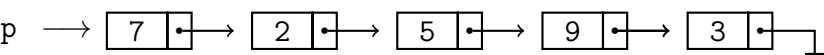


Na aula passada, nós vimos as listas encadeadas



E vimos algoritmos que percorriam a lista encadeada para inspecionar os seus elementos.

Hoje nós vamos aprender a fazer modificações na lista.

E a tarefa mais simples de todas é inserir um novo elemento na lista.

Mas para isso, é preciso aprender a criar um novo registro.

Só que isso é uma coisa bem fácil

Quer dizer, basta utilizar o comando `Novo()`, que nos dá um ponteiro para um registro vazio



Daí, nós podemos preencher o registro como nós quisermos

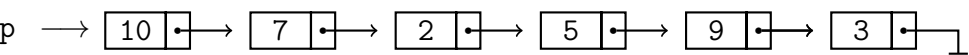


Legal.

Isso já é suficiente para inserir um novo registro na lista do exemplo acima

```
q = Novo(RegInt)
q->num = 10
q->prox = p           // aponta para o 1o elemento
p = q                // esse é o novo 1o elemento
```

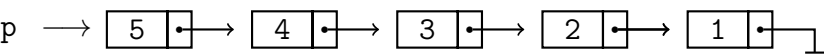
O que nos dá



Agora, a gente também pode criar uma lista de qualquer tamanho a partir do zero

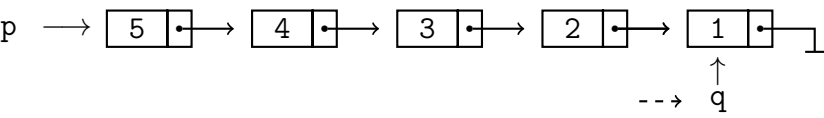
```
p = Nulo
Para k=1 Ate 5
    q = Novo(RegInt)
    q->num = k
    q->prox = p
    p = q
```

O que produz a lista



E agora, a gente também pode escrever uma função que insere um novo elemento no fim da lista.

Para isso, a gente leva um ponteiro auxiliar até o último elemento



E daí, a gente realiza a operação de inserção

```
func insercaoFim (k, p)

    Se ( p == Nulo )           // a lista ainda está vazia
        r = Novo(RegInt)
        r->num = k
        r->prox = Nulo
        Retorna(r)

    q = p                      //
    Enquanto ( q->prox != Nulo ) // vai até o último elemento
        q = q->prox           //

    r = Novo(RegInt)
    r->num = k                 //
    r->prox = Nulo             // insere o registro no fim
    q->prox = r                //
    Retorna (p)
```