FACULTA DE INGENIERÍA Y ARQUITECTURA
CARRERA DE INGENIERÍA DE SISTEMAS

TDA PILA

UNIVERSIDAD DE LIMA

UNIVERSIDAD DE LIMA

Definición:

Una pila (stack) es una estructura de dato lineal, en la que la inserción y eliminación de datos se realiza solo por un extremo denominado cima o tope.

Por ejemplo:

- ❑ Una pila de bandejas en una cafetería.
- ❑ Una pila de platos en un fregadero.
- ❑ Una pila de latas en un expositor de un supermercado.

UNIVERSIDAD DE LIMA

En cualquiera de estos ejemplos, los elementos se retiran y se añaden por un mismo extremo.

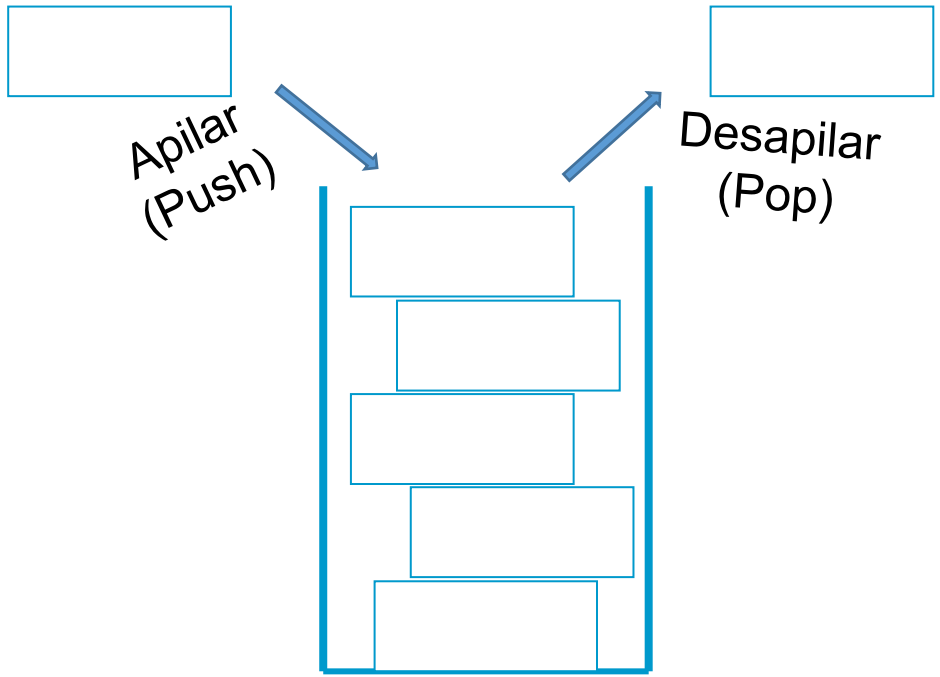
Las pilas siguen la lógica LIFO (Last In First Out), porque su característica principal es que el Ultimo en Entrar es el Primero en Salir.

Pila de platos

Pila de libros

Pila de tazas

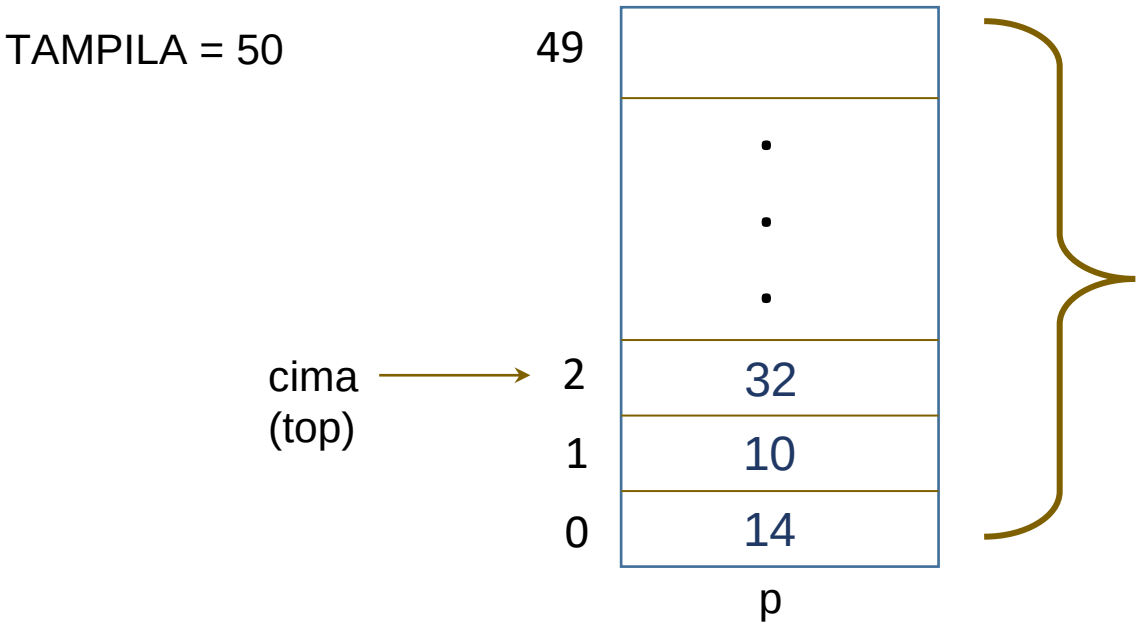
Se define el tope o cima de la pila al único dato visible de la estructura que seria el ultimo que se colocó (el que está encima).



Representación de una Pila

Representación en Memoria

Es posible representarlas con un arreglo p de tamaño TAMPILA, y denominando cima a la variable que contiene la posición del elemento que se encuentra en la cima de la pila.



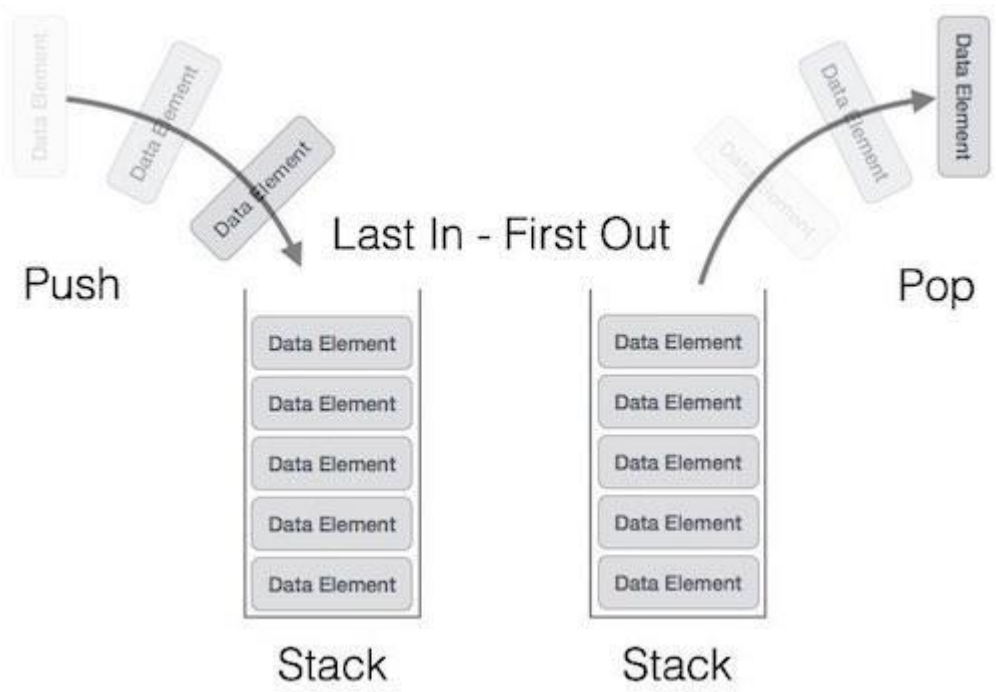
La pila p almacena números enteros.

Otra opción: Listas enlazadas

UNIVERSIDAD DE LIMA

Operaciones

- ❑ crearPila
- ❑ estaVacia (isEmpty)
- ❑ estaLlena (isFull)
- ❑ apilar (push)
- ❑ Desapilar (pop)



UNIVERSIDAD DE LIMA

(1) Crear Pila

Especificaciones:

Objetivo: crea la pila vacía

Entrada: ninguna

Precondición: ninguna

Salida: ninguna

Postcondición: Pila creada y vacía

```
public Stack(){
        data = new int[MAX_SIZE];
        top = -1;
}
```

UNIVERSIDAD DE LIMA

(2) Está Vacía

Especificaciones:

Objetivo: verifica si la pila está vacía.
Entrada: ninguna.
Precondición: ninguna.
Salida: estado (lógico).
Postcondición: retorna verdadero si la pila está vacía, falso en caso contrario.

```
public boolean isEmpty(){
    boolean ans = (top == -1);
    return ans;
}
```

UNIVERSIDAD DE LIMA

(3) Está Llena

Especificaciones:

Objetivo: verifica si la pila p está llena.
Entrada: ninguna.
Precondición: ninguna.
Salida: estado (lógico)
Postcondición: estado verdadero si la pila p está llena, falso en caso contrario.

```
private boolean isFull(){
      boolean ans = (top + 1 == MAX_SIZE);
      return ans;
}
```

UNIVERSIDAD DE LIMA

(4) Apilar (Push)

Especificaciones:

Objetivo: añadir un nuevo elemento a la pila p.
Entrada: item.
Precondición: ninguna.
Salida: ninguna.
Postcondición: pila p con un nuevo elemento apilado.

```
public void push(Integer item){
        if(!this.isFull()){
                top++;
                data[top] = item;
        }
}
```

UNIVERSIDAD DE LIMA

(5) Desapilar (Pop).

Especificaciones:

Objetivo: elimina un elemento de la pila p.
Entrada: ninguna
Precondición: Pila p no vacía
Salida: valor
Postcondición: pila p con un elemento menos.

```
public Integer pop(){
        Integer value = null;
        if(!this.isEmpty()){
                value = data[top];
                top--;
        }
        return value;
```

UNIVERSIDAD DE LIMA

EJERCICIO

Considere la operación Intercambiar Datos, el cual permite intercambiar valorAntiguo por valorNuevo en una estructura pila p, que contiene números enteros.
Escriba el algoritmo para esta operación.

Ejemplo: