

# Autómata Celular

## Simulación de sistemas

Marco Antonio Guajardo Vigil

5 de febrero de 2019

### 1. Introducción

Un **Autómata Celular** [4] es un modelo *matemático* para un sistema dinámico, o sea, un sistema en constante movimiento, compuesto por un conjunto de celdas o células que adquieren distintos estados o valores, en este caso, 0 y 1.

Estos estados son alterados de un instante a otro en unidades de tiempo discreto, es decir, que se puede cuantificar con valores enteros a intervalos regulares. De esta manera este conjunto de células logran una evolución según una determinada expresión matemática, que es sensible a los estados de las células vecinas, y que se conoce como regla de transición local.

En la segunda práctica [5] trabajamos con autómatas celulares en dos dimensiones, particularmente el famoso *juego de la vida* [3]. El estado del autómata se representa con una matriz booleana (es decir, contiene ceros y unos como se mencionó anteriormente). Cada celda es o viva (uno) o muerta (cero). En cada paso, la supervivencia de cada celda se determina a partir de los valores de sus ocho vecinos que están a su alrededor.

La regla de supervivencia es sencilla: una celda está viva si exactamente tres vecinos suyos están vivos.

### 2. Implementación de R

Para la elaboración de este experimento, se hace uso de un software libre para computación estadística y gráficos llamado R [2], el cual nos permite realizar los cálculos necesarios para dicho experimento. Con él, se pueden controlar los datos estadísticos que se ocupan para dar seguimiento con la práctica, se necesita graficarlos para así poder compararlos mejor, ya que se maneja una cantidad de datos considerable y trabajaremos con ellos en forma estadística, por lo tanto, se recomienda el uso de este software ya que ayuda a paralelizar las acciones que sean necesarias, así se ahorra tiempo, haciéndolas simultáneamente.

### 3. Experimentación

Para llevar a cabo este experimento, se debe diseñar de tal forma en la que se pueda determinar el **número de iteraciones** que procede la simulación en una malla de 30 por 30 celdas hasta que se mueran todas, variando la *probabilidad inicial* de celda viva entre cero y uno, con pasos de **0.10**.

```

1 dim <- 30
2 num <- dim^2
3 lim <- 100 #Limite de pasos a realizar
4 repeticiones <- 30 #Veces en las que se repite el experimento
5 probabilidades <- seq(0,1,0.10)

```

Lo siguiente se hace con:

```

1 1 * (runif(num) < p)

```

Esta modificación permite controlar la probabilidad con la cual se generan las celdas de la matriz.

El experimento se repite 30 veces y se promedia las iteraciones obtenidas de acuerdo a sus probabilidades, para así, precisar las iteraciones dadas en cada probabilidad. Se hizo uso de script realizado en R, versión 3.5.2.

Se modifica el código de tal manera que realice una malla de 30 por 30 celdas, con un crecimiento de probabilidad desde 0 a 1 en pasos de 0.10 y 30 repeticiones para cada probabilidad. La condición para que una celda este viva, es que a su alrededor tenga exactamente 3 vecinos. Dicha condición se ve reflejada en la siguiente parte del código, línea **6**:

```

1 paso <- function(pos) {
2   fila <- floor((pos - 1) / dim) + 1
3   columna <- ((pos - 1) %% dim) + 1
4   vecindad <- actual[max(fila - 1, 1) : min(fila + 1, dim), max(columna - 1, 1) : min(columna + 1, dim)]
5   return(1 * ((sum(vecindad) - actual[fila, columna]) == 3))
6 }

```

## 4. Resultados

```

1 resultados <- rbind(resultados, iteraciones)

```

Los resultados se guardaron en un **rbind** como se muestra en la línea **1** del código para poder hacer uso de ellos, creando una gráfica con ayuda de la librería **ggplot2** y así poder graficar los datos obtenidos en relación de la probabilidad y sus iteraciones 1.

```

1 ggplot(data=resultados, aes(x=Prob, y=resultados$value, fill=Prob)) +
2   geom_bar(stat="identity", position="dodge") +
3   theme_gray(base_size = 14) + xlab("Probabilidad") + ylab("Iteraciones")
4 ggsave("GraficaP2.png")

```

Se observa que tanto en la probabilidad cero como en la uno, toma cero iteraciones en en terminar de realizar la condición, ya que, en la probabilidad cero, cada una tiene cero probabilidades de salir viva, por lo tanto, finaliza en esa misma iteración.

En la probabilidad uno, todas se generan vivas y esto hace que la condición no se pueda cumplir, todas las células estan rodeadas de células vivas, sin ningun peligro de llegar a morir, ya que todas las celdas tienen 8 celdas vecinas a su alrededor.

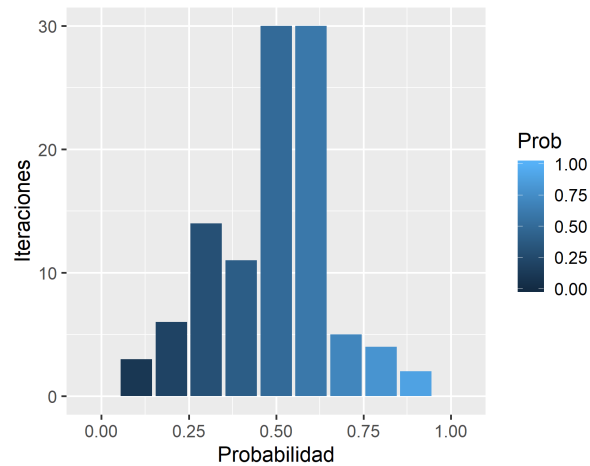


Figura 1: Gráfica en relación de probabilidades e iteraciones.

Cuando la probabilidad abarca los 0.50 hasta 0.80, la supervivencia de las celdas sobrepasa el límite de iteraciones establecido (100), por lo tanto, estas celdas no llegan a morir. El inicio de esta matriz 2 se encuentra animado con ayuda de Giphy [1], en la carpeta de la práctica 2 del repositorio.

## 5. Conclusión

De acuerdo a los datos obtenidos y mostrados en la gráfica:

1. Las iteraciones aumentan cuando la probabilidad llega a ser un poco más alta de la uniforme, con lo cual, le toma más iteraciones para que estas se mueran.
2. En casos donde la probabilidad llega a ser de cero y uno, toman la menor de las iteraciones posibles siendo estas de cero.

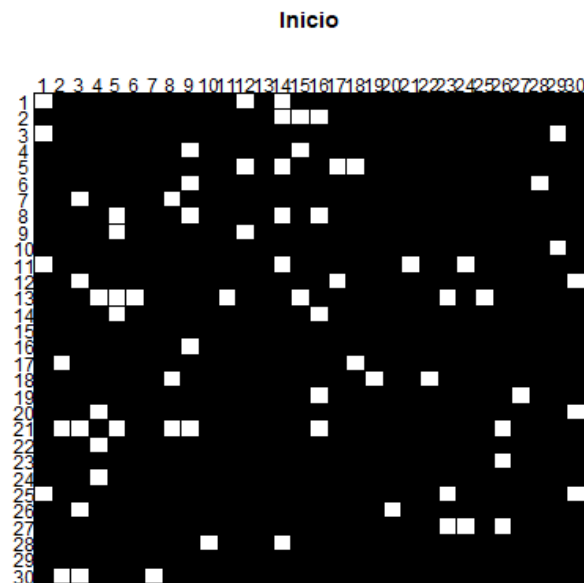


Figura 2: Matriz inicial con probabilidad de 0.80

## Referencias

- [1] Giphy. 2019. URL <https://giphy.com/create/gifmaker>.
- [2] The R Project for Statistical Computing. 2019. URL <https://www.r-project.org/>.
- [3] Eric Weisstein. Game of Life — Wolfram Mathworld. 2017. URL <http://mathworld.wolfram.com/GameofLife.html>.
- [4] Fernando Sancho Caparrini. Autómatas Celulares. 2016. URL <http://www.cs.us.es/~fsancho/?e=66>.
- [5] Satu Elisa Schaeffer. Práctica 2: Autómata Celular. 2019. URL <https://elisa.dyndns-web.com/teaching/comp/par/p2.html>.