

# Prácticas de Estructuras de Datos y de la Información.

## Práctica 5. La clase Pila

### **Objetivos**

El principal objetivo de esta práctica es trabajar el concepto elemental en POO de herencia como base de la reutilización del código, uno de los principales factores de calidad del software. En este caso reutilizamos lo hecho en la clase Lista de la práctica anterior para desarrollar en muy poco tiempo la clase Pila. Por otro lado es muy importante entender la diferencia entre herencia pública y herencia privada.

### **Enunciado**

Diseñar e implementar el TAD Pila a partir de el TAD Lista de la práctica anterior. El interfaz del TAD

Pila estará formado por las siguientes operaciones típicas de las pilas:

1. `tama()`: número de elementos de la pila.
2. `vacía()`: comprueba si la pila se encuentra vacía.
3. `push()`: apila un elemento que recibe como parámetro.
4. `pop()`: saca de la pila un elemento y lo devuelve.
5. `top()`: devuelve el elemento que hay en la cima de la pila sin eliminarlo (esta operación también suele denominarse `cima()`).

Se deben considerar los siguientes aspectos:

- Se creará una clase en C++, denominada Pila.
- La definición de la clase irá en el fichero `pila.h`, y el cuerpo de las funciones en el fichero `pila.cpp`
- Se realizará un programa de prueba (`menupila.cpp`), que desde la consola permitirá acceder a cada una de las operaciones anteriores además de permitir visualizar todos los elementos de la pila en pantalla para comprobar el estado de la pila cuando se desee.

### **Referencias**

Más información sobre herencia en la siguiente dirección:

<http://ucoedi.wikispaces.com/Practica5>