

# Termodinámica de un sistema dinámico discreto basado en el juego de la vida de Conway en 3 dimensiones

Universidad Nacional de Colombia

Departamento de Física

Mecánica Estadística

Carlos Esteban Gutierrez Guarnizo, Jhon Sebastian Moreno Triana

Daniel Felipe Rojas Sanabria

*cagutierrezgu@unal.edu.co, jmorenotr@unal.edu.co, drojassa@unal.edu.co*

2 de Agosto, 2021

## Resumen

Los sistemas físicos compuestos de muchas partículas o elementos suelen estar fuera del equilibrio, y conocer su comportamiento resulta una tarea complicada debido a su alta no linealidad; sin embargo, el tipo de arreglos de 3 dimensiones que estudiamos son finitos, así que tras esperar el suficiente tiempo el sistema llega al equilibrio

Se plantea una variante del modelo de “el Juego de la Vida” descrito por Conway donde se propone analizar el comportamiento del sistema con un grado de libertad adicional (3D) considerando un arreglo cúbico discreto, para lo cual se modifican las reglas usando como tentativa R(5766). El proceso se calculan las variables termodinámicas definidas para el sistema con condiciones iniciales. Se hace un contraste experimental con ayuda de simulaciones y por ultimo se comparan los resultados obtenidos con los del sistema original de Conway en 2D.

## Introducción

En este proyecto analizamos la mecánica estadística de un arreglo dinámico y discreto de 3 dimensiones, donde su dinámica se rige por pasos discretos de tiempo y bajo unas reglas de juego basadas en el Juego de la Vida de Conway, específicamente la regla R(5766) [2].

Al hablar del juego de la vida de Conway es inevitable relacionarlo con el concepto de autómata celular, el cual es aún más antiguo que el trabajo de Conway. En la década de 1950, John Von Neumann introdujo este término en su libro *Theory of Self-reproducing Automata*, describiendo la dinámica de un sistema de celdas cuyos estados posibles son vi-

vas o muertas y dependen de condiciones del instante anterior. Sin embargo, fue a partir de que Conway expuso su sistema que generaba patrones periódicos con la regla R(2333)[2], que se despertó el interés en los autómatas celulares.

Además de generar patrones periódicos en el plano de estudio, esta regla solía llevar al sistema a un estado estable, lo que llevo a realizar más estudios de sistemas de este tipo, por ejemplo modelos termodinámicos a partir de la mecánica estadística [3], el cual se tratará en el presente artículo.

Para estudiar un sistema dinámico y discreto como el del juego de la vida de Conway es necesario contar con la descripción del sistema, sus reglas y las bases teóricas correspondientes para hacer un análisis de la estructura en estudio. Es por esto que expondremos las reglas de juego en un sistema de 2 dimensiones, la más conocida es la propuesta por Conway que se conoce como R(2333), compuesta por las siguientes condiciones:

1. Si una celda viva tiene 2 o 3 celdas vecinas vivas esta seguirá viva, si tiene menos de 2 morirá por soledad y si tiene más de 3 morirá por sobre población.
2. Si una celda muerta tiene 3 celdas vecinas vivas esta revivirá.

Al extender el sistema a 3 dimensiones la regla a usar suele ser la R(5766) que de manera similar a la R(2333), para que una celda viva continúe viva es necesario que tenga de 5 a 7 vecinas vivas y para que una celda reviva necesita 6 celdas vecinas vivas. A continuación definimos la densidad de celdas vivas, que no es más que la medida de celdas vivas en un instante de tiempo comparado con el total de celdas del arreglo. A partir de esta definición y las reglas

de juego es posible predecir la densidad de celdas vivas un instante de tiempo posterior como lo dice la siguiente ecuación [3]:

$$\sigma_{\alpha}(t+1) = \delta(6, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) + \sigma_{\alpha}(t)(\delta(5, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) + \delta(7, \sum_{\alpha'=1}^{26} \sigma_{\alpha'})) \quad (1)$$

Además, haciendo uso de la teoría de campo medio fue posible realizar aproximaciones para un tratamiento global del arreglo y conocer la probabilidad de encontrar  $h$  celdas vivas en este sistema, a partir de la cual puede calcularse la densidad de celdas vivas cuando el sistema llega al equilibrio. Se notó que la probabilidad mencionada es igual para todas las celdas, de donde puede encontrarse el número total de estados  $W$  y por lo tanto usar el ensamble microcanónico para modelar el arreglo estudiado. A partir de este ensamble fue posible analizar termodinámicamente el sistema y encontrar expresiones para la temperatura, energía interna y entropía.

El breve desarrollo expuesto anteriormente a modo de resumen, y de manera extendida en el marco teórico, se hizo con la intención de cumplir con nuestro principal objetivo en este artículo que es hallar de manera analítica y computacional la densidad de celdas vivas y variables termodinámicas de un sistema dinámico y discreto de 3 dimensiones cuando llega al equilibrio, estas últimas calculadas bajo el ensamble microcanónico de la mecánica estadística.

Para cumplir el objetivo anterior, se extendió el formalismo mecánico-estadístico hecho por Seiden y Schulman en 2 dimensiones [3], lo cual se realizó en el marco teórico. Así que con las definiciones de las variables termodinámicas extendidas para 3 dimensiones se procedió a la verificación computacional para contrastarlo con el procedimiento analítico y las aproximaciones en este. Se usaron configuraciones iniciales aleatorias así como la variación de parámetros como el tamaño del sistema y la densidad inicial de celdas vivas, para luego calcular las respectivas variables termodinámicas y su posterior análisis. Para las simulaciones se usarán lenguajes de programación y software JavaScript, Python y Octave.

Asimismo, los resultados obtenidos analítica y computacionalmente se comparan con los que se obtendrían en un arreglo en 2 dimensiones, donde las reglas de juego son diferentes y existe un grado de libertad menos. Todo lo anterior puede considerarse como una aproximación a sistemas físicos complejos fuera del equilibrio y la manera en la que llegan a este, como el estudio de gases y líquidos; no obstante, estudiar el Juego de la Vida también tiene motivaciones biológicas en el estudio de la reproducción y evolución

de las especies [3], y en el estudio del comportamiento mismo de las sociedades a grandes escalas.

Tras desarrollar analíticamente el planteamiento del problema, los resultados esperados de la simulación computacional se muestran en el Cuadro 2, que presenta valores de número de partículas, densidad, temperatura, energía interna y entropía que esperan obtenerse luego de hacer el análisis correspondiente. Con los resultados de la simulación es posible analizar el comportamiento de la densidad en función del tiempo y de la densidad de estabilidad final en función de la densidad inicial del sistema, y se esperan obtener gráficas similares a las presentadas en la Figura 1 y Figura 2.

Todos los detalles de lo anteriormente descrito se encuentran en la sección del marco teórico donde se realizó el procedimiento analítico necesario para resolver el problema planteado en este informe, la sección del estado del arte donde se brinda la evolución temporal del problema en cuestión y sus orígenes, la sección del planteamiento del problema tentativo en la que se expone a profundidad el problema a resolver y en los trabajos en los cuales está inspirado este, la sección de motivación y justificación que da una perspectiva del porqué se quiere resolver el problema tentativo en términos académicos y sus posibles aplicaciones, las secciones de objetivos general y específicos, la sección de metodología que da los detalles del cumplimiento de los objetivos específicos, la sección de resultados esperados que brinda una idea de los resultados que se pretenden obtener teniendo en cuenta trabajos relacionados a este. Posterior a las secciones anteriores se encuentra lo relacionado con los resultados de la simulación y el desarrollo analítico, el análisis de estos y las conclusiones del estudio.

## Marco Teórico

### Juego de la vida de Conway

Este juego ha sido creado por el matemático británico John Horton Conway en 1970. Este es un juego de cero jugadores, es decir el juego solo necesita una una entrada inicial para ver como esta evoluciona en el tiempo.[1]

Este juego consiste en un arreglo, inicialmente en 2 dimensiones, con celdas que pueden estar entre dos estados “Vivas” o “Muertas” según las “reglas de vida”.

### Reglas del juego en 2 dimensiones

Para formalizar las reglas del juego es necesario definir un entorno  $E$  como el número de vecinos ne-

cesarios de cada celda viva para que esta pueda seguir viviendo, esto es  $E_{min} \leq E \leq E_{max}$ . Y la fertilidad  $F$  como el número de vecinos necesarios para que una celda que no este viva, reviva, es decir  $F_{min} \leq F \leq F_{max}$ . Definiendo la regla de transición  $R$  como una 4-tupla  $R(E_{min}, E_{max}, F_{min}, F_{max})$ . Para el caso del juego de la vida de Conway se tiene  $R(2333)$ . [2] Es decir:

1. Si una celda viva cuenta con 2 o 3 celdas vecinas vivas esta continuará viva, si tiene menos de 2 morirá por soledad, mientras que si tiene más de 3 morirá por sobre población.
2. Si una celda que ya está muerta tiene 3 celdas vecinas vivas esta revivirá.

En el caso bidimensional existen también otras configuraciones, una de ellas es la  $R(3434)$ , llamada también una “3-4” vida.

Sin embargo algunas de estas reglas pueden llevar a que el sistema nunca se estabilice o que incluso todo el sistema se aniquile de manera segura para cualquier condición inicial, con lo cual el matemático Alexander Dewdney definió el siguiente par de condiciones para que una regla pueda ser llamada una vida.

1. Una celda debe existir o debe ocurrir “naturalmente” si se aplica la regla  $R(E_{min}, E_{max}, F_{min}, F_{max})$  a un ordenamiento inicial y arbitrario de celdas vivas y muertas.
2. Todas las configuraciones iniciales, sujetas a  $R(E_{min}, E_{max}, F_{min}, F_{max})$ , deben tener una cota superior en su crecimiento.

Lo cual permite iniciar una búsqueda por unas reglas válidas en el caso tridimensional. [2]

### Reglas de juego para el caso tridimensional

En el caso del juego de Conway en 2 dimensiones se tiene un grado  $k = 8$ , esto es que cada celda tiene 8 vecinos al rededor. Pero ahora se tiene un grado  $k = 26$ , con lo cual la regla  $R(2333)$  que se usaba antes no cumple las condiciones antes mencionadas.

Es posible mostrar que  $4 < F < 10$  son los únicos valores válidos para que la fertilidad cumpla con ser una vida. Encontrando así que la fertilidad está acotada.

De todas las opciones investigadas solo  $R(4555)$  y  $R(5766)$  cumplen con la primera condición, estas vidas son denotadas como “vida 4555” y “vida 5766”.

La regla 5766 permite encontrar formas estables y oscilantes, de alguna forma similares a la regla 2333 en el caso bidimensional, es por ello que esta regla es la más prolífica y usada. Además, la regla 4555 toma más tiempo para estabilizarse.

### Sistemas dinámicos discretos y uso del ensamble microcanónico

Los sistemas dinámicos discretos se caracterizan porque las variables que los definen avanzan en periodos discretos de tiempo. Esto es  $x_{n+1} = f(x_n)$ , lo cual pasa en el juego de la vida de Conway.

Es posible hacer un análisis del valor de la densidad de celdas vivas en todo el sistema, esto gracias a definir las siguientes variables según la regla  $R(5766)$ :

$$\sigma_\alpha(t+1) = \delta(6, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) + \sigma_\alpha(t) (\delta(5, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) + \delta(7, \sum_{\alpha'=1}^{26} \sigma_{\alpha'})) \quad (2)$$

donde cada  $\sigma_\alpha(t)$  toma el valor de 1 si la celda  $\alpha$  está viva y 0 si la celda está muerta. Además, cada  $\sigma_{\alpha'}$  representa la celda  $\alpha'$  vecina a la celda  $\alpha$ . Esta es otra manera de expresar la regla de vida  $R(5766)$ .

Por tanto se tiene que:

$$\rho(t) = \langle \sigma_\alpha(t) \rangle \quad (3)$$

Usando la aproximación de campo medio (“mean-field approximation”), la cual permite aproximar las interacciones entre cada una de las celdas, como el promedio de interacciones de las celdas aledañas, para el calculo de la densidad, es posible calcular la probabilidad ignorando las condiciones espaciales, teniendo  $P(h, \sigma_\alpha)$  la probabilidad de tener  $h$  celdas vecinas vivas para una celda en estado  $\sigma_\alpha(t)$ .

La probabilidad de encontrar una celda viva viene dada por  $P_t(\sigma_\alpha = 1) = \rho(t)$  y de encontrar una celda en el estado contrario sería  $P_t(\sigma_\alpha = 0) = 1 - \rho(t)$ , ignorando las dependencias de las celdas aledañas. La probabilidad de encontrar  $h$  celdas vivas sigue una distribución binomial.

$$P(h, t) = \binom{26}{h} \rho^h (1 - \rho)^{26-h} \quad (4)$$

Por tanto la probabilidad total en un tiempo  $t$ , viene dada por:

$$P_t(\sigma_\alpha, h) = (\sigma_\alpha \rho(t) + (1 - \sigma_\alpha)(1 - \rho(t))) P(h, t) \quad (5)$$

Calculando el valor esperado de  $\sigma_\alpha(t)$  a partir de las ecuaciones (4) y (20), para cuándo el sistema se

vuelve estable (es decir  $\rho(t) = \rho(t+1)$ ), se obtiene que:

$$\langle \sigma_\alpha(t+1) \rangle = \binom{26}{6} \rho^6 (1-\rho)^{20} + \binom{26}{5} \rho^6 (1-\rho)^{21} + \binom{26}{7} \rho^8 (1-\rho)^{19} = \rho \quad (6)$$

Resolviendo la anterior ecuación se tienen los valores de densidad  $\rho_1 = 0$ ,  $\rho_2 \approx 0,162$  y  $\rho_3 \approx 0,256$ , donde el valor de  $\rho_3$  es concordante con el valor de densidad encontrado en varias simulaciones en el juego de Conway en dos dimensiones ( $\rho_{2D} \approx 0,37$ ).[4]

Dado que la probabilidad para cada una de las celdas en estados  $\sigma_i$  distinta son constantes, si se tienen  $h$  celdas vivas y  $26-h$  celdas muertas alrededor de una celda en estado  $\sigma_\alpha(t)$ , se puede encontrar  $W$  el número total de estados en la vecindad, obteniendo el siguiente resultado:

$$W = \frac{26!}{h!(26-h)!} \quad (7)$$

Por lo tanto es posible usar el ensamble microcanónico sobre el sistema para obtener información, como por ejemplo la entropía. Por tanto, usando resultados del ensamble microcanónico se obtiene el siguiente resultado:

$$\begin{cases} h = 26\rho(t) \\ (26-h) = 26(1-\rho(t)) \end{cases} \quad (8)$$

$$\therefore W = \frac{26!}{(26\rho(t))!(26(1-\rho(t)))!}$$

Y por tanto se puede encontrar la entropía del sistema, tomando  $\rho(t)$  como  $\rho$  para simplificar la escritura.

$$\frac{S}{k_B} = \ln(W) \quad (9)$$

$$= (\ln(26!) - (\ln((26\rho)!) + \ln((26(1-\rho))!)))$$

Dado que el segundo y tercer término no son necesariamente grandes, se debe usar la aproximación de Stirling sin despreciar el término  $\sqrt{2\pi n}$ , es decir,  $\ln(n!) \approx n \ln(n) - n + \ln(\sqrt{2\pi n})$ , obteniendo así la siguiente expresión para la entropía:

$$\frac{S}{k_B} \approx -26\rho \ln\left(\frac{\rho}{1-\rho}\right) - \frac{1}{2} \ln(52\pi\rho(1-\rho)^{53}) \quad (10)$$

Del cual, teniendo en cuenta los valores obtenidos en la ecuación (6) para la densidad se pueden calcular

diferentes resultados teóricos. Definiendo la energía de la vecindad similar a la energía del gas ideal, es decir  $E = \frac{3N}{2} k_B T$ , con  $N$  el número de celdas vivas en una vecindad y  $T$  a la probabilidad de encontrar  $h = 26\rho$  celdas vivas en una vecindad, con lo cual  $E \approx 39\rho(t)k_B T$ .

Con lo cual, se tienen las siguiente variables termodinámicas:

Var.	Expresión
$V$	27
$N$	$(V-1)\rho$
$T$	$P(26\rho, t) = \binom{26}{26\rho} \rho^{26\rho} (1-\rho)^{26(1-\rho)}$
$E$	$\frac{3N}{2} k_B T$
$S$	$-k_B \left( N \ln\left(\frac{\rho}{1-\rho}\right) + \frac{1}{2} \ln(2\pi N(1-\rho)^{53}) \right)$

Cuadro 1: Variables termodinámicas del sistema discreto de estudio.

## Estado del arte

Si bien el concepto de autómatas celulares (CA) ya era conocido desde la década de 1950 gracias a John Von Neumann por su libro *Theory of Self-reproducing Automata* donde describe la dinámica de un sistema de celdas cuyo estado de viva/muerta en un instante de tiempo está definido por las condiciones del instante inmediatamente anterior.

Originalmente las reglas que rigen al sistema se expresan como

$$E_1, E_2, \dots / F_1, F_2, \dots$$

donde las  $E_i$  representan el número de vecinos vivos necesarias para mantener la celda viva y las  $F_i$  el número de vecinos vivos para que “nazca” una celda.

Los CA's no eran de mucho interés popular hasta que el “Juego de la Vida” de John Conway fue descrito por primera vez en 1970 [6]. El motivo de su popularidad fue la novedad al descubrir la regla 2,3/3 la cual provoca que las figuras observadas sobre el plano se repitan periódicamente, aquellas figuras fueron llamadas osciladores traslacionales o *Gliders*.

A pesar de la existencia de múltiples reglas para CA's, la mayoría de ellas dan lugar a comportamiento creciente ilimitado o a una muerte rápida de toda la población. Por lo que fue de particular interés el caso donde hay estabilidad como el generado por la regla 2,3/3.

Aunque, si bien la regla de Conway no siempre resulta en un comportamiento acotado, las figuras nunca crecen en configuraciones aleatorias por lo tanto fue posible definir cuidadosamente para sistemas también crecientes las reglas del “Juego de la Vida” como se describen informalmente a continuación[5]:

1. Todos los vecinos deben tocar a la celda candidata y se tratan de igual manera (vecindad de Moore).
2. Debe existir al menos un *glider*.
3. Las configuraciones aleatorias eventualmente se deben estabilizar.

La descripción formal de éstas reglas se puede detallar en el artículo de Bays [7].

Tiempo después del debut de Bays en este tema, muchos más se interesaron en el comportamiento de los autómatas celulares, dando lugar a nuevas propuestas. Una de las propuestas interesantes es el estudio termodinámico del sistema mediante la mecánica estadística [3]. Donde bajo cierta arbitrariedad se logró encontrar un equivalente a la temperatura para el sistema y de allí trabajar en el resto de relaciones termodinámicas consistentes con la teoría. Además del análisis de la entropía, se enfocó el estudio en la densidad total final del sistema y la estabilidad de los valores iniciales, obteniendo 0.37 como la densidad de convergencia aproximadamente.

Dada la popularidad de aquellos sistemas dinámicos, hubo interés también en el estudio de la dinámica de un sistema tridimensional. Al añadir una dimensión extra al arreglo, es necesario encontrar nuevas reglas que cumplan con las condiciones dadas por Adamatzky. Por lo que Bays escribe tiempo después un artículo proponiendo dichas reglas[2]. En el artículo se observaron como mejores candidatas las reglas R(4555) y las R(5766).

## Planteamiento del problema tentativo

A partir de un espacio finito, discreto y dinámico de 3 dimensiones compuesto de cubos como “unidades fundamentales” de este sistema que tienen dos posibles estados: “*vivas*” o “*muertas*” según las reglas del juego inspiradas en el Juego de la Vida de Conway [6], se definirán y analizarán las propiedades termodinámicas y mecánico-estadísticas de este sistema.

Con base en unas condiciones iniciales y reglas de juego impuestas por Conway en un espacio de 2 dimensiones, numerosos trabajos y modificaciones surgieron de la idea original; entre estos el artículo de Schulman y Seiden [3] en el que se estudia la mecánica estadística de un sistema dinámico y discreto de 2 dimensiones basado en el juego de la vida de Conway. Partiendo de conceptos como la densidad de celdas vivas y definiendo variables termodinámicas como la temperatura y la entropía lograron realizar mecánica estadística del equilibrio y no equilibrio, así como a altas y bajas temperaturas según sus definiciones.

Por otro lado, un trabajo de Shao y Chen [4] hace referencia al juego de la vida de Conway como un autómata celular para hacer simulaciones computacionales con el método de Monte Carlo y teniendo en cuenta la aproximación de la teoría de campo medio. En este artículo se obtuvo una densidad de celdas vivas  $\rho = 0,37017$  usando la teoría de campo medio frente a  $\rho = 0,37 \pm 0,003$  en la simulación de Monte Carlo.

Las propuestas de los artículos anteriores motivan un estudio que relacione ambos. Es por esto que como problema a resolver en este proyecto se ha escogido el Juego de la Vida de Conway en 3 dimensiones, así que en este caso las reglas del juego se ven modificadas debido al grado de libertad adicional en el sistema; asimismo, la vecindad de una celda aumenta de  $k = 8$  a  $k = 26$  al pasar de 2 a 3 dimensiones.

En primer lugar, se establecerán las reglas del sistema dinámico, debido a que el problema original es en 2 dimensiones, a continuación se introduce el concepto de densidad de celdas vivas en el arreglo y a partir de las reglas establecidas es posible asociar una probabilidad de encontrar una celda viva. Luego es posible hallar el valor esperado de la densidad de celdas vivas cuando el sistema llega a un estado de equilibrio.

Dada la forma de la probabilidad de encontrar una celda viva, se usará el ensamble microcanónico para hallar valores termodinámicos como la entropía, temperatura y energía interna. Todo el procedimiento mencionado hasta este punto está justificado analíticamente e inspirado en el artículo de Schulman y Seiden [3] que trata propiedades mecánico-estadísticas de un arreglo de 2 dimensiones.

Para el ámbito computacional se tendrá en cuenta el trabajo de Shao y Chen [4], en especial por sus predicciones para la densidad de celdas vivas en el tiempo, su límite asintótico (figura 1) y su dependencia de la densidad de celdas vivas inicial (figura 2). Claro está que al tratarse de un arreglo de 3 dimensiones, los resultados que obtendremos diferirán en ciertos aspectos respecto a los de este trabajo referencia, pero aun así es posible realizar algunas comparaciones.

## Motivación y Justificación

Desde el ámbito académico, las motivaciones para realizar este estudio radican en conocer comportamientos de sistemas físicos en el no equilibrio [3], que en nuestro caso al trabajar con sistemas programables y ejecutables computacionalmente permiten analizar desde una mirada *micro* sistemas de este tipo, tales como gases o líquidos. Es claro que las “reglas de juego” en sistemas físicos reales no son iguales a las utilizadas en este proyecto, sin embargo, es importante hacer una primera aproximación a la física del no equilibrio, y con el uso del ensamble microcanónico conocer variables termodinámicas como la temperatura, energía interna y entropía.

Por otro lado, el estudiar estos arreglos de 3 dimensiones tiene motivaciones de tipo biológicas [3], estos porque puede dar patrones de reproducción y evolución de diferentes especies, así como la evolución de las células y el alcance en la propagación de virus. Asimismo, puede tener aplicaciones directas en la sociedad ya que es un posible modelo de flujo de tráfico.

Dicho lo anterior, es notable la relevancia de dar solución al problema propuesto, desde sus aplicaciones en ramas como la biología o en el estudio de sistemas de muchos elementos, hasta la física misma para sistemas fuera del equilibrio donde los modelos se vuelven más complicados, o sistemas sin Hamiltoniano o energía asociada y que son altamente no lineales, como el Juego de la Vida.

## Objetivo general

Analizar el comportamiento de los valores estables de la densidad de celdas vivas y variables termodinámicas como la temperatura, energía interna y entropía para diferentes configuraciones iniciales de un sistema dinámico, finito y discreto de 3 dimensiones con reglas basadas en el Juego de la Vida de Conway (regla R(5766)[2]) bajo la mecánica estadística con el ensamble microcanónico, frente a simulaciones en ordenador.

## Objetivos específicos

1. Definir el concepto de densidad de celdas vivas y probabilidad de encontrar una celda viva en el contexto del juego de la vida de Conway.
2. Modelar el sistema dinámico a estudiar bajo el ensamble microcanónico de la mecánica estadística.
3. Calcular variables termodinámicas como la temperatura, energía interna y entropía para

sistemas con condiciones iniciales, definidas por el ensamble microcanónico.

4. Simular computacionalmente sistemas de 3 dimensiones bajo la regla R(5766) para calcular en estos las variables termodinámicas del objetivo anterior, así como la densidad de celdas vivas cuando el sistema está en equilibrio.
5. Comparar los resultados analíticos y computacionales de las variables termodinámicas y la densidad de celdas vivas en sus valores estables, al igual que contrastar los sistemas de 3 dimensiones frente a lo que se obtiene en 2 dimensiones.

## Metodología

Para analizar el problema del arreglo en tres dimensiones discretizado es necesario replantear las condiciones que dan lugar a la dinámica del sistema. Puesto que la forma de abordar dicho sistema no es tan directa como parece, dentro de la bibliografía se encuentran propuestas para las condiciones como las reglas R(4555) y las R(5766) [2].

Una vez definidas las nuevas reglas se hace una descripción de la dinámica del sistema estableciendo una densidad de celdas vivientes cuyo valor depende inmediatamente del estado del sistema en el estado del instante anterior. Sobre este modelo se toma inspiración en el trabajo de Schulman y Seiden [3] quienes con cierta arbitrariedad definen una “temperatura” asociada al sistema y a partir de allí se hace un análisis termodinámico incluyendo las variables de energía interna y entropía.

Después de haber construido el modelo con sus predicciones se hace uso de herramientas computacionales para realizar una serie de simulaciones del sistema de celdas 3D con distintas condiciones iniciales y de esta manera probar la efectividad del método. Para lo cual se cuenta con los lenguajes y software JavaScript, Python y Octave.

Finalmente los resultados experimentales y teóricos de las variables termodinámicas del sistema se comparan con los del sistema original en dos dimensiones y se analizan las diferencias en busca de un cambio en la contribución de cada estado del sistema que se pueda asociar a la dimensionalidad de éste mismo.

## Resultados esperados

Para el análisis de los datos se espera simular un sistema de celular automata en 3-dimensiones de manera satisfactoria que cumpla con las características

del juego de la vida con reglas  $R(5766)$ . Adquiriendo así un conocimiento de las simulaciones de sistemas a través de celular automata, util en sistemas cuyos estados estan en cambios según unas reglas.

Teniendo en cuenta los resultados obtenidos según la ecuación (6) es posible calcular las diferentes definiciones para el sistema en equilibrio, es decir la entropía y la energía, encontrando que los valores esperados son los siguientes.

Variable	Valor esperado
Número de partículas ( $N$ )	6,656
Densidad ( $\rho$ )	0,256
Temperatura ( $T$ )	0,176 K
Energía interna ( $E$ )	$2,425 \times 10^{-23}$ J
Entropía ( $S$ )	$1,803 \times 10^{-22} \frac{J}{K}$

Cuadro 2: Valores numéricos esperados de la simulación.

Además de confirmar el comportamiento de la densidad para cubos de diferentes tamaño, esperando que esta se estabilice al rededor de un valor a partir de diferentes densidades  $\rho_0$  esperando obtener una gráficos similares a los siguientes:

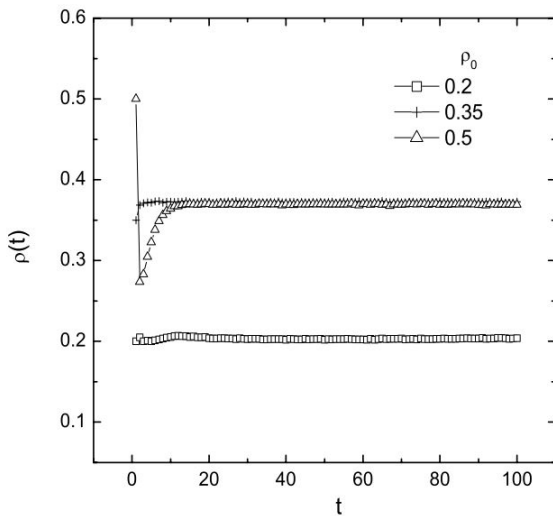


Figura 1: Comportamiento de la densidad respecto al tiempo para un sistema de dos dimensiones.

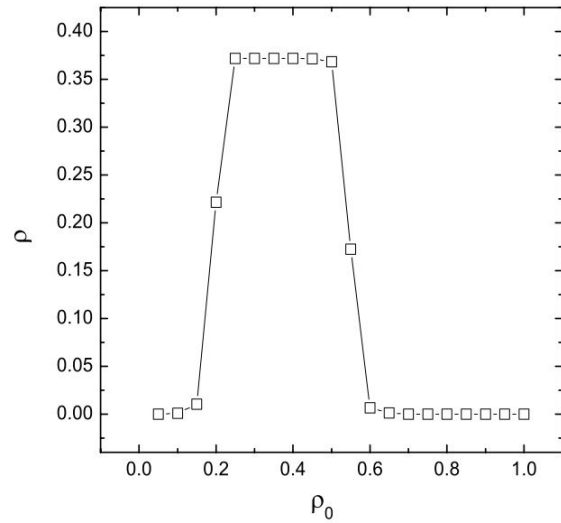


Figura 2: Densidad a la que se estabiliza el sistema para diferentes densidades iniciales.

Las anteriores figuras fueron tomadas del artículo “Game of Life on the Equal Degree Random Lattice” [4] página 3.

Adicionalmente se espera tener claridad del comportamiento de cada una de las variables del sistema y su interpretación de manera física; interiorizando así mejor los conceptos de la mecánica estadística y el ensamble microcanónico en este caso.

## Cronograma

Las diferentes actividades propuestas para la realización de este artículo se encuentran ordenadas de acuerdo con el siguiente diagrama de Gantt:

Actividad	Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Sem 6	Sem 7	Sem 8	Sem 9	Sem 10
Definir el concepto de densidad de celdas vivasy probabilidad de encontrar una celda viva.										
Modelar el sistema dinámico a estudiar bajo el ensamble microcanónico de la mecánica estadística.										
Calcular variables termodinámicas para sistemas con condiciones iniciales, definidas sobre el sistema.										
Simular computacionalmente sistemas de 3 dimensiones bajo la regla R(5766)										
Comparar los resultados analíticos y computacionales de las variables termodinámicas obtenidas para un sistemas de 3 dimensiones frente a lo que se obtiene en 2 dimensiones.										

Figura 3: Cronograma previsto para el cumplimiento de cada uno de los objetivos específicos.

## Recursos disponibles

Con la finalidad de cumplir cada uno de los objetivos planteados anteriormente se cuentan con diversos recursos digitales, contando con el ingreso a las bases de datos que ofrece la Universidad Nacional de Colombia, artículos diversos de la página “ResearchGate” y los artículos o libros de libre acceso disponibles en la red.

Adicionalmente se cuenta con diferentes lenguajes de programación, entre ellos JavaScript (con librerías como Three.js, la cuál permite el modelado 3D sobre el navegador), Octave y Python (con librerías como Numpy, matplotlib y cython, este último si es necesario optimizar el rendimiento del código) , con los cuales se procederá en los métodos numéricos y modelos 2D y 3D necesarios para la realización del presente artículo.

Finalmente debido al carácter computacional de los objetivos, es necesario el uso de computadores para cada uno de los autores del presente artículo.

## Desarrollo de los objetivos específicos

### 1. Definir el concepto de densidad de celdas vivas y probabilidad de encontrar una celda viva en el contexto del juego de la vida de Conway.

Con base en el trabajo realizado por Schulman y Seiden en el juego de la vida de 2 dimensiones [3], se desarrollaron los conceptos de densidad de celdas vivas y probabilidad de encontrar un celda viva para un sistema de 3 dimensiones en la sección del marco teórico. En el caso de la densidad de celdas vivas  $\rho$ , no es más que la medida de celdas vivas en un instante de tiempo comparado con el total de celdas en el arreglo. Lo interesante de esta variable es que fue posible encontrar una expresión para conocer el estado de una celda un instante de tiempo después, cuyo desarrollo completo está en los anexos y el resultado fue:

$$\begin{aligned} \sigma_\alpha(t+1) = & \delta(6, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) + \sigma_\alpha(t)(\delta(5, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) \\ & + \delta(7, \sum_{\alpha'=1}^{26} \sigma_{\alpha'})) \end{aligned} \quad (11)$$

Donde  $\sigma_\alpha = 1$  si la celda  $\alpha$  está viva y toma el valor de 0 si está muerta. Por otro lado, para el

caso de la probabilidad de encontrar una celda viva se tiene la expresión  $P_t(\sigma_\alpha = 1) = \rho(t)$ , luego para encontrar  $h$  celdas vivas en la vecindad se encontró una distribución binomial explicada en los anexos y que se resume en la siguiente expresión:

$$P(h, t) = \binom{26}{h} \rho^h (1 - \rho)^{26-h} \quad (12)$$

Y al considerar la propia celda en cuestión junto con sus vecinas obteniendo la probabilidad total en un tiempo  $t$ , también justificada en los anexos, expuesta a continuación:

$$P_t(\sigma_\alpha, h) = (\sigma_\alpha \rho(t) + (1 - \sigma_\alpha)(1 - \rho(t))) P(h, t) \quad (13)$$

### 2. Modelar el sistema dinámico a estudiar bajo el ensamble microcanónico de la mecánica estadística.

A partir de la expresión para la probabilidad encontrada en el desarrollo del objetivo anterior, se notó que para cada una de las celdas en los estados  $\sigma_i$  distinta son constantes, de donde fue posible encontrar el número total de estados  $W$  en la vecindad de cada celda, y el resultado se mostró en la ecuación (7):

$$W = \frac{26!}{h!(26-h)!} \quad (14)$$

Luego era posible usar el ensamble microcanónico para modelar el sistema dinámico en estudio, y expresando el número de estados  $W$  en función de la densidad de celdas vivas se encontró el valor de la entropía del sistema usando la aproximación de Stirling, vista en la ecuación (10):

$$\frac{S}{k_B} \approx -26\rho \ln\left(\frac{\rho}{1-\rho}\right) - \frac{1}{2} \ln(52\pi\rho(1-\rho)^{53}) \quad (15)$$

La deducción de la anterior expresión de la entropía partiendo del número total de estados  $W$  se explica en detalle en la sección de anexos. A continuación, se definió la energía de la vecindad de manera similar a la del gas ideal:  $E = \frac{3N}{2} k_B T$ , la cual en términos de la densidad de celdas vivas resulta como  $E \approx 39\rho(t) k_B T$ . Dado lo anterior fue posible calcular variables termodinámicas relevantes mostradas en el Cuadro 1 en la sección del marco teórico.



3. Calcular variables termodinámicas como la temperatura, energía interna y entropía para sistemas con condiciones iniciales, definidas por el ensamble microcanónico:

Tomando los resultados obtenidos para la expresión de la energía interna de un gas ideal de partículas libres obtenidas a través del uso del ensamble microcanónico [8]; se tiene que el valor de energía total del sistema es  $E = \frac{DN}{2}k_B T$ , donde  $D$  es el número de grados de libertad del sistema y  $N$  el número de celdas vivas. En el caso del sistema de estudio  $D = 3$  y  $N = 26\rho(t)$ , con lo cual se obtiene que la energía interna del sistema cuando se estabiliza viene dado por:

$$E \approx 39\rho k_B T \quad (16)$$

Adicionalmente para el caso de la temperatura del sistema, esta se definió como  $T = P(26\rho)$  tal que esta es la probabilidad de encontrar  $\langle N \rangle = N = 26\rho$  celdas vivas cuando el sistema está estable, con lo cual:

$$T = \binom{26}{26\rho} \rho^{26\rho} (1-\rho)^{26(1-\rho)} \quad (17)$$

Usando la simulación hecha y usando el código comentado e ilustrado en el anexo mostrado para los objetivos específicos 3 y 4 se tomaron los datos de la densidad respecto de las generaciones que van pasando para dos casos diferentes. Inicialmente dentro de la simulación es posible modificar dos variables iniciales, el número total de celdas en estado arbitrario (manejado a través del valor  $l$  el cual corresponde a  $l \times l \times l = l^3$  número total de celdas en estado inicial arbitrario), y, la densidad de celdas vivas para  $t = 0$ . En el siguiente gráfico se presentan los resultados obtenidos a través de la simulación para los cambios obtenidos en la densidad respecto de variaciones discretas de  $t$  (las variaciones discretas de  $t$  se denominarán generaciones), para diferentes valores de densidad inicial:

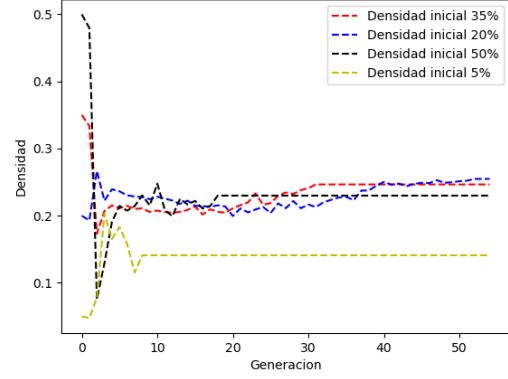


Figura 4: Variación de la densidad, para cada intervalo discreto de tiempo, para diferentes densidades iniciales en un cubo de  $50 \times 50 \times 50$  celdas en estado inicial arbitrario. Resultados obtenidos a través de la simulación realizada a lo largo del presente proyecto.

Adicionalmente a través de los resultados obtenidos en la gráfica anterior, junto con diferentes datos adicionales obtenidos a través de la simulación, se encontraron los valores en los cuales el valor de la densidad cuando el sistema es estable respecto de la densidad definida inicialmente, es decir  $\rho_0$ , obteniendo los siguientes resultados:

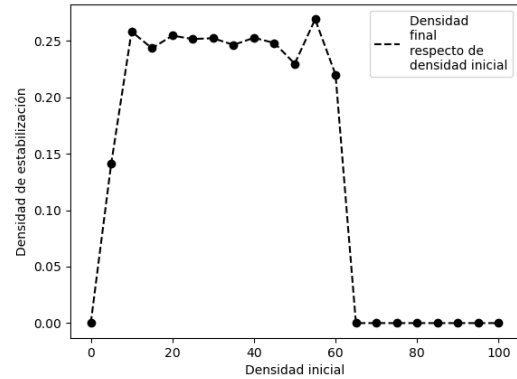


Figura 5: Variación de la densidad final respecto de la densidad inicial en un cubo de  $50 \times 50 \times 50$  celdas en estado inicial arbitrario. Los datos fueron obtenidos a través de las simulaciones desarrolladas a lo largo del presente proyecto.

Ahora, variando el número de celdas en el sistema con estado inicial arbitrario, a través de la simulación (anexo objetivos 3 y 4), se registraron los siguientes datos para la variación de la densidad respecto de las variaciones discretas de

tiempo (generaciones), para diferentes número de celdas iniciales:

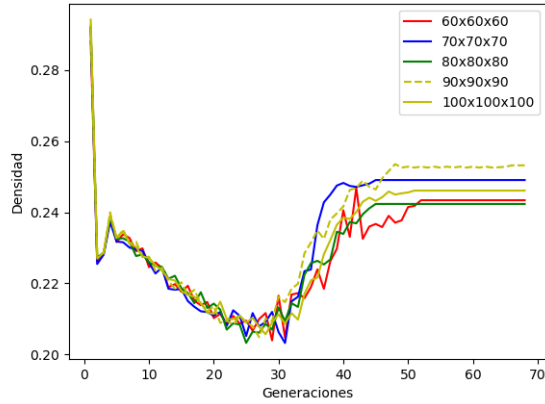
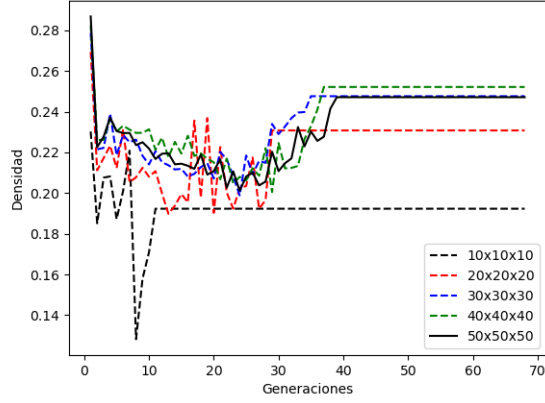


Figura 6: Resultados obtenidos de la densidad final respecto de las variaciones discretas de tiempo, para diferente número total de celdas en el espacio, a una densidad constante de  $0,3 \frac{\text{Celdas vivas}}{\text{Número de celdas}}$ . Datos obtenidos a través de la simulación realizada para el desarrollo del presente proyecto.

Adicionalmente, a partir de los resultados anteriores y los valores obtenidos a través de la simulación es posible encontrar el siguiente comportamiento de la densidad final a la que el sistema está en equilibrio respecto del número de celdas totales en cada cara del cubo más grande ( $l$ ), resultado que se registra en la siguiente gráfica:

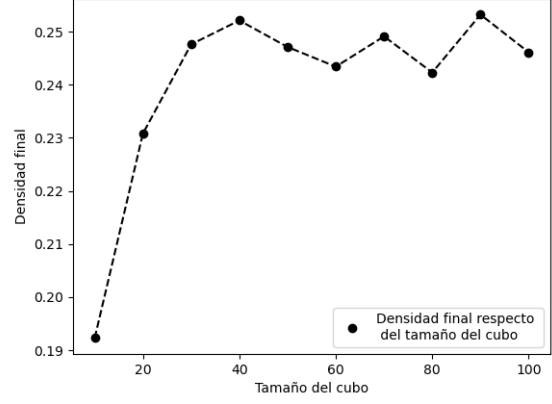


Figura 7: Densidad final a la que el sistema está en equilibrio respecto de la variación de  $l$ , asociado al número total de celdas en estado arbitrario del sistema. Resultados obtenidos a través de la simulación presentada en los anexos del presente objetivo.

Teniendo en cuenta las definiciones que se establecieron en el cuadro 1 para cada una de las variables termodinámicas del sistema, las cuales son definidas por la densidad final ( $\rho$ ) del sistema, es posible calcular cada una de ellas usando que  $\rho = \langle \rho \rangle$ , es decir el promedio estadístico de los resultados obtenidos en las gráficas 5 y 7, de esta manera es posible encontrar los siguientes resultados:

Var.	$L_{var}$	$\rho_{0var}$	Promedio
$N$	6,39	6,52	6,45
$\rho = \langle \rho \rangle$	0,245	0,251	0,248
$T(K)$	0,179	0,178	0,178
$E(\times 10^{-23} J)$	2,37	2,40	2,39
$S(\times 10^{-22} \frac{J}{K})$	1,765	1,783	1,774

Cuadro 3: Resultados obtenidos para cada variable termodinámica definida inicialmente. Cada valor es obtenido al reemplazar los valores de  $\rho = \langle \rho \rangle$  según las definiciones hechas en el cuadro 1.

Encontrando así los valores de cada una de las variables termodinámicas planteadas en al inicio (Cuadro 1).

#### 4. Simular computacionalmente sistemas de 3 dimensiones bajo la regla R(5766) para calcular en estos las variables termodinámicas del objetivo anterior, así como la densidad de celdas vivas cuando el sistema está en equilibrio.

Se realizó una simulación del sistema dinámico mediante un programa usando JavaScript en el

cual se varían las condiciones iniciales de tamaño del cubo y densidad de celdas vivas. Allí se establece la dependencia de cada generación en la generación inmediatamente anterior.

Tomando un cubo de tamaño 30 y la densidad inicial del programa como 0.3 se puede observar gráficamente el paso de la generación 12 a la generación 13. El cambio entre generaciones está regido por la regla R(5766), es decir, si una celda viva tiene de 5 a 7 celdas vivas en su vecindad ésta se mantiene viva y si la celda muerta cuenta con 6 celdas vivas en su vecindad ésta cambiará de estado muerto a vivo como se muestra en la figura 7.

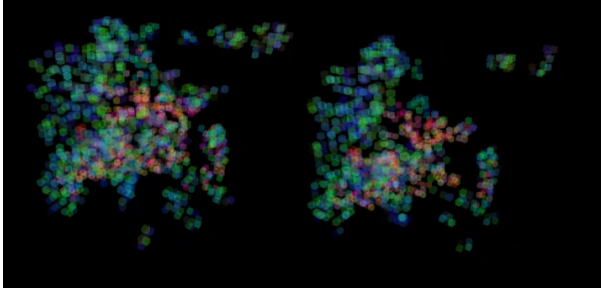


Figura 8: Comparación gráfica de dos generaciones de celdas.

El programa se creó de tal forma que se dejan de procesar generaciones siguientes cuando la densidad de celdas vivas se mantiene constante. Por lo que finalmente la simulación muestra numérica y gráficamente datos de la densidad final y las densidades intermedias dependiendo de las condiciones iniciales.

En este caso a partir de la generación 17 (figura 8) se dejan de procesar más generaciones puesto que la densidad permanecerá constante como 0.246 aproximadamente a partir de este punto.

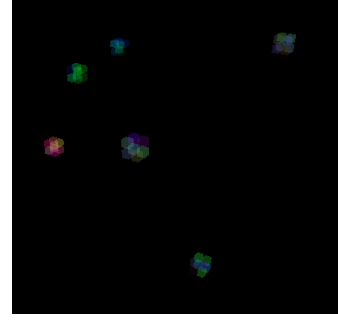


Figura 9: Estado final estable del sistema en representación gráfica.

5. **Comparar los resultados analíticos y computacionales de las variables termodinámicas y la densidad de celdas vivas en sus valores estables, al igual que contrastar los sistemas de 3 dimensiones frente a lo que se obtiene en 2 dimensiones.**

Usando los resultados obtenidos en los trabajos de Zhi-Gang Shao y T. Chen [4] se obtiene que la densidad para un sistema en 2 dimensiones es de  $\rho_{2D} = 0,37017$ ; reemplazando este valor en las ecuaciones obtenidas para las variables termodinámicas del sistema bidimensional (definidas en los anexos referentes al presente objetivo) se encontraron así los resultados para cada una de las variables termodinámicas del sistema bidimensional, adicionalmente teniendo en cuenta el cuadro 3 se contrastan los resultados obtenidos para los dos grados de libertad que se estudiaron:

Variable	2D	3D
$N$	2,96	6,45
$\rho$	0,370	0,248
$T(K)$	0,282	0,176
$E(\times 10^{-23}J)$	1,151	2,39
$S(\times 10^{-22}\frac{J}{K})$	0,553	1,774

Cuadro 4: Comparación de las variables termodinámicas para los sistemas de 2 y 3 dimensiones.

Además de encontrar las gráficas 5 y 2, junto con las gráficas 4 y 1, sobre las cuales es posible realizar un análisis detallado para encontrar las diferencias entre estos dos sistemas.

## Análisis de resultados

1. **Definir el concepto de densidad de celdas vivas y probabilidad de encontrar una cel-**

## da viva en el contexto del juego de la vida de Conway.

El procedimiento realizado para un tratamiento tridimensional del juego de la vida fue inspirado en el trabajo en 2 dimensiones de Seiden y Schulman [3], y la generalización, definiciones y cálculos hechos en este artículo tienen su justificación en dicho trabajo guía.

Además cabe destacar la expresión encontrada para predecir el estado de una celda en función de su estado actual y el de su vecindad, ya que este arreglo estudiado es similar a sistemas físicos de 2 estados tales como materiales paramagnéticos afectados por un campo magnético, donde los átomos pueden alinearse paralela o antiparalelamente al campo magnético. Por lo tanto, el sistema aquí usado guarda similitud con sistemas de este, pero se diferencia en que en el paramagnetismo el campo actúa de manera independiente sobre cada momento magnético y no hay interacción entre estos.

Asimismo, debido a que no hay distinción alguna entre las celdas que componen el arreglo, la probabilidad de encontrar  $h$  celdas vivas en la vecindad de una celda también aplica para cualquier celda del sistema, lo que permitió usar el ensamble microcanónico y que se tratará en el siguiente apartado.

### 2. Modelar el sistema dinámico a estudiar bajo el ensamble microcanónico de la mecánica estadística.

Al encontrar que la probabilidad para cada celda en estados  $\sigma_i$  distintos era la misma se consideró apropiado el uso del ensamble microcanónico para modelar el sistema y conocer variables termodinámicas de este como la entropía, la temperatura y la energía interna. Para el caso de la entropía, su cálculo es consecuencia directa de hacer uso del ensamble microcanónico, ya que al conocer el número total de estados en una vecindad es posible conocer la entropía a través del principio de Boltzmann.

Por otro lado, la definición de temperatura utilizada se basó en la definición empleada para el sistema de 2 dimensiones de Seiden y Schulman [3], y en el caso de la energía interna se decidió por una definición similar a la de un gas ideal:  $E = \frac{3N}{2}k_B T$ , que como se ha mencionado secciones atrás, el modelo de este sistema dinámico es útil para entender el comportamiento de muchos sistemas físicos reales, entre estos el gas ideal que a pesar de ser un modelo simple para comprender el comportamiento

de los gases, permite una visión microscópica aproximada a las interacciones moleculares y atómicas que ocurren dentro de este, haciendo claridad que la forma de dichas interacciones no se modelan con las reglas del juego estudiadas en el sistema estudiado.

### 3. Calcular variables termodinámicas como la temperatura, energía interna y entropía para sistemas con condiciones iniciales, definidas por el ensamble microcanónico.

A pesar de los resultados de la simulación y la obtención de los datos registrados en el cuadro 3, estos tienen diferencias respecto de los valores esperados, registrados en el cuadro 2. Mostrando los siguientes errores relativos:

Variable	Error relativo
$N$	3,03 %
$\rho$	3,03 %
$T$	1,52 %
$E$	1,61 %
$S$	1,60 %

Cuadro 5: Errores relativos para cada variable termodinámica.

Dado que los errores más grandes que se obtienen son de apenas el 3 % es posible considerar que los resultados obtenidos son satisfactorios, dado que estos errores se dan debido a las aproximaciones hechas, dado que un cubo cuyos lados tienen tamaño 100 no equivale a todo el espacio. No se pudieron trabajar con cubos más grandes debido a las limitaciones de la máquina virtual sobre la que trabaja JavaScript.

Adicionalmente a pesar de tener un valor de entropía, el cual es de  $1,774 \times 10^{-22}$ , es difícil deducir algo de este puesto que solo nos indica que las celdas del sistema están correlacionadas y tienen algún orden, en el cual podemos conocer que tenemos un 12,85 de incertidumbre (Según la entropía de Shannon  $\frac{S}{k_b}$ ) lo que indica el valor de máxima de información perdida, debido a la equiprobabilidad del estado de cada celda.

Por otra parte, del que más se puede concluir es del valor de la densidad y el número de partículas por vecindad, de los cuales se obtiene que en cada vecindad es posible encontrar entre 5 y 7 celdas vivas, cosa que es esperable pues es el

valor esperado de la regla  $R(5766)$ . Además cada vecindad tiene una temperatura o probabilidad de estar vivas del 17,6 %, cosa que podría cambiar cambiando las reglas del sistema, es decir la temperatura permite notar la regla bajo la cual se está trabajando. Finalmente el valor de la energía encontrado ( $2,39 \times 10^{-23}$ ) es una extensión de la información que provee la temperatura, puesto que el número de partículas depende también de la regla impuesta al inicio.

#### 4. Simular computacionalmente sistemas de 3 dimensiones bajo la regla $R(5766)$ para calcular en estos las variables termodinámicas del objetivo anterior, así como la densidad de celdas vivas cuando el sistema está en equilibrio.

Las gráficas mostradas en la sección anterior muestran una tendencia del sistema a estabilizarse en una densidad final de  $\rho = 0,248$  aproximadamente para densidades iniciales entre 10 % y 60 % tal como se esperaba.

Por otro lado dada la aleatoriedad en la que aparecían las celdas vivas de las condiciones iniciales, la dinámica del sistema no se puede predecir con exactitud en los instantes iniciales de la simulación por lo que se observa un comportamiento caótico para las primeras generaciones. Igualmente por esto las generaciones necesarias para estabilizar la densidad tienen una variación bastante amplia (entre 10 y 55).

Sin embargo para valores en el rango más alto de tamaño del cubo (6) se observa que hay mas uniformidad en la secuencia de los datos. Aunque la densidad final del sistema parece mantenerse constante en los rangos de valores iniciales mencionados, cuando la densidad es diferente de cero hay variaciones notables respecto de la densidad promedio. Para obtener una gráfica más similar a la del sistema 2D es necesario tomar datos en un intervalo menor de densidades iniciales (en este caso se tomaron densidades en intervalos de 5) y dado el comportamiento aleatorio también conviene tomar el promedio de varios valores para cada densidad inicial para un análisis más profundo. Para el caso de este proyecto los resultados obtenidos por la simulación son suficientes para notar una tendencia en el comportamiento.

#### 5. Comparar los resultados analíticos y computacionales de las variables termodinámicas y la densidad de celdas vivas en

**sus valores estables, al igual que contrastar los sistemas de 3 dimensiones frente a lo que se obtiene en 2 dimensiones.**

Con los datos registrados en el cuadro (4) es posible notar que la densidad en el sistema tridimensional es más baja que en el caso bidimensional, esto implica que haya una mayor probabilidad de encontrar una celda viva en el sistema en dos dimensiones comparada con el sistema en estudio, esto, seguramente, debido a que hay una menor cantidad de celdas posible, aumentando la probabilidad de encontrar 2 o 3 celdas vivas en 8 celdas vecinas, comparado con encontrar entre 5 y 7 celdas vivas entre 26 celdas vecinas.

De los datos de entropía, aunque en los dos casos está maximizada es posible notar que hay una menor pérdida de información en el sistema bidimensional, con lo cual es posible asegurar que el añadir un grado de libertad espacial al sistema hace que se pierda más información y se aumente el desorden de este.

A pesar de las diferencias reportadas por los datos, los comportamientos encontrados en las gráficas 5, 2, 4 y 1 es posible notar un comportamiento similar de la vida en los dos sistemas, en especial entre las gráficas (5) y (2), las cuales toman una forma de campana en intervalos similares entre los dos casos salvo un leve corrimiento hacia la derecha en el caso tridimensional. En el caso de las gráficas (4) y (1) es posible notar una similitud en la velocidad en la que el sistema se estabiliza, encontrando en ambos casos, que el sistema estará estable antes de la generación 50.

## Conclusiones

- El ensamble microcanónico fue el más apropiado para analizar el sistema dinámico discreto de 3 dimensiones de el juego de la vida de Conway donde los datos de densidad fueron consistentes con los esperados tomando como guía el trabajo de Seiden y Schulman [3]
- Se encontró el valor de la densidad del sistema, encontrando un valor de 0,248 con un error relativo del 3 %.
- Se encontraron los valores de todas y cada una de las variables termodinámicas definidas sobre el sistema, encontrando un valor de 0,178K para la temperatura,  $2,39 \times 10^{-23} J$  para la energía

y  $1,774 \frac{J}{K}$  para la entropía, con un error relativo máximo de 3 %.

- Se encontró que el sistema tiene una incertidumbre de 12,85 según la entropía de Shannon, encontrando el máximo de información perdida del sistema.
- La temperatura del sistema es una variable que es característica de la regla impuesta al inicio, por tanto cambios en la temperatura altera todas las variables.
- El sistema dinámico y discreto de 3 dimensiones estudiado puede considerarse como una primera aproximación a sistemas físicos no lineales, fuera del equilibrio, pero especialmente para sistemas de 2 estados como en materiales paramagnéticos en un campo magnético. Por lo tanto este trabajo podría ser un punto de partida para modelar sistemas como los anteriores.
- La simulación es bastante efectiva como una aproximación al sistema, sin embargo conforme se puede hacer un sistema más grande más tiende al problema ideal.
- Los principales errores obtenidos en las medidas fueron debido a las limitaciones del software.
- Se encontró, a través de la entropía, que el añadir un grado de libertad al sistema hace que haya un mayor desorden y pérdida de la información.
- Tanto el sistema bidimensional como el sistema tridimensional tienden a estabilizarse (o convertirse en cíclicos) antes de las 50 generaciones.
- La densidad final respecto de la inicial sigue un comportamiento de tipo campana para cualquiera de los dos sistemas, salvo un leve corrimiento.

## Anexos

### • Definir el concepto de densidad de celdas vivas y probabilidad de encontrar una celda viva en el contexto del juego de la vida de Conway.

El resultado expuesto en la sección de desarrollo de objetivos específicos, ecuación (11), se deduce a partir de las reglas establecidas para el juego de la vida estudiado en este proyecto, las cuales son las R(5766). De esta manera, por complejo que pueda lucir el sistema en un instante determinado de tiempo se puede tener certeza del estado de cada una de

las celdas según sus vecinas y las reglas, ya que estas continúan vivas si tienen de 5 a 7 celdas vecinas vivas y revivirán cuando 6 de sus vecinas viven.

Lo anterior sugiere definir un parámetro que dé cuenta del estado de una celda, por lo que se define  $\sigma_\alpha$  como dicho parámetro, el cual es igual a 1 cuando la celda  $\alpha$  está viva y 0 si está muerta. Por lo tanto, si se quiere obtener una expresión que diga el estado de una celda específica un instante de tiempo posterior al presente en función de su estado presente y el de sus vecinas, es intuitivo pensar en introducir una delta de Kronecker y según las reglas R(5766) expresar sus variables.

Si la celda está actualmente muerta, necesitará 6 vecinas vivas para revivir, luego es claro que la delta de Kronecker a usar es  $\delta(6, \sum_{\alpha'=1}^{26} \sigma_{\alpha'})$ , donde  $\alpha'$  expresa las celdas vecinas de una determinada celda y se suma sobre las 26 celdas vecinas de esta. Por otro lado, si la celda está actualmente viva necesita de 5 a 7 celdas vecinas vivas para seguir en dicho estado, así que en este caso se hace uso de la suma de deltas para hacer la predicción:  $\sigma_\alpha(\delta(5, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}) + \delta(7, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}))$ , donde se ha ignorado la delta  $\delta(6, \sum_{\alpha'=1}^{26} \sigma_{\alpha'})$  ya que de darse esta posibilidad, ya estaría contándose por el primer caso ya descrito, además se ha puesto por delante  $\sigma_\alpha$  para que esta contribución solo se tenga en cuenta cuando  $\sigma_\alpha = 1$ , que es cuando la celda está actualmente viva. Con lo anterior, es posible escribir la expresión para el estado de una celda un instante de tiempo posterior al actual sumando cada contribución:

$$\sigma_\alpha(t+1) = \delta(6, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}(t)) + \sigma_\alpha(t)(\delta(5, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}(t)) + \delta(7, \sum_{\alpha'=1}^{26} \sigma_{\alpha'}(t))) \quad (18)$$

Para el caso de la probabilidad de encontrar  $h$  celdas vecinas vivas de una celda determinada se usa la distribución de probabilidad binomial, ya que a pesar de que está suele usarse cuando se quiere contar el número de éxitos tras un determinado número de ensayos independientes entre sí con una probabilidad fija para los resultados exitosos, es posible usarla en nuestro caso ya que  $h$  es el equivalente del número de éxitos, el número total de celdas vecinas (26) es el número total de ensayos y la probabilidad de encontrar una celda viva  $\rho$  es la probabilidad de éxito, luego usando la definición de distribución binomial puede escribirse la probabilidad de encontrar  $h$  celdas vecinas vivas como sigue:

$$P(h, t) = \binom{26}{h} \rho^h (1 - \rho)^{26-h} \quad (19)$$

Además, al considerar la celda en cuestión junto

con sus vecinas, es posible calcular una probabilidad total en un tiempo  $t$  como sigue a continuación:

$$P_t(\sigma_\alpha, h) = (\sigma_\alpha \rho(t) + (1 - \sigma_\alpha)(1 - \rho(t)))P(h, t) \quad (20)$$

Lo anterior porque cuando la celda está viva  $\sigma_\alpha = 1$  y se considera su respectiva probabilidad  $\rho(t)$ , mientras que si está muerta se tiene  $\sigma_\alpha = 0$ , luego  $1 - \sigma_\alpha = 1$  y se tiene en cuenta su probabilidad  $1 - \rho(t)$ . Todo esto se multiplica por la probabilidad encontrada en la ecuación (18) para obtener la probabilidad total.

### • Modelar el sistema dinámico a estudiar bajo el ensamble microcanónico de la mecánica estadística.

Partiendo del número total de estados posibles en la vecindad de una celda:

$$W = \frac{26!}{h!(26-h)!} \quad (21)$$

Es posible usar el ensamble microcanónico sobre el sistema para obtener información, entre ellos la entropía. Por lo tanto, usando resultados del ensamble microcanónico se tiene lo siguiente:

$$\begin{cases} h = 26\rho(t) \\ (26-h) = 26(1-\rho(t)) \end{cases} \quad (22)$$

$$\therefore W = \frac{26!}{(26\rho(t))!(26(1-\rho(t)))!}$$

Por lo que se puede encontrar la entropía del sistema, tomando  $\rho(t)$  como  $\rho$  para simplificar la escritura.

$$\begin{aligned} \frac{S}{k_B} &= \ln(W) \\ &= (\ln(26!) - (\ln((26\rho)!)) + \ln((26(1-\rho))!)) \end{aligned} \quad (23)$$

Dado que el segundo y tercer término no son necesariamente grandes, debe usarse la aproximación de Stirling sin despreciar el término  $\sqrt{2\pi n}$ , es decir,  $\ln(n!) \approx n \ln(n) - n + \ln(\sqrt{2\pi n})$ , es así como se llega a la siguiente expresión para la entropía:

$$\frac{S}{k_B} \approx -26\rho \ln\left(\frac{\rho}{1-\rho}\right) - \frac{1}{2} \ln(52\pi\rho(1-\rho)^{53}) \quad (24)$$

### • Simular computacionalmente sistemas de 3 dimensiones bajo la regla R(5766) para calcular en estos las variables termodinámicas del objetivo anterior, así como la densidad de celdas vivas cuando el sistema está en equilibrio.

Para realizar la simulación del sistema fue necesario realizar una aplicación usando el lenguaje de programación JavaScript, esto dado que es un lenguaje que permite fácilmente crear un contexto de 3 dimensiones.

El programa se realizó usando el paradigma de programación orientada a objetos, donde el programa completo se modularizó en varias partes, las cuales se muestran a continuación:

#### Intefaz.js

Para dar inicio al programa fue necesario programar por separado los parámetros que podría modificar el usuario del programa, con lo cual se realizó el siguiente script:

```
class Main{
  constructor(){
    this.cubesSize=0; //Se inicializan las variables del programa,
    this.dens=0; //tamaño del cubo y densidad inicial
    if(this.cubesSize==0){
      this.cubesSize=prompt("Valor de los lados del cubo (30 por defecto)");
      //A través de la maquina virtual se pregunta el tamaño del cubo al usuario.
    }
    if(this.cubesSize==0){
      this.cubesSize=30;
      //Si no ingresa un valor, por defecto será 30.
    }
    if(this.dens==0){
      this.dens=prompt("Porcentaje de cubos vivos 0-100 (30 por defecto)");
      //A través de la maquina virtual se pregunta la densidad inicial al usuario.
    }
    if(this.dens==0){
      this.dens=30;
      //Si no ingresa un valor, por defecto será 30.
    }
    if(this.cubesSize%2!=0){
      this.cubesSize+=1;
      //Por preferencia el tamaño del cubo será par.
    }
  };
  getDens(){return this.dens; //Da el valor de la densidad inicial
  getCubesSize(){return this.cubesSize; //Da el valor del tamaño del cubo
}
```

Figura 10: Parte del código encargado del inicio de la simulación y las variables que modifica el usuario.

#### Cells.js

Cada una de las celdas contiene una información que debe “saber” y unos métodos que debe “saber hacer”, por ello es necesario realizar la clase Cells, un script que maneja el comportamiento individual de cada celda, el script comentado se muestra a continuación:



```

class Cell{
  //El constructor es el método inicial que se encarga de crear cada celda.
  constructor(x,y,z,life,rule,geometry,material){
    //Se preguntan los parámetros iniciales de cada una de las celdas

    this.x=x;
    this.y=y; //Posición en el tensor o matriz de 3 dimensiones.
    this.z=z;

    this.life=life; //Estado inicial de la celda.
    this.life2=life; //Estado auxiliar de la celda (Se verá más adelante).

    this.rule=rule; //La regla con la que el sistema trabajará.
    this.n=0; //Variable que cuenta el número de celdas vivas al rededor de cada celda(Solo se inicializó).

    this.cube=new THREE.Mesh(geometry,material);
    this.cube.position.z=this.z*2/cubesSize-1; //Variables para la parte gráfica.
    this.cube.position.y=this.y*2/cubesSize;
    this.cube.position.x=this.x*2/cubesSize-1;
  }
}

```

Figura 11: Parte del script que contiene la información que cada celda “sabe”.

```

//Métodos que cada celda sabe hacer.
draw(){ //Dibujarse en la pantalla
  if (this.life){
    scene.add(this.cube)
  }
  else{
    scene.remove(this.cube)
  }
}

compare(O){ //Sabiendo el número de celdas que tiene vivas al rededor decide que hacer,
  //si cambiar o mantener el estado actual.
  this.n=0;
  if (this.life){ //Si la celda está viva
    if(this.rule.E1<this.n && this.nc<this.rule.E2){
      //Y se cumple la condición de mantener el estado, se guarda en una variable auxiliar.
      this.life2=true;
    }
    else{
      //Si no se cumplen las condiciones se cambia el estado de la celda.
      this.life2=false;
    }
  }
  else{ //Si la celda no está viva
    if(this.rule.F1<this.n && this.nc<this.rule.F2){
      //Y se cumple la regla para revivir, se cambia el valor del estado auxiliar.
      this.life2=true;
    }
  }
}

//NO SE PUEDE CAMBIAR AÚN EL ESTADO DE LA CELDA, PRIMERO SE DEBE VERIFICAR EL ESTADO DEL SISTEMA COMPLETO.
dens(){
  //Retorna la densidad local de cada celda.
  return this.n/26
}

```

Figura 12: Métodos que cada celda “sabe hacer”.

## Cube.js

Dado que se debe crear un sistema completo es necesario de un script que contenga todo el sistema, este es el script encargado de esto, conteniendo todos los métodos del sistema como un todo y no individualmente, mostrando todo el script a continuación:

```

class Cubes{
  //De forma análoga a cells es lo que el cubo en general sabe hacer.
  constructor(cubesize,dens,rule){
    this.size=cubesize; //Tamaño del cubo que contendrá las celdas.
    this.dens=dens; //Densidad inicial
    this.rule=rule; //Regla con la que trabaja

    this.geometry= new THREE.BoxGeometry(3/(2*this.size),3/(2*this.size),3/(2*this.size));
    //Variable que guarda la información visual de cada celda

  }
  //Métodos
  createCube(){
    //Crea las celdas dentro de una matriz de tres dimensiones.
    let material;
    let cubeColor;
    this.cubes=[]; //Creación de la matriz tridimensional.
    for (let i=0;i<this.size;i++){
      this.cubes.push([]) //Se crea la primera dimensión
      for (let j=0;j<this.size;j++){
        this.cubes[i].push([]) //Luego la segunda dimensión
        for (let k=0;k<this.size;k++){
          //Finalmente la tercera dimensión de la matriz se llena de celdas.
          cubeColor=((i+k*k+j*j))/((3*this.size*this.size)+50)*0xf0000
          material= new THREE.MeshBasicMaterial({color: cubeColor,transparent:true, opacity:0.3});
          //Según el valor de la densidad inicial se llena de nX celdas vivas.
          if (Math.random()*100<this.dens){
            this.cubes[i][j].push(new Cell(i,j,k,true,this.rule,this.geometry,material,this.size));
          }
          else{
            this.cubes[i][j].push(new Cell(i,j,k,false,this.rule,this.geometry,material,this.size));
          }
        }
      }
    }
    return this.cubes; //Retorna la matriz creada, la cual contiene toda la información.
  }
}

```

```

doThings(){ //En esete método cada cubo hace una cosa en específico
  this.cubes.forEach(xc->{
    xc.forEach(yc->{
      yc.forEach(zc->{
        let O=this.createEntorn(zc); //Crea una vecindad con todas las celdas vivas próximas.
        zc.compare(O) //Compara si debe estar viva o muerta según las condiciones del juego.
      });
    });
  });
  createEntorn(zc){ //Selección de las 26 celdas de al rededor sólo las que están vivas
    let n=0;
    for(let i=-1;i<2;i++){
      for(let j=-1;j<2;j++){
        for(let k=-1;k<2;k++){
          if(i!=0 || k!=0 || j!=0){
            if(zc.x+i!=1 && zc.y+j!=1 && zc.z+k!=1 && zc.z+k!=this.size &&
              zc.x+i!=this.size && zc.y+j!=this.size){
              if (this.cubes[zc.x+i][zc.y+j][zc.z+k].life){
                n++; //Y las cuenta para que se pueda comparar con este valor.
              }
            }
          }
        }
      }
    }
    return n; //Retorna el número de celdas vivas en la vecindad.
  }
}

```

```

actualice(renderer,scene,camera){
  //Aún no se ha actualizado el estado de todo el sistema, así que de eso se encarga este método.
  this.dens=0 //Resetea la densidad DEL SISTEMA COMPLETO.
  this.lives=0 //Resetea el número de celdas vivas en cada generación.
  this.cubes.forEach(xc->{
    xc.forEach(yc->{
      yc.forEach(zc->{
        if(zc.life){
          this.lives++; //Cuenta el número de celdas vivas
          this.dens+=zc.dens(); //Hace un conteo de la densidad
        }
        zc.life=zc.life2; //Actualiza el estado del sistema
        if(this.size<50){
          //Por temas de optimización el sistema no se dibuja para cubos muy grandes.
          zc.draw()
        }
      });
    });
  });
  renderer.render(scene,camera); //Actualiza la imagen que se verá en pantalla.
}

getDens(){return this.dens; //Devuelve la densidad del sistema.
getLives(){return this.lives; //Devuelve el número de celdas vivas.
}

```

Figura 13: Script completo y comentado que corresponde al archivo Cube.js

## GOL3D.js

Para coordinar todos los scripts anteriores es necesario un script que sea el principal, el cual se muestra a continuación:

```

const scene=new THREE.Scene(); //Se crea la escena donde estarán los cubos interactuando.
scene.background =new THREE.Color(0x000000) //Color del fondo del cubo.

//Creación de la cámara, desde donde se verá todo.
const camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight);
const renderer = new THREE.WebGLRenderer({precision:'lowp'});

renderer.setSize(window.innerWidth,window.innerHeight); //Ajustes para la parte visual.
document.body.appendChild(renderer.domElement);

let rule={E1:5,E2:7,F1:6,F2:6} //Regla con la que trabajaremos

let main=new Main(); //Creación de los datos iniciales.
let dens=main.getDens(); //Densidad inicial
let cubesSize=main.getCubesSize(); //Tamaño del cubo inicial

//Creación del sistema completo
let cube= new Cubes(cubesSize,dens,rule);
cube.createCube();

camera.position.y=2;

//Variables varias para el funcionamiento del programa.
let angle=0;
let gen=0;
let densn=0
let densn1=1

```



```
function animate(){
  if (densn0!=densn1 && isNaN(densn1)!=true){
    //Si el sistema está estable, o si desaparecen todas las celdas el programa se detiene.
    requestAnimationFrame(animate);
  }
  //Posición de la cámara en el espacio
  camera.position.z=3;
  camera.position.x=2;
  camera.position.y=2.87;
  camera.lookAt(0,0.5,0);

  densn0=cube.getDens();//Densidad de la generación n-1
  //Funcionamiento del programa
  cube.doThings();
  cube.actualice(renderer,scene,camera); //Actualizar a la generación n
  gen+=1//Conteo de las generaciones
  densn1=cube.getDens();//Densidad de la generación n

  //Arrojar los datos de densidad a la consola de Javascript.
  if (isNaN(densn1)){
    console.log(gen,0)
  }
  else{
    console.log(gen,densn1)
  }
}
animate();//Inicio del programa.
```

Figura 14: Script comentado que se encarga de coordinar los scripts anteriores.

### 3D.html

A pesar de que todo ya aparentemente funciona, como se está trabajando sobre un servidor web estos solo entienden el lenguaje de etiquetas html, es necesario implementar este segundo lenguaje para incluir el JavaScript. El código usado para esto se muestra a continuación:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Game of life 3D</title>
    <style>
      body{
        margin : 8px;
      }
    </style>
  </head>
  <body>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/r127/three.min.js"
    integrity="sha512-481987279749305584d273289144633b28044972106552284926833936247240"
    crossorigin="anonymous"></script> <!--Cargar la libreria Three.js-->
    <!--Cargar las librerias antes descritas-->
    <script src="Interfaz.js"></script>
    <script src="Cells.js"></script>
    <script src="Cube.js"></script>
    <script src="GOL3D.js"></script>
  </body>
</html>
```

Figura 15: Código en html que permite que el servidor web interprete el código de JavaScript.

### • Calcular variables termodinámicas como la temperatura, energía interna y entropía para sistemas con condiciones iniciales, definidas por el ensamble microcanónico.

Dado que el código realizado en JavaScript solo devuelve la densidad en términos de cada generación que pasa, se realizó un código en python, el cual calcula y grafica cada una de las variables termodinámicas. El código comentado se presenta a continuación:

```
#Se importan las librerías necesarias para todos los cálculos
import numpy as np
import matplotlib.pyplot as mp
from scipy.special import gamma

#Creación de la clase "Graf"
class Graf():
  #En python el constructor es __init__, el cual contiene lo que la clase "sabe".
  def __init__(self,Data,delim="\t",skip=0):
    #Toma los datos de la "base de datos" (Archivos de texto).
    self.x0,self.y0=np.loadtxt(Data,delimiter=delim,unpack=True,skiprows=skip)
  def plot(self,labeldat="",color="ro"):
    #Plotea los datos directamente, sin hacer ningún trato de estos.
    mp.plot(self.x0,self.y0,color,label=labeldat)
  #Inicio funciones estadísticas de las gráficas.
  def set_lim(self,xmin,xmax,ymin,ymax):
    mp.xlim(xmin,xmax)
    mp.ylim(ymin,ymax)
    return self
  def show(self):
    mp.legend()
    mp.show()
    return self
  def setLabels(self,xl,yl):
    mp.xlabel(xl)
    mp.ylabel(yl)
    return self
  #Fin funciones estadísticas de las gráficas.
  def getEntropy(self): #Calcula y devuelve el valor de la entropía.
    return -26*self.y0*mp.log(self.y0*(1-self.y0))-(4/2)*np.log(52*np.pi*self.y0*(1-self.y0)**53)
  def getT(self): #Calcula y devuelve el valor de la temperatura.
    #Calculo de la combinatoria a través de la función gamma
    comb=(gamma(27)/(gamma(26*self.y0+1)*gamma(26*(1-self.y0+1))))
    return comb*self.y0**(self.y0*26)*(1-self.y0)**(26*(1-self.y0))
```

```
def plotData(self,what,labelE="Entropía",colorE="ro"):
  #Plotea de los datos ya calculados dados los datos iniciales.
  if(what=="S"): #Plotea la entropía.
    y1=1.38*self.getEntropy()
  elif(what=="T"): #Plotea la temperatura.
    y1=1.38*self.getT()
    mp.ylim(0,0.3)
  elif(what=="E"): #Plotea la energía interna.
    y1=1.38*(3/2)*26*self.y0*self.getT()
  elif(what=="N"): #Plotea el número de partículas.
    y1=26*self.y0
  else:
    self.plot() #Si no se decide ninguno plotea los datos iniciales.
    return
  mp.plot(self.x0,y1,colorE,label=labelE) #Plotea lo que pase de los condicionales anteriores.
  #Método que pasa los datos calculados a un archivo .txt para hacer un tratamiento estadístico
  def getAData(self,Name,what):
    file=open(Name+".txt","w")
    if(what=="S"):
      y1=1.38*self.getEntropy()
    elif(what=="T"):
      y1=self.getT()
    elif(what=="E"):
      y1=1.38*(3/2)*26*self.y0*self.getT()
    elif(what=="N"):
      y1=26*self.y0
    for i in range(len(self.x0)):
      if(what=="T" or what=="E"):
        if(self.y0[i]!=0):
          file.write(str(self.x0[i])+"\t"+str(y1[i])+"\n")
        else:
          file.write(str(self.x0[i])+"\t"+str(y1[i])+"\n")
      else:
        file.write(str(self.x0[i])+"\t"+str(y1[i])+"\n")
    file.close()
  #FIN
```

Figura 16: Código para el manejo de datos para el cálculo de las variables termodinámicas del sistema.

### Obtención de los datos:

Al inicio el programa pedirá al usuario ingresar el valor de  $l$ , valor asociado a la cantidad de celdas totales (pues  $l^3$  es el número de celdas en el sistema), y seguido de esto pide la densidad inicial de celdas en estado “vivo” ( $\rho_0 = \rho(o)$ ), con lo cual el programa arrojará este par de mensajes a través del navegador:

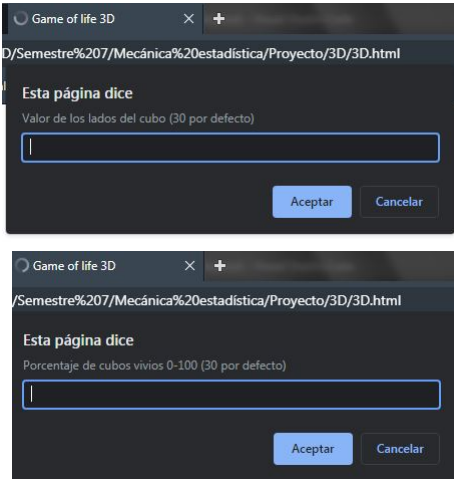


Figura 17: Inicialización de la simulación, dentro del navegador Opera GX.

Dado que la simulación es de un “juego” de cero jugadores, el sistema evolucionará automáticamente, sin embargo en cada paso discreto de la simulación arrojará el valor de la densidad en ese momento, es decir, para un tiempo discreto  $t_i$  el programa deberá arrojar a la consola el valor de  $\rho(t_i)$ , lo cuál se logra con el siguiente código:

```
//Arrojan los datos de densidad a la consola de Javascript.
if (isNaN(dens1)){
  console.log(gen,0)
}
else{
  console.log(gen,dens1)
}
```

Figura 18: Código que devuelve los datos en la consola.

Con el cual mientras el sistema no esté en equilibrio dará el valor de la densidad actual ( es decir la variable dens1), si en algún caso no hay vidas en el sistema entonces en la consola se verá le generación y el valor de la densidad cero, la función de Javascript que permite obtener los datos en la consola es la función “console.log()”. [9]. Los datos del sistema se muestran en la consola virtual del navegador que se esté usando, los datos se verán de una forma similar a la siguiente:

Generacion	// Densidad
1	0.28265332982364505
2	0.22196567213993382
3	0.22844508596277505
4	0.23665227429853397
5	0.2263361812751982
6	0.23145604395604377
7	0.22413137424373275
8	0.22749888298093414
9	0.22658896450892077
10	0.22374725437308865
11	0.22140835691693203
12	0.2267029972752068
13	0.21297443405789523
14	0.2133916930527151
15	0.20992233936396712
16	0.22074098894260963
17	0.2086204979991153
18	0.20916821425080528
19	0.20768098992718825

Figura 19: Datos obtenidos en la consola para la densidad en cada cambio discreto de tiempo. Dentro de la consola virtual del navegador Opera GX.

• **Comparar los resultados analíticos y computacionales de las variables termodinámicas y la densidad de celdas vivas en sus valores estables, al igual que contrastar los sistemas de 3 dimensiones frente a lo que se obtiene en 2 dimensiones.**

Calculo de cada una de las expresiones analíticas de los resultados expuestos en la tabla 4:

**Temperatura:** Es posible partir de la ecuación (4) para tres dimensiones llegar a la ecuación equivalente para la probabilidad en dos dimensiones, en el caso de 2 dimensiones la vecindad es de 8 celdas, con lo cual se tiene que:

$$P(h, t) = \binom{8}{h} \rho^h (1 - \rho)^{8-h} = T(h, t) \quad (25)$$

Luego, tomando  $h = 8\rho$  se tiene:

$$P(\rho) = \binom{8}{8\rho} \rho^{8\rho} (1 - \rho)^{8(1-\rho)} = T(\rho) \quad (26)$$

Expandiendo la combinatoria y usando la función gamma, se obtiene la siguiente expresión:

$$T(\rho) = \frac{8! \rho^{8\rho} (1 - \rho)^{8(1-\rho)}}{\Gamma(8\rho + 1) \Gamma(9 - 8\rho)} \quad (27)$$

Donde, teniendo el valor de  $\rho$  se encuentra la temperatura.

**Densidad:** Partiendo de la ecuación (25), la cual se cumple cuando el estado está en equilibrio, se tiene que:

$$\langle \sigma_\alpha \rangle = \binom{8}{3} \rho^3 (1 - \rho)^5 + \binom{8}{2} \rho^2 (1 - \rho)^6 = \rho \quad (28)$$

Encontrando que las soluciones a la ecuación anterior son  $\rho = 0$  y  $\rho \approx 0,37$ .

**Energía:** Análogo con los resultados obtenidos para los gases ideales, se define la energía interna del sistema de dos dimensiones (es decir  $D = 2$ ,  $yN = 8\rho$ ) como:

$$E = \frac{DN}{2}k_B T = \frac{2}{2}Nk_B T = 8\rho k_B T(\rho) \quad (29)$$

**Entropía:** Partiendo de la ecuación (14) y que  $S = k_B \ln(W)$ , se tiene que:

$$W_{2D} = \frac{8!}{(8\rho)!(8(1-\rho))!} \quad (30)$$

Usando la definición de la entropía y usando la función gamma, se tiene que:

$$S(\rho) = k_B \ln \left( \frac{8!}{\Gamma(8\rho+1)\Gamma(9-8\rho)} \right) \quad (31)$$

**El código que se muestra en la presente sección y los datos obtenidos se encontrarán adjunto al presente avance.**

## Bibliografía

- [ 1 ] The Game of Life and Other Cellular Automata. M. Yu, A. Reevesman. University of California Santa Barbara. January 28, 2015.
- [ 2 ] Candidates for the Game of Life in Three Dimensions. C. Bays. Department of Computer Science, University of South Carolina.
- [ 3 ] Statistical Mechanics of a Dynamical System Based on Conway's Game of Life. L. S. Schulman, P. E. Seiden. Journal of Statistical Physics, 1978.
- [ 4 ] Game of Life on the Equal Degree Random Lattice. Zhi-Gang Shao, T. Chen. Journal of Statistical Physics, 2010.
- [ 5 ] Adamatzky, A. (Ed.). (2010). Game of life cellular automata (Vol. 1). London: Springer.
- [ 6 ] Games, M. (1970). The fantastic combinations of John Conway's new solitaire game "life" by Martin Gardner. Scientific American, 223, 120-123.
- [ 7 ] Bays, C.: A note on the Game of Life in hexagonal and pentagonal tessellations. Complex Syst. 15, 245–252 (2005)
- [ 8 ] Notas de clase mecánica estadística, capítulo 3, sección 3.5.2 Entropía para el gas ideal de partículas libres.
- [ 9 ] Console.log() - Referencia de la Api Web — MDN, Mozilla, [sitio Web] <https://developer.mozilla.org/es/docs/Web/API/Console/log>