Conway y el juego de la vida - Trabajo opcional TAC

Grupo 83
Ignacio Talavera Cepeda - 100383487

Índice

Introducción

El **Juego de la Vida**, *Game of Life* o simplemente *Life*, es un autómata celular ideado por el matemático John Horton Conway (1937 - 2020), que vió por primera vez la luz en 1970 en la revista *Scientific American*, en una sección de Martin Gardner dedicada a juegos matemáticos [1]. Conway concibió *Life* como un juego, un sistema matemático recreativo para entusiastas como él; pero su impacto en el mundo de la informática fue muy notable. Atrajo interés hacia el campo de los autómatas celulares y ganó popularidad entre investigadores de varias disciplinas, como la informática, las matemáticas, la física, la biología, la economía, la filosofía.

En este trabajo cubriremos la historia de John Horton Conway, la historia de *Life,* su funcionamiento, patrones y estructuras relevantes dentro del juego (incluyendo aquellas auto-replicantes), su no decibilidad, su condición de sistema Turing Completo, diferentes variaciones del juego, y la comunidad que se ha construido alrededor de él desde 1970 hasta la actualidad.

John Horton Conway y Life

Conway nace el 26 de diciembre de 1937 en Liverpool, y se convierte en un aficionado de las matemáticas desde muy temprana edad. Estudia la carrera de Matemáticas en Cambridge. En 1959 termina su grado y, bajo la tutela de Harold Davenport, comienza su investigación en teoría de números [2]. Uno de sus primeros hitos conseguidos consistió en resolver el problema planteado por Davenport sobre la escritura de números como sumas de sus quintas potencias, derivación del problema de Waring [3]. Obtiene su doctorado en 1964 y es nombrado profesor, aunque dejaría Cambridge en 1986 para mudarse a los Estados Unidos y asumir el cargo de presidente de la cátedra de matemáticas John von Neumann en Princeton [4].

John Conway permanecería como director de la cátedra hasta su fallecimiento, en 2020, debido a complicaciones por COVID-19 [5]. Durante su larga carrera, además de *Life*, dejó otras contribuciones a diferentes campos de las matemáticas, entre los que destacan teoría de juegos [6], [7], geometría [8], [9] [10], etc. Sus dos trabajos más conocidos, después de *Life*, son la constante de Conway, una constante matemática ligada a la sucesión de desintegración audioactiva (o secuencia *Look-and-say*) [11]; y la regla *Doomsday* o regla del fin del mundo, un método para calcular el día de la semana en el que cae una fecha determinada inspirado en el algoritmo del calendario perpetuo de Lewis Carroll [12].

Motivado por el nuevo campo de los autómatas celulares, Conway comenzó a experimentar con composiciones bidimensionales de estos autómatas y reglas para regirlos, con el objetivo de crear un sistema impredecible a partir de reglas sencillas. Según una lista expresada verbalmente del propio Conway a sus compañeros en 1999, quería que las reglas cumpliesen los siguientes criterios:

1. No debe haber explosión exponencial en el crecimiento.

2. Deben existir patrones iniciales pequeños que tengan desenlaces caóticos e imprevisibles.
3. Debe haber potencial para crear constructores universales de von Neumann dentro del entorno.
4. Las reglas deben ser lo más simples posibles.

La primera aparición de *Life*, como ya se ha mencionado antes, fue en la revista *Scientific American*, en una sección de Martin Gardner dedicada a juegos matemáticos, en 1970 [1]. El juego obtuvo una enorme popularidad desde su salida, y su influencia fue aumentando hasta sobrepasar los círculos académicos y entrar en la cultura popular de finales del siglo XX. *Life* es uno de los primeros juegos de simulación. Además, su lanzamiento coincidió con la creciente democratización de los ordenadores personales, cuyo acceso era cada vez más fácil. Gracias a la facilidad de sus reglas, resultaba sencillo para un aficionado programar una versión de *Life* con el hardware y software disponible, establecer una disposición y dejar el juego correr durante horas para ver los resultados.

Reglas de Life

El sistema creado en *Life* es un universo bidimensional de células ortogonales. Cada celda tiene dos estados: viva o muerta. Cada una de las celdas interacciona con sus ocho celdas vecinas, correspondientes a las ocho células que la rodean. En cada instante de tiempo t se aplican las siguientes reglas [1]:

1. Cualquier célula viva con menos de dos células vecinas vivas muere, por soledad.
2. Cualquier célula viva con dos o tres células vecinas se mantiene viva.
3. Cualquier célula viva con más de tres células vecinas muere, por sobrepoblación.
4. Cualquier célula muerta exactamente tres células vecinas vivas vive, por nacimiento.

Se establece una disposición inicial, que podría ser vista como una *semilla*, y se comienzan a simular generaciones infinitas en las que, en cada instante, se ejecutan las reglas sobre todas las células

Patrones y Estructuras Relevantes

Una de las primeras clasificaciones de estructuras dentro de *Life* que se realizó es en base al comportamiento [13]:

- *Still lifes* o Unidades estáticas: estructuras que no cambian entre generaciones.
- *Oscillators* u Osciladores: estructuras que, tras varias generaciones, vuelven a su estado inicial
- *Spaceships* o Naves: estructuras capaces de moverse por el mapa conservando la mayor parte de su integridad.

Los primeros integrantes de cada uno de estos grupos fueron descubiertos antes de la adaptación de *Life* a los ordenadores. Conway y sus compañeros solían utilizar pizarras y tableros de Go, un juego de estrategia chino con una cuadrícula de 19x19.

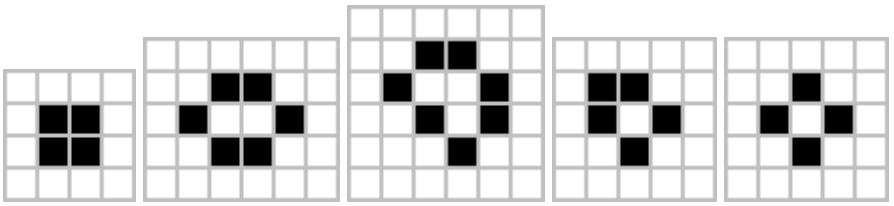


Figura 1. Unidades estáticas comunes. De izquierda a derecha: *block*, *bee-hive, loaf, boat, tube.*

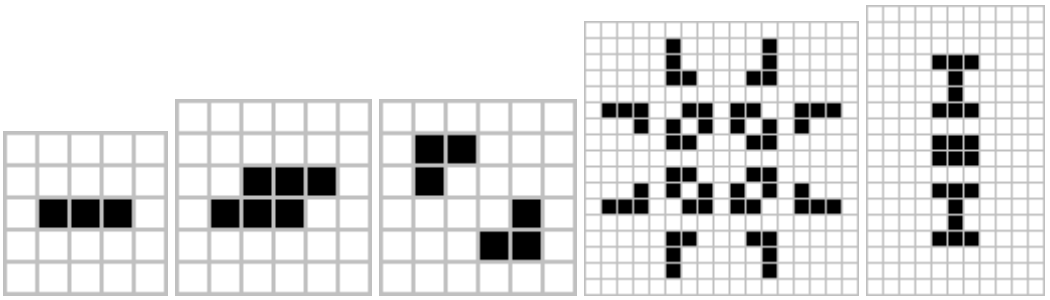


Figura 2. Osciladores comunes. De izquierda a derecha: *blinker, toad, beacon* (periodo 2)*, pulsar* (periodo 3)*, penta-decathlon* (periodo 15).

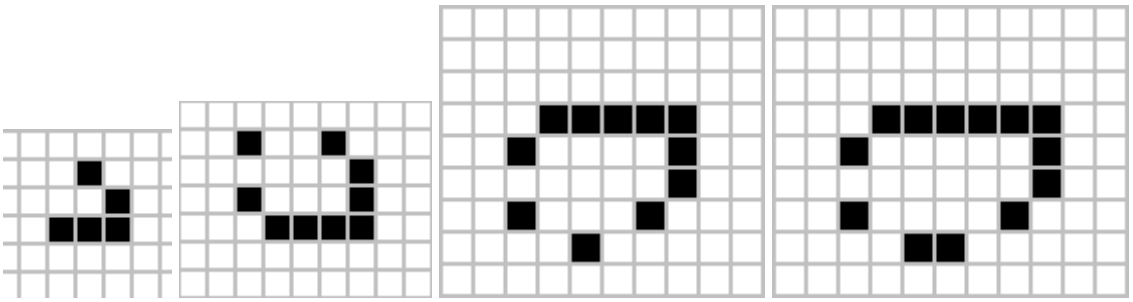


Figura 3. Naves comunes. De izquierda a derecha: *glider, LWSS, MWSS, HWSS*.

Pese a que los osciladores de período 2 son los más comunes, hay osciladores conocidos de casi todos los periodos [14]. Algunos han teorizado que los osciladores y las unidades estáticas comprenden un mismo grupo, puesto que los segundos se repiten también indefinidamente en cada generación [15]

Del estudio de los osciladores se llegaron a estructuras asociadas a estos, y entre ellas destacan [16]:

- ***Methuselahs***: estructuras que, tras variar, se estabilizan en cierta generación, y se vuelven estáticas. Uno de los más comunes es el R-Pentomino.
- ***Diehards***: patrones similares a los *Methuselahs* pero que, en lugar de estabilizarse, se desvanecen tras cierto número de generaciones. Se ha conjeturado que, para un *diehard* de siete o menos células, el máximo de generaciones que se puede mantener vivo es 130. A partir de ahí, no hay límite para *diehards* con más células.
- **Propagadores**: patrones autoreplicantes. Hay de varios tipos, los más comunes son lineales, cuyas estructuras hijas bloquean el nacimiento de nuevos individuos y hacen que el sistema pierda su condición. Los propagadores lineales supusieron el comienzo de la investigación sobre los propagadores cuadráticos, estructuras capaces de ignorar esa limitación, pero pese a que se han descubierto propagadores cuadráticos en sistemas similares a *Life*, no se ha encontrado una estructura así en el juego de *Conway.*
- **Reflectores**: patrones estables u oscilantes que pueden reflejar naves sin perder su integridad (aquellos que son capaces de reflejar una vez se llaman *one-time turners*, y se desestabilizan tras esto).
- ***Guns*** o **armas**: patrones estables u oscilantes capaces de crear y lanzar *gliders* o naves más complejas

Conjetura de Conway

Con la publicación de *Life*, su autor teorizó que no se podía crear una estructura capaz de crecer indefinidamente, y que debía existir un límite superior tras el cual las estructuras colapsarían. *Conway* ofreció un premio de cincuenta dólares a la primera persona que pudiese probar la veracidad o falsedad de esta conjetura antes de 1970 [1]. Este premio fue obtenido en noviembre de ese mismo año por un equipo del MIT liderado por Bill Gosper, y la estructura fue bautizada como *Gosper glider gun*. A partir de la decimoquinta generación, el *Gosper glider gun* crea su primer *glider*, y continúa creando *gliders* indefinidamente cada quince generaciones [16].

Este *glider* fue el más pequeño conocido hasta 2015, con el descubrimiento del *Skimin glider gun*, capaz de crear y lanzar un *glider* cada 120 generaciones [17].

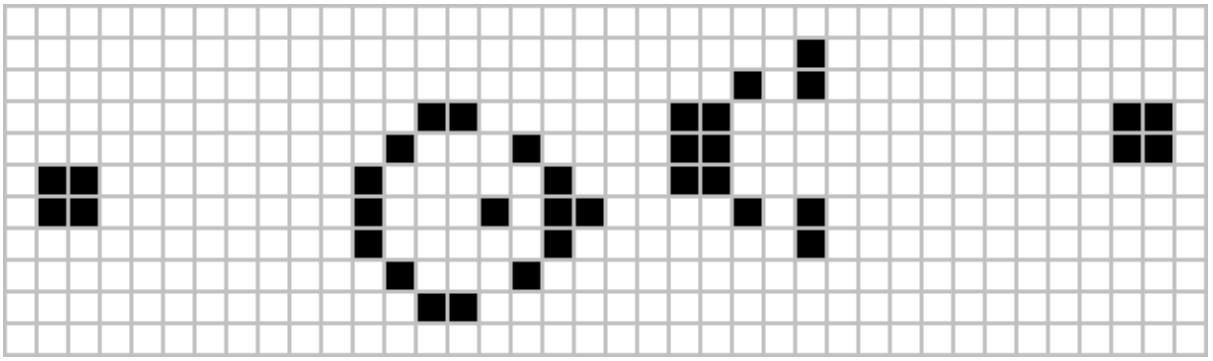


Figura 4. *Gosper Glider Gun.*

La investigación sobre las estructuras capaces de crecer indefinidamente continuó, y se buscó su mínima expresión. En 1997 se descubrió el patrón de diez células con esta cualidad, y se demostró que este era el límite inferior [16].

Figura 5. Patrón mínimo capaz de crecer indefinidamente.

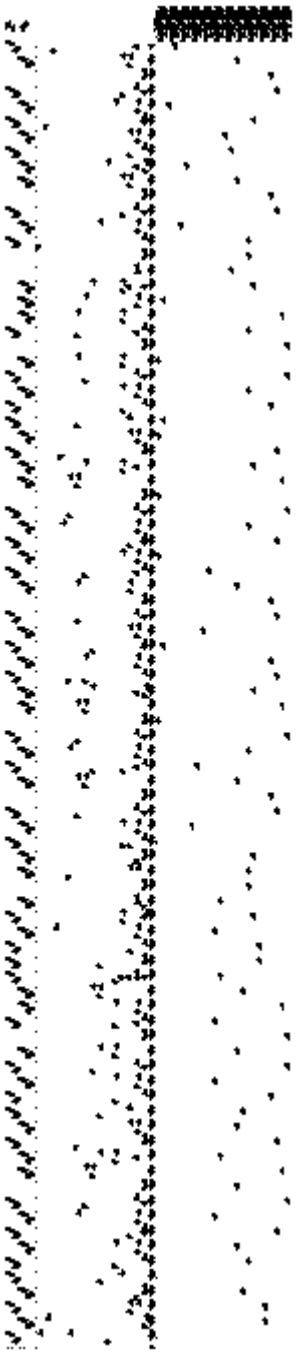
Puertas lógicas

Aunque los ejemplos vistos previamente podrían verse cómo puramente lúdicos, la investigación acerca de los patrones autoreplicantes condujo a uno de los descubrimientos más importantes sobre *Life*.

Dos *gliders* pueden interactuar con otros objetos creando ciertos comportamientos. Si dos *gliders* son disparados hacia un bloque con una trayectoria y disposición específica, este bloque se moverá hacia la fuente de los *gliders*. Algo similar también ocurre con tres *gliders*, pero produciendo el efecto contrario: su alejamiento. Esta propiedad podía ser usada para simular un contador, y a partir de este contador se pudieron construir puertas lógicas *AND*, *OR* y *NOT* [18], [19].

Esto provocaría un descubrimiento de estructuras derivadas de una forma similar a la descubierta en la vida real, pero mucho más rápida.

Turing Completo y Máquina de Turing Universal



Con el desarrollo de las primeras puertas lógicas y su posterior combinación, se alcanzaron estructuras que funcionaban como autómatas finitos. La comunidad científica alrededor de *Life* se centró, entonces, en crear una Máquina de Turing Universal dentro del sistema.

Paul Rendell creó en 2001 una máquina de Turing en *Life*, pero con una cinta finita. Apoyados en su trabajo y en las estructuras *Sliding Block Memory* de Dean Hickerson, Paul Chapman consiguió construir *Minsky Register Machine* (MRM), una modelo computacional equivalente a una máquina de Turing universal en 2002. La primera versión de esta máquina tenía una gran tamaño (268,096 células vivas en un area de 4,558 x 21,469), pero considerablemente lenta, debido al hardware de la época y al estado de los simuladores de *Life*. Sin embargo, había quedado experimental probado la suposición realizada cuando se consiguieron realizar las primeras puertas lógicas: *Life* es un entorno Turing-completo [18].

El funcionamiento de la máquina de Turing de Chapman, y de todos los diseños derivados, se basa en patrones de 30x30. Se usan *LWSS* para comunicar los diferentes componentes, que tardan 60 generaciones en cruzar ese patrón, por lo que cada 60 generaciones cualquier pulso (*LWSS* en movimiento) está en la misma posición relativa para el patrón. Los componentes permanecen estáticos o rotando armónicamente hasta que un pulso les alcanza, y este dispara su comportamiento, lo que más tarde desemboca en la producción de zero, uno o dos pulsos emitidos y la posibilidad de un cambio en el sistema interno del componente. Tras el tiempo de procesamiento, el componente vuelve a un estado estático o armónico. Solo un pulso entrante puede ser procesado correctamente por cada componente, pero mucho pulso puede coexistir en entornos con varios componentes. Una característica de esta máquina de Turing es que los pulsos no están obligados a llegar de forma síncrona a los componentes; se asegura una disposición que dé el suficiente espacio para que no haya colisiones entre dos pulsos y un componente. Los componentes que se utilizaron en esta primera versión de la MRM son: *Register*, *Latch, Split Left*, *Split Right*, *Merge Left*, *Merge Right*, and *Eater*.

Figura 6: MRM a escala 1/32

No decidibilidad de *Life*

Al ser *Life* un sistema Turing completo, y debido al problema de la parada, podemos demostrar, por extensión, que *Life* es un sistema no decidible [20], utilizando las mismas aproximaciones que se usan para demostrar que un problema es no decidible a través de máquinas de Turing. Que un problema o entorno sea no decidible implica que no hay ningún algoritmo capaz de indicar el *output* de cierto *input* a priori, la única forma es ejecutar el juego. La única forma de acelerar la obtención de esa respuesta sería aumentar la velocidad de computación del propio juego.

Otra aproximación muy relevante hacia el problema de la no decidibilidad de *Life* pasa, no por reducir el problema a una máquina de Turing, sino mediante su tratamiento como autómata celular. A través del teorema de Rice [26] se demuestra que todas las propiedades de los conjuntos de autómatas celulares de dimensiones *d* son indecidibles para todo $d \geq 1$. Algunos de los métodos más usados para probar este teorema se basan en la reducción. En concreto, la demostración llevada a cabo por Kari pasa por la reducción del problema a su nilpotente, y el análisis de las propiedades del problema nilpotente para los conjuntos no limitados. Es importante destacar que, en un conjunto limitado, no se puede demostrar que todas las propiedades de los autómatas celulares sean indecidibles [26].

Se ha hecho una clasificación de los autómatas celulares en cuatro tipos, incluyendo los tres primeros aquellos que convergen a estados estables en todas sus transiciones. *Life* pertenece al cuarto tipo, uno pensado para autómatas celulares que emulan Máquinas de Turing. En esta categoría, todas las propiedades de los autómatas celulares son no decidibles y, por tanto, el autómata en sí es no decidible [27].

Sin embargo, la aproximación más común, la misma que siguió el propio Conway en su aportación a *Winning Ways for your Mathematical Plays* pasa por demostrar que *Life* es un sistema Turing completo y que, por tanto, y a través del problema de la parada, pueden ejecutarse en el programas cuya salida sea no decidible [6].

Variaciones de Life

A partir de *Life* se han creado una gran variedad de autómatas celulares que simulan entornos parecidos a *Life*, entornos parecidos a ecosistemas. Un autómata celular se considera similar a *Life*, o *Life-like*, sí y solo si cumplen las siguientes condiciones:

- El entorno es bidimensional y dividido en celdas.
- Cada celda tiene solo dos estados, referidos como “vivo” y “muerto”.
- El vecindario de cada celda es un vecindario de Moore (ocho celdas adyacentes).

- En cada instante de tiempo el nuevo estado puede ser expresado como una función de los estados de las celdas en el vecindario de Moore y el estado de la propia celda.

Se creó una nomenclatura para distinguir a los *Life-like* que expresa de forma concisa las reglas del mismo. Por ejemplo, el propio *Life* sería B3/S23, puesto que una celda nace (**B**orn) si tiene tres celdas vivas o más a su alrededor, sobrevive (**S**urvive) si tiene dos o tres celdas vivas a su alrededor, y muere en el resto de situaciones. Hay 2^{18} posibles juegos *Life-like* [21]. En la siguiente tabla se recogen algunos de los más populares:

Reglas	**Nombre**	**Descripción**
B1357/S1357	Replicator	Todos los patrones son reemplazados por múltiples copias de sí mismos.
B2/S	Seeds	Todos los patrones reviven después de morir.
R	Life without Death	Celdas vivas nunca mueren.
B35678/S5678	Diamoeba	Se forman grandes patrones con forma de diamante.
B36/S125	2x2	Si un patrón es 2x2, seguirá evolucionando, agrupando esos bloques en estructuras similares a las potencias de dos, con comportamientos más lentos.
B36/S23	HighLife	Similar a *Life*
B3678/S34678	Day & Night	Simetría dentro de las reversiones vivo-muerto. Patrones de alta complejidad.
B368/S245	Morley	En honor a Stephen Morley. Periodos muy altos y naves lentas.
B4678/S35678	Anneal	Simetría dentro de las reversiones vivo-muerto

Figura 7: Diferentes autómatas celulares basados en *Life* [13]

Comunidad alrededor de Life

Tras el lanzamiento de *Life* como entorno teórico, la joven comunidad de ciencias de la computación y matemáticas comenzó a interesarse en simular el juego en programas informáticos. El primer programa de *Life* fue escrito en una versión temprana de ALGOL 68C, y estos resultados fueron publicados en octubre de 1970 en *Scientific American* [22].

Desde entonces, diferentes versiones de *Life* han sido diseñadas, siguiendo el flujo marcado por el desarrollo de la tecnología. Actualmente hay una gran variedad de versiones de *Life*, la mayoría *open-source* o gratuitas, que permiten que personas de todo el mundo, sin importar la capacidad de sus equipos informáticos, puedan jugar. El más popular es *Golly*, multiplataforma (Linux, Windows, MacOS, iOS y Android), *open-source* y con simulaciones no solo de *Life*, sino también de otros autómatas celulares (como algunos de los mencionados anteriormente). Además, ofrece herramientas de scripting en Lua y Python para personalizar las simulaciones [23].

Life puede correr en navegadores, como acreditan multitud de versiones [24], [25]. Google incluyó un *easter egg* en 2012, permitiendo a los usuarios jugar una versión reducida de *Life* directamente desde la pantalla de resultados del buscador.

También es muy reseñable la comunidad *ConwayLife*, basado en la arquitectura wiki, que sirve como punto de encuentro, tablón de anuncios y fuente de información fiable y actualizada de todo lo relacionado con *Life* y con otros autómatas celulares, sobre todo en lo que respecta a la información más avanzada del juego. Desde *ConwayLife* se desarrolló el proyecto *Golly.*

Conclusiones

El juego de la vida, desde su nacimiento en 1970, que coincidió con el comienzo de la popularización de los ordenadores y, ha pasado de ser un interesante pasatiempo teórico a un sistema complejo, precursor de su propio campo de estudio y con el potencial de atraer a una gran cantidad de entusiastas que han conseguido que, incluso en el siglo XXI, siga siendo relevante. Sus implicaciones van mucho más allá de un juego matemático o una curiosidad teórica: se han estudiado sus implicaciones a nivel matemático, filosófico, biológico, y un largo etcétera. En este trabajo se han intentado cubrir sus implicaciones en el campo de la computación. Pese a que *Life* fue concebido con papel y bolígrafo, y puede que porque llegó en el momento adecuado, ha estado desde sus inicios relacionado con el mundo de los ordenadores, y posteriores descubrimientos le han colocado más cerca de una máquina de Turing que de un table de *go*.

Life es un sistema con reglas muy sencillas, pero con una explosión exponencial en cuanto a profundidad sin igual. Nada impide a cualquier persona entender sus tres reglas y jugar varias partidas, sin más afán que el de probar el propio entorno, pero la comunidad ha demostrado que se pueden construir patrones y estructuras de una enorme complejidad, capaces de resolver tareas o responder al estado del entorno en el que se encuentran. El propio campo de los autómatas celulares se ha retroalimentado de *Life*, siendo este uno de sus precursores y primeros modelos en los que realizar simulaciones, y ayudando al establecimiento de entornos similares que pueden ser catalogados y clasificados (los *Life-like*).

Este trabajo pretende ser un punto de partida, en el que se traten sus reglas, se aborde su relación con la asignatura de Teoría Avanzada de la Computación y se expliquen los avances que la comunidad ha hecho sobre el juego. Se incluye, además de referencias a otros simuladores ya establecidos (en el anterior apartado), un pequeño simulador de *Life* en formato de Jupyter Notebook y Google Colab. En el siguiente anexo se cubre su funcionamiento.

Escribiendo e implementando *Life*, en lo personal, he ahondado mucho en este sistema, y he aprendido multitud de detalles sobre el juego y sobre las ciencias de la computación. Pese a que la curva de aprendizaje en los materiales y la bibliografía es un poco abrupta, pasando de conceptos básicos a muy avanzados de una forma rápida, todas las lecturas realizadas para la documentación del trabajo han resultado ser de un gran valor, tanto teórico como divulgativo.

Anexo

Jupyter Notebook desde Github

Se ha creado un repositorio de Github (https://github.com/ignacioct/GameOfLife) con los diferentes elementos de este trabajo (presentación, memoria, bibliografía ...) y el código en un Jupyter Notebook. Para correr este notebook en local, se debe disponer de un entorno con Python 3 y Jupyter Notebook instalados. El resto de paquetes requeridos se instalan dentro del propio notebook, para agilizar este proceso.

Para correr el simulador, se debe ejecutar el comando Jupyter Notebook en una terminal sobre el directorio raíz. Esto lanzará la UI de Jupyter en una pestaña de navegador a través de host local, en la que podremos establecer un kernel de Python 3 y correr el notebook.

Jupyter Notebook desde Google Colab

Google Colab ofrece la ventaja de correr el simulador sin realizar descargas ni establecer un entorno. Sin embargo, la visualización que ofrece el paquete utilizado para mostrar por pantalla *Life* es menos legible: esto se debe a la forma en la que Colab procesa los outputs de terminal por pantalla. Por esto, se recomienda el camino anterior. Sin embargo, se ha preparado un notebook en Google Colab, que puede ser ejecutado directamente: https://drive.google.com/file/d/1Mj2mu4rSo3cDZ2roYTrZw2n2dINDqKFb/view?usp=sharing

Referencias

[1] M. Gardner, «Mathematical Games - The fantastic combinations of John Conway's new solitaire game "life" - M. Gardner - 1970», *Sci. Am.*, vol. 223, pp. 120-123, 1970.

[2] M. J. Bradley, *Mathematics Frontiers*. Infobase Publishing, 2006.

[3] D. Hilbert, «Beweis für die Darstellbarkeit der ganzen Zahlen durch eine feste Anzahl n-ter Potenzen (Waringsches Problem)», en *Gesammelte Abhandlungen: Erster Band Zahlentheorie*, D. Hilbert, Ed. Berlin, Heidelberg: Springer, 1932, pp. 510-527.

[4] S. Chang, «British School», en *Academic Genealogy of Mathematicians*, World Scientific, 2011, p. 205.

[5] «COVID-19 Kills Renowned Princeton Mathematician, "Game Of Life" Inventor John Conway In 3 Days», *Mercer Daily Voice*, abr. 12, 2020. https://dailyvoice.com/new-jersey/mercer/obituaries/covid-19-kills-renowned-princeton-mathematician-game-of-life-inventor-john-conway-in-3-days/786461/ (accedido abr. 03, 2021).

[6] E. R. Berlekamp, J. H. Conway, y R. K. Guy, *Winning Ways for Your Mathematical Plays: Volume 1*. Natick, Mass, 2001.

[7] J. H. Conway, *On Numbers and Games*. Natick, Mass, 2000.

[8] John H. Conway, «Four-dimensional Archimedean polytopes», Copenhagen, 1965, pp. 38-39.

[9] G. C. Rhoads, «Planar tilings by polyominoes, polyhexes, and polyiamonds», *J. Comput. Appl. Math.*, vol. 174, pp. 329-353, feb. 2005.

[10] E. W. Weisstein, «Conway Polynomial». https://mathworld.wolfram.com/ConwayPolynomial.html (accedido abr. 03, 2021).

[11] John H. Conway, «The Weird and Wonderful Chemistry of Audioactive Decay», *Eureka*, vol. 46, pp. 5-16, ene. 1986.

[12] John H. Conway, «Tomorrow is the Day After Doomsday», *Eureka*, pp. 28-32, oct. 1973.

[13] «Conway's Game of Life», *Wikipedia*. mar. 25, 2021, Accedido: abr. 08, 2021. [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=Conway%27s_Game_of_Life&oldid=1014218578.

[14] «Game of Life Status page». http://entropymine.com/jason/life/status.html#oscper (accedido abr. 08, 2021).

[15] «El Juego de la Vida - YouTube». https://www.youtube.com/ (accedido abr. 08, 2021).

[16] «Life Lexicon». https://www.conwaylife.com/ref/lexicon/ (accedido abr. 08, 2021).

[17] «The Hunting of the New Herschel Conduits - Page 9 - ConwayLife.com». https://conwaylife.com/forums/viewtopic.php?f=2&t=1599&start=200#p19125 (accedido abr. 08, 2021).

[18] «Igblan - Life Universal Computer». http://www.igblan.free-online.co.uk/igblan/ca/ (accedido abr. 08, 2021).

[19] «Digital Logic Gates on Conway's Game of Life - Part 1». https://nicholas.carlini.com/writing/2020/digital-logic-game-of-life.html (accedido abr. 08, 2021).

[20] R. Wainwright, «Life is universal!», 1974, vol. 2, pp. 449-459.

[21] T. Ceccherini-Silberstein y M. Coornaert, *Cellular Automata and Groups*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[22] Gardner y Martin, «Mathematical Games: The fantastic combinations of John Conway's

new solitaire game “Life”".», *Scientific American*, vol. 223, n.º 4, pp. 120-123, oct. 1970.
[23]«Golly Game of Life Home Page». http://golly.sourceforge.net/ (accedido abr. 21, 2021).
[24]«Game of Life - Customized Conway’s GoL & Variants - Online Simulator». https://www.dcode.fr/game-of-life(accedido abr. 21, 2021).
[25]«Play John Conway’s Game of Life». https://playgameoflife.com/ (accedido abr. 21, 2021).
[26]J. Kari, “Rice’s theorem for the limit sets of cellular automata,” 1994, doi: 10.1016/0304-3975(94)90041-8.
[27] K. Culik II and S. Yu, “Yu, S.: Undecidability of CA Classification Schemes. Complex Systems 2, 177-190,” *Complex Systems*, vol. 2, Jan. 1988.