Unique algorithm

main:

```
val list = listOf(1, 3, 1, 4, 1, 3, 5, 4, 5)

val uniqueList = unique(list)

for (elem in uniqueList)
    println(elem)
```

→ pode ser uma lista, array, range

```
for (i in 0 until 10)
    println(i) // 0, 1, ... , 9
```

range

→ devido ao until ↓ Exclusive

```
for (i in 0..10)
    println(i) // 0, 1, ..., 10
```

range → inclusive

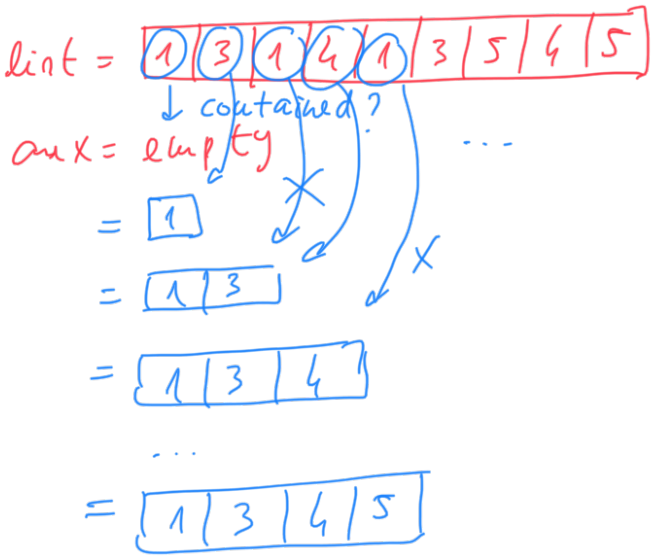
```
for (i in 10 downTo 0)
    println(i)      // 10, 9, 8, ... , 0
```

→ inclusive

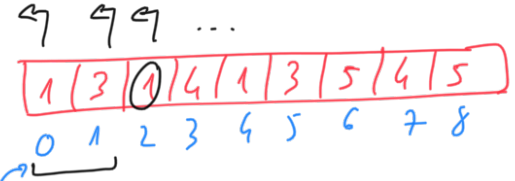
```
fun unique ( list: List<Int> ) {
    // #1 - Use auxiliar list
    // for repeated
    var aux = emptyList<Int>()
    var i = 0
    while ( i < list.size() ) {
        // check if repeated
        if ( list[i] !in aux )
            aux += list[i]

        ++i
    }

    return aux
}
```



```
fun unique ( ... ) {
    // #2 - search in the same list
    var aux = emptyList<Int> ()
    var i = 0;   var found : Boolean
    while ( i < list.size) {
        // search repeated
        // in the
        // same list
        // from the
        // beggining to the
        // current index - 1
        found = false
        for ( j in i-1 downTo 0 )
            if (list[i] == list[j]){
                found = true
                break // sai do ciclo corrente
            }
        }
        if (! found)
            aux += list[i]
        ++i
    }
    return aux
}
```



<=>

```
for (j in 0 until i)
```

OR

```
for (j in 0..i-1)
```

ou:

```
var j = i-1 ; found = false
while ( j >= 0 && ! found)
{
    if ( list[i] == list[j])
    {  found = true
       break // not necessary
    }
    --j
}
```

→ ...