

# Laboratorio de Conocimiento y Razonamiento Automatizado. Curso 2022-23

Manuel de Buenaga Rodríguez  
José Luis Castillo Sequera  
José Enrique Morais San Miguel



Universidad  
de Alcalá

# Práctica 1: Sudoku

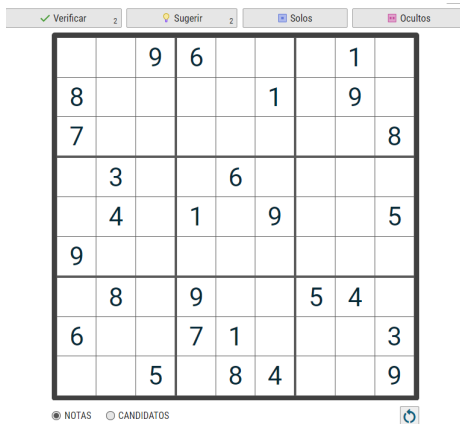
Un Sudoku está formado por una cuadrícula de 9x9 casillas, dividida en nueve cuadros 3x3. A partir de algunos números (del 1 al 9) ya colocados, hay que completar las casillas vacías con dígitos del 1 al 9 de tal forma que estos no deben repetirse en una misma fila, columna o cuadro 3x3. No discutiremos cómo se puede garantizar que exista solución única a un sudoku. Por otro lado, el objetivo de la práctica no es resolver sudokus: en el manual de SWI-Prolog (<http://www.swi-prolog.org/man/clpfd.html>) se dice cómo hacerlo usando la librería clpfd (Constraint Logic Programming over Finite Domains) que trabaja sobre problemas de programación con restricciones. Esta es una práctica sobre el manejo de listas en Prolog tomando como excusa los sudokus.

# Práctica 1: Sudoku

Un Sudoku será una lista de longitud 81. Cada lugar en la lista se corresponde con una casilla del tablero 9x9. Se inicia con un tablero con números entre uno y nueve donde haya un número y con un punto en el resto. Por ejemplo,

```
[.,.,3,.,2,.,7,.,.,  
5,.,.,.,.,4,.,3,  
.,.,.,3,.,.,2,5,  
.,.,5,.,1,.,6,.,.,  
.,.,4,8,.,7,.,.,.,  
2,3,7,6,.,4,8,.,.,  
.,8,.,.,.,2,.,7,.,  
3,.,.,4,.,.,2,.,8,  
.,.,9,.,.,.,6,.]
```

# Ejemplo de sudoku



Captura de

<https://www.sudokumania.com.ar/juegos/sudoku/>

## Objetivo 1. Posibilidades

Dado un sudoku, que recordemos, es una lista de longitud 81, se debe generar la lista de posibilidades para cada uno de los lugares del sudoku. Es decir, cada uno de los lugares ocupados por un punto debe sustituirse por una lista con todos los posibles números que pueden ir en ese lugar, mientras que los lugares ocupados inicialmente por dígitos no se tocan. La lista de posibilidades, por lo tanto, es una lista de la forma

$[[2, 3, \dots], 1, [3, 4, \dots], \dots, [7, 9]]$  también de longitud 81.

# Ejemplo de sudoku y posibilidades

☒ Verificar 2

☐ Sugerir 2

☐ Solos

☐ Ocultos

2345	25	9	6	23457	23578	2347	1	247
8	256	2346	2345	23457	1	23467	9	2467
7	1256	12346	2345	23459	235	2346	2356	8
125	3	1278	2458	6	2578	12478 9	278	1247
2	4	2678	1	237	9	23678	23678	5
9	12567	12678	23458	23457	23578	12346 78	23678	12467
123	8	1237	9	23	236	5	4	1267
6	29	24	7	1	25	28	28	3
123	127	5	23	8	4	1267	267	9

☐ NOTAS

☒ CANDIDATOS

Captura de

<https://www.sudokumania.com.ar/juegos/sudoku/>

## Objetivo 2. Simplificación

Una vez generada la lista de posibilidades, se trata de simplificarla siguiendo una serie de reglas. Esta simplificación en algunos casos llevará a resolver el Sudoku, pero debe quedar claro que no siempre es el caso. Las cuatro reglas que hay que implementar son las siguientes:

- Regla 0: Si hay un lugar donde solo cabe un número, lo escribimos en el lugar correspondiente y lo eliminamos de los lugares en los que aparezca de los que son conflictivos.
- Regla 1: Si hay un número que aparece en una sola de las listas que aparecen en una fila, columna o cuadro, cambiamos la lista por el número y borramos el número del resto de listas de la fila, columna o cuadro.

## Objetivo 2. Simplificación (y 2)

- Regla 2: Si dos números aparecen solos en dos lugares distintos de una fila, columna o cuadro, los borramos del resto de lugares de la fila, columna o cuadro correspondiente.
- Regla 3: Si en tres lugares de una fila, columna o cuadro sólo aparecen tres números distintos, borramos los números de las restantes listas de la fila, columna o cuadro.

**Nota 1:** Estas reglas no se aplican de manera independiente unas de otras sino que están relacionadas. Si en una fila se aplica la regla 2, por ejemplo, esto puede dar lugar a que se puedan reaplicar alguna de las reglas anteriores.

**Nota 2:** En el Aula Virtual se encuentra un fichero con 25 ejemplos de sudokus ya codificados, *Sudokus de prueba.pl*, para poder hacer pruebas.



# Posibles mejoras de la práctica

Se valorará positivamente cualquier mejora de lo expuesto anteriormente. Algunas ideas son las siguientes:

- Mejorar la interfaz con algún elemento de tipo 'ASCII art' o enlazando Prolog con alguna interfaz gráfica.
- Realizar un estudio sistemático sobre un banco de pruebas para averiguar cuáles son los sudokus resueltos con las simplificaciones propuestas.
- Implementar nuevas reglas de simplificación y estudiar si las mismas llevan a resolver sudokus no resueltos con las reglas propuestas en la práctica.
- Buscar algún algoritmo para resolver sudokus, posiblemente ya implementado en Prolog, estudiar cómo y por qué funciona y confrontarlo con lo hecho en la práctica.
- Cualquier otra que cuente con el visto bueno del profesor.

# Detalles de la entrega

- La práctica se realizará en equipos de a lo más 3 personas.
- Además del código fuente del programa (.pl), convenientemente estructurado y comentado, se deberá incluir una breve memoria (PDF) indicando los miembros del equipo, reparto de tareas entre los mismos, grado de cumplimiento de cada uno de los requisitos, mejora realizada, errores y/o aspectos no implementados, así como las fuentes consultadas para la resolución de la práctica.
- La entrega se realizará través de la plataforma en un único fichero .zip antes de las 23:59 horas del día 15 de marzo de 2022. El nombre del fichero será NombreApellido1Apellido2.zip de uno de los integrantes del grupo. Una vez recibidos los trabajos, se publicará una fecha con la hora de defensa para cada equipo.