

Capítulo 5 - Ranks

Vimos nas aulas passadas diversos tipos de problemas os quais o *redis* é capaz de resolver: estruturas, características, armazenar *strings*, números, tipos *booleanos*, fizemos operações com elementos e armazenamos listas e trabalhamos com elas como se fossem filas ou conjuntos.

Nesta aula veremos outros tipos de estruturas. À medida em que interagimos com algum sistema, por exemplo no site do Alura, ganhamos pontos e experiência, para mostrar que estamos nos desenvolvendo.

Existem maneiras subjetivas de mostrar isso: "Você concluiu o curso!". E maneiras mais objetivas: "Você subiu de nível!".

Queremos armazenar todos esses usuários, cada um com sua pontuação, no *Redis*. Com isso podemos mostrar tais informações de forma otimizada e rápida com os elementos rankeados, **ordenados**.

O primeiro passo é, de fato, armazenar a pontuação de todos os usuários. Precisamos definir o tipo do dado, seu nome (que deve ser único) e sua pontuação, ou seja, o valor atrelado ao dado. Como o *id* do usuário deve ser único, vamos simplesmente defini-lo com um número.

```
SET "jogador:1" 50076
OK
```

Então o jogador 1 tem 50076 pontos e, realmente, ele foi armazenado:

```
GET "jogador: 1"
"50076"
```

Na hora de mostrarmos a pontuação do usuário, seria interessante ela vir com seu nome, e não apenas um número. Para sabermos o tipo da chave, no caso "jogador:1", fazemos:

```
TYPE "jogador:1"
string
```

O tipo é *string*. Mas, como queremos armazenar mais de um tipo de dado, devemos usar o `HSET`. Então vamos deletar a informação e setar o *hash*:

```
DEL "jogador:1"
(integer) 1

HSET "jogador:1" "nome" "guilherme"
(integer) 1
HSET "jogador:1" "pontos" 50076
(integer) 1
```

Vamos setar outro jogador:

```
HSET "jogador:2" "nome" "carlos"
(integer) 1
HSET "jogador:2" "pontos" 76564
(integer) 1
```

Para verificarmos se *todos* os dados de cada jogador foram inputados:

```
HGETALL "jogador:1"
1) "nome"
2) "guilherme"
3) "pontos"
4) "50076"

HGETALL "jogador:2"
1) "nome"
2) "carlos"
3) "pontos"
4) "76564"
```

O `HSET` e o `HGET` nós já conhecíamos. As novidades aqui foram o `TYPE` - para sabermos o tipo da chave - e o `HGETALL` - para trazer todos os dados da *hash*.

O que nos falta aqui é ordenar essas chaves usando sua pontuação, para tal usamos a ideia de lista que daqui a pouco veremos. Primeiramente vamos pensar como podemos aumentar a pontuação dos jogadores. Se usarmos simplesmente o comando `INCR`, não irá funcionar, uma vez que o tipo não é mais *string* e sim *hash*, logo `HINCRBY`:

```
HINCRBY "jogador:1" "pontos" 10
(integer) 50086
```

Também temos o `HINCRBYFLOAT`, para incrementarmos com números decimais.

Agora sim vamos pensar como ordenar um conjunto de jogadores. Ora, criamos então este conjunto (sem repetição de elementos) usando o `ZADD` (o "Z" é para não repetir os "SS" de *Sorted Set*). Após o comando, escrevemos o nome do conjunto que armazenará todas as pontuações, depois vêm a informação que queremos ordenar seguida do nome do jogador. Então temos:

```
ZADD pontuacoes 50076 guilherme
(integer) 1
ZADD pontuacoes 65543 carlos
(integer) 1
ZADD pontuacoes 33786 daniela
(integer) 1
ZADD pontuacoes 8754 paulo
(integer) 1
```

Na chave "pontuacoes" guardamos os jogadores com seus pontos e nomes, respectivamente. Vamos ver qual o tipo da chave "pontuacoes":

```
TYPE pontuacoes  
zset
```

Ou seja, é um conjunto ordenado. E para saber a sua cardinalidade:

```
ZCARD pontuacoes  
(integer) 4
```

De fato, o conjunto contém 4 elementos.

Para visualizarmos os elementos do conjunto ordenado, fazemos:

```
ZRANGE pontuacoes 0 3  
1) "paulo"  
2) "daniela"  
3) "guilherme"  
4) "carlos"
```

De fato, os elementos estão ordenados. Porém perceba que o conjunto está em ordem crescente. O que queremos é o inverso (reverted), em ordem decrescente, para que o jogador com mais pontos fique em primeiro lugar. Para tal, fazemos:

```
ZREVRANGE pontuacoes 0 3  
1) "carlos"  
2) "guilherme"  
3) "daniela"  
4) "paulo"
```

Mais para frente veremos como fazer algumas operações com conjuntos ordenados, mas agora já sabemos como montá-los.