Taller 1 — Modelado y resolución de CSP en MiniZinc

Estudio de Sudoku, Kakuro, Secuencia Mágica, Acertijo Lógico, Reunión y Rectángulo

John Freddy Belalcazar
Samuel Galindo Cuevas
Nicolas Herrera Marulanda

1 de octubre de 2025

Índice

1. Sodoku

El *Sudoku* es un rompecabezas lógico en una cuadrícula 9×9 dividida en nueve cajas 3×3. El tablero se entrega con algunas celdas ya llenas (*pistas*) y el objetivo es completar todas las celdas con dígitos del 1 al 9 cumpliendo simultáneamente: (i) en cada fila no se repiten dígitos, (ii) en cada columna no se repiten, y (iii) en cada caja 3×3 no se repiten.

El *Sudoku clásico 9×9* puede modelarse naturalmente como un *Problema de Satisfacción de Restricciones* (CSP): cada celda es una variable con dominio $\{1, \ldots, 9\}$, y las reglas del juego se expresan como restricciones sobre filas, columnas y subcuadrículas 3×3.

1.1. Modelo

Parámetros

P1 — N**:** Tamaño del tablero (lado). En el Sudoku clásico, $N = 9$.

$$N = 9.$$

P2 — S**:** Índices válidos de filas y columnas.

$$S = \{1, \ldots, N\}.$$

P3 — DIG**:** Conjunto de dígitos permitidos en cada celda.

$$DIG = \{1, \ldots, N\}.$$

P4 — G**:** Matriz de *pistas*; 0 indica celda vacía y un valor en DIG fija la celda.

$$G \in \{0, \ldots, N\}^{S \times S}, \qquad G_{r,c} = 0 \text{ (vacía)}, \quad G_{r,c} \in DIG \text{ (valor fijo)}.$$

Variables

V1 — $X_{r,c}$**:** Valor de la celda en fila r y columna c.

$$X_{r,c} \in DIG, \qquad (r, c) \in S \times S.$$

Restricciones principales

R1 — **Fijación de pistas:** Las pistas son hechos inmutables: si hay pista en (r, c), la celda queda fijada.

$$\forall (r, c) \in S \times S : \quad (G_{r,c} > 0) \;\Rightarrow\; (X_{r,c} = G_{r,c}).$$

R2 — **No repetición por fila:** En cada fila, los nueve valores deben ser todos distintos.

$$\forall r \in S : \quad \mathit{all\_different}([\, X_{r,c} \mid c \in S \,]).$$

R3 — **No repetición por columna:** En cada columna, los nueve valores deben ser todos distintos.

$$\forall c \in S : \quad \mathit{all\_different}([\, X_{r,c} \mid r \in S \,]).$$

R4 — **No repetición por caja** 3×3**:** En cada subcuadrícula 3×3, los nueve valores deben ser todos distintos.

$$\forall b_r, b_c \in \{0, 1, 2\} : \quad \mathit{all\_different}([\, X_{3b_r+i,\, 3b_c+j} \mid i, j \in \{1, 2, 3\} \,]).$$

Restricciones redundantes

R5 — Suma por fila = 45: Cada fila contiene los dígitos 1.,9 sin repetición; por tanto su suma es 45. Aporta poda lineal cuando faltan pocas celdas.

$$\forall r \in S : \sum_{c \in S} X_{r,c} = 45.$$

R6 — Suma por columna = 45: Análogo para columnas; refuerza la propagación vertical.

$$\forall c \in S : \sum_{r \in S} X_{r,c} = 45.$$

R7 — Suma por caja 3×3 = 45: Cada subcuadrícula 3×3 reúne los nueve dígitos sin repetición; su suma también es 45. Útil para cerrar cajas cuando faltan pocas celdas.

$$\forall\, b_r, b_c \in \{0,1,2\} : \sum_{i=1}^{3} \sum_{j=1}^{3} X_{3b_r+i,\, 3b_c+j} = 45.$$

1.2. Implementación

Archivos y organización

- `sudoku.mzn`: modelo completo.
- `tests/*.dzn`: instancias con la matriz G (0 = vacío).

Ramificación solo en celdas vacías

Se construye el arreglo $\mathcal{B} = \{X_{r,c} \mid G_{r,c} = 0\}$ para ramificar únicamente sobre celdas no fijadas por pistas; así se evita trabajo sobre variables ya determinadas.

Restricciones redundantes

Las ecuaciones de suma 45 y suma de cuadrados 285 para filas/columnas se activan siempre. Son *redundantes*: no cambian las soluciones posibles pero fortalecen la propagación.

Ruptura de simetría

En Sudoku con *pistas fijas* la mayoría de simetrías del modelo (permutaciones de filas/columnas/bandas, renombrar dígitos, transposición) *no* preservan la instancia: moverían las pistas a posiciones distintas. Por ello *no imponemos* restricciones de ruptura de simetría globales, ya que podrían eliminar la única solución válida de la instancia.

Estrategias de búsqueda

Para las pruebas sobre el modelo de Sudoku se plantean diferentes combinaciones de heurísticas, con el objetivo de analizar su impacto en el tamaño del árbol de búsqueda y el tiempo de resolución. Se considerarán, de manera tentativa, las siguientes familias:

Heurísticas de selección de variables

- **first_fail**: prioriza la variable con dominio más pequeño.
- **dom_w_deg**: usa la razón dominio/grado ponderado.
- **input_order**: sigue el orden natural de las variables.
- **sin anotación explícita**: dejar que el solver elija su estrategia por defecto.

Heurísticas de selección de valores

- **indomain_min**: intenta primero el valor mínimo del dominio.
- **indomain_split**: divide el dominio y explora por mitades.
- **indomain_median**: comienza por la mediana.
- **sin anotación explícita**: que el solver decida por defecto.

Solvers y métricas Se evaluarán combinaciones representativas con **Gecode** y **Chuffed**, registrando: *tiempo de resolución*, *nodos explorados*, *fallos*, *profundidad*, *reinicios* (si aplica) y *memoria pico*. El objetivo es comparar rendimiento entre estrategias y verificar la robustez del modelo frente a distintos motores de propagación.

1.3. Pruebas

Se evaluó el modelo sobre una batería de instancias `.dzn` con distinta dificultad (variando la cantidad y distribución de pistas). En cada corrida se registraron *tiempo*, *nodos*, *fallos*, *profundidad* y *número de soluciones*. A continuación se presenta una tabla plantilla para consolidar dichos resultados.

Cuadro 1: Resultados de pruebas.

Archivo	Solver	Var heur	Val heur	time	nodes	fail	depth
example-e.dzn	Chuffed	first_fail	indomain_min	0.000	0	0	0
example-e.dzn	Gecode	dom_w_deg	indomain_split	0.000	0	0	0

1.4. Arboles de búsqueda

En esta sección se presentan las **visualizaciones del árbol de búsqueda** generadas por el modelo de Sudoku bajo distintas combinaciones de *solver* y *heurísticas*. Cada imagen muestra la estructura explorada para una instancia específica.

1.5. Análisis

Con base en la Tabla 1.3 y las figuras del árbol de búsqueda, comparamos las configuraciones por tres criterios: **nodes**, **failures** y **tiempo**. En general, el modelo con *all_different + redundantes* tiende a producir menos fallos; y, entre heurísticas, *dom_w_deg + indomain_split* suele explorar menos nodos que *first_fail + indomain_min* en instancias más difíciles, mientras que *first_fail + indomain_min* es un buen baseline cuando hay más pistas.

- **Tamaño del árbol (nodes).** *[Estrategía A]* explora *[_ % menos]* nodos que *[Estrategía B]* en *[instancias]*.
- **Backtracking (failures).** *[Estrategía A]* reduce los fallos frente a *[Estrategía B]*, coherente con mayor propagación.
- **Profundidad/forma.** Árboles más "anchos y podados" (fallos tempranos) se asocian a menos *nodes*; *[Fig. X]* vs. *[Fig. Y]*.
- **Tiempo.** Depende de *nodes* y del coste por nodo: aunque *[Estrategía A]* tenga menos nodos, *[Estrategía B]* puede igualar si su coste/nodo es menor (*[___]* s vs. *[___]* s).
- **Solver.** *[Chuffed/Gecode]* rinde mejor en *[tipo de instancia]* con *[heurística]*, según *[___]* en *nodes/failures/tiempo*.

Conclusión breve. Para *[instancias difíciles]*, recomendamos *dom_w_deg + indomain_split*; para *[fáciles/medias]*, *first_fail + indomain_min* es suficiente. Active siempre las redundantes.

1.6. Conclusiones

- A
- B
- C
- D

2. Kakuro

Introducción al problema y alcance del modelado. Supuestos y parámetros clave.a

2.1. Modelo

Definición de variables, dominios y restricciones principales. Justificación del modelo.

2.2. Detalles de implementación

Restricciones redundantes, rompimiento de simetrías y decisiones técnicas.

2.3. Pruebas

Casos de prueba, entradas, métricas y tablas o figuras de apoyo.

2.4. Árboles de búsqueda

Nodos explorados, fallos, tiempos y efecto de estrategias de distribución.

2.5. Análisis

Comparación de variantes y discusión de resultados.

2.6. Conclusiones

Lecciones, limitaciones y trabajo futuro.

3. Secuencia Mágica

Introducción al problema y alcance del modelado. Supuestos y parámetros clave.

3.1. Modelo

Definición de variables, dominios y restricciones principales. Justificación del modelo.

3.2. Detalles de implementación

Restricciones redundantes, rompimiento de simetrías y decisiones técnicas.

3.3. Pruebas

Casos de prueba, entradas, métricas y tablas o figuras de apoyo.

3.4. Árboles de búsqueda

Nodos explorados, fallos, tiempos y efecto de estrategias de distribución.

3.5. Análisis

Comparación de variantes y discusión de resultados.

3.6. Conclusiones

Lecciones, limitaciones y trabajo futuro.

4. Acertijo Lógico

Introducción al problema y alcance del modelado. Supuestos y parámetros clave.

4.1. Modelo

Definición de variables, dominios y restricciones principales. Justificación del modelo.

4.2. Detalles de implementación

Restricciones redundantes, rompimiento de simetrías y decisiones técnicas.

4.3. Pruebas

Casos de prueba, entradas, métricas y tablas o figuras de apoyo.

4.4. Árboles de búsqueda

Nodos explorados, fallos, tiempos y efecto de estrategias de distribución.

4.5. Análisis

Comparación de variantes y discusión de resultados.

4.6. Conclusiones

Lecciones, limitaciones y trabajo futuro.

5. Ubicación de personas en una reunión

Un grupo de N personas desea tomarse una fotografía en una sola fila. Algunas parejas de personas imponen preferencias de proximidad: *adyacencia* (`next`), *no adyacencia* (`separate`) y *cota máxima de distancia* (`distance`), que limitan cuántas personas pueden quedar entre dos individuos.

Este problema puede modelarse como un *Problema de Satisfacción de Restricciones* (CSP): cada persona debe ocupar exactamente una posición en la fila y las preferencias se expresan como restricciones sobre las posiciones relativas (por ejemplo, $|\text{pos}(A) - \text{pos}(B)| = 1$ para `next`, $|\text{pos}(A) - \text{pos}(B)| \geq 2$ para `separate`, y $|\text{pos}(A) - \text{pos}(B)| \leq M + 1$ para `distance`). El objetivo es encontrar cualquier orden que satisfaga simultáneamente todas las preferencias, o certificar que no existe.

5.1. Modelo

Definición de variables, dominios y restricciones principales. Justificación del modelo.

5.2. Detalles de implementación

Restricciones redundantes, rompimiento de simetrías y decisiones técnicas.

5.3. Pruebas

Casos de prueba, entradas, métricas y tablas o figuras de apoyo.

5.4. Árboles de búsqueda

Nodos explorados, fallos, tiempos y efecto de estrategias de distribución.

5.5. Análisis

Comparación de variantes y discusión de resultados.

5.6. Conclusiones

Lecciones, limitaciones y trabajo futuro.

6. Construcción de un rectángulo

Introducción al problema y alcance del modelado. Supuestos y parámetros clave.

6.1. Modelo

Definición de variables, dominios y restricciones principales. Justificación del modelo.

6.2. Detalles de implementación

Restricciones redundantes, rompimiento de simetrías y decisiones técnicas.

6.3. Pruebas

Casos de prueba, entradas, métricas y tablas o figuras de apoyo.

6.4. Árboles de búsqueda

Nodos explorados, fallos, tiempos y efecto de estrategias de distribución.

6.5. Análisis

Comparación de variantes y discusión de resultados.

6.6. Conclusiones

Lecciones, limitaciones y trabajo futuro.