

# Juego de la vida de Conway

El juego de la vida tiene su origen en el año de 1970 en la revista Scientific American; debido a la gran variedad que presentan la evolución de sus patrones este a adquirido una gran cantidad de atención dado que esta es equivalente a una maquina universal de Turing dado que todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida.

## Reglas del juego

Se trata de un juego de cero jugadores dado que el progreso o evolución del juego viene dado por un patron inicial el cual no requiere la intervención posterior de la persona. Ademas consta de un tablero en el cual cada celda representa una célula las cuales pueden estar en uno de dos posibles estados "vivas" o "muertas"; una célula puede ir intercambiando su estado en cada turno durante el progreso del juego, las células constan de las siguientes reglas para actuar:

- Una célula muerta con exactamente 3 células vecinas vivas "nace" (es decir, al turno siguiente estará viva).
- Una célula viva con 2 o 3 células vecinas vivas sigue viva, en otro caso muere (por "soledad" o "superpoblación").

Aquí tenemos un pequeño ejemplo en python de nuestro algoritmo para el juego de la vida:

```
In [1]: import matplotlib.pyplot as plt; plt.rcParamsdefaults()
import numpy as np
import matplotlib.pyplot as plt

class Game(object):

    def __init__(self, state, infinite_board = True):

        self.state = state
        self.width = state.width
        self.height = state.height
        self.infinite_board = infinite_board

    def step(self, count = 1):
        total_vivas = {}
        idx = 0
        for generation in range(count):
            print('Generacion:', generation)
            new_board = [[False] * self.width for row in range(self.height)]
            vivas = 0
            muertas = 0
            for y, row in enumerate(self.state.board):
                for x, cell in enumerate(row):
                    neighbours = self.neighbours(x, y)
                    previous_state = self.state.board[y][x]
                    should_live = neighbours == 3 or (neighbours == 2 and previous_state == True)
                    new_board[y][x] = should_live
                    if cell:
                        vivas += 1
                    else:
                        muertas += 1
            total_vivas[idx] = vivas
            idx += 1
            print('vivas:', vivas, 'muertas:', muertas)

        self.state.board = new_board
        total = []
```

```

        generaciones = []
        for i in total_vivas:
            total.append(total_vivas[i])
            generaciones.append(i)
        print(self.display())
        print('Celulas por generacion:', total_vivas)
        print('Total de celulas que han vivido:', np.sum(total))
        print('Generacion con mas celulas vivas:', np.argmax(total), ':', np.amax(total))
        print('Generacion con menos celulas vivas:', np.argmin(total), ':', np.amin(total))
        print('Promedio de celulas vivas por generacion:', np.mean(total))
        if generation >= count - 1:
            self.graficar(generaciones, total)

def neighbours(self, x, y):

    count = 0

    for hor in [-1, 0, 1]:
        for ver in [-1, 0, 1]:
            if not hor == ver == 0 and (self.infinite_board == True or (0 <= x + hor < self
                count += self.state.board[(y + ver) % self.height][(x + hor) % self.width]

    return count

def display(self):
    return self.state.display()

def graficar(self, generaciones, celulas):
    objects = generaciones
    y_pos = np.arange(len(objects))
    x_pos = celulas

    plt.bar(y_pos, x_pos, align='center', alpha=0.5)
    plt.xticks(y_pos, objects)
    plt.ylabel('Celulas')
    plt.title('Juego de la vida')

    plt.show()

class State(object):

    def __init__(self, positions, x, y, width, height):

        active_cells = []

        for y, row in enumerate(positions.splitlines()):
            for x, cell in enumerate(row.strip()):
                if cell == 'o':
                    active_cells.append((x,y))

        board = [[False] * width for row in range(height)]

        for cell in active_cells:
            board[cell[1] + y][cell[0] + x] = True

        self.board = board
        self.width = width
        self.height = height

    def display(self):

        output = ''

        for y, row in enumerate(self.board):
            for x, cell in enumerate(row):
                if self.board[y][x]:
                    output += ' o '
                else:
                    output += ' . '

```

```
output += '\n'

return output
```

Ahora ejecutamos nuestro algoritmo con un patrón de inicio dado:

```
In [2]: celula = """ 000.000
                0.000.0
                0.....0
                """

f = int(input("Ingrese el numero de filas: "))
c = int(input("Ingrese el numero de columnas: "))
tablero = input("Desea que el tablero se infinito (S/N)?")
n = int(input("Ingrese el numero de iteraciones: "))
t = False
if(tablero == 'S' or tablero == 's'):
    t = True
my_game = Game(State(celula, x = 7, y = 3, width = c, height = f), t)
print(my_game.display())
my_game.step(n)
```

```
Ingrese el numero de filas: 14
Ingrese el numero de columnas: 14
Desea que el tablero se infinito (S/N)?s
Ingrese el numero de iteraciones: 10
```

```
. . . . .
. . . . .
. . . . .
. . . . . 0 0 0 . 0 0 0 .
. . . . . 0 . 0 0 0 . 0 .
. . . . . 0 . . . . . 0 .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

```
Generacion: 0
vivas: 13 muertas: 183
```

```
. . . . .
. . . . .
. . . . . 0 . . . 0 . .
. . . . . 0 . 0 . 0 . 0 .
. . . . . 0 0 . 0 . 0 . 0 0
. . . . . 0 . 0 . 0 . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

```
Celulas por generacion: {0: 13}
Total de celulas que han vivido: 13
Generacion con mas celulas vivas: 0 : 13
Generacion con menos celulas vivas: 0 : 13
Promedio de celulas vivas por generacion: 13.0
Generacion: 1
vivas: 15 muertas: 181
```

```
. . . . .
. . . . .
. . . . . 0 . . . 0 . .
. . . . . 0 0 . 0 . 0 . 0 0
. . . . . 0 0 . 0 . 0 . 0 0
. . . . . 0 0 0 0 0 0 0 .
. . . . .
```

```
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 13, 1: 15}  
Total de celulas que han vivido: 28  
Generacion con mas celulas vivas: 1 : 15  
Generacion con menos celulas vivas: 0 : 13  
Promedio de celulas vivas por generacion: 14.0  
Generacion: 2  
vivas: 21 muertas: 175

```
. . . . .
. . . . .
. . . . . 0 0 . . 0 0 .
. . . . . 0 . . 0 . 0 . 0
. . . . .
. . . . . 0 0 . 0 . 0 . 0 0
. . . . . . 0 0 0 0 0 . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 13, 1: 15, 2: 21}  
Total de celulas que han vivido: 49  
Generacion con mas celulas vivas: 2 : 21  
Generacion con menos celulas vivas: 0 : 13  
Promedio de celulas vivas por generacion: 16.333333333333332  
Generacion: 3  
vivas: 19 muertas: 177

```
. . . . .
. . . . .
. . . . . 0 0 . . 0 0 .
. . . . . 0 0 . . 0 0 .
. . . . . 0 0 0 . . 0 0 0
. . . . . 0 . 0 . 0 . 0 .
. . . . . 0 0 0 . 0 0 0 .
. . . . . . 0 0 0 . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19}  
Total de celulas que han vivido: 68  
Generacion con mas celulas vivas: 2 : 21  
Generacion con menos celulas vivas: 0 : 13  
Promedio de celulas vivas por generacion: 17.0  
Generacion: 4  
vivas: 27 muertas: 169

```
. . . . .
. . . . .
. . . . . 0 0 . . 0 0 .
. . . . . . 0 . 0 . . .
. . . . . 0 . . 0 . 0 . 0
. . . . . . 0 . 0 . . .
. . . . . 0 . . . . 0 .
. . . . . . 0 . 0 . . .
. . . . . . 0 . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27}

Total de células que han vivido: 95

Generacion con mas celulas vivas: 4 : 27

Generacion con menos celulas vivas: 0 : 13

Promedio de celulas vivas por generacion: 19.0

Generacion: 5

vivas: 17 muertas: 179

A 20x20 grid of dots. The dots are arranged in a pattern that forms a lowercase letter 'o'. The 'o' is centered in the grid, with its top and bottom dots missing. The grid is 20 dots wide and 20 dots high. The 'o' is formed by removing dots from the grid, leaving a hollow shape. The top and bottom dots of the 'o' are missing, and the dots forming the 'o' are also missing from the grid.

Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27, 5: 17}

Total de células que han vivido: 112

Generacion con mas celulas vivas: 4 : 27

Generacion con menos celulas vivas: 0 : 13

Promedio de células vivas por generación: 18.666666666666668

Generacion: 6

vivas: 16 muertas: 180

A 10x10 grid of dots. The dots are arranged in a regular pattern. Some dots are replaced by the letter 'O'. The 'O's are located at the following (row, column) positions (assuming the top-left dot is at (0,0) and the bottom-right dot is at (9,9)):

- (1, 5)
- (1, 8)
- (2, 4)
- (2, 5)
- (2, 6)
- (2, 7)
- (2, 8)
- (2, 9)
- (3, 4)
- (3, 5)
- (3, 6)
- (3, 7)
- (3, 8)
- (3, 9)
- (4, 4)
- (4, 5)
- (4, 6)
- (4, 7)
- (4, 8)
- (4, 9)
- (5, 5)
- (5, 6)
- (5, 7)
- (5, 8)
- (5, 9)
- (6, 5)
- (6, 6)
- (6, 7)
- (6, 8)
- (6, 9)
- (7, 5)
- (7, 6)
- (7, 7)
- (7, 8)
- (7, 9)
- (8, 5)
- (8, 6)
- (8, 7)
- (8, 8)
- (8, 9)
- (9, 5)
- (9, 6)
- (9, 7)
- (9, 8)
- (9, 9)

Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27, 5: 17, 6: 16}

Total de células que han vivido: 128

Generacion con mas celulas vivas: 4 : 27

Generacion con menos celulas vivas: 0 : 13

Promedio de células vivas por generación: 18.285714285714285

Generacion: 7

vivas: 18 muertas: 178

Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27, 5: 17, 6: 16, 7: 18}

Total de células que han vivido: 146

Generacion con mas celulas vivas: 4 : 27

Generacion con mas celulas vivas: 0 : 12  
Generacion con menos celulas vivas: 0 : 13

Promedio de células vivas por generación: 18.25

Generacion: 8

vivas: 18 muertas: 178

• • • • •

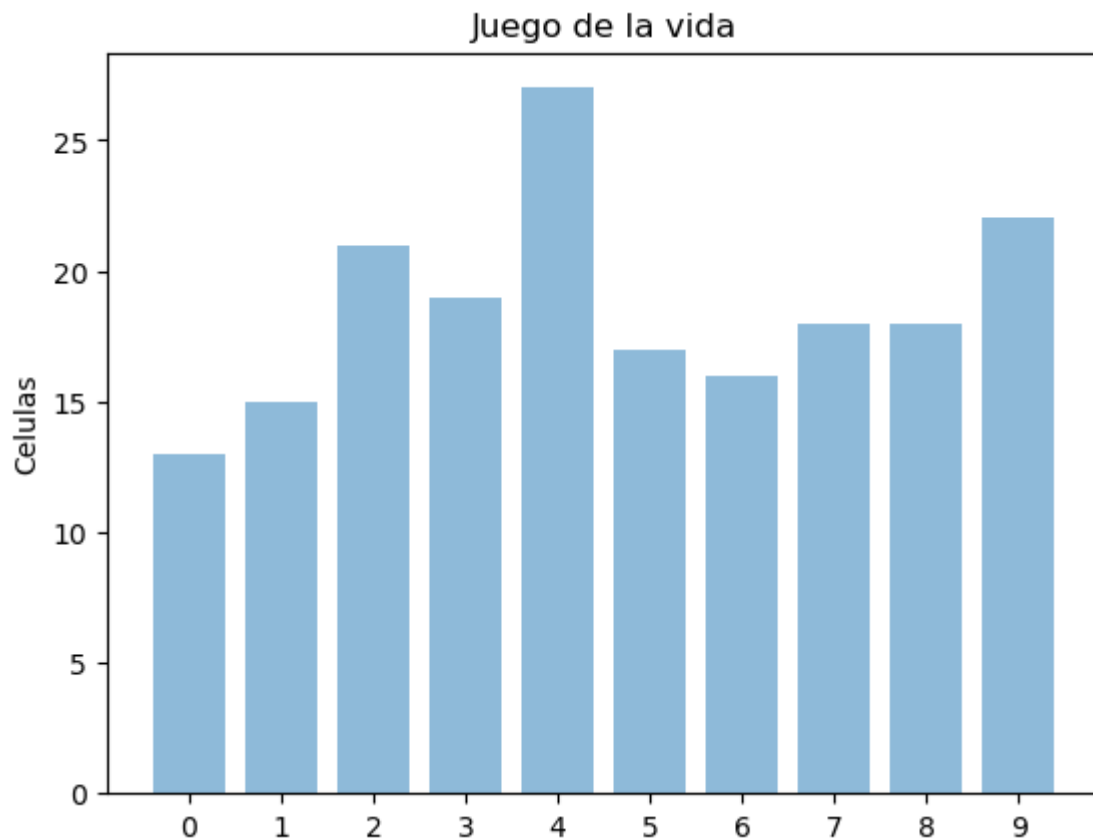
```
Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27, 5: 17, 6: 16, 7: 18, 8: 18}
Total de celulas que han vivido: 164
Generacion con mas celulas vivas: 4 : 27
Generacion con menos celulas vivas: 0 : 13
Promedio de celulas vivas por generacion: 18.22222222222222
Generacion: 9
vivas: 22 muertas: 174
```

A 15x15 grid of dots. The dots are arranged in a regular pattern. Some dots are replaced by the letter 'O'. The 'O's are located at the following (row, column) positions (starting from 0,0 at the top-left): (1, 6), (1, 7), (1, 11), (1, 12), (2, 5), (2, 8), (2, 10), (2, 13), (3, 2), (3, 4), (3, 7), (3, 9), (3, 12), (4, 1), (4, 3), (4, 6), (4, 8), (4, 11), (4, 14), (5, 4), (5, 7), (5, 9), (5, 12), (6, 5), (6, 6), (6, 10), (6, 11), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7), (7, 8), (7, 9), (7, 10), (7, 11), (7, 12), (7, 13), (7, 14), (7, 15), (8, 1), (8, 2), (8, 3), (8, 4), (8, 5), (8, 6), (8, 7), (8, 8), (8, 9), (8, 10), (8, 11), (8, 12), (8, 13), (8, 14), (8, 15), (9, 1), (9, 2), (9, 3), (9, 4), (9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (9, 10), (9, 11), (9, 12), (9, 13), (9, 14), (9, 15), (10, 1), (10, 2), (10, 3), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (10, 11), (10, 12), (10, 13), (10, 14), (10, 15), (11, 1), (11, 2), (11, 3), (11, 4), (11, 5), (11, 6), (11, 7), (11, 8), (11, 9), (11, 10), (11, 11), (11, 12), (11, 13), (11, 14), (11, 15), (12, 1), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12, 7), (12, 8), (12, 9), (12, 10), (12, 11), (12, 12), (12, 13), (12, 14), (12, 15), (13, 1), (13, 2), (13, 3), (13, 4), (13, 5), (13, 6), (13, 7), (13, 8), (13, 9), (13, 10), (13, 11), (13, 12), (13, 13), (13, 14), (13, 15), (14, 1), (14, 2), (14, 3), (14, 4), (14, 5), (14, 6), (14, 7), (14, 8), (14, 9), (14, 10), (14, 11), (14, 12), (14, 13), (14, 14), (14, 15).

```

Celulas por generacion: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27, 5: 17, 6: 16, 7: 18, 8: 18, 9: 22}
Total de celulas que han vivido: 186
Generacion con mas celulas vivas: 4 : 27
Generacion con menos celulas vivas: 0 : 13
Promedio de celulas vivas por generacion: 18.6

```



Luego volvemos a ejecutar nuestro código, sin embargo en este caso usamos solamente la mitad del patrón de células establecido en el punto anterior:

In [3]:

```
celula = """ 000
              0.0
              0.0"""

f = int(input("Ingrese el numero de filas: "))
c = int(input("Ingrese el numero de columnas: "))
tablero = input("Desea que el tablero se infinito (S/N)?")
n = int(input("Ingrese el numero de iteraciones: "))
t = False
if(tablero == 'S' or tablero == 's'):
    t = True
my_game = Game(State(celula, x = 7, y = 3, width = c, height = f), t)
print(my_game.display())
my_game.step(n)
```

Ingrese el numero de filas: 14  
Ingrese el numero de columnas: 14  
Desea que el tablero se infinito (S/N)?s  
Ingrese el numero de iteraciones: 10

```
. . . . .
. . . . .
. 0 0 0 . . . . .
. 0 . 0 . . . . .
. 0 . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Generacion: 0  
vivas: 6 muertas: 190

```
. . . . .
. . 0 . . . . .
. 0 . 0 . . . . .
0 0 . 0 . . . . .
. . 0 . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 6}  
Total de celulas que han vivido: 6  
Generacion con mas celulas vivas: 0 : 6  
Generacion con menos celulas vivas: 0 : 6  
Promedio de celulas vivas por generacion: 6.0  
Generacion: 1  
vivas: 7 muertas: 189

```
. . . . .
. . 0 . . . . .
0 0 . 0 . . . . .
0 0 . 0 . . . . .
. 0 0 . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 6, 1: 7}  
Total de celulas que han vivido: 13  
Generacion con mas celulas vivas: 1 : 7  
Generacion con menos celulas vivas: 0 : 6  
Promedio de celulas vivas por generacion: 6.5  
Generacion: 2  
vivas: 9 muertas: 187

```
. . . . .
. 0 0 . . . . .
0 . . 0 . . . . .
. . . 0 . . . . .
0 0 0 . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 6, 1: 7, 2: 9}  
Total de celulas que han vivido: 22  
Generacion con mas celulas vivas: 2 : 9  
Generacion con menos celulas vivas: 0 : 6  
Promedio de celulas vivas por generacion: 7.333333333333333  
Generacion: 3  
vivas: 8 muertas: 188

```
. . . . .
. 0 0 . . . . .
. 0 . 0 . . . . .
0 . . 0 . . . . .
. 0 0 . . . . .
. 0 . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8}  
Total de celulas que han vivido: 30  
Generacion con mas celulas vivas: 2 : 9  
Generacion con menos celulas vivas: 0 : 6  
Promedio de celulas vivas por generacion: 7.5  
Generacion: 4  
vivas: 9 muertas: 187

```
. . . . .
. 0 0 . . . . .
0 0 . 0 . . . . .
0 . . 0 . . . . .
0 0 0 . . . . .
. 0 0 . . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9}  
Total de celulas que han vivido: 39  
Generacion con mas celulas vivas: 2 : 9  
Generacion con menos celulas vivas: 0 : 6  
Promedio de celulas vivas por generacion: 7.8  
Generacion: 5  
vivas: 12 muertas: 184

```
. . . . .
```



```

Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9, 5: 12}
Total de celulas que han vivido: 51
Generacion con mas celulas vivas: 5 : 12
Generacion con menos celulas vivas: 0 : 6
Promedio de celulas vivas por generacion: 8.5
Generacion: 6
vivas: 11 muertas: 185

```

```

Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9, 5: 12, 6: 11}
Total de celulas que han vivido: 62
Generacion con mas celulas vivas: 5 : 12
Generacion con menos celulas vivas: 0 : 6
Promedio de celulas vivas por generacion: 8.857142857142858
Generacion: 7
vivas: 18 muertas: 178

```

```

Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9, 5: 12, 6: 11, 7: 18}
Total de celulas que han vivido: 80
Generacion con mas celulas vivas: 7 : 18
Generacion con menos celulas vivas: 0 : 6
Promedio de celulas vivas por generacion: 10.0
Generacion: 8
vivas: 11 muertas: 185

```

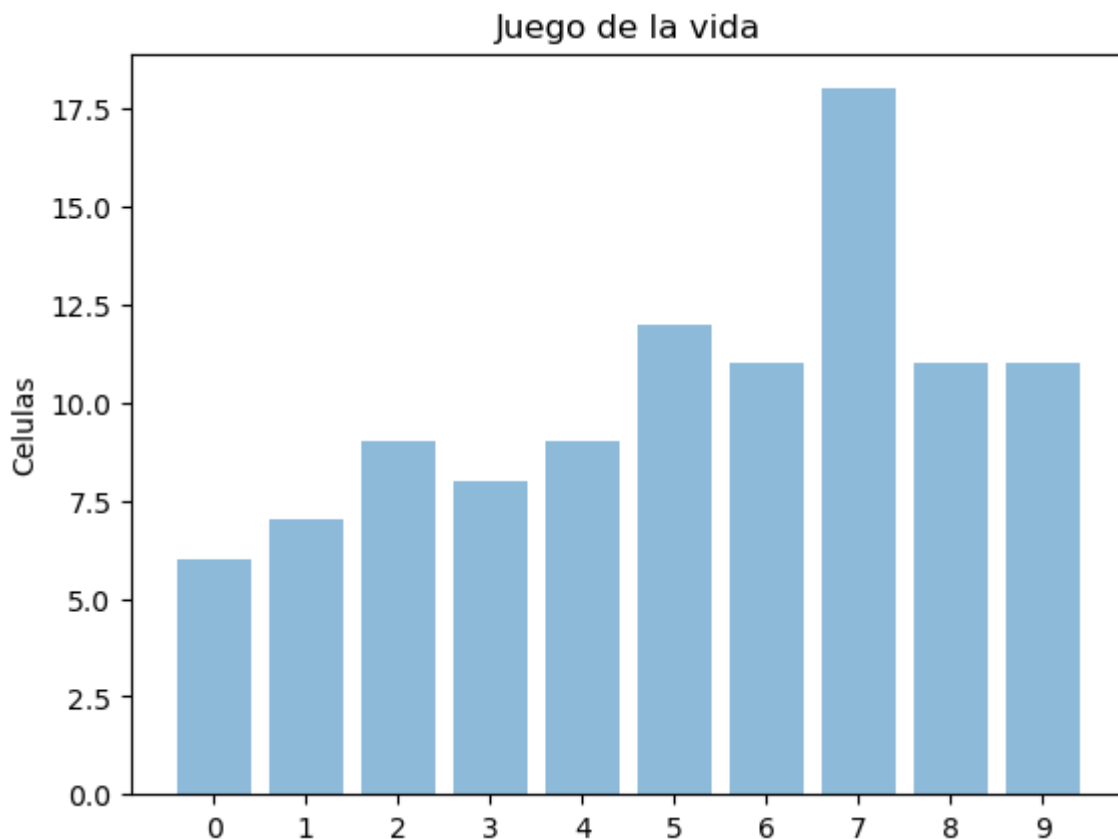
A 10x10 grid of dots. The dots are arranged in 10 rows and 10 columns. Some dots are replaced by the letter 'O'. The 'O's are located at the following (row, column) positions: (1,1), (1,2), (1,3), (2,2), (2,3), (2,4), (3,3), (4,3), (4,4), (4,5). All other positions contain a dot.

```
Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9, 5: 12, 6: 11, 7: 18, 8: 11}
Total de celulas que han vivido: 91
Generacion con mas celulas vivas: 7 : 18
Generacion con menos celulas vivas: 0 : 6
Promedio de celulas vivas por generacion: 10.111111111111111
Generacion: 9
vivas: 11 muertas: 185
```

```

Celulas por generacion: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9, 5: 12, 6: 11, 7: 18, 8: 11, 9: 11}
Total de celulas que han vivido: 102
Generacion con mas celulas vivas: 7 : 18
Generacion con menos celulas vivas: 0 : 6
Promedio de celulas vivas por generacion: 10.2

```



Luego de haber ejecutado nuestro código podemos ver un pequeño resumen en el cual nos enseña la cantidad de generaciones que se generaron, también la cantidad de células que han vivido en cada generación, así como el promedio de células que han vivido y las generaciones que mas células tuvieron y las que menos.

Como podemos apreciar en nuestro programa hay una gran diferencia entre ambos patrones del juego, en el primer patrón que ejecutamos obtuvimos la siguiente salida:

Células por generación: {0: 13, 1: 15, 2: 21, 3: 19, 4: 27, 5: 17, 6: 16, 7: 18, 8: 18, 9: 22}  
Total de células que han vivido: 186  
Generación con mas células vivas: 4 : 27  
Generación con menos células vivas: 0 : 13  
Promedio de células vivas por generación: 18.6  
Mientras que para el segundo patrón obtuvimos la siguiente salida:

Células por generación: {0: 6, 1: 7, 2: 9, 3: 8, 4: 9, 5: 12, 6: 11, 7: 18, 8: 11, 9: 11}  
Total de células que han vivido: 102  
Generación con mas células vivas: 7 : 18  
Generación con menos células vivas: 0 : 6  
Promedio de células vivas por generación: 10.2

- Células por generación:

{0: 7, 1: 8, 2: 12, 3: 11, 4: 21, 5: 5, 6: 5, 7: 0, 8: 7, 9: 11}

- Promedio de células vivas por generación:

$$18.6 - 10.2 = 8.4$$

```
In [4]: import numpy as np
import matplotlib.pyplot as plt

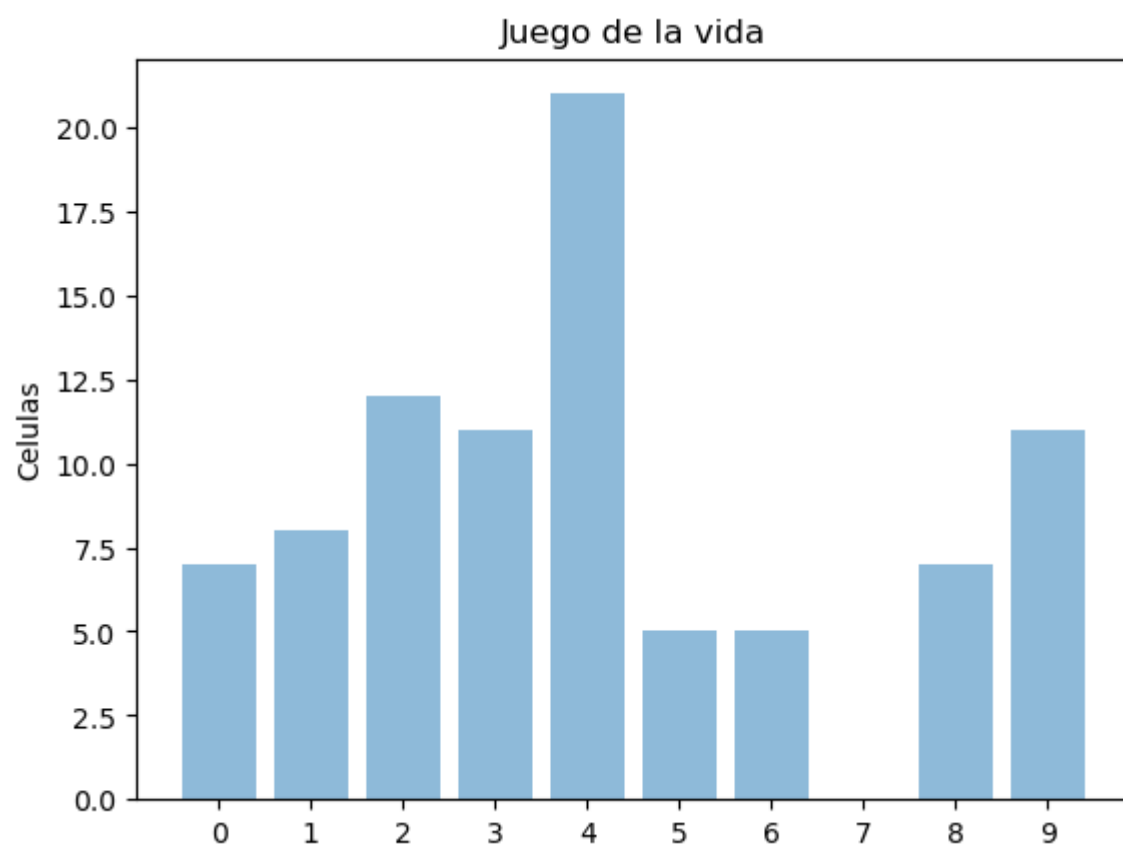
diff = {0: 7, 1: 8, 2: 12, 3: 11, 4: 21, 5: 5, 6: 5, 7: 0, 8: 7, 9: 11}
generaciones = []
celulas = []

for i in diff:
    generaciones.append(i)
    celulas.append(diff[i])

def graficar(generaciones, celulas):
    objects = generaciones
    y_pos = np.arange(len(objects))
    x_pos = celulas

    plt.bar(y_pos, x_pos, align='center', alpha=0.5)
    plt.xticks(y_pos, objects)
    plt.ylabel('Celulas')
    plt.title('Juego de la vida')
    plt.show()

graficar(generaciones, celulas)
```



In [ ]: