



**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO**



---

**FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN**

Nombre del curso:  
**ESTRUCTURA DE DATOS**

Profesor:  
**Jesús Hernández Cabrera**

Tarea:  
**T6 Juego de la Vida**

Alumno:  
**Navarro Rocha Miguel Ángel**

Fecha:  
**17-09-24**

## Juego funcionando:

```
"C:\Program Files\Java\jdk-17\bin\java."
Generación: 1
, , , , , ,
, 0, 0, 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,
, , , , , ,
, , , , , ,

Generación: 2
, , 0, , , ,
, 0, 0, 0, , ,
, 0, 0, 0, , ,
, , , , , ,
, , , , , ,
, , , , , ,
, , , , , ,

Generación: 3
, 0, 0, 0, , ,
, , , , , ,
, 0, , 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,
, , , , , ,

Generación: 4
, , 0, , , ,
, , 0, , 0, , ,
, , 0, , , ,
, , 0, , , ,
, , , , , ,
, , , , , ,
, , , , , ,

Generación: 5
, , 0, , , ,
, , 0, , 0, , ,
, , 0, 0, 0, , ,
, , , , , ,
, , , , , ,
, , , , , ,
, , , , , ,

Generación: 6
, , 0, , , ,
, , 0, , 0, , ,
, , 0, , 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,
, , , , , ,
```

```
Generación: 7
, , 0, , , ,
, 0, , 0, , ,
, 0, , 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,

Generación: 8
, , 0, , , ,
, 0, , 0, , ,
, 0, , 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,

Generación: 9
, , 0, , , ,
, 0, , 0, , ,
, 0, , 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,

Generación: 10
, , 0, , , ,
, 0, , 0, , ,
, 0, , 0, , ,
, , 0, , , ,
, , , , , ,
, , , , , ,
```

## Código main:

```
package ico.fesa.unam.mx.arreglos;

public class JuegoDeLaVida {
    private Array2d rejillaActual;
    private Array2d rejillaSiguiente;
    private int filas;
    private int columnas;

    public JuegoDeLaVida(int filas, int columnas) {
        this.filas = filas;
        this.columnas = columnas;
        this.rejillaActual = new Array2d(filas, columnas);
        this.rejillaSiguiente = new Array2d(filas, columnas);
        this.rejillaActual.clear(' '); // Inicialmente, todas las células están muertas
    }

    // Inicializar con una configuración de población
    public void inicializar(int[][] posicionesVivas) {
        for (int[] pos : posicionesVivas) {
            int ren = pos[0];
            int col = pos[1];
            rejillaActual.setItem(ren, col, 'O'); // 'O' representa una célula viva
        }
    }

    // Cuenta el número de vecinos vivos de una célula en una posición dada
    private int contarVecinosVivos(int ren, int col) {
        int[] dRow = {-1, -1, -1, 0, 0, 1, 1, 1};
        int[] dCol = {-1, 0, 1, -1, 1, -1, 0, 1};
        int count = 0;

        for (int i = 0; i < 8; i++) {
            int newRow = ren + dRow[i];
            int newCol = col + dCol[i];
            if (newRow >= 0 && newRow < filas && newCol >= 0 && newCol < columnas &&
                rejillaActual.getItem(newRow, newCol) == 'O') {
                count++;
            }
        }
        return count;
    }

    // Calcula la siguiente generación según las reglas del juego
    public void siguienteGeneracion() {
        for (int i = 0; i < filas; i++) {
            for (int j = 0; j < columnas; j++) {
                int vecinosVivos = contarVecinosVivos(i, j);
                char estadoActual = rejillaActual.getItem(i, j);

                // Aplicar las reglas del juego
                if (estadoActual == 'O') { // Célula viva
                    if (vecinosVivos < 2 || vecinosVivos > 3) {
                        rejillaSiguiente.setItem(i, j, ' '); // Muere por soledad o sobrepoblación
                    } else {
                        rejillaSiguiente.setItem(i, j, 'O'); // Sobrevive
                    }
                } else { // Célula muerta
                    if (vecinosVivos == 3) {

```

```

        rejillaSiguiente.setItem(i, j, 'O'); // Nacimiento
    } else {
        rejillaSiguiente.setItem(i, j, ' '); // Permanece muerta
    }
}
}
// Actualizar rejilla actual
Array2d temp = rejillaActual;
rejillaActual = rejillaSiguiente;
rejillaSiguiente = temp;
rejillaSiguiente.clear(' '); // Limpiar para la siguiente generación
}

// Ejecutar el juego durante un número específico de generaciones
public void jugar(int generaciones) {
    for (int i = 0; i < generaciones; i++) {
        System.out.println("Generación: " + (i + 1));
        System.out.println(rejillaActual);
        siguienteGeneracion();
    }
}

public static void main(String[] args) {
    JuegoDeLaVida juego = new JuegoDeLaVida(6, 6);
    int[][] posicionesVivas = {
        {1, 1}, {1, 2}, {1, 3}, // Configuración inicial de células vivas
        {2, 2}
    };
    juego.inicializar(posicionesVivas);
    juego.jugar(10); // Jugar x generaciones
}
}

```

## Código Array:

```

package ico.fesa.unam.mx.arreglos;

import java.util.Arrays;

public class Array2d {
    Character [][]data;
    int rowSize;
    int colSize;

    public Array2d(int ren, int col) {
        this.rowSize = ren;
        this.colSize = col;
        this.data = new Character[ren][col];
    }

    public void clear(Character dato){
        for (int i = 0; i < this.rowSize; i++) {
            for (int j = 0; j < this.colSize; j++) {
                this.data[i][j] = dato;
            }
        }
    }

    public int getRowSize() {

```

```

        return rowSize;
    }

    public int getColSize() {
        return colSize;
    }

    @Override
    public String toString() {
        String str = "";

        for (int i = 0; i < this.rowSize; i++) {
            for (int j = 0; j < this.colSize; j++) {
                str = str + this.data[i][j] + ", ";
            }
            str = str + "\n";
        }
        return str;
    }

    public void setItem(int ren, int col, Character dato){
        if (ren>=0 && ren < this.rowSize && col >= 0 && col < this.colSize){
            this.data[ren][col]=dato;
        }else{
            System.out.println("Indices fuera de rango");
        }
    }

    public Character getItem(int ren, int col){
        if (ren>=0 && ren < this.rowSize && col >= 0 && col < this.colSize){
            return this.data[ren][col];
        }else{
            System.out.println("Indices fuera de rango");
        }
        return '\0';
    }
}

```