

Main:

```
public class Main {  
    public static void main(String[]  
args) {  
        JuegoDeLaVida juego = new  
JuegoDeLaVida(5, 5);  
  
        // Configuración inicial:  
        Células vivas en posiciones específicas  
        int[][] configuracionInicial = {  
            {1, 1}, {1, 2}, {1, 3},  
            // Una línea horizontal  
            {2, 2}  
        };  
  
        juego.setConfiguracionInicial(configura  
cionInicial);  
  
        // Calcular 5 generaciones  
        juego.calcularGeneraciones(5);  
    }  
}
```

Juego de la vida:

```
public class JuegoDeLaVida {  
    private Array2d grid;
```

```
    public JuegoDeLaVida(int filas, int
columnas) {
        grid = new Array2d(filas,
columnas);
        grid.clear('D'); // 'D'
representa celula muerta
    }
```

```
    public void
setConfiguracionInicial(int[][]
configuracion) {
        for (int[] celula :
configuracion) {
            int fila = celula[0];
            int col = celula[1];
            grid.setItem(fila, col,
'V'); // 'V' representa celula viva
        }
    }
```

```
    public void calcularGeneraciones(int
generaciones) {
        for (int g = 0; g <
generaciones; g++) {
            grid =
calcularSiguienteGeneracion();
        }
```

```
System.out.println("Generación " + (g +  
1) + ":\n" + grid);  
    }  
}
```

```
    private Array2d  
    calcularSiguienteGeneracion() {  
        Array2d nuevaGeneracion = new  
Array2d(grid.getRowSize(),  
grid.getColSize());
```

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

```
                if (estadoActual == 'V')  
{ // Célula viva  
                    if (vecinosVivos < 2  
|| vecinosVivos > 3) {
```

```
nuevaGeneracion.setItem(i, j, 'D'); //  
Muere
```

```
    } else {
```

```
nuevaGeneracion.setItem(i, j, 'V'); //
```

```
Sobrevive
```

```
    }
```

```
    } else { // Célula
```

```
muerta
```

```
        if (vecinosVivos ==
```

```
3) {
```

```
nuevaGeneracion.setItem(i, j, 'V'); //
```

```
Nace
```

```
    } else {
```

```
nuevaGeneracion.setItem(i, j, 'D'); //
```

```
Permanece muerta
```

```
    }
```

```
    }
```

```
    }
```

```
    }
```

```
    return nuevaGeneracion;
```

```
}
```

```
    private int contarVecinosVivos(int  
fila, int col) {
```

```
        int conteo = 0;
```

```
        for (int i = -1; i <= 1; i++) {
```

```
        for (int j = -1; j <= 1;
j++) {
            if (i == 0 && j == 0)
continue; // Ignorar la celula misma
            int nuevaFila = fila +
i;
            int nuevaCol = col + j;
            if (nuevaFila >= 0 &&
nuevaFila < grid.getRowSize() &&
nuevaCol >= 0 &&
nuevaCol < grid.getColSize() &&

grid.getItem(nuevaFila, nuevaCol) ==
'V') {
                conteo++;
            }
        }
    }
    return conteo;
}
}
```

This screenshot shows a VS Code terminal window with the title bar 'Run: Main x'. The terminal displays the output of a program for 'Generación 5'. The output consists of a header line 'Generación 5:' followed by three lines of data, each containing five comma-separated values. The values are 'D' and 'V'. The first line is 'D, D, V, D, D,', the second is 'D, V, D, V, D,', and the third is 'D, V, D, V, D,'. The terminal interface includes a left sidebar with icons for Explorer, Search, and Run and Debug. The bottom status bar shows 'Version Control', 'Run', 'TODO', 'Problems', 'Terminal', and 'Services'.

```
Run: Main x
Generación 5:
D, D, V, D, D,
D, V, D, V, D,
D, V, D, V, D,
```

This screenshot shows a VS Code terminal window with the title bar 'Run: Main x'. The terminal displays the output of a program for 'Generación 2'. The output consists of a header line 'Generación 2:' followed by three lines of data, each containing five comma-separated values. The values are 'D' and 'V'. The first line is 'D, V, V, V, D,', the second is 'D, D, D, D, D,', and the third is 'D, V, D, V, D,'. The terminal interface includes a left sidebar with icons for Explorer, Search, and Run and Debug. The bottom status bar shows 'Version Control', 'Run', 'TODO', 'Problems', and 'Terminal'. Below the status bar, a message reads 'Build completed successfully in 7 sec, 66 ms (2 minutes ago)'.

```
Run: Main x
Generación 2:
D, V, V, V, D,
D, D, D, D, D,
D, V, D, V, D,
```

Build completed successfully in 7 sec, 66 ms (2 minutes ago)