

# Comportamiento Emergente

## El Juego de la Vida de Conway

Dpto. de Matemática Aplicada a la Ingeniería Aeroespacial

Informática, 1<sup>er</sup> semestre  
2022 – 2023

**Profesor:** Juan Antonio Hernández Ramos ([juanantonio.hernandez@upm.es](mailto:juanantonio.hernandez@upm.es))

**Coordinador:** Javier de Vicente Buendía ([fj.devicente@upm.es](mailto:fj.devicente@upm.es))

**Colaborador:** Víctor Javier Llorente Lázaro ([victorjavier.llorente@upm.es](mailto:victorjavier.llorente@upm.es))

# Índice

---

**Introducción**

**Algoritmo y código**

**Validación y resultados**

**Conclusiones**

# Introducción

**Comportamiento Emergente:** *"Se trata de aquel comportamiento complejo, o que da la impresión de inteligencia, que aparece espontáneamente y de forma imprevista, como consecuencia de una serie de acciones simples."*

$$\begin{cases} x(t + \Delta t) = f(x(t)), & \forall t \geq 0 \\ x(0) \text{ dado} \end{cases}$$



Figura 1. Bandada de aves [1]

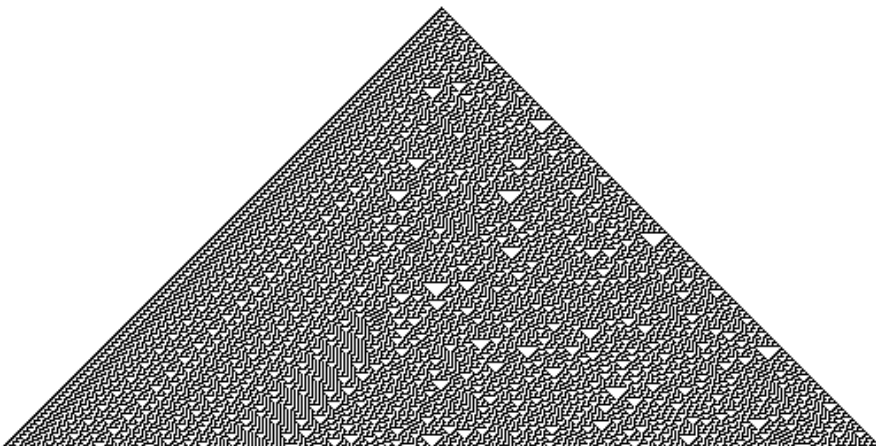


Figura 4. Autómata celular regla 30 [4]

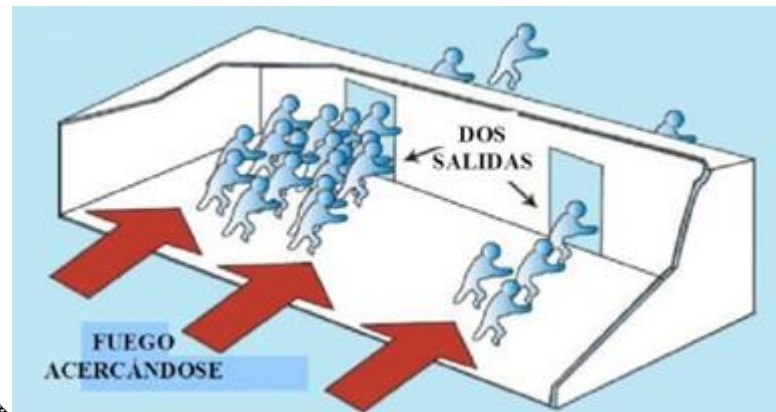


Figura 3. Comportamiento humano en incendios [3]



Figura 2. Hormigas en búsqueda de comida [2]

# Introducción

---

**El Juego de la Vida:** autómata celular diseñado por el matemático británico John Horton Conway en 1970

- Juego de cero jugadores
- El tablero del juego es una malla formada por cuadrados denominados células
- Cada célula tiene células vecinas de su alrededor
- Las células tienen dos estados: “vivas” o “muertas”
- El estado de las células evoluciona en tiempos discretos
- Cada célula sigue las siguientes reglas:
  1. Una célula muerta con exactamente 3 células vecinas vivas “nace”
  2. Una célula viva con 2 o 3 células vivas sigue viva, en otro caso muere

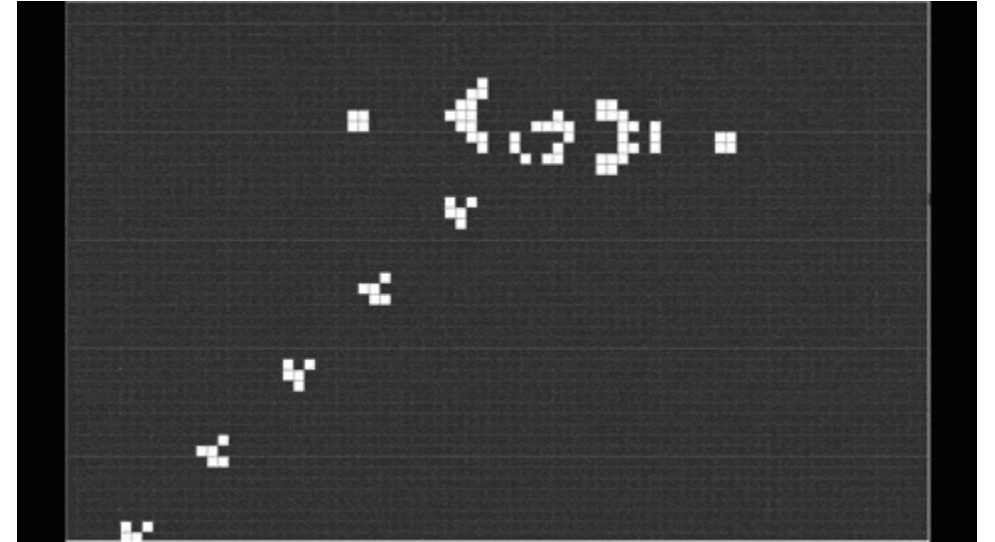


Figura 5. El Juego de la vida de Conway [5]

# Algoritmo y código

```
1. GENERACIONES: número de generaciones
2. N,M: tamaño del tablero
3. /* 0: célula muerta; 1: célula viva */
4.  $T_{ij} \leftarrow 0 \ \forall i,j \in \mathbb{D}$ 
5.  $T_{ij} \leftarrow 1 \ \forall i,j \in \mathbb{V}$ 
6. for generacion  $\leftarrow 0, \text{GENERACIONES}$  do
7.   write  $T_{ij}$ 
8.   for i  $\leftarrow 1, N$  do
9.     for j  $\leftarrow 1, M$  do
10.      |  $V_{ij} \leftarrow 1_{\text{adj}(T)_{ij}1_j} - T_{ij}$ 
11.    end for
12.  end for
13.   $T_{ij}^\dagger \leftarrow T_{ij}$ 
14.  for i  $\leftarrow 1, N$  do
15.    for j  $\leftarrow 1, M$  do
16.      if  $V_{ij} = 3 \vee (V_{ij} = 2 \wedge T_{ij}^\dagger = 1)$  then
17.        |  $T_{ij}^\dagger \leftarrow 1$ 
18.      else
19.        |  $T_{ij}^\dagger \leftarrow 0$ 
20.      end if
21.    end for
22.  end for
23.   $T_{ij} \leftarrow T_{ij}^\dagger$ 
24. end for
```

Interno

0	0	1
1		1
0	0	0

Cont. Sup.

1		1
1	1	0

Cont. Sup. Izq.

	1
0	0

Etc.



# Algoritmo y código

```
17 # Librería numérica
18 import numpy as np
19 # Librería de gráficos
20 import matplotlib.pyplot as plt
21 # Librería de colores
22 from matplotlib.colors import ListedColormap
```

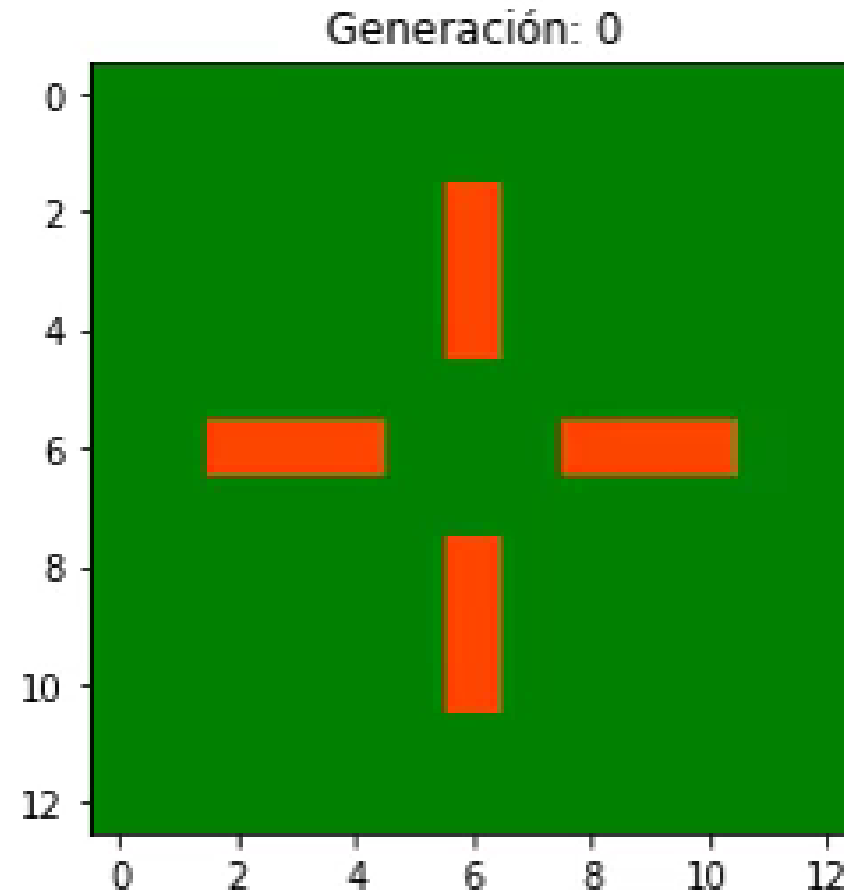
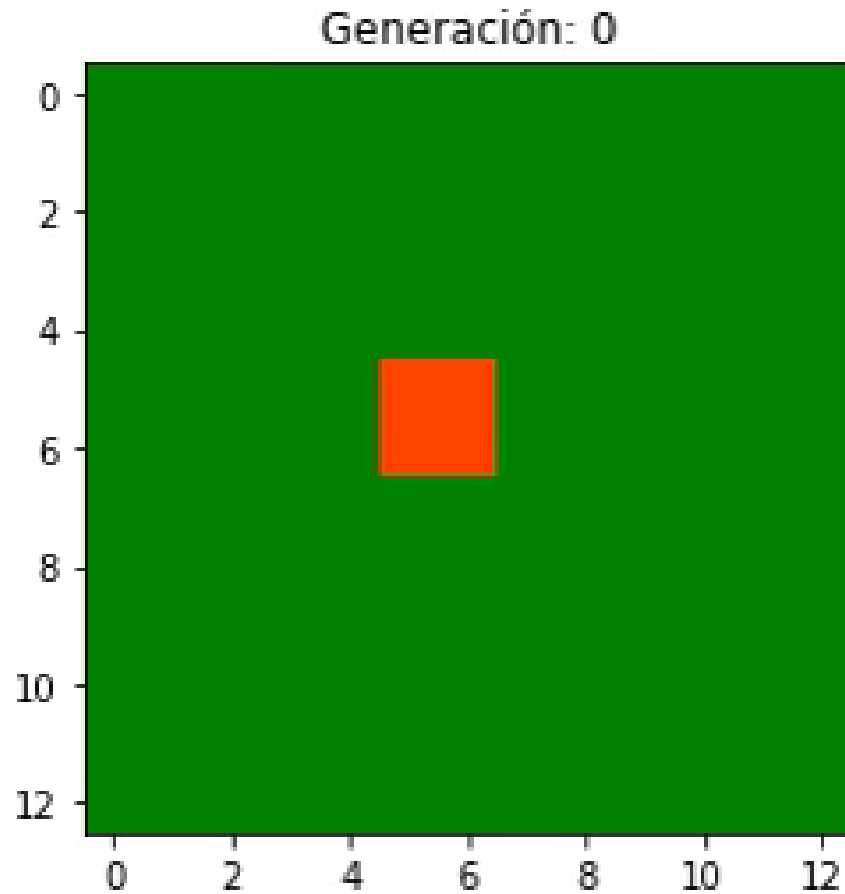
```
68 def paso(b):
69     """
70     Paso en el juego de la vida de Conway
71     """
72     v = vecindario(b)
73     bc = b[:, :]
74     N = bc.shape[0]
75     M = bc.shape[1]
76     for i in range(N):
77         for j in range(M):
78             if v[i, j] == 3 or (v[i, j] == 2 and bc[i, j] == 1):
79                 bc[i, j] = 1
80             else:
81                 bc[i, j] = 0
82     return bc
83
```

```
149 # Elección del problema
150 problema = "upm"
151
152 # Parámetros del problema
153 GENERACIONES = 60
154 N,M = dominio(problema)
155
156 # Construimos el tablero
157 tablero = np.zeros((N, M), dtype=int)
158
159 # Estado celular inicial
160 tablero = inicial(problema, tablero)
161
162 # Bucle
163 for generacion in range(GENERACIONES + 1):
164     imprimir(tablero, generacion)
165     tablero = paso(tablero)
```

```
30 def vecindario(b):
31     """
32     Array de células vivas en el vecindario
33     """
34     N = b.shape[0]
35     M = b.shape[1]
36     vecinos = np.zeros((N, M), dtype = int)
37     for i in range(N):
38         for j in range(M):
39             if (i == 0 and j == 0):
40                 # Esquina superior izquierda
41                 vecinos[i,j] = np.sum(b[i :i+2, j :j+2]) - b[i,j]
42             elif (i == N-1 and j == 0):
43                 # Esquina superior derecha
44                 vecinos[i,j] = np.sum(b[i-1:i , j :j+2]) - b[i,j]
45             elif (i == 0 and j == M-1):
46                 # Esquina inferior izquierda
47                 vecinos[i,j] = np.sum(b[i :i+2, j-1:j ] - b[i,j]
48             elif (i == N-1 and j == M-1):
49                 # Esquina inferior derecha
50                 vecinos[i,j] = np.sum(b[i-1:i , j-1:j ] - b[i,j]
51             elif (j == 0):
52                 # Contorno superior
53                 vecinos[i,j] = np.sum(b[i-1:i+2, j :j+2]) - b[i,j]
54             elif (i == 0):
55                 # Contorno izquierda
56                 vecinos[i,j] = np.sum(b[i :i+2, j-1:j+2]) - b[i,j]
57             elif (i == N-1):
58                 # Contorno derecha
59                 vecinos[i,j] = np.sum(b[i-1:i , j-1:j+2]) - b[i,j]
60             elif (j == M-1):
61                 # Contorno inferior
62                 vecinos[i,j] = np.sum(b[i-1:i+2, j-1:j ] - b[i,j]
63             else:
64                 # Interno
65                 vecinos[i,j] = np.sum(b[i-1:i+2, j-1:j+2]) - b[i,j]
66     return vecinos
```

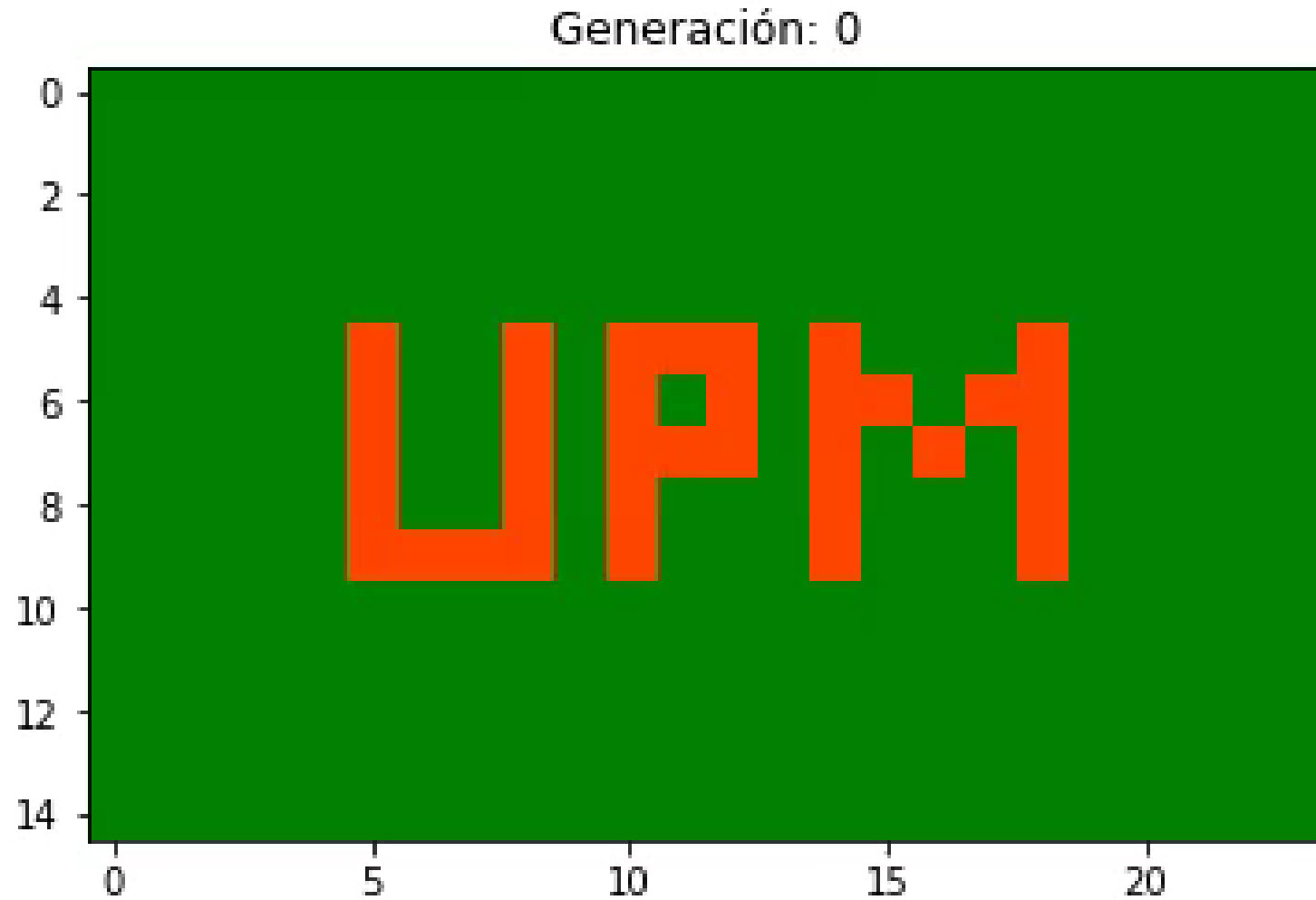
# Validación y resultados

---



# Validación y resultados

---





# Conclusiones

---

Que os ha aportado este trabajo

# Referencias

---

- [1] <https://www.investigacionyciencia.es/blogs/medicina-y-biologia/106/posts/el-comportamiento-de-bandada-en-las-aves-18226>
- [2] <https://vidaconmascotas.com/por-que-las-hormigas-caminan-en-fila-india-aqui-esta-la-verdad/>
- [3] <https://bajadesignengineeringblog.academy/2017/11/10/comportamiento-humano-en-incendios-2-premisas-importantes/>
- [4] Wolfram, S. (1983). Statistical mechanics of cellular automata. *Rev. Mod. Phys.* 55: 601-644.
- [5] Gardner, Martin (1970). Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American* 223: 120-123.