

## TAREA DE ESTRUCTURAS DE DATOS

### ÁRBOLES DE BÚSQUEDA BINARIA

**1. Enunciado:** Diseñe un programa en JAVA, que permita el almacenamiento de números extraídos desde una matriz **A** en un árbol binario de búsqueda. Deberá cumplirse los siguientes requerimientos:

- Deberá existir un método que permita el ingreso de números en una matriz de tamaño **n x n** desde teclado (**solamente bajo la diagonal principal**).
- Los números extraídos desde la matriz deberán almacenarse según su valor en el árbol binario.
- Deberá almacenarse además en cada nodo el número de veces que se repite cada número.
- Permitir conocer en orden, los números ingresados y cuantas veces se repiten.

**2. Enunciado:** Realice un programa en Java, que permita almacenar en ABB los siguientes datos de Empleados:

- Nombre** // leído desde teclado
- Id** // leído desde teclado
- Categoría (A, B o C)** // leído desde teclado
- Antigüedad** // leído desde teclado
- Sueldo total** // Calculado de acuerdo a los siguientes parámetros
  - Categoría A: Bonificaciones: de \$800 por Coordinación y \$500 por responsabilidad más \$30 por cada año de antigüedad a partir del 5to año
  - Categoría B: Bonificaciones: de \$400 por Coordinación y de \$300 por responsabilidad más \$30 por cada año de antigüedad a partir del 5to año
  - Categoría C: Bonificaciones: de \$300 por responsabilidad más \$25 por cada año de antigüedad a partir del 5to año

Sueldo básico \$450

Descuentos: Si el total es mayor a \$2100, se aplicará el 8% de Impuesto a la Renta, caso contrario 5%

El programa deberá almacenar los datos de los empleados en árboles binarios de acuerdo a su categoría (**A, B y C**) y ordenados por su **Id**.

Se requiere la presentación ordenada (de acuerdo a la categoría) de: **Id, nombre y sueldo total a recibir**.

**3. Enunciado:** Realice un programa en Java, que permita almacenar en ABB los siguientes datos de Clientes:

- a. **Nombre** // leído desde teclado
- b. **Id** // leído desde teclado
- c. **Antigüedad** // leído desde teclado
- d. **Historial crediticio (A, B, C)** // leído desde teclado
- e. **Categoría** // Calculado de acuerdo a los siguientes parámetros
  - i. Monto de compra > \$1000 → 3 puntos; entre 500 y 1000 → 2 puntos; menor a 500 → 0 puntos
  - ii. Historial crediticio: A → 3 puntos; B → 2 puntos y C → 1 punto.
  - iii. Antigüedad: > a 5 años → 3 puntos; entre 2 y 5 años → 2 puntos y menor a 2 → 1 punto

Categoría A: 7 puntos o más

Categoría B: entre 3 y 7 puntos

Categoría C: menor a 3 puntos

Descuentos: Categorías (A 10%, B 6% y C 2%)

El programa deberá almacenar los datos de los Clientes en árboles binarios de acuerdo a su categoría (**A, B y C**) y ordenados por su **Id**.

Se requiere la presentación ordenada (de acuerdo a la categoría) de: **Id, nombre y descuento a aplicar**.