

Uma fila de espera emails e mensagens de celular

Capítulo 2 - Uma fila de espera, e-mails e mensagens de celular

Veremos nesta aula uma outra aplicação do *Redis* bem comum em sites e aplicativos *mobile*. Quando nos cadastramos em um site, é normal que peçam nosso e-mail ou telefone para confirmar nossa identidade, por questões de segurança e para ter certeza de que ele está se comunicando realmente conosco e não com outro usuário. Nesse momento recebemos um e-mail ou SMS para confirmação.

Existe, então, um mecanismo, em que, por meio da *Web* ou de um aplicativo de celular, nos cadastramos num site. Um tempo depois recebemos um e-mail, uma ligação ou um SMS para confirmarmos o cadastro. Essa segunda parte ocorre no *background*, como uma tarefa paralela.

Para que uma sequência de tarefas sejam executadas em paralelo, é necessário que elas sejam enfileiradas em algum lugar. Um **outro programa** olha para essa fila e vai executando as tarefas na ordem que elas estão.

Um outro bom exemplo de filas de tarefas é o atendimento via chat, que provavelmente possui uma fila de espera de atendimento dos clientes. Sempre lembrando que a primeira tarefa/primeiro cliente será executada/atendido primeiro.

Vimos na aula passada a inserção de elementos pela esquerda de uma lista. Desta vez estamos fazendo pela direita (right) e dando prioridade de saída para aqueles que foram inseridos primeiramente, em vez de a apartarmos. Com isso criamos uma *fila de espera*.

Para fazer a implementação, vamos usar o exemplo de um site que confirma o e-mail dos usuários. Para isso colocamos o e-mail em uma fila e a atendemos sempre da esquerda para a direita. O e-mail mais antigo será o primeiro a sair. Para criar uma fila com elementos usamos o comando `RPUSH` ("R" de *right*):

```
RPUSH "fila:confirma-email" "guilherme.silveira@alura.com.br"
(integer) 1
```

Para visualizarmos a quantidade de elementos em uma lista:

```
LLEN "fila:confirma-email"
(integer) 1
```

De fato, há um elemento na lista. Vamos acrescentar mais dois e-mails:

```
RPUSH "fila:confirma-email" "daniela.mikyung@alura.com.br"
(integer) 2
RPUSH "fila:confirma-email" "carlos.felicio@alura.com.br"
(integer) 3
```

Vamos confirmar que os três estão lá:

```
LRANGE "fila:confirma-email" 0 2
1) "guilherme.silveira@alura.com.br"
```

- 2) "daniela.mikyung@alura.com.br"
- 3) "carlos.felicio@alura.com.br"

Agora que já temos uma fila formada, um outro programa atende, na ordem, os elementos:

```
LINDEX fila:confirma-email 0  
"guilherme.silveira@alura.com.br"
```

Tiramos esse elemento da fila:

```
LTRIM fila:confirma-email 1 3  
OK
```

Aqui precisamos tomar muito cuidado! Se fizermos o `LTRIM` dessa forma, corremos o risco de um cliente (no caso o de número 3) ser cadastrado no meio tempo e, logo após o comando, ser excluído. Para solucionar esse problema, precisamos retirá-lo da fila ao mesmo tempo que o pegarmos, para não dar tempo de outro usuário se cadastrar. O comando que faz esse trabalho é o `POP`, se for da esquerda, `LPOP`:

```
LPOP fila:confirma-email
```

Dessa maneira pegamos o elemento e o retiramos da lista sem correremos o risco de que outro se perca.