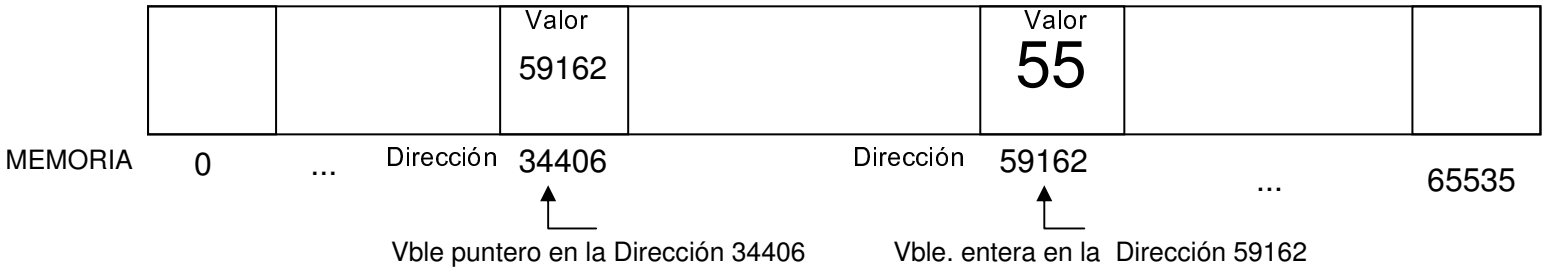
Estructuras dinámicas de datos

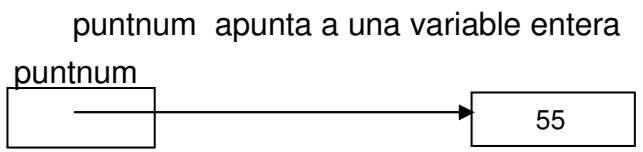
- Utilidad:
  - Todas las estructuras de datos vistas hasta ahora son **estáticas**, esto es, no pueden cambiar su tamaño durante la ejecución del programa. Cuando las estructuras de datos cambian de tamaño durante la ejecución del programa se utilizan las estructuras **dinámicas** de datos.
- En este tema se verán tres estructuras dinámicas de datos:
  - **Listas**: Colección de elementos del mismo tipo, organizados arbitrariamente con un orden ya definido y con la capacidad de inserción y eliminación de sus elementos.
  - **Pila**: Lista en la cual la inserción y eliminación de elementos se realizan únicamente al final de la lista
  - **Cola**:Lista en la cual la inserción de un elemento se hace por un extremo mientras que la eliminación se realiza por el otro extremo.
- Las estructuras dinámicas de datos se construyen utilizando punteros.

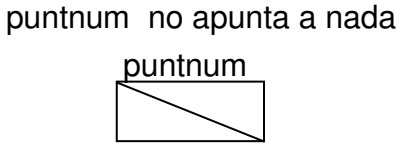
Punteros

- ¿Qué es un puntero?
  - Un puntero es una variable que sirve para almacenar la dirección en memoria de otra variable.
  - Ejemplo gráfico:



- Representación gráfica:





Punteros: Declaración y uso (1)

- Declaración de punteros

```
var
    variable_puntero:^nombre_tipo;
```

```
type
    tipo_puntero=^nombre_tipo;
var
    variable_puntero:tipo_puntero;
```

- Ejemplos de declaración de punteros
  - Declaración de un puntero `p` a una variable de tipo real

```
var p: ^real;
```

  - Declaración de un puntero `p` a una variable de tipo real

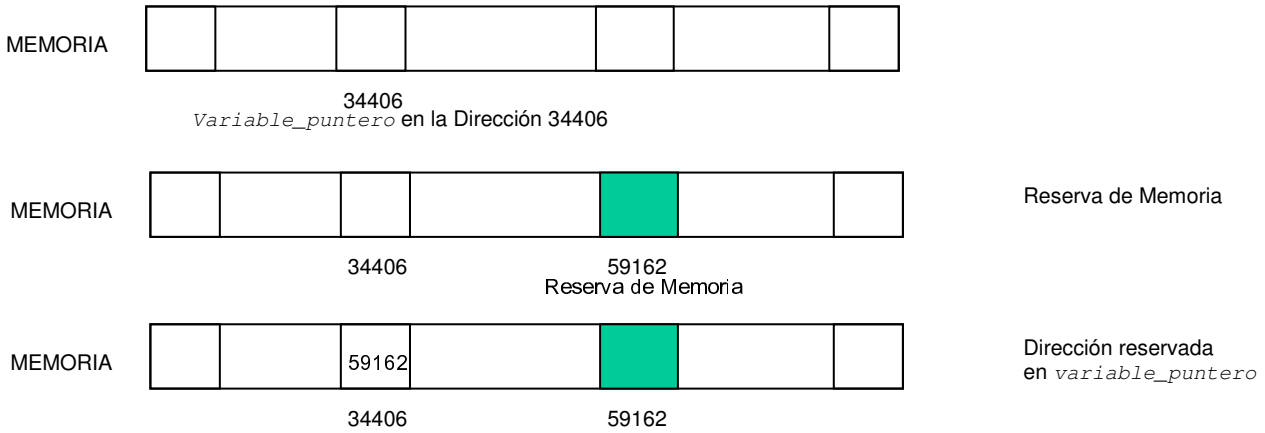
```
type puntero = ^real;
var p: puntero;
```

  - Declaración de un puntero `q` a una variable de tipo registro

```
type registro = record
          nombre:string[10];
          edad :integer;
          sexo :char
          end;
var q :^registro;
```

Punteros: Declaración y uso (2)

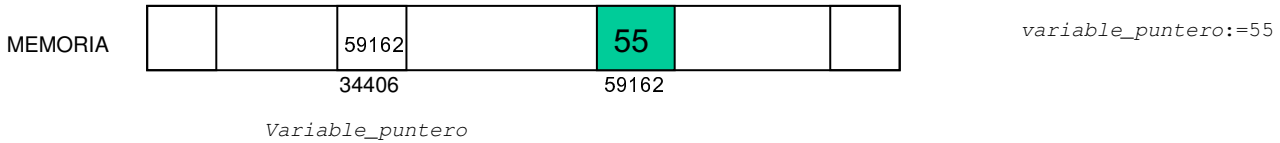
- Inicialización de punteros
  - Para que apunte a una variable o estructura de datos: **new**(*variable_puntero*);
    - Características
      - Al ejecutarse **new** se asigna memoria a una nueva variable del tipo al que apunta *variable_puntero* y se coloca la dirección de esa nueva variable en *variable_puntero*.



  - Para que no apunte a ninguna variable: *variable_puntero*=**nil**;

Punteros: Declaración y uso (3)

- Asignación de contenidos **a la variable apuntada** por un puntero**.**
  - Para darle un valor a la variable a la que apunta *variable_puntero* utilizamos *variable_puntero^:=valor*.



- Operar con el **valor de la variable a la que apunta** un puntero
  - Para utilizar el valor de la variable a la que apunta *variable_puntero* utilizamos *variable_puntero^*

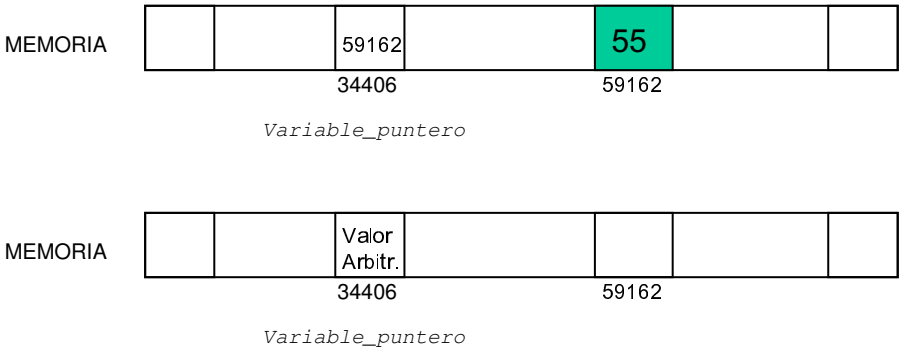
Punteros: Declaración y uso (4)

- Eliminar la variable apuntada por un puntero:

  Se llama al procedimiento **`dispose`**`(`*`variable_puntero`*`);`

  – Características

  - Libera la memoria a la que apuntaba `variable_puntero` y deja indefinido (con un valor arbitrario) al puntero.



Punteros: Declaración y uso (4)

- Ejemplo de utilización de punteros:

```
program ejemplo;
   var
       puntcar1,puntcar2:^char;
   begin
       new(puntcar1);
       puntcar2:=nil;
       puntcar1^:='A';
       puntcar2:=puntcar1;
       writeln('Los contenidos de puntcar1
       y puntcar2 son ',puntcar1^,' y ',puntcar2^);
       dispose(puntcar1);
   end.
```

Listas

- ¿Qué es una lista?
  - Es una colección de elementos del mismo tipo llamados **nodos,** organizados arbitrariamente con un orden ya definido y con la capacidad de inserción y eliminación de sus elementos.
  - Para cada elemento hay un **predecesor** y un **sucesor**. El precedesor del primer elemento y el sucesor del último es el elemento vacío.
- Características de las listas
  - Las listas se **caracterizan** por:
    - La forma en que se encadenan los elementos (esto es, la forma en que se identifican el predecesor y sucesor de un elemento). Para ello se suelen utilizar los punteros.
    - Cómo y dónde se pueden insertar y eliminar sus elementos.
    - El número de predecesores y sucesores de cada elemento.

Listas:Operaciones Básicas (1)

- Definición de un nodo:

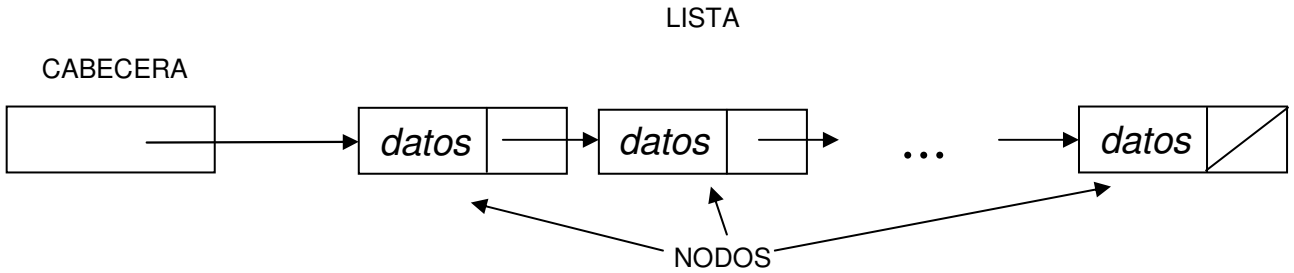
```
type
    ptrnodo=^tiponodo;
    tiponodo= record
              datos:tipo_datos;
              siguiente:ptrnodo;
              end;
```

- Declaración de la cabecera de una lista:

```
var
   cabecera_lista:ptrnodo;
```

- Inicialización de la cabecera de una lista:

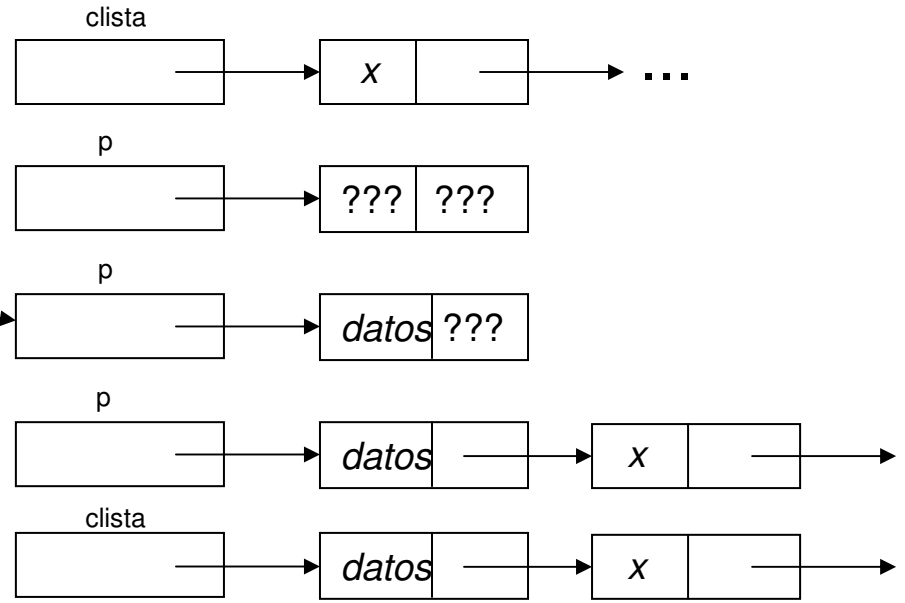
```
   cabecera_lista:=nil;
```



Listas:Operaciones Básicas (2)

- Inserción de un nodo en la cabecera de una lista (en su principio)

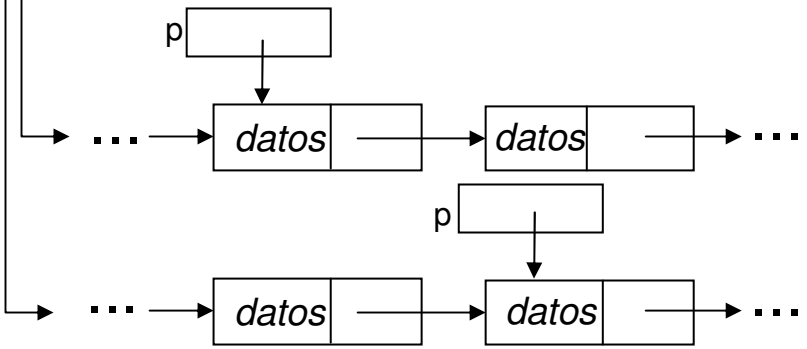
```
var
   p:ptrnodo;
   clista:ptrnodo;
begin
                ...
   new(p);
   p^.datos:=datos;
   p^.siguiente:=clista;
   clista:=p;
      ...
```



Listas:Operaciones Básicas (3)

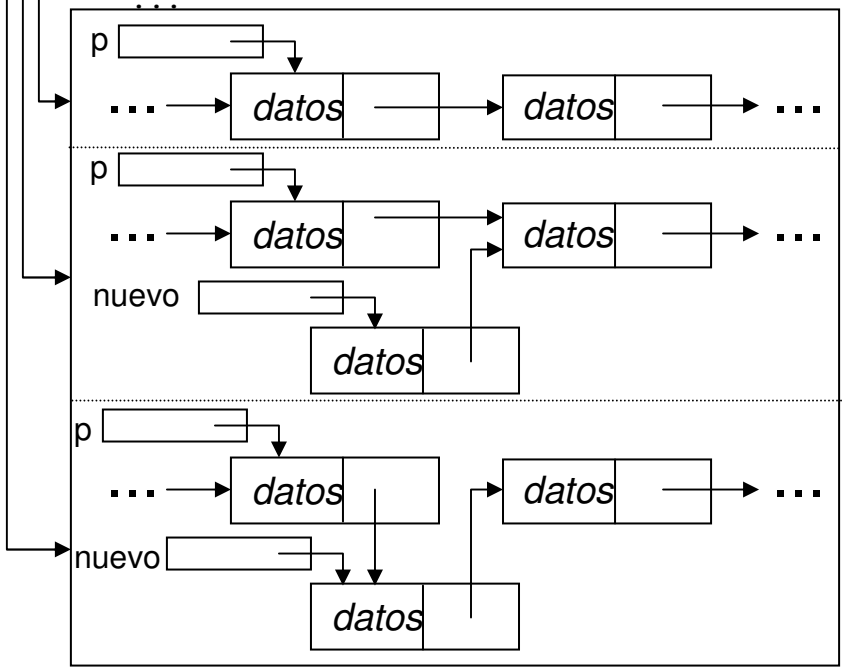
- Procesamiento de los datos de una lista

```
var
    p:ptrnodo;clista:ptrnodo;
...
p:=clista
while p<>nil do
begin
    procesa(p^datos);
    p:=p^siguiente;
end
...
```



- Inserción en un lugar arbitrario

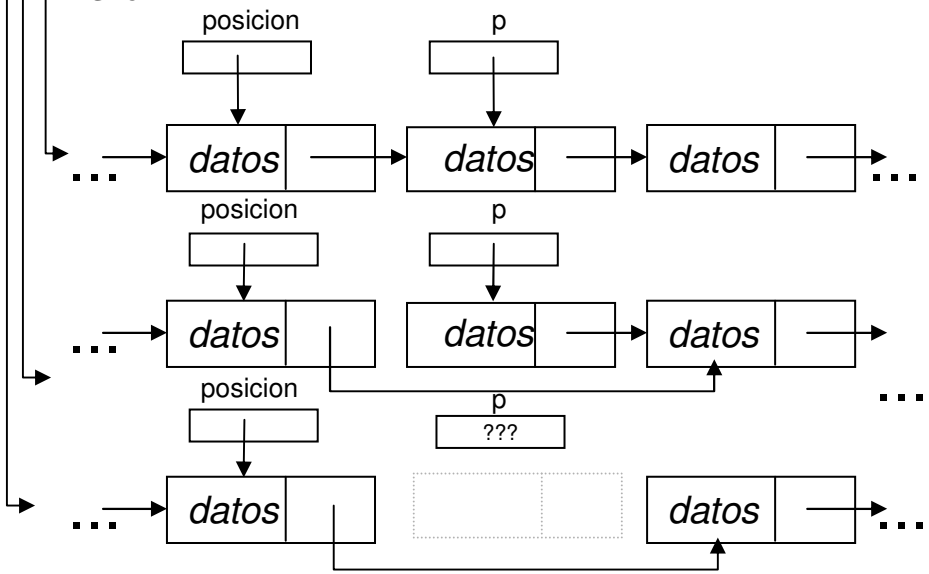
```
var
    p,nuevo,clista:ptrnodo;
...
new(nuevo);
nuevo^.datos:=datos;
nuevo^.siguiente:=p^.siguiente;
p^.siguiente:=nuevo;
...
```



Listas:Operaciones Básicas (4)

- Eliminación de un elemento

```
var
        posicion,p:ptrnodo;
  ...
if posicion^.siguiente<>nil then
begin
        p:=posicion^.siguiente;
        posicion^.siguiente:=p^.siguiente
        dispose(p);
end
```

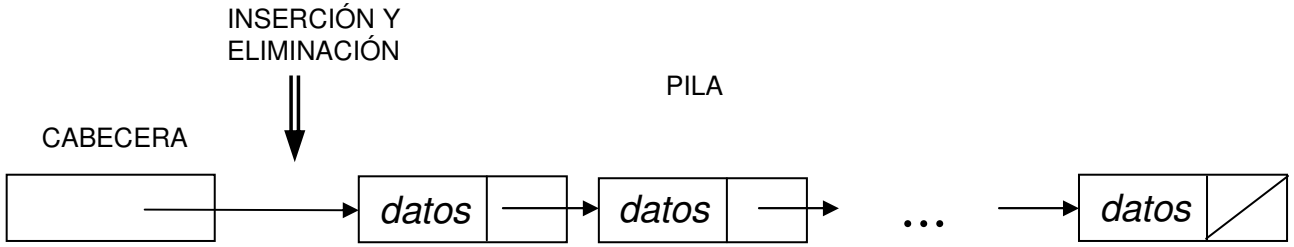


- Eliminación de un elemento al final de una lista.

```
var
        posicion,p,lista:ptrnodo;
  ...
if lista<>nil then
begin
        posicion:=lista;
        p:=lista;

        while p^siguiente<>nil do
        begin
                posicion:=p;
                p:=posicion^.siguiente;
        end

        if p=posicion then
                lista=nil;
        else

        posicion^.siguiente=nil;
        dispose(p);

end
```

Pilas

- ¿Qué es un pila?
  - Las pilas son estructuras de datos en las que las inserciones, inspecciones y eliminaciones ocurren solo al principio, o tope de la pila; cuando se agrega un nodo al tope de la pila se dice que lo metemos (push) y cuando lo retiramos se dice que lo sacamos (pop).
  - Las pilas se implementan como un caso especial de las listas.



Colas

- ¿Qué es una cola?
  - Las colas son estructuras de datos en las que las inserciones ocurren al final y eliminaciones ocurren solo al principio
  - Las colas se implementan como un caso especial de las listas.