



Pilas y Colas

Laboratorio Estructuras de Datos

Viernes 08 de marzo de 2024

Pilas

Las pilas (Stacks en inglés) nos permiten guardar y sacar datos de forma LIFO (Last In First Out), es decir el último en entrar es el primero en salir, recuerda la frase popular "Los últimos serán los primeros".

Ejemplo: Imagina que tuvieras una pila enorme de libros, si intentaras sacar un libro de la parte de en medio o el libro de hasta abajo probablemente la pila perdería el equilibrio y todos tus libros se caerían, entonces ¿qué es lo más lógico si quieres obtener un elemento que esta a la mitad de la pila?, ir sacando uno a uno los libros desde el último (la posición más arriba) hasta llegar al que quieres.



Figura 1: Pila de libros: Imagina que pudieras hacer *pop()* el primer libro que saldría sería el que esta hasta arriba.

En java podemos crear una pila de la siguiente forma.

```
Stack<String>pila = new Stack<>() // Pila de cadenas  
Stack<Integer>pila = new Stack<>() // Pila de enteros
```

Colas

Las colas (Queues en inglés) nos permiten guardar y sacar elementos de forma FIFO (First In First Out), es decir el primer elemento que entra es el primero que sale por lo tanto es más "justo".

Ejemplo: Imagina que vas a la tortillería más cercana a tu casa. Si hay 5 personas antes de ti, tendrás que esperar 5 turnos antes de que te despachen y puedas volver a tu casa (digo, podrías intentar meterte, pero eso no saldría bien porque las personas que llegaron antes querrán que se respete su turno). La idea en las colas es la misma, los elementos saldrán en el mismo orden que llegaron.



Figura 2: Cola de personas esperando su turno para comprar tortillas, si pudieras hacer *poll()* la primera persona en salir (ser despachada) sería el chico de blusa azul y pantalón gris.

En java podemos crear una cola de la siguiente forma.

```
Queue<String>cola = new LinkedList<>() // Cola de cadenas
Queue<Integer>cola = new LinkedList<>() // Cola de enteros
```

Notemos que para crearla usamos *LinkedList* esto se debe a que en java *Queue* es una interfaz por lo que no podemos usarla directamente para crear la cola.

Métodos para manejar las estructuras

Métodos de la pila

- *push()*: Este método sirve para guardar algo en la pila.
- *pop()*: Este método nos permite sacar el último elemento agregado de la pila.
- *peek()*: Este método nos permite ver el último elemento agregado sin quitarlo de la pila.

Métodos de la cola

- `add()`: Este método nos permite añadir algo a la cola.
- `poll()`: Este método nos permite sacar el primer elemento de la cola.
- `peek()`: Este método nos permite visualizar el primer elemento a salir de la cola sin eliminarlo.

Capturas de EjemploPDF

Revisa el archivo EjemploPDF.java donde puedes ver el código completo.

Guardar en la estructura

```
for (int i = 0; i < 10; i++) {  
    // agregamos elementos a la pila y a la cola  
    pila.push(i);  
    cola.add(i);  
}  
imprimirPila();  
imprimirCola();
```

Figura 3: Fragmento del código EjemploPDF.java donde guardamos los números del 0 al 9 en ambas estructuras.

```
Pila -> 0 1 2 3 4 5 6 7 8 9  
Cola -> 0 1 2 3 4 5 6 7 8 9
```

Figura 4: Resultado al guardar e imprimir

Ver un elemento

```
System.out.println(x:"\nUso de peek() para ver el elemento que está en la cima de la pila y el  
frente de la cola");  
System.out.println(GREEN + "\nElemento en la cima de la pila: " + pila.peek() + RESET);  
System.out.println(YELLOW + "\nElemento en el frente de la cola: " + cola.peek() + RESET);  
System.out.println(x:"\nSi imprimimos la pila y la cola, vemos que no se eliminan los  
elementos");  
imprimirPila();  
imprimirCola();
```

Figura 5: Fragmento del código EjemploPDF.java donde usamos `peek()` con ambas estructuras.

```

Uso de peek() para ver el elemento que está en la cima de la pila y el frente de la cola

Elemento en la cima de la pila: 9
Elemento en el frente de la cola: 0

Si imprimimos la pila y la cola, vemos que no se eliminan los elementos

Pila -> 0 1 2 3 4 5 6 7 8 9
Cola -> 0 1 2 3 4 5 6 7 8 9

```

Figura 6: Resultado de consola al usar `peek()` e imprimir

Eliminar/Sacar elementos

```

/**
 * Método que va eliminando elementos de la pila y de la cola
 * y los imprime en consola para observar el orden en el que se eliminan
 * y ver el cambio en la pila y la cola
 */
private static void eliminarElementos() {
    String eliminadosPila = "";
    String eliminadosCola = "";
    for (int i = 0; i < 10; i++) {
        // eliminamos elementos de la pila y de la cola
        eliminadosPila += pila.pop() + " ";
        eliminadosCola += cola.poll() + " ";
    }
    // las cadenas representan los elementos eliminados y el orden en el que se eliminaron
    System.out.println(GREEN + "\nElementos eliminados de la pila: " + eliminadosPila + RESET);
    System.out.println(YELLOW + "\nElementos eliminados de la cola: " + eliminadosCola + RESET
        + "\n");
}

```

Figura 7: Fragmento del código EjemploPDF.java donde sacamos elementos de ambas estructuras.

```

Ve el orden en el que se eliminan los elementos de la pila y de la cola

Elementos eliminados de la pila: 9 8 7 6 5 4 3 2 1 0
Elementos eliminados de la cola: 0 1 2 3 4 5 6 7 8 9

```

Figura 8: Resultado de consola al sacar elementos, observa que la cola los saca en el mismo orden que los guardo mientras que la pila los saca en el orden inverso.