

## Tipo de Datos Abstracto *Lista*

**Operación, funcionalidad, comportamiento:** *Insertar un nuevo elemento*

**Objetivo:** dado un nuevo elemento, agregarlo al final de la lista.

**Lenguaje Natural:** si el nuevo elemento es un válido, recorrer la lista hasta el final y “enganchar” allí el nuevo elemento (apuntar el que era último a éste nuevo). Condición especial: si la lista está vacía, el nuevo elemento pasará a ser el “primero”.

### PRECONDICIONES.

- El nuevo elemento debe existir, y ser del tipo correcto (el que corresponde a los elementos de la lista definida)
- La etiqueta / clave del elemento debe ser valida
- ¿hay estados específicos de la lista que deban ser considerados?

### POSTCONDICIONES:

- Si la lista estaba vacía antes de insertar, ahora el nuevo elemento es el *primero* y la cantidad de elementos es 1
- La lista contendrá un elemento más (la cantidad de elementos se ha incrementado en 1)
- El elemento insertado se encuentra en la lista
- El elemento se encuentra al final de la lista (apunta a nulo en su campo “siguiente”)

### SEUDOCÓDIGO

**TLista.Insertar** (unElemento : tipoElemento)

**COM**

```
Si vacía entonces
    primero ← unElemento
SiNo
    Actual ← primero
    Mientras Actual.siguiente <> nulo
        Actual ← Actual. Siguiente
    FinMientras
    Actual.siguiente ← unElemento // insertamos al final...
FinSi
```

**FIN**

### CASOS DE PRUEBA

1. A partir de la lista vacía, insertar un elemento conocido y comprobar que
  - a. El elemento está en la lista
  - b. Cantidad de elementos es 1
2. Luego de varias inserciones, agregar un nuevo elemento y comprobar que:
  - a. El elemento está en la lista
  - b. El elemento es el último de la lista
  - c. Cantidad de elementos se ha incrementado en 1
3. OTROS????

## Tipo de Datos Abstracto *Lista*

**Operación, funcionalidad, comportamiento:** *buscar y devolver un elemento que tenga cierta etiqueta*

**Objetivo:** dada una etiqueta, devolver el elemento de la lista que tenga esa etiqueta o indicar que no hay en la lista un elemento con esa etiqueta.

**Lenguaje Natural:** Si la lista no está vacía, recorrerla comenzando por el principio y comparando la etiqueta buscada con la de cada elemento visitado. Si se encuentra una coincidencia, devolver el elemento correspondiente. Si, de lo contrario, se llega al final de la lista sin coincidencia, o si la lista está vacía, indicar que la lista no contiene ningún elemento con clave como la indicada..

### PRECONDICIONES.

- La etiqueta a buscar es válida (discutir)
- ¿hay otros estados específicos de la lista que deban ser considerados?
- ¿Lista no vacía? Discutir....

### POSTCONDICIONES:

- La lista no se ve alterada de ninguna forma (la cantidad de elementos no varía, el “primero” no se altera, etc.)
- En caso de existir, el elemento devuelto no se ve afectado de ninguna manera.

### SEUDOCÓDIGO

**TLista.buscar** (unaEtiqueta : tipoEtiqueta): TElemento // devuelve un TElemento

```
COM
  Si vacía entonces
    Devolver nulo                                // salir
  SiNo
    Actual ← primero
    Mientras Actual <> nulo
      Si Actual.Etiqueta = unaEtiqueta
        Devolver Actual                          //salir
      Actual ← Actual. Siguiente
    FinMientras
    Devolver nulo                                // salir
  FinSi
FIN
```

### Otra forma....

**TLista.buscar** (unaEtiqueta : tipoEtiqueta): TElemento // devuelve un TElemento

```
COM
..tempElem ←nulo
Actual ← primero
Mientras Actual <> nulo
  Si Actual.Etiqueta = unaEtiqueta
    tempElem ← Actual
    salir bucle                                //break
  FinSi
  Actual ← Actual.Siguiente
FinMientras
Devolver tempElem                              // salir
FIN
```

## CASOS DE PRUEBA

1. A partir de la lista **vacía**
  - a. Al invocar *buscar* sobre la lista vacía, con cualquier etiqueta, la operación devuelve *nulo*
  - b. Insertar un elemento conocido y comprobar que Al invocar *buscar* con la etiqueta del elemento que se ha insertado, se lo encuentra (es decir, devuelve un elemento con etiqueta igual a la buscada)
2. Insertar varios elementos – con etiquetas conocidas- en la lista y
  - a. Comprobar que encuentra un elemento entre los recientemente insertados
  - b. Comprobar que devuelve *nulo* cuando se invoca con una etiqueta que sabemos que no está.
  - c. Cantidad de elementos se ha incrementado en 1
3. OTROS????