

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

Objetivo

Maximizar la **disposición total del personal asignado**, asegurando simultáneamente el cumplimiento de las **restricciones operativas** y de **bienestar del personal**, tales como la cobertura de turnos, la carga laboral diaria, los descansos adecuados y la equidad en la asignación de fines de semana.

La función objetivo se define como la suma de las disposiciones individuales de los trabajadores asignados a cada turno, buscando así una planificación que favorezca la satisfacción del personal sin comprometer la continuidad del servicio.

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

H : Número total de días del horizonte de planificación. Define la cardinalidad del conjunto D .

$c_{i,j,t}$: 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 $r_{j,t} \in \mathbb{N}$.

W_j : Indicador binario que vale 1 si el día j corresponde a un fin de semana (sábado o domingo), y 0 en caso contrario.

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:

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

Función Objetivo

El objetivo del modelo es **maximizar la disposición total del personal asignado**, es decir, la suma de los niveles de disposición de los trabajadores que efectivamente son asignados a turnos. Esto se representa mediante la siguiente función:

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

donde:

- $x_{i,j,t}$ indica si el trabajador i es asignado al turno t del día j ,
- $c_{i,j,t}$ representa la disposición del trabajador i para trabajar en esa combinación día-turno.

Esta función busca asignar turnos a quienes tienen mayor disposición, promoviendo así una planificación que favorezca el bienestar del personal sin comprometer la cobertura requerida.

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.

$$x_{i,j,t} \leq \begin{cases} 1, & \text{si } c_{i,j,t} > 0 \\ 0, & \text{si } c_{i,j,t} = 0 \end{cases} \quad \forall i \in T, \forall j \in D, \forall t \in S$$

- (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.

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

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

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

Para la restricción (R5), se define una variable auxiliar $w_{i,k} \in \{0, 1\}$ que indica si el trabajador i realiza al menos un turno durante el fin de semana k . Esta variable se relaciona con las asignaciones reales mediante:

$$w_{i,k} \geq x_{i,d,t} \quad \forall d \in \text{días del fin de semana } k, \forall t \in S$$

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 fija (`random.seed(1234)`) para asegurar reproducibilidad. 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.
- 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": "noche"}, {"dia": "tarde"}, {"dia": "manana"}]
    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 busca evaluar si las instancias generadas permiten encontrar soluciones viables que cumplan todas las restricciones del modelo. Dado que el generador produce datos de forma aleatoria, es posible que algunas instancias resulten infactibles, especialmente en los casos de mayor tamaño o con parámetros extremos.

Factores que afectan la factibilidad

- **Demanda vs. oferta:** La demanda por turno y día se genera mediante una distribución normal truncada, con media proporcional al número de trabajadores disponibles. Sin embargo, al tratarse de una generación aleatoria, pueden producirse casos en que la demanda total supere la capacidad máxima del personal, especialmente si muchos trabajadores tienen baja o nula disposición.
- **Disposición del personal:** La disposición $c_{i,j,t} \sim \mathcal{U}\{0, 10\}$ puede generar combinaciones en las que no haya suficientes trabajadores disponibles para ciertos turnos, lo que impide cumplir la cobertura requerida.
- **Restricciones de bienestar:** Condiciones como el máximo de dos turnos por día, la prohibición de turnos noche-mañana consecutivos y la limitación de fines de semana trabajados pueden restringir aún más el espacio de soluciones factibles, especialmente en instancias con pocos trabajadores o alta demanda.

Observaciones preliminares

Durante la revisión manual de las instancias generadas, se observó que:

- Las instancias *small* tienden a ser factibles con mayor frecuencia, dado que las restricciones son menos exigentes y el número de turnos es menor.
- En instancias *medium* y *large*, la factibilidad depende fuertemente de la relación entre la demanda generada y la disposición del personal. En algunos casos, se detectaron instancias infactibles debido a una combinación de alta demanda y baja disposición.
- La aleatoriedad en la generación permite evaluar el modelo en escenarios diversos, incluyendo casos límite que son útiles para validar la robustez del enfoque.

Justificación del diseño del generador

El uso de distribuciones probabilísticas permite simular situaciones realistas y variadas, incluyendo tanto instancias factibles como infactibles. Esto es intencional, ya que permite:

- Evaluar la capacidad del modelo para detectar infactibilidades.
- Analizar cómo las restricciones afectan la existencia de soluciones.
- Ajustar parámetros del generador para controlar la dificultad de las instancias.

Este análisis será complementado en la Entrega 2 mediante la resolución efectiva de las instancias y el estudio de los resultados obtenidos.

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#>.