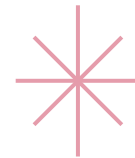
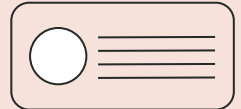
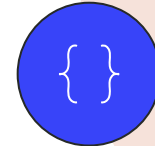






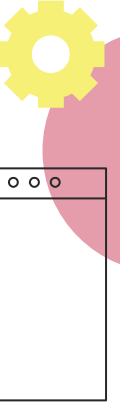
Tipos abstractos de datos TDA

Pilas



Unidad 3 TAD Lineales

-  Tipos abstractos de datos
-  Pilas - Stack
-  Implementaciones
-  Implementación con arreglos



Tipos abstractos de datos

Algunos lenguajes de programación tienen características que nos permiten **ampliar el lenguaje** añadiendo sus propios tipos de datos.

Un tipo de dato definido por el programador se denomina **tipo abstracto de datos (TAD)** para diferenciarlo del tipo fundamental (predefinido) de datos.

Ejemplo: tipo Punto, que representa las coordenadas x e y de un sistema de coordenadas rectangulares

Es posible implementar el tipo abstracto de datos, considerando los valores que se almacenan en las variables y qué operaciones están disponibles para manipular estas variables

Un tipo abstracto es un tipo de dato que consta de datos (estructuras de datos propias) y operaciones que se pueden realizar sobre ellos.

Un TAD se compone de estructuras de datos y los procedimientos o funciones que manipulan esas estructuras de datos.





Tipos abstractos de datos

TIPOS ABSTRACTOS DE DATOS EN JAVA

La implementación de un TAD en Java se realiza de forma natural con una clase. Dentro de la clase va a residir la **representación de los datos** junto a **las operaciones** (métodos de la clase). La interfaz del tipo abstracto queda perfectamente determinada con la etiqueta public, que se aplicará a los métodos de la clase que representen operaciones.

- •
- •
- •
- •
- •
- •
- •
- •



Tipos abstractos de datos

Ejemplo

Si se ha especificado el TAD Punto para representar la abstracción punto en el espacio tridimensional, la siguiente clase implementa el tipo:

```
class Punto
```

```
{// representación de los datos
```

```
private double x, y, z;
```

```
// operaciones
```

```
public double distancia(Punto p);
```

```
public double modulo();
```

```
public double anguloZeta();
```

```
};
```

• •
• •
• •
• •
• •
• •
• •
• •

Pilas - Stack

Característica principal:

Estructura de datos de acceso restrictivo a sus elementos

Definición:

- ... Lista ordenada de elementos a la cual se puede insertar o
- ... eliminar elementos sólo por uno de los extremos (Al que
- ... llamaremos TOPE).
- ... Los elementos se eliminan en orden inverso al que se insertan.
- ... El último elementos que se inserta en la pila es el primero en
- ... salir

Estructura LIFO: Last In First Out



Pilas - Stack

Ejemplos:

Pila de platos

... Pilas de libros

... Envase de pelotas de Tenis

... Invocación de subrutinas en lenguajes
... de programación



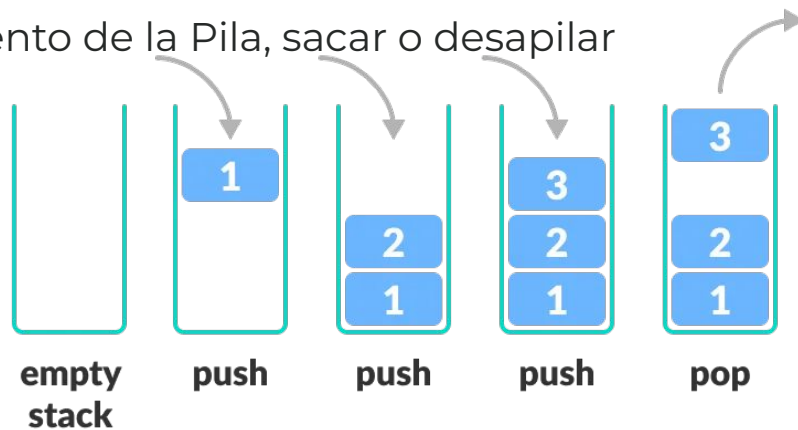
Pilas - Stack (Operaciones)

El ementos superior se llama **TOPE**

PUSH: Insertar o agregar un elemento a la Pila, también se puede decir poner o apilar.

NO existe límite o restricción para agregar elementos (crece indefinidamente), aunque dependiendo de la implementación puede haber límite.

POP: Quitar un elemento de la Pila, sacar o desapilar



Pilas - Stack (más operaciones)

Pila Vacía

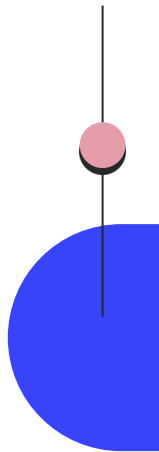
Comprobar si la pila está vacía.
Operación necesaria para saber si se pueden eliminar elementos
NO se puede eliminar de una pila vacía, se produciría underflow

Ver tope

Ver el elemento en el tope de la pila sin removerlo.

Pila Llena

Comprueba si la pila está llena.
Operación necesaria para controlar si se puede insertar un nuevo elemento.
Si la pila está llena y se intenta insertar se produciría overflow



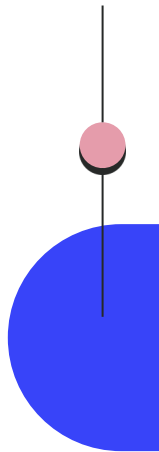
Pilas - Stack Implementación

La codificación de una pila requiere equilibrio: si la longitud máxima (o tamaño) de la pila es grande puede gastar mucha memoria innecesariamente.

Si la longitud máxima tiene un valor muy pequeño, se producirán desbordamientos frecuentes.

Se deben diseñar los métodos:

- poner (push)
- quitar (pop)
- pilavacia: Comprobar si la pila está vacía
- pilallena: Comprobar si la pila está llena
- vertope: Obtener el valor de elemento que está en el tope de la pila sin removerlo de la pila

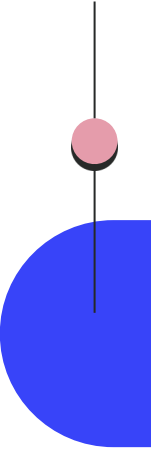
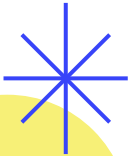


Pilas - Stack Implementación

Las pilas son estructuras de datos fundamentales que por lo general **no están** definidas en los Lenguajes de Programación

Dos formas de implementar las pilas son mediante:

- Arreglos (opción estática)
- Listas Enlazadas (opción dinámica)



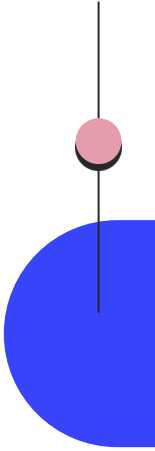
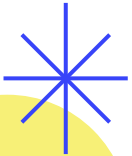
Pilas - Stack Implementación con arreglos

Se necesita un vector o arreglo de tamaño N y una variable numérica llamada tope que va a indicar la posición en el vector donde se encuentre el tope de la pila.

Previamente se limita el máximo número de elementos que la pila puede contener (tamaño del vector N).

Ejemplo:

$N = 100$





Pilas - Stack Implementación con arreglos

Estructura - Atributos de la clase

```
int n = 100  
tipoDeDato Pila[n]  
int tope = 0
```

..
..
..
..
..
..
..
..
..
..





Pilas - Stack Implementación con arreglos

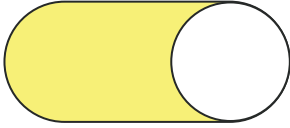
Comportamiento- Métodos de la clase

```
int n = 100  
tipoDeDato Pila[n]  
int tope = -1
```

..
..
..
..
..
..
..
..
..
..

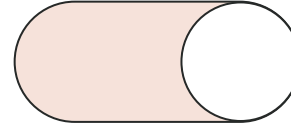


Pilas - Stack Implementación con arreglos



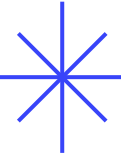
Push

```
PUSH(tipoDeDato x)
si (Tope < n-1)
    Tope = Tope + 1
    Pila[Tope] = x
sino
    Mostrar("La estructura está
    completa, no se puede
    insertar")
```

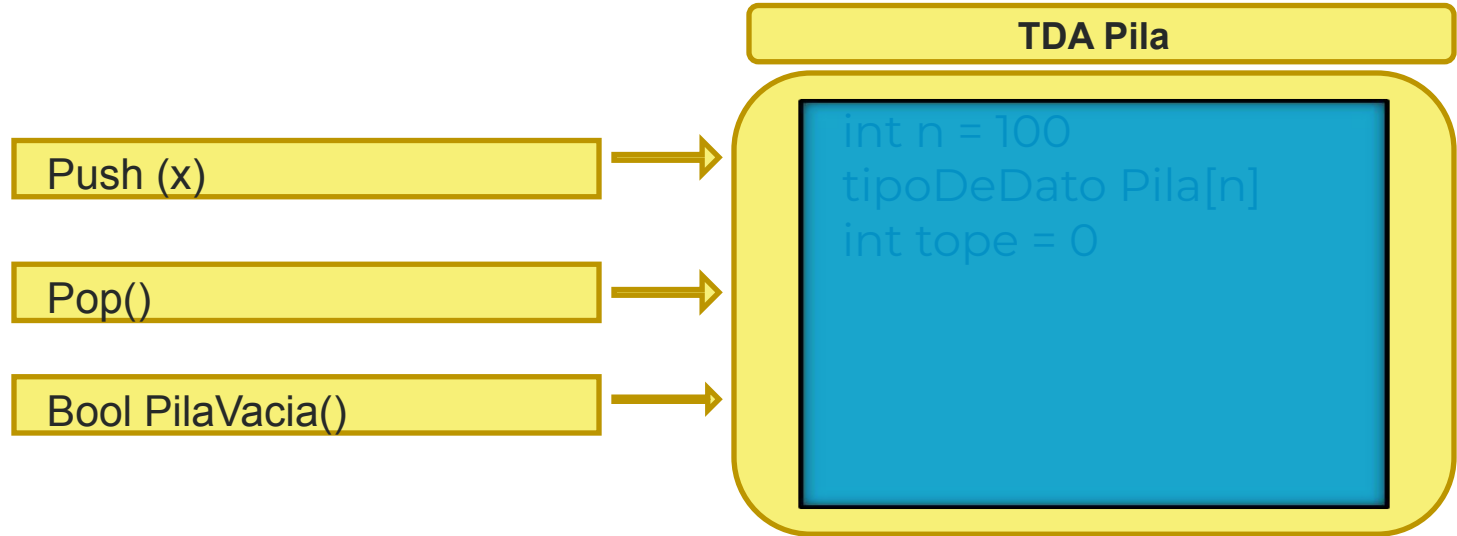


Pop

```
POP()
si (Tope >= 0)
    Tope = Tope - 1
sino
    Mostrar("La estructura está
    vacía, no se puede eliminar")
```



Pilas - Stack



Pilas - Stack

Una vez definido un TDA se podrán utilizar variables/objetos de ese tipo

... Si tenemos el TDA Pila podremos definir variables/objetos de tipo Pila

... Pila P1,P2

... Para utilizarlas utilizaremos la sintaxis

... <nombre de la variable>.<nombre de la operación>

...
...



Pilas - Stack

Pila P1,P2

TipoDeDato s

// Lee 10 elementos y los coloca en la pila P1

para int i = 1 hasta 10

Leer (s)

P1.push(s)

fin para

para i = 1 hasta 10

s = P1.LeerTope()

P2.push(s)

fin para

Se desea realizar un método que copie el contenido de una pila a otra.

El código de la izquierda lo resuelve?

En caso de no ser así, ¿cómo lo modificaremos?



Resumen

Una pila es un TDA tipo LIFO (last in first out, último en entrar primero en salir) en la que los datos (todos del mismo tipo) se añaden y se eliminan por el mismo extremo, denominado cima de la pila.

Se definen las siguientes operaciones básicas sobre pilas: crear, push, verTope, pop, pilaVacía, pilaLlena y limpiarPila.

Las pilas son TAD relevantes para representar distintos problemas de la realidad

