

# Proyecto

Emily Esmeralda Carvajal Camelo 1630436  
Jaime Cuartas Granada - 1632664

Programación con restricciones  
UNIVERSIDAD DEL VALLE

## Parte 1 - El problema de la planificación de los ensayos en una telenovela

En este trabajo se trata de resolver el problema de los ensayos en una telenovela. Es bien sabido que las escenas de las telenovelas no necesariamente se ensayan en el orden en que se van a emitir (en algunas telenovelas el orden de las escenas se podría variar en cualquier manera sin que la trama sufriera, porque no la hay). No todos los actores participan en todas las escenas. Como a los actores les pagan por el tiempo total en que deben estar en el set, es muy bueno no dejarlos esperando después de una escena en que participan durante varias en que no lo hacen, hasta que llegue otra en que vuelven a participar.

Para simplificar, supondremos que se conoce de antemano el número de escenas que componen la telenovela (lo que en Colombia es falso porque la duración de una telenovela puede ser infinita). El cuadro siguiente muestra un ejemplo de la participación en escenas para la telenovela “Desenfreno de Pasiones”. Para efectos prácticos utilizaremos “Desenfreno de Pasiones” para explicar el trabajo.

Escena	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Costo
Actor 1	1	1	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	10
Actor 2	1	1	1	0	0	0	1	1	0	1	0	0	1	1	1	0	1	0	0	1	4
Actor 3	0	1	1	0	1	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	5
Actor 4	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	5
Actor 5	0	1	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	1	1	5
Actor 6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	40
Actor 7	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	4
Actor 8	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	20
Durac.	2	1	1	1	1	3	1	1	1	2	1	1	2	1	2	1	1	2	1	1	

Cuadro 1: Escenas de “Desenfreno de pasiones”.

Para cada actor, un uno en la columna de la escena  $i$  indica que ese actor debe ensayar esa escena. Por ejemplo, el actor 6 (Narciso Perfiles, venezolano) participa en las escenas de la 14 a la 18. La última columna indica lo que cobra

cada actor (en cientos de miles) por cada unidad de tiempo que le toque estar en el estudio. La última fila establece la duración de cada escena. Por ejemplo, la escena 6 dura 3 unidades de tiempo.

El problema consiste en encontrar el orden en que se deben ensayar (o filmar) las escenas de modo que se minimice el costo total que se debe pagar a los actores. Hay que tener en cuenta que un actor debe estar presente al comienzo de la primera escena que le toca y solamente puede salir después de la última que le toca. Por ejemplo, si la secuencia de escenas se programara en el orden en que están en la tabla, el actor 3 debería estar en el estudio desde la escena 2 hasta la escena 15, o sea durante  $1+1+1+1+3+1+1+1+2+1+1+2+1+2=19$ , y cobraría  $19 \times 5 = 95$ , o sea \$9,500,000 pesos.

### Solución

#### Parámetros:

- $m$ : Indica el número de actores que tiene la telenovela.
- $n$ : Indica el número de escenas que tiene la telenovela.
- $Duracion = (Duracion_1, Duracion_2, \dots, Duracion_n)$   
Donde  $Duracion_i$  representa la duración de la  $i$ -ésima escena, tal que  $i \in \{1, 2, \dots, n\}$ .
- $Costo = (Costo_1, Costo_2, \dots, Costo_n)$   
Donde  $Costo_i$  representa el costo de la  $i$ -ésima escena, tal que  $i \in \{1, 2, \dots, n\}$ .
- $EscenaDe = (EscenaDe_1, EscenaDe_2, \dots, EscenaDe_m)$   
Donde  $EscenaDe_i$  representa el conjunto de escenas en las que participa el  $i$ -ésimo actor, tal que  $i \in \{1, 2, \dots, m\}$ .

#### Ejemplo de la instancia de parámetros para el caso de “Desenfreno de Pasiones”:

- $m = 8$ .
- $n = 20$ .
- $Duracion = (2, 1, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1)$ .
- $Costo = (10, 4, 5, 5, 5, 40, 4, 20)$ .
- $EscenaDe = (\{1, 2, 3, 4, 6, 8, 10, 11\},$   
 $\{1, 2, 3, 7, 8, 10, 13, 14, 15, 17, 20\},$   
 $\{2, 3, 5, 7, 8, 13, 14, 15\},$   
 $\{10, 11, 12, 13\},$   
 $\{2, 7, 8, 12, 14, 18, 19, 20\},$   
 $\{14, 15, 16, 17, 18\},$   
 $\{5, 7, 8, 15\},$   
 $\{6, 7, 8, 9\})$

**Dominios de los parámetros:**

- $m \in \mathbb{N}$ .
- $n \in \mathbb{N}$ .
- $Duracion_i \in \mathbb{N}$ .
- $Costo_i \in \mathbb{N}$ .
- $EscenaDe_i \in \mathcal{P}(A)$ , donde  $A = \{1, 2, \dots, n\}$  y  $\mathcal{P}(A)$  denota el conjunto potencia de  $A$ .

**Variables:**

- $PosicionDeEscena = (PosicionDeEscena_1, PosicionDeEscena_2, \dots, PosicionDeEscena_n)$   
Es la tupla de salida donde  $PosicionDeEscena_i$  representa la posición donde debe ubicarse la escena  $i$ , tal que  $i \in \{1, 2, \dots, n\}$ .
- $costo$  Es la variable que almacena el costo de realizar la novela dada la configuración de  $PosicionDeEscena$ .

**Dominios de las variables:**

- $PosicionDeEscena_i \in \{1, 2, \dots, n\}$ .
- $costo \in \mathbb{N}$

**Restricciones:**

Sea el conjunto  $C = \{1, 2, \dots, n\}$

- $\forall i, j \in C \quad (PosicionDeEscena_i \neq PosicionDeEscena_j \iff i \neq j)$

Cada elemento de la tupla  $PosicionesDeEscena$  debe ser diferente a otro elemento de la misma tupla, siempre que se trate de elementos diferentes. Esto debido a que dos escenas  $i$  y  $j$  diferentes no pueden tener la misma posición de grabación.

**Función objetivo:**

Sea  $A \subseteq \mathbb{N}$ , un conjunto finito y distinto del vacío.

- $\min(A)$  es el mínimo de  $A$ , es decir, un número  $k \in A$  menor o igual que cualquier número de  $A$ .
- $\max(A)$  es el máximo de  $A$ , es decir, un número  $k \in A$  mayor o igual que cualquier número de  $A$ .

Sea  $b \in B = \{1, 2, \dots, n\}$ ,  $c \in C = \mathcal{P}(B)$  y  $d \in D = \{0, 1\}$ .

- $f : B \times C \longrightarrow D$   
 $(b, c) \mapsto f(b, c) = d$

Donde  $f$  es una función que dada posición y un conjunto de posiciones de las escenas en las que participa un actor determinado, retorna 1 si  $b$  cumple estar  $\min(C) \leq b \leq \max(C)$  y 0 en otro caso.

**Minimizar:**

$$\sum_{j=0}^m \left( Costo_j \cdot \sum_{i=0}^n Duracion_i \cdot f( PosicionDeEscena_i, \{ PosicionDeEscena_k | k \in EscenaDe_j \} ) \right)$$

$$f(x, A) = \begin{cases} 1 & \text{si } min(A) \leq x \leq max(A) \\ 0 & \text{si } min(A) > x \vee x > max(A) \end{cases}$$

Se suman los costos de cada actor. Para ello se calcula si cada escena tiene una posición entre la primer escena y la ultima de cada actor, si se encuentra en medio entonces se suma la duración de la escena y al verificar todas las escenas se multiplica el tiempo que está el actor en el estudio con el valor que cobra. Se realiza el mismo proceso para todos los actores, sumándolos, y de esta manera se calcula el costo total de la participación de los actores en la telenovela, cuyo objetivo es que sea mínimo.

## Pruebas de parte 1

Para la ejecución de pruebas se realizó una implementación del modelo descrito anteriormente en Minizinc. El computador utilizado tiene como características un procesador Intel ®Core™i5-2430M CPU @ 2.40GHz 4 y 5,7 GiB.

Las instancias probadas son las siguientes:

– **Trivial1.dzn**

ACTORES = {Actor1, Actor2, Actor3} ;

Escenas = [ | 0,1,0,0,0,0,10  
| 0,1,1,1,0,1,20  
| 0,0,0,0,1,1,15 | ];

Duracion = [2,1,1,1,3,4];

– **Med1-1.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4} ;

Escenas = [ | 0,1,0,0,0,0,10  
| 0,1,1,1,0,1,20  
| 1,1,1,1,0,1,5  
| 1,0,0,0,1,1,15 | ];

Duracion = [2,1,1,1,3,4];

– **Med1-2.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4} ;

Escenas = [ | 0,1,0,0,0,0,1,10  
| 0,1,1,1,0,1,1,20

| 1,1,1,1,0,1,0,5  
 | 1,0,0,0,1,1,0,15 | ];  
 Duracion = [2,1,1,1,3,4,2];

– **Med1-3.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5} ;  
 Escenas = [ | 0,1,0,0,0,0,1,10  
 | 0,1,1,1,0,1,1,20  
 | 1,1,1,1,0,1,0,5  
 | 0,1,1,1,0,0,1,5  
 | 1,0,0,0,1,1,0,15 | ];  
 Duracion = [2,1,1,1,3,4,2];

– **Med1-4.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5} ;  
 Escenas = [ | 0,1,0,0,0,0,1,0,10  
 | 0,1,1,1,0,1,1,1,20  
 | 1,1,1,1,0,1,0,0,5  
 | 0,1,1,1,0,0,1,1,5  
 | 1,0,0,0,1,1,0,0,15 | ];  
 Duracion = [2,1,1,1,3,4,2,3];

– **Propia1-1.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6, Actor7} ;  
 Escenas = [ | 0,1,0,0,0,0,1,0,10  
 | 0,1,1,1,0,1,1,1,20  
 | 1,1,1,1,0,1,0,0,5  
 | 0,1,1,1,0,0,1,1,5  
 | 1,0,0,0,1,1,0,0,15  
 | 1,0,0,0,1,0,1,1,10  
 | 1,1,0,1,0,1,1,0,10 | ];  
 Duracion = [2,1,1,1,3,4,2,3];

– **Propia1-2**

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5} ;  
 Escenas = [ | 0,1,0,0,0,0,1,0,1,1,10  
 | 0,1,1,1,0,1,1,1,0,1,20

| 1,1,1,1,0,1,0,0,1,1,5  
 | 0,1,1,1,0,0,1,1,0,1,5  
 | 1,0,0,0,1,1,0,0,1,0,15 | ];  
 Duracion = [2,1,1,1,3,4,2,3,2,3];

Archivo	Características		Solver (tiempos en msec)			Costo
	Nº de actores	Nº de escenas	Chuffed	COIN-BC	Gecode	
Trivial1.dzn	3	6	162	486	165	255
Med1-1.dzn	4	6	171	878	164	330
Med1-2.dzn	4	7	172	1.977	175	390
Med1-3.dzn	5	7	215	5.107	173	415
Med1-4.dzn	5	8	192	7.200	324	490
Propia1-1.dzn	7	8	185	11.181	433	770
Propia1-2.dzn	5	10	208	8.600	6.189	720

Cuadro 2: Pruebas parte 1 con diferentes Solvers

De la tabla anterior podemos comparar el tiempo de ejecución de diferentes problemas utilizando 3 solvers implementando el modelo descrito anteriormente en Minizinc. Entre ellos se evidencia un mejor rendimiento por parte de Chuffed.

De los resultados podemos ver un incremento en el tiempo de respuesta notorio tanto en COIN-BC como Gecode. Por un lado el incremento de los actores sobre el número de escenas parece tener un gran impacto en los tiempos de respuesta para COIN-BC; por otro lado para Gecode el incremento en el número de escenas tuvo un impacto realmente notorio comparado con tamaños de escenas menores.

A pesar de que tanto el número de actores como el número de escenas tienen una incidencia en los tiempos de respuesta, para problemas de tamaño mayor, podemos pensar que el número de escenas juega un papel más relevante en el tiempo de respuesta debido a que la cantidad de escenas implica una mayor cantidad de posibilidades a la hora de ordenar la mejor distribución de escenas.

## Estrategias de distribución:

Para el análisis de las diferentes estrategias de distribución se utilizó el solver Gecode Gist debido a que ofrece la herramienta de visualización del árbol de búsqueda apropiado para ver las diferencias entre una distribución y otra. Se utilizó como archivo de datos Propia1-2.dzn debido a que cuenta con más escenas que los anteriores.

Selección de variable Restricción sobre variable	input_order	first_fail	smallest	dom_w_deg
indomain_min	1.105.880	1.105.887	666.292	363.961
indomain_median	1.382.131	1.381.978	1.382.151	414.178
indomain_random	1.211.462	1.168.892	1.057.033	362.354
indomain_split	1.074.495	1.074.529	818.504	278.038

Cuadro 3: Número de nodos explorados con diferentes distribuciones para Propia1-2.dzn.

Buscador por defecto de Gecode Gist
283.528

Cuadro 4: Número de nodos explorados con el buscador por defecto de Gecode Gist para Propia1-2.dzn.

Teniendo en cuenta los dos cuadros anteriores que describen a grandes rasgos el comportamiento de las estrategias de distribución utilizadas en este proyecto para llevar a cabo su comparación encontramos dos estrategias que superan notablemente al resto, ellas son la búsqueda por defecto de Gecode Gist y el uso de dom\_w\_deg con indomain\_split para la selección de la variable y la restricción añadida para continuar las búsquedas respectivamente. La selección de la variable en función de aquellas con dominio más pequeño y que hayan provocado pocas fallas favorece a la solución del problema debido a que el conocimiento de la posición de una escena favorece la poda de los dominios de las posiciones de las otras escenas. La posición de una escena es más probable de ser acertada en los casos en los que los dominios de las variables sean pequeños y si no se ha encontrado muchas fallas es posible que sea fácil de asignar, y no que se pueda localizar en muchos lugares. Respecto a la restricción sobre la variable parte de indomain\_split puede deberse a que todas las escenas deben ser ubicadas, y tener una estrategia que tienda a la elección de los valores pequeños favorece debido a que es seguro que algunas escenas deben ocupar las primeras posiciones.

En el caso del buscador por defecto de Gecode Gist es probable que se trate de Branch and Bounds o de una estrategia similar que favorece la solución de problemas de optimización con el uso de cotas que podan ramas del árbol cuyo mejor caso no superará el ya encontrado.

## “Desenfreno de pasiones”

A manera de ejercicio para el modelo descrito se ejecutó el problema de “Desenfreno de pasiones” usando el solver Chuffed que dio los mejores resultados para casos pequeños y con la búsqueda por defecto del mismo solver. A continuación se presenta el archivo con la solución:

– Desenfreno1.dzn

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6, Actor7, Actor8} ;

Escenas = [ | 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 10  
| 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 4  
| 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 5  
| 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 5  
| 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 5  
| 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 40  
| 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 4  
| 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20 | ];

Duracion = [2 , 1 , 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1];

Con el modelo implementado en un computador con procesador Intel Xeon(R) W-2123 CPU @ 3.60GHz x 8 y 15,3 GiB se obtuvieron los resultados.

[4 1 11 10 13 3 12 2 6 8 7 9 20 5 15 14 17 16 18 19] Costo=871 Finished in 3h 38m 35s
---

## Parte 2 - El problema de la planificación de los ensayos en una telenovela

### Restricciones Adicionales

Eliminación de simetrías:

1. Este problema tiene simetrías, que pueden eliminarse para que la solución sea más eficiente. Diga qué restricciones adicionales para eliminar simetrías y sustente su escogencia.
2. Incluir estas restricciones en su programa y correrlo, comparándolo con el original.

Adicionalmente se deben considerar las siguientes restricciones:

1. Disponibilidad de los actores: en general todos los actores deben estar disponibles todo el tiempo, pero hay ciertos actores que por ser estrellas pueden imponer restricciones como por ejemplo un máximo número de unidades de tiempo en el estudio. Por ejemplo Narciso Perfiles podría decir 10, y el actor 3 podría decir 15.
2. Puede haber actores que no la van bien con otros actores y sería ideal que el tiempo que estuvieran en el estudio al mismo tiempo fuera el menor posible (pero tenga en cuenta que la prioridad es reducir costo).



Esta parte consiste, entonces, en extender el CSP de la Parte 1 que modele el problema con las restricciones adicionales: Diga cuáles son sus:

- Datos de entrada.
- Variables: Explique qué representa cada una.
- Restricciones: para cada una explique su semántica intuitiva.
- Función objetivo: Explique su significado.

Implementar su modelo utilizando MiniZinc; definir, sustentar e implementar la estrategia de distribución a utilizar; correrlo con diferentes ejemplos, y mostrar la solución (preferiblemente de manera gráfica).

### Solución

#### Parámetros:

Se incluyen los parámetros de la parte 1 y se incluye:

- $Disponibilidad = (Disponibilidad_1, Disponibilidad_2, \dots, Disponibilidad_m)$

Donde  $Disponibilidad_i$  representa una 2-tupla donde el primer elemento es el actor y el segundo el tiempo disponible para hacer la telenovela, tal que  $i \in \{1, 2, \dots, m\}$ . Si el tiempo disponible para la novela es cero el tiempo disponible es ilimitado.

- $Evitar = (Evitar_1, Evitar_2, \dots, Evitar_e)$

Donde  $Evitar_i$  representa un conjunto de dos actores que desean reducir el tiempo que comparten en estudio, tal que  $e \in \mathbb{N}$ .

#### Ejemplo de la instancia de parámetros para el caso de “Desenfreno de Pasiones”:

Se incluyen las instancias de los parametros descritos en la parte 1 y los siguientes:

- $Disponibilidad = ((1, 0)$   
 $(2, 0)$   
 $(3, 23)$   
 $(4, 0)$   
 $(5, 80)$   
 $(6, 10)$   
 $(7, 5)$   
 $(8, 100)).$

- $Evitar = ((1, 3)$   
 $(1, 4)$   
 $(2, 3)$   
 $(2, 4))$

**Dominios de parámetros:**

- $Disponibilidad_i \in \{1, 2, \dots, m\} \times \mathbb{N}$ ,  $i \in \{1, 2, \dots, m\}$
- $Evitar_i \in \{1, 2, \dots, m\} \times \{1, 2, \dots, m\}$ ,  $i \in \{1, 2, \dots, e\}$  donde  $e$  es el número de pares de actores que desean evitarse

**Variables:**

A las variables presentes en el modelo de la parte 1 se adiciona:

- $evitar$  Esta suma la duración de las escenas en que los actores presentes en cada  $Evitar_i$  en el parametro  $Evitar$
- $TiempoDe = (TiempoDe_1, TiempoDe_2, \dots, TiempoDe_m)$  donde  $TiempoDe_i$  representa el total de unidades de tiempo que estará en el set el i-esimo actor dada la configuración de  $PosicionDeEscena$ .

**Dominios de variables:**

A los dominios de variables presentes en el modelo de la parte 1 se adiciona:

- $evitar \in \mathbb{N}$
- $TiempoDe_i \in \mathbb{N}$

**Restricciones:**

A las restricciones presentes en el modelo de la parte 1 se adiciona:

- $\forall Disponibilidad_i \in Disponibilidad$  con  $Disponibilidad_i \neq 0$   
 Sea  $q = Disponibilidad_{i,1}$  representa el actor al que se le limita el tiempo en el set.  
 Sea  $r = Disponibilidad_{i,2}$  representa el tiempo maximo que puede estar en escena el actor  $q$

$$TiempoDe_q \leq r$$

La restricción anterior asegura que la configuración de  $posicionDeEscena$  cumpla con la disponibilidad para el i-esimo actor.

**Función objetivo:**

Tomando como base la función objetivo definida en la parte 1, la cual llamaremos  $CostoTotal$ , se tomará como uno de los factores de la función, debido a que queremos que el  $CostoTotal$  tenga más importancia sobre el  $evitar$  variable definida anteriormente como parte de las restricciones adicionales, vamos a agregar pesos a estos dos factores.

Frente a un ensayo y error se llego a la siguiente asignación de pesos:

**Minimizar:**

$$9 * CostoTotal + evitar$$

## Pruebas de parte 2

Para la ejecución de pruebas se realizó una implementación del modelo descrito anteriormente en Minizinc. El computador utilizado tiene como características un procesador Intel ®Core™i5-2430M CPU @ 2.40GHz 4 y 5,7 GiB.

Las instancias probadas son las siguientes:

– **Trivial2.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4} ;

Escenas = [ | 1,1,0,0,0,0,10  
              | 0,1,1,1,0,1,20  
              | 0,0,0,0,1,1,15  
              | 1,1,0,0,1,1,13 | ];

Duracion = [2,1,1,1,3,4];

Disponibilidad = [ | Actor1, 5  
                      | Actor2, 8  
                      | Actor3, 0  
                      | Actor4, 10 | ];

Evitar = [ | Actor1, Actor2  
           | Actor2, Actor3 | ];

– **Med2-1.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4} ;

Escenas = [ | 0,1,0,0,0,0,10  
              | 0,1,1,1,0,1,20  
              | 1,1,1,1,0,1,5  
              | 1,0,0,0,1,1,15 | ];

Duracion = [2,1,1,1,3,4];

Disponibilidad = [ | Actor1, 5  
                      | Actor2, 8  
                      | Actor3, 0  
                      | Actor4, 10 | ];

Evitar = [ | Actor1, Actor2  
           | Actor2, Actor3 | ];

– **Med2-2.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4} ;

Escenas = [ | 0,1,0,0,0,0,1,10  
                   | 0,1,1,1,0,1,1,20  
                   | 1,1,1,1,0,1,0,5  
                   | 1,0,0,0,1,1,0,15 | ];

Duracion = [2,1,1,1,3,4,2];

Disponibilidad =[ | Actor1, 8  
                           | Actor2, 9  
                           | Actor3, 0  
                           | Actor4, 10 | ];

Evitar =[ | Actor1, Actor2  
                   | Actor2, Actor3 | ];

– **Med2-3.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5} ;

Escenas = [ | 0,1,0,0,0,0,1,0,0,10  
                   | 0,1,1,1,0,1,1,1,1,20  
                   | 1,1,1,1,0,1,0,0,1,5  
                   | 0,1,1,1,0,0,1,1,1,5  
                   | 1,0,0,0,1,1,0,0,0,15 | ];

Duracion = [2,1,1,1,3,4,2,3,1];

Disponibilidad =[ | Actor1, 13  
                           | Actor2, 14  
                           | Actor3, 15  
                           | Actor4, 14  
                           | Actor5, 10 | ];

Evitar =[ | Actor1, Actor2  
                   | Actor2, Actor3  
                   | Actor4, Actor5 | ];

– **Med2-4.dzn**

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5} ;

Escenas = [ | 0,1,0,0,0,0,1,0,0,10  
                   | 0,1,1,1,0,1,1,1,1,20

```

        | 1,1,1,1,0,1,0,0,1,5
        | 0,1,1,1,0,0,1,1,1,5
        | 1,0,0,0,1,1,0,0,0,15
        | 0,0,0,0,1,0,0,1,0,19 | ];
Duracion = [2,1,1,1,3,4,2,3];
Disponibilidad =[ | Actor1, 15
                   | Actor2, 15
                   | Actor3, 16
                   | Actor4, 17
                   | Actor5, 12
                   | Actor6, 0 | ];
Evitar =[ | Actor1, Actor2
          | Actor2, Actor3
          | Actor4, Actor5
          | Actor1, Actor6 | ];
– Med2-5.dzn
ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6} ;
Escenas = [ | 0,1,0,0,0,0,1,0,0,10
              | 0,1,1,1,0,1,1,1,1,20
              | 1,1,1,1,0,1,0,0,1,5
              | 0,1,1,1,0,0,1,1,1,5
              | 1,0,0,0,1,1,0,0,0,15
              | 0,0,0,0,1,0,0,1,0,19
              | 0,1,0,0,1,0,0,0,0,8 | ];
Duracion = [2,1,1,1,3,4,2,3,1];
Disponibilidad =[ | Actor1, 15
                   | Actor2, 15
                   | Actor3, 16
                   | Actor4, 17
                   | Actor5, 12
                   | Actor6, 0
                   | Actor6, 0 | ];
Evitar =[ | Actor1, Actor2

```

| Actor2, Actor3  
| Actor4, Actor5  
| Actor1, Actor6 | ];

Archivo	Características		Solver (tiempos en msec)			Costo	Tiempo Compartido
	Nº de actores	Nº de escenas	Chuffed	COIN-BC	Gecode		
Trivial2.dzn	4	6	174	2.065	178	450	5
Med2-1.dzn	4	6	185	1.403	189	330	8
Med2-2.dzn	4	7	184	3.372	183	390	10
Med2-3.dzn	5	9	205	5.152	1.022	520	11
Med2-4.dzn	6	9	218	30.842	1.685	729	11
Med2-5.dzn	7	9	260	57.508	3.404	841	11

Cuadro 5: Pruebas parte 2 con diferentes Solvers

De la tabla anterior podemos comparar el tiempo ejecución de diferentes problemas utilizando 3 solvers implementando el modelo descrito anteriormente en Minizinc. Enre ellos se evidencia un mejor rendimiento por parte de Chuffed.

De los resultados podemos ver un incremento en el tiempo de respuesta notorio tanto en COIN-BC como Gecode. Vemos que con solo aumentar una escena para COIN-BC el tiempo de ejecución puede duplicarse. De manera similar ocurre con el incremento de actores. Para la parte 2 del proyecto parece que tanto el incremento en el número de actores como el incremento en el número de escenas tiene un impacto grande en los tiempos de ejecución (tanto para COIN-BC, como para Gecode), esto podría deberse a que para esta situación se realizan permutaciones tanto en las escenas como en el arreglo de disponibilidad para garantizar la satisfactibilidad de la solución, siendo evidente la participación de ambas variables de manera importante.

## Estrategias de distribución:

Para el analisis de las diferentes estrategias de distribución se utilizó el solver Gecode Gist debido a que ofrece la herramienta de visualización del arbol de búsqueda apropiado para ver las diferencias entre una distribución y otra. Se utilizó como archivo de datos Med2-5.dzn debido a que cuenta con más actores que los anteriores y el máximo de escenas.

Restricción sobre variable	Selección de variable	input_order	first_fail	smallest	dom_w_deg
	indomain_min	85231	85017	94205	71698
	indomain_median	97897	95776	97901	90349
	indomain_random	90097	89033	86482	76909
	indomain_split	82909	82749	97319	50273

Cuadro 6: Número de nodos explorados con diferentes distribuciones para Med2-5.dzn.

Buscador por defecto de Gecode Gist
62632

Cuadro 7: Número de nodos explorados con el buscador por defecto de Gecode Gist para Med2-5.dzn.

Los resultados de los dos cuadros anteriores describen a grandes rasgos el comportamiento de las estrategias de distribución tenidas en cuenta para la segunda parte del proyecto. En el comportamiento del número de nodos necesarios para encontrar la solución con las estrategias presentadas, podemos evidenciar nuevamente comportamientos similares a la parte 1. Donde se considera el buscador por defecto de Gecode Gist y el uso de `dom_w_deg` con `indomain_split` como las dos estrategias notablemente mejores. Debido a que el problema continúa teniendo la misma variable de decisión con consideramos que las estrategias favorecen la solución del problema debido a las mismas razones descritas en la parte 1.

## “Desenfreno de pasiones”

A manera de ejercicio para el modelo descrito se ejecutó el problema de “Desenfreno de pasiones 2” usando el solver Chuffed que dio los mejores resultados para casos pequeños y con la búsqueda por defecto del mismo solver. A continuación se presenta el archivo con la solución:

### – Desenfreno2.dzn

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6, Actor7, Actor8} ;

Escenas = [ | 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 10  
| 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 4  
| 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 5  
| 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 5  
| 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 5  
| 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 40  
| 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 4  
| 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20 | ];

Duracion = [2 , 1 , 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 1];

Disponibilidad =[ | Actor1, 0  
| Actor2, 0  
| Actor3, 23  
| Actor4, 0  
| Actor5, 80  
| Actor6, 10

```

| Actor7, 5
| Actor8, 100 | ];
Evitar =[ | Actor1, Actor3
| Actor1, Actor4
| Actor2, Actor3
| Actor2, Actor4 | ];

```

Con el modelo implementado en un computador con procesador Intel Xeon(R) W-2123 CPU @ 3.60GHz x 8 y 15,3 GiB se obtuvieron los resultados.

[19 20 16 18 17 14 15 5 7 8 9 6 2 12 3 13 10 11 1 4] Tiempo Compartido: 33 Costo: 880 Finished in 8s 546msec
---

El enorme incremento en la solución del problema debe corresponder a un incremento de restricciones que favorecieron a tener un número más pequeño de posibilidades en el orden en el que se va a filmar la telenovela.