

Capítulo 8 – Redis

Dia 1 – CRUD e Datatypes

O Redis é um banco **chave-valor** avançado. Diferente dos demais, ele oferece múltiplos **tipos de dados nativos** que permitem modelar problemas variados com simplicidade e altíssimo desempenho.

Operações CRUD básicas

Criando e lendo chaves (Strings)

```
SET usuario:1 "Douglas"  
GET usuario:1
```

Atualizando valores

```
SET usuario:1 "Douglas Monquero"
```

Deletando chaves

```
DEL usuario:1
```

 Strings também suportam operações numéricas:

```
SET contador 0  
INCR contador # 1  
INCR contador # 2  
DECR contador # 1
```

Listas (Lists)

Listas são sequências ordenadas de strings.

```
LPUSH fila "tarefa1"  
LPUSH fila "tarefa2"  
LRANGE fila 0 -1
```

 Saída: ["tarefa2", "tarefa1"]

- `LPUSH` insere no início.
- `RPUSH` insere no final.
- `LPOP` remove do início.
- `RPOP` remove do final.

💡 Ideal para **filas e pilhas**.

Conjuntos (Sets)

Conjuntos armazenam valores únicos, sem ordem definida.

```
SADD cores "vermelho"  
SADD cores "azul"  
SADD cores "verde"  
SMEMBERS cores
```

🔔 Saída: {vermelho, azul, verde}

Operações comuns: - `SISMEMBER cores "azul"` → verifica existência. - `SREM cores "verde"` → remove elemento. - `SUNION set1 set2` → união de conjuntos. - `SINTER set1 set2` → interseção.

💡 Muito usado para **tags, permissões e conjuntos únicos**.

Conjuntos Ordenados (Sorted Sets)

Cada elemento possui uma **pontuação numérica**, permitindo ordenação automática.

```
ZADD ranking 100 "Douglas"  
ZADD ranking 200 "Ana"  
ZRANGE ranking 0 -1 WITHSCORES
```

🔔 Saída: Douglas 100, Ana 200

- `ZINCRBY ranking 50 "Douglas"` → incrementa pontuação.
- `ZRANK ranking "Douglas"` → posição no ranking.
- `ZREVRANGE ranking 0 0 WITHSCORES` → maior pontuação.

💡 Excelente para **leaderboards e sistemas de pontuação**.

Hashes

Mapas de campo-valor, ideais para representar objetos.

```
HSET usuario:100 nome "Douglas" idade 30 cidade "Maringá"  
HGET usuario:100 nome  
HGETALL usuario:100
```

⚠️ Saída: {nome: Douglas, idade: 30, cidade: Maringá}

- HDEL usuario:100 idade → remove campo.
- HINCRBY usuario:100 idade 1 → incrementa idade.

📌 Muito usado para **modelar entidades** (usuários, produtos, configs).

⌚ Expiração de Chaves

Qualquer chave pode ter tempo de vida (TTL).

```
SET sessao:abc123 "ativo"  
EXPIRE sessao:abc123 60    # expira em 60s  
TTL sessao:abc123
```

📌 Ideal para **sessões de usuário e caches temporários**.

🕒 Resumo do Dia 1

- Redis suporta múltiplos tipos de dados além de strings.
- **Strings** → valores simples e contadores.
- **Lists** → filas e pilhas.
- **Sets** → coleções únicas.
- **Sorted Sets** → rankings e ordenação.
- **Hashes** → entidades complexas.
- **TTL** → controle de expiração.

⚠️ No próximo passo: **Dia 2 - Advanced Usage e Distribution**, exploraremos replicação, particionamento e usos mais avançados do Redis.