

## Estatísticas de acesso a uma página

### Estatísticas de acesso a uma página

Empresas possuem um sistema para armazenar as informações sobre quais páginas de seus sites estão sendo acessadas e quantas vezes. Conseguimos não só isso, mas também saber quantas pessoas assistiram a um determinado vídeo ontem ou hoje. Queremos armazenar quantas pessoas acessaram cada funcionalidade do nosso site com o passar do tempo.

Perceba que queremos armazenar dia/página e total de acessos, ou seja, *chave-valor*. Vamos ver como fazer isso usando o *Redis*:

```
<ip> SET pagina:/contato:25-05-2015 1
OK
<ip> SET pagina:/contato:25-05-2015 2
OK
```

Isso significa que duas pessoas acessaram a página de contato. Porém não podemos ficar contando dessa forma, uma vez que poderíamos não saber a quantidade anterior de acessos. Devemos achar uma forma de colocar o valor atual +1. Poderíamos pensar que uma forma é visualizar o atual e acrescentar um:

```
GET pagina:/contato:25-05-2015
4
SET pagina:/contato:25-05-2015 5
```

O problema é que alguém já poderia ter feito alguma alteração nesse meio tempo. Queremos uma forma de adicionar 1 aos valores automaticamente. Seria basicamente juntar as duas tarefas de GET e SET em uma só.

Todo comando do *Redis* é garantido que é executado de maneira atômica, ou seja, não existe a possibilidade de que ele seja sobreescrito por outro, ou que outro aconteça no meio tempo entre dois.

Logo, para conseguirmos essa incrementação de 1, usamos o comando `INCR`:

```
<ip> INCR pagina:/contato:25-05-2015
(integer) 10
<ip> INCR pagina:/contato:25-05-2015
(integer) 11
```

Da mesma forma, conseguimos decrementar:

```
<ip> DECR pagina:/contato:25-05-2015
(integer) 10
<ip> DECR pagina:/contato:25-05-2015
(integer) 9
```

Garantimos que se outra pessoa também utilizar o `INCR`, também funcionará sem perda de informação.

Mas e se quisermos incrementar o valor de compras em um site e não o número de acessos? Se fizermos:

```
<ip> INCR compras:25-05-2015
```

Porém assim nós estaremos incrementando o número de compras. Devemos encontrar um meio de incrementar valores diferentes de 1. Para isso usamos o comando `INCRBY`:

```
<ip> INCRBY compras:25-05-2015:valor 15
(integer) 15
<ip> INCRBY compras:25-05-2015:valor 17
(integer) 32
<ip> INCRBY compras:25-05-2015:valor 2
(integer) 34
```

Idem para a decrementação, se houve, por exemplo, uma compra cancelada:

```
<ip> DECRBY compras:25-05-2015:valor 2
(integer) 32
```

Para incrementar valores quebrados usamos o comando `INCRBYFLOAT`. Para decrementar, também usamos o `INCRBYFLOAT` passando um valor negativo:

```
<ip> INCRBYFLOAT compras:25-05-2015:valor 10.50
"42.5"
<ip> INCRBYFLOAT compras:25-05-2015:valor -0.50
"42.0"
```