

# Diagrama de Implantação

João Pedro Oliveira Batisteli

# Tipos de Diagramas UML

- **Estruturais** (foco estático):
  - Classe, Objeto, Pacotes, Componentes, Implantação...
- **Comportamentais** (foco dinâmico):
  - Casos de Uso, Sequência, Atividade, Comunicação, Estados...
- **Por que dividir?**

# Tipos de Diagramas UML

- **Estruturais** (foco estático):
  - Classe, Objeto, Pacotes, Componentes, Implantação...
- **Comportamentais** (foco dinâmico):
  - Casos de Uso, Sequência, Atividade, Comunicação, Estados...
- **Por que dividir?**
  - **Estruturais** mostram *o que existe*
  - **Comportamentais** mostram *como acontece*

# Onde o Diagrama de Implantação se Encaixa

- Pertence ao grupo de **diagramas estruturais**
- Mostra a **arquitetura física** do sistema
- Representa:
  - Dispositivos de hardware
  - Ambientes de execução
  - Softwares/artefatos implantados
- Complementa outros diagramas (ex.: **componentes e casos de uso**).

# Exemplo - Sistema de e-commerce

## **Componentes principais:**

- Interface Web
- Módulo de Pagamento
- Módulo de Carrinho
- Banco de Dados

organização lógica do código e  
suas dependências

# Exemplo - Sistema de e-commerce

## Componentes principais:

- Interface Web
- Módulo de Pagamento
- Módulo de Carrinho
- Banco de Dados

organização lógica do código e suas dependências

## Diagrama de Implantação (visão física)

- **Servidor Web:** hospeda os componentes Interface Web, Pagamento e Carrinho.
- **Servidor de Banco de Dados:** hospeda o artefato do Banco de Dados
- **Cliente (PC ou Smartphone):** acessa a Interface Web via navegador

**infraestrutura necessária** para suportar os componentes.

# Complementaridade

- **Diagrama de Componentes:** “O que o sistema tem” (visão de software lógico).
- **Diagrama de Implantação:** “Onde o sistema roda” (visão de arquitetura física).
- Juntos: permitem **ligar software ↔ hardware**, facilitando a comunicação entre equipe de desenvolvimento e infraestrutura.

# Modelagem

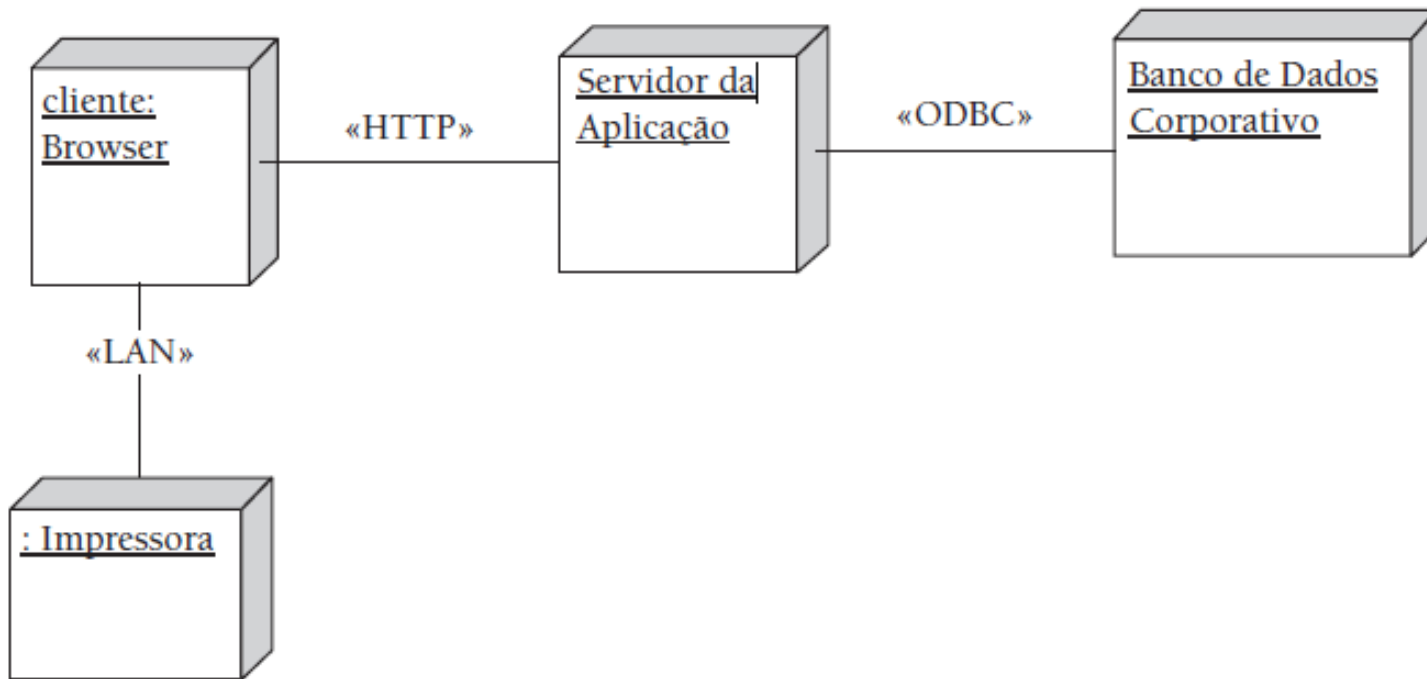
- Foco na parte física do projeto:
  - Os diagramas de **instalação/implantação** mostram o **layout físico** de um sistema, revelando **quais partes do software são executadas em quais partes do hardware**.
- O foco é a arquitetura física de um sistema
- São relativamente simples.



# Diagrama de Implantação

- Existem **dois tipos** de diagrama de implementação:
  - Diagrama de Implantação (*deployment diagram*).
  - Diagrama de Componentes.

# Diagrama de Implantação - Elementos



Os elementos do diagrama são os **nós** e suas **conexões**

# Diagrama de Implantação - Nós

- **Definição:** Um **nó** é uma unidade física que representa um **recurso computacional** na arquitetura do sistema.
- Normalmente possui **memória, capacidade de processamento** ou **armazenamento**.
- Pode ser um dispositivo **físico** ou um **ambiente de execução**.

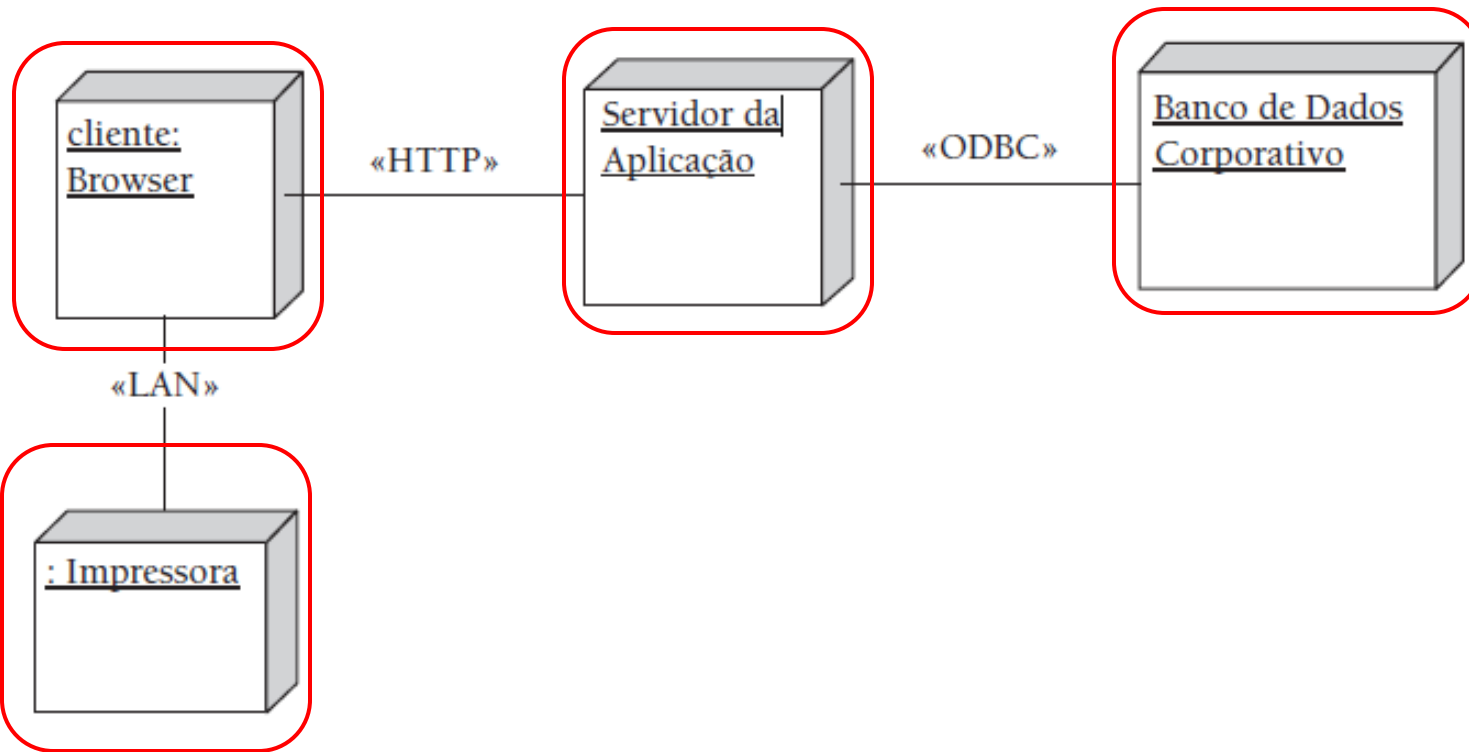
# Diagrama de Implantação - Nós

## **Exemplos:**

- Servidor de Aplicação (processa requisições do sistema).
- Servidor de Banco de Dados (armazenamento de dados).
- Smartphone ou PC Cliente (onde o usuário acessa o sistema).
- Máquina Virtual na Nuvem (ambiente de execução remoto).

**Em geral:** qualquer objeto físico relevante para o sistema de software.

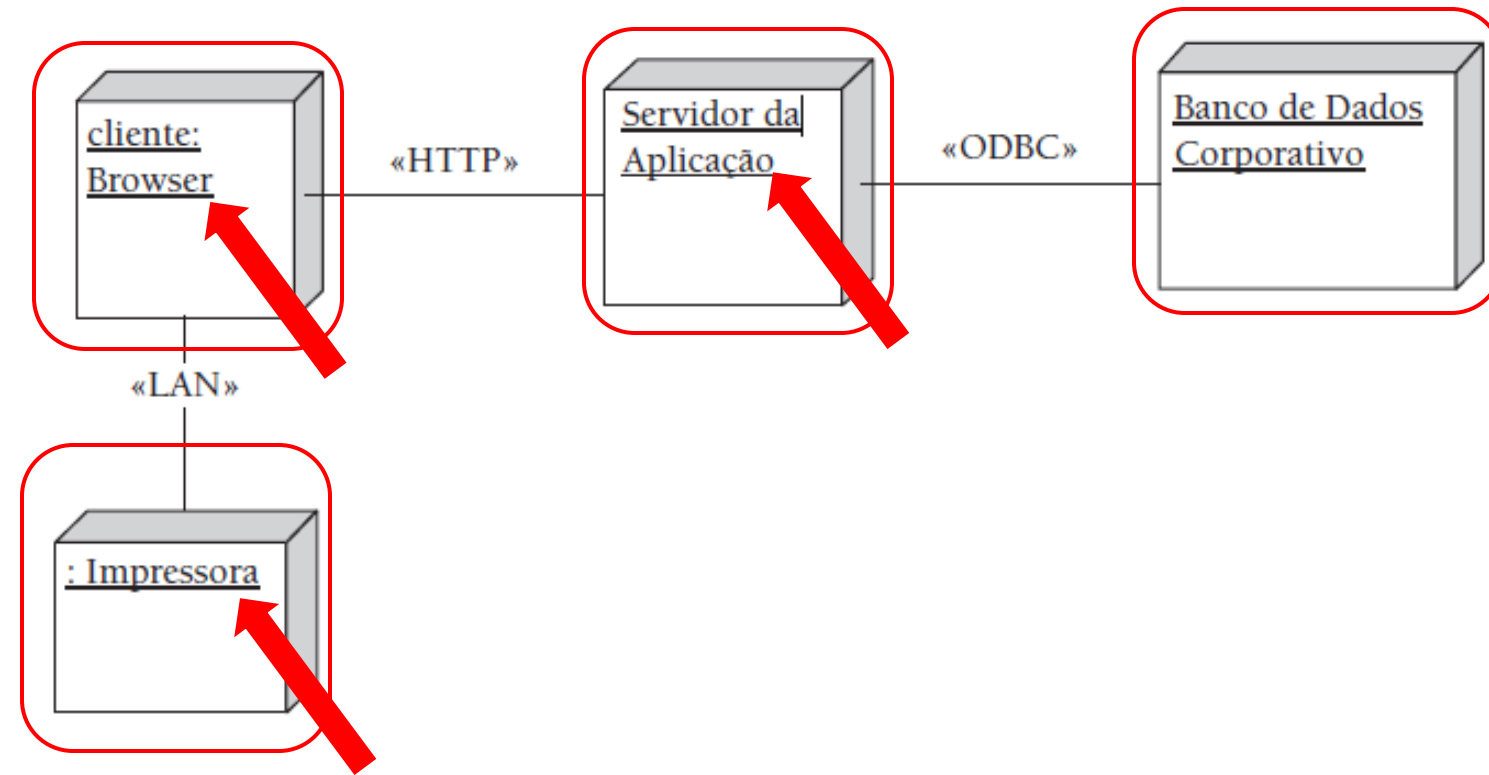
# Diagrama de Implantação - Nós



Os nós são representados por um cubo.

**O nome e o tipo do nó são definidos no interior do cubo.**

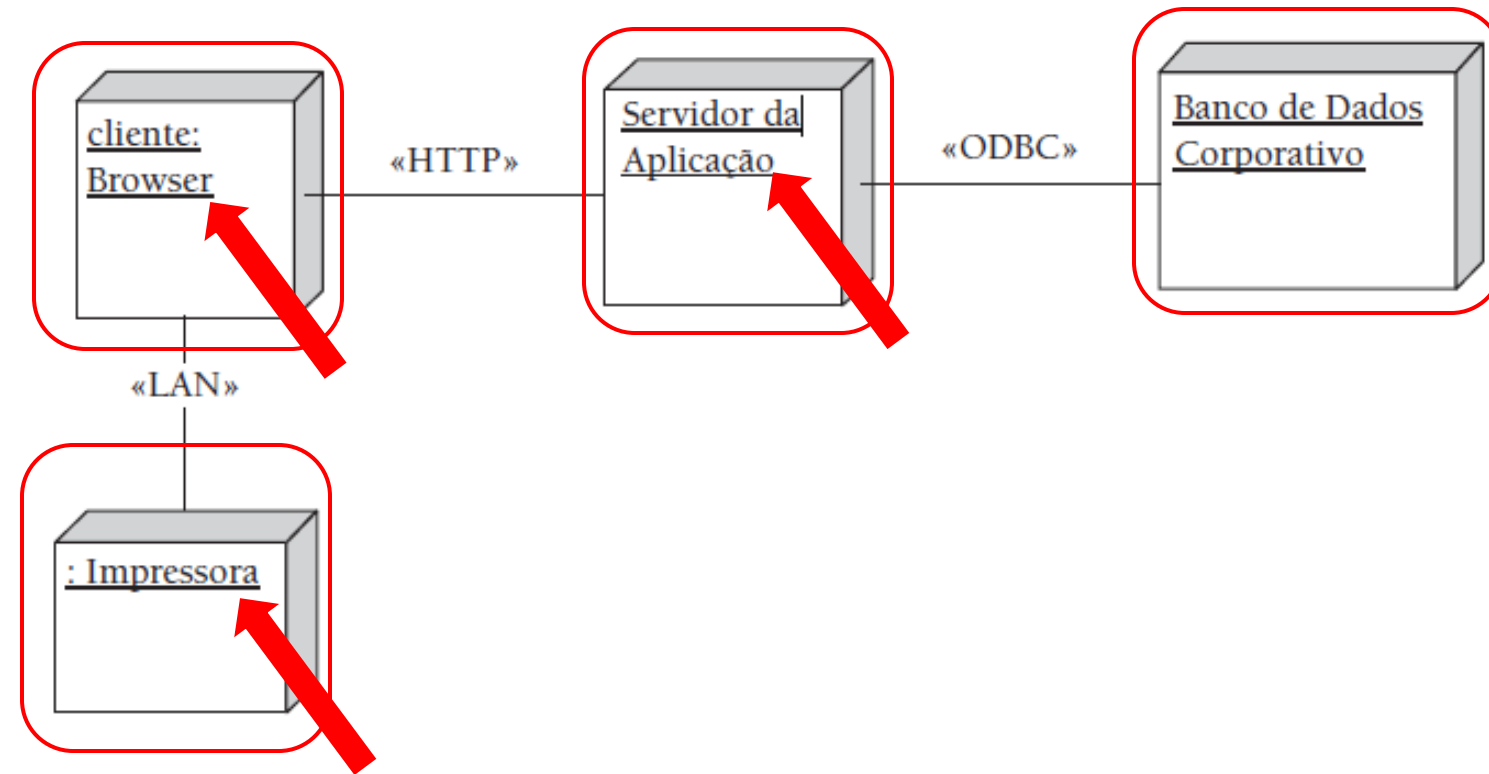
# Diagrama de Implantação - Nós



O nome e o tipo são sublinhados e separados por um sinal de dois pontos

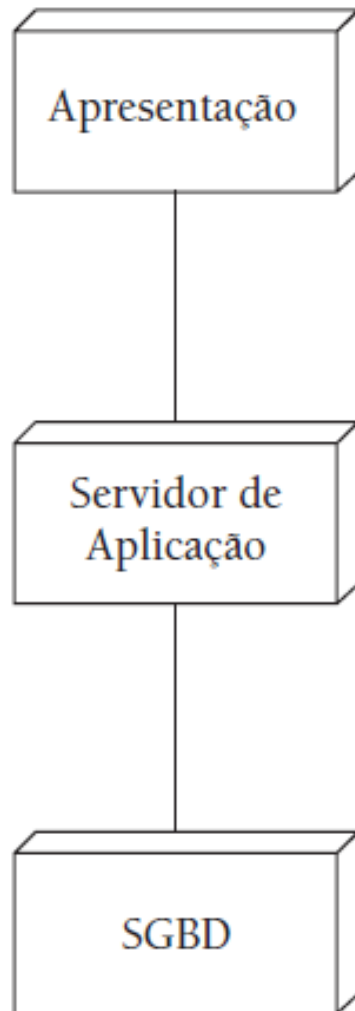
Nome e tipo são opcionais.

# Diagrama de Implantação - Nós



Pode-se representar outras informações importantes a respeito de um nó, como: **fornecedor, sistema operacional, localização ou qualquer coisa que você desejar.**

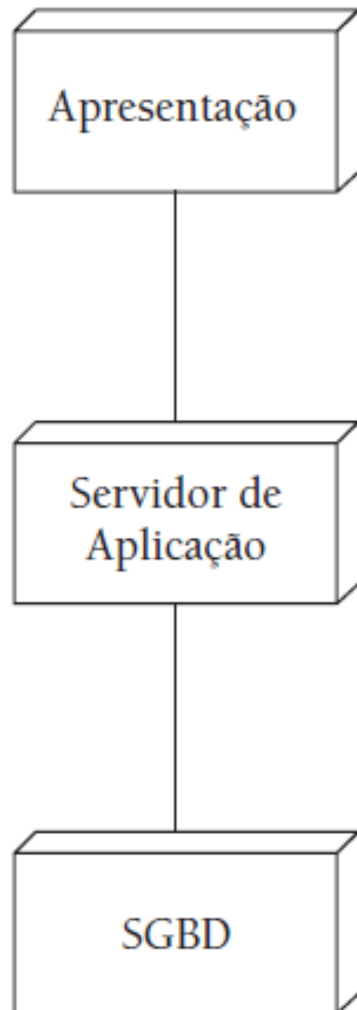
# Diagrama de Implantação - Nós



Podemos ter representações mais simples (**algo similar a uma divisão em camadas sem especificar onde cada uma delas será executada**)



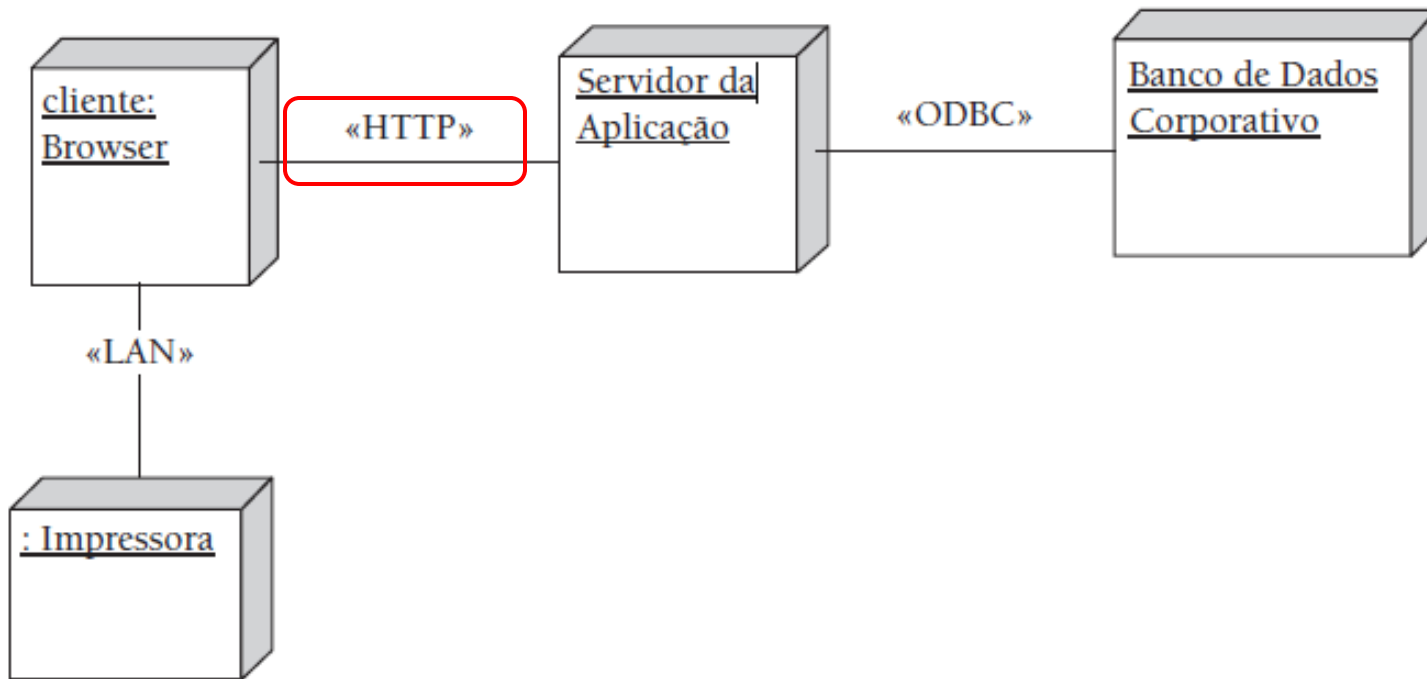
# Diagrama de Implantação - Nós



## Implantação Física em Arquitetura Cliente-Servidor:

- A camada de apresentação é alocada na máquina do usuário:
  - Normalmente responsável pela interface gráfica com o usuário
- O servidor (camadas lógicas inferiores) é executado em outra máquina
  - Possui maior capacidade de processamento
  - Pode atender a diversos clientes simultaneamente
- Outras configurações de alocação entre cliente e servidor também são possíveis

# Diagrama de Implantação - Conexões



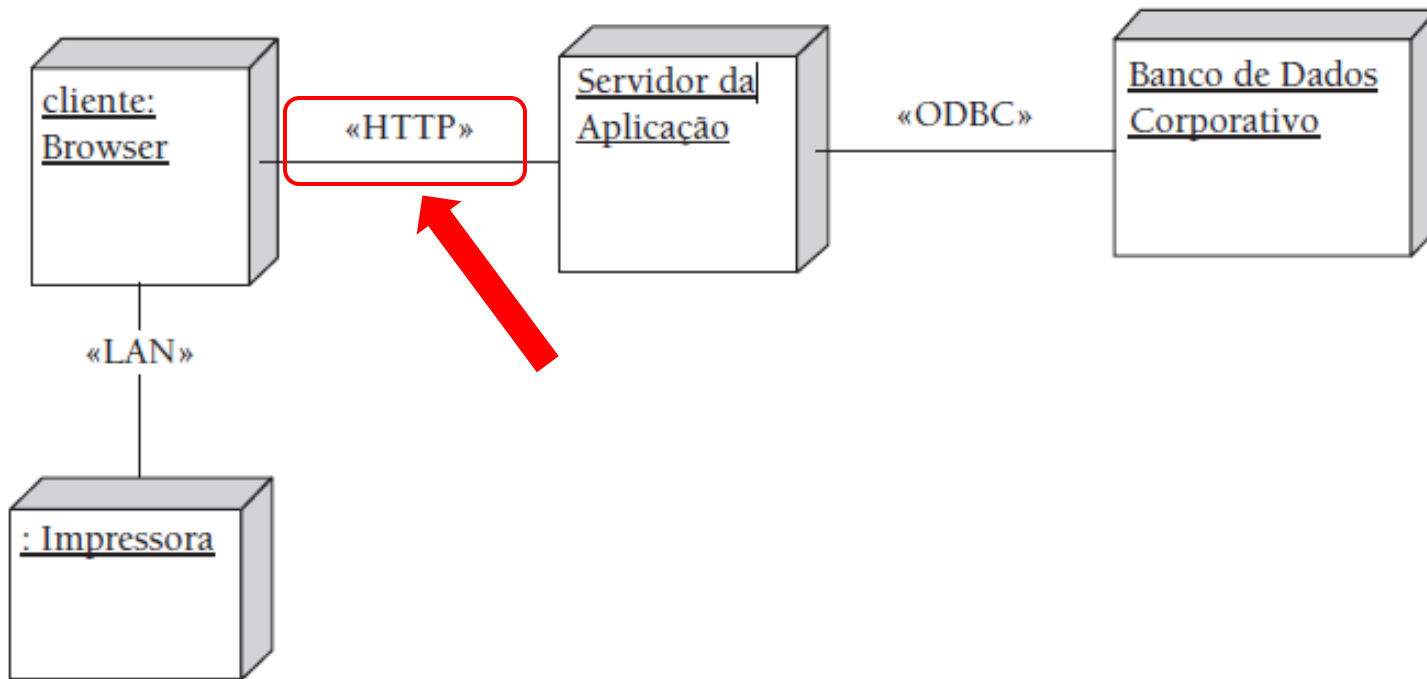
Os nós são ligados por meio de conexões.

# Diagrama de Implantação - Conexões

As conexões mostram os mecanismos de comunicação entre os nós:

- Meios físicos (cabo, fibra ótica, etc);
- Protocolos de comunicação (TCP/IP, HTTP, etc).

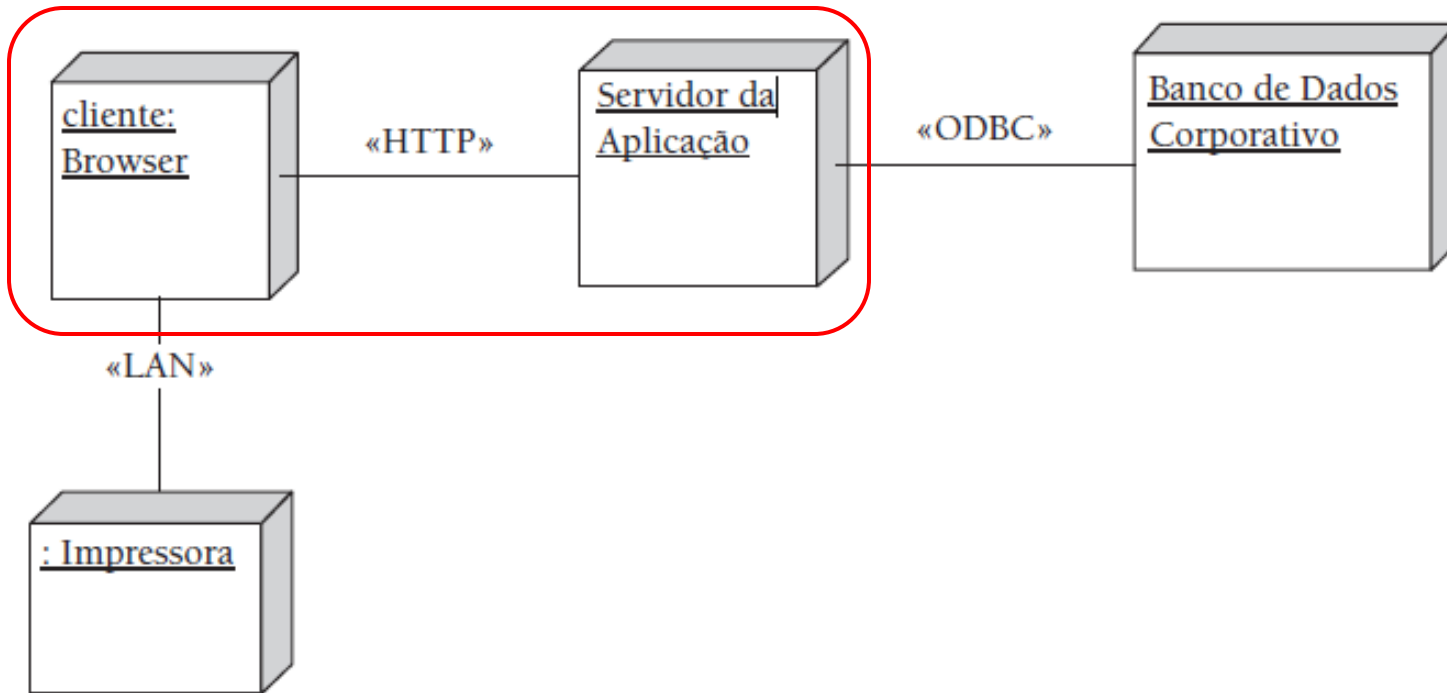
# Diagrama de Implantação - Conexões



Utilizamos uma linha para representar a conexão

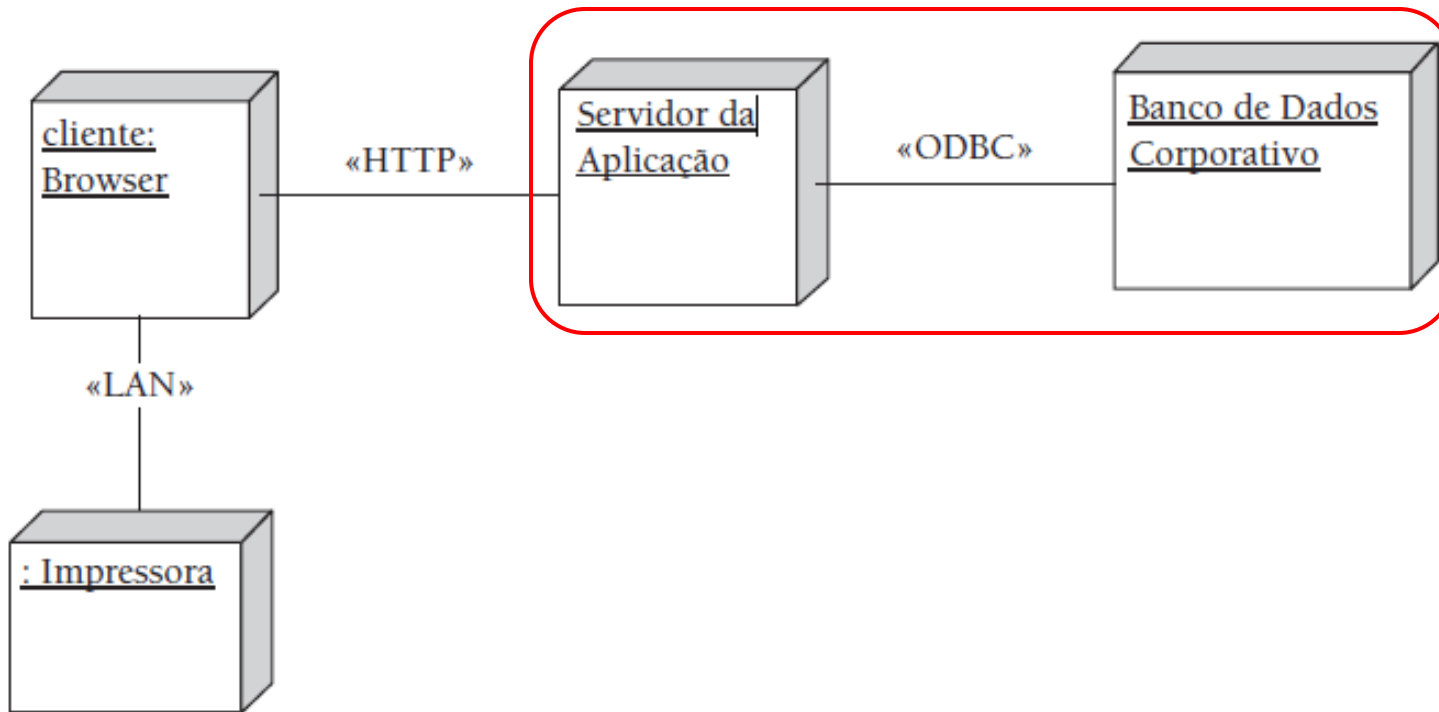
A conexão pode ser estereotipada

# Diagrama de Implantação - Conexões

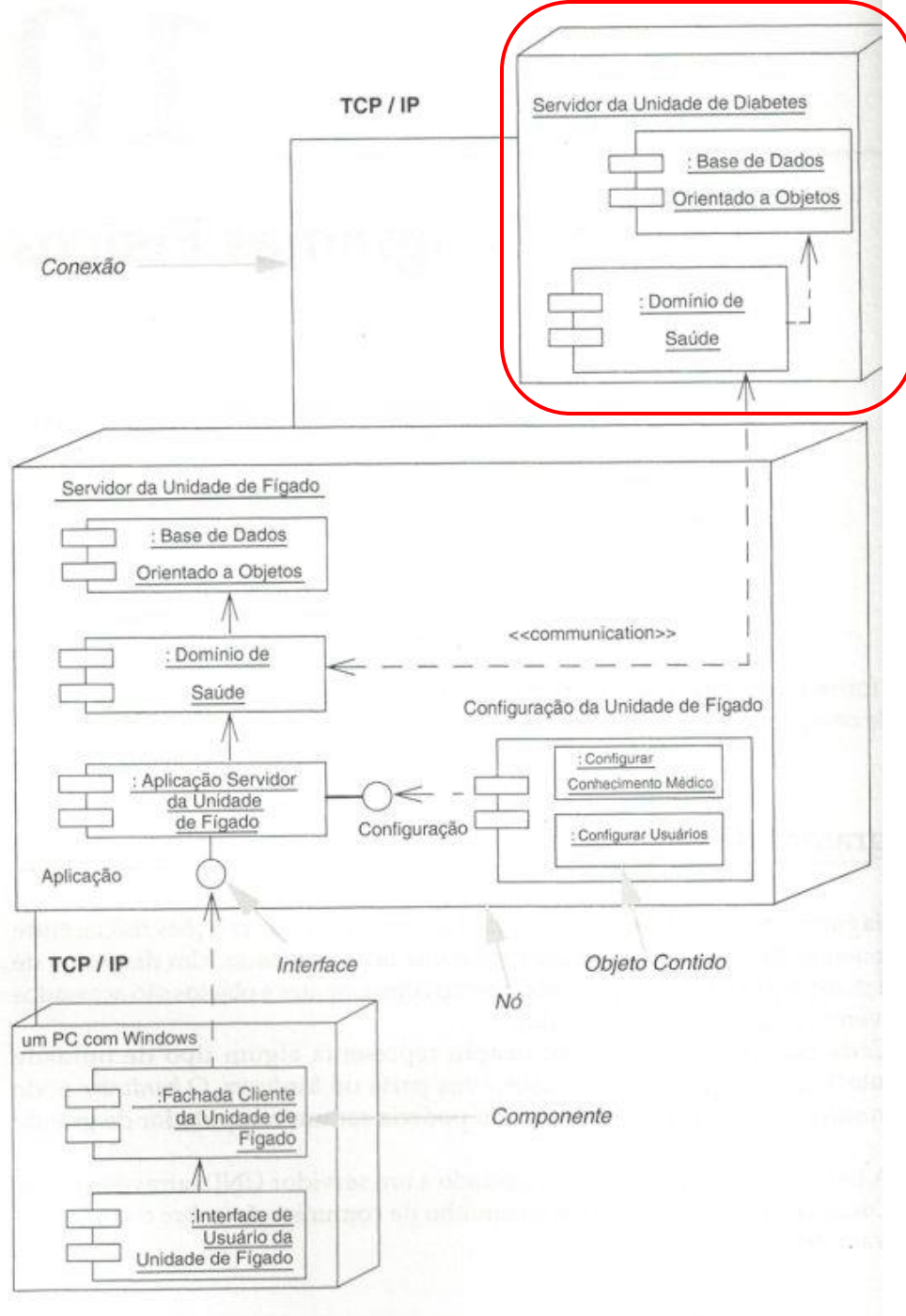


**Exemplo:** Computadores pessoais se comunicação através do protocolo HTTP ao servidor de aplicação

# Diagrama de Implantação - Conexões

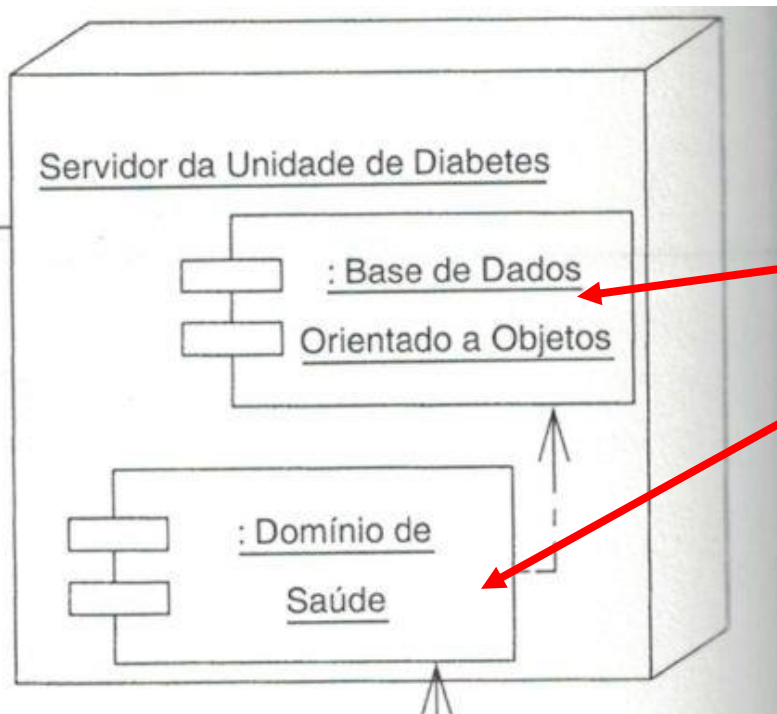


**Exemplo:** A aplicação no servidor se comunica com o SGBD através da API ODBC.



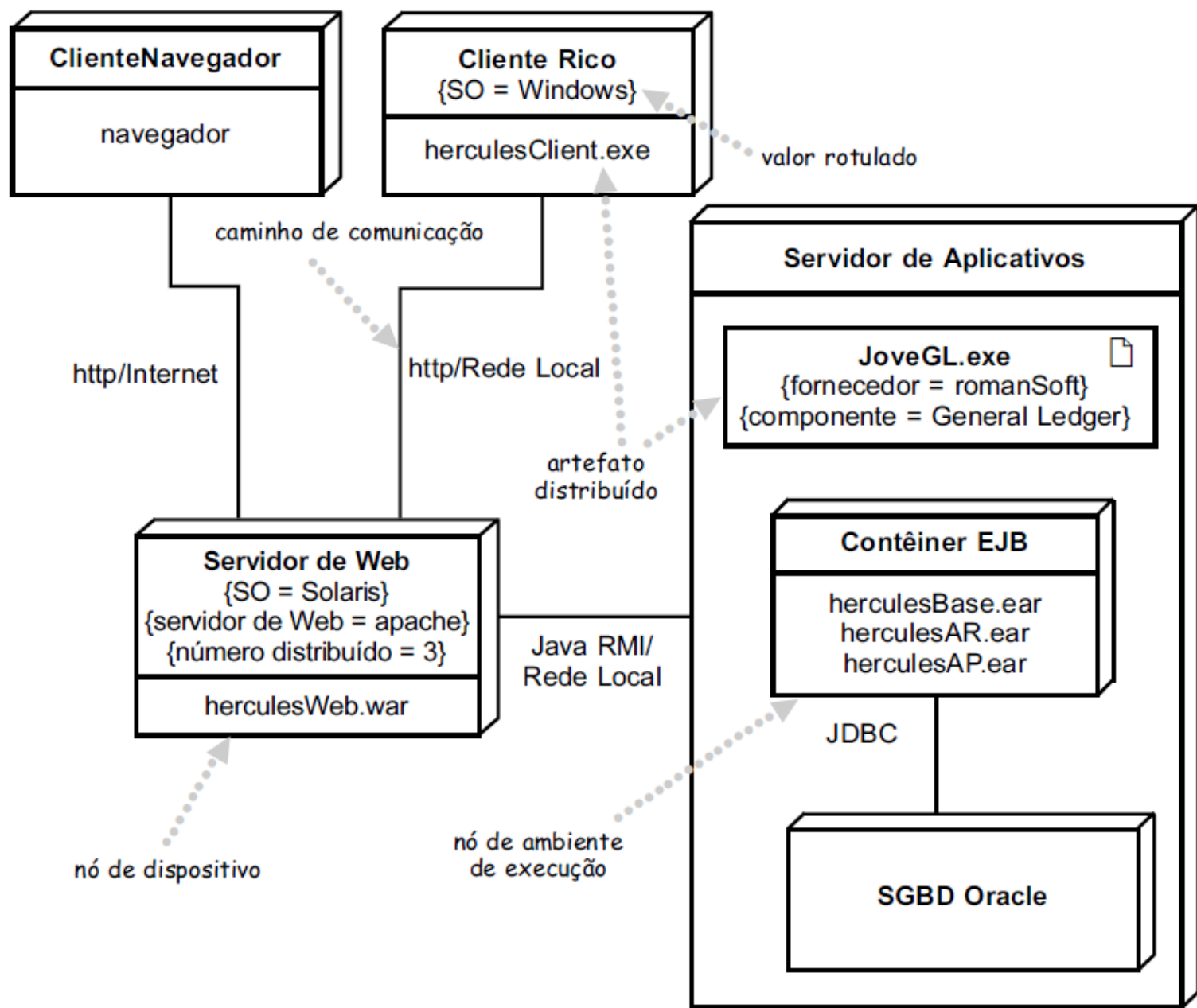
**Nó do diagrama de implantação**

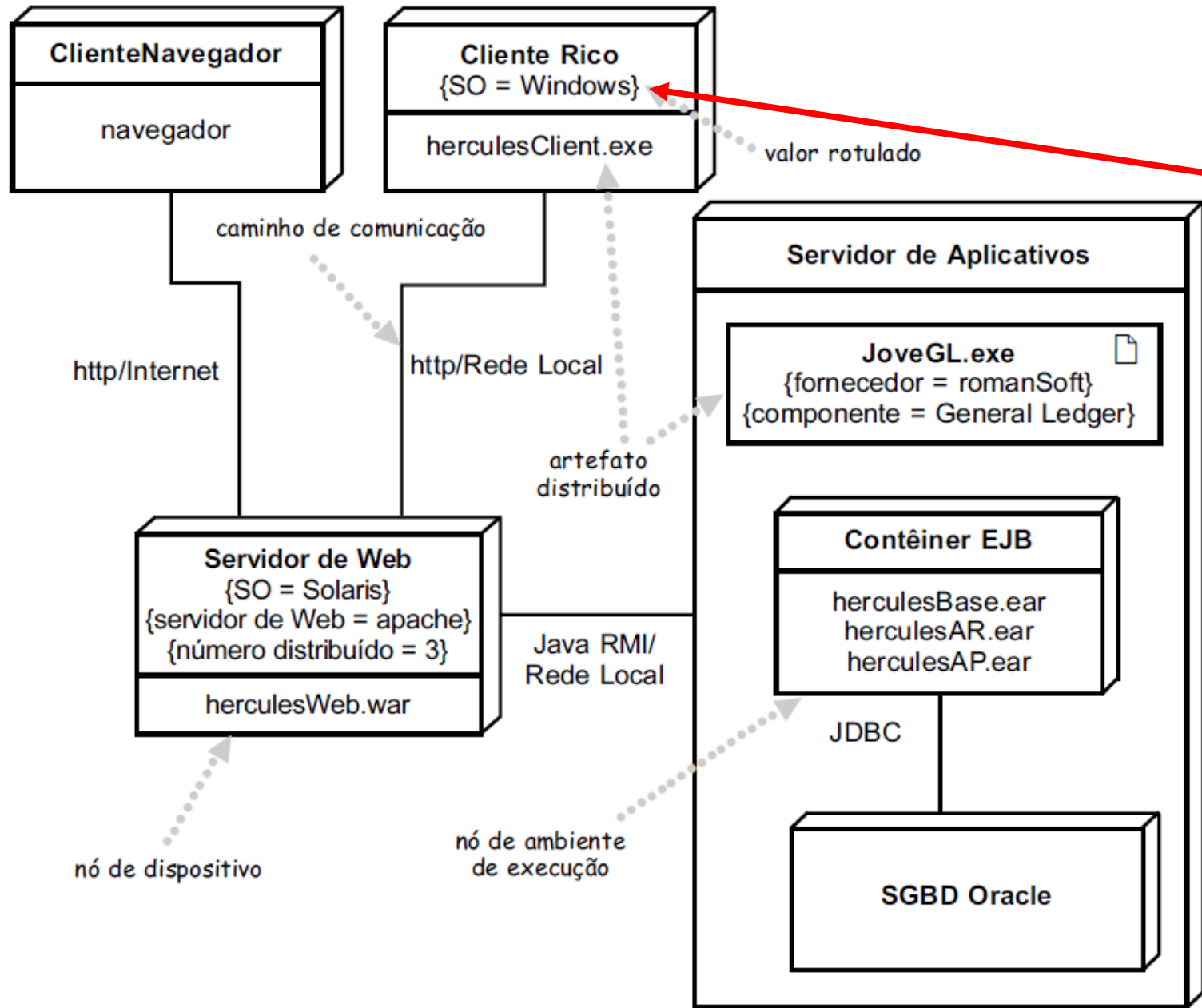
TCP / IP



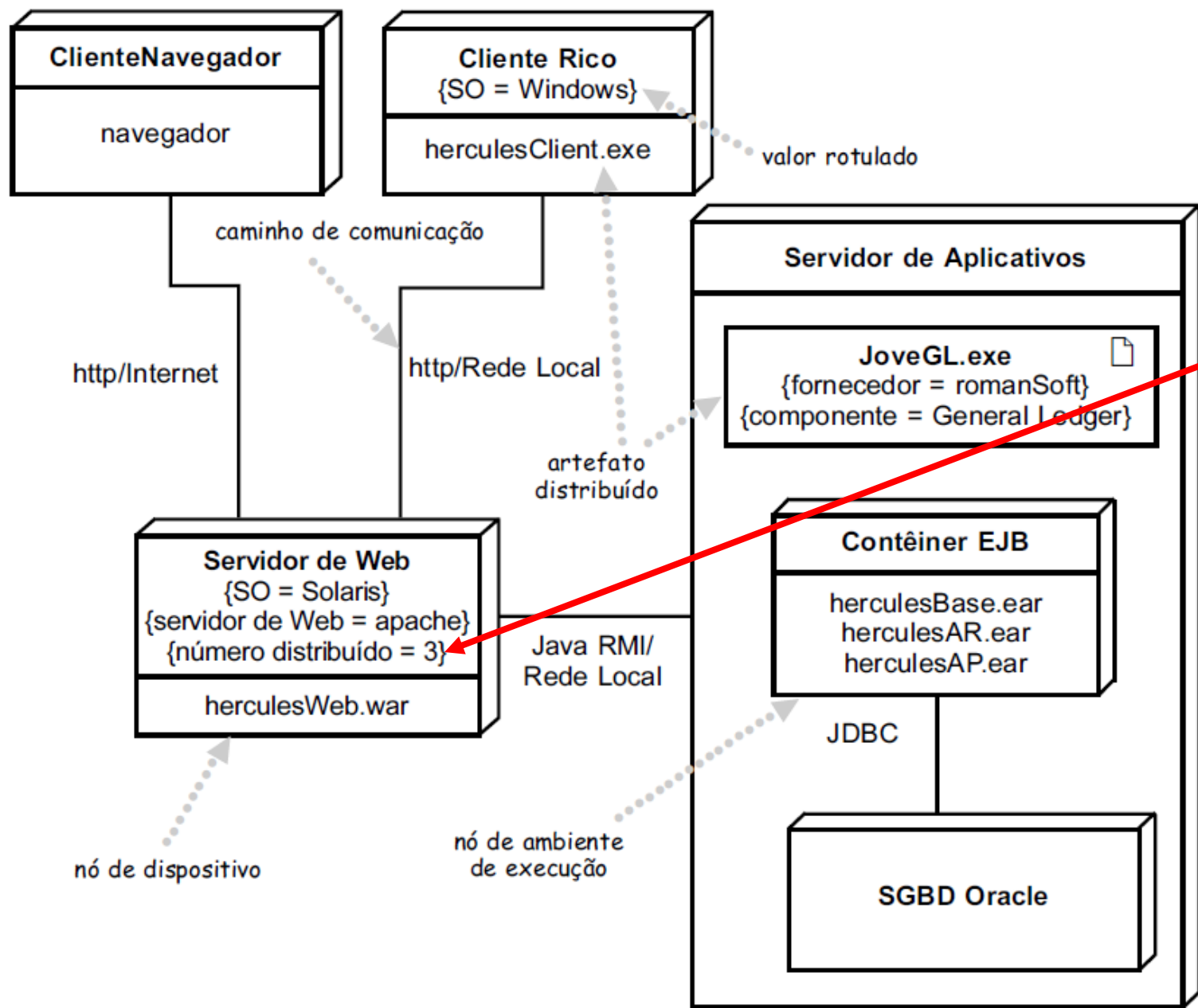
**Componentes**



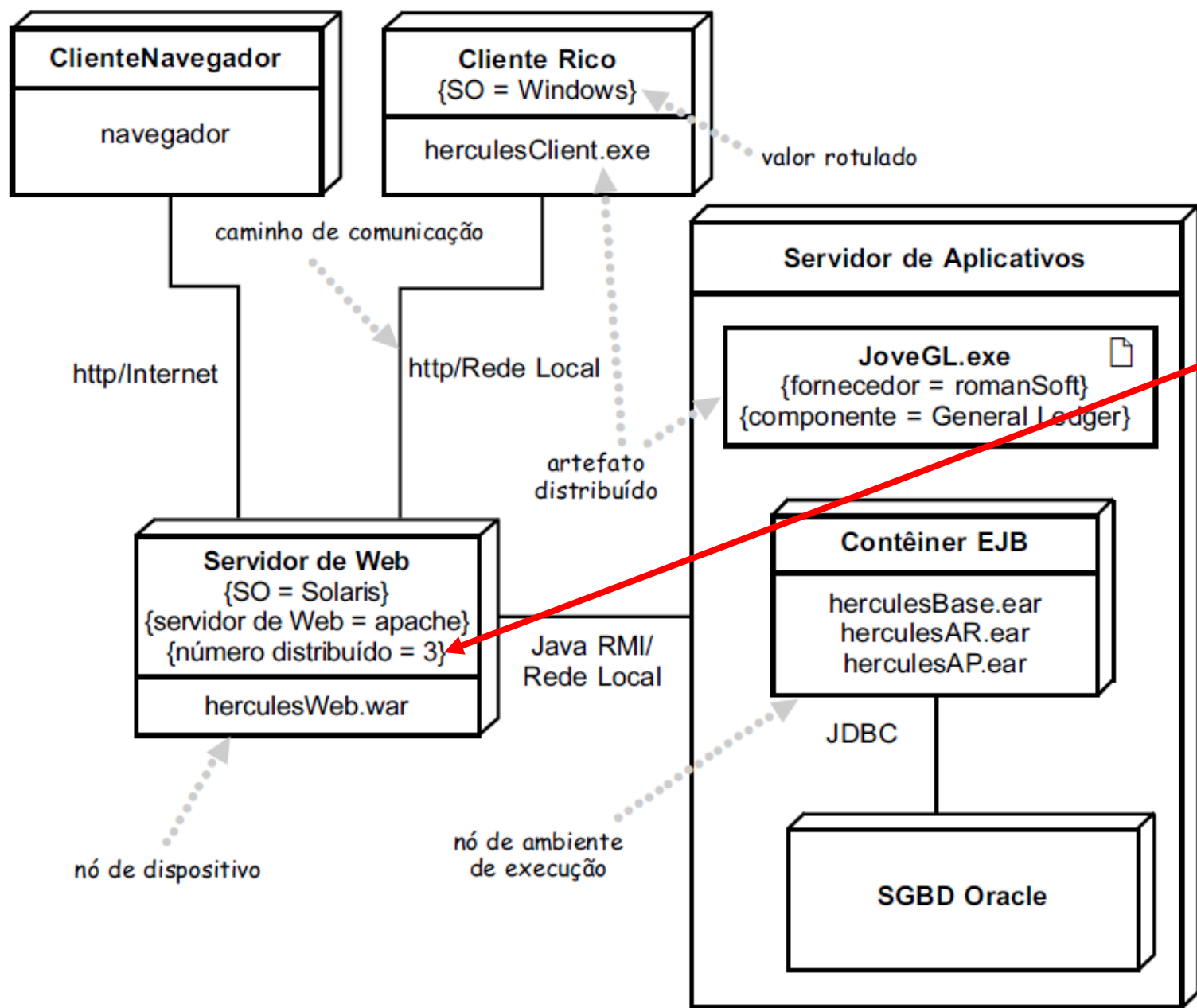




Pode-se adicionar rótulos de sistemas operacionais/fornecedores (e outras informações relevantes)



Em alguns casos, vários nós físicos podem executar a mesma tarefa lógica. Isso pode ser representado no diagrama de implantação por múltiplos nós (caixas) ou indicando a quantidade diretamente como um valor afixado ao nó.



Em alguns casos, vários nós físicos podem executar a mesma tarefa lógica. Isso pode ser representado no diagrama de implantação por múltiplos nós (caixas) ou indicando a quantidade diretamente como um valor afixado ao nó.

**Não existe nenhum rótulo padrão para isso!**

# Comentários Finais

- Não deixe que a simplicidade faça pensar que os diagramas de instalação não devem ser usados.
- Eles são muito úteis para mostrar o que é instalado e onde, portanto, qualquer instalação mais complicada pode fazer bom uso deles.



# Exercícios

---

# Dicas para Dividir um Sistema em Pacotes

- **Pacotes (Packages):** agrupam elementos relacionados, ajudando na organização e modularização do sistema.
  - Agrupe **classes e elementos com forte coesão** (funcionalidade relacionada)
  - Mantenha **baixo acoplamento** entre pacotes diferentes
  - Nomeie pacotes de forma **clara e descritiva** (ex.: cliente, pagamento, relatorio)
  - Use pacotes para refletir **camadas da arquitetura** (ex.: apresentação, negócio, persistência)
  - Evite pacotes excessivamente grandes ou com responsabilidades muito variadas

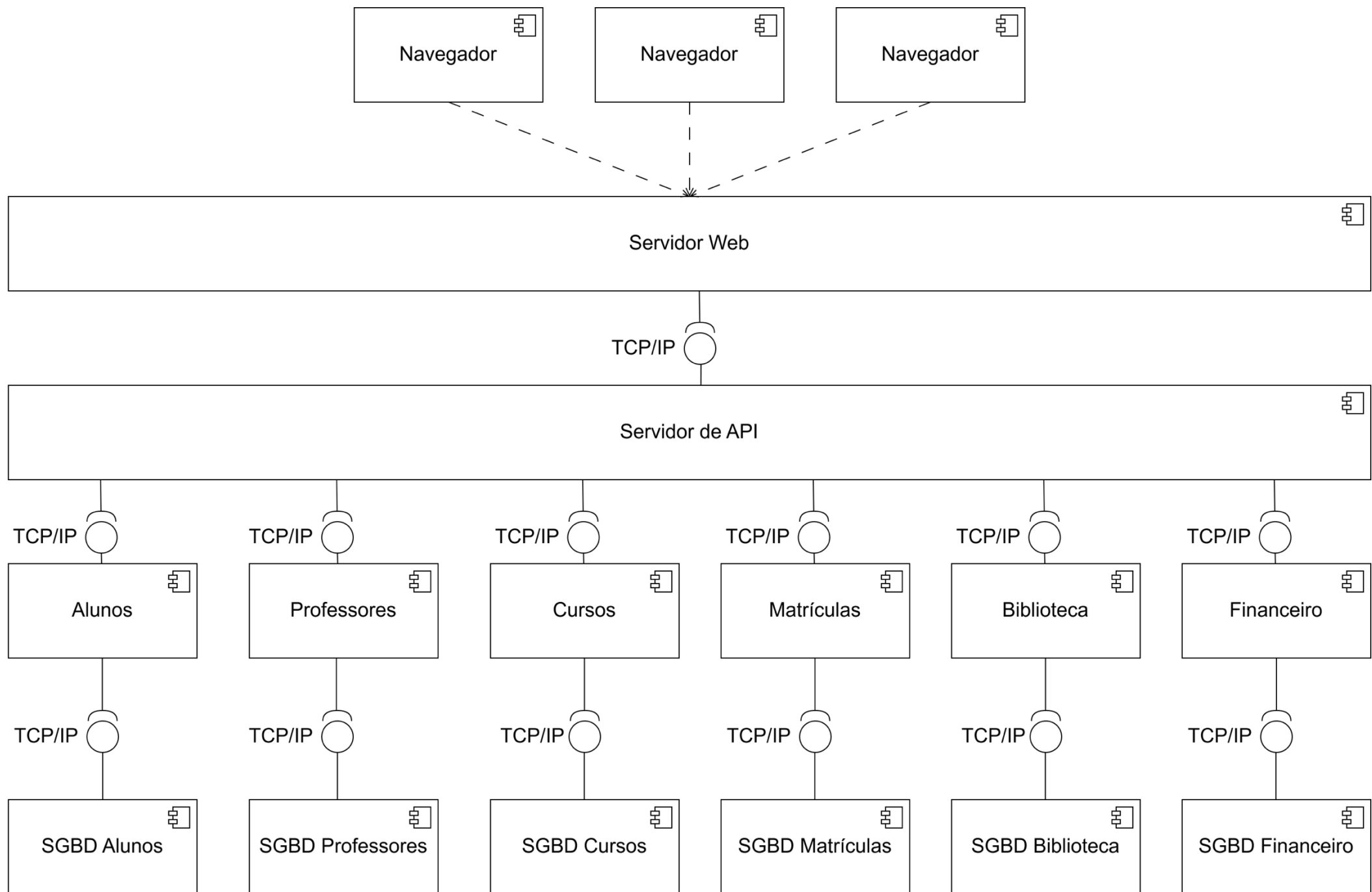
# Dicas para Dividir um Sistema em Componentes

- **Componentes (Components):** representam módulos de software independentes e reutilizáveis.
  - Cada componente deve ter uma **responsabilidade clara**.
  - Relacione componentes com **interfaces bem definidas**.
  - Prefira componentes **fracamente acoplados** (mínima dependência).
  - Use componentes para **representar serviços, módulos ou subsistemas** (ex.: Carrinho de Compras, Autenticação, Processador de Pagamentos)
  - Lembre-se: no diagrama de implantação, os componentes serão **implantados em nós físicos**.



# Exercício 1

- Desenvolva o **diagrama de componentes** de um sistema de **gestão acadêmica**, cuja arquitetura também é baseada no estilo de microsserviços.
  - O sistema possui os seguintes seis serviços:
    - **Alunos, Professores, Cursos, Matrículas, Biblioteca e Financeiro.**
- Cada serviço:
  - Está hospedado em um **servidor dedicado**
  - Possui um **SGBD exclusivo**, também implantado em um servidor separado
- Além disso:
  - Todos os servidores de serviços se comunicam com um **servidor de API**
  - O **servidor de API** se comunica com um **servidor Web**, responsável por gerar as páginas acessadas pelos usuários finais via browser
  - Todas as comunicações utilizam o protocolo **TCP/IP**



# Exercício 2

- Desenvolva o **diagrama de implantação** para um sistema de **gestão de biblioteca universitária**, detalhando seus componentes.
  - Neste sistema, existe um **Servidor de Aplicação** que inclui três módulos: **Empréstimos**, **Devoluções** e **Catálogo**.
  - Um segundo servidor, denominado **Banco de Dados da Biblioteca**, é responsável pela gerência dos dados. Este servidor inclui dois módulos: **SGBD** e **Persistência**.
  - O **Servidor de Aplicação** se comunica com o **Banco de Dados da Biblioteca** via **JDBC**, um padrão para comunicação com bancos de dados em Java.
  - Existem **computadores pessoais (browser)** que se comunicam com o **Servidor de Aplicação** através do protocolo **HTTP**. Nestes clientes, há um módulo de **Interface de Usuário da Biblioteca**.
  - Há também um **terminal de autoatendimento** conectado à rede LAN, que permite aos usuários consultar o catálogo e renovar empréstimos.
  - O módulo de **Interface de Usuário da Biblioteca** utiliza serviços providos pelos módulos **Empréstimos**, **Devoluções** e **Catálogo**.
  - O módulo de **Persistência** provê uma interface com serviços que são utilizados pelos módulos **Empréstimos**, **Devoluções** e **Catálogo**.
  - Por fim, o módulo de **Persistência** faz uso de serviços fornecidos pelo módulo **SGBD**.

