

## ESTRUCTURA DE DATOS Y ALGORITMOS - GRUPO C

### PRÁCTICA Nº4

19 de marzo 2023

#### INSTRUCCIONES

---

1. No se permitirá la consulta de ningún otro tipo de material durante la realización de la práctica.
2. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de prácticas de EDA del grupo C antes de la hora de finalización de la sesión de prácticas.
3. Aunque las prácticas se tengan que realizar en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual, la misma práctica**. En otro caso, la práctica quedará **sin evaluar** y supondrá un **0 en la calificación del estudiante que no la haya entregado**.
4. Se deben entregar **los ficheros** correspondientes a la práctica en formato **.cpp y/o .h, sin comprimir**. El nombre de los ficheros cabecera (.h) y su correspondiente .cpp debe coincidir.
5. Los ficheros entregados deben **incluir, al principio del cada fichero entregado, el nombre de los integrantes del equipo**.
6. El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su **descalificación**.

### IMPORTANTE:

- Todas las opciones deberán ser resueltos de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general** y **no particular** para unos determinados datos/valores.
- Todos los **ejercicios resueltos sin utilizar funciones** cuando sea apropiado se valorarán con una **nota máxima del 60% sobre la calificación prevista**.
- Los métodos/funciones de las clases que se indican en el enunciado de los ejercicios deben **mantener la definición de su interfaz**, es decir, **nombres de las funciones, número y tipo de argumentos/parámetros** indicados.
- Los ficheros entregados tienen que **compilar**.
- Se recomienda y será valorado de forma positiva que los nombres de las variables/objetos/clases/atributos/etc. sean **significativos** y **sustantivos**.
- Se recomienda y será valorado de forma positiva que los nombres de las funciones y los métodos sean **significativos** y **expresiones verbales**.
- Se pueden **diseñar e implementar otras funciones/métodos auxiliares o complementarias** a las solicitadas en el enunciado, siempre que tengan sentido.
- **Se pueden utilizar funciones propias de otras librerías de C++** (iostream, array, vectores, etc.), pero **NO para sustituir contenidos propios de la asignatura** (listas, pilas, colas, algoritmos de ordenación, etc.).
- Se recomienda una **primera lectura del enunciado de la práctica** para planificar su realización.
- Conviene ir **probando los diferentes recursos programados** de forma gradual. Programar todo seguido y realizar las pruebas al final cuando quedan 10 minutos para la entrega, suele acabar con **errores de compilación sin resolver en la entrega** y por lo tanto con una calificación de 0 puntos.

## EJERCICIO

---

Esta práctica debe ser desarrollada en el lenguaje C++11 y utilizando los **tipos de datos**, las **estructuras de control** y las **funciones** necesarias para cumplir con las especificaciones que se describen a continuación.

Diseñar los métodos de las clases facilitadas que permitan generar y gestionar dos árboles binarios de búsqueda (ABB) de números enteros. Uno de los árboles será creado de forma **iterativa** y se identificará por **ABBiter**, mientras que el segundo se creará de forma **recursiva** y se llamará **ABBrec**.

Las clases **NodoABB** y **TADABB** deberán utilizar las interfaces que se proporcionan en los ficheros **.h** correspondientes. La implementación de los métodos **constructores**, **gets** y **sets** puede estar realizada en el fichero **.h** de cada clase. El resto de métodos de ambas clases podrán estar implementados en su fichero **.h** o **.cpp**. **(1,5 puntos)**

El fichero **main.cpp** incluye la estructura general del programa que permitirá al usuario realizar pruebas de los procesos desarrollados:

1. Crear el árbol **ABBiter** (**crearABBiter**) de manera **iterativa** **(2 puntos)**:

Esta opción consistirá en solicitar al usuario **números enteros positivos**, uno a uno, y añadirlos en el árbol **ABBiter** en su lugar correspondiente. El proceso finalizará cuando el usuario introduzca un **valor negativo**.

2. Crear el árbol **ABBrec** (**crearABBrec**) de manera **recursiva** **(2 puntos)**:

Esta opción consistirá en solicitar al usuario **números enteros positivos**, uno a uno, y añadirlos en el árbol **ABBrec** en su lugar correspondiente. El proceso finalizará cuando el usuario introduzca un **valor negativo**.

3. Mostrar el contenido de ambos árboles ABB, **ABBiter** y **ABBrec** **(1,5 puntos)**:

Esta opción, **mostrarABB**, consistirá en mostrar por pantalla los números guardados en los dos árboles ABB, tanto **ABBiter** como **ABBrec**, de forma que aparezcan en pantalla ordenados de **menor a mayor**.

Si alguno de los árboles está **vacío**, se deberá de mostrar un mensaje para indicárselo al usuario.

4. Determinar el número de veces que un determinado valor se encuentra en el árbol **ABBrec** (1,5 puntos):

Esta opción, **contarValorABBrec**, consistirá en buscar un determinado número en el árbol **ABBrec** y contar cuántas veces está incluido en el árbol. Finalmente, desde el **main**, se mostrará el resultado a través de un mensaje por pantalla que deberá incluir el valor buscado y el número de veces que se localizó. Para realizar esta operación se solicitará al usuario el número que se quiere buscar.

Si el árbol **ABBrec** está **vacío**, se deberá de mostrar un mensaje para indicárselo al usuario.

5. Determinar la suma de los números tiene el árbol **ABBiter** (1,5 puntos):

Esta opción, **sumarABBiter**, consistirá en determinar la suma de los números almacenados en los nodos que forman el árbol **ABBiter**. Finalmente, desde el **main**, se mostrará el resultado a través de un mensaje por pantalla.

Si el **ABBiter** está **vacío**, se deberá de mostrar un mensaje para indicárselo al usuario.

Si el árbol **ABBiter** está **vacío**, se deberá de mostrar un mensaje para indicárselo al usuario.