

## ACTIVIDAD 2

### Resolución de Sudokus mediante PSR

El Sudoku es un rompecabezas matemático cuyo objetivo es rellenar una cuadrícula de  $N \times N$  celdas dividida en subcuadrículas con las cifras del 1 al  $N$  partiendo de algunos números ya dispuestos en algunas de las celdas. No se debe repetir ninguna cifra en una misma fila, columna o subcuadrícula. Sea el siguiente ejemplo para el caso de  $N=4$  (<http://www.sudoku-online.org>); así como la tercera columna indica cómo Matlab internamente numera cada celda, así la celda de la fila 1 columna 1, corresponde con la celda 1, la de la fila 1, columna 2, corresponde a la celda 5, etc (ver `help ind2sub`, `sub2ind`)

	4	3	2
3	2	4	
4			3
2	3		4

1	4	3	2
3	2	4	1
4	1	2	3
2	3	1	4

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Se dispone de dos funciones:

- Para crear dos ejemplos de Sudokus que tienen solución:

```
function [X,dominios]=crea_Sudoku(N)
```

- Para obtener la submatriz de una determinada posición:

```
function m=submatrix(x,k,N)
%% para valores de N=4, 9, 16, 25, etc.
```

Resolución del Problema:

- isSafe:** Implementa la función que comprueba que dado un tablero de sudoku,  $X$ , la asignación a la variable,  $k$  del valor  $c$ , es una asignación consistente.
- BACTRACKING:** Resuelve el problema aplicando el algoritmo de backtracking y comprueba que todo funciona correctamente.

3. **ARCO-CONSISTENCIA:** Añade una función para ir eliminando de los dominios de las variables adyacentes a una variable dada aquellos valores no consistentes. Esta función debe incorporarse al principio del problema, para actualizar los dominios de la asignación inicial dada en `crea_Sudoku`, y después de cada asignación realizada en el proceso de Backtracking.

4. **PRUEBA FINAL:** Prueba final del código con el siguiente Sudoku 9x9:

0	0	2	0	0	0	0	0	0
3	7	6	1	0	0	0	2	0
8	0	0	0	6	9	0	3	0
0	0	8	0	0	2	0	4	5
0	4	3	0	5	0	2	0	6
0	0	0	0	0	0	0	0	0
4	0	0	7	3	0	6	0	0
0	0	0	8	0	0	0	5	0
6	0	7	0	2	5	0	8	3

9	1	2	5	7	3	8	6	4
3	7	6	1	8	4	5	2	9
8	5	4	2	6	9	7	3	1
7	6	8	3	1	2	9	4	5
1	4	3	9	5	8	2	7	6
5	2	9	6	4	7	3	1	8
4	8	5	7	3	1	6	9	2
2	3	1	8	9	6	4	5	7
6	9	7	4	2	5	1	8	3