



**Universidad Nacional Autónoma de
México**



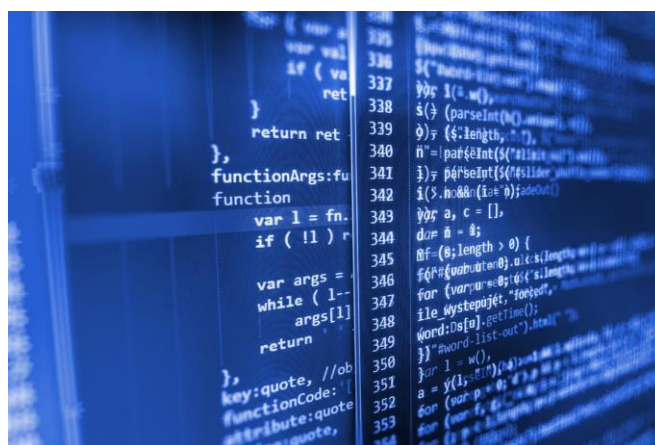
Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Estructura de Datos

Jesús Hernández Cabrera

Axel Yahir Moreno Rodríguez



Turno Vespertino

Grupo 1360

```
1 package unam.mx;
2
3 import java.util.Random;
4 import java.util.Scanner;
5
6 public class Arreglos2dMain {
7
8     Run | Debug
9     Scanner teclado = new Scanner(System.in); Resource leak: 'teclado' is never closed
10
11     int filas = 6;
12     int columnas = 6;
13     Array2d rejilla = new Array2d(filas, columnas);
14
15     inicializarAleatoriamente(rejilla);
16
17     System.out.println(x:"Estado inicial:");
18     System.out.println(rejilla);
19
20     System.out.print(s:"Introduce el número de generaciones a simular: ");
21     int generaciones = teclado.nextInt();
22
23     for (int i = 0; i < generaciones; i++) {
24         siguienteGeneracion(rejilla);
25         System.out.println("Generación " + (i + 1) + ":");
26         System.out.println(rejilla);
27     }
28
29     // 'V' = viva 'M' = muerta
30     public static void inicializarAleatoriamente(Array2d rejilla) {
31         Random random = new Random();
32         for (int i = 0; i < rejilla.getRowSize(); i++) {
33             for (int j = 0; j < rejilla.getColSize(); j++) {
```

Aplicación de las reglas

```
31     public static void inicializarAleatoriamente(Array2d rejilla) {
32     }
33
34     public static void siguienteGeneracion(Array2d rejilla) {
35         Array2d nuevaRejilla = new Array2d(rejilla.getRowSize(), rejilla.getColSize());
36
37         for (int i = 0; i < rejilla.getRowSize(); i++) {
38             for (int j = 0; j < rejilla.getColSize(); j++) {
39                 int vecinosVivos = contarVecinosVivos(rejilla, i, j);
40                 char estadoActual = rejilla.getItem(i, j);
41
42                 if (estadoActual == 'V') {
43                     if (vecinosVivos < 2 || vecinosVivos > 3) {
44                         nuevaRejilla.setItem(i, j, dato:'M');
45                     } else {
46                         nuevaRejilla.setItem(i, j, dato:'V');
47                     }
48                 } else {
49                     if (vecinosVivos == 3) {
50                         nuevaRejilla.setItem(i, j, dato:'V');
51                     } else {
52                         nuevaRejilla.setItem(i, j, dato:'M');
53                     }
54                 }
55             }
56         }
57
58         for (int i = 0; i < rejilla.getRowSize(); i++) {
59             for (int j = 0; j < rejilla.getColSize(); j++) {
60                 rejilla.setItem(i, j, nuevaRejilla.getItem(i, j));
61             }
62         }
63     }
64
65     }
66
67     }
68
69     }
70
71     }
72
73     }
```

Arreglos2dMain (Estructura de Datos > Tareas > Tarea 6 > tarea6 > src > main > java > unam > mx > Arreglos2dMain > Arreglos2d > toString)

Arreglos2dMain.java 1

Array2d.java 1

```
4 public class Array2d {
27     public int getColSize() {
29     }
30
31     public void setItem(int ren, int col, Character dato) {
32         if (ren >= 0 && ren < this.rowSize && col >= 0 && col < this.colSize) {
33             this.data[ren][col] = dato;
34         } else {
35             System.out.println(x:"Indices fuera de rango");
36         }
37     }
38
39     public Character getItem(int ren, int col) {
40         if (ren >= 0 && ren < this.rowSize && col >= 0 && col < this.colSize) {
41             return this.data[ren][col];
42         } else {
43             System.out.println(x:"Indices fuera de rango");
44         }
45         return '\0';
46     }
47
48     @Override
49     public String toString() {
50         StringBuilder str = new StringBuilder();
51
52         for (int i = 0; i < this.rowSize; i++) {
53             for (int j = 0; j < this.colSize; j++) {
54                 str.append(this.data[i][j]).append(str: " "); // Mejorar formato
55             }
56             str.append(str: "\n");
57         }
58         return str.toString();
59     }
60 }
61
```

Ln 54, Col 52 Spaces: 4 UTF-8 CRLF () Java

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Run: Arreglos2dMain

Estado inicial:
M V V V M V
M V V M V M
V M M V V V
M M V M V V
M V V M V M
M V V M V M

Introduce el número de generaciones a simular:

Ln 75, Col 1 Spaces: 4 UTF-8 LF () Java

```
PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS
M V V M V M
M V V M V M

Introduce el número de generaciones a simular: 2
Generación 1:
M V M V V M
V M M M M M
M M M M M M
M M V M M M
M M M M V M
M V V M M M

Generación 1:
M M M M M M
M M M M M M
M M M M M M
M M M M M M
M V V M M M
M M M M M M

PS C:\Users\axelx\Documents\FES\3er semestre\Estructura de Datos>
```

```
package unam.mx;
import java.util.Random;
import java.util.Scanner;

public class Arreglos2dMain {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);

        int filas = 6;
        int columnas = 6;
        Array2d rejilla = new Array2d(filas, columnas);

        inicializarAleatoriamente(rejilla);

        System.out.println("Estado inicial:");
        System.out.println(rejilla);

        System.out.print("Introduce el número de generaciones a simular: ");
        int generaciones = teclado.nextInt();

        for (int i = 0; i < generaciones; i++) {
            siguienteGeneracion(rejilla);
            System.out.println("Generación " + (i + 1) + ":");
            System.out.println(rejilla);
        }
    }

    // 'V' = viva 'M' = muerta
    public static void inicializarAleatoriamente(Array2d rejilla) {
        Random random = new Random();
```

```

        for (int i = 0; i < rejilla.getRowSize(); i++) {
            for (int j = 0; j < rejilla.getColSize(); j++) {
                if (random.nextBoolean()) {
                    rejilla.setItem(i, j, 'V');
                } else {
                    rejilla.setItem(i, j, 'M');
                }
            }
        }
    }

    public static void siguienteGeneracion(Array2d rejilla) {
        Array2d nuevaRejilla = new Array2d(rejilla.getRowSize(),
rejilla.getColSize());

        for (int i = 0; i < rejilla.getRowSize(); i++) {
            for (int j = 0; j < rejilla.getColSize(); j++) {
                int vecinosVivos = contarVecinosVivos(rejilla, i, j);
                char estadoActual = rejilla.getItem(i, j);

                if (estadoActual == 'V') {
                    if (vecinosVivos < 2 || vecinosVivos > 3) {
                        nuevaRejilla.setItem(i, j, 'M');
                    } else {
                        nuevaRejilla.setItem(i, j, 'V');
                    }
                } else {
                    if (vecinosVivos == 3) {
                        nuevaRejilla.setItem(i, j, 'V');
                    } else {
                        nuevaRejilla.setItem(i, j, 'M');
                    }
                }
            }
        }

        for (int i = 0; i < rejilla.getRowSize(); i++) {
            for (int j = 0; j < rejilla.getColSize(); j++) {
                rejilla.setItem(i, j, nuevaRejilla.getItem(i, j));
            }
        }
    }
}

```

```

    public static int contarVecinosVivos(Array2d rejilla, int fila, int
columna) {
        int[] dFila = {-1, -1, -1, 0, 0, 1, 1, 1};
        int[] dColumna = {-1, 0, 1, -1, 1, -1, 0, 1};
        int vecinosVivos = 0;

        for (int i = 0; i < dFila.length; i++) {
            int nuevaFila = fila + dFila[i];
            int nuevaColumna = columna + dColumna[i];

            if (nuevaFila >= 0 && nuevaFila < rejilla.getRowSize() &&
nuevaColumna >= 0 && nuevaColumna < rejilla.getColSize()) {
                if (rejilla.getItem(nuevaFila, nuevaColumna) == 'V') {
                    vecinosVivos++;
                }
            }
        }

        return vecinosVivos;
    }
}

```

```

package unam.mx;
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;
    }

    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';
    }

    @Override
    public String toString() {
        StringBuilder str = new StringBuilder();

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

```