

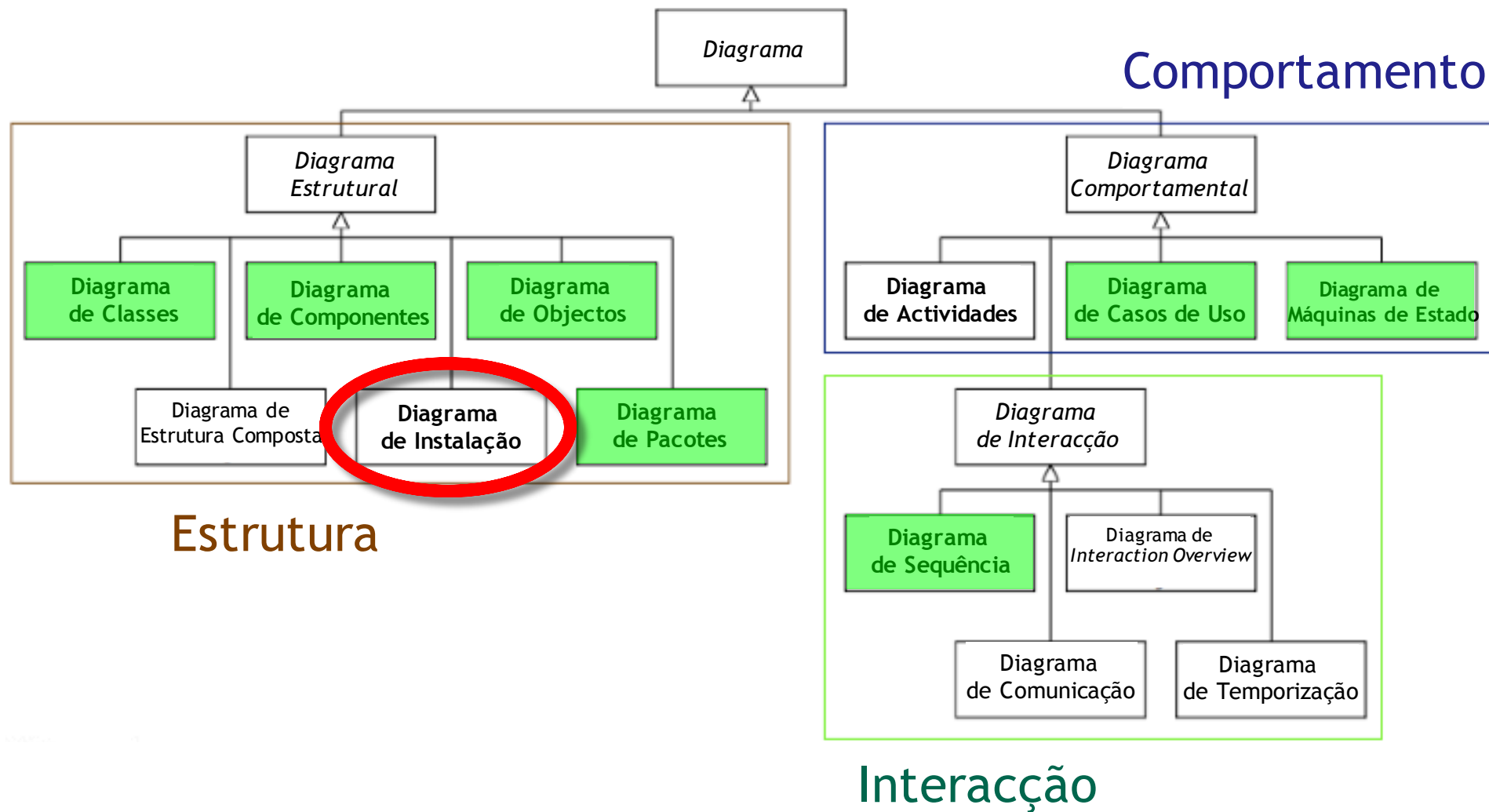


Desenvolvimento de Sistemas Software

Aula Teórica 21/22: Modelação Estrutural



Diagramas da UML 2.x





Diagramas de Instalação (*Deployment*)

- Qual a disposição física dos componentes que constituem o sistema?
 - Qual é a configuração do sistema em tempo de execução?
- Diagramas de Instalação especificam a arquitectura física do sistema
 - Topologia (ambiente) de hardware sobre a qual são executados os componentes de software
- Permitem:
 - Especificar a distribuição de componentes
 - Identificar estrangulamentos de desempenho



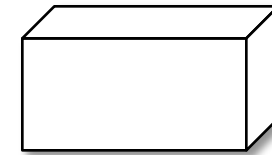
Diagramas de Instalação

- Elementos de um diagrama de deployment

- Nós
- Ligações

- **Nós (*nodes*):**

- Computadores ou outros dispositivos (hardware)
- Sistema operativo, *web servers*, *application servers*, etc. (ambientes de execução)
- Os componentes localizados (*deployed*) em cada nó são representados explicitamente
- É possível agrupar nós em pacotes (*packages*)



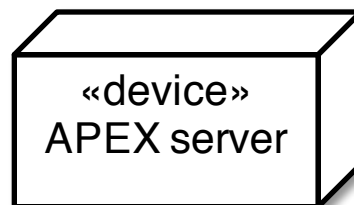
- **Ligações (*connections*):**

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

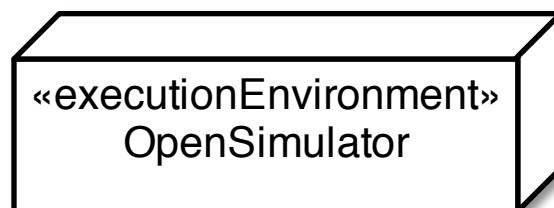


Diagramas de Instalação

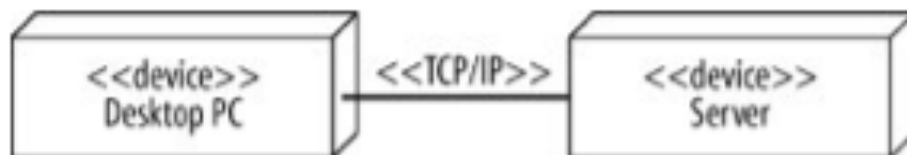
- Por vezes utiliza-se o estereótipo <<device>> para identificar os nós de hardware



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

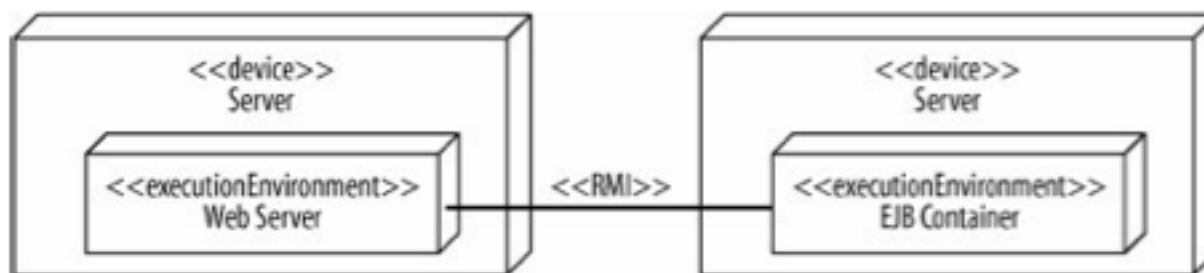


- Comunicação entre dois nós

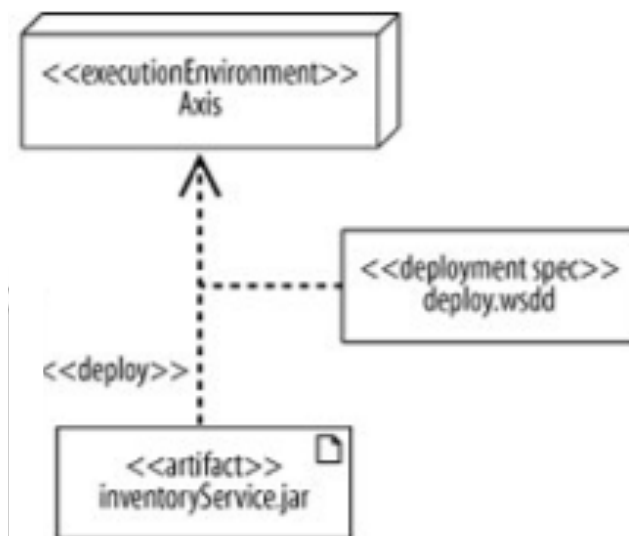
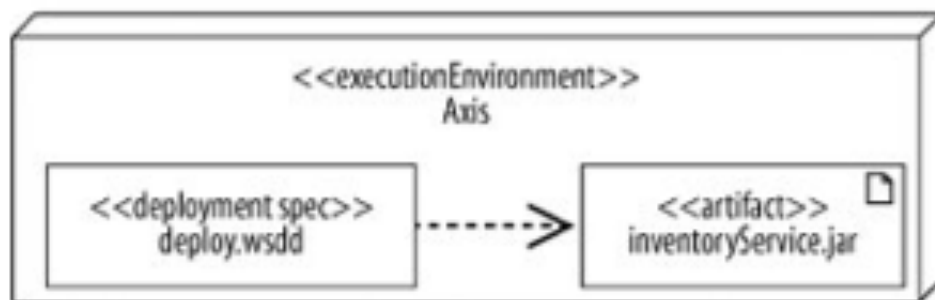


Diagramas de Instalação

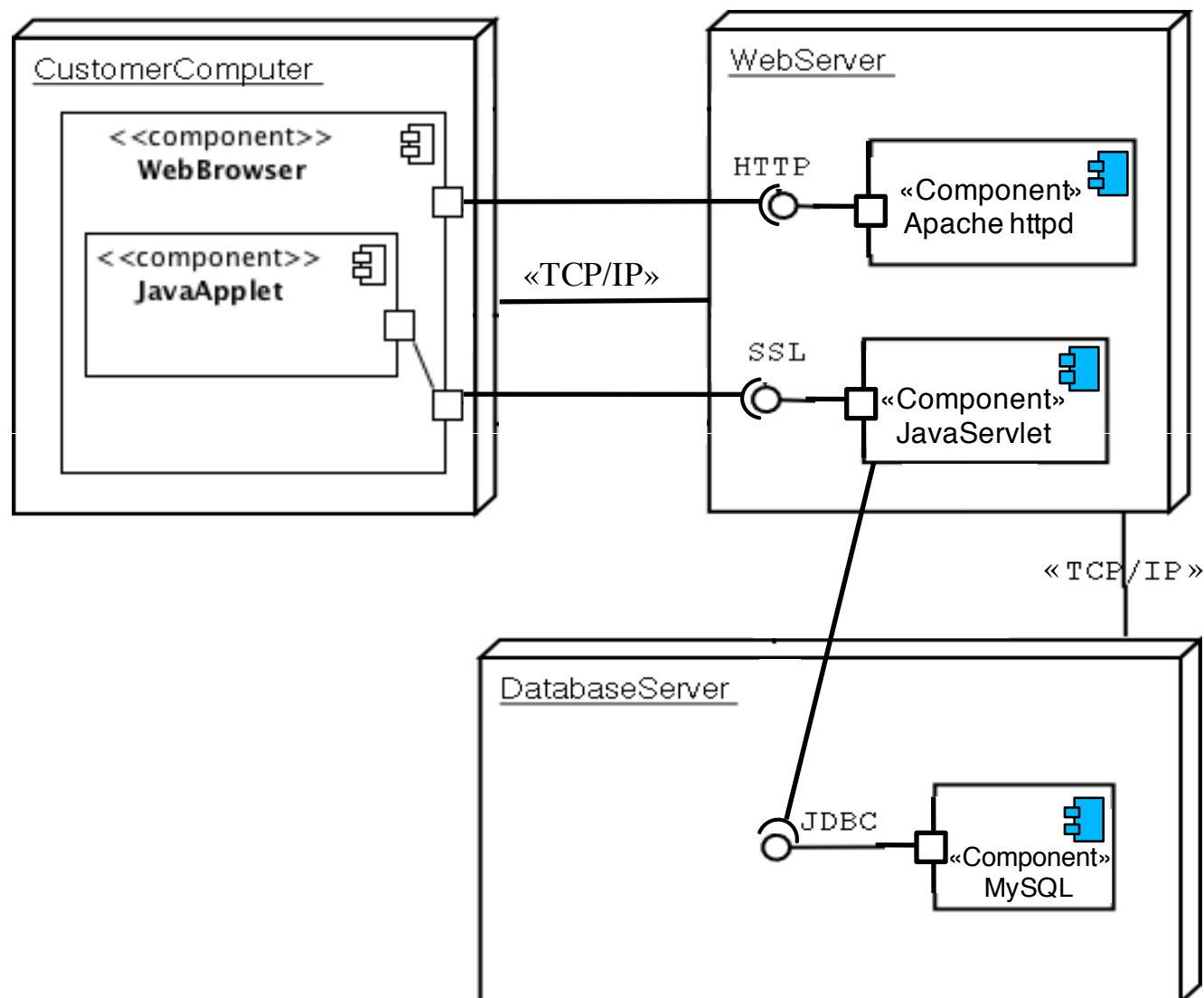
- A descrição pode ser refinada para detalhar os ambientes de execução em cada nó



- Especificação de dependências em tempo de execução

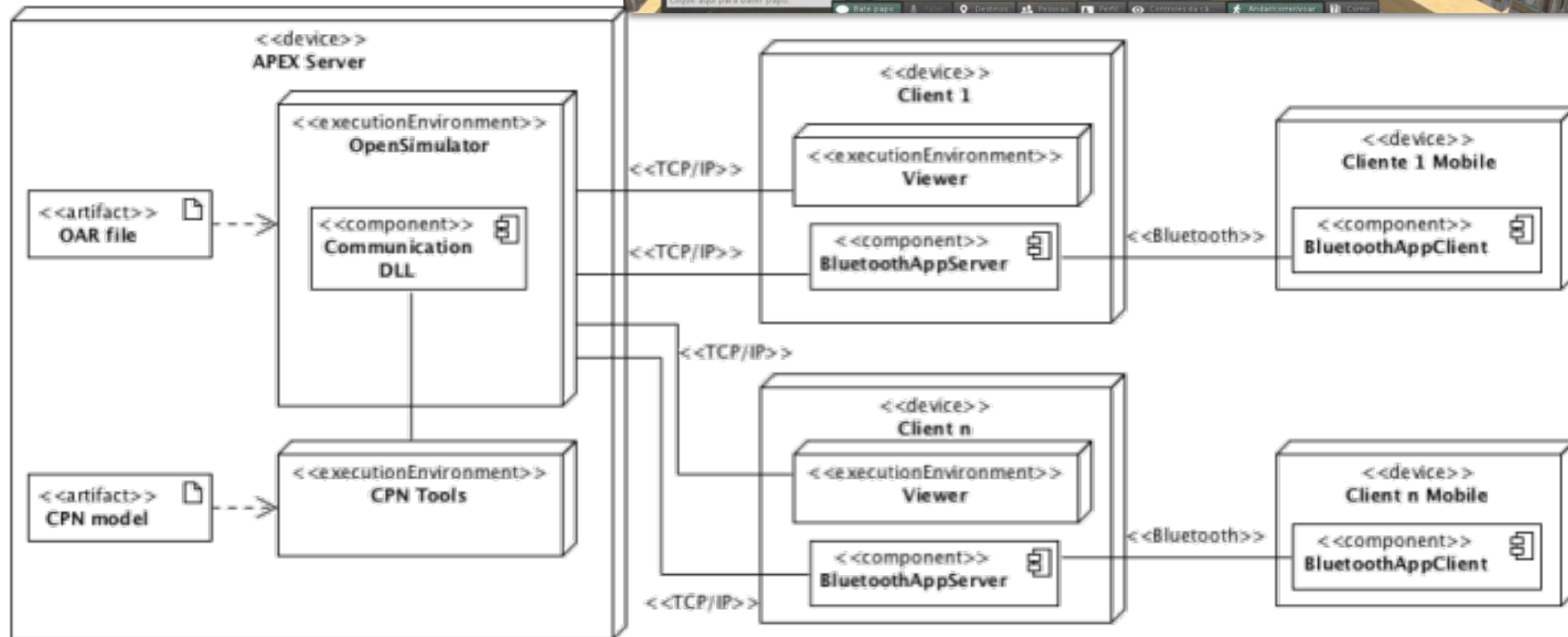


Instalação e componentes



Diagramas de Instalação

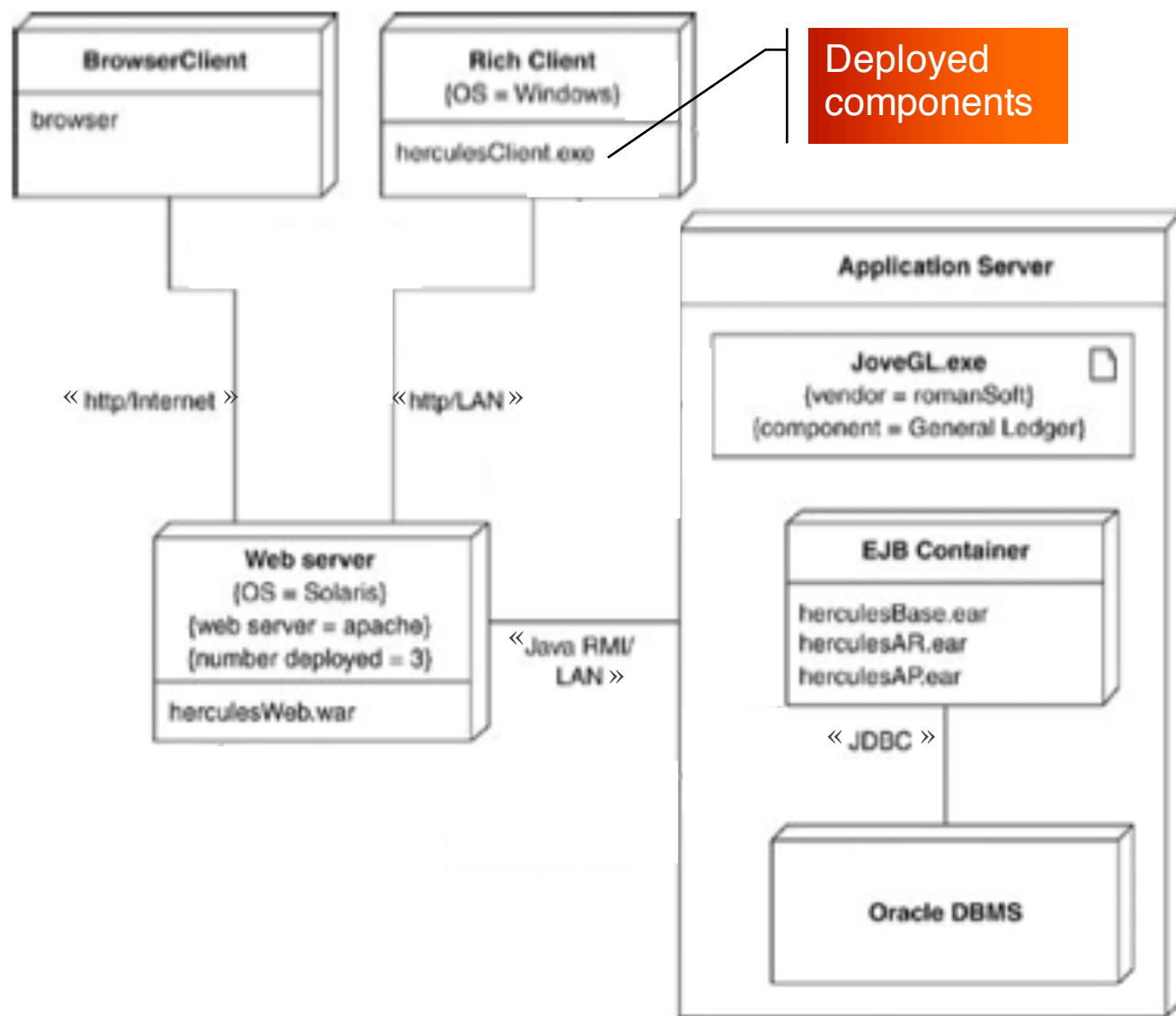
- Um exemplo





Diagramas de Instalação

- Outro exemplo





Modelação Estrutural/Comportamental

Sumário

- Modelação Estrutural com Diagramas de Instalação e de Componentes

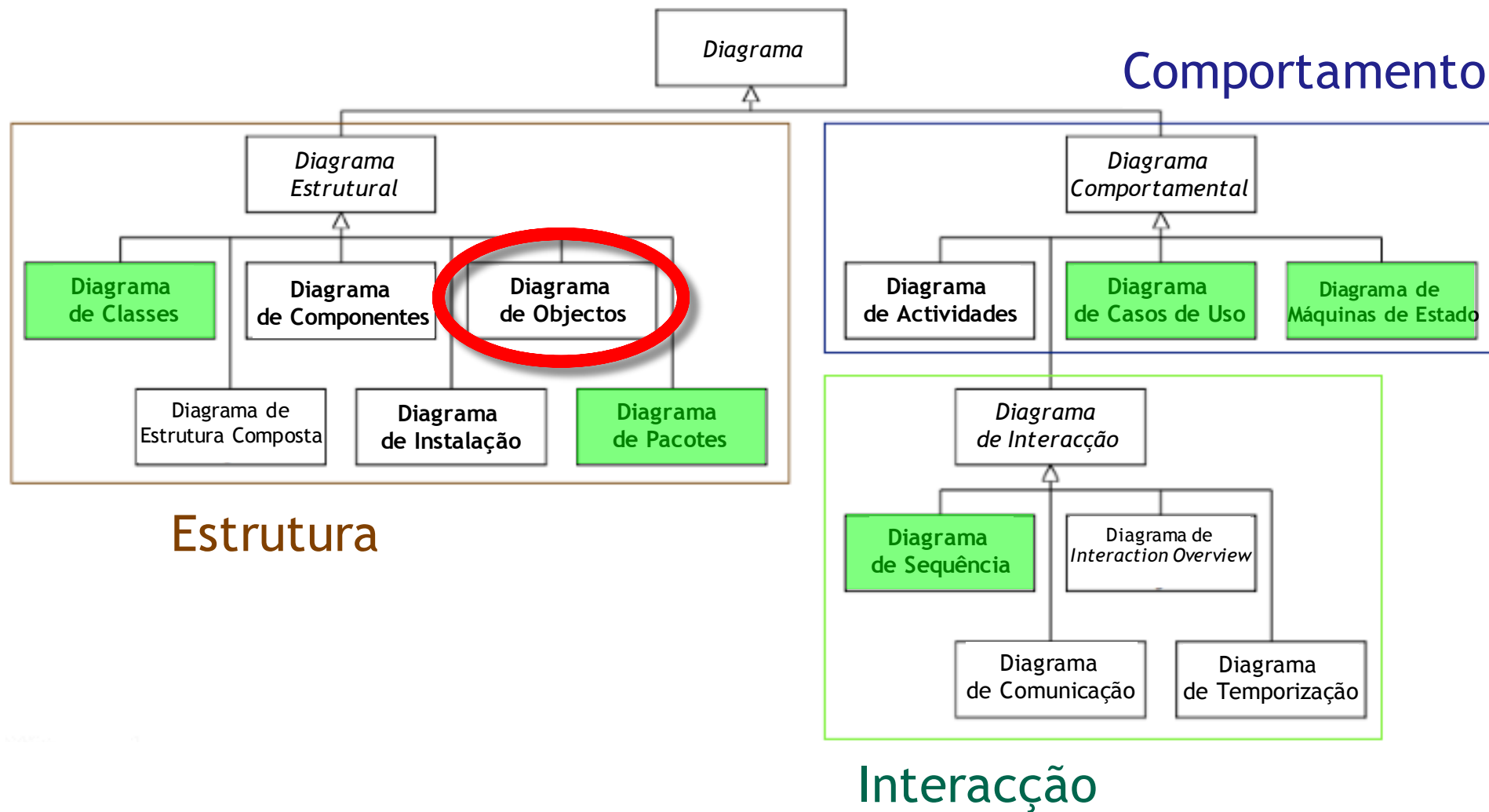


Desenvolvimento de Sistemas Software

Aula Teórica 22: Modelação Estrutural/Comportamental



Diagramas da UML 2.x



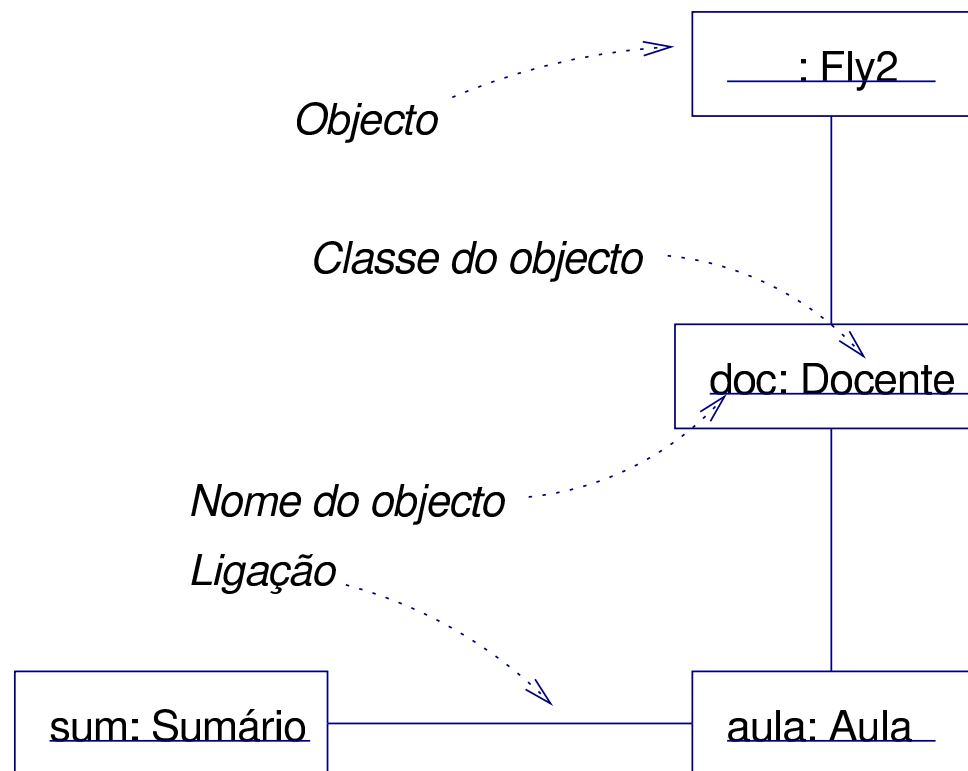


Diagramas de Objectos

- Imaginem que estavam a fazer *debug* de um módulo...
 - Como representariam o estado do módulo em cada momento?
- Diagramas de Classe representam uma visão de tempo de implementação
- Em tempo de execução, existem *instâncias* das classes
 - Quais?
 - Quantas?
 - Com que valores nos atributos?
 - Organizadas de que modo?



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.



Diagramas de Objectos

- Representação de objectos

umObjecto: UmaClasse

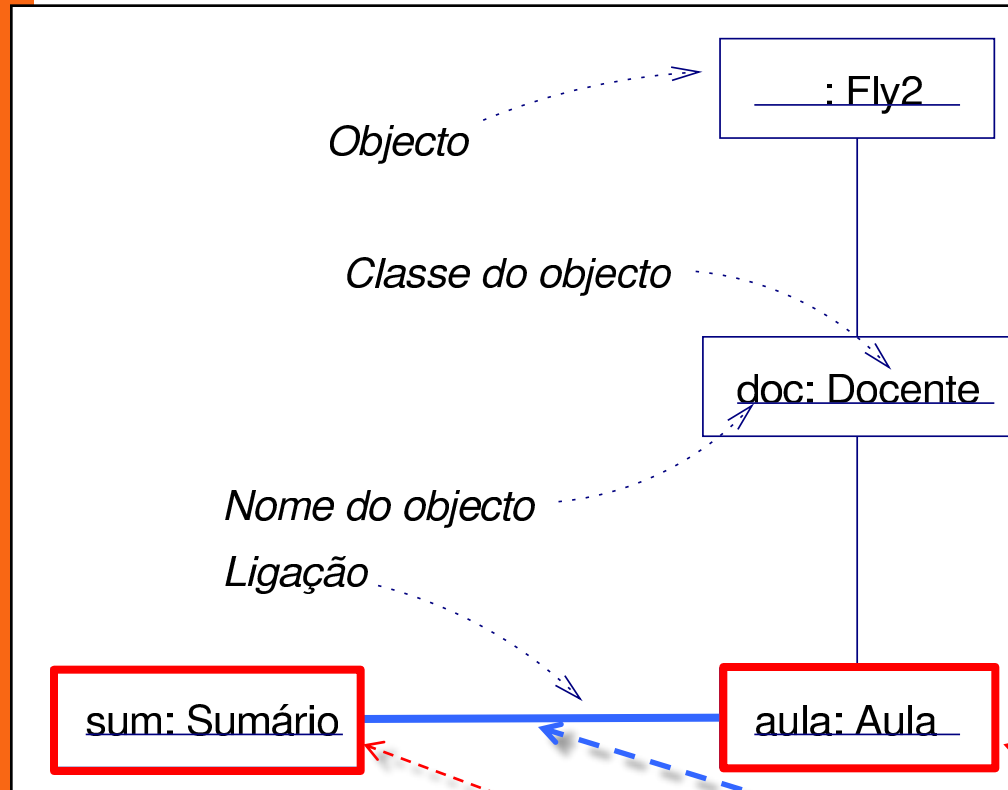
- Atributos

doc: Docente

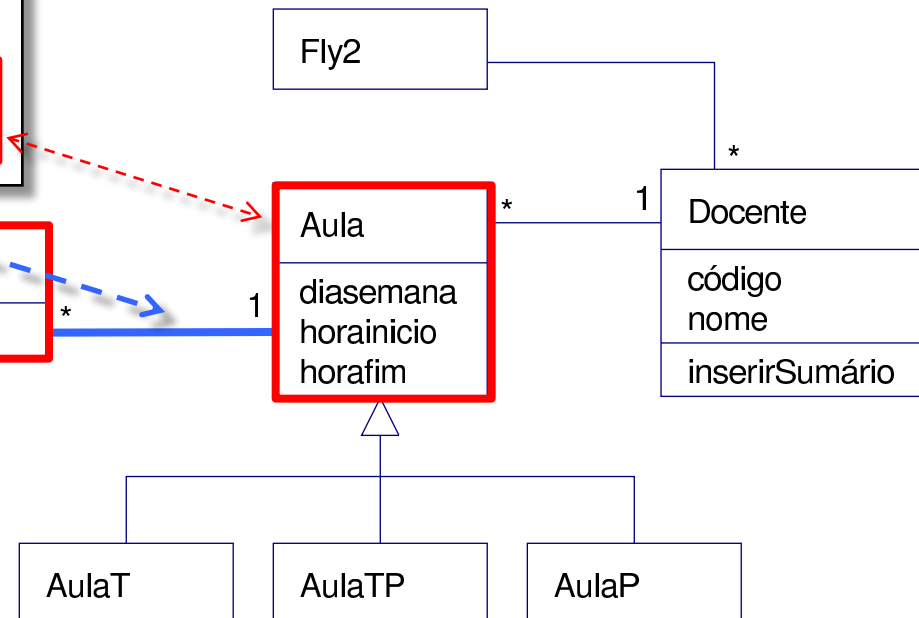
nome = “José”
número = 1234



Consistência de Diagramas de Objectos

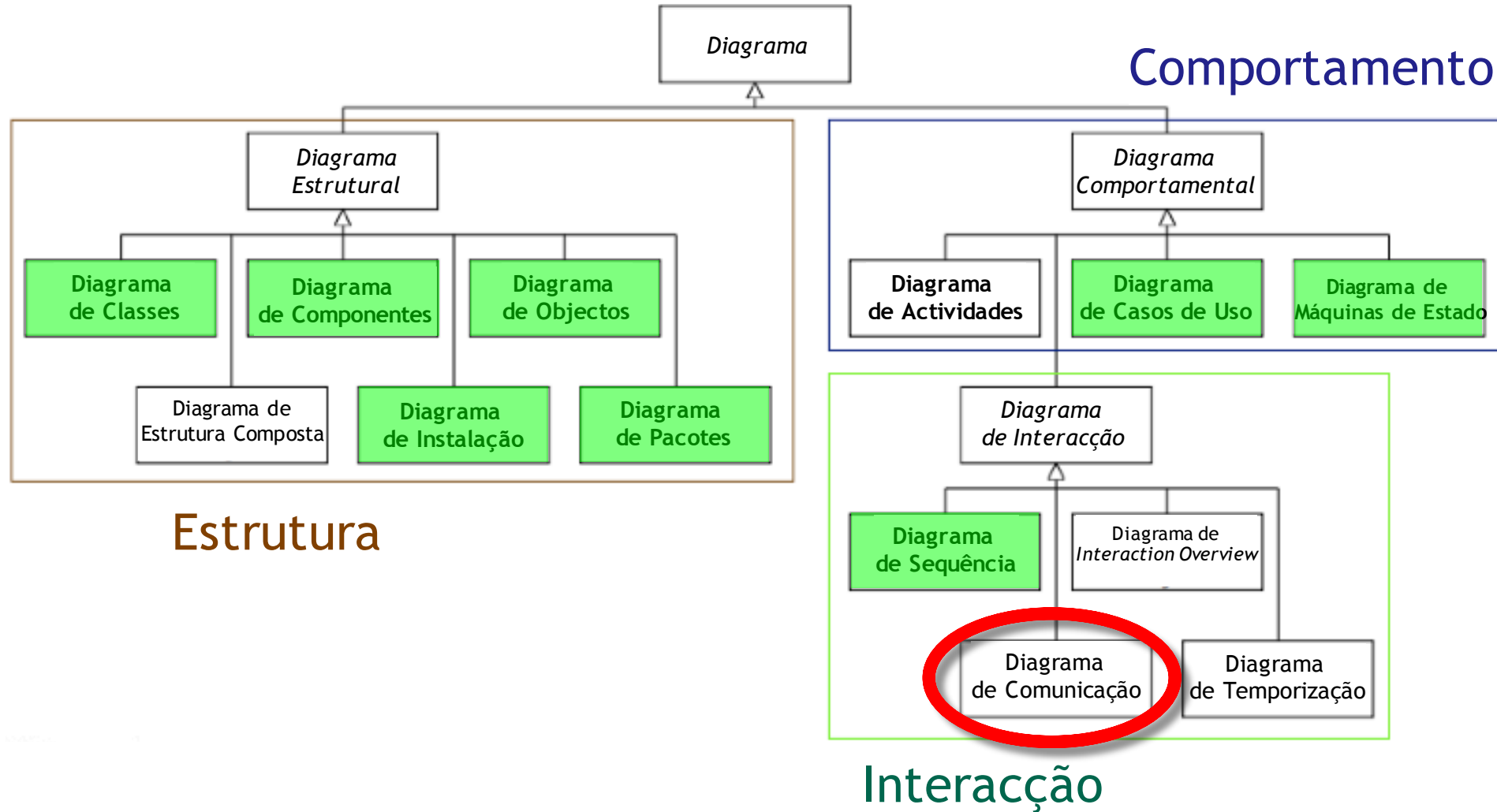


- Objetos no Diagrama de Objetos são instâncias de classes no correspondente Diagrama de Classes.
- Ligações no Diagrama de Objetos são instâncias de associações (ou de dependências) no correspondente Diagrama de Classes.





Diagramas da UML 2.x





Diagramas de Comunicação

- Representam comunicação (tal como Diag. Sequência)
- Centram-se nas relação entre objectos (e não na ordenação temporal das mensagens)
- Chamavam-se Diagramas de Colaboração em UML 1.x

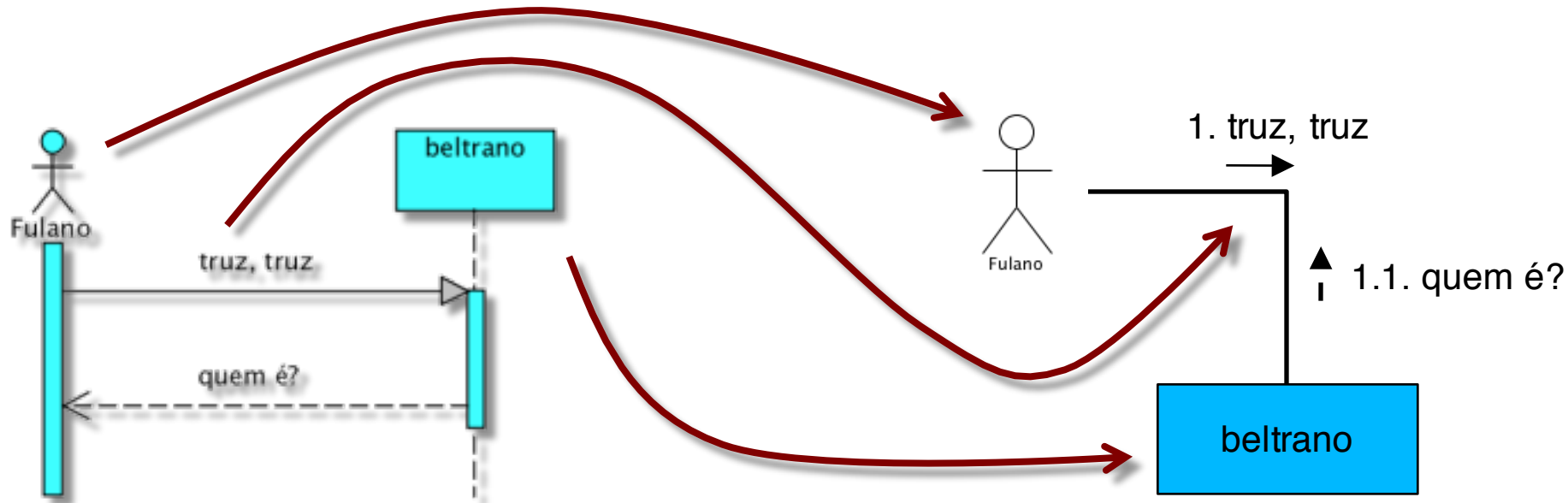
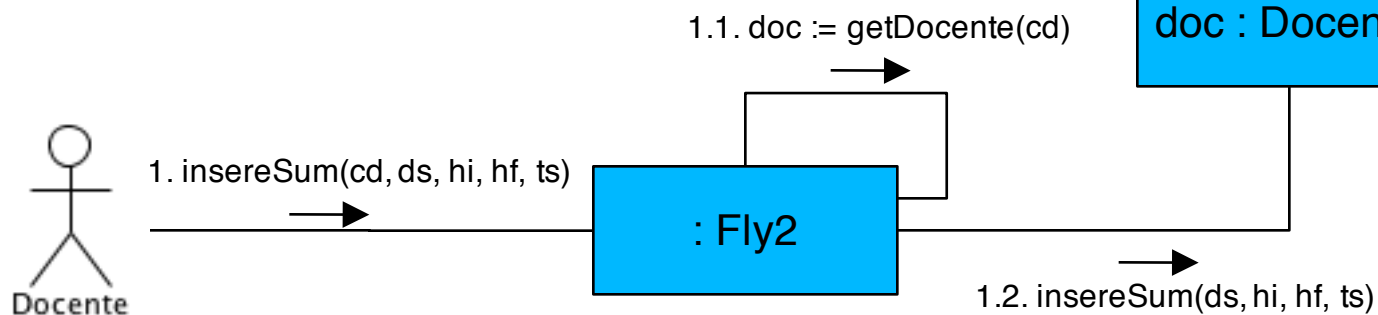
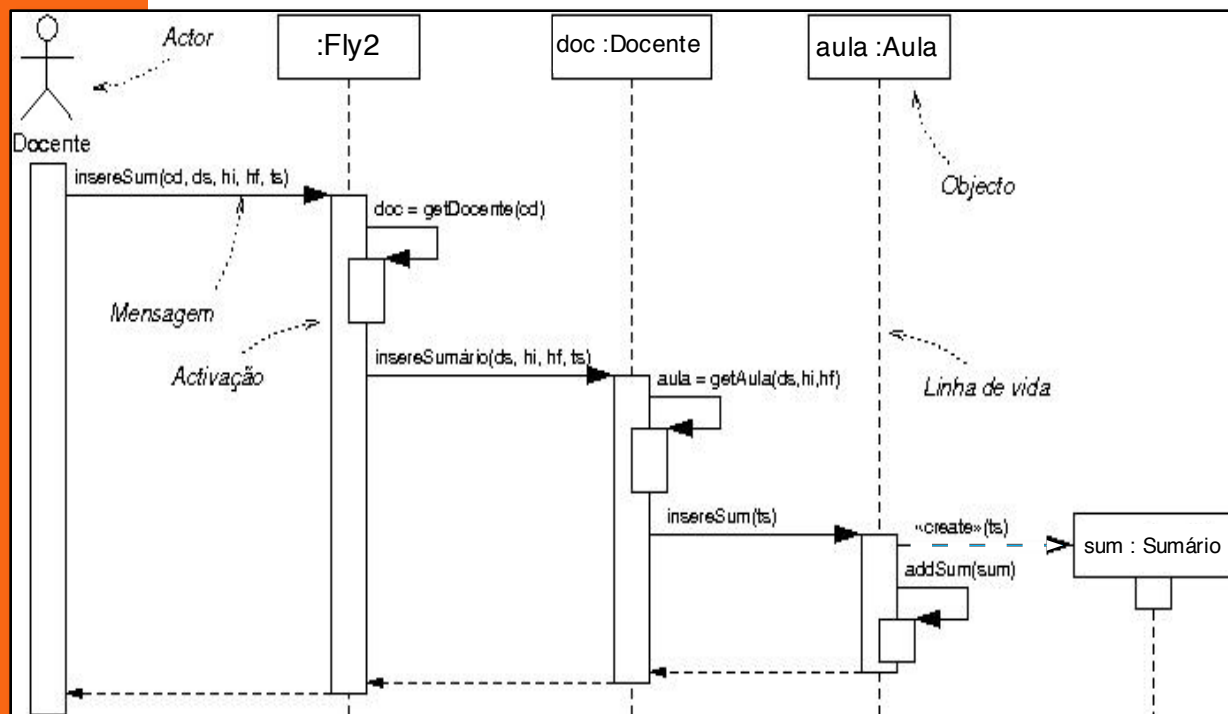


Diagrama de Sequência

Diagrama de Comunicação

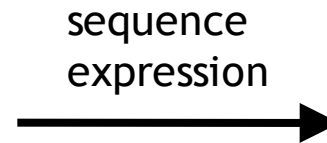




Diagramas de Comunicação

• Notação - Mensagens

- invocação síncrona



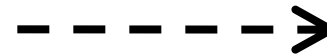
- invocação assíncrona



- return/resultado



- criar objectos



• *Sequence Expression*

sequence-expression ::= sequence-number ':' message-name

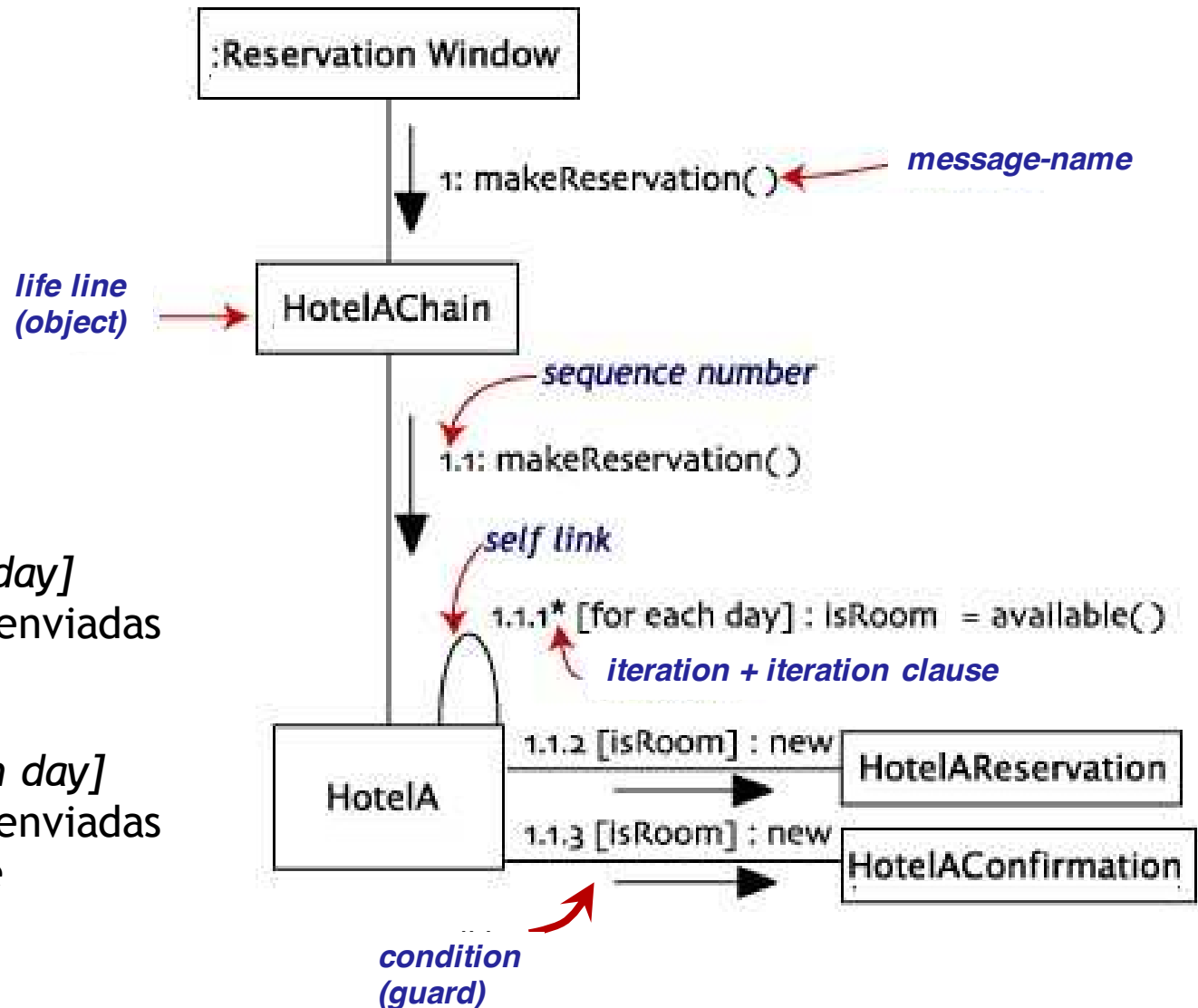
sequence-number ::= sequence-term ('.' sequence-term)*

sequence-term ::= [integer [name]] [recurrence]

recurrence ::= '[' guard ']' | '*' ['||'] ['[' iteration-clause ']']

Diagramas de Comunicação

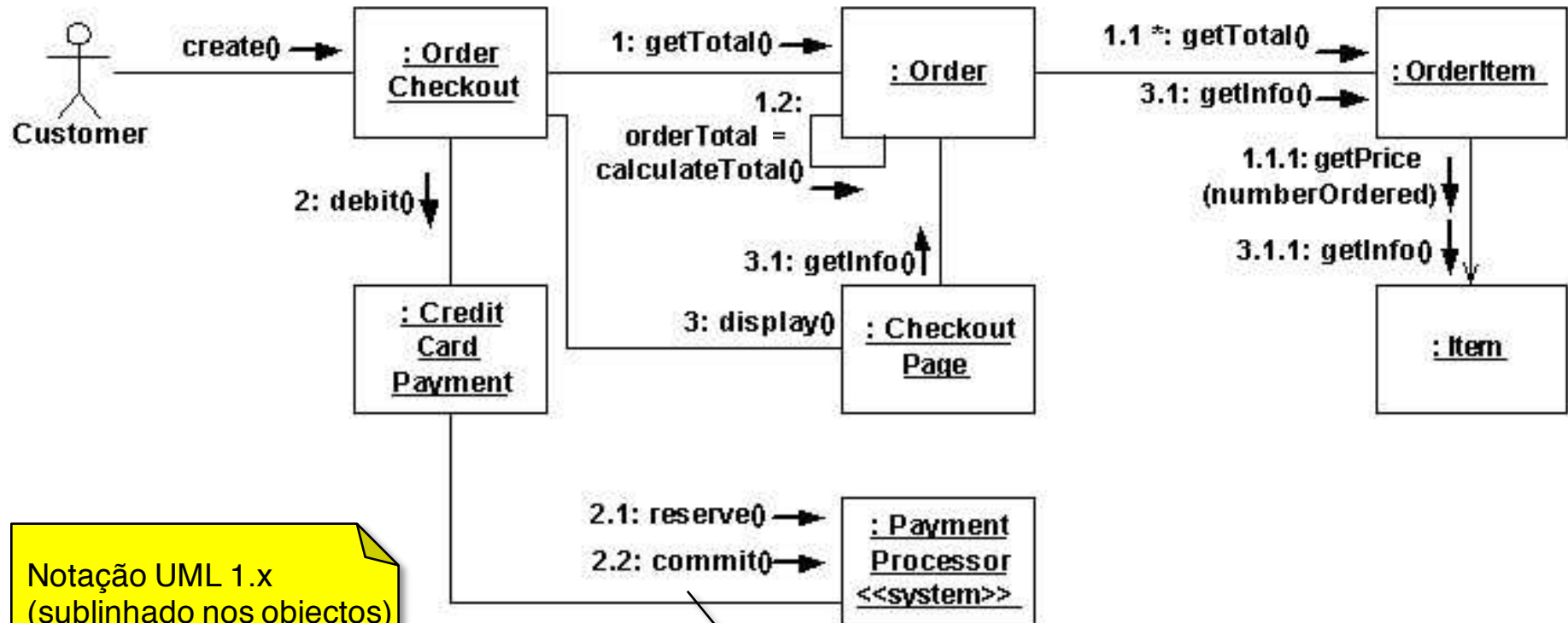
• Notação



- **1.1.1: * [for each day]**
as mensagens são enviadas sequencialmente
- **1.1.1: * || [for each day]**
as mensagens são enviadas concorrentemente

Diagramas de Comunicação

- Outro exemplo...



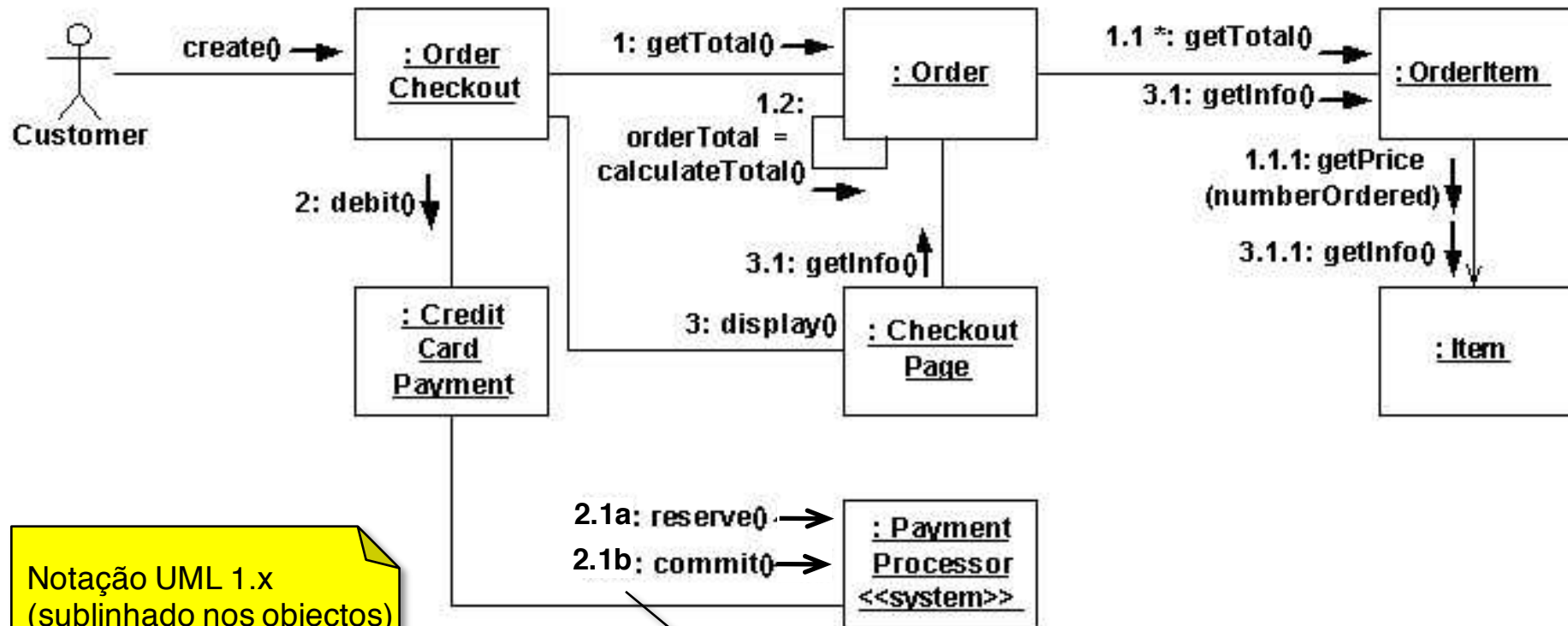
Notação UML 1.x
(sublinhado nos objectos)

Em UML 2.x passam a
life lines

Enviados em
sequência

Diagramas de Comunicação

- Outro exemplo...



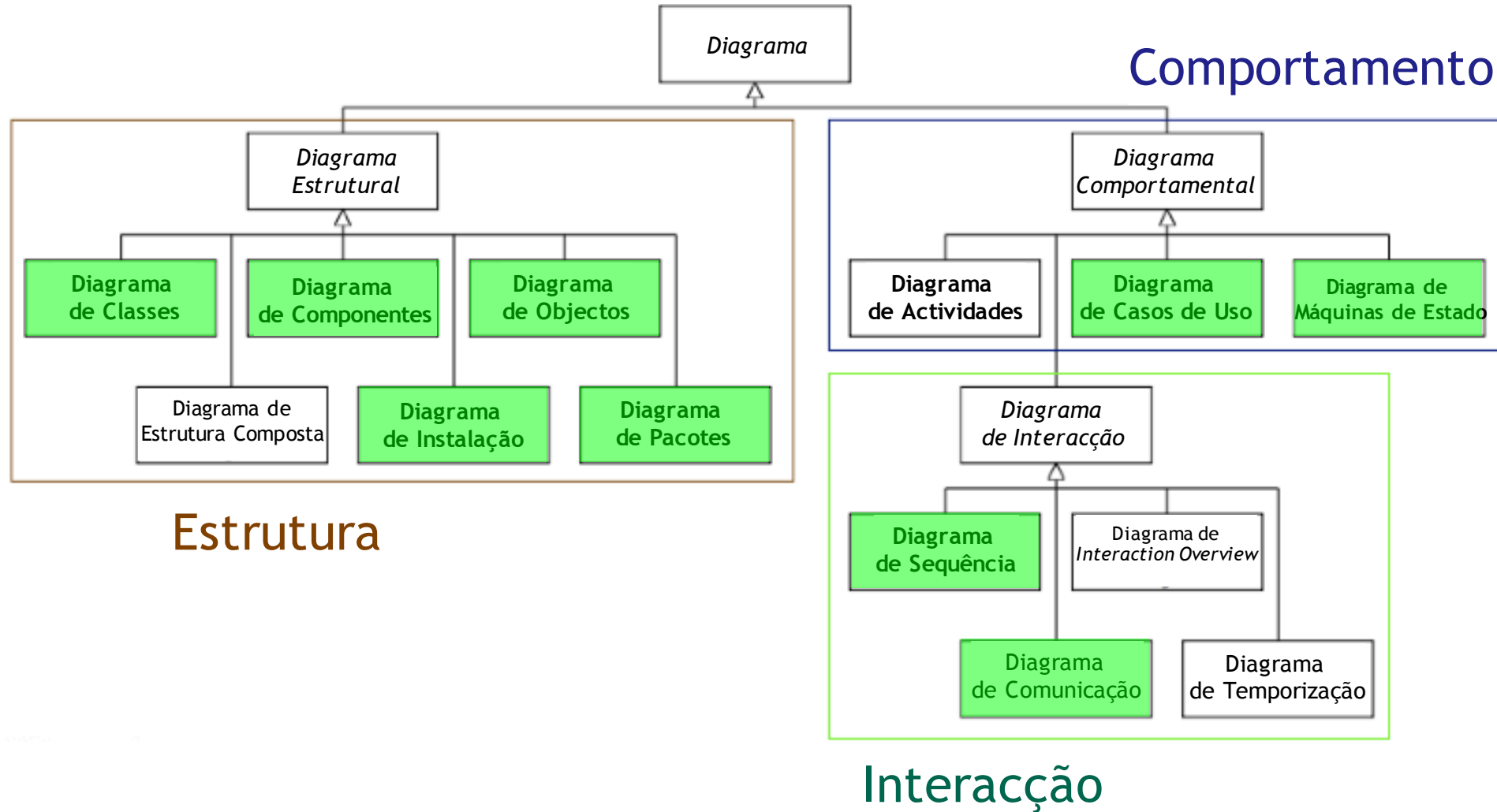
Notação UML 1.x
(sublinhado nos objectos)

Em UML 2.x passam a
life lines

Enviados
concorrentemente



Diagramas da UML 2.x





Modelação Estrutural/Comportamental

Sumário

- 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