

ESTRUCTURAS DE DATOS

TIPOS ABSTRACTOS DE DATOS LINEALES

# Aplicaciones de pilas

Manuel Montenegro Montes  
Departamento de Sistemas Informáticos y Computación  
Facultad de Informática – Universidad Complutense de Madrid

# Expresiones en forma postfija



# Expresiones en forma postfija

- Expresión en forma infija:

$$(3 * (5 + 2)) - 6$$

- La misma expresión en forma postfija:

$$(3 (5 2 +) *) 6 -$$

$$3 5 2 + * 6 -$$

- **Objetivo:** evaluar expresión en forma postfija.
  - Por ejemplo,  $3 5 2 + * 6 -$  se evalúa al valor 15.

# Expresiones en forma postfija

- Las expresiones en forma postfija no contienen ambigüedades, no necesitan paréntesis o reglas de precedencia entre operadores, al contrario que las expresiones en forma infija.

$$2 + 4 * 5$$



# Otros ejemplos

- 2 3 + 6 + 1 +
- 3 1 - 6 5 \* +



# Evaluando las expresiones mediante una pila

- Comenzamos con una pila vacía.
- Recorremos de izquierda a derecha los caracteres de la expresión.
- Si el carácter actual es un número:
  - Insertar el número en la pila.
- Si el carácter actual es un operador:
  - Desapilar los dos operandos y realizar la operación con ellos.
  - Apilar el resultado.
- Al finalizar el recorrido, el elemento que quede en la pila es el resultado de evaluar la expresión.

# Ejemplo de ejecución

↑ 3 5 2 + \* 6 -



# Ejemplo de ejecución

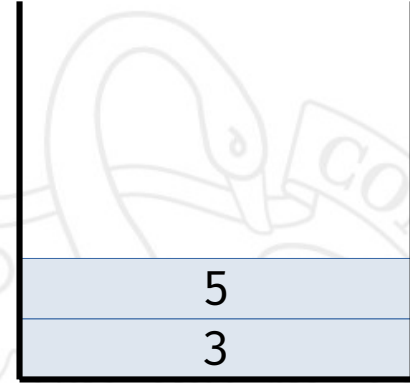

3 5 2 + \* 6 -  
↑





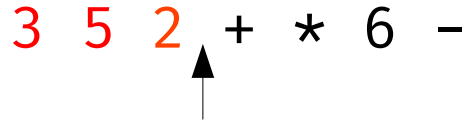
# Ejemplo de ejecución

3 5 2 + \* 6 -



# Ejemplo de ejecución

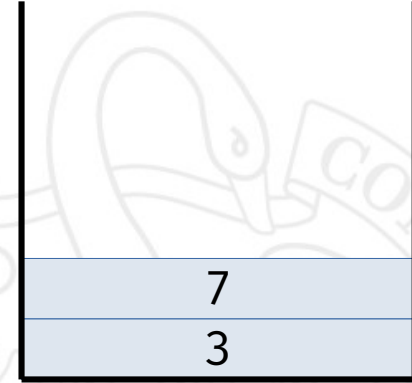

3 5 2 + \* 6 -



2
5
3

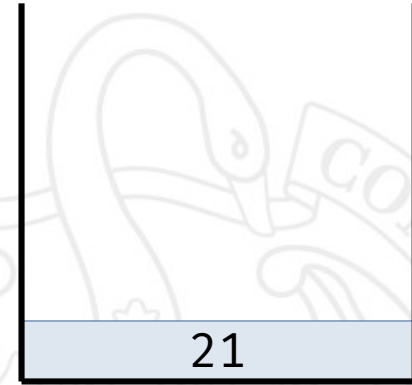

# Ejemplo de ejecución

3 5 2 + \* 6 -



# Ejemplo de ejecución

3 5 2 + \* 6 -



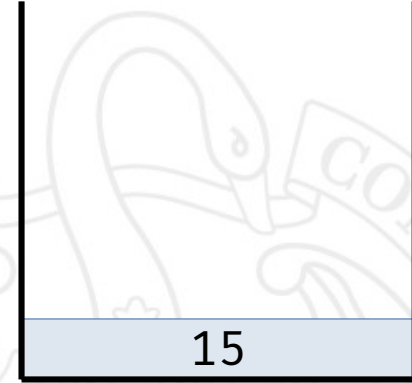
# Ejemplo de ejecución

3 5 2 + \* 6 -  
↑

6
21

# Ejemplo de ejecución

3 5 2 + \* 6 -

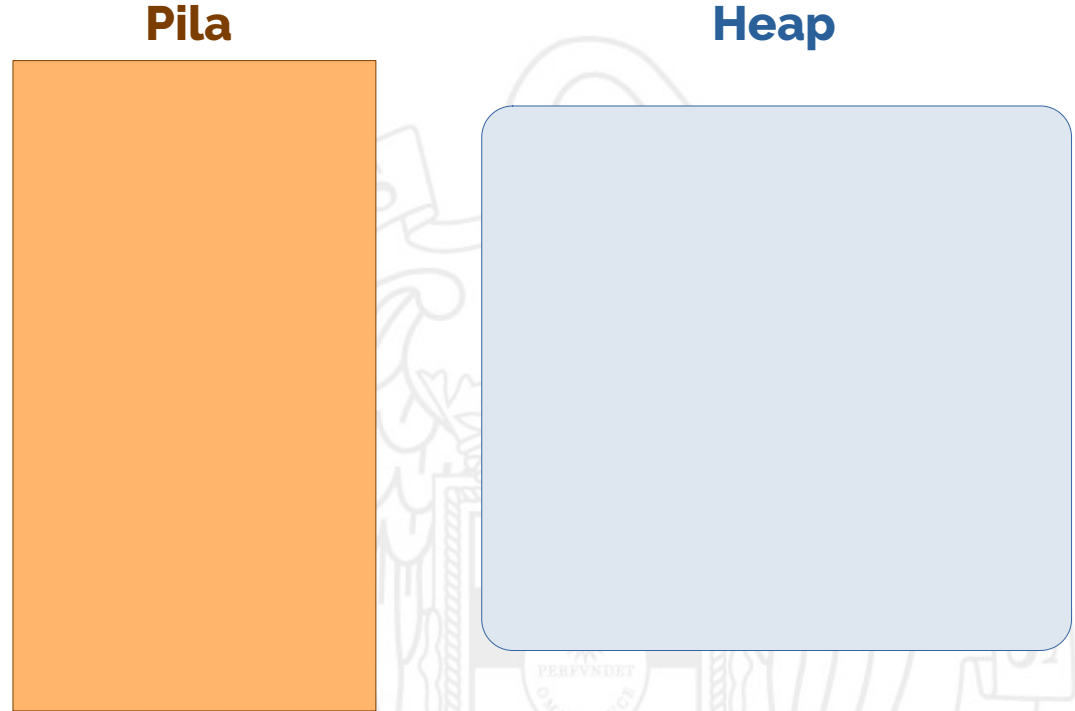


# Pila de llamadas a funciones



# Ejemplo de ejecución

- La región de memoria en la que se almacenan las variables locales y parámetros de un programa funciona como una pila.





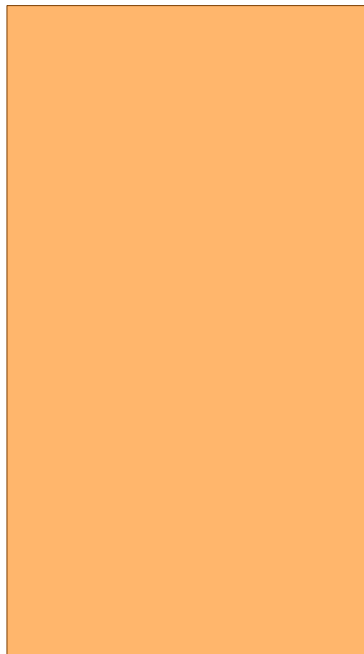
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

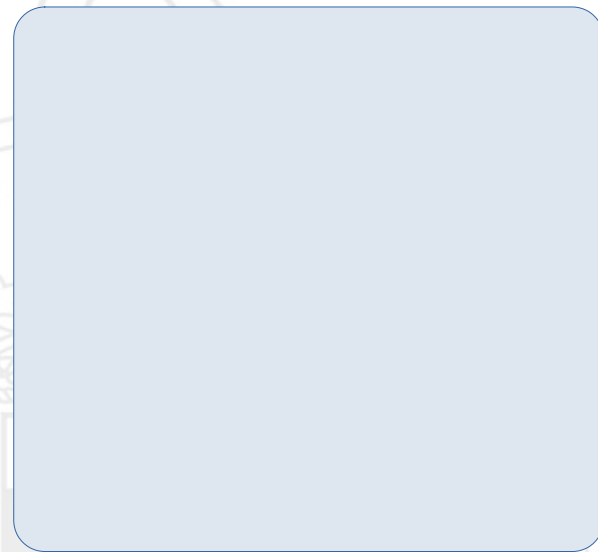
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**



**Heap**



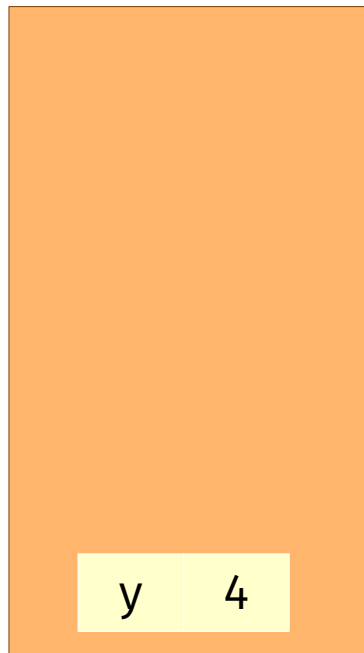
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

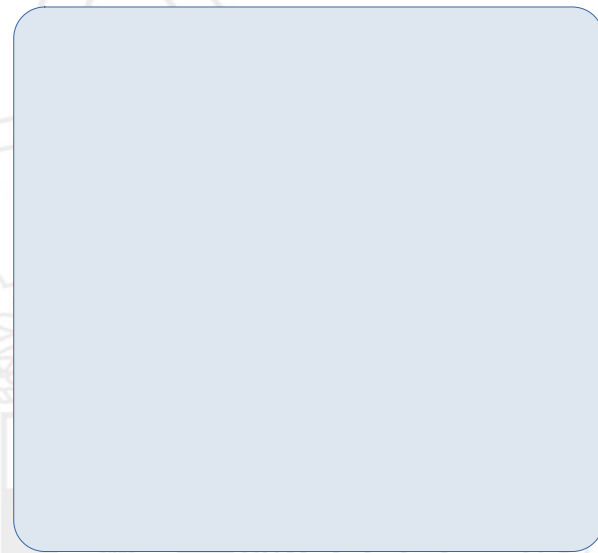
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**



**Heap**



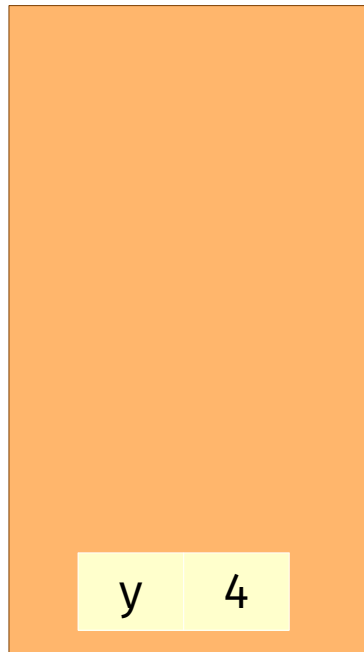
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

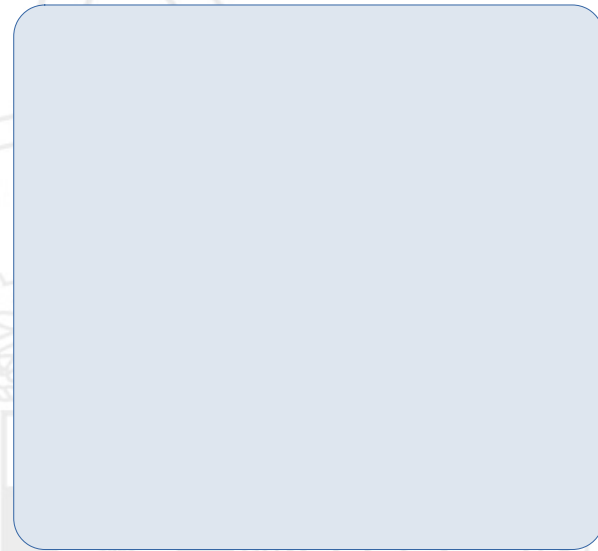
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**



**Heap**



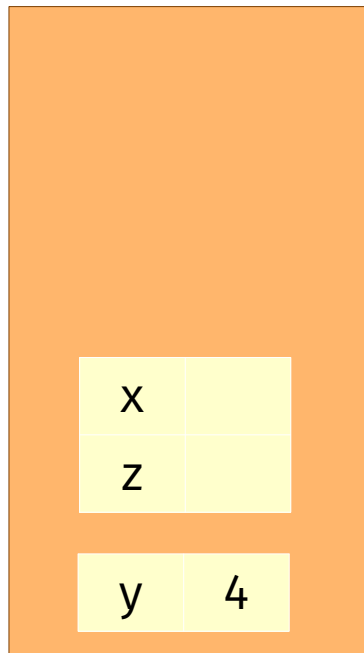
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

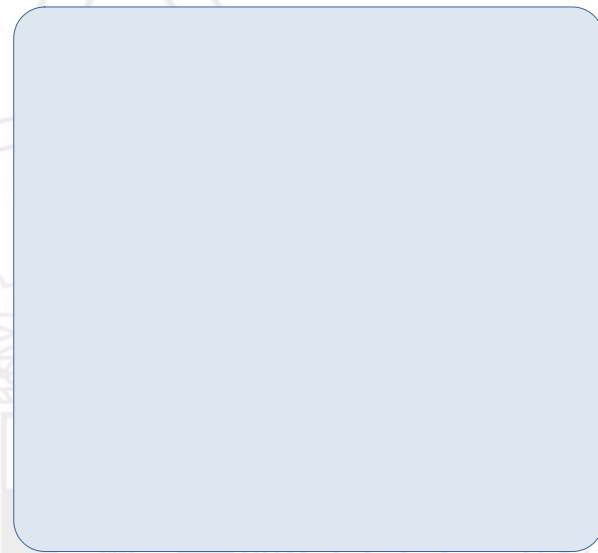
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**



**Heap**



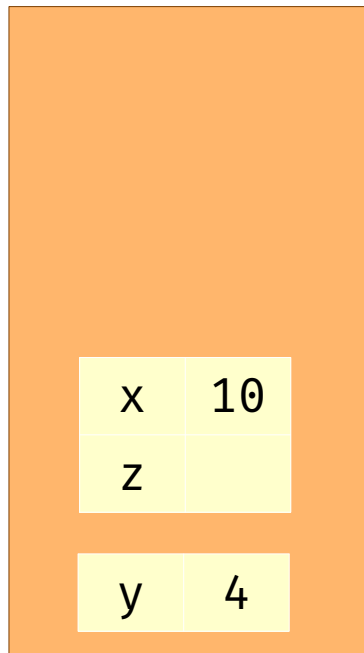
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

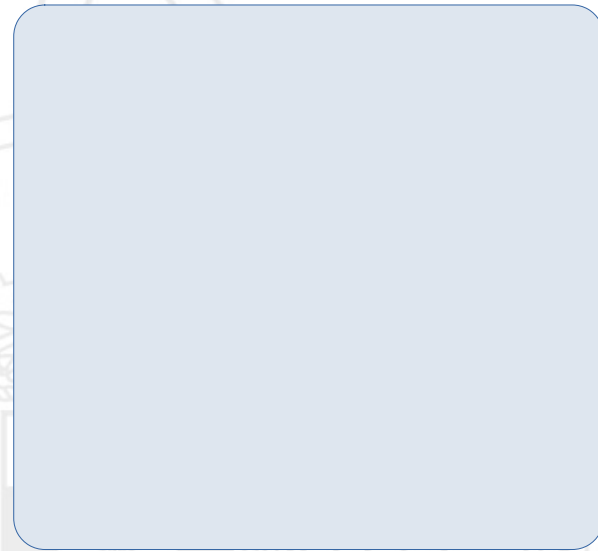
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**



**Heap**



# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

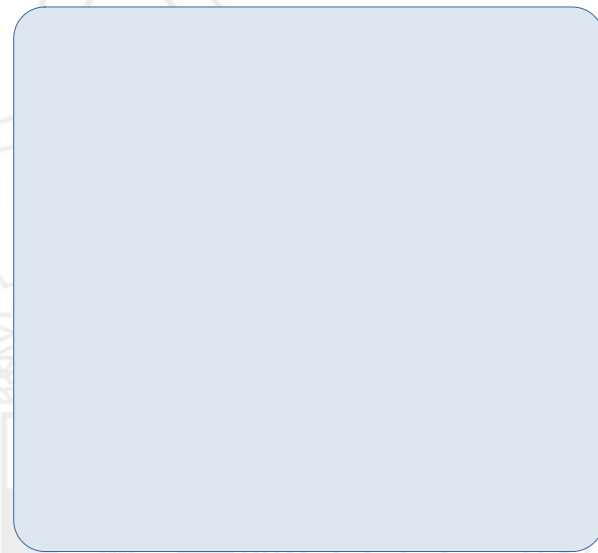
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

Pila

x	10
z	20
y	4

Heap



# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

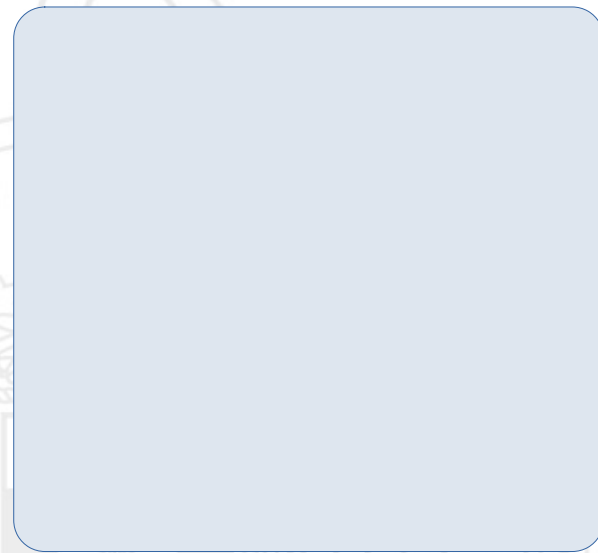
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**

x	10
z	20
y	4

**Heap**



# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

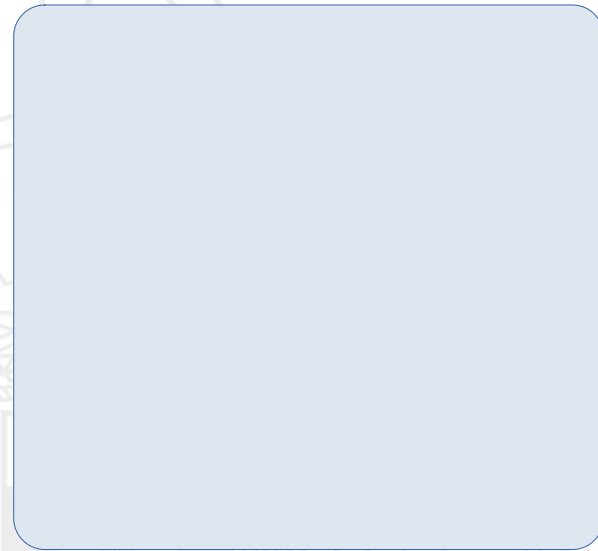
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

Pila

x	30
y	
x	10
z	20
y	4

Heap





# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

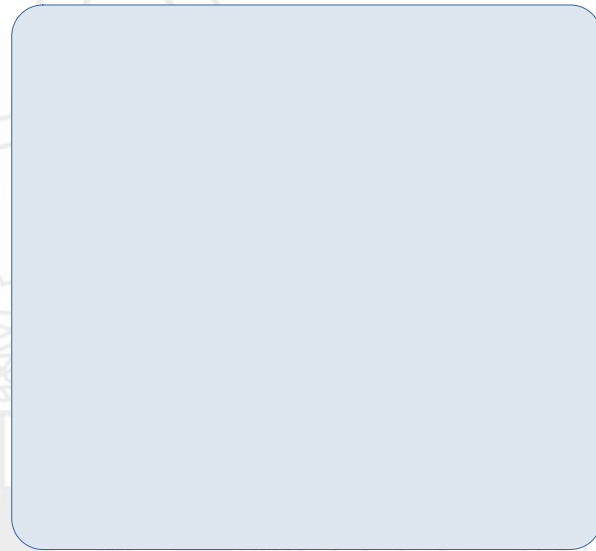
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

Pila

x	30
y	40
x	10
z	20
y	4

Heap



# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

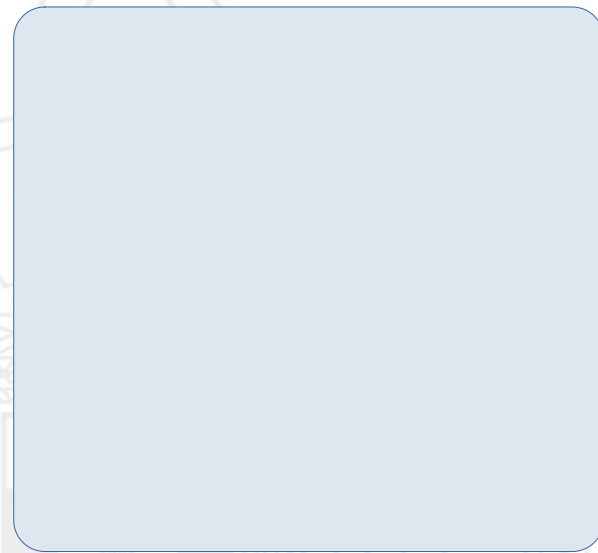
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

Pila

x	30
y	40
x	10
z	20
y	4

Heap



# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

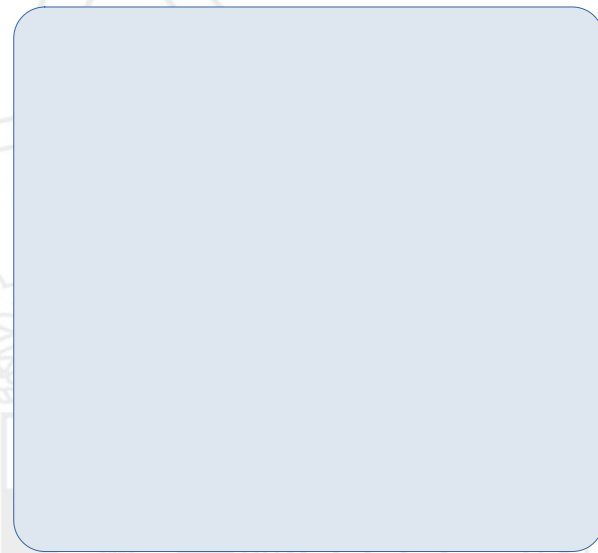
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**

x	50
z	20
y	4

**Heap**



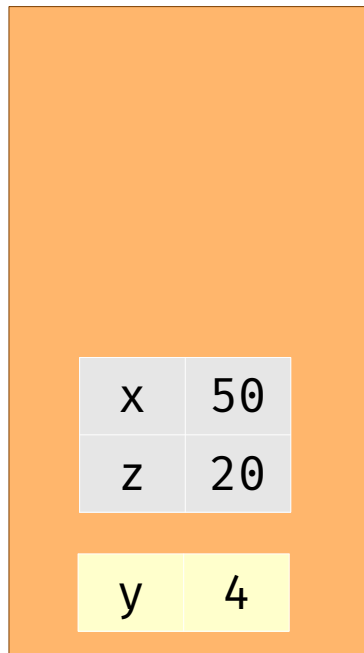
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

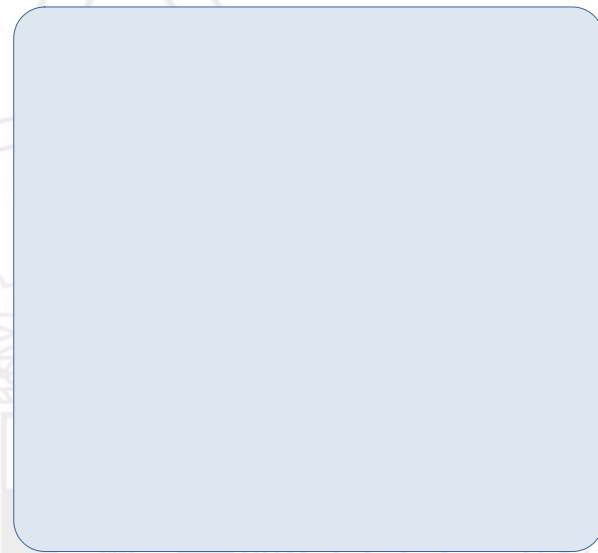
```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

**Pila**



**Heap**



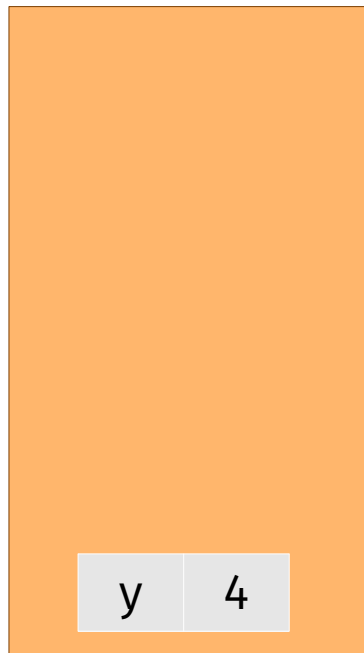
# Ejemplo de ejecución

```
void f(int x) {  
    int y = 40;  
}
```

```
void g() {  
    int x = 10;  
    int z = 20;  
    f(30);  
    x = 50;  
}
```

```
int main() {  
    int y = 4;  
    g();  
}
```

Pila



Heap

