



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN BIOINFORMÁTICA

Algoritmos y Estructuras de Datos

Laboratorio 4 - Unidad I
Árboles Binarios de Búsqueda.

1. Realice un programa que:

- a) Cree un árbol binario de búsqueda que contenga solo número enteros.
- b) Verifique que los números NO se repitan. Por lo tanto contendrá elementos únicos.
- c) Realice las operaciones básicas sobre la estructura:
 - 1) Insertar número.
 - 2) Eliminar número buscado.
 - 3) Modificar un elemento buscado (eliminar valor viejo e insertar el nuevo valor).
 - 4) Mostrar el contenido del árbol en Preorden, Inorden y Posorden.
 - 5) Generar el grafo correspondiente de la estructura creada mediante la herramienta **Graphviz** (<http://www.graphviz.org/>). Pruebe su instalación con el comando: `$ sudo apt install graphviz`.
- d) Recuerde que éste tipo de árbol tiene la característica de mantener sus elementos ordenados.
- e) Controle las acciones con un menú básico que se muestre hasta decidir terminar el programa.
- f) Realice las validaciones correspondientes de manera que el programa opere correctamente.
- g) El programa NO recibirá parámetros de entrada.
- h) Para ejecutar los comandos que generan el grafo y lo visualiza, pueden utilizar la función **system()**.
- i) Ejemplo. Para el ingreso de los siguientes valores: 120, 87, 140, 43, 99, 130, 22, 65, 93, 135, 56. Los resultados de los diferentes recorridos serían:

— Preorden—
120 - 87 - 43 - 22 - 65 - 56 - 99 - 93 - 140 - 130 - 135

— Inorden—
22 - 43 - 56 - 65 - 87 - 93 - 99 - 120 - 130 - 135 - 140

— Posorden—
22 - 56 - 65 - 43 - 93 - 99 - 87 - 135 - 130 - 140 - 120
- j) Para generar el grafo: **dot -Tpng -ografo.png grafo.txt**
- k) Para visualizarlo se puede invocar a algún programa como: **eog grafo.png**
- l) El código que genera el grafo es el siguiente (grafo.txt):

```

digraph G {
node [style=filled fillcolor=yellow];
120->87;
120->140;
87->43;
87->99;
43->22;
43->65;
"22i" [shape=point];
22->"22i";
"22d" [shape=point];
22->"22d";
65->56;
"65d" [shape=point];
65->"65d";
"56i" [shape=point];
56->"56i";
"56d" [shape=point];
56->"56d";
99->93;
"99d" [shape=point];
99->"99d";
"93i" [shape=point];
93->"93i";
"93d" [shape=point];
93->"93d";
140->130;
"140d" [shape=point];
140->"140d";
"130i" [shape=point];
130->"130i";
130->135;
"135i" [shape=point];
135->"135i";
"135d" [shape=point];
135->"135d";
}

```

