TAD Lista, Pila, Cola y Deque
(introducción)

ALGORITMOS Y ESTRUCTURAS DE DATOS I

UNIVERSIDAD ORT
Uruguay

Tipo Abstracto de Datos (TAD)

- Un TAD es **un tipo de datos junto con las operaciones definidas para él (procedimientos de acceso)**, independientemente de su implementación.

- Si bien permite una especificación con precisión, no está dado como un tipo de datos concreto del lenguaje.

Tipo Abstracto de Datos (TAD)

Como se define?

Básicamente dándole un nombre y asociando a él un conjunto de operaciones aplicables a los elementos del tipo. **Definimos qué es lo que hace y no cómo lo hace.**

Como se implementa?

Asociando un método (código) a cada una de las operaciones definidas sobre el conjunto. **Definimos cómo lo hace.**

Tipo Abstracto de Datos (TAD)

Ventajas

- La separación de la especificación (definición) y la implementación ayuda a reducir la complejidad de la tarea a realizar.

- El cambio en la implementación de alguna de las operaciones del TAD no afecta a los programas que utilizan las operaciones de dicho TAD.

TAD Lista

- Una lista es una estructura de datos que contiene una secuencia lineal de un número arbitrario de ítems del mismo tipo.

- Llamaremos Nodo a cada elemento de la lista. Cada nodo tiene al menos:

✓ Un ítem de información (de cualquier tipo).
✓ Un puntero al siguiente nodo de la lista.

TAD Lista (posibles operaciones)

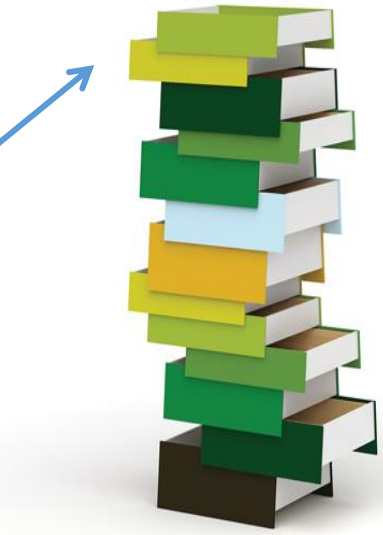
```
public boolean esVacia();
public void agregarInicio(int n);
public void agregarFinal(int n);
public void borrarInicio();
public void borrarFin();
public void vaciar();
public void mostrar();

public void agregarOrd(int n);
public void borrarElemento(int n);
public int cantElementos();
public NodoLista obtenerElemento(int n);
public void mostrarREC(NodoLista l);
```

Pila (Stack)

- Las pilas son listas con la restricción que las inserciones y eliminaciones debe realizarse solamente en una posición: la cima de la pila (final de la lista), llamada top.

- Las pilas son conocidas también como LIFO (Last In First Out, último en ingresar es el primero en irse)

(Solo el elemento top es accesible)



Pila (Stack)

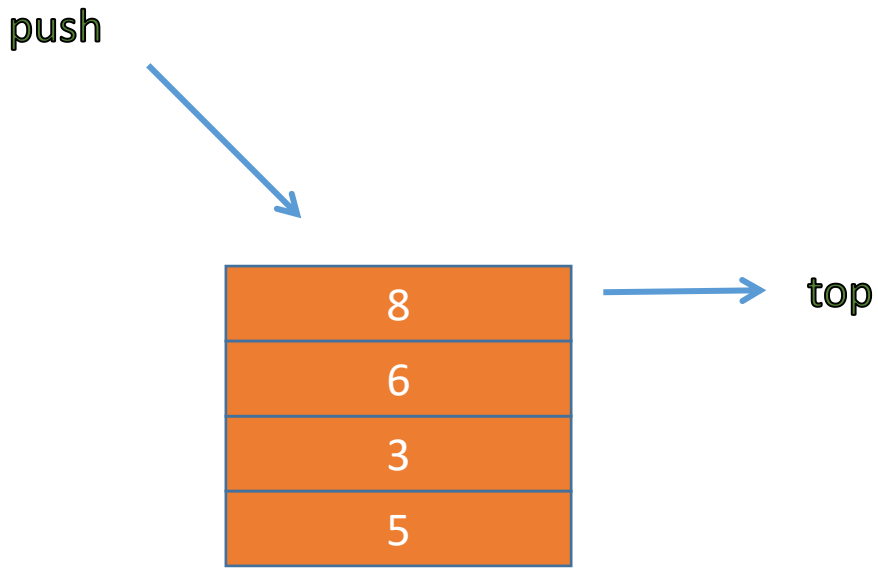
Operaciones básicas

COMBINA

push
Insertar un elemento
(la pila no está llena)

pop
Elimina el ultimo elemento insertado
(la pila no está vacía)

top
Retorna el ultimo elemento insertado
(la pila no está vacía)



Pila (Stack)

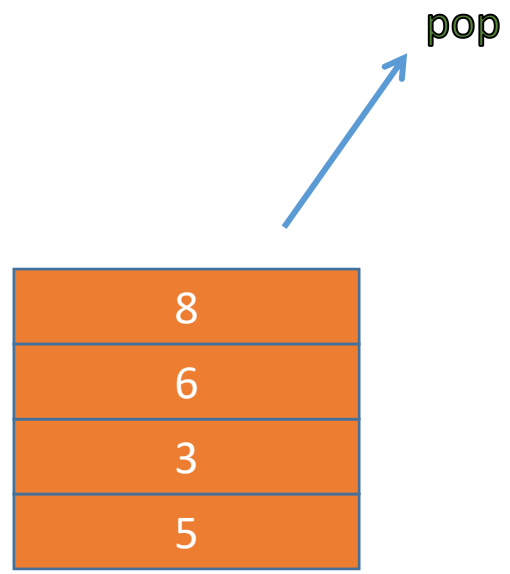
Operaciones básicas

push
Insertar un elemento
(la pila no está llena)

COMBINA

pop
Elimina el ultimo elemento insertado
(la pila no está vacía)

top
Retorna el ultimo elemento insertado
(la pila no está vacía)

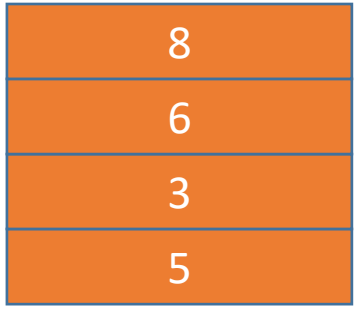


Pila (Stack)

Ejemplo

Si realizamos las siguientes operaciones obtendremos:

```
unaPila.Push(5)
unaPila.Push(3)
unaPila.Push(6)
unaPila.Push(8)
```

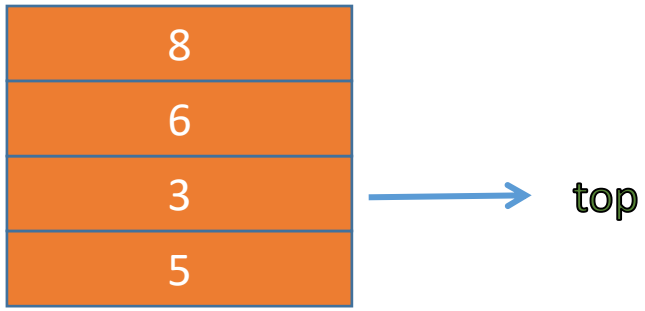


Pila (Stack)

Ejemplo

Si realizamos las siguientes operaciones obtendremos:

```
unaPila.pop()
unaPila.pop()

unaPila.top()
```



Cola (Queue)

- Las colas son listas con la restricción que, las inserciones se deben realizar al final (back) y se debe eliminar desde el principio (front).

Cola (Queue)

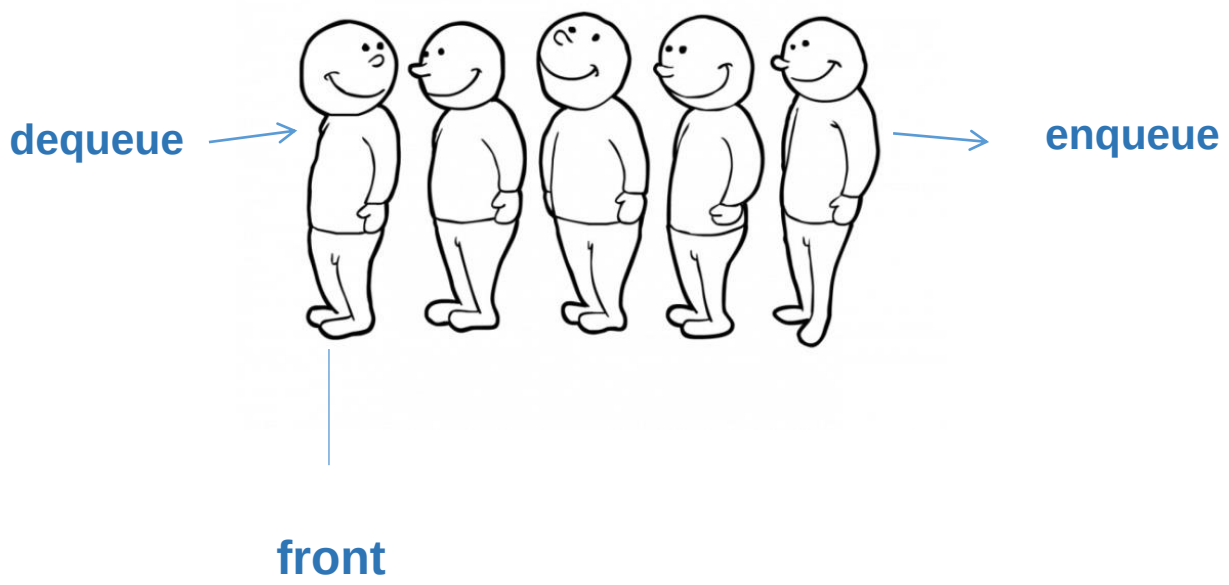
Operaciones básicas

enqueue
Insertar un elemento al final de la cola
(la cola no está llena)

COMBINA

dequeue
Elimina el primer elemento de la cola
(la cola no está vacía)

front
Retorna el primer elemento de la cola
(la cola no está vacía)



Deque

- Una deque es una fila (o cola) de dos puntas. Esto es, una estructura lineal donde las inserciones y las bajas pueden efectuarse en cualquiera de sus extremos.

Deque

Operaciones básicas

pushFront
Insertar un elemento al comienzo de la cola

pushBack
Insertar un elemento al final de la cola

front
Retorna el primer elemento de la cola

back
Retorna el último elemento de la cola

popFront
Elimina el primer elemento de la cola

popBack
Elimina el último elemento de la cola