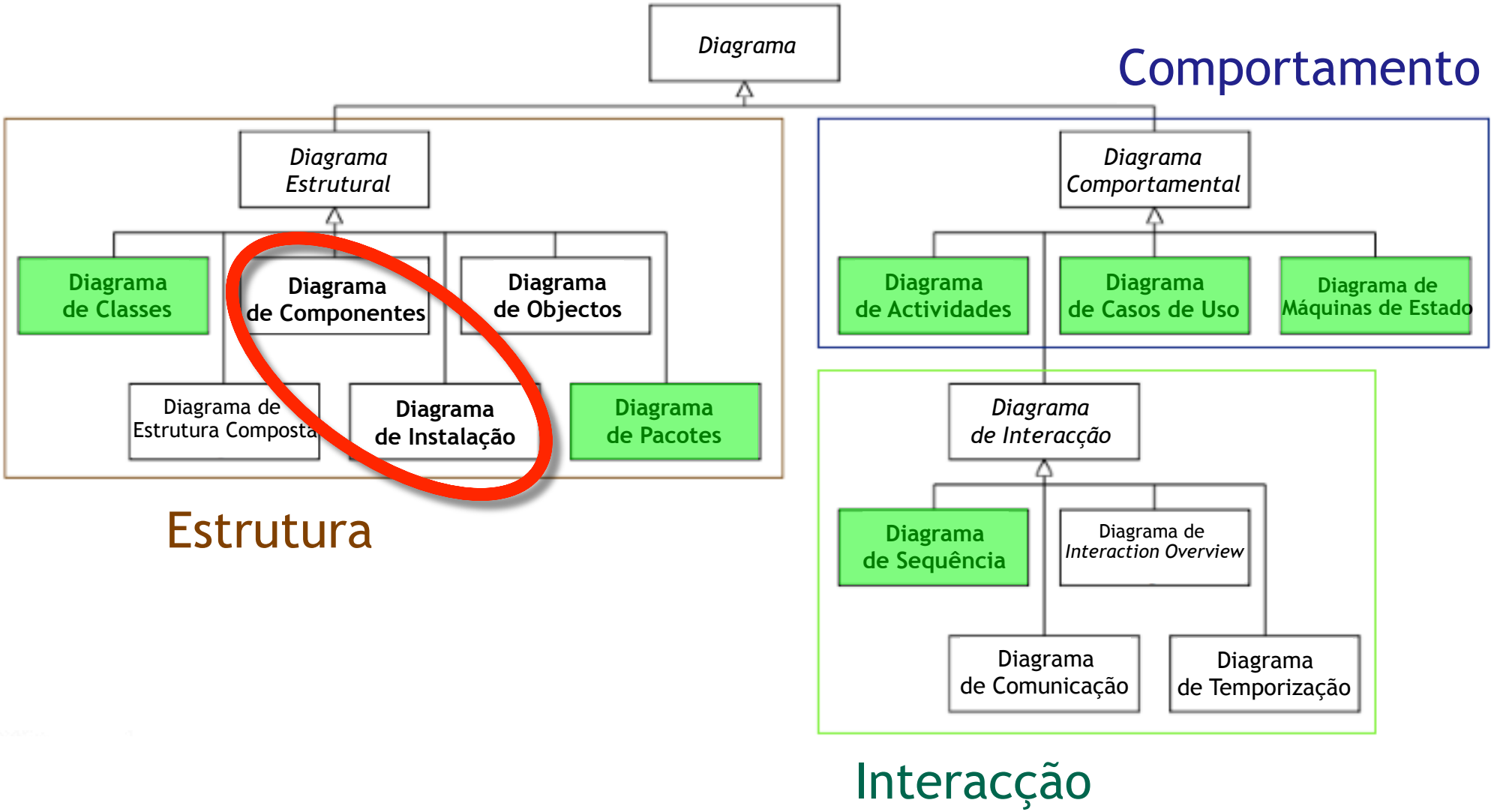
Desenvolvimento de Sistemas Software

Aula Teórica 21: Modelação Estrutural / Modelação Comportamental

Desenvolvimento de Sistemas Software
António Nestor Ribeiro, José Creissac Campos, F. Mário Martins

Diagramas da UML 2.x



Diagramas de Deployment

- Captura a topologia (ambiente) de hardware de um sistema sobre a qual são executados os componentes de software

- Construído como parte da especificação da arquitectura física

- Objectivo:
  - Especificar a distribuição de componentes
  - Identificar estrangulamentos de desempenho

- Permitem que a equipa de engenheiros especifique a disposição física dos elementos que constituem o sistema
  - Acrescenta detalhe que tem a ver com a configuração do sistema em tempo de execução
  - Permite cruzar competências de engenharia de software com redes de comunicações, sistemas operativos e bases de dados

Diagramas de Deployment

- Elementos de um diagrama de deployment
  - Nós
  - Ligações

- **Nós (*nodes*):**

  - Computadores ou outros dispositivos
  - Existem nós que são nós de hardware (server, desktop, disk drives) ou nós de ambiente de execução (sistema operativo, web server, application server, etc.)
  - Os componentes localizados (deployed) em cada nó são representados explicitamente
  - É possível agrupar nós em pacotes (packages)

- **Ligações (*connections*):**
  - Entre nós (podem ser decoradas com multiplicidades)
  - Podem ter estereótipos que indicam o tipo de ligação. Exemplo: «TCP/IP» ou «RMI»
  - Representam comunicação entre os nós.

Diagramas de Deployment

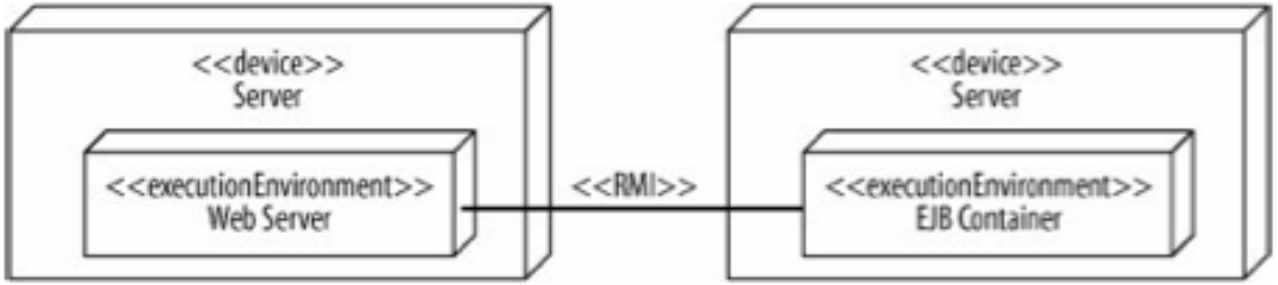
- Por vezes utiliza-se o estereótipo <<device>> para identificar os nós de hardware

- Para identificar os ambientes de execução utiliza-se o estereótipo <<executionEnvironment>>

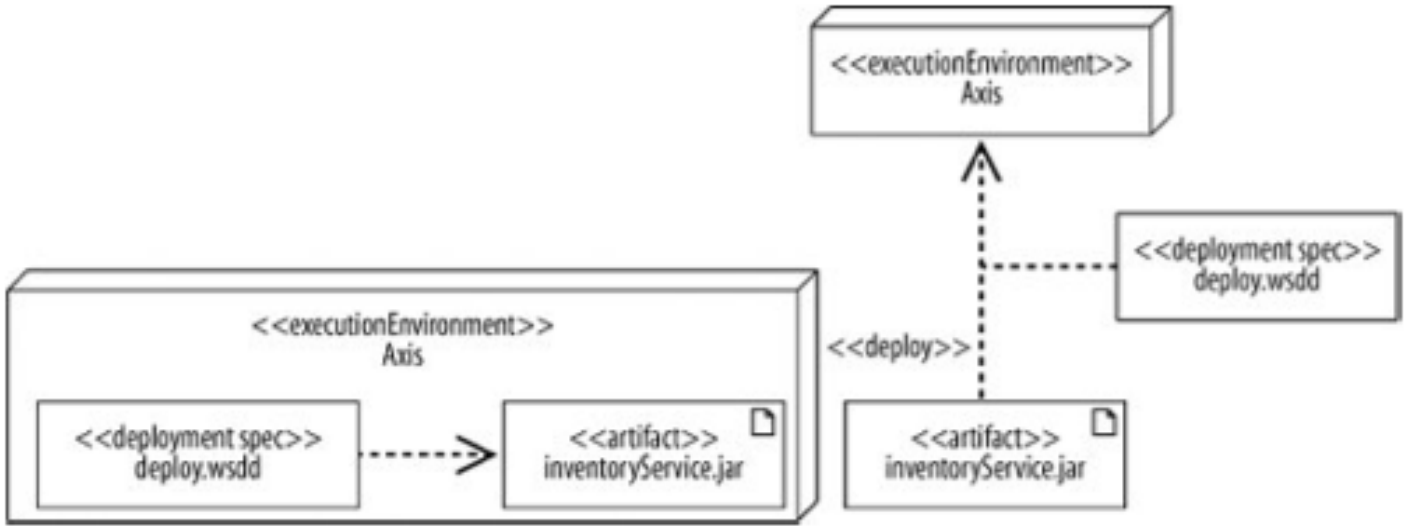
- Comunicação entre dois nós

Diagramas de Deployment

- A descrição pode ser refinada para detalhar os ambientes de execução em cada nó

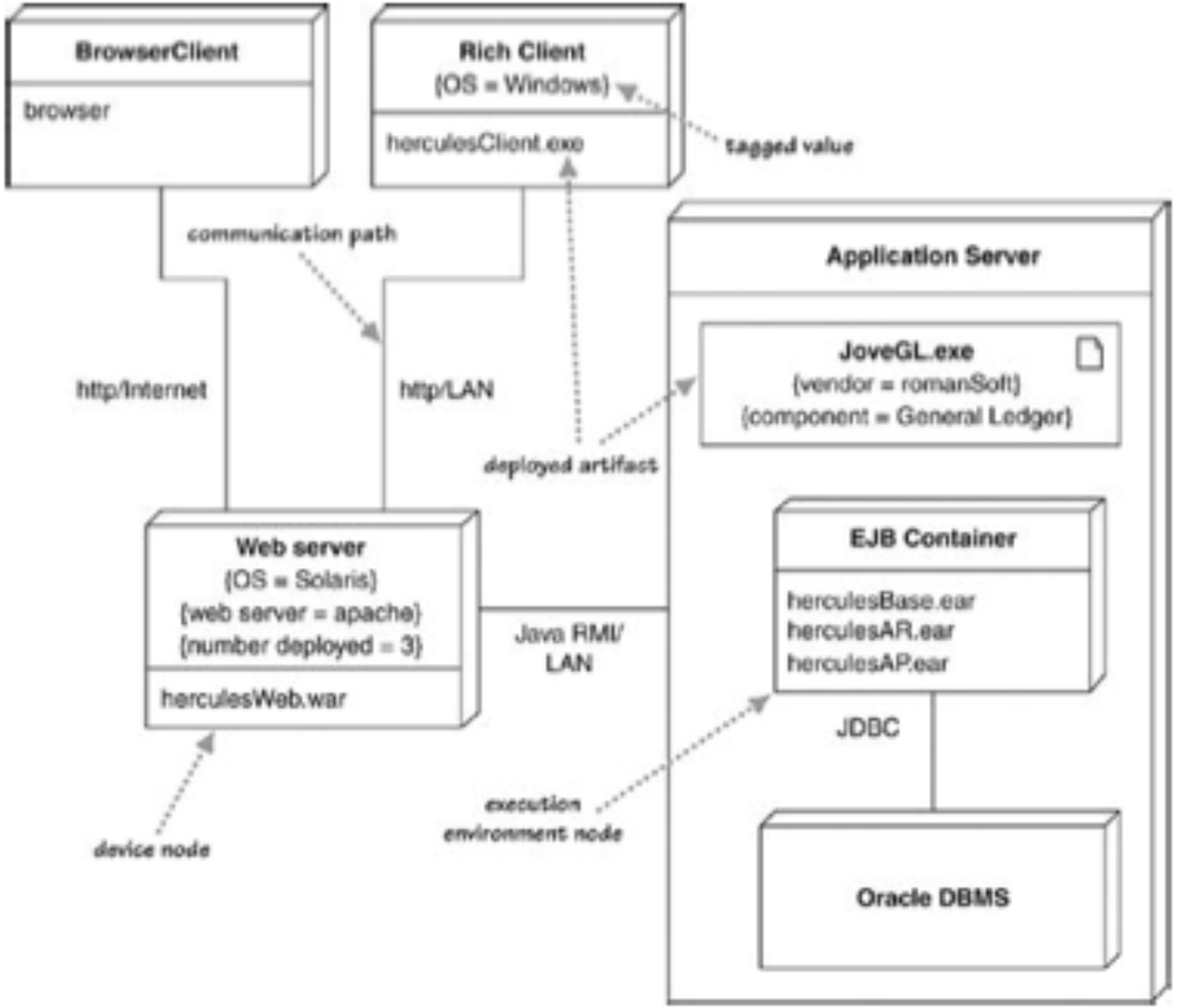


- Especificação de dependências em tempo de execução



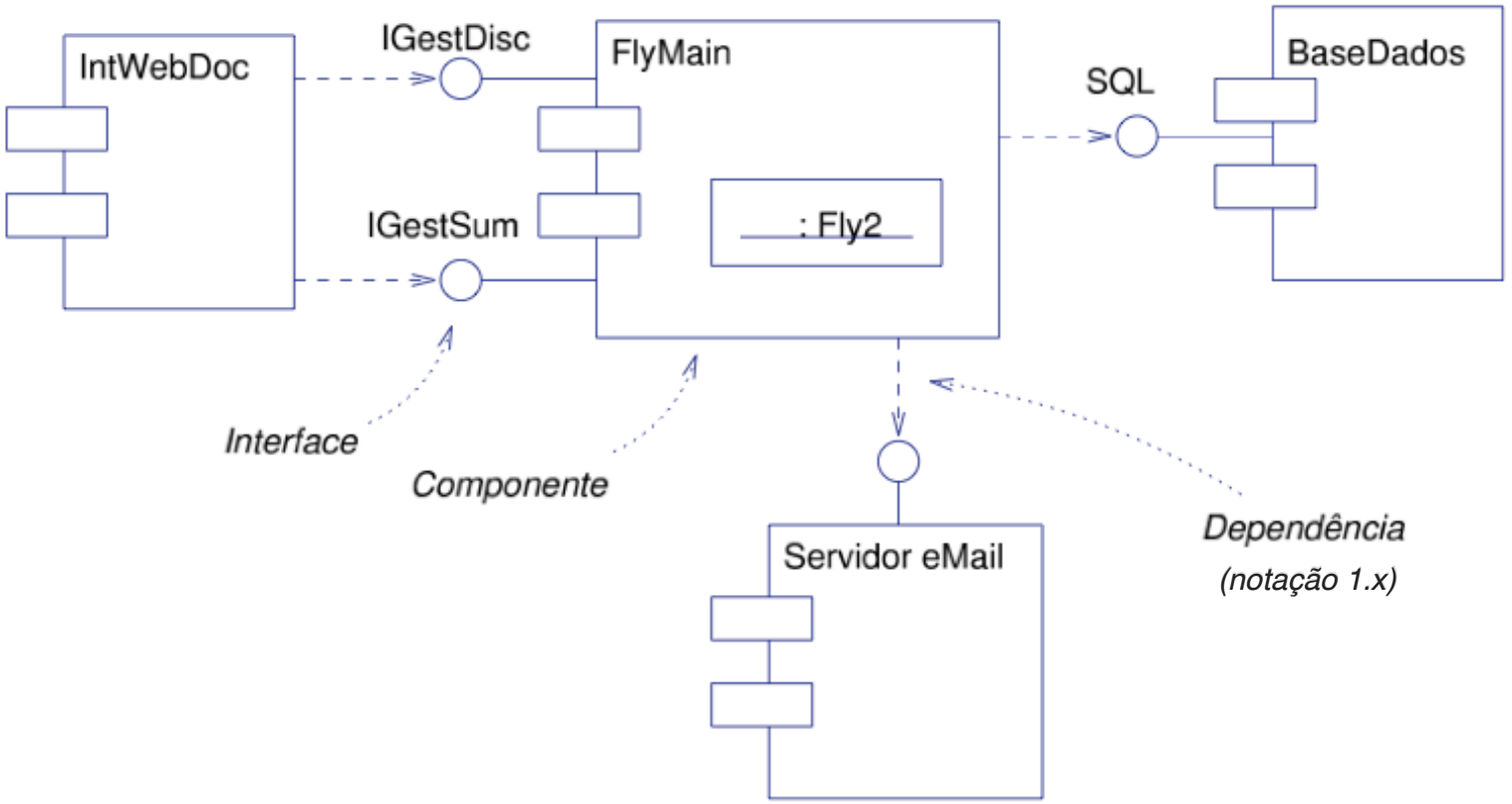
Diagramas de Deployment

- Um diagrama mais completo



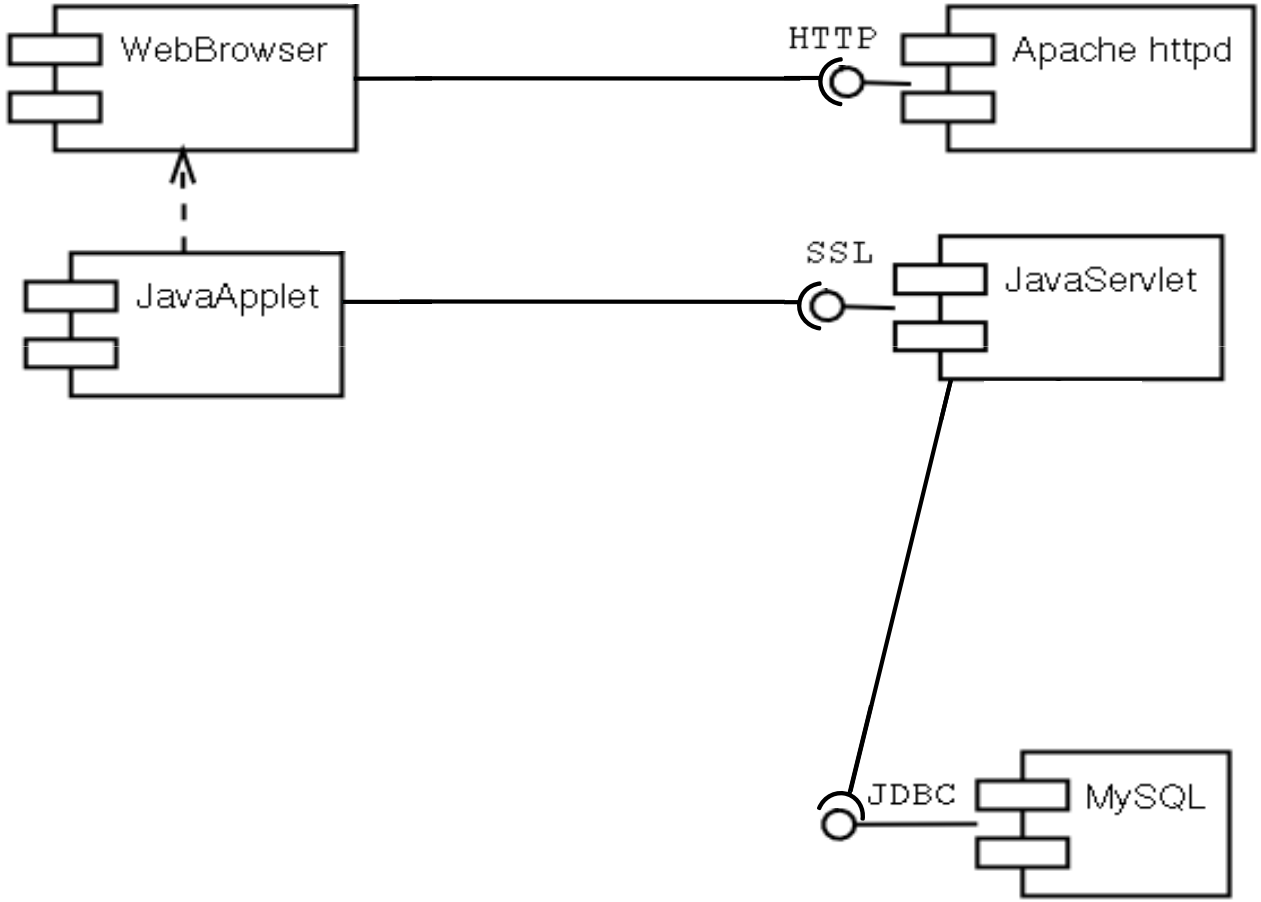
Diagramas de Componentes

- Um diagrama de componentes mostra as depêndencias entre os componentes software do sistema.
- Pode ser desenhado a diferentes níveis: código fonte, componentes binários (dlls), componentes executáveis.
- Permitem identificar, a cada nível, o que é necessário para construir o sistema.

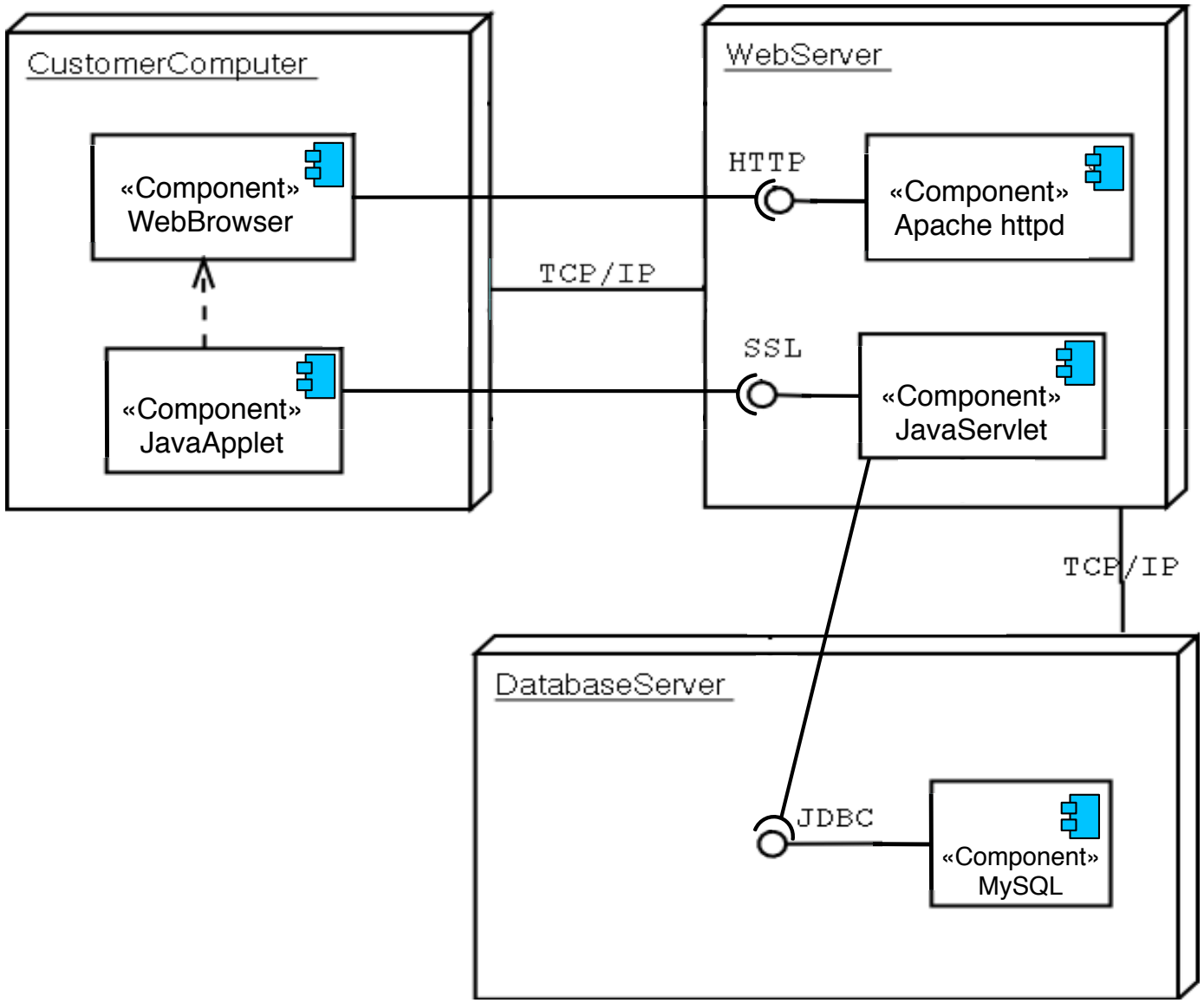


Deployment e Componentes

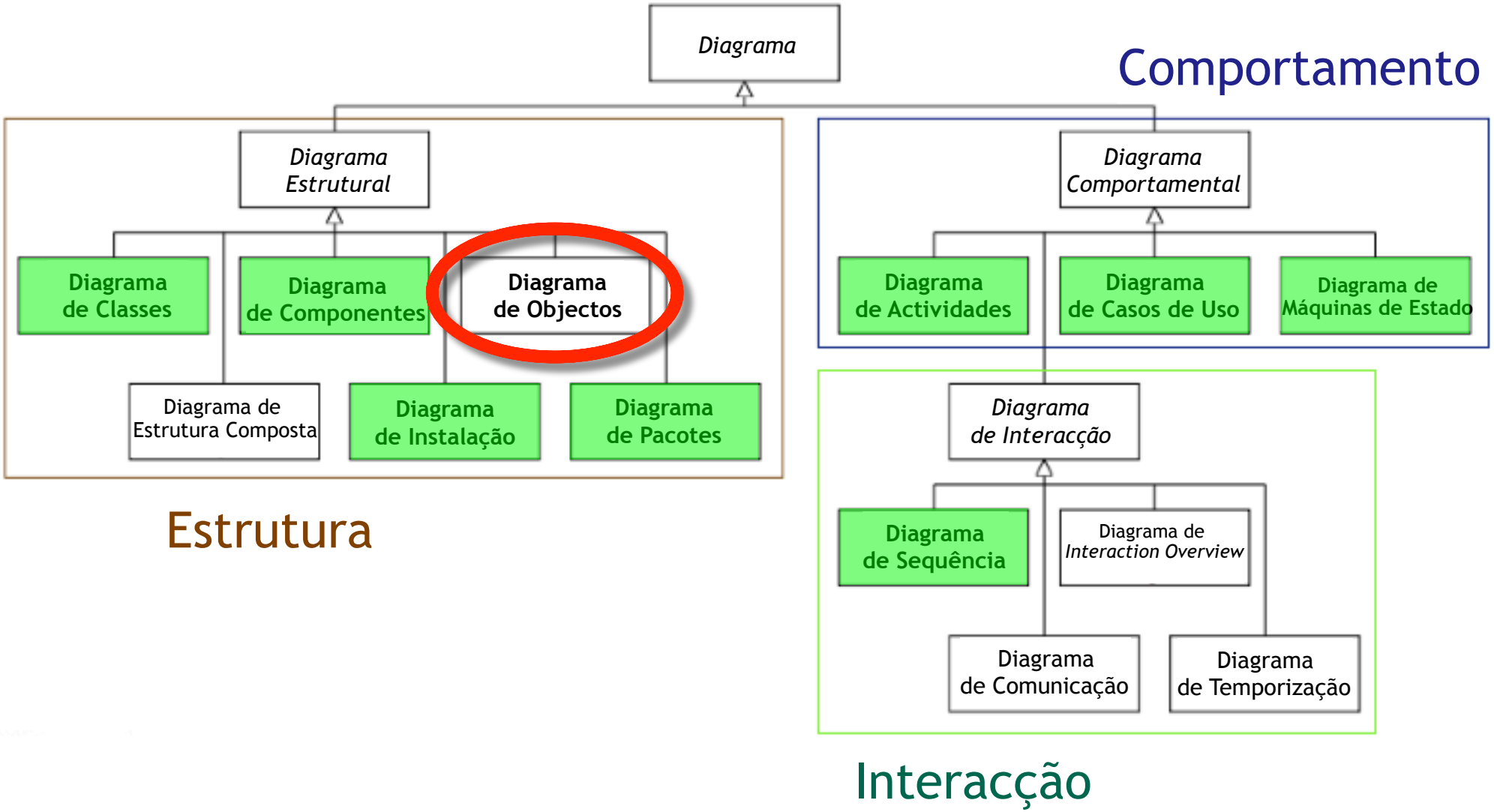
- **Diagrama de Deployment representa a configuração física do software e do hardware**
- **Componentes são módulos de código; Nodos são hardware (exº servidores);**



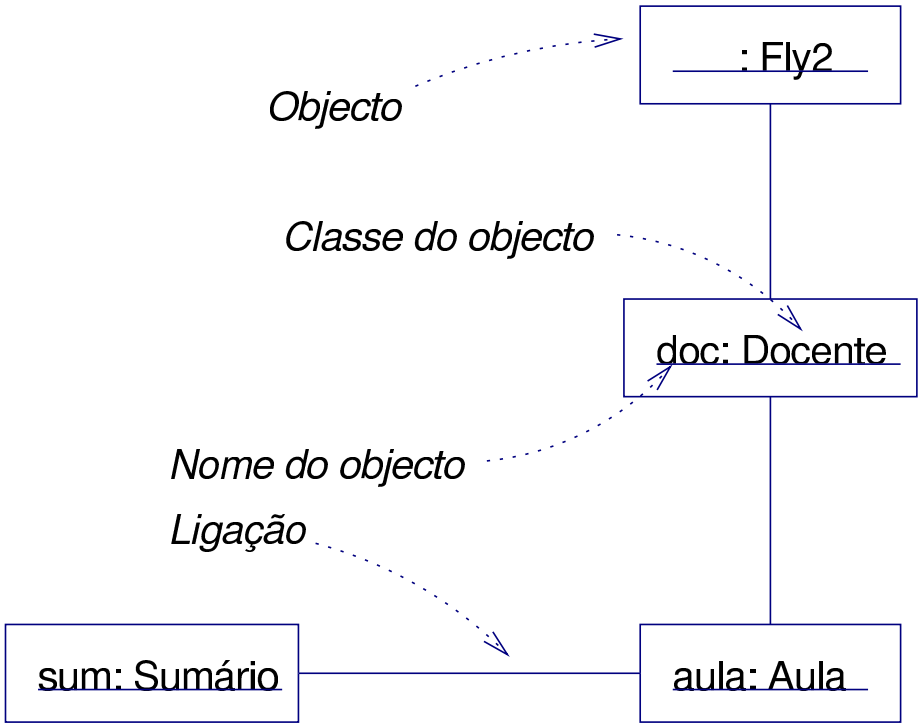
Componentes - notação 2.0



Diagramas da UML 2.x

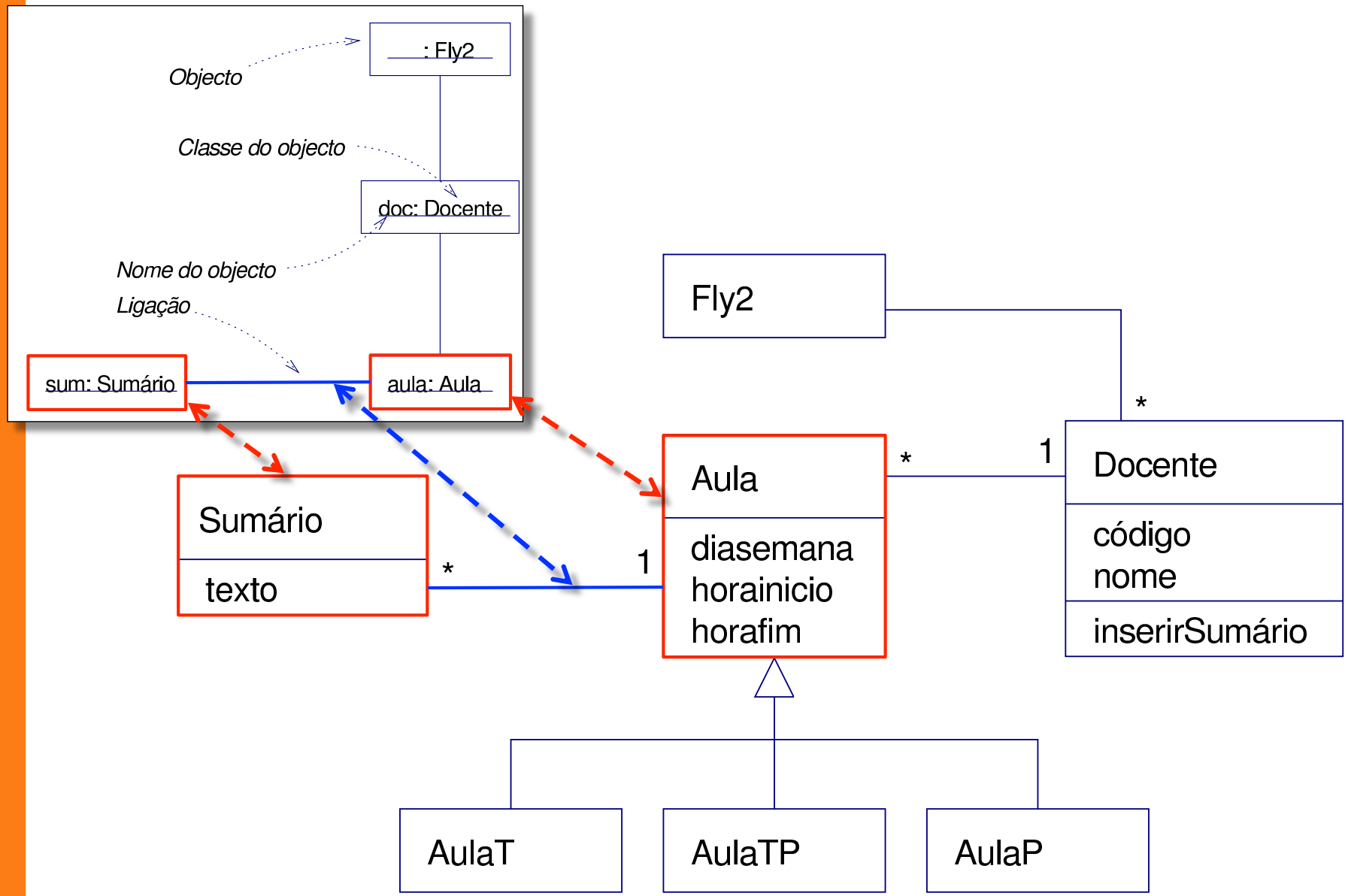


Diagramas de Objectos

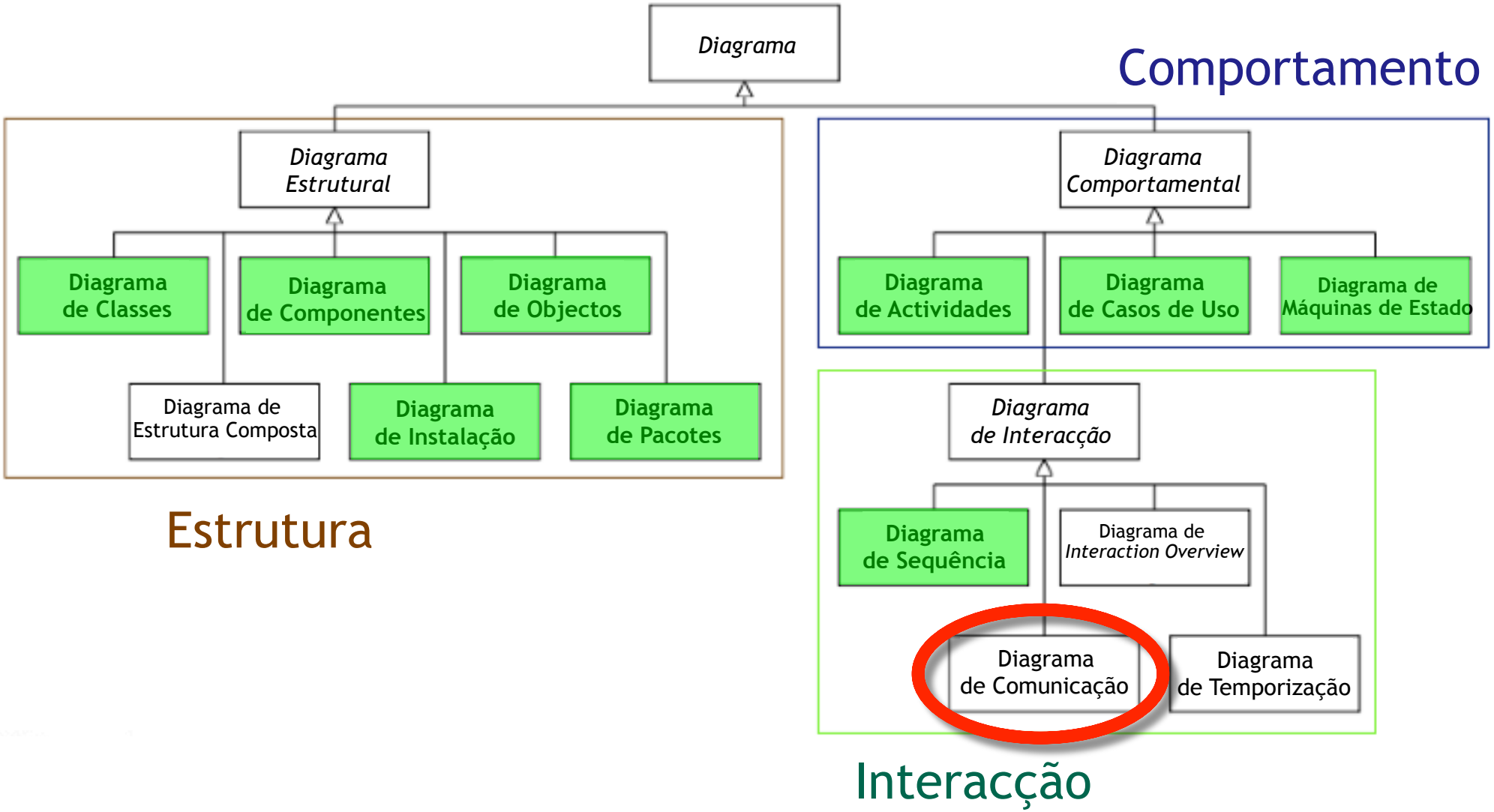


- Apresentam uma configuração particular de objectos no sistema.
- Modelam a visão estática do sistema, do ponto de vistas de instâncias reais:
  - objectos são instâncias das classes do modelo;
  - ligações são instâncias das associações entre as classes.
- Útil, por exemplo, durante *debugging*.

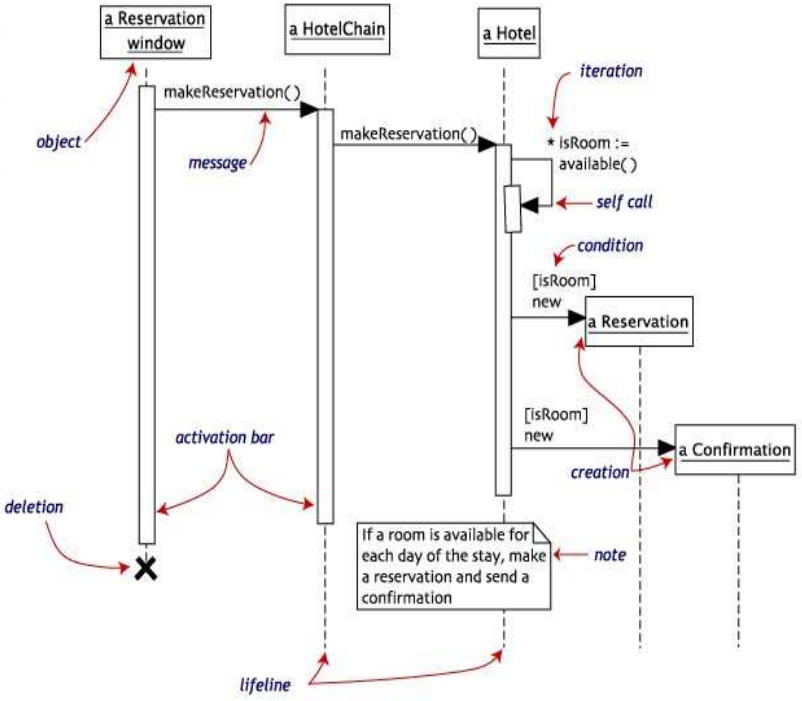
- Objetos e ligações no Diagrama de Objetos são instâncias de classes e associações no correspondente Diagrama de Classes.

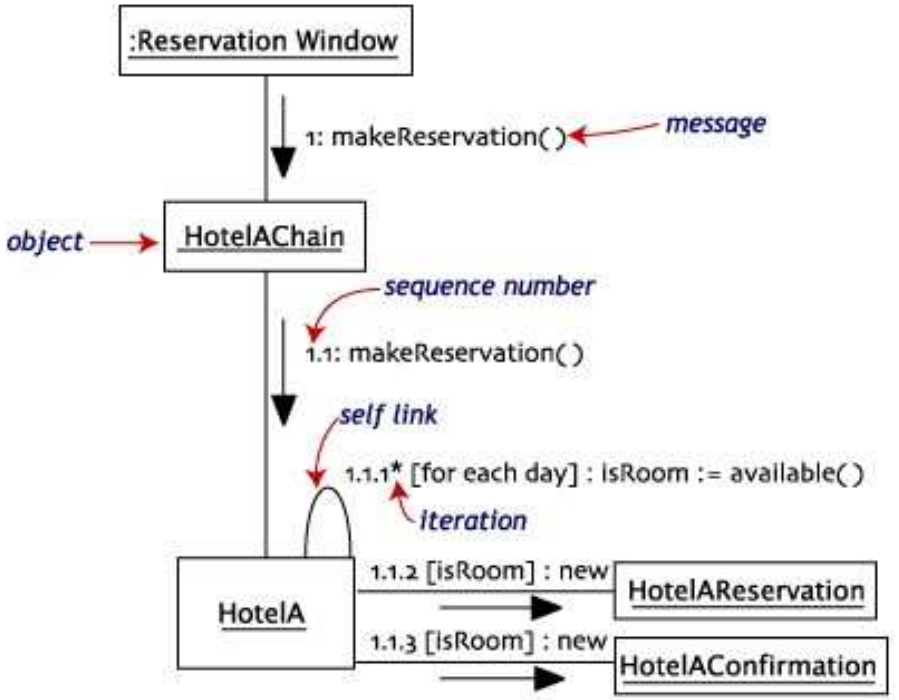


Diagramas da UML 2.x



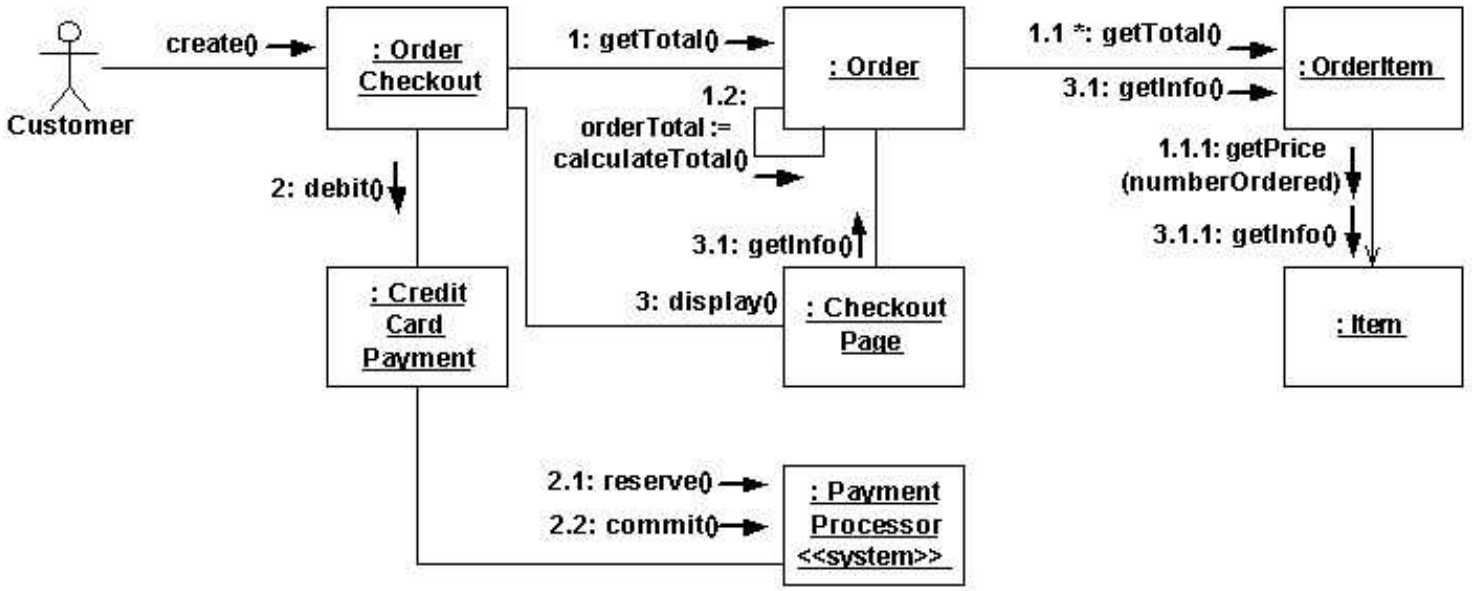
Diagramas de Comunicação





- São **Diagramas de Interacção** tais como os DS mas centram-se no papel dos objectos em vez da sequência temporal das mensagens;
- As mensagens são numeradas indicando a sua sequência;

Diagramas de Comunicação



- Representam uma outra perspectiva (visão) sobre a colaboração entre classes, respectiva troca de mensagens e sua sequência.

Modelação Estrutural/Comportamental

Sumário

- Modelação Estrutural com Diagramas de Instalação e de Componentes
- Modelação Estrutural com Diagramas de Objetos
- Modelação Comportamental com Diagramas de Comunicação
  - Dos Diagramas de Objetos aos Diagramas de Comunicação
  - Diagramas de Comunicação vs. Diagramas de Sequência