

### **PARTE 3: Ejercicio de programación JAVA**

**Duración 60 minutos**

Este ejercicio comprende **3** partes:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo de los casos de prueba ("test case", al menos uno) para verificar la corrección de la funcionalidad implementada.
3. Desarrollo del programa ejecutable con datos de ejemplo provistos y entrega del archivo "**salidaNivelDeHojas.txt**" en la tarea correspondiente de la webasignatura.

#### **ESCENARIO**

Los árboles binarios de búsqueda son usados para organizar un conjunto de elementos con el objetivo de que las búsquedas sobre él tengan un orden del tiempo de ejecución logarítmico en el mejor de los casos. Este mejor caso corresponde a que el árbol resulte armado completo, con todas sus hojas al mismo nivel. Desafortunadamente, dependiendo del orden en el que se inserten las claves, el árbol podría degenerar en una lista y el orden del tiempo de ejecución de las búsquedas será lineal.

Por ese motivo, el director de producción ha encargado desarrollar algunas métricas que permitan extraer alguna conclusión sobre el comportamiento de las búsquedas sobre un árbol binario de búsqueda de acuerdo a cómo haya quedado armado. Entre ellas, una posible es la determinación del nivel de las hojas, que en el mejor de los casos será el mismo para todas, y en el peor de los casos será "n" para la única hoja del árbol.

#### **PARTE 1: Funcionalidad a desarrollar:**

Descargar y abrir el proyecto Netbeans desde la webasignatura (Archivo **PARCIAL2-PARTE3**).

La clase principal se denomina "Parcial2", y tiene su correspondiente método "main".

Se debe desarrollar un algoritmo (y calcular el orden del tiempo de ejecución) que devuelva una lista implementada sobre una string con los elementos separados por un guión, con el nivel de cada hoja de un árbol binario de búsqueda, implementando las interfaces

**de IArbolBB :**

**public String nivelDeHojas ()**

**de IElementoAB:**

**public String nivelDeHojas(int nivel)**

#### **PARTE 2: TEST CASES.**

Implementa los **Casos de Prueba** necesarios para verificar el correcto funcionamiento del método.

#### **PARTE 3: PROGRAMA.**

En el método "**main**" de la clase "**Parcial2**", implementar lo necesario para:

- 1- Crear una instancia de Árbol Binario de Búsqueda
- 2- Insertar en la misma los elementos cuyas claves se han de leer desde el archivo "**claves.txt**".
- 3- Invocando a los métodos desarrollados en la PARTE 1, calcular la cantidad de nodos internos completos del árbol. Registrar este valor en el archivo de texto "**salidaNivelDeHojas.txt**".

**ENTREGA:** Debes entregar **TODO** el proyecto Netbeans, más el archivo de salida "**salidaCantNodosCompleto.txt**", en un archivo comprimido "**Parcial2.zip**" en la tarea "**PARCIAL2-PARTE3**" publicada en la webasignatura, hasta la hora indicada.