

TP 3 Estructura de Datos:Pilas y Colas

**Alumnos:Mariano Salto,
Santiago Smith**

Docente:Sergio González

Año: 2020

Pilas:

La pila es una estructura de datos para almacenar datos y recuperarlos, y es de tipo LIFO (Last In, First Out), esto quiere decir que el último elemento que entra en ésta es el primer elemento en salir de la pila.

Sus funciones principales son push(), que añade un elemento que recibe por parámetro y lo apila a la parte superior de la pila, pop() que devuelve el último elemento de la pila y lo elimina de la misma y top() que devuelve el elemento que está arriba de la pila sin eliminarlo; También cuenta con las operaciones empty() y size(), la primera devuelve si está vacía y la segunda devuelve el número de elementos que tiene la pila.

En nuestro trabajo al implementar estas funciones le asignamos los nombres de:

push = apilarElemento(elemento)

pop = desapilarElemento()

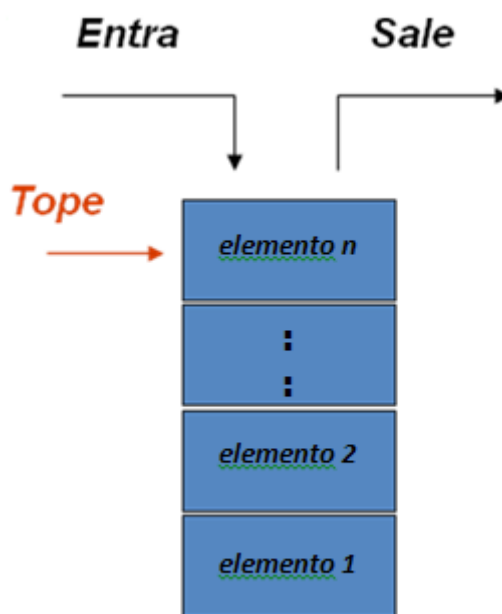
top = topePila()

empty = estaVacía()

size = tamañoPila()

y por último agregamos clonarPila, que como dice el nombre clona una pila ya existente.

Ejemplo Pila:



Colas:

La cola es una estructura de datos que almacena los datos encolándolos por un extremo y desencolándolos por el otro, es de tipo FIFO(First In First Out), se puede pensar también como una cola de supermercado, es decir el primer cliente que llega a la cola, es el primero en ser atendido, luego sigue el siguiente y así hasta tener la cola vacía.

Cuenta con las operaciones `queue()`, que añade el elemento en la posición 0 que a su vez es el final de la cola, y a medida que se van agregando elementos estos se van corriendo, `dequeue()` que devuelve el primer elemento que entró y lo elimina de la cola y `top()` o `front()` que devuelve el primer elemento de la cola, o sea el primero que entró; También cuenta con `empty()`, `len()` o `size()`, que devuelven si está vacía, y el tamaño de la cola.

En nuestra implementación también agregamos `clonarCola()`, que clona una cola ya creada y `vaciarCola()` que elimina todo los elementos que hay en la cola.

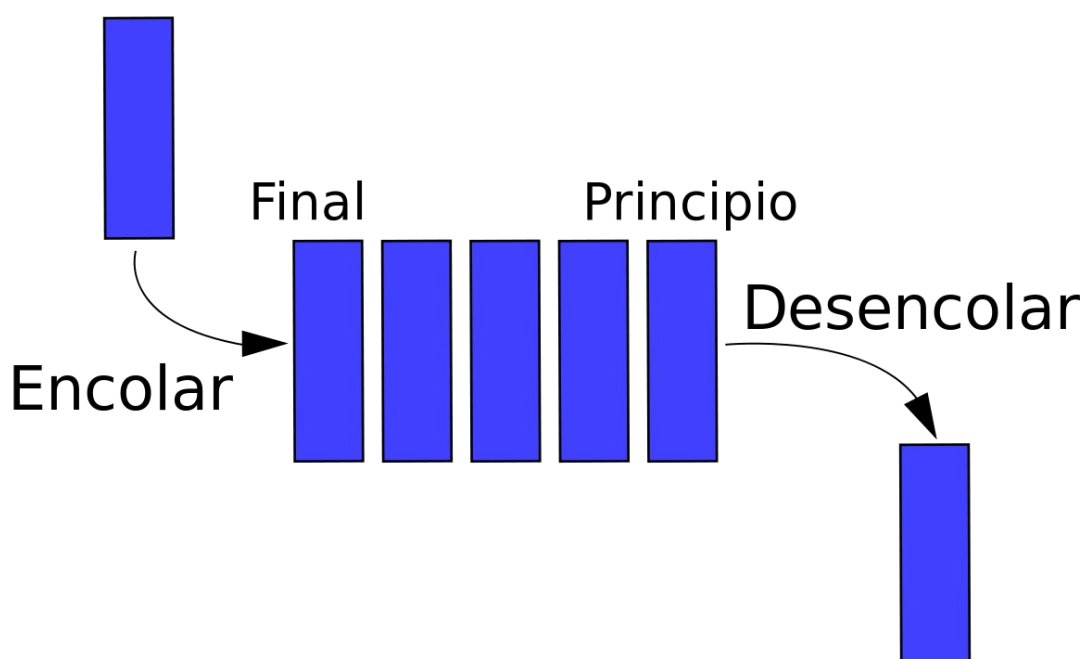
También le asignamos diferentes nombres para poder hacer un seguimiento del código y entenderlo de mejor manera :

`queue = encolar(elemento)`

`dequeue = desencolar()`

`top = primerElementoFila()`

Ejemplo Cola:



Conclusión:

Nuestra manera de ir desarrollando el código fue dividiendo, en problemas más pequeños para poder ir probando el código en las que nosotros considerabamos funciones un poco más complejas, y si el problema que planteábamos no funcionaba partíamos desde la base del problema, así hasta llegar a la solución que pedía o requería la función.

También al implementar las pilas y las colas, a las operaciones les asignamos nombres para mejor entendimiento y para que el código sea un poco más legible; Somos conscientes que a la hora de implementar Tribunales el código podría estar mejor, sin repeticiones pero se nos complicó, por ejemplo en la operación `cantidadDeJuzgadosCriticos()`, esta función tendría que ser recursiva pero no conseguimos la manera de implementarla, igualmente la resolvimos de forma iterativa.

La explicación de las funciones están explicadas y comentadas en el código de cada TDA.

Bibliografía:

[https://es.wikipedia.org/wiki/Pila_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Pila_(inform%C3%A1tica))

[https://es.wikipedia.org/wiki/Cola_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Cola_(inform%C3%A1tica))

<https://es.slideshare.net/diwal10/pilas-colas-y-listas-estructura-de-datos>

http://agrega.juntadeandalucia.es/repositorio/02122016/a5/es-an_2016120212_9131705/34_colas.html