

# Capítulo 8 – Redis

## Dia 2 – Advanced Usage e Distribution

No segundo dia com o **Redis**, vamos além das operações básicas e exploramos **recursos avançados**, bem como estratégias de **distribuição e escalabilidade**.

### Operações Avançadas

#### Transações

O Redis oferece suporte a transações com os comandos `MULTI`, `EXEC`, `DISCARD`.

```
MULTI  
INCR contador  
INCR contador  
EXEC
```

 Executa os dois `INCR` de forma atômica.

#### Scripts com Lua

Permite escrever lógicas customizadas diretamente no servidor.

```
EVAL "return redis.call('GET', KEYS[1])" 1 contador
```

 Scripts Lua são atômicos e eficientes para operações complexas.

#### Publicação e Assinatura (Pub/Sub)

O Redis pode ser usado como **sistema de mensageria**.

- Publicar mensagem:

```
PUBLISH canal1 "Olá, Redis!"
```

- Assinar canal:

```
SUBSCRIBE canal1
```

 Muito usado em **notificações em tempo real e chat**.

## **Streams**

Streams permitem processar eventos de forma confiável.

```
XADD pedidos * produto "Notebook" preco 4500  
XRANGE pedidos - +
```

⚠️ Ideal para **event sourcing, logs e pipelines de eventos**.

---

## **Persistência Avançada**

- **RDB (Schemas)** → cópias periódicas do banco.
- **AOF (Append Only File)** → log de operações.
- **Modo híbrido** → combina os dois.

⚠️ Estratégia comum: usar RDB para backup rápido + AOF para garantir durabilidade.

---

## **Replicação**

O Redis suporta replicação **master-replica**:

```
# No replica  
REPLICAOF 192.168.0.10 6379
```

- O nó replica todas as operações do master.
- Failover pode ser automatizado com **Redis Sentinel**.

⚠️ Isso garante alta disponibilidade.

---

## **Cluster Redis**

O Redis pode ser distribuído em um **cluster**. - Divide os dados em **16384 slots**. - Cada nó é responsável por um conjunto de slots. - Redistribui automaticamente os dados.

⚠️ Isso garante **escalabilidade horizontal**.

---

## **Exemplo de Uso: Ranking Global**

Imagine um game com milhões de jogadores. - Usar **ZADD** em Redis Cluster para distribuir rankings. - Cada nó mantém parte do conjunto. - Consultas globais podem ser balanceadas.

---

## Resumo do Dia 2

- Redis suporta **transações e scripts Lua** para lógicas avançadas.
- Oferece **Pub/Sub** e **Streams** para mensageria e eventos.
- Suporta persistência híbrida (RDB + AOF).
- Permite replicação master-replica e clusters distribuídos.
- Excelente para aplicações em tempo real, escaláveis e resilientes.

👉 No próximo passo: **Dia 3 – Playing with Other Databases**, veremos como o Redis pode ser integrado a outros bancos para maximizar performance e escalabilidade.