



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador ArbolAVL<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas
2014



MANUAL DE USUARIO: Simulador “ArbolAVL<T>”

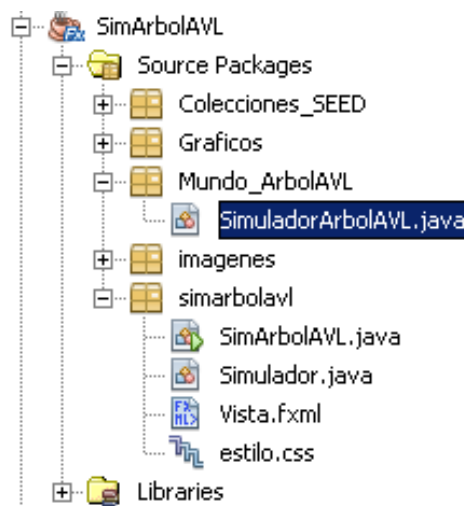
Descripción General

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 ArbolAVL<T>.

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

La implementación de este Simulador se desarrollada 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 el AVL que lo hacen diferente a los demás Árboles que heredan la estructura del Árbol Binario General.

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



“Interface principal del Simulador para ArbolAVL<T>”

Descripción de las Funcionalidades del Simulador

El simulador para ArbolAVL<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. A continuación el ArbolAVL, debido a sus propiedades de equilibrio por altura, realizara las operaciones de balanceo necesarias y el Árbol será pintado según como queda después de la inserción del nuevo dato.



"Inserción en el Árbol AVL de los datos: 44-22-10-1-23-77-56-45-70-88-99"

2. Eliminar Datos:

Para eliminar datos el estudiante deberá ingresar simplemente el dato que desea eliminar del Árbol AVL (previamente insertado) e inmediatamente el Árbol será balanceado y pintado correctamente en pantalla del Simulador.



"Eliminación del dato: 44 del ÁrbolAVL. 44 es bajado del Árbol y desaparece"

3. Búsqueda de un dato (Ubicar):

Para ubicar un dato dentro del ÁrbolAVL 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 Arbol AVL, se realiza una búsqueda binaria menos costosa.



"Búsqueda del dato '70' dentro del ÁrbolAVL"

4. Obtener las Hojas del Árbol

Para obtener los elementos Hoja del Árbol AVL 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 AVL"

5. Podar el Árbol AVL

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



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

6. Determinar el Peso y Altura del Árbol AVL

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



"Determinar el peso del Árbol AVL: 13 Elementos"

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



"Determinar la altura del Árbol AVL: Altura 4"

7. Recorridos en el Árbol AVL

Para recorrer el Árbol AVL 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 ArbolAVL<T>

Recorrido en PostOrden: 1, 10, 22, 44, 33, 11, 55, 66, 56, 99, 88, 77, 45.

Exito, Se esta recorriendo el Arbol en PostOrden!

Opciones:

Datos:		Datos:		Datos:		Hojas:		Peso:		PreOrden:		PostOrden:	
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 ArbolAVL<T>

Recorrido por Niveles: 45, 11, 77, 10, 33, 56, 88, 1, 22, 44, 55, 66, 99.

Exito, Se esta recorriendo el Arbol por Niveles!

Opciones:

Datos:		Datos:		Datos:		Hojas:		Peso:		PreOrden:		PostOrden:	
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 AVL, 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 ArbolAVL:

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

```
/**
 * 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 **SimuladorArbolAVL** del paquete **Mundo_ArbolAVL**, que realice el **llamado** al Método con el nuevo algoritmo creado en la Estructura de Datos **ArbolAVL**. Para el llamado debe utilizar el objeto creado en el Mundo **miArbolAVL**.

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

8.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **SimuladorArbolAVL**. 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 Łukasiewicz" />
```

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 **simarbolavl** 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 Łukasiewicz calculado!",0);
}
```

Además del llamado al Método creado en **SimuladorArbolAVL**, 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:

