

# Mejorando el Agente de Sudoku

Deybby Angel Rosario Almonte

Centro en excelencia de Inteligencia Artificial, Itla

Puerto Plata, República Dominicana

deybyrosario@gmail.com

20240504@itla.edu.do

**Abstract**— Este trabajo presenta el desarrollo de un agente inteligente capaz de resolver tableros de Sudoku Diagonales mediante técnicas deterministas y sin recurrir a algoritmos de búsqueda exhaustiva. El agente, denominado *SudokuAgent*, implementa estrategias clásicas de resolución como la técnica de única posibilidad (*Only Choice*) y la eliminación de pares desnudos (*Naked Twins*), aplicando inferencia lógica para reducir el espacio de búsqueda. La solución se modela sobre una representación matricial del tablero (*SudokuMatrix*) y se evalúa con ejemplos reales, mostrando que el agente puede resolver tableros de dificultad básica y media de forma completamente autónoma, rápida y eficiente. Los resultados demuestran que, mediante el uso de reglas bien definidas y un ciclo iterativo de inferencias, es posible alcanzar la resolución completa de un Sudoku sin necesidad de técnicas de backtracking ni aprendizaje previo.

## I. INTRODUCCIÓN

La resolución automatizada de rompecabezas lógicos ha sido, durante décadas, un terreno fértil para la investigación en inteligencia artificial, especialmente en el ámbito de los sistemas basados en reglas y los agentes de razonamiento determinista. Uno de los desafíos más conocidos y accesibles dentro de esta categoría es el Sudoku, un juego de lógica numérica ampliamente reconocido por su estructura clara y sus restricciones definidas.

Resolver un Sudoku de forma automática plantea un problema interesante desde el punto de vista computacional: aunque las reglas son simples, la cantidad de combinaciones posibles en tableros parcialmente llenos puede crecer exponencialmente. Esto ha llevado a la exploración de múltiples enfoques, que van desde algoritmos de fuerza bruta hasta soluciones evolutivas o redes neuronales.

Este trabajo se enfoca en una solución intermedia: la implementación de un agente inteligente que, sin necesidad de entrenarse ni explorar árboles de búsqueda, es capaz de resolver eficazmente Sudokus usando reglas lógicas y técnicas heurísticas inspiradas en las estrategias utilizadas por jugadores humanos expertos. A través de una representación estructurada del tablero y un conjunto de técnicas de inferencia, el agente propuesto logra identificar celdas determinísticas y patrones recurrentes como los “pares desnudos”, avanzando paso a paso hasta completar el rompecabezas.

En las siguientes secciones se describe el contexto del Sudoku, las estrategias implementadas, el diseño y funcionamiento del agente inteligente, así como los resultados obtenidos tras su aplicación práctica. Se busca demostrar que la combinación de técnicas clásicas con una implementación ordenada y modular puede ofrecer una solución elegante y eficiente a este tipo de problemas lógicos.

## II. SUDOKU

### A. Que es el sudoku

El sudoku es un rompecabezas matemático que consiste en rellenar una cuadrícula de  $9 \times 9$  celdas dividida en subcuadrículas de  $3 \times 3$ , respetando tres reglas: cada fila, columna y región debe contener los números del 1 al 9 sin repeticiones. Se originó en 1970 como “Number Place” y se popularizó en Japón durante los años 80 con el nombre “Sudoku”[1].

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

Fig. 1 Imagen tomada de un tablero de Sudoku diagonal publicada en el sitio SudokuMania. Disponible en: <https://www.sudokumania.com.ar/sudoku/diagonal/>. Consultado el 7 de junio de 2025.

### B. Estrategias clave para resolver sudoku

#### 1) Estrategia “One Choice” (Celda Única):

Cuando una celda es la única en su fila, columna o región que puede contener un número específico. Por ejemplo, si los números 1-8 ya están presentes en una fila, la celda vacía debe ser 9[2]. Esta técnica se aplica mediante el proceso de eliminación, identificando números faltantes en grupos completos.

#### 2) Técnica “Naked Twins” (Gemelos Desnudos):

Identifica dos celdas en la misma unidad (fila, columna o región) que comparten exactamente los mismos dos candidatos. Estos pares permiten eliminar esos números de otras celdas en la misma unidad. Ejemplo:

Si dos celdas en una fila tienen candidatos {3,7}, Ambos números se eliminan de las demás celdas de esa fila[3].

## III. AGENTE INTELIGENTE

### A. Desarrollo del agente

El agente inteligente propuesto, denominado *SudokuAgent*, constituye el núcleo lógico del sistema encargado de resolver tableros de Sudoku. Está diseñado para trabajar directamente sobre

una instancia del objeto SudokuMatrix, la cual representa el estado actual del tablero, incluyendo sus celdas bloqueadas (valores dados inicialmente) y celdas por resolver.

Este agente implementa un enfoque determinista basado en técnicas clásicas de resolución, como la estrategia de única posibilidad (Only Choice) y la detección de pares desnudos (Naked Twins), con el objetivo de reducir progresivamente el espacio de búsqueda sin recurrir a backtracking ni algoritmos de fuerza bruta.

1) *Inicialización del Dominio Posible (\_fill\_board)*: El primer paso que ejecuta el agente es una función de propagación de restricciones sobre el tablero. Para cada celda vacía (o con valores no numéricos), se le asigna un conjunto de posibles valores numéricos del 1 al 9, calculados dinámicamente en función de la validez de cada número dentro de su fila, columna y subcuadro 3x3. Esta inicialización establece el dominio de posibles valores para cada celda, sentando las bases para la inferencia lógica posterior.

	1	2	3	4	5	6	7	8	9
A	5	24	124	8	9	3	146	1467	167
B	7	249	124	26	246	26	5	8	3
C	6	3489	348	57	457	1	4	2	79
D	124	245	9	126	3	256	7	16	8
E	123	7	12358	9	126	4	136	136	156
F	13	35	6	157	8	7	2	139	4
G	249	1	24	3	267	2679	468	5	67
H	8	345	7	156	156	56	9	1346	2
I	239	6	235	4	1257	8	13	137	17

Fig. 2. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

2) *Estrategia de Única Posibilidad (only\_choise)*: La técnica de única posibilidad se aplica iterativamente sobre el tablero. Si una celda presenta un único valor posible (es decir, una cadena de longitud uno), este valor se convierte en una jugada válida y se establece como valor fijo (entero). Esta técnica continúa aplicándose hasta que no se detectan más cambios en el tablero, o se alcanza un estado bloqueado donde no hay posibles soluciones viables para una o más celdas. Esta heurística simple pero poderosa permite resolver un amplio rango de tableros con mínima complejidad computacional.

	1	2	3	4	5	6	7	8	9
A	5	24	124	8	9	3	146	1467	167
B	7	249	124	26	246	26	5	8	3
C	6	3489	348	57	457	1	4	2	79
D	124	245	9	126	3	256	7	16	8
E	123	7	12358	9	126	4	136	136	156
F	13	35	6	157	8	7	2	139	4
G	249	1	24	3	267	2679	468	5	67
H	8	345	7	156	156	56	9	1346	2
I	239	6	235	4	1257	8	13	137	17

Fig. 3. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

3) *Identificación y Eliminación de Pares Desnudos (solve\_naked\_twins)*: El agente implementa la estrategia de Naked Twins. Esta consiste en identificar pares de celdas dentro de una misma unidad (fila, columna o caja) que comparten exactamente los mismos dos posibles valores. Al detectar estos pares gemelos, se asume que dichos valores están reservados exclusivamente para esas dos celdas. Como consecuencia, dichos valores pueden eliminarse del conjunto de posibilidades de las demás celdas dentro de la misma unidad, reduciendo el espacio de búsqueda de manera indirecta pero significativa.

	1	2	3	4	5	6	7	8	9
A	5	24	124	8	9	3	16	167	167
B	7	49	14	26	4	26	5	8	3
C	6	89	38	57	57	1	4	2	9
D	124	24	9	126	3	256	7	16	8
E	123	7	12358	9	126	4	136	136	156
F	13	35	6	15	8	7	2	139	4
G	249	1	2	3	267	269	68	5	67
H	8	35	7	156	156	56	9	1346	2
I	239	6	235	4	1257	8	13	137	1

Fig. 4. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

En la imagen se puede notar como la fila B elimina todos los valores '2' y '6' de cada celda a excepción de los gemelos desnudos.

4) *Ciclo Principal de Resolución (solve\_sudoku)*: El método principal del agente coordina la ejecución de las estrategias anteriores en un ciclo de convergencia. Se compara el estado

anterior del tablero con el actual en cada iteración. Si no hay cambios, se asume que se ha alcanzado una solución estable. Este ciclo puede resolver muchos tableros de Sudoku sin necesidad de exploración exhaustiva o búsqueda recursiva.

5) *Detección de Conflictos y Validación*: Durante el proceso, se integran validaciones dinámicas utilizando el método `_validate_play()` para garantizar que cada valor propuesto mantenga la coherencia del tablero. Si se detecta un valor inválido (por ejemplo, un número que ya existe en su fila, columna o subcuadro), el agente lanza una excepción que indica una inconsistencia, útil tanto para debugging como para la validación formal de tableros inválidos.

## B. Resultados obtenidos

Para validar la efectividad del agente inteligente desarrollado, se aplicó sobre un tablero de Sudoku con un nivel de dificultad Fácil. A continuación, se describe detalladamente el proceso de resolución paso a paso, destacando cómo las estrategias implementadas permiten avanzar progresivamente hacia la solución final sin intervención humana ni técnicas de búsqueda complejas.

1) *Estado Inicial*: El tablero inicial contiene una cantidad moderada de celdas dadas (bloqueadas), y el resto de las celdas se encuentran vacías. Las celdas dadas fueron preservadas como valores enteros fijos, mientras que las vacías se inicializaron como espacios sin contenido asignado.

	1	2	3	4	5	6	7	8	9
A	5			8	9	3			
B	7						5	8	3
C	6					1		2	
D			9		3		7		8
E		7		9		4			
F			6		8		2		4
G		1		3				5	
H	8		7				9		2
I		6		4		8			

Fig. 5. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

2) *Generación de Dominios Posibles (`_fill_board`)*: El agente recorrió todas las celdas vacías del tablero y calculó para cada una el conjunto de valores posibles del 1 al 9, filtrando aquellos que violaban las reglas del Sudoku. Por ejemplo, si una celda estaba en una fila que ya contenía los números 2, 4 y 9, estos eran excluidos automáticamente de sus posibilidades. Como resultado, la mayoría de las celdas quedaron con entre 1 y 5 posibles valores.

	1	2	3	4	5	6	7	8	9
A	5	24	124	8	9	3	146	1467	167
B	7	249	124	26	246	26	5	8	3
C	6	3489	348	57	457	1	4	2	79
D	124	245	9	126	3	256	7	16	8
E	123	7	12358	9	126	4	136	136	156
F	13	35	6	157	8	7	2	139	4
G	249	1	24	3	267	2679	468	5	67
H	8	345	7	156	156	56	9	1346	2
I	239	6	235	4	1257	8	13	137	17

Fig. 6. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

3) *Aplicación de la Técnica “Única Posibilidad” (`only_choise`)*: Una vez establecidos los dominios iniciales, el agente comenzó a identificar celdas en las que solo era posible colocar un único valor. Estas celdas fueron inmediatamente resueltas, convirtiendo su valor string en un número entero.

	1	2	3	4	5	6	7	8	9
A	5	24	124	8	9	3	146	1467	167
B	7	249	124	26	246	26	5	8	3
C	6	3489	348	57	457	1	4	2	79
D	124	245	9	126	3	256	7	16	8
E	123	7	12358	9	126	4	136	136	156
F	13	35	6	157	8	7	2	139	4
G	249	1	24	3	267	2679	468	5	67
H	8	345	7	156	156	56	9	1346	2
I	239	6	235	4	1257	8	13	137	17

Fig. 7. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

Ejemplo: En la celda D7, el único valor posible era '4'. El agente validó que colocar un 4 en esa posición no generaba conflictos, y por tanto lo estableció como valor definitivo.

Este proceso se repitió iterativamente. En cada ciclo, nuevos valores únicos aparecían como consecuencia de las restricciones impuestas por los valores ya colocados. El agente continuó aplicando esta técnica hasta que no fue posible resolver más celdas mediante única posibilidad.

4) *Identificación de Pares Desnudos (`_solve_naked_twins`)*: Al agotarse la estrategia anterior, el agente pasó a buscar patrones avanzados, como los pares desnudos. En múltiples unidades del tablero (por ejemplo, la fila B), se identificaron pares de celdas con

exactamente los mismos dos valores posibles, como '26' en B4 y B6.

	1	2	3	4	5	6	7	8	9
A	5	24	124	8	9	3	16	167	167
B	7	49	14	26	4	26	5	8	3
C	6	89	38	57	57	1	4	2	9
D	124	24	9	126	3	256	7	16	8
E	123	7	12358	9	126	4	136	136	156
F	13	35	6	15	8	7	2	139	4
G	249	1	2	3	267	269	68	5	67
H	8	35	7	156	156	56	9	1346	2
I	239	6	235	4	1257	8	13	137	1

Fig. 8. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

El agente reconoció que esos dos valores debían estar obligatoriamente en esas dos posiciones, y por tanto los eliminó del dominio de las demás celdas de esa unidad. Esto provocó ocasionó nuevas simplificaciones en el tablero.

4) *Iteración y Convergencia* (*solve naked twins*): El proceso de resolución alternó entre las técnicas de única posibilidad y eliminación por pares desnudos, en ciclos consecutivos. En cada iteración se evaluaba si el tablero había cambiado respecto a su estado anterior. Una vez que no se detectaron más cambios, y el número de celdas resueltas alcanzó las 81, el agente detuvo la ejecución.

	1	2	3	4	5	6	7	8	9
A	5	24	14	8	9	3	16	167	67
B	7	9	1	26	4	2	5	8	3
C	6	8	38	57	57	1	4	2	9
D	124	24	9	26	3	56	7	16	8
E	123	7	1358	9	6	4	136	136	5
F	13	35	6	15	8	7	2	139	4
G	49	1	2	3	67	9	8	5	67
H	8	35	7	15	156	56	9	346	2
I	9	6	35	4	257	8	3	37	1

Fig. 9. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

	1	2	3	4	5	6	7	8	9
A	5	24	4	8	9	3	16	167	7
B	7	9	1	6	4	2	5	8	3
C	6	8	3	57	57	1	4	2	9
D	124	24	9	2	3	5	7	16	8
E	123	7	38	9	6	4	1	13	5
F	13	35	6	15	8	7	2	139	4
G	4	1	2	3	7	9	8	5	67
H	8	35	7	156	15	56	9	4	2
I	9	6	5	4	257	8	3	7	1

Fig. 10. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

	1	2	3	4	5	6	7	8	9
A	5	2	4	8	9	3	6	16	7
B	7	9	1	6	4	2	5	8	3
C	6	8	3	57	5	1	4	2	9
D	1	4	9	2	3	5	7	6	8
E	23	7	8	9	6	4	1	3	5
F	13	35	6	1	8	7	2	39	4
G	4	1	2	3	7	9	8	5	6
H	8	3	7	15	15	6	9	4	2
I	9	6	5	4	2	8	3	7	1

Fig. 11. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

	1	2	3	4	5	6	7	8	9
A	5	2	4	8	9	3	6	1	7
B	7	9	1	6	4	2	5	8	3
C	6	8	3	7	5	1	4	2	9
D	1	4	9	2	3	5	7	6	8
E	2	7	8	9	6	4	1	3	5
F	3	5	6	1	8	7	2	9	4
G	4	1	2	3	7	9	8	5	6
H	8	3	7	5	1	6	9	4	2
I	9	6	5	4	2	8	3	7	1

Fig. 12. Estado parcial del tablero de Sudoku generado por el agente

inteligente SudokuAgent durante el proceso de resolución automática.

5) *Resultado Final:* El agente logró resolver completamente el tablero de Sudoku Diagonal propuesto en 1.6 segundos (graficando el paso a paso) sin recurrir a backtracking. Todas las celdas fueron asignadas con valores válidos, manteniendo la consistencia de las reglas del juego (sin duplicados en filas, columnas o subcuadros).

	1	2	3	4	5	6	7	8	9
A	5	2	4	8	9	3	6	1	7
B	7	9	1	6	4	2	5	8	3
C	6	8	3	7	5	1	4	2	9
D	1	4	9	2	3	5	7	6	8
E	2	7	8	9	6	4	1	3	5
F	3	5	6	1	8	7	2	9	4
G	4	1	2	3	7	9	8	5	6
H	8	3	7	5	1	6	9	4	2
I	9	6	5	4	2	8	3	7	1

Fig. 13. Estado parcial del tablero de Sudoku generado por el agente inteligente SudokuAgent durante el proceso de resolución automática.

#### IV.CONCLUSIONES

El desarrollo del agente inteligente SudokuAgent demuestra que es posible resolver tableros de Sudoku de forma totalmente autónoma mediante un enfoque determinista, sin necesidad de algoritmos de búsqueda recursiva ni técnicas de aprendizaje automático. A través de la implementación de estrategias clásicas como la técnica de única posibilidad (Only Choice) y la eliminación de pares desnudos (Naked Twins), el agente es capaz de reducir progresivamente el espacio de búsqueda y completar el tablero mediante inferencia lógica.

Los resultados obtenidos validan la efectividad del modelo propuesto en tableros de dificultad baja a media, con tiempos de ejecución mínimos y sin errores de validación. El sistema presenta un comportamiento estable, replicable y explicable, lo que lo hace especialmente útil como herramienta educativa o como base para soluciones más complejas.

Este enfoque confirma que la inteligencia artificial no siempre requiere aprendizaje profundo o grandes volúmenes de datos: en muchos casos, basta con una lógica bien estructurada y estrategias fundamentadas para construir agentes eficientes y confiables.

Como líneas futuras, se plantea la incorporación de nuevas técnicas heurísticas avanzadas (como Hidden Pairs, X-Wing o Swordfish)/ Esta mejora permitirá escalar el agente hacia escenarios más complejos.

#### RECONOCIMIENTOS

Quiero expresar mi agradecimiento a mis compañeros Sarah Peña, Jean Castellanos, Geraldo Cabrera, Carlos Roa, Darwin Mendez y Juan David Vázques. Su valiosa colaboración y apoyo

en el proceso fueron fundamentales para el éxito de este proyecto.

#### REFERENCIAS

- [1] Wikipedia contributors, Sudoku, Wikipedia. [Online]. Available: <https://es.wikipedia.org/wiki/Sudoku>
- [2] SudokuOnline.io, Técnicas de resolución avanzadas para Sudoku. [Online]. Available: <https://www.sudokuonline.io/es/consejos/tecnicas-resolucion-sudoku-avanzadas>
- [3] Toolify.ai, “Resolviendo Sudoku con IA | Aprendiendo Inteligencia 1”, Updated Feb. 19, 2024. [Online]. Available: <https://www.toolify.ai/es/ai-news-es/resolviendo-sudoku-con-ia-aprendiendo-inteligencia-1-1317761>