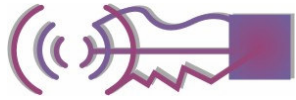


SEMANTICA OPERACIONAL

UNIDADES DE PROGRAMA



UNIDADES

UNIDAD



acción abstracta

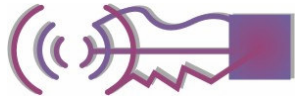
- En general se las llama **rutinas**

PROCEDIMIENTOS

FUNCIONES

→ un valor

- mecanismos que controlan el flujo de ejecución entre rutinas con todas las ligaduras involucradas.

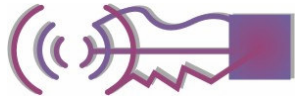


<nombre, alcance, tipo, l-value, r-value>

nombre: string de caracteres que se usa para invocar a la rutina.

alcance: es el rango de instrucciones donde se conoce su nombre.

- **Activación:** dentro del alcance de la rutina
- **Ambiente de referencia**
 - local
 - no local
 - global



EJEMPLO

/ sum es una funcion que suma los n primeros naturales,
1 + 2 + ... + n; suponemos que el parametro n es positivo */*

int sum(int n)

{

nombre

int i, s;

declaraciones locales

s = 0;

for (i = 1; i <= n; ++i)

s += i;

return s;

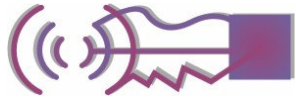
}

sum

i

s

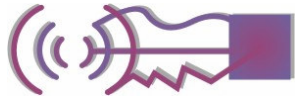
end del file en el que esta la
declaración



tipo

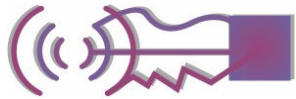
tipo: tipo de los parámetros

- **Signatura:** permite especificar el tipo de una rutina
Una rutina *fun* que tiene como entrada parámetros e tipo T_1, T_2, T_n y devuelve un valor de tipo R , puede especificarse con la siguiente signatura
$$fun: T_1 \times T_2 \times \dots \times T_n \rightarrow R$$
- un llamado a una rutina es correcto si esta de acuerdo al tipo de la rutina.
- la conformidad requiere la correspondencia de tipos entre parámetros formales y reales.



l-valor / r-valor

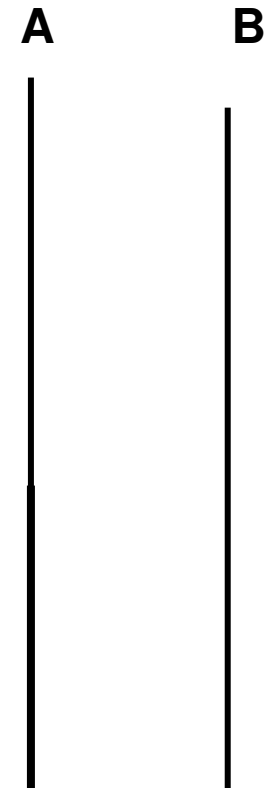
- ***l-value***: es el lugar en el que se almacena el cuerpo de la rutina.
- ***r-value***: es el código de la rutina que se ejecuta en la invocación
 - **estático**
 - **dinámica**

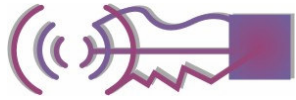


Definición-Declaración

Si el lenguaje distingue entre la declaración y la definición de una rutina permite manejar esquemas de **rutinas mutuamente recursivas**.

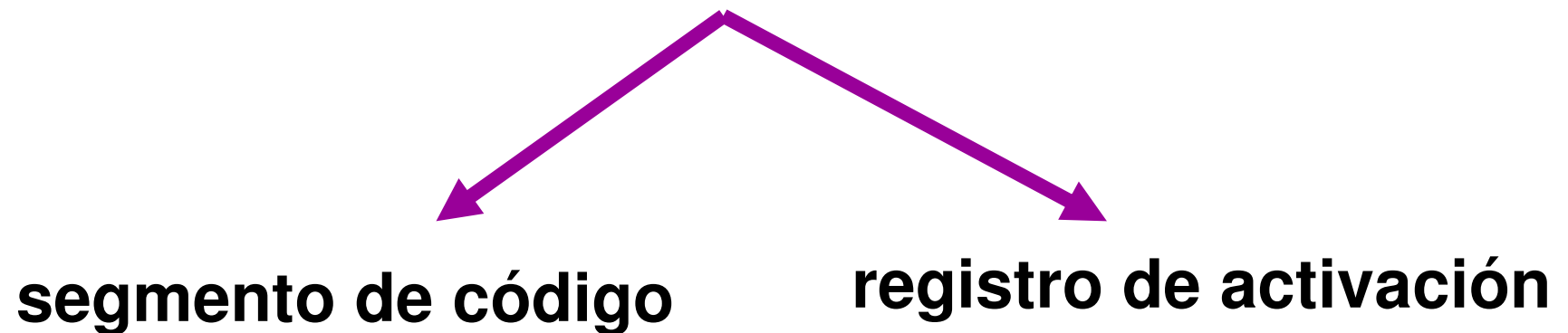
```
int A(int x, int y);  
float B(int z)  
{  
    int w, u;  
    ...  
    w = A(z, u);  
};  
int A(int x, int y)  
{  
    float t;.....  
    t = B(x);.....  
}
```

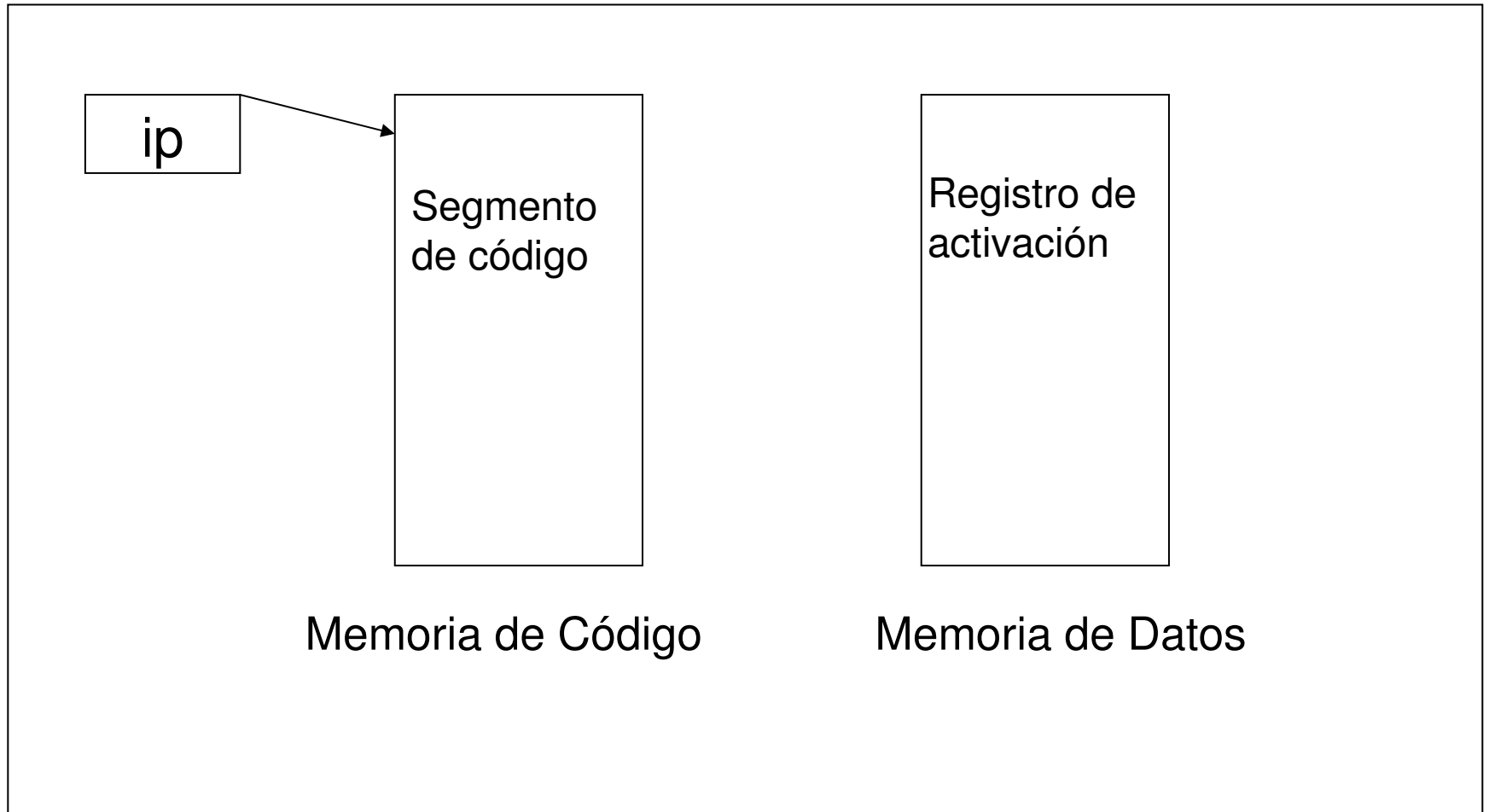
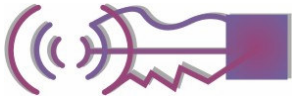


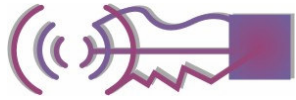


REPRESENTACION EN EJECUCION

- .
- **instancia de la unidad:** es la representación de la rutina en ejecución.

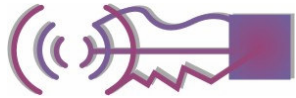






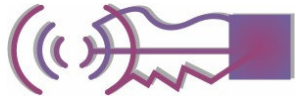
Elementos en ejecución

- **punto de retorno**
- **ambiente de referencia**
 - ambiente local
 - ambiente no local
 - Alias
 - Efecto lateral



Rutinas recursivas

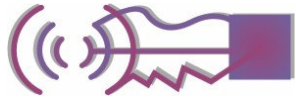
- **Segmento de código:**
el mismo para todas las instancias de la unidad
- **Registro de activación:**
uno nuevo para cada activación



Comunicación entre rutinas

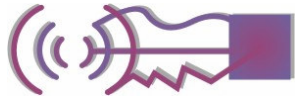
- **Ambiente no local**
- **Parámetros**





Rutinas genéricas

- las rutinas factorean un segmento de código, que se ejecutará customizada a través de los parámetros.
- **rutina genérica:** se puede parametrizar el tipo (C++, Ada)
- Una rutina genérica es un molde.



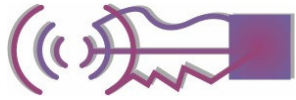
Alias y sobrecarga

1 nombre \longrightarrow 1 entidad

No hay ambigüedad

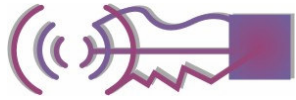
Sobrecarga: un nombre esta sobrecargado si:

- en un momento, referencia mas de una entidad y
- hay suficiente información para permitir establecer la ligadura univocamente.



Alias y sobrecarga

Alias: dos nombres que denotan la misma entidad en el mismo ambiente.



PROCESADOR ABSTRACTO

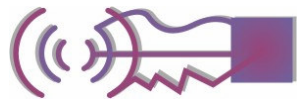
Memoria de Código: $C(y)$ valor almacenado en la y ésima celda de la memoria de código. Comienza en cero

Memoria de Datos: $D(y)$ valor almacenado en la y ésima celda de la memoria de datos. Comienza en cero

Ip: puntero a la instrucción que se esta ejecutando.

Ejecución:

- obtener la instrucción actual para ser ejecutada ($C[ip]$)
- incrementar ip
- ejecutar la instrucción actual



PROCESADOR ABSTRACTO- Instrucciones

SET: setea valores en la memoria de datos

set target,source

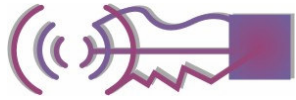
copia el valor representado por source en la
dirección representada por target

E/S: read y write

set 15,read

set write,D[50]

*set 99, $D[15]+D[33]*D[4]$*



PROCESADOR ABSTRACTO- Instrucciones

JUMP: bifurcación incondicional

JUMPT: bifurcación condicional, bifurca si la
expresión se evalúa como verdadera

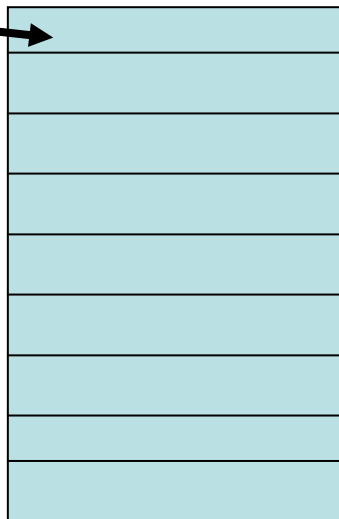
direccionamiento indirecto:

set D[10], D[20]

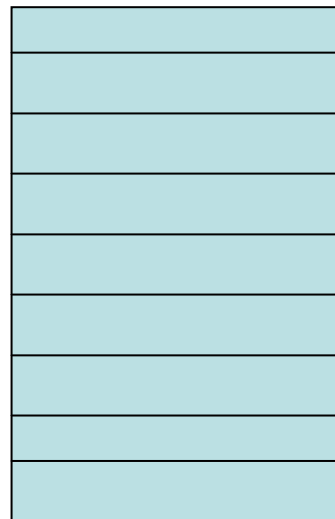


procesador

ip



C



D