

# Punteros y Listas

- Uso de Memoria.
- Asignación Estática y Dinámica.
- Creación por Demanda.
- Tipo de Registro con puntero.
- Manejo de punteros:
  - Vincular dos nodos.
  - Carga de una lista de longitud variable.
  - Baja de un Nodo.
  - Alta de un Nodo.
- Tipos de Listas (Simple, Circular y Doble).
- Punteros pasados como parámetros.

# Tipos de Asignación de Memoria

## Estática

- ✓ Se asigna memoria cuando se define la variable.
- ✓ No se puede utilizar más de lo definido.
- ✓ Tampoco menos, es necesario definir límites !
- ✓ Ejemplos: Variables, Arreglos, Matrices, etc.

## Dinámica

- ✓ Se asigna memoria en ejecución (por demanda)
- ✓ No se asigna ni más ni menos de lo necesario.
- ✓ Su uso es un poco más engorroso.
- ✓ Punteros

# Asignación de Memoria Estática

Type .....

Var .....

Begin .....  
End; .....



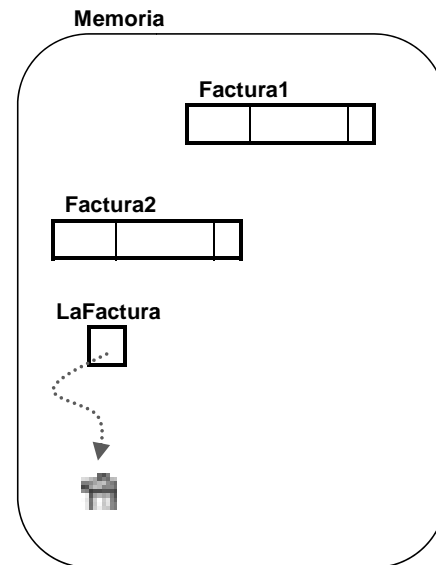
## Asignación de Memoria en Ejecución

```
Type TipoFactura = record
  Nro : Integer;
  Importe : Real;
  IVA: Char
End;
```

```
Var Factura1 : TipoFactura;
```

```
Var Factura2 : TipoFactura;
```

```
Var LaFactura : ^TipoFactura;
```



## Asignación de Memoria en Ejecución

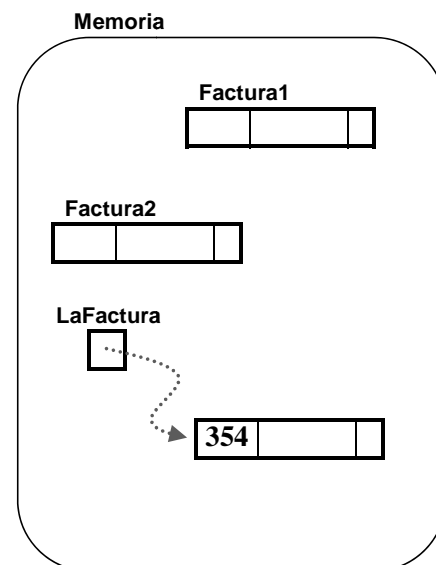
```
Type TipoFactura = record
  Nro : Integer;
  Importe : Real;
  IVA: Char
End;
```

```
Var Factura1 : TipoFactura;
```

```
Var Factura2 : TipoFactura;
```

```
Var LaFactura : ^TipoFactura;
```

```
Begin
  ...
  new(LaFactura);
  ...
  LaFactura^.Nro := 354
```




---

---

---

---

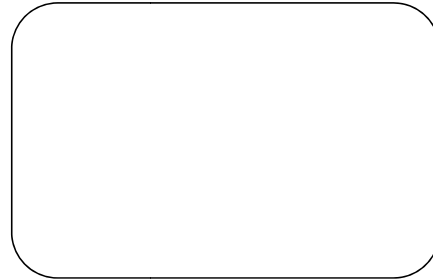
---

---

## Doble New

```
Type TipoFactura = record  
    Nro : Integer;  
    Importe : Real;  
    IVA: Char  
End;
```

Memoria

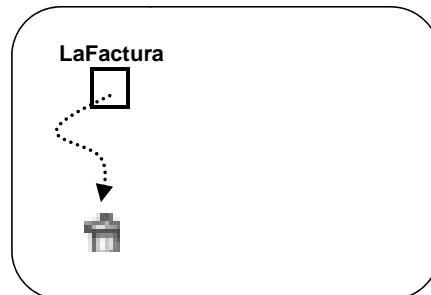


## Doble New

```
Type TipoFactura = record  
    Nro : Integer;  
    Importe : Real;  
    IVA: Char  
End;
```

```
Var LaFactura : ^TipoFactura;
```

Memoria



---

---

---

---

---

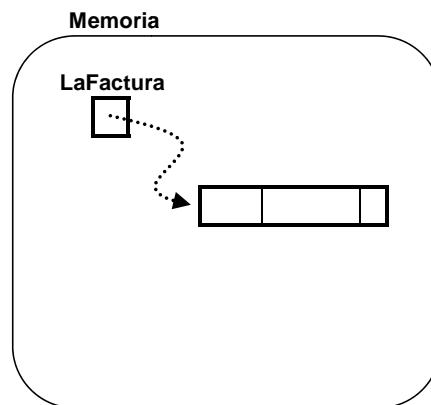
---

## Doble New

```
Type TipoFactura = record
    Nro : Integer;
    Importe : Real;
    IVA: Char
End;

Var LaFactura : ^TipoFactura;

Begin
    ...
    new(LaFactura);
    ...
```

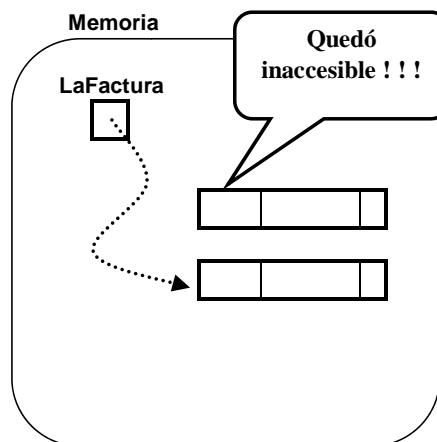


## Doble New

```
Type TipoFactura = record
    Nro : Integer;
    Importe : Real;
    IVA: Char
End;

Var LaFactura : ^TipoFactura;

Begin
    ...
    new(LaFactura);
    ...
    new(LaFactura);
```



---

---

---

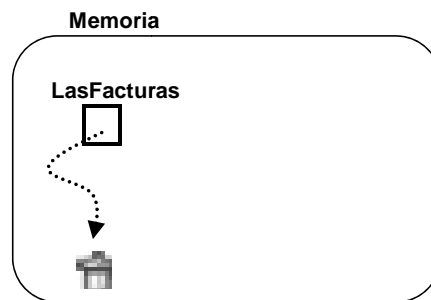
---

---

---

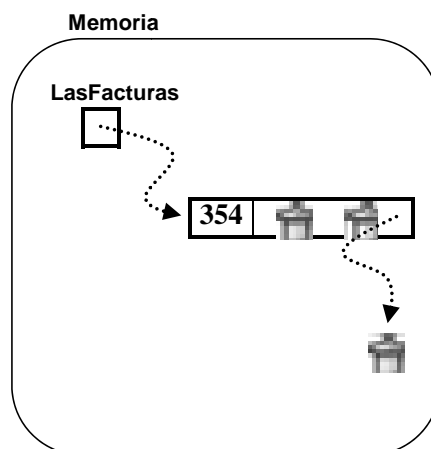
## Una lista de dos nodos

```
Type PuntFactura = ^TipoFactura;  
TipoFactura = record  
  Nro : Integer;  
  Importe : Real;  
  IVA: Char;  
  Sig: PuntFactura  
End;  
Var LasFacturas : PuntFactura;
```



## Una lista de dos nodos

```
Type PuntFactura = ^TipoFactura;  
TipoFactura = record  
  Nro : Integer;  
  Importe : Real;  
  IVA: Char;  
  Sig: PuntFactura  
End;  
Var LasFacturas : PuntFactura;  
  
Begin  
  ...  
  new(LasFacturas);  
  LasFacturas^.Nro := 354;  
  ...
```



---

---

---

---

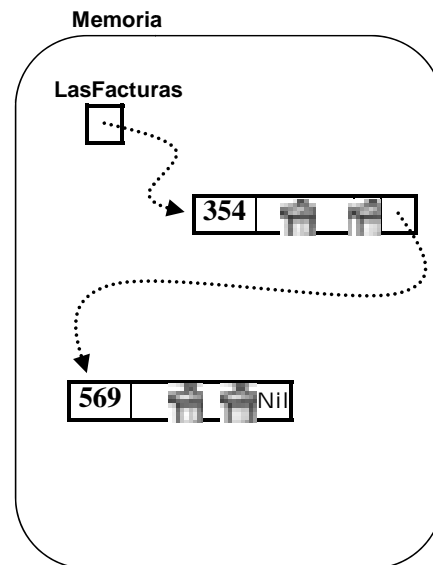
---

---

## Una lista de dos nodos

```
Type PuntFactura = ^TipoFactura;  
TipoFactura = record  
  Nro : Integer;  
  Importe : Real;  
  IVA: Char;  
  Sig: PuntFactura  
End;  
Var LasFacturas : PuntFactura;
```

```
Begin  
  ...  
  new(LasFacturas);  
  LasFacturas^.Nro := 354;  
  ...  
  new(LasFacturas^.Sig);  
  LasFacturas^.Sig^.Nro := 569;  
  ...  
  LasFacturas^.Sig^.Sig := Nil;
```



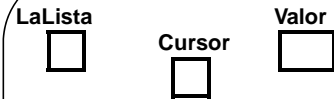


# Una lista de longitud variable

```

Program CreaLista;
{ Este programa crea una lista de valores enteros con tantos nodos como valores
distintos de cero ingresa el usuario, luego imprime todos los nodos }
Type PuntNodo = ^TipoNodo;
TipoNodo = Record
    Nro: Integer;
    Sig: PuntNodo;
End;
Var LaLista, Cursor : PuntNodo;
    Valor : Integer;
Begin
    LaLista := Nil;
    Readln(Valor);
    if (Valor > 0) then Begin
        New(LaLista);
        LaLista^.Sig := Nil;
        LaLista^.Nro := Valor;
        Cursor := LaLista;
        Readln(Valor);
        While (Valor > 0) do Begin
            New(Cursor^.Sig);
            Cursor := Cursor^.Sig;
            Cursor^.Sig := Nil;
            Cursor^.Nro := Valor;
            Readln(Valor);
        End
    End;
    Cursor := LaLista;
    While (Cursor <> Nil) do Begin
        Writeln(Cursor^.Nro);
        Cursor := Cursor^.Sig;
    End;
    ReadLn(Valor);
End.
    
```

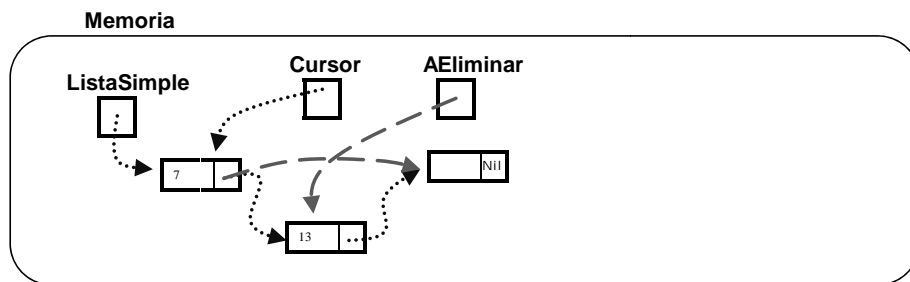
## Memoria



## Detalles a tener en cuenta

- ✓ Diferenciar el puntero principal de los auxiliares para recorrido.
- ✓ Cuidar el uso del principal, nunca perder el primer nodo.
- ✓ Ponerle nombres claros a los punteros para no confundirse ni confundir.
- ✓ Inmediatamente después de crear un nodo inicializar sus campos de punteros con (New o Nil)

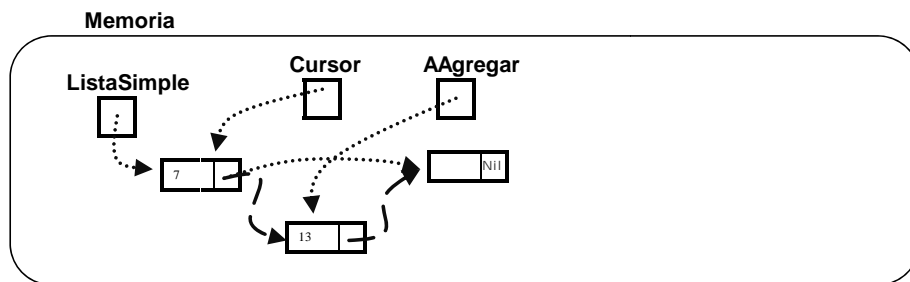
## Baja de un Nodo



### Baja

```
...  
AEliminar := Cursor^.Sig;  
Cursor^.Sig := Cursor^.Sig^.Sig;  
Dispose(AEliminar);  
...
```

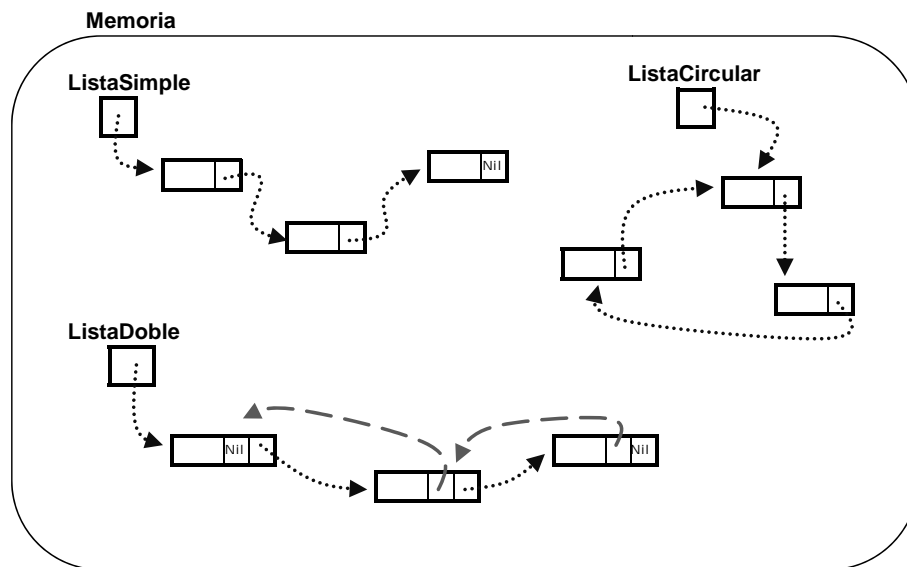
## Alta de un Nodo



### Alta

```
...  
AAgregar^.Sig := Cursor^.Sig;  
Cursor^.Sig := AAgregar;  
...
```

# Tipos de Lista



# Pasaje de Parámetros

Lo que se pasa por parámetro es la variable de tipo Puntero.

**Por Referencia:** Se pasa la propia variable, por lo tanto toda modificación impactará de igual forma que si estaría en el módulo invocador.

**Por Valor:** Se pasa una copia de la variable, toda modificación de SU valor no impactará pero todo cambio en los Nodos de la Lista SI.