

Juego de la Vida y Analizador

Frías Mercado Carlos Elliot

September 23, 2017

3CM6

Computing Selected Topics

Prof: Genaro Juárez Martínez

Contents

1	Juego de la Vida	3
1.1	Descripción del programa	3
1.2	Pruebas del funcionamiento	3
1.3	Código fuente	6
1.3.1	Clase Universo	6
1.3.2	Clase Dios	11
1.3.3	Clase Escritor	12
1.3.4	Clase Vida	14
2	Analizador	16
2.1	Descripción del programa	16
2.2	Pruebas del funcionamiento	16
2.3	Código fuente	18
2.3.1	Clase Escritor	18
2.3.2	Clase Divisor	20
2.3.3	Clase Analiza	22

1 Juego de la Vida

1.1 Descripción del programa

Este programa permite simular el algoritmo conocido como Juego de la Vida de Conway. Contiene células las cuales se representan en el tablero a través de un cuadrito de un color a elegir, estando algunas vivas y otras muertas, además se puede controlar la velocidad con la que

cada iteración va cambiando. Se puede iniciar la simulación desde un archivo precargado, dibujada por el usuario o aleatoriamente. El tamaño máximo del tablero (o cantidad máxima de células) es de 1000 y el mínimo es de 10.

1.2 Pruebas del funcionamiento

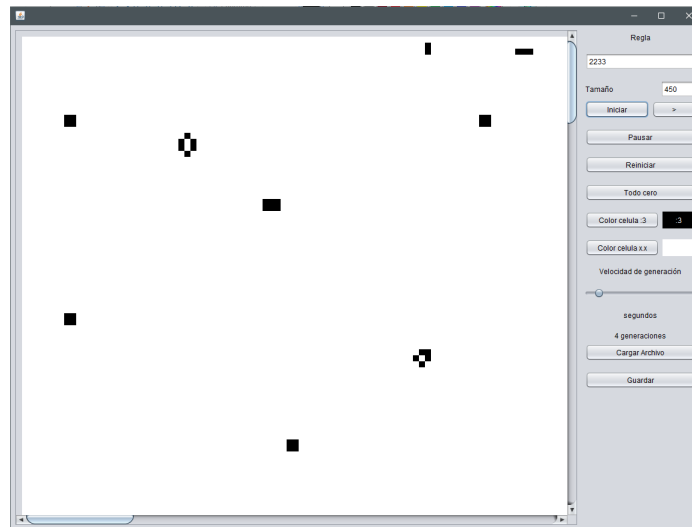


Figura 1: Inicialización del simulador con valores aleatorios

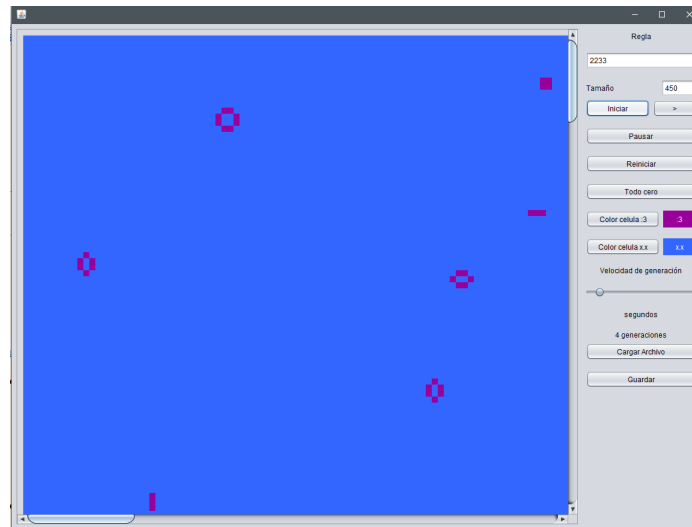


Figura 2: Cambio de colores del programa

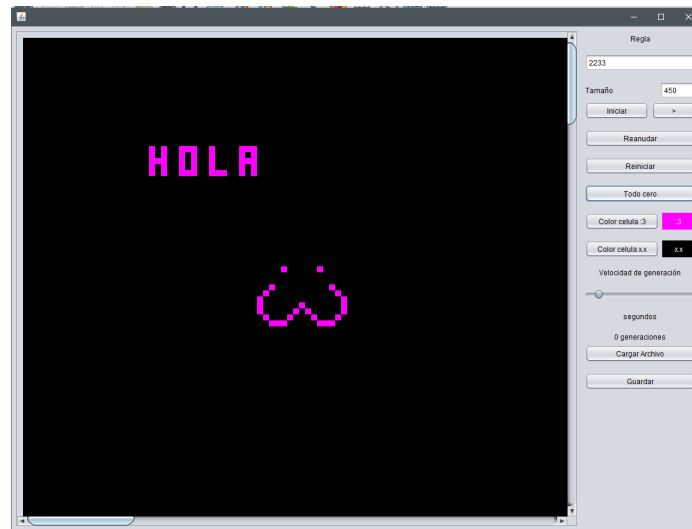


Figura 3: Patrón introducido por el usuario

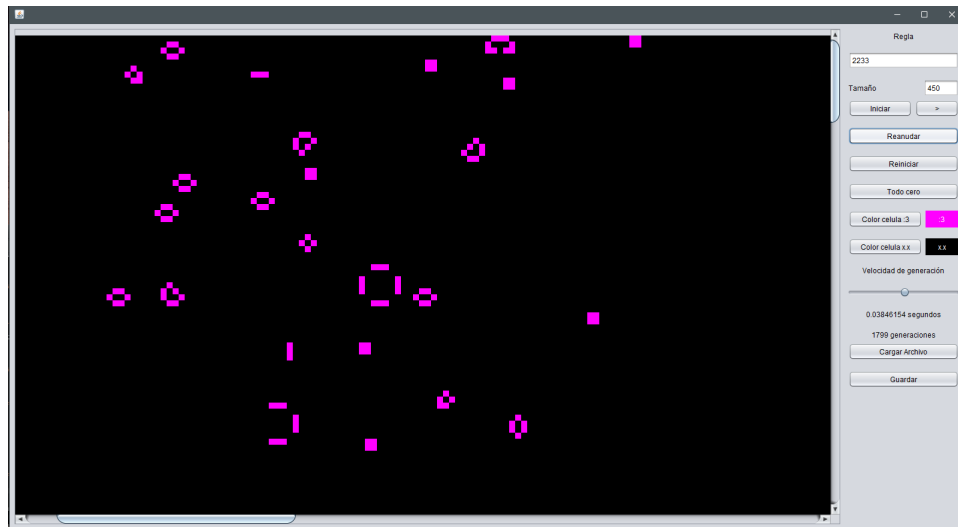


Figura 4: Patrón del usuario tras 1800 Iteraciones

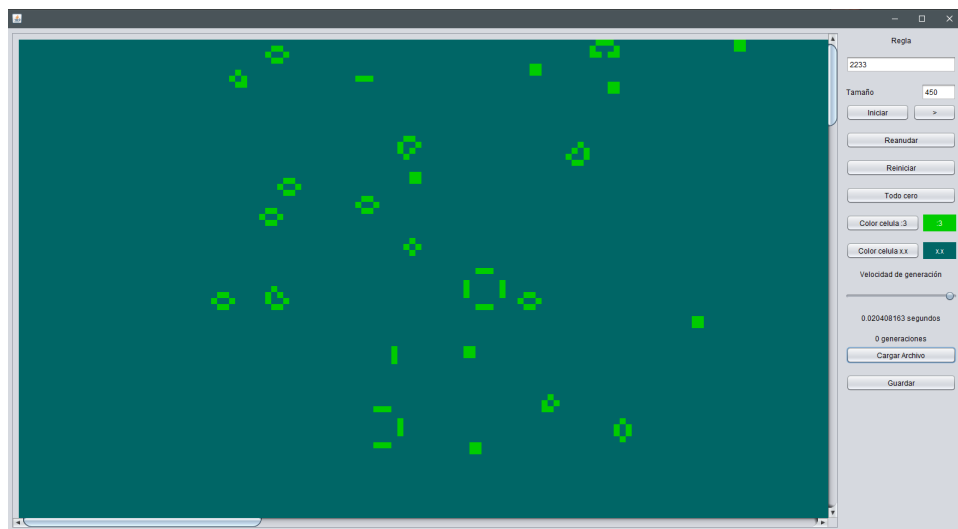


Figura 5: Formato guardado en un archivo y luego recargado

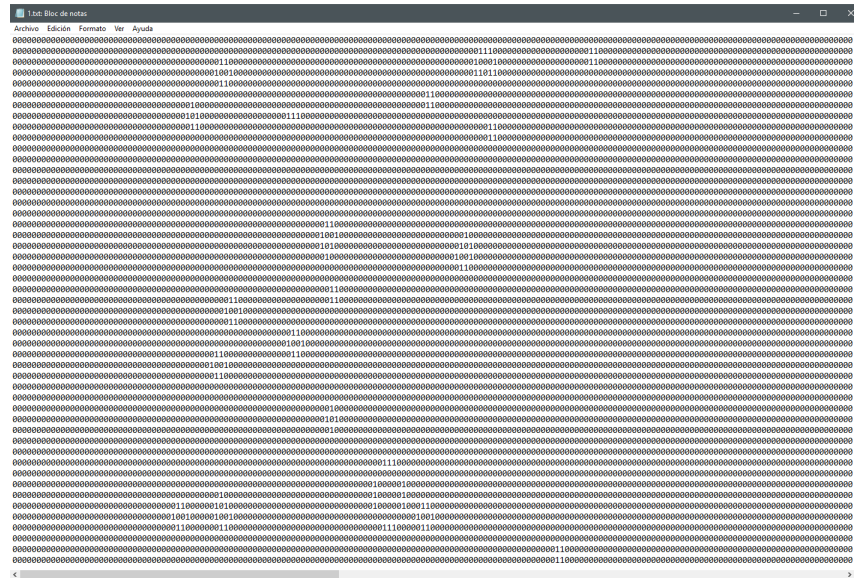


Figura 6: Aspecto del archivo guardado

1.3 Código fuente

1.3.1 Clase Universo

```
/* * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor. */
package gol;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Point;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseWheelEvent;
import java.awt.event.MouseWheelListener;
import javax.swing.JPanel;
import logica.Dios;
import logica.Vida;
/** * * @author Maku */
class Universo extends JPanel {
    int[][] m;

    int tam, click, tamCelula = 10, tiempo; //tam = num de celulas
```

```

        private Graphics2D g2d;
        Color viva, muerta;
        int[] regla = new int[4];
        Dios gg = new Dios();
        private int zoom = 0;
        private static final double ZOOM_AMOUNT = 1.1;
        Universo () {
        Universo(int[] regla, int tam, int tiempo, Color vi, Color mu)
        {
            this.regla = regla;
            this.tam = tam;
            setPreferredSize(new Dimension(tam*tamCelula,tam*tamCelula)); //x, y
            m = gg.creaVida(tam);
            this.tiempo = tiempo;
            viva = vi;
            muerta = mu;
            this.addMouseListener(new MouseListener()
            {
                @Override
                public void mouseClicked(MouseEvent e)
                {
                    int x, y;
                    Point p = e.getPoint();
                    y = (int) Math.floor((float)p.getX()/tamCelula);
                    x = (int) Math.floor((float)p.getY()/tamCelula);
                    if(x < tam && y < tam)
                    {
                        if(m[x][y] == 1)
                        {
                            m[x][y] = 0;
                        }
                        else
                        {
                            m[x][y] = 1;
                        }
                        click = 1;
                        repaint();
                    }
                }

                @Override
                public void mousePressed(MouseEvent e) {
                    //throw new UnsupportedOperationException("Not supported yet.");
                    //To change body of generated methods, choose Tools | Templates.
                }
            }
        }
    }

```

```

@Override
public void mouseReleased(MouseEvent e) {
    // throw new UnsupportedOperationException("Not supported yet.");
    //To change body of generated methods, choose Tools | Templates.
}

@Override
public void mouseEntered(MouseEvent e) {
    // throw new UnsupportedOperationException("Not supported yet.");
    //To change body of generated methods, choose Tools | Templates.
}

@Override
public void mouseExited(MouseEvent e) {
    // throw new UnsupportedOperationException("Not supported yet.");
    //To change body of generated methods, choose Tools | Templates.
}

Universo(int[] regla, int tam, int tiempo, int cero, Color vi, Color mu)
{
    this.regla = regla;
    this.tam = tam;
    setPreferredSize(new Dimension(tam*tamCelula,tam*tamCelula)); //x, y
    m = gg.destruyeVida(tam);
    this.tiempo = tiempo;
    viva = vi;
    muerta = mu;
    this.addMouseListener(new MouseListener() {
        @Override
        public void mouseClicked(MouseEvent e)
        {
            int x, y;
            Point p = e.getPoint();
            y = (int) Math.floor((float)p.getX()/tamCelula);
            x = (int) Math.floor((float)p.getY()/tamCelula);
            if(x < tam && y < tam)
            {
                if(m[x][y] == 1)
                {
                    m[x][y] = 0;
                }
                else
                {
                    m[x][y] = 1;
                }
                click = 1;
            }
        }
    });
}

```



```

repaint();
}
}

@Override
public void mousePressed(MouseEvent e) {
//throw new UnsupportedOperationException("Not supported yet."); //To change body of ge

@Override
public void mouseReleased(MouseEvent e) {
// throw new UnsupportedOperationException("Not supported yet."); //To change body of g

@Override
public void mouseEntered(MouseEvent e) {
// throw new UnsupportedOperationException("Not supported yet."); //To change body of

@Override
public void mouseExited(MouseEvent e) {
throw new UnsupportedOperationException("Not supported yet."); //To change body of gene
}

Universo(int[] regla, int tam, int tiempo, int[][] car, Color vi, Color mu)
{
this.regla = regla;
this.tam = tam-2;

System.out.println(this.tam);
setPreferredSize(new Dimension(tam*tamCelula,tam*tamCelula)); //x, y
m = car;
this.tiempo = tiempo;
viva = vi;
click = 1;
muerta = mu;
this.addMouseListener(new MouseListener() {
@Override
public void mouseClicked(MouseEvent e)
{
int x, y;
Point p = e.getPoint();
y = (int) Math.floor((float)p.getX()/tamCelula);
x = (int) Math.floor((float)p.getY()/tamCelula);
if(x < tam && y < tam)
{
if(m[x][y] == 1)
{
m[x][y] = 0;

```

```

    }
    else
    {

        m[x][y] = 1;
    }
    click = 1;
    repaint();
    }
    }

    @Override
    public void mousePressed(MouseEvent e) {
        //throw new UnsupportedOperationException("Not supported yet."); //To change body of ge

    @Override
    public void mouseReleased(MouseEvent e) {
        // throw new UnsupportedOperationException("Not supported yet."); //To change body of g

    @Override
    public void mouseEntered(MouseEvent e) {
        // throw new UnsupportedOperationException("Not supported yet."); //To change body of

    @Override
    public void mouseExited(MouseEvent e) {
        // throw new UnsupportedOperationException("Not supported yet.");
        //To change body of generated methods, choose Tools | Templates.

    });
    }
    @Override
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);
        if(click == 1)
        {
            click = 0;
            // repaint();
        }
        else
        {
            m = analiza(m, regla);
        }
        g2d = (Graphics2D) g;
        for(int i = 1; i <= tam; i++)
        {
            for(int j = 1; j <= tam; j++)

```

```

if(m[i][j] == 1)
{
g2d.setColor(viva);
g2d.fillRect(tamCelula*j, tamCelula*i, tamCelula, tamCelula); //x, y, tamx, tamy
g2d.drawRect(tamCelula*j, tamCelula*i, tamCelula, tamCelula); //x, y, tamx, tamy //x, y
}
else
{
g2d.setColor(muerta);
g2d.fillRect(tamCelula*j, tamCelula*i, tamCelula, tamCelula); //x, y, tamx, tamy
g2d.drawRect(tamCelula*j, tamCelula*i, tamCelula, tamCelula); //x, y, tamx, tamy //x, y
}
}
}
// repaint();
try
{
//System.out.println("tiempo " +tiempo);
Thread.sleep(tiempo);
}
catch (Exception e) {}
} //paint component
public int[][] analiza(int[][]m, int[] regla)
{
Vida v = new Vida();
//System.out.println("hay"+regla[0]);
return v.existe(m,regla, tam);
}

public int[][] getMatriz()
{
return m;
}
}
} //class

```

1.3.2 Clase Dios

```

/* * To change this license header, choose License Headers in Project Properties. * T
import java.util.Random;
/** * * @author Maku */
public class Dios
{
public Dios(){}
public int[][] creaVida(int tam)
{
int[][] inicio = new int[tam + 2][tam + 2];
Random r = new Random();
for(int i = 1; i <= tam; i++)

```

```

{
for(int j = 1; j <= tam; j++)
{
inicio[i][j] = r.nextInt(3);

}
}
return inicio;
}
public int[][] destruyeVida(int tam)
{

int[][] fin = new int[tam + 2][tam + 2];
    for(int i = 1; i <= tam; i++)
    {
        for(int j = 1; j <= tam; j++)
        {
            fin[i][j] = 0;
        }
    }
    return fin;
}
}

```

1.3.3 Clase Escritor

```

/* * To change this license header, choose License Headers in Project Properties. * T
| Templates * and open the template in the editor. */ package logica;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.Character.getNumericValue;
/** * * @author Maku */
public class Escritor
{
public Escritor()
{
}
public int escribe(int[][] m)
{
try
{

PrintWriter writer = new PrintWriter("1.txt", "UTF-8");

for(int i = 0; i < m.length; i++)

```

```

{
    for(int j = 0; j < m.length; j++)
    {
        writer.print(m[i][j]);
    }
    writer.println("");
}
    writer.close();
    return 1;
}
    catch (IOException e)
    {
        return 0;
    }
}
}
public int[][] lee(File f) throws FileNotFoundException, IOException
{
    int[][] x;
    int i = 0, j = 0, tam = 0;
    String c = "", d;
    if(f!=null)
    {
        FileReader archivos=new FileReader(f);
        BufferedReader lee=new BufferedReader(archivos);
        //c+= lee.readLine();
        while((d = lee.readLine()) !=null)
        {
            tam++;
            c += d;
            /System.out.println("c es "+c);
        }
        lee.close();
        x = new int[tam][tam];
        for(int k = 0; k < c.length(); k++)
        {
            String car = "";
            car += c.charAt(k);
            System.out.print("["+car+"]");
            if(j == tam )
            {
                System.out.println("");
                i++;
                j = 0;
                k--;
            }
        }
    }
}

```

```

else
{
x[i][j] = Integer.parseInt(car);
j++;
}
} //while2
return x;
}

//if
else
{
x = new int[10][10];
for(i = 0; i < 10; i++)
{
for(j = 0; j < 10; j++)
{
x[i][j] = 0;
}
}
return x;
}
} //funcion lee
}

```

1.3.4 Clase Vida

```

/* * To change this license header, choose License Headers in Project Properties. *
/** * * @author Maku */
import java.io.*; import java.util.*; public class Vida
{
public Vida()
{

}

public int[][] existe (int[][] m, int[] regla, int tam)
{
int fila, colum, vec;
fila = tam;
colum = fila;
//Random vida = new Random();
//Integer[][] m = new Integer[fila+2][colum+2];
int[][] aux = new int[fila+2][colum+2];
// Integer[][][] saves = new Integer[fila+2][colum+2][10]; //[fila][columna][indice]
// Integer[][] saves2 = new Integer[fila+2][colum+2]; //[fila][columna][indice]

for(int i = 1; i <= fila; i++) //analizar vida
{
for(int j = 1; j <= colum; j++)
{

```

```

vec = 0;
vec += m[i-1][j-1] + m[i-1][j] + m[i-1][j+1]; //las de arriba
vec += m[i][j-1] + m[i][j+1]; //las de al lado
vec += m[i+1][j-1] + m[i+1][j] + m[i+1][j+1]; //las de abajo
if(m[i][j] == 1) //[fila][columna] Si está viva entonces:
{

if(vec < regla[0]) //morir de aislamiento :c Smin
{
aux[i][j] = 0;
}
if(vec > regla[2]) //morir de sobrepoblacion
Smax
{
aux[i][j] = 0;
}
if(vec >= regla[1] && vec <= regla[2]) //sobrevivir :3
{
aux[i][j] = 1;
}
}
else //si está muerta
{
if(vec == regla[3]) //nacer *3* //Tiene 3 vecinos B

{
aux[i][j] = 1;
}
else
{
aux[i][j] = 0;
}

} //else muerta
} //for columna

} //for fila analizar vida
return aux;
} //main } //clase

```

2 Analizador

2.1 Descripción del programa

Este programa se encarga de meterse dentro del juego de la vida (únicamente con 10 iteraciones) y trata de encontrar dentro de él los 3 tipos principales de elementos que se pueden formar (Glider, Still Life o Flip Flop). Lo realiza dividiendo todo el universo de células vivas y muertas en pequeñas secciones de 3x3 dentro de las cuales analiza el patrón de comportamiento obtenido al sumar un arreglo tridimensional de cada iteración de células vivas.

2.2 Pruebas del funcionamiento



Figura 7: Inicio del simulador y analizador aleatoriamente.



Figura 8: Análisis final generado tras las 10 iteraciones.

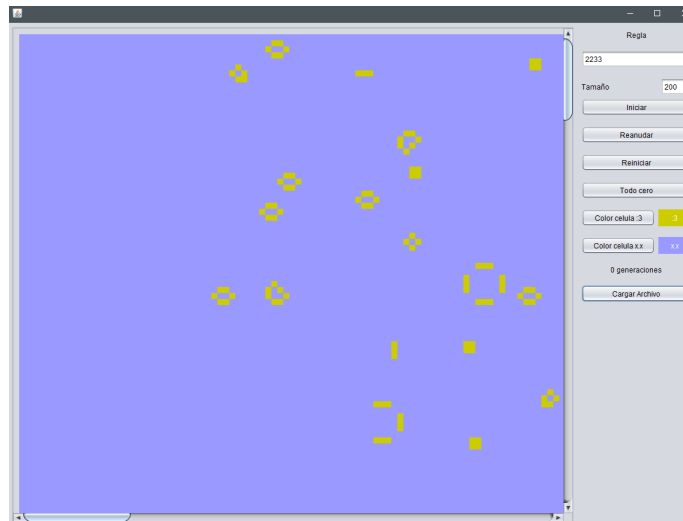


Figura 9: Archivo generado en el programa anterior, cargado dentro de éste y con nuevos colores.

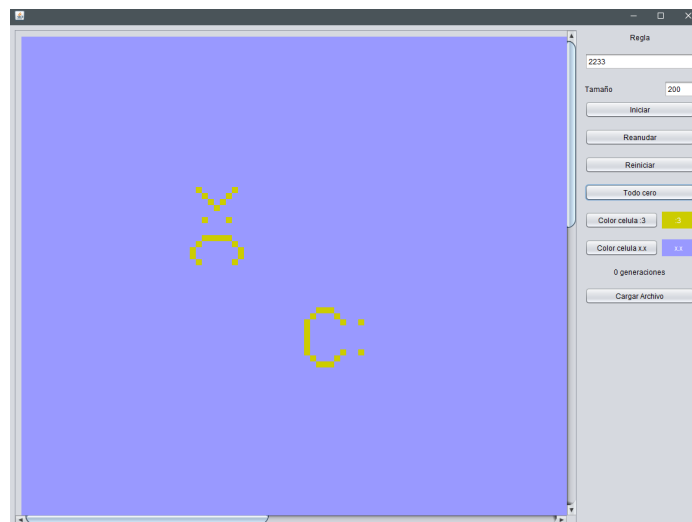


Figura 10: Sistema configurado por el usuario.

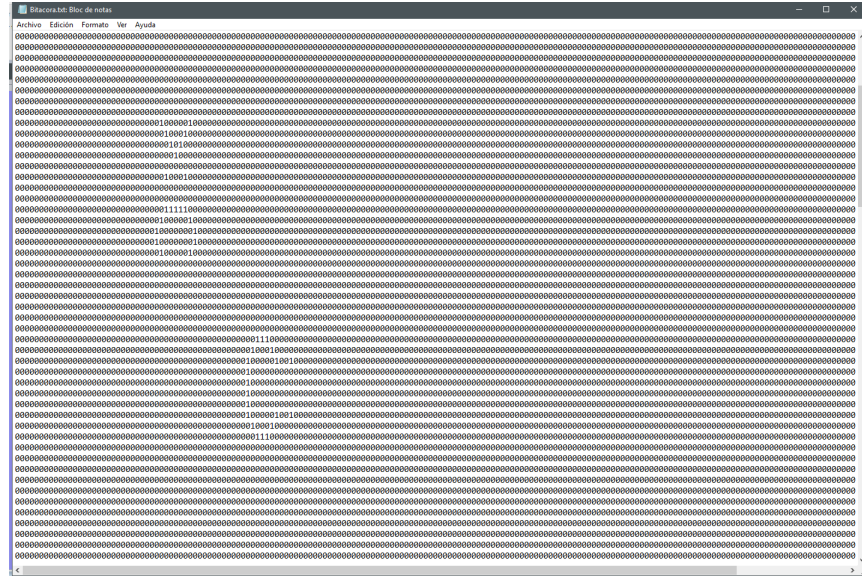


Figura 11: Archivo generado al guardar el analizador.

2.3 Código fuente

2.3.1 Clase Escriitor

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package logica;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.Character.getNumericValue;
/** * *
@author Maku */
public class Escriitor
{
    public Escriitor()
    {
    }

    public int escribe(int[][] m, int ff, int gl, int sl)
    {

```

```

try
{
    PrintWriter writer = new PrintWriter("Bitacora.txt", "UTF-8");
    for(int i = 0; i < m.length; i++)
    {
        for(int j = 0; j < m.length; j++)
        {
            if(m[i][j] == 1
            {
                writer.print(m[i][j]);
            }
            else
            {
                writer.print(0);
            }
        }
        writer.println("");
    }
    writer.println("Con 10 iteraciones existen");
    writer.println("" + ff + " posibles Flip Flops");
    writer.println(" " + gl + " posibles Gliders");
    writer.println(" " + sl + " posibles Still Lives");
    writer.close();
    return 1;
}
catch (IOException e)
{
    return 0;
}
}

public int[][] lee(File f) throws FileNotFoundException, IOException
{
    int[][] x;
    int i = 0, j = 0, tam = 0;
    String c = "", d;
    if(f!=null)
    {
        FileReader archivos=new FileReader(f);
        BufferedReader lee=new BufferedReader(archivos);
        //c+= lee.readLine();
        while((d = lee.readLine()) !=null)
        {
            tam++;
            c += d;
            //System.out.println("c es "+c);
        }
    }
}

```

```

        lee.close();
        x = new int[tam][tam];
        for(int k = 0; k < c.length(); k++)
    {
        String car = "";
        car += c.charAt(k);
        System.out.print("["+car+"]");
        if(j == tam )
        {
            System.out.println("");
            i++;
            j = 0;
            k--;
        }
        else
        {
            x[i][j] = Integer.parseInt(car);
            j++;
        }
    }
}

return x;

} //if
else
{
    x = new int[10][10];
    for(i = 0; i < 10; i++)
    {
        for(j = 0; j < 10; j++)
        {
            x[i][j] = 0;
        }
    }
    return x;
}

} //funcion lee
}

```

2.3.2 Clase Divisor

```

/* * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package logica;
import analizador.biatcora;

```

```

/** * * @author Maku */
public class Divisor
{
    int[][][] saves;
    int[][] saves2, ini; int sl = 0, gl = 0, ff = 0;
    int[][] mini = new int[3][3];
    int i0 = 1, i, j0 = 1, j, fil = 1, col = 1, tamSecc = 3, x = 0, y = 0, tam, gens; //mm,
    Analiza an = new Analiza();
    public Divisor(int[][][] saves, int tam, int gens, int[][] in)
    {
        this.saves = saves;
        this.tam = tam;
        this.gens = gens;
        this.ini = in;
        saves2 = new int[saves.length][saves.length];
    }
    public void analisis() {
        for(int ii = 1; ii <= tam; ii++)
        {
            for(int jj = 1; jj <= tam; jj++)
            {
                saves2[ii][jj] = 0; //inicializo matriz de analisis
            }
        }
        for(int ii = 1; ii <= tam; ii++) {
            for(int jj = 1; jj <= tam; jj++) {
                for(int kk = 0; kk < gens; kk++) {
                    saves2[ii][jj] += saves[ii][jj][kk]; //lleno mi matriz de analisis
                }
            }
        }
        for(int li = 1; li <= tam; li++)
        { for(int lj = 1; lj <= tam; lj++)
        {
            System.out.print(saves2[li][lj]);
        }
        System.out.print("\n"); } //imprimo matriz de analisis
        while(j0 <= tam)
        {
            if(i0 <= tam + 1) {
                for(i = i0; i < (fil*tamSecc+1); i++) //dividiendo la matriz
                {
                    if(i > tam) { break; }
                    for(j = j0 ; j < (col*tamSecc+1); j++)
                    { if(j > tam) { break; } //
                    System.out.print(i +","+" j +" "+ saves2[i][j]+" ");

```

```

mini[x][y] = saves2[i][j];
y++; //columnas } //
System.out.print("\n");
y = 0;
x++;
if(x > 2) { //
System.out.println("Mini tiene: \n");
/* for(int a = 0; a < 3; a++)
{
for(int b = 0; b < 3; b++)
System.out.print(mini[a][b]);
System.out.print("\n"); }*/
if(an.FF(mini) == 1)
{
ff++;
System.out.println("POSIBLE FLIP FLOP ENCONTRADO\n");
}
if(an.SL(mini) == 1)
{
sl++;
System.out.println("POSIBLE STILL LIFE ENCONTRADO\n");
}
if(an.GL(mini) == 1)
{
gl++;
System.out.println("POSIBLE GLIDER ENCONTRADO\n");
}
x = 0;
}
}
i0 = fil*tamSecc+1;
j0 = 1;
fil++; //
System.out.println("-----");
} else { i0 = 1; j0 = col*tamSecc+1; col++; }
} //while analisis
biatcora b = new biatcora(ff, gl, sl, ini);
b.setVisible(true);
} //metodo analisis
}

```

2.3.3 Clase Analiza

```

/* * To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates

```

```

* and open the template in the editor. */
package logica;
/** * * @author Maku */
public class Analiza
{
    public Analiza()
    {
    }

    public int FF(int[][] matriz)
    {
        int[] FFC = {0, 0, 0}; //{valor, i, j}
        int max = 0;
        int[] maxes = {0, 0, 0, 0};
        while(max < 3)
        {
            for(int i = 0; i < 3; i++)
            {
                for(int j = 0; j < 3; j++)
                {
                    if(matriz[i][j] > max)
                    {
                        max = matriz[i][j];
                        FFC[0] = max;
                        FFC[1] = i;
                        FFC[2] = j;
                    }
                }
            }
        }
        //for de encontrar centro
        if(FFC[0] < 4)
        {
            return 0;
        }
        if(FFC[1] == FFC[2]) //esta en el centro
        {
            if(FFC[0] > matriz[FFC[1]][FFC[2]])
            {
                return 1;
            }
        }
        else //está en los lados
        {
            if(FFC[1] == 1 && FFC[2] == 1)
            {
                if(matriz[FFC[1]][FFC[2]] > matriz[FFC[1]-1][FFC[2]])
                {
                    return 1;
                }
            }
            if(FFC[1] == 1 && FFC[2] == 2)
            {
                if(matriz[FFC[1]][FFC[2]] > matriz[FFC[1]-1][FFC[2]])
                {
                    return 1;
                }
            }
            if(FFC[1] == 2 && FFC[2] == 1)
            {
                if(matriz[FFC[1]][FFC[2]] > matriz[FFC[1]][FFC[2]-1])
                {
                    return 1;
                }
            }
            if(FFC[1] == 2 && FFC[2] == 2)
            {
                if(matriz[FFC[1]][FFC[2]] > matriz[FFC[1]][FFC[2]-1])
                {
                    return 1;
                }
            }
        }
    }
}

```



```

    }
    }
    if(m[2][2] > vida) // esquinita de abajo der
    {
        if((m[2][1] > vida && m[1][2] > vida && m[1][1] > vida))
        {
            return 1;
        }
    }
    return 0;
} //still life
public int GL(int[][] m)
{
    int tam = 2;
    if(m[0][0] >= tam) // esquinita de arriba izr
    {
        if(m[1][0] >= m[0][0] && m[0][1] >= m[0][0])
        {
            if(m[2][1] >= m[1][0] && m[1][1] >= m[1][0])
            {
                return 1;
            }
        }
    }
    if(m[0][2] >= tam) // esquinita de arriba der
    {
        if(m[0][1] >= m[0][2] && m[1][2] >= m[0][2])
        {
            if(m[1][0] >= m[0][1] && m[2][1] >= m[1][0])
            {
                return 1;
            }
        }
    }
    if(m[2][0] >= tam) // esquinita de abajo izq
    {
        if(m[1][0] >= m[2][0] && m[2][1] >= m[1][0])
        {
            if(m[0][1] >= m[1][0] && m[1][2] >= m[0][1])
            {
                return 1;
            }
        }
    }
    if(m[2][2] >= tam) // esquinita de abajo der
    {

```

```
if(m[2][1] >= m[2][2] && m[1][2] >= m[2][2] || (m[1][1] >= m[2][2] && (m[2][1] >= m[2][2] ||
{
    if(m[1][0] >= m[2][2])
    {
        ret = 1;
    }
})
}

return 0;

}

//glider
}
```