

# Tópico 7:

# Diagramas Complementares

---

Luiz Antônio M. Pereira

[lpereira@uninet.com.br](mailto:lpereira@uninet.com.br)

[lpereira@luizantoniopereira.com.br](mailto:lpereira@luizantoniopereira.com.br)

# Para Complementar...

- A UML dispõe de 14 diagramas:
  - 5 deles já foram apresentados:
    - UCs, Classes, DMEs, DAs, DS;
    - Compõem o conjunto dos mais utilizados em modelagem de sistemas.
  - 5 não serão tratados no curso:
    - Perfis, Objetos, Estrutura Composta, Temporização, Comunicação.
  - 4 serão apresentados nesta aula:
    - Complementam os 5 que já vimos;
    - Usados bem menos frequentemente e/ou poucos diagramas por modelo.

# 4 diagramas em uma aula?

---

- Isso é possível porque:
  - Os conceitos envolvidos nesses diagramas são simples e/ou já foram tratados nos diagramas que já estudamos; e/ou
  - A notação é simples.

# Agenda da Aula

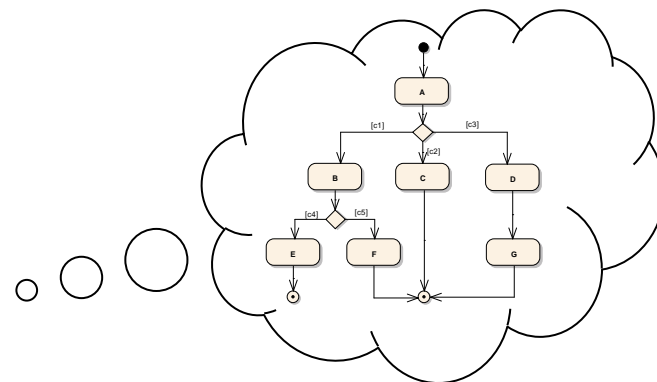
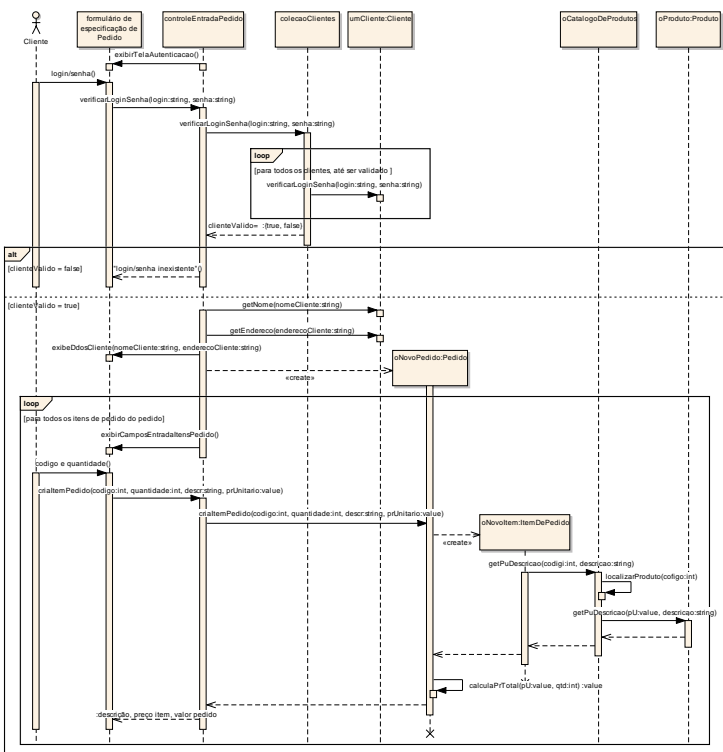
---

- Diagrama de Visão Geral da Interação;
- Diagrama de Pacotes;
- Diagrama de Componentes;
- Diagrama de Instalação.

# Diagrama de Visão Geral da Interação

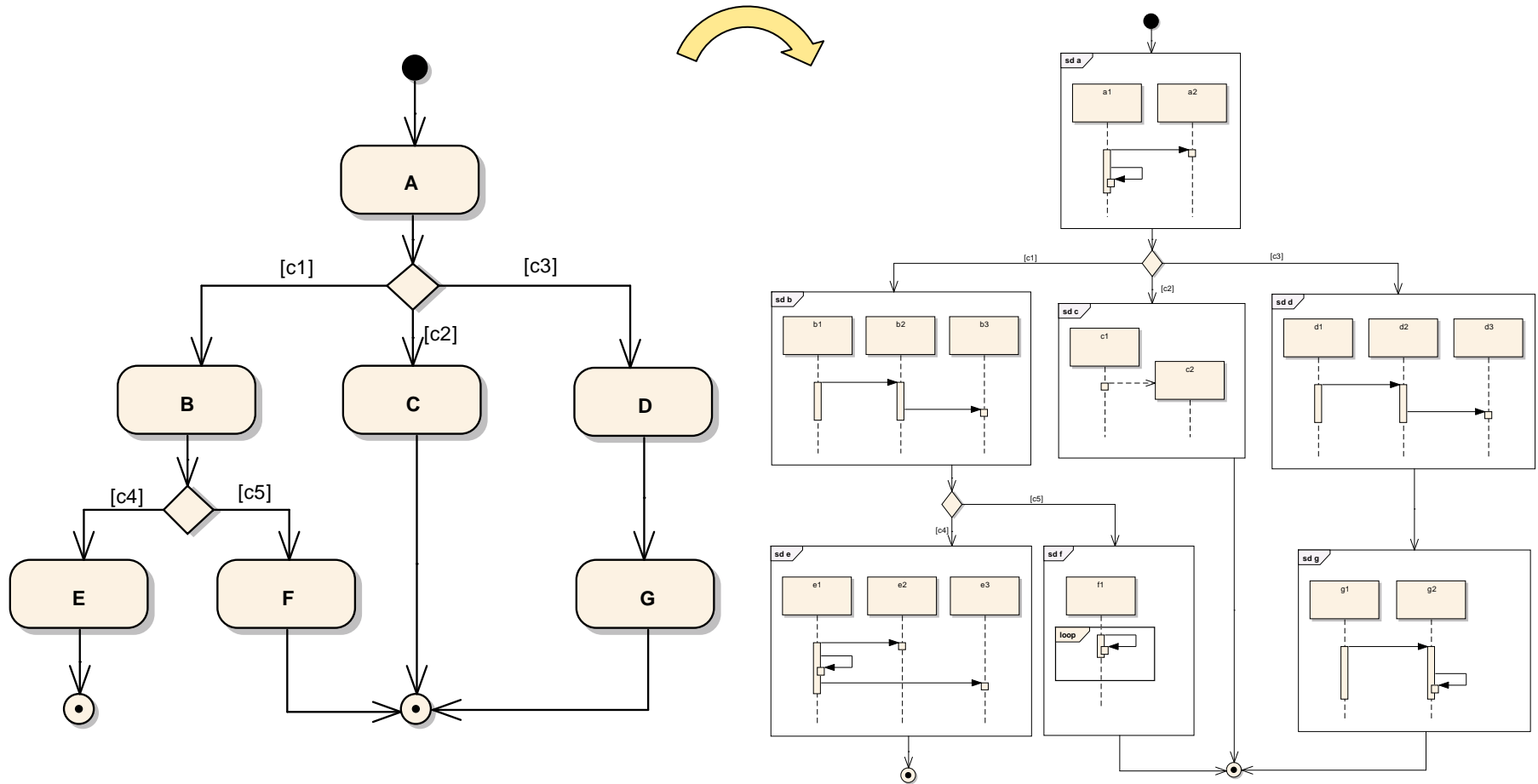
# Diagramas de Visão Geral da Interação

Era uma vez um DS que queria ser um DA...



- ...
- Assim ele modelaria todos os cenários de um processo,
- Seria visualmente mais simples, embora podendo ser mais detalhado,
- Teria melhores condições para geração automática de código com base em modelos,
- ...

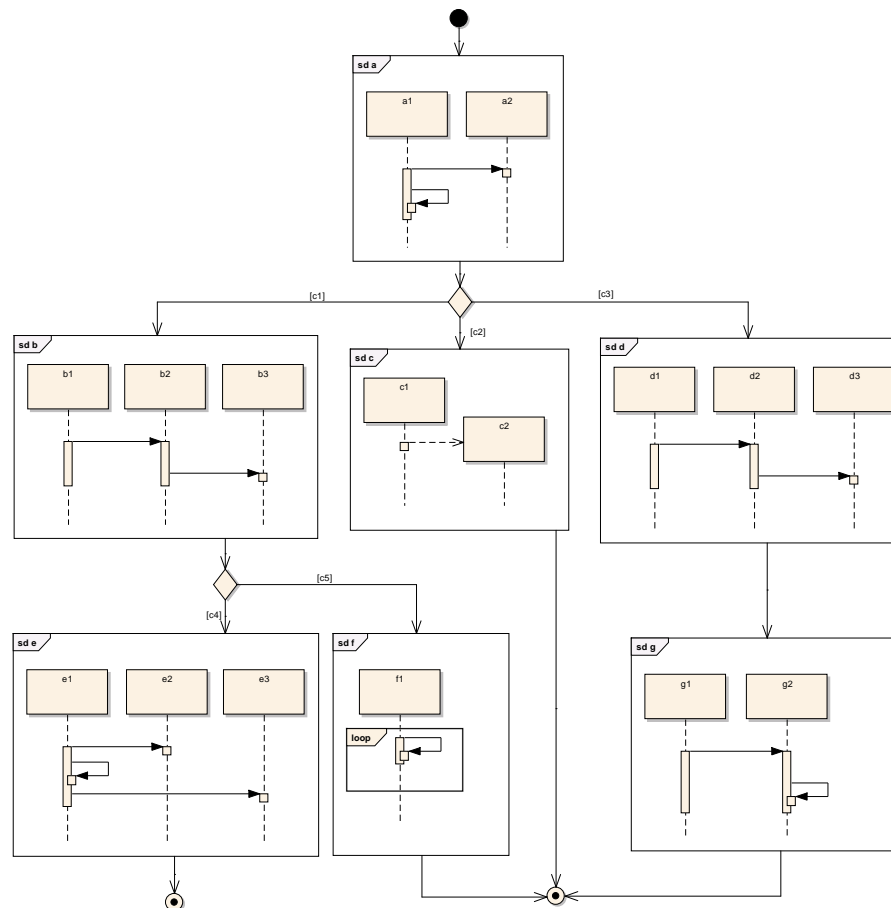
# Diagramas de Visão Geral da Interação



# Diagramas de Visão Geral da Interação

## Principais Características

- São combinações entre DAs e DSs;



Continua...

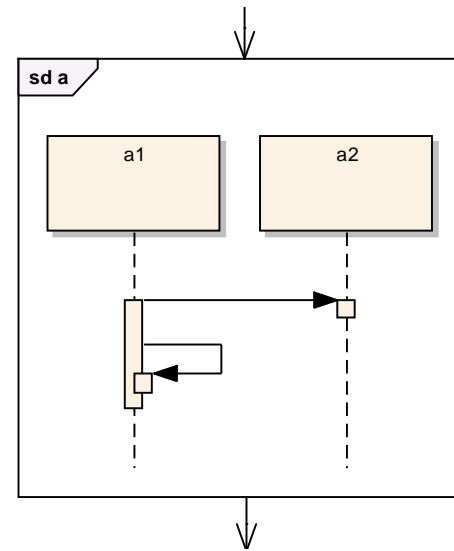
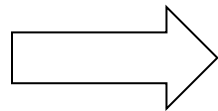
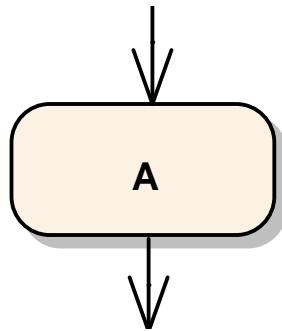




# Diagramas de Visão Geral da Interação

## Principais Características

- Os quadros de interação dos diagramas de sequência tomam os lugares das caixas de ações dos diagramas de atividade;

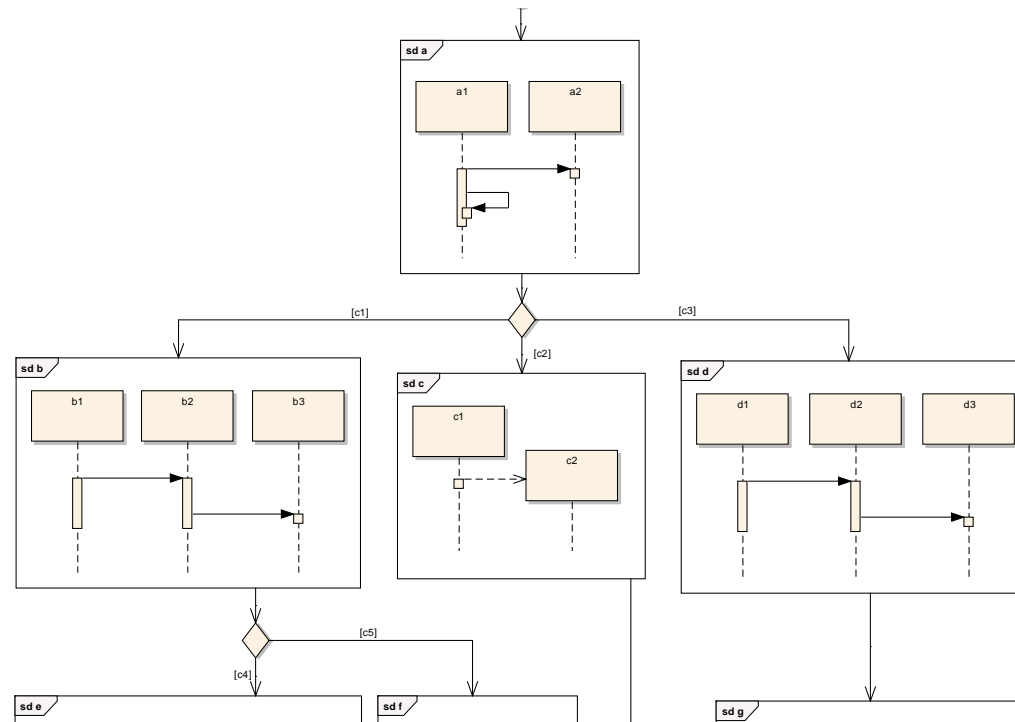


Continua...

# Diagramas de Visão Geral da Interação

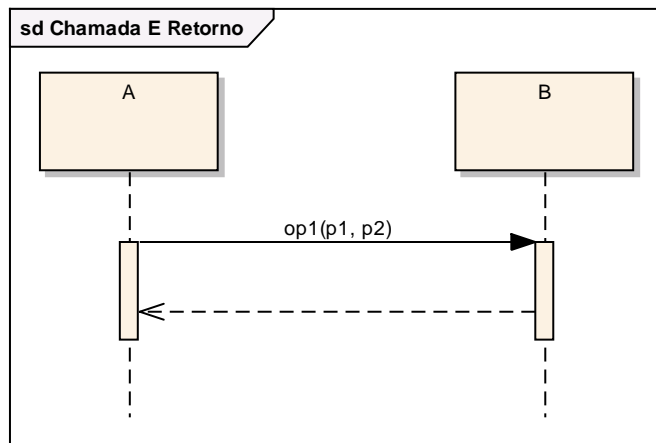
## Principais Características

- Fragmentos de baixo nível dos DSs são “costurados” usando-se a estrutura de controle de alto nível (fluxos, decisões, intercalações, separações e junções) dos DAs.

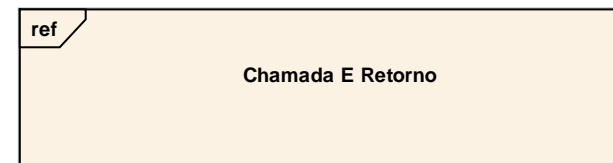


# Diagramas de Visão Geral da Interação

- Os diagramas de visão geral da interação usam quadros com trechos de colaborações e/ou referências para outros quadros;



OU



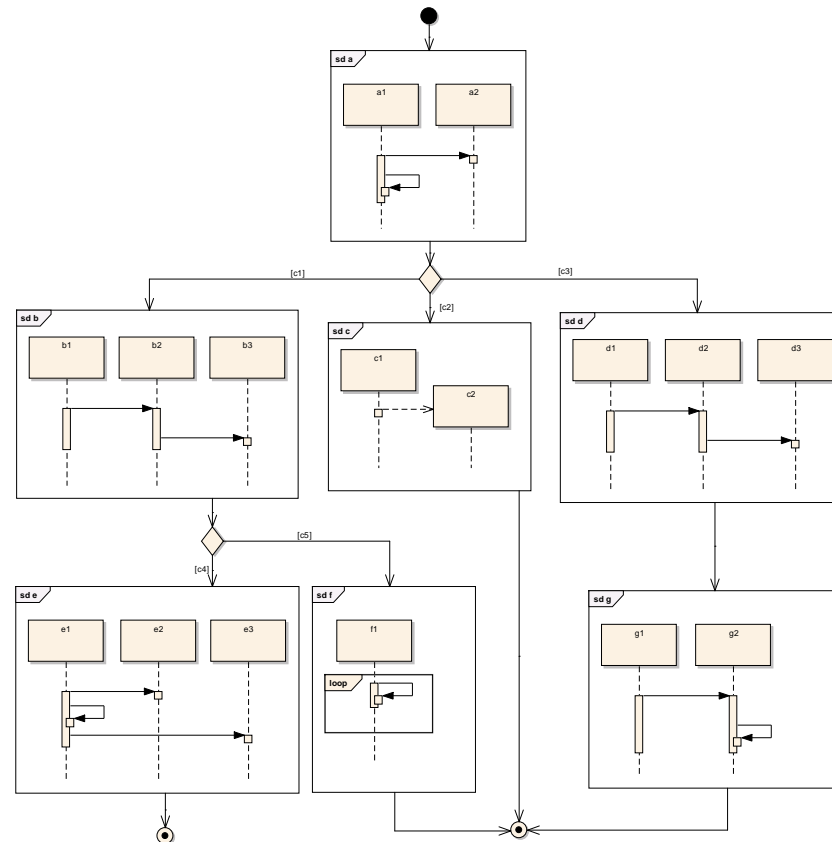
Ummm, com isso...

# Diagramas de Visão Geral da Interação

- Com isso...
  - Posso utilizar um Diagrama de Visão Geral da Interação “no estilo DA” para definir uma estrutura de controle de mais alto nível (ou seja, no lugar das ações do DA eu só tenho quadros REF), **E**
  - Posso detalhar esses pequenos fragmentos em outros diagramas de sequência do modelo.

# Diagramas de Visão Geral da Interação

- Podem ser visualmente mais complexos do que os DAs, se modelados da forma abaixo:



☹ Mas...

# Diagramas de Visão Geral da Interação

- Mas também podem ter a relativamente baixa complexidade visual dos DAs:
  - Se usarmos no lugar das atividades apenas referências a quadros especificados externamente (substituindo-se caixas de ações por quadros de operadores "ref" 😊);
  - Melhor ainda se não usarmos operadores "loop", "alt", "opt", "par" nos quadros externos, lançando mão dos mecanismos de controle de fluxo dos DAs, resultando em vários diagramas visualmente simples 😊.

## Diagramas de Visão Geral da Interação

- A idéia de refinarmos as ações de DAs, tornando-as simples o suficiente para que possam ser modeladas com o uso de fragmentos sem loops, alts, opts, pars, etc parece interessante;
- Poderíamos, com isso, pensar em ter “cartuchos” de colaborações reutilizáveis, facilitando, inclusive, a geração automática de código...

# Diagrama de Pacotes

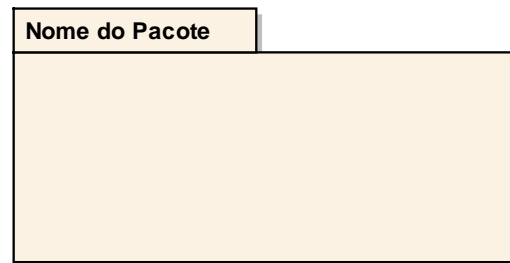


# Diagramas de Pacotes

- Pacotes são contêineres;
- Contêineres são usados para agrupar elementos da notação;
- Pacotes proveem:
  - Um *espaço de nomes* distinto para os elementos agrupados;
  - Uma forma de organizar o modelo, dividindo-o em partes;
  - Uma melhor visualização e compreensão do modelo.

# Diagramas de Pacotes

- Pacotes têm a forma gráfica de uma pasta suspensa;



- O nome do pacote é usualmente colocado no centro do retângulo, no topo ou dentro da etiqueta.

# Diagramas de Pacotes

- Como um pacote define um espaço de nomes, os identificadores dos elementos empacotados têm de ser distintos dentro do pacote;
- Apenas elementos *empacotáveis*\* de um modelo podem ser agrupados em pacotes nesse modelo.

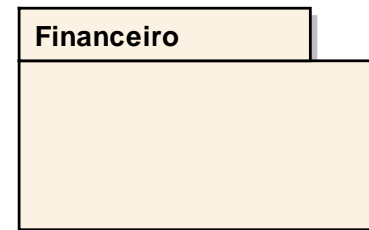
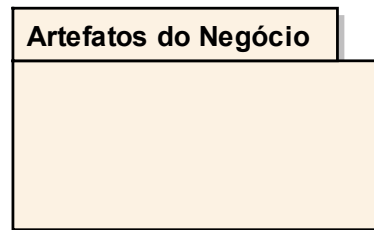
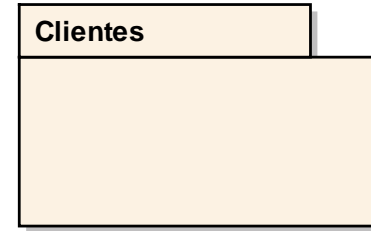
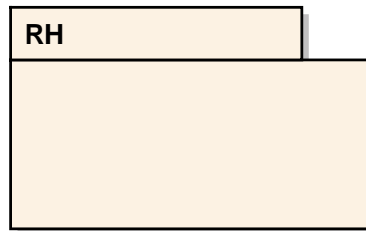
(\*) Aqueles que têm nomes e que podem ser empacotados, segundo a UML. ☺

# Diagramas de Pacotes

- Na prática, empacotamos:
  - Classes;
  - Atores;
  - Casos de uso; e
  - Outros pacotes.
- Pacotes contendo outros pacotes permitem formarmos hierarquias de pacotes.

# Diagramas de Pacotes

- Pacotes de classes:
  - Podem, simplesmente, agrupar conceitos fortemente relacionados no diagrama conceitual;

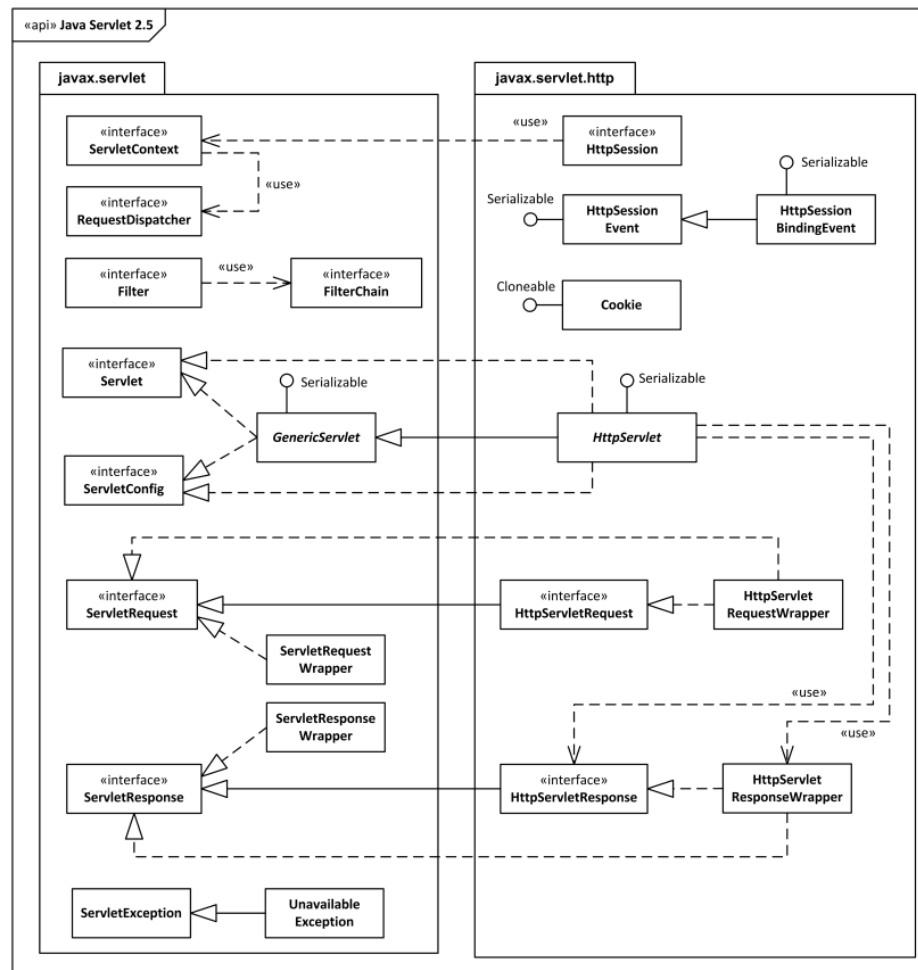


Continua...

# Diagramas de Pacotes

- Pacotes de classes:
  - Podem ser associados a componentes de implementação (por exemplo, as DLL/OCX do Windows) e/ou aos *packages* do Java, por exemplo (ver ilustração a seguir);

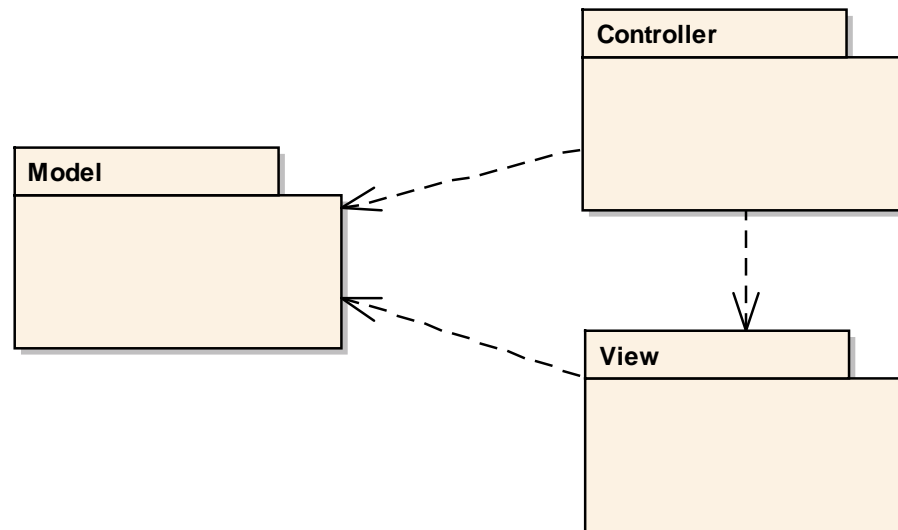
# Diagramas de Pacotes



Continua...

# Diagramas de Pacotes

- Pacotes de classes:
  - Um pacote pode estar associado a uma camada em um sistema cuja arquitetura é em camadas;



Continua...



# Diagramas de Pacotes

---

- Pacotes de classes:
  - Usualmente os pacotes contêm classes que cumprem uma função específica, bem definida, no sistema (classes que possuem alta coesão).

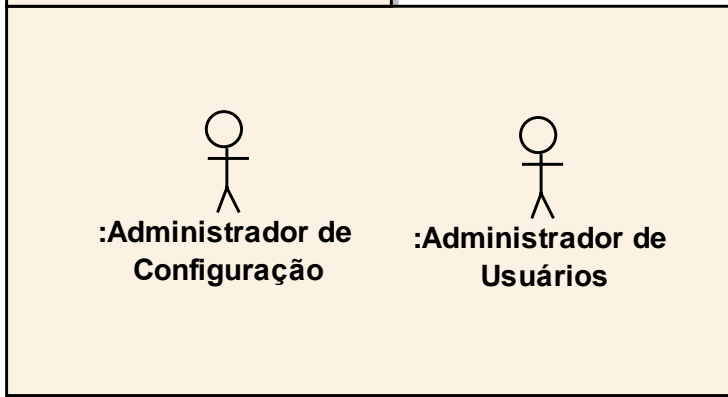
# Diagramas de Pacotes

- Pacotes de atores:
  - Podem agrupar atores com perfis de uso do sistema com muitas características em comum, como usuários de um mesmo setor em uma organização;
  - Podem agrupar atores com perfis administrativos com muitas características em comum, como usuários de níveis hierárquicos semelhantes em uma organização;
  - Especialmente interessantes quando há muita generalização/especialização de atores.

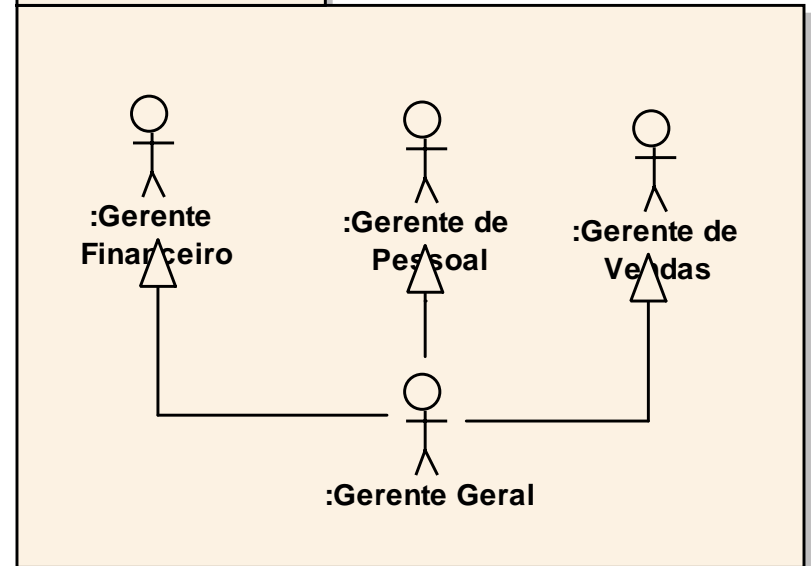


# Diagramas de Pacotes

Atores Administradores



Atores Gerentes



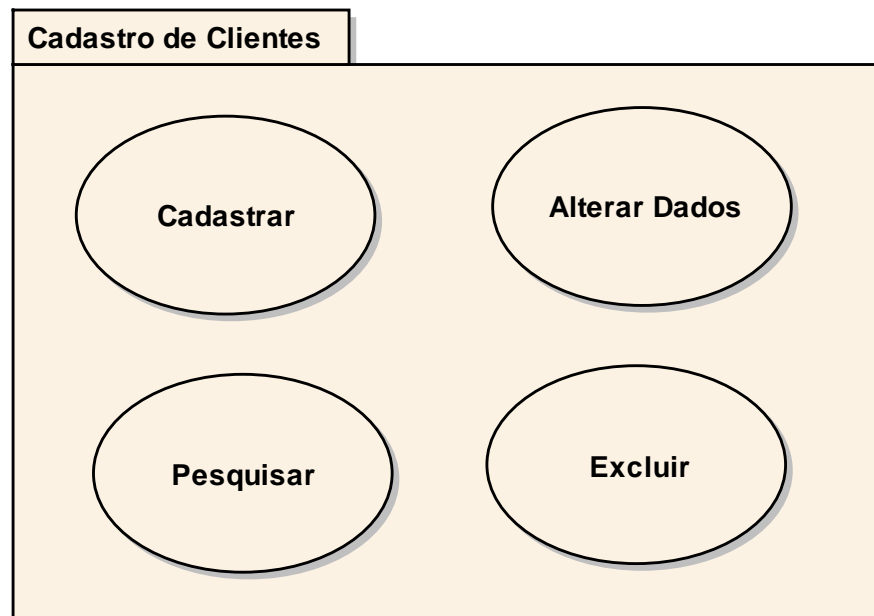
# Diagramas de Pacotes

---

- Pacotes de casos de uso:
  - Agrupam UCs coesos entre si, ou seja, que cumprem uma função bem definida no sistema.

# Diagramas de Pacotes

- Pacotes de casos de uso (exemplos):
  - Podem agrupar casos de uso de um mesmo grupamento CRUD;

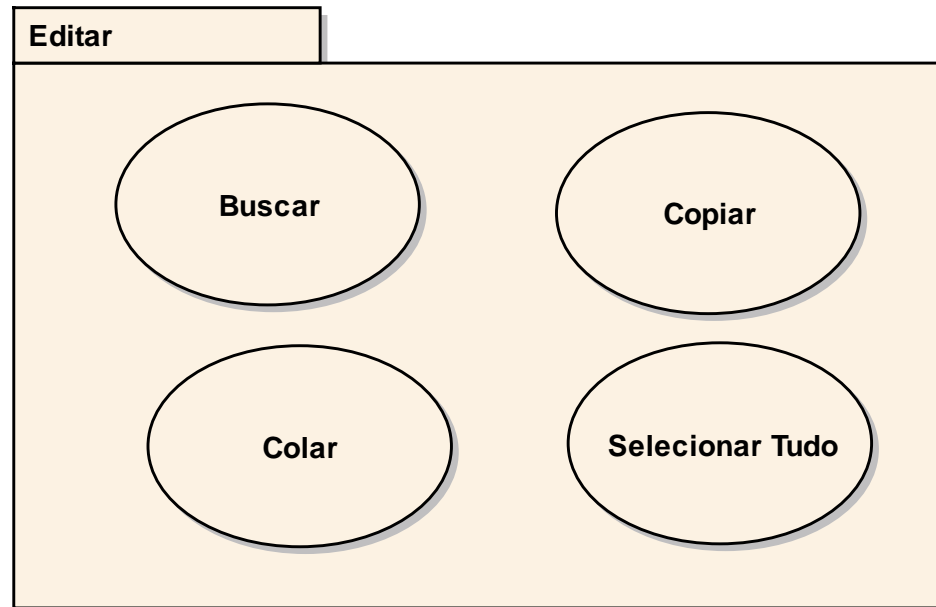


Continua...



# Diagramas de Pacotes

- Pacotes de casos de uso (exemplos):
  - Podem agrupar casos de uso que compõem um mesmo ramo na árvore de menus;

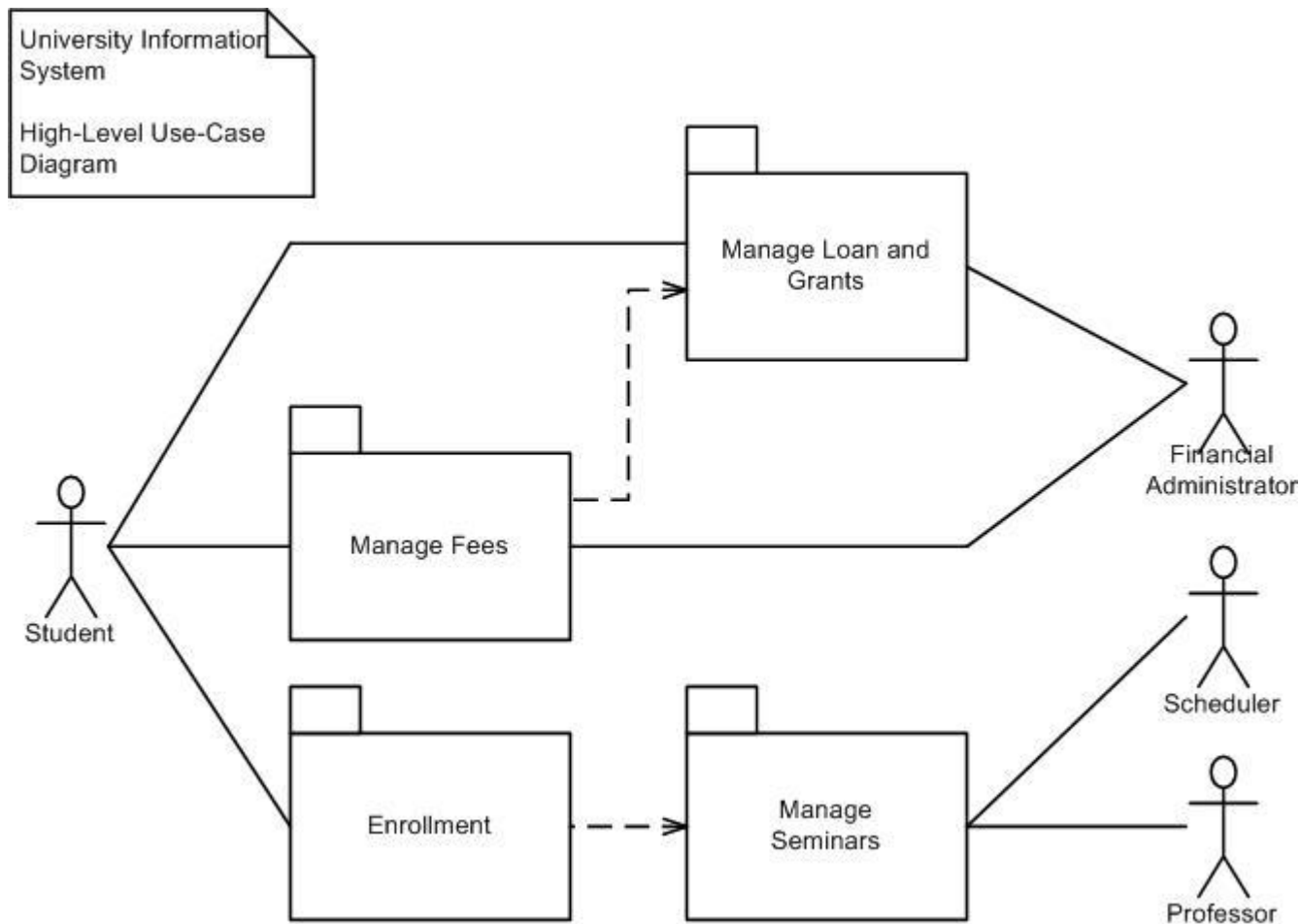


Continua...

# Diagramas de Pacotes

- Pacotes de casos de uso (detalhes):
  - Quando um pacote é associado a um ator, significa que o ator participa de todos os UCs do pacote de alguma forma;
  - Pacotes (de classes, de atores, de casos de uso...) podem se relacionar entre si por meio de relacionamentos de dependência (- - - - ->).

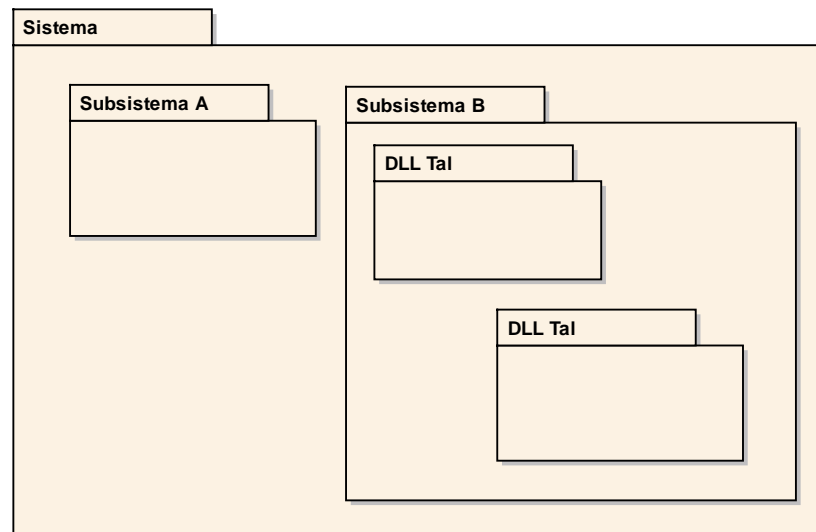
# Diagramas de Pacotes





# Diagramas de Pacotes

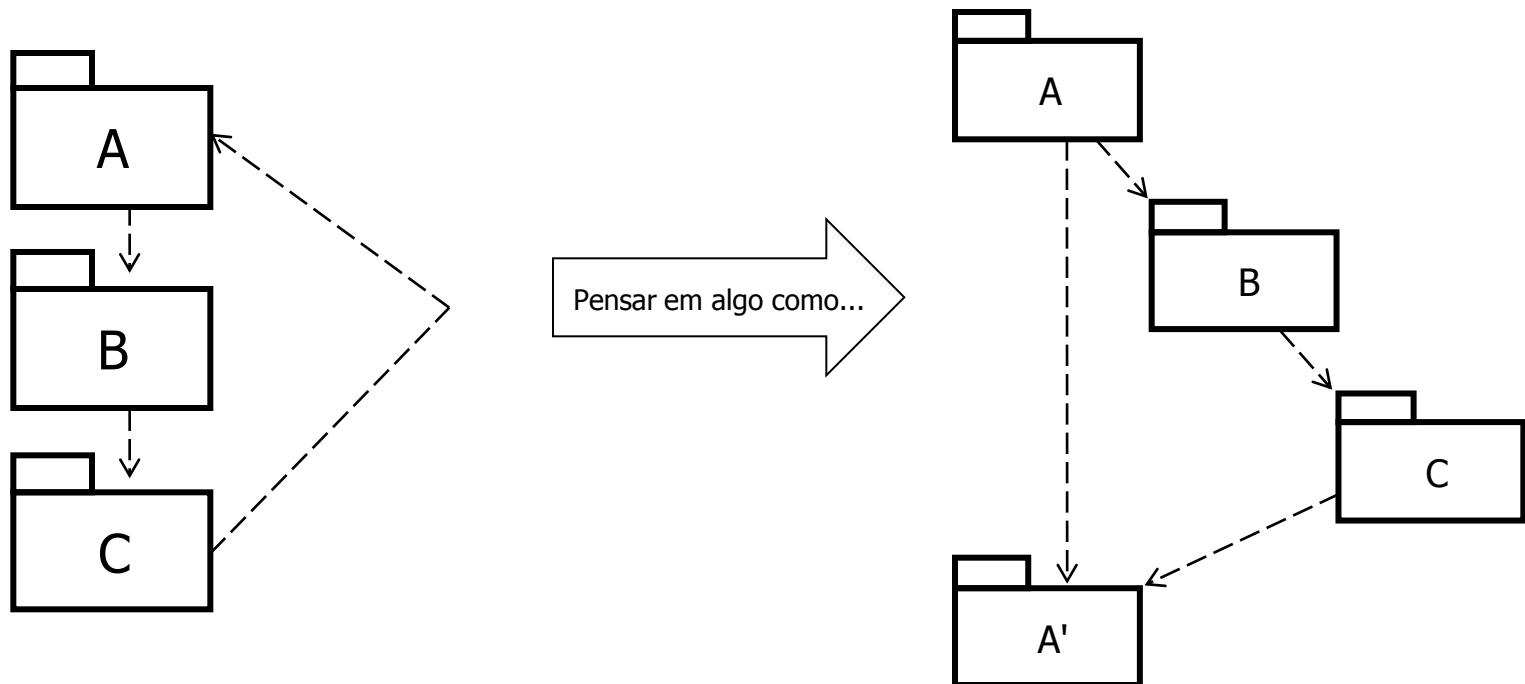
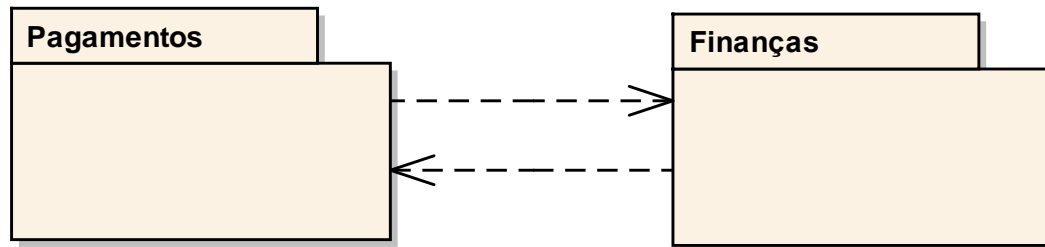
- Pacotes de outros pacotes:
  - Definem uma estrutura hierárquica, como, por exemplo, componente-subsistema-sistema.



# Diagramas de Pacotes

- Após organizarmos as partes de um sistema em pacotes, é sempre interessante ilustrar essa organização em um ou mais diagramas de pacotes que mostrem seus pacotes e os relacionamentos de dependência
  - Esses diagramas são colocados na documentação antes dos demais diagramas;
  - Dependências mútuas/circulares não são bem vistas por questões de manutenibilidade e porque reusar um pacote demanda que usemos o(s) outro(s) (**ver a diante**).

# Diagramas de Pacotes



# Diagramas de Pacotes

- Ao movermos ou deletarmos um pacote, é esperado que todos os elementos em seu interior se movam ou sejam deletados, respectivamente\*.

(\*) Esse, por sinal, é o comportamento esperado de todo contêiner na UML.

# Diagramas de Pacotes

- A organização em pacotes é, antes de tudo, uma organização lógica entre partes.
- ⇒ Não tem, necessariamente, relação com a organização física do software na implementação.

# Diagrama de Componentes

# Diagramas de Componentes

- Um componente corresponde a um agrupamento físico de classes do modelo
  - Tem um propósito e provê uma ou mais funcionalidades para o sistema;
  - É entendido como uma “caixa preta” reutilizável em um sistema;
  - Tem uma interface bem definida;
  - Tem a ver com como projetamos e implementamos o sistema.
- Componentes são também contêineres.

# Diagramas de Componentes

- Um componente pode corresponder a
  - Um subsistema;
  - A um elemento com características e funcionalidade próprias que possa ser adicionado ao nosso sistema;
  - Qualquer outra porção reutilizável de um sistema.
- É o elemento estrutural básico de softwares concebidos segundo a *orientação a componentes*.



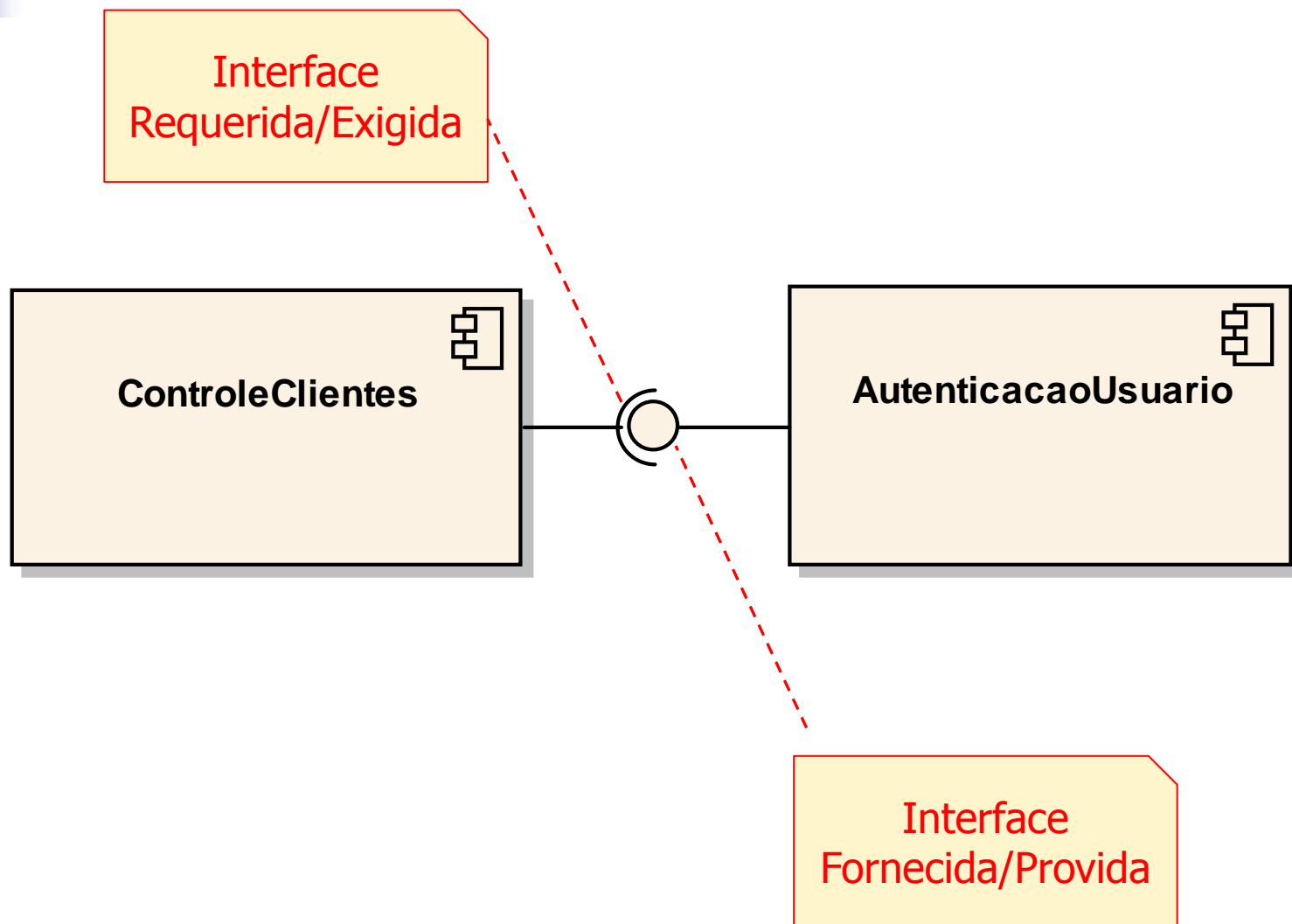
# Diagramas de Componentes

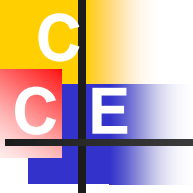
- Exemplos de componentes comumente construídos nas organizações são:
  - Componentes para autenticação de usuários;
  - Componentes para tratamento de datas, de cadeias de caracteres e outras utilidades;
  - Componentes para manipulação de dados de clientes;
  - Componentes para manipulação de ordens de serviço;
  - Componentes para manipulação de faturas...

# Diagramas de Componentes

- Um diagrama de componentes contém:
  - Um conjunto de símbolos que representam os componentes;
  - Conectores que estabelecem os relacionamentos de dependência (uso) entre componentes;
  - Símbolos que representam as interfaces fornecidas e requeridas entre componentes.

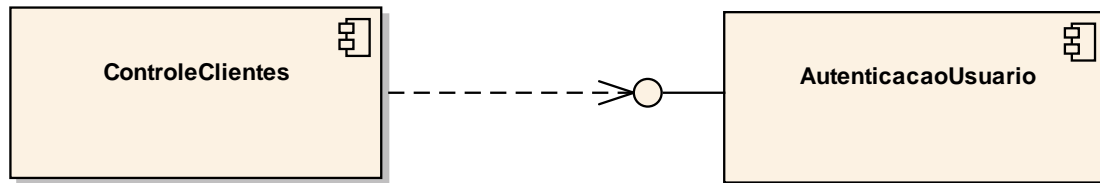
# Diagramas de Componentes





# Diagramas de Componentes

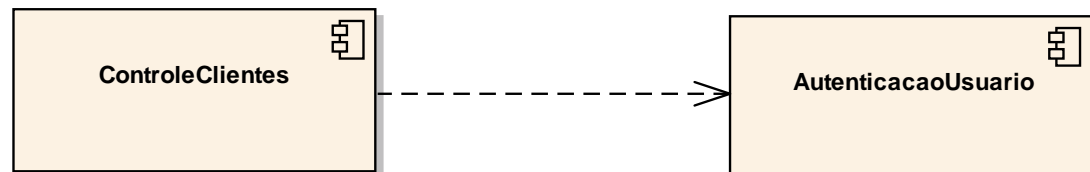
Notações alternativas:



ou



ou ainda



# Diagrama de Instalação

# Diagramas de Instalação

- Há situações em que necessitamos representar que partes do software processam em que partes do hardware e quais são os mecanismos de comunicação entre essas partes
  - É o caso, por exemplo, da definição da arquitetura de sistemas distribuídos.

# Diagramas de Instalação

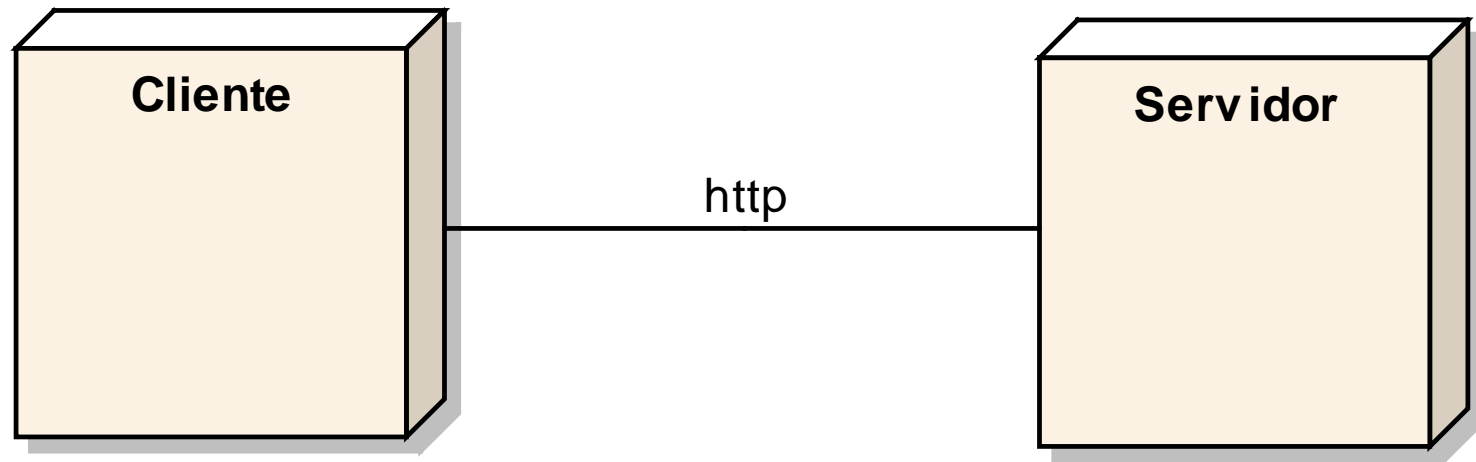
- Um diagrama de instalação consiste de um conjunto de caixas paralelepipedicas associadas por meio de links de comunicação
  - As caixas paralelepipedicas representam os nós de processamento;
  - As associações usualmente indicam o protocolo usado para a comunicação entre os nós.

# Diagramas de Instalação

- Os nós podem conter as representações dos componentes contidos neles, conforme o sistema foi/será implementado.
- As caixas representando os nós são, portanto, outros contêineres da UML.



# Diagramas de Instalação





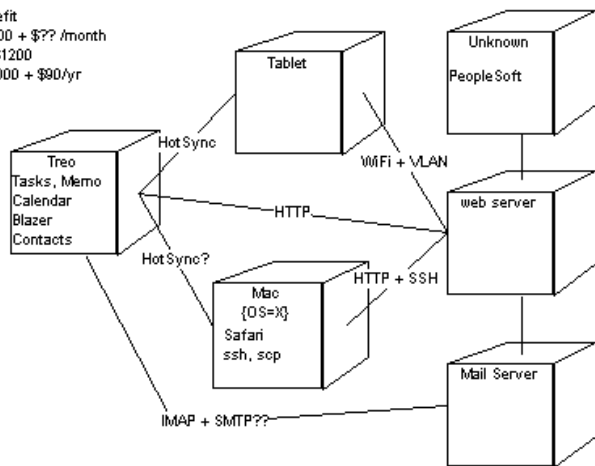
# Diagramas de Instalação

- Diagramas de instalação são bastante úteis em sistemas concebidos segundo arquiteturas complexas e pouco usuais
  - Sistemas compostos de várias tecnologias distintas
  - Sistemas integrando outros sistemas legados que demandam componentes de mediação entre eles), o diagrama pode ser bastante complexo.

# Diagramas de Instalação

UML Deployment Diagram analysing future hardware

Cost/Benefit  
 Treo -- \$400 + \$?? /month  
 Tablet -- \$1200  
 Mac -- \$2000 + \$90/yr



Rational Rose Student Edn.

