

# Practica No 5: Juego de la Vida

Nombre:

Fernando Sanchez

Enunciado:

**Emplear el programa del "Juego de la vida" de Jhon Conway para realizar el siguiente proceso de simulación :**

- Determinar las variables que rigen el sistemas
- Diseñar un plan de simulación que permita llegar a una configuración en que los autómatas celular no varíen.
- Diseñar un plan de pruebas automatizado, es decir, que sea controlado por el ordenador y que permita lanzar una bateria de experimentos.
- Recopilar los resultados de estos planes.

```
In [7]: 1 # Importar librerias necesarias
        2 import pygame
        3 import numpy as np
        4 import time
        5
        6 col_about_to_die = (200, 200, 225)
        7 col_alive = (255, 255, 215)
        8 col_background = (10, 10, 40)
        9 col_grid = (30, 30, 60)
       10
```

```

In [11]: 1 def update(surface, cur, sz):
2         # inicializado con ceros, las células muertas no están escritas
3         aux = np.zeros((cur.shape[0], cur.shape[1]))
4
5         for r, c in np.ndindex(cur.shape):
6             num_alive = np.sum(cur[r-1:r+2, c-1:c+2]) - cur[r, c]
7
8             if cur[r, c] == 1 and num_alive < 2 or num_alive > 3:
9                 time.sleep(0.5)
10                col = col_about_to_die
11            elif (cur[r, c] == 1 and 2 <= num_alive <= 3) or (cur[r, c] == 0 and num_alive == 1):
12                aux[r, c] = 1
13                col = col_alive
14
15            col = col if cur[r, c] == 1 else col_background
16            pygame.draw.rect(surface, col, (c*sz, r*sz, sz-1, sz-1))
17
18        return aux

```

```

In [9]: 1 def inicio(dim_X, dim_Y):
2
3         celdas = np.zeros((dim_Y, dim_X))
4         #Comienza en la esquina superior izquierda, continuando con el patrón
5         patron = np.array([[1,1,1],[0,1,0]]);
6         pos = (3,3)
7         celdas[pos[0]:pos[0]+patron.shape[0], pos[1]:pos[1]+patron.shape[1]] = patron
8         return celdas

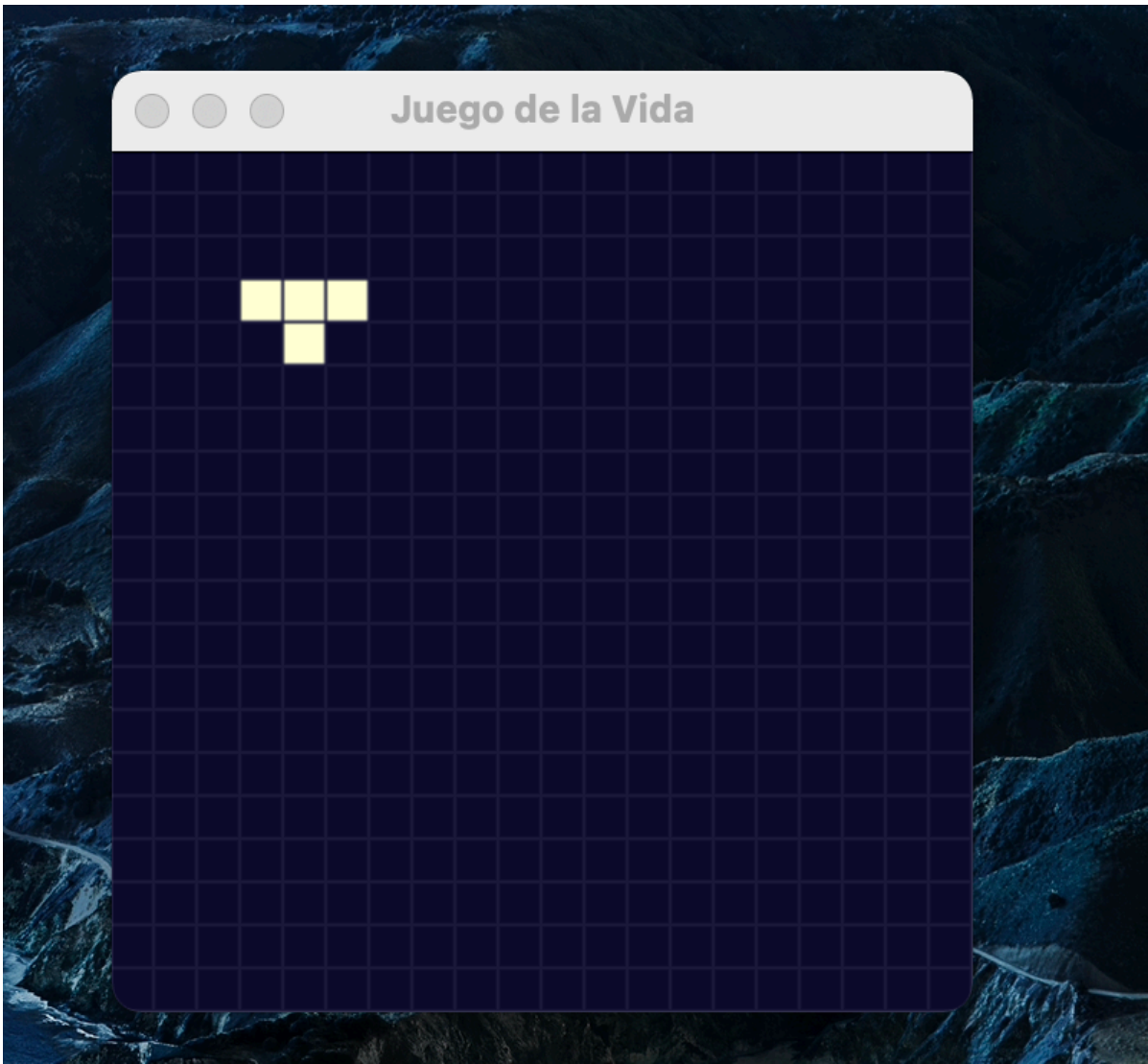
```

```

In [*]: 1 def main(dim_X, dim_Y, tamaño_Celda):
2         pygame.init()
3         surface = pygame.display.set_mode((dim_X * tamaño_Celda, dim_Y * tamaño_Celda))
4         pygame.display.set_caption("Juego de la Vida")
5
6         celdas = inicio(dim_X, dim_Y)
7
8         while True:
9             for event in pygame.event.get():
10                if event.type == pygame.QUIT:
11                    pygame.quit()
12                    return
13
14            surface.fill(col_grid)
15            celdas = update(surface, celdas, tamaño_Celda)
16            pygame.display.update()
17
18        if __name__ == "__main__":
19            main(20, 20, 15)

```

## Resultados





In [ ]:

1