



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador ArbolBinario<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas

2014



MANUAL DE USUARIO: Simulador “ArbolBinario<T>”

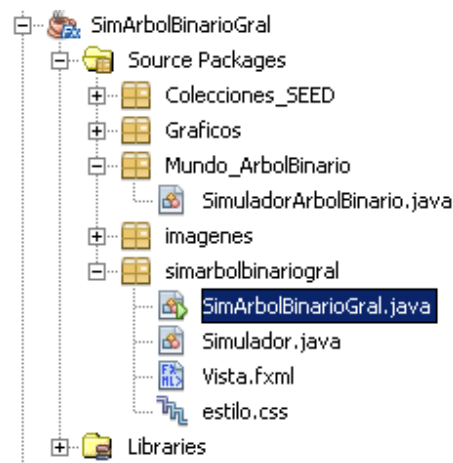
Descripción General

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 ArbolBinario<T>.

Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Árbol Binario: 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 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 ArbolBinario<T>”

De la implementación de este simulador se desprende un conjunto de Simuladores que se realizan en base al conjunto de funcionalidades desarrolladas para el presente Simulador, por lo que estructuras de datos como: Árbol AVL, Árbol RojiNegro, Árbol Splay y Árbol Binario de Búsqueda

son diferentes solo en un conjunto de propiedades a la hora de insertar y eliminar los datos, mas no en la estructura heredada del Árbol Binario General.

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



“Interface principal del Simulador para ArbolBinarioGral<T>”

Descripción de las Funcionalidades del Simulador

El simulador para ArbolBinario<T> permite al Estudiante:

1. Insertar Datos:

Para Insertar datos el estudiante primeramente deberá ingresar el dato que desea insertar en el Árbol en la caja de Texto indicada, ingresando también el padre del dato que desea insertar y seleccionando en el combo en cuál de los lados (hijos) desea insertar el nuevo elemento para que sea pintado en pantalla: a la izquierda del padre o a la derecha del padre. Para insertar la raíz del árbol solo deberá ingresar el dato que desea ubicar en la raíz del Árbol.



"Inserción en el Árbol Binario de los datos: 23 - 12 - 9 - 30 - 79 - 41"

2. Eliminar Datos:

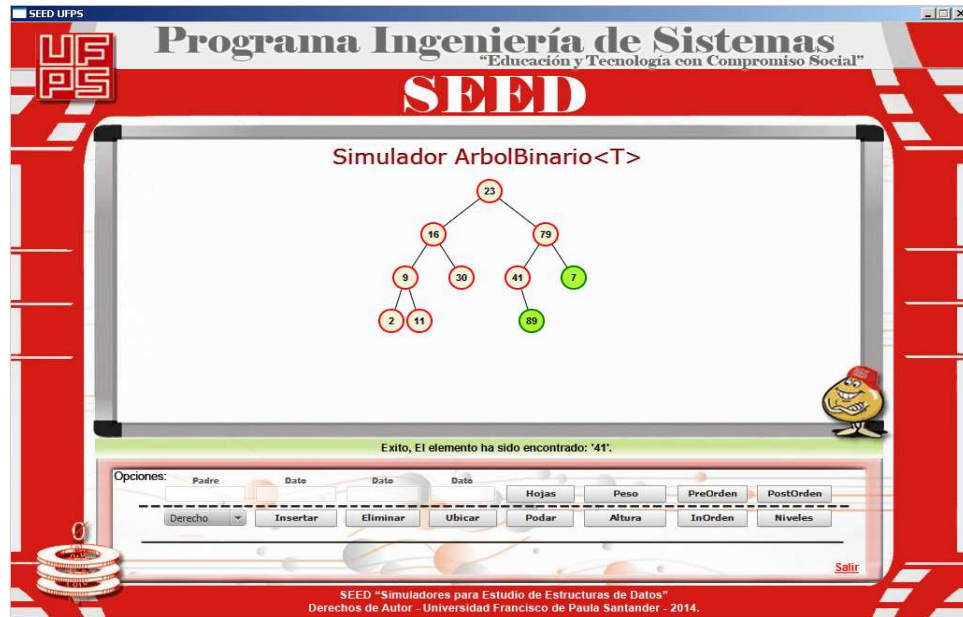
Para eliminar datos el estudiante deberá ingresar simplemente el dato que desea eliminar del Árbol Binario (previamente insertado) e inmediatamente el Árbol será pintado sin el dato que ha sido eliminado del mismo.



"Eliminación del dato: 12 del Árbol Binario. 12 es bajado del Árbol y desaparece"

3. Búsqueda de un dato (Ubicar):

Para ubicar un dato dentro del Árbol Binario 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.



"Búsqueda del dato '41' dentro del Árbol Binario"

4. Obtener las Hojas del Árbol

Para obtener los elementos Hoja del Árbol Binario 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 Binario"

5. Podar el Árbol Binario

El estudiante podrá podar las hojas de un Árbol Binario dar clic en el botón **podar** de la aplicación. Los elementos Hoja del árbol serán eliminados de forma que las hojas del Árbol Binario serán ahora los padres de estos elementos.



"Podar el Árbol Binario anterior. Las hojas han sido eliminadas"

6. Determinar el Peso y Altura del Árbol Binario

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 Binario en un momento determinado.



"Determinar el peso del Árbol Binario: 12 Elementos"

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

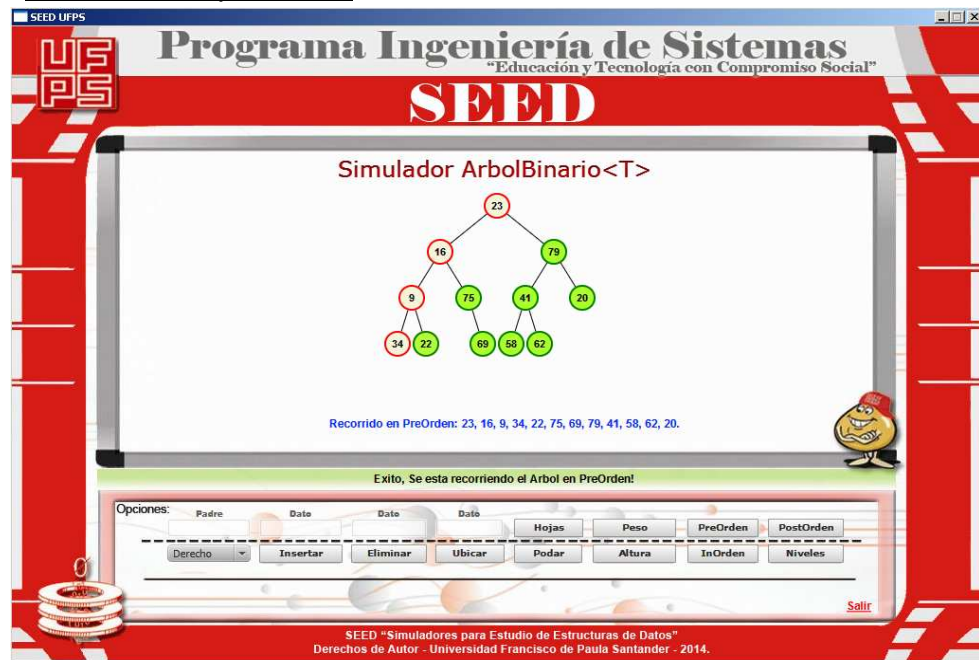


"Determinar la altura del Árbol Binario: Altura 4"

7. Recorridos en el Árbol Binario

Para recorrer el Árbol Binario 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 ArbolBinario<T>

Recorrido en PostOrden: 34, 22, 9, 68, 75, 16, 58, 62, 41, 20, 79, 23.

Exito, Se esta recorriendo el Arbol en PostOrden!

Opciones: Padre Date Date Date Hojas Peso PreOrden PostOrden
Derecho Insertar Eliminar Ubicar Podar Altura InOrden Niveles

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 ArbolBinario<T>

Recorrido por Niveles: 23, 16, 79, 9, 75, 41, 20, 34, 22, 68, 58, 62.

Exito, Se esta recorriendo el Arbol por Niveles!

Opciones: Padre Date Date Date Hojas Peso PreOrden PostOrden
Derecho Insertar Eliminar Ubicar Podar Altura InOrden Niveles

Salir

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

8. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Árbol Binario, 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 grafica del Simulador.

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

- 8.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **ArbolBinario**, 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 **Lukasiewicz** del Árbol Binario:

```
/**
 * 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 representeta 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 **SimuladorArbolBinario** del paquete **Mundo_ArbolBinario**, que realice el **llamado** al Método con el nuevo algoritmo creado en la Estructura de Datos **ArbolBinario**. Para el llamado debe utilizar el objeto creado en el Mundo **miArbolBin**.

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

8.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **SimuladorArbolBinario**. 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="#darLuca" 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 **simarbolbinariogralen** 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 **SimuladorArbolBinario**, 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 Binario"