

Banco de dados Distribuído

SISTEMAS DISTRIBUÍDOS – PROF. ÁLVARO COÊLHO

LARISSA DE BRITO E HENRIQUE SERRA

UESC – 2025.2

Necessidade

- Limites Físicos e de Custo
- Tempo de Inatividade
- Ponto Único de Falha

BDD

Banco de Dados Distribuído

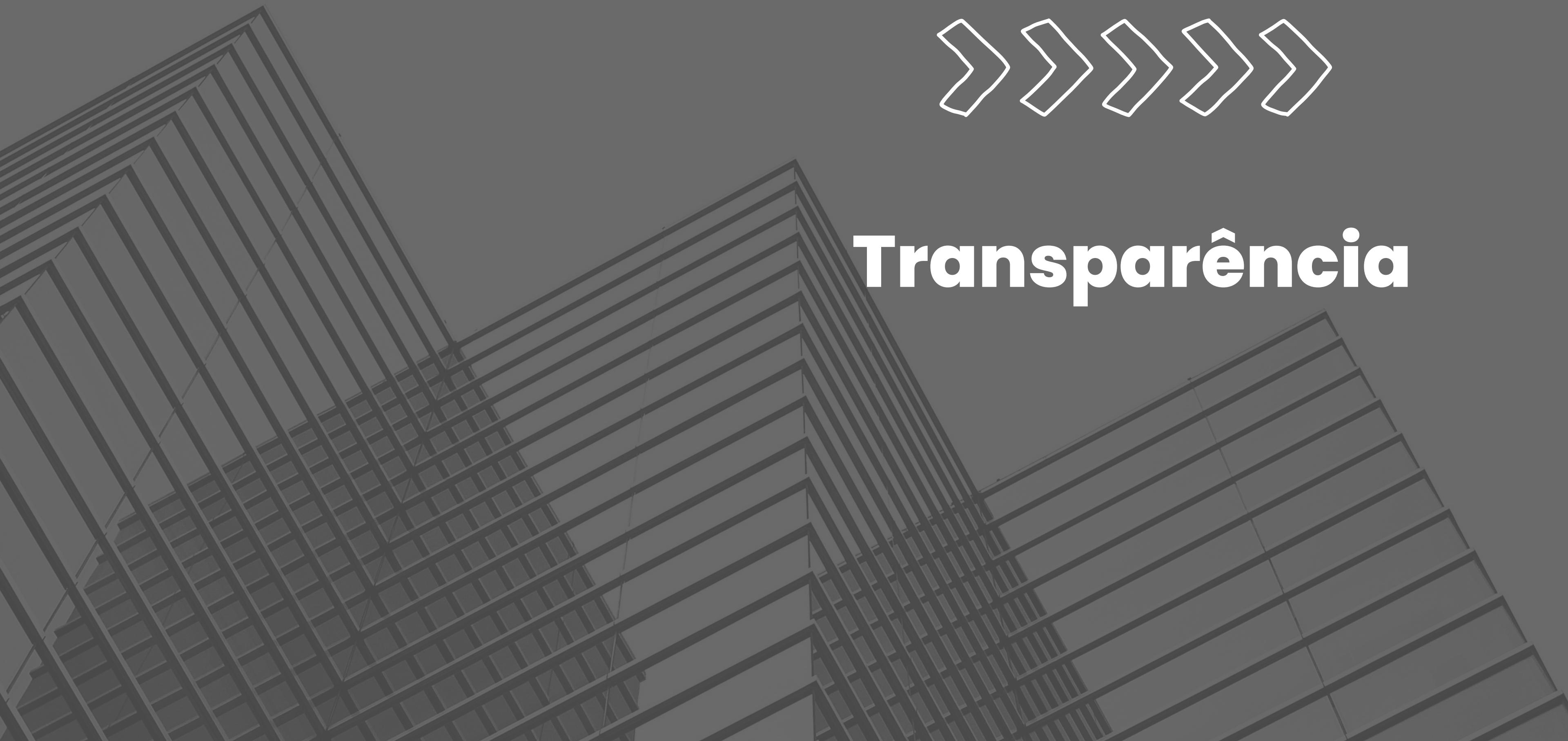
logicamente
inter-
relacionados

fisicamente
distribuídos

acoplamento
fraco

- Escalabilidade quase linear
- Alta Disponibilidade
- Tolerância a Falhas

**São Possíveis
Ganhos**



Transparência



Transparência

Localização e Acesso

O usuário não precisa saber onde um dado está fisicamente armazenado

Replicação

O usuário não têm conhecimento de que os dados podem estar em múltiplos nós

Transparência

Fragmentação

Uma tabela pode ser dividida em múltiplos fragmentos, com cada fragmento armazenado em um nó diferente

Falhas

O sistema deve ser capaz de detectar e se recuperar de falhas sem interromper a execução das transações

Embora a transparência total seja o ideal teórico

desenvolvedores de aplicações distribuídas de alto desempenho muitas vezes precisam ter consciência da topologia da rede e da localização dos dados para otimizar a performance

Teorema CAP

Consistência (C - Consistency)

Todos os nós veem os mesmos dados ao mesmo tempo

Disponibilidade (A - Availability)

O sistema está sempre disponível para processar requisições

Tolerância a Partições (P - Partition Tolerance)

O sistema continua a operar mesmo que ocorra uma quebra na comunicação

Sistemas CP (Consistência e Tolerância a Partições)

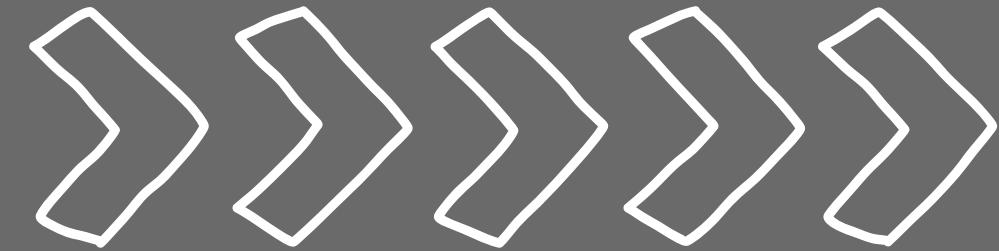
Quando uma partição ocorre, para garantir a consistência, o sistema pode ter que se tornar indisponível

Sistemas AP (Disponibilidade e Tolerância a Partições)

Quando uma partição ocorre, para não frustrar o usuário, o sistema continua a aceitar operações



Estratégias de Distribuição de Dados



Replicação

A replicação é o processo de criar e manter múltiplas cópias idênticas dos mesmos dados em nós diferentes da rede

Principais objetivos

Alta Disponibilidade e Tolerância a Falhas

Se um nó que hospeda uma réplica de dados falhar haverá uma cópia disponível

Escalabilidade de Leitura

Aumenta a vazão de leitura, permitindo um número maior de usuários simultâneos

Redução de Latência

As réplicas podem ser posicionadas mais próximas dos usuários finais

Replicação

Síncrona

Os dados são constantemente copiados para o servidor principal e para todos os servidores de réplica simultaneamente

Assíncrona

Os dados são copiados primeiro para o servidor principal e só depois copiados para servidores de réplica em lotes

Replicação

Técnica de Replicação

Tabela completa

Incrementais
baseadas em
chaves

Baseada em log

Líder-Único

Todas as operações de escrita (INSERT, UPDATE, DELETE) são obrigatoriamente direcionadas a um único nó designado como o líder

Após processar uma escrita, o mestre é responsável por registrar as alterações e propagá-las para um ou mais nós seguidores

Um ponto único de falha e carga de escrita concentrada

Multilíder

Múltiplos nós (ou todos os nós, em alguns sistemas) são designados como mestres e podem aceitar operações de escrita

Quando um mestre processa uma escrita, ele a replica para os outros mestres no sistema, garantindo que todos os nós eventualmente converjam para o mesmo estado

A desvantagem fundamental e mais complexa do modelo mestre-mestre é a resolução de conflitos de escrita

Replicação

Tipos de Conflito

Atualização

Singularidade

Exclusão

Replicação

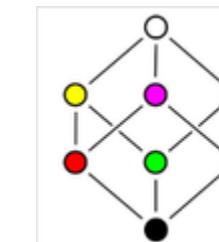
Como Resolver?

Last Write Wins

CRDTs (Conflict-free Replicated Data Types)

PN-Counter

Observe-Remove Set



About CRDTs • Conflict-free Replicated Data Types

Resources and community around CRDT technology
— papers, blog posts, code and more.

Conflict-free Replicated Data Types

Fragmentação

É a técnica de dividir um banco de dados logicamente coeso em partes menores e mais gerenciáveis, chamadas de fragmentos (shards), e distribuir esses fragmentos por múltiplos servidores

A combinação de replicação e fragmentação é uma prática comum e poderosa

Horizontal

Neste método, uma tabela é dividida por linhas (tuplas). Por exemplo, uma tabela de Clientes pode ser fragmentada com base na região geográfica

A eficácia da fragmentação horizontal depende criticamente da escolha da chave de fragmentação

Consultas que precisam acessar dados de múltiplos fragmentos são complexas, ineficientes e lentas

Vertical

Neste método, uma tabela é dividida por colunas (atributos). Cada fragmento contém todas as linhas da tabela original, mas apenas um subconjunto de suas colunas

Para permitir a reconstrução de uma linha completa, um identificador único, como a chave primária, deve ser replicado em todos os fragmentos verticais

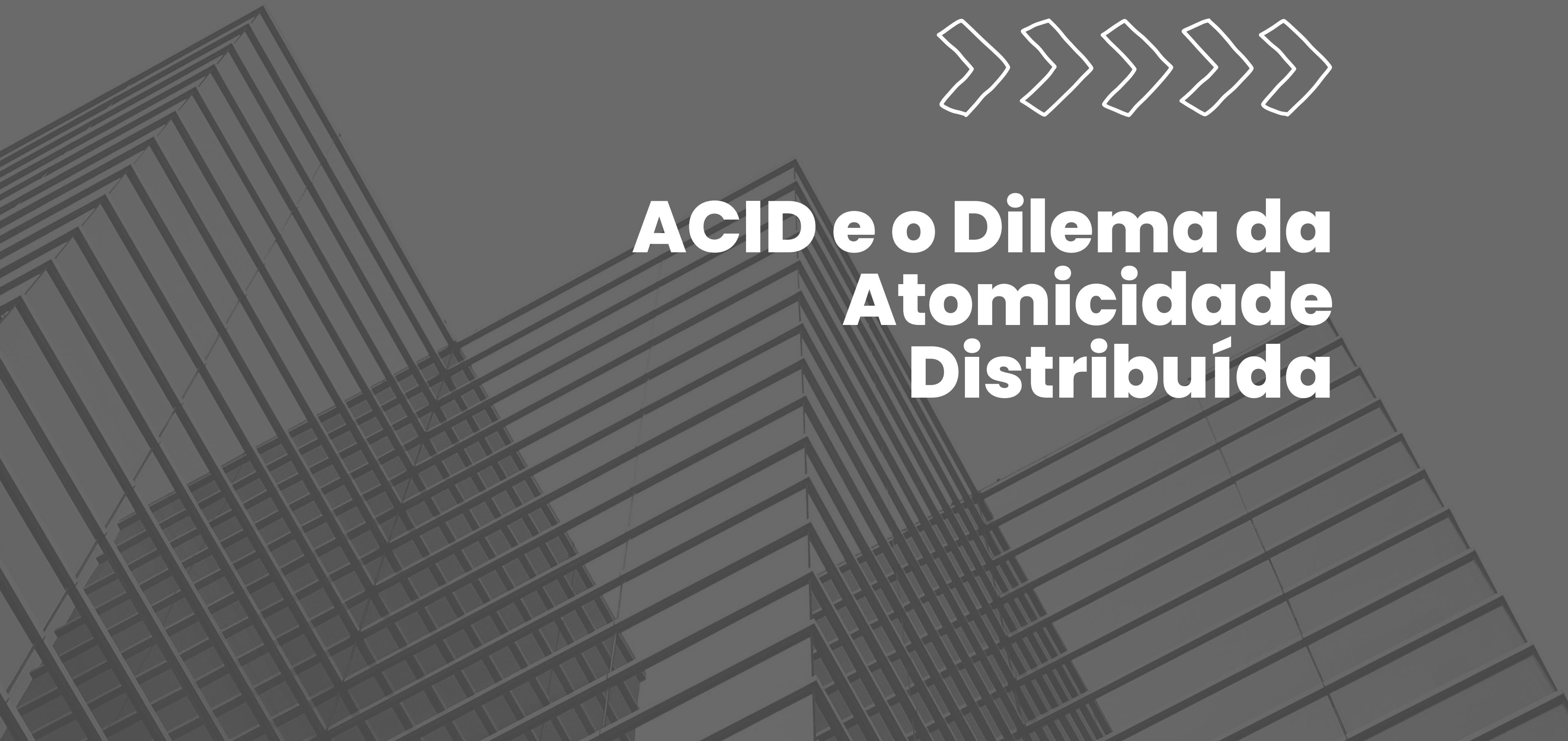
A fragmentação vertical é útil em cenários onde diferentes aplicações ou consultas acessam subconjuntos distintos de colunas

Critério para escolha

A análise dos padrões de consulta é um pré-requisito crítico para projetar um esquema de fragmentação eficaz

"obter todos os dados do cliente X"

"calcular a média de uma única coluna para todos os usuários"



ACID e o Dilema da Atomicidade Distribuída



ACID

Relembrando...

Atomicidade

Consistência

Isolamento

Durabilidade

ACID

Relembrando...

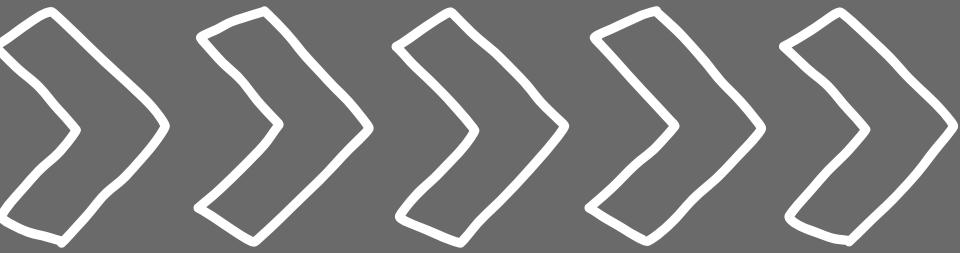
Atomicidade

Consistência

Isolamento

Durabilidade

*Em um sistema distribuído, o desafio central reside na **atomicidade***



Protocolo de Commit de Duas Fases (2PC)

2PC

O Coordenador envia uma mensagem de “prepare” para todos os Participantes envolvidos na transação

Ao receber a mensagem “prepare”, cada Participante determina se pode garantir o commit da sua parte da transação

2PC

Commit

Se o Coordenador receber votos “ok” de todos os Participantes, ele decide confirmar a transação. Ele registra essa decisão em seu próprio log durável e, em seguida, envia uma mensagem de commit para todos os Participantes

2PC

Abort

Se o Coordenador receber pelo menos um “não consigo”, ou se algum Participante não responder dentro de um tempo limite, ele decide abortar a transação. Ele registra essa decisão e envia uma mensagem de abort para todos os Participantes

Não importa o que aconteça os recursos ficam reservados até receber a decisão final



Protocolo de Commit de Três Fases (3PC)

3PC

Uma Solução Não-Bloqueante

3PC

O Coordenador envia uma mensagem CanCommit para todos os Participantes, que respondem "Sim" ou "Não" com base em sua capacidade de realizar a transação

3PC

PreCommit

Se o Coordenador recebeu "Sim" de todos os
Participantes na Fase 1, ele envia uma mensagem de
PreCommit para todos eles

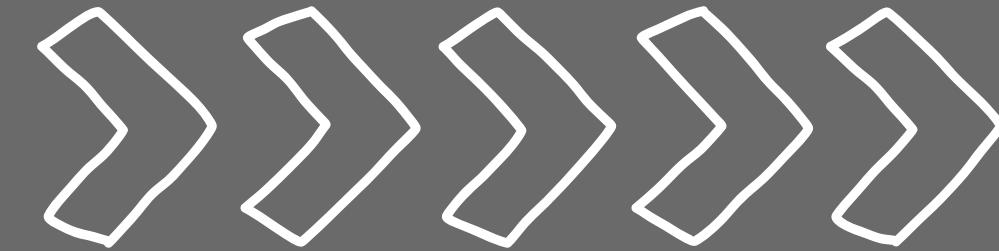
3PC

Após receber a confirmação do PreCommit de todos os participantes, o Coordenador envia a mensagem final de DoCommit, e os participantes finalizam a transação

3PC

O 3PC é um protocolo não-bloqueante, mas não é tolerante a partições de rede

PAXOS e RAFT



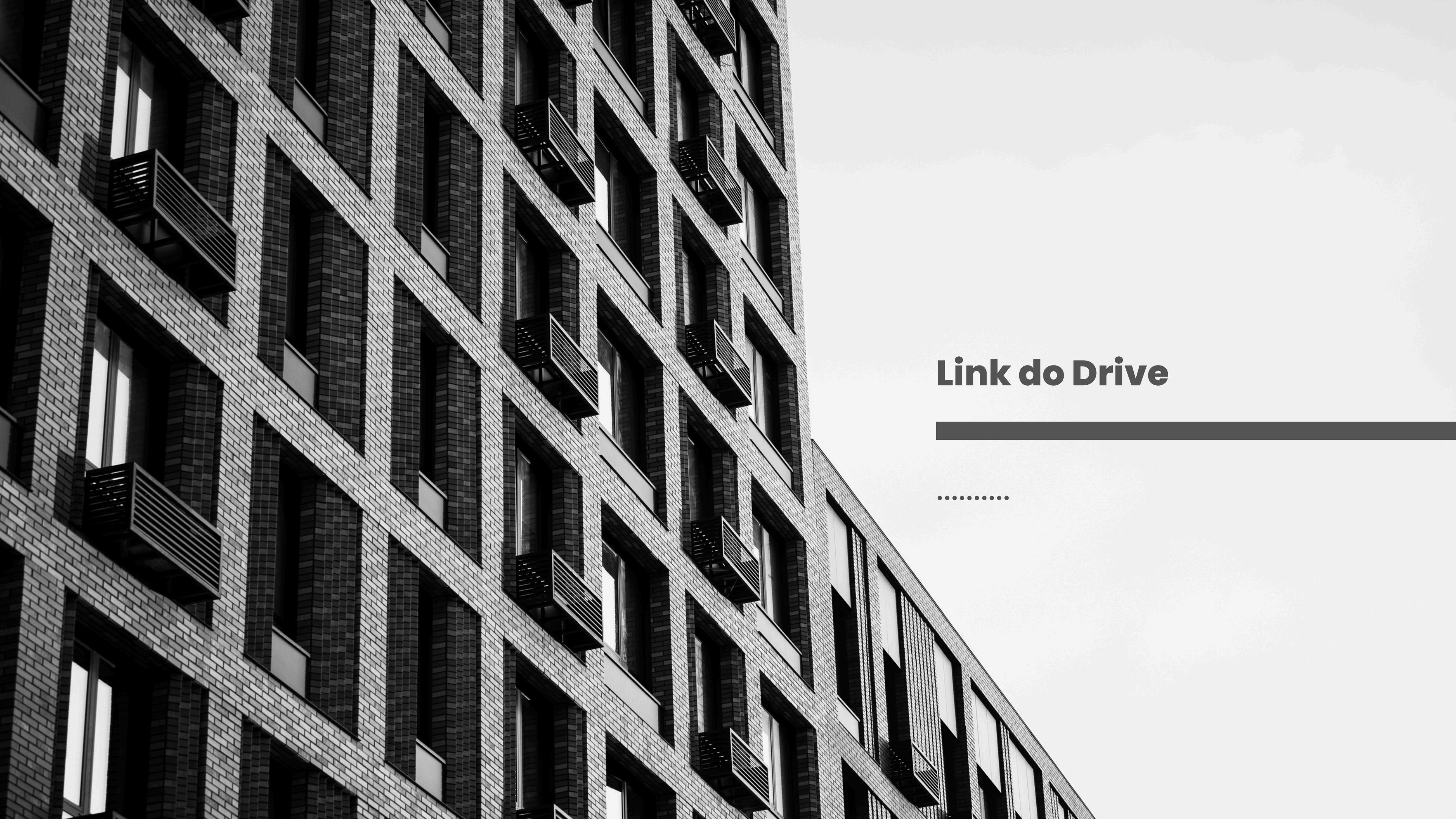
Raft

Eleição de Líder

Replicação do Log

Raft

O Raft garante a segurança mantendo sempre o log mais atualizado sobre o líder. Se um seguidor descobrir um log mais recente de outro nó, ele reverte para o estado de seguidor (se já não for) e segue o novo líder

A black and white photograph of a modern residential building. The building features a repetitive pattern of rectangular windows and balconies. The balconies are supported by thin metal railings. The facade appears to be made of small, square tiles or bricks. The lighting creates strong shadows, emphasizing the geometric nature of the building's design.

Link do Drive

.....