PCS3413

Engenharia de Software e Banco de Dados

Aula 23

Acoplamento

- Indica dependência entre classes.
- Deve ser o menor possível.
- Direcionar associações quando possível: diagrama de sequência.

Coesão

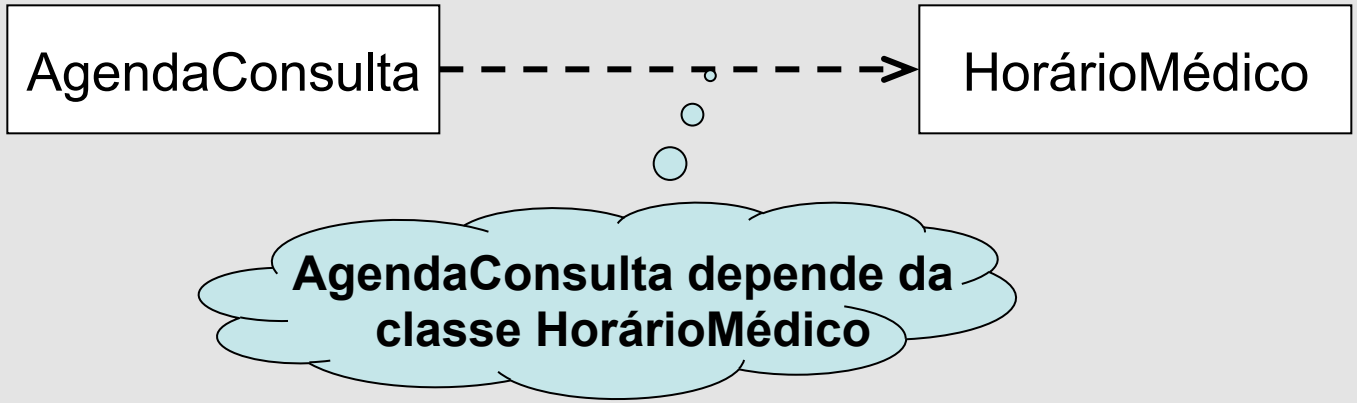
- medida do quão fortemente são as responsabilidades de uma classe.
- atributos e operações de uma classe devem estar altamente correlacionadas

Encapsulamento de Objetos

- O acesso a informações internas da classe só acessível via sua interface, que permite acesso as operações públicas ou serviços da classe.

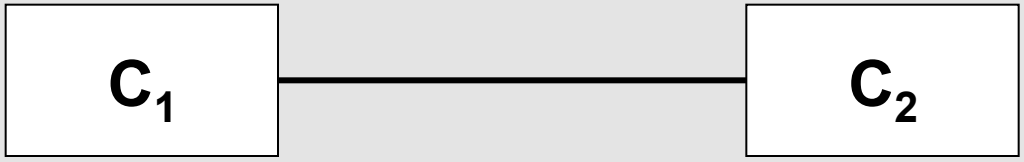
Dependência entre classes

- Uma dependência entre classes indica que uma classe depende dos serviços fornecidos por outra.
- Associações entre classes indicam a existência de dependência entre elas.
- A forma de dependência entre classes influência a implementação dessas classes.



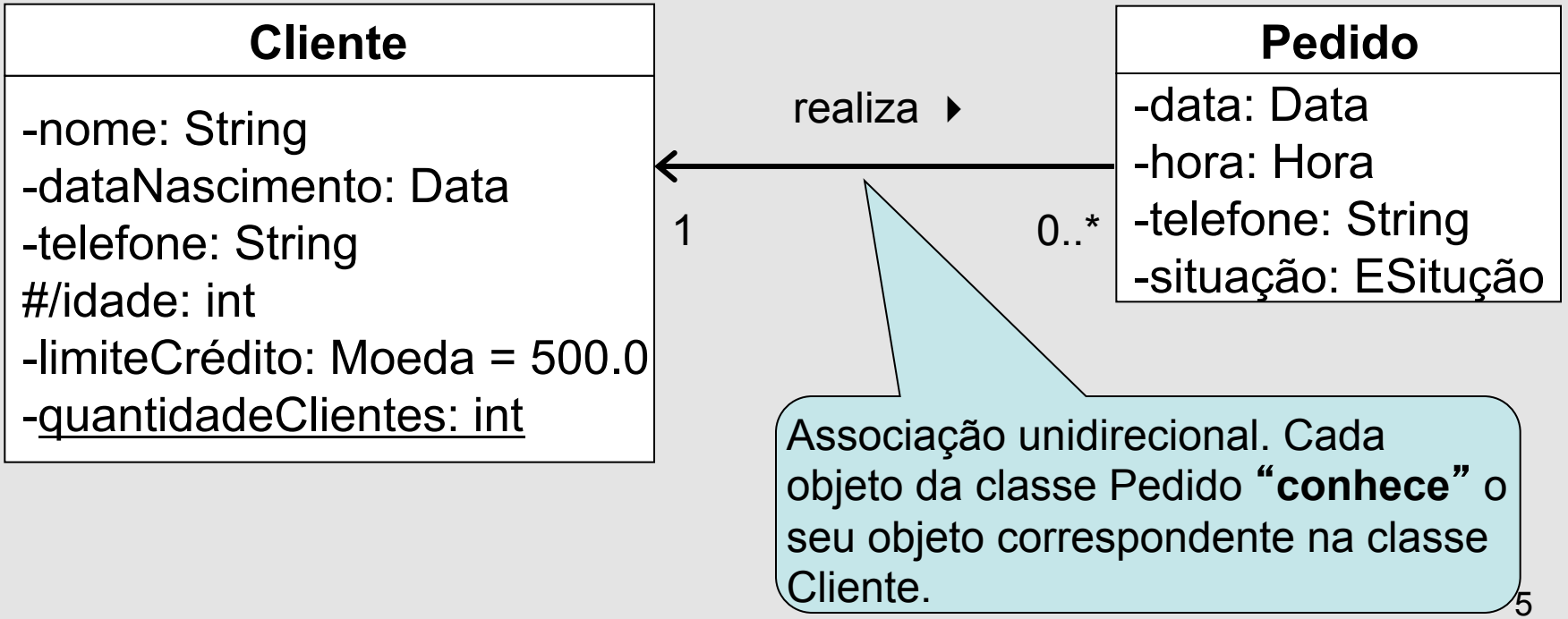
Navegabilidade das associações

- No modelo de classes de análise as associações são modeladas como sendo bidirecionais.
- Para o modelo, se a associação é birecional, então:



1. Cada objeto de C_1 conhece todos os objetos de C_2 aos quais ele está associado.
2. Cada objeto de C_2 conhece todos os objetos de C_1 aos quais ele está associado.

- Associações bidirecionais aumentam o acoplamento entre as classes.
- Sempre que possível definir o sentido unidirecional para as associações.



Navegabilidade das associações - continuação

A definição do sentido da navegabilidade é feita em função dos diagramas de interação construídos para o sistema. Devemos analisar os fluxos das mensagens entre objetos no diagrama de interações.

Para dois objetos associados A e B no diagrama de classe, há pelo menos uma mensagem de A para B em algum diagrama de interação, então a associação deve ser navegável da classe A para a B.

Se também existir, pelo menos uma mensagem de B para A, então a associação deve ser navegável da classe B para a A.

Arquitetura

Arquitetura de Software

"estrutura organizacional do software. Uma arquitetura pode ser recursivamente decomposta em partes que interagem através de interfaces. Relacionamentos conectam as partes e restrições que se aplicam ao agrupamento das partes."

Documento de especificação da UML (OMG, 2001)

Arquitetura Lógica

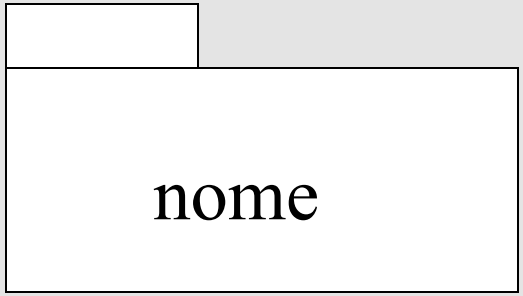
- Organização das classes em subsistemas
- facilita o entendimento do sistema

- define de que forma o sistema está dividido e as interfaces entre essas partes (relacionamentos definem interfaces)

- Vantagens:
  - unidades menores de desenvolvimento
  - maximizar o reuso (uso de componentes já construídas para outros sistemas)
  - ajuda a gerenciar a complexidade no desenvolvimento

Packages

- *Package* é um mecanismo para organizar elementos em grupos.
- Para a representação da arquitetura de software, um *package* é uma coleção de classes para uma determinada aplicação.
- Representação:

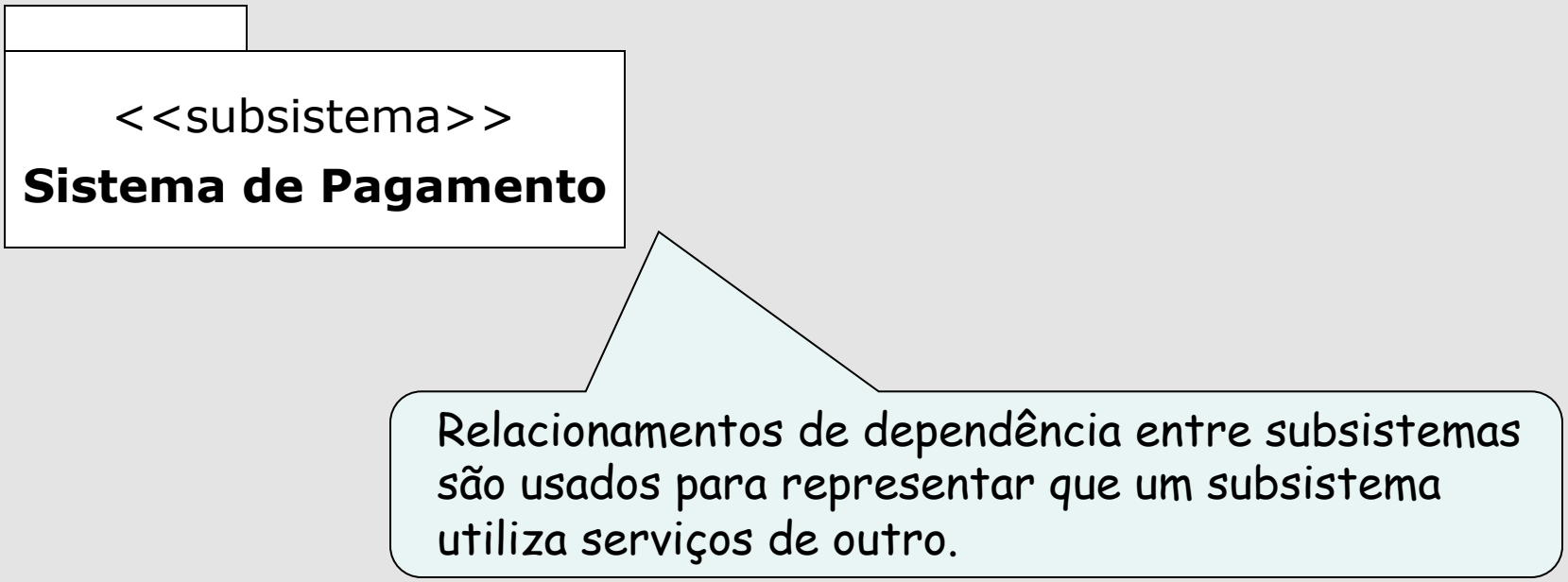


Dependência entre *Packages*

O *package* A depende do *package* B.

Subsistema

- corresponde a um aglomerado de classes.

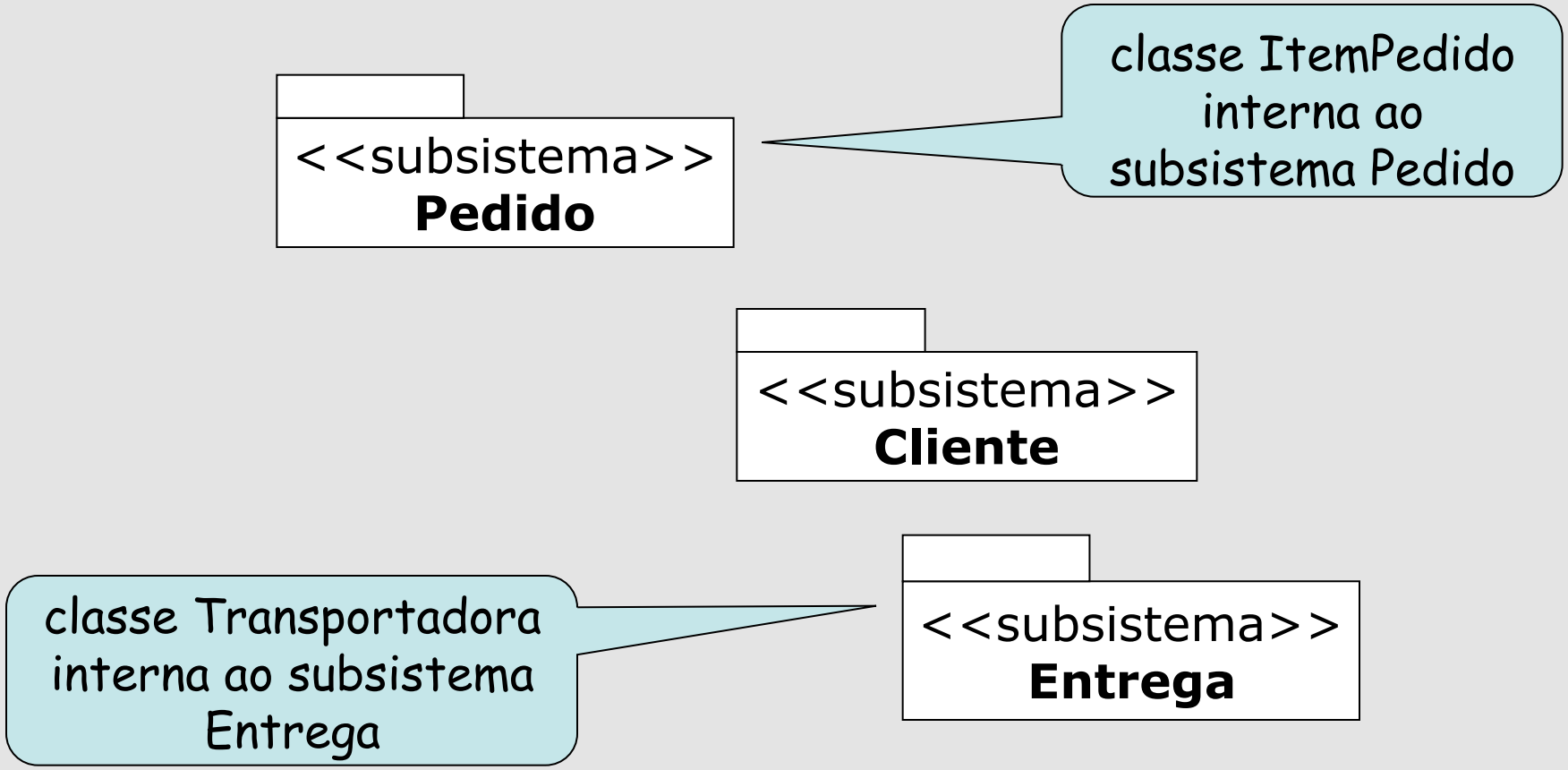


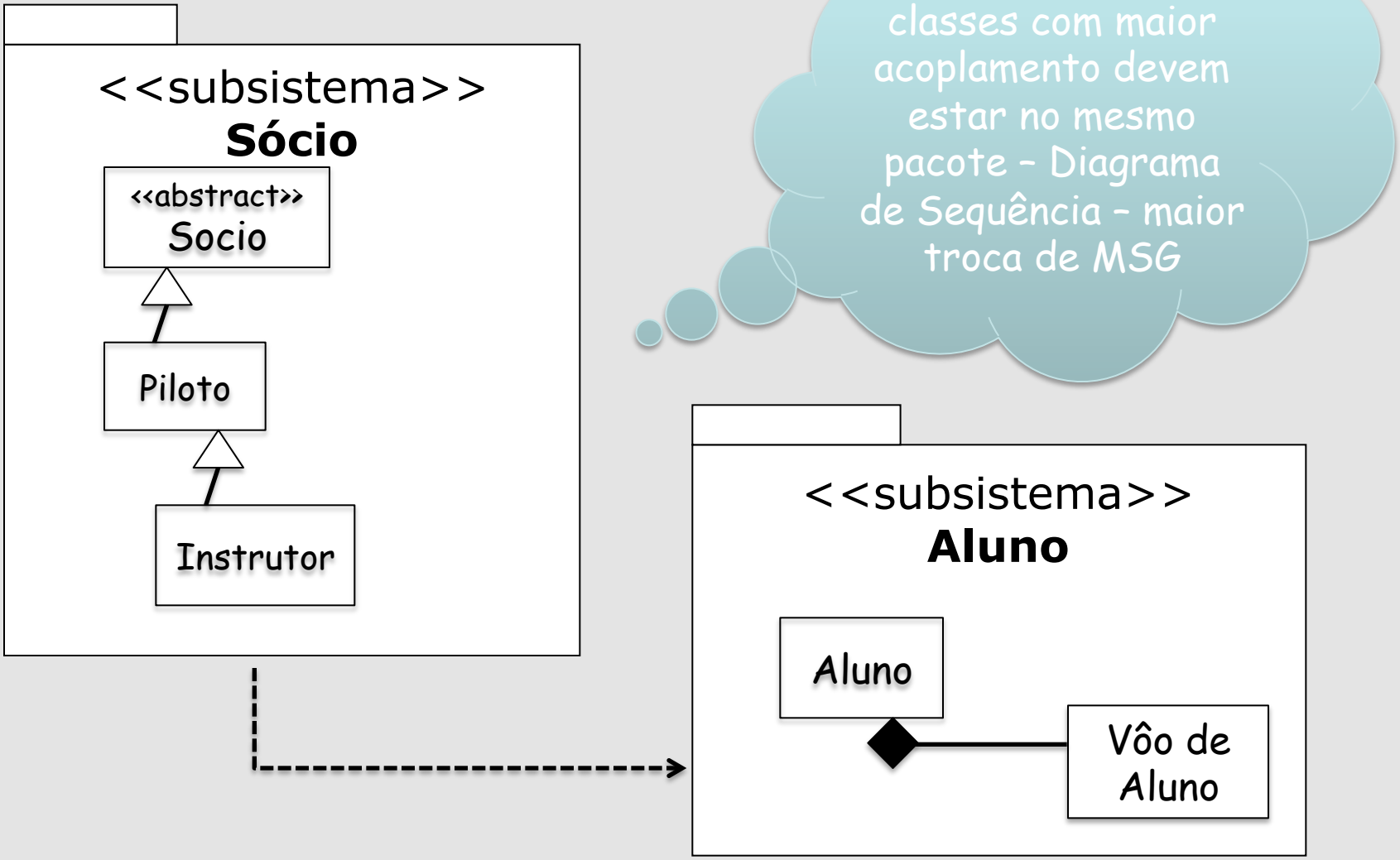
Alocação de classes a subsistemas

- modelo de classes do domínio:
  - classificar classes mais importantes e classes menos importantes.
  - para cada mais importante criar um subsistema
- Acoplamento e coesão
  - Deve-se manter baixo acoplamento: manter no mínimo a quantidade de associações entre classes de diferentes subsistemas
  - Dentro de um subsistema deve-se ter máxima coesão: deve haver mais associações dentro de um subsistema do que entre elementos de diferentes subsistemas.

Alocação de classes a subsistemas - exemplo

- Ex. Sistema de Vendas ela Web




<<subsistema>>
Sócio
<<abstract>>
Socio
Piloto
Instrutor
classes com maior acoplamento devem estar no mesmo pacote - Diagrama de Sequência - maior troca de MSG
<<subsistema>>
Aluno
Aluno
Vôo de Aluno

Arquitetura em camadas

- camada de software:
  - coleção de unidades de software podem ser executadas ou acessadas.
- objetivo:
  - Atender requisitos não funcionais
    - portabilidade –
    - manutenabilidade – mudanças em camadas mais baixas que não afetem sua interface não implica em mudanças em camadas mais altas. Mudanças em camadas mais altas que não signifiquem em novo serviço em camada mais baixa não altera estas.

- a dependência entre camadas pode ser representada por relacionamento de dependência.
- camadas mais altas devem depender das camadas mais baixas e não ao contrário.
  - Isso significa que camadas mais altas têm conhecimento da interface das camadas mais baixas.

Mais de duas Camadas

- camada de apresentação

composta por classe que fornecem a visualização dos dados. – Classes de Fronteira

- camada da aplicação ou de serviço,

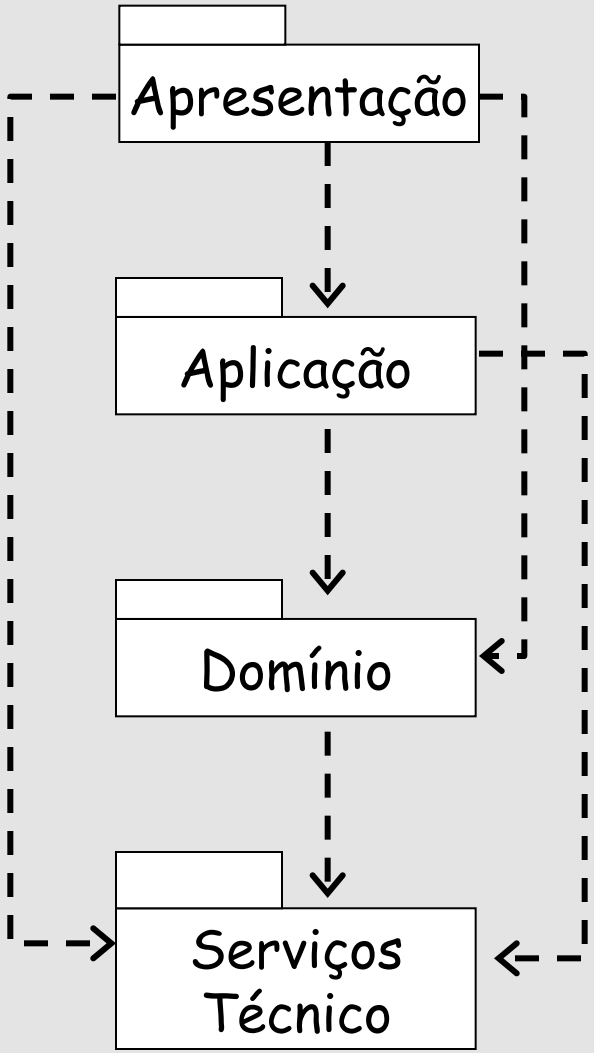
Traduz mensagens da camada de apresentação em mensagens compreendidas pelos objetos do domínio. Também controla a navegação do usuário de uma janela para outra. - Classes de Controle

- camada de domínio

composta por classes que implementam as regras do negócio, validam dados provenientes da camada de aplicação. – Classes de Domínio

- camada de serviço técnico ou de infraestrutura,

Contém classes para permitir que o sistema se comunique com outros sistemas (ex. acesso a um SGBD ou a serviços Web. Todas as camadas superiores são dependentes desta camada.



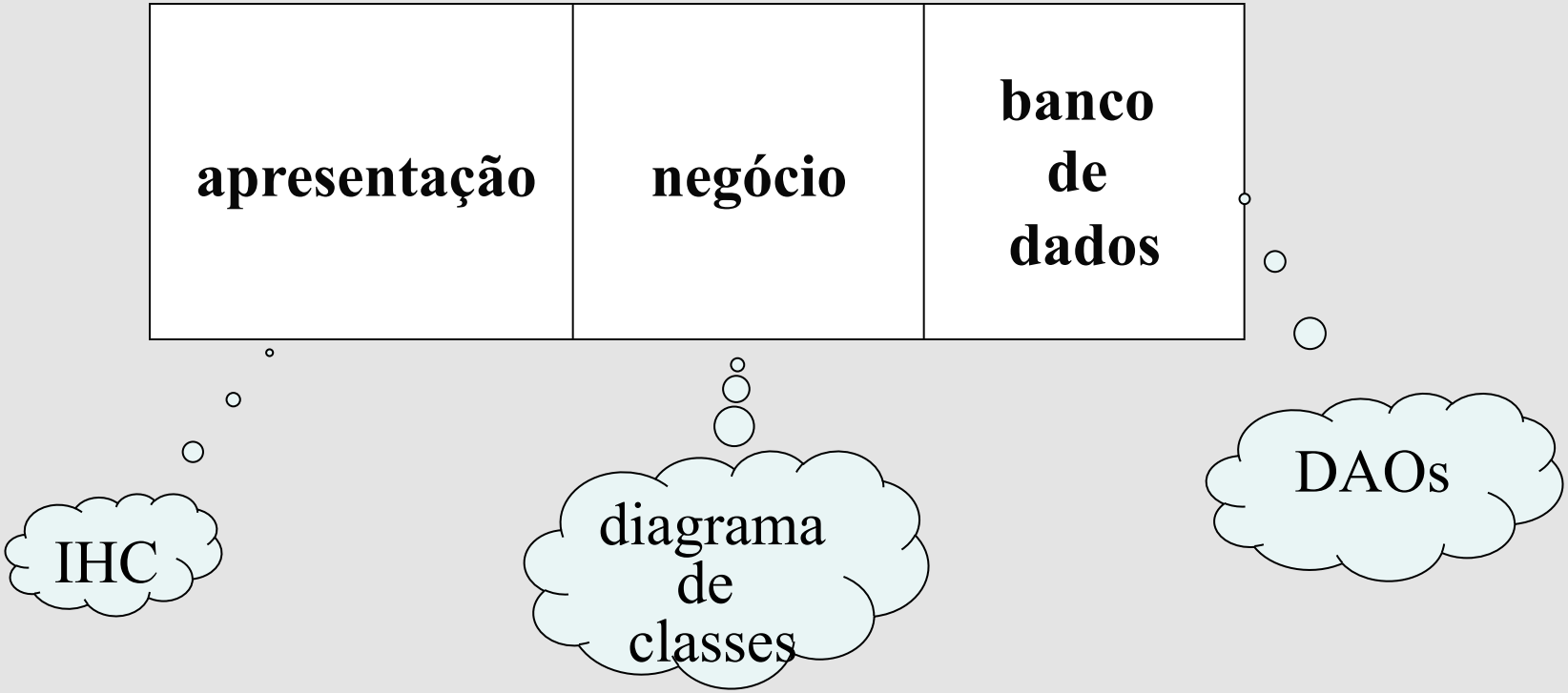
Arquitetura aberta

Camadas mais altas dependem das mais baixas

Facilita o gerenciamento da complexidade

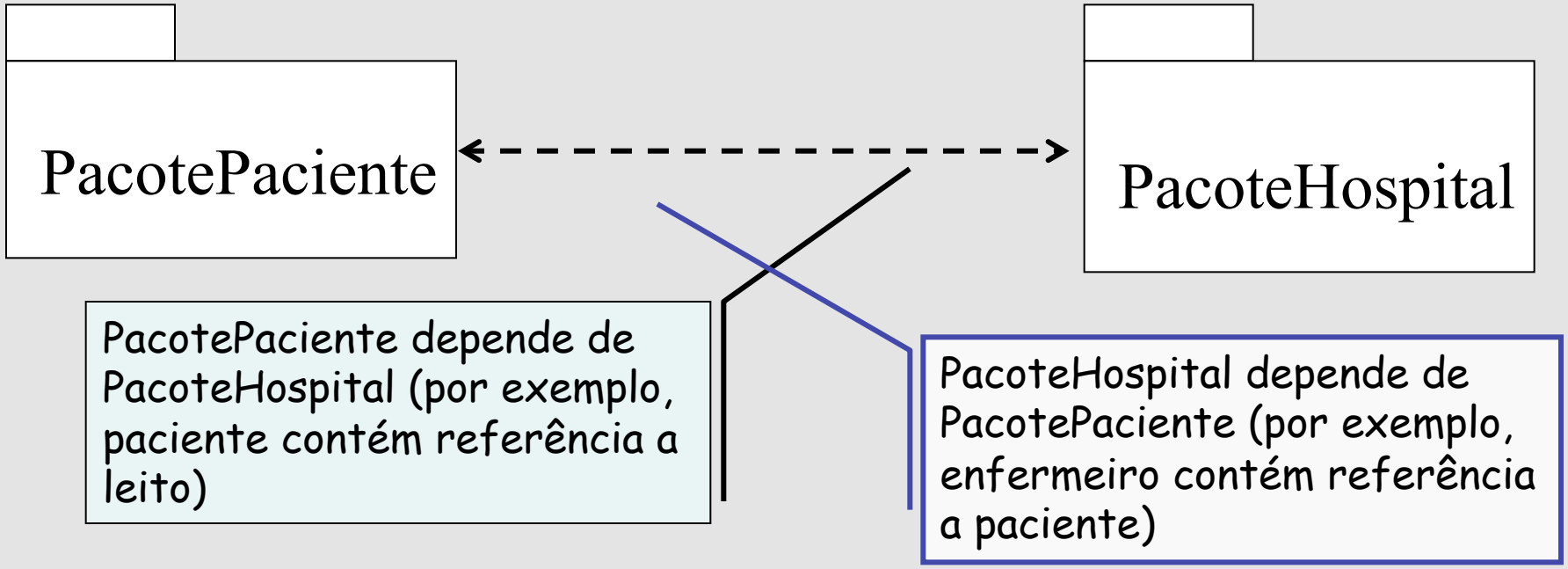
Facilita o reuso – camadas inferiores projetadas para serem independentes das superiores.

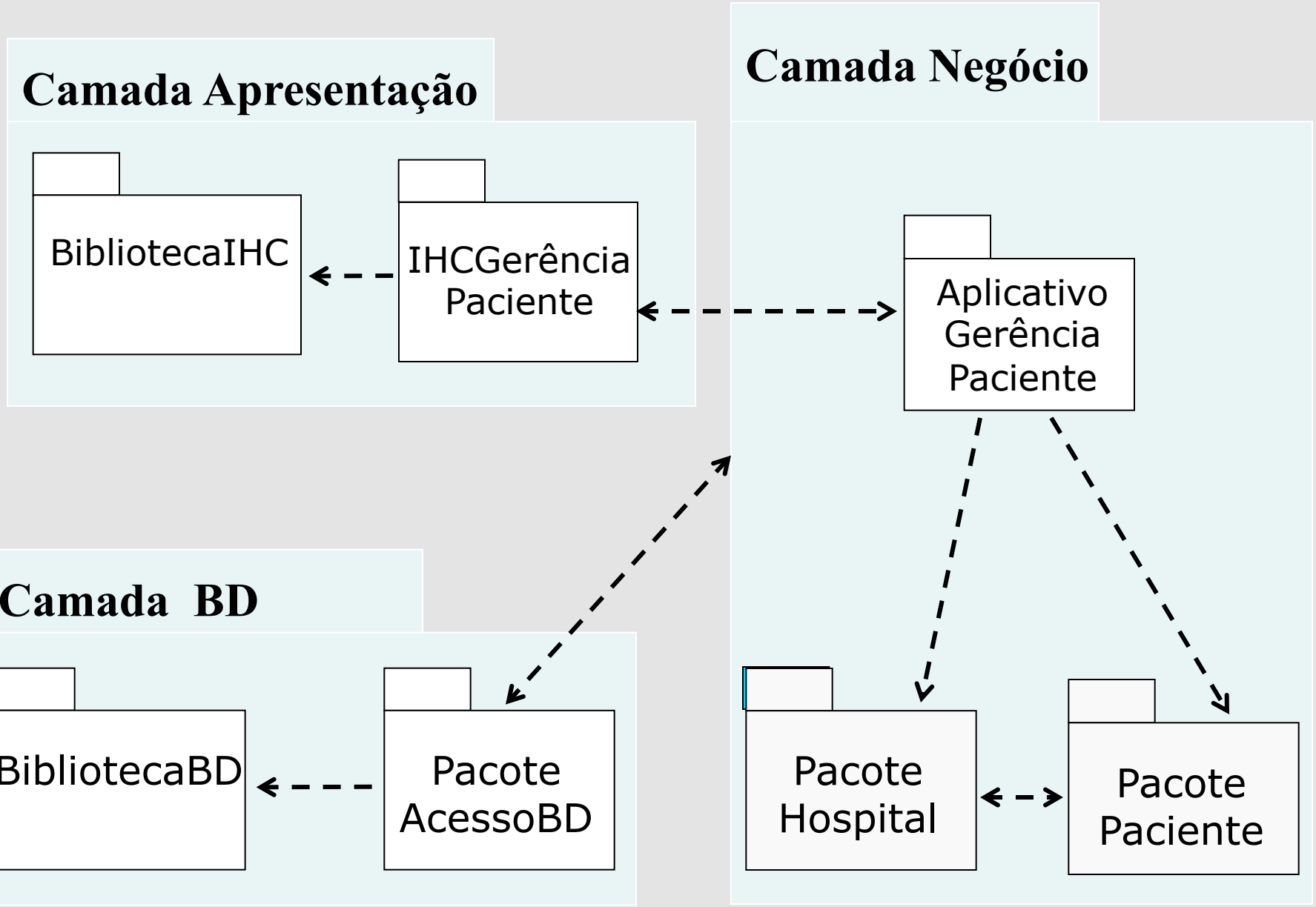
Arquitetura de três camadas



Sistema para Hospital

- PacotePaciente: coleção de classes relacionadas com paciente (ex.: Paciente, HistóricoPaciente).
- PacoteHospital: coleção de classes relacionadas com hospital (Quarto, Leito, Enfermeiro, Médico).




Camada Apresentação
Camada Negócio
BibliotecaIHC
IHCGerência Paciente
Aplicativo Gerência Paciente
Camada BD
BibliotecaBD
Pacote AcessoBD
Pacote Hospital
Pacote Paciente

Camada de Negócio

- PacotePaciente: coleção de classes relacionadas com paciente (ex.: Paciente, HistoricoPaciente).
- PacoteHospital: coleção de classes relacionadas com a instalação do hospital (Quarto, Leito, Enfermeiro, Medico).
- AplicativoGerênciaPaciente: contém o software que implementa as políticas e os procedimentos para internação e alta de paciente.

Camada de Apresentação

- IHCGerênciaPaciente: contém o software para a interface da aplicação
- BibliotecaIHC: contém classes para implementar a interface gráfica

Camada de BD

- Pacote AcessoBD: contém o software para acesso a banco de dados
- Pacote BibliotecaBD: contém classes para suporte de banco de dados

Vantagens e Desvantagens

- manutenabilidade
- divisão dos objetos pelas três camadas – menor acoplamento entre elas – mais fácil reutilização.
- maior número de usuários
- pode-se dividir a carga de processamento entre camadas de negócio e acesso a dados

- menor desempenho
  - ☞ mapeamento dos objetos entre as camadas
- maior complexidade na implementação