

Entrega 1: Formulación Matemática y Generación de Instancias para el Problema de Asignación de Turnos

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

Octubre 2025

Resumen Ejecutivo

Este informe presenta la formulación de un modelo de **programación lineal entera binaria** para resolver el problema de *asignación de turnos en una clínica de atención integral*, considerando tanto la cobertura operativa como el bienestar del personal. El modelo incorpora restricciones clave como la carga máxima diaria, descansos entre turnos críticos y equidad en fines de semana trabajados, maximizando la disposición declarada por los trabajadores.

Además, se implementó un **generador de instancias sintéticas** que permite evaluar el modelo en distintos tamaños de problema (*small, medium, large*), generando datos de demanda y disposición de forma aleatoria bajo distribuciones controladas.

El lector encontrará en este documento la definición formal del modelo, junto con una descripción del generador de instancias, sentando las bases para su posterior resolución computacional y análisis en la segunda entrega.

1. Modelo Matemático

Conjuntos e Índices

T : Conjunto de trabajadores, indexado por $i \in T$.

D : Conjunto de días del horizonte de planificación, definido como

$$D = \{1, 2, \dots, H\},$$

donde H representa el número total de días. Cada día se indexa por $j \in D$.

S : Conjunto de turnos disponibles por día, indexado por $t \in S$.

Parámetros

$c_{i,j,t}$: Nivel de disposición del trabajador i para realizar el turno t del día j , donde $c_{i,j,t} \in \{0, 1, \dots, 10\}$.

Un valor de 0 indica que el trabajador no puede realizar ese turno.

$r_{j,t}$: Cantidad requerida de trabajadores para cubrir el turno t del día j , con $r_{j,t} \in \mathbb{N}$.

$a_{i,j,t}$: indica si el trabajador i está disponible para realizar el turno t del día j , definida como:

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

W_k : Conjunto de días correspondientes al fin de semana de la semana k , utilizado para controlar la asignación de turnos en sábados y domingos. Se define como:

$$W_k = \{\text{sábado de la semana } k, \text{ domingo de la semana } k\}, \quad \text{para } k = 1, \dots, K$$

donde el número total de semanas K se calcula como:

$$K = \left\lfloor \frac{H}{7} \right\rfloor$$

siendo H el número total de días del horizonte de planificación.

L_i : Límite máximo de turnos que el trabajador i puede ser asignado durante todo el horizonte de planificación. Este parámetro permite controlar la carga total de trabajo por persona, promoviendo equidad y evitando sobreasignaciones. Se puede definir como:

$$L_i = \left\lfloor \frac{2H}{3} \right\rfloor$$

donde H es el número total de días del horizonte.

Variables de decisión

$x_{i,j,t}$: Variable binaria que indica si el trabajador i es asignado al turno t del día j , definida como:

$$x_{i,j,t} = \begin{cases} 1, & \text{si el trabajador } i \text{ es asignado al turno } t \text{ del día } j, \\ 0, & \text{en caso contrario.} \end{cases}$$

$y_{i,k}$: Variable binaria que indica si el trabajador i realiza al menos un turno durante el fin de semana correspondiente a la semana k . Se define como:

$$y_{i,k} = \begin{cases} 1 & \text{si el trabajador } i \text{ realiza al menos un turno en el fin de semana } k, \\ 0 & \text{en caso contrario.} \end{cases}$$

Función Objetivo

Función Objetivo: Maximizar la disposición total del personal asignado, favoreciendo la asignación de turnos a quienes tienen mayor disposición declarada.

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

Restricciones

El modelo considera las siguientes restricciones para asegurar la factibilidad operativa y el bienestar del personal:

- (R1) Cobertura de turnos:** Cada turno debe ser cubierto exactamente por la cantidad requerida de trabajadores.

$$\sum_{i \in T} x_{i,j,t} = r_{j,t} \quad \forall j \in D, \forall t \in S$$

- (R2) Disponibilidad individual:** Un trabajador solo puede ser asignado a un turno si tiene disposición estrictamente positiva para realizarlo.

$$x_{i,j,t} \leq a_{i,j,t} \quad \forall i \in T, \forall j \in D, \forall t \in S$$

- (R3) Máximo de dos turnos por día:** Ningún trabajador puede realizar más de dos turnos en un mismo día.

$$\sum_{t \in S} x_{i,j,t} \leq 2 \quad \forall i \in T, \forall j \in D$$

- (R4) Descanso entre noche y mañana:** No se permite asignar a un trabajador al turno de noche de un día y al turno de mañana del día siguiente.¹

$$x_{i,j,\text{noche}} + x_{i,j+1,\text{mañana}} \leq 1 \quad \forall i \in T, \forall j \in D \setminus \{H\}$$

- (R5) No trabajar tres fines de semana consecutivos:** Un trabajador no puede estar asignado a turnos en tres fines de semana seguidos.

$$y_{i,k} + y_{i,k+1} + y_{i,k+2} \leq 2 \quad \forall i \in T, \forall k \in \{1, \dots, K-2\}$$

- (R5.1) Definición de actividad en fin de semana:** La variable $y_{i,k}$ se activa si el trabajador realiza al menos un turno en el fin de semana k .

$$\begin{aligned} y_{i,k} &\geq x_{i,d,t} \quad \forall d \in W_k, \forall t \in S \\ y_{i,k} &\leq \sum_{d \in W_k} \sum_{t \in S} x_{i,d,t} \quad \forall i \in T, \forall k \in \{1, \dots, K\} \end{aligned}$$

- (R6) Límite de carga total por trabajador:** Cada trabajador puede ser asignado como máximo a L_i turnos en todo el horizonte de planificación.

$$\sum_{j \in D} \sum_{t \in S} x_{i,j,t} \leq L_i \quad \forall i \in T$$

- (R7) Naturaleza de las variables:** Se definen como binarias las variables de asignación y de actividad en fin de semana.

$$x_{i,j,t} \in \{0, 1\}, \quad y_{i,k} \in \{0, 1\}$$

¹En instancias *small*, el conjunto de turnos S solo incluye *día* y *noche*, por lo que esta restricción no aplica.

2. Generador de Instancias

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 [1].

Además, el repositorio incluye un script de chequeo que genera resúmenes estadísticos por instancia, útiles para el análisis posterior. Este script complementa el generador y se encuentra en el mismo repositorio citado.

Lógica de Generación y Formato de Salida

- Se definen aleatoriamente los días y trabajadores según el tipo de instancia:
 - *Small*: 5–7 días, 5–15 trabajadores, turnos dia 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 una distribución normal truncada en cero, con media proporcional al número de trabajadores y turnos, y desviación estándar del 20 %. Luego:
 1. Se limita la demanda por la disponibilidad real del turno (trabajadores con disposición positiva).
 2. Si la suma diaria excede la capacidad máxima (`max_turnos_dia × trabajadores`), se escala proporcionalmente.
- Para cada trabajador, día y turno, se genera una disposición $c_{i,j,t} \sim \mathcal{U}\{0, 10\}$.
- Cada instancia se guarda en dos formatos:
 - `.json`: contiene `id_instancia, tipo, dias, trabajadores, demanda_dias y disposicion`.
 - `.csv`: archivo plano con las disposiciones individuales, con columnas:

```
trabajador, dia, turno, disposicion
```

Este formato permite inspeccionar rápidamente las preferencias de cada trabajador, pero no incluye la demanda ni otros metadatos.

Descripción del Código

El script realiza dos tareas principales:

1. **Inicialización:** Elimina las carpetas existentes en `../instancias/` para evitar duplicados y crea nuevas subcarpetas para cada tipo de instancia.
2. **Generación:** Para cada tipo de instancia, se generan 5 archivos con datos aleatorios. El script permite fijar una semilla manualmente para reproducir instancias específicas, o usar una semilla aleatoria por defecto (`random.seed()`) para obtener variabilidad entre ejecuciones. Para esta entrega, se utilizó la semilla 42, al igual que en ejecuciones anteriores, con el fin de mantener consistencia en las pruebas. El proceso incluye:
 - Asignación de días y trabajadores dentro de los rangos definidos.
 - Generación de nombres de días con formato `lunes_1`, `martes_1`, etc.
 - Cálculo de demanda por turno usando una distribución normal truncada en cero.
 - Ajuste de demanda por disponibilidad y escalado por capacidad diaria.
 - Generación de disposiciones por trabajador, día y turno.
 - Escritura de archivos `.json` y `.csv`.

Fragmento del Código

Listing 1: Generador de instancias con control de capacidad

```
def generar_instancias(max_turnos_dia=2, dispo_umbral=0):
    random.seed() # Semilla aleatoria para variabilidad
    semana_dias = ["lunes", "martes", "miercoles", "jueves", "viernes", "sabado", "domingo"]

    tipos = ["small", "medium", "large"]
    dias_rangos = [(5, 7), (7, 14), (14, 28)]
    trabajadores_rangos = [(5, 15), (15, 45), (45, 90)]
    turnos_rangos = [{"dia": "noche"}, {"dia": "manana"}, {"dia": "tarde"}]
    turnos_rangos += [{"dia": "tarde"}, {"dia": "noche"}]

    id_instancia = 1
    for tipo, dias_rng, trab_rng, turnos in zip(tipos, dias_rangos, trabajadores_rangos,
                                                turnos_rangos):
        for i in range(5): # 5 instancias por tipo
            dias = random.randint(*dias_rng)
            trabajadores = random.randint(*trab_rng)
            cantidad_turnos = len(turnos)
            # Generacion de disposicion y demanda omitida por brevedad
```

3. Análisis de Factibilidad

El análisis de factibilidad evalúa si las instancias generadas permiten soluciones que cumplan las restricciones del modelo matemático. El generador no fuerza factibilidad, pero incorpora controles estructurales que favorecen la viabilidad operativa.

Controles incorporados

- **Cobertura por turno (R1):** La demanda por turno se ajusta según la cantidad de trabajadores con disposición positiva, evitando requerimientos imposibles de cubrir.
- **Capacidad diaria (R3):** La suma de la demanda por día se limita a un máximo de dos turnos por trabajador, respetando la restricción de carga diaria.
- **Carga total (R6):** Aunque el parámetro $L_i = \lfloor \frac{2H}{3} \rfloor$ no se usa directamente en el generador, se respeta de forma indirecta al limitar la demanda diaria y la carga agregada por trabajador. Esto favorece la factibilidad sin imponerla explícitamente.
- **Fines de semana (R5):** Se controla que la demanda agregada en sábados y domingos no exceda la capacidad global del equipo.

Chequeo automatizado

Se implementaron tres verificaciones automáticas:

- **Check A:** Demanda por turno \leq disponibilidad efectiva.
- **Check B:** Demanda diaria \leq capacidad total.
- **Check C:** Demanda en fines de semana \leq capacidad agregada.

Estas verificaciones permiten detectar y corregir sobrecargas antes de resolver el modelo.

Restricciones delegadas al modelo

Las siguientes restricciones dependen de decisiones de asignación y no pueden verificarse en la etapa de generación o chequeo. Por ello, se delegan a la resolución del modelo:

- **(R2)** Asignación solo si el trabajador tiene disposición positiva.
- **(R4)** Prohibición de turno noche seguido de mañana.
- **(R5)** No trabajar tres fines de semana consecutivos.

Observaciones

Las instancias *small* tienden a ser factibles. En *medium* y *large*, la factibilidad depende de la relación entre demanda y disposición. La aleatoriedad permite explorar escenarios diversos, incluyendo casos límite útiles para validar el modelo.

Referencias

- [1] Matías Romo Vargas et al. *proyecto-inf292-Generador de Instancias para Asignación de Turnos Hospitalarios*. Accedido en octubre de 2025. 2025. URL: <https://github.com/josephors/proyecto-inf292#>.