



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

ESCOM

Trabajo Terminal

Caracterización de las evoluciones fractales de los autómatas celulares elementales

Presentan

Eric Fabián Peralta Ramírez

Carlos Zacarías Reyes Martínez

Directores

Doctor Genaro Juárez Martínez

Noviembre 2019





INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA



No de TT:2018-B141 13 de Noviembre de 2019

Documento Técnico

**Caracterización de las evoluciones fractales de los autómatas
celulares elementales**

Trabajo Terminal No.2018-B141

Presentan:

Eric Fabián Peralta Ramírez¹

Carlos Zacarías Reyes Martínez²

Director

Doctor Genaro Juárez Martínez

RESUMEN Este trabajo terminal propone una caracterización de los autómatas celulares elementales que producen en su configuración global patrones fractales. Se pretende desarrollar un simulador con el lenguaje de programación Python, esta caracterización en conjunto con el simulador nos permite describir las reglas fractales mediante un análisis de las historias de evoluciones, las gráficas de los polinomios de la teoría del campo promedio, los campos de atractores y una medida de entropía de los sistemas caóticos basada en la entropía de Shannon.

Palabras Clave: Autómatas Celulares Elementales, Sistemas Caóticos, Fractales.

¹fabperaltar@gmail.com

²carloszrm90@gmail.com

Resumen

El presente trabajo terminal propone una caracterización de los patrones fractales en los autómatas celulares elementales a partir de la realización de un estudio sistemático de las características fenotípicas, genotípicas o morfológicas de estos autómatas celulares que presentan comportamiento caótico y que en la configuración global proyectan fractales. Se definirá esta caracterización mediante la aplicación de herramientas matemáticas utilizadas en los sistemas dinámicos que nos permiten describir la dinámica de los autómatas a nivel fenotípico, genotípico y morfológico [12][1]. Una pregunta central en los autómatas celulares es la siguiente ¿Qué clasificación del comportamiento de los autómatas celulares pueden establecerse? [35] Así que varios artículos definen una forma de caracterizar los autómatas celulares [19] y dada la diversidad de clasificaciones podemos establecer otra pregunta: ¿Puede establecerse una clasificación completa? La respuesta que intentamos ejemplificar es que no se puede establecerse esta clasificación y para hacerlo tomamos el conjunto de autómatas celulares elementales que producen fractales y mostramos que en las clasificaciones en la literatura este conjunto no puede empatarse con alguna clasificación.

Índice general

1. Introducción	7
1.1. Resumen del capítulo	7
1.2. Autómatas celulares	7
1.3. Fractales	10
1.4. El problema de la caracterización de los sistemas dinámicos	11
1.4.1. Justificación	13
1.5. Organización del trabajo	16
2. Definiciones y conceptos	17
2.1. Resumen del capítulo	17
2.2. Sistemas dinámicos discretos	17
2.3. Autómatas celulares	22
2.4. Fractales	28
2.5. Observaciones finales	31
3. Caracterización	33
3.1. Resumen del capítulo	33
3.2. Construyendo fractales en los Autómatas Celulares elementales	34
3.2.1. Fractales en Autómatas celulares	34
3.2.2. Morfología de las reglas	55
3.3. Caracterización Fenotípica	56
3.3.1. Exponentes de Lyapunov	56

3.3.2.	Perfiles de Lyapunov	57
3.3.3.	Entropía	58
3.4.	Caracterización genotípica	66
3.4.1.	Teoría del campo promedio	66
3.4.2.	Cuenca de Atractores	73
3.4.3.	Propuesta	84
3.5.	Observaciones finales	85
4.	Comparación con clasificaciones en la literatura	87
4.1.	Resumen del capítulo	87
4.2.	Propuesta de la caracterización	88
4.2.1.	Caracterizaciones propuestas	88
4.3.	Clasificaciones fenotípicas	93
4.3.1.	Clasificación de Wolfram	93
4.3.2.	Surface dynamics classification	93
4.3.3.	Clasificación de Wuensche	94
4.3.4.	Expressivity analysis	94
4.4.	Clasificaciones genotípicas	95
4.4.1.	Clasificación de Li y Packard	95
4.4.2.	Topological dynamics classification	96
4.4.3.	Topological classification	97
4.4.4.	Creativity	97
4.5.	Observaciones finales	100
5.	Simulador	101
5.1.	Introducción	101
5.2.	Simulador	102
5.2.1.	Metodología	102
5.2.2.	Patrón de arquitectura	106

<i>ÍNDICE GENERAL</i>	5
5.2.3. Diagramas de clase	108
5.2.4. Software	112
5.3. Conclusiones	138
5.4. Limitaciones	139
5.5. Trabajo futuro	140
A. Tablas y Figuras	143
A.1. Tablas	143
A.2. Imágenes	148
A.3. Índice	153
B. Bibliografía	155

Capítulo 1

Introducción

1.1. Resumen del capítulo

En el primer primer capítulo presentaremos el marco referencial de la teoría de autómatas celulares y los fractales. También el concerniente al problema de la clasificación y mencionaremos algunos obstáculos que ocurren al realizar una clasificación de los sistemas dinámicos.

1.2. Autómatas celulares

En la década de los 40 John von Neumann inicia la investigación de los autómatas celulares. El modelo de von Neumann es un autómata celular de dos dimensiones y cuyas células son cuadradas, éstas pueden tener un valor de un conjunto de veintinueve estados y la función de transición considera la vecindad de von Neumann. [25],

John von Neuman buscaba simplificar la complejidad del universo en sistemas cuya interacción de elementos simples produjera en algún grado la complejidad del Universo. Además el buscaba maquinas que tuvieran: Universalidad lógica,.

- A. Universalidad lógica ¿Cuándo una clase de autómatas es lógicamente universal, es decir, capaz de realizar todas esas operaciones lógicas que se pueden realizar con

medios finitos (pero arbitrariamente extensos)?.

Este es un campo muy extenso en los autómatas celulares universales, Una de las vías principales para lograr computo universal es construir compuertas universales, son aquellas compuertas u operadores lógicos que pueden expresar a todos los demás operadores únicamente con composiciones o arreglos de ellos mismos. Para lograr esto puede utilizarse computo basado en colisiones, en donde se modelan partículas que al colisionar simulan estas operaciones o compuertas lógicas [20].

- B. Constructibilidad. ¿Puede un autómata ser construido, es decir, ensamblado y construido a partir de "materias primas" adecuadamente definidas por otro autómata? O, comenzando desde el otro extremo y extendiendo la pregunta, ¿qué clase de autómatas se puede construir con un autómata y un conjunto de datos adecuado? Los apogos variables, pero esencialmente estándar, a este último, en el sentido de la segunda pregunta de (A), pueden ser permitidos.

También nos podemos preguntar ¿Puede cualquier autómata construir otros autómatas que luzcan exactamente igual? ¿Se puede hacer además para realizar otras tareas, por ejemplo, también construir otros autómatas prescritos? Y si existe tal autómata que variables, ingredientes o condiciones necesita para crear ese Autómata.

Esta idea podemos verla en la reproducción celular. Dónde una célula puede dividir y duplicar su código genético para crear una segunda célula con la misma información e idéntica a la primera. Un ejemplo en el mundo de los autómatas celulares podemos observarlo en la búsqueda de procedimientos para generar Gliders Gun. Estas estructuras son capaces de generar otras estructuras más simples que pueden moverse independientemente por el espacio de evoluciones de los autómatas celulares. Se puede generalizar los dos puntos anteriores.

- Es posible crear un autómata que construyan autómatas lógicamente universales. Y

si existe tal autómeta que variables, ingredientes o condiciones necesita para crear un autómeta lógicamente universales.

Quizá el ejemplo más adecuado para ilustrar está idea lo volveremos a encontrar en la reproducción celular. Que células o mecanismos son necesarios para crear una célula madre , recorremos que la célula madre pueden convertirse en cualquier célula de un organismo. En el universo de los autómatas está idea puede ilustrar con investigaciones que intentan simular otro tipo de Autómatas con un tipo específico de autómatas celulares. Por ejemplo hay trabajos que simulan máquinas de Turing o autómatas en dos dimensiones que simulan autómatas en una dimensión.

A estas preocupaciones de von Neumann podemos extender una. ¿Que tipo de Autómatas pueden producir o simular funciones o algoritmos que no son lógicamente Universales ? Es decir , pueden encontrarse funciones que sean resueltas con ciertas variables o condiciones iniciales pero a partir de la evolución de los autómatas .

Sin duda no encontramos con un campo muy extenso de investigación en la teoría de autómatas celulares. Este campo aplicativo busca replicar o generar funciones matemáticas y físicas en algunos modelos de autómatas celulares. Por ejemplo se pueden construir autómatas que generen números primos, resuelvas ciertas ecuaciones diferenciales o simulen operaciones criptográficas.

Nuestro trabajo terminal puede inscribirse en este campo ya que elegimos un tipo específico de autómatas, condiciones iniciales para que se pueda producir una función que genera fractales. Es importante mencionar que la mayoría de funciones que general el triangulo de Sierspinski son funciones recursivas y la evolución de fractales en autómatas celulares elementales es iterativa .

1.3. Fractales

Para hablar de fractales tomaremos como referencia la tesis: "Introducción a la geometría fractal y a la dinámica del caos"[26], esta tesis será la guía en la que basaremos el desarrollo y caracterización de nuestro trabajo ya que presenta una distinción entre fractales regulares e irregulares, la misma que ocuparemos nosotros para caracterizar los fractales que son producidos en los autómatas elementales.

Como en el concepto de caos en el concepto de fractal no existe una definición aceptada, esto se debe a que no todos los objetos que deberían ser considerados como fractales cumplen con todas las características de las definiciones propuestas. Los fractales fueron propuestos por Benoît Mandelbrot en 1979 [16][17].

El propósito que tiene para describir estos objetos es sobre el alcance de las matemáticas, éstas solo podían describir objetos regulares y continuos, por lo que muchos objetos presentes en la realidad o naturaleza no pueden ser descritos por esta. Para Mandelbrot la etimología donde nace fractal pueda dar cuenta del significado de estos objetos, también menciona que un fractal natural es aquel que puede representarse por una función matemática y que posee una dimensión no entera. Él escribe en [17]:

Yo acuñé *fractal* del adjetivo latino *fractus*. Este corresponde al verbo en latín *frangere* que significa romper; Crear fragmentos irregulares. Esto es apropiado y razonable para nuestros propósitos. Adicionalmente a fragmentado (como en fracción y refracción) *fractus* debería también significar irregular.

El propósito de la geometría fractal es representar los objetos que pueden encontrarse en la naturaleza. Un ejemplo es la geometría de algunos árboles. Podemos describir la geometría de un árbol como una dendrita, es decir un tronco que se bifurca en varias ramas y cada rama se bifurca en ramas más pequeñas hasta llegar a las hojas. Pero también podemos describir de la misma forma a las raíces y a las venas de las hojas e

1.4. EL PROBLEMA DE LA CARACTERIZACIÓN DE LOS SISTEMAS DINÁMICOS 11

incluso la observamos en la timidez de los arboles, este fenómeno ocurre entre algunas especies de arboles de copas frondosas donde las copas no se tocan. Por esto podemos afirmar que a diferentes escalas el objeto posee la misma geometría.



Dendritas en árboles. Podemos observar como se ven los mismos patrones a diferentes escalas del objeto, tanto en las hojas como en las ramas

1.4. El problema de la caracterización de los sistemas dinámicos

El problema de la caracterización de los sistemas dinámicos es uno de los temas importantes en la investigación en ciencias de la complejidad y caos. Al proponer una caracterización se seleccionan una serie de herramientas matemáticas para describir alguna propiedad del sistema. Estas herramientas nos ayudan a responder que tan **complejo** es un sistema. [35] [22][19].

Esta selección no es fácil ya que nos enfrentamos a varios problemas. Un problema está relacionada con la complejidad de los sistemas que se manifiesta en múltiples y diversas particularidades, por lo que para describir la dinámica del sistema en su totalidad se necesitarían una alta diversidad de herramientas, tantas como características podemos observar. Debido a ésta dificultad se realiza una selección de las propiedades mas

importantes de los sistemas, pero esto también implica un problema metodológico; El comportamiento de un sistema depende de la herramienta utilizada para medirlo y los resultados al aplicar diversas herramientas de observación pueden ser aparentemente contradictorios, por ejemplo la dualidad onda partícula.

Un segundo problema es la escala a la que se puede analizar el sistema, por ejemplo algunas propiedades describen el comportamiento global del sistema y otras propiedades son el resultado del comportamiento local de los elementos. En un sistema complejo los elementos que lo conforman tienen comportamientos simples entre ellos y el resultado de esta dinámica local emerge a una dinámica global. Pero también la teoría de sistemas nos permite modelar sistemas que son parte de otros sistemas[4]. Un ejemplo es modelar un planeta como un sistema *súper* complejo, formado de sistemas naturales, físicos, químicos. Y en una escala mayor describir las posiciones y velocidades de los planetas en un sistema planetario con el problema de los N cuerpos.

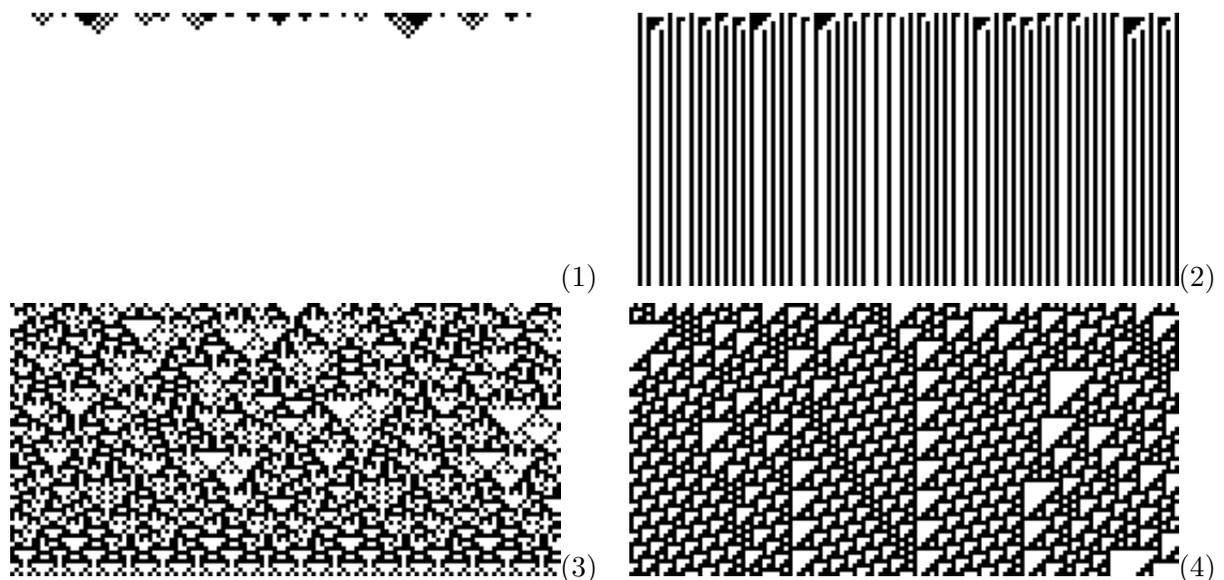
Stephen Wolfram es el primer investigador en proponer los autómatas celulares de una dimensión, que son los autómatas más sencillos que se pueden representar [33][34][8]. Wolfram también propone una caracterización que realizó mediante propiedades estadísticas, utilizó la entropía de Shannon para describir el comportamiento de las configuraciones globales a través del tiempo. En *Twenty Problems in the Theory of Cellular Automata* [35] escribe:

Las matemáticas experimentales proporcionan un primer acercamiento a éste problema. Si se realizan simulaciones de la evolución del autómata celular se puede intentar encontrar reglas empíricas para su comportamiento. Estas pueden sugerir resultados que pueden ser investigados por métodos matemáticos más convencionales.

En este artículo plantea el problema: ¿Cual clasificación general de los autómatas celulares se puede establecer? y propone una clasificación empírica y cualitativa. A partir de configuraciones desordenadas (Una configuración desordenada es en la que cada estado es elegido de forma independiente para tener un valor de 1 con probabilidad p) evoluciona cada regla y compara las características de sus espacios de evolución. Las clases propuestas

son :

- Clase I. La evolución conduce a un estado homogéneo
- Clase II. La evolución conduce a un conjunto de estructuras simples y separadas o a estructuras periódicas.
- Clase III. La evolución conduce a patrones caóticos.
- Clase IV. La evolución conduce a estructuras complejas y localizadas, a veces de larga vida.



Cuadro 1.1: Ejemplos de la clasificación de Wolfram. 1. Autómata celular Clase I. 2. Autómata celular Clase II . 3. Autómata celular Clase III . 4. Autómata celular Clase IV.

1.4.1. Justificación

Las caracterizaciones y clasificaciones existentes de los autómatas celulares están enfocadas en describir el grado de complejidad y caos de las reglas por lo que existen pocas caracterizaciones en los autómatas celulares elementales que describan primordialmente los patrones fractales [27] [13]. Podemos observar una extensa revisión y comparación de las principales características en [19] . Mientras que los trabajos realizados sobre fractales

en autómatas celulares se enfocan en las propiedades matemáticas de los fractales [32] [18].

Uno de los trabajos más importantes sobre fractales lo realizó Karel Culik II en [13]. Él propone una subclasificación de los autómatas celulares unidimensionales con comportamiento caótico. Algunos autómatas celulares pertenecientes a la clase caótica muestran un comportamiento más regular, es decir no es completamente caótico, y algunos incluso pueden generar patrones similares a un fractal. La clasificación es fenotípica y divide a la clase caótica por el comportamiento de las evoluciones globales. Ésta subclasificación considera tres clases

1. La evolución conduce a un patrón altamente regular y recurrente (casi fractal) En pequeñas configuraciones iniciales, el patrón es un fractal puro con una dimensión fractal computable. (Kulick menciona que para ser un fractal puro su evolución puede ser caracterizada por una fórmula recursiva)
2. Para configuraciones iniciales aleatorias, la evolución conduce a un patrón caótico, excepto en configuraciones iniciales cortas para las cuales el patrón es fractal o casi fractal
3. La evolución conduce a un comportamiento caótico.

La caracterización de los fractales que se producen en el espacio de evoluciones de los autómatas celulares elementales que realizaremos en el presente Trabajo Terminal la construimos a partir de herramientas matemáticas que describan a los autómatas a través de dos tipos de rasgos [12][1][31]:

- Fenotípicos, qué estructura tienen los fractales que se producen.
- Genotípicos, qué propiedades tienen los autómatas que producen fractales.
- Morfológicas, Cuál es la diversidad de fractales que son producidos.

Las caracterizaciones y clasificaciones generalmente se enfocan en solo un tipo de rasgos, por lo que un aporte importante del Trabajo Terminal es cubrir las tres características

1.4. EL PROBLEMA DE LA CARACTERIZACIÓN DE LOS SISTEMAS DINÁMICOS¹⁵

y de esta manera obtener mas información del comportamiento caótico y fractal de las reglas. La características fenotípicas se describen a partir de las siguientes herramientas.

- Entropía de Shannon.
- Exponentes de Lyapunov.

La características genotípicas se describen a partir de las siguientes herramientas.

- Campos de Atractores
- Polinomios del Campo Promedio

La características morfológicas se describen a partir de las siguientes herramientas.

- Morfología Matemática

1.5. Organización del trabajo

En este capítulo se introdujeron los autómatas celulares, los fractales y el problema de la clasificación, también se presentó el objetivo y la justificación del trabajo.

- En el segundo capítulo se presentan las definiciones formales de Sistemas Dinámicos, Fractales y Caos.
- En el tercer capítulo se presenta la caracterización que realizamos.
- En el cuarto capítulo comparamos nuestra clasificación con trabajos en la literatura de autómatas celulares elementales.
- El capítulo cinco mostramos las conclusiones del trabajo.

Capítulo 2

Definiciones y conceptos

2.1. Resumen del capítulo

En este capítulo se escriben las definiciones, notaciones y elementos que se utilizarán en este trabajo para realizar un análisis sistemático del comportamiento caótico de los autómatas celulares elementales que en la dinámica global del sistema producen fractales parecidos al triángulo de Sierpinski. La organización de este capítulo es la siguiente: Primero presentaremos definiciones formales de Sistemas Dinámicos y Caos además de algunos conceptos que nos permitan mostrar un panorama introductorio, después mostraremos las definiciones que presenten a los Autómatas celulares elementales y su funcionamiento, también nos interesa mostrar las propiedades dinámicas de estos sistemas por lo que ejemplificaremos los conceptos presentados en la primera sección en estos sistemas. En la tercera parte mostraremos una definición formal de Fractales así como las consideraciones que tomamos para poder utilizar una definición de fractal a lo largo de la tesis.

2.2. Sistemas dinámicos discretos

La teoría de sistemas dinámicos (o dinámica) se refiere a la descripción y predicción de sistemas que muestran un comportamiento cambiante complejo a nivel macroscópico, que

surge de las acciones colectivas de muchos componentes que interactúan. La palabra dinámica significa cambiar, y los sistemas dinámicos son sistemas que cambian con el tiempo de alguna manera. [24]. Algunos ejemplos de sistemas dinámicos son : Sistema solar, sistemas cazador-presa, péndulo, sistemas de reacción - difusión, modelo logístico.

Un sistema dinámico puede ocurrir en tiempo continuo o en tiempo discreto, por lo que ésta es la primera división de los sistemas dinámicos. Para entender la naturaleza de la división debemos conocer la diferencia entre variables discretas y continuas. Una variable es continua si su conjunto de posibles valores está contenido en el conjunto de los números reales mientras que una variable es discreta si se puede asignar a cada elemento de su conjunto de posibles valores un número en una sucesión de números naturales. Las definiciones formales se establecen de la siguiente manera:

Definición 2.2.1. Sistemas dinámicos continuos

Un sistema dinámico continuo es la pareja $P < X, f >$. Donde X es un espacio vectorial y f un conjunto de funciones que lo definen, estas son diferenciales ordinarias de la forma $\frac{\partial x}{\partial t} = X(x)$ donde $x \in X$ [29].

Definición 2.2.2. Sistemas dinámicos discretos

Un Sistema dinámico discreto Es la pareja $P < X, f >$. Donde X es un espacio métrico y f una función continua del espacio X en sí mismo, $f : X \rightarrow X$. Al conjunto X suele denominarse *Espacio de fase* y es el conjunto de todos los valores globales que puede tener el sistema.

Para cada $n \in \mathbb{N}$, consideramos la n-esima composición o iteración de la función f como $f^n = f \circ \dots \circ f \circ f$ (\mathbb{N} veces) . Además decimos que f es invertible si $f^{-n} = f^{-1} \circ \dots \circ f^{-1}$ (\mathbb{N} veces) [14].

Se pueden pedir varias características al conjunto X pero nosotros solo pedimos que sea un *espacio métrico*, es decir existe una función métrica que nos da la distancia entre dos elementos del sistema. Así que definimos la métrica de la siguiente manera.

Definición 2.2.3. Métrica

Sea un espacio X no vacío, se define la función métrica $d : X \times X \rightarrow \mathbb{R}$ y satisface que

- $d(x, y) \geq 0$ y $d(x, y) = 0 \iff x = y$
- $d(x, y) = d(y, x)$
- $d(x, z) \leq d(x, y) + d(y, z)$

Para poder describir la *dinámica* del sistema debemos describir el comportamiento de cada elemento del espacio de fase producido por la función de evolución y lo hacemos al mostrar el camino o la serie de iteraciones que tiene que recorrer cada elemento a través del tiempo, entonces podemos decir que el sistema estaba en un estado x_i y después de un tiempo t ahora está en un estado x_k y también nos interesa relacionar estos dos estados mediante la métrica definida, ahora decimos que el sistema se movió una distancia $d(x_i, x_j)$ en un tiempo y .

Cuando intentamos describir la dinámica de todo el sistema tomamos cada estado o valor posible y al observar sus historias nos damos cuenta que existen estados *atractores* a los que converge un conjunto de elementos después de un tiempo o que en ocasiones volvemos al estado de partida. Para definir formalmente la historia de cada elemento se debe definir el concepto de *orbitas* de la siguiente manera:

Definición 2.2.4. Orbita

Para todo x en X existe una sucesión de puntos u orbita de x bajo f tal que :

$$o(x, f) = \{x, f(x), f^2(x), \dots, f^n(x)\}$$

Donde n pertenece al conjunto de los naturales \mathbb{N} y f^n es la n-esima composición consigo misma de f [14].

Mientras que para definir los puntos a los que converge el sistema debemos definir los conceptos de *punto fijo* de la siguiente manera:

Definición 2.2.5. Punto fijo

Tomando un punto $x_0 \in X$ decimos que es un punto fijo de periodo n si $f^n(x_0) = x_0$ por lo que la orbita de x_0 se expresa de la siguiente manera

$$o(x_0, f) = \{x_0, x_1, x_2, \dots, x_i, x_0, x_1, x_2, \dots\}$$

Podemos decir que si $\lim_{x \rightarrow \infty} f^n(x_0) = y_0$ la orbita $o(x_0, f)$ converge a x_0 [14].

Al conjunto de puntos periódicos de f se le nombra $Per(x)$

Estas definiciones son necesarias para definir el caos como lo hizo Devaney en [3]:

Definición 2.2.6. Sistemas caóticos

Teniendo un sistema dinámico $P < X, f >$ Decimos que es caótico si cumple que:

1. El conjunto $Per(X)$ es denso en X .
2. f es transitiva en X
3. f es sensible a las condiciones iniciales

Definición 2.2.7. Un sistema dinámico $P < X, f >$ tiene una dependencia sensible en condiciones iniciales en un subconjunto $X' \subset X$ si existe una $\epsilon > 0$ tal que para toda $x \in X'$ y $\delta > 0$ existe una $y \in X$ y $n \in \mathbb{N}$ para los cuales $d(x, y) < \delta$ y $d(f^n(x), f^n(y)) > \epsilon$ [5]

Este trabajo está enfocado en el tercer punto de la definición de caos, pero en [6] podemos observar una descripción precisa de cada requisito de la definición de caos de Devaney en los Autómatas celulares. Esta definición analiza el cambio de las distancias de dos elementos a través del tiempo. Nos dice que un conjunto es sensible a las condiciones iniciales si para cada elemento en ese subconjunto podemos encontrar otro elemento en el subconjunto cuya distancia inicial crezca al aplicar la función en un tiempo n así que estos elementos están separados en algún instante. Usualmente se dice que un sistema es muy sensible a las condiciones iniciales si la distancia inicial δ es muy pequeña, una pequeña perturbación en el sistema genera una distancia $d(f^n(x), f^n(y))$ muy grande.

La herramienta elegida en este trabajo para hablar de las condiciones iniciales son *los exponentes de Lyapunov* estos nos dice la velocidad con la que se van alejando los elementos del sistema.

La definición considera un caos basado en las propiedades topológicas de los sistemas dinámicos por lo que la definición de *Sensible* menciona los conceptos de vecindad, conjuntos abiertos, cerrados, densos y transitividad, estas definiciones topológicas pueden verse de forma clara en [14] [9] Algunas son explicadas en el caso de Autómatas celulares en [31] .

2.3. Autómatas celulares

Un autómata celular es un sistema dinámico discreto en espacio y tiempo, Al utilizar la definición de Sistemas dinámicos 2.2.2 debemos describir el espacio X y la función f que forman este sistema dinámico. También es necesario que el espacio sea métrico. Seck Tuoh muestra en [30] las propiedades topológicas de los autómatas celulares unidimensionales .

Un autómata celular esta formado por un espacio E que es N -dimensional. Es de longitud infinita y esta fracturado en espacios de longitud finita x_i dónde $i \in \mathbf{Z}$ llamados células, éstas pueden tener formas geométricas diversas y están conectadas entre si. Pero al no tener una memoria infinita se debe limitar este espacio a uno finito estableciendo condiciones de contorno. Una condición de contorno es una operación en los extremos del espacio de evolución de manera que todas las células siempre estén conectadas.

Cada célula posee un valor o estado a_k que pertenece a un conjunto limitado de valores posibles o alfabeto Σ , este conjunto únicamente contiene elementos que son enteros. Una configuración \mathcal{C} es la asignación de un atributo a_k a cada célula x_i . Una configuración finita tiene una longitud l , donde l es el número de células del espacio de evoluciones. [36].

En las siguientes imágenes se muestran ejemplos de autómatas celulares en varias dimensiones, se puede observar que la geometría y dimensión de las células concuerdan con las características del espacio. Si se gráfica el comportamiento del sistema en el tiempo se debe agregar otra dimensión, así un autómata en una dimensión se observa como un objeto en dos dimensiones.

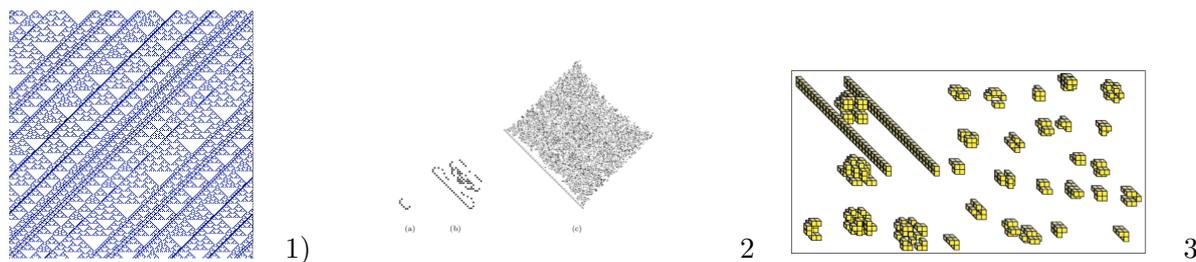


Figura 2.1: 1) Evolución en una dimensión de la regla 154.
 2) Evolución en dos dimensiones de un glider en la regla de Difusión $R(7722)$.
 3) Evoluciones en tres dimensiones de la regla $R(4, 5, 5, 5)$

Para la descripción de la función es necesario mencionar el concepto de de vecindad. Una vecindad r define una distancia r que conecta a una célula con los vecinos conectados dentro de esta vecindad. La vecindad depende de la dimensión del autómata celular y la forma de las células. La función local φ determina las transformaciones locales de cada posible vecindad.

La función global Φ relaciona cada configuración C en el tiempo. Esta función es la órbita del autómata celular. Llamaremos *espacio de evoluciones* a la concatenación de los resultados de aplicar la función global a través del tiempo, es decir a una *órbita* del autómata celular. Los autómatas son N-dimensionales por lo que el espacio de evoluciones es una gráfica de una dimensión superior, por ejemplo en un autómata de una dimensión con una condición de frontera el espacio de evoluciones será un cilindro hueco.

Autómatas celulares elementales

Wolfram establece la notación de los Autómatas celulares de una dimensión a partir de dos elementos (k, r) donde k representa la cardinalidad del conjunto de estados, entonces $k = |\Sigma|$ y r es el numero de vecinos a la izquierda y derecha de una célula central x_i [33]. Un **autómata celular elemental**, ECAM por sus siglas en inglés esta representado por la pareja $(2, 1)$. Definimos a los ECAM como:

Definición 2.3.1. Autómata celular elemental

Está formado por un arreglo unidimensional de células de longitud l que pueden tener el estado de 0 o 1.

x_0	x_1	...	x_i	...	x_l
-------	-------	-----	-------	-----	-------

Un ECAM se puede definir como la tupla.

$$\langle \Sigma, r, \varphi, c_i \rangle$$

Donde

- Σ es el conjunto de estados posibles.
- r es el número de vecinos con respecto a una célula central
- φ es la función de transición.
- c_i configuración inicial del sistema. [21]

La *vecindad* está compuesta por r vecinos a la derecha y a la izquierda de una célula central x_i , por lo que la vecindad esta compuesta por las células: $x_{i-r} \dots x_i \dots x_{i+r}$. La vecindad tiene una dimensión $2r + 1$. El numero de vecindades posibles esta dado por k^{2r+1} La función local φ es una función que indica la transformación específica de cada vecindad a un elemento de Σ . Los valores x_i están dados por $x_i = \varphi(x_{i-r} \dots x_i \dots x_{i+r})$. Para producir la siguiente evolución del sistema se aplica a cada célula la función φ . No

se puede calcular un autómata de longitud infinita por lo que se aplica una condición de contorno. La condición que se ocupa en el trabajo es la concatenación de la primera célula y la última para formar un anillo como se muestra en la tabla 2.2.

Una regla de evolución es la función de transición aplicada en cada una de las vecindades con respecto al valor k y r , cada vecindad tendrá una correspondencia con un elemento del conjunto de estados Σ . Las reglas son numeradas de acuerdo a la lectura de las producciones x_i^{t+1} como un número binario, siendo el dígito menos significativo la primera producción [20]. Este número es representado en base decimal. Observar tabla 2.1.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Cuadro 2.1: Ejemplo de la regla de evolución local. Se produce la regla 90

Espacio de evoluciones y Árboles de atractores

El *Espacio de Fase* es el conjunto de todos los posibles estados globales que puede tener el autómata celular. Existe un subconjunto llamado *Jardines del edén* al que pertenecen todas las configuraciones que no pueden ser producidas por la evolución de otra configuración global perteneciente al espacio de fase. El comportamiento en los autómatas celulares puede observarse a través el espacio y tiempo en el que evoluciona el sistema dinámico. Para poder observarlo se concatenan las configuraciones globales que son producidas por una configuración inicial c_0 en un tiempo t_0 hasta un tiempo t_i , esta concatenación es la *Orbita* de c_0 bajo φ , también se le conoce como *trayectoria*. Para describir el comportamiento

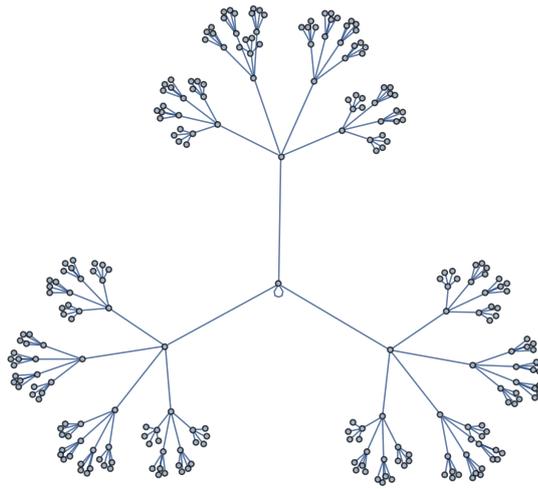


Figura 2.2: Órbita de la regla 90 con un anillo de ocho células

2.4. Fractales

La primera noción de fractal es visual, un fractal es un objeto que es auto similar, esto es: A todas las escalas a las que podamos contemplarlo, descubrimos que la parte es igual que el todo. Es decir si descomponemos un objeto en sus partes y observamos estas partes descubrimos que tienen la misma estructura que el objeto. [9]. [29]

Mientras que en matemáticas existen varias formas para definir los fractales, la primera definición es por la propiedad de auto similitud y la segunda por su dimensión:

Definición 2.4.1. Definición de fractal por Auto similitud Un objeto que conserva la propiedad de auto similitud en varias escalas de magnificación del objeto.

Definición 2.4.2. Definición de fractal por Dimensión Un objeto cuya dimensión fractal es mayor que su dimensión topológica y menor que su dimensión euclidiana.

Estas definiciones nos permiten realizar una distinción entre fractales regulares y Aleatorios. Un *fractal regular* es aquel que posee auto similitud exacta y una dimensión fractal mayor que la dimensión topológica, mientras que un *fractal aleatorio* no tiene auto similitud exacta, sino cada pequeña parte de un fractal tiene las mismas propiedades estadísticas que el total.

Dimensión de auto similitud

Los fractales son objetos con geometría compleja y la definición formal no existe. Una de las principales características que poseen es su dimensión fraccionaria. La dimensión de auto similaridad se obtiene a través del análisis de las iteraciones de la función que genera el fractal. Se cuentan las N copias más pequeñas generadas y se considera el factor de escala M , que es la relación entre la longitud del objeto original y las copias más pequeñas. La dimensión esta dada por la función. $N = (M)^D$ por lo que $D = \frac{\log N}{\log M}$ Para ejemplificarlo tomaremos una linea, un cuadrado y un cubo y lo amplificaremos para tener el doble de longitud y crearemos las copias para que conserven su dimensión [11].

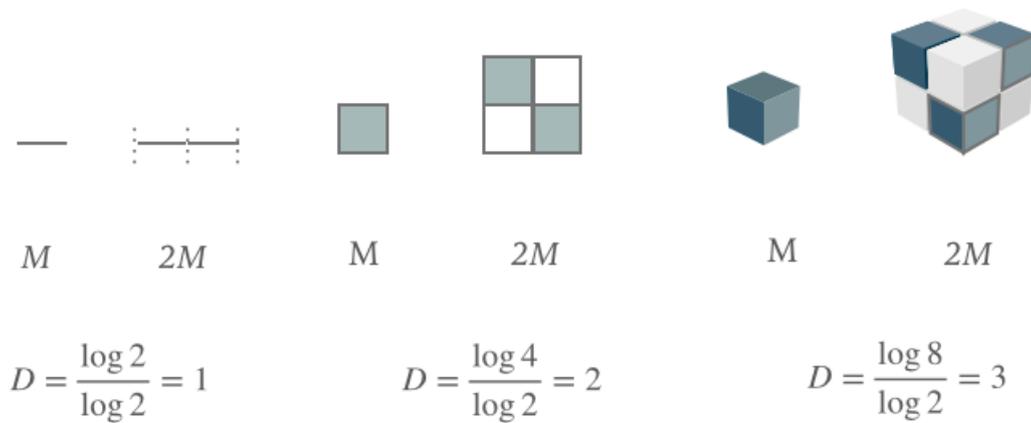


Figura 2.3: Ejemplos de dimensión de auto similitud. 1. La línea es de dimensión uno, 2. El cuadrado es de dimensión dos, 3. El cubo es de dimensión tres

Triángulo de Sierpinski

Podemos producir fractales a partir de una función recursiva que es aplicada a cada parte del objeto, El triángulo de Sierpinski es un triángulo de área D y perímetro P al que se le sustrae el triángulo equilátero interior que se forma uniendo los puntos medios de cada vértice. Quedan unidos tres triángulos de área $\frac{A}{4}$ y perímetro $\frac{3}{2}P$. Al iterar la función infinitamente se observa que el área decrece hasta converger a 0, mientras que la longitud crece hasta que diverge a infinito. Las primeras tres iteraciones del triángulo se describen en la tabla 2.3

Iteración	Perímetro	Área
	$3L$	A
	$\left(\frac{3}{2}\right) 3L$	$\left(\frac{3}{4}\right) A$
	$\left(\frac{3}{2}\right)^2 3L$	$\left(\frac{3}{4}\right)^2 A$
	$\left(\frac{3}{2}\right)^n 3L$	$\left(\frac{3}{4}\right)^n A$
$n \rightarrow \infty$	$\lim_{n \rightarrow \infty} \left(\frac{3}{2}\right)^n 3L = \infty$	$\lim_{n \rightarrow \infty} \left(\frac{3}{4}\right)^n A = 0$

Cuadro 2.3: Iteraciones del triángulo de Sierpinski

2.5. Observaciones finales

En este capítulo relacionamos los conceptos de Sistemas dinámicos caóticos y los autómatas celulares elementales. Una limitación de este trabajo es el concepto de Caos. Por un lado se presentó una noción de fuerte de caos con la definición de Devaney y por el otro el otro una definición débil de caos en los conjuntos atractores. Queda como trabajo futuro mostrar formalmente que los conjuntos atractores de las reglas fractales son caóticas en el sentido de Devaney. Para esto debemos mostrar que : *Son sensibles a las condiciones iniciales, Son densos y la función (regla) es transitiva.* Debemos recordar que en todas las reglas mostradas se presentó caos en la evolución del autómata.

Capítulo 3

Caracterización

3.1. Resumen del capítulo

En este capítulo mostraremos y describiremos los fractales que se presentan en los autómatas celulares elementales. Este capítulo se construyó a partir de la separación de clasificaciones que se hizo en el capítulo 1.4.1. Para describir la capacidad generativa de fractales en los autómatas elementales retomaremos la definición presentada de fractales regulares e irregulares, además mostraremos los fractales regulares más pequeños que se forman por cada una de las reglas de evolución así como el conjunto atractor donde están contenidas las orbitas de estos fractales. Al hacer la caracterización por cualidades genotípicas y fenotípicas introduciremos las herramientas matemáticas que utilizamos para describir cada característica, además de hacer una comparación entre las reglas fractales. En todo momento intentaremos ordenar las reglas dependiendo la dinámica mostrada por cada herramienta y utilizaremos como criterio de desempate la Entropía calculada ya que muestra el comportamiento global de todo el sistema.

3.2. Construyendo fractales en los Autómatas Celulares elementales

3.2.1. Fractales en Autómatas celulares

En este apartado presentamos las reglas que hemos encontrado que producen patrones fractales y agregamos las reglas equivalentes. Las reglas equivalentes fueron propuestas por Li y Packard en [15] y definen cluster de reglas cuyos elementos pueden expresarse al aplicar una función a sus elementos. Wuensche realiza una clasificación a partir de estas operaciones [36]

Regla equivalente	Left-right	color	both
18	18	183	183
22	22	151	151
26	82	167	181
60	102	195	153
90	90	165	165
94	94	133	133
105	105	105	105
106	120	169	225
122	122	161	161
126	126	129	129
146	146	182	182
150	150	150	150
154	210	166	180
164	218	218	218

Cuadro 3.1: Reglas fractales

La primera característica que nosotros observamos en los fractales es la regularidad del triángulo de Sierpinski producido. Un **fractal regular** es aquel que posee una autosimilitud exacta entre el triángulo de Sierpinski más pequeño que se genera con el fractal total. Mientras que un **fractal irregular** es aquel que tiene una autosimilitud estadística entre el triángulo más pequeño generado y el fractal total. A continuación veremos un ejemplo de fractal regular e irregular producido por la misma regla

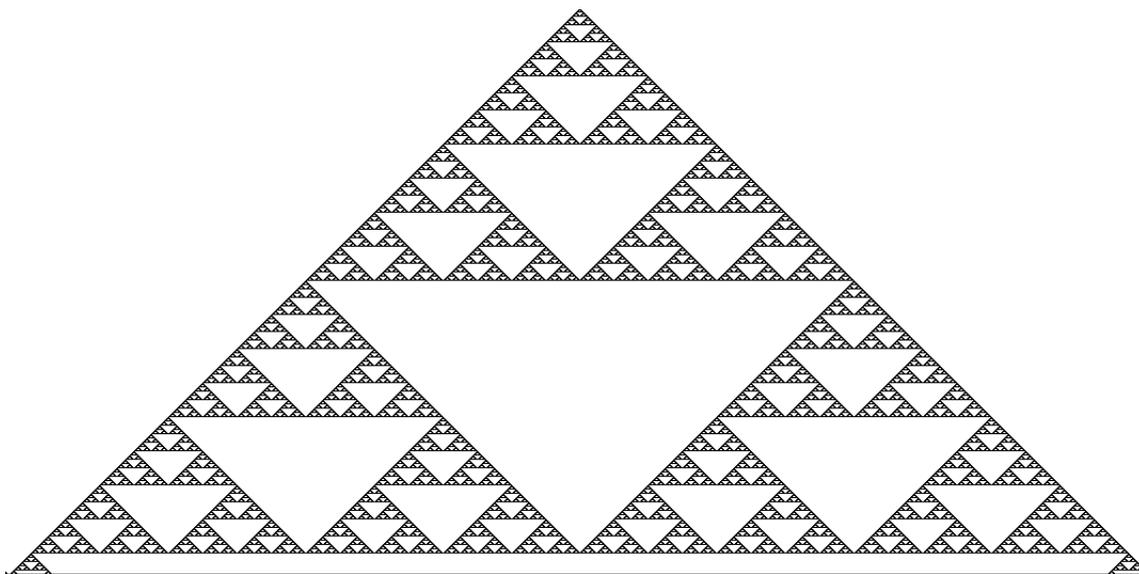


Figura 3.1: Fractal Regular, Regla 126 . Semilla 1

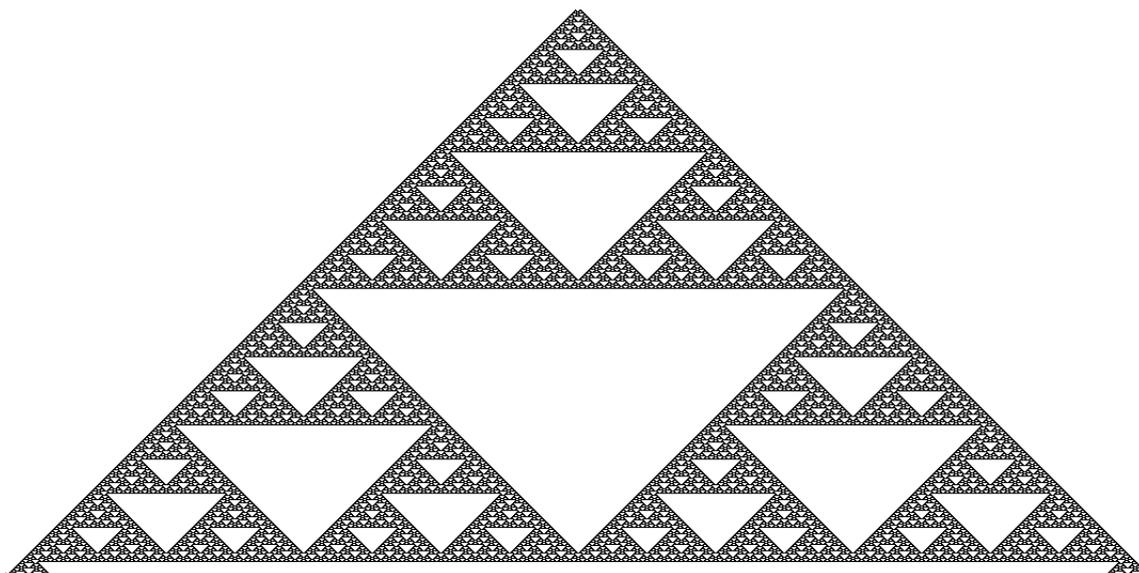


Figura 3.2: Fractal Irregular, Regla 126 . Semilla 0000100011

Fractales Regulares

El triángulo más pequeño en un fractal es un mosaico, el cual se repetirá durante la evolución de este. Consideramos que los mosaicos más pequeños son los de la longitud de semilla homogénea más corta con la que se pueden generar un fractal regular en cada regla, es decir las configuraciones $\{1, 11, 0, 00\}$.

Definición 3.2.1. Fractal regular en los autómatas celulares

Son los fractales donde se puede establecer un triángulo mínimo de recurrencia y también se puede establecer una ecuación de recurrencia que describe el número de triángulos producidos.

En el cuadro ?? mostramos los fractales regulares más pequeños que se pueden formar y en los siguientes apartados describiremos cada mosaico mediante la tabla de evolución necesaria para generar el triángulo, algunas iteraciones y las reglas que lo producen. En algunos reglas no son necesarias todas las vecindades para formar el mosaico más pequeño por lo que se dejan como variables estas vecindades. Al sustituir las variables por 0 o 1 se generan el conjunto de reglas que puede evolucionar ese mosaico.

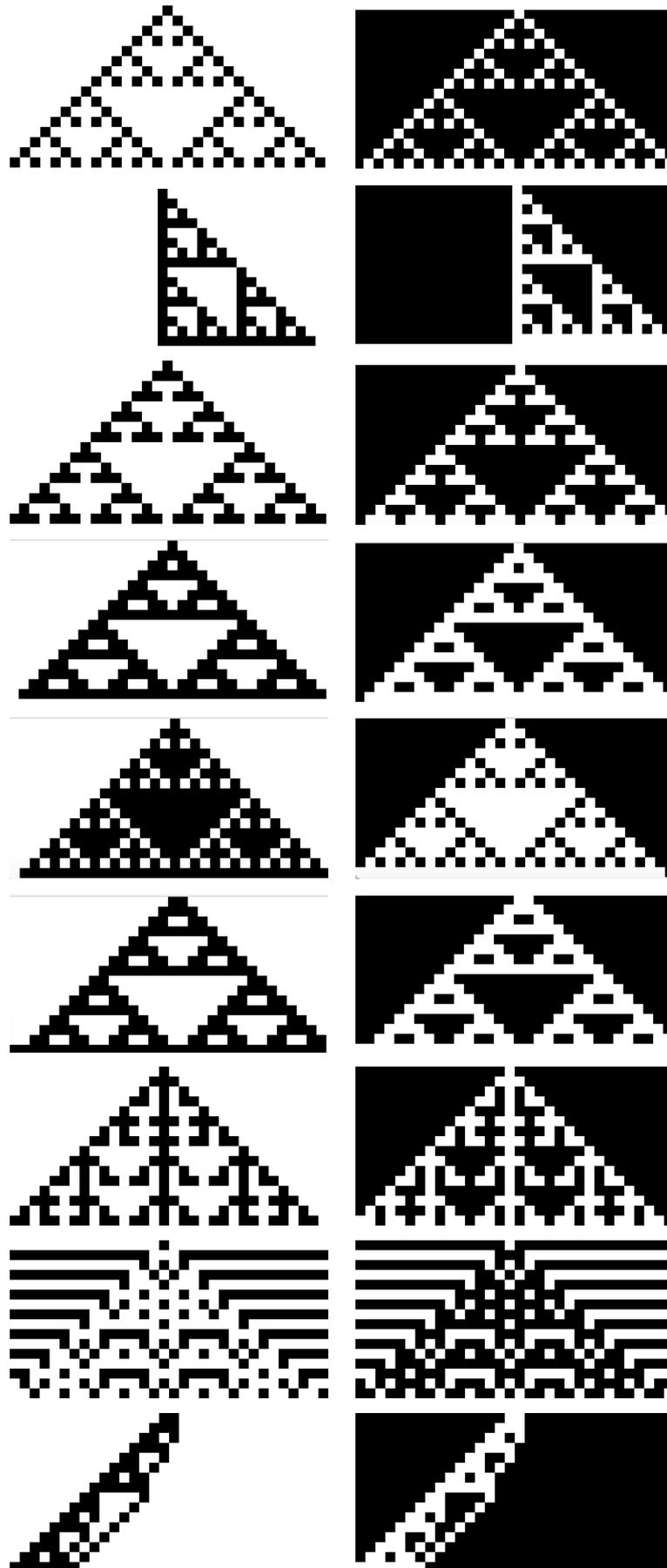


Figura 3.3: Lista de mosaicos regulares en automatas celulares elementales

Triángulo de Sierpinski

La ecuación de recurrencia está dada por $T(\frac{n}{2}) = 3T(n)$ donde la longitud de la base está dada por $n = 2^k$ y k es el numero de células de una configuración homogénea llamada semilla [33].

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
			■				
		■		■			
	■				■		
■		■		■		■	

Cuadro 3.2: Mosaico MS_1

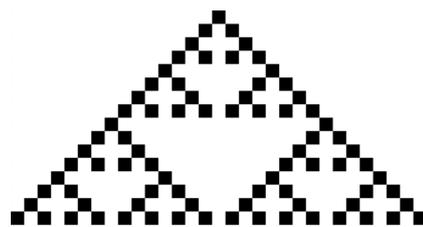


Figura 3.4: Primera iteración del Mosaico M1

Mosaicos MS_1 y $\overline{MS_1}$ La ecuación que describe a este mosaico es:

$$18 + 8a + 64b + 128c \tag{3.1}$$

Las reglas que cumplen esa ecuación son : 18,26,82,90,146,154,210, 218.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	a
1	0	0	1
1	0	1	0
1	1	0	b
1	1	1	c

Cuadro 3.3: Regla del mosaico MS_1

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
■	■	■	□	■	■	■	■
■	■	□	■	□	■	■	■
■	□	■	■	■	□	■	■
□	■	□	■	□	■	□	■

Cuadro 3.4: Mosaico $\overline{MS_1}$

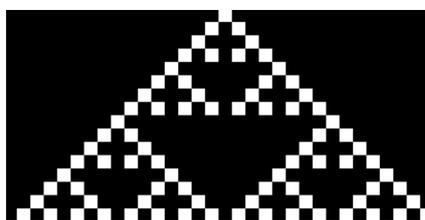


Figura 3.5: Primera iteración del Mosaico $\overline{MS_1}$

Mosaico $\overline{MS_1}$ La ecuación que describe a este mosaico es $164 + a + 2b + 16c$ y las reglas que cumplen esa ecuación son : 164,165,166, 167, 180, 181, 182, 183.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	a
0	0	1	b
0	1	0	1
0	1	1	0
1	0	0	c
1	0	1	1
1	1	0	0
1	1	1	1

Cuadro 3.5: Regla del mosaico \overline{MS}_1

Regla de Equivalencia		Left-right		Color		Both	
Regla	Mosaico	Regla	Mosaico	Regla	Mosaico	Regla	Mosaico
18	MS_1	18	MS_1	183	\overline{MS}_1	183	\overline{MS}_1
26	MS_1	82	MS_1	167	\overline{MS}_1	181	\overline{MS}_1
90	MS_1	90	MS_1	165	\overline{MS}_1	165	\overline{MS}_1
146	MS_1	146	MS_1	182	\overline{MS}_1	182	\overline{MS}_1
154	MS_1	210	MS_1	166	\overline{MS}_1	180	\overline{MS}_1
164	\overline{MS}_1	164	\overline{MS}_1	218	MS_1	218	MS_1

Cuadro 3.6: Reglas equivalentes mosaico MS_1

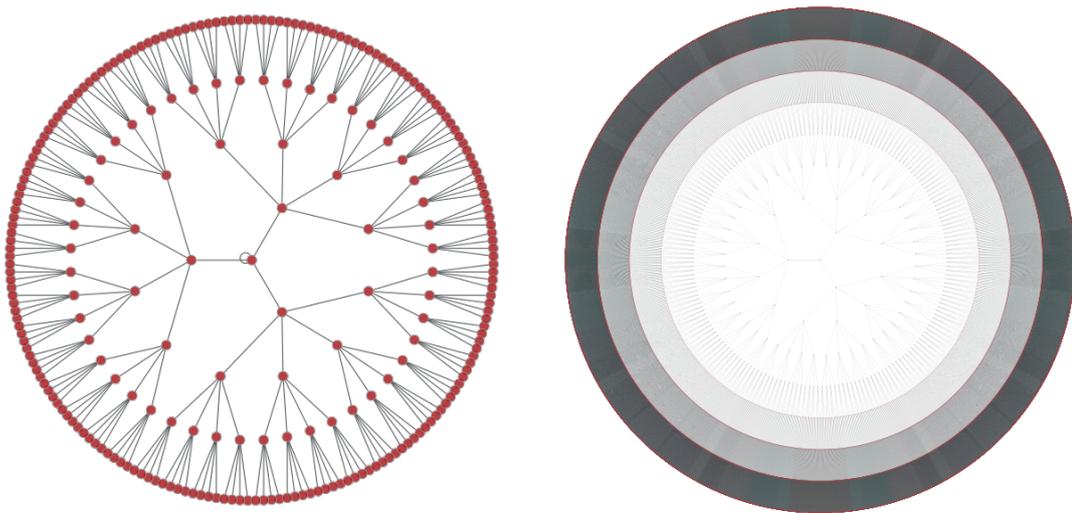


Figura 3.6: Atractor de la regla 90 y longitud 8
Atractor de la regla 90 y longitud 16

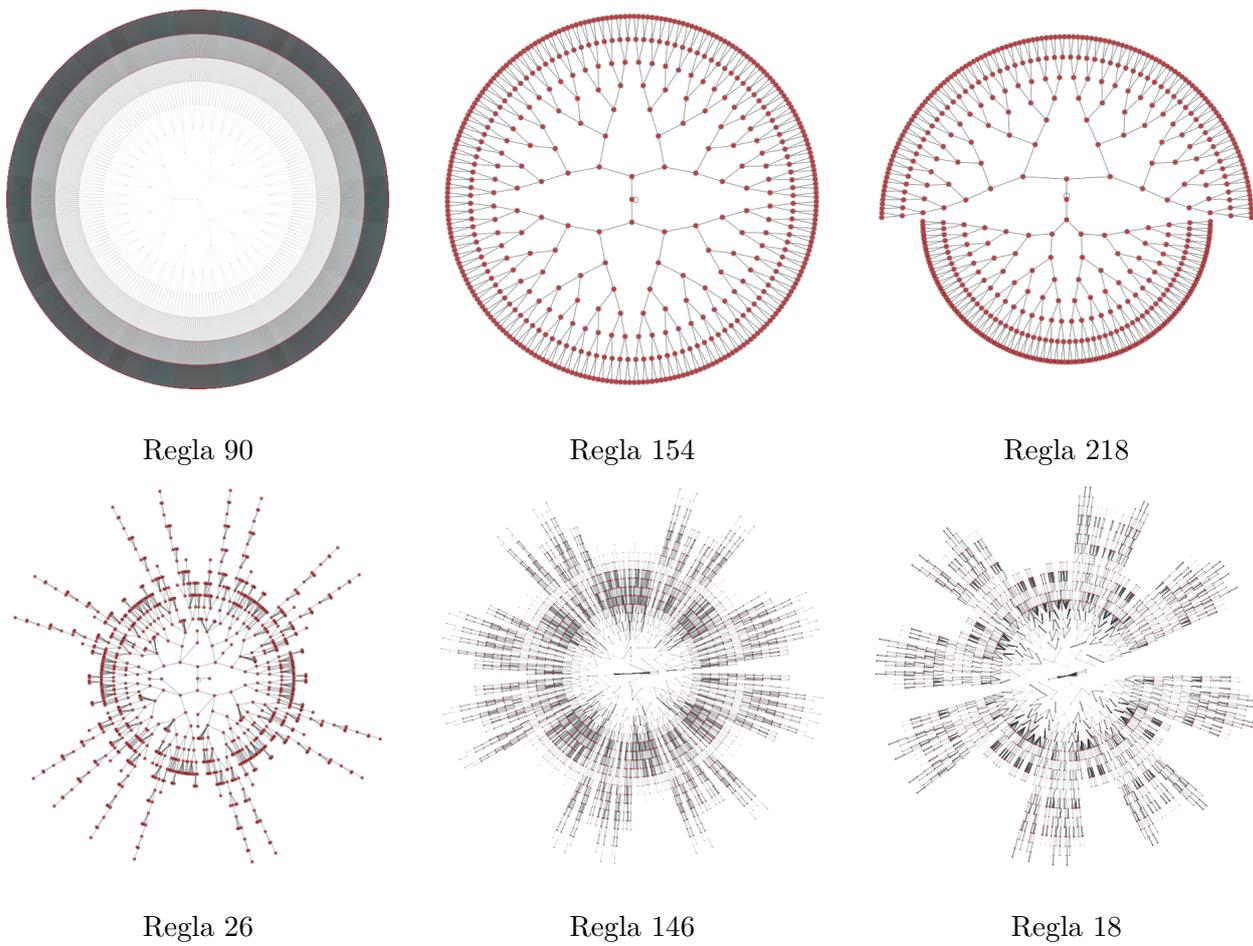
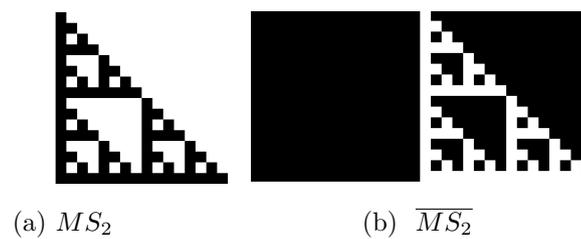


Figura 3.7: Atractores para el mosaico M1 y longitud 16.

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
			■				
			■	■			
			■	□	■		
			■	■	■	■	

Cuadro 3.7: Mosaico MS_2

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
■	■	■	□	■	■	■	■
■	■	■	□	□	■	■	■
■	■	■	□	■	□	■	■
■	■	■	□	□	□	□	■

Cuadro 3.8: Mosaico $\overline{MS_2}$ Figura 3.8: Primera iteración del Mosaico MS_2 y $\overline{MS_2}$

Mosaico MS_2 y $\overline{MS_2}$ El mosaico solo se puede producir ocupando todas las producciones, por lo que la ecuación que describe al mosaico MS_2 es 60 y el que produce el mosaico $\overline{MS_2}$ es 195

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Cuadro 3.9: Regla del mosaico MS_2

Regla de Equivalencia		Left-right		Color		Both	
Regla	Mosaico	Regla	Mosaico	Regla	Mosaico	Regla	Mosaico
60	MS_2	102	MS_2	195	$\overline{MS_2}$	153	$\overline{MS_2}$

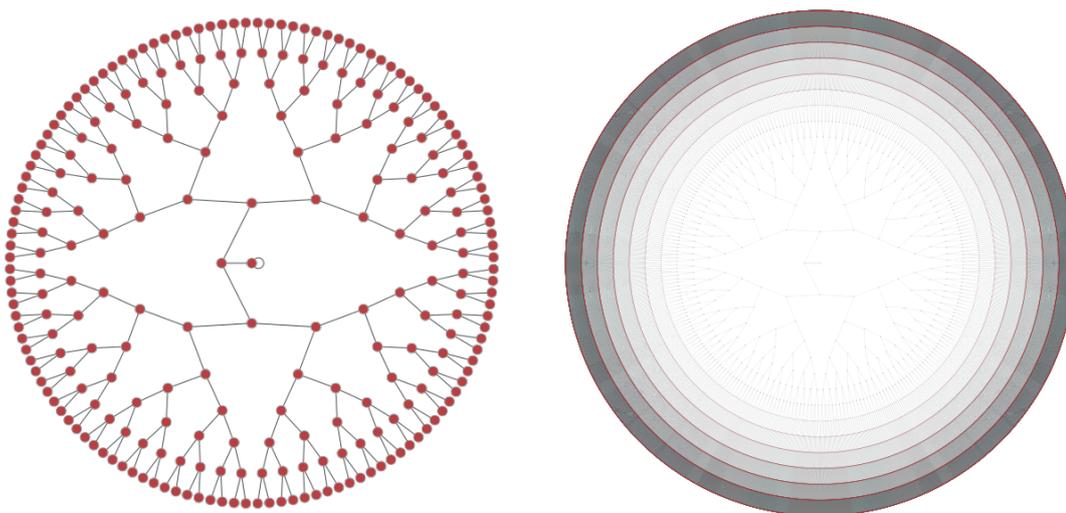


Figura 3.9: Atractor de la regla 60 y longitud 8
Atractor de la regla 60 y longitud 16

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
34							

Cuadro 3.10: Mosaico MS_3

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0

Cuadro 3.11: Mosaico \overline{MS}_3

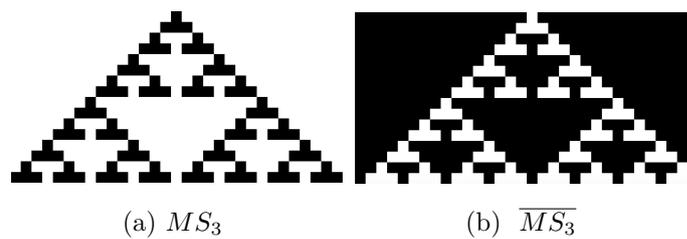


Figura 3.10: Primera iteración del Mosaico MS_3 y \overline{MS}_3

Mosaico MS_3 y \overline{MS}_3 El mosaico solo se puede producir ocupando todas las producciones, por lo que la ecuación que describe al mosaico MS_3 es 22 y el que produce el mosaico \overline{MS}_3 es 151

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Cuadro 3.12: Regla del mosaico MS_3

Regla de Equivalencia		Left-right		Color		Both	
Regla	Mosaico	Regla	Mosaico	Regla	Mosaico	Regla	Mosaico
22	MS_3	22	MS_3	151	$\overline{MS_3}$	151	$\overline{MS_2}$

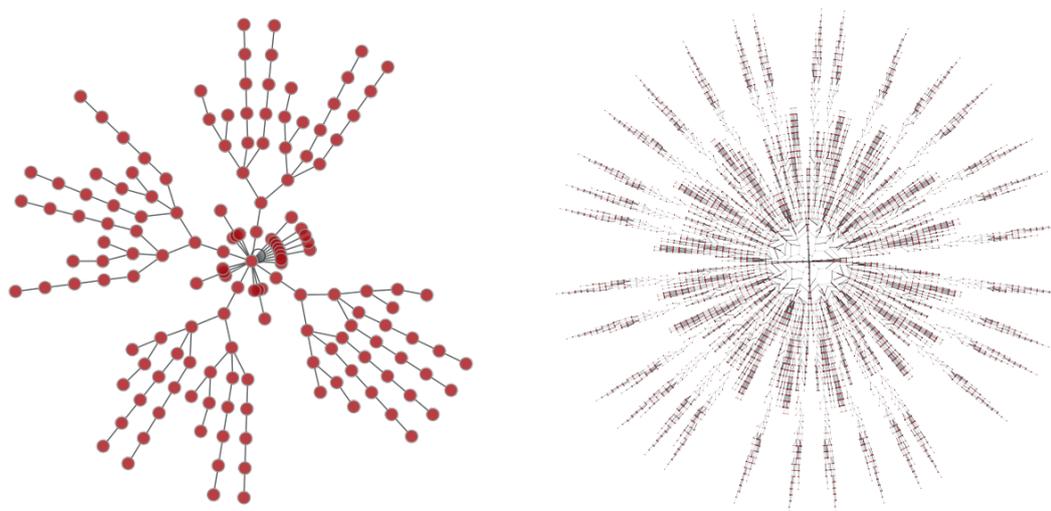


Figura 3.11: Atractor de la regla 22 y longitud 8
Atractor de la regla 22 y longitud 16

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
			■				
		■	■	■			
	■	■	■	■	■		
■	■	■	■	■	■	■	

Cuadro 3.13: Mosaico MS_4

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
■	■	■	■	■	■	■	■
■	■	■	■	■	■	■	■
■	■	■	■	■	■	■	■
■	■	■	■	■	■	■	■

Cuadro 3.14: Mosaico $\overline{MS_4}$

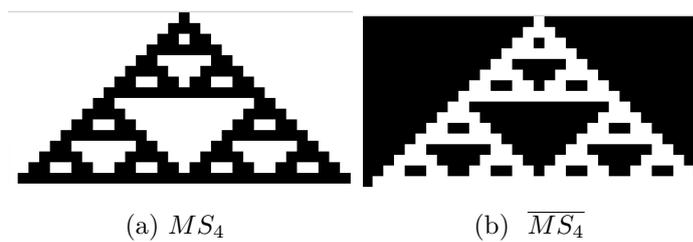


Figura 3.12: Primera iteración del Mosaico MS_4 y $\overline{MS_4}$

Mosaico MS_4 y $\overline{MS_4}$ El mosaico solo se puede producir ocupando todas las producciones, por lo que la ecuación que describe al mosaico MS_4 es 126 y el que produce el mosaico $\overline{MS_4}$ es 129

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Cuadro 3.15: Regla del mosaico MS_4

Regla de Equivalencia		Left-right		Color		Both	
Regla	Mosaico	Regla	Mosaico	Regla	Mosaico	Regla	Mosaico
126	MS_4	126	MS_4	129	$\overline{MS_4}$	129	$\overline{MS_4}$

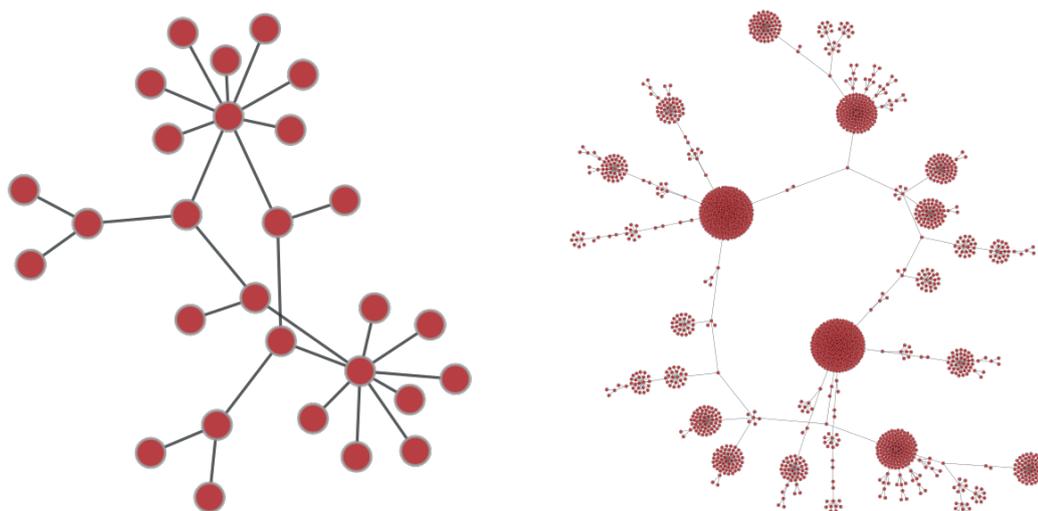
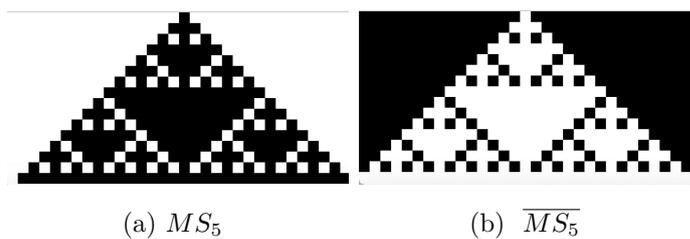


Figura 3.13: Atractor de la regla 126 y longitud 8
 Atractor de la regla 126 y longitud 16

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
			■				
		■	■	■			
	■		■		■		
■	■	■	■	■	■	■	

Cuadro 3.16: Mosaico MS_5

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
■	■	■		■	■	■	■
■	■			■	■	■	■
■		■		■		■	■
							■

Cuadro 3.17: Mosaico \overline{MS}_5 Figura 3.14: Primera iteración del Mosaico MS_5 y \overline{MS}_5

Mosaico MS_5 y \overline{MS}_5 El mosaico solo se puede producir ocupando todas las producciones, por lo que la ecuación que describe al mosaico MS_5 es 182 y el que produce el mosaico \overline{MS}_5 es 146

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Cuadro 3.18: Regla del mosaico MS_5

Regla de Equivalencia		Left-right		Color		Both	
Regla	Mosaico	Regla	Mosaico	Regla	Mosaico	Regla	Mosaico
126	MS_5	126	MS_5	129	$\overline{MS_5}$	129	$\overline{MS_5}$

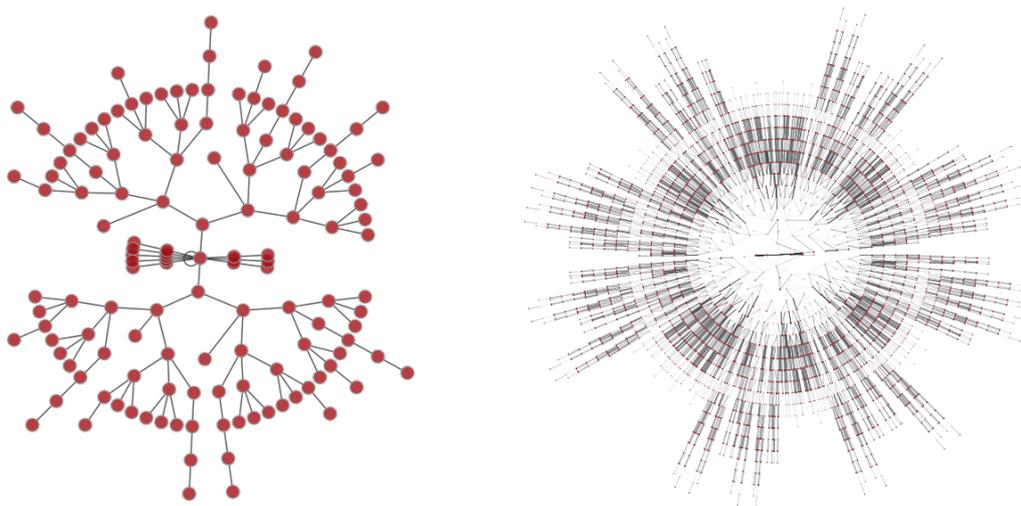


Figura 3.15: Atractor de la regla 146 y longitud 8
Atractor de la regla 146 y longitud 16

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
			■	■			
		■	■	■	■		
	■	■			■	■	
■	■	■	■	■	■	■	■

Cuadro 3.19: Mosaico MS_6

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
■	■	■			■	■	■
■	■						■
			■	■			

Cuadro 3.20: Mosaico \overline{MS}_6

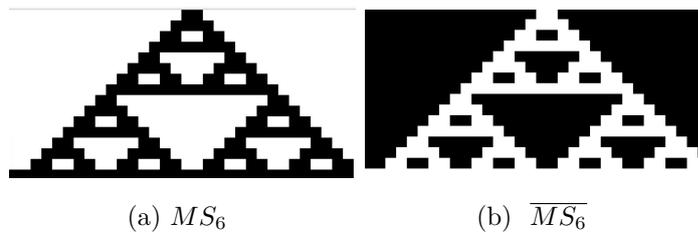


Figura 3.16: Primera iteración del Mosaico MS_6 y \overline{MS}_6

Mosaico MS_6 y \overline{MS}_6 El mosaico solo se puede producir ocupando todas las producciones, por lo que la ecuación que describe al mosaico MS_6 es:

$$90 + 4a + 32b \tag{3.2}$$

el conjunto: 90, 94, 122, 126 y el que produce el mosaico \overline{MS}_6 es : 129, 133, 161, 165.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	a
0	1	1	1
1	0	0	1
1	0	1	b
1	1	0	1
1	1	1	0

Cuadro 3.21: Regla del mosaico MS_6

Regla de Equivalencia		Left-right		Color		Both	
Regla	Mosaico	Regla	Mosaico	Regla	Mosaico	Regla	Mosaico
90	MS_6	90	MS_6	165	$\overline{MS_6}$	165	$\overline{MS_6}$
94	MS_6	94	MS_6	133	$\overline{MS_6}$	133	$\overline{MS_6}$
122	MS_6	122	MS_6	161	$\overline{MS_6}$	161	$\overline{MS_6}$
126	MS_6	126	MS_6	129	$\overline{MS_6}$	129	$\overline{MS_6}$

Triangulo Sierpinski Áureo

La ecuación de recurrencia está dada por $T(n) = 2T(2^{k-1}) + 4T(2^{k-2})$ donde la longitud de la base está dada por $n = 2^k$ y k es el numero de células de una configuración homogénea llamada semilla [33].

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0

Cuadro 3.22: Mosaico MA_1

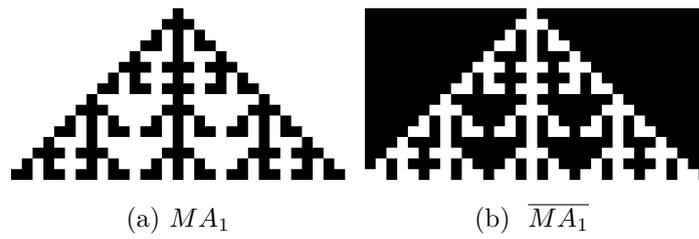


Figura 3.17: Primera iteración del Mosaico MA_1 y $\overline{MA_1}$

Mosaicos MA_1 La reglas que forma este mosaico es :150.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Cuadro 3.23: Regla del mosaico MA_1

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0

Cuadro 3.24: Mosaico MA_2

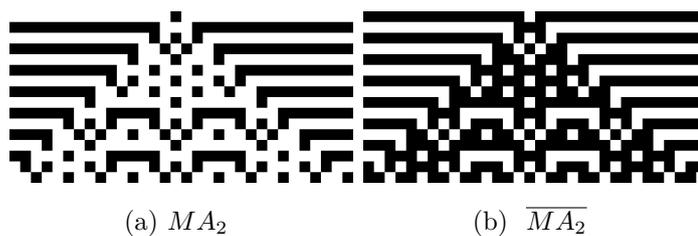


Figura 3.18: Primera iteración del Mosaico MA_2 y $\overline{MA_2}$

Mosaicos MA_2 La reglas que forma este mosaico es : 105.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Cuadro 3.25: Regla del mosaico MA_2

Triangulo Sierpinski con Corrimiento

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
				■	■		
			■	■	■		
		■	■		■		
	■	■	■	■			

Cuadro 3.26: Mosaico MA_1



Figura 3.19: Primera iteración del Mosaico MC_1 y $\overline{MC_1}$

Mosaicos MC_1 La reglas que forma este mosaico es :150.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Cuadro 3.27: Regla del mosaico MC_1

3.2.2. Morfología de las reglas

Regla equivalente	MS_1	MS_2	MS_3	MS_4	MS_5	MS_6	MA_1	MA_2	MC_1	Total
18	•									1
22			•							1
26	•									1
60		•								1
90	•					•				2
94						•				1
105								•		1
106									•	1
122						•				1
126				•		•				2
146	•				•					2
150							•			1
154	•									1
164	•									1
Total	6	1	1	1	1	4	1	1	1	

Cuadro 3.28: Reglas fractales

3.3. Caracterización Fenotípica

3.3.1. Exponentes de Lyapunov

Con el fin de hacer un exhaustivo análisis del comportamiento de un Autómata Celular Elemental (ECA), han sido propuestas diferentes medidas, tales como los Exponentes de Lyapunov. En las últimas dos décadas han surgido dos puntos de vista con respecto a los Exponentes de Lyapunov. Una de ellas se remonta a una sugerencia de Wolfram, que posteriormente Shereshevski formalizó y fue adoptada por diferentes autores, la cual establece a los Exponentes de Lyapunov como las velocidades con las cuales el daño se propaga hacia la izquierda y hacia la derecha. Consecuentemente, cada Autómata Celular Elemental está caracterizado por medio de dos exponentes, los cuales pueden diferir. Por otra parte, la definición presentada por Bagnoli, involucra sólo un exponente para cada autómata, el cuál cuantifica la velocidad exponencial por la cual se acumulan los defectos si un ECA evoluciona con una configuración inicial con un solo defecto sin tomar en cuenta la dirección en la cual se propagan dichos defectos. Jan M. Baetens y Janko Gravner proponen otra definición, la cuál, al unir las dos definiciones permite realizar un análisis exhaustivo de la estabilidad de un ECA. El exponente de Lyapunov de acuerdo a Bagnoli se calcula rastreando todos los defectos que surgen de introducir un solo defecto en la condición inicial del ECA, es decir, cambiando el estado de una de las células, de la siguiente manera:

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \left(\frac{\epsilon_t}{\epsilon_0} \right)$$

Donde ϵ_t denota el número total de defectos en el t -ésimo paso. Al introducir un defecto en una de las células de la configuración inicial, se propagará y afectará cada vez a más células debido a la dinámica de los ECAs. El valor ϵ_t se calcula evolucionando la configuración inicial s_0 y su contra-parte perturbada s^*_0 para lo cual se cumple que $d(s_0, s^*_0) = 1$, para un paso de tiempo, después de lo cual una replica $s^*_i(\cdot, 1)$ de $s(\cdot, 1)$ es creada para cada célula c_i para lo cual se cumple que $s(c_i, 1) \neq s^*(c_i, 1)$ de tal manera que

$d(s(c_i, 1), s^*(c_i, 1)) = 1$. Entonces, todas las replicas evolucionan un paso de tiempo mas y la configuración resultante $s^*_i(\cdot, 2)$ son comparadas con $s(\cdot, 2)$, de esta manera un nuevo grupo de replicas puede ser construido, este proceso se repite hasta que la evolución del ECA se detiene. Baetens y Gravner proponen redefinir ϵ_t como un vector ϵ_t el cuál el i -ésimo elemento ϵ_t^i es el numero de defectos en la célula c_i en el paso de tiempo dado, de esta manera se tiene una construcción que incorpora tanto cantidad como posición de los defectos. Bagnoli también define este vector de daño, pero descarta la información del contenido ya que su análisis posterior se basa en la suma de sus elementos. El resultado se conoce como Máximo Exponente de Lyapunov (MLE) y por analogía el MLE en el vector ϵ_t es la posición donde se acumula el mayor numero de defectos.

3.3.2. Perfiles de Lyapunov

El perfil de Lyapunov de un ECA se define como:

$$\Lambda = \frac{1}{T} \ln(\epsilon_t)$$

Donde \ln se aplica a cada uno de los elementos. Por lo tanto el Perfil de Lyapunov consiste en las velocidades de propagación promediadas en el tiempo en cada una de las direcciones posibles. Dado que la propagación de los defectos esta limitada por el tamaño de la vecindad $N(c_i)$, debe existir un límite superior en los elementos de Λ_T . Este esta dado por la $T+1$ -ésima fila del triangulo trinomial de tal manera que:

$$\epsilon_t^i = \binom{T}{i - i^*_0}_2$$

En donde i^*_t denota el índice de la célula que fue perturbada en $t = 0$.

3.3.3. Entropía

En termodinámica la Entropía es una medida para indicar el calor disipado por un cuerpo. Cuando un cuerpo libera mayor cantidad de energía calorífica es que las moléculas que lo componen se mueven a mayor velocidad chocando unas con otras y en cada choque de moléculas se libera alguna cantidad de energía en forma de calor. Por lo tanto una alta medida de entropía significa más calor liberado por un cuerpo. En autómatas celulares se adoptó este termino para indicar el grado de desorden en que las configuraciones de alguna longitud específica pueda ocurrir, el grado de desorden en que las configuraciones globales aparecen o la variación de valores de estados que en una sola célula presenta a lo largo de un número determinado de pasos de tiempo.

Entropía espacial

Para conocer el grado de caos en una fase de la evolución del autómata, definimos la entropía espacial como:

$$h^{(x)}(X) = \frac{1}{X} \log_k \left(\sum_{j=1}^{K^x} \theta(p_j^{(x)}) \right)$$

Donde el superíndice (x) indica que se trata de secuencias espaciales de células y X indica el tamaño de las secuencias, p_j indica la probabilidad de que ocurra la secuencia en turno y $\theta(p)$ es una función $\theta : [0, 1] \rightarrow \{0, 1\}$ definida como:

$$\theta(p) = 0 \quad \text{si} \quad p = 0 \quad \text{y} \quad \theta(p) = 1 \quad \text{en} \quad \text{otro caso}$$

Esta función $\theta(p)$ cuenta el orden exponencial de todas las secuencias que es posible que ocurran.

A esta entropía se le conoce también como Entropía Topológica y va acompañado de una medida de entropía que se conoce como métrica para la entropía y se define de la siguiente

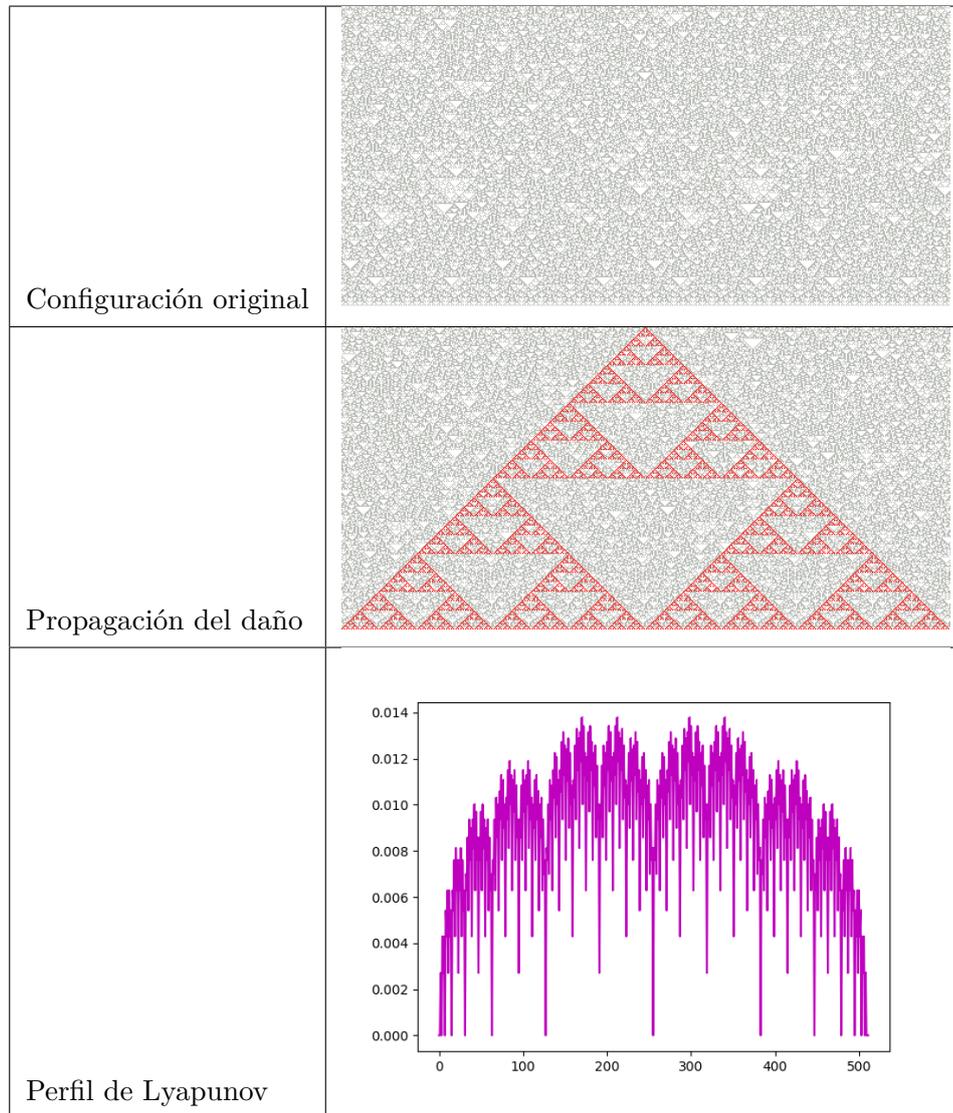
manera:

$$h_{\mu}^{(x)}(X) = -\frac{1}{X} \sum_{j=1}^{k^x} p_j^{(x)} \log_k p_j^{(x)}$$

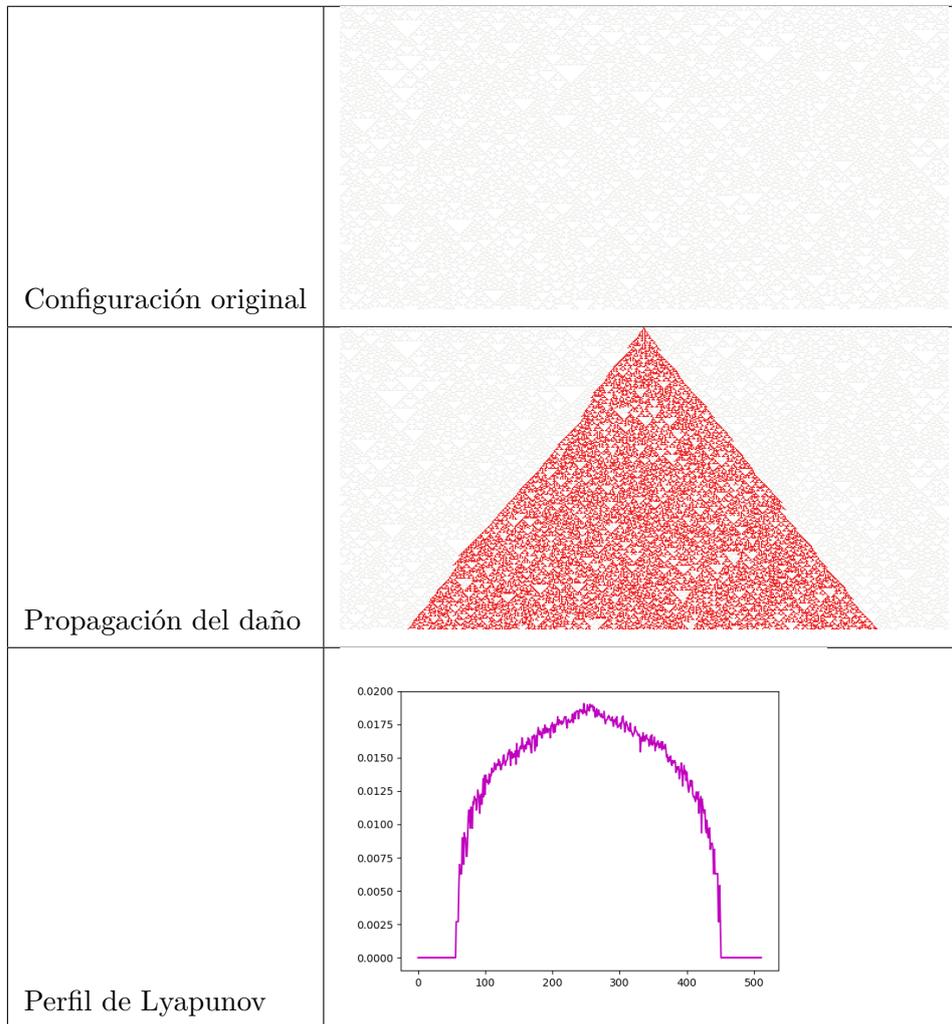
En la entropía espacial de conjunto se observa, en promedio, cuál es el orden exponencial de las ocurrencias de bloques de longitud X . En el caso de la medida de entropía, lo observado es el promedio de la razón exponencial de que un bloque de longitud fija X aparezca en cualquier parte de la evolución de un autómata celular en un tiempo fijo t .

Hemos de notar que el valor máximo de entropía se logra cuando todos los posibles K^X bloques de longitud X tiene igual probabilidad de ocurrir. El máximo valor de entropía es 1.

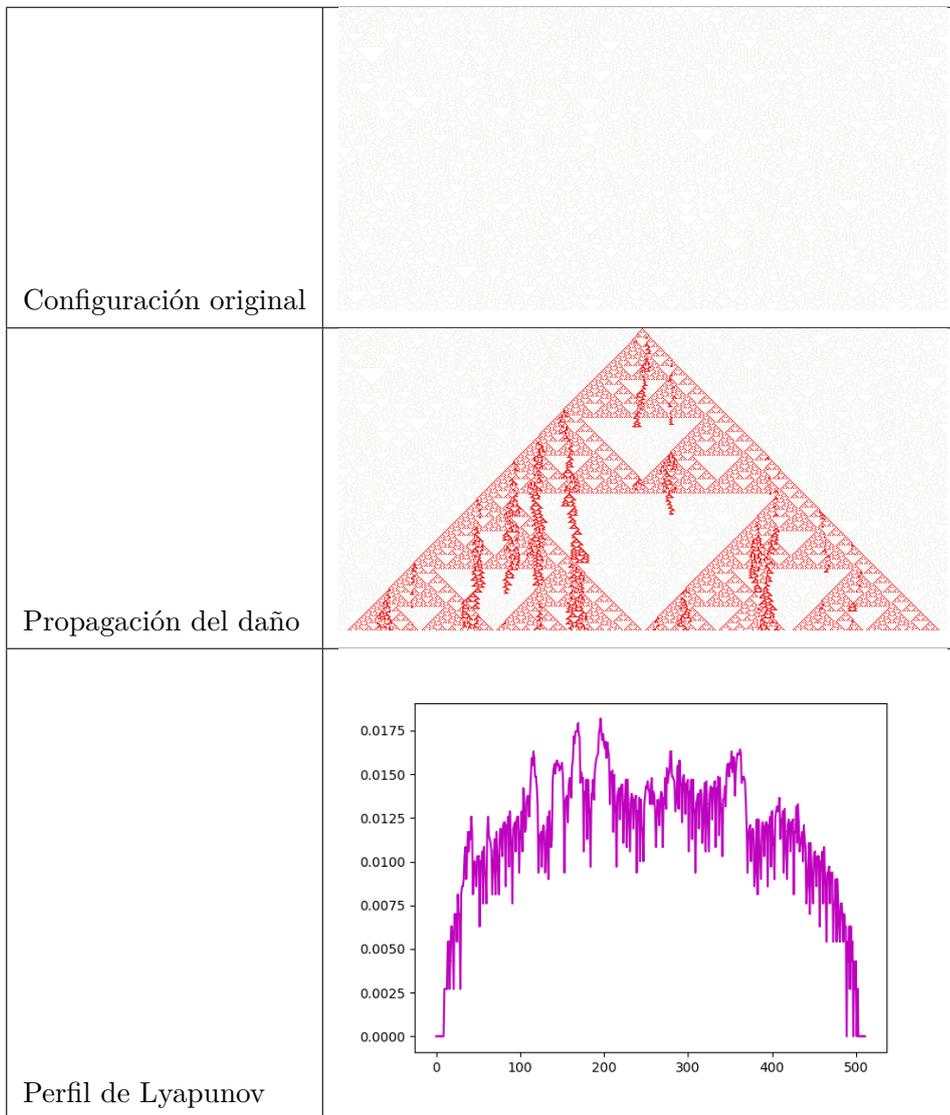
Otro aspecto a tomar en cuenta es que las secuencias que son "Jardines de edén" solo ocurren una vez y jamás podrán aparecer de nuevo como imagen de una configuración global.

E1: Fractales

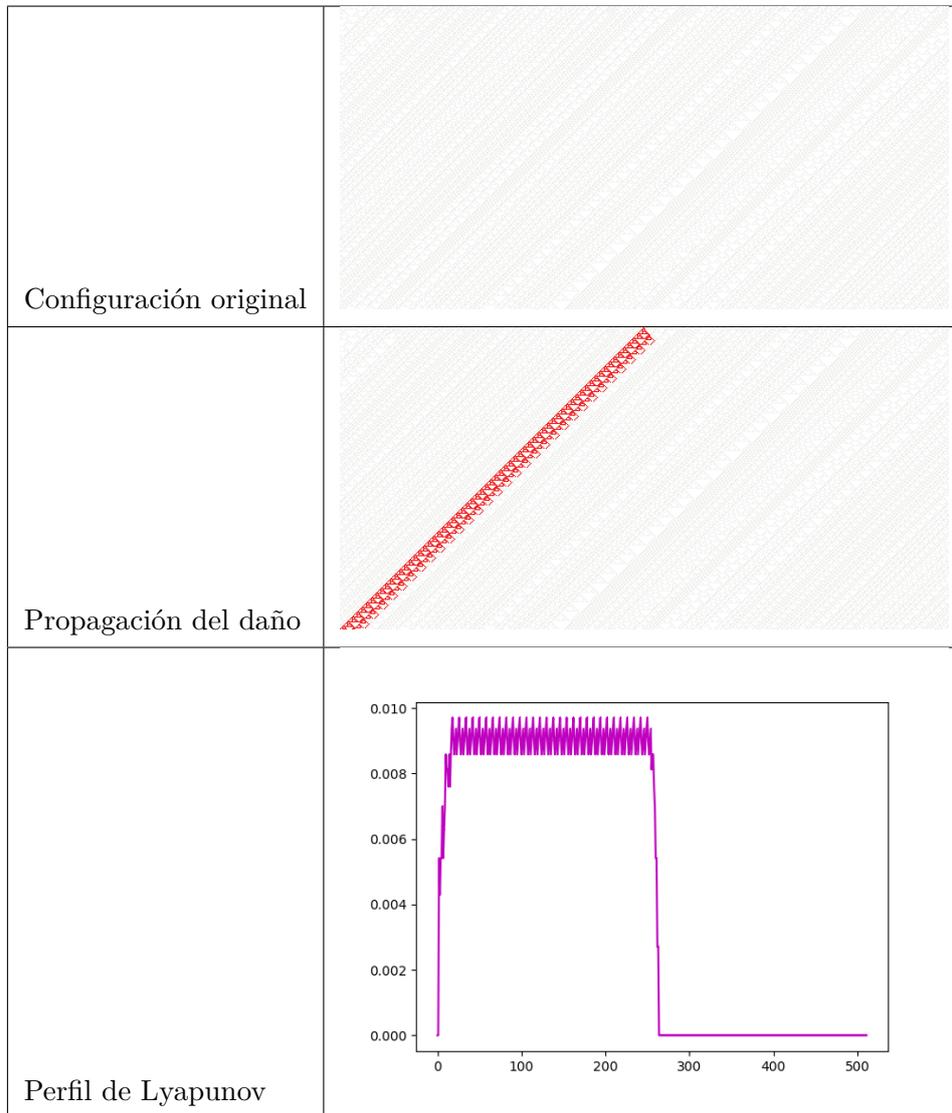
Al introducir un defecto en la configuración inicial y evolucionar podemos notar patrones fractales en la propagación del daño (células en rojo). Las reglas 60, 90, 105, 150 presentan este comportamiento, los defectos se propagan por toda la vecindad y forman estos patrones. En el caso de la regla 60, la propagación aparece por el lado derecho.

E2: Caos

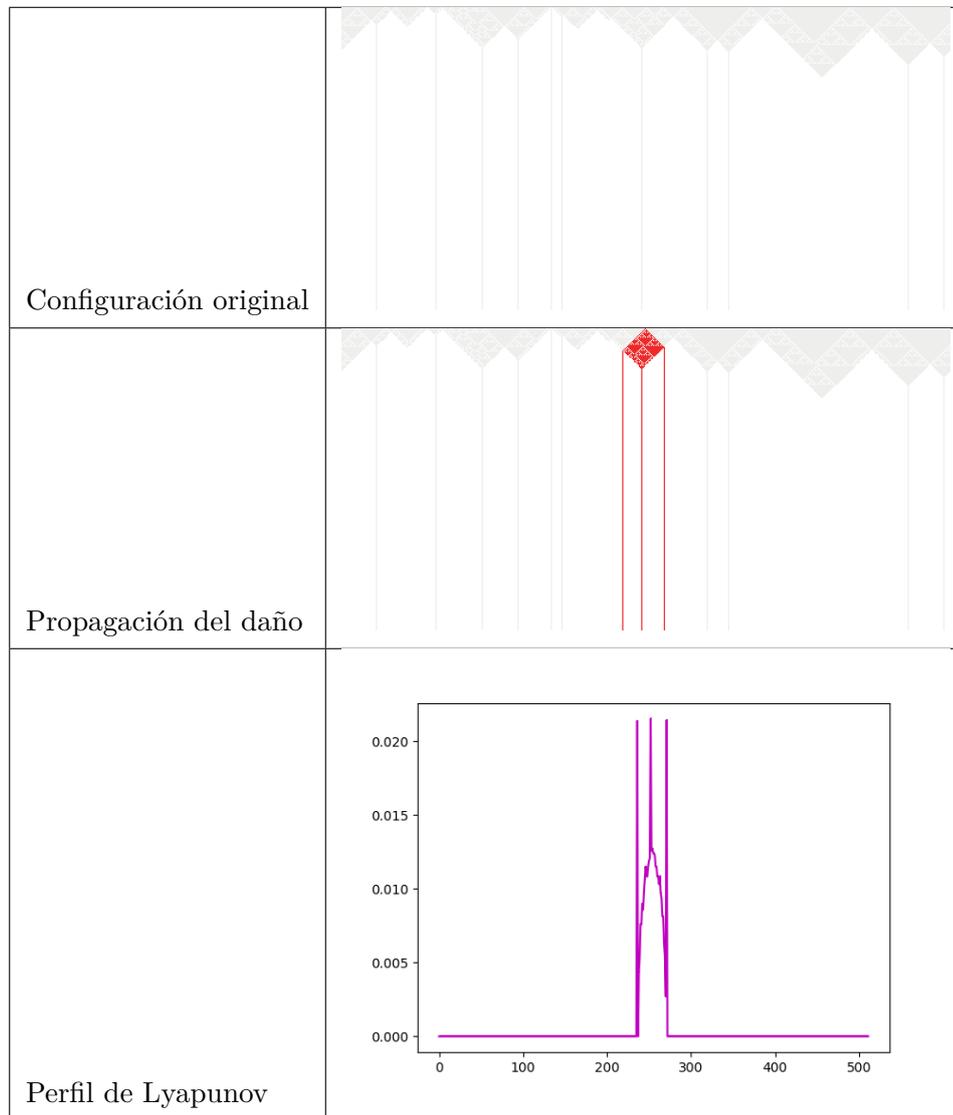
Al introducir un defecto en la configuración inicial y evolucionar podemos notar caos en la propagación del daño (células en rojo). Las reglas 22, 106, 122, 126 presentan este comportamiento, el volumen de defectos es mayor pero no se propagan por toda la vecindad. En el caso de la regla 106 la propagación de los defectos es por la izquierda.

E3: Caos y Fractales

Al introducir un defecto en la configuración inicial y evolucionar podemos notar algunas regiones caóticas sobre fractales en la propagación del daño (células en rojo). Las reglas 18 y 146 presentan este comportamiento.

E4: Estructuras Periódicas

Al introducir un defecto en la configuración inicial y evolucionar podemos notar patrones periódicos en la propagación del daño (células en rojo). Las reglas 26 y 154 presentan este comportamiento, la propagación de los defectos en ambas reglas es por la izquierda.

E5: Estado homogéneo

Al introducir un defecto en la configuración inicial y evolucionar podemos notar que la propagación de daño (células en rojo) evoluciona a un estado homogéneo. Las reglas 94 y 164 presentan este comportamiento, para conseguir generar fractales con la regla 164 necesitamos una configuración inicial con densidad alta (células en estado 1) y con la regla 94 una semilla de más de al menos 5 células.

Regla equivalente	Exponente I	Exponente II	Exponente III	Exponente IV	Exponente V
18			•		
22		•			
26				•	
60	•				
90	•				
94					•
105	•				
106		•			
122		•			
126		•			
146			•		
150	•				
154				•	
164					•

Cuadro 3.29: Comparación de las reglas fractales y la caracterización por los exponentes de Lyapunov

3.4. Caracterización genotípica

3.4.1. Teoría del campo promedio

La teoría del campo promedio proporciona la probabilidad de que las células tengan cada valor del espacio de evoluciones en un tiempo $t + 1$. La probabilidad de ocurrencia de un estado es la suma de las probabilidades de las vecindades que se transforman a éste en un tiempo t anterior. La probabilidad de que una célula tenga el valor de 1 está dado por la ecuación

$$p_t = \sum_{j=1}^{k^{2r+1}} \varphi_j(X) p_{t+1}^v (1-p)^{n-v} \quad (3.3)$$

Donde φ es la regla de evolución de un autómata de orden (k, r) y $\Sigma = \{0, 1\}$. X tomará el valor de la transformación de la regla de evolución de cada vecindad. [12]

Por ejemplo la curva del campo promedio para la regla 90 está dada a partir de sus evoluciones

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	1

$$p_t = \sum_{j=1}^{k^{2r+1}} \varphi_j(X) p_{t+1}^v (1-p)^{n-v} = pq^2 + p^2q + pq^2 + p^2q = 2p^2q + 2pq^2$$

Al tener el polinomio podemos calcular la probabilidad de las siguientes t generaciones al componer el polinomio consigo mismo t veces.

$$p_{t+1} = p_t(p_t)$$

Pero observamos que llega a los puntos de equilibrio en pocas iteraciones. Los puntos

de equilibrio son los puntos de intersección del polinomio con la función identidad en el intervalo $(0, 1)$ [12][21].

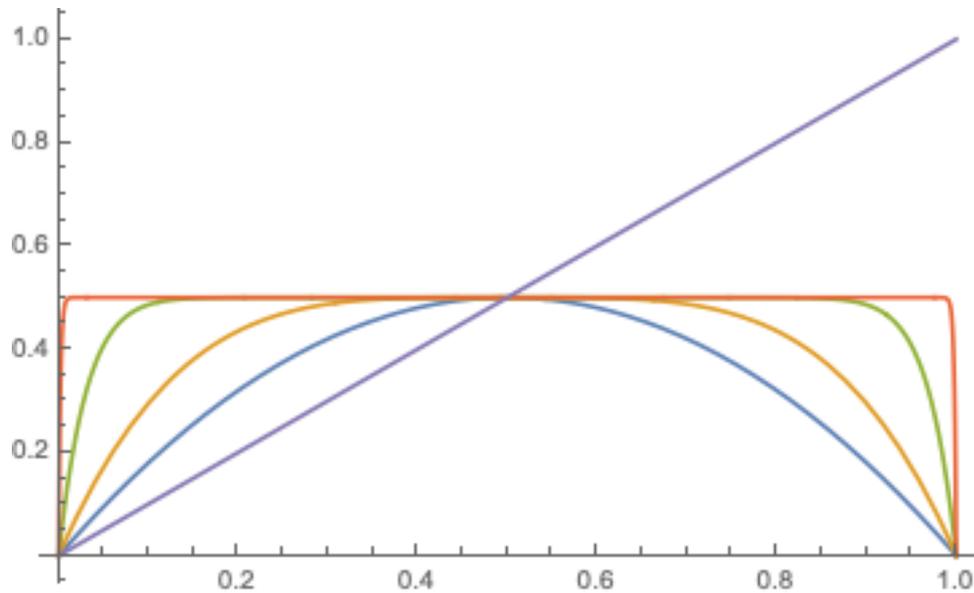


Figura 3.20: Polinomio del campo promedio de la regla 90, iterado tres veces

Fondo periódico Consideramos las características morfológicas de la curva del campo promedio además del espacio de evoluciones.

La caracterización morfológica la consideramos a partir de la definición de tres regiones de la curva.

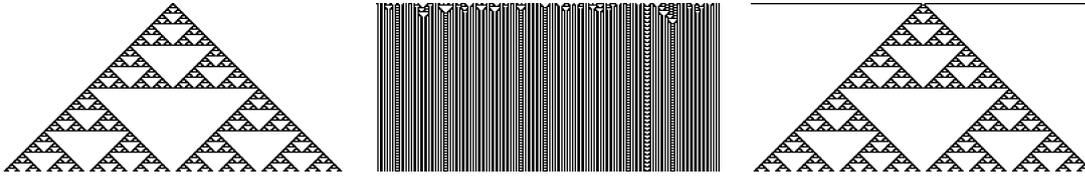
- La primera región son las configuraciones que tienen poca densidad de celdas con estado uno.
- La segunda es un intervalo de estabilización de las configuraciones, en esta región la probabilidad de unos está dada por el punto de equilibrio.
- La última región son configuraciones donde la densidad de células con número uno es mayor.

Encontramos tres comportamientos

1. En las dos regiones de los extremos generan fractales que evolucionan en un fondo

blanco.

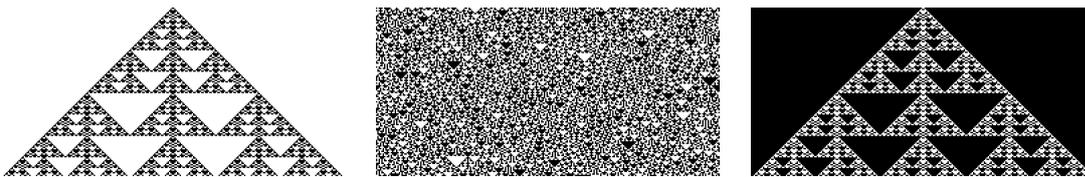
Las reglas son: 18, 22, 26, 60, 90, 94, 122, 126.



Cuadro 3.30: Característica 1

2. En la primera región se generan fractales en un fondo blanco y en la tercera región se generan fractales en un fondo negro.

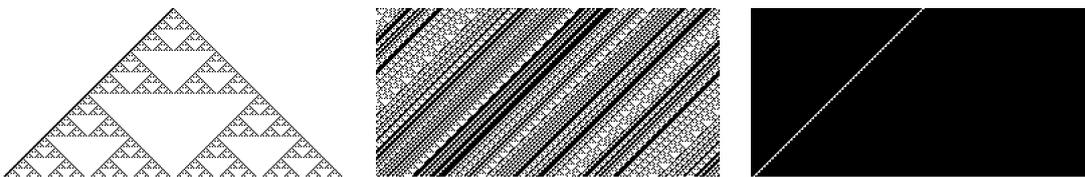
Las reglas son: 105, 146, 150.



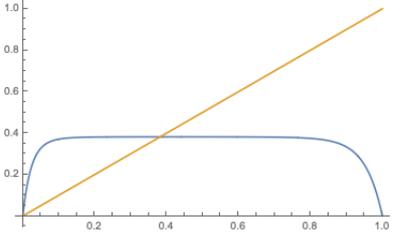
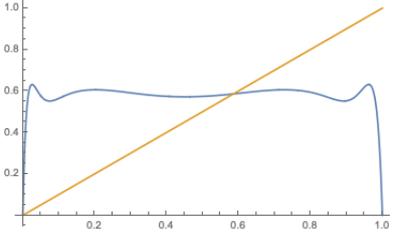
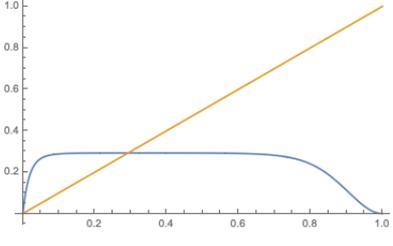
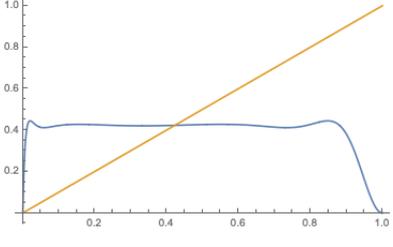
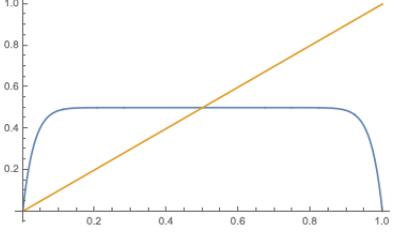
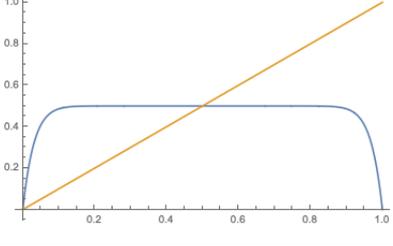
Cuadro 3.31: Característica II

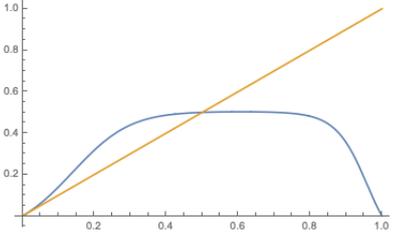
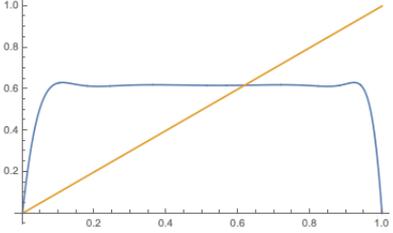
3. En la primera región se generan fractales en un fondo blanco y en la tercera región se generan fondos periódicos.

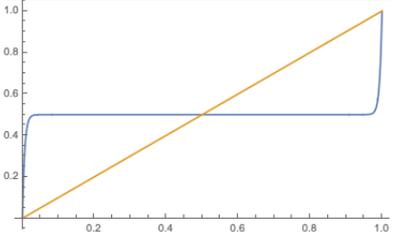
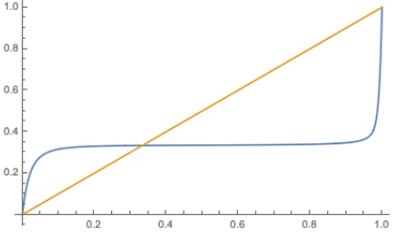
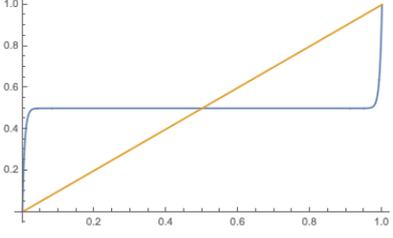
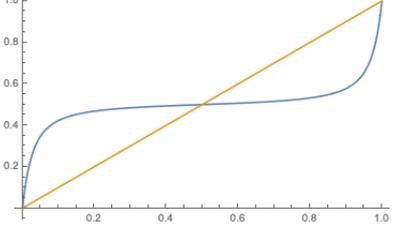
Las reglas son: 154, 218

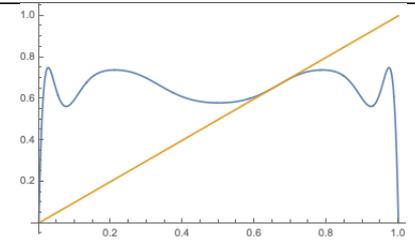


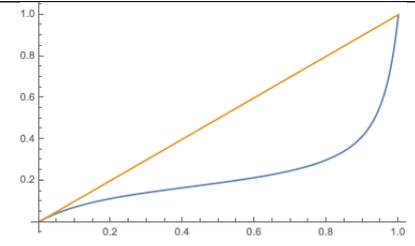
Cuadro 3.32: Característica III

Regla	Polinomio	Punto fijo	Derivada	Curva
26	$p^2q + 2pq^2$	0,381966	0,000453	
94	$2p^2q + 3pq^2$	0,585786	0,186155	
18	$2pq^2$	0,292893	0,000866	
22	$3pq^2$	0,42265	0,046392	
60	$2p^2q + 2pq^2$	0,5	0	
90	$2p^2q + 2pq^2$	0,5	0	

Regla	Polinomio	Punto fijo	Derivada	Curva
106	$3p^2q + pq^2$	0,5	0,0625	
122	$3p^2q + 2pq^2$	0,618034	0,021345	

Regla	Polinomio	Punto fijo	Derivada	Curva
105	$3p^2q + q^3$	0,5	0	
146	$p^3 + 2pq^2$	0,333333	0,012345	
150	$p^3 + 3pq^2$	0,5	0	
154	$p^3 + p^2q + 2pq^2$	0,5	0,0625	

Regla	Polinomio	Punto fijo	Derivada	Curva
126	$3p^2q + 3pq^2$	0,666667	0,999999	

Regla	Polinomio	Punto fijo	Derivada	Curva
164	$p^3 + p^2q + pq^2$	0	0	

Punto fijo Para agrupar las reglas se evoluciono cada una con una probabilidad $p = p_t[t = p_f]$, donde p_t es la probabilidad obtenida por el polinomio del campo promedio de la regla y p_f es el punto fijo del polinomio y las agrupamos por el polinomio que representa la regla.

Regla equivalente	Gráfica I	Gráfica II	Gráfica III	Gráfica IV
18	•			
22	•			
26	•			
60	•			
90	•			
94	•			
105		•		
106	•			
122	•			
126			•	
146		•		
150		•		
154	•			
164				•

Cuadro 3.37: Caracterización de las Reglas fractales de acuerdo a la teoría del campo promedio

3.4.2. Cuenca de Atractores

Como planteamos en el segundo capítulo, los sistemas dinámicos pueden describirse a partir de las orbitas que producen. Algunos de los estudios más extensos se han realizado en [36] [23]. Es necesario precisar que :

- Un sistema puede tener varias cuencas de atractores.
- En los autómatas celulares elementales con condiciones de frontera siempre se llega a un atractor de periodo $N \geq 1$
- Una cuenca de atractores puede tener un subconjunto de configuraciones que no pueden ser producidas por la regla del autómata. Estas configuraciones son conocidas como *Jardines del edén*

Los grafos de la cuenca de atractores al representar todas las fases del sistema nos permiten conocer el comportamiento dinámico del sistema por completo. Por lo que aplicar métricas a estos grafos nos permitirá relacionar las diferentes reglas de los autómatas celulares elementales (sistemas) y describir las similitudes en la dinámica global de estos sistemas. En principio se pueden aislar los grafos que representa la evolución de los fractales y reconocer comparar equivalentes en cada una de las reglas.

Wuensche establece la relación entre Caos y atractores extraños dentro de los autómatas celulares. Una atractor extraño en los autómatas celulares elementales tiene un alto numero de conexiones con otros nodos (pre-imágenes) en sus ramas y estas son de corta longitud. En las cuencas presentadas se observa estas características, pero es importante notar que existen arboles en las reglas presentadas como no caóticas en otras clasificaciones, este concuerda con la definición de caos utilizada, un sistema es caótico si un subconjunto del estado de fase es caótico. Esto nos permite ilustrar que el caos y los fractales están relacionados además de que las reglas presentadas deben considerarse como caóticas, un ejemplo de esto es la regla 94 y 164 que normalmente no son consideradas

como caóticas y son consideradas con comportamiento periódico o estacionario. Pero si tomamos esta definición de Caos y atractores extraños nos encontramos con un problema. La regla 106 es considerada compleja ya que posee un bajo grado en sus nodos y posee una larga transición en las ramas para llegar al atractor.

Para ordenar los grafos utilizamos el grado de un nodo y la entropía de ese atractor, esta medida nos permite conocer estadísticamente la cantidad de información que posee un árbol. **Grado de un nodo**

Las métricas asociadas al grado de un nodo nos permite conocer que tan conectado o denso están los nodos en un grafo. Wuensche señala que los ECA complejos poseen un numero pequeño de niveles y mayor grado de nodos en comparación con los sistemas complejos. El grado d_i de un nodo c_i esta dado por

$$\sum_{j=1}^N a_{ij}$$

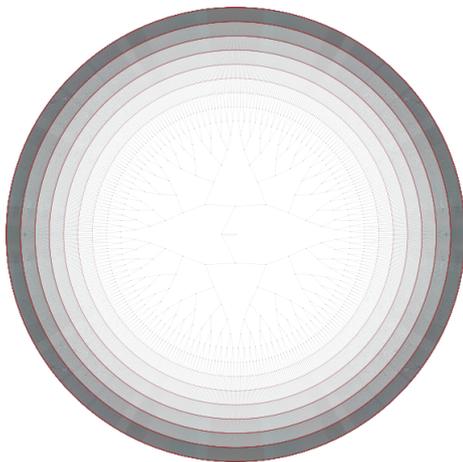
. También podemos definir la probabilidad de que un nodo c_i tenga la densidad d_k al establecer N_k como el número de nodos con densidad d_k y N como el numero total de nodos del grafo. $P[d_k] = \frac{N_k}{N}$ Al tomar en cuenta todos los d_k se puede establecer la formula de entropía de Shannon para un atractor AF .

$$H(AF) = - \sum_{i=0}^N P(d_k) \log_2 P(d_k)$$

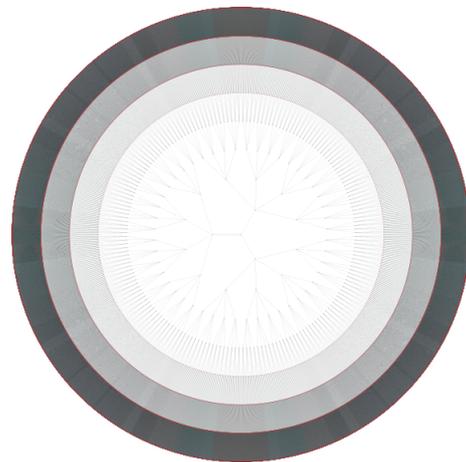
C1: Fractales completos

En el árbol de la regla 90 y la regla 60 al limitar el sistema a una longitud de tamaño 4,8,16 todo el espacio de evoluciones está contenido en una sola cuenca de atractores. Suponemos que esta propiedad se conserva en las longitudes $l = 2^n$, esta longitud de semilla aparece en la definición de fractales [13] [34].

Regla equivalente	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
60	1	17	65536	0 : 000000...	32767 : 0111111111111111	1
90	1	9	65536	0:000000...	9417 : 0010010011001001	0.811278



1)



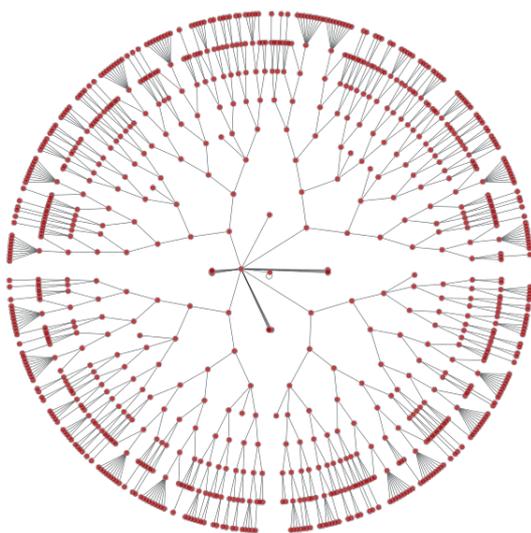
2)

Figura 3.21: 1) Único atractor de la regla 60 con periodo 1. 2) Único atractor de la regla 90 con periodo 1.

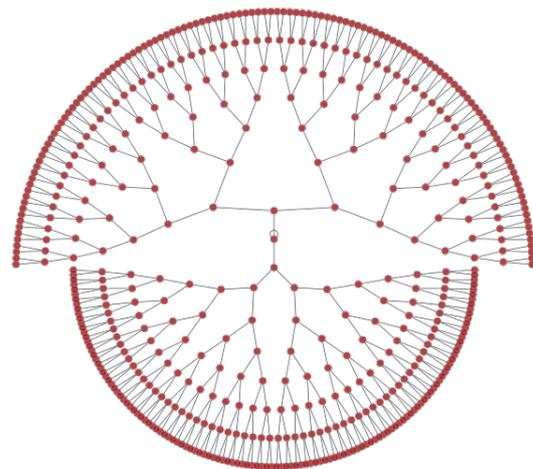
C2 : Localmente caótico

Este conjunto de reglas es interesante para nuestra caracterización ya que son reglas localmente caóticas, es decir solo un subconjunto de estos sistemas presenta caos. Pero este conjunto de valores iniciales que producen caos es muy pequeño. Aunque esté subconjunto es visualmente parecido al fractal $C1$ y también en este subconjunto solo es un grafo que produce caos y fractales es de una clase diferente que $C1$ porque no todo el conjunto de elementos del espacio de fase está contenido en ese grafo

Regla	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
94	1	10	718	0 : 00000000...	32772 : 10000000000000100	1.775944
164	1	9	511	65535 : 11111111...	21973 : 101010111010101	1.01845



1)



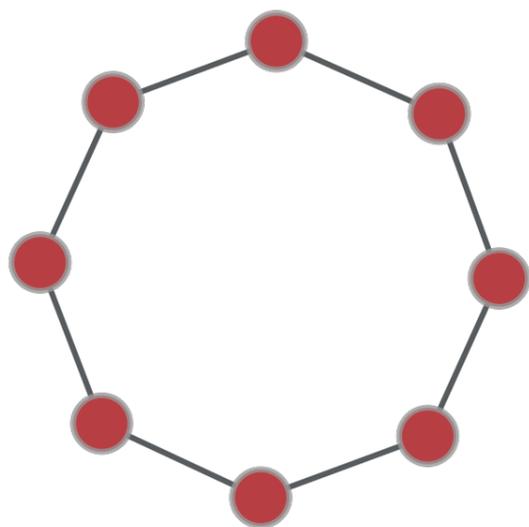
2)

Figura 3.22: 1) Atractor de la regla 94 con periodo uno. 1) Atractor de la regla 164 con periodo uno.

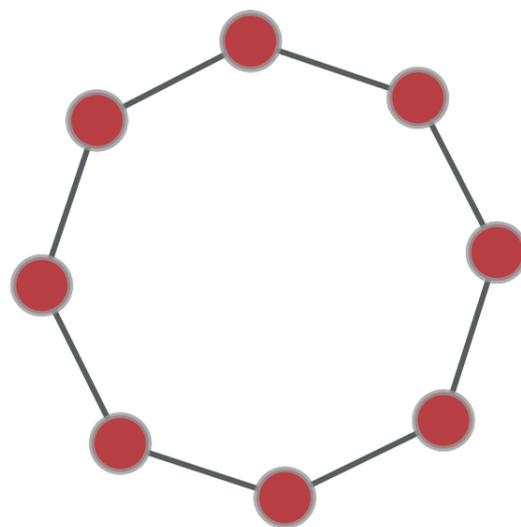
C3 : Sistemas conservativos

La regla 105 y la regla 150 muestran grafos de un ciclo sin que sus nodos tengan arborescencia. Estos nodos son de longitud $l = \frac{2^n}{2}$. Por un lado estos grafos son ejemplos de sistemas conservativos, ya que no hay pérdida de la información y además puede pensarse en una función reversible. Aparentemente estos grafos no presentan propiedades caóticas pero aparece un fenómeno parecido a la bifurcación en el período ya que al cambiar el valor de una célula (que es el cambio más pequeño que puede hacerse) El elemento modificado ya no pertenece al conjunto atractor del primer elemento. Una característica muy importantes de estos sistemas el fractal no aparece en las arborescencias del atractor, como sucede en las reglas 90 y 60 sino que el atractor del sistema es un fractal regular, estos conjuntos no tienen jardones del edén.

Regla	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
150	8	1	8	1 : 0000000000000001	5201 : 0001010001010001	0.0
105	8	1	8	1 : 0000000000000001	5201 : 0001010001010001	0.0



1)



2)

Figura 3.23: 1) Atractor de la regla 105 con periodo uno. 1) Atractor de la regla 150 con periodo uno

C4 : Subconjuntos caóticos disipativos y conservativos

Este conjunto contiene las características anteriores en dos tipos de atractores. El primer tipo de atractor es un sistema disipativo al tener Jardines del Edén, este grafo produce un fractal regular y tiene periodo uno, además no contiene todo el conjunto de estados. El segundo tipo de árbol posee un atractor que es un sistema conservativo, este segundo atractor produce un fractal irregular de periodo l .

Regla	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
154	1	9	511	0 : 0000000000000000	1089 : 0000010001000001	1.0184594
	16	1	16	3 : 0000000000000011	34956 : 1000100010001100	0.0

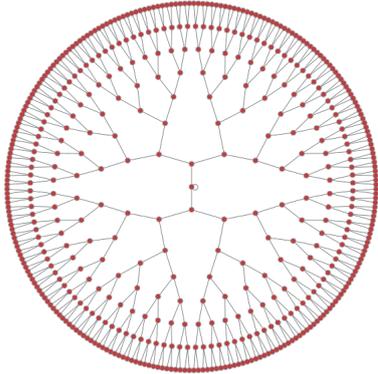
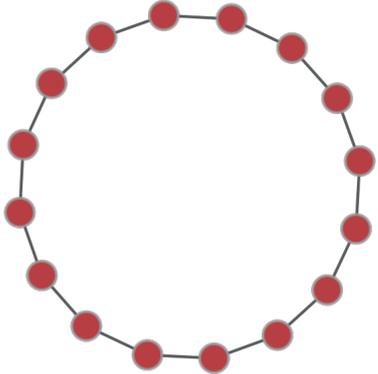
Regla	Periodo 1	Periodo N
154		
Entropía	1,0184594	0,0

Figura 3.24: Atractores para el mosaico M1

C5 : Sistemas disipativos con ciclos de longitud $l \geq 1$ I

Estos sistemas poseen dos tipos de atractores, los dos son sistemas disipativos y cumplen con las características de Wuensche para atractores caóticos: Altamente ramificados y cortas transiciones. El primer árbol produce un fractal regular y el sistema tiene un atractor de periodo uno. El segundo atractor es de periodo $N > 1$ y produce un fractal irregular dentro del ciclo.

Podemos observar *visualmente* que los grafos son parecidos y además los valores de entropía también lo son. Una característica de importante es que los atractores de periodo uno a pesar de que son visualmente parecidos los Mosaicos que producen cada uno de los arboles son diferentes, aquí hay una diferencia entre las capacidades Morfológicas y las capacidades Genotípicas, ver la sección de Fractales regulares 3.2.1 y la comparación entre las clasificaciones. 4.1.

Regla	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
26	1	17	491	0 : 0000000000000000	48 : 0000000000110000	2.299160
	16	7	52	9 : 0000000000001001	11008 : 0010101100000000	2.486458
22	1	42	11721	0 : 0000000000000000	38145 : 1001010100000001	1.869437
	14	12	550	3 : 0000000000000011	10816 : 0010101001000000	2.169767
146	1	23	12287	0 : 0000000000000000	24523 : 0101111111001011	2.509847
	14	9	452	3 : 0000000000000011	52348 : 1100110001111100	2.519789
122	1	24	8958	0 : 0000000000000000	3081 : 0000110000001001	2.554090
	14	9	570	7 : 0000000000000111	33228 : 1000000111001100	2.568387

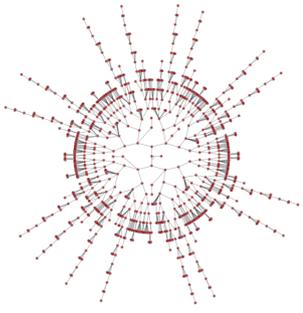
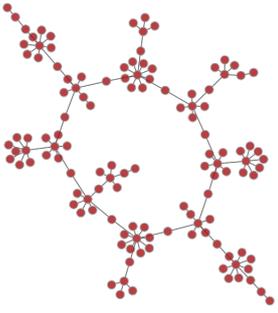
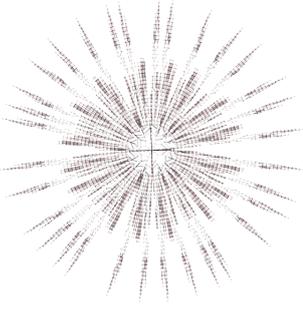
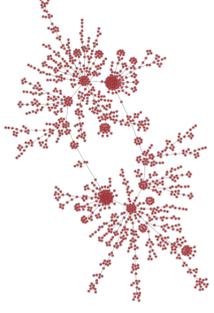
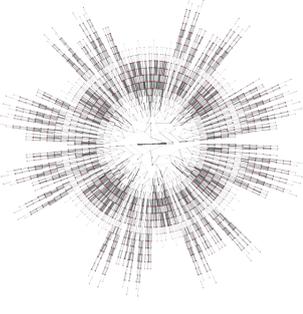
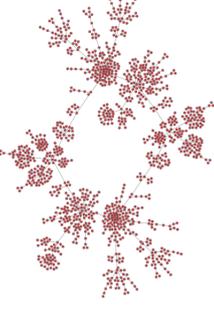
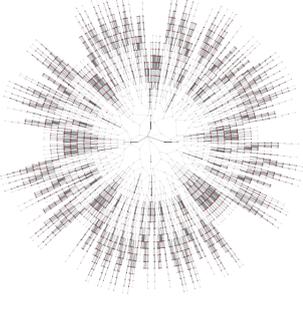
Regla	Periodo 0	Periodo N
26		
Entropía	2,29916	2,21924
22		
Entropía	1,86943	2,16974
146		
Entropía	2,50984	2,51978
122		
Entropía	2,555409	2,56838

Figura 3.25: Atractores de la clase 5

C6 : Sistemas disipativos con ciclos de longitud $l \geq 1$ II

Estos sistemas tienen las mismas características que el grupo C5. La diferencia entre las clases es la entropía y que son visualmente diferentes.

Estos sistemas poseen dos tipos de atractores, los dos son sistemas disipativos y cumplen con las características de Wuensche para sistemas caóticos: Atractores altamente ramificados y con cortas transiciones. El primer árbol produce un fractal regular y el sistema tiene un atractor de periodo uno. El segundo atractor es de periodo $N > 1$ y produce un fractal irregular dentro del ciclo.

Podemos observar *visualmente* que los grafos son parecidos y además los valores de entropía también lo son. Una característica de importante es que los atractores de periodo uno a pesar de que son visualmente parecidos los Mosaicos que producen cada uno de los arboles son diferentes, aquí hay una diferencia entre las capacidades Morfológicas y las capacidades Genotípicas, ver la sección de Fractales regulares 3.2.1 y la comparación entre las clasificaciones.

Regla	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
126	1	22	3782	0 : 0000000000000000	29327 : 0111001010001111	2.9426314
	14	8	120	7 : 0000000000000111	29303 : 0111001001110111	2.638545
18	1	21	3743	0 : 0000000000000000	51912 : 1100101011001000	3.8277185
	14	9	130	3 : 0000000000000011	13105 : 0011001100110001	2.822521

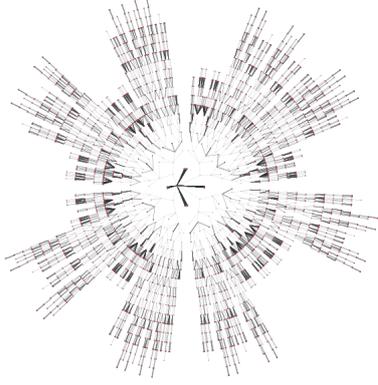
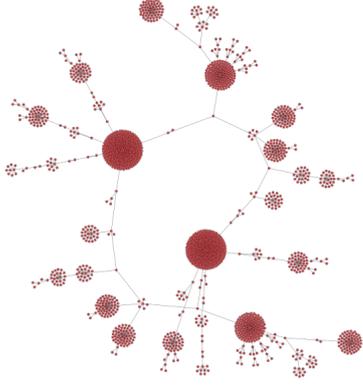
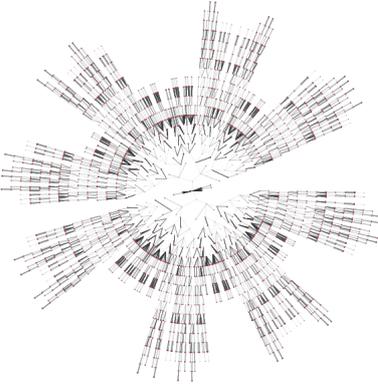
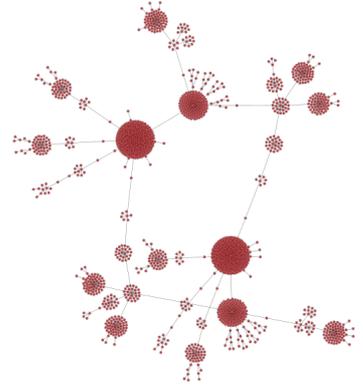
Regla	Periodo 0	Periodo N
126		
Entropía	2,94263	2,63854
18		
Entropía	3,82771	2,82252

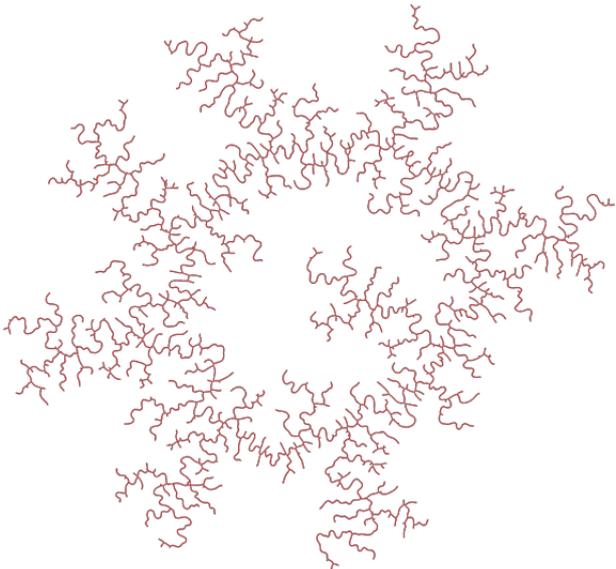
Figura 3.26: Atractores para el mosaico M1

C7 : Sistemas complejos y caóticos

La regla en el fractal 106 es un caso particular, ya que que es un fractal que tiene una operación de corrimiento o **Shift** y no está considerada como una clase que produce al triángulo de Sierspinski en [27]

Las clasificaciones la consideran compleja o caótica. Es considerada por Wolfram [34] como caótica y dentro del trabajo de Wuensche el atractor no cuenta con las características de los sistemas caóticos sino a los sistemas complejos: Los atractores cuentan con pocas ramificaciones y largas transiciones. Pero para Li y Packard [15] y Schüle [28] es considerada caótica.

Regla	Ciclo	Altura	Nodos	Atractor	Jardín del edén	Entropía
106	-	-	-	-	-	-
	2688	401	16224	56220:1101101110011100	61693:1111000011111101	0.232058

Regla	Periodo 0
106	
Entropía	0,232058

3.4.3. Propuesta

Regla	Cuenca I	Cuenca II	Cuenca III	Cuenca IV	Cuenca V	Cuenca VI	Cuenca VII
18						•	
22					•		
26					•		
60	•						
90	•						
94		•					
105			•				
106							•
122					•		
126						•	
146					•		
150			•				
154				•			
164		•					

Cuadro 3.38: Reglas fractales

3.5. Observaciones finales

- Se mostró el conjunto de fractales regulares más pequeños producidos por las reglas analizadas.
- Se mostró que el caos se propaga de distinta manera en las diferentes reglas caóticas.
- Se mostró que los exponentes de Lyapunov no pueden aplicarse a las reglas localmente caóticas ya que el intervalo de densidad en los que emerge el caos y los fractales es muy pequeño.
- Utilizando la teoría del campo promedio Se mostró el punto fijo al que converge el caos en las reglas propuestas
- Utilizando la teoría del campo promedio se mostró que las reglas 94 y 164 son localmente caóticas y el comportamiento global converge a regiones homogéneas o periódicas.
- Se mostró que las reglas que producen fractales tienen atractores extraños en los que emergen los fractales y por lo tanto son sistemas caóticos.
- Se mostró que algunos elementos del conjunto $Per(X)$ producen fractales. y que las reglas 105 y 105 son sistemas conservativos con atractor fractal.
- Se mostró que el atractor que produce el fractal con corrimiento es complejo si se toman algunos resultados anteriores.
- Una limitación del enfoque genotípico utilizado es que no se describió a profundidad los fractales no regulares como se hizo con los fractales regulares.
- Nuestra definición de dimensión fractal es el resultado de resolver la ecuación de recurrencia que describe a ese fractal, y un problema abierto es definir las ecuaciones de recurrencia de los fractales no regulares y del fractal con corrimiento.

- En cuanto al análisis Genotípico y Fenotípico es posible generar una caracterización más específica de las reglas. Mientras que la clasificación por campos de atractores puede extenderse al contemplar configuraciones más grandes

Capítulo 4

Comparación con clasificaciones en la literatura

4.1. Resumen del capítulo

En este capítulo mostraremos la comparaciones entre las reglas utilizando las clasificaciones realizadas en el capítulo anterior, pero se presenta una agrupación de estas reglas y se mencionaran las reglas que no se pueden agrupar de manera completa ya que no cumplen todos los criterios que establecimos para formar los subconjuntos. En las siguientes secciones se hace una comparación con reglas utilizadas en clasificaciones de la literatura que utilizan herramientas similares a las utilizadas en este trabajo, nos interesa mostrar las reglas fractales atraviesan estas clasificaciones y que algunas reglas están consideradas como sistemas con comportamiento *trivial* por lo que no están clasificadas como sistemas caóticos. Es importante tener en mente la definición de sistemas caóticos presentada al inicio del trabajo, un sistema es caótico si algún subconjunto del estado de fases es caótico.

4.2. Propuesta de la caracterización

4.2.1. Caracterizaciones propuestas

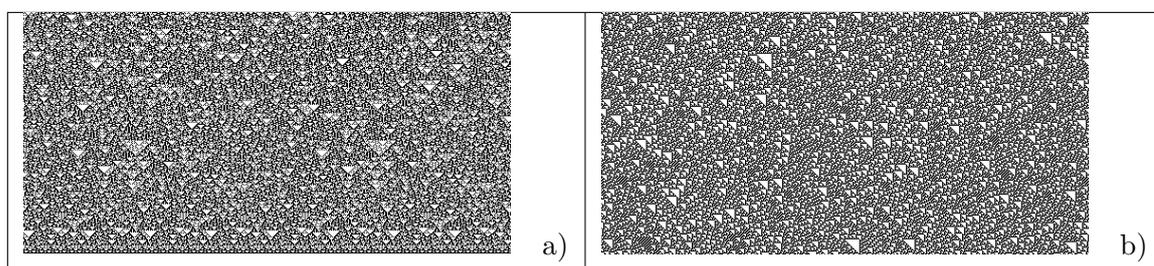
Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
18	MS_1	0.74	E3	C6	G1
22	MS_3	0.90	E2	C5	G1
26	MS_1	0.86	E4	C5	G1
60	MS_2	0.97	E1	C1	G1
90	MS_1, MS_6	0.97	E1	C1	G1
94	MS_6	0.77	E5	C2	G1
105	MA_2	0.97	E1	C3	G2
106	MC_1	0.97	E2	C7	G1
122	MS_6	0.74	E2	C5	G1
126	MS_4, MS_6	0.76	E2	C6	G3
146	MS_1, MS_5	0.75	E3	C5	G2
150	MA_1	0.97	E1	C3	G2
154	MS_1	0.97	E4	C4	G1
164	MS_1	0.44	E5	C2	G4

Cuadro 4.1: Comparación entre las caracterizaciones propuestas para las reglas fractales

Triangulo de Sierpinski

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
60	MS_1	0.97	E1	C1	G1
90	MS_1	0.97	E1	C1	G1

Cuadro 4.2: Reglas fractales



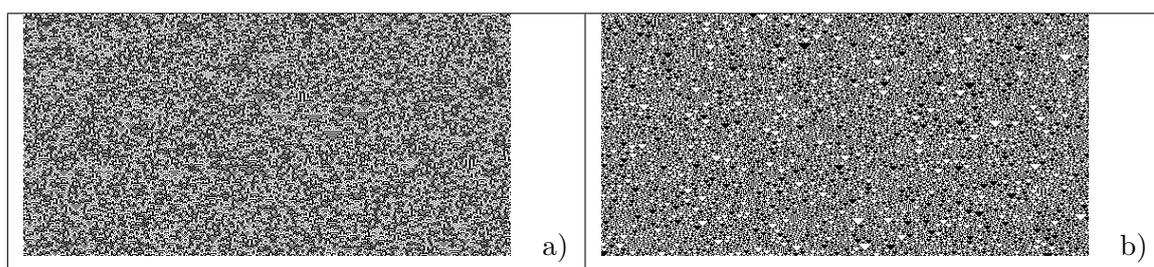
Clase Triangulo de Sierpinski.

a) Regla 60 con densidad inicial de .5. b) Regla 90 con densidad inicial de .5

Triangulo de Sierpinski áureo

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
105	MA_2	0.97	E1	C3	G2
150	MA_1	0.97	E1	C3	G2

Cuadro 4.3: Reglas fractales



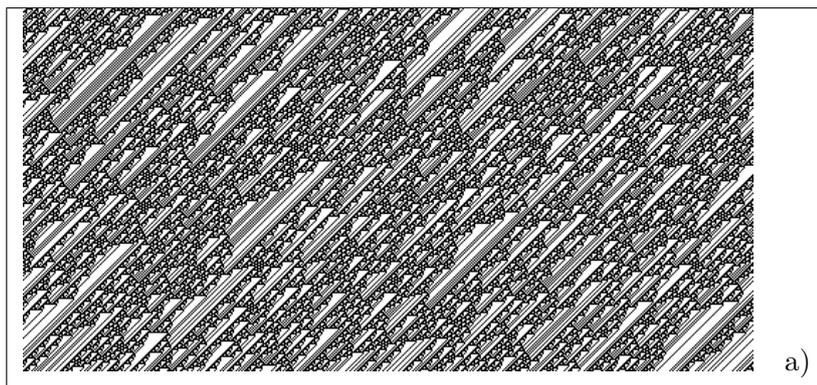
Clase: Triangulo de Sierpinski áureo.

a) Regla 105 con densidad inicial de .5. b) Regla 150 con densidad inicial de .5

Fractales con corrimiento

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
106	MC_1	0.97	E2	C7	G1

Cuadro 4.4: Reglas fractales

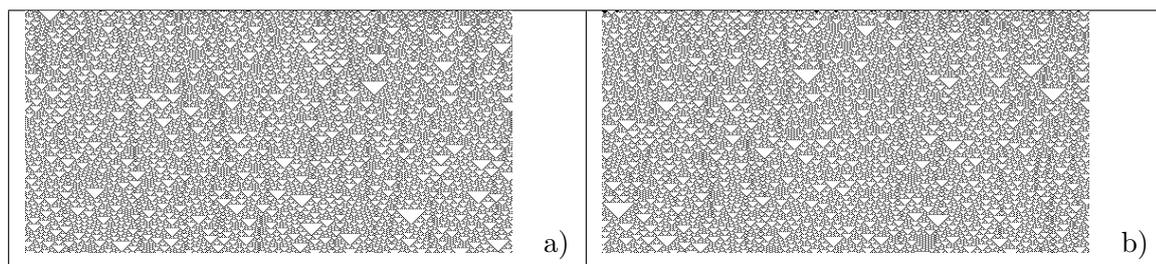


Clase: Fractales con corrimiento.
a) Regla 106 con densidad inicial de .5.

Fractales y caos 1

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
18	MS_1	0.74	E3	C6	G1
146	MS_1, MS_5	0.75	E3	C5	G2

Cuadro 4.5: Reglas fractales

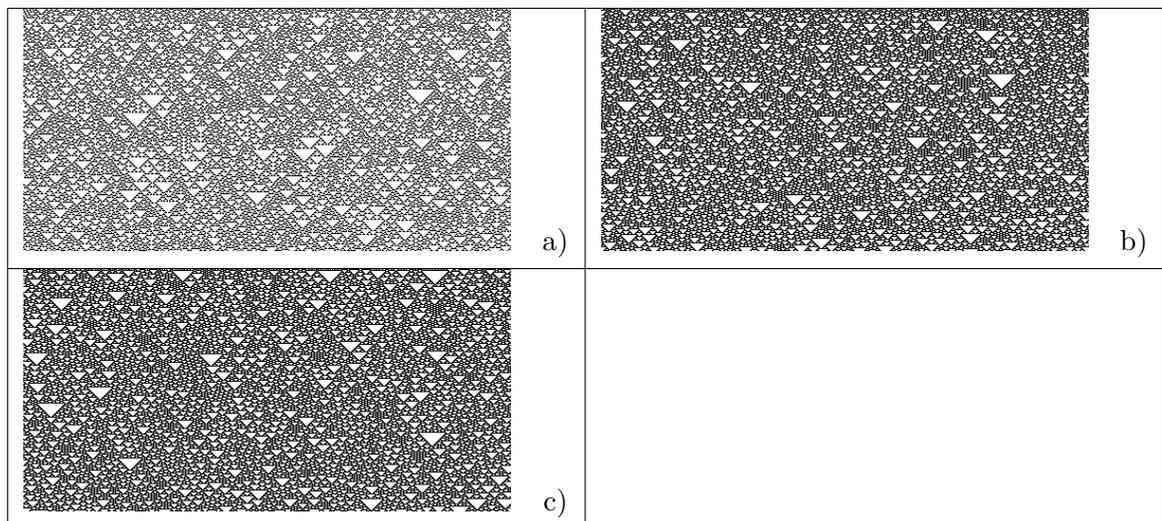


Clase: Fractales y caos I.
a) Regla 18 con densidad inicial de .29. b) Regla 146 con densidad inicial de .33

Fractales y caos II

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
22	MS_3	0.90	E2	C5	G1
122	MS_6	0.74	E2	C5	G1
126	MS_4, MS_6	0.76	E2	C6	G3

Cuadro 4.6: Reglas fractales



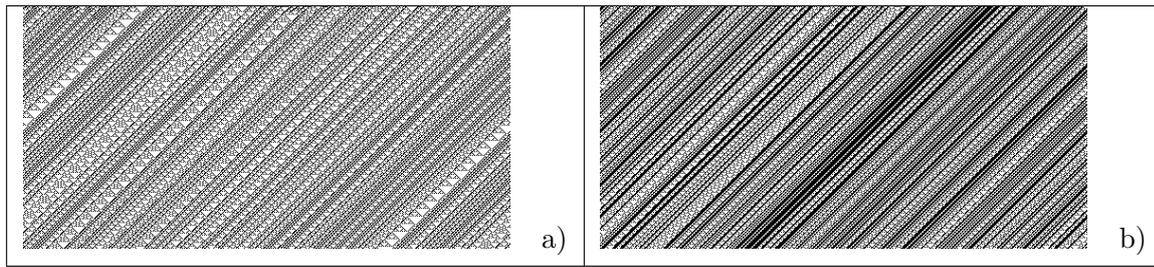
Clase: Fractales y caos II.

a) Regla 22 con densidad inicial de .42. b) Regla 122 con densidad inicial de .61. c) Regla 122 con densidad inicial de .66

Fractales en regiones periódicas

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
26	MS_1	0.86	E4	C5	G1
154	MS_1	0.97	E4	C4	G1

Cuadro 4.7: Reglas fractales



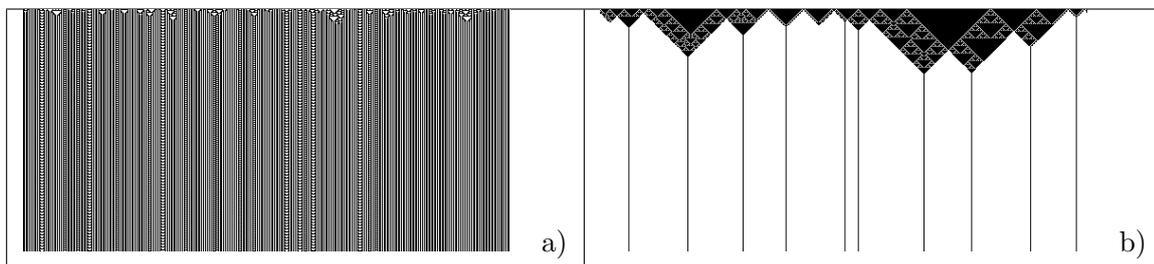
Clase: Fractales y regiones periódicas.

a) Regla 26 con densidad inicial de .38. b) Regla 154 con densidad inicial de .50.

Fractales en regiones homogéneas

Regla equivalente	Diversidad	Entropía	E. Lyapunov	Atractores	Campo promedio
94	MS_6	0.77	E5	C2	G1
164	MS_1	0.44	E5	C2	G4

Cuadro 4.8: Reglas fractales



Clase: Fractales en regiones homogéneas.

a) Regla 94 con densidad inicial de .58. b) Regla 164 con densidad inicial de .50

4.3. Clasificaciones fenotípicas

4.3.1. Clasificación de Wolfram

Clase	Reglas
Uniform (I)	0 , 8 , 32 , 40 , 128 , 136 , 160 , 168.
Periodic (II)	1 ,2 ,3 ,4 ,5, 6, 7 , 9, 10 , 11 , 12, 13 , 14 , 15 , 19 , 23 , 24 , 25 , 26 , 27 , 28 , 29 , 33 , 34 , 35 , 36 , 37 , 38 , 42, 43 , 44 , 46 , 50 , 51 , 56 , 57 , 58 , 62 , 72 , 73 , 74 , 76 , 77 , 78 , 94 , 104 , 108 , 130 , 132 , 134 , 138 , 140 , 142 , 152 , 154 , 156 , 162 , 164 , 170 , 172 , 178 , 184 , 200 ,204 , 232 .
Chaos (III)	18 , 22 , 30 , 45 , 60 , 90 , 105 , 122 , 126 , 146 , 150 .
Complex (IV)	41, 54 , 106 , 110 .

Cuadro 4.9: Clasificación de Wolfram

[34]

4.3.2. Surface dynamics classification

Clase	Reglas
Type A	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 19, 23, 24, 25, 27, 28, 29, 32, 33, 34, 35, 36, 37, 38, 40, 42, 43, 44, 46, 50, 51, 56, 57, 58, 72, 74, 76, 77, 78, 94 , 104, 108, 128, 130, 132, 134, 136, 138, 140, 142, 152, 156, 160, 162, 164 , 168, 170, 172, 178, 184, 200, 204, 232
Type B	18 , 22 , 26 , 30, 41, 45, 60 , 90 , 105 , 106 , 122 , 126 , 146 , 150 , 154
Type C	54, 62, 73, 110

Cuadro 4.10: Surface dynamics classification

[31]

4.3.3. Clasificación de Wuensche

Symmetric	0, 1, 4, 5, 18 , 19, 22 , 23, 32, 33, 36, 37, 50, 51, 54, 72, 73, 76, 77, 90 , 94 , 104, 105 , 108, 122 , 126 , 128, 132, 146 , 150 , 160, 164 , 178, 200, 204, 232.
semi-asymmetric	2, 3, 6, 7, 8, 9, 12, 13, 26 , 27, 30, 34, 35, 38, 40, 41, 44, 45, 58, 62, 74, 78, 106 , 110, 130, 134, 136, 140, 154 , 162, 168, 172.
Full-asymmetric	10, 11, 14, 15, 24, 25, 28, 29, 42, 43, 46, 57, 60 , 138, 142, 152, 156, 170, 184.

Cuadro 4.11: Clasificación de Wuensche

[36]

4.3.4. Expressivity analysis

0	0, 7, 8, 19, 23, 31, 32, 40, 55, 63, 72, 104, 127, 128, 136, 160, 168, 200, 232.
Periodic patterns	13, 28, 50, 54, 57, 58, 62, 77, 78, 94 , 99, 109, 122 , 156, 178.
Complex	30, 45, 73, 75, 110.
Sierpinski patterns	18 , 22 , 26 , 60 , 90 , 105 , 126 , 146 , 150 , 154 .
Finite growth	1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 14, 15, 24, 25, 27, 29, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 46, 47, 51, 56, 59, 71, 74, 76, 103, 106 , 107, 108, 111, 130, 132, 134, 138, 140, 142, 152, 162, 164 , 170, 172, 184, 204.

Cuadro 4.12: Clasificación Expressivity analysis

[27]

4.4. Clasificaciones genotípicas

4.4.1. Clasificación de Li y Packard

Clase	Reglas
Null	0 , 8 , 32 , 40 , 128 , 136 , 160 , 168.
Fixed Point	2 , 4 , 10 , 12 , 13 , 24 , 34 , 36 , 42 , 44 , 46 , 56 , 57 , 58 , 72 , 76 , 77 , 78 , 104 , 130 , 132 , 138 , 140 , 152 , 162 , 164 , 170 , 172 , 184 , 200 , 204 , 232 .
Peridic	1 , 3 , 5 , 6 , 7 , 9 , 11 , 14 , 15 , 19 , 23 , 25 , 27 , 28 , 29 , 33 , 35 , 37 , 38 , 41 , 43 , 50 , 51 , 74 , 108 , 131 , 94 , 134 , 142 , 156 , 178
Locally chaotic	26 , 73 , 154 .
Chaotic	18 , 22 , 30 , 45 , 54 , 60 , 90 , 105 , 106 , 126 , 137 , 146 , 150 , 122 .

Cuadro 4.13: Clasificación de Li y Packard

4.4.2. Topological dynamics classification

Equicontinuous	0 , 1 , 4 , 5 , 8 , 12 , 19 , 29 , 36 , 51 , 72 , 76 , 108 , 200 , 204
Almost-equicontinuous	13 , 23 , 28 , 32 , 33 , 40 , 44 , 50 , 73 , 77 , 78 , 94 , 104 , 128 , 132 , 136 , 140 , 156 , 160 , 164 , 168 , 172 , 178 , 232
Sensitive	2 , 3 , 6 , 7 , 9 , 10 , 11 , 14 , 15 , 18 , 22 , 24 , 25 , 26 , 27 , 30 , 34 , 35 , 37 , 38 , 41 , 42 , 43 , 45 , 46 , 54 , 56 , 57 , 58 , 60 , 62 , 74 , 106 , 110 , 122 , 126 , 130 , 134 , 138 , 142 , 146 , 152 , 154 , 162 , 170 , 184
Sensitive positively expansive	90, 105, 150

Cuadro 4.14: Clasificación Topological dynamics classification

[28]

Weakly periodic	2 , 3 , 10 , 15 , 24 , 34 , 38 , 42 , 46 , 138 , 170
Surjective	15 , 30 , 45 , 51 , 60 , 90 , 105 , 106 , 150 , 154 , 170 , 204
Chaotic	15 , 30 , 45 , 60 , 90 , 105 , 106 , 150 , 154 , 170

Cuadro 4.15: Clasificación Topological dynamics classification

4.4.3. Topological classification

Period-1	0, 4, 8, 12, 13, 32, 36, 40, 44, 72, 76, 77, 78, 94 , 104, 128, 132, 136, 140, 160, 164 , 168, 172, 200, 204, 232
Period-2	1, 5, 19, 23, 28, 29, 33, 37, 50, 51, 108, 156, 178
Period-3	62
Bernoulli σ_t -shift	2, 3, 6, 7, 9, 10, 11, 14, 15, 24, 25, 27, 34, 35, 38, 42, 43, 46, 56, 57, 58, 74, 130, 134, 138, 142, 152, 162, 170, 184
Complex Bernoulli-shift	18, 22 , 54, 73, 90, 105, 122, 126, 146, 150
Hyper Bernoulli-shift	26, 30, 41, 45, 60, 106 , 110, 154

Cuadro 4.16: Clasificación Topological classification

[7]

4.4.4. Creativity

Creative	3, 5, 11, 13, 15, 35
Schizophrenic	9, 18, 22 , 25, 26 , 28, 30, 37, 41, 43, 45, 54, 57, 60 , 62, 73, 77, 78, 90, 94, 105 , 110, 122, 126, 146, 150, 154 , 156
Autistic savants	1, 2, 4, 7, 8, 10, 12, 14, 19, 32, 34, 42, 50, 51, 76, 128, 136, 138, 140, 160, 162, 168, 170, 200, 204
Severely autistic	23, 24, 27, 29, 33, 36, 40, 44, 46, 56, 58, 72, 74, 104, 106 , 108, 130, 132, 142, 152, 164 , 172, 178, 184, 232

Cuadro 4.17: Clasificación Creativity classification

[2]

Regla	Propuesta	Wolfram	Surface	Wuensche	Expresivity
60	TS	Chaos	Type B	Full-asymmetric	Sierpinski
90	TS	Chaos	Type B	Symmetric	Sierpinski
105	TSA	Chaos	Type B	Symmetric	Sierpinski
150	TSA	Chaos	Type B	Symmetric	Sierpinski
106	FS	Complex	Type B	Semi-asymmetric	Finite
18	FC_1	Chaos	Type B	Symmetric	Sierpinski
146	FC_1	Chaos	Type B	Symmetric	Sierpinski
22	FC_2	Chaos	Type B	Symmetric	Sierpinski
122	FC_2	Chaos	Type B	Symmetric	Periodic
126	FC_2	Chaos	Type B	Symmetric	Sierpinski
26	FRP	Periodic	Type B	Semi-asymmetric	Sierpinski
154	FRP	Periodic	Type B	Semi-asymmetric	Sierpinski
94	FRH	Periodic	Type A	Symmetric	Periodic
164	FRH	Periodic	Type A	Symmetric	Finite

Cuadro 4.18: Comparación con clasificaciones fenotípicas

Regla	Propuesta	Li y Pac- kard	Topological dynamics	Topological	Creativity
60	TS	Chaotic	Chaotic	Hyper Bernoulli-shift	Schizophrenic
90	TS	Chaos	Chaotic	Complex Bernoulli-shift	Schizophrenic
105	TSA	Chaotic	Chaotic	Complex Bernoulli-shift	Schizophrenic
150	TSA	Chaotic	Chaotic	Complex Bernoulli-shift	Schizophrenic
106	FS	Chaotic	Chaotic	Hyper Bernoulli-shift	Severely autistic
18	FC_1	Chaotic	Sensitive	Complex Bernoulli-shift	Schizophrenic
146	FC_1	Chaotic	Sensitive	Complex Bernoulli-shift	Schizophrenic
22	FC_2	Chaotic	Sensitive	Complex Bernoulli-shift	Schizophrenic
122	FC_2	Chaotic	Sensitive	Complex Bernoulli-shift	Schizophrenic
126	FC_2	Chaotic	Sensitive	Complex Bernoulli-shift	Schizophrenic
26	FRP	Locally chaotic	Sensitive	Hyper Bernoulli-shift	Schizophrenic
154	FRP	Locally chaotic	Chaotic	Hyper Bernoulli-shift	Schizophrenic
94	FRH	Periodic	Almost- equicontinuous	Period-1	Schizophrenic
164	FRH	Fixed Point	Almost- equicontinuous	Period-1	Severely autistic

Cuadro 4.19: Comparación con clasificaciones genotípicas

4.5. Observaciones finales

Al comparar las clasificaciones anteriores con la **Caracterización de las evoluciones fractales de los autómatas celulares elementales** podemos establecer los siguientes resultados:

- No existe otra clasificación que sea exclusiva de los autómatas celulares elementales que produzcan fractales.
- Las clasificaciones que contemplan como una criterio de clasificación a los patrones fractales observamos que éstas no consideran todas las reglas ni la capacidad generativa de los fractales, por lo que tampoco son completas.
- Las clasificaciones que contemplan como una criterio de clasificación medidas de caos se puede observar que las reglas que consideramos localmente caóticas que producen fractales no son consideradas como caóticas en éstas clasificaciones por lo que tampoco son completas.

Mientras que las limitaciones de este trabajo se pueden expresar de esta forma :

- La caracterización no es completa. Si buscamos ejemplificar como es *imposible* generar una clasificación completa y al presentar en este trabajo una clasificación entonces esta clasificación tampoco será definitiva, seguramente si se aplica otra medida de caos o se utiliza otra herramienta matemática los resultados serán incompatible con nuestra propuesta.
- Este trabajo podría considerarse exploratorio así que también es probable que el conjunto de reglas fractales no sea completo y si se realiza una exploración más extensa es posible que se descubran otro tipo de fractales en el conjunto de reglas caóticas.

Capítulo 5

Simulador

5.1. Introducción

Para realizar la caracterización de Autómatas Celulares Elementales que presentan patrones fractales se desarrolló el simulador llamado Fi (ϕ), éste simulador nos servirá para realizar simulaciones de prueba, realizar análisis y mostrar los resultados de éstas de manera intuitiva para facilitar la visualización del comportamiento del Autómata Celular Elemental. Este simulador fue desarrollado utilizando el lenguaje Python y puede usarse en sistemas operativos Linux y Mac OS. En este capítulo detallamos la estructura de ϕ describiendo los módulos por los que está compuesto, mostrando la interfaz gráfica de usuario y su funcionalidad.

5.2. Simulador

5.2.1. Metodología

Para este trabajo terminal se había planteado hacer un desarrollo en espiral, aunque debido a los objetivos generales que se plantearon este modelo no se ajustaba completamente al trabajo. Uno de estos objetivos principales es desarrollar una caracterización de autómatas celulares elementales que generan patrones fractales, para esto se desarrolló un simulador que ayudara con las diferentes pruebas y a aplicar las diferentes métricas propuestas para la caracterización.

Debido a que la mayoría del software en esta área es desarrollado por investigadores que no están familiarizados con la ingeniería de software no existe algún modelo o metodología que seguir para el desarrollo de software enfocado a la investigación. Una alternativa fue usar el desarrollo en espiral pero se planteó usar la etapa de análisis de riesgos para planificar las pruebas que nos ayudarán a cumplir con el objetivo del software.

Aunque la alternativa propuesta es viable para solucionar el problema de elección de metodología de software la metodología no es óptima, este problema ya ha sido tratado por una comunidad de investigación que para identificar los retos en esta área se refiere a esta como *Desarrollo de Software Científico*. Con el paso de los años las prácticas de investigación de la ingeniería de software se ha enfocado en técnicas y conceptos con el propósito de ser aplicables de manera general al desarrollo de software. A pesar de esto, las prácticas no han sido del todo adoptadas por los científicos ya que éstos se enfocan en la ciencia y no en el desarrollo de software. Además la variación en las diferentes disciplinas, la madurez y la motivación de los proyectos de software científico tienen influencia en los métodos de desarrollo.

La solución que encontramos para éste problema fue usar diferentes características del Manifiesto Ágil. Este puede ser benéfico para el desarrollo de software científico ya que se ajusta al desarrollo iterativo y el carácter exploratorio de la ciencia [10].

Los cuatro principios del Manifiesto Ágil son:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcional sobre documentación exhaustiva.
- Colaboración con el cliente sobre negociación de contrato.
- Respuesta al cambio sobre seguir un plan.

Nosotros optamos por seguir estos principios durante el desarrollo del simulador. Siguiendo el cronograma propuesto , ver Cuadro 5.2.1 se desarrollaron los módulos que integrarían el simulador. El proceso de desarrollo se llevó a cabo primero planificando la funcionalidad de cada módulo, después la codificación y al final se realizaron pruebas unitarias y pruebas de integración con los otros módulos.

Cuadro 5.1: Cronograma de actividades propuesto para el TT

Actividad	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV
Desarrollo del simulador.										
Pruebas del simulador.										
Análisis fenotípico: Generación histograma de densidades.										
Análisis genotípico: Generación de los campos de atractores										
Análisis genotípico: Generación de los polinomios del campo promedio										
Análisis de resultados obtenidos										
Evaluación de TT I.										
Análisis fenotípico: Generación entropía de Shannon										
Análisis fenotípico: Generación exponentes de Lyapunov										
Análisis morfológico: caracterización de la diversidad morfológica										
Generación de la caracterización										
Comparación de las clasificaciones										
Análisis de resultados obtenidos										
Evaluación de TT II.										

Análisis de riesgos

Durante la planificación del trabajo detectamos los siguientes riesgos. Es importante mencionar que la probabilidad estimada no se pudo establecer con certeza, esto debido a no tener conocimiento a priori de las herramientas matemáticas utilizadas ni del proceso de desarrollo de software científico.

Riesgo	Probabilidad	Impacto	Momento
A pesar del desarrollo de la herramienta no poder replicar los resultados de la bibliografía consultada	.5	Catastrófico	Desarrollo
No concluir el desarrollo de alguna de las herramientas matemáticas propuestas	.8	Tolerable	Desarrollo
Que los resultados de las herramientas matemáticas no aporten datos a la clasificación	.3	Graves	Desarrollo
No tener los recursos computacionales suficientes para implementar las herramientas matemáticas.	.2	Tolerable	Desarrollo
Que los resultados obtenidos no sean concluyentes y no podamos realizar la caracterización.	.3	Grave	Desarrollo
Encontrar algún trabajo que ya haya realizado la caracterización de fractales en los autómatas celulares	.2	Tolerable	Desarrollo
Encontrar nuevas reglas fractales de las consideradas inicialmente	.8	Tolerable	Desarrollo

5.2.2. Patrón de arquitectura

El patrón MVC nos permite separar la funcionalidad de la interfaz gráfica de usuario y facilitar la descripción del sistema. Es preciso mencionar que en Python la palabra módulo se usa para referirse a cualquier archivo con extensión .py, este archivo puede contener funciones, variables o clases. La arquitectura propuesta es la siguiente:

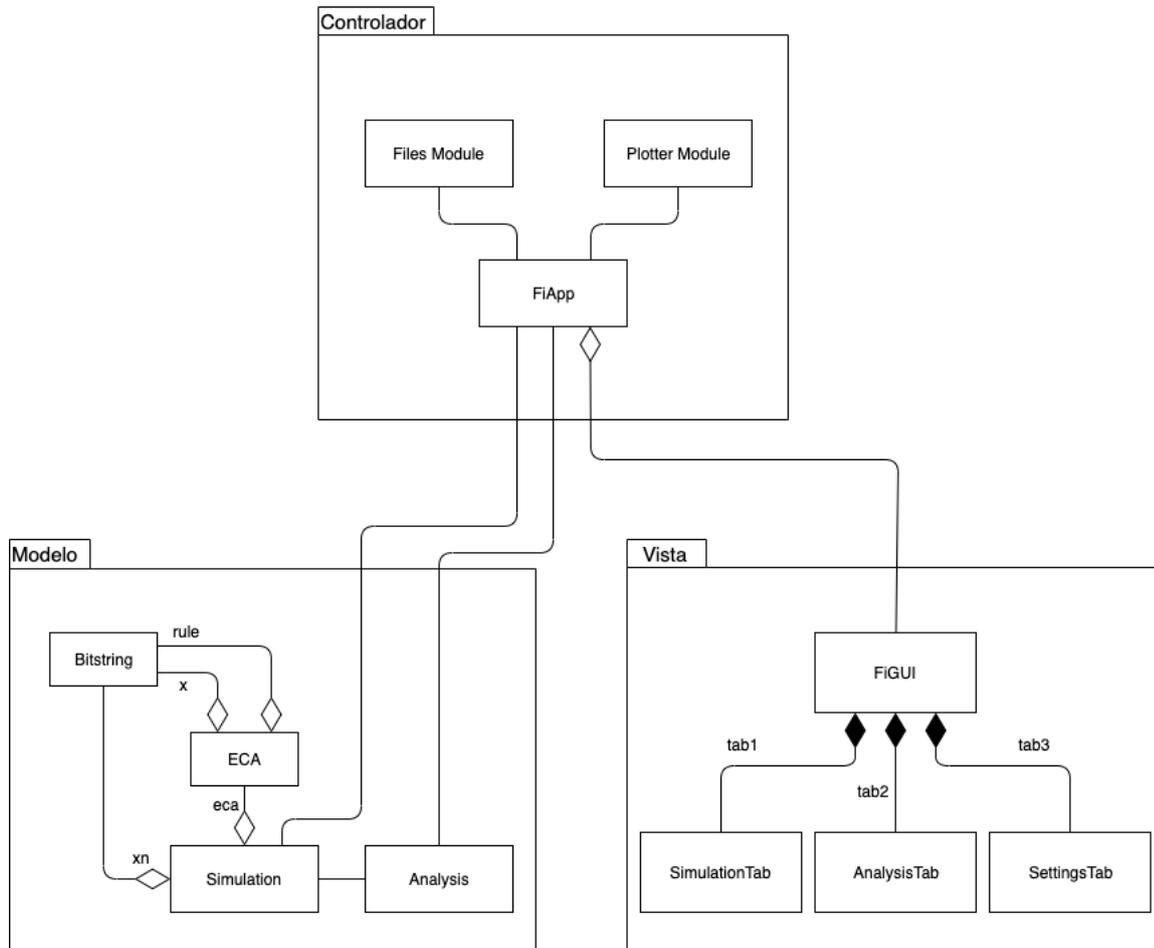


Figura 5.1: Modelo Vista Controlador

- Modelo: En el componente de modelo definimos 4 clases que representan las estructuras y funcionalidad del simulador.
- Vista: En el componente de la vista se encuentra la interfaz gráfica de usuario (GUI). Definimos 4 clases para la GUI, una para cada una de las pestañas y una para la ventana principal.

- **Controlador:** En el componente de controlador tenemos una clase que se encarga de escuchar las señales de cada componente de la GUI y ejecutar la funcionalidad adecuada. Además de dos módulos de Python, uno que nos permite crear gráficas y reportes para presentar los resultados del análisis, mientras que el otro nos ayuda con la manipulación de archivos JSON para guardar y cargar ajustes.

5.2.3. Diagramas de clase

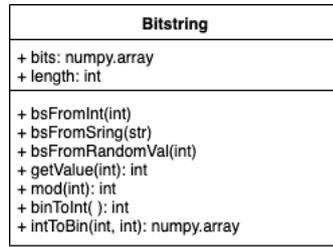


Figura 5.2: Clase Bitstring

La clase Bitstring representa una cadena binaria y es usada para representar las reglas de un ECA y configuraciones de células.

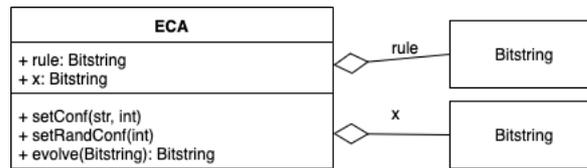


Figura 5.3: Clase ECA

La clase ECA representa un Autómata Celular Elemental

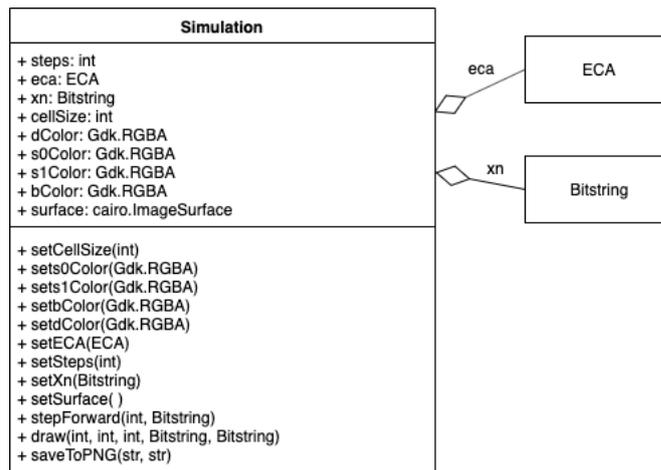


Figura 5.4: Clase Simulation

La clase Simulation representa la evolución de un ECA, nos ayudará a evolucionar configuraciones utilizando una instancia de la clase ECA.

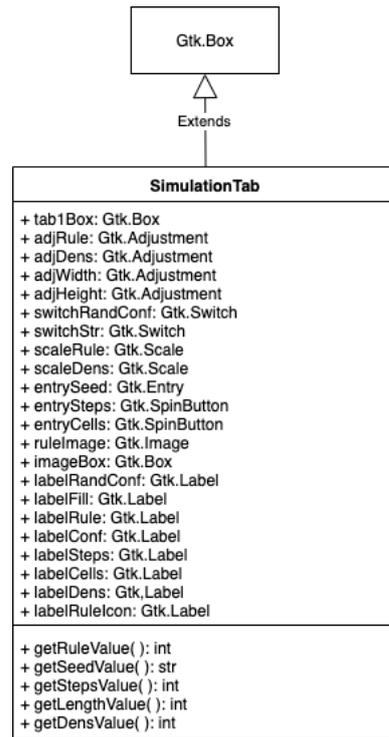


Figura 5.5: Clase SimulationTab

La clase **SimulationTab** contiene los componentes gráficos de la pestaña "Simulation" que sirven para que el usuario especifique los parámetros para una simulación.

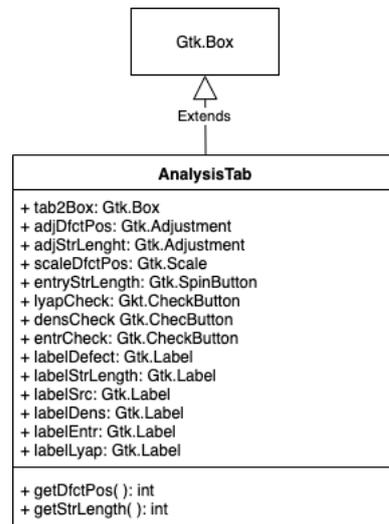


Figura 5.6: Clase AnalysisTab

La clase **AnalysisTab** contiene los componentes gráficos de la pestaña "Analysis" que sirven para que el usuario especifique que métricas se aplicaran en el análisis.

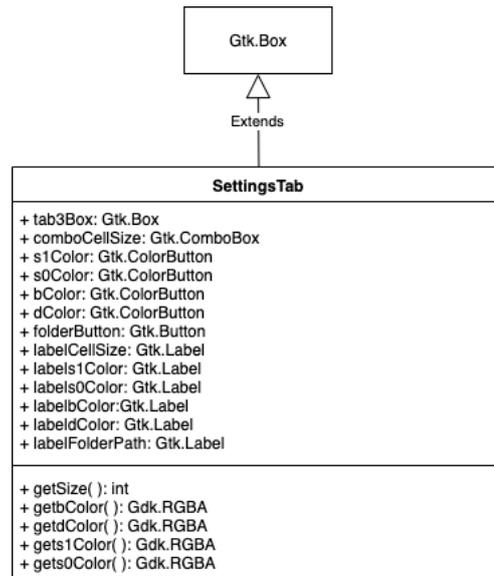


Figura 5.7: Clase SettingsTab

La clase SettingsTab contiene los componentes gráficos de la pestaña "Settings" que sirven para que el usuario modifique las opciones de visualización de las simulaciones.

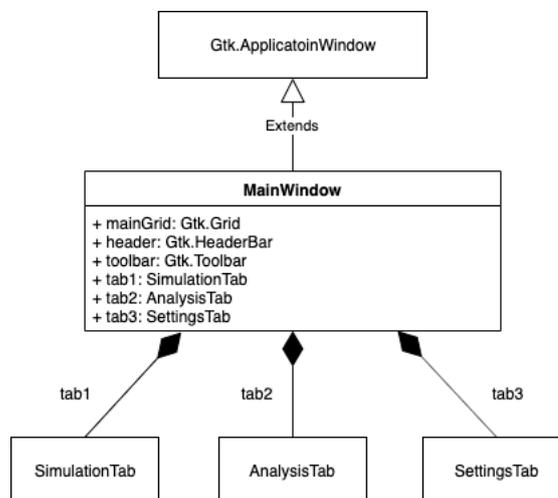


Figura 5.8: Clase MainWindow

La clase MainWindow contiene la ventana principal del simulador la cual contiene la barra de herramientas que permiten al usuario interactuar con el sistemas y las pestañas que permiten al usuario seleccionar los ajustes para simulaciones, análisis y visualización.

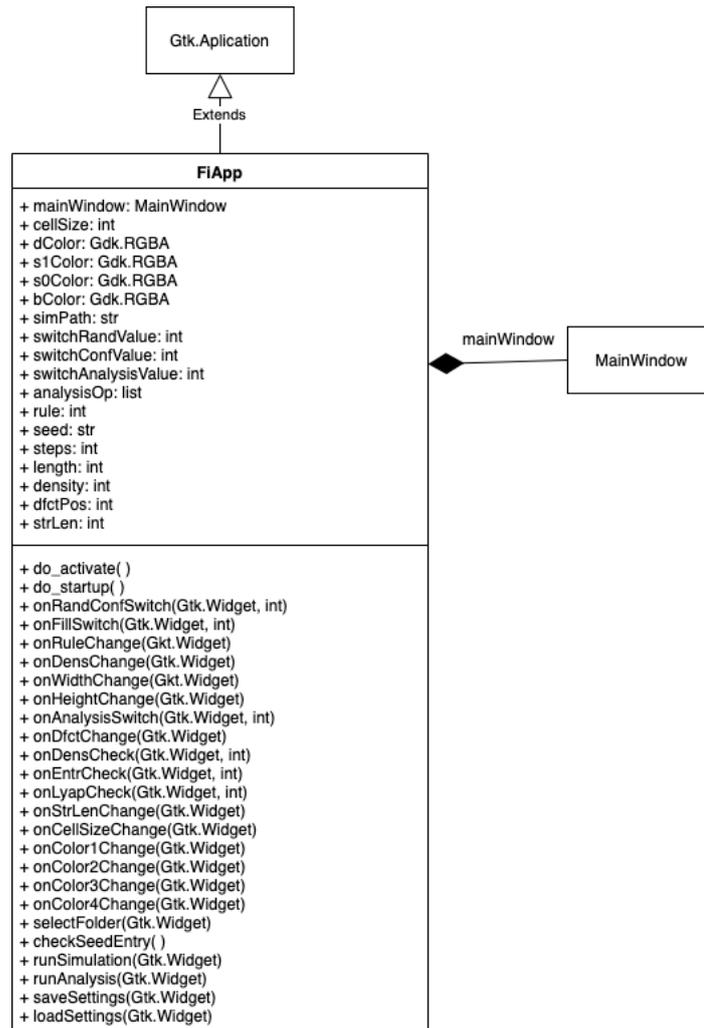


Figura 5.9: Clase FiApp

La clase **FiApp** permite interactuar la GUI con los módulos de Python y con la funcionalidad definida e las otras clases.

Plotter
<pre>+ plotDensity(numpy.array, int, str) + plotEntropy(numpy.array, str) + plotLyap(numpy.array, numpy.array, str)</pre>

Files
<pre>+ openFile(str) + writeJSON(str, dictionary) + imageToString(str): str + stringToImage(str, str) + loadSettings(str): dictionary + writeGenAnalysis(File, dictionary, dictionary) + writePhenAnalysis(File, dictionary, dictionary) + generateReport(str, dictionary, dictionary)</pre>

Figura 5.10: Módulos de Python Plotter y Files

Los módulos Plotter y Files permiten generar gráficas de los resultados obtenidos en el análisis, escribir y leer archivos JSON para guardar y cargar configuraciones y escribir un .tex para generar un reporte del análisis.

5.2.4. Software

Fi ($\phi()$) es un simulador de Autómatas Celulares Elementales (ECA) que además nos permite realizar un análisis fenotípico y genotípico de éstos. $\phi()$ fue desarrollado con el lenguaje de programación Python en su versión 3 para sistemas operativos Linux (Ubuntu) y Mac OS. Las simulaciones y análisis realizados con $\phi()$ nos permitieron realizar la caracterización de ECAs que presentan patrones fractales. Mediante imágenes y gráficas generadas. Con el uso de este simulador hemos logrado sustentar nuestros resultados.

Requerimientos de Software

Funcionales:

- Fi permite realizar simulaciones con una configuración aleatoria definiendo la dimensión y la densidad de células con estado 1 o con una configuración definida por el usuario introduciendo una semilla.
- Fi permite establecer el tamaño de la configuración y los pasos de tiempo que se simularán.
- Fi permite realizar simulaciones con cualquiera de las 256 reglas de los autómatas celulares elementales.
- Fi permite realizar el análisis de una regla o de una simulación de cualquier autómata celular elemental y mostrará un reporte con extensión .pdf.
- El reporte de análisis de una simulación puede contener densidad, exponentes de Lyapunov o entropía y el usuario puede definir la posición en la que se introduce el defecto y el tamaño de cadenas para calcular la entropía topológica.
- El el reporte de análisis de una regla contiene densidad, exponentes Lyapunov, entropía y atractores en una configuración de dimensiones con un tamaño superior para mejor visualización de los resultados.

No Funcionales:

- Fi permite al usuario indicar el tamaño en píxeles de cada célula en la simulación además de los colores para distinguir los estados de las células y el fondo de la simulación.
- Fi permite al usuario guardar y cargar ajustes de simulación.
- Fi permite al usuario indicar el directorio en el que se guardarán las imágenes de la simulación y análisis.

Dependencias:

- Python 3.6
- PyGObject 3.3x (Gtk, Gio, Gdk)
- pycairo 1.18.1
- numpy 1.16.4
- matplotlib 3.0.2
- graph-tool 2.29

Requerimientos de Hardware

- Procesador Quad-Core intel o amd.
- 4GB de memoria RAM (Recomendado 8GB).
- 200MB de almacenamiento.

GUI

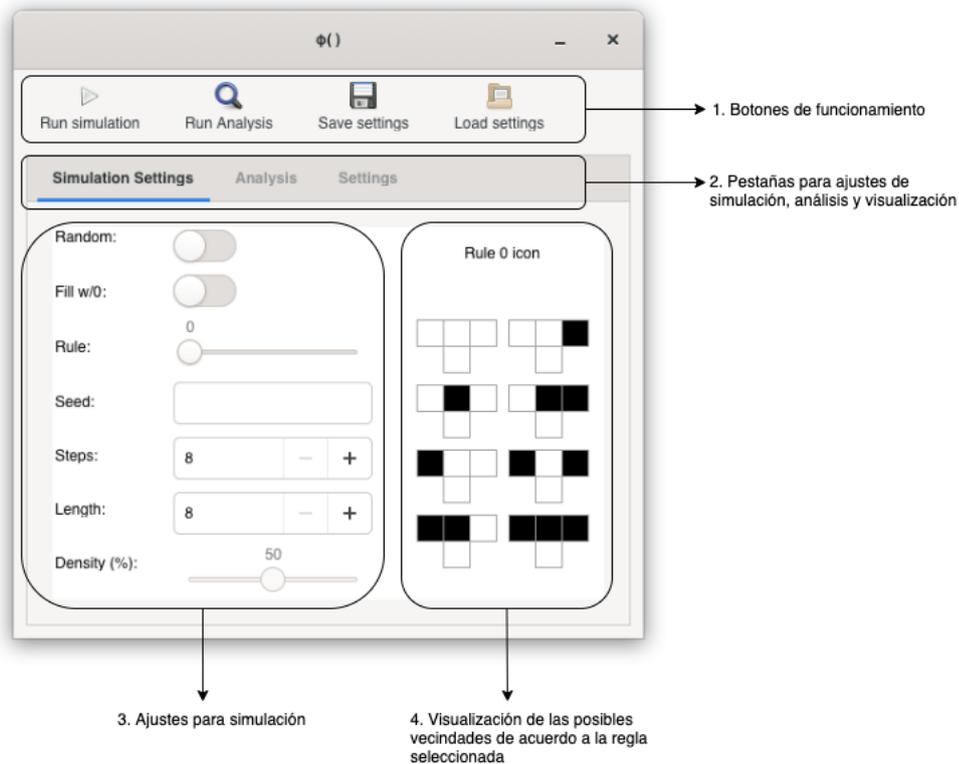
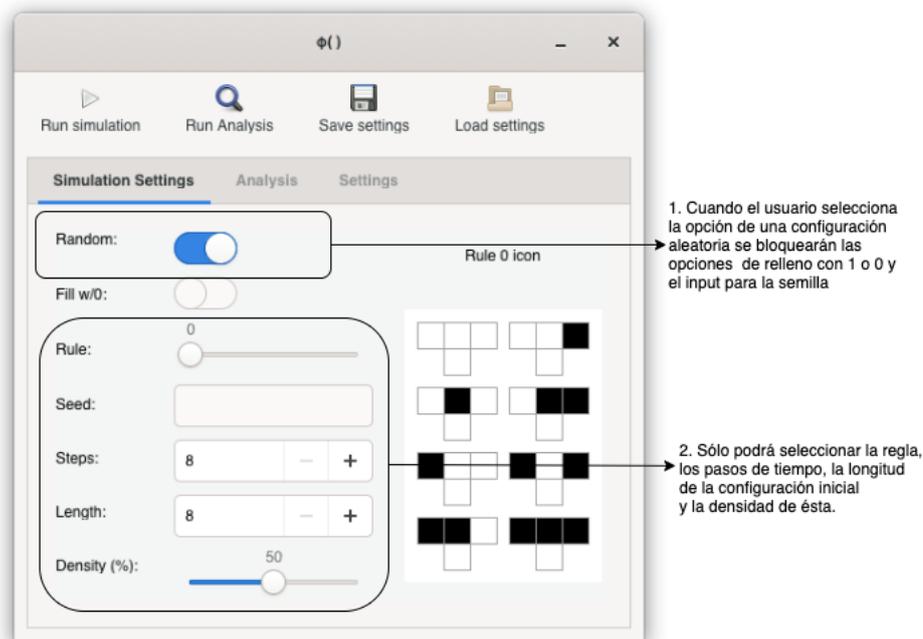


Figura 5.11: Pestaña principal

1. Los botones de funcionamiento nos permiten realizar simulaciones y análisis de acuerdo a los ajustes seleccionados además guardar o cargar datos.
2. Las pestañas de ajustes nos permiten navegar por los ajustes de simulación, análisis y visualización.
3. Ajustes para la simulación, el usuario puede seleccionar si quiere una simulación con una configuración inicial aleatoria y cierta densidad o introducir una semilla. El usuario selecciona la regla, la dimensión de la configuración y los pasos de tiempo de la evolución.
4. En la imagen se muestran las posibles vecindades de acuerdo a la regla seleccionada en los ajustes, está imagen cambia en cuanto el usuario selecciona una regla diferente.



1. Cuando el usuario selecciona la opción de una configuración aleatoria se bloquearán las opciones de relleno con 1 o 0 y el input para la semilla

2. Sólo podrá seleccionar la regla, los pasos de tiempo, la longitud de la configuración inicial y la densidad de ésta.

Figura 5.12: Selección de configuración aleatoria

1. En caso de que el usuario quiera una simulación con una configuración inicial aleatoria, debe accionar el "switch" etiquetado como "Random", al activarse se inhabilitarán el "switch" para rellenar una configuración inicial y el input en el que se introduce una semilla.

2. Para una simulación con una configuración inicial aleatoria el usuario sólo deberá ingresar la regla, el tamaño de la configuración, los pasos de tiempo y se habilitará un "scale" para seleccionar la densidad de células con estado 1.

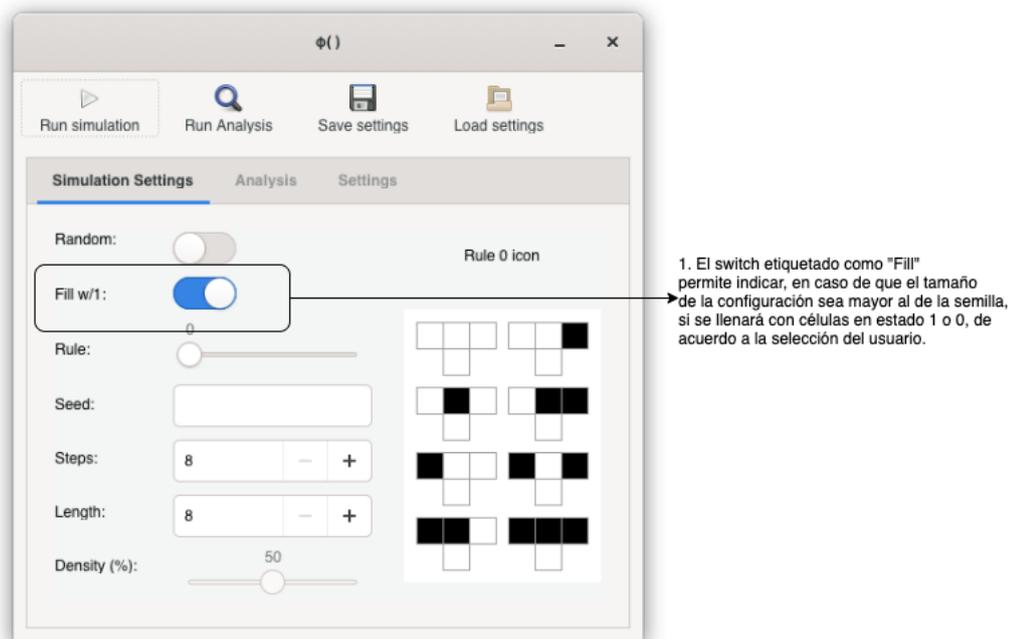


Figura 5.13: Selección de estado para las células de relleno

1. Para ciertas simulaciones el usuario puede requerir un espacio de evoluciones lo suficientemente grande para observar el comportamiento del sistema, es por eso que el simulador permite definir un tamaño de configuración mayor al de la semilla y mediante el "switch" etiquetado como "Fill", el usuario indicará si las células que rellenarán el espacio de evoluciones tienen estado 1 o 0.

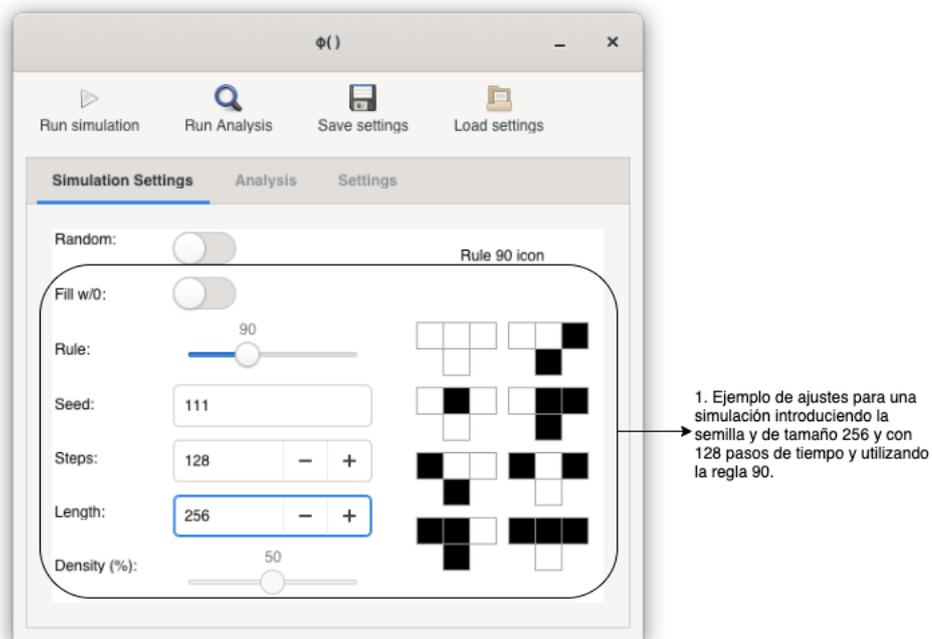


Figura 5.14: Ejemplo de simulación a partir de una semilla

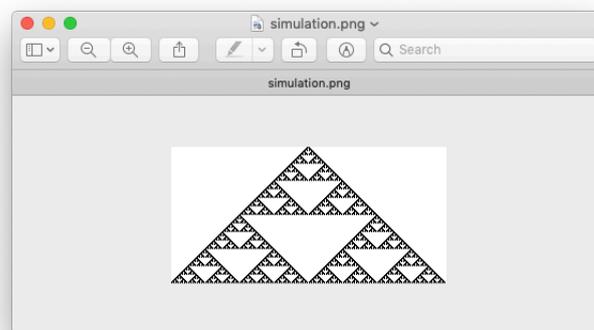


Figura 5.15: Ejemplo de simulación a partir de una semilla

1. De acuerdo a los ajustes seleccionados por el usuario se simulará la evolución del Autómata Celular Elemental, al finalizar se guardará una imagen "simulation.png" en el directorio especificado en la pestaña etiquetada como "Settings" se mostrará con la aplicación especificada por defecto en el sistema operativo para abrir imágenes.

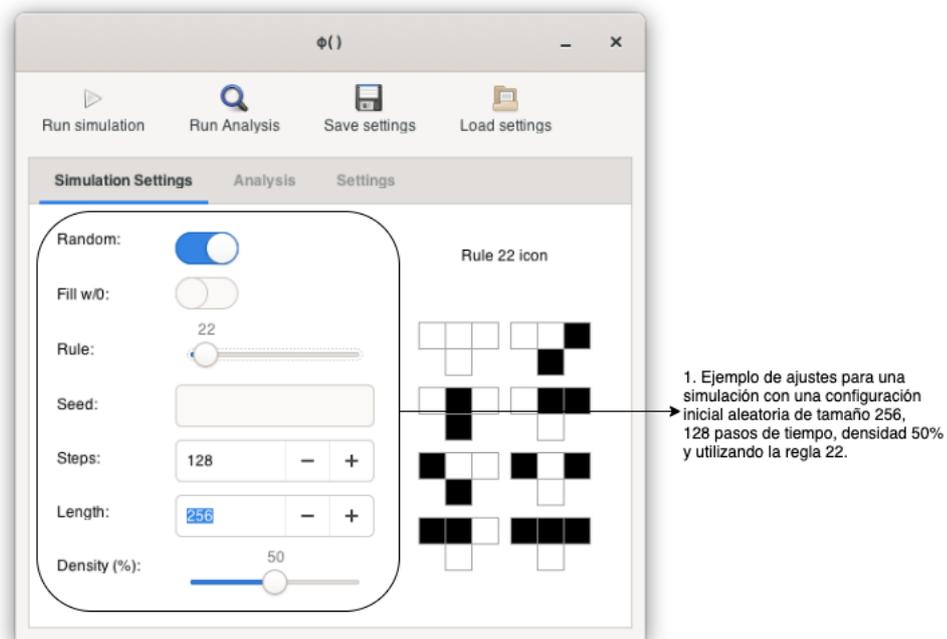


Figura 5.16: Ejemplo de simulación con una configuración inicial aleatoria

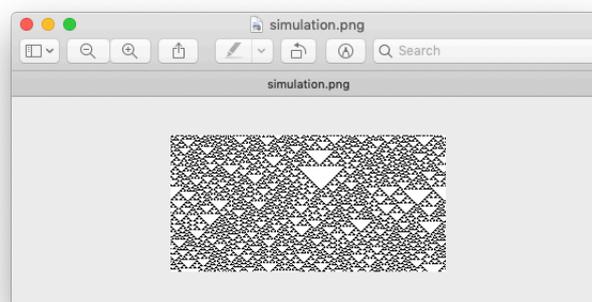


Figura 5.17: Ejemplo de simulación con una configuración inicial aleatoria

1. Si el usuario quiere realizar una simulación con una configuración inicial aleatoria tiene que activar el switch etiquetado como "Random", el usuario debe especificar la regla, el tamaño de la configuración y la densidad de células con estado 1. Al finalizar la simulación se guardará una imagen "simulation.png" en el directorio especificado en la pestaña etiquetada como "Settings" se mostrará con la aplicación especificada por defecto en el sistema operativo para abrir imágenes.

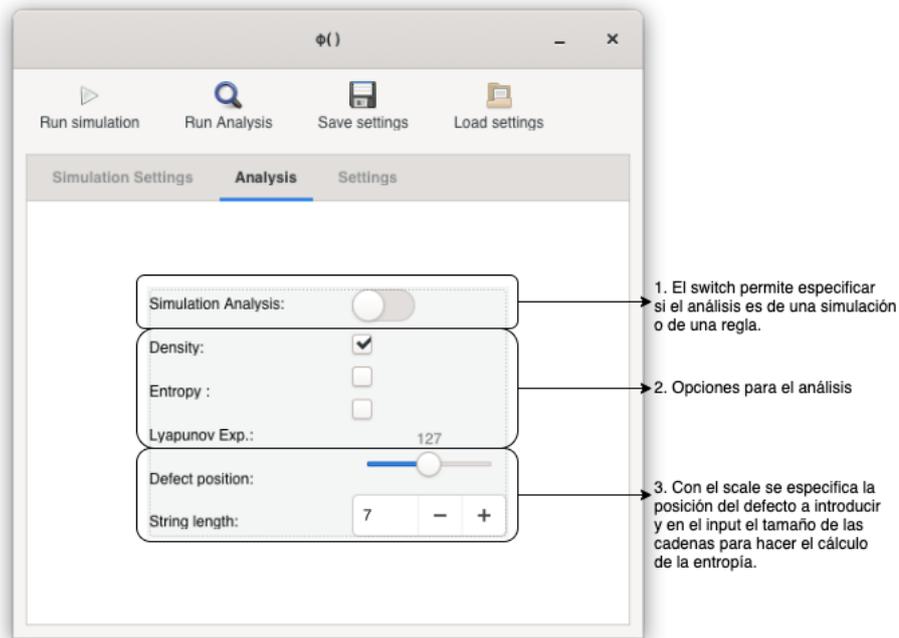


Figura 5.18: Opciones para el análisis

1. El switch etiquetado como "Simulation Analysis" permite al usuario elegir si el simulador hará el análisis de una simulación en específico o si el análisis será de una regla.
2. Si se crea un reporte con extensión .pdf que se mostrará con la aplicación por defecto del sistema operativo. Las opciones seleccionadas serán parte del análisis mostrado en dicho reporte.
3. El scale permite al usuario seleccionar la posición en la que se introducirá el defecto para el análisis de exponentes de Lyapunov, los valores de máximo, mínimo y la posición se modificarán automáticamente al incrementar el tamaño de la configuración en la pestaña de "Simulation Settings" por defecto la posición se establecerá a la mitad de la configuración pero el usuario puede modificar esta posición. El tamaño de las cadenas para el análisis de la entropía también se modificará cuando el tamaño de la configuración se modifique, Si se establecerá el tamaño de cadena recomendado para un mejor análisis.

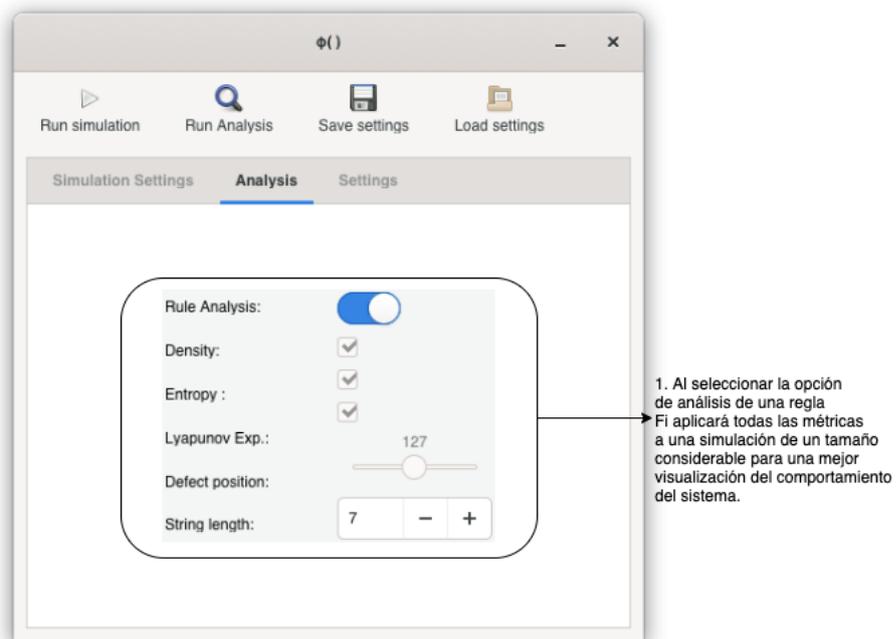


Figura 5.19: Análisis de una regla

1. Para el análisis de una regla se aplicaran todas las métricas además de mostrar el grafo para poder visualizar los atractores y el como se relaciona con el comportamiento del sistema.

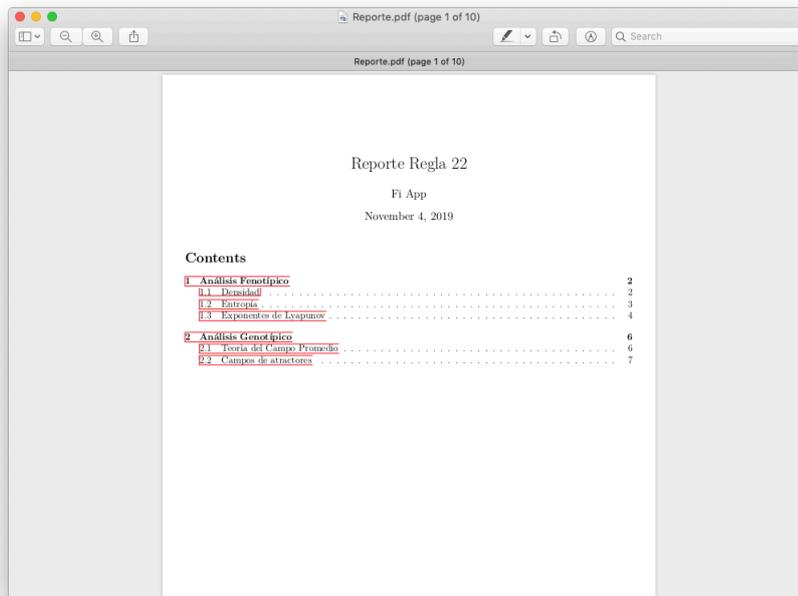


Figura 5.20: Reporte generado

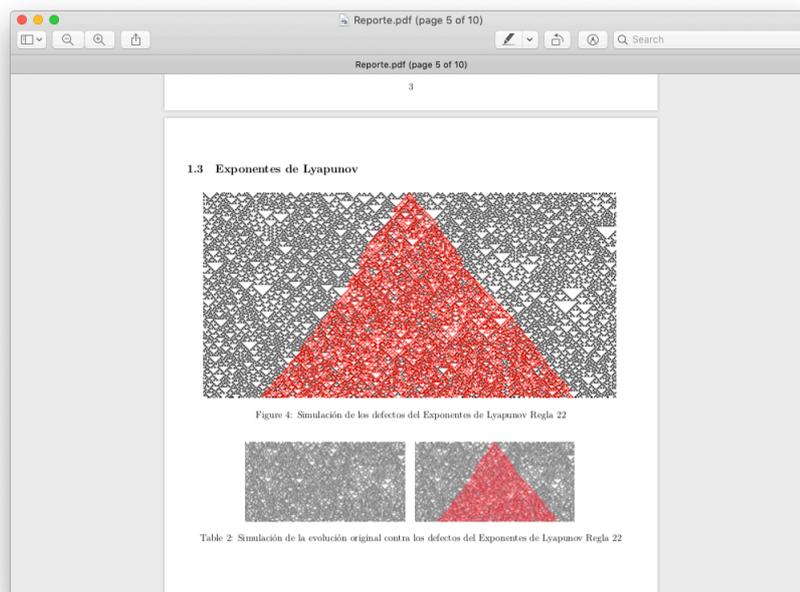


Figura 5.21: Reporte generado

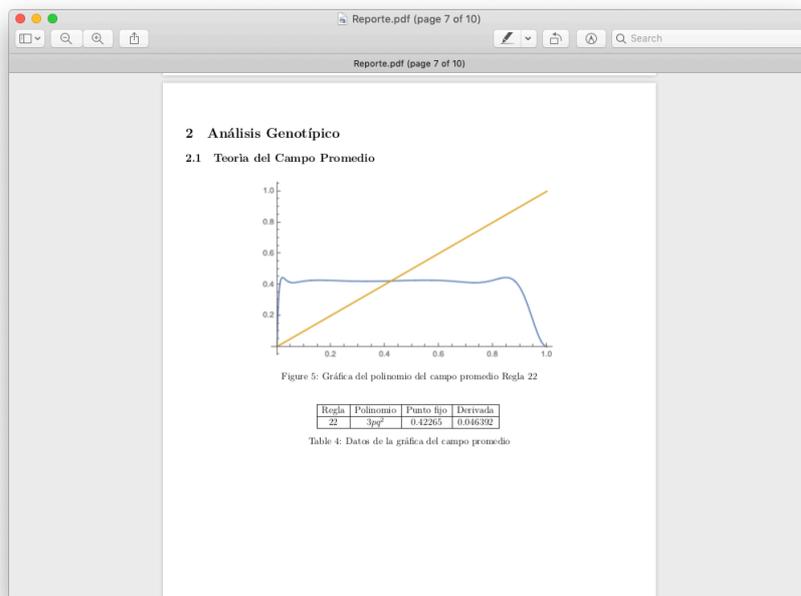


Figura 5.22: Reporte generado

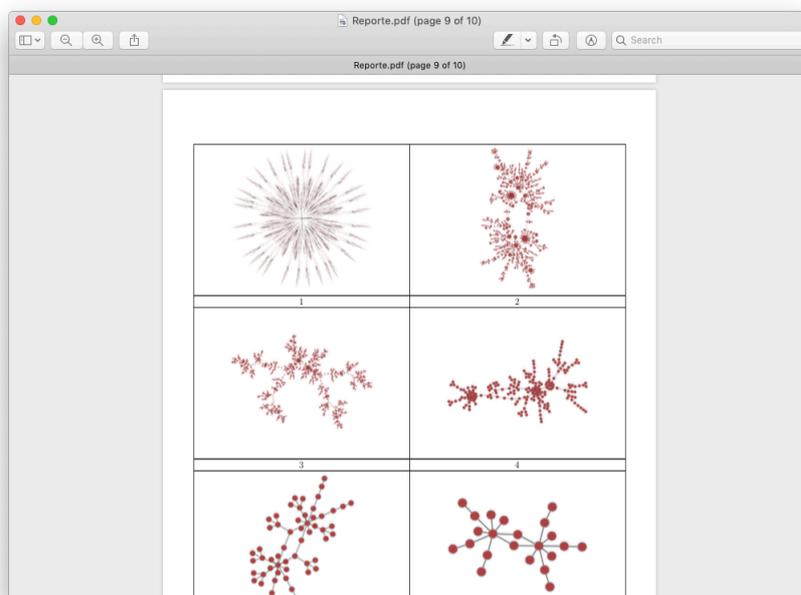


Figura 5.23: Reporte generado

En el reporte generado por Fi el usuario podrá visualizar el resultado de las diferentes métricas aplicadas durante el análisis de una simulación o de una regla.



Figura 5.24: Ajustes de visualización

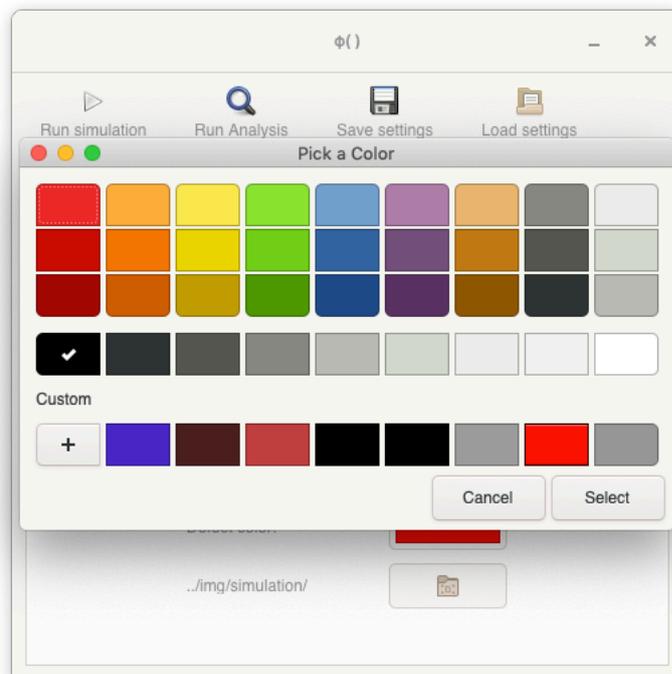


Figura 5.25: Seleccionar color

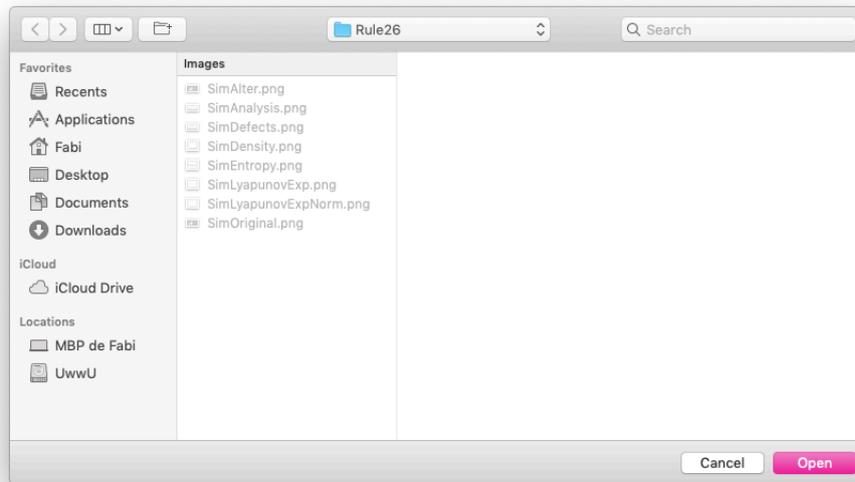


Figura 5.26: Seleccionar directorio

1. El usuario podrá seleccionar el tamaño en píxeles que las células tendrán en la visualización de la simulación y el análisis. El tamaño máximo que las configuraciones y que los pasos de tiempo pueden tener se modificarán al cambiar el tamaño de las células.
2. El usuario podrá seleccionar el color de los elementos de la simulación y el análisis de acuerdo a sus preferencias. Los colores que se muestran son los colores por defecto que Fi usará para realizar las simulaciones.
3. El directorio donde se guardaran las imágenes correspondientes a las simulaciones y análisis, el usuario puede cambiar el directorio si quiere guardar los resultados obtenidos de cierta simulación o análisis. En caso de no seleccionar un directorio en específico las imágenes resultantes se guardaran en el directorio por defecto y los resultados se sobrescribirán cada que Fi realice una simulación o análisis.

```

{"fill": 0, "rule": 150, "seed": "11111", "steps": 32, "cells": 64, "img":
"b' iVBORw0KGgoAAAANSUhEUgAAAAAAAgCAIAAAAt/+nTAAABmJLR0QA/wD/AP+gvaeTAAABZ0LEQVRYhb1WQ7DMAiDav//
cnaoRBkNbUzM0EwqCwh0iEHHGNJgquo/m3YRkaMjQKr6jMcYgQ/R+AQs150DMWnioNzLtb0fLLC4Gc60xIg+aV82oYS4HGgl5E/
XF08oJPp74BDwh20pht/zr4DfNwIBS85nLMkjDmDC7psVmek96i/b1g1keo/6d6x0ICsVs0X/JodiCWV6L/m5Tv37/
aGyMtN7Ad8ApT/AJZTpvYBvgNUfMAK23nuN93LL8gaytQU0AIFM70Vk6jHnK/IeHMhqs fJQ/e7Gmy3dAKrf3Xhv7wTQ+b4bH5c/
XxY633fjMQLoFN+Nn1paQuh8343HCKDzvVkt/oHDhAA63wFadPxz f518Kv+ZncHf
+sBrrnJ8bYM3xLPxKnIsA0t9394HFf58ZnuqxgHM8qw8A+4aI98+7eqCYjpiG0VZm9B1MR0yP+fi/
zwVZay9g0mIGzCELCL3zoH2g5vkCbsC3+I7tXwAAAAASUVRK5CYII="}

```

Figura 5.27: Ajustes de simulación guardados

El usuario puede guardar ajustes de una simulación realizada con Fi, estos ajustes y la imagen de la simulación se guardarán en un archivo .json que el usuario podrá cargar al simulador posteriormente y visualizar o hacer cambios a partir de estos ajustes.

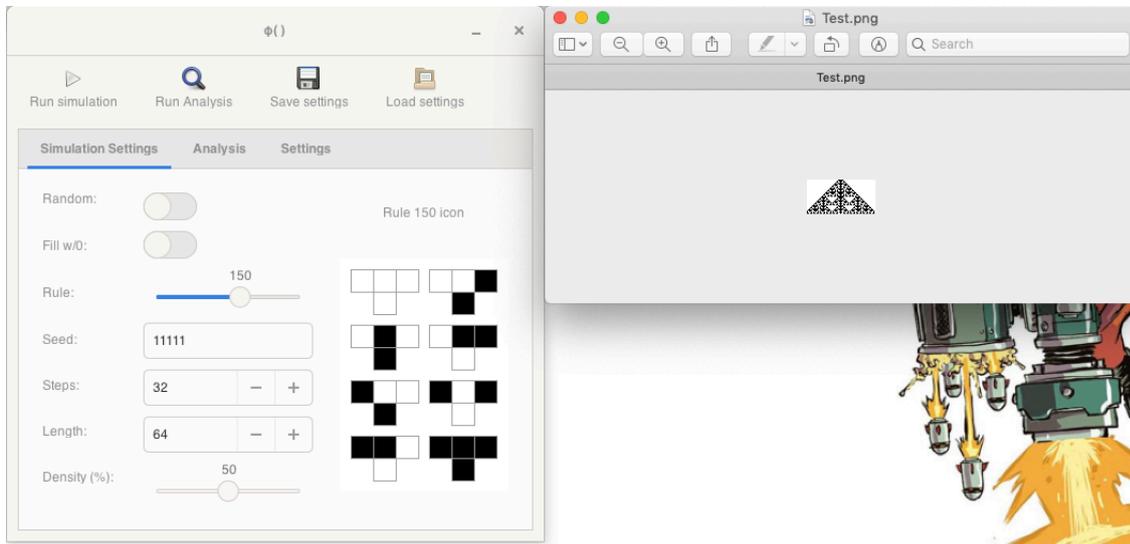


Figura 5.28: Ajustes de simulación cargados a partir de un archivo guardado previamente

Al cargar los ajustes guardados previamente en un archivo .json los ajustes se podrán visualizar en las diferentes entradas de Fi, además la imagen resultado de la simulación se mostrará, esta se abrirá con la aplicación por defecto del sistema operativo para la visualización de imágenes.

Funcionalidad

Antes de comenzar el usuario puede crear un nuevo directorio o definir un directorio diferente para guardar los resultados de las simulaciones y el análisis.

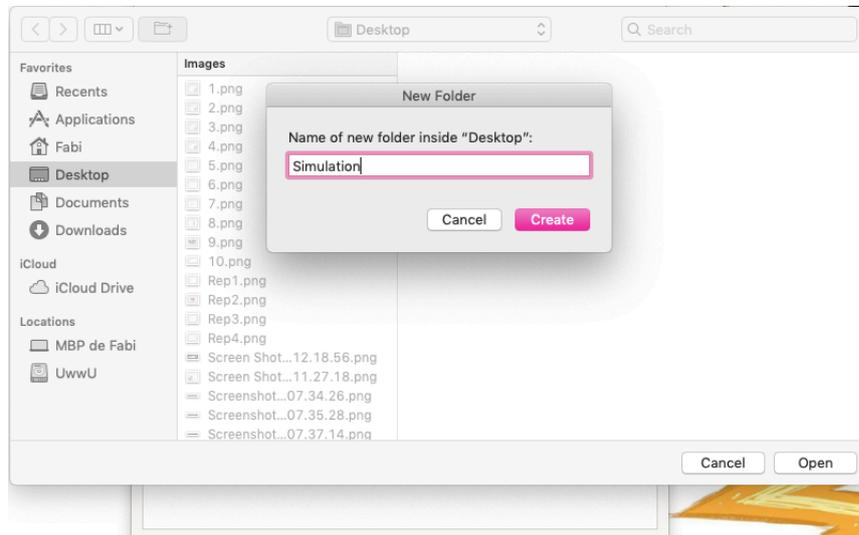


Figura 5.29: Selección de un directorio diferente para guardar los resultados.

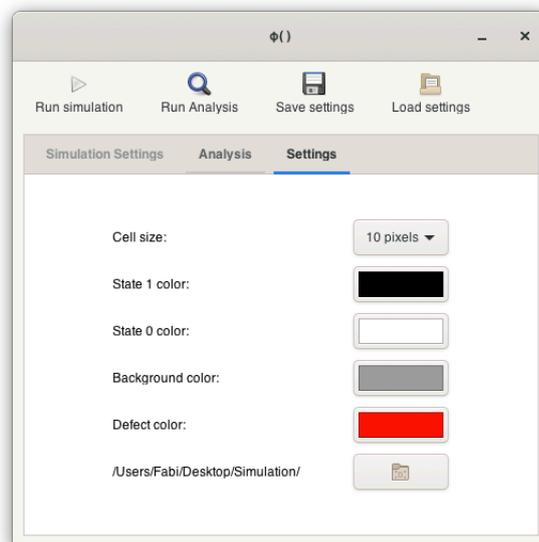


Figura 5.30: Ajustes de visualización.

El directorio seleccionado aparecerá a un lado del botón para seleccionar directorio en la pestaña Settings, además se puede seleccionar el tamaño en píxeles de las células y el color de los diferentes componentes para la visualización de las simulaciones.

Para realizar una simulación el usuario debe indicar en la pestaña Simulation Settings la regla, el tamaño de la configuración inicial, los pasos de tiempo y si es una configuración aleatoria o quiere introducir una semilla.

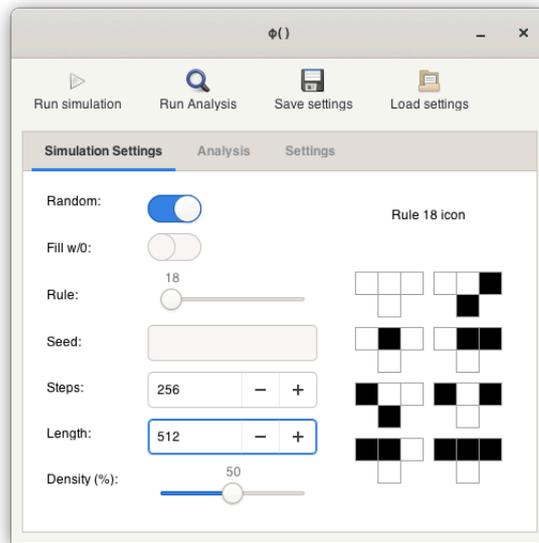


Figura 5.31: Ajustes para realizar una simulación.

Al finalizar la simulación se guardará la simulación en una imagen con extensión .png y se mostrará con la aplicación por defecto del sistema operativo para abrir imágenes.

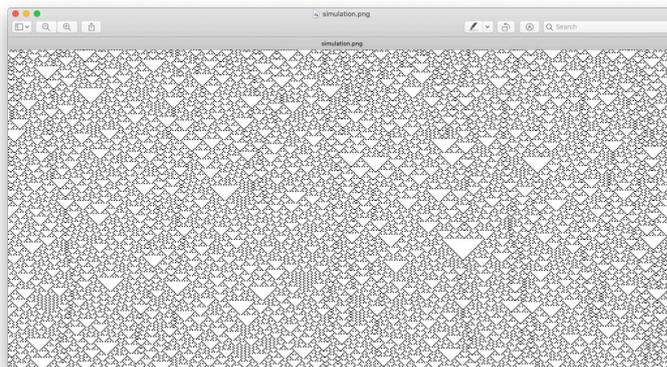


Figura 5.32: Resultado de la simulación.

Para realizar la caracterización se realizaron las pruebas definiendo configuraciones aleatorias y con una semilla para generar un patrón fractal en la simulación.

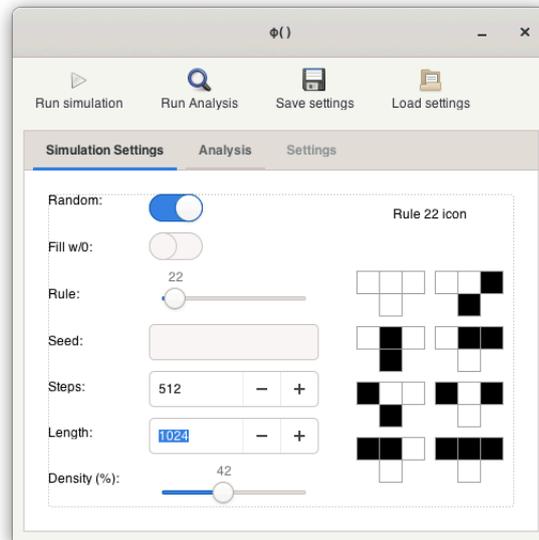


Figura 5.33: Ejemplo de simulación para la caracterización.

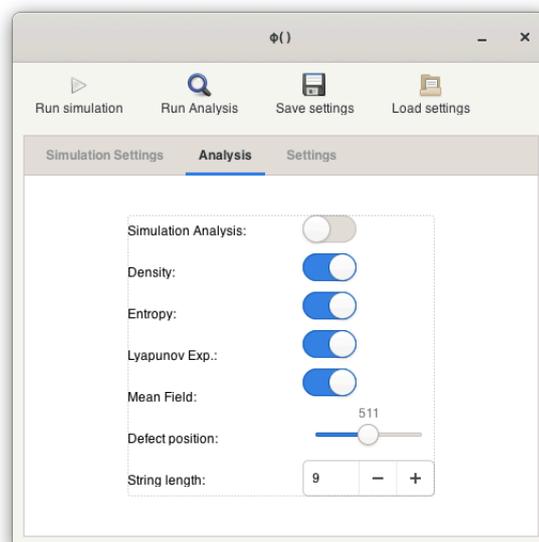


Figura 5.34: Ajustes de análisis.

En la pestaña Analysis se pueden seleccionar las métricas que se aplicarán y opciones para éstas.

Para la caracterización seleccionamos todas las métricas y con el ejemplo mostrado anteriormente el simulador generará las siguientes imágenes como resultado del análisis.

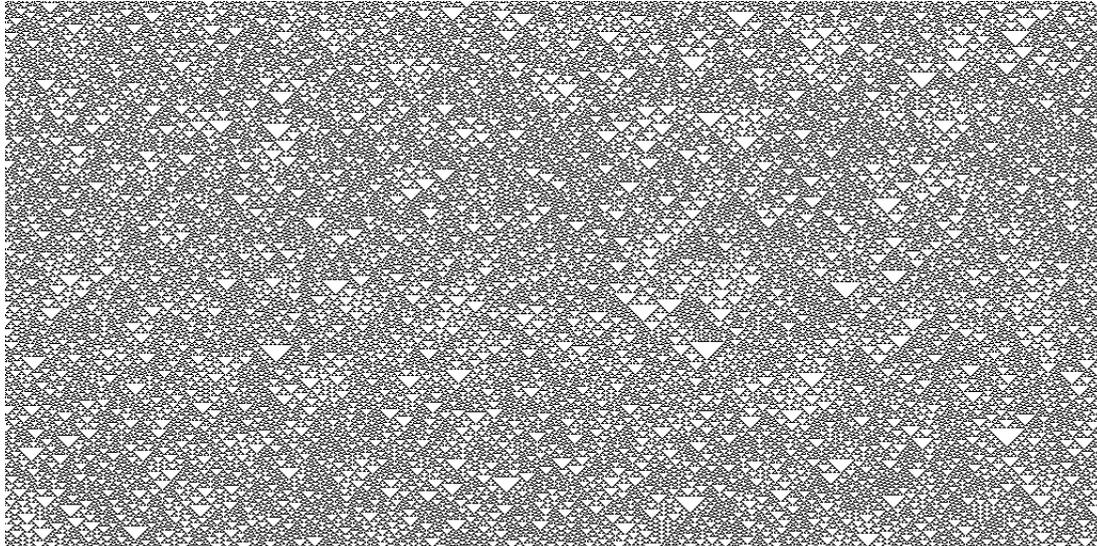


Figura 5.35: Evolución original

Para el análisis de Exponentes de Lyapunov se introducirá un defecto en la célula central de la configuración inicial. La configuración inicial original y la configuración inicial alterada evolucionarán y se compararán para detectar la propagación del daño.

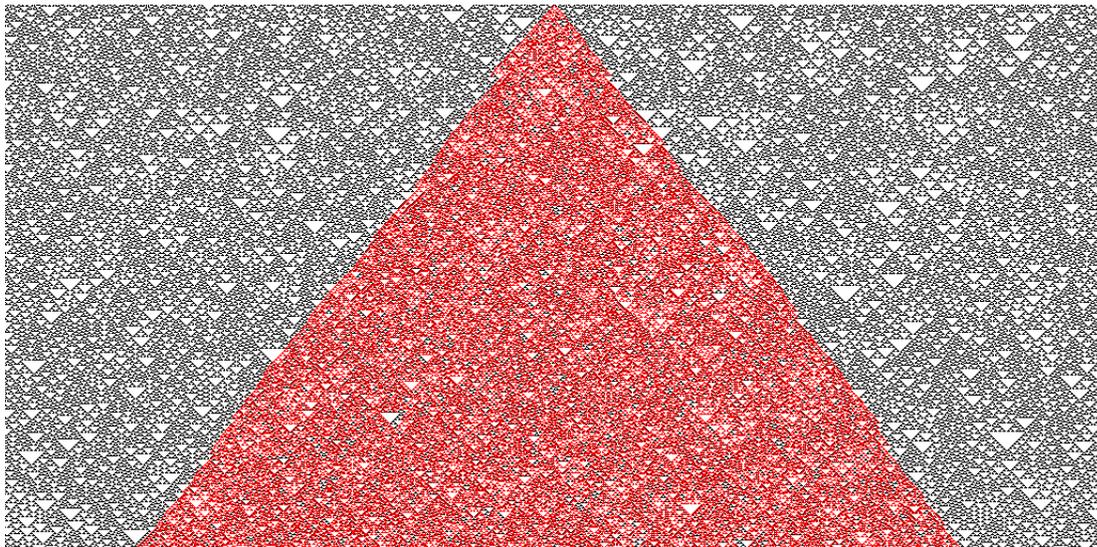


Figura 5.36: Propagación del daño

Para comparar la evolución original y la evolución alterada se generan las siguientes imágenes donde sólo se muestran las células dentro del radio por el cual se propaga el daño.

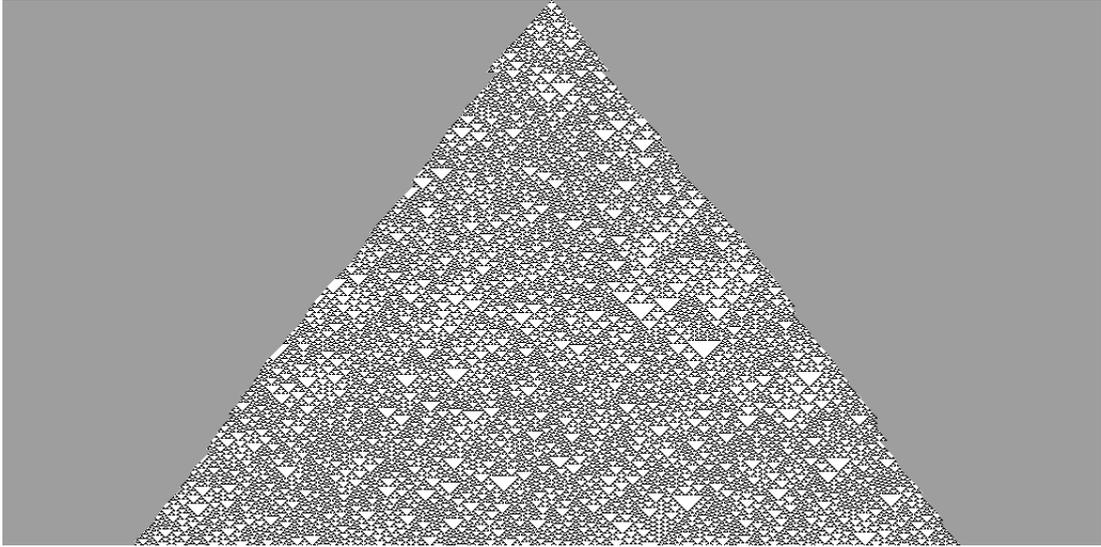


Figura 5.37: Evolución original

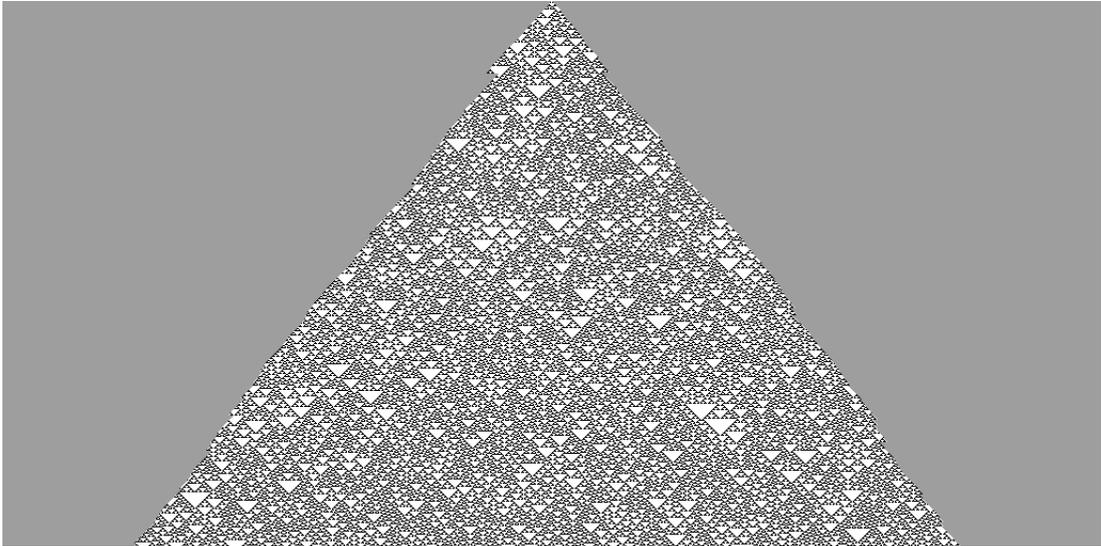


Figura 5.38: Evolución de configuración alterada

La gráfica obtenida es conocida como perfil de Lyapunov, muestra las velocidades promedio de propagación en todas las direcciones posibles.

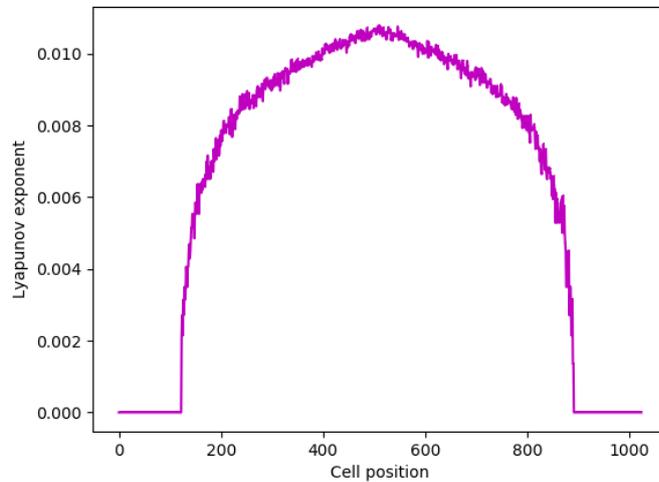


Figura 5.39: Perfil de Lyapunov

La entropía topológica nos muestra la cantidad de información en el sistema en cada paso de tiempo. Para autómatas celulares elementales el análisis pretende determinar que cantidad de todas las posibles cadenas de una longitud dada se pueden encontrar. A mayor cantidad de cadenas encontradas mayor es la entropía y se considera que el comportamiento del sistema es más caótico.

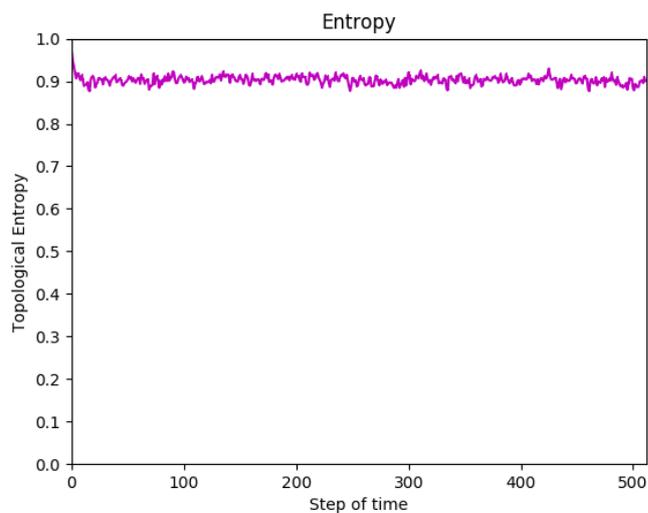


Figura 5.40: Gráfica de entropía topológica

La gráfica de la densidad nos muestra la cantidad de células en estado uno en cada paso de tiempo.

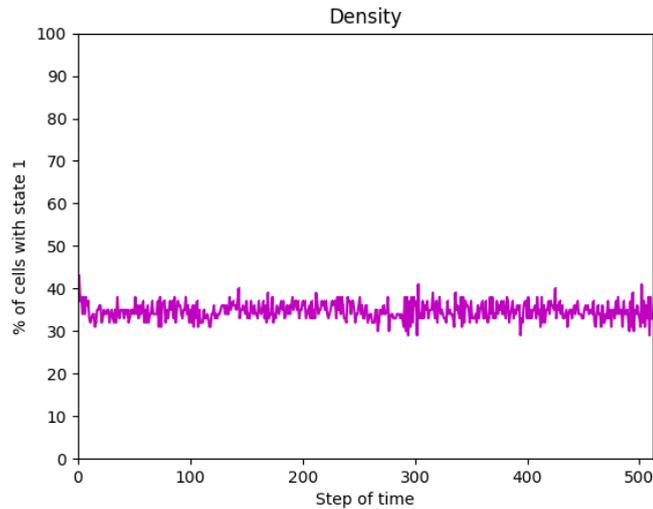


Figura 5.41: Gráfica de densidad

La gráfica del campo promedio es una medida global del sistema. Relaciona la densidad inicial de células con estado uno con la probabilidad de que cada células del sistema evolucione al estado uno, por esta característica se dice que puede *predecir* el estado global del sistema. Si el polinomio se cruza con la recta identidad en un punto se dice que es un punto fijo. En las reglas caóticas este valor nos indica el promedio de unos que tendrá las configuraciones caóticas.

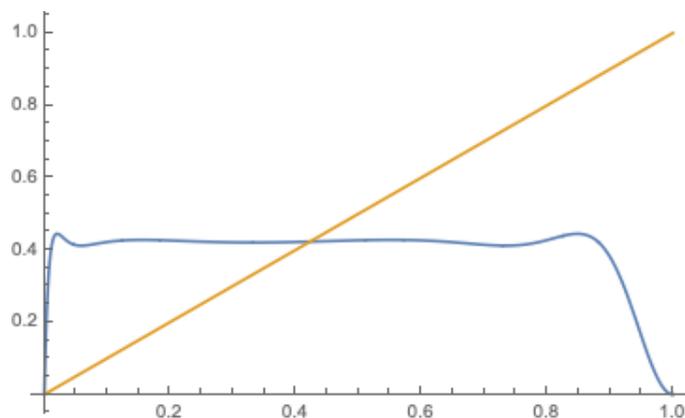


Figura 5.42: Gráfica del campo promedio

Al terminar las simulaciones y el análisis se escribirá un reporte donde se muestran los resultados obtenidos.

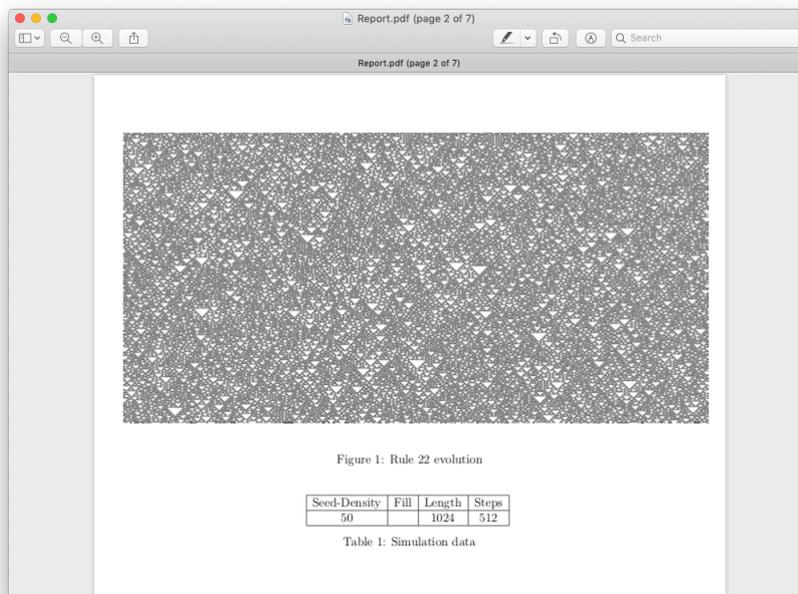


Figura 5.43: Reporte de análisis

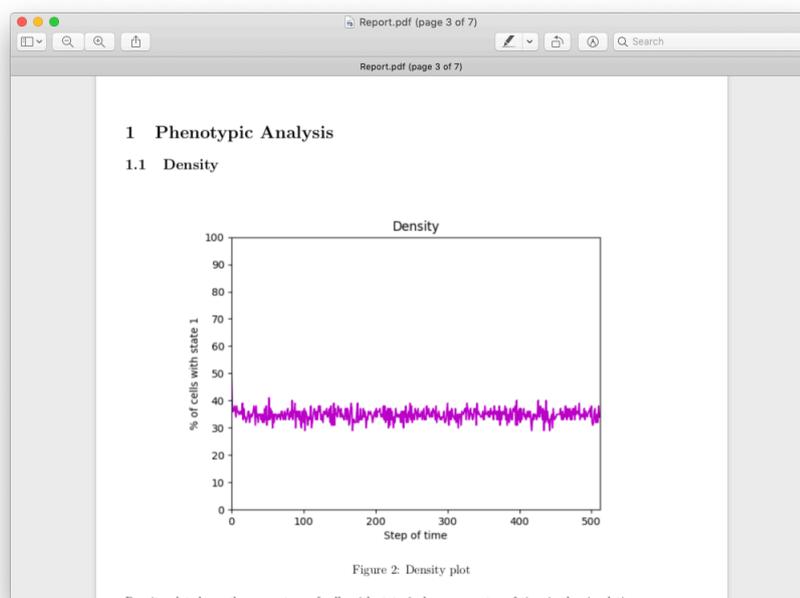


Figura 5.44: Reporte de análisis

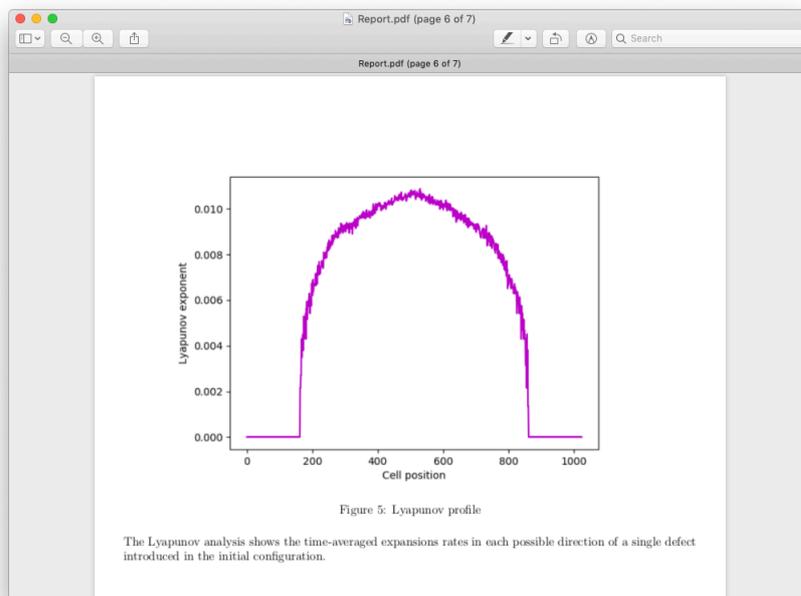


Figura 5.45: Reporte de análisis

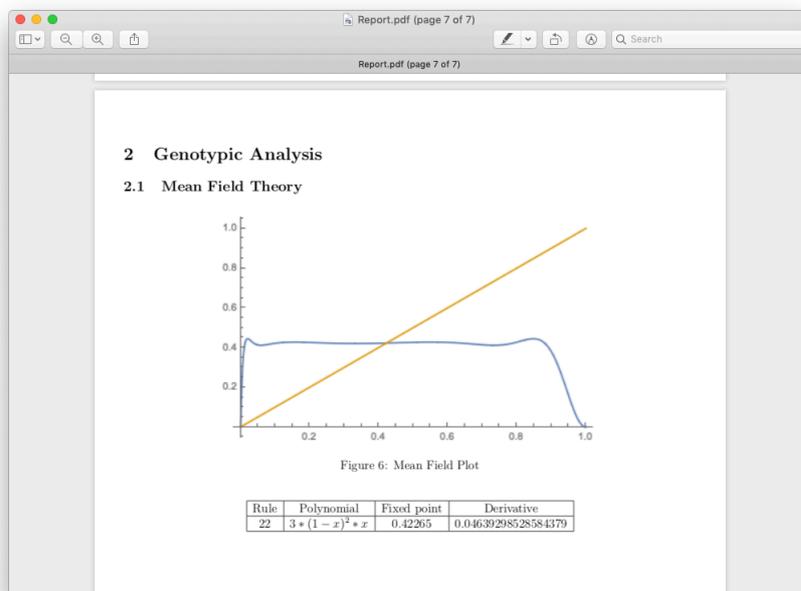


Figura 5.46: Reporte de análisis

Para generar patrones fractales en las simulaciones usamos la opción de configuración inicial a partir de una semilla.

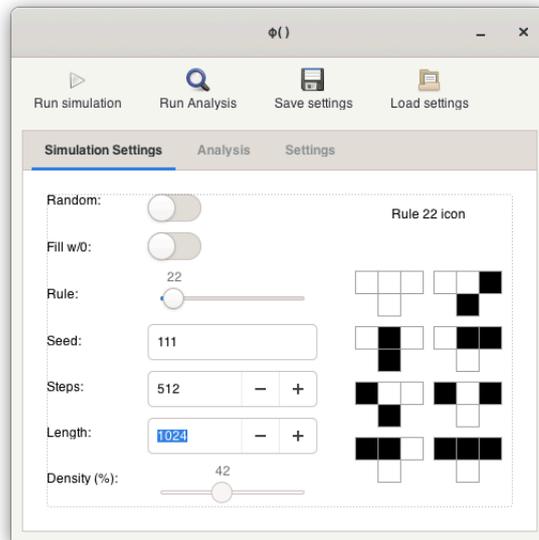


Figura 5.47: Ajustes para generar un patrón fractal

Al introducir una semilla, en caso de que sea menor a la longitud de la configuración inicial se rellenará con células en estado 0 o 1 (dependiendo de la opción elegida), de esta manera se tiene un espacio de evoluciones mayor y se puede visualizar mejor el comportamiento del sistema.

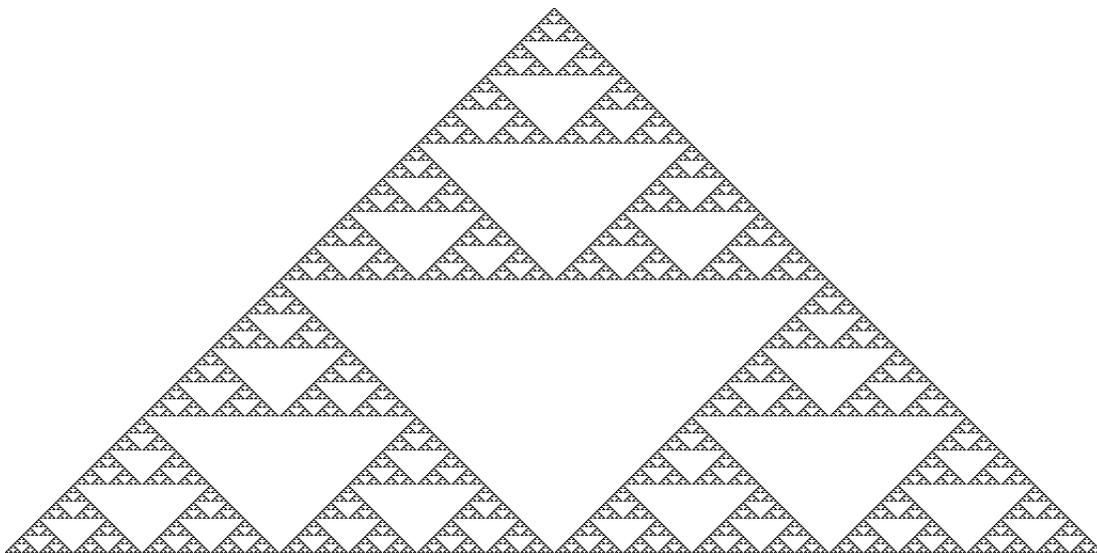


Figura 5.48: Triángulo de Sierpinski en un autómata celular elemental

Los ajustes de simulación pueden guardarse, de esta manera podemos registrar algunos de los patrones fractales encontrados durante la investigación.

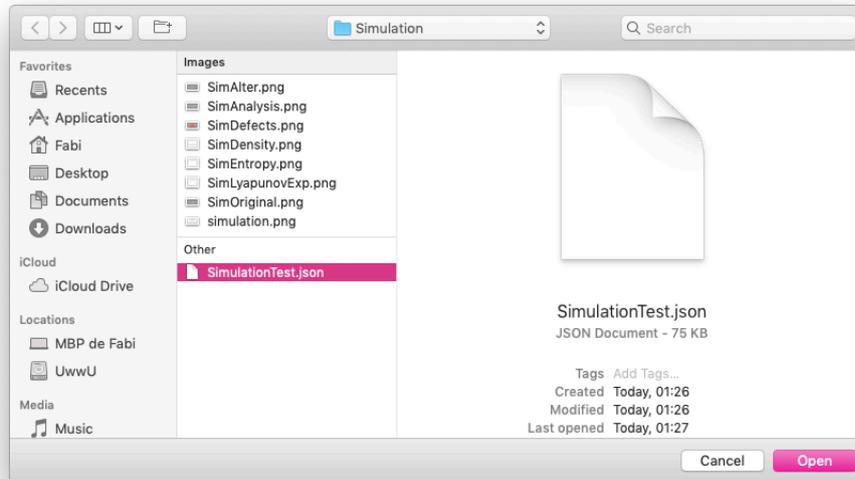


Figura 5.49

5.3. Conclusiones

Los objetivos planteados al inicio de este trabajo terminal eran:

- Desarrollar un simulador de autómatas celulares elementales que además nos permitiera hacer el análisis de su comportamiento caótico.
- Elaborar una caracterización de los autómatas que presentaran patrones fractales en sus evoluciones con la ayuda del simulador.

Observando los resultados del análisis generado por el simulador que describe el comportamiento morfológico, fenotípico y genotípico de los autómatas celulares elementales, elaboramos una caracterización que contempla los patrones fractales que aparecen en estos autómatas. Al revisar las caracterizaciones presentes en la literatura nos dimos cuenta que las clases no eran completas ya que había ambigüedad en estas. Mediante la caracterización que proponemos pretendemos mostrar que el conjunto de reglas fractales puede ser considerada una sub-clasificación de los autómatas celulares caóticos y que esta clasificación difiere de las clasificaciones existentes de autómatas celulares elementales caóticos.

5.4. Limitaciones

Las limitaciones de este trabajo las dividimos en dos partes, las primeras son las limitaciones teóricas y las segundas limitaciones técnicas. En cuanto a las limitaciones teóricas o metodológicas no hemos podido caracterizar los fractales no regulares, que son un conjunto más grande de los fractales regulares, además el análisis de los atractores se hizo a partir de una medida de entropía global, esta medida no refleja las principales características locales de cada grafo, siendo una de las características más importantes la distinción de las orbitas que producen fractales regulares, fractales irregulares o caos.

En el desarrollo del simulador Φ se utilizó el lenguaje de programación Python, este como todos los lenguajes de programación tiene sus ventajas y desventajas. Las principales ventajas por las que se optó usar este lenguaje es por su funcionalidad y su facilidad y herramientas disponibles para la manipulación y visualización de información que nos sería de ayuda para mostrar los resultados de los análisis realizados y poder mostrar resultados de manera intuitiva para la caracterización que se tiene como objetivo. Las desventajas de usar Python son que, al ser un lenguaje interpretado, se perdería velocidad al momento del cálculo de los resultados durante las simulaciones y análisis, esto es un obstáculo al querer realizar simulaciones en el que el espacio de evoluciones es mayor, a pesar de esto el simulador puede realizar grandes cálculos pero el tiempo que tardará en obtener resultados será considerablemente mayor. Existen algunas alternativas que nos pueden ayudar a solucionar estas limitaciones como utilizar Numba o Cython, Numba es un compilador e interprete que sirve para acelerar el uso de loops en Python que además puede trabajar con una Unidad de Procesamiento Gráfico (GPU), por otra parte Cython es un lenguaje con su propio compilador, este nos permite combinar Python y C al facilitar la codificación de extensiones en C para Python. Aunque estas soluciones eran viables para optimizar el código del simulador se debía reescribir gran parte del código y esto retrasaría el cronograma propuesto y tal vez no nos permitiría terminar por completo el desarrollo del simulador.

5.5. Trabajo futuro

En cuanto al trabajo futuro podemos tomar dos caminos, el primero es extender las capacidades computacionales del simulador por lo que debemos estudiar la viabilidad de optimizar el código del simulador Fi utilizando algunas de las tecnologías de las que se hablo en las limitaciones del trabajo, esto implicaría reescribir ciertas partes del código debido a que en las herramientas propuestas hay sintaxis muy especifica. También se puede trabajar en mas herramientas de análisis de Autómatas Celulares Elementales y hacer aún más intuitiva la interfaz de usuario.

La otra vía para el trabajo futuro es extender el análisis teórico de la caracterización Uno de los posibles caminos a seguir es explicar la capacidad generativa de los fractales a partir de las propiedades lógicas o algebraicas que tiene cada regla y que operan tanto en la semilla (condiciones iniciales) como en el espacio de evoluciones (evolución del autómata) y así por un lado relacionar las condiciones iniciales con el fractal producido y por el otro relacionar la producción de un fractal aleatorio como el resultado de alguna operación lógica entre fractales regulares.

Otro camino que puede explorarse es la capacidad generativa a través de herramientas de la teoría de Autómatas y lenguajes formales o dinámica simbólica y ver al atractor del fractal como un autómata o gramática que produzca el alfabeto de los fractales. Así también encontrar el conjunto de las configuraciones jardines del Edén , que son las palabras prohibidas, y caracterizar las reglas a partir de estos conjuntos. Quizá estos abordajes nos acerquen a resolver los siguientes problemas.

¿Existe alguna función o conjunto de funciones $F(X)$ que produzcan los fractales y que sean independientes de las reglas elementales de los autómatas celulares?. Si existen se puede explicar porqué las cuencas de atractores son parecidas **fenotípicamente** a pesar que produce Mosaicos diferentes.

Existe alguna caracterización completa que describa y explique los fractales regulares e irregulares en los autómatas celulares. Si es posible entonces también es posible establecer

una medida de distancia d que nos explique qué tan cercano está algún fractal de otro o que tan cercano está una regla de otra regla.

Apéndice A

Tablas y Figuras

A.1. Tablas

Índice de cuadros

1.1. Ejemplos de la clasificación de Wolfram. 1. Autómata celular Clase I. 2. Autómata celular Clase II . 3. Autómata celular Clase III . 4. Autómata celular Clase IV.	13
2.1. Ejemplo de la regla de evolución local. Se produce la regla 90	25
2.2. Espacio de evoluciones de la regla 90 con una configuración inicial formada de una única celda en uno [21]	26
2.3. Iteraciones del triangulo de Sierpinski	30
3.1. Reglas fractales	34
3.2. Mosaico MS_1	38
3.3. Regla del mosaico MS_1	39
3.4. Mosaico $\overline{MS_1}$	39
3.5. Regla del mosaico $\overline{MS_1}$	40
3.6. Reglas equivalentes mosaico MS_1	40
3.7. Mosaico MS_2	42
3.8. Mosaico $\overline{MS_2}$	42
3.9. Regla del mosaico MS_2	43
3.10. Mosaico MS_3	44
3.11. Mosaico $\overline{MS_3}$	44
3.12. Regla del mosaico MS_3	45
3.13. Mosaico MS_4	46

3.14. Mosaico \overline{MS}_4	46
3.15. Regla del mosaico MS_4	47
3.16. Mosaico MS_5	48
3.17. Mosaico \overline{MS}_5	48
3.18. Regla del mosaico MS_5	49
3.19. Mosaico MS_6	50
3.20. Mosaico \overline{MS}_6	50
3.21. Regla del mosaico MS_6	51
3.22. Mosaico MA_1	51
3.23. Regla del mosaico MA_1	52
3.24. Mosaico MA_2	52
3.25. Regla del mosaico MA_2	53
3.26. Mosaico MA_1	53
3.27. Regla del mosaico MC_1	54
3.28. Reglas fractales	55
3.29. Comparación de las reglas fractales y la caracterización por los exponentes de Lyapunov	65
3.30. Característica 1	68
3.31. Característica II	68
3.32. Característica III	68
3.37. Caracterización de las Reglas fractales de acuerdo a la teoría del campo promedio	72
3.38. Reglas fractales	84
4.1. Comparación entre las caracterizaciones propuestas para las reglas fractales	88
4.2. Reglas fractales	89
4.3. Reglas fractales	89
4.4. Reglas fractales	90

4.5. Reglas fractales	90
4.6. Reglas fractales	91
4.7. Reglas fractales	91
4.8. Reglas fractales	92
4.9. Clasificación de Wolfram	93
4.10. Surface dynamics classification	93
4.11. Clasificación de Wuensche	94
4.12. Clasificación Expressivity analysis	94
4.13. Clasificación de Li y Packard	95
4.14. Clasificación Topological dynamics classification	96
4.15. Clasificación Topological dynamics classification	96
4.16. Clasificación Topological classification	97
4.17. Clasificación Creativity classification	97
4.18. Comparación con clasificaciones fenotípicas	98
4.19. Comparación con clasificaciones genotípicas	99
5.1. Cronograma de actividades propuesto para el TT	104

A.2. Imágenes

Índice de figuras

2.1.	1) Evolución en una dimensión de la regla 154. 2) Evolución en dos dimensiones de un glider en la regla de Difusión $R(7722)$. 3) Evoluciones en tres dimensiones de la regla $R(4, 5, 5, 5)$	23
2.2.	Orbita de la regla 90 con un anillo de ocho células	27
2.3.	Ejemplos de dimensión de auto similitud. 1. La línea es de dimensión uno, 2. El cuadrado es de dimensión dos, 3. El cubo es de dimensión tres	29
3.1.	Fractal Regular, Regla 126 . Semilla 1	35
3.2.	Fractal Irregular, Regla 126 . Semilla 0000100011	35
3.3.	Lista de mosaicos regulares en automatas celulares elementales	37
3.4.	Primera iteración del Mosaico $M1$	38
3.5.	Primera iteración del Mosaico $\overline{MS_1}$	39
3.6.	Atractor de la regla 90 y longitud 8 Atractor de la regla 90 y longitud 16	41
3.7.	Atractores para el mosaico $M1$ y longitud 16.	41
3.8.	Primera iteración del Mosaico MS_2 y $\overline{MS_2}$	42
3.9.	Atractor de la regla 60 y longitud 8 Atractor de la regla 60 y longitud 16	43
3.10.	Primera iteración del Mosaico MS_3 y $\overline{MS_3}$	44
3.11.	Atractor de la regla 22 y longitud 8 Atractor de la regla 22 y longitud 16	45
3.12.	Primera iteración del Mosaico MS_4 y $\overline{MS_4}$	46
3.13.	Atractor de la regla 126 y longitud 8 Atractor de la regla 126 y longitud 16	47
3.14.	Primera iteración del Mosaico MS_5 y $\overline{MS_5}$	48

3.15. Atractor de la regla 146 y longitud 8 Atractor de la regla 146 y longitud 16	49
3.16. Primera iteración del Mosaico MS_6 y $\overline{MS_6}$	50
3.17. Primera iteración del Mosaico MA_1 y $\overline{MA_1}$	52
3.18. Primera iteración del Mosaico MA_2 y $\overline{MA_2}$	53
3.19. Primera iteración del Mosaico MC_1 y $\overline{MC_1}$	54
3.20. Polinomio del campo promedio de la regla 90, iterado tres veces	67
3.21. 1) Único atractor de la regla 60 con periodo 1. 2) Único atractor de la regla 90 con periodo 1.	75
3.22. 1) Atractor de la regla 94 con periodo uno. 1) Atractor de la regla 164 con periodo uno.	76
3.23. 1) Atractor de la regla 105 con periodo uno. 1) Atractor de la regla 150 con periodo uno	77
3.24. Atractores para el mosaico M1	78
3.25. Atractores de la clase 5	80
3.26. Atractores para el mosaico M1	82
5.1. Modelo Vista Controlador	106
5.2. Clase Bitstring	108
5.3. Clase ECA	108
5.4. Clase Simulation	108
5.5. Clase SimulationTab	109
5.6. Clase AnalysisTab	109
5.7. Clase SettingsTab	110
5.8. Clase MainWindow	110
5.9. Clase FiApp	111
5.10. Módulos de Python Plotter y Files	112
5.11. Pestaña principal	115
5.12. Selección de configuración aleatoria	116

5.13. Selección de estado para las células de relleno	117
5.14. Ejemplo de simulación a partir de una semilla	118
5.15. Ejemplo de simulación a partir de una semilla	118
5.16. Ejemplo de simulación con una configuración inicial aleatoria	119
5.17. Ejemplo de simulación con una configuración inicial aleatoria	119
5.18. Opciones para el análisis	120
5.19. Análisis de una regla	121
5.20. Reporte generado	122
5.21. Reporte generado	122
5.22. Reporte generado	123
5.23. Reporte generado	123
5.24. Ajustes de visualización	124
5.25. Seleccionar color	124
5.26. Seleccionar directorio	125
5.27. Ajustes de simulación guardados	126
5.28. Ajustes de simulación cargados a partir de un archivo guardado previamente	126
5.29. Selección de un directorio diferente para guardar los resultados.	127
5.30. Ajustes de visualización.	127
5.31. Ajustes para realizar una simulación.	128
5.32. Resultado de la simulación.	128
5.33. Ejemplo de simulación para la caracterización.	129
5.34. Ajustes de análisis.	129
5.35. Evolución original	130
5.36. Propagación del daño	130
5.37. Evolución original	131
5.38. Evolución de configuración alterada	131
5.39. Perfil de Lyapunov	132
5.40. Gráfica de entropía topológica	132

5.41. Gráfica de densidad	133
5.42. Gráfica del campo promedio	133
5.43. Reporte de análisis	134
5.44. Reporte de análisis	134
5.45. Reporte de análisis	135
5.46. Reporte de análisis	135
5.47. Ajustes para generar un patrón fractal	136
5.48. Triángulo de Sierpinski en un autómata celular elemental	136
5.49.	137

A.3. Índice

Índice alfabético

Autómata celular, 22

Autómata celular elemental, 24

Autómatas celulares elementales, 24

Dimensión de auto similitud, 28

Entropía, 58

Espacio de fase, 18

Espacio métrico, 18

Exponentes de Lyapunov, 56

Fractal, 28

Fractal irregular, 35

Fractal regular, 35

Función de evolución local, 24

Orbita, 19

Punto fijo, 20

Sistemas caóticos, 20

Sistemas dinámicos continuos, 18

Sistemas dinámicos discretos, 18

Teoría del campo promedio, 66

Triángulo de Sierpinski, 29

Apéndice B

Bibliografía

- [1] A. Adamatzky y G. Martínez. “On generative morphological Diversity of elementary cellular automata”. En: *Kybernetes* 39.1 (2010).
- [2] A. Adamatzky y A. Wuensche. “On Creativity of Elementary Cellular Automata”. En: *Complex Systems* 22.4 (2013).
- [3] *An Introduction To Chaotic Dynamical Systems*. Addison-Wesley, 1989.
- [4] Y. Bar-Yam. *Dynamics of Complex Systems*. Addison-Wesley, 1997.
- [5] M. Brin y G. Stuck. *Introduction to Dynamical Systems*. Cambridge University Press, 2002.
- [6] G. Cattaneo, M. Finelli y L Margara. “Investigating topological chaos by elementary cellular automata dynamics”. En: *Theoretical Computer Science* 244.1 (2000), págs. 219-241.
- [7] L. chua y col. “A NONLINEAR DYNAMICS PERSPECTIVE OF WOLFRAM’S NEW KIND OF SCIENCE PART VII: ISLES OF EDEN”. En: *International Journal of Bifurcation and Chaos* 17.09 (2007), págs. 2839-3012.
- [8] K. Culik y S. Yu. “Undecidability of CA Classification Schemes”. En: *Complex Systems* 2 (1988).

- [9] U. Díaz. “¿Cuántos fractales hay?” Bachelor’s Thesis. Facultad de ciencias, Universidad Nacional Autónoma de México., 2011.
- [10] R. Farhoodi y col. “Development of scientific software: A systematic mapping, a bibliometrics study, and a paper repository”. En: *International Journal of Software Engineering and Knowledge Engineering* 23 (jul. de 2013).
- [11] D. Feldman. *Exploring the Sierpiński triangle*. Last accessed 4 May 2019. URL: <https://www.youtube.com/watch?v=4I-mvjYt12k>.
- [12] H. Gutowitz. “Mean Field vs. Wolfram Classification of Cellular Automata”. En: (1989).
- [13] K. Culik y S. Dube. “Fractal and Recurrent Behavior of Cellular Automata”. En: *Complex Systems* 3.3 (1989).
- [14] J. King y H. Méndez. *Sistemas dinámicos discretos*. Universidad Nacional Autónoma de México, 2014.
- [15] W. Li y N. Packard. “The Structure of the Elementary Cellular Automata Rule Space”. En: *Complex Systems* 4 (1990).
- [16] B. Mandelbrot. “Fractals: Form, Chance and Dimension”. En: *Journal of Fluid Mechanics* 92.1 (1979), págs. 206-208. DOI: 10.1017/S0022112079210586.
- [17] B. Mandelbrot. *The fractal geometry of nature*. Freeman, 1982.
- [18] B. Martin. “Inherent Generation of Fractals by Cellular Automata”. En: *Complex Systems* 8.5 (1994).
- [19] G. Martínez. “A Note on Elementary Cellular Automata Classification”. En: *J. Cellular Automata* 8.3-4 (2013), págs. 233-259.
- [20] G. Martínez. “Procedimiento para producir comportamientos complejos en regla 110”. Tesis doct. Centro de Investigación y de Estudios Avanzados del IPN, 2006.

- [21] G. Martínez. “Teoría del campo promedio en autómatas celulares similares a ”The game of life””. Tesis de maestría. Centro de Investigación y de Estudios Avanzados del IPN, 2000.
- [22] G. Martínez, A. Adamatzky y R. Alonso-Sanz. “Designing Complex Dynamics in Cellular Automata with Memory”. En: *J. Bifurcation and Chaos* 23.10 (2013).
- [23] H. McIntosh. *One Dimensional Cellular Automata*. Luniver Press, 2009.
- [24] M. Mitchell. *Complexity: a guided tour*. Oxford University Press, Inc., 2009.
- [25] John von Neumann. *Theory of self-reproducing automata*. University of Illinois Press, 1966.
- [26] J. Ramirez. “Introducción a la geometría fractal y a la dinámica del caos”. Bachelor’s Thesis. Facultad de ciencias, Universidad Nacional Autónoma de México., 2002.
- [27] M. Redeker, A. Adamatzky y G. Martínez. “Expressiveness of Elementary Cellular Automata”. En: *International Journal of Modern Physics C* 24.3 (2013).
- [28] M. Schüle y R. Stoop. “A Full Computation-relevant Topological Dynamics Classification of Elementary Cellular Automata”. En: *Chaos (Woodbury, N. Y.)* 22 (dic. de 2012), pág. 043143.
- [29] R. Solé y S. Manrubia. *Orden y caos en sistemas complejos*. UPC, 2001.
- [30] J. Seck Tuoh. “Análisis dinámico y topológico de los autómatas celulares unidimensionales reversibles”. Tesis doct. Centro de Investigación y de Estudios Avanzados del IPN, 2002.
- [31] J. Seck Tuoh y col. “Emergence of density dynamics by surface interpolation in elementary cellular automata”. En: *Communications in Nonlinear Science and Numerical Simulation* 19.4 (2014), págs. 941-966.
- [32] S. Willson. “Cellular automata can generate fractals”. En: *Discrete Applied Mathematics* 8.1 (1984), págs. 91-99.

- [33] S. Wolfram. “Cellular Automata as Simple Self-Organizing Systems”. En: *Caltech Preprint* (1982).
- [34] S. Wolfram. “Statistical Mechanics of Cellular Automata”. En: *Reviews of Modern Physics* 55.3 (1983).
- [35] S. Wolfram. “Twenty Problems in the Theory of Cellular Automata”. En: *Physica Scripta* T9 (1985), págs. 170-183.
- [36] A. Wuensche y M. Lesser. *The global dynamics of Cellular Automata*. Addison-Wesley Publishing Company, 1992.