

# **Entrega 2: Solver, Optimización de Turnos y Análisis de Resultados**

INF292 – Optimización  
Campus San Joaquín - Grupo 2

## **Integrantes:**

Fernanda López Saldías	202373638-4
Matías Romo Vargas	202373632-5
Simón Parra Saldías	202373560-4
Vicente Jiménez Sepúlveda	202373523-k

Noviembre 2025

# 1. Modelo Matemático

## Conjuntos, Parámetros y Variables

### Conjuntos e índices:

$T$ : Trabajadores, indexados por  $i \in T$ .

$D$ : Días del horizonte de planificación,  $D = \{1, 2, \dots, H\}$ , con  $j \in D$ .

$S$ : Turnos disponibles por día, indexados por  $t \in S$ .

$W_k$ : Días del fin de semana de la semana  $k$ , donde  $W_k = \{\text{sábado, domingo}\}$ , y  $K = \lfloor \frac{H}{7} \rfloor$ .

### Parámetros:

$c_{i,j,t}$ : Disposición de  $i$  para el turno  $t$  del día  $j$ , con valores entre 0 y 10.

$r_{j,t}$ : Demanda requerida de trabajadores para el turno  $t$  del día  $j$ .

$a_{i,j,t}$ : Disponibilidad binaria, definida como:

$$a_{i,j,t} = \begin{cases} 1 & \text{si } c_{i,j,t} > 0, \\ 0 & \text{si } c_{i,j,t} = 0. \end{cases}$$

### Variables de decisión:

$x_{i,j,t}$ : Binaria. 1 si el trabajador  $i$  es asignado al turno  $t$  del día  $j$ .

$y_{i,k}$ : Binaria. 1 si el trabajador  $i$  realiza al menos un turno en el fin de semana  $k$ .

## Función Objetivo

$$\max Z = \sum_{i \in T} \sum_{j \in D} \sum_{t \in S} c_{i,j,t} \cdot x_{i,j,t}$$

**Interpretación:** Se maximiza la disposición total del personal asignado, favoreciendo la asignación de turnos a quienes tienen mayor disposición declarada. Esto permite una planificación más eficiente y respetuosa con las preferencias individuales.

## Restricciones

$$\begin{aligned}
 \text{(R1)} \quad & \sum_{i \in T} x_{i,j,t} = r_{j,t} \quad \forall j \in D, t \in S \\
 \text{(R2)} \quad & x_{i,j,t} \leq a_{i,j,t} \quad \forall i, j, t \\
 \text{(R3)} \quad & \sum_{t \in S} x_{i,j,t} \leq 2 \quad \forall i, j \\
 \text{(R4)} \quad & x_{i,j,\text{noche}} + x_{i,j+1,\text{mañana}} \leq 1 \quad \forall i, j < H \\
 \text{(R5)} \quad & y_{i,k} + y_{i,k+1} + y_{i,k+2} \leq 2 \quad \forall i, k \leq K - 2 \\
 \text{(R5.1)} \quad & x_{i,d,t} \leq y_{i,k} \quad \forall d \in W_k, t \in S \\
 & y_{i,k} \leq \sum_{d \in W_k} \sum_{t \in S} x_{i,d,t} \quad \forall i, k \\
 \text{(R6)} \quad & x_{i,j,t}, y_{i,k} \in \{0, 1\} \quad \forall i, j, t, k
 \end{aligned}$$

## Explicación de las Restricciones

**R1:** Garantiza que cada turno sea cubierto exactamente por la cantidad requerida de trabajadores.

**R2:** Asegura que solo se asignen turnos a trabajadores disponibles.

**R3:** Limita la carga diaria de cada trabajador a un máximo de dos turnos.

**R4:** Impide asignaciones consecutivas de noche y mañana para respetar los descansos.

**R5:** Evita que un trabajador trabaje tres fines de semana seguidos, promoviendo el equilibrio.

**R5.1:** Define cuándo se activa la variable  $y_{i,k}$ , vinculándola con la actividad real en fines de semana.

**R6:** Define la naturaleza binaria de las variables de decisión, asegurando que las asignaciones sean discretas.

## Notas explicativas

- **Eliminación de tope de carga:** Se descarta la restricción sobre el total de turnos por trabajador planteada en la entrega 1. La carga se regula mediante R3, R5 y la función objetivo.
- **R5 y R5.1:** R5 limita la frecuencia de trabajo en fines de semana; R5.1 define cuándo se activa esa condición. Juntas aseguran control lógico y preciso.
- **R4 en instancias pequeñas:** Si  $S$  solo incluye día y noche, R4 no aplica. Debe considerarse al validar instancias.

## 2. Generador de Instancias y Factibilidad

El generador fue implementado en **Python** y se encuentra en:

```
generador/src/Generador_1_Grupo2_OPTI_SJ.py
```

Este script genera **cinco instancias por cada tamaño** (*small, medium, large*), siguiendo los rangos establecidos en el enunciado. Las instancias se almacenan en carpetas separadas dentro de:

```
generador/instancias/
```

El código fuente completo está disponible en el repositorio de GitHub [**generador2025**]. Además, se incluye un script de chequeo que valida condiciones de factibilidad y genera resúmenes estadísticos por instancia.

### Lógica de Generación

Se utilizó la semilla 42 para asegurar reproducibilidad. El generador elimina carpetas previas para evitar residuos y garantizar un entorno limpio en cada ejecución.

- Se generan aleatoriamente los días, trabajadores y turnos según el tipo de instancia:
  - *Small*: 5–7 días, 5–15 trabajadores, turnos día y noche.
  - *Medium*: 7–14 días, 15–45 trabajadores, turnos mañana, tarde, noche.
  - *Large*: 14–28 días, 45–90 trabajadores, mismos turnos que *medium*.
- Para cada día y turno, se genera una demanda  $r_{j,t}$  con distribución normal truncada en cero, ajustada por:
  - Disponibilidad efectiva (trabajadores con disposición positiva).
  - Capacidad máxima diaria (`max_turnos_dia` × trabajadores).
- Las disposiciones  $c_{i,j,t} \sim \mathcal{U}\{0, 10\}$  se asignan por trabajador, día y turno.
- Cada instancia se guarda en dos formatos:
  - `.json`: incluye metadatos, demanda y disposiciones.
  - `.csv`: contiene las disposiciones individuales por fila.

## Condiciones de Factibilidad en la Generación

El generador fue diseñado para producir instancias estructuralmente válidas, es decir, con datos completos, consistentes y compatibles con el modelo matemático. Esto implica que cada instancia contiene trabajadores, días, turnos, disposiciones y demandas bien definidas, sin errores lógicos ni violaciones formales.

Además, se incorporan controles que reflejan las restricciones del modelo desde la etapa de generación:

- **R1 (Cobertura):** La demanda por turno se ajusta según la cantidad de trabajadores disponibles con disposición positiva.
- **R3 (Carga diaria):** Se limita la suma de turnos por día, escalando la demanda si excede la capacidad total del equipo.
- **R5 (Fines de semana):** Se evita congestión en sábado y domingo, respetando la carga global del equipo.

Aunque no se incluye una restricción explícita sobre el total de turnos por trabajador (R6), la carga se regula indirectamente mediante R3, R5 y la función objetivo. Las instancias *small* tienden a ser factibles, mientras que en *medium* y *large* la factibilidad depende de la relación entre demanda y disposición.

**Importante:** Que una instancia sea estructuralmente válida no implica que sea resoluble. La factibilidad se evalúa posteriormente mediante el modelo LPE, que determina si existe una asignación que cumpla todas las restricciones. Este enfoque permite estudiar cómo influyen los parámetros generados aleatoriamente en la posibilidad de encontrar soluciones válidas, sin alterar artificialmente las instancias para forzar su resolución.

### 3. Resultados y Análisis

#### Resumen general de la ejecución

El modelo fue resuelto usando la herramienta **LPSolve** a través de un script que procesó las quince instancias generadas (*small*, *medium* y *large*).

El archivo `ejecutar_solver_batch.py` ejecutó el solver para cada instancia y registró los resultados en formato JSON.

En total, se obtuvieron 13 soluciones óptimas y 2 instancias infactibles. Los tiempos de ejecución fueron muy bajos (del orden de milisegundos), confirmando la eficiencia del modelo en instancias pequeñas y medianas.

Tamaño	Total	Óptimas	Infactibles	Tiempo promedio (s)
Small	5	5	0	0.0155
Medium	5	5	0	0.0476
Large	5	3	2	0.1579
<b>Total</b>	<b>15</b>	<b>13</b>	<b>2</b>	—

Cuadro 1: Resumen de resultados globales por tipo de instancia.

De la tabla anterior se puede observar que las instancias *small* y *medium* fueron completamente factibles y alcanzaron el óptimo en menos de 0.05 segundos. En cambio, dos de las instancias *large* resultaron infactibles, lo cual se analizará más adelante.

#### (a) Análisis de la función objetivo

La función objetivo maximiza la disposición total del personal asignado. A medida que el tamaño de la instancia crece (más trabajadores y días), el valor objetivo aumenta de manera proporcional al número de asignaciones posibles, manteniendo así, la coherencia del modelo.

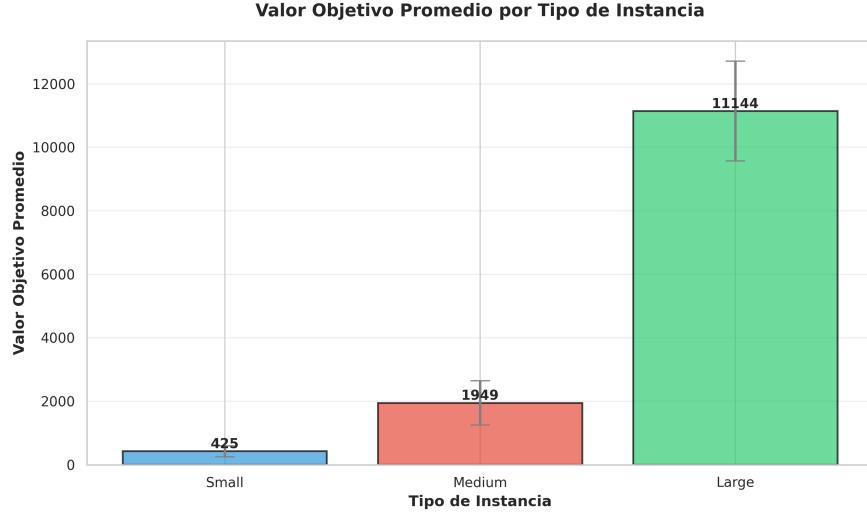


Figura 1: Promedio del valor objetivo por tipo de instancia.

En la Figura anterior se puede apreciar un crecimiento proporcional al tamaño de la instancia del valor objetivo promedio al pasar de *small* a *large*, esto es coherente con el aumento de la demanda y la disponibilidad total de turnos. Lo que demuestra que el modelo logra mantener la consistencia de asignaciones al priorizar trabajadores con mayor disposición.

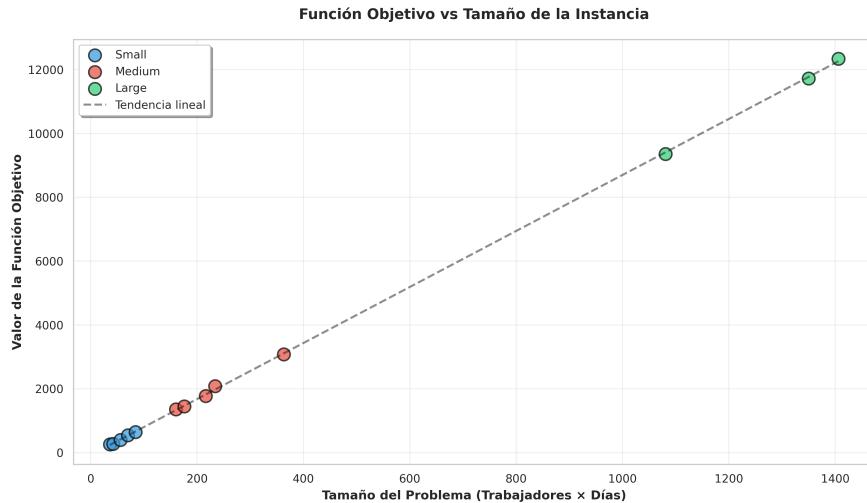


Figura 2: Comportamiento del valor objetivo en función del tamaño (número de variables).

En la Figura 2 se puede observar que la relación entre el valor objetivo y el número de variables muestra una tendencia ascendente con una leve dispersión. Esto refleja que el solver sigue encontrando soluciones óptimas, pero la complejidad combinatoria puede introducir ligeras variaciones en la calidad o el tiempo de resolución. En resumen, el modelo mantiene su desempeño y escalabilidad dentro del rango planificado.

## (b) Análisis de infactibilidad

Que una solución sea infactible (o que el modelo es infeasible) significa que no existe ninguna combinación de valores que cumpla todas las restricciones del modelo al mismo tiempo.

En total, dos instancias de tipo *large* fueron declaradas infactibles. Esto puede deberse principalmente a la interacción entre las restricciones de cobertura (R1), carga diaria (R3) y descanso (R4). En ciertas combinaciones aleatorias de disponibilidad y demanda, puede ocurrir que la cantidad de trabajadores dispuestos para cubrir un turno crítico sea insuficiente o que las reglas de descanso hagan imposible una cobertura completa.

- En particular, R4 (*noche → mañana*) puede reducir la disponibilidad efectiva cuando los turnos nocturnos son muy numerosos.
- R3 (*máx. 2 turnos por día*) restringe aún más la flexibilidad, limitando las posibles reasignaciones.

Es importante destacar que esto no representa un error del modelo, sino una propiedad esperable en escenarios donde la demanda supera la capacidad disponible. De hecho, la existencia de instancias infactibles es útil para validar la robustez del generador y del modelo de optimización.

## (c) Visualización tipo calendario para instancias pequeñas

En las instancias pequeñas, el modelo permite visualizar fácilmente la asignación diaria por trabajador. La siguiente figura muestra un ejemplo de calendario generado por el script `generar_calendarios.py`, donde cada celda indica los turnos asignados a cada empleado durante la semana.

Instancia 1 - Asignación de Turnos							
	LUN Día 1	MAR Día 2	MIE Día 3	JUE Día 4	VIE Día 5	SAB Día 6	DOM Día 7
Trabajador 6	D (10) N (4)	D (10) N (4)	N (9)	D (5)	D (8)	D (10)	N (5)
	—	—	N (6)	N (10)	D (10) N (5)	—	—
Trabajador 5	N (4)	—	N (6)	N (10)	D (10) N (5)	—	—
	D (8)	—	D (8)	—	N (9)	D (10) N (9)	D (9)
Trabajador 4	D (8)	—	D (8)	—	N (9)	D (10) N (9)	D (9)
	D (4)	N (5)	—	D (6)	N (5)	—	—
Trabajador 3	N (8)	N (10)	D (8) N (6)	D (8)	D (9)	N (7)	D (3)
	—	D (10) N (8)	N (9)	—	—	D (6)	—
Trabajador 2	—	D (10) N (8)	N (9)	—	—	D (6)	—
	—	—	—	—	—	—	—
Trabajador 1	—	—	—	—	—	—	—
	—	—	—	—	—	—	—

Nota: Números entre paréntesis indican la disposición del trabajador (0-10)

Figura 3: Ejemplo de calendario de asignaciones para una instancia *small*.

En este calendario, se puede observar que las asignaciones respetan las restricciones R3 y R4: ningún trabajador realiza más de dos turnos diarios ni tiene secuencias noche-mañana. El modelo logra tener una distribución equilibrada de la carga laboral entre los trabajadores disponibles, maximizando la disposición total.

## Análisis de tiempos de resolución

Finalmente, se evaluó el tiempo de resolución promedio y su relación con el tamaño de las instancias y la cantidad de variables involucradas.

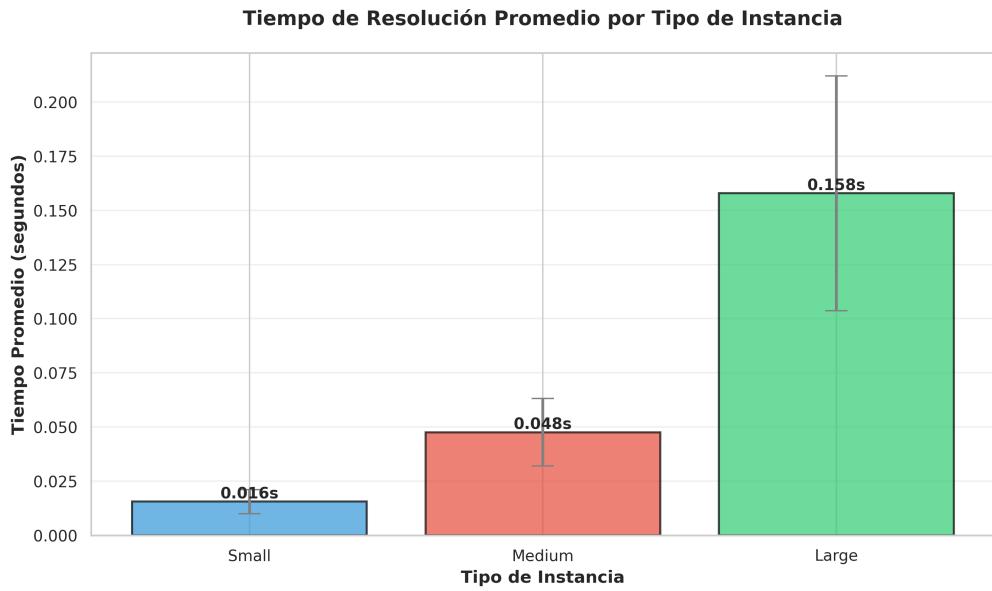


Figura 4: Tiempo promedio de resolución por tipo de instancia.

En el gráfico se puede observar que el tiempo crece de forma proporcional al pasar de instancias pequeñas a grandes. También se puede ver, que incluso en el caso *large*, el tiempo promedio se mantiene bajo (0.16 s), lo que confirma la eficiencia del modelo implementado.

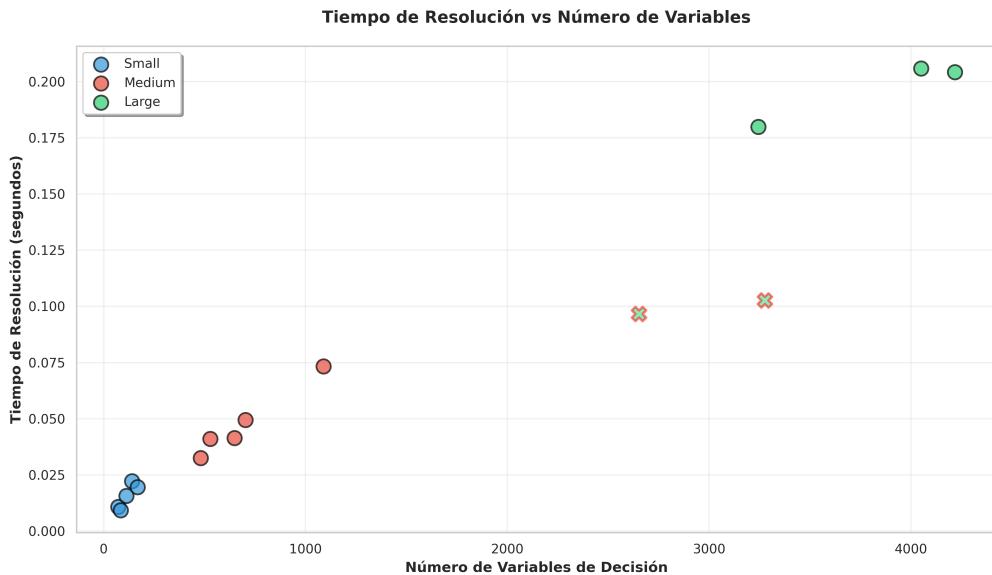


Figura 5: Tiempos de resolución en función del número de variables.

En la figura anterior se puede observar que, aunque el tiempo de resolución tiende a aumentar con el número de variables, este crecimiento no es abrupto ni exponencial, lo que indica que el modelo escala de forma controlada y que el solver maneja de forma eficiente los casos con mayor tamaño. En general, el comportamiento temporal es coherente con la naturaleza del modelo, donde el aumento de combinaciones posibles no afecta de manera drástica la capacidad de resolución gracias a la estructura binaria y a las restricciones acotadas del problema.

Al comparar los resultados de tiempo entre los tres tamaños, se evidencia que la diferencia entre *small* y *medium* es mínima, mientras que el salto hacia *large* representa el aumento esperado en complejidad, pero sin llegar a comprometer la factibilidad ni el rendimiento. Esto demuestra que el modelo es estable y eficiente, incluso cuando la cantidad de trabajadores, días o turnos se multiplica varias veces.

## Conclusión general

En conclusión, el modelo propuesto demuestra ser sólido, eficiente y coherente con los objetivos planteados. A lo largo de las distintas instancias, se ha podido confirmar que:

- El modelo mantiene la escalabilidad, incrementando el valor de la función objetivo de forma proporcional al tamaño de la instancia.
- Los tiempos de resolución son reducidos incluso en casos de gran tamaño, evidenciando una implementación eficiente en **LPSolve**.
- Las instancias infactibles se explican por condiciones realistas de sobrecarga de

demandas, lo cual logra validar la robustez del modelo.

- Las soluciones factibles respetan todas las restricciones definidas, reflejando una distribución equilibrada de la carga laboral.

Además, el análisis conjunto de los resultados y los tiempos de resolución permite concluir que el modelo no solo optimiza correctamente la asignación de personal, sino que también lo hace con un costo computacional bastante bajo. Esto sugiere que la formulación matemática podría escalar a entornos reales más amplios sin comprometer la eficiencia del proceso.

En resumen, el sistema de generación, modelamiento y resolución cumple con los criterios de calidad, desempeño y factibilidad requeridos, demostrando así, un comportamiento consistente, interpretable y aplicable en contextos reales de planificación de turnos.