

## Laboratorio 9: Árboles BST

M. Sc. Ricardo Román-Brenes - `ricardo.roman@ucr.ac.cr`

II-2019

---

### Tabla de contenidos

1. Enunciado	1
2. Consideraciones	2

---

### 1. Enunciado

Implementar la estructura de datos *BinarySearchTree*. Implemente una clase emplantillada que reciba el tipo de dato a almacenar.

Implemente los siguientes métodos:

- |  |  |
|--|--|
| ▪ <code>Node&amp; insert(Data &amp;d)</code>             | ▪ <code>Node&amp; NodeOf(Data &amp;d)</code> |
| ▪ <code>void remove(Data &amp;d)</code>                  | ▪ <code>void preOrden()</code>               |
| ▪ <code>Node&amp; largestToTheLeft(Node &amp;n)</code>   | ▪ <code>void inOrden()</code>                |
| ▪ <code>Node&amp; smallestToTheRight(Node &amp;n)</code> | ▪ <code>void postOrden()</code>              |
| ▪ <code>Data&amp; dataIn(Node &amp;n)</code>             |  |

Debe además implementar una clase `Node`, emplantillada que contenga datos, punteros a los hijos izquierdo y derecho. Al igual que con los estructuras de lista, NO UTILICE tipos primitivos como datos.

Haga un programa de prueba para los ítemes anteriores.

## 2. Consideraciones

- Haga grupos de hasta 3 personas.
- Genere un reporte en  $\text{\LaTeX}$  con sus conclusiones y el código fuente como anexos.
- Suba su código, aproximación a la solución y documentación (doxygen, README, INSTALL, etc) al git respectivo de su grupo y el directorio del laboratorio.
- Cada estudiante debe subir el reporte a Schoology.
- Recuerde que por cada día tardío de entrega se le rebajaran puntos de acuerdo con la formula:  $4^d$ , donde  $d > 1$  es la cantidad de días tardíos.