



SEED

“Simuladores para Estudio de Estructuras de Datos”

Manual de Usuario

Simulador ArbolHeap<T>

Versión: 1.0

Universidad Francisco de Paula Santander
Programa Ingeniería de Sistemas

2014



MANUAL DE USUARIO: Simulador “ÁrbolHeap<T>”

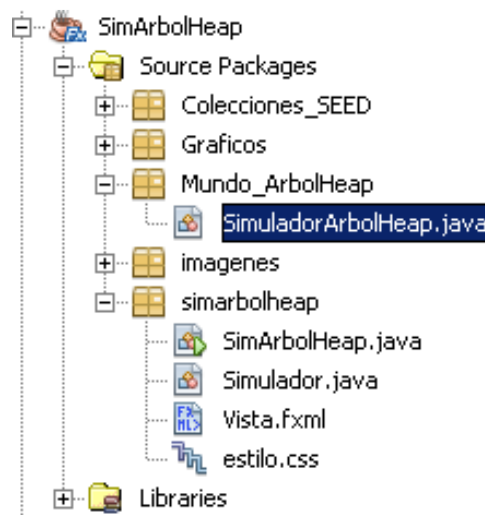
Descripción General

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 ÁrbolHeap<T>.

Dentro de esta aplicación encontrará el estudiante un conjunto de operaciones relacionadas con las funciones básicas implementadas para la estructura Árbol Heap:Insertar, Eliminary ordenar los datos utilizando HeapSort especialmente para este tipo de 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 Heap, 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 ArbolHeap<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 Heap 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 “ÁrbolHeap”.



“Interface principal del Simulador para ArbolHeap<T>”

Descripción de las Funcionalidades del Simulador

El simulador para ArbolHeap<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 ArbolHeap, pintará el nuevo dato de acuerdo a las propiedades, subiendo el dato hasta la posición correcta.



"Inserción en el ÁrbolHeap de los datos: 11-12-66-9-99-78-65-76-87-44-45-33-23-2-5"

2. Eliminar Datos:

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



"Eliminación del dato: 87 del ÁrbolHeap.87 es bajado del Árbol y desaparece"

3. Obtener las Hojas del Árbol

Para obtener los elementos Hoja del Árbol Heapel 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 Heap"

4. Podar el ÁrbolHeap

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



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

5. Determinar el Peso y Altura del Árbol Heap

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 Heap en un momento determinado.



"Determinar el peso del Árbol Heap: 14 Elementos"

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



“Determinar la altura del Árbol Heap: Altura 4”

6. Recorridos en el Árbol Heap

Para recorrer el Árbol Heap 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

SEED UFPS

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

SEED

Simulador ArbolHeap<T>

```
graph TD; 99((99)) --> 78((78)); 99 --> 88((88)); 78 --> 65((65)); 78 --> 76((76)); 88 --> 66((66)); 88 --> 77((77)); 65 --> 11((11)); 65 --> 22((22)); 76 --> 33((33)); 76 --> 55((55)); 66 --> 44((44)); 66 --> 45((45)); 77 --> 5((5));
```

Recorrido en InOrden: 11, 65, 22, 78, 33, 76, 55, 99, 44, 66, 45, 88, 5, 77.

Exito, Se esta recorriendo el Arbol en InOrden!

Opciones:

Date:	Date:	Hojas	Peso	PreOrden	PostOrden	HeapSort
Insertar	Eliminar	Podar	Altura	InOrden	Niveles	Vaciar

Salir

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

Recorrido en postOrden

SEED UFPS

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

SEED

Simulador ArbolHeap<T>

```
graph TD; 99((99)) --> 78((78)); 99 --> 88((88)); 78 --> 65((65)); 78 --> 76((76)); 88 --> 66((66)); 88 --> 77((77)); 65 --> 11((11)); 65 --> 22((22)); 76 --> 33((33)); 76 --> 55((55)); 66 --> 44((44)); 66 --> 45((45)); 77 --> 5((5));
```

Recorrido en PostOrden: 11, 22, 65, 33, 55, 76, 78, 44, 45, 66, 5, 77, 88, 99.

Exito, Se esta recorriendo el Arbol en PostOrden!

Opciones:

Date:	Date:	Hojas	Peso	PreOrden	PostOrden	HeapSort
Insertar	Eliminar	Podar	Altura	InOrden	Niveles	Vaciar

Salir

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

Recorrido por Niveles



7. Ordenar por HeapSort

Para ordenar los datos del Árbol por HeapSort el estudiante deberá oprimir **clic** en el botón **HeapSort**, de esta forma los datos del Árbol (organizados en un vector), serán sacados del Árbol, creando un nuevo vector con los datos ordenados ascendentemente.



8. Adicionar nuevas funcionalidades:

Adicionalmente a las funciones incorporadas para el Simulador de Árbol Heap, 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 ArbolHeap:

- 8.1. El estudiante debe generar el nuevo Algoritmo dentro de la Estructura de Datos **ArbolHeap**, presente en el paquete **SEED_Colecciones** y que desea adicionar a la funcionalidad del Simulador. (Para el ejemplo, se creará un algoritmo que permita eliminar la raíz del Árbol Heap (que siempre es el dato de mayor valor en el Árbol):

```
/**
 * Metodo que permite eliminar el primer elemento (raiz) de Arbol Heap; El elemento MAYOR. <br>
 * <b>post:</b> Se retorno y elimino el elemento raiz del Arbol. El dato MAYOR. <br>
 * @return El elemento Mayor del Arbol que se ubica en la raiz.
 */
public T eliminarRaiz(){
    if(peso<=0)
        return (null);
    this.datos = intercambiar(this.datos, 0, --peso);
    if (peso!=0)
        reorganiza(0);
    T x = this.datos[peso];
    this.datos[peso] = null;
    return (x);
}
```

- 8.2. A continuación el Estudiante debe generar un Método en la clase **SimuladorArbolHeap** del paquete **Mundo_ArbolHeap**, que realice el **llamado** al Método con el nuevo algoritmo creado en la Estructura de Datos **ArbolHeap**. Para el llamado debe utilizar el objeto creado en el Mundo **miArbolHeap**.

```
public int eliminarRaiz(){
    return (this.miHeap.eliminarRaiz());
}
```

- 8.3. Por último, se deberá crear el componente grafico (para el ejemplo **Button**) que permita realizar el llamado al Método creado en **SimuladorArbolHeap**. 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="#btEliRaiz" prefWidth="106.0" text="Eliminar Raiz" />
```

Es importante resaltar, para ambos casos, que se debe asignar el evento **"OnAction"** del Button, para el ejemplo **"btEliRaiz"**, 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 **simarbolheap** en la clase **Simulador** que permite realizar el llamado a la nueva funcionalidad.

```
@FXML
private void btEliRaiz() {
    int dato = this.simulador.eliminarRaiz();
    this.pintarTDA();
    this.msg.setText("Dato eliminado de la raiz: "+dato);
    msg.setVisible(true);
    this.impNota("Raiz del Arbol eliminada!",0);
}
```

Además del llamado al Método creado en **SimuladorArbolHeap**, 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 en un mensaje Exitoso y uno (1) en caso de ser un mensaje erróneo.

A continuación se comprueba el funcionamiento del Algoritmo realizado:



Después de eliminada la raíz “99” el Árbol queda de la siguiente forma:



“Se elimino la raíz del Árbol Heap”