



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

MÁSTER UNIVERSITARIO EN COMPUTACIÓN LÓGICA E  
INTELIGENCIA ARTIFICIAL

ANÁLISIS DE LAS TRANSICIONES DE  
COMPORTAMIENTO EN CRECIMIENTO DE TUMORES  
USANDO UNA SIMULACIÓN CON AUTÓMATA  
CELULAR

Realizado por  
Sergio Rodríguez Calvo  
XXXXXXXX

Supervisado por  
Dr. Miguel Ángel Gutiérrez Naranjo

Departamento  
Ciencias de la Computación e Inteligencia Artificial

Sevilla, noviembre 2017

Realizado por  
Sergio Rodríguez Calvo  
[sergiorodriguezcalvo@gmail.com](mailto:sergiorodriguezcalvo@gmail.com)

# ÍNDICE GENERAL

<b>1. Resumen</b>	<b>7</b>
<b>2. Introducción</b>	<b>9</b>
2.1. Objetivos . . . . .	10
2.2. Alcance . . . . .	10
2.3. El cáncer . . . . .	10
2.4. Los autómatas celulares . . . . .	12
<b>3. Estado del arte</b>	<b>15</b>
3.1. Resultado de la búsqueda de artículos . . . . .	15
3.2. <i>Cellular automaton of idealized brain tumor growth dynamics</i> . . . . .	16
3.3. <i>Analysis of behaviour transitions in tumour growth using a cellular automaton simulation</i> . . . . .	17
<b>4. Modelo de eventos</b>	<b>19</b>
4.1. Objetivos del artículo . . . . .	19
4.2. Las células, genoma y propiedades . . . . .	20
4.3. Parámetros globales . . . . .	21
4.4. Pruebas previas a la mitosis . . . . .	21
4.5. Mitosis . . . . .	22
4.6. Eventos mitóticos . . . . .	23
4.7. Equivalencia temporal . . . . .	23

## ÍNDICE GENERAL

4.8. Bucle principal del modelo de eventos . . . . .	24
<b>5. Implementación del sistema</b>	<b>27</b>
5.1. Tecnologías . . . . .	27
5.2. Arquitectura y diseño . . . . .	28
5.3. Módulos del sistema . . . . .	28
5.3.1. Módulo <i>genome.py</i> . . . . .	28
5.3.2. Módulo <i>cell.py</i> . . . . .	29
5.3.3. Módulo <i>simulation_globals.py</i> . . . . .	30
5.3.4. Módulo <i>experiments.py</i> . . . . .	31
5.3.5. Módulo <i>grid.py</i> . . . . .	32
5.3.6. Módulo <i>automata.py</i> . . . . .	32
5.3.7. Módulo <i>analytics.py</i> . . . . .	33
5.4. Instalación y ejecución . . . . .	33
5.4.1. Prerrequisitos . . . . .	34
5.4.2. Ejecución . . . . .	34
5.4.3. Parámetros de la simulación . . . . .	35
<b>6. Experimentos de los autores</b>	<b>37</b>
6.1. Influencia del parámetro <i>Tasa de mutación base (m)</i> . . . . .	37
6.1.1. Experimento 1: Tasa de mutación base igual a 10.000 . . . . .	38
6.1.2. Experimento 2: Tasa de mutación base igual a 1.000 . . . . .	38
6.1.3. Experimento 3: Tasa de mutación base igual a 100 . . . . .	39
6.2. Influencia del resto de parámetros . . . . .	39
6.3. Influencia de parámetros con rejilla completa de células sanas . . . . .	41
6.4. Relevancia de los marcadores . . . . .	41

6.5. Comportamiento de las transiciones . . . . .	43
<b>7. Resultados propios</b>	<b>45</b>
7.1. Influencia del parámetro <i>Tasa de mutación base (m)</i> . . . . .	45
7.1.1. Experimento 1: Tasa de mutación base igual a 10.000 . . . . .	45
7.1.2. Experimento 2: Tasa de mutación base igual a 1.000 . . . . .	47
7.1.3. Experimento 3: Tasa de mutación base igual a 100 . . . . .	49
<b>8. Conclusión</b>	<b>53</b>
<b>9. Trabajo futuro</b>	<b>55</b>



# ÍNDICE DE FIGURAS

2.1. <i>Conus textile shell</i> , especie de gastrópodo que presenta un autómata celular natural en su concha. . . . .	12
2.2. Tipos de vecindarios en los autómatas celulares. . . . .	13
3.1. triangulación de Delaunay junto a Polígonos de Thiessen o diagramas de Voronoi para modelizar el tamaño de las células. . . . .	17
4.1. Modelo de eventos para simulación del cáncer. . . . .	24
5.1. Código con el bucle principal de la simulación. . . . .	33
5.2. Código para configuración por defecto de una simulación. . . . .	36
6.1. Resultados obtenidos por los autores para el experimento con $m = 10,000$ . . . . .	38
6.2. Resultados obtenidos por los autores para el experimento con $m = 1,000$ . . . . .	39
6.3. Resultados obtenidos por los autores para el experimento con $m = 100$ . . . . .	40
6.4. Resultado de la simulación para comprobar la influencia del resto de parámetros. . . . .	41
6.5. Completar. . . . .	42
6.6. Completar. . . . .	42
6.7. Completar. . . . .	43
6.8. Completar. . . . .	44
7.1. Células sanas frente a células cancerosas como resultado de la simulación para el experimento con $m = 10,000$ . . . . .	46

7.2. Evolución de los marcadores como resultado de la simulación para el experimento con $m = 10,000$ . . . . .	47
7.3. Rejilla resultante de la simulación para el experimento con $m = 10,000$ . . . . .	48
7.4. Células sanas frente a células cancerosas como resultado de la simulación para el experimento con $m = 1,000$ . . . . .	49
7.5. Evolución de los marcadores como resultado de la simulación para el experimento con $m = 1,000$ . . . . .	50
7.6. Rejilla resultante de la simulación para el experimento con $m = 10,000$ . . . . .	51
7.7. Células sanas frente a células cancerosas como resultado de la simulación para el experimento con $m = 100$ . . . . .	51
7.8. Evolución de los marcadores como resultado de la simulación para el experimento con $m = 100$ . . . . .	52
7.9. Rejilla resultante de la simulación para el experimento con $m = 100$ . . . . .	52

## RESUMEN

El cáncer es un nombre genérico que agrupa más de 200 enfermedades<sup>1</sup> que causan proliferación descontrolada de células provocando la aparición de masas anormales. A toda masa anormal se le conoce como neoplasia y, según su invasividad y factor de crecimiento, puede ser maligna (carcinoma) o benigna (adenoma).

Esta enfermedad puede ser vista como un comportamiento emergente, que puede ser explicado a partir de la presencia de ciertos marcadores cancerosos a nivel local.

José Santos y Ángel Monteagudo, autores de una serie de artículos [7] [8] [9] y, entre ellos, el artículo en el que se centra este trabajo [6], proponen el uso de autómatas celulares como técnica para simular dicho comportamiento dada su idoneidad.

Se propone una modelización en la cual se utiliza una rejilla en tres dimensiones donde, sin tener en cuenta el tamaño de la célula, se presenta al comienzo una única célula en el centro de dicha rejilla. Cada célula, cuenta con un genoma asociado que representa la aparición de mutaciones, así como, dos propiedades: tamaño del telómero, y tasa de mutación base.

Todo ello, es utilizado dentro de un modelo de eventos, en el cual, se programan eventos mitóticos en el futuro y en el que se consideran una serie de pruebas previas a la mitosis. Dichas pruebas, modelizan el comportamiento individual de cada célula a la hora de ejecutar la división, así como, todos aquellos factores que pueden derivar en la muerte de la célula y factores provocados por las mutaciones adquiridas.

Finalmente, se realizan una serie de simulaciones, alterando los parámetros de configuración del autómata, con el objetivo de estudiar el nuevo comportamiento, así como, la incidencia de determinadas terapias en el comportamiento global del sistema.

<sup>1</sup><https://www.aacr.org/Pages/what-is-cancer.aspx>



# INTRODUCCIÓN

Desde el punto de vista de la biología computacional el cáncer puede ser visto como un sistema ecológico y de comportamiento emergente. Lo cual, hace apropiado el uso de autómatas celulares para simular su comportamiento. Es por ello que, cada vez más se opta por el uso de simulaciones por ordenador como la que se presenta en este trabajo ayudando, desde el estudio y comprensión del cáncer, hasta el estudio y desarrollo de nuevas terapias contra él.

El cáncer es una enfermedad cada vez más común debido a sus muy diversos factores que propician su aparición, esto es, mayor esperanza de vida de la población y, por tanto, individuos cuyas células han sufrido más mutaciones, factores ambientales y/o formas de vida poco saludables, mayor estrés de la población, y hasta exposición a fuentes de radiación artificiales.

A pesar de todo, el cuerpo tiene una serie de mecanismos para prevenir la aparición y la proliferación de esta enfermedad. Estos, son muchos y muy diversos, así como, los mecanismos propios de las células que intervienen en este proceso. En esta simulación, por tanto, se pretenden tener en cuenta el enfoque de los autores del trabajo original, eligiendo sólo un subconjunto de todas estas características y comportamientos.

La elección de un autómata celular como enfoque para modelizar el comportamiento de esta enfermedad proviene por la relativa facilidad de obtener comportamientos complejos a partir de unas reglas relativamente sencillas a nivel local en cada célula.

Este trabajo, como se ha comentado previamente, pretende reproducir el trabajo de José Santos y Ángel Monteagudo [6], uno de los trabajos [7] [8] [9] que dichos autores han realizado siguiendo la idea de realizar simulaciones sobre esta enfermedad y, por ello, aquí se realiza una nueva implementación con el fin de reproducir los resultados obtenidos en el citado artículo.

La implementación consiste en seguir un modelo de eventos, es decir, se pretende simular el proceso que siguen las células para su división, que consiste en, la realización de una serie de pruebas para comprobar si la célula debe reproducirse, la reali-

zación de la propia división, y la programación de un nuevo evento de división en el futuro. A lo largo de la ejecución del programa en cada iteración puede haber ninguna, una o varias células que deben seguir el proceso que se acaba de describir brevemente.

Finalmente, para estudiar el comportamiento del cáncer se realizan diversas ejecuciones utilizando diferentes parámetros y configuraciones, y observando en cada caso qué factor es el que provoca el cambio en el comportamiento global.

### 2.1 OBJETIVOS

En este trabajo, se esperan alcanzar los siguientes objetivos:

- Encontrar artículos científicos de calidad sobre esta temática.
- Extraer información del artículo (o los artículos).
- Elegir un artículo y hacer una implementación propia.
- Reproducir los resultados del artículo.
- Comparar los resultados obtenidos con los del artículo.

### 2.2 ALCANCE

El presente trabajo tiene como finalidad reproducir el trabajo de José Santos y Ángel Monteagudo [6], realizando una implementación propia del sistema que se describe en su artículo.

No obstante, el comportamiento que se pretende simular cuenta con una serie de pasos en los cuales el comportamiento dependerá de ciertos sorteos con una probabilidad dada. Esta es una limitación a tener en cuenta a la hora de obtener los mismos resultados que los autores de dicho trabajo.

### 2.3 EL CÁNCER

El cáncer es el nombre genérico usado para referirse a más de 200 enfermedades cuyo resultado es la proliferación descontrolada de las células que se ven afectadas.

Aunque no era común encontrar signos de cáncer en los seres vivos en la antigüedad debido a una menor esperanza media de vida, lo cierto es que es inherente al reino animal.

El origen etimológico del nombre proviene de la antigua grecia, donde un médico llamado Hipócrates utilizaba los términos *carcino* y *carcinoma* para referirse a dicha enfermedad, y las cuales hacían referencia al cangrejo debido a la forma en la que se proyectaban los cánceres.

El cáncer se clasifica como:

- Una **enfermedad genética**, debido a que es causada por un cambio en el material genético (ADN) de las células.
- Una **enfermedad multigénica**, ya que, afecta a diferentes genes.
- Una **enfermedad multifactorial**, es decir, tiene causas muy diversas.
- Una **enfermedad multiorgánica**, por tanto, afecta a diferentes tejidos y órganos.

Se conoce como neoplasia a toda masa anormal que tenga lugar en el cuerpo. Esto es, la división descontrolada de la célula y el aumento de tamaño de cada una de ellas, y ocurre por un problema en el ADN de la célula, la cual, contiene la información acerca de cuáles son las funciones de la misma.

Al producirse en todo tipo de tejidos la evolución y el tratamiento de esta enfermedad varía. El factor determinante entre que este tipo de masas anormales sea benigna (adenoma) o maligna (carcinoma) es el factor de crecimiento y la invasividad de tejido adyacente.

Presenta diferentes fases, partiendo de un crecimiento anormal en tamaño, en la cual, cada vez necesitará más nutrientes. Esta necesidad de nutrientes hace que la célula entre en una segunda fase conocida como *angiogénesis*, en la cual, crea nuevos vasos sanguíneos para lograr su objetivo. Finalmente, la célula se vasculariza, es decir, partes de ella pasan a la sangre, continuando con este comportamiento allí donde se depositen y, además, en muchas ocasiones se observa metástasis.

Es obvio, que la célula para llegar a la última fase va a necesitar cierto grado de daño genético, así como, superar determinadas barreras. Por un lado, obtener todas las capacidades necesarias para poder dividirse sin control, crear sus propios vasos sanguíneos, etc. dependerá de la obtención de mutaciones en su ADN que permitan

todo esto. Por otro lado, el cuerpo tiene mecanismos para evitar que esto ocurra, como por ejemplo, muerte por daño genético, evitar enviar orden de división e, incluso, el propio sistema inmune interviene para intentar evitar su crecimiento, además de otros muchos mecanismos.

Por tanto, esta es una enfermedad compleja de tipo ecológico y emergente, lo cual, dificulta su estudio, así como, su tratamiento.

### 2.4 LOS AUTÓMATAS CELULARES

El patrón mostrado en ciertas conchas marinas es generado mediante autómatas celulares naturales. Estas conchas muestran un aspecto exterior en el cual se observan patrones, los cuales les han permitido en muchas ocasiones ocultarse mejor en su entorno y, por ello, aumentar su supervivencia. Dichos patrones son de muy diversos tipos, por ejemplo, forman formas reconocibles, o formas fractales, entre otras.



Figura 2.1: *Conus textile shell*, especie de gastrópodo que presenta un autómata celular natural en su concha.

En ellos, cada célula toma un color o forma diferente según sus células vecinas, obedeciendo a una versión natural de unas reglas matemáticas. Basándose en esto, se conoce como autómata celular al modelo matemático abstracto que modela sistemas dinámicos que evolucionan en pasos discretos.

Son muy adecuados para ser implementados por ordenadores, útiles para simular comportamientos de sistemas complejos, como por ejemplo: colonias, epidemias, dinámica de fluidos, etc., y en muy diversas áreas, como por ejemplo, en física, biología, química, socioeconomía, etc.

En el libro *Simulating Complex Systems by Cellular Automata* de Jiri Kroc, Peter M.A. Sloot y Alfons G. Hoekstra [5] se presentan los fundamentos matemáticos que presentan los autómatas celulares.

Fueron descubiertos por John von Neumann en la década de 1950, aunque basado en trabajos teóricos previos de la década de 1940. A modo de resumen, los autómatas celulares cuentan con los siguientes componentes:

- Red regular de nodos, de cantidad finita o infinita, que representa la estructura espacial.
- En cada nodo, se coloca un autómata finito, es decir, representa un número finito de estados posibles. Aquellos que se encuentran ocupados, reciben el nombre de células o celdas.
- Cada célula se comunica con su entorno, pero sólo con una parte de él. Esto se conoce como vecindario, y existen varios tipos, por ejemplo, el vecindario de Moore, o el vecindario de vonNeumann, entre otros.
- Función que determina la evolución de los estados de las distintas células.

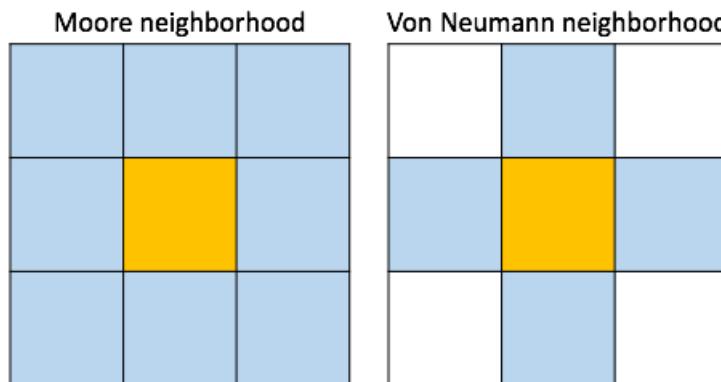


Figura 2.2: Tipos de vecindarios en los autómatas celulares.

Dada una configuración se puede estudiar las transformaciones aplicando las reglas, lo que se conoce como órbita. Dichas órbitas pueden ser muy complejas, a pesar de la sencillez del autómata celular.

Uno de los autómatas celulares más famosos es el *Juego de la vida de Conway*. Diseñado por el matemático John Horton Conway en la década de 1970. Es tan famoso

debido a que es equivalente a una máquina universal de Turing o, lo que es lo mismo, es un modelo de computación universal<sup>1</sup>. Esto es, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida.

Existen, además, muy diversas implementaciones y aplicaciones, como por ejemplo, para hacer música. En el libro *Game of Life Cellular Automata* de Andrew Adamatzky [1] se presenta el juego de la vida, y en el último capítulo se expone una forma de hacer música con él. Para ello, el autor utiliza cada celda ocupada por un autómata en cada iteración para tocar tres notas utilizando las coordenadas  $(x, y)$  de la posición del autómata en el plano cartesiano. De este modo, obtiene las tres notas utilizando una nota fija, y dos notas más mediante el valor de  $x$  e  $y$  como intervalo entre notas de un piano respecto de la nota fija y de la segunda nota respectivamente.

Se puede concluir, por tanto, que un autómata celular es una de las formas más adecuadas de simular comportamientos emergentes, al igual que los sistemas de Lindenmayer y, por tanto, adecuado para este tipo de problemas. José Santos y Ángel Monteagudo [6] optaron por utilizar autómatas celulares para sus trabajos.

En este caso, los autores pretenden simular el comportamiento emergente que presentan los tumores mediante un autómata celular. Ellos parten de los trabajos previos de Douglas Hanahan y Robert A. Weinberg llamados *The hallmarks of cancer* [3] y *The hallmarks of cancer: The next generation* [4] respectivamente. En ellos, se identifican una serie de marcadores asociados a ciertas mutaciones presentes en el genoma de las células que permiten la aparición y evolución de tumores.

---

<sup>1</sup><http://eprints.uwe.ac.uk/22323/1/thesis.pdf>

## ESTADO DEL ARTE

Para cumplir con uno de los primeros objetivos de este trabajo, se necesitaba realizar una búsqueda de artículos científicos sobre esta temática: *Simulación de crecimiento de tumores con autómatas celulares*.

Tras una primera selección, donde se descartaron los artículos de *arXiv*<sup>1</sup>, ya que, podrían estar aún en fase de revisión, así como, artículos demasiado antiguos o artículos cuyo enfoque quedaran fuera del ámbito de la inteligencia artificial, se obtuvo una lista con dos candidatos.

De los artículos seleccionados, se procedió a su estudio y extracción de información. A continuación, se presenta la lista de trabajos obtenidos y, posteriormente, se describen los trabajos candidatos.

### 3.1 RESULTADO DE LA BÚSQUEDA DE ARTÍCULOS

Los artículos encontrados durante la fase de búsqueda de información respecto a la temática *Simulación de crecimiento de tumores con autómatas celulares* son los siguientes:

- *Simulated Bain Tumor Growth Dynamics Using a Three-Dimensional Cellular Automaton* de A.R. Kansal, S. Torquato, G.R. Harsh IV, E.A. Chiocca and T.S. Deisboeck, 2000.
- *A cellular automaton model for tumour growth in inhomogeneous environment* de T. Alarcon, H.M. Byrne, P.K. Maini, 2003.
- *A Cellular automata model of tumor-immune system interactions* de D.G. Mallet, L.G. De Pillis, 2006.
- *Cellular Automaton of Idealized Brain Tumor Growth Dynamics* de , 2009.

---

<sup>1</sup><https://arxiv.org/>

- *A Review of Cellular Automata Models of tumor Growth* de Ankana Boondirek, Wan-napong Triampo, Narin Nuttavut, 2010.
- *Emergent Behaviors from A Cellular Automaton Model for Invasive Tumor Growth in Heterogeneous Microenvironments* de Yang Jiao, Salvatore Torquato, 2011-2012.
- *Study of cancer hallmarks relevance using a cellular automaton tumor growth model* de José Santos, Ángel Monteagudo, 2012.
- *Studying the capability of different cancer hallmarks to initiate tumor growth using a cellular automaton simulation. Application in a cancer stem cell context* de José Santos, Ángel Monteagudo, 2013.
- *A Cellular Automaton Model for Tumor Dormancy: Emergence of a Proliferative Switch* de Duyu Chen, Yang Jiao, Salvatore Torquato, 2014.
- *Analysis of behaviour transitions in tumor growth using a cellular automaton simulation* de José Santos, Ángel Monteagudo, 2014.
- *Treatment analysis in a cancer stem cell context using a tumor growth model based on cellular automata* de José Santos, Ángel Monteagudo, 2015.

### 3.2 *Cellular automaton of idealized brain tumor growth dynamics*

Este artículo [2] es uno de los dos candidatos estudiados con el objetivo de desarrollar este proyecto.

En él, se presenta un nuevo modelo de autómata celular para simular crecimientos emergente de tumores cerebrales. En concreto, se enfoca a un tipo de tumor, el tumor de Gompertzian. Los autores presentan la consecución de simular el crecimiento de dicho tumor en casi tres órdenes de magnitud en radio utilizando sólo cuatro parámetros microscópicos.

Su estudio tiene un elevado enfoque clínico modelizando la densidad de células y el tamaño de cada célula, empleando el algoritmo de triangulación *Delaunay*<sup>2</sup> junto a Polígonos de *Thiessen* o diagramas de *Voronoi*<sup>3</sup> para su representación.

<sup>2</sup>[https://es.wikipedia.org/wiki/Triangulaci%F3n\\_de\\_Delaunay](https://es.wikipedia.org/wiki/Triangulaci%F3n_de_Delaunay)

<sup>3</sup>[https://es.wikipedia.org/wiki/Pol%C3%ADgonos\\_de\\_Thiessen](https://es.wikipedia.org/wiki/Pol%C3%ADgonos_de_Thiessen)

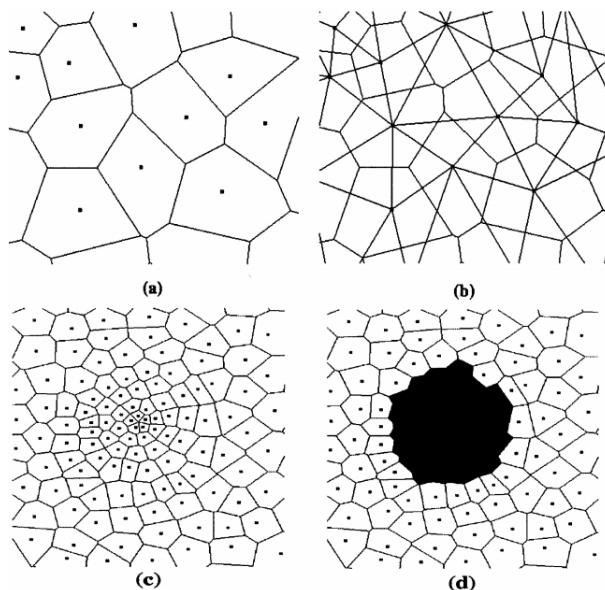


Figura 3.1: triangulación de Delaunay junto a Polígonos de Thiessen o diagramas de Voronoi para modelizar el tamaño de las células.

Realizan también un modelado temporal, para así estudiar la velocidad de crecimiento de los tumores. Sus resultados están muy especializados en un tipo de tumor, por lo que no realizan un estudio genérico para varios tipos de tumores, con diferentes parámetros, configuraciones o marcadores.

### *3.3 Analysis of behaviour transitions in tumour growth using a cellular automaton simulation*

Este artículo [6] forma parte de una serie de artículos [7] [8] [9] en el cual los autores, con un enfoque genérico, pretenden simular el crecimiento de tumores.

Para ello, se basan en varios trabajos, pero principalmente en los trabajos de Douglas Hanahan y Robert A. Weinberg [3] [4], donde se presentan varios marcadores presentes en las células, los cuales consisten en una serie de mutaciones que permiten a las células presentar un comportamiento canceroso.

En su enfoque, utilizan un autómata celular que sigue un modelo de eventos, esto es, porque los autores necesitan, por un lado, simular las escalas de tiempo, y por otro, modelizar cuándo las células necesitan realizar la mitosis.

En cada uno de sus trabajos, presentan diferentes enfoques. Esto es, se centran, o descartan, varios marcadores y comportamientos. Por ejemplo, en el trabajo en el que se centra este proyecto [6] los autores deciden ignorar dos marcadores, *AG* o angiogénesis, y *MT* o metástasis.

Además, deciden no modelizar el crecimiento celular, es decir, todas las células en la rejilla ocupan el mismo espacio.

Es un enfoque genérico, que permite simular cualquier tipo de tumor modificando los parámetros de la simulación, y permite también estudiar los distintos comportamientos emergentes según qué marcador toma o no relevancia frente al resto.

# MODELO DE EVENTOS

Los autores, José Santos y Ángel Monteagudo, decidieron utilizar un autómata celular siguiendo un modelo de eventos. Es decir, por un lado, la simulación se realiza sobre una rejilla en tres dimensiones, comenzando con una única célula en el centro de la misma. Por otro lado, se programan una serie de eventos para cada una de las células de la rejilla, aleatoriamente entre 5 y 10 (ambos inclusive) iteraciones en el futuro.

Las características y propiedades del sistema se describen en las secciones de este capítulo que se muestran a continuación.

## 4.1 OBJETIVOS DEL ARTÍCULO

En este trabajo, los autores tiene como objetivo estudiar las fases iniciales del crecimiento tumoral a nivel celular, estudiando diferentes configuraciones iniciales y observando las diferentes dinámicas y comportamientos. Así como, estudiar, también, la relevancia en la progresión del cancer de cada uno de los marcadores, y qué dinámicas presenta cada uno de ellos.

Además, centran el foco de su trabajo en las transiciones cuando las células cancerosas mueren con una probabilidad dada durante la progresión del sistema celular.

No son objetivos de este trabajo simular las fases posteriores al crecimiento inicial de tumores, ni el aumento de tamaño de las células. Tampoco es objetivo modelar los diferentes factores que pueden provocar mutaciones y/o muerte celular.

Por último, en este artículo no se presta atención a las células madre del cáncer. Los autores en otros trabajos [9] sí modelizan la presencia de dichas células.

## 4.2 LAS CÉLULAS, GENOMA Y PROPIEDADES

Cada célula estará alojada en una única posición del autómata. En esta simulación, no se modeliza el tamaño de las células, es decir, aunque en las células cancerosas se observa, además de un comportamiento replicativo sin control, un crecimiento en su tamaño sin control, los autores no han tenido en cuenta esto.

Cada célula presente en la rejilla cuenta con un genoma y unas propiedades únicas para cada una de ellas. En cuanto a su genoma, cuenta con 5 variables binarias que representan la presencia o no de hasta cinco mutaciones, las cuales, definen un determinado comportamiento canceroso en cada caso. Estas mutaciones son las siguientes:

- **SG:** Autogeneración de los mensajes de crecimiento. Esto es, la mutación que permite que la célula genere sus propios mensajes para ejecutar la división con independencia externa.
- **IGI:** Inhibición de las señales de anticrecimiento. Esto es, ante la recepción de una orden de detener su crecimiento, la célula tiene una mutación que le permite un mecanismo de ignorancia de los mismos.
- **EA:** Evasión de apoptosis. Esto es, la célula puede, mediante mutación, no hacer caso ante una orden de apoptosis, o muerte celular controlada.
- **EI:** Inmortalidad efectiva. Esto es, la célula adquiere una mutación que permite evitar un límite replicativo existente, entre otros factores, por el tamaño del telómero.
- **GI:** Instabilidad genética. Esto es, una mutación que permite a la célula acumular más daño genético, es decir, la tasa de mutación base se va incrementando con el paso del tiempo.

Además, cada célula tiene una tasa de mutación base y un tamaño de telómero. El primero, es utilizado a la hora de añadir nuevas mutaciones a la célula. Y la segunda, es un límite replicativo debido a que el ADN queda desprotegido para futuras mutaciones y podrían ocurrir errores.

## 4.3 PARÁMETROS GLOBALES

De cara a la simulación, existen una serie de parámetros globales que inciden en determinados mecanismos y, por tanto, afectan en su evolución. Esto son:

- **t**: Tamaño de la rejilla.
- **m**: Valor por defecto que indica la tasa de mutación base de cada célula al inicio.
- **tl**: Valor por defecto que indica el tamaño del telómero de cada célula al inicio.
- **e**: Valor por defecto utilizado para definir la probabilidad de una célula de morir por daño genético.
- **i**: Valor por defecto utilizado como factor de aumento de la tasa de mutación base de las células.
- **g**: Valor por defecto para ver qué probabilidad hay de que una célula cancerosa mate a un vecino para poder reproducirse.
- **a**: Valor por defecto para ver con qué probabilidad una célula muere de forma aleatoria. Este parámetro se introduce como forma de simular las muy diversas causas que pueden originar en la muerte de la célula, por ejemplo, recibir una alta dosis de radiación entre otras.

## 4.4 PRUEBAS PREVIAS A LA MITOSIS

El modelo de eventos, a la hora de realizar la mitosis (división celular), realiza tres pruebas:

- **Prueba de muerte aleatoria**: La célula muere con una probabilidad dada ( $1/a$ ).
- **Prueba de daño genético**: A mayor cantidad de mutaciones, mayor probabilidad de que la célula muera ( $n/e$ , donde  $n$  es el número de mutaciones de la célula). Excepto, que la célula tenga la mutación *EA* presente en su genoma.
- **Prueba de mitosis**: En realidad, se trata de tres pruebas cuyo resultado debe ser positivo para que la célula pueda ejecutar la mitosis:

- **Comprobación del factor de crecimiento:** La célula sólo puede realizar la mitosis si se encuentra dentro de un límite espacial predefinido. Es decir, hay suficiente factor de crecimiento. Fuera de este área no podrá realizar la mitosis, excepto si el marcador *SG* está activo.
- **Comprobación de ignorancia de inhibición de crecimiento:** Si no hay espacio en el vecindario, la célula no podrá realizar la división. Excepto, que la mutación *IGI* esté presente, en cuyo caso, reemplazará a un vecino con una probabilidad dada ( $1/g$ ).
- **Comprobación de potencial replicativo sin límites:** Si el tamaño del telómero es 0, la célula muere. Excepto si el marcador *EI* está presente.

En estas pruebas, se utilizan los parámetros globales, además, de las propiedades y el genoma de la célula.

Por tanto, si una célula pasa todas las pruebas podrá entrar en fase de mitosis.

## 4.5 MITOSIS

La mitosis es la fase donde se realiza la división celular, lo cual origina una nueva célula, llamada célula hija, que es una copia exacta de la célula original.

Durante este proceso, se pueden producir una serie de errores, bien de forma natural, o bien, debido a mutaciones presentes en la célula original que hacen más probable un error.

En este trabajo, el proceso de mitosis cuenta con los siguientes pasos:

- Incremento de la tasa de mutación base  $m$  (de acuerdo al parámetro global  $i$ ) si el marcador *GI* está presente.
- Añadir nuevas mutaciones a la célula de acuerdo a la tasa base de mutación ( $1/m$ ).
- Decrementar el telómero en ambas células en una unidad, siempre que su tamaño no sea 0.

A continuación, se programan nuevos eventos mitóticos para ambas hijas como se describe a continuación.

## 4.6 EVENTOS MITÓTICOS

Un evento mitótico se puede ver como la posibilidad de que una célula ejecute o no la división (o mitosis), en el futuro.

Todas las células en la rejilla contarán con un evento mitótico en el futuro. Cuando se selecciona una de las candidatas en cada iteración la célula candidata debe pasar las pruebas previas a la mitosis para ver si la misma puede o no ejecutar la mitosis. Si el resultado es positivo, se realiza la mitosis y se crean nuevos eventos mitóticos en el futuro para la célula original y, también, para la célula hija. Si el resultado es negativo, pero la célula no ha muerto, se programa un nuevo evento mitótico en el futuro.

La programación de eventos mitóticos futuros consiste en la generación, de forma aleatoria, de un valor entre 5 y 10 iteraciones (ambas inclusive) del bucle principal. Estos valores tienen que ver con el ciclo de vida temporal de las células, sobre el cual, se detalla más información en la siguiente sección.

## 4.7 EQUIVALENCIA TEMPORAL

El ciclo de vida de las células biológicas es de entre 15 y 24 horas. Este ciclo se divide en 5 fases, que son: fase G0, fase G1, fase S, fase G2 y fase M.

Una célula parte en fase de reposo (G0). Si hay espacio suficiente a su alrededor (vecindario) automáticamente pasa a fase G1. En la simulación, G1 se simula mediante el paso del tiempo (iteraciones) y la programación de eventos mitóticos en el futuro (entre 5 y 10 iteraciones, ambas inclusive). Además, en la simulación no se tiene en cuenta el crecimiento celular.

La fase S es cuando tiene lugar la replicación del ADN. La cual, puede introducir una mutación ocasionalmente. La célula entra en una última fase previa a la mitosis, llamada fase G2, en la cual se producen una serie de comprobaciones sobre el daño genético. Esto, puede provocar la apoptosis (muerte celular programada) en la célula.

Finalmente, la célula entra en la fase M o de mitosis. Todo este ciclo, que en la simulación toma entre 5 y 10 iteraciones (15 y 24 horas en células biológicas) da una media de 2,6 horas por iteración. Por ejemplo, para 5000 iteraciones se tienen unas 77,4 semanas aproximadamente.

## 4.8 BUCLE PRINCIPAL DEL MODELO DE EVENTOS

El modelo de eventos consiste en realizar un número de iteraciones, y en cada iteración realizar una serie de pruebas con aquellas células que tienen una posibilidad de evento mitótico en ese momento.

Se realizan tres pruebas, la primera prueba consiste en comprobar si a la célula que tiene un posible evento mitótico en esta iteración aplica o no muerte aleatoria. Esto se realiza mediante un sorteo con una probabilidad dada ( $1/a$ ). Mediante esta prueba se pretende simular las diferentes causas que pueden llevar a la muerte de una célula.

En la segunda prueba, se pretende simular los mecanismos del cuerpo que evitan que una célula siga reproduciéndose con un elevado daño genético. Esto se realiza mediante un sorteo con una probabilidad dada ( $n/e$ ) a partir del número de mutaciones de la célula entre un parámetro global establecido para la simulación.

Y, por último, se realiza la prueba para ver si dicha célula realiza o no la mitosis. Para ello, se realizan tres pruebas que deben ser positivas, estas son: comprobación del factor de crecimiento, comprobación de ignorancia de inhibición de crecimiento, y comprobación de potencial replicativo sin límites.

```

t ← 0
Nuevo_Evento_Mitótico(t+5,t+10)
MIENTRAS existan eventos en la cola HACER
    Obtener_Evento(t)
    PRUEBA 1: MUERTE_ALEATORIACELULA()
    PRUEBA 2: MUERTE_DAÑO_GENÉTICO()
    PRUEBA MITOSIS():
        PRUEBA 3: COMPROBACIÓN_FACTOR_CRECIMIENTO()
        PRUEBA 4: COMPROBACIÓN_IGNORANCIA_INHIBICIÓN_CRECIMIENTO()
        PRUEBA 5: COMPROBACIÓN_POTENCIAL_REPLICATIVO_SIN_LIMITES()
    SI se han pasado las tres pruebas para la mitosis ENTONCES
        HACER_MITOSIS()
        PUSH_EVENT()
    SINO
        PUSH_EVENT()
    
```

Figura 4.1: Modelo de eventos para simulación del cáncer.

La prueba número tres, o comprobación del factor de crecimiento, consiste en com-

probar si la célula está dentro de un espacio predefinido en el cual llegan los mensajes para poder realizar la mitosis, y si no es así, se comprueba que tiene la mutación *SG* que le permite realizar la mitosis a pesar de estar en dicha posición. En cuanto a la prueba número cuatro, o comprobación de ignorancia de inhibición de crecimiento, consiste en comprobar si hay espacio en el vecindario para realizar la mitosis, o en caso contrario, si tiene el marcador *IGI* reemplazará a un vecino con una probabilidad dada ( $1/g$ ) mediante la muerte del vecino elegido. Por último, la prueba número cinco, o comprobación de potencial replicativo sin límites, consiste en comprobar si el tamaño del telómero es mayor a cero, o en caso contrario, si tiene la mutación *EI* que le permite realizar la mitosis a pesar de no contar con telómero.

Si la célula pasa todas las pruebas se procede a realizar la mitosis y a programar un nuevo evento mitótico para ambas. En otro caso, se programa un nuevo evento mitótico para la misma célula si esta no ha muerto como resultado de alguno de las pruebas previas a la mitosis.

La implementación final en un lenguaje de programación de propósito general varía dada la complejidad de implementación de las pruebas descritas. En el siguiente capítulo se comenta y se muestra la implementación final del algoritmo presentado en esta sección.



# IMPLEMENTACIÓN DEL SISTEMA

En este capítulo, se detalla el diseño e implementación del sistema. Tecnologías utilizadas, arquitectura y diseño, así como, detalles de implementación y patrones que se han seguido.

## 5.1 TECNOLOGÍAS

Este proyecto ha sido desarrollado empleando diversas tecnologías y soluciones de *Python*.

*Python*<sup>1</sup> es un lenguaje multiparadigma que es ampliamente utilizado en ámbitos científicos y de investigación. Cuenta con multitud de herramientas para la realización de *tests*, gestión de dependencias, etc., entre las que se encuentran *Unittest*<sup>2</sup> (tests unitarios) y *Pip*<sup>3</sup> (gestor de dependencias), que han sido empleadas en este trabajo.

Se ha optado por utilizar el lenguaje de programación *Python*, en su versión 3, concretamente, en el momento de la realización de este trabajo el desarrollo se ha realizado bajo la versión 3.6.2 de dicho lenguaje.

La implementación de *Python* utilizada es *CPython*<sup>4</sup>.

Además, se han empleado los siguientes paquetes:

- *NumPy* como librería fundamental para computación científica del ecosistema *SciPy*.
- *Matplotlib* como librería para creación de gráficas del ecosistema *SciPy*.
- *Plotly* como librería gráfica, utilizada en este caso sólo para representar figuras en tres dimensiones.

---

<sup>1</sup><https://www.python.org/>

<sup>2</sup><https://docs.python.org/3/library/unittest.html>

<sup>3</sup><https://pypi.python.org/pypi/pip>

<sup>4</sup><https://www.toptal.com/python/por-que-hay-tantos-pythons/es>

Por último, para su desarrollo, se han utilizado otras herramientas como: *Git* y *Git-flow*, *Github*, *Travis*, *Landscape*, *Coveralls* y *Docker*.

## 5.2 ARQUITECTURA Y DISEÑO

Para el desarrollo de este proyecto se ha optado por seguir dos paradigmas simultáneamente. Por un lado, se ha seguido el paradigma orientado a objeto, esto es, modelizar las partes del sistema con objetos que encapsulan las propiedades y métodos necesarios según las necesidades u objetivos de cada una de ellas y, las cuales, se describen en la siguiente sección de este capítulo. Por otro lado, se ha seguido el paradigma funcional, esto es, diseñar el sistema, o partes de él, en forma de funciones, las cuales para ejecutar su lógica utilizan únicamente los parámetros que recibe.

Ambos paradigmas, comentado brevemente en el párrafo anterior, tienen muchos más aspectos, enfoques y soluciones de las que se comentan. El programa cuenta con un único punto de entrada, en este caso, siguiendo el patrón *Singleton*. En cuanto a los patrones utilizados, se van a describir en la siguiente sección de este capítulo donde se presentan los diferentes módulos del mismo.

## 5.3 MÓDULOS DEL SISTEMA

En este caso, el programa cuenta con 6 módulos. El módulo principal, *automata.py*, presenta una clase que sigue el patrón *Estrategia* y, el cual, engloba la función principal de ejecución, así como, el resto de funciones necesarias para su correcta ejecución. Además, importa y utiliza el resto de módulos, los cuales, se describen a continuación.

### 5.3.1 Módulo *genome.py*

Este es el primero de los módulos que se ha desarrollado, y es el más simple de todos. Consta de una clase que contiene únicamente cinco variables binarias, una por cada mutación que se modeliza en este sistema.

Como se ha descrito previamente, las mutaciones que contiene esta clase son las siguientes:

- **SG:** Autogeneración de los mensajes de crecimiento. Esto es, la mutación que

permite que la célula genere sus propios mensajes para ejecutar la división con independencia externa.

- **IGI:** Inhibición de las señales de anticrecimiento. Esto es, ante la recepción de una orden de detener su crecimiento, la célula tiene una mutación que le permite un mecanismo de ignorancia de los mismos.
- **EA:** Evasión de apoptosis. Esto es, la célula puede, mediante mutación, no hacer caso ante una orden de apoptosis, o muerte celular controlada.
- **EI:** Inmortalidad efectiva. Esto es, la célula adquiere una mutación que permite evitar un límite replicativo existente, entre otros factores, por el tamaño del telómero.
- **GI:** Inestabilidad genética. Esto es, una mutación que permite a la célula acumular más daño genético, es decir, la tasa de mutación base se va incrementando con el paso del tiempo.

Además, expone un método que permite obtener el número de mutaciones que tiene cada instancia de este tipo.

En definitiva, se trata de una clase que, por composición, estará contenida dentro de otra clase que modeliza a la célula, representando su genoma, y que se explica a continuación.

### 5.3.2 Módulo *cell.py*

El siguiente módulo, partiendo de la clase genoma, es el que contiene la clase que modeliza a cada una de las células de la simulación.

Como se ha comentado previamente contiene, por composición, un atributo que representa al genoma de la célula y que se trata de un objeto tal y como se ha descrito en la sección anterior de este capítulo. Además, presenta algunos atributos más necesarios para la simulación, y son:

- Atributo que representa el tamaño del telómero.
- Atributo que representa la tasa de mutación de la célula.
- Atributo que representa su posición en la rejilla, es decir, un conjunto de tres elementos que representan su posición en cada una de las dimensiones  $((x, y, z))$ .

Por último, cuenta con varios métodos necesarios para la simulación, y son los siguientes:

- *decrease\_telomer()*: Método que se encarga de cambiar el estado interno del objeto, realizando un decrecimiento en una unidad del telómero.
- *increment\_base\_mutation\_rate(i)*: Las células ven alterada su tasa de mutación base, utilizada para originar nuevas mutaciones durante la mitosis, con la presencia de la mutación asociada a la inestabilidad genética o *GI*. Este método realiza este proceso de acuerdo a una tasa de modificación, o incremento, de esta tasa, la cual es recibida por parámetros.
- *mutations()*: Método que devuelve el número de mutaciones que tiene la célula en su genoma.
- *add\_mutations()*: Método que se encarga de generar las nuevas mutaciones que pueden tener lugar durante la mitosis.
- *perform\_mitosis()*: El proceso de mitosis queda modelado en este método, esto es, incrementar tasa de mutación base, añadir nuevas mutaciones y realizar una copia de sí misma.

### 5.3.3 Módulo *simulation\_globals.py*

De cara a realizar la simulación son necesarios una serie de parámetros que se necesitarán manipular para realizar los experimentos que se describirán en capítulos posteriores. Esto lleva al siguiente módulo, en el cual se tiene una clase que contiene, sólamente, atributos que almacenan todos estos parámetros.

Dichos parámetros son los siguientes:

- Tasa de mutación base o *m*.
- Tamaño del telómero o *tl*.
- Probabilidad de evasión de apoptosis o *e*.
- Factor de incremento de la tasa de mutación base o *i*.
- Probabilidad de matar a un vecino para realizar la mitosis o *g*.

- Probabilidad de muerte aleatoria de la célula o  $a$ .
- Límite espacial predefinido.
- Límite inferior para evento mitótico en el futuro.
- Límite superior para evento mitótico en el futuro.

En este fichero, además, se encuentran los valores por defecto de la simulación según los autores del artículo [6] en el que se basa este trabajo. En este caso, se declaran como constantes, ya que estos, en caso de ser utilizados en la simulación, no varían.

#### 5.3.4 Módulo *experiments.py*

Para realizar la mitosis, se necesitan realizar varias pruebas o experimentos. Estos están contenidos dentro de una clase como métodos. Los parámetros necesarios para ejecutar dichas pruebas se reciben por parámetros. Los métodos, son los siguientes:

- *random\_death\_test()*: Prueba que indica si hay o no muerte aleatoria de la célula.
- *random\_death\_test(n, ea)*: Prueba que indica si hay muerte por daño genético según el número de mutaciones de la célula que está realizando el proceso de mitosis, excepto que esté presente el marcador con la mutación que permite evadir la apoptosis.
- *random\_death\_test(sg, spatial\_boundary)*: Prueba que está dentro del límite espacial o, lo que es lo mismo, si existe suficiente factor de crecimiento como para ejecutar la mitosis. Excepto, si tiene presente el marcador *sg*, que permite ejecutar la mitosis si la célula se encuentra fuera de dicho espacio.
- *random\_death\_test(igi)*: Prueba que comprueba si existe espacio para ubicar la célula hija una vez realizada la mitosis. Si no hay espacio y la célula tiene la mutación asociada al marcador *igi*, podrá matar a un vecino para ubicar a la célula hija.
- *random\_death\_test(tl, ei)*: Prueba que comprueba si el telómero tiene tamaño mayor a 0 para realizar la mitosis. Si el tamaño es 0, la célula puede realizar mitosis si tiene presente la mutación asociada al marcador *ei*.

### 5.3.5 Módulo *grid.py*

El módulo *grid.py* almacena la información relativa a la rejilla en la que tiene lugar la simulación. Esto es, tiene como propiedad o atributos el ancho, alto y largo del mismo.

Pero, la parte útil de este módulo son los métodos que permiten a la simulación obtener el vecindario, controlando sus límites, y obtener qué celdas del vecindario están ocupadas y cuales no. Por tanto, sus métodos son los siguientes:

- *neighborhood(origin, radio)*: Método que recibe un centro de tipo  $(x, y, z)$  y un radio, para devolver una lista con las posiciones que conforman el vecindario.
- *filter(value, limit)*: Método que permite saber si un valor está dentro de un límite espacial. Es utilizada en únicamente en el siguiente métodos.
- *check\_limits(cube, limit)*: Método que comprueba que, de una lista de posiciones que conforman el vecindario, no existe una posición que exceda el límite de la rejilla. Devuelve, la lista sin dichas posiciones.
- *classify\_neighborhood(cube, cells)*: Métdodo que clasifica las células de un cubo que representa un vecindario en función de las posiciones ocupadas y las que están libres.

### 5.3.6 Módulo *automata.py*

Módulo principal del sistema que contiene la lógica de ejecución del mismo. El sistema está diseñado siguiendo el patrón *Singleton*, es decir, sólo hay un objeto autómata que, implementando el patrón *estrategia*, tiene un, método con la lógica de ejecución.

La lógica de ejecución se encuentra descrita en la sección 4.8 de este documento.

Además, tiene como atributos la agenda de eventos mitóticos, así como, el objeto que encapsula los experimentos necesarios para saber si se aplica o no mitosis a cada célula, los parámetros de simulación, el objeto con la lógica necesaria para gestionar la rejilla y, por último, la responsabilidad de generar las medidas utilizando el módulo de analítica que se describe a continuación.

```

def run(self, statics_enable):
    if statics_enable:
        statics = dict()
    for iteration in range(self.iterations):
        if statics_enable and iteration%50 == 0:
            statics[iteration] = copy.deepcopy(self.cells)
        events = self.pop_events(iteration) if iteration in self.mitotic_agenda else []
        for event in events: # event is a tuple with three elements == position
            cell = self.cells[event]
            if self.experiments.random_death_test():
                self.kill_cell(event)
            elif self.experiments.genetic_damage_test(cell.mutations(), cell.genome.ea):
                self.kill_cell(event)
            else:
                test_1, test_2, test_3 = self.first_test(cell), self.second_test(cell, iteration), self.third_test(cell)
                if test_1 and test_2 and test_3: #Perform mitosis
                    self.copy_and_choose_new_position(event, cell, iteration)
                    self.push_event(iteration + self.future_mitotic_event(), event)
                elif test_3:
                    self.push_event(iteration + self.future_mitotic_event(), event)
                else: # Telomer is 0 and EI is OFF
                    self.kill_cell(event)
            if iteration in self.mitotic_agenda: # Remove current iteration
                del self.mitotic_agenda[iteration]
        if statics_enable:
            result = self.analytics.get_measurements(statics)
            self.analytics.plot_health_vs_carcino(result[0])
            self.analytics.plot_mutations(result[1])
            self.analytics.plot_grid_plotly(self.cells)
            self.analytics.pretty_show(self.iterations, self.cells)

```

Figura 5.1: Código con el bucle principal de la simulación.

### 5.3.7 Módulo *analytics.py*

Este último módulo, reúne las funciones necesarias para hacer medidas sobre las células de la rejilla, como por ejemplo, células sanas o cancerígenas, cuántas células tienen un determinado marcador del cáncer activo, etc.

Además, contiene funciones para construir y mostrar gráficas con vista a presentar la evolución del sistema a lo largo de la simulación.

## 5.4 INSTALACIÓN Y EJECUCIÓN

En esta última sección del capítulo dedicado a la implementación, se expone la información necesaria para su instalación y uso. Existen dos formas de conseguirlo, la primera, a través de una ejecución nativa, es decir, sobre la propia máquina local. En segundo lugar, sobre *Docker*, lo cual permite no realizar ninguna instalación sobre la máquina local.

### 5.4.1 Prerrequisitos

Dependiendo del tipo de instalación, se dan los siguientes prerrequisitos:

- Para una ejecución sobre la máquina local, se necesita:
  - Instalar la versión 3.6+ de *Python*<sup>5</sup>.
  - Instalar el gestor de paquetes de *Python*, *pip*<sup>6</sup>.
  - Instalar el control de versiones *Git*<sup>7</sup>.
  - Clonar el repositorio ejecutando en consola *git clone https://github.com/MULCIA/TFMSTGCA.git*.
  - Instalar dependencias con *pip*, para ello, ejecutar en consola *pip install -r requirements.txt*.
  - Para una ejecución con *Docker* se necesita:
    - Instalar la última versión de *Docker*<sup>8</sup>.
    - Instalar el control de versiones *Git*<sup>9</sup>.
    - Clonar el repositorio ejecutando en consola *git clone https://github.com/MULCIA/TFMSTGCA.git*.

### 5.4.2 Ejecución

Dependiendo del modo elegido, es decir, ejecución en máquina local o ejecución en *Docker*, su ejecución es diferente.

En primer lugar, se describe como hacer una ejecución en máquina local. Una vez cumplidos los requisitos descritos anteriormente, para ejecutar el proyecto sólo hay que ejecutar, dentro del directorio que contiene el proyecto, el siguiente comando: *python main.py*.

---

<sup>5</sup><https://www.python.org/downloads/>

<sup>6</sup><https://pip.pypa.io/en/stable/installing/>

<sup>7</sup><https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

<sup>8</sup><https://docs.docker.com/engine/installation/>

<sup>9</sup><https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Su ejecución se realiza automáticamente y al finalizar mostrará una serie de gráficas:

- Gráfica que muestra la evolución en número de las células sanas frente a las células cancerosas.
- Gráfica que muestra la evolución en número de los marcadores.
- Gráfica que muestra la rejilla en tres dimensiones mostrando las células sanas (gris) y las células cancerosas (verde).

En segundo lugar, se describe como hacer la ejecución en *Docker*. Una vez cumplido los requisitos anteriores, existen dos formas de ejecutar la simulación: ejecutar los comandos de docker, o emplear unos scripts que se encuentran embebidos en el proyecto. Dichos scripts sólo se pueden emplear de tratarse de una máquina *Linux*.

Los comandos de consola de *Docker* son los siguientes:

- Para construir la imagen: `docker build -t TFMSTGCA/tfm ..`
- Para levantar el contenedor: `docker run --name tfm -d TFMSTGCA/tfm:latest tail -f /dev/null`. Se ejecuta en modo demonio y se mantiene su ejecución. Al finalizar el proceso de creación y ejecución del contenedor, se realiza una ejecución de los tests adjuntos al proyecto.
- Para ejecutar la simulación: `docker exec -it tfm bash`. Una vez ejecutado, con la terminal del contenedor en la consola, ejecutar como se describe previamente para la ejecución en máquina local.

Completar (Eliminar contenedores).

### 5.4.3 Parámetros de la simulación

La simulación cuenta con la posibilidad de ajustar los parámetros que se describen en la sección anterior, donde se comenta el módulo *simulation\_globals.py*. En el punto de entrada de la aplicación, en *main.py*, se encuentra la declaración del objeto *SimulationGlobals*.

Para realizar ajustes, existen dos formas:

- Utilizar la configuración por defecto de los autores.
- Utilizar una configuración propia.

Para el primer caso, se cuenta con una serie de constantes que permiten dejar el sistema con la configuración por defecto que dan los autores

Para el segundo caso, sólo hay que establecer el parámetro deseado sustituyendo cualquiera de las constantes.

Además, existe una configuración en cuanto a tamaño de la rejilla y número de iteraciones, esto es:

```
simulationGlobals = SimulationGlobals(BASE_MUTATION_RATE, TELOMER_LENGTH, EVADE_APOPTOSIS,  
FACTOR_INCREASE_BASE_RATE_MUTATION, KILL_NEIGHBOR_PROBABILITY, RANDOM_CELL_DEATH,  
PREDEFINED_SPATIAL_BOUNDARY, MIN_FUTURE_MITOTIC_EVENT, MAX_FUTURE_MITOTIC_EVENT)  
  
automata = Automata(50, 1000, simulationGlobals)
```

Figura 5.2: Código para configuración por defecto de una simulación.

# EXPERIMENTOS DE LOS AUTORES

José Santos y Ángel Monteagudo realizaron una serie de simulaciones en su artículo donde intentan estudiar la influencia de los marcadores en la evolución de la simulación. Todas las simulaciones, y sus resultados, se comentan en esta sección.

El tamaño de rejilla utilizada para todos los experimentos es de  $50^3$ . Esto son, 125,000 células posibles en la rejilla.

## 6.1 INFLUENCIA DEL PARÁMETRO *Tasa de mutación base* ( $m$ )

Los autores en su artículo presentan 3 experimentos utilizando los valores por defecto y variando el parámetro *Tasa de mutación base* ( $m$ ) para estudiar como afecta dicho parámetro en la proliferación del cáncer.

Cada experimento se muestra en una serie de gráficas en las cuales se presenta como resultado la media de 5 ejecuciones diferentes.

A continuación, se muestra cada uno de los experimentos especificando en cada caso el valor  $m$  utilizado. El resto de parámetros de la simulación se mantiene constante con los valores considerados por defectos, que son:

Cuadro 6.1: Valores de los parámetros, excepto  $m$ .

Nombre	Símbolo	Valor
Tamaño del telómero	tl	50
Muerte por daño genético	e	10
Factor de incremento de tasa de mutación base	i	100
Muerte de un vecino	g	30
Muerte aleatoria	a	1000

### 6.1.1 Experimento 1: Tasa de mutación base igual a 10.000

En este primer experimento, los autores presentan un valor para el parámetro tasa de mutación base de  $m = 10000$ . Realizan una simulación de 1000 iteraciones.

Es importante recordar, que al utilizar  $1/m$  para representar la probabilidad de que aparezcan diferentes mutaciones al realizar la mitosis, a mayor valor del parámetro  $m$ , menor probabilidad de que ocurran mutaciones.

En la simulación, el número de células cancerosas es muy bajo y crece muy levemente, no superando las 10000 células de las 125000 totales. En cuanto a lo marcadores, el marcador *SG* toma ventaja y tiene ocurrencia en la mayoría de células cancerosas (una célula cancerosa es aquella que tiene, al menos, una mutación). El siguiente marcador presente, aunque con mucha diferencia, es *EA*. El resto de marcadores presenta un comportamiento similar y su presencia es casi inexistente.

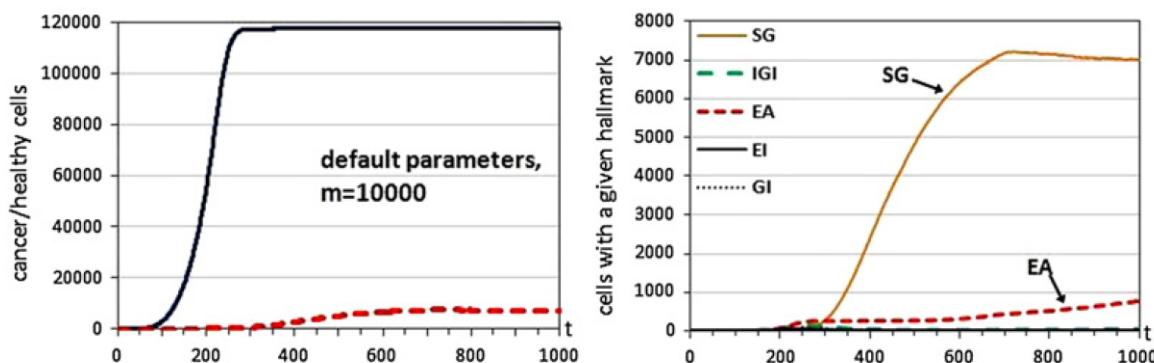


Figura 6.1: Resultados obtenidos por los autores para el experimento con  $m = 10,000$ .

### 6.1.2 Experimento 2: Tasa de mutación base igual a 1.000

En el segundo experimento, los autores alteran el valor para el parámetro tasa de mutación base a  $m = 1000$ .

Los resultados obtenidos para 1000 iteraciones presentan varias diferencias. En primer lugar, el número de células cancerosas aumenta, lo cual, se explica por la mayor probabilidad de mutaciones provocadas al realizar la mitosis. En segundo lugar, en cuanto a los marcadores, aparecen todos los marcadores, con un comportamiento bastante similar entre ellos. Se pueden destacar los marcadores *EA* y *SG*, que toman algo de ventaja respecto al resto.

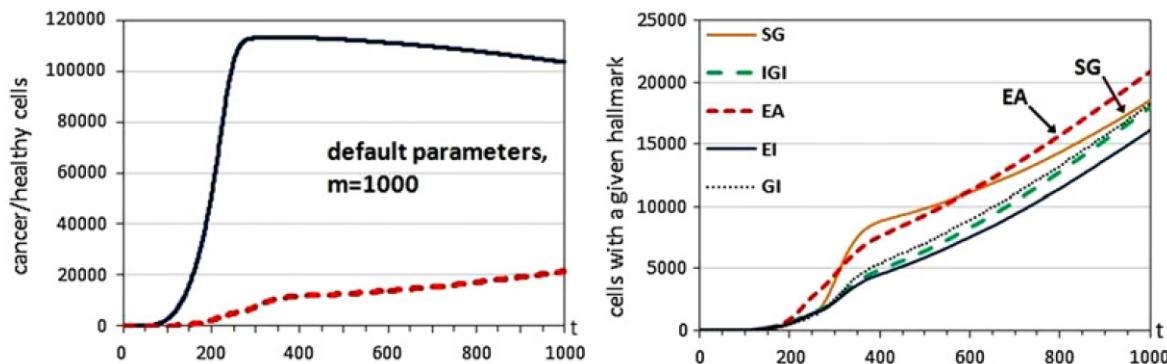


Figura 6.2: Resultados obtenidos por los autores para el experimento con  $m = 1,000$ .

### 6.1.3 Experimento 3: Tasa de mutación base igual a 100

En este tercer y último experimento de este tipo, los autores obtienen cambios relevantes frente a los dos experimentos anteriores.

En primer lugar, el número de células cancerosas pronto supera a las células sanas, e incluso, a lo largo de las iteraciones terminan, las células cancerosas, representando casi la totalidad de la rejilla. Esto se explica por la mayor probabilidad de aparición de mutaciones durante la mitosis, en este caso, la tasa de mutación base es de  $m = 100$ .

En segundo lugar, en cuanto a marcadores, ocurren todos los marcadores con un comportamiento similar, tomando ventaja, pero muy levemente, los marcadores *EA* e *IGI*. La principal diferencia en este punto es la presencia del marcador *IGI*, el cual, tiene mayor presencia que otros. El marcador *SG* no toma ventaja en este punto, como si lo hacía en los dos experimentos anteriores.

El marcador *EA* siempre está presente y cobra un papel fundamental, ya que, permite la proliferación de células cancerosas debido a que supone que la célula con dicho marcador puede evadir la muerte celular programada.

## 6.2 INFLUENCIA DEL RESTO DE PARÁMETROS

Los autores realizan un segundo experimento con el objetivo de observar los efectos que tienen el resto de parámetros sobre la simulación. Para ello, varían los parámetros de la simulación, como se puede observar en la siguiente tabla:

Se observa un comportamiento bastante representativo, por ejemplo, un menor va-

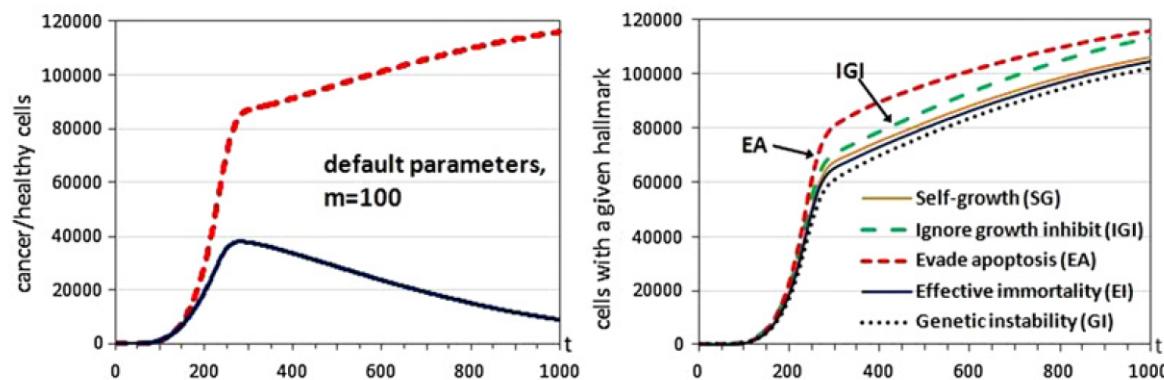


Figura 6.3: Resultados obtenidos por los autores para el experimento con  $m = 100$ .

Cuadro 6.2: Valores de los parámetros.

Nombre	Símbolo	Valor
Tasa de mutación base	$m$	100.000
Tamaño del telómero	$tl$	35
Muerte por daño genético	$e$	20
Factor de incremento de tasa de mutación base	$i$	100
Muerte de un vecino	$g$	10
Muerte aleatoria	$a$	400

lor para el número de iteraciones implica menos eventos mitóticos en las células sanas, y un menor valor del parámetro  $a$  facilita la aparición de sitios vacantes que quedan disponibles para la propagación de células cancerosas, en conexión con una mayor probabilidad de reemplazar a un vecino para realizar la mitosis (parámetro  $g$ ).

En la gráfica de evolución de los marcadores se observa que en este caso el marcador dominante es  $EI$ . En este caso, se realizan 5000 iteraciones. El marcador  $EI$  implica la progresión de estas células debido a que evitan la muerte por agotamiento del telómero. En la simulación, por contra, tener un telómero más corto implica tener menos oportunidades de división, como ocurre con la célula inicial del centro de la rejilla.

El resto de marcadores aparecen más tarde en la simulación. El siguiente marcador en hacer presencia es  $EA$ , lo cual permite a las células evadir la muerte celular programada. Este marcador también favorece la proliferación del cáncer, por tanto, toma cada vez más ventaja. El marcador  $SG$  aparece algo más tarde que los anteriores y permite proliferar al cáncer fuera del límite reproductivo, por tanto, alcanza un límite y se mantiene estable.

### 6.3. INFLUENCIA DE PARÁMETROS CON REJILLA COMPLETA DE CÉLULAS SANAS

El último marcador en aparecer es *IGI*. No prolifera rápidamente debido a no disponer de espacio vacante para realizar la división.

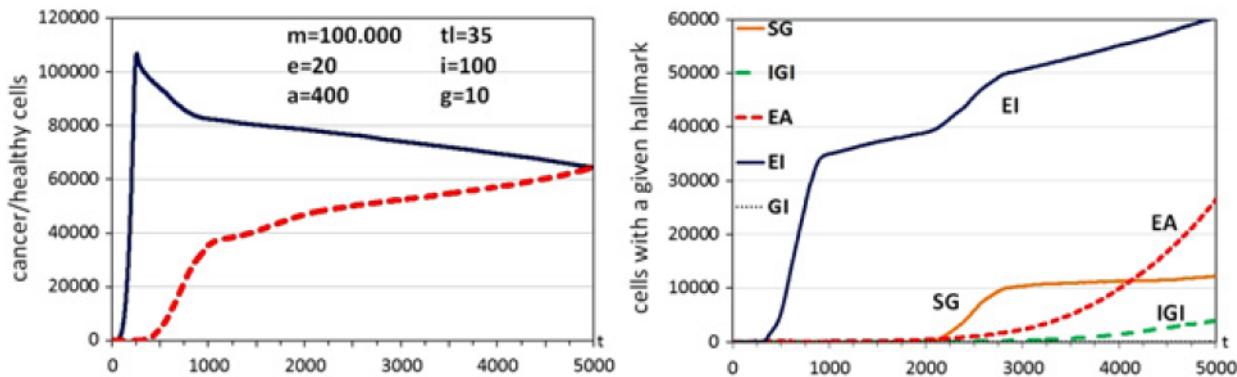


Figura 6.4: Resultado de la simulación para comprobar la influencia del resto de parámetros.

### 6.3 INFLUENCIA DE PARÁMETROS CON REJILLA COMPLETA DE CÉLULAS SANAS

En este caso, se parte con una rejilla completa de células sanas. En cuanto a los parámetros, se utilizan los mismos que en la sección anterior.

Se realizan diferentes estrategias en cuanto al número de iteraciones, desde 8000 iteraciones hasta 100000 iteraciones, es decir, la simulación se corresponde con una equivalencia temporal de 2,3 años y 29,7 años respectivamente.

Completar.

### 6.4 RELEVANCIA DE LOS MARCADORES

En este caso, los autores intentan responder a la siguiente pregunta: *¿Cuál sería el comportamiento emergente si algún marcador no estuviera presente y no aplicaran sus efectos?*.

Conocer el efecto de cada marcador en el comportamiento emergente para crecimiento de tumores puede resultar útil para mejorar las terapias contra el cáncer.

En su estudio, los autores encuentran el marcador *EA* o de evasión de apoptosis

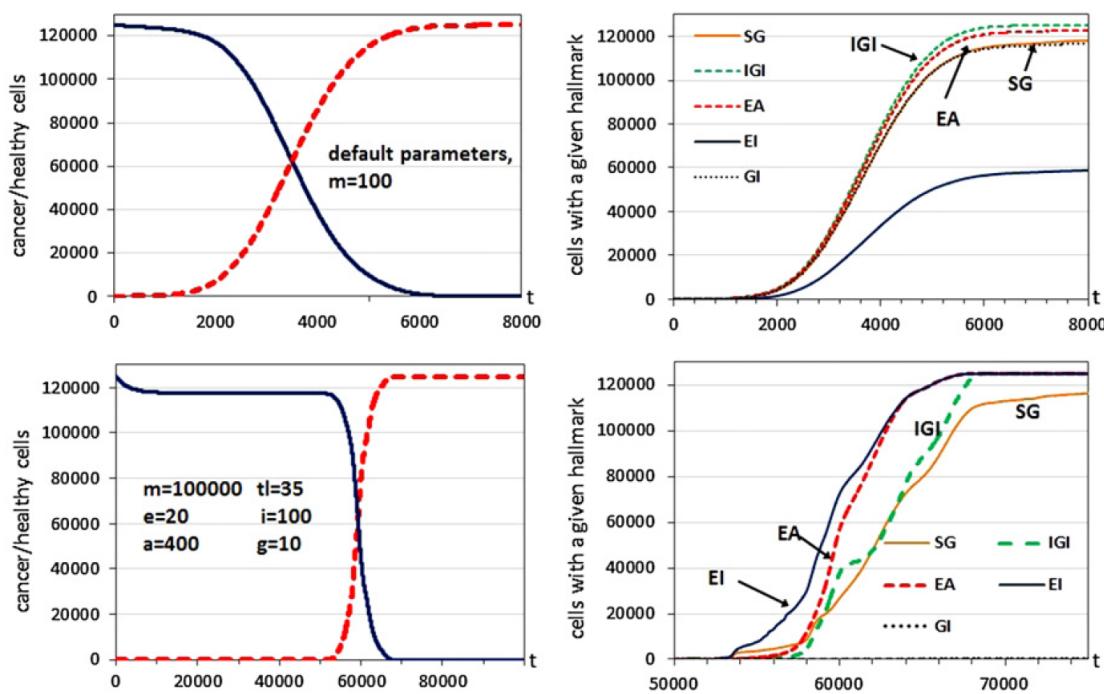


Figura 6.5: Completar.

como el más relevante de todos, debido a que se reduce drásticamente el número de células cancerígenas. Tras él, los siguientes marcadores más relevantes por orden son *GI* o de inestabilidad genética y, a continuación, *IGI* o inhibición de señal de parada de crecimiento. Esto se debe a, en el caso del marcador *GI*, se reducen las posibilidades de adquirir una mutación. Respecto al marcador *IGI*, esto se debe a que cuando la rejilla no tiene espacio libre, sobre todo con células sanas, se reduce la probabilidad de, para realizar la división, una célula mate a un vecino para conseguir espacio.

Completar.

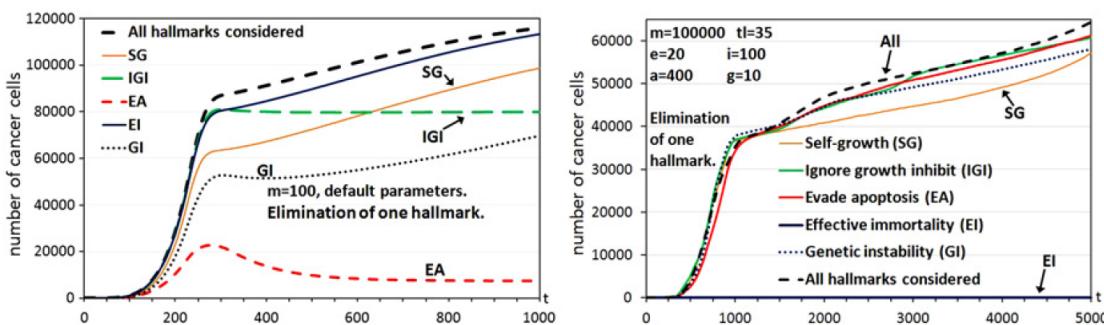


Figura 6.6: Completar.

## 6.5 COMPORTAMIENTO DE LAS TRANSICIONES

El objetivo es estudiar el comportamiento de las transiciones cuando un agente actúa contra las células cancerosas. En este caso, la simulación comienza con la rejilla completa de células cancerosas, y se supone una terapia perfecta, en el sentido que la droga, en este caso, actúa sólo contra las células cancerosas.

Completar.

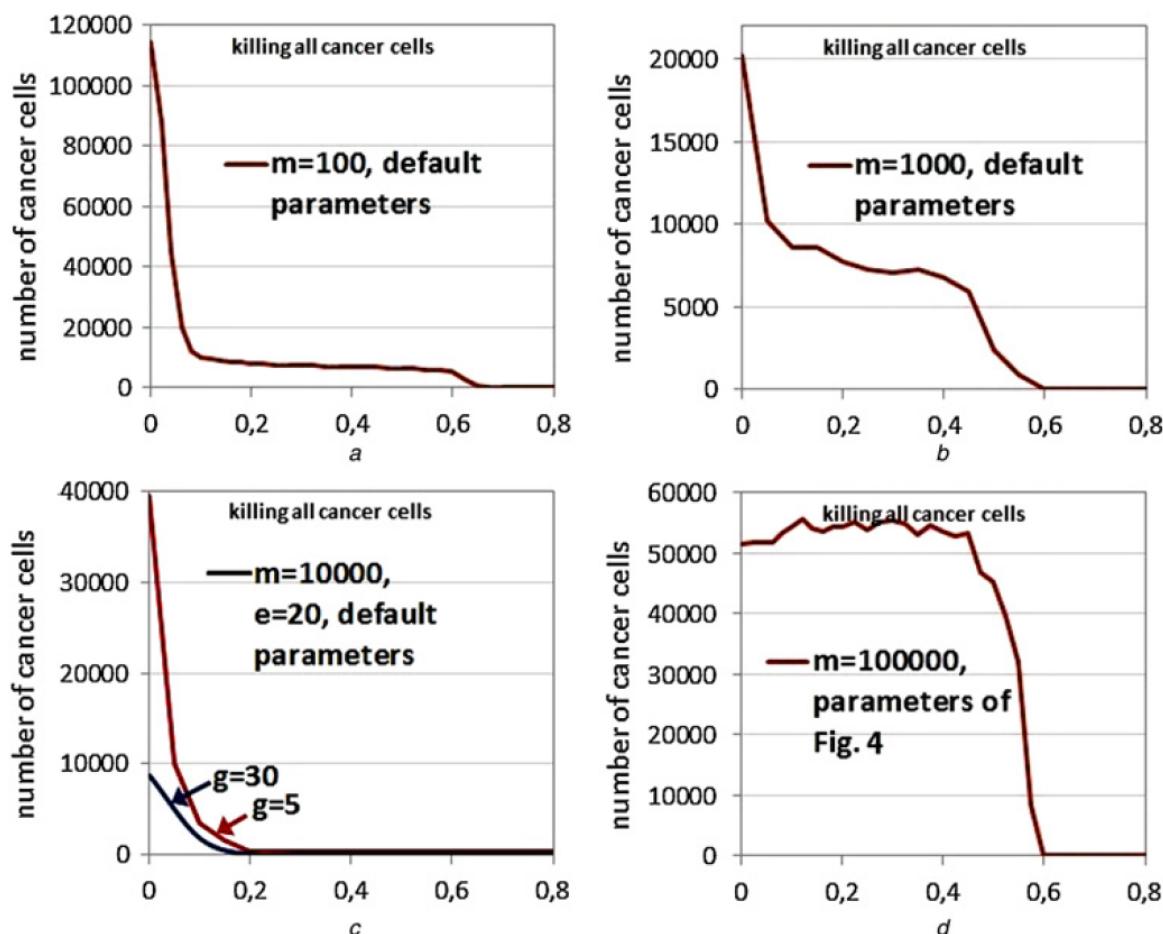


Figura 6.7: Completar.

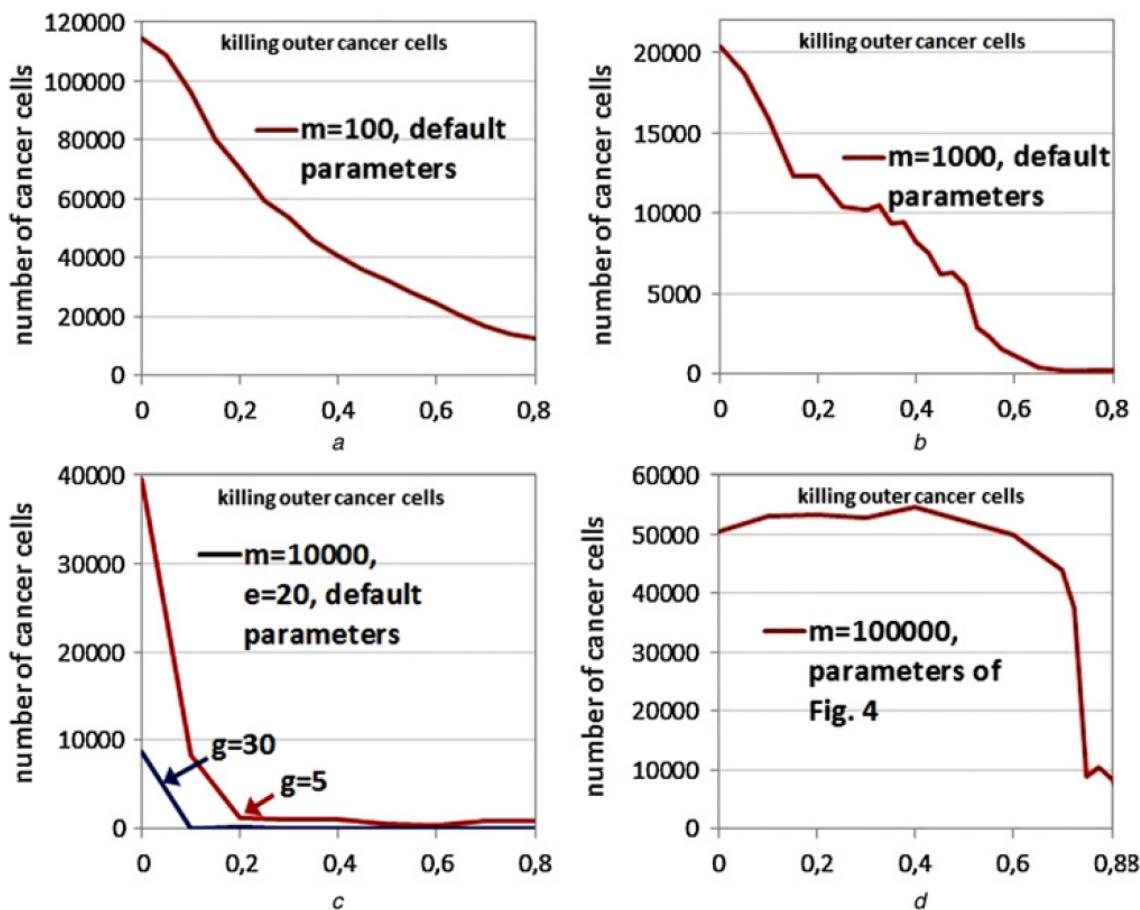


Figura 6.8: Completar.

# RESULTADOS PROPIOS

En esta sección, se presentan los resultados obtenidos al reproducir cada uno de los experimentos de la sección anterior, es decir, los experimentos que los autores presentan en su trabajo [6].

Antes de todo, hay que tener en cuenta la componente aleatoria de la simulación. Esto es, por la realización de sorteos de cara a ejecutar la mitosis u otra acción. En consecuencia, la aparición de un marcador u otro, como se verá a continuación, provoca alteraciones en la simulación, de modo que el comportamiento puede variar. En este caso, lo que se pretende es llegar a la misma conclusión que los autores, por tanto, el análisis de los resultados propios debe hacerse con ese objetivo en mente.

## 7.1 INFLUENCIA DEL PARÁMETRO *Tasa de mutación base* ( $m$ )

Las células de la simulación, y como se ha descrito en secciones previas, tienen asociado una propiedad que está relacionada con la aparición de nuevas mutaciones durante la mitosis, este es, el parámetro *tasa de mutación base* o  $m$ . En este experimento, se pretende someter a la simulación a diferentes configuraciones de este parámetros para estudiar qué progresión presentan los tumores.

Se parte de una probabilidad de aparición de mutaciones baja, ya que, el sorteo se efectua con una probabilidad de  $1/m$ , por lo que, a mayor valor del parámetro  $m$ , menor probabilidad de que ocurre. En este caso, se realizan tres simulaciones, para  $m = 10000$ ,  $m = 1000$  y  $m = 100$ .

### 7.1.1 Experimento 1: Tasa de mutación base igual a 10.000

En esta simulación, se obtienen tres gráficas respecto de la misma. En la primera, se observa la progresión en el número de células sanas frente a células cancerosas.

En cuanto al número de células de uno u otro tipo, se observa una progresión bastante similar a la obtenida por los autores. Por un lado, el número de células sanas crece muy rápido y ocupa el 90% del espacio en torno a las 300 iteraciones. Por otro lado, el número de células cancerosas presenta una progresión similar con un repunte leve en torno a la iteración 300 con un crecimiento muy leve.

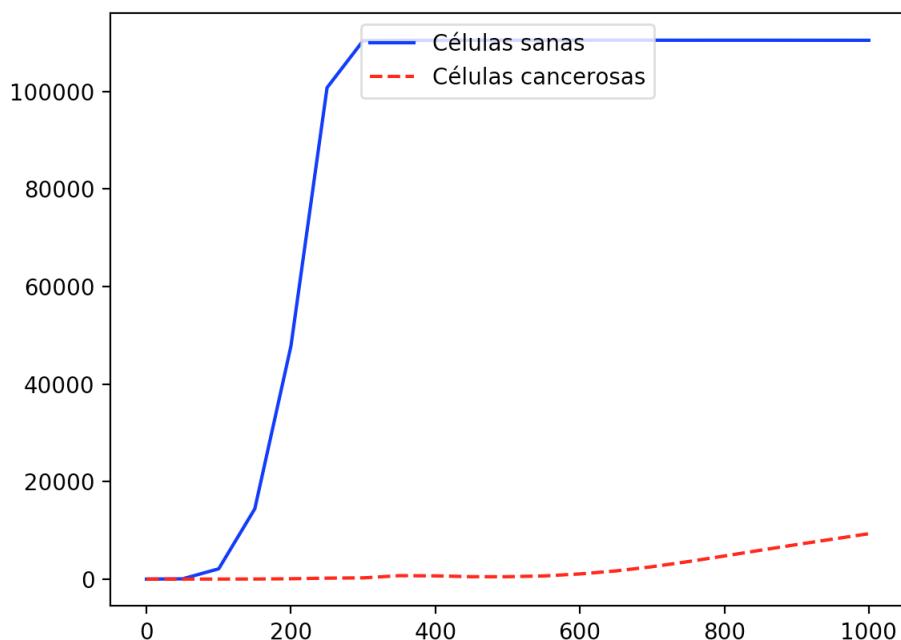


Figura 7.1: Células sanas frente a células cancerosas como resultado de la simulación para el experimento con  $m = 10,000$ .

En cuanto a los marcadores presentes y a su progresión, se observan algunas diferencias. En primer lugar, el marcador *SG* toma ventaja respecto al resto, hasta que en torno a la iteración 500 aproximadamente también toma relevancia el marcador *EI*. Esto es especialmente relevante, e incluso, coherente con la simulación de los propios autores.

El marcador *SG* tiene una leve caída que se ve frenada cuando aparece el marcador *EI*. Este último, provoca que las células puedan evitar la muerte celular o apoptosis. Esto hace que el marcador *SG* tenga mayor presencia y que el número de células vaya aumentando.

Se produce la peculiaridad de que ambas mutaciones, *SG* y *EI* se dan en las mismas células. Esto, como se observa en la gráfica anterior, no tiene por qué afectar al número total de células cancerosas.

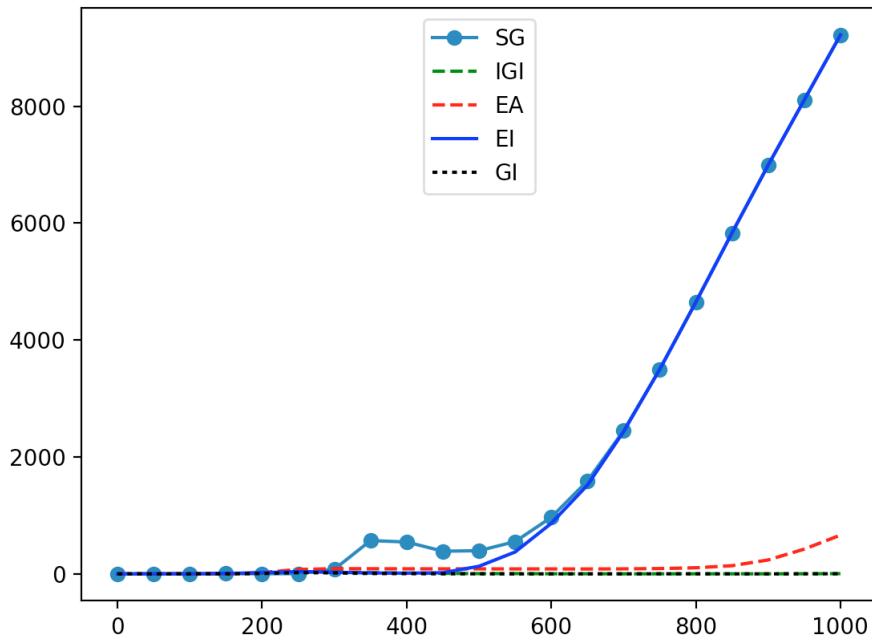


Figura 7.2: Evolución de los marcadores como resultado de la simulación para el experimento con  $m = 10,000$ .

Por último, se muestra una vista de la rejilla, marcando las células cancerosas en verde. Tiene aspecto de estar completamente ocupado por células cancerosas, pero esto es un efecto visual. Las células encuentran cabida en los límites de la rejilla al haber obtenido el marcador  $SG$ , el cual permite ejecutar la mitosis más allá del espacio pre-definido con suficiente factor de crecimiento.

En la simulación de los autores se observa este mismo comportamiento. Esto es, por tanto, otro punto que es coherente con los que presentan los autores en su estudio.

### 7.1.2 Experimento 2: Tasa de mutación base igual a 1.000

El segundo experimento sobre el parámetro *tasa de mutación base* se mantienen todos los parámetros y se cambia dicho parámetro a  $m = 1000$ .

Al igual que antes, se muestran tres gráficas. En la primera de ellas, que muestra la progresión de células sanas frente a células cancerosas, se observa un comportamiento similar al obtenido por los autores. Es decir, aparición de células cancerosas en torno a la iteración 150, crecimiento moderado, aunque algo mayor, lo que se explica por una mayor probabilidad de aparición de mutaciones durante la mitosis. Al finalizar la simulación, se observa algo más de 17000 células cancerosas frente a unas 20000 que

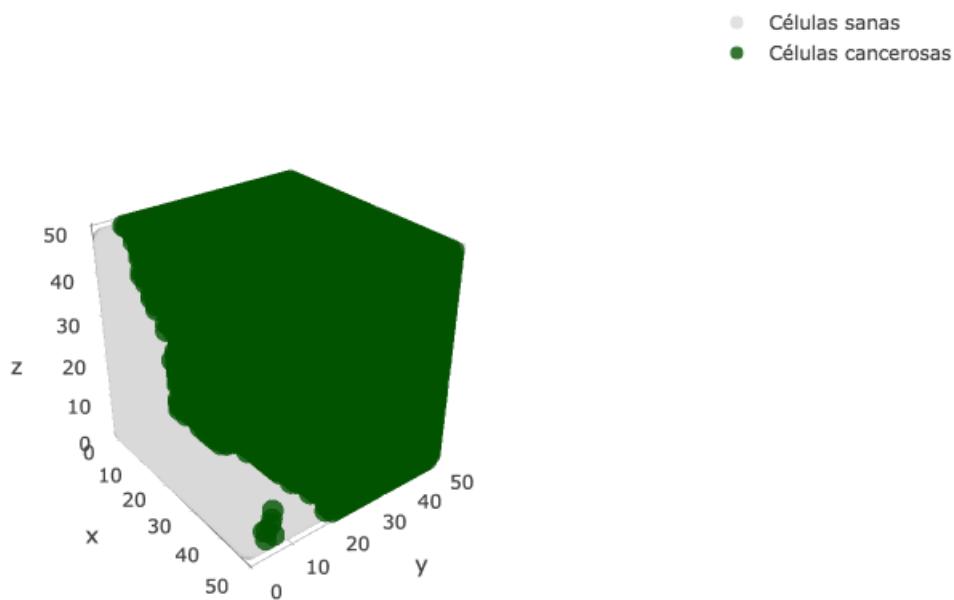


Figura 7.3: Rejilla resultante de la simulación para el experimento con  $m = 10,000$ .

obtienen los autores.

En cuanto a los marcadores, su progresión es diferente. En las primera 250 iteraciones el comportamiento es similar al obtenido por los autores, el primer marcador en tomar ventaja es *EA* y justo después se ve superado por *SG*. La principal diferencia es que en torno a la iteración 450 el marcador *EI* aparece y tiene un crecimiento fuerte colocándose como el segundo marcador que más relevancia toma.

Como se ha descrito, la aparición de los marcadores se realiza en base a una serie de sorteos con una probabilidad dada. No se puede controlar su aparición. Lo relevante es que el resto de marcadores, *IGI* y *GI*, casi no tienen relevancia ni presencia.

Es decir, se obtiene un comportamiento parecido en cuanto a número de células sanas contra células cancerosas pero con diferente presencia de marcadores. Esto quiere decir que los marcadores que más ventaja toman son los que realmente afectan a la simulación. En este caso, obtener el marcador *EI* hace que perduren las células que cuentan con dicha mutación en su genoma y que se extienda en posteriores divisiones.

En cuanto a la rejilla, al igual que los resultados presentados por los autores, las células cancerosas tienden a ocupar los límites de la rejilla, esto es debido a la pronta aparición del marcador *SG*. Las células con otro tipo de marcadores tienden a encon-

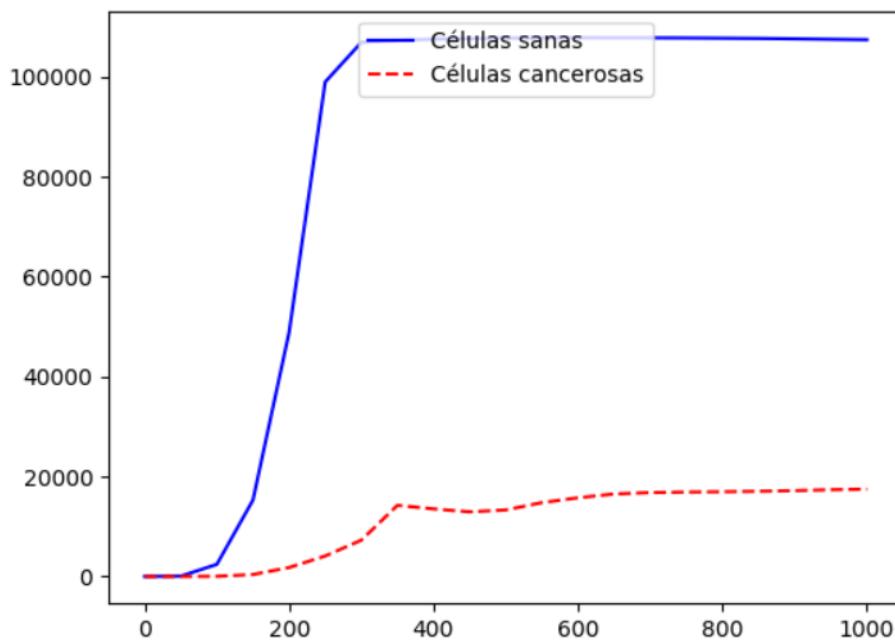


Figura 7.4: Células sanas frente a células cancerosas como resultado de la simulación para el experimento con  $m = 1,000$ .

trarse más hacia el centro de la rejilla.

### 7.1.3 Experimento 3: Tasa de mutación base igual a 100

En este último experimento para el parámetros *tasa de mutación base*, se quiere aumentar la probabilidad de que se produzcan mutaciones durante la mitosis.

En cuanto a células sanas frente a células cancerosas se observa como a lo largo de la simulación la tendencia es que las células cancerosas van tomando relevancia frente a las células sanas. Los autores observan que las células cancerosas superan a las sanas en torno a la iteración 180 aproximadamente. En este caso, en la simulación propia no se llega a ese resultado, pero si se observa una tendencia esperada, y es que las células cancerosas superen en algún punto a las células sanas.

En cuanto a las mutaciones, al igual que los autores el marcador *EA* es el más relevante, aunque lo hace con mayor diferencia. Esto puede explicar la diferencia de comportamiento frente a lo mostrado anteriormente. El marcador *IGI* va tomando ventaja al final de la simulación, y no desde un principio como si ocurre en la simulación de los autores.

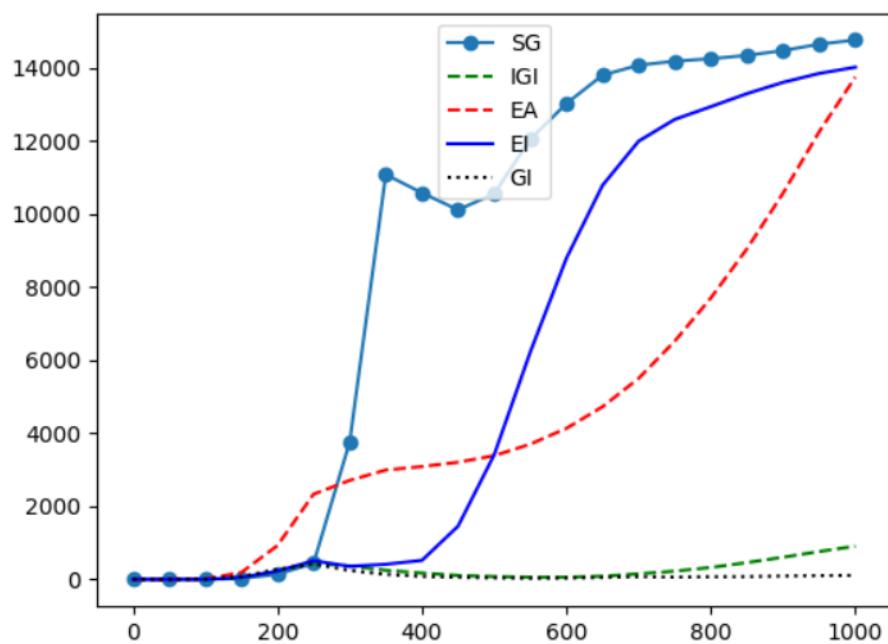


Figura 7.5: Evolución de los marcadores como resultado de la simulación para el experimento con  $m = 1,000$ .

Es relevante como el marcador  $GI$  en todas las simulaciones no toma relevancia.

En cuanto a la posición de las células cancerosas en la rejilla, las conclusiones son las mismas que en los dos experimentos anteriores, con mayor presencia de células cancerosas en la parte central de la rejilla, lo que es coherente con lo que han obtenido los autores.

## 7.1. INFLUENCIA DEL PARÁMETRO TASA DE MUTACIÓN BASE ( $M$ )

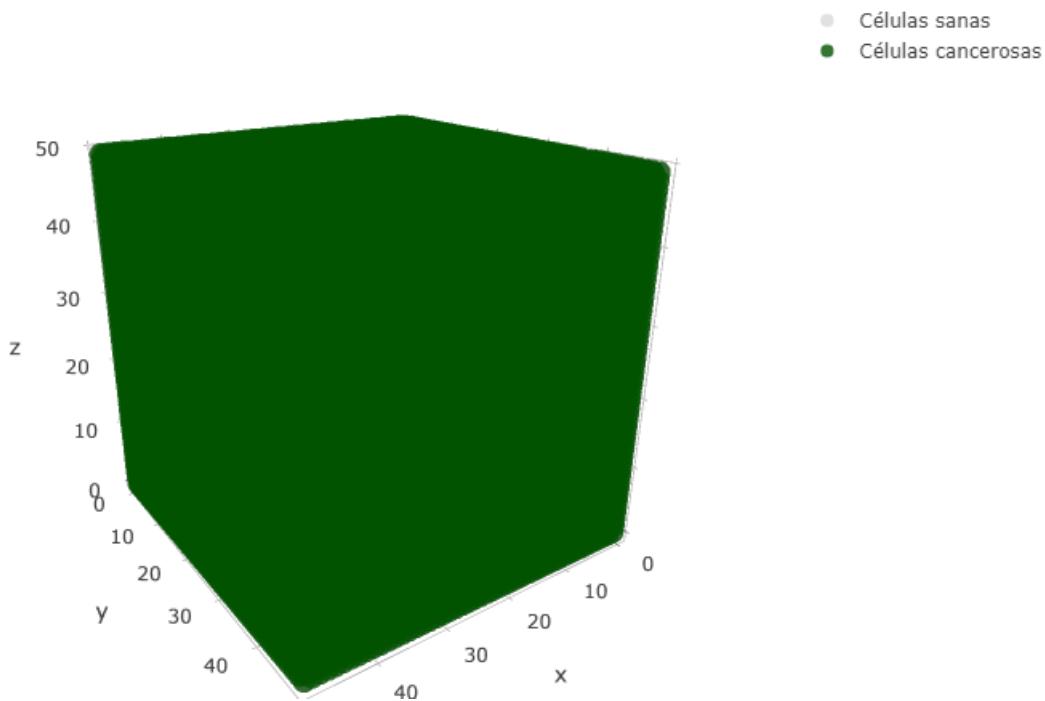


Figura 7.6: Rejilla resultante de la simulación para el experimento con  $m = 10,000$ .

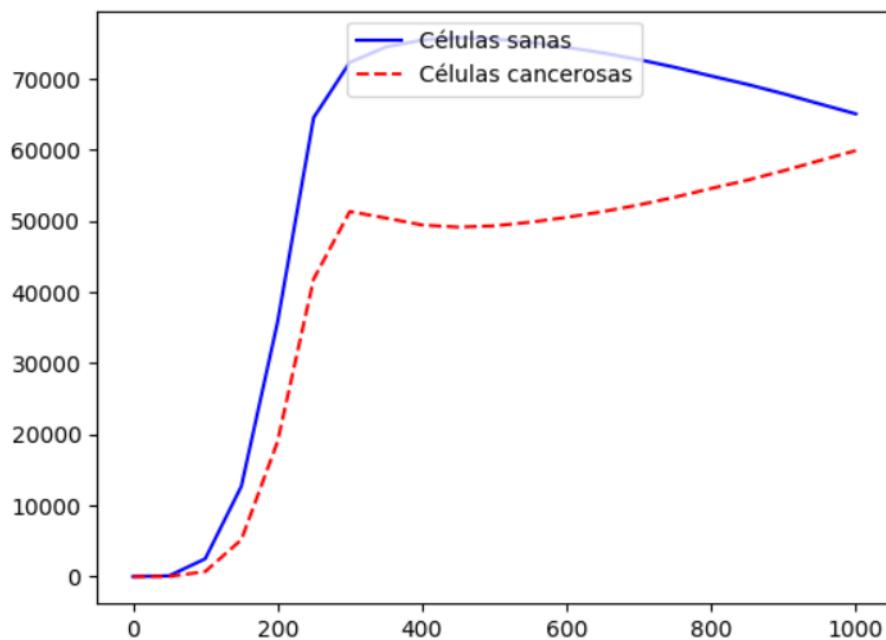


Figura 7.7: Células sanas frente a células cancerosas como resultado de la simulación para el experimento con  $m = 100$ .

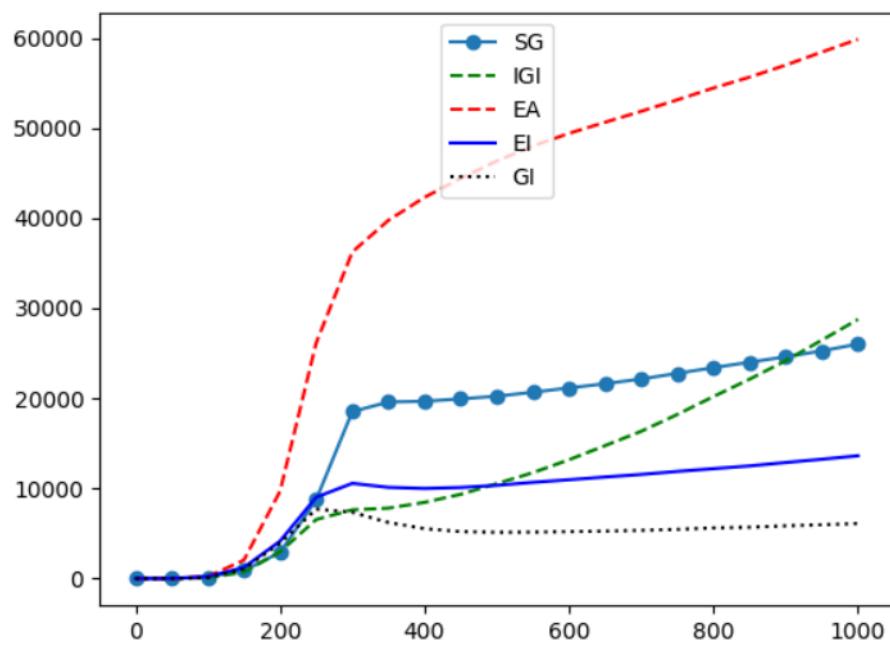


Figura 7.8: Evolución de los marcadores como resultado de la simulación para el experimento con  $m = 100$ .

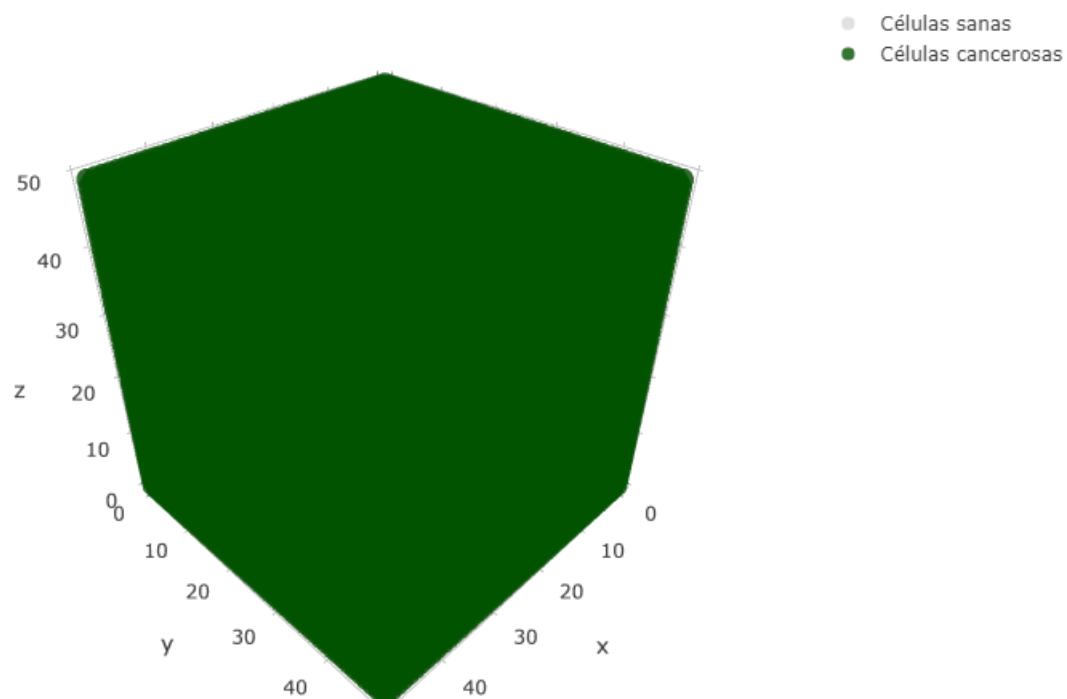


Figura 7.9: Rejilla resultante de la simulación para el experimento con  $m = 100$ .

## CONCLUSIÓN

El cáncer es una enfermedad compleja y que requiere de una gran cantidad de eventualidades para que se dé su aparición y proliferación.

El cuerpo humano cuenta con mecanismos para evitar la aparición de esta enfermedad, que van desde muerte por daño genético hasta la intervención del propio sistema inmune. Este trabajo se ha centrado en la reproducción de los experimentos para intentar reproducir los resultados de los autores del trabajo original [6].

Desde el punto de vista de los marcadores y los parámetros asociados a la simulación se han realizado distintas configuraciones a fin de estudiar qué comportamiento tiene el sistema y, por tanto, cómo crece el tumor. La aparición y proliferación de determinadas mutaciones en las células, así como, en qué lugar ocurren, provocan determinados comportamientos.

El marcador más destacable es el marcador asociado a la capacidad de las células de generar sus propios mensajes de división celular, o *SG*, el cual propicia la proliferación por la parte exterior de la rejilla. Esto es algo observado en todos los tipos de tumores, y es lo que hace que se convierta en una enfermedad mortal, ya que, provoca que crezca y se expanda provocando compresión en tejidos adyacentes.

Este comportamiento se explica porque encuentra espacio suficiente, y esto permite, además, pasar con éxito dos pruebas previas a la mitosis: comprobación del factor de crecimiento, y comprobación de ignorancia de inhibición de crecimiento. Es decir, obtiene sus propios mensajes para realizar la división celular, y no necesita matar a un vecino para poder realizar la mitosis.

El resto de marcadores, excepto el marcador de inestabilidad genética o *GI*, provocan que la proliferación e invasividad sea mayor, o lo que es lo mismo, que estemos ante una masa anormal maligna, o también llamado carcinoma.

El marcador de inestabilidad genética o *GI* provoca que existan células más propensas a sufrir mutaciones en el proceso de división celular. Esto conlleva la aparición de células cancerosas en la parte central de la rejilla, que combinado a otros marcadores

termina por encontrar el camino a su proliferación. Esto, requiere de mayor tiempo de simulación ya que en el centro de la rejilla hay menos espacio y actúan mecanismos de protección contra el cáncer de manera más efectiva.

Si esta simulación fuese completa y fiel a la realidad de cualquier tipo de cancer se podría utilizar como herramienta de capacidad predictiva y, también, como herramienta para probar la acción de fármacos mediante la inhibición de varios de los mecanismos que forman la simulación.

## TRABAJO FUTURO

Este trabajo presenta varias de limitaciones que permiten varias vias de trabajo futuro.

En este trabajo se modelan algunos de los mecanismos que permiten el crecimiento de tumores, así como, su proliferación. Existen más marcadores asociados a mutaciones que derivan en un comportamiento canceroso de las células que lo contienen que no se tienen en cuenta en este trabajo. Dos de ellos son el marcador que permite una angiogénesis sostenida (*AG*) y el marcador que permita la metastasis (*MT*).

Tampoco se tiene en cuenta las células madres del cancer, que son aquellas células que si tras aplicar una terapia no son completamente eliminadas, provocan la reaparición del tumor, llegando a ser más invasivas que el tumor original en algunos casos.

Otra posibilidad, es tener en cuenta el tamaño y la densidad de las células que componen el tumor, permitiendo una representación más fiel de la realidad.

Por último, se podría permitir activar o desactivar ciertos mecanismos en su conjunto, para poder simular la aplicación de determinadas terapias contra esta enfermedad.



# GLOSARIO DE TÉRMINOS

**Adenoma** Masa anormal que tiene un comportamiento benigno, es decir, un crecimiento leve y una invasividad de tejido adyacente bajo.

**ADN** Abreviación de ácido desoxirribonucleico, consiste en un ácido nucléico que contiene las instrucciones genéticas usadas en el desarrollo y funcionamiento de todos los organismos vivos.

**Angiogénesis** Proceso mediante el cual se forman nuevos vasos sanguíneos nuevos a partir de los preexistentes.

**Apoptosis** Mecanismo de muerte celular programada para controlar y frenar el crecimiento de células que tienen su código genético dañado y, en consecuencia, pueden provocar problemas en el organismo.

**Automata Celular** Modelo matemático para un sistema dinámico que evoluciona en pasos discretos en base a unas reglas locales.

**Cáncer** Nombre genérico para describir a las enfermedades que causa proliferación descontrolada de células que provocan la aparición de masas anormales.

**Carcinoma** Masa anormal que tiene un comportamiento maligno, es decir, tiene una tasa de crecimiento e invasividad alta.

**Genoma** Conjunto de genes contenidos en el cromosomas.

**Metástasis** Reproducción o extensión de una enfermedad o de un tumor a otra parte del cuerpo.

**Mitosis** Proceso de reproducción de una célula, mediante el cual, se crea una copia exacta de la misma. Puede contener daños genéticos que no se dan en la célula original.

**Mutación** Alteración repentina y permanente de la estructura genética o cromosómica de la célula de un ser vivo que se transmite a sus descendientes por herencia.

**Neoplasia** Masa anormal que aparece en alguna parte del cuerpo por el crecimiento descontrolado de células.

**Singleton** Patrón de diseño en el cual el software tiene un punto único de entrada, o una sola instancia de la aplicación.

**Telómero** Son los extremos de los cromosomas, regiones del ADN no codificante, cuyo objetivo es proteger la parte codificante para evitar que sufra daños durante el proceso de replicación.

## BIBLIOGRAFÍA

- [1] Andrew Adamatzky. *Game of Life Cellular Automata*. Springer-Verlag London Limited, 2010.
- [2] G.R. Harsh IV E.A. Chiocca A.R. Kansal S. Torquato y T.S. Deisboeck. “Cellular automaton of idealized brain tumor growth dynamics”. En: *BioSystem, Elsevier* (2000).
- [3] Douglas Hanahan y Robert A. Weinberg. “The hallmarks of cancer”. En: *Cell, Cell Press* (2000).
- [4] Douglas Hanahan y Robert A. Weinberg. “The hallmarks of cancer: The next generation”. En: *Cell, Cell Press* (2011).
- [5] Peter M.A. Sloot Jiri Kroc y Alfons G. Hoekstra. *Simulating Complex Systems by Cellular Automata*. Springer-Verlag Berlin Heidelberg, 2010.
- [6] José Santos y Ángel Monteagudo. “Analysis of behaviour transitions in tumor growth using a cellular automaton simulation”. En: *IET Systems Biology* (2014).
- [7] José Santos y Ángel Monteagudo. “Study of cancer hallmarks relevance using a cellular automaton tumor growth model”. En: *Springer* (2012).
- [8] José Santos y Ángel Monteagudo. “Studying the capability of different cancer hallmarks to initiate tumor growth using a cellular automaton simulation. Application in a cancer stem cell context”. En: *BioSystem, Elsevier* (2013).
- [9] José Santos y Ángel Monteagudo. “Treatment analysis in a cancer stem cell context using a tumor growth model based on cellular automata”. En: *PLoS ONE* (2015).