos a um determinado problema.
- UML apresenta uma relação de palavras reservadas que são utilizadas como estereótipo.

Enumeration: Aplicado a classe. Especifica um tipo enumerado, incluindo seus possíveis valores como um conjunto de identificadores

<<enumeration>>

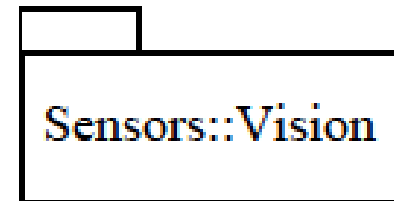
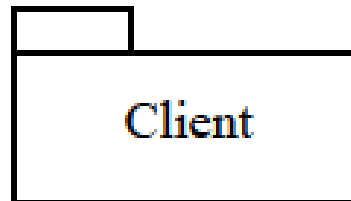
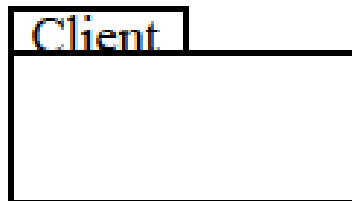
Tipo Cliente

Especial
Aposentado
Normal
Estudante

Notações Genéricas

■ Pacote

- Definição: Um pacote(*package*) em UML é um mecanismo de agrupamento genérico
- Objetivo: divisão do sistemas em módulos.
- Notação: pasta com o nome no interior ou na aba
- No caso de um pacote contido em outro, o nome completo do pacote contido inclui o nome do seu



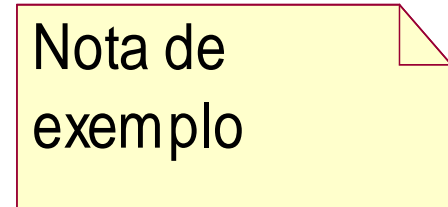
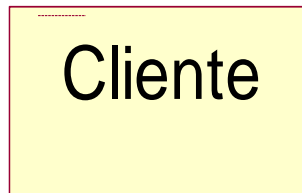
Notações Genéricas

- Podem ser empregadas nos vários diagramas da UML. Os principais são:

- ☐ Nota
- ☐ Estereótipo
- ☐ Pacote

- **Nota**

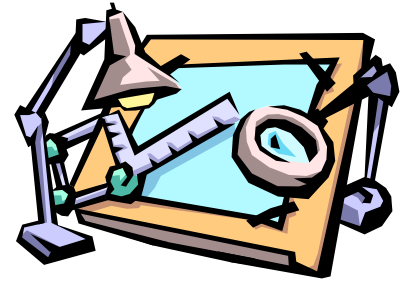
- ☐ Comentário colocado em um diagrama sem qualquer conteúdo semântico.
- ☐ Pode ser anexada a vários elementos da modelagem
- ☐ EX:



Notações Genéricas

- Permite a divisão de tarefas na equipe.
- Essencial em sistemas complexos.
- Atividade bottom-up.
- Remete aos conceitos de Package no Java e Namespace no .NET
- Regras:
 - Uma classe pode pertencer a mais de um pacote
 - Nenhuma classe pode ficar sem pacote
 - Nomes genéricos

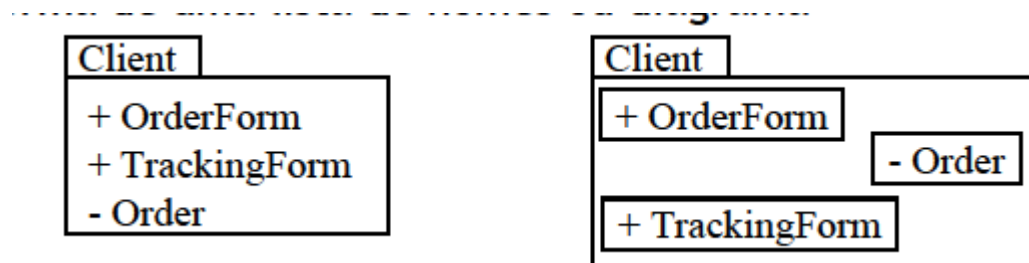
Critérios para Definição de Pacotes de Classes



- Utilizar o “bom senso” para agrupar os elementos em pacotes.
- Mesmo Pacote
 - ☐ Mudanças no comportamento ou estrutura de uma classe requerem mudanças em outras classes
 - ☐ Remoção de uma classe impacta outra classe
 - ☐ Duas classes interagem bastante através de mensagens
 - ☐ Duas classes são afetadas por mudanças no mesmo ator.
 - ☐ Duas classes possuem relacionamentos entre si
- Pacotes Diferentes
 - ☐ Classes relacionadas com diferentes atores

Conteúdo de um pacote

- Uma vez que representa um agrupamento, um pacote é em geral *dono* de diversos elementos: classes, interfaces, componentes, nós, colaborações, casos de uso, diagramas, e até outros pacotes
- Esses elementos podem ser indicados no interior do pacote, na forma de uma lista de nomes ou diagrama

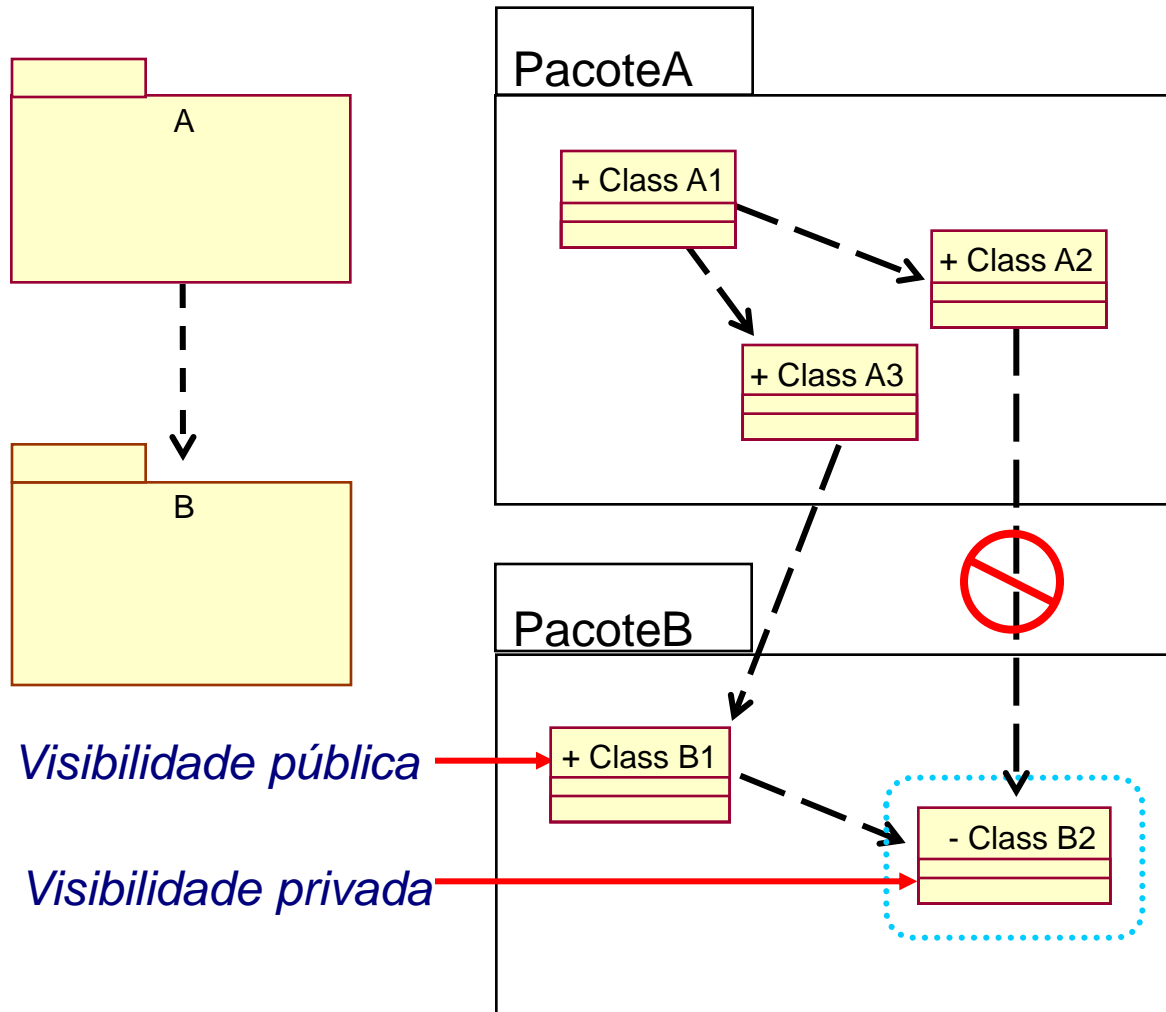


- Um pacote forma um *espaço de nomes*
 - classe *Order* do pacote *Client* é designada *Client::OrderClient*

Visibilidade dos elementos contidos num pacote

- Pode-se indicar a visibilidade dos elementos:
 - + (público) : visível por todos que *importam* ou *acessam* o pacote (nomes sem :: no 1º caso, com :: no 2º caso)
 - # (protegido): visível só pelos pacotes-filhos (por relação de generalização -ver adiante)
 - -(privado): visível só por outros elementos do pacote
- Os elementos públicos de um pacote são chamados também os elementos *exportados* pelo pacote

Visibilidade entre Pacotes

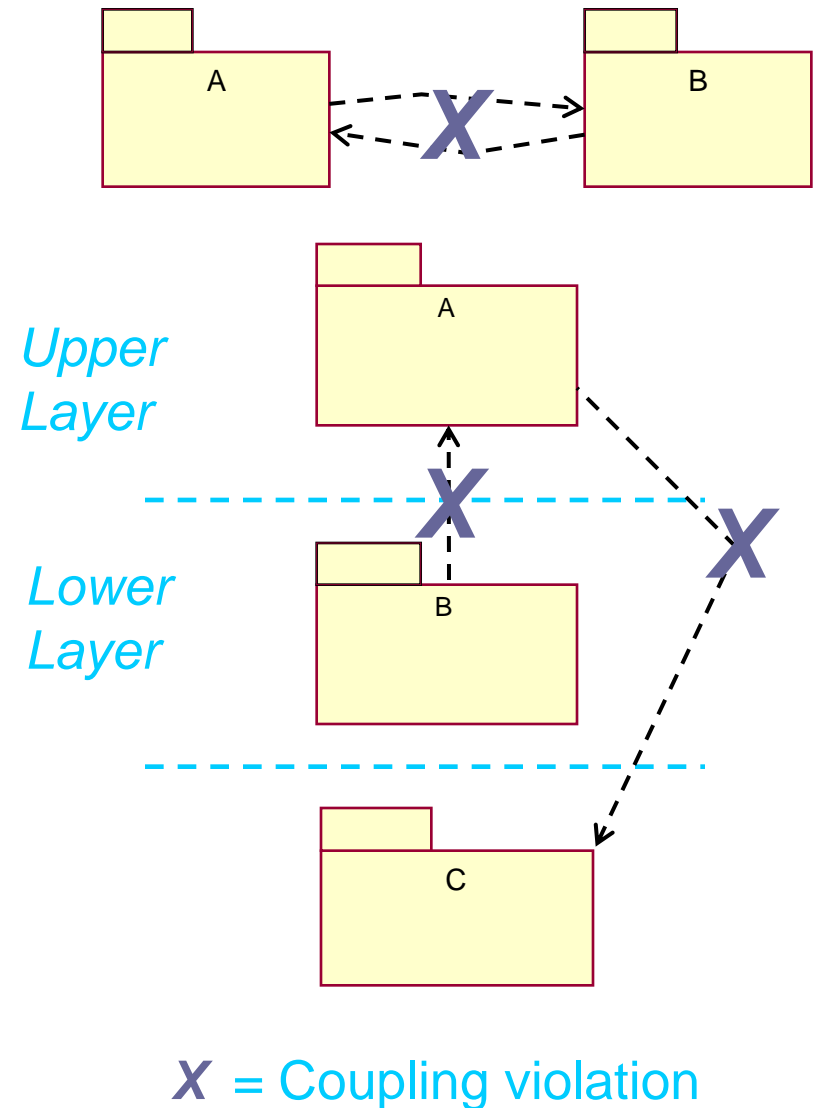


Somente as classes públicas podem ser referenciadas fora do seu próprio pacote

Encapsulamento

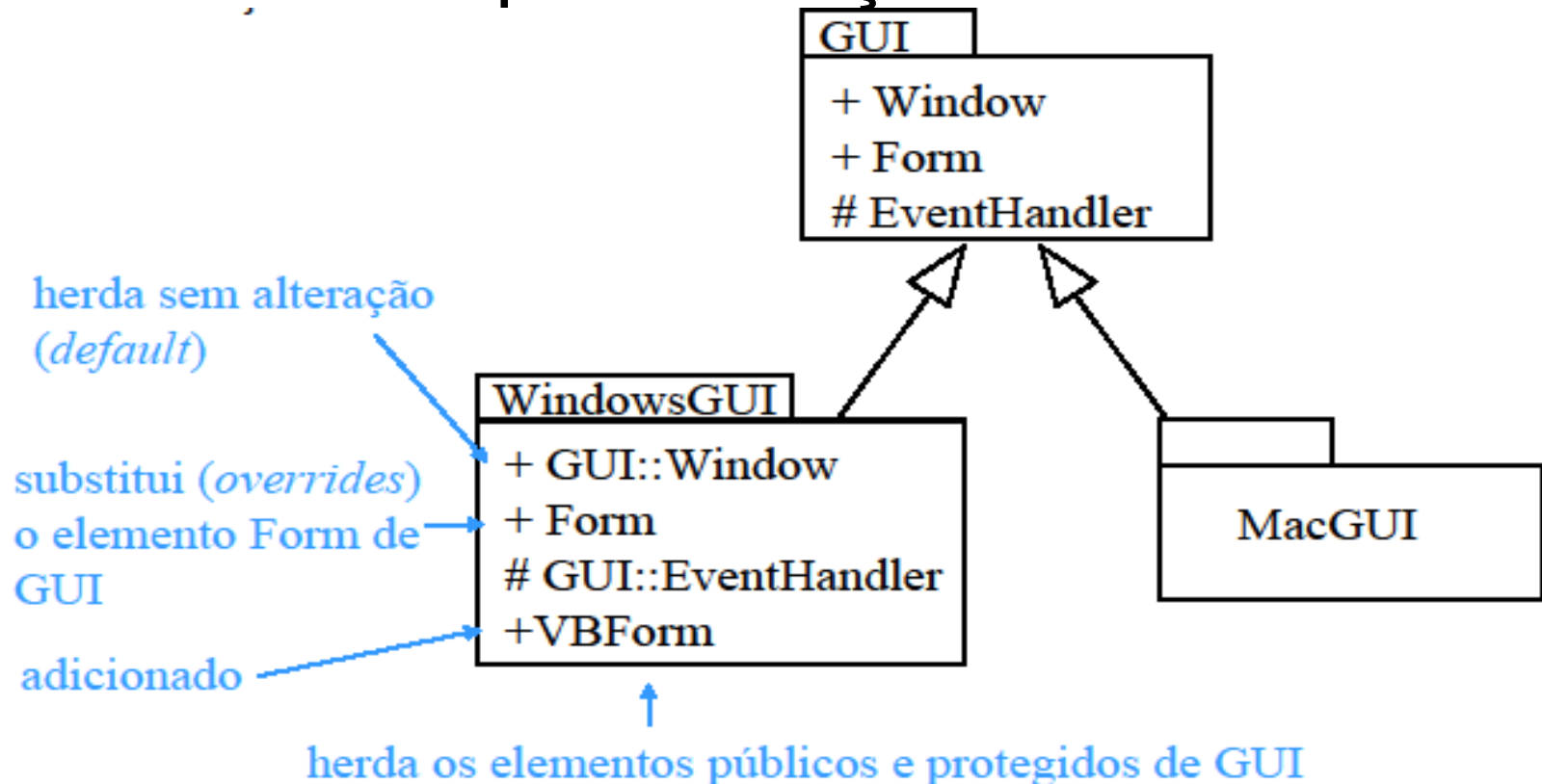
Relações entre Pacotes

- Pacotes não devem ser interdependentes
- Pacotes em camadas inferiores não devem depender de pacotes em camadas superiores
- Em geral, dependências não pulam camadas



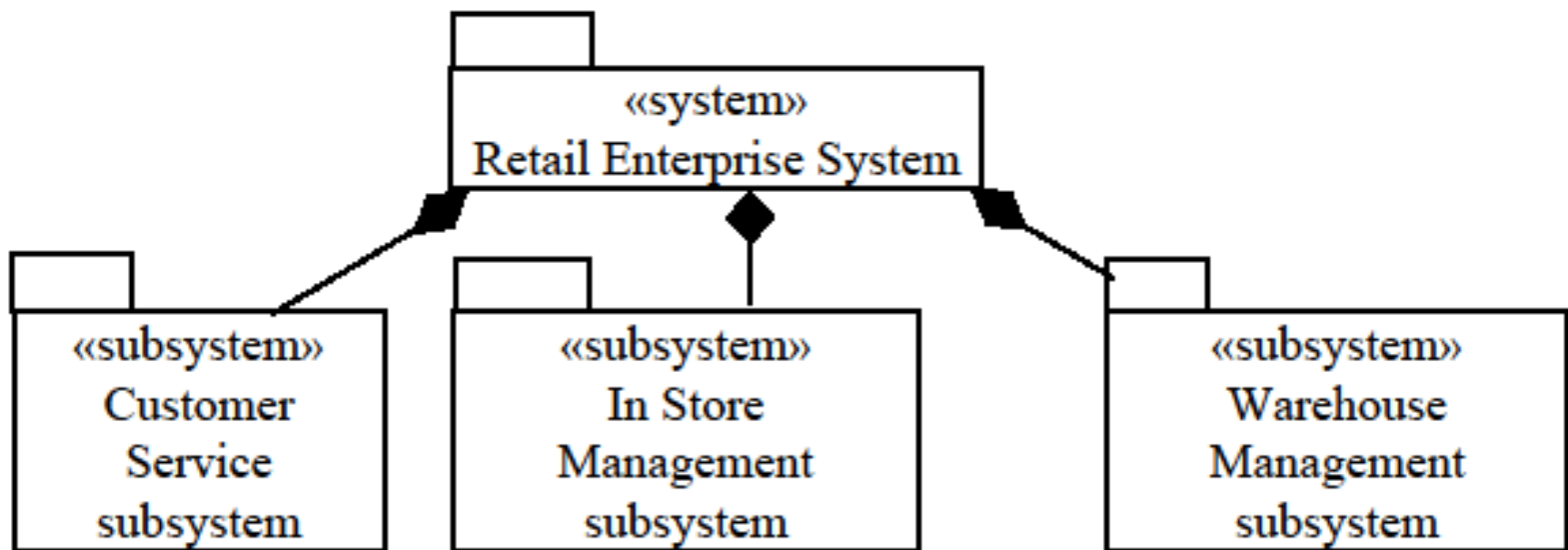
Generalização de pacotes

- Usada para especificar famílias de pacotes relacionados por herança



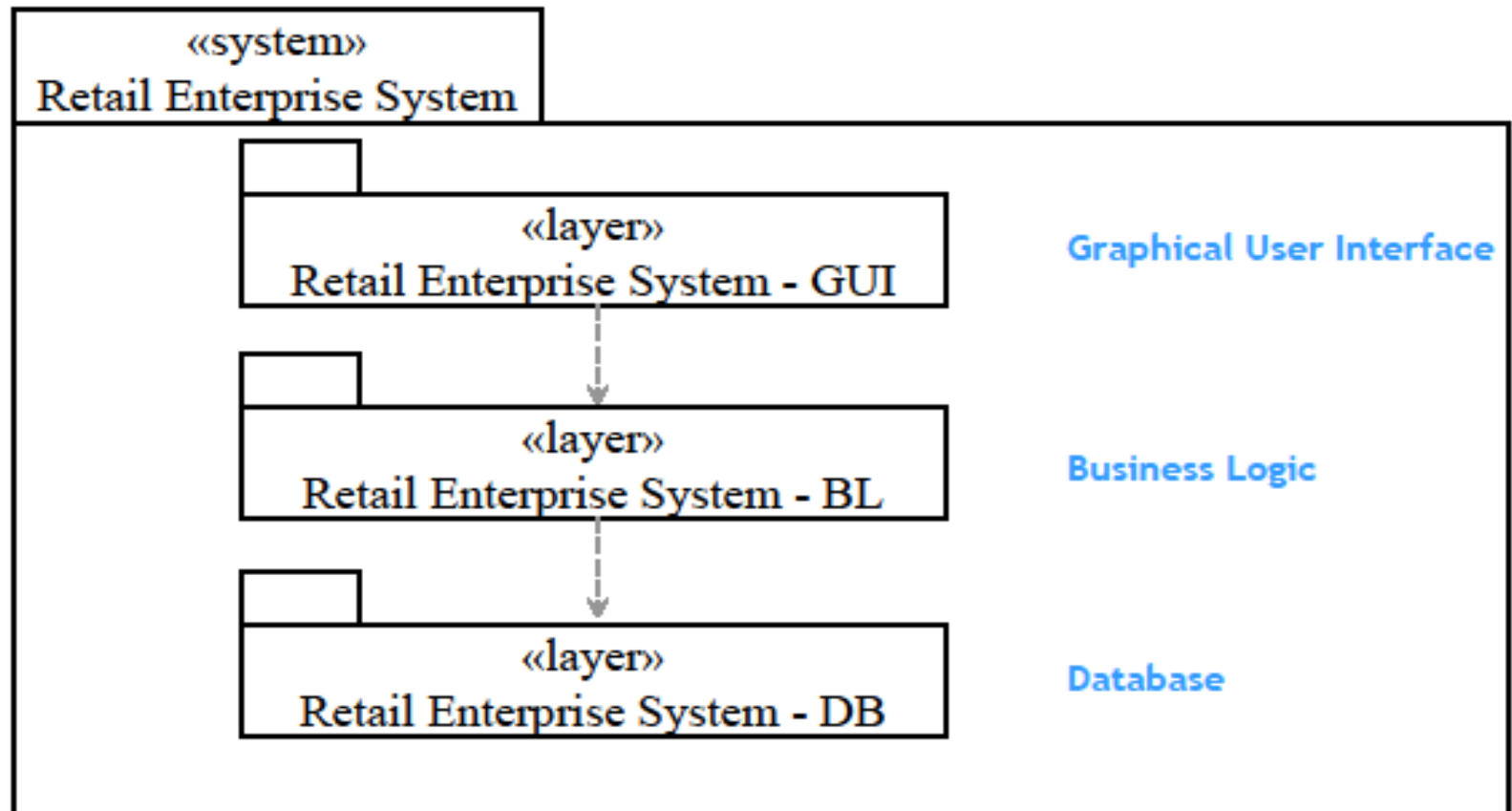
Composição de pacotes (1)

- Sub-pacotes podem ser indicados dentro do pacote-dono ou com relação de composição



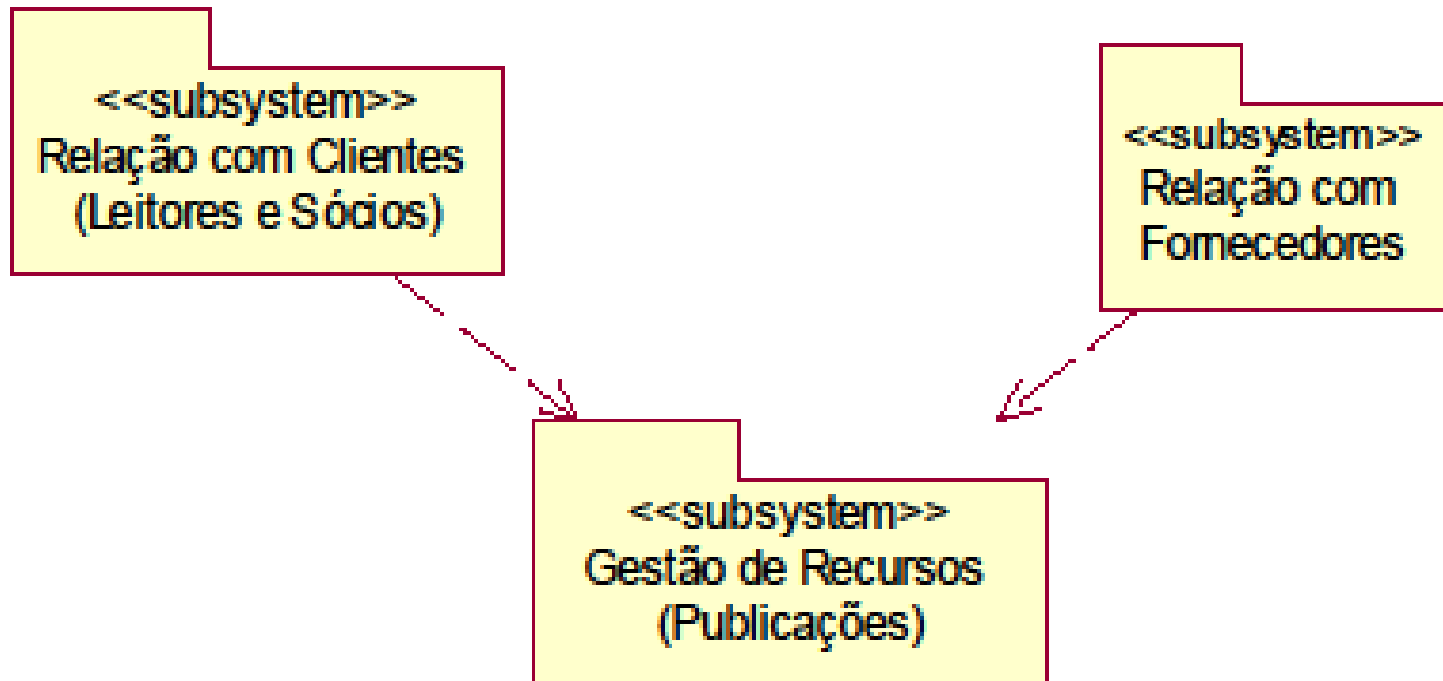
Neste exemplo segue-se uma divisão vertical, por subsistemas!

Composição de pacotes (2)

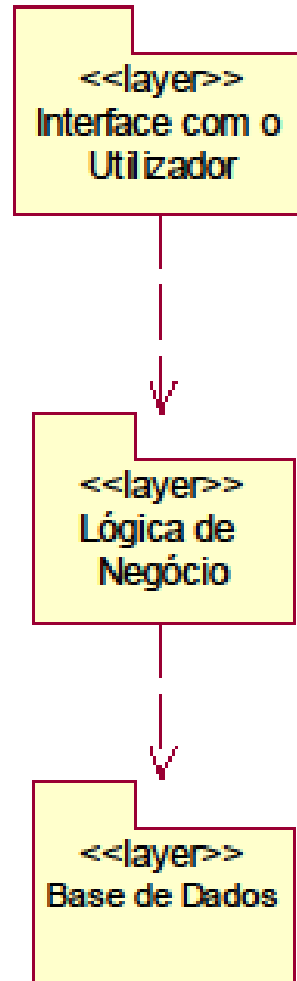


Neste exemplo segue-se uma divisão horizontal, por camadas!

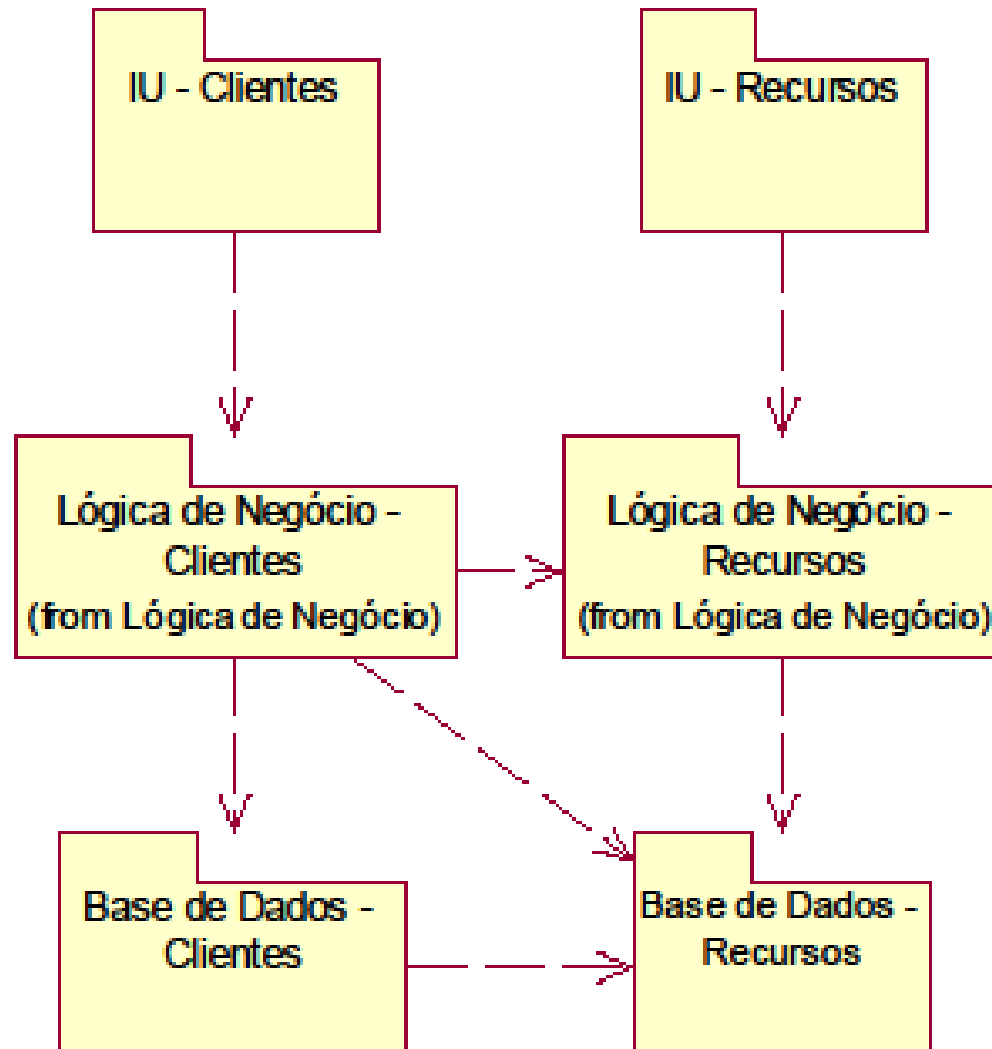
Estudo de caso (biblioteca): divisão em áreas funcionais



Estudo de caso (biblioteca): divisão em camadas técnicas



estudo de Caso (biblioteca): divisão em camadas técnicas e áreas funcionais



Estereótipos em pacotes

- «system»-pacote que representa o sistema completo que está a ser modelado (incluindo todos os modelos e elementos dos modelos)
- «subsystem»-pacote que representa uma parte independente de sistema completo que está a ser modelado; corresponde normalmente a um corte "vertical"
- «facade»(fachada) -pacote que constitui uma vista sobre outro pacote (não acrescenta funcionalidades, apenas apresenta de forma diferente)
- «framework»(infra-estrutura aplicacional) -pacote que representa um conjunto de classes abstratas e concretas concebido para ser estendido, implementando a funcionalidade típica de um determinado domínio de aplicação
- «stub»-pacote que serve como *proxy* para o conteúdo público de outro pacote
- «layer»-pacote que representa uma camada horizontal de um sistema