

```
public class Celda {
    private int valor; // Valor de la celda (0 si está vacía)
    private boolean bloqueada; // Indica si la celda está bloqueada o no (es decir, si es
una celda inicial)

    // Constructor para inicializar la celda
    public Celda() {
        this.valor = 0; // 0 significa vacía
        this.bloqueada = false; // Por defecto la celda no está bloqueada
    }

    // Métodos getters y setters
    public int getValor() {
        return valor;
    }

    public void setValor(int valor) {
        if (!bloqueada) {
            this.valor = valor;
        }
    }

    public boolean isBloqueada() {
        return bloqueada;
    }

    public void setBloqueada(boolean bloqueada) {
        this.bloqueada = bloqueada;
    }
}
```

```

public class Tablero {
    private Celda[][] tablero;

    // Constructor que inicializa un tablero vacío
    public Tablero() {
        tablero = new Celda[9][9];
        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 9; j++) {
                tablero[i][j] = new Celda();
            }
        }
    }

    // Método para mostrar el tablero
    public void mostrarTablero() {
        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 9; j++) {
                System.out.print(tablero[i][j].getValor() + " ");
            }
            System.out.println();
        }
    }

    // Método para colocar un valor en una celda
    public boolean colocarValor(int fila, int columna, int valor) {
        if (fila < 0 || fila >= 9 || columna < 0 || columna >= 9) {
            System.out.println("Posición inválida.");
            return false;
        }
        if (tablero[fila][columna].isBloqueada()) {
            System.out.println("No puedes cambiar esta celda, está bloqueada.");
            return false;
        }
        if (esValido(fila, columna, valor)) {
            tablero[fila][columna].setValor(valor);
            return true;
        }
        return false;
    }

    // Método para verificar si un valor es válido en una celda
    private boolean esValido(int fila, int columna, int valor) {
        return !estaEnFila(fila, valor) && !estaEnColumna(columna, valor) &&
!estaEnSubcuadro(fila, columna, valor);
    }

    // Verificar si el valor ya está en la fila
    private boolean estaEnFila(int fila, int valor) {

```

```

for (int j = 0; j < 9; j++) {
    if (tablero[fil][j].getValor() == valor) {
        return true;
    }
}
return false;
}

```

```

// Verificar si el valor ya está en la columna
private boolean estaEnColumna(int columna, int valor) {
    for (int i = 0; i < 9; i++) {
        if (tablero[i][columna].getValor() == valor) {
            return true;
        }
    }
    return false;
}

```

```

// Verificar si el valor ya está en el subcuadro 3x3
private boolean estaEnSubcuadro(int fila, int columna, int valor) {
    int filaInicio = (fila / 3) * 3;
    int colInicio = (columna / 3) * 3;
    for (int i = filaInicio; i < filaInicio + 3; i++) {
        for (int j = colInicio; j < colInicio + 3; j++) {
            if (tablero[i][j].getValor() == valor) {
                return true;
            }
        }
    }
    return false;
}

```

```

// Método para bloquear una celda (por ejemplo, para colocar las celdas iniciales)
public void bloquearCelda(int fila, int columna) {
    if (fila >= 0 && fila < 9 && columna >= 0 && columna < 9) {
        tablero[fila][columna].setBloqueada(true);
    }
}

```

```

// Método para verificar si el tablero está resuelto
public boolean estaResuelto() {
    for (int i = 0; i < 9; i++) {
        for (int j = 0; j < 9; j++) {
            if (tablero[i][j].getValor() == 0 || !esValido(i, j, tablero[i][j].getValor())) {
                return false;
            }
        }
    }
}

```

```
        return true;
    }
}
```

```

import java.util.Scanner;

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

        // Crear un tablero de Sudoku
        Tablero tablero = new Tablero();

        // Configuración del tablero (bloquear algunas celdas con valores iniciales)
        tablero.colocarValor(0, 0, 5);
        tablero.colocarValor(1, 1, 3);
        tablero.colocarValor(4, 4, 7);
        // Puedes agregar más valores iniciales aquí

        tablero.bloquearCelda(0, 0); // Bloquear la celda para que no sea modificable
        tablero.bloquearCelda(1, 1);
        tablero.bloquearCelda(4, 4);

        // Bucle de juego
        while (!tablero.estaResuelto()) {
            tablero.mostrarTablero();

            // Solicitar al jugador que ingrese una fila, columna y valor
            System.out.println("Introduce la fila (0-8): ");
            int fila = scanner.nextInt();
            System.out.println("Introduce la columna (0-8): ");
            int columna = scanner.nextInt();
            System.out.println("Introduce el valor (1-9): ");
            int valor = scanner.nextInt();

            // Intentar colocar el valor
            if (tablero.colocarValor(fila, columna, valor)) {
                System.out.println("Valor colocado correctamente.");
            } else {
                System.out.println("Valor no válido o celda bloqueada.");
            }
        }

        // Mostrar el tablero resuelto
        tablero.mostrarTablero();
        System.out.println("¡Felicidades, has resuelto el Sudoku!");
    }
}

```