

## Momento 1: o meu primeiro decodificador

Pouco tempo depois, recebes um email:

*Jovem,*

*Recebemos mensagens codificadas com sequências de letras e números, mas apenas as letras são importantes. Segue abaixo um exemplo e num anexo as mensagens que os criminosos trocaram entre eles, ordenadas cronologicamente. Torna-se custoso e lento decifrá-las manualmente. Um programa que permita separar os números do resto seria uma ajuda preciosa. Trate disso. Obrigado.*

*Morcela*

Exemplo:

A mensagem [5, f, 7, 9, 8, r, 8, a, 99, s, 1, e, 12, s, 13, 44, 1, s, 7, 18, e, 1, 8, 9, m, s, e, 1, 7, 2, 15, n, t, 19, 18, 33, 2, i, d, o], sendo decodificada (remoção dos números) resulta em [f, r, a, s, e, s, s, e, m, s, e, n, t, i, d, o].

Pensas na aula de LP, em que estudastes listas (lembras-te que a tua professora disse que há um código maravilha na página da cadeira). Decides implementar o programa pedido em **Prolog**. Será útil para o próximo MAP e para o projeto que se avizinha.

### Exercício: implementa, em Prolog, o predicado

`eliminaNumeros(Lista, ListaSemNumeros)`

em que `ListaSemNumeros` é a lista resultante de eliminar de `Lista` os números.

#### Exemplos:

?- `eliminaNumeros([1, o, 2, 4, l, 1, a], ListaSemNumeros).`

`ListaSemNumeros = [o,l,a].`

?- `eliminaNumeros([1], ListaSemNumeros).`

`ListaSemNumeros = [].`

Podes também testar com uma lista explícita:

?- `Lista = [o, 66, 4, l, 1, a, 67], eliminaNumeros(Lista, ListaSemNumeros).`

Sugestão: existe em Prolog o predicado `number(N)`, que é avaliado para `true` se `N` for um número e para `false` caso contrário. Tenta usá-lo.

Quando acabas de correr o teu decodificador sobre as sete mensagens codificadas do anexo, ficas a olhar para o resultado do teu trabalho. Que mensagens mais estranhas...