

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Modelado de la dinámica de células tumorales en un tejido,
usando un autómata celular de retícula de gases

Jesús Mata Castro
Matrícula 204203432

Trimestre:
2013 Otoño

Asesor:
Germán Téllez Castillo
Profesor Asociado
Departamento de Sistemas

CAPITULO 1	4
Introducción	4
Justificación	4
Objetivo General.....	5
Objetivos específicos	5
CAPITULO 2	7
Estado del arte.....	7
CAPITULO 3	9
Antecedentes.....	9
La Estructura de un Autómata Celular.....	9
Autómata Celular de Red de Gas (LGCA)	11
Que es el cáncer?.....	13
CAPITULO 4	15
Propuesta de solución.....	15
Capítulo 5.....	23
Simulación y Resultados	23
Escenario de la simulación 2	25
Escenario de la simulación 3	28
Escenario de la simulación 4	30
CAPITULO 6	33
Conclusiones	33
Trabajos futuros.....	34
RERERENCIAS	36
APENDICE	37
Clase BorrarFichero.....	37
Clase Cancer.....	39
Clase CrearFichero.....	45
Clase Glaticce	47
Clase Gmuerenxy.....	50
Clase Gnacenxy	53
Clase Inicio	56
Clase Propagar.....	60

CAPITULO 1

Introducción

Hoy en día existe la visión predominante de que la invasión de un tumor es consecuencia de la mutación de las células, pero también dos de las características del crecimiento de un tumor son la proliferación no controlada y la migración de células anormales. Una pregunta en esta dirección es, cuales son los mecanismos que disparan la progresión de un neoplasma¹ benigno a tumores invasivos malignos. Experimentos de células glioma² muestran una relación entre el comportamiento migratorio y el proliferativo.

En este proyecto terminal nos centraremos en el glioblastoma multiforme (tumor que es particularmente agresivo e invasivo) mostraremos la recurrencia del glioblastoma multiforme al poco tiempo de una intervención quirúrgica, debido a esto no se puede atribuir exclusivamente su recurrencia a la mutación ya que por la intervención quirúrgica no existe ninguna célula maligna, por lo que la transición a fenotipos tumorales invasivos puede explicarse sobre la base del mecanismo microscópico “ir o crecer” (dicotomía de migrar/proliferar) y el nivel de oxígeno en el ambiente del crecimiento de un tumor.

En este proyecto terminal construiremos un ejemplo que apoya la hipótesis migración proliferación para explicar el proceso de invasión de un tumor, auxiliándonos de un autómata celular de retícula de gases.

Los autómatas celulares son sistemas dinámicos discretos cuya dinámica se define en una retícula regular en donde cada uno de los sitios de la retícula se le llama célula, la cual toma un valor de un conjunto finito de valores llamados estados. El estado que se le asigna a una célula está influenciado por los valores que tienen sus vecinos cercanos, la función de evolución del

Justificación

El cáncer en la actualidad es una de las principales causas de muerte en el mundo por tal motivo, se han realizado investigaciones que han logrado importantes avances relacionados con el descubrimiento de los cambios que ocurren en las células de los tumores malignos, así como también en la evolución de los enfoques de diagnóstico; la investigación sobre las neoplasias malignas ha colaborado en la lucha para revertir o controlar este mal.

¹**Neoplasma:** masa anormal de tejido, cuyo crecimiento excede y está descoordinado con el de los tejidos normales y que persiste en su anormalidad después de que haya cesado el estímulo que provocó el cambio

²**Glioma:** Es un tipo de neoplasia que se produce en el cerebro o en la médula espinal. Se llama glioma, ya que surge a partir de células gliales.

El proceso de la transformación de las células malignas es un proceso complicado en el que intervienen múltiples factores en su origen y mecanismo, muchos conocimientos acerca del cáncer se generaron a través del estudio de modelos experimentales en animales pero con limitaciones por el alto costo que genera el uso de seres vivos y el tiempo que se requiere para obtener resultados.

Partiendo de las observaciones y características básicas del cultivo tradicional de células cancerosas y combinado con la introducción de los modelos de computo, se han creado instrumentos, que han demostrado ser una poderosa herramienta para imitar, comparar y explicar el comportamiento de la célula, se ha logrado una mejor comprensión de como los tumores pueden ser dirigidos de forma independiente por factores que influyen en el crecimiento, logrando así nuevas perspectivas sobre las estrategias de tratamiento

A través de un autómata celular se pueden mostrar patrones y generar indicadores en un tiempo dado tomando como base la observación y el análisis de su comportamiento, esta información arrojada puede ir modificando las reacciones y tratamientos a problemas que se han planteado, es aquí donde surge la importancia de diseñar un modelo sobre células tumorales que bajo algunas particularidades y especificaciones reproduzca la invasión en un tejido, el cual podría servir como una herramienta de apoyo para la investigación médica sobre este mal.

La pregunta que nos interesa contestar es cuales son los mecanismos celulares que generan el surgimiento de un fenotipo tumoral invasivo a expensas de uno proliferativo ya que el punto de vista preponderante está basado en la mutación. Las mutaciones aleatorias de combinaciones apropiadas de genes pueden cambiar el fenotipo de proliferativo a invasivo, o pueden disparar cualquier otro cambio fenotípico.

Objetivo General.

Diseñar un autómata celular de retícula de gases que permita simular la fase invasiva de un tumor bajo condiciones de escasez de oxígeno

Objetivos específicos

Diseñar un algoritmo basado en autómatas celulares de dos dimensiones que permita simular la fase invasiva de un tumor bajo condiciones de escasez de oxígeno.

Implementar el algoritmo diseñado.

Simular el comportamiento del algoritmo implementado.

Validar los resultados generados por la simulación y de ser necesario hacer ajustes al algoritmo.

Crear una interfaz gráfica para el algoritmo validado.

CAPITULO 2

Estado del arte

El glioblastoma multiforme muestra una conducta altamente invasiva. Al parecer la propagación de las células invasivas es guiada por medio de los tractos fibrosos de la materia blanca del cerebro

El más común pero también el más maligno de los gliomas es el glioblastoma multiforme. La terapia convencional para el glioblastoma multiforme consiste en una microcirugía que extirpa la masa que forma el tumor seguido de radiaciones. La quimioterapia se prescribe en pacientes con recurrencia del tumor. La tasa media de supervivencia no excede de los 10 a los 16 meses aunque hay un porcentaje de pacientes que logran sobrevivir por varios años.

La proximidad del tumor a áreas del cerebro vitales así como la misma estructura del cerebro son aspectos limitantes en la ejecución de las microcirugías que eliminan a un glioblastoma multiforme. Pacientes con una expectativa de vida limitada frecuentemente no soportan una reducción significativa de su calidad de vida relacionada a serios déficits neurológicos. Las estrategias quirúrgicas generalmente se eligen en base a la experiencia personal o la intuición por parte del equipo de médicos, parámetros bastante subjetivos, lo ideal es formalizar el análisis de elección de los factores antes mencionados. Los efectos de la radiación y la quimioterapia dependen de la localización del tumor, tamaño, geometría y de la estructura del tejido, si como de factores biológicos. La dosis de radiación que se suministra con seguridad está limitada por la radio sensibilidad de los tejidos vecinos.

Un glioblastoma multiforme típico consiste de una masa vascularizada de forma irregular con un corazón necrótico, no existe una frontera bien delimitada entre el tumor y la parte del tejido cerebral vecino. Las células del tumor se infiltran en el tejido cerebral circundante y pueden llegar a ser encontradas a grandes distancias de la masa que compone al tumor principal en áreas apenas visibles por los análisis histopatológicos de rutina.

La proliferación no controlada y la migración de células anormal son dos de las características principales del crecimiento de un tumor.

La progresión del cáncer puede ser descrita como una secuencia de rasgos o fenotipos³ que las células tienden a adquirir, si un neoplasma se convierte en un tumor invasivo maligno. Hanahan [1] ha identificado seis fenotipos de células de cáncer que están caracterizadas por un potencial de proliferación sin límites

³**Fenotipo:** característica o rasgo observable de un organismo, como su morfología, desarrollo, propiedades bioquímicas, fisiología y comportamiento.

en ambientes independientes de crecimiento, evasión de apoptosis⁴, angiogénesis⁵, altos promedios de movilidad (invasión) y metástasis⁶ Las células de tumor cambian su fenotipo debido a mutaciones que son adquiridas durante la progresión del cáncer. Inicialmente las mutaciones alteran el control de la proliferación de las células que lleva a una división celular no controlada [1] entonces las células transformadas forman una lesión neoplástica y el tumor puede crecer a un tamaño en el que el manejo de la difusión de oxígeno se vuelve insuficiente (hipoxia⁷) para soportar un crecimiento posterior.

Un ambiente hipoxico y los altos promedios de mutación de las células tumorales debido a su material genético puede llevar a la emergencia de fenotipos caracterizados por metabolismo anaeróbico [2] alta movilidad y/o angiogénesis. La principal diferencia entre células benignas y malignas es la presencia de células con una movilidad creciente en etapas posteriores [3]. Dada la emergencia de una población de células de tumor altamente movibles es importante para la progresión de un tumor hacia ser maligno estar interesado en las condiciones que conducen a la transición de la proliferación al fenotipo móvil.

Experimentos de células glioma (giese et al,2003)[3] muestra una relación entre comportamiento migratorio y proliferativo. El movimiento de las células y su proliferación son procesos mutuamente exclusivos ya que el alto movimiento de las células glioma tiende a bajar sus tasas de proliferación; esto es la célula prolifera solo cuando ellas no se mueven este fenómeno es conocido como dicotomía migración proliferación (giese et al .1996 a,b). EN el 2010 godlewski et al.[4] Identificaron una proteína que es responsable de la regulación de la migración y la proliferación de acuerdo al estrés metabólico.

⁴**Apoptosis:** Proceso ordenado por el que la célula muere ante estímulos extra o intracelulares

⁵**Angiogénesis:** Formación de vasos sanguíneos. La angiogénesis tumoral es el crecimiento de vasos sanguíneos nuevos que los tumores necesitan para crecer

⁶**Metástasis:** Diseminación del cáncer de una parte del cuerpo a otra. Un tumor formado por células que se han diseminado se llama "tumor metastásico" o "metástasis".

⁷**Hipoxia:** Trastorno en el que hay una disminución del suministro de oxígeno a un tejido.

CAPITULO 3

Antecedentes

Los autómatas celulares son sistemas dinámicos discretos cuya dinámica se define en una retícula regular en donde cada uno de los sitios de la retícula se le llama célula, la cual toma un valor de un conjunto finito de valores llamados estados. El estado que se le asigna a una célula está influenciado por los valores que tienen sus vecinos cercanos, la función de evolución del autómata toma como argumentos los valores de la vecindad y asigna un nuevo estado a la célula.

La Estructura de un Autómata Celular

Los autómatas celulares son sistemas dinámicos discretos que involucran cuatro elementos de construcción:

- El espacio celular
- Los estados de las células
- La configuración de vecindades
- La regla de evolución
- El espacio celular

Los autómatas celulares están definidos por un arreglo de células de dimensión 'd', llamado espacio celular; donde cada uno de los elementos del arreglo se denomina célula , que es un término tomado de las ciencias biológicas como algunos otros tales como ``evolución", ``vida", ``muerte"; que son adecuados para comprender el comportamiento individual de los elementos

El espacio celular, en principio y tal como fue definido originalmente, es infinito en todas sus direcciones, y posteriormente debido principalmente a las limitaciones prácticas de cómputo, el espacio celular puede estar acotado .se muestra un ejemplo de un espacio celular finito en la figura 3.1

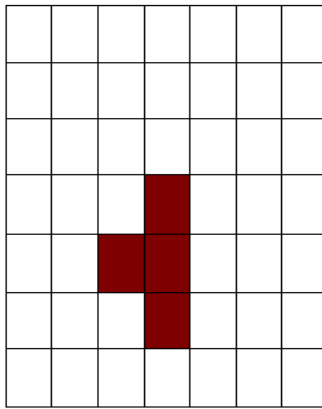


Figura 3.1

Para poder simular el espacio celular infinito, se ha decidido tomar condiciones periódicas de frontera, esto es, que la columna de célula s siguiente a la última es la primera, y la columna de célula s anterior a la primera es la última. Cuando $d=1$ el espacio celular es un anillo, en $d=2$ se forma un espacio toroidal.

El conjunto de estados de las células

El estado de cada célula es el valor que tendrá ésta en el siguiente paso del tiempo y está determinado por el estado actual de ella misma y de las células vecinas más cercanas en un radio de vecindad predeterminado. Un autómata celular lineal está definido por el par (k,r) , donde 'k' es el número de estados para cada célula y 'r' es el radio de vecindad. Los autómatas celulares que se usaran como muestra a continuación son los acl $(2,1)$ que significa: "autómata celular lineal de dos estados y radio de vecindad de uno".

La configuración de vecindades

La manera en que las células actúan dentro del espacio celular, está determinada por cómo son afectadas por el entorno que las rodea. A este entorno se le llama vecindad.

Así, la vecindad de una célula, llamada célula en transición es el conjunto de células cercanas a ella.

Dos de los conjuntos de vecinos más usados son los que tienen en cuenta sólo las celdas izquierda/derecha/arriba/abajo (vecinos de Von Neumann) y los que tratan las diagonales (vecinos de Moore). La figura 3.2 muestra un ejemplo.

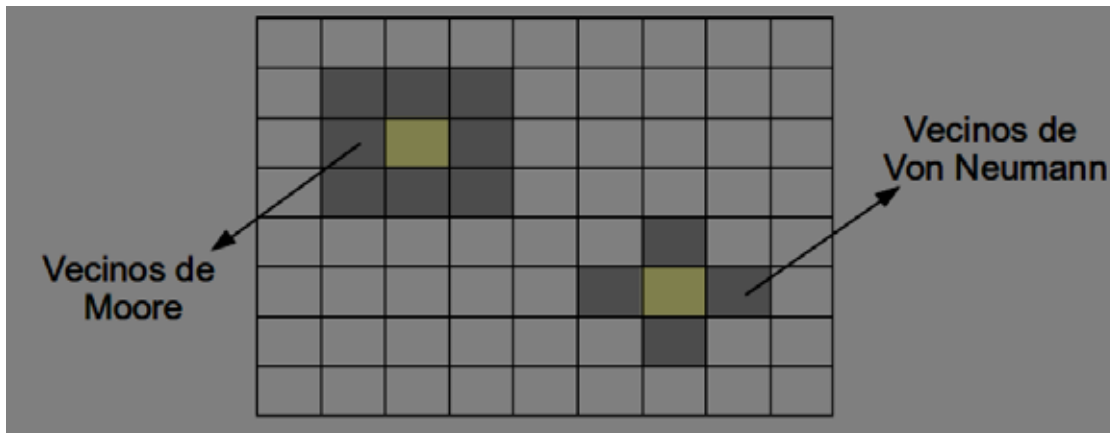


Figura 3.2

La regla de evolución local

Cada célula del espacio celular, toma su valor en dependencia de la configuración de su vecindad.

A través de un autómata celular se pueden mostrar patrones y generar indicadores en un tiempo dado tomando como base la observación y el análisis de su comportamiento, esta información arrojada puede ir modificando las reacciones y tratamientos a problemas que se han planteado, es aquí donde surge la importancia de diseñar un modelo sobre células tumorales que bajo algunas particularidades y especificaciones reproduzca su invasión en un tejido, el cual podría servir como un herramienta de apoyo para la investigación médica sobre este mal.

Autómata Celular de Red de Gas (LGCA)

El modelo de retícula de gas tiene su origen en los autómatas celulares

En 1986 Uriel Frish, Brosl Hasslacher, e Yves Pomeau demostraron que el movimiento interno de un fluido, no tiene que ser tan detallado como la dinámica molecular real, esto con el fin de dar lugar a la mecánica de fluido realista. Esto implica que un fluido puede ser modelado y simulado por medio de un conjunto de partículas ficticias con igual masa y moviéndose con la misma rapidez, aunque con dirección diferente (vector desplazamiento).

El autómata celular de red de gas (LGCA) es un método, que es diferente a la dinámica molecular por un lado y por otro a los métodos basados en la discretización de un sistema de ecuaciones diferenciales parciales no lineales. La ecuación de Navier-Stokes junto con la ecuación de continuidad, son conocidas por describir la conducta de un fluido en una amplia gama de

situaciones, desde una corriente no turbulenta de un río hasta condiciones atmosféricas extremas con turbulencias complejas. A saber:

$$\frac{\partial P}{\partial t} + \nabla \cdot (\rho u) = 0$$

Donde ρ es la densidad del fluido y u su velocidad, la ecuación 1 se simplifica cuando la densidad es constante, es decir, se tiene un fluido incompresible esto da lugar a la ecuación

$$u \cdot \nabla = 0$$

Diferentes fluidos como el aire, agua o aceite se caracterizan por tener diferentes valores de masa, densidad y viscosidad. Los flujos incompresibles de estos fluidos obedecen de la misma forma a la ecuación de Navier-Stokes, mientras que sus interacciones microscópicas son muy diferentes. Esta ecuación es no lineal lo que implica que tener una solución analítica es una excepción; además pertenece a la segunda ley del movimiento de Newton lo que significa que la razón de cambio del momento de una partícula en el fluido es igual a la fuerza total que actúa sobre dicha partícula.

Al considerar a cualquier fluido como formado por una gran cantidad de partículas ficticias, podemos imaginarlas habitando una red o malla la cual estará preparada para que ahí las partículas se muevan, siempre cuidando de respetar el principio de exclusión, a saber, que dos partículas no ocupen el mismo espacio en el mismo instante de tiempo, si dos o más partículas llegan al mismo sitio, pueden chocar, algunas colisiones pueden ocasionar que las partículas se dispersen, lo que significa que su velocidad cambia, mientras que otras no. Cabe notar que las colisiones no cambian el número total de partículas, ni el vector suma de sus desplazamientos, ni la masa y tampoco el momento, es decir, estos se conservan. La conservación del momento y de la masa, permite obtener una ecuación que describe la evolución de estas cantidades en el modelo.

Un autómata celular de red de gas (LGCA) es un autómata celular que tiene un conjunto de estados así como una dinámica.

Se tiene una lattice cuadrada regular d-dimensional, donde $L_1 \dots L_d$ son las dimensiones de la lattice. Las partículas se mueven sobre la lattice discreta con velocidades discretas, es decir, que las partículas saltan en pasos de tiempo discretos $k \in \mathbb{N}$ de un nodo a un nodo vecino.

Que es el cáncer?

El cáncer es una enfermedad provocada por un grupo de células que se multiplican sin control y de manera autónoma, invadiendo localmente y a distancia otros tejidos. En general tiende a llevar a la muerte a la persona afectada, si no se somete a un tratamiento adecuado. Se conocen más de 200 tipos diferentes de cáncer, los más frecuentes son los de piel, pulmón, mama y colonrectal.

La malignidad del cáncer es variable, según la agresividad de sus células y demás características biológicas de cada tipo tumoral. En general el comportamiento de las células cancerosas se caracteriza por escapar al control reproductivo que requería su función original, perdiendo sus capacidades primitivas y adquiriendo otras que no les corresponden, invadiendo de forma progresiva y por distintas vías órganos próximos, o incluso diseminándose a distancia (metástasis), con crecimiento y división más allá de los límites normales del órgano al que pertenecían primitivamente, diseminándose por el organismo fundamentalmente a través del sistema linfático o el sistema circulatorio, y ocasionando el crecimiento de nuevos tumores en otras partes del cuerpo alejadas de la localización original.

Las diferencias entre tumores benignos y malignos consisten en que los primeros son de crecimiento lento, no se propagan a otros tejidos y rara vez recidivan tras ser extirpados, mientras que los segundos son de crecimiento rápido, se propagan a otros tejidos, recidivan con frecuencia tras ser extirpados y provocan la muerte en un periodo variable de tiempo, si no se realiza tratamiento.

Los tumores benignos pueden recurrir localmente en ciertos casos, pero no suelen dar metástasis a distancia ni matar al portador, con algunas excepciones. Las células normales al entrar en contacto con las células vecinas inhiben su multiplicación, pero las células malignas no tienen este freno. La mayoría de los cánceres forman tumores sólidos, pero algunos no, por ejemplo la leucemia.

El proceso por el cual se produce el cáncer es causado por anomalías en el material genético de las células. Estas anomalías pueden ser provocadas por distintos agentes carcinógenos, como la radiación ionizante, ultravioleta, productos químicos procedentes de la industria, del humo del tabaco y de la contaminación en general, o de agentes infecciosos como el virus del papiloma humano o el virus de la hepatitis B.2 Otras anomalías genéticas cancerígenas son adquiridas durante la replicación normal del ADN, al no corregirse los errores que se producen durante dicho proceso, o bien son heredadas y, por consiguiente, se presentan en todas las células desde el nacimiento y originan mayor probabilidad de que se presente la enfermedad.

Existen complejas interacciones entre el material genético y los carcinógenos, un motivo por el que algunos individuos desarrollan cáncer después de la exposición a carcinógenos y otros no. Nuevos aspectos de la genética del cáncer, como la metilación del ADN y los microARNs, están siendo estudiados como importantes factores a tener en cuenta por su implicación

Las anomalías genéticas encontradas en las células cancerosas pueden consistir en una mutación puntual, translocación, amplificación, delección, y ganancia o pérdida de un cromosoma completo. Existen genes que son más susceptibles a sufrir mutaciones que desencadenen cáncer. Esos genes, cuando están en su estado normal, se llaman protooncogenes, y cuando están mutados se llaman oncogenes.⁷ Lo que esos genes codifican suelen ser receptores de factores de crecimiento, de manera que la mutación genética hace que los receptores producidos estén permanentemente activados, o bien codifican los factores de crecimiento en sí, y la mutación puede hacer que se produzcan factores de crecimiento en exceso y sin control.

El cáncer se clasifica según el tejido a partir del cual las células cancerosas se originan. El diagnóstico está, en gran medida, influenciado por el tipo de tumor y la extensión de la enfermedad

CAPITULO 4

Propuesta de solución

Definición de reglas del modelo a realizar.

El propósito de este módulo es analizar y redactar las reglas necesarias que serán utilizadas para el análisis y diseño del autómata celular de retícula de gases en el cual se pretende mostrar una correspondencia con los sistemas reales, tomando las características biológicas de las células tumorales consideradas en la publicación [6] .

Para este proyecto terminal ocuparemos una retícula regular $L = L1 \times L2$ donde $L1$ y $L2$ son la dimensión de la retícula, las células tumorosas se mueven sobre la retícula brincando en etapas discretas de tiempo desde un nodo a su nodo vecino.

Existe un conjunto de canales de velocidad (r, c_i) , $i = 1 \dots b$, asociado con cada nodo $r \in L \subset \mathbb{Z}^d$ de la lattice. El parámetro b es el número de coordinación, que representa el número de canales de desplazamiento que coincide con el número de vecinos que tiene un nodo en la lattice.

Los canales de velocidad serán representados por vectores con las siguientes coordenadas

$$c_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, c_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, c_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, c_4 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, c_5 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

B es el número de canales con velocidad 0 (en reposo), lo que quiere decir es que en cada nodo podrán habitar 5 células tumorosas, como se muestra en la figura 4.1

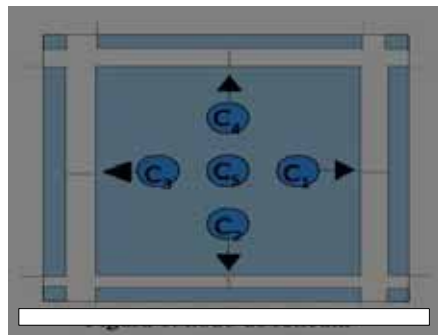


Figura 4.1

- Denotamos $\tilde{b} = b + B$ como número total de canales por nodo que están

ocupados.

- La densidad de un nodo r es la cantidad total de partículas presentes en un instante de tiempo $k \in \mathbb{N}$ y se denota por:

$$n(r,k) = \sum_{i=1}^{\tilde{b}} \eta_i(r,k)$$

Donde $\eta_i(r,k) = 0$ o 1 .
1 = ocupado.
0 = libre.

- El parámetro $c \in [0,5]$ representa la ocupación máxima del nodo y depende de la disponibilidad de oxígeno.
- El número de células tumorosas en cada nodo es proporcional al oxígeno disponible en cada nodo.
- El oxígeno es distribuido homogéneamente en la lattice.

La dinámica celular

La dinámica celular será modelada por un conjunto de reglas del autómata (operadores: operador P de propagación, operador O de orientación y operador cinético R) la composición R, O, P de los 3 operadores será aplicada independientemente a cada nodo de la retícula en cada etapa de tiempo.

Suposición sobre la dinámica de las células

Las células tumorales viajan de forma individual o en grupo, en este proyecto terminal se toma como base que viajan de forma individual.

Las células tumorales se mueven aleatoriamente

La célula tumoral se considera inhibida si todos los puntos adyacentes a los que pueda viajar están ocupados por otras células tumorales y se mantendrá en reposo hasta que algún espacio vecino esté disponible

El promedio mitótico y apoptótico dependen de la densidad de células locales.

Un mecanismo de crecimiento y desplazamiento influenciado por el proceso de muerte -nacimiento será considerado

El nivel oxígeno es un factor importante para la proliferación y muerte

Propagación P

El desplazamiento celular sobre el sustrato es modelado por la etapa de propagación.

Para la aplicación de P todas las células son transportadas simultáneamente a los nodos en la dirección de sus vecindades, esto es una célula que vive en el nodo r y canal c_i , (r, c_i) en un tiempo $k \in \mathbb{Z}^+$ es movida a un canal vecino $(r + m_{c_i}, c_i)$ durante una etapa de tiempo donde:

- m es un número entero y determina la velocidad
- m_{c_i} es el desplazamiento de la célula.
- Las células en el canal de reposo no se mueven dado que tienen velocidad cero.

El estado de un canal del nodo r :

$$(r + m_{c_i}, c_i)$$

Después de la propagación es:

$$n_i^p = (r + m_{c_i}, k + \aleph) = n(r, k)$$

Donde \aleph es la etapa de tiempo del automata celular.

Trabajamos el desarrollo del algoritmo que describe el movimiento celular basado en las reglas establecidas en el módulo 1 y considerando los operadores P, O y R. y teniendo como guía el modelo computacional del ciclo de vida de las células realizado por Heiko Enderling [5]. Ver figura 4.2

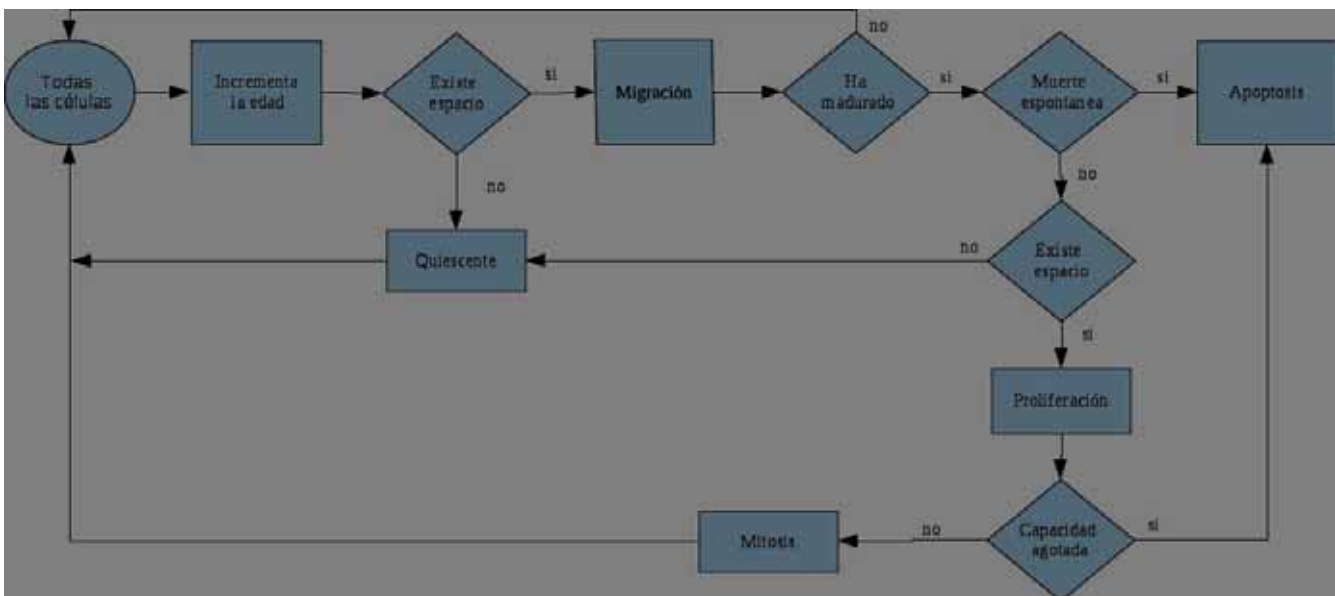


Figura 4.2

- Definiremos el tamaño de la lattice el cual es $L=120$ esto por el problema de visualización de los pixeles.
- A cada sitio de la lattice lo llamaremos nodo y a cada canal del nodo lo llamaremos célula.
- Los estados posibles de la célula será 0=sana, 1=infectada.

La configuración inicial se compone de:

- Definir el porcentaje de la mitosis y de la apoptosis de las células cancerígenas en experimento.
- Células sanas en la lattice.
- Una cantidad de células cancerígenas definida aleatoriamente y distribuidas uniformemente de forma automática.

En un lapso de tiempo, la lattice completa se actualiza de forma paralela y sincronizada, de acuerdo a las reglas que rigen al autómata celular

Regla de Orientación. O

Para cada uno de los nodos (r, c_i) con $c_i (1-5)$, cada una de las células contenidas en el nodo r cambiara de posición dentro del mismo nodo r (reorientación) haciéndolo de forma aleatoria debido a que en este organismo no hay procesos deterministas. Como se muestra en la figura 4.3

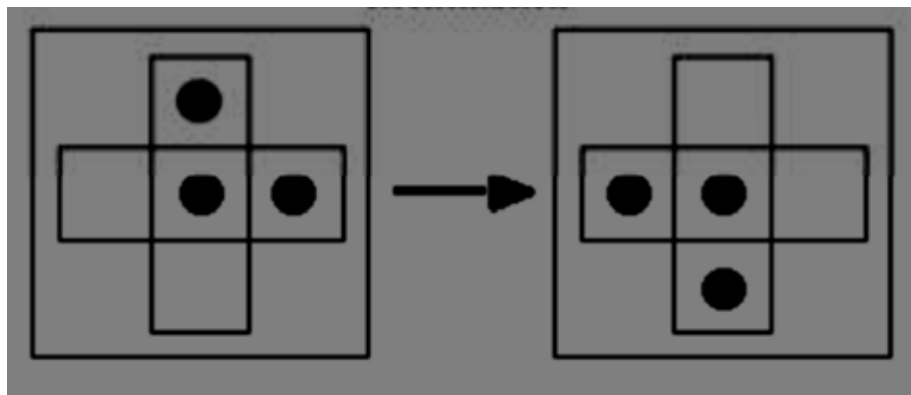


Figura 4.3

Regla Cinética C

La evolución de los estados de un nodo en el LGCA se encuentra definido por los números de ocupación $\eta_i(r, k) \in \{0,1\}$ con $i=1 \dots \tilde{b}$.

Mitosis: se llevara a cabo si $\tilde{b} < 5$ en el nodo y en la posición (0,0) que es el canal de reposo existe una célula cancerígena, haciéndolo de forma aleatoria debido a que en este organismo no hay procesos deterministas, esto es al cumplirse la regla existe la probabilidad de que haya mitosis o no. Ejemplo figura 4.4

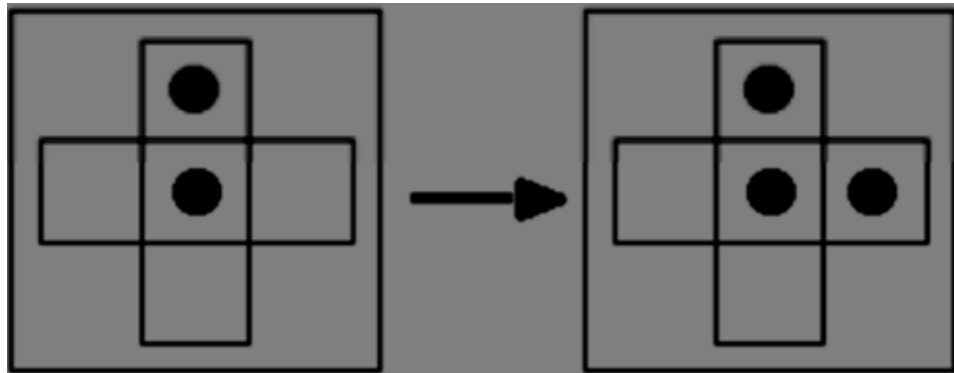


Figura 4.4

Apoptosis se llevara a cabo si $\tilde{b} \geq 3$ en el nodo, haciéndolo de forma aleatoria debido a que en este organismo no hay procesos deterministas, esto es al cumplirse la regla existe la probabilidad de que haya apoptosis o no. Ejemplo figura 4.5.

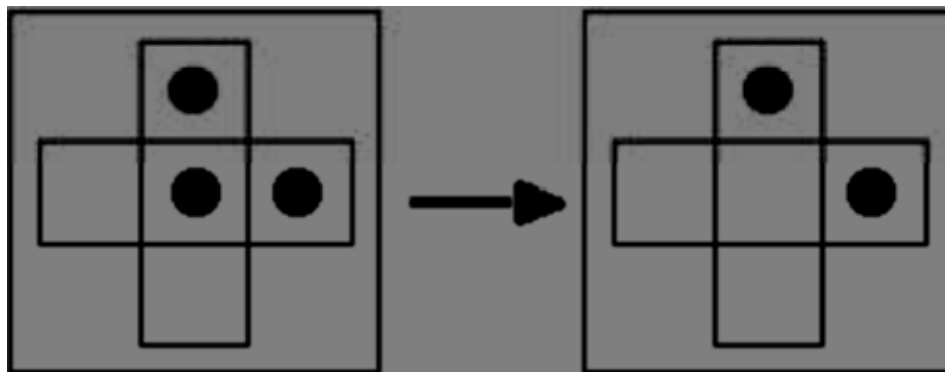


Figura 4.5

Regla de Propagación P

Para esta etapa todas las partículas se mueven al mismo tiempo hacia los nodos que les indica su velocidad. Esto es, una partícula que se encuentra en (r, c_i) durante el instante de tiempo k se moverá hacia otro nodo $(r + mc_i, c_i)$ donde m representa al número de unidades por saltar en la red, es decir, representa la Velocidad con la que se mueve la partícula la cual tomara valores de 0 a 3 y esta velocidad se asignara aleatoriamente. Ejemplo figura 4.6

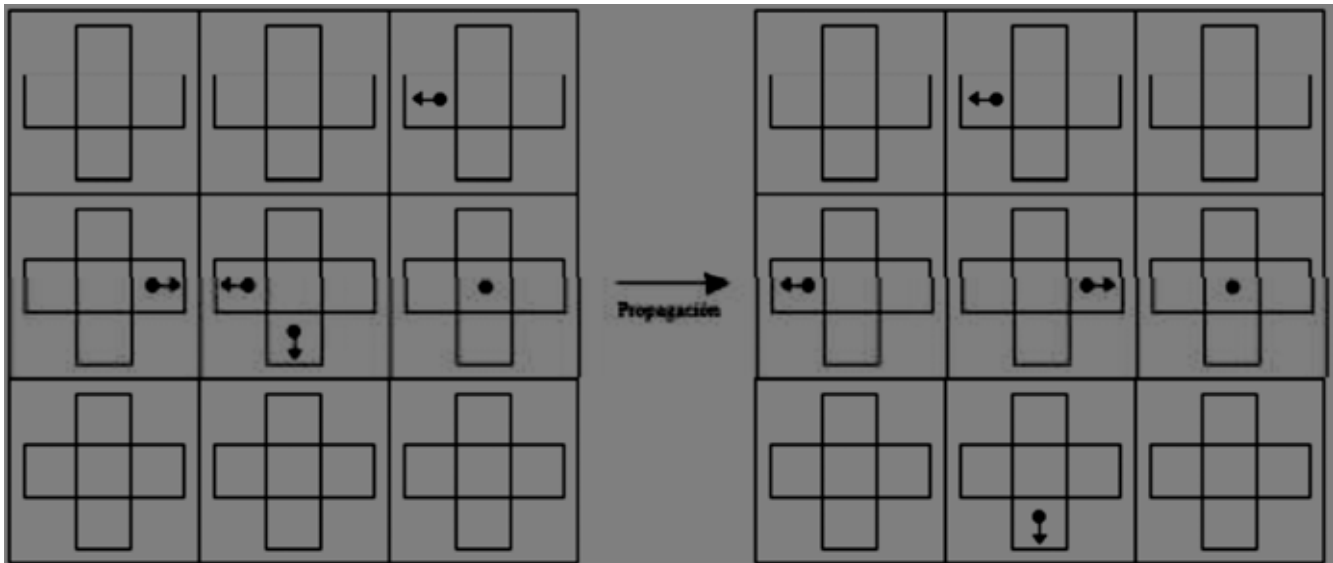


Figura 4.6

Con respecto a la implementación, se desarrolló lo siguiente:

Un módulo que recibe la información necesaria para inicializar el autómata celular, el cual valida que ésta sea la correcta y se encuentre con las especificaciones requeridas para iniciar la simulación de la dinámica celular, revisando su estructura y contenido, es tomada de la publicación [6].

Un módulo de procesamiento de información, el cual contiene el algoritmo que se diseñó específicamente para la simulación de la fase invasiva de un tumor por células tumorales, donde se procesa la información ya validada por el anterior módulo y realiza las iteraciones con el autómata celular para la correcta simulación del sistema, el cual toma en cuenta posibles cambios y variaciones de la información previamente ingresada, todo esto con forme avanza el experimento. El algoritmo matemático utilizado en nuestra simulación, fue diseñado y analizado por el alumno con apoyo del asesor, para su correcta implementación, siendo el diseño de éste una parte fundamental del proyecto terminal.

Una interfaz gráfica la cual contendrá las siguientes opciones:

Menú. En este menú se tendrá opción inicio del proceso. Figura 4.7

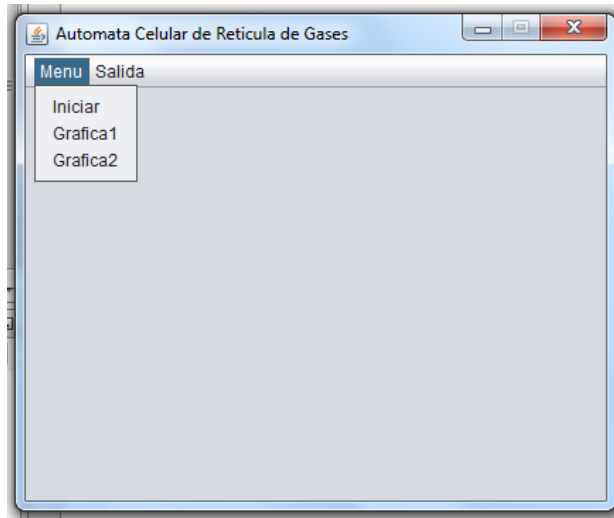


Figura 4.7

Opción para ver el estado de la retícula, donde se observará el comportamiento de las células a través del tiempo. Figura 4.8

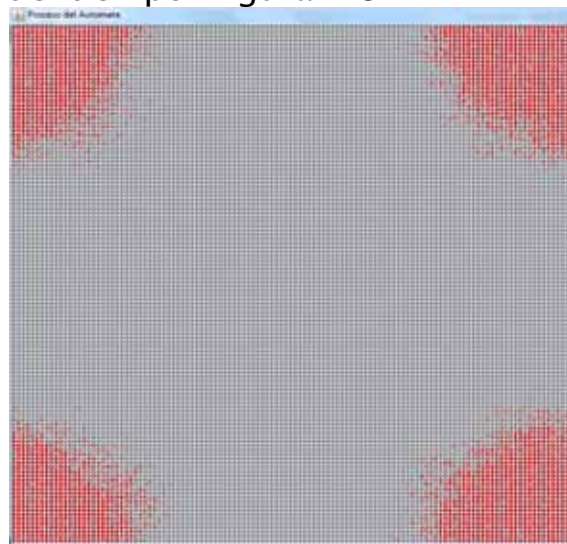


Figura 4.8

Opción para visualizar la gráfica Espacio-fase la cual mostrará el crecimiento y disminución de las células en el tiempo; ver el ejemplo mostrado en la figura 4.9.

Automata Celular

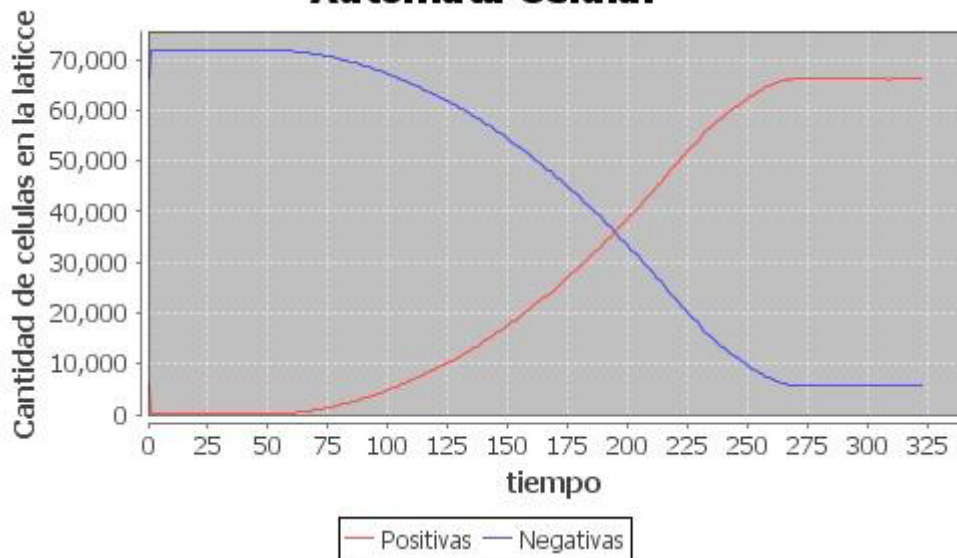


Figura 4.9

Un módulo que almacene la gráfica Espacio-fase de la dinámica celular, todo esto se almacenara en archivos jpg, indispensables para obtener las conclusiones de comportamiento de las células tumorales. Figura 4.10

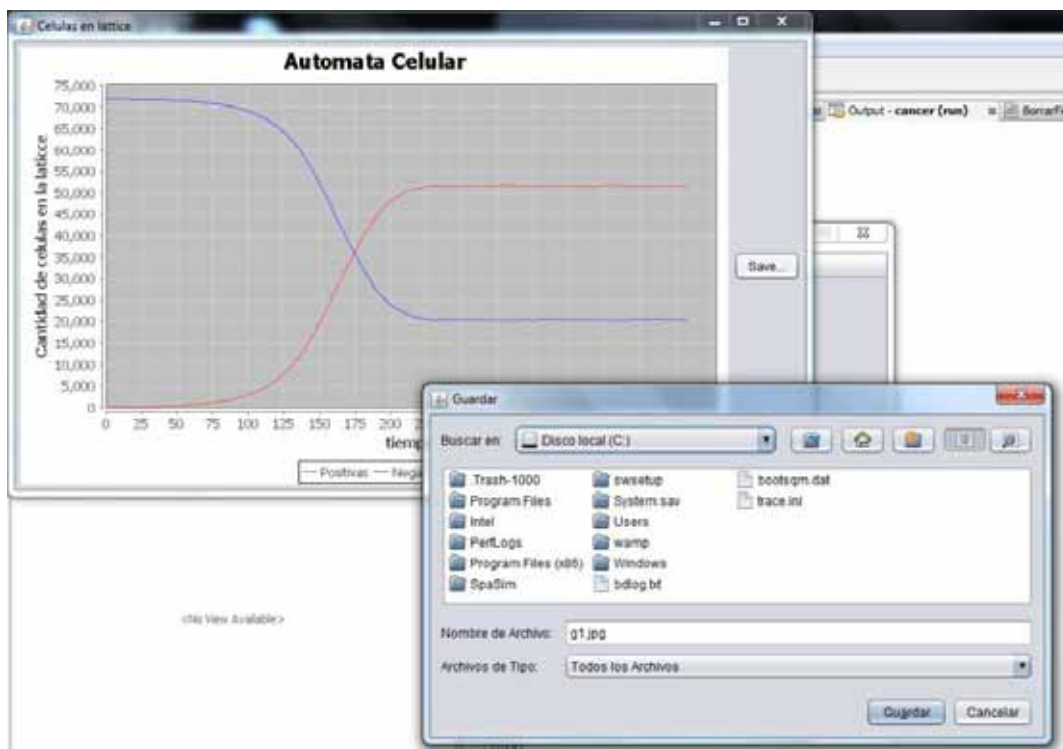


Figura 4.10

Capítulo 5

Simulación y Resultados

El tamaño de la lattice es de 120 x 120
Cantidad de nodos es de 14400
Cantidad de canales 72000

Escenario de simulación I

Tabla 5.1

Parámetro	Valores
Células cancerígenas	1
Canal de reposo	1
Promedio de probabilidad de mitosis	70%
Promedio de probabilidad apoptosis	30%
Posición de las células cancerígenas	centro

Con los parámetros de la tabla 5.1 se muestra la evolución del sistema

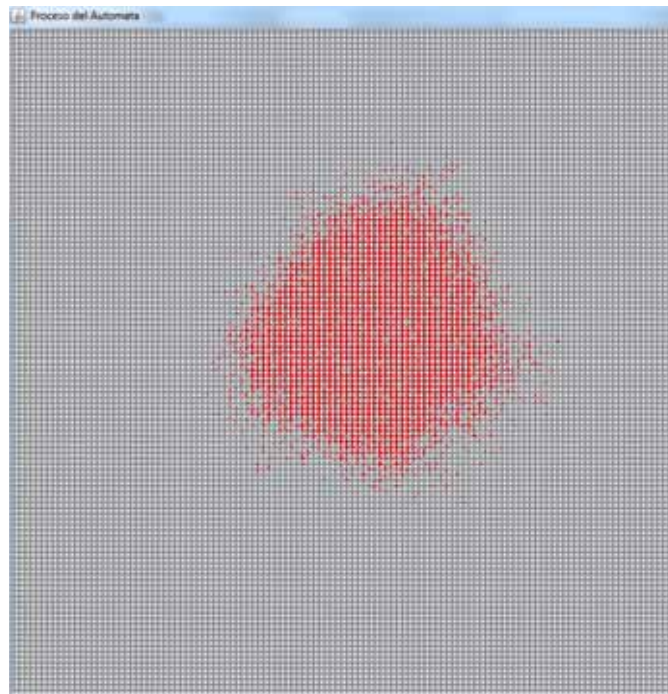


Figura 5.1 muestra el proceso de reproducción de la célula cancerígena (en color rojo) con su mayor concentración en el centro

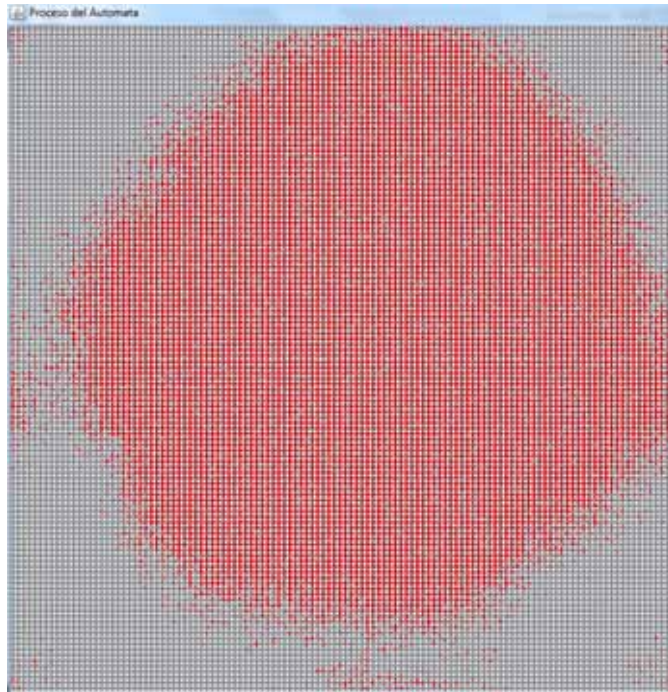
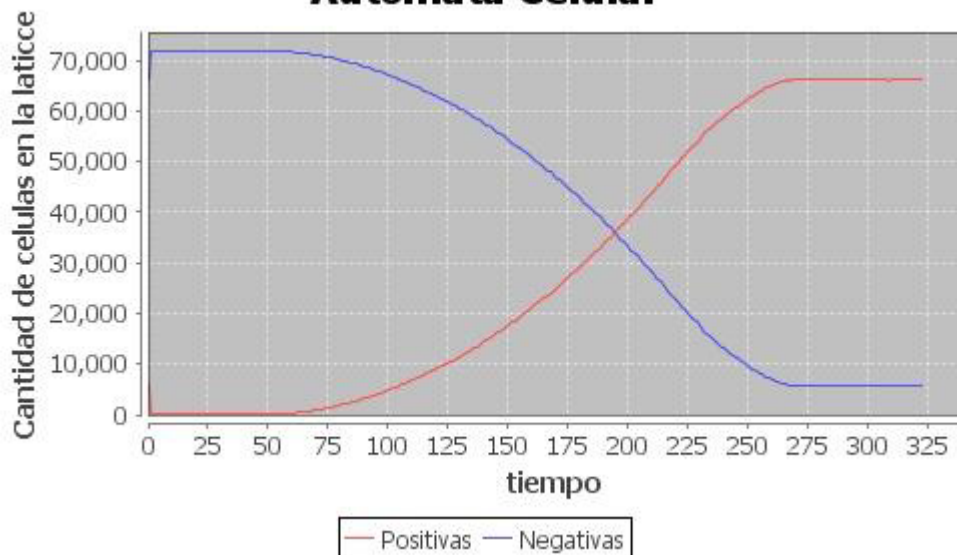


Figura 5.2 Se observa concentración de las células cancerígenas partiendo del centro y extendiéndose con la misma densidad.

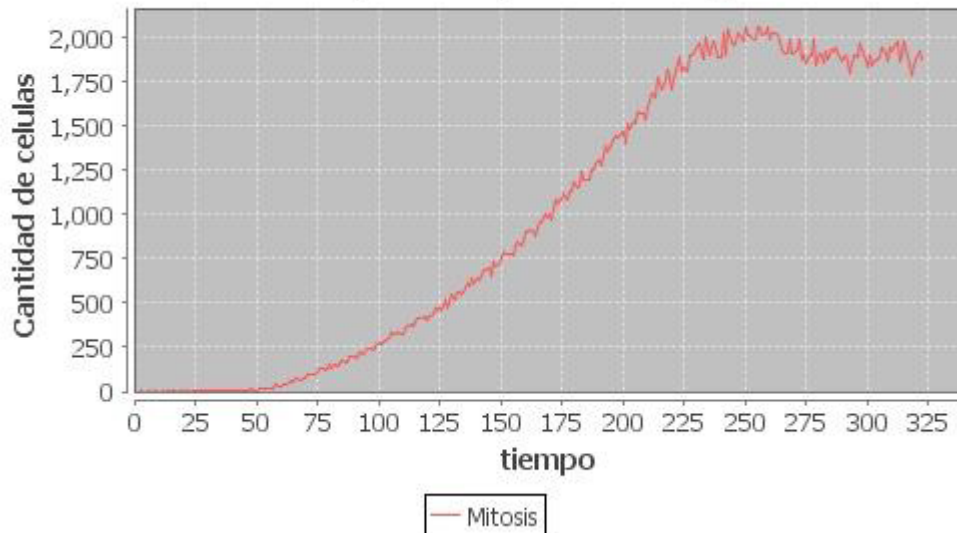
Con los parámetros de la tabla 5.1 se obtuvieron las siguientes graficas

Automata Celular



Grafica 5.1 se muestra el crecimiento de las células positivas y negativas durante la simulación, la invasión es en un lapso de tiempo corto

Mitosis por etapa de tiempo



Grafica 5.2 muestra el proceso de crecimiento de las células positivas durante el tiempo de ejecución del sistema

Escenario de la simulación 2

Tabla 5.2

Parámetro	Valores
Células cancerígenas	1
Canal de reposo	1
Promedio de probabilidad de mitosis	.05%
Promedio de probabilidad apoptosis	.95%
Posición de las células cancerígenas	centro

Con los parámetros de la tabla 5.2 se muestra la evolución del sistema

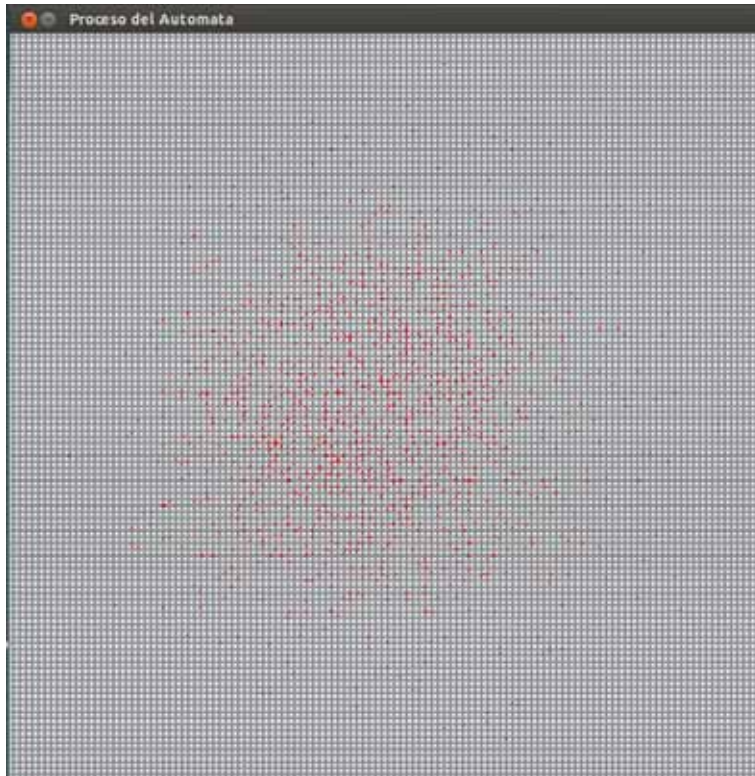


Figura 5.3 se observa que las células cancerígenas se empiezan a dispersar en la lattice.

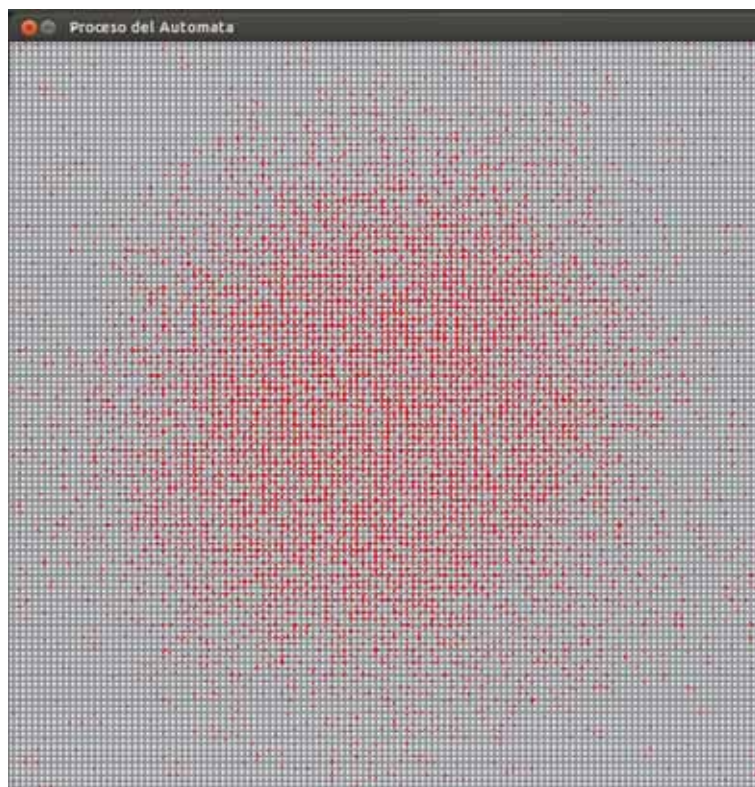
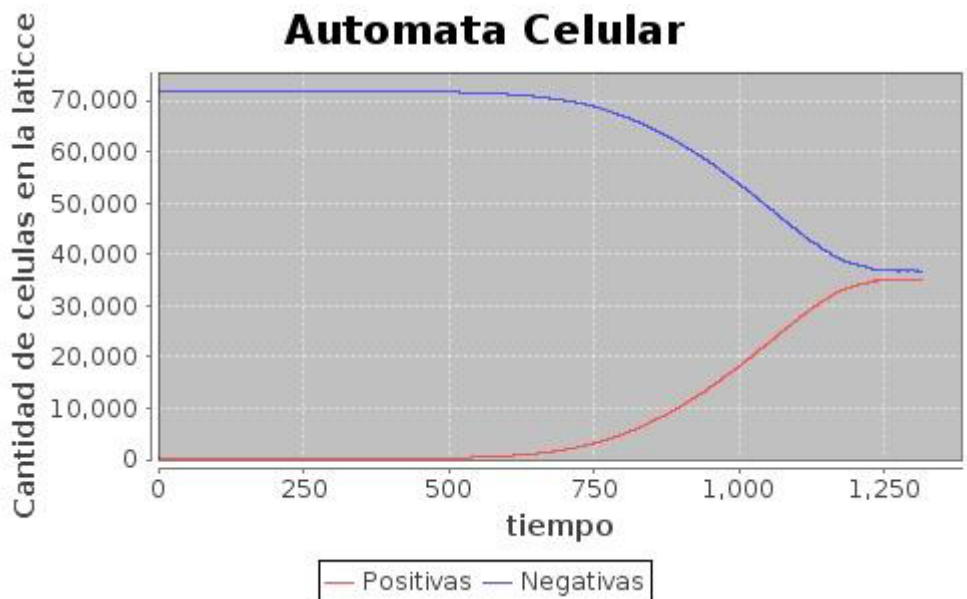
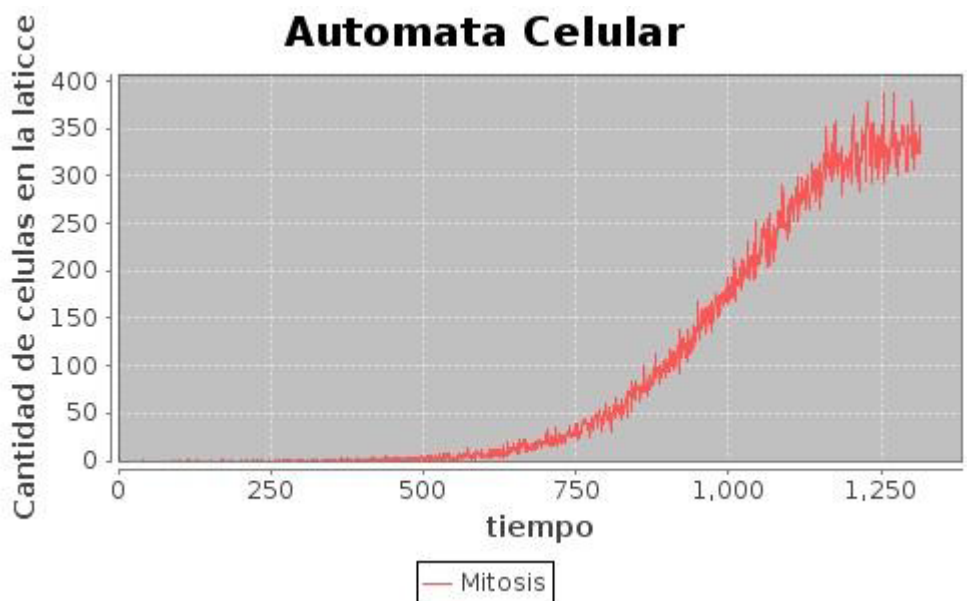


Figura 5.4 Las células cancerígenas están dispersas en casi toda la lattice y la concentración de las células es mínima en ciertos puntos de la lattice

Con los parámetros de la tabla 5.2 se obtuvieron las siguientes graficas



Grafica 5.3 se muestra el crecimiento de las células positivas y negativas durante la simulación con un tiempo mayor para la invasión



Grafica 5.4 muestra el proceso de apoptosis de las células positivas durante el tiempo de ejecución del sistema

Escenario de la simulación 3

Tabla 5.3

Parámetro	Valores
Células cancerígenas	1
Canal de reposo	2
Promedio de probabilidad de mitosis	.05%
Promedio de probabilidad apoptosis	.95%
Posición de las células cancerígenas	centro

Con los parámetros de la tabla 5.3 se muestra la evolución del sistema

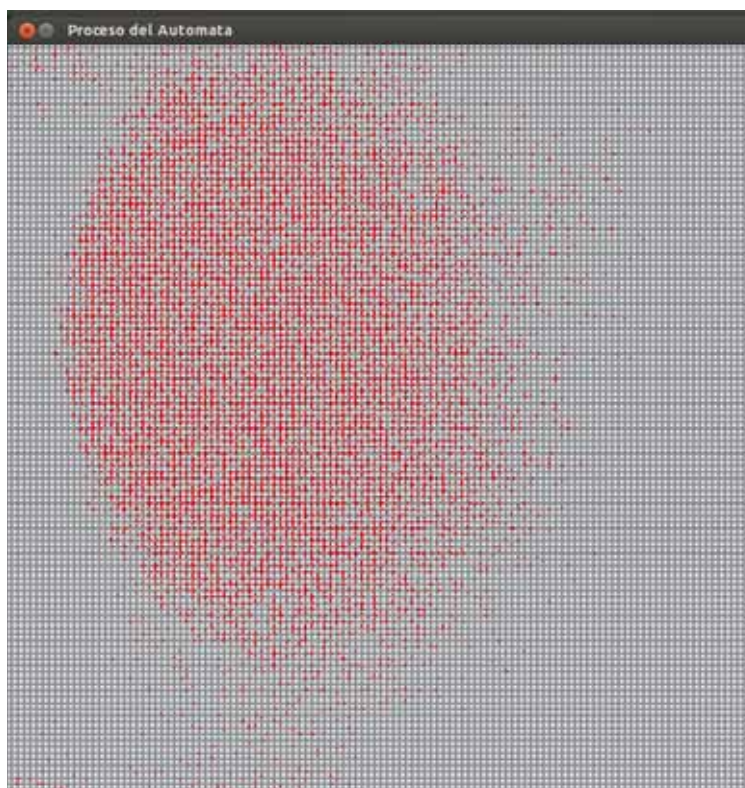


Figura 5.5 la dispersión de las células positivas es mayor debido al aumento de número de canales en reposo

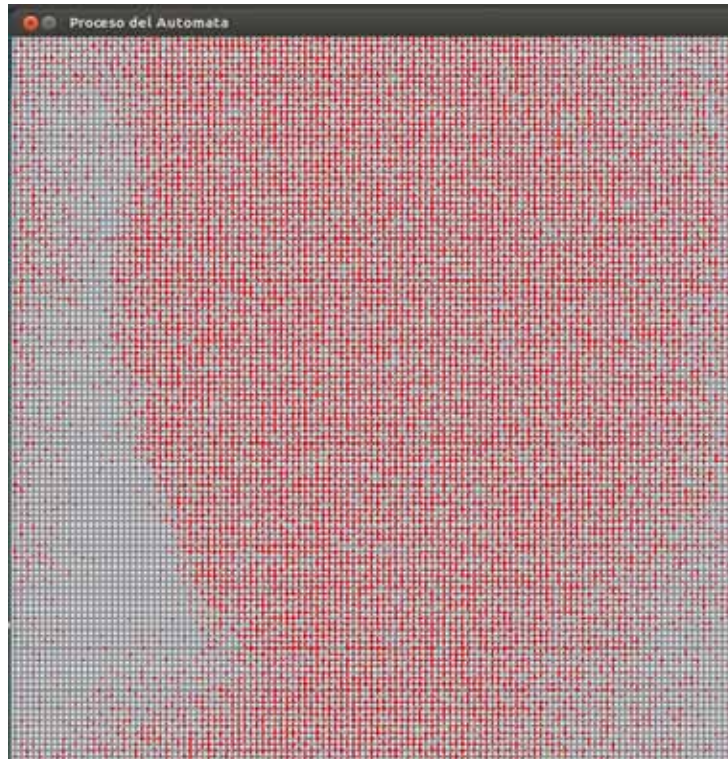
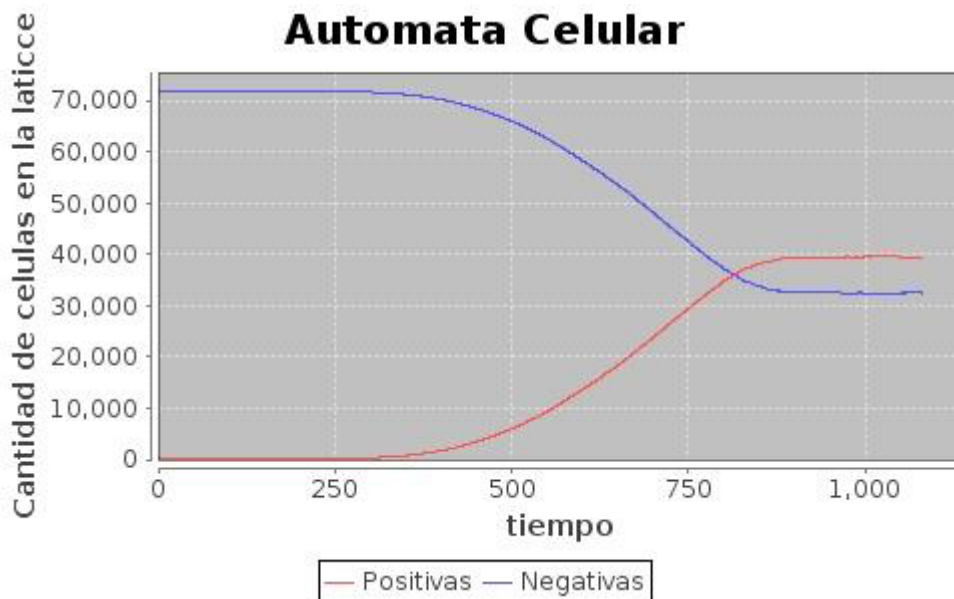
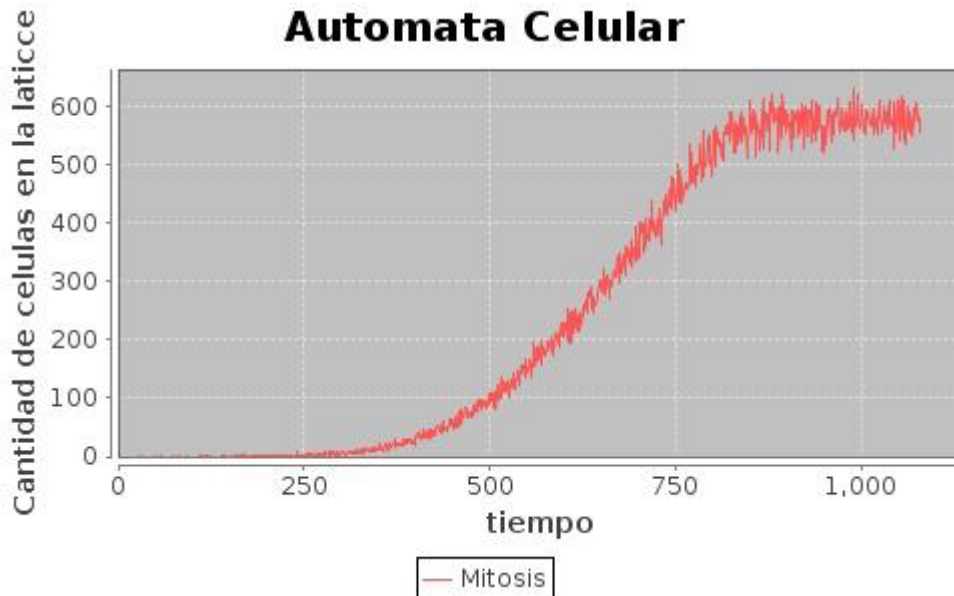


Figura 5.6 se muestra la lattice ya invadida de células cancerígenas con puntos de concentración dispersos

Con los parámetros de la tabla 5.3 se obtuvieron las siguientes graficas



Grafica 5.5 se muestra el crecimiento de las células positivas y negativas durante la simulación con un tiempo menor para la invasión



Grafica 5.6 muestra el proceso de apoptosis de las células positivas durante el tiempo de ejecución del sistema

Escenario de la simulación 4

Tabla 5.4

Parámetro	Valores
Células cancerígenas	1
Canal de reposo	3
Promedio de probabilidad de mitosis	.05%
Promedio de probabilidad apoptosis	.95%
Posición de las células cancerígenas	centro

Con los parámetros de la tabla 5.1 se muestra la evolución del sistema

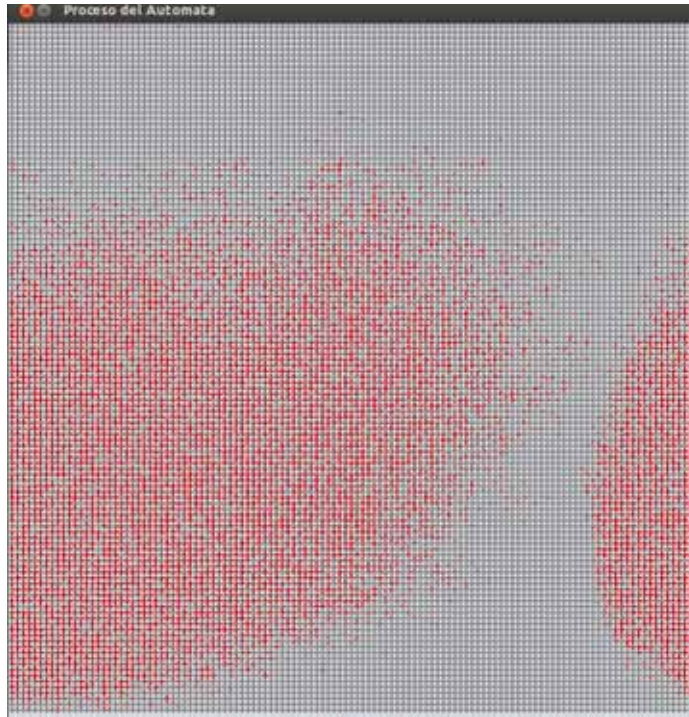


Figura 5.7 Las células cancerígenas se dispersan cada vez más si se aumenta el número de canales de reposo en el sistema en un tiempo considerablemente menor.

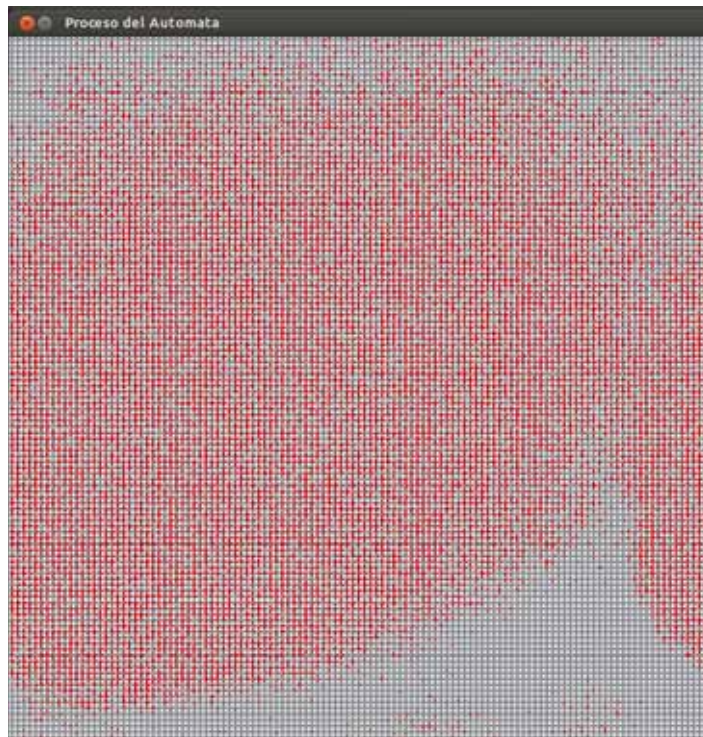
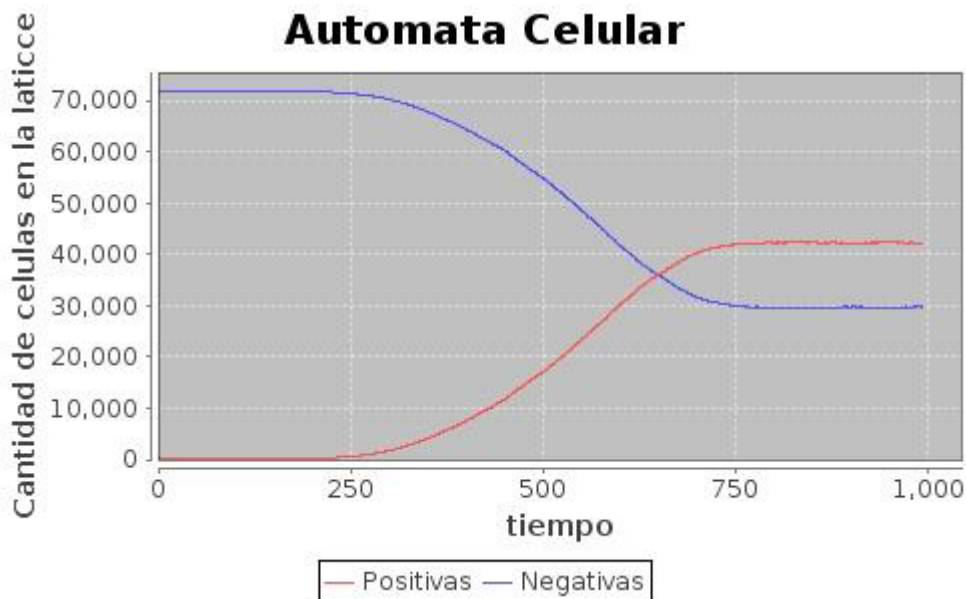
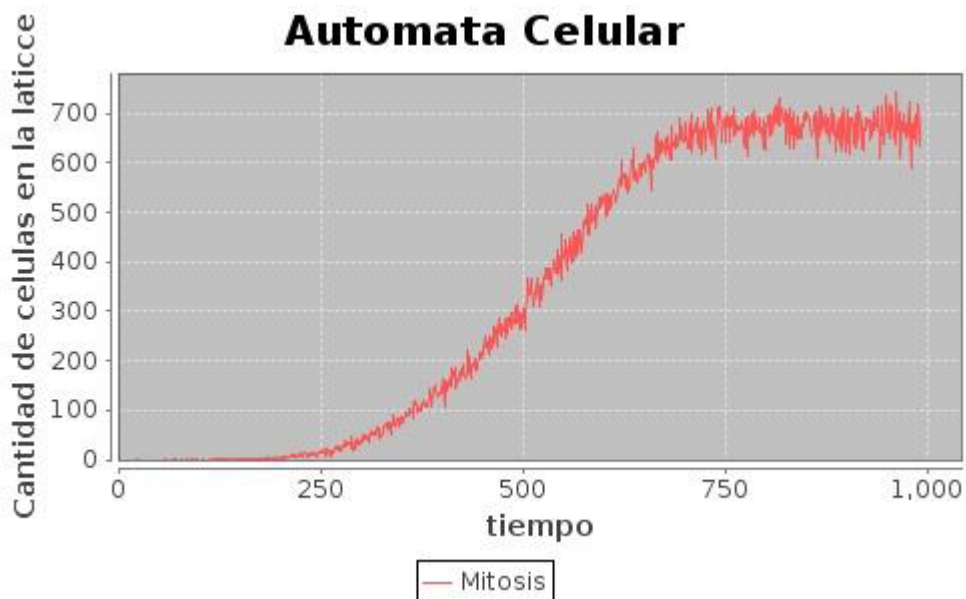


Figura 5.8 muestra la dispersión ya por toda la lattice de las células cancerígenas

Con los parámetros de la tabla 5.1 se obtuvieron las siguientes graficas



Grafica 5.7 se muestra el crecimiento de las células positivas y negativas durante la simulación con un tiempo menor para la invasión



Grafica 5.8 muestra el proceso de apoptosis de las células positivas durante el tiempo de ejecución del sistema

De acuerdo a los resultados arrojados en las simulaciones realizadas se analizó lo siguiente

La influencia del número de canales de reposo en el sistema; ya que estos provocan la dispersión de la células afectadas en el sistema, mientras más canales en reposo haya existen menos puntos de concentración, las células positivas se extienden más, pero la invasión es en menos tiempo

CAPITULO 6

Conclusiones

En el presente trabajo se cumplió con lo planteado en el objetivo general y los objetivos específicos

Se muestra el desarrollo y simulación de un Autómata Celular dos dimensional

El autómata celular de retícula de gases permite estudiar fenómenos biológicos de manera más simple y precisa con interacciones locales, es por ello que lo seleccionamos para nuestro objeto de estudio.

Su propósito ha sido mediante reglas desarrolladas y el algoritmo de movimiento propuesto, mostrar la dinámica de reproducción e invasión de neoplasias malignas resultante de la simulación a partir de una célula en el centro de su retícula.

Este tipo de sistemas nos permite dejar de experimentar con la vida de aquellas personas que sufren esta enfermedad.

Los resultados obtenidos en las simulaciones realizadas para el autómata celular propuesto, nos permitieron observar, que el modelo representa comportamientos caóticos, sin embargo a pesar de ellos, los resultados obtenidos muestran que a mayor capacidad de proliferación el tiempo de invasión será más corto. Otro resultado importante reflejado en los resultados arrojados por el modelo, es que la migración celular, está fuertemente implicada en el aumento de la población y la formación de nuevas colonias de células cancerígenas, independientemente a la capacidad de proliferación de cada célula

Se implementó un sistema de simulación bajo el lenguaje de programación JAVA con la finalidad de que el modelo esté disponible para diferentes sistemas operativos y no dependa de una arquitectura en específica.

Se diseñó software basado en computación biológica en el cual puede interactuar con los siguientes parámetros

Parámetros	Valores
porcentaje de mitosis	de .001 a 1
porcentaje de apoptosis	de .001 a 1
tamaño de lattice (L)	de 1 a 120
cantidad de células cancerígenas	de 1 a 72000
numero de canales en reposo por nodo	0 a 5

Trabajos futuros.

Dada la naturaleza del modelo propuesto, en especial la de sus operadores de reorientación, reacción y cambio fenotípico existe una gran capacidad de ser rediseñados para operar en un entorno paralelo o distribuido. Por lo que, en caso de incrementar el periodo de simulación, esta será una alternativa a considerar.

Realizar la investigación y análisis en la literatura a fin de identificar y en su caso, incrementar el conjunto de mediciones y / o variables manejadas en el modelo, para buscar que el sistema se ajuste a las nuevas verdades biológicas descubiertas.

Llevar a cabo la simulación en un clúster de computadoras con la previa paralización del algoritmo del modelo.

RERERENCIAS

- [1] A. Giese, R. Bjerkvig and M. Westphal, " Cost of migration: invasion of malignant gliomas and implications for treatment" *Journal of Clinical Oncology*, vol. 21, no. 10, app. 1624– 1636, Abril, 2003.
- [2] A. Wolf Gladrow, " Lattice-Gas Cellular Automata and Lattice Boltzmann Models An Introduction" Alfred Wegener Institute (Alemania) (2005) 124.
- [3] GIESE,A., BJERKVIG,R., BERENS,M. & WESTPHAL,M. (2003) Cost of migration: invasion of malignant gliomas and implications for treatment. *Journal of Clinical Oncology*.,21, 1624-1636
- [4] GODLEWSKI,J., NOWICKI,M. O., BRONISZ,A., NUOVO,G., PALATINI,J., LAY,M. D., BROCKLYN,J. V.,OSTROWSKI,M. C. & CHIOCCA,E. A. (2010) Microrna-451 regulates Ikb1/ampk signaling and allows adaptation to metabolic stress in glioma cells. *Mol. Cell*,37, 620-632
- [5] H. Enderling, L. Hlatky, P. Hahnfeldt, " Migration rules: tumours are conglomerates of selfmetastases" *Br. J. Cancer* , vol. 100, no. 12, app. 1917–1925, Mayo, 2009.
- [6] A. Anderson, A. Weaver , P. Cummings, V. Quaranta , "Tumor morphology and phenotypic evolution driven by selective pressure from the microenvironment," *Cell*, vol. 127, no. 5, app. 905–915, Diciembre, 2006.

APENDICE

Clase Borrarfichero

```
package cancer;
import java.io.*;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.WindowConstants;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;

public class Gmuerenxy extends JFrame implements ActionListener {

    public int minX=1;
    public int maxX=100;

    JFreeChart chart;

    public Gmuerenxy(String title){
        super(title);

        XYSeries series = new XYSeries("Mitosis");

        File archivo = null;

        FileReader fr = null;

        try {
            archivo = new File
("C:\\Users\\a2\\Documents\\NetBeansProjects\\cancer\\nace.txt");
            String linea,linea1;
```

```

fr = new FileReader (archivo);

BufferedReader br = new BufferedReader(fr);

double i=0,j=0;
double entero,entero1;
System.out.println("Nacen");
while((linea=br.readLine())!=null){

entero = Integer.parseInt(linea);
    series.add(i,entero);
    System.out.println(linea);
i++;}

}

catch(IOException e){
    System.out.println(e);
}finally{
    try{
        if( fr != null){
            fr.close();
            //fr1.close();
        }
    }catch (IOException e){
        System.out.println(e);
    }
}

}

XYSeriesCollection dataset = new XYSeriesCollection();
dataset.addSeries(series);

chart = ChartFactory.createXYLineChart(
    "Mitosis por etapa de tiempo",           // Title
    "tiempo",                               // x-axis Label
    "Cantidad de celulas ",                 // y-axis Label
    dataset,                                 // Dataset
    PlotOrientation.VERTICAL,               // Plot Orientation
    true,                                   // Show Legend
    true,                                   // Use tooltips
    false                                   // Configure chart to generate URLs?
);

```

```

ChartPanel panel = new ChartPanel(chart);

JButton saveButton = new JButton("Save...");
saveButton.addActionListener(this);

getContentPane().setLayout(new FlowLayout());
getContentPane().add(panel);
getContentPane().add(saveButton);

setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
pack();
setVisible(true);
}

```

@Override

```

public void actionPerformed(ActionEvent e) {
    JFileChooser fc=new JFileChooser("C:/");
    fc.setCurrentDirectory(new File("C:/"));
    fc.setSelectedFile(new File("C:/g1.jpg"));
    int ret=fc.showSaveDialog(null);
    if(ret==fc.APPROVE_OPTION)
    {
        try {
            System.out.print("Saving... "+
fc.getSelectedFile().getAbsolutePath()+"\n");
            ChartUtilities.saveChartAsJPEG(fc.getSelectedFile(), chart, 500, 300);
        } catch (IOException IOe) {
            System.err.println("Problem occurred creating chart.");
            IOe.printStackTrace();
        }
    }
}
}
}

```

Class Cancer

```

package cancer;
import java.util.*;
import java.util.Random;
import java.util.Hashtable;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class Cancer {
int vivas;
int muertas;
int negativas=0;
int positivas=0;

public Cancer () {

        }

public void llenar(int n,Hashtable l){

    negativas=0;
    positivas=0;
    int np=0;

    int col=0;
    int fil= 0;
    for(int k=0;k<n;k++){ //fila
        for(int i=0;i<n;i++){//columna
            for(int j=0;j<5;j++){ //canal
                Random r = new Random ();// numero aleatorio
                //que sean enteros
                int x = r.nextInt(1000);
                if(x<999){x=0 ;
                    l.put((j+fil+col),x );
                    negativas++;
                }
                else{ x=1;

```



```

        l.put((j+fil+col),x );

        positivas++;

    }

    }//cierre canal

    col=col+10;
    }//cierre columna
    col=0;
    fil=fil+10000;
    }//cierre fila

    String posi = Integer.toString(positivas);
    String neg = Integer.toString(negativas);
    try
    {
        try (BufferedWriter out = new BufferedWriter(new FileWriter("positiva.txt",
true))) {
            out.write(posi);
            out.newLine();
            out.close();
        }
        } catch (IOException e1){}
    try
    {
        try (BufferedWriter out = new BufferedWriter(new FileWriter("negativa.txt",
true))) {
            out.write(neg);
            out.newLine();
            out.close();
        }
        } catch (IOException e2){}

}
public void mostrar(Hashtable h){

```

```

        Enumeration e = h.keys();
        Object obj;
        while (e.hasMoreElements()) {
            obj = e.nextElement();
            System.out.println(" key "+ obj +": "+ h.get(obj));
        }
        System.out.println("existen "+h.size()+" elementos en la matriz");
    }
    public void rotar(Hashtable h,Hashtable N, int n){
        vivas=0;
        muertas=0;
        int[] intercambio = new int [5];
        int col=0;
        int fil= 0;
        int unos,unoi;

        for(int k=0;k<n;k++){ //fila
            for(int i=0;i<n;i++){//columna
                unos=0;
                unoi=0;
                for(int j=0;j<5;j++){ //canal
                    intercambio[j]=0; //poner en ceros el vector
                    Object clave= h.get(j+fil+col );

                    intercambio[j]=(int) clave;//asignamos celula a vector
                    if(intercambio[j]==1)unos++; //verificamos cuantas celulas positivas
                    hay en el nodo
                    if(j>0 && intercambio[j]==1)unoi++;
                }//cierre canal

                int colectorv=0;
                for(int l=0;l<5;l++){//itercambio por medio de random

                    Random rnd = new Random();
                    int rr = (int)(rnd.nextDouble() * 4 + 1);
                    colectorv=intercambio[rr];
                    intercambio[rr]=intercambio[l];
                    intercambio[l]=colectorv;
                }//cierrre intercambio

                vida( intercambio, unos);//crear vida o dar muerte
            }
        }
    }

```



```

if(v[0]==1 && unos<5){//condicion para vida
    for(int i=0;i<5;i++){
        if(v[i]==0){
            recorre[a]=i;
            a++;
        }
    }
}
int rr = (int)(rnd.nextDouble() * 10 + 1);
if(rr<3){

    int rrr = (int)(rnd.nextDouble() * a);
    b=recorre[rrr];
    v[b]=1;
    vivas++;
}

} //fin condicion vida

if( unos>4){//inicio muerte de cancer
    int r2=(int)(rnd.nextDouble() * 10 + 1);
    if(r2<7){

        int muerte= (int)(rnd.nextDouble() * 4 + 1);
        if(v[muerte]==0 && muerte==4)v[muerte-1]=0;
        v[muerte]=0;
        muertas++;
    }
} //fin muerte

}

```

```

public void ajustar(Hashtable h){
    negativas=0;
    positivas=0;
    //mostrar tabla de la celula
    Enumeration e = h.keys();
    Object obj;
    while (e.hasMoreElements()) {
        obj = e.nextElement();
        if(h.get(obj)==10)h.put(obj,1);
        if(h.get(obj)==1)positivas++;
        if(h.get(obj)==-1)h.put(obj,0);
        if(h.get(obj)==0)negativas++;
    }
}

```

```

        }
        String posi1 = Integer.toString(positivas);
        String neg1 = Integer.toString(negativas);
        try
        {
            try (BufferedWriter out = new BufferedWriter(new FileWriter("positiva.txt",
true))) {
                out.write(posi1);
                out.newLine();
                out.close();
            }
            } catch (IOException e3){}
        try
        {
            try (BufferedWriter out = new BufferedWriter(new FileWriter("negativa.txt",
true))) {
                out.write(neg1);
                out.newLine();
                out.close();
            }
            } catch (IOException e4){}
        }
    }
}

```

Clase CrearFichero

```

package cancer;

import java.io.*;

public class CrearFichero {

    public void creaf() {

        File positivo = new File( "positiva.txt");
        File negativo = new File("negativa.txt");
        File nacen = new File("nace.txt");
        File mueren = new File("muere.txt");
    }
}

```

```

try {
    if (positivo.createNewFile())
        System.out.println("El fichero se ha creado correctamente");
    else
        System.out.println("No ha podido ser creado el fichero");
} catch (IOException ioe) {
    ioe.printStackTrace();
}
try {
    if (negativo.createNewFile())
        System.out.println("El fichero se ha creado correctamente");
    else
        System.out.println("No ha podido ser creado el fichero");
} catch (IOException ioe) {
    ioe.printStackTrace();
}
try {
    if (nacem.createNewFile())
        System.out.println("El fichero se ha creado correctamente");
    else
        System.out.println("No ha podido ser creado el fichero");
} catch (IOException ioe) {
    ioe.printStackTrace();
}
try {
    if (mueren.createNewFile())
        System.out.println("El fichero se ha creado correctamente");
    else
        System.out.println("No ha podido ser creado el fichero");
} catch (IOException ioe) {
    ioe.printStackTrace();
}
}
}

```

Clase Glaticce

```
package cancer;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.util.Enumeration;
import java.util.Hashtable;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Glaticce extends JPanel{

    private Hashtable laticce;
    Hashtable<Integer,Integer> laticce2;
    Cancer C= new Cancer();
    int bandera=0;
    propagar P= new propagar();
    int n=120;//tamaño de la laticce
    public Glaticce(){

        laticce = new Hashtable();
        laticce2= new Hashtable();
        C.llenar(n,laticce);}
    @Override
    public void paintComponent(Graphics g) {

        Graphics2D graphics2D = (Graphics2D)g;
        int width = getWidth();
        int height = getHeight();
        int f=0,c,s,desp,i,j,k,t=2;
        int clave;

        //recorro matriz base
        Enumeration ee = laticce.keys();
        Object obj;
        int vector[]={0,0,0};
        int ss;
        //System.out.println("EN EL GRAFICO");
        while (ee.hasMoreElements()) { //recorro la tabla hash
            obj = ee.nextElement();
            clave= (Integer) obj;
```

```

P.tok(clave,vector);//DIVIR CLAVE
Object cl=latice.get(clave);
ss=(Integer)cl;//valor de s
i=vector[0];
j=vector[1];
k=vector[2];

if(k==0){
    if(ss==0){
        g.setColor(Color.gray);
        g.fill3DRect((((i*3)+1)*t),(((j*3)+1)*t),t,t,true);
    }
    if(ss==1){
        g.setColor(Color.red);
        g.fill3DRect((((i*3)+1)*t),(((j*3)+1)*t),t,t,true);
    }
}
if(k==1){
    if(ss==0){
        g.setColor(Color.gray);
        g.fill3DRect((((i*3)+1)*t),((j*3)*t),t,t,true);
    }
    if(ss==1){
        g.setColor(Color.red);
        g.fill3DRect((((i*3)+1)*t),((j*3)*t),t,t,true);
    }
}
if(k==2){
    if(ss==0){
        g.setColor(Color.gray);
        g.fill3DRect((((i*3)+2)*t),(((j*3)+1)*t),t,t,true);
    }
    if(ss==1){
        g.setColor(Color.red);
        g.fill3DRect((((i*3)+2)*t),(((j*3)+1)*t),t,t,true);
    }
}
if(k==3){
    if(ss==0){
        g.setColor(Color.gray);
        g.fill3DRect((((i*3)+1)*t),(((j*3)+2)*t),t,t,true);
    }
    if(ss==1){

```



```

        g.setColor(Color.red);
        g.fill3DRect((((i*3)+1)*t),(((j*3)+2)*t),t,t,true);
    }
}
if(k==4){
    if(ss==0){
        g.setColor(Color.gray);
        g.fill3DRect(((i*3)*t),(((j*3)+1)*t),t,t,true);
    }
    if(ss==1){
        g.setColor(Color.gray);
        g.fill3DRect(((i*3)*t),(((j*3)+1)*t),t,t,true);
    }
}
}

repaint();

C.rotar(laticce,laticce2,n); // este me genera pedo

P.propaga(laticce2, n);

C.ajustar(laticce2);

laticce.putAll(laticce2);//copiar 2 en 1

bandera++;

}

```

```

public void ron() {

    BorrarFichero B= new BorrarFichero();
    CrearFichero Cr= new CrearFichero();
    B.Borra();
    Cr.creaf();
    JFrame f = new JFrame();
    f.setSize(950, 900);
    f.setTitle("Proceso del Automata");
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```

```

        f.setContentPane(new Glasticce());
        f.setVisible(true);
        f.setResizable(false);
    }
}

```

Clase Gmuerenxy

```

package cancer;
import java.io.*;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.WindowConstants;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;

public class Gmuerenxy extends JFrame implements ActionListener {

    public int minX=1;
    public int maxX=100;

    JFreeChart chart;

    public Gmuerenxy(String title){
        super(title);

        XYSeries series = new XYSeries("Mitosis");

        File archivo = null;

```

```

        FileReader fr = null;

    try {
        archivo = new File
("C:\\Users\\a2\\Documents\\NetBeansProjects\\cancer\\nace.txt");
        String linea,linea1;
        fr = new FileReader (archivo);

        BufferedReader br = new BufferedReader(fr);

        double i=0,j=0;
        double entero,entero1;
        System.out.println("Nacen");
        while((linea=br.readLine())!=null){

            entero = Integer.parseInt(linea);
            series.add(i,entero);
            System.out.println(linea);
            i++;}

        }

        catch(IOException e){
            System.out.println(e);
            }finally{
                try{
                    if( fr != null){
                        fr.close();
                        //fr1.close();
                    }
                }catch (IOException e){
                    System.out.println(e);
                }
            }

        XYSeriesCollection dataset = new XYSeriesCollection();
        dataset.addSeries(series);

        chart = ChartFactory.createXYLineChart(
            "Mitosis por etapa de tiempo",           // Title
            "tiempo",                                // x-axis Label

```

```

        "Cantidad de células ",          // y-axis Label
        dataset,                        // Dataset
        PlotOrientation.VERTICAL,       // Plot Orientation
        true,                            // Show Legend
        true,                            // Use tooltips
        false                            // Configure chart to generate URLs?
    );

    ChartPanel panel = new ChartPanel(chart);

    JButton saveButton = new JButton("Save...");
    saveButton.addActionListener(this);

    getContentPane().setLayout(new FlowLayout());
    getContentPane().add(panel);
    getContentPane().add(saveButton);

    setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    pack();
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    JFileChooser fc=new JFileChooser("C:/");
    fc.setCurrentDirectory(new File("C:/"));
    fc.setSelectedFile(new File("C:/g1.jpg"));
    int ret=fc.showSaveDialog(null);
    if(ret==fc.APPROVE_OPTION)
    {
        try {
            System.out.print("Saving... "+
fc.getSelectedFile().getAbsolutePath()+"\n");
            ChartUtilities.saveChartAsJPEG(fc.getSelectedFile(), chart, 500, 300);
        } catch (IOException IOe) {
            System.err.println("Problem occurred creating chart.");
            IOe.printStackTrace();
        }
    }
}
}
}

```

```
}
```

Clase Gnacenxy

```
package cancer;
import java.io.*;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.WindowConstants;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
```

```
public class Gnacenxy extends JFrame implements ActionListener {
    // define X range
    public int minX=1;
    public int maxX=100;

    JFreeChart chart;

    public Gnacenxy(String title){
        super(title);
        // Create a simple XY chart
        XYSeries series = new XYSeries("Positivas");
        XYSeries series1 = new XYSeries("Negativas");

        //abri primer archivo
        File archivo = null;
```

```

File archivo1=null;
    FileReader fr = null;
    FileReader fr1 = null;
try {
        archivo = new File
("C:\\Users\\a2\\Documents\\NetBeansProjects\\cancer\\positiva.txt");
        archivo1 = new File
("C:\\Users\\a2\\Documents\\NetBeansProjects\\cancer\\negativa.txt");
    String linea,linea1;
    fr = new FileReader (archivo);
    fr1=new FileReader (archivo1);
    BufferedReader br = new BufferedReader(fr);
    BufferedReader br1 = new BufferedReader(fr1);
    double i=0,j=0;
    double entero,entero1;
    while((linea=br.readLine())!=null){

entero = Integer.parseInt(linea);
        series.add(i,entero);
        System.out.println(linea);
i++;}
        System.out.println("muertas");
while((linea1=br1.readLine())!=null){

entero1 = Integer.parseInt(linea1);
        series1.add(j,entero1);
        System.out.println(linea1);
j++;}

}

catch(IOException e){
        System.out.println(e);
    }finally{
        try{
            if( fr != null){
                fr.close();
                fr1.close();
            }
        }catch (IOException e){
            System.out.println(e);
        }
    }
}

```

```

// Add the series to your data set
XYSeriesCollection dataset = new XYSeriesCollection();
dataset.addSeries(series);
dataset.addSeries(series1);
// Generate the graph
chart = ChartFactory.createXYLineChart(
    "Automata Celular",           // Title
    "tiempo",                    // x-axis Label
    "Cantidad de celulas en la laticce", // y-axis Label
    dataset,                     // Dataset
    PlotOrientation.VERTICAL,    // Plot Orientation
    true,                        // Show Legend
    true,                        // Use tooltips
    false                         // Configure chart to generate URLs?
);

ChartPanel panel = new ChartPanel(chart);

JButton saveButton = new JButton("Save...");
saveButton.addActionListener(this);

getContentPane().setLayout(new FlowLayout());
getContentPane().add(panel);
getContentPane().add(saveButton);

setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
pack();
setVisible(true);
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    JFileChooser fc=new JFileChooser("C:/");
    fc.setCurrentDirectory(new File("C:/"));
}

```

```

fc.setSelectedFile(new File("C:/g1.jpg"));
int ret=fc.showSaveDialog(null);
if(ret==fc.APPROVE_OPTION)
{
    try {
        System.out.print("Saving... "+
fc.getSelectedFile().getAbsolutePath()+"\n");
        ChartUtilities.saveChartAsJPEG(fc.getSelectedFile(), chart, 500, 300);
    } catch (IOException IOe) {
        System.err.println("Problem occurred creating chart.");
        IOe.printStackTrace();
    }
}
}
}
}

```

Clase Inicio

```

package cancer;

import javax.swing.JOptionPane;

public class Inicio extends javax.swing.JFrame {

    public Inicio() {

        initComponents();
        setLocationRelativeTo(null);
        setResizable(false);
        setTitle("Automata Celular de Reticula de Gases");
    }

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

        jPopupMenu1 = new javax.swing.JPopupMenu();

```



```
jPopupMenu2 = new javax.swing.JPopupMenu();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
InicioA = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
SalirA = new javax.swing.JMenuItem();
```

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

```
jMenu1.setText("Menu");
```

```
InicioA.setText("Iniciar");
InicioA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        InicioAActionPerformed(evt);
    }
});
jMenu1.add(InicioA);
```

```
jMenuItem2.setText("Grafica1");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem2);
```

```
jMenuItem1.setText("Grafica2");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem1);
```

```
jMenuBar1.add(jMenu1);
```

```
jMenu2.setText("Salida");
```

```
SalirA.setText("Salir");
```

```

    SalirA.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            SalirAActionPerformed(evt);
        }
    });
    jMenu2.add(SalirA);

    jMenuBar1.add(jMenu2);

    setJMenuBar(jMenuBar1);

    javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 400, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 283, Short.MAX_VALUE)
                .addContainerGap())
            .addComponent(pack());
    }

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

        Glaticce obj1= new Glaticce();
        obj1.ron();
    }

    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
        Gnacenxy obj= new Gnacenxy("Celulas en lattice");
        obj.setVisible(true);

    // TODO add your handling code here:
    }

    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        Gmuerenxy obj= new Gmuerenxy("Proceso de Vida o Muerte");

```

```

    obj.setVisible(true);
}

private void SalirAActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    //<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(Inicio.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

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

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

```

```

    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Inicio().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
public javax.swing.JMenuItem InicioA;
private javax.swing.JMenuItem SalirA;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPopupMenu jPopupMenu1;
private javax.swing.JPopupMenu jPopupMenu2;
// End of variables declaration
}

```

Clase Propagar

```

package cancer;

import java.util.*;
import java.util.Hashtable;

public class propagar {
    int nexts;
    public void propaga(Hashtable N, int n){
        int clave;

        int vector[]={0,0,0};
        propagar P = new propagar(); //recorrer la celula
        Enumeration e = N.keys();
        Object obj;
    }
}

```

```

while (e.hasMoreElements()) {
    int s;

    obj = e.nextElement();
    clave= (int) obj;

    P.tok(clave,vector);//DIVIR CLAVE
    Object cl=N.get(clave);
    s=(Integer)cl;//valor de s

    Random rnd = new Random();
    if(s==1 && vector[2]!=0){//si la celula es positiva

        int m1 = (int)(rnd.nextDouble() * 3 + 0);
        if(m1==0){nexts=vector[2]+(vector[1]*10)+(vector[0]*10000);
            }
        if(m1==1){
            if(vector[2]==1 &&
vector[0]==0)nexts=vector[2]+(vector[1]*10)+((n-1)*10000);
            if(vector[2]==2 && vector[1]==(n-
1))nexts=vector[2]+(vector[1]*0)+(vector[0]*10000);
            if(vector[2]==3 && vector[0]==(n-
1))nexts=vector[2]+(vector[1]*10)+(vector[0]*0);
            if(vector[2]==4 && vector[1]==0)nexts=vector[2]+(10*(n-
1))+vector[0]*10000);

                if(vector[2]==1 &&
vector[0]!=0)nexts=vector[2]+(vector[1]*10)+((vector[0]-1)*10000);//sube una
fila
                if(vector[2]==2 && vector[1]!=(n-1))nexts=
vector[2]+((vector[1]+1)*10)+(vector[0]*10000);
                if(vector[2]==3 && vector[0]!=(n-
1))nexts=vector[2]+(vector[1]*10)+((vector[0]+1)*10000);
                if(vector[2]==4 && vector[1]!=0)nexts= vector[2]+((vector[1]-
1)*10)+(vector[0]*10000);
            }
        if(m1==2){
            if(vector[2]==1 &&
vector[0]==0)nexts=vector[2]+(vector[1]*10)+((n-2)*10000);
            if(vector[2]==2 && vector[1]==(n-
1))nexts=vector[2]+(10)+(vector[0]*10000);
            if(vector[2]==3 && vector[0]==(n-
1))nexts=vector[2]+(vector[1]*10)+(10000);

```

```

        if(vector[2]==4 && vector[1]==0)nexts=vector[2]+(10*(n-
2))+(vector[0]*10000);

        if(vector[2]==1 &&
vector[0]==1)nexts=vector[2]+(vector[1]*10)+((n-1)*10000);
        if(vector[2]==2 && vector[1]==(n-
2))nexts=vector[2]+(vector[1]*0)+(vector[0]*10000);
        if(vector[2]==3 && vector[0]==(n-
2))nexts=vector[2]+(vector[1]*10)+(vector[0]*0);
        if(vector[2]==4 && vector[1]==1)nexts=vector[2]+(vector[1]*10*(n-
1))+((n-1)*0);

        if(vector[2]==1 &&
vector[0]>1)nexts=vector[2]+(vector[1]*10)+((vector[0]-2)*10000);//sube una
fila
        if(vector[2]==2 && vector[1]<(n-2))nexts=
vector[2]+((vector[1]+2)*10)+(vector[0]*10000);
        if(vector[2]==3 && vector[0]<(n-
2))nexts=vector[2]+(vector[1]*10)+((vector[0]+2)*10000);
        if(vector[2]==4 && vector[1]>1)nexts= vector[2]+((vector[1]-
2)*10)+(vector[0]*10000);
    }

    if(m1==3){
        if(vector[2]==1 &&
vector[0]==0)nexts=vector[2]+(vector[1]*10)+((n-3)*10000);
        if(vector[2]==2 && vector[1]==(n-
1))nexts=vector[2]+(20)+(vector[0]*10000);
        if(vector[2]==3 && vector[0]==(n-
1))nexts=vector[2]+(vector[1]*10)+(20000);
        if(vector[2]==4 && vector[1]==0)nexts=vector[2]+(10*(n-
3))+vector[0]*10000);

        if(vector[2]==1 &&
vector[0]==1)nexts=vector[2]+(vector[1]*10)+((n-2)*10000);
        if(vector[2]==2 && vector[1]==(n-
2))nexts=vector[2]+(10)+(vector[0]*10000);
        if(vector[2]==3 && vector[0]==(n-
2))nexts=vector[2]+(vector[1]*10)+(10000);
        if(vector[2]==4 && vector[1]==1)nexts=vector[2]+(10*(n-
2))+vector[0]*10000);

```

```

        if(vector[2]==1 &&
vector[0]==2)nexts=vector[2]+(vector[1]*10)+((n-1)*10000);
        if(vector[2]==2 && vector[1]==(n-
3))nexts=vector[2]+(vector[1]*0)+(vector[0]*10000);
        if(vector[2]==3 && vector[0]==(n-
3))nexts=vector[2]+(vector[1]*10)+(vector[0]*0);
        if(vector[2]==4 && vector[1]==2)nexts=vector[2]+(vector[1]*10*(n-
1))+((n-1)*0);

        if(vector[2]==1 &&
vector[0]>2)nexts=vector[2]+(vector[1]*10)+((vector[0]-3)*10000);//sube una
fila
        if(vector[2]==2 && vector[1]<(n-3))nexts=
vector[2]+((vector[1]+3)*10)+(vector[0]*10000);
        if(vector[2]==3 && vector[0]<(n-
3))nexts=vector[2]+(vector[1]*10)+((vector[0]+3)*10000);
        if(vector[2]==4 && vector[1]>2)nexts= vector[2]+((vector[1]-
3)*10)+(vector[0]*10000);

    }

    if(N.get(nexts)== 0){//si en la celula a donde brinca hay un 0
        N.put(nexts,10);
        N.put(clave, -1);
    }

    } //cierrrre celula positiva
}

}

public void tok(int clave,int vectorp[]) {

    int uno;
    String cadena = "";

    int n;
    int tam;
    int f=0;
    int c=0;
    int s=0;
    n= clave;
    cadena = String.valueOf(n);

```

```

cadena= Integer.toString(n);
tam=cadena.length();
if (tam==1){
    f=0;
    c=0;
    s = Integer.parseInt(cadena);
    vectorp[0]=f;
    vectorp[1]=c;
    vectorp[2]=s;
}
if(tam<5 && tam>1){
    f=0;
    String subcadenac= cadena.substring(0,(tam-1));
    c= Integer.parseInt(subcadenac);
    //System.out.println("paso por aqui"+c);
    Character ch= new Character (cadena.charAt(tam-1)) ;
    String subcadenas =ch.toString();
    s= Integer.parseInt(subcadenas );
    vectorp[0]=f;
    vectorp[1]=c;
    vectorp[2]=s;
}
if(tam>4 && tam<8){
    String subcadenaf= cadena.substring(0,(tam-4));
    f= Integer.parseInt(subcadenaf);

    String subcadenac= cadena.substring((tam-4),(tam-1));
    c= Integer.parseInt(subcadenac);
    Character ch= new Character (cadena.charAt(tam-1)) ;
    String subcadenas =ch.toString();
    s= Integer.parseInt(subcadenas );
    vectorp[0]=f;
    vectorp[1]=c;
    vectorp[2]=s;
}
}
}

```