

última aula: listas simplesmente ligadas
sem sentinela

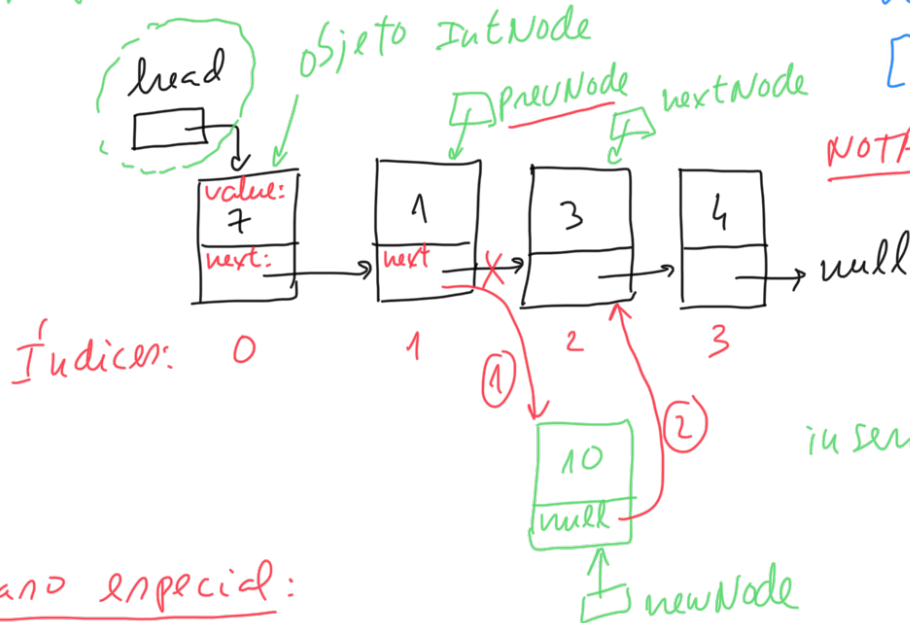
①

método insert(v: Int, index: Int)

Referência

valor
a inserir

índice onde
vai inserir
[0; size-1]

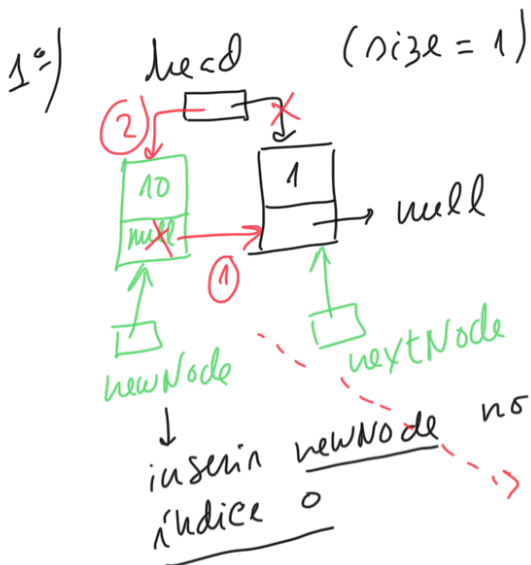


NOTA: Ao ter a referência
prevNode e
nextNode, posso
fazer por qualquer
ordem.
inserir (10) no índice (2)

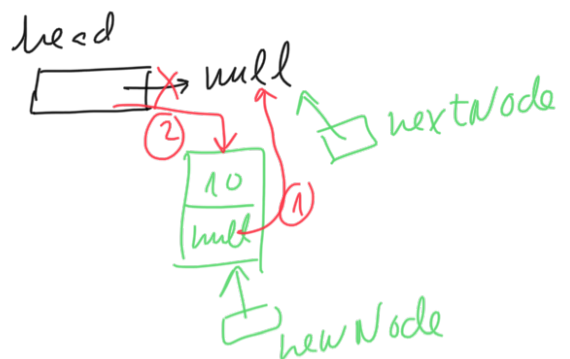
Caso especial:

Inserir no índice 0?

↳ Não existe prevNode, e tenho que manipular
o head diretamente



2º) (size = 0)



- (1) newNode.next = nextNode
- (2) head = newNode

```
class SIntList {
```

```
    ...
    var size: Int
    var head: IntNode?
    constructor() { head = null }
```

opcional, pois é
feito por
omissão

```
    ...
    fun insert (v: Int, index: Int) {
        val newNode = IntNode(v)
        var prevNode: IntNode? = null
        var nextNode: IntNode? = null
```

```
        if (index == 0) {
```

```
            nextNode = head
```

```
            ① newNode.next = nextNode
```

```
            ② head = newNode
```

```
            ++ size
```

```
            return
```

```
        } // index > 0
```

```
        var i = 0
```

```
        prevNode = head // prevNode inicial
```

```
        while (i < index - 1) {
```

```
            prevNode = prevNode.next
```

```
            ++ i
```

```
        }
        nextNode = prevNode.next
```

```
        ① prevNode.next = newNode
```

```
        ② newNode.next = nextNode
```

```
        ++ size
```

```
    }
```

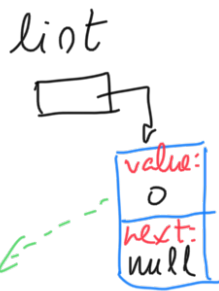
COM A
SENTINELA,
ESTE
CÓDIGO
DESAPARECE

Lista simplesmente ligada

com sentinela

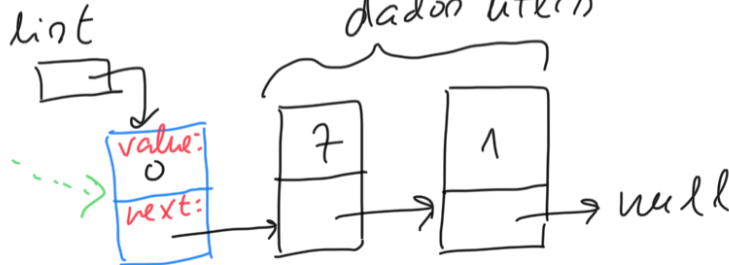
3.

1º caso) lista vazia (não com a sentinela)



no sentinela

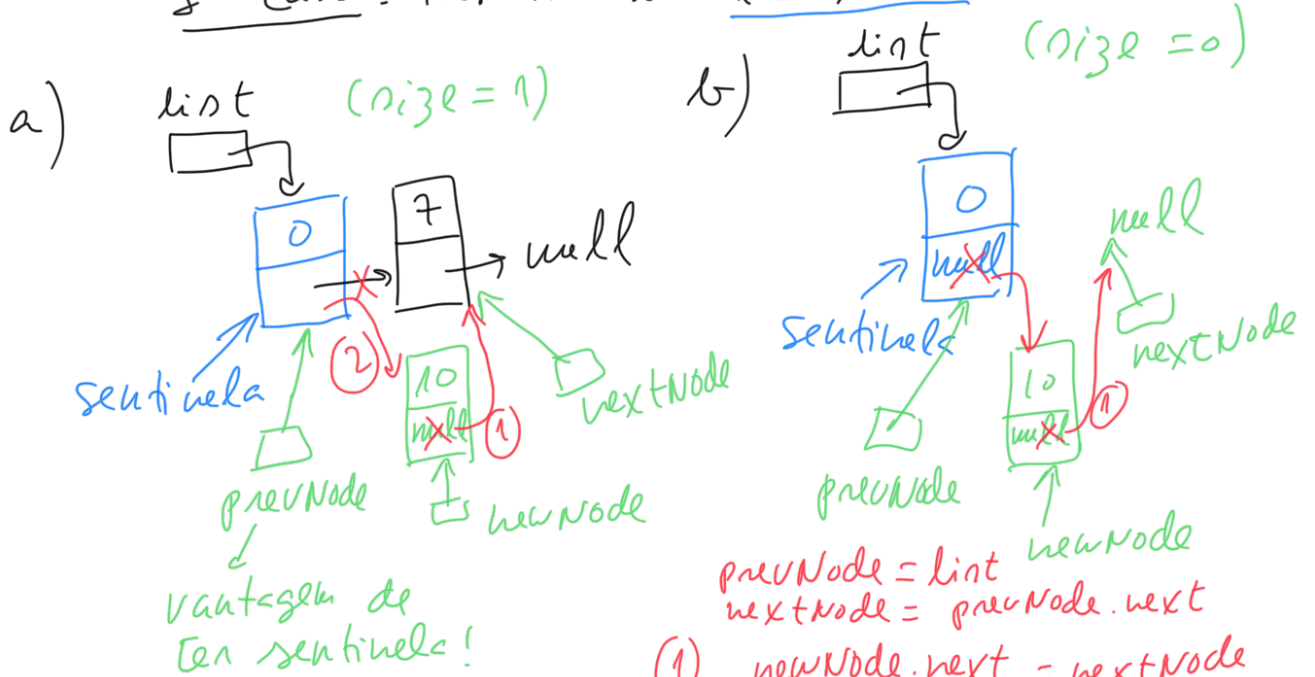
2º caso) lista com n elementos
ex: $n = 2$



método insert (v, index) com sentinela

4.

1º caso: inserir no index = 0

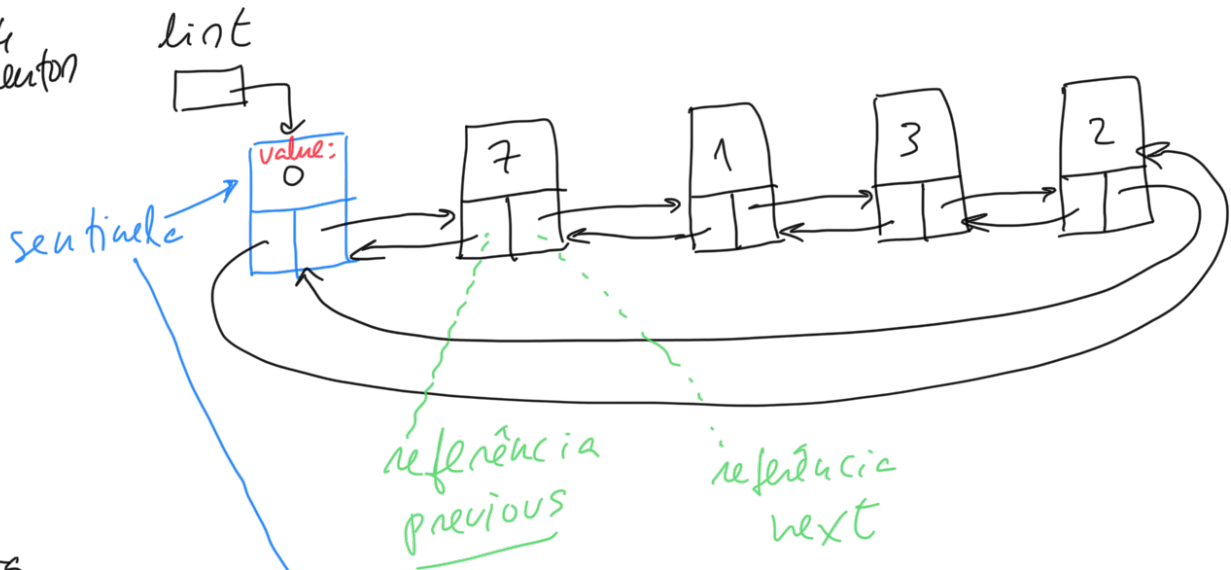


- $prevNode = list$
 $nextNode = prevNode.next$
- (1) $newNode.next = nextNode$
 - (2) $prevNode.next = newNode$
- ← código igual para o caso geral ($index > 0$)

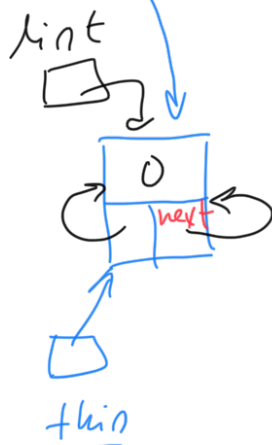
Lista duplamente ligada com sentinela e circular

5.

lista
com 4
elementos



lista
vazia:



no visível
no constructor

modo
para
construir
a
sentinela

modo
para
outros
nós

```
class IntNode {
    var value: Int
    var next: IntNode?
    var previous: IntNode?

    constructor() {
        this.value = 0 // opcional
        this.next = this
        previous = this
    }

    constructor(v: Int) {
        value = v
        // next e previous
        // são null
    }
}
```

```
class DList {
```

6.

```
    val list: IntNode?
```

```
    constructor() {
```

```
        list = IntNode() // constructor;
                           no sentinel
```

```
}
```

```
    fun addFront(v: Int) {
```

```
        (T.B.C.) + addLast
```

```
}
```