SIMULACIONES COMPUTACIONALES: AUTÓMATAS CELULARES

Marta G. Caligaris y Georgina B. Rodríguez

Grupo Ingeniería & Educación
Facultad Regional San Nicolás – Universidad Tecnológica Nacional
Colón 332 (2900) San Nicolás, Argentina
gie@frsn.utn.edu.ar

RESUMEN

El modelado ha consistido tradicionalmente en plantear y resolver ecuaciones que describen el fenómeno en estudio. Pero en la realidad, se presentan sistemas que no pueden ser representados en términos de ecuaciones; las simulaciones computacionales son la herramienta adecuada para el estudio de este tipo de sistemas. En estas simulaciones, intervienen directamente algoritmos, plasmados en programas computacionales.
Los autómatas celulares son una idealización de sistemas físicos, en los cuales las variables analizadas son consideradas magnitudes discretas. El concepto inicial fue planteado alrededor de 1940, logrando con el tiempo aplicaciones diversas. Por ejemplo, en la teoría de computación los autómatas celulares promovieron el impulso del procesamiento paralelo y el procesamiento de imágenes; en biología se han utilizado para modelar problemas de genética; en química, para estudiar problemas de reacciones de difusión, y en física, para estudiar problemas de dinámica de fluidos.
En este trabajo se presentan algunos ejemplos de autómatas celulares en una, dos y tres dimensiones, en aplicaciones diseñadas a medida en el software libre Scilab, en las que los alumnos pueden interactuar ya sea para visualizar los diversos modelos, o para modificar la programación de manera de lograr otros ejemplos diferentes.

Palabras clave: simulaciones, autómatas celulares, juego de la vida, sistemas dinámicos discretos, Scilab

1. INTRODUCCIÓN

El término "computación natural" se refiere al estudio de sistemas computacionales que se inspiran y usan ideas a partir de sistemas de la naturaleza, incluyendo sistemas biológicos, ecológicos y físicos. Estos sistemas se pueden organizar en tres tipos: algoritmos computacionales para resolución de problemas inspirados en fenómenos naturales, sistemas computacionales para la simulación y emulación de la naturaleza y dispositivos o paradigmas nuevos que utilicen medios que no sean silicio para almacenar y procesar información [1]. Algunas de las áreas que abarca la computación natural son, vida artificial, sistemas auto-organizados, computación neuronal, computación evolutiva, robótica, sistemas biológicos, entre otras.
Una de las características más destacadas de la computación natural es la conjunción de una gran variedad de disciplinas y campos de investigación. Ideas, principios, conceptos y modelos teóricos de la biología, física y química son requeridos para un buen entendimiento de la naturaleza, y consecuentemente un buen desarrollo de la computación natural.
Los autómatas celulares constituyen uno de los modelos más antiguos de la computación natural. John Von Neumann fue quien, inspirado en la biología, alrededor de 1940, intentó diseñar sistemas artificiales auto-replicables, que sirvieran para otros propósitos. La idea de Von Neumann era investigar dispositivos computacionales análogos al cerebro humano, en los que la memoria y las unidades de procesamiento no estuvieran separadas entre sí, sino que trabajaran en paralelo y fueran capaces de repararse y reconstruirse con el material necesario. Así, comenzó a pensar en un universo discreto que consistía en una malla bidimensional de máquinas de estado finito, llamadas celdas, interconectadas localmente. Las celdas modificaban su estado sincrónicamente, dependiendo de algunas celdas cercanas, a partir de ciertas reglas.
Los autómatas celulares comparten varias propiedades del mundo físico, permitiendo modelar con ellos distintos fenómenos físicos y biológicos. Por ejemplo, la simulación discreta de flujo de fluidos usando autómatas celulares se ha vuelto un campo en sí mismo [2]. Otras simulaciones clásicas que utilizan autómatas celulares para sistemas físicos son los modelos "Ising spin" [3] y fenómenos de difusión [4, 5].

El modelado de crecimiento de celdas o la cristalografía, son algunas aplicaciones frecuentes. Los autómatas celulares también se utilizan en criptografía.
En este trabajo se mostrarán aplicaciones desarrolladas en el software libre SCILAB [6], para presentar algunos ejemplos de autómatas celulares en una, dos y tres dimensiones. En dichas aplicaciones los alumnos pueden interactuar, ya sea para visualizar los diversos modelos o para modificar la programación de manera de lograr otros ejemplos diferentes.

2. EL SOFTWARE SCILAB

SCILAB es un lenguaje de programación de alto nivel para cálculo científico. Es de uso libre y está disponible en múltiples sistemas operativos, como por ejemplo Mac OS X, GNU/Linux, Windows. Fue desarrollado por INRIA (Institut National de Recherche en Informatique et Automatique) y la ENPC (École Nationale des Ponts et Chaussées) en 1989. SCILAB es ahora soportado por Scilab Consortium dentro de la fundación Digiteo. Al ser un software libre, siempre se puede disponer de la última versión en Internet.
SCILAB no es un programa de cálculo simbólico, como MAPLE o MATHEMATICA. Está orientado principalmente al trabajo matricial. Esta es una de las grandes ventajas de la utilización del mismo para el desarrollo de este trabajo, ya que los programas que permiten simular autómatas celulares, por su naturaleza se basan en el almacenamiento de datos en forma de arreglos o matrices.

3. DEFINICIÓN DE AUTÓMATA CELULAR

Un autómata celular es un objeto matemático que consiste en un conjunto de celdas, un conjunto finito de estados y un conjunto de reglas de transición. La grilla de celdas puede ser finita o infinita. Cada celda puede estar sólo en uno de los estados del conjunto de estados, y las reglas de transición son las que permiten que las celdas pasen de un estado a otro, simultáneamente, en pasos de tiempo discretos. Estas reglas están basadas en la definición de "vecindad de una celda". Lo más común es que el número de estados sea dos. En ese caso, se suele decir que las celdas están "vivas" o "muertas". Muchas veces, el conjunto completo de celdas se llama generación; cuando se aplican las reglas de transición a una generación, se produce otra generación nueva, y así sucesivamente.
En una dimensión, un autómata celular es un conjunto de celdas alineadas. En dos dimensiones, se pueden utilizar grillas de celdas finitas o infinitas. En caso de ser finitas, pueden adoptar distintas formas: rectangulares, triangulares, hexagonales, entre otras. En tres dimensiones, los conjuntos de celdas pueden constituir superficies o volúmenes.
La vecindad de una celda consiste en la celda propiamente dicha y sus celdas vecinas más cercanas, llamadas “vecinos”. En el caso de un autómata unidimensional, la vecindad de una celda es obvia: la de la derecha, y la de la izquierda. Para el caso de un autómata con grilla rectangular bidimensional, las vecindades se definen generalmente de dos maneras:

- ✓ Vecindad de Von Neumann, consiste en la celda y los cuatro vecinos más cercanos, siendo estas la superior, inferior, derecha e izquierda, como muestra la Figura 1(a).
- ✓ Vecindad de Moore, consiste en la celda y los ocho vecinos más cercanos, según se muestra en la figura 1(b).

En estas figuras se indica la celda en naranja, y los vecinos en azul.

Figura 1. (a) Vecindad de Von Neumann y (b) Vecindad de Moore

En el caso de una grilla finita bidimensional, los vecinos más cercanos a lo largo de los bordes de la grilla se determinan de diferentes maneras, según se consideren fronteras periódicas, absorbentes o sesgadas [5]. A continuación se determinan las reglas que definen los diferentes tipos de fronteras, y se muestran gráficamente siguiendo el mismo código de colores de la figura 1.

Fronteras periódicas

1. El vecino más cercano izquierdo de la primera celda de una fila es la última celda de la misma fila.
2. El vecino más cercano derecho de la última celda de una fila es la primera celda de la misma fila.
3. El vecino más cercano de arriba de la primera celda de una columna es la última celda de la misma columna.
4. El vecino más cercano de abajo de la última celda de una columna es la primera celda de la misma columna.

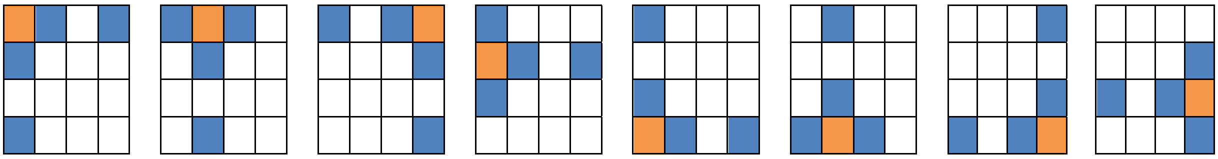


Figura 2. Vecinos Von Neumann en fronteras periódicas

Fronteras absorbentes

1. No hay vecino más cercano izquierdo de la primera celda de una fila.
2. No hay vecino más cercano derecho de la última celda de una fila.
3. No hay vecino más cercano de arriba de la primera celda de una columna.
4. No hay vecino más cercano de abajo de la última celda de una columna.

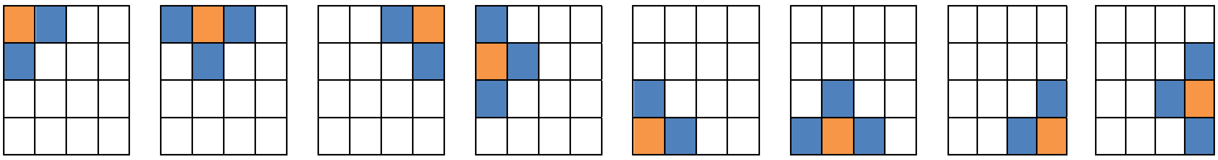


Figura 3. Vecinos Von Neumann en fronteras absorbentes

Fronteras sesgadas

1. El vecino más cercano izquierdo de la primera celda del borde izquierdo es la última celda del borde derecho.
2. El vecino más cercano izquierdo de otra celda cualquiera del borde izquierdo es la última celda de la fila anterior.
3. El vecino más cercano derecho de la última celda del borde derecho es la primera celda del borde izquierdo.
4. El vecino más cercano derecho de otra celda cualquiera del borde derecho es la primera celda de la fila posterior.
5. El vecino más cercano de arriba de una celda cualquiera del borde superior es la última celda de la misma columna.
6. El vecino más cercano de abajo de una celda cualquiera del borde inferior es la primera celda de la misma columna.

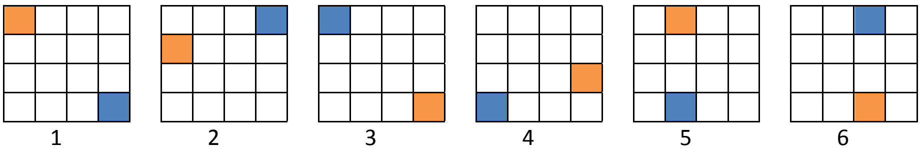


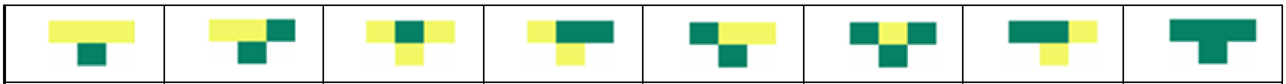
Figura 4. Vecinos en los bordes según las reglas para fronteras sesgadas

Las reglas de un autómata celular se definen de acuerdo a lo que se desea modelar. Estas reglas definen el estado de una celda en el próximo paso de tiempo, dado el estado de la celda en el paso de tiempo actual, que depende de la ocupación de la vecindad. A continuación se verán diferentes ejemplos de autómatas celulares, en una, dos y tres dimensiones.

4. AUTÓMATAS CELULARES ELEMENTALES

Estos autómatas celulares unidimensionales tienen dos valores posibles para cada celda y reglas para determinar ese valor, que dependen sólo de los valores de los vecinos más cercanos. Como resultado, la forma en que evolucionarán estos autómatas celulares puede describirse por una tabla que especifique el estado que una celda dada tendrá en la próxima generación, en función de los valores de la misma celda y de las ubicadas a su izquierda y a su derecha, en la generación actual. Estas reglas pueden representarse de la manera que se muestra en la Tabla 1. Los dos estados posibles pueden simbolizarse con diferentes colores (lo más usual es blanco y negro, aunque no es indispensable que sea así) o con 0 y 1. La representación con los valores 0 y 1 de la regla presentada en la Tabla 1 puede interpretarse como el número $00110010_2 = 50$, de ahí el nombre de la regla. Estos nombres fueron introducidos por Stephen Wolfram [7].

Tabla 1. Representación gráfica y código binario de la regla 50



0	0	1	1	0	0	1	0

Como para cada celda los distintos estados posibles para las tres celdas consideradas en la generación anterior son 8, hay un total de $2^8 = 256$ autómatas celulares elementales.

La evolución de un autómata celular unidimensional se puede ilustrar a partir del estado inicial (generación cero) en la primera fila, la primera generación en la segunda fila, y así sucesivamente.

El conjunto completo de autómatas celulares elementales, correspondientes a las reglas desde 0 hasta 255 con una configuración inicial de una única celda pueden obtenerse en la aplicación que se muestra en las Figuras 5 y 6. En esta ventana pueden elegirse la regla a utilizar y la cantidad de generaciones que quieren mostrarse. Con el botón **Ver la regla**, se obtiene la representación gráfica de la regla que se está utilizando.

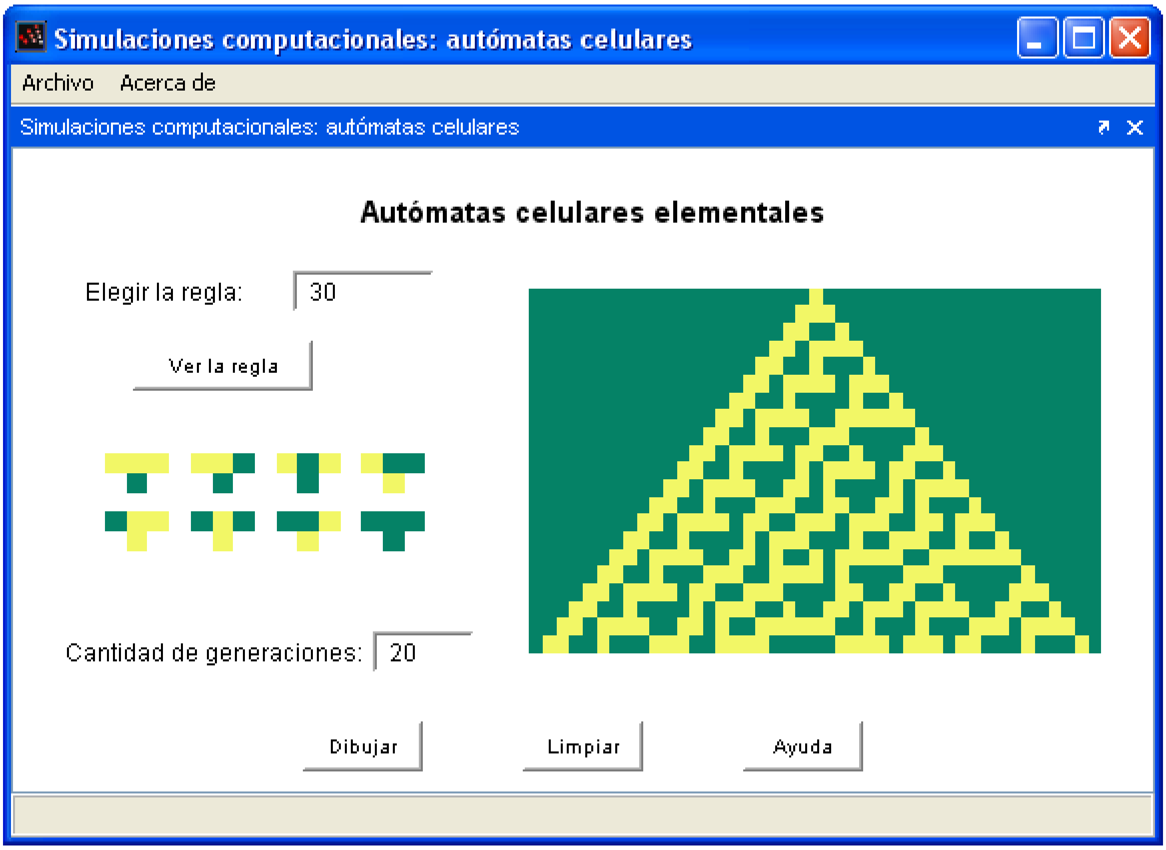


Figura 5. Ejemplo que muestra 20 generaciones de la regla 30

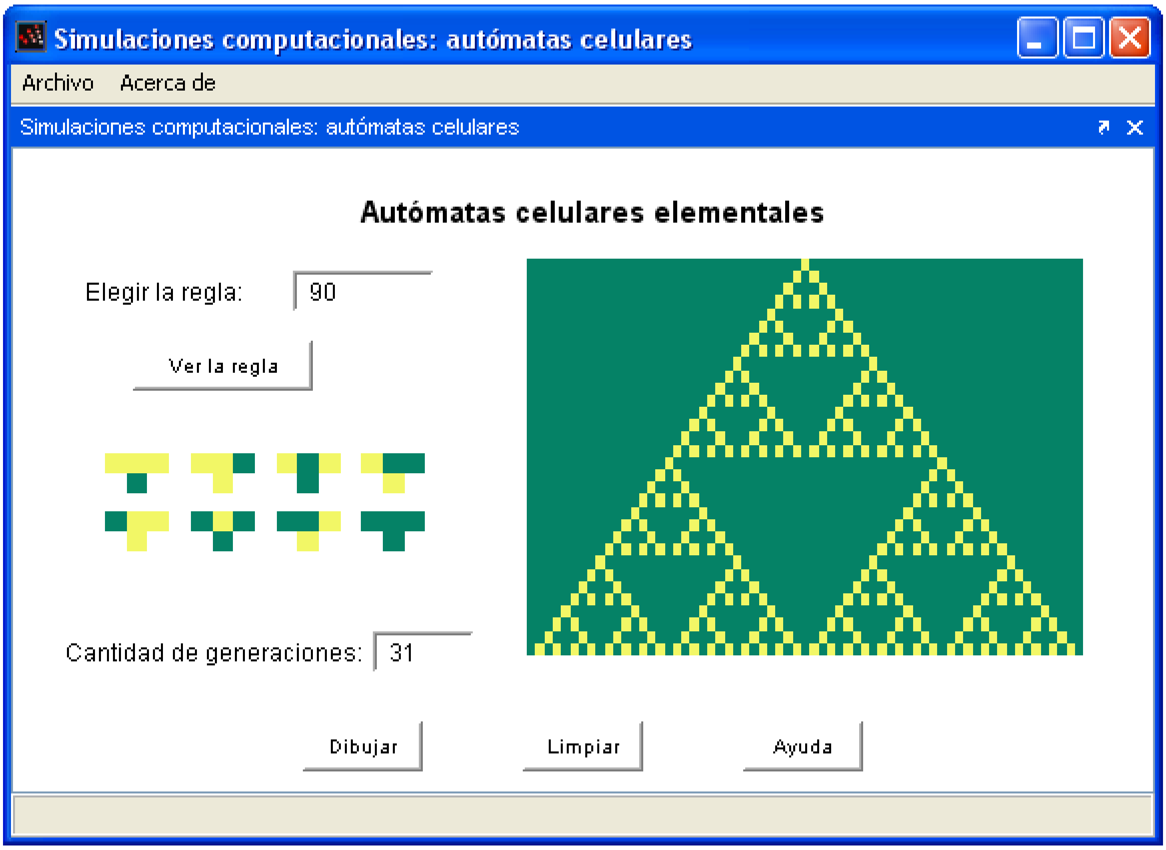


Figura 6. Ejemplo que muestra 31 generaciones de la regla 90

5. EL JUEGO DE LA VIDA Y SIMILARES (EN 2D)

El juego de la vida es un autómata celular bidimensional, creado por John Conway. Fue el primer programa que corrió en "la máquina conectada", la primera computadora en paralelo del mundo. Por otro lado, es el antecesor de los llamados "sistemas de vida artificial", cuyo interés radica no solo por sus implicaciones biológicas sino por el desarrollo de los "agentes inteligentes" para computadoras [5].

El juego de la vida es posiblemente el autómata celular más conocido. Este modelo consiste en una matriz de celdas, definida en una red cuadrada de dos dimensiones, cada una de ellas en el estado 0 ó 1. Una celda interactúa con sus vecinos más cercanos de acuerdo a una función de transición.

La transición para el juego de la vida, en el que se utiliza la vecindad de Moore, se define de la siguiente manera:

- ✓ Si una celda se encuentra en estado 1 y dos o tres de sus vecinos se encuentran en el estado 1, el estado de la celda sigue siendo uno: una celda "viva", con dos o tres vecinos "vivos", permanece "viva".
- ✓ Si una celda se encuentra en estado 0 y tres de sus vecinos están en el estado 1, el estado de la celda se convierte en 1: una celda "nace" si tiene tres vecinos "vivos".
- ✓ En todos los demás casos el estado de una celda se convierte en 0 o permanece en 0.

Las transiciones de las celdas se desarrollan simultáneamente en pasos de tiempo discretos.

Existen numerosos tipos de patrones que pueden tener lugar en el juego de la vida: estáticos, osciladores o patrones que se trasladan por la cuadrícula (naves espaciales). Los ejemplos más simples de estas tres clases de patrones se muestran en la Tabla 2. Las celdas vivas se muestran en amarillo y las muertas en verde.

Tabla 2. Posibles configuraciones iniciales para el juego de la vida

Formas estáticas		Osciladores		Naves espaciales	
					

Un conjunto de autómatas celulares similares al juego de la vida pueden observarse en la aplicación que se muestra en las Figuras 7 y 8.

Figura 7. 12 generaciones de "HighLife" con la configuración 7 como inicial

Figura 8. 100 generaciones de "Seeds" con una configuración inicial definida por el usuario

En esta ventana pueden elegirse el autómata celular a utilizar, la cantidad de generaciones que quieren mostrarse y la configuración inicial.
Los autómatas celulares pueden elegirse de una lista desplegable, que ofrece el juego de la vida y otros con las siguientes reglas:

Seeds:

- ✓Una celda nace si tiene dos vecinos vivos.
- ✓En todos los demás casos las celdas mueren o permanecen muertas.

Day&Night:

- ✓Una celda sobrevive con tres, cuatro, seis, siete u ocho vecinos vivos.
- ✓Una celda nace si tiene tres, seis, siete u ocho vecinos vivos.
- ✓En todos los demás casos las celdas mueren o permanecen muertas.

HighLife:

- ✓Una celda sobrevive con dos o tres vecinos vivos.
- ✓Una celda nace si tiene tres o seis vecinos vivos.
- ✓En todos los demás casos las celdas mueren o permanecen muertas.

Además, pueden elegirse otros autómatas definiendo la cantidad de vecinos vivos que necesita una celda para nacer o para sobrevivir. Esto se indica en los cuadros de diálogo que se muestran en la Figura 5 y que se abren cuando en la lista de autómatas celulares se elige **Otro**. En la Figura 9 se han indicado las reglas del juego de la vida sin muertes.

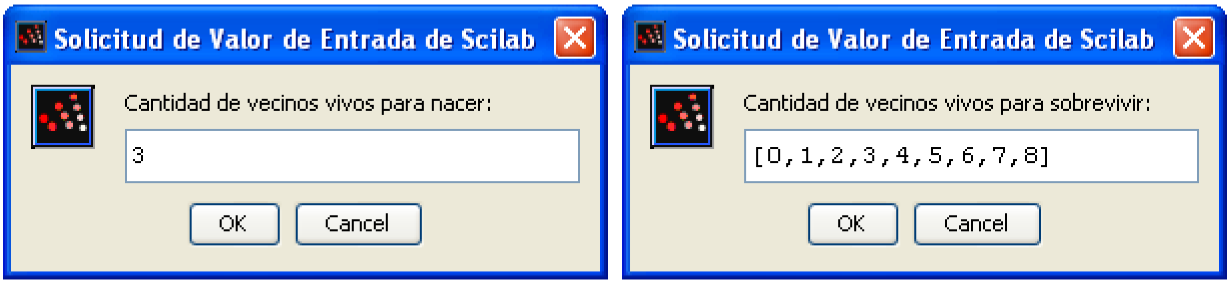


Figura 9. Cuadros en los que se definen las reglas para un nuevo autómata celular

Algunas posibilidades para la configuración inicial se presentan en la interfaz. Éstas pueden elegirse desde la lista desplegable correspondiente. También puede definirse otra configuración inicial, eligiendo la opción **Otra** en la lista. En la Figura 10 se muestra el cuadro de diálogo que se abre para realizar esta acción. El ejemplo presentado (el vector columna de unos) es la configuración inicial utilizada para obtener la salida en la Figura 8.

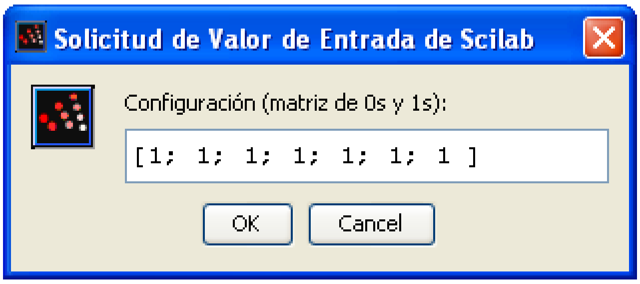


Figura 10. Cuadros en los que se define una configuración inicial diferente

6. UN AUTOMATA CELULAR EN 3D

Se pueden definir también autómatas celulares en tres dimensiones. La grilla es ahora de dimensión 3, y las vecindades de una celda pueden definirse como vecindades de Von Neumann o de Moore. En estos casos, la cantidad de vecinos son 6 y 26 respectivamente.
Se presenta en la figura 11 una ventana desarrollada en Scilab en donde se muestra un autómata celular en tres dimensiones, que es una generalización del juego de la vida. Las celdas en este caso son cubos de dimensión 1x1x1.

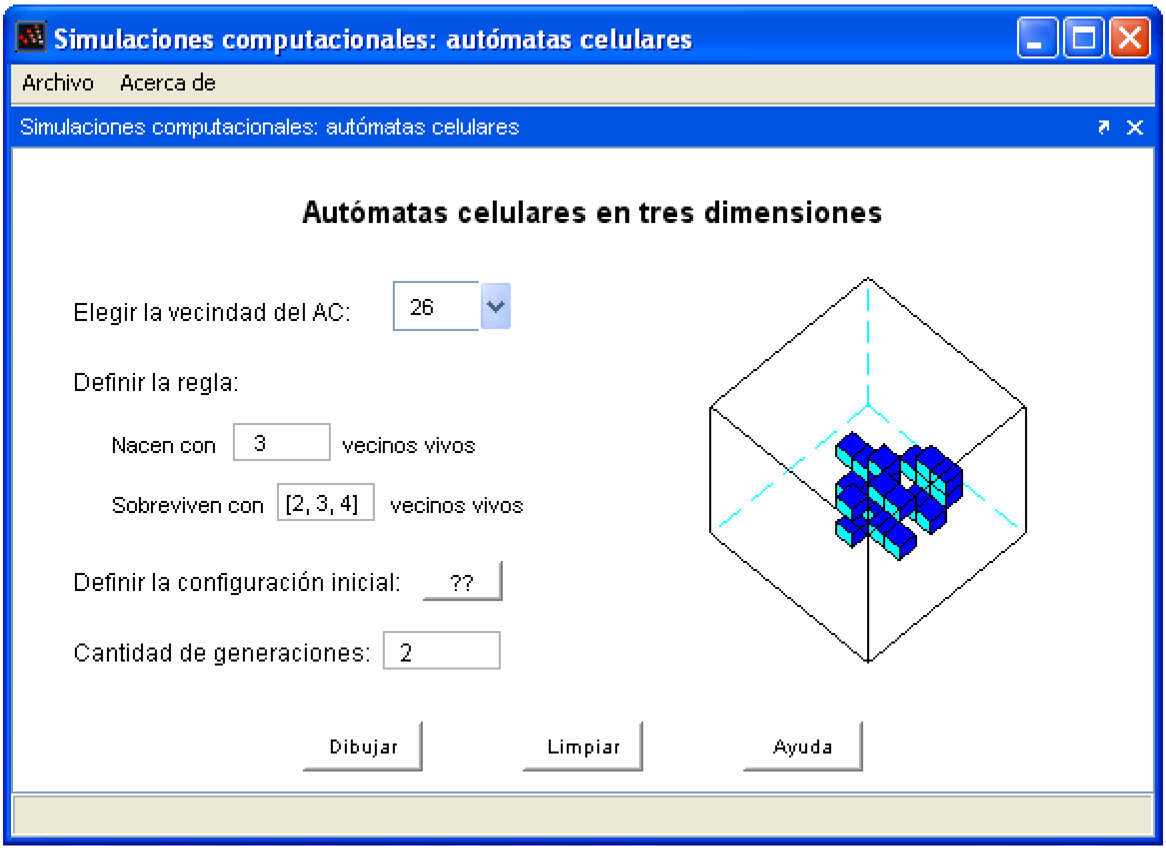


Figura 11. La ventana para autómatas celulares en 3D

En esta ventana se debe elegir el tipo de vecindad con el que se va a trabajar, 6 corresponde al tipo Von Neumann y 26 al tipo Moore. También se deben definir las reglas del juego, es decir, con cuántos vecinos vivos de una celda, hay supervivencia, o se produce un nacimiento o una muerte. También se debe elegir la configuración inicial, es decir, las celdas que se encuentran vivas en la primera generación. Esto se indica ingresando un vértice de cada una de dichas celdas (determinado por convención a partir del cual todas las celdas se construyen con la misma orientación) en el cuadro de diálogo que se muestra en la figura 12.

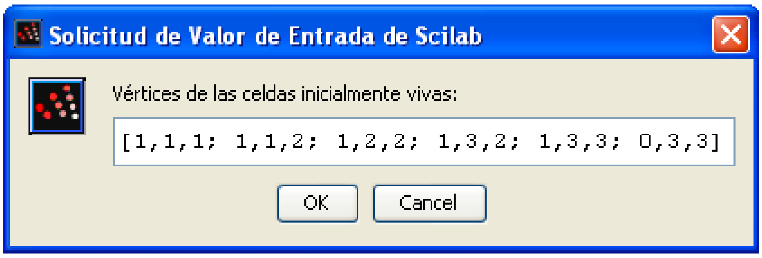


Figura 12. Cuadro para ingresar la generación inicial

7. CONCLUSIONES

El juego de la vida es un modelo de autómata celular, que a pesar de no ser actual, aún está vigente, y sirve de modelo para iniciar a los alumnos en el estudio de las simulaciones discretas. La programación planteada aquí es relativamente sencilla, y puede ser encarada con cualquier lenguaje de programación de alto nivel. Si se utilizan lenguajes distintos a Scilab, se deberá tener en cuenta que probablemente se deba destinar más tiempo a la programación de los gráficos, como se debió hacer en el caso 3D.
La posibilidad de manejar software libre permite a los alumnos trabajar en sus computadoras, permitiéndoles administrar su tiempo de mejor manera, al no depender de la disponibilidad horaria de los laboratorios.

REFERENCIAS

1. "Fundamentals of natural computing. Basic concepts, algorithms, and applications", Leandro Nunes de Castro. Chapman & Halls CRS Taylor & Francis Group (2006)
2. "Lattice-gas automata for the Navier–Stokes equation", U. Frisch, B. Hasslacher, Y. Pomeau. Phys. Rev. Lett. **56** (1986) 1505–1508.
3. "Simulating physics with cellular automata", G. Vichniac. Physica D **10** (1984) 96–115.
4. "Simulando la naturaleza: fenómenos de diffusion", M. Caligaris, G. Rodríguez y R. Caligaris. 2das Jornadas sobre la Enseñanza de los Métodos Numéricos y Empleo de Herramientas de Simulación (EMNUS 2001) Facultad Regional Haedo, 6 y 7 de septiembre de 2001.
5. "Computer Simulations with Mathematica", R. Gaylord & P. Wellin. Telos – Springer Verlag. 1995
6. www.scilab.org
7. "Theory of cellular automata: A survey", J. Kari. Theoretical Computer Science **334** (2005) 3–33