**Manual Técnico**

**INTRODUCCION:**

Mi programa esta creado con el fin de crear arboles genealógicos teniendo en cuenta la estructura de arboles aprendida en la materia Lógica y Representación 3. Este programa le permite al usuario crear el árbol paso por paso de forma dinámica y poder ver en cualquier momento la información de cualquier nodo que desee.

**DESARROLLO:**

Esta aplicación fue desarrollada principalmente en Java con el Apache Netbeans IDE 12.6, para sentar las bases operativas de la misma se crearon cinco clases fundamentales, dos JFrames y un panel de fondo que se encargaran de se la interfaz grafica amigable con el usuario y la que maneje toda la lógica del programa, estas se llaman de la siguiente manera: **Arbol, ArbolExpresionGrafica, Nodo, Persona, Simulador, ViewArbol, ViewPersona, PanelFondo.** Para un mejor entendimiento vamos a definir cada una:

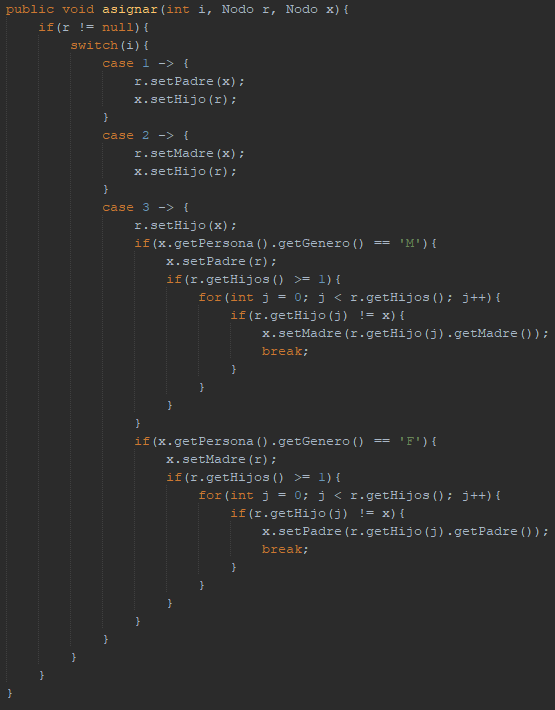
* **Clase Árbol:** esta clase es la que se encarga de crear el árbol principal, su único atributo es un nodo raíz del cual se puede recorrer todo el árbol, y sus métodos mas importantes son el de asignar, que recibe dos nodos, y el dato, es decir, si un nodo va a ser padre, madre o hijo del otro. También un método para recorrer el árbol, para buscar un Nodo en especifico y contar las personas de cada tipo.
* **Clase Nodo:** Los nodos son las representaciones de las personas, es decir, estos tienen un dato persona. Además, cuentan con una liga hacia el nodo Padre, una liga hacia el nodo Madre y una lista de hijos, ya que una persona puede no tener hijos, tener uno solo o varios, por lo que es necesario juntarlos en un arreglo para su posterior implementación.
* **Clase Persona:** Son las personas que estarán ingresadas en cada nodo del árbol, y guardan toda la información requerida de una persona, como nombre, apellido, genero, edad, ciudad, fecha de nacimiento, fecha de fallecimiento (si la tiene).
* **Clase Simulador:** La clase simulador va a crear una copia del árbol genealógico, y esta será la encargada de manejar la vista del árbol, insertando los datos necesarios al árbol.
* **Clase ArbolExpresionGrafico:** Esta clase es la encargada de pintar el árbol en pantalla con métodos como calcular Posición, encargados de verificar las posiciones correspondientes de cada nodo, para evitar que uno no tape a otro, o el método de dibujar árbol, que dibuja un nodo nuevo cada que se añade al árbol, para mostrarlo completamente.

Ahora veamos el nombre de las vistas del programa:

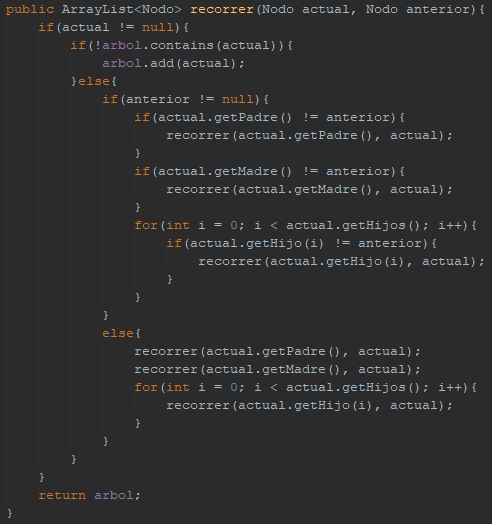
* **JFrame ViewArbol:** es el JFrame con la vista principal del programa, cuenta con todos los botones necesarios para que el programa funcione y metodos para crear el árbol, añadir, seleccionar, buscar, asignar y borrar nodos, entre otros.
* **JFrame ViewPersona:** es el JFrame que muestra toda la información de la persona que hemos seleccionado, el nombre, apellido, genero, ciudad, edad, fecha de nacimiento y fecha de fallecimiento.

**CLASE ARBOL:**

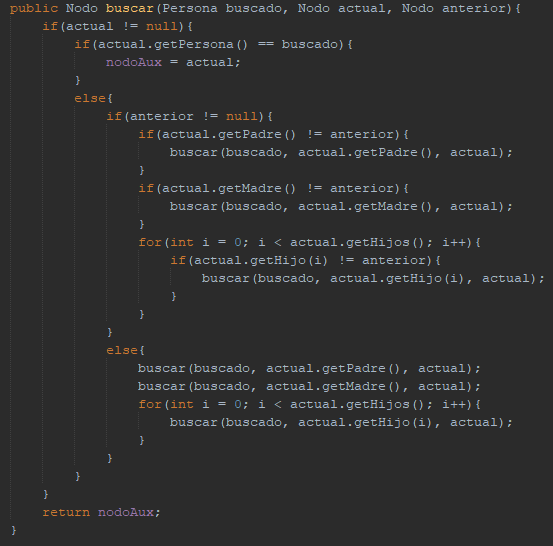
La clase árbol es la encargada de asignar las relaciones entre nodos, recorrer el árbol, buscar y contar personas, ahora veamos algunos de los métodos mas importantes de la clase.



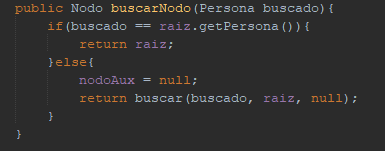
Para el caso 1, asigna el padre a un nodo y su hijo respectivamente, para el caso 2, asigna la madre a un nodo y su hijo respectivamente, y para el caso 3, le asigna el hijo a alguien, es necesario dividirlo para cuando le queremos poner un hijo a una mujer o un hombre, puesto que le debemos asignar como padre o como madre.



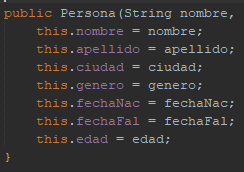
El método recorrer, pasa por todo el árbol y va añadiendo el recorrido a una lista, para tenerla de referencia mas adelante.



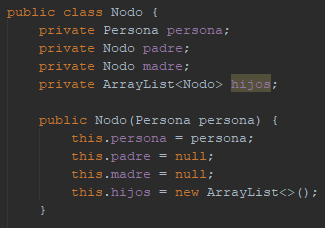
El método buscar recibe el nodo que estamos buscando, el nodo principal, que casi siempre va a ser la raíz del árbol y el nodo anterior, cuando se llama por primera vez el nodo actual es la raíz y el nodo anterior es nulo puesto que no hay anterior, pero a medida que se va trabajando con la recursión, estos valores cambian, cuando encuentra el nodo que buscamos, lo inicia como un auxiliar y al finalizar el recorrido lo retorna.



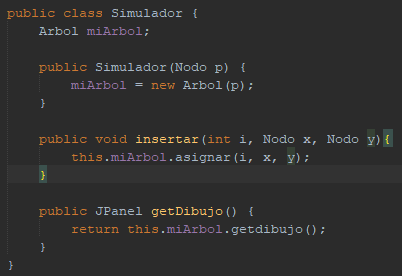
La primera parte del método buscar, necesita verificar que el que busquemos no sea la raíz, y que el nodo auxiliar que vamos a cambiar sea nulo, luego llama la primera vez del método buscar con los datos iniciales ya mencionados, el nodo buscado, la raíz del árbol y el nodo anterior nulo.

**CLASE PERSONA:**

La clase que guarda toda la información de las personas, como se puede ver en la foto, tambien tiene los metodos getters y setters y un método que me dice si la persona ha fallecido o no, revisando que el string fechaFal no sea nulo.

**CLASE PERSONA:**

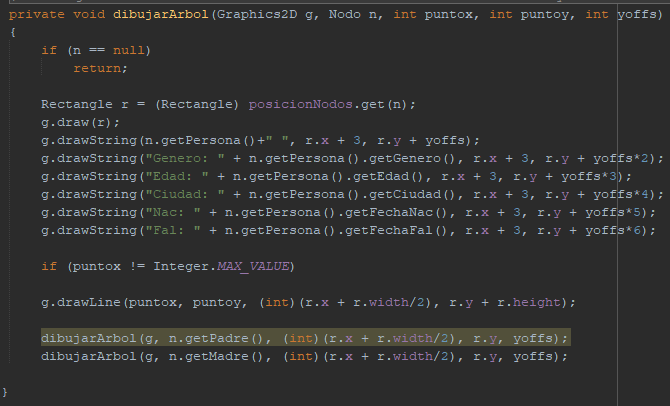
La clase que guarda en un nodo, una persona especifica, el nodo padre, el nodo madre y una lista que contenga todos los hijos de esta persona en específico, al iniciar, estos valores son declarados como nulos puesto que necesitan de métodos externos para su inicialización.

**CLASE PERSONA**:

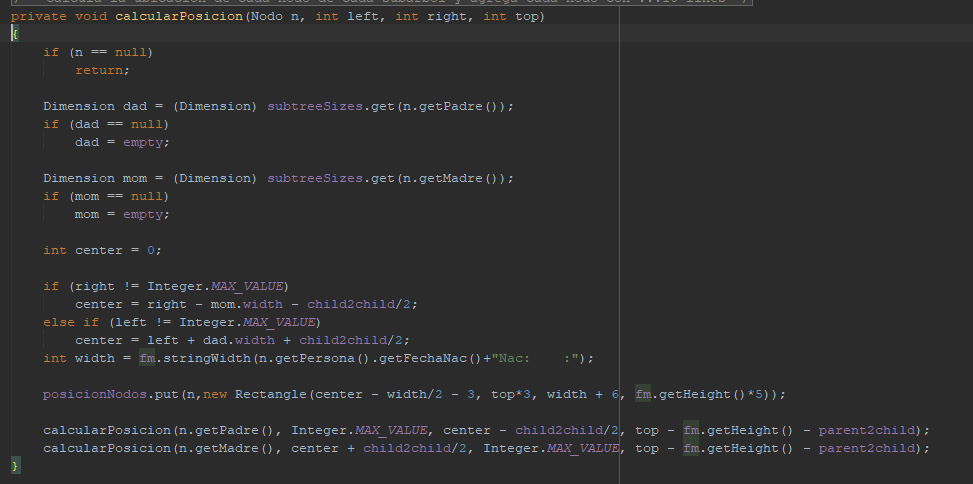
Esta clase crea una simulación de nuestro árbol genealógico, este árbol copia es el que será utilizado para pintar en pantalla nuestro árbol principal, es necesario crear uno nuevo para evitar fallas, y dañar el árbol base.

**CLASE ARBOLEXPRESIONGRAFICO**

Esta clase tiene los metodos para pintar el árbol en pantalla, los cuales se ven de la siguiente forma:

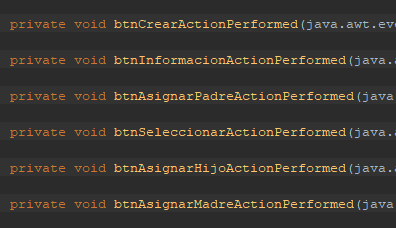


Este método crea un rectángulo que es la representación gráfica del nodo persona, luego dibuja strings en pantalla con la información que queremos que aparezca dentro de cada nodo, en este caso, el nombre completo, genero, edad, ciudad, fecha de nacimiento y fecha de fallecimiento. Luego tenemos el método drawLine, que es el que pinta las líneas de conexión entre los nodos y por ultimo esta la llamada al método recursivo para dibujar el árbol completo.



También tenemos el método de calcular Posición, encargado de evitar que los nodos vayan a ponerse unos encima de otros, que la información sea tan larga que tapen otro nodo, o que no se vea en pantalla porque esta mal posicionado. Cambiando la posición relativa de los nodos con respecto a los padres.

**JFRAME VIEWARBOL**

Esta clase es la que nos permite trabajar con la vista del programa, por lo tanto no me voy a centrar en los métodos ya que he mencionado sus contrapartes anteriormente. Pero es necesario saber que en esta clase es donde se controlan todas las excepciones, es decir, que verifica que los datos ingresados sean validos para que el programa no tenga errores.