UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS

CURSO	**:**	**BIOLOGIA COMPUTACIONAL**
TEMA	:	SIMULACION TUMORAL
PROFESOR	:	DAVID A. LAVAN QUIROZ
ALUMNOS	:	TOMAS J. CASAS RODRIGUEZ

LIMA – PERÚ

2016

INDICE

SIMULACION DE COMO SE GENERA UN TUMOR

I-INTRODUCCION

Este trabajo presenta un modelo sobre las células , desde el punto de vista matemático.
A causa de la mitosis las celulas se van reproduciendo .
Las simulaciones matemáticas son muy útiles para entender la transición del comportamiento de las celulas , de simple a complejo y determinar las condiciones en que se producen tumores cancerigenos.

El modelo será analizado, considerando los procesos de combinación de aleatoriedad , reproduccion ,aislamiento y borrosidad. De estos acercamientos, pueden evidenciarse diferentes comportamientos.
Las simulaciones numéricas se realizaron utilizando el software Netbeans y un lenguaje de programacion orientada a objetos que es java.
Se ha dado la interpretación biológica respectiva a partir del modelo matemático determinístico para las células buenas y las celulas cancerigenas “malas” .

II-ESTADO DE ARTE

Un modelo matemático contiene información detallada sobre la evolución en el tiempo del proceso. Cuando se está interesado en la evolución de una célula, la dinámica con respecto al tiempo puede no ser necesaria, sin embargo, su modelación toma la mayor parte de los esfuerzos y, a menudo se necesita conocer un gran numero de parámetros biologicos .
Cuando el análisis se basa en la experimentación, la biología computacional proporciona una herramienta esencial para el estudio de estos mecanismos que, por su complejidad, no puede ser comprendido por simple intuición.
Es por eso que se muestra como los modelos y simulaciones se utilizan para abordar el origen, propiedades y funciones de algunos de los principales ritmos celulares. También se describe como la biología computacional puede ayudar, considerando diferentes aproximaciones en el proceso de simulacion [1].

La secuencia de los acontecimientos, por el cual una célula se divide y se duplica, es llamada ciclo de división celular. En la mayoría de las células, el ciclo de replicación-división de ADN se suma a la duplicación de otros componentes de la célula (ribosomas, las membranas, maquinaria metabólicas, etc.), por lo que el tiempo entre la división de la célula es idéntico al tiempo de duplicación de su masa .
La duplicación de la masa es un proceso lento, es por eso que aparecen lagunas temporales (G1 y G2) entre la fase S (síntesis de ADN) y la fase M (mitosis). La fase G1 es la fase donde las células aumentan de tamaño. En la fase de síntesis, la replicación del ADN se lleva a cabo.
En la fase G2, la masa de células sigue creciendo y la célula se prepara para entrar en la última fase, en la mitosis, donde los cromosomas se separan en dos núcleos hijos. Al final de la división celular, las células hijas comienzan un nuevo ciclo de división [1].

MODELO MATEMATICO

En la última década, los modelos matemáticos se han desarrollado para proveer una visión de los mecanismos dinámicos fundamentales del ciclo celular .
Las dinámicas del ciclo celular han sido modeladas como ciclos límites sistemas biestables por la regulación de la masa celular , y procesos transitorios .
Mientras cada una de estas aproximaciones ha dado lugar a modelos que normalmente replican la dinámica del ciclo celular en condiciones especificas[1] .
El mejor método para modelar la complejidad de un ciclo celular es analizar cada componente individual, antes de enlazarlos. El soporte biológico de esta aproximación proviene de la existencia de dos puntos de control fuertes, uno al principio de cada transición .
El ciclo de replicacion-division del ADN, en todas las células eucariotas es controlado por un conjunto común de proteínas que interactúan entre sí, por un conjunto de reglas comunes [1].

MITOSIS Y MEIOSIS

Al estudiarlos comparativamente veremos como el resultado final de ambos procesos es distinto por lo que los cromosomas tienen que comportarse de modo diferente. Mediante la visualización de simulaciones de las características de las distintas fases del proceso mitótico y meiótico iremos comprendiendo la base física de la herencia de los caracteres (los genes se transmiten con los cromosomas) y cómo se mantiene el número cromosómico durante el desarrollo de un individuo, la regeneración de los tejidos o en los procesos reproductivos [2].
Al observar el comportamiento cromosómico durante la meiosis podremos relacionar el apareamiento y la segregación de cromosomas homólogos que ocurren durante la misma, con la constancia del número cromosómico de una especie a lo largo de las generaciones, y la combinación de caracteres paternos y maternos que se da en cualquier individuo.

MITOSIS

La división celular consta de dos procesos fundamentales: la mitosis o división del núcleo y la citocinesis o división del citoplasma. Ambos procesos son independientes pero deben ocurrir de forma sincronizada. El resultado son dos células hijas con una dotación cromosómica idéntica entre sí y a la de la célula madre.
La mitosis es el mecanismo estable que tienen las células para distribuir de forma exacta la información genética entre las células hijas durante las divisiones celulares. Durante la mitosis los cromosomas se reparten equitativamente, incluyéndose una dotación cromosómica completa en cada célula hija. Para facilitar este reparto, los cromosomas se condensan haciendo patente su morfología.
Esto hace que podamos conocer en ese momento cuántos cromosomas tiene una especie, dónde se localiza el centrómero (o constricción primaria), el número de brazos cromosómicos que presentan o la existencia de constricciones secundarias y satélites cromosómicos[2].
Cuando una célula no está dividiéndose se dice que está en interfase, lo que corresponde al lapso de tiempo que transcurre entre dos mitosis sucesivas. Durante este periodo la cromatina está descondensada y hay una gran actividad metabólica porque es cuando la mayor parte de los genes se expresan, aunque en cada tipo celular lo harán solo los necesarios para que desarrolle su función específica.
Un suceso importante de la interfase es la replicación del ADN, que ocurre en el periodo denominado S, tras la cual los cromosomas ya tienen dos réplicas idénticas denominadas cromátidas

hermanas. Esta fase S va precedida por el periodo G1 y seguida del periodo G2 en los que hay crecimiento celular, actividad transcripcional y la célula se prepara para dividirse. A continuación se

iniciaría la mitosis.
Si después de una mitosis la célula no va a dividirse de nuevo, se queda en lo que llamamos fase G0. Si anotamos como M a la mitosis, el ciclo celular es una sucesión cíclica de los distintos periodos según esta secuencia (Figura 1):

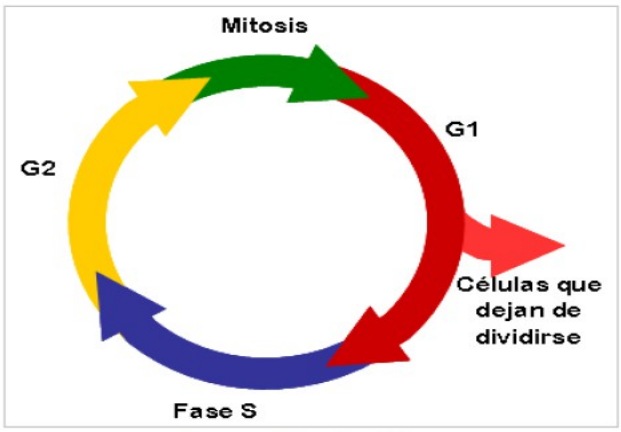


Figura 1: Ciclo celular

Una vez que se inicia, el proceso mitótico transcurre de forma continua sin que haya interrupciones, pero ocurren una serie de sucesos destacados que son clave para que el reparto de la información genética sea correcto y en los que nos basamos para distinguir de forma arbitraria cuatro etapas:
1. Profase
2. Metafase
3. Anafase
4. Telofase

La cronología y características de los sucesos clave mitóticos que describiremos a continuación son comunes a la mayoría de los organismos donde se ha estudiado. No obstante, se han descrito excepciones en algunas especies afectando al momento en que se inicia la condensación cromosómica (hay especies en las que los cromosomas no se condensan nunca), a la desaparición de la membrana nuclear o, por ejemplo, al establecimiento de la placa metafásica. La anafase parece ser la etapa más conservada en los diferentes organismos[2].

PROFASE

Durante este periodo la fibra de cromatina, que ha ido organizándose en plegamientos cada vez más complejos, aparece como cromosomas visibles que van condensándose gradualmente. Hay 2n cromosomas en la célula y cada cromosoma consta de dos cromátidas hermanas con igual información y morfología. Éstas aparecen unidas a nivel del centrómero y a lo largo de los brazos cromosómicos gracias a complejos proteicos. Al final de la profase se desorganizan los nucleolos y desparece la membrana nuclear cuyos componentes quedan dispersos en el núcleo.

METAFASE

Los cromosomas se encuentran ahora libres en el citoplasma y los centrómeros de cada cromosoma contactan con las fibras del huso, que se organizan en el centro organizador de microtúbulos (MTOC), formado por los centriolos, que actúan como centro de atracción de los cromosomas hacia los polos, y la masa amorfa pericentriolar. La intervención de las fibras del huso y de otras proteínas de movimiento cromosómico permite a los cromosomas organizarse en la llamada placa

metafásica. Cada cromátida hermana se orienta hacia un polo distinto lo que garantiza el reparto de la información genética de cada cromosoma a las dos células hijas.

Ahora los cromosomas alcanzan su máximo grado de condensación y su morfología se hace patente (Figura 2). Por eso en esta fase es donde mejor se pueden estudiar todas las características morfológicas de los cromosomas, lo que será de utilidad para, por ejemplo, identificar homólogos y confeccionar un cariotipo o realizar estudios comparativos entre especies y conocer la evolución cariotípica ocurrida dentro de un taxón.

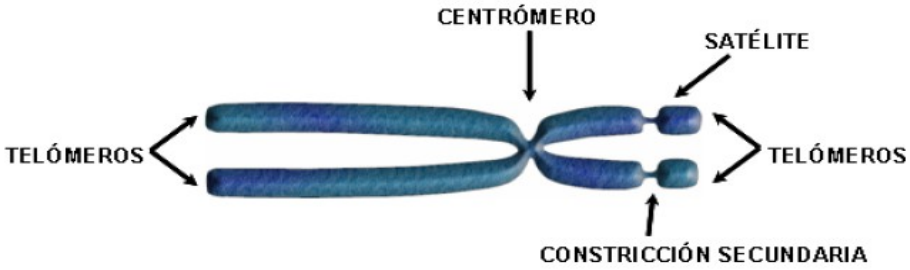


Figura 2: Morfología de los cromosomas

El centrómero es la constricción primaria que aparece en todos los cromosomas y donde se asocian los cinetocoros o estructura proteica a la que se unen las fibras dehuso mitótico. Su posición define el número de brazos de un cromosoma. Si está situado en un extremo del cromosoma, éste tendrá un solo brazo y si está en otra posición veremos cromosomas con dos brazos. El cinetocoro funciona a modo de un "interfaz" entre el centrómero y las fibras del huso.
En algunos cromosomas aparecen constricciones secundarias, normalmente asociadas a la región organizadora nucleolar (NOR), donde se encuentran los genes para ARN ribosómico. El fragmento de cromosoma que va desde la constricción secundaria al telómero se denomina satélite cromosómico. El telómero constituye el extremo cromosómico, por lo que hay uno en cada brazo cromosómico y juega un papel fundamental en el mantenimiento de la integridad del cromosoma.

ANAFASE

La anafase es, en general, la etapa más corta de la mitosis Cada centrómero se divide en dos y se desorganizan las proteínas que mantenían unidas a las cromátidas hermanas, lo que les permite segregar (migrar) a polos opuestos. En cada polo celular veremos un grupo de cromosomas (*2n*) con una sola cromátida orientados hacia el polo correspondiente.

TELOFASE

Los cromosomas agrupados en cada polo comienzan a descondensarse y los nucleolos y la membrana nuclear vuelven a organizarse a partir de material preexistente y de nueva síntesis.
La división celular se completa al final de esta etapa con la citocinesis, donde hay también un reparto de los orgánulos y componentes citoplásmicos a las dos células hijas, aunque no se realiza de forma tan precisa como durante la mitosis. El resultado final del proceso mitótico son dos células con *2n* cromosomas. Con esta práctica vamos a poder observar con técnicas citogenéticas evidencias de los siguientes fenómenos genéticos:
- Replicación del ADN y la cromatina: cromosomas con dos cromátidas.
- Conservación del material hereditario y constancia del número cromosómico: segregación de las cromátidas hermanas de cada cromosoma durante la anafase.

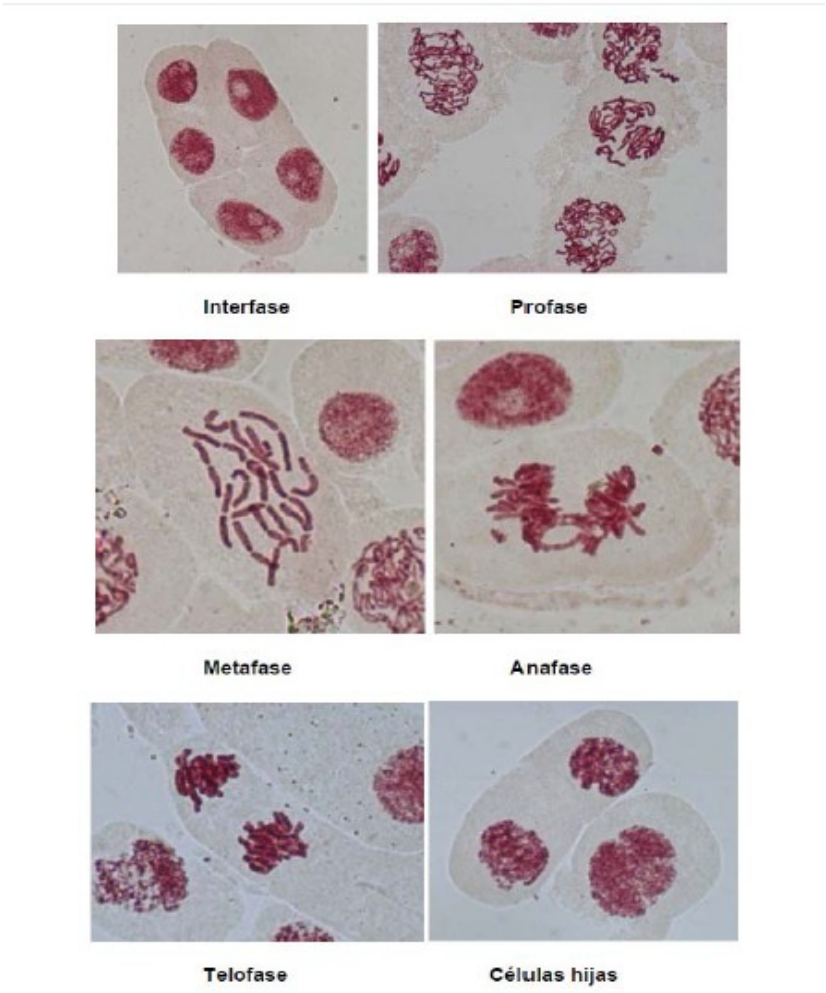


Figura 3: Fases de la mitosis

MEIOSIS

La meiosis es un proceso que consta de dos divisiones celulares consecutivas sin que haya replicación del ADN entre ellas. Su función es diferente a la de la mitosis ya que al final del proceso lo que se consigue es reducir el número cromosómico a la mitad, obteniendo gametos haploides en los organismos con reproducción sexual.

La forma para conseguir esa reducción es que los cromosomas con el mismo tipo de información genética u homólogos se apareen primero para después segregar a distintos polos celulares. Durante el apareamiento cromosómico y la sinapsis ambos homólogos pueden recombinar, es decir,

intercambiar segmentos cromosómicos , generando nuevas combinaciones alélicas en los cromosomas resultantes.

Las diferentes combinaciones posibles de cromosomas paternos y maternos que puedan quedar incluidas en un gameto, como consecuencia de la segregación de homólogos, es otra fuente adicional de variabilidad genética.

En la meiosis también distinguimos cuatro etapas en cada una de las dos divisiones:
profase, metafase, anafase y telofase.

III-METODOS

La modelización matemática de tumores puede ser vista como el primer paso en la construcción de modelos para el crecimiento tumoral en las etapas posteriores y puede jugar un papel muy importante en la investigación del cáncer.
En particular, la modelización del crecimiento del tumor puede proporcionar una valiosa información sobre mecanismos de crecimiento de células tumorales y proliferación.
Se ha utilizado un enfoque para desarrollar modelos matemáticos de como se genera los tumores de la siguiente manera :
-Se tiene celulas buenas y malas que tienen las siguientes caracteristicas la celula mala se mueve mas rapido que la celula buena (Vb<Vm).
-La celula tiene como parametros la velocidad , area y longitud (l,A,V).
-A medida que la celula se mueve y encuentra nutrientes va alimentandose aumentando su area, (Af=A0+n*delta(A)) cuando la celula llega a obtener el doble de su area hace el proceso de mitosis dividiendose en 2 celulas con area original A0 .
-Cuando se tiene que la celula ya sea buena o mala su Af=2*A0 , hay mitosis .
-Las celulas se mueven en una cuadrilla con contorno regular y swe pueden mover de cuadro en cuadro.
-Al final las celulas buenas quedan encerradas por celulas malas ,con el paso del tiempo las celulas buenas al no tener nutrientes para no comer se van pudriendo y estas son las que generan el tumor .

A pesar de ser un formalismo extremadamente valioso, la división de las células en compartimentos separados no deja de ser artificial.
Detalladas investigaciones experimentales basadas en la medición de las tasas de consumo de oxígeno muestran que las transiciones entre las capas pueden ser graduales en lugar de bruscas.
Ward y King [4] desarrollaron el primer modelo de estructura espacial de tumores que no asumía la división de las células en capas separadas.
Su modelo estaba formulado en términos de densidades de poblaciones de células vivas y muertas, con las células quiescentes omitidas por simplicidad.
Así, mostraron que un tumor crece de manera exponencial al principio y después se estabiliza en un crecimiento lineal.
Sherratt y Chaplain [5] formularon un modelo en términos de densidades de las células proliferantes, quiescentes y necróticas en un dominio unidimensional, en el que incorporaron el movimiento celular.
Más recientemente, **Tan y Ang** [6] modificaron el modelo para incluir la variación aleatoria en los procesos celulares y extracelulares. Este modelo proporcionaba una descripción más realista del crecimiento del tumor y es razonablemente un buen intento de modelar los procesos biológicos complejos de crecimiento tumoral usando términos aleatorios en las ecuaciones del modelo.

IV-RESULTADOS

-A partir del modelo matemático determinístico genérico para las células , se ha construido el sistema correspondiente.
-Esto es caracteristico en un proceso Wiener con parámetros aleatorios. De los gráficos se observa que no se tiene una simetría alternante de ciclos cortos y largos, así como salida de la célula de la mitosis e inicia otro ciclo de división celular .
-Se ha ilustrado el caso de un ciclo celular normal .
-El resultado se muestra como una alternancia de ciclos cortos y largos, pero no es constante en todas las divisiones .
-Los algoritmos implementados en java para el modelo usan hilos que hacen el movimiento de cada

celula independiente y la aleatorización se realiza mediante el uso de tablas aleatoria indirectamente al usar la funcion random de java .
-Se muestran los resultados de la simulacion en el transcurso del tiempo , lo de color verde son los nutrientes con lo que se alimentan las celulas buenas (color rojo) y celulas malas (color negro) .

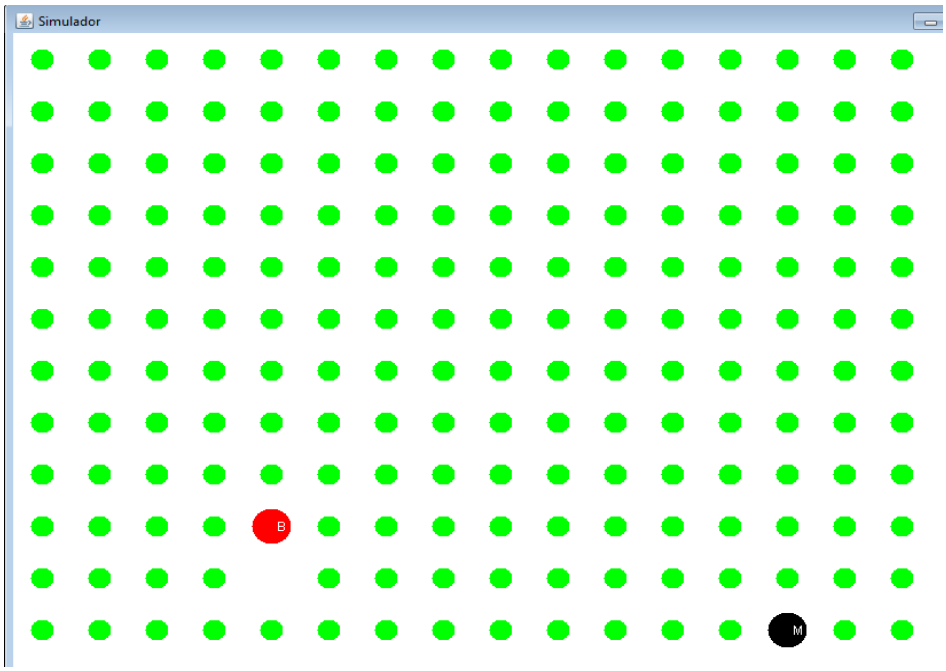


Fig. 4 : Inicio de la simulacion con celula buena y mala la velocidad de movimiento de la celula mala es mayor que la celula buena (Vb<Vm) .

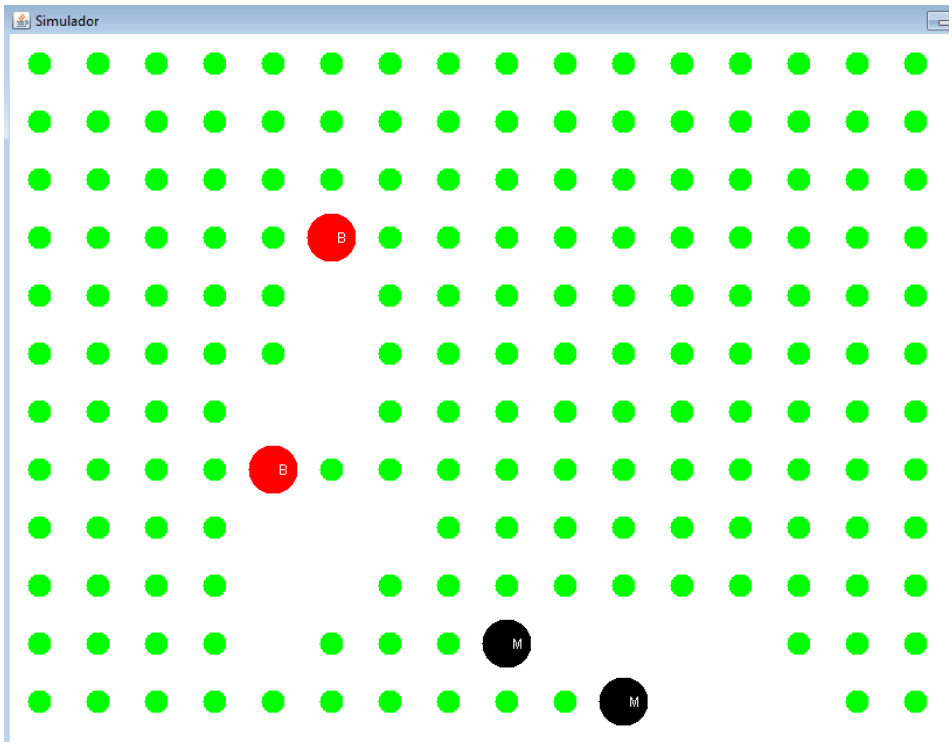


Fig. 5 : Las celulas se van alimentando de los nutrientes y aumentando su area hasta hacer el proceso de division similar llamado mitosis donde se reproducen.

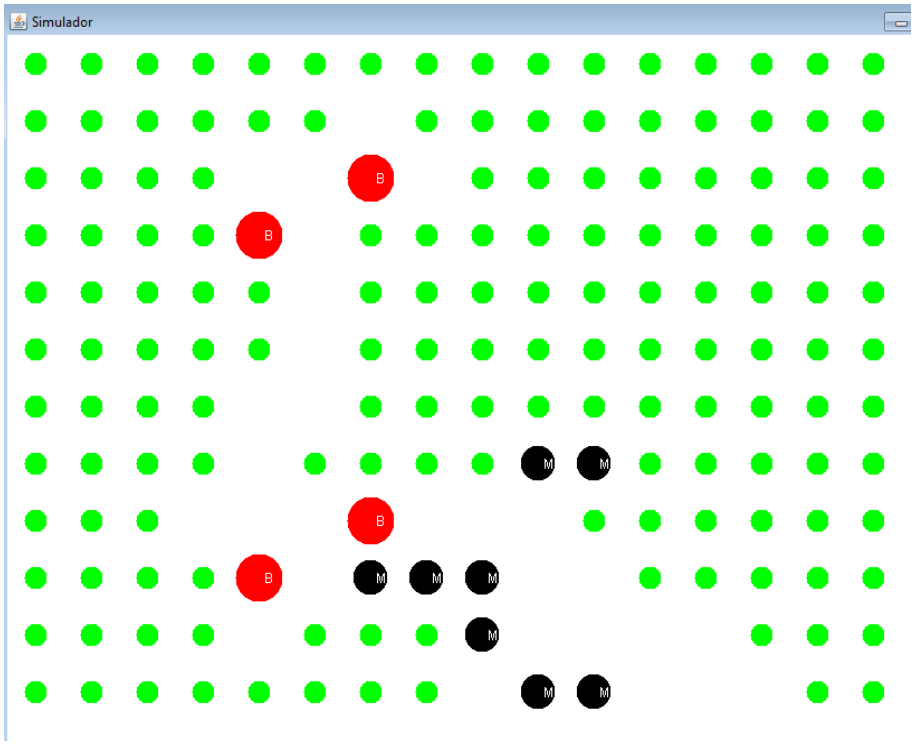


Fig. 6 : Como se observa el incremento de las celulas malas es mucho mayor que de las celulas buenas.

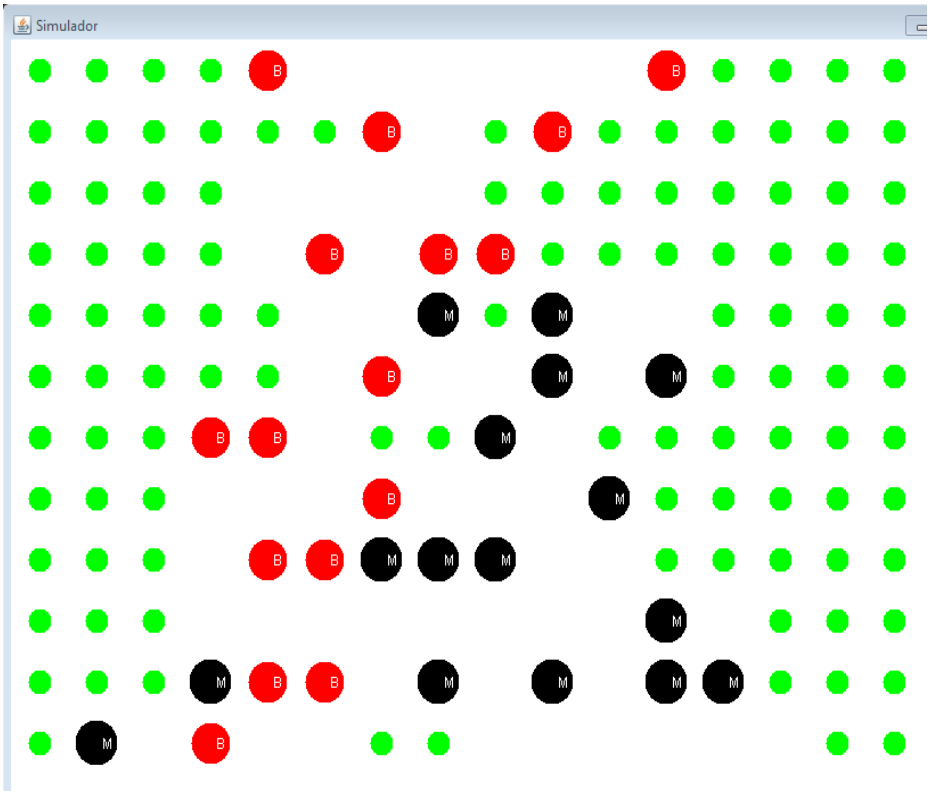


Fig. 7 : La celulas se alimentan y hacen mitosis una mas rapidas que otras.

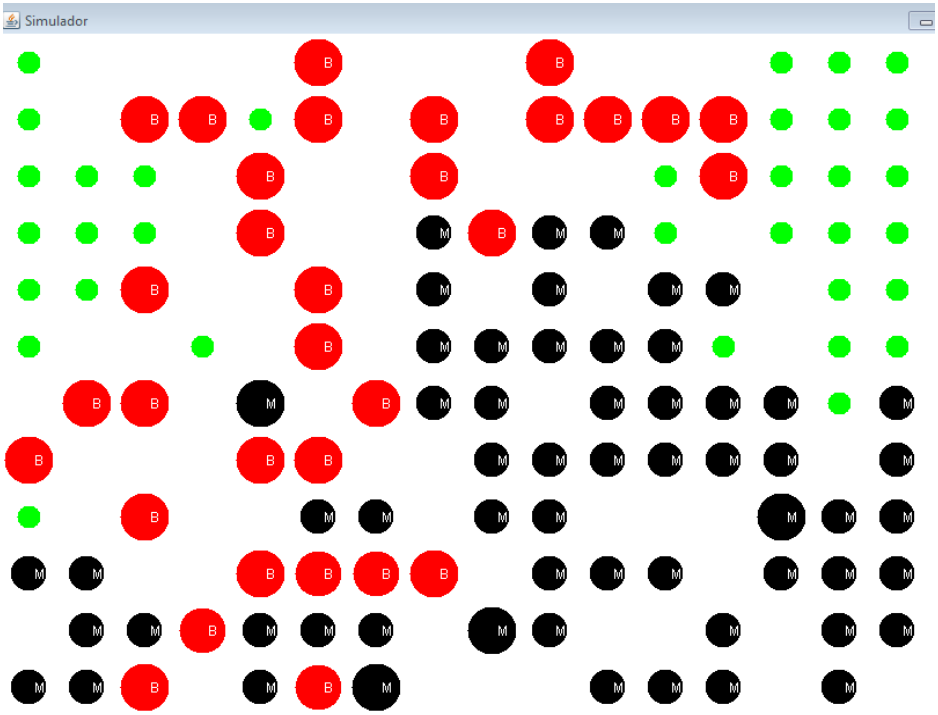


Fig . 8 : Se observa que el numero de celulas malas es mayor que el de celulas buenas.

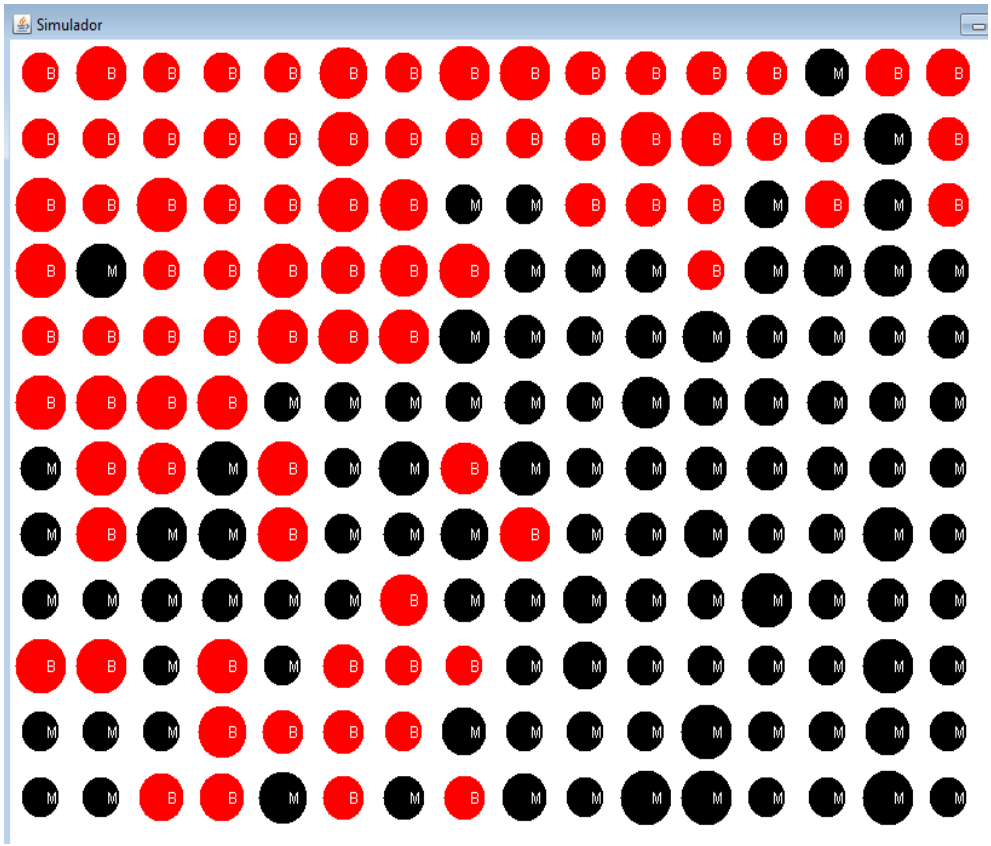


Fig . 9 : Resultado final de la simulacion , los tumores son los encerrados .

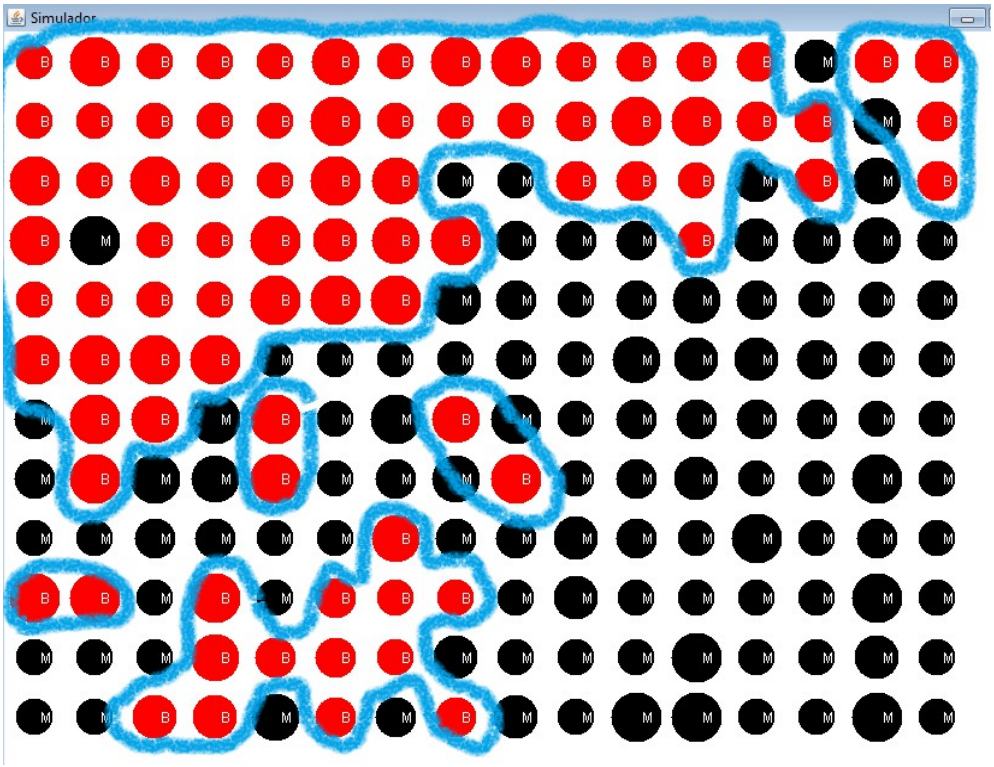


Fig . 10 : Los tumores son encerrados en lineas celestes como se muestra .

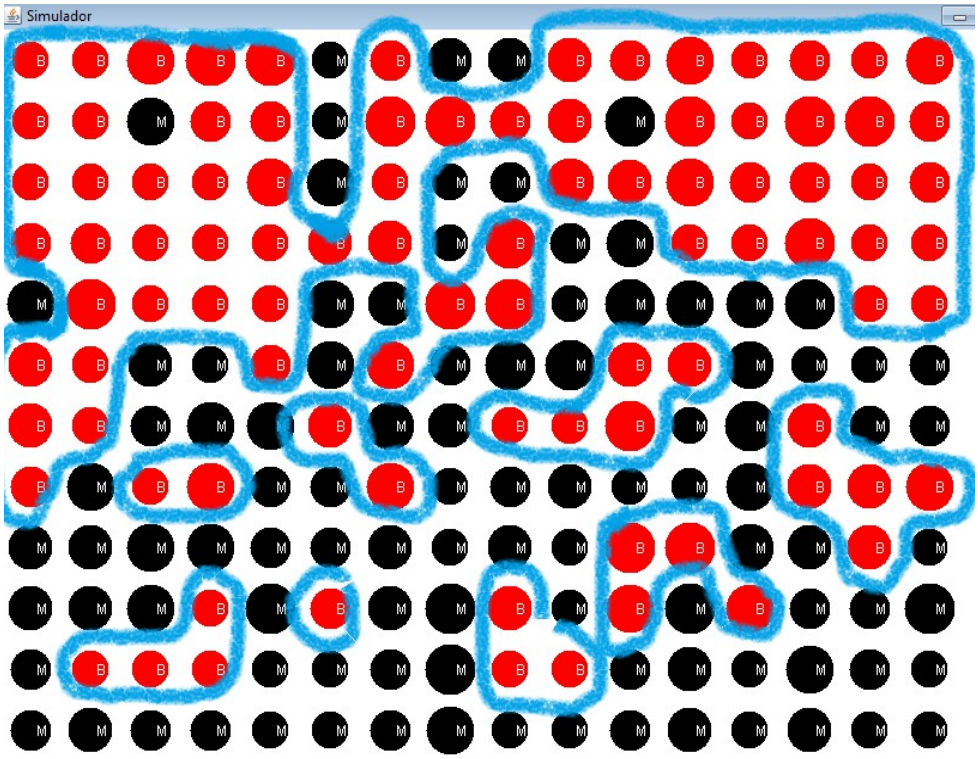


Fig .11 : Este es el resultado de otra simulacion donde se aprecia como se genera los tumores.

V-DISCUSION

-El modelaje computacional se muestra como una herramienta esencial en el tratamiento de las dinámicas de crecimiento, siendo ésta la forma más eficiente de estudiar y visualizar el comportamiento de poblaciones que obedecen estructuras discretas y fenómenos de aleatoriedad.

-Con el modelo estudiado de competencia por nutrientes se analizaron diversas morfologías de crecimiento, estudiando y evaluando sistemáticamente el comportamiento y la dinámica tumoral al variar los parámetros que regulan el modelo, como absorción y consumo de nutrientes, migración y mitosis celular.

-En este proceso se encontraron comportamientos particulares asociados con la disponibilidad de nutrientes y de espacio, resaltándose que las células cancerosas compiten por estos dos.

-De los resultados de la simulacion se obtiene que los tumores son generados por celulas que se putrefactan con el tiempo ya que al ser encerradas por celulas malas ya no pueden nutrirse y migrar en busca de nutrientes ; con el tiempo estas zonas generan los llamados tumores .

VI-REFERENCIAS BIBLIOGRAFIA

[1] Dialnet-ModeloMatematicoGenericoEstocasticoYCredibilistico-3741025.pdf

[2] *Departamento de Genética, Universidad de Granada ,Manual de practicas.*

[3]http://pendientedemigracion.ucm.es/info/genetica/grupod/index.htm

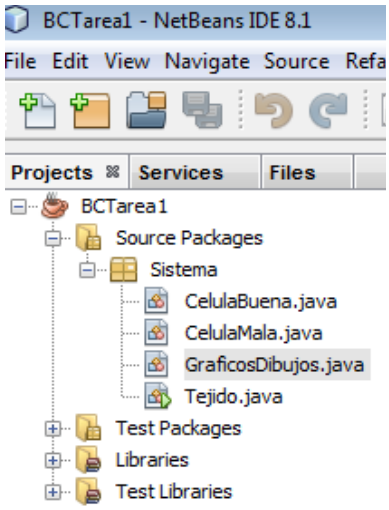
[4] Ward, J. P. y King, J. R., Mathematical modelling of avascular tumour growth, IMA Journal of Mathematics Applied in Medicine & Biology, 1997, 14, 39-69.

[5] Sherratt, J. A. y Chaplain, M. A. J., A new mathematical model for avascular tumour growth, Journal of Mathematical Biology, 2001, 43, 291-312.

[6] Tan, L. S. y Ang, K. C., A numerical simulation of avascular tumour growth, ANZIAM Journal, 2005, 46(E), C902-C917.

ANEXO

Este es el codigo usado para la implementacion para la simulacion , la clase principal es Tejido , las otras son CelulaBuena , CelulaMala y GraficosDibujos .Para su elaboración se uso el IDE Netbeans 8.1 y compilación .



Tejido.java

```
package Sistema;
import static Sistema.GraficosDibujos.espacioCuadrilla;
import java.awt.Graphics;
import java.awt.Graphics2D;
import javax.swing.JFrame;
import javax.swing.JPanel;

@SuppressWarnings("serial")
public class Tejido extends JPanel  {
    GraficosDibujos lapiz=new GraficosDibujos();
    static Graphics g;
    Graphics2D g2d = (Graphics2D) g;
   static CelulaBuena[] cb=new CelulaBuena[1000];
   static CelulaMala[] cm=new CelulaMala[1000];
   //tablero tejido con la comida
   static  double tejido[][]=new double[12][16];
   //Posicion inicial buena
   static int xab,yab;
   static int xam,yam;
    int longitudx=800,longitudy=600;
    int contador=0;

@Override
public void paint(Graphics g) {
   //CELULAS MALAS
   //lapiz.dibujarCelulasMala(g,cm);
   mitosisCelulaMala(cm);
   lapiz.dibujarTejido(tejido, contador,g);
   lapiz.dibujarComida(g,tejido);
   lapiz.dibujarCelulasMala(g,cm);
   lapiz.dibujarCelulasBuena(g,cb);
```

```
//CELULAS BUENAS
mitosisCelulaMala(cm);
mitosisCelulaBuena(cb);
 repaint();
 contador++;
}

        public static void main(String[] args){
                JFrame frame = new JFrame("Simulador");
         frame.add(new Tejido());
                frame.setSize(900, 700);
                frame.setVisible(true);
         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         //Posicion inicial buena : aleatorio xa 0,1,2,3...16
         xab=(int)(Math.random()*17);
         yab=(int)(Math.random()*13);
         //celula buena=1
          cb[0] = new CelulaBuena(0,xab*espacioCuadrilla,yab*espacioCuadrilla);
          cb[0].start();
          //Posicion inicial buena : aleatorio xa 0,1,2,3...16
         xam=(int)(Math.random()*17);
         while(true){
         if(xam==xab) xam=(int)(Math.random()*17);
         else break;
         }
         yam=(int)(Math.random()*13);
         while(true){
         if(yam==yab)xam=(int)(Math.random()*13);
         else break;
         }
        //celula mala=0
         cm[0] = new CelulaMala(0,xam*espacioCuadrilla,yam*espacioCuadrilla);
         cm[0].start();
        }

    public  void mitosisCelulaBuena(CelulaBuena[] celulas){
            int numc=0;// System.out.println("Padres");
            for(int i=0;celulas[i]!=null;i++){// System.out.print(celulas[i].getIdCelulaBuena()+"\t");
               numc++;
            }
            int cont=0;
            for(int i=0;i<numc;i++){
                  if(celulas[i].getAreaCelula()==10){  //  System.out.println("Mitosis");
                        celulas[i].setAreaCelula(5);
                        int mitosisPos[]=new int[2];
                        mitosisPosicion(mitosisPos,celulas[i].getX(),celulas[i].getY());
                        celulas[numc+cont] = new CelulaBuena(numc+cont,mitosisPos[0],mitosisPos[1]);
                        celulas[numc+cont] .start();
                        cont++;
                  }
            }
    }

      public  void mitosisCelulaMala(CelulaMala[] celulas){
            int numc=0;// System.out.println("Padres");
            for(int i=0;celulas[i]!=null;i++){// System.out.print(celulas[i].getIdCelulaBuena()+"\t");
               numc++;
            }
            int cont=0;
            for(int i=0;i<numc;i++){
```

```
                    if(celulas[i].getAreaCelula()==10){  //  System.out.println("Mitosis");
                        celulas[i].setAreaCelula(5);
                        int mitosisPos[]=new int[2];
                        mitosisPosicion(mitosisPos,celulas[i].getX(),celulas[i].getY());
                        celulas[numc+cont] = new CelulaMala(numc+cont,mitosisPos[0],mitosisPos[1]);
                        celulas[numc+cont] .start();
                        cont++;
                    }
                }
        }

        public static void mitosisPosicion(int mitosisPos[],int px,int py){
         boolean  noreproduce=true;
         boolean estoyChocandoCelulas=true;
         int posibilidades=0;
         while(noreproduce && estoyChocandoCelulas){
              //aleatorio 0,1,2,3
        int n=(int)(Math.random()*4);
        //izquierda
        if(n==0 && chocaContorno(px-espacioCuadrilla,py) && !chocaCelulas(px-espacioCuadrilla,py)){
           px=px-espacioCuadrilla;
           noreproduce=false;
           estoyChocandoCelulas=false;
        }
        //derecha
        else if(n==1 && chocaContorno(px+espacioCuadrilla,py) && !chocaCelulas(px+espacioCuadrilla,py) ){
             px=px+espacioCuadrilla;
             noreproduce=false;
             estoyChocandoCelulas=false;
        }
        //arriba
        else if(n==2 && chocaContorno(px,py-espacioCuadrilla) && !chocaCelulas(px,py-espacioCuadrilla) ){
             py=py-espacioCuadrilla;
             noreproduce=false;
             estoyChocandoCelulas=false;
        }
        //abajo
        else if(n==3 && chocaContorno(px,py+espacioCuadrilla) && !chocaCelulas(px,py+espacioCuadrilla)){
             py=py+espacioCuadrilla;
             noreproduce=false;
             estoyChocandoCelulas=false;
        }
        else{
           //System.out.println("No me reproduci, choco contorno ocelulas");
           if(posibilidades==1000)break;
           posibilidades++;
        }
        }
        mitosisPos[0]=px;
        mitosisPos[1]=py;
      }

    public static boolean chocaCelulas(int px,int py){
     int numcb=0;
     int numcm=0;
     for(int i=0;cb[i]!=null;i++){
    numcb++;
    }
     for(int i=0;cm[i]!=null;i++){
    numcm++;
```

```
}
 for(int i=0; i<numcb ;i++){
if(cb[i].getX()==px && cb[i].getY()==py) return true;
}
 for(int i=0; i<numcm ;i++){
if(cm[i].getX()==px && cm[i].getY()==py) return true;
}

return false;
}

public static boolean  chocaContorno(int px,int py){
if( px<0 || px>750) return false;
else if(py<0 || py>550) return false;
else return true;
}
 }
```

CelulaBuena.java

```
package Sistema;
import static Sistema.Tejido.tejido;
import static Sistema.Tejido.cb;
import static Sistema.Tejido.cm;
import static Sistema.GraficosDibujos.porcionComida;
import static Sistema.GraficosDibujos.espacioCuadrilla;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.util.logging.Level;
import java.util.logging.Logger;
public class CelulaBuena extends Thread{
//variables
private int idCelula;
private int areaCelula;
private int x;
private int y;
/*CONSTRUCTOR*/
public CelulaBuena(int idCelula,int x,int y)
{
this.idCelula=idCelula;
this.x=x;
this.y=y;
this.areaCelula=5;
}

 @Override
   public void run()
   {
      try{
         Thread.sleep( (int)(10) );
      }
      catch(InterruptedException e){};
      while (true) {
         try {
            Thread.sleep( (int)(2000) );
         } catch (InterruptedException ex) {
            Logger.getLogger(CelulaBuena.class.getName()).log(Level.SEVERE, null, ex);
         }
        moverCelula();
      }
```

```
    }
//Pinta celula
 public void pintarCelula(Graphics g){
Graphics2D g2d = ( Graphics2D ) g;
//BUENA
g.setColor(Color.RED);
int diametro=(int)Math.round(2*7*Math.sqrt((float)getAreaCelula()));
int desplazamiento=(int)(espacioCuadrilla-diametro)/2;
/*System.out.println("diametro : "+diametro);
System.out.println("radio : "+7*Math.sqrt((float)getAreaCelula()));
System.out.println("desplazamiento : "+desplazamiento);
System.out.println("**************************************************************************");*/
g.fillOval(getX()+desplazamiento, getY()+desplazamiento,diametro,diametro);
g.setColor(Color.WHITE);
g.drawString("B", getX()+30, getY()+30);
}

public void moverCelula()
{
  try{
      //TIEMPO PARA ACELERAR O RETARDAR LA ACELERACION
      Thread.sleep(220);}
catch(InterruptedException e){}
  boolean estoyFuera=true;
  boolean estoyChocandoCelulas=true;
  int posibilidades=0;
  while(estoyFuera && estoyChocandoCelulas){
 //aleatorio 0,1,2,3
int n=(int)(Math.random()*4);
//izquierda
if(n==0 && chocaContorno(getX()-espacioCuadrilla,getY()) && !chocaCelulas(getX()-espacioCuadrilla,getY())){
    chocaComida(getX()-espacioCuadrilla,getY());
    setX(getX()-espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
//derecha
else if(n==1 && chocaContorno(getX()+espacioCuadrilla,getY()) && !chocaCelulas(getX()+espacioCuadrilla,getY()))
{
    chocaComida(getX()+espacioCuadrilla,getY());
    setX(getX()+espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
//arriba
else if(n==2 && chocaContorno(getX(),getY()-espacioCuadrilla) && !chocaCelulas(getX(),getY()-espacioCuadrilla)){
    chocaComida(getX(),getY()-espacioCuadrilla);
    setY(getY()-espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
//abajo
else if(n==3 && chocaContorno(getX(),getY()+espacioCuadrilla) && !chocaCelulas(getX(),getY()
+espacioCuadrilla) ){
    chocaComida(getX(),getY()+espacioCuadrilla);
    setY(getY()+espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
else{
```

```
    //System.out.println(idCelula+"===No me muevo me choco contorno o con celulas");
    if(posibilidades==1000)break;
          posibilidades++;
}
}
}

public boolean chocaCelulas(int px,int py){
 int numcb=0;
 int numcm=0;
 for(int i=0;cb[i]!=null;i++){
numcb++;
}
 for(int i=0;cm[i]!=null;i++){
numcm++;
}
 for(int i=0; i<numcb ;i++){
if(cb[i].getX()==px && cb[i].getY()==py) return true;
}
 for(int i=0; i<numcm ;i++){
if(cm[i].getX()==px && cm[i].getY()==py) return true;
}

return false;
}

public void chocaComida(int px,int py){
    for(int y=0;y<tejido.length;y++){
        for(int x=0;x<tejido[y].length;x++){
            if(x==(px/espacioCuadrilla) && y==(py/espacioCuadrilla) ){
                        tejido[y][x]=0.0;
                        setAreaCelula(getAreaCelula()+porcionComida);
                        break;
                       }
        }
    }
}

public boolean  chocaContorno(int px,int py){
if( px<0 || px>750) return false;
else if(py<0 || py>550) return false;
else return true;
}

//ENCAPSULAMIENTO DE VARIABLES
    public int getIdCelula() {
        return idCelula;
    }

    public void setIdCelulaBuena(int idCelula) {
        this.idCelula = idCelula;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }
```

```
    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    public int getAreaCelula() {
        return areaCelula;
    }

    public void setAreaCelula(int areaCelula) {
        this.areaCelula = areaCelula;
    }
}
```

CelulaMala.java

```
package Sistema;
import static Sistema.Tejido.tejido;
import static Sistema.Tejido.cb;
import static Sistema.Tejido.cm;
import static Sistema.GraficosDibujos.porcionComida;
import static Sistema.GraficosDibujos.espacioCuadrilla;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.util.logging.Level;
import java.util.logging.Logger;
public class CelulaMala extends Thread{
//variables
private int idCelula;
private int areaCelula;
private int x;
private int y;
/*CONSTRUCTOR*/
public CelulaMala(int idCelula,int x,int y)
{
this.idCelula=idCelula;
this.x=x;
this.y=y;
this.areaCelula=5;
}

 @Override
    public void run()
    {
        try{
            Thread.sleep( (int)(10) );
        }
        catch(InterruptedException e){};
        while (true) {
            try {
                Thread.sleep( (int)(2000) );
            } catch (InterruptedException ex) {
                Logger.getLogger(CelulaMala.class.getName()).log(Level.SEVERE, null, ex);
            }
           moverCelula();
        }
    }
```

```
//Pinta celula
 public void pintarCelula(Graphics g){
Graphics2D g2d = ( Graphics2D ) g;
//BUENA
g.setColor(Color.BLACK);
int diametro=(int)Math.round(2*7*Math.sqrt((float)getAreaCelula()));
int desplazamiento=(int)(espacioCuadrilla-diametro)/2;
g.fillOval(getX()+desplazamiento, getY()+desplazamiento,diametro,diametro);
g.setColor(Color.WHITE);
g.drawString("M", getX()+30, getY()+30);
}

public void moverCelula()
{
  try{
      //TIEMPO PARA ACELERAR O RETARDAR LA ACELERACION
      Thread.sleep(150);}
catch(InterruptedException e){}
  boolean estoyFuera=true;
  boolean estoyChocandoCelulas=true;
  int posibilidades=0;
  while(estoyFuera && estoyChocandoCelulas){
 //aleatorio 0,1,2,3
int n=(int)(Math.random()*4);
//izquierda
if(n==0 && chocaContorno(getX()-espacioCuadrilla,getY()) && !chocaCelulas(getX()-espacioCuadrilla,getY())){
    chocaComida(getX()-espacioCuadrilla,getY());
    setX(getX()-espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
//derecha
else if(n==1 && chocaContorno(getX()+espacioCuadrilla,getY()) && !chocaCelulas(getX()+espacioCuadrilla,getY()))
{
    chocaComida(getX()+espacioCuadrilla,getY());
    setX(getX()+espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
//arriba
else if(n==2 && chocaContorno(getX(),getY()-espacioCuadrilla) && !chocaCelulas(getX(),getY()-espacioCuadrilla)){
    chocaComida(getX(),getY()-espacioCuadrilla);
    setY(getY()-espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
//abajo
else if(n==3 && chocaContorno(getX(),getY()+espacioCuadrilla) && !chocaCelulas(getX(),getY()
+espacioCuadrilla) ){
    chocaComida(getX(),getY()+espacioCuadrilla);
    setY(getY()+espacioCuadrilla);
    estoyFuera=false;
    estoyChocandoCelulas=false;
}
else{
    //System.out.println(idCelula+"===No me muevo me choco contorno o con celulas");
    if(posibilidades==1000)break;
          posibilidades++;
}
}
```

```
}

public boolean chocaCelulas(int px,int py){
 int numcb=0;
 int numcm=0;
 for(int i=0;cb[i]!=null;i++){
numcb++;
}
 for(int i=0;cm[i]!=null;i++){
numcm++;
}
 for(int i=0; i<numcb ;i++){
if(cb[i].getX()==px && cb[i].getY()==py) return true;
}
 for(int i=0; i<numcm ;i++){
if(cm[i].getX()==px && cm[i].getY()==py) return true;
}

return false;
}

public void chocaComida(int px,int py){
   for(int y=0;y<tejido.length;y++){
      for(int x=0;x<tejido[y].length;x++){
         if(x==(px/espacioCuadrilla) && y==(py/espacioCuadrilla) ){
                        tejido[y][x]=0.0;
                        setAreaCelula(getAreaCelula()+porcionComida);
                        break;
                       }
      }
   }
}

public boolean  chocaContorno(int px,int py){
if( px<0 || px>750) return false;
else if(py<0 || py>550) return false;
else return true;
}

//ENCAPSULAMIENTO DE VARIABLES
   public int getIdCelula() {
      return idCelula;
   }

   public void setIdCelulaBuena(int idCelula) {
      this.idCelula = idCelula;
   }

   public int getX() {
      return x;
   }

   public void setX(int x) {
      this.x = x;
   }

   public int getY() {
      return y;
   }
```

```
    public void setY(int y) {
        this.y = y;
    }

    public int getAreaCelula() {
        return areaCelula;
    }

    public void setAreaCelula(int areaCelula) {
        this.areaCelula = areaCelula;
    }
}
```

GraficosDibujos.java

```
package Sistema;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import static Sistema.Tejido.xab;
import static Sistema.Tejido.yab;
import static Sistema.Tejido.xam;
import static Sistema.Tejido.yam;

public class GraficosDibujos {
   static int  espacioCuadrilla=50;
   static int  porcionComida=1;
 //CONSTRUCTOR
public GraficosDibujos()
{
}
  //DIBUJA TEJIDO POR PRIMERA VEZ
 public void dibujarTejido(double tejido[][] ,int contador,Graphics g){
      Graphics2D g2d = (Graphics2D) g;
              g2d.setColor(Color.WHITE);
                      g2d.fillRect(0,0,1200,900);
     if(contador==0){
          for(int y=0;y<tejido.length;y++){
              for(int x=0;x<tejido[y].length;x++){
                 if( y==yam && x==xam) tejido[y][x]=0;
                 if( y==yab && x==xab) tejido[y][x]=0;
                else tejido[y][x]=porcionComida;
              }
           }
     }
}
   //DIBUJA COMIDA EN EL TABLERO
    public void dibujarComida(Graphics g,double tejido[][]){
    Graphics2D g2d = ( Graphics2D ) g;
    g.setColor(Color.GREEN);
    for(int y=0;y<tejido.length;y++){
        for(int x=0;x<tejido[y].length;x++){
        if( tejido[y][x]==porcionComida)
           g.fillOval(x*espacioCuadrilla+15,y*espacioCuadrilla+15,espacioCuadrilla-30,espacioCuadrilla-30);
        }
    }
    }
    //Dibuja celula BUena
     public  void dibujarCelulasBuena(Graphics g,CelulaBuena[] celulas){
        int numcb=0;
        for(int i=0;celulas[i]!=null;i++){
         numcb++;
```

```
        }
        for(int i=0;i<numcb;i++){
        celulas[i].pintarCelula(g);
        }
        }

      //Dibuja celula mala
     public  void dibujarCelulasMala(Graphics g,CelulaMala[] celulas){
        int numcb=0;
        for(int i=0;celulas[i]!=null;i++){
         numcb++;
        }
        for(int i=0;i<numcb;i++){
        celulas[i].pintarCelula(g);
        }
        }
}
```

EJECUCION