



# SEED

“Simuladores para Estudio de Estructuras de Datos”

## Manual de Usuario

*Simulador Arbol1-2-3<T>*

*Versión: 1.0*

Universidad Francisco de Paula Santander  
Programa Ingeniería de Sistemas

2014



## MANUAL DE USUARIO: Simulador “Árbol1-2-3<T>”

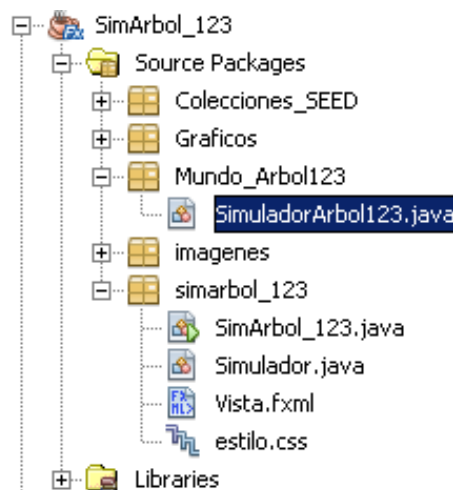
### Descripción General

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 Árbol1-2-3<T>.

Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Árbol1-2-3: 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 Árboles1-2-3, 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 Árbol1-2-3<T>”*

La implementación de este Simulador, aunque no se fundamenta en copiar las características de otro, se ha implementado teniendo en cuenta las funciones desarrolladas para los Simuladores anteriores como: Árbol Binario, Binario de Búsqueda, AVL, Rojinegros y Heaps.

A continuación se presenta la interface principal del simulador para “Árbol1-2-3”.



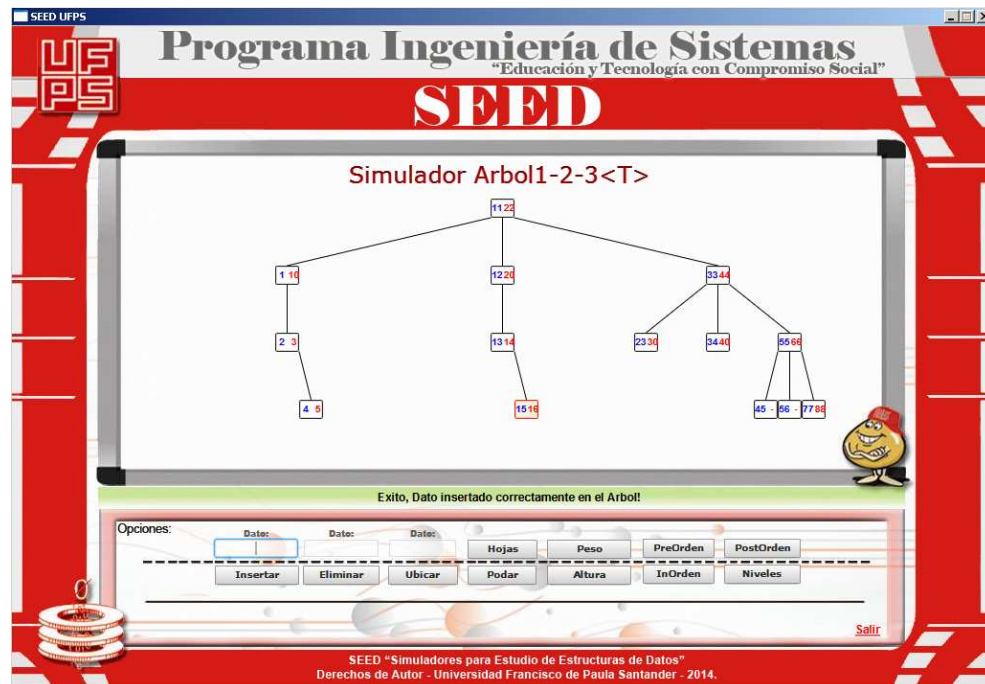
*“Interface principal del Simulador para Arbol1-2-3<T>”*

## Descripción de las Funcionalidades del Simulador

El simulador para Arbol1-2-3<T> permite al Estudiante:

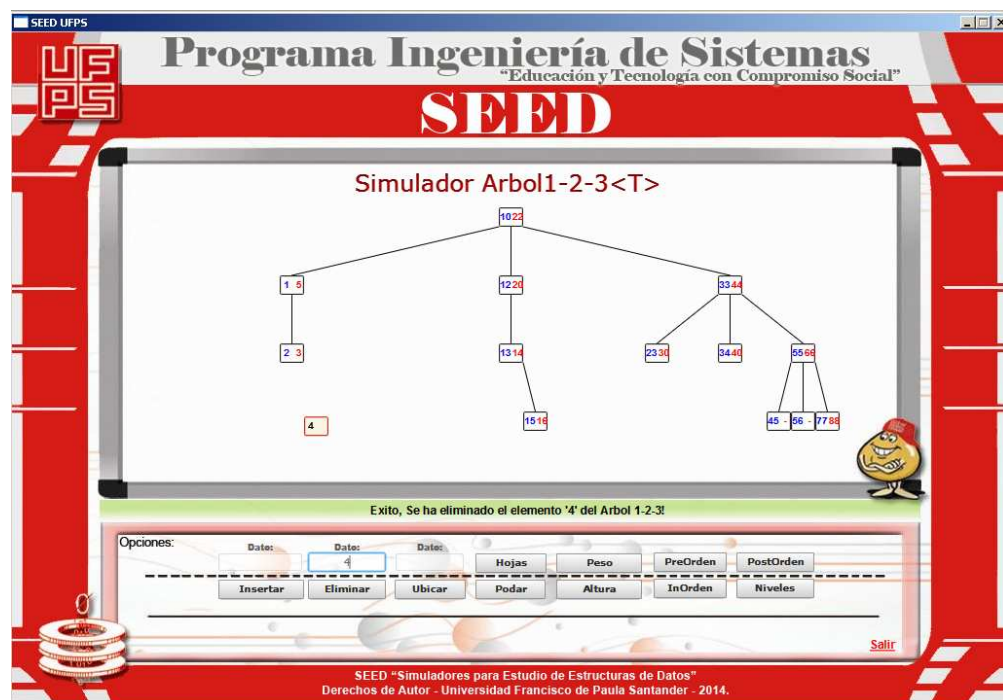
### 1. Insertar Datos:

Para Insertar datos el estudiante deberá ingresar en la caja de texto el dato que desea insertar en el Árbol, de manera que este será insertado en el lugar correspondiente dependiendo del valor de su dato. Inmediatamente el Arbol1-2-3 es pintado con el dato insertado:



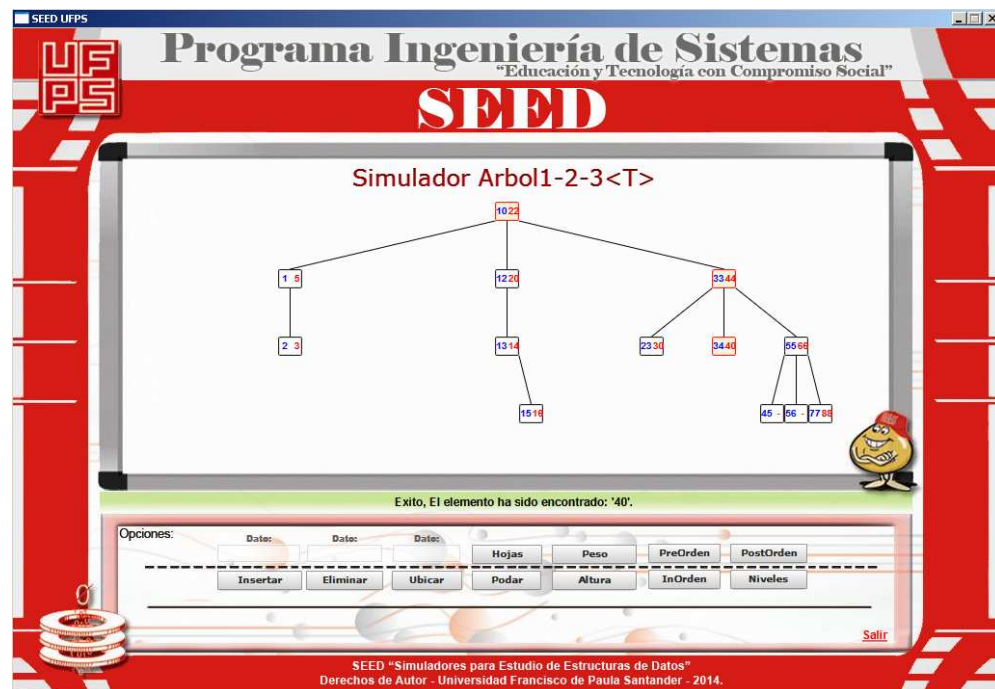
## 2. Eliminar Datos:

Para eliminar datos el estudiante deberá ingresar simplemente el dato que desea eliminar del Árbol1-2-3 (previamente insertado) e inmediatamente el Árbol será pintado sin el dato que ha sido eliminado del mismo.



### 3. Búsqueda de un dato (Ubicar):

Para ubicar un dato dentro del Árbol1-2-3 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.

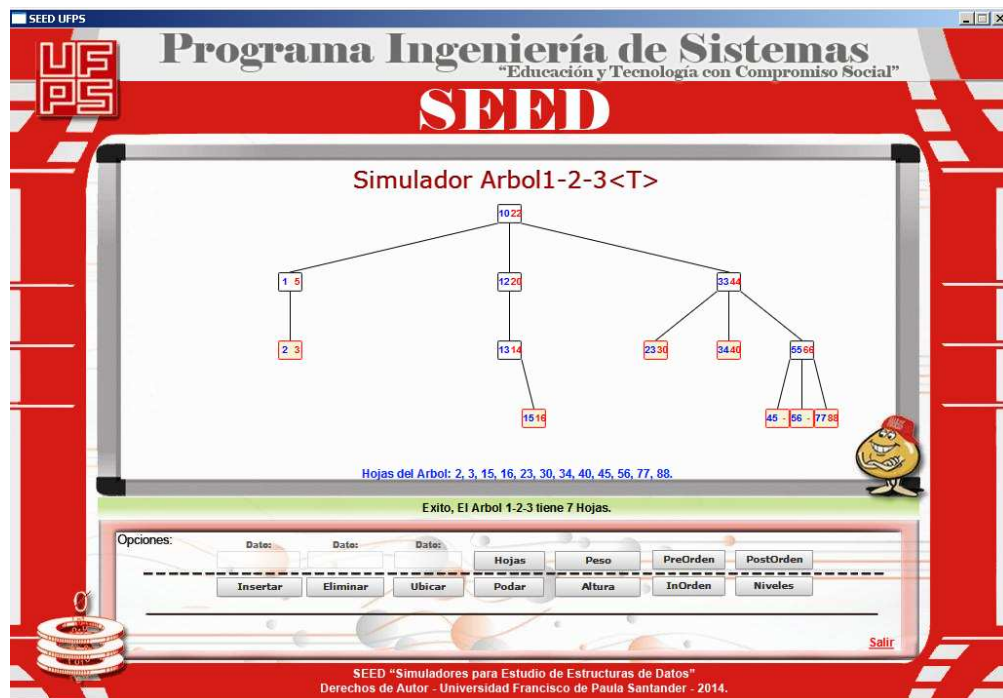


*"Búsqueda del dato '40' dentro del Árbol1-2-3"*

### 4. Obtener las Hojas del Árbol

Para obtener los elementos Hoja del Árbol1-2-3 el estudiante deberá dar **clic** en el botón **Hojas**, y estas serán pintadas en el Árbol para que el estudiante las pueda identificar.

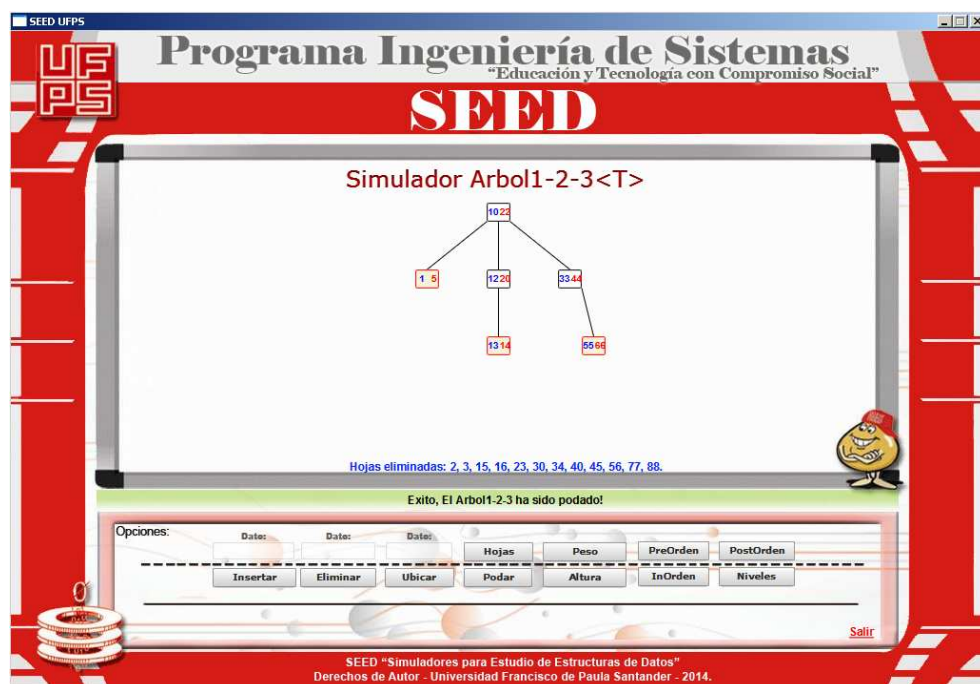




*"Determinación de los elementos hoja del Árbol1-2-3"*

## 5. Podar el Árbol1-2-3

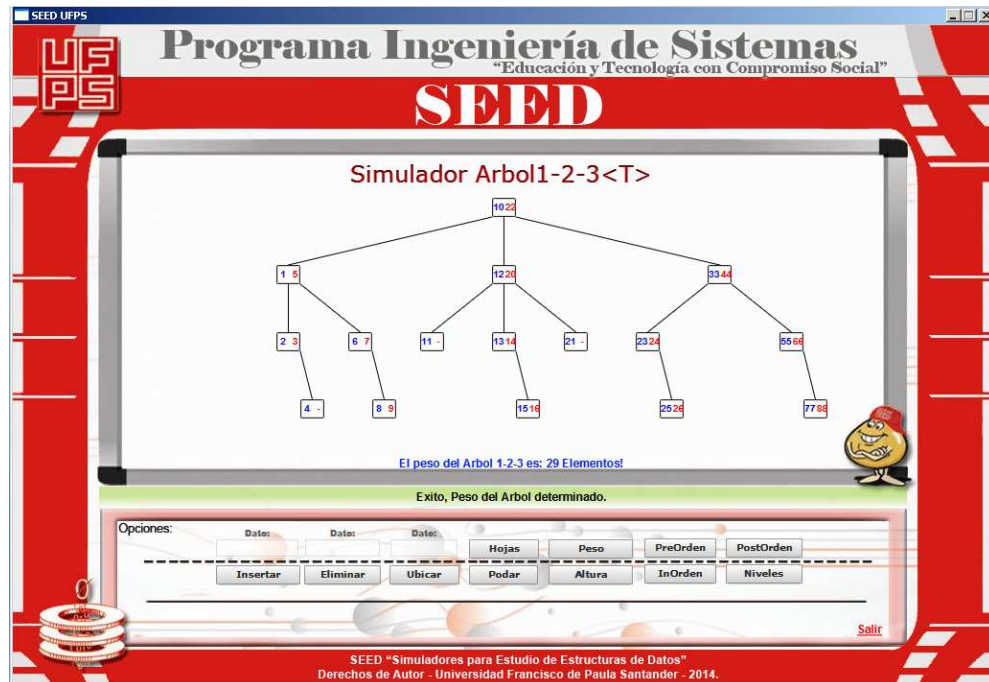
El estudiante podrá podar las hojas de un Árbol1-2-3 oprimiendo **clic** en el botón **podar** de la aplicación. Los elementos Hoja del árbol serán eliminados de forma que las hojas del Árbol1-2-3 serán ahora los padres de estos elementos.



*"Podar el Árbol1-2-3 anterior. Las hojas han sido eliminadas"*

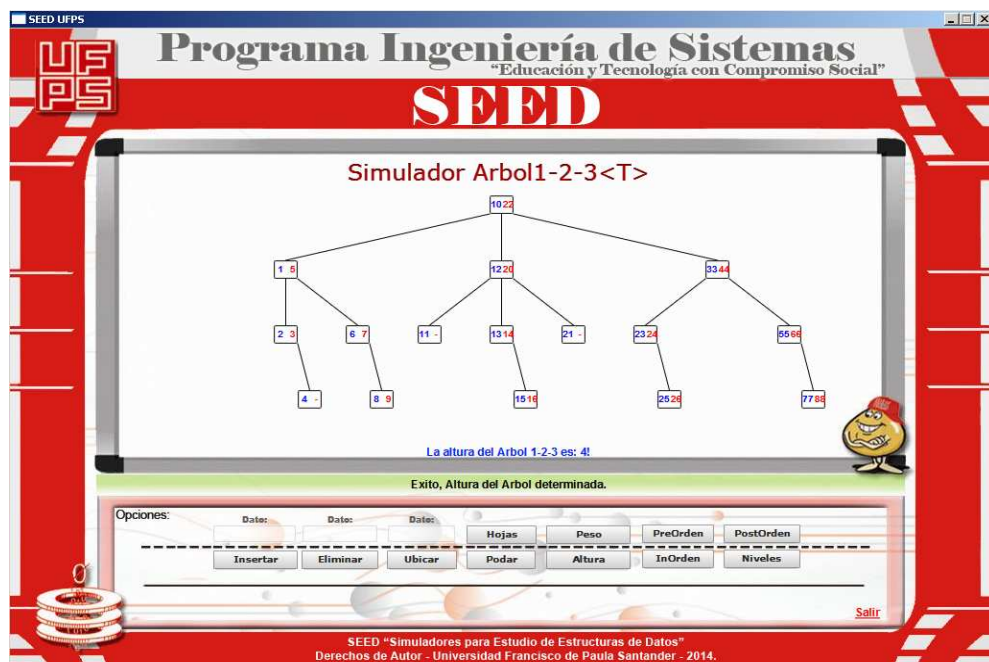
## 6. Determinar el Peso y Altura del Árbol1-2-3

Para determinar el Peso del Árbol el estudiante deberá oprimir el botón **peso**, inmediatamente la aplicación indicará la cantidad de datos presentes en el Árbol1-2-3 en un momento determinado.



*"Determinar el peso del Árbol1-2-3: 29 Elementos"*

Ahora para determinar la Altura del Árbol el estudiante deberá oprimir el botón **altura**, inmediatamente la aplicación indicará la altura del Árbol1-2-3 en un momento determinado.

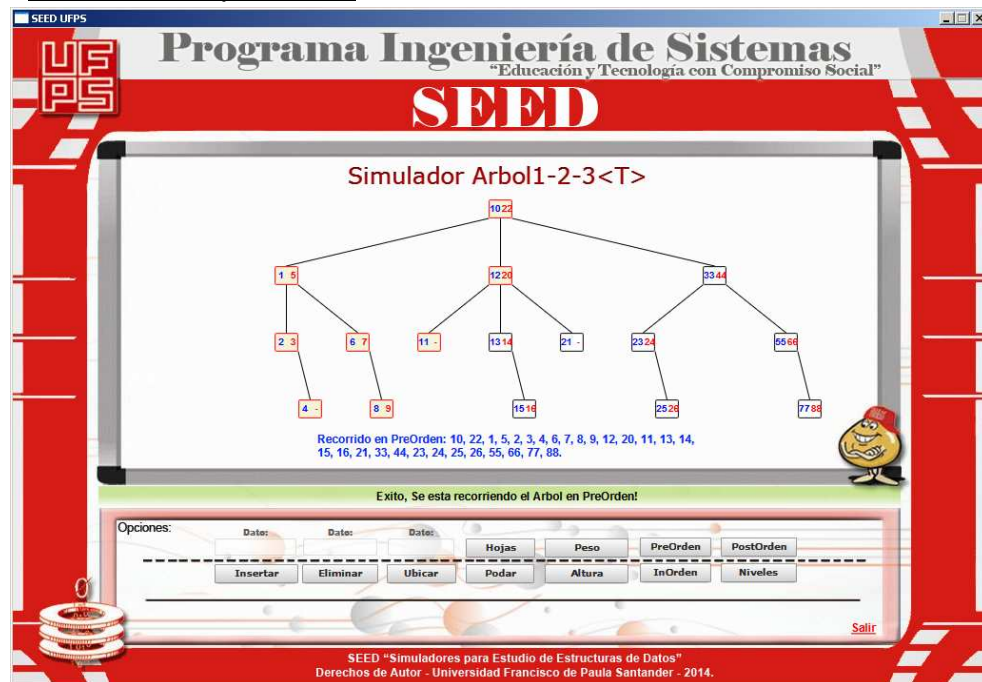


*"Determinar la altura del Árbol1-2-3: Altura 4"*

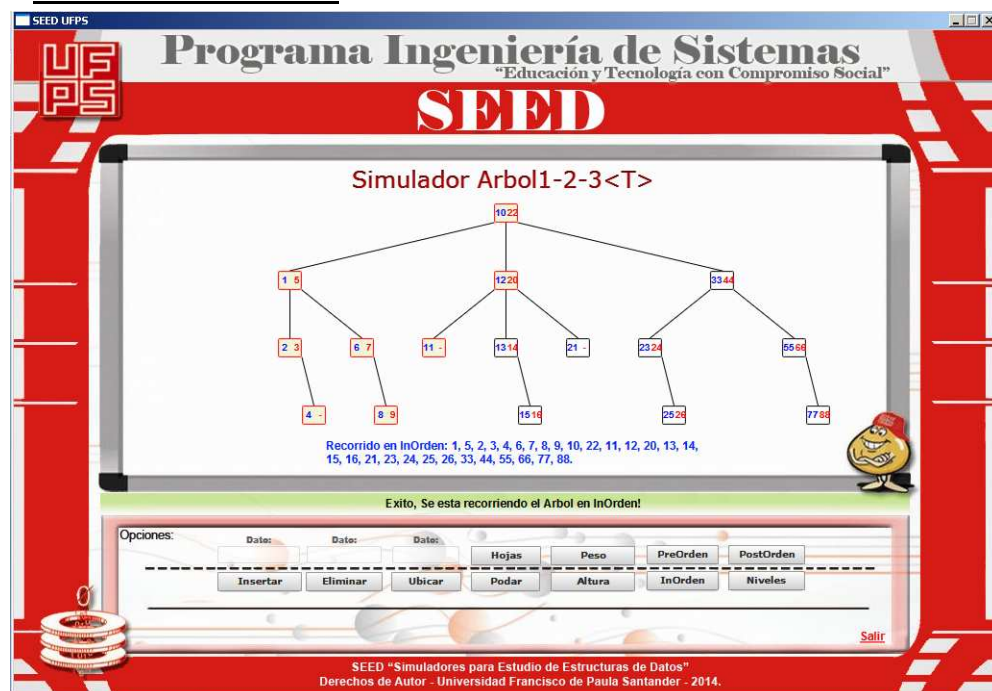
## 7. Recorridos en el Árbol1-2-3

Para recorrer el Árbol1-2-3 el estudiante deberá seleccionar el recorrido que desea realizar en el Árbol e inmediatamente el simulador realizará una animación. Debe dar clic en el botón: **preOrden**, **inOrden**, **postOrden** y **por Niveles**.

Recorrido en preOrden

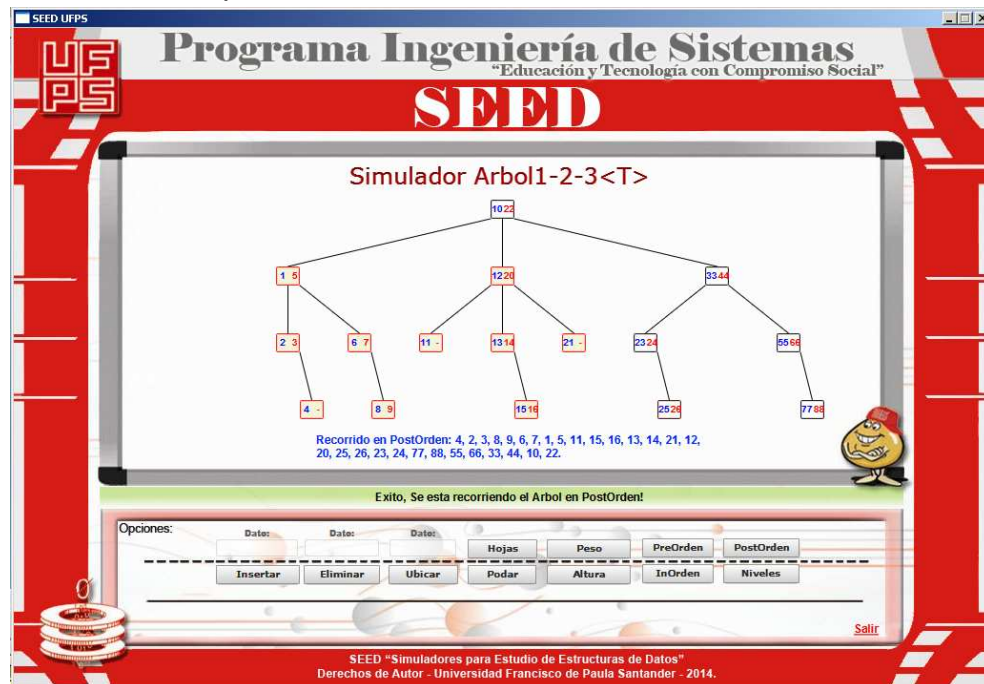


Recorrido en inOrden

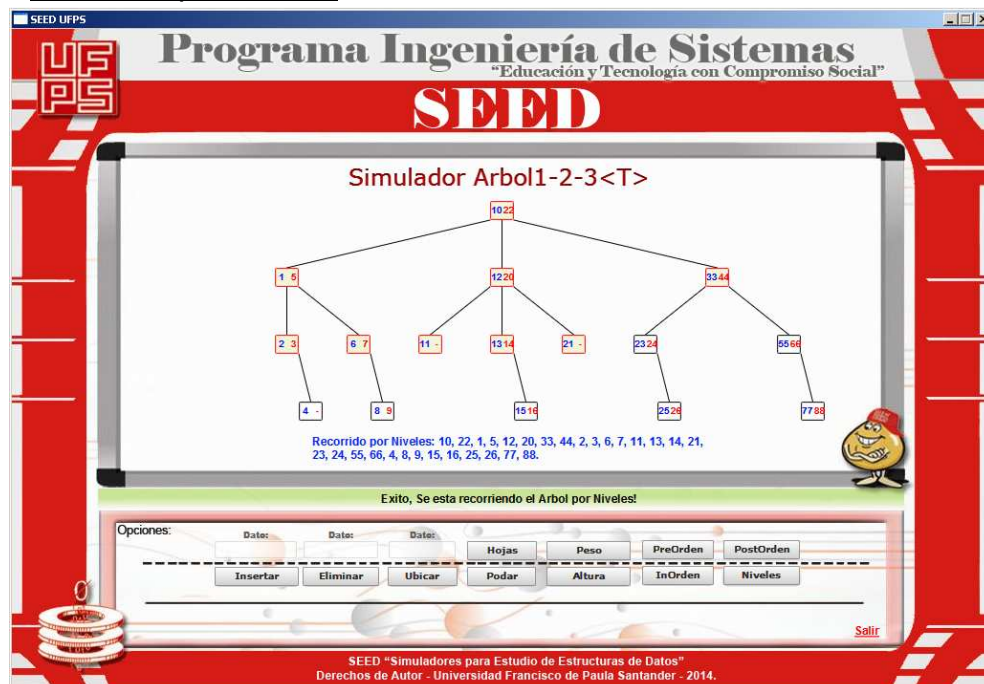




## Recorrido en postOrden



## Recorrido por Niveles



## 8. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Árbol1-2-3, 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 Arbol1-2-3:

- 8.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **Arbol123**, presente en el paquete **SEED\_Colecciones** y que desea adicionar a la funcionalidad del Simulador. (Para el ejemplo, se creará un algoritmo que permita imprimir en pantalla las raíces del Arbol1-2-3):

```
/**
 * Metodo que permite conocer el dato menor en la raiz del Arbol 1-2-3. <br>
 * <b>post: </b> Se obtuvo la raiz del Arbol 1-2-3.<br>
 * @return Un objeto de tipo Nodo1-2-3<T> que es la raiz del Arbol 1-2-3.
 */
public T getInfoMenRaiz() {
    return raiz.getInfoMen();
}

/**
 * Metodo que permite conocer el dato mayor en la raiz del Arbol 1-2-3. <br>
 * <b>post: </b> Se obtuvo la raiz del Arbol 1-2-3.<br>
 * @return Un objeto de tipo Nodo1-2-3<T> que es la raiz del Arbol 1-2-3.
 */
public T getInfoMayRaiz() {
    return raiz.getInfoMay();
}
```

- 8.2. A continuación el Estudiante debe generar un Método en la clase **SimuladorArbol123** del paquete **Mundo\_Arbol123**, que realice el **llamado** al Método con el nuevo algoritmo creado en la Estructura de Datos **Arbol123**. Para el llamado debe utilizar el objeto creado en el Mundo **miArbol1-2-3**.

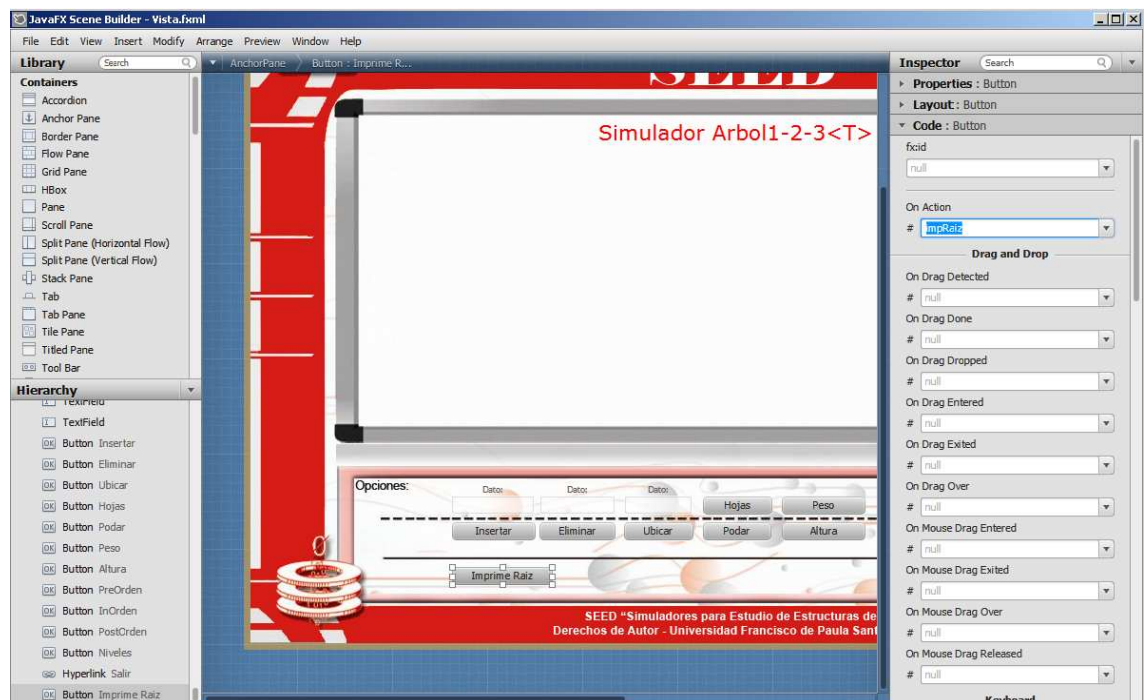
```

public String impRaices(){
    String cad = "Raices del Arbol: \t";
    cad+= "Raiz menor(";
    if(this.miArbol123.getInfoMenRaiz() !=null)
        cad+= this.miArbol123.getInfoMenRaiz().toString()+") ";
    else
        cad+="null) ";
    cad+= ", Raiz mayor(";
    if(this.miArbol123.getInfoMayRaiz() !=null)
        cad+= this.miArbol123.getInfoMayRaiz().toString()+") ";
    else
        cad+="null) ";
    return (cad);
}

```

8.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **SimuladorArbol123**. 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="212.0" layoutY="610.0" mnemonicParsing="false" onAction="#impRaiz" prefWidth="106.0" text="Imprime Raiz" />

```

Es importante resaltar, para ambos casos, que se debe asignar el evento “**OnAction**” del Button, para el ejemplo “**impRaiz**”, 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 **simarbol\_123** en la clase **Simulador** que permite realizar el llamado a la nueva funcionalidad.

```
@FXML
private void impRaiz() {
    this.pintarTDA();
    this.msg.setText(this.simulador.impRaices());
    this.msg.setVisible(true);
    this.impNota("Se han imprimido las raices del Arbol 1-2-3", 0);
}
```

Además del llamado al Método creado en **SimuladorArbol123**, el estudiante deberá invocar el método que le permita volver pintar el Árbol, el cual siempre será **"pintarTDA()"**. Opcionalmente se recomienda enviar un mensaje con la respuesta a la operación realizada utilizando **"impNota(" Mensaje a enviar ", tipo)"** donde tipo es cero (0) si es un mensaje Exitoso y uno (1) en caso de ser un mensaje erróneo.

A continuación se comprueba el funcionamiento del Algoritmo realizado:



*"Se imprimió la raíz del Árbol Heap"*