



# INSTITUTO TECNOLÓGICO DE CERRO AZUL



**CARRERA:** ING. SISTEMAS COMPUTACIONALES

**MATERIA:** SIMULACION

**SEMESTRE:** 4

**INTEGRANTES:**

MIGUEL ÁNGEL SEGURA BRAVO – 17500681  
GONZALEZ HERNANDEZ FRANCISCO JAVIER –  
17500660

**Trabajo:**

NÚMEROS PSEUDO ALEATORIOS

**Docente**

**MCTE ALICIA MAGDALENA BRIDAT CRUZ**

## Contenido

INTRODUCCIÓN .....	3
ALGORITMO DE CUADRADOS MEDIOS .....	4
ALGORITMO DE PRODUCTOS MEDIOS .....	13
ALGORITMO DE MULTIPLICADOR CONSTANTE .....	23
ALGORITMO LINEAL .....	33
ALGORITMO LINEAL (BANKS, CARSON, NELSON, NICO).....	43
ALGORITMO CONGRUENCIAL ADICTIVO.....	53
ALGORITMO CONGRUENCIAL NO LINEAL .....	65
CONCLUSIONES .....	76
BIBLIOGRAFÍA .....	77

## INTRODUCCIÓN

En este proyecto, se llevó a cabo la programación de 7 diferentes programas que sean capaces de generar números pseudoaleatorios, cada uno de ellos los genera de forma distinta, estos números pseudoaleatorios son muy importantes para los cálculos que haremos posteriormente en la materia de simulación.

Estos programas están basados en algoritmos que ya llevan tiempo de ser creados, los algoritmos los desarrollamos mediante su fórmula en clase, para verificar manualmente la correcta obtención de los números pseudoaleatorios.

Nuestro trabajo como programadores, fue el crear los programas de cada uno de ellos, la lógica de los programas, estaba determinada por la fórmula y el comportamiento que tiene cada algoritmo. Desde el número de variables (parámetros), si es que manejará alguna constante, y que operaciones se realizará cada uno de los programas, para posteriormente mostrar el resultado en pantalla.

Decidimos crear una Interfaz gráfica para el usuario (GUI), mediante la cual le será muy fácil generar números pseudoaleatorios según el algoritmo deseado. Creamos la interfaz ya que resulta mucho más fácil manejarla, que el interactuar con la consola de nuestro IDE, en la interfaz solo ingresamos los datos en cada uno de los campos, y al pulsar un botón, este generará la impresión de los números pseudoaleatorios.

Todos los programas están guardados en un mismo paquete, y creamos un menú que almacena todos los programas que creamos, al dar un botón, nos permitirá ir al algoritmo que deseamos utilizar, y cada programa de algoritmo, tiene un botón para regresar, ya sea al menú o para finalizar el programa.



## ALGORITMO DE CUADRADOS MEDIOS

### DESCRIPCIÓN

Este algoritmo no congruencial fue propuesto en la década de los cuarenta por Von Newman y Metrópolis. Requiere un número entero detonador (llamado semilla) con D dígitos, el cual es elevado al cuadrado para seleccionar del resultado los D dígitos del centro, el primer número  $r_i$  se determina simplemente anteponiendo el 0. a estos dígitos. Para obtener el segundo  $r_i$  se sigue el mismo procedimiento, solo que ahora se elevan al cuadrado los d dígitos del centro que se seleccionaron para obtener el primer  $r_i$ . Este método se repite hasta obtener n números  $r_i$ .

### OBJETIVO

Desarrollar un programa que sea capaz de generar números pseudoaleatorios utilizando el algoritmo de cuadros medios, basándose en un formula y en el comportamiento del algoritmo, conociendo como variables la semilla inicial ( $X_0$ ) y la cantidad de números que se esperan generar (n).

Programando la interfaz gráfica, mediante el uso de textfields, textareas y botones, en los primeros obtendremos el valor de la semilla y la cantidad de números que se buscan generar, en los segundos imprimiremos los resultados, y en los últimos habrá 4 tipos diferentes de botones, para salir del programa, para volver al menú, para generar los números y para limpiar, es decir borrar lo última ejecución, y poder volver a usar el programa.

### DESARROLLO

Generar los primeros 5 números  $r_i$  a partir de una semilla  $X_0 = 5735$ , de donde se puede

Observar que  $D = 4$  dígitos.

Solución

$$Y_0 = (5735)^2 = 32890225 \quad X_1 = 8902 \quad r_1 = 0.8902$$

$$Y_1 = (8902)^2 = 79245604 \quad X_2 = 2456 \quad r_2 = 0.2456$$

$$Y_2 = (2456)^2 = 6031936 \quad X_3 = 0319 \quad r_3 = 0.0319$$

$$Y_3 = (0319)^2 = 101761 \quad X_4 = 0176 \quad r_4 = 0.0176$$

$$Y_4 = (0176)^2 = 30976 \quad X_5 = 3097 \quad r_5 = 0.3097$$

### CÓDIGO

```
package GNPpseudoAleatorios;  
import javax.swing.JOptionPane;
```

```
public class CuadrosMedios extends javax.swing.JFrame {  
  
    public CuadrosMedios() {  
        initComponents();  
        this.setLocationRelativeTo(null);  
    }  
}
```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    Exit = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    back = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    Semilla = new javax.swing.JTextField();
    Generar = new javax.swing.JButton();
    jLabel3 = new javax.swing.JLabel();
    R = new javax.swing.JTextField();
    jScrollPane1 = new javax.swing.JScrollPane();
    ri = new javax.swing.JTextPane();
    jScrollPane2 = new javax.swing.JScrollPane();
    xi = new javax.swing.JTextPane();
    Clear = new javax.swing.JButton();
    jScrollPane3 = new javax.swing.JScrollPane();
    y1 = new javax.swing.JTextPane();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jPanel1.setBackground(new java.awt.Color(153, 255, 204));
    jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    Exit.setBackground(new java.awt.Color(255, 0, 0));
    Exit.setText("Salir");
    Exit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ExitActionPerformed(evt);
        }
    });
    jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
    jLabel2.setText("Algoritmo de Cuadros Medios");

```

```

jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(150,
50, -1, 30));

back.setBackground(new java.awt.Color(255, 255, 51));
back.setText("Volver");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setText("Ingrese la cantidad de ri a generar:");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
140, -1, -1));
jPanel1.add(Semilla, new org.netbeans.lib.awtextra.AbsoluteConstraints(310,
110, 220, -1));

Generar.setText("Generar");
Generar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        GenerarActionPerformed(evt);
    }
});
jPanel1.add(Generar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 390, -1, -1));

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel3.setText("Ingrese la semilla:");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
110, -1, -1));
jPanel1.add(R, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 140,
220, -1));

jScrollPane1.setViewportView(ri);

jPanel1.add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(480, 180, 120, 180));

jScrollPane2.setViewportView(xi);

jPanel1.add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 180, 120, 180));

Clear.setText("Limpiar");

```

```

        Clear.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ClearActionPerformed(evt);
            }
        });
jPanel1.add(Clear, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
390, -1, -1));

jScrollPane3.setViewportView(y1);

jPanel1.add(jScrollPane3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 180, 120, 180));

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
20, 20, 450));

jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/5.jpg"))
); // NOI18N
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

pack();

```

```

} // </editor-fold>

private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {
    // Verifica si las cajas estan vacias
    if (evt.getSource()==Generar)
    {

        if(Semilla.getText().equals("")) //***** Permite verificar si la caja de texto
        esta vacia *****//
        {
            JOptionPane.showMessageDialog(null,"Ingresa valor de semilla");
        }

        if(R.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingresa la cantidad de ri a generar:");
        }

        //Declaracion de las variables//

        String semilla=Semilla.getText();
        Integer Ri= new Integer(R.getText());

        String snumero2, snumero3,cadenaxi = "",cadenari = "",cadenay1 = "";
        int tam1, tam2, i, primerc;
        long numero1, numero2;
        tam1 = semilla.length();
        numero1 = Integer.parseInt(semilla);

        //Permite que la semilla no sea menor a tres digitos, en caso de que se
        cumpla la condicion se arrojava un mensaje que el valor de //
        //de xi sea mayor a tres digitos//
        if(numero1<=999)
        {
            JOptionPane.showMessageDialog(null,"Valor de Xi debe ser mayor a tres
            digitos");
            Semilla.setText("");
            R.setText("");
            ri.setText("");
            xi.setText("");
            y1.setText("");

        }
        try
        {

```



```
//Permite realizar el Algoritmo de cuadrados medios en caso de que la semilla  
sea mayor a tres numeros//
```

```
if(numero1>=999){
```

```
    for (i=1; i<=Ri; i++){
```

```
        numero2 = (long)Math.pow(numero1,2);
```

```
        snumero2 = Long.toString(numero2);
```

```
        tam2 = snumero2.length();
```

```
        primerc = (tam2-tam1) / 2;
```

```
        snumero3 = snumero2.substring(primerc,primerc+tam1);
```

```
        cadenaxi+=("x"+i+ " = " +snumero3+" \n");
```

```
        cadenari+=( "r"+i+ " = 0. " + snumero3+ " \n");
```

```
        cadenay1+=( "y"+i+ " = " + snumero2+ " \n");
```

```
        numero1 = Integer.parseInt(snumero3);
```

```
    }
```

```
    xi.setText(cadenaxi);
```

```
    ri.setText(cadenari);
```

```
    y1.setText(cadenay1);
```

```
    }
```

```
    } catch(java.lang.NumberFormatException e)
```

```
{
```

```
    //En caso de que sea invalido un dato ingresado se limpiara todas las  
casillas//
```

```
    JOptionPane.showMessageDialog(null,"Dato invalido");
```

```
    Semilla.setText("");
```

```
    R.setText("");
```

```
    ri.setText("");
```

```
    xi.setText("");
```

```
    y1.setText("");
```

```
    }
```

```
    }
```

```
}
```

```
}
```

```
private void backActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    //Permite regrear al menu//
```

```
    Menú Menú = new Menú();
```

```
    Menú.setVisible(true);
```

```

        CuadrosMedios.this.dispose();

    }

    private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
        // Permite salir del programa
        //
        System.exit(0);
    }

    private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        //Permite limpiar las casillas//
        Semilla.setText(null);
        R.setText(null);
        ri.setText(null);
        xi.setText(null);
        y1.setText(null);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(CuadrosMedios.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(CuadrosMedios.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(CuadrosMedios.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(CuadrosMedios.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
    }
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new CuadrosMedios().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JButton Exit;
private javax.swing.JButton Generar;
private javax.swing.JTextField R;
private javax.swing.JTextField Semilla;
private javax.swing.JButton back;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextPane ri;
private javax.swing.JTextPane xi;
private javax.swing.JTextPane y1;
// End of variables declaration
}

```

## EJEMPLO

**Algoritmo de Cuadros Medios**

Salir

Ingrese la semilla: 5735

Ingrese la cantidad de ri a generar: 5

y1 = 32890225	x1 = 8902	r1 = 0.8902
y2 = 79245604	x2 = 2456	r2 = 0.2456
y3 = 6031936	x3 = 0319	r3 = 0.0319
y4 = 101761	x4 = 0176	r4 = 0.0176
y5 = 30976	x5 = 3097	r5 = 0.3097

Generar Limpiar

Volver

## CONCLUSIONES

Gracias al uso de una interfaz gráfica, podemos demostrar que el usuario final, tiene una gran facilidad para obtener los números pseudoaleatorios que busca generar, a diferencia de que, si usará la consola sería ligeramente más complicado, además una GUI es más llamativo e interactivo con el usuario.

Como se muestra en la captura, en los textfields ingresamos el valor de la semilla  $X_0$  y la cantidad de números a generar, al pulsar el botón de generar, imprimirá los resultados de haber llevado las operaciones en cada una de las iteraciones, hasta generar los 5, son 3 diferentes cadenas que almacenan cada uno de los resultados de los textareas, el botón limpiar, deja todos los elementos en blanco.

Si comparamos los resultados, con los obtenidos en clase, notamos que son iguales, por lo tanto, nuestro programa tiene un funcionamiento correcto, y el resultado es el esperado.

## ALGORITMO DE PRODUCTOS MEDIOS

### DESCRIPCIÓN

La mecánica de generación de números pseudo aleatorios de este algoritmo no congruencial es similar a la del algoritmo de cuadrados medios. La diferencia entre ambos radica entre el algoritmo de productos medios requiere de dos semillas, ambos con D dígitos, además, en vez de elevarlo al cuadrado, la semilla se multiplica y del producto se selecciona los D dígitos del centro, los cuales formaran el primer número pseudo aleatorio  $r_1 = 0.D$  dígitos, después se elimina una semilla y la otra se multiplica por el primer número de D dígitos para luego seleccionar el producto de los D dígitos que conforman el segundo número  $r_2$ . Entonces se elimina la segunda semilla y se multiplican el primer número de D dígitos por el segundo de D dígitos; del producto se obtiene el tercer número  $r_3$ . Siempre se ira eliminando el número más antiguo y el procedimiento se repetirá hasta generar los n números pseudo aleatorios.

### OBJETIVO

Desarrollar un programa que sea capaz de generar números pseudoaleatorios, utilizando el algoritmo de productos medios, conociendo las variables, en este caso las dos semillas iniciales  $X_0$  y  $X_1$ , así como la cantidad de números que se busca generar, es importante conocer el funcionamiento de este algoritmo, ya que va sustituyendo semillas en cada iteración, según la nueva semilla resultante. Programar la interfaz gráfica correspondiente de este algoritmo, conociendo el uso de los textfields, textareas, buttons, entre otros componentes del entorno gráfico, para que el usuario final interactué de una manera más fácil con nuestro programa.

### DESARROLLO

Generar los primeros 5 números a partir de las semillas  $X_0 = 5015$  y  $X_1 = 5734$ ; observe

Que ambas semillas tienen  $D = 4$  dígitos.

Solución:

$$\begin{aligned} Y_0 &= (5015)(5734) = 28756010 \quad X_2 = 7560 \quad r_1 = 0.7560 \\ Y_1 &= (5734)(7560) = 43349040 \quad X_3 = 3490 \quad r_2 = 0.3490 \\ Y_2 &= (7560)(3490) = 26384400 \quad X_4 = 3844 \quad r_3 = 0.3844 \\ Y_3 &= (3490)(3844) = 13415560 \quad X_5 = 4155 \quad r_4 = 0.4155 \\ Y_4 &= (3844)(4155) = 15971820 \quad X_6 = 9718 \quad r_5 = 0.9718 \end{aligned}$$

### CÓDIGO

```
package GNPseudoAleatorios;  
  
import javax.swing.JOptionPane;
```

```

public class ProductosMedios extends javax.swing.JFrame {

    public ProductosMedios() {
        initComponents();
        this.setLocationRelativeTo(null);
    }
    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        Semilla2 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        Semilla1 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        R = new javax.swing.JTextField();
        jScrollPane2 = new javax.swing.JScrollPane();
        xi = new javax.swing.JTextPane();
        jScrollPane1 = new javax.swing.JScrollPane();
        ri = new javax.swing.JTextPane();
        Generar = new javax.swing.JButton();
        Clear = new javax.swing.JButton();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jScrollPane3 = new javax.swing.JScrollPane();
        y1 = new javax.swing.JTextPane();
        jLabel9 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        Exit = new javax.swing.JButton();
        back = new javax.swing.JButton();
        jLabel10 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel1.setBackground(new java.awt.Color(153, 255, 204));
        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel2.setText("Algoritmo de Productos Medios");
        jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
50, -1, 30));

```

```

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel4.setText("Ingrese la semilla número 1:");
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
100, -1, -1));
jPanel1.add(Semilla2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 140, 220, -1));

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel3.setText("Ingresa la semilla número 2:");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
140, -1, -1));
jPanel1.add(Semilla1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 100, 220, -1));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setText("Ingrese la cantidad de ri a generar:");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
180, -1, -1));
jPanel1.add(R, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 180,
220, -1));

jScrollPane2.setViewportView(xi);

jPanel1.add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 220, 120, 180));

jScrollPane1.setViewportView(ri);

jPanel1.add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(460, 220, 120, 180));

Generar.setText("Generar");
Generar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        GenerarActionPerformed(evt);
    }
});
jPanel1.add(Generar, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
430, -1, -1));

Clear.setText("Limpiar");
Clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClearActionPerformed(evt);
    }
});

```

```

jPanel1.add(Clear, new org.netbeans.lib.awtextra.AbsoluteConstraints(330,
430, -1, -1));

jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

jScrollPane3.setViewportViewView(y1);

jPanel1.add(jScrollPane3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 220, 120, 180));

jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20,
20, 450));

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

Exit.setBackground(new java.awt.Color(255, 0, 0));
Exit.setText("Salir");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});
jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

back.setBackground(new java.awt.Color(255, 255, 51));
back.setText("Volver");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});

```



```

    }
    });
    jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

    jLabel10.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/5.jpg"))
); // NOI18N
    jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

    getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

    pack();
} // </editor-fold>

private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {
    // Permite verificar que las cajas esten vacias
    if (evt.getSource()==Generar)
    {
        if(Semilla1.getText().equals("")) //***** Permite verificar si la caja de texto
esta vacia *****//
        {
            JOptionPane.showMessageDialog(null,"Ingresa valor de semilla 1");
        }
        if(Semilla2.getText().equals("")) //***** Permite verificar si la caja de texto
esta vacia *****//
        {
            JOptionPane.showMessageDialog(null,"Ingresa valor de semilla 2");
        }
        if(R.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingresa la cantidad de ri a generar:");
        }
    } //Declaracion de variables
    String semilla1=Semilla1.getText();
    String semilla2=Semilla2.getText();
    Integer Ri= new Integer(R.getText());

    String snumero2, snumero3, snumero4, cadenaxi = "",cadenari = "", cadenay1
= "";
    int tam1, tam3, i, primerc;
    long numero1, numero2, multi;
    tam1 = semilla1.length();
    numero1 = Integer.parseInt(semilla1);
    numero2 = Integer.parseInt(semilla2);

```

```

        //Permite que la semilla 1 no sea menor a 3 digitos
        if(numero1<=999)
        {
            JOptionPane.showMessageDialog(null,"Valor de la semilla 1 debe ser mayor a
tres digitos");
            Semilla1.setText("");
            R.setText("");
            ri.setText("");
            xi.setText("");
            y1.setText("");

            //Permite que la semilla 2 no sea menor a 3 digitos
            if(numero2<=999)
            {
                JOptionPane.showMessageDialog(null,"Valor de la semilla 2 debe ser mayor a
tres digitos");
                Semilla2.setText("");
                R.setText("");
                ri.setText("");
                xi.setText("");
                y1.setText("");

            }try
            {
                //En caso de que la semilla 1 y la semilla 2 sean mayores a 3 digitos el algoritmo
                de productos medios se ejecutara
                if(numero1>=999 && numero2>=999){
                    for (i=1; i<=Ri; i++){
                        multi = (numero1*numero2);
                        snumero3 = Long.toString(multi);

                        tam3 = snumero3.length();

                        primerc = (tam3-tam1) / 2;
                        snumero4 = snumero3.substring(primerc,primerc+tam1);

                        cadenaxi+=("x"+i+ " = " +snumero4+" \n");
                        cadenari+=("r"+i+ " = 0. " + snumero4+ " \n");
                        cadenay1+=("y"+i+ " = " + snumero3+ " \n");

                        numero1=numero2;
                        numero2 = Integer.parseInt(snumero4);

                    }
                    xi.setText(cadenaxi);
                    ri.setText(cadenari);
                    y1.setText(cadenay1);
                }
            }

```

```

    }
    }catch(java.lang.NumberFormatException e)
    { //En caso de que algun dato ingresado sea invalido, se vaciara la caja en
question
    JOptionPane.showMessageDialog(null,"Dato invalido");
    Semilla1.setText("");
    Semilla2.setText("");
    R.setText("");
    ri.setText("");
    xi.setText("");
    y1.setText("");
    }
    }

    }

    private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Semilla1.setText(null);
        Semilla2.setText(null);
        R.setText(null);
        ri.setText(null);
        xi.setText(null);
        y1.setText(null);
    }

    private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void backActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Menú Menú = new Menú();
        Menú.setVisible(true);
        ProductosMedios.this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.

```

```

*                                For                                details                                see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ProductosMedios.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ProductosMedios.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ProductosMedios.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ProductosMedios.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
    }
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new ProductosMedios().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JButton Exit;
private javax.swing.JButton Generar;
private javax.swing.JTextField R;
private javax.swing.JTextField Semilla1;

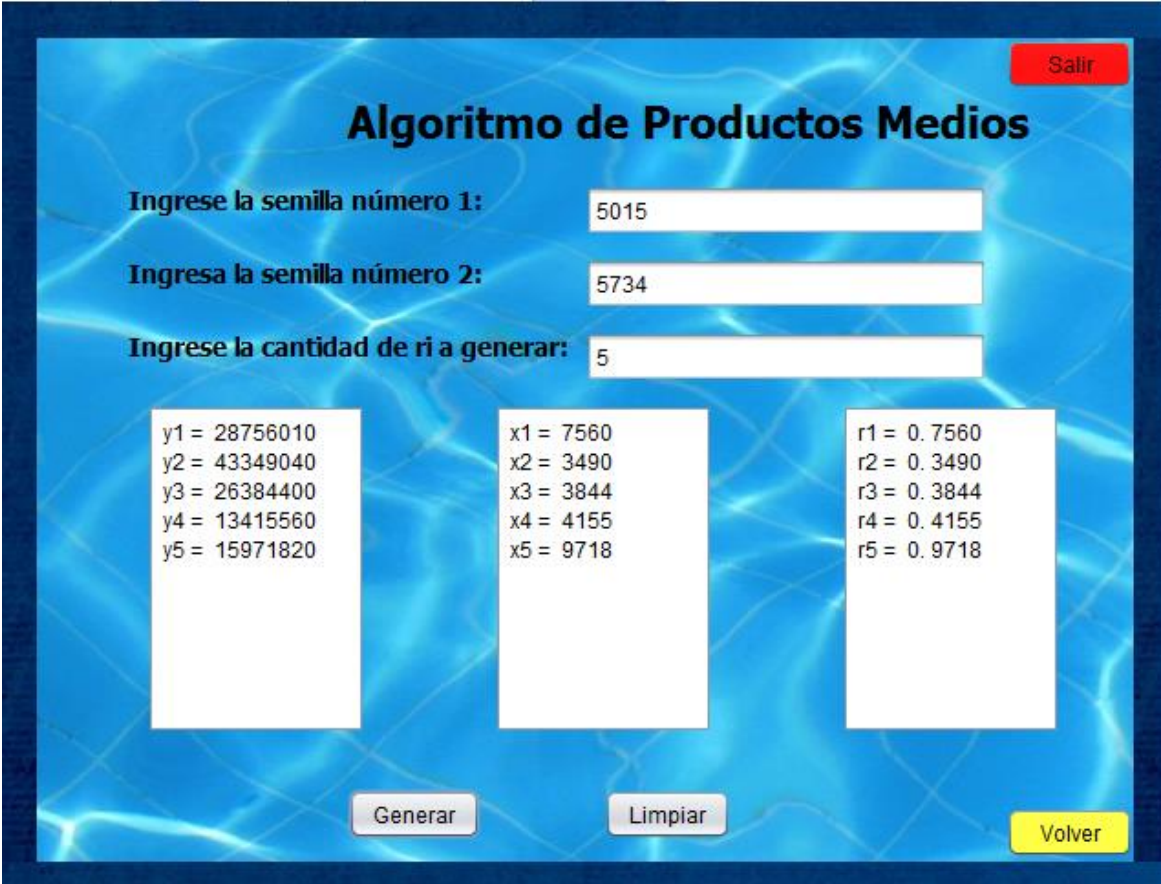
```

```

private javax.swing.JTextField Semilla2;
private javax.swing.JButton back;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextPane ri;
private javax.swing.JTextPane xi;
private javax.swing.JTextPane y1;
// End of variables declaration
}

```

## EJEMPLO



**Algoritmo de Productos Medios**

Salir

Ingrese la semilla número 1: 5015

Ingresa la semilla número 2: 5734

Ingrese la cantidad de ri a generar: 5

y1 = 28756010	x1 = 7560	r1 = 0.7560
y2 = 43349040	x2 = 3490	r2 = 0.3490
y3 = 26384400	x3 = 3844	r3 = 0.3844
y4 = 13415560	x4 = 4155	r4 = 0.4155
y5 = 15971820	x5 = 9718	r5 = 0.9718

Generar Limpiar Volver

## **CONCLUSIONES**

Nuestro código resulta extenso debido a la creación de los diferentes componentes que forman nuestra Interfaz del programa, en este programa, se ejecuta el algoritmo de productos medios, para esto, es necesario contar con 2 semillas iniciales, y pedir la cantidad de números que se busca generar, una vez que los datos estén escritos en sus campos, se pulsa el botón generar, que se encarga de ejecutar el código que contiene la fórmula para crear los números que buscamos, en cada iteración, la primera semilla se reemplazara con el numero nuevo, hasta completar los números que se buscaban generar.

Si comparamos ambos resultados, el que se muestra en la captura anterior, y el de nuestro desarrollo, observaremos que son iguales, por lo tanto, nuestro programa desarrolla el algoritmo de productos medios de forma eficiente.

## ALGORITMO DE MULTIPLICADOR CONSTANTE

### DESCRIPCIÓN

Este algoritmo no congruencial es similar al algoritmo de cuadrados medios. Lo siguiente son los pasos necesarios para generar números pseudo aleatorios, con el algoritmo de multiplicador constante.

### OBJETIVO

Desarrollar un programa, que funcione según el algoritmo multiplicador constante, contando con un valor constante, así como una semilla inicial  $X_0$  y conociendo la cantidad de datos que se busca generar mediante este algoritmo. El funcionamiento de este algoritmo, facilita su programación, ya que maneja una constante, una única operación que se repetirá en cada iteración, cambiando solamente el valor de la semilla inicial, por la del nuevo número generado.

Programar la Interfaz gráfica, para facilitar el uso del algoritmo al usuario final, con el uso de los diferentes componentes que se pueden usar en el entorno gráfico, tales como buttons, textfields, textareas, labels, entre otros.

### DESARROLLO

Generar los primeros 5 números  $r_i$  a partir de la semilla  $X_0 = 9803$  y con la constante  $a = 6965$ . Observe que tanto la semilla como la constante tienen  $D = 4$  dígitos.

Solución:

$Y_0 = (6965)(9803) = 68277895$   $X_1 = 2778$   $r_1 = 0.2778$   
 $V_1 = (6965)(2778) = 19348770$   $X_2 = 3487$   $r_2 = 0.3487$   
 $Y_2 = (6965)(3487) = 24286955$   $X_3 = 2869$   $r_3 = 0.2869$   
 $Y_3 = (6965)(2869) = 19982585$   $X_4 = 9825$   $r_4 = 0.9825$   
 $Y_4 = (6965)(9825) = 68431125$   $X_5 = 4311$   $r_5 = 0.4311$

### CÓDIGO

```
package GNPpseudoAleatorios;

import javax.swing.JOptionPane;

public class MultiplicadorConstante extends javax.swing.JFrame {

    public MultiplicadorConstante() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```

private void initComponents() {

    jLabel4 = new javax.swing.JLabel();
    Semilla = new javax.swing.JTextField();
    Constante = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    R = new javax.swing.JTextField();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    ri = new javax.swing.JTextPane();
    jScrollPane2 = new javax.swing.JScrollPane();
    xi = new javax.swing.JTextPane();
    jScrollPane3 = new javax.swing.JScrollPane();
    y1 = new javax.swing.JTextPane();
    Generar = new javax.swing.JButton();
    Clear = new javax.swing.JButton();
    Exit = new javax.swing.JButton();
    back = new javax.swing.JButton();
    jLabel10 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setUndecorated(true);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    jLabel4.setText("Ingresa la semilla:");
    getContentPane().add(jLabel4,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 100, -1, -1));
    getContentPane().add(Semilla,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 100, 220, -1));
    getContentPane().add(Constante,                                   new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 140, 220, -1));

    jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    jLabel3.setText("Ingresa la constante:");
    getContentPane().add(jLabel3,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 140, -1, -1));

    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
    jLabel1.setText("Ingresa la cantidad de ri a generar:");

```



```

        getContentPane().add(jLabel1,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 180, -1, -1));
        getContentPane().add(R,                                       new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 180, 220, -1));

        jPanel1.setBackground(new java.awt.Color(153, 255, 204));
        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel2.setText("Algoritmo de Multiplicador Constante");
        jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(130,
50, -1, 30));

        jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
        jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

        jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
        jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

        jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
        jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20,
20, 450));

        jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
        jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

        jScrollPane1.setViewportView(ri);

        jPanel1.add(jScrollPane1,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(450, 220, 120, 180));

        jScrollPane2.setViewportView(xi);

        jPanel1.add(jScrollPane2,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(280, 220, 120, 180));

```

```

jScrollPane3.setViewportView(y1);

jPanel1.add(jScrollPane3,
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 220, 120, 180));

Generar.setText("Generar");
Generar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        GenerarActionPerformed(evt);
    }
});
jPanel1.add(Generar, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
420, -1, -1));

Clear.setText("Limpiar");
Clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClearActionPerformed(evt);
    }
});
jPanel1.add(Clear, new org.netbeans.lib.awtextra.AbsoluteConstraints(340,
420, -1, -1));

Exit.setBackground(new java.awt.Color(255, 0, 0));
Exit.setText("Salir");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});
jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

back.setBackground(new java.awt.Color(255, 255, 51));
back.setText("Volver");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

jLabel10.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/lmg/5.jpg"))
); // NOI18N

```

```

        jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

        getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

        pack();
    } // </editor-fold>

    private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (evt.getSource()==Generar)
        {

            if(Semilla.getText().equals("")) //***** Permite verificar si la caja de texto esta
vacía *****//
            {
                JOptionPane.showMessageDialog(null,"Ingresa valor de semilla 1");
            }
            if(Constante.getText().equals("")) //***** Permite verificar si la caja de texto
esta vacía *****//
            {
                JOptionPane.showMessageDialog(null,"Ingresa valor de la constante");
            }
            if(R.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingresa la cantidad de ri a generar:");
            }
            //Declaracion de variables
            String semilla1=Semilla.getText();
            String semilla2=Constante.getText();
            Integer Ri= new Integer(R.getText());

            String snumero3, snumero4, cadenaxi = "",cadenari = "",cadenay1 = "";
            int tam1, tam3, i, primerc;
            long numero1, numero2, multi;
            tam1 = semilla1.length();
            numero1 = Integer.parseInt(semilla1);
            numero2 = Integer.parseInt(semilla2);
            if(numero1<=999)//Permite que la semilla 1 no sea menor a 3 digitos
            {
                JOptionPane.showMessageDialog(null,"Valor de la semilla debe ser mayor a
tres digitos");
                Semilla.setText("");
                R.setText("");
                ri.setText("");
                xi.setText("");
                y1.setText("");
            }
        }
    }

```

```

    }
    if(numero2<=999)//Permite que la semilla 2 sea menor a 3 digitos
    {
        JOptionPane.showMessageDialog(null,"Valor de la constante debe ser mayor a
tres digitos");
        Constante.setText("");
        R.setText("");
        ri.setText("");
        xi.setText("");
        y1.setText("");

    }try
    {
        //En caso de que la semilla sea mayor a tres digitos se aplica el algoritmo de
multiplicador constante
        if(numero1>=999 && numero2>=999){
            for (i=1; i<=Ri; i++){
                multi = (numero1*numero2);
                snumero3 = Long.toString(multi);

                tam3 = snumero3.length();

                primerc = (tam3-tam1) / 2;
                snumero4 = snumero3.substring(primerc,primerc+tam1);

                cadenaxi+="x"+i+" = " + snumero4+" \n");

                cadenari+=( "r"+i+" = 0. " + snumero4+ " \n");

                cadenay1+=( "y"+i+" = " + snumero3+ " \n");

                numero1=Integer.parseInt(snumero4);

            }
            xi.setText(cadenaxi);
            ri.setText(cadenari);
            y1.setText(cadenay1);
        }
    }catch(java.lang.NumberFormatException e)
    {
        //Permite que al ingresar un dato no corecto vacie la caja en cuestion
        JOptionPane.showMessageDialog(null,"Dato invalido");
        Semilla.setText("");
        Constante.setText("");
        R.setText("");
        ri.setText("");
        xi.setText("");
        y1.setText("");
    }

```

```

    }
}
}
}

private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
    // Boton que permite vacar las cajas
    Semilla.setText(null);
    Constante.setText(null);
    R.setText(null);
    ri.setText(null);
    xi.setText(null);
    y1.setText(null);
}

private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    // Boton que permite salir del programa
    System.exit(0);
}

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    // Permite regresar al menu
    Menú Menú = new Menú();
    Menú.setVisible(true);
    MultiplicadorConstante.this.dispose();
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    *
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```

```
java.util.logging.Logger.getLogger(MultiplicadorConstante.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(MultiplicadorConstante.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(MultiplicadorConstante.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(MultiplicadorConstante.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
```

```
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new MultiplicadorConstante().setVisible(true);
    }
});
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JTextField Constante;
private javax.swing.JButton Exit;
private javax.swing.JButton Generar;
private javax.swing.JTextField R;
private javax.swing.JTextField Semilla;
private javax.swing.JButton back;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
```

```

private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextPane r1;
private javax.swing.JTextPane r2;
private javax.swing.JTextPane r3;
private javax.swing.JTextPane r4;
private javax.swing.JTextPane r5;
// End of variables declaration
}

```

## EJEMPLO

**Algoritmo de Multiplicador Constante**

Ingresa la semilla: 9803

Ingresa la constante: 6965

Ingresa la cantidad de r1 a generar: 5

y1 = 68277895	x1 = 2778	r1 = 0.2778
y2 = 19348770	x2 = 3487	r2 = 0.3487
y3 = 24286955	x3 = 2869	r3 = 0.2869
y4 = 19982585	x4 = 9825	r4 = 0.9825
y5 = 68431125	x5 = 4311	r5 = 0.4311

Generar Limpiar Volver Salir

## **CONCLUSIONES**

Este algoritmo, fue con el que más nos gustó trabajar, dada la simplicidad para programar su correcto funcionamiento, únicamente se obtiene una semilla inicial, una constante, y la cantidad de números a generar, usamos la fórmula para calcular el valor, y mediante una serie de iteraciones, en la que sustituimos la semilla inicial por el valor resultante, hasta obtener la cantidad de números pseudoaleatorios que se buscaban generar en un principio.

Gracias a la interfaz, el usuario final puede interactuar fácilmente con nuestro programa, únicamente tiene que llenar los campos necesarios, pulsar un botón, y obtendrá el resultado que buscaba, de una manera entendible.

Al comparar los resultados que se obtuvieron de nuestro programa y los calculados en el desarrollo (lo que calculamos en clase), podemos identificar que los valores son iguales, y se concluye que nuestro programa está en perfecto funcionamiento.



## ALGORITMO LINEAL

### DESCRIPCIÓN

Este algoritmo congruencial fue propuesto por D. H. Lehmer en 1951. Según Law y Kelton este algoritmo ha sido el más usado. El algoritmo congruencial lineal genera una secuencia de números enteros por medio de la siguiente ecuación recursiva:

$$X_{i+1} = (AX_i + C) \bmod(M) \quad i=0,1,2, \quad 3,\dots,n$$

Donde  $X_0$  es la semilla,  $A$  es la constante multiplicativa,  $C$  es la constante aditiva y  $M$  es el módulo;  $X_0$  es mayor a 0,  $A$  es mayor a 0,  $C$  mayor a 0 y  $M$  mayor a 0. Deben ser números enteros. Es importante señalar que la ecuación recursiva del algoritmo congruencial lineal genera una secuencia de números enteros:  $s=\{0,1,2,3,\dots, m-1\}$  Y que para obtener números pseudoaleatorios en el intervalo  $(0,1)$  se requiere la siguiente ecuación:

$$r_i = (X_i)/(m-1) \quad i=1,2,3,\dots,n.$$

### OBJETIVO

Desarrollar un programa, que realice el algoritmo lineal, el cual también es el más común de este tipo de algoritmos, en este caso se generaran únicamente 4 números, ese es la longitud de vida de este algoritmo, gracias a los parámetros que serán almacenados en variables en nuestro código, nos permitirá calcular el primer número pseudoaleatorio siguiendo la primer fórmula de arriba, para sustituir únicamente el valor de  $x_i$ , reemplazando el inicial por el nuevo, haciendo lo mismo en cada iteración.

Programar una interfaz gráfica, mediante la cual el usuario final, pueda ejecutar el algoritmo, de manera que se sienta más cómodo mediante el uso de botones, y campos donde escribirá la información que posee.

### DESARROLLO

Generar 4 números entre 0 y 1 con los siguientes parámetros:  $X_0 = 37$ ,  $a = 19$ ,  $c = 33$  y  $m = 100$

*Solución:*

$$\begin{aligned} X_1 &= (19 \cdot 37 + 33) \bmod 100 = 36 \quad r_1 = 36/99 = 0.3636 \\ X_2 &= (19 \cdot 36 + 33) \bmod 100 = 17 \quad r_2 = 17/99 = 0.1717 \\ X_3 &= (19 \cdot 17 + 33) \bmod 100 = 56 \quad r_3 = 56/99 = 0.5656 \\ X_4 &= (19 \cdot 56 + 33) \bmod 100 = 97 \quad r_4 = 97/99 = 0.9797 \end{aligned}$$

### CÓDIGO

```
package GNPpseudoAleatorios;

import java.math.BigDecimal;
import java.math.RoundingMode;
```

```

import javax.swing.JOptionPane;

public class Lineal extends javax.swing.JFrame {

    public Lineal() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel4 = new javax.swing.JLabel();
        Semilla = new javax.swing.JTextField();
        a = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        Generar = new javax.swing.JButton();
        Clear = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        ri = new javax.swing.JTextPane();
        jScrollPane2 = new javax.swing.JScrollPane();
        xi = new javax.swing.JTextPane();
        jLabel5 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        m = new javax.swing.JTextField();
        c = new javax.swing.JTextField();
        jScrollPane3 = new javax.swing.JScrollPane();
        div = new javax.swing.JTextPane();
        jLabel7 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        Exit = new javax.swing.JButton();
        back = new javax.swing.JButton();
        jLabel11 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        jLabel4.setText("Ingrese la semilla:");
        getContentPane().add(jLabel4,
            org.netbeans.lib.awtextra.AbsoluteConstraints(70, 100, -1, -1));
    }
}

```

new

```

        getContentPane().add(Semilla,
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 100, 220, -1));
        getContentPane().add(a,
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 140, 220, -1));

        jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        jLabel3.setText("Ingrese el parámetro a:");
        getContentPane().add(jLabel3,
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 140, -1, -1));

        jPanel1.setBackground(new java.awt.Color(153, 255, 204));
        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel2.setText("Algoritmo Lineal");
        jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(240,
50, -1, 30));

        Generar.setText("Generar");
        Generar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                GenerarActionPerformed(evt);
            }
        });
        jPanel1.add(Generar, new org.netbeans.lib.awtextra.AbsoluteConstraints(520,
250, -1, -1));

        Clear.setText("Limpiar");
        Clear.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ClearActionPerformed(evt);
            }
        });
        jPanel1.add(Clear, new org.netbeans.lib.awtextra.AbsoluteConstraints(520,
300, 70, -1));

        jScrollPane1.setViewportView(ri);

        jPanel1.add(jScrollPane1,
org.netbeans.lib.awtextra.AbsoluteConstraints(360, 260, 120, 180));

        jScrollPane2.setViewportView(xi);

        jPanel1.add(jScrollPane2,
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 260, 120, 180));

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

```

```

jLabel5.setText("Ingrese el parámetro m:");
jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
220, -1, -1));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setText("Ingrese el parámetro c:");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
180, -1, -1));
jPanel1.add(m, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 220,
220, -1));
jPanel1.add(c, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 180,
220, -1));

jScrollPane3.setViewportView(div);

jPanel1.add(jScrollPane3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 260, 120, 180));

jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20,
20, 450));

jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

Exit.setBackground(new java.awt.Color(255, 0, 0));
Exit.setText("Salir");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});

```

```

    }
    });
    jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

    back.setBackground(new java.awt.Color(255, 255, 51));
    back.setText("Volver");
    back.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            backActionPerformed(evt);
        }
    });
    jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

    jLabel11.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/5.jpg"))
); // NOI18N
    jPanel1.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

    getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

    pack();
} // </editor-fold>

private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {

    if (evt.getSource()==Generar)
    {

        if(Semilla.getText().equals("")) //***** Permite verificar si las cajas de texto
están vacías *****//
        {
            JOptionPane.showMessageDialog(null,"Ingresa valor de semilla");
        }

        if(a.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro a:");
        }

        if(c.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro c:");
        }
    }
}

```

```

if(m.getText().equals(""))
{
JOptionPane.showMessageDialog(null,"Ingrese el parámetro m:");
}

//Declaracion de variables//
String semilla1=Semilla.getText();
Integer A= new Integer (a.getText());
Integer C= new Integer(c.getText());
Integer M= new Integer(m.getText());

String snumero2, snumero3, snumero4, cadenaxi = "",cadenari =
"",cadenadiv="";
int tam1, tam3, i, primerc, MR ;
long numero1, numero3, formula;

tam1 = semilla1.length();
numero1 = Integer.parseInt(semilla1);
MR=M-1;
//permite que la caja sea mayor o igual a 1//
if(numero1<1)
{
JOptionPane.showMessageDialog(null,"Valor de la semilla debe ser mayor ó
igual a 1");
Semilla.setText("");
a.setText("");
c.setText("");
m.setText("");
ri.setText("");
xi.setText("");
div.setText("");

}try
{//En caso de que sea mayor a 1 se aplica el algoritmo lineal//
if(numero1>=1){
for (i=1; i<=4; i++){
formula = ((A*numero1+C)%M);

snumero2 = Long.toString(formula);
numero3 = Integer.parseInt(snumero2);
double rs= (double) numero3/MR;
BigDecimal formatNumber = new BigDecimal(rs);
formatNumber = formatNumber.setScale(4, RoundingMode.DOWN);

cadenaxi+=("x"+i+ " = " +snumero2+" \n");

cadenadiv+=(snumero2+"/"+ MR +" \n");
}
}

```

```

        cadenari+=( "r"+i+" = " + formatNumber + " \n");

        numero1=Integer.parseInt(snumero2);

    }
    xi.setText(cadenaxi);
    ri.setText(cadenari);
    div.setText(cadenadiv);
}

}catch(java.lang.NumberFormatException e)
{//En caso de que sea invalido un dato ingresado se limpiara todas las casillas//
JOptionPane.showMessageDialog(null,"Dato invalido");
Semilla.setText("");
a.setText("");
c.setText("");
m.setText("");
ri.setText("");
xi.setText("");
div.setText("");
}
}
}

private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
    //Permite vaciar las cajas//
    Semilla.setText(null);
    a.setText(null);
    c.setText(null);
    ri.setText(null);
    xi.setText(null);
    div.setText(null);
    m.setText(null);
}

private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    //Permite salir del programa//
    System.exit(0);
}

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    //Permite regrear al menu//
    Menú Menú = new Menú();
    Menú.setVisible(true);
    Lineal.this.dispose();
}

```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    *
    For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Lineal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Lineal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Lineal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Lineal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Lineal().setVisible(true);
        }
    });
}

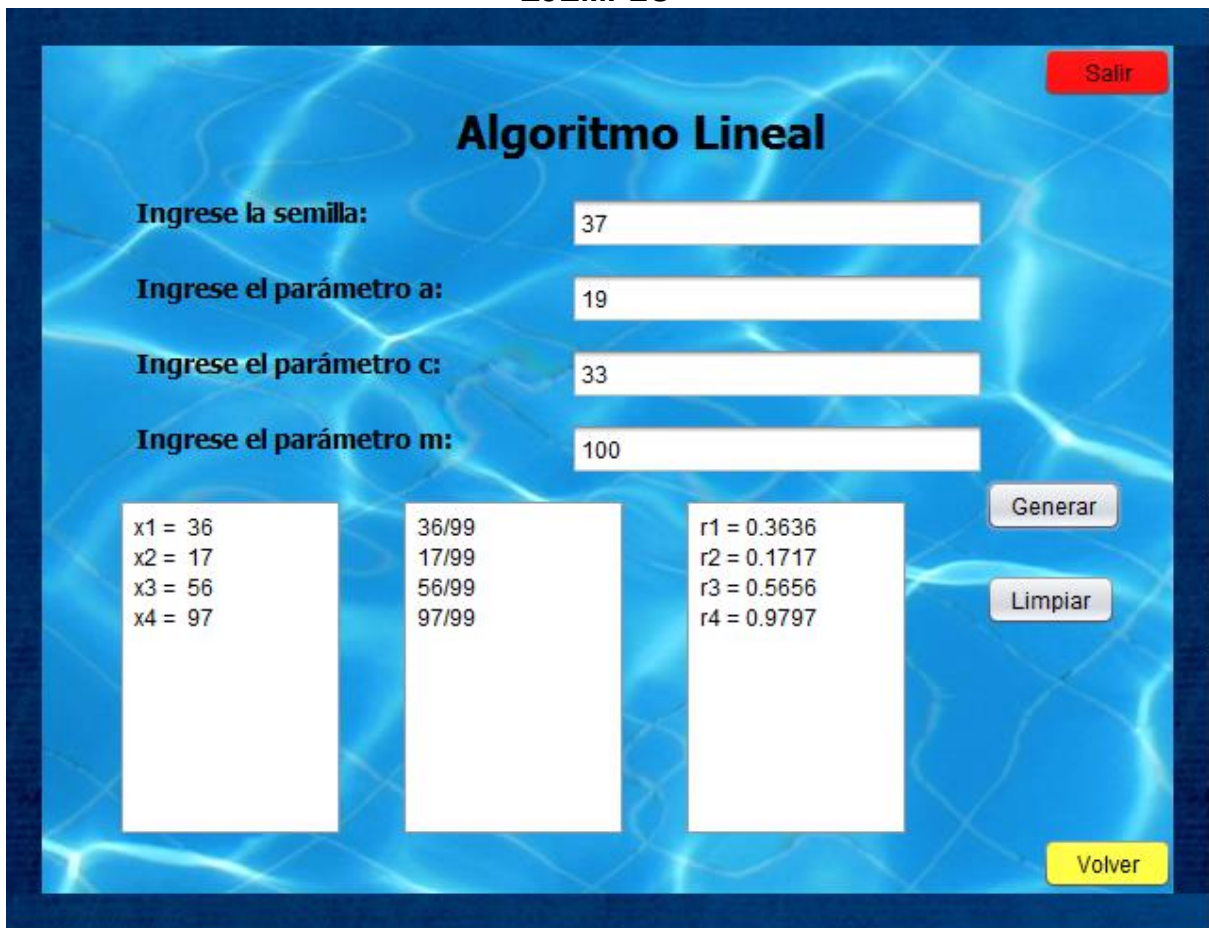
// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JButton Exit;

```



```
private javax.swing.JButton Generar;  
private javax.swing.JTextField Semilla;  
private javax.swing.JTextField a;  
private javax.swing.JButton back;  
private javax.swing.JTextField c;  
private javax.swing.JTextPane div;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel11;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane2;  
private javax.swing.JScrollPane jScrollPane3;  
private javax.swing.JTextField m;  
private javax.swing.JTextPane ri;  
private javax.swing.JTextPane xi;  
// End of variables declaration  
}
```

## EJEMPLO



**Algoritmo Lineal**

Salir

Ingrese la semilla: 37

Ingrese el parámetro a: 19

Ingrese el parámetro c: 33

Ingrese el parámetro m: 100

Generar

Limpiar

Volver

x1 = 36	36/99	r1 = 0.3636
x2 = 17	17/99	r2 = 0.1717
x3 = 56	56/99	r3 = 0.5656
x4 = 97	97/99	r4 = 0.9797

## CONCLUSIONES

Como se observa en la captura de arriba, es necesario escribir en los campos, la semilla inicial, los parámetros a, c y el mod que se utilizará, al presionar el botón generar, todo nuestro código hará la operación de manera repetitiva para generar todos los números, guardando el resultado en una cadena, que terminará imprimiéndose en un textarea, el primero de estos nos muestra el valor de x, el segundo la división de  $x_i / \text{mod} - 1$ , y el tercero nos muestra el número pseudoaleatorio.

La programación de este algoritmo es un poco diferente a la de los otros, ya que, en vez de manejar posiciones, nos imprime un valor  $r_i$  en decimal que puede ser muy grande, por eso era necesario limitar su impresión a únicamente 4 valores decimales, así como evitar su redondeo.

Una vez más, programar una interfaz gráfica que interactúe con el usuario, facilita el uso a nuestro usuario, y el botón que limpia todos los campos, permite volver a usar el algoritmo, sin tener que volver a ejecutar el programa.

Los resultados obtenidos en nuestro programa y en nuestro desarrollo del algoritmo, es el mismo en ambos casos, confirmando el correcto funcionamiento de nuestro programa.

## ALGORITMO LINEAL (BANKS, CARSON, NELSON, NICO)

### DESCRIPCIÓN

Para que el algoritmo sea capaz de lograr el máximo periodo de vida  $n$ , es preciso que dichos parámetros cumplan dichas condiciones, Banks, Carson, Nelson y Nicol sugirieron lo siguiente:

$$\begin{aligned}m &= 2^{g^2} \\ a &= 1 + 4k \\ k &\text{ debe de ser entero} \\ c &\text{ relativamente a } m \\ g &\text{ debe ser entero}\end{aligned}$$

Bajo estas condiciones se obtiene un periodo de vida máximo  $N = m = 2g$

### OBJETIVO

Desarrollar un programa que ejecute el algoritmo lineal propuesto por Banks, disponiendo de los valores obtenidos de los parámetros  $X_0$ ,  $k$ ,  $g$  y  $c$ , el último parámetro  $m$ , el cual será el mod y también será el ciclo de vida de nuestro algoritmo. Se realizará la fórmula para calcular el valor, mediante una serie de iteraciones, hasta que se termine el ciclo de vida, de otra manera, los números se repetirían.

Programar una interfaz gráfica del algoritmo propuesto por Banks, para que de esta manera el usuario final, pueda generar números pseudoaleatorios de manera más sencilla.

### DESARROLLO

Generar suficientes números entre 0 y 1 con los parámetros  $X_0 = 6$ ,  $k = 3$ ,  $g = 3$  y  $c = 7$ , hasta encontrar el periodo de vida máximo ( $A$ ).

Como podemos ver, si se cumplen las condiciones que Banks, Carson, Nelson y Nicol sugieren, se logrará el periodo máximo  $N = m = 8$ . A continuación se presenta el desarrollo de la generación de los números  $r_i$ .

$$a = 1 + 4(3) = 13 \text{ y } m = 23 = 8 \text{ } X_0 = 6$$

$$X_1 = (13 \cdot 6 + 7) \bmod 8 = 5 \text{ } r_1 = 5/7 = 0.714$$

$$X_2 = (13 \cdot 5 + 7) \bmod 8 = 0 \text{ } r_2 = 0/7 = 0.000$$

$$X_3 = (13 \cdot 0 + 7) \bmod 8 = 7 \text{ } r_3 = 7/7 = 1.000$$

$$X_4 = (13 \cdot 7 + 7) \bmod 8 = 2 \text{ } r_4 = 2/7 = 0.285$$

$$X_5 = (13 \cdot 2 + 7) \bmod 8 = 1 \text{ } r_5 = 1/7 = 0.142$$

$$X_6 = (13 \cdot 1 + 7) \bmod 8 = 4 \text{ } r_6 = 4/7 = 0.571$$

$$X_7 = (13 \cdot 4 + 7) \bmod 8 = 3 \text{ } r_7 = 3/7 = 0.428$$

$$X_8 = (13 \cdot 3 + 7) \bmod 8 = 6 \text{ } r_8 = 6/7 = 0.857$$

## CÓDIGO

```
package GNPpseudoAleatorios;

import java.math.BigDecimal;
import java.math.RoundingMode;
import javax.swing.JOptionPane;

public class LinealBanks extends javax.swing.JFrame {

    public LinealBanks() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel4 = new javax.swing.JLabel();
        Semilla = new javax.swing.JTextField();
        k = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        g = new javax.swing.JTextField();
        c = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        xi = new javax.swing.JTextPane();
        jScrollPane3 = new javax.swing.JScrollPane();
        div = new javax.swing.JTextPane();
        jScrollPane1 = new javax.swing.JScrollPane();
        ri = new javax.swing.JTextPane();
        Generar = new javax.swing.JButton();
        Clear = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        Exit = new javax.swing.JButton();
        back = new javax.swing.JButton();
        jLabel11 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setUndecorated(true);
```

```

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel4.setText("Ingrese la semilla:");
getContentPane().add(jLabel4,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 100, -1, -1));
getContentPane().add(Semilla,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 100, 220, -1));
getContentPane().add(k,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 140, 220, -1));

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel3.setText("Ingrese el parámetro k:");
getContentPane().add(jLabel3,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 140, -1, -1));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setText("Ingrese el parámetro g:");
getContentPane().add(jLabel1,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 180, -1, -1));
getContentPane().add(g,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 180, 220, -1));
getContentPane().add(c,                                          new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 220, 220, -1));

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel5.setText("Ingrese el parámetro c:");
getContentPane().add(jLabel5,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 220, -1, -1));

jScrollPane2.setViewportView(xi);

getContentPane().add(jScrollPane2,                                 new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 260, 120, 180));

jScrollPane3.setViewportView(div);

getContentPane().add(jScrollPane3,                                 new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 260, 120, 180));

jScrollPane1.setViewportView(ri);

getContentPane().add(jScrollPane1,                                 new
org.netbeans.lib.awtextra.AbsoluteConstraints(360, 260, 120, 180));

Generar.setText("Generar");
Generar.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            GenerarActionPerformed(evt);
        }
    });
    getContentPane().add(Generar,
org.netbeans.lib.awtextra.AbsoluteConstraints(520, 250, -1, -1));

    Clear.setText("Limpiar");
    Clear.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ClearActionPerformed(evt);
        }
    });
    getContentPane().add(Clear,
org.netbeans.lib.awtextra.AbsoluteConstraints(520, 300, 70, -1));

    jPanel1.setBackground(new java.awt.Color(153, 255, 204));
    jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
    jLabel2.setText("Algoritmo Lineal (Banks, Carson, Nelson, Nicol)");
    jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
50, -1, 30));

    jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
    jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

    jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
    jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20,
20, 450));

    jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
    jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

    jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
    jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

```

```

Exit.setBackground(new java.awt.Color(255, 0, 0));
Exit.setText("Salir");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});
jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

back.setBackground(new java.awt.Color(255, 255, 51));
back.setText("Volver");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

jLabel11.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/5.jpg"))
); // NOI18N
jPanel1.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

getContentPane().add(jPanel1,                                     new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

pack();
} // </editor-fold>

private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {

    if (evt.getSource()==Generar)
    {

        if(Semilla.getText().equals("")) //***** Permite verificar si las cajas de texto
        estan vacias *****//
        {
            JOptionPane.showMessageDialog(null,"Ingresa valor de semilla");
        }

        if(k.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro k:");
        }
    }
}

```

```

    }

    if(g.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null,"Ingrese el parámetro g:");
    }
    if(c.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null,"Ingrese el parámetro c:");
    }
    //Declaracion de Variables y operaciones
    String semilla1=Semilla.getText();
    Integer K= new Integer (k.getText());
    Integer G= new Integer(g.getText());
    Integer C= new Integer(c.getText());

    String snumero2, snumero3, snumero4, cadenaxi = "",cadenari =
    "",cadenadiv="";
    int tam1, tam3, i, primerc, M, A;
    long numero1, numero2, formula;

    tam1 = semilla1.length();
    numero1 = Integer.parseInt(semilla1);

    M=(int)Math.pow(2,G);
    A=1+(4*K);
    //Permite que la semilla sea mayor o igual a 1, encaso contrario se vacian las
    cajas
    if(numero1<1)
    {
        JOptionPane.showMessageDialog(null,"Valor de la semilla debe ser mayor ó
    igual a 1");
        Semilla.setText("");
        g.setText("");
        c.setText("");
        k.setText("");
        ri.setText("");
        xi.setText("");
        div.setText("");
    }
    }try
    {
        //En caso de que sea mallor a 1 la semmilla, se comenzara a realizar El algoritmo
        Lineal Banks
        if(numero1>=1){
            for (i=1; i<=M; i++){
                formula = (long) ((A*numero1+C)%M);
                snumero2 = Long.toString(formula);
                numero2 = Integer.parseInt(snumero2);
            }
        }
    }

```



```

        double rs= (double) numero2/C;
        BigDecimal formatNumber = new BigDecimal(rs);
        formatNumber = formatNumber.setScale(4, RoundingMode.DOWN);

        cadenaxi+=("x"+i+ " = " +snumero2+" \n");

        cadenari+=( "r"+i+ " = " + formatNumber + " \n");

        cadenadiv+=(snumero2+"/"+C+"\n");

        numero1=Integer.parseInt(snumero2);

    }
    xi.setText(cadenaxi);
    ri.setText(cadenari);
    div.setText(cadenadiv);
}
}catch(java.lang.NumberFormatException e)//En caso de que un dato ingresado
sea invalido se vacia la caja en cuestion
{
    JOptionPane.showMessageDialog(null,"Dato invalido");
    Semilla.setText("");
    k.setText("");
    g.setText("");
    c.setText("");
    ri.setText("");
    xi.setText("");
    div.setText("");
}
}
}
private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
    // Permite vaciar las cajas
    Semilla.setText(null);
    k.setText(null);
    g.setText(null);
    ri.setText(null);
    xi.setText(null);
    div.setText(null);
    c.setText(null);
}

private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    // Permite salir del programa
    System.exit(0);
}

```

```

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    // Permite regresar al menu
    Menú Menú = new Menú();
    Menú.setVisible(true);
    LinealBanks.this.dispose();
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    *
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(LinealBanks.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LinealBanks.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LinealBanks.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LinealBanks.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new LinealBanks().setVisible(true);
        }
    }

```

```

    });
}

// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JButton Exit;
private javax.swing.JButton Generar;
private javax.swing.JTextField Semilla;
private javax.swing.JButton back;
private javax.swing.JTextField c;
private javax.swing.JTextPane div;
private javax.swing.JTextField g;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextField k;
private javax.swing.JTextPane ri;
private javax.swing.JTextPane xi;
// End of variables declaration
}

```

## EJEMPLO

**Algoritmo Lineal (Banks, Carson, Nelson, Nicol)**

Salir

Ingrese la semilla: 6

Ingrese el parámetro k: 3

Ingrese el parámetro g: 3

Ingrese el parámetro c: 7

Generar

Limpiar

Volver

x1 = 5	5/7	r1 = 0.7142
x2 = 0	0/7	r2 = 0.0000
x3 = 7	7/7	r3 = 1.0000
x4 = 2	2/7	r4 = 0.2857
x5 = 1	1/7	r5 = 0.1428
x6 = 4	4/7	r6 = 0.5714
x7 = 3	3/7	r7 = 0.4285
x8 = 6	6/7	r8 = 0.8571

## CONCLUSIONES

El usuario únicamente debe llenar los campos que corresponden a los parámetros, ingresando los valores correspondientes, la semilla  $X_0$ , los parámetros  $k$ ,  $g$  (el cual nos dirá la potencia a la que se elevará el 2, es decir  $2^g$ , dándonos el ciclo de vida de nuestro programa) y el parámetro  $c$ , pulsando el botón generar, procede a calcular cada uno de los números pseudoaleatorios, para finalmente imprimir cada una de las cadenas, en su correspondiente textarea, para dar el  $x_i$ , la división de  $x_i$  entre  $m-1$  y el número  $r_i$  calculado en cada iteración.

Como en todos nuestros programas, decidimos crear una interfaz gráfica, a manera de hacer que nuestro usuario final, tenga más fácil el usar nuestro programa y el obtener la información que desea.

Al comparar los resultados, observamos que coinciden con los calculados manualmente previamente, nuestro programa está funcionando correctamente.

## ALGORITMO CONGRUENCIAL ADICTIVO

### DESCRIPCIÓN

Este algoritmo requiere una secuencia previa de números enteros,  $X_1, X_2, X_3, \dots, X_n$ . para generar una secuencia nueva de números enteros que empieza en  $X_{n+1}, X_{n+2}, \dots$

Su ecuación recursiva es:

$$X_i = (X_{i-1} + X_{i-n}) \bmod (M)$$

$$i = n+1, n+2, n+3, \dots, N.$$

Los primeros  $r_i$  pueden ser generados mediante la ecuación:  $r_i = X_i / (M-1)$

### OBJETIVO

Desarrollar un programa que genere números pseudoaleatorios, usando el algoritmo congruencial aditivo, es necesario tomar en cuenta las variables que se usaran, en este caso, la cantidad de números a generar, la cantidad de valores que disponemos como información previa, y el modulo  $m$  que se usará en nuestras operaciones.

Programar una interfaz gráfica que ayude al fácil uso del algoritmo, mediante la ayuda de textfields, textareas, buttons y más componentes.

### DESARROLLO

Generar 7 números pseudoaleatorios entre cero y uno a partir de la siguiente secuencia de números enteros: 65, 89, 98, 3, 69;  $m = 100$ .

Sean  $X_1 = 65, X_2 = 89, X_3 = 98, X_4 = 03, X_5 = 69$ . Para generar  $r_1, r_2, r_3, r_4, r_5, r_6$  y  $r_7$  antes es necesario generar  $X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}$ .

Solución:

$X_6 = (X_5 + X_1) \bmod 100 = (69 + 65) \bmod 100 = 34$	$r_1 = 34/99 = 0.3434$
$X_7 = (X_6 + X_2) \bmod 100 = (34 + 89) \bmod 100 = 23$	$r_2 = 23/99 = 0.2323$
$X_8 = (X_7 + X_3) \bmod 100 = (23 + 98) \bmod 100 = 21$	$r_3 = 21/99 = 0.2121$
$X_9 = (X_8 + X_4) \bmod 100 = (21 + 03) \bmod 100 = 24$	$r_4 = 24/99 = 0.2424$
$X_{10} = (X_9 + X_5) \bmod 100 = (24 + 69) \bmod 100 = 93$	$r_5 = 93/99 = 0.9393$
$X_{11} = (X_{10} + X_6) \bmod 100 = (93 + 34) \bmod 100 = 27$	$r_6 = 27/99 = 0.2727$
$X_{12} = (X_{11} + X_7) \bmod 100 = (27 + 23) \bmod 100 = 50$	$r_7 = 50/99 = 0.5050$

### CÓDIGO

```
package GNPseudoAleatorios;  
  
import java.awt.GridBagConstraints;
```

```

import java.math.BigDecimal;
import java.math.RoundingMode;
import java.util.LinkedList;
import javax.swing.JOptionPane;

public class CongruencialAditivo extends javax.swing.JFrame {
    LinkedList Lista=new LinkedList();
    //Declaracion de variables
    String valor, ccuadrado="", sale="", cua="", erre="", algo="";
    String cadenativ="",cadenari="",cadenaxi="";
    private GridBagConstraints restricciones;
    int no, num1, num2, ta1=0, ta2=0, ta3=0;
    double y;
    String dato;
    int mr;

    public CongruencialAditivo() {

        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel3 = new javax.swing.JLabel();
        n = new javax.swing.JTextField();
        m = new javax.swing.JTextField();
        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        Exit = new javax.swing.JButton();
        back = new javax.swing.JButton();
        Clear = new javax.swing.JButton();
        Generar = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        ri = new javax.swing.JTextPane();
        jScrollPane3 = new javax.swing.JScrollPane();
        div = new javax.swing.JTextPane();
        jScrollPane2 = new javax.swing.JScrollPane();
        xi = new javax.swing.JTextPane();
        jLabel10 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
    }

```

```

num = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel3.setText("Ingrese la cantidad de números a generar:");
getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 110, -1, -1));
getContentPane().add(n, new
org.netbeans.lib.awtextra.AbsoluteConstraints(370, 110, 220, 30));
getContentPane().add(m, new
org.netbeans.lib.awtextra.AbsoluteConstraints(370, 160, 220, -1));

jPanel1.setBackground(new java.awt.Color(153, 255, 204));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
jLabel2.setText("Algoritmo Congruencial Aditivo");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
50, -1, 30));

jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20,
20, 450));

jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

```

```

Exit.setBackground(new java.awt.Color(255, 0, 0));
Exit.setText("Salir");
Exit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExitActionPerformed(evt);
    }
});
jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

back.setBackground(new java.awt.Color(255, 255, 51));
back.setText("Volver");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

Clear.setText("Limpiar");
Clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClearActionPerformed(evt);
    }
});
jPanel1.add(Clear, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
430, -1, -1));

Generar.setText("Generar ");
Generar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        GenerarActionPerformed(evt);
    }
});
jPanel1.add(Generar, new org.netbeans.lib.awtextra.AbsoluteConstraints(220,
430, -1, -1));

jScrollPane1.setViewportView(view(ri);

jPanel1.add(jScrollPane1,
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 240, 120, 180));

jScrollPane3.setViewportView(view(div);

```

new



```

        jPanel1.add(jScrollPane3,
org.netbeans.lib.awtextra.AbsoluteConstraints(280, 240, 120, 180));

        jScrollPane2.setViewportView(xi);

        jPanel1.add(jScrollPane2,
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 240, 120, 180));

        jLabel10.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        jLabel10.setText("Ingrese el parámetro m:");
        jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
160, -1, -1));

        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        jLabel5.setText("Ingresa el numero de datos disponibles:");
        jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
200, -1, -1));
        jPanel1.add(num, new org.netbeans.lib.awtextra.AbsoluteConstraints(370,
200, 220, -1));

        jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/lmg/5.jpg"))
); // NOI18N
        jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

        getContentPane().add(jPanel1,
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

        pack();
    } // </editor-fold>

    private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void backActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Menú Menú = new Menú();
        Menú.setVisible(true);
        CongruencialAditivo.this.dispose();
    }

    private void ClearActionPerformed(java.awt.event.ActionEvent evt) {
        //Permite vaciar las cajas
        m.setText(null);

```

```

        n.setText(null);
        num.setText(null);
        ri.setText(null);
        xi.setText(null);
        div.setText(null);
    }

    public void establecerValores(int columna,int fila, int anchura, int altura)
    {
        restricciones.gridx=columna;
        restricciones.gridy=fila;

        restricciones.gridwidth=anchura;
        restricciones.gridheight=altura;
    }

    private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        //Permite verificar si las cajas estan vacias
        if (evt.getSource()==Generar)
        {
            if(num.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingresa numero de datos deseados");
            }

            if(m.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingresa modulo");
            }

            if(n.getText().equals(""))
            {
                JOptionPane.showMessageDialog(null,"Ingresa cantidad de datos a generar");
            }

            else
            {
                try
                {
                    Integer datos= new Integer(num.getText());
                    Integer modulo= new Integer(m.getText());
                    Integer cantidad= new Integer(n.getText());

                    if(datos<=0)//Permite que el dato no sea menor o igual a 0
                    {
                        JOptionPane.showMessageDialog(null,"El numero de datos debe ser \nmayorrr
a 0","error",JOptionPane.ERROR_MESSAGE);
                        n.setText("");
                    }
                }
            }
        }
    }

```

```

    }

    else
    {

        if(modulo<=0)//Permite que el modulo no sea menor o igual a Cero
        {
            JOptionPane.showMessageDialog(null,"El numero modulo debe ser \nmayor a
0","error",JOptionPane.ERROR_MESSAGE);
            m.setText("");
        }

        else
        {

            if(cantidad<=0)//Permite que la Cantidad no sea menor o igual sea cero
            {
                JOptionPane.showMessageDialog(null,"El numero de datos debe ser \nmayor a
0","error",JOptionPane.ERROR_MESSAGE);
                num.setText("");
            }

            else
            {

                for(int i=0; i<datos; i++)
                {

                    dato = (JOptionPane.showInputDialog("Ingresa el valor de X"+(i+1)+" : "));
                    no = Integer.parseInt( dato );
                    Lista.addLast(no);
                    sale+=no+ " ";

                    ta1++;
                }

                for(int i=0; i<cantidad; i++)
                {
                    String dat1=String.valueOf(Lista.get(datos-1+i));
                    int da1=Integer.parseInt(dat1);

                    String dat2=String.valueOf(Lista.get(i));
                    int da2=Integer.parseInt(dat2);

                    ta2++;
                    ta3=ta1+ta2;
                    mr=modulo-1;
                }
            }
        }
    }

```

```

no=(da1+da2)% modulo;
Lista.addLast(no);
cadenaxi+="x"+ta3+"= "+no+"\n";
cadenadiv+=no+"/"+mr+"\n";

y=no*1.0/(modulo-1);

BigDecimal formatNumber = new BigDecimal(y);
formatNumber = formatNumber.setScale(4, RoundingMode.DOWN);
cadenari+="r"+i+"= "+formatNumber+"\n";
}

}
}
xi.setText(cadenaxi);
ri.setText(cadenari);
div.setText(cadenadiv);
}

catch(java.lang.NumberFormatException e)
{
JOptionPane.showMessageDialog(null,"Dato invalido");
n.setText("");
m.setText("");
num.setText("");
}
}
}

public static void main(String args[]) {
/* Set the Nimbus look and feel */
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
*
For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
*/
try {
for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

```



```

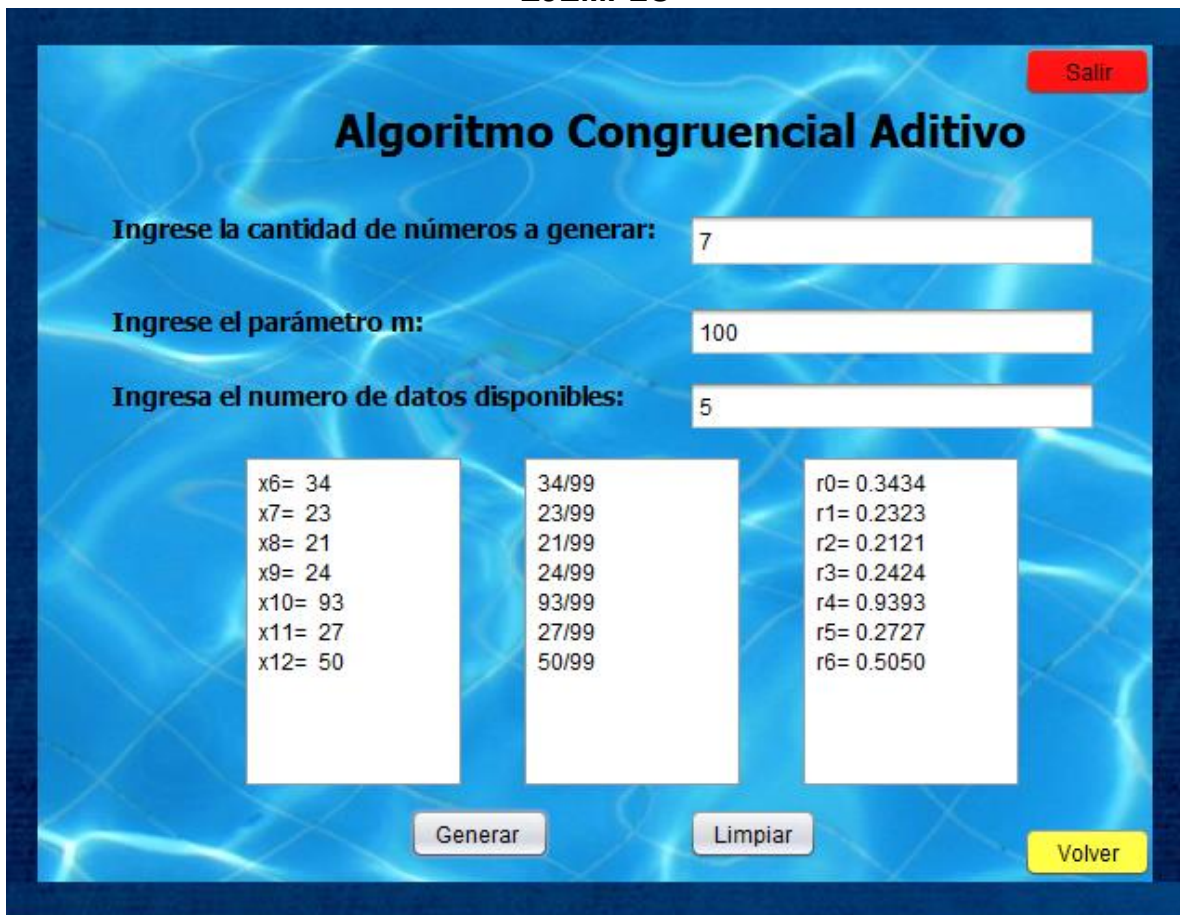
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new CongruencialAditivo().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JButton Exit;
private javax.swing.JButton Generar;
private javax.swing.JButton back;
private javax.swing.JTextPane div;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextField m;
private javax.swing.JTextField n;
private javax.swing.JTextField num;
private javax.swing.JTextPane ri;
private javax.swing.JTextPane xi;
// End of variables declaration
}

```

## EJEMPLO



**Algoritmo Congruencial Aditivo**

Salir

Ingrese la cantidad de números a generar: 7

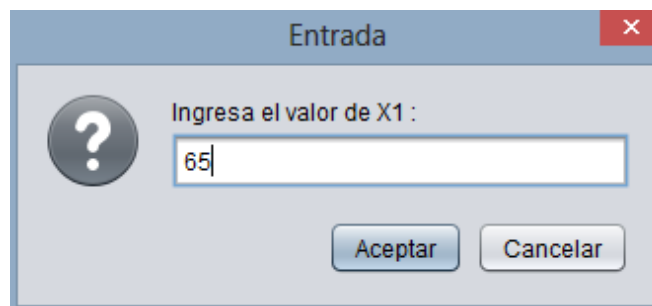
Ingrese el parámetro m: 100

Ingresa el numero de datos disponibles: 5

x6= 34	34/99	r0= 0.3434
x7= 23	23/99	r1= 0.2323
x8= 21	21/99	r2= 0.2121
x9= 24	24/99	r3= 0.2424
x10= 93	93/99	r4= 0.9393
x11= 27	27/99	r5= 0.2727
x12= 50	50/99	r6= 0.5050

Generar Limpiar Volver

**Nota:** al presionar el botón generar, nos salen estas ventanas, para ingresar los 5 números que tenemos previamente de la secuencia. Después de eso, procede a imprimir los 7 números resultantes.

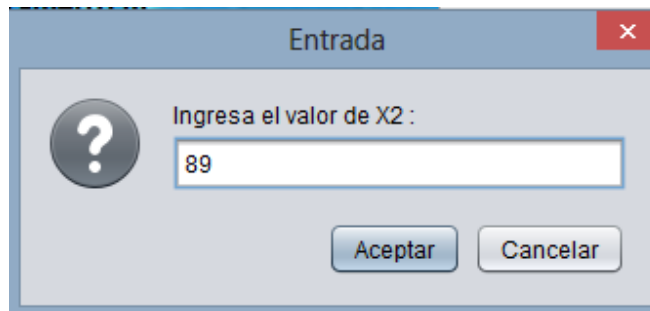


Entrada

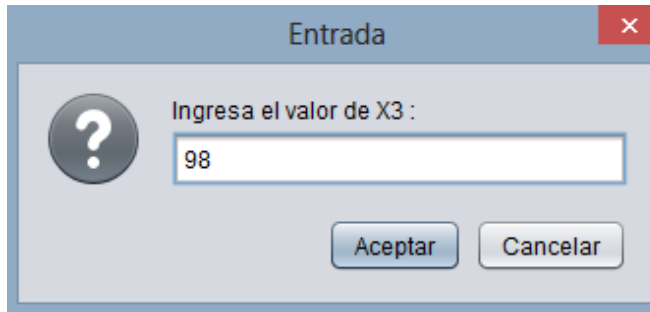
Ingresa el valor de X1 :

65

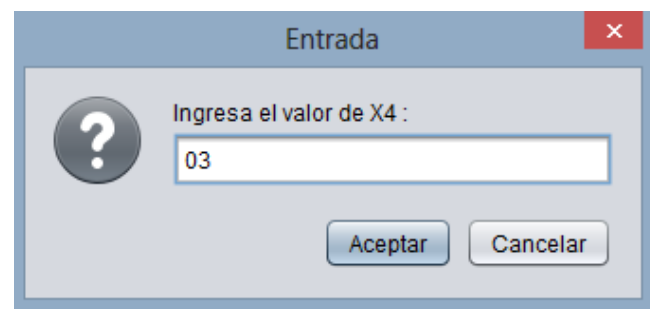
Aceptar Cancelar



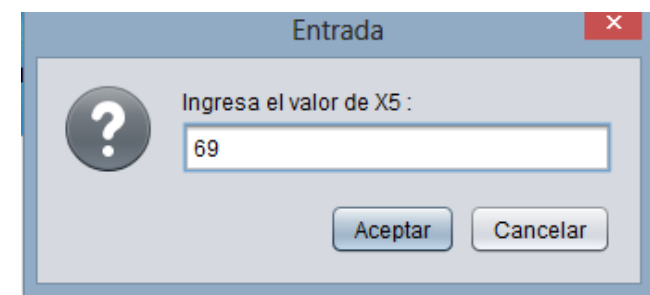
A screenshot of a Windows-style dialog box titled "Entrada" with a red close button in the top right corner. On the left is a circular icon containing a question mark. To the right of the icon is the text "Ingresa el valor de X2 :". Below this text is a text input field containing the number "89". At the bottom right of the dialog are two buttons: "Aceptar" and "Cancelar".



A screenshot of a Windows-style dialog box titled "Entrada" with a red close button in the top right corner. On the left is a circular icon containing a question mark. To the right of the icon is the text "Ingresa el valor de X3 :". Below this text is a text input field containing the number "98". At the bottom right of the dialog are two buttons: "Aceptar" and "Cancelar".



A screenshot of a Windows-style dialog box titled "Entrada" with a red close button in the top right corner. On the left is a circular icon containing a question mark. To the right of the icon is the text "Ingresa el valor de X4 :". Below this text is a text input field containing the number "03". At the bottom right of the dialog are two buttons: "Aceptar" and "Cancelar".



A screenshot of a Windows-style dialog box titled "Entrada" with a red close button in the top right corner. On the left is a circular icon containing a question mark. To the right of the icon is the text "Ingresa el valor de X5 :". Below this text is a text input field containing the number "69". At the bottom right of the dialog are two buttons: "Aceptar" and "Cancelar".

### CONCLUSIONES

En la nota de arriba, hable un poco del funcionamiento, se escriben la cantidad de números a generar, el modulo y la cantidad de elementos que disponemos, al pulsar el botón generar procedemos a ingresar los números uno por uno, para que finalmente imprima el resultado de ejecutar el algoritmo en cada iteración.

Los resultados son los mismos, así como en los programas anteriores, su funcionamiento es correcto.

La interfaz gráfica, hace que el usuario final pueda interactuar de manera más sencilla al momento de calcular números pseudoaleatorios.



## ALGORITMO CONGRUENCIAL NO LINEAL

### DESCRIPCIÓN

Algoritmo congruencial cuadrático

Este algoritmo tiene la siguiente ecuación recursiva  $X_{i+1} = (AX^2_i + BX_i + C) \bmod(M)$   $i = 0, 1, 2, 3, \dots, N$

En este caso, los  $r_i$ , puede ser generado con la ecuación  $r_i = \frac{X_i}{(m-1)}$  de acuerdo con Lecuyer, las condiciones que deben cumplir los parámetros  $m$ ,  $a$ ,  $b$  y  $c$  para alcanzar un periodo de vida máximo de  $N = m$  son:

$$\begin{aligned} m &= 2^g \\ a &\text{ debe de ser un numero par} \\ c &\text{ debe de ser un numero impar} \\ g &\text{ debe ser entero} \\ (b-1) \bmod 4 &= 1 \end{aligned}$$

### OBJETIVO

Desarrollar un programa, que ejecute el algoritmo congruencial cuadrático, conociendo los diferentes parámetros que ayudarán al correcto funcionamiento, los cuales son la semilla inicial  $x_0$ ,  $m$ ,  $a$ ,  $b$  y  $c$ . Estos valores se usarán en la fórmula para calcular los números pseudoaleatorios mediante una serie de iteraciones, algunos valores permanecerán constantes y otros tomarán nuevos valores, es necesario comprender previamente la lógica de este algoritmo, en el momento de la programación.

Programar una GUI para que el usuario final pueda interactuar de una manera sencilla y efectiva, y logre obtener los números pseudoaleatorios que desea calcular con el algoritmo congruencia cuadrático.

### DESARROLLO

Generar, a partir del algoritmo congruencial cuadrático, suficientes números enteros hasta alcanzar el periodo de vida, para esto considere los parámetros  $X_0 = 13$ ,  $m = 8$ ,  $a = 26$ ,  $b = 27$  y  $c = 27$ . Como todas las condiciones estipuladas para los parámetros se satisfacen, es de esperarse que el periodo de vida del generador sea  $N = m = 8$ , tal como podrá comprobar al revisar los cálculos correspondientes, que se presentan a continuación.

Solución:

$$X_1 = (26 \cdot 13^2 + 27 \cdot 13 + 27) \bmod(8) = 4$$

$$X_2 = (26 \cdot 4^2 + 27 \cdot 4 + 27) \bmod(8) = 7$$

$$X_3 = (26 \cdot 7^2 + 27 \cdot 7 + 27) \bmod(8) = 2$$

$$X_4 = (26 \cdot 2^2 + 27 \cdot 2 + 27) \bmod(8) = 1$$

$$X_5 = (26 \cdot 1^2 + 27 \cdot 1 + 27) \bmod(8) = 0$$

$$X_6 = (26 \cdot 0^2 + 27 \cdot 0 + 27) \bmod(8) = 3$$

$$X_7 = (26 \cdot 32 + 27 \cdot 3 + 27) \bmod (8) = 6$$

$$X_8 = (26 \cdot 62 + 27 \cdot 6 + 27) \bmod (8) = 5$$

$$X_9 = (26 \cdot 52 + 27 \cdot 5 + 27) \bmod (8) = 4$$

## CÓDIGO

```
package GNPpseudoAleatorios;

import java.math.BigDecimal;
import java.math.RoundingMode;
import javax.swing.JOptionPane;

public class AlgoritmoCongruencialCuadratico extends javax.swing.JFrame {

    public AlgoritmoCongruencialCuadratico() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel4 = new javax.swing.JLabel();
        Semilla = new javax.swing.JTextField();
        a = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        b = new javax.swing.JTextField();
        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        Generar = new javax.swing.JButton();
        Clear = new javax.swing.JButton();
        jLabel10 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        r1 = new javax.swing.JTextPane();
        jScrollPane2 = new javax.swing.JScrollPane();
        x1 = new javax.swing.JTextPane();
        jLabel5 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        m = new javax.swing.JTextField();
        c = new javax.swing.JTextField();
        jScrollPane3 = new javax.swing.JScrollPane();
        div = new javax.swing.JTextPane();
        jLabel7 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
    }
}
```

```

Exit = new javax.swing.JButton();
back = new javax.swing.JButton();
jLabel11 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setUndecorated(true);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel4.setText("Ingrese la semilla:");
getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 100, -1, -1));
getContentPane().add(Semilla, new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 100, 220, -1));
getContentPane().add(a, new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 140, 220, -1));

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel3.setText("Ingrese el parámetro a:");
getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(70, 140, -1, -1));
getContentPane().add(b, new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 210, 220, -1));

jPanel1.setBackground(new java.awt.Color(153, 255, 204));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
jLabel2.setText("Algoritmo Congruencial Cuadrático");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
50, -1, 30));

Generar.setText("Generar");
Generar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        GenerarActionPerformed(evt);
    }
});
jPanel1.add(Generar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 260, -1, -1));

Clear.setText("Limpiar");
Clear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClearActionPerformed(evt);
    }
});

```

```

jPanel1.add(Clear, new org.netbeans.lib.awtextra.AbsoluteConstraints(530,
310, 70, -1));

jLabel10.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel10.setText("Ingresa el parámetro b:");
jPanel1.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
210, -1, -1));

jScrollPane1.setViewportView(ri);

jPanel1.add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(360, 280, 120, 180));

jScrollPane2.setViewportView(xi);

jPanel1.add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 280, 120, 180));

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel5.setText("Ingresa el parámetro m:");
jPanel1.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
240, -1, -1));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setText("Ingresa el parámetro c:");
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70,
180, -1, -1));
jPanel1.add(m, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 240,
220, -1));
jPanel1.add(c, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 180,
220, -1));

jScrollPane3.setViewportView(div);

jPanel1.add(jScrollPane3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 280, 120, 180));

jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
jPanel1.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0,
640, 20));

jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N

```

```

        jPanel1.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(620,
20, 20, 450));

        jLabel9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
        jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
20, 20, 450));

        jLabel8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/a.jpg"))
); // NOI18N
        jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
470, 640, 20));

        Exit.setBackground(new java.awt.Color(255, 0, 0));
        Exit.setText("Salir");
        Exit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ExitActionPerformed(evt);
            }
        });
        jPanel1.add(Exit, new org.netbeans.lib.awtextra.AbsoluteConstraints(550, 20,
70, -1));

        back.setBackground(new java.awt.Color(255, 255, 51));
        back.setText("Volver");
        back.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                backActionPerformed(evt);
            }
        });
        jPanel1.add(back, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
440, 70, -1));

        jLabel11.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/GNPpseudoAleatorios/Img/5.jpg"))
); // NOI18N
        jPanel1.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(20,
20, 600, 450));

        getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 640, 500));

        pack();
    } // </editor-fold>

```

```

private void GenerarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    if (evt.getSource()==Generar)
    {

        if(Semilla.getText().equals("")) //***** Permite verificar si la caja de texto
        esta vacia *****//
        {
            JOptionPane.showMessageDialog(null,"Ingresa valor de semilla");
        }

        if(a.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro a:");
        }

        if(c.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro c:");
        }
        if(m.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro m:");
        }
        if(b.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null,"Ingrese el parámetro m:");
        }

        String semilla1=Semilla.getText();
        Integer A= new Integer (a.getText());
        Integer C= new Integer(c.getText());
        Integer M= new Integer(m.getText());
        Integer B= new Integer(b.getText());

        String snumero2, snumero3, snumero4, cadenaxi = "",cadenari =
        "",cadenadiv="";
        int tam1, tam3, i, primerc, MR ;
        long numero1, numero3, formula;
        int xicuadrado;
        tam1 = semilla1.length();
        numero1 = Integer.parseInt(semilla1);
        MR=M-1;

        if(numero1<1)

```

```

{
    JOptionPane.showMessageDialog(null,"Valor de la semilla debe ser mayor ó
igual a 1");
    Semilla.setText("");
    a.setText("");
    c.setText("");
    m.setText("");
    ri.setText("");
    xi.setText("");
    div.setText("");
    b.setText("");

}try
{
    if(numero1>=1){
        for (i=1; i<=M; i++){
            xicuatado =(int)Math.pow(numero1,2);
            formula = ((A*xicuatado+B*numero1+C)%M);

            snumero2 = Long.toString(formula);
            numero3 = Integer.parseInt(snumero2);
            double rs= (double) numero3/MR;
            BigDecimal formatNumber = new BigDecimal(rs);
            formatNumber = formatNumber.setScale(4, RoundingMode.DOWN);

            cadenaxi+=("x"+i+ " = " +snumero2+" \n");

            cadenadiv+=(snumero2+"/"+ MR +" \n");

            cadenari+=( "r"+i+ " = " + formatNumber + " \n");

            numero1=Integer.parseInt(snumero2);

        }
        xi.setText(cadenaxi);
        ri.setText(cadenari);
        div.setText(cadenadiv);
    }

}catch(java.lang.NumberFormatException e)
{
    JOptionPane.showMessageDialog(null,"Dato invalido");
    Semilla.setText("");
    a.setText("");

```

```

        c.setText("");
        m.setText("");
        ri.setText("");
        xi.setText("");
        div.setText("");
        b.setText("");
    }
}
}

private void ClearActionPerformed(java.awt.event.ActionEvent evt) {

    Semilla.setText(null);
    a.setText(null);
    c.setText(null);
    ri.setText(null);
    xi.setText(null);
    div.setText(null);
    m.setText(null);
    b.setText(null);
}

private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Menú Menú = new Menú();
    Menú.setVisible(true);
    AlgoritmoCongruencialCuadratico.this.dispose();
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {

```



```

        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(AlgoritmoCongruencialCuadratico.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(AlgoritmoCongruencialCuadratico.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(AlgoritmoCongruencialCuadratico.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(AlgoritmoCongruencialCuadratico.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new AlgoritmoCongruencialCuadratico().setVisible(true);
    }
});

```

```

}

// Variables declaration - do not modify
private javax.swing.JButton Clear;
private javax.swing.JButton Exit;
private javax.swing.JButton Generar;
private javax.swing.JTextField Semilla;
private javax.swing.JTextField a;
private javax.swing.JTextField b;
private javax.swing.JButton back;
private javax.swing.JTextField c;
private javax.swing.JTextPane div;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextField m;
private javax.swing.JTextPane ri;
private javax.swing.JTextPane xi;
// End of variables declaration
}

```

## EJEMPLO

**Algoritmo Congruencial Cuadratico**

Salir

Ingrese la semilla: 13

Ingrese el parámetro a: 26

Ingrese el parámetro c: 27

Ingrese el parámetro b: 27

Ingrese el parámetro m: 8

Generar

Limpiar

Volver

x1 = 4	4/7	r1 = 0.5714
x2 = 7	7/7	r2 = 1.0000
x3 = 2	2/7	r3 = 0.2857
x4 = 1	1/7	r4 = 0.1428
x5 = 0	0/7	r5 = 0.0000
x6 = 3	3/7	r6 = 0.4285
x7 = 6	6/7	r7 = 0.8571
x8 = 5	5/7	r8 = 0.7142

## CONCLUSIONES

Para el correcto funcionamiento de nuestro programa, es necesario llenar cada uno de los campos requeridos, semilla ( $X_0$ ), y los parámetros a, c, b y m. M nos indicará la vida de nuestro algoritmo, en este caso 8 iteraciones, una vez que los datos están escritos en los campos, al pulsar el botón generar, procederá a tomar el valor de cada casilla y ejecutar la formula una y otra vez, calculando cada valor, y sustituyendo los necesarios en la formula según el funcionamiento de nuestro algoritmo. Una vez que haya terminado, imprimirá las 3 cadenas donde almacenamos los valores de  $x_i$ , las divisiones de  $x_i$  entre  $m-1$ , y los números  $r_i$  generados en cada iteración de nuestro algoritmo.

Usamos la interfaz gráfica para facilitar al usuario final el uso de nuestro programa, y pueda calcular los números pseudoaleatorios de una manera que no necesite usar la consola o interactuar con ella.

Si comparamos los resultados de  $r_i$ , con los resultados previamente obtenidos, podemos concluir que ambos son iguales, así que nuestro programa ejecuta el algoritmo de manera correcta.

## CONCLUSIONES

La generación de números pseudoaleatorios es nuestro tema en esta unidad, para ello vimos previamente en clase diferentes métodos para la obtención de números, algunos muy fáciles, y otros un poco más complicado, por estos últimos me refiero a los que usan la función “mod”, una vez que comprendimos la lógica detrás de cada algoritmo, procedimos a programar cada uno de estos algoritmos en un programa para cada uno de ellos.

En nuestro caso, ya que decidimos crear interfaces gráficas en cada programa, en primer paso creamos un menú que nos envía a cada programa con un simple clic al botón que contenga el nombre del algoritmo que desea usar, una vez terminado el menú, procedimos a crear visualmente los componentes de cada programa (algoritmo), creamos los labels, los textfields, los textareas, los buttons, los primeros botones que se programaron fueron los que devuelven al menú y los que terminan el programa, cerrando la ventana.

Una vez que terminamos el diseño básico, comenzamos a programar, los valores que se obtienen de los textfields, son almacenados en variables gracias a la conversión a números, después de obtener el valor mediante el método `getText()`, después codificamos el correcto funcionamiento de nuestros programas, esto fue fácil ya que la fórmula del algoritmo, nos está dando la información necesaria para programar de manera fácil, fue necesario usar iteraciones, en nuestro caso usamos el ciclo `for`, ya que las iteraciones se repiten una tras otra, las variables cambian al final de cada ciclo, ya que en la mayoría de los programas al calcular el primer número pseudoaleatorio, este sustituye algún valor en nuestra fórmula, y así hasta terminar el ciclo de vida de nuestro programa. Los resultados que se muestran en cada textarea, son gracias a que guardamos cada número  $x_i$  y  $r_i$  en una cadena para posteriormente imprimirlo con un método `setText()`. Una vez que se terminó el cálculo de números pseudoaleatorios, si es necesario volver a empezar, simplemente damos clic al botón limpiar, y podemos volver a escribir parámetros para calcular nuevamente números nuevos.

Si las casillas no se llenan o se llenan incorrectamente, nos saldrán mensajes de error indicando que está mal, de esta forma validamos nuestro programa.

La desventaja de usar interfaces gráficas, fue que nuestro código es muy extenso, pero vale la pena, ya que, en contraparte, el usuario final de nuestro programa, puede interactuar fácilmente con nuestro programa, y calcular números pseudoaleatorios de forma sencilla y eficaz.

## BIBLIOGRAFÍA

García Dunna, E., García Reyes, H., & Cardenás Barrón, L. E. (2013). Simulación y Análisis con Promodel 2° Edición. México: Pearson.

W, D., Sadowski, R., & Sturrock, D. (2008). *Simulación con software arena 4° Edición*. México: McGrawHill.