

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

12 de octubre de 2025

Índice

1. Sudoku	2
1.1. Modelo	2
1.2. Implementación	3
1.3. Pruebas	3
1.4. Árboles de búsqueda	4
1.5. Análisis y conclusiones	5
2. Kakuro	5
2.1. Modelo	5
2.2. Detalles de implementación	6
2.3. Pruebas	6
2.4. Árboles de búsqueda	6
2.5. Análisis y conclusiones	6
3. Secuencia Mágica	6
3.1. Modelo	6
3.2. Detalles de implementación	6
3.3. Pruebas	6
3.4. Árboles de búsqueda	7
3.5. Análisis y conclusiones	7
4. Acertijo Lógico	7
4.1. Modelo	7
4.2. Detalles de implementación	7
4.3. Pruebas	7
4.4. Árboles de búsqueda	7

4.5. Análisis y conclusiones	8
5. Ubicación de personas en una reunión	8
5.1. Modelo	8
5.2. Detalles de implementación	9
5.3. Pruebas	10
5.4. Árboles de búsqueda	11
5.5. Análisis y conclusiones	11
6. Construcción de un rectángulo	11
6.1. Modelo	11
6.2. Detalles de implementación	11
6.3. Pruebas	11
6.4. Árboles de búsqueda	11
6.5. Análisis y conclusiones	12

1. Sudoku

Puzzle en una grilla 9×9 dividida en nueve cajas 3×3 . Se entregan algunas celdas como *pistas* y el objetivo es completar las restantes con dígitos 1–9 de modo que en cada fila, en cada columna y en cada caja 3×3 no se repita ningún dígito.

1.1. Modelo

Parámetros

P1 — N : Tamaño del tablero. En Sudoku clásico, $N = 9$.

P2 — S : Índices de filas/columnas: $S = \{1, \dots, N\}$.

P3 — DIG : Dígitos válidos: $DIG = \{1, \dots, N\}$.

P4 — G : Matriz de pistas $G \in \{0, \dots, N\}^{S \times S}$; $G_{r,c} = 0$ indica vacío y $G_{r,c} \in DIG$ fija la celda.

Variables

V1 — $X_{r,c}$: Valor de la celda (r, c) : $X_{r,c} \in DIG$, para $r, c \in S$.

Restricciones principales

R1 — **Pistas fijas**: Si hay pista, se respeta: $(G_{r,c} > 0) \Rightarrow X_{r,c} = G_{r,c}$ para todo $r, c \in S$.

R2 — **Filas sin repetición**: $\forall r \in S : all_different([X_{r,c} \mid c \in S])$.

R3 — Columnas sin repetición: $\forall c \in S : all_different([X_{r,c} \mid r \in S])$.

R4 — Cajas 3×3 sin repetición: $\forall b_r, b_c \in \{0, 1, 2\} : all_different([X_{3b_r+i, 3b_c+j} \mid i, j \in \{1, 2, 3\}])$.

Restricciones redundantes (opcionales)

R5 — Suma por fila = 45: $\forall r \in S : \sum_{c \in S} X_{r,c} = 45$. Aporta poda lineal cuando faltan pocas celdas en la fila.

R6 — Suma por columna = 45: $\forall c \in S : \sum_{r \in S} X_{r,c} = 45$. Refuerza la propagación vertical.

R7 — Suma por caja = 45: $\forall b_r, b_c \in \{0, 1, 2\} : \sum_{i=1}^3 \sum_{j=1}^3 X_{3b_r+i, 3b_c+j} = 45$. Útil para cerrar subcuadrículas casi completas.

1.2. Implementación

Modelo

Definimos el conjunto de ramificación $\mathcal{B} = \{X_{r,c} \mid G_{r,c} = 0\}$ y sólo exploramos celdas sin pista. Así evitamos ramificar en valores ya fijados por G y concentramos la búsqueda donde hay incertidumbre, reduciendo el árbol sin afectar la corrección.

Restricciones redundantes

Añadimos las sumas a 45 como poda ligera: no cambian el conjunto de soluciones y, en teoría, deberían ayudar a detectar inconsistencias temprano, reduciendo *nodos* y *fallos*.

Ruptura de simetría

Las pistas G fijan la instancia y aplicar simetrías del Sudoku (permutar filas, columnas o bandas, renombrar dígitos, transponer) movería o alteraría G . Para no arriesgar la solución válida, no añadimos rompedores de simetría.

1.3. Pruebas

Las instancias `test_01`, `test_02` y `test_03` siguen una dificultad *aprox.* creciente; no obstante, según solver/heurística `test_02` puede comportarse tan difícil como `test_03`, algo visible en *nodes/fail* y la profundidad del árbol.

Tabla 1: Resultados de pruebas **con** restricciones redundantes.

Archivo	Solver	Var	heur	Val	heur	time	nodes	fail	depth
test_01	Chuffed	first_fail		indomain_min		2.000e−03	5	3	2
test_01	Chuffed	dom_w_deg		indomain_split		1.000e−03	7	4	3
test_01	Chuffed	input_order		indomain_min		1.000e−03	6	4	2
test_01	Gecode	first_fail		indomain_min		9.761e−03	61	28	7
test_01	Gecode	dom_w_deg		indomain_split		1.065e−03	43	20	8
test_01	Gecode	input_order		indomain_min		1.937e−03	131	65	7
test_02	Chuffed	first_fail		indomain_min		8.000e−03	418	367	13
test_02	Chuffed	dom_w_deg		indomain_split		1.200e−02	550	496	14
test_02	Chuffed	input_order		indomain_min		1.200e−02	567	537	11
test_02	Gecode	first_fail		indomain_min		1.359e−02	1964	979	17
test_02	Gecode	dom_w_deg		indomain_split		7.073e−03	703	349	21
test_02	Gecode	input_order		indomain_min		4.939e−02	7273	3633	20
test_03	Chuffed	first_fail		indomain_min		1.000e−03	28	22	6
test_03	Chuffed	dom_w_deg		indomain_split		1.200e−02	337	324	9
test_03	Chuffed	input_order		indomain_min		7.000e−03	352	341	7
test_03	Gecode	first_fail		indomain_min		7.257e−03	727	360	11
test_03	Gecode	dom_w_deg		indomain_split		2.804e−03	218	107	11
test_03	Gecode	input_order		indomain_min		1.308e−02	1452	723	15

Tabla 2: Resultados de pruebas **sin** restricciones redundantes.

Archivo	Solver	Var	heur	Val	heur	time	nodes	fail	depth
test_01	Chuffed	first_fail		indomain_min		1.000e−03	5	3	2
test_01	Chuffed	dom_w_deg		indomain_split		1.000e−03	7	4	3
test_01	Chuffed	input_order		indomain_min		1.000e−03	6	4	2
test_01	Gecode	first_fail		indomain_min		7.911e−03	61	28	7
test_01	Gecode	dom_w_deg		indomain_split		2.536e−03	41	19	7
test_01	Gecode	input_order		indomain_min		1.918e−03	131	65	7
test_02	Chuffed	first_fail		indomain_min		7.000e−03	417	369	13
test_02	Chuffed	dom_w_deg		indomain_split		9.000e−03	498	459	14
test_02	Chuffed	input_order		indomain_min		1.100e−02	565	527	11
test_02	Gecode	first_fail		indomain_min		1.150e−02	1964	979	17
test_02	Gecode	dom_w_deg		indomain_split		3.780e−03	457	226	18
test_02	Gecode	input_order		indomain_min		4.329e−02	7273	3633	20
test_03	Chuffed	first_fail		indomain_min		1.000e−03	28	22	6
test_03	Chuffed	dom_w_deg		indomain_split		7.000e−03	338	324	9
test_03	Chuffed	input_order		indomain_min		6.000e−03	351	340	7
test_03	Gecode	first_fail		indomain_min		6.152e−03	727	360	11
test_03	Gecode	dom_w_deg		indomain_split		1.854e−03	151	72	11
test_03	Gecode	input_order		indomain_min		1.172e−02	1452	723	15

1.4. Árboles de búsqueda

Se capturaron con *Gecode Gist*. Mostramos **solo** corridas *con* redundancias, pues los árboles *con/sin* ellos se ven practicamente indenticos.

Árboles de búsqueda (Google Drive).

1.5. Análisis y conclusiones

La comparación entre solvers mostró que Chuffed resolvió el Sudoku en tiempos menores que Gecode. Chuffed aprovecha propagación y aprendizaje de conflictos para recortar el árbol de búsqueda, lo que a menudo le da ventaja de velocidad, aunque en algunos casos exploró un número de nodos/fallos comparable o incluso mayor. Gecode, en cambio, depende más de la heurística de búsqueda elegida: con heurísticas potentes también pudo resolver eficazmente, pero en general sus tiempos fueron superiores a los de Chuffed.

Entre las estrategias de búsqueda, la heurística `wdeg_split` (dom/wdeg con división de dominio) produjo los mejores resultados en nodos y fallos. Esta heurística prioriza las variables que participan en más conflictos y las aborda primero, reduciendo drásticamente el espacio de búsqueda. La estrategia `ff_min` (first_fail con valor mínimo) también ayudó a disminuir los retrocesos al elegir celdas con dominios pequeños, pero su beneficio fue menor que el de `wdeg_split`. En contraste, `inorder_min` (orden de entrada con valor mínimo) resultó la menos eficiente: exploró muchos más nodos y alcanzó una profundidad similar (hasta completar el tablero) sin predecir bien dónde ocurren los fallos, lo que se tradujo en más retrocesos y tiempos mayores.

Finalmente, se observó que añadir las restricciones redundantes de suma (fila/columna/caja suman 45) no aportó mejoras y, en varios casos, introdujo un ligero sobrecoste. Aunque la literatura sugiere que las redundancias pueden ayudar, en nuestro modelo de Sudoku el propagador de *all_different* ya realiza una poda muy fuerte, de modo que las sumas apenas añaden información y sí más trabajo de propagación. En nuestras pruebas, las métricas con redundancias fueron en general similares o algo peores (ligeros aumentos de tiempo y nodos), especialmente con estrategias como `wdeg_split`. Con heurísticas simples tampoco se observó un beneficio claro. En conjunto, el modelo con sumas no redujo el backtracking ni el tiempo de resolución, por lo que preferimos dejarlas desactivadas por defecto.

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 y conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur at dui sed justo viverra ultrices. Integer a nisl id enim ornare dictum. Mauris non lectus vel turpis posuere tincidunt. In hac habitasse platea dictumst. Donec et urna non velit tempus vulputate.

Suspendisse potenti. Phasellus lacinia, arcu et gravida pharetra, tortor nisl iaculis augue, eget porta libero sapien in odio. Sed imperdiet, turpis at facilisis varius, arcu velit aliquet justo, vitae convallis lorem ipsum id urna. Cras ut sem vel ex sagittis bibendum.

Praesent euismod, sapien a cursus molestie, risus metus feugiat lorem, vitae gravida enim felis id magna. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

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 y conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur at dui sed justo viverra ultrices. Integer a nisl id enim ornare dictum. Mauris non lectus vel turpis posuere tincidunt. In hac habitasse platea dictumst. Donec et urna non velit tempus vulputate.

Suspendisse potenti. Phasellus lacinia, arcu et gravida pharetra, tortor nisl iaculis augue, eget porta libero sapien in odio. Sed imperdiet, turpis at facilisis varius, arcu velit aliquet justo, vitae convallis lorem ipsum id urna. Cras ut sem vel ex sagittis bibendum.

Praesent euismod, sapien a cursus molestie, risus metus feugiat lorem, vitae gravida enim felis id magna. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

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 y conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur at dui sed justo viverra ultrices. Integer a nisl id enim ornare dictum. Mauris non lectus vel turpis posuere tincidunt. In hac habitasse platea dictumst. Donec et urna non velit tempus vulputate.

Suspendisse potenti. Phasellus lacinia, arcu et gravida pharetra, tortor nisl iaculis augue, eget porta libero sapien in odio. Sed imperdiet, turpis at facilisis varius, arcu velit aliquet justo, vitae convallis lorem ipsum id urna. Cras ut sem vel ex sagittis bibendum.

Praesent euismod, sapien a cursus molestie, risus metus feugiat lorem, vitae gravida enim felis id magna. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

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.

5.1. Modelo

Parámetros

P1 — N : Número de personas a ubicar. $N \in \mathbb{Z}_{\geq 1}$.

P2 — S : Índices válidos para personas. $S = \{1, \dots, N\}$.

P3 — POS : Conjunto de posiciones disponibles en la fila. $POS = \{1, \dots, N\}$.

P4 — **personas**: Vector de nombres. **personas** $\in \text{String}^S$.

P5 — $K_{\text{next}}, K_{\text{sep}}, K_{\text{dist}}$: Cantidad de preferencias de cada tipo. $K_{\text{next}}, K_{\text{sep}}, K_{\text{dist}} \in \mathbb{Z}_{\geq 0}$.

P6 — **NEXT, SEP, DIST**: Matrices de preferencias: **NEXT** $\in S^{K_{\text{next}} \times 2}$, **SEP** $\in S^{K_{\text{sep}} \times 2}$, **DIST** $\in (S \times S \times \{0, \dots, N-2\})^{K_{\text{dist}}}$. Cada fila codifica un par de personas y, en **DIST**, una cota M de separación.

Variables

V1 — POS_OF_p : Posición que ocupa la persona p . $POS_OF_p \in POS$, $p \in S$.

V2 — PER_AT_i : Persona ubicada en la posición i . $PER_AT_i \in S$, $i \in POS$.

Restricciones principales

- R1 — Biección (canalización):** La asignación es una permutación válida: cada persona ocupa exactamente una posición y cada posición contiene exactamente una persona. $inverse(POS_OF, PER_AT)$.
- R2 — Preferencias next(A, B):** A y B deben quedar adyacentes. $\forall(A, B) \in \text{NEXT} : |POS_OF_A - POS_OF_B| = 1$.
- R3 — Preferencias separate(A, B):** A y B no pueden quedar adyacentes. $\forall(A, B) \in \text{SEP} : |POS_OF_A - POS_OF_B| \geq 2$.
- R4 — Preferencias distance(A, B, M):** A lo sumo M personas entre A y B , equivalente a cota sobre distancia de posiciones. $\forall(A, B, M) \in \text{DIST} : |POS_OF_A - POS_OF_B| \leq M + 1$.

Restricciones de simetrías

- R5 — Rompimiento de simetría izquierda–derecha:** Las soluciones reflejadas son equivalentes; para evitar duplicados, se fija $PER_AT_1 < PER_AT_N$.

5.2. Detalles de implementación

Modelo

Se usan dos vistas de la permutación: POS_OF (persona->posición) y PER_AT (posición->persona), enlazadas con $inverse$. Esta canalización refuerza la propagación respecto a usar solo una vista con *all_different*, simplifica la salida (recorriendo PER_AT en orden) y facilita la ruptura de simetría comparando extremos.

Restricciones redundantes

Se decidió no incorporar redundancias internas adicionales. Con $inverse$ y las restricciones principales de adyacencia, no adyacencia y distancia máxima, el núcleo del modelo ya resulta suficientemente fuerte; duplicar restricciones sobre la vista dual o añadir igualdades tautológicas no introdujo mejoras y sí sobrecarga. Tampoco se identificaron refuerzos simples y universales que aportaran poda adicional sin alterar la semántica; equivalencias locales habituales ya quedan implícitas en la formulación. Además, en pruebas A/B con distintos solvers y heurísticas, activar “redundancias” o “sanity checks” dentro del modelo no mostró reducciones medibles en nodos, fallos ni tiempo.

Ruptura de simetría

Existe simetría de reflexión izquierda–derecha: invertir la fila produce otra solución equivalente. Para evitar duplicados se fija un orden canónico comparando los extremos (`PER_AT[1]` frente a `PER_AT[N]`). Esto reduce la búsqueda sin afectar satisfacibilidad ni óptimos, siempre que no haya reglas que distingan explícitamente los extremos.

5.3. Pruebas

Las instancias `test_01`, `test_02` y `test_03` escalan en dificultad: `test_01` es *UNSAT* para evidenciar entradas inviables; `test_02` explota la simetría espejo (sensible a romper o no la simetría); y `test_03` es factible pero más exigente por restricciones solapadas.

Tabla 3: Resultados de pruebas **con** restricciones de simetría.

Archivo	Solver	Var	heur	Val	heur	time	nodes	fail	depth
test_01	chuffed	dom_w_deg	indomain_split	2.200e-02		1055	273	13	
test_02	chuffed	dom_w_deg	indomain_split	3.000e-03		83	77	8	
test_03	chuffed	dom_w_deg	indomain_split	8.000e-03		531	426	9	
test_01	chuffed	first_fail	indomain_min	1.800e-02		180	180	3	
test_02	chuffed	first_fail	indomain_min	3.000e-03		93	84	3	
test_03	chuffed	first_fail	indomain_min	3.000e-03		147	145	6	
test_01	gecode	dom_w_deg	indomain_split	1.634e-03		355	178	8	
test_02	gecode	dom_w_deg	indomain_split	4.356e-03		1019	506	12	
test_03	gecode	dom_w_deg	indomain_split	1.332e-03		237	98	11	
test_01	gecode	first_fail	indomain_min	7.061e-02		30981	15491	6	
test_02	gecode	first_fail	indomain_min	1.864e-03		1019	506	7	
test_03	gecode	first_fail	indomain_min	1.596e-03		395	177	7	

Tabla 4: Resultados de pruebas **sin** restricciones de simetría.

Archivo	Solver	Var	heur	Val	heur	time	nodes	fail	depth
test_01	chuffed	dom_w_deg	indomain_split	2.400e-02		1055	273	13	
test_02	chuffed	dom_w_deg	indomain_split	3.000e-03		112	105	8	
test_03	chuffed	dom_w_deg	indomain_split	8.000e-03		645	526	9	
test_01	chuffed	first_fail	indomain_min	1.800e-02		180	180	3	
test_02	chuffed	first_fail	indomain_min	3.000e-03		157	157	3	
test_03	chuffed	first_fail	indomain_min	4.000e-03		189	188	7	
test_01	gecode	dom_w_deg	indomain_split	1.438e-03		355	178	8	
test_02	gecode	dom_w_deg	indomain_split	2.777e-03		1103	544	12	
test_03	gecode	dom_w_deg	indomain_split	1.774e-03		263	90	11	
test_01	gecode	first_fail	indomain_min	7.473e-02		31331	15666	6	
test_02	gecode	first_fail	indomain_min	3.032e-03		1103	544	7	
test_03	gecode	first_fail	indomain_min	3.170e-03		429	173	8	

5.4. Árboles de búsqueda

Se capturaron con *Gecode Gist*.

Árboles de búsqueda (Google Drive).

5.5. Análisis y conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur at dui sed justo viverra ultrices. Integer a nisl id enim ornare dictum. Mauris non lectus vel turpis posuere tincidunt. In hac habitasse platea dictumst. Donec et urna non velit tempus vulputate.

Suspendisse potenti. Phasellus lacinia, arcu et gravida pharetra, tortor nisl iaculis augue, eget porta libero sapien in odio. Sed imperdiet, turpis at facilisis varius, arcu velit aliquet justo, vitae convallis lorem ipsum id urna. Cras ut sem vel ex sagittis bibendum.

Praesent euismod, sapien a cursus molestie, risus metus feugiat lorem, vitae gravida enim felis id magna. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

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 y conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur at dui sed justo viverra ultrices. Integer a nisl id enim ornare dictum. Mauris non lectus vel turpis posuere tincidunt. In hac habitasse platea dictumst. Donec et urna non velit tempus vulputate.

Suspendisse potenti. Phasellus lacinia, arcu et gravida pharetra, tortor nisl iaculis augue, eget porta libero sapien in odio. Sed imperdiet, turpis at facilisis varius, arcu velit aliquet justo, vitae convallis lorem ipsum id urna. Cras ut sem vel ex sagittis bibendum.

Praesent euismod, sapien a cursus molestie, risus metus feugiat lorem, vitae gravida enim felis id magna. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Referencias

- [1] John Freddy Belalcazar. Notas sobre modelamiento de csp. *Preprint*, 2024.
- [2] Samuel Galindo Cuevas. Estrategias de búsqueda en minimización. *Preprint*, 2024.
- [3] Nicolas Herrera Marulanda. Restricciones redundantes y rompimiento de simetrías. *Preprint*, 2024.