

El Juego de La Vida - John Horton Conway

Juanita Gómez

Matemática Computacional

Pontificia Universidad Javeriana

27 de mayo de 2016

1. Introducción: El Juego de la Vida

Los matemáticos se han preocupado por diseñar herramientas que permitan modelar fenómenos de la vida real a través de modelos matemáticos. La computación es uno de los recursos mas utilizados para dicho propósito dado que permiten representar algoritmos que muestren la evolución de dichos fenómenos.

El Juego de la vida es un juego de cero jugadores que fue diseñado en 1970 por John Horton Conway, matemático británico. Se dice que dicho juego es un automata celular, ya que es un modelo matemático de un sistema dinámico que va evolucionando en pasos discretos, es decir unidades de tiempo cuantificables con valores enteros en intervalos regulares. Consiste en conjunto de celdas o células que adquieren distintos valores que cambian de acuerdo a expresiones matemáticas que están determinadas por los estados de las células vecinas. Ha sido encasillado además, en la categoría de los juegos de simulación porque pretende imitar procesos de la vida real.

1.1. J.H. Conway

John H. Conway es uno de los teóricos más prominentes en el estudio de los grupos finitos y uno de los teóricos de nudos más importantes del mundo. Él ha escrito o co-escrito más de diez libros y más de ciento treinta y artículos de revistas sobre una amplia variedad de temas matemáticos. Ha realizado un trabajo importante en la teoría de números, la teoría de juegos, teoría de codificación, en la creación de nuevos sistemas de numeración, incluyendo los "números surreales" es ampliamente conocido como el inventor del "juego de la vida". Nacido en 1937, Conway recibió su Ph.D. en 1967 de la Universidad de Cambridge donde estuvo hasta 1986 a la Universidad de Princeton, donde conoció a John von Neumann, matemático que impulso la creación de su "Juego de La Vida"

1.2. Origen

El matemático John Horton Conway, diseñador del juego, pretendía resolver un problema que fue presentado en los años cuarenta por John Von Newman matemático húngaro quien intentó crear una máquina hipotética que fuera capaz de construir copias de si misma. Neumann llegó a un modelo matemático de la maquina que se desarrollo en un tablero cuadriculado con varias reglas complejas. Inicialmente esto se interpretó como un conjunto de células que crecían, se reproducían y morían con el paso del tiempo por lo que fueron conocidas como autómata celular. El propósito de Conway era entonces simplificar el modelo de Neumann. Su idea era iniciar con una configuración simple de células y observar como cambian de acuerdo a unas reglas determinadas. Dichas reglas fueron escogidas después de un largo periodo de experimentación, en el que básicamente quería lograr lo siguiente:

- No debe haber un patrón inicial para el cual haya una prueba simple de que la población puede crecer sin limite.
- Debe hacer patrones iniciales que aparenten un crecimiento sin limite

- Debe haber patrones iniciales simples que crezcan y cambien por un periodo considerable de tiempo antes de terminarse ya sea por la muerte de todas las células o por el establecimiento de las células que terminan en oscilaciones constantes.

Básicamente, lo que Conway buscaba era que sus reglas hicieran que el comportamiento de la población fuera impredecible. De esta forma, invento el Juego de la Vida en 1970 y fue mostrado al público por primera vez en Octubre de ese año, por medio de un artículo que se publicó en la revista Scientific American, en la columna Mathematical Games de Martin Gardner. Una vez publicado, atrajo rápidamente la atención de quienes lo conocían debido a el surgimiento de patrones tan complejos a partir de reglas sencillas. Adicionalmente, llamaba la atención su similitud con algunos de los procesos evolutivos que determinan el nacimiento y decadencia de las sociedades de seres vivos.

2. Método: Las Reglas del Juego

El Juego de la vida, se desarrolla en teoría un tablero infinito que se conoce como "mundo", pero en la práctica ha sido desarrollado en un tablero con dimensiones fijadas, que se divide en celdas mediante una retícula.

Cada una de las celdas contiene lo que es conocido como célula es rodeada por 8 casillas, 2 que se encuentran adyacentes lateralmente, 2 adyacentes verticalmente y 4 adyacentes en forma diagonal. Estas 8 casillas se conocen como "vecindad", la cual determina el estado de la célula en la siguiente generación. De esta manera, el juego está basado en la evolución de los estados sucesivos de las células, en los cuales las condiciones de un estado dependen solamente de las condiciones del estado anterior. Por lo mismo, no se requiere de la entrada de datos durante el desarrollo del juego, sino que el estado inicial del mismo, es lo que determina su evolución. La participación de un usuario en el juego sólo consiste en la determinación de su estado inicial, creando lo que se denomina "población inicial" o "generación cero".

Los posibles estados de cada una de las células del juego son, célula viva que se puede considerar un "uno lógico" o célula muerta considerada como el "cero lógico". El estado de cada célula depende de su estado actual y del estado actual de sus 8 células vecinas, siguiendo unas reglas. Una célula al morir deja vacía la celda que ocupaba.

Las reglas establecidas por Conway para el desarrollo del juego fueron:

- Supervivencia: Las células vivas que tengan 2 ó 3 vecinos vivos sobreviven a la siguiente generación (su estado se mantiene inalterado)
- Fallecimiento: Las células vivas que tengan menos de 2 vecinos fallecen por aislamiento o soledad y aquellas que tengan más de tres células vecinas vivos mueren por superpoblación en el siguiente turno.
- Nacimiento: Una célula muerta (vacía) que tenga exactamente 3 células vecinas vivas, se convertirá en una célula viva en el siguiente turno (nacimiento de nuevo individuo).

Una vez establecido el estado inicial de las células, se realizan iteraciones de acuerdo a las reglas establecidas, dándose de esta manera, la evolución de la población en la cual se pueden dar los siguientes resultados:

- Extinción: Después de un numero finito iteraciones se desaparecen todos los miembros de la población.
- Estabilización: Después de un numero finito iteraciones, la población queda estabilizada de dos maneras: Las células restantes permanecen en su estado de manera constante o bien puede suceder que las formas de las células restantes queden oscilando.
- Crecimiento constante: La población crece turno tras turno y se mantiene un número infinito de generaciones.

3. Implementación en Python y Matlab

En la presente sección se pretende mostrar el código realizado para modelar dicho fenómeno, en Python y en Matlab, dos herramientas de software que permiten visualizar la evolución del juego. En ambas se utilizó el mismo algoritmo, con modificaciones sólo en cuanto al lenguaje de las plataformas. Por esto, se explicara cada una de las secciones del código sólo en uno de los IDE, junto con su funcionamiento.

3.1. Matlab

A continuación se presenta el código utilizado en Matlab para diseñar el programa del juego de la vida.

```
1 Titulo = [ '////////=====\\\\\\\\\\n'...
2           '//////// WELCOME TO          \\\\\\\\\\\n'...
3           '//////// THE GAME OF LIFE     \\\\\\\\\\\n'...
4           '//////// Presentado por: Juanita Gomez \\\\\\\\\\\n'...
5           '////////=====\\\\\\\\\\n'];
6
7
8
9
10 separator = '\\n////////=====\\\\\\\\\\n';
11 fprintf(Titulo);
12 fprintf('\\nPresentado Por: Juanita Gomez');
13 r = input('\\nEscoja la probabilidad de que una celula inicie viva: ');
14 m = input('\\nEscoja el tamaño de la matriz cuadrada a usar: ');
15 state = 0;
16 a = '#';
17 b = '.';
18 %Creación de matriz
19 TheGameOfLife=char(ones(m));
20 for i=1:m
21     for j=1:m
22         y = randi([0 100],1,1);
23         if (y>r)
24             TheGameOfLife(i,j)=b;
25         else
26             TheGameOfLife(i,j)=a;
27         end
28     end
29 end
30 %Evolución de la matriz (Ciclo infinito)
31 while 1>0
32     clc;
33     fprintf(Titulo);
34     fprintf('\\nUd óescogi una probabilidad inicial de vida del: %d ',r);
35     fprintf('\\nUd óescogi una matriz cuadrada de tamaño: %d', m);
36     fprintf(separator);
37     display(TheGameOfLife);
38     fprintf(separator);
39     fprintf('\\nState: %d', state);
40     fprintf('\\nGracias por Jugar\\nPara finalizar el juego oprima (ctrl + c)\\nHasta la óproxima\\n');
41     state = state + 1;
42     pause(0.1);
43     %Cada una de las iteraciones del programa
44     for i=1:m
45         for j=1:m
46             v=0;
47             if i+1<m+1
48                 if TheGameOfLife(i+1,j)==a
49                     v=v+1;
50                 end
51                 if j+1<m+1
52                     if TheGameOfLife(i+1,j+1)==a
53                         v=v+1;
54                     end
55                 end
56                 if j-1>0
57                     if TheGameOfLife(i+1,j-1)==a
58                         v=v+1;
59                     end
60                 end
61             end
62             if i-1>0
63                 if TheGameOfLife(i-1,j)==a
64                     v=v+1;
65                 end
66                 if j+1<m+1
67                     if TheGameOfLife(i-1,j+1)==a
68                         v=v+1;
69                     end
70                 end
71                 if j-1>0
72                     if TheGameOfLife(i-1,j-1)==a
73                         v=v+1;
74                     end
75                 end
76             end
77             if j+1<m+1
78                 if TheGameOfLife(i,j+1)==a
79                     v=v+1;
80                 end
81             end
82             if j-1>0
83                 if TheGameOfLife(i,j-1)==a
84                     v=v+1;
85                 end
86             end
87             %Encontrar células vivas
```

```

89         if TheGameOfLife(i,j)==a
90             %Una celula viva con menos de dos vecinos vivos muere
91             if v<2
92                 TheGameOfLife(i,j)=b;
93             end
94             %Una celula viva con 2 o 3 vecinos vivos permanece en su estado
95             if (v==2||v==3)
96                 TheGameOfLife(i,j)=a;
97             end
98             %Una celula viva con mas de 3 vecinos muere
99             if v>3
100                 TheGameOfLife(i,j)=b;
101             end
102         end
103         %Encontrar células muertas
104         if TheGameOfLife(i,j)==b;
105             %Una celula muerta con exactamente 3 vecinos vivos se convierte en viva
106             if v==3
107                 TheGameOfLife(i,j)=a;
108             end
109         end
110     end
111 end
112 end
113 end

```

3.1.1. Título y configuración

En esta parte del código, se muestran los títulos del programa y se da una bienvenida al usuario para que empiece el juego. Como se mencionó anteriormente la participación del usuario en el juego sólo consiste en la determinación de su estado inicial, creando la “población inicial”. Para esto, se han implementado dos parámetros: El primero, es el tamaño de la matriz cuadrada en la cual se va a desarrollar, y el segundo es la probabilidad de que una celula se encuentre viva en un estado inicial. Esto quiere decir que el usuario va a determinar cuantas células vivas van a aparecer en la población inicial a través de su probabilidad, por ejemplo si escogió una matriz de tamaño 10x10 y una probabilidad del 30, entonces aparecerán 30 células vivas y 70 células muertas. En esta parte del programa, también se presentan un contador 'state' que va a permitir al usuario ver el número de iteraciones que lleva el programa desde que inició. Por último se definen las variables a y b que representan los estados 'vivo' y 'muerto' de las células y se definen como caracteres para poder visualizarse en la matriz. En este caso se escogió el caracter numeral para el estado vivo y el espacio para el estado muerto.

```

114 Titulo =
115 [ '//////////=====\\n'...
116 '////////// WELCOME TO \\n'...
117 '////////// \\n'...
118 '////////// THE GAME OF LIFE \\n'...
119 '////////// \\n'...
120 '////////// Presentado por: Juanita Gomez \\n'...
121 '//////////=====\\n'];
122
123 separator = '\\n//////////=====\\n';
124
125 fprintf(Titulo);
126 fprintf('\\nPresentado Por: Juanita Gomez');
127 r = input('\\nEscoja la probabilidad de que una celula inicie viva: ');
128 m = input('\\nEscoja el tamaño de la matriz cuadrada a usar: ');
129
130 state = 0;
131 a='#';
132 b=' ';

```

3.1.2. Creación de la matriz inicial

En esta parte del código, se crea una matriz de tamaño mXm, dimensiones fijadas anteriormente por el usuario. Para determinar el estado de la matriz inicial se utiliza un ciclo en el que se recorre la matriz celda por celda y para cada una se asigna un numero aleatorio entre 0 y 100. De acuerdo a la probabilidad 'r' escogida por el usuario, si el valor aleatorio es mayor que r la célula se encontrará muerta en el estado inicial; si por el contrario el valor es menor que r, la célula estará viva en su estado inicial. De esta manera habrá 'r' células vivas en la matriz pero se encontrarán en posiciones completamente aleatorias cada vez que se corra el programa.

```

133 TheGameOfLife=char(ones(m));
134 for i=1:m
135     for j=1:m
136         y = randi([0 100],1,1);
137         if (y>r)
138             TheGameOfLife(i,j)=b;
139         else
140             TheGameOfLife(i,j)=a;
141         end
142     end
143 end

```

3.1.3. Evolución general del programa

Una vez creada la matriz inicial, empieza la evolución de las células. Este algoritmo, se presenta como un ciclo infinito que esta definido con un 'while' que es verdadero siempre (while 1>0). de modo que el programa no se detenga a menos que el usuario lo desee. Luego del while lo primero que aparece es un comando que permite "limpiar" la pantalla donde se corre el programa, de tal manera que en cada una de las iteraciones del programa, la matriz aparezca en el mismo sitio que en la iteración anterior. En esta parte del código, se presenta la información del programa al usuario: El título aparece en cada una de las iteraciones, junto con información acerca de la probabilidad y tamaño de la matriz inicial escogida por el usuario y se muestra la variable 'state' que refleja el número de iteraciones que lleva el programa desde que inició, la cual va sumando uno en cada iteración del ciclo. Se presenta también un anuncio que le informa al usuario como debe finalizar el programa, en este caso oprimiendo ctrl + c. Finalmente, lo último de esta sección es un comando que permite pausar el programa antes de cada iteración para que se pueda visualizar detenidamente la evolución del programa.

```
144 while 1>0
145     clr;
146     fprintf(Titulo);
147     fprintf('\nUd óescogi una probabilidad inicial de vida de: %d ',r);
148     fprintf('\nUd óescogi una matriz cuadrada de tamaño: %d', m);
149     fprintf(separator);
150     display(TheGameOfLife);
151     fprintf(separator);
152     fprintf('\nState: %d', state);
153     fprintf('\nGracias por Jugar\nPara finalizar el juego oprima (ctrl + c)\nHasta la óproxima\n');
154     state = state + 1;
155     pause(0.3);
```

3.1.4. Cada una de las iteraciones del programa

Cada iteración del programa, esta determinada por un ciclo que recorre cada una de las células de la matriz realizando 2 procedimientos:

- **Conteo de vecinos:** En este proceso se verifica el estado de los ocho vecinos de cada una de las células y se cuentan aquellas que se encuentran vivas a través de un contador 'v' que se inicializa al comienzo del ciclo y utilizando la comparación de los estados definidos inicialmente como 'a' y 'b'. Para realizar este conteo, es necesario tener cuidado cuando se recorren las casillas de la matriz que se encuentran en los bordes laterales o verticales de la matriz, para no salirse de los límites de la misma. Para ello, antes de comparar el estado de un vecino de una célula es necesario verificar que dicho vecino si exista dentro de los límites de la matriz. Si existe, se compara su estado con 'a' y 'b' para saber si esta vivo, si no existe, se procederá a verificar los otros vecinos.
- **Determinación del estado de la célula:** Una vez, finalizado el conteo de los vecinos vivos de la célula se verifica si dicha célula esta en su estado 'vivo' o 'muerto', pues dependiendo de esto y de su número de vecinos vivos se determinará su estado en el siguiente turno. En caso de que la célula este viva, se verifica si el número de vecinos es mayor a 3 o menor a 2, en cuyo caso la célula pasa a ser una célula muerta, cambiando su estado de 'a' a 'b'. Si el número de vecinos vivos es exactamente 2 o 3, la célula permanece en su estado actual el siguiente turno. En caso de que la célula este muerta, solo se verifica si el número de vecinos vivos es exactamente igual a 3, caso en el cual se convierte en una célula viva pasando de 'b' a 'a'. De lo contrario su estado permanece igual.

```
156 %Cada una de las iteraciones del programa
157 for i=1:m
158     for j=1:m
159         v=0;
160
161         %Conteo de vecinos
162         if i+1<m+1
163             if TheGameOfLife(i+1,j)==a
164                 v=v+1;
165             end
166             if j+1<m+1
167                 if TheGameOfLife(i+1,j+1)==a
168                     v=v+1;
169                 end
170             end
171             if j-1>0
172                 if TheGameOfLife(i+1,j-1)==a
173                     v=v+1;
174                 end
175             end
176         end
177     end
178 end
```

```

174         end
175     end
176 end
177 if i-1>0
178     if TheGameOfLife(i-1,j)==a
179         v=v+1;
180     end
181     if j+1<m+1
182         if TheGameOfLife(i-1,j+1)==a
183             v=v+1;
184         end
185     end
186     if j-1>=1
187         if TheGameOfLife(i-1,j-1)==a
188             v=v+1;
189         end
190     end
191 end
192 if j+1<m+1
193     if TheGameOfLife(i,j+1)==a
194         v=v+1;
195     end
196 end
197 if j-1>0
198     if TheGameOfLife(i,j-1)==a
199         v=v+1;
200     end
201 end
202
203 %Encontrar células vivas
204 if TheGameOfLife(i,j)==a
205
206     %Una célula viva con menos de dos vecinos vivos muere
207     if v<2
208         TheGameOfLife(i,j)=b;
209     end
210     %Una célula viva con 2 o 3 vecinos vivos permanece en su estado
211     if (v==2||v==3)
212         TheGameOfLife(i,j)=a;
213     end
214     %Una célula viva con mas de 3 vecinos muere
215     if v>3
216         TheGameOfLife(i,j)=b;
217     end
218 end
219
220 %Encontrar células muertos
221 if TheGameOfLife(i,j)==b;
222
223     %Una célula muerta con exactamente 3 vecinos vivos se convierte en viva
224     if v==3
225         TheGameOfLife(i,j)=a;
226     end
227 end
228 end
229 end

```

3.2. Python

A continuación se presenta el código de Python utilizado para el programa.

```

230
231 # -*- coding: utf-8 -*-
232 import os
233 import time
234 import random
235 import platform
236
237
238
239 Titulo = ( '//////////=====\\\\\\\\\\\\\\\\\\\\\\n'
240           '                WELCOME TO                \\\\\\\\\\\\\\\\\\\\\\\n'
241           '                THE GAME OF LIFE                \\\\\\\\\\\\\\\\\\\\\\\n'
242           '                Presentado por: Juanita Gomez                \\\\\\\\\\\\\\\\\\\\\\\n'
243           '//////////=====\\\\\\\\\\\\\\\\\\\\\\n' );
244
245 separator = ('\\n//////////=====\\\\\\\\\\\\\\\\\\\\\\n')
246
247 footer = ('Gracias por Jugar\\nPara finalizar el juego oprima (ctrl + .)\\nHasta la óprxima\\n')
248
249
250
251
252
253
254 if platform.system() == 'Darwin' or platform.system() == 'Linux' :
255     os.system('clear')
256 elif platform.system() == 'win32' :
257     os.system('cls')
258
259 print(Titulo)
260
261 r = input('Determine la probabilidad de que una célula inicie viva: ')
262 m = input('Determine el tamaño de la matriz cuadrada que desea usar: ')
263 if m > 35 :
264     print('Tenga en cuenta que para ese tamaño de matriz puede ser necesario \nreajustar la pantalla')
265     time.sleep(2)
266 n=0
267 a=str(' #')
268 b=str(' ')
269
270 #óCreacin de matriz
271 matriz = [[0 for x in range(m)] for y in range(m)]
272 i=0;
273 for i in range(0,m):
274     for j in range(0,m):

```

```

275         y=random.randint(0,100)
276         if (y>r):
277             matriz[i][j]=b
278         else:
279             matriz[i][j]=a
280
281 #óEvolucin de la matriz (Ciclo infinito)
282 while 1>0:
283
284     if platform.system() == 'Darwin' or platform.system() == 'Linux' :
285         os.system('clear')
286     elif platform.system() == 'win32' :
287         os.system('cls')
288
289     print (Titulo)
290
291     print('Ud óescogi una probabilidad inicial de vida del: ' + str(r));
292     print('Ud óescogi una matriz cuadrada de nétamaclulas: ' + str(m));
293
294     print (separator + '\n')
295     print('\n'.join([' '.join(['{:2}'.format(item) for item in row])
296         for row in matriz]))
297     print (separator + '\n')
298     print ('Ud lleva ' + str(n) + ' pasos')
299     print (footer)
300
301     n += 1
302     time.sleep(0.1)
303
304 #Cada una de las iteraciones del programa
305 for i in range(0,m):
306     j=0
307     for j in range(0,m):
308         v=0;
309         if i+1<m:
310             if matriz[i+1][j]==a:
311                 v=v+1
312             if j+1<m:
313                 if matriz[i+1][j+1]==a:
314                     v=v+1
315             if j-1>=0:
316                 if matriz[i+1][j-1]==a:
317                     v=v+1
318         if i-1>=0:
319             if matriz[i-1][j]==a:
320                 v=v+1
321             if j+1<m:
322                 if matriz[i-1][j+1]==a:
323                     v=v+1
324             if j-1>=0:
325                 if matriz[i-1][j-1]==a:
326                     v=v+1
327         if j+1<m:
328             if matriz[i][j+1]==a:
329                 v=v+1
330         if j-1>=0:
331             if matriz[i][j-1]==a:
332                 v=v+1
333
334 #Encontrar éclulas vivas
335 if matriz[i][j]==a:
336
337     #1Una celula viva con menos de dos vecinos vivos muere
338     if v<2:
339         matriz[i][j]=b
340     #2Una celula viva con 2 o 3 vecinos vivos permanece en su estado
341     if (v==2 or v==3):
342         matriz[i][j]=a
343     #3Una celula viva con mas de 3 vecinos muere
344     if v>3:
345         matriz[i][j]=b
346
347 #Encontrar éclulas muertos
348 if matriz[i][j]==b:
349
350     #4Una celula muerta con exactamente 3 vecinos vivos se convierte en viva
351     if v==3:
352         matriz[i][j]=a

```

3.2.1. Título y configuración

En esta parte del código, al igual que en matlab, se muestran los títulos del programa y se da una bienvenida al usuario para que empiece el juego, pidiéndole que inserte el tamaño de la matriz cuadrada y la probabilidad de que una celula se encuentre viva en un estado inicial. También se encuentra el contador de iteraciones y la definición de las variables a y b como se explico anteriormente. Se encuentra adicional a lo que vimos en matlab, una advertencia cuando los tamaños de la matriz son muy grandes puesto que puede necesitarse ajustar la pantalla. En dicho caso, hay un tiempo de pausa que permite que el usuario lea la advertencia. Para ello, a diferencia de matlab, fue necesario usar una función que depende del sistema operativo, por lo que es necesario identificarlo en esta parte del proceso.

```

355
356 # - *- coding: utf- 8 - *-
357 import os
358 import time
359 import random
360 import platform
361

```

```

362
363
364 Titulo = ('//////////=====\\\\\\\\\\n'
365           '//////////                                \\\\\\\\\\\n'
366           '//////////                                \\\\\\\\\\\n'
367           '//////////                                \\\\\\\\\\\n'
368           '//////////                                \\\\\\\\\\\n'
369           '//////////                                \\\\\\\\\\\n'
370           '//////////                                \\\\\\\\\\\n'
371           '//////////                                \\\\\\\\\\\n');
372
373 separator = ('\\n//////////=====\\\\\\\\\\\\')
374
375 footer = ('Gracias por Jugar\\nPara finalizar el juego oprima (ctrl + .)\\nHasta la óprxima\\n')
376
377 if platform.system() == 'Darwin' or platform.system() == 'Linux' :
378     os.system('clear')
379 elif platform.system() == 'win32' :
380     os.system('cls')
381
382 print(Titulo)
383
384 r = input('Determine la probabilidad de que una éclula inicie viva: ')
385 m = input('Determine el tama~no de la matriz cuadrada que desea usar: ')
386 if m > 35 :
387     print('Tenga en cuenta que para ese ñtamao de matriz puede ser necesario \\nreajustar la pantalla')
388     time.sleep(2)
389
390 n=0
391 a=str(' #')
392 b=str(' ')

```

3.2.2. Creación de la matriz inicial

En esta parte del código, al igual que en matlab, se crea la matriz y se determina su estado inicial.

```

391
392 #4 Creacin de matriz
393 matriz = [[0 for x in range(m)] for y in range(m)]
394 i=0;
395 for i in range(0,m):
396     for j in range(0,m):
397         y=random.randint(0,100)
398         if (y>r):
399             matriz[i][j]=b
400         else:
401             matriz[i][j]=a

```

3.2.3. Evolución general del programa

En esta parte del código, se crea un un ciclo infinito de la misma manera que en matlab, aparece el comando de 'Limpiar la pantalla' y se presenta la información del programa al usuario que incluye el título, la probabilidad y tamaño de la matriz, y el número de iteraciones que lleva el programa desde que inició. Finalmente, se encuentra el comando que permite pausar el programa antes de cada iteración.

```

403 #6Evolucin de la matriz (Ciclo infinito)
404 while 1>0:
405
406     if platform.system() == 'Darwin' or platform.system() == 'Linux' :
407         os.system('clear')
408     elif platform.system() == 'win32' :
409         os.system('cls')
410
411     print (Titulo)
412
413     print('Ud óescogi una probabilidad inicial de vida del: ' + str(r));
414     print('Ud óescogi una matriz cuadrada de ñtamao: ' + str(m));
415
416     print (separator + '\n')
417     print('\n'.join([''.join(['{:2}'.format(item) for item in row])
418                       for row in matriz]))
419     print (separator + '\n')
420     print ('Ud lleva ' + str(n) + ' pasos')
421     print (footer)
422
423     n += 1
424     time.sleep(0.1)

```

3.2.4. Cada una de las iteraciones del programa

En el código de Python, cada iteración del programa también está determinada por un ciclo que recorre cada una de las células de la matriz realizando los 2 procedimientos explicados anteriormente: Conteo de Vecinos, y determinación del estado de la célula. Las únicas diferencias en esta parte del código con matlab, son la sintaxis, y las diferencias en los límites de los ciclos debido a la diferencia en la numeración de las casillas de una matriz en cada programa. En matlab, las matrices empiezan desde la fila y columna 1, mientras en python su numeración empieza en 0.


```

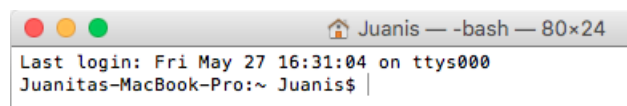
426 #Cada una de las iteraciones del programa
427 for i in range(0,m):
428     j=0
429     for j in range(0,m):
430         v=0;
431         if i+1<m:
432             if matriz[i+1][j]==a:
433                 v=v+1
434             if j+1<m:
435                 if matriz[i+1][j+1]==a:
436                     v=v+1
437             if j-1>=0:
438                 if matriz[i+1][j-1]==a:
439                     v=v+1
440         if i-1>=0:
441             if matriz[i-1][j]==a:
442                 v=v+1
443             if j+1<m:
444                 if matriz[i-1][j+1]==a:
445                     v=v+1
446             if j-1>=0:
447                 if matriz[i-1][j-1]==a:
448                     v=v+1
449         if j+1<m:
450             if matriz[i][j+1]==a:
451                 v=v+1
452         if j-1>=0:
453             if matriz[i][j-1]==a:
454                 v=v+1
455
456 #Encontrar células vivas
457 if matriz[i][j]==a:
458
459     #1Una célula viva con menos de dos vecinos vivos muere
460     if v<2:
461         matriz[i][j]=b
462     #2Una célula viva con 2 o 3 vecinos vivos permanece en su estado
463     if (v==2 or v==3):
464         matriz[i][j]=a
465     #3Una célula viva con mas de 3 vecinos muere
466     if v>3:
467         matriz[i][j]=b
468
469 #Encontrar células muertas
470 if matriz[i][j]==b:
471
472     #4Una célula muerta con exactamente 3 vecinos vivos se convierte en viva
473     if v==3:
474         matriz[i][j]=a

```

4. Ejemplo

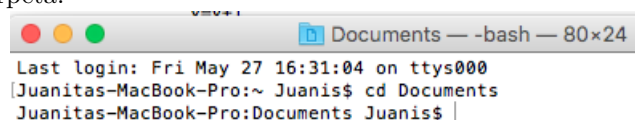
A continuación se mostrara un ejemplo del programa en python y se revisara lo que sucede en las primeras 5 iteraciones del código. En primer lugar, debe aclararse que para abrir el programa, es necesario hacerlo en el terminal del computador. Para lo cual, abrimos el terminal, accedemos a la carpeta en donde se encuentre el archivo y lo corremos desde ahí. Esto se muestra a continuación.

4.1. Abrir el terminal



4.2. Abrir la carpeta donde esta el archivo

Para esto es necesario escribir 'cd' que significa 'change directory', junto con el nombre de la carpeta.



4.3. Verificar que el archivo este

Para esto escribimos 'ls', lo cual nos muestra la lista de archivos que se encuentran en la carpeta a la que accedimos

```
Documents — -bash — 80x24
Last login: Fri May 27 17:48:16 on ttys000
Juanitas-MacBook-Pro:~ Juanis$ cd Documents
Juanitas-MacBook-Pro:Documents Juanis$ ls
Abroad          Notas de Voz    TheGameOfLife.py
Emocion Creativa Processing       ~$anford.docx
Interfaz        Python          ~$versheets.docx
Javeriana       Scans
Musica          Stuff
Juanitas-MacBook-Pro:Documents Juanis$
```

4.4. Poner a correr el archivo

Para esto es necesario escribir el nombre del programa 'Python' junto con el nombre del archivo separado por un espacio, es decir 'TheGameOfLife.py'.

```
Documents — -bash — 80x24
Last login: Fri May 27 17:48:16 on ttys000
Juanitas-MacBook-Pro:~ Juanis$ cd Documents
Juanitas-MacBook-Pro:Documents Juanis$ ls
Abroad          Notas de Voz    TheGameOfLife.py
Emocion Creativa Processing       ~$anford.docx
Interfaz        Python          ~$versheets.docx
Javeriana       Scans
Musica          Stuff
Juanitas-MacBook-Pro:Documents Juanis$ Python TheGameOfLife.py
```

4.4.1. Solicitud de información al usuario

La primera ventana al correr el programa, muestra los títulos del mismo junto con las preguntas de los datos al usuario. En este caso se colocó probabilidad inicial de vida del 30 y tamaño de la matriz de 10X10.

```
Documents — Python TheGameOfLife.py — 80x24
/////////=====/////////
/////////                          WELCOME TO          ///////////
/////////                          ///////////
/////////                          THE GAME OF LIFE      ///////////
/////////=====/////////
/////////                          Presentado por: Juanita Gomez  ///////////
/////////=====/////////

Determine la probabilidad de que una célula inicie viva: 30
Determine el tamaño de la matriz cuadrada que desea usar: 10
```

4.5. Primera matriz

En la pantalla aparece una matriz de 10X10, que tiene 30 células vivas ya que este valor corresponde al 30 por ciento del total de las células. Aparece también el contador y las instrucciones para finalizar el programa.

```

//////////=====//////////
//////////          WELCOME TO          //////////
//////////          THE GAME OF LIFE    //////////
//////////=====Presentado por: Juanita Gomez=====//////////
//////////=====//////////

Ud escogio una probabilidad inicial de vida del: 30
Ud escogio una matriz cuadrada de tamano: 10

//////////=====//////////

  ■ ■ ■
  ■ ■
    ■ ■ ■ ■ ■
  ■ ■ ■ ■ ■ ■ ■
    ■ ■ ■ ■
  ■ ■ ■ ■ ■
    ■ ■ ■ ■ ■
      ■ ■ ■ ■
        ■ ■ ■
          ■ ■
            ■

//////////=====//////////

Ud lleva 0 pasos
Gracias por Jugar
Para finalizar el juego oprima (ctrl + .)
Hasta la proxima

```

4.6. Primera iteración

En la primera iteración observamos que se muestra la información del programa al usuario, junto con los títulos y el número de iteraciones que cambió de 0 a 1. La nueva matriz ahora tiene 28 células vivas después de seguir las reglas de Conway.

```

//////////=====//////////
//////////          WELCOME TO          //////////
//////////          THE GAME OF LIFE    //////////
//////////=====Presentado por: Juanita Gomez=====//////////
//////////=====//////////

Ud escogio una probabilidad inicial de vida del: 30
Ud escogio una matriz cuadrada de tamano: 10

//////////=====//////////

  ■
  ■ ■ ■
    ■ ■ ■
      ■ ■ ■ ■
  ■ ■ ■ ■ ■ ■ ■
    ■ ■ ■ ■ ■ ■
      ■ ■ ■ ■ ■
        ■ ■ ■ ■ ■
          ■ ■ ■ ■
            ■ ■ ■
              ■ ■
                ■

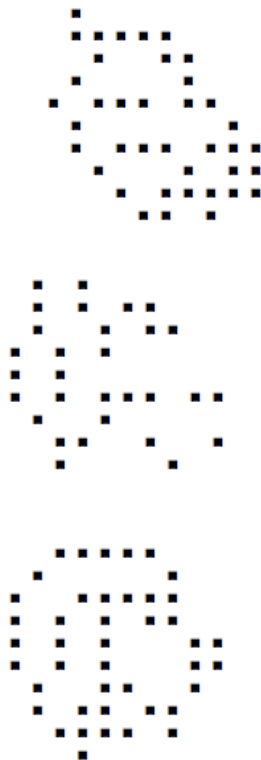
//////////=====//////////

Ud lleva 1 pasos
Gracias por Jugar
Para finalizar el juego oprima (ctrl + .)
Hasta la proxima

```

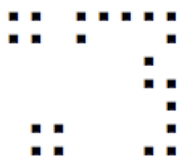
4.7. Evolución del programa

A continuación se presentan las siguientes 3 iteraciones del programa en donde podemos visualizar las transformaciones de la matriz.



4.8. Finalización del programa

En este caso, después de 197 iteraciones, la matriz empieza a ser constante por lo que a partir de ahí todas las matrices siguientes son iguales. Se da una estabilización de la población y dado que el programa tiene una duración infinita, es el usuario quien ahora decide cuando detiene el programa utilizando el comando ctrl+.



Referencias

- [1] Melissa Gymrek, *Conway's Game of Life*, <http://web.mit.edu/sp.268/www/2010/lifeSlides.pdf>, (2010)
- [2] David Alejandro Reyes Gómez, *Descripción y Aplicaciones de los Autómatas Celulares*, http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/Summer_Research_files/Arti_Ver_Inv_2011_DARG.pdf, (2011)
- [3] Manuel Romero Dopico, *EL JUEGO DE LA VIDA*, <http://www.it.uc3m.es/jvillena/irc/practicas/09-10/04mem.pdf>, (s.f)
- [4] Dierk Schleicher, *Interview with John Horton Conway*, <http://www.ams.org/notices/201305/rnoti-p567.pdf>, (2013)
- [5] Martin Gardner, *MATHEMATICAL GAMES*, Scientific American 223, (1970)