***Actividades de Clase 14***

**Alumnas:** Pierrot María Paula, Korniansky Eden, Prevosti Chiara y Chen Hui Jun.

**Actividad 2:** *modificar un programa de Sudoku para que trabaje con un tablero de 6x6, usando los números del 1 al 6 y con tres subcuadrantes de 6 números cada uno. Siguiendo estas instrucciones: Cambiar el tamaño del tablero de 9x9 a 6x6. Ajustar las reglas del juego para que en lugar de 9 subcuadrantes de 3x3, ahora el tablero tenga tres subcuadrantes de 6 números. Es decir, cada sub cuadrante contendrá 6 celdas en total, de 2x3. Asegurarse de que los números utilizados sean del 1 al 6.*

Tablero 6x6, números [1;6]

public class SudokuSolver {

// Tamaño del tablero

private static final int *SIZE* = 6;

// Método principal que llama a la función de resolución

public static void main(String[] args) {

int[][] board = {

{5, 3, 0, 0, 7, 0},

{6, 0, 0, 1, 9, 5},

{0, 9, 8, 0, 0, 0},

{8, 0, 0, 0, 6, 0},

{4, 0, 0, 8, 0, 3},

{7, 0, 0, 0, 2, 0},

};

if (*solveSudoku*(board)) {

*printBoard*(board);

} else {

System.*out*.println("No existe solución para este Sudoku.");

}

}

// Método que resuelve el Sudoku usando backtracking

public static boolean solveSudoku(int[][] board) {

for (int row = 0; row < *SIZE*; row++) {

for (int col = 0; col < *SIZE*; col++) {

if (board[row][col] == 0) { // Encuentra una celda vacía

for (int num = 1; num <= 6; num++) {

if (*isValid*(board, row, col, num)) {

board[row][col] = num;

// Recursivamente intenta resolver el tablero

if (*solveSudoku*(board)) {

return true;

}

// Deshacer el intento si falla

board[row][col] = 0;

}

}

return false; // Si ningún número es válido, retrocede

}

}

}

return true; // El tablero está completo y es válido

}

// Método para verificar si un número puede ser colocado en una posición

private static boolean isValid(int[][] board, int row, int col, int num) {

// Verificar si el número está en la fila

for (int i = 0; i < *SIZE*; i++) {

if (board[row][i] == num) {

return false;

}

}

// Verificar si el número está en la columna

for (int i = 0; i < *SIZE*; i++) {

if (board[i][col] == num) {

return false;

}

}

// Verificar si el número está en el subcuadro 3x3

int startRow = row - row % 2;

int startCol = col - col % 3;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

if (board[i + startRow][j + startCol] == num) {

return false;

}

}

}

return true; // Si pasa todas las pruebas, es válido

}

// Método para imprimir el tablero

private static void printBoard(int[][] board) {

for (int row = 0; row < *SIZE*; row++) {

for (int col = 0; col < *SIZE*; col++) {

System.*out*.print(board[row][col] + " ");

}

System.*out*.println();

}

}

}