
Diagrama Físicos

- Diagramas de Componentes e de Implantação -

Diagramas Físicos

- Os diagramas físicos permitem definir detalhes de implementação e implantação do software nas fases de projeto e implementação.
 - A UML provê dois tipos de diagramas físicos:
 - Diagrama de componentes: mostra as dependências de compilação e tempo de execução entre componentes de *software* (tais como, arquivos-fontes, DLLs).
 - Diagrama de implantação: mostra a distribuição de processos e os componentes pelos nós de processamento.
-

Diagramas Físicos

- Quando utilizar os diagramas físicos:
 - Para mostrar a estrutura física de um ambiente.
 - Para mostrar a estrutura de servidores.
 - Para mostrar a comunicação entre diversas máquinas com diversos protocolos.
-

Diagrama de Componentes

Diagrama de Componentes

- Diagramas de componentes mostram uma visão estática da implementação de um sistema.
 - São compostos por **componentes**, **interfaces** e **relações entre componentes**.
-

Diagrama de Componentes

- Principais usos:
 - **Modelagem do código-fonte:** na implementação das classes definidas durante a modelagem, o código gerado é armazenado fisicamente em arquivos. Um diagrama de componentes pode ser utilizado para gerenciar estes arquivos.
 - **Modelagem de versões executáveis:** uma versão de um sistema envolve combinações específicas de diversas partes. O diagrama de componentes pode modelar os diversos componentes necessários para uma determinada versão executável do sistema.
-

Diagrama de Componentes

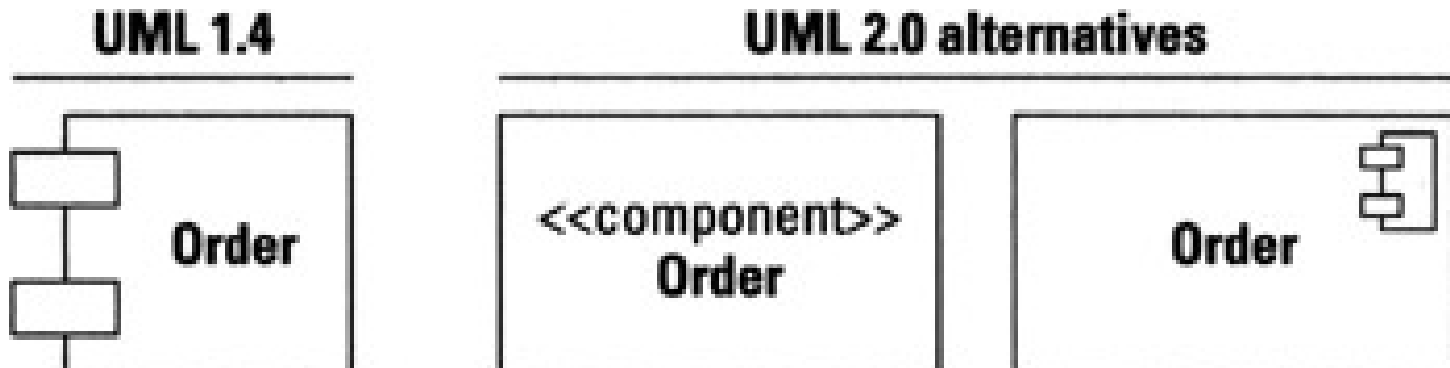
- Principais usos:
 - **Modelagem de bancos de dados físicos:** considerando-se que as informações do sistema serão armazenadas em arquivos ou tabelas de um banco de dados, um diagrama de componentes pode mostrar os arquivos (ou tabelas) do banco de dados e seus relacionamentos.
 - **Modelagem de sistemas adaptáveis:** a execução de alguns sistemas baseia-se no uso de componentes dinâmicos (carga dinâmica, agentes móveis, etc.), que podem ser descritos, de forma conjunta, por diagramas de componentes e outros diagramas da UML.
-

Diagrama de Componentes

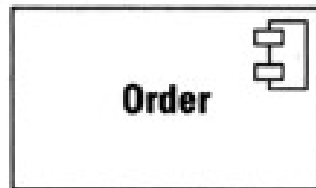
∴ Notação ∴

Componente

- Um componente descreve um módulo físico do sistema, como código-fonte, código executável, bibliotecas, arquivos, tabelas, documentos, entre outros.
- Representação gráfica:



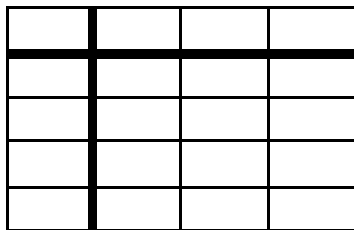
Ícones e Estereótipos



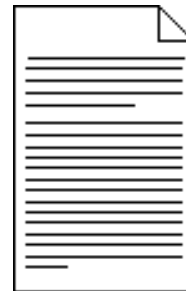
<<executable>>



<<library>>



<<table>>



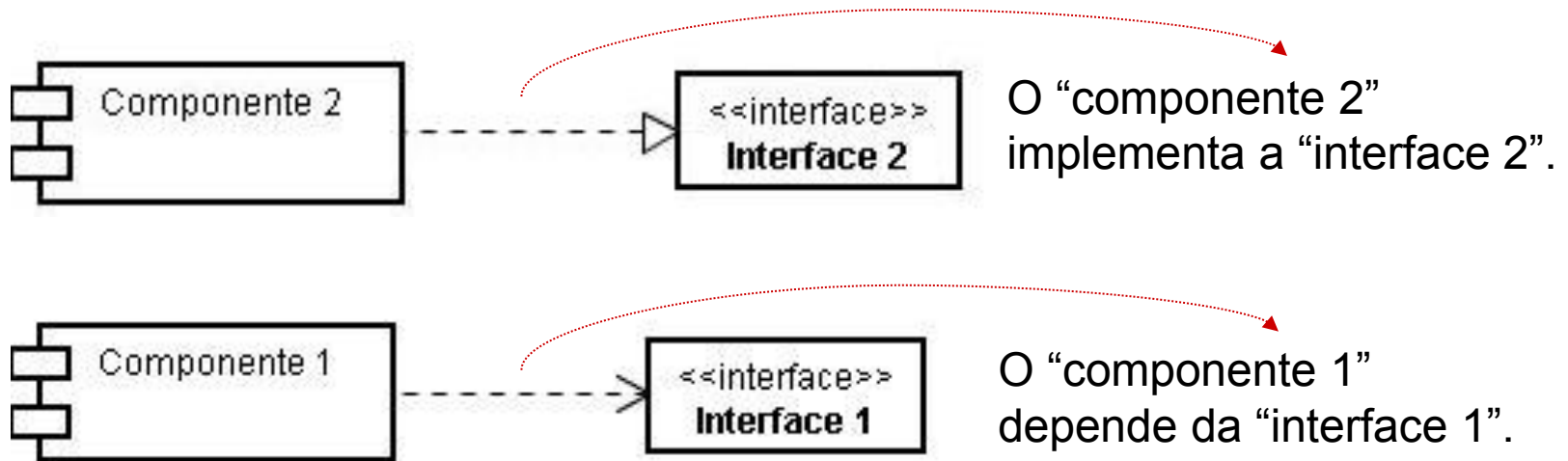
<<file>>



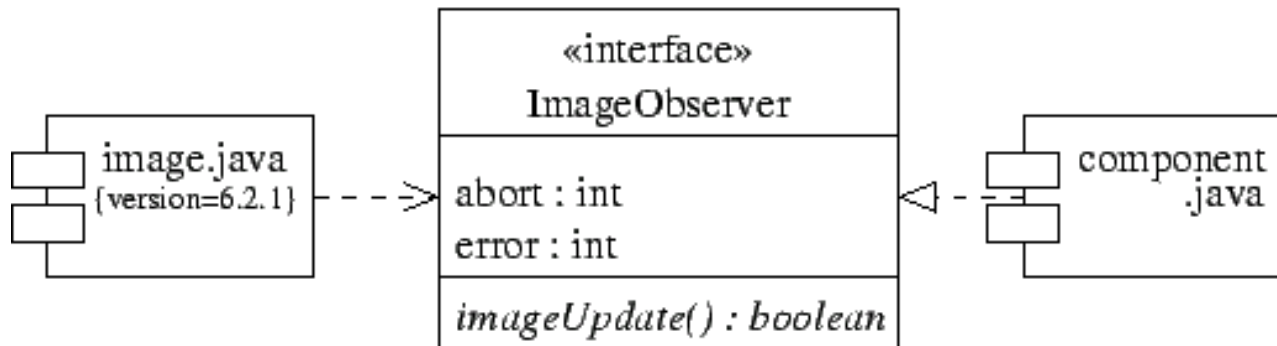
<<document>>

Interfaces

- Interfaces pode ser entendidas como uma coleção de operações usadas para especificar um serviço disponibilizado pelo componente.
- Usualmente um componente **implementa** (realiza) ou **depende** de uma interface.



Interfaces - Exemplo

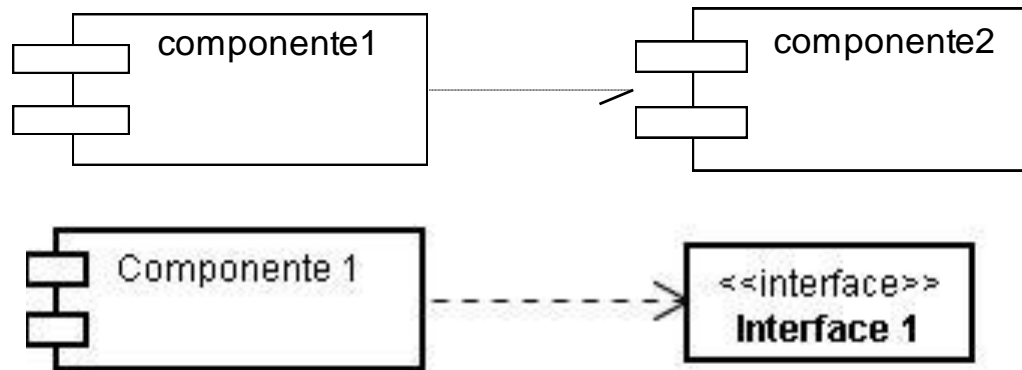


OU



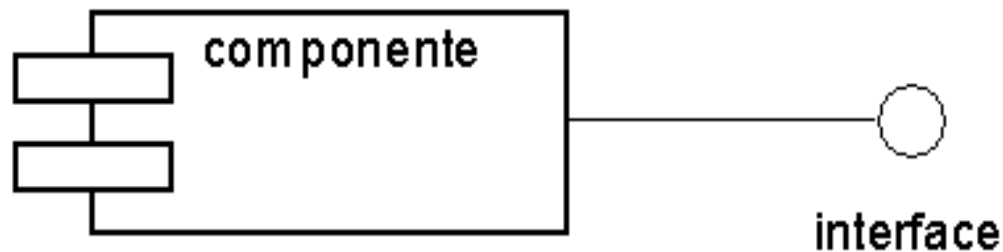
Relacionamentos

1. **Dependência:** quando um componente depende de outro componente ou de uma interface. A dependência pode ser unidirecional ou bidirecional. Pode-se definir dependências com estereótipos.
 - Uma dependência descreve como um componente alterado pode afetar outros componentes (por exemplo: comunicação entre componentes, dependências de compilação).

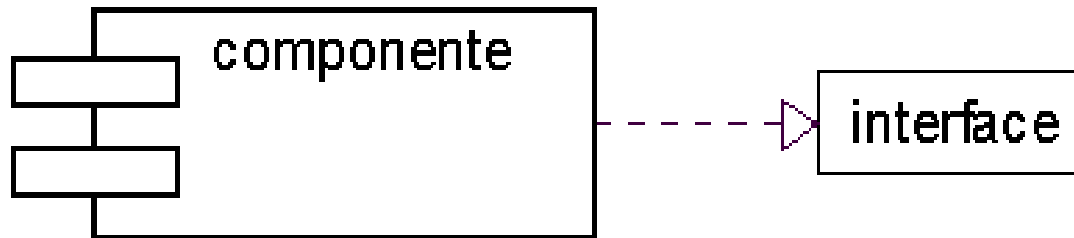


Relacionamentos

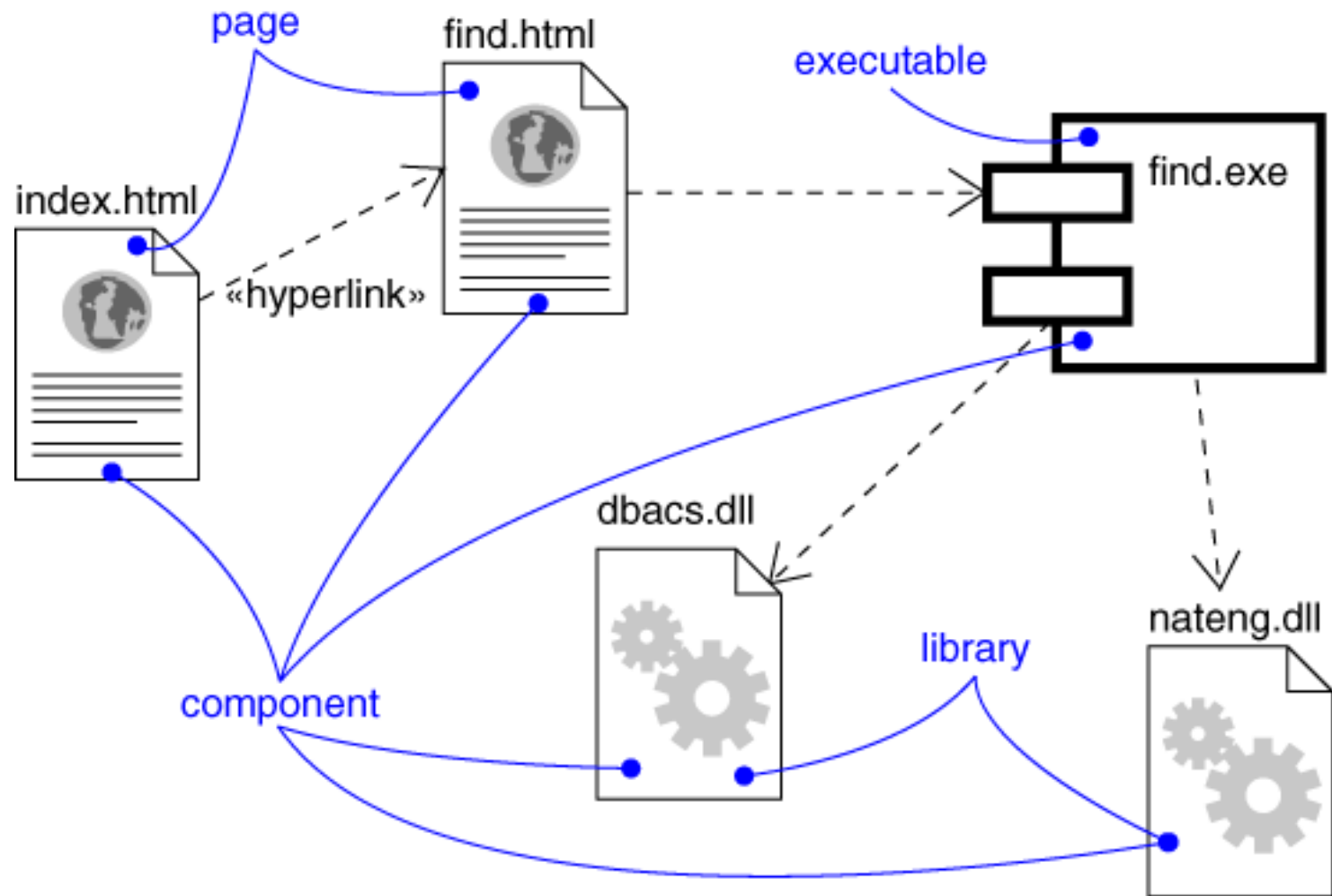
2. **Realização:** quando um componente implementa uma interface.



ou



Exemplo



Tarefa 1

- Faça um diagrama de componentes para os componentes de instalação da aplicação **WinAp**. A aplicação consiste em um conjunto de componentes, que são:
 - **winap.exe**: executável da aplicação
 - **pplib32.dll**, **sde32.dll**, **sdemdb32.dll**: bibliotecas com código binário que providenciam funcionalidades adicionais à aplicação.
 - **winap.hlp**: material de ajuda sobre a aplicação.
 - **winap.ini**: material de configuração da aplicação.
 - **entrada.db**, **saida.db**: tabelas da base de dados de suporte.
 - Observações:
 - A biblioteca **sdemdb32.dll** depende da biblioteca **sde32.dll**.
 - O executável **winap.exe** (i.e., a aplicação WinAp) só funciona se todos os outros componentes tiverem sido instalados adequadamente.
-

Tarefa 1 – Possível Resposta

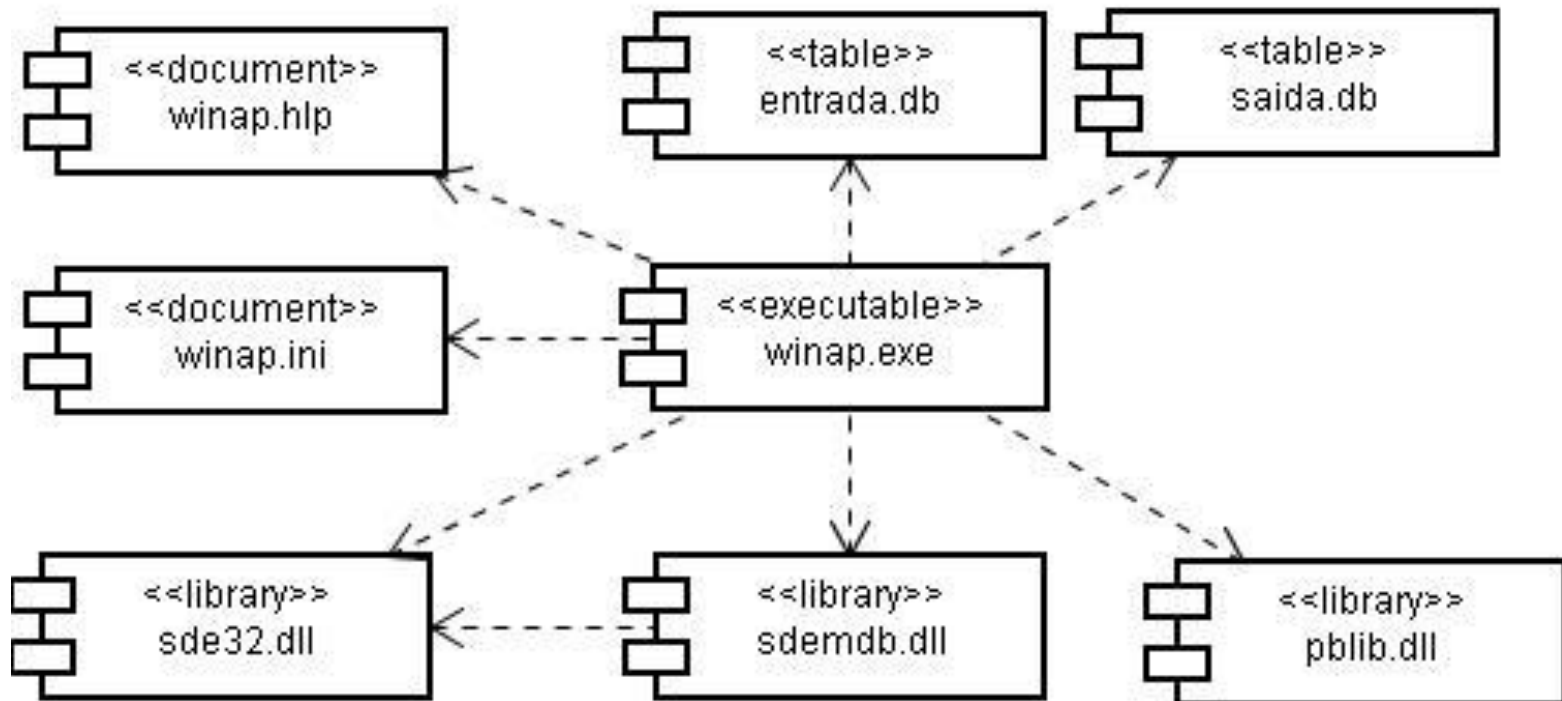


Diagrama de Implantação *(Deployment)*

Diagrama de Implantação

- Um diagrama de implantação provê uma visão da relação física entre componentes de *software* e de *hardware*.
 - Cada nodo de um diagrama de implantação tipicamente representa um tipo de hardware, com um PC, um servidor, um drive de CD, entre outros.
 - Mostra os **nodos do sistema**, os **componentes existentes e suas conexões**.
-

Diagrama de Implantação

- Principais usos:
 - **Modelagem de sistemas embarcados:** sistemas embarcados interagem com um conjunto de sensores. Um diagrama de implantação permite modelar como um processador atua conjuntamente com uma série de dispositivos (sensores).
 - **Modelagem de sistemas cliente/servidor:** um sistema cliente/servidor separa a interface com o usuário (cliente) dos dados persistentes (servidor). Um diagrama de implantação serve para modelar a topologia destes sistemas.
-

Diagrama de Implantação

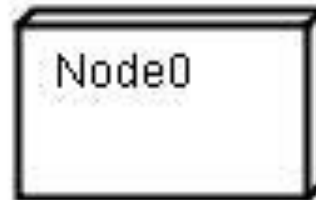
- Principais usos:
 - **Modelagem de sistemas totalmente distribuídos:** é possível modelar um sistema com processamento distribuído em diversos processadores. Um diagrama de implantação pode descrever como serão as conexões entre os processadores e seus periféricos.
-

Diagrama de Implantação

∴ Notação ∴

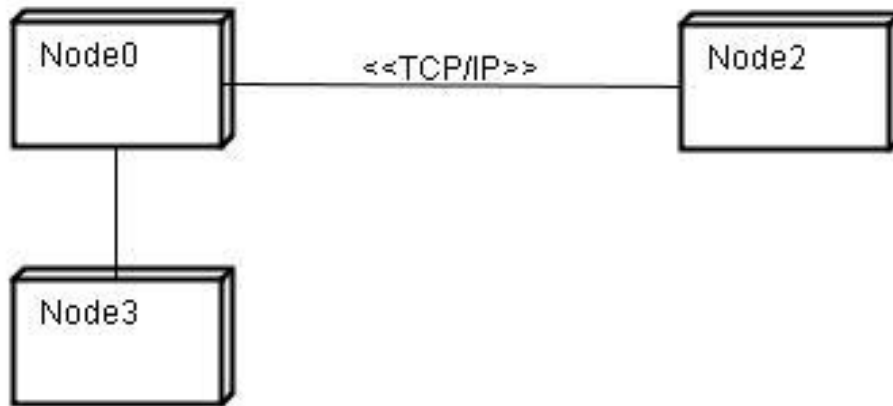
Nodo

- Um nodo é um objeto físico, em tempo de execução, que representa um recurso computacional que possui, geralmente, memória e, freqüentemente, capacidade de processamento.
- Os nodos geralmente são estereotipados como:
 - «processor»: denota um nó que pode executar um componente de software.
 - «device»: denota um nó que não tem capacidade para executar componentes de software, e.g., uma impressora, um scanner, ou um monitor.
- Representação gráfica:



Conexão

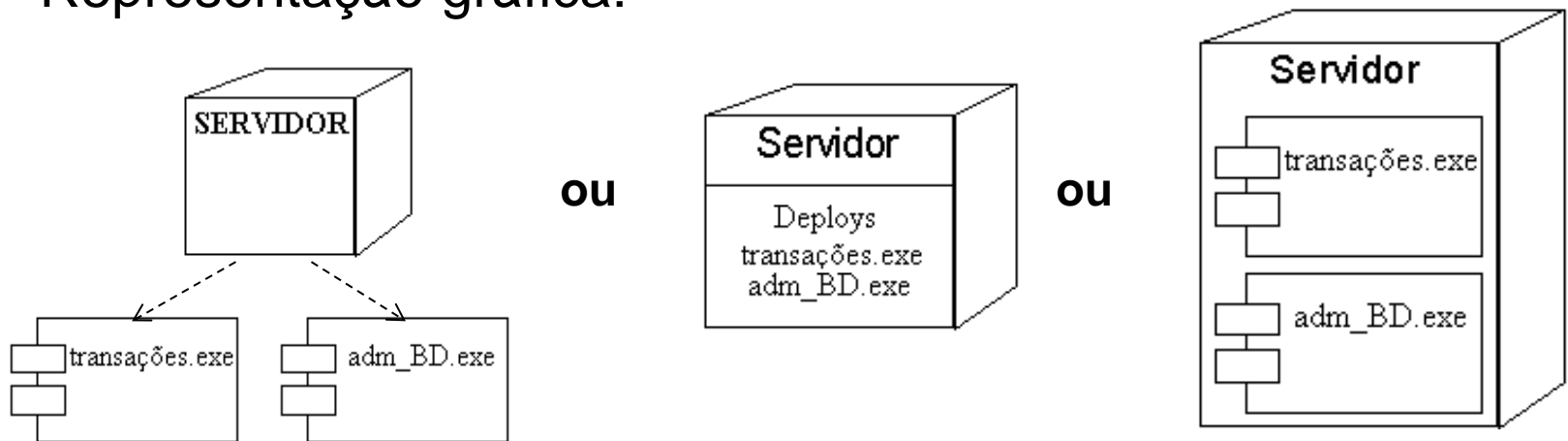
- Uma conexão representa a ligação de *hardware* entre dois nodos. Usualmente é bidirecional. Uma conexão pode ser direta (tal como um cabo serial) ou indireta (tal como comunicação via-satélite).
- Representação gráfica:



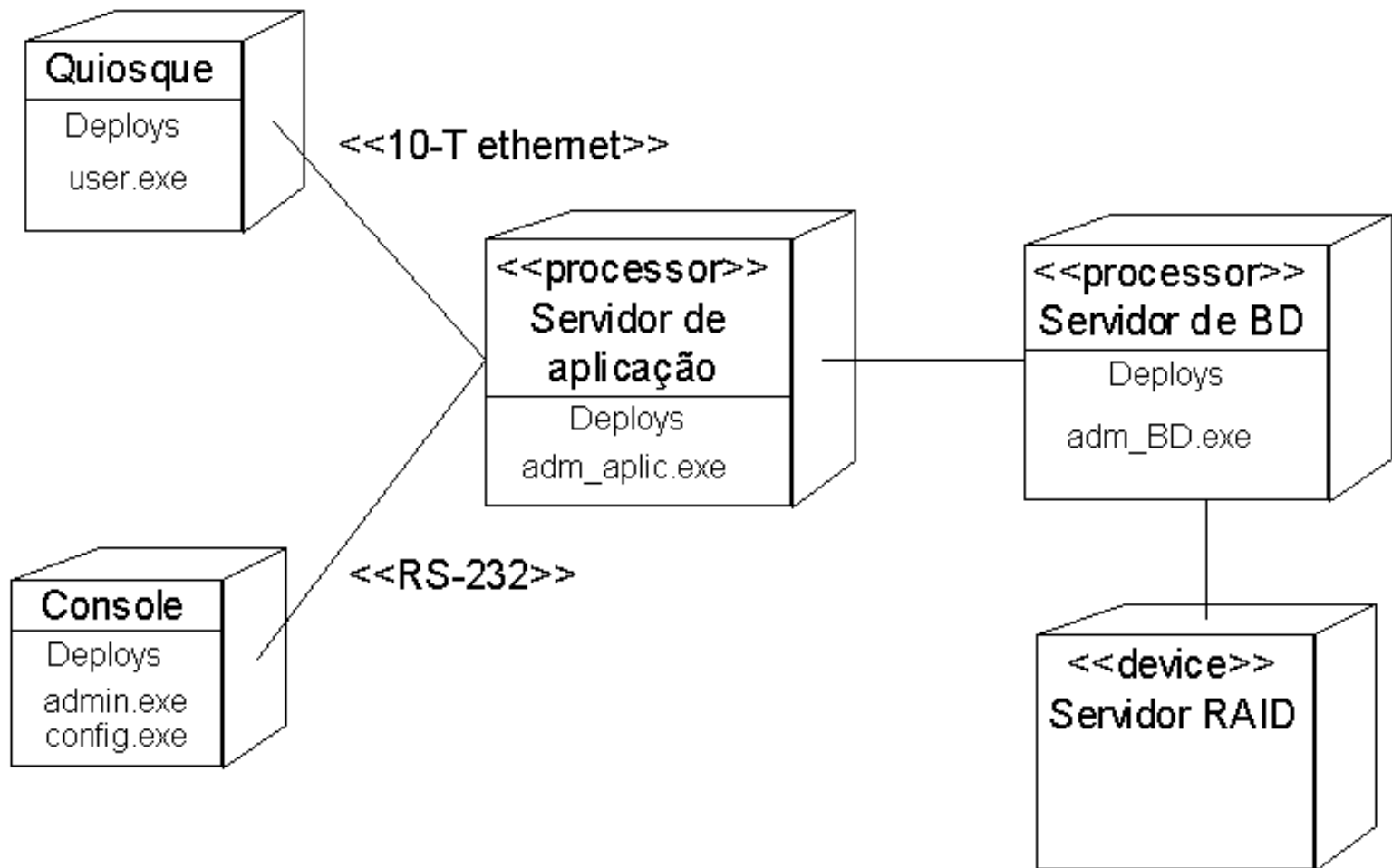
Uma conexão não precisa de nome. Opcionalmente pode-se definir um estereótipo para a conexão.

Nodos e componentes

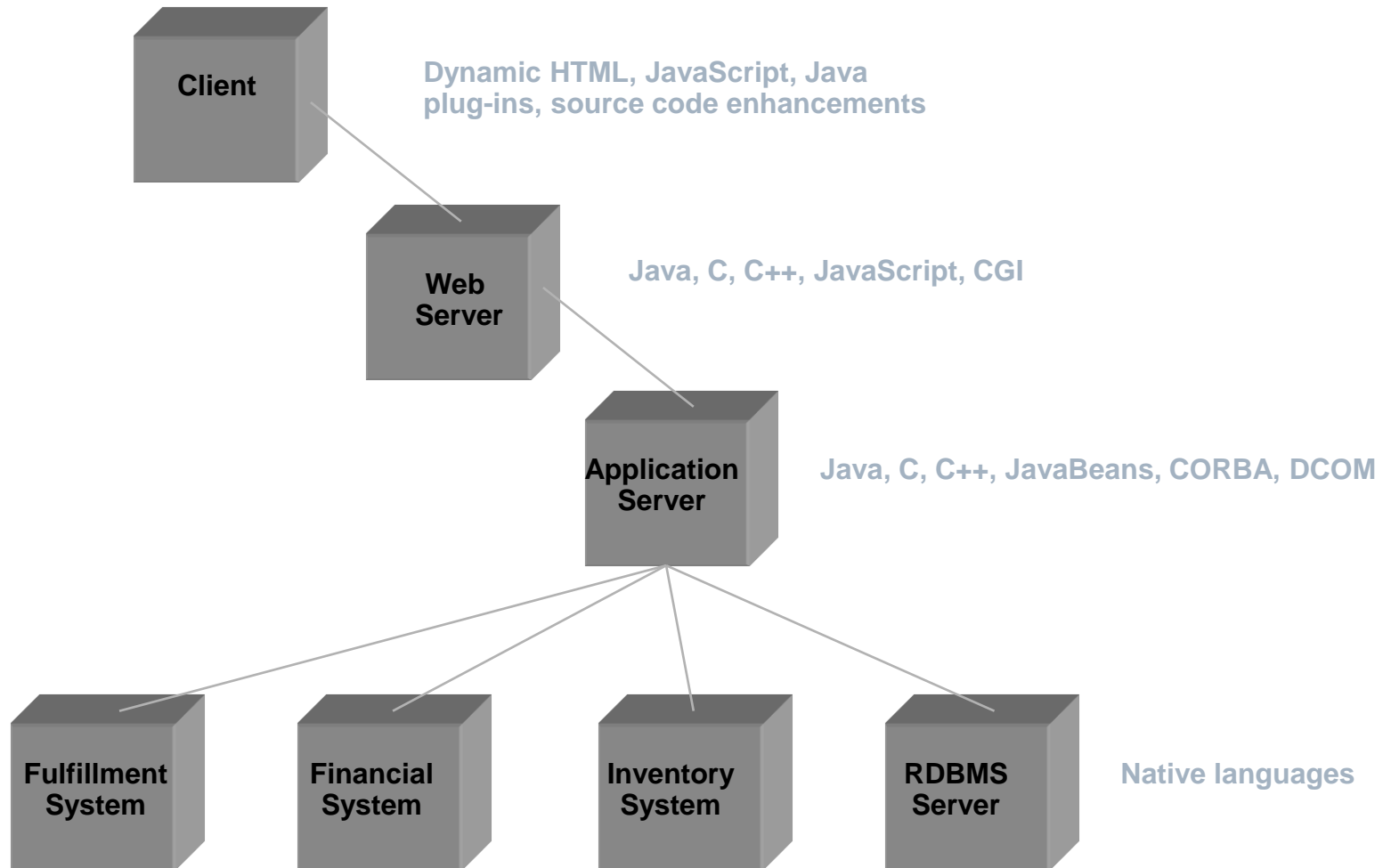
- A conexão entre o *software* e *hardware* é definida pelo relacionamento entre nodos e componentes. Um nodo executa um (ou mais) componentes.
- É possível indicar os componentes localizados (*deployed*) em cada nó, representando-os dentro do nó ou fora com dependência estereotipada.
- Representação gráfica:



Exemplo 1



Exemplo 2: Sistema na Internet



Tarefa 2

Utilizando a notação do diagrama de implantação, modele um *data center* com os seguintes recursos:

- 1 servidor de aplicação
 - 1 servidor de impressão
 - 2 impressoras (uma laser e outra jato de tinta)
 - 3 micros
 - 1 laptop
 - 1 switch ligando os equipamentos via TCP/IP
-

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.