



# Pilas y Colas

## Laboratorio Estructuras de Datos

### 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 pantalon 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(E item)*: 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.

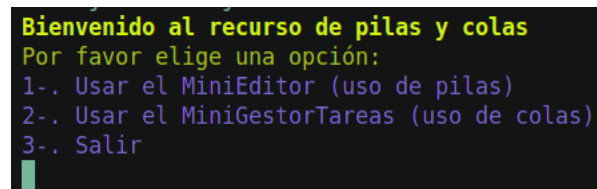
---

## Métodos de la cola

- *add(E item)*: 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.

## Código Recurso 02: Pilas y Colas

Al correr el programa podemos observar un primer menú, dependiendo de que queramos usar tecleamos el número correspondiente y pulsamos enter.



```
Bienvenido al recurso de pilas y colas
Por favor elige una opción:
1-. Usar el MiniEditor (uso de pilas)
2-. Usar el MiniGestorTareas (uso de colas)
3-. Salir
```

Figura 3: Podemos elegir si ver el ejemplo de pilas o el de colas.

## MiniEditor

El mini editor es un programa que simula un editor muy básico de texto puedes agregar texto, guardarlo en un archivo, deshacer y rehacer un renglón.

Este programa esta diseñado para ver un posible uso de la estructura *Pila*, usamos dos pilas una que guarda los renglones hechos y otra que guarda los renglones deshechos, las acciones van cambiando de una pila a otra dependiendo de la situación, también usa una lista como auxiliar para mantener en orden el texto ingresado, cuando una acción se rehace o se deshace también cambia la lista auxiliar para mantener el texto en la versión actual.

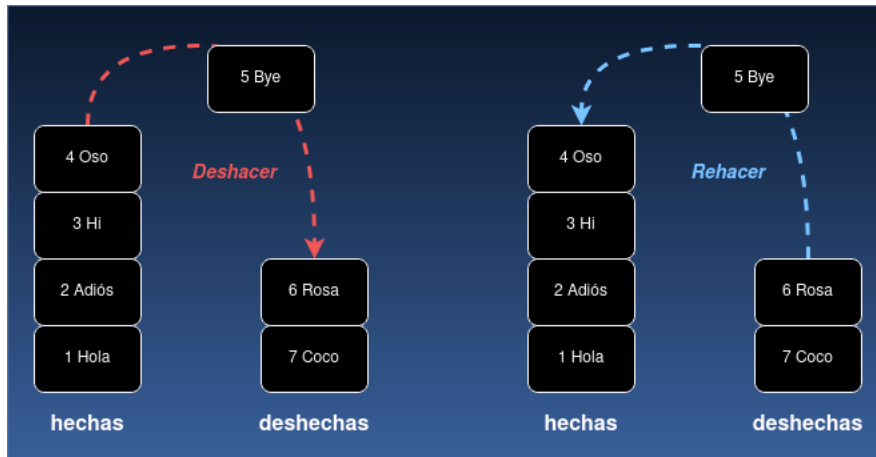


Figura 4: Tenemos dos pilas *hechas* y *deshechas* las cuales nos ayudan a mantener un registro de los cambios, los renglones del texto van a moverse entre ellas en las acciones de deshacer y rehacer.

El menú que se despliega al ejecutar el programa nos muestra las acciones que podemos realizar, debemos empezar por agregar texto.

```
Bienvenid@ al Mini Editor
1. Agregar texto
2. Mostrar texto
3. Deshacer
4. Rehacer
5. Guardar en un archivo
6. Salir
█
```

Figura 5: Menú del editor.

```
Ingrese el texto, para mayor claridad puede separar por renglones.
Cuando haya terminado deje una línea en blanco y pulse enter
1 Hola
2 Adiós
3 Hi
4 Oso
5 Bye
6 Rosa
7 Coco
```

Figura 6: Agregamos el texto de la misma forma que se ve en la Figura 3.

Si después de agregar texto elegimos la opción de deshacer 3 veces podemos ver como se van eliminando los renglones lo que pasa internamente se puede visualizar en la Figura 3.

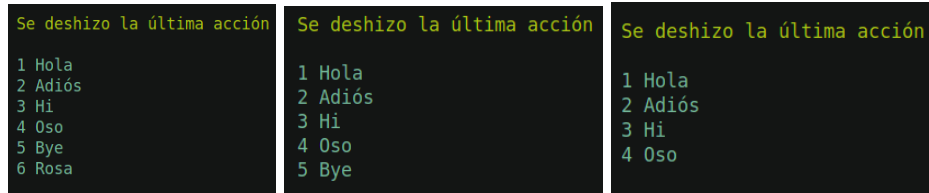


Figura 7: Cada imagen muestra la opción de deshacer.

La acción de rehacer es muy similar solo que ahora se van agregando los renglones que fuimos quitando.

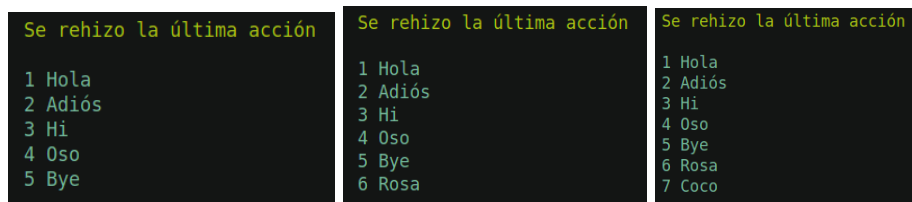


Figura 8: Cada imagen muestra la opción de rehacer.

## MiniGestorTareas

El gestor de tareas es un programa que simula un pequeño gestor al que podemos añadir tareas, ver la tarea actual, completar la tarea actual o ver todas las tareas pendientes.

Este programa esta diseñado como un posible ejemplo del uso de la estructura *Cola*, usamos una pila en donde se irán agregando las tareas.

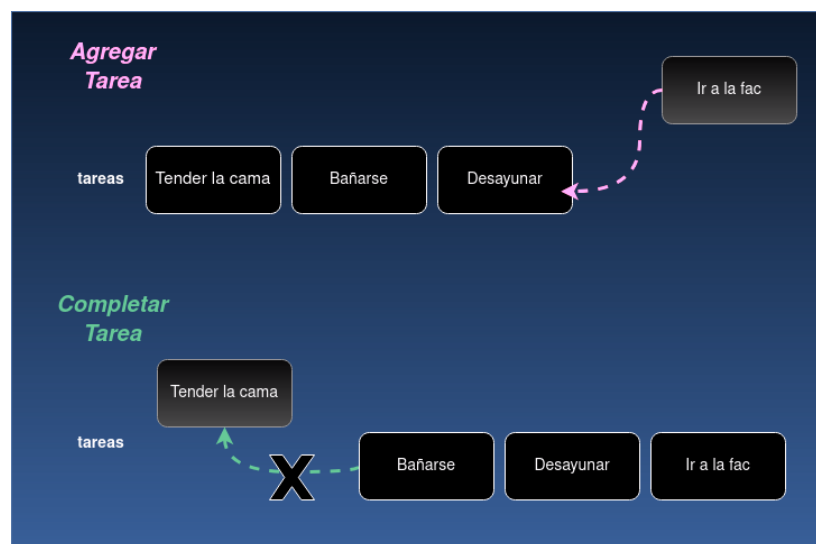


Figura 9: Operaciones de agregar tarea y completar tarea, cuando agregamos una tarea la incluimos al final y cuando completamos una tarea debe ser la primera agregada.

---

El menú que se despliega al ejecutar el programa nos muestra las posibles acciones que podemos realizar con las tareas.

```
Bienvenid@ al Mini Gestor de Tareas
1. Agregar tarea
2. Ver tareas
3. Ver la tarea actual
4. Completar la tarea actual
5. Salir
█
```

Figura 10: Menú del gestor.

La idea es que las tareas van por orden de prioridad entonces las primeras tareas agregadas deben ser las más importantes o las que se deben realizar primero.

<pre>Bienvenid@ al Mini Gestor de Tareas 1. Agregar tarea 2. Ver tareas 3. Ver la tarea actual 4. Completar la tarea actual 5. Salir 1 Ingrese la tarea: Tender la cama Tarea agregada</pre>	<pre>Bienvenid@ al Mini Gestor de Tareas 1. Agregar tarea 2. Ver tareas 3. Ver la tarea actual 4. Completar la tarea actual 5. Salir 1 Ingrese la tarea: Bañarse Tarea agregada</pre>	<pre>Bienvenid@ al Mini Gestor de Tareas 1. Agregar tarea 2. Ver tareas 3. Ver la tarea actual 4. Completar la tarea actual 5. Salir 1 Ingrese la tarea: Desayunar Tarea agregada</pre>
--	---	---

Figura 11: Tareas agregadas.

Podemos ver las tareas que tenemos:

```
Bienvenid@ al Mini Gestor de Tareas
1. Agregar tarea
2. Ver tareas
3. Ver la tarea actual
4. Completar la tarea actual
5. Salir
2
Tareas Guardadas:
[Tender la cama, Bañarse, Desayunar]
```

Figura 12: Tareas Guardadas.

También cuando queramos completar una tarea debe ser la actual, es decir, la primera en ser añadida.

---

```
Bienvenid@ al Mini Gestor de Tareas
1. Agregar tarea
2. Ver tareas
3. Ver la tarea actual
4. Completar la tarea actual
5. Salir
4
Desea completar la tarea actual? (s/n)
Tender la cama
s
La tarea Tender la cama ha sido completada
```

Figura 13: La tarea que podemos completar es aquella que ingresamos primero.