

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].

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 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 una **distribución normal truncada**, con media proporcional al número de trabajadores y turnos, y desviación estándar del 20 %. Esto permite simular variabilidad en la carga de trabajo diaria.
- Para cada trabajador, día y turno, se genera una disposición $c_{i,j,t} \sim \mathcal{U}\{0, 10\}$, donde 0 indica que el trabajador no puede realizar ese turno. Esta variabilidad permite representar preferencias y restricciones personales.
- Cada instancia se guarda en dos formatos:
 - **.json**: contiene la estructura completa de la instancia, incluyendo:
 - **meta**: metadatos como el tipo de instancia, número de días y trabajadores.
 - **sets**: definición de los conjuntos T, D, S .
 - **demand**: matriz con los valores de $r_{j,t}$.
 - **preferences**: matriz tridimensional con los valores de $c_{i,j,t}$.
 - **.csv**: archivo plano con las disposiciones individuales, útil para inspección rápida o visualización.

Descripción del Código

El script realiza dos tareas principales:

1. **Inicialización:** Elimina las carpetas existentes en `generador/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. Se utiliza una semilla aleatoria para permitir variabilidad entre ejecuciones y facilitar pruebas de factibilidad. 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., para facilitar la identificación de semanas.
 - Cálculo de demanda por turno usando una distribución normal con media ajustada dinámicamente.
 - Generación de disposiciones por trabajador, día y turno, con valores enteros entre 0 y 10.
 - Cálculo del límite máximo de turnos $L_i = \lfloor \frac{2H}{3} \rfloor$ por trabajador, en cumplimiento de la restricción (R6) del modelo matemático.
 - Escritura de archivos `.json` y `.csv` con la información generada.

Fragmento del Código

```
def generar_instancias():
    random.seed(1234)
    tipos = ["small", "medium", "large"]
    dias_rangos = [(5, 7), (7, 14), (14, 28)]
    trabajadores_rangos = [(5, 15), (15, 45), (45, 90)]
    turnos_rangos = [{"dia": "dia", "noche": "noche"}, {"manana": "manana", "tarde": "tarde"}, {"manana": "manana", "tarde": "tarde"}, {"noche": "noche"}]
    for tipo, dias_rng, trab_rng, turnos in zip(tipos, dias_rangos, trabajadores_rangos, turnos_rangos):
        for _ in range(5):
            dias = random.randint(*dias_rng)
            trabajadores = random.randint(*trab_rng)
            cantidad_turnos = len(turnos)
            demanda_dias = {}
            disposicion = []
            for j in range(dias):
                mu = (trabajadores / cantidad_turnos) * random.uniform(1.0, 1.2)
                sigma = mu * 0.2
                demandas_turnos = {t: max(0, int(random.normalvariate(mu, sigma))) for t in turnos}
                demanda_dias[f"dia_{j+1}"] = demandas_turnos
            for i in range(1, trabajadores + 1):
                for d in range(1, dias + 1):
                    for t in turnos:
                        dispo = random.randint(0, 10)
                        disposicion.append({"trabajador": i, "dia": d, "turno": t, "disposicion": dispo})
    # Escritura de archivos omitida por brevedad
```

El código completo incluye la escritura de archivos y está documentado en el repositorio citado.

3. Análisis de Factibilidad

El análisis de factibilidad evalúa si las instancias generadas permiten soluciones que cumplan las restricciones del modelo. Dado que el generador utiliza datos aleatorios, pueden aparecer casos infactibles, especialmente en instancias grandes o con parámetros extremos. Por ello, el diseño del generador considera explícitamente los factores que más afectan la factibilidad, con el objetivo de producir instancias razonables y representativas.

Factores que influyen

- **Cobertura por turno:** Para cada día y turno, la demanda se ajusta según la capacidad disponible, es decir, el número de trabajadores que pueden cubrir ese turno según su disposición. Esto evita generar demandas imposibles de cubrir.
- **Capacidad diaria:** La suma de la demanda de todos los turnos de un día se limita a la capacidad total teórica del personal, considerando un máximo de turnos por trabajador por día (por ejemplo, dos). Esto previene sobrecargas que violarían restricciones del modelo.
- **Congestión en fines de semana:** Los días sábado y domingo suelen tener menor disponibilidad. El generador controla que la demanda agregada en estos días no supere la capacidad global del equipo, considerando el límite total de turnos por trabajador en el horizonte.

Observaciones

- Las instancias *small* tienden a ser factibles, mientras que en *medium* y *large* la factibilidad depende de la relación entre demanda y disposición.
- El diseño del generador evita casos triviales o claramente infactibles, lo que permite concentrarse en instancias relevantes para el modelo.
- La aleatoriedad en la generación permite explorar una variedad de escenarios, incluyendo casos límite útiles para validar la robustez del enfoque.
- El control sobre parámetros como la media de demanda, los umbrales de disposición y el límite de turnos por trabajador permite ajustar la dificultad de las instancias.

Justificación del generador

El generador fue diseñado para producir instancias que respeten las restricciones estructurales del modelo, en particular aquellas relacionadas con la capacidad de cobertura y la carga máxima por trabajador. El uso de distribuciones probabilísticas permite simular situaciones realistas y diversas, facilitando el análisis de factibilidad y el estudio del comportamiento del modelo bajo distintas condiciones. Además, al incorporar directamente el límite de turnos por trabajador ($L_i = \lfloor \frac{2H}{3} \rfloor$), se garantiza el cumplimiento de la restricción (R6) desde la etapa de generación.

Este análisis se complementará en la Entrega 2 con la resolución efectiva de las instancias y el estudio de resultados.

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#>.