

## C5: datos más dinámicos

C5'

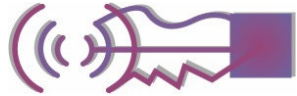
registro de activación  
de tamaño conocido  
al activar.

C5''

puede alocarse  
durante la  
ejecución.

Datos  
semidinámicos

Datos  
dinámicos



## C5': datos semidinámicos

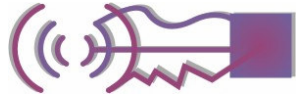
- **Arreglos dinámicos**

*type VECTOR is array (INTEGER range <>);*

*A: VECTOR (0..N);*

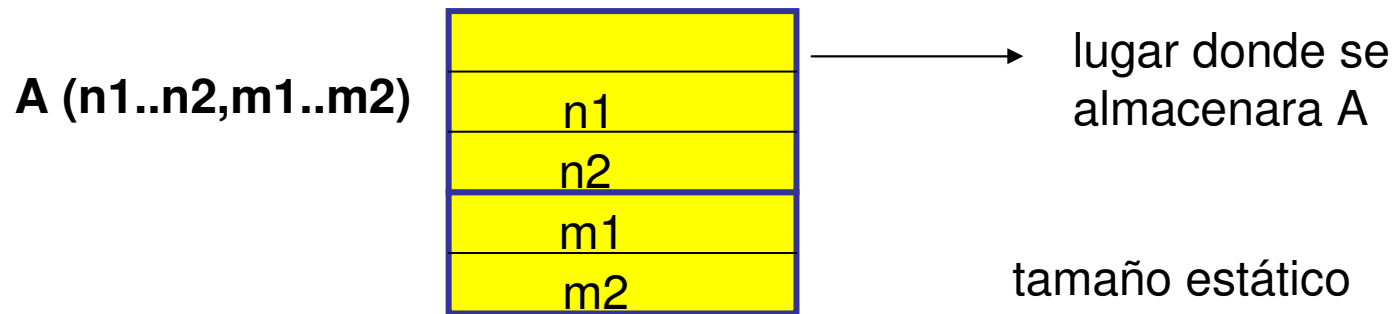
*B: VECTOR(1..M);*

N y M conocidos cuando se invoca la unidad

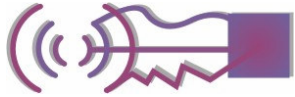


implementación: arreglos dinámicos

- **COMPILACION: descriptores**

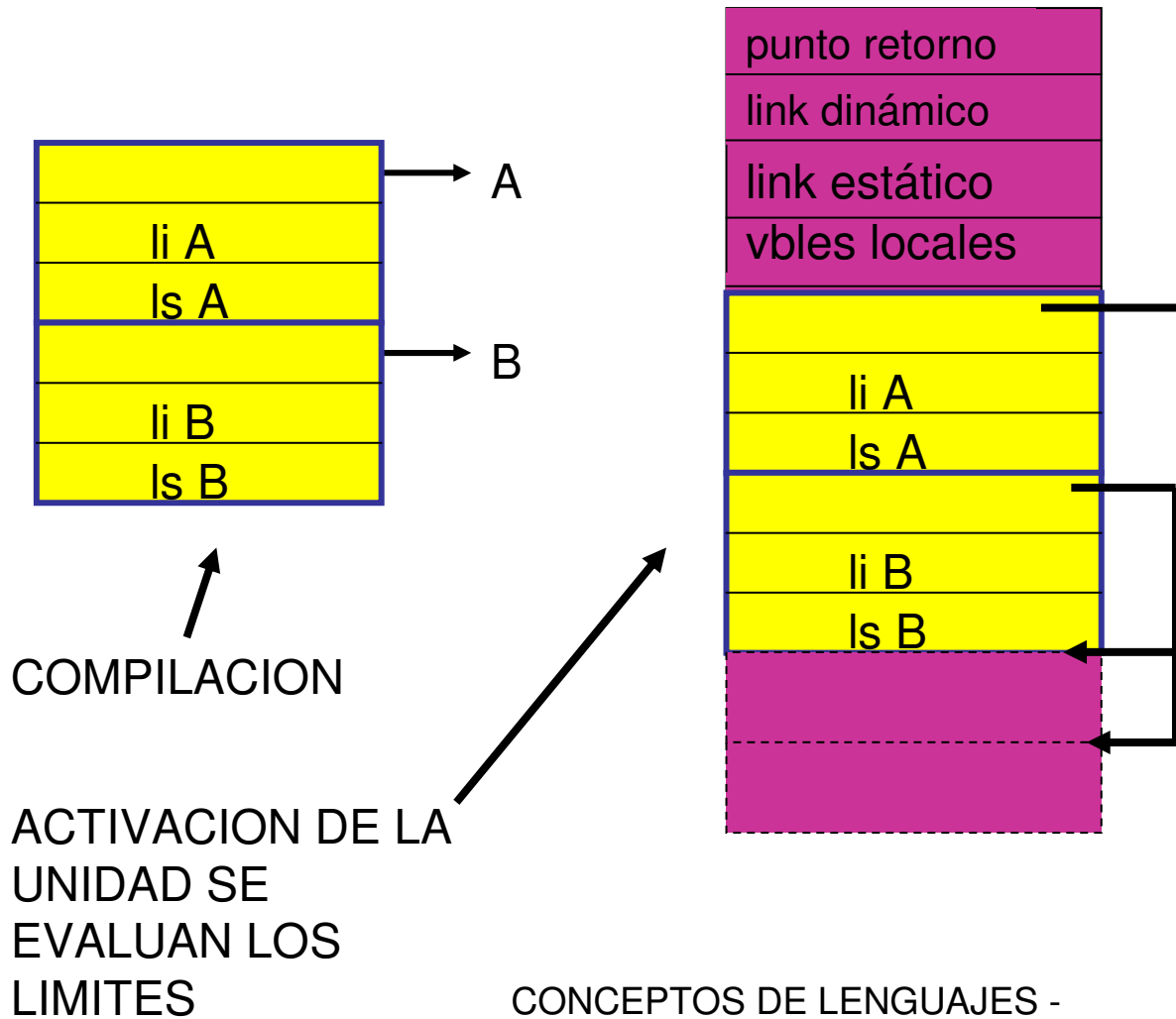


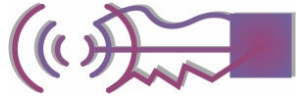
Todos los accesos al arreglo son traducidos como referencias indirectas



*A: VECTOR (0..N);*  
*B: VECTOR(1..M);*

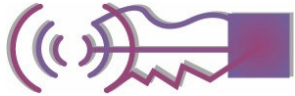
## EJECUCION





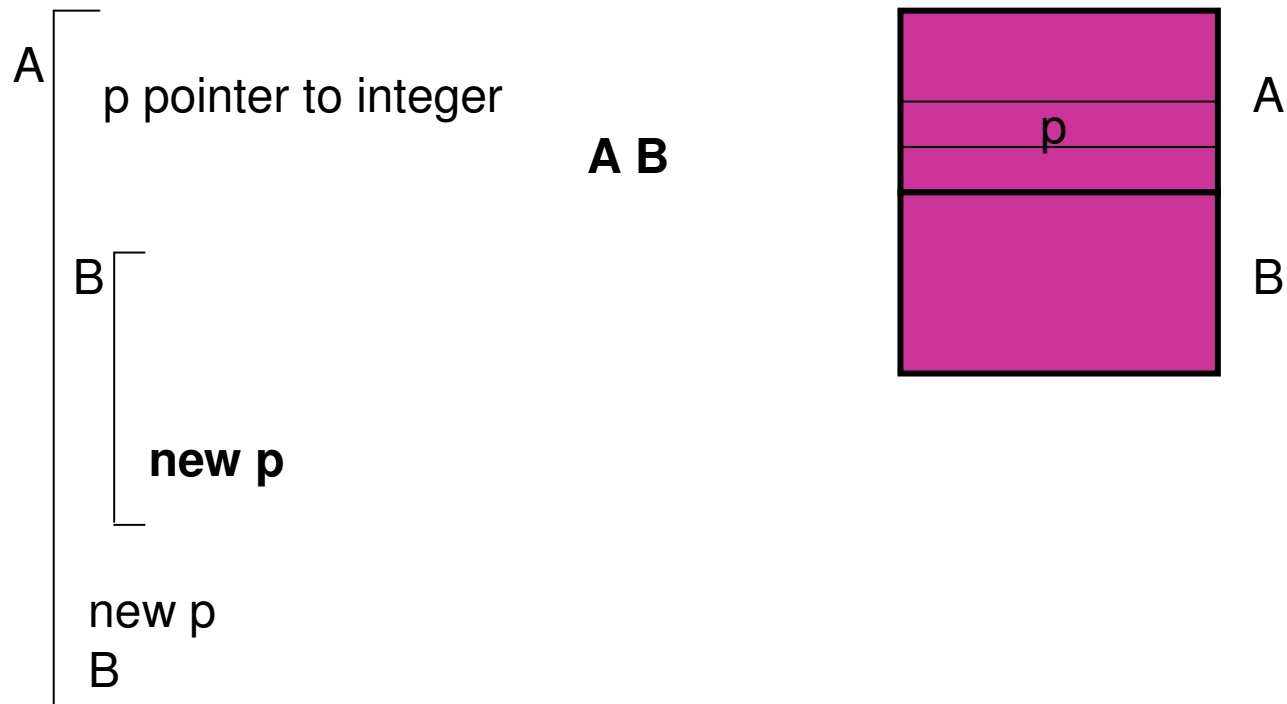
## C5'': datos dinámicos

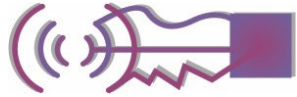
- se alocan explícitamente durante la ejecución



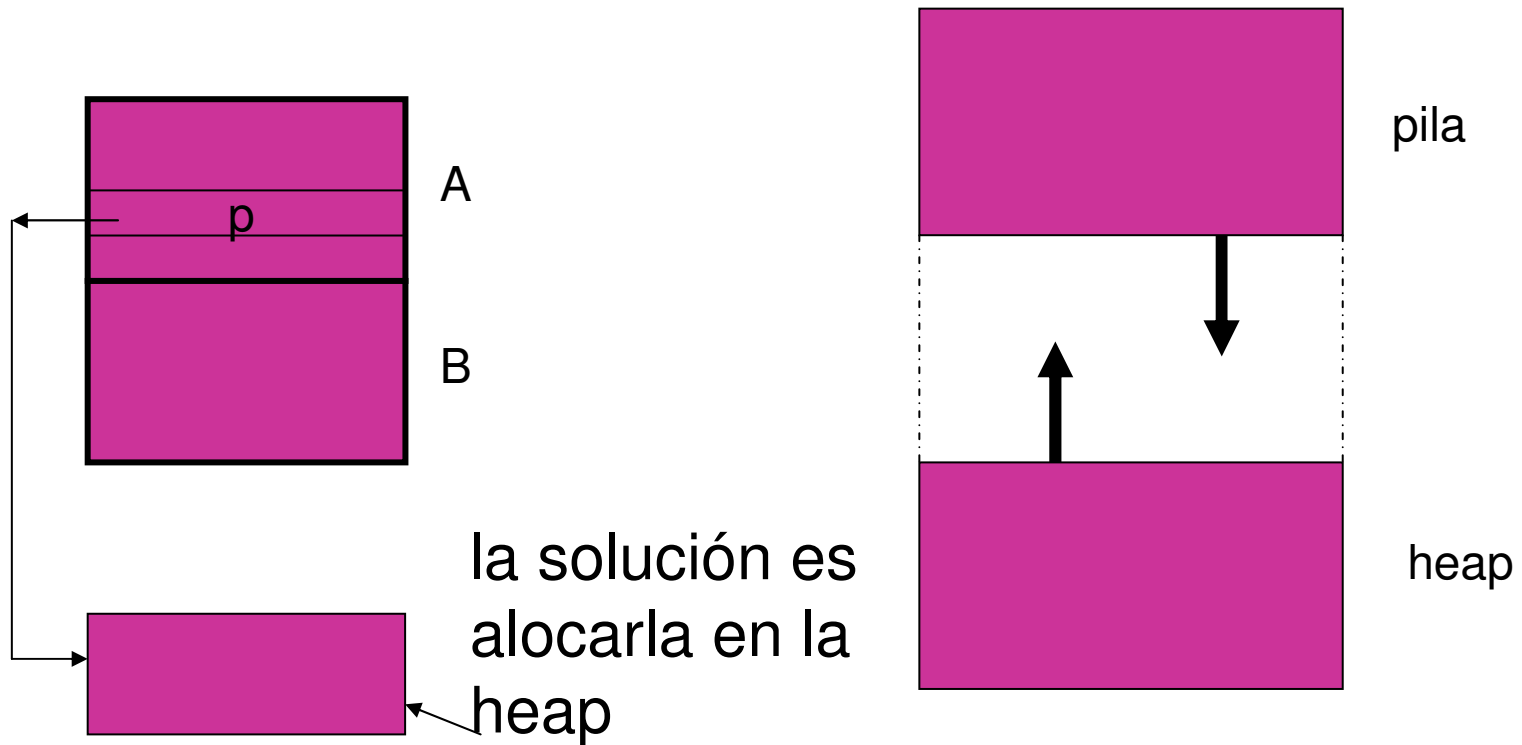
# C5''

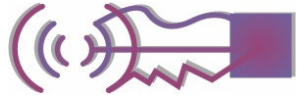
donde allocar el objeto  
apuntado?





el tiempo de vida no depende de la sentencia de  
alocación, vivirá mientras este apuntada

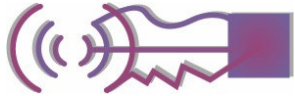




## C6: LENGUAJES dinámicos

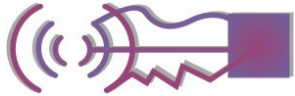
- se trata de aquellos lenguajes que adoptan más reglas dinámicas que estáticas.
- Una propiedad dinámica significa que las ligaduras correspondientes se llevan a cabo en ejecución y no en compilación.



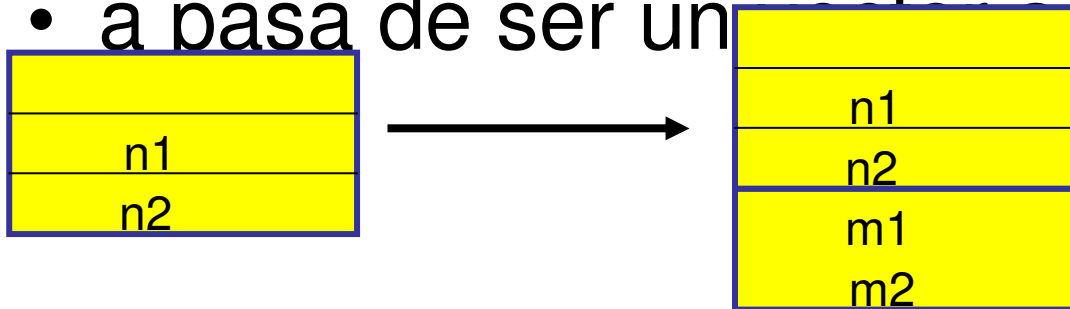


## C6: tipado dinámico

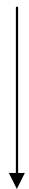
- el tipo de una variable **no** pueden determinarse en compilación.
- en el descriptor debe mantenerse
- el descriptor debe almacenarse en la heap
- para cada variable se reserva en el registro de activación un puntero



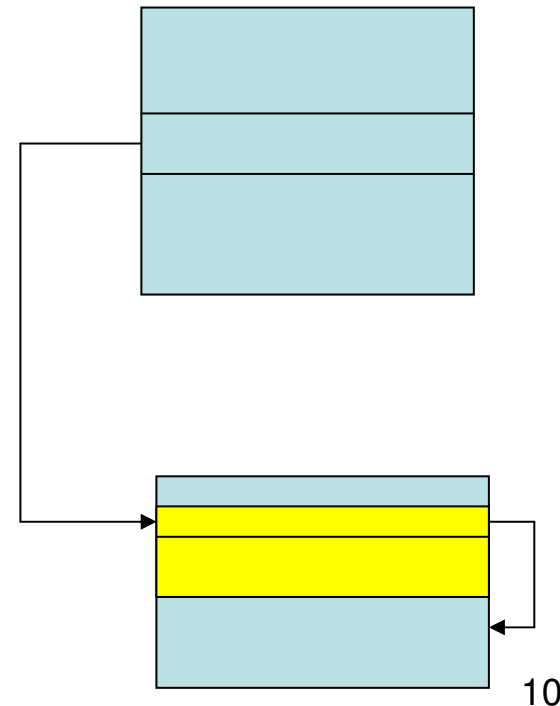
- a pasa de ser un ~~vector~~ a ser una matriz

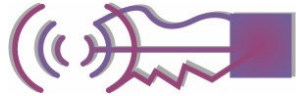


$V = (1,2,3)$  vector



$V = (3,4,5) (6,7,8)$  matriz

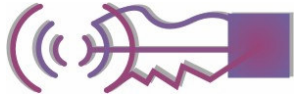




## C6: alcance dinámico

Un programa

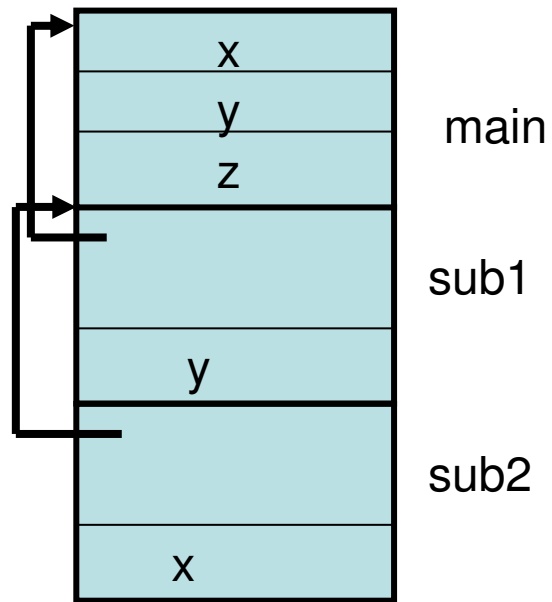
- n rutinas
  - variables locales  
sin definición de tipo
- un main
- regla de alcance: depende de la cadena dinámica



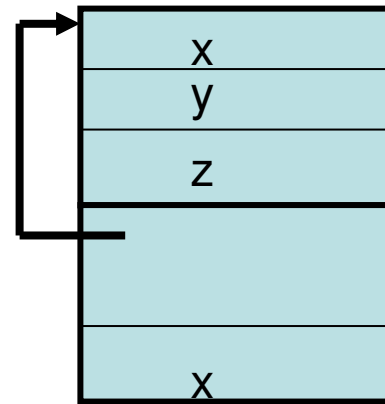
## C6: alcance dinámico

```
sub2() {  
    dec x  
    ...x....  
    ...y...  
}  
sub1() {  
    dec y  
    ...x...  
    ...z..  
    sub2() ↗  
}
```

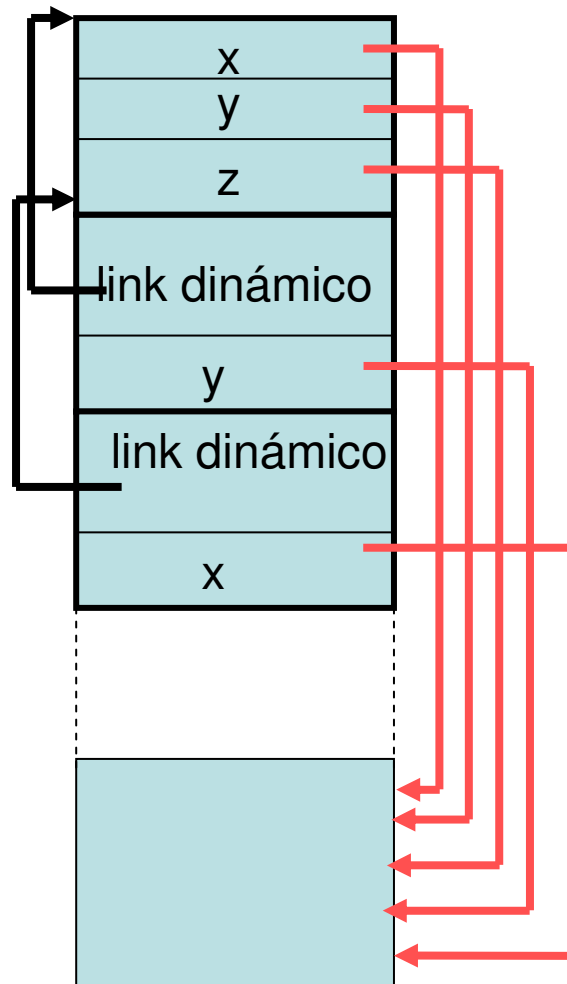
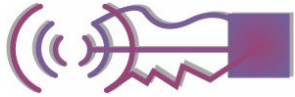
```
main(){  
    dec x,y,z  
    z = 0  
    x= 5  
    y= 7  
    sub1 ↖  
    sub2 ↖  
}
```



**main sub1 sub2**



**main sub2**



main sub1 sub2