

UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORÍA ACADÉMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
CARRERA INGENIERÍA INFORMÁTICA



CÁTEDRA INGENIERÍA DE SOFTWARE

ASIGNATURA

00825 ESTRUCTURA DE DATOS

PROYECTO 3

VALOR: 30% (3.0)

I CUATRIMESTRE 2024

Proyecto 3

Objetivo de aprendizaje

- Aplicar el funcionamiento de la estructura de datos conocida como árbol y árbol binario de búsqueda.

Temas de estudio

- Árboles.
- Árboles binarios de búsqueda.

Descripción del trabajo

Precondiciones:

1. Como herramienta de desarrollo se deberá utilizar el Netbeans en su versión 19 o superior.
2. El programa debe realizarse en modo gráfico (GUI), es decir, no se permite en modo consola.
3. Cuando se soliciten o se muestren datos al usuario, no se permite el uso de cuadros de diálogo tipo MessageBox. Para ello se pueden utilizar cajas de texto, etiquetas o listas gráficas según sea. Esto a menos que sea para mostrar excepciones de la aplicación o para dar un mensaje al usuario de que omitió algo en la operación. Si la operación se realiza exitosamente, no se deben mostrar mensajes utilizando estos cuadros de diálogo.
4. Los datos deben persistir en memoria en todo momento hasta que se cierre la aplicación.

Instrucciones:

Se debe crear un programa que gestione un árbol binario.

El árbol debe crearse con clases Nodo, donde cada objeto contendrá la información requerida y el puntero al siguiente elemento.

No se podrá utilizar clases ya implementadas de Java para la gestión de este árbol binario.

En una única pantalla, el usuario tendrá las siguientes opciones:

Inserción:

El programa le solicitará al usuario los datos de un dispositivo móvil para incluirlo en el árbol binario. Estos datos son:

Nombre del campo	Tipo de dato
Id	Número entero
Marca	Texto
Año de creación	Número entero de cuatro dígitos
Sistema operativo	Se muestra una lista tipo DropDownList con las siguientes opciones: <ul style="list-style-type: none">● Android● iOS

- Windows 10 Mobile
- Symbian OS
- Firefox OS
- Ubuntu Touch
- Harmony OS

El usuario debe crear un nodo con el objeto “Móvil” (basado en los campos anteriores) y el programa lo guardará en el árbol.

La clase Móvil será la siguiente:

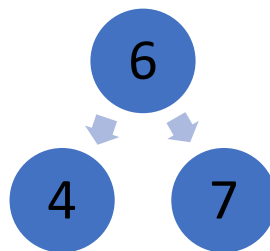
```
public class Movil
{
    private int Id;
    private String Marca;
    private int AnhoCreacion;
    private String SO;
}
```

Antes de incluir el nodo, el programa debe buscar el valor del campo Id entre los nodos existentes en el árbol binario a ver si existe, ya que no se permitirá crear un nodo con un Id que ya fue incluido anteriormente.

El primer Nodo ingresado será la raíz. A partir del segundo Nodo a ingresar, se debe insertar en forma ordenada por el Id, de modo que, si el Id es menor al Id de la raíz, se debe insertar a la izquierda, de lo contrario a la derecha.

Imagen 1

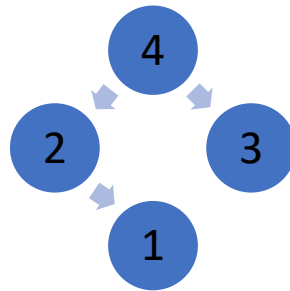
Ejemplo de árbol correctamente ordenado (el número mostrado es el del campo ID):



Nota: la figura muestra un árbol cuya raíz es 6, y con un nodo hijo a la izquierda con el ID 4 y otro nodo hijo a la derecha con el ID 7.

Imagen 2

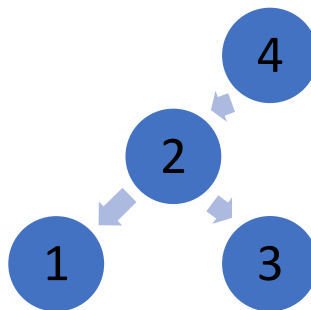
Ejemplo de árbol ordenado incorrectamente:



Nota: la figura muestra un árbol cuya raíz es 4, y con nodo hijo a la izquierda con el ID 2 (este tiene a su vez, un nodo hijo a la derecha con ID 1) y otro nodo hijo a la derecha con el ID 3.

Imagen 3

El ordenamiento correcto del árbol anterior es:



Nota: la figura muestra un árbol cuya raíz es 4, y con nodo hijo a la izquierda con el ID 2, este a su vez tiene un nodo hijo a la izquierda con ID 1 y a la derecha con ID 3.

En el ejemplo anterior, para lograr ese ordenamiento, el primer elemento a insertar sería el que tiene el Id = 4, el cual queda de raíz. A partir de ahí se empieza a ordenar cada nuevo Nodo incluido. Posteriormente, ingresó el Id = 2, la cual al ser menor que el Id = 4 se ubica a su izquierda. Luego ingresa el Id = 3 el cual es menor que el Id de la raíz y mayor que el Id = 2, por lo que se ubica a la derecha del Id = 2.

Posteriormente ingresa el Id = 1, el cual es menor que el Id = 4 y menor que el Id = 2, entonces va a la izquierda de este.

No se deben insertar Nodos con el Id repetido. El programa deberá realizar la validación correspondiente recorriendo el árbol binario (no se permite utilizar otra estructura paralela). Si el valor del Id ya existe, se muestra un mensaje indicando que no se pudo incluir el elemento.

Eliminación:

El usuario digita el valor del Id a eliminar. El programa procederá a eliminar el nodo correspondiente considerando lo siguiente:

1. Si el nodo es un nodo hoja, solo se elimina y el puntero de su padre se hace null.
2. Si el nodo tiene solo un subárbol hijo, se elimina y el padre del subárbol toma su lugar.
3. Si el nodo tiene dos subárboles hijos, no se elimina y se muestra una etiqueta en la pantalla indicando que no se puede eliminar el nodo porque tiene 2 hijos.

Búsqueda de Id de un dispositivo móvil

El usuario indicará el valor del Id a Buscar dentro del árbol. Si existe, se mostrará la información completa del nodo, es decir, la marca, el año de creación y el sistema operativo del nodo correspondiente.

Esta búsqueda se realizará haciendo un recorrido del árbol usando los punteros de los nodos. No se permite el uso de una estructura paralela.

Recorrido PRE-ORDEN

Esta opción mostrará la lista de nodos (campos Id separados por guiones) utilizando un recorrido pre-orden (no usar un MessageBox).

Recorrido POST-ORDEN

Esta opción mostrará la lista de nodos (campos Id separados por guiones) utilizando un recorrido post-orden (no usar un MessageBox).

Recorrido IN-ORDEN

Esta opción mostrará la lista de nodos (campos Id separados por guiones) utilizando un recorrido in-orden (no usar un MessageBox).

Mostrar nodos hojas

Esta opción se mostrarán los nodos que son hojas en el árbol.

Mostrar altura del árbol

En esta opción se mostrará la altura del árbol.

Debe existir un menú con cada una de las opciones a ejecutar.

Cada una de las funciones se pueden realizar en cualquier momento y en cualquier orden.

No se debe limpiar el árbol binario en ningún momento, a menos que sea utilizando la función de Eliminación (para todos los nodos) o al detener la ejecución del programa.

Rúbrica

NO.	INDICADORES POR EVALUAR	CUMPLIMIENTO		PUNTOS
		Cumple	No cumple	
INSERCIÓN				
1.	El programa le solicita al usuario los datos de un dispositivo móvil incluyendo la información de los 4 campos especificados en el apartado Inserción del enunciado, considerando sus respectivos tipos de dato. 1 punto por cada uno.			4
2.	El programa crea un árbol binario con clases Nodo, las cuales incluyen los 4 campos del dispositivo móvil indicados en el enunciado, así como el puntero al siguiente elemento. No utiliza clases ya implementadas de Java para la gestión del árbol binario.			5
3.	No se permite la inserción de nodos con el Id repetido. El programa realiza la validación correspondiente recorriendo el árbol binario (no utiliza otra estructura paralela). Si el valor del Id ya existe, se muestra un mensaje indicando que no se pudo incluir el elemento.			1
4.	La inserción se hace considerando el ordenamiento indicado en el enunciado en el apartado Inserción.			1
ELIMINACIÓN				
5.	El usuario puede digitar el valor del Id a eliminar.			1
6.	El programa elimina correctamente nodos hojas.			1
7.	El programa elimina correctamente nodos con un solo hijo o subárbol.			1
8.	El programa valida si el nodo a eliminar tiene dos hijos e impide su eliminación mostrando el mensaje correspondiente.			1
BÚSQUEDA DEL ID DE UN DISPOSITIVO MÓVIL				
9.	El usuario puede indicar el valor del Id a Buscar dentro del árbol.			1
10.	Si el nodo a buscar existe, se mostrará la información completa del nodo, es decir, la marca, el año de creación y el sistema operativo del nodo correspondiente. Esta búsqueda se realizará haciendo un recorrido del árbol usando los punteros de los nodos. No se permite el uso de una estructura paralela.			3
RECORRIDOS				
11.	El programa hace un recorrido PRE-ORDEN del árbol y muestra el resultado (lista de IDs separados por guiones) sin usar un MessageBox.			1
12.	El programa hace un recorrido POST-ORDEN del árbol y muestra el resultado (lista de IDs separados por guiones) sin usar un MessageBox.			1

13.	El programa hace un recorrido IN-ORDEN del árbol y muestra el resultado (lista de IDs separados por guiones) sin usar un MessageBox.			1
MOSTRAR NODOS HOJAS				
14.	Se muestran los nodos que son hojas en el árbol.			1
MOSTRAR ALTURA				
15.	Se muestra la altura del árbol.			1
GENERALIDADES				
16.	Se utiliza un menú con cada una de las opciones.			1
17.	Se utiliza el modo gráfico para la inserción de nodos en el árbol.			1
18.	Se utiliza el modo gráfico para la eliminación de nodos del árbol.			1
19.	Se utiliza el modo gráfico para la búsqueda de un nodo en el árbol.			1
20.	Se utiliza el modo gráfico para mostrar los nodos hojas del árbol.			1
21.	Se utiliza el modo gráfico para mostrar la altura1 del árbol.			1
22.	No utiliza cuadros de diálogo tipo MessageBox para dar ni para solicitar información al usuario, a menos que sean excepciones del programa o que el usuario omitió algo necesario para la operación que quiere realizar.			1
23.	Cada una de las funciones se pueden realizar en cualquier momento y en cualquier orden.			1
24.	No se limpia o reinicia el árbol binario en ningún momento, a menos que sea utilizando la función de Eliminación (para todos los nodos) o al detener la ejecución del programa.			1