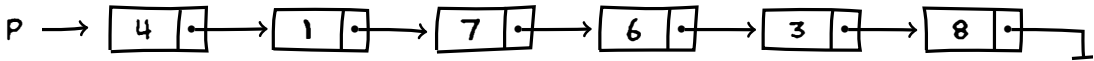


Encontrando o número k

Imagine que o ponteiro p aponta para uma lista encadeada



A tarefa é

→ Verificar se o número k está na lista ou não

Bom, a ideia é realizar a tarefa com uma função recursiva.

Quer dizer, a medida que a gente vai percorrendo a lista, nós verificamos se o elemento atual é o número k.

Se isso é o caso, a gente imprime uma mensagem na tela — (e já começa a voltar)

Mas se a gente chega no Nulo, isso significa que o número k não está na lista.

E se isso é o caso, a gente imprime outra mensagem na tela — (e depois começa a voltar)

A coisa fica assim

```
func busca-Rec (q,k)
  Se ( q == Nulo )
    Imprime ('Não está lá ...')
    Retorna
  Sse ( q->num == k )
    Imprime ('Achei!')
    Retorna
  Senão
    busca-Rec ( q->prox, k )
  Retorna
```

Legal.

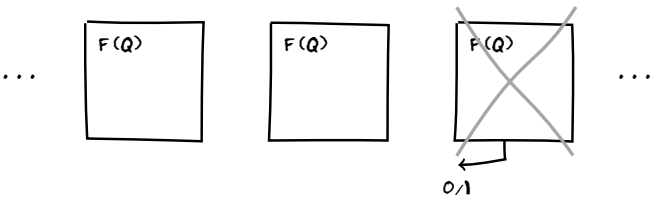
Mas, essa não é a única maneira de fazer as coisas.

Quer dizer, ao invés de imprimir uma mensagem na tela, a gente pode retornar um sinal para a função main() saber se o número k está lá ou não.

A ideia é a seguinte

```
Se ( q == Nulo )
  Retorna (0)
Sse ( q->num == k )
  Retorna (1)
```

Mas para chegar na função main(), esse sinal precisa passar por todas as caixinhas que vão fechando



Para fazer isso, cada caixinha recebe a resposta enviada pela próxima, e retorna essa resposta para a anterior.

A coisa fica assim

```
func busca2-Rec (q,k)
  Se ( q == Nulo )
    Retorna (0)
  Sse ( q->num == k )
    Retorna (1)
  Senão
    resp = busca2-Rec (q->prox, k)
    Retorna (resp)
```

Dessa maneira, a função main() recebe essa informação como o valor de retorno da função busca2-Rec().

E daí, ela pode fazer o que quiser com isso

```
func main ()
  resp = busca2-Rec (p, k)
  Se ( resp == 0 )
    Imprime ('Não está lá...!')
  Senão
    Imprime ('Achei!')
```