
TDAs

— Pila y Cola —

TDA PILA

TDA PILA

Una pila es una colección ordenada de elementos en la que pueden insertarse y eliminarse por un extremo, denominado **tope**, sus elementos.

Normalmente existe un conjunto de operaciones que se puede realizar sobre un tda. ¿Estas operaciones son estándares? No. Existe un pequeño conjunto que siempre debería estar y después las mismas varían según las necesidades del programador.

Al conjunto que siempre debería estar se lo denominará **conjunto mínimo de operaciones**.

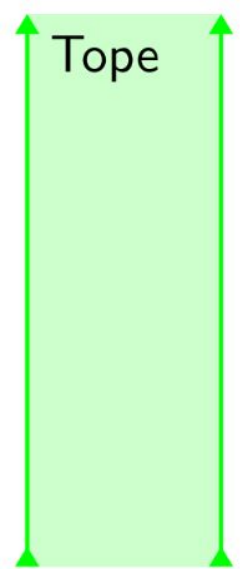
TDA PILA

Para el caso de una pila el conjunto mínimo de operaciones es:

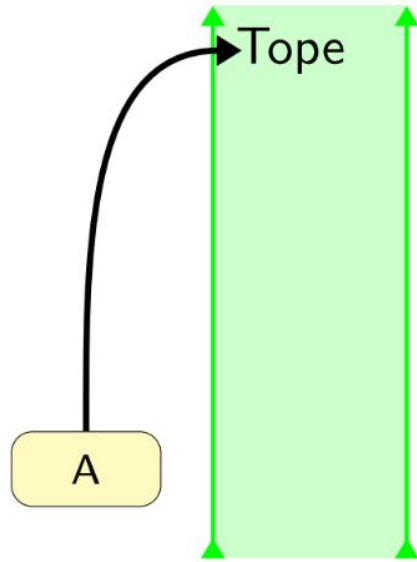
- crear (create)
- poner (push)
- sacar (pop)
- tope (top)
- esta_vacia (is_empty)
- destruir (destroy)

TDA PILA: poner() / push()

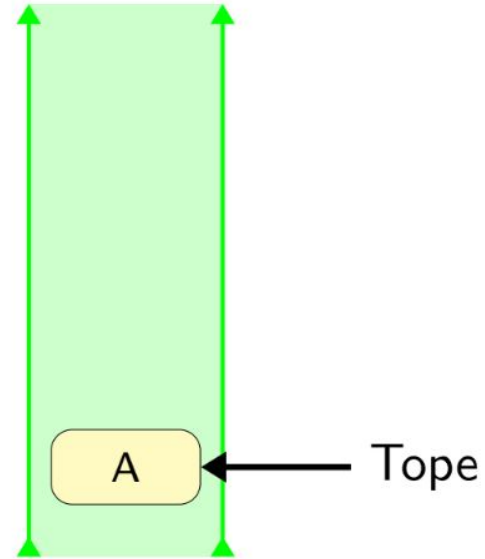
Esta operación pone un elemento en la pila por el tope de la misma, haciendo que el tope de la pila pase a ser el nuevo elemento introducido.



Pila Vacía

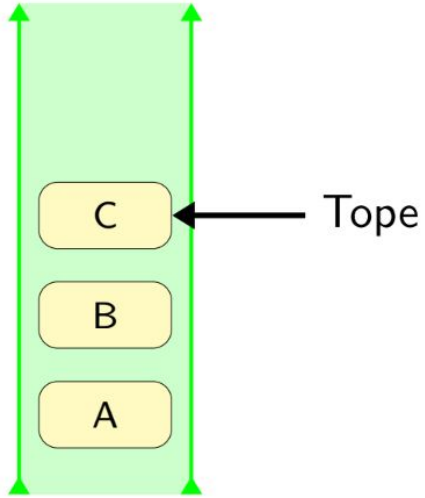


Pila: Poner(A)

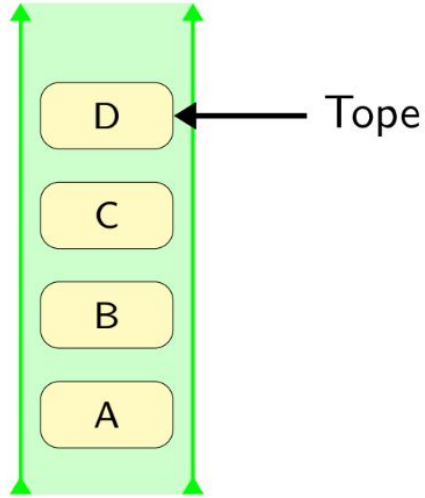


Pila

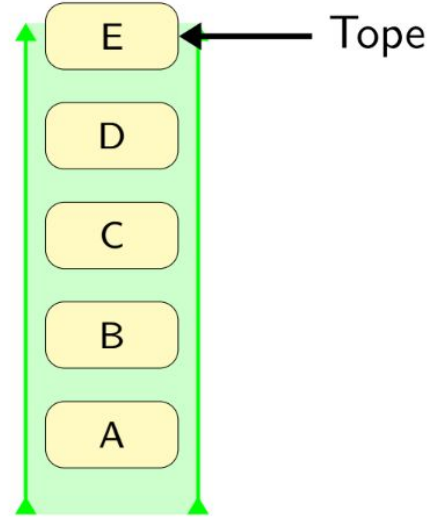
TDA PILA: poner() / push()



Pila: Poner(C)



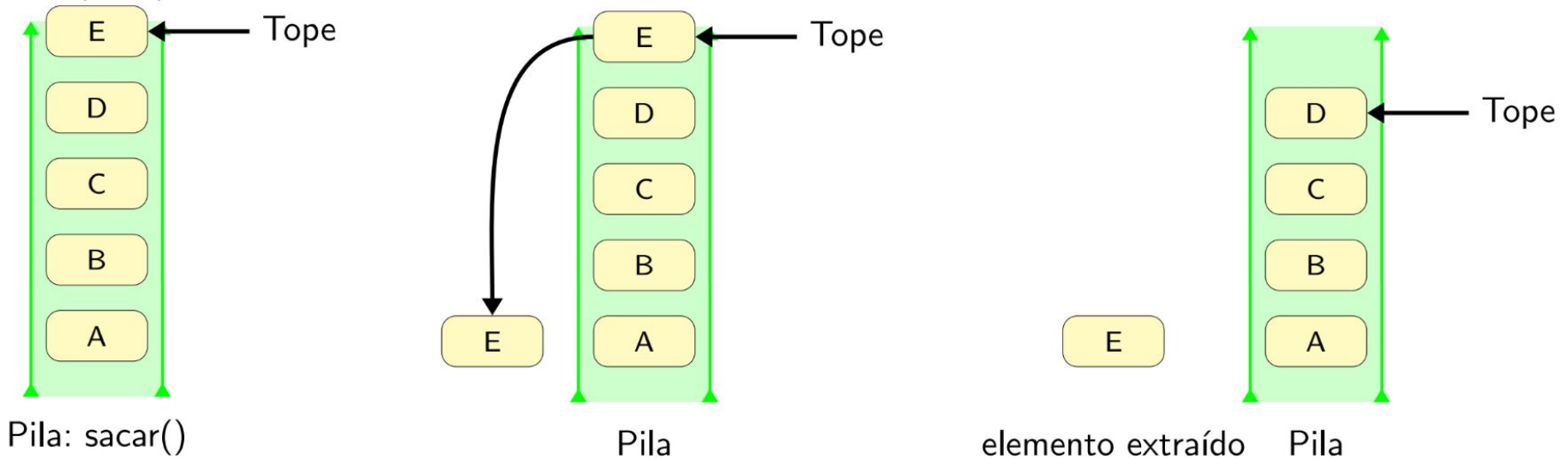
Pila: Poner(D)



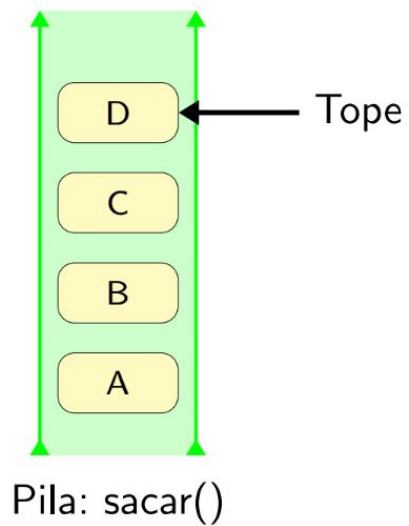
Pila: Poner(E)

TDA PILA: sacar() / pop()

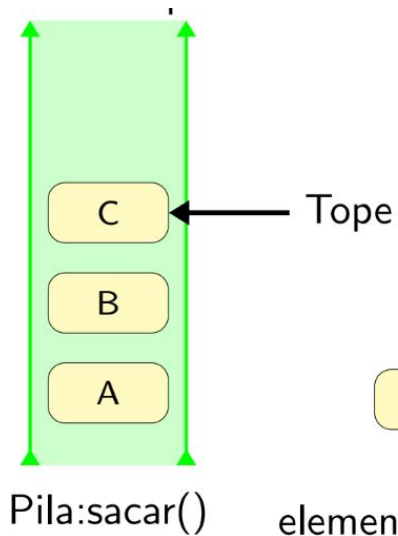
Esta operación retira el elemento del tope de la pila y mueve el tope de la pila al elemento anterior al extraído, si el elemento extraído es el último la pila queda vacía:



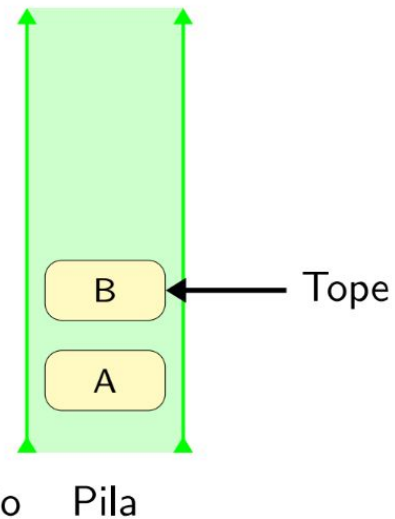
TDA PILA: sacar() / pop()



elemento extraído



elemento extraído



TDA PILA

- **crear (create)**: Esta operación se ocupa de crear y reservar todos los recursos iniciales que se utilizarán para la pila.
- **tope (top)**: Esta operación permite observar el valor del tope de la pila.
- **esta_vacia (is_empty)**: Esta operación permite determinar si una pila tiene o no tiene elementos a través de un valor de verdad.
- **destruir (destroy)**: Esta operación se ocupa de liberar y limpiar todos los recursos que se utilizan para la creación de una pila.

Ejemplo

TDA COLA

TDA COLA

Una cola es una estructura que posee dos extremos por los que se realizan operaciones:

Un extremo es el inicio o frente de la cola y el otro extremo es el final o rear de la cola.

Este tipo de estructura sirve para modelar procesos como la cola de un colectivo, la cola de un supermercado, el despacho de combustible, entre muchos otros.

TDA COLA

Para el caso de una cola el conjunto mínimo de operaciones es:

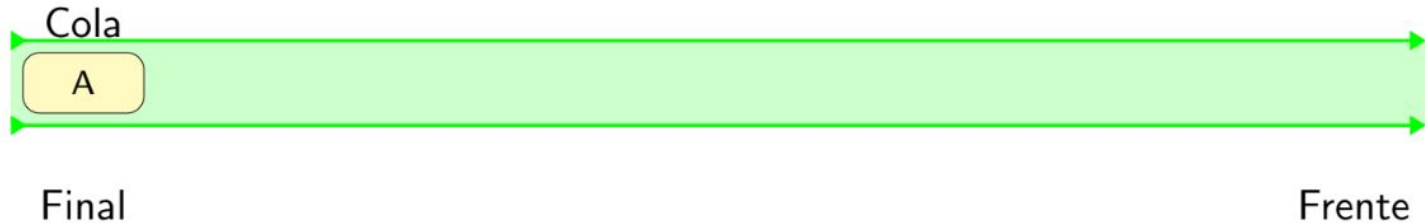
- crear (create)
- encolar (enqueue)
- desencolar (dequeue)
- primero (first)
- esta_vacia (is_empty)
- destruir (destroy)

TDA COLA: encolar() / enqueue()

La idea de esta operación es que los elementos se encolan por el final de la cola. Un elemento nuevo llega y se coloca al final de la cola para respetar el orden de llegada.



```
1 encolar(A)
```



TDA COLA: encolar() / enqueue()

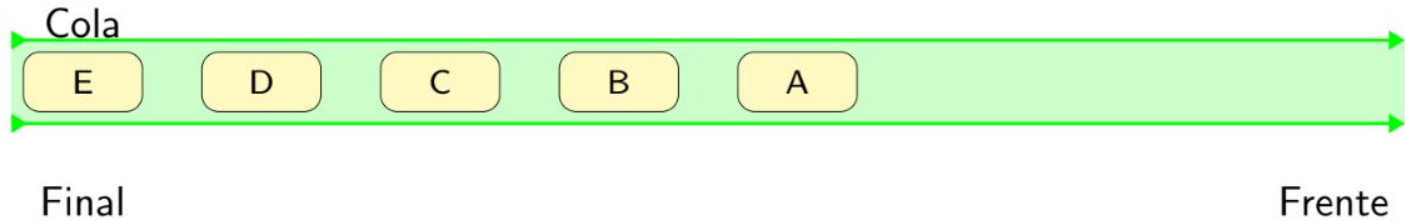
```
1 encolar(B)
```



```
1 encolar(C)
```

```
2 encolar(D)
```

```
3 encolar(E)
```



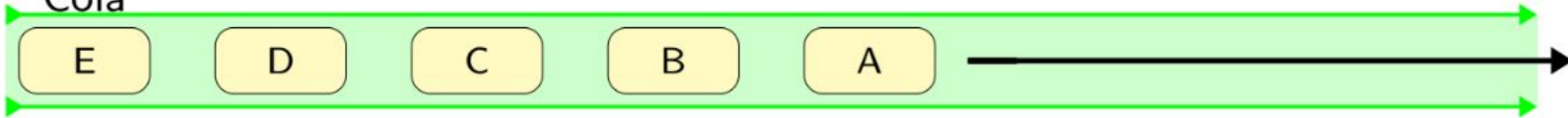
TDA COLA: desencolar() / dequeue()

Esta operación, extrae del frente de la cola el elemento que se encuentra en él.

La norma en que opera este tipo de estructura se denomina **First In First Out o FIFO**. El primer elemento que entra en la cola es el primer elemento que sale de la misma, es decir, se respeta el orden de llegada de los mismos.

```
1 desencolar()
```

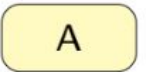
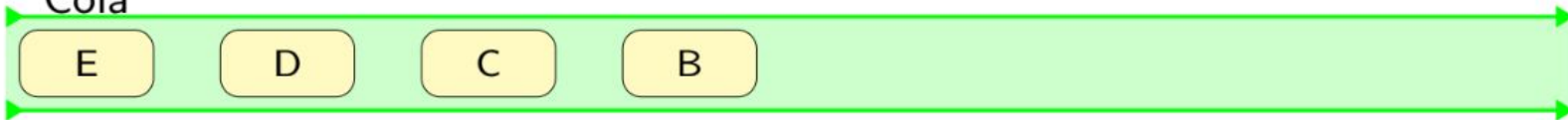
Cola



Final

Frente

Cola



Final

Frente

Desencolado

TDA COLA

- **crear (create)**: Esta operación se ocupa de crear y reservar todos los recursos iniciales que se utilizarán para la cola.
- **primero (first)**: Esta operación permite observar cual es el primer elemento del frente de la cola.
- **esta_vacia (is_empty)**: Esta operación permite determinar si una cola tiene o no tiene elementos a través de un valor de verdad.
- **destruir (destroy)**: Esta operación se ocupa de liberar y limpiar todos los recursos que se utilizan para la creación y uso de una cola.

Ejemplo