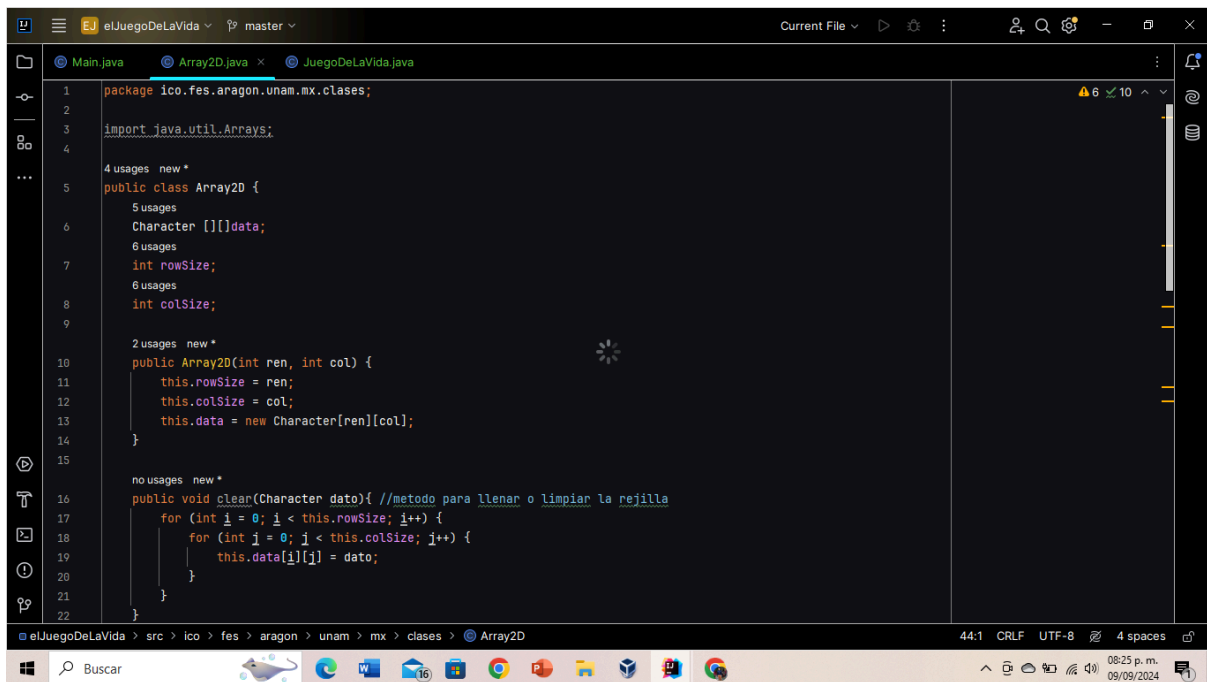
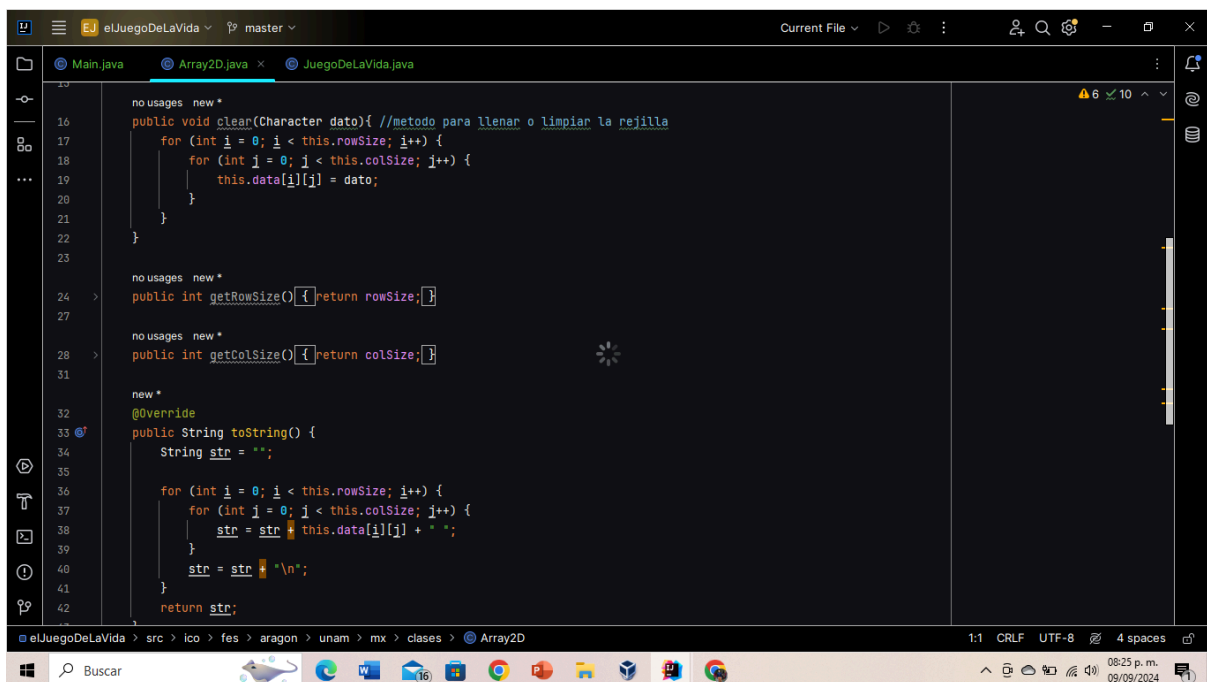


Captura del código:



```
1 package ico.fes.aragon.unam.mx.clases;
2
3 import java.util.Arrays;
4
5 4 usages new *
6 public class Array2D {
7     5 usages
8     Character [][]data;
9     6 usages
10    int rowSize;
11    6 usages
12    int colSize;
13
14    2 usages new *
15    public Array2D(int ren, int col) {
16        this.rowSize = ren;
17        this.colSize = col;
18        this.data = new Character[ren][col];
19    }
20
21    no usages new *
22    public void clear(Character dato){ //metodo para llenar o limpiar la rejilla
23        for (int i = 0; i < this.rowSize; i++) {
24            for (int j = 0; j < this.colSize; j++) {
25                this.data[i][j] = dato;
26            }
27        }
28    }
29 }
```



```
30
31
32    no usages new *
33    public int getRowSize(){ return rowSize;}
34
35    no usages new *
36    public int getColSize(){ return colSize;}
37
38    new *
39    @Override
40    public String toString() {
41        String str = "";
42
43        for (int i = 0; i < this.rowSize; i++) {
44            for (int j = 0; j < this.colSize; j++) {
45                str = str + this.data[i][j] + " ";
46            }
47            str = str + "\n";
48        }
49        return str;
50    }
51 }
```

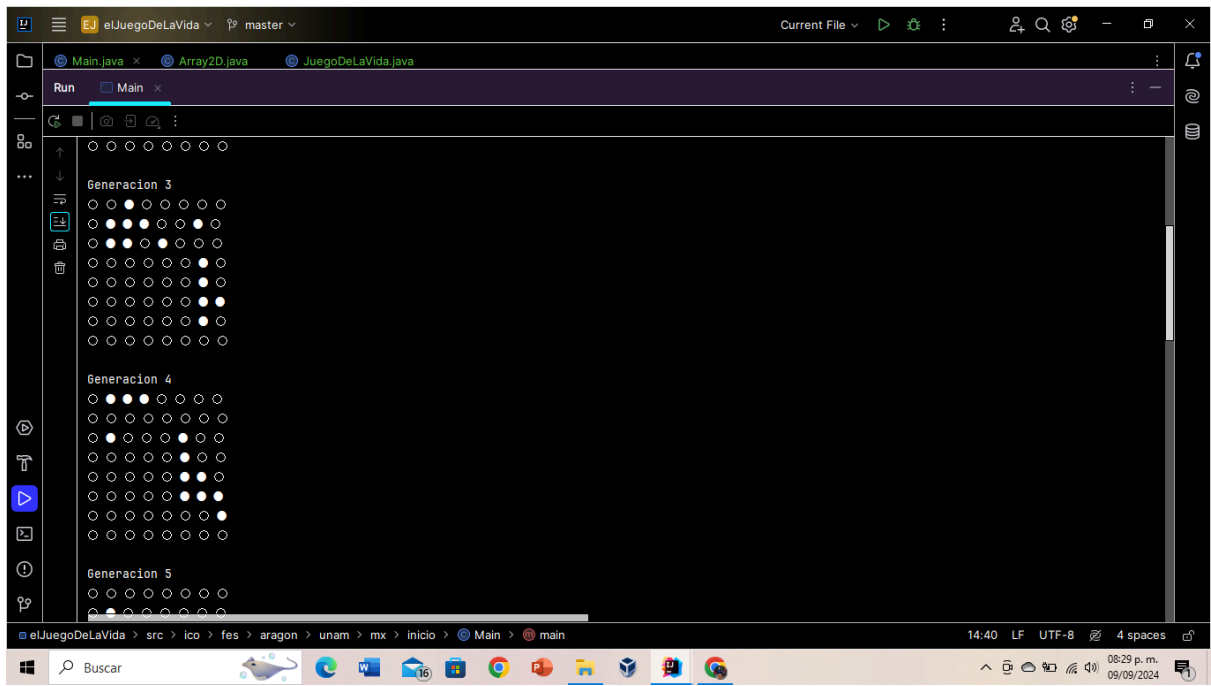
```
elJuegoDeLaVida master
Current File
6 10
Main.java Array2D.java JuegoDeLaVida.java
41 }
42 return str;
43 }
44
45 5 usages new *
46 public void setItem(int ren, int col, Character dato){
47     if (ren>=0 && ren < this.rowSize && col >= 0 && col < this.colSize){
48         this.data[ren][col]=dato;
49     }else{
50         System.out.println("Indices fuera de rango");
51     }
52 }
53
54 2 usages new *
55 public Character getItem(int ren, int col){
56     if (ren>=0 && ren < this.rowSize && col >= 0 && col < this.colSize){
57         return this.data[ren][col];
58     }else{
59         System.out.println("Indices fuera de rango");
60     }
61     return '\0';
62 }
63
64 }
65

elJuegoDeLaVida > src > ico > fes > aragon > unam > mx > clases > Array2D
1:1 CRLF UTF-8 4 spaces
Buscar
```

```
elJuegoDeLaVida master
Current File
2 46
Main.java Array2D.java JuegoDeLaVida.java
1 package ico.fes.aragon.unam.mx.clases;
2
3 3 usages new *
4 public class JuegoDeLaVida {
5     6 usages
6     private Array2D rejilla;
7     5 usages
8     private int cols;
9     5 usages
10    private int rows;
11
12    1 usage new *
13    public JuegoDeLaVida(int col, int ren) {
14        rejilla = new Array2D(ren, col);
15        this.cols = col;
16        this.rows = ren;
17    }
18
19    1 usage new *
20    public void inicializarRandom() {
21        for (int i = 0; i < rows; i++) {
22            for (int j = 0; j < cols; j++) {
23                char estado = Math.random() < 0.5 ? 'O' : '●';
24                rejilla.setItem(i, j, estado);
25            }
26        }
27    }
28
29 }
30
31 Database
32
33 74:1 CRLF UTF-8 4 spaces
Buscar
```

```
elJuegoDeLaVida master
Main.java Array2D.java JuegoDeLaVida.java
15 public void inicializarRandom() {
16     for (int i = 0; i < rens; i++) {
17         for (int j = 0; j < cols; j++) {
18
19             char estado = Math.random() < 0.5 ? 'O' : '●';
20             rejilla.setItem(i, j, estado);
21         }
22     }
23 }
24
25 1 usage new *
26 public int contadorVecinosVivos(int ren, int col) {
27     int contador = 0;
28     for (int i = -1; i <= 1; i++) {
29         for (int j = -1; j <= 1; j++) {
30             if (i == 0 && j == 0) continue;
31             int vecinosRen = ren + i;
32             int vecinosCol = col + j;
33             if (posicionValida(vecinosRen, vecinosCol) && rejilla.getItem(vecinosRen, vecinosCol) == '●') {
34                 contador++;
35             }
36         }
37     }
38     return contador;
39 }
40 // Verifica si una posición es válida dentro del tablero
elJuegoDeLaVida > src > ico > fes > aragon > unam > mx > clases > JuegoDeLaVida > imprimirRejilla
74:1 CRLF UTF-8 4 spaces
```

```
elJuegoDeLaVida master
Main.java Array2D.java JuegoDeLaVida.java
40 public boolean posicionValida(int ren, int col) { return ren >= 0 && ren < rens && col >= 0 && col < cols; }
43
44 2 usages new *
45 public void actualizacionRejilla() {
46     Array2D siguienteRejilla = new Array2D(rens, cols);
47
48     for (int i = 0; i < rens; i++) {
49         for (int j = 0; j < cols; j++) {
50             int vecinosVivos = contadorVecinosVivos(i, j);
51             char estadoActual = rejilla.getItem(i, j);
52
53             if (estadoActual == '●') { // Célula viva
54                 if (vecinosVivos < 2 || vecinosVivos > 3) {
55                     siguienteRejilla.setItem(i, j, dato: 'O');
56                 } else {
57                     siguienteRejilla.setItem(i, j, dato: '●');
58                 }
59             } else { // Célula muerta
60                 if (vecinosVivos == 3) {
61                     siguienteRejilla.setItem(i, j, dato: '●');
62                 } else {
63                     siguienteRejilla.setItem(i, j, dato: 'O');
64                 }
65             }
66         }
67     }
68     rejilla = siguienteRejilla; // Actualizamos el estado
elJuegoDeLaVida > src > ico > fes > aragon > unam > mx > clases > JuegoDeLaVida
43:1 CRLF UTF-8 4 spaces
```

```
elJuegoDeLaVida master
Current File
Main.java Array2D.java JuegoDeLaVida.java
Run Main
Generation 5
Generation 6
Generation 7
elJuegoDeLaVida > src > ico > fes > aragon > unam > mx > inicio > Main > main 14:40 LF UTF-8 4 spaces
08:29 p.m. 09/09/2024
```

```
elJuegoDeLaVida master
Current File
Main.java Array2D.java JuegoDeLaVida.java
Run Main
Generation 7
Generation 8
Generation 9
elJuegoDeLaVida > src > ico > fes > aragon > unam > mx > inicio > Main > main 14:40 LF UTF-8 4 spaces
08:29 p.m. 09/09/2024
```