



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO

Reporte Final.- Atractores y Expresiones Regulares para la  
regla 22 y 54

Unidad de aprendizaje: Computing Selected Topics

Grupo: 3CM19

*Alumno:*  
Sanchez Mendez Edmundo Josue

*Profesor:*  
Juarez Martinez Genaro

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Atractores . . . . .	3
1.2. Expresiones regulares . . . . .	3
1.3. Entropía de Shannon . . . . .	3
<b>2. Expresiones Regulares</b>	<b>4</b>
2.1. Regla 22 . . . . .	4
2.1.1. Probabilidad del 50 % de unos . . . . .	4
2.1.2. Probabilidad del 95 % de unos . . . . .	7
2.1.3. Generación de condición inicial completamente aleatoria . . . . .	10
2.2. Regla 54 . . . . .	12
2.2.1. Probabilidad del 50 % de unos . . . . .	13
2.2.2. Probabilidad del 95 % de unos . . . . .	16
2.2.3. Generación de condición inicial completamente aleatoria . . . . .	18
<b>3. Atractores</b>	<b>21</b>
3.1. Frontera abierta . . . . .	21
3.2. Frontera reflectora . . . . .	21
<b>4. Programa</b>	<b>22</b>
4.1. Descripción . . . . .	22
4.2. Pruebas . . . . .	23
4.3. Añadidos . . . . .	36
4.4. Código . . . . .	39
<b>5. Conclusiones</b>	<b>39</b>
<b>Referencias</b>	<b>39</b>

## Resumen

Para este reporte abarcaremos la generación de atractores y generación de cadenas aleatorias con base en las expresiones regulares para los autómatas celulares elementales con regla 22 y 54. Nuestro objetivo es poder examinar el comportamiento de estas reglas con base en sus atractores y expresiones regulares, como en el siguiente ejemplo.

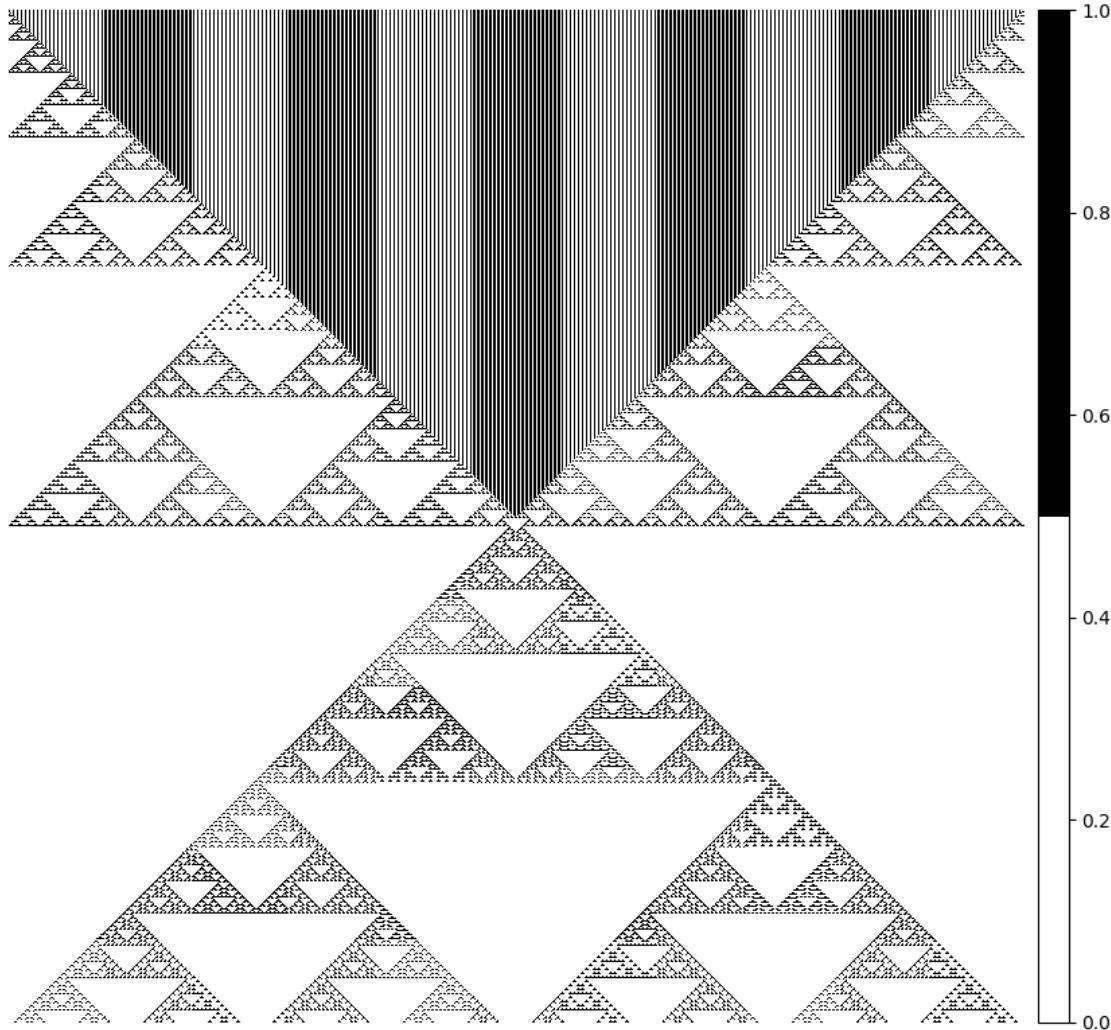


Figura 1: Autómata celular elemental de 1000 células evolucionado 1000 veces, regla 22 usando su expresión regular.

## 1. Introducción

### 1.1. Atractores

Los atractores son un conjunto de valores numéricos hacia los cuales un sistema tiende a evolucionar, dada por una variedad de condiciones iniciales en el sistema, existen tres tipos de atractores que son los siguientes:

- Atractor de punto fijo: El sistema que tenga un atractor de punto fijo tenderá a estabilizarse en un único punto.
- Atractor de ciclo límite o atractor periódico: Este tipo de atractor tiende a mantenerse en un periodo o en un mismo ciclo.
- Atractor caótico: Aparece en sistemas no lineales que tienen una gran sensibilidad a las condiciones. Un famoso ejemplo de estos atractores es el atractor de Lorenz.

### 1.2. Expresiones regulares

Las expresiones regulares nos permiten describir el lenguaje aceptado por un autómata finito, en nuestro caso buscamos ver como el uso de la expresiones regulares que describen a nuestras reglas afectan a la generación de autómatas celulares cuando se usan como condición inicial.

### 1.3. Entropía de Shannon

La entropía de Shannon es una de las métricas más importantes en la teoría de la información. La entropía mide la incertidumbre asociada con una variable aleatoria, es decir, el valor esperado de la información en el mensaje (en informática clásica se mide en bits).

El concepto fue introducido por Claude E. Shannon en el artículo “A Mathematical Theory of Communication” (1948). La entropía de Shannon permite estimar el número mínimo promedio de bits necesarios para codificar una cadena de símbolos en función del tamaño del alfabeto y la frecuencia de los símbolos y viene dada por la siguiente formula:

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i)$$

En donde b es la base del logaritmo utilizado. Los valores comunes de b son 2, el número de Euler e y 10, y las unidades correspondientes de entropía son los bits para b = 2 , nats para b = e y bans para b = 10 y  $P(x_i)$  como la probabilidad de un evento, para nuestro caso de estudio b=2.

La situación de máxima incertidumbre nos genera que sea más difícil predecir el resultado siguiente, esto es debido a que hay probabilidad uniforme. La entropía, entonces, solo puede disminuir a partir del valor asociado con la probabilidad uniforme. El caso extremo es donde un estado tiene probabilidad de 1 ya que ahí no hay incertidumbre, es decir la entropía es cero.

## 2. Expresiones Regulares

### 2.1. Regla 22

La expresión regular que nos describe a la regla 22 es el siguiente:

$$(0 + 1(01)^*00)(0 + 0(01)^*00)^*$$

La expresión regular anterior nos permite poder generar cadenas aleatorias que nos servirán como condiciones iniciales para la generación de nuestros autómatas celulares elementales. Para poder visualizar como el uso de nuestra expresión regular afecta al comportamiento de nuestro autómata haremos las siguientes pruebas:

- Probabilidad del 50 % de unos.
- Probabilidad del 95 % de unos.
- Generación de condición inicial completamente aleatoria.

Estas serán comparadas con cadenas aleatorias generadas mediante la expresión regular, todo esto se hará en un espacio de 400x400 células y se compararán tanto la imagen que nos genera el autómata así como con diferentes métricas que el programa nos proporciona.

#### 2.1.1. Probabilidad del 50 % de unos

En la figura 2 podemos ver los valores de entrada en nuestro programa, en donde definimos un espacio de 400 x 400 células con una probabilidad de unos del 50 %

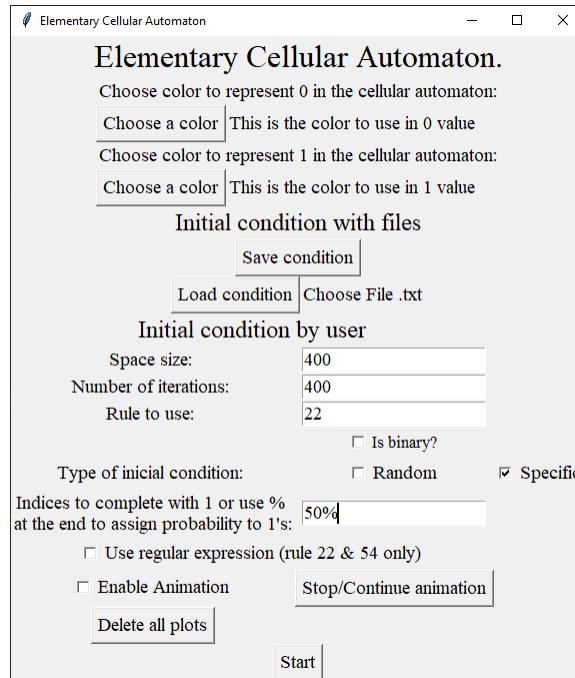


Figura 2: Entrada de nuestro programa para la primera prueba.

En la figura 3 vemos el autómata resultante así como las métricas que el programa nos genera.

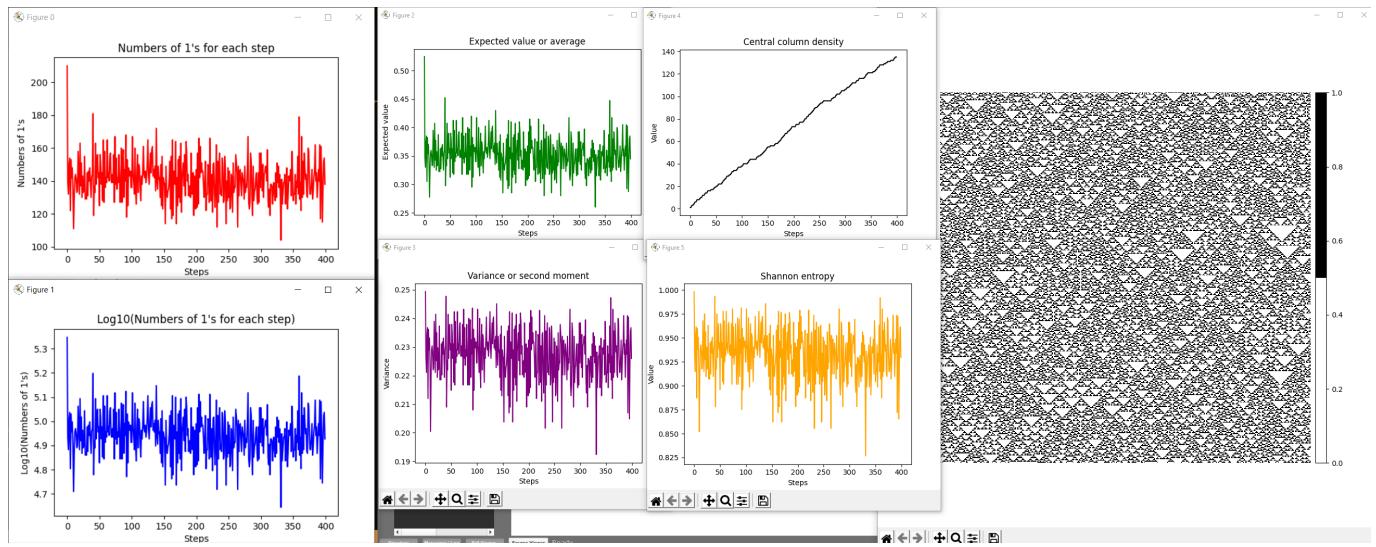


Figura 3: Autómata resultante con sus respectivas métricas.

En la figura 4 de igual manera que con la figura 2 podemos ver los valores de entrada de nuestro programa, en donde lo único que cambia es que en esta ocasión hacemos uso de la expresión regular de nuestra regla media la selección de la casilla con la etiqueta “Use regular expression (rule 22 & 54 only)”

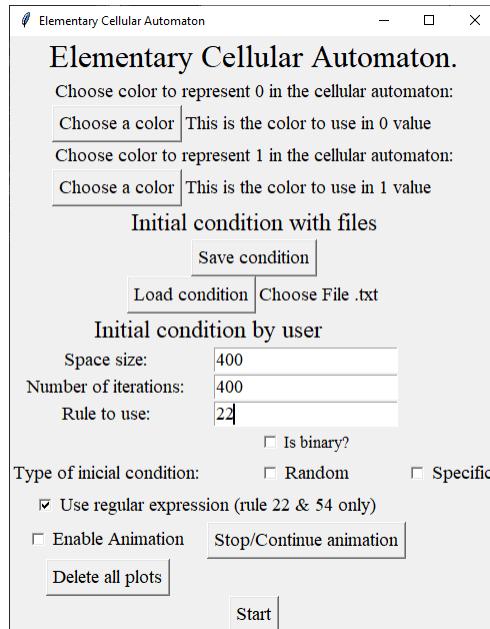


Figura 4: Entrada de nuestro usando nuestra expresión regular r22.

En la figura 5 vemos el autómata resultante así como las métricas que el programa nos genera.

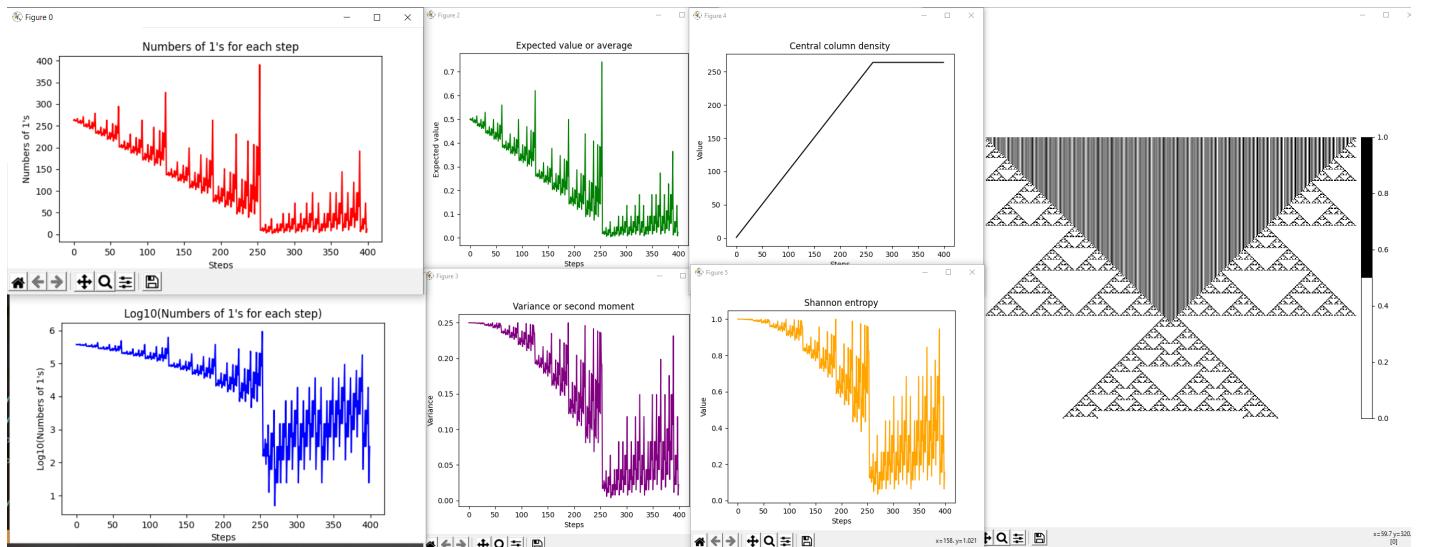


Figura 5: Autómata resultante con sus respectivas métricas.

Antes de empezar con el estudio de estos dos autómatas recordar que la expresión regular se utiliza para poder generar condiciones iniciales aleatorias en las cuales tenemos un control hasta cierto punto, por lo que el autómata mostrado en la figura 5 puede variar en forma y en consecuencia en sus métricas si los comparamos con los autómatas que se generaran en las dos pruebas restantes, una vez mencionado lo anterior, comparemos los resultados.

Para nuestro autómata con probabilidad del 50 % de unos en la condición inicial vemos como 5 de nuestras 6 gráficas tienen una silueta muy similar entre ellas, vemos como el numero de unos por paso se mantienen entre los valores de 160 y 120 en donde ocasionalmente hay pasos en donde se salen de esos valores, pero nunca llegan a superar a la primera iteración en donde el numero de pasos fue de un poco mas de 200, si vemos la entropía de Shannon vemos que el valor mas alto fue de 1 y el mas bajo de 0.825 esto nos quiere decir que tenemos casi una probabilidad uniforme, esto lo podemos afirmar ya que para poder obtener el valor máximo de la entropía de shannon necesitamos probabilidades iguales, de forma general podemos decir que tenemos una gran incertidumbre y nos vemos incapaces de poder predecir el próximo estado que tomara nuestro autómata, examinando nuestra gráfica de la densidad en la columna central vemos como nos recuerda a una linea recta, es decir su crecimiento es lineal, aunque hay ciertos puntos en donde la densidad fue constante y no tenemos un patrón reconocible en nuestro autómata.

Por otro lado tenemos al autómata generado por expresión regular y a simple vista podemos observar que es un caso completamente diferente al primero ya que podemos observar cierto patrón en nuestro autómata, el cual pareciese ser la formación de un triangulo isósceles de cabeza y a sus lados patrones formados de triángulos incrustados dentro de un triangulo superior que pareciese que avanzan hasta un punto de unión formando un triangulo diferente. Observado las métricas que tenemos disponibles podemos afirmar que hay un patrón en nuestro autómata ya que las mediciones parecen repetirse por determinados ciclos, podemos ver como el numero de unos va disminuyendo a como pasan las iteraciones y por ello vemos como la densidad de la columna central llega a un máximo de 265 células con valor 1 para pasar a ser constante, vemos a la entropía iniciar con el máximo valor que puede tomar en nuestro caso de estudio a tocar el valor mas bajo, esto quiere decir que en nuestro sistema pasamos de ser incapaces de predecir el siguiente estado a poder predecir el siguiente estado de nuestro autómata.

En resumen vemos como la expresión regular en este caso nos genera un patrón muy claro en nuestro autómata y como esto influye tanto en la generación de nuestro autómata como en sus métricas en especial caso podemos ver como la entropía de nuestros dos casos de estudio son muy distintos ya que pasamos

de tener una entropía casi máxima a tener una entropía casi mínima, es decir pasamos de tener una gran incertidumbre a prácticamente no tener incertidumbre.

### 2.1.2. Probabilidad del 95 % de unos

En la figura 6 podemos ver los valores de entrada en nuestro programa, en donde definimos un espacio de 400 x 400 células con una probabilidad de unos del 95 %

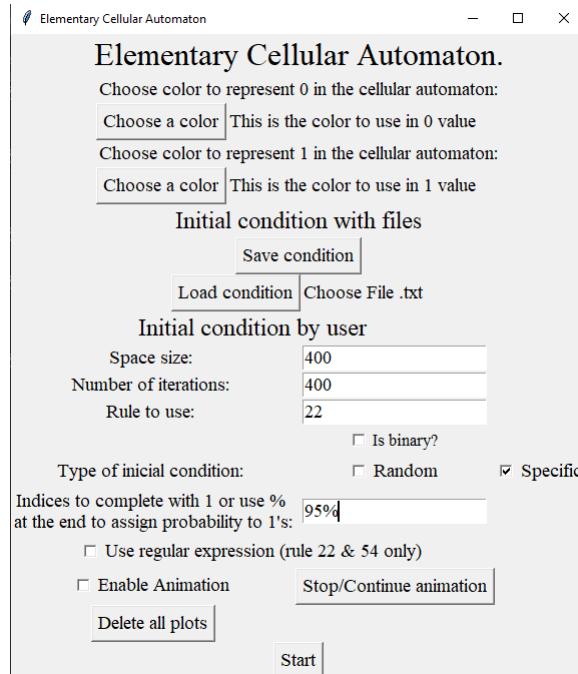


Figura 6: Entrada de nuestro programa para la segunda prueba.

En la figura 7 vemos el autómata resultante así como las métricas que el programa nos genera.

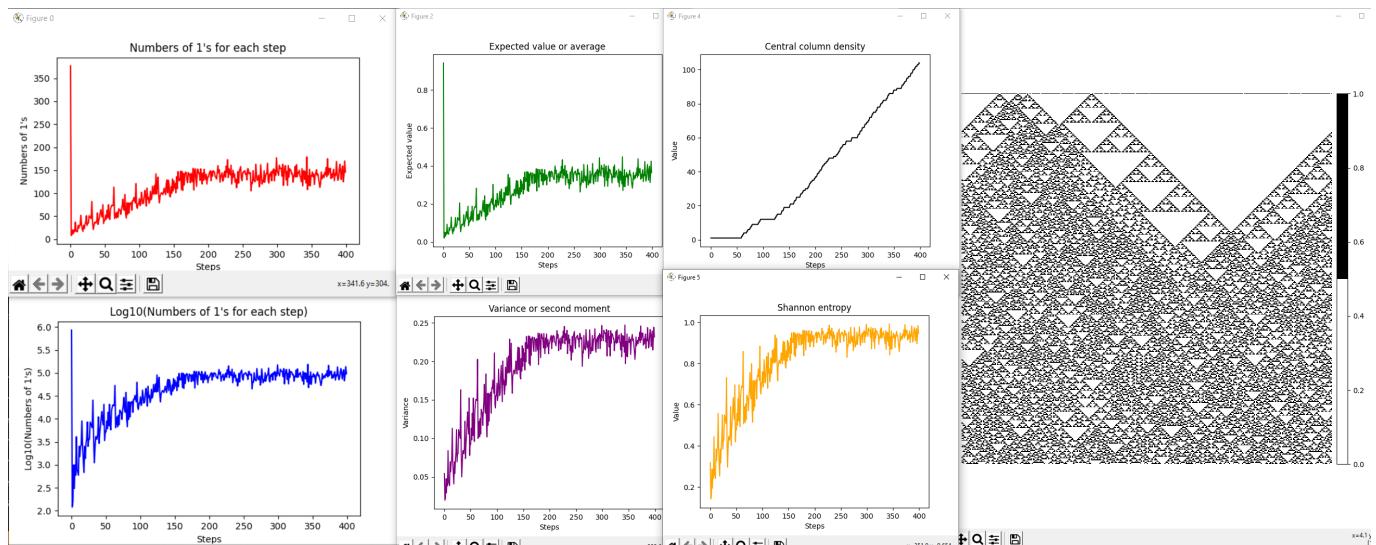


Figura 7: Autómata resultante con sus respectivas métricas.

En la figura 8 de igual manera que con la figura 6 podemos ver los valores de entrada de nuestro programa, en donde lo único que cambia es que en esta ocasión hacemos uso de la expresión regular de nuestra regla media la selección de la casilla con la etiqueta “Use regular expression (rule 22 & 54 only)”

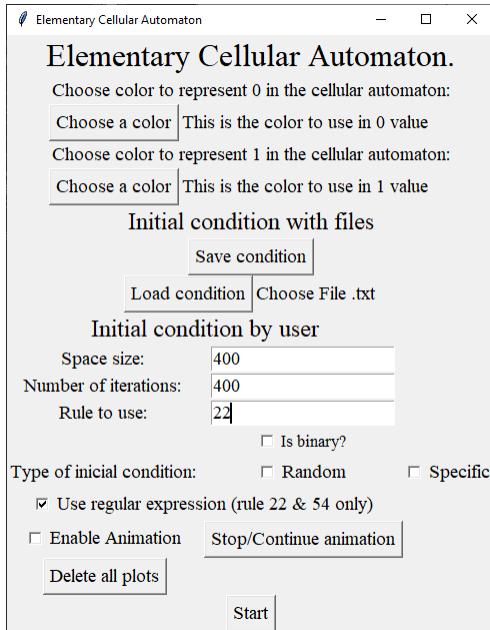


Figura 8: Entrada de nuestro usando nuestra expresión regular r22.

En la figura 9 vemos el autómata resultante así como las métricas que el programa nos genera.

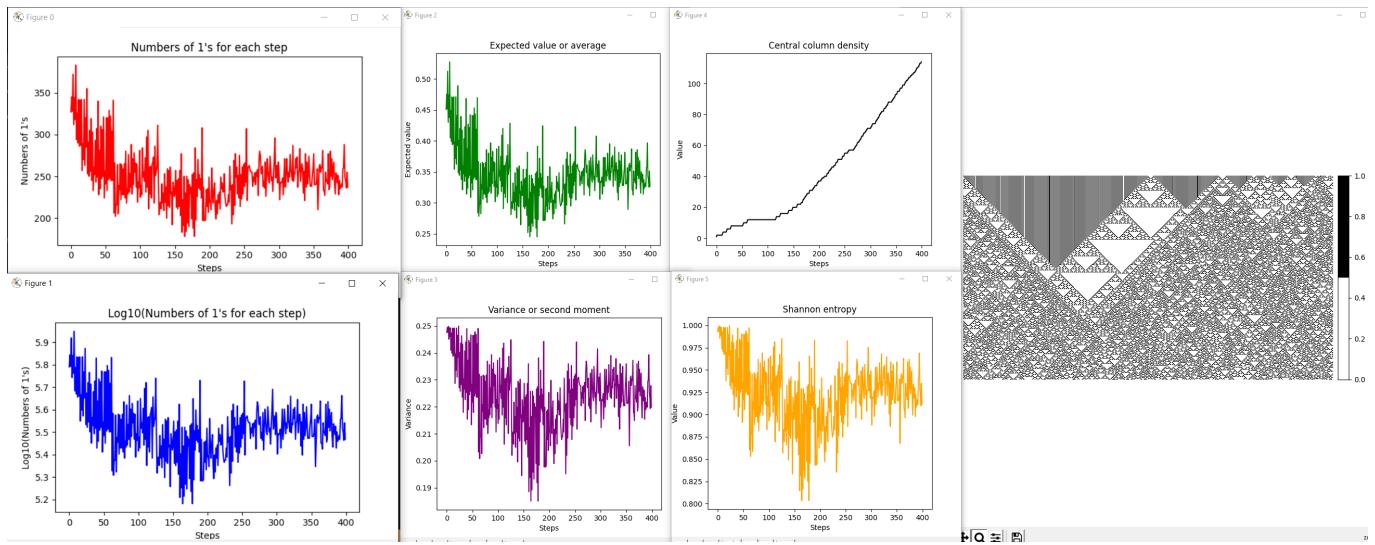


Figura 9: Autómata resultante con sus respectivas métricas.

Para nuestro autómata con probabilidad del 95 % de unos en la condición inicial vemos como el numero de unos, el  $\log_{10}$  y el valor esperado tienen una silueta muy similar entra ellas de igual forma la varianza con la entropía de Shannon, vemos como el numero de unos por paso tiene un máximo al inicio con un valor de 377 a tener un bajo en la primera iteración teniendo como valor de 9, después de este máximo y mínimo los unos tienden a crecer de una forma logarítmica ya que parece tener una asíntota horizontal en el valor 150, si vemos la entropía de Shannon podemos ver de manera clara como

al inicio tenemos el valor mínimo de entropía, es decir no tenemos incertidumbre es prácticamente nula, sin embargo, con el paso de las iteraciones esta incertidumbre va aumentando hasta llegar prácticamente al punto máximo de entropía, es decir tenemos una gran incertidumbre y nos vemos incapaces de poder predecir el próximo estado que tomara nuestro autómata, examinando nuestra gráfica de la densidad en la columna central vemos como es constante el numero de 1's hasta la iteración 58 ya que desde la siguiente iteración comienza a ver un crecimiento escalonado, finalizando el análisis de este autómata podemos decir que no tenemos un patrón reconocible, sin embargo, podríamos decir que al inicio teníamos formas triangulares inscritas en un triángulo mayor pero en un punto estas figuras se perdieron.

Por otro lado tenemos al autómata generado por expresión regular y a simple vista podemos observar como comparte un patrón con el autómata de la figura 5 y es que vemos como en una parte del autómata tenemos líneas rectas que nos forman un triángulo invertido y vemos como las gráficas no se parecen en nada a la prueba de 95 % de unos. Observado las métricas que tenemos disponibles observamos una gran variación en la información que nos da el autómata y es algo completamente diferente a la prueba que estudiamos ya que en esa prueba vemos como tienden a un valor, en cambio, en este autómata podemos ver como el numero de unos va disminuyendo a como pasan las iteraciones pero no es una disminución constante si no que es como saltado y no es una reducción continua, sin embargo, vemos como la densidad de la columna central va aumentando de una forma casi lineal, pero volvemos a ver como hay puntos en donde el valor no cambia pero en su mayoría son lapsos muy pequeños, pero en la entropía podemos ver como inicia prácticamente en el máximo valor que puede tomar y después de 16 iteraciones empieza a disminuir y oscilar entre los valores 0.8 y 0.975, esto quiere decir que en nuestro sistema dentro de todas las iteraciones nos vemos incapaces de predecir el siguiente estado nuestro autómata y tenemos una alta incertidumbre.

En resumen vemos como la expresión regular en este caso no nos genera un patrón muy claro en este autómata, sin embargo, vemos como comparte un pequeño patrón con el autómata de la figura 5, así vemos como la expresión regular sigue influyendo en la generación de nuestro autómata. En esta ocasión volvemos a ver como las métricas no coinciden, sin embargo, podemos ver como hay partes de nuestro autómata que se parecen un poco pero que nos proporcionan información distinta.

### 2.1.3. Generación de condición inicial completamente aleatoria

En la figura 10 podemos ver los valores de entrada en nuestro programa, en donde definimos un espacio de 400 x 400 células haciendo uso de la función random de la librería numpy de Python el cual nos generara un arreglo de determinado tamaño, en este caso de 400 células, lleno aleatoriamente de números 0 y 1.

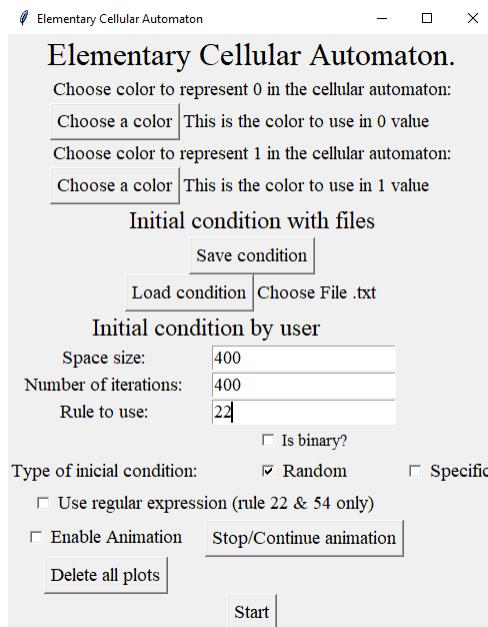


Figura 10: Entrada de nuestro programa para la tercera prueba.

En la figura 11 vemos el autómata resultante así como las métricas que el programa nos genera.

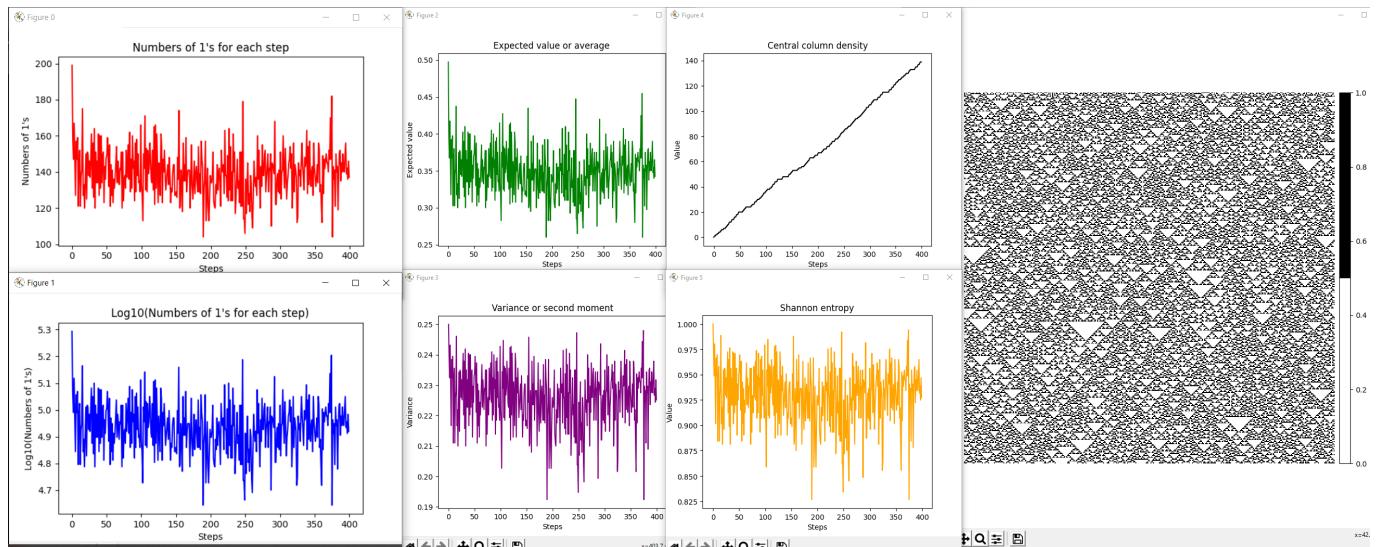


Figura 11: Autómata resultante con sus respectivas métricas.

En la figura 12 de igual manera que con la figura 10 podemos ver los valores de entrada de nuestro programa, en donde lo único que cambia es que en esta ocasión hacemos uso de la expresión regular de nuestra regla media la selección de la casilla con la etiqueta “Use regular expression (rule 22 & 54 only)”

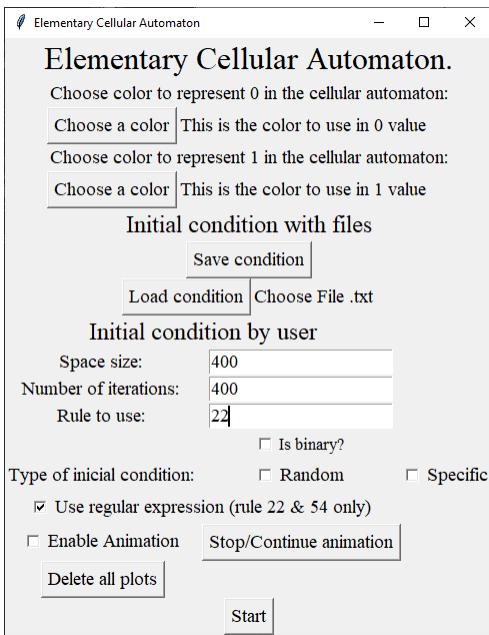


Figura 12: Entrada de nuestro usando nuestra expresión regular r22.

En la figura 13 vemos el autómata resultante así como las métricas que el programa nos genera.

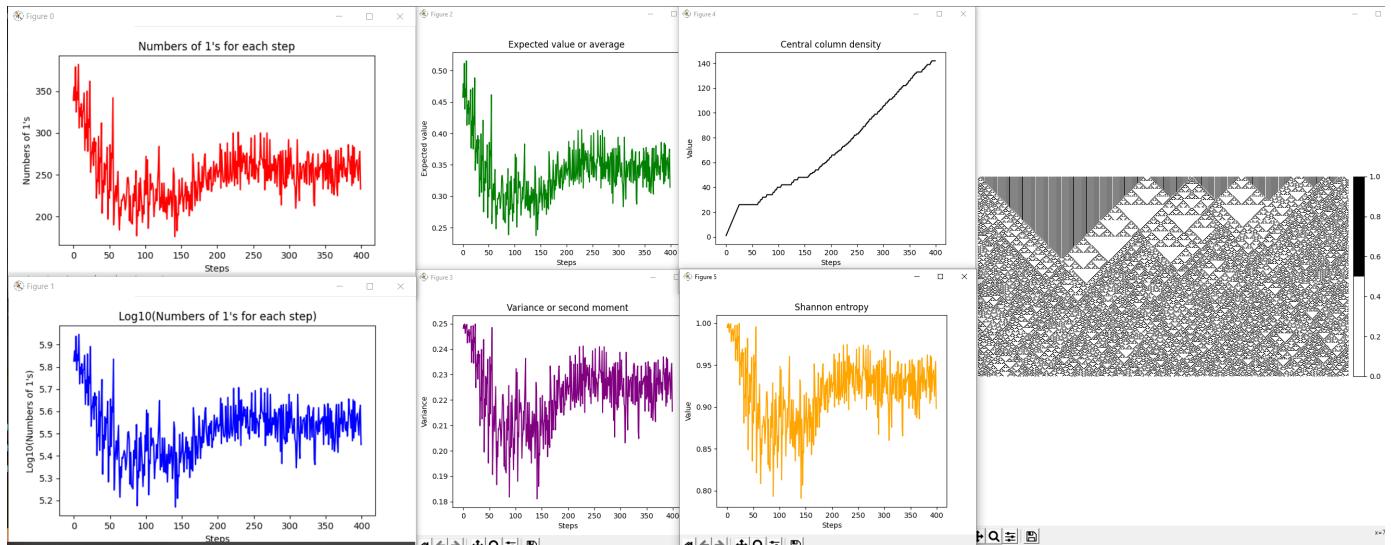


Figura 13: Autómata resultante con sus respectivas métricas.

Para nuestro autómata generado de manera aleatoria vemos como las gráficas comparten una silueta muy similar entra, vemos como el numero de unos por paso tiene un máximo al inicio con un valor de 200 para después disminuir y rondar valores de 104 como el mas bajo a 182 como el valor mas alto, si vemos la entropía de Shannon podemos ver que volvemos a tener un patrón similar a las pruebas pasadas, en donde comenzamos con prácticamente la entropía máxima a disminuir un poco y oscilar entre los valores 0.825 y 0.9941, esto nos vuelve a indicar que tenemos incertidumbre, examinando nuestra gráfica de la densidad en la columna central vemos un crecimiento prácticamente lineal en el numero de 1's pero hay que mencionar que volvemos a ver partes en donde la gráfica de densidad es constante, finalizando el análisis de este autómata podemos decir que no tenemos un patrón reconocible a diferencia de pruebas

pasadas en donde si podríamos llegar a visualizar patrones.

Por otro lado tenemos al autómata generado por expresión regular y a simple vista podemos observar como volvemos a compartir el patrón de triangulo invertido formado por líneas rectas de valores 0 y 1 intercaladas de los autómatas generados por expresión regular de las pruebas anteriores y de hecho podemos ver un gran parecido con el autómata de la figura 9 tanto en las gráficas como en el mismo autómata solo que el autómata que estamos estudiando tenemos el patrón de los triángulos incrustados en otro aún mayor pero en esta ocasión podemos ver esa patrón mas veces pero volvemos a llegar un punto en donde los patrones desaparecen. Observado las métricas que tenemos disponibles observamos como el numero de unos tiene un máximo de 382 unos a tener un mínimo de 177 para después empezar a rondar los valores de 211 a 300 células con valor 1, sin embargo, vemos como la densidad de la columna central al inicio tienen un crecimiento lineal entre la iteración inicial hasta la 25 para después volverse constante hasta la iteración 60 en donde ya empieza a existir un incremento casi lineal pero volvemos a ver como hay partes en donde es constante, pero en la entropía podemos ver que es muy similar a la vista en la figura 9 en donde vemos como inicia prácticamente en el máximo valor que puede tomar y después de 16 iteraciones empieza a disminuir y oscilar entre los valores 0.79 y 0.99, esto quiere decir que en nuestro sistema dentro de todas las iteraciones nos vemos incapaces de predecir el siguiente estado nuestro autómata y tenemos una alta incertidumbre.

Una vez visto estas tres pruebas podemos decir que la expresión regular nos genera patrones comunes entre los diferentes autómatas aun a pesar de que esta se genera de manera aleatoria, el patrón más reconocible son las intercalaciones de columnas de ceros y unos que forman un triangulo invertido, mientras que también existe otro patrón en donde hay triángulos inscritos en un uno mayor y es común ver estos patrones juntos, también vemos como si usamos el 50 %, 95 % e inclusive la formación aleatoria nos generan autómatas en donde la mayor parte de la iteraciones tenemos una gran entropía aunque esto también lo podemos ver en la generada mediante la expresión regular aunque existen formaciones como la figura 5 en donde la entropía llega casi al mínimo valor.

## 2.2. Regla 54

La expresión regular que nos describe a la regla 54 es el siguiente:

$$(0 + 1(11)^*10)(0 + 0(11)^*10)^*$$

La expresión regular anterior nos permite poder generar cadenas aleatorias que nos servirán como condiciones iniciales para la generación de nuestros autómatas celulares elementales. Para poder visualizar como el uso de nuestra expresión regular afecta al comportamiento de nuestro autómata haremos las siguientes pruebas:

- Probabilidad del 50 % de unos.
- Probabilidad del 95 % de unos.
- Generación de condición inicial completamente aleatoria.

Estas serán comparadas con cadenas aleatorias generadas mediante la expresión regular, todo esto se hará en un espacio de 400x400 células y se compararan tanto la imagen que nos genera el autómata así como con diferentes métricas que el programa nos proporciona.

### 2.2.1. Probabilidad del 50 % de unos

En la figura 14 podemos ver los valores de entrada en nuestro programa, en donde definimos un espacio de 400 x 400 células con una probabilidad de unos del 50 %

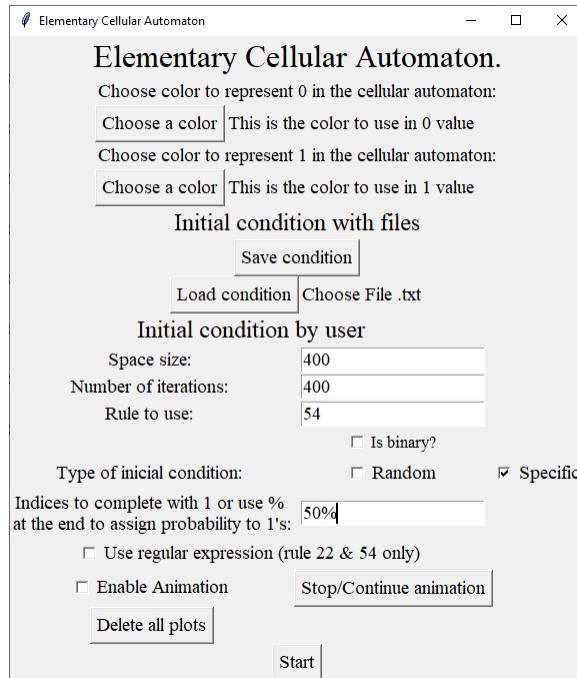


Figura 14: Entrada de nuestro programa para la primera prueba.

En la figura 15 vemos el autómata resultante así como las métricas que el programa nos genera.

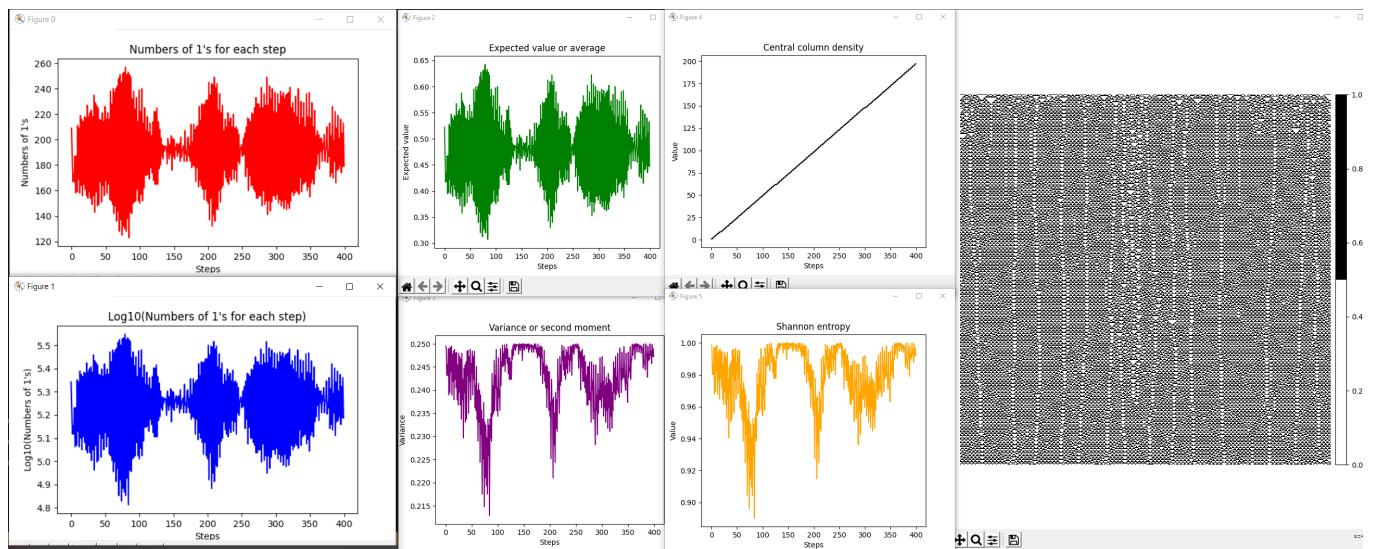


Figura 15: Autómata resultante con sus respectivas métricas.

En la figura 16 de igual manera que con la figura 14 podemos ver los valores de entrada de nuestro programa, en donde lo único que cambia es que en esta ocasión hacemos uso de la expresión regular de nuestra regla media la selección de la casilla con la etiqueta “Use regular expression (rule 22 & 54 only)”

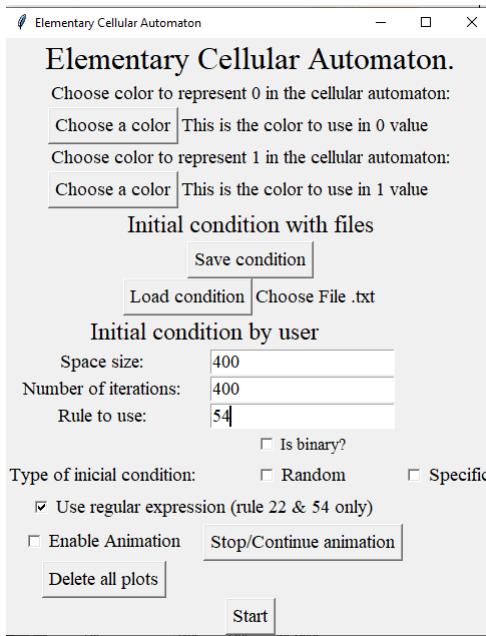


Figura 16: Entrada de nuestro usando nuestra expresión regular r54.

En la figura 17 vemos el autómata resultante así como las métricas que el programa nos genera.

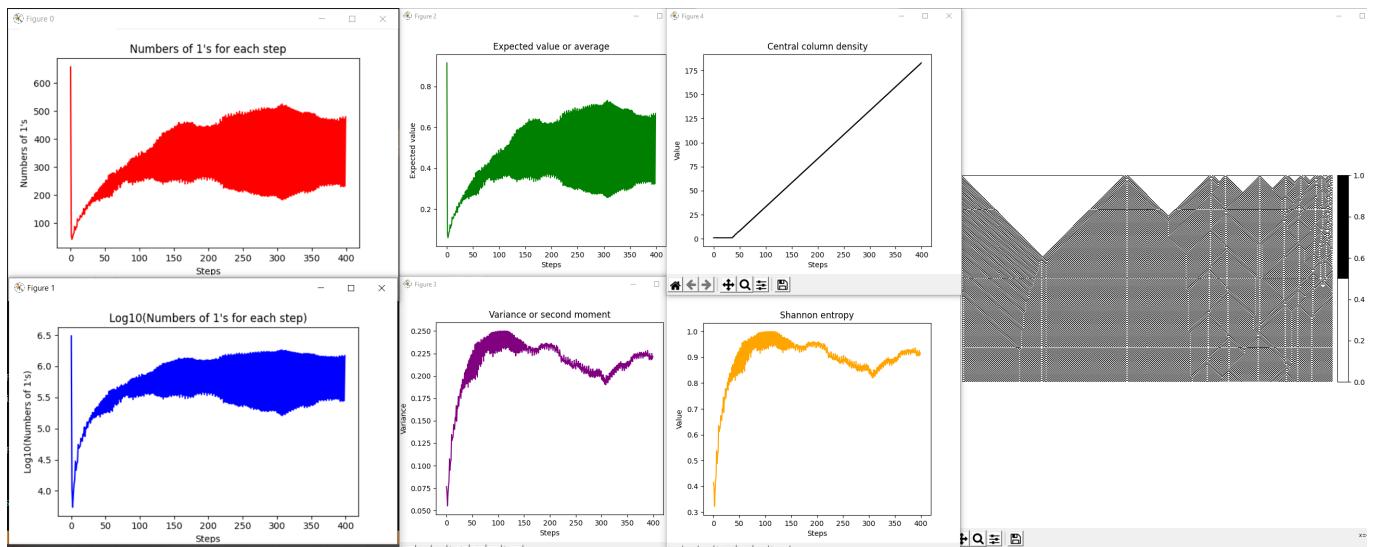


Figura 17: Autómata resultante con sus respectivas métricas.

Para nuestro autómata con probabilidad del 50 % de unos en la condición inicial vemos como 3 de nuestras gráficas tienen una silueta muy similar entre ellas, vemos como el numero de unos por paso se mantienen entre los valores de 123 y 258, si examinamos un poco la gráfica vemos como hay pasos intermedios en donde se encuentran prácticamente a la mitad de los valores anteriormente mencionados y si nos ponemos algo imaginativos podría llegar a parecer la gratificación de una parte de algún audio, si vemos la entropía de Shannon vemos que el valor prácticamente se mantiene en el valor mas alto que es 1 tenemos pasos intermedios en donde la entropía disminuye un poco, las partes en donde la entropía es prácticamente el valor máximo coincide con los puntos en donde el numero de 1 por paso se encuentra cercano a 200 1's por paso, esto es lógico ya que la probabilidad de los eventos sera igualmente probable

provocando así la entropía mas alta, como ya sabemos al tener esta entropía tan alta tenemos una gran incertidumbre y nos vemos incapaces de poder predecir el próximo estado que tomara nuestro autómata, examinando nuestra gráfica de la densidad en la columna central vemos como nos recuerda a una linea recta, es decir su crecimiento es lineal, pero si examinamos a detalle nuestra gráfica parece una escalera en donde los escalones tienen diferente longitud, esto se debe a que hay ciertas iteaciones en donde el numero de 1's se mantiene constante, y no tenemos un patrón reconocible en nuestro autómata pero si tenemos lineas formadas por triángulos invertidos pequeños a través de todo nuestro autómata.

Por otro lado tenemos al autómata generado por expresión regular y a simple vista podemos observar que es un caso diferente al primero ya que podemos observar cierto patrón en nuestro autómata, el cual tiene triángulos invertidos compuesto de células con valor 0 al inicio, para después desaparecer, y también los triángulos que se complementan con los anteriores justo a la mitad, empiezan a generar lineas prácticamente rectas que se dirigen hacia la parte baja. Observando las métricas que tenemos disponibles podemos ver como el numero de unos tiene un comportamiento muy interesante y es que vemos como al inicio de nuestro autómata tiene un máximo de 1's teniendo un valor de 657 que disminuye a un mínimo de 42 en la iteración 2 y a partir de ahí va incrementando de una manera casi logarítmica pero con el detalle de oscilar entre los valores 184 y 525, de hecho la densidad de la columna central al inicio es constante hasta la iteración 36 ya que desde ahí empieza a tener un incremento casi lineal ya que volvemos a ver ese escalonado pero en esta ocasión es prácticamente de una iteración, vemos a la entropía iniciar con un valor de 0.41 de entropía para luego caer a 0.32 y a partir de ahí comportarse como las demás gráficas y empezar a subir prácticamente hasta el valor máximo esto quiere decir que en nuestro sistema pasamos de poder tener la capacidad de predecir el siguiente estado a ser incapaces de predecir el siguiente estado de nuestro autómata, en otras palabras pasamos de tener baja incertidumbre a tener prácticamente la máxima incertidumbre posible.

En resumen vemos como la expresión regular en este caso nos genera algo completamente a la prueba del 50 %, algo curioso de estas pruebas es que en la entropía a pesar de que el autómata generado por medio de la expresión regular tiene baja entropía al inicio pocas iteraciones después tenemos prácticamente la máxima al igual que el caso de estudio del 50 % de probabilidad, también vemos que la gráfica de la columna de densidad central ambas tienen un crecimiento casi lineal y en los autómatas nos encontramos con pequeñas columnas que atraviesan al autómata en su totalidad.

### 2.2.2. Probabilidad del 95 % de unos

En la figura 18 podemos ver los valores de entrada en nuestro programa, en donde definimos un espacio de  $400 \times 400$  células con una probabilidad de unos del 95 %

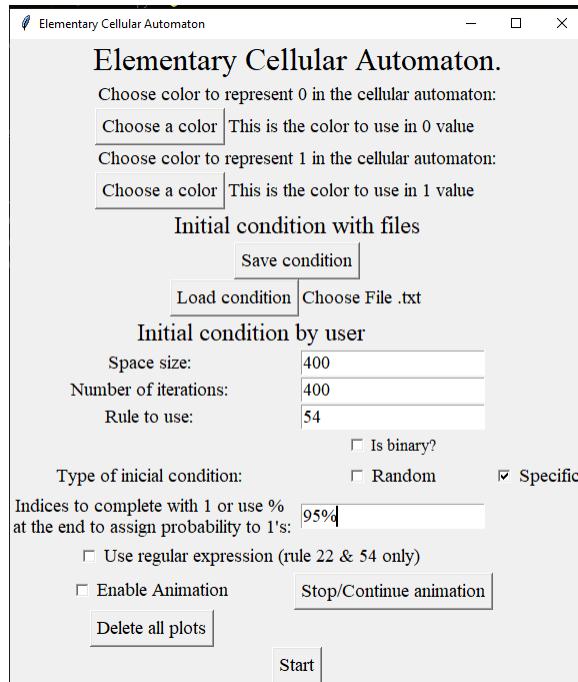


Figura 18: Entrada de nuestro programa para la segunda prueba.

En la figura 19 vemos el autómata resultante así como las métricas que el programa nos genera.

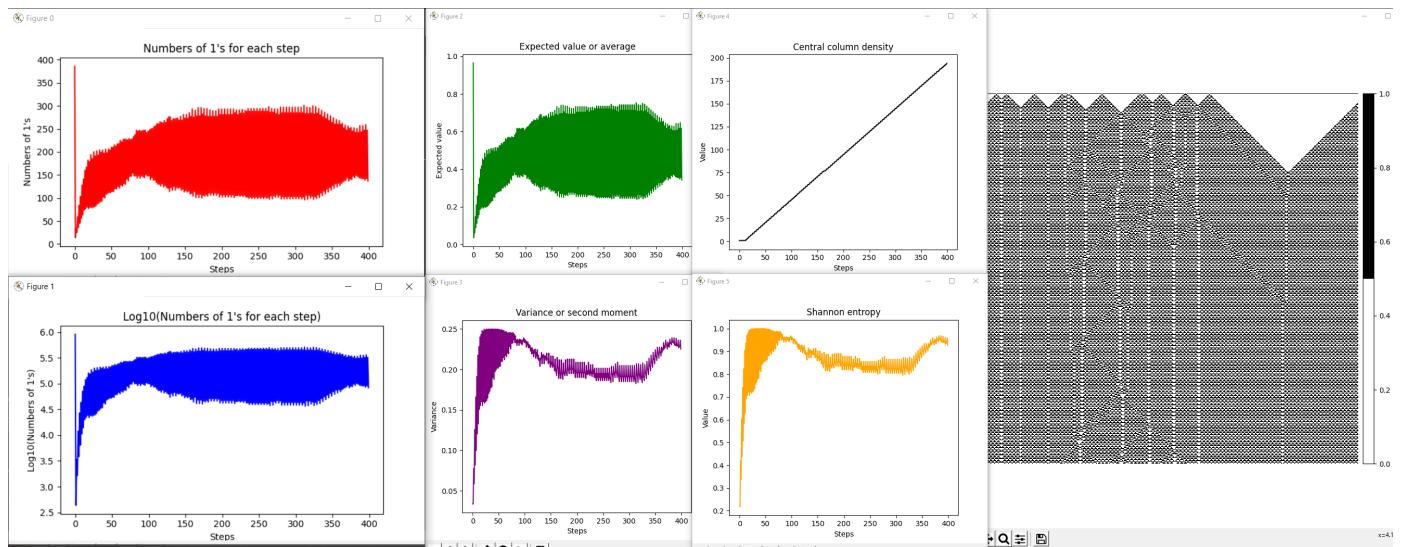


Figura 19: Autómata resultante con sus respectivas métricas.

En la figura 20 de igual manera que con la figura 18 podemos ver los valores de entrada de nuestro programa, en donde lo único que cambia es que en esta ocasión hacemos uso de la expresión regular de nuestra regla media la selección de la casilla con la etiqueta “Use regular expression (rule 22 & 54 only)”

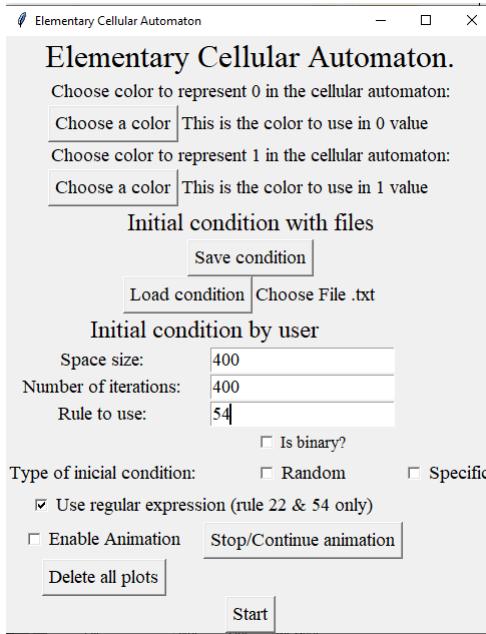


Figura 20: Entrada de nuestro usando nuestra expresión regular r54.

En la figura 21 vemos el autómata resultante así como las métricas que el programa nos genera.

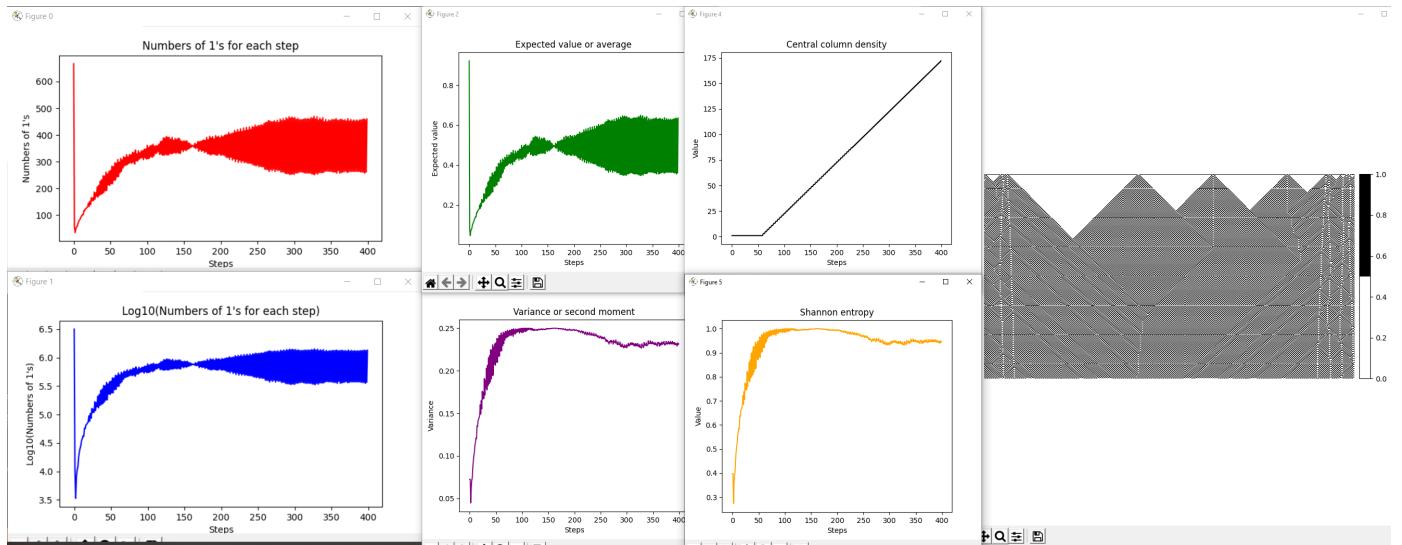


Figura 21: Autómata resultante con sus respectivas métricas.

Para nuestro autómata con probabilidad del 95 % de unos en la condición inicial vemos nos recuerda un poco al autómata de la figura 17 el cual fue generado con base en la expresión regular, ya que vemos métricas con siluetas prácticamente idénticas, de hecho a simple vista podríamos llegar a pensar en que nuestro autómata generado forma parte del autómata de la figura 17, entrando directamente en el estudio de las métricas podemos observar como el numero de unos por paso coincide con el comportamiento del autómata de la figura 17, ya que pasamos de un máximo, en este caso con valor de 386, para pasar a un mínimo de 14 unos en el paso 1, para empezar a subir en el numero de unos como si fuese una función logarítmica, solo que se mantiene oscilando entre los valores 98 como el límite inferior y de 300 como el límite superior, si vemos la entropía de Shannon podemos ver sin un duda alguna un comportamiento

parecido a la entropía de nuestro autómata de la figura 17, ya que de forma general partimos de un valor muy bajo, en este caso de estudio partimos del valor 0.2189 a empezar a subir de manera muy rápida a prácticamente el valor máximo posible de entropía, de hecho entre el paso 10 al 80 tiene una oscilación en los valores pasando de la entropía máxima a una inferior, el mínimo fue de 0.70, del paso 81 en adelante vemos como decrece y crece la entropía, esto entre los valores 0.80 y 0.9680, como algo interesante y al mismo tiempo curioso vemos que entre paso 81 al 400 nos recuerda un poco al electrocardiograma, por supuesto, esto en silueta, para terminar la parte de la entropía podemos decir que en nuestro sistema pasamos de poder tener la capacidad de poder saber el siguiente estado de nuestro autómata a perder dicha capacidad, en otras palabras pasamos de una incertidumbre muy baja a prácticamente la más alta posible, por último, examinando nuestra gráfica de la densidad en la columna central vemos nuevamente un comportamiento parecido con el autómata de la figura 17, en donde tenemos una parte constante de 1's para empezar a subir prácticamente de forma lineal, pero viendo de nuevo ese parte escalonada, en donde hay pasos en donde la densidad no es afectada, en este caso la parte constante es hasta el paso 11 y es ahí en donde vemos el crecimiento en la densidad de la columna central.

Por otro lado tenemos al autómata generado por expresión regular y a simple vista podemos observar una similitud con el autómata de la figura 17, tanto en comportamiento de las gráficas como en el autómata mismo, solo que en esta ocasión no tenemos una linea formada de triángulos que atravesen todo nuestro autómata en algunos de nuestros triángulos. Observando las métricas que tenemos disponibles podemos ver como el numero de unos tiene un comportamiento muy interesante y es que vemos como al inicio de nuestro autómata tiene un máximo de 1's teniendo un valor de 667 que disminuye a un mínimo de 34 en la iteración 2 y a partir de ahí va incrementando de una manera casi logarítmica con un oscilamiento muy interesante ya que pareciese que se complementan los pasos entre si, para que a partir del paso 167 oscile entre los valores 252 y 471, de igual forma los pasos parecen complementarse, de hecho la densidad de la columna central al inicio es constante hasta la iteración 58 ya que desde ahí empieza a tener un incremento casi lineal ya que volvemos a ver ese escalonado en donde tenemos 2 pasos que comparten la misma densidad hasta el paso 230 en donde ahora 1 paso es el que comparte la densidad anterior, vemos a la entropía iniciar con un valor de 0.39 de entropía para luego caer a 0.2743 y a partir de ahí comportarse como las demás gráficas y empezar a subir prácticamente hasta el valor máximo, sin embargo, vale la pena mencionar que parece que la entropía va decayendo poco a poco, pero esto quiere decir que en nuestro sistema pasamos de poder tener la capacidad de predecir el siguiente estado a ser incapaces de predecir el siguiente estado de nuestro autómata, en otras palabras pasamos de tener baja incertidumbre a tener prácticamente la máxima incertidumbre posible.

En resumen vemos como la expresión regular en este caso nos genera algo similar a la prueba del 95 %, esto nos quiere decir que nuestra expresión regular nos genera condiciones iniciales en donde podríamos encontrar un mínimo del 95 % de unos ya que los autómatas que se generaron fueron muy similares, principalmente en el comportamiento del autómata en donde el podemos ver ciertas similitudes entre ambos, también así en el comportamiento de nuestras métricas.

### 2.2.3. Generación de condición inicial completamente aleatoria

En la figura 22 podemos ver los valores de entrada en nuestro programa, en donde definimos un espacio de 400 x 400 células haciendo uso de la función random de la librería numpy de Python el cual nos generara un arreglo de determinado tamaño, en este caso de 400 células, lleno aleatoriamente de números 0 y 1.

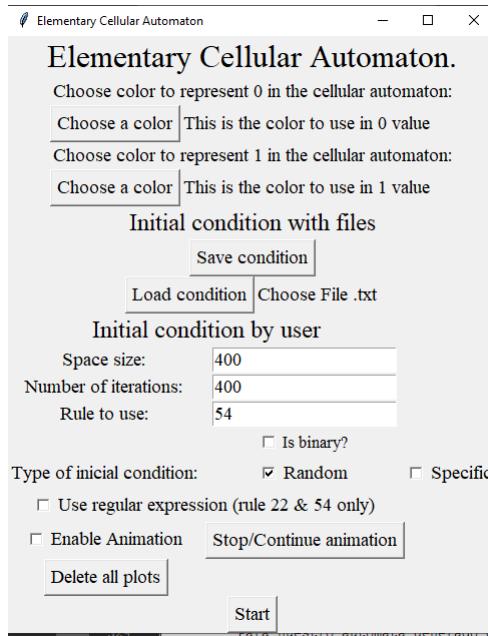


Figura 22: Entrada de nuestro programa para la tercera prueba.

En la figura 23 vemos el autómata resultante así como las métricas que el programa nos genera.

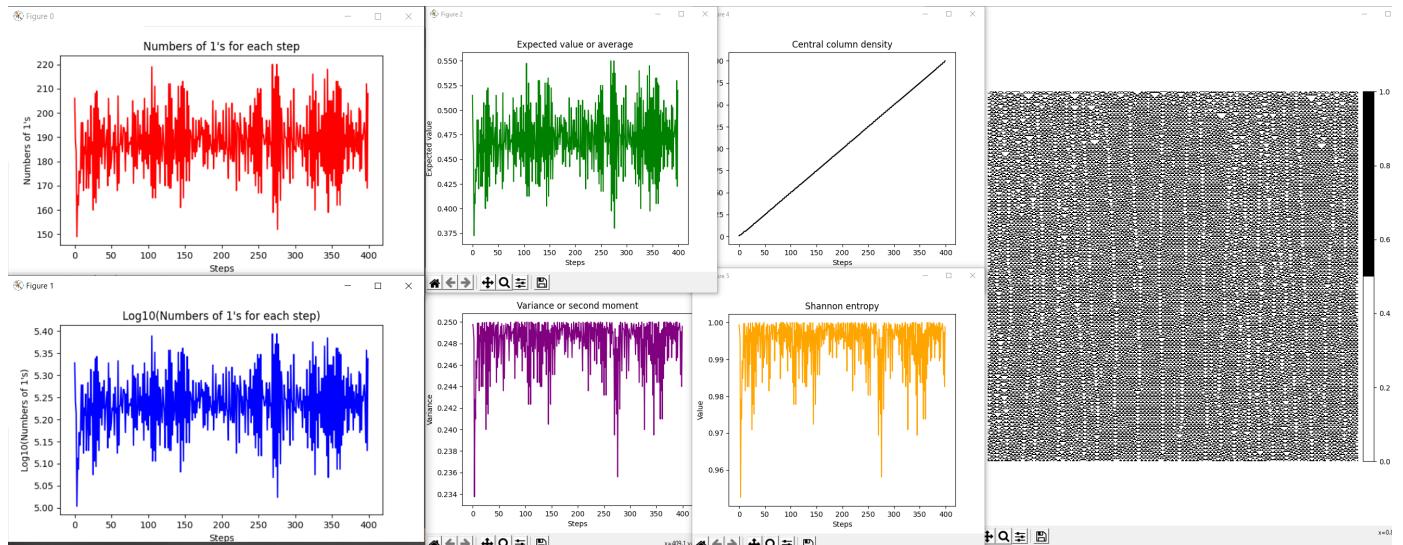


Figura 23: Autómata resultante con sus respectivas métricas.

En la figura 12 de igual manera que con la figura 10 podemos ver los valores de entrada de nuestro programa, en donde lo único que cambia es que en esta ocasión hacemos uso de la expresión regular de nuestra regla media la selección de la casilla con la etiqueta “Use regular expression (rule 22 & 54 only)”

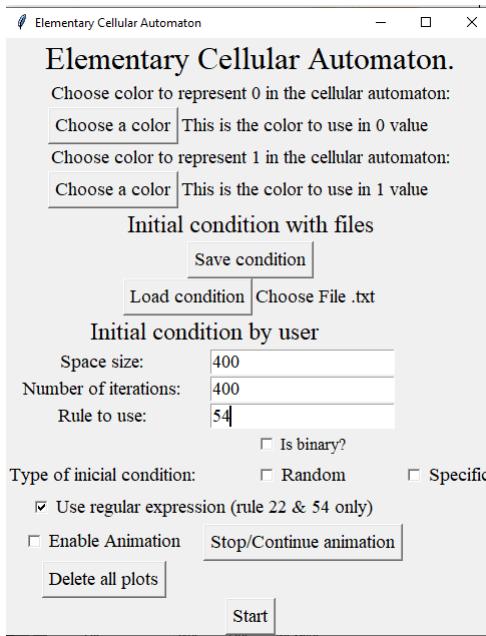


Figura 24: Entrada de nuestro usando nuestra expresión regular r22.

En la figura 13 vemos el autómata resultante así como las métricas que el programa nos genera.

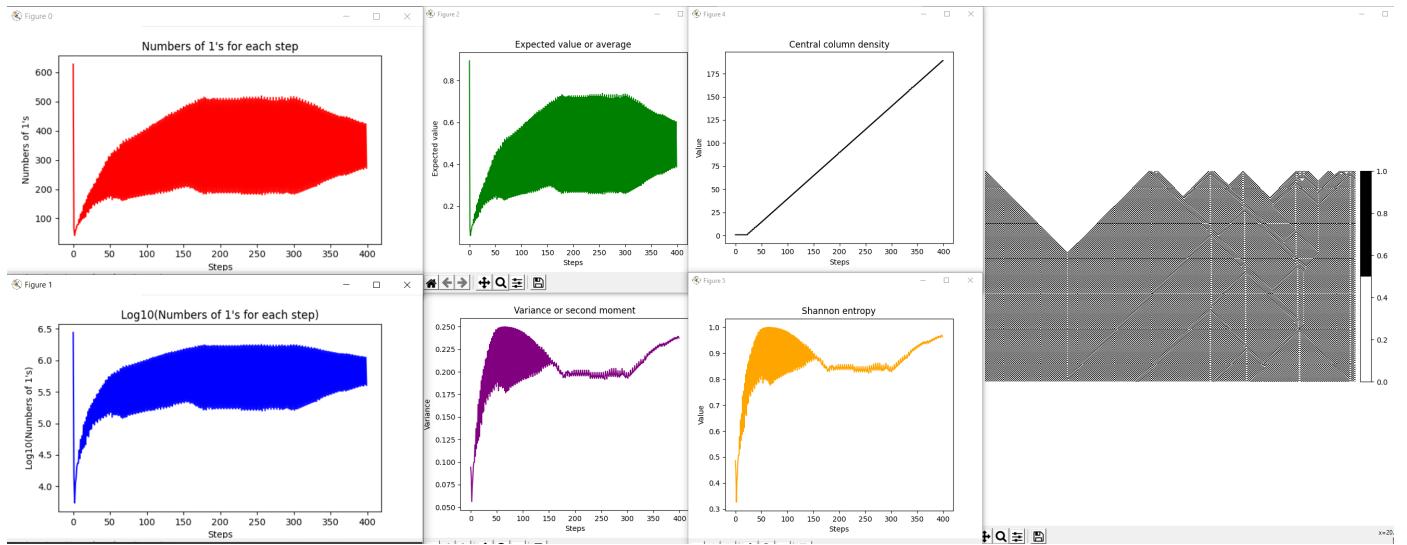


Figura 25: Autómata resultante con sus respectivas métricas.

Para nuestro autómata generado de manera aleatoria vemos como las gráficas son completamente diferentes a los autómatas generados con la misma regla anteriores, esto se debe a la generación aleatoria que tenemos y es que no tenemos control alguno, vemos como el numero de unos por paso tiene un mínimo global en el paso 3 con un valor de 149 1's por paso, y un máximo global en el paso 269 con un valor de 220 1's, si vemos la entropía de Shannon podemos ver que comenzamos con prácticamente la entropía máxima a diminuir a un mínimo global de 0.9525, que es relativamente poco, la entropía se mantiene prácticamente en el máximo por todos los pasos, esto nos vuelve a indicar que tenemos incertidumbre prácticamente máxima, examinando nuestra gráfica de la densidad en la columna central vemos un crecimiento prácticamente lineal en el numero de 1's pero hay que mencionar que volvemos a

ver partes en donde la gráfica de densidad es constante, finalizando el análisis de este autómata podemos decir que no tenemos un patrón reconocible.

Por otro lado tenemos al autómata generado por expresión regular y a simple vista podemos observar como volvemos a ver un comportamiento muy similar a los autómatas generados por la expresión regular. Observado las métricas que tenemos disponibles observamos como el numero de unos tiene un máximo de 652 unos a tener un mínimo de 35 para después incrementar y empezar a rondar los valores de 152 a 500 células con valor 1, también vemos como la densidad de la columna central al inicio tienen un crecimiento constante hasta el paso 23 para después crecer de una forma linea pero volvemos a ver como hay partes en donde es constante, estudiando la entropía podemos ver que es ligeramente diferentes a las anteriormente estudiadas, aunque el inicio es similar, empezamos de valores bajos para poco después subir prácticamente al máximo, oscilar entre el valor máximo y 0.775, para después no oscilar tanto y al final, en este autómata, para que la entropía crece entre la iteración 300 y la 400 llegando a un valor de 0.965, esto quiere decir que en nuestro sistema empezamos con una baja entropía para después oscilar entre valores altos de entropía, esto nos dice que somos incapaces de predecir el siguiente estado nuestro autómata y tenemos una alta incertidumbre.

Una vez visto estas tres pruebas podemos decir que la expresión regular nos genera autómatas comunes, sin embargo, en la prueba del 95 % vemos como se nos genera un autómata muy similar a los generados por la expresión regular, al igual es similar en las métricas que se nos genera, esto nos quiere decir que nuestros autómatas que genera la expresión regular tienen una gran cantidad de 1's, volvemos a ver como el uso de la opción de generar una condición completamente aleatoria nos genera resultados completamente diferentes tanto a la expresión regular como al porcentaje de unos.

### 3. Atractores

Podemos hacer una representación visual de los autómatas celulares, y para que podamos entenderlo de mejor manera es necesario mencionar los límites y las fronteras, del espacio en el cual existe el autómata celular

#### 3.1. Frontera abierta

Considera que todas las células fuera del espacio del autómata tienen un valor el cual es fijo.

#### 3.2. Frontera reflectora

Las células fuera del espacio del autómata toman los valores que están dentro como si se tratase de un espejo.

## 4. Programa

#### 4.1. Descripción

Para este primer programa se ha creado nuestro primer autómata celular, el autómata celular a crear debido a sus características se le conoce como un autómata celular elemental (ECA) que se regirá por la regla 30 (hay 256 reglas de la 0 hasta la 255), pero ¿Cuáles son esas características para que nuestro autómata celular sea considerado como un ECA? Las características son las siguientes:

- Espacio unidimensional.
- Tiene dos posibles estados o valores para cada célula (0 o 1).
- La reglas que dependen solo de los valores del vecino más cercano.
- Es de tipo frontera periódica o circular.

Nuestro ECA sera regido por la regla 30 o  $00011110_2$ , la cual nos estable nuestra función local, es decir, nos define como las células cambiarian de estado en la siguiente iteración dependiendo de los estados o valores de la generación anterior teniendo en cuenta que necesitaremos de una célula central y dos células vecinas, una a cada lado, por lo que el comportamiento seria el siguiente:

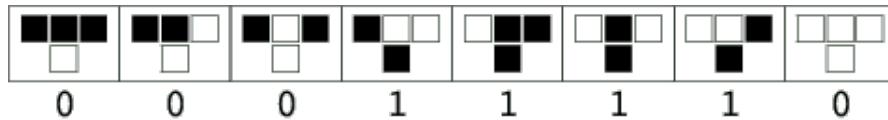


Figura 26: Comportamiento de la regla 30

En la imagen 3 podemos ver 8 figuras diferentes entre si, si vemos la primera fila compuesta de 3 células son las combinaciones posibles de los estados (0 o 1, blanco o negro) la célula de abajo nos da el estado que poseerá la hija de la siguiente iteración, es decir, si encontramos 3 células con estado 1 (color negro) en la siguiente iteración nos generaran una célula con el estado 0 (color blanco) o por ejemplo si encontramos que nuestras dos primeras células tienen estado 0 y la ultima tiene el estado 1 entonces la célula de la próxima iteración tendrá el valor 1.

Para el desarrollo de este programa se uso el lenguaje de programación Python usando Tkinter y Matplotlib para la creación de nuestra GUI y las gráficas correspondientes, el entorno de desarrollo utilizado fue Visual Studio Code. Mencionar que aunque nuestro programa nos permite usar cualquiera de las 256 reglas, en esta ocasión solo nos enfocaremos en la regla 30. Mencionar que nuestro archivo principal y el que se debe de ejecutar es el archivo con el nombre Interfaz.py ya que el archivo Logica.py solo contiene la parte lógica del programa y parte de la graficación de los resultados.

## 4.2. Pruebas

Este programa tiene varios aspectos que han sido cubiertos. Para iniciar las pruebas crearemos un autómata de 1000 células con 1000 iteraciones, algo especial de este autómata es que tenemos solo una célula con valor 1 en medio de todas las demás células que tienen valor 0.

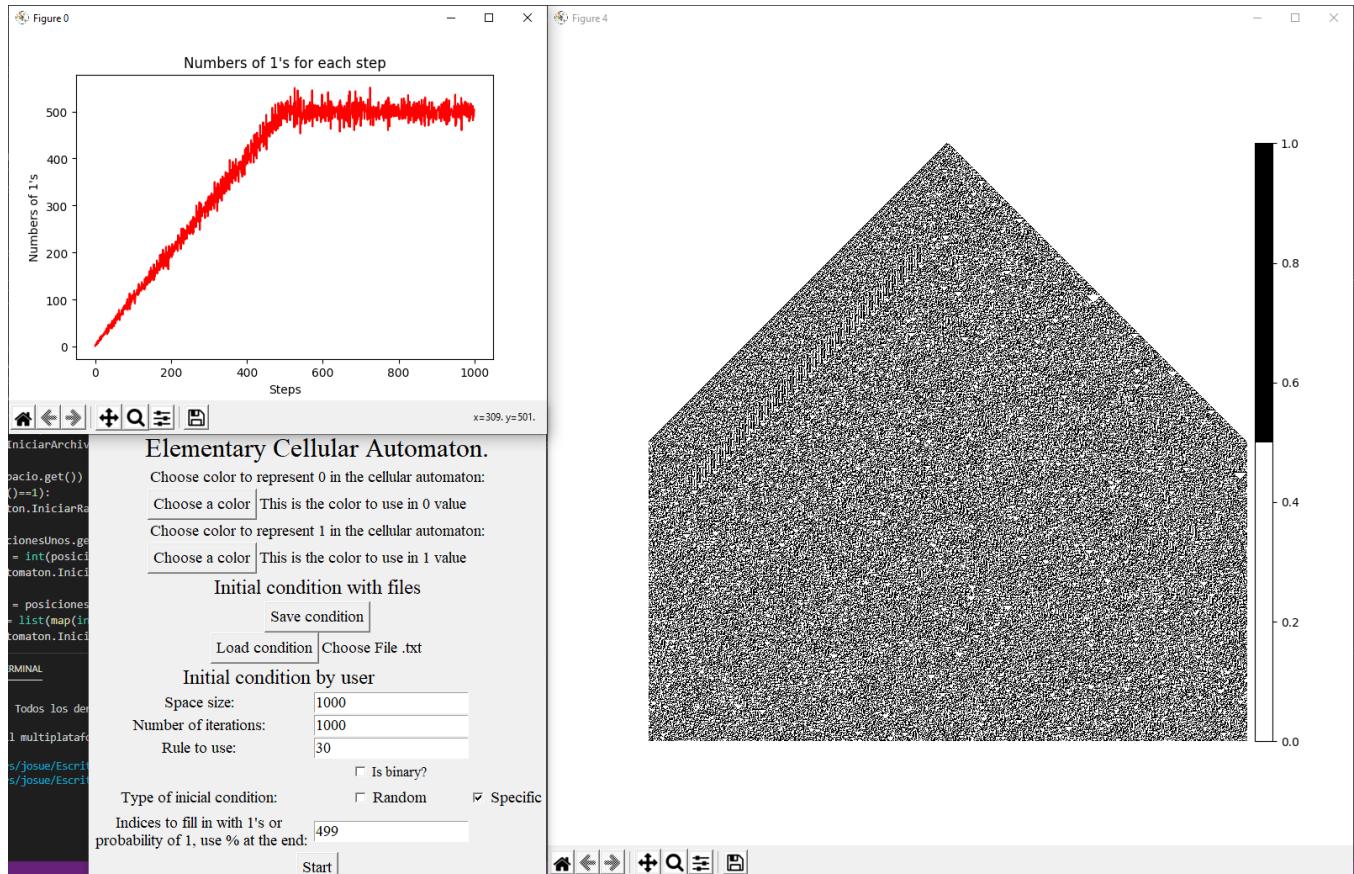


Figura 27: ECA con solo una célula con valor 1. Regla 30

Como vemos en la figura 4 tenemos en el lado derecho de la imagen nuestro ECA resultante después de aplicar 1000 iteraciones con la regla 30 y en la parte superior izquierda tenemos el crecimiento de células con valor 1 durante esas 1000 iteraciones, podemos apreciar que su crecimiento nos recuerda a un crecimiento logarítmico. En ambas gráficas tenemos la posibilidad de agrandar el tamaño mediante el ícono de la lupa que tenemos en la parte inferior de ambas ventanas, una vez seleccionada la lupa podremos seleccionar un área rectangular en nuestra gráfica para poder agrandar lo que abarque dicha área, esto con el fin de visualizar aun mejor nuestro autómata y el crecimiento de nuestras células con valor 1 como se ve en la figura 5.

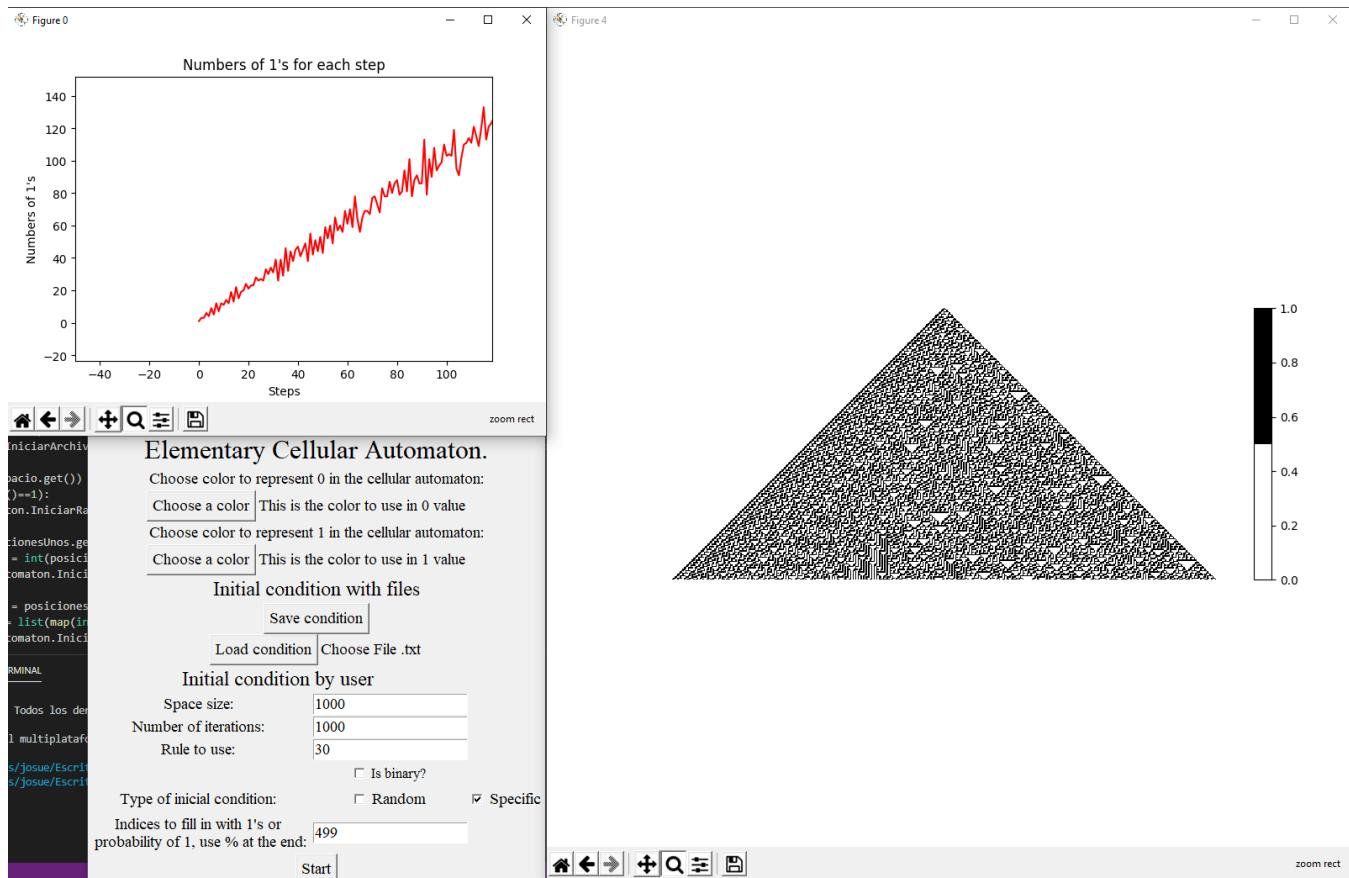


Figura 28: ECA con solo una célula con valor 1 agrandando el tamaño para poder visualizar aun mejor el comportamiento. Regla 30

Nuestro programa también nos permite guardar nuestra configuración de nuestro ECA, en este caso solo toma la iteración 0 y es guardado en un archivo con el siguiente nombre "celulasIniciales.txt" en caso de existir sobre escribirá la nueva información, en caso contrario lo creara desde cero, mencionar que si se desea se puede modificar el nombre pero no la extensión posterior de haber creado el archivo. En este caso nuestro programa nos arroja un mensaje diciendo que el archivo fue guardo exitosamente y se nos crea un archivo con el nombre y extensión anteriormente mencionados (ver figura 6 y 7).

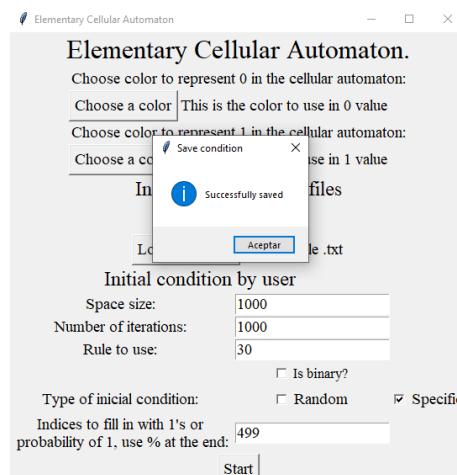


Figura 29: Iteración 0 de nuestro ECA guardada exitosamente

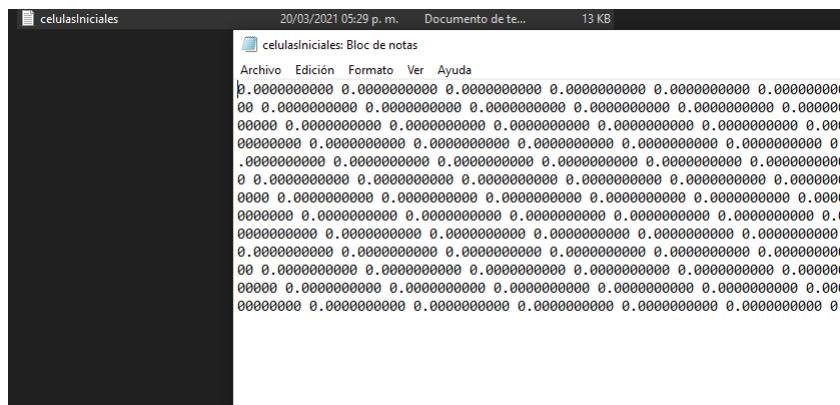


Figura 30: Contenido del archivo

También como era de esperarse se puede cargar archivo de texto al programa para volver a usarlo, algo importante a mencionar es que no guarda ni el numero de iteraciones, los colores para los valores 0 y 1 ni la regla usada esto con fines de facilitar al usuario poder usar un mismo inicio con diferentes reglas y numero de iteraciones cada vez que le plazca hacerlo. En la figura 8 cargamos nuestro archivo y solo modificamos el color que se le asigna a los valores 0 y 1, colocamos la regla a usar (regla 30) y nuestro numero de iteraciones sera de nuevo 1000, el panel para poder elegir el color se puede observar en la figura 9.

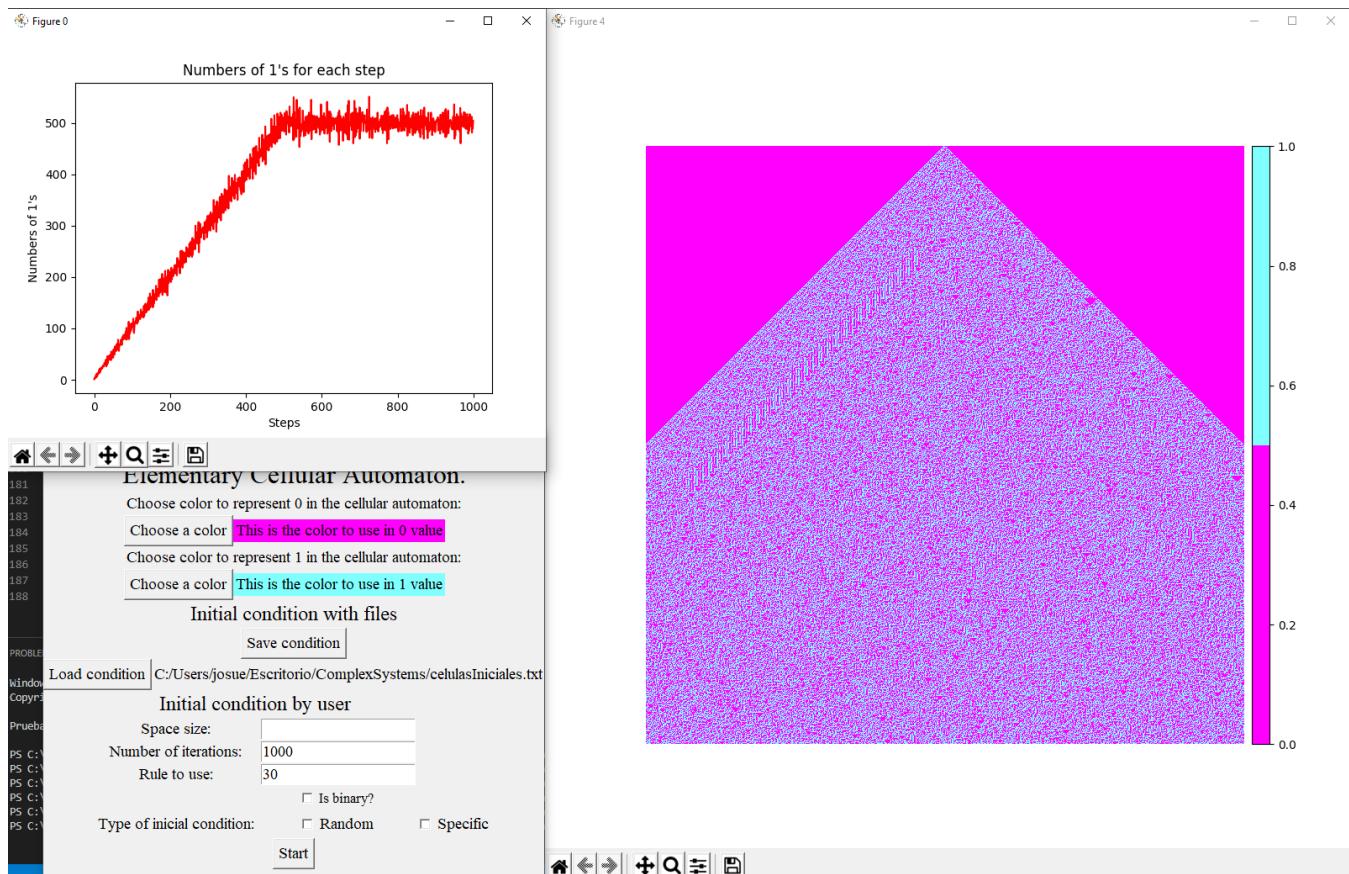


Figura 31: ECA con diferentes colores para los valores 0 y 1

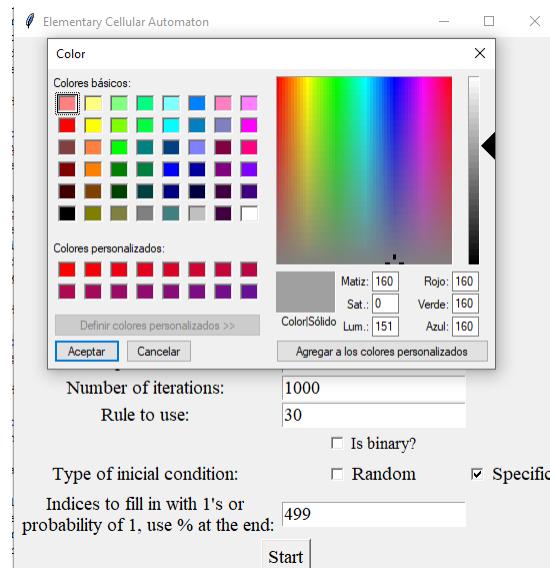


Figura 32: Panel para poder elegir color para valores 0 y 1

Realizando otra prueba crearemos un ECA con un tamaño de 500 células con 500 iteraciones, pero en esta ocasión la regla 30 sera ingresada como numero binario ( $00011110_2$ ) y serán asignadas células con valor 1 de forma aleatoria.

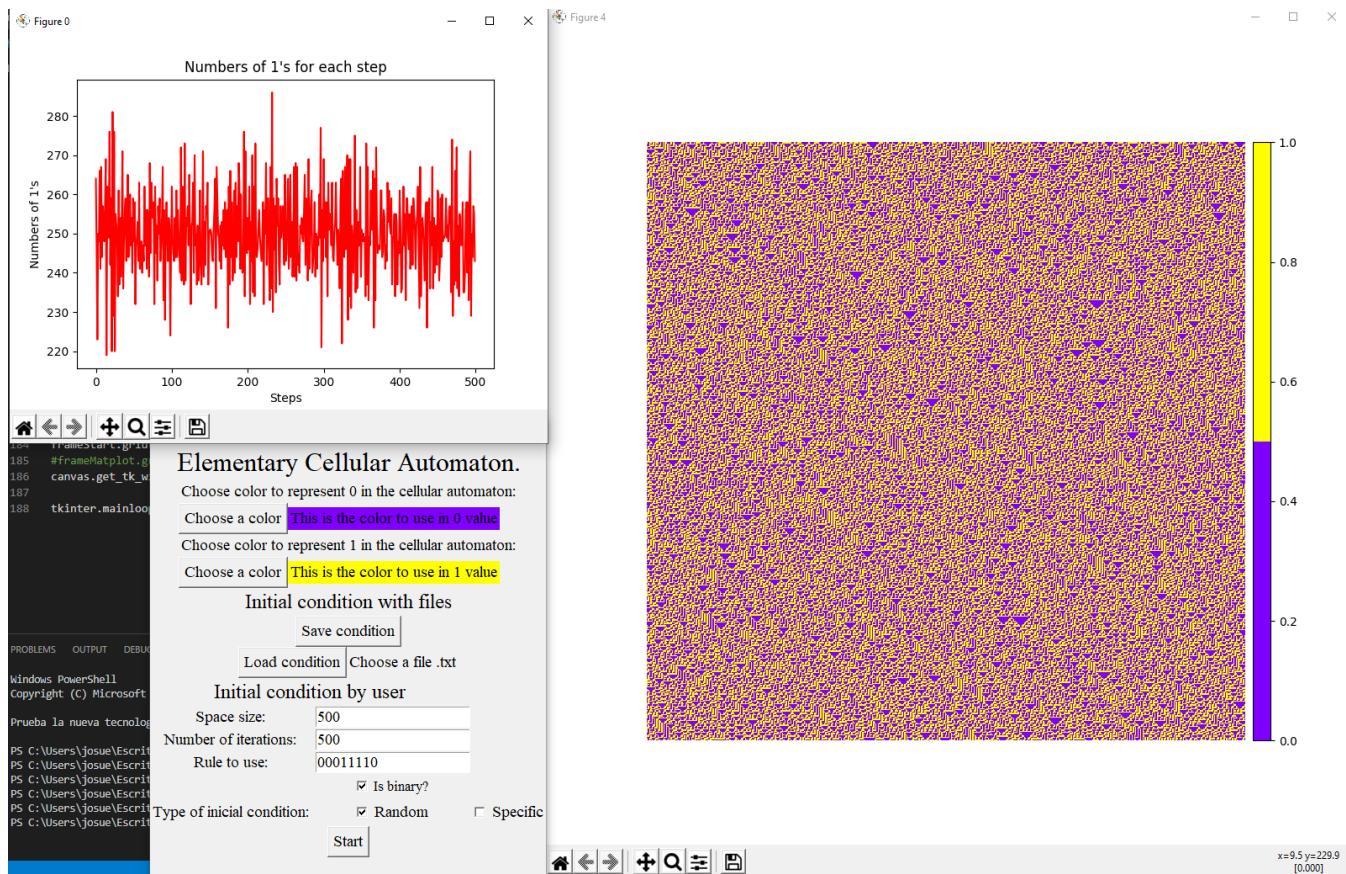


Figura 33: ECA de 500 células por 500 iteraciones creado de forma aleatoria

Como ultimas pruebas realizaremos lo siguiente: mediante las reglas 15, 22, 30, 54, 110 y 126 para la construcción de nuestro ECA mostraremos 3 evoluciones por cada regla en espacios de 400 células por 400 iteraciones, una empezara con un uno al centro, otra iniciando con un 50 % de probabilidad de unos y la última iniciando con un 95 % de probabilidad de unos.

Empecemos así con nuestra regla 15, veremos a través de las imágenes 11 a la 14 el comportamiento de nuestro ECA con base en donde se encuentren las células iniciales con valor 1.

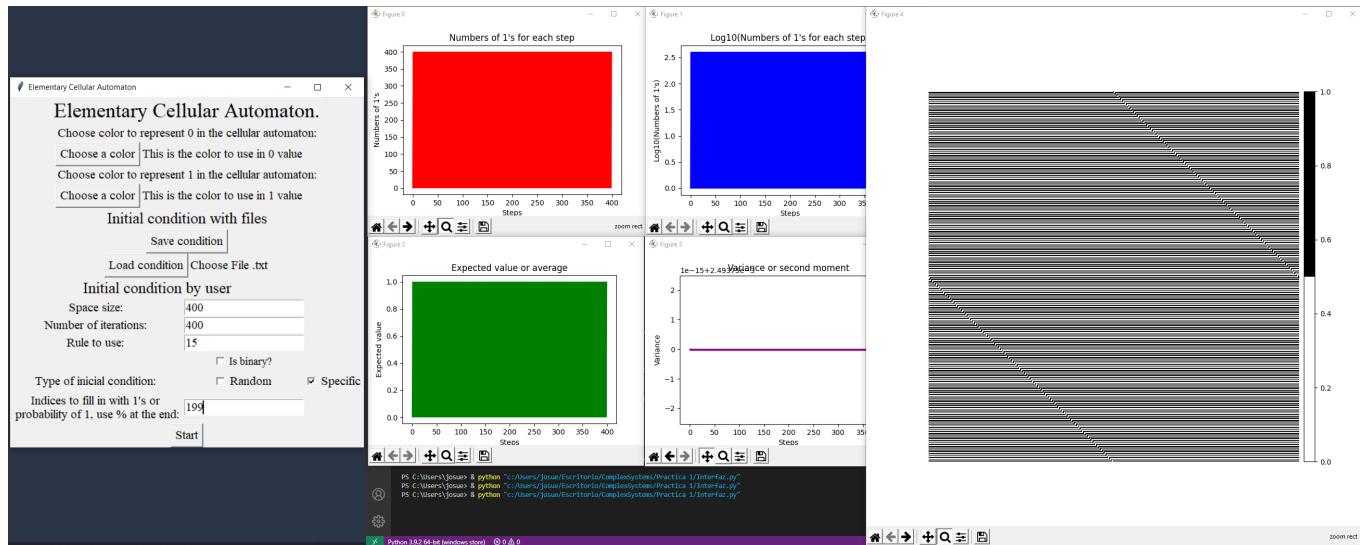


Figura 34: ECA de 400 células por 400 iteraciones con una célula inicial central, regla 15

Para visualizar mejor la información nuestras gráficas hagamos una ampliación de las mismas para las primeras 50 iteraciones.

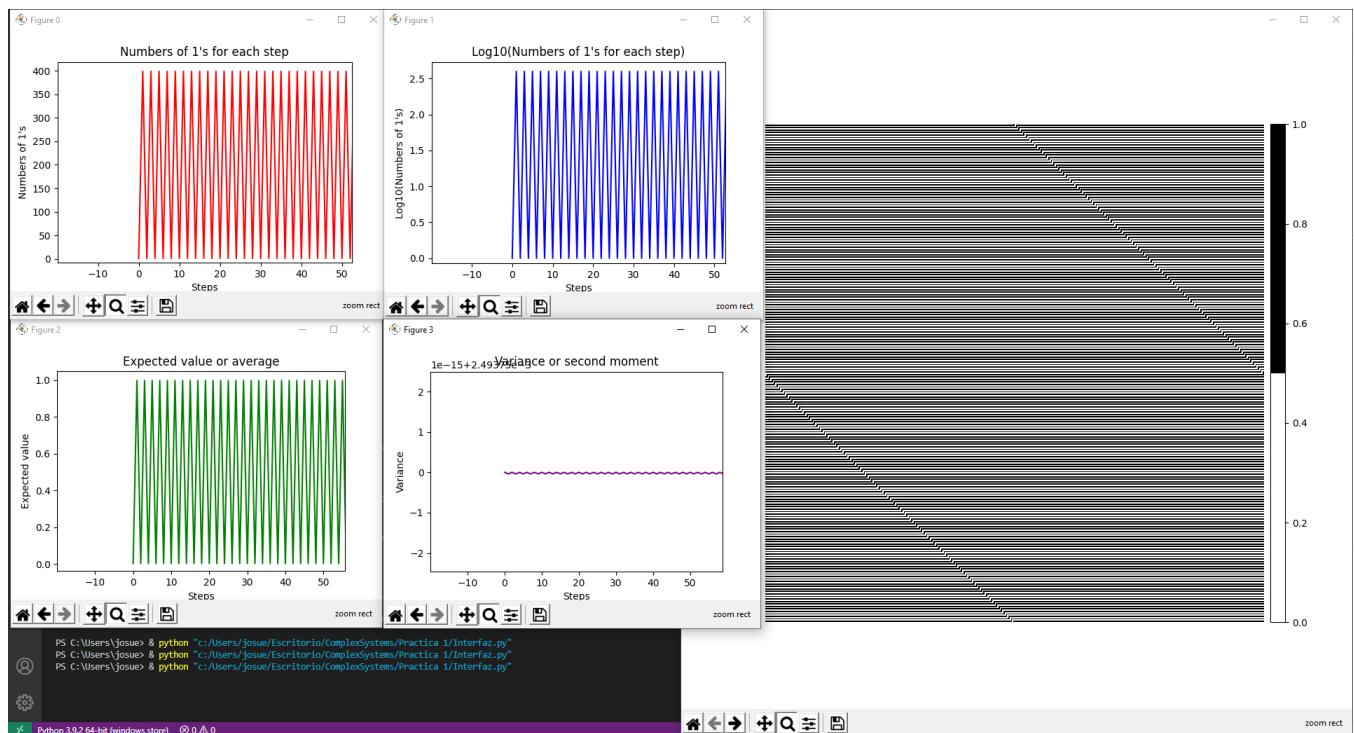


Figura 35: ECA de 400 células por 400 iteraciones con una célula inicial central con aumento, regla 15

Como vemos el comportamiento de nuestro ECA es de una forma constante y es que en la iteración tenemos 1 célula viva a pasar a tener 399 células y a la siguiente volvemos a tener solo una, como si de un ciclo se tratase.

Pasando a la siguiente prueba, usaremos el 50 % de unos y observemos su comportamiento

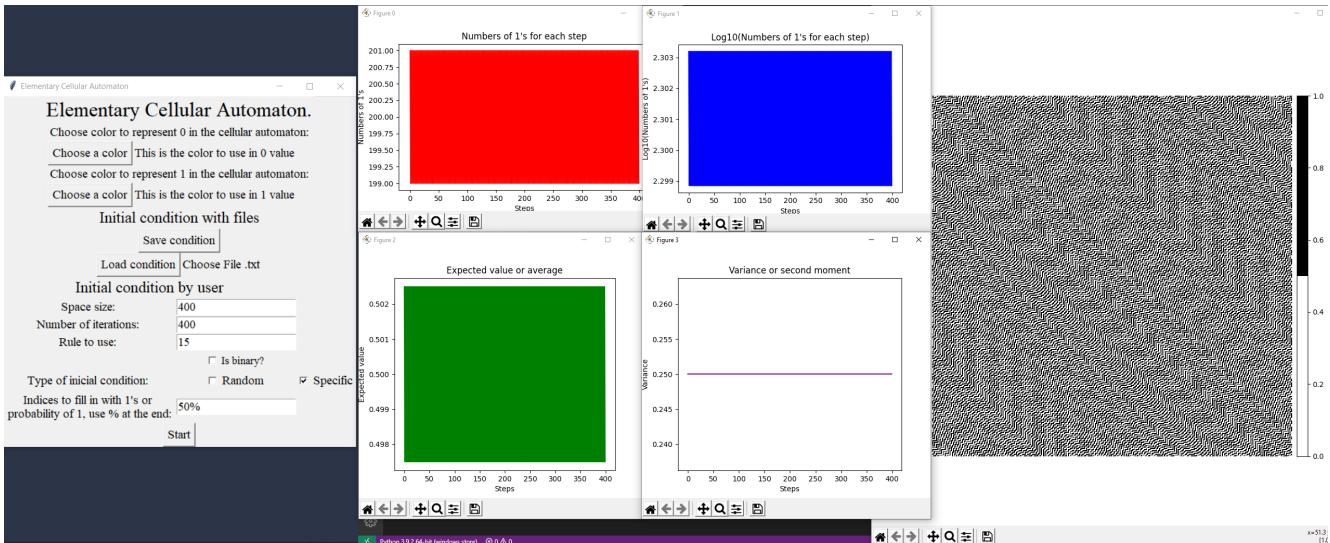


Figura 36: ECA de 400 células por 400 iteraciones con 50 % de probabilidad de 1's, regla 15

Como vemos tiene un comportamiento similar a la primera prueba solo que el numero de células vivas por iteración cambiaron, ahora son menos que en la prueba anterior. Como ultima prueba para esta regla, la probabilidad a usar sera del 95 % así que veamos el resultado

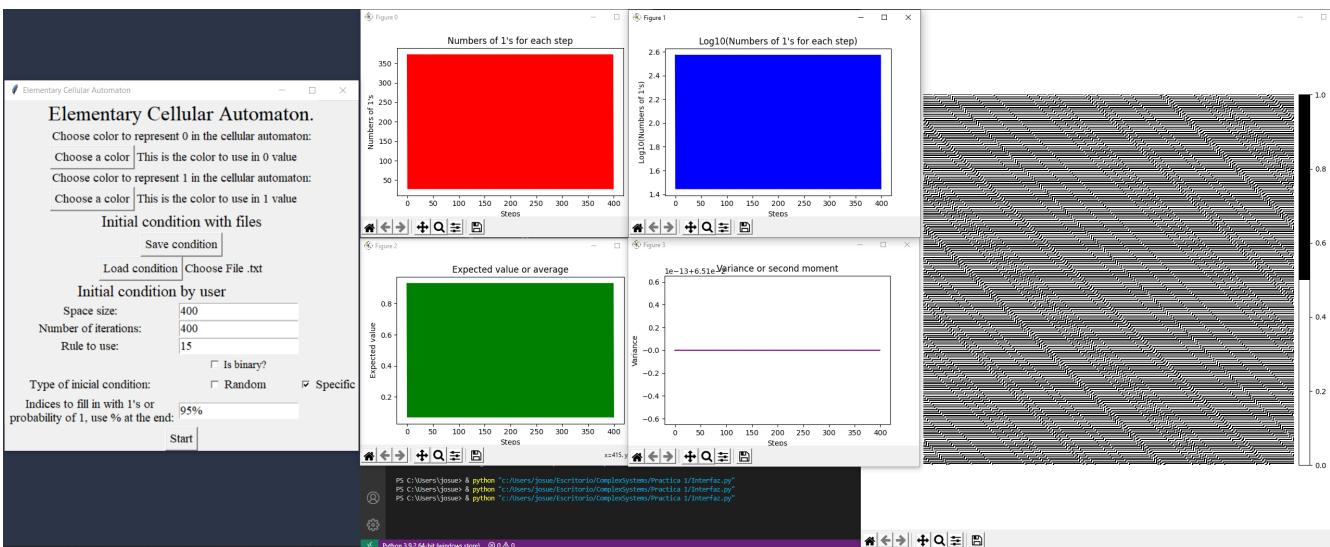


Figura 37: ECA de 400 células por 400 iteraciones con 95 % de probabilidad de 1's, regla 15

El resultado nuevamente es muy similar a los anteriores, por lo que podemos suponer que si probamos diferentes configuraciones iniciales siempre serán parecidas, solo cambiarian el numero de 1's.

Continuando con las pruebas, ahora vayamos con la regla 22 de aquí en adelante se mostraran las 3 imágenes correspondientes con los 3 valores iniciales diferentes para posteriormente dar un pequeño

estudio de los resultados.

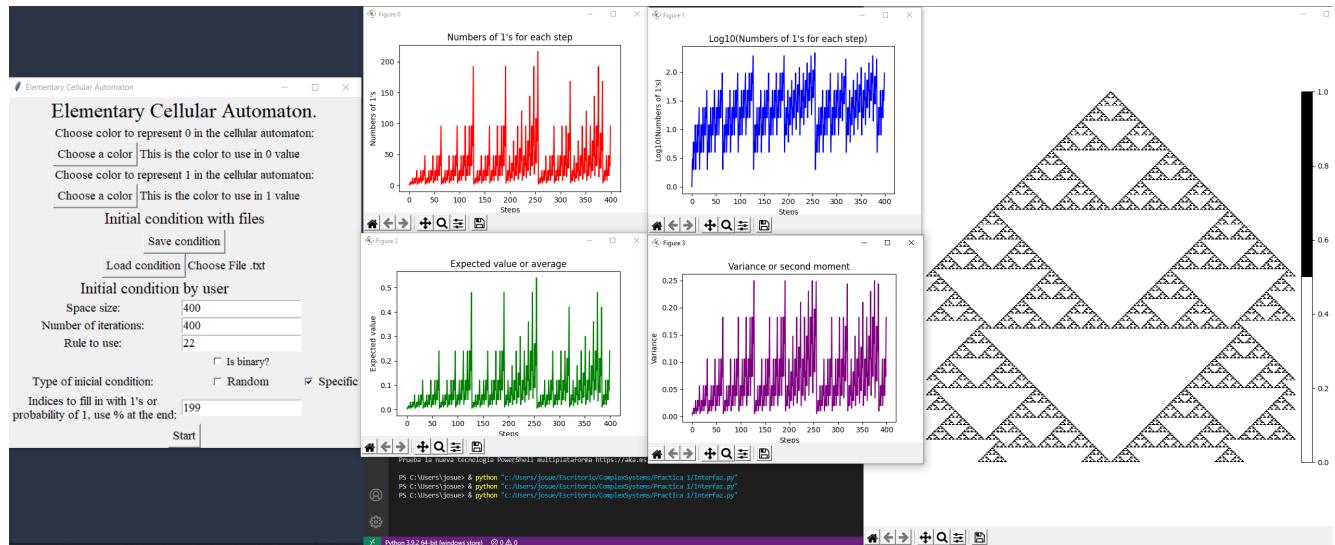


Figura 38: ECA de 400 células por 400 iteraciones con una célula inicial central, regla 22

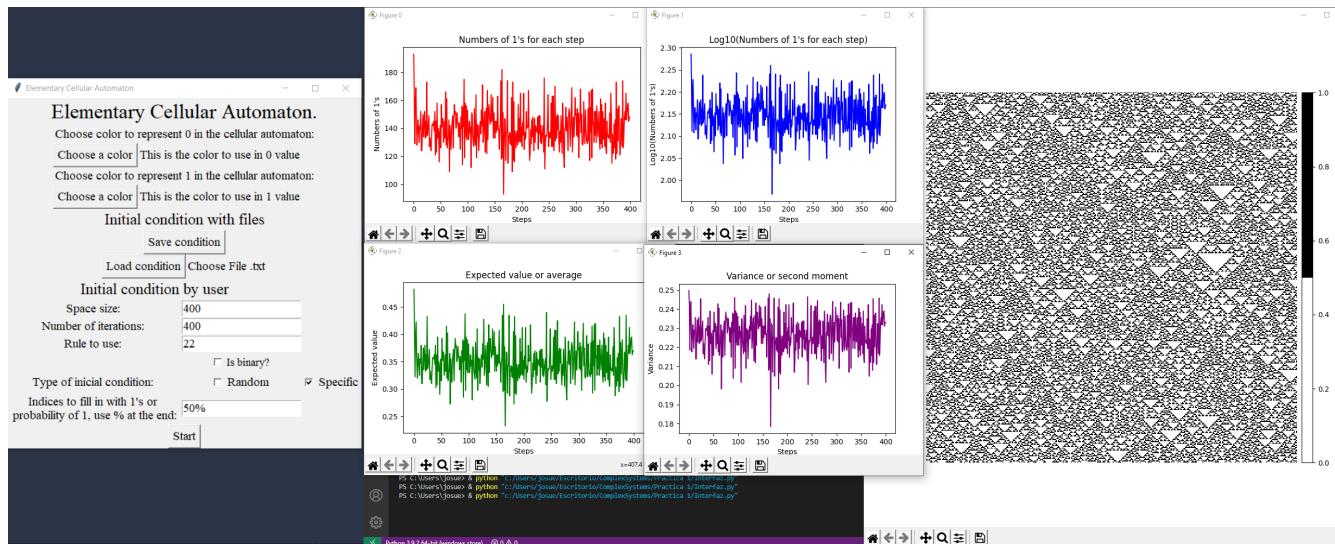


Figura 39: ECA de 400 células por 400 iteraciones con 50 % de probabilidad de 1's, regla 22

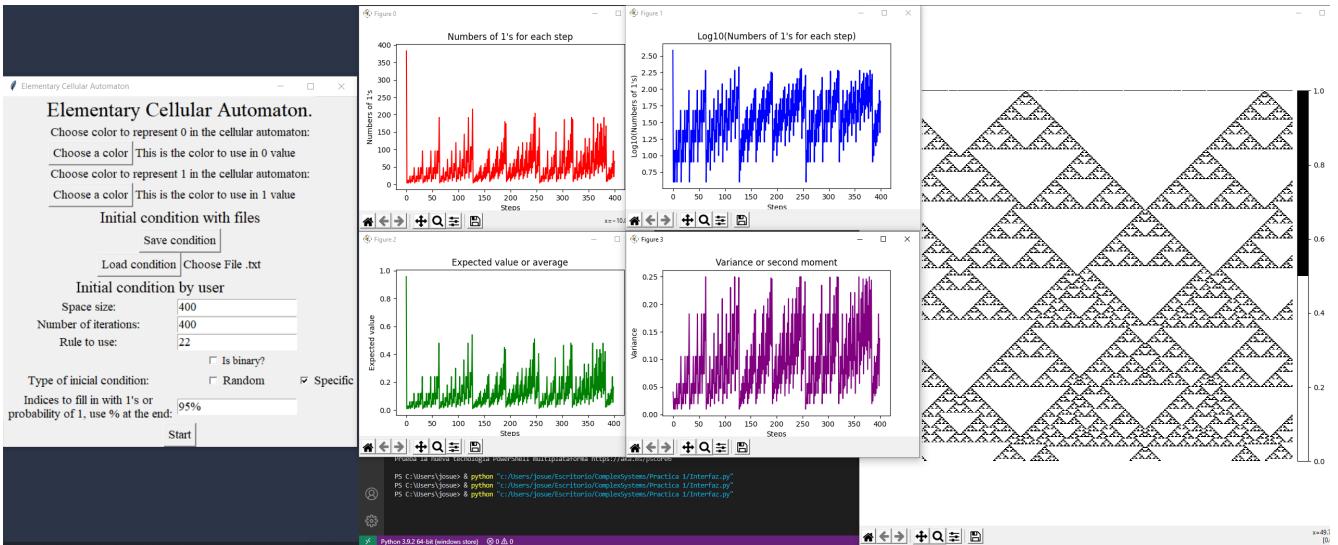


Figura 40: ECA de 400 células por 400 iteraciones con 95 % de probabilidad de 1's, regla 22

Como se ven desde la figura 15 a la 17 con esta regla si cambia el comportamiento de nuestras gráficas con base a cada tipo de condición inicial, aunque la figura 15 y 17 tiene ciertas similitudes pero no son lo mismo, empezamos a ver que tan influyente es la regla en el comportamiento de nuestro ECA.

Ahora pasemos a la regla 30 con las mismas pruebas.

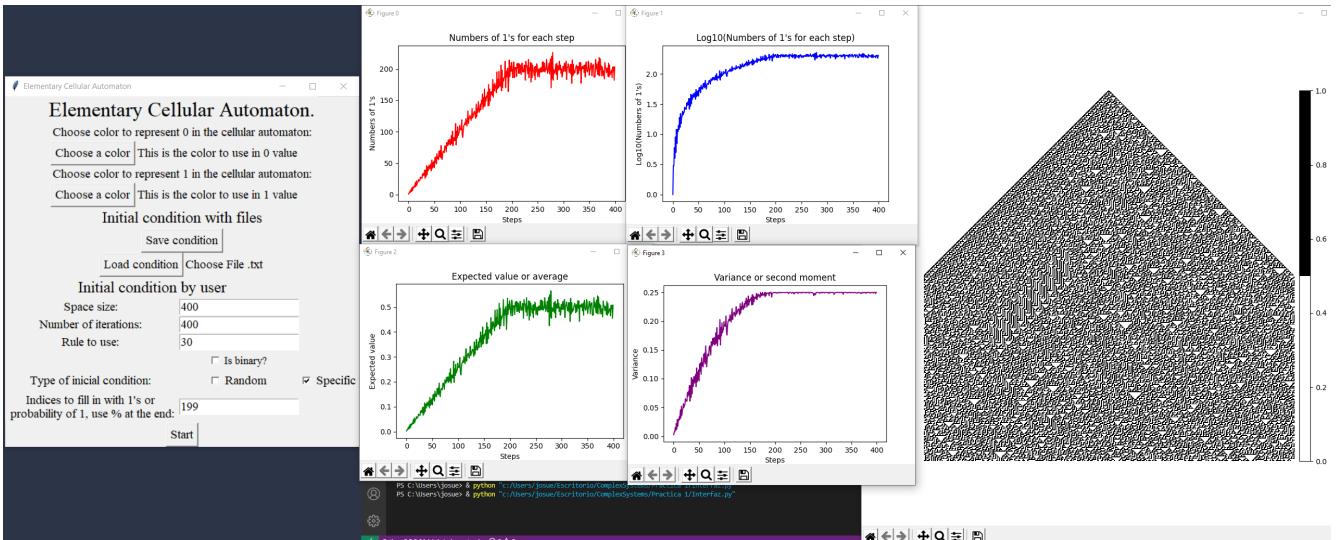


Figura 41: ECA de 400 células por 400 iteraciones con una célula inicial central, regla 30

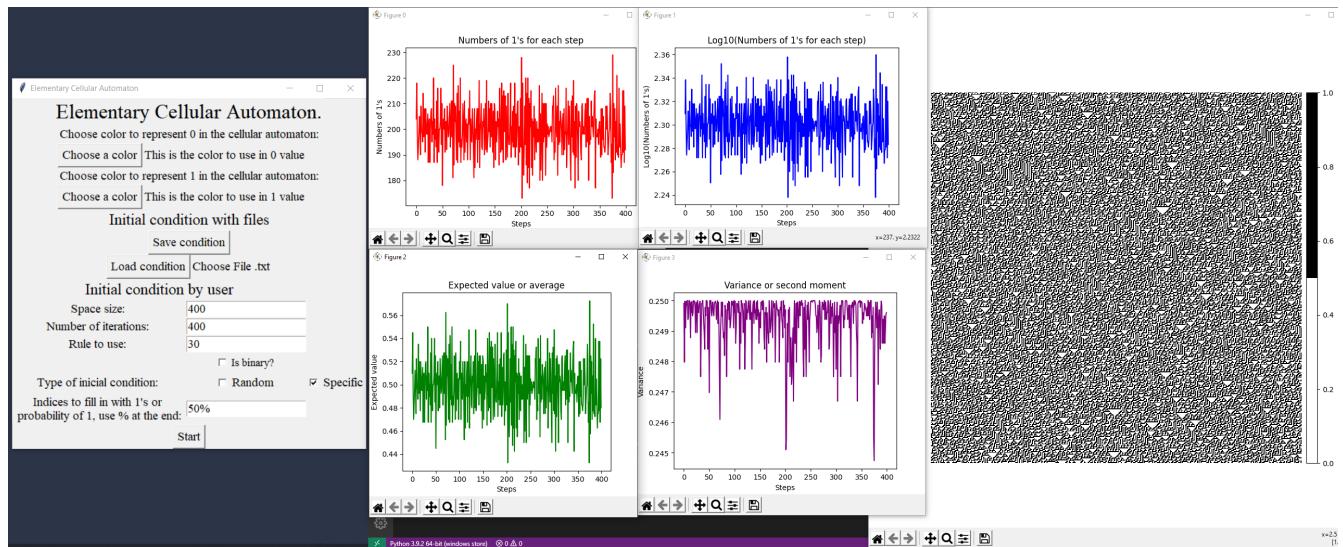


Figura 42: ECA de 400 células por 400 iteraciones con 50 % de probabilidad de 1's, regla 30

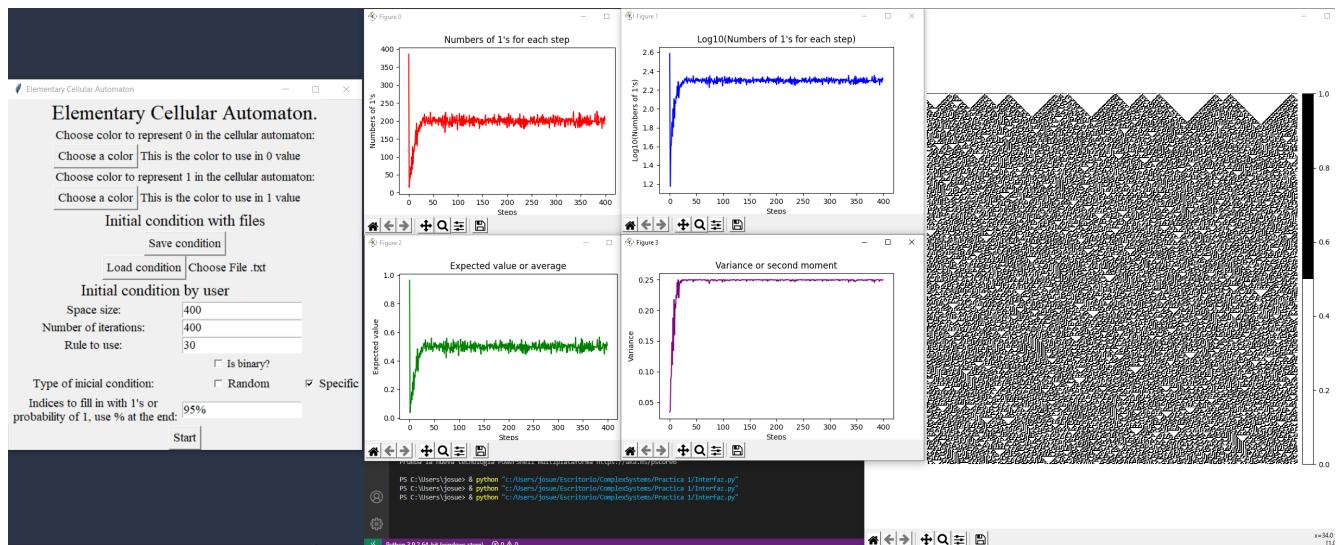


Figura 43: ECA de 400 células por 400 iteraciones con 95 % de probabilidad de 1's, regla 30

Como vemos en las figuras 18 a la 20 esta regla nos recuerda el comportamiento de la regla anterior, en el sentido de que las figuras 18 y 20 son muy similares en su comportamiento el cual es como un crecimiento logarítmico, en cambio la figura 19 tiene un comportamiento caótico.

Continuando, ahora veremos la regla 54.

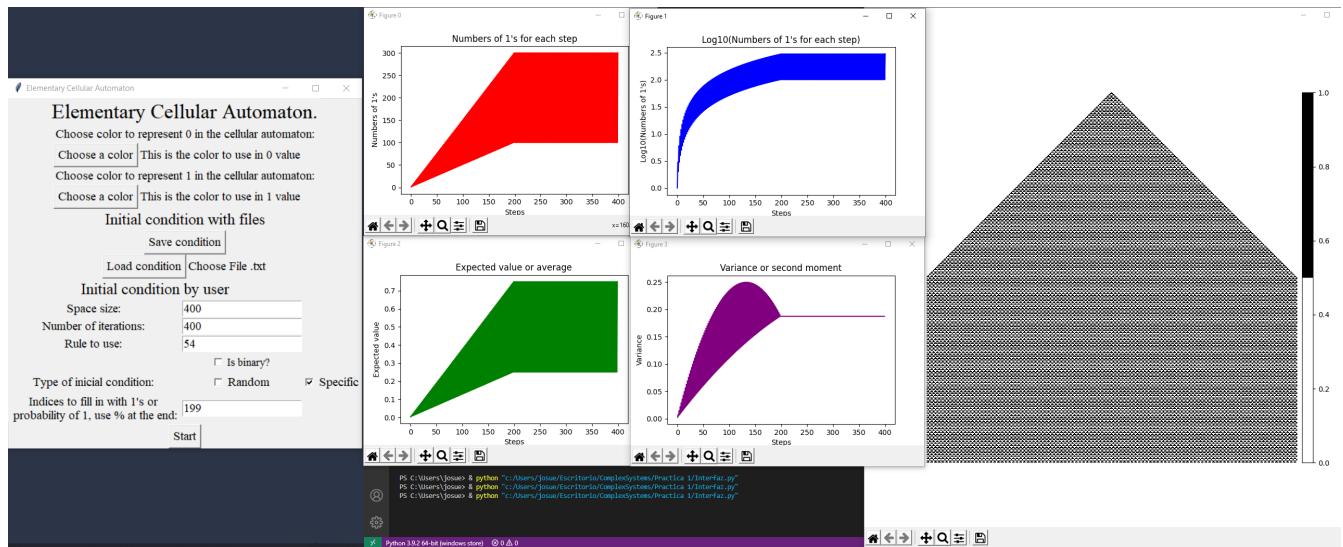


Figura 44: ECA de 400 células por 400 iteraciones con una célula inicial central, regla 54

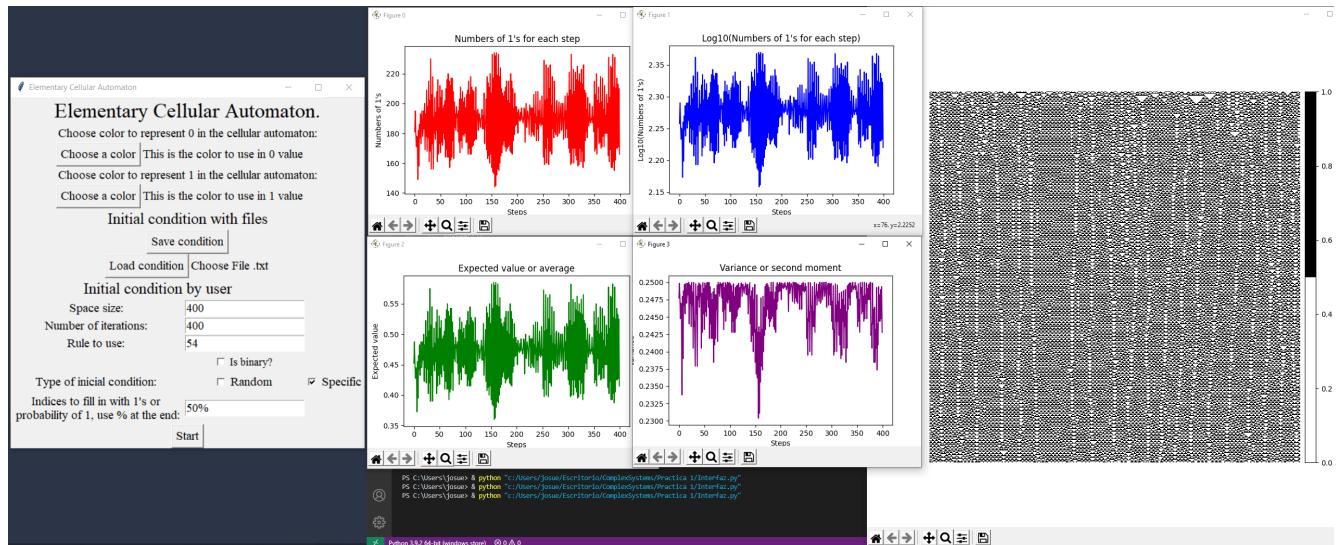


Figura 45: ECA de 400 células por 400 iteraciones con 50 % de probabilidad de 1's, regla 54

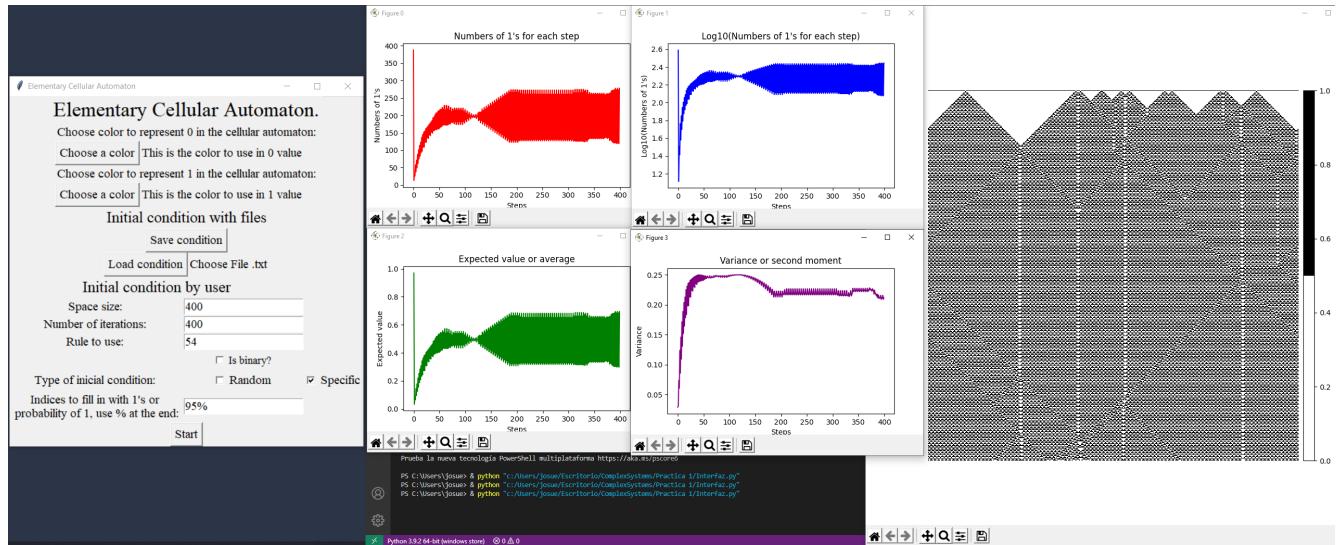


Figura 46: ECA de 400 células por 400 iteraciones con 95 % de probabilidad de 1's, regla 54

Con esta regla si observamos un cambio en nuestras 3 pruebas que las 3 son diferentes entre si, pero, si observamos la figura 22 y la 23 encontramos una similitud un poco sutil en nuestro ECA y es que ambos generan lineas rectas que "dividen" a nuestro ECA de formar vertical, caso contrario a la figura 21 que podríamos decir que esa división pasa a su forma horizontal

Pasando con la penúltima regla a probar la cual es la regla 110.

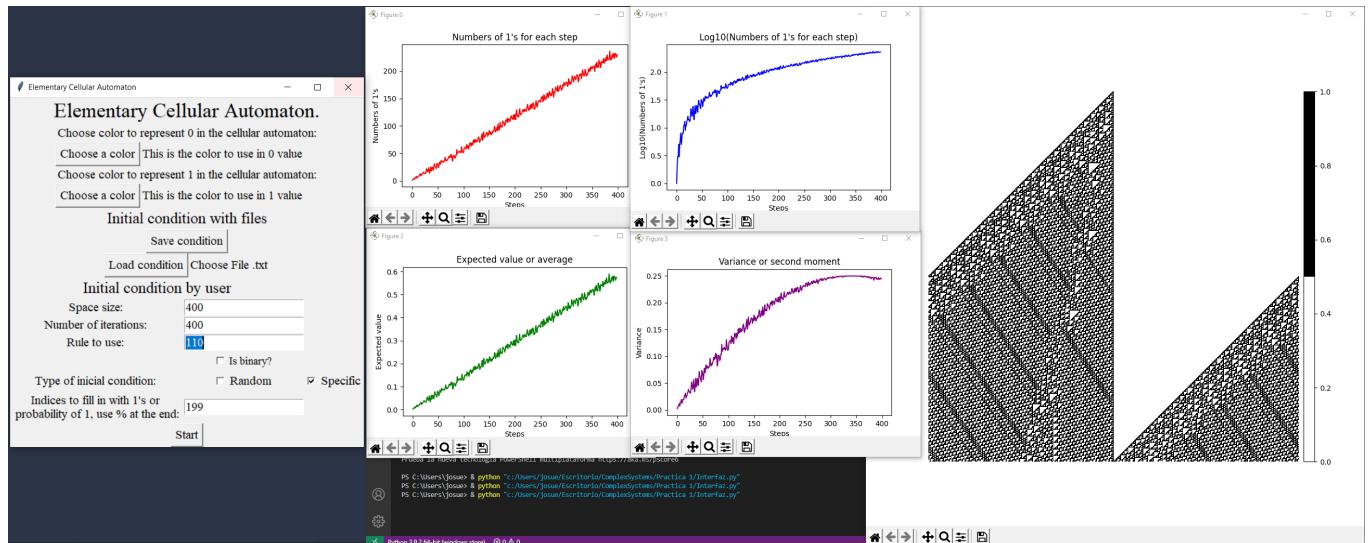


Figura 47: ECA de 400 células por 400 iteraciones con una célula inicial central, regla 110

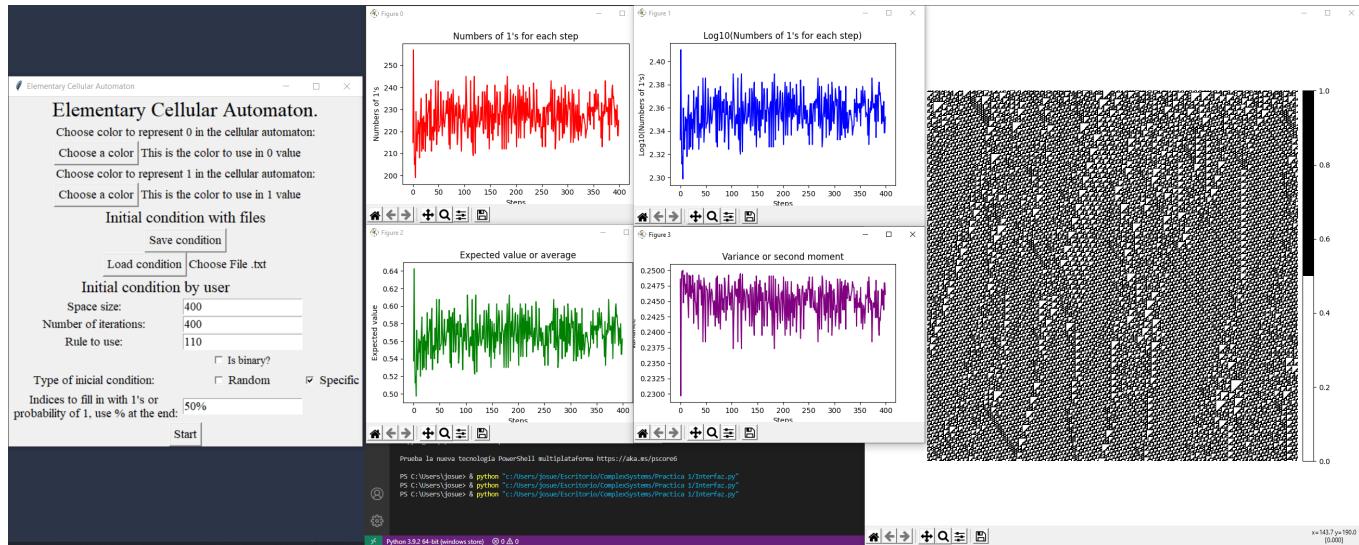


Figura 48: ECA de 400 células por 400 iteraciones con 50 % de probabilidad de 1's, regla 110

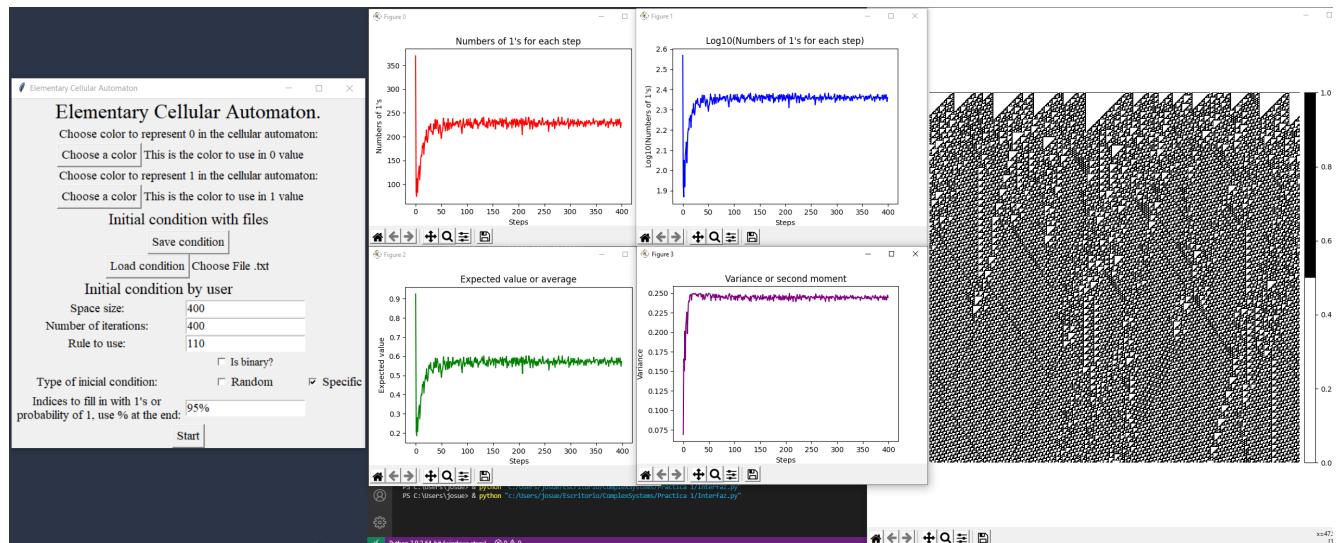


Figura 49: ECA de 400 células por 400 iteraciones con 95 % de probabilidad de 1's, regla 110

Sin duda es la regla con un comportamiento muy interesante, en la figura 24 vemos un crecimiento lineal en los números de células con valor 1, caso contrario de las figuras 25 y 26 las cuales tienen un comportamiento diferente la cual nos recuerda al crecimiento logarítmico, pero, mencionar que la figura 25 no se visualiza tan claramente este comportamiento, de hecho pareciera que estuvieramos algún tipo de filtro en las gráficas de la figura 25 y así obtenemos las gráficas de la figura 26.

Finalmente, tenemos la regla 126.

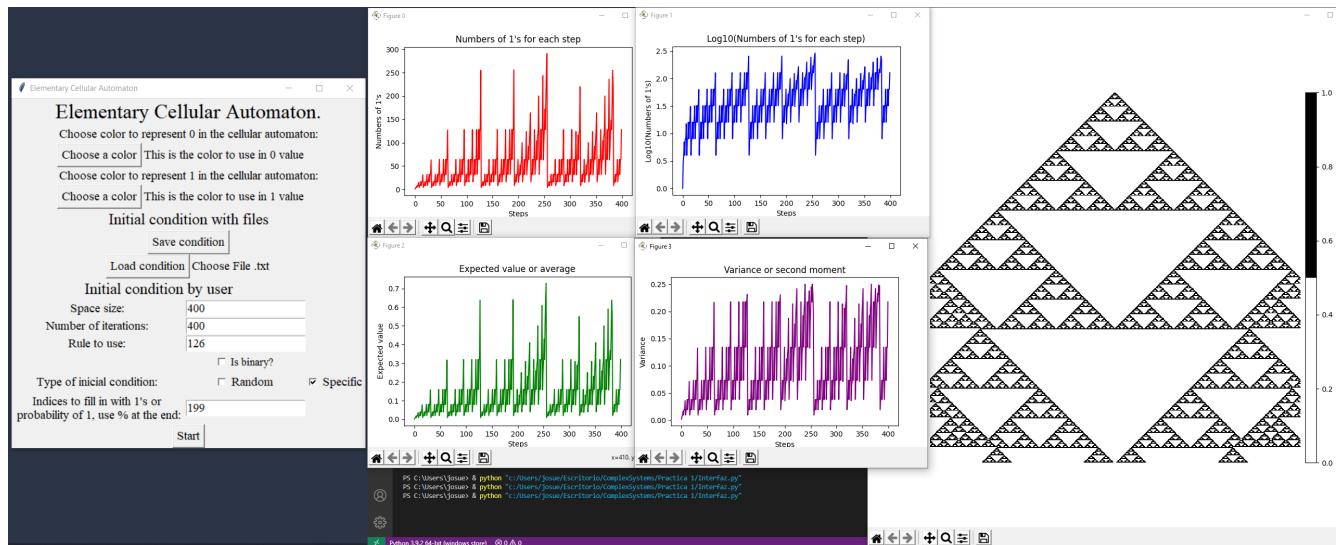


Figura 50: ECA de 400 células por 400 iteraciones con una célula inicial central, regla 126

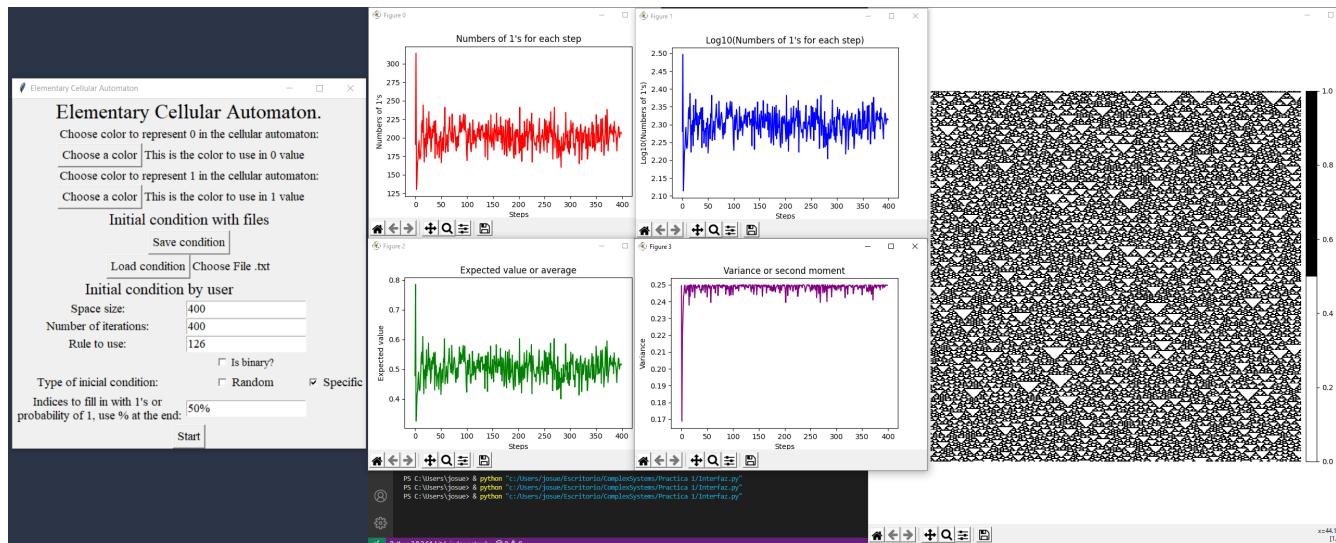


Figura 51: ECA de 400 células por 400 iteraciones con 50% de probabilidad de 1's, regla 126

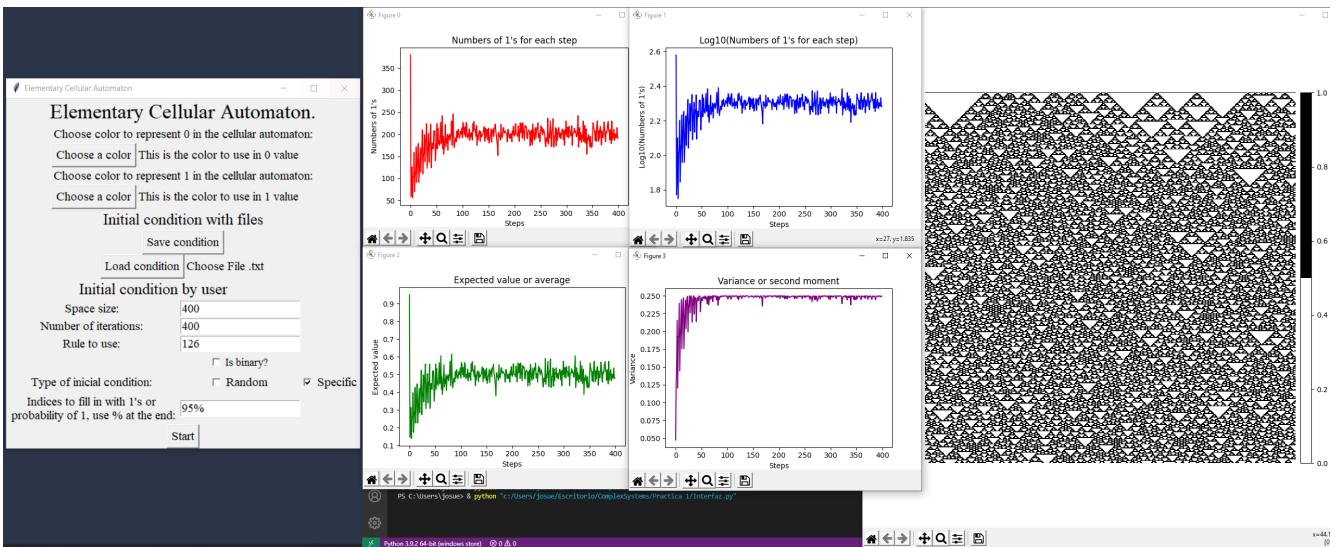


Figura 52: ECA de 400 células por 400 iteraciones con 95 % de probabilidad de 1's, regla 126

En el caso de la regla 126 nos recuerda a la regla 22 y es que las imágenes resultantes son muy parecidas, sin embargo, en los casos del 50 % y 95 % vemos como ahora las gráficas son mas definidas en contraste a la regla 22 y que de hecho se parecen a gráficas anteriores en donde mencionamos que se nos recuerda a un crecimiento logarítmico.

### 4.3. Añadidos

Como añadidos al programa se agregaron 3 tipos de gráficas, las cuales son las siguientes:

- Gráfica de logaritmo base 10 de la cantidad de células con valor 1 por cada iteración.
- Gráfica del valor esperado o promedio.
- Gráfica de la varianza o segundo momento.

Finalmente para la gráfica de la varianza o segundo momento se usa la siguiente formula para el caso discreto.

$$\sigma^2 = E[(x - \mu)^2]$$

La cual puede ser reducida mediante desarrollo matemático quedando de la siguiente manera:

$$\begin{aligned} \sigma^2 &= E[(x - \mu)^2] = E[x^2 - 2x\mu + \mu^2] \\ \sigma^2 &= E[x^2] - E[2x\mu] + E[\mu^2] \\ \sigma^2 &= E[x^2] - 2\mu E[x] + \mu^2 \\ \sigma^2 &= E[x^2] - 2\mu\mu + \mu^2 \\ \sigma^2 &= E[x^2] - 2\mu^2 + \mu^2 \end{aligned}$$

Finalmente tenemos que:

$$\sigma^2 = E[x^2] - \mu^2$$

Por lo que, retomando lo mencionado para el desarrollo de la gráfica 2 o la gráfica de nuestro promedio o valor esperado solo necesitamos el valor de  $f(x = 1)$ , nuestra formula para la varianza y finalmente saber que  $E[x^n] = \sum_x x^n * f(x)$ , así, sustituyendo valores obtendríamos lo siguiente sera:

$$\sigma^2 = (1^2 * f(x = 1)) - f(x = 1)^2 = f(x = 1) - f(x = 1)^2$$

Finalmente la formula a usar para la varianza:

$$\sigma^2 = f(x = 1) - f(x = 1)^2$$

Ahora hagamos una prueba, en la cual la regla a usar sera la regla 30, un tamaño de 1000 células con 1000 iteraciones, colores por defecto y asignación de células con valor 1 de manera aleatoria.

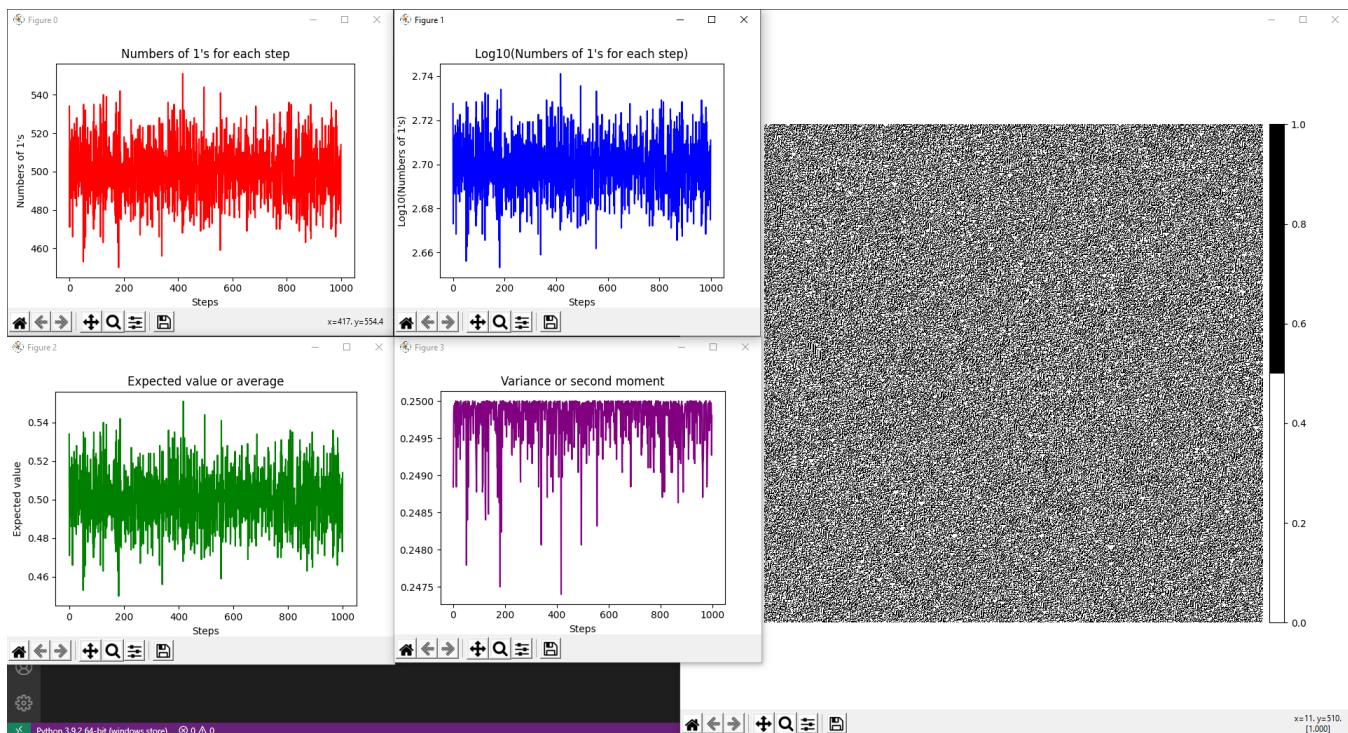


Figura 53: ECA de 1000 células por 1000 iteraciones creado de forma aleatoria

Como podemos observar tenemos 3 graficas que tienen una forma similar o casi iguales las cuales son la cantidad de 1's por iteración,  $\log$  de la cantidad de 1's y el valor esperado, esto es lógico ya que solo estamos cambiando la "escala" de una misma representación, la única gráfica que difiere de las demás es la varianza y esto resulta, como dato adicional al momento de usar diferentes reglas el comportamiento de esta difiere e incluso en ocasiones llega a ser similar con las otras 3. Como nota adicional mencionar que a cada gráfica le podemos incrementar el tamaño de la gráfica para poder visualizar de una mejor visualización de los datos, lo anterior se realiza exactamente igual que a como se menciono unas hojas atrás.

Adicionalmente a lo anterior, la barra de opciones que tiene cada gráfica nos permite realizar diferentes cosas que en seguida enlistaremos, para comodidad tomar como referencia la figura 12 y ver los iconos de izquierda a derecha.

1. El icono de la casa nos permite reajustar la vista de nuestra gráfica a como lo era originalmente.

2. El icono de la flecha a la izquierda o derecha nos permite navegar entre las distintas vistas que hayamos realizado en una gráfica, es decir, si hicimos dos aumentos y queremos regresar al primero hacemos de la flecha hacia la izquierda y en caso de que estando en el primer aumento queremos regresar al segundo hacemos uso de la flecha a la derecha.
3. El icono que nos recuerda a una cruz nos permite mover la gráfica a voluntad, un ejemplo de uso seria que al hacer zoom en un área y querer mover ese mismo zoom en un área que tenemos a lado hacemos uso de esta función para poder mover nuestra gráfica.
4. El icono de la lupa como ya se ha mencionado nos permite aumentar el tamaño de cierta área en nuestra gráfica.
5. El icono que nos recuerda a configuración no recomiendo usarlo ya que cambia la configuración inicial de nuestra gráfica y en ocasiones nos arruina el como se ve la gráfica
6. El ultimo icono, que es el icono de disquete nos permite, como es incluso lógico, guardar la gráfica como una imagen con varias extensiones a elegir, esto nos permite poder guardar resultados como una imagen para el posterior uso.



Figura 54: Barra de opciones

#### 4.4. Código

A continuación se anexa el código generado para la creación de este primer programa. Primero tenemos la parte de la interfaz de nuestro programa que se crea con el siguiente código.

Por ultimo tenemos el código que nos proporciona la parte lógica del programa y que ademas es el encargado de hacer las gráficas, este código es el cerebro de nuestro programa.

### 5. Conclusiones

Esta programa es bastante interesante, ya que es increíble ver al cambiar los valores de las células de nuestra iteración cero cambia por mucho el comportamiento de nuestro ECA que se podria pensar que no es tan significativo pero eso es un pensamiento errado ya que vemos que cambia demasiado, podríamos llegar a decir que una célula de mas o de menos provoca un caos en nuestro ECA y por tanto un ECA diferente cada vez que cambiemos nuestra iteración inicial. Finalmente solo me queda comentar que al haber ya probado diferentes reglas con la misma iteración inicial el resultado es muy diferente con cada regla.

### Referencias

- Weisstein, E., 2021. Elementary Cellular Automaton - from Wolfram MathWorld. [online] Mathworld.wolfram.com. Available at: <https://mathworld.wolfram.com/ElementaryCellularAutomaton.html> [Accessed 15 March 2021].
- Lees, E., 2020. Elementary Cellular Automata . Matplotlibblog. [online] Matplotlib.org. Available at: <https://matplotlib.org/matplotlibblog/posts/elementary-cellular-automata/> [Accessed 15 March 2021].
- Caparrini, F., 2016. Autómatas Celulares - Fernando Sancho Caparrini. [online] Cs.us.es. Available at: <http://www.cs.us.es/~fsancho/?e=66> [Accessed 15 March 2021].