**Manual Técnico**

**INTRODUCCION:**

Mi programa esta creado con el fin de Resolver sistemas de ecuaciones lineales, en este caso, de tres variables por tres incógnitas. Con el fin de que este permita que el usuario pueda trabajar los sistemas de ecuaciones como el quiera, y mostrarle el paso a paso de la solución del sistema, ya sean resueltas por el método de Gauss Jordan o el de Cramer

**DESARROLLO:**

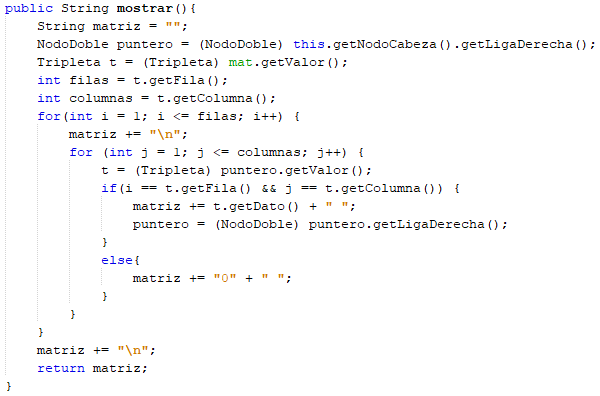
Esta aplicación fue desarrollada principalmente en Java con el Apache Netbeans IDE 12.6, para sentar las bases operativas de la misma se crearon tres clases fundamentales y un JFrame que se encargara de ser la interfaz grafica con el usuario y la que maneje toda la lógica del programa, estas se llaman de la siguiente manera: **MatrizForma2, Tripleta, NodoDoble, Grafica**. Para un mejor entendimiento vamos a definir cada una:

* **Clase MatrizForma2:** esta clase es la que sirve para manejar la matriz dispersa que funciona como la matriz inicial, de la cual se van a hacer las operaciones hasta que se resuelva el sistema de ecuaciones, Contiene los métodos que recorren la matriz como lista ligada además de los que construyen la matriz para que esté representada de la forma 2 vista en clase con anterioridad.
* **Clase Tripleta:** en la clase tripleta se crean los datos que van a ser almacenados en los nodos dobles que conforman a las matrices en forma 2 que se utilizan para representar las coordenadas, por así decirlo, de los datos dentro de la matriz. Tal como su nombre indica, esta clase se compone de tres atributos que significan respectivamente, la fila, la columna y el valor contenido en una coordenada especifica de la matriz.
* **Clase NodoDoble:** la clase nodo doble, es la que permite la creación de los nodos que componen las matrices en forma 2. Sus atributos son simplemente liga izquierda, liga derecha y dato. Este ultimo se utilizará principalmente para almacenar las tripletas de las coordenadas de la matriz dispersa que no sean iguales a cero.
* **Clase Grafica:** es la clase principal en la que nos encontramos con una interfaz que se ejecuta en pantalla para saber cuales son las acciones que desea realizar el usuario, pero lo mas importante de esta clase es que cuenta con los metodos resolverGauss y resolverCramer, que son la base principal del programa para que pueda funcionar.

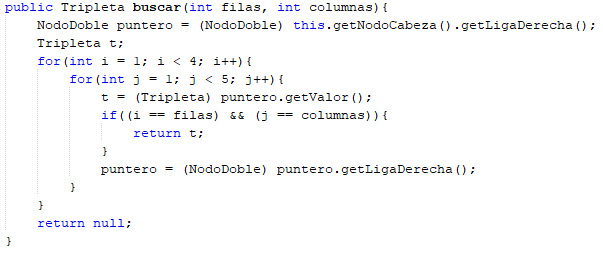
Ahora, voy a mencionar algunas partes del código que sean las mas importantes del programa, cabe aclarar que entre estas no van a estar las generales de las clases MatrizForma2, Tripleta y NodoDoble, puesto que estas las vimos en clase y aunque son importantes para el programa, ya todos las hemos trabajado.

**CLASE MATRIZFORMA2:**

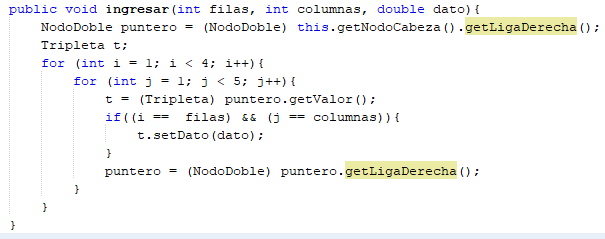
En la clase MatrizForma2 nos encontramos con tres metodos novedosos e importantes para la realización de este programa:

****

Este método, como su nombre lo dice, se encarga de crear un String con la matriz completa, que al momento de ser impreso en consola o agregado a un TextBox, va a mostrar la matriz que se haya invocado transformada de una MatrizForma2, a una matriz normal para poder verificar de una forma más cómoda los datos ingresados. Se basa en recorrer la matriz en Forma2 con una tripleta puntero que se usa para recorrerla, de ahí imprimimos el dato que este guardando, así hasta que se complete la matriz, como sabemos, la forma 2 no guarda los valores con cero, por lo que si no encuentra un valor dentro de la MatrizForma2, imprime un cero.



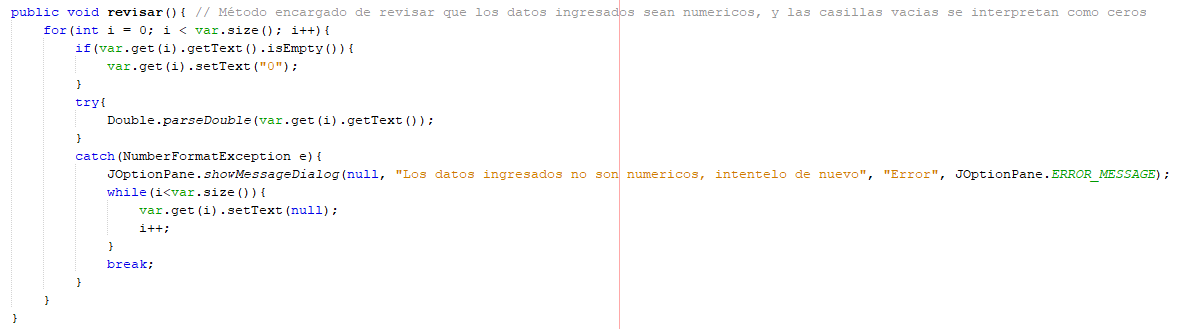
Como hubo varios problemas para el manejo de índices en las matrices de Forma 2, me tomé la tarea de crear un método de búsqueda que nos devuelve una tripleta, este recibe las filas y columnas, con esta información va a recorrer una matrizForma2 hasta que encuentre alguna tripleta con estos valores, y la devuelve, también puede ocurrir el caso en el que no exista dicha tripleta, por lo que devuelve null dando a entender que no existe esa tripleta en la matriz.



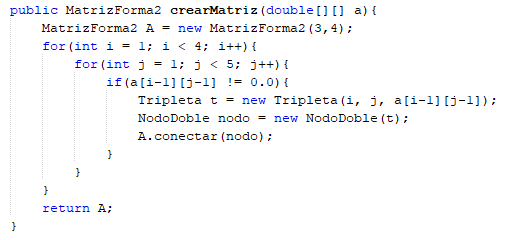
Ahora, como al momento de querer reescribir los datos que tiene una tripleta es bastante complejo, cree este método ingresar, NO es para ingresar una nueva tripleta a la matriz, sino que es para sobrescribir la información en una tripleta, es decir, cambiar el dato por uno existente.

**CLASE GRAFICA:**

En esta clase existen varios métodos muy importantes para la realización del programa, voy a tratar de nombrar todos:



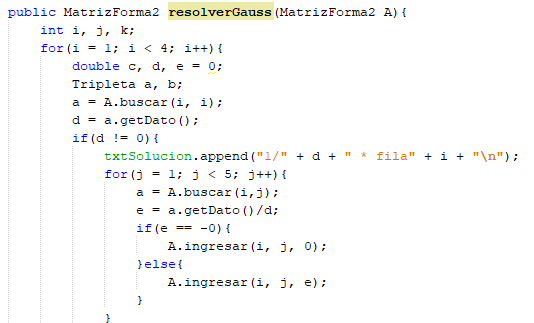
Revisar() se encarga de verificar que los datos que hayan en los textBox sean numéricos, si no lo son lanza una excepción y limpia todos los datos, dejando que el usuario vuelva a ingresarlos de una forma valida, además, si el usuario deja un campo vacío lo convierte en un cero, para facilitar su ingreso de información



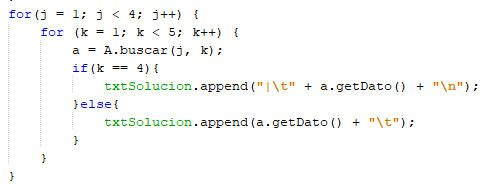
Se encarga de tomar una matriz normal, que fue creada a partir de la lectura de los textBox, y con esta matriz vamos llenando una MatrizForma2 A, conectando todos los nodos en donde los datos sean diferentes de cero. Este método devuelve la matriz A que es la auxiliar con la que se trabaja el resto del programa.

**Método resolverGauss():**

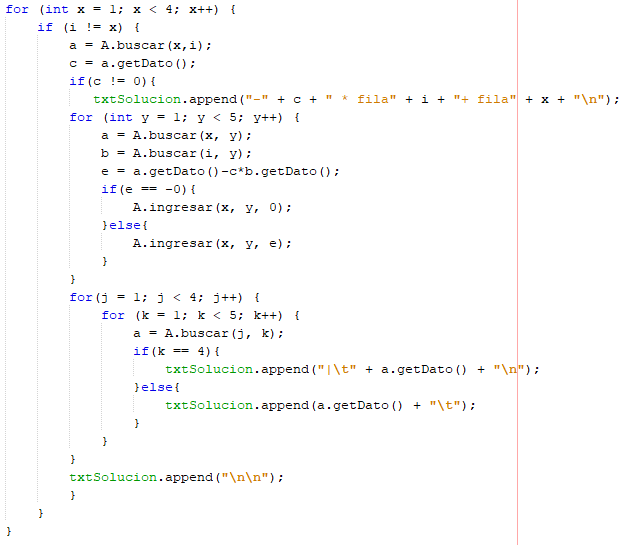
Como este método es bastante extenso, lo voy a mostrar por partes explicando que hace cada cosa en el mismo.

****

La primera parte del método cuenta con la generación de variables, un for encargado de recorrer la matriz por filas, tripletas, double, entero, etc. Lo importante en esta parte de código es el primer if, el cual hace que el primer pivote se divida por si mismo, para poder hacer del primer dato de la diagonal un 1, este hace que toda la fila se divida por el primer dato, y usa el método ingresar para reescribir la información de la matriz.

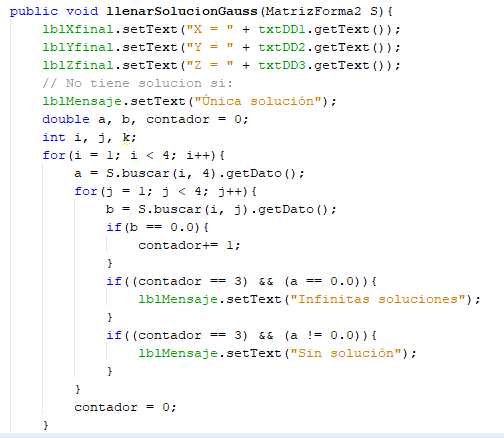
****

Luego aparece este ciclo cuya función es imprimir la matriz como se encuentra en este momento, para así hacer que cada que se haga una operación, esta se muestre y el usuario pueda ver el paso a paso del sistema de ecuaciones.

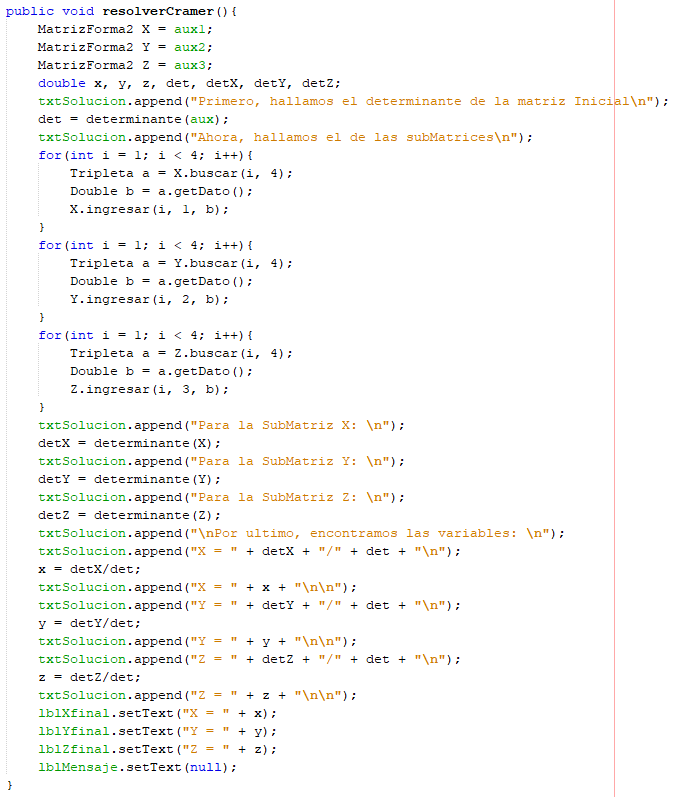
****

Y por ultimo, tenemos esta parte que es la encargada de multiplicar una fila por un escalar y sumarla a otra fila, primero verifica que el escalar no sea cero ya que si es cero, no hace falta trabajarlo, luego hace las operaciones en el primer for importante, y ya después tenemos otro ciclo igual al del pantallazo anterior, cuya función es imprimir la matriz después de estos pasos

Este método se trabaja hasta que la matriz encuentre una solución, infinitas soluciones o no llegue a una solución valida.



Ahora, tenemos el método llenarSolucionesGauss, cuya función es tomar todos los datos de la matriz solucionada e imprimirlos en los labeles del JFrame, ademas de sobreescribir el mensaje alternartivo con la información de la solución, es decir, si el sistema de ecuaciones tiene una solución, infinitas soluciones o no tiene solución.



Por ultimo tenemos el método de Cramer, el cual se basa en la creación de tres matrices alternativas que son iguales a la auxiliar, y en cada una de estas cambiamos los valores de la fila correspondiente, por ejemplo, en la matriz auxiliar X, se cambiar los valores de la columna 1 por los de la 4, es decir, los valores que tienen la variable x por los valores de la fila de solución,

Luego se encuentran los determinantes de dichas matrices y la matriz principal y por ultimo nos queda la solución, que es el determinante de la matriz alternativa dividió el determinante principal, Hay muchos labeles y txt para mostrar en el recuadro del centro todos los pasos que utilizamos para llegar a la solución del sistema.