



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador ArbolRojinegro<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas
2014



MANUAL DE USUARIO: Simulador “ArbolRojiNegro<T>”

Descripción General

El presente Manual de usuario pretende describir en detalle el conjunto de funcionalidades de la Aplicación desarrollada para la Simulación del comportamiento de la Estructura de Datos ArbolRojiNegro<T>.

Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Árbol RojiNegro: Insertar, Eliminar y Buscar un dato dentro de cada una de las ramas del Árbol. Adicionalmente el estudiante podrá conocer algunas de las propiedades del Árbol como: Altura, peso y cantidad de Hojas presentes (ilustradas gráficamente).

Por último se implementa un conjunto de animaciones que pretenden ilustrar las diversas formas en que pueden ser recorridos los Arboles Binarios de Búsqueda como el RojiNegro, de manera que sea agradable para el estudiante la forma de comprender cada recorrido: preOrden, inOrden, postOrden y por Niveles.

Para la implementación de este Simulador se ha determinado la siguiente distribución de paquetes, ya conocida por el Estudiante, de forma que sea fácilmente apropiable a futuras modificaciones con el fin de hacer buen uso de esta aplicación.



“Directorio del Simulador para ArbolRojiNegro<T>”

La implementación de este Simulador se desarrolla basada en la implementación del Simulador de Árbol Binario de Búsqueda y las funciones (operaciones) desarrolladas para dicho Simulador. La diferencia de este

simulador radica en el conjunto de propiedades a la hora de insertar y eliminar los datos en la estructura que lo hacen diferente a la estructura Árbol Binario de Búsqueda.

A continuación se presenta la interface principal del simulador para “Árbol RojiNegro”.



“Interface principal del Simulador para ArbolRojiNegro<T>”

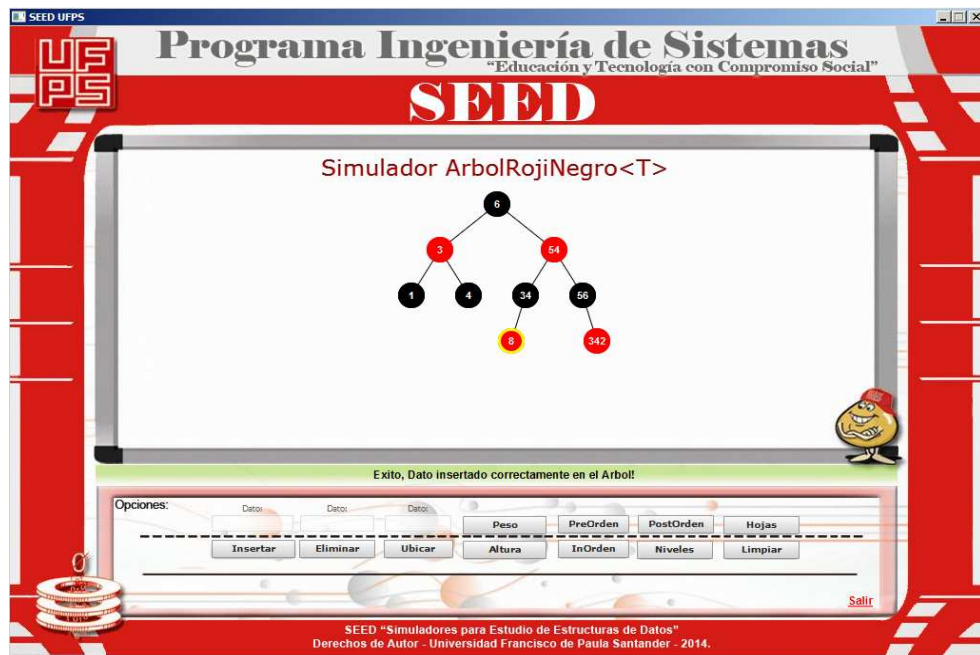
Descripción de las Funcionalidades del Simulador

El simulador para ArbolRojiNegro<T> permite al Estudiante:

1. Insertar Datos:

Para Insertar datos el estudiante simplemente deberá ingresar a la caja de texto el dato que desea insertar en el Árbol, este dato no puede ser menor a **-99** ni mayor a **999**, rango seleccionado por cuestiones de que no se desborde el número del nodo gráfico, además el árbol no puede superar la **altura 5**, con el fin de que la ilustración de la estructura no se desborde del Área de Dibujo del mediador pedagógico digital.

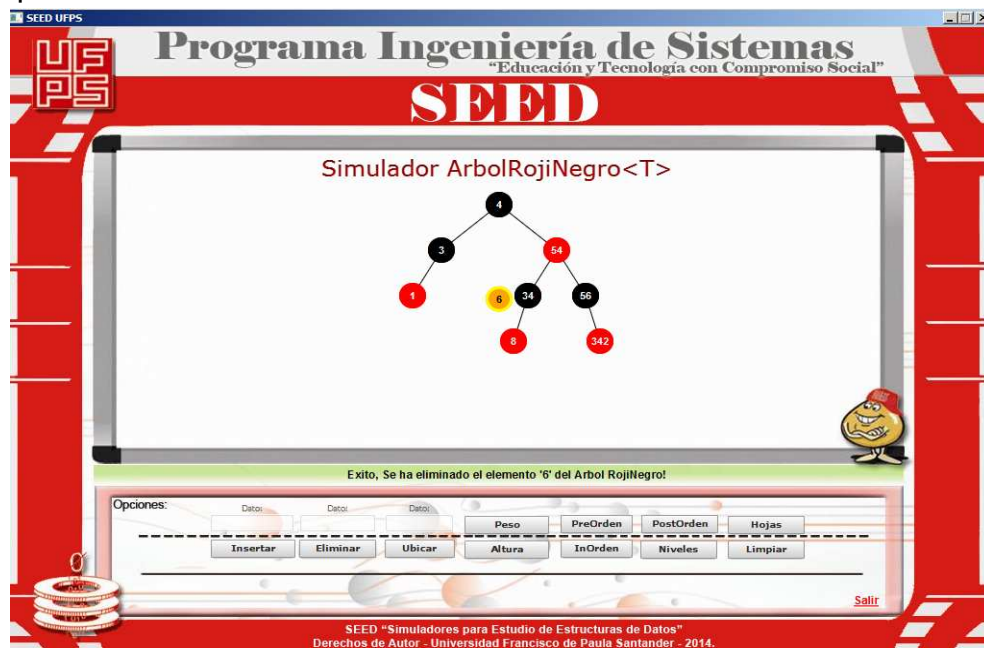
El Árbol RojiNegro, debido a sus propiedades de equilibrio por altura negra, realizara las operaciones de balanceo por color necesarias y el Árbol será pintado según como queda después de la inserción del nuevo dato.



"Inserción en el Árbol RojiNegro de los datos: 6 – 3 – 54 – 1 – 4 – 34 – 56 – 342 y 8".

2. Eliminar Datos:

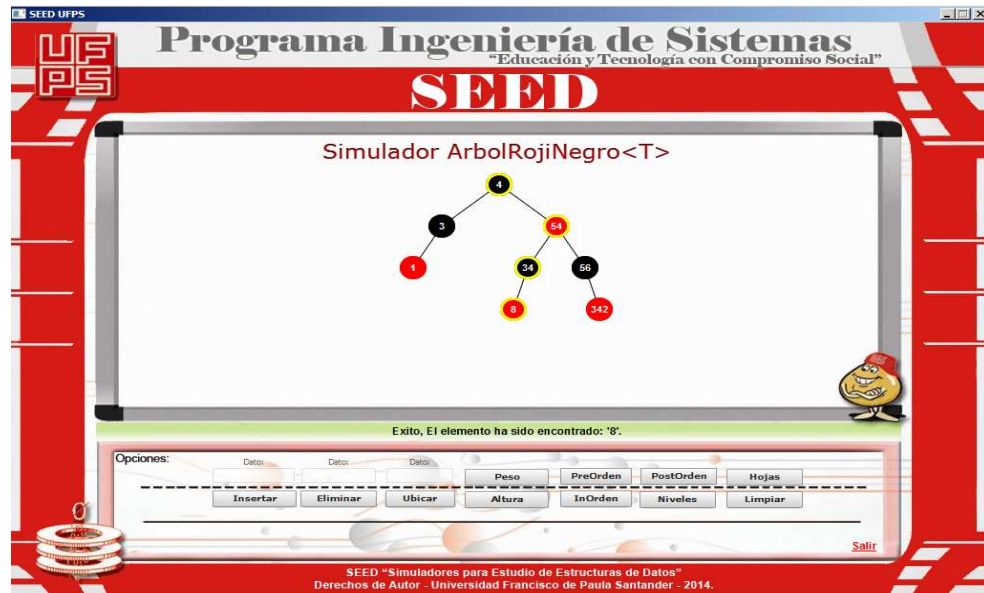
Para eliminar datos el estudiante deberá ingresar simplemente el dato que desea eliminar del Árbol RojiNegro (previamente insertado) en la caja de texto. Una vez eliminado el dato del Árbol RojiNegro este no será pintado e inmediatamente el Árbol será balanceado por color y pintado correctamente en pantalla del Simulador.



"Eliminación del dato: 6 del Árbol RojiNegro. 6 es bajado del Árbol y desaparece".

3. Búsqueda de un dato (Ubicar):

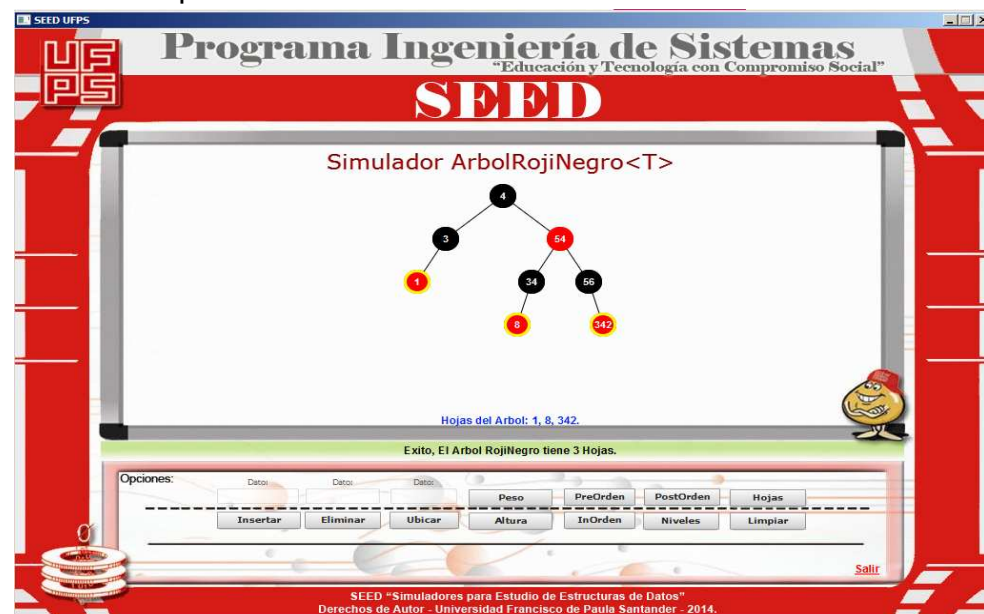
Para ubicar un dato dentro del Árbol RojiNegro el estudiante deberá ingresar en la caja de texto el dato que desea ubicar. La búsqueda del dato se realiza de acuerdo a las propiedades del Árbol y el costo algorítmico que cada uno genera, para esta estructura se indica el camino de la búsqueda del dato.



"Búsqueda del dato '8' dentro del Árbol RojiNegro"

4. Obtener las Hojas del Árbol

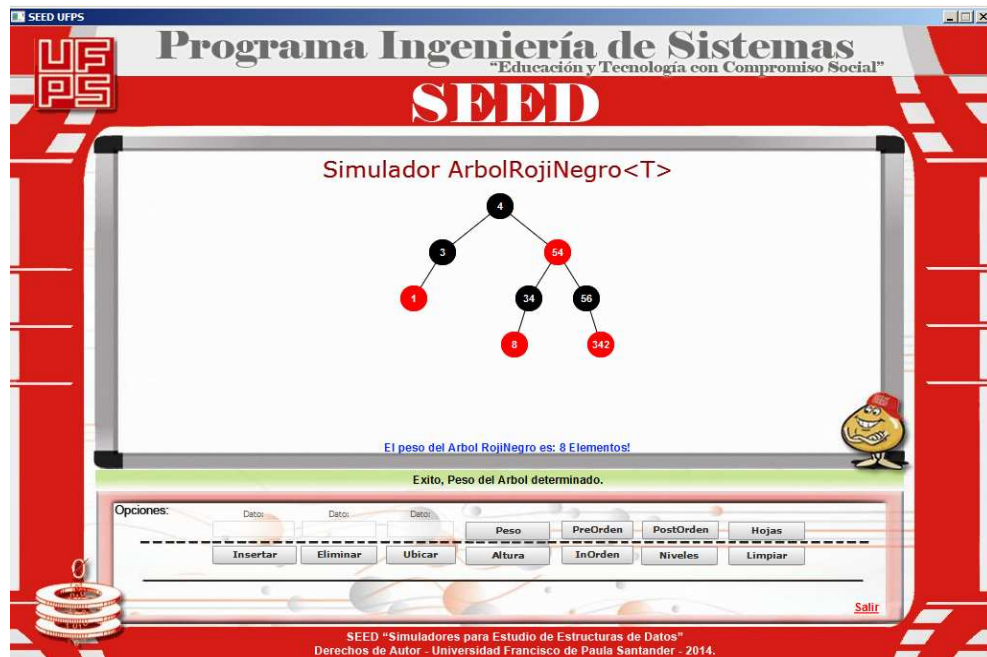
Para obtener los elementos Hoja del Árbol RojiNegro el estudiante deberá dar **clic** en el botón **Hojas**, y estas serán pintadas en el Árbol para que el estudiante las pueda identificar.



"Determinación de los elementos hoja del Árbol RojiNegro"

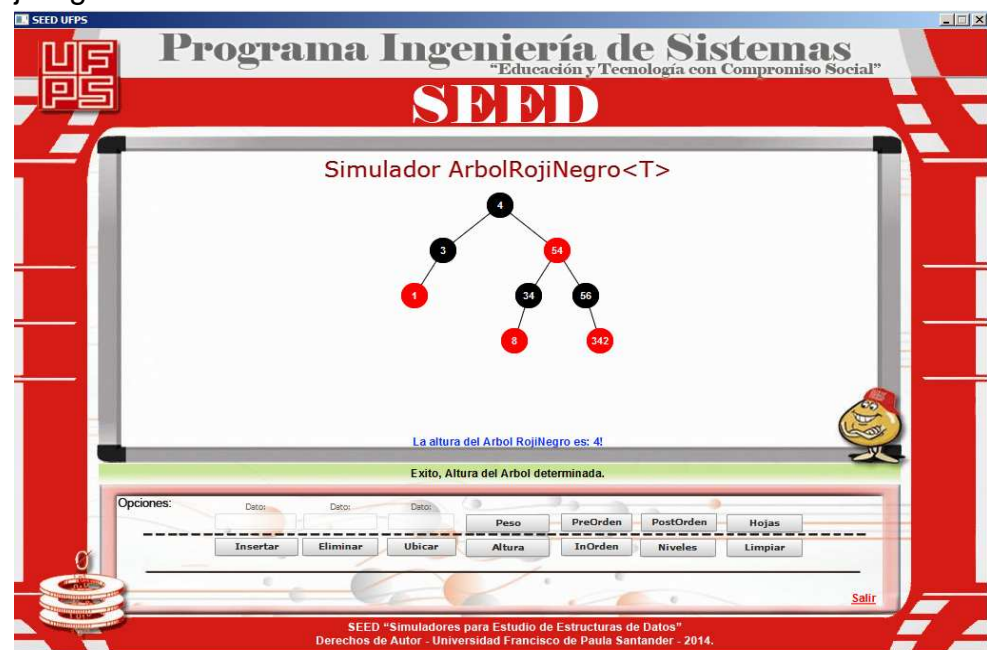
5. Determinar el Peso y Altura del Árbol RojiNegro

Para determinar el Peso del Árbol el estudiante deberá dar clic en el botón **Peso**, inmediatamente la aplicación indicará la cantidad de datos presentes en el Árbol RojiNegro en un momento determinado.



"Determinar el peso del Árbol RojiNegro: 8 Elementos"

Ahora para determinar la Altura del Árbol el estudiante deberá dar clic en el el botón **Altura**, inmediatamente la aplicación indicará la altura del Árbol RojiNegro en un momento determinado.

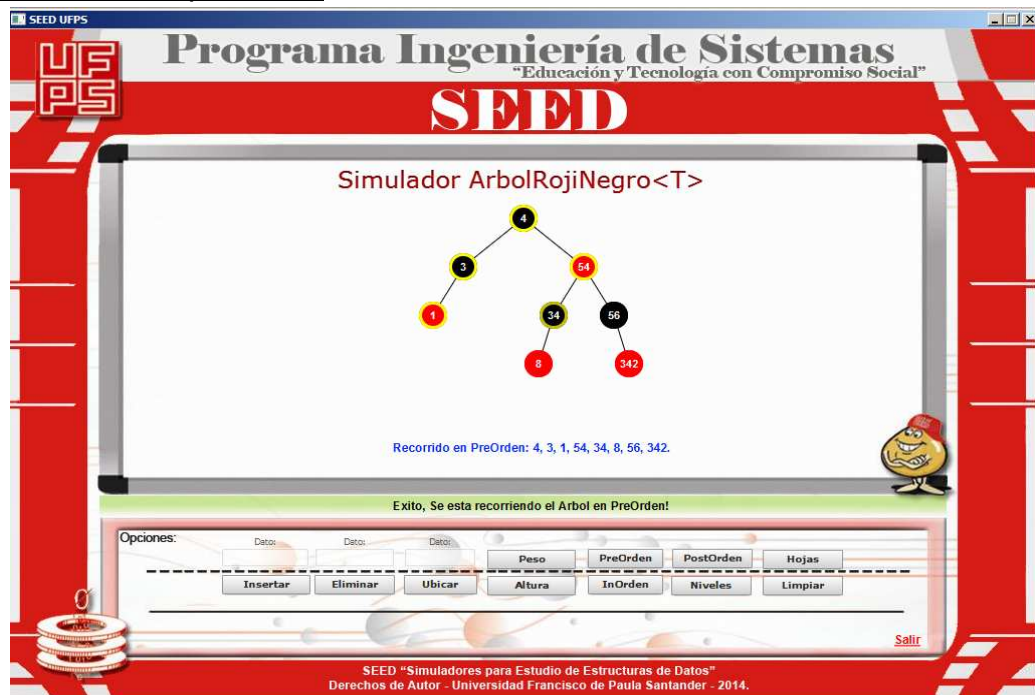


"Determinar la altura del Árbol RojiNegro: Altura 4"

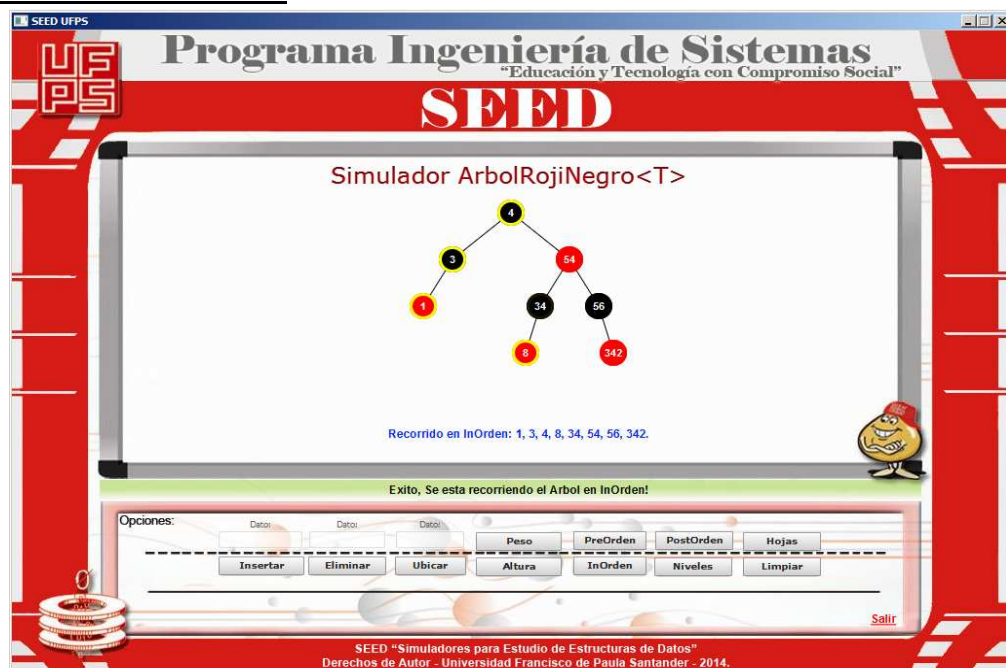
6. Recorridos en el Árbol RojiNegro

Para recorrer el Árbol RojiNegro el estudiante deberá seleccionar el recorrido que desea realizar en el Árbol e inmediatamente el simulador realizara una animación. Debe dar clic en el botón: **PreOrden**, **InOrden**, **PostOrden** y por **Niveles**.

Recorrido en preOrden



Recorrido en inOrden



Recorrido en postOrden

SEED UFPS

Programa Ingeniería de Sistemas
"Educación y Tecnología con Compromiso Social"

SEED

Simulador ArbolRojiNegro<T>

```
graph TD; 4((4)) --- 3((3)); 4 --- 54((54)); 3 --- 1((1)); 54 --- 34((34)); 54 --- 56((56)); 34 --- 8((8)); 56 --- 342((342));
```

Recorrido en PostOrden: 1, 3, 8, 34, 342, 56, 54, 4.

Exito, Se esta recorriendo el Arbol en PostOrden!

Opciones:

Datos:	Datos:	Datos:	Peso	PreOrden	PostOrden	Hojas
Insertar	Eliminar	Ubicar	Altura	InOrden	Niveles	Limpiar

Salir

SEED "Simuladores para Estudio de Estructuras de Datos"
Derechos de Autor - Universidad Francisco de Paula Santander - 2014.

Recorrido por Niveles

SEED UFPS

Programa Ingeniería de Sistemas
"Educación y Tecnología con Compromiso Social"

SEED

Simulador ArbolRojiNegro<T>

```
graph TD; 4((4)) --- 3((3)); 4 --- 54((54)); 3 --- 1((1)); 54 --- 34((34)); 54 --- 56((56)); 34 --- 8((8)); 56 --- 342((342));
```

Recorrido por Niveles: 4, 3, 54, 1, 34, 56, 8, 342.

Exito, Se esta recorriendo el Arbol por Niveles!

Opciones:

Datos:	Datos:	Datos:	Peso	PreOrden	PostOrden	Hojas
Insertar	Eliminar	Ubicar	Altura	InOrden	Niveles	Limpiar

Salir

SEED "Simuladores para Estudio de Estructuras de Datos"
Derechos de Autor - Universidad Francisco de Paula Santander - 2014.

7. Limpiar el Árbol RojiNegro:

Para limpiar el Árbol RojiNegro, el estudiante deberá dar clic en el botón “Limpiar”, esta acción elimina todos los datos del Árbol RojiNegro dejándolo vacío para que el estudiante comience a ilustrar de nuevo las funciones básicas de la estructura.



“Se limpió el Árbol RojiNegro”

8. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Árbol RojiNegro, existe la posibilidad de que el estudiante pueda **“adicionar nuevas funcionalidades”** a la aplicación, de acuerdo a las actividades asignadas por los docentes o el interés propio de generar nuevos algoritmos en cada estructura y poder simularlos gracias a la herramienta gráfica del Simulador.

A continuación se presentan los pasos que deberá seguir el estudiante para crear una nueva funcionalidad dentro del Simulador de ArbolRojinegro:

- 8.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **ArbolRojiNegro**, presente en el paquete **SEED_Colecciones** y que desea adicionar a la funcionalidad del Simulador. (Para el ejemplo, se creará un algoritmo que permita obtener el código **Łukasiewicz** del Árbol RojiNegro:

```

/**
 * Metodo que retorna el codigo Lukasiewicz del arbol binario; Este codigo etiqueta los nodos internos con "a"
 * y los externos con una "b" y realiza el recorrido en preorden con estas convenciones. <br>
 * <b>post: </b> Se retorno el codigo Lukasiewicz del Arbol Binario.<br>
 * @return un String con el codigo Lukasiewicz del arbol binario
 */
public String Luca()
{
    return(Luca(this.raiz));
}

/**
 * Metodo que retorna el codigo Lukasiewicz del arbol binario; Este codigo etiqueta los nodos internos con "a"
 * y los externos con una "b" y realiza el recorrido en preorden con estas convenciones. <br>
 * <b>post: </b> Se retorno el codigo Lukasiewicz del arbol binario.<br>
 * @param r representa la raiz del arbol, o raiz de subarbol
 * @return un String con el codigo Lukasiewicz del arbol binario
 */
private String Luca(NodoBin<T> r){
    if(r==null)
        return("b");
    return("a"+Luca(r.getIzq())+Luca(r.getDer()));
}

```

8.2. A continuación el Estudiante debe generar un Método en la clase **SimuladorArbolRojiNegro** del paquete **Mundo_ArbolRojiNegro**, que realice el llamado al Método con el nuevo algoritmo creado en la Estructura de Datos **ArbolRojiNegro**. Para el llamado debe utilizar el objeto creado en el Mundo **miArbolRN**.

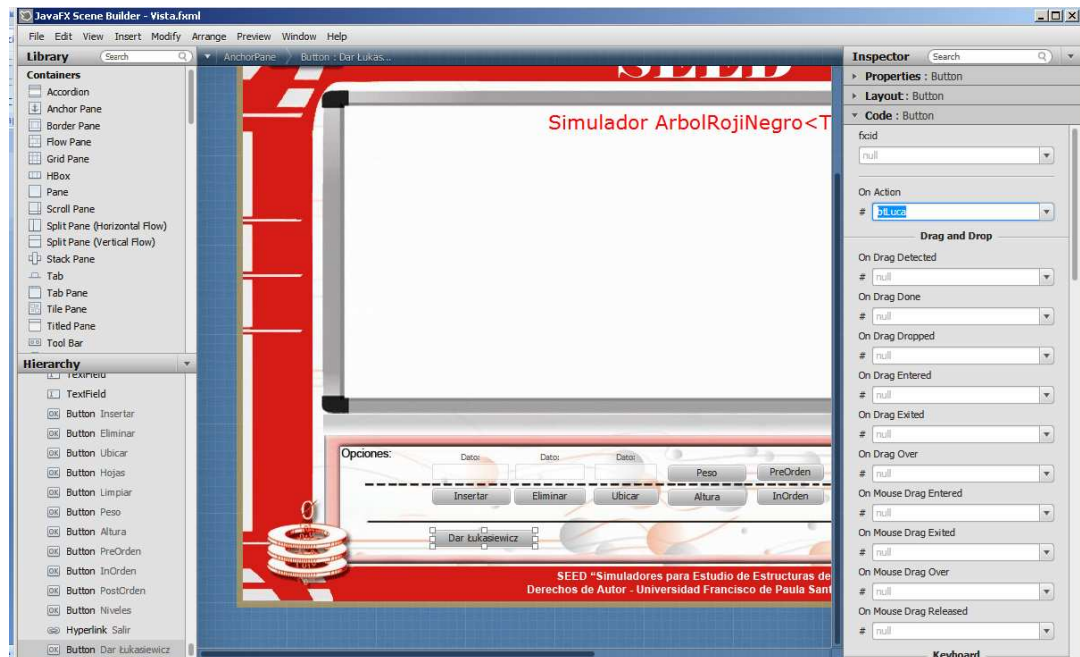
```

public String getLuca(){
    return " Lukasiewicz: "+ this.miArbolRN.Luca();
}

```

8.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **SimuladorArbolRojiNegro**. Existe dos posibilidades para ello: Utilizar la herramienta “**JavaFX SceneBuilder**” para insertarlo, o agregar el código del Button en el Archivo **Vista.fxml**.

8.3.1. Utilizando JavaFX SceneBuilder



8.3.2. Insertando directamente el elemento en Vista.fxml

```
<Button layoutX="164.0" layoutY="610.0" mnemonicParsing="false" onAction="#btLuca" prefWidth="135.0" text="Dar Lukasiewicz" />
```

Es importante resaltar, para ambos casos, que se debe asignar el evento **“OnAction”** del Button, para el ejemplo **“darLuca”**, el cual será el nombre del **Método** dentro de la clase **Controlador** que permite realizar la nueva funcionalidad del Simulador.

A continuación el Método dentro del paquete **simarbolroji negro** en la clase **Simulador** que permite realizar el llamado a la nueva funcionalidad.

```
@FXML
private void btLuca() {
    this.pintarTDA();
    msg.setText(this.simulador.getLuca());
    msg.setVisible(true);
    this.impNota("Codigo Lukasiewicz calculado!",0);
}
```

Además del llamado al Método creado en **SimuladorArbolRojiNegro**, el estudiante deberá invocar el método que le permita volver pintar el Árbol, el cual siempre será **“pintarTDA()”**. Opcionalmente se recomienda enviar una mensaje con la respuesta a la operación realizada utilizando **“impNota(“**

Mensaje a enviar “ , tipo)” donde tipo es cero **(0)** si en un mensaje Exitoso y uno **(1)** en caso de ser un mensaje erróneo.

A continuación se comprueba el funcionamiento del Algoritmo realizado:



"Código Luca del Árbol RojiNegro"