



PREGUNTA 3 – PED 2.

PREDAS. – UNED – LAS TABLAS. PAULINO ESTEBAN BERMÚDEZ
RODRÍGUEZ

3.- CUESTIONES TEÓRICAS DE LA PRÁCTICA

1) ¿Cuáles son las heurísticas que permiten condiciones de poda? Expón las que hayas utilizado en el diseño del algoritmo.

Las condiciones que he usado para seleccionar si un nodo es válido es basándome en la estrategia FIFO, ya que, al llegar a una celda vacía, pruebo con todos los números y compruebo su fila y columna correspondiente, en caso de ser un candidato válido. Lo escribo en la celda y continúo con el siguiente cuadrado.

1

```

if(inicial[i][j]){
    for(int k = 1; k <= 9; k++){
        matrizSudoku[i][j]=k;
        if(SiVale(i, j, matrizSudoku)){
            if (i==8 && j == 8){
                int aux[][]= matrizSudoku;
                mostrar(matrizSudoku);
            }
            if(i<8 && j==8){
                solucion(i+1, 0, matrizSudoku, inicial);
            }
            if(i<=8 && j<8){
                solucion(i, j+1, matrizSudoku, inicial);
            }
        }
    }
    matrizSudoku[i][j] = 0;
} else {
    if (i == 8 && j == 8 ){
        mostrar(matrizSudoku);
    }
    if (i<8 && j ==8){
        solucion(i+1, 0 , matrizSudoku, inicial);
    }
    if ( i <= 8 && j < 8){
        solucion(i, j+1, matrizSudoku, inicial);
    }
}
}

```

2) Explica qué otros esquemas pueden resolver el problema, y razona sobre su idoneidad.

Otra forma de resolver el problema es mediante programación lineal, calculando la raíces de cada número posible de entrada para saber si es un número válido.

```
public static boolean esSeguro(int[][] tablero, int fila, int columna, int num)
{
    for (int d = 0; d < tablero.length; d++) {
        if (tablero[fila][d] == num) {
            return false;
        }
    }

    for (int r = 0; r < tablero.length; r++) {
        if (tablero[r][columna] == num) {
            return false;
        }
    }

    int sqrt = (int) Math.sqrt(tablero.length);
    int inicioFila = fila - fila % sqrt;
    int inicioColumna = columna - columna % sqrt;

    for (int r = inicioFila;
         r < inicioFila + sqrt; r++) {
        for (int d = inicioColumna;
             d < inicioColumna + sqrt; d++) {
            if (tablero[r][d] == num) {
                return false;
            }
        }
    }

    return true;
}
```

```
int[][] tablero, int N)
{
    for (int r = 0; r < N; r++) {
        for (int d = 0; d < N; d++) {
            System.out.print(tablero[r][d]);
            System.out.print(" ");
        }
        System.out.print("\n");

        if ((r + 1) % (int) Math.sqrt(N) == 0) {
            System.out.print("");
        }
    }
}
```