

Computação Distribuída

Odorico Machado Mendizabal



Universidade Federal de Santa Catarina – UFSC
Departamento de Informática e Estatística – INE



Replicação

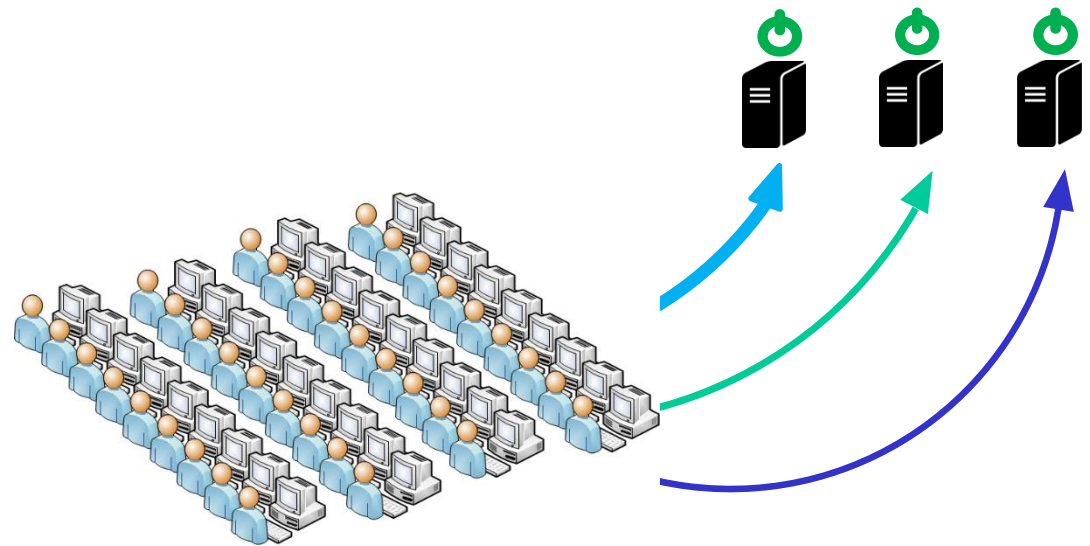
Replicação de Dados/Serviços

- Técnica fundamental para prover alta disponibilidade e tolerância a falhas em Sistemas Distribuídos
 - Redundância física
- Pode ser empregada em diferentes partes de aplicação
 - Exemplo: Arquiteturas multicamadas podem ter réplicas em uma ou mais camadas

Vantagens ao Replicar Dados/Serviços

Replicação para aumentar desempenho:

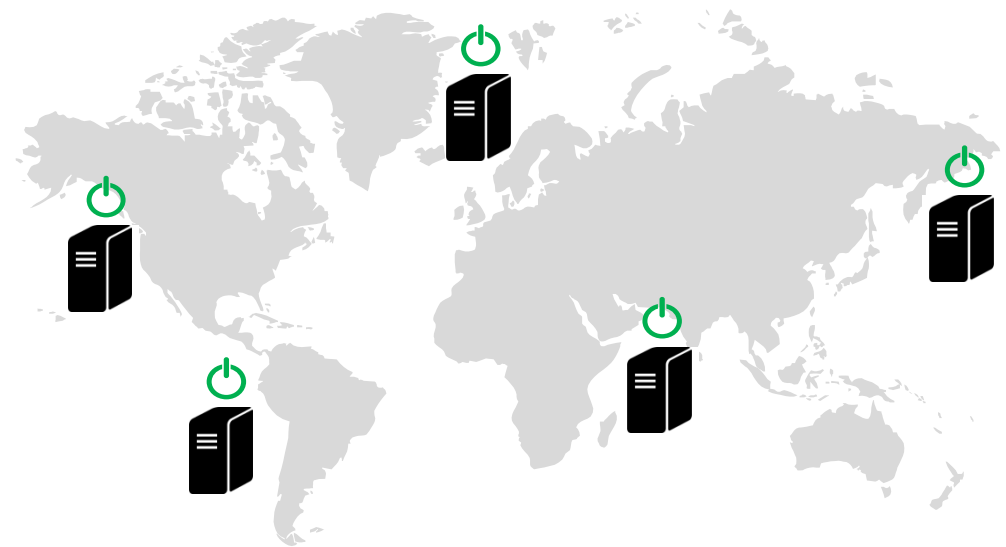
- Com o acréscimo de carga, mais réplicas são necessárias para atender o volume de requisições



Vantagens ao Replicar Dados/Serviços

Replicação para aumentar desempenho:

- Com o acréscimo de carga, mais réplicas são necessárias para atender o volume de requisições
- Diversidade de localidades geográficas oferece menores latências em ambientes amplamente distribuídos



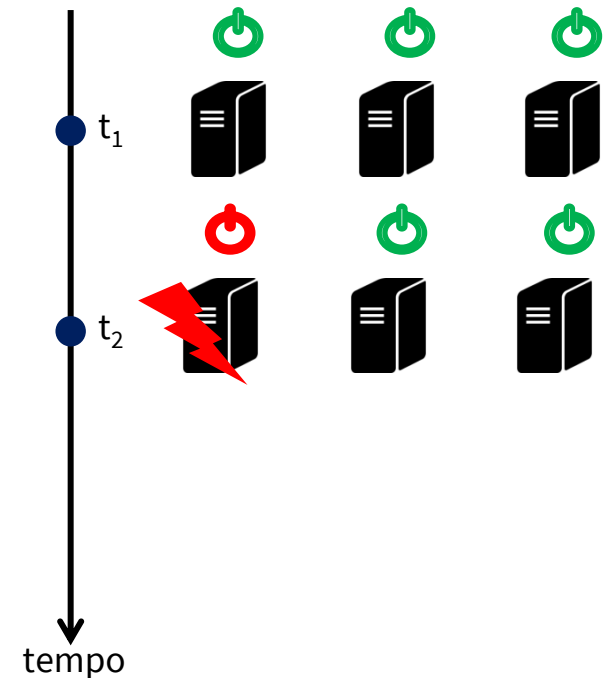
Vantagens ao Replicar Dados/Serviços

Replicação para aumentar disponibilidade:

Tolerância a falhas

- Com réplicas de um serviço, mesmo que alguma réplica falhe, outras estarão operacionais
- Evita que a execução do serviço seja interrompida (aumenta a disponibilidade com redundância física)

Um quórum de réplicas corretas capaz de tolerar até f réplicas falhas consegue manter o serviço operacional

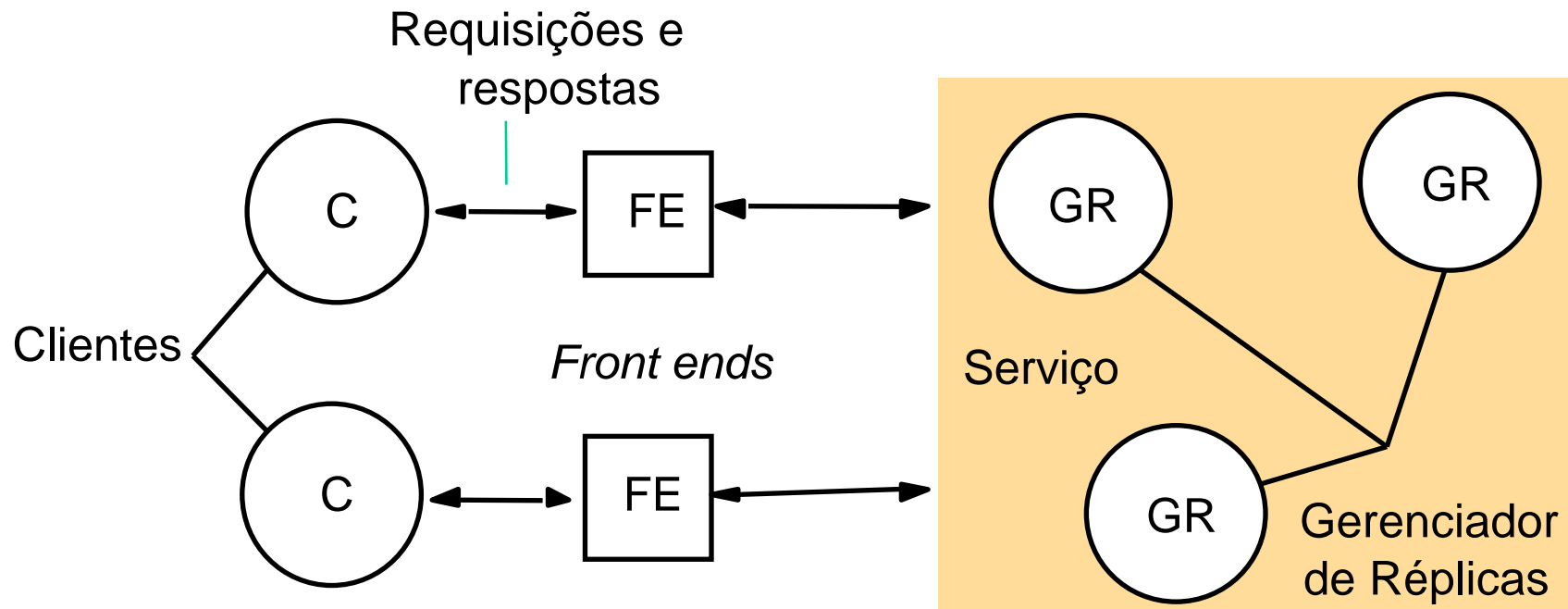


Cálculo da Disponibilidade:

- probabilidade de 1 servidor falhar: $p = 5\%$
- número de servidores: $n = 2$

$$d = 1 - 0,05^2 \therefore d = 99,75\%$$

Arquitetura Básica de um Sistema com Replicação



- Gerenciador de Réplicas aplica operações em réplicas de forma recuperável
 - Evita resultados inconsistentes

Gerenciadores de Réplicas

Podem ser **estáticos** ou **dinâmicos**

- Gerenciadores estáticos são compostos por um grupo fixo de réplicas
- Gerenciadores dinâmicos permitem a inclusão de novas réplicas em tempo de execução
 - Exemplo: Sistemas P2P, sistemas ubíquos, etc..

Comunicação em Grupo

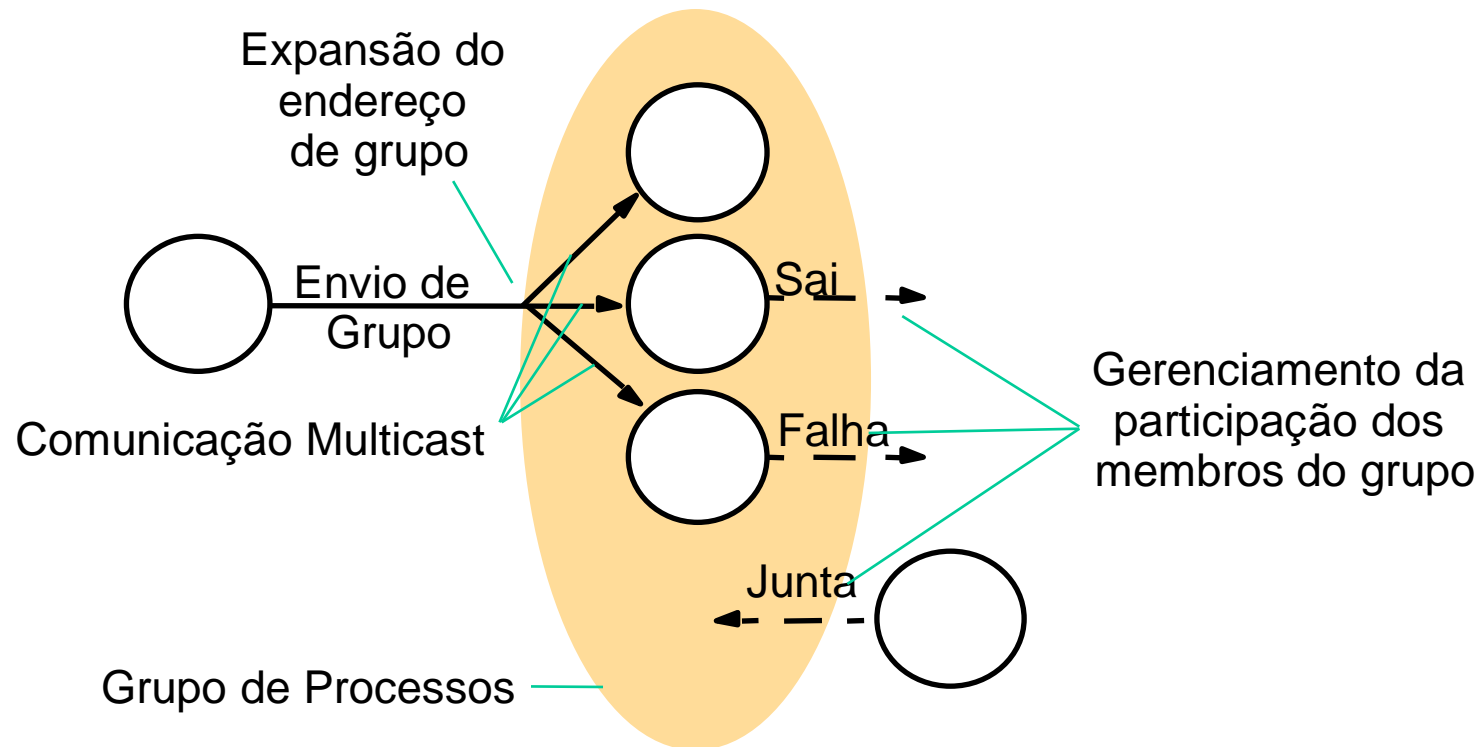
Serviço de Comunicação em Grupo é um bom estudo de caso sobre sistemas de replicação

Propriedades da Comunicação em Grupo

- Tipos de comunicação (*unicast*, *multicast* e *broadcast*)
- Tipos de Grupos (abertos e fechados)
- Hierarquia de Grupos (hierárquicos e não hierárquicos)
- Dinâmica de Grupos (criação, destruição de grupos)
- Atomicidade e Ordenação

Dinâmica de Grupos

- novos grupos podem ser criados
- grupos podem ser destruídos
- processos podem juntar-se ou deixar um grupo
- um processo pode ser membro de mais de um grupo simultaneamente



Atomicidade e Ordenação

- **Atomicidade:**

- garantir que uma mensagem enviada um grupo, seja entregue a todos os membros, caso contrário ela é desconsiderada
- propriedade para **todos** ou para **nenhum**

- **Ordenação:**

- garantir que as mensagens sejam entregues aos processos em uma mesma ordem
 - Atomic (all-or-none),
 - Fifo,
 - Causal,
 - Total (sequencer, token)

Modos de Replicação

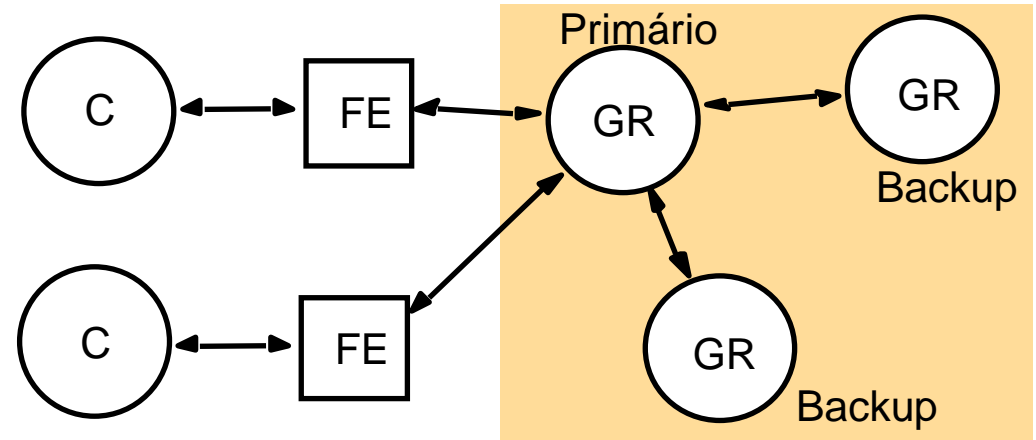
Replicação Passiva

- Modelo Primário / *Backup*
- Clientes comunicam-se apenas com um gerenciador de réplica primário
- o primário executa as requisições e as propaga aos gerenciadores *backup*
- se o primário falhar, um *backup* assume como primário

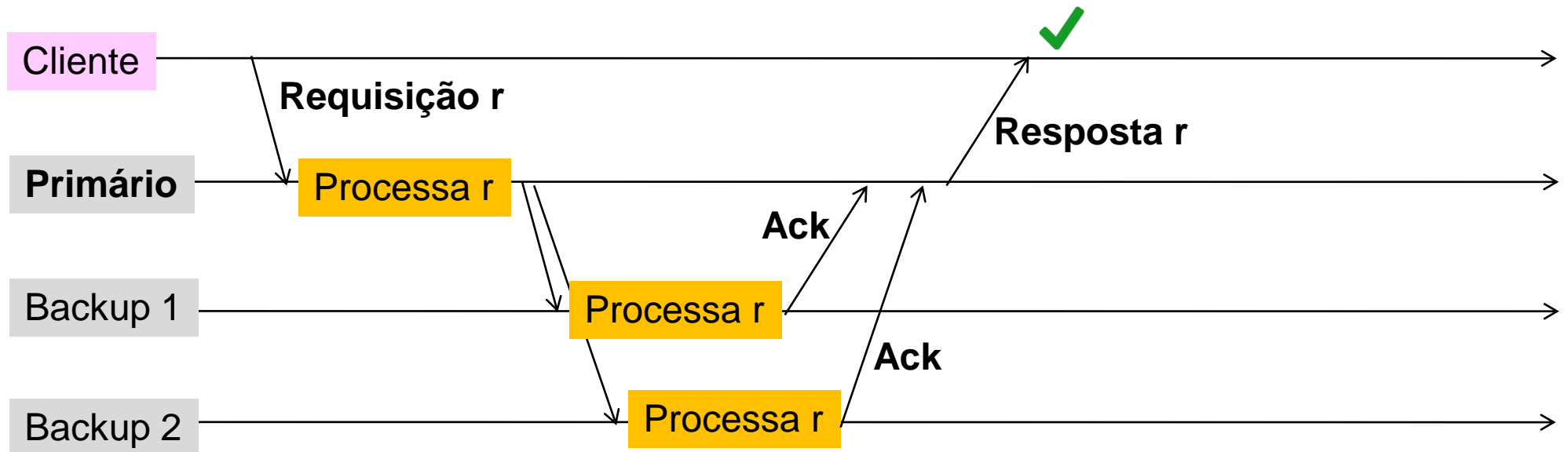
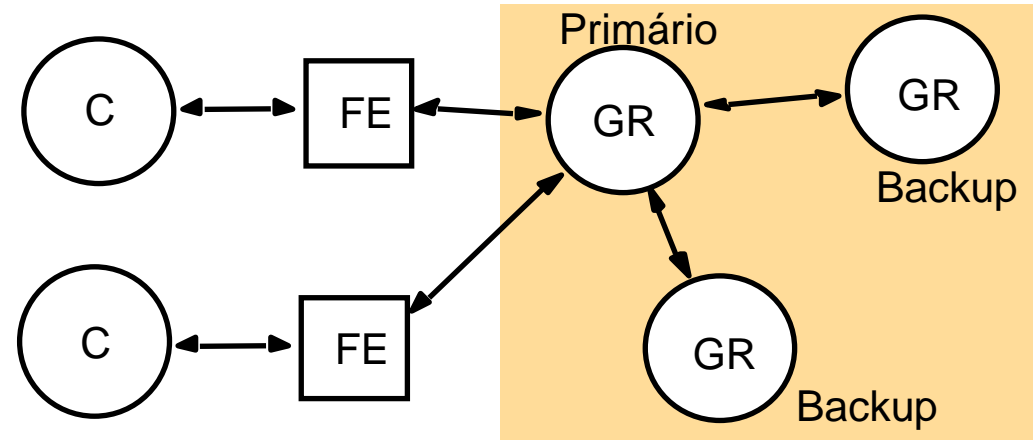
Replicação Ativa

- Todas as réplicas executam as requisições independentemente, mas de forma determinística
- Requisições são enviadas por *multicast* e pode ser necessário alguma garantia de ordem na entrega

Replicação Passiva



Replicação Passiva



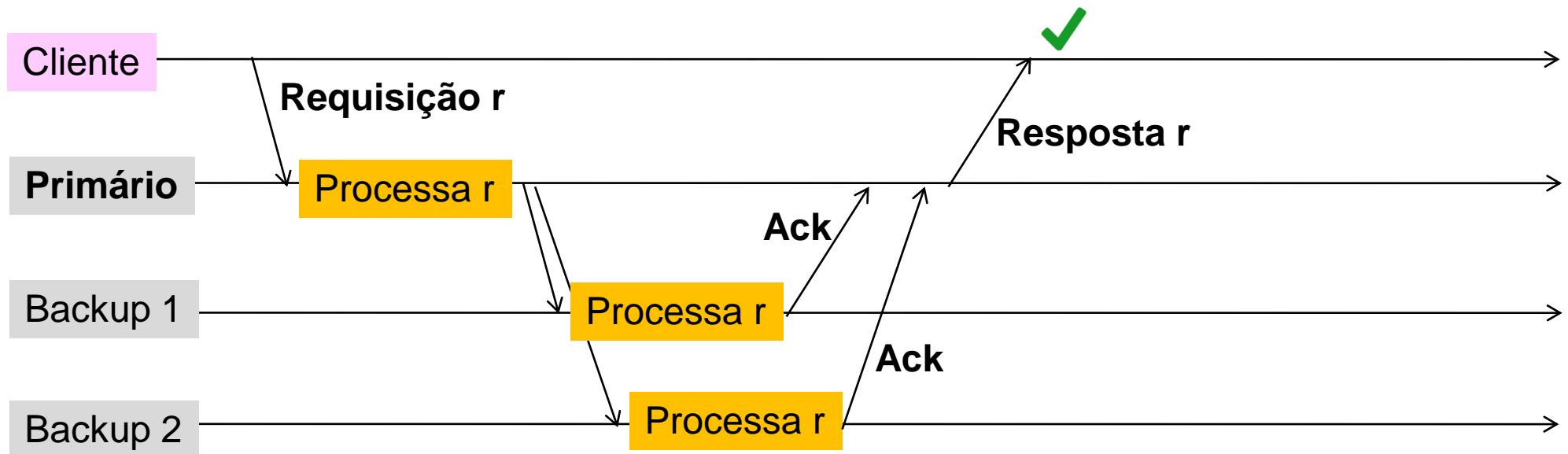
Replicação Passiva

Algumas considerações:

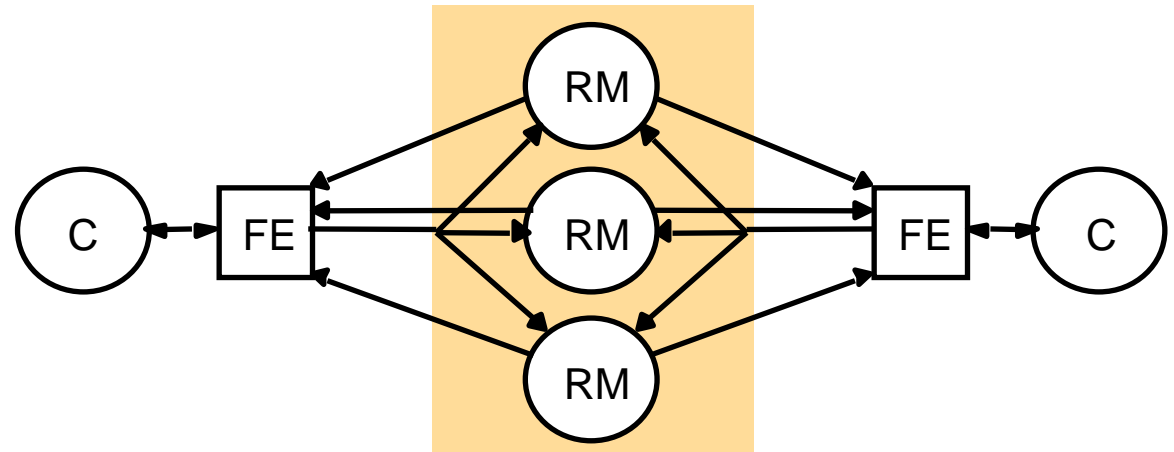
- Réplicas são equipadas com detectores de falha (para detectar falha do primário)
- Eleição de líder é necessária para eleger novo primário

Algumas otimizações:

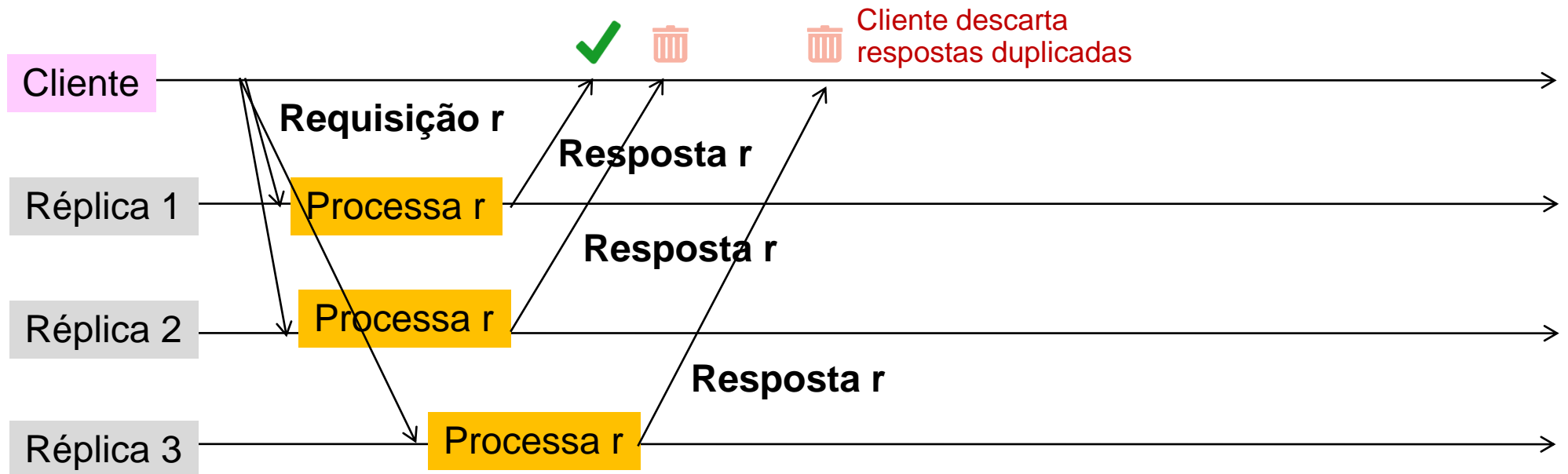
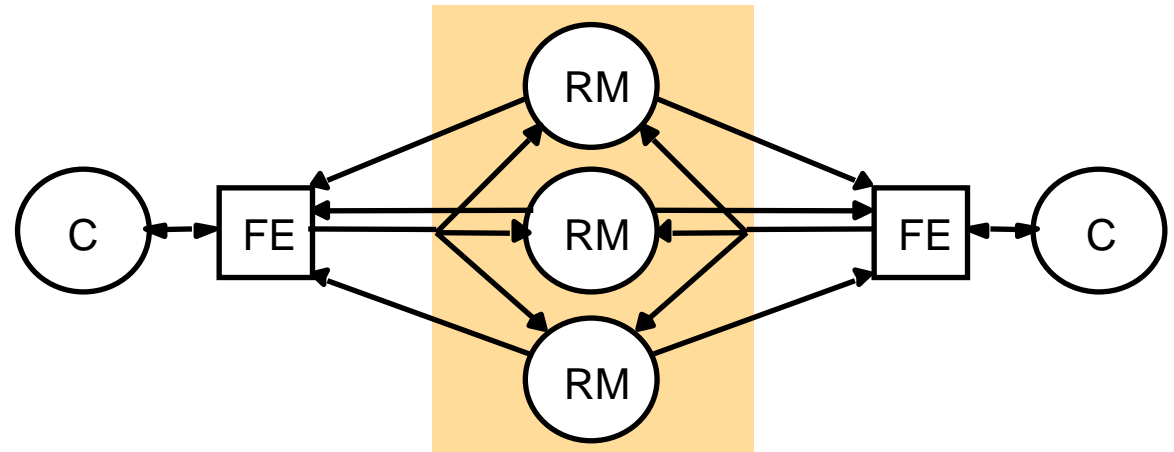
- Primário pode repassar comandos a serem processados ou o estado já atualizado para os backups



Replicação Ativa

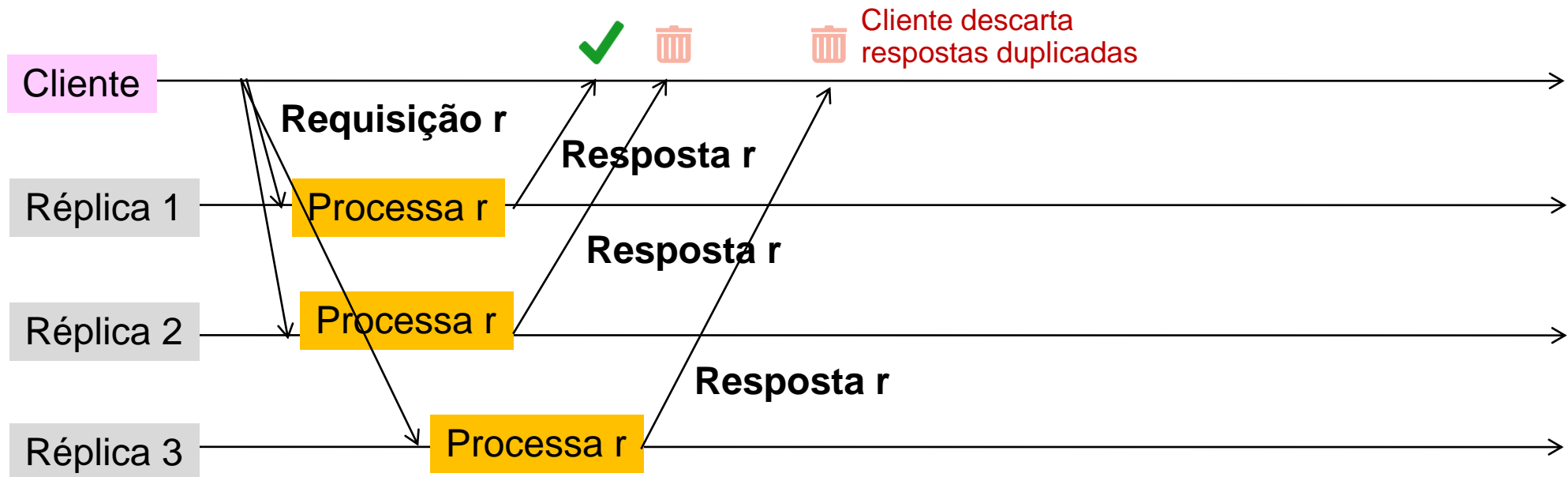
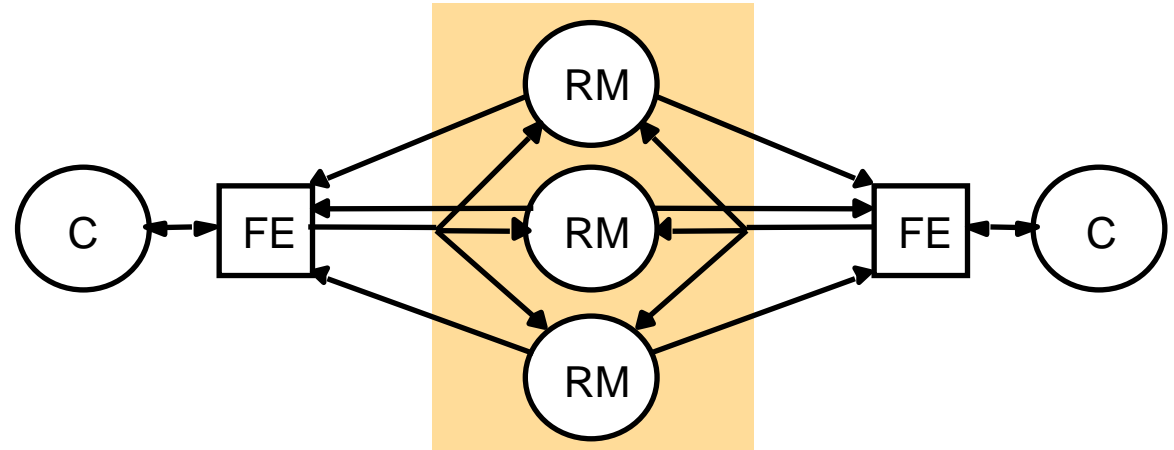


Replicação Ativa



Replicação Ativa

Como garantir que requisições de múltiplos clientes cheguem na mesma ordem em todas as réplicas?



Replicação Ativa – *State Machine Replication (SMR)*

Replicação Máquina de Estados (SMR)

- Um conjunto de servidores comportam-se como máquinas de estado replicadas
- Réplicas iniciam no mesmo estado inicial
- Operações do serviço são determinísticas
- Clientes enviam comandos para todas as réplicas, garantindo entrega e ordem total entre os comandos:
 - Réplicas corretas recebem cada comando
 - Se uma réplica processa o comando c_1 antes de c_2 , então nenhuma réplica processa c_2 antes de c_1

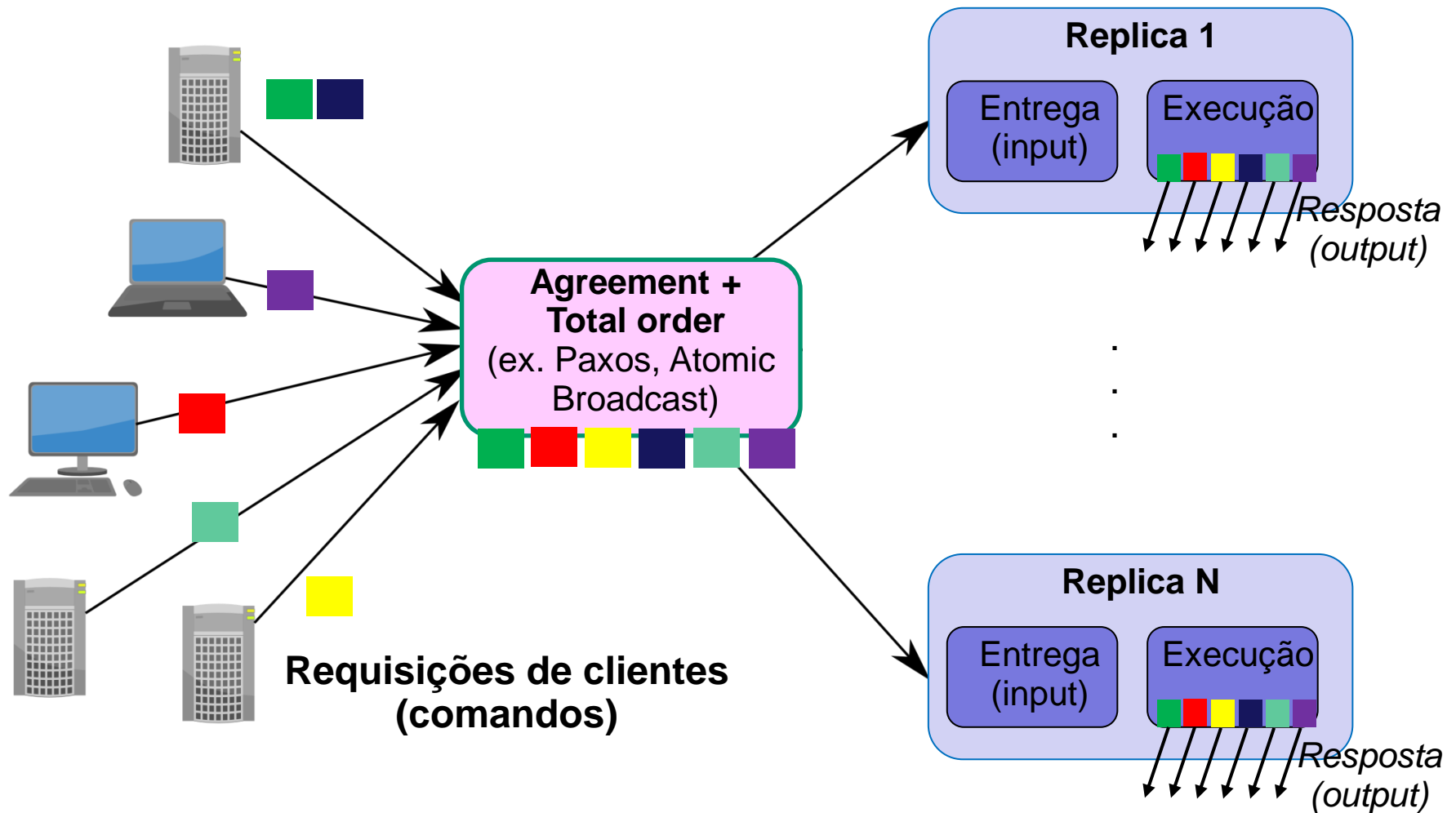


Schneider, F. B.
Implementing fault-tolerant services using the state machine approach: A tutorial
ACM Computing Surveys (CSUR), 1990



Castro, M., Liskov, B.
Practical byzantine fault tolerance
OSDI, 1999

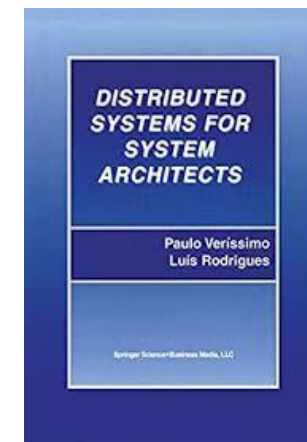
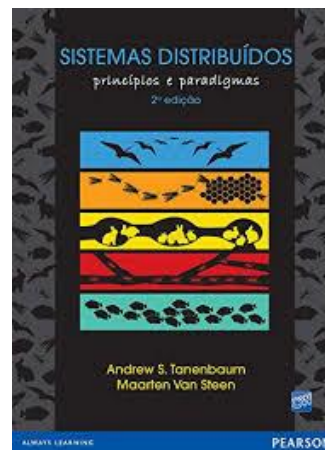
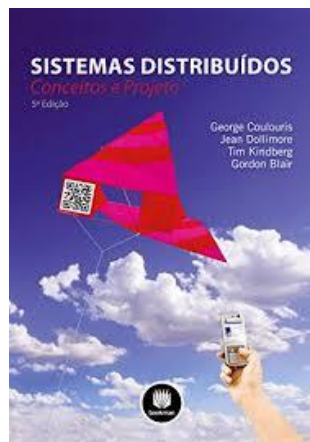
Replicação Ativa – *State Machine Replication (SMR)*



Referências

Parte destes slides são baseadas em material de aula dos livros:

- *Coulouris, George; Dollimore, Jean; Kindberg, Tim; Blair, Gordon. Sistemas Distribuídos: Conceitos e Projetos. Bookman; 5ª edição. 2013.*
- *Tanenbaum, Andrew S.; Van Steen, Maarten. Sistemas Distribuídos: Princípios e Paradigmas. 2007. Pearson Universidades; 2ª edição.*
- *Veríssimo, P.; Rodrigues, L. Distributed Systems for System Architects. Springer; 1ª edição. 2001.*



- *Imagens e clip arts diversos: <https://free-icon-rainbow.com/>, <https://www.gratispng.com/>*