**Manual Técnico**

**INTRODUCCION:**

El programa de representación gráfica de árboles binarios busca exponer de forma sencilla el funcionamiento de estas estructuras de datos tan fundamentales. Para ingresar un árbol binario, existen 4 distintas alternativas. Ya sea introduciendo los nodos separados por comas o a partir de sus recorridos inorden, preorden o posorden. Además, también se creó la opción de generar árboles aleatorios.

**DESARROLLO:**

Esta aplicación fue desarrollada principalmente en Java con el IDE NetBeans, para sentar las bases operativas de la misma, se crearon cinco clases distintas con diferentes funcionalidades las cuales son **Interfaz, ArbolBB, NodoBinario, SimuladorArbolBinario, ArbolExpresionGrafico**. Para un mejor entendimiento vamos a definir cada una:

* **Clase Interfaz:** es la clase principal de nuestro programa, se trata de un archivo JFrame perteneciente a JavaFX. A través de esta clase se interactúa con el programa, permitiendo al usuario la creación de árboles binarios de diferentes maneras además de la visualización gráfica del mismo e información adicional sobre nodos o recorridos del árbol.
* **Clase ArbolBB:** esta clase es la que sirve para manejar el árbol binario que funciona como el mapa que indica de qué manera se dibuja el árbol en la vista del programa. Contiene los métodos que recorren el árbol como una lista doblemente ligada, además de algunos otros que permiten la consulta de información de un nodo determinado
* **Clase ArbolExpresionGrafico:** esta clase es una extensión de JPanel, y es la que se encarga de dibujar el árbol mediante de Graphics2D de JavaFX. Funciona utilizando métodos que analizan la estructura de la clase ArbolBB correspondiente al árbol que se está intentando construir.
* **Clase SimuladorArbolBinario:** esta clase funciona como un controlador, un enlace entre el arbolBB y el ArbolExpresionGrafico para permitir que la clase interfaz pueda representar dinámicamente los diferentes árboles que se vayan ingresando en el programa.
* **Clase NodoBinario:** la clase NodoBinario, es la que permite la creación de los nodos que componen el árbol binario a representar. Sus atributos son simplemente liga izquierda, liga derecha, dato y padre. Este último se utilizará principalmente para almacenar los padres de un nodo dado para facilitar su seguimiento.

Al comenzar la aplicación, el programa primero propone la creación de un árbol binario con condiciones específicas que dependen de la elección del usuario por medio de los botones respectivos en los que se puede escoger la modalidad de creación o bien generar un árbol aleatorio.

Por medio de un menú desplegable es posible elegir de qué manera se quiere ingresar el árbol binario, las distintas maneras que ofrece el programa se almacenan en un objeto de Java Swing llamado JComboBox. Cuando se presiona el botón de insertar, el programa se fija en el índice que se encuentra seleccionado en este menú desplegable y a partir de ahí genera un árbol de una manera distinta.

**Árbol mediante nodos completos**

Esta opción le permite al usuario ingresar la estructura del árbol binario al escribir los nodos que lo conformarán separados estos por comas. Una vez el usuario elige la opción e ingresa la hilera con la información solicitada, el programa se encarga de separar los nodos mediante el método .split y los va ingresando en orden dentro del árbol binario y el controlador del mismo.

**Árbol mediante recorridos**

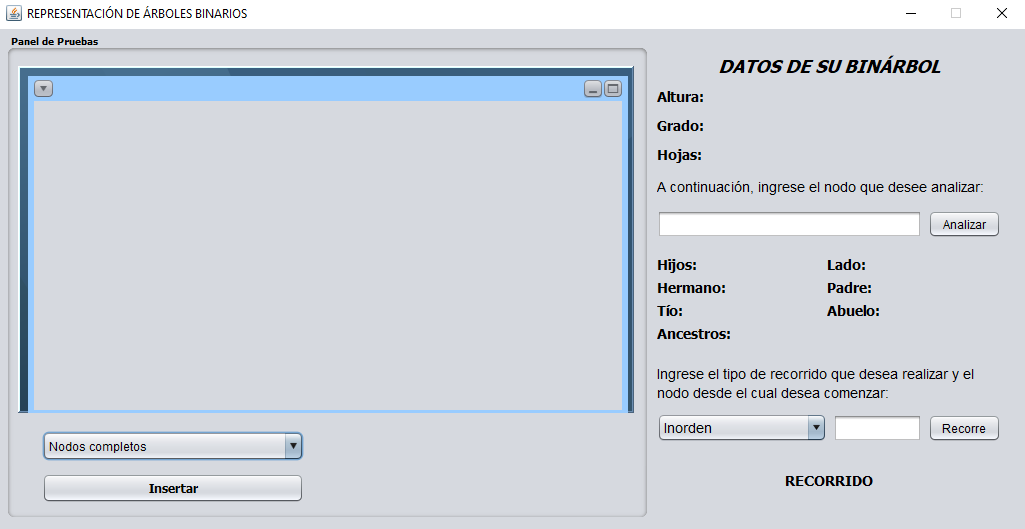
Tanto para la opción 2 como la opción 3, que consisten n la construcción del árbol binario por medio de dos recorridos dados. En primer lugar, se recibe el recorrido InOrden del árbol seguido del recorrido PreOrden. Una vez se tienen ambos recorridos, el programa almacena ambos en listas de nodos. Luego construye una matriz auxiliar que facilita la construcción del árbol. El recorrido preorden representará las filas de la matriz mientras que el recorrido inorden representará las columnas, luego donde el nodo de la columna sea igual al nodo de la fila se pondrá el número uno. A partir de esta matriz se hace muy sencillo construir el árbol binario, porque bastará con recorrer la matriz normalmente y se van ingresando los nodos cuyos valores en la matriz sea uno. Para la opción tres, el proceso es prácticamente idéntico, sólo varía que para construir el árbol se debe recorrer la matriz auxiliar de abajo hacia arriba debido a la definición de recorrido PosOrden.

**Árbol aleatorio**

Realmente no hay ninguna complicación en esta opción en particular, debido a que basta con generar un número aleatorio que representará la cantidad de nodos a crear y una vez hecho esto sólo hay que generar números aleatorios que serán los nodos que conformen este árbol binario.

**FUNCIONAMIENTO:**

Al ejecutar el programa tenemos esta interfaz de usuario. En primer lugar, está la opción de insertar un nuevo árbol y al costado de la ventana se puede visualizar la información general del árbol, así como la información específica de un nodo determinado. Finalmente, en el costado inferior derecho se pueden mostrar los distintos tipos de recorridos que se le pueden realizar al árbol binario.

****