



UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERIA.
Análisis Numérico ciclo I/2011

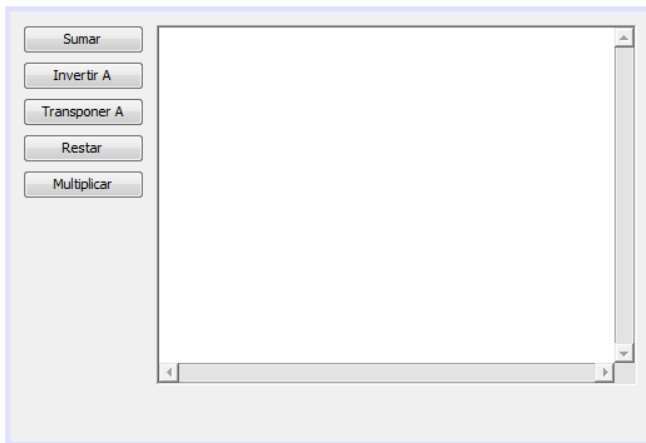
Docente: Ing. Carlos Arturo Ruano
Instructor: Br. Oscar G. Rodríguez

IMPLEMENTACION CLASE MATRIZ

Ahora ya terminada la clase matriz haremos una implementación simple con entorno grafico.

Procedimiento:

1. Primero cree en netbeans un nuevo del tipo Aplicación java. Y en en agregue una nueva clase del tipo JFrame.
2. A nuestro nuevo proyecto importe las librerías de Fraccion y Matriz como se hizo en la guía 2.
3. Ahora diseñe el JFrame de la siguiente forma:



4. Ahora cree como variables globales en el JFrame dos matrices.

```
Matriz A;  
Matriz B;
```



Guía 4

Análisis Numérico

Implementación Matriz

5. Proceda a llenar manualmente en el constructor las matrices de la siguiente forma:

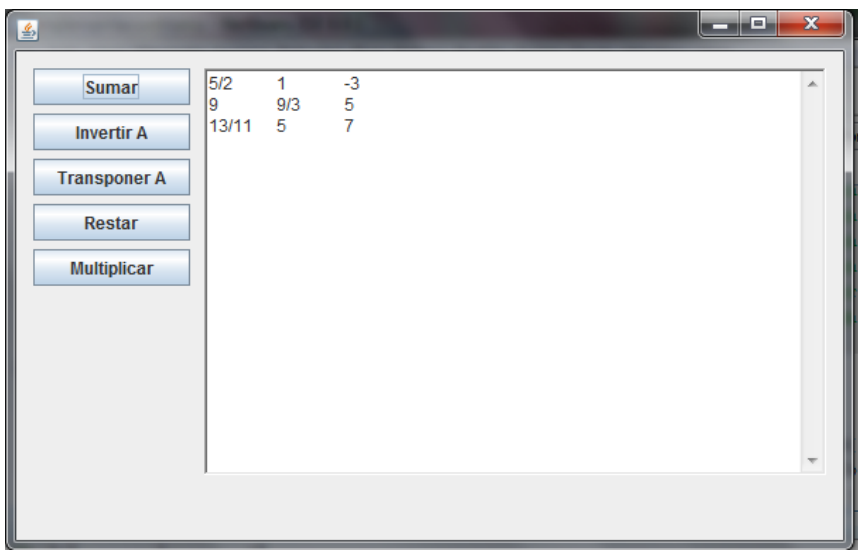
```
A.setPos(Fraccion.parseFraccion("5/2"), 0, 0);
A.setPos(Fraccion.parseFraccion("1"), 0, 1);
A.setPos(Fraccion.parseFraccion("-3"), 0, 2);
A.setPos(Fraccion.parseFraccion("9"), 1, 0);
A.setPos(Fraccion.parseFraccion("9/3"), 1, 1);
A.setPos(Fraccion.parseFraccion("5"), 1, 2);
A.setPos(Fraccion.parseFraccion("13/11"), 2, 0);
A.setPos(Fraccion.parseFraccion("5"), 2, 1);
A.setPos(Fraccion.parseFraccion("7"), 2, 2);
```

De igual forma llene la matriz B (con valores diferentes).

6. Cree un método imprimir en con el cual imprimiremos una matriz cada vez que sea necesario.

```
141 private void imprimir(Matriz a) {
142     String salida = "";
143     for (int i = 0; i < a.getFilas(); i++) {
144         for (int j = 0; j < a.getColumnas(); j++) {
145             salida += a.getPos(i, j) + "\t";
146             System.out.println(salida + "\n\n\n");
147         }
148         salida += "\n";
149     }
150     txtSalida.setText(salida);
151 }
152 }
153
```

Si imprimimos a nos quedara de la siguiente forma:





Guía 4
Análisis Numérico
Implementación Matriz

7. Ahora cree un `ActionPerformed` para el botón sumar en el cual sumara $A + B$ y luego imprimirá el resultado en el `textArea`.

```
127 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
128     try {  
129         imprimir(Matriz.sumar(A, B));  
130     } catch (MatrizNoOperableException ex) {  
131         lblError.setText(ex.toString());  
132     }  
133 }
```

En la línea 131 lo que hacemos es mostrar el error (en caso que surja) en nuestra ventana.

8. Cree los eventos para los botones restar, transponer, invertir y multiplicar.
9. Pruebe el proyecto con matrices en las que usted crea que pueda surgir un error en la operación por ejemplo sume dos matrices de distinta dimencion y vea como se muestra el error.