

Últimas notícias

Capítulo 1 - Últimas notícias

Já sabemos usar o *Redis* para resolver vários problemas, utilizando *strings*, tipos numéricos, tipos *booleanos*, etc em conjuntos muito grandes. Nesta segunda parte do curso de *Redis* veremos mais alguns problemas para resolvermos.

Um dos problemas, muito comum, é querermos saber quais foram as últimas notícias publicadas no site de um jornal para também podermos deixá-lo sempre atualizado. No site do Alura nós mostramos os três últimos depoimentos de cada curso. Poderíamos querer saber quais foram os últimos usuários logados.

Veremos aqui como armazenamos **listas de valores** de um tamanho determinado. Vamos modelar, então, um sistema de notícias com as últimas três de um site.

A primeira notícia é

Jogador de futebol é flagrado jogando basquete na rua.

No momento em que escrevemos a segunda notícia, mais recente, queremos deixá-la em cima, ou seja, antes. Então, toda nova notícia deve aparecer primeiro:

Novo curso de Redis é lançado pelo Alura

Jogador de futebol é flagrado jogando basquete na rua.

E, para a terceira notícia:

Jogador de basquete é encontrado jogando futebol na quadra.

Novo curso de Redis é lançado pelo Alura

Jogador de futebol é flagrado jogando basquete na rua.

Pode ser que apareça uma quarta notícia:

Guilherme Silveira faz aniversário e espera presentes.... sentado.

Jogador de basquete é encontrado jogando futebol na quadra.

Novo curso de Redis é lançado pelo Alura

Jogador de futebol é flagrado jogando basquete na rua.

Se aparecer uma quarta notícia, devemos esconder aquela primeira, a mais em baixo, afinal só estamos interessados em mostrar as últimas três:

Guilherme Silveira faz aniversário e espera presentes.... sentado.

Jogador de basquete é encontrado jogando futebol na quadra.

Novo curso de Redis é lançado pelo Alura

Todo o instante em que chegar uma notícia nova, ela será colocada antes das outras na fila e a mais antiga some.

Vamos numerar as notícias para já começarmos a pensar na implementação no *Redis*. Quando tínhamos apenas uma notícia, ela estava na posição 0. Assim que a segunda notícia chegou, ela entra na posição 0 e a primeira vai para a posição 1. Para a terceira a mesma coisa: ela fica em 0, a segunda em 1 e a primeira em 3. Com a quarta repetimos:

- quarta - 0,
- terceira - 1,
- segunda - 2,
- primeira - 3.

Porém, não queremos que a posição 3 apareça. Então, mandamos o banco de dados manter a lista na memória só até a posição 2. Fazemos isso primeiramente inserindo os elementos na lista.

Cada elemento novo é empurrado (*push*) para a esquerda (*left*) da lista. Precisamos também de um nome para a chave na qual os elementos (notícias) serão armazenados: "ultimas_noticias". E o nome de cada elemento: a manchete da notícia. O comando para a primeira notícia fica assim:

```
LPUSH ultimas_noticias "Jogador de futebol e flagrado jogando basquete na rua"  
(integer) 1
```

Para a segunda, devemos encaixá-la também à esquerda:

```
LPUSH ultimas_noticias "Novo curso de Redis e lancado pelo Alura"  
(integer) 2
```

Toda vez que chega uma notícia nova, usamos esse comando. Mas ainda precisaremos de um comando que apare (trim) as últimas notícias à esquerda (left) da lista e deixe-a com os elementos nas posições de 0 a 2:

```
LTRIM ultimas_noticias 0 2
OK
```

Vamos incluir todas as notícias já vistas para visualizar os elementos usamos o comando `LINDEX` :

```
LINDEX ultimas_noticias 0
"Guilherme Silveira faz aniversario e espera presentes... sentado"
```

De fato, esta foi a última notícia a ser inserida na lista, a mais à esquerda. Para sabermos o tamanho (length) da lista fazemos:

```
LLEN ultimas_noticias
(integer) 4
```

Para visualizar um trecho (range) específico da lista, fazemos

```
LRANGE ultimas_noticias 1 2
1) "Jogador de basquete e encontrado jogando futebol na quadra"
2) "Novo curso de Redis lançado pelo Alura"
```

Aqui visualizamos os elementos nas posições de 1 a 2. Perceba que se passarmos valores maiores de trechos, como aplicamos o `LTRIM` na lista, serão retornados aqueles três primeiros elementos, como queríamos.