



Práctica 1: Implementación de Estructuras de Datos

1. Objetivo

El objetivo de esta práctica es tomar contacto con el entorno de desarrollo que se utilizará durante el cuatrimestre, y realizar un repaso de las estructuras de datos estudiadas en cursos anteriores.

2. Entrega

Esta práctica se realizará en dos sesiones de laboratorio en las siguientes fechas:

Sesión tutorada: 16 al 19 de febrero de 2016.

Sesión de entrega: 23 al 26 de febrero de 2016.

3. Enunciado

Implementar las siguientes estructuras de datos de forma que permitan almacenar datos del tipo `TDATO` definido al mediante una sentencia `typedef` de algún tipo básico.

```
typedef TDATO int;
```

Cada estructura de datos se implementa mediante una clase que oculta los detalles de su implementación. Esto es, la representación interna de la estructura no es visible desde fuera de la clase. La clase define, mediante métodos, las operaciones disponibles en la estructura de datos. Forma parte de la realización de la práctica determinar el conjunto de operaciones a implementar para cada estructura.

- **Vector:** un vector (llamado en inglés array) es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo.
- **Lista enlazada:** La lista enlazada es una estructura de datos que nos permite almacenar datos de una forma organizada, al igual que los vectores pero, a diferencia de estos, esta estructura es dinámica, por lo que no tenemos que saber "a priori" los elementos que puede contener. En una lista enlazada, cada elemento apunta al siguiente excepto el último que no tiene sucesor.
- **Pila:** Una pila es una estructura de datos en la que el último elemento en entrar es el primero en salir, por lo que también se denominan estructuras LIFO (Last In, First Out). En esta estructura sólo se tiene acceso a la cabeza o cima de la pila.
- **Cola:** Una cola es una estructura de datos donde el primer elemento en entrar es el primero en salir, también denominadas estructuras FIFO (First In, First Out).

Cada estructura de datos se desarrolla en un módulo de código (archivo `.cpp` y `.h`) separado. Para almacenar los datos dentro de una clase (representación interna) se utilizará memoria dinámica.

Para comprobar el correcto funcionamiento de las estructuras de datos se realizarán programas de prueba para todas las operaciones implementadas. La entrega de esta práctica consiste en la presentación de los códigos utilizados para probar el correcto funcionamiento de cada una de las operaciones implementadas en las estructuras de datos.

Durante las sesiones de laboratorio se podrán proponer modificaciones y mejoras en el enunciado de la práctica.