



# Estructura de Datos y Algoritmos

## Pilas

*Semana 5*



- **Definición de Pila (TDA Pila)**
  - Ejemplos
  - Operaciones
- **Implementación en Java**
- **Aplicaciones de las pilas**
- **Notaciones aritméticas**
  - Expresiones postfijas

# Pilas - Definición



- Es una lista de elementos en la cual se insertan (push) y eliminan datos (pop) por un mismo extremo de la lista.
- Las escrituras de datos son inserciones de elementos, y las lecturas eliminan el elemento leído.
- El comportamiento de lista es tipo LIFO (del inglés Last In First Out), el último en entrar es el primero en salir.

# Pilas - Ejemplos



- Pila de platos
- Pila de procesos
- Evaluación de expresiones aritméticas



Se implementan usando:

## Arreglos

- Variables estáticas
- Tamaño es fijo
- Peligro de desbordamiento (*overflow*)
- Uso ineficiente de memoria

## Listas enlazadas

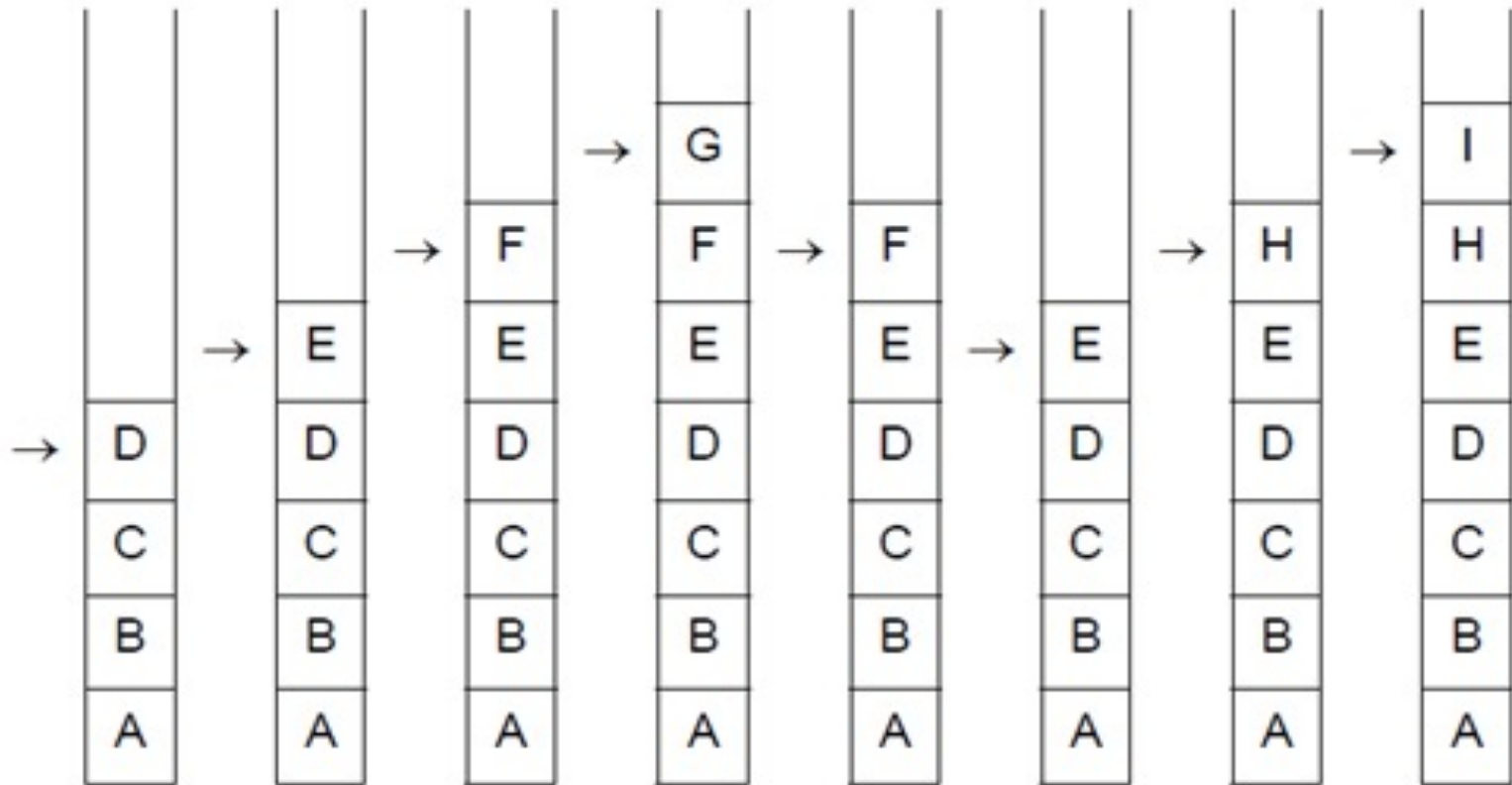
- Variables dinámicas
- No hay tamaño fijo
- No hay peligro de *overflow*
- Uso eficiente de memoria

# Pilas - Operaciones



- Crear la pila.
- Está vacía.- Determinar si la pila está vacía.
- Agregar o apilar (push).- Agrega o inserta un elemento en el tope de la pila.
- Eliminar o desapilar (pop).- Elimina el elemento que se encuentra en el tope de la pila.
- Acceso.- Examinar el elemento que se encuentra en el tope de la pila.

# Pilas – Operaciones push y pop



push(D) push(E) push(F) push(G) pop pop push(H) push(I)

# Pilas – Implementación en Java



La pila en Java se implementa con la clase Stack, la cual hereda de java.util. Para importar dicha clase usaremos:

```
import java.util.Stack
```

- **Creación:**

- Stack pila = new Stack()

- Stack<Integer> pila = new Stack<>();



# Pilas – Implementación en Java



- **Métodos:**
  - `push(E item)` : Agrega un elemento a la pila. Devuelve el elemento insertado.
  - `pop()` : Elimina el último elemento agregado a la pila. Devuelve el elemento eliminado.
  - `peek()` : Permite ver el ultimo elemento agregado. Devuelve el elemento del tope de la pila.

# Pilas – Implementación en Java



- **Métodos:**
  - `empty()`: Permite saber si la pila esta vacía. Devuelve un valor booleano.
  - `search(Object o)`: Devuelve la posición (int) del objeto a partir del tope de la pila, empieza en 1.
  - `size()`: Devuelve el numero de elementos (int) de la pila.

# Pilas – Implementación en Java



```
Stack pila = new Stack();  
pila.push("David");  
pila.push("Jorge");  
pila.push("Miguel");  
System.out.println(pila.size());  
// resultado: 3  
System.out.println(pila.pop());  
// resultado: Miguel  
System.out.println(pila.size());  
// resultado: 2  
System.out.println(pila.peek());  
// resultado: Jorge  
System.out.println(pila.size());  
// resultado: 2
```

# Pilas – Aplicaciones



- **Llamadas a sub programas.**
- **Procesos recursivos.**
- Análisis de Expresiones (Equilibrio de símbolos)
- **Tratamiento de expresiones aritméticas**
  - **Conversión de expresiones aritméticas.**
  - **Evaluación de expresiones postfijas**
- Borrado de caracteres en un editor de textos
- **Métodos de ordenación**

# Pilas – Notaciones aritméticas



- Para convertir una expresión infija a una postfija se toman en cuenta lo siguiente:

- Orden de los operadores

- ( )

- $\wedge$

- \* /

- + -



$$A \wedge B = A * A \dots$$

- Los operadores de mas alta prioridad se ejecutan primero
- Si hubiera dos operadores de igual prioridad, se ejecutan de izquierda a derecha.



# Pilas – Notaciones aritméticas



- Se desea convertir la expresión  $A + (B * C)$  a su representación en postfija.
- Aplicando las reglas de precedencia, primero se convierte la parte de la expresión que se evalúa primero:
- $A + (B * C)$
- $A + (BC *)$  se convierte la multiplicación.
- $A (BC *) +$  convierte la suma.
- $A B C * +$  representación postfija.



# Pilas – Notaciones aritméticas



Los operandos van a la expresión postfija y los operadores van a una pila.

Pasos a seguir:

- Se evalúa el símbolo de la expresión infija,
- Si es operando, se agrega a la expresión postfija.
- Si es operador, se evalúa la pila,
  - Si está vacía, se agrega el operador a la pila.
  - Si existe en la cima de la pila un operador de menor jerarquía, se agrega el operador evaluado a la pila
  - Si existe en la cima de la pila un operador de mayor jerarquía, se saca dicho operador de la pila y va a la expresión postfija. Se repite la operación de evaluación de la pila.



# Pilas – Notaciones aritméticas



Los operandos van a la expresión postfija y los operadores van a una pila.

Pasos a seguir:

- Se evalúa el símbolo de la expresión infija,
- Si es operando, se agrega a la expresión postfija.
- Si es operador, se evalúa la pila,
  - Si está vacía, se agrega el operador a la pila.
  - Si existe en la cima de la pila un operador de menor jerarquía, se agrega el operador evaluado a la pila
  - Si existe en la cima de la pila un operador de mayor jerarquía, se saca dicho operador de la pila y va a la expresión postfija. Se repite la operación de evaluación de la pila.





# Pilas – Notaciones aritméticas



Conversión de una expresión infija a postfija

Paso	Expresión	Simbolo a analizar	Pila	Expresión Postfija
0	$a + b * c$			
1	$+ b * c$	$a$		$a$
2	$b * c$	$+$	$+$	$a$
3	$* c$	$b$	$+$	$a b$
4	$c$	$*$	$+ *$	$a b$
5		$c$	$+ *$	$a b c$
6			$+$	$a b c *$
7				$a b c * +$



# Pilas – Notaciones aritméticas



- Realizar las siguientes conversiones:

1.-  $a + b^c * d / e^f$

2.-  $a * b^{(c + d * e)} / f + g * h - i$

3.-  $a^b^c * d + e / f - g + h$

4.-  $(a + b) * (c - d)$

5.-  $a^b * c - d + e / f / (g + h)$

6.-  $((a + b) * c - (d - e))^{(f + g)}$

7.-  $a - b / (c * d^e)$



# Pilas – Operaciones con postfijos



- Las operaciones en expresiones en notación postfija son mas fáciles de evaluar, porque no es necesario tener en cuenta las reglas de precedencia de operadores ni el paréntesis.
- No usan paréntesis.



# Pilas – Operaciones con postfijos



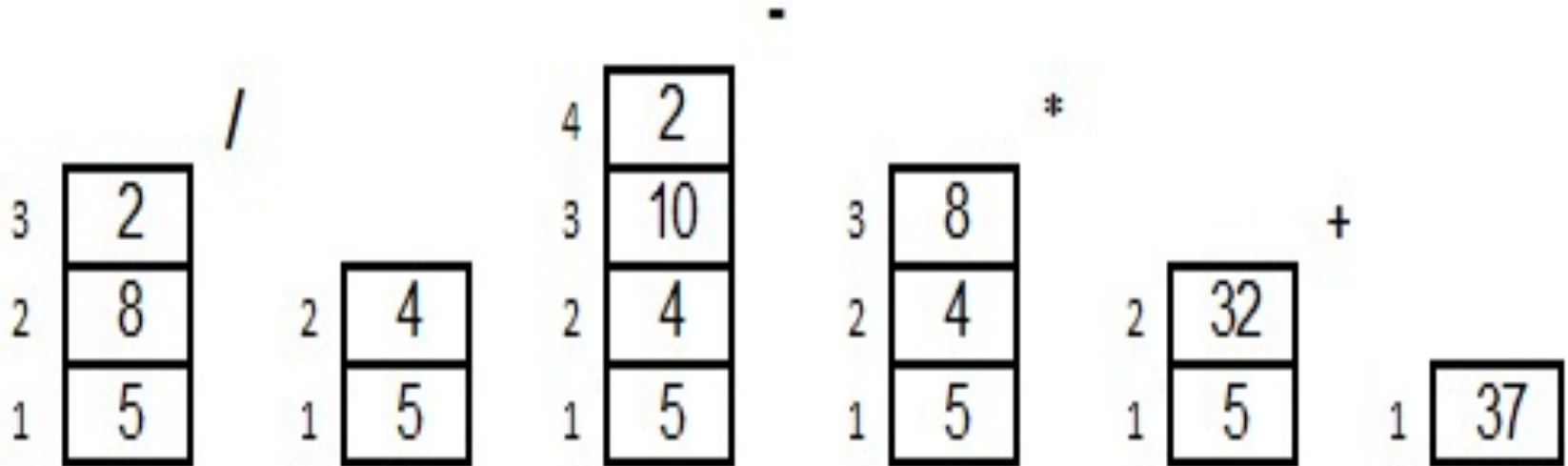
- Se analiza la expresión de izquierda a derecha.
- Si el elemento analizado es un operando, se inserta en la pila.
- Si es un operador, se extraen los dos elementos que se encuentran en la parte superior de la pila, se aplica la operación y el resultado se inserta en la pila.



# Pilas – Operaciones con postfijos



- Sea la siguiente operación de una expresión.
- $5 \ 8 \ 2 \ / \ 10 \ 2 \ - \ * \ +$



# Pilas – Operaciones con postfijos



- Ejercicios con notación POSFIJA (POLACA INVERSA)

1.- C A B + \*

2.- 4 6 3 / +

3.- 27 3 3 2 4 + + \* /

4.- X Y Z C \* + -

5.- 8 4 + 2 - 5 /

6.- 45 3 8 + 2 \* -

7.- 7 8 2 5 + \* +

8.- 6 3 8 \* 2 / +



# Pilas – Operaciones con postfijos



- Desarrollar un programa que permita implementar la conversión de una expresión infija en una expresión postfija.
- Ingresar N números a una pila, luego extraer dos números, sumarlos y guardar el resultado en la pila. Repetir este proceso hasta que solo quede un elemento en la pila
- En estos ejercicios debe experimentar con diferentes valores y analizar e interpretar el resultado obtenido para cada uno de los ejercicios.



# Conclusiones



- Esta presentación permite conocer el funcionamiento de una pila, los usos y la forma de trabajar con la notación post fija.







# Referencias bibliográficas

- CAIRO O. (2013). Estructuras de Datos. México: Mc Graw Hill.





**UPC**

Universidad Peruana  
de Ciencias Aplicadas