

Banco de Dados Relacionais e Não Relacionais

Prof. Henrique Batista da Silva



Introdução à banco de dados

Chave-Valor

Modelo de votação

Vimos no início a especificação do nosso problema para cadastro de álbuns de músicas.

E vimos que o modelo relacional pode não ser o mais adequado para as características deste tipo de problema.

Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo de votação

Agora suponha que gostaríamos de implementar um serviço de votação (exemplo apresentado anteriormente). Teremos uma entidade com o **título** da eleição, uma outra com os **candidatos**, e uma última com os **votos**.

Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo de votação

Conforme vimos anteriormente, o problema poderia ser modelado para um banco relacional. No entanto, precisamos alta **disponibilidade** e **performance** (suponha um sistema online durante um programa ao vivo).

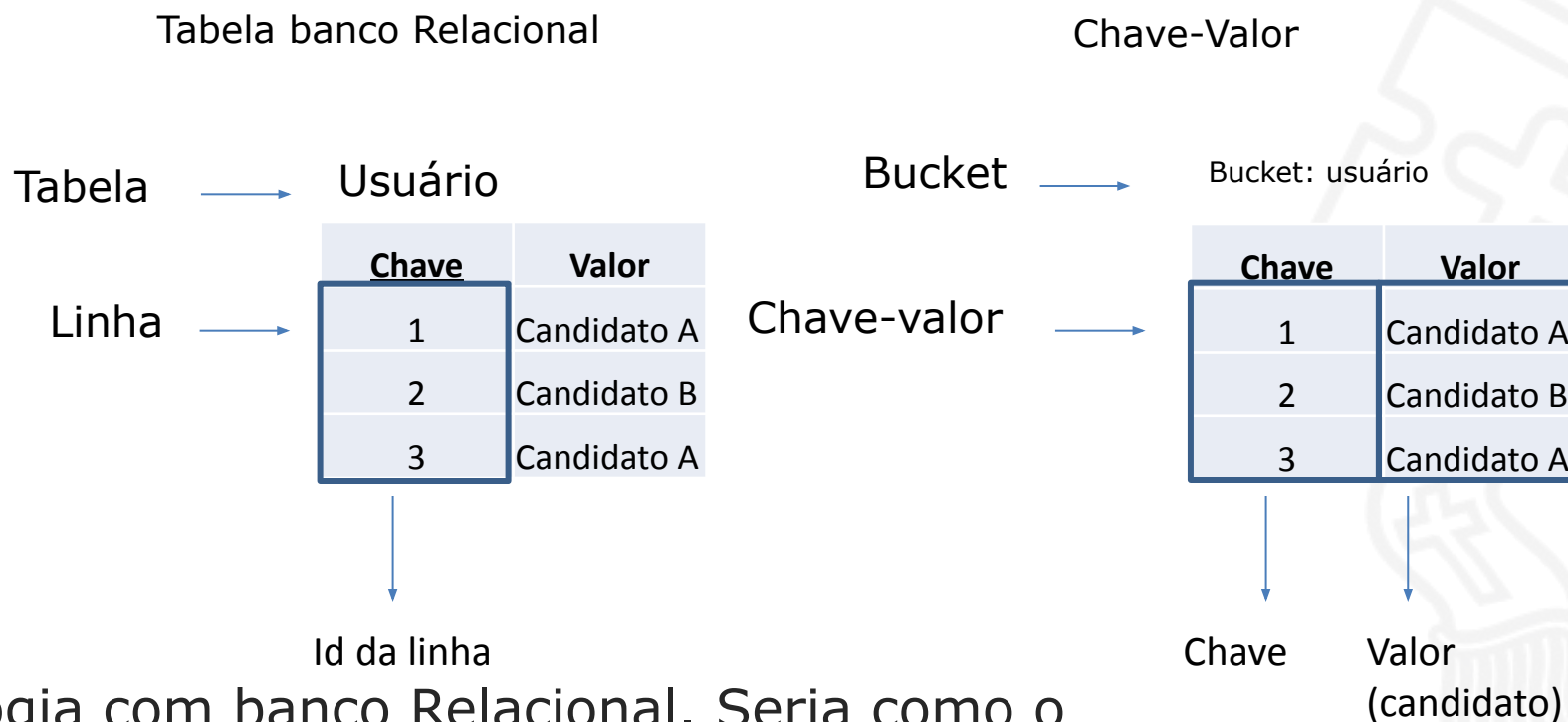
Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo chave-valor

Para o problema de votação, é necessário armazenar dados do **usuário** e em qual **artista** ele votou. Problema relativamente simples, ideal para banco de dados do tipo chave-valor (um banco também muito simples).

Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo chave-valor



Analogia com banco Relacional. Seria como o banco inteiro fosse uma única tabela.

Modelo chave-valor

Por serem muito rápidos, este tipo de banco pode usado em diversas situações.

Ex: compartilhamento de dados de seção de usuário entre vários servidores de aplicação.

Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo chave-valor

Iremos analisar um exemplo utilizando o banco chave-valor para a aplicação **Redis**.

Outro banco chave-valor é o Riak. São soluções *open source* e estão disponíveis para Linux e MacOS (Riak).

O Redis possui uma versão para Windows.

Modelo chave-valor

O banco pode armazenar qualquer tipo de dados.

Caracteriza-se por ser um banco de dados distribuído de alta disponibilidade.

Modelo chave-valor

Redis significa REmote DIctionary Server e foi criado por Salvatore Sanfilippo. Armazena seus dados em memória (extremamente rápido para escrita e leitura), mas possui opção para persistir em disco.

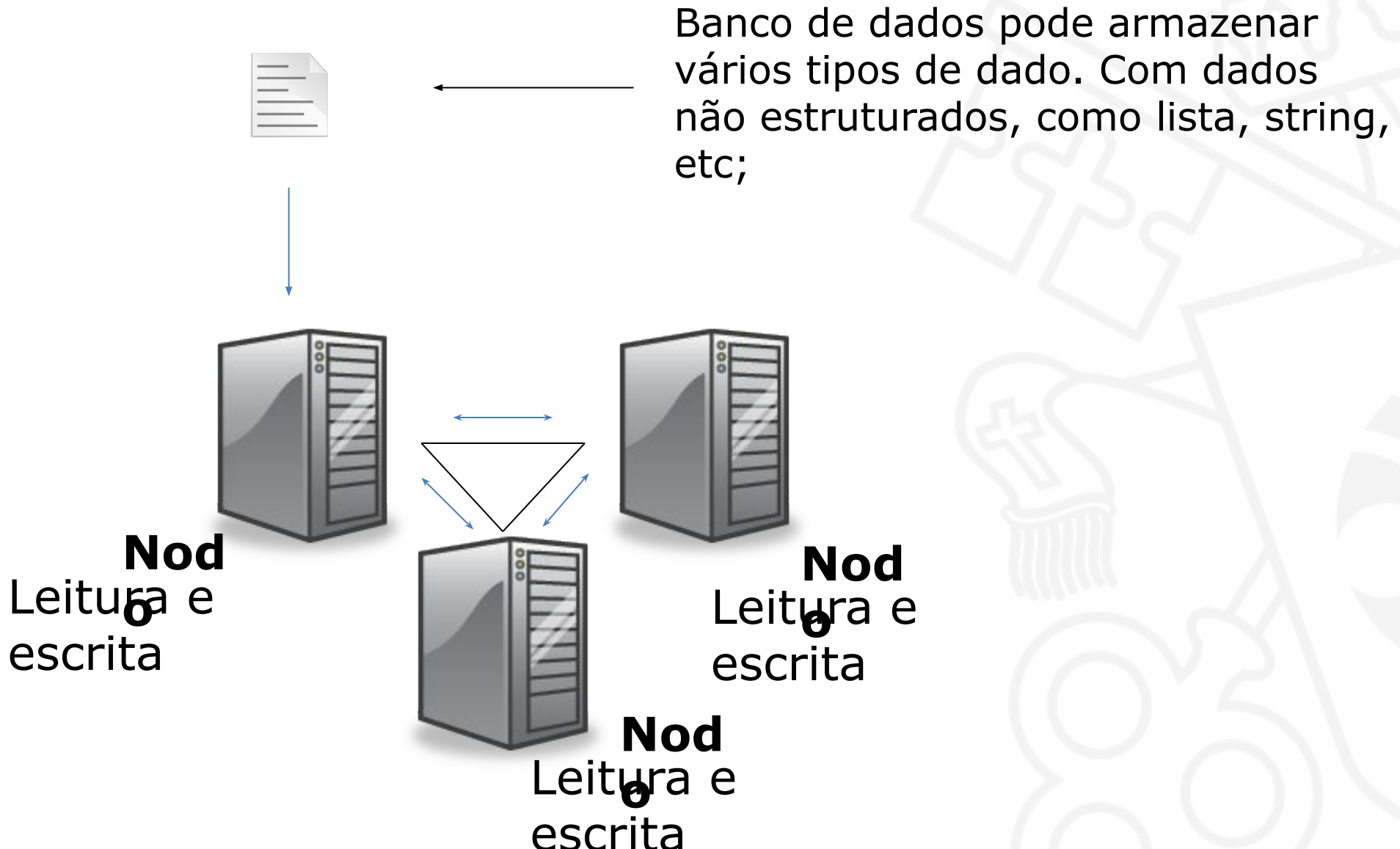
Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo chave-valor

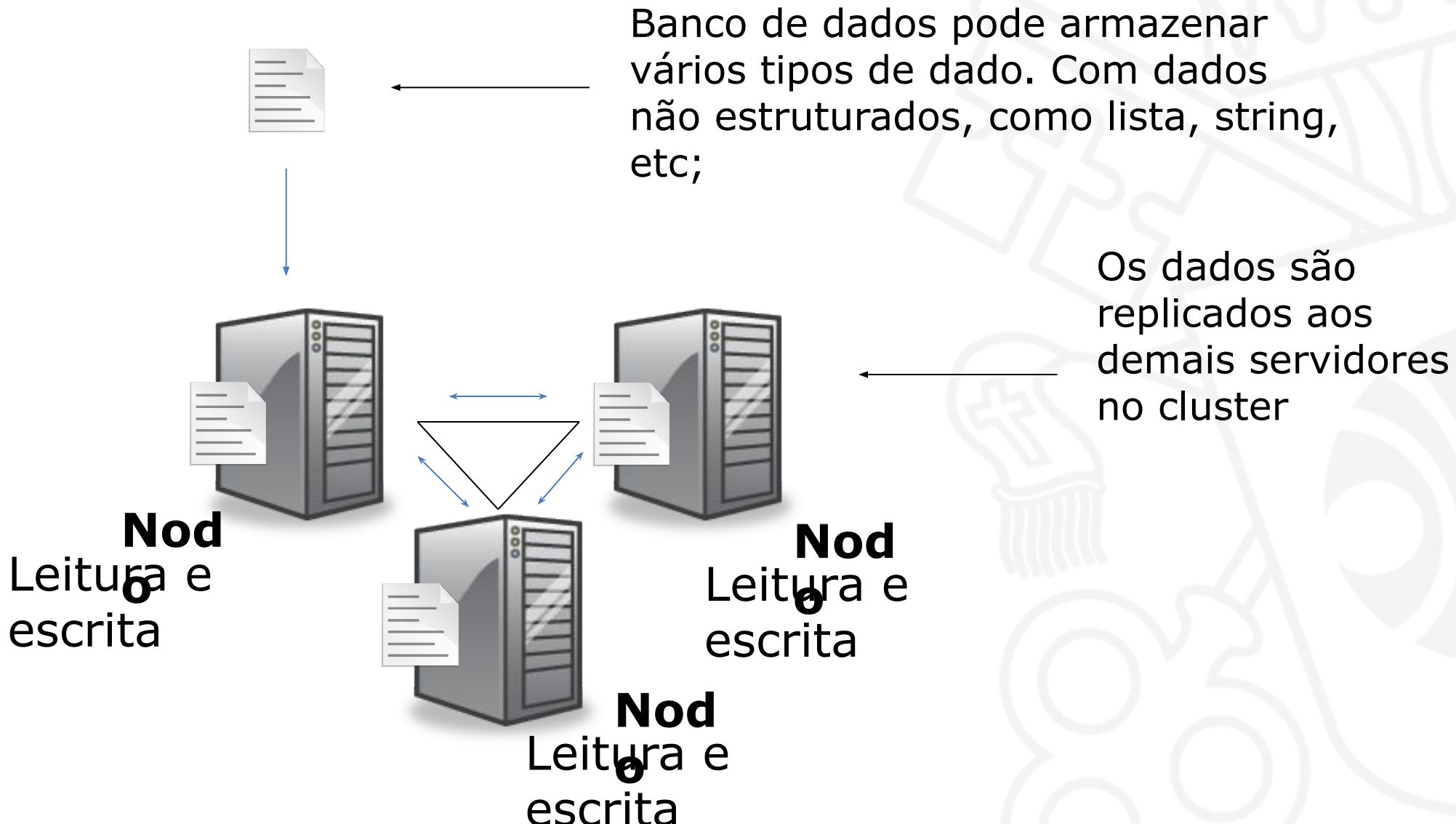
Seus comandos são atômicos (single thread), enquanto um comando é executado, nenhuma outro será.

Referências: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Modelo chave-valor

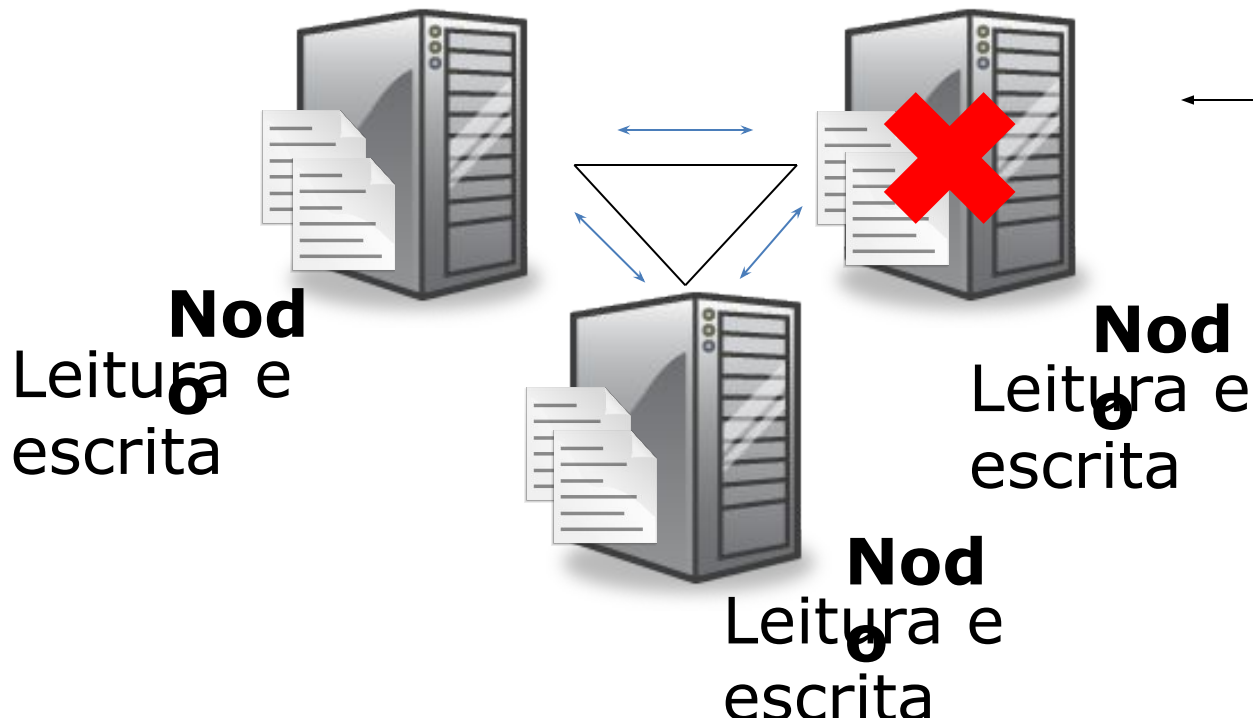


Modelo chave-valor



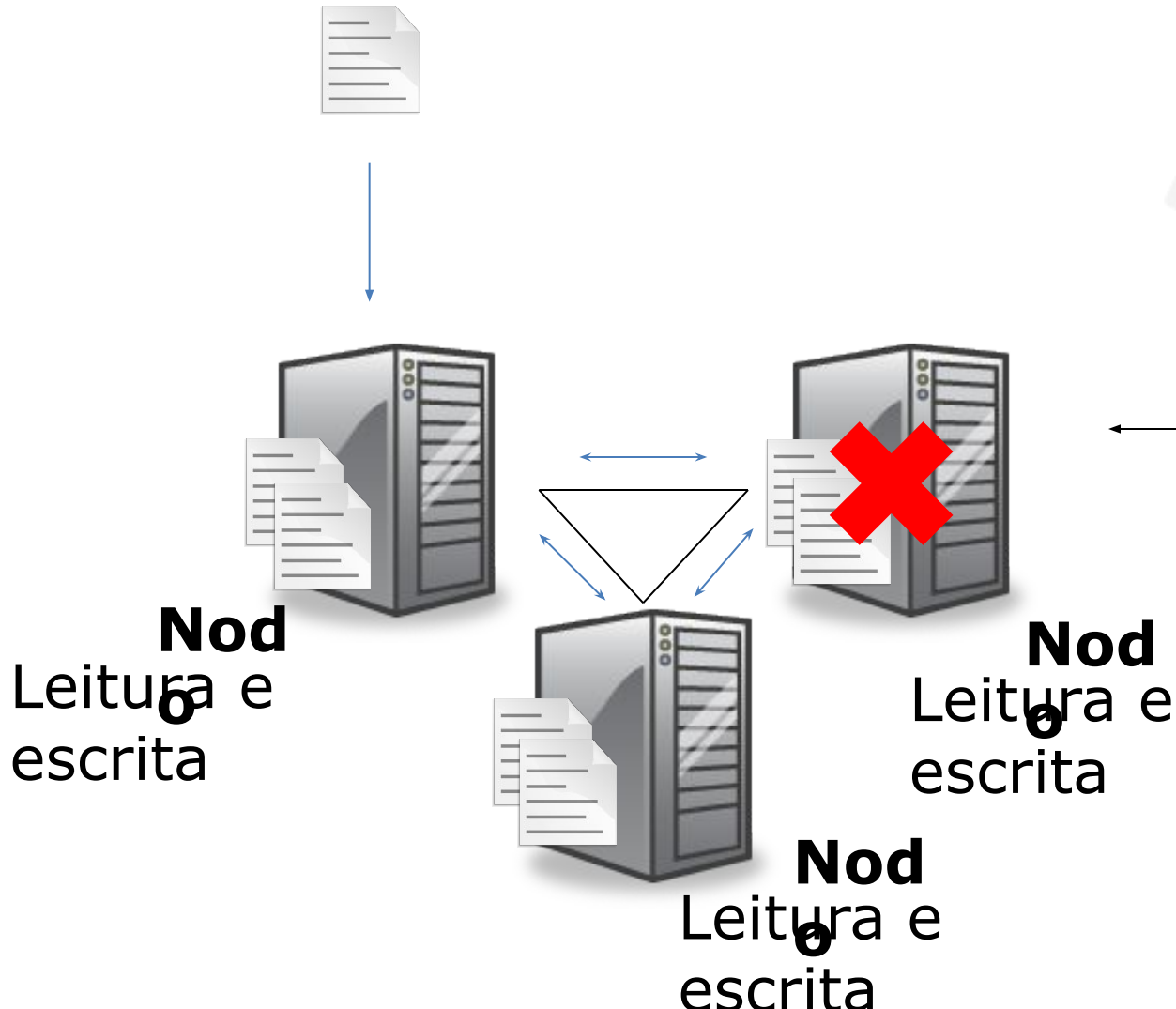
Modelo chave-valor

Assim, os dados estão sempre disponíveis mesmo em caso de falhas de um dos servidores



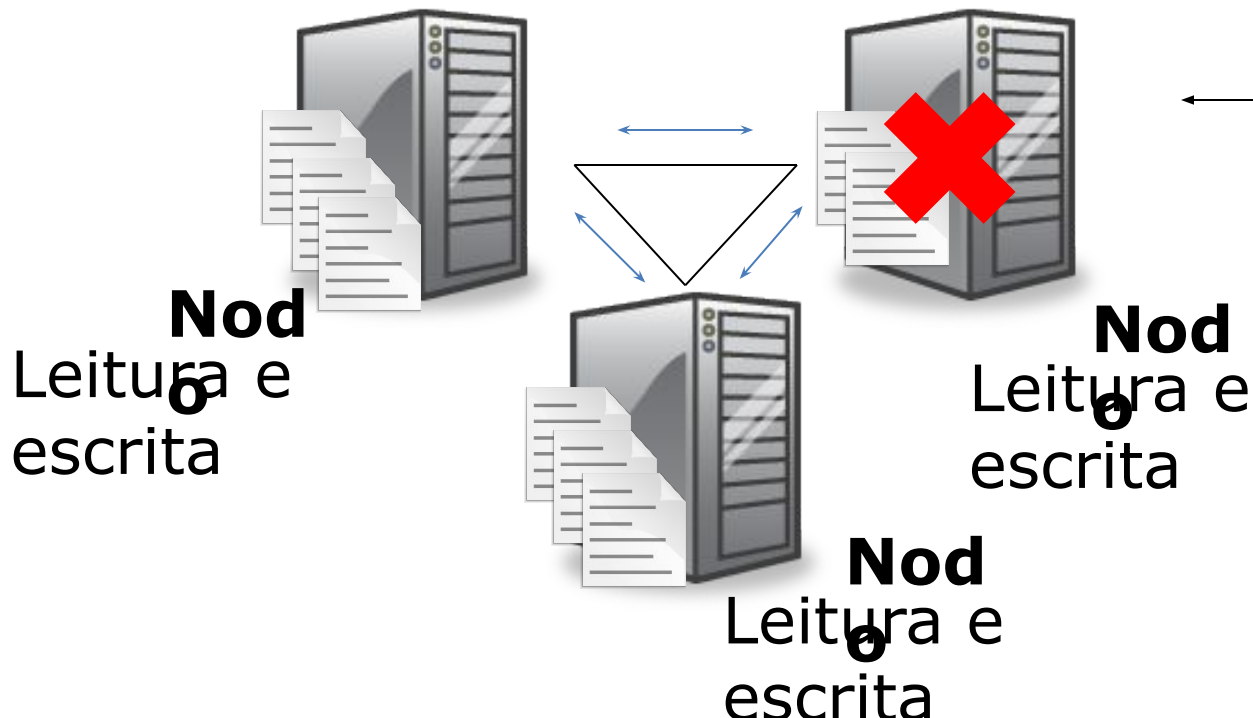
Os dados são replicados aos demais servidores no cluster

Modelo chave-valor



Os dados são replicados apenas aos servidores disponíveis sem que o serviço fique indisponível

Modelo chave-valor

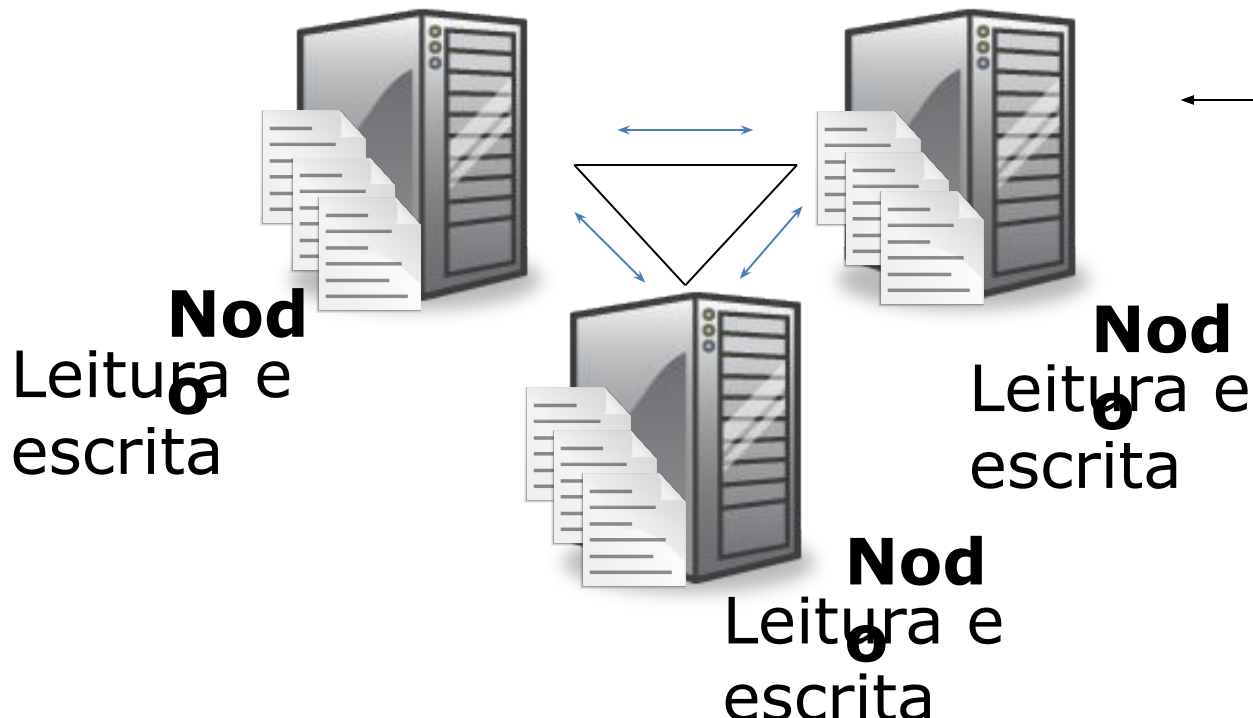


Os dados são replicados apenas aos servidores disponíveis sem que o serviço fique indisponível

Modelo chave-valor

Quando o servidor volta a ficar disponível, os dados são copiados para ele.

Observe que assim, novos nodos podem ser adicionados para aumentar a robustez do cluster.



Modelo chave-valor



Empresas como BestBuy, Betzold, Dubzille, ideel, e Shopzilla usam banco chave-valor (Riak) para armazenamento de produtos e catálogos.

Referências: <http://basho.com/>

Modelo chave-valor

Algumas empresas que utilizam
Redis:

American Express, Verizon, Microsoft,
DreamWorks, etc..

Referências: <https://redislabs.com/>

Manipulando dados no Redis



Manipulando dados no Redis

Instalação do Redis para Windows:

Baixar a versão 3.0.504

<https://github.com/MicrosoftArchive/redis/releases>

Manipulando dados no Redis

Após a instalação o Redis fornece um console (Redis CLI) para manipulação dos dados no banco.

Abra o prompt de comando e acesse a pasta de instalação do Redis e execute o comando:

```
cd C:\Program Files\Redis  
.\redis-cli.exe
```

Criando objetos no Redis

Para criar objetos no Redis, precisamos especificar dois valores: **chave**, **valor**.

Chave-Valor

Votacao

<u>Chave</u>	Valor
--------------	-------

Criando objetos no Redis

Chave-Valor

Votacao

<u>Chave</u>	Valor
usuarioA	candidato1

Chave-valor →

Chave: login do usuário
Valor: o voto do usuário

No Redis utilizamos o comando **set** para especificar a chave e o valor da mesma no banco.

```
> set usuarioA "candidato1"
```

↑
chave

↑
valor

O valor é uma string (cadeia de caractere).

Criando objetos no Redis

Chave-Valor

Votacao

Chave-valor →

<u>Chave</u>	Valor
usuarioA	candidato1

Chave: login do usuário
Valor: o voto do usuário

No Redis utilizamos o comando **set** para especificar a chave e o valor da mesma no banco.

Esta operação é semelhante a um comando de insert no SQL (SGBDR):

```
INSERT  
INTO TABELA (chave, valor)  
VALUES ("usuarioA", "candidato1");
```

Criando objetos no Redis

Chave-Valor

Votacao

Chave-valor →

<u>Chave</u>	Valor
usuarioA	candidato1

Chave: login do usuário
Valor: o voto do usuário

Para recuperar o registros, utilizamos o método get

```
> get usuarioA  
> "candidato1"
```

← Resultado da consulta

Esta operação é semelhante a um comando de select no SQL (SGBDR):

```
SELECT valor  
FROM TABELA  
WHERE chave = "usuarioA"
```

Criando objetos no Redis

Chave-Valor

Votacao

Chave-valor →

<u>Chave</u>	<u>Valor</u>
usuarioA	candidato1

Chave: login do usuário
Valor: o voto do usuário

Para recuperar o registros, utilizamos o método get

```
> get usuarioA  
> "candidato1"
```

Observe que no Redis (ou qualquer banco chave-valor) a busca é feita pela **chave** e o banco retornar o seu **valor** correspondente. As chaves nunca possuem valores repetidos no banco.

Criando objetos no Redis

Chave-Valor

Votacao

Chave-valor →

<u>Chave</u>	Valor
usuarioA	candidato1

SELECT chave
FROM TABELA

Chave: login do usuário

Valor: o voto do usuário

É possível exibir também o valor de todas as chaves do banco

```
> keys *
```

Esta operação é semelhante a um comando de select no SQL (SGBDR):

Manipulando objetos no Redis

Neste tipo de banco, normalmente não se diferencia operações de inserção e de atualização

A atualização é inserir (método **set**) novamente um valor para uma chave.

O que vale é o último valor.

Removendo registros no Redis

Chave-Valor

Votacao

Chave-valor →

<u>Chave</u>	Valor
usuarioA	candidato1

Chave: login do usuário

Valor: o voto do usuário

O Redis também permite deletar registros no banco de dados.

Removendo registros no Redis

Chave-Valor

Votacao

<u>Chave</u>	Valor
--------------	-------

Chave: login do usuário
Valor: o voto do usuário

O Redis também permite deletar registros no banco de dados.

```
> del usuarioA  
> (integer) 1
```

Equivalente em SQL

```
DELETE  
FROM TABELA  
WHERE chave = "usuarioA"
```




Quando usar e não usar

Situações apropriadas para o uso

Armazenamento de informações de sessão: Geralmente sessões Web são únicas (possuem *sessionid*). Assim, ao utilizar sistema chave-valor (ao invés de SGBDR), tudo que está relacionado à sessão é armazenado em um único objeto (única operação para armazenar e recuperar).

Situações apropriadas para o uso

Perfis de usuários (preferências): Como todo usuário tem um *userid* e *username*, além de idioma, fuso horário, produtos que o usuário tem acesso, etc. Tudo pode ser colocado em um único objeto. E assim recuperados com uma única operação.

Situações apropriadas para o uso

Carrinhos de compra: E-commerce possuem carrinhos de compra para cada *userid*. Todas as informações de compra podem ser colocadas no valor da chave *userid* (par chave-valor).

Situações apropriadas para o uso

Publicação de mensagem: usado para sistemas de mensagens para assinantes. Usa de chaves com TTL (Time to Live – tempo de vida). Estas mensagens são excluídas automaticamente após um tempo, evitando sobrecarga do banco com mensagens desnecessárias.

Situações apropriadas para o uso

Teste o uso do TTI no Redis:

```
> set A "candidato1"
```

```
OK
```

```
> expire A 5
```

← Chave A irá expirar em 5 segundos

```
(integer) 1
```

```
> get A
```

```
"candidato1"
```

```
> get A
```

← após 5 segundos

```
(nil)
```

Referências: <https://redis.io/topics/data-types-intro>

Referências: <https://aws.amazon.com/pt/elasticache/what-is-redis/>

Situações para não usar chave-valor

Relacionamento entre dados: Não é apropriado quando é necessário saber o relacionamento de diferentes conjuntos de dados utilizando as chaves para correlação.

Situações para não usar chave-valor

Consulta por dados: Necessidade de pesquisar as chaves baseado nos seus valores (par chave-valor). Não há como inspecionar pelo banco.

Redis e Python (material extra)

Modelo chave-valor

Para implementar a solução do problema de votação, descrito anteriormente, temos um exemplo prático em python usando o banco de dados chave-valor Redis (Linux).

Modelo chave-valor

Página oficial do Redis:

<https://redis.io/>

Instalação do redis-py: `pip install redis`

<https://pypi.python.org/pypi/redis>

Modelo chave-valor

```
import redis
```

← Importação da biblioteca do redis-py

```
r = redis.StrictRedis(host='localhost', port=6379,  
charset="utf-8", decode_responses=True, db=0)
```

← Conexão com o banco de dados (deve-se iniciar o serviço antes na pasta de instalação do redis)

```
r.set('A', 'candidato1')  
r.set('B', 'candidato1')  
r.set('C', 'candidato2')  
r.set('D', 'candidato1')  
r.set('E', 'candidato2')  
r.set('F', 'candidato2')  
r.set('G', 'candidato1')  
r.set('H', 'candidato1')  
r.set('I', 'candidato1')
```

← Inserção de chave-valor (eleitor 'A' votou no 'candidato1'). Neste exemplo o eleitor é a chave e o candidato o valor.

```
value = r.get('C')  
print(value)
```

← Obtendo o candidato em que o eleitor 'C' votou (obtendo o "valor" da chave 'C')

Modelo chave-valor

Salve o programa (votacao.py) em um diretório e execute-o para testar a verificação de qual candidato cada eleitor votou.

```
>python votacao.py
```

Modelo chave-valor

```
import redis

r = redis.StrictRedis(host='localhost', port=6379,
charset="utf-8", decode_responses=True, db=0)

a = [0, 1, 1, 2, 3, 5, 8, 13]
b = [0, 1, 1, 2, 3]

r.set('A', a)
r.set('B', b)

value = r.get('A')
print(value)
```

Lembrem-se que em um banco de dados chave-valor, a coluna "valor" pode ser composta por qualquer tipo de objeto.

Portanto, o programa ao lado, que insere uma lista de inteiros no banco, deve funcionar.

Salva e execute o programa ao lado.

Veja que a lista salva no campo valor pode representar uma lista de itens em um carrinho de comprar de um site de e-commerce.



Dicas!

Para saber mais sobre Redis e Riak: Consulte o site dos desenvolvedores:

<http://docs.basho.com/riak/kv/2.2.3/learn/why-riak-kv/>

<https://redis.io/>



Dicas!

Para saber mais sobre Redis: Veja alguns casos de uso:

<https://redislabs.com/solutions/use-cases/>

<https://redislabs.com/>



Dicas!

Para saber mais sobre Redis: introdução aos tipos de dados no Redis e sobre a CLI (respectivamente):

<https://redis.io/topics/data-types-intro>

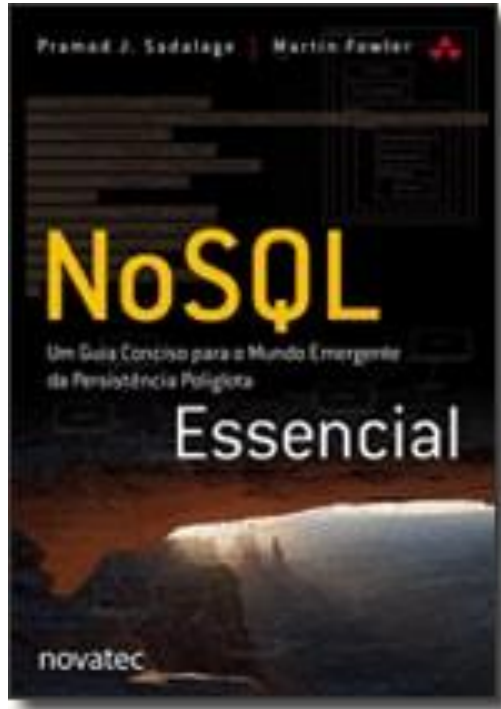
<https://redis.io/topics/rediscli>

Dicas!

Para saber mais sobre Riak: Consulte o site dos desenvolvedores (não compatível com Windows):

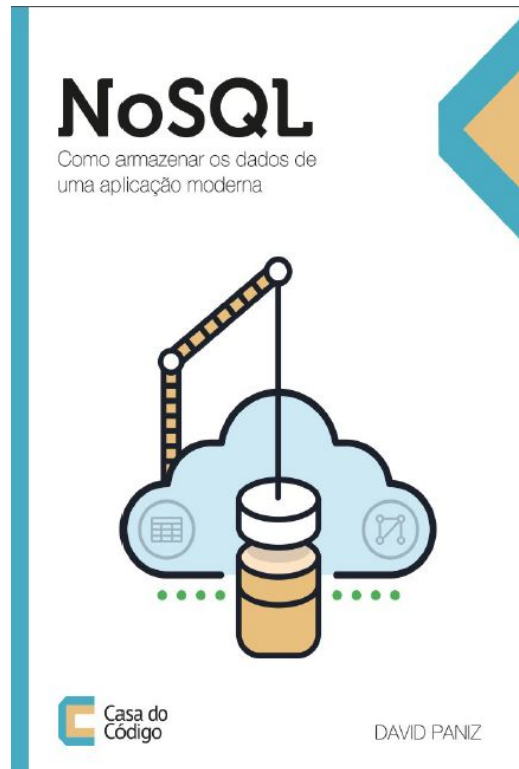
<http://docs.basho.com/riak/kv/2.2.3/downloads/>

Principais Referências



Pramod J.; Sadalage, Martin Fowler.
**NoSQL Essencial: Um Guia Conciso
para o Mundo Emergente da
Persistência Poliglota.** Novatec
Editora, 2013.

Principais Referências



Paniz, David. NoSQL: **Como armazenar os dados de uma aplicação moderna**. Casa do Código, 2017.

Principais Referências

Armazenando dados com

Redis



 Casa do
Código

RODRIGO LAZOTI

Lazoti, Rodrigo. **Armazenando dados com Redis**. Casa do Código, 2017