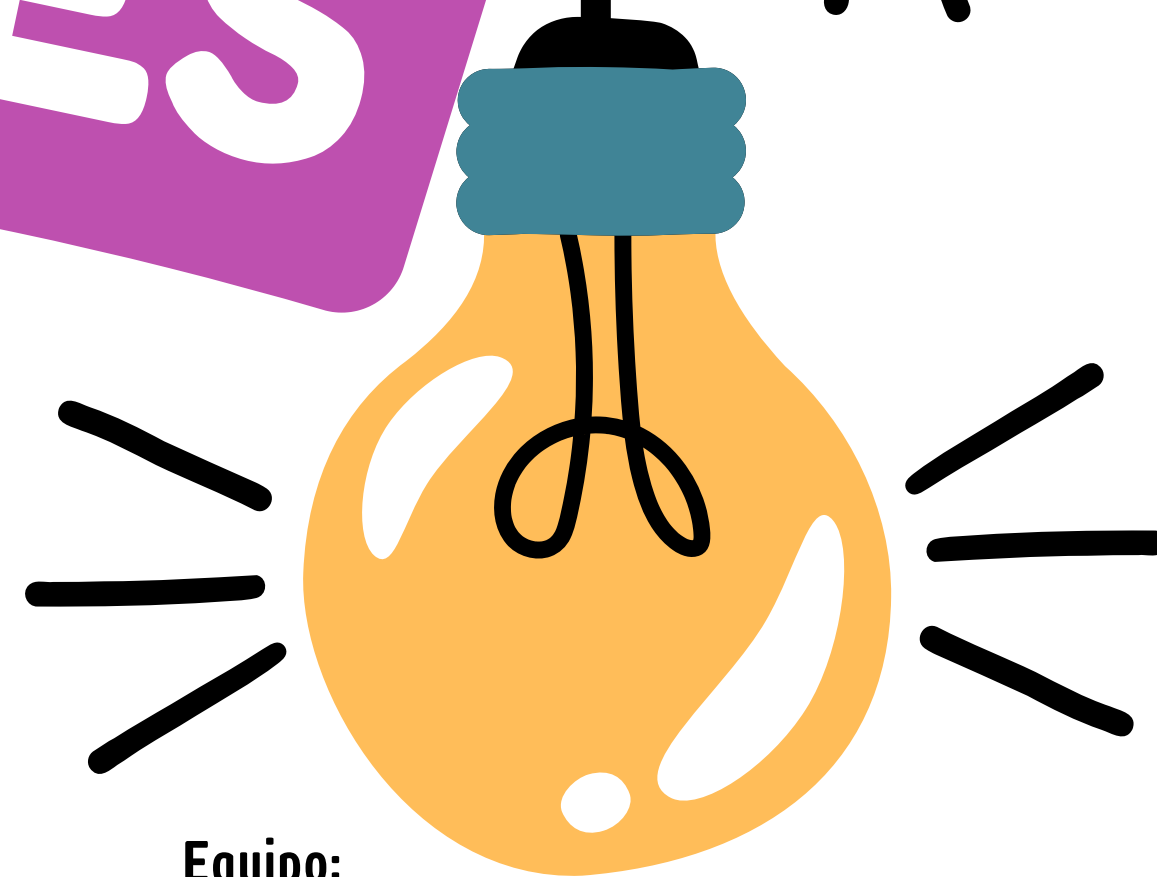




# APLICACIONES

de las

# PILAS



Equipo:

Bandala Hernández Sebastián

Brito García Sebastián

Jiménez Jiménez Carlos Yael



## LLAMADAS A SUBPROGRAMAS

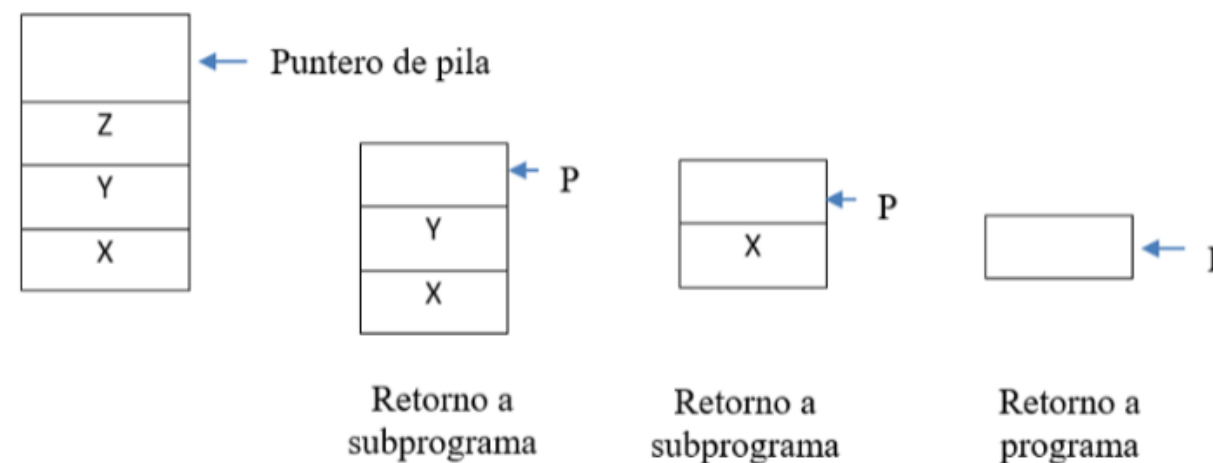


Cuando dentro de un programa se realizan llamadas a subprogramas, el programa principal debe recordar el lugar donde se hizo la llamada, de modo que pueda retornar allí cuando el subprograma se haya terminado de ejecutar.

Esta operación se consigue disponiendo las direcciones de retorno de una pila.

Cuando un subprograma termina, debe retornarse a la dirección siguiente a la instrucción que le llamo (llamar\_a). Cada vez se invoca un subprograma, la dirección siguiente (X, Y o Z) se introduce en la pila.

El vaciado de la pila se realizara por los sucesivos retornos, decrementándose el puntero de pila que queda libre apuntando a la siguiente dirección de retorno.



## EJEMPLO

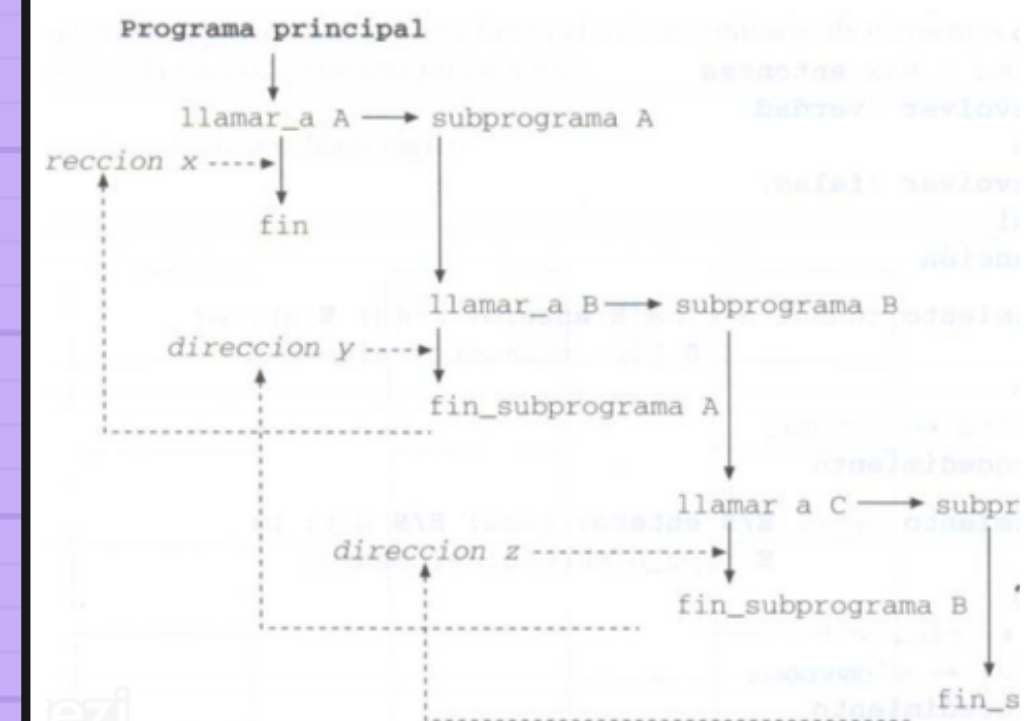


Tenemos 3 subprogramas llamados A, B y C.

A invoca a B y B invoca a C.

B no terminara su trabajo hasta que C haya terminado y devuelva su control a B.

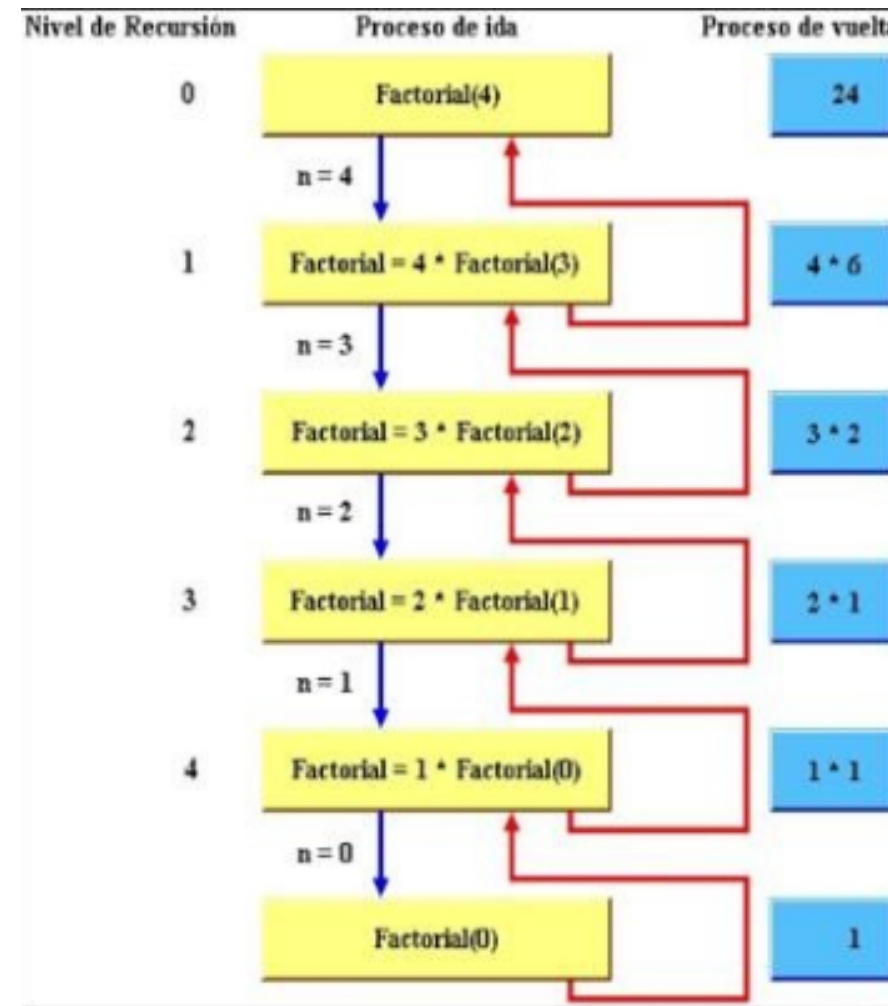
A es el primero que arranca su ejecución, pero es el último que lo termina, tras la terminación y retorno de B



## PASO DE PROGRAMAS RECURSIVOS A ITERATIVOS



Para simular un programa recursivo es necesario la utilización de pilas, ya que se está llamando continuamente a subprogramas que a la vez vuelven a llamarse a si mismo. Veamos cómo se utiliza esta eliminación de la recursión mediante un ejemplo sencillo hecho en clase. El cálculo del número factorial.



# TRATAMIENTOS DE EXPRESIONES ARITMÉTICAS



Una expresión aritmética está formada por operandos y operadores, como Por ejemplo  $(a + b) - c * d$ .

En este caso  $+$ ,  $-$ ,  $*$  son los operadores y  $a$ ,  $b$ ,  $c$ ,  $d$  los operandos.

Esta forma de escribir una expresión, se denomina notación usual o infija.

Existen otras formas de escribir expresiones aritméticas, en el que se diferencian por la situación de los operadores respecto de los operandos. La forma en la que los operadores se colocan delante de los operandos es conocida como notación polaca o notación prefija.

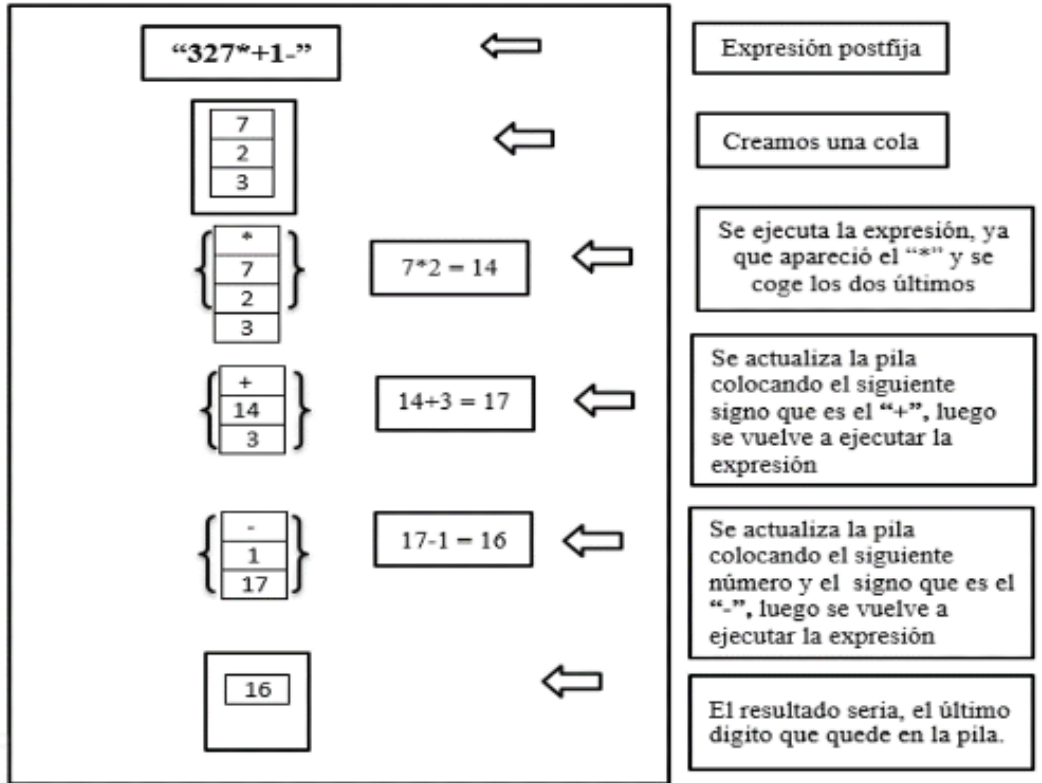
Ejemplo:

Notación infija:  $a*b/(a+c)$

Notación polaca:  $*/ab+ac$

Notación infija	Notación postfija
$\text{sen}(x)$	$x \text{ sen}$
$1 + 3 * 4$	$1 \ 3 \ 4 \ * \ +$
$\text{sen}(9 + x)$	$9 \ x \ + \ \text{sen}$

Notación infija	Notación postfija
$3 * 6$	$3 \ 6 \ *$
$4 - \text{sen}(x)$	$4 \ x \ \text{sen} \ -$
$\text{sen}(x^2) + 4 * x$	$x \ 2 \ ^ \ \text{sen} \ 4 \ x \ * \ +$



# ERRORES



Ingresar algún carácter que no se puede interpretar ni como número ni como una de las cinco operaciones válidas.

Ejemplo:

Si la cadena de entrada es "5 3 &"

No hay suficientes operandos para realizar la operación.

Ejemplo:

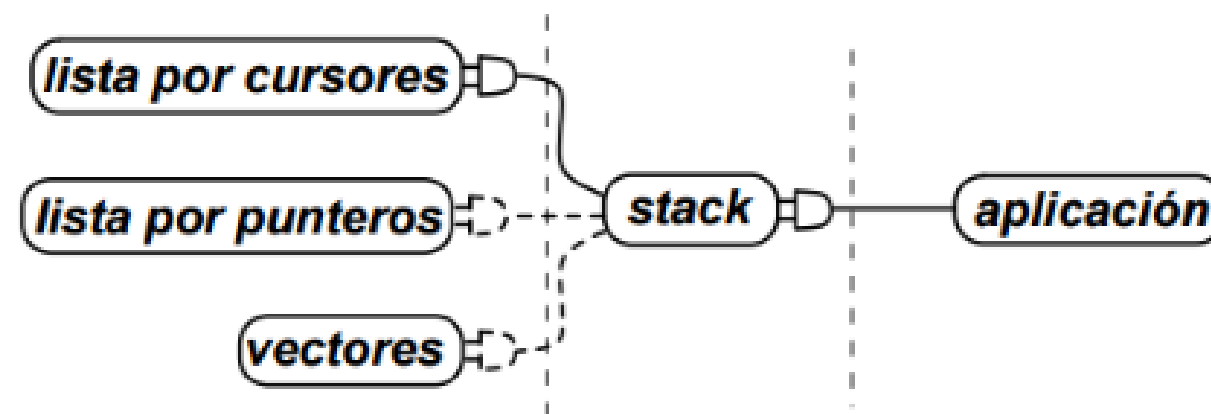
Si la cadena de entrada es "5 3 - +"

## LA PILA COMO UN ADAPTADOR



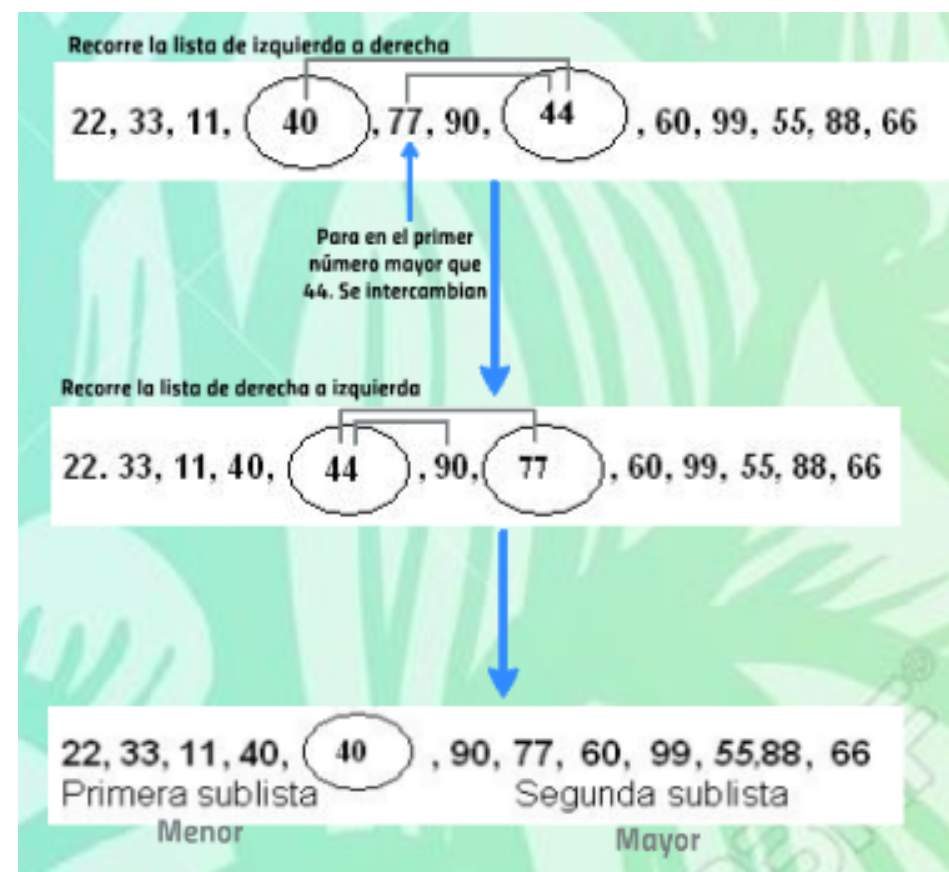
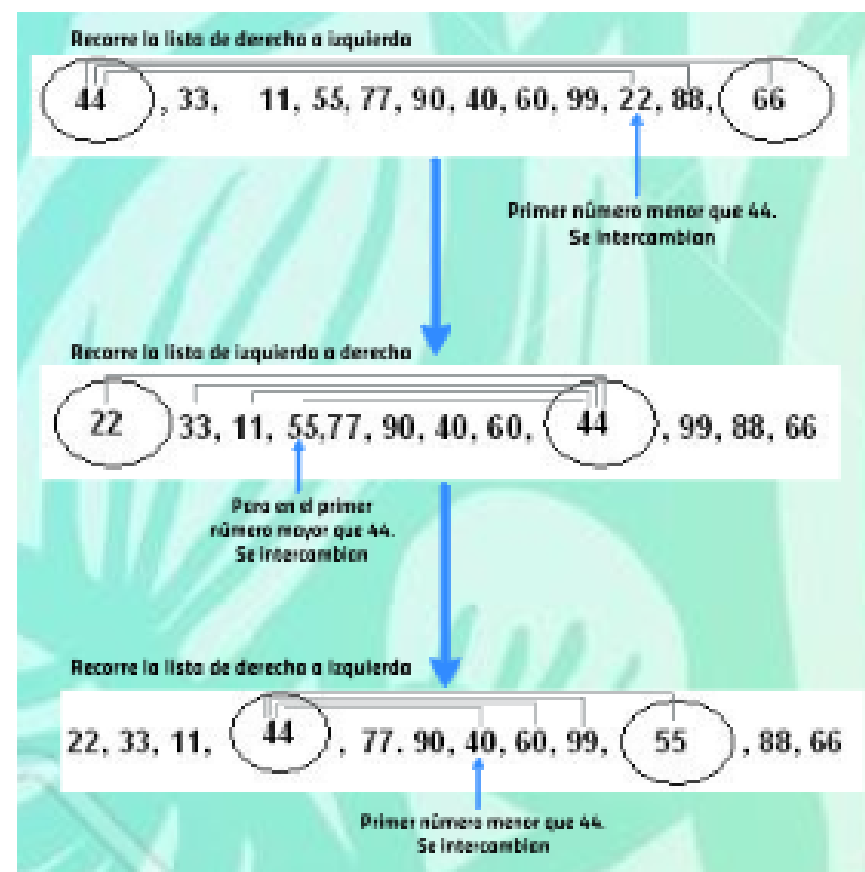
**Notar que la pila deriva directamente de la lista, pero con una declaración `private`. De esta forma el usuario de la clase `stack` no puede usar métodos de la clase `lista`.**

**Como la pila en si misma no contiene iteradores no hay necesidad de clases anidadas ni sobrecarga de operadores, de manera que la única diferencia con la interfaz STL es el uso de templates. Una interfaz compatible con STL**



## ORDENACIÓN RÁPIDA

El «ordenador A» se refiere a la operación de reorganizar los elementos de A.  
El siguiente algoritmo de ordenación llamado ordenación rápida es para ilustrar lo que sería una aplicación de las pilas



Como sólo podemos procesar una sublista cada vez, tenemos que poder seguir la pista de algunas sublistas para procesarlas posteriormente.

Esto se hace usando dos pilas, llamadas MENOR y MAYOR, para <<mantener>> temporalmente tales sublistas.

Las direcciones de los elementos primero y último de cada sublista, llamados valores límite, se meten en las pilas MENOR y MAYOR respectivamente; y el paso de reducción se aplica a una sublista sólo tras sacar de las pilas sus valores límite.

**MUCHAS  
GRACIAS**

