



# Arquitetura Cliente-Servidor

**Sistemas Distribuídos e Mobile**

Prof. Gustavo Torres Custodio  
gustavo.custodio@anhembi.br

# Agenda

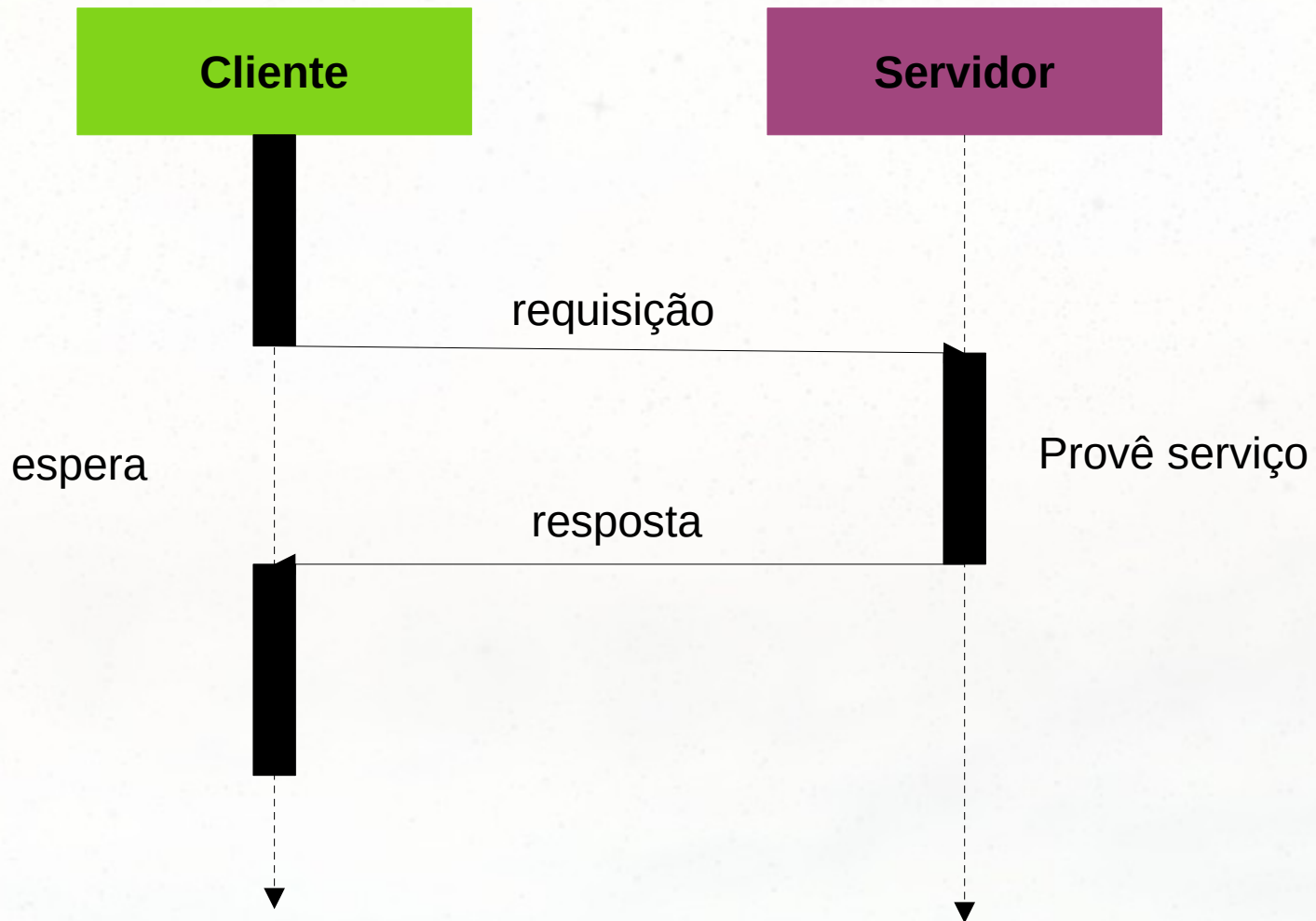
- Introdução
- Comunicação Interprocesso
- Troca de Mensagens
- Modelo Cliente-Servidor
  - Vantagens e Desvantagens
  - Modelo em Camadas



# Introdução

- No modelo básico cliente-servidor:
  - os processos em um sistema distribuído são divididos em dois grupos:
    - Um servidor é um processo que implementa um serviço específico, por exemplo, um serviço de sistema de arquivos ou um serviço de banco de dados.
    - Um cliente é um processo que solicita um serviço de um servidor enviando-lhe um pedido e, posteriormente, aguardando a resposta do servidor.
- Essa interação cliente-servidor, também conhecido como comportamento de solicitação-resposta (*request-reply*)

# Introdução



# Comunicação Interprocessos: Modelo Cliente-Servidor

- Servidores são basicamente projetados para:
  - Receber mensagem de requisição
  - Processar as requisições
  - Enviar as respostas
- Clientes são basicamente projetados para:
  - Interagir com o usuário final
  - Solicitar os serviços dos servidores
  - Aguardar respostas
  - Interagir novamente com o usuário, exibindo as respostas de forma clara.





# Introdução: Comunicação em Sistemas Distribuídos

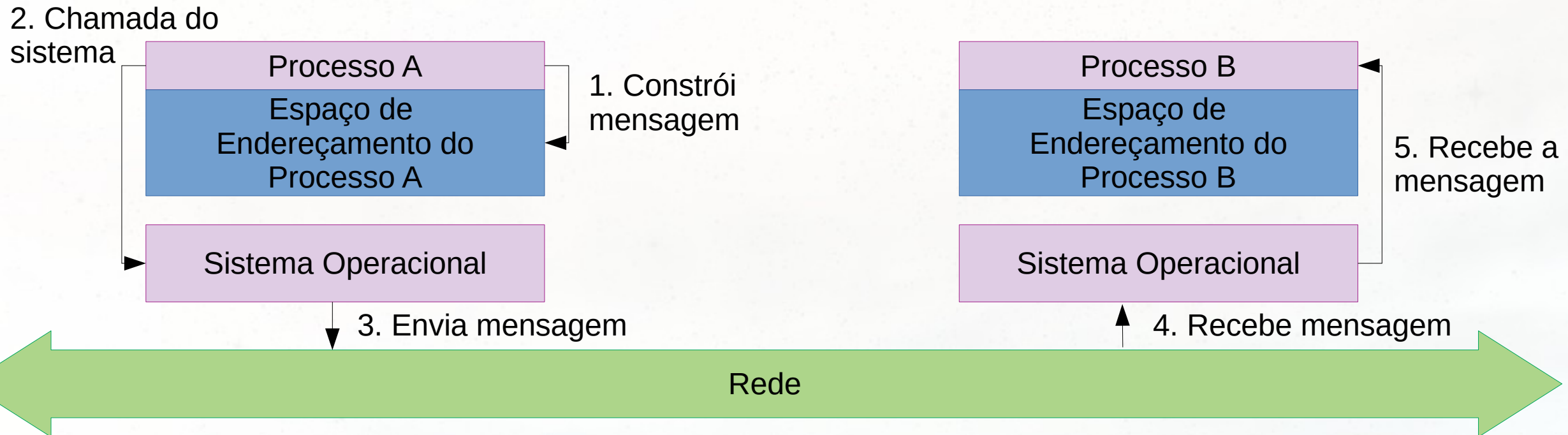
- A diferença mais importante entre Sistemas Distribuídos e Sistemas Centralizados é a Comunicação Interprocesso (Interprocess Communication – IPC);
  - Sistemas Centralizados
    - memória compartilhada;
  - Sistemas Distribuídos
    - não existe memória compartilhada;
    - Comunicação Interprocesso reformulada
    - Permitir que os processos se comuniquem para troca de dados ou acessos a recursos ou serviços em processadores remotos.

# Introdução: Comunicação em Sistemas Distribuídos

- **Em Sistemas Centralizados**, uma região da memória é compartilhada por dois ou mais processos.
- A região da memória é ligada a diferentes processos.
  - Uma alteração feita por um processo é vista por todos.
- Regiões de memória compartilhadas são gerenciadas no núcleo (kernel).

# Introdução: Comunicação Interprocessos

- Em Sistemas Distribuídos, a comunicação é feita baseada na Troca de Mensagens (Message Passing);



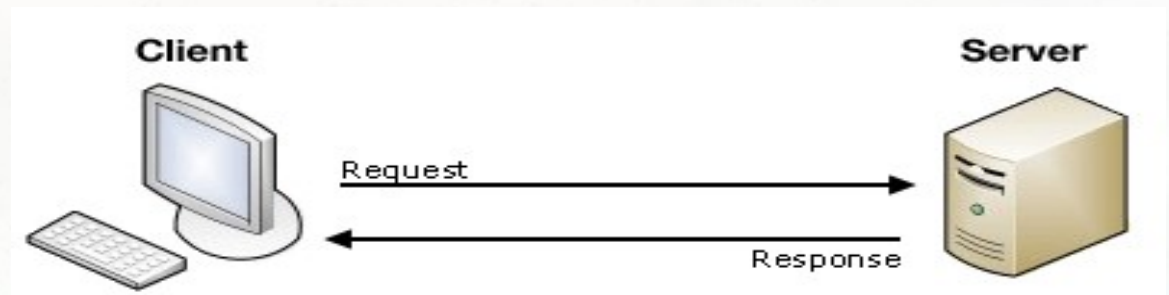


# Comunicação Interprocessos: Modelo Cliente-Servidor

- Vantagens da comunicação modelo cliente-servidor:
  - Simplicidade;
  - Não tem necessidade de estabelecer e finalizar a conexão explicitamente;
  - A mensagem de resposta de um servidor pode funcionar como confirmação de entrega de uma requisição;
  - Eficiência.

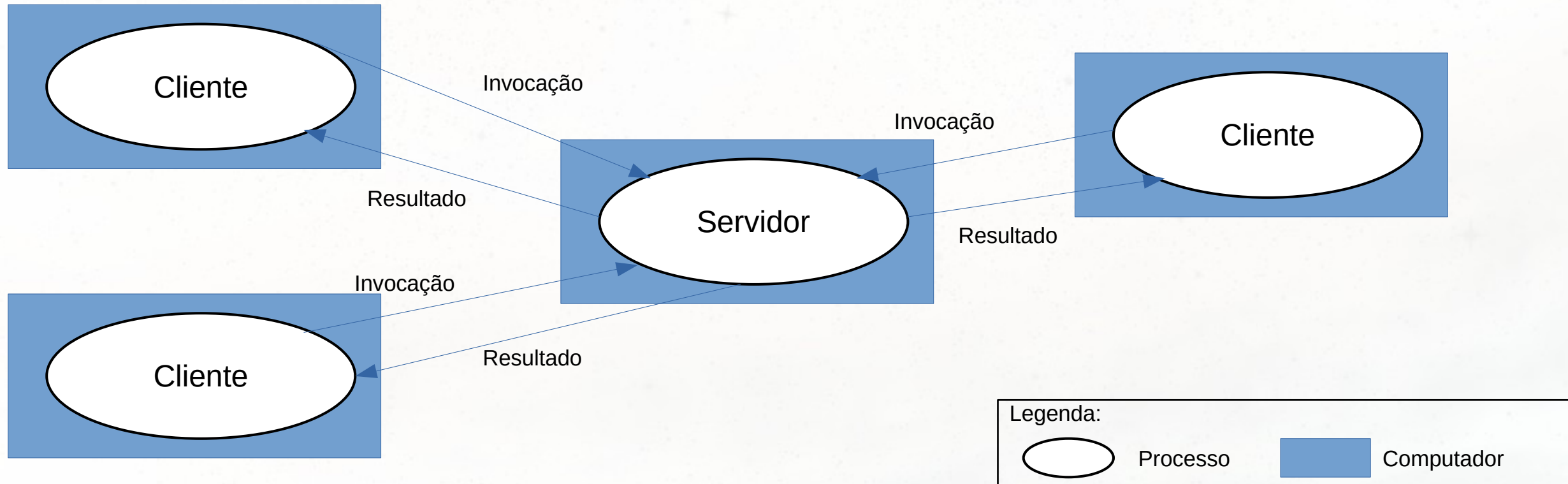
# Comunicação Interprocessos: Modelo Cliente-Servidor

- Uma máquina pode executar:
  - Um único ou múltiplos processos clientes;
  - Um único ou múltiplos processos servidores;
  - Ou uma combinação das alternativas anteriores.



# Comunicação Interprocessos: Modelo Cliente-Servidor

- Processos Clientes acionando individualmente Processos Servidores:

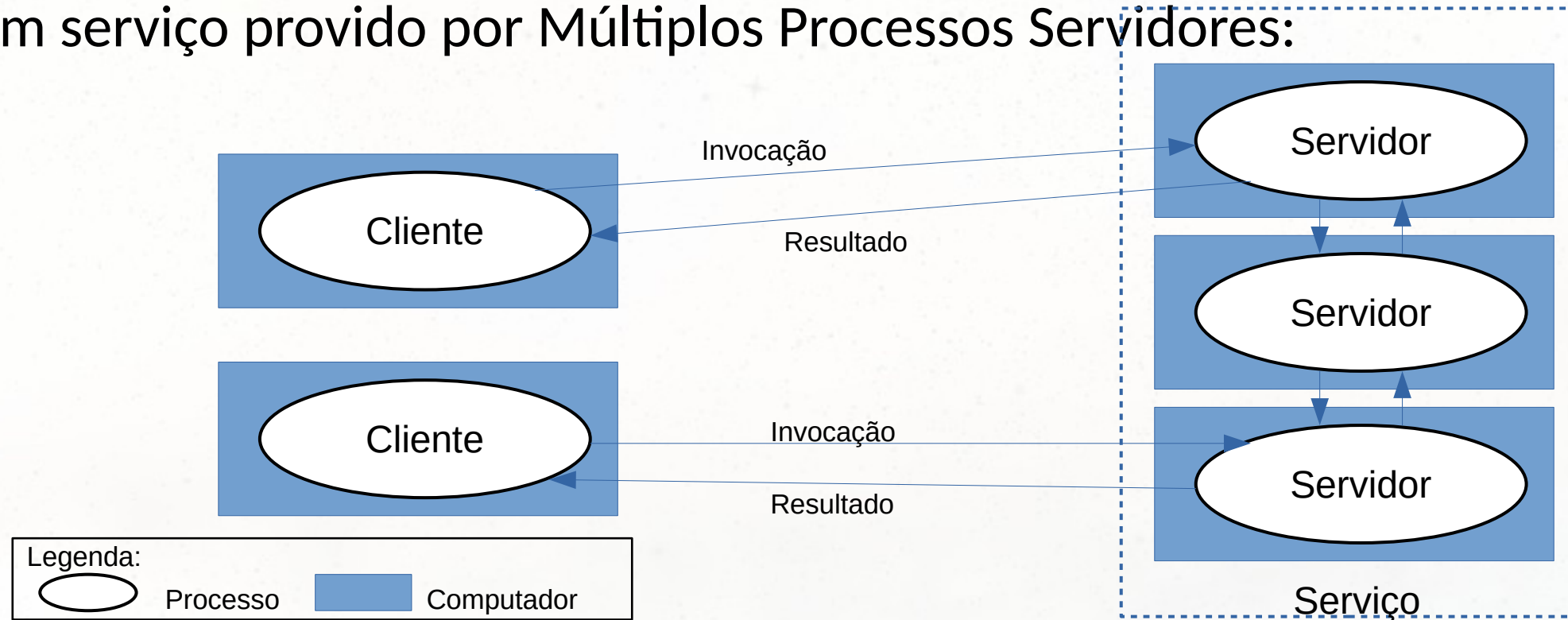


- Único ponto de falha



# Comunicação Interprocessos: Modelo Cliente-Servidor

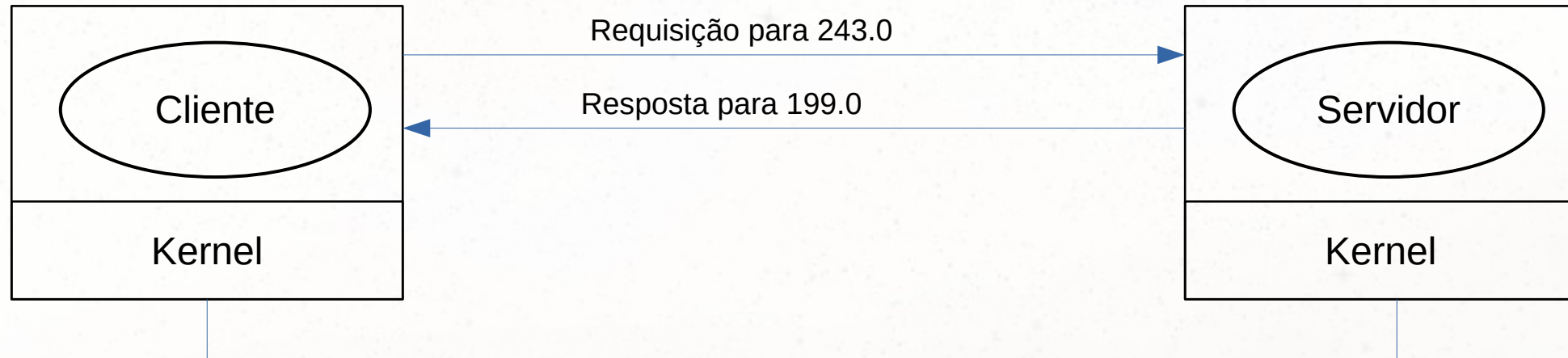
- Um serviço provido por Múltiplos Processos Servidores:



- Aplicações: processamento paralelo.
- Vantagens: menos dependência de um único servidor.

# Troca de Mensagem: Endereçamento

- Endereçamento Estático:



Rede

- Problemas...

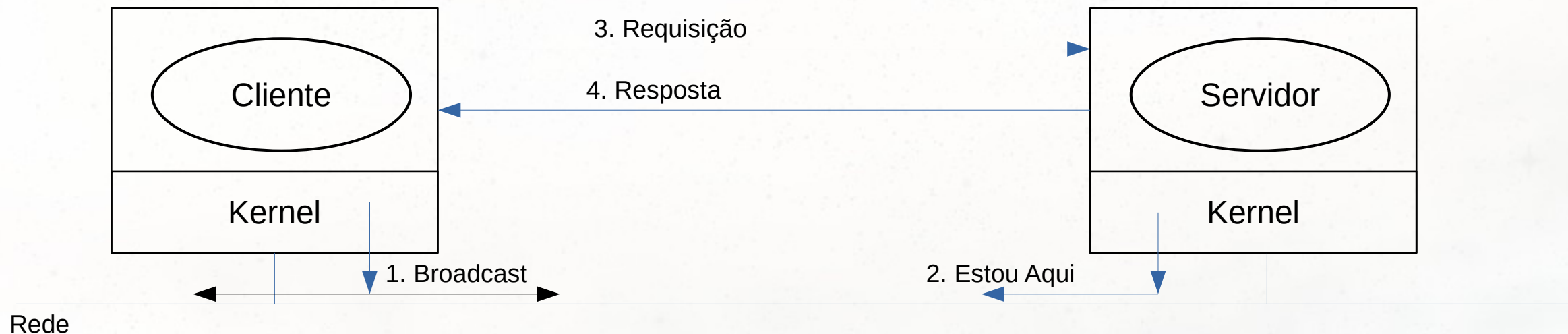
- Não é transparente
  - se um servidor não estiver disponível teremos recompilação para poder realizar o serviço em outro servidor.

- Vantagens...

- Não necessita de coordenação global
  - Cliente e Servidor comunicam-se diretamente

# ➤ Troca de Mensagem: Endereçamento

- Endereçamento Aleatório:

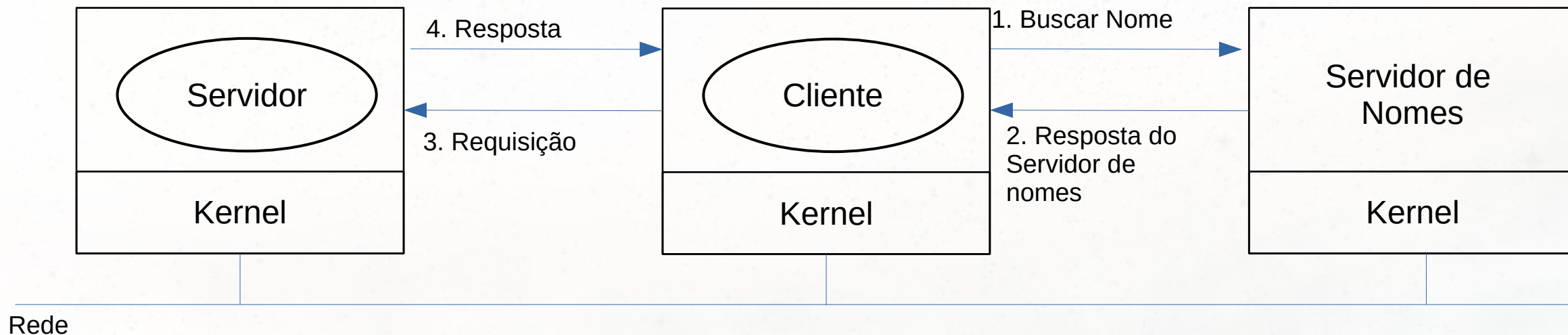


- Broadcast gera carga extra de comunicação no sistema



# ➤ Troca de Mensagem: Endereçamento

- Endereçamento usando um Servidor de Nomes:



- Problemas?

# Modelo Cliente-Servidor: Vantagens

- **Segurança**
  - Os dados são bem protegidos devido à sua arquitetura centralizada. Ele pode ser aplicado com controles de acesso de forma que apenas usuários autorizados tenham acesso.
- **Escalabilidade**
  - Sempre que os usuários precisam, eles podem aumentar o número de recursos, como clientes e servidores.
- **Gerenciamento**
  - A rede cliente-servidor pode rastrear e localizar facilmente os registros dos arquivos necessários.
    - Mais difícil fazer isso em uma rede *Peer-to-Peer*.
- **Acessibilidade**
  - Independentemente do local ou da plataforma, todos os clientes têm a oportunidade de fazer login no sistema.

# Modelo Cliente-Servidor: Desvantagens

- Sobrecarga de tráfego
  - Se muitos clientes fizerem solicitações do mesmo servidor, isso resultará em travamentos ou lentidão da conexão.
- Robustez
  - Caso o servidor principal sofra falha ou interferência, toda a rede será interrompida.
- Custo
  - Às vezes, configurar e manter o servidor é caro na rede cliente-servidor.
- Manutenção
  - Quando os servidores são implementados, geralmente funcionam sem parar. Isso significa que a devida atenção deve ser dada a cada servidor. Se houver algum problema, ele deve ser resolvido imediatamente, sem qualquer atraso.

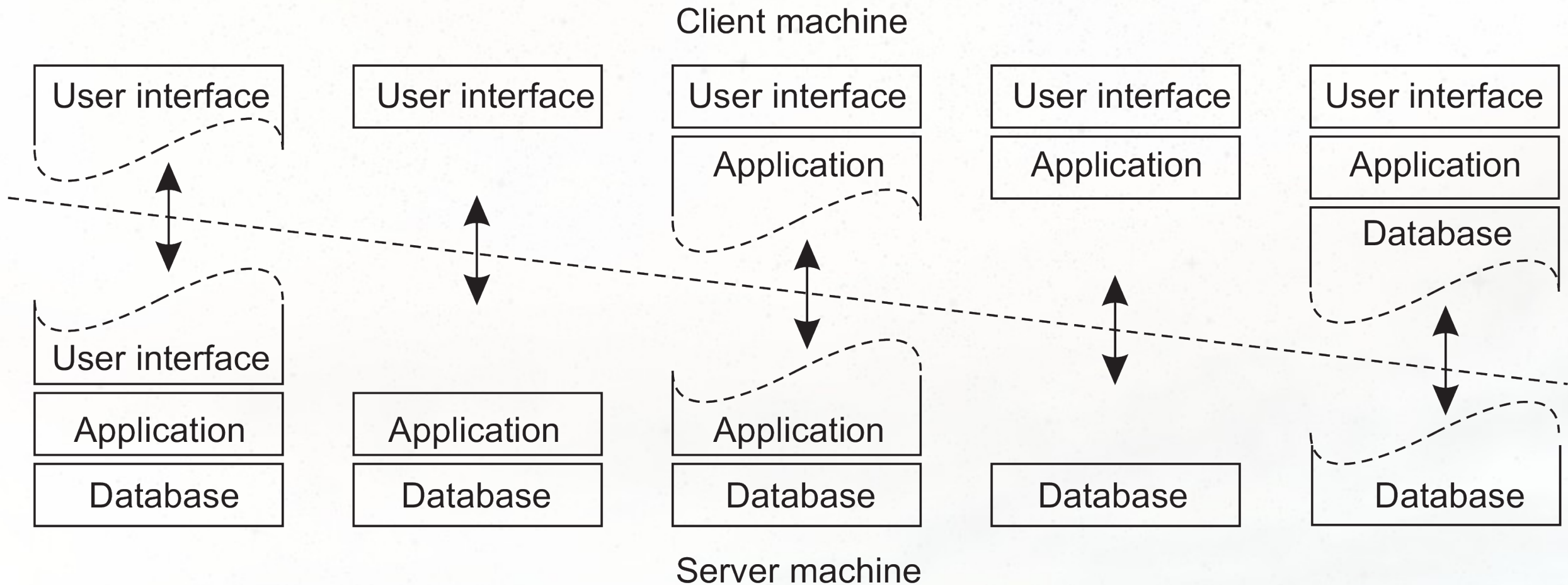


# Modelo Cliente-Servidor em Camadas

- Camada única (Single-tiered):
  - Terminal burro e mainframe
  - Camadas de apresentação, aplicação e banco de dados todos no mesmo servidor.
- Duas camadas (Two-tiered):
  - configuração de cliente/servidor único
  - Cliente apenas requisitar, servidor apenas fornece.
- Três camadas (Three-tiered):
  - cada camada em uma máquina separada
  - Servidor pode se comportar como cliente

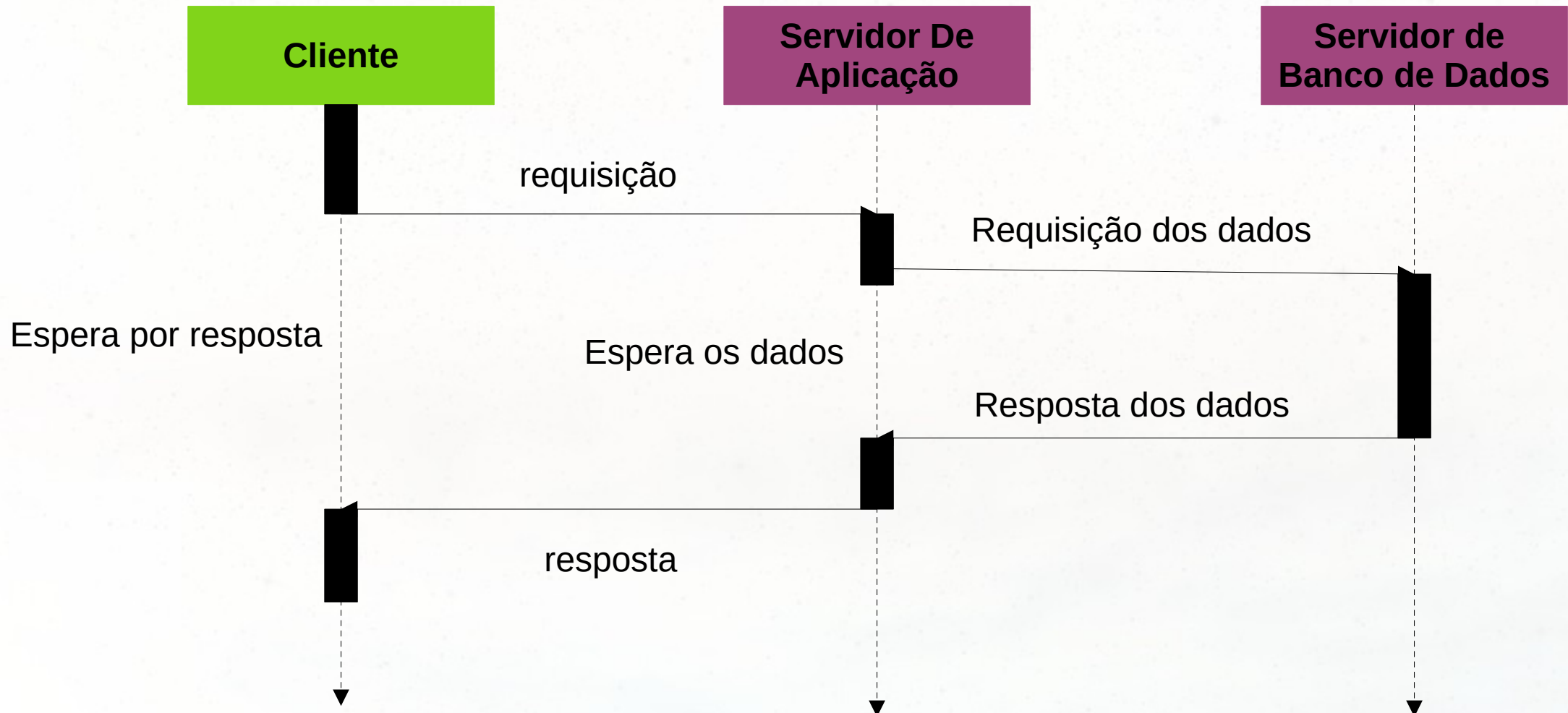
# Modelo Cliente-Servidor em Camadas

## Duas Camadas (Two-tiered)



# Modelo Cliente-Servidor em Camadas

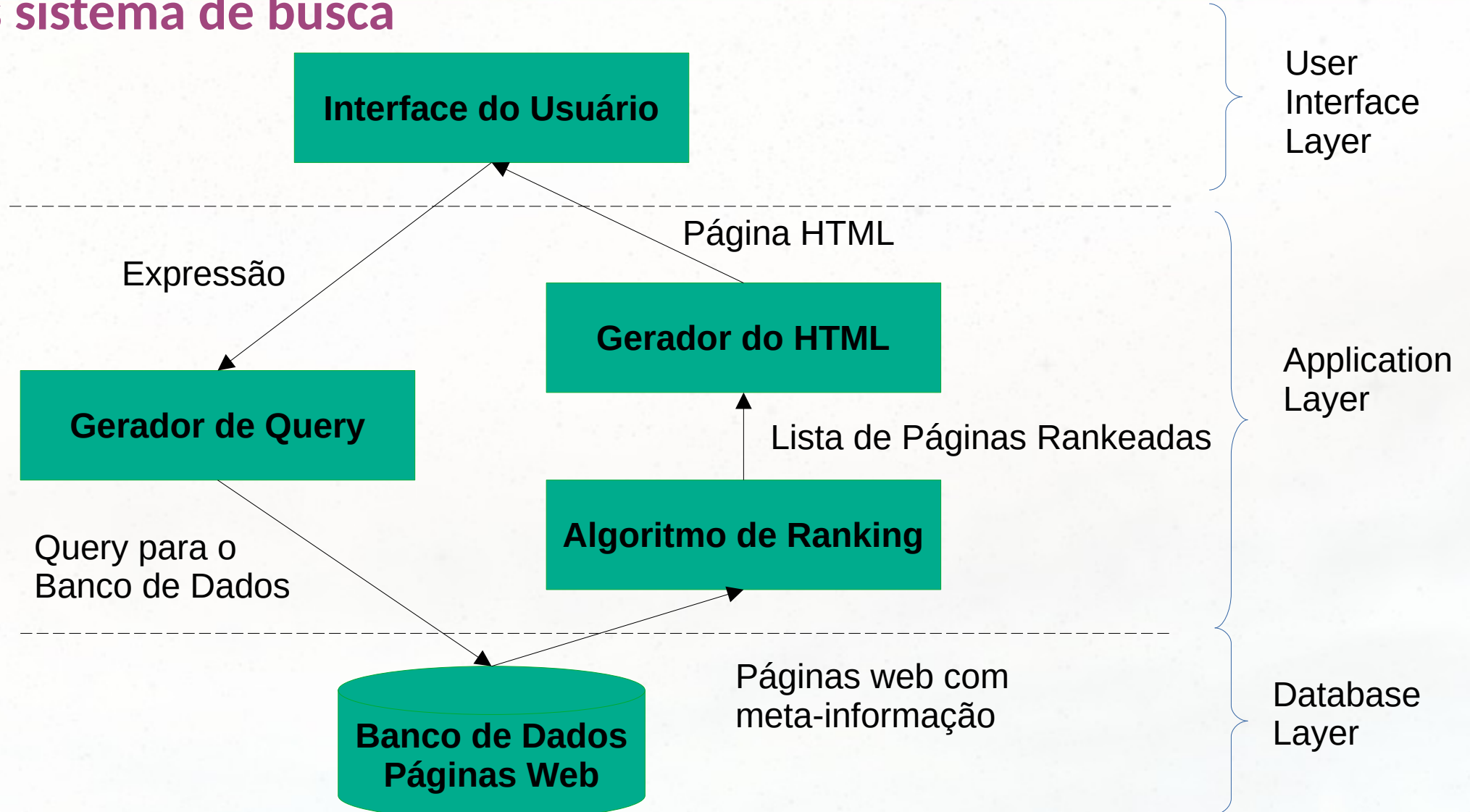
## Três Camadas (Three-tiered)





# Exemplo - Modelo Cliente-Servidor em Camadas

## Um simples sistema de busca



Dúvidas??





# Exercícios



@theAwkwardYeti



# Exercícios

Descreva e ilustre uma arquitetura cliente-servidor

- para um serviço de e-mail
- para um serviço de notícias
- Para um servidor FTP (transferência de arquivos)

# Referências

- Livro
  - Sistemas Distribuídos: Princípios e Paradigmas - Tanenbaum
    - Cap 1 - Introdução
- Livro
  - Sistemas Distribuídos: Conceitos e Projetos - Coulouris
    - Cap 2 – Modelo de Sistemas

**Obrigado!**  
**Bom Dia!**  
**Boa Tarde!**  
**Boa Noite!**

[gustavo.custodio@anhembi.br](mailto:gustavo.custodio@anhembi.br)





ecosistema  
ănimă