

Deber Seminario

Autómatas celulares de una dimensión y reglas de Wolfram

Fausto Fabian Crespo Fernandez

Los autómatas celulares de una dimensión son un arreglo de celdas lineal infinito que evolucionan de manera discreta en el tiempo y en el que cada celda tienen un número finito de estados. La evolución está dada por reglas que dependen del valor de la celda y de los valores de las celdas vecinas y dichas reglas no dependen de la celda ni del tiempo o sea son fijas (por tanto el proceso es determinístico a diferencia de los autómatas celulares probabilísticos o estocásticos donde el nuevo estado de una celda es escogido de manera probabilística a partir de una distribución de probabilidad y se actualizan las celdas de manera paralela o de manera sincrónica). Las actualizaciones de los estados de las celdas se hacen simultáneamente, o sea no se toma en cuenta el cambio del estado o valor de una celda vecina, realizada en un paso dado, para modificar el valor de una celda en el mismo paso, o sea los cambios en la celda son tomados en cuenta por las reglas en el próximo paso.

En el caso más simple de dos estados 0 y 1 y que las reglas solo dependen del valor de la celda y de las vecinas inmediatas (izquierda y derecha), a cada combinación de 0 y 1 de 3 celdas contiguas (2^3 combinaciones) la regla le puede asignar 0 o 1, o sea el número posible de reglas es: $2^{2^3} = 2^8 = 256$ posibles reglas distintas. A estas se les conoce como reglas de Wolfram.

Por ejemplo, para la regla número $30 = (11110)_2$, corresponde a

$$1 \ 1 \ 1 \rightarrow 0$$

$$1 \ 1 \ 0 \rightarrow 0$$

$$1 \ 0 \ 1 \rightarrow 0$$

$$1 \ 0 \ 0 \rightarrow 1$$

$$0 \ 1 \ 1 \rightarrow 1$$

$$0 \ 1 \ 0 \rightarrow 1$$

$$0 \ 0 \ 1 \rightarrow 1$$

$$0 \ 0 \ 0 \rightarrow 0$$

Como se ve si se guarda los dígitos de la representación en base 2 del número de la regla en un arreglo de derecha a izquierda, o sea

Digito(der a izq)	0	1	1	1	1	0	0	0
Posición	0	1	2	3	4	5	6	7

Rep. base 2 de Posición y además valores de 3 celdas contiguas	000	001	010	011	100	101	110	111
-------------------------------------------------------------------------------------	-----	-----	-----	-----	-----	-----	-----	-----

Y la regla se puede obtener evaluando el arreglo que contiene los dígitos de la representación en base 2 del número de la regla colocados de derecha a izquierda en la posición que es la representación en base 10 (de 0 a 7) que corresponde al número en base 2 formado por los 3 valores de las celdas consecutivas, por ejemplo regla_30 [100]=1 que es el valor de la posición 4 del arreglo con los dígitos de 30 en base 2 (de derecha a izquierda).

Además para llevar a un número de decimal a base 2 debemos dividir el número por 2, el primer resto es el primer dígito de derecha a izquierda, luego dividir el cociente entre 2 y el resto es el segundo dígito de derecha a izquierda y así sucesivamente. Esto equivale a tomar sucesivamente los cocientes del número al dividirlos por potencias de $2(2^0, 2^1, \dots, 2^8)$ y hallar sus restos al dividir por 2 son los dígitos de derecha a izquierda. O sea en python

```
regla = [(noregla/pow(2,i)) % 2 for i in range(8)]
```

En Python se implementó la evolución de autómatas celulares a partir de un estado inicial de un único 1 o de un estado inicial aleatorio y de un número de regla determinado en decimal. Se generó una imagen .png con la evolución del autómata.

Ejemplo de evolución de autómatas a partir de estado inicial con un único 1:

1-Regla 30

```
alturamagen = 500
```

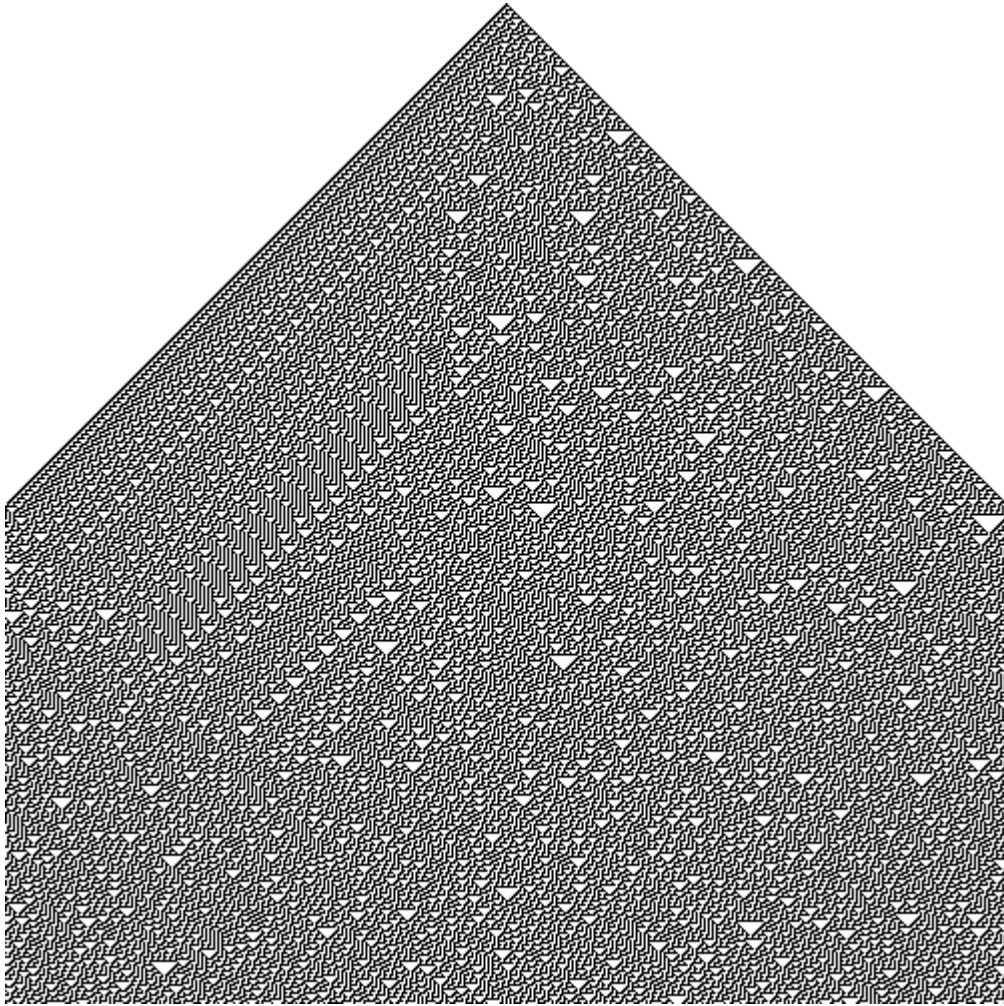
```
ancholmagen = 500
```

```
aleatorio = 0
```

```
noregla = 30
```

```
nombre_archivo= "regla_" +str(noregla) + ".png"
```

```
ReglaWolfram(noregla, alturamagen,ancholmagen,aleatorio, nombre_archivo)
```



2-Regla 50

alturamagen = 500

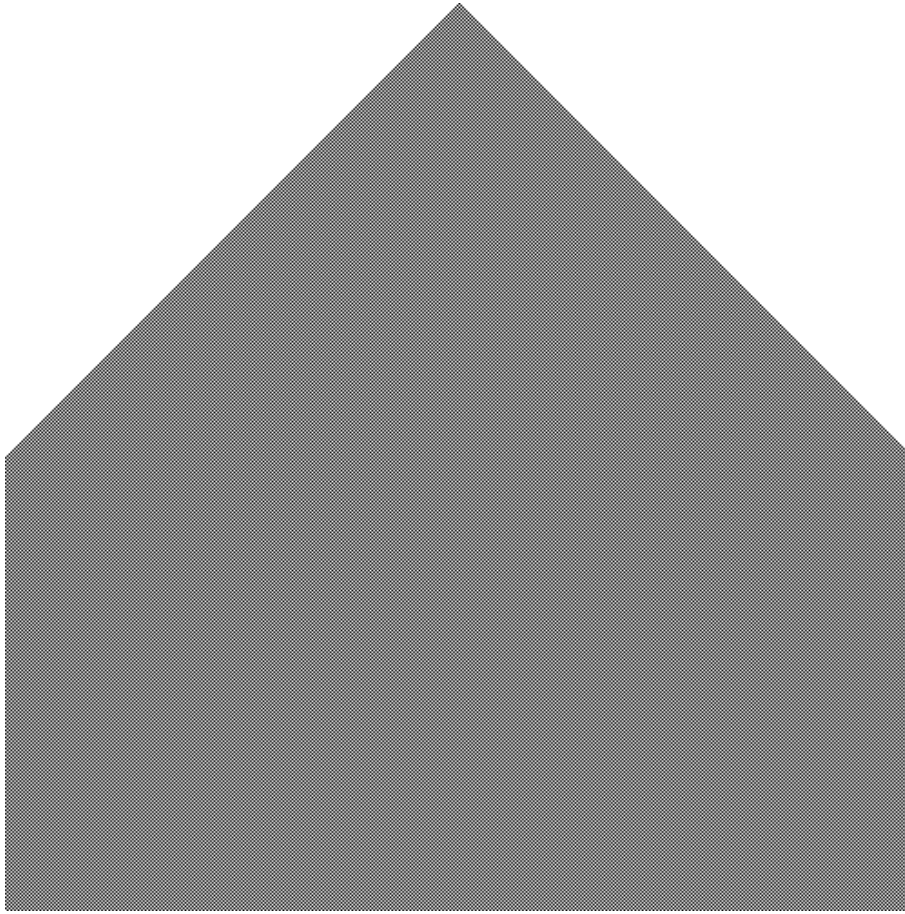
ancholmagen = 500

aleatorio = 0

noregla = 50

nombre_archivo= "regla_" +str(noregla) +".png"

ReglaWolfram(noregla, alturamagen,ancholmagen,aleatorio, nombre_archivo)



Existen reglas que evolucionan como una única recta partir de la configuración inicial con un único 1, por ejemplo la regla $n=184$ o $n=2=2^1(0 \rightarrow 0 \text{ y } 1 \rightarrow 1 \text{ y } 0 \text{ en las demás configuraciones})$ o sea el 1 de la derecha se desplaza una posición a la izquierda y por tanto da una recta con pendiente 1. Pero como la imagen generada tiene un ancho y largo fijos aparece como dos rectas, en realidad la segunda es la continuación de la primera

3-Regla 184

alturamagen = 500

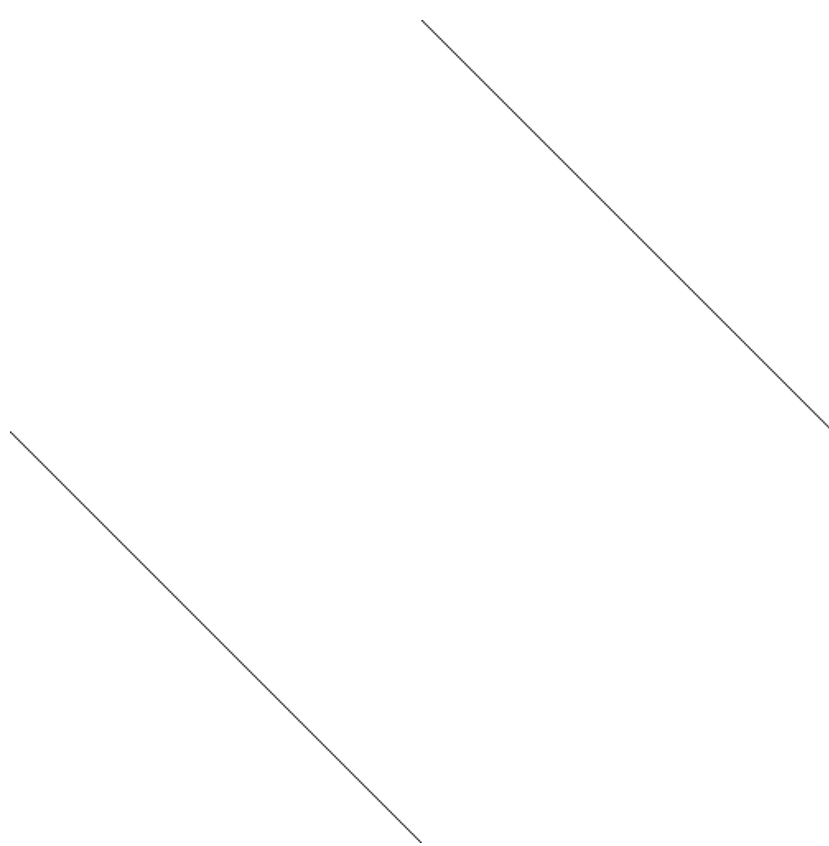
ancholmagen = 500

aleatorio = 0

noregla = 184

nombre_archivo= "regla_" +str(noregla) +".png"

ReglaWolfram(noregla, alturamagen,ancholmagen,aleatorio, nombre_archivo)



4-Regla 2

alturamagen = 500

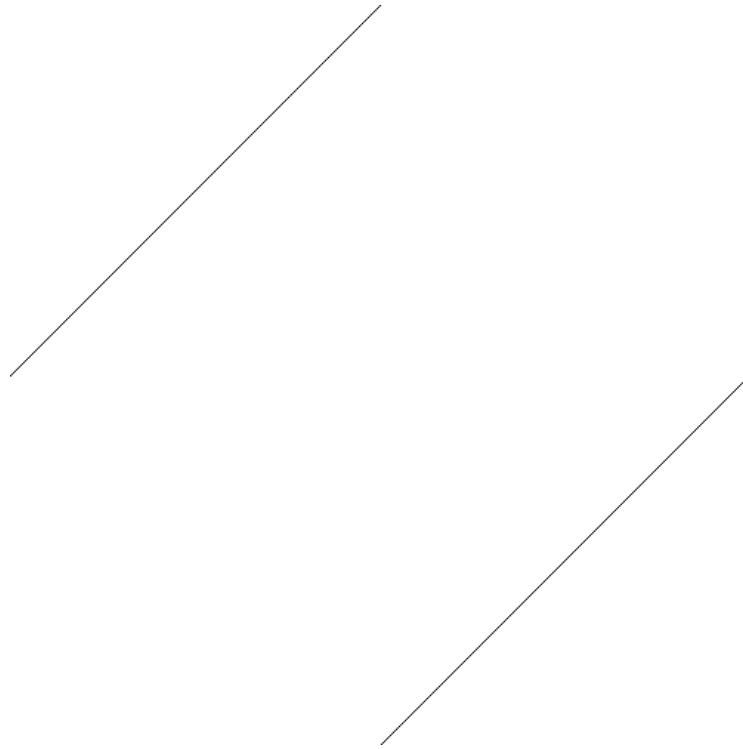
ancholmagen = 500

aleatorio = 0

noregla = 2

nombre_archivo= "regla_" +str(noregla) + ".png"

ReglaWolfram(noregla, alturamagen,ancholmagen,aleatorio, nombre_archivo)



5-Regla 127

alturamagen = 500

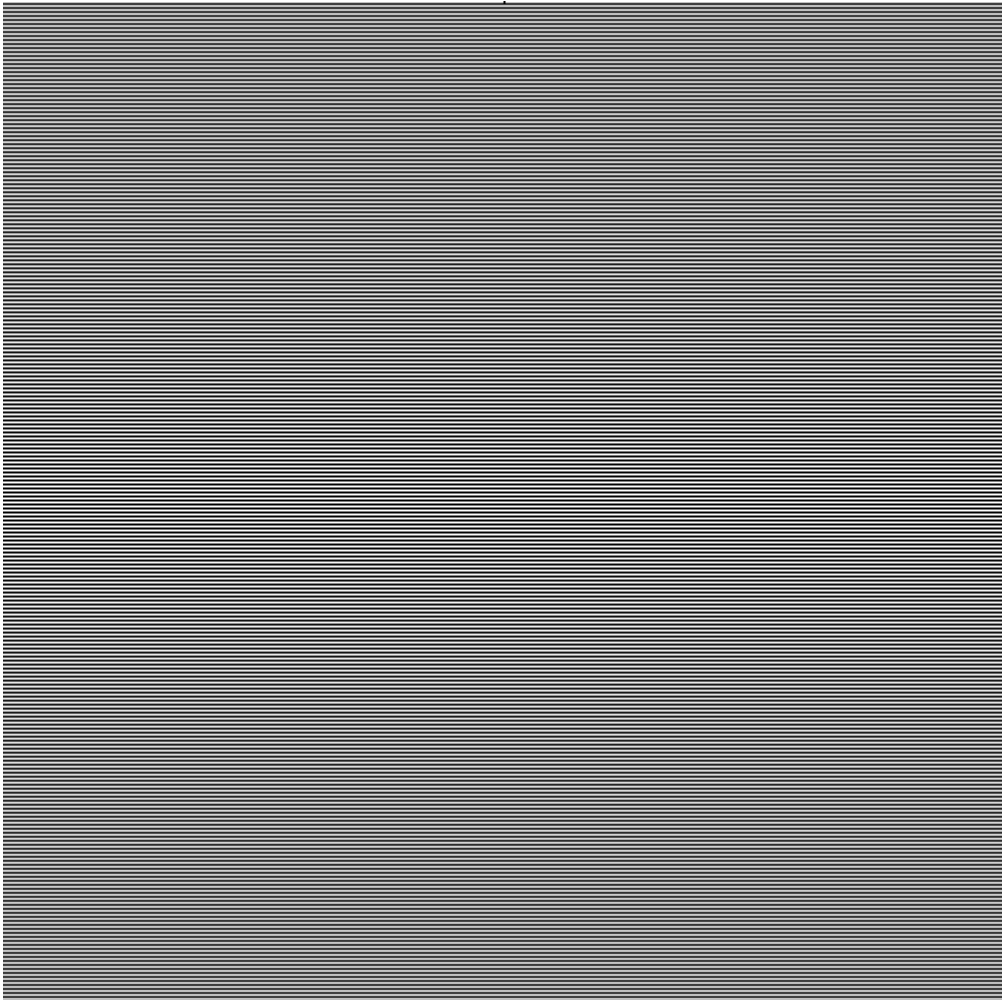
ancholmagen = 500

aleatorio = 0

noregla = 127

nombre_archivo= "regla_" +str(noregla) + ".png"

ReglaWolfram(noregla, alturamagen,ancholmagen,aleatorio, nombre_archivo)



6-Regla 64(una imagen en blanco)

alturaimagen = 500

ancholimagen = 500

aleatorio = 0

noregla = 64

nombre_archivo= "regla_" +str(noregla) +".png"

ReglaWolfram(noregla, alturaimagen,ancholimagen,aleatorio, nombre_archivo)

En general comenzando desde un estado inicial con un único 1, las reglas que solo tienen 1 con configuraciones que incluyen dos 1 adyacentes no van a evolucionar pues no hay dos 1 adyacentes en el estado inicial, esto ocurre con $n=64=2^6$ (en las demás configuraciones $1 \ 1 \ 0 \rightarrow 1 \text{ y } 0$ en las demás configuraciones), $n=8=2^3(0 \ 1 \ 1 \rightarrow 1 \text{ y } 0)$, $n=128=2^7(1 \ 1 \ 1 \rightarrow 1 \text{ y } 0$ en las demás configuraciones). Esto no pasa si los estados iniciales son aleatorios, ejemplo para la regla 128.

alturaimagen = 500

ancholimagen = 500

aleatorio = 1

noregla = 128

nombre_archivo= "regla_" +str(noregla) +".png"

ReglaWolfram(noregla, alturaimagen,ancholimagen,aleatorio, nombre_archivo)

También para las reglas de número $n=2^p - 1 = (1..1)_2$ todas las configuraciones iniciales (con un único 1 o aleatorias) evolucionan hacia 1 en todas las posiciones y por eso su imagen son líneas horizontales paralelas.

7-Regla 90

alturaimagen = 500

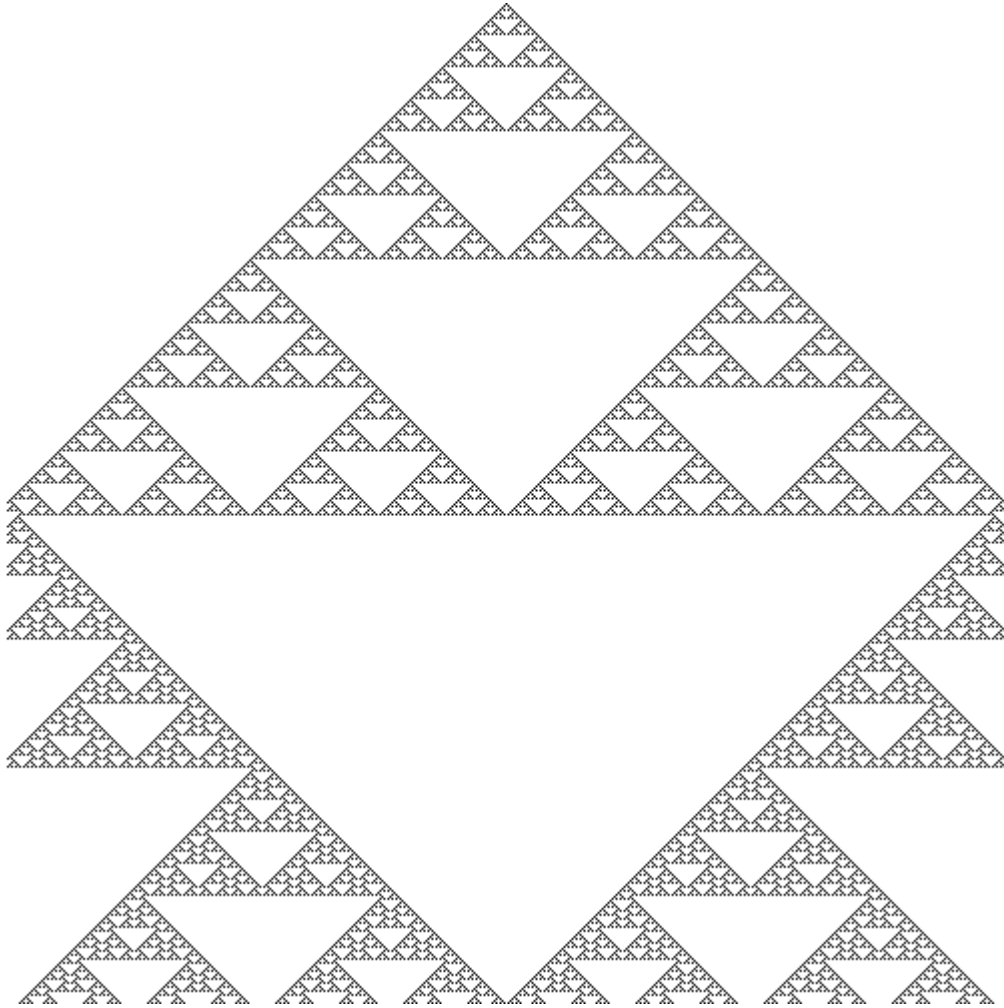
ancholimagen = 500

aleatorio = 0

noregla = 90

nombre_archivo= "regla_" +str(noregla) +".png"

ReglaWolfram(noregla, alturaimagen,ancholimagen,aleatorio, nombre_archivo)



Código en Python:

```
from random import randint
import Image, ImageDraw

def genera_datos(alturaImagen, anchoImagen, aleatorio, noregla):
    if aleatorio:
        primera_fila = [randint(0,1) for i in range(anchoImagen)]
    else:
        primera_fila = [0]*anchoImagen
        primera_fila[anchoImagen/2] = 1
    resultados = [primera_fila]

    regla = [(noregla/pow(2,i)) % 2 for i in range(8)]

    for i in range(alturaImagen-1):
        data = resultados[-1]
```

```
nueva_linea = [regla[4*data[(j-1)%anchoImagen]+2*data[j]+data[(j+1)%anchoImagen]]
for j in range(anchoImagen)]
resultados.append(nueva_linea)
return resultados
```

```
def genera_imagen(datos,alturaImagen,anchoImagen,nombre_archivo="regla.png"):
imagen = Image.new("RGB",(anchoImagen,alturaImagen),(255,255,255))
draw = ImageDraw.Draw(imagen)
```

```
for y in range(alturaImagen):
for x in range(anchoImagen):
if datos[y][x]: draw.point((x,y),(0,0,0))
imagen.save(nombre_archivo,"PNG")
return
```

```
def ReglaWolfram(noregla, alturaImagen,anchoImagen,aleatorio, nombre_archivo ):
datos= genera_datos(alturaImagen,anchoImagen,aleatorio,noregla)
genera_imagen(datos,alturaImagen,anchoImagen,nombre_archivo)
```

```
alturaImagen = 500
anchoImagen = 500
aleatorio = 0
noregla = 90
```

```
nombre_archivo= "regla_" +str(noregla) +".png"
ReglaWolfram(noregla, alturaImagen,anchoImagen,aleatorio, nombre_archivo)
aleatorio = 0
noregla = 126
```

```
nombre_archivo= "regla_" +str(noregla) +".png"
ReglaWolfram(noregla, alturaImagen,anchoImagen,aleatorio, nombre_archivo)
```