

ALOCACIÓN DINÁMICA

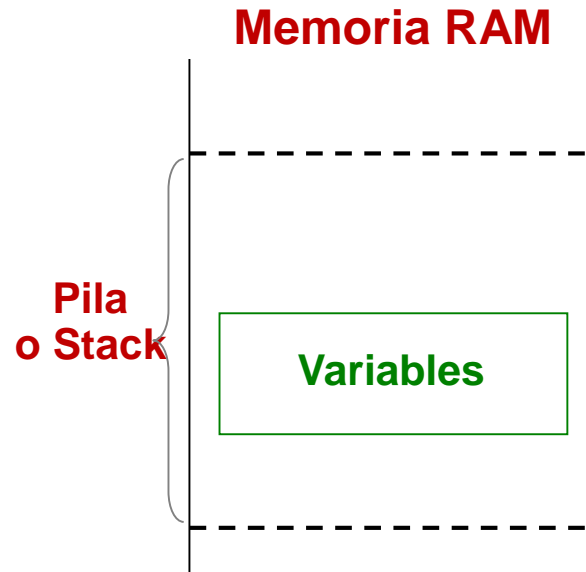
PUNTEROS

EXPLICACIÓN PRÁCTICA 6

CADP 2018

VARIABLES ESTÁTICAS

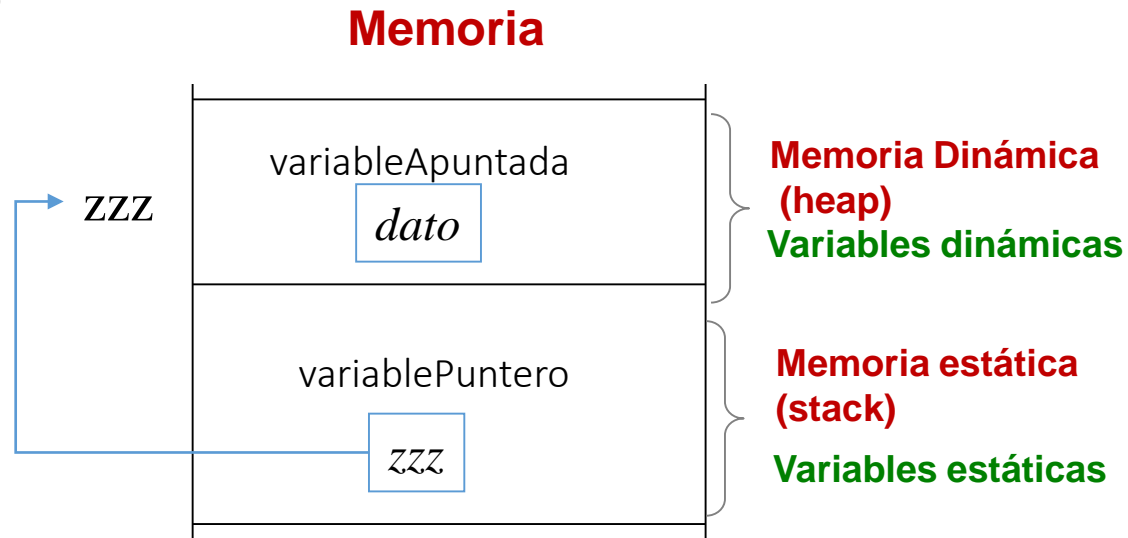
Las variables estáticas ocupan un espacio de memoria que se reserva al declararlas y que no cambia durante la ejecución del programa.



VARIABLES DINÁMICAS

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

Ejemplo



Declaración *(ejemplo)*

TYPE

PtroEntero = ^ integer

VAR

pun, otropun: PtroEntero

Valores posibles de un puntero

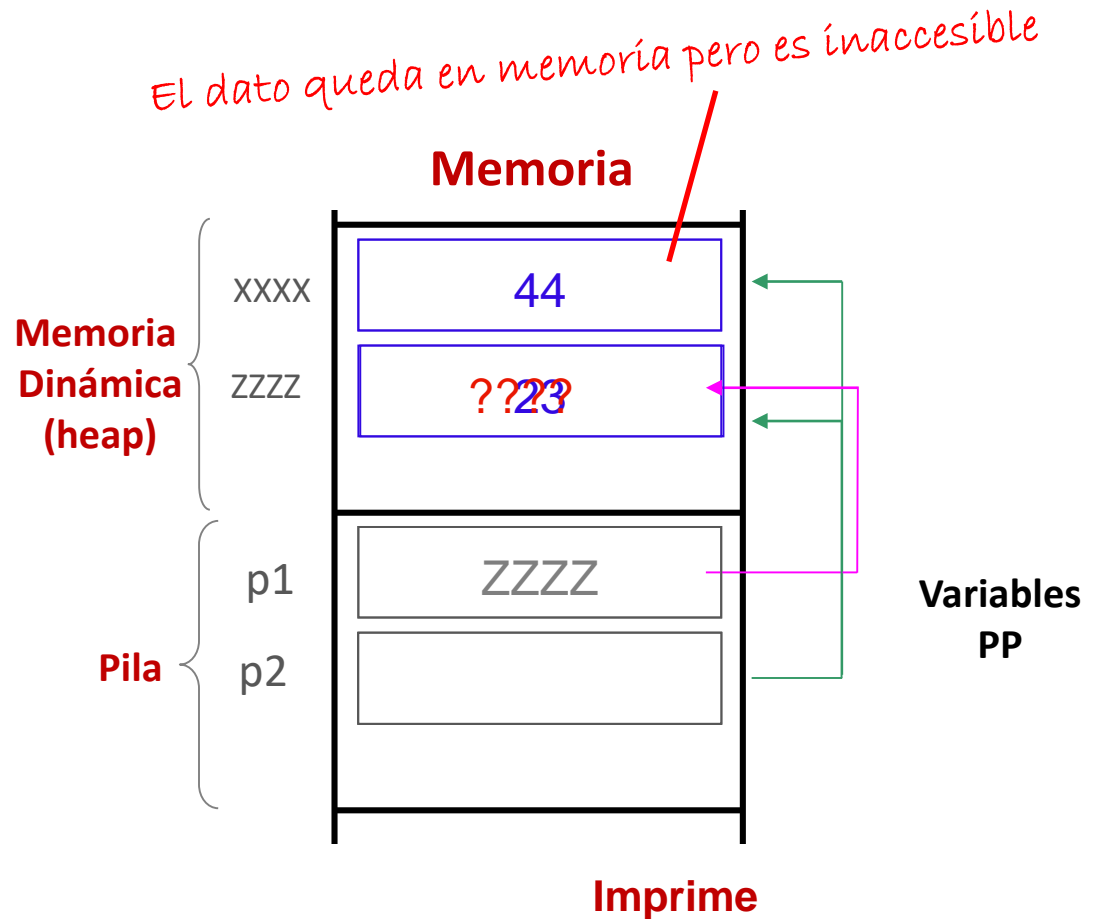
- NIL
- Dirección de memoria dinámica

¿Qué se puede hacer con punteros?

- Reservar memoria dinámica (**new**)
- Liberar memoria dinámica (**dispose**)
- Asignar punteros
- Comparar punteros
- Acceder al dato apuntado por el puntero (**^**)

Ejemplo

```
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.
```



23
ERROR

Ejemplo 2

```
Program ejemplo;
```

```
Type
```

```
  casa = record
    met_cua: real;
    cant_hab: integer;
  end;
  punt_casa = ^casa;
```

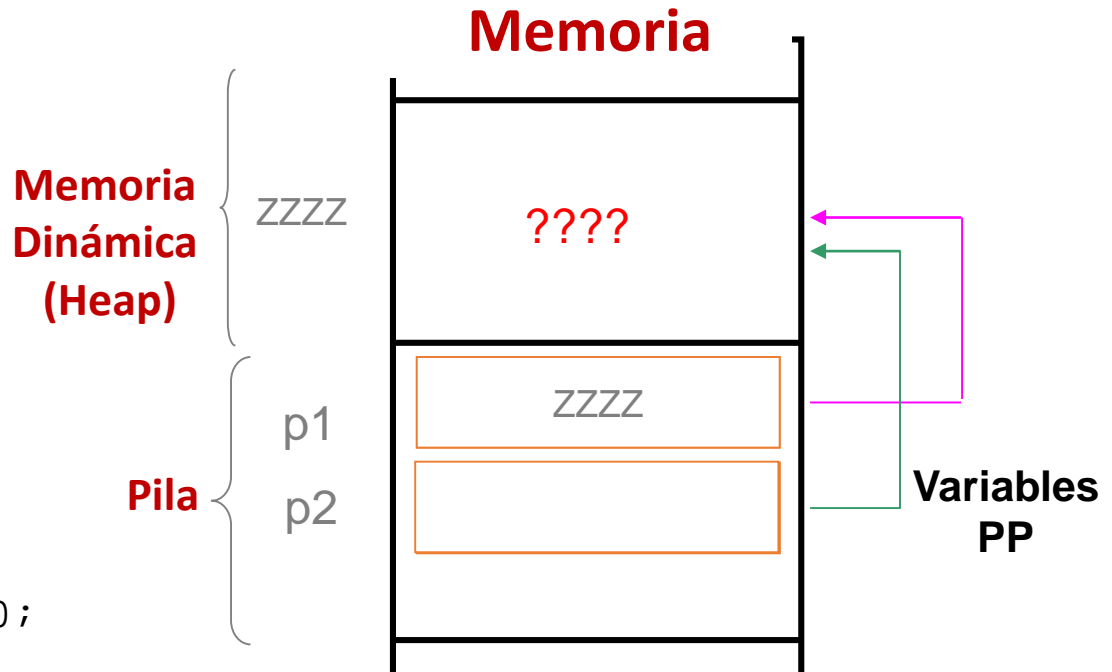
```
Var
```

```
→ 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

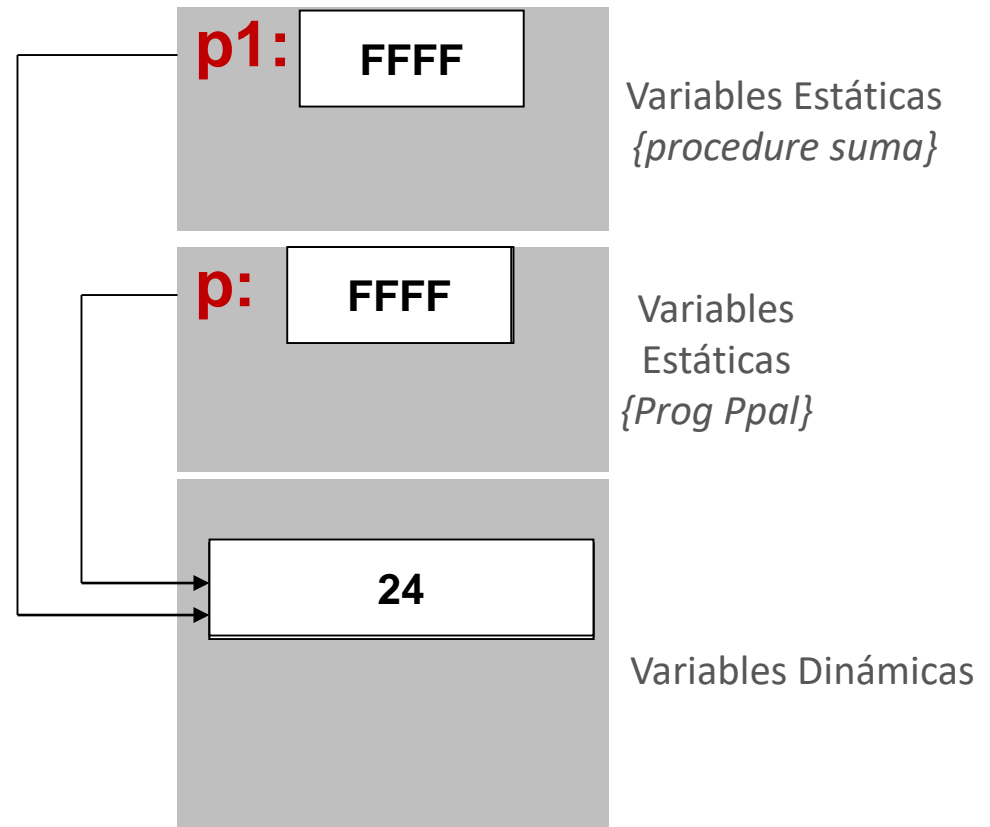
6

ERROR

Ejemplo 3: Punteros como parámetros

```
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.
```

p1 recibe una copia de la dirección de **p**. Si modifico el dato apuntado por **p1**, el llamador verá el cambio.



IMPRIME 24

Ejemplo 4: Punteros como parámetros

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

p1 recibe una copia de la dirección de **p**.
Si modifico la dirección **p1**, **p** en el programa ppal
tiene la misma dirección que tenía antes de la llamada.

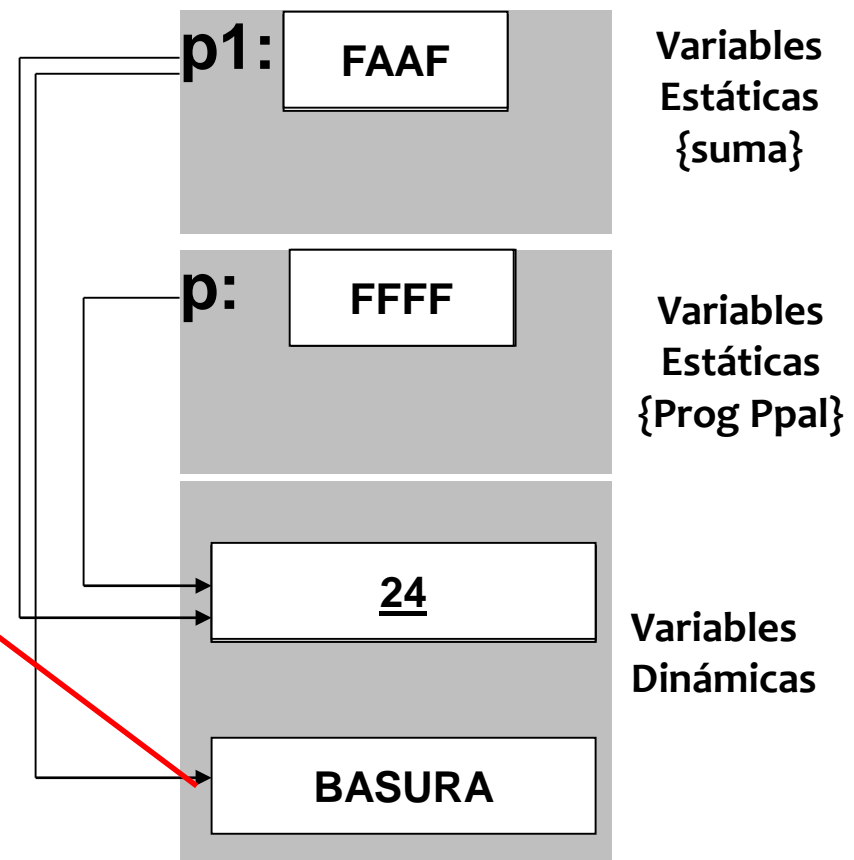
```
Procedure suma (p1:punt);  
Begin  
  p1^ := p1^ + 1;  
  new(p1);  
End;  
  
Var  
  p: punt;
```

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

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

IMPRIME 24

**El dato queda en
memoria pero
es inaccesible**



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.
```

IMPRIME 44

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

El dato queda en memoria pero es inaccesible

