

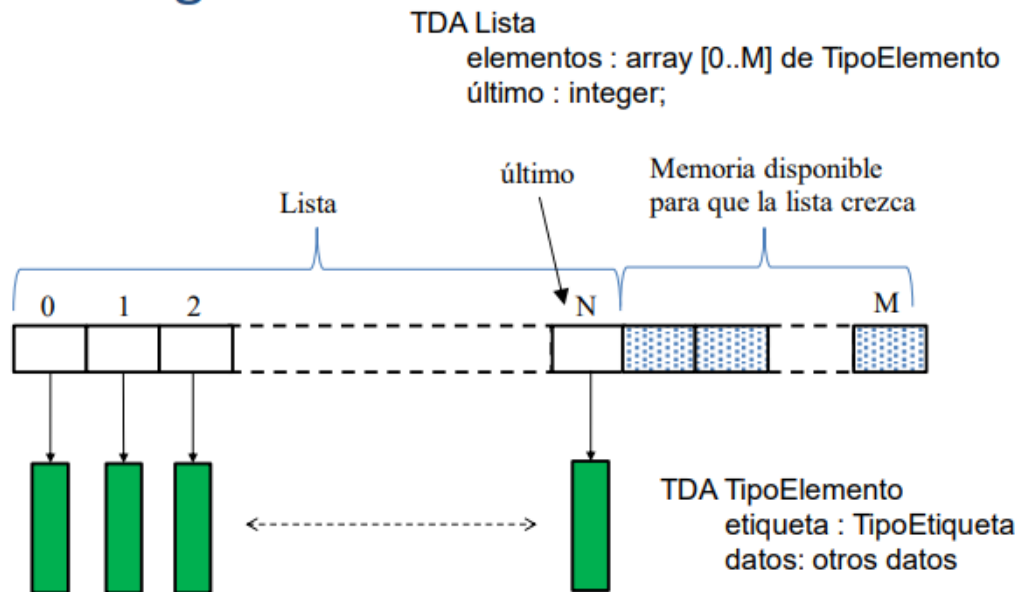
## UT3 – Listas, pilas y colas

### Listas – funcionalidades básicas



- Agregar un elemento “E” a una lista; la lista permite agregar un elemento en cualquier lugar:
  - Al final, el nuevo elemento será ahora el último de la lista
  - Al principio, el nuevo elemento será ahora el primero de la lista
  - En cualquier posición o de acuerdo a algún criterio (ejemplo: en una posición “x”, en orden alfabético ...)
- Eliminar un elemento “E” de una lista:
  - Sea que conozcamos al elemento “E”, o alguna característica del mismo (por ejemplo el número de la cédula de identidad)
  - Sea que queramos eliminar el elemento de cierta posición “x” (el primero, el último, el cuarto,...)
- Acceder a cierto elemento “E” para realizar alguna acción sobre el mismo
- Otras básicas:
  - saber si está vacía
  - saber cuántos elementos tiene
  - vaciarla

# Implementación del TDA Lista con arreglo



# Implementación del TDA Lista con referencias



- TDA Lista
  - Atributos:
    - Primero: TipoNodoLista;
  - Operaciones:
    - Inserta (E de TipoElemento, pos de TipoNodoLista): boolean
    - Elimina (pos de TipoNodoLista): TipoElemento
    - Busca (x de TipoEtiqueta): TipoNodoLista
- TDA Etiqueta
  - Atributos:
    - Etiqueta
  - Operaciones:
    - Imprimir
- TDA NodoLista
  - Atributos:
    - siguiente : TipoNodoLista;
    - elemento: TipoElemento
    - etiqueta: TipoEtiqueta
  - Operaciones:
    - imprimirEtiqueta
- TDA Elemento
  - Atributos:
    - Etiqueta: TipoEtiqueta
    - Datos: TipoDatos

- PRECONDICIONES
  - **Estado** en que *debe* encontrarse el objeto *antes* de comenzar el algoritmo
  - Lenguaje natural
  - Especificación rigurosa
  - Mapeo con el código!!!
  - Ayudan a escribir los casos de prueba
- POSTCONDICIONES
  - **Estado** en que *ha de quedar* el objeto *después* de terminar el algoritmo
  - **Facilitan** la escritura de los casos de prueba.

## LISTAS: Algoritmos de ejemplo con pseudocódigo TDA



- Implementar las operaciones básicas:
  - Inserta (E de TipoElemento, pos de TipoNodoLista): Boolean // Si "pos"
    - es nulo, inserta al final
    - es primero, inserta al principio
    - sino inserta, el elemento como siguiente de "pos"
    - devuelve verdadero si la inserción fue exitosa
  - Elimina (pos de TipoNodoLista): TipoElemento
    - luego de la eliminación, "pos" será el que era su siguiente
    - devuelve el elemento apuntado por "pos"
  - Busca (x de TipoEtiqueta): TipoNodoLista
    - devuelve la posición del elemento con etiqueta "x"

## Pilas



- Tipo especial de lista en que todas las inserciones y eliminaciones se hacen en el mismo extremo, denominado tope.
- Listas LIFO (Last In First Out).
- Operaciones normales para el TDA Pila:
  - Anula
  - Tope
  - Saca – “pop”
  - Mete(unElemento) – “push”
  - Vacía

- Tipo especial de lista en el que los elementos se insertan en un extremo (**el posterior**) y se extraen por el otro (**el anterior o frente**).
- Listas **FIFO** (First In First Out).
- Operaciones:
  - Anula
  - Frente
  - PoneEnCola(unElemento)
  - QuitaDeCola
  - Vacía

## Resolución del Problema: PASO 2(2)



De tipo TLista **InsertarOrdenado**(de tipoElemento elElemento);

COM

```
unNodo ← nuevoNodo (elElemento)
Si Vacía () entonces
    primero ← unNodo           // será el primero
    salir
FinSi
Si unNodo.clave < primero.clave // va al principio
    unNodo.Siguiente ← primero
    primero ← unNodo
    salir
Fin Si
tempNodo ← primero
Mientras (tempNodo.Siguiente() <> nulo Y tempNodo.Siguiente.Clave > unNodo.clave) hacer
    tempNodo ← tempNodo.Siguiente
finMientras
unNodo ← tempNodo.siguiente
tempNodo.Siguiente ← unNodo
```

FIN