

# ***Banco de Dados NoSQL***

***(NS0204)***

**Prof. Giovane Barcelos**  
giovane\_barcelos@uniritter.edu.br

# ***Redis* – Tipos de Dados**

## **Conteúdo Programático**

- 1. Introdução ao Redis**
- 2. Primeiros Passos**
- 3. Tipos de Dados**
- 4. Persistência de Dados**
- 5. Aplicação prática com Node.js**

# ***Redis* – Persistência dos Dados**

## **Itens Abordados**

- 1. Visão geral**
- 2. *RDB & Snapshotting***
- 3. AOF – Acrescenta somente em arquivo**
- 4. *RDB & AOF* em ação**

## 4. Redis – Persistência – Visão Geral

### Quais as características da persistência de dados?

- Os conjuntos de dados são todos armazenados na memória
- Os conjuntos de dados podem ser salvos no disco
- **Redis fork** – possível criar processos filho que são uma cópia exata do pai
- Cópia de *snapshot*

# 4. Redis – Persistência – Visão Geral

## Processo de Persistência

1. O cliente envia o comando de gravação ao banco de dados (**memória do cliente**)
2. O banco de dados recebe a gravação (**memória do servidor**)
3. O banco de dados envia a chamada do sistema que grava dados no disco (**buffer do Kernel**)
4. O sistema operacional transfere o *buffer* de gravação para o controlador de disco (**cache de disco**)
5. O controlador de disco grava na mídia física (**Disco físico**)

## **4. Redis – Persistência – Visão Geral**

### ***Pools***

- **Múltiplos Servidores Redis** executados na mesma máquina usando diferentes portas
  - ✓ **Uso eficiente de memória**
  - ✓ **Mais *CPUs***
  - ✓ **Melhor ajuste**

## **4. Redis – Persistência – Visão Geral**

### **Replicação**

- **A replicação mestre escravo simples** permite que os servidores Redis escravos sejam cópias de servidores mestres
  - ✓ Replicação assíncrona
  - ✓ Múltiplos escravos
  - ✓ Conexões de outros escravos
  - ✓ Não bloqueio do lado escravo
  - ✓ Escalabilidade e redundância de dados
  - ✓ Escravo somente leitura

## **4. Redis – Persistência – Visão Geral**

### ***Processo de Replicação***

- 1. O mestre começa a salvar em segundo plano e começa a armazenar no *buffer* todos os novos comandos que modificarão o conjunto de dados**
- 2. Após a gravação em segundo plano, o mestre transfere o arquivo de banco de dados para o escravo**
- 3. O escravo salva os arquivos no disco e o carrega na memória**
- 4. O mestre envia todos os comandos de *buffer* para o escravo**



## **4. Redis – Persistência – Visão Geral**

### ***Opções de Persistência***

- ***RDB (Redis Database): Point-in-time snapshots***
- ***AOF (Append Only File): Arquivo de log***
- **Possível habilitar ambos: *RDB & AOF***

## **4. Redis – Persistência – RDB**

### ***RDB – Redis DataBase File***

- **Modo de persistência simples, compacto**
- **Ativado por padrão**
- **Usa *snapshots***

## **4. Redis – Persistência**

## **4. Redis – Persistência – RDB**

### ***Quais as vantagens do RDB?***

- **Fácil de usar**
- **Muito compacto**
- **Perfeito para backup e recuperação**
- **Maximiza o desempenho do *Redis***
- **Permite reiniciar mais rapidamente com conjuntos de dados grandes em comparação com o *AOF***

## **4. Redis – Persistência – RDB Snapshoting**

- **Controlado pelo usuário**
- **Pode ser modificado em tempo de execução**
- **Os *snapshots* são produzidos como arquivos .rdb**
- **Comandos SAVE & BGSAVE**

## **4. Redis – Persistência – RDB**

### **SAVE & BGSAVE**

#### ➤ **SAVE**

- ✓ Grava o conjunto de dados de forma síncrona produzindo um *snapshot* ponto a ponto de todos os dados dentro da instância do Redis na forma de um arquivo .rdb
- ✓ Não deve ser chamado em ambientes de produção, pois bloqueará todos os outros clientes. Em vez disso, use o BGSAVE

#### ➤ **SAVE 60 100**

- ✓ Grave o conjunto de dados para o disco a cada 60 segundos se pelo menos 100 chaves mudaram

#### ➤ **BGSAVE**

- ✓ Salva o banco de dados em segundo plano e o pai continua a atender os clientes

## **4. Redis – Persistência – RDB**

### ***Desvantagens do RDB***

- **Possui limite de *savepoints***
- **Não é bom se você deseja minimizar a chance de perda de dados se o Redis parar de funcionar**
- **Precisa de `fork()`, muitas vezes, que pode levar muito tempo e pode prejudicar o desempenho da CPU**

## **4. Redis – Persistência – AOF**

### ***AOF – Append of File***

- **Toda operação é registrada**
- ***Log* é o mesmo formato usado pelos clientes**
- **Pode ser canalizado para outra instância**
- **O conjunto de dados pode ser reconstruído**

## **4. Redis – Persistência – AOF**

### ***Reescrita AOF***

- **Usado quando o arquivo AOF fica muito grande**
- **Reescreve o banco de dados a partir do zero**
- **Acessa diretamente os dados na memória**
- **Não há necessidade de acesso ao disco**
- **Uma vez escrito, o arquivo temporário está sincronizado no disco**



## 4. Redis – Persistência – AOF

### *Políticas fsync*

- Sem ***fsync*** - Feito pelo sistema operacional. Normalmente, cada 30s ou mais
- ***Fsync*** a cada segundo (padrão)
- ***Fsync*** em cada consulta (lento)

## **4. Redis – Persistência – AOF**

### ***Vantagens do AOF***

- **Muito mais durável**
- **Arquivo único sem corrupção**
- **Reescrita automática em segundo plano, se ficar muito grande**
- **Registro / instruções fáceis de entender**

# Lembre-se

“Que a força esteja contigo.”

*Star Wars*

