



# Sistemas Distribuidos

Prof. Walter Gima



# Arquitecturas Sistemas Distribuídos



## Agenda

- Replicação

## Tolerância a Falhas

Tolerância a falhas é a capacidade do sistemas distribuídos em prevenir, se recuperar, ou ainda mascarar as falhas. E garantir que o serviço continuará a ser provido com QoS (*Quality of Service*) aceitável.

**Tolerância a Falhas  $\neq$  Sistemas Confiáveis**

## Sistemas Confiáveis

- Disponibilidade.
- Confiabilidade.
- Capacidade de Manutenção.
- Segurança.

Propriedade do sistema em estar pronto para ser usado em qualquer momento.

Propriedade do sistema em funcionar continuamente sem falhas.

Facilidade com que o sistema após a ocorrência de falha poderá ser consertado.

Propriedade do sistema em deixar de funcionar corretamente, mais nada de catastrófico ocorrerá.

## Tolerância a Falhas

Tolerância a falhas é a capacidade do sistemas distribuídos em prevenir, se recuperar, ou ainda mascarar as falhas. E garantir que o serviço continuará a ser provido com QoS (*Quality of Service*) aceitável.

**Tolerância a Falhas  $\neq$  Sistemas Confiáveis**

## Replicação

- Replicação em Sistemas Distribuídos
- Replicação Passiva
- Replicação Ativa
- Replicação Semi-Ativa

## Replicação em Sistemas Distribuídos

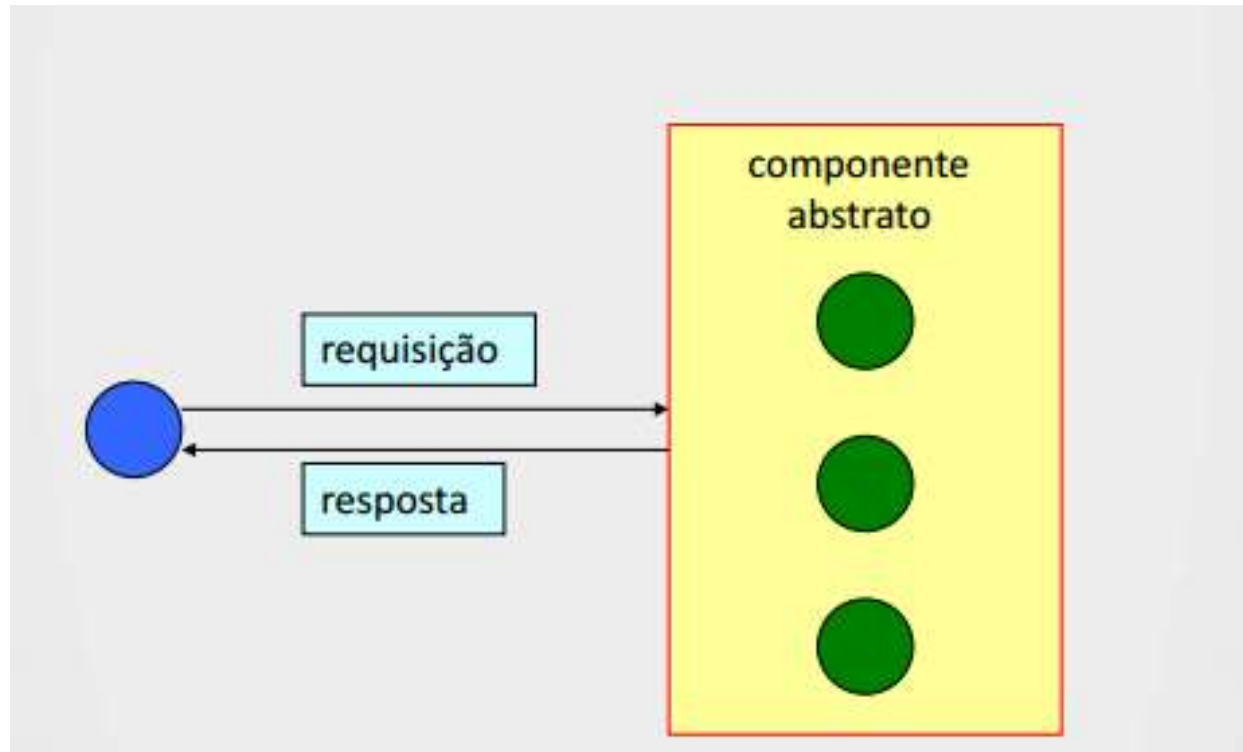
- **Objetivos**
  - Melhora na confiabilidade;
  - Aumento da disponibilidade de recursos;
  - Melhor desempenho do sistema;



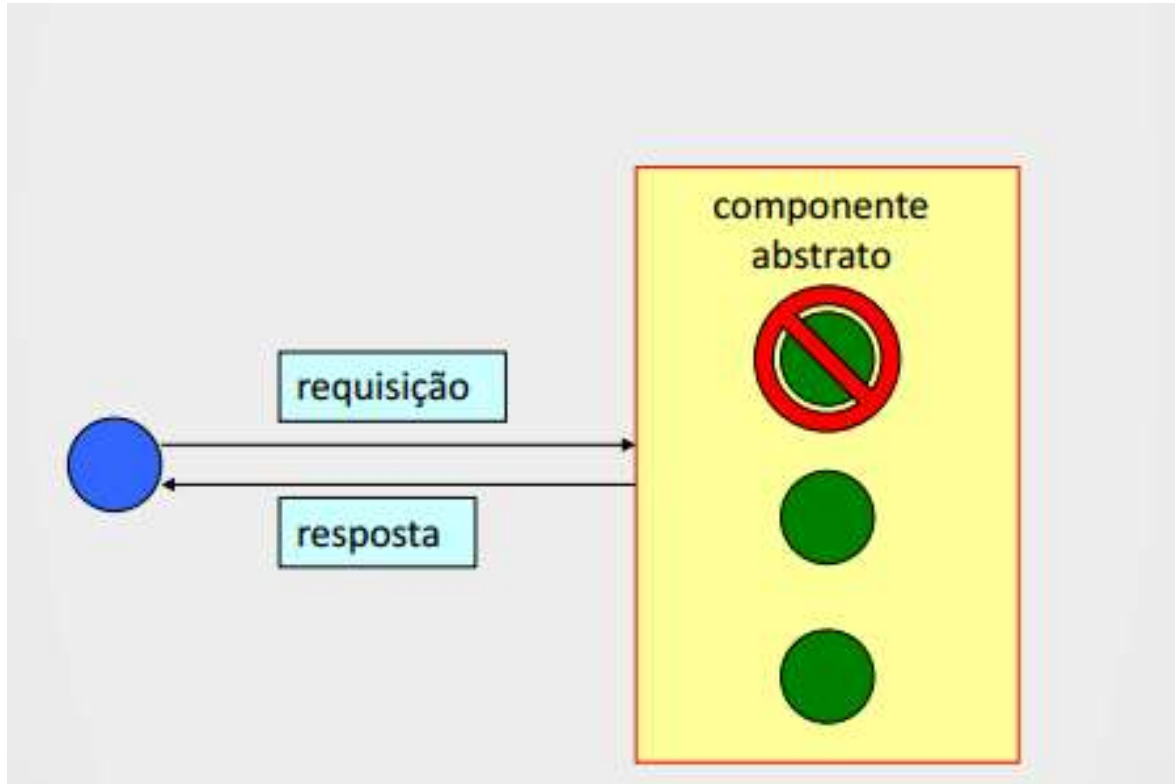
## Replicação em Sistemas Distribuídos

- Construção de componentes (e serviços) tolerantes a faltas;
- Componente abstrato:
  - Formado por um conjunto de componentes replicados;
  - Exibe as propriedades de um componente único.

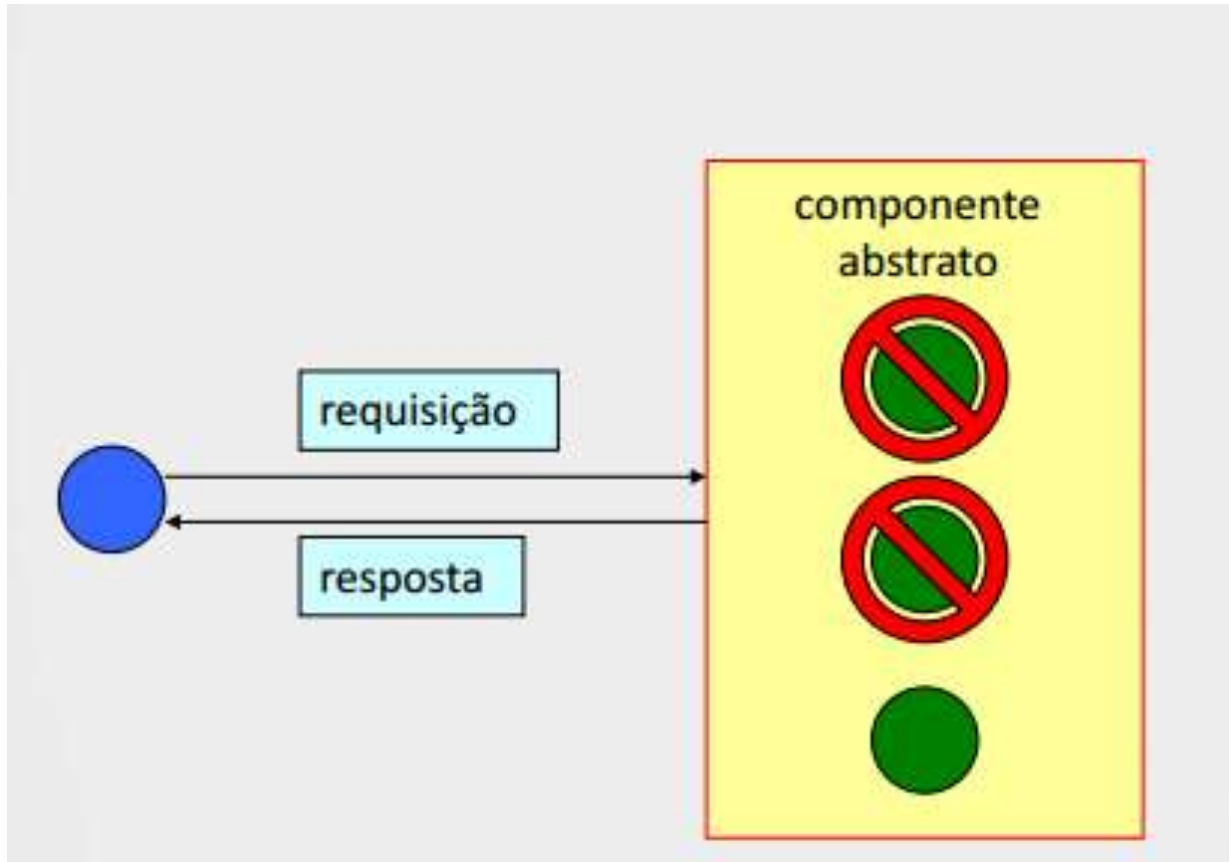
## Replicação em Sistemas Distribuídos



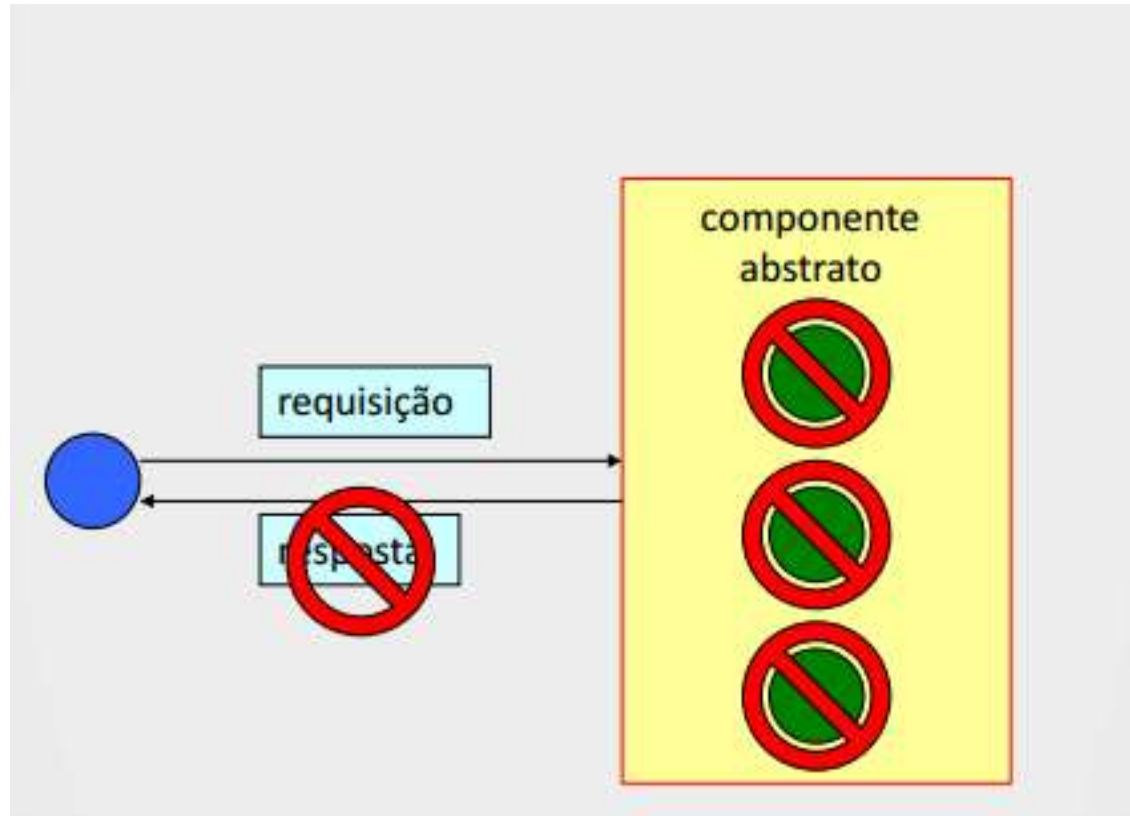
## Replicação em Sistemas Distribuídos



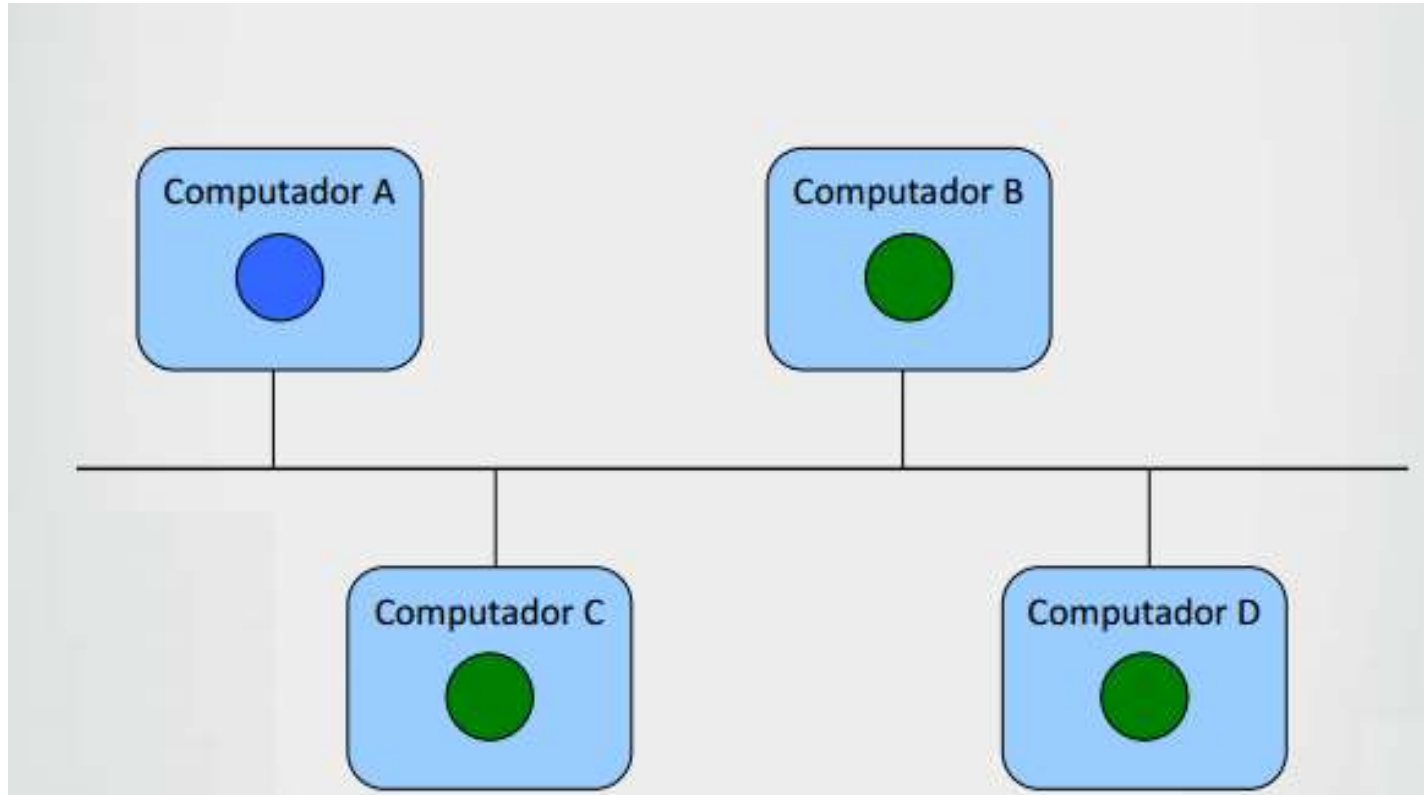
## Replicação em Sistemas Distribuídos



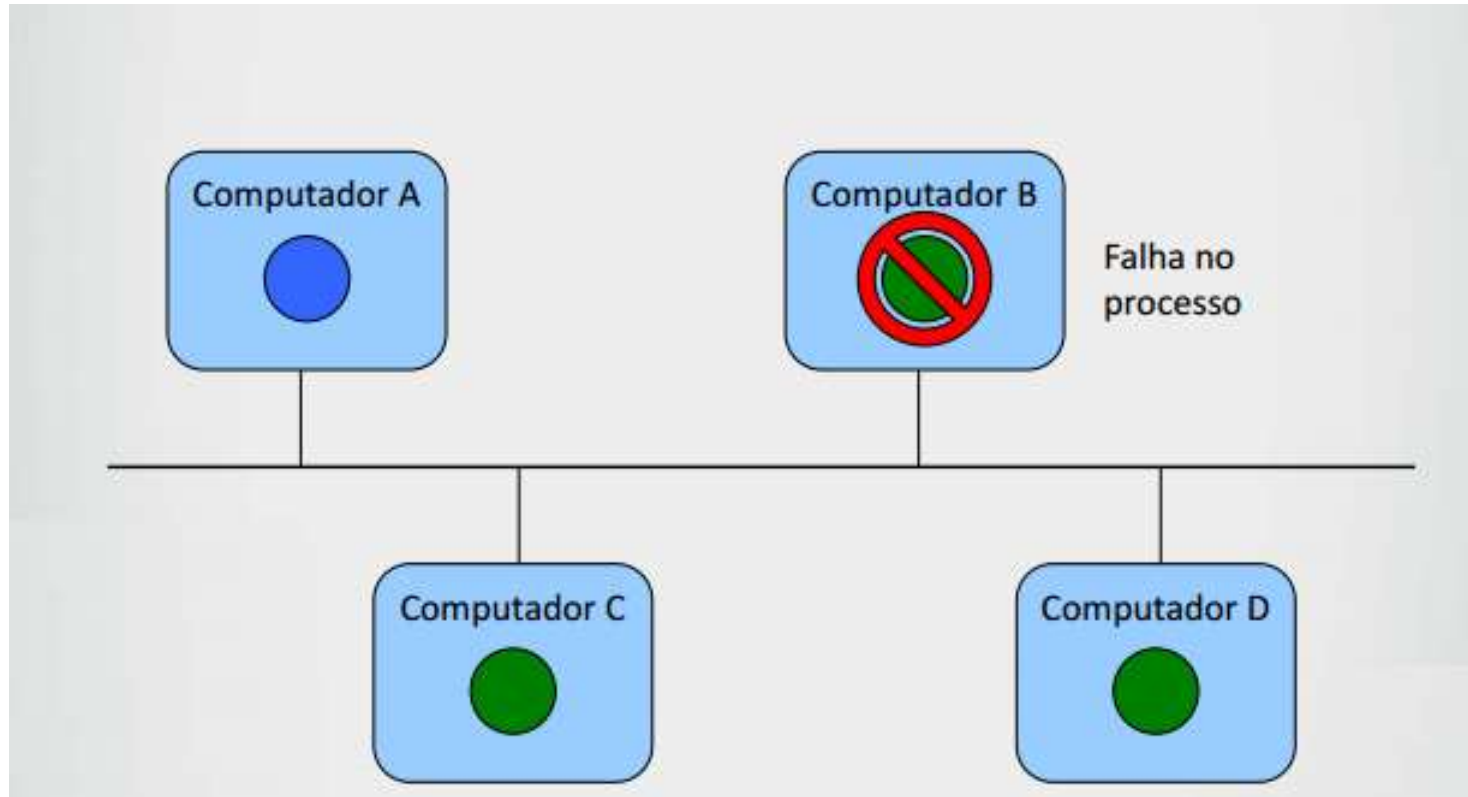
## Replicação em Sistemas Distribuídos



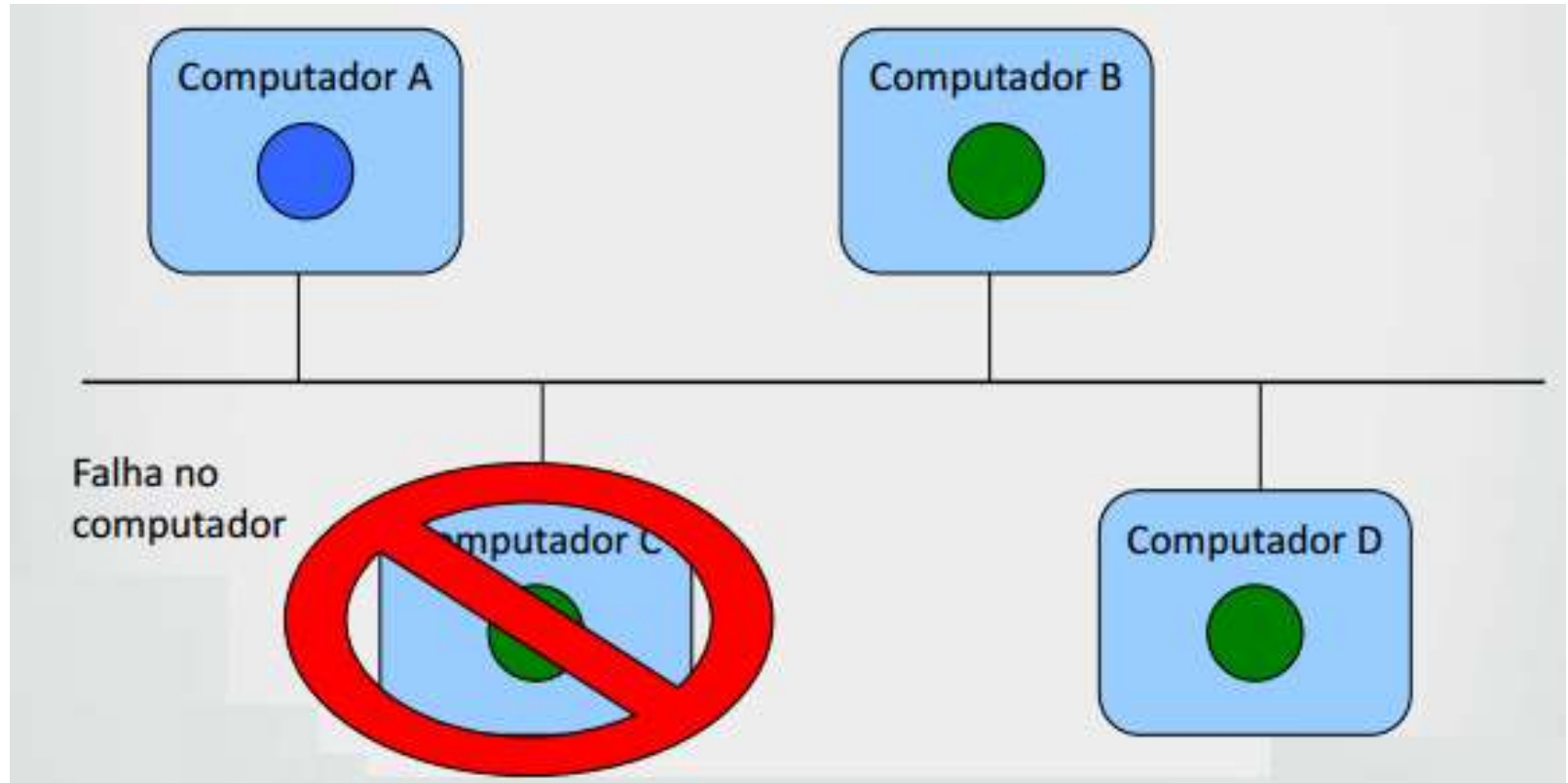
## Replicação em Sistemas Distribuídos



## Replicação em Sistemas Distribuídos

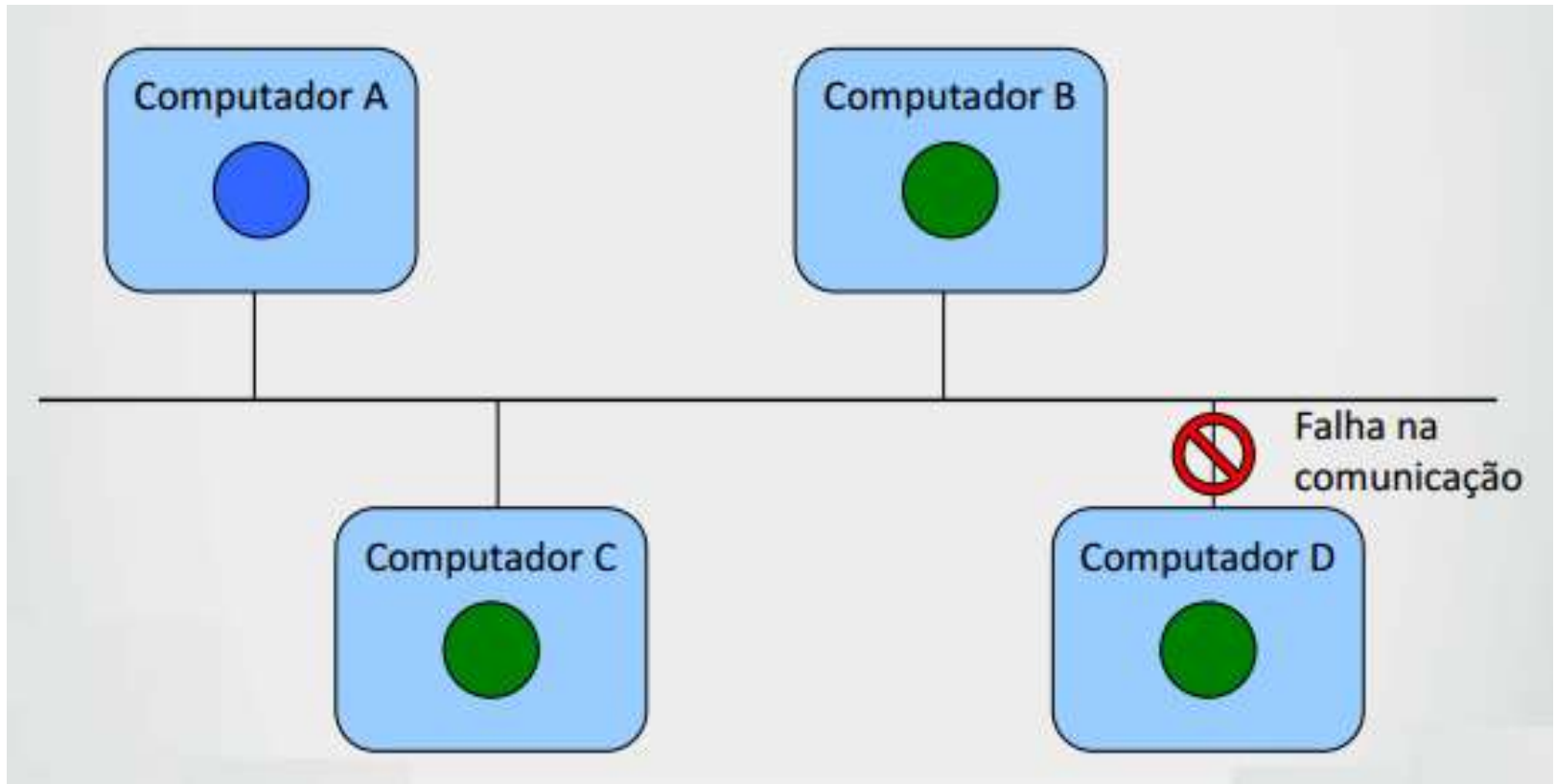


## Replicação em Sistemas Distribuídos

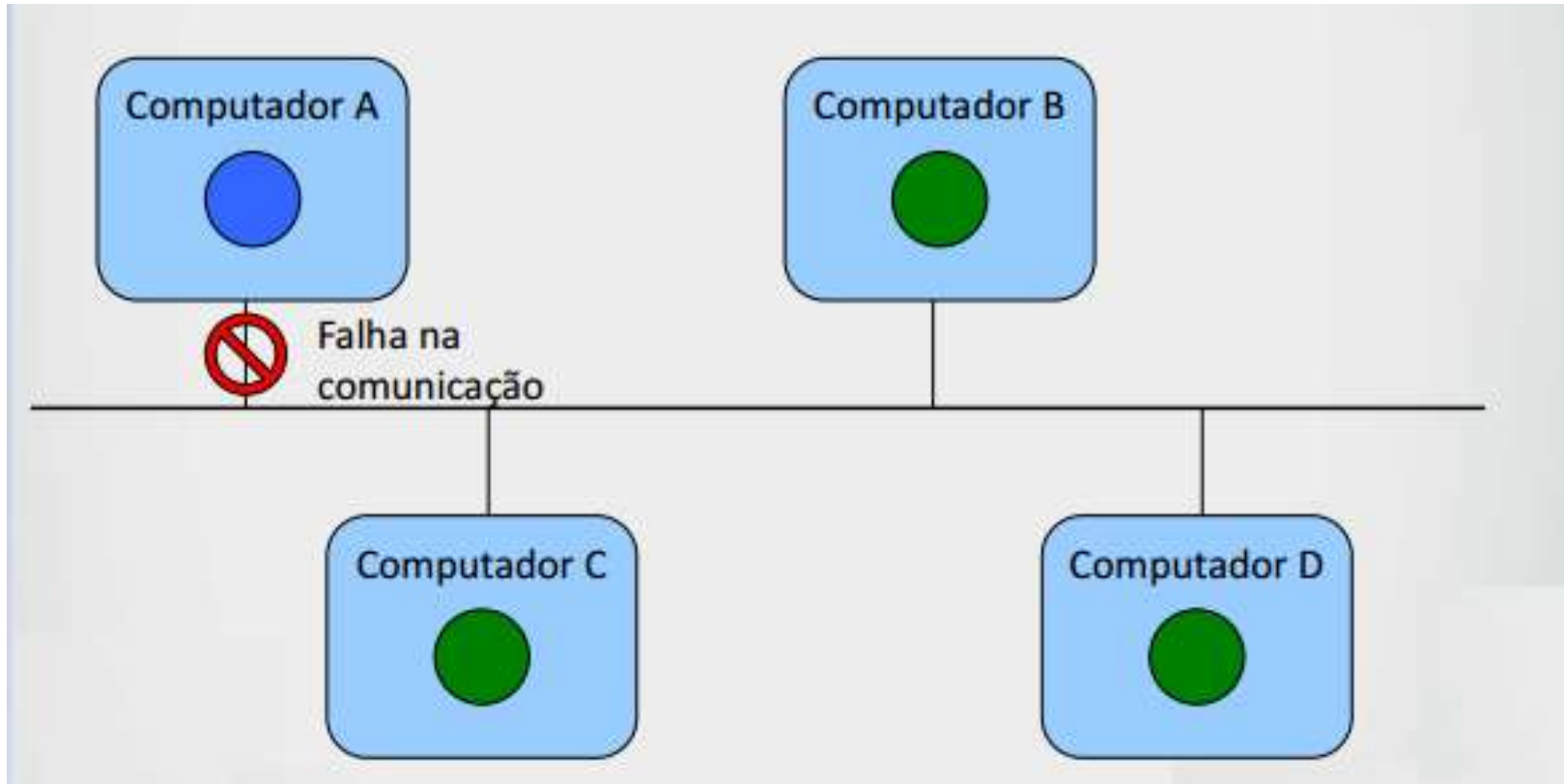




## Replicação em Sistemas Distribuídos



## Replicação em Sistemas Distribuídos



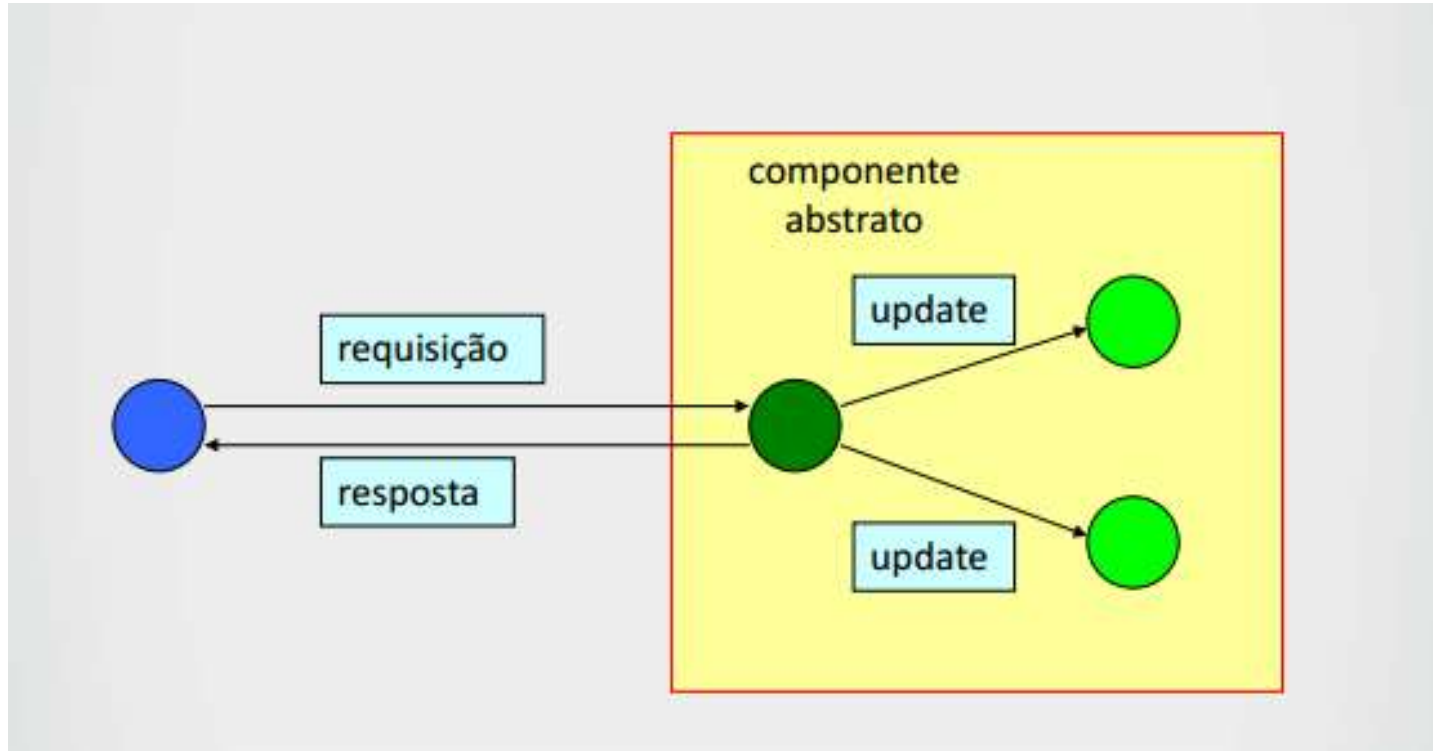
## Replicação em Sistemas Distribuídos

- Protocolo de coordenação:
  - Transparência do conjunto;
  - Consistência de estado;
  - Controle de concorrência;
- Recuperação:
  - Falha parcial;
  - Falha total.

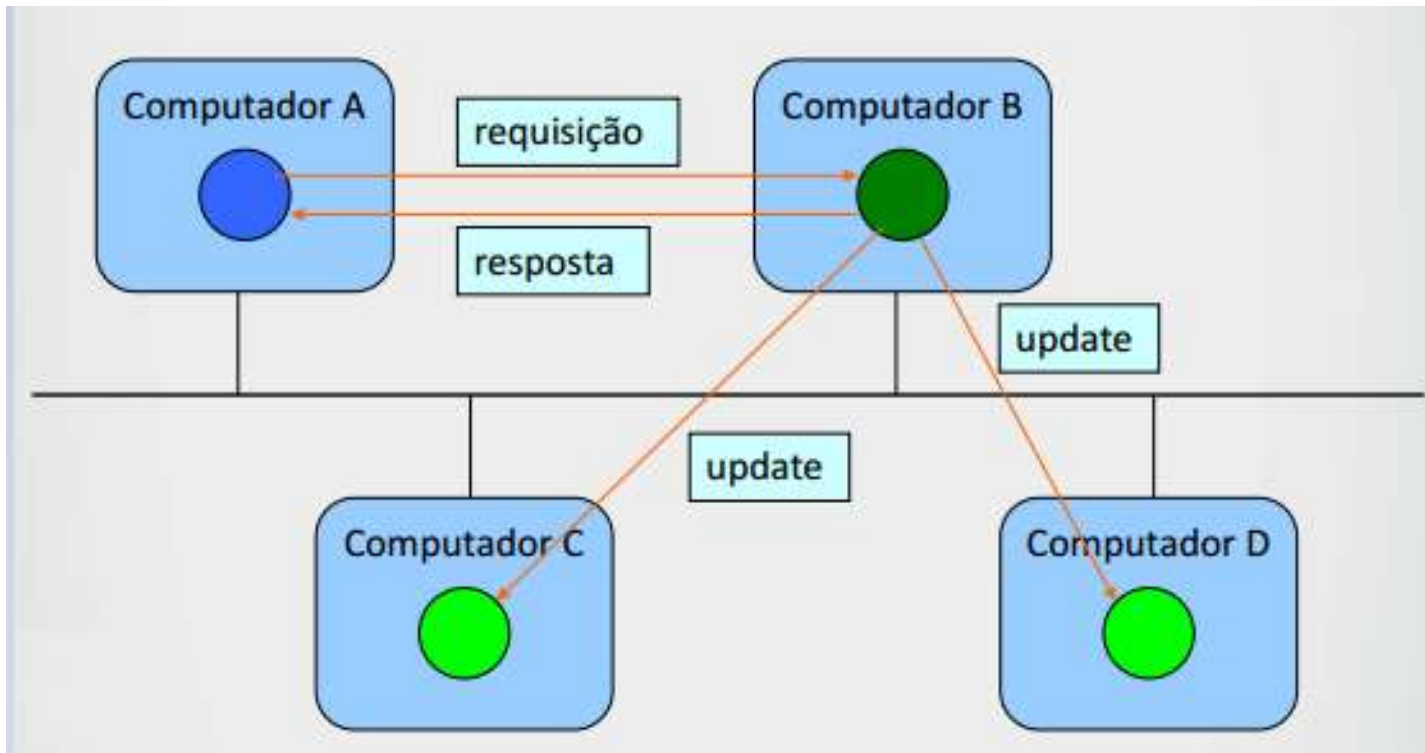
## Replicação Passiva

- Somente um membro (**primário**) recebe, executa e responde as requisições;
- Os demais membros (**backups**) são atualizados por meio de mensagens de *update* enviadas pelo primário (*checkpoint*);
- Em caso de falha do primário, um novo primário é eleito entre os backups;
- Nenhuma requisição é processada durante a eleição do novo primário;

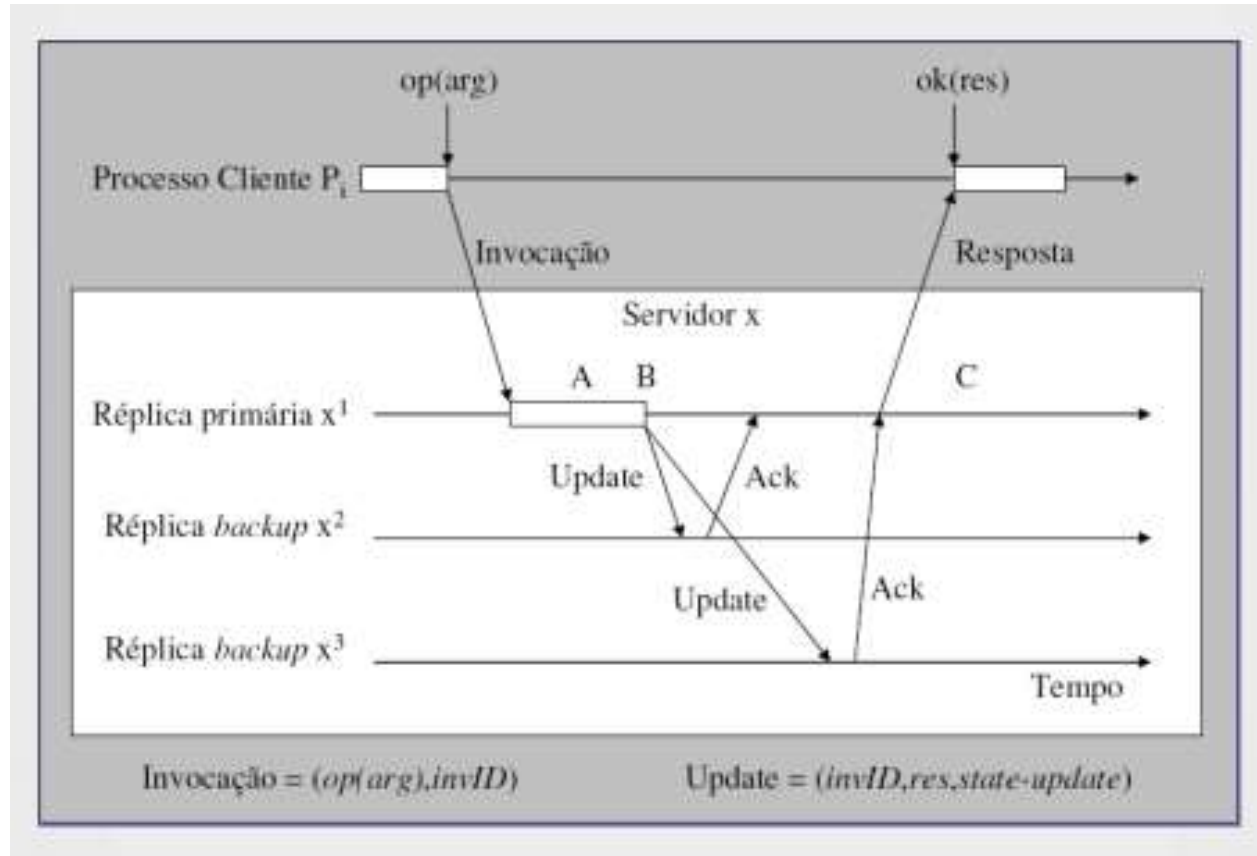
## Replicação Passiva



## Replicação Passiva



## Replicação Passiva

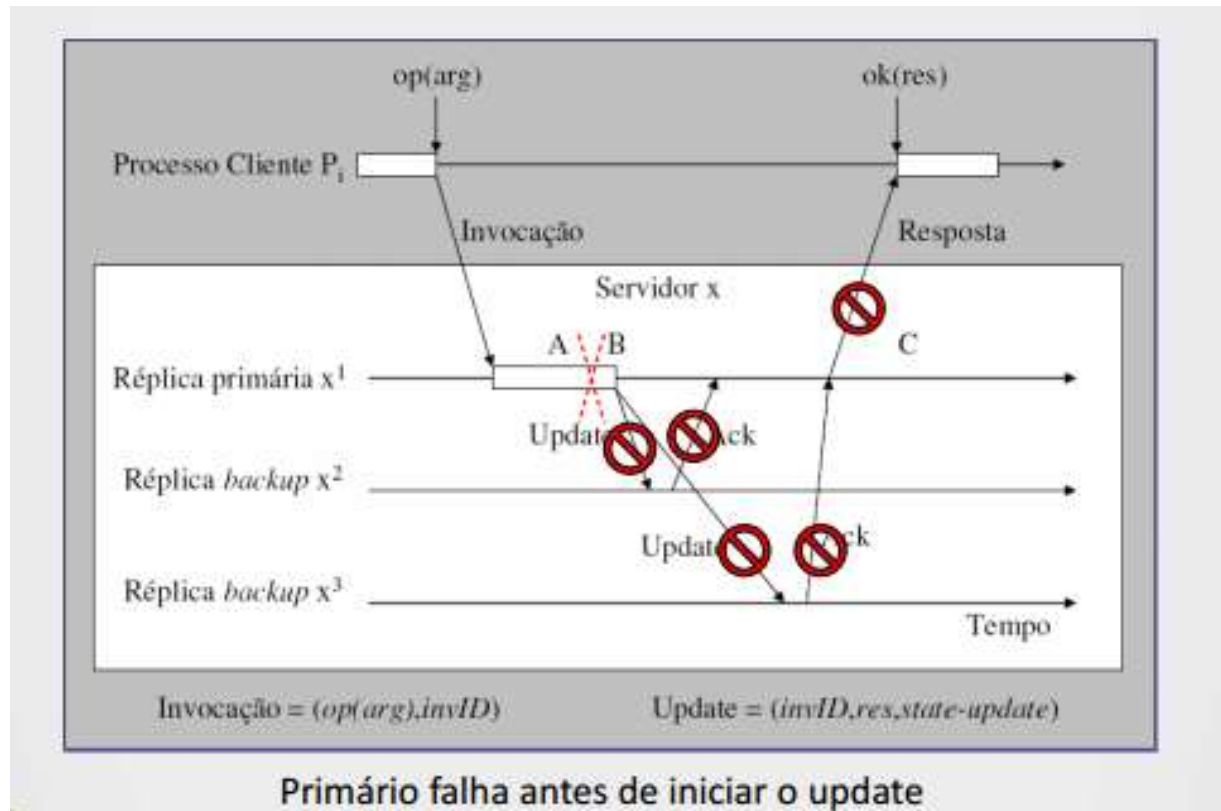


## Replicação Passiva

- Falha do primário
  - Detecção:
    - Cliente estabelece *timeout* para requisição;
    - Backups fazem verificação (*keepalive*).
  - Um novo primário é eleito;
  - Casos:
    - Antes de iniciar o update;
    - Durante ou após o update, mas antes de enviar a resposta ao cliente;
    - Após enviar a resposta ao cliente;



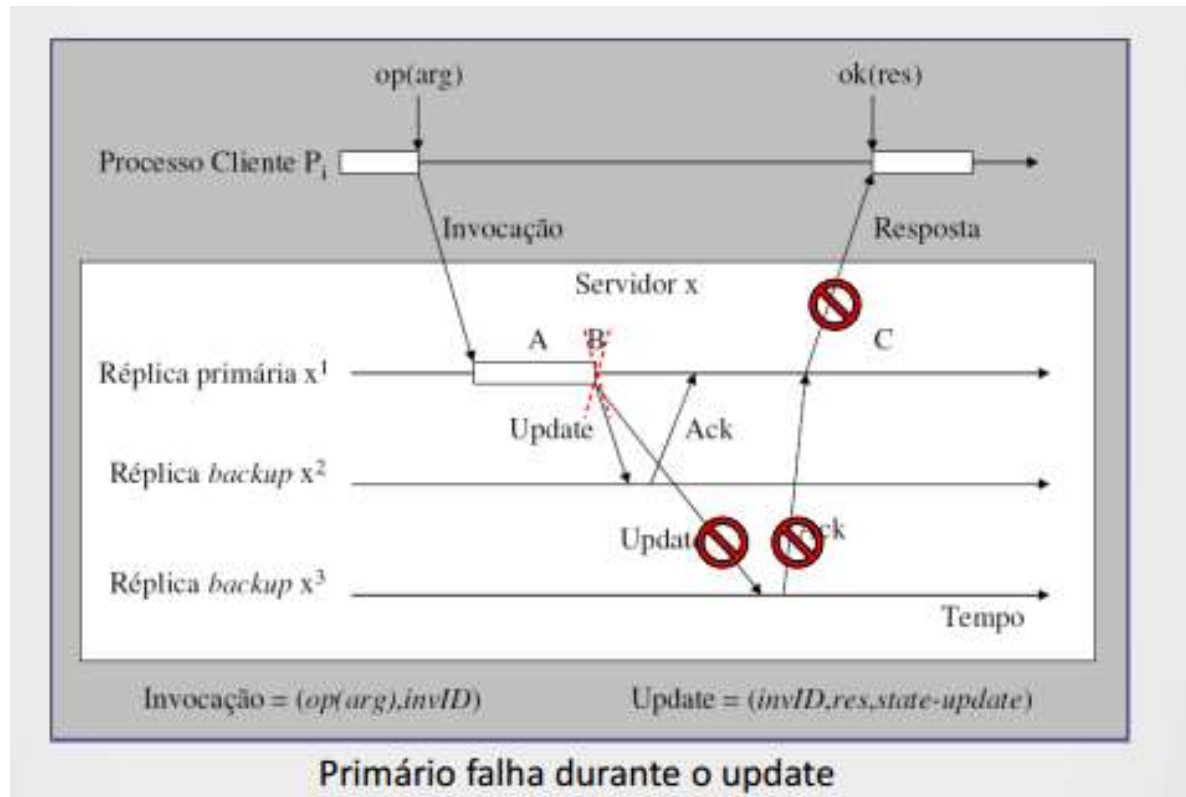
## Replicação em Sistemas Distribuídos



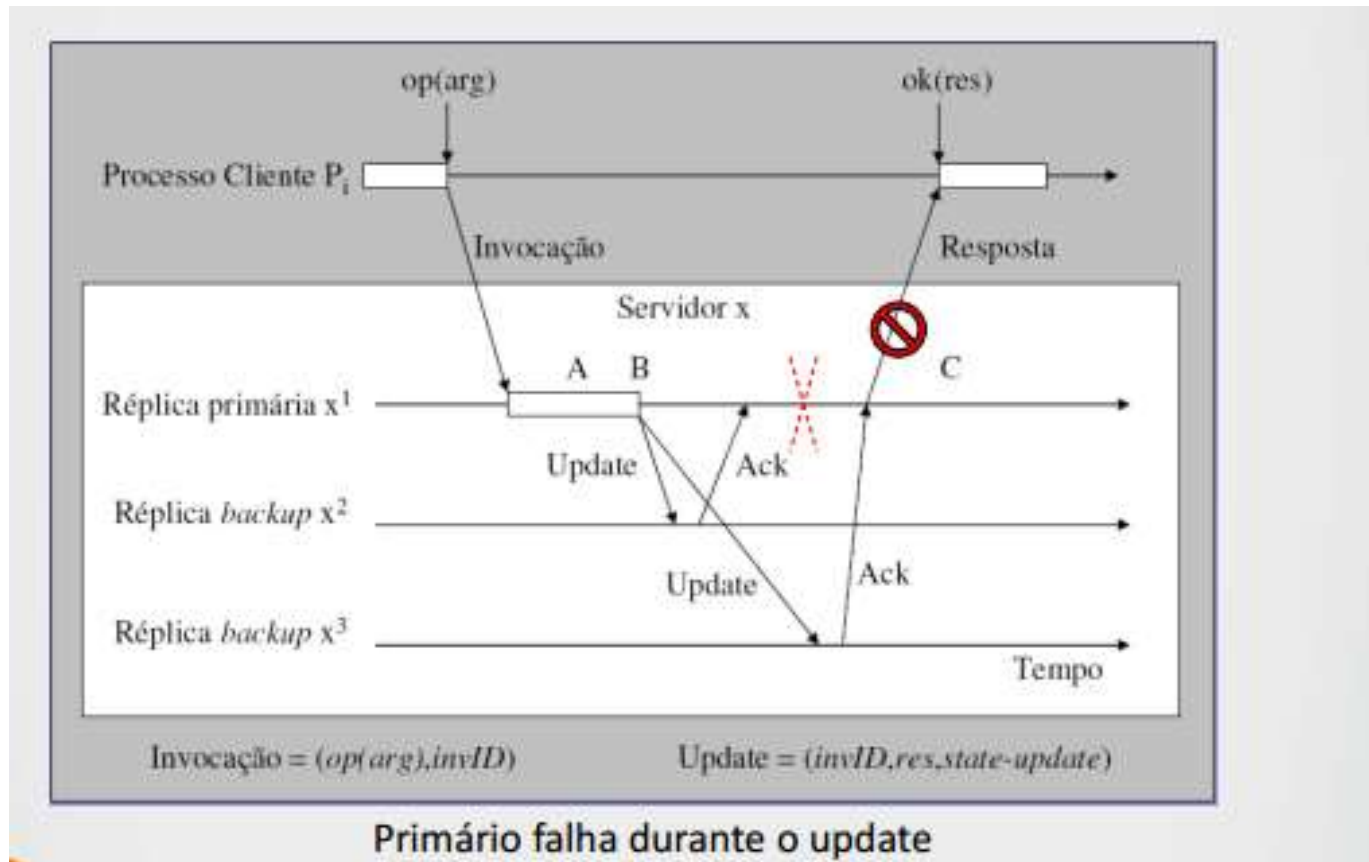
## Replicação Passiva

- Primário falha antes de iniciar o update
  - Cliente não recebe resposta e suspeita que servidor está falho;
  - Cliente refaz a requisição;
  - Novo primário considera a requisição como nova e segue adiante.

## Replicação Passiva



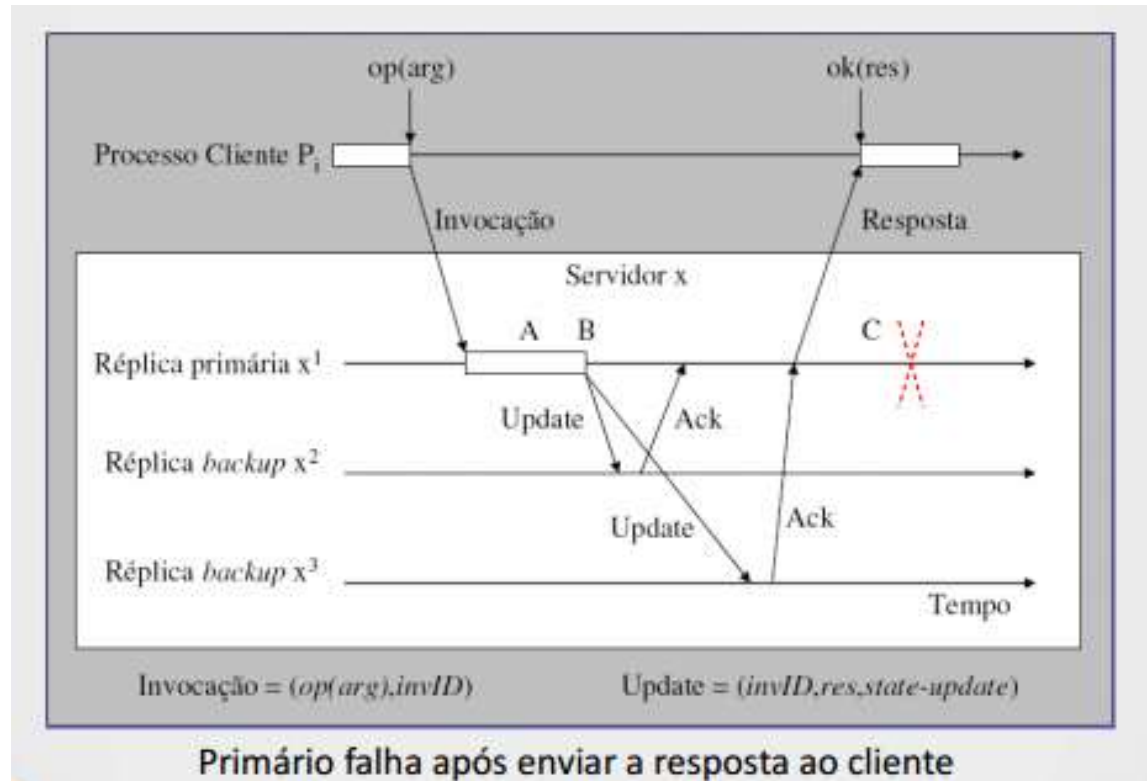
## Replicação Passiva



## Replicação Passiva

- Primário falha durante o update
  - Cliente não recebe resposta e suspeita que servidor está falho;
  - Cliente refaz a requisição;
  - Exige **consistência** na comunicação
    - *Nenhum* backup recebe o update (equivale ao caso de falha do primário antes de iniciar o update);
    - *Todos* os backups recebem o update:
      - O novo primário percebe a repetição da requisição e envia a resposta ao cliente.

## Replicação Passiva



## Replicação Passiva

- Primário falha após enviar a resposta ao cliente
  - A falha é transparente para o cliente.

## Replicação Passiva

- Vantagens

- Réplicas não precisam ter comportamento determinístico (obter sempre o mesmo efeito para uma certa requisição);
- Interação simples entre cliente e servidor.

- Desvantagens

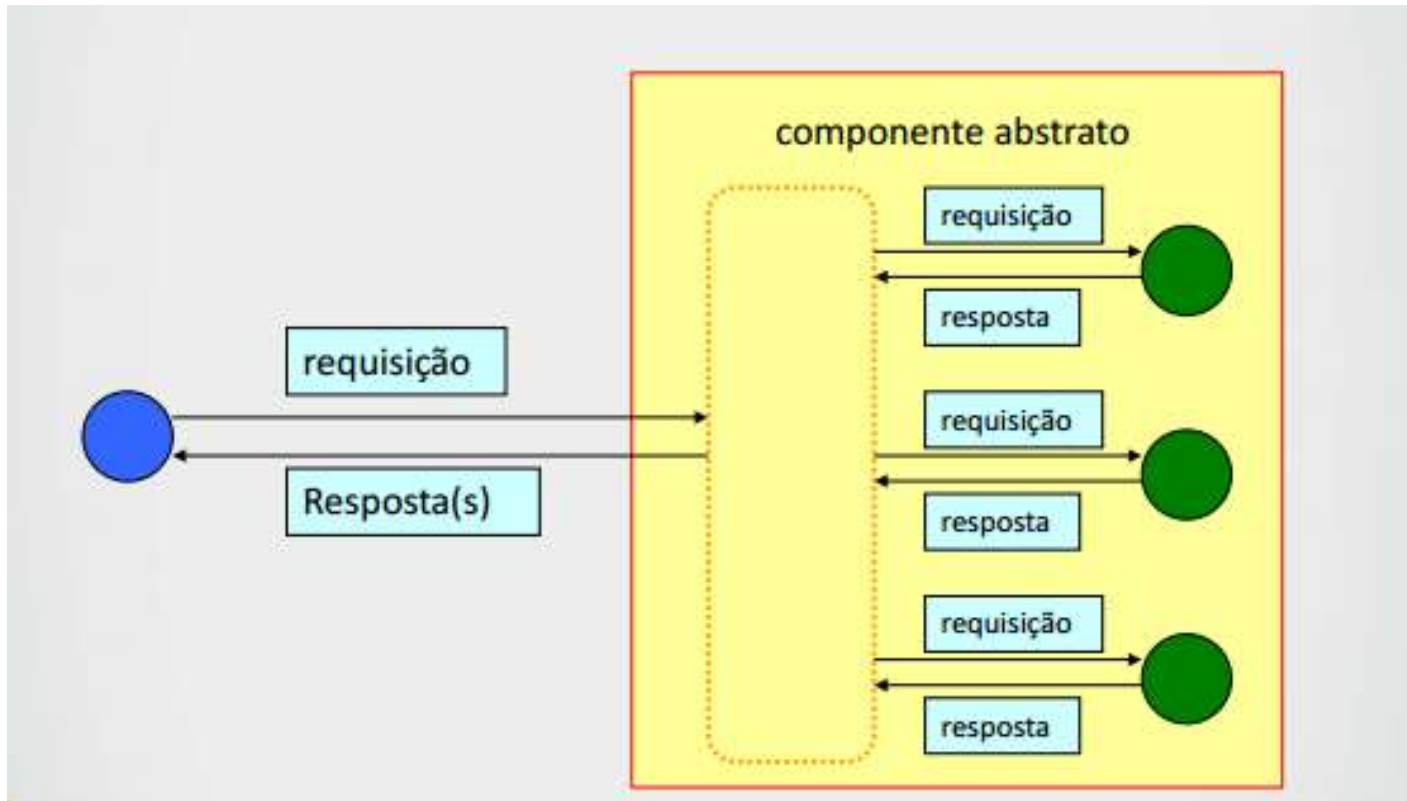
- A frequência de checkpoints pode prejudicar o desempenho do serviço replicado.
  - Solução: fazer update a cada  $n$  requisições.
    - Cliente retransmite requisições entre o último update realizado e a falha do primário.
    - Um mecanismo de *log* em disco armazena todas as requisições desde o último checkpoint.



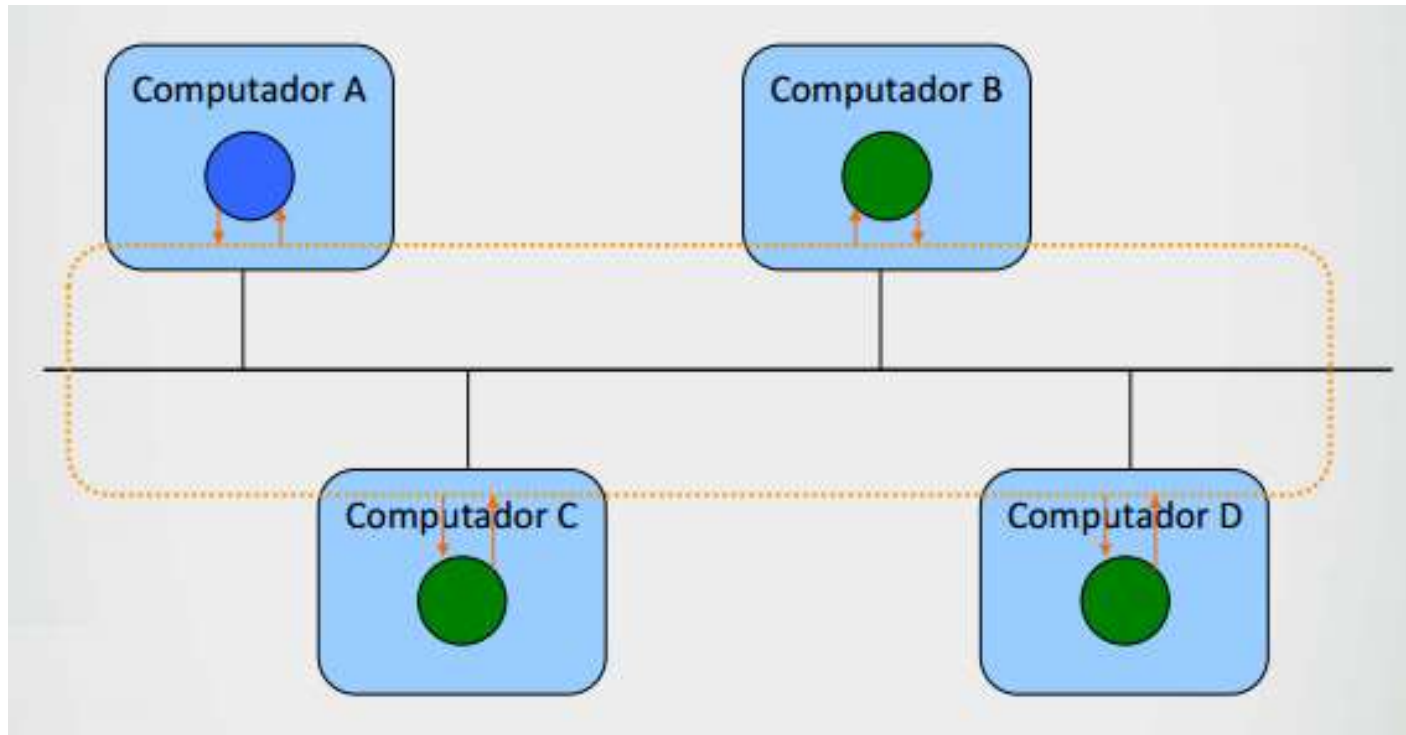
## Replicação Ativa

- Todas as réplicas são ativas: recebem, executam e respondem a todas as requisições dos clientes;
- Formas de entrega das respostas
  - A primeira resposta a chegar é passada ao cliente.
  - As respostas são concatenados em sequência e enviadas ao cliente.
  - As respostas passam por um **votador** que seleciona a mais frequente (maioria) para ser passada ao cliente.

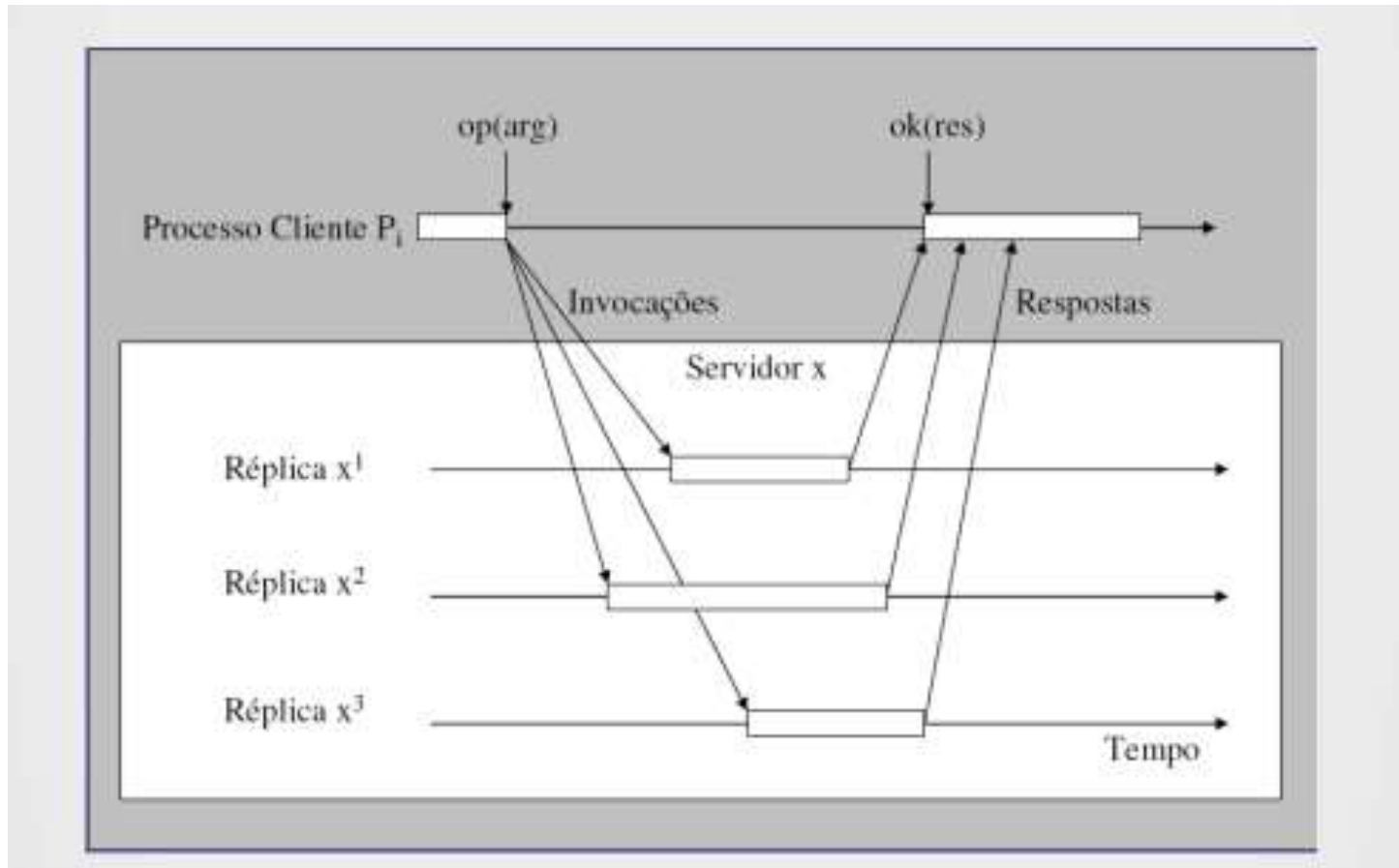
## Replicação Ativa



## Replicação Ativa



## Replicação Ativa



## Replicação Ativa

- Falha de uma réplica:
  - Transparente para o cliente;
  - Recuperação de uma réplica:
    - Atualização do estado
    - Execução das requisições perdidas

## Replicação Ativa

- Exige que as réplicas tenham comportamento determinístico:
  - Uma requisição produz o mesmo efeito em todas as réplicas.
  - Caso contrário, pode ocorrer divergência de estados.
- Exige consistência na comunicação:
  - Uma certa mensagem (requisição) é recebida por todas as réplicas ou, então, não é recebida por nenhuma réplica.
- Exige ordenação na comunicação:
  - Todas as réplicas recebem as mensagens (requisições) na mesma ordem.

## Replicação Ativa

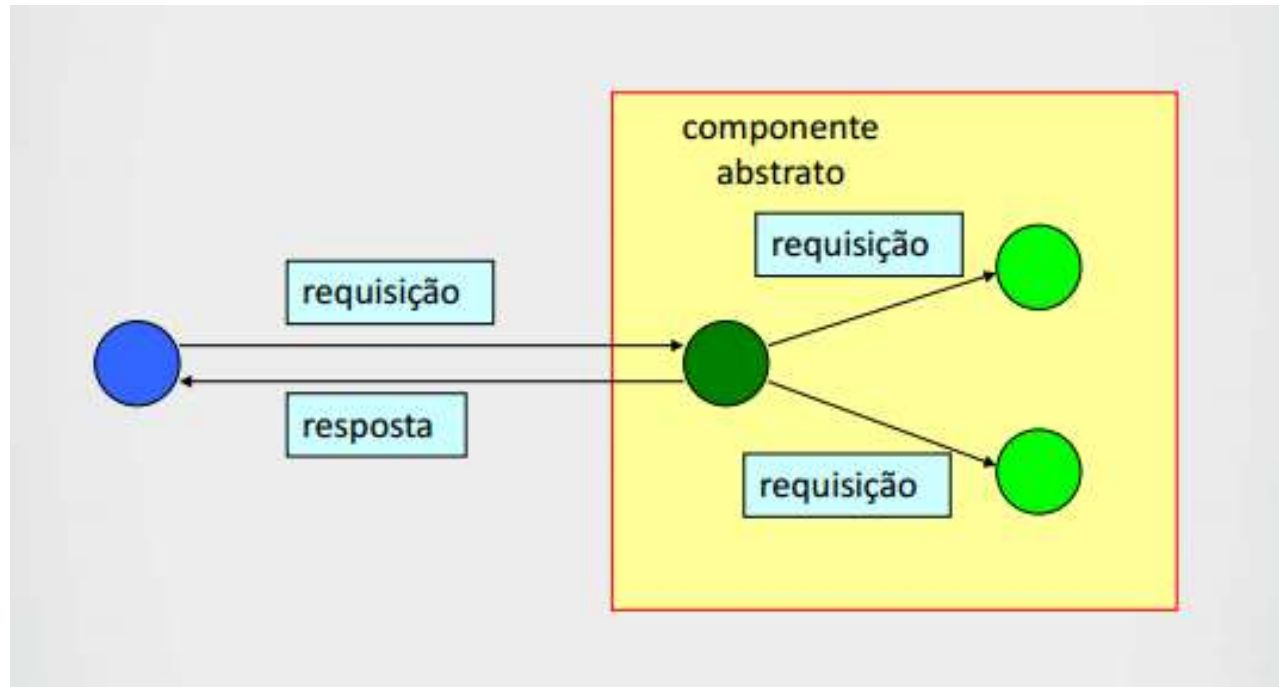
- **Vantagens**
  - Falhas são mascaradas quase que instantaneamente;
  - Adequada para aplicações que exigem serviços ininterruptos e com sobrecarga mínima em situações de falha, como aplicações de tempo real;
  - Cobre um amplo espectro de faltas: crash, omissão, temporização, valor e arbitrária;
- **Desvantagens**
  - Tem alto custo na comunicação;
  - Exige muitos recursos do sistema (memória, processador, etc).

## Replicação Semi-Ativa

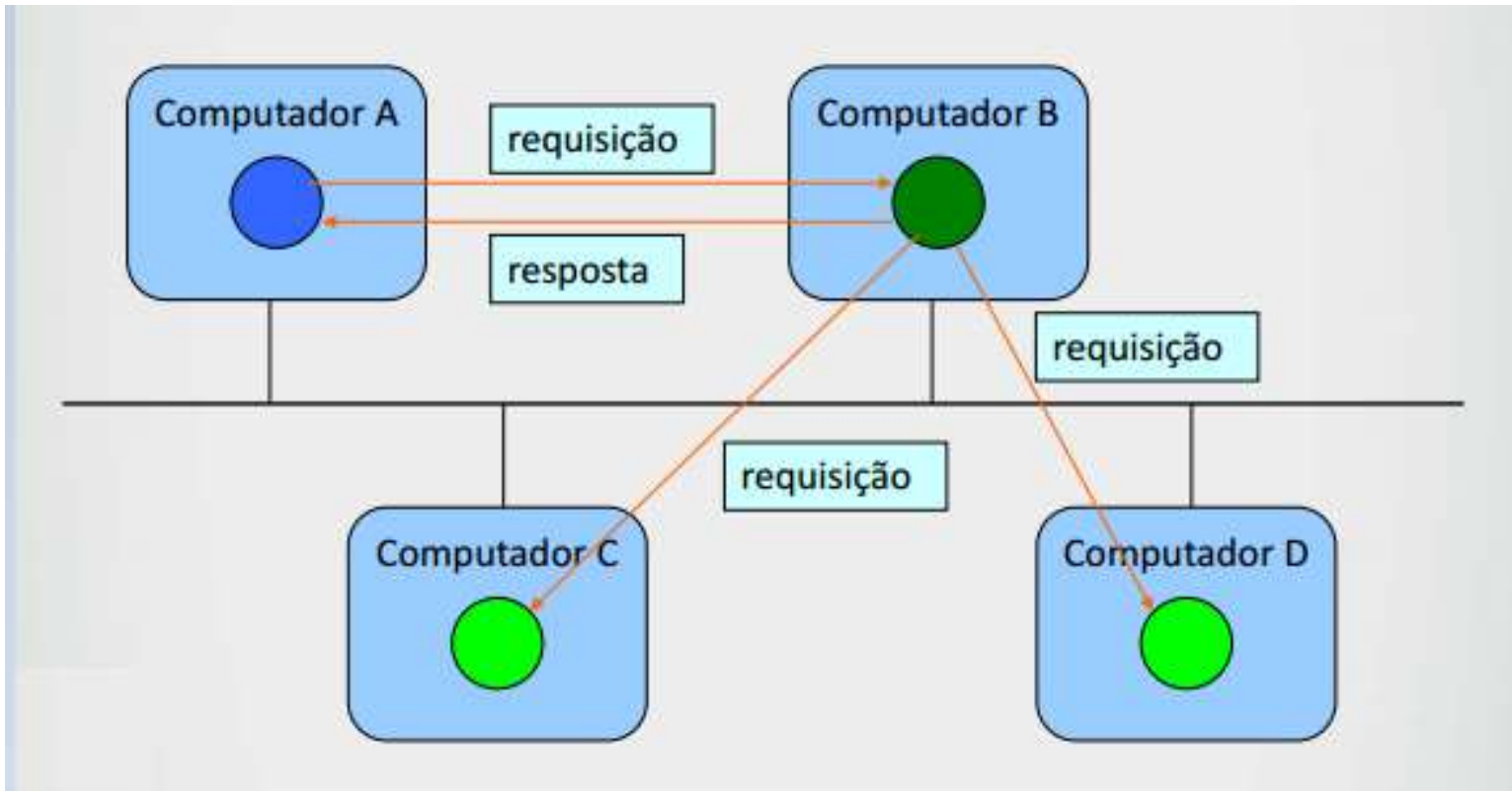
- Todas as réplicas são ativas, mas uma delas é a **líder**.
- Líder
  - Recebe todas as requisições;
  - Determina a ordem de execução das requisições;
  - Encaminha as requisições para as demais réplicas;
  - Fornece a resposta ao cliente.



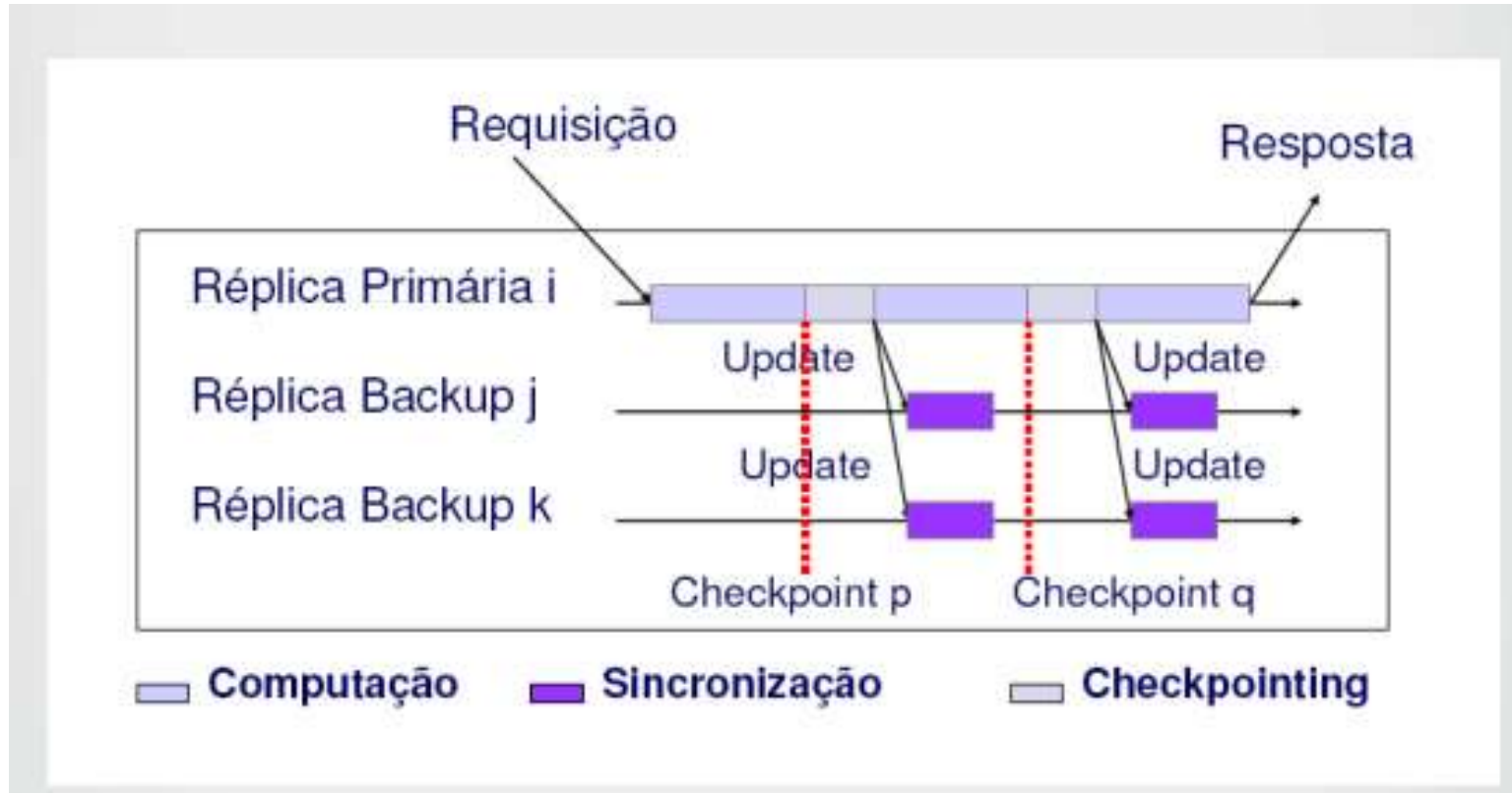
## Replicação Semi-Ativa



## Replicação em Sistemas Distribuídos



## Replicação Semi-Ativa



## Replicação Semi-Ativa

- **Vantagens**

- Ordenação das mensagens é simples, pois é feita por uma líder.
- Interação simples cliente-servidor
- Falha de uma réplica é transparente para o cliente
- Adequada para aplicações de tempo real

- **Desvantagens**

- Exige comportamento determinístico das réplicas
- Exige atomicidade na comunicação
- Exige muitos recursos do sistema

- Tanenbaum, Andrew S; Steen, Maarten Van. Sistemas Distribuídos: Princípios e Paradigmas, 2ª ed. São Paulo: Pearson – Prentice Hall, 2007.
- Coulouris, George. Sistemas Distribuídos: Conceitos e Projetos, 4ª ed. Rio de Janeiro: Bookman, 2007.
- fonte: <https://pt.wikipedia.org/wiki/MVC>



Anhanguera

Dúvidas ?

walter.gima@anhanguera.com