



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador ArbolBinarioBusqueda<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas
2014



MANUAL DE USUARIO: Simulador “ArbolBinarioBusqueda<T>”

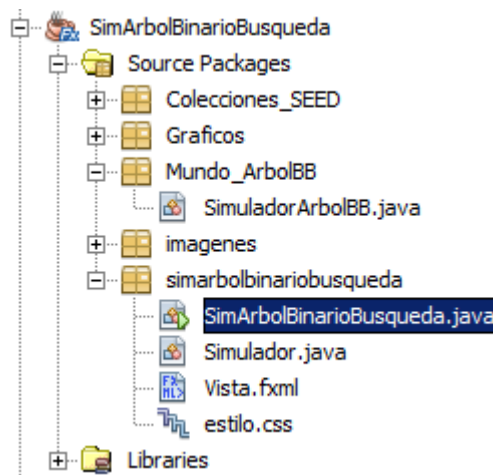
Descripción General

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 ArbolBinarioBusqueda<T>.

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

La implementación de este Simulador se desarrolla basada en la implementación del Simulador de Árbol Binario 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 General.

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



“Interface principal del Simulador para ArbolBinarioBusqueda<T>”

Descripción de las Funcionalidades del Simulador

El simulador para ArbolBinarioBusqueda<T> permite al Estudiante:

1. Insertar Datos:

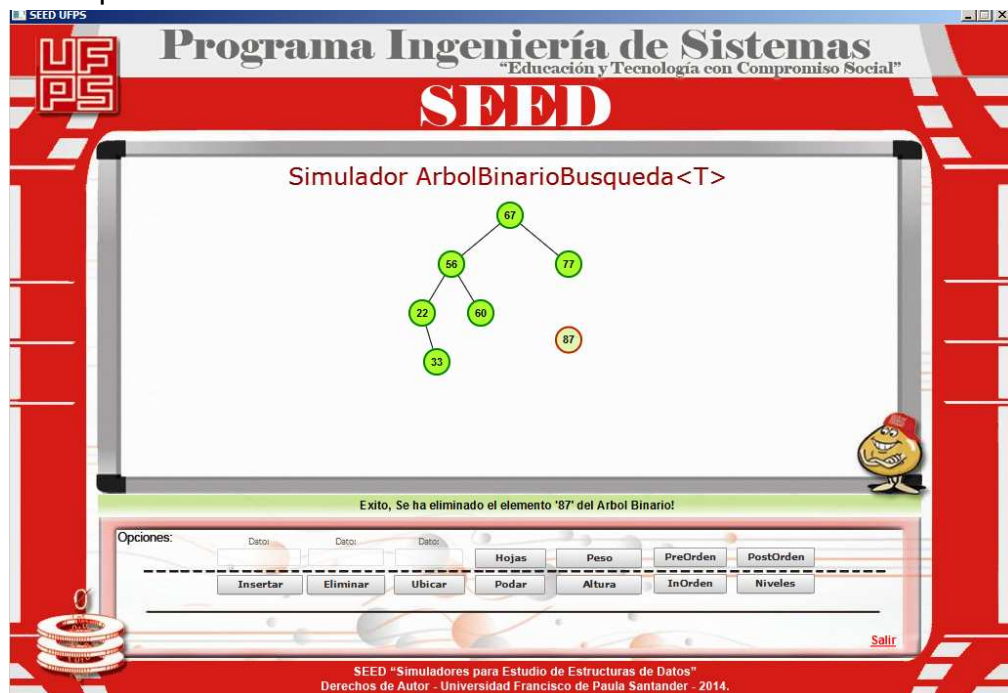
Para Insertar datos el estudiante simplemente deberá ingresar a la caja de texto el dato que desea insertar en el Árbol, este dato no puede ser menor a **-99** ni mayor a **999**, rango seleccionado por cuestiones de que no se desborde el número del nodo gráfico, además el árbol no puede superar la **altura 5**, con el fin de que la ilustración de la estructura no se desborde del Área de Dibujo del mediador pedagógico digital. A continuación el Árbol Binario De Búsqueda, debido a sus propiedades, ya no se debe seleccionar el padre ni el lado para insertar, ya que esta estructura inserta los datos en orden de forma inmediata, indicando el nuevo nodo que se insertó.



"Inserción en el Árbol Binario de Búsqueda de los datos: 67 – 56 – 22 – 33 – 60 – 87 y 77".

2. Eliminar Datos:

Para eliminar datos el estudiante deberá ingresar simplemente el dato que desea eliminar del Árbol Binario de Búsqueda (previamente insertado) en la caja de texto. Una vez eliminado el dato del Árbol Binario de Búsqueda, este no será pintado.



"Eliminación del dato: 87 del Árbol Binario De Búsqueda. 87 es bajado del Árbol y desaparece"

3. Búsqueda de un dato (Ubicar):

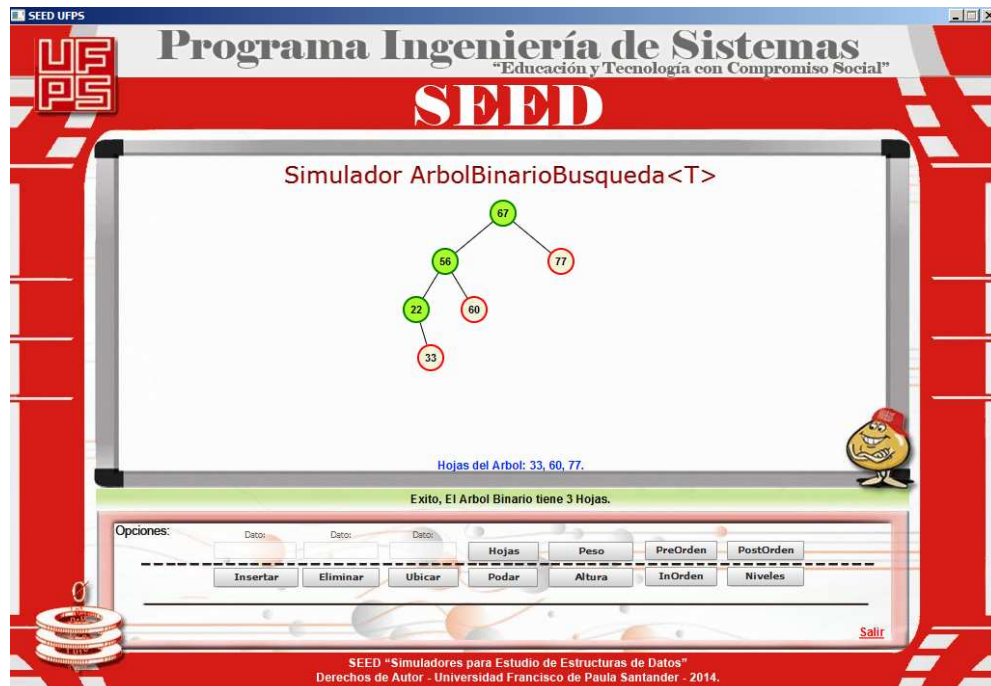
Para ubicar un dato dentro del Árbol Binario De Búsqueda el estudiante deberá ingresar en la caja de texto el dato que desea ubicar. La búsqueda del dato se realiza de acuerdo a las propiedades del Árbol y el costo algorítmico que cada uno genera. Para el Árbol Binario De Búsqueda, se indica el camino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



"Búsqueda del dato '33' dentro del Árbol Binario De Búsqueda"

4. Obtener las Hojas del Árbol

Para obtener los elementos Hoja del Árbol Binario de Búsqueda 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 de Búsqueda"

5. Podar el Árbol Binario de Búsqueda

El estudiante podrá podar las hojas de un Árbol Binario de Búsqueda dando **clic** en el botón **podar** de la aplicación. Los elementos Hoja del árbol serán eliminados, una vez podado el Árbol Binario de Búsqueda, las hojas dejaran de ilustrarse en el Área de dibujo.



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

6. Determinar el Peso y Altura del Árbol Binario de Búsqueda

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 de Búsqueda en un momento determinado.



"Determinar el peso del Árbol Binario de Búsqueda: 3 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 de Búsqueda en un momento determinado.



"Determinar la altura del Árbol Binario de Búsqueda: Altura 3"

7. Recorridos en el Árbol Binario de Búsqueda

Para recorrer el Árbol Binario de Búsqueda 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



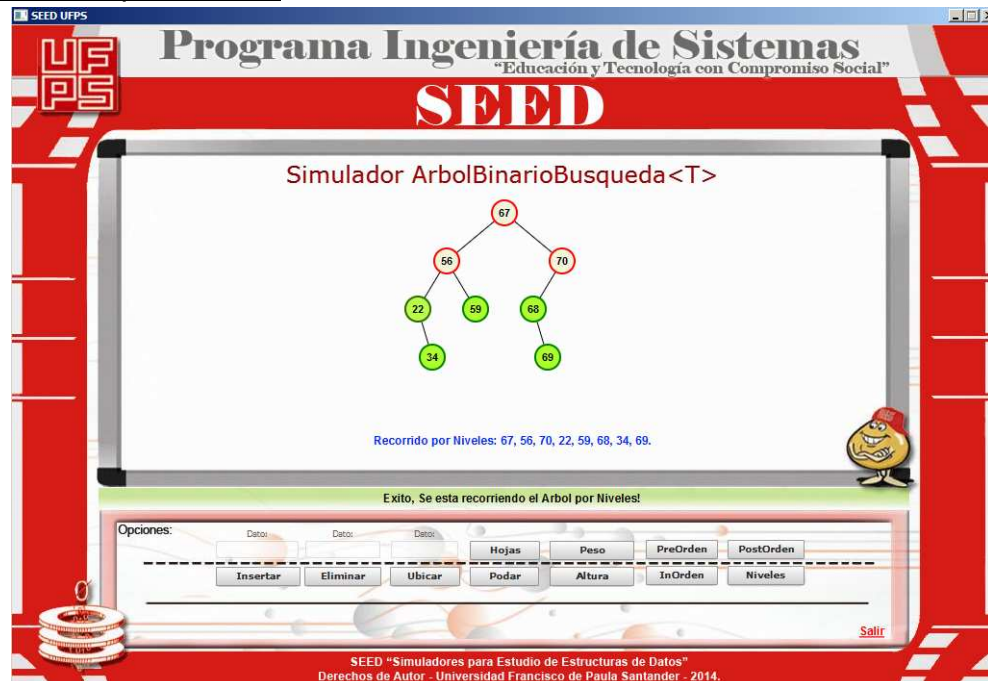
Recorrido en inOrden



Recorrido en postOrden



Recorrido por Niveles



8. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Árbol Binario de Búsqueda, 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 ArbolBinarioBusqueda:

8.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **ArbolBinarioBusqueda**, 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 de Busqueda:

```
/**
 * 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));
}

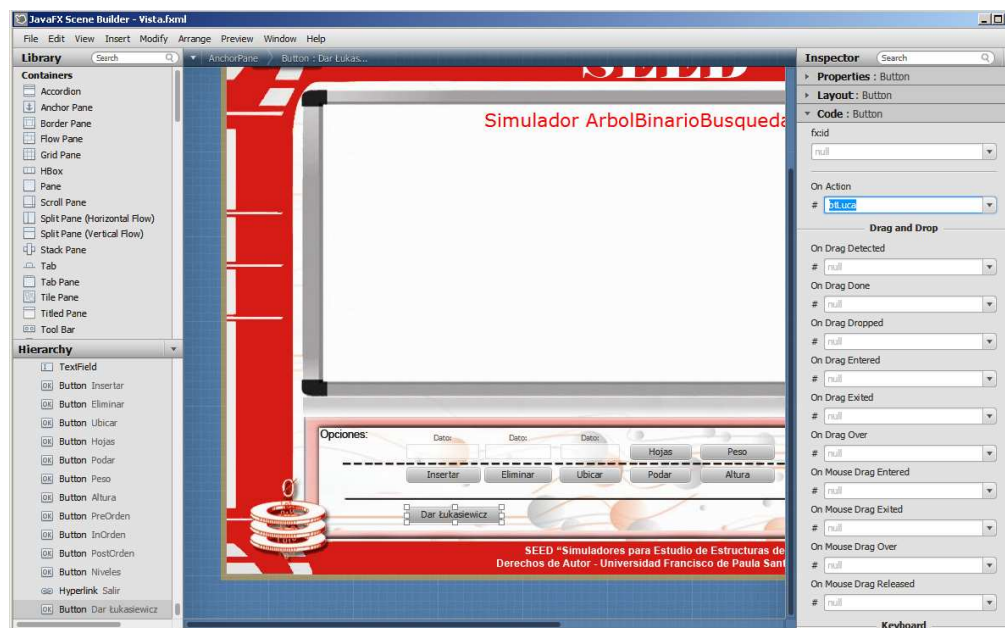
/**
 * Metodo que retorna el codigo Lukasiewicz del arbol binario; Este codigo etiqueta los nodos internos con "a"
 * y los externos con una "b" y realiza el recorrido en preorden con estas convenciones. <br>
 * <b>post: </b> Se retorno el codigo Lukasiewicz del arbol binario.<br>
 * @param r representa 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 **SimuladorArbolBB** del paquete **Mundo_ArbolBB**, que realice el llamado al Método con el nuevo algoritmo creado en la Estructura de Datos **ArbolBinarioBusqueda**. Para el llamado debe utilizar el objeto creado en el Mundo **miAbb**.

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

8.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **SimuladorArbolBB**. 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 **simarbolbinariobusqueda** en la clase **Simulador** que permite realizar el llamado a la nueva funcionalidad.

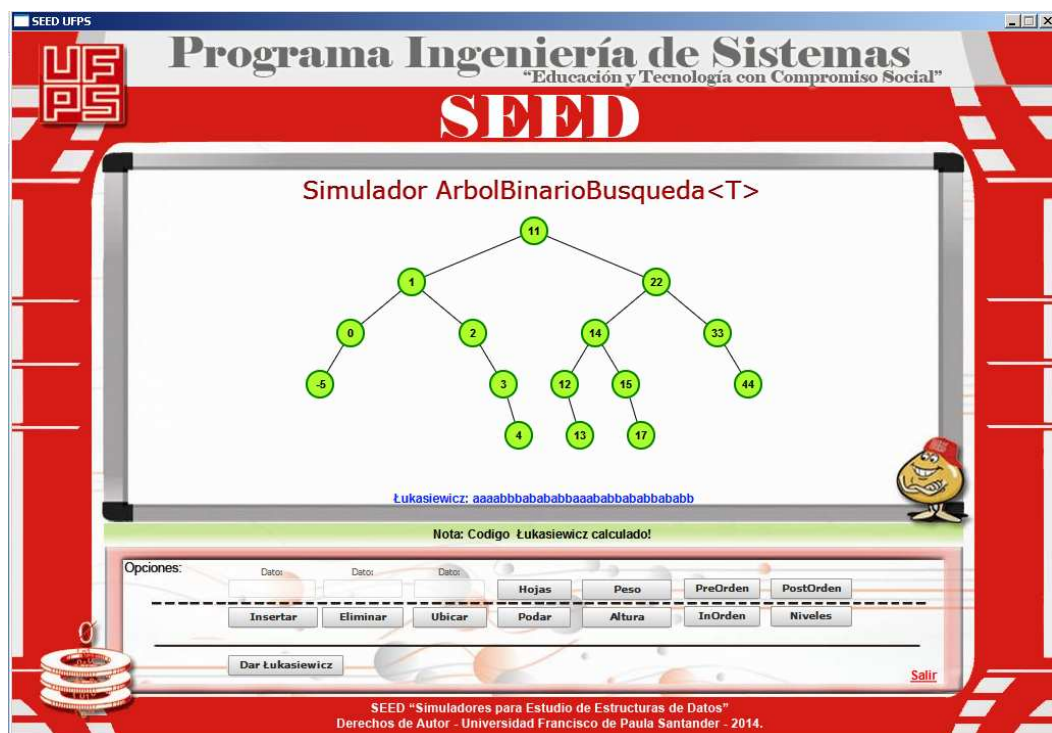
```

@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 **SimuladorArbolBB**, 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 de Búsqueda"