

Simulación de Sistemas

Segundo Cuatrimestre 2019

Trabajo Práctico N°2: Conway's Game of Life



GRUPO 4

Integrantes

Della Sala, Rocío - 56507

Giorgi, María Florencia - 56239

Rodríguez, Ariel Andrés - 56030

Índice

1. Introducción al juego	4
1.1 Introducción al trabajo práctico realizado	4
2. Sistema físico	5
3. Modelo matemático	5
4. Implementación computacional	5
4.1 Implementación 2D	5
4.2 Implementación 3D	6
5. Mediciones	6
5.1 Población	6
5.2 Distancia máxima	6
5.3 Tasa de reproducción	6
5.4 Tasa de mortalidad	6
5.4.1 Tasa de mortalidad por sobrepoblación	6
5.4.2 Tasa de mortalidad por aislamiento	6
5.5 Tasa de supervivencia	6
6. Patrones iniciales 2D	6
6.1 Battle	6
6.2 Linear	7
6.3 Random	7
6.4 Fatality	8
6.5 Cube	8
6.6 Cube2	8
6.7 Oscilator	8
6.8 Oscilator2	9
7. Reglas 2D	9
7.1 General 2D	9
7.2 General 2D Modified	9
7.3 Sliced	10
7.4 Sliced Reverse	10
7.5 Parity Flexible	11
7.6 Parity Mortal	11
7.7 Contourn	11
8. Resultados 2D	11
8.1 Número de partículas vivas	11
8.1.1 Gráfico 1 – Cube	12
8.1.2 Gráfico 2 – Battle	12
8.1.3 Gráfico 3 – Linear	13
8.1.4 Gráfico 4 – Fatality	14
8.1.5 Gráfico 5 – Oscilator	15
8.1.6 Gráfico 6 – Random	15
8.1.7 Conclusiones de los gráficos de población	16
8.2 Distancia máxima	17
8.2.1 Gráfico 9 – Linear	17
8.2.2 Gráfico 10 – Battle	18
8.2.3 Gráfico 11 – Random	19
8.3.4 Gráfico 12 – Cube	19

8.3.5 Gráfico 13 – Fatality	20
8.2.6 Gráfico 14 – Oscilator	21
8.2.7 Conclusiones de los gráficos de distancia máxima	22
9. Otros resultados y conclusiones	22
9.1 Resultados con patrón inicial constante y reglas distintas	22
9.1.1 Parity_flexible vs Parity_mortal	22
9.1.2 Sliced vs SlicedReverse	22
9.1.3 General_2d vs General_2d_modified	22
9.2 Resultados con regla constante y patrón inicial distinto	23
9.2.1 Gráfico 16 y Gráfico 17 - Cube vs Cube2	23
10. Patrones iniciales 3D	24
10.1 Random	24
10.2 Linear	24
10.3 Cube	25
10.4 Border	25
10.5 Glider	25
11. Reglas 3D	26
11.1 General 3D	26
11.2 Rule_WXYZ	26
11.3 Reglas derivadas	26
12. Resultados 3D	27
12.1 Número de partículas vivas	27
12.1.1 Gráfico 18 – Cube	27
12.1.2 Gráfico 19 – Glider	27
12.1.3 Gráfico 20 – Border	28
12.1.4 Conclusiones de los gráficos de población	28
12.2 Distancia máxima	29
12.2.1 Gráfico 21 – Glider	29
12.2.2 Gráfico 22 – Cube	29
12.2.3 Gráfico 23 – Border	30
12.2.4 Conclusiones de los gráficos de población	30
13. Conclusiones generales	31
14. Bibliografía	31

1. Introducción al juego

El juego de la vida es un autómata celular diseñado por el matemático John Horton Conway. El mismo simula la evolución de colonias de organismos vivos. Desde un punto de vista teórico, es interesante porque todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida.

Se trata de un juego de cero jugadores donde el tablero (de longitud arbitraria) está formado por una grilla que llamaremos **partícula** y cada una tiene un **estado** particular correspondiente a un valor binario (viva/muerta).

Al iniciar el juego, el tablero tendrá una configuración inicial de partículas vivas y muertas que llamaremos **patrón inicial**. A partir de una o más **reglas** definidas (de carácter determinístico y uniforme), se generará la transición del estado de cada partícula para cada paso temporal. La cantidad de pasos temporales estará definida por el jugador.

Cada patrón inicial puede tener uno de los siguientes 4 comportamientos, que se definirá según la regla aplicada:

1. Desaparecer con el tiempo
2. Evolucionar a un tamaño fijo finito
3. Crecer indefinidamente a una velocidad fija
4. Crecer y contraerse periódicamente

Las reglas están definidas de acuerdo a la vecindad de la partícula. Se define **partícula vecina** a cualquier partícula que esté ubicada en el tablero a una distancia de una unidad, tanto horizontal, vertical u oblicua. Cada partícula tiene 8 partículas vecinas si tenemos en cuenta condiciones periódicas de contorno.

Por último, definiremos la **regla default** (o también llamada general) de dos dimensiones, que será mencionada a lo largo del informe:

- a. Una partícula muerta con exactamente 3 partículas vecinas vivas nace
- b. Una partícula viva con 2 o 3 partículas vecinas vivas, sigue viva.
 - i. Si la partícula tiene más de 3 partículas vecinas vivas, muere de sobrepoblación
 - ii. Si la partícula tiene menos de 2 partículas vecinas vivas, muere de aislamiento

1.1 Introducción al trabajo práctico realizado

En el trabajo realizado se implementó el juego de la vida para 2 y 3 dimensiones en Java. Para cada dimensión, se definieron una serie de reglas que detallaremos en las secciones *7. Reglas 2D* y *10. Reglas 3D*. Además, se generaron varios patrones para cada dimensión detallados en las secciones *6. Patrones iniciales 2D* y *9. Patrones iniciales 3D*.

Las reglas que se definieron más allá de la default del juego fueron implementadas con el fin de mostrar que ante leves cambios en la misma, al aplicarla sobre el mismo patrón, puede generar resultados diferentes en relación con una o varias métricas.

Por otro lado, se definieron varios patrones para constatar que leves cambios en los patrones también generan resultados diferentes.

Para generar los gráficos se utilizó la herramienta Tableau, un software de visualización de datos.

2. Sistema físico

Se define un sistema como el subconjunto del universo, que posee componentes interactuantes, una función de transformación y observables medibles. En este caso, nuestro sistema físico está constituido por las partículas, las reglas del juego y todas las métricas que nos proporcionan datos observables, medibles y cuantificables.

Todo esto se detalla más adelante en las secciones 7. *Reglas 2D* y 10. *Reglas 3D* en el caso de las reglas y 5. *Métricas* en el caso de los datos medibles.

3. Modelo matemático

El modelo de un sistema es la abstracción y simplificación de este. El mismo mapea a través de una cierta función los inputs de los outputs. Para este caso, nuestro input será todos los parámetros que están detallados en el archivo de configuración que se comentará en la sección 4. *Implementación* y nuestro output las estadísticas tomadas, además de la posición y el estado de cada partícula. Ahora explicaremos las características de nuestro modelo.

En primer lugar, tenemos un modelo **no lineal** debido a su propiedad de sensibilidad a las condiciones iniciales. Esto es lo que trataremos de demostrar a lo largo del trabajo mediante gráficos.

En segundo lugar, en este modelo las variables de estado son **discretas**, debido a que como dijimos anteriormente, cada partícula puede tomar un número discreto de estados.

En tercer lugar, el modelo es **determinista**, debido a que conoce todas las condiciones iniciales (patrones iniciales) y leyes de la naturaleza (reglas aplicadas) entonces puede determinar la evolución futura del universo.

En cuarto y último lugar, es un modelo que avanza en **tiempo discreto** debido a que el paso temporal es fijo. El sistema evoluciona cada un cierto tiempo discreto, generalmente pequeño.

4. Implementación computacional

Para la implementación de la simulación se utilizó una matriz de 2 dimensiones para el caso 2D y una matriz de 3 dimensiones para el caso 3D. Internamente cada celda contiene una partícula definida en la clase *Particle* la cual tiene una posición en (x,y) para el primer caso o (x,y,z) para el segundo caso, un id y un estado (viva o muerta). En cada paso de la simulación se recorre la matriz celda por celda para calcular la cantidad de vecinos correspondientes de cada celda para así aplicar la regla correspondiente y definir la transición al próximo estado.

Para poder estudiar el comportamiento de la evolución del sistema, se definieron una serie de métricas que serán detalladas en la sección 5. *Mediciones* con las cuales se realizaron los gráficos.

Para ambas dimensiones se realizó un archivo de configuración llamado *configuration_file.txt* ubicado en la carpeta *src/* desde el cual se levantan los parámetros correspondientes que definen la longitud de la partícula, la cantidad de partículas por fila, columna y altura, la dimensión, la regla a usar, el patrón inicial y la cantidad de pasos/iteraciones.

4.1 Implementación 2D

Para el caso de dos dimensiones, se estudió el alrededor de cada celda la cual puede tener como máximo 8 celdas vecinas. Se implementaron condiciones periódicas de contorno únicamente para este caso.

4.2 Implementación 3D

Para el caso de tres dimensiones, se estudió el alrededor de cada celda la cual puede tener como máximo 26 celdas vecinas.

5. Mediciones

Para generar los gráficos que veremos en las secciones 8. *Resultados 2D* y 11. *Resultados 3D*, se definieron distintas métricas ubicadas en la clase *Statistics* que mencionaremos a continuación:

5.1 Población

Establece la cantidad de partículas vivas que hay en cada transición del juego.

5.2 Distancia máxima

Establece la distancia de la partícula más lejana al centro que hay en cada transición del juego. Cabe destacar que el centro se calculó en función del largo, ancho y altura del universo utilizado en cada prueba.

5.3 Tasa de reproducción

Establece la tasa de reproducción que hay en cada transición del juego.

5.4 Tasa de mortalidad

Establece la tasa de mortalidad que hay en cada transición del juego.

5.4.1 Tasa de mortalidad por sobrepoblación

Establece la tasa de mortalidad por sobrepoblación que hay en cada transición del juego. Por ejemplo, cuando una partícula viva tiene más vecinos que los que necesita para sobrevivir. Para el caso de la regla default del juego, sería tener más de 3 vecinos vivos.

5.4.1 Tasa de mortalidad por aislamiento

Establece la tasa de mortalidad por aislamiento que hay en cada transición del juego. Por ejemplo, cuando una partícula viva tiene menos vecinos que los que necesita para sobrevivir. Para el caso de la regla default del juego, sería tener menos de 2 vecinos vivos.

5.5 Tasa de supervivencia

Establece la tasa de supervivencia que hay en cada transición del juego.

6. Patrones iniciales 2D

Para la implementación en 2 dimensiones, se definieron 8 patrones iniciales.

Es importante destacar que los patrones se clasifican de acuerdo a su comportamiento. Existen patrones estáticos (que no cambian entre iteraciones), osciladores (que a medida que pasan las iteraciones repiten N configuraciones infinitamente) y naves espaciales que se transportan a lo largo de la grilla.

6.1 Battle

Patrón inicial que llamamos *battle* (pero se lo conoce como *Gosper glider gun*) debido a que simula la batalla entre naves espaciales. Fue el primer patrón encontrado de carácter finito que crece indefinidamente.

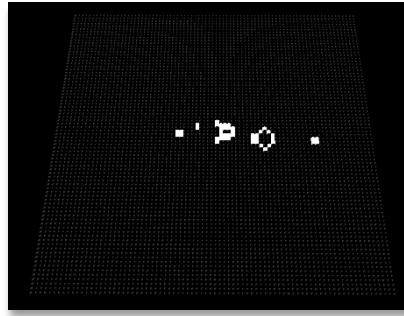


Figura 1: Patrón inicial battle

6.2 Linear

Patrón inicial que llamamos *linear* por estar conformado por tres líneas verticales paralelas entre sí ubicadas a una distancia dada por la ecuación (1)

$$D = \text{particulas_por_fila} / 4 \quad (1)$$

donde D es la distancia entre las líneas y *particulas_por_fila* es la cantidad de partículas que existen en cada fila del tablero.

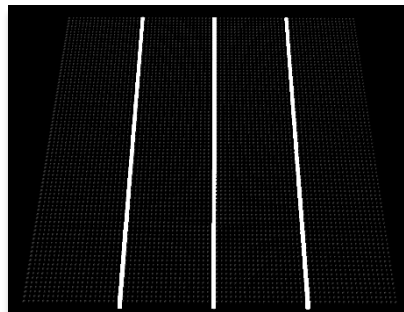


Figura 2: Patrón inicial linear

6.3 Random

Patrón inicial que llamamos *random* debido a que se genera una configuración inicial de partículas vivas de forma aleatoria. Para cada partícula, se genera un número aleatorio que de ser menor a 0.8, se determina que la partícula está viva.

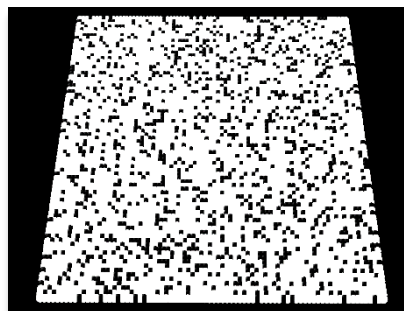


Figura 3: Patrón inicial random

6.4 Fatality

Patrón inicial que llamamos *fatality* debido a que al ser utilizado con la regla default de 2 dimensiones su comportamiento es de desaparecer con el paso de las transiciones del juego.

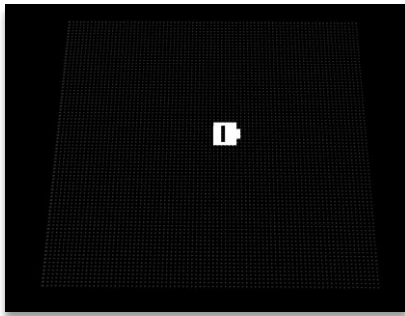


Figura 4: Patrón inicial fatality

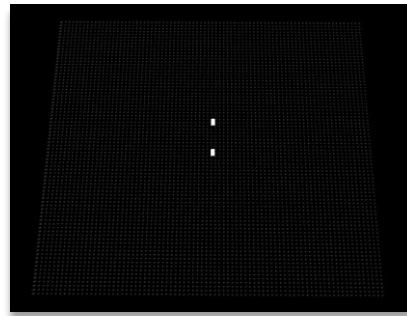


Figura 5: Patrón Fatality en transición 59

6.5 Cube

Patrón inicial que llamamos *cube* debido a que está formado con 4 partículas que forman un cuadrado.

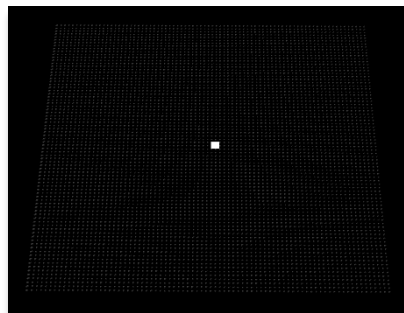


Figura 6: Patrón inicial cube

6.6 Cube2

Patrón inicial que llamamos *cube2* debido a que es una leve modificación del patrón visto en la sección 6.5 *Cube* al cual se le agregan 6 partículas nuevas. La idea de este patrón es estudiar el comportamiento bajo la misma regla que se utiliza para *cube*.

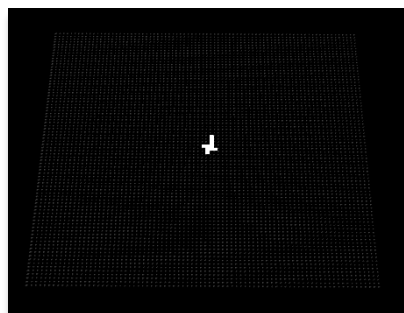


Figura 7: Patrón inicial cube2

6.7 Oscillator

Patrón inicial que llamamos *oscillator* debido a que al ser utilizado con la regla default de 2 dimensiones su comportamiento es de iterar infinitamente entre 3 estructuras distintas.

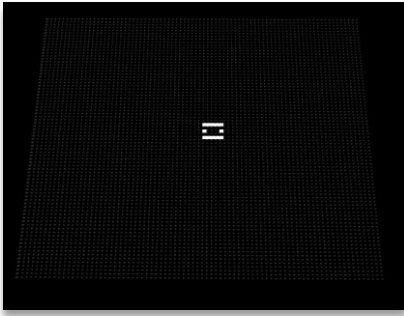


Figura 8: Patrón inicial oscillator

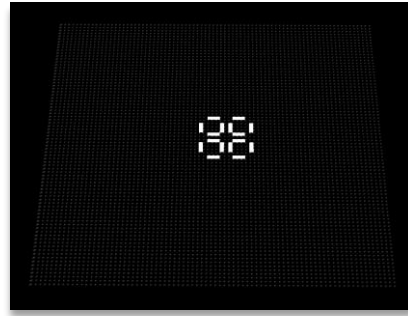


Figura 9: Estructura oscilatoria 1

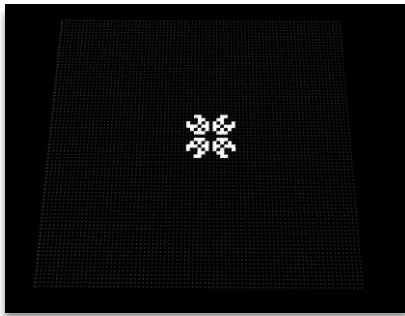


Figura 10: Estructura oscilatoria 2

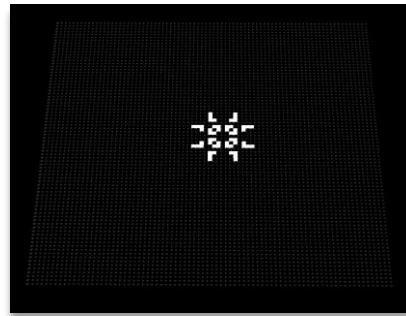


Figura 11: Estructura oscilatoria 3

6.8 Oscillator2

Patrón inicial que llamamos *oscillator2* debido a que es una leve modificación del patrón visto en la sección 6.6 *Oscillator* al cual no se le agregan partículas nuevas, pero sí se cambian dos de ellas de lugar.

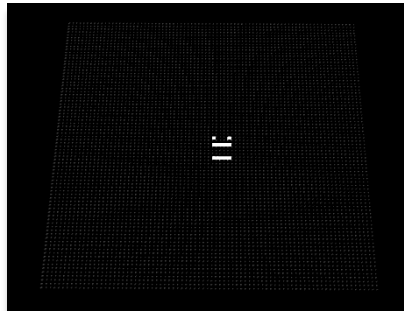


Figura 12: Patrón inicial oscillator2

7. Reglas 2D

Para la implementación en 2 dimensiones, se definieron 7 reglas que se detallarán a continuación.

7.1 General 2D

Regla que llamamos *general_2d* debido a que es la regla default del juego. La misma se definió en la sección 1. *Introducción al juego*.

7.2 General 2D Modified

Regla que llamamos *general_2d_modified* debido a que es una regla con una leve modificación de la regla original *general_2d* vista en la sección 7.1 *General 2D*. La regla se define de la siguiente forma:

- a. Si la partícula tiene exactamente 2 o 3 partículas vecinas vivas, entonces en el siguiente paso vive independientemente si estaba viva o muerta
- b. Si la partícula tiene menos de 2 o más de 3 partículas vecinas vivas entonces muere

7.3 Sliced

Regla que llamamos *sliced* debido a que revisa sólo la mitad izquierda y la celda superior de la partícula en cuestión. La regla se define de la siguiente forma:

- a. Si la partícula está viva y tiene una cantidad mayor o igual de partículas vivas en las posiciones en verde de la figura mostrada a continuación que partículas vivas en las posiciones en rojo, en la siguiente transición la partícula seguirá con vida
- b. Si la partícula está muerta y tiene una cantidad estrictamente mayor de partículas vivas en las posiciones en verde que en las posiciones en rojo, en la siguiente transición, la partícula estará viva.
- c. Si la cantidad de partículas vivas en las posiciones en rojo es mayor que la cantidad de partículas vivas en las posiciones en verde, en la próxima transición la partícula muere.
- d. Si no tiene partículas vecinas vivas, en la próxima etapa muere.

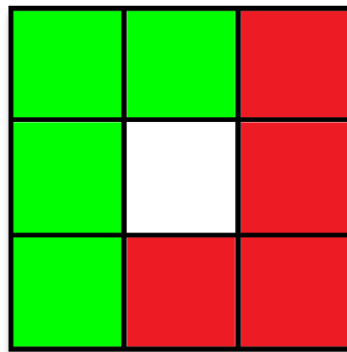


Figura 13: Regla Sliced

7.4 Sliced Reverse

Regla que llamamos *slicedreverse* debido a que es una modificación a la regla *sliced* de la sección 7.3 *Sliced*. La regla funciona de la misma forma que la anterior pero de manera simétricamente inversa, es decir, las que antes eran las posiciones en verde ahora son rojas y viceversa como se muestra a continuación.

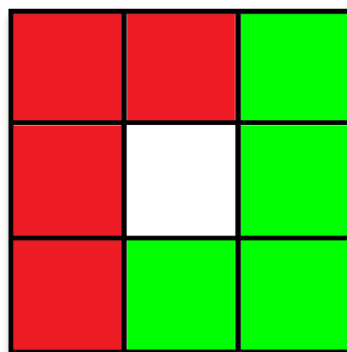


Figura 14: Regla SlicedReverse

7.5 Parity Flexible

Regla que llamamos *parity_flexible* debido a que la supervivencia o reproducción de una partícula depende de si tiene o no una cantidad par de partículas vecinas vivas. El término “flexible” se utilizó debido a los resultados obtenidos con dicha regla que demostraron que en la mayoría de los escenarios probados se obtenían universos crecientes. Esto se detalla más adelante en la sección 8. *Resultados 2D*. La regla se define de la siguiente forma:

- Una partícula muerta con una cantidad de partículas vecinas vivas múltiplo de 2 que no sea cero, nace
- Una partícula viva con una cantidad de partículas vecinas vivas múltiplo de 4 que no sea cero, se mantiene viva
- Cualquier otro caso, la partícula muere o permanece muerta.

7.6 Parity Mortal

Regla que llamamos *parity_mortal* debido a que tiene la característica de destruir los patrones luego de una cantidad de transiciones determinada. Es importante destacar que esta regla se derivó de *parity_flexible*. La regla se define de la siguiente forma:

- Una partícula muerta con una cantidad de partículas vecinas vivas múltiplo de 4 que no sea cero, nace
- Una partícula viva con una cantidad de partículas vecinas vivas múltiplo de 2 que no sea cero, se mantiene viva
- Cualquier otro caso, la partícula muere o permanece muerta.

7.7 Contourn

Regla que llamamos *contourn* debido a que revisa las esquinas y los lados de la celda para determinar la supervivencia o mortalidad de la partícula. La regla se define de la siguiente forma:

- Si la partícula está viva y tiene una cantidad igual o superior de partículas vivas en las posiciones en verde de la imagen mostrada a continuación que de partículas vivas en las posiciones en rojo, permanecerá con vida en la próxima etapa.
- Si la partícula está muerta y tiene una cantidad estrictamente mayor de partículas vivas en las posiciones en verde que de partículas vivas en rojo, la partícula vivirá en la próxima transición.
- Si la cantidad de partículas vivas en las posiciones en rojo es mayor que la cantidad de partículas vivas en las posiciones en verde, en la próxima transición la partícula muere.
- Si no tiene partículas vecinas vivas, en la próxima etapa muere.

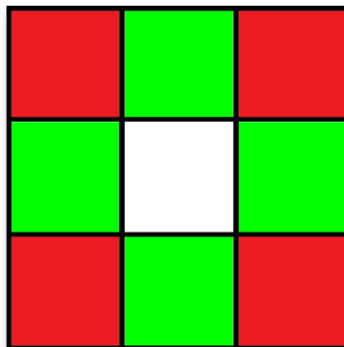


Figura 15: Regla Contourn

8. Resultados 2D

A continuación, se detallan los resultados obtenidos

8.1 Número de partículas vivas

En los siguientes gráficos se realizó una medición de la cantidad de población a lo largo de cada transición, cambiando el patrón inicial utilizado.

8.1.1 Gráfico 1 - Cube

Para el primer gráfico se realizó una única corrida del patrón **cube** con cada una de las reglas dispuestas en la columna de la derecha. Podemos observar que las únicas reglas que maximizan la población (partículas vivas) respecto de la población inicial son *general_2d_modified* y *parity_flexible*. Para el caso de *parity_mortal*, el patrón desaparece y para los casos de *general_2d* y *contourn*, el patrón permanece sin cambiar su estructura, con las mismas 4 partículas vivas.

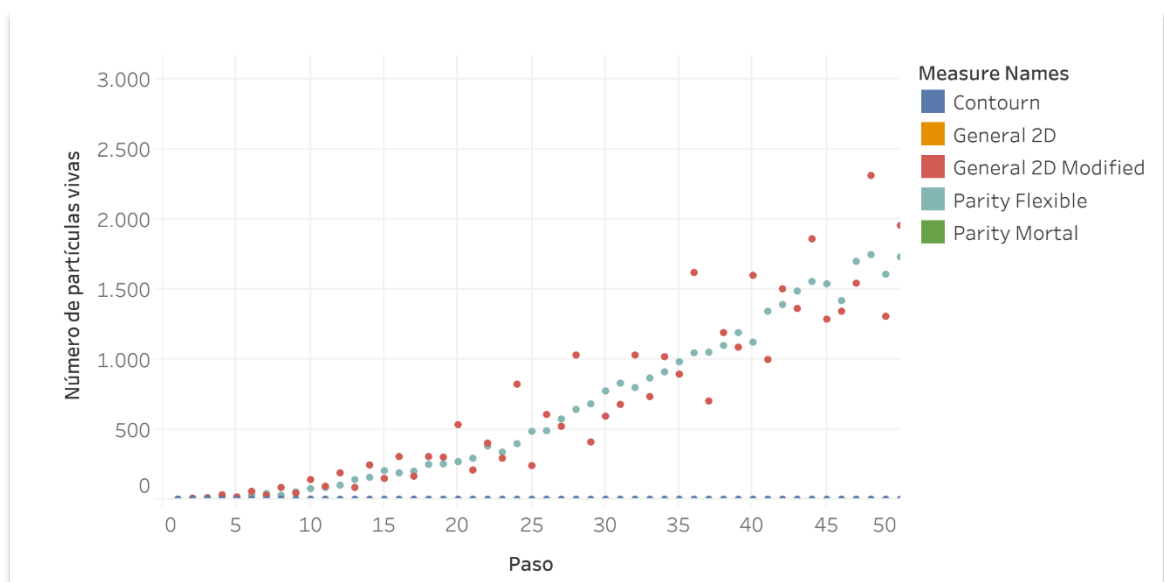


Gráfico 1: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón **cube** con las reglas *contourn*, *general_2D*, *general_2D_modified*, *parity_flexible* y *parity_mortal*

8.1.2 Gráfico 2 - Battle

Para el segundo gráfico se realizó una única corrida del patrón **battle** con cada una de las reglas dispuestas en la columna de la derecha. Podemos observar que en comparación al *Gráfico 1* en este caso todas las reglas utilizadas maximizan la población respecto de la población inicial salvo *parity_mortal* que en 5 transiciones hace desaparecer la estructura. Además, se llega a un máximo de partículas mayor que en el gráfico anterior. Por ejemplo, *general_2d_modified* llega a su máximo que es aproximadamente 2600 partículas vs 2400 partículas del *Gráfico 1*.

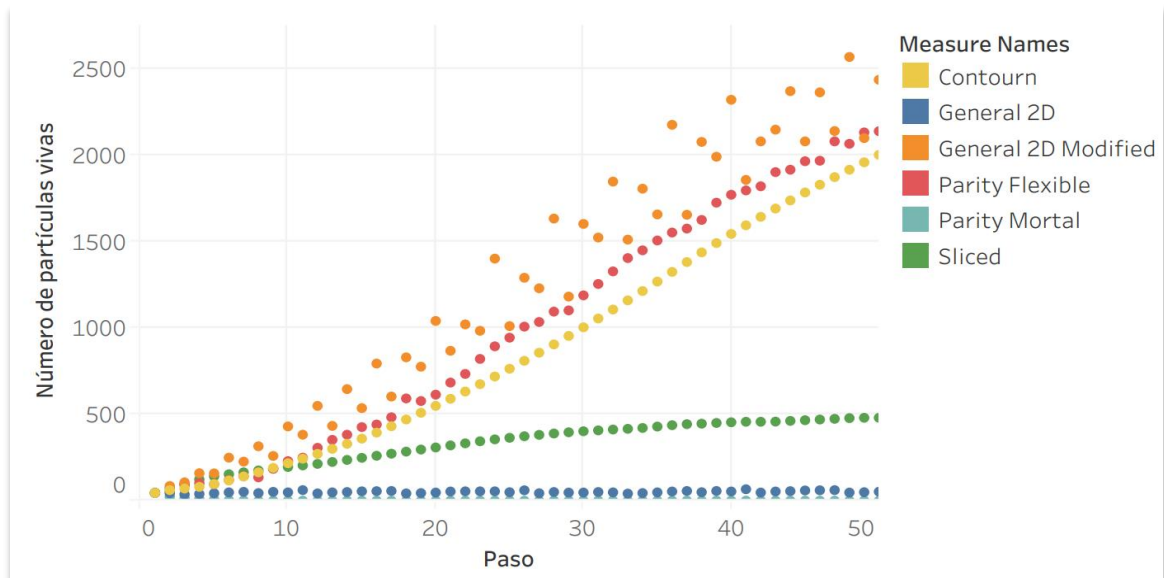


Gráfico 2: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón battle con las reglas contourn, general_2D, general_2D_modified, parity_flexible, parity_mortal y sliced

8.1.3 Gráfico 3 - Linear

Para el tercer gráfico se realizó una única corrida del patrón **linear** con cada una de las reglas dispuestas en la columna de la derecha. Podemos observar que únicamente las reglas *general_2d* y *general_2d_modified* maximiza la población respecto de la población inicial, teniendo un comportamiento oscilatorio. Ambas reglas son iguales para este patrón, por ende, en el Gráfico 2 no podemos observar ambas debido a que están superpuestas. Para el caso de *sliced* que se mantiene constante, la regla desplaza las líneas hacia la izquierda. Al tener condiciones periódicas, las mismas vuelven a aparecer por el borde derecho de la grilla. De este modo la población se mantiene constante. El patrón permanece sin cambiar su estructura al aplicar *contourn* o *parity_mortal*. Por último, con *parity_flexible* el patrón desaparece.

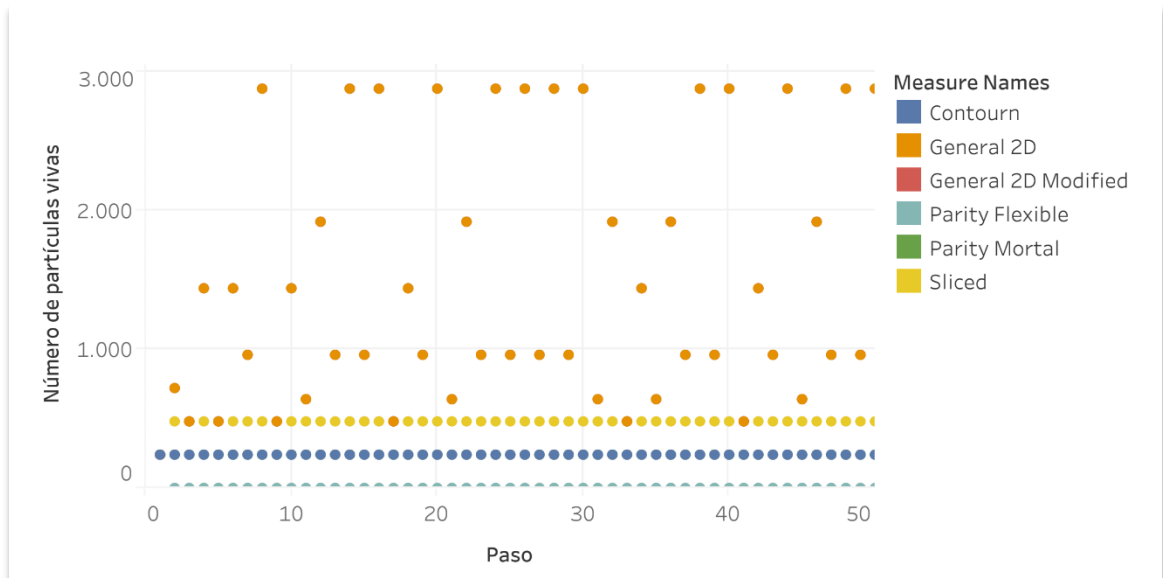


Gráfico 3: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón linear con las reglas contourn, general_2D, general_2D_modified, parity_flexible, parity_mortal y sliced

8.1.4 Gráfico 4 - Fatality

Para el cuarto gráfico se realizó una única corrida del patrón **fatality** con cada una de las reglas dispuestas en la columna de la derecha. Podemos observar que solo dos reglas maximizan la población respecto de la población inicial. Además, cabe destacar que como dijimos en 6.4 *Fatality*, el patrón había sido creado porque tenía un comportamiento característico que consiste en desaparecer al aplicar la regla *general_2d*. Eso lo podemos visualizar en el gráfico ya que no se ven puntos naranjas, aunque el patrón termina de desaparecer en la iteración 61. Sumado a esto, el patrón también desaparece con *parity_mortal*.

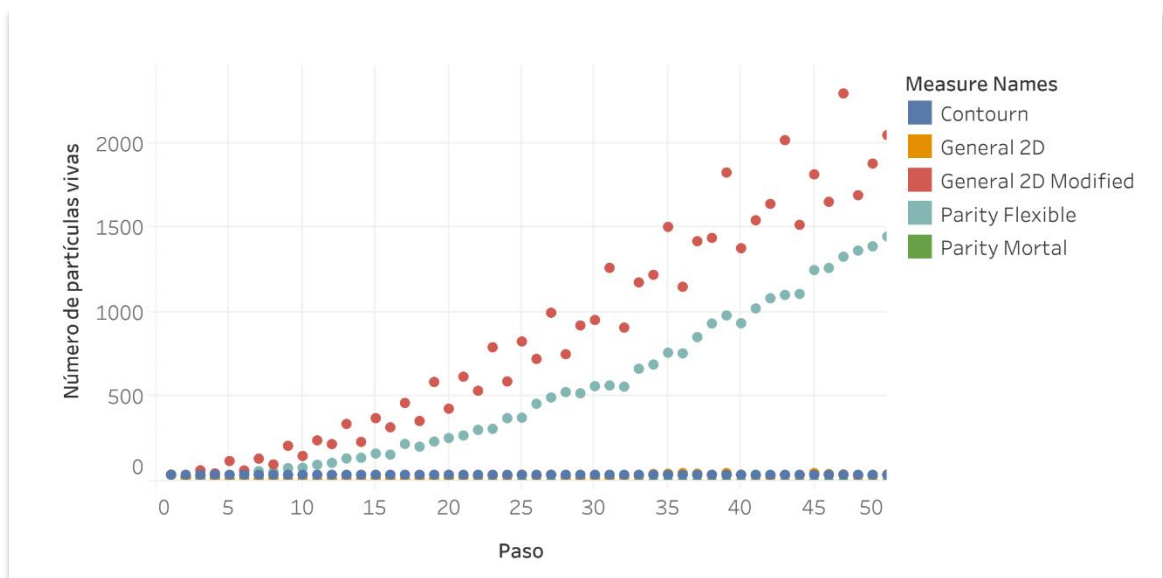


Gráfico 4: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón fatality con las reglas contourn, general_2D, general_2D_modified, parity_flexible y parity_mortal

8.1.5 Gráfico 5 - Oscilator

Para el quinto gráfico se realizó una única corrida del patrón oscilator con cada una de las reglas dispuestas en la columna de la derecha. Podemos observar que ciertas reglas maximizan la población inicial. Además, aparece un comportamiento nuevo que no habíamos visto que es el de *contourn*, aumentando su población y estabilizándose en las transiciones siguientes. También podemos reiterar lo que se vio en los gráficos anteriores, *parity_mortal* hace desaparecer el patrón.

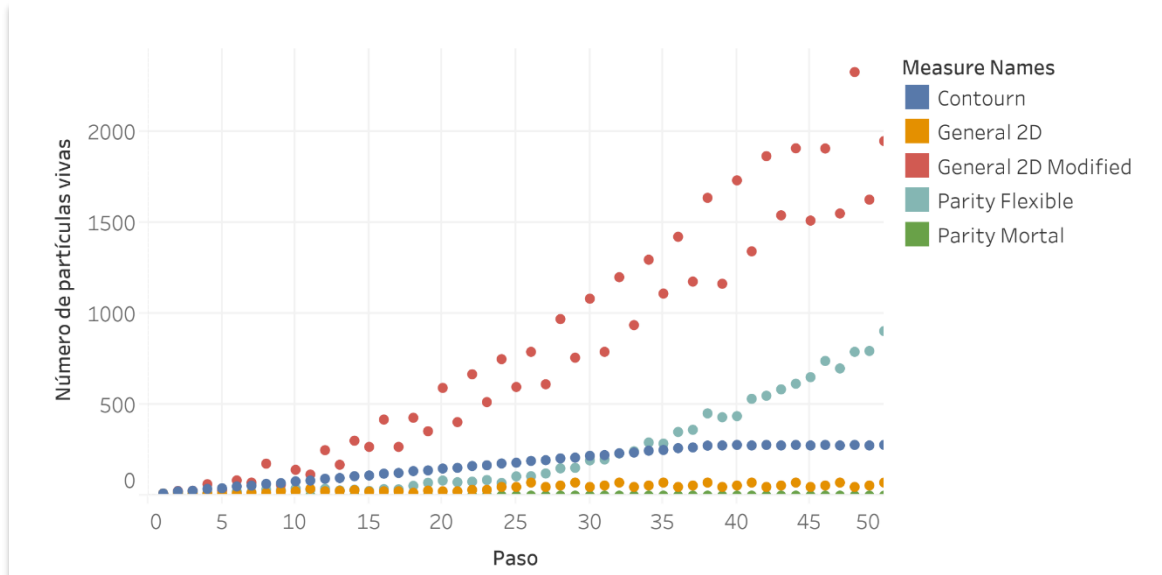


Gráfico 5: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón oscilator con las reglas *contourn*, *general_2D*, *general_2D_modified*, *parity_flexible* y *parity_mortal*

8.1.6 Gráfico 6 - Random

Para el sexto gráfico se realizaron 10 corridas del patrón **random** con cada una de las reglas dispuestas en la columna de la derecha, y luego se calculó un promedio. No se muestran barras de error en el Gráfico 6 debido a que la escala no lo permite. En el Gráfico 7 se muestran a modo de ejemplo las barras de error para la regla *general_2d_modified* que representan un desvío estándar. Esta regla fue la única que mostró un desvío estándar lo suficientemente alto como para que sea observable en una escala acorde.

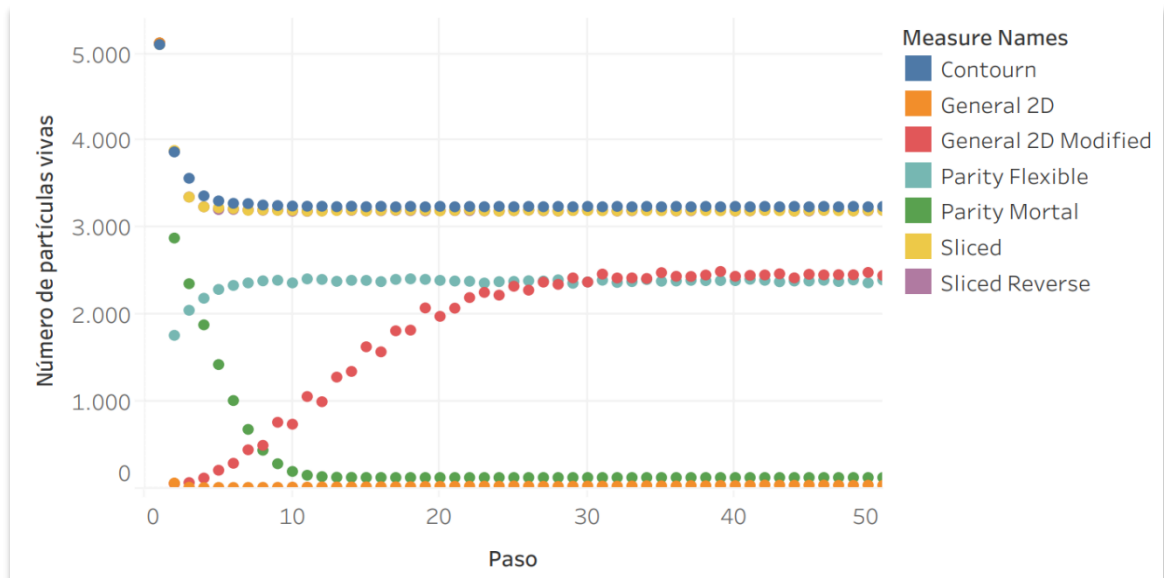


Gráfico 6: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón random con las reglas contour, general_2d, general_2d_modified, parity_flexible, parity_mortal, sliced y sliced_reverse

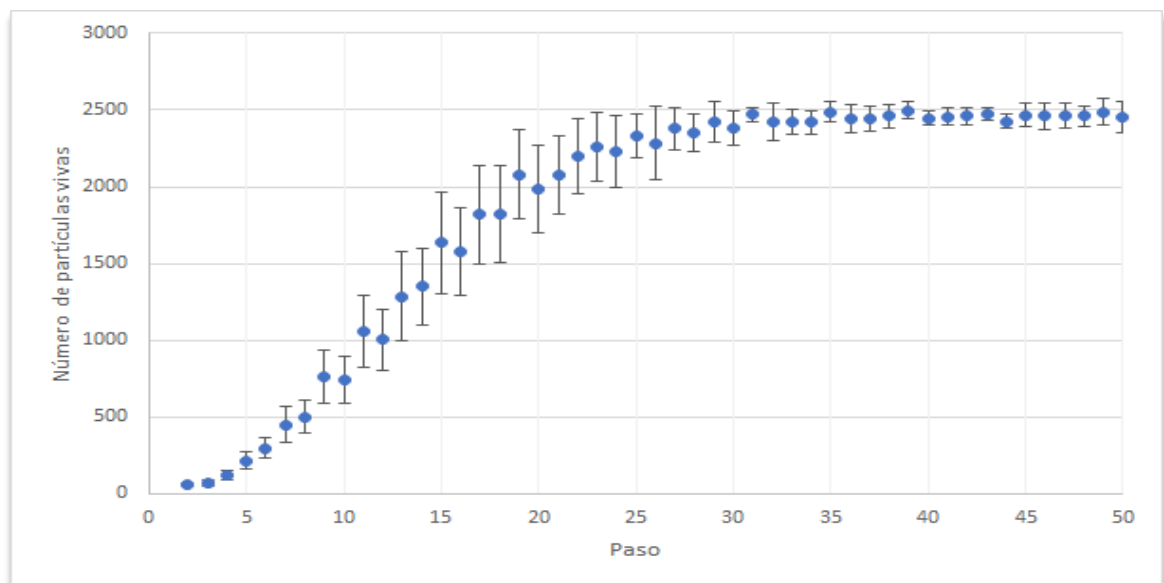


Gráfico 7: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón random con la regla general_2d_modified. Las barras de error representan el desvío estándar.

8.1.7 Conclusiones de los gráficos de población

A partir de los gráficos analizados anteriormente se pueden extraer conclusiones acerca de las reglas. Las mismas se detallarán mediante la siguiente lista:

- i. Se puede clasificar el comportamiento de cada regla, al igual que se puede hacerlo con los patrones
 - a. La regla *general_2d_modified* tiene un carácter expansivo, en general la población creció en comparación al patrón original salvo en linear donde osciló.
 - b. La regla *parity_mortal* tiene un carácter destructivo, en todos los gráficos anteriores el patrón desapareció.

- c. La regla *contourn* tiene un carácter estable. En algunos casos modificó levemente la población que luego quedó estable y en otros nunca la modificó.
- d. La regla *general_2d* tiene un carácter completo o complejo, porque podemos ver distintos comportamientos al aplicarla. En el *Gráfico 1* la regla no modifica el patrón. En el *Gráfico 2* si modifica el patrón, pero se llega a una configuración estable al aplicar reiteradas veces la regla, es decir oscila al igual que el *Gráfico 3* y *Gráfico 5*. Por último, en el *Gráfico 4* hace desaparecer al patrón. Esto refuerza lo detallado al principio del informe, que el juego de la vida de Conway es Turing-completo.

Para tener una comprensión aún mayor del comportamiento de los patrones se hicieron mediciones respecto de las tasas de mortalidad, supervivencia y reproducción de los patrones respecto de las reglas aplicadas. A partir del siguiente gráfico podemos ver, por ejemplo, cómo va aumentando la tasa de mortalidad para el patrón Battle al aplicarle la regla *contourn* mientras que se mantiene relativamente bajo y constante al aplicarle la regla *general_2d*.

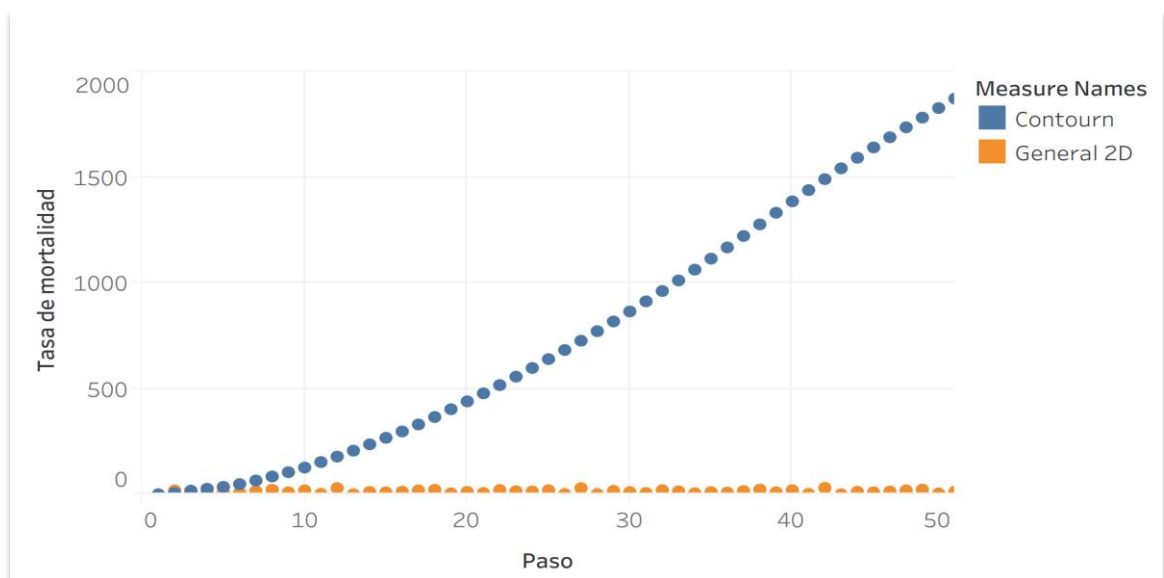


Gráfico 8: Evolución de la tasa de mortalidad en el tiempo, medido en pasos, para el patrón battle con las reglas *contourn* y *general_2d*

8.2 Distancia máxima

8.2.1 Gráfico 9 - Linear

En este caso se realizó una única corrida del patrón **linear** con cada una de las reglas dispuestas en la columna de la derecha. Se puede observar que las reglas que maximizan la distancia máxima son *parity_flexible*, *sliced* y *general_2d_modified*, las cuales son reglas expansivas. Mientras que las demás reglas, mantienen valores muy bajos de distancia máxima.

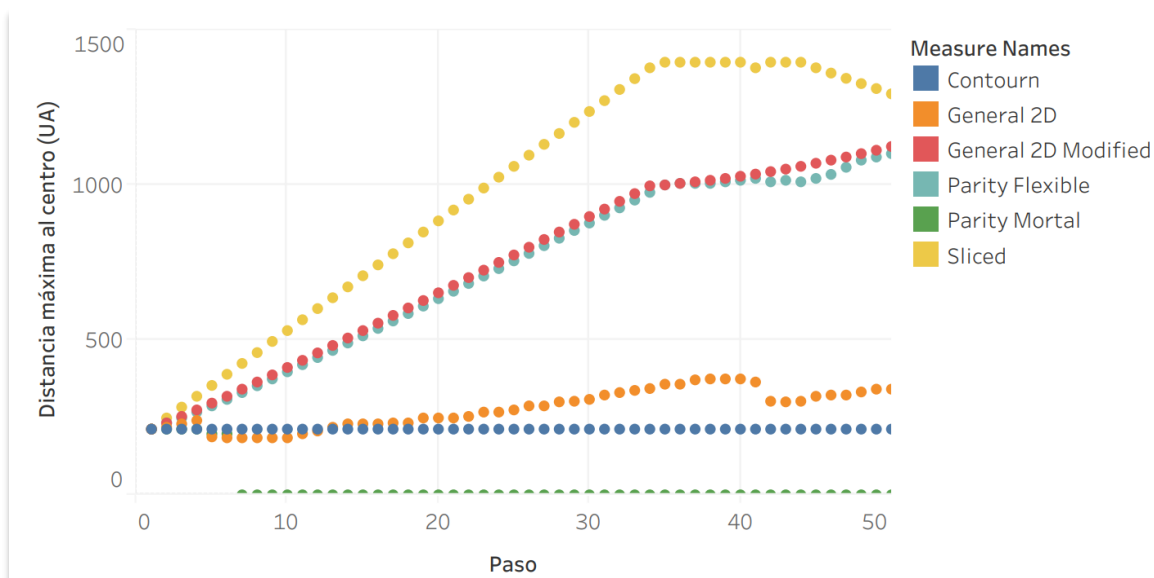


Gráfico 9: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón linear con las reglas *contourn*, *general_2D*, *general_2D_modified*, *parity_flexible*, *parity_mortal* y *sliced*

8.2.2 Gráfico 10 - Battle

Para este gráfico se realizó una única corrida del patrón battle con cada una de las reglas dispuestas en la columna de la derecha. Se pudo observar una clara diferencia entre aquellas reglas que priorizan la supervivencia y la reproducción de la población ya que estas admiten valores elevados de distancias máximas. En cambio, las que presentan más restricciones muestran valores bajos o nulos de distancia máxima.

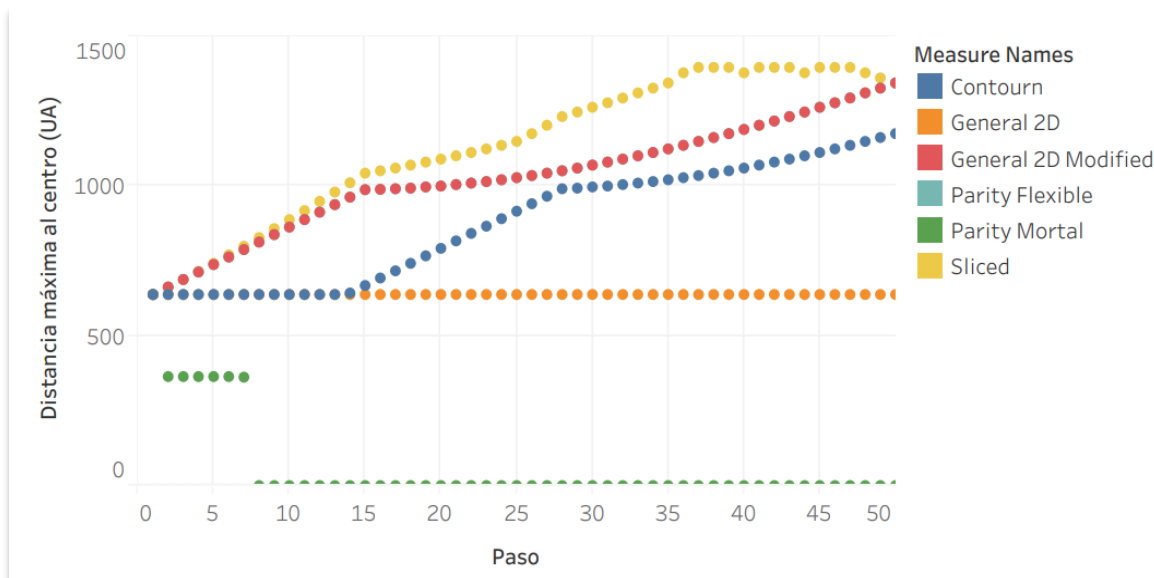


Gráfico 10: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón battle con las reglas *contourn*, *general_2D*, *general_2D_modified*, *parity_flexible*, *parity_mortal* y *sliced*

8.2.3 Gráfico 11 - Random

Para el onceavo gráfico se realizaron 10 corridas del patrón **random** con cada una de las reglas dispuestas en la columna de la derecha y se generó un promedio del resultado. Debido a que las estructuras se generan de manera aleatoria, por más que existan reglas restrictivas como *parity_mortal*, se mantienen resultados elevados de distancia máxima. Por otro lado, se puede ver que aquellas reglas que maximizan dicho valor son las de carácter expansivo.

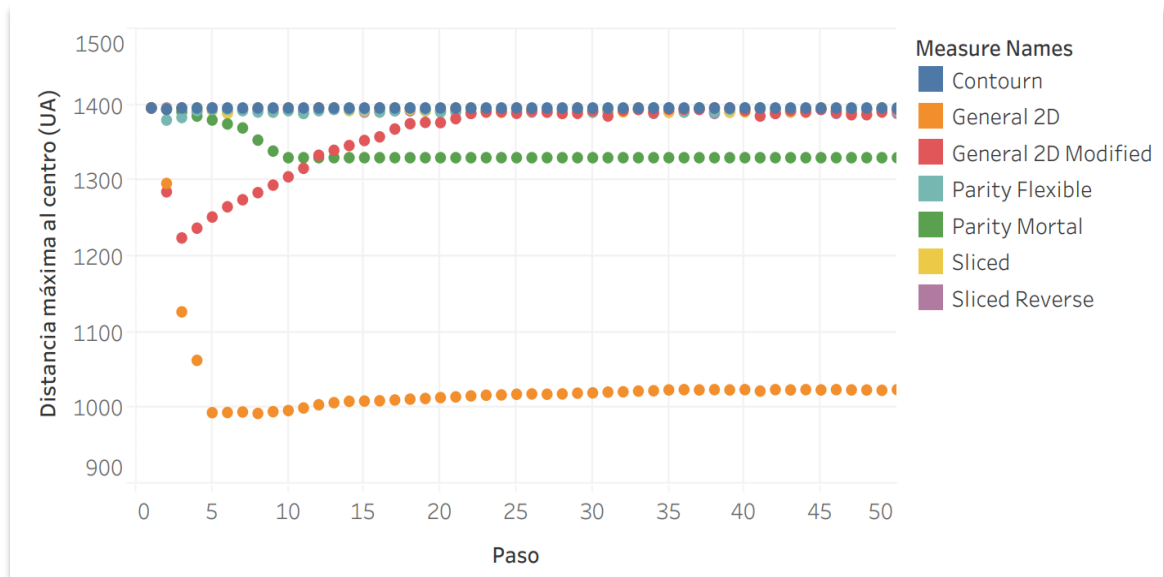


Gráfico 11: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón random con las reglas *contourn*, *general_2D*, *general_2D_modified*, *parity_flexible*, *parity_mortal*, *sliced* y *sliced_reverse*

8.2.4 Gráfico 12 - Cube

Para este caso se realizó una única corrida del patrón **cube** con cada una de las reglas dispuestas en la columna de la derecha. Se pudo observar que la mayoría de las reglas no generaron cambio alguno en el patrón a lo largo de su desarrollo, por lo que como consecuencia se obtuvo una distancia máxima constante. Se puede ver que *general_2d_modified* al afectar rotundamente la expansión de este patrón generó un cambio positivo en el aumento de la distancia máxima.

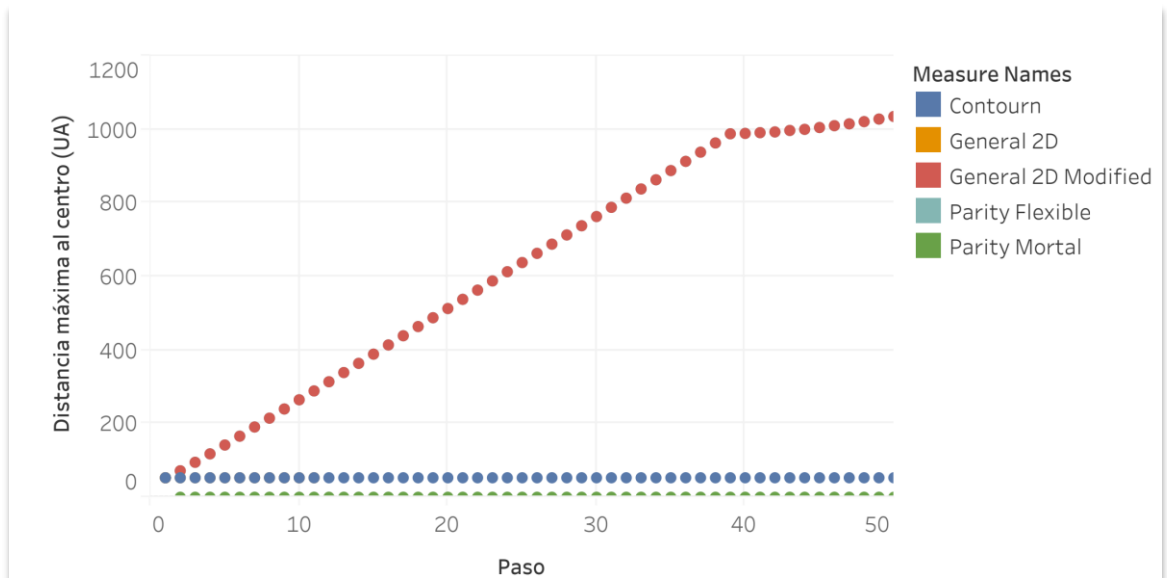


Gráfico 12: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón cube con las reglas contourn, general_2D, general_2D_modified, parity_flexible y parity_mortal

8.2.5 Gráfico 13 - Fatality

Para este gráfico se realizó una única corrida del patrón fatality con cada una de las reglas dispuestas en la columna de la derecha. Se puede observar aquellas reglas que maximizan el aumento de la distancia máxima es *general_2d* y *parity_flexible*, lo cual se debe a su carácter expansivo. Por otro lado, las reglas más restrictivas lo mantienen en valores relativamente bajos.

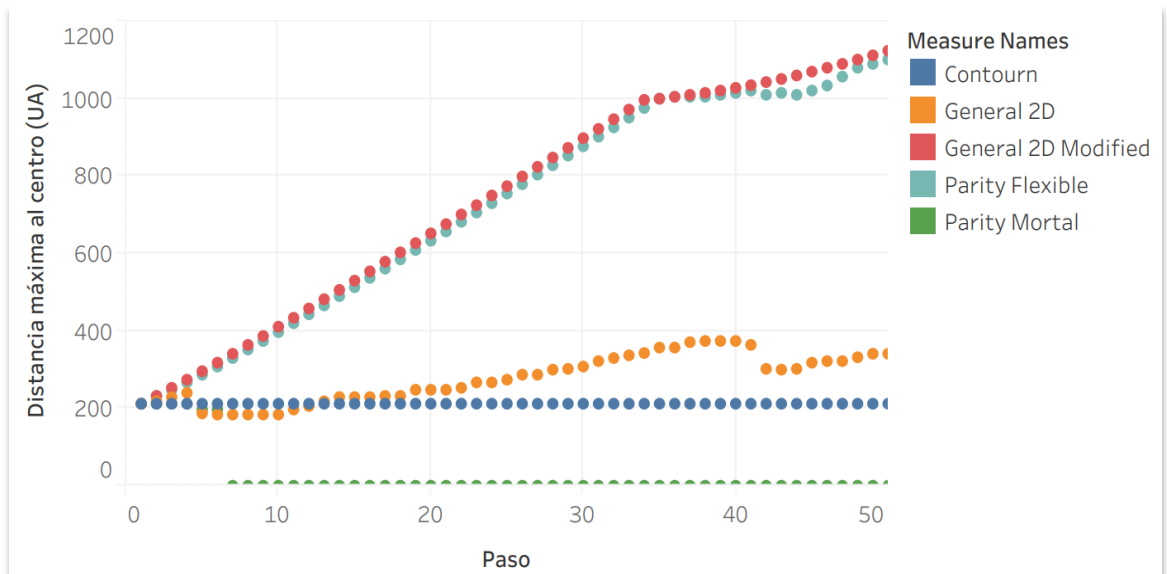


Gráfico 13: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón fatality con las reglas contourn, general_2D, general_2D_modified, parity_flexible y parity_mortal

8.2.6 Gráfico 14 - Oscilator

Para este caso se realizó una única corrida del patrón **oscilator** con cada una de las reglas dispuestas en la columna de la derecha. Se puede observar que, aquellas reglas que tienen un comportamiento menos restrictivo como *general_2d_modified*, *contourn* y *parity_flexible* al permitir el desarrollo del crecimiento del universo, afectan al aumento de la distancia máxima paso a paso. Mientras que las reglas que tienen un comportamiento más restrictivo como *general_2d* y *parity_mortal*, al degenerar el universo o mantener su población relativamente constante generan una distancia máxima baja o nula.

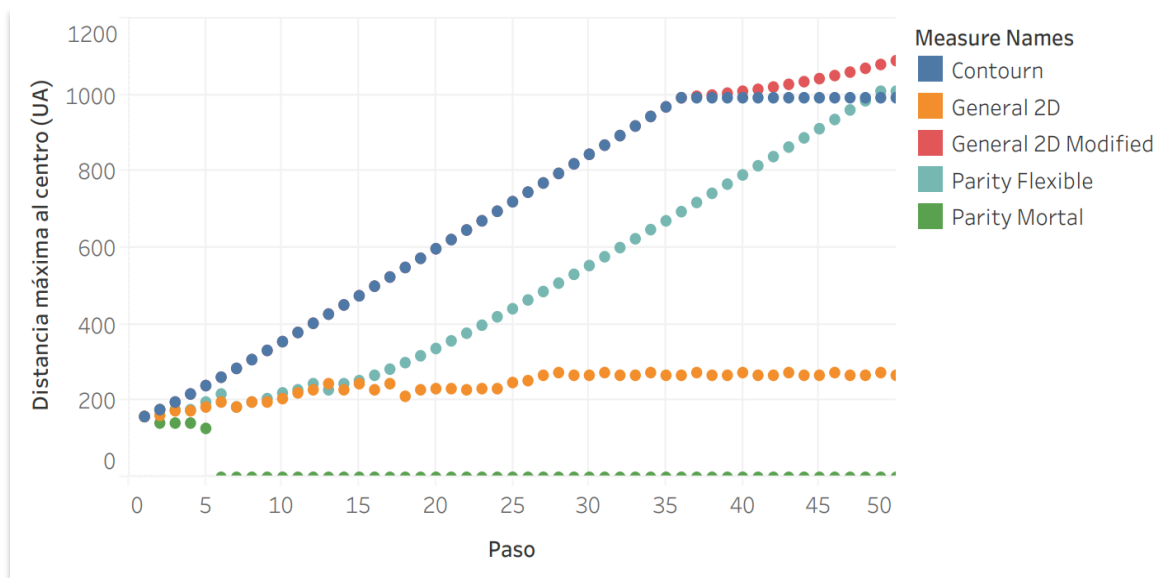


Gráfico 14: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón oscilator con las reglas *contourn*, *general_2D*, *general_2D_modified*, *parity_flexible* y *parity_mortal*

8.2.7 Conclusiones de los gráficos de distancia máxima

Como se pudo ver en los gráficos anteriores, existe una estrecha relación entre el aumento de la población en los patrones durante su desarrollo con el aumento de la distancia máxima obtenida. Al haber más población en el universo en cada paso es más factible que se obtenga una distancia máxima mayor.

Se pudo ver claramente que aquellas reglas con comportamientos restrictivos, al degenerar o destruir la población del universo, afectaron drásticamente a las distancias máximas obtenidas paso a paso dado que estas se reducían. Pero al utilizar reglas menos restrictivas se obtenía el resultado opuesto.

Lo dicho anteriormente se puede ver claramente con *general_2d* y *general_2d_modified*. Ambas reglas son muy parecidas, dado que *general_2d_modified* deriva de la otra. Sin embargo, al ser *general_2d_modified* menos restrictiva, permite un crecimiento mayor de la población que *general_2d*, y esto en consecuencia produce que en los gráficos se observe que con *general_2d_modified* se obtienen distancias máximas mayores.

Vemos que lo mismo sucede con *parity_flexible* y *parity_mortal* ya que uno tiene un carácter expansivo mientras que el otro tiene un carácter destructivo

Debido a los resultados obtenidos en los gráficos de distancia máxima, se decidió elegir el patrón más representativo para graficar la velocidad para cada regla. En el siguiente gráfico de velocidad podemos apreciar que las reglas *contourn* y *parity_flexible* son las que tienen las mayores velocidades a lo largo

del desarrollo del patrón. Por otro lado, *general_2d* presenta un comportamiento oscilatorio entre velocidades negativas y velocidades positivas.

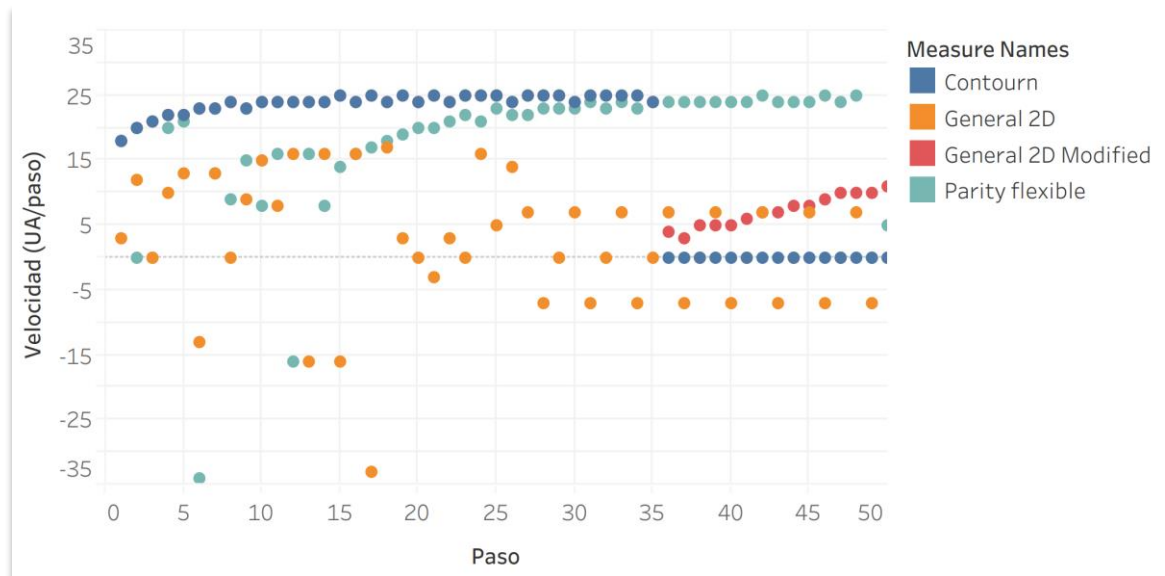


Gráfico 15: Velocidad de las partículas respecto del tiempo, medido en pasos, para el patrón oscilator con las reglas *contourn*, *general_2d*, *general_2d_modified* y *parity_flexible*.

9. Otros resultados y conclusiones

9.1 Resultados con patrón inicial constante y reglas distintas

Como vimos en la sección 7. *Reglas 2D* existen cuatro reglas principales y tres reglas que derivan de otras, es decir, que son modificaciones leves a una regla original. Se realizaron simulaciones con el mismo patrón inicial para estudiar cómo cambiaba el comportamiento del universo en función de las modificaciones impuestas a las reglas

9.1.1 Parity_flexible vs Parity_mortal

Como se pudo ver en los gráficos anteriores *parity_flexible* y *parity_mortal* son dos reglas muy parecidas pero que a la hora de realizar las simulaciones mostraron comportamientos completamente contrarios. Como su nombre lo indica, *parity_flexible* permite el desarrollo de universos crecientes de manera indefinida mientras que con *parity_mortal* se obtienen universos que se degeneran en el tiempo hasta que terminan destruyéndose completamente o queda una población de partículas vivas muy pequeña, en general menor a la inicial.

Por ello, podemos concluir que *parity_flexible* tiene un **comportamiento expansivo** mientras que *parity_mortal* tiene un **comportamiento destructivo**.

9.1.2 Sliced vs SlicedReverse

Como se pudo ver en los gráficos anteriores *sliced* y *sliced_reverse* son reglas simétricamente contrarias que mostraron un comportamiento muy interesante en las diferentes simulaciones realizadas.

En cada simulación, con la regla *sliced* se partió de un escenario inicial que luego se transformó en mareas que se mueven hacia la dirección izquierda.

sliced_reverse al ser una regla simétrica, mostró el mismo comportamiento de las mareas, pero en dirección opuesta.

9.1.3 General_2d vs General_2d_modified

Como se pudo ver en los gráficos anteriores ambas reglas presentan un comportamiento similar en cuanto a las estadísticas de supervivencia, reproducción y mortalidad. Se pudo observar que *general_2d_modified* presentó un incremento en la tasa de supervivencia y reproducción y un decremento en la tasa de mortalidad.

Por otro lado, se encontró un patrón inicial muy interesante en el que se mostró un comportamiento completamente diferente para ambas reglas. Al usar el patrón inicial **cube** y la regla *general_2d*, el universo se mantenía estático sin mostrar ningún crecimiento o decrecimiento, mientras que al usar *general_2d_modified* sucedía todo lo contrario; se obtuvo un universo creciente indefinidamente que generaba un patrón interesante y completamente simétrico a lo largo del desarrollo del universo.

9.2 Resultados con regla constante y patrón inicial distinto

Para constatar que la evolución del juego no depende únicamente de la regla aplicada sino del patrón utilizado se realizaron patrones levemente modificados como vimos en la sección 6. *Patrones iniciales 2D*. Se realizaron simulaciones con la misma regla, cambiando el patrón inicial para estudiar cómo cambiaba el comportamiento del universo.

9.2.1 Gráfico 16 y Gráfico 17 - Cube vs Cube2

Para los Gráficos 16 y 17 se realizó una única corrida de cada patrón inicial con la regla *general_2d*. Como vimos antes, al aplicar esta regla a **cube**, la población permanecía constante, pero si se lo aplicamos a **cube2**, la población crece. Además, se expande a lo largo del universo porque podemos observar en el Gráfico 17 que la distancia de la partícula más lejana al centro aumentó. Por ende, muchas veces podemos cambiar levemente el patrón y el comportamiento de la regla es totalmente diferente.

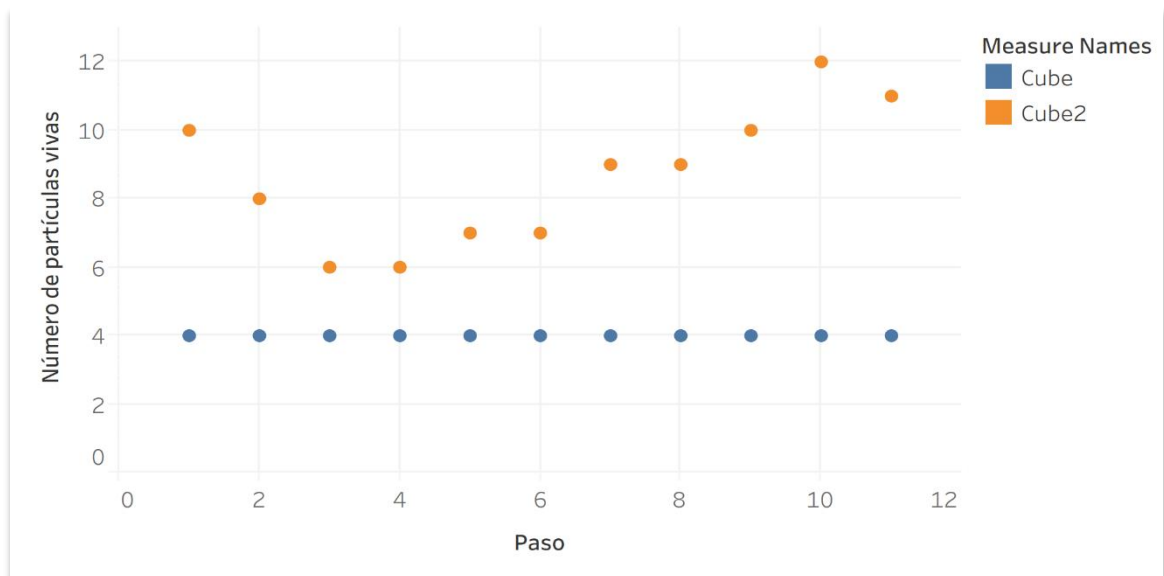


Gráfico 16: Evolución del número de partículas vivas en el tiempo, medido en pasos, para los patrones *cube* y *cube2*, con la regla *general_2D*

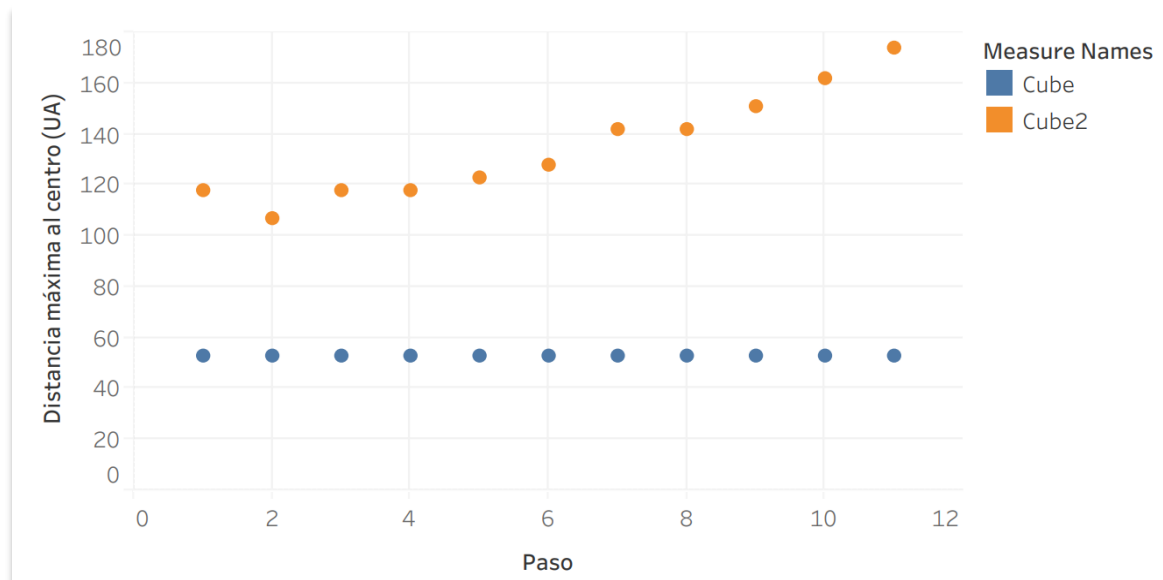


Gráfico 17: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para los patrones *cube* y *cube2*, con la regla *general_2D*

10. Patrones iniciales 3D

Para la implementación en 3 dimensiones, se definieron 5 patrones iniciales que se detallarán a continuación:

10.1 Random

Patrón inicial que llamamos *random* debido a que se genera una configuración inicial de partículas vivas de forma aleatoria. Para cada partícula, se genera un número aleatorio que, si es menor a 0.005, entonces determina que la partícula está viva.

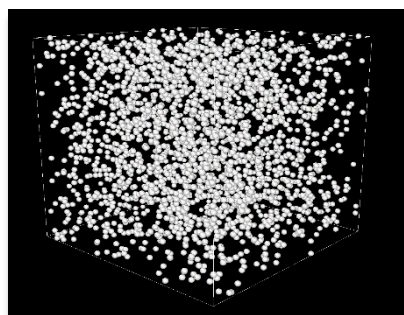


Figura 16: Patrón Random

10.2 Linear

Patrón inicial que llamamos *linear* por estar conformado por tres líneas verticales paralelas entre sí ubicadas a una distancia dada por la ecuación (1)

$$D = \text{particulas_por_fila} / 20$$

donde D es la distancia entre las líneas y $\text{particulas_por_fila}$ es la cantidad de partículas que existen en cada fila del tablero.

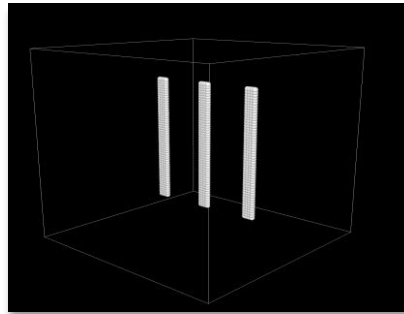


Figura 17: Patrón Linear

10.3 Cube

Patrón inicial que llamamos *cube* debido a que, si la cantidad de partículas totales en columna, fila y altura son iguales, entonces forman un cubo.

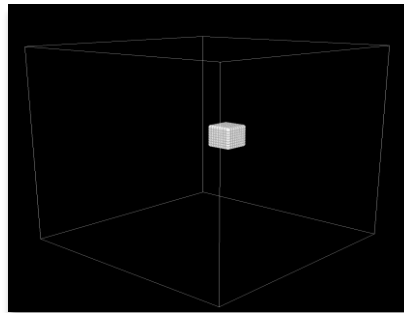


Figura 18: Patrón Cube

10.4 Border

Patrón inicial que llamamos *border* debido a que está formado mayormente por bordes del tablero tridimensional total cubierto.

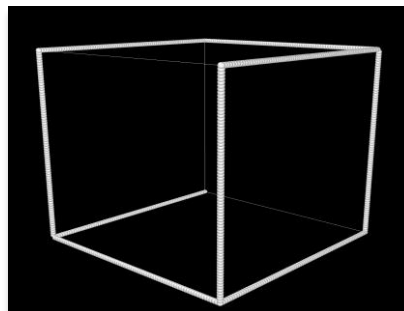


Figura 19: Patrón Border

10.5 Glider

Patrón inicial que llamamos *glider* debido a que es una estructura que oscila como lo muestra la *Figura X* si es usada con la regla 5655 que veremos en la sección 11. *Rule_WXYZ* donde se explicita el formato general de cada una de estas reglas.

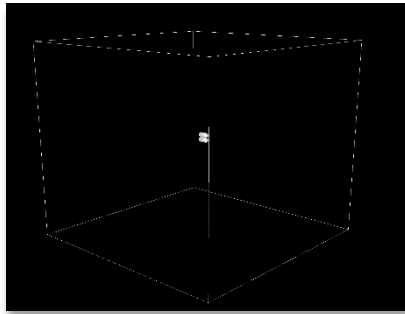
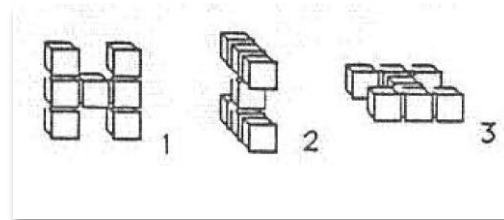


Figura 20: Patrón Glider



11. Reglas 3D

Para la implementación en 3 dimensiones, se definieron 2 reglas principales y 5 reglas derivadas que se detallarán a continuación:

11.1 General 3D

Regla que llamamos *general_3d* debido a que es la regla principal que se usó en las primeras simulaciones en 3 dimensiones. La regla se define de la siguiente forma:

- Si la partícula tiene entre 6 y 9 partículas vecinas vivas, entonces en el siguiente paso vive independientemente si estaba viva o muerta
- Si la partícula tiene menos de 6 o más de 9 partículas vecinas vivas entonces muere

11.2 Rule_WXYZ

Regla que llamamos *rule_WXYZ* debido a que es una regla que conseguimos a partir de la bibliografía [1]. La regla se define de la siguiente forma:

- Una partícula viva sobrevivirá al siguiente paso si y sólo si se cumple con la relación (1) detallada a continuación:

$$W \leq Pv \leq Z \quad (1)$$

donde W y X son constantes predefinidas en la regla y Pv es la cantidad de partículas vecinas vivas.

- Una partícula muerta vivirá al siguiente paso si y sólo si se cumple con la relación (2) detallada a continuación:

$$Y \leq Pv \leq Z \quad (2)$$

donde Y y Z son constantes predefinidas en la regla y Pv es la cantidad de partículas vecinas vivas.

- En cualquier otro caso la partícula muere

Entonces si se crea la regla *rule_2346* por ejemplo se tiene que:

- Una partícula viva sobrevivirá al siguiente paso si y sólo si tiene exactamente entre 2 y 3 partículas vecinas vivas.
- Una partícula muerta vivirá en el siguiente paso si y sólo si tiene exactamente en 4 y 6 partículas vivas.
- En cualquier otro caso la partícula muere.

11.3 Reglas derivadas

A partir de la regla *rule_WXYZ* detallada anteriormente se crearon múltiples reglas que, según la bibliografía consultada, son reglas idóneas para la formación de posibles universos que muestren un comportamiento el cual posibilita su estudio para obtener conclusiones al respecto.

12.Resultados 3D

12.1 Número de partículas vivas

En los siguientes gráficos se realizó una medición de la cantidad de población a lo largo de cada transición, cambiando el patrón inicial utilizado.

12.1.1 Gráfico 18 - Cube

Para este gráfico se realizó una única corrida del patrón **cube** con cada una de las reglas dispuestas en la columna de la derecha. Se puede observar que únicamente dos reglas, *rule_4526* y *rule_2333* responden positivamente a la evolución de la población mientras que las demás reglas la degeneran hasta eliminarlas mostrando la dualidad del comportamiento destructivo y expansivo entre las reglas.

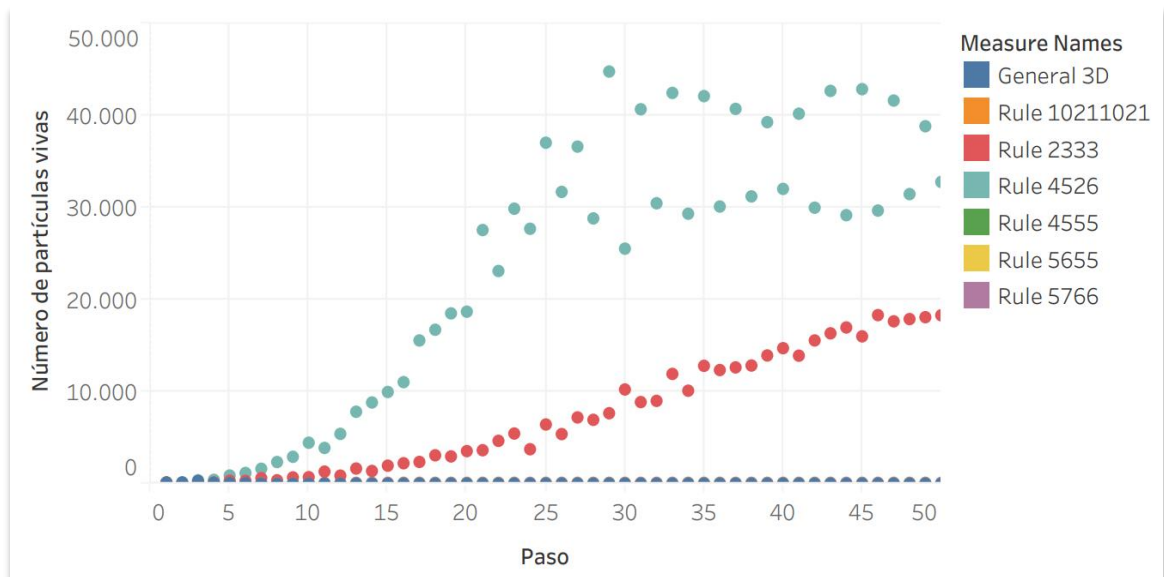


Gráfico 18: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón **cube** con las reglas *general_3d*, *rule_10211021*, *rule_2333*, *rule_4526*, *rule_4555*, *rule_5655* y *rule_5766*

12.1.2 Gráfico 19 - Glider

En este caso se realizó una única corrida del patrón **glider** con cada una de las reglas dispuestas en la columna de la derecha. Se puede observar que solo las reglas *rule_2333* y *rule_4526* afectan de manera positiva a la población mientras que las demás la destruyen en pocas transiciones reafirmando la dualidad anteriormente descrita.

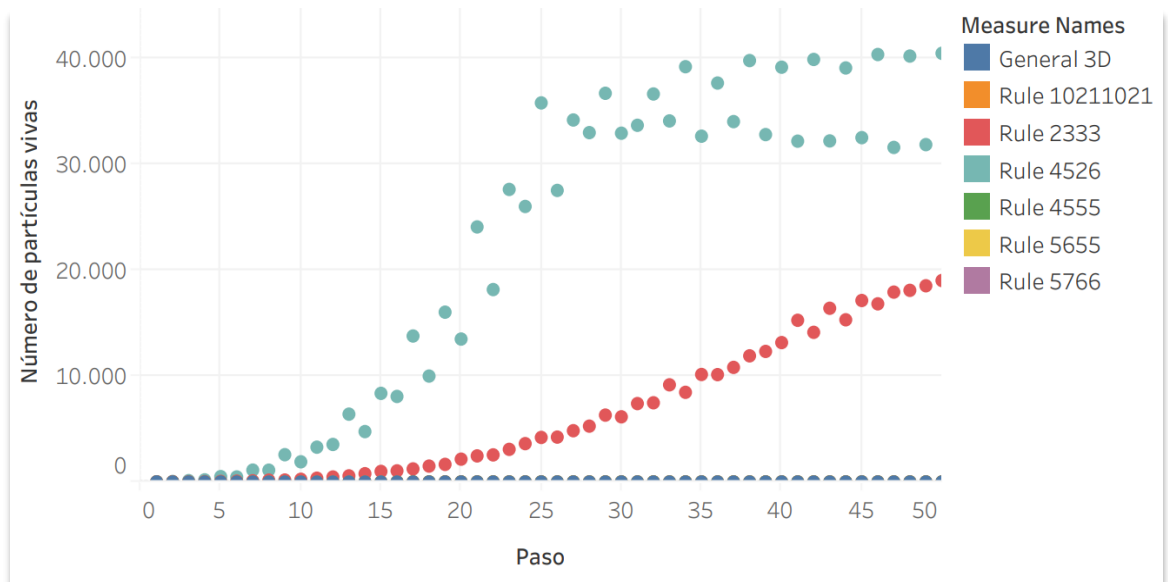


Gráfico 19: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón glider con las reglas general_3d, rule_10211021, rule_2333, rule_4526, rule_4555, rule_5655 y rule_5766

12.1.3 Gráfico 20 - Border

Para el gráfico 20 se realizó una única corrida del patrón **border** con cada una de las reglas dispuestas en la columna de la derecha. Se muestran resultados parecidos a los gráficos anteriores con la particularidad que la regla *rule_4526* además de mostrar un comportamiento expansivo muestra un comportamiento oscilatorio durante la expansión de la población mientras que *rule_2333* muestra un carácter expansivo relativamente constante.

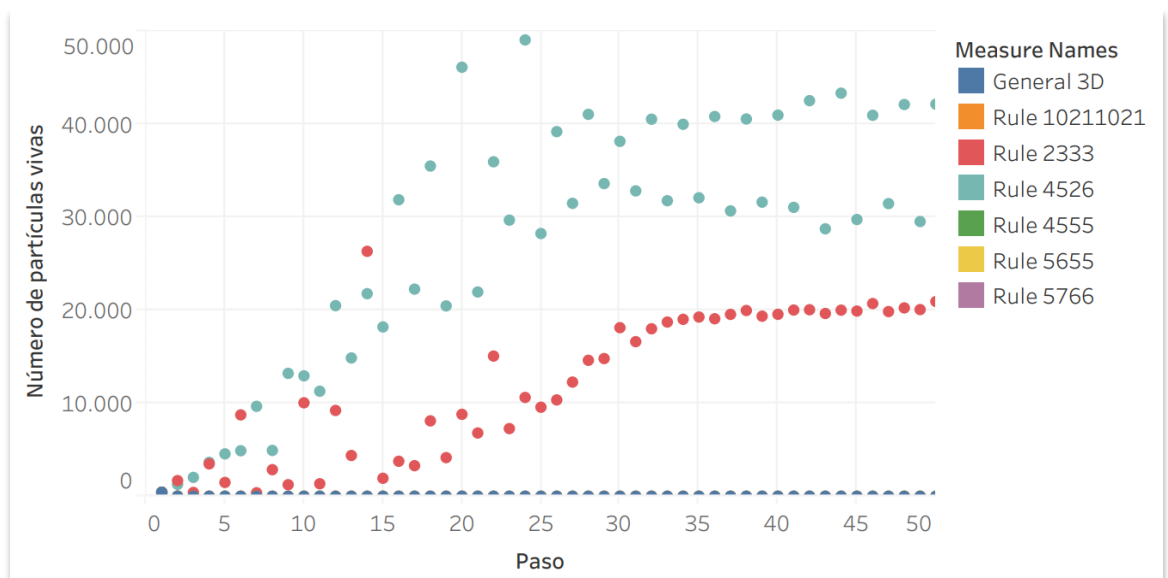


Gráfico 20: Evolución del número de partículas vivas en el tiempo, medido en pasos, para el patrón border con las reglas general_3d, rule_10211021, rule_2333, rule_4526, rule_4555, rule_5655 y rule_5766

12.1.4 Conclusiones de los gráficos de población

Como se pudo observar en los gráficos, existe una clara diferencia entre ciertas reglas. Por un lado, la regla *rule_4526* y la regla *rule_2333* son de carácter expansivo. Sin embargo, la regla *rule_2333*

presenta un crecimiento relativamente uniforme de la población debido a que los números que presentan los límites de dicha regla son muy próximos. Por otro lado, la regla *rule_4526* al presentar límites más dispersos afecta directamente al crecimiento de la población, la cual creció de manera oscilatoria.

Las demás reglas demostraron tener un carácter más destructivo en los escenarios propuestos ya que se vio que las poblaciones se degeneraron violentamente hasta ser eliminadas.

12.2 Distancia máxima

12.2.1 Gráfico 21 - Glider

Para el gráfico 21 se realizó una única corrida del patrón **glider** con cada una de las reglas dispuestas en la columna de la derecha. Se muestra que las mismas dos reglas de carácter expansivo afectan de manera positiva a las distancias máximas obtenidas, pero en este caso la regla *rule_4526* no presenta ese carácter oscilatorio que se presencié anteriormente. Además, se observa que la regla *rule_5655* muestra una distancia máxima relativamente constante a lo largo de las transiciones.

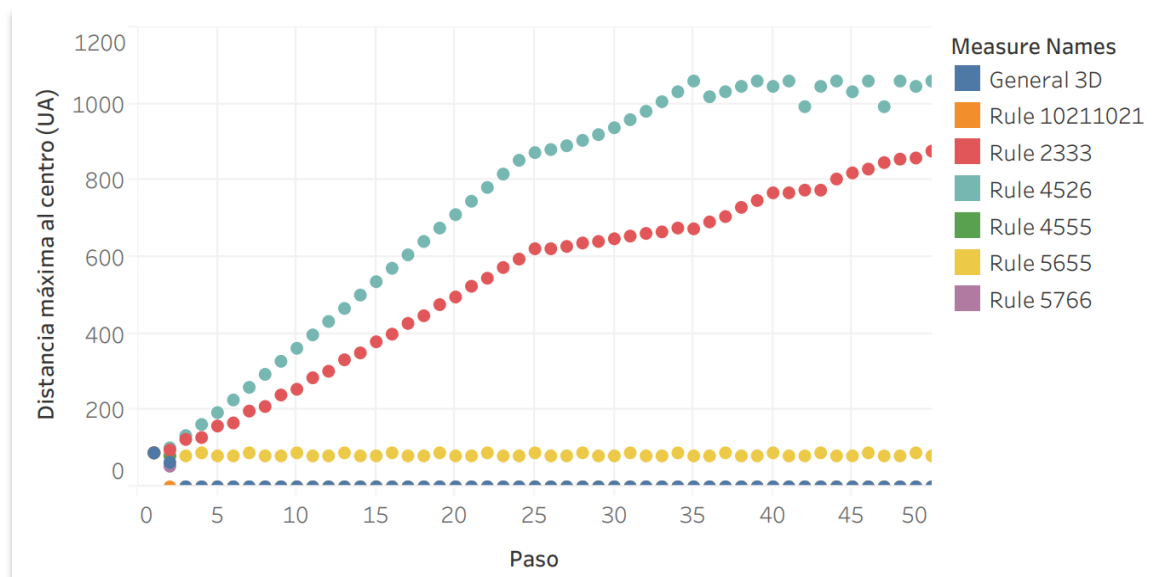


Gráfico 21: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón glider con las reglas *general_3d*, *rule_10211021*, *rule_2333*, *rule_4526*, *rule_4555*, *rule_5655* y *rule_5766*

12.2.2 Gráfico 22 - Cube

Para el gráfico 22 se realizó una única corrida del patrón **cube** con cada una de las reglas dispuestas en la columna de la derecha. Como se puede ver en el gráfico, las únicas dos reglas que afectan positivamente a las distancias máximas son las mismas de carácter expansivo. En adición, la regla *rule_5766* parece presentar un comportamiento constante en cuanto a la distancia máxima para este patrón en específico.

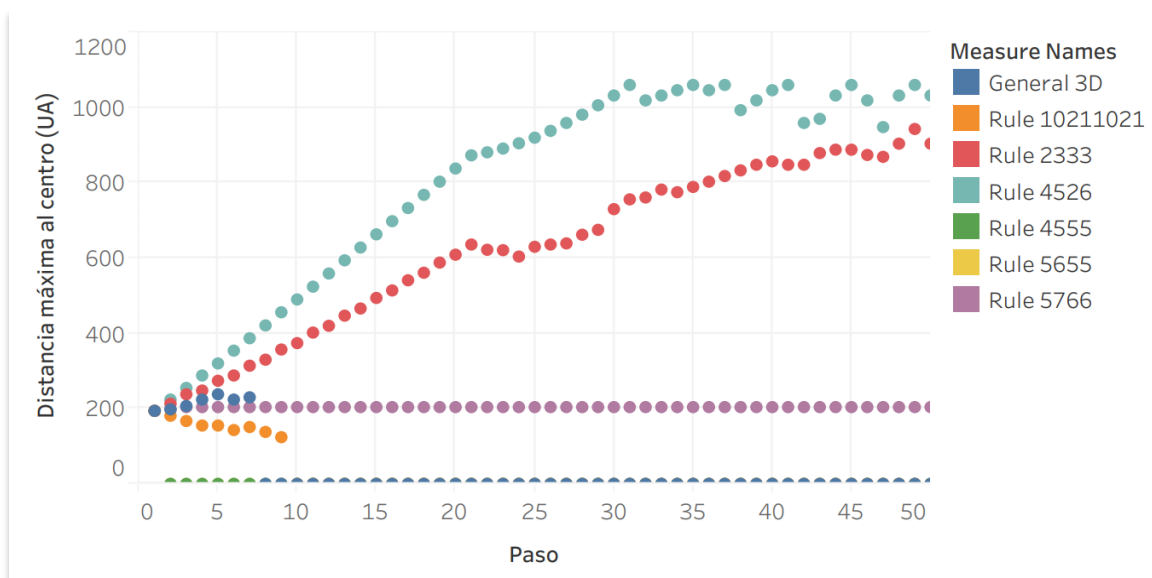


Gráfico 22: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón cube con las reglas *general_3d*, *rule_10211021*, *rule_2333*, *rule_4526*, *rule_4555*, *rule_5655* y *rule_5766*

12.2.3 Gráfico 23 - Border

Para el décimo gráfico se realizó una única corrida del patrón **border** con cada una de las reglas dispuestas en la columna de la derecha. Dada la naturaleza del patrón, se puede ver que las reglas de carácter expansivo muestran un comportamiento oscilatorio para este escenario, mientras que las demás reglas mantienen valores nulos para las distancias máximas.

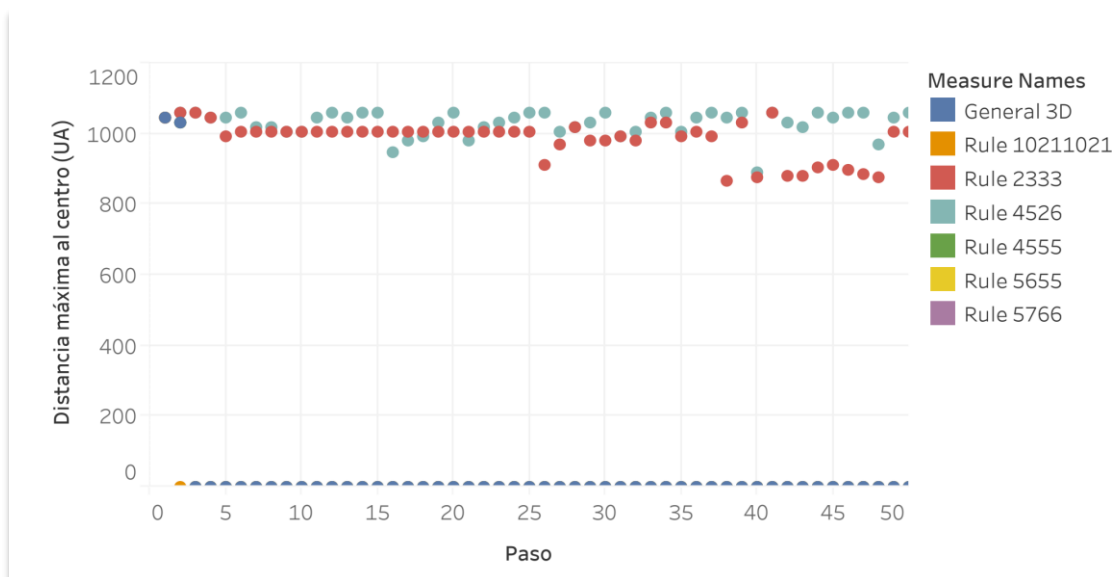


Gráfico 23: Evolución de la distancia máxima al centro, medida en unidades arbitrarias, en el tiempo, medido en pasos, para el patrón border con las reglas *general_3d*, *rule_10211021*, *rule_2333*, *rule_4526*, *rule_4555*, *rule_5655* y *rule_5766*

12.2.7 Conclusiones de los gráficos de distancia máxima

Como concluimos en la sección 8.2.7 *Conclusiones de los gráficos de distancia máxima* se ve una clara relación entre la cantidad de población, la distancia máxima y el carácter de la regla utilizada.

Con ambas dimensiones analizadas y luego de observar todos los gráficos podemos concluir que en la mayoría de los casos una regla de carácter expansivo hará aumentar la población en el tiempo lo cual, como consecuencia, afectará a las distancias máximas que se obtengan. Por otro lado, reglas de carácter destructivo degenerarán o destruirán la población por lo que se obtendrán valores bajos o nulos de distancia máxima.

13. Conclusiones generales

Con los resultados obtenidos, en adición a la información provista por la bibliografía, se puede ver que es más sencillo generar reglas en 2D que puedan generar universos estudiables y estables o que al menos generen una cierta simetría durante su desarrollo.

Debido a las simulaciones en las que se generaban leves modificaciones, ya sea en la regla utilizada o en las condiciones iniciales, se puede ver que esos dos factores son los más importantes para el desarrollo del universo, ya que afectan directamente a las tasas de supervivencia, reproducción y mortalidad, así como también modifican la rapidez con la que el “radio” del universo crece paso a paso en la simulación.

Como última conclusión, se pudo observar que las reglas de carácter expansivo permitieron un estudio más amplio en las estadísticas de los diferentes escenarios mientras que las de carácter destructivo degeneraban demasiado rápido la población.

14. Bibliografía

A continuación, se detalla la bibliografía utilizada:

- <https://playgameoflife.com>
- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
- [1] Candidates for the Game of Life in Three Dimensions - Carter Bays
- Bibliografía de la materia (clases teóricas)