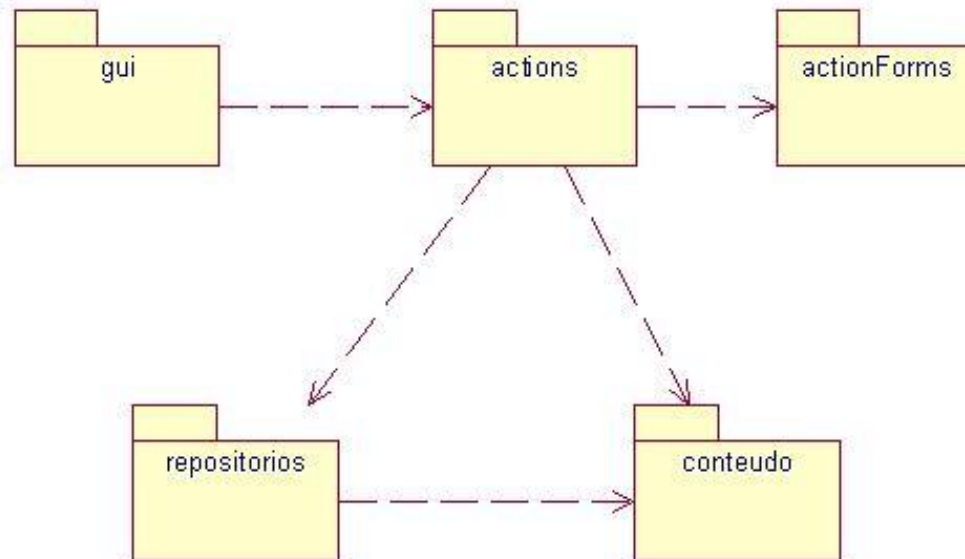

Diagrama de Pacotes

Diagrama de Pacotes

- Um diagrama de pacotes mostra pacotes e as relações entre eles.
- Representa subsistemas ou módulos englobados por um sistema, de forma a determinar as partes que o compõe.



Diagramas de pacotes

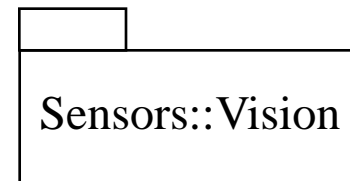
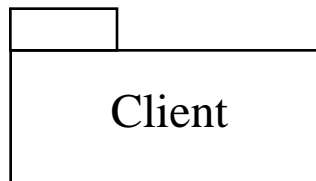
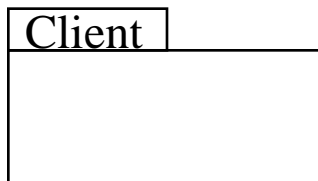
- Pacotes e relações entre pacotes aparecem em vários diagramas da UML:
 - **Pacotes de classes (pacotes lógicos) - em diagramas de classes.**
 - Pacotes de componentes – em diagramas de componentes.
 - Pacotes de nós – em diagramas de distribuição.
 - Pacotes de casos de uso – em diagramas de casos de uso.
-

Diagramas de pacotes

- **Benefício:** organizar elementos em pacotes aglutina elementos em abstrações maiores, dando uma visão de alto nível e permitindo visualizar um modelo em agrupamento simples.
 - Organizar a notação e os diagramas em pacotes é uma das boas práticas sugeridas pelo livro “UML 2.0 for Dummies”. Segundo os autores, em projetos grandes, conforme o número de classes aumenta, você precisará de ajuda para organizar os seus diagramas.
-

Pacotes

- Um **pacote** (*package*) pode ser definido como um mecanismo de agrupamento genérico em UML.
- Um pacote é um conjunto de elementos do modelo de qualquer tipo, tais como classes, casos de uso, diagramas de interação, ou mesmo outros pacotes.
- Notação: pasta com o nome no interior ou na aba.



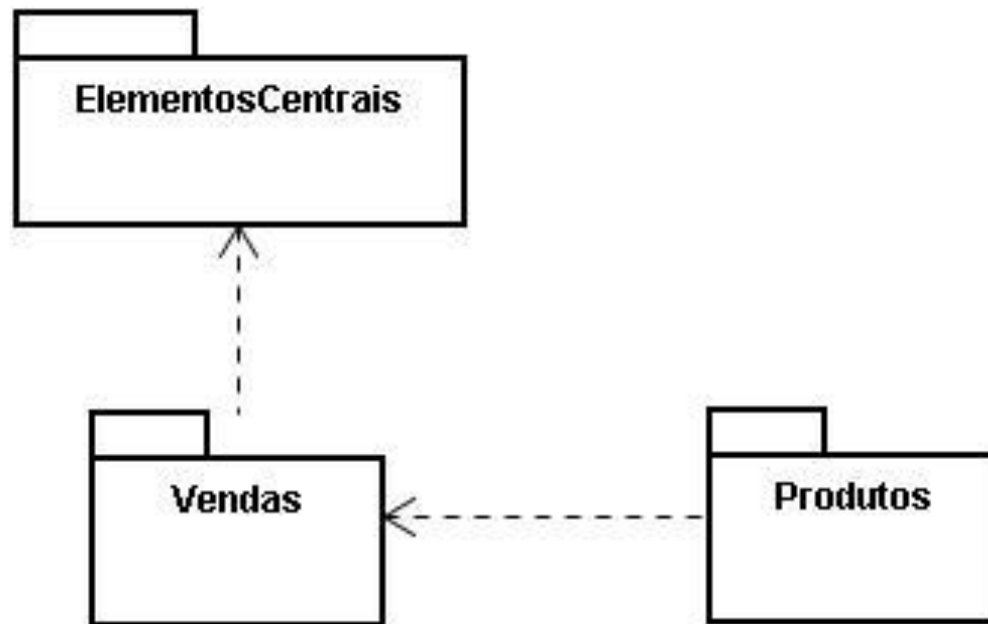
Pacotes lógicos

- Um **pacote lógico** (ou módulo lógico) é um agrupamento lógico de classes e relações entre essas classes.
 - Dividir o sistema em pacotes lógicos é como dividir responsabilidades.
 - Corresponde ao conceito de *package* em Java ou de *namespace* em C++ e C#.
-

Dependências entre Pacotes

- Um dependência de pacote indica que um ou mais elementos do pacote dependente têm, de alguma forma, conhecimento dos elementos do pacote-alvo, do qual o pacote em questão depende.
 - Uma alteração do pacote de destino afeta o pacote de origem (dependente).
-

Exemplo



Exemplo

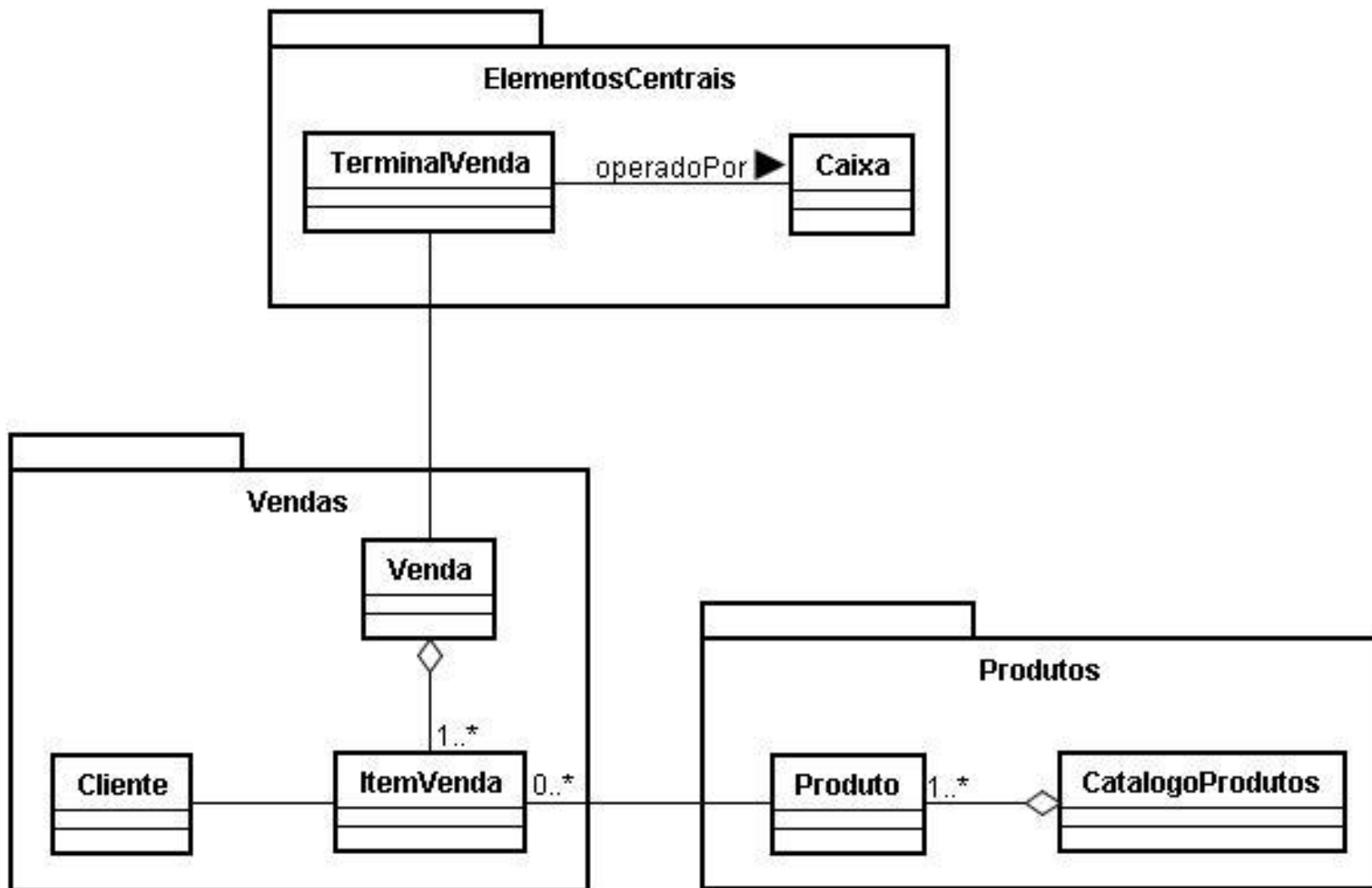


Diagrama de Interação

- Seqüência, Comunicação, **Tempo** e **Visão Geral** -

Diagramas de Interação

- Mostram a troca de mensagens em uma colaboração (um grupo de objetos que cooperam), para atingir um objetivo.
 - São a base para a definição de operações nas classes. A direção de uma mensagem indica a classe que deve conter a operação que trata a mensagem correspondente.
 - Um Diagrama de Interação é composto por:
 - Objetos
 - Ligações
 - Mensagens
-

Diagramas de Interação

- Permitem examinar:
 - A troca de mensagens entre os objetos sob o ponto de vista temporal;
 - As interações dos objetos dentro do contexto de suas relações estruturais, especificando as mensagens trocadas em função destas relações.
 - Principais aplicações:
 - Visualização, especificação, construção e documentação da dinâmica de uma sociedade particular de objetos.
 - Podem ser usados para modelar o fluxo de controle de um caso de uso. No contexto de um caso de uso, uma interação representa um cenário.
-

Tipos de Diagramas de Interação

- Na UML 2.5, são definidos 4 tipos de diagramas de interação:
 - **Sequência:** enfatiza o ordenamento das mensagens trocadas entre os objetos.
 - **Comunicação:** enfatiza a organização estrutural dos objetos que trocam mensagens.
 - **Tempo:** descreve as mudanças no estado ou condição de um objeto de uma classe durante um tempo.
 - **Interação Geral:** definidos para visualizar o fluxo geral de controle, logo, não mostram em detalhes as mensagens trocadas pelos objetos.
-

Diagrama de Tempo (*Timing*)

Diagrama de Tempo

- Descreve a mudança no estado ou condição de um objeto de uma classe durante um tempo.
- Foi projetado para especificar as restrições de tempo relacionadas ao envio e recebimento de mensagens durante uma interação*.
- “Como os diagramas de seqüência focam no ordenamento das mensagens trocadas entre os objetos e os diagramas de comunicação mostram os links entre os participantes, não há como detalhar informações de tempo”**.
 - Uma interação que não pode levar mais de 10 segundos para ser completada.
 - Uma mensagem que não pode levar mais de 10 segundos para apresentar o valor de retorno.
 - Um objeto que não pode ficar mais que um determinado tempo no mesmo estado.

* Livro “UML Bible”.

** Livro “Learning UML 2.0”.

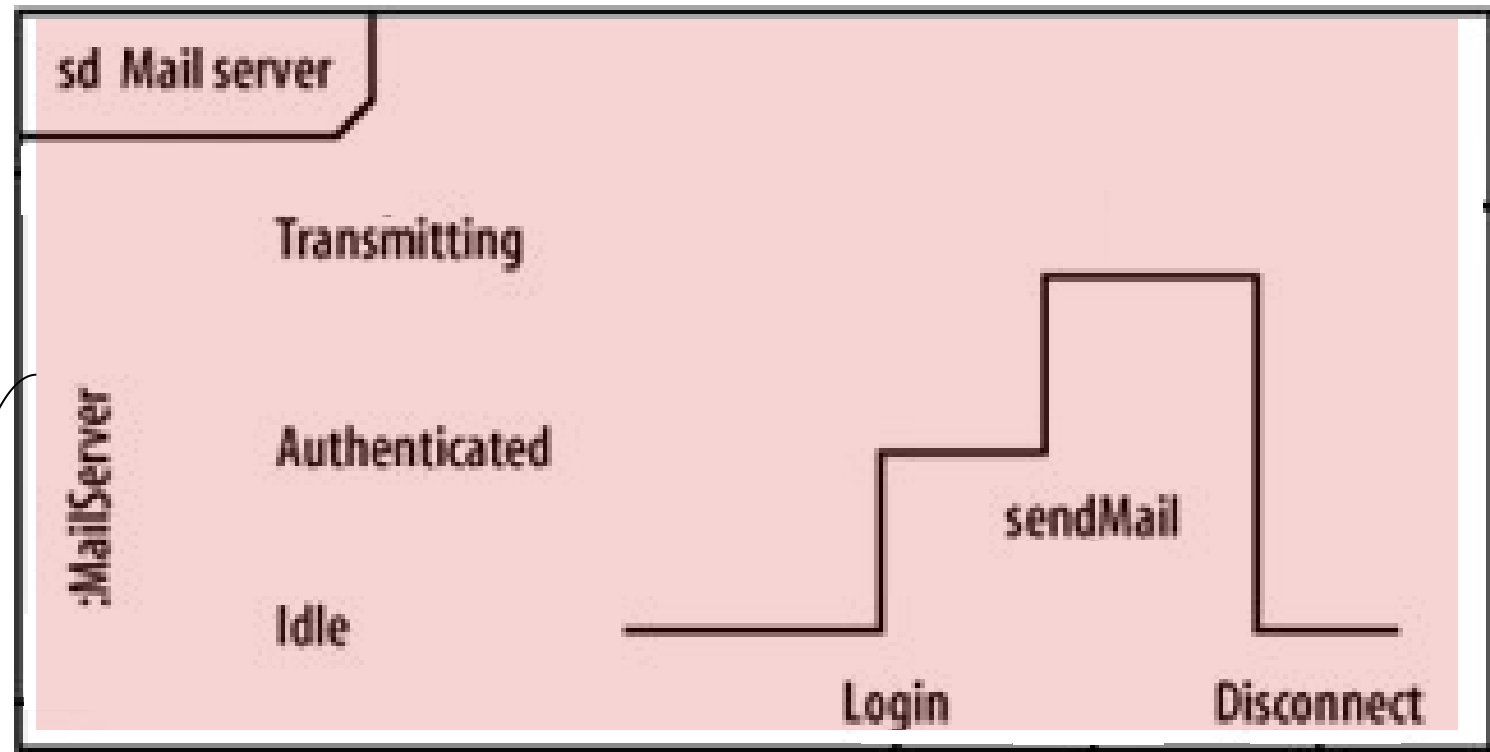
Diagrama de Tempo

- Aplicações mais comuns*:
 - Para mostrar, de forma detalhada, as restrições de tempo relacionadas as mensagens trocadas entre os objetos.
 - Para mostrar quando ocorrem mudanças relacionadas ao tempo na linha de vida de um objeto.
- Principais usos*:
 - Aplicações de tempo-real.
 - Sistemas embarcados.

Diagrama de Tempo

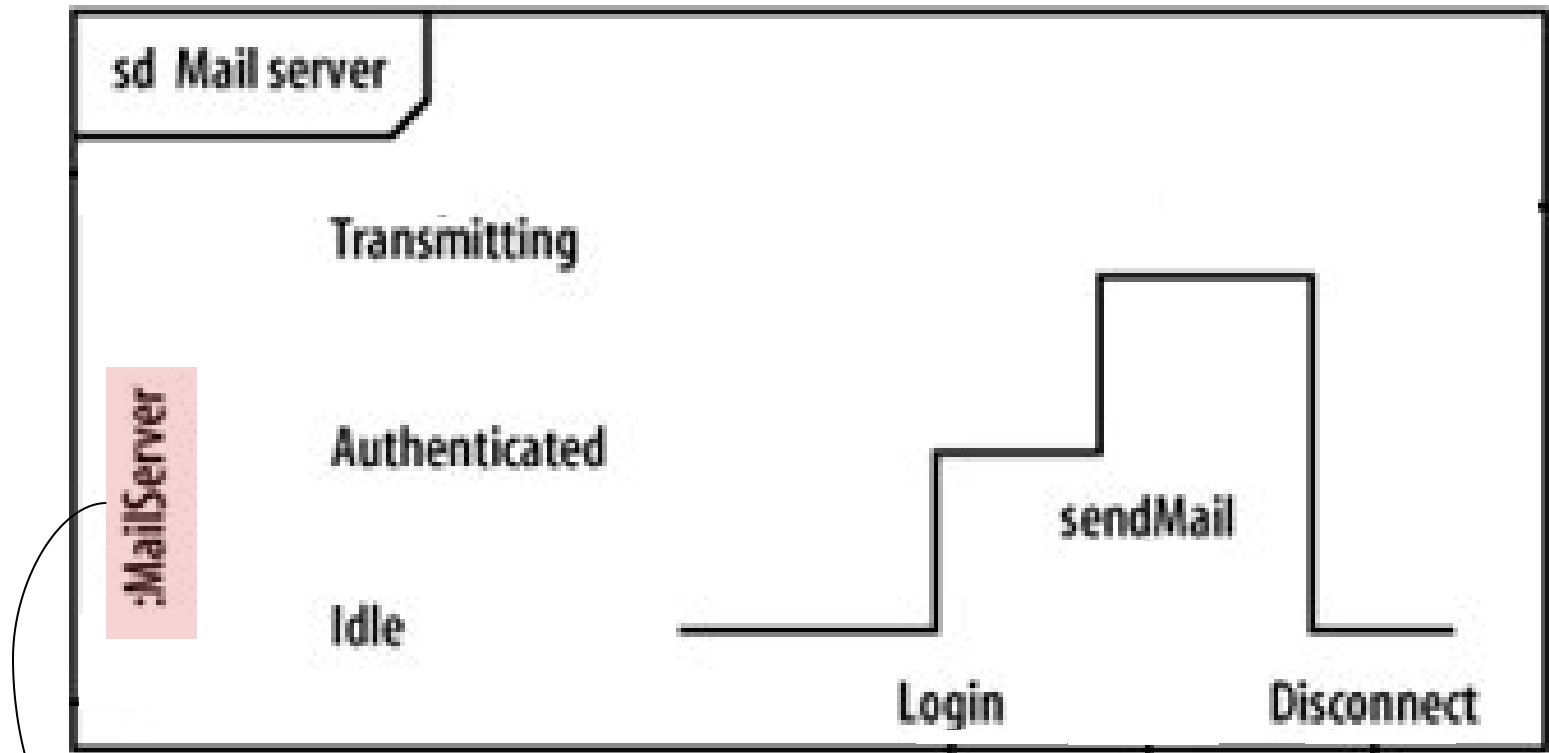
- Overview da Notação -

Lifeline



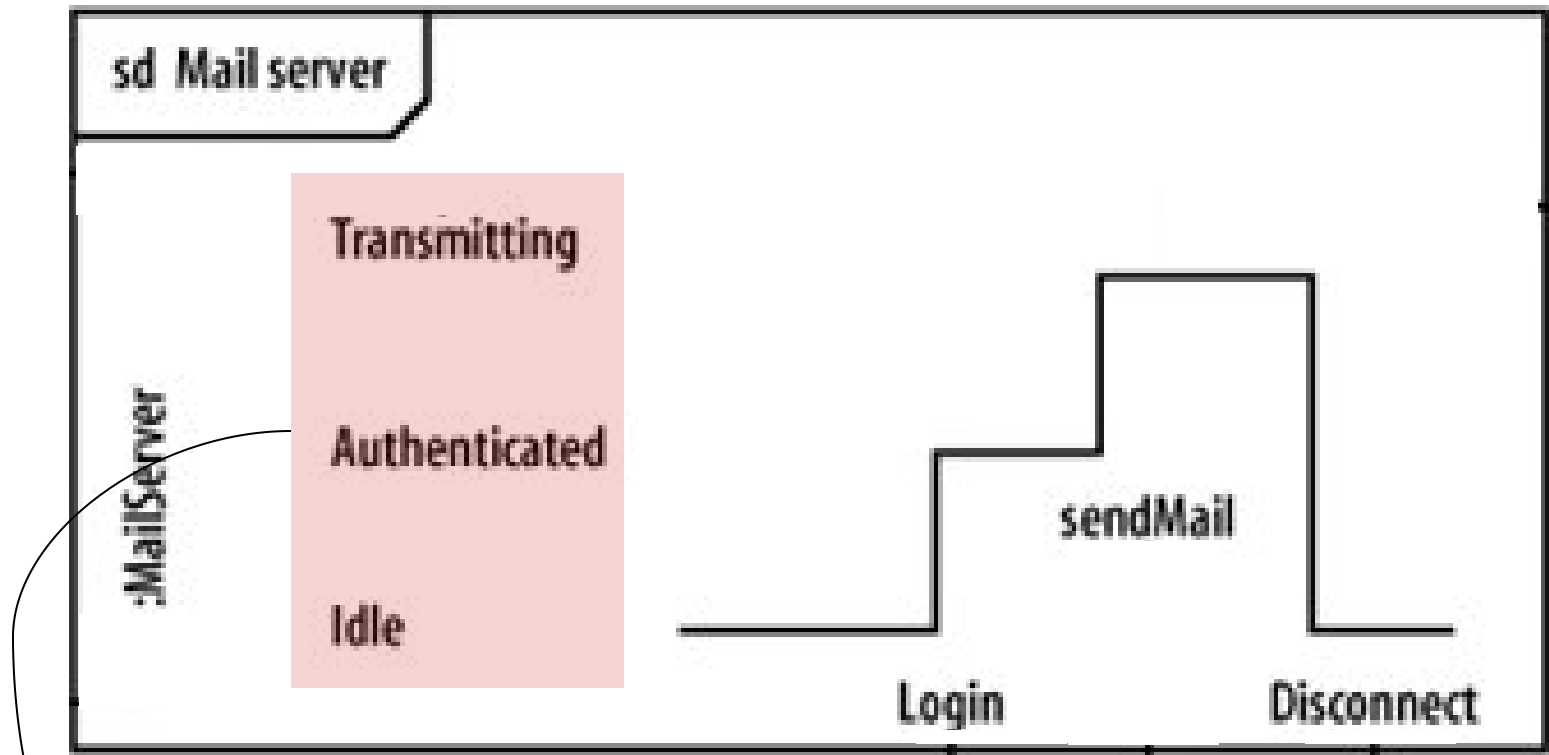
Lifeline: retângulo onde são ilustrados os estados de um objeto (participante).

Participantes



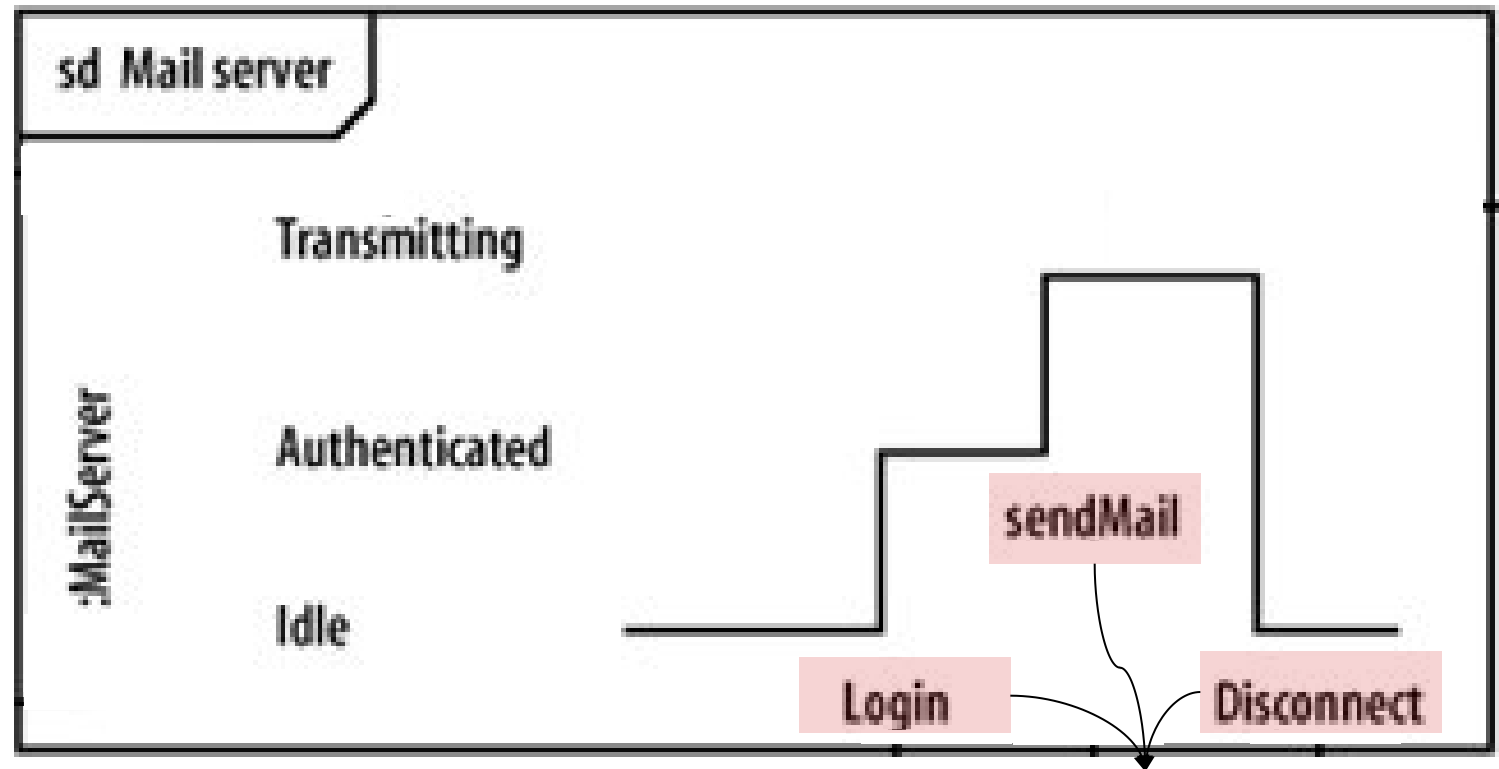
Nome do participante envolvido na interação.

Estados



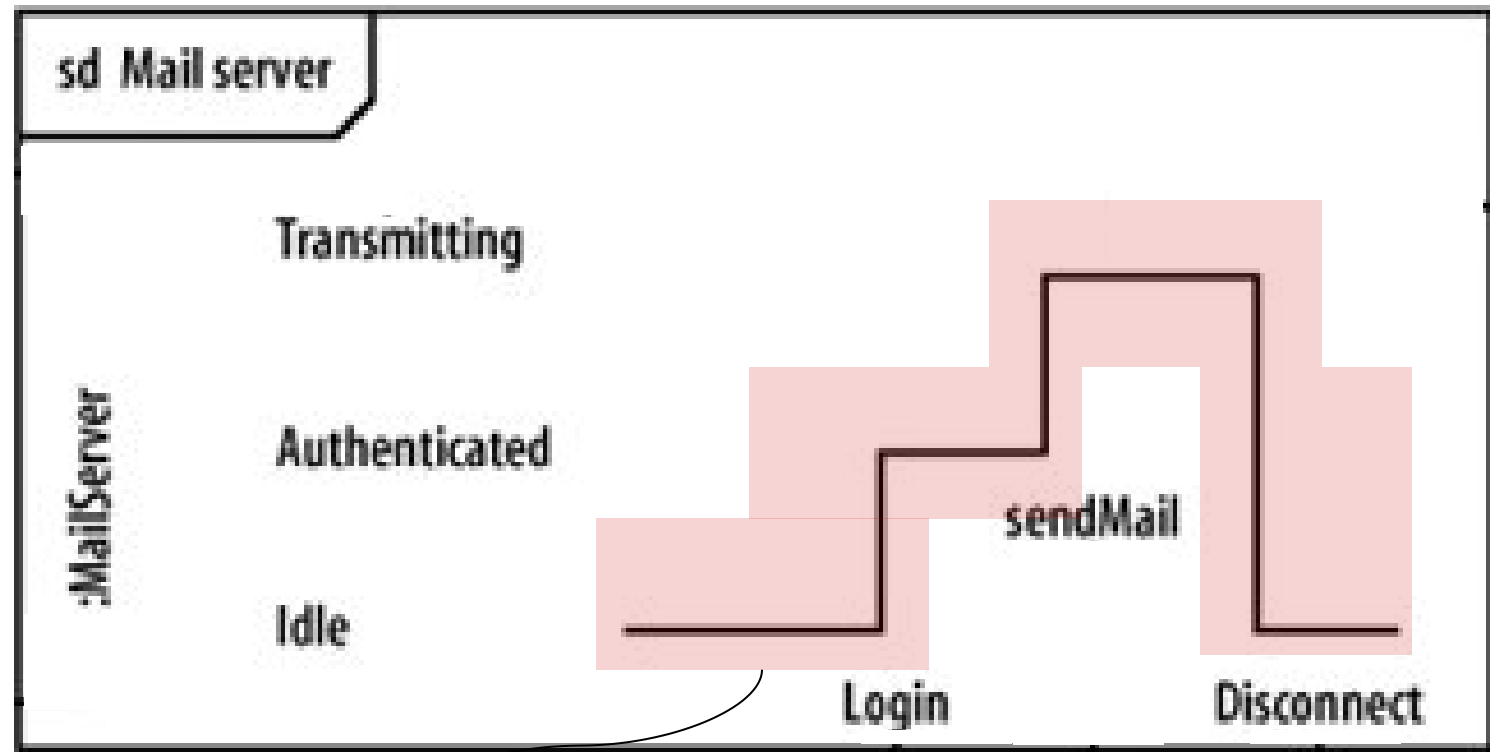
Estados ou condições: estados que um participante pode estar durante a sua execução.

Mensagens



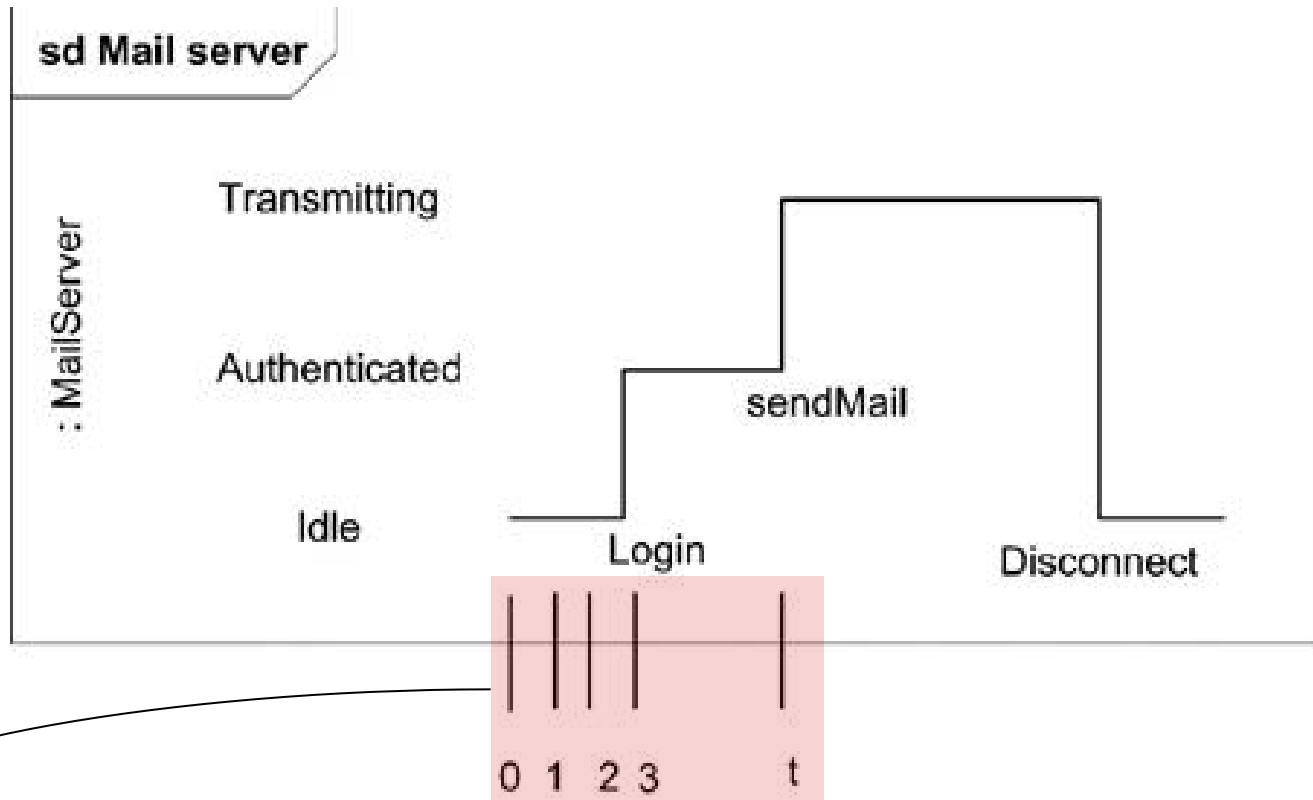
Mensagens que “despertam” a troca de estados

Estado corrente



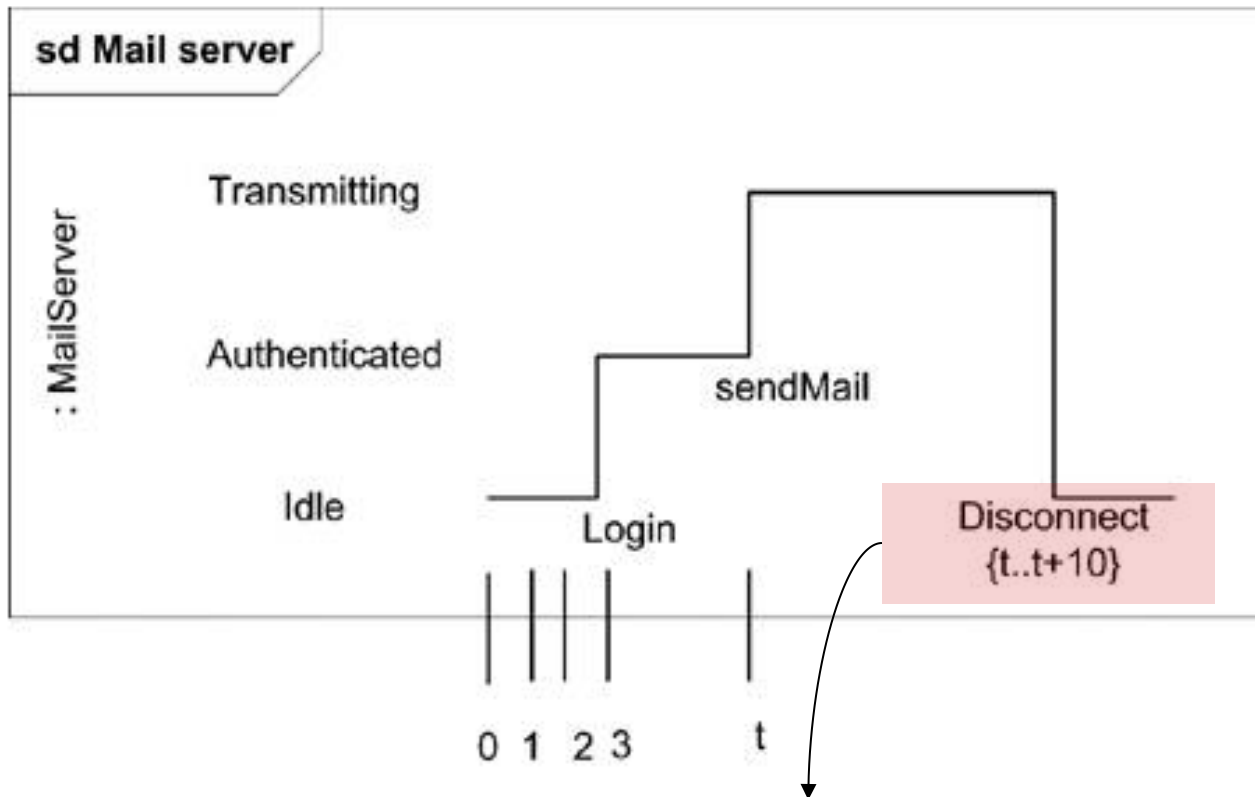
Linha indicando o estado corrente do participante em um determinado tempo.

Tempo



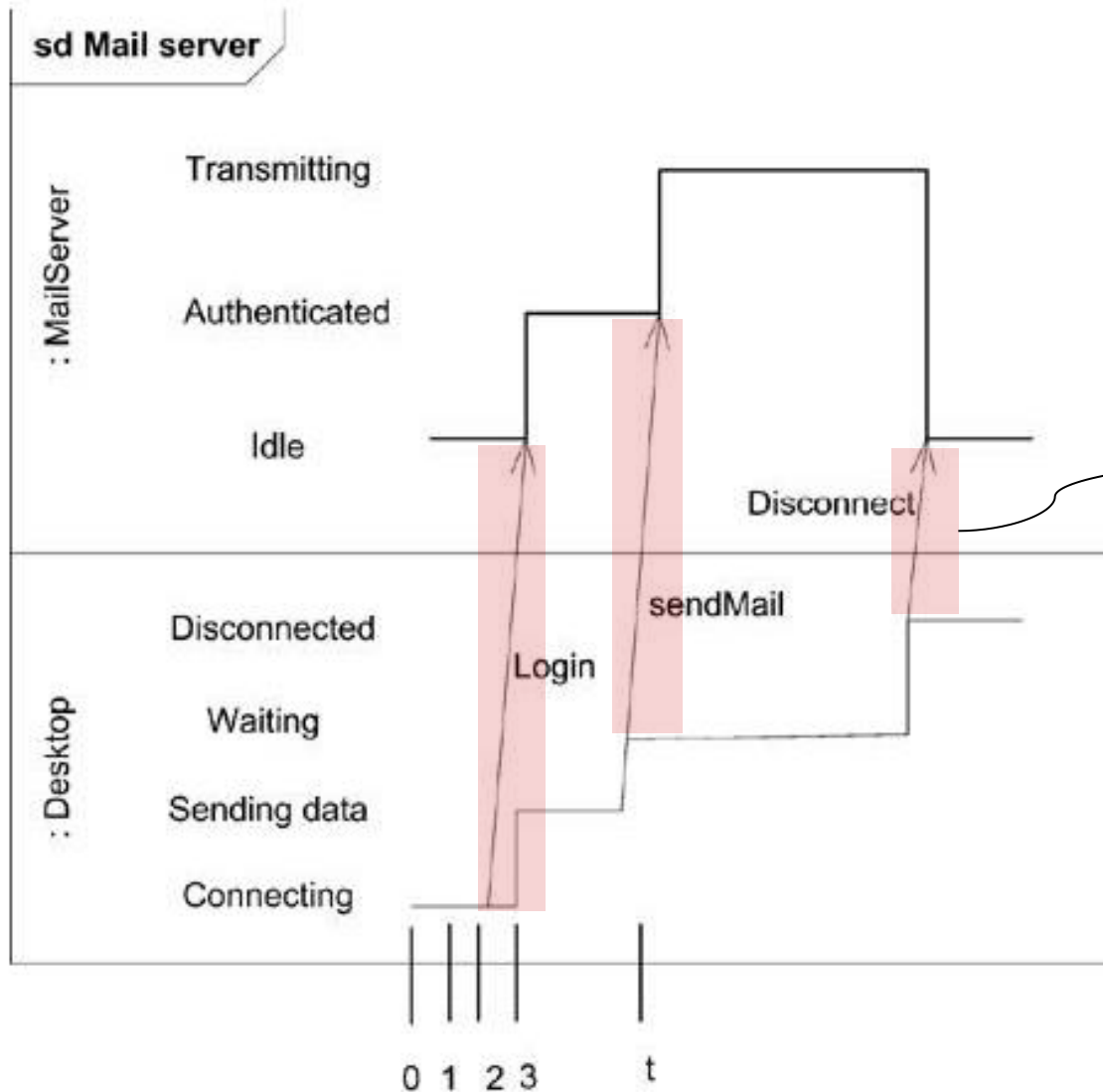
Pode-se expressar o tempo exato ou usar indicadores de tempo relativos.
→ t representa um ponto de tempo de interesse para o sistema, cujo valor exato não se sabe.

Restrições de Tempo



Utilizando t como referência, pode-se definir restrições de tempo.

Múltiplos participantes



Pode-se mostrar a troca de mensagens entre objetos utilizando uma seta (o início da seta indica quando a mensagem é enviada e o final quando a mensagem é recebida).

Diagrama de Interação – Visão Geral

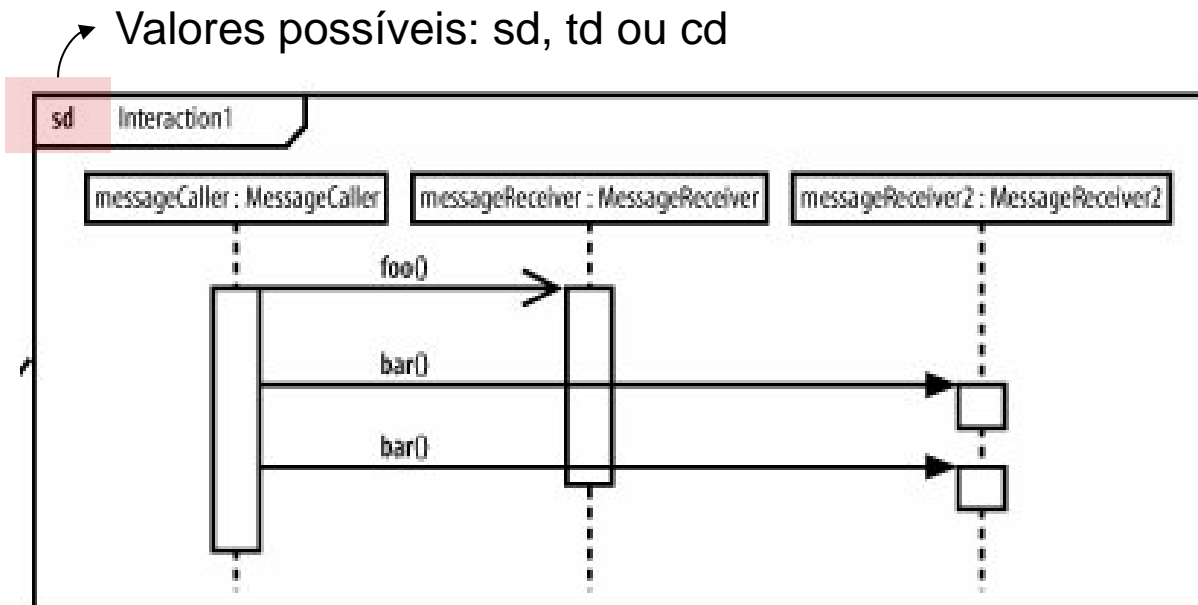
(Interaction Overview Diagram)

Diagrama de Interação – Visão Geral

- Representa interações usando uma simplificação da notação do diagrama de atividades. Foram definidos para visualizar o fluxo geral de controle, logo, eles não mostram em detalhes as mensagens trocadas entre os objetos.
 - Suporta todos os mecanismos de controle de fluxo do diagrama de atividades, como pontos de junção e bifurcação, nodos de decisão e união, nodos de início e final.
-

Diagrama de Interação – Visão Geral

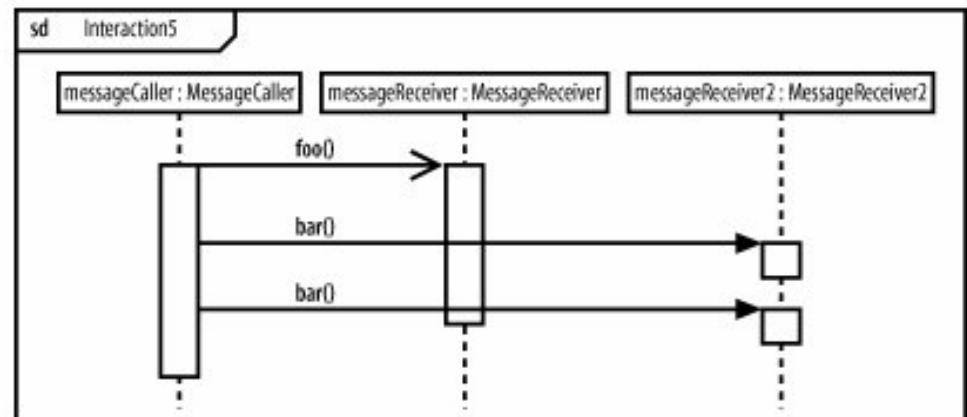
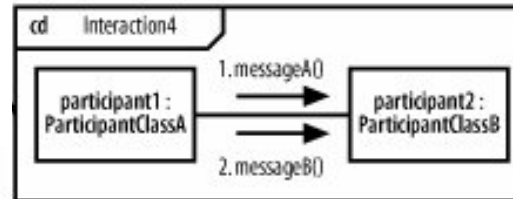
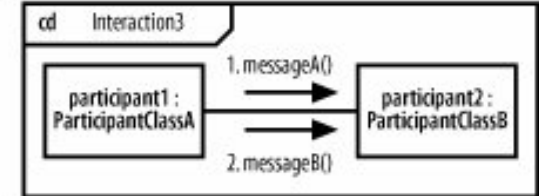
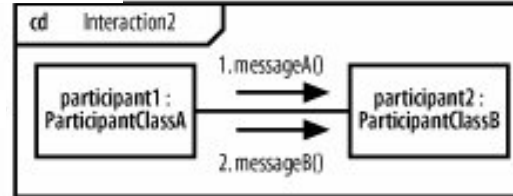
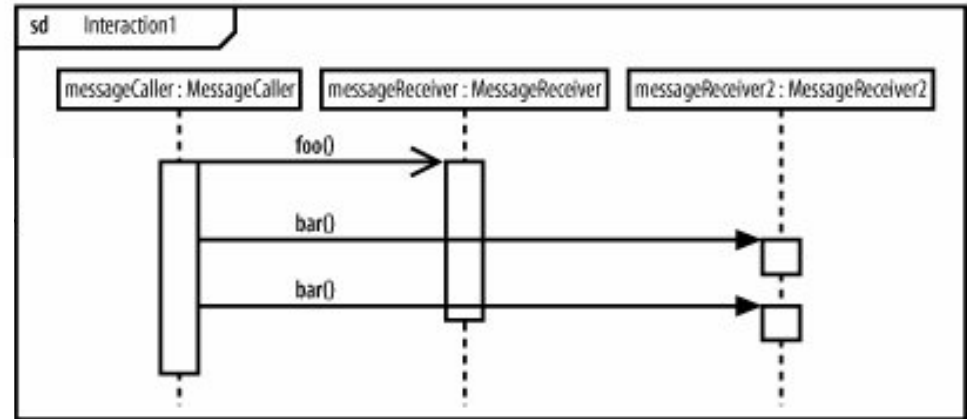
- Pode-se imaginar um diagrama de interação – visão geral como um diagrama de atividades onde cada ação é, na verdade, uma interação.
- Ou seja, dentro de cada “ação” do diagrama visão geral pode-se ter um diagrama de seqüência, ou um diagrama de colaboração ou um diagrama de tempo.



Cada “ação” do diagrama encapsula uma interação

Exemplo

iod : InteractionOverview1



Exemplo

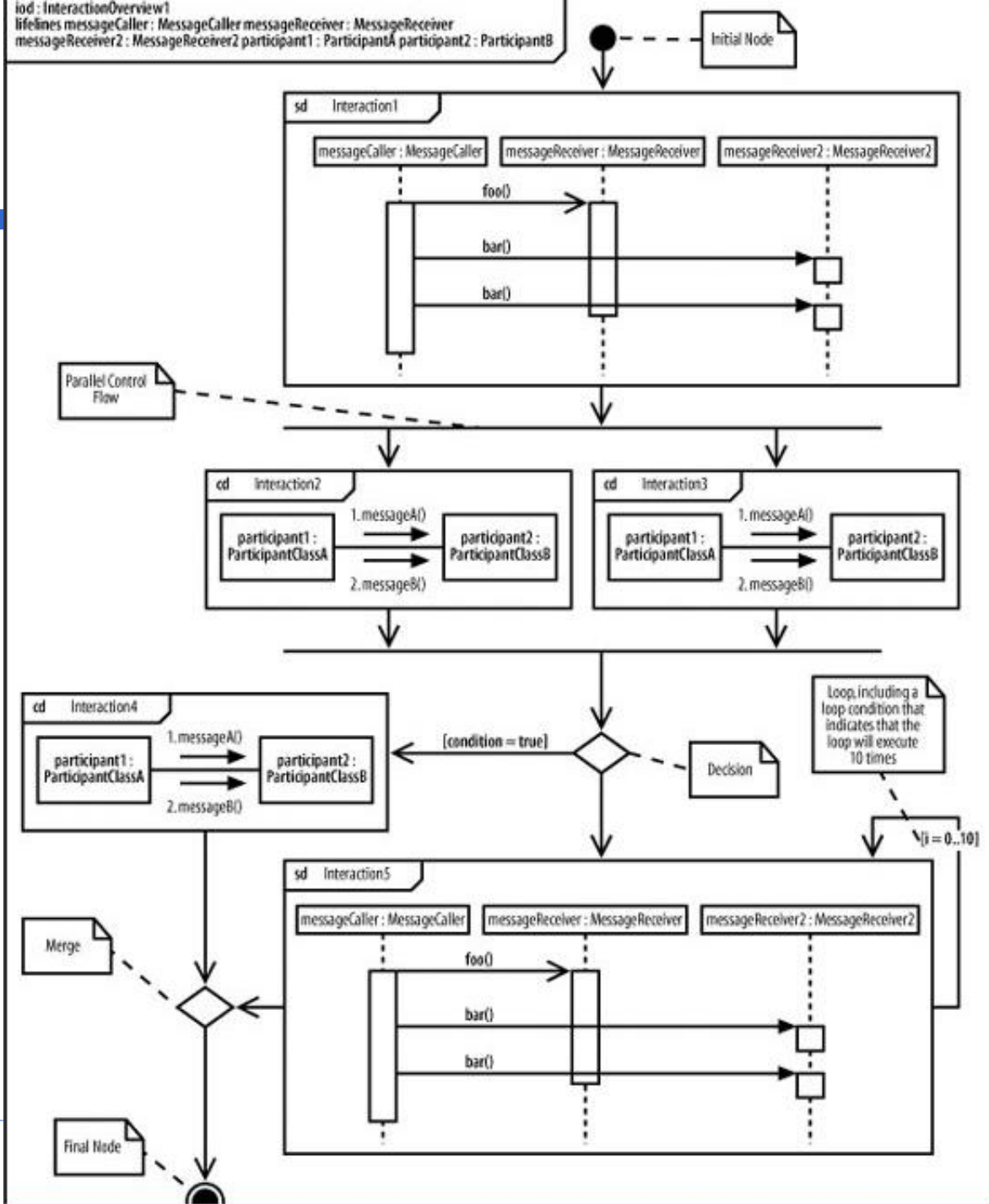


Diagrama de Estruturas Compostas

Diagrama de Estrutura Composta

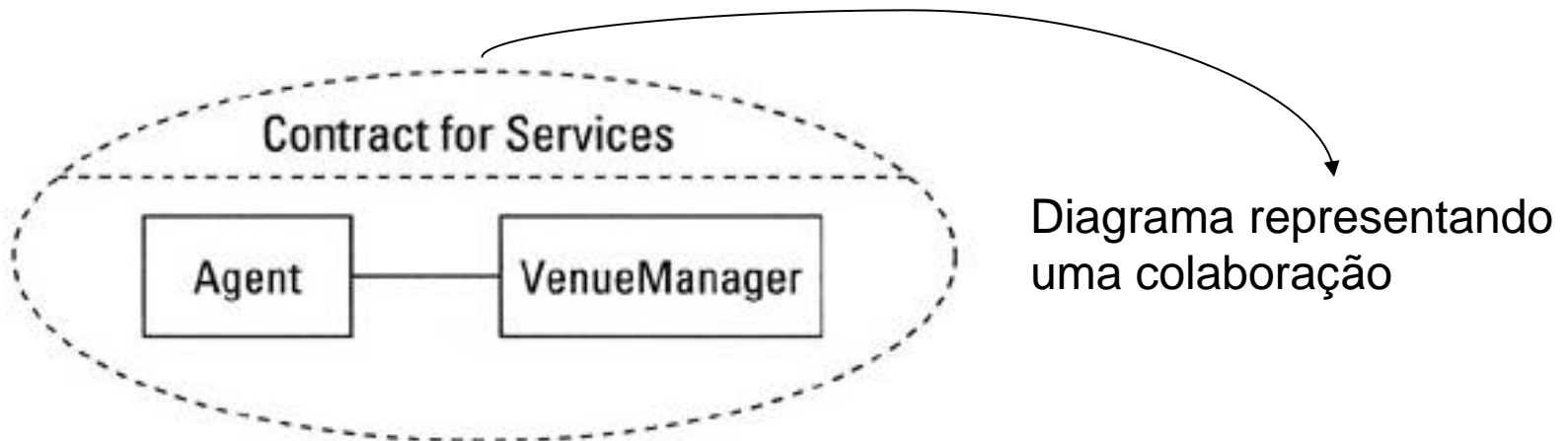
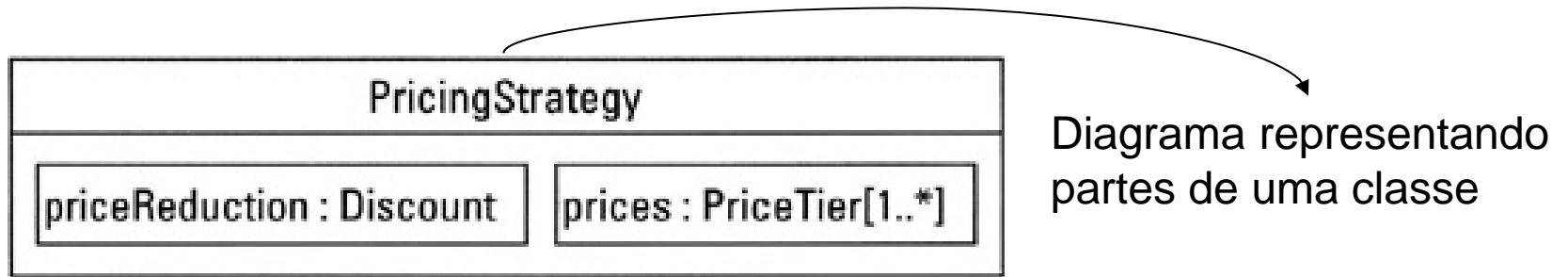
- Foi criado para representar visualmente partes de classes, componentes ou colaborações, incluindo os pontos de interação, usados para acessar as *features* do sistema.
- São usados para modelar situações específicas, como situações onde é necessário mostrar*:
 - A **estrutura interna** das classes.
 - Como classes podem ser acessadas a partir de **portas**.
 - Como objetos **colaboram** para atingir um determinado objetivo.

* Livro “Learning UML 2.0”.

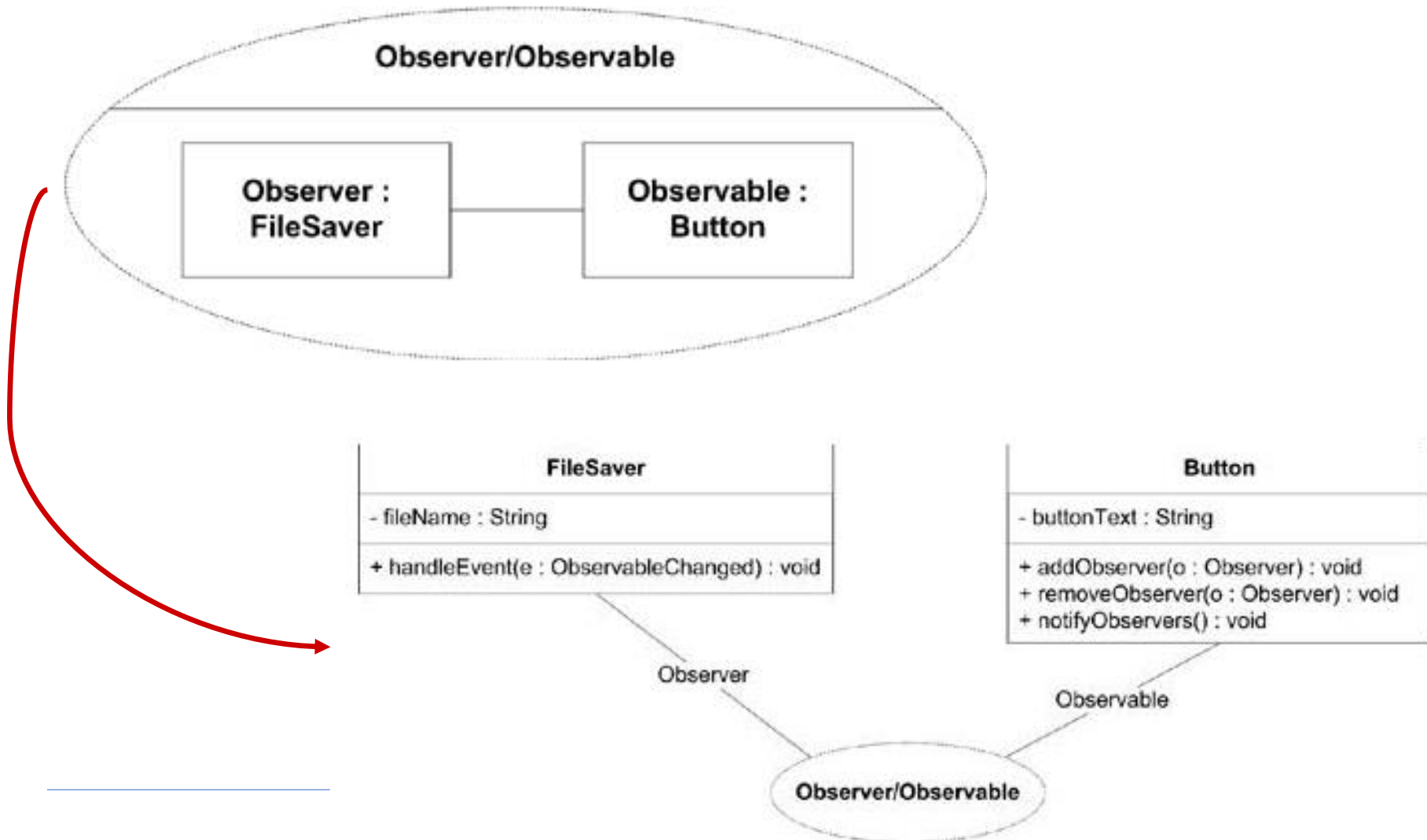
Diagrama de Estrutura Composta

- Benefícios:
 - Permitem mostrar o projeto de componentes complexos.
 - Permitem mostrar a interface de um componente de forma separada de sua estrutura.
 - Permitem descrever os papéis que diferentes elementos exercem para atingir os propósitos de uma determinada estrutura e satisfazer as interações requeridas.
-

Exemplos



Exemplos



***Overview* dos diagramas da UML 2.0**

Características dos Diagramas

- **Casos de Uso:** organiza e modela o comportamento do sistema. Mostra as interações entre o sistema que está sendo construído e os usuários ou outros sistemas.
 - **Atividades:** enfatiza o fluxo de atividades tanto do ator quanto do sistema (atividades seqüenciais e paralelas dentro do sistema).
 - **Classes:** mostra a estrutura de classes, seus relacionamentos, atributos e métodos.
 - **Objetos:** mostra um conjunto de objetos e seus relacionamentos. Mostra instâncias dos objetos definidos com base no diagrama de classes.
-

Características dos Diagramas

- **Sequência:** enfatiza o ordenamento das mensagens trocadas entre os objetos (interações entre objetos onde a ordem da interação é importante).
 - **Comunicação:** enfatiza a organização estrutural dos objetos que trocam mensagens. As maneiras que os objetos interagem e as conexões que são necessárias para suportar esta interação.
 - **Estados:** enfatiza o comportamento de um objeto de acordo com um conjunto de eventos. Mostra como que estímulos externos causam mudanças no objeto ao longo de seu tempo de vida.
 - **Interação geral:** utilizado para capturar uma interação importante no sistema. Coloca os diagramas de seqüência, comunicação e tempo juntos. Não mostram em detalhes as mensagens.
-

Características dos Diagramas

- **Componentes:** mostra um conjunto de componentes e seus relacionamentos.
 - **Implantação:** mostra como o sistema é implantado em uma situação do mundo real.
 - **Tempo:** descreve as mudanças no estado ou condição de um objeto de uma classe durante um tempo.
 - **Estrutura composta:** modela um conjunto de instâncias que cooperam entre si para executar uma função específica (classes, componentes, interações).
 - **Pacotes:** mostra a organização hierárquica de grupos de elementos. Podem ser utilizados para organizar quaisquer elementos UML.
-

Síntese dos diagramas UML 2.0

- Estruturais
 - Mostram as características do sistema que não mudam com o tempo.
 - **Diagramas:** Classe, Objeto, Pacote, Estrutura composta, Componentes e Implantação.
- Comportamentais
 - Mostram como o sistema responde as requisições ou vai se modificando ao longo do tempo.
 - **Diagramas:** Máquina de Estados, Caso de Uso e Atividade.
- De Interação
 - São também diagramas comportamentais. Mostram a troca de mensagens em uma colaboração (um grupo de objetos cooperantes), para atingir um objetivo.
 - **Diagramas:** Seqüência, Comunicação, Interação Geral e Tempo.

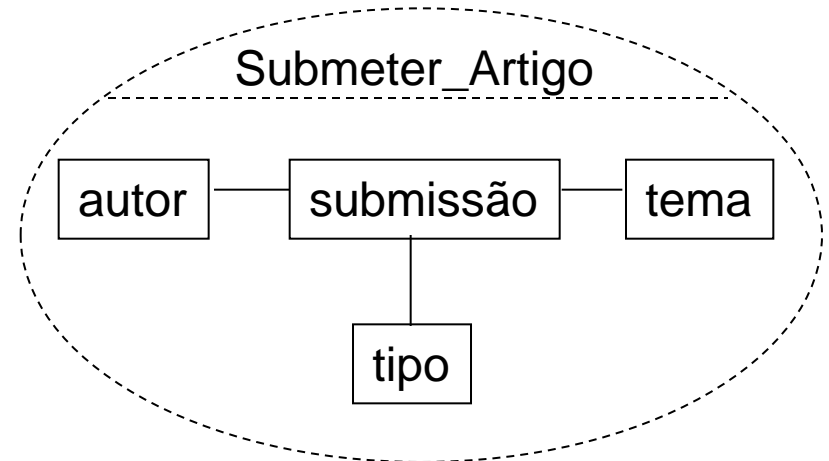
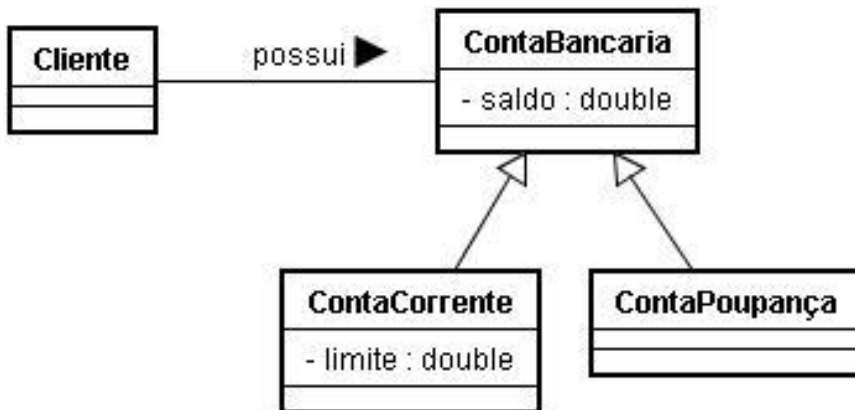
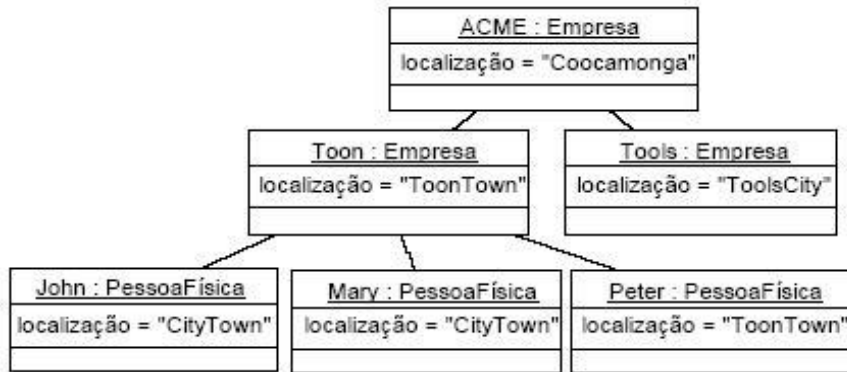
Modelando com UML

- Principais perguntas em um sistema e como respondê-las usando diagramas UML:
 - **Quem usa o sistema?**
 - Mostre os **atores** em seus **diagramas de casos de uso**.
 - **O sistema é constituído de quê?**
 - Desenhe **diagramas de classes** para mostrar a estrutura lógica do sistema e **diagramas de componentes** para mostrar a estrutura física.
 - **Em que lugar estão situados os componentes do sistema?**
 - Indique seus planos de onde os componentes irão estar localizados e “rodar” nos **diagramas de implantação**.

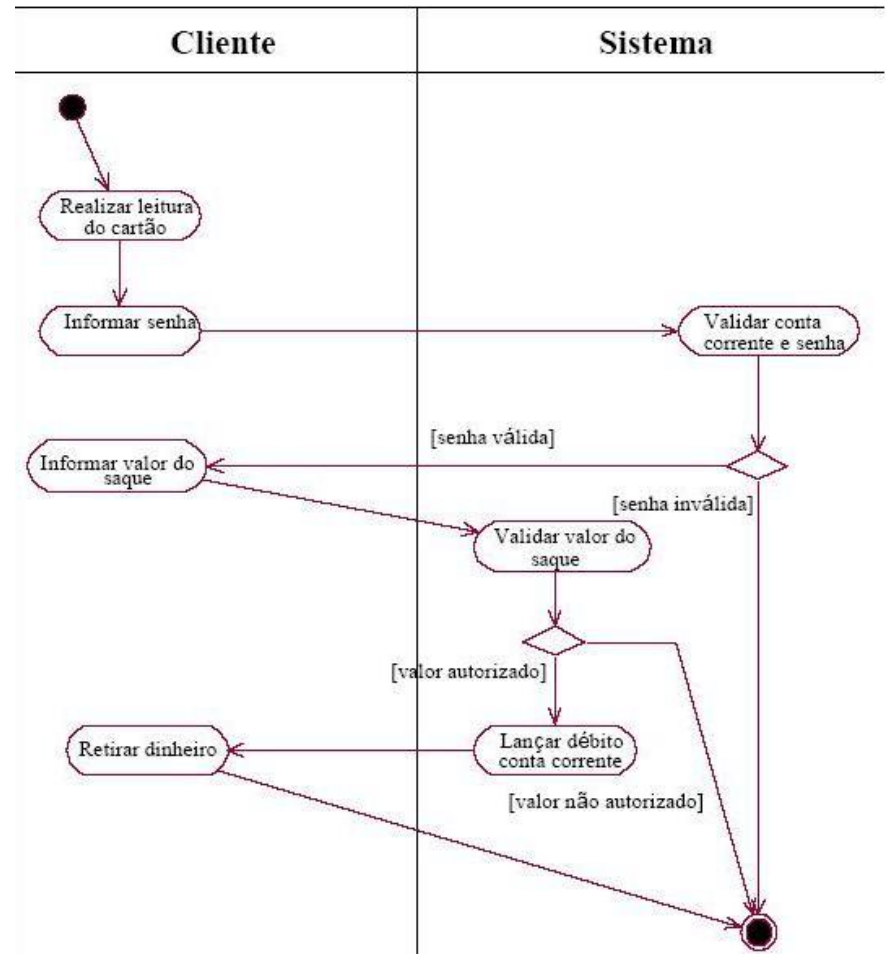
Modelando com UML [cont]

- Principais perguntas em um sistema e como respondê-las usando diagramas UML:
 - **Quais e quando importante eventos ocorrem no sistema?**
 - Mostre quais os eventos que os objetos reagem com **diagramas de estados e de interação**.
 - **Por que o sistema está fazendo as coisas que ele está fazendo?**
 - Identifique os objetivos dos usuários do sistema e capture eles dentro de **casos de uso**.
 - **Como o sistema começa a trabalhar?**
 - Mostre as partes do sistema em **diagramas de estruturas compostas** e use **diagramas de comunicação** para mostrar as interações no nível de abstração necessário para projeto e implementação.

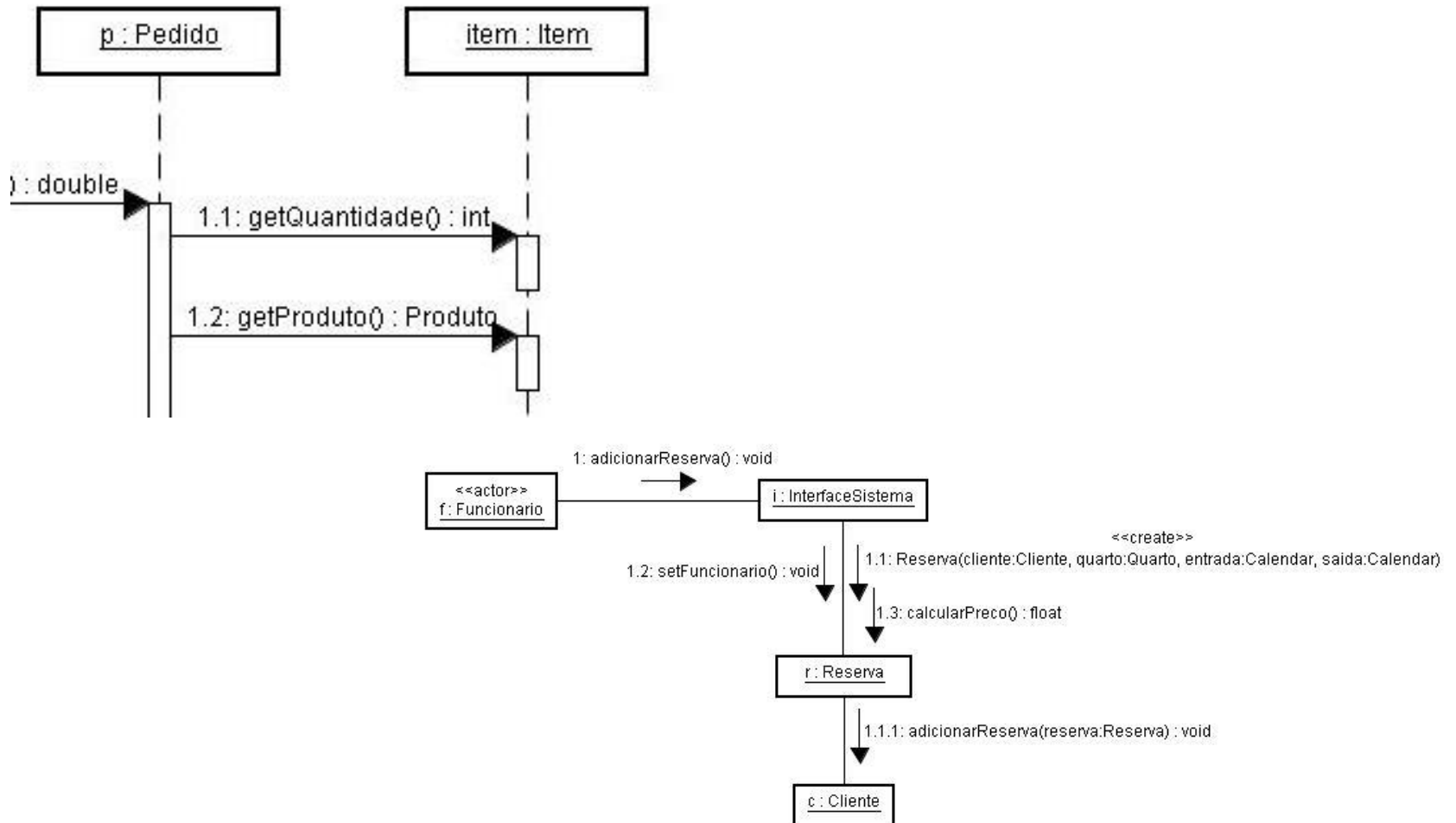
Revisão rápida...



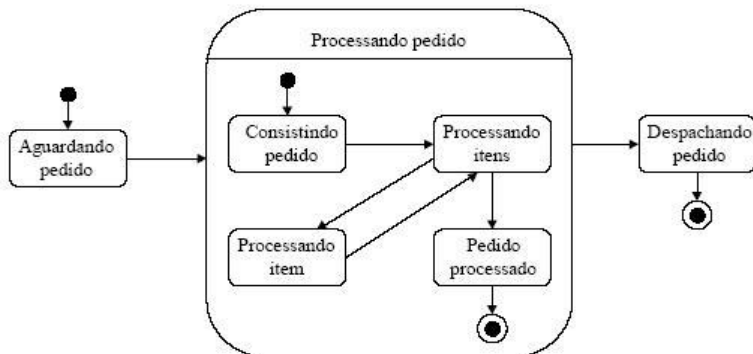
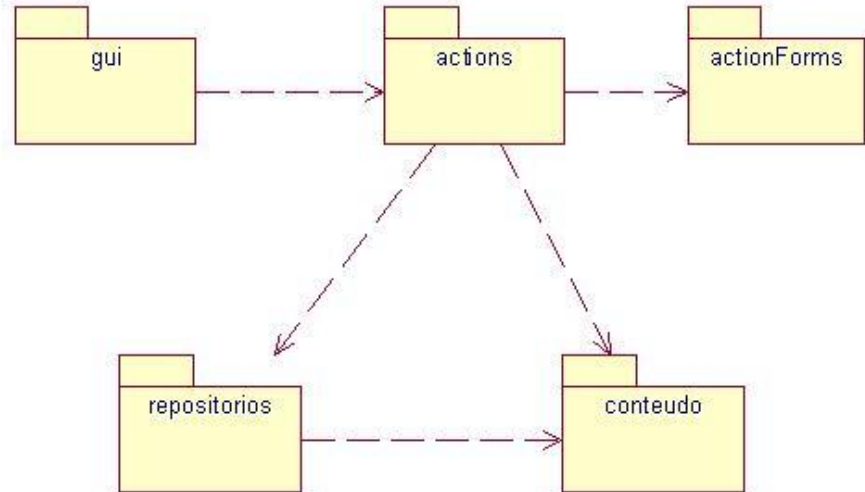
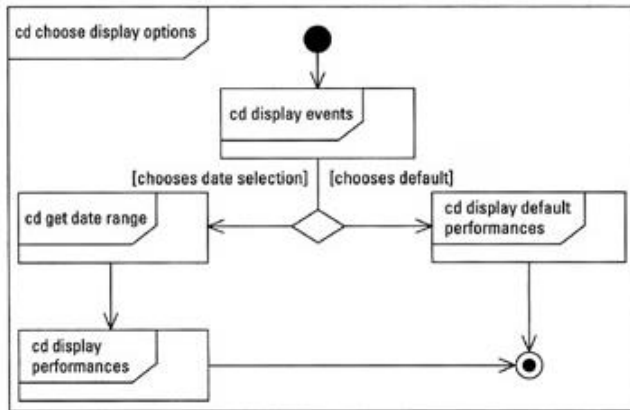
Revisão rápida...



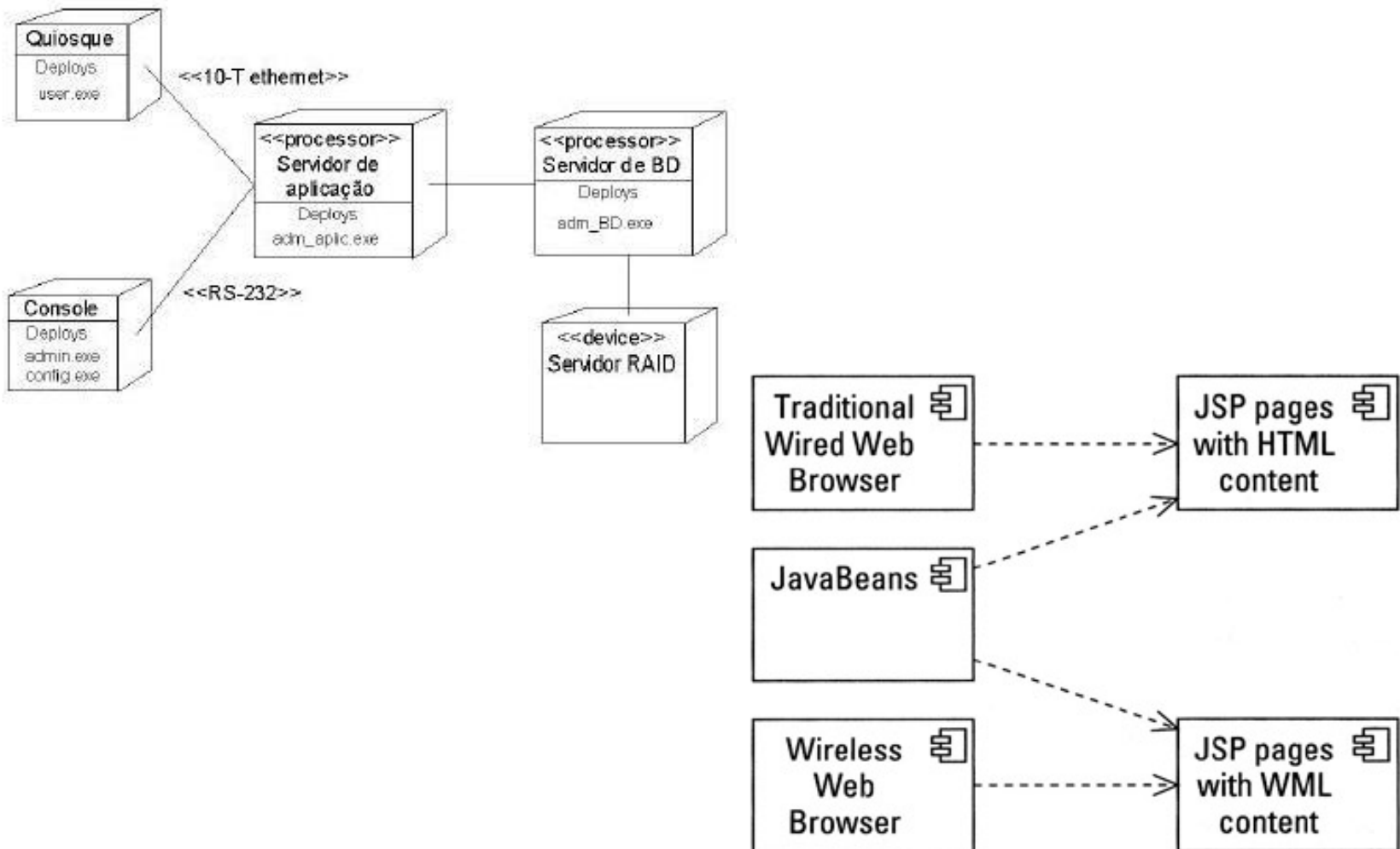
Revisão rápida...



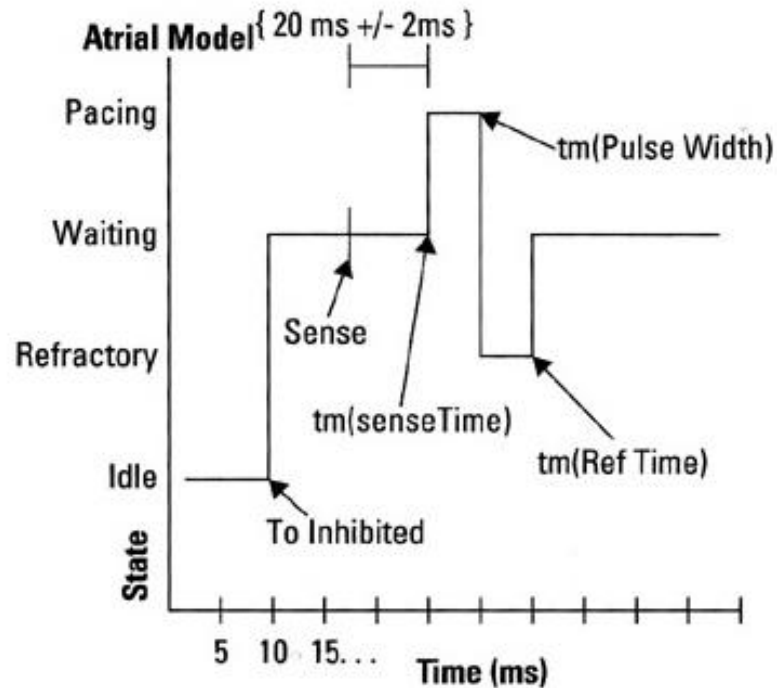
Revisão rápida...



Revisão rápida...



Revisão rápida...



Tarefa 1

- Liste os diagramas da UML que podem ser usados para auxiliar no processo de desenvolvimento de software:
 - Na fase de levantamento de requisitos.
 - Na fase de projeto.
 - Para documentação do sistema.
-

Tarefa 1 – Possível resposta

Fase de levantamento de requisitos

- A atividade de análise de requisitos tenta entender o que os usuários e clientes do software esperam que o sistema faça. Muitas técnicas da UML podem ser usadas para ajudar nesta atividade, como:
 - **Casos de uso**, que descrevem como as pessoas interagem com o sistema.
 - **Diagramas de atividades**, que podem ajudar a entender o fluxo de trabalho dentro da organização.
 - **Diagrama de classes**, para construir o vocabulário do domínio.
 - **Diagrama de estados** (se um conceito tem um ciclo de vida complexo, com vários estados e vários eventos que podem acarretar mudanças de estado).
-

Tarefa 1 – Possível resposta

Fase de projeto do sistema

- Diagrama de classe (perspectiva de software).
 - Diagrama de sequência para os cenários mais interessantes dos casos de uso.
 - Diagrama de pacotes, para mostrar com que as classes do sistema ficarão organizadas.
 - Diagramas de estados para classes com “histórias de vida” complexas.
 - Diagramas de implantação para mostrar onde o sistema será implantado (layout físico).
-

Tarefa 1 – Possível resposta

Documentação

- Um diagrama de pacotes pode ser um bom mapa lógico do sistema.

Referências adicionais

- Booch, J.; Rumbaugh, J.; and Jacobson, I. **“The Unified Modeling Language User Guide”**, Addison Wesley, 1998, 512 p.
- Chonoles, M. and Schardt J. **“UML 2.0 for Dummies”**, 2003.
- Hamilton, Kim; Miles, Russell **“Learning UML 2.0”**, O'Reilly, 2006, 286 p.
- http://paginas.fe.up.pt/~jpf/teach/ES/Exercicios/exercicio_4.html.