

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

5 de octubre de 2025

## Índice

<b>1. Sudoku</b>	<b>2</b>
1.1. Modelo . . . . .	2
1.2. Implementación . . . . .	3
1.3. Pruebas . . . . .	4
1.4. Árboles de búsqueda . . . . .	4
1.5. Análisis y conclusiones . . . . .	5
<b>2. Kakuro</b>	<b>5</b>
2.1. Modelo . . . . .	5
2.2. Detalles de implementación . . . . .	5
2.3. Pruebas . . . . .	5
2.4. Árboles de búsqueda . . . . .	5
2.5. Análisis . . . . .	5
<b>3. Secuencia Mágica</b>	<b>6</b>
3.1. Modelo . . . . .	6
3.2. Detalles de implementación . . . . .	6
3.3. Pruebas . . . . .	6
3.4. Árboles de búsqueda . . . . .	6
3.5. Análisis y conclusiones . . . . .	6
<b>4. Acertijo Lógico</b>	<b>7</b>
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 . . . . .	7
<b>5. Ubicación de personas en una reunión</b>	<b>7</b>
5.1. Modelo . . . . .	8
5.2. Detalles de implementación . . . . .	10
5.3. Pruebas . . . . .	12
5.4. Árboles de búsqueda . . . . .	12
5.5. Análisis y conclusiones . . . . .	12
<b>6. Construcción de un rectángulo</b>	<b>13</b>
6.1. Modelo . . . . .	13
6.2. Detalles de implementación . . . . .	13
6.3. Pruebas . . . . .	13
6.4. Árboles de búsqueda . . . . .	13
6.5. Análisis y conclusiones . . . . .	13

## 1. Sudoku

Puzzle en una grilla  $9 \times 9$  dividida en nueve cajas  $3 \times 3$ . 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  $3 \times 3$  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:**  $\forall c \in S : \text{all\_different}([X_{r,c} \mid r \in S])$ .

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

### 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 (toda solución coincide con  $G$  y difiere sólo en  $\mathcal{B}$ ).

### Restricciones redundantes

Usamos las sumas = 45 (véase R5–R7 en §1.1) como poda adicional: no alteran el conjunto de soluciones y aceleran la detección de inconsistencias locales (p. ej., parciales que superan 45 o no alcanzables con los dominios), reduciendo *nodos/fallos*.

### Ruptura de simetría

No imponemos rompedores globales: con pistas, las simetrías clásicas (permutar filas/columnas/bandas, renombrar dígitos, transponer) ya no preservan la instancia porque moverían  $G$ . Forzar simetrías podría eliminar la única solución compatible.

### 1.3. Pruebas

**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		4.000e-03	5	3	2
test_01	Chuffed	dom_w_deg		indomain_split		1.000e-03	7	4	3
test_01	Chuffed	—		—		1.000e-03	8	6	2
test_01	Gecode	first_fail		indomain_min		8.044e-03	61	28	7
test_01	Gecode	dom_w_deg		indomain_split		2.857e-03	43	20	8
test_01	Gecode	—		—		1.022e-03	32	14	7
test_02	Chuffed	first_fail		indomain_min		1.200e-02	418	367	13
test_02	Chuffed	dom_w_deg		indomain_split		1.300e-02	550	496	14
test_02	Chuffed	—		—		5.000e-03	170	157	8
test_02	Gecode	first_fail		indomain_min		1.733e-02	1964	979	17
test_02	Gecode	dom_w_deg		indomain_split		6.567e-03	703	349	21
test_02	Gecode	—		—		6.834e-04	6	1	4
test_03	Chuffed	first_fail		indomain_min		2.000e-03	28	22	6
test_03	Chuffed	dom_w_deg		indomain_split		7.000e-03	337	324	9
test_03	Chuffed	—		—		4.000e-03	172	160	7
test_03	Gecode	first_fail		indomain_min		1.157e-02	727	360	11
test_03	Gecode	dom_w_deg		indomain_split		4.690e-03	218	107	11
test_03	Gecode	—		—		3.631e-03	305	149	11

**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		2.000e-03	5	3	2
test_01	Chuffed	dom_w_deg		indomain_split		1.000e-03	7	4	3
test_01	Chuffed	—		—		1.000e-03	8	6	2
test_01	Gecode	first_fail		indomain_min		8.472e-03	61	28	7
test_01	Gecode	dom_w_deg		indomain_split		2.133e-03	41	19	7
test_01	Gecode	—		—		7.576e-04	32	14	6
test_02	Chuffed	first_fail		indomain_min		9.000e-03	417	369	13
test_02	Chuffed	dom_w_deg		indomain_split		9.000e-03	498	459	14
test_02	Chuffed	—		—		3.000e-03	147	131	8
test_02	Gecode	first_fail		indomain_min		1.318e-02	1964	979	17
test_02	Gecode	dom_w_deg		indomain_split		3.817e-03	457	226	18
test_02	Gecode	—		—		5.463e-04	6	1	4
test_03	Chuffed	first_fail		indomain_min		2.000e-03	28	22	6
test_03	Chuffed	dom_w_deg		indomain_split		7.000e-03	338	324	9
test_03	Chuffed	—		—		4.000e-03	210	198	6
test_03	Gecode	first_fail		indomain_min		6.176e-03	727	360	11
test_03	Gecode	dom_w_deg		indomain_split		2.267e-03	151	72	11
test_03	Gecode	—		—		2.946e-03	297	145	12

### 1.4. Árboles de búsqueda

Los árboles de búsqueda se encuentran disponibles en la siguiente carpeta compartida: **Árboles de búsqueda** ([Google Drive](#)).

### **1.5. 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.

## **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.

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

### 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.

$$\text{personas} \in \text{String}^S.$$

**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:

$$\text{NEXT} \in S^{K_{\text{next}} \times 2}, \quad \text{SEP} \in S^{K_{\text{sep}} \times 2}, \quad \text{DIST} \in (S \times S \times \{0, \dots, N-2\})^{K_{\text{dist}}}.$$

Cada fila codifica un par (o trío) de personas y, en **DIST**, una cota  $M$  de separación.

### Variables

**V1** —  $POS\_OF_p$ : Posición que ocupa la persona  $p$ .

$$POS\_OF_p \in POS, \quad p \in S.$$

**V2** —  $PER\_AT_i$ : Persona ubicada en la posición  $i$ .

$$PER\_AT_i \in S, \quad i \in POS.$$



### 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.

$$\forall(A, B) \in \text{NEXT} : \quad |POS\_OF_A - POS\_OF_B| = 1.$$

**R3 — Preferencias separate( $A, B$ ):**  $A$  y  $B$  no pueden quedar adyacentes.

$$\forall(A, B) \in \text{SEP} : \quad |POS\_OF_A - POS\_OF_B| \geq 2.$$

**R4 — Preferencias distance( $A, B, M$ ):** A lo sumo  $M$  personas entre  $A$  y  $B$ , equivalente a cota sobre distancia de posiciones.

$$\forall(A, B, M) \in \text{DIST} : \quad |POS\_OF_A - POS\_OF_B| \leq M + 1.$$

### Restricciones redundantes

**R5 — Refuerzo de permutación:** Duplican la biección y aceleran la propagación.

$$all\_different([POS\_OF_p \mid p \in S]), \quad all\_different([PER\_AT_i \mid i \in POS]),$$

$$\sum_{p \in S} POS\_OF_p = \frac{N(N+1)}{2}.$$

**R6 — Validación de datos:** Índices válidos y pares distintos; en DIST, cota  $M$  dentro de  $[0, N-2]$ .

$$\forall(A, B) \in \text{NEXT} : \quad A, B \in S, \quad A \neq B,$$

$$\forall(A, B) \in \text{SEP} : \quad A, B \in S, \quad A \neq B,$$

$$\forall(A, B, M) \in \text{DIST} : \quad A, B \in S, \quad A \neq B, \quad 0 \leq M \leq N - 2.$$

**R7 — No contradicción next vs sep:** Se prohíbe declarar simultáneamente que dos personas deban y no deban estar juntas.

$$\forall(A, B) \in \text{NEXT}, \quad \forall(C, D) \in \text{SEP} : \quad (A, B) \notin \{(C, D), (D, C)\}.$$

**R8 — Implicación local:** distance( $A, B, 0$ ) es equivalente a next( $A, B$ ).

$$\forall(A, B, 0) \in \text{DIST} : \quad |POS\_OF_A - POS\_OF_B| = 1.$$

## Restricciones de simetrías

**R9 — Rompimiento de simetría izquierda–derecha:** La solución reflejada (espejada) es equivalente; para evitar exploración duplicada, se fija un orden canónico en los extremos.

$$PER\_AT_1 < PER\_AT_N.$$

*Justificación:* si  $[p_1, \dots, p_N]$  es solución, entonces  $[p_N, \dots, p_1]$  también suele serlo cuando las preferencias dependen sólo de distancias absolutas. Forzar  $PER\_AT_1 < PER\_AT_N$  descarta exactamente una de las dos, reduciendo el espacio de búsqueda sin eliminar soluciones no simétricas.

## 5.2. Detalles de implementación

### Archivos y organización

- `reunion.mzn`: modelo completo.
- `tests/*.dzn`: instancias con:
  - `N`, `personas` (vector de nombres),
  - `K_NEXT`, `K_SEP`, `K_DIST`,
  - $NEXT \in \mathbb{Z}^{K_{\text{next}} \times 2}$ ,  $SEP \in \mathbb{Z}^{K_{\text{sep}} \times 2}$ ,
  - $DIST \in \mathbb{Z}^{K_{\text{dist}} \times 3}$  con triples  $(A, B, M)$ .

### Modelo

#### Modelo

Usamos dos vistas de la permutación,  $POS\_OF$  (persona→posición) y  $PER\_AT$  (posición→persona), enlazadas con *inverse*, porque refuerzan la propagación sin cambiar la semántica. Con una sola vista (p. ej.,  $POS\_OF$ ) ya podríamos expresar  $|POS\_OF_A - POS\_OF_B|$ , pero la canalización bidireccional permite a los propagadores recortar dominios antes y con menos fallos que usar solo *all\_different*. Además, la salida es más directa y limpia: imprimir la fila en orden  $1..N$  se reduce a recorrer  $PER\_AT[i]$  sin accesos indirectos ni restricciones *element*. También simplifica la ruptura de simetría  $PER\_AT[1] < PER\_AT[N]$ . En suma, la doble vista mejora eficiencia, claridad de salida y facilidad de formulación, manteniendo dominios exactos  $POS = S = \{1, \dots, N\}$  sin sentinelas.

### Restricciones redundantes

Se activan de manera permanente porque fortalecen la propagación sin alterar el conjunto de soluciones. En primer lugar, se duplican las

vistas de permutación mediante *all\_different* tanto sobre *POS\_OF* como sobre *PER\_AT*; aunque la biyección ya está garantizada por *inverse*(*POS\_OF*, *PER\_AT*), imponer *all\_different*( $[POS\_OF_p \mid p \in S]$ ) y *all\_different*( $[PER\_AT_i \mid i \in POS]$ ) añade redundancia estructural que los propagadores explotan para reducir dominios antes y con menos fallos. En segundo lugar, la igualdad

$$\sum_{p \in S} POS\_OF_p = \frac{N(N+1)}{2}$$

no introduce información nueva sobre las soluciones, pero actúa como restricción global lineal útil cuando quedan pocas posiciones libres, cerrando brechas y detectando inconsistencias parciales con bajo costo. En tercer lugar, se materializa la implicación local  $\text{distance}(A, B, 0) \Rightarrow |POS\_OF_A - POS\_OF_B| = 1$ ; esta equivalencia ya está contenida semánticamente en *distance*, pero declararla explícitamente evita que el solver deba inferirla por combinación de propagadores, acelerando los casos triviales de “cero personas entre *A* y *B*”.

Además, integramos como redundantes los chequeos de consistencia de datos, pues acotan tempranamente la búsqueda sin modificar el conjunto de soluciones factibles cuando la instancia es válida. Exigir  $A, B \in S$  y  $A \neq B$  en *NEXT* y *SEP*, así como  $A, B \in S$ ,  $A \neq B$  y  $0 \leq M \leq N - 2$  en *DIST*, no cambia la semántica del problema, pero evita que dominios inválidos o cotas imposibles entren al árbol de búsqueda. De igual modo, prohibir la contradicción obvia entre *next* y *separate* para el mismo par (*A*, *B*) (en cualquier orden) no elimina soluciones legítimas: únicamente descarta instancias mal especificadas o ramas inconsistentes que, de otro modo, el solver podaría más tarde a mayor costo.

### Ruptura de simetría

El modelo presenta simetría de reflexión izquierda–derecha: si  $[p_1, \dots, p_N]$  es solución,  $[p_N, \dots, p_1]$  también lo es, pues todas las restricciones dependen de distancias absolutas entre posiciones ( $|POS\_OF_A - POS\_OF_B|$ ) y de la biyección *inverse*, que son invariantes ante la transformación  $i \mapsto N+1-i$ . Por ello conviene imponer  $PER\_AT[1] < PER\_AT[N]$ , que fija un representante canónico por cada par de soluciones espejo sin eliminar soluciones no simétricas: dado que  $PER\_AT[1] \neq PER\_AT[N]$ , exactamente una de las dos disposiciones reflejadas satisface  $<$ , mientras la otra la viola. Esta ruptura reduce a la mitad, en términos ideales, el espacio de búsqueda explorado por el solver sin afectar satisfacibilidad ni, si se añadiera un objetivo, la optimalidad. La condición es segura mientras no existan reglas que distingan explícitamente los extremos, caso en el cual la simetría ya no está presente.

## Estrategias de búsqueda

Para las pruebas sobre el modelo de Reunion 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.

## Solvers

Se utilizarán los *solvers* XXX, YYY y ZZZ para contrastar el comportamiento del modelo bajo motores de propagación diferentes. El objetivo es observar variaciones en tiempo y tamaño del árbol manteniendo la misma formulación.

### 5.3. Pruebas

Se evaluó el modelo sobre una batería de instancias *.dzn*. En cada corrida se registraron *tiempo*, *nodos*, *fallos*, *profundidad* y *número de soluciones*. A continuación se presenta una tabla plantilla para consolidar dichos resultados.

**Tabla 3:** 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

### 5.4. Árboles de búsqueda

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

### 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 minizinc. *Preprint*, 2024.
- [3] Nicolas Herrera Marulanda. Restricciones redundantes y rompimiento de simetrías. *Preprint*, 2024.