

# LAB11

**Objetivos:**

- Realización de ejercicios con apuntadores
- Realización de ejercicios con listas dinámicas

# Lab 11. Apuntadores y Listas dinámicas – Básico

## 11.1. Lista de ejercicios

Varios de los ejercicios que aquí se proponen se han propuesto previamente. En este laboratorio cambian el tipo de datos en el que se almacenan los datos y/o los resultados. Así, la resolución del ejercicio no radicará en los cálculos a realizar sino en la manera de acceder a los datos y/o resultados.

### 11.1.1. Está en lista dinámica

Implementa la función **esta** que, dado un valor entero N y una lista dinámica con valores que pueden no estar ordenados, diga si el entero está o no en la lista.

### 11.1.2. Posición en lista dinámica

Implementa la función **posición** que, dado un valor entero X y una lista dinámica cuyos valores pueden no estar ordenados, diga en qué posición se encuentra X en la lista. Si dicho entero estuviera repetido, bastará con devolver la posición de la primera de sus apariciones (la primera posición es 1); y en caso de no encontrarse en el vector, se devolverá el valor *Integer'Last*.

### 11.1.3. Media lista dinámica

Implementa la función **Media** que, dada una lista de enteros L, devuelva la media aritmética de los enteros de la lista. Si la lista es vacía, se devolverá *Float'Last*.

#### 11.1.4. Insertar delante en lista dinámica

Implementa la función **Prepend** que, dada una lista dinámica y un entero, modifique dicha lista anteponiendo el entero a la lista que había. Por ejemplo, dados (1 2 3 4 5) y 0, debe devolver (0 1 2 3 4 5).

#### 11.1.5. Insertar detrás en lista dinámica

Implementa la función **Append** que, dada una lista dinámica y un entero, modifique dicha lista añadiendo al final de la lista el entero. Por ejemplo, dados (1 2 3 4 5) y 6, debe devolver (1 2 3 4 5 6).

#### 11.1.6. Insertar elemento en una posición en lista dinámica

Implementa la función **Insertar\_elemento\_en\_pos** que, dada una lista dinámica, un entero, y una posición, modifique dicha lista incluyendo el entero en la lista que había. Se considera que el primer elemento es la posición 1. Todas las posiciones son válidas: si la posición de inserción es previa al primer elemento, se añade por delante de la lista; Si la posición es mayor que la posición del último se añade al final; y si es intermedia, se intercala para que el elemento añadido ocupe la posición indicada.

Por ejemplo, añadir 0 en la posición -2500 de (1 2 3 4 5), debe devolver (0 1 2 3 4 5), porque -2500 es una posición anterior a la 1. Si la fuera añadir 1000 en la posición 1000 de (1 2 3 4 5) debería devolver (1 2 3 4 5 1000). Y si hubiera que añadir 25 en la posición 3 de (10 20 30 40 50), el resultado debería ser (10 20 25 30 40 50).

**11.1.7. Insertar en lista dinámica ordenada**

Implementa el procedimiento **Insertar** que, a partir de una lista dinámica de enteros ordenada ascendentemente y un número natural N, lo inserte en la posición que le corresponde (o sea, manteniendo el orden de los elementos de la lista).

**11.1.8. Borrar de lista dinámica**

Implementa la función **Borrar** que, dada una lista y un entero N, modifica la lista eliminando de la lista la primera aparición del valor N. Si N no está en la lista, la lista se queda como está.

**11.1.9. Concatenar listas dinámicas**

Implementa la función **Concatenar** que, dadas dos listas dinámicas L1 y L2, las modifica para construir una lista dinámica que tiene los nodos de L2 a continuación de los de L1. No se puede usar **new** en este ejercicio.

**11.1.10. Invertir lista dinámica**

Implementa el procedimiento **Invertir** que, dada una lista dinámica de enteros, la modifique para que los enteros aparezcan en orden inverso. O sea, que si se le pasa la lista (1 2 3 4 5) al procedimiento, éste debe devolver (5 4 3 2 1).

