

# Reporte 2: Autómata Celular

Jorge Torres

20 de febrero de 2022

## 1. Objetivo

El objetivo de la práctica se centra en diseñar y ejecutar un experimento con por lo menos 30 réplicas para estimar la probabilidad de creación de vida dentro de 200 iteraciones, usando niveles de 10, 15, 20 y 25 para el tamaño de matriz y los niveles 0.2, 0.4, 0.6 y 0.8 para la densidad inicial de vida.

## 2. Desarrollo

Utilizando como base el [código](#) desarrollado por E. Schaeffer [1], para generar un autómata celular, primero se definen los tamaños de matriz, las densidades iniciales de vida, la cantidad de réplicas y las iteraciones que dura el experimento. Estos parámetros se observan en el código 1. En seguida se definen dos funciones, `mapeo` y `paso`, que sirven respectivamente para: 1) mapear la matriz en cuestión y 2) revisar la condición de vida de cada celda individual en la matriz, la cual consiste en tener exactamente tres vecinos vivos (ver código 2). Por último, se comienzan los ciclos `for` para iterar entre los parámetros definidos. En el ciclo donde se llama a la función `paso` se determina si el sistema termina en un estado de vida, para lo cual se lleva un contador que se usa posteriormente para determinar el porcentaje de supervivencia de acuerdo a la ecuación 1,

$$P_s = \frac{C_v}{R} \times 100 \quad (1)$$

donde  $P_s$  es el porcentaje de supervivencia,  $C_v$  es la cantidad de sistemas que lograron sobrevivir y  $R$  es la cantidad de veces que se replicó el experimento. Estas iteraciones se pueden apreciar en el código 3, mientras que el código completo se puede revisar en mi [repositorio](#) de GitHub.

```
dim = [10, 15, 20, 25] #matrix sizes
p = [0.2, 0.4, 0.6, 0.8] #initial life density
runs = 30 #replicas of the experiment
dur = 200 #iterations
```

Código 1: Parámetros

```
def mapeo(pos):
    fila = pos // lado
    columna = pos % lado
    return actual[fila, columna]

def paso(pos):
    fila = pos // lado
    columna = pos % lado
    vecindad = actual[max(0, fila - 1):min(lado, fila + 2),
                     max(0, columna - 1):min(lado, columna + 2)]
    return 1 * (np.sum(vecindad) - actual[fila, columna] == 3)
```

Código 2: Funciones

```

for lado in dim:
    num = lado**2
    for densidad in p:
        contador_viv=0
        for rep in range(runs):
            valores = [1 * (random() < densidad) for i in range(num)]
            actual = np.reshape(valores, (lado, lado))
            assert all([mapeo(x) == valores[x] for x in range(num)])
            for iteracion in range(dur):
                valores = [paso(y) for y in range(num)]
                vivos = sum(valores)
                if vivos == 0:
                    break;
                if iteracion == (dur-1):
                    contador_viv += 1
                    actual = np.reshape(valores, (lado, lado))
            vivieron = ((contador_viv*100)/(runs))
            resultados = {'Tamano_matriz': lado,
                          'P_inicial': densidad,
                          'Porcentaje_Supervivencia(%)': vivieron}
            datos = datos.append(resultados, ignore_index=True)

```

Código 3: Iteración de Parámetros

### 3. Resultados

En la figura 1 se observa el comportamiento del autómata celular conforme se varían el tamaño de matriz y la densidad inicial de vida.

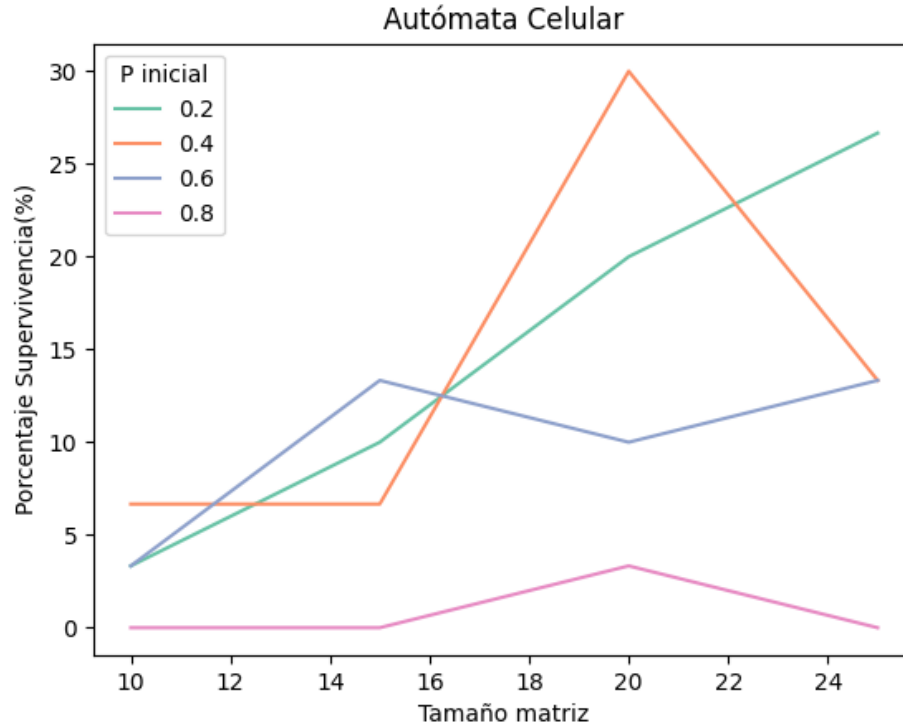


Figura 1: Porcentajes de supervivencia para sistemas con tamaños de matriz 10, 15, 20 y 25, y para densidades iniciales de vida de 0.2, 0.4, 0.6 y 0.8

## 4. Conclusiones

Una observación inicial indica que los sistemas tienen una mayor probabilidad de sobrevivir conforme se aumenta el tamaño de la matriz y para densidades iniciales de vida medias y bajas (0.2, 0.4). Esto puede deberse a que las células tendrían un mayor espacio para proliferar en un área más grande y a que no tendrían tanta competencia al haber menos densidad.

Algo que destaca es la baja o nula probabilidad de supervivencia de los sistemas con una densidad inicial de 0.8, esto debido a la condición de supervivencia de tener exactamente tres vecinos vivos. Podría cambiarse la condición para estimular un mejor o peor crecimiento, pero eso está fuera del rango de estudio de esta práctica.

## Referencias

- [1] E. Schaeffer. Cellularautomata. *Repositorio, GitHub*, 2020. URL <https://github.com/satuelisa/Simulation/tree/master/CellularAutomata>.