

# EJERCICIO RESUELTO

## Módulo: Programación

---

### Colas y pilas

#### Descripción:

Crear una clase a través de arrays que simule una PILA (LIFO) de información.

Una pila nos permite almacenar y recuperar datos en memoria. Sigue la política LIFO (Último en Entrar Primero en Salir).

En este ejercicio se pide implementar una clase que modele el comportamiento de una pila de enteros, a través de arrays. La clase debe permitirnos las siguientes funciones:

- ☐ Crear la pila vacía.
- ☐ Insertar elementos.
- ☐ Extraer elementos.
- ☐ Debe haber un método para saber si pila está vacía.
- ☐ Un método para saber si la pila está llena.
- ☐ Un método que nos indique el número de elementos actual de la pila.
- ☐ Método que imprima todos los elementos de la pila

#### Objetivos:

- Aprender a trabajar con arrays
- Manejo básico de arrays.
- Utilizar el entorno integrado de desarrollo en la creación y compilación de programas

#### Recursos:

- Ordenador con su correspondiente sistema operativo
- NetBeans+JDK

## Resolución:

```
/**
 * Pila
 */
class Pila
{
    /** Máximo número de elementos a insertar en la pila*/
    private int numElementos;

    /** Array donde se insertan los elementos */
    private int elementos[];
    /** Referencia al último elemento insertado. */
    private int indice;

    /** Inicialización de la pila*/
    Pila(int numElementos)
    {
        this.numElementos = numElementos;
        indice = -1;
        // Creamos el array
        elementos = new int [numElementos];
    }

    /** Indica si la pila está vacía */
    public boolean vacia()
    {
        return (indice == -1);
    }

    /** Indica si la pila está llena */
    public boolean llena()
    {
        return (indice == numElementos-1);
    }

    /** Inserta un elemento en la pila. */
    public void apilar(int elemento)
    {
        indice++;
        elementos[indice] = elemento;
    }

    /** Saca un elemento de la pila. */
    public int desapilar()
    {
        int elemento = elementos[indice];
        indice--;
        return elemento;
    }

    /** Devuelve el número de elementos que tiene la pila */
    public int numElementos()
    {
        return indice+1;
    }

    /** Imprime los elementos de la pila */
    public void imprimir()
    {
        for (int i=0; i<=indice; i++)
            System.out.print(elementos[i] + "-");
    }
}
```

```
        System.out.println();
    }
} // Pila

////////////////////////////////////
En Main realizaremos unas llamadas a los métodos para comprobar el funcionamiento de la clase:

/** Prueba de pila */
public static main()
{
    Pila miPila = new Pila(5);

    System.out.println("Vacía pila: " + miPila.vacia());
    miPila.apilar(5);
    miPila.apilar(4);
    miPila.apilar(3);
    miPila.apilar(2);
    miPila.apilar(1);
    System.out.println("Llena pila: " + miPila.llena());
    System.out.println("Número de elementos: " +
        miPila.numElementos());

    miPila.imprimir();
}
```

Con estos datos de ejemplo el resultado obtenido será:

```
Vacia pila: true
Llena pila: true
Numero de elementos: 5
5-4-3-2-1-
```