

```

public class Celda {
    private boolean tieneMina;
    private int minasAdyacentes;
    private boolean descubierta;

    // Constructor
    public Celda() {
        this.tieneMina = false;
        this.minasAdyacentes = 0;
        this.descubierta = false;
    }

    // Métodos get y set
    public boolean tieneMina() {
        return tieneMina;
    }

    public void ponerMina() {
        this.tieneMina = true;
    }

    public int getMinasAdyacentes() {
        return minasAdyacentes;
    }

    public void setMinasAdyacentes(int minasAdyacentes) {
        this.minasAdyacentes = minasAdyacentes;
    }

    public boolean estaDescubierta() {
        return descubierta;
    }

    public void descubrir() {
        this.descubierta = true;
    }

    // Mostrar la celda (revela la cantidad de minas adyacentes o una 'X' si tiene una
mina)
    public String mostrar() {
        if (descubierta) {
            return tieneMina ? "X" : (minasAdyacentes == 0 ? " " :
String.valueOf(minasAdyacentes));
        } else {
            return "***"; // Celda oculta
        }
    }
}

```

```

import java.util.Random;

public class Tablero {
    private Celda[][] tablero;
    private int filas;
    private int columnas;
    private int totalMinas;

    // Constructor
    public Tablero(int filas, int columnas, int totalMinas) {
        this.filas = filas;
        this.columnas = columnas;
        this.totalMinas = totalMinas;
        this.tablero = new Celda[filas][columnas];

        // Inicializar el tablero con celdas vacías
        for (int i = 0; i < filas; i++) {
            for (int j = 0; j < columnas; j++) {
                tablero[i][j] = new Celda();
            }
        }

        // Colocar minas aleatoriamente
        colocarMinas();
        // Calcular las minas adyacentes para cada celda
        calcularMinasAdyacentes();
    }

    // Colocar las minas aleatoriamente en el tablero
    private void colocarMinas() {
        Random rand = new Random();
        int minasColocadas = 0;

        while (minasColocadas < totalMinas) {
            int fila = rand.nextInt(filas);
            int columna = rand.nextInt(columnas);

            if (!tablero[fila][columna].tieneMina()) {
                tablero[fila][columna].ponerMina();
                minasColocadas++;
            }
        }
    }

    // Calcular las minas adyacentes para cada celda
    private void calcularMinasAdyacentes() {
        for (int i = 0; i < filas; i++) {

```

```

        for (int j = 0; j < columnas; j++) {
            if (!tablero[i][j].tieneMina()) {
                int minasAdyacentes = contarMinasAdyacentes(i, j);
                tablero[i][j].setMinasAdyacentes(minasAdyacentes);
            }
        }
    }

    // Contar cuántas minas hay alrededor de una celda
    private int contarMinasAdyacentes(int fila, int columna) {
        int minas = 0;
        for (int i = -1; i <= 1; i++) {
            for (int j = -1; j <= 1; j++) {
                int nuevaFila = fila + i;
                int nuevaColumna = columna + j;

                if (nuevaFila >= 0 && nuevaFila < filas && nuevaColumna >= 0 &&
nuevaColumna < columnas) {
                    if (tablero[nuevaFila][nuevaColumna].tieneMina()) {
                        minas++;
                    }
                }
            }
        }
        return minas;
    }

    // Mostrar el tablero (las celdas que están descubiertas se muestran, las ocultas con
    "**")
    public void mostrarTablero() {
        for (int i = 0; i < filas; i++) {
            for (int j = 0; j < columnas; j++) {
                System.out.print(tablero[i][j].mostrar() + " ");
            }
            System.out.println();
        }
    }

    // Descubrir una celda, si es una mina, el juego termina
    public boolean descubrirCelda(int fila, int columna) {
        if (tablero[fila][columna].estaDescubierta()) {
            return true; // Si ya está descubierta, no hacer nada
        }

        tablero[fila][columna].descubrir();

        // Si la celda tiene una mina, el jugador pierde

```

```

return !tablero[fila][columna].tieneMina();
}

// Verificar si el juego ha terminado (si no hay más celdas sin descubrir)
public boolean juegoTerminado() {
for (int i = 0; i < filas; i++) {
for (int j = 0; j < columnas; j++) {
    if (!tablero[i][j].estaDescubierta() && !tablero[i][j].tieneMina()) {
        return false; // Si hay alguna celda sin descubrir, el juego sigue
    }
}
}
return true; // Si no hay celdas sin descubrir, el juego ha terminado
}
}

```

```

import java.util.Scanner;

public class Buscaminas {
    private Tablero tablero;
    private Scanner scanner;

    // Constructor
    public Buscaminas(int filas, int columnas, int totalMinas) {
        tablero = new Tablero(filas, columnas, totalMinas);
        scanner = new Scanner(System.in);
    }

    // Método para jugar
    public void jugar() {
        while (true) {
            System.out.println("Tablero actual:");
            tablero.mostrarTablero();

            System.out.println("\nIngrese las coordenadas para descubrir una celda (formato: fila
columna): ");
            int fila = scanner.nextInt();
            int columna = scanner.nextInt();

            // Verificar que las coordenadas estén dentro de los límites
            if (fila < 0 || fila >= tablero.getFilas() || columna < 0 || columna >=
tablero.getColumnas()) {
                System.out.println("¡Coordenadas inválidas! Intenta de nuevo.");
                continue;
            }

            // Intentar descubrir la celda
            if (!tablero.descubrirCelda(fila, columna)) {
                System.out.println("¡Oh no! Has descubierto una mina. ¡Juego terminado!");
                tablero.mostrarTablero();
                break;
            }

            // Comprobar si el juego ha terminado (sin minas restantes sin descubrir)
            if (tablero.juegoTerminado()) {
                System.out.println("¡Felicidades! Has ganado el juego.");
                tablero.mostrarTablero();
                break;
            }
        }
    }

    public static void main(String[] args) {
        Buscaminas juego = new Buscaminas(5, 5, 5); // 5x5 tablero con 5 minas
    }
}

```

```
juego.jugar();  
}  
}
```