

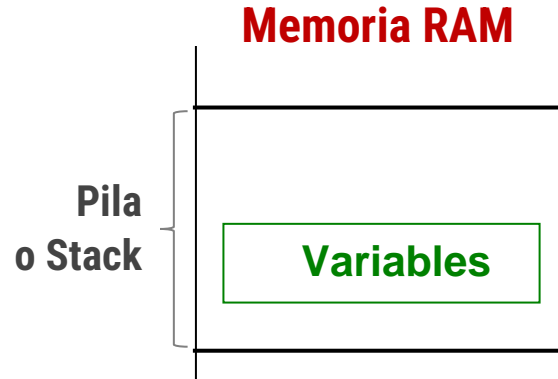
# Alocación dinámica

## Punteros

Explicación P5

# VARIABLES DINAMICAS

Las variables estáticas ocupan un espacio de memoria que se reserva al declararlas y que no cambia durante la ejecución del programa.



# VARIABLES DINAMICAS

**Variable Puntero:** variable estática que almacena la dirección en memoria de otra variable (*llamada variable dinámica*).

miVariablePuntero

Direcciones de  
memoria

Memoria

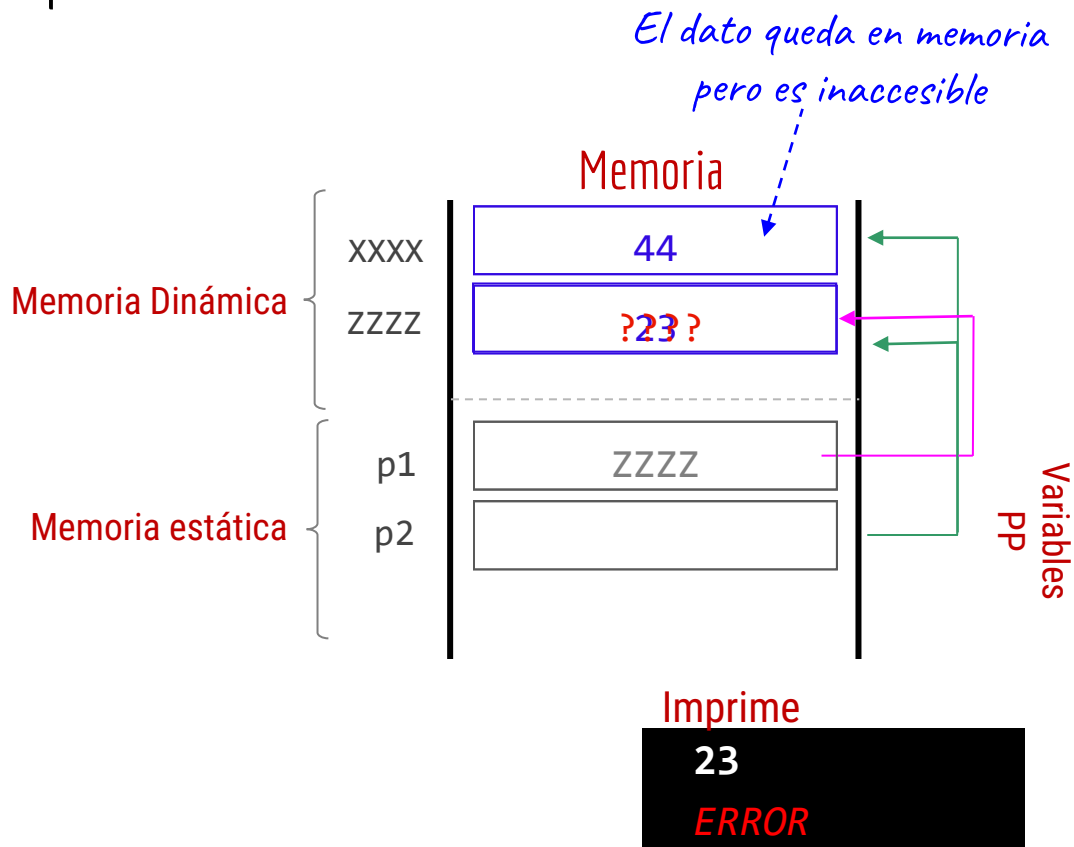
		Memoria Dinámica
xxxxxxxxxx		
xxxxxxxxxx		
5623	Dato	Memoria Dinámica
xxxxxxxxxx		
xxxxxxxxxx		
xxxxxxxxxx		Memoria Dinámica
xxxxxxxxxx		
xxxxxxxxxx		
xxxxxxxxxx		Memoria Dinámica
xxxxxxxxxx		
xxxxxxxxxx		
xxxxxxxxxx	5623	Memoria Estática
xxxxxxxxxx		
xxxxxxxxxx		

# USO DE PUNTEROS

Declaración <i>(ejemplo)</i>	Valores posibles de un puntero
<pre>TYPE   PunteroEntero = ^ integer;  VAR   pun, otropun: PunteroEntero;</pre>	<ul style="list-style-type: none"><li>• NIL</li><li>• Dirección de memoria dinámica</li></ul> <h2 data-bbox="625 609 1273 655">¿Qué se puede hacer con punteros?</h2> <ul style="list-style-type: none"><li>• Reservar memoria dinámica (<b>new</b>)</li><li>• Liberar memoria dinámica (<b>dispose</b>)</li><li>• Asignar punteros</li><li>• Comparar punteros</li><li>• Acceder al dato apuntado por el puntero (<b>^</b>)</li></ul>

# USO DE PUNTEROS - Ejemplo 1

```
Program ejemplo;  
Type  
  Ptro= ^integer;  
Var  
  → p1, p2: Ptro;  
Begin  
  → new (p1);  
  → p1^ := 23;  
  → new (p2);  
  → p2^ := 44;  
  → p2 := p1;  
  → write (p2^);  
  → dispose (p2);  
  → write(p1^);  
End.
```



# USO DE PUNTEROS - Ejemplo 2

Program ejemplo;

Type

  casa = record

    met\_cua: real;

    cant\_hab: integer;

  end;

  punt\_casa = ^casa;

Var

→ p1, p2: punt\_casa;

Begin

→ new (p1);

→ p1^.met\_cua := 125.50;

→ p1^.cant\_hab := 5;

→ p2:= p1;

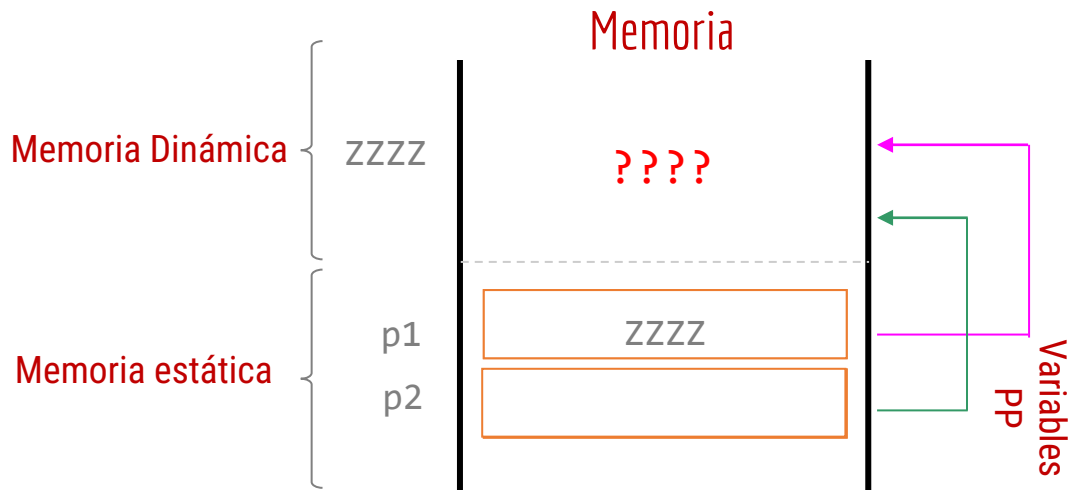
→ p2^.cant\_hab := 6;

→ write (p1^.cant\_hab);

→ dispose (p2);

→ write(p1^.met\_cua);

End.



Imprime

6

**ERROR**

# USO DE PUNTEROS - Ejemplo 3: Punteros como parámetros

*p1 es una copia de p (parámetro por valor). Pero el dato apuntado por ambos es el mismo. Si modifico p1^, también modifico p^*

```
Program ejemplo;  
Type  
  punt = ^integer;
```

```
→ Procedure suma (p1:punt);  
  Begin  
→   p1^ := p1^ + 1;  
→ End;
```

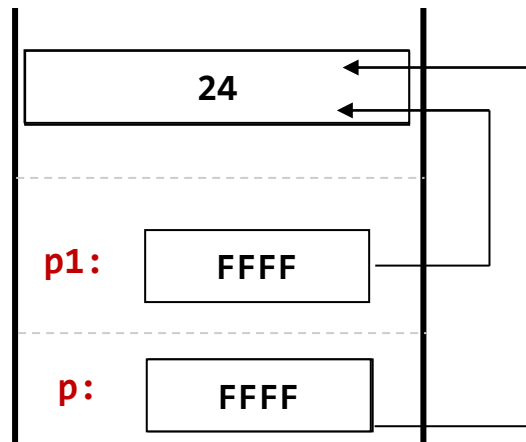
```
  Var  
→   p: punt;
```

```
  Begin  
→   new (p);  
→   p^ := 23;  
→   suma(p);  
→   write(p^);  
  End.
```

Variables dinámicas

Variables  
estáticas del  
Proceso suma

Variables estáticas  
del Prog.Ppal



Imprime

24

# USO DE PUNTEROS - Ejemplo 4: Punteros como parámetros

```
Program ejemplo;  
Type  
  punt = ^integer;
```

```
Procedure suma (p1:punt);  
Begin  
  p1^ := p1^ + 1;  
  new(p1);  
End;
```

```
Var  
  p: punt;
```

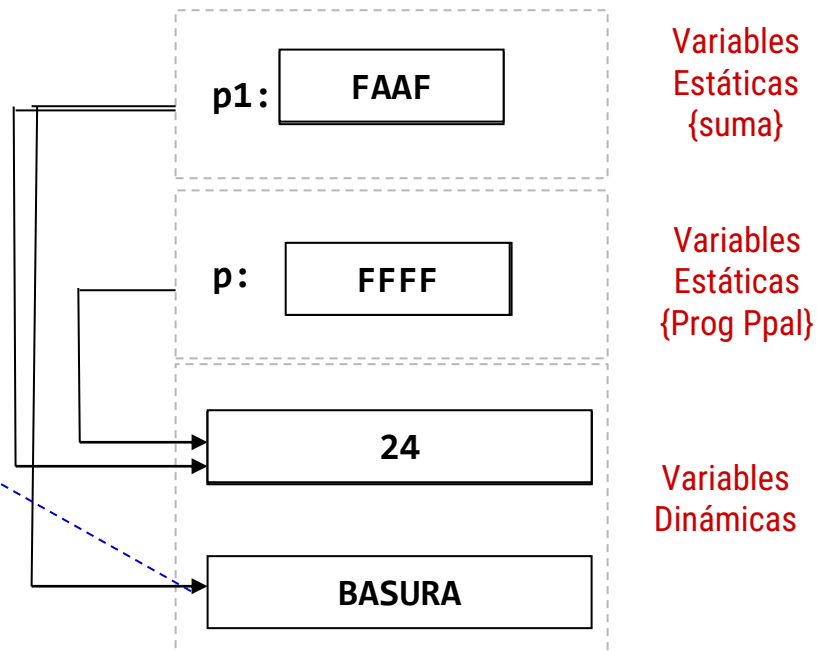
```
Begin  
  new (p);  
  p^ := 23;  
  suma(p);  
  write(p^);  
End.
```

*p1 es una copia de p. Si modifico p1, p en el programa ppal NO va a ver el cambio*

*¿Qué ocurre al reemplazar new(p1) por dispose(p1)?*

*El dato queda en memoria pero es inaccesible*

**IMPRIME 24**





## Ejemplo 5: Punteros como parámetros

```
Program ejemplo;  
Type  
  punt = ^integer;
```

```
Procedure suma (VAR p1:punt);  
Begin  
  p1^ := p1^ + 1;  
  new(p1);  
  p1^ := 44;  
End;
```

```
Var  
  p: punt;
```

```
Begin  
  new (p);  
  p^ := 23;  
  suma(p);  
  write(p^);  
End.
```

*p1 recibe la referencia de p. Si modifico p1, estoy modificando p en el programa ppal.*

*El dato queda en memoria pero es inaccesible*

**IMPRIME 44**

