

UNIVERSIDAD AUTÓNOMA METROPOLITANA

DEPARTAMENTO DE FÍSICA



ESTUDIO DE LA ECONOMÍA Y EPIDEMIAS COMO SISTEMAS COMPLEJOS

POR:

GERARDO EHECATL ORTIZ TREJO
2153043506

CDMX, MÉXICO.

JULIO 2020

Índice general

Índice de Figuras	VI
1. Introducción	1
1.1. Sistemas Complejos	1
1.1.1. Características	1
1.2. Economía y sociedad	2
1.2.1. Economía, modelos y econometría	3
1.3. De la psicohistoria a la econofísica	4
2. Autómatas celulares	5
2.1. Autómatas celulares conceptos y definiciones	6
2.1.1. Teoría de juegos, la punta del iceberg	6
2.1.2. Definiciones	7
2.2. Maquinas de Turing	11
2.2.1. Test de Turing, la IA fuerte y una cruda realidad	11
2.2.2. Algoritmos	13
2.2.3. Tesis de Church Turing	15
2.3. Aplicaciones	16
2.3.1. Juego de la vida	16
2.3.2. Modelo (Física) de Condensación	19
2.3.3. Modelo (Sociología) de segregación	21

2.3.4. Modelo (Economía) del mercado Inmobiliario	26
3. Conclusiones y proyecciones	31
4. Códigos 1	33
5. Modelos inspirados en la mecánica estadística	59
5.1. conceptos generales	59
5.2. Loterías bilaterales	61
6. Modelos interdisciplinarios (Epidemias)	64
6.1. Revoluciones científicas, menos Popper mas Kuhn	64
6.2. Epidemias	65
6.2.1. Modelo SIR	65
6.2.2. Modelo con autómatas celulares	66
6.2.3. Modelo con movilidad	66
6.2.4. Modelo con comunidades	67
6.3. Critica y conclusiones	74
6.3.1. vida media vs probabilidad de contagio	74
6.3.2. Medidas oportunas y rigurosas (México, Europa y el mundo) .	74
6.3.3. No todo esta perdido (Cooperación y medidas de prevención) .	75
7. Conclusión	76
8. Códigos 2	77
Bibliografía	112

Índice de figuras

2.1. Alan Turing	5
2.2. Regla 90	8
2.3. Autómata1	8
2.4. Autómata2	9
2.5. Autómata3	9
2.6. Autómata4	10
2.7. Diagrama1	14
2.8. Distribución 1	16
2.9. Distribución 2	17
2.10. Distribución 3	17
2.11. Distribución 3	17
2.12. Modelo de condensación con una concentración inicial de células condensadas del 30%	20
2.13. Modelo de condensación con una concentración inicial de células condensadas del 35%	20
2.14. Modelo de Schelling en una red de 50x50 agentes, donde es favorable pertenecer a la minoría, es decir los agentes de diferente tipo al propio aumentan el nivel de felicidad. Tolerancia = 3/8.	22
2.15. Modelo de Schelling en una red de 50x50 agentes donde es favorable pertenecer a la mayoría, es decir los agentes del mismo tipo al propio aumentan el nivel de felicidad. Tolerancia = 5/8.	23
2.16. Distribución étnica en Nueva York	23

2.17. Modelo de Schelling en una red de 50x50 con cuatro tipos diferentes de agentes donde es favorable pertenecer a la mayoría, es decir los agentes del mismo tipo al propio aumentan el nivel de felicidad. Tolerancia = $5/8$	24
2.18. Modelo de Schelling en una red de 100x100.	24
2.19. Modelo de Schelling en una red de 100x100 con un afinidad al grupo propio y otro mas para cada tipo de agente.	25
2.20. Felicidad del sistema, afinidad a la mayoría	25
2.21. inmobiliario México	28
2.22. Propiedades del mercado mexicano	29
2.23. Propiedades del mercado mexicano	30
2.24. Propiedades del mercado mexicano	30
5.1. lotería bilateral	62
6.1. Evolución del autómata celular con 9500 agentes susceptibles al inicio y 20 infectados con una probabilidad de contagio de 0.125 y una vida media del virus de 14 días.	68
6.2. Evolución del autómata celular con 2400 agentes susceptibles al inicio y 16 infectados con una probabilidad de contagio de 0.05, una vida media del virus de 18 días y una movilidad de 0.05.	69
6.3. Evolución del autómata celular con 2400 agentes susceptibles al inicio y 16 infectados con una probabilidad de contagio de 0.05, una vida media del virus de 18 días y una movilidad de 0.01.	70
6.4. Evolución del autómata celular con 2700 agentes susceptibles al inicio y 40 infectados con una probabilidad de contagio de 0.1, una vida media del virus de 18 días y una movilidad de 0.05 para la comunidad negra con la comunidad gris estática.	71
6.5. Evolución del autómata celular con 2700 agentes susceptibles al inicio y 40 infectados, una vida media del virus de 18 días, una probabilidad de contagio de 0.1 para la comunidad gris, 0.05 para la comunidad negra y una movilidad de 0.05 para la comunidad negra con una probabilidad de 0.005 para la gris.	72

6.6. Evolución del autómata celular con 2700 agentes susceptibles al inicio y 40 infectados, una vida media del virus de 18 días, una probabilidad de contagio de 0.1 para la comunidad gris, 0.05 para la comunidad negra y una movilidad de 0.05 para la comunidad negra con la comunidad gris estática. 73

Capítulo 1

Introducción

1.1. Sistemas Complejos

Los sistemas complejos (SC) adolecen de una definición unívoca sin embargo si se les puede asignar ciertas características que ayuden a delimitar su estudio, como se hará más adelante.

En general podemos decir que los SC parten del estudio de grandes conglomerados de agentes que interactúan entre sí bajo ciertas reglas y condiciones específicas que dependen de la escala de observación, tanto los agentes como las interacciones se pueden presentar de un sin fin de formas. Podemos referir a un SC al estudiar un gas confinado en un recipiente, donde los agentes son las partículas del gas y las interacciones son simples colisiones, del mismo modo para un cierto mercado de valores donde los agentes pueden ser compradores, vendedores o empresas y las interacciones pueden ser un amplio abanico de relaciones económicas entre los agentes.

1.1.1. Características

Los SC suelen estar relacionados con las siguientes características:

- Escalas espaciales y temporales: Al estudiar un SC, dependerá de que escala temporal y espacial utilizemos para observar diferentes comportamientos de los cuales esperamos que sean diferentes entre sí.
- Propiedades emergentes: Se refieren a las propiedades que surgen de manera no evidente o intuitiva al cambiar de una escala a otra y que además no pueden ser explicados al menos no de forma trivial por las reglas de la escala anterior, como los cambios evidentes al pasar de estudiar un pequeño conjunto de células al estudio de un tejido formado por un conjunto mucho más grande de esas mismas células.

- Organización jerárquica : Generalmente al pasar de una escala de magnitud en el tamaño del sistema encontramos una organización que crece en complejidad para cada orden de escala.
- Contienen y pueden procesar información: Las interacciones que definen al sistema se pueden entender como ciertos paquetes de información que son intercambiados por los agentes del sistema y que con el paso del tiempo pueden ser transformados o “procesados“ en algo totalmente diferente.
- Autorregulación : Los SC poseen mecanismos de regulación que generalmente son antagónicos, lo cual a su vez puede generar estados de equilibrio dinámico donde los componentes no necesariamente tienen que estar en equilibrio por sí mismos para que el sistema se encuentre en equilibrio, generalmente los sistemas buscan romper los gradientes dentro de sí.

1.2. Economía y sociedad

Para el estudio de las sociedades contamos con tres aproximaciones conocidas, el individualismo, el globalismo (o colectivismo) y el sistemismo. El enfoque individualista dicta que una sociedad es el resultado de las interacciones que se dan entre los miembros de estas y que todas sus propiedades son resultado de estas. El globalismo asigna a la sociedad un carácter superior al de sus componentes, sus propiedades y características son propios del sistema como un todo y son ajenos a sus componentes primigenios. Los enfoques anteriores parecen complementarios entre sí puesto que cada uno estudia diferentes perspectivas de la sociedad, el enfoque sistémico toma los puntos positivos de ambos enfoques por un lado toma en cuenta las propiedades emergentes a las sociedades como el caso de la cohesión social que el enfoque individualista ignora pero además da una explicación a estos fenómenos, algo que el globalismo no puede. El enfoque sistémico entonces lo podemos resumir en una frase “el todo es más que la suma de las partes”, para el enfoque sistémico la esencia de la sociedad radica en sus componentes, sus características y propiedades pero asimismo en el número y tipo de interacciones que están presentes dentro de la sociedad. En este trabajo, nos centraremos y delimitaremos al estudio de la sociedad y la economía a través del enfoque sistémico primeramente porque es la aproximación compatible con las versiones modernas de la economía y la sociología y en segundo lugar porque esto implica a su vez que es el único enfoque compatible con la implementación de modelos matemáticos para su estudio.

1.2.1. Economía, modelos y econometría

La economía es el estudio de la riqueza , su generación y su distribución, en la sociedad los agentes de importancia para la economía son las empresas y los consumidores. El comportamiento de la economía va a estar dado por la interacción de estos dos agentes cada uno con sus objetivos y limitaciones. Las interacciones van a girar en torno a la oferta y la demanda de productos y servicios.

Con la definición anterior se podría pensar que la economía se presta un tanto sencilla y elemental sin embargo pueden existir un sin fin de diferentes tipos de empresas y de consumidores que aunque sus limitantes generales son la tecnología para suministrar la oferta necesaria y la presupuestaria para ofertar por los bienes respectivamente, si se retoma el enfoque sistémico se puede entender a cada agente como un nodo dentro de una red con conexiones infinitamente complejas esto resulta evidente si se piensa en cualquier producto o servicio y en todos los agentes (nodos) que interactuaron para poder entregar ese producto.

Ahora entonces se encuentra una barrera casi insondable, no obstante al igual que las ciencias naturales se pueden recurrir a modelos que delimiten el problema a estudiar, con las condiciones adecuadas para encontrar una solución pertinente y satisfactoria, en particular el uso de modelos matemáticos es el campo de estudio de la econometría, la econometría no es mas que el estudio a través del lenguaje de las matemáticas de la economía. El desarrollo de estos modelos no es nada trivial pues al igual que en cada rama de las ciencias presenta sus peculiaridades, para el caso de la economía sus modelos funcionan o se explican empleando dos tipos de variables diferentes, la endógenas y las exógenas, Las endógenas son las variables que el modelo se dispone a dar una explicación mientras que las exógenas son las variables que se presentan o que se integran desde afuera del modelo, pueden ser consideradas como condiciones de contorno o frontera, o parámetros que dan estructura al modelo a estudiar.

Con toda esta complejidad presentada la economía se ha centrado en estudiar casos muy específicos como es el caso del equilibrio, pero mas aun de un equilibrio estático el estudio de sistemas fuera del equilibrio o en un equilibrio dinámico comprenden la mayor complejidad dentro de este campo y es casi exclusivo de las investigaciones de frontera dentro de esta ciencia, el objetivo general de este trabajo radica precisamente en buscar nuevas aproximaciones para estudiar los casos mas complicados y complejos en la economía moderna utilizando metáforas y analogías con las técnicas propias de las ciencias exactas mas precisamente de la física y los sistemas complejos.

1.3. De la psichistoria a la econofisica

En el año de 1942 y hasta el año 1992 el afamado escritor ruso Isaac Asimov nos maravillo con su serie de relatos de la fundación, estos relatos nos plantean un futuro distópico donde el hombre ha colonizado el espacio prácticamente en su totalidad sin embargo la estructura de organización de la sociedad es muy parecida a la que existía en la época feudal, aun así goza de una aparente estabilidad, no obstante aparece un peculiar personaje, Hari Seldon, que se denomina el primer psichistoriador que mediante la técnica de la psichistoria predice que el imperio galáctico caerá en aproximadamente 500 años, por lo que se dispone a crear una fundacion que sera la encargada de preservar el conocimiento del imperio galáctico y tratar de impedir el colapso que se avecina en los próximos 500 años. El autor define a la psichistoria como una ciencia híbrida entre la psicología, la historia y la termodinámica que utiliza las descripciones individuales provenientes de la psicología, la estructura social descrita por la historia y la técnica fenomenológica provista por la termodinámica para realizar predicciones de modelos sumamente complejos como es la sociedad esto basado en el hecho de que la termodinámica brinda descripciones relativamente simples y con pocos parámetros de sistemas de N (tendiendo a infinito) componentes.

Si bien aun es imposible concebir una técnica con alcances similares a la psichistoria, la economía y la sociología moderna apuntan en sentidos similares a los que lo siguió Hari Seldon. En los orígenes de la economía y hasta mediados del siglo pasado la palabra mas utilizada en los textos era “historia“ sin embargo a partir de los años 70 comenzaron a aparecer con mayor frecuencia conceptos como “ecuación“, “axioma“ o “teorema“, se puede notar una marcada tendencia a la matematización, un proceso que han realizado prácticamente todas las ciencias incluida la física pues esta surgió como parte de la filosofía en la antigua Grecia y no fue hasta que apareció Galileo que comenzó a tomar la forma que en general ahora conocemos.

Otro aspecto importante que esta tomando importancia en las ciencias actuales es la interdisciplina, la economía como ejemplo ha decidido tomar elementos de campos tan variados como lo son la psicología con los trabajos de Daniel Kahneman por ejemplo, en los cuales integra a la descripción económica la capacidad de toma de decisiones de los agentes que participan en la economía o las matemáticas del azar provistas por los trabajos de Von Neumann y Jhon Nash. Ya analizamos a grandes rasgos que son los sistemas complejos y bosquejamos de forma muy general a la economía y la sociología por comparación resulta sumamente sugerente y pertinente utilizar las descripciones propias de los sistemas complejos en la economía debido a que la economía como tal pasa las pruebas necesarias para ser considerada como un sistema complejo ya que cumple a cabildada los cinco puntos propuestos en la primer sección. En los siguientes capítulos partiremos de problemas y técnicas ya utilizadas en los sistemas complejos y se las asignaremos a casos prácticos dentro de la economía y la sociología, en el capitulo 2 abordaremos a los autómatas celulares y estudiaremos un par de aplicaciones dentro de la economía.

Capítulo 2

Autómatas celulares

Las herramientas computacionales y de cálculo son ahora uno de los pilares de la física moderna, en particular los sistemas complejos y la física computacional muestran un avance paralelo y coexistente, propiamente son dos caras de una misma moneda.

Como se analizó en el capítulo anterior, el estudio de la economía resulta bastante complejo y requiere una capacidad de análisis y procesamiento de datos imposible de concebir si se parte del análisis íntegro de todos los agentes económicos, no obstante si apelamos a la implementación de los modelos adecuados para estudiar casos particulares y regiones bien delimitadas podemos simplificar lo suficiente el problema y con ayuda de los avances tecnológicos actuales que nos entregan herramientas de cálculo otrora inimaginables, podemos desarrollar algoritmos que no solamente nos brinden información cuantitativa sino que también nos den instrumentos para visualizar su comportamiento cualitativo. Aun para con las más sofisticadas computadoras es imperativo la elección y desarrollo de los algoritmos adecuados para optimizar el sistema y poder resolver problemas más complejos. En este capítulo invocaremos a los padres de la computación y la lógica matemática como lo son Alan Turing, George Boole, Von Neumann y Kurt Gödel, cuyos conceptos y definiciones del área en combinación con ciertos modelos económicos ya definidos servirán de base para desarrollar nuevos modelos híbridos auto-contenidos capaces de resolver y dar explicación a problemas específicos de la economía.



Figura 2.1: imagen de Alan Turing reconstruida a partir de la imagen del supercomputador Summit, usando el comando ImageRestyle de Wolfram.

2.1. Autómatas celulares conceptos y definiciones

2.1.1. Teoría de juegos, la punta del iceberg

Quizá el obstáculo mas grande presente en el estudio de la economía desde una perspectiva axiomática es la capacidad de raciocinio que poseen los agentes económicos involucrados, la frase un tanto trillada "cada cabeza es un mudo" no se aleja demasiado de lo que conocemos actualmente, de todo el universo conocido hasta ahora no se sabe de un sistema mas complejo y misterioso que el cerebro humano, tratar de entender como funciona o predecir el actuar de un hombre a través de cualquier mecanismo resulta quimérico, mas aun si tratamos de describir como interactúa con otro cerebro, al menos de forma determinista resulta inconcebible, el azar impera en nuestro accionar y en nuestras relaciones, aunque sabemos que es posible modelar el azar. En la década de los cuarenta el húngaro Von Neumann centraría su interés en el estudio de los juegos competitivos, colocando los primeros cimientos de una nueva rama de la matemática denominada teoría de juegos, en general cualquier juego se basa en la toma de decisiones de los participantes, la arena perfecta para entender como interactuamos y decidimos, si pensamos en los elementos mas simples de un juego como son el ganar y el perder, y los concebimos en el mundo de las matemáticas como elementos binarios (1,0) presentan un obstáculo menor pero con resultados sumamente interesantes, Von Neumann emplearía todo su arsenal matemático en el estudio de esta neófito rama, encontramos una aplicación inmediata a la economía en uno de sus trabajos mas importantes y celebres "Teoría de Juegos y conducta económica"¹. sin embargo con completa disimilitud al Titanic llevaría la flota científica en trayectoria de colisión con un iceberg en cuya inmensidad se puede encontrar la clave para entender y replicar lo que conocemos como inteligencia.²

¹"Titulo original, "Theory of games and economic behavior", que publicaria en 1944 junto con el economista Oskar Morgenstern.

²Evidencias científicas recientes en el área de la neurociencia muestran que el cerebro humano trabaja de forma competitiva y conflictiva, al igual que lo hacemos nosotros dentro de la sociedad, la toma de decisiones no es mas que el resultado del conflicto entre secciones antagónicas del cerebro en la basta red neuronal que se enfrentan para "tomar el control" (Ver [2]), básicamente se puede afirmar que el modelo que buscaba sortear los avatares del comportamiento humano puede ser el mismo que explique su funcionamiento y con la aplicación de las tecnologías de frontera llegar a desarrollar y mejorar a las ahora plausibles mentes sintéticas o inteligencias artificiales.

2.1.2. Definiciones

Como herramienta de modelado emplearemos a los autómatas celulares (AC), en términos simples y a groso modo los podemos definir como agentes que interactúan dentro de un medio, que son capaces de almacenar y procesar información y cuyo actuar depende del estado de sus vecinos y el suyo. Cuando Von Neumann estaba idealizando sistemas autorreplicantes se le sugirió que recurriera a emplear modelos algorítmicos y a los recientemente llegados sistemas computacionales para así finalmente llegar a los (AC).

En los sistemas computacionales de múltiples agentes se emplea la noción de autómata finito para evidenciar su funcionamiento.

“Autómata finito” *Un autómata finito recibe información y produce respuestas. Esta definido por dos listas, cuya riqueza determina la capacidad de pensamiento del sistema.* (Daniel Heymann et al., 2013: 57).

- $\{S_j\}$: Este conjunto reúne los posibles estados internos de la maquina
- $\{T_{i,j}(J)\}$: Los elementos del conjunto indican las transiciones posibles entre los estados del autómata. La transición $\{T_{i,j}(J)\}$ del estado interno $\{S_j\}$ esta determinada por la información J que recibe el autómata.

Un autómata finito es el mínimo elemento que compone al autómata celular, el autómata celular estará compuesto por una distribución de estos elementos, cada uno de ellos tiene la capacidad de intercambiar información con sus vecinos, a partir de esta información el autómata tomara un estado definido, entre mas y mas complejas conexiones pueda realizar cada autómata finito mas capacidad de procesamiento tendrá el sistema, la definición de las vecindades (es decir, con que otros autómatas tiene permitido interactuar dentro de la red) y el tipo de interacción que pueden realizar definirá el comportamiento y las capacidades del sistema (AC).

Los autómatas celulares son sistemas que evolucionan con el tiempo, de una manera mas formal están determinados por las regla de evolución de sus estados, esta regla es fija a lo largo del tiempo, esta evolución puede ser síncrona si todos los estados evolucionan al mismo tiempo o asíncrona si cada célula cambia de manera independiente al de sus compañeras, el estado del autómata celular se definirá entonces como el conjunto que alberga a cada uno de los estados de las células que integran el sistema.

$$E_t = \{S_i, S_{i+1}, \dots, S_N\} \quad (2.1)$$

Cada estado del autómata celular E_t puede cambiar con el tiempo de forma determinista si a un tiempo $t+1$ solo hay un estado posible de evolución o de forma aleatoria si la transición entre estados esta definido por una distribución de probabilidades de ocurrencia, también existe el caso totalistico donde el estado en cada sitio (célula) no depende del estado de sus vecinas en si si no de la suma de los estados de estas e incluso el suyo también.

Ejemplos

Para entender mejor, las diferencias entre estos dos modelos se pueden contrastar ambos caso, en el primer ejemplo se observa como funcionan los autómatas deterministas, un AC unidimensional cuyas reglas de evolución estarán dadas por la aplicación de la función lógica XOR entre las células y sus vecinas mas cercanas.

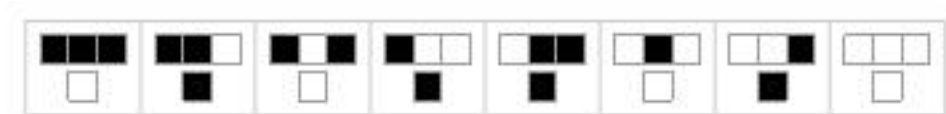


Figura 2.2: Representación gráfica de la evolución de estados obedeciendo la función XOR.

Cada celda representa una célula, si esta encendida o pintada de negro le podemos asignar el valor binario 1, por su parte si esta apagada es cero, de esta forma podemos realizar las operaciones lógicas deseadas para las células, en este caso la información que pueden intercambiar las células es simplemente el estado binario en el que se encuentran, dicha información solo la pueden compartir con sus vecinas temporales mas cercanas, si se toma la condición XOR(condición 90 en Mathematica) y se aplica a un estado inicial (1,0,1) célula encendida, la evolución del sistema estará representado gráficamente de la siguiente manera.

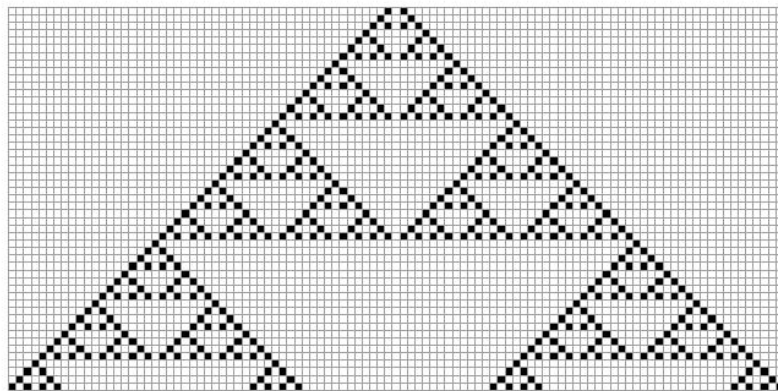


Figura 2.3: Evolución del sistema obedeciendo la función XOR, por un periodo de tiempo de 50 pasos.

Para la distribución aleatoria se utiliza la misma condición empleando la función lógica XOR pero ahora la función solo determina el 90% los casos, mientras que el otro 10% sera definido por una función booleana diferente de la siguiente manera .

$$S_i(t+1) = \begin{cases} XOR(S_{i-1}(t), S_i(t), S_{i+1}(t)) & \text{probabilidad } 0,9 \\ XOR(S_{i-1}(t), OR(S_i(t), S_{i+1}(t))) & \text{probabilidad } 0,1 \end{cases} \quad (2.2)$$

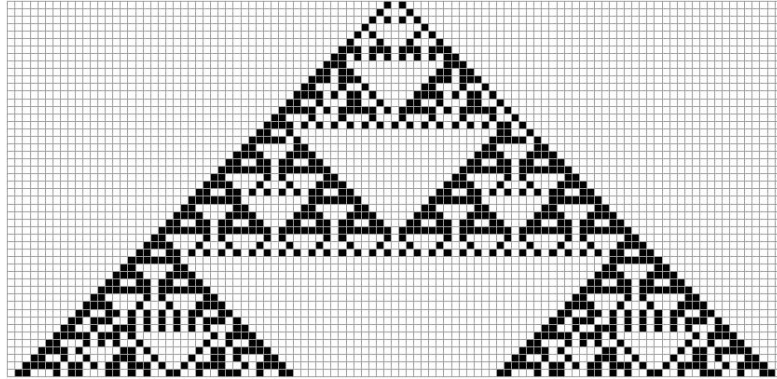


Figura 2.4: Evolución del sistema obedeciendo la función 2.2, por un periodo de tiempo de 50 pasos.

Si se modifican los porcentajes que definen al autómata, el comportamiento se ve alterado de manera evidente.

$$S_i(t+1) = \begin{cases} XOR(S_{i-1}(t), S_i(t), S_{i+1}(t)) & \text{probabilidad } 0,8 \\ XOR(S_{i-1}(t), OR(S_i(t), S_{i+1}(t))) & \text{probabilidad } 0,2 \end{cases} \quad (2.3)$$

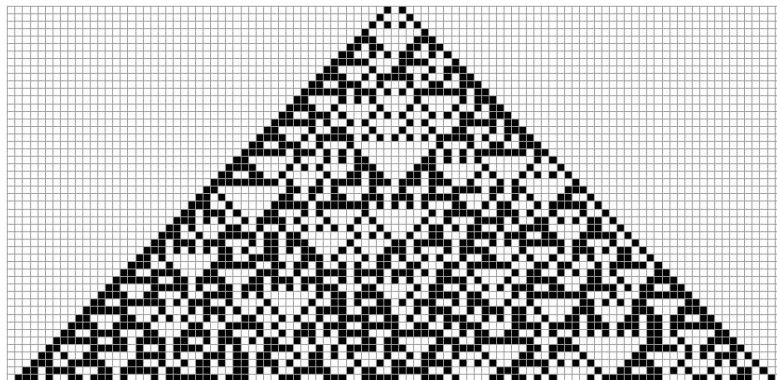


Figura 2.5: Evolución del sistema obedeciendo la función 2.3, por un periodo de tiempo de 50 pasos.

El carácter estocástico de este nuevo sistema resulta evidente en comparación con el sistema anterior si ahora volvemos a correr el sistema siguiendo la función 2.3, con las mismas condiciones iniciales. A diferencia del autómata determinista, que no importa cuantas veces reinicie el sistema, mientras conserve las mismas condiciones iniciales el comportamiento es el mismo, el autómata estocástico varía su comportamiento considerablemente cada vez que se reinicia el sistema.

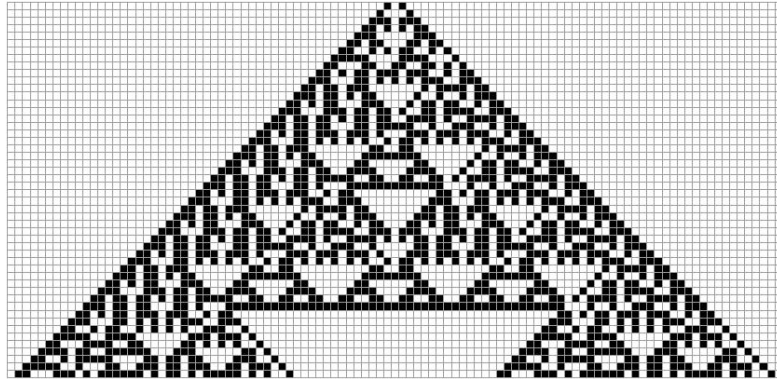


Figura 2.6: Evolución del sistema obedeciendo la función 2.3, por un periodo de tiempo de 50 pasos, por segunda ocasión.

En ambos casos, el desarrollo del sistema es sumamente sensible a los cambios en las condiciones iniciales, ya observamos las diferencias entre el modelo determinista y el estocástico sin embargo ambos comparten una imposibilidad de calculo o resolución de forma analítica, es decir para ciertas reglas, aunque el sistema sea determinista, no existe medios analíticos para decir por ejemplo si la condición inicial es tal, cual sera el estado del sistema después de 100 periodos de tiempo, solo podemos conocer el estado 100 si dejamos evolucionar al sistema por ese periodo de tiempo, por razones obvias el modelo estocástico obedece este fenómeno para cualquier regla.

En general un sistema de autómatas celulares, puede evolucionar de diferentes formas, pueden evolucionar desde estados de equilibrio u homogéneos hasta estados sumamente complejos y caóticos, como en el caso de los ejemplos anteriores. En la teoría de juegos los procesos estocásticos tienen una particular importancia pues en cada toma de decisiones existe un cierto nivel de aleatoriedad, en general las características caóticas serán vitales en el estudio de modelos económicos porque a nivel del experimento es el comportamiento observado y por lo tanto en la teoría es un comportamiento que se espera este presente.

2.2. Maquinas de Turing

Como analogía para estudiar el comportamiento de la economía, los autómatas celulares funcionan casi perfectamente, pues en esencia el comportamiento de los agentes económicos está descrito por un conjunto de autómatas finitos que están interactuando entre sí, no obstante hasta ahora las posibles interacciones que pueden realizar o que se han propuesto para estos sistemas no parecen ser suficientes, si suponemos que dentro de la economía cada agente goza de libre albedrío el modelo del autómata celular será el ideal pues si consideramos el suficientes autómatas diferentes y lo suficientemente complejos llegaríamos a una copia casi totalmente fiel de la dinámica económica, sin embargo el modelo no está completo, de la sociología en general se puede extraer que el comportamiento de la sociedad y el de la economía no está definido de esa forma, cada agente no es libre de ejercer su libre albedrío en su totalidad, cada agente está restringido y alienado a desempeñar un papel específico dentro de esta.

2.2.1. Test de Turing, la IA fuerte y una cruda realidad

Como se analizara en esta sección a partir de una posible respuesta del problema de lo indecible de Hilbert, Alan Turing cimentó las bases de la computación, no solo eso si no que además fundó el estudio de lo que hoy se conoce como inteligencia artificial, una máquina universal de Turing (que se analizara a fondo más adelante) es una aún mejor analogía de como se constituye y conforma una sociedad y la economía en general, una máquina universal de Turing está compuesta de un hardware y un software, el software es el que contiene las instrucciones o los algoritmos que debe ejecutar el hardware, una máquina de Turing va a estar caracterizada en su totalidad por el software que es capaz de ejecutar, sin importar que en esencia el hardware diste mucho en composición o forma de otro hardware, mientras sean capaces de ejecutar los mismos algoritmos se dirá que son máquinas semejantes.

Turing propuso que una máquina lo suficientemente sofisticada sería capaz de igualar o imitar la inteligencia humana, para lo cual propuso su famoso test que consiste en una serie de preguntas donde un aplicador (humano) al recibir las respuestas de la máquina determinara si está conversando con una máquina o con un ser humano. De este sencillo test se ha librado casi por 100 años un debate y una fuerte controversia acerca de lo que definimos como inteligencia, de si podemos considerar a una máquina como un ente consciente, un debate tanto de carácter filosófico como técnico que acertó otro golpe al estatus privilegiado del que se ha dotado a los seres humanos en el cosmos.

En este encarnizado debate existe la postura de la IA (inteligencia artificial) fuerte que nos dice que al igual que el concepto de maquina universal de Turing que los componentes de una maquina en si son irrelevantes mientras sea capaz de ejecutar el software deseado, en este caso especifico algo sera consciente mientras sea capaz de ejecutar el software de “consciencia“ sin importar si lo ejecuta en una estructura a base de Carbono o de Silicio. Este enfoque asigna una definición concisa de lo que es la inteligencia, inteligencia bajo este enfoque es toda respuesta a un estimulo, entonces todo lo que responde a un estimulo es inteligente pero el nivel de inteligencia depende de la cantidad de estímulos a los que puede reaccionar y a la complejidad de sus reacciones, entonces aunque bajo este enfoque tanto una planta como un ser humano son seres inteligentes porque la planta puede reaccionar a ciertos estímulos como lo son temperatura, humedad o iluminación no es ni de cerca una inteligencia equiparable a los miles de estímulos a los que es capaz de responder un cuerpo humano.

Bajo la definición dada por la IA fuerte, la economía y la sociedad son entes inteligentes que son capaces de responder a diversos estímulos, desde tiempos de Thomas Hobbes (1588-1679) ya se hablaba de la sociedad como un ente con vida y consciencia, con su ya legendario leviatán Hobbes postula una imagen onírica propia de la literatura de Lovecraft de lo que es la sociedad, un ser de proporciones descomunales que posee consciencia y voluntad propia, dicta que entenderlo en toda su extensión es un acto infructuoso, mas adelante pensadores como Foucault retoman estas ideas. Foucault tiene toda una teoría de la normalización y la alienación de la sociedad en particular analiza el panóptico, que son todas las estructuras que sirve para regular, vigilar y corregir a los agentes que integran la sociedad, estos conceptos podrían ser considerados como fenomenológicos, a grandes escalas son aceptables y entendidos pero carecen de una descripción fundamental desde los estratos mas bajos de organización. Al estudiar a la sociedad y la economía a partir del concepto de maquina universal de Turing presenta estos conceptos como algo mas orgánico, tiene sentido hablar de estructuras que se encargan de autorregularse, términos como los tan utilizados autorreguladores automáticos en la macroeconomía escalan sus dimensiones de importancia, se encuentra en este punto del trabajo la primera aportación importante en el aspecto teórico de una aproximación a partir de la econofísica para entender las estructuras clásicas dadas en la economía.

Estudiar estas aportaciones de forma detenida no son prioridad del presente trabajo sin embargo es importante hacer este paréntesis para resaltar que las bondades de la interdisciplina y el uso en general de la física en pro de otras áreas va mas allá de una herramienta que ayude a librar obstáculos de carácter operacional o técnico, por ejemplo con esta introducción para la sección, donde en unas pocas lineas se han entremezclado cuestiones teóricas de la computación y la sociología, podemos plantearnos la siguiente conclusión que parece ser evidente y que de igual forma tiene sus contra partes desde un punto de vista meramente sociológico, si la economía como tal es un ente inteligente caracterizado además empleando la IA fuerte, los agentes que la componen son prescindibles y en términos generales irrelevantes mientras se siga corriendo el algoritmo predeterminado no importa en términos del conjunto que

sucede con las partes, desde el punto de vista económico los pobres o todos los agentes debajo de un cierto estatus económico carecen de una huella o identidad dentro del modelo, premios Nobel como Stiglitz estudian esta cuestión con críticas y teorías sobre la desigualdad, por su parte en el aspecto social en específico en países como México, notamos una marcada tendencia por la deshumanización y la despersonalización de la población al considerar a víctimas, muertos y demás agraviados meramente como números dentro de una lista con elementos intercambiables y reemplazables.

2.2.2. Algoritmos

En forma sumamente simplificada se puede decir el décimo problema de Hilbert sobre lo indecible consiste en buscar los elementos primigenios y elementales de los que pueden estar constituidas las matemáticas, Hilbert pretendía sentar unas bases sólidas para la matemática a través de axiomas elementales sobre los cuales se pudieran derivar las matemáticas o al menos ciertas clases definidas. Para atacar este problema Alan Turing lo abordó desde el punto de vista del cómo se razona empleando como base suponemos los estudios previos sobre el razonamiento humano y la noción que ya existía de la mente humana como una máquina capaz de realizar cálculos, al entrar la mente y su funcionamiento al escenario es necesario definir y estandarizar el concepto de mente o de máquina pensante.

Turing definió entonces la hoy conocida máquina de Turing, esta máquina no es en esencia un concepto de una máquina como tal tangible con sus elementos como las computadoras que conocemos hoy, aunque su descripción y elementos aplican casi a la perfección a los sistemas computacionales modernos la mayoría de los cuales están contruidos a partir de estos conceptos dados por Turing. El concepto original es más bien una representación del proceso de cálculo o pensamiento, la máquina de Turing es entonces el conjunto de pasos o instrucciones necesarias para realizar algún procedimiento, un algoritmo, en su teoría de números computables Turing busca encontrar los axiomas elementales o los pasos básicos que debe de seguir una máquina para realizar en general cualquier cálculo o resolver cualquier problema. Si una máquina es capaz de resolver cualquier problema siguiendo estos pasos elementales con cierto orden en un tiempo finito entonces de alguna forma se encontraría una posible solución al problema de Hilbert, la máquina de Turing universal es precisamente este concepto ideal que debe de poseer ciertas características, la analogía que encuentra Turing para describir el proceso es el de una cinta infinita que está siendo leída y escrita en todo momento, esta cinta está compuesta o delimitada por casillas donde almacenaría un único elemento, cada vez que la cinta avanza una casilla en alguna dirección (por convención y facilidad entiéndase derecha o izquierda) decimos que está llevando a cabo un proceso mecánico, aunque la cinta es de tamaño infinito la cantidad de información que contiene debe de ser finita, cada vez que se escribe algo sobre la cinta decimos que se está modificando su estado interno, el número de estados internos o pasos a seguir debe de ser finito (considerando que el problema debe ser resuelto en un tiempo menor a infinito).

Es conveniente para entender mejor el posible funcionamiento de estos sistemas mecánicos ideales emplear los elementos ya estandarizados en la computación moderna, primeramente si bien se ha dicho que en cada casilla puede escribirse un dígito o símbolo cualquiera pero limitado a uno es conveniente desde ahora invocar la notación binaria primeramente porque es el lenguaje que habla ahora las computadoras y en segundo lugar porque es el lenguaje que se menciona en secciones anteriores habla la teoría de juegos o la toma de decisión. La notación binaria tiene la facilidad de poder contener primeramente todos los números y caracteres que se deseen simplemente reservando una secuencia de '1' y '0' para dicho carácter, de igual forma podemos reservar secuencias que signifiquen procedimientos o pasos en concreto a seguir, otra bondad es que de forma orgánica los elementos '1' y '0' sumamente factibles para operar bajo las reglas del álgebra booleana que en esencia son las reglas de operación de la lógica cuyos orígenes datan de la época Aristotélica. Ahora bien tenemos el concepto de maquina universal de Turing como una secuencia casi infinita de '1' y '0' lo cual no dice mucho, no obstante este es uno de esos famosos conceptos que se dicen mas sencillos de entender que de tratar de explicar, sera mas fácil su comprensión si analizamos una representación gráfica mas intuitiva de lo que significa un algoritmo.

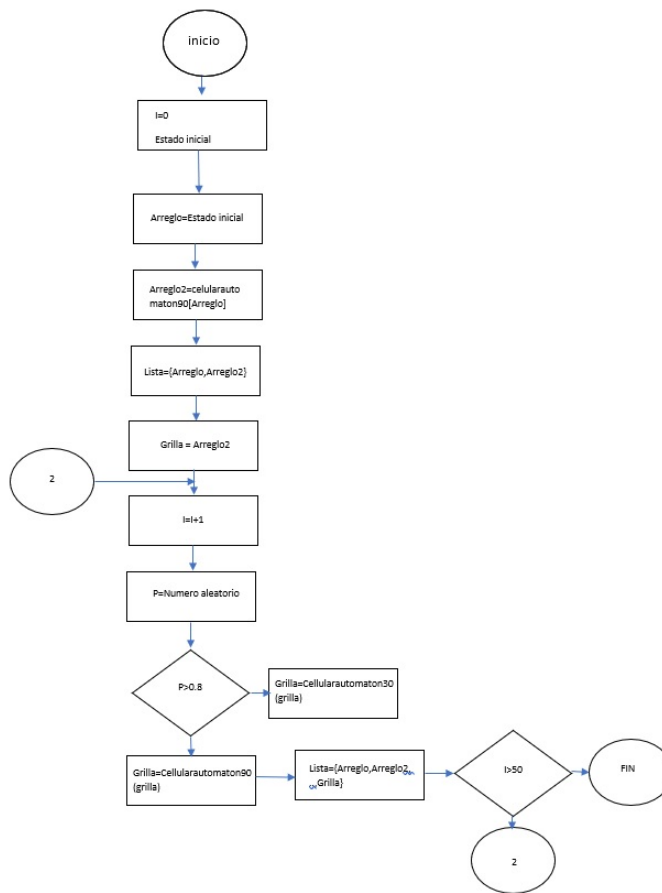


Figura 2.7: Diagrama de flujo del programa mostrado en la sección anterior, para generar la evolución del autómata celular.

En la figura 2.7 se observa una representación mas familiar de lo que significa algoritmo, esta representación es denominada diagrama de flujo, la generación del programa en cuestión no es mas que la maquina en turno realizando los pasos mostrados en la figura siguiendo la secuencia y la recursividad mostrada según los casos, esta representación es muy simplificada y no muestra en realidad que cada uno de los procesos (denotados como rectángulos en la figura) o tomas de decisión (rombos) ya implican un algoritmo que puede ser igual o mas complicado que el mostrado, en función de lo que implica la lectura y escritura de '1' y '0' pero brinda una idea clara de lo que significa el concepto de algoritmo , no obstante que esta sea una traducción del lenguaje maquina (binario) al lenguaje humano o al lenguaje de la lógica humana.

2.2.3. Tesis de Church Turing

Para definir de una manera lo suficientemente formal de lo que es una maquina de Turing para el presente trabajo, podemos recurrir a la tesis de Church Turing, esta tesis nos dice que cualquier proceso considerado como proceso efectivo puede ser ejecutado por una maquina de Turing, de esta forma el concepto de maquina de Turing queda definido por el concepto de procedimiento efectivo y este a su vez esta determinado por el concepto de maquina de Turing. Una maquina debe constar de los siguientes elementos.

- Programa o algoritmo.

- Numero finito de estados.

- Memoria.

- Cabeza de lectura y escritura.

De los elementos anteriores cabe destacar la importancia, ya remarcada en los párrafos anteriores, del programa o algoritmo, incluso en la actualidad se podría asociar el desarrollo en la computación únicamente al incremento en la velocidad de calculo de los procesadores o el numero de estos que se pueden conectar para que trabajen de forma paralela sin embargo la programación y el desarrollo de nuevos algoritmos ha jugado un papel crucial en los últimos años y en particular es el principal campo de interés del presente trabajo.

2.3. Aplicaciones

2.3.1. Juego de la vida

El juego de la vida de Conway presenta una analogía prácticamente perfecta de los conceptos que se han presentado, el "juego" consiste en una red bidimensional (generalmente, aunque existen modelos 3D), en cada nodo o casilla de la red existe una célula cuyo estado estará determinado de forma binaria por un "1" o '0', 1 representa que la célula esta viva mientras que 0 representa que la célula esta muerta, una célula sobrevive si tiene dos vecinas vivas, muere si es una o ninguna y una célula puede nacer si tiene tres vecinas vivas. El sistema evolucionara de forma síncrona para cada paso de tiempo siguiendo las reglas establecidas. El juego de la vida en si representa una maquina de Turing universal, si como condiciones iniciales se emplean las distribuciones indicadas el sistema puede imitar todas las maquinas de Turing, por otra parte como primer ensayo en busca de un modelo consistente para la economía es excepcional pues es como una versión muy simplificada de "la vida" o en este caso de la economía, cada célula sigue reglas establecidas bastante simples, pero en conjunto las distribuciones y algoritmos que se pueden ejecutar son prácticamente infinitos.

Podemos partir de diferentes patrones, cada uno nos entregara comportamientos distintos con sus propias peculiaridades como en los siguientes casos.

Distribución 1 (semáforo)

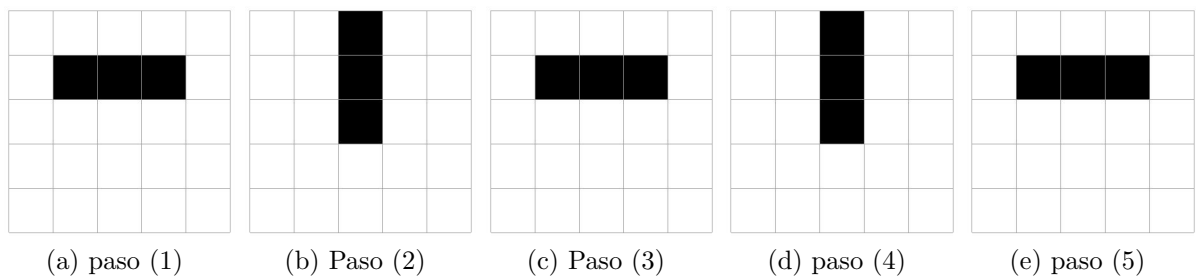


Figura 2.8: Distribución 1

Distribución 2 (Punto estacionario)

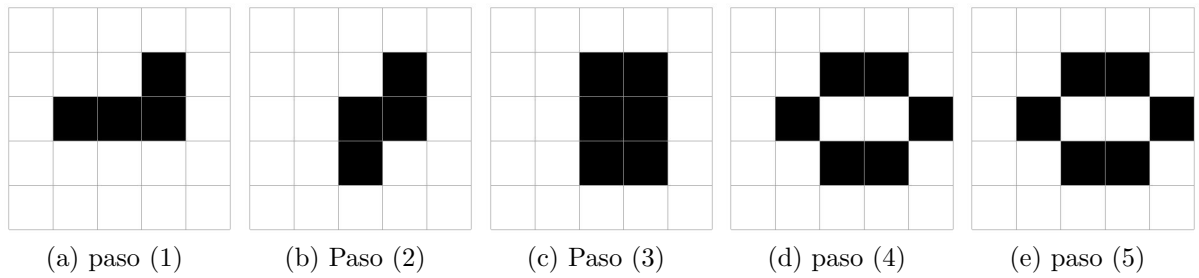


Figura 2.9: Distribución 2

Distribución 3 (Trineo)

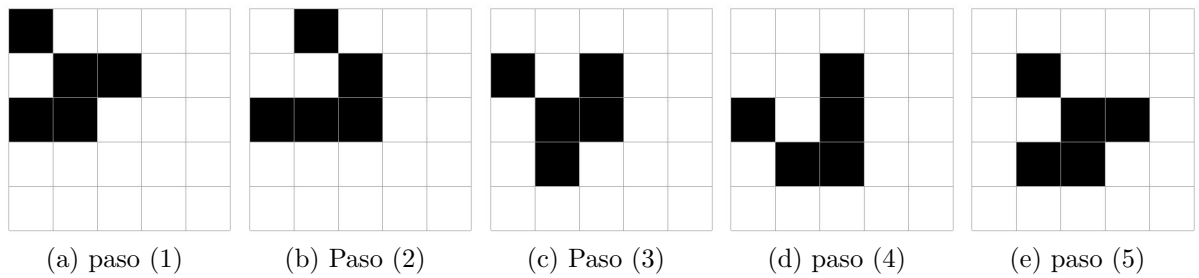


Figura 2.10: Distribución 3

Distribución 4 (Aleatoria)

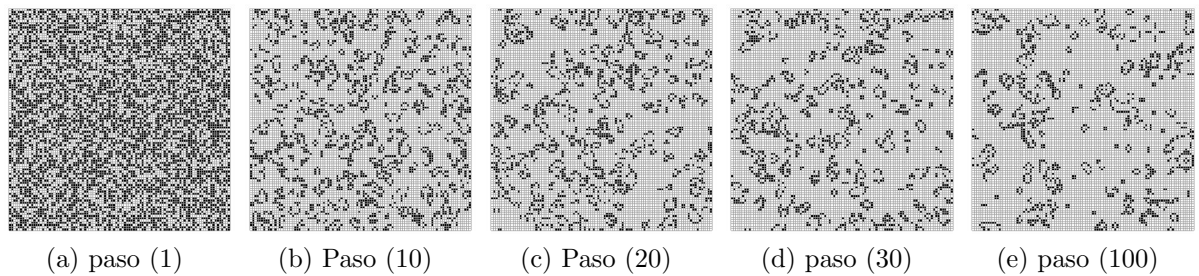


Figura 2.11: Distribución 3

En términos generales el sistema denominado como juego de la vida de Conway es una maquina universal de Turing, si se coloca la distribución adecuada en las condiciones iniciales del sistema se puede recrear cualquiera de las maquinas de Turing. En las distribuciones anteriores tenemos cuatro ejemplos muy diferentes del comportamiento que puede tener el sistema. En la primer distribución (figura 2.8) tenemos un comportamiento repetitivo y periódico, en este caso el sistema solo contara de dos estados diferentes que se repetirán de forma intermitente. Para la segunda distribución (figura 2.9), encontramos cuatro estados muy diferentes entre si, el sistema los "recorre" en orden sin embargo al llegar al cuarto estado el sistema encuentra a un equilibrio y no modifica mas su estado. La tercer distribución (figura 2.10) presenta una situación sumamente interesante, tenemos dos configuraciones locales de células que van a permanecer inalteradas para los pasos pares y para los pasos nones, es decir todos los pasos nones tendrán la misma distribución local de igual forma para los nones sin embargo esta distribución se va a desplazar, se puede observar que cada cuatro pasos la distribución se desplaza exactamente un cuadro en dirección diagonal. Finalmente la ultima distribución (figura 2.11), nos presenta un estado inicial aleatorio, conforme el sistema evoluciona podemos observar patrones o comportamientos que esperaríamos observar en un microscopio que analiza alguna colonia de bacterias, es en estas distribuciones donde se aprecia de mejor forma el porque resulta pertinente el nombre de "juego de la vida" para este sistema.

En su libro "El gran diseño" el profesor Stephen Hawking menciona este sistema en su capitulo el milagro aparente precisamente para construir una argumentación en torno al hecho de que la vida es complicada sin embargo no tanto como esperaríamos, a partir de unas reglas de interacción sumamente sencillas se encuentran patrones y comportamientos sumamente complejos e interesantes. Podemos partir de este hecho y tratar de modelar sistemas sociales y económicos a partir de reglas sencillas de interacción y tratar de argumentar de la misma forma que lo hizo Stephen Hawking la desmitificación de la aparente infranqueable complejidad intrínseca a la economía y la sociedad. En los modelos subsecuentes partiremos de esta premisa, en el primer caso plantaremos un sistema en extremo sencillo que sin embargo describe de forma fenomenológica como se comporta un fenómeno físico estudiado desde esta perspectiva posteriormente se estudiara el modelo de segregación poblacional de Schelling para observar como decisiones individuales sobre gustos o afinidades a ciertos grupos moldean nuestras comunidades y finalmente se integrara el factor monetario para ver como afecta la distribución y disponibilidad de dinero para que los individuos tomen sus decisiones y a nivel macro como se ve reflejado en la sociedad en cuestión.

2.3.2. Modelo (Física) de Condensación

Se supone una superficie que tiene las condiciones necesarias para que una cierta sustancia se encuentre tanto en su estado líquido como en su estado gaseoso, ahora sobre dicha superficie se pueden tener diferentes porcentajes de la sustancia en estado líquido y se necesita estudiar que sucederá con el sistema después de x cantidad de pasos, tanto en la concentración de sustancia en estado líquido como del tamaño que tendrán las "gotas" de la sustancia en el sistema.

Para modelar este fenómeno podemos utilizar el modelo anterior pero con unas ciertas modificaciones, se definen entonces la estructura que debe tener el sistema y el algoritmo que se seguirá para modelar de forma adecuada el sistema.

Arquitectura

- El sistema tendrá una geometría propia del toro, con la finalidad de poder emplear condiciones de frontera periódicas y facilitar el computo del sistema.
- Cada célula estará caracterizada por "0" y "1" donde cero representa que la sustancia esta en estado gaseoso y el uno en estado líquido.
- La vecindad de cada célula esta compuesta por los ocho vecinos mas cercanos.

Dinámica

- El sistema evolucionara de forma síncrona es decir cada una de las células cambiara de estado al mismo tiempo.
- Cada célula calcula cuantos vecinos tiene condensados incluido la misma célula.
- La célula cambia de estado, $(0 \rightarrow 1)$ se condensa, si en su vecindad hay cuatro o mas células condensadas.
- La célula cambia de estado, $(1 \rightarrow 0)$ se evapora, si en su vecindad hay menos de cuatro células condensadas.

Los resultados para una red bidimensional de 100x100 elementos son los siguientes:

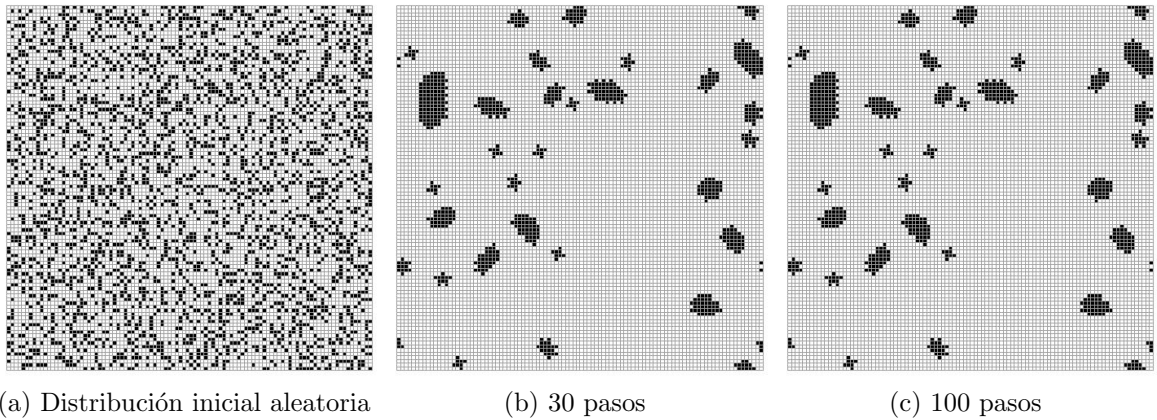


Figura 2.12: Modelo de condensación con una concentración inicial de células condensadas del 30 %

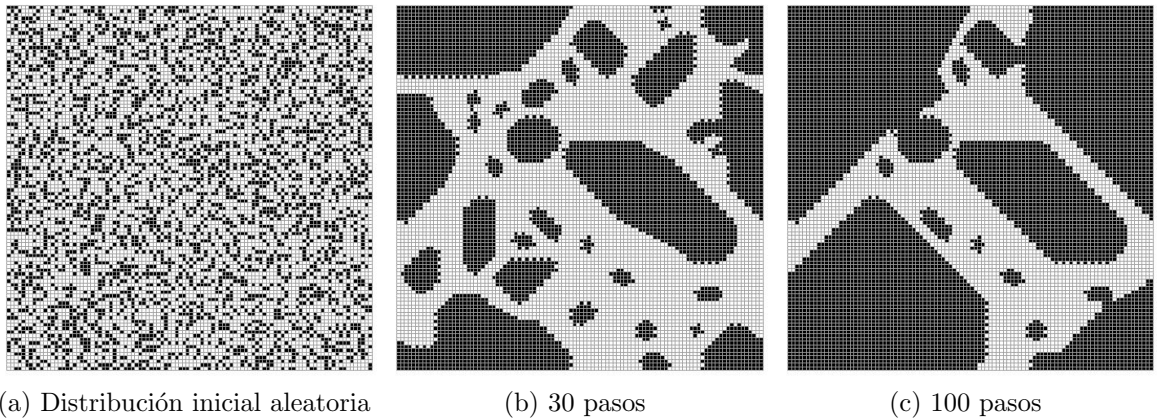


Figura 2.13: Modelo de condensación con una concentración inicial de células condensadas del 35 %

Como se puede observar en las figuras anteriores, el sistema es sensible a los cambios en la concentración de partículas o células condensadas. A una concentración del 30 % el sistema llega rápidamente a un estado estacionario con algunas gotas de tamaños relativamente pequeñas, con solo un 5 % de incremento en la concentración, al 35 %, el comportamiento cambia totalmente, ahora el sistema no llega a un estado estacionario tan rápido y la condensación del sistema es prácticamente absoluta a los 100 pasos temporales.

2.3.3. Modelo (Sociología) de segregación

El enfoque sistémico en general implica partir del estudio de los individuos, sus decisiones, afinidades y aversiones e incluir sus interacciones entre individuos para modelar a la sociedad. El modelo de segregación de Schelling utiliza este enfoque para partir de reglas de interacción sencillas para modelar y describir procesos que en conjunto presentan una complejidad importante.

Este modelo de segregación emplea un sistema para cuantificar el grado de comodidad o afinidad de algún individuo a su entorno, con base a este grado de afinidad el individuo puede decidir mudarse, el grado de afinidad puede ser modelado de diferentes formas y obedeciendo diferentes dinámicas. El principio en general consiste en sondear cuantos vecinos tiene un individuo que pertenece a algún grupo étnico, racial o social que sean de su grupo afín lo que incrementara sus puntos de felicidad o afinidad, también existe la posibilidad de incluir grupos aversivos que pueden o no contribuir a su grado de afinidad o incluso restar puntos de felicidad, los individuos afines al individuo en cuestión no necesariamente son los de su mismo grupo, puesto que el individuo puede sentir una afinidad por ejemplo a ser parte de la minoría. A partir del sondeo de sus vecinos cada individuo , puede mudarse si su grado de felicidad no cubre un cierto rango, para que pueda realizar la mudanza cada individuo inconforme debe encontrar a otro individuo en su misma situación para que puedan intercambiar lugares. De forma mas explicita el sistema se modela de la siguiente manera.

Arquitectura

- El sistema tendrá una geometría propia del toro, con la finalidad de poder emplear condiciones de frontera periódicas y facilitar el computo del sistema.
- Cada agente estará caracterizado por un cierto numero de símbolos numéricos o alfabéticos que representan su pertenencia al grupo caracterizado por dicho símbolo.
- La vecindad del agente esta compuesta por los ocho vecinos mas cercanos, aunque esta puede ser variable para ciertos casos en especifico.

Dinámica

- El sistema evolucionara de forma asíncrona.
- Se eligen dos agentes del sistema de forma aleatoria.

- Cada agente realiza el calculo de su nivel de felicidad y lo compara con su tolerancia para determinar si es "infeliz" en su estado actual.
- Si ambos agentes son infelices, se lleva a cabo la mudanza.
- El sistema debe poseer un mecanismo de halting, es decir se debe detener cuando en un periodo considerablemente grande ya no pueda realizar mudanzas o ya haya realizado un cierto numero critico para el sistema .

Resultados

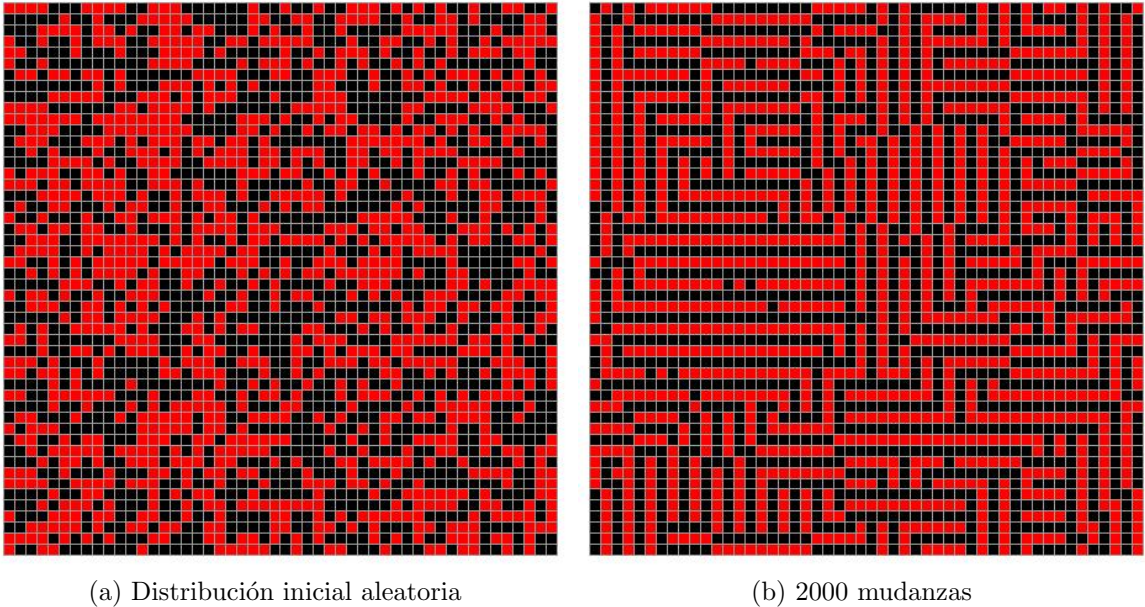
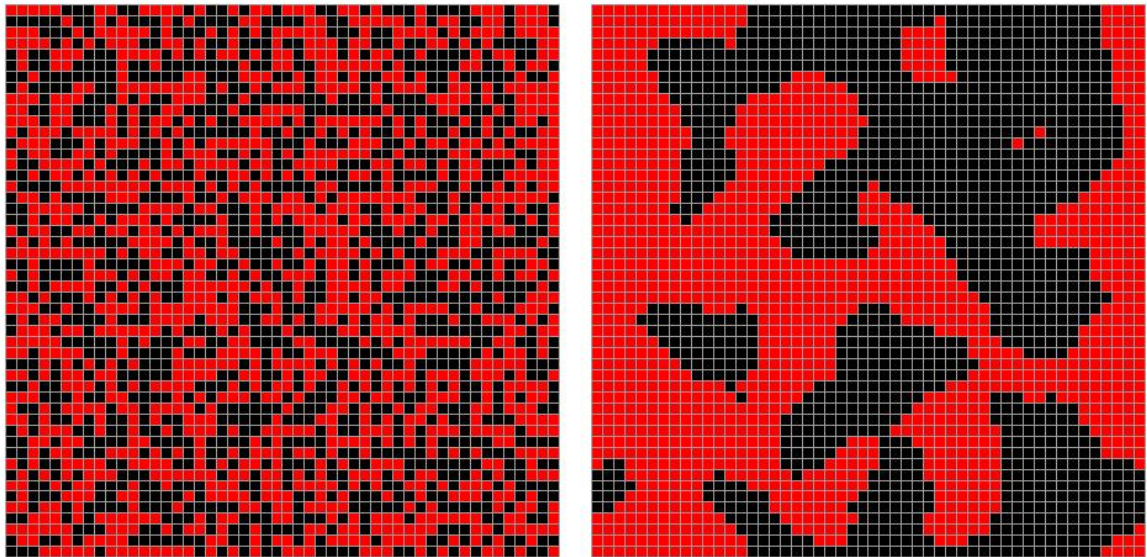


Figura 2.14: Modelo de Schelling en una red de 50x50 agentes, donde es favorable pertenecer a la minoría, es decir los agentes de diferente tipo al propio aumentan el nivel de felicidad. Tolerancia = $3/8$.



(a) Distribución inicial aleatoria

(b) 2000 mudanzas

Figura 2.15: Modelo de Schelling en una red de 50x50 agentes donde es favorable pertenecer a la mayoría, es decir los agentes del mismo tipo al propio aumentan el nivel de felicidad. Tolerancia = $5/8$.

En las imágenes anteriores se puede observar como criterios muy simples para el actuar de cada uno de los agentes resulta en sistemas que parecen que se organizan de forma inteligente sin embargo cada uno de los agentes ignora el actuar de los demás agentes. En este caso solo existe una dinámica que impera en el sistema, en el primer caso tanto los agentes negros como los rojos tienen preferencia por pertenecer a la minoría, ambos tienen una tolerancia de $3/8$ es decir a lo más toleran tres vecinos de su misma especie. En el segundo caso ambos desean pertenecer a la mayoría y tienen una tolerancia de $5/8$ es decir a lo menos necesitan tener 5 vecinos iguales a ellos para estar conforme.

Dada estas dos simples dinámicas cabe preguntarse si alguna de ellas predice la forma en que las comunidades se distribuyen según sus grupos étnicos, si analizamos la siguiente imagen (figura 2.16) podemos observar que resulta mucho más pertinente y acertado el segundo modelo, es decir en términos generales preferimos formar parte de la mayoría. Si ahora agregamos más agentes que obedezcan esta dinámica podemos observar patrones cada vez más parecidos a los de esta imagen.

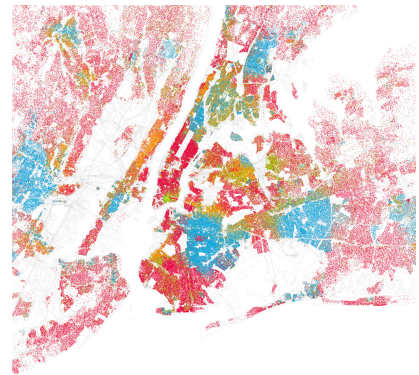


Figura 2.16: Imagen de la distribución étnica en Nueva York 2016.

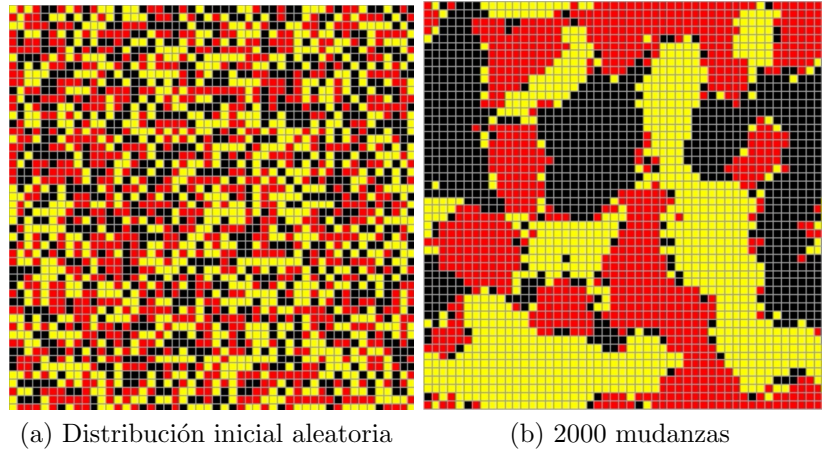


Figura 2.17: Modelo de Schelling en una red de 50x50 con cuatro tipos diferentes de agentes donde es favorable pertenecer a la mayoría, es decir los agentes del mismo tipo al propio aumentan el nivel de felicidad. Tolerancia = $5/8$.

En esta nueva dinámica entre cuatro tipos de agentes diferentes observamos una estructura mas similar a la figura 2.16, podemos suponer que si aumentamos el tamaño de la red e incluimos mas tipos de agentes las estructuras se van a guardar mas similitudes sin embargo resulta evidente que en la realidad no todos tienen la misma tolerancia o incluso aunque el pertenecer a la mayoría resulta ser la dinámica predominante, no necesariamente debe ser la única.

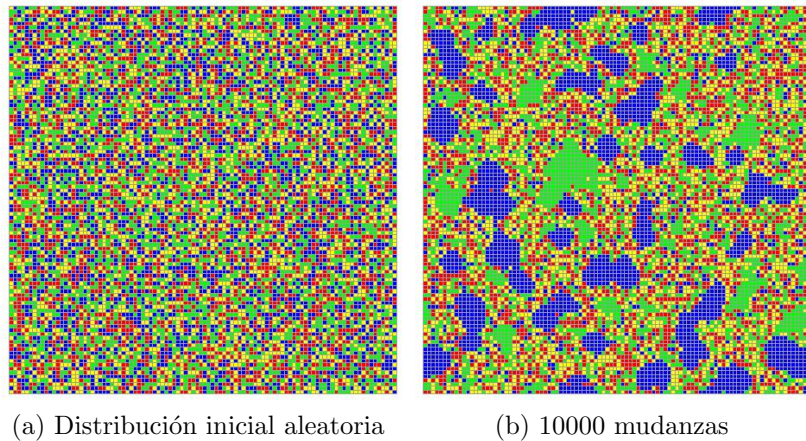


Figura 2.18: Modelo de Schelling en una red de 100x100.

El ultimo par de imágenes representa la complejidad en la organización que puede alcanzar este modelo de sociedad mezclando reglas de interacción sencillas. En particular este ultimo sistema mezcla a grupos que obedecen la primera y la segunda dinámica y con diferentes tolerancias resulta evidente que agentes verdes y azules obedecen la dinámica dos.

Nuevamente se puede suponer que si la geometría del sistema es correcta y si se censan de forma adecuada los gustos de la población este modelo puede predecir a casi cabalidad como se estructuraría una sociedad hipotética de un grupo de individuos recién llegados y distribuidos al azar. Quizá el aspecto mas importante de este modelo sea el inverso, a partir de imágenes como la de la figura 2.16 se pueden correr diferentes simulaciones del sistema variando los parámetros de afinidad y de vecindad de los agentes para encontrar el que mas se aproxime y dar respuestas sobre las características de la población real haciendo la analogía con el sistema simulado.

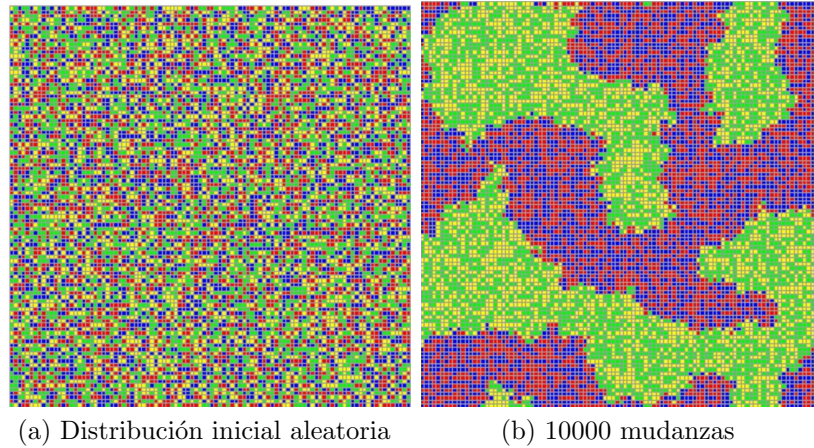


Figura 2.19: Modelo de Schelling en una red de 100x100 con un afinidad al grupo propio y otro mas para cada tipo de agente.

Este modelo se basa en en ajustar el parámetro de felicidad global y lo logra casi en su totalidad debido a que no existen distinciones de algún tipo entre etnias.

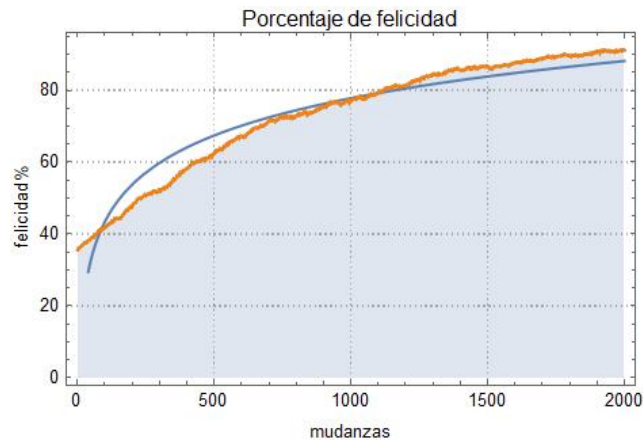


Figura 2.20: Gráfico de la evolución del porcentaje de felicidad para el sistema correspondiente a la figura 2.15 donde es favorable pertenecer a la mayoría.

2.3.4. Modelo (Economía) del mercado Inmobiliario

El modelo de Schelling resulto ser una herramienta sumamente útil para abordar temas sociológicos desde la perspectiva de los sistemas complejos no obstante como ya mencionamos sus alcances no están delimitados a este ramo de las ciencias sociales. Si se integra el factor monetario a este sistema podemos estudiar un modelo muy simplificado del mercado inmobiliario, también podemos analizar como el sistema se encarga de regular no solo el factor global de felicidad sino también el precio máximo que una propiedad puede alcanzar.

En términos generales este nuevo modelo opera siguiendo el mismo algoritmo que el modelo de Schelling sin embargo tendrá cambios importantes. En primer lugar no se estudiara la afinidad a un grupo étnico, en este sistema cada individuo va a poseer una cierta cantidad de dinero que se le asignara de forma aleatoria posteriormente según la cantidad de dinero que posee se le asigna un estrato social o económico (clase baja, media baja, media alta y alta), cada agente tendrá ahora una afinidad a alguna de estas clases, en particular se programo para que sientan afinidad por las clases mas altas, donde resulta evidente que la clase baja sera la menos favorecida y la que en general causara aversión a todas las demás clase e incluso la propia. Para realizar una mudanza ahora cada individuo no solo revisara su nivel de afinidad si no que evaluara cual es su utilidad antes y al finalizar el proceso, utilizando la relación $U \propto k^\alpha P^{\alpha-1}$ donde U es la utilidad, k es la cantidad de dinero que posee el agente, P es el valor de la propiedad y α es un parámetro empírico y solo en dado caso de que su utilidad mejore llevara acabo la transacción.

Arquitectura

- El sistema tendrá una geometría propia del toro, con la finalidad de poder emplear condiciones de frontera periódicas y facilitar el computo del sistema.
- Cada agente tendrá una cierta cantidad de dinero asignada de forma aleatoria siguiendo una distribución dada.
- Cada agente estará caracterizado por un cierto numero de símbolos numéricos o alfabéticos que representan su pertenencia a una clase social por dicho símbolo.
- La vecindad del agente esta compuesta por los ocho vecinos mas cercanos, aunque esta puede ser variable para ciertos casos en especifico.

Dinámica

- El sistema evolucionara de forma asíncrona.
- Se eligen dos agentes del sistema de forma aleatoria.
- Cada agente realiza el calculo de su nivel de felicidad y lo compara con su tolerancia para determinar si es "infeliz" en su estado actual.
- Cada agente valuara su propiedad y la propiedad a donde se podría mudar siguiendo la formula $P = A[M - D] + B$ donde M son los vecinos del mismo tipo y D son vecinos diferentes, A y B son constantes ajustables.
- Cada agente realiza el calculo de su utilidad antes y después de la hipotética transacción.
- Si ambos agentes son infelices, ambos mejoran su utilidad y tienen dinero suficiente para pagar la propiedad, se lleva a cabo la mudanza.
- El precio máximo que puede alcanzar una propiedad varia dependiendo si se concreto o no la mudanza.
- El sistema debe poseer un mecanismo de halting, es decir se debe detener cuando en un periodo considerablemente grande ya no pueda realizar mudanzas o ya haya realizado un cierto numero critico para el sistema .

resultados

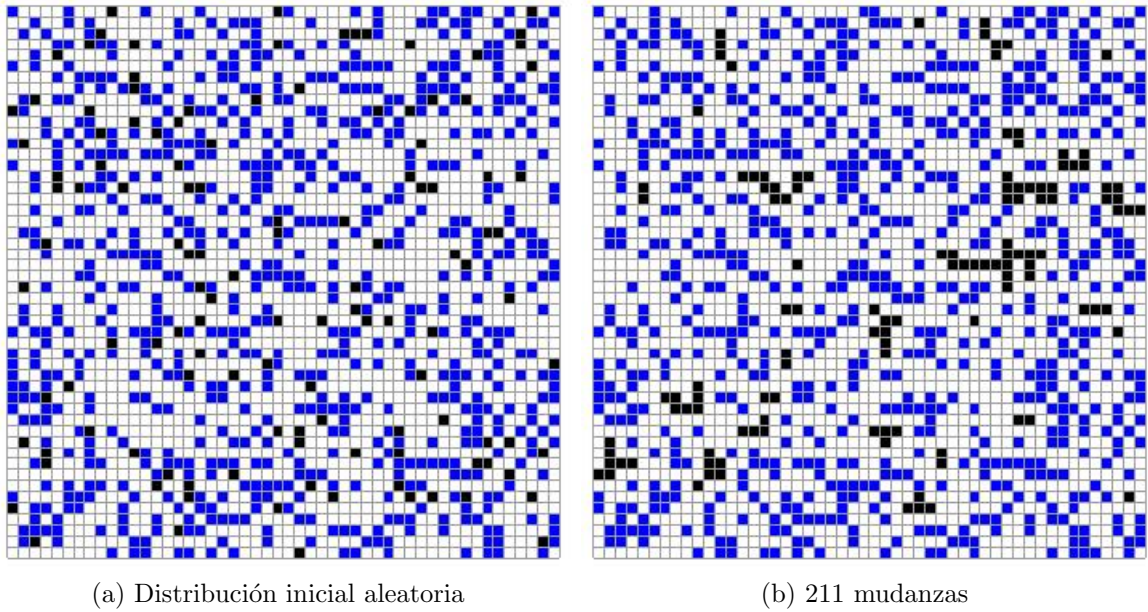


Figura 2.21: Modelo inmobiliario en una red de 50x50, cada agente recibió una cantidad de dinero similar a la situación de distribución de clases en México .

Este sistema funciona primeramente al dar a cada agente una cantidad de dinero siguiendo la estructura social en México, es decir un 50 % de pobres un 45 % de clase media, que posee entre un 75 % y un 200 % mas que la línea de pobreza y un 5 % de clase alta con al menos un 500 % por arriba de la línea de pobreza, gráficamente los agentes blancos son pobres, los agentes azules son clase media baja y los negros son clase alta, los agentes rojos serian la clase media alta pero el sistema no encontró agentes que entraran en esta clasificación. El valor de α se fijo en 2, mientras que los parámetros A y B varían para buscar regular el precio de los inmuebles, si se concreta una mudanza el precio máximo aumenta en una medida dv y si no se concreta disminuye en una medida dv . que es una cuarta parte de la diferencia del valor máximo de dinero asignado y el valor mínimo.

A diferencia del modelo simple de Schelling el sistema se frustró y no pudo concretar un número elevado de mudanzas, esto se debe principalmente a que la mayoría de los agentes son pobres y no pueden costear las propiedades, aun cuando el mercado trata de ajustar sus precios. En este caso la línea de pobreza de trazo en 1, el precio máximo de salida para una propiedad se fijo en 2.91, evidentemente un precio muy por encima de la línea de pobreza. Si realizamos el seguimiento de los precios de las propiedades observamos que el precio máximo promedio en el que se ofertaban las propiedades fue de 1.67, un precio aun mayor a la línea de pobreza. El precio máximo final fue de 0.4 lo que implica que las propiedades en este mercado perdieron aproximadamente un 90 % de su valor. En general podemos observar que solo las clases altas llevaron a cabo mudanzas.

resultados

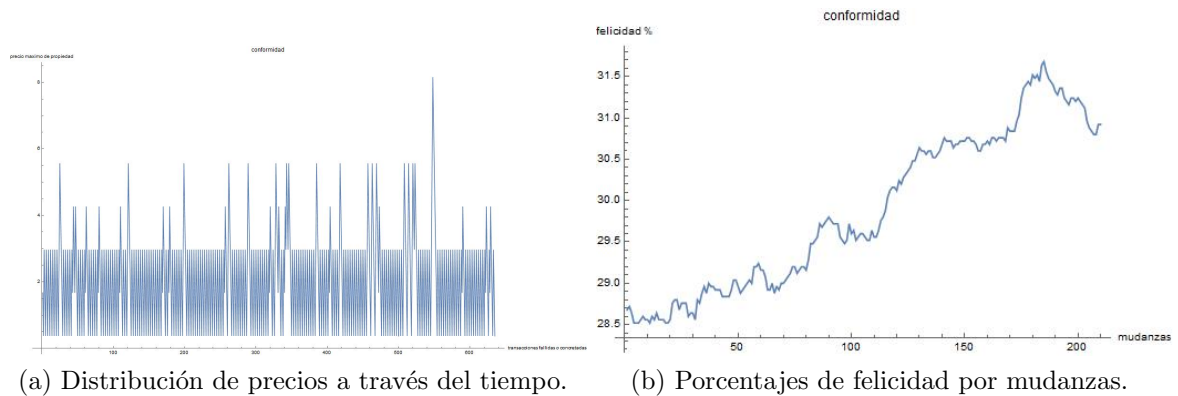


Figura 2.22: Modelo inmobiliario en una red de 50x50, gráficas de precios y felicidad para el sistema.

En las gráficas anteriores podemos observar como el sistema al igual que en el modelo de Schelling busca mejorar la propiedad global de conformidad llegando a un máximo aunque mucho mas alejado del 100 %. Por otra parte vemos como el sistema modifica los precios de las propiedades para intentar maximizar la cantidad de agentes felices dentro del sistema y al mismo tiempo mejorar la utilidad de los agentes.

Este sencillo modelo solo nos da información real sobre un hipotético mercado inmobiliario que funciona en estas condiciones sin embargo tenemos información al menos sobre el comportamiento de un mercado razonablemente similar o que al menos obedezca esa distribución de ingresos. Al menos podemos observar problemas evidentes que posee el mercado inmobiliario mexicano como es el estancamiento de las ventas la devaluación en ciertas zonas y en general los problemas propios de la desigualdad.

Este sistema igualmente es sensible a las modificaciones de sus parámetros conductores, desde que definimos el sistema inicial, en la forma en que distribuimos el dinero, si modificamos este parámetro o el factor α en que define a la utilidad tendremos comportamientos sumamente diferentes.

resultados

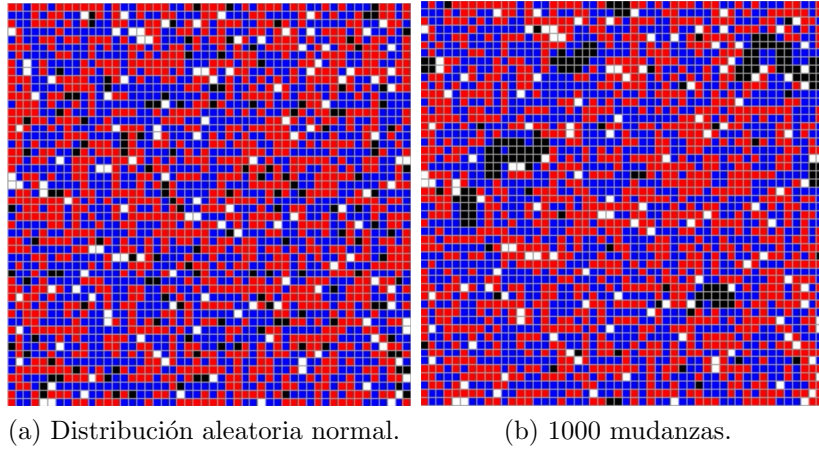


Figura 2.23: Modelo inmobiliario en una red de 50x50, con una distribución normal y precio fijo accesible a los pobres .

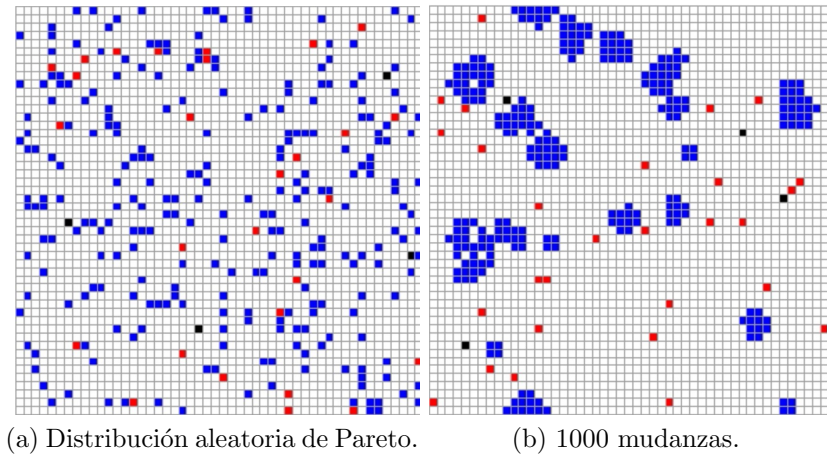


Figura 2.24: Modelo inmobiliario en una red de 50x50, con una distribución de Pareto y precio fijo accesible a los pobres.

Capítulo 3

Conclusiones y proyecciones

Como primera aproximación al tema de la econofísica y en particular desde el punto de vista de los sistemas complejos resultan cuanto menos satisfactorios los resultados de esta primera parte del proyecto. De forma teórica encontramos una gran cantidad de similitudes entre los enfoques más modernos para estudiar a la economía y los sistemas complejos, por lo que las técnicas utilizadas posteriormente a la introducción están justificadas y sustentadas en su totalidad.

De forma práctica encontramos que estudiar la economía desde la perspectiva de los sistemas complejos cumple con la máxima de esta disciplina, que es que la suma del todo es más que el todo. En las secciones finales se observó como al incluir el factor de las interacciones a agentes que funcionan de una forma extremadamente simple da como resultados dinámicas presentes en la economía cuyas explicaciones anteriores se encontraban incompletas desde una única perspectiva. Siguiendo este nuevo enfoque podemos dar explicación a estos fenómenos empleando solo un marco teórico y una metodología. Aunque se emplearon los sistemas más simples y como se prosigue en términos generales en la física, siguiendo un enfoque heurístico, se encontraron resultados que tienen una considerable proximidad con la realidad al menos de forma cualitativa, que sin embargo son suficientes para hacer un diagnóstico bastante acertado de un sistema social en particular.

Se puede suponer que si se toman los modelos construidos anteriormente en especial el modelo inmobiliario y se le agregan más factores, se pueden desarrollar herramientas mucho más precisas y útiles, en particular para el mercado inmobiliario propuesto, se pueden agregar los valores y parámetros propios de la macroeconomía por ejemplo para tratar de vislumbrar como los reguladores autónomos funcionan a nivel micro por ejemplo como influyen las tasas de interés y de impuestos en los bienes inmobiliarios para que un agente decida o no llevar a cabo una mudanza.

Para la segunda parte del proyecto resulta sumamente sugerente incluir por un lado mas factores propios de la economía a los modelos desarrollados y por otra parte incorporar términos propios de la física, en especial de la física estadística a la discusión, por ejemplo que interpretación podrían tener en este contexto términos como lo son la entropía del sistema o el significado de un ensamble desde un punto de vista de la econofísica.

Al inicio del presente trabajo se realizo una discusión acerca de la inteligencia artificial y el nacimiento de la computación, durante el resto del trabajo se mantuvo una estrecha relación entre los trabajos técnicos concernientes a la programación de los modelos y el desarrollo de teoría por lo que resulta imperativo continuar con este enfoque y a la par de mejorar y pulir los aspectos teóricos mejorar los algoritmos que vertebran a los sistemas para que el computo sea mas eficaz y por ende se puedan incluir mas factores.

Se pueden enlistar los objetivos y proyecciones para la segunda parte de la siguiente forma.

- Incluir términos y técnicas propias de la física estadística.
- Mejorar los algoritmos, para obtener una mayor capacidad de computo.
- Incluir la posibilidad de que los agentes dentro de los sistemas tengan la capacidad de aprender y recordar, introduciendo el trabajo del Machine- learnig.
- Retomar y emplear el estudio de la teoría de juegos.
- Incrementar la documentación económica, por ejemplo la información proveniente de las cuentas nacionales para analizar las predicciones cuantitativas que puedan realizar los sistemas.

Capítulo 4

Códigos 1

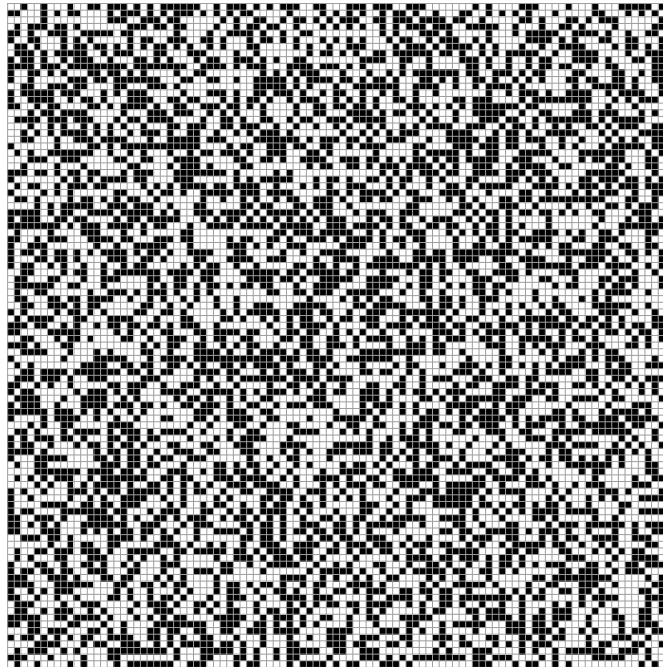
Cada uno de los siguientes códigos es el caso mas general empleado para cada uno de los modelos, para generar todos las figuras en cada sección se modificaron los parámetros como lo indica el documento.

```
(*Codigo para generar al automata celular  
con condiciones iniciales aleatorias de la figura 2.11)*)
```

```
long = 100;  
(*Definimos la longitud del arreglo o tablero de juego (tablero cuadrado)*)  
pasos = 100;  
(*Definimos la cantidad de pasos en la que evolucionara nuestro sistema*)
```

```
vida = Table[RandomInteger[], {p, long}, {q, long}];  
|tabla |entero aleatorio  
(*Damos condiciones aleatorias al arreglo*)  
MatrixForm[vida] ;  
|forma de matriz  
(* forma matricial del arreglo*)
```

```
In[390]:= ArrayPlot[vida, Mesh → True] (* representacion matricial del arreglo*)  
|representación de a·· |malla |verdadero
```



```
In[391]:= arreglo = vida;
```

```
In[392]:= vecinos = 0;
```

```
In[393]:= arreglo2 = arreglo;  
animacion = {arreglo};  
(*DEFINIMOS EL ARREGLO DE ARREGLOS QUE VA A CONTENER TODOS LOS PASOS *)
```



```

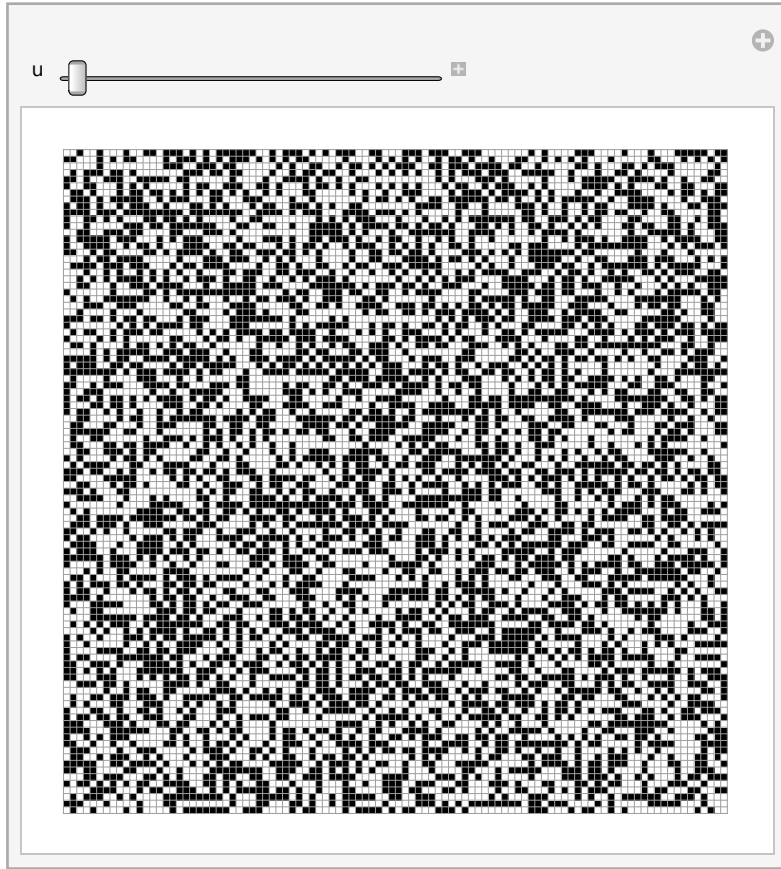
In[395]:= For[t = 1, t < pasos, t++,
  para cada
    For[i = 1, (i < long + 1), i++, For[j = 1, (j < long + 1), j++,
      para cada
        If[(i - 1) < 1, i1 = long, i1 = (i - 1)];
        si
          (*ESTAS LINEAS DEFINEN LAS CONDICIONES DE FRONTERA PERIODICAS*)
          If[(i + 1) > long, i2 = 1, i2 = (i + 1)];
          si
            If[(j - 1) < 1, j1 = long, j1 = (j - 1)];
            si
              If[(j + 1) > long, j2 = 1, j2 = (j + 1)];
              si
                esq1 = arreglo[[i1, j1]];
                (*ZONDEAMOS EL ESTADO DE LOS PRIMEROS VECINOS PARA EL ELEMENTO I,J DEL ARREGLO*)
                numero i
                esq2 = arreglo[[i1, j2]];
                esq3 = arreglo[[i2, j1]];
                esq4 = arreglo[[i2, j2]];
                arriba = arreglo[[i1, j]];
                abajo = arreglo[[i2, j]];
                izq = arreglo[[i, j1]];
                der = arreglo[[i, j2]]; vecinos = esq1 + esq2 + esq3 + esq4 + der + izq + arriba + abajo;
                If[vecinos == 3, arreglo2[[i, j]] = 1];
                si
                  If[vecinos > 3, arreglo2[[i, j]] = 0]; If[vecinos == 3, arreglo2[[i, j]] = 1];
                  si
                    If[vecinos < 2, arreglo2[[i, j]] = 0];
                    si
                      If[vecinos > 3, arreglo2[[i, j]] = 0]; ]];
                si
                  (*DEFINIMOS LAS REGLAS DEL JUEGO DE LA VIDA PARA SOBREVIVENCIA Y NACIMIENTO*)
                  arreglo = arreglo2; AppendTo[animacion, arreglo2];]
                añade al final

```

In[396]:=

```
In[397]:= Manipulate[ArrayPlot[animacion[[u]], Mesh -> True], {u, 1, pasos, 1}]  
[manipula [representación de arreglo [malla [verdadero
```

Out[397]=



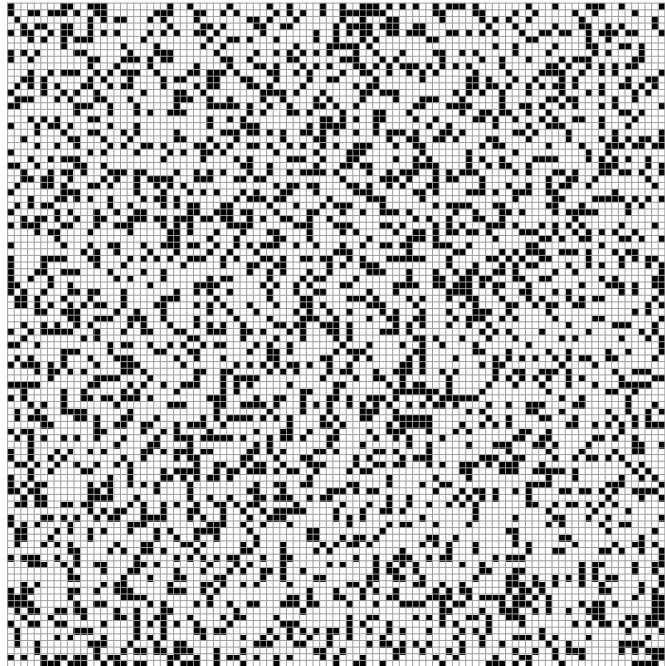
In[398]=

In[406]:= (*Codigo para la generacion del modelo de condensacion de la figura 2.12 y 2.13*)

```
long = 100;  
(*Definimos la longitud del arreglo o tablero de juego (tablero cuadrado)*)  
pasos = 1000;  
(*Definimos la cantidad de pasos en la que evolucionara nuestro sistema*)  
  
vida = Table[RandomChoice[{0.30, 0.70} -> {1, 0}], {p, long}, {q, long}];  
  tabla  |elección aleatoria  
  
(*Damos las condiciones con los pesos indicados al arreglo*)
```

In[407]:= ArrayPlot[vida, Mesh -> True] (* representacion matricial del arreglo*)
 representación de a· |malla |verdadero

Out[407]=



In[408]:= arreglo = vida;

In[409]:= vecinos = 0;

In[410]:= arreglo2 = arreglo;
animacion = {arreglo};
(*DEFINIMOS EL ARREGLO DE ARREGLOS QUE VA A CONTENER TODOS LOS PASOS *)

```

In[412]:= For[t = 1, t < pasos, t++,
  para cada
    For[i = 1, (i < long + 1), i++, For[j = 1, (j < long + 1), j++,
      para cada
        If[(i - 1) < 1, i1 = long, i1 = (i - 1)];
        si
          (*ESTAS LINEAS DEFINEN LAS CONDICIONES DE FRONTERA PERIODICAS*)
          If[(i + 1) > long, i2 = 1, i2 = (i + 1)];
          si
            If[(j - 1) < 1, j1 = long, j1 = (j - 1)];
            si
              If[(j + 1) > long, j2 = 1, j2 = (j + 1)];
              si
                esq1 = arreglo[[i1, j1]];
                (*ZONDEAMOS EL ESTADO DE LOS PRIMEROS VECINOS PARA EL ELEMENTO I,J DEL ARREGLO*)
                numero i
                esq2 = arreglo[[i1, j2]];
                esq3 = arreglo[[i2, j1]];
                esq4 = arreglo[[i2, j2]];
                arriba = arreglo[[i1, j]];
                abajo = arreglo[[i2, j]];
                izq = arreglo[[i, j1]];
                der = arreglo[[i, j2]]; vecinos = esq1 + esq2 + esq3 + esq4 + der + izq + arriba + abajo;
                If[vecinos >= 4, arreglo2[[i, j]] = 1, arreglo2[[i, j]] = 0]; ];
                si
          (*DEFINIMOS LAS REGLAS DE CONDENSACION*)
          arreglo = arreglo2; AppendTo[animacion, arreglo2];] (*Agregamos los nuevos
          añade al final
          valores del arreglo (arreglo 2) al arreglo de arreglos (animacion)*)

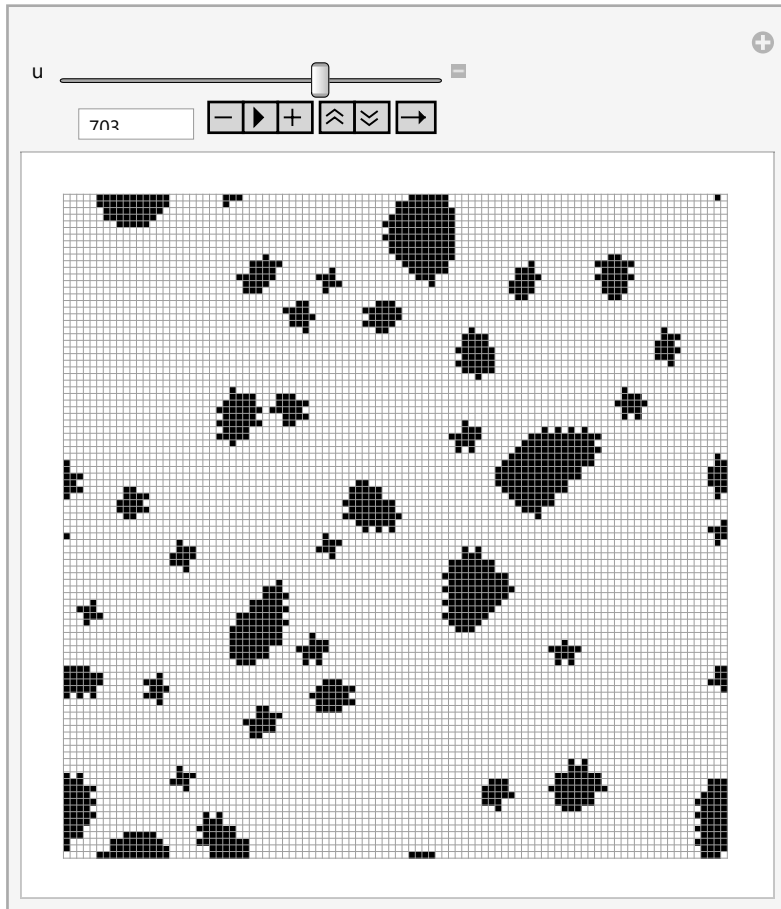
```

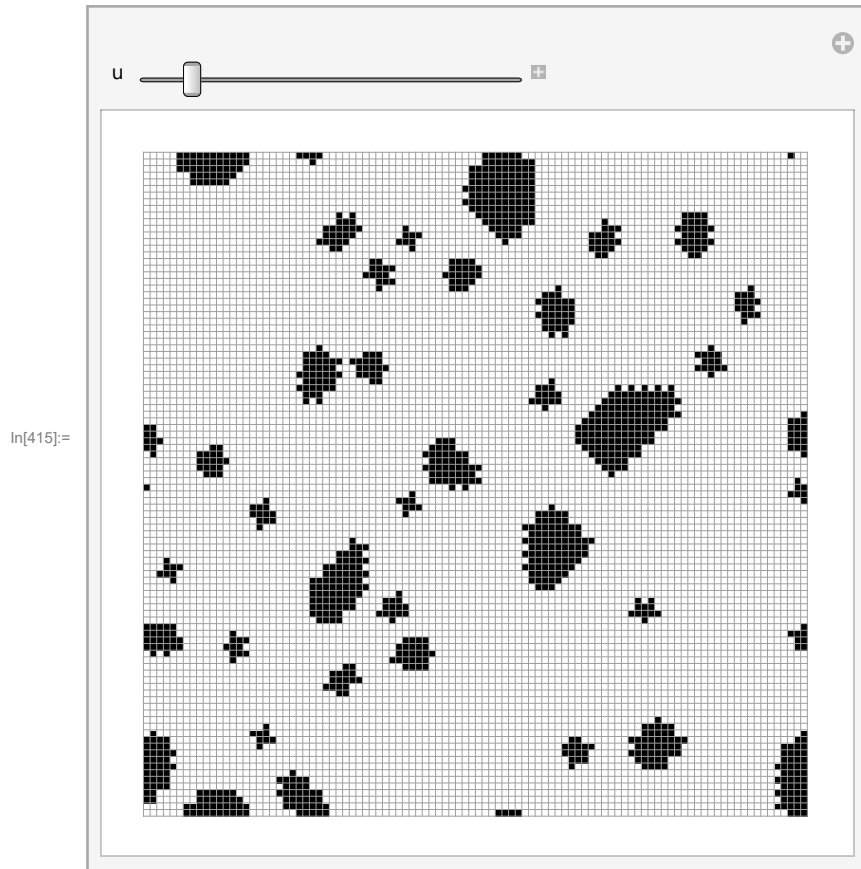
In[413]:=

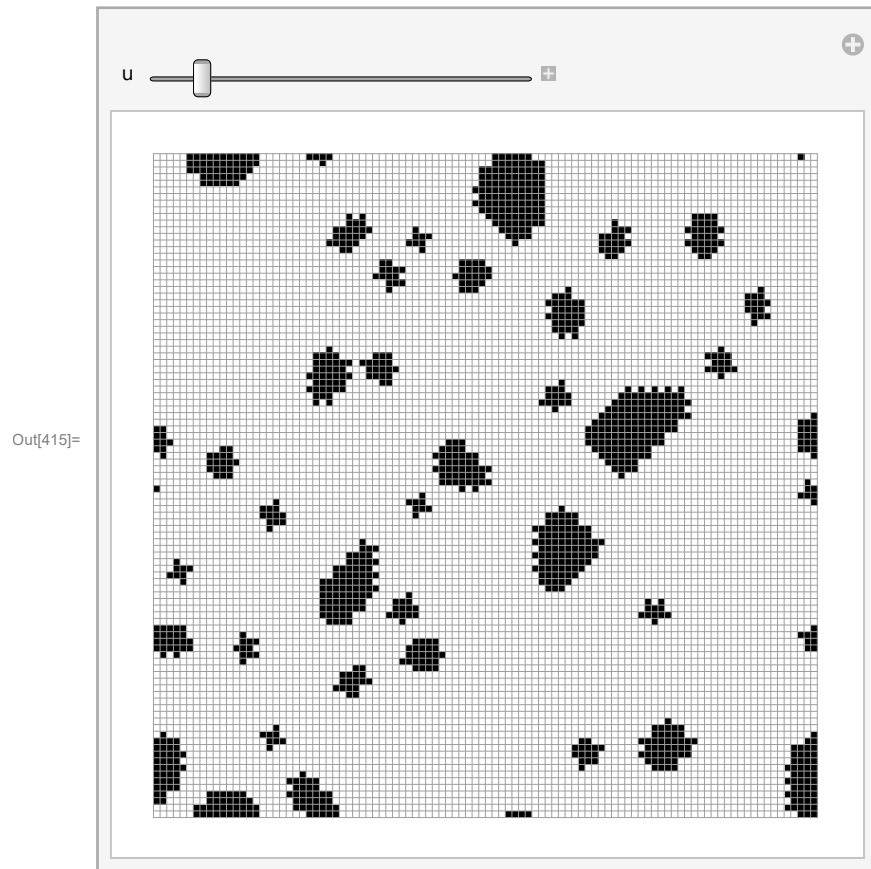
```
In[414]:= video = Manipulate[ArrayPlot[animacion[[u]], Mesh -> True], {u, 1, pasos, 1}]
```

[manipula](#) [representación de arreglo](#) [malla](#) [verdadero](#)

Out[414]=







In[416]=

In[417]= `Export["condensado25.gif", video]`
`exporta`

Out[417]= `condensado25.gif`

```
(*Codigo para generar el modelo de Schellign en
su forma mas general utilizando diferentes tolerancias*)
```

```
long = 100;
(*Definimos la longitud del arreglo o tablero de juego (tablero cuadrado)*)

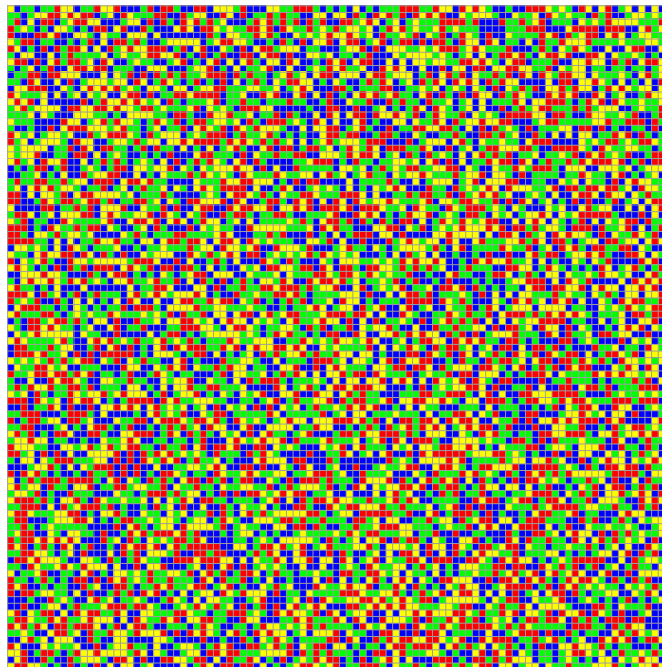
tolerancia1 = 4 / 8;
tolerancia2 = 5 / 8; (*Definimos las tolerancias para cada tipo de agente*)
tolerancia3 = 5 / 8;
tolerancia4 = 3 / 8;
supertotal = 100000000; (*Definimos el numero de iteraciones que realizara el sistema,
si no hay una interrupcion*)
total = 100000; (*Definimos el numero de errores
consecutivos o mudanzas frustradas para detener la iteracion*)
mudanzas = 100000;
(*Definimos el numero de mudanzas a realizar para detener las iteraciones*)
vida = Table[RandomChoice[{0.25, 0.25, 0.25, 0.25} -> {0, 1, 3, 4}], {p, long}, {q, long}];
|tabla |elección aleatoria

(*Damos las condiciones con los pesos indicados al arreglo*)

ArrayPlot[vida, Mesh -> True, ColorRules -> {1 -> Yellow, 0 -> Green, 3 -> Red, 4 -> Blue}]
|representación de a·· |malla |verd··· |reglas de color |amarillo |verde |rojo |azul

(* representacion grafica del arreglo,
donde cada color representa un tipo de agente distinto*)
```

Out[]:=



```
arreglo = vida;
(* en las lineas siguientes definimos los parametros
de inicio para que el sistema comience a funcionar*)
```



```

In[259]:= suma1 = 0;

In[*]:= check = 0;

In[*]:= suma2 = 0;

In[262]:= mudanza = 0;
animacion = {arreglo};
fallo = 0;
conformidad = {};

In[*]:= vecinos = 0;

In[*]:= estado = 0;

In[*]:= contento1 = true;

In[*]:= contento2 = true;

While[check < supertotal,
  Mientras
    suma1 = 0; suma2 = 0;
    Label[aleatorio];
    Etiqueta
    ri1 = RandomInteger[{1, (long)}];
      entero aleatorio
    rj1 = RandomInteger[{1, (long)}]; ri2 = RandomInteger[{1, (long)}];
      entero aleatorio          entero aleatorio
    rj2 = RandomInteger[{1, (long)}]; (*llamamos a los numeros aleatorios
      entero aleatorio
      que corresponden a las coordenadas de dos de los agentes del sistema*)
    pos1 = arreglo[[ri1, rj1]];
    pos2 = arreglo[[ri2, rj2]];
    If[pos1 == pos2, Goto[aleatorio], check = check + 1];
      si          ve a
      (*Definimos la condicion de que los agentes deben de ser diferentes, de lo contrario
      regresamos a la seleccion de los mismos*) If[(ri1 - 1) < 1, ri11 = long, ri11 = (ri1 - 1)];
        si
    If[(ri2 - 1) < 1, ri21 = long, ri21 = (ri2 - 1)];
      si
    If[(ri1 + 1) > long, ri12 = 1, ri12 = (ri1 + 1)];
      si
    (*Definimos las condiciones de frontera periodicas*)
    If[(ri2 + 1) > long, ri22 = 1, ri22 = (ri2 + 1)];
      si
    If[(rj1 - 1) < 1, rj11 = long, rj11 = (rj1 - 1)];
      si
    If[(rj2 - 1) < 1, rj21 = long, rj21 = (rj2 - 1)];
      si
    If[(rj1 + 1) > long, rj12 = 1, rj12 = (rj1 + 1)];
      si

```

```

If[(rj2 + 1) > long, rj22 = 1, rj22 = (rj2 + 1)];
If[arreglo[[ri11, rj11]] == pos1, suma1 = suma1 + 1];
If[arreglo[[ri11, rj12]] == pos1, suma1 = suma1 + 1];
If[arreglo[[ri12, rj11]] == pos1, suma1 = suma1 + 1]; If[arreglo[[ri12, rj12]] == pos1,
suma1 = suma1 + 1]; If[arreglo[[ri11, rj1]] == pos1, suma1 = suma1 + 1];
If[arreglo[[ri12, rj1]] == pos1, suma1 = suma1 + 1]; If[arreglo[[ri1, rj11]] == pos1,
suma1 = suma1 + 1]; If[arreglo[[ri1, rj12]] == pos1, suma1 = suma1 + 1];

(*Realizamos el cumulo de felicidad o afinidad de las
vecindades de cada uno de los dos agentes seleccionados,
preguntando a cada vecino si es igual al valor del agente definido como pos1 o pos2*)

If[arreglo[[ri21, rj21]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri21, rj22]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri22, rj21]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri22, rj22]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri21, rj2]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri22, rj2]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri2, rj21]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri2, rj22]] == pos2, suma2 = suma2 + 1];

vecindad1 = suma1 / 8;
vecindad2 = suma2 / 8;
(*Calculamos la afinidad de cada agente que posteriormente se comparara
con su nivel de tolerancia para decidir si se llevara a cabo la mudanza*)
Switch[pos1, 0, If[vecindad1 <= tolerancia1, contenido1 = false, contenido1 = true],
1, If[vecindad1 <= tolerancia2, contenido1 = false, contenido1 = true],
3, If[vecindad1 <= tolerancia3, contenido1 = false, contenido1 = true],
4, If[vecindad1 <= tolerancia4, contenido1 = false, contenido1 = true]];
Switch[pos2, 0, If[vecindad2 <= tolerancia1, contenido2 = false, contenido2 = true],

```

```

Levalua deperidientu... [si
1, If[vecindad2 <= tolerancia2, contenido2 = false, contenido2 = true],
  [si
3, If[vecindad2 <= tolerancia3, contenido2 = false, contenido2 = true],
  [si
4, If[vecindad2 <= tolerancia4, contenido2 = false, contenido2 = true]];
(*Preguntamos para cada uno de los dos agentes a que valor corresponde o a que
grupo pertenece para poder compararlo con su tolerancia correspondiente*)

If[contenido1 == false && contenido2 == false, fallo = 0;
[si
  mudanza = mudanza + 1;
  arreglo[[ri1, rj1]] = pos2; arreglo[[ri2, rj2]] = pos1;
  AppendTo[animacion, arreglo], fallo = fallo + 1];
[añade al final
(*Si se lleva a cabo la mudanza se intercambian los lugares pos1 y
pos2 y el nuevo arreglo se incluye en el arreglo animacion *)
contenido1 = true;
contenido2 = true;
If[mudanza > mudanzas, check = supertotal];
[si
If[fallo > total, check = supertotal];]
[si
(*Se definen las condiciones de halting del sistema*)

```

In[*]:= **fallo**

Out[*]:= 0

In[*]:= **check**

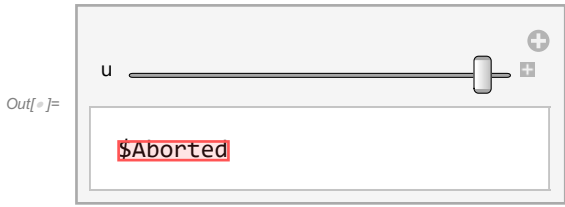
Out[*]:= 100 000 000

In[*]:= **mudanza**

Out[*]:= 100 001

In[*]:=

```
In[ ]:= video = Manipulate[ArrayPlot[animacion[ u ], Mesh -> True,
    {manipula {representación de arreglo {malla {verdadero
    ColorRules -> {1 -> Yellow, 0 -> Green, 3 -> Red, 4 -> Blue}}, {u, 1, mudanza, 1}}
    {reglas de color {amarillo {verde {rojo {azul
```



Part: Part 100001 of
 {{{0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, <<50>>},
 {1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,
 <<50>>}, {0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0,
 1, 0, <<50>>}, <<45>>, {1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 1, 0, 1, 1, 1, 0, 1, <<50>>}, {1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0,
 1, 0, 1, 1, 0, 1, <<50>>}, <<50>>}, <<49>>, <<50>>} does not exist.

ArrayPlot: Argument
 {{{0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, <<50>>},
 {1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,
 <<50>>}, {0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1,
 0, 1, 0, <<50>>}, <<45>>, {1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
 0, 0, 1, 0, 1, 1, 1, 0, 1, <<50>>}, {1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1,
 1, 0, 0, 1, 0, 1, 1, 0, 1, <<50>>}, <<50>>}, <<49>>, <<50>>}<<1>><<6>>]] at position 1 is not a list of lists.

Part: Part 100001 of
 {{{1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, <<50>>},
 {1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
 <<50>>}, {0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0,
 0, 0, <<50>>}, <<45>>, {0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
 0, 1, 1, 0, 0, 1, 1, <<50>>}, {1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0,
 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, <<50>>}, <<50>>}, <<49>>, <<950>>} does not exist.

ArrayPlot: Argument
 {{{1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, <<50>>},
 {1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
 <<50>>}, {0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0,
 0, 0, <<50>>}, <<45>>, {0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
 1, 1, 0, 1, 1, 0, 0, 1, 1, <<50>>}, {1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1,
 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, <<50>>}, <<50>>}, <<49>>, <<950>>}}[100001]] at position 1 is not a list of lists.

```
In[ ]:= Export["segregaciond12.gif", video]
    {exporta
```

Out[]:= segregaciond12.gif

```

(*Codigo para generar el modelo de Schelling con mercado inmobiliario integrado,
con precio maximo de compra modular en el sistema y una distribucion
normal alrededor de los valores propios de un mercado mexicano promedio *)
long = 50;      (*Definimos la longitud del sistema *)

(*A cada agente se le asigna de forma aleatoria una cantidad de dinero*)

dinero =
  Table[RandomChoice[{0.5, 0.2, 0.25, 0.05} -> {RandomVariate[NormalDistribution[1, 0.2]],
    [tabla [elección aleatoria [variable aleatoria [distribución normal
      RandomVariate[NormalDistribution[1.5, 0.2]], RandomVariate[NormalDistribution[2,
    [variable aleatoria [distribución normal [variable aleatoria [distribución normal
      0.2]], RandomVariate[NormalDistribution[5, 0.2]]}], {p, long}, {q, long}]];
    [variable aleatoria [distribución normal

(*Al igual que en el caso estandar de Schelling se definen niveles de
tolerancia para cada una de las clases sociales y se
definen los valores para el halting del algoritmo*)
tolerancia1 = 5 / 8;
tolerancia2 = 2 / 8;
tolerancia3 = 3 / 8;
tolerancia4 = 2 / 8;
supertotal = 500000;
total = 2500;
mudanzas = 2000;
conformidad = {};

clase = Table[4, {p, long}, {q, long}];
  [tabla

(*Iniciamos el arreglo que contendra la representacion
grafica de la clase social a la que pertenece el agente*)

In[316]:= maximo = Max[dinero]
  [máximo
Out[316]= 5.62016

In[317]:= minimo = Min[dinero]
  [mínimo
Out[317]= 0.450403

dv = (maximo - minimo) / 4 (*A partir de los valores maximos del
arreglo de dinero definimos la diferencia de dinero entre clase social*)
Out[318]= 1.29244

dv1 = (dv * 2) + minimo (*Fijamos un valor maximo de salida para las propiedades*)
Out[319]= 3.03528

(*Con este For le asignamos la representacion grafica
  [para cada
de cada clase social calculando cuantos dv posee cada agente*)

```

```

In[320]= For[i = 1, (i < long + 1), i++, For[j = 1, (j < long + 1), j++,
  para cada
    If[dinero[[i, j]] ≤ (minimo + dv), clase[[i, j]] = 0];
  si
    If[dinero[[i, j]] > (minimo + dv) &&
      si
        dinero[[i, j]] ≤ (minimo + (2 * dv)), clase[[i, j]] = 1];
    If[dinero[[i, j]] > (minimo + (2 * dv)) && dinero[[i, j]] ≤ (minimo + (3 * dv)),
      si
        clase[[i, j]] = 2];
    If[dinero[[i, j]] > (minimo + (3 * dv)), clase[[i, j]] = 3];
  si
  ]];

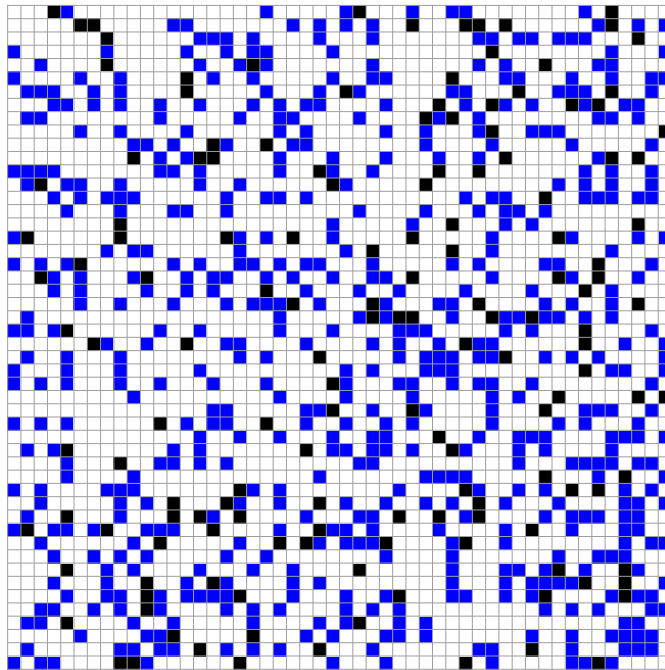
```

```

ArrayPlot[clase, Mesh → True, ColorRules → {0 → White, 1 → Blue, 2 → Red, 3 → Black}]
representación de ar... malla verd... reglas de color blanco azul rojo negro
(*Representacion grafica del arreglo de clases*)

```

Out[321]=



```

In[322]:= cuenta1 = 0;
          cuenta2 = 0;
          cuenta3 = 0;
          cuenta4 = 0;
          For[i = 1, i < (long + 1), i++,
            para cada
              cuenta1 = cuenta1 + Count[clase[[i]], 0];
              cuenta2 = cuenta2 + Count[clase[[i]], 1];
              cuenta3 = cuenta3 + Count[clase[[i]], 2];
              cuenta4 = cuenta4 + Count[clase[[i]], 3];]

          (*Numero de agentes por clase social *)

```

```
In[327]:= cuenta1
```

```
Out[327]= 1744
```

```
In[328]:= cuenta2
```

```
Out[328]= 606
```

```
In[329]:= cuenta3
```

```
Out[329]= 0
```

```
In[330]:= cuenta4
```

```
Out[330]= 150
```

```
In[331]:= total = cuenta1 + cuenta2 + cuenta3 + cuenta4
```

```
Out[331]= 2500
```

```

          (*Valores iniciales para comenzar el sistema *)
          dinero1 = dinero;
          suma1 = 0;

```

```
In[334]:= clase1 = clase;
```

```
In[335]:= check = 0;
          checkin = 0;
```

```
In[337]:= suma2 = 0;
```

```
In[338]:= mudanza = 0;
          animacion = {clase};
          fallo = 0;
```

```
In[341]:= vecinos = 0;
```

```
In[342]:= estado = 0;
```

```
In[343]:= contenido1 = true;
```

```
In[344]:= contenido2 = true;
```

```
alpha = 1.5;
```

```
B = dv1 / 2
```

```
A = B / 8
```

```
c11 = 0;
```

```
p11 = 0;
```

```
c12 = 0;
```

```
p12 = 0;
```

```
c21 = 0;
```

```
p21 = 0;
```

```
c22 = 0;
```

```
p22 = 0;
```

```
valor = {};
```

```
ivalor = 0;
```

```
Out[346]= 1.51764
```

```
Out[347]= 0.189705
```

```
While[check < supertotal && fallo < total && mudanza < mudanzas,  
  mientras
```

```
    Label[aleatorio];
```

```
    etiqueta
```

```
    suma11 = 0;
```

```
    suma21 = 0; suma12 = 0;
```

```
    suma22 = 0; A = B / 8; (*Asignacion reiterativa de valores*)
```

```
    ri1 = RandomInteger[{1, (long)}];
```

```
    entero aleatorio
```

```
    rj1 = RandomInteger[{1, (long)}];
```

```
    entero aleatorio
```

```
    ri2 = RandomInteger[{1, (long)}];
```

```
    entero aleatorio
```

```
    rj2 = RandomInteger[{1, (long)}];
```

```
    entero aleatorio
```

```
    (*Coordenadas aleatorias para seleccionar a los dos agentes*)
```

```
    pos1 = clase1[[ri1, rj1]];

```

```
    pos2 = clase1[[ri2, rj2]];

```

```
    (*Se verifica que los dos agentes sean de clases sociales diferentes*)
```

```
    If[pos1 == pos2, Goto[aleatorio], check = check + 1];
```

```
    si ve a
```

```
    k1i = dinero1[[ri1, rj1]];

```

```
    k2i = dinero1[[ri2, rj2]];

```

```
    (*Definimos con cuanto dinero dispone cada agente
```

```
    para realizar sus transacciones a partir del arreglo de dinero*)
```



```

(*Definimos las condiciones de frontera periodica*)
If[(ri1 - 1) < 1, ri11 = long, ri11 = (ri1 - 1)];
|_si
If[(ri2 - 1) < 1, ri21 = long, ri21 = (ri2 - 1)];
|_si
If[(ri1 + 1) > long, ri12 = 1, ri12 = (ri1 + 1)];
|_si
If[(ri2 + 1) > long, ri22 = 1, ri22 = (ri2 + 1)];
|_si
If[(rj1 - 1) < 1, rj11 = long, rj11 = (rj1 - 1)];
|_si
If[(rj2 - 1) < 1, rj21 = long, rj21 = (rj2 - 1)];
|_si
If[(rj1 + 1) > long, rj12 = 1, rj12 = (rj1 + 1)];
|_si
If[(rj2 + 1) > long, rj22 = 1, rj22 = (rj2 + 1)];
|_si

(*El primer agente calcula su vecindad actual*)
If[clase1[[ri11, rj11]] ≥ pos1 && clase1[[ri11, rj11]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri11, rj12]] ≥ pos1 && clase1[[ri11, rj12]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri12, rj11]] ≥ pos1 && clase1[[ri12, rj11]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri12, rj12]] ≥ pos1 && clase1[[ri12, rj12]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri11, rj1]] ≥ pos1 && clase1[[ri11, rj1]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri12, rj1]] ≥ pos1 && clase1[[ri12, rj1]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri1, rj11]] ≥ pos1 && clase1[[ri1, rj11]] ≠ 0, suma11 = suma11 + 1];
|_si
If[clase1[[ri1, rj12]] ≥ pos1 && clase1[[ri1, rj12]] ≠ 0, suma11 = suma11 + 1];
|_si

(*El primer agente calcula su posible nueva vecindad *)
If[clase1[[ri21, rj21]] ≥ pos1 && clase1[[ri21, rj21]] ≠ 0, suma12 = suma12 + 1];
|_si
If[clase1[[ri21, rj22]] ≥ pos1 && clase1[[ri21, rj22]] ≠ 0, suma12 = suma12 + 1];
|_si
If[clase1[[ri22, rj21]] ≥ pos1 && clase1[[ri22, rj21]] ≠ 0, suma12 = suma12 + 1];
|_si
If[clase1[[ri22, rj22]] ≥ pos1 && clase1[[ri22, rj22]] ≠ 0, suma12 = suma12 + 1];
|_si
If[clase1[[ri21, rj2]] ≥ pos1 && clase1[[ri21, rj2]] ≠ 0, suma12 = suma12 + 1];
|_si
If[clase1[[ri22, rj2]] ≥ pos1 && clase1[[ri22, rj2]] ≠ 0, suma12 = suma12 + 1];
|_si
If[clase1[[ri2, rj21]] ≥ pos1 && clase1[[ri2, rj21]] ≠ 0, suma12 = suma12 + 1];
|_ei

```

```

[>]
If[clase1[[ri2, rj22]] ≥ pos1 && clase1[[ri2, rj22]] ≠ 0, suma12 = suma12 + 1];
[si]
(*El segundo agente calcula su vecindad actual*)
If[clase1[[ri11, rj11]] ≥ pos2 && clase1[[ri11, rj11]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri11, rj12]] ≥ pos2 && clase1[[ri11, rj12]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri12, rj11]] ≥ pos2 && clase1[[ri12, rj11]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri12, rj12]] ≥ pos2 && clase1[[ri12, rj12]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri11, rj1]] ≥ pos2 && clase1[[ri11, rj1]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri12, rj1]] ≥ pos2 && clase1[[ri12, rj1]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri1, rj11]] ≥ pos2 && clase1[[ri1, rj11]] ≠ 0, suma22 = suma22 + 1];
[si]
If[clase1[[ri1, rj12]] ≥ pos2 && clase1[[ri1, rj12]] ≠ 0, suma22 = suma22 + 1];
[si]
(*El primer agente calcula su posible nueva vecindad *)
If[clase1[[ri21, rj21]] ≥ pos2 && clase1[[ri21, rj21]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri21, rj22]] ≥ pos2 && clase1[[ri21, rj22]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri22, rj21]] ≥ pos2 && clase1[[ri22, rj21]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri22, rj22]] ≥ pos2 && clase1[[ri22, rj22]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri21, rj2]] ≥ pos2 && clase1[[ri21, rj2]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri22, rj2]] ≥ pos2 && clase1[[ri22, rj2]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri2, rj21]] ≥ pos2 && clase1[[ri2, rj21]] ≠ 0, suma21 = suma21 + 1];
[si]
If[clase1[[ri2, rj22]] ≥ pos2 && clase1[[ri2, rj22]] ≠ 0, suma21 = suma21 + 1];
[si]
(*El primer agente tasa su propiedad*)
c11 = suma11;
p11 = (A * ((2 * c11) - 8)) + B;
(*El primer agente tasa la propiedad del segundo agente*)
c12 = suma12;
p12 = (A * ((2 * c12) - 8)) + B;

(*El segundo agente tasa su propiedad*)
c21 = suma21;
p21 = (A * ((2 * c21) - 8)) + B;

(*El segundo agente tasa la propiedad del primer agente*)
c22 = suma22;
p22 = (A * ((2 * c22) - 8)) + B;

```

```

(*ambos agentes calculan su utilidad inicial*)
u1i = ((k1i)^(alpha)) * ((p11)^(alpha - 1));
u2i = ((k2i)^(alpha)) * ((p21)^(alpha - 1));
(*El segundo agente tasa la propiedad del primer agente*)
p1 = (p12 + p21) / 2;
p2 = (p22 + p11) / 2;
(*Se calcula el dinero con el que contarían al final de la hipotética transacción*)
k1f = k1i - p1 + p2;
k2f = k2i - p2 + p1;

(*Si ambos agentes quedan con una cantidad de dinero positiva el sistema
prosigue si no selecciona otros dos agentes y modifica el parámetro B*)
If[k1f >= 0 && k2f >= 0, u1f = ((k1f)^(alpha)) * ((p12)^(alpha - 1));
si
u2f = ((k2f)^(alpha)) * ((p22)^(alpha - 1));, If[B > (dv/2), B = (B - (dv/2));
si

valores = B * 2;
AppendTo[valor, valores]]; Goto[aleatorio];];
añade al final ve a

(*Si ambos agentes quedan con una utilidad positiva el sistema
prosigue si no selecciona otros dos agentes y modifica el parámetro B*)
If[u1f <= u1i || u2f <= u2i, If[B > (dv/2), B = (B - (dv/2));
si si

valores = B * 2;
AppendTo[valor, valores]];];
añade al final

Goto[aleatorio];];
ve a

vecindad1 = suma11 / 8;
vecindad2 = suma21 / 8;
Switch[pos1, 0, If[vecindad1 <= tolerancia1, contenido1 = false, contenido1 = true],
evalúa dependiendo... si
1, If[vecindad1 <= tolerancia2, contenido1 = false, contenido1 = true],
si
2, If[vecindad1 <= tolerancia3, contenido1 = false, contenido1 = true],
si
3, If[vecindad1 <= tolerancia4, contenido1 = false, contenido1 = true]];];
si

Switch[pos2, 0, If[vecindad2 <= tolerancia1, contenido2 = false, contenido2 = true],
evalúa dependiendo... si
1, If[vecindad2 <= tolerancia2, contenido2 = false, contenido2 = true],
si
2, If[vecindad2 <= tolerancia3, contenido2 = false, contenido2 = true],
si

```

```

3, If[vecindad2 <= tolerancia4, contenido2 = false, contenido2 = true]];

```

```

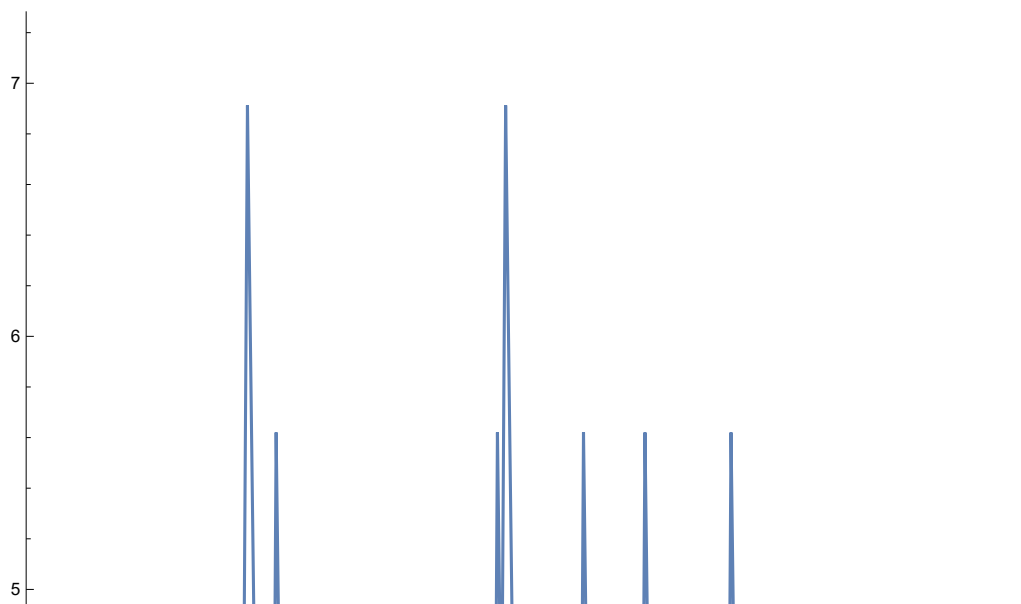
If[contenido1 == false && contenido2 == false, fallo = 0;
]
  mudanza = mudanza + 1 ;
  B = (B + dv) ;
  valores = B * 2;
  AppendTo[valor, valores];
  clase1[[ri1, rj1]] = pos2; clase1[[ri2, rj2]] = pos1;
  dinero1[[ri1, rj1]] = k2f; dinero1[[ri2, rj2]] = k1f;
  maximo = Max[dinero];
  minimo = Min[dinero];
  dv = (maximo - minimo) / 4;
  AppendTo[animacion, clase1], If[B > (dv / 2), B = (B - (dv / 2))];
  valores = B * 2;
  AppendTo[valor, valores];
  fallo = fallo + 1];
contenido1 = true;
contenido2 = true;
];

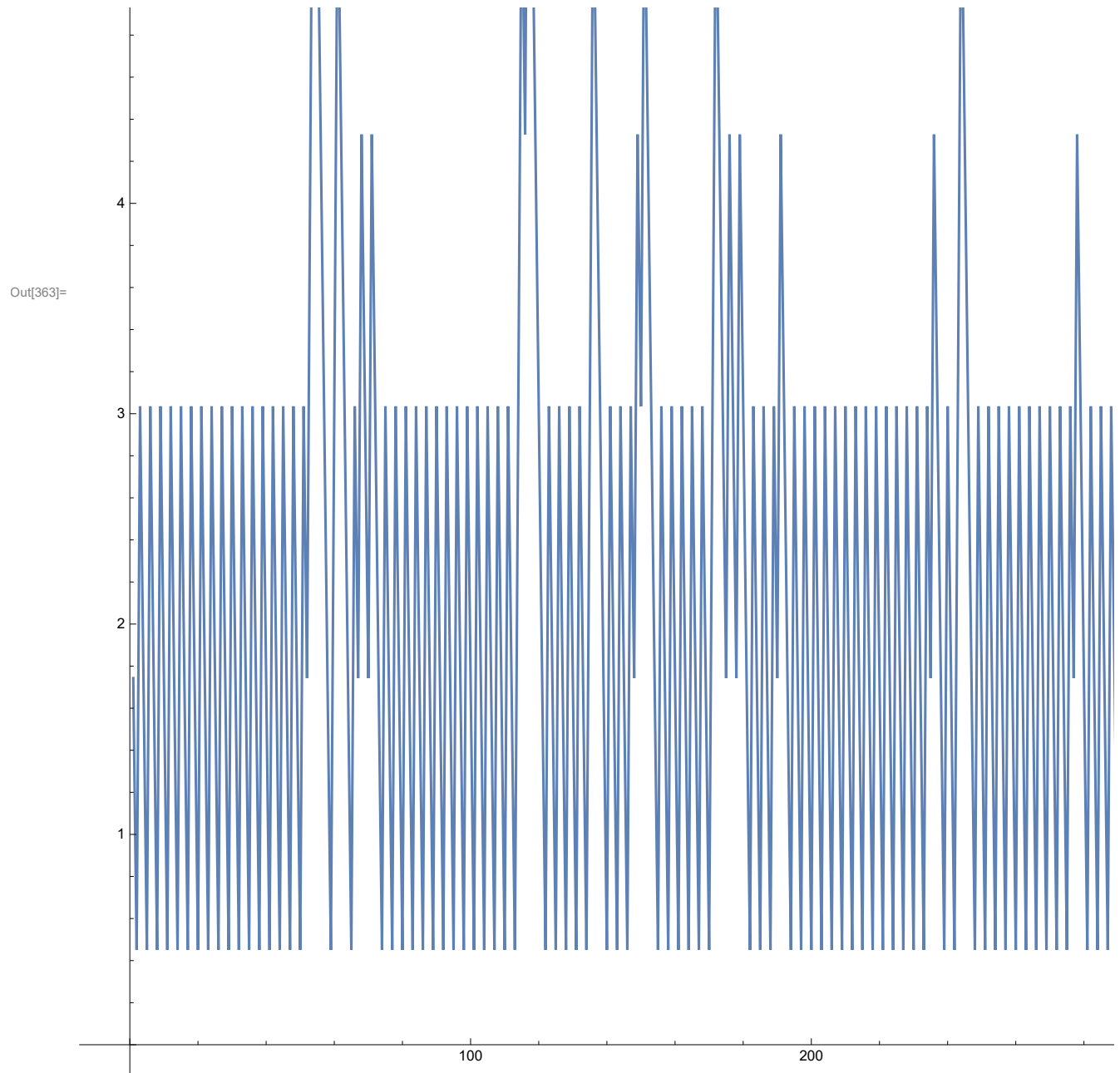
```

ListLinePlot[valor] (*Sondeo de lospresios maximos*)

Median[valor] (*Precio maximo medio*)

preciofinal = B * 2 (*Precio maximo final*)





Out[364]= 1.74284

Out[365]= 0.450403

plusvalia =

$$\left(\frac{\text{preciofinal} - dv1}{dv1} \right) * 100 \text{ (*Depreciacion o apreciacion de las propiedades *)}$$

Out[366]= -85.1611

In[367]:= **p1**

Out[367]= 0.281502

In[368]= **fallo**

Out[368]= 0

In[369]= **check**

Out[369]= 500 341

In[370]= **mudanza**

Out[370]= 237

In[371]=

```

video = Manipulate[ArrayPlot[animacion[[u]], Mesh → True,
  [manipula [representación de arreglo [malla [verdadero
    ColorRules → {0 → White, 1 → Blue, 2 → Red, 3 → Black}], {u, 1, mudanza, 1}];
  [reglas de color [blanco [azul [rojo [negro
In[153]= Export["inmobiliariadistribucionmexico11.avi", video]
  [exporta

```

Out[153]= inmobiliariadistribucionmexico11.avi

(*Calculo de los valores de conformidad o felicidad para cada mudanza*)

```

For[k = 1, k < mudanza, k ++,
  [para cada
    vecinos = 0;
    estado = 0;
    q = k * 10;
    arreglo2 = animacion[[k]];
    For[i = 1, (i < long + 1), i ++, For[j = 1, (j < long + 1), j ++,
      [para cada [para cada
        If[(i - 1) < 1, i1 = long, i1 = (i - 1)];
        [si
        If[(i + 1) > long, i2 = 1, i2 = (i + 1)];
        [si
        If[(j - 1) < 1, j1 = long, j1 = (j - 1)];
        [si
        If[(j + 1) > long, j2 = 1, j2 = (j + 1)];
        [si
        If[arreglo2[[i1, j1]] ≥ arreglo2[[i, j]] && arreglo2[[i1, j1]] ≠ 0,
        [si
          vecinos = vecinos + 1 ;
        If[arreglo2[[i1, j2]] ≥ arreglo2[[i, j]] && arreglo2[[i1, j2]] ≠ 0,
        [si
          vecinos = vecinos + 1 ;
        If[arreglo2[[i2, j1]] ≥ arreglo2[[i, j]] && arreglo2[[i2, j1]] ≠ 0,
        [si

```

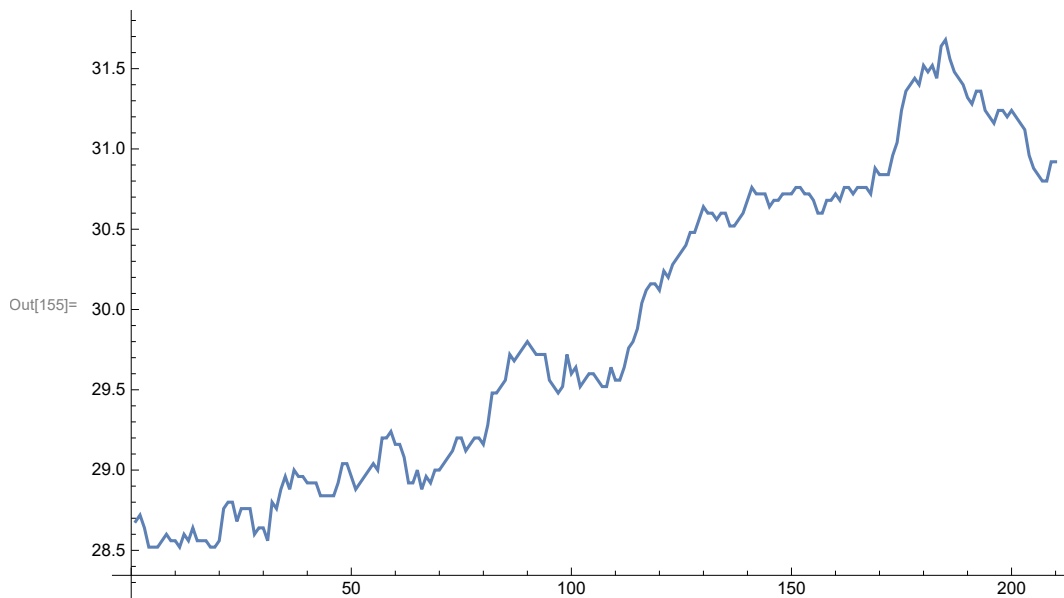
```

↳
vecinos = vecinos + 1 ;
If[arreglo2[[i2, j2]] ≥ arreglo2[[i, j]] && arreglo2[[i2, j2]] ≠ 0,
↳
vecinos = vecinos + 1 ;
If[arreglo2[[i1, j]] ≥ arreglo2[[i, j]] && arreglo2[[i1, j]] ≠ 0,
↳
vecinos = vecinos + 1 ;
If[arreglo2[[i2, j]] ≥ arreglo2[[i, j]] && arreglo2[[i2, j]] ≠ 0,
↳
vecinos = vecinos + 1 ;
If[arreglo2[[i, j1]] ≥ arreglo2[[i, j]] && arreglo2[[i, j1]] ≠ 0,
↳
vecinos = vecinos + 1 ;
If[arreglo2[[i, j2]] ≥ arreglo2[[i, j]] && arreglo2[[i, j2]] ≠ 0,
↳
vecinos = vecinos + 1 ;
Switch[arreglo2[[i, j]], 0, If[(vecinos / 8) >= tolerancia1, estado = estado + 1], 1,
↳
evalúa dependiendo de ↳
If[(vecinos / 8) >= tolerancia2, estado = estado + 1], 2, If[(vecinos / 8) >= tolerancia3,
↳
↳
estado = estado + 1], 3, If[(vecinos / 8) >= tolerancia4, estado = estado + 1]];
↳

vecinos = 0; ]];
estado = (estado / (long * long)) * 100;
AppendTo[conformidad, estado];];
↳
añade al final

```

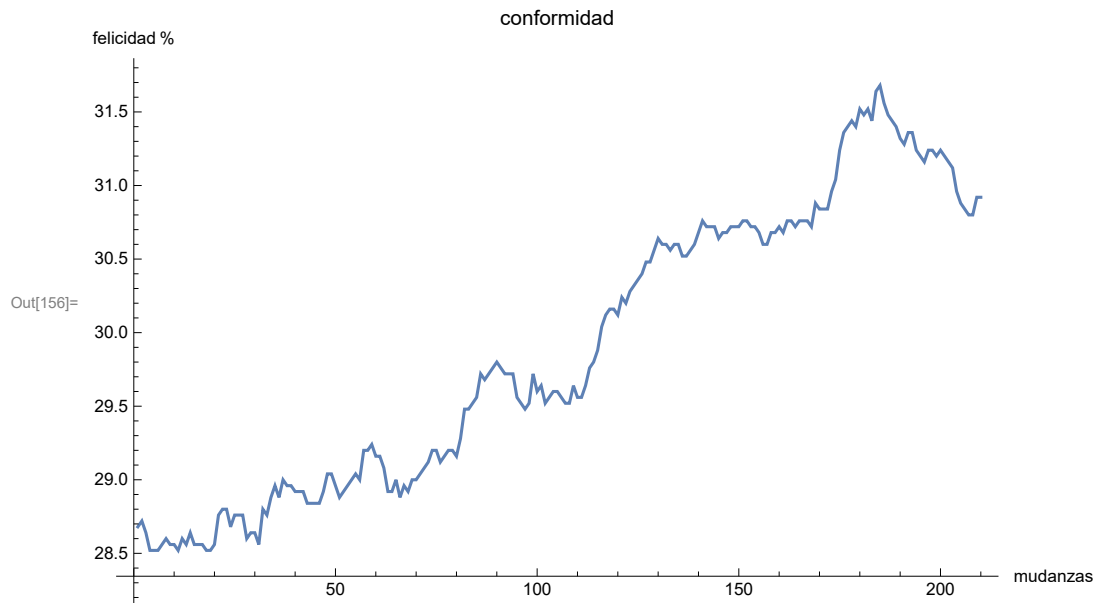
In[155]:= ListLinePlot[conformidad]
↳ gráfico de línea de una lista



```

In[156]:= Show[%155, AxesLabel -> {HoldForm[mudanzas], HoldForm[felicidad %]},
[muestra [etiqueta de ejes [forma sin evaluación [forma sin evaluación
PlotLabel -> HoldForm[conformidad], LabelStyle -> {GrayLevel[0]}]
[etiqueta de r... [forma sin evaluación [estilo de etiqueta [nivel de gris

```



Capítulo 5

Modelos inspirados en la mecánica estadística

Hasta este punto, el concepto de autómatas celulares ha servido como punto de partida para afrontar los problemas generados por la enorme cantidad de agentes e información contenida en los sistemas sociales y económicos. No obstante la información que pudimos obtener de los sistemas fue de un carácter más cualitativo y empírico al no contar con un marco teórico para la recaudación y procesamiento de la información. En esta sección partiremos de las definiciones clásicas de la mecánica estadística como la entropía así como las herramientas que nos provee la estadística y el estudio de las variables aleatorias en general, para desarrollar un marco propio de la econofísica para procesar la información y obtener resultados más formales y de un carácter enteramente cuantitativo.

5.1. conceptos generales

Desde tiempos de Leibniz, Galileo, Laplace y Hobbes, la física, las matemáticas, la sociología y la economía han avanzado de forma no tan independiente como se podría pensar, los desarrollos de cada una de las disciplinas han hecho numerosas y cuantiosas aportaciones a las demás pero incluso la forma en que algunas han evolucionado a través de la historia tiene muchas similitudes. En particular la física, desarrolló dos vertientes para estudiar la naturaleza, por un lado la enfocada al estudio de los cuerpos, colisiones y movimientos de pocas partículas, cuya descripción se consideraba completa por la mecánica, y por otro lado se encontraba a la termodinámica surgida de los procesos industriales y enfocada en el estudio de sistemas de muchas partículas empleando argumentos fenomenológicos para su descripción.

Aparentemente física contaba con un marco completo para el estudio de la naturaleza, no obstante nuevos fenómenos que no podían ser explicados por ninguno de los dos enfoques, fomentaron el desarrollo de una nueva rama, la mecánica estadística. La mecánica estadística tiene como objetivo encontrar argumentos microscópicos que implican las interacciones más sencillas entre partículas para describir fenómenos macroscópicos. En general también encuentra relaciones directas entre cantidades y magnitudes termodinámicas con las propias de la mecánica como la temperatura, con el promedio de la energía cinética de las partículas del sistema o la presión con la cantidad y fuerza de los impactos que realizan las partículas sobre las paredes del recipiente.

Al igual que la física la economía y la sociología se enfrenta a paradigmas similares, cuenta con dos enfoques para el estudio de su universo, por un lado estudia el comportamiento individual de cada uno de los agentes, que cuenta con su propio marco teórico, definiciones y magnitudes propias, por otro lado estudia el entorno en general del sistema, con cientos de miles de agentes interactuando de diversas formas, pero estudiando y analizando únicamente las propiedades macroeconómicas o macro-sociales del sistema. Al igual que en la física y como se puede observar de los capítulos anteriores el centrar el estudio en los efectos macros pero con argumentos micro, arroja nueva información sobre la naturaleza económica y social del sistema.

Entropía

El concepto de la entropía, dentro de las ciencias sociales es quizá uno de los que más a permeado dentro del medio, por ejemplo la entropía o un neologismo de esta, asocia a la entropía en el intercambio de información de un sistema social con su grado óptimo de comunicación, por ejemplo una baja entropía en el sistema implicaría un muy bajo intercambio de contenido y por ende una comunicación muy pobre, por otro lado si la entropía es muy elevada implicaría que hay un exceso en el tránsito de la información, ocasionando que en conjunto la información circulante no sea más que ruido en el sistema, el estado óptimo sería uno con un nivel moderado en la entropía.

Aunque el ejemplo anterior sirve para ejemplificar como los conceptos propios de la física se adentran cada vez más no resulta ser del todo correcto o acertado y no nos brinda una forma cuantitativa de introducir a la variable. Para aproximarnos de una forma más analítica podemos partir de dos definiciones o interpretaciones del concepto, la entropía de Boltzmann y la entropía de Gibbs, pero con una redefinición orientada a sistemas sociales o informáticos.

El concepto de entropía de Boltzmann viene dado por la idea de microestado. Un sistema termodinámico con variables termodinámicas dadas va a estar compuesto por un microestado que está especificado por la configuración de los estados de los componentes del sistema, estos estados están sujetos a constricciones dadas por las variables termodinámicas como presión, volumen o energía.

En la interpretación de Boltzmann, la entropía va a estar dada por el número de microestados accesibles para la configuración dada del sistema. El estado más probable y estable será el que contenga la mayor cantidad de microestados accesibles compatibles con las restricciones.

$$S(X) = \log \Omega \quad (5.1)$$

Donde Ω son los microestados accesibles del sistema, si el logaritmo es base dos la entropía estará dada en forma de bits, si el logaritmo es base e estará dado en nats. Al contar el número de estados o de configuraciones posibles que guarda el sistema podemos pensar en la entropía de Boltzmann como la cantidad de información que almacenará el sistema.

La interpretación de Gibbs desde el punto de vista de la mecánica estadística consiste en integrar la interacción entre las partículas, para los fines que perseguimos podemos entender el concepto de entropía de Gibbs como la medida de la forma en que están distribuidas las probabilidades de ocurrencia de alguna variable aleatoria X .

$$S(X) = - \sum (P(X_i) \log(P(X_i))) \quad (5.2)$$

Donde $P(X_i)$ es la probabilidad de al realizar un muestreo obtener a la variable X_i , en esta interpretación podemos observar que para un sistema cuyos estados correspondientes a la variable aleatoria X son todos iguales la entropía asociada es cero, y será máxima cuando tengamos una distribución uniforme.

5.2. Loterías bilaterales

Para observar con mayor claridad como se incorporan estos conceptos al ámbito económico y social podemos pensar en el ejemplo más simple de un mercado, un juego de lotería donde un grupo de agentes posee una cantidad de dinero inicial dada, al azar se seleccionan dos agentes, un perdedor y un ganador, el perdedor le cede una cantidad definida de dinero al ganador y la lotería prosigue a elegir otros dos agentes hasta los pasos que se le hayan asignado al sistema.

Este sencillo ejemplo guarda una serie de cuestiones nada triviales dentro de la econofísica por una parte esconde la analogía más empleada en el medio donde se estudian los intercambios económicos como colisiones dentro de un sistema de partículas cuyo momento se define como la cantidad de dinero que poseen, haciendo más que sugerente emplear el concepto de entropía. Por otro lado desde el punto de vista de los autómatas celulares no es más que una versión muy simplificada del mercado inmobiliario generado en el capítulo dos por lo que las conclusiones obtenidas de este modelo se pueden extrapolar de forma casi inmediata al modelo inmobiliario.

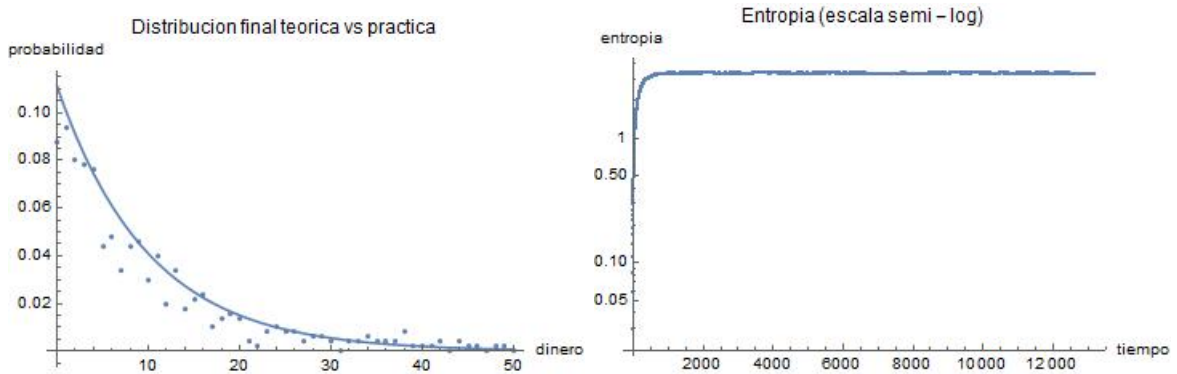
Arquitectura

- El sistema no poseerá una geometría definida.
- Cada agente estará caracterizada por una cantidad dada de dinero. De inicio todos los agentes van a poseer la misma cantidad de dinero m_i .

Dinámica

- El sistema evolucionara de forma asíncrona es decir cada una de los agentes cambiara de estado a diferentes tiempos.
- El sistema selecciona dos agentes de forma aleatoria.
- Si el perdedor cuenta con el dinero para pagar (condición de no bancarrota) se hace el intercambio del perdedor al ganador.
- Se calcula la distribución del dinero después de cada intercambio y la entropía asociada al sistema.

Resultados



(a) Distribución final vs distribución teórica. (b) Entropía a través del tiempo con valor máximo de $\log(50)$.

Figura 5.1: Resultados para un arreglo de 500 agentes con una cantidad de dinero inicial m_0 de 10 y 20000 iteraciones del algoritmo .

El comportamiento del sistema a lo largo del tiempo queda descrito por la figura 5.1. En la imagen (5.1 a) se observa la distribución final, comparado con la gráfica teórica $\frac{e^{-\frac{m}{m^*}}}{\sum -\frac{m}{m^*}}$. Por otro lado en la figura (5.1 b) vemos que rápidamente la entropía se estabiliza en un valor máximo aproximado de $\log(50)$ que corresponde a $\log(500/10) = \log(N/m)$, valor máximo esperado de forma teórica.

Este es un ejemplo claro de un equilibrio dinámico en torno a la entropía y a la distribución del dinero, aunque se siguen realizando intercambios la distribución del dinero y la entropía ya no cambian con el tiempo, si extrapolamos este concepto al modelo inmobiliario de las secciones anteriores podemos ver que la distribución ya no cambia para el caso donde se inicio con una distribución de tipo Pareto, por lo que para un mercado ideal sin la intervención de banca (en ambos casos se empleo la condición de no bancarrota) la distribución de la riqueza siempre sera próxima a una distribución de Pareto que se mantendrá constante a través del tiempo aun cuando se sigan realizando intercambios a gran escala.

Capítulo 6

Modelos interdisciplinarios (Epidemias)

Como se puede observar en la actualidad mundial, existen pocos fenómenos que exijan una cooperación y coordinación de todas las áreas del conocimiento como lo es el escenario de una epidemia global. Es obligación de las diferentes academias y áreas de conocimiento dotar de los argumentos necesarios a los gobiernos y a la población en general para llevar a cabo las mejores medidas posibles.

Con argumentos de carácter científicos y experimentales el presente trabajo plantea una crítica moderada a las medidas efectuadas por los diferentes gobiernos y organizaciones encargadas de realizar y ejecutar los protocolos de salud ante escenarios epidémicos, evadiendo cualquier carácter u orientación política.

6.1. Revoluciones científicas, menos Popper mas Kuhn

En su curso de epistemología, el profesor Mario Bunge plantea que el enfoque a seguir de las ciencias pero en específico, de las ciencias sociales, debe ser el trazado por Thomas Kuhn y se deben abandonar las ideas positivistas planteadas por una corriente de pensadores y científicos liderada por Karl Popper.

Thomas S. Kuhn define el proceso en el que la ciencia se desarrolla y evoluciona como el ciclo de una revolución de la siguiente forma. La ciencia normal que él define se encarga de estudiar y hacer coincidir la teoría actual con los experimentos más recientes, esta teoría y experimentos constituyen el paradigma del momento que podemos definir como la verdad que se encuentra en disputa entre los diferentes saberes científicos.

El paradigma actual poco a poco se ve comprometido por la acumulación de anomalías, las anomalías son los puntos de quiebre o ruptura de las teorías actuales, fenómenos que no pueden ser explicados con las condiciones actuales, cuando un suficiente número de anomalías se presentan se entra en una etapa de crisis que exigen el cambio del paradigma. Con el cambio de paradigma viene el progreso en forma de una nueva visión del universo.

En resumen, para Kuhn el objetivo de la ciencia radica en conseguir una visión renovada del universo, el conseguir explicar e integrar a la sala de trofeos de la razón más fenómenos, o el simple hecho de poder entender un fenómeno en cuestión desde una perspectiva completamente nueva. Bajo esta consigna se van a integrar modelos y algoritmos empleados en la econofísica y en particular en los capítulos anteriores del presente trabajo para generar una perspectiva renovada en torno al cómo entendemos a las pandemias, para integrar en un solo marco conceptual los factores propios de las características intrínsecas del virus y su propagación con los factores provenientes de la dinámica social, como son diferentes estructuras sociales y la interacción entre los agentes de dichas estructuras.

6.2. Epidemias

6.2.1. Modelo SIR

Para modelar la propagación de un agente infeccioso dentro de una red, en particular una epidemia, el mecanismo más usado es el modelo S.I.R (Por sus siglas susceptibles, infectados y recuperados) que consiste en un conjunto de ecuaciones diferenciales que describen la evolución de las tres diferentes comunidades, cada una de ellas con su tasa de crecimiento. En general las ecuaciones diferenciales no son lineales y son sumamente sensibles a cambios en las condiciones iniciales.

Este modelo describe con lujo de detalle las grandes epidemias del pasado como es el caso de la gripe española y en general para cualquier epidemia que haya concluido es relativamente fácil encontrar los parámetros que ajusten al modelo para describir a la epidemia en particular sin embargo el modelo resulta insuficiente y poco práctico para realizar predicciones sobre una epidemia en curso. Otro problema considerable del modelo es que para integrar factores como la movilidad es necesario incluir factores de difusión a las ecuaciones lo que vuelve al modelo casi totalmente inviable de obtener una solución o al menos resulta sumamente complicado.

6.2.2. Modelo con autómatas celulares

La forma perfecta de estudiar una pandemia es observar su evolución, para obtener las características de la propagación de un virus en una sociedad o comunidad no existe mejor forma que tomar un grupo de control y ponerlo en interacción con un grupo de infectados, no obstante este sería un procedimiento sumamente falto de ética. Para sortear los problemas éticos y morales podemos llevar a cabo dicho experimento en un grupo de individuos virtuales, si partimos nuevamente de los autómatas celulares podemos simplemente generar un arreglo de autómatas finitos con la información necesaria para replicar el comportamiento de un grupo de control de personas reales.

El primer paso es estudiar las propiedades de propagación del virus, para ello colocamos una serie de agentes susceptibles dentro de un arreglo de autómatas celulares, posteriormente se coloca de forma aleatoria un grupo de agentes infectados. Los agentes susceptibles van a poseer una probabilidad de contagio por cada vecino infectado que tengan, para cada paso de tiempo de forma aleatoria siguiendo la probabilidad calculada los agentes susceptibles pasaran a ser agentes contagiados. El grupo de los infectados va a contar con un contador que registrara el tiempo de la enfermedad, para este modelo se supone que el infectado se recupera o muere en un tiempo dado (vida media del virus), al cumplir su tiempo el agente infectado pasa a pertenecer al grupo de los recuperados.

6.2.3. Modelo con movilidad

En un mundo tan conectado, justo en el "apogeo" de la globalización, la restricción de la movilidad se ha convertido en el tema central de las políticas y medidas de contingencia contra la epidemia. Para integrar el factor de la movilidad a nuestro modelo primero necesitamos disponer de espacio en el arreglo (A.C), ahora el sistema contara con muchos mas espacios vacíos, posteriormente para que el movimiento no se resuma al absurdo de partículas suspendidas en un gas, con movimiento errático vamos a construir comunidades dentro del arreglo. Para construir a las comunidades emplearemos el algoritmo utilizado para llevar a cabo la segregación racial descrita por el modelo de Schelling del capítulo dos. Una vez construidas las comunidades, colocamos a los agentes infecciosos en lugares aislados a las comunidades. Además de los factores de probabilidad de contagio y vida media, en este modelo incluimos una probabilidad o grado de movilidad a los agentes infectados, esta probabilidad determinara si los agentes infectados se van a incorporar a alguna de las comunidades establecidas.

6.2.4. Modelo con comunidades

Finalmente podemos incorporar comunidades con características diferentes en cuanto a movilidad o susceptibilidad a infectarse, de esta forma podemos estudiar que comportamientos emergen de la interacción de dos comunidades con diferente actuar, en el contexto actual podríamos compararlo con comunidades renuentes a llevar a cabo las medidas de prevención y reducción de la movilidad. En este modelo nos centraremos en el desplazamiento de los susceptibles, ahora el movimiento de los infectados sera completamente aleatorio sin preferencia por incorporarse a las comunidades, mientras que los susceptibles se van a desplazar en la periferia de sus comunidades. El algoritmo es idéntico al anterior solo que ahora el sistema reconoce a dos grupos susceptibles.

Resultados, modelo base

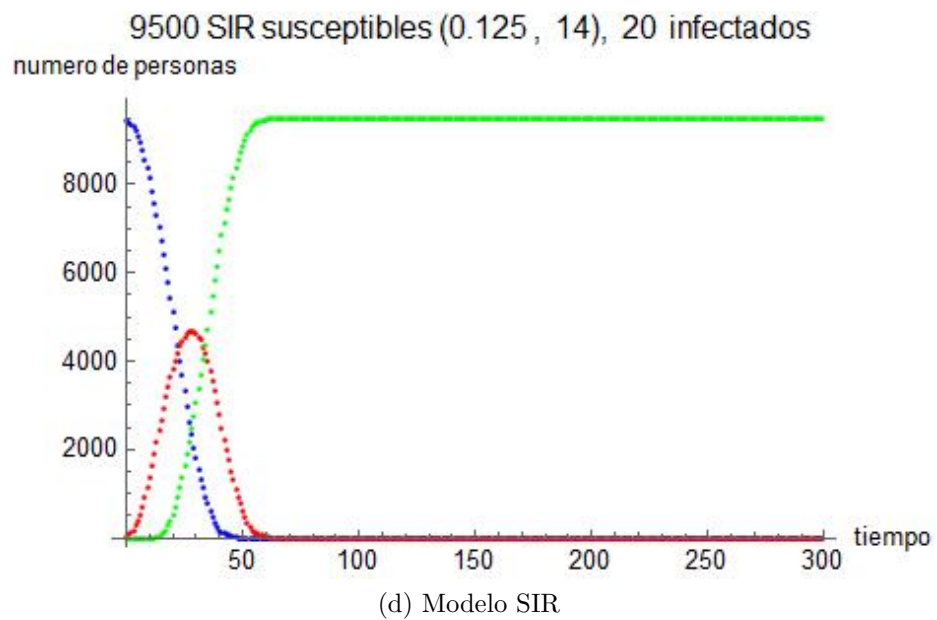
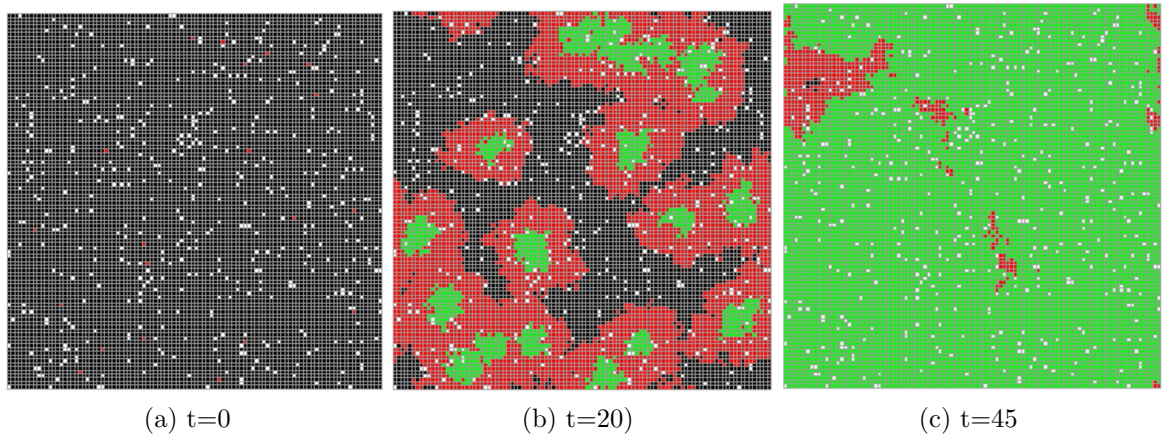
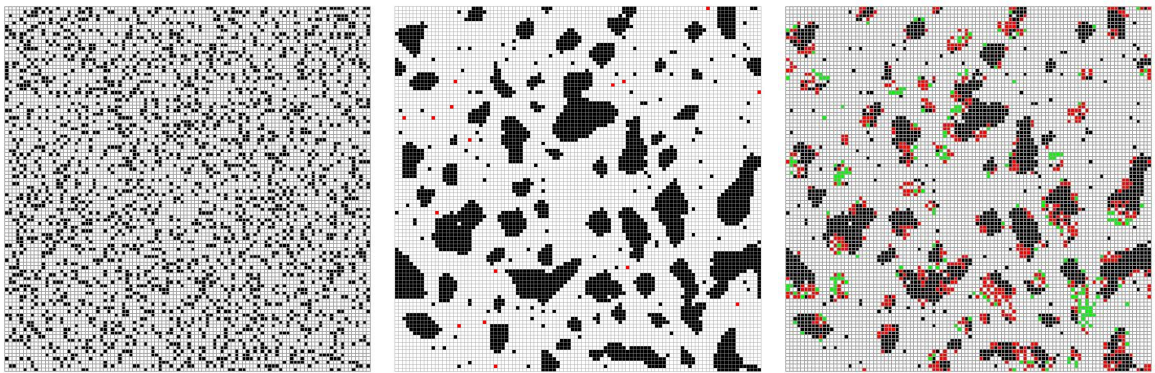


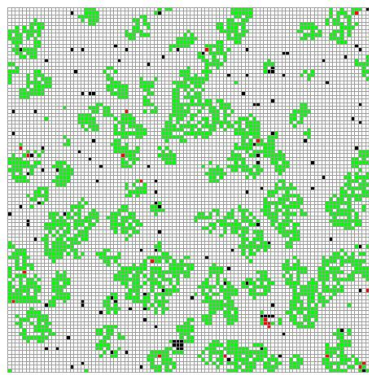
Figura 6.1: Evolución del autómata celular con 9500 agentes susceptibles al inicio y 20 infectados con una probabilidad de contagio de 0.125 y una vida media del virus de 14 días.

Resultados, modelo con movilidad (caso 1)

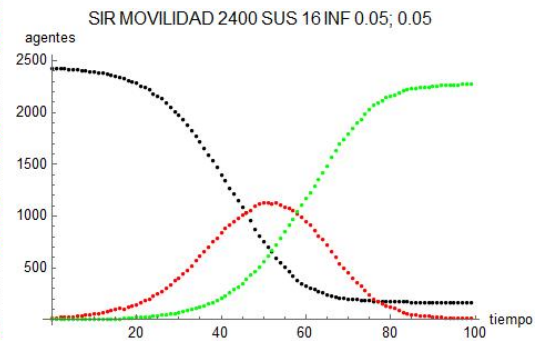


(a) agentes antes de agruparse (b) agentes agrupados, con agentes infecciosos incorporados

(c) $t=50$



(d) $t=100$



(e) Modelo SIR

Figura 6.2: Evolución del autómata celular con 2400 agentes susceptibles al inicio y 16 infectados con una probabilidad de contagio de 0.05, una vida media del virus de 18 días y una movilidad de 0.05.

Resultados, modelo con movilidad (caso 2) (movilidad moderada)

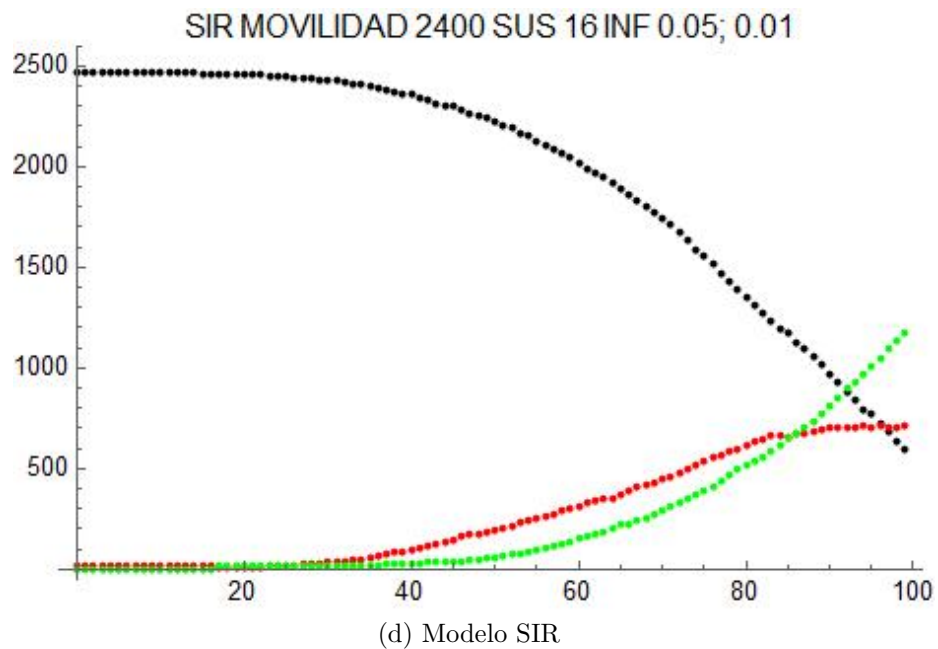
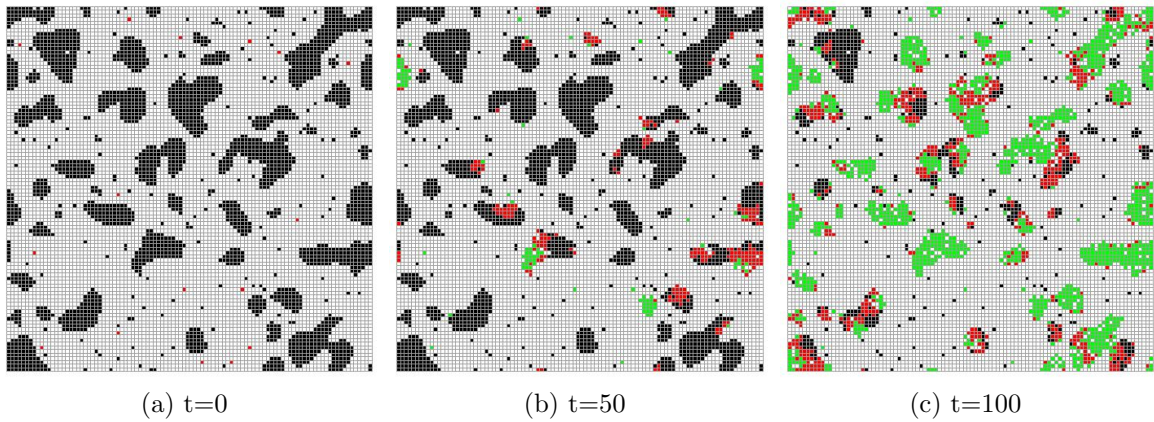


Figura 6.3: Evolución del autómata celular con 2400 agentes susceptibles al inicio y 16 infectados con una probabilidad de contagio de 0.05, una vida media del virus de 18 días y una movilidad de 0.01.

Resultados, modelo con comunidades

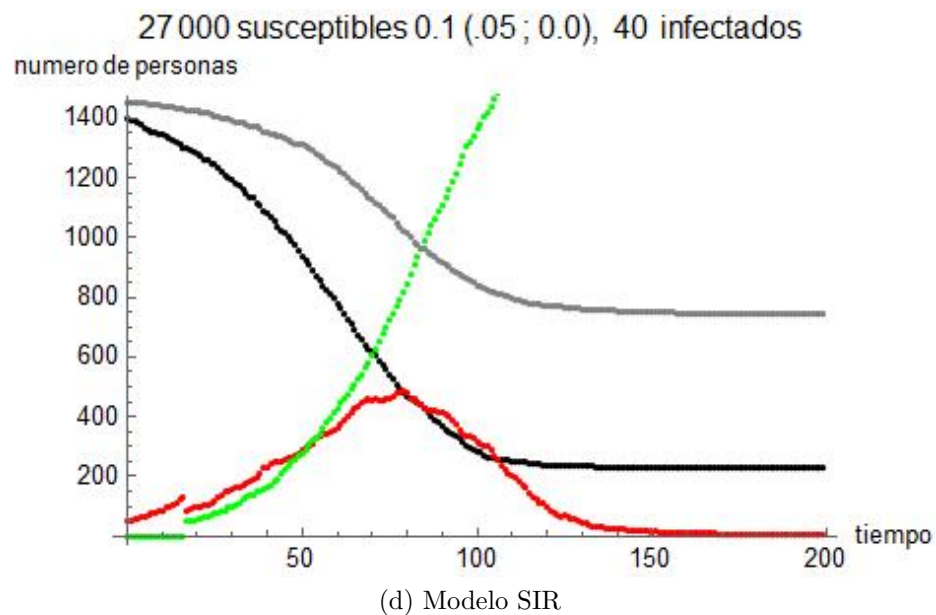
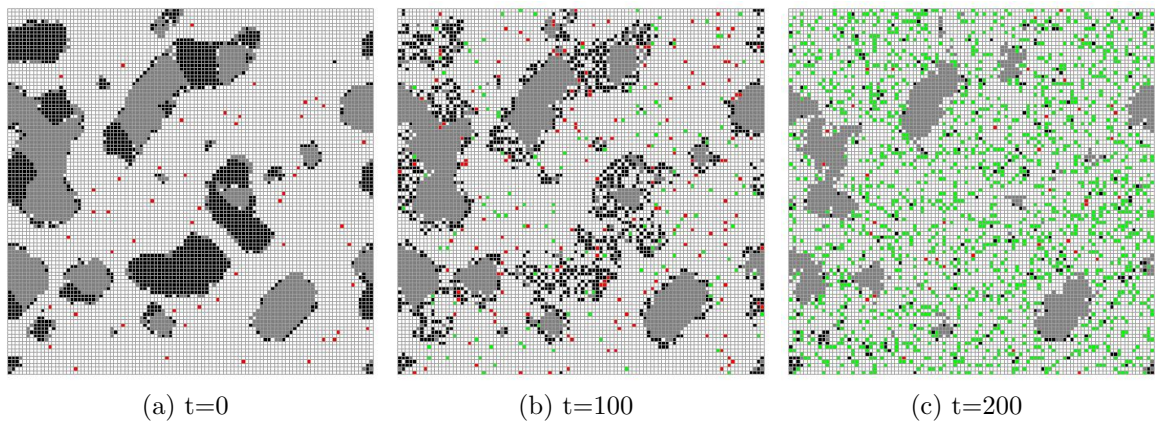


Figura 6.4: Evolución del autómata celular con 2700 agentes susceptibles al inicio y 40 infectados con una probabilidad de contagio de 0.1, una vida media del virus de 18 días y una movilidad de 0.05 para la comunidad negra con la comunidad gris estática.

Resultados, modelo con comunidades, probabilidad de contagio vs movilidad

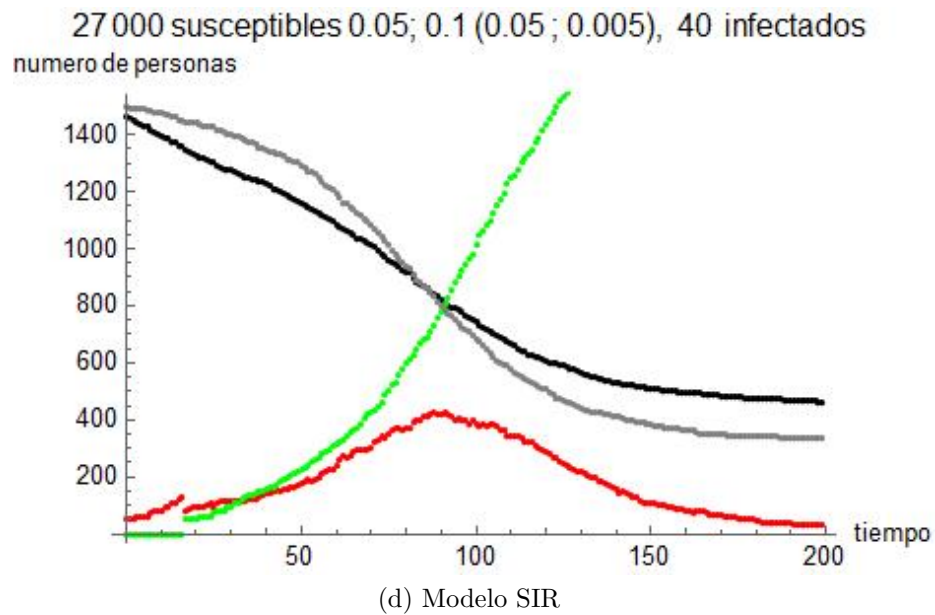
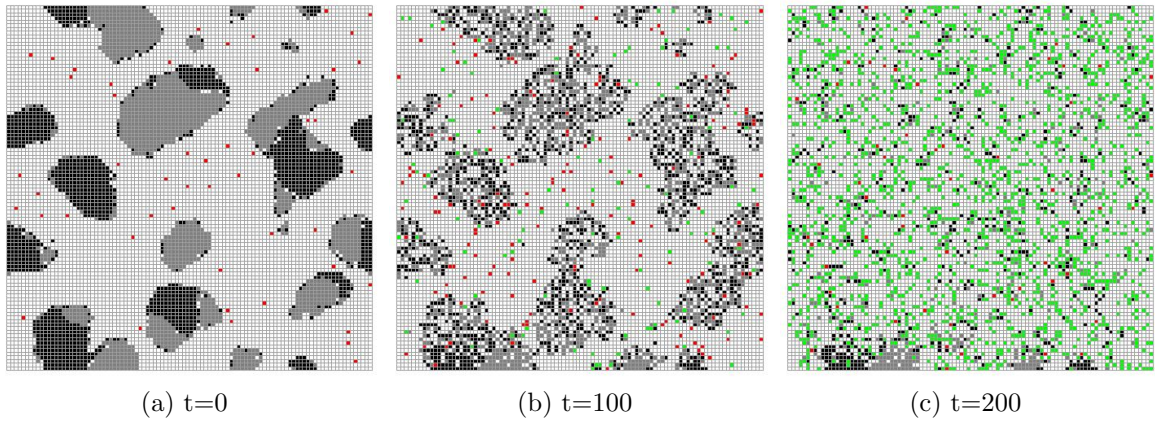


Figura 6.5: Evolución del autómata celular con 2700 agentes susceptibles al inicio y 40 infectados, una vida media del virus de 18 días, una probabilidad de contagio de 0.1 para la comunidad gris, 0.05 para la comunidad negra y una movilidad de 0.05 para la comunidad negra con una probabilidad de 0.005 para la gris.

Resultados, modelo con comunidades, cooperación y solidaridad

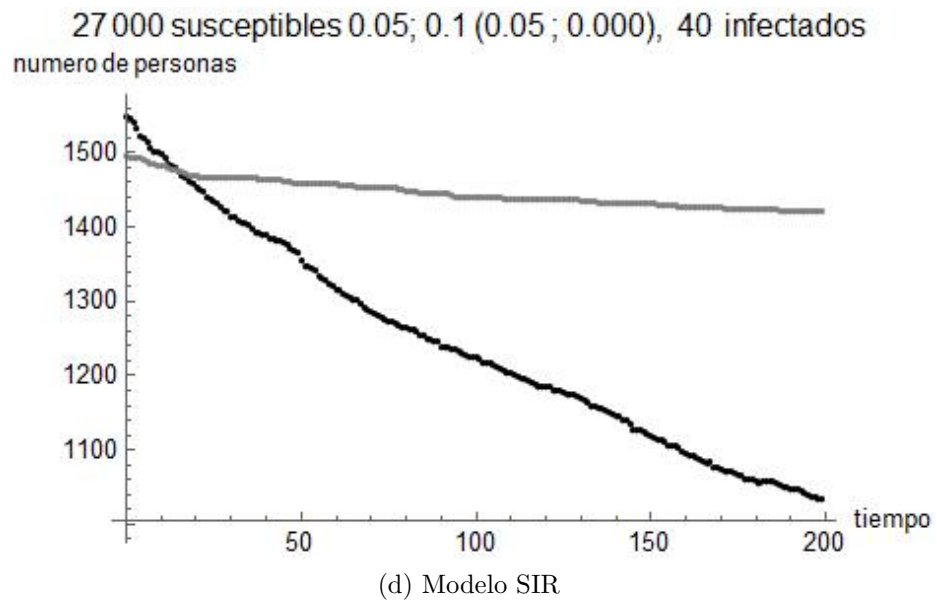
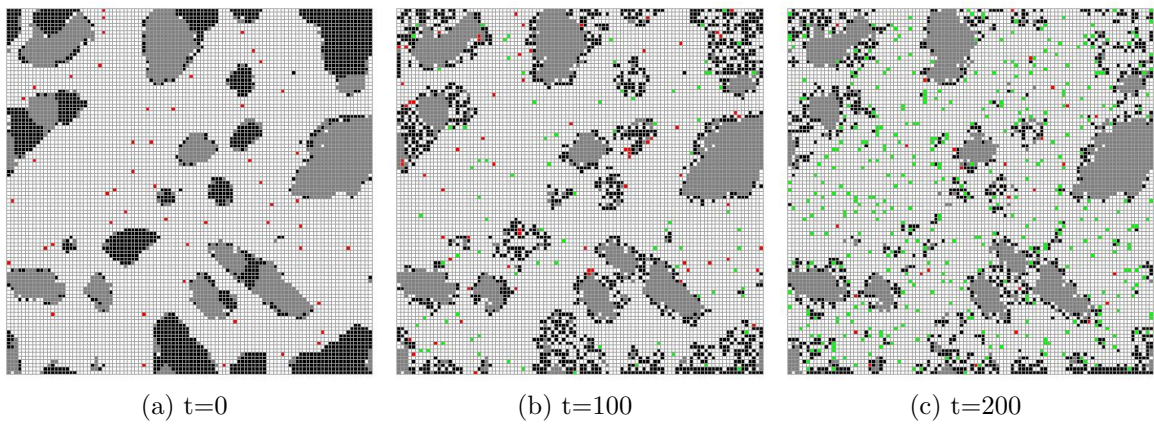


Figura 6.6: Evolución del autómata celular con 2700 agentes susceptibles al inicio y 40 infectados, una vida media del virus de 18 días, una probabilidad de contagio de 0.1 para la comunidad gris, 0.05 para la comunidad negra y una movilidad de 0.05 para la comunidad negra con la comunidad gris estática.

6.3. Crítica y conclusiones

6.3.1. vida media vs probabilidad de contagio

Del modelo base para este sistema de epidemia con autómatas celulares, podemos concluir que es mucho más significativo lo infeccioso del virus que la vida media del mismo, una vida media muy elevada pero con una capacidad de contagio baja generara una enfermedad endémica pero difícilmente encontraremos el comportamiento de una pandemia. Que la probabilidad de contagio sea tan relevante en el modelo invita a la comunidad a reflexionar sobre las medidas de prevención impuestas, como el uso del cubre bocas ya que todas estas medidas contribuyen a disminuir la velocidad y la magnitud del pico de la pandemia.

6.3.2. Medidas oportunas y rigurosas (México, Europa y el mundo)

Conforme la pandemia del COVID-19 avanza, en el mundo se pone en tela de juicio el accionar de los gobiernos ante la crisis. Podemos resumir el debate y el contraste de las políticas implementadas, como un conflicto de intereses económicos y médicos. Tanto los factores de salud como los factores económicos son responsables de que día a día se sigan perdiendo vidas mientras el conflicto de prolongar la pandemia a costa de empleos y por ende del sustento de las familias o relajar las medidas y exponer a la población a la enfermedad continua. Se puede entonces plantear las siguientes preguntas, ¿Se puede encontrar el punto óptimo para efectuar las medidas necesarias, priorizando ambos aspectos?, ¿Como y cuando efectuar las medidas? y ¿Estamos a tiempo para tomar ese camino?.

Con base en los resultados anteriores podemos responder a la primer pregunta con un rotundo si, ahora lamentablemente la respuesta a la ultima pregunta es un desalentador no. Las medidas pertinentes se debieron tomar ayer, en específico en los primeros días del 2020 o un poco antes, del modelo se puede determinar que el momento ideal para imponer medidas rigurosas de confinamiento es al inicio de la epidemia, al retrasar o evitar por completo la integración de los agentes infecciosos a las comunidades podemos prevenir el escenario epidémico. Cuanto más prontas y más rigurosas sean las medidas menos prolongadas deben de ser y menos peligrosos los posibles escenarios que se deben afrontar después, un enero de cuarentena global o mejor aun una ciudad de Wuhan en cuarentena total en diciembre son los escenarios óptimos, después cada país tuvo una ventana de oportunidad para ponerse en cuarentena preventiva de forma oportuna, ventanas que sin excepciones se dejaron cerrar. Podemos concluir entonces que en la totalidad de los gobiernos existe una sistemática negligencia, no solamente durante el desarrollo de la pandemia, si no también de forma más alarmante y descarada antes de esta.

La falta de protocolos homologados y efectivos para enfrentar una epidemia en pleno siglo XXI colocan a cada una de las muertes actuales por esta enfermedad y sus consecuencias colaterales a un nivel de demencia tan grande como la ultima escena de la pelicula Dr. Strangelove de Kubrick donde un soldado monta una bomba atómica cual vaquero, en una escena disparatada y cómica que termina con el fin de la tierra y el exterminio de la raza humana.

Una vez dicho que ningún gobierno ha actuado de forma correcta, si podemos ponderar y analizar su ineptitud con base en los modelos anteriores, el caso europeo en concreto de España e Italia en contraste con México ejemplifican que en este peculiar escenario es preferible actuar antes y de forma algo tibia que actuar tarde de forma enérgica, en ambos casos no se detuvo la movilidad a tiempo sin embargo México adelanto medidas, al menos en comparación con sus homólogos europeos ganando algo de tiempo, de estos países europeos aprendimos como actuar de la forma mas incompetente posible ante estos fenómenos, ejemplo que lamentablemente U.S.A siguió al pie de la letra replicando los desastrosos resultados, tanto USA como Europa se asemejan a la figura (6.1), donde podemos observar un pico epidémico muy marcado, México por su parte se parece mas a la figura (6.3) aunque con una curva menos aplanada y aun en crecimiento.

6.3.3. No todo esta perdido (Cooperación y medidas de prevención)

El ultimo modelo arroja resultados muy interesantes o cuanto menos muy sugerentes, primeramente al comparar la movilidad con la probabilidad de contagio resulta mucho mas relevante para la evolución del sistema la probabilidad de contagio. En un escenario como el que atraviesa México donde la pandemia aun esta proceso y en pleno crecimiento pero que a su vez la cuarentena parece insostenible mucho tiempo mas, por nuestras peculiares y precarias condiciones socioeconómicas, resulta sumamente importante este resultado, si el gobierno y en general nuestras estructuras sociales no pueden hacer nada frente a este fenómeno, si se puede formar un frente de batalla desde el actuar de cada miembro de la sociedad, evidentemente si puedes continuar la cuarentena resulta imperativo respetarla pero si no esta dentro de las posibilidades aun puedes emplear cobrebocas, caretas y todo lo posible para resguardarte y resguardar a los demás.

Finalmente la cooperación y el sentido de responsabilidad no solamente yacen en el hecho de que cada miembro de la sociedad se ponga en riesgo por su propia cuenta si no que al hacerlo pone en riesgo a los demás integrantes, es difícil encontrar argumentos apegados a las ciencias formales en áreas sociales sin embargo en las figuras (6.5) y (6.6), los encontramos, ambos modelos solo difieren en la movilidad del grupo gris, cuando este se mueve (rompe la cuarentena sin medidas de protección) sus contagios se disparan pero sorpresivamente también dispara los contagios en su comunidad vecina de agentes negros.

Capítulo 7

Conclusión

Retomando las ideas de Kuhn, el objetivo del presente trabajo queda mas que superado, si bien soy modesto en las aspiraciones sobre el impacto de esta investigación y dudo completamente que algo tenga que aportar al curso de la revolución científica, si cambia la forma de entender el universo al menos el universo de las ciencias sociales y desde el punto de vista del autor.

Replicar resultados de disciplinas aparentemente tan distantes de la física como es el caso de la teoría económica y social desde el punto de vista de un profesional de las ciencias duras resulta tan gratificante y fructífero como el ajustar alguna curva en un experimento dado en la física o encontrar la ecuación que describa algún fenómeno en particular que esta siendo estudiado.

En la actualidad se denuncia que las ciencias sociales son un terreno infértil debido a que las escuelas no solo físicas si no de pensamiento están tomadas por grupos políticos radicales impidiendo el desarrollo objetivo debido a que si un argumento es correcto pero puede considerarse políticamente incorrecto este es digno de censura, con este escenario resulta indispensable poder abordar estas disciplinas desde enfoques diferentes que además tengan la bondad de estar de estar exentos de cuestiones personales o políticas al sustentarse en cuestiones numéricas, matemáticas, informáticas o meramente de la lógica pura, este proceso resulta mas necesario cuando se plantea el escenario donde las academias de ciencias básicas también pueden ser tomadas por la demagogia. Urge entonces recurrir a la figura del científico y el político de Weber, para defender a la ciencia en todos los frentes posibles.

«¡Despertad, oh jóvenes de la nueva era! ¡Oponed vuestras frentes a los mercenarios ignorantes! Pues tenemos mercenarios en el campamento, en la corte y en la universidad: los cuales, si pudieran, rebajarían lo mental para siempre y prolongarían la guerra corpórea.» Kenzaburo Oe.

Capítulo 8

Códigos 2

```
In[1]:= long = 50;

tolerancia = 5 / 8;
supertotal = 1000000;
total = 1000;
pasos = 2000;
vidamedia = 14;
infect = 0.20;
pb = 0.4
vida = Table[RandomChoice[{0.7, 0.3, 0.00} -> {0, 1, 2}], {p, long}, {q, long}];
      [tabla [elección aleatoria]
```

Out[8]= 0.4

```
In[10]:= ArrayPlot[vida, Mesh -> True, ColorRules -> {1 -> Black, 0 -> White, 2 -> Red}]
      [representación de a [malla [verd... [reglas de color [negro [blanco [rojo]
```



```
In[11]:= arreglo = vida;
```

```
In[12]:= suma1 = 0;
```

```
In[13]:= check = 0;
```

```
In[14]:= suma2 = 0;
```

```
In[15]:= pasos = 0;
animacion = {arreglo};
fallo = 0;
```

```
In[18]:= While[check < supertotal,
      [mientras
```

```

Label[aleatorio];
  etiqueta
suma1 = 0; suma2 = 0;
ri1 = RandomInteger[{1, (long)}];
  entero aleatorio
rj1 = RandomInteger[{1, (long)}];
  entero aleatorio
ri2 = RandomInteger[{1, (long)}];
  entero aleatorio
rj2 = RandomInteger[{1, (long)}]; pos1 = arreglo[[ri1, rj1]];
  entero aleatorio

pos2 = arreglo[[ri2, rj2]];
If[pos1 == pos2, Goto[aleatorio], check = check + 1];
  si      ve a
If[(ri1 - 1) < 1, ri11 = long, ri11 = (ri1 - 1)];
  si
If[(ri2 - 1) < 1, ri21 = long, ri21 = (ri2 - 1)];
  si
  If[(ri1 + 1) > long, ri12 = 1, ri12 = (ri1 + 1)];
  si
  If[(ri2 + 1) > long, ri22 = 1, ri22 = (ri2 + 1)];
  si
  If[(rj1 - 1) < 1, rj11 = long, rj11 = (rj1 - 1)];
  si
  If[(rj2 - 1) < 1, rj21 = long, rj21 = (rj2 - 1)];
  si
  If[(rj1 + 1) > long, rj12 = 1, rj12 = (rj1 + 1)];
  si
  If[(rj2 + 1) > long, rj22 = 1, rj22 = (rj2 + 1)];
  si

  If[arreglo[[ri11, rj11]] == pos1, suma1 = suma1 + 1];
  si
  If[arreglo[[ri11, rj12]] == pos1, suma1 = suma1 + 1];
  si
  If[arreglo[[ri12, rj11]] == pos1, suma1 = suma1 + 1]; If[arreglo[[ri12, rj12]] == pos1,
  si      si
    suma1 = suma1 + 1]; If[arreglo[[ri11, rj1]] == pos1, suma1 = suma1 + 1];
    si
  If[arreglo[[ri12, rj1]] == pos1, suma1 = suma1 + 1]; If[arreglo[[ri1, rj11]] == pos1,
  si      si
    suma1 = suma1 + 1]; If[arreglo[[ri1, rj12]] == pos1, suma1 = suma1 + 1];
    si

  If[arreglo[[ri21, rj21]] == pos2, suma2 = suma2 + 1];
  si
  If[arreglo[[ri21, rj22]] == pos2, suma2 = suma2 + 1];
  si
  If[arreglo[[ri22, rj21]] == pos2, suma2 = suma2 + 1]; If[arreglo[[ri22, rj22]] == pos2,
  si      si

```

```

suma2 = suma2 + 1]; If[arreglo[[ri21, rj2]] == pos2, suma2 = suma2 + 1];
If[arreglo[[ri22, rj2]] == pos2, suma2 = suma2 + 1]; If[arreglo[[ri2, rj21]] == pos2,
suma2 = suma2 + 1]; If[arreglo[[ri2, rj22]] == pos2, suma2 = suma2 + 1];

felicidad1 = suma1 / 8;
felicidad2 = suma2 / 8;

If[felicidad1 < tolerancia && felicidad2 < tolerancia, fallo = 0;
pasos = pasos + 1;
arreglo[[ri1, rj1]] = pos2; arreglo[[ri2, rj2]] = pos1;
AppendTo[animacion, arreglo], fallo = fallo + 1];
If[mudanza > pasos, check = supertotal];
If[fallo > total, check = supertotal];]

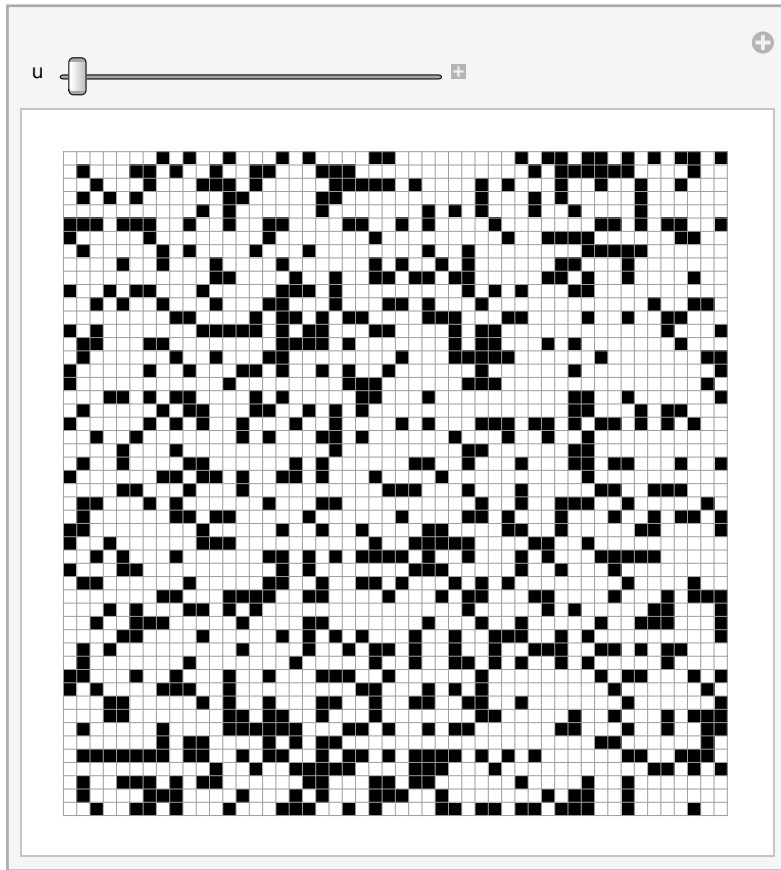
```

```

In[19]:= video = Manipulate[ArrayPlot[animacion[[u]], Mesh -> True,
ColorRules -> {2 -> Red, 1 -> Black, 0 -> White}], {u, 1, pasos, 1}

```

Out[19]=



```

In[20]:= paso = 0;
inf = {};
For[i = 0, i < 10, i ++,
  para cada
    suma1 = 0; suma2 = 0;
    Label[aleatorio2];
    etiqueta
    suma1 = 0;
    ri1 = RandomInteger[[1, (long)]]; rj1 = RandomInteger[[1, (long)]];
    entero aleatorio
    pos1 = arreglo[[ri1, rj1]];

    If[(ri1 - 1) < 1, ri11 = long, ri11 = (ri1 - 1)];
    si
    If[(ri1 + 1) > long, ri12 = 1, ri12 = (ri1 + 1)];
    si
    If[(rj1 - 1) < 1, rj11 = long, rj11 = (rj1 - 1)];
    si
    If[(rj1 + 1) > long, rj12 = 1, rj12 = (rj1 + 1)];
    si

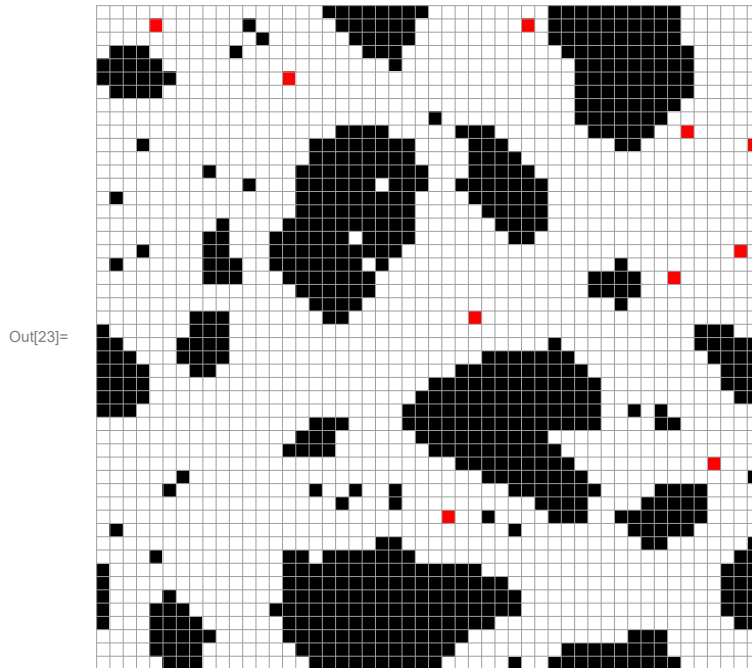
    If[arreglo[[ri11, rj11]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri11, rj12]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri12, rj11]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri12, rj12]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri11, rj1]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri12, rj1]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri1, rj11]] == 1, suma1 = suma1 + 1];
    si
    If[arreglo[[ri1, rj12]] == 1, suma1 = suma1 + 1];
    si

    If[suma1 ≠ 0, Goto[aleatorio2], arreglo[[ri1, rj1]] = 2; AppendTo[inf, {ri1, rj1, 1}]];
    si ve a añade al final
    paso = paso + 1;

];

ArrayPlot[arreglo, Mesh → True, ColorRules → {2 → Red, 1 → Black, 0 → White}]
representación de arreglo malla verd... reglas de color rojo negro blanco

```

In[24]=

In[25]= `pos = inf[[3]]`

Out[25]= `{39, 27, 1}`

In[26]= `pos[[1]]`

Out[26]= `39`

In[27]= `arreglo[[17, 43]]`

Out[27]= `0`

In[28]= `arreglo1 = arreglo;`
`animacion2 = {arreglo};`

`ninf = Sum[Count[arreglo[[1]], 2], {1, 1, long}];`
[s... [conteo]

In[31]= `ninf`

Out[31]= `10`

In[32]=

In[33]= `inf1 = inf;`

In[34]=

`histo1 = {};`
`histo2 = {};`
`histo3 = {};`
`Rec = {};`

```
In[38]:= animacion2 = {};
```

```
In[39]:= arreglo1 = arreglo;
```

```
borrar = {};
```

```
suma = 0;
```

```
For[i = 0, i < 100, i++,
```

```
para cada
```

```
For[j = 1, j < (ninf + 1), j++,
```

```
para cada
```

```
al = RandomReal[];
```

```
real aleatorio
```

```
pos = inf[[j]];
```

```
al = RandomReal[];
```

```
real aleatorio
```

```
ii = pos[[1]];
```

```
jj = pos[[2]];
```

```
kk = pos[[3]] + 1;
```

```
inf[[j, 3]] = kk;
```

```
Label[aleatorio3];
```

```
etiqueta
```

```
ri = RandomInteger[{1, (long)}];
```

```
entero aleatorio
```

```
rj = RandomInteger[{1, (long)}];
```

```
entero aleatorio
```

```
If[ii == ri && jj == rj || arreglo[[ri, rj]] ≠ 0, Goto[aleatorio3]];
```

```
si
```

```
ve a
```

```
If[pb ≥ al && kk ≤ vidamedia, arreglo1[[ii, jj]] = arreglo[[ri, rj]];
```

```
si
```

```
arreglo1[[ri, rj]] = arreglo[[ii, jj]];
```

```
inf[[j, 1]] = ri; inf[[j, 2]] = rj];
```

```
If[kk ≥ vidamedia, AppendTo[borrar, j]]];
```

```
si
```

```
añade al final
```

```
];
```

```
lborrar = Length[borrar];
```

```
longitud
```

```
For[b = 0, b < (lborrar + 1), b++,
```

```
para cada
```

```
indice = borrar[[b]];
```

```
ii = inf[[indice, 1]];
```

```
jj = inf[[indice, 2]];
```

```
AppendTo[Rec, {ii, jj}];
```

```
añade al final
```

```
arreglo1[[ii, jj]] = 3;
```

```

inf = Delete[inf[[]], indice];
      |elimina
];
borrar = {};
arreglo = arreglo1;
AppendTo[animacion2, arreglo];
      |añade al final

For[l = 1, (l < long + 1), l++, For[m = 1, (m < long + 1), m++,
      |para cada
      |para cada
suma = 0;
If[(l - 1) < 1, l1 = long, l1 = (l - 1)];
      |si
If[(l + 1) > long, l2 = 1, l2 = (l + 1)];
      |si
If[(m - 1) < 1, m1 = long, m1 = (m - 1)];
      |si
If[(m + 1) > long, m2 = 1, m2 = (m + 1)];
      |si

If[arreglo[[l1, m1]] == 2, suma = suma + 1];
      |si
If[arreglo[[l1, m2]] == 2, suma = suma + 1];
      |si
If[arreglo[[l2, m1]] == 2, suma = suma + 1];
      |si
If[arreglo[[l2, m2]] == 2, suma = suma + 1];
      |si
If[arreglo[[l1, m]] == 2, suma = suma + 1];
      |si
If[arreglo[[l2, m]] == 2, suma = suma + 1];
      |si
If[arreglo[[l, m1]] == 2, suma = suma + 1];
      |si
If[arreglo[[l, m2]] == 2, suma = suma + 1];
      |si

prob = suma * infect;

r = RandomReal[];
      |real aleatorio
If[prob > r && arreglo[[l, m]] == 1, arreglo1[[l, m]] = 2;
      |si
AppendTo[inf, {l, m, 1}]; ];
      |añade al final

```

[añade al final](#)

];];

arreglo = arreglo1;

ninf = Sum[Count[arreglo[[n]], 2], {n, 1, long}];
[s...](#) [conteo](#)

nsus = Sum[Count[arreglo[[n]], 1], {n, 1, long}];
[s...](#) [conteo](#)

nrec = Sum[Count[arreglo[[n]], 3], {n, 1, long}];
[s...](#) [conteo](#)

AppendTo[histo1, {i, nsus}];

[añade al final](#)

AppendTo[histo2, {i, ninf}];

[añade al final](#)

AppendTo[histo3, {i, nrec}];

[añade al final](#)

AppendTo[animacion2, arreglo];

[añade al final](#)

];

- ... **Part:** Part 2 of {{30, 16, 14}} does not exist.
- ... **Part:** Part 2 of {{30, 16, 14}} does not exist.
- ... **Set:** The expression {{30, 16, 14}}[[2, 1]] cannot be used as a part specification.
- ... **Delete:** Part {2} of {{30, 16, 14}} does not exist.
- ... **Part:** Part 2 of {{30, 16, 14}} does not exist.
- ... **General:** Further output of Part::partw will be suppressed during this calculation.
- ... **Set:** Part 3 of {{30, 16, 14}} does not exist.
- ... **Set:** Part 3 of {{30, 16, 14}} does not exist.
- ... **Set:** Part 3 of {{30, 16, 14}} does not exist.
- ... **General:** Further output of Set::partw will be suppressed during this calculation.

In[54]= **inf**

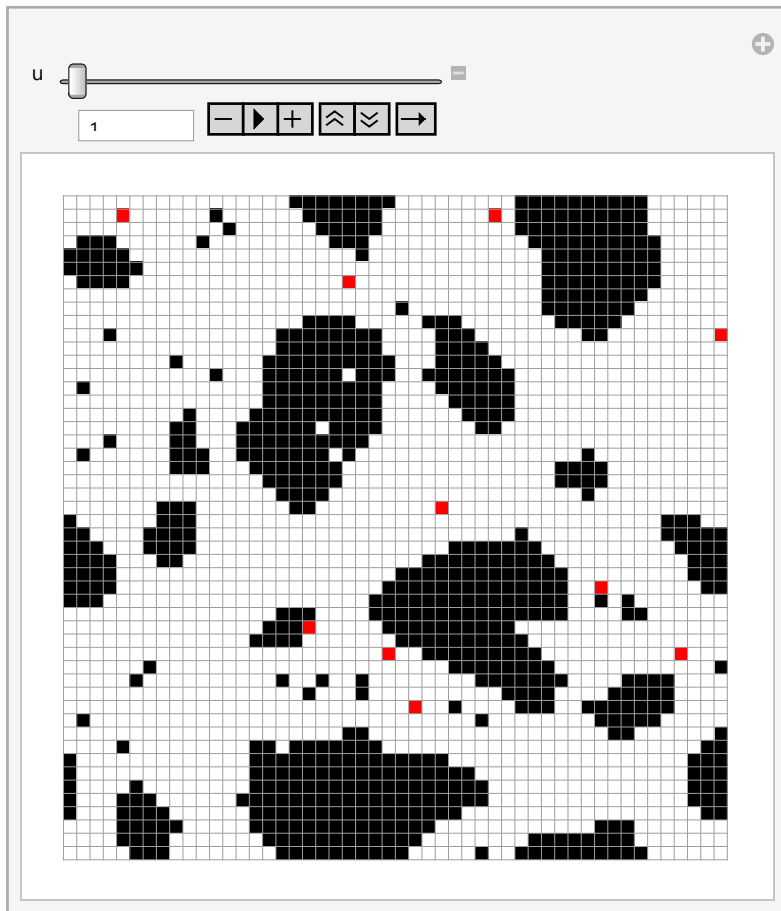
```

h1 = ListPlot[histo1, PlotStyle → Black];
  [representación de lista [estilo de repr... [negro
h2 = ListPlot[histo2, PlotStyle → Red];
  [representación de lista [estilo de repr... [rojo
h3 = ListPlot[histo3, PlotStyle → Green];
  [representación de lista [estilo de repr... [verde
Manipulate[ArrayPlot[animacion2[ [u] ], Mesh → True,
  [manipula [representación de arreglo [malla [verdadero
  ColorRules → {2 → Red, 1 → Black, 0 → White, 3 → Green}, {u, 1, 200, 1}]
  [reglas de color [rojo [negro [blanco [verde

```

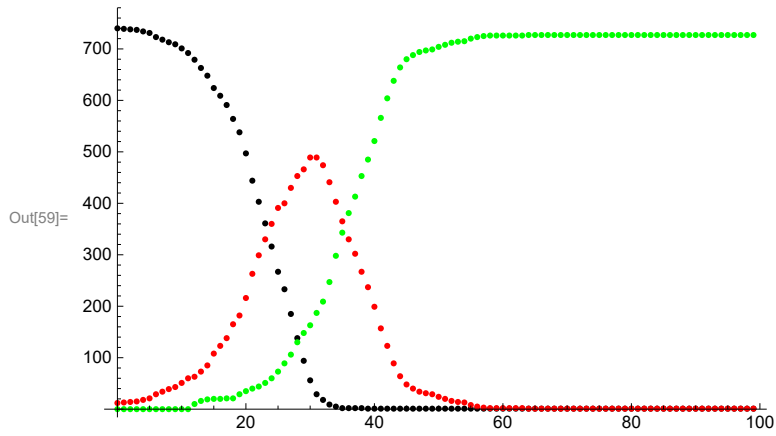
Out[54]= **Delete[{{30, 16, 14}}, 2]**

Out[58]=



In[59]:= Show[h1, h2, h3]

[muestra



```
In[140]= long = 100;
```

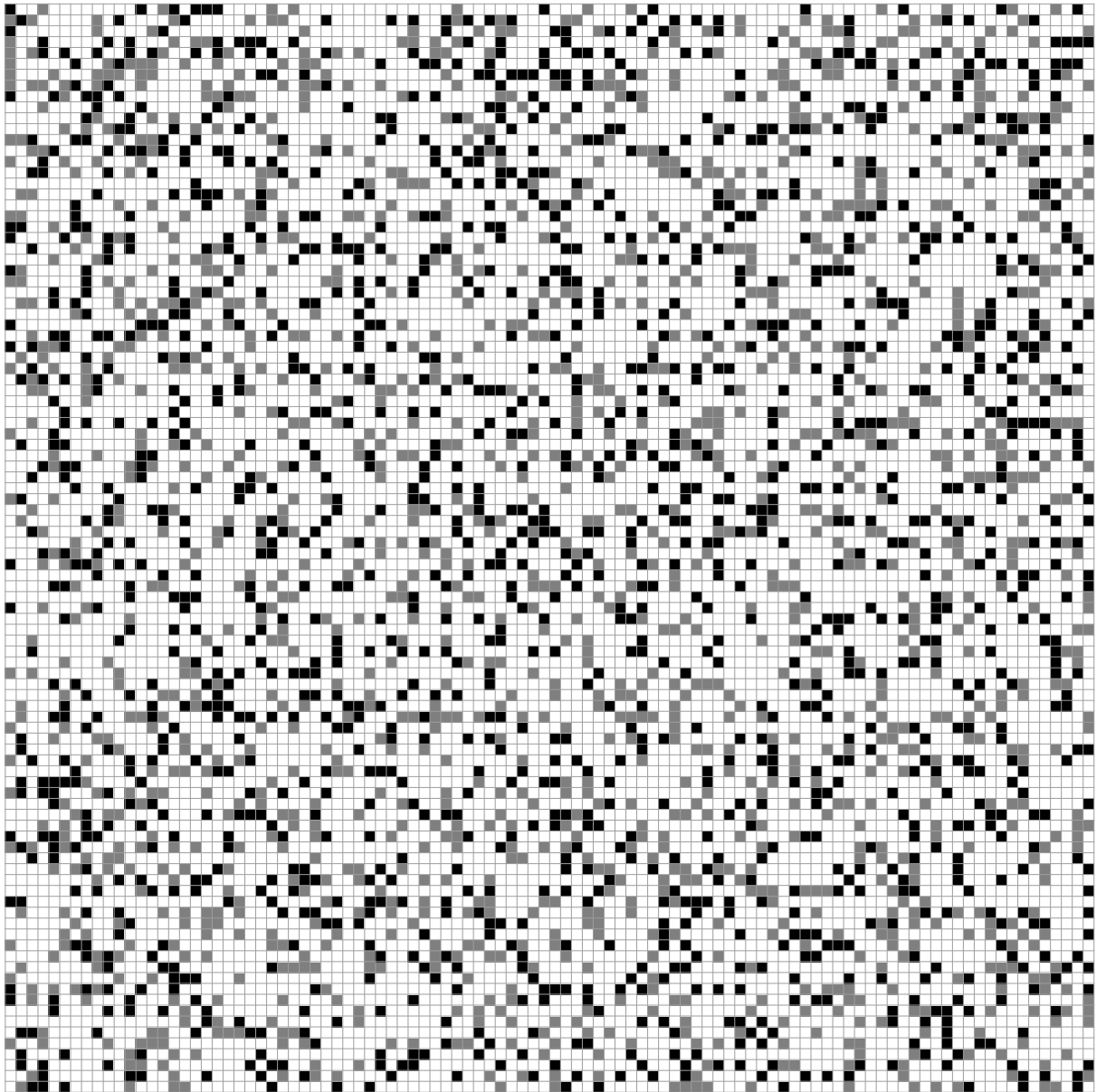
```
tolerancia = 5 / 8;  
supertotal = 1000000;  
total = 1000;  
pasos = 2000;  
tiempo = 200;  
tiempos = 2 * tiempo;  
vidamedia = 18; (*Vida media del virus*)  
infect = 0.00; (*Probabilidad de contagio*)  
pbi = 1; (*Movilidad infectados*)  
pb = 0.0  
  
infect1 = 0.05; (*Probabilidad de contagio en comunidad1*)  
pb1 = 0.05; (*Movilidad comunidad1*)  
  
infect2 = 0.1; (*Probabilidad de contagio en comunidad2*)  
pb2 = 0.00; (*Movilidad comunidad 2*)  
  
io = 50; (*INFECTADOS INICIALES*)  
vida = Table[RandomChoice[{0.70, 0.15, 0.15} -> {0, 1, 4}], {p, long}, {q, long}];  
      | tabla |elección aleatoria
```

```
Out[148]= 0.
```

```
In[151]= ArrayPlot[vida, Mesh -> True, ColorRules -> {1 -> Black, 0 -> White, 2 -> Red, 4 -> Gray}]
```

representación de a... malla verd... reglas de color negro blanco rojo gris

Out[151]=



```
In[152]= arreglo = vida;
```

```
In[153]= suma1 = 0;
```

```
In[154]= check = 0;
```

```
In[155]= suma2 = 0;
```

```
In[156]= pasos = 0;  
animacion = {arreglo};  
fallo = 0;
```



```
In[159]:= While[check < supertotal,  
  |mientras
```

```

Label[aleatorio];
|etiqueta
suma1 = 0; suma2 = 0;
ri1 = RandomInteger[{1, (long)}];
|entero aleatorio
rj1 = RandomInteger[{1, (long)}];
|entero aleatorio
ri2 = RandomInteger[{1, (long)}];
|entero aleatorio
rj2 = RandomInteger[{1, (long)}]; pos1 = arreglo[[ri1, rj1]];
|entero aleatorio

pos2 = arreglo[[ri2, rj2]];
If[pos1 == pos2, Goto[aleatorio], check = check + 1];
|si |ve a
If[(ri1 - 1) < 1, ri11 = long, ri11 = (ri1 - 1)];
|si
If[(ri2 - 1) < 1, ri21 = long, ri21 = (ri2 - 1)];
|si
If[(ri1 + 1) > long, ri12 = 1, ri12 = (ri1 + 1)];
|si
If[(ri2 + 1) > long, ri22 = 1, ri22 = (ri2 + 1)];
|si
If[(rj1 - 1) < 1, rj11 = long, rj11 = (rj1 - 1)];
|si
If[(rj2 - 1) < 1, rj21 = long, rj21 = (rj2 - 1)];
|si
If[(rj1 + 1) > long, rj12 = 1, rj12 = (rj1 + 1)];
|si
If[(rj2 + 1) > long, rj22 = 1, rj22 = (rj2 + 1)];
|si
If[arreglo[[ri11, rj11]] == pos1, suma1 = suma1 + 1];
|si
If[arreglo[[ri11, rj12]] == pos1, suma1 = suma1 + 1];
|si
If[arreglo[[ri12, rj11]] == pos1, suma1 = suma1 + 1]; If[arreglo[[ri12, rj12]] == pos1,
|si |si
suma1 = suma1 + 1]; If[arreglo[[ri11, rj1]] == pos1, suma1 = suma1 + 1];
|si
If[arreglo[[ri12, rj1]] == pos1, suma1 = suma1 + 1]; If[arreglo[[ri1, rj11]] == pos1,
|si |si
suma1 = suma1 + 1]; If[arreglo[[ri1, rj12]] == pos1, suma1 = suma1 + 1];
|si

If[arreglo[[ri21, rj21]] == pos2, suma2 = suma2 + 1];
|si
If[arreglo[[ri21, rj22]] == pos2, suma2 = suma2 + 1];
|si

```

```

↳
If[arreglo[[ri22, rj21]] == pos2, suma2 = suma2 + 1]; If[arreglo[[ri22, rj22]] == pos2,
↳
  suma2 = suma2 + 1]; If[arreglo[[ri21, rj2]] == pos2, suma2 = suma2 + 1];
↳
If[arreglo[[ri22, rj2]] == pos2, suma2 = suma2 + 1]; If[arreglo[[ri2, rj21]] == pos2,
↳
  suma2 = suma2 + 1]; If[arreglo[[ri2, rj22]] == pos2, suma2 = suma2 + 1];
↳

felicidad1 = suma1/8;
felicidad2 = suma2/8;

If[felicidad1 < tolerancia && felicidad2 < tolerancia, fallo = 0;
↳
  pasos = pasos + 1;
  arreglo[[ri1, rj1]] = pos2; arreglo[[ri2, rj2]] = pos1;
  AppendTo[animacion, arreglo], fallo = fallo + 1];
↳añade al final
If[mudanza > pasos, check = supertotal];
↳
If[fallo > total, check = supertotal];]
↳

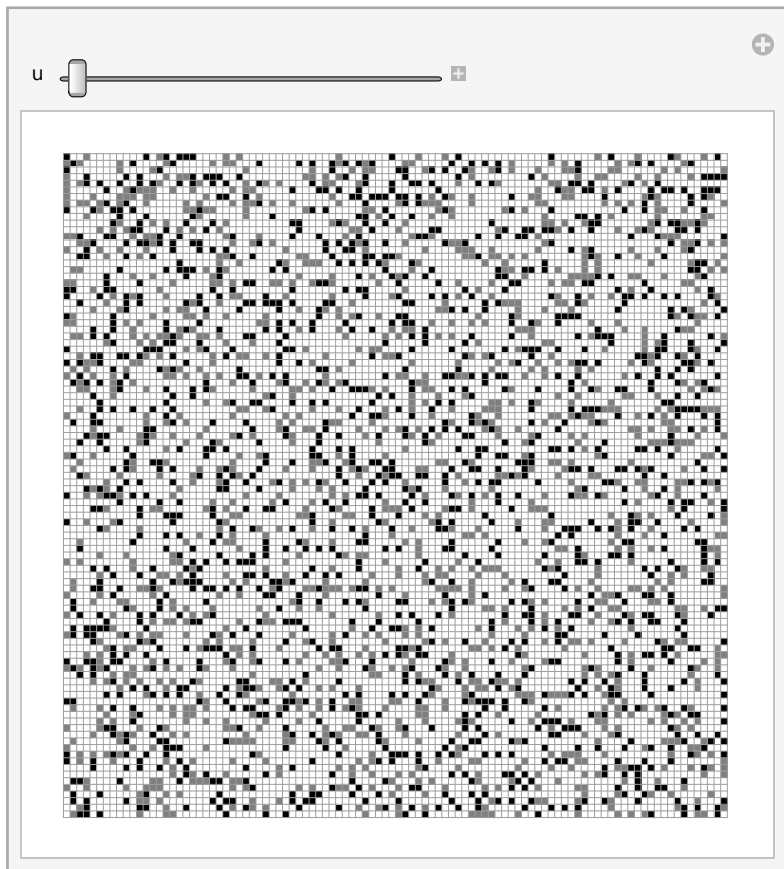
```

```

In[160]:= video = Manipulate[ArrayPlot[animacion[[u]], Mesh → True,
↳manipula ↳representación de arreglo ↳malla ↳verdadero
  ColorRules → {2 → Red, 1 → Black, 0 → White, 4 → Gray}], {u, 1, pasos, 1}]
↳reglas de color ↳rojo ↳negro ↳blanco ↳gris

```

Out[160]=



```

In[161]:= paso = 0;
inf = {};
For[i = 0, i < io, i++,
  para cada
    suma1 = 0; suma2 = 0;
    Label[aleatorio2];
    etiqueta
    suma1 = 0;
    ri1 = RandomInteger[{1, (long)}]; rj1 = RandomInteger[{1, (long)}];
    entero aleatorio      entero aleatorio
    pos1 = arreglo[[ri1, rj1]];

    If[(ri1 - 1) < 1, ri11 = long, ri11 = (ri1 - 1)];
    si
    If[(ri1 + 1) > long, ri12 = 1, ri12 = (ri1 + 1)];
    si
    If[(rj1 - 1) < 1, rj11 = long, rj11 = (rj1 - 1)];
    si
    If[(rj1 + 1) > long, rj12 = 1, rj12 = (rj1 + 1)];
    si

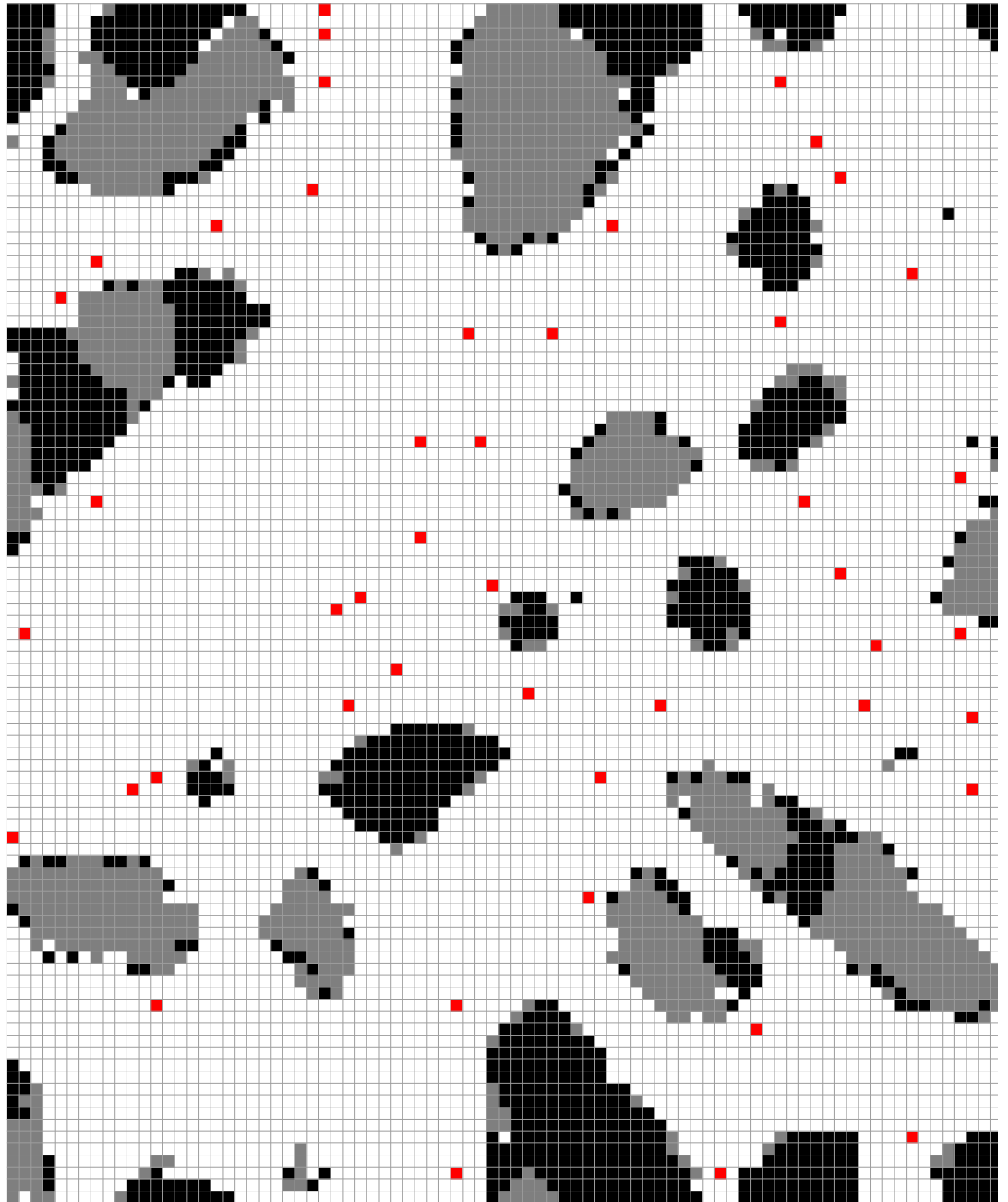
    If[arreglo[[ri11, rj11]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri11, rj12]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri12, rj11]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri12, rj12]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri11, rj1]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri12, rj1]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri1, rj11]] ≠ 0, suma1 = suma1 + 1];
    si
    If[arreglo[[ri1, rj12]] ≠ 0, suma1 = suma1 + 1];
    si
    If[suma1 ≠ 0, Goto[aleatorio2],
      si      ve a
    arreglo[[ri1, rj1]] = 2; AppendTo[inf, {ri1, rj1, 1}];];
    añade al final
    paso = paso + 1;

];

ArrayPlot[arreglo, Mesh → True, ColorRules → {2 → Red, 1 → Black, 0 → White, 4 → Gray}]
representación de arreglo malla verd... reglas de color rojo negro blanco gris

```

Out[164]=



In[165]:=

```

su = {};
su1 = {};
su2 = {};

For[1 = 1, (1 < long + 1), 1++, For[m = 1, (m < long + 1), m++,
  para cada
    Switch[arreglo[[1, m]], 1, AppendTo[su, {1, m}];, 4, AppendTo[su, {1, m}];];
  para cada
    Switch[arreglo[[1, m]], 1, AppendTo[su, {1, m}];, 4, AppendTo[su, {1, m}];];
  evalúa dependiendo de
  añade al final
  añade al final
];];

```

In[169]:=

In[170]:= **pos = inf[[3]]**

Out[170]= {89, 87, 1}

In[171]:= **pos[[1]]**

Out[171]= 89

In[172]:= **arreglo[[17, 43]]**

Out[172]= 4

```

In[173]:= arreglo1 = arreglo;
animacion2 = {arreglo};

```

```

ninf = Sum[Count[arreglo[[1]], 2], {1, 1, long}];

```

```

s... conteo

```

```

nsus1 = Sum[Count[arreglo[[n]], 1], {n, 1, long}];

```

```

s... conteo

```

```

nsus2 = Sum[Count[arreglo[[n]], 4], {n, 1, long}];

```

```

s... conteo

```

```

nrec = Sum[Count[arreglo[[n]], 3], {n, 1, long}];

```

```

s... conteo

```

```

nsus = nsus1 + nsus2;

```

In[180]:= **nsus**

Out[180]= 2969

In[181]:= **ninf**

Out[181]= 50

In[182]:=

In[183]:=

In[184]:=

```

histo1 = {};
histo2 = {};
histo3 = {};
histo4 = {};
histo5 = {};
Rec = {};

```

In[190]:= **animacion2** = {arreglo};

```

In[191]:= arreglo1 = arreglo;
borrar = {};

```

```

In[193]:= suma = 0;
suma1 = 0;

```

```

For[i = 0, i < tiempo, i++,
  para cada

```

```

  For[j = 1, j < (ninf + 1), j++,
    para cada

```

```

      pos = inf[[j]];
      al = RandomReal[];
           real aleatorio
      ii = pos[[1]];
      jj = pos[[2]];
      kk = pos[[3]] + 1;
      inf[[j, 3]] = kk;

```

```

      Label[aleatorio3];
           etiqueta

```

```

      ri = RandomInteger[{1, (long)}];
           entero aleatorio
      rj = RandomInteger[{1, (long)}];
           entero aleatorio

```

```

      If[ii == ri && jj == rj || arreglo[[ri, rj]] ≠ 0, Goto[aleatorio3];];
           si ve a

```

```

      If[ri == 1, ri1 = long, ri1 = (ri - 1)];
           si

```

```

      If[ri == long, ri2 = 1, ri2 = (ri + 1)];
           si

```

```

      If[rj == 1, rj1 = long, rj1 = (rj - 1)];
           si

```

```

    ]];
    If[rj == long, rj2 = 1, rj2 = (rj + 1)];
    ]];

    If[arreglo[[ri1, rj1]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri1, rj2]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri2, rj1]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri2, rj2]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri1, rj]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri2, rj]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri, rj1]] != 0, suma1 = suma1 + 1];
    ]];
    If[arreglo[[ri, rj2]] != 0, suma1 = suma1 + 1];
    ]];

    (*If[suma1<3 || suma1>6, suma1=0; Goto[aleatorio3];];*)
    ]];

    suma1 = 0;

    If[pbi ≥ al && kk ≤ vidamedia, arreglo1[[ii, jj]] = 0;
    ]];
    arreglo1[[ri, rj]] = 2;
    inf[[j, 1]] = ri; inf[[j, 2]] = rj];];

    If[kk > vidamedia, arreglo1[[ii, jj]] = 3;
    ]];
    AppendTo[Rec, {ii, jj}];
    ]];
    inf[[j]] = {};];

    arreglo = arreglo1;
  ];

  inf = inf /. {} → Nothing;
  ]];
  AppendTo[animacion2, arreglo];
  ]];

```



```

For[k = 1, k < (nsus + 1), k++,
  para cada
    pos = su[k];
    al = RandomReal[];
      real aleatorio
    ii = pos[[1]];
    jj = pos[[2]];

    If[arreglo[[ii, jj]] == 1, pb = pb1; infect = infect1];
      si
    If[arreglo[[ii, jj]] == 4, pb = pb2; infect = infect2];
      si

    Label[aleatorio4];
      etiqueta
    ri = RandomInteger[{1, (long)}];
      entero aleatorio
    rj = RandomInteger[{1, (long)}];
      entero aleatorio
    If[ii == ri && jj == rj || arreglo[[ri, rj]] != 0, Goto[aleatorio4]];
      si ve a

    If[ri == 1, ri1 = long, ri1 = (ri - 1)];
      si
    If[ri == long, ri2 = 1, ri2 = (ri + 1)];
      si
    If[rj == 1, rj1 = long, rj1 = (rj - 1)];
      si
    If[rj == long, rj2 = 1, rj2 = (rj + 1)];
      si
    suma = 0;
    If[arreglo[[ri1, rj1]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri1, rj2]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri2, rj1]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri2, rj2]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri1, rj]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri2, rj]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri, rj1]] != 0, suma = suma + 1];
      si
    If[arreglo[[ri, rj2]] != 0, suma = suma + 1];
      si

```

```

If[suma < 3 || suma > 6, suma = 0; Goto[aleatorio4];];
|si |ve a

If[pb > a1 , arreglo1[[ii, jj]] = arreglo[[ri, rj]];
|si
  arreglo1[[ri, rj]] = arreglo[[ii, jj]];
  su[[k, 1]] = ri; su[[k, 2]] = rj;];

pos = su[[k]];

l = pos[[1]];
m = pos[[2]];

If[(1 - 1) < 1, l1 = long, l1 = (1 - 1)];
|si
  If[(1 + 1) > long, l2 = 1, l2 = (1 + 1)];
|si
  If[(m - 1) < 1, m1 = long, m1 = (m - 1)];
|si
  If[(m + 1) > long, m2 = 1, m2 = (m + 1)];
|si
suma1 = 0;

If[arreglo[[l1, m1]] == 2, suma1 = suma1 + 1];
|si
If[arreglo[[l1, m2]] == 2, suma1 = suma1 + 1];
|si
If[arreglo[[l2, m1]] == 2, suma1 = suma1 + 1];
|si
If[arreglo[[l2, m2]] == 2, suma1 = suma1 + 1];
|si
  If[arreglo[[l1, m]] == 2, suma1 = suma1 + 1];
|si
  If[arreglo[[l2, m]] == 2, suma1 = suma1 + 1];
|si
  If[arreglo[[1, m1]] == 2, suma1 = suma1 + 1];
|si
  If[arreglo[[1, m2]] == 2, suma1 = suma1 + 1];
|si

prob = suma1 * infect;

r = RandomReal[];
|real aleatorio
If[prob > r && arreglo[[1, m]] != 3 && arreglo[[1, m]] != 0, arreglo1[[1, m]] = 2;
|ei

```

```

↳
AppendTo[inf, {1, m, 1}];
  [añade al final
su[[k]] = {}; ];

];

arreglo = arreglo1;
AppendTo[animacion2, arreglo];
  [añade al final

su = su /. {} → Nothing;
  [nada

arreglo = arreglo1;

ninf = Sum[Count[arreglo[[n]], 2], {n, 1, long}];
  [s... [conteo
nsus1 = Sum[Count[arreglo[[n]], 1], {n, 1, long}];
  [s... [conteo
nsus2 = Sum[Count[arreglo[[n]], 4], {n, 1, long}];
  [s... [conteo
nrec = Sum[Count[arreglo[[n]], 3], {n, 1, long}];
  [s... [conteo
nsus = nsus1 + nsus2;

AppendTo[histo1, {i, nsus}];
  [añade al final
AppendTo[histo2, {i, ninf}];
  [añade al final
AppendTo[histo3, {i, nrec}];
  [añade al final
AppendTo[histo4, {i, nsus1}];
  [añade al final
AppendTo[histo5, {i, nsus2}];
  [añade al final

AppendTo[animacion2, arreglo];
  [añade al final

];

```

... **Part:** Part 58 of

{ {}, {}, {7, 12, 18}, {86, 18, 18}, {13, 34, 18}, {5, 51, 17}, {36, 70, 17}, {46, 78, 17}, {50, 13, 17}, {2, 33, 17}, {9, 23, 17}, {33, 77, 17}, {95, 5, 17}, {62, 34, 17}, {12, 68, 16}, {12, 37, 16}, {29, 93, 16}, {10, 78, 16}, {72, 54, 15}, {69, 15, 15}, {5, 35, 14}, {71, 31, 14}, {66, 100, 14}, {92, 11, 14}, {70, 86, 13}, {3, 69, 13}, {1, 92, 12}, {32, 48, 12}, {11, 3, 12}, {46, 8, 10}, {21, 2, 9}, {63, 13, 8}, {77, 20, 8}, {100, 22, 8}, {91, 4, 7}, {41, 64, 7}, {87, 98, 7}, {39, 17, 7}, {24, 27, 6}, {8, 89, 6}, {54, 91, 5}, {72, 99, 5}, {36, 24, 4}, {66, 52, 4}, {77, 33, 4}, {20, 35, 3}, {11, 1, 3}, {72, 80, 3}, {26, 4, 3}, {75, 78, 3}, <<7>> does not exist.

... **Part:** Part 3 of

{ {}, {}, {7, 12, 18}, {86, 18, 18}, {13, 34, 18}, {5, 51, 17}, {36, 70, 17}, {46, 78, 17}, {50, 13, 17}, {2, 33, 17}, {9, 23, 17}, {33, 77, 17}, {95, 5, 17}, {62, 34, 17}, {12, 68, 16}, {12, 37, 16}, {29, 93, 16}, {10, 78, 16}, {72, 54, 15}, {69, 15, 15}, {5, 35, 14}, {71, 31, 14}, {66, 100, 14}, {92, 11, 14}, {70, 86, 13}, {3, 69, 13}, {1, 92, 12}, {32, 48, 12}, {11, 3, 12}, {46, 8, 10}, {21, 2, 9}, {63, 13, 8}, {77, 20, 8}, {100, 22, 8}, {91, 4, 7}, {41, 64, 7}, {87, 98, 7}, {39, 17, 7}, {24, 27, 6}, {8, 89, 6}, {54, 91, 5}, {72, 99, 5}, {36, 24, 4}, {66, 52, 4}, {77, 33, 4}, {20, 35, 3}, {11, 1, 3}, {72, 80, 3}, {26, 4, 3}, {75, 78, 3}, <<7>>[[58]] does not exist.

... **Set:** Part 58 of

{ {}, {}, {7, 12, 18}, {86, 18, 18}, {13, 34, 18}, {5, 51, 17}, {36, 70, 17}, {46, 78, 17}, {50, 13, 17}, {2, 33, 17}, {9, 23, 17}, {33, 77, 17}, {95, 5, 17}, {62, 34, 17}, {12, 68, 16}, {12, 37, 16}, {29, 93, 16}, {10, 78, 16}, {72, 54, 15}, {69, 15, 15}, {5, 35, 14}, {71, 31, 14}, {66, 100, 14}, {92, 11, 14}, {70, 86, 13}, {3, 69, 13}, {1, 92, 12}, {32, 48, 12}, {11, 3, 12}, {46, 8, 10}, {21, 2, 9}, {63, 13, 8}, {77, 20, 8}, {100, 22, 8}, {91, 4, 7}, {41, 64, 7}, {87, 98, 7}, {39, 17, 7}, {24, 27, 6}, {8, 89, 6}, {54, 91, 5}, {72, 99, 5}, {36, 24, 4}, {66, 52, 4}, {77, 33, 4}, {20, 35, 3}, {11, 1, 3}, {72, 80, 3}, {26, 4, 3}, {75, 78, 3}, <<7>> does not exist.

... **Part:** Part 67 of

{ {}, {}, {}, {82, 19, 18}, {96, 44, 18}, {47, 66, 18}, {82, 33, 18}, {49, 75, 18}, {47, 47, 18}, {27, 87, 18}, {36, 44, 18}, {54, 20, 18}, {22, 27, 17}, {28, 98, 17}, {6, 78, 17}, {7, 39, 17}, {19, 83, 16}, {36, 55, 16}, {51, 57, 15}, {63, 72, 15}, {97, 82, 15}, {53, 29, 15}, {70, 99, 14}, {93, 21, 14}, {31, 79, 13}, {7, 83, 13}, {60, 43, 13}, {60, 34, 11}, {93, 45, 10}, {56, 42, 9}, {21, 80, 9}, {1, 22, 9}, {53, 99, 8}, {51, 69, 8}, {47, 35, 8}, {93, 6, 8}, {79, 47, 7}, {56, 59, 7}, {93, 16, 6}, {24, 24, 6}, {64, 56, 5}, {69, 46, 5}, {84, 22, 5}, {100, 31, 4}, {67, 26, 4}, {2, 78, 4}, {57, 84, 4}, {99, 29, 4}, {61, 70, 4}, {53, 54, 4}, <<16>> does not exist.

... **General:** Further output of Part::partw will be suppressed during this calculation.

... **Set:** Part 67 of

{ {}, {}, {}, {82, 19, 18}, {96, 44, 18}, {47, 66, 18}, {82, 33, 18}, {49, 75, 18}, {47, 47, 18}, {27, 87, 18}, {36, 44, 18}, {54, 20, 18}, {22, 27, 17}, {28, 98, 17}, {6, 78, 17}, {7, 39, 17}, {19, 83, 16}, {36, 55, 16}, {51, 57, 15}, {63, 72, 15}, {97, 82, 15}, {53, 29, 15}, {70, 99, 14}, {93, 21, 14}, {31, 79, 13}, {7, 83, 13}, {60, 43, 13}, {60, 34, 11}, {93, 45, 10}, {56, 42, 9}, {21, 80, 9}, {1, 22, 9}, {53, 99, 8}, {51, 69, 8}, {47, 35, 8}, {93, 6, 8}, {79, 47, 7}, {56, 59, 7}, {93, 16, 6}, {24, 24, 6}, {64, 56, 5}, {69, 46, 5}, {84, 22, 5}, {100, 31, 4}, {67, 26, 4}, {2, 78, 4}, {57, 84, 4}, {99, 29, 4}, {61, 70, 4}, {53, 54, 4}, <<16>> does not exist.

... **Set:** Part 68 of

{ {}, {}, {}, {}, {}, {}, {}, {56, 16, 18}, {27, 76, 18}, {16, 97, 18}, {44, 76, 18}, {27, 92, 17}, {69, 27, 17}, {90, 69, 16}, {24, 79, 16}, {87, 43, 16}, {30, 40, 16}, {36, 82, 15}, {13, 100, 15}, {19, 58, 14}, {46, 63, 14}, {91, 57, 14}, {69, 4, 12}, {58, 73, 11}, {71, 84, 10}, {26, 2, 10}, {13, 99, 10}, {54, 95, 9}, {96, 19, 9}, {50, 41, 9}, {11, 31, 9}, {72, 35, 8}, {34, 73, 8}, {57, 96, 7}, {77, 90, 7}, {47, 76, 6}, {5, 81, 6}, {28, 65, 6}, {29, 84, 5}, {30, 66, 5}, {49, 4, 5}, {54, 53, 5}, {70, 17, 5}, {63, 40, 5}, {32, 29, 5}, {66, 42, 4}, {28, 79, 4}, {3, 27, 4}, <<17>> does not exist.

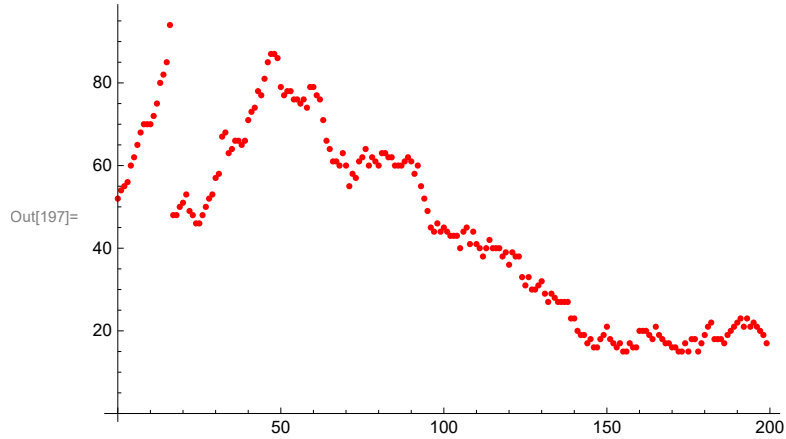
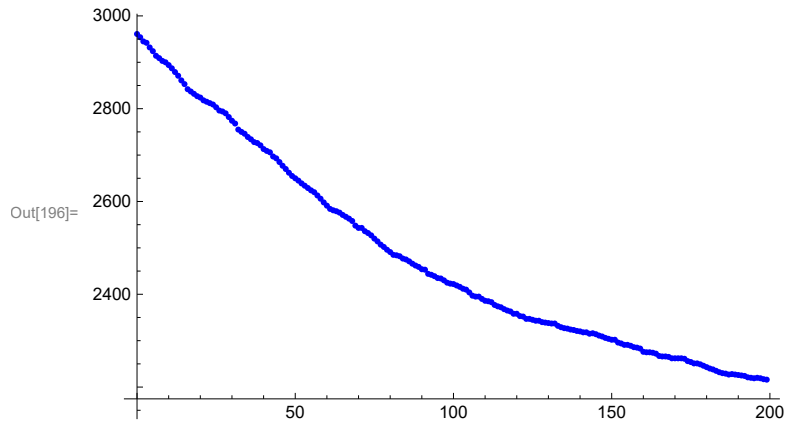
... **General:** Further output of Set::partw will be suppressed during this calculation.

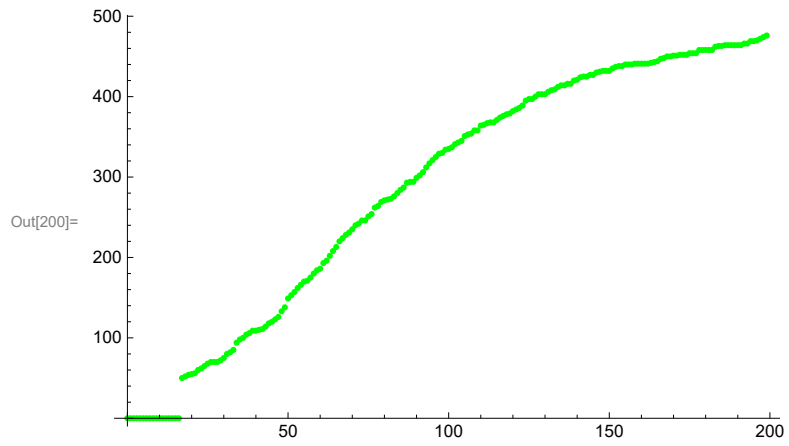
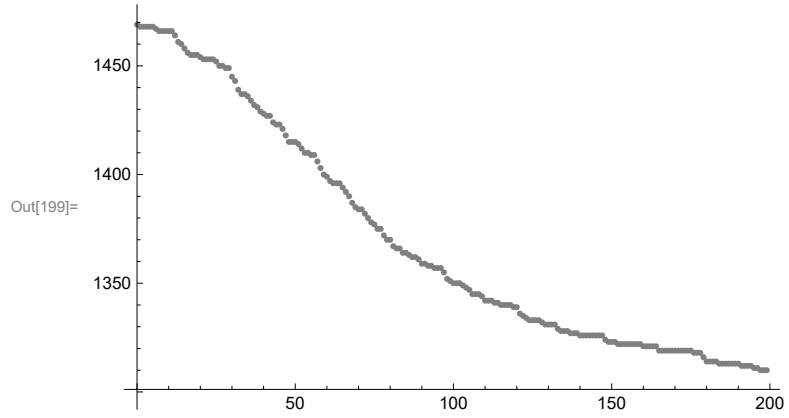
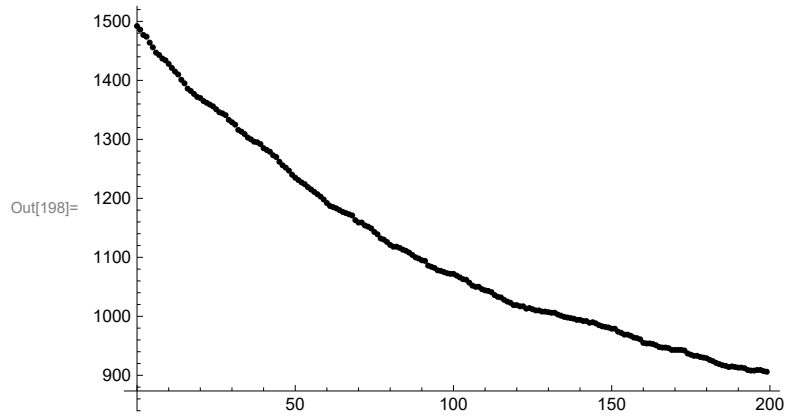
In[196]:=

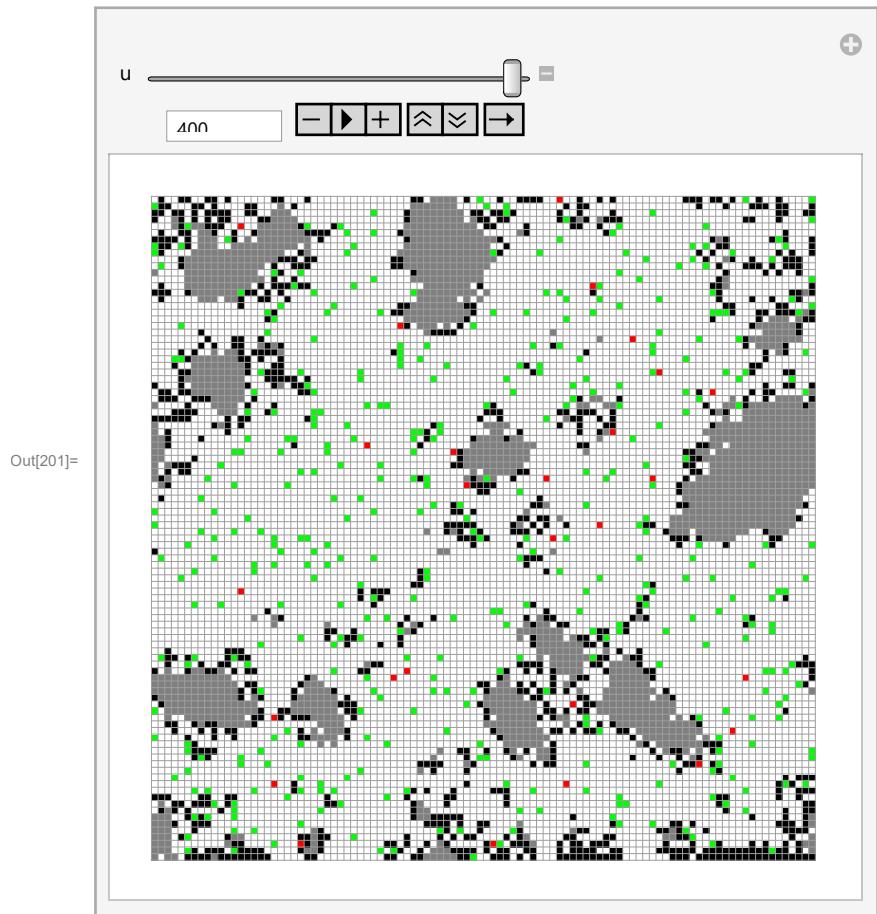
```

h1 = ListPlot[histo1, PlotStyle → Blue]
  [representación de lista [estilo de repr... [azul
h2 = ListPlot[histo2, PlotStyle → Red]
  [representación de lista [estilo de repr... [rojo
h4 = ListPlot[histo4, PlotStyle → Black]
  [representación de lista [estilo de repr... [negro
h5 = ListPlot[histo5, PlotStyle → Gray]
  [representación de lista [estilo de repr... [gris
h3 = ListPlot[histo3, PlotStyle → Green]
  [representación de lista [estilo de repr... [verde
videos = Manipulate[ArrayPlot[animacion2[[u]], Mesh → True,
  [manipula [representación de arreglo [malla [verdadero
  ColorRules → {2 → Red, 1 → Black, 0 → White, 3 → Green, 4 → Gray}], {u, 1, tiempos, 1}]
  [reglas de color [rojo [negro [blanco [verde [gris

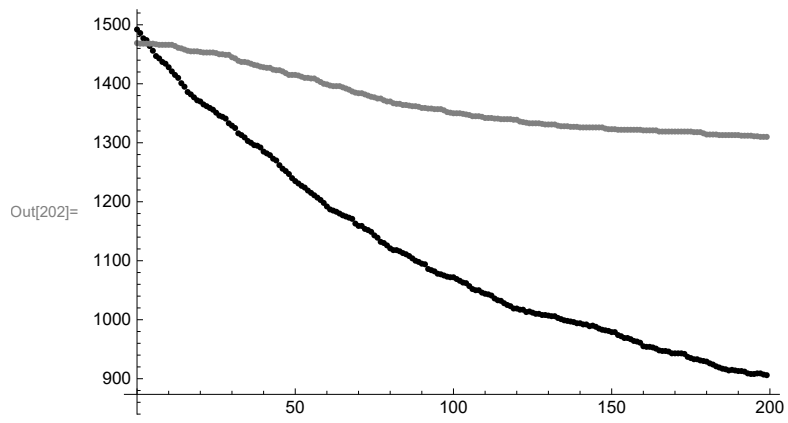
```



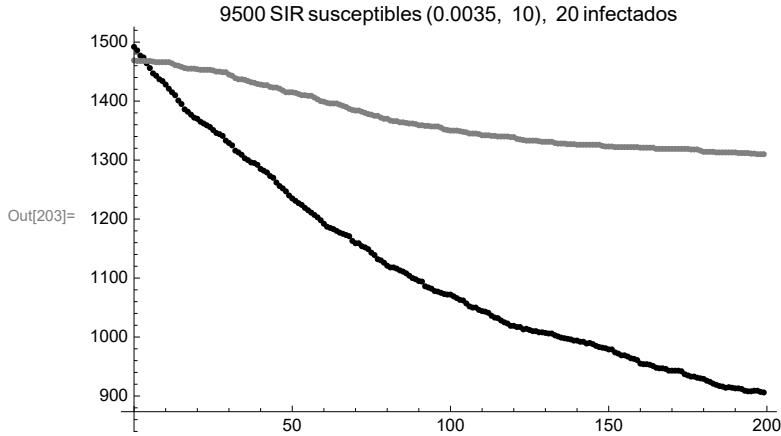




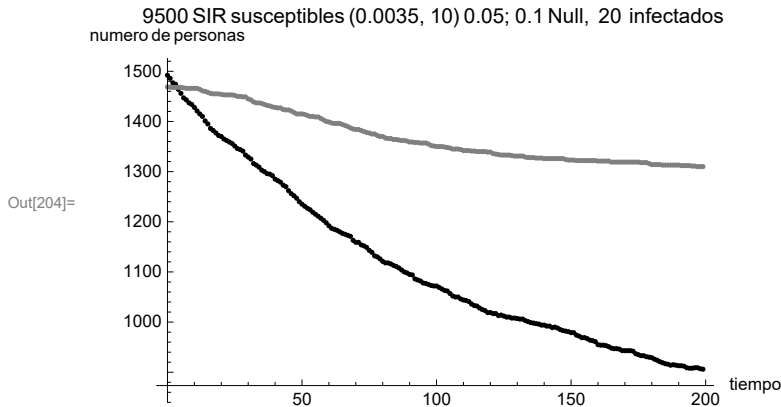
```
In[202]:= grafica = Show[h4, h2, h3, h5]
```



```
In[203]:= Show[grafica,
  muestra
  PlotLabel -> RowBoxes[RowBox[{"9500", " ", "SIR", " ", "susceptibles", RowBox[
    etiqueta de r... muestra c... caja de fila caja de fila
    {"(", RowBox[{"0.0035", " ", " ", " ", "10"}], " )"}], " ,",
    caja de fila
    " ", RowBox[{"20", "infectados"}]}]], LabelStyle -> {GrayLevel[0]}
    caja de fila
    estilo de etiqueta nivel de gris
```



```
In[204]:= graficas =
  Show[grafica, AxesLabel -> {HoldForm[tiempo], HoldForm[numero de personas]}, PlotLabel ->
    muestra
    etiqueta de ejes forma sin evaluación forma sin evaluación
    etiqueta de representa
    RowBoxes[RowBox[RowBox[{"9500SIR", " ", "susceptibles (0.0035,10) 0.05;0.1"}, {"",
      muestra c... caja de fila caja de fila
      " ", RowBox[{"20", " ", "infectados", " "}}]], LabelStyle -> {GrayLevel[0]}
      caja de fila
      estilo de etiqueta nivel de gris
```



In[205]=

In[206]=

In[207]=

In[208]=

```
In[209]:= (*Export["SIR (COMUNIDADES1).avi", videos];*)
  exporta
```



```
In[187]:= n = 500;  
  
In[188]:= mini = 10;  
          partes = 50;  
  
In[190]:= mcambio = 10;  
  
In[191]:= pasos = 2000;  
  
In[192]:= fallos = n;  
  
In[193]:= agentes = Table[mini, {i, n}];  
          |tabla  
  
In[194]:=  
  
In[195]:= S = Entropy[agentes]  
          |entropía  
  
Out[195]= 0  
  
In[196]:=  
  
In[197]:= St = {};  
  
In[198]:= conteo = 0;
```

```

In[199]:= For[j = 1, (j < pasos), j++, Label[inicio];
  para cada
    perdedor = RandomInteger[{1, (n)}];
      entero aleatorio
    ganador = RandomInteger[{1, (n)}];
      entero aleatorio
    tasa = RandomInteger[{0, mcambio}];
      entero aleatorio
    If[agentes[[perdedor]] ≥ tasa,
      si
        agentes[[perdedor]] = agentes[[perdedor]] - tasa;
        agentes[[ganador]] = agentes[[ganador]] + tasa;
        conteo = conteo + 1;
        S = Entropy[agentes];
          entropía

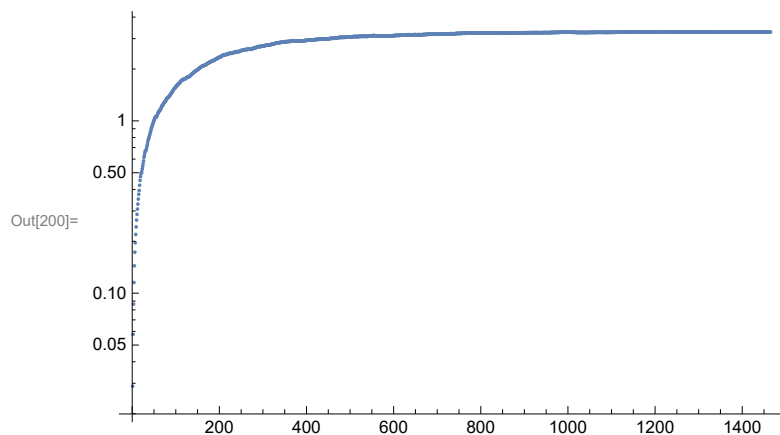
      AppendTo[St, S];
        añade al final
    ];

];

```

```
ListLogPlot[St]
```

```
representación log de lista
```



```
In[201]:=
```

```
In[202]:= histo = {}
```

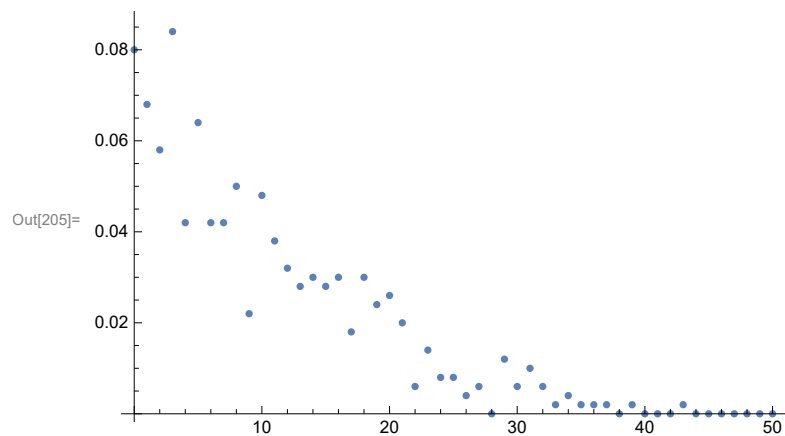
```
Out[202]= {}
```

```
In[203]:= For[i = 0, (i < 51), i++,
  para cada
    c = Count[agentes, i];
    conteo
    c = c / n;
    AppendTo[histo, {i, c}];
  añade al final
];
```

```
In[204]:= histo
```

```
Out[204]= {{0,  $\frac{2}{25}$ }, {1,  $\frac{17}{250}$ }, {2,  $\frac{29}{500}$ }, {3,  $\frac{21}{250}$ }, {4,  $\frac{21}{500}$ }, {5,  $\frac{8}{125}$ }, {6,  $\frac{21}{500}$ },
  {7,  $\frac{21}{500}$ }, {8,  $\frac{1}{20}$ }, {9,  $\frac{11}{500}$ }, {10,  $\frac{6}{125}$ }, {11,  $\frac{19}{500}$ }, {12,  $\frac{4}{125}$ }, {13,  $\frac{7}{250}$ },
  {14,  $\frac{3}{100}$ }, {15,  $\frac{7}{250}$ }, {16,  $\frac{3}{100}$ }, {17,  $\frac{9}{500}$ }, {18,  $\frac{3}{100}$ }, {19,  $\frac{3}{125}$ }, {20,  $\frac{13}{500}$ },
  {21,  $\frac{1}{50}$ }, {22,  $\frac{3}{500}$ }, {23,  $\frac{7}{500}$ }, {24,  $\frac{1}{125}$ }, {25,  $\frac{1}{125}$ }, {26,  $\frac{1}{250}$ }, {27,  $\frac{3}{500}$ },
  {28, 0}, {29,  $\frac{3}{250}$ }, {30,  $\frac{3}{500}$ }, {31,  $\frac{1}{100}$ }, {32,  $\frac{3}{500}$ }, {33,  $\frac{1}{500}$ }, {34,  $\frac{1}{250}$ },
  {35,  $\frac{1}{500}$ }, {36,  $\frac{1}{500}$ }, {37,  $\frac{1}{500}$ }, {38, 0}, {39,  $\frac{1}{500}$ }, {40, 0}, {41, 0}, {42, 0},
  {43,  $\frac{1}{500}$ }, {44, 0}, {45, 0}, {46, 0}, {47, 0}, {48, 0}, {49, 0}, {50, 0}}
```

```
In[205]:= histograma = ListPlot[histo]
representación de lista
```



In[206]=

```

suma = Round[Sum[Rationalize[E^(-k/mini)], {k, 50}]];
maximo = Max[agentes];
sdaga = {};
For[i = 0, (i < 51), i++,
sd = E^(-i/mini) / suma;
AppendTo[sdaga, {i, sd}];
];

```

In[210]= sdaga

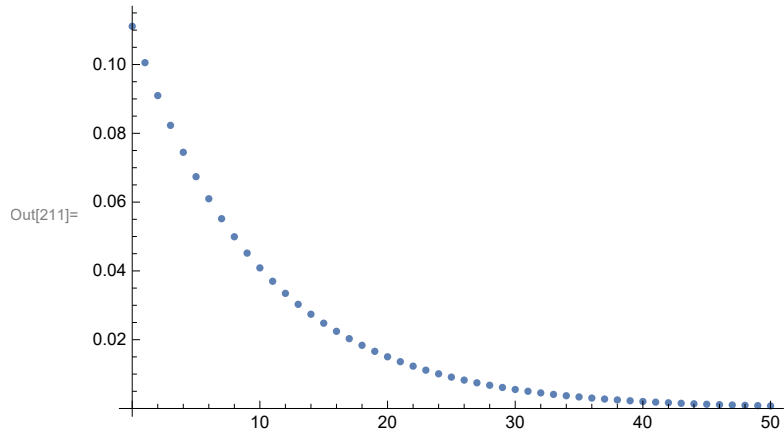
```

Out[210]= {{0, 1/9}, {1, 1/(9 e^(1/10))}, {2, 1/(9 e^(1/5))}, {3, 1/(9 e^(3/10))}, {4, 1/(9 e^(2/5))}, {5, 1/(9 e^(1/2))},
{6, 1/(9 e^(3/5))}, {7, 1/(9 e^(7/10))}, {8, 1/(9 e^(4/5))}, {9, 1/(9 e^(9/10))}, {10, 1/9 e}, {11, 1/(9 e^(11/10))},
{12, 1/(9 e^(6/5))}, {13, 1/(9 e^(13/10))}, {14, 1/(9 e^(7/5))}, {15, 1/(9 e^(3/2))}, {16, 1/(9 e^(8/5))}, {17, 1/(9 e^(17/10))},
{18, 1/(9 e^(9/5))}, {19, 1/(9 e^(19/10))}, {20, 1/(9 e^2)}, {21, 1/(9 e^(21/10))}, {22, 1/(9 e^(11/5))}, {23, 1/(9 e^(23/10))},
{24, 1/(9 e^(12/5))}, {25, 1/(9 e^(5/2))}, {26, 1/(9 e^(13/5))}, {27, 1/(9 e^(27/10))}, {28, 1/(9 e^(14/5))},
{29, 1/(9 e^(29/10))}, {30, 1/(9 e^3)}, {31, 1/(9 e^(31/10))}, {32, 1/(9 e^(16/5))}, {33, 1/(9 e^(33/10))},
{34, 1/(9 e^(17/5))}, {35, 1/(9 e^(7/2))}, {36, 1/(9 e^(18/5))}, {37, 1/(9 e^(37/10))}, {38, 1/(9 e^(19/5))},
{39, 1/(9 e^(39/10))}, {40, 1/(9 e^4)}, {41, 1/(9 e^(41/10))}, {42, 1/(9 e^(21/5))}, {43, 1/(9 e^(43/10))}, {44, 1/(9 e^(22/5))},
{45, 1/(9 e^(9/2))}, {46, 1/(9 e^(23/5))}, {47, 1/(9 e^(47/10))}, {48, 1/(9 e^(24/5))}, {49, 1/(9 e^(49/10))}, {50, 1/(9 e^5)}}

```

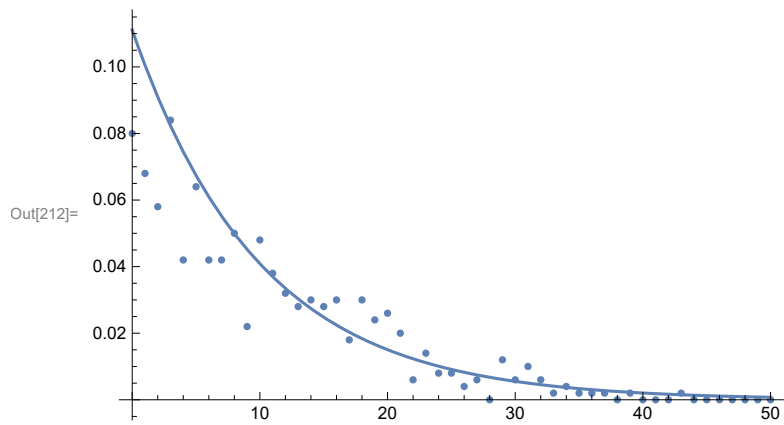
```
In[211]:= funcion = ListPlot[sdaga]  

[representación de lista
```



```
In[212]:= Show[Plot[(E^(-x/mini))/9, {x, 0, 50}], ListPlot[histo]  

[mue... [repre... [número e [representación de lista
```



Bibliografía

- [1] HEYMANN, PEARAZZO Y ZIMMERMANN, *Economía de fronteras abiertas*, primera edición, Teseo, Argentina, Buenos aires, 2013.
- [2] EAGLEMAN, DAVID, *Incognito las vidas cecretas de tu cerebro*, tercera edición, Anagrama, España, Barcelona, 2013.
- [3] PENROSE, ROGER, *La mente nueva del emperador*, cuarta reimpresión, FCE, México, Ciudad de México, 2018.
- [4] HAWKING, STEPHEN, *Dios creo los números*, quinta mpresión, Critica, España, Barcelona, 2013.
- [5] WADSWORTH, G.P ET AL., *Aplicaciones de la teoría de probabilidades y variables aleatorias*, primera edición, Alhambra, España, Barcelona, 1979.
- [6] BUNGE, MARIO, *Epistemología*, octava reimpresión, Siglo XXI, México, Ciudad de México, 2015.