



Facultad de Ciencias Económicas y Empresariales

ICADE

**OPTIMIZACIÓN DE TURNOS DE
ENFERMERÍA EN UNA UNIDAD DE
HOSPITALIZACIÓN: ANÁLISIS DEL
COMPROMISO ENTRE EFICIENCIA Y
BIENESTAR LABORAL**

Clave: 202100637

Autor: Mario Laborda Plaza

Director: María José Martín Rodrigo

MADRID | Junio 2026

Resumen

La planificación de turnos de enfermería en una unidad de hospitalización exige conciliar objetivos contrapuestos: cubrir la demanda asistencial al menor coste y, a la vez, ofrecer al personal un calendario equitativo y compatible con su vida personal. En un contexto de escasez estructural de profesionales, esta tensión entre eficiencia y bienestar adquiere especial relevancia. Este trabajo formula y resuelve dos modelos de optimización que compartan las mismas restricciones duras y se diferencian en su función objetivo: un Modelo A de referencia, orientado a la eficiencia, y un Modelo B que incorpora seis criterios de bienestar laboral mediante términos ponderados. Ambos se implementan con el solver exacto CP-SAT de OR-Tools y se aplican a tres instancias del benchmark público de Curtois y Qu, tras validar el Modelo A frente a sus óptimos publicados. Variando de forma sistemática los pesos de bienestar se cuantifica el compromiso entre ambos objetivos. Los resultados revelan una frontera con rendimientos decrecientes, en la que una parte sustancial del bienestar, como la reducción de las rachas de días consecutivos trabajados, se obtiene a un coste de eficiencia inferior al uno por ciento, mientras que las exigencias más ambiciosas tienen un coste que depende fuertemente de cada unidad. La principal aportación es convertir una decisión tradicionalmente implícita en un compromiso explícito, cuantificado y, por tanto, gestionable.

Palabras clave: planificación de turnos de enfermería; optimización; programación con restricciones; bienestar laboral; compromiso eficiencia-bienestar; gestión de recursos humanos sanitarios.

Abstract

Nurse scheduling in a hospitalization unit requires reconciling conflicting objectives: covering care demand at the lowest cost while, at the same time, offering staff a fair roster compatible with their personal life. In a context of structural shortage of professionals, this tension between efficiency and well-being becomes particularly relevant. This work formulates and solves two optimization models that share the same hard constraints and differ in their objective function: a baseline Model A, oriented toward efficiency, and a Model B that incorporates six staff well-being criteria through weighted terms. Both are implemented with the exact CP-SAT solver from OR-Tools and applied to three instances of the public Curtois and Qu benchmark, after validating Model A against its published optima. By systematically varying the well-being weights, the trade-off between the two objectives is quantified. The results reveal a trade-off frontier with diminishing returns, in which a substantial part of the well-being gains, such as reducing stretches of consecutive working days, is obtained at an efficiency cost below one percent, whereas more ambitious demands carry a cost that depends heavily on each unit. The main contribution is to turn a traditionally implicit decision into an explicit, quantified, and therefore manageable trade-off.

Keywords: nurse rostering; optimization; constraint programming; staff well-being; efficiency-well-being trade-off; healthcare human resource management.

Índice general

1	Introducción	9
1.1	Justificación del tema y motivación personal	9
1.2	Objetivo y preguntas de investigación	10
1.3	Metodología	11
1.4	Estructura del documento	12
2	Marco conceptual y revisión de la literatura	13
2.1	La problemática de los turnos en hospitalización	13
2.2	El Nurse Scheduling/Rostering Problem: definición	16
2.3	Restricciones duras y blandas	17
2.4	Indicadores de bienestar laboral	19
2.5	Metodologías de optimización	21
2.6	Estado del arte reciente y revisiones sistemáticas	22
3	Dataset y preparación de datos	25
3.1	Descripción del benchmark principal (Curtois & Qu)	25
3.2	Otros benchmarks de referencia	27
3.3	Formato y preprocesamiento	29
3.3.1	Formatos disponibles	29
3.3.2	Estructura de una instancia: secciones y campos	30
3.3.3	Carga y preprocesamiento	30
3.4	Selección y caracterización de la muestra experimental	31
3.4.1	Criterios de selección	31
3.4.2	Muestra seleccionada y su caracterización	32
3.4.3	Alcance experimental y justificación metodológica	33
4	Modelización del problema	34
4.1	Conjuntos, parámetros y variables de decisión	34
4.1.1	Conjuntos	34
4.1.2	Parámetros	35

4.1.3	Variables de decisión	35
4.2	Restricciones duras	36
4.3	Restricciones blandas y variables de penalización	38
4.4	Función objetivo del Modelo A (baseline)	40
4.5	Función objetivo del Modelo B (bienestar)	40
4.5.1	Los seis términos de bienestar	41
4.5.2	Función objetivo combinada y relación con el Modelo A	43
5	Resolución y herramientas	44
5.1	Herramientas seleccionadas	44
5.2	Implementación del modelo	45
5.3	Configuración del solver	48
5.4	Diseño experimental	49
5.4.1	Muestra y obtención de las instancias	49
5.4.2	Esquema de pesos del Modelo B	49
5.4.3	Métricas registradas	50
5.4.4	Protocolo de validación y reproducibilidad	50
6	Experimentos y resultados	52
6.1	Validación del modelo y configuración de referencia	52
6.2	Resultados del Modelo B	54
6.2.1	Calibración de los pesos	54
6.2.2	Análisis término a término	55
6.2.3	Escenarios combinados	58
6.3	Comparación global y análisis de los compromisos	59
7	Discusión	61
7.1	Interpretación de los resultados	61
7.2	Implicaciones para la gestión de RRHH	62
7.3	Limitaciones y riesgos	62
8	Conclusiones	64
8.1	Síntesis de hallazgos	64
8.2	Recomendaciones prácticas	65
8.3	Líneas futuras	65
	Declaración de uso de IA	67
	Bibliografía y referencias	69

Índice de Figuras

2.1	Evolución del déficit mundial de enfermería (millones), 2020–2030. . . .	14
2.2	Ratio de enfermeras por cada 1.000 habitantes: España frente a la media de la UE y a los países con mayor dotación	15
6.1	Frontera de compromiso entre eficiencia y rachas largas para las tres instancias.	56
6.2	Precio del bienestar: coste de eficiencia de la mejor mejora alcanzada en cada dimensión, por instancia.	57
6.3	Escenarios combinados para las tres instancias: número de rachas largas en la configuración de referencia y en los perfiles moderado y alto. . . .	59

Índice de cuadros

3.1	Características de las instancias del benchmark de Curtois & Qu	27
3.2	Caracterización de la muestra experimental.	32
6.1	Validación del Modelo A frente a los óptimos publicados de Curtois y Qu.	52
6.2	Indicadores de bienestar de la configuración de referencia (solución eficiente óptima).	53
6.3	Niveles de peso por tipo de término de bienestar.	55
6.4	Mejor mejora alcanzada por cada término de bienestar respecto a la configuración de referencia, con su coste de eficiencia.	56
6.5	Escenarios combinados: indicadores principales y coste de eficiencia frente a la configuración de referencia.	58
6.6	Efecto del perfil combinado moderado, con el coste de eficiencia expresado en valor absoluto y como porcentaje de la configuración de referencia.	60

8.1	Parámetros de las instancias utilizadas.	72
8.2	Niveles de peso calibrados por término de bienestar.	89
8.3	Perfiles combinados (nivel medio y alto de cada término).	89
8.4	Resultados completos. Instance1 (8 enfermeras, 1 turno).	90
8.5	Resultados completos. Instance2 (14 enfermeras, 2 turnos).	91
8.6	Resultados completos. Instance3 (20 enfermeras, 3 turnos).	91

Capítulo 1

Introducción

1.1. Justificación del tema y motivación personal

La elaboración del cuadrante de turnos es una de las tareas de gestión más recurrentes y delicadas de una unidad de hospitalización. Decidir qué enfermera trabaja cada turno a lo largo de varias semanas obliga a conciliar objetivos que tiran en direcciones opuestas: garantizar la cobertura asistencial las veinticuatro horas, cumplir la normativa laboral y, al mismo tiempo, ofrecer al personal un calendario equitativo y compatible con su vida personal. Esa tensión, intrínseca al problema, es la que da sentido a este trabajo.

Su relevancia se ha agudizado en los últimos años por la escasez estructural de profesionales de enfermería, un reto sanitario global que el marco conceptual documenta con detalle. En un contexto de plantillas escasas y difíciles de reponer, cada profesional disponible resulta más valioso, y la forma en que se planifican sus turnos deja de ser un asunto puramente operativo para convertirse en una palanca de retención: una planificación que acelere el agotamiento agrava la falta de personal, mientras que una cuidadosa ayuda a conservarlo.

Tradicionalmente, esta planificación se ha resuelto a mano, un procedimiento lento, propenso a errores y difícil de optimizar por la naturaleza combinatoria del problema. Las herramientas de analítica y de optimización ofrecen una alternativa sistemática, capaz de explorar el espacio de soluciones y de cuantificar el equilibrio entre objetivos contrapuestos, lo que las hace especialmente pertinentes para apoyar la decisión de la dirección de enfermería.

Ahora bien, la mayor parte de los enfoques de optimización ha orientado su objetivo hacia la eficiencia, es decir, hacia cubrir la demanda al menor coste, mientras que la incorporación explícita del bienestar del personal como objetivo, y la medición de cuánto cuesta

perseguirlo, han recibido una atención escasa. Precisamente en ese hueco se sitúa este trabajo, que adopta una perspectiva de Business Analytics: no se limita a generar un cuadrante, sino que emplea la optimización para cuantificar el compromiso entre eficiencia y bienestar y convertirlo en información útil para la gestión. Esa orientación, que une el rigor de la optimización con una pregunta de gestión de recursos humanos, es la que justifica el tema y enmarca los objetivos que se detallan a continuación.

En el plano personal, la elección de este tema nace de la confluencia de dos intereses. El primero es académico: de toda la carrera de Business Analytics, fueron las asignaturas de optimización las que más me atrajeron, tanto por el atractivo intelectual de resolver los problemas matemáticos como por su capacidad para incidir en la vida real y mejorar las estructuras que organizan el trabajo de las personas. El segundo es vocacional: siempre he sentido una profunda admiración por los profesionales sanitarios, hasta el punto de que la medicina fue una de las opciones que consideré antes de decantarme por mi grado, atraído por esa vocación de ayudar a los demás y por el propósito que da a su trabajo. Que hoy sea tan visible la presión laboral que soporta el personal sanitario no hace sino reforzar esa motivación: poder dedicar este trabajo a una herramienta que, siquiera modestamente, contribuya a mejorar sus condiciones laborales le da al esfuerzo un sentido que trasciende el ejercicio puramente académico.

1.2. Objetivo y preguntas de investigación

El objetivo general de este trabajo es formular y resolver dos modelos de planificación de turnos de enfermería para una unidad de hospitalización, uno de referencia orientado a la eficiencia (Modelo A) y otro que incorpora de forma explícita criterios de bienestar laboral (Modelo B), con el fin de cuantificar el compromiso entre eficiencia y bienestar y analizar sus implicaciones para la gestión.

Este objetivo general se desglosa en cuatro objetivos específicos:

1. Formular un modelo de referencia (Modelo A) que reproduzca el enfoque tradicional centrado en la eficiencia y validarlo frente a los óptimos publicados de un benchmark de referencia.
2. Extender ese modelo a un Modelo B que incorpore seis criterios de bienestar laboral mediante términos ponderados en la función objetivo.
3. Diseñar y ejecutar un conjunto de experimentos que, variando los pesos de bienestar, permita medir cuánto mejora cada criterio y cuánta eficiencia cuesta esa mejora.
4. Analizar de forma cuantitativa el compromiso resultante y extraer recomendaciones

aplicables a la dirección de enfermería.

A partir de estos objetivos, el trabajo se articula en torno a tres preguntas de investigación:

1. ¿Existe un compromiso cuantificable entre eficiencia y bienestar en la planificación de turnos, y qué forma adopta?
2. ¿Cuánto bienestar puede ganarse y a qué coste de eficiencia? En particular, ¿existen mejoras de bienestar asequibles, alcanzables a un coste muy reducido?
3. ¿Es ese coste estable, o depende de las características concretas de cada instancia y de cada unidad?

La respuesta a estas tres preguntas constituye el núcleo del trabajo y se aborda a lo largo de los capítulos de modelización, experimentación y discusión.

1.3. Metodología

La metodología de este trabajo es de carácter cuantitativo y se apoya en la optimización exacta. Los modelos se formulan como programas con restricciones y se resuelven con el solver CP-SAT de la biblioteca OR-Tools, en Python, una combinación de software libre elegida por su capacidad para demostrar la optimalidad de las soluciones y por la reproducibilidad que aporta. Como base de datos experimental se emplea el benchmark público de Curtois y Qu, ampliamente adoptado en la literatura, del que se selecciona un subconjunto de tres instancias cuyo óptimo es conocido.

Sobre esos datos se construyen dos modelos que comparten exactamente las mismas restricciones duras y se diferencian únicamente en la función objetivo. El Modelo A reproduce el enfoque tradicional de eficiencia y se valida comprobando que recupera los óptimos publicados, lo que garantiza la corrección de la implementación antes de experimentar. El Modelo B añade a ese objetivo seis términos de bienestar laboral, cada uno ponderado por un peso que regula su importancia relativa.

El núcleo experimental consiste en variar esos pesos de forma sistemática, primero término a término, para aislar el efecto de cada criterio, y después en perfiles que combinan todos los criterios a la vez. De cada ejecución se registran el coste en eficiencia y los indicadores de bienestar en sus unidades naturales, lo que permite trazar el compromiso entre ambos y analizarlo de forma cuantitativa. El detalle de los datos, la formulación, las herramientas y el diseño experimental se desarrolla en los capítulos siguientes.

1.4. Estructura del documento

El presente trabajo se organiza en ocho capítulos. Tras esta introducción, el segundo capítulo desarrolla el marco conceptual y la revisión de la literatura: define el problema de planificación de turnos de enfermería, distingue entre restricciones duras y blandas, presenta los indicadores de bienestar laboral y repasa las metodologías de optimización y el estado del arte. El tercer capítulo describe el conjunto de datos, esto es, el benchmark de Curtois y Qu, su formato y preprocesamiento, y la selección de la muestra experimental. El cuarto capítulo formula matemáticamente los dos modelos, detallando conjuntos, parámetros, variables, restricciones y las dos funciones objetivo. El quinto capítulo expone las herramientas empleadas, la implementación del modelo y la configuración del solver, junto con el diseño experimental. El sexto capítulo presenta los experimentos y sus resultados, desde la validación del modelo hasta el análisis del compromiso entre eficiencia y bienestar. El séptimo capítulo discute e interpreta esos resultados, sus implicaciones para la gestión de recursos humanos y sus limitaciones. Finalmente, el octavo capítulo sintetiza los hallazgos, formula recomendaciones prácticas y propone líneas de trabajo futuras. Cierran el documento la bibliografía y los anexos, que recogen las instancias utilizadas, el código fuente y la documentación de los experimentos.

Capítulo 2

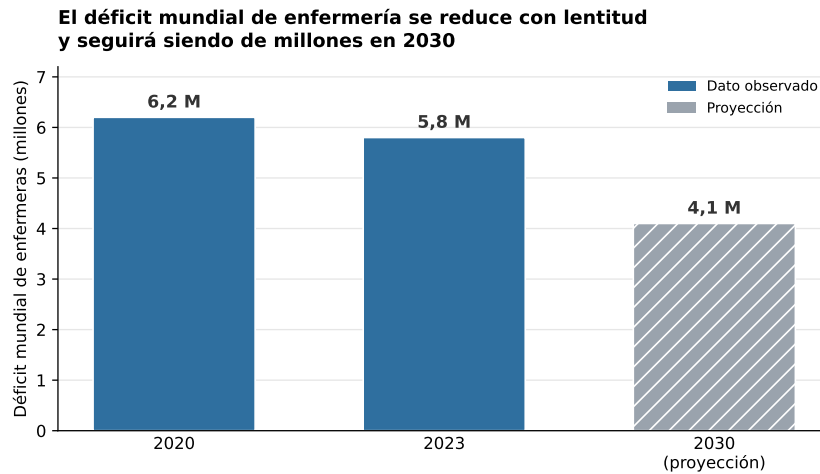
Marco conceptual y revisión de la literatura

2.1. La problemática de los turnos en hospitalización

Una unidad de hospitalización presta cuidados de forma ininterrumpida durante las veinticuatro horas del día y los siete días de la semana, lo que obliga a garantizar en todo momento una dotación de enfermería suficiente y con la cualificación adecuada. Esta exigencia de cobertura continua, combinada con la variabilidad de la demanda asistencial, las ausencias imprevistas y la multiplicidad de reglas laborales y contractuales que deben respetarse, convierte la elaboración del cuadrante en una tarea intrínsecamente compleja y conflictiva (Burke et al., 2004). El planificador debe conciliar simultáneamente objetivos que tiran en direcciones opuestas: cubrir la demanda al menor coste, cumplir la normativa laboral y, al mismo tiempo, ofrecer al personal un calendario equitativo y compatible con su vida personal.

La pertinencia de este problema se ha intensificado en los últimos años a causa de la escasez estructural de profesionales de enfermería, hoy reconocida como un reto sanitario global. Según el informe *State of the World's Nursing 2025* de la Organización Mundial de la Salud, la fuerza de trabajo de enfermería mundial alcanzó los 29,8 millones de profesionales en 2023, pero persiste un déficit estimado en 5,8 millones, distribuido además de forma muy desigual, ya que cerca del 78 % de las enfermeras se concentra en países que reúnen menos de la mitad de la población mundial (World Health Organization, 2025). Aunque ese déficit se ha reducido respecto a los 6,2 millones de 2020 y se proyecta que descienda hasta 4,1 millones en 2030, seguirá midiéndose en millones de profesionales (véase la Figura 2.1). En el ámbito europeo, el informe *Health at a Glance: Europe 2024*

advierte de que la escasez de personal sanitario exige medidas urgentes y proyecta que en torno a un tercio de las enfermeras en activo en Europa se jubilará en la próxima década (OECD/European Union, 2024).



Fuente: elaboración propia a partir de OMS, State of the World's Nursing 2025.

Figura 2.1: Evolución del déficit mundial de enfermería (millones), 2020–2030.

Fuente: *Elaboración propia a partir de World Health Organization (2025)*

España ilustra con especial crudeza esta tensión. Con alrededor de 6 enfermeras por cada 1.000 habitantes, frente a una media europea cercana a 8,4, el país se sitúa entre las ratios más bajas del entorno (OECD/European Union, 2024), de modo que el Ministerio de Sanidad cifra en unas 100.000 las enfermeras adicionales necesarias para converger con la media de la Unión Europea (Ministerio de Sanidad, 2024). La Figura 2.2 muestra esta brecha de forma comparada. En un contexto de personal escaso, envejecido y difícil de reponer, cada profesional disponible resulta más valioso, y la forma en que se planifican sus turnos adquiere una relevancia crítica: una planificación deficiente que acelere el agotamiento o el abandono agrava directamente un problema de dotación ya de por sí severo, mientras que una planificación cuidadosa puede contribuir a retener al personal existente.

En la práctica clínica, esta planificación se ha realizado tradicionalmente de forma manual, normalmente por una supervisora o un responsable de unidad que construye el cuadrante a partir de su experiencia y de un conjunto de reglas implícitas. Este procedimiento consume una cantidad considerable de tiempo de gestión, es propenso a errores y resulta difícil de optimizar cuando el número de enfermeras, días y turnos crece, dada la naturaleza combinatoria del problema (Cheang et al., 2003). Además, los cuadrantes elaborados manualmente suelen percibirse como poco transparentes y desiguales, lo que genera insatisfacción cuando determinadas cargas, como los fines de semana o las noches, no se reparten de manera equilibrada (Van den Bergh et al., 2013). Las consecuencias de una planificación deficiente o de una dotación insuficiente trascienden el ámbito puramente

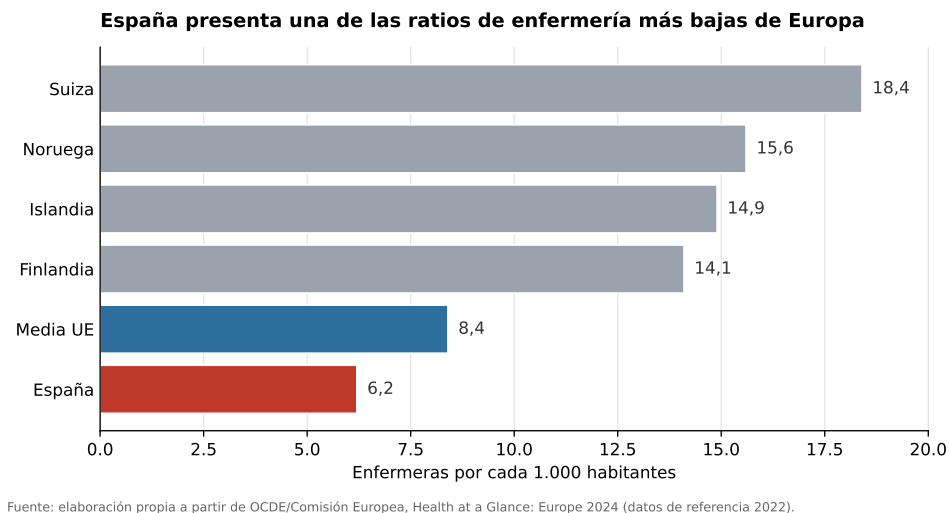


Figura 2.2: Ratio de enfermeras por cada 1.000 habitantes: España frente a la media de la UE y a los países con mayor dotación.
Fuente: Elaboración propia a partir de OECD/European Union (2024)

organizativo y afectan directamente a la seguridad del paciente. La evidencia muestra que niveles inadecuados de personal de enfermería se asocian con peores resultados clínicos: en un estudio de referencia, cada paciente adicional por enfermera se relacionó con un incremento del riesgo de mortalidad a treinta días y de las muertes evitables tras complicaciones en pacientes quirúrgicos, así como con un aumento del 23 % en la probabilidad de agotamiento profesional y del 15 % en la de insatisfacción laboral del personal (Aiken et al., 2002). Dado que es precisamente el cuadrante el que determina cuántas enfermeras están presentes en cada turno, la calidad de la planificación tiene una incidencia indirecta pero relevante sobre estos resultados.

El impacto sobre el propio personal constituye la otra cara del problema. Determinadas características de los turnos, en especial la rotación y la acumulación de noches, se asocian de forma significativa con el agotamiento profesional, con un efecto particularmente acusado sobre el agotamiento emocional (Qtait et al., 2025). El malestar derivado de calendarios poco compatibles con la vida personal se traduce, a su vez, en mayor absentismo, mayor intención de abandono y dificultades de retención, fenómenos que agravan la ya tensionada disponibilidad de profesionales y que generan costes adicionales para la organización. En sentido contrario, la evidencia sobre sistemas de planificación electrónica y de autoplanificación apunta a mejoras en la satisfacción del personal, en el equilibrio entre vida laboral y personal y en la reducción del absentismo, aunque condicionadas a la calidad de su implantación (O’Connell et al., 2024).

De este modo, la problemática de los turnos en hospitalización puede sintetizarse como una tensión persistente entre eficiencia y bienestar. Garantizar la cobertura asistencial al

menor coste y respetar simultáneamente las preferencias, la equidad y la conciliación del personal son objetivos que con frecuencia entran en conflicto, y que el método manual tradicional difícilmente resuelve de forma satisfactoria. Esta tensión es la que justifica el recurso a herramientas analíticas y de optimización capaces de explorar sistemáticamente el espacio de soluciones, cuantificar el equilibrio entre objetivos contrapuestos y apoyar la toma de decisiones de la dirección de enfermería. Sobre esta motivación se construye el resto del presente trabajo.

2.2. El Nurse Scheduling/Rostering Problem: definición

El problema de planificación de turnos de enfermería consiste en construir, para un conjunto de enfermeras y a lo largo de un horizonte de varias semanas, un calendario que especifique qué turno, mañana, tarde, noche o descanso, realiza cada profesional en cada jornada, de modo que se garantice la cobertura asistencial de la unidad y se respeten las reglas laborales, contractuales y organizativas aplicables (Burke et al., 2004). En la literatura se designa indistintamente como Nurse Scheduling Problem (NSP) o Nurse Rostering Problem (NRP); algunos autores reservan rostering para la construcción del calendario nominal e individualizado, una vez fijadas las necesidades de cobertura (Ernst et al., 2004). El NRP pertenece a la familia más amplia de la planificación de personal (personnel scheduling), con la que comparte estructura matemática, pero adquiere rasgos propios en el ámbito sanitario: operación ininterrumpida las veinticuatro horas, heterogeneidad de cualificaciones y elevada criticidad de una cobertura insuficiente (Van den Bergh et al., 2013).

Su estudio formal se remonta a Warner y Prawda (1972), quienes propusieron uno de los primeros modelos de programación matemática para asignar personal de enfermería en un hospital. Aquellos modelos pioneros sentaron las bases del campo, pero la capacidad de cálculo de la época obligaba a restringir el tamaño de las instancias y el número de reglas, produciendo soluciones demasiado simplificadas para un servicio real (Cheang et al., 2003). La investigación posterior respondió a esa limitación diversificándose hacia enfoques heurísticos, metaheurísticos y, más recientemente, híbridos, una evolución documentada en las revisiones de referencia del campo (Burke et al., 2004; Cheang et al., 2003; Ernst et al., 2004).

Una instancia del problema queda definida por cinco elementos que se formalizarán como datos del modelo en el Capítulo 4: un conjunto de enfermeras, caracterizadas por su cualificación y su tipo de contrato; un horizonte de planificación discretizado en jornadas; un conjunto de tipos de turno; los requisitos de cobertura, que fijan cuántas enfermeras,

y con qué cualificación, deben estar presentes en cada turno; y el conjunto de reglas que el calendario debe o debería satisfacer. La decisión central puede expresarse de forma intuitiva como «qué turno realiza cada enfermera en cada día». Para ordenar el proceso resulta útil la descomposición por etapas de Ernst et al. (2004), modelización de la demanda, programación de días libres, asignación de turnos, construcción de las líneas de trabajo individuales y asignación de tareas, dentro de la cual este trabajo se concentra en la asignación de turnos y la formación de las líneas de trabajo, tomando la demanda de cobertura como dato de entrada.

Conviene distinguir, además, entre planificación cíclica y no cíclica. La cíclica diseña un patrón fijo de turnos que rota periódicamente entre el personal, lo que aporta previsibilidad pero escasa flexibilidad; la no cíclica construye un calendario nuevo en cada periodo, ajustado a la demanda y a las preferencias, a costa de una mayor complejidad de cálculo (Burke et al., 2004; Van den Bergh et al., 2013). Este trabajo adopta el enfoque no cíclico e individualizado, por ser el que mejor permite incorporar de forma explícita los criterios de bienestar analizados en el apartado 2.3. Para describir de forma sistemática instancias tan heterogéneas, De Causmaecker y Vanden Berghe (2011) proponen categorizarlas según tres dimensiones: las características del personal, los requisitos de cobertura y las restricciones de carácter temporal, marco que se retomará en la modelización.

Por último, el NRP es un problema NP-difícil (Cheang et al., 2003), característica analizada formalmente incluso en trabajos recientes sobre la complejidad de sus variantes (den Hartog et al., 2023). En la práctica, esto implica que el número de calendarios posibles crece de forma exponencial con el número de enfermeras, días y turnos, hasta superar cualquier posibilidad de exploración exhaustiva, y que no se conocen algoritmos que garanticen la solución óptima en tiempo polinómico para el caso general. Esta dificultad es la que justifica el recurso a las técnicas de optimización que se revisan más adelante.

2.3. Restricciones duras y blandas

La estructura del NRP se organiza en torno a una distinción fundamental entre dos tipos de restricciones, cuya separación condiciona tanto la formulación del modelo como la interpretación de sus resultados (Burke et al., 2004). Las restricciones duras (hard constraints) son de obligado cumplimiento: definen qué calendarios son admisibles y cuáles no, de modo que una solución que viole una sola de ellas se considera no factible y queda descartada. Las restricciones blandas (soft constraints), en cambio, expresan criterios de calidad o preferencias que se desea satisfacer en la medida de lo posible, pero cuyo incumplimiento no invalida la solución, sino que se penaliza. En consecuencia, las restricciones

duras delimitan el espacio de soluciones factibles y las blandas permiten ordenar esas soluciones según su calidad relativa.

Entre las restricciones duras más habituales figura:

1. La cobertura mínima de cada turno, que exige un número suficiente de enfermeras (y, cuando procede, con la cualificación adecuada) en cada turno de cada día para garantizar una atención segura.
2. La prohibición de asignar a una misma persona dos turnos incompatibles en la misma jornada.
3. El respeto de los descansos legales mínimos entre turnos consecutivos (por ejemplo, evitar una mañana inmediatamente después de una noche).
4. Los límites contractuales sobre el número total de horas o turnos en el periodo.
5. Los requisitos de cualificación que impiden asignar a una enfermera funciones para las que no está habilitada.

Este conjunto de reglas refleja obligaciones normativas y de seguridad asistencial que la organización no puede vulnerar.

Las restricciones blandas recogen, por su parte, los criterios que mejoran la calidad del cuadrante desde el punto de vista del personal y de la equidad. Pertenecen a esta categoría:

1. El reparto equitativo de los fines de semana y de los turnos indeseables.
2. La limitación del número de turnos consecutivos (en especial de noches).
3. El respeto de los días libres solicitados.
4. La satisfacción de preferencias individuales.
5. El equilibrio de la carga de trabajo entre los profesionales. (Brucker et al., 2010; De Causmaecker & Vanden Berghe, 2011).

La calidad de una solución se cuantifica mediante una función objetivo que suma de forma ponderada las penalizaciones asociadas a cada restricción blanda incumplida, de manera que resolver el problema equivale a buscar, dentro del espacio de soluciones factibles, aquella que minimiza la penalización total (Burke & Curtois, 2014). Los pesos asignados a cada penalización determinan la importancia relativa de cada criterio y, por tanto, el perfil de las soluciones que el modelo considerará preferibles.

Conviene subrayar que la frontera entre lo duro y lo blando, así como los pesos de las restricciones blandas, no son neutrales: reflejan decisiones organizativas y de política de recursos humanos. Una misma regla, como no encadenar más de cierto número de noches,

puede tratarse como dura en un hospital y como blanda en otro, según el margen que la dirección esté dispuesta a conceder frente a las preferencias del personal. Del mismo modo, elevar el peso de un criterio de bienestar lo aproxima en la práctica a una restricción dura, mientras que reducirlo lo vuelve casi irrelevante. Esta plasticidad es precisamente la palanca metodológica sobre la que se apoya el presente trabajo, que define dos funciones objetivo alternativas (Modelos A y B) ponderando de manera distinta los criterios de bienestar, según se detalla en el Capítulo 4.

Por último, esta arquitectura de restricciones duras y blandas no es un mero recurso expostivo, sino el esquema que adoptan de forma explícita los principales conjuntos de datos de referencia del campo, que codifican cada regla como dura o como blanda y asignan a las segundas un peso concreto (Brucker et al., 2010; Burke et al., 2004). Esta correspondencia entre la teoría y los formatos de los benchmarks facilita la reproducibilidad y la comparación de resultados, y conecta directamente con la descripción de los datos.

2.4. Indicadores de bienestar laboral

Si la planificación de turnos pretende incorporar el bienestar del personal como objetivo, resulta imprescindible precisar qué se entiende por bienestar y, sobre todo, cómo puede medirse a partir de un cuadrante. El bienestar laboral en enfermería es un constructo multidimensional que la literatura suele articular en torno a varios resultados interrelacionados: la satisfacción laboral, el agotamiento profesional (burnout), el equilibrio entre la vida laboral y la personal, el absentismo y la intención de abandono o, en su reverso, la retención del personal (Gerlach et al., 2025; O’Connell et al., 2024). Estos resultados no son independientes entre sí, ni ajenos a la organización: la evidencia muestra que una conciliación deficiente erosiona la satisfacción y se traduce en mayores tasas de rotación, de modo que el bienestar del personal y la sostenibilidad de la plantilla están estrechamente ligados.

El interés operativo de estos resultados reside en que dependen, en buena medida, de características del cuadrante que el planificador controla. La rotación de turnos y la acumulación de noches consecutivas se asocian de forma significativa con el agotamiento profesional, con un efecto especialmente acusado sobre la dimensión de agotamiento emocional, por lo que limitar el número de noches seguidas y garantizar tiempos de recuperación suficientes entre turnos figura entre las medidas organizativas recomendadas (Qtait et al., 2025). De forma complementaria, la previsibilidad del calendario, su publicación con suficiente antelación, la posibilidad de descansar dos jornadas consecutivas y la disponibilidad de fines de semana libres se identifican como factores que mejoran la

conciliación y la percepción de calidad del cuadrante (Gerlach et al., 2025). La autonomía del personal sobre su propio horario, materializada en sistemas de autoplanificación, apunta asimismo a mejoras en la satisfacción y en la reducción del absentismo, si bien condicionadas a la calidad de su implantación (O’Connell et al., 2024).

Una dimensión transversal y especialmente relevante es la equidad. La percepción de injusticia en el reparto de los turnos indeseables, en particular las noches y los fines de semana, constituye una de las principales fuentes de insatisfacción y de conflicto, y se asocia con mayores tasas de rotación (Gerlach et al., 2025). Por ello, un cuadrante de calidad no solo debe respetar las preferencias individuales, sino también distribuir de manera equilibrada las cargas desagradables y el volumen total de trabajo entre los profesionales, evitando que recaigan sistemáticamente sobre las mismas personas (Van den Bergh et al., 2013). La equidad introduce así un criterio de justicia distributiva que va más allá de la mera satisfacción agregada de preferencias. El valor de estos hallazgos para el presente trabajo radica en que todos ellos pueden traducirse en indicadores cuantificables sobre un cuadrante concreto, susceptibles de ser penalizados en la función objetivo o medidos sobre las soluciones obtenidas. Entre los más directos cabe señalar: el número y la longitud de las secuencias de noches consecutivas; el número de descansos insuficientes entre turnos; el número de fines de semana libres de cada enfermera y su dispersión entre el personal; la tasa de preferencias y de días libres solicitados que se satisfacen; y el grado de equilibrio de la carga total de trabajo, o de los turnos indeseables, medido a través de su dispersión entre los profesionales. Cada uno de estos indicadores captura una faceta distinta del bienestar y puede incorporarse al modelo de optimización con un peso determinado, lo que permite orientar la búsqueda hacia soluciones más respetuosas con el personal.

En conjunto, estos indicadores constituyen el fundamento del Modelo B, orientado al bienestar, que se formula en el Capítulo 4, y proporcionan además las métricas con las que se compararán de forma cuantitativa los Modelos A y B en el Capítulo 6. Conviene matizar, no obstante, que la mejora de estos indicadores en el papel no garantiza por sí sola un efecto positivo sobre el bienestar real: la evidencia subraya que los beneficios de cualquier sistema de planificación dependen de su aceptación por el personal y de un diseño que preserve el juicio humano y el componente relacional que los criterios automáticos no capturan (Gerlach et al., 2025; O’Connell et al., 2024). Esta cautela se retomará en la discusión del trabajo.

2.5. Metodologías de optimización

La naturaleza NP-difícil del problema, unida a la enorme variedad de variantes que se encuentran en la práctica, explica la abundancia de técnicas de resolución desarrolladas a lo largo de las últimas décadas. Las revisiones de referencia coinciden en agrupar estas técnicas en unas pocas familias metodológicas, cuya elección depende del tamaño de las instancias, de la necesidad de garantías de optimalidad y de los recursos de cálculo disponibles (Abdalkareem et al., 2021; Burke et al., 2004; Ernst et al., 2004; Ngoo et al., 2022; Van den Bergh et al., 2013).

Métodos exactos

La primera familia la constituyen los métodos exactos, basados en programación matemática. El problema se formula habitualmente como un programa lineal entero (mixto), a menudo mediante modelos de tipo recubrimiento o partición de conjuntos en los que cada variable representa un calendario completo o una secuencia de turnos admisible para una enfermera (Jaumard et al., 1998). Para instancias de mayor tamaño se recurre a técnicas más sofisticadas como la generación de columnas y el branch-and-price, que evitan enumerar explícitamente el inmenso número de variables y las generan solo cuando resultan prometedoras (Jaumard et al., 1998). La programación con restricciones constituye otro enfoque exacto especialmente adecuado para problemas de planificación, por su capacidad para modelar de forma natural restricciones lógicas y de secuencia, y puede combinarse con la generación de columnas (He & Qu, 2012). La ventaja de estos métodos es que ofrecen garantía de optimalidad o, al menos, cotas que acotan la distancia a la solución óptima; su limitación es la escalabilidad, ya que el tiempo de cálculo puede crecer de forma prohibitiva en instancias grandes (Burke et al., 2004).

Métodos heurísticos y metaheurísticas

La segunda familia agrupa las heurísticas y metaheurísticas. Las heurísticas constructivas generan soluciones razonables con rapidez siguiendo reglas específicas del problema, mientras que las metaheurísticas, como la búsqueda tabú, el recocido simulado (simulated annealing), los algoritmos genéticos o la búsqueda en grandes vecindarios, exploran el espacio de soluciones de forma más sistemática para escapar de óptimos locales. Estos métodos renuncian a la garantía de optimalidad a cambio de obtener soluciones de buena calidad en tiempos razonables, incluso para instancias grandes, lo que los hace atractivos en contextos reales; como contrapartida, requieren un ajuste cuidadoso de sus parámetros y no informan de cuán lejos queda la solución del óptimo. Un ejemplo reciente es la resolución de un problema real de turnos de enfermería en hospitales del norte de Ita-

lia mediante recocido simulado, capaz de manejar numerosas restricciones y preferencias (Ceschia et al., 2023).

Métodos híbridos

La tercera familia, en auge en los últimos años, son los métodos híbridos y las matheurísticas, que combinan la potencia de los solvers de programación matemática con la flexibilidad de las metaheurísticas. Un ejemplo representativo es la integración de la programación con restricciones y la generación de columnas, que aprovecha la capacidad de modelado de la primera y la eficiencia de la segunda para resolver subproblemas (He & Qu, 2012). A esta línea se suma, de forma todavía incipiente, la incorporación de técnicas de aprendizaje automático e hiperheurísticas para guiar o acelerar la búsqueda, una tendencia que las revisiones más recientes identifican como una de las direcciones emergentes del campo (Ngoo et al., 2022).

Para el presente trabajo se ha optado por un enfoque exacto, basado en programación con restricciones o programación lineal entera e implementado con herramientas de software libre. Esta elección responde a la finalidad del trabajo, que no consiste en competir en escalabilidad con las mejores metaheurísticas, sino en comparar de forma rigurosa y reproducible dos formulaciones alternativas de la función objetivo (los Modelos A y B) sobre instancias de tamaño moderado. En ese contexto, la garantía de optimalidad, o la disponibilidad de cotas, resulta especialmente valiosa porque permite atribuir las diferencias observadas entre ambos modelos a la formulación del objetivo y no a la calidad variable de un procedimiento heurístico. Los solvers exactos de propósito general modernos permiten, además, modelar con relativa naturalidad las restricciones del problema.

2.6. Estado del arte reciente y revisiones sistemáticas

El problema de planificación de turnos de enfermería cuenta con una tradición de revisión bibliográfica consolidada. Los trabajos seminales de Cheang et al. (2003), Burke et al. (2004) y Ernst et al. (2004) sistematizaron por primera vez el campo, distinguiendo problemas, formulaciones y métodos, y siguen siendo referencia obligada. La revisión posterior de Van den Bergh et al. (2013) amplió la perspectiva al conjunto de la planificación de personal y propuso una clasificación multidimensional de la literatura que facilita localizar trabajos según el tipo de problema y las técnicas empleadas. Estas revisiones establecieron el vocabulario y los marcos conceptuales sobre los que se apoya la investigación actual.

Las revisiones más recientes permiten actualizar este panorama desde tres ángulos com-

plementarios. Abdalkareem et al. (2021) ofrecen una revisión amplia de la planificación sanitaria en clave de optimización, útil para encuadrar el problema general, sus formulaciones, conjuntos de datos y métodos, antes de aterrizar en el caso específico de la hospitalización. Ngoo et al. (2022), por su parte, se centran en las metodologías de resolución a partir de trabajos publicados entre 2012 y 2021, y las clasifican de forma sistemática en heurísticas, metaheurísticas, hiperheurísticas, optimización matemática, matheurísticas y enfoques híbridos; su survey concluye que las metaheurísticas son la opción más empleada, que las matheurísticas constituyen la tendencia emergente desde 2018 y que el conjunto de instancias de la primera competición internacional es el benchmark más utilizado por los investigadores. Finalmente, O'Connell et al. (2024) aportan la perspectiva del bienestar, al revisar el impacto de los sistemas electrónicos y de autoplanificación sobre los trabajadores y las organizaciones sanitarias.

Un elemento que ha impulsado de manera decisiva el avance del campo ha sido la estandarización de instancias y formatos a través de benchmarks públicos. La Primera Competición Internacional de Nurse Rostering (INRC-I), celebrada en 2010, popularizó un conjunto de instancias y una formulación que se han convertido en referencia comparativa (Haspeslagh et al., 2014); a ella se sumaron las instancias de Brucker et al. (2010) y, posteriormente, las nuevas instancias propuestas por Burke y Curtois (2014). Estos repositorios, junto con la segunda competición (INRC-II) y conjuntos de datos procedentes de hospitales reales (Ceschia et al., 2023), conforman el ecosistema de datos sobre el que se construirá la parte empírica de este trabajo. Su existencia ha sido clave para la reproducibilidad y para la comparación rigurosa de métodos.

Diferencia de enfoque del trabajo

La actividad investigadora reciente confirma que el problema sigue plenamente vigente. Trabajos aplicados de 2024 muestran cómo modelos de programación entera permiten construir cuadrantes que equilibran la distribución de la carga de trabajo y acomodan las preferencias individuales, apoyando simultáneamente el bienestar del personal y la calidad asistencial en entornos clínicos concretos (Mystakidis et al., 2024). No obstante, del conjunto de la literatura se desprende una observación que motiva directamente este Trabajo de Fin de Grado: la mayoría de los estudios ha orientado la función objetivo hacia la eficiencia, esto es, hacia la minimización del coste o de las violaciones de cobertura, mientras que la incorporación explícita del bienestar laboral como objetivo de optimización, y la cuantificación del trade-off que ello implica frente al enfoque tradicional, han recibido comparativamente menos atención sistemática (Gerlach et al., 2025; Ngoo et al., 2022; O'Connell et al., 2024).

Precisamente sobre ese hueco se articula la contribución del presente trabajo. En lugar de proponer un nuevo método de resolución que compita en escalabilidad, se formulan y resuelven dos modelos sobre instancias estándar, uno de referencia orientado a la cobertura y otro orientado al bienestar, y se analiza de forma cuantitativa qué se gana en términos de bienestar del personal y a qué coste, en el contexto de una unidad de hospitalización. De este modo, el trabajo enlaza la tradición metodológica del campo con la creciente preocupación por las condiciones laborales del personal de enfermería que las revisiones recientes ponen de relieve.

Capítulo 3

Dataset y preparación de datos

3.1. Descripción del benchmark principal (Curtois & Qu)

El conjunto de instancias seleccionado como base experimental de este trabajo es el publicado por Burke y Curtois (2014), conocido en la literatura como las instancias de Curtois & Qu y disponible públicamente en el repositorio schedulingbenchmarks.org. Su elección responde a tres criterios. El nombre con el que circulan en la literatura, instancias de Curtois y Qu, no debe inducir a confusión sobre su autoría: las instancias fueron introducidas formalmente por Burke y Curtois (2014), y la etiqueta alude a que el repositorio que las difunde lo mantiene Tim Curtois y a que buena parte de sus soluciones de referencia, tanto óptimas como mejores soluciones conocidas, se establecieron en trabajos posteriores firmados junto a Yi Qu (Strandmark et al., 2020).

En primer lugar, es un conjunto abierto, ampliamente adoptado y reproducible, lo que facilita comparar los resultados con los de otros autores. En segundo lugar, codifica el problema mediante una estructura explícita de restricciones duras y blandas con pesos, que se ajusta de forma natural a la formulación de los dos modelos planteados. Y en tercer lugar, y de manera decisiva para un trabajo que persigue rigor empírico, la mayoría de sus instancias cuentan con la solución óptima demostrada, lo que permite verificar que el modelo construido resuelve correctamente el problema y no solo produce una solución plausible.

El conjunto principal está formado por veinticuatro instancias, numeradas de Instance1 a Instance24, cuya dificultad crece de forma progresiva. Sus dimensiones abarcan un rango muy amplio: el horizonte de planificación oscila entre 2 y 52 semanas, el número de enfermeras entre 8 y 150, y el número de tipos de turno entre 1 y 32. Esta gradación es especialmente útil para los objetivos del trabajo, porque permite empezar por instancias

pequeñas, resolubles a optimalidad en tiempos reducidos, y observar después cómo evoluciona el comportamiento del modelo a medida que aumenta el tamaño del problema. Cada instancia se distribuye en dos formatos equivalentes, texto plano y XML.

Un rasgo que distingue a este benchmark de otros es la disponibilidad de cotas y soluciones de referencia verificadas. De las veinticuatro instancias, veintitrés tienen su solución óptima demostrada, es decir, se ha probado matemáticamente que no existe ningún calendario mejor; únicamente la instancia 24, la de mayor tamaño, permanece sin demostración de optimalidad, con una mejor solución conocida superior a su cota inferior (Burke & Curtois, 2014; Strandmark et al., 2020). Esta información cumple en el presente trabajo una función de control de calidad: al resolver una instancia con el Modelo A, el valor obtenido debe coincidir con el óptimo publicado, lo que confirma que tanto la formulación como su implementación son correctas antes de introducir las modificaciones del Modelo B.

En cuanto a su contenido, cada instancia especifica el conjunto de enfermeras y sus contratos, el horizonte de jornadas, los tipos de turno, los requisitos de cobertura por turno y un conjunto de reglas que se traducen en restricciones duras (cobertura, secuencias prohibidas, descansos, límites contractuales) y en restricciones blandas penalizadas (preferencias de trabajo o descanso en días concretos, equilibrio de la carga, distribución de fines de semana, límites sobre turnos consecutivos). La calidad de una solución se mide mediante una función objetivo que suma de forma ponderada las penalizaciones de las restricciones blandas incumplidas, de modo que el valor de la mejor solución conocida representa la mínima penalización alcanzable.

Esta arquitectura encaja con el planteamiento de una unidad de hospitalización, en la que conviven exigencias de cobertura ineludibles con preferencias del personal cuya satisfacción mejora el bienestar sin ser obligatoria. La Tabla 3.1 resume las características de las veinticuatro instancias del conjunto principal, que servirán de referencia para la selección de la muestra experimental.

Cuadro 3.1: Características de las instancias del benchmark de Curtois & Qu

Instancia	Semanas	Enfermeras	Tipos de turno	Mejor sol. conocida
Instance1	2	8	1	607*
Instance2	2	14	2	828*
Instance3	2	20	3	1001*
Instance4	4	10	2	1716*
Instance5	4	16	2	1143*
Instance6	4	18	3	1950*
Instance7	4	20	3	1056*
Instance8	4	30	4	1300*
Instance9	4	36	4	439*
Instance10	4	40	5	4631*
Instance11	4	50	6	3443*
Instance12	4	60	10	4040*
Instance13	4	120	18	1348*
Instance14	6	32	4	1278*
Instance15	6	45	6	3828*
Instance16	8	20	3	3225*
Instance17	8	32	4	5746*
Instance18	12	22	3	4459*
Instance19	12	40	5	3149*
Instance20	26	50	6	4769*
Instance21	26	100	8	21133*
Instance22	52	50	10	30241*
Instance23	52	100	16	17285*
Instance24	52	150	32	40680

Nota. El asterisco (*) indica solución demostrada óptima. La instancia 24 muestra la mejor solución conocida, no demostrada óptima (cota inferior: 33724).

Fuente: Elaboración propia a partir de Burke y Curtois (2014) y del repositorio *schedulingbenchmarks.org*.

3.2. Otros benchmarks de referencia

Además del conjunto principal descrito en el apartado anterior, este trabajo toma en consideración otros conjuntos de instancias que cumplen una función de referencia y contraste. No se emplean como base de los experimentos, sino que sirven para situar el benchmark principal dentro del panorama del campo, para mostrar formulaciones alternativas del problema y, en el caso de las instancias con competencias, para ilustrar variantes que el conjunto numerado no recoge. Su inclusión refuerza la validez externa del estudio y aporta perspectiva sobre hasta qué punto los resultados serían trasladables a formulaciones más complejas o más cercanas a la práctica hospitalaria real.

Burke & Curtois

En primer lugar, el propio repositorio de Curtois reúne, junto a las veinticuatro instancias numeradas, una colección de instancias clásicas con nombre propio procedentes de la literatura, como GPost, ORTEC, QMC, Ikegami, BCV, Valouxis o Musa, entre otras (Burke & Curtois, 2014). Estas instancias presentan tamaños y orígenes diversos y, a diferencia del conjunto numerado, algunas incorporan cobertura por nivel de competencia, es decir, exigen no solo un número mínimo de enfermeras por turno, sino también que cuenten con una cualificación determinada. El subconjunto BCV, por ejemplo, procede de los trabajos sobre rostering en hospitales belgas y fue introducido como banco de pruebas en la literatura previa (Brucker et al., 2010). Disponer de estas instancias permite, llegado el caso, ampliar la variedad de condiciones de prueba más allá del conjunto principal.

INRC-I

En segundo lugar, conviene situar el benchmark principal en relación con las competiciones internacionales de nurse rostering, que han sido un motor de estandarización en el campo. La Primera Competición Internacional de Nurse Rostering (INRC-I), celebrada en 2010, fijó una formulación de una sola etapa y un conjunto de instancias que se convirtieron en referencia comparativa para numerosos métodos posteriores (Haspeslagh et al., 2014).

INRC-II

En tercer lugar, la Segunda Competición Internacional (INRC-II) introdujo una novedad relevante: una formulación multietapa (Ceschia et al., 2015, 2019). En lugar de resolver todo el horizonte de planificación de una sola vez, el planificador resuelve semanas consecutivas de forma secuencial, y cada una utiliza el histórico de la anterior como punto de partida. Este planteamiento modela una situación más realista, en la que el futuro no se conoce por completo en el momento de decidir, pero a cambio complica considerablemente la comparación limpia entre dos funciones objetivo, que es el núcleo metodológico de este trabajo. El conjunto INRC-II abarca instancias de 30 a 120 enfermeras y horizontes de 4 u 8 semanas, y mantiene un sitio web propio con la infraestructura de validación. Por su prestigio y su carácter de referencia obligada, se cita y se describe aquí, aunque, por el motivo expuesto, no se emplea como benchmark principal. Por último, como referencia de realismo se considera el conjunto de instancias presentado por Ceschia et al. (2023), que recoge una formulación real surgida en instituciones sanitarias italianas y desarrollada en colaboración con una empresa de software del sector. Comprende 34 instancias e incorpora elementos propios de la práctica que los benchmarks puramente académicos no

siempre reflejan: enfermeras con distintas competencias, turnos especiales dependientes de esas competencias, límites de carga de trabajo, varios tipos de días libres y, de forma destacada, preferencias e incompatibilidades entre enfermeras. Este conjunto resulta útil para discutir, en el Capítulo 7, hasta qué punto las conclusiones obtenidas sobre el benchmark principal serían extrapolables a un entorno hospitalario real.

En conjunto, estos benchmarks no constituyen la materia de los experimentos, pero proporcionan el contexto necesario para interpretarlos: las instancias con nombre amplían la variedad de condiciones, INRC-I e INRC-II representan los estándares de competición frente a los que se mide el campo, y el conjunto italiano aporta la perspectiva de la aplicación real. La justificación detallada de qué instancias del benchmark principal se resuelven y por qué se aborda en el apartado 3.4.

3.3. Formato y preprocesamiento

Antes de poder resolver una instancia es necesario comprender cómo están almacenados los datos y transformarlos en estructuras que el modelo de optimización pueda utilizar. Este apartado describe el formato de los ficheros del benchmark principal, la información que contiene cada instancia y el proceso de carga y validación de los datos.

3.3.1. Formatos disponibles

Cada instancia se distribuye en dos formatos equivalentes: un fichero de texto plano y un fichero XML con extensión .ros. Ambos contienen exactamente la misma información, de modo que la elección entre uno y otro es una cuestión de comodidad técnica. En este trabajo se parte del formato de texto plano por su sencillez de lectura y por estar organizado en secciones claramente delimitadas, lo que facilita un preprocesamiento transparente y fácil de verificar.

Conviene fijar desde el principio dos convenciones del formato que tienen consecuencias prácticas. La primera es que todas las instancias comienzan en lunes. La segunda es que los días se numeran empezando por cero, de manera que el primer día del horizonte es el día 0. Estas dos reglas son importantes porque las restricciones relacionadas con los fines de semana dependen de identificar correctamente qué índices de día corresponden a sábados y domingos.

3.3.2. Estructura de una instancia: secciones y campos

El fichero de cada instancia se organiza en secciones, cada una de las cuales aporta una parte de la información necesaria para reconstruir el problema. Las principales son las siguientes:

- Sección de personal: enumera las enfermeras y, para cada una, sus límites individuales: el número máximo de turnos de cada tipo que puede realizar, el tiempo total de trabajo máximo y mínimo expresado en minutos, el número máximo y mínimo de turnos consecutivos, el número mínimo de días libres consecutivos y el número máximo de fines de semana que puede trabajar.
- Sección de turnos: define cada tipo de turno, su duración en minutos y las secuencias prohibidas, es decir, qué turnos no pueden realizarse el día siguiente a otro determinado (por ejemplo, impedir una mañana tras una noche).
- Sección de días libres: indica los días concretos en que una enfermera no puede recibir ningún turno.
- Secciones de peticiones: recogen las solicitudes del personal de trabajar (shift on requests) o de no trabajar (shift off requests) un turno concreto en un día concreto, cada una con un peso asociado
- Sección de cobertura: especifica, para cada día y cada turno, cuántas enfermeras se requieren y con qué peso se penaliza tanto la falta como el exceso de personal.

Esta estructura traduce de forma directa la distinción entre restricciones duras y blandas. Son restricciones duras la asignación de un único turno por día, las secuencias prohibidas, los límites de minutos totales y de turnos consecutivos, los días libres obligatorios y el número máximo de fines de semana. Son restricciones blandas penalizadas las peticiones de trabajar o no trabajar y las desviaciones respecto a la cobertura requerida, que penalizan la infradotación según el déficit y la sobredotación según el exceso, en ambos casos multiplicados por su peso. El repositorio acompaña las instancias de una formulación de programación entera de referencia, que servirá de base para el modelo.

3.3.3. Carga y preprocesamiento

El preprocesamiento consiste en leer el fichero de texto, recorrer cada sección y volcar su contenido en estructuras de datos que el modelo pueda consultar de forma eficiente. En la práctica, el código lee el fichero línea a línea, detecta el comienzo de cada sección por su etiqueta y, dentro de ella, separa los campos de cada registro para almacenarlos en diccionarios y listas de Python. El resultado es un conjunto de objetos que representan,

por un lado, los conjuntos del problema (las enfermeras, los días y los tipos de turno) y, por otro, los parámetros asociados (los límites de cada enfermera, la duración de cada turno, la cobertura requerida por día y turno con sus pesos, y las peticiones con los suyos).

A modo ilustrativo, una vez procesada una instancia, la información de cobertura podría representarse mediante una estructura que asocia cada par (día, turno) con el número de enfermeras requeridas y los pesos de penalización, de la forma:

```
1 cobertura = {  
2   (0, "E"): {"requerido": 3, "peso_falta": 100, "peso_exceso":  
3     1},  
4   (0, "D"): {"requerido": 2, "peso_falta": 100, "peso_exceso":  
5     1},  
6   (0, "L"): {"requerido": 2, "peso_falta": 100, "peso_exceso": 1}  
}
```

De manera análoga, los límites de cada enfermera se guardan en otra estructura indexada por enfermera, y las peticiones en sendas listas. Esta organización permite que, al construir el modelo, cada restricción se exprese consultando directamente el parámetro correspondiente, sin tener que volver a interpretar el fichero original.

3.4. Selección y caracterización de la muestra experimental

No todas las instancias del benchmark principal son igual de adecuadas para los objetivos de este trabajo. Este apartado establece los criterios con los que se selecciona el subconjunto sobre el que se ejecutan los experimentos, presenta y caracteriza la muestra elegida, y justifica por qué este diseño es suficiente y riguroso pese a partir de un número acotado de instancias.

3.4.1. Criterios de selección

La elección de las instancias responde a cuatro criterios:

1. Que el solver pueda alcanzar el óptimo conocido dentro de un presupuesto de tiempo acotado. Dado que el objetivo no es competir en escalabilidad con metaheurísticas, sino comparar con precisión dos formulaciones, interesa trabajar con instancias en las que la solución óptima esté al alcance del solver, de modo que las diferencias observadas entre el Modelo A y el Modelo B se deban a la formulación y no a la calidad variable de una búsqueda heurística.

2. La disponibilidad de óptimo conocido, que permite validar la implementación antes de experimentar.
3. La gradación de dificultad, no tanto en la longitud del horizonte como en el tamaño de la plantilla y en la variedad de turnos, para observar cómo evoluciona el comportamiento del modelo a medida que el problema gana complejidad.
4. La riqueza de restricciones blandas: para que el análisis de bienestar tenga sentido, las instancias deben incorporar varios tipos de turno, peticiones del personal, límites de fines de semana y de turnos consecutivos, ya que son precisamente esos elementos los que el Modelo B reordena.

3.4.2. Muestra seleccionada y su caracterización

Atendiendo a estos criterios, se selecciona como muestra experimental el conjunto de las tres primeras instancias del benchmark (Instance1, Instance2 e Instance3). Las tres comparten un horizonte de dos semanas y se diferencian en una progresión ordenada de complejidad: la plantilla crece de 8 a 14 y a 20 enfermeras, y el número de tipos de turno pasa de 1 a 2 y a 3, incorporando además secuencias prohibidas entre turnos en las dos últimas. Esta gradación recorre, en un rango computacionalmente manejable, las dimensiones que de verdad importan para el análisis de bienestar: a más tipos de turno y más personal, mayor es el margen sobre el que actúan los criterios de equidad, descanso y preferencias. Las tres instancias alcanzan el óptimo publicado dentro del presupuesto de tiempo, de forma demostrada en la más pequeña y en valor en las otras dos, lo que las hace idóneas tanto para validar el modelo como para servir de base fiable al estudio del compromiso.

La Tabla 3.2 caracteriza la muestra seleccionada. Junto a las dimensiones básicas se incluye una estimación del número de variables de decisión binarias, calculada como el producto del número de enfermeras, los días del horizonte y los tipos de turno, que ofrece una medida intuitiva del tamaño combinatorio de cada instancia.

Cuadro 3.2: Caracterización de la muestra experimental.

Instancia	Semanas	Días	Enfermeras	Tipos turno	Var. binarias	Óptimo
Instance1	2	14	8	1	112	607
Instance2	2	14	14	2	392	828
Instance3	2	14	20	3	840	1001

Nota: Días = semanas \times 7. Las variables binarias se estiman como enfermeras \times días \times tipos de turno y representan una cota aproximada del tamaño del problema. Los tres óptimos están demostrados en la literatura. Elaboración propia a partir de Burke y Curtois (2014).

3.4.3. Alcance experimental y justificación metodológica

La restricción de la muestra a tres instancias es una decisión deliberada y conviene justificarla con transparencia. Las instancias de cuatro semanas del benchmark (de la Instance4 en adelante) resultaron demasiado costosas de resolver hasta el óptimo, o muy cerca de él, dentro del presupuesto de tiempo disponible en el entorno de cálculo de este trabajo. Emplearlas habría introducido en el análisis un ruido difícil de controlar, porque las diferencias entre configuraciones quedarían contaminadas por la calidad variable de soluciones no óptimas. Limitar el estudio a las tres instancias de dos semanas, que sí se resuelven de forma fiable, garantiza que cada resultado de los experimentos sea atribuible a la formulación y no a un artefacto del solver. Es preferible un estudio más estrecho pero limpio que uno más amplio y contaminado.

Hecha esa aclaración, el número de instancias no debe confundirse con el tamaño de la base experimental. En optimización combinatoria, cada instancia no es un dato aislado, sino un problema completo cuyo espacio de soluciones crece de forma exponencial con sus dimensiones, como refleja la columna de variables binarias de la Tabla 3.2. Además, el alcance real de los experimentos es muy superior al número de instancias, porque cada una se resuelve bajo numerosas configuraciones. El Modelo A se ejecuta una vez por instancia como referencia, y el Modelo B se ejecuta variando los pesos de los seis criterios de bienestar, primero término a término en varios niveles de intensidad y después en escenarios que combinan todos los criterios. En conjunto, cada instancia se resuelve bajo una veintena de configuraciones distintas, de modo que el estudio se sustenta en del orden de sesenta experimentos, no en un puñado de cifras sueltas.

En síntesis, la robustez del estudio no descansa en la cantidad de instancias, sino en tres pilares: la resolución de cada instancia hasta su óptimo conocido, que elimina la incertidumbre sobre la calidad de las soluciones; la validación contra óptimos publicados, que garantiza la corrección de la implementación; y el barrido de cada instancia por múltiples configuraciones de pesos, que aporta densidad analítica al estudio del compromiso entre eficiencia y bienestar.

Capítulo 4

Modelización del problema

4.1. Conjuntos, parámetros y variables de decisión

Para formular el problema como un modelo de optimización es necesario fijar previamente la notación. Este apartado define los conjuntos sobre los que se construye el modelo, los parámetros que provienen de los datos descritos en el Capítulo 3 y las variables de decisión que el solver determinará. La formulación toma como referencia el modelo de programación entera publicado por Burke y Curtois (2014).

4.1.1. Conjuntos

- N : conjunto de enfermeras, indexado por n .
- $D = \{0, 1, \dots, |D| - 1\}$: conjunto de días del horizonte, indexado por d , con $|D| = 7 \cdot |\text{semanas}|$. Siguiendo el formato del benchmark, los días se numeran desde cero y el horizonte comienza en lunes.
- S : conjunto de tipos de turno, indexado por s . Se identifica además el turno más penoso de la instancia como $s^* \in S$, entendido como el turno de noche cuando la instancia lo incluye y, en su defecto, como el turno de tarde. Este turno alimenta el término de equidad del turno más penoso del Modelo B. En instancias con un único tipo de turno no existe un turno penoso diferenciado y s^* no llega a definirse.
- $W = \{0, 1, \dots, |\text{semanas}| - 1\}$: conjunto de fines de semana, indexado por w . Dado que el horizonte empieza en lunes, el sábado del fin de semana w es el día $7w + 5$ y el domingo, el día $7w + 6$.

4.1.2. Parámetros

Asociados a cada enfermera n :

- $\bar{q}_{n,s}$: número máximo de turnos de tipo s que puede realizar.
- \bar{m}_n y \underline{m}_n : número máximo y mínimo de minutos totales de trabajo.
- \bar{c}_n y \underline{c}_n : número máximo y mínimo de turnos consecutivos.
- \underline{p}_n : número mínimo de días libres consecutivos.
- \bar{f}_n : número máximo de fines de semana que puede trabajar.
- $O_n \subseteq D$: conjunto de días libres obligatorios.

Asociados a cada tipo de turno s :

- ℓ_s : duración del turno en minutos.
- $F \subseteq S \times S$: conjunto de pares de turnos prohibidos en días consecutivos, de modo que $(s, s') \in F$ significa que el turno s' no puede asignarse el día siguiente al turno s .

Asociados a la cobertura:

- $r_{d,s}$: número de enfermeras requeridas el día d en el turno s .
- $\alpha_{d,s}$ y $\beta_{d,s}$: pesos de penalización por infradotación y por sobredotación, respectivamente.

Asociados a las peticiones del personal:

- \mathcal{P}^{on} : conjunto de peticiones de trabajar, donde cada elemento (n, d, s) lleva asociado un peso $p_{n,d,s}^{\text{on}}$.
- \mathcal{P}^{off} : conjunto de peticiones de no trabajar, donde cada elemento (n, d, s) lleva asociado un peso $p_{n,d,s}^{\text{off}}$.

4.1.3. Variables de decisión

La variable principal del modelo es binaria:

$$x_{n,d,s} = \begin{cases} 1 & \text{si la enfermera } n \text{ trabaja el turno } s \text{ el día } d, \\ 0 & \text{en caso contrario.} \end{cases}$$

Además, se definen las siguientes variables auxiliares, comunes a varias restricciones:

- $y_{n,w} \in \{0, 1\}$: vale 1 si la enfermera n trabaja el fin de semana w , entendiéndose

que lo trabaja si realiza algún turno el sábado o el domingo de ese fin de semana. Esta variable interviene tanto en la restricción dura de número máximo de fines de semana como en los términos de bienestar relacionados con los fines de semana.

- $u_{d,s} \geq 0$ y $e_{d,s} \geq 0$: déficit y exceso de cobertura, respectivamente, el día d en el turno s . Miden cuántas enfermeras faltan o sobran respecto al requisito $r_{d,s}$ y alimentan la penalización de cobertura.

Cada término de bienestar del Modelo B introduce, además, sus propias variables auxiliares (por ejemplo, para medir la dispersión de la carga o las violaciones de fin de semana completo). Para no anticipar conceptos, esas variables se definen al formular cada término.

4.2. Restricciones duras

Las restricciones duras delimitan el conjunto de calendarios admisibles: toda solución debe cumplirlas sin excepción. A continuación se formula cada una sobre las variables y parámetros definidos en el apartado 4.1. Para abreviar, se denota por $w_{n,d} = \sum_{s \in S} x_{n,d,s}$ el indicador (de valor 0 o 1) de si la enfermera n trabaja algún turno el día d ; este valor está acotado entre 0 y 1 gracias a la primera restricción. La formulación sigue el modelo de programación entera de Burke y Curtois (2014).

1. Un turno como máximo por día.

Cada enfermera puede tener asignado, a lo sumo, un turno en cada jornada:

$$\sum_{s \in S} x_{n,d,s} \leq 1 \quad \forall n \in N, \forall d \in D. \quad (4.1)$$

2. Secuencias de turnos prohibidas.

Para cada par de turnos $(s, s') \in F$ que no puede encadenarse, se impide que el primero se asigne un día y el segundo al día siguiente:

$$x_{n,d,s} + x_{n,d+1,s'} \leq 1 \quad \forall n \in N, \forall d \in \{0, \dots, |D| - 2\}, \forall (s, s') \in F. \quad (4.2)$$

3. Máximo de turnos de cada tipo.

Cada enfermera no puede superar su límite de turnos de un tipo dado a lo largo del horizonte:

$$\sum_{d \in D} x_{n,d,s} \leq \bar{q}_{n,s} \quad \forall n \in N, \forall s \in S. \quad (4.3)$$

4. Máximo de minutos totales.

El tiempo total trabajado, sumando la duración de cada turno asignado, no puede exceder el máximo de la enfermera:

$$\sum_{d \in D} \sum_{s \in S} \ell_s x_{n,d,s} \leq \bar{m}_n \quad \forall n \in N. \quad (4.4)$$

5. Mínimo de minutos totales.

De forma simétrica, el tiempo total trabajado debe alcanzar al menos el mínimo exigido:

$$\sum_{d \in D} \sum_{s \in S} \ell_s x_{n,d,s} \geq \underline{m}_n \quad \forall n \in N. \quad (4.5)$$

6. Máximo de turnos consecutivos.

En cualquier ventana de $\bar{c}_n + 1$ días consecutivos, la enfermera no puede trabajar todos ellos, lo que limita la racha máxima a \bar{c}_n :

$$\sum_{j=0}^{\bar{c}_n} w_{n,d+j} \leq \bar{c}_n \quad \forall n \in N, \forall d \in \{0, \dots, |D| - \bar{c}_n - 1\}. \quad (4.6)$$

7. Mínimo de turnos consecutivos.

Se prohíbe que un bloque de trabajo dure menos de \underline{c}_n días. Para ello se impide el patrón consistente en un día libre, seguido de k días de trabajo (con $1 \leq k < \underline{c}_n$) y de otro día libre:

$$(1 - w_{n,d}) + \sum_{j=1}^k w_{n,d+j} + (1 - w_{n,d+k+1}) \leq k + 1, \quad (4.7)$$

para toda enfermera, todo $k \in \{1, \dots, \underline{c}_n - 1\}$ y toda posición interior válida.

8. Mínimo de días libres consecutivos.

De manera análoga, se prohíbe que un bloque de descanso dure menos de \underline{p}_n días, impidiendo el patrón de un día de trabajo, seguido de k días libres (con $1 \leq k < \underline{p}_n$) y de otro día de trabajo:

$$w_{n,d} + \sum_{j=1}^k (1 - w_{n,d+j}) + w_{n,d+k+1} \leq k + 1, \quad (4.8)$$

para toda enfermera, todo $k \in \{1, \dots, \underline{p}_n - 1\}$ y toda posición interior válida.

9. Máximo de fines de semana.

Primero se vincula la variable auxiliar $y_{n,w}$ al hecho de trabajar el fin de semana w , que se considera trabajado si la enfermera realiza algún turno el sábado (día $7w + 5$) o el domingo (día $7w + 6$):

$$y_{n,w} \geq w_{n,7w+5}, \quad y_{n,w} \geq w_{n,7w+6}, \quad y_{n,w} \leq w_{n,7w+5} + w_{n,7w+6} \quad \forall n \in N, \forall w \in W. \quad (4.9)$$

A continuación se limita el número de fines de semana trabajados:

$$\sum_{w \in W} y_{n,w} \leq \bar{f}_n \quad \forall n \in N. \quad (4.10)$$

10. Días libres obligatorios.

En los días marcados como libres para una enfermera, no puede asignarse ningún turno:

$$\sum_{s \in S} x_{n,d,s} = 0 \quad \forall n \in N, \forall d \in O_n. \quad (4.11)$$

Las restricciones 7 y 8 emplean la convención del benchmark según la cual los extremos del horizonte se asumen prolongados (con trabajo o descanso infinitos, según el caso) fuera del periodo de planificación, de modo que los bloques truncados por el inicio o el final del horizonte no se penalizan por ser cortos. Junto a estas restricciones, el carácter binario de $x_{n,d,s}$ e $y_{n,w}$, declarado al definir las variables, completa la definición del espacio de soluciones factibles.

4.3. Restricciones blandas y variables de penalización

A diferencia de las duras, las restricciones blandas no excluyen soluciones: su incumplimiento se permite, pero se penaliza. Cada una contribuye con un término de penalización que se acumulará en la función objetivo del Modelo A. Conviene destacar una característica relevante de este benchmark: la cobertura es una restricción blanda, no dura. Esto significa que el modelo puede dejar un turno con menos personal del requerido y, simplemente, pagar la penalización correspondiente, lo que garantiza que siempre exista una solución factible aunque la demanda no pueda cubrirse por completo.

1. Peticiones de trabajar (*shift on requests*).

Para cada petición $(n, d, s) \in \mathcal{P}^{\text{on}}$, en la que la enfermera n solicita trabajar el turno s el día d , se incurre en una penalización de peso $p_{n,d,s}^{\text{on}}$ si dicho turno no se le asigna. Como la asignación viene dada directamente por la variable $x_{n,d,s}$, esta penalización no requiere variables auxiliares y se expresa como:

$$P^{\text{on}} = \sum_{(n,d,s) \in \mathcal{P}^{\text{on}}} p_{n,d,s}^{\text{on}} (1 - x_{n,d,s}). \quad (4.12)$$

El término $(1 - x_{n,d,s})$ vale 1 cuando la petición no se satisface (la enfermera no trabaja ese turno) y 0 cuando sí se satisface.

2. Peticiones de no trabajar (*shift off requests*).

De forma simétrica, para cada petición $(n, d, s) \in \mathcal{P}^{\text{off}}$, en la que la enfermera solicita no trabajar un turno, se incurre en una penalización de peso $p_{n,d,s}^{\text{off}}$ si ese turno sí se le asigna:

$$P^{\text{off}} = \sum_{(n,d,s) \in \mathcal{P}^{\text{off}}} p_{n,d,s}^{\text{off}} x_{n,d,s}. \quad (4.13)$$

Aquí la variable $x_{n,d,s}$ vale 1 precisamente cuando la petición se incumple.

3. Cobertura (*cover*).

Para cada día d y turno s , el requisito de cobertura es $r_{d,s}$ enfermeras. El número realmente asignado es $\sum_{n \in N} x_{n,d,s}$. La diferencia entre lo asignado y lo requerido se descompone en déficit $u_{d,s}$ y exceso $e_{d,s}$ mediante la siguiente restricción:

$$\sum_{n \in N} x_{n,d,s} - r_{d,s} = e_{d,s} - u_{d,s} \quad \forall d \in D, \forall s \in S, \quad (4.14)$$

con $u_{d,s} \geq 0$ y $e_{d,s} \geq 0$. Dado que el modelo minimiza, en la solución óptima solo una de las dos variables será no nula: $u_{d,s}$ recoge cuántas enfermeras faltan respecto al requisito y $e_{d,s}$, cuántas sobran. La penalización de cobertura pondera el déficit con el peso $\alpha_{d,s}$ y el exceso con el peso $\beta_{d,s}$:

$$P^{\text{cov}} = \sum_{d \in D} \sum_{s \in S} (\alpha_{d,s} u_{d,s} + \beta_{d,s} e_{d,s}). \quad (4.15)$$

En conjunto, las restricciones blandas del modelo dan lugar a tres términos de penalización, P^{on} , P^{off} y P^{cov} , que cuantifican respectivamente el incumplimiento de las peticiones de trabajar, el de las peticiones de no trabajar y las desviaciones de cobertura. Estos tres

términos constituyen la base de la función objetivo del Modelo A, que se formula en el apartado siguiente.

4.4. Función objetivo del Modelo A (baseline)

El Modelo A constituye la formulación de referencia o *baseline* del trabajo. Su función objetivo es exactamente la del benchmark: minimizar la suma de las tres penalizaciones blandas definidas en el apartado 4.3, esto es, el incumplimiento de las peticiones de trabajar, el de las peticiones de no trabajar y las desviaciones de cobertura. Formalmente:

$$\text{mín } Z_A = P^{\text{on}} + P^{\text{off}} + P^{\text{cov}}, \quad (4.16)$$

o, de forma desarrollada:

$$\text{mín } Z_A = \sum_{(n,d,s) \in \mathcal{P}^{\text{on}}} p_{n,d,s}^{\text{on}} (1 - x_{n,d,s}) + \sum_{(n,d,s) \in \mathcal{P}^{\text{off}}} p_{n,d,s}^{\text{off}} x_{n,d,s} + \sum_{d \in D} \sum_{s \in S} (\alpha_{d,s} u_{d,s} + \beta_{d,s} e_{d,s}), \quad (4.17)$$

sujeto a todas las restricciones duras del apartado 4.2 y a la restricción que define las desviaciones de cobertura del apartado 4.3.

Es importante subrayar que los pesos de este objetivo (p^{on} , p^{off} , α y β) no son elegidos por el investigador, sino que vienen fijados en los datos de cada instancia. Esta es precisamente la razón por la que el Modelo A permite validar la implementación: al resolverlo, el valor óptimo de Z_A debe coincidir con la mejor solución conocida recogida en la Tabla 3.1. Si la coincidencia se produce, queda confirmado que tanto la carga de datos como la formulación de variables y restricciones son correctas.

El Modelo A representa, por tanto, el enfoque tradicional de la planificación de turnos: se persigue cubrir la demanda y atender las peticiones del personal al menor coste de penalización, sin incorporar de forma explícita ningún criterio de bienestar más allá de los ya contenidos en las peticiones. Constituye así el punto de comparación frente al cual se medirá la aportación del Modelo B, que se desarrolla a continuación.

4.5. Función objetivo del Modelo B (bienestar)

El Modelo B amplía el Modelo A incorporando de forma explícita criterios de bienestar laboral. Mantiene intactas todas las restricciones duras y todos los términos del objetivo del Modelo A, y les añade seis términos de bienestar, cada uno controlado por un peso $\lambda_k \geq 0$. La elección de estos pesos es la palanca experimental del trabajo: variándolos se

traza la curva de compromiso entre eficiencia y bienestar que se analiza en el Capítulo 6. Como se verá al final del apartado, cuando todos los pesos de bienestar valen cero, el Modelo B se reduce exactamente al Modelo A.

4.5.1. Los seis términos de bienestar

1. Equidad de carga total.

Un reparto desigual del trabajo es una fuente reconocida de insatisfacción. Para medirlo, se define la carga de cada enfermera como el número total de turnos asignados, $T_n = \sum_{d \in D} \sum_{s \in S} x_{n,d,s}$, y se penaliza la diferencia entre la enfermera más cargada y la menos cargada. Con las variables auxiliares $T^{\text{máx}}$ y $T^{\text{mín}}$ y las restricciones

$$T^{\text{máx}} \geq T_n \quad \forall n \in N, \quad T^{\text{mín}} \leq T_n \quad \forall n \in N, \quad (4.18)$$

el término se define como:

$$B_1 = T^{\text{máx}} - T^{\text{mín}}. \quad (4.19)$$

2. Equidad de fines de semana.

Más allá de la carga total, el reparto de los fines de semana trabajados afecta directamente a la conciliación (O'Connell et al., 2024). Definida la cantidad de fines de semana de cada enfermera como $F_n = \sum_{w \in W} y_{n,w}$, y con las auxiliares $F^{\text{máx}}$ y $F^{\text{mín}}$ tales que

$$F^{\text{máx}} \geq F_n \quad \forall n \in N, \quad F^{\text{mín}} \leq F_n \quad \forall n \in N, \quad (4.20)$$

el término es:

$$B_2 = F^{\text{máx}} - F^{\text{mín}}. \quad (4.21)$$

3. Suavizado de secuencias largas.

El límite legal de turnos consecutivos no equivale a un límite cómodo: las rachas largas, aun siendo legales, aumentan la fatiga (Qtait et al., 2025). Se introduce un umbral de confort θ y se penaliza cada ventana de $\theta + 1$ días trabajados de forma consecutiva. Con la variable auxiliar $g_{n,d} \geq 0$ definida por

$$g_{n,d} \geq \sum_{j=0}^{\theta} w_{n,d+j} - \theta \quad \forall n \in N, \forall d \in \{0, \dots, |D| - \theta - 1\}, \quad (4.22)$$

el término acumula esas violaciones:

$$B_3 = \sum_{n \in N} \sum_d g_{n,d}. \quad (4.23)$$

La variable $g_{n,d}$ toma el valor 1 cuando los $\theta + 1$ días de la ventana se trabajan todos, y 0 en caso contrario.

4. Satisfacción reforzada de preferencias.

Las peticiones del personal ya figuran en el Modelo A con su peso original, pero el bienestar aconseja darles mayor relevancia (O'Connell et al., 2024). Este término reutiliza las penalizaciones de peticiones del apartado 4.3 para amplificar su importancia relativa frente a los criterios de eficiencia:

$$B_4 = P^{\text{on}} + P^{\text{off}}. \quad (4.24)$$

Al ponderarlo con λ_4 , el peso efectivo de las preferencias pasa a ser su peso original incrementado en un factor $(1 + \lambda_4)$.

5. Equidad del turno más penoso.

Repartir la carga total no garantiza repartir los turnos más penosos, habitualmente las noches y, cuando no las hay, las tardes, que son los más asociados al agotamiento (Qtait et al., 2025). Cuando la instancia tiene más de un tipo de turno, se toma su turno más penoso s^* (la noche si existe; en su defecto, la tarde) y se define el número de turnos penosos de cada enfermera como $Nt_n = \sum_{d \in D} x_{n,d,s^*}$, penalizando su dispersión. Con las auxiliares $N^{\text{máx}}$ y $N^{\text{mín}}$ tales que

$$N^{\text{máx}} \geq Nt_n \quad \forall n \in N, \quad N^{\text{mín}} \leq Nt_n \quad \forall n \in N, \quad (4.25)$$

el término es:

$$B_5 = N^{\text{máx}} - N^{\text{mín}}. \quad (4.26)$$

En las instancias con un único tipo de turno, como Instance1, no hay turno penoso diferenciado y el término queda inactivo.

6. Fines de semana completos.

Trabajar un único día del fin de semana fragmenta el descanso; es preferible trabajar los dos días o ninguno (O'Connell et al., 2024). Para cada enfermera y fin de semana, con

sábado en el día $7w + 5$ y domingo en el día $7w + 6$, se define la variable auxiliar $v_{n,w} \geq 0$ mediante

$$v_{n,w} \geq w_{n,7w+5} - w_{n,7w+6}, \quad v_{n,w} \geq w_{n,7w+6} - w_{n,7w+5} \quad \forall n \in N, \forall w \in W, \quad (4.27)$$

y el término acumula los fines de semana incompletos:

$$B_6 = \sum_{n \in N} \sum_{w \in W} v_{n,w}. \quad (4.28)$$

La variable $v_{n,w}$ vale 1 cuando se trabaja exactamente uno de los dos días del fin de semana, y 0 cuando se trabajan ambos o ninguno.

4.5.2. Función objetivo combinada y relación con el Modelo A

La función objetivo del Modelo B suma al objetivo del Modelo A los seis términos de bienestar, cada uno ponderado por su peso:

$$\text{mín } Z_B = Z_A + \lambda_1 B_1 + \lambda_2 B_2 + \lambda_3 B_3 + \lambda_4 B_4 + \lambda_5 B_5 + \lambda_6 B_6, \quad (4.29)$$

sujeto a todas las restricciones duras del apartado 4.2, a la restricción de cobertura del apartado 4.3 y a las restricciones que definen las variables auxiliares de cada término ($T^{\text{máx}}, T^{\text{mín}}, F^{\text{máx}}, F^{\text{mín}}, g_{n,d}, N^{\text{máx}}, N^{\text{mín}}$ y $v_{n,w}$).

La relación con el Modelo A es directa y constituye una de las fortalezas del diseño: si $\lambda_1 = \lambda_2 = \dots = \lambda_6 = 0$, todos los términos de bienestar desaparecen del objetivo y Z_B coincide con Z_A . El Modelo A es, por tanto, un caso particular del Modelo B. Esto permite interpretar los experimentos como un recorrido continuo: partiendo del *baseline* (pesos nulos) e incrementando progresivamente los pesos de bienestar, se observa cómo mejora el bienestar del personal y a qué coste en términos del objetivo original.

Capítulo 5

Resolución y herramientas

5.1. Herramientas seleccionadas

La resolución de los modelos formulados en el Capítulo 4 se aborda con Python como lenguaje de programación y con la biblioteca OR-Tools como entorno de optimización, empleando en concreto su solver CP-SAT. Esta combinación es gratuita, de código abierto y ampliamente utilizada, lo que la hace especialmente adecuada para un trabajo que prioriza la reproducibilidad y la resolución exacta sobre instancias de tamaño moderado.

OR-Tools & CP-SAT

OR-Tools es la suite de optimización de código abierto desarrollada por el equipo de Investigación Operativa de Google, liberada públicamente en 2015 y en desarrollo desde 2008 (Perron & Furnon, 2024). Su principal virtud es que ofrece una interfaz uniforme a varios solvers (de programación lineal, programación lineal entera y programación con restricciones), de modo que el modelo se describe una sola vez y puede resolverse con el motor más apropiado. Para este trabajo se utiliza CP-SAT, el solver de programación con restricciones de la suite.

CP-SAT es un solver que combina varias tecnologías de resolución: está construido sobre un motor de satisfacibilidad booleana (SAT) con generación perezosa de cláusulas, integra un método símplex para la relajación lineal del problema e incorpora técnicas propias de la programación lineal entera, además de una búsqueda paralela basada en una cartera de estrategias diversas (Perron & Didier, 2024). Esta hibridación le permite ser muy eficaz en problemas combinatorios y, lo que es decisivo para este trabajo, demostrar la optimalidad de las soluciones que encuentra. Su solidez está avalada por su reconocimiento en competiciones internacionales de programación con restricciones, en las que ha obtenido

los primeros puestos de forma reiterada.

La elección de un enfoque de programación con restricciones, frente a una formulación puramente de programación lineal entera, responde a la naturaleza del problema. El NRP está repleto de restricciones de carácter lógico y secuencial (secuencias de turnos prohibidas, límites de turnos y de descansos consecutivos, fines de semana), que la programación con restricciones expresa de forma natural y compacta. Al mismo tiempo, dado que CP-SAT integra internamente técnicas de programación lineal entera, no se renuncia a la garantía de optimalidad ni a las cotas que esta proporciona. Se obtiene así lo mejor de ambos paradigmas: la expresividad de la programación con restricciones y el rigor de la optimización exacta.

Respecto a la opción de emplear solvers comerciales como Gurobi o CPLEX, se ha descartado deliberadamente. Aunque son herramientas de altísimo rendimiento, requieren licencia y, en el rango de tamaños de la muestra experimental, no aportan una ventaja que justifique esa barrera. El uso de una herramienta gratuita refuerza, además, la reproducibilidad del trabajo: cualquier persona puede replicar los experimentos sin coste ni licencias, lo que es una virtud metodológica en investigación.

El resto del entorno de software se apoya también en Python. El preprocesamiento descrito en el Capítulo 3 se implementa como un módulo de carga que transforma los ficheros del benchmark en las estructuras de datos que consume el modelo. Para el tratamiento de los resultados y la elaboración de las figuras del Capítulo 6 se emplean librerías estándar del ecosistema científico de Python. El conjunto conforma una cadena reproducible que va desde la lectura de la instancia hasta la obtención y el análisis de las soluciones, coherente con el enfoque exacto y reproducible argumentado en el Capítulo 2.

5.2. Implementación del modelo

La traducción del modelo matemático del Capítulo 4 a código con OR-Tools es prácticamente directa: cada conjunto se convierte en un bucle, cada variable en una llamada al modelo y cada restricción en una línea que se añade al objeto del modelo. Este apartado muestra los fragmentos esenciales de esa traducción; el código completo se recoge en el Anexo B. Los comentarios del código se escriben sin acentos por compatibilidad con el formato de listados.

El primer paso es crear el objeto del modelo y declarar la variable de decisión principal. La variable $x_{n,d,s}$ se representa como una variable booleana, almacenada en un diccionario indexado por la terna (enfermera, día, turno):

```
1 from ortools.sat.python import cp_model
```

```

2
3 model = cp_model.CpModel()
4
5 # Variable principal: x[n,d,s] = 1 si la enfermera n trabaja el
   turno s el día d
6 x = {}
7 for n in N:
8     for d in D:
9         for s in S:
10            x[(n, d, s)] = model.NewBoolVar(f"x_{n}_{d}_{s}")

```

Listing 5.1: Creación del modelo y de la variable de decisión principal

Las restricciones duras se añaden con el método `model.Add`, traduciendo casi literalmente las desigualdades del apartado 4.2. Por ejemplo, la restricción de un turno como máximo por día se escribe sumando las variables de un mismo día y exigiendo que esa suma no supere uno:

```

1 # Restriccion dura: un turno como maximo por dia
2 for n in N:
3     for d in D:
4         model.Add(sum(x[(n, d, s)] for s in S) <= 1)

```

Listing 5.2: Restricción dura: un turno como máximo por día

Las restricciones blandas requieren variables auxiliares y aportan términos a la función objetivo. La cobertura es el caso más representativo: para cada día y turno se crean las variables de déficit y exceso, se vinculan a la diferencia entre lo asignado y lo requerido, y se acumula la penalización correspondiente en una lista de términos del objetivo:

```

1 obj_terms = []
2 u, e = {}, {}
3 for d in D:
4     for s in S:
5         u[(d, s)] = model.NewIntVar(0, len(N), f"u_{d}_{s}")
6         e[(d, s)] = model.NewIntVar(0, len(N), f"e_{d}_{s}")
7         model.Add(sum(x[(n, d, s)] for n in N) - req[(d, s)] == e[(
8             d, s)] - u[(d, s)])
9         obj_terms.append(alpha[(d, s)] * u[(d, s)] + beta[(d, s)] *
10            e[(d, s)])

```

Listing 5.3: Restricción blanda de cobertura y acumulación de penalizaciones

Los términos de bienestar del Modelo B se construyen de forma análoga. El de equidad de carga, por ejemplo, calcula la carga total de cada enfermera, introduce las variables auxiliares de máximo y mínimo y añade su diferencia al objetivo, ponderada por su peso:

```

1 # Término de bienestar B1: equidad de carga
2 T = {n: model.NewIntVar(0, len(D), f"T_{n}") for n in N}
3 for n in N:
4     model.Add(T[n] == sum(x[(n, d, s)] for d in D for s in S))
5
6 Tmax = model.NewIntVar(0, len(D), "Tmax")
7 Tmin = model.NewIntVar(0, len(D), "Tmin")
8 for n in N:
9     model.Add(Tmax >= T[n])
10    model.Add(Tmin <= T[n])
11
12 obj_terms.append(lambda1 * (Tmax - Tmin)) # lambda1 = 0 en el
    Modelo A

```

Listing 5.4: Término de bienestar: equidad de carga (B_1)

Este último fragmento ilustra una decisión de diseño importante: el Modelo A y el Modelo B comparten exactamente el mismo código. La única diferencia es el valor de los pesos de bienestar. Si un peso vale cero, su término se anula y desaparece del objetivo, reproduciendo el Modelo A; si es positivo, el término actúa, dando lugar al Modelo B. En la práctica, esto se implementa con una única función que recibe los datos de la instancia y el vector de pesos, y construye el modelo correspondiente, lo que evita duplicar código y garantiza que ambos modelos son idénticos salvo en los pesos.

Finalmente, se fija la función objetivo como la suma de todos los términos acumulados y se invoca al solver:

```

1 model.Minimize(sum(obj_terms))
2
3 solver = cp_model.CpSolver()
4 solver.parameters.max_time_in_seconds = 30
5 solver.parameters.num_search_workers = 8
6 solver.parameters.random_seed = 42
7 status = solver.Solve(model)
8
9 if status == cp_model.OPTIMAL:
10     valor = solver.ObjectiveValue()

```

Listing 5.5: Definición del objetivo y llamada al solver

La implementación completa organiza estos bloques en tres componentes: un módulo de carga, que transforma el fichero de la instancia en los conjuntos y parámetros (Capítulo 3); una función de construcción, que recibe esos datos y los pesos de bienestar y devuelve el modelo con todas sus variables y restricciones; y una función de resolución, que lanza el

solver, comprueba el estado y extrae la solución y sus métricas.

5.3. Configuración del solver

CP-SAT expone un conjunto de parámetros que permiten ajustar su comportamiento. Para este trabajo se configuran cuatro aspectos, orientados a tres objetivos: garantizar la resolución exacta cuando es posible, asegurar la reproducibilidad de los resultados y permitir la ejecución sistemática de los experimentos.

Límite de tiempo. Mediante el parámetro `max_time_in_seconds` se fija un límite de 30 segundos por ejecución. Con este presupuesto, la instancia más pequeña se resuelve a optimalidad demostrada, mientras que las de mayor tamaño alcanzan el valor óptimo conocido aunque sin llegar a certificar la optimalidad. Si una ejecución agota el límite sin demostrar optimalidad, el solver devuelve la mejor solución encontrada, circunstancia que se registra explícitamente.

Paralelismo. El parámetro `num_search_workers` se fija en 8, acorde con los núcleos de cálculo disponibles en el entorno de ejecución. CP-SAT aprovecha el paralelismo ejecutando simultáneamente una cartera de estrategias de búsqueda distintas, de modo que varias aproximaciones compiten por encontrar y demostrar la mejor solución, lo que acelera la resolución.

Semilla aleatoria. El parámetro `random_seed` se fija en un valor constante (42), de manera que las decisiones internas que dependen del azar sean siempre las mismas entre ejecuciones.

Reproducibilidad. Conviene ser preciso sobre el alcance de la reproducibilidad. Con esta configuración, el valor objetivo de la solución y el estado de optimalidad son reproducibles: al repetir un experimento se obtiene el mismo valor óptimo. En cambio, debido a la búsqueda en paralelo, pueden variar ligeramente el tiempo de cálculo y, cuando existen varias soluciones igual de buenas, el calendario concreto que se devuelve. Para los fines de este trabajo, lo relevante es la métrica que se analiza, el valor objetivo, que sí es reproducible. Una reproducibilidad bit a bit del calendario exigiría usar un único hilo de búsqueda, a costa de un mayor tiempo de cálculo; se ha priorizado la configuración paralela por su eficiencia, sin perjuicio de la reproducibilidad de los resultados analizados.

Detección de optimalidad y métricas registradas. Tras la resolución, el solver devuelve un estado que indica la naturaleza de la solución. Los estados relevantes son `OPTIMAL` (solución demostrada óptima), `FEASIBLE` (solución encontrada pero sin demostrar óptima, típicamente por agotar el tiempo) e `INFEASIBLE` (el problema no admite solución

factible, situación que en este modelo no puede darse porque la cobertura es blanda). De cada ejecución se registran cuatro métricas:

```
1 status = solver.Solve(model)
2
3 resultado = {
4     "estado": solver.StatusName(status),      # OPTIMAL, FEASIBLE,
5     ...,
6     "objetivo": solver.ObjectiveValue(),     # valor de la solución
7     "cota": solver.BestObjectiveBound(),     # mejor cota inferior
8     "tiempo_s": solver.WallTime(),          # tiempo de cálculo en
        segundos
}
```

Listing 5.6: Extracción de las métricas de cada ejecución

El estado permite saber si la solución es óptima; el valor objetivo es la cifra que se compara y analiza; la cota inferior, junto con el objetivo, permite calcular la distancia a la optimalidad (el *gap*) en los casos en que esta no se demuestre; y el tiempo de cálculo documenta el coste computacional. Para la validación del modelo, se exige que el Modelo A alcance el estado OPTIMAL y que su valor objetivo coincida con el óptimo publicado, condición que confirma la corrección de la implementación antes de proceder con los experimentos de bienestar.

5.4. Diseño experimental

Este apartado define el plan de experimentos: las instancias, el esquema de pesos del Modelo B, las métricas registradas y el protocolo de validación.

5.4.1. Muestra y obtención de las instancias

Los experimentos se realizan sobre las tres instancias seleccionadas (Instance1, Instance2 e Instance3). Las tres alcanzan su óptimo publicado dentro del presupuesto de tiempo, de forma demostrada en la primera y en valor en las otras dos. Cada instancia se obtiene del repositorio público del benchmark y se transforma, mediante el módulo de carga descrito en el Capítulo 3, en los conjuntos y parámetros que consume el modelo.

5.4.2. Esquema de pesos del Modelo B

Explorar todas las combinaciones de los seis pesos $\lambda_1, \dots, \lambda_6$ sería inabarcable, por lo que se adopta un diseño en tres bloques:

1. *Baseline*: el Modelo A (todos los pesos a cero), una ejecución por instancia. Proporciona la referencia y valida la implementación.
2. *Análisis término a término*: para cada término se activa solo ese término y se barren tres niveles de peso (bajo, medio y alto), manteniendo los demás a cero. Aísla el efecto de cada dimensión y traza su curva de compromiso.
3. *Escenarios combinados*: dos perfiles con todos los términos activos a la vez, uno moderado (pesos medios) y otro alto (pesos máximos), para observar el efecto agregado y los conflictos entre criterios.

En conjunto, el diseño supone tres ejecuciones del baseline, cincuenta y cuatro del análisis término a término (seis términos, tres niveles, tres instancias) y seis de los escenarios combinados (dos perfiles, tres instancias), esto es, del orden de sesenta y tres ejecuciones, muy por encima del número de instancias.

Cada ejecución del Modelo B arranca desde la solución del baseline de su instancia (técnica de arranque en caliente), de modo que la búsqueda solo se desplaza a mejorar el bienestar y las curvas resultan monótonas y reproducibles, a costa de reportar mejoras conservadoras.

Como los seis términos tienen escalas distintas, los niveles bajo, medio y alto de cada uno se calibran en relación con la escala del objetivo base, tras una breve ejecución piloto; los valores resultantes se documentan en el Anexo C.

5.4.3. Métricas registradas

De cada ejecución se registran tres clases de información: de resolución (estado del solver, tiempo de cálculo y, cuando no se demuestra optimalidad, la distancia a la cota inferior); de eficiencia (el valor del objetivo base Z_A , que permite medir el coste de cada configuración como incremento de Z_A respecto al baseline); y de bienestar (cada indicador en sus unidades naturales: diferencia de carga, diferencia de fines de semana trabajados, número de rachas largas, número de fines de semana incompletos, diferencia en el turno más penoso y tasa de satisfacción de preferencias). Registrar los indicadores en unidades naturales es lo que hace interpretable el análisis.

5.4.4. Protocolo de validación y reproducibilidad

Antes de cualquier experimento de bienestar se resuelve el Modelo A de cada instancia y se comprueba que alcanza el óptimo publicado; solo entonces se procede con el Modelo B. La reproducibilidad se garantiza con la semilla aleatoria fija, la versión exacta de OR-

Tools empleada (9.15.6755), la inclusión del código completo en el Anexo B, el uso de un benchmark público y el registro estructurado de todos los resultados en el Anexo D.

Capítulo 6

Experimentos y resultados

6.1. Validación del modelo y configuración de referencia

El primer paso del estudio experimental es comprobar que el Modelo A (la formulación base, sin términos de bienestar) reproduce el comportamiento esperado sobre instancias cuyo óptimo es conocido. Esta validación es imprescindible: si el modelo base no recupera los valores publicados, ningún resultado posterior sobre bienestar sería fiable.

La validación se apoya en la propiedad más útil de un benchmark académico: para cada instancia se conoce el valor óptimo de la función objetivo, verificado y publicado por la comunidad. El procedimiento consiste entonces en resolver el Modelo A (la formulación base, con las diez restricciones duras y la función objetivo de eficiencia, sin ningún término de bienestar) sobre cada instancia de la muestra y comprobar si el valor que obtiene nuestra implementación coincide con el valor publicado. Si coincide en las tres instancias, la probabilidad de que exista un error de modelización que no se haya manifestado es muy baja, porque un fallo en cualquiera de las restricciones alteraría el conjunto de soluciones factibles y, con casi total seguridad, el óptimo. La resolución se realizó con el solver OR-Tools CP-SAT, con un presupuesto de tiempo de 30 segundos por instancia.

Los resultados de la validación se recogen en la Tabla 6.1.

Cuadro 6.1: Validación del Modelo A frente a los óptimos publicados de Curtois y Qu.

Instancia	Enfermeras	Turnos	Z_A obtenido	Estado del solver	Óptimo public.	Coincide
1	8	1	607	Óptimo demostrado	607	Sí
2	14	2	828	Mejor solución hallada	828	Sí
3	20	3	1001	Mejor solución hallada	1001	Sí

Las tres instancias alcanzan exactamente su óptimo publicado, lo que confirma que el mo-

delo es correcto. Conviene, no obstante, detenerse en la diferencia entre las dos columnas de “estado”, porque encierra un matiz metodológico importante. En la Instancia 1, la más pequeña, el solver no solo encuentra el valor 607, sino que además demuestra que no existe ninguna solución mejor: termina con un certificado de optimalidad en menos de un segundo. En las Instancias 2 y 3, en cambio, el solver encuentra el valor óptimo (828 y 1001), pero dentro del presupuesto de tiempo no llega a emitir ese certificado: sabe que ha hallado una solución de coste 828, pero no ha terminado de descartar que pudiera existir una mejor. Esta distinción no afecta a la validación. El motivo es que el valor 828 ya está demostrado como óptimo en la literatura; que nuestro modelo alcance ese mismo valor confirma que admite exactamente la misma mejor solución. Demostrarlo de nuevo desde cero sería computacionalmente caro y redundante. Dicho de otro modo: para validar basta con que el modelo “toque” el óptimo conocido, no hace falta que lo vuelva a probar.

Una vez verificado el modelo base, su solución óptima adquiere un segundo papel, que será central en todo el resto del capítulo: se convierte en la configuración de referencia o baseline. Representa la planificación más eficiente posible bajo las reglas duras del problema, sin que ninguna consideración de bienestar haya entrado en juego. Frente a ese punto se medirá, dimensión a dimensión, cuánto bienestar es capaz de aportar el Modelo B y cuánta eficiencia cuesta cada aportación. Para que esa comparación sea posible, sobre la solución de referencia se calculan los seis indicadores de bienestar, expresados en sus unidades naturales (número de rachas, diferencia de turnos entre la persona más y la menos cargada, fracción de preferencias satisfechas, etc.). La Tabla 6.2 los reúne.

Cuadro 6.2: Indicadores de bienestar de la configuración de referencia (solución eficiente óptima).

Instancia	Rachas	Desig. carga	Desig. finde	Desig. penoso	Finde part.	Satisf. pref.
1	7	2	0	n.a.	2	0,808
2	15	5	0	8	1	0,710
3	25	5	0	5	0	0,984

Nota: “Desig.” indica la diferencia entre el valor máximo y mínimo entre el personal; n.a. indica que el término no aplica.

Esta tabla no es un resultado intermedio sin más: describe la “forma” que adopta una planificación cuando solo se persigue la eficiencia, y de ella se desprende un patrón muy informativo de cara al análisis posterior. La eficiencia pura tolera bastantes rachas largas de días consecutivos trabajados, entre 7 y 25 según el tamaño de la instancia, y el número crece con la plantilla. La explicación es intuitiva: concentrar muchos turnos seguidos en unas pocas personas es una forma barata de cubrir la demanda, porque evita los “huecos” que obligarían a activar a más personal. La desigualdad de carga también aparece (diferencias de hasta cinco turnos entre la persona más y la menos cargada) y la del turno penoso es notable en la Instancia 2 (una

diferencia de ocho turnos tardíos). En el otro extremo, la equidad de fin de semana ya está perfectamente resuelta de partida (desigualdad 0 en las tres instancias), y los fines de semana partidos son escasos. Esto último no es casualidad: la restricción dura que limita el número máximo de fines de semana trabajados ya impone, por sí sola, un reparto equilibrado de los fines de semana, de modo que el término de bienestar correspondiente apenas tendrá margen de actuación.

La lectura de conjunto es la que da sentido a todo lo que sigue. Las dimensiones en las que la planificación eficiente “se porta mal” (rachas largas, desigualdad de carga, equidad del turno penoso) son precisamente aquellas que las restricciones duras no controlan, y por tanto las que el Modelo B tendrá que corregir activamente. Las dimensiones en las que ya se porta bien (fines de semana) son las que ya están protegidas por las reglas duras. El análisis del Modelo B consistirá, en buena medida, en cuantificar cuánto cuesta corregir lo primero.

6.2. Resultados del Modelo B

Esta sección presenta los experimentos realizados con el Modelo B. Se describe primero cómo se calibraron los pesos de los términos de bienestar, después el análisis término a término que aísla el efecto de cada dimensión, y por último los escenarios que combinan todos los términos a la vez.

6.2.1. Calibración de los pesos

El Modelo B funciona sumando a la función de eficiencia un conjunto de penalizaciones de bienestar, cada una multiplicada por un peso. Su función objetivo tiene la forma

$$Z_B = Z_A + \sum_{k=1}^6 \lambda_k B_k,$$

donde cada término de bienestar B_k aparece multiplicado por un peso λ_k . Ese peso es la pieza clave de todo el experimento, porque actúa como un tipo de cambio: indica cuántas unidades de eficiencia está dispuesto el modelo a sacrificar a cambio de una unidad de mejora en esa dimensión de bienestar. Con un peso de cero, el término no influye y recuperamos el Modelo A. Con un peso muy alto, esa dimensión de bienestar pasa a mandar sobre todo lo demás, incluso por encima de la cobertura asistencial. La calibración consiste, precisamente, en encontrar pesos intermedios que hagan visible el compromiso sin destruir la cobertura.

El problema es que los seis términos no son comparables entre sí, porque miden cosas distintas en unidades distintas. Las rachas largas se cuentan por decenas (en la configuración de referencia llegan a 25), mientras que la desigualdad de carga o la del turno penoso son diferencias pequeñas, de entre cero y ocho turnos. Si se aplicara el mismo peso a todos, los términos de mayor recuento dominarían y los de diferencia pequeña serían invisibles. Por eso cada término necesita su propio rango de pesos, ajustado a su escala. Para fijarlos se realizaron ejecuciones exploratorias sobre la instancia más rica en turnos, observando dos umbrales para cada término: el peso a partir del cual el indicador empieza a moverse, y el peso a partir del cual empieza a comerse la eficiencia hasta comprometer la cobertura. Entre ambos umbrales se eligieron tres niveles representativos, bajo, medio y alto, que son los que se barren en el resto del capítulo. La Tabla 6.3 los recoge.

La razón de que los rangos difieran tanto se entiende mejor con una referencia: en estas instancias, dejar sin cubrir un solo puesto de la plantilla cuesta 100 puntos en la función de eficiencia. Por eso los términos

Cuadro 6.3: Niveles de peso por tipo de término de bienestar.

Tipo de término	Bajo	Medio	Alto
Desigualdad (carga, fin de semana, turno penoso)	10	50	200
Recuento (rachas largas, fines de semana partidos)	5	20	50
Preferencias	1	3	10

de desigualdad, que miden diferencias pequeñas, necesitan pesos del orden de decenas o centenas para que reducir esa diferencia en una unidad pueda competir con el coste de la cobertura. Los términos de recuento, en cambio, acumulan muchas unidades, así que un peso por unidad más modesto ya produce un efecto apreciable en conjunto. Y el término de preferencias se expresa directamente en las mismas unidades de penalización que la eficiencia, de modo que un multiplicador pequeño basta para reforzarlo. Cada experimento se resolvió con un presupuesto de 30 segundos, suficiente para que el solver encontrara soluciones de buena calidad en instancias de este tamaño.

Una última decisión metodológica resultó determinante para la limpieza de los resultados. En lugar de resolver cada configuración partiendo de cero, cada ejecución arranca desde la solución de referencia mediante la técnica de arranque en caliente, en la que el solver recibe la solución eficiente óptima como punto de partida sugerido. El motivo es práctico. Al resolver en frío con un presupuesto de tiempo acotado, las instancias más grandes producían resultados incoherentes en los indicadores secundarios, como configuraciones cuyo coste de eficiencia aumentaba sin que mejorara ninguna dimensión de bienestar. Partir de la solución eficiente reduce de forma notable esas incoherencias y orienta la búsqueda hacia la mejora, de modo que las curvas son, en su mayor parte, monótonas y reproducibles. No obstante, en las instancias de mayor tamaño, que no llegan a certificar la optimalidad dentro del presupuesto de tiempo, persiste cierto ruido del solver, que en algún caso puntual devuelve una solución ligeramente peor que la de partida en una dimensión secundaria. Por esa razón, el análisis término a término retiene siempre la mejor configuración de cada término, y las mejoras reportadas se interpretan como un suelo y no como un techo. Esta lectura conservadora es coherente con el propio arranque en caliente: como el solver permanece cerca de la solución de partida, el bienestar realmente alcanzable es al menos el documentado y probablemente mayor.

6.2.2. Análisis término a término

Para entender qué aporta cada dimensión de bienestar por separado, el primer bloque de experimentos activa un único término cada vez, dejando los otros cinco a cero. Esta estrategia aísla el efecto de cada criterio: permite ver, sin interferencias, hasta dónde se puede mejorar cada indicador y cuánta eficiencia cuesta esa mejora. Para cada término se barren sus tres niveles de peso, y de las soluciones obtenidas se retiene la que más mejora el indicador. La Tabla 6.4 resume el resultado, expresando cada celda como el paso desde el valor de la configuración de referencia hasta el mejor valor alcanzado, junto con el coste de eficiencia que conlleva.

El resultado más claro y robusto de todo el estudio es el de las rachas largas, que la Figura 6.1 representa como una frontera de compromiso. En las tres instancias, las rachas de días consecutivos trabajados se pueden reducir de forma sustancial, y el coste crece de manera ordenada conforme se exige más descanso. En la instancia más pequeña, bajar de siete a cuatro rachas cuesta apenas seis puntos de eficiencia, una mejora casi gratuita. En la instancia mediana, llevarlas de quince a cinco cuesta poco más de cien puntos. Y en la mayor, reducirlas de veinticinco a doce cuesta doscientos dieciocho. La pendiente de cada curva en la

Cuadro 6.4: Mejor mejora alcanzada por cada término de bienestar respecto a la configuración de referencia, con su coste de eficiencia.

Término	Instancia 1	Instancia 2	Instancia 3
Rachas largas	7 → 4 (coste +6)	15 → 5 (coste +106)	25 → 12 (coste +218)
Desigualdad de carga	2 → 0 (coste +109)	5 → 5 (sin mejora)	5 → 5 (sin mejora)
Desigualdad turno penoso	no aplica	8 → 7 (coste +95)	5 → 4 (coste +2)
Fines de semana partidos	2 → 0 (coste +1)	1 → 1 (sin mejora)	0 (ya óptimo)
Equidad de fin de semana	0 (ya óptimo)	0 (ya óptimo)	0 (ya óptimo)
Satisfacción preferencias	0,808 → 0,846	0,710 (ya máx.)	0,984 (ya máx.)

figura tiene una lectura directa para la gestión: indica lo cara que sale cada unidad adicional de descanso, y muestra que las primeras reducciones son siempre las más baratas, mientras que apurar hacia el mínimo se encarece progresivamente. Que el patrón se repita en tres instancias de tamaño y estructura distintos es lo que le da solidez: no es un artefacto de un caso concreto, sino un comportamiento general de la planificación hospitalaria.

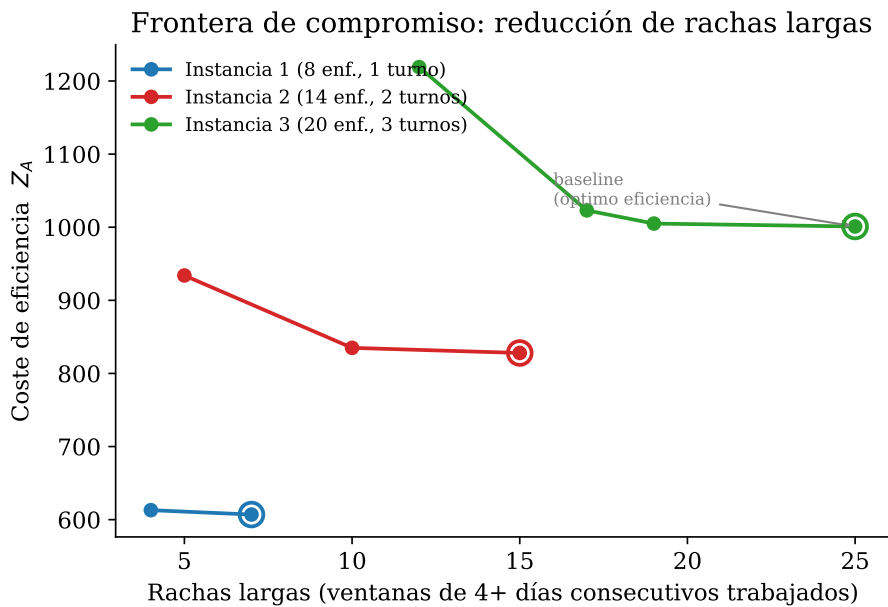


Figura 6.1: Frontera de compromiso entre eficiencia y rachas largas para las tres instancias.

Nota: Cada punto corresponde a una planificación distinta; los círculos huecos marcan la configuración de referencia.

El segundo resultado nítido son los fines de semana partidos, es decir, las situaciones en que una persona trabaja el sábado pero libra el domingo o viceversa. En la instancia pequeña se eliminan por completo, pasando de dos a cero, por apenas un punto de eficiencia. Es una mejora prácticamente gratuita, y tiene una explicación organizativa sencilla: reordenar quién trabaja cada mitad del fin de semana rara vez afecta a la cobertura total, de modo que el modelo puede ofrecer fines de semana completos sin apenas penalización.

Los términos de equidad presentan un comportamiento más rígido, que la Figura 6.2 visualiza bajo la eti-

queta de precio del bienestar. La desigualdad de carga se reduce hasta cero en la instancia pequeña, aunque a un coste apreciable de ciento nueve puntos, pero no se mueve del valor cinco en las dos instancias mayores dentro del presupuesto de tiempo. El análisis de cotas inferiores realizado sobre esos casos mostró que esa rigidez tiene dos componentes. Una parte es estructural: equilibrar la carga exigiría sacrificar bastante eficiencia, porque las peticiones, los días libres obligatorios y la demanda de cobertura encajan mejor con un reparto algo desigual. La otra parte es una limitación del tiempo de cálculo: encontrar la solución equilibrada en una instancia grande es costoso, y treinta segundos no siempre bastan para dar con ella. La desigualdad del turno penoso se comporta de forma parecida, con una mejora pequeña en ambas instancias grandes, de ocho a siete en un caso y de cinco a cuatro en el otro, a costes muy distintos. Estas dimensiones, por tanto, son mejorables, pero su precio es alto y su margen, estrecho.

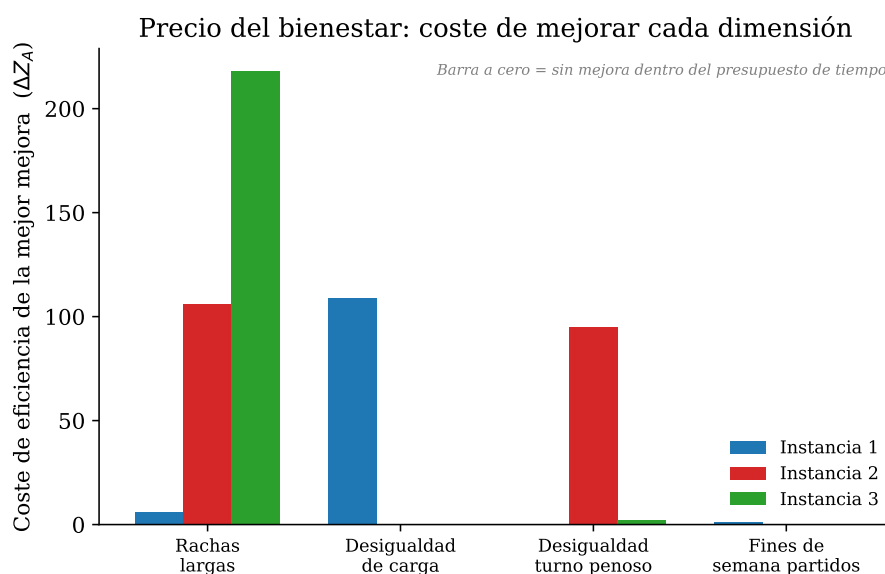


Figura 6.2: Precio del bienestar: coste de eficiencia de la mejor mejora alcanzada en cada dimensión, por instancia.

Nota: Una barra a cero indica que no se logró mejora dentro del presupuesto de tiempo.

Por último, la satisfacción de preferencias y la equidad de fin de semana apenas ofrecen recorrido en esta muestra, sencillamente porque ya parten de un valor muy bueno en la configuración de referencia. La satisfacción de preferencias de la instancia mayor ya es de 0,984 en el baseline, y la desigualdad de fin de semana es cero en las tres instancias. Esto no es una limitación del modelo, sino una propiedad de las instancias y de sus reglas duras, que por sí solas ya producen un reparto razonable en esas dos dimensiones. Reforzarlas con su término de bienestar solo confirma que estaban cerca de su mejor valor posible.

La conclusión de conjunto del análisis término a término conecta directamente con lo observado en la configuración de referencia. Las dimensiones con mayor margen de mejora son justo aquellas que la eficiencia pura descuidaba, las rachas largas por encima de todas, y las que ofrecen mejoras baratas son las que el modelo puede reorganizar sin tocar la cobertura. Las dimensiones de equidad, en cambio, son las más caras de mejorar, porque chocan de frente con la forma en que la demanda y las peticiones encajan en el cuadrante.

6.2.3. Escenarios combinados

El análisis término a término muestra el efecto de cada dimensión por separado, pero un gestor real no querría mejorar una sola cosa, sino varias a la vez. El último bloque de experimentos responde a esa situación activando los seis términos simultáneamente, en dos perfiles de intensidad creciente. El perfil moderado asigna a cada término su nivel de peso medio, y el perfil alto, su nivel máximo. La idea es observar cómo se comportan los criterios de bienestar cuando compiten entre sí por un mismo presupuesto de eficiencia. La Tabla 6.5 recoge los resultados, centrándose en las tres dimensiones con más recorrido.

Cuadro 6.5: Escenarios combinados: indicadores principales y coste de eficiencia frente a la configuración de referencia.

Instancia	Escenario	Rachas	Desig. carga	Desig. penoso	Z_A	Coste vs. base
1	baseline	7	2	n.a.	607	(ref.)
1	moderado	4	2	n.a.	613	+6
1	alto	2	0	n.a.	811	+204
2	baseline	15	5	8	828	(ref.)
2	moderado	10	5	8	835	+7
2	alto	1	4	7	1325	+497
3	baseline	25	5	5	1001	(ref.)
3	moderado	19	5	4	1010	+9
3	alto	20	5	4	1006	+5

Lo primero que confirman estos escenarios es que las dimensiones de bienestar compiten entre sí, de modo que activarlas todas a la vez no equivale a sumar las mejoras que cada una lograba por separado. El caso más llamativo es el de la instancia mediana, donde el perfil alto consigue una reducción casi total de las rachas largas, de quince a una sola, pero a un coste de eficiencia muy elevado de casi quinientos puntos. Es el precio de exigir simultáneamente descanso, equidad y preferencias en una plantilla donde la demanda deja poco margen.

La instancia mayor ilustra el otro lado del fenómeno, el del reparto de esfuerzo. En ella, el perfil alto no lleva las rachas a su mínimo posible, sino que las deja en veinte, lejos de las doce que se alcanzaban activando únicamente el término de rachas. La razón es que parte del presupuesto de eficiencia se desvía hacia la equidad del turno penoso, que mejora de cinco a cuatro. Dicho de otro modo, cuando todos los criterios tiran a la vez, el modelo actúa como árbitro y reparte la mejora disponible entre objetivos que apuntan en direcciones distintas, en lugar de volcarla en uno solo. Conviene señalar que, en esta instancia, los perfiles moderado y alto quedan prácticamente indistinguibles en eficiencia y rachas, e incluso se aprecia una pequeña inversión entre ambos que cae dentro del margen de ruido propio de resolver con un presupuesto de tiempo acotado. La lectura prudente es que, una vez gastado el esfuerzo en equilibrar el turno penoso, añadir más peso apenas cambia el resultado.

Más allá de los números concretos, los escenarios combinados arrojan una conclusión útil para la gestión. El perfil moderado ofrece, en las tres instancias, mejoras apreciables a un coste de eficiencia muy bajo, de entre seis y nueve puntos, lo que lo convierte en una opción atractiva y de bajo riesgo. El perfil alto, en cambio, persigue mejoras mayores, pero su precio depende enormemente de la instancia: barato en la instancia grande, donde apenas cuesta cinco puntos, y muy caro en la mediana, donde supera los cuatrocientos. Esta heterogeneidad es en sí misma un hallazgo: no existe un nivel de exigencia de bienestar universalmente

recomendable, porque el coste de un mismo perfil de pesos varía mucho según la estructura concreta de la unidad, su plantilla y su demanda. La Figura 6.3 sintetiza este comportamiento, mostrando para cada instancia cómo descienden las rachas largas al pasar de la configuración de referencia a los perfiles moderado y alto, con el coste de eficiencia anotado sobre cada barra.

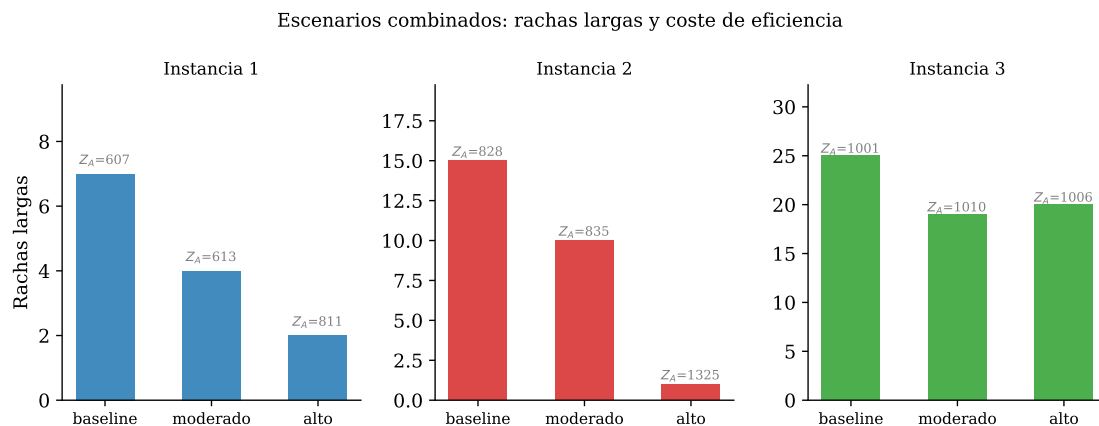


Figura 6.3: Escenarios combinados para las tres instancias: número de rachas largas en la configuración de referencia y en los perfiles moderado y alto.
Nota: El valor de Z_A anotado sobre cada barra indica el coste de eficiencia.

6.3. Comparación global y análisis de los compromisos

Las secciones anteriores examinan los términos de bienestar uno a uno y en conjunto. Esta sección da un paso atrás para leer todos los experimentos como un único cuadro y responder a la pregunta que motiva el trabajo: ¿qué relación hay entre eficiencia y bienestar, y cuánto cuesta moverse de una a otra? El Modelo A y el Modelo B no son dos modelos rivales, sino los dos extremos de un mismo abanico de soluciones. El Modelo A ocupa un extremo, el de la máxima eficiencia, donde el bienestar es solo un subproducto de la solución más barata. El Modelo B, al variar los pesos, recorre el resto del abanico, generando una familia de planificaciones que ceden eficiencia a cambio de bienestar. Cada experimento del capítulo es, en realidad, un punto de esa familia.

La primera lectura global se refiere a la forma de ese abanico, que es la de una frontera con rendimientos decrecientes. Las primeras unidades de bienestar son siempre las más baratas, y el precio de cada unidad adicional crece a medida que se exprime el margen disponible. La frontera de las rachas largas de la Figura 6.1 lo muestra con claridad: su pendiente es suave al principio, cuando se eliminan las rachas más fáciles de deshacer, y se vuelve cada vez más empinada conforme se intenta apurar hacia el mínimo. Esta convexidad tiene una consecuencia práctica directa, y es que casi siempre existe un tramo inicial de mejora de bienestar que se obtiene a un coste de eficiencia muy pequeño.

Ese tramo barato es justamente lo que captura el perfil combinado moderado, que constituye el resultado más vendible del estudio. La Tabla 6.6 expresa sus efectos en términos relativos, comparando la reducción de rachas con el coste de eficiencia como porcentaje de la solución de referencia.

El mensaje de esta tabla es contundente: en las tres instancias, un perfil de bienestar equilibrado reduce las rachas largas entre un 24 % y un 43 %, y en dos de ellas mejora además otra dimensión, todo ello

Cuadro 6.6: Efecto del perfil combinado moderado, con el coste de eficiencia expresado en valor absoluto y como porcentaje de la configuración de referencia.

Instancia	Reducción de rachas	Otras mejoras	Coste de eficiencia
1	7 → 4 (43 %)	fines de semana partidos 2 → 0	+6 (+1,0 %)
2	15 → 10 (33 %)	ninguna adicional	+7 (+0,8 %)
3	25 → 19 (24 %)	turno penoso 5 → 4	+9 (+0,9 %)

por un coste de eficiencia inferior al 1 %. Dicho en términos de gestión, se puede ofrecer al personal una mejora apreciable en descanso y en reparto de los turnos más penosos prácticamente sin tocar la calidad del servicio.

El perfil alto cuenta una historia complementaria, la de los rendimientos decrecientes y la dependencia de la instancia. Cuando se exige el máximo a todos los términos, las mejoras pueden ser espectaculares, como la práctica eliminación de las rachas en la instancia mediana, pero su precio se dispara y, sobre todo, deja de ser predecible: el mismo perfil de pesos cuesta un 60 % de la eficiencia en la instancia mediana, un 34 % en la pequeña y apenas un 0,5 % en la grande. Esta variabilidad no es un defecto del método, sino un reflejo de que el coste del bienestar depende de cómo encajen la plantilla, las peticiones y la demanda en cada unidad concreta. En la instancia grande aún quedaba holgura para acomodar exigencias altas casi gratis, mientras que en la mediana la demanda estaba tan ajustada que cada mejora adicional obligaba a sacrificar mucho.

Conviene precisar qué significa exactamente ese coste de eficiencia, para no malinterpretarlo. El valor de la función de eficiencia recoge sobre todo penalizaciones por desajustes de cobertura y por preferencias incumplidas, de modo que un incremento de, por ejemplo, cien puntos representa pequeñas relajaciones controladas en esas dos partidas, no un colapso del servicio. Y hay un matiz importante para la tranquilidad del lector: todas las soluciones del Modelo B, sea cual sea el peso del bienestar, siguen respetando íntegramente las restricciones duras del problema, es decir, los límites de jornada, los descansos mínimos, el número máximo de fines de semana y los días libres obligatorios. El bienestar adicional nunca se compra violando esas reglas, sino reorganizando los turnos dentro de la región factible y aceptando un sobrecoste cuantificado en la parte blanda del objetivo. Esta garantía es la que permite presentar los compromisos del capítulo como decisiones legítimas de gestión y no como atajos a costa de la seguridad o la legalidad.

En síntesis, el capítulo establece tres hechos cuantitativos. Primero, que existe una frontera de compromiso real y medible entre eficiencia y bienestar, con rendimientos decrecientes. Segundo, que una parte sustancial del bienestar, en especial la reducción de rachas largas, se consigue a un coste casi nulo, lo que convierte al perfil moderado en una recomendación de bajo riesgo. Y tercero, que el coste de las exigencias más ambiciosas es muy sensible a la estructura de cada unidad, de modo que la herramienta no dicta una respuesta única, sino que pone precio a cada nivel de ambición para que la decisión final la tome quien gestiona. Estos tres hechos son el material que la discusión traduce a implicaciones organizativas.

Capítulo 7

Discusión

7.1. Interpretación de los resultados

El hallazgo central del estudio es que existe una frontera de compromiso real y medible entre eficiencia y bienestar, y que esa frontera tiene rendimientos decrecientes. La explicación es estructural. Una planificación que solo persigue la eficiencia dispone de cierta holgura, porque suele haber muchos cuadrantes distintos con un coste casi idéntico. Las primeras mejoras de bienestar se obtienen reutilizando esa holgura, esto es, reordenando los turnos sin tocar la cobertura, y por eso resultan casi gratuitas. Una vez agotada, seguir mejorando obliga a relajar la cobertura o las preferencias, y ahí el coste empieza a crecer. Esto es justo lo que muestran las curvas obtenidas: pendiente suave al principio y cada vez más empinada al apurar hacia el mínimo.

El estudio también explica por qué la eficiencia pura maltrata ciertas dimensiones. Cubrir la demanda concentrando muchos turnos seguidos en pocas personas es barato, porque evita los huecos que obligarían a activar a más personal. De ahí que la solución eficiente tolere rachas largas, un reparto desigual de la carga y una distribución irregular del turno más penoso. Y son precisamente esas tres dimensiones, las que las restricciones duras no regulan, las que el Modelo B puede corregir. En cambio, la equidad de fines de semana ya viene resuelta de partida, porque la restricción dura que limita el número máximo de fines de semana trabajados impone por sí sola un reparto equilibrado, de modo que su término de bienestar apenas tiene margen de actuación.

Esta lectura conecta de forma directa con la evidencia sobre bienestar. Las dimensiones con más recorrido, las rachas de días consecutivos, la acumulación del turno penoso y la equidad de la carga, son justamente las que la literatura identifica como motores del agotamiento y de la percepción de injusticia (Gerlach et al., 2025; Qtait et al., 2025). El modelo, por tanto, actúa sobre las palancas correctas. Conviene subrayar además que todas las soluciones del Modelo B respetan íntegramente las restricciones duras: el bienestar adicional se compra reorganizando dentro de la región factible y aceptando un sobrecoste cuantificado en la parte blanda del objetivo, nunca vulnerando límites de jornada, descansos o días libres.

Situado frente a la literatura, el aporte del trabajo se precisa mejor. Las revisiones del campo coinciden en que la mayoría de los estudios ha orientado la función objetivo hacia la eficiencia, y que la incorporación explícita del bienestar como objetivo, junto con la cuantificación de lo que cuesta, ha recibido menos atención sistemática (Ngoo et al., 2022; O'Connell et al., 2024). Trabajos aplicados recientes sí integran criterios

de equidad y de preferencias en modelos de programación entera y obtienen cuadrantes que mejoran las condiciones del personal (Mystakidis et al., 2024), pero suelen presentar un único cuadrante mejorado, no el precio de cada nivel de bienestar. La diferencia de este trabajo no está, por tanto, en proponer un método nuevo, sino en hacer explícito y medible ese precio: en lugar de una solución más amable, una frontera que indica cuánta eficiencia cuesta cada grado de bienestar. Esa es la lectura que convierte una decisión hasta ahora implícita en un compromiso cuantificado y, por ello, gestionable.

Por último, la fuerte dependencia de la instancia en los perfiles más exigentes, donde un mismo perfil de pesos cuesta desde un 0,5 % hasta un 60 % según el caso, no es un defecto del método, sino el reflejo de que la holgura disponible depende de cómo encajen plantilla, peticiones y demanda en cada unidad concreta. Donde la demanda está muy ajustada, cada mejora adicional obliga a sacrificar más eficiencia.

7.2. Implicaciones para la gestión de RRHH

La implicación más útil para la gestión es que una parte sustancial del bienestar se consigue casi gratis. El perfil moderado reduce las rachas largas entre un 24 % y un 43 % por un coste de eficiencia inferior al 1 % en las tres instancias. En un contexto de escasez estructural de enfermería y de dificultades de retención (Aiken et al., 2002; World Health Organization, 2025), esta es una palanca muy atractiva: si una mejora barata en los factores asociados al agotamiento ayuda a retener personal, y el personal es precisamente el recurso escaso, el retorno organizativo puede superar con mucho su coste aparente.

El segundo aporte es de otra naturaleza. El valor de la herramienta no está en dictar una solución, sino en hacer explícito y cuantificable el compromiso. Frente a un cuadrante manual, opaco y difícil de justificar, el modelo pone precio a cada nivel de ambición de bienestar y permite que sea quien gestiona quien decida cuánto está dispuesto a invertir. Es, por tanto, un apoyo a la decisión, no un sustituto de ella.

De aquí se derivan dos recomendaciones. La primera es adoptar por defecto un perfil moderado, porque ofrece mejoras apreciables con un riesgo casi nulo para el servicio. La segunda es tratar las metas de equidad con realismo: mejorarlas es más caro y a veces choca de frente con la forma en que la demanda encaja en el cuadrante, de modo que conviene decidir de forma consciente cuánta equidad adicional se quiere pagar, en lugar de darla por supuesta.

Todo ello con una cautela que la propia literatura subraya: mejorar los indicadores sobre el papel no garantiza un efecto positivo real si el sistema no es aceptado por el personal o si erosiona el juicio profesional y el componente relacional que los criterios automáticos no capturan (Gerlach et al., 2025; O'Connell et al., 2024). La herramienta rinde más como apoyo a la supervisora, idealmente con participación del personal en la fijación de los criterios, que como un automatismo impuesto desde arriba.

7.3. Limitaciones y riesgos

Como todo estudio acotado, este trabajo presenta limitaciones que conviene reconocer con transparencia, entendidas no como fallos sino como las fronteras del diseño elegido y como la agenda natural de trabajo futuro.

La primera es el tamaño de la muestra: tres instancias, todas de dos semanas. Es una muestra estrecha, pero esa estrechez es precisamente lo que garantiza la limpieza del análisis, porque cada instancia se resuelve hasta su óptimo conocido y la densidad de configuraciones de pesos compensa en buena medida el número

reducido de casos. La generalización a unidades mayores o a horizontes más largos queda como extensión inmediata del estudio.

La segunda es de naturaleza computacional. Las instancias de cuatro semanas no se resolvían de forma fiable dentro del presupuesto de tiempo y por eso se excluyeron, y entre las tres utilizadas la optimalidad solo se demuestra en la más pequeña. Conviene leer en positivo el efecto del arranque en caliente: al ser una estrategia conservadora, las mejoras de bienestar reportadas son un suelo y no un techo, de modo que el beneficio real es al menos el documentado y probablemente mayor. Abordar instancias más grandes con más tiempo de cálculo o con una metaheurística es una línea futura clara.

La tercera es la distancia entre el benchmark y la práctica real. El conjunto de Curtois y Qu no recoge elementos presentes en entornos hospitalarios reales, como la cobertura por nivel de competencia, los turnos especiales o las incompatibilidades entre enfermeras, que sí incorporan formulaciones más realistas (Ceschia et al., 2023). Las conclusiones son por tanto válidas para el problema modelizado, y su traslado a una planta concreta, incorporando esa complejidad, constituye el paso aplicado siguiente.

La cuarta afecta a la medición del bienestar. Los seis indicadores son aproximaciones, y la equidad se ha operacionalizado como la diferencia entre el máximo y el mínimo, que es solo una de las nociones posibles de justicia distributiva. Más relevante aún, los pesos de bienestar los fija el investigador mediante calibración y no se obtienen de las preferencias del personal, de modo que decidir cuánto bienestar perseguir es una elección de gestión, no un resultado técnico. El riesgo asociado, el exceso de confianza en la automatización, se mitiga precisamente con lo que sugieren estos límites: que los pesos se fijen de forma participada y que la herramienta apoye el juicio profesional en lugar de sustituirlo. Antes que debilidades, estas fronteras dibujan el camino por el que el trabajo puede crecer.

Capítulo 8

Conclusiones

8.1. Síntesis de hallazgos

El objetivo de este trabajo era formular dos modelos de planificación de turnos de enfermería, uno de referencia orientado a la eficiencia (Modelo A) y otro que incorpora de forma explícita el bienestar del personal (Modelo B), y cuantificar el compromiso entre ambos en una unidad de hospitalización. Para ello se construyó una formulación común de restricciones duras, sobre la que ambos modelos comparten la misma estructura y se diferencian únicamente en la función objetivo; se validó el Modelo A reproduciendo los óptimos publicados del benchmark; y se exploró el Modelo B variando los pesos de los seis criterios de bienestar, primero término a término y después en perfiles combinados, sobre tres instancias de referencia. El objetivo se ha cumplido, y de los experimentos se desprenden tres conclusiones principales.

1. Existe una frontera de compromiso real y medible entre eficiencia y bienestar, con rendimientos decrecientes. Los dos modelos no son rivales, sino los extremos de un mismo continuo de soluciones: el Modelo A ocupa el extremo de máxima eficiencia y el Modelo B, al aumentar los pesos, recorre el resto del abanico. La consecuencia práctica es que la elección entre eficiencia y bienestar deja de ser una disyuntiva de todo o nada y se convierte en una decisión graduable, en la que cada nivel de bienestar tiene un precio conocido.
2. Una parte sustancial del bienestar se obtiene a un coste casi nulo. Un perfil de pesos moderado reduce las rachas de días consecutivos trabajados entre un 24% y un 43% por menos de un 1% de eficiencia, y en dos de las tres instancias mejora además otra dimensión, sin vulnerar en ningún caso las restricciones duras del problema. Este es el resultado más accionable del estudio, porque identifica un margen de mejora del bienestar prácticamente gratuito que cualquier unidad podría aprovechar.
3. El coste de las exigencias de bienestar más ambiciosas depende fuertemente de la instancia, entre un 0,5% y un 60% según el caso. No existe, por tanto, un nivel de bienestar universalmente recomendable, porque el precio de un mismo perfil de pesos varía mucho con la forma en que encajan la plantilla, las peticiones y la demanda de cada unidad. Esta heterogeneidad es en sí misma un hallazgo, y refuerza la idea de que la herramienta no debe dictar una respuesta única, sino poner precio a cada grado de ambición para que la decisión final la tome quien gestiona.

En conjunto, la aportación del trabajo no reside en proponer un nuevo método de resolución, sino en transformar una decisión que tradicionalmente se tomaba de forma implícita en un compromiso explícito, cuantificado y, por ello, gestionable. De este modo se cubre el hueco señalado en la revisión de la literatura, donde la incorporación del bienestar como objetivo y la medición de su coste habían recibido una atención escasa, y se hace sobre una base validada que garantiza la corrección de los resultados.

8.2. Recomendaciones prácticas

De los hallazgos anteriores se derivan cuatro recomendaciones prácticas para una dirección de enfermería que quiera incorporar criterios de bienestar a la planificación de turnos.

1. **Adoptar un perfil de bienestar moderado como configuración por defecto.** Es la opción con mejor relación entre mejora y riesgo: aporta avances apreciables en descanso por un coste de eficiencia inferior al 1 %, sin comprometer la cobertura ni vulnerar las reglas duras. Salvo que exista un motivo en contra, debería ser el punto de partida de cualquier cuadrante, en lugar de la solución puramente eficiente.
2. **Empezar por las dimensiones baratas y tratar la equidad como una decisión costeada.** Las mejoras en rachas largas y en fines de semana completos suelen conseguirse reorganizando turnos sin apenas penalización, por lo que conviene agotarlas en primer lugar. Las metas de equidad de carga o de turno penoso, en cambio, son más caras y dependen mucho de cada unidad, de modo que deben decidirse de forma consciente, conociendo su precio, y no darse por supuestas.
3. **Presentar el abanico de opciones, no un único cuadrante.** El mayor valor de la herramienta está en mostrar el precio de cada nivel de bienestar, así que conviene ofrecer a quien decide un menú de cuadrantes con su coste de eficiencia asociado, en vez de una solución cerrada. Esto aporta transparencia y convierte la elaboración del cuadrante en una decisión informada y justificable ante el personal.
4. **Fijar los pesos de forma participada y preservar el juicio profesional.** Dado que decidir cuánto bienestar perseguir es una cuestión normativa y no técnica, los pesos deberían acordarse con participación del personal afectado, y la herramienta emplearse como apoyo a la supervisora y no como un automatismo que sustituya su criterio (Gerlach et al., 2025; O'Connell et al., 2024).

En conjunto, estas recomendaciones apuntan a un uso prudente y transparente de la optimización: aprovechar las mejoras baratas, poner precio a las caras y dejar la decisión final en manos de quien conoce la unidad.

8.3. Líneas futuras

Las limitaciones señaladas no son puntos finales, sino la agenda natural por la que el trabajo puede continuar. Se concretan en cinco líneas.

1. **Escalar a instancias mayores y horizontes más largos.** Las instancias de cuatro o más semanas quedaron fuera por su coste computacional; abordarlas con más tiempo de cálculo o con una metaheurística permitiría comprobar si la frontera de mejoras baratas se mantiene a mayor escala.

2. **Acercar la formulación a la práctica hospitalaria real.** Incorporar la cobertura por nivel de competencia, los turnos especiales y las incompatibilidades entre enfermeras aproximaría el modelo a las condiciones reales de una planta y otorgaría mayor validez externa a las conclusiones.
3. **Obtener los pesos de bienestar de forma empírica y participada.** En lugar de fijarlos por calibración, podrían derivarse de las preferencias declaradas del personal mediante encuestas o métodos de decisión multicriterio, lo que reforzaría la legitimidad de la elección normativa de cuánto bienestar perseguir.
4. **Validar el enfoque en una unidad real.** Aplicar el modelo a los datos de un servicio concreto y contrastar los cuadrantes generados con la planificación manual permitiría medir su efecto sobre indicadores de bienestar reales, más allá de los benchmarks académicos.
5. **Explorar técnicas multiobjetivo y otras nociones de equidad.** Generar de forma sistemática el frente de Pareto entre eficiencia y bienestar, y ensayar medidas de equidad distintas de la diferencia entre máximo y mínimo, afinaría el análisis del compromiso.

En conjunto, estas líneas llevan el trabajo desde un estudio metodológico riguroso pero acotado hacia su validación y su aplicación en entornos hospitalarios reales.

Declaración de uso de herramientas de Inteligencia Artificial Generativa en Trabajos de Fin de Grado

ADVERTENCIA: Desde la Universidad consideramos que ChatGPT u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Mario Laborda Plaza, estudiante de E2+Business Analytics, de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado “OPTIMIZACIÓN DE TURNOS DE ENFERMERÍA EN UNA UNIDAD DE HOSPITALIZACIÓN: ANÁLISIS DEL COMPROMISO ENTRE EFICIENCIA Y BIENESTAR LABORAL”, declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Brainstorming de ideas de investigación:** Utilizado para idear y esbozar posibles áreas de investigación.
2. **Crítico:** Para encontrar contra-argumentos a una tesis específica que pretendo defender.
3. **Referencias:** Usado conjuntamente con otras herramientas, como Science, para identificar referencias preliminares que luego he contrastado y validado.
4. **Metodólogo:** Para descubrir métodos aplicables a problemas específicos de investigación.
5. **Interpretador de código:** Para realizar análisis de datos preliminares.
6. **Estudios multidisciplinares:** Para comprender perspectivas de otras comunidades sobre temas de naturaleza multidisciplinar.
7. **Constructor de plantillas:** Para diseñar formatos específicos para secciones del trabajo.
8. **Corrector de estilo literario y de lenguaje:** Para mejorar la calidad lingüística y estilística del texto.
9. **Generador previo de diagramas de flujo y contenido:** Para esbozar diagramas iniciales.

10. **Sintetizador y divulgador de libros complicados:** Para resumir y comprender literatura compleja.
11. **Generador de datos sintéticos de prueba:** Para la creación de conjuntos de datos ficticios.
12. **Generador de problemas de ejemplo:** Para ilustrar conceptos y técnicas.
13. **Revisor:** Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
14. **Generador de encuestas:** Para diseñar cuestionarios preliminares.
15. **Traductor:** Para traducir textos de un lenguaje a otro.

Bibliografía y referencias

- Abdalkareem, Z. A., Amir, A., Al-Betar, M. A., Ekhan, P., & Hammouri, A. I. (2021). Healthcare scheduling in optimization context: A review. *Health and Technology*, 11(3), 445-469. <https://doi.org/10.1007/s12553-021-00547-5>
- Aiken, L. H., Clarke, S. P., Sloane, D. M., Sochalski, J., & Silber, J. H. (2002). Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. *JAMA*, 288(16), 1987-1993. <https://doi.org/10.1001/jama.288.16.1987>
- Brucker, P., Burke, E. K., Curtois, T., Qu, R., & Vanden Berghe, G. (2010). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, 16(4), 559-573.
- Burke, E. K., & Curtois, T. (2014). New approaches to nurse rostering benchmark instances. *European Journal of Operational Research*, 237(1), 71-81.
- Burke, E. K., De Causmaecker, P., Vanden Berghe, G., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of Scheduling*, 7(6), 441-499.
- Ceschia, S., Dang, N., De Causmaecker, P., Haspeslagh, S., & Schaerf, A. (2019). The Second International Nurse Rostering Competition. *Annals of Operations Research*, 274(1-2), 171-186. <https://doi.org/10.1007/s10479-018-2816-0>
- Ceschia, S., Dang, N. T. T., De Causmaecker, P., Haspeslagh, S., & Schaerf, A. (2015). *Second International Nurse Rostering Competition (INRC-II): Problem description and rules*. arXiv: 1501.04177. <https://arxiv.org/abs/1501.04177>
- Ceschia, S., Di Gaspero, L., Mazzaracchio, V., Policante, G., & Schaerf, A. (2023). Solving a real-world nurse rostering problem by simulated annealing. *Operations Research for Health Care*, 36, 100379.
- Cheang, B., Li, H., Lim, A., & Rodrigues, B. (2003). Nurse rostering problems — A bibliographic survey. *European Journal of Operational Research*, 151(3), 447-460.
- De Causmaecker, P., & Vanden Berghe, G. (2011). A categorisation of nurse rostering problems. *Journal of Scheduling*, 14(1), 3-16.
- den Hartog, S. G. M., Hoogeveen, H., & van der Zanden, T. C. (2023). On the complexity of nurse rostering problems. *Operations Research Letters*, 51(5), 483-487.

- Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, *153*(1), 3-27. [https://doi.org/10.1016/S0377-2217\(03\)00095-X](https://doi.org/10.1016/S0377-2217(03)00095-X)
- Gerlach, M., Renggli, F. J., Bieri, J. S., Sariyar, M., & Golz, C. (2025). Exploring nurse perspectives on AI-based shift scheduling for fairness, transparency, and work-life balance. *BMC Nursing*, *24*. <https://doi.org/10.1186/s12912-025-03808-0>
- Haspeslagh, S., De Causmaecker, P., Schaerf, A., & Stølevik, M. (2014). The first international nurse rostering competition 2010. *Annals of Operations Research*, *218*(1), 221-236.
- He, F., & Qu, R. (2012). A constraint programming based column generation approach to nurse rostering problems. *Computers & Operations Research*, *39*(12), 3331-3343.
- Jaumard, B., Semet, F., & Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, *107*(1), 1-18.
- Ministerio de Sanidad. (2024). *Situación actual y estimación de la necesidad de enfermeras en España, 2024* (Informe técnico). Ministerio de Sanidad. Consultado el 16 de junio de 2026, desde https://www.sanidad.gob.es/areas/cuidadosEnSalud/investigacionDatos/docs/SITUACION_ACTUAL_Y_ESTIMACION_DE_LA_NECESIDAD_DE_ENFERMERAS_EN_ESPANA_2024_PPT.pdf
- Mystakidis, A., Koukaras, C., Koukaras, P., Kaparis, K., Stavrinides, S. G., & Tjortjis, C. (2024). Optimizing nurse rostering: A case study using integer programming to enhance operational efficiency and care quality. *Healthcare*, *12*(24), 2545. <https://doi.org/10.3390/healthcare12242545>
- Ngoo, C. M., Goh, S. L., Sabar, N. R., Sze, S. N., Abdullah, S., & Kendall, G. (2022). A survey of the nurse rostering solution methodologies: The state-of-the-art and emerging trends. *IEEE Access*, *10*, 56504-56524. <https://doi.org/10.1109/ACCESS.2022.3177280>
- O'Connell, M., Barry, J., Hartigan, I., Cornally, N., & Saab, M. M. (2024). The impact of electronic and self-rostering systems on healthcare organisations and healthcare workers: A mixed-method systematic review. *Journal of Clinical Nursing*, *33*(7), 2374-2387. <https://doi.org/10.1111/jocn.17114>
- OECD/European Union. (2024). *Health at a glance: Europe 2024: State of health in the EU cycle*. OECD Publishing. <https://doi.org/10.1787/b3704e14-en>
- Perron, L., & Didier, F. (2024). *CP-SAT (Versión 9.x)*. https://developers.google.com/optimization/cp/cp_solver/
- Perron, L., & Furnon, V. (2024). *OR-Tools (Versión 9.x)*. <https://developers.google.com/optimization/>

- Qtait, M., Al Ali, M. F., & Jaradat, Y. (2025). The impact of rotating shift work on nurse burnout: A systematic review of contributing factors and organizational strategies. *SAGE Open Nursing, 11*. <https://doi.org/10.1177/23779608251374232>
- Strandmark, P., Qu, Y., & Curtois, T. (2020). First-order linear programming in a column generation-based heuristic approach to the nurse rostering problem. *Computers & Operations Research, 120*, 104945. <https://doi.org/10.1016/j.cor.2020.104945>
- Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research, 226*(3), 367-385. <https://doi.org/10.1016/j.ejor.2012.11.029>
- Warner, D. M., & Prawda, J. (1972). A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science, 19*(4-part-1), 411-422. <https://doi.org/10.1287/mnsc.19.4.411>
- World Health Organization. (2025). *State of the world's nursing 2025: Investing in education, jobs, leadership and service delivery*. World Health Organization. <https://www.who.int/publications/i/item/9789240110236>

Anexo A. Instancias utilizadas

Las tres instancias proceden del benchmark público de Curtois y Qu; su formato se describió en el capítulo de datos. Este anexo detalla sus parámetros concretos. La Tabla 8.1 los resume y el Listado 8.1 reproduce, como ejemplo del formato, el fichero completo de la instancia más pequeña. Los ficheros de las tres están disponibles en el repositorio público del benchmark.

Cuadro 8.1: Parámetros de las instancias utilizadas.

Parámetro	Instance1	Instance2	Instance3
Enfermeras	8 (A–H)	14 (A–N)	20 (A–T)
Días (semanas)	14 (2)	14 (2)	14 (2)
Tipos de turno	D	E, L	E, D, L
Duración de cada turno (min)	480	480	480
Secuencias prohibidas	ninguna	L→E	D→E, L→E, L→D
Tipos de contrato	1	2	3
Máx. fines de semana	1	1	1
Días libres obligatorios	1/enf.	1/enf.	1/enf.
Peticiones de trabajar	21	50	39
Peticiones de no trabajar	5	12	25
Entradas de cobertura	14	28	42
Óptimo publicado	607	828	1001

En *Instance1* la plantilla es homogénea: un único turno de día, con todas las enfermeras entre 7 y 9 turnos en el horizonte (máximo 4320 y mínimo 3360 minutos), un máximo de 5 turnos consecutivos, un mínimo de 2 y dos días libres consecutivos. *Instance2* añade los turnos de mañana (E) y tarde (L) con la secuencia L→E prohibida (no se puede trabajar una mañana justo después de una tarde) y distingue dos contratos: uno a jornada completa (enfermeras A–J) y otro reducido (K–N), con menos horas y mínimos de consecutividad más laxos. *Instance3* incorpora un tercer turno (D) y un sistema de secuencias prohibidas más estricto que fuerza el descanso entre turnos; presenta tres patrones de contrato, incluido un grupo con hasta 6 turnos consecutivos y 3 días libres mínimos, y limita el acceso al turno de tarde para varias enfermeras.

Listing 8.1: Fichero de la instancia Instance1 (formato de texto del benchmark).

```
SECTION_HORIZON
# All instances start on a Monday. Horizon length in days:
14

SECTION_SHIFTS
# ShiftID, Length in mins, Shifts which cannot follow this shift
```

D,480,

SECTION_STAFF

ID, MaxShifts, MaxTotalMinutes, MinTotalMinutes, MaxConsecutiveShifts,
MinConsecutiveShifts, MinConsecutiveDaysOff, MaxWeekends

A,D=14,4320,3360,5,2,2,1
B,D=14,4320,3360,5,2,2,1
C,D=14,4320,3360,5,2,2,1
D,D=14,4320,3360,5,2,2,1
E,D=14,4320,3360,5,2,2,1
F,D=14,4320,3360,5,2,2,1
G,D=14,4320,3360,5,2,2,1
H,D=14,4320,3360,5,2,2,1

SECTION_DAYS_OFF

EmployeeID, DayIndexes (start at zero)

A,0
B,5
C,8
D,2
E,9
F,5
G,1
H,7

SECTION_SHIFT_ON_REQUESTS

EmployeeID, Day, ShiftID, Weight

A,2,D,2
A,3,D,2
B,0,D,3
B,1,D,3
B,2,D,3
B,3,D,3
B,4,D,3
C,0,D,1
C,1,D,1
C,2,D,1
C,3,D,1
C,4,D,1
D,8,D,2
D,9,D,2
F,0,D,2
F,1,D,2
H,9,D,1
H,10,D,1
H,11,D,1
H,12,D,1
H,13,D,1

SECTION_SHIFT_OFF_REQUESTS

EmployeeID, Day, ShiftID, Weight

C,12,D,1
C,13,D,1
F,8,D,3
H,2,D,3
H,3,D,3

SECTION_COVER

Day, ShiftID, Requirement, Weight for under, Weight for over

0,D,5,100,1

1,D,7,100,1

2,D,6,100,1

3,D,4,100,1

4,D,5,100,1

5,D,5,100,1

6,D,5,100,1

7,D,6,100,1

8,D,7,100,1

9,D,4,100,1

10,D,2,100,1

11,D,5,100,1

12,D,6,100,1

13,D,4,100,1

Anexo B. Código fuente

La implementación se organiza en tres módulos: la carga de datos y el Modelo A, el Modelo B con los seis términos de bienestar y las métricas, y el script que ejecuta los experimentos. El código es el mismo con el que se obtuvieron los resultados del capítulo de experimentos.

B.1. Carga de datos y Modelo A (nrp_model.py)

```
1
2 """Parser y Modelo A (baseline) para el benchmark de Curtois & Qu."""
3
4
5
6 from ortools.sat.python import cp_model
7
8
9
10 def parse_instance(path):
11     data = {
12         "horizon": 0,
13         "shifts": {},          # id -> {"length": int, "forbidden": [
14             "ids"]}
15         "staff": {},          # id -> dict de limites
16         "days_off": {},      # id -> [dias]
17         "on_req": [],         # (emp, day, shift, weight)
18         "off_req": [],        # (emp, day, shift, weight)
19         "cover": {},          # (day, shift) -> (req, w_under,
20             "w_over")
21     }
22     section = None
23     with open(path) as f:
24         for raw in f:
25             line = raw.strip()
26             if not line or line.startswith("#"):
27                 continue
28             if line.startswith("SECTION_"):
29                 section = line
30                 continue
31             parts = line.split(",")
32             if section == "SECTION_HORIZON":
```

```

27         data["horizon"] = int(line)
28     elif section == "SECTION_SHIFTS":
29         sid = parts[0]
30         length = int(parts[1])
31         forbidden = []
32         if len(parts) > 2 and parts[2].strip():
33             forbidden = parts[2].split("|")
34         data["shifts"][sid] = {"length": length, "forbidden
35                                     ": forbidden}
36     elif section == "SECTION_STAFF":
37         sid = parts[0]
38         maxshifts = {}
39         for token in parts[1].split("|"):
40             s, n = token.split("=")
41             maxshifts[s] = int(n)
42         data["staff"][sid] = {
43             "maxshifts": maxshifts,
44             "maxmin": int(parts[2]),
45             "minmin": int(parts[3]),
46             "maxcons": int(parts[4]),
47             "mincons": int(parts[5]),
48             "minoff": int(parts[6]),
49             "maxwe": int(parts[7]),
50         }
51         data["days_off"].setdefault(sid, [])
52     elif section == "SECTION_DAYS_OFF":
53         emp = parts[0]
54         data["days_off"].setdefault(emp, [])
55         for d in parts[1:]:
56             if d.strip():
57                 data["days_off"][emp].append(int(d))
58     elif section == "SECTION_SHIFT_ON_REQUESTS":
59         data["on_req"].append((parts[0], int(parts[1]),
60                                 parts[2], int(parts[3])))
61     elif section == "SECTION_SHIFT_OFF_REQUESTS":
62         data["off_req"].append((parts[0], int(parts[1]),
63                                 parts[2], int(parts[3])))
64     elif section == "SECTION_COVER":
65         day, sid = int(parts[0]), parts[1]
66         data["cover"][(day, sid)] = (int(parts[2]), int(
67             parts[3]), int(parts[4]))
68
69     return data
70
71 def weekends(horizon):

```

```

68     """Lista de (sat, sun_or_None) por fin de semana. Lunes = dia
69         0."""
70     we = []
71     w = 0
72     while 7 * w + 5 < horizon:
73         sat = 7 * w + 5
74         sun = 7 * w + 6 if 7 * w + 6 < horizon else None
75         we.append((sat, sun))
76         w += 1
77     return we
78
79 def build_model_A(data):
80     model = cp_model.CpModel()
81     N = list(data["staff"].keys())
82     H = data["horizon"]
83     D = list(range(H))
84     S = list(data["shifts"].keys())
85     WE = weekends(H)
86
87     # Variable principal
88     x = {}
89     for n in N:
90         for d in D:
91             for s in S:
92                 x[(n, d, s)] = model.NewBoolVar(f"x_{n}_{d}_{s}")
93
94     def works(n, d):
95         return sum(x[(n, d, s)] for s in S)
96
97     # HC1: un turno como maximo por dia
98     for n in N:
99         for d in D:
100             model.Add(works(n, d) <= 1)
101
102     # HC2: secuencias prohibidas
103     for n in N:
104         for d in range(H - 1):
105             for s in S:
106                 for s2 in data["shifts"][s]["forbidden"]:
107                     model.Add(x[(n, d, s)] + x[(n, d + 1, s2)] <=
108                                 1)
109
110     # HC3: maximo de turnos de cada tipo
111     for n in N:

```

```

111     for s in S:
112         mx = data["staff"][n]["maxshifts"].get(s, H)
113         model.Add(sum(x[(n, d, s)] for d in D) <= mx)
114
115     # HC4/HC5: minutos totales max y min
116     for n in N:
117         total = sum(data["shifts"][s]["length"] * x[(n, d, s)] for
118                     d in D for s in S)
119         model.Add(total <= data["staff"][n]["maxmin"])
120         model.Add(total >= data["staff"][n]["minmin"])
121
122     # HC6: maximo de turnos consecutivos
123     for n in N:
124         mc = data["staff"][n]["maxcons"]
125         for d in range(H - mc):
126             model.Add(sum(works(n, d + j) for j in range(mc + 1))
127                       <= mc)
128
129     # HC7: minimo de turnos consecutivos
130     for n in N:
131         mn = data["staff"][n]["mincons"]
132         for k in range(1, mn):
133             for d in range(0, H - k - 1):
134                 model.Add((1 - works(n, d)) + sum(works(n, d + j)
135                                                     for j in range(1, k + 1))
136                           + (1 - works(n, d + k + 1)) <= k + 1)
137
138     # HC8: minimo de dias libres consecutivos
139     for n in N:
140         mo = data["staff"][n]["minoff"]
141         for k in range(1, mo):
142             for d in range(0, H - k - 1):
143                 model.Add(works(n, d) + sum((1 - works(n, d + j))
144                                             for j in range(1, k + 1))
145                           + works(n, d + k + 1) <= k + 1)
146
147     # HC9: maximo de fines de semana
148     y = {}
149     for n in N:
150         for w, (sat, sun) in enumerate(WE):
151             y[(n, w)] = model.NewBoolVar(f"y_{n}_{w}")
152             model.Add(y[(n, w)] >= works(n, sat))
153             if sun is not None:
154                 model.Add(y[(n, w)] >= works(n, sun))

```

```

151         model.Add(y[(n, w)] <= works(n, sat) + works(n, sun
152             ))
153         else:
154             model.Add(y[(n, w)] <= works(n, sat))
155             model.Add(sum(y[(n, w)] for w in range(len(WE))) <= data["
156                 staff"][n]["maxwe"])
157
158     # HC10: dias libres obligatorios
159     for n in N:
160         for d in data["days_off"].get(n, []):
161             model.Add(works(n, d) == 0)
162
163     # Objetivo (Modelo A): peticiones + cobertura
164     obj = []
165     for (emp, day, s, wgt) in data["on_req"]:
166         obj.append(wgt * (1 - x[(emp, day, s)]))
167     for (emp, day, s, wgt) in data["off_req"]:
168         obj.append(wgt * x[(emp, day, s)])
169     for (day, s), (req, wu, wo) in data["cover"].items():
170         u = model.NewIntVar(0, len(N), f"u_{day}_{s}")
171         e = model.NewIntVar(0, len(N), f"e_{day}_{s}")
172         model.Add(sum(x[(n, day, s)] for n in N) - req == e - u)
173         obj.append(wu * u + wo * e)
174
175     model.Minimize(sum(obj))
176     return model, x
177
178 if __name__ == "__main__":
179     import sys
180     data = parse_instance(sys.argv[1] if len(sys.argv) > 1 else "
181         instances/Instance1.txt")
182     print("Enfermeras:", len(data["staff"]), "| Dias:", data["
183         horizon"],
184         "| Turnos:", list(data["shifts"].keys()))
185     model, x = build_model_A(data)
186     solver = cp_model.CpSolver()
187     solver.parameters.max_time_in_seconds = 30
188     solver.parameters.num_search_workers = 8
189     solver.parameters.random_seed = 42
190     status = solver.Solve(model)
191     print("Estado:", solver.StatusName(status))
192     print("Objetivo:", solver.ObjectiveValue())
193     print("Tiempo:", round(solver.WallTime(), 3), "s")

```

B.2. Modelo B y métricas (nrp_full.py)

```
1 """Modelo B (bienestar) sobre el benchmark de Curtois & Qu.
2 Construye Modelo A + 6 terminos de bienestar. Si todos los pesos
   son 0, es Modelo A.
3 """
4 from ortools.sat.python import cp_model
5 from nrp_model import parse_instance, weekends
6
7
8 def burdensome_shift(data):
9     """Turno mas penoso para B5: noche si existe, si no el Late, si
10        no None."""
11     S = list(data["shifts"].keys())
12     if "N" in S:
13         return "N"
14     if "L" in S:
15         return "L"
16     return None
17
18 def build_model(data, w=None, theta=3):
19     """w = dict con 11..16 (pesos de bienestar). None => Modelo A."
20        ""
21     if w is None:
22         w = {}
23     L = {k: w.get(k, 0) for k in ["11", "12", "13", "14", "15", "16
24        "]}
25
26     model = cp_model.CpModel()
27     N = list(data["staff"].keys())
28     H = data["horizon"]
29     D = list(range(H))
30     S = list(data["shifts"].keys())
31     WE = weekends(H)
32     star = burdensome_shift(data)
33
34     x = {(n, d, s): model.NewBoolVar(f"x_{n}_{d}_{s}")
35          for n in N for d in D for s in S}
36
37     def works(n, d):
38         return sum(x[(n, d, s)] for s in S)
39
40     # ---- Restricciones duras (igual que Modelo A) ----
41     for n in N:
42         for d in D:
```

```

41         model.Add(works(n, d) <= 1)
42     for n in N:
43         for d in range(H - 1):
44             for s in S:
45                 for s2 in data["shifts"][s]["forbidden"]:
46                     model.Add(x[(n, d, s)] + x[(n, d + 1, s2)] <=
47                                 1)
48     for n in N:
49         for s in S:
50             model.Add(sum(x[(n, d, s)] for d in D) <= data["staff"
51                                 ][n]["maxshifts"].get(s, H))
52     for n in N:
53         total = sum(data["shifts"][s]["length"] * x[(n, d, s)] for
54                     d in D for s in S)
55         model.Add(total <= data["staff"][n]["maxmin"])
56         model.Add(total >= data["staff"][n]["minmin"])
57     for n in N:
58         mc = data["staff"][n]["maxcons"]
59         for d in range(H - mc):
60             model.Add(sum(works(n, d + j) for j in range(mc + 1))
61                         <= mc)
62     for n in N:
63         mn = data["staff"][n]["mincons"]
64         for k in range(1, mn):
65             for d in range(0, H - k - 1):
66                 model.Add((1 - works(n, d)) + sum(works(n, d + j)
67                     for j in range(1, k + 1))
68                         + (1 - works(n, d + k + 1)) <= k + 1)
69     for n in N:
70         mo = data["staff"][n]["minoff"]
71         for k in range(1, mo):
72             for d in range(0, H - k - 1):
73                 model.Add(works(n, d) + sum((1 - works(n, d + j))
74                     for j in range(1, k + 1))
75                     + works(n, d + k + 1) <= k + 1)
76     y = {}
77     for n in N:
78         for wk, (sat, sun) in enumerate(WE):
79             y[(n, wk)] = model.NewBoolVar(f"y_{n}_{wk}")
80             model.Add(y[(n, wk)] >= works(n, sat))
81             if sun is not None:
82                 model.Add(y[(n, wk)] >= works(n, sun))
83                 model.Add(y[(n, wk)] <= works(n, sat) + works(n,
84                     sun))
85             else:

```

```

79         model.Add(y[(n, wk)] <= works(n, sat))
80     model.Add(sum(y[(n, wk)] for wk in range(len(WE))) <= data[
81         "staff"][n]["maxwe"])
82 for n in N:
83     for d in data["days_off"].get(n, []):
84         model.Add(works(n, d) == 0)
85
86 # ---- Objetivo: Modelo A ----
87 objA = []
88 for (emp, day, s, wgt) in data["on_req"]:
89     objA.append(wgt * (1 - x[(emp, day, s)]))
90 for (emp, day, s, wgt) in data["off_req"]:
91     objA.append(wgt * x[(emp, day, s)])
92 u, e = {}, {}
93 for (day, s), (req, wu, wo) in data["cover"].items():
94     u[(day, s)] = model.NewIntVar(0, len(N), f"u_{day}_{s}")
95     e[(day, s)] = model.NewIntVar(0, len(N), f"e_{day}_{s}")
96     model.Add(sum(x[(n, day, s)] for n in N) - req == e[(day, s)
97         ]) - u[(day, s)])
98     objA.append(wu * u[(day, s)] + wo * e[(day, s)])
99
100 # ---- Terminos de bienestar ----
101 aux = {"u": u, "e": e, "star": star}
102 objB = []
103
104 # B1: equidad de carga
105 T = {n: model.NewIntVar(0, H, f"T_{n}") for n in N}
106 for n in N:
107     model.Add(T[n] == sum(x[(n, d, s)] for d in D for s in S))
108 Tmax = model.NewIntVar(0, H, "Tmax"); Tmin = model.NewIntVar(0,
109     H, "Tmin")
110 for n in N:
111     model.Add(Tmax >= T[n]); model.Add(Tmin <= T[n])
112 aux["Tmax"], aux["Tmin"] = Tmax, Tmin
113 if L["l1"]:
114     objB.append(L["l1"] * (Tmax - Tmin))
115
116 # B2: equidad de fines de semana
117 F = {n: model.NewIntVar(0, len(WE), f"F_{n}") for n in N}
118 for n in N:
119     model.Add(F[n] == sum(y[(n, wk)] for wk in range(len(WE))))
120 Fmax = model.NewIntVar(0, len(WE), "Fmax"); Fmin = model.
121     NewIntVar(0, len(WE), "Fmin")
122 for n in N:
123     model.Add(Fmax >= F[n]); model.Add(Fmin <= F[n])

```

```

120 aux["Fmax"], aux["Fmin"] = Fmax, Fmin
121 if L["12"]:
122     objB.append(L["12"] * (Fmax - Fmin))
123
124 # B3: suavizado de secuencias largas (umbral theta)
125 g = []
126 for n in N:
127     for d in range(0, H - theta):
128         gv = model.NewIntVar(0, 1, f"g_{n}_{d}")
129         model.Add(gv >= sum(works(n, d + j) for j in range(
130             theta + 1)) - theta)
131         g.append(gv)
132 aux["g"] = g
133 if L["13"]:
134     objB.append(L["13"] * sum(g))
135
136 # B4: refuerzo de preferencias (reutiliza penalizaciones de
137     peticiones)
138 pref_terms = []
139 for (emp, day, s, wgt) in data["on_req"]:
140     pref_terms.append(wgt * (1 - x[(emp, day, s)]))
141 for (emp, day, s, wgt) in data["off_req"]:
142     pref_terms.append(wgt * x[(emp, day, s)])
143 if L["14"] and pref_terms:
144     objB.append(L["14"] * sum(pref_terms))
145
146 # B5: equidad del turno mas penoso
147 if star is not None and len(S) > 1:
148     Nt = {n: model.NewIntVar(0, H, f"Nt_{n}") for n in N}
149     for n in N:
150         model.Add(Nt[n] == sum(x[(n, d, star)] for d in D))
151     Nmax = model.NewIntVar(0, H, "Nmax"); Nmin = model.
152         NewIntVar(0, H, "Nmin")
153     for n in N:
154         model.Add(Nmax >= Nt[n]); model.Add(Nmin <= Nt[n])
155     aux["Nmax"], aux["Nmin"] = Nmax, Nmin
156     if L["15"]:
157         objB.append(L["15"] * (Nmax - Nmin))
158 else:
159     aux["Nmax"], aux["Nmin"] = None, None
160
161 # B6: fines de semana completos
162 v = []
163 for n in N:
164     for wk, (sat, sun) in enumerate(WE):

```

```

162         if sun is None:
163             continue
164         vv = model.NewIntVar(0, 1, f"v_{n}_{wk}")
165         model.Add(vv >= works(n, sat) - works(n, sun))
166         model.Add(vv >= works(n, sun) - works(n, sat))
167         v.append(vv)
168     aux["v"] = v
169     if L["l6"]:
170         objB.append(L["l6"] * sum(v))
171
172     model.Minimize(sum(objA) + sum(objB))
173     aux["objA_terms"] = objA
174     return model, x, aux
175
176
177 def metrics(solver, data, x, aux, theta=3):
178     """Calcula Z_A e indicadores de bienestar en unidades naturales
179     ,
180     SIEMPRE a partir de la solucion (no de las auxiliares, que solo
181     se
182     fijan cuando su termino esta activo)."""
183     val = solver.Value
184     N = list(data["staff"].keys())
185     H = data["horizon"]; S = list(data["shifts"].keys())
186     WE = weekends(H)
187     star = aux["star"]
188
189     def works(n, d):
190         return sum(val(x[(n, d, s)]) for s in S)
191
192     # Z_A
193     za = 0; sat = 0; tot = 0
194     for (emp, day, s, wgt) in data["on_req"]:
195         tot += 1
196         if val(x[(emp, day, s)]) == 1: sat += 1
197         else: za += wgt
198     for (emp, day, s, wgt) in data["off_req"]:
199         tot += 1
200         if val(x[(emp, day, s)]) == 0: sat += 1
201         else: za += wgt
202     for (day, s), (req, wu, wo) in data["cover"].items():
203         za += wu * val(aux["u"][(day, s)]) + wo * val(aux["e"][(day
204         , s)])
205
206     Tn = {n: sum(works(n, d) for d in range(H)) for n in N}

```

```

204     gap_carga = max(Tn.values()) - min(Tn.values())
205
206     Fn = {}
207     for n in N:
208         c = 0
209         for (sd, ud) in WE:
210             wsat = works(n, sd)
211             wsun = works(n, ud) if ud is not None else 0
212             if wsat or wsun:
213                 c += 1
214         Fn[n] = c
215     gap_finde = max(Fn.values()) - min(Fn.values())
216
217     rachas = 0
218     for n in N:
219         for d in range(0, H - theta):
220             if all(works(n, d + j) == 1 for j in range(theta + 1)):
221                 rachas += 1
222
223     finc = 0
224     for n in N:
225         for (sd, ud) in WE:
226             if ud is None:
227                 continue
228             if works(n, sd) + works(n, ud) == 1:
229                 finc += 1
230
231     gap_pen = None
232     if star is not None and len(S) > 1:
233         Ntn = {n: sum(val(x[(n, d, star)])) for d in range(H) for n
234                in N}
235         gap_pen = max(Ntn.values()) - min(Ntn.values())
236
237     return {
238         "Z_A": za,
239         "gap_carga": gap_carga,
240         "gap_finde": gap_finde,
241         "rachas_largas": rachas,
242         "findes_incompletos": finc,
243         "satisf_pref": round(sat / tot, 3) if tot else None,
244         "gap_penoso": gap_pen,
245     }

```

B.3. Runner de experimentos (experiments.py)

```

1  """Runner de experimentos del Modelo B. Ejecuta por lotes (rango de
    indices)
2  y va anadiendo resultados a results.jsonl."""
3  import sys, json, os
4  from nrp_full import build_model, metrics
5  from nrp_model import parse_instance
6  from ortools.sat.python import cp_model
7
8  INSTANCES = [1, 2, 3]
9  TLIMIT = 30
10
11 # Niveles de peso calibrados a partir de las pruebas piloto
12 LEVELS = {
13     "11": [10, 50, 200], "12": [10, 50, 200], "15": [10, 50, 200],
14     "13": [5, 20, 50],   "16": [5, 20, 50],
15     "14": [1, 3, 10],
16 }
17 TERM_NAME = {"11": "carga", "12": "finde", "13": "rachas",
18             "14": "pref", "15": "penoso", "16": "findeinc"}
19 COMBINED = {
20     "moderado": {"11": 50, "12": 50, "13": 20, "14": 3, "15": 50, "
21                 16": 20},
22     "alto":     {"11": 200, "12": 200, "13": 50, "14": 10, "15":
23                 200, "16": 50},
24 }
25
26 def build_configs():
27     cfigs = []
28     # Baseline
29     for inst in INSTANCES:
30         cfigs.append({"inst": inst, "bloque": "baseline", "label": "
31                     baseline", "w": {}})
32     # Termino a termino
33     for inst in INSTANCES:
34         for term in ["11", "12", "13", "14", "15", "16"]:
35             for lvl in LEVELS[term]:
36                 cfigs.append({"inst": inst, "bloque": "termino",
37                             "label": f"{TERM_NAME[term]}={lvl}", "
38                             w": {term: lvl}})
39     # Combinados
40     for inst in INSTANCES:
41         for name, w in COMBINED.items():

```

```

39         cfgs.append({"inst": inst, "bloque": "combinado", "
40                     label": name, "w": dict(w)})
41     return cfgs
42
43 CONFIGS = build_configs()
44
45
46 def get_seed(inst):
47     """Solucion baseline (optimo de eficiencia) cacheada, para warm
48     start."""
49     path = f"seed_{inst}.json"
50     if os.path.exists(path):
51         return json.load(open(path))
52     data = parse_instance(f"instances/Instance{inst}.txt")
53     model, x, aux = build_model(data, w=None)
54     s = cp_model.CpSolver()
55     s.parameters.max_time_in_seconds = 30
56     s.parameters.num_search_workers = 8
57     s.parameters.random_seed = 42
58     s.Solve(model)
59     seed = {f"{n}|{d}|{sh}": int(s.Value(v)) for (n, d, sh), v in x
60            .items()}
61     json.dump(seed, open(path, "w"))
62     return seed
63
64 def run_one(cfg):
65     data = parse_instance(f"instances/Instance{cfg['inst']}.txt")
66     model, x, aux = build_model(data, w=cfg["w"])
67     if cfg["bloque"] != "baseline":
68         seed = get_seed(cfg["inst"])
69         for (n, d, sh), v in x.items():
70             model.AddHint(v, seed[f"{n}|{d}|{sh}"])
71     s = cp_model.CpSolver()
72     s.parameters.max_time_in_seconds = TLIMIT
73     s.parameters.num_search_workers = 8
74     s.parameters.random_seed = 42
75     st = s.Solve(model)
76     m = metrics(s, data, x, aux)
77     return {
78         "inst": cfg["inst"], "bloque": cfg["bloque"], "label": cfg[
79             "label"],
80         "w": cfg["w"], "estado": s.StatusName(st), "obj": s.
81             ObjectiveValue(),

```

```

79         "cota": s.BestObjectiveBound(), "tiempo": round(s.WallTime
80             (), 1), **m,
81     }
82
83 if __name__ == "__main__":
84     lo, hi = int(sys.argv[1]), int(sys.argv[2]) # rango [lo, hi)
85     with open("results.jsonl", "a") as f:
86         for i in range(lo, min(hi, len(CONFIGS))):
87             r = run_one(CONFIGS[i])
88             r["idx"] = i
89             f.write(json.dumps(r) + "\n")
90             f.flush()
91             print(f"[{i}] inst{r['inst']} {r['bloque']}/{r['label
92                 ']}: "
93                   f"obj={r['obj']} Z_A={r['Z_A']} carga={r['
94                       gap_carga']} "
95                   f"rachas={r['rachas_largas']} penoso={r['
96                       gap_penoso']} "
97                   f"findeinc={r['findes_incompletos']} pref={r['
98                       satisf_pref']} ({r['tiempo']}s)",
99                   flush=True)
100     print(f"TOTAL configs: {len(CONFIGS)}")

```

Anexo C. Calibración de los pesos de bienestar

Cada peso λ_k controla un término de bienestar del Modelo B. Como los seis términos miden magnitudes en escalas distintas, sus niveles se calibraron por separado mediante ejecuciones piloto sobre la instancia más rica en turnos, eligiendo entre el umbral en que el indicador empieza a moverse y aquel en que empieza a comprometer la cobertura. El umbral de confort de las rachas se fijó en $\theta = 3$.

Cuadro 8.2: Niveles de peso calibrados por término de bienestar.

Peso	Término de bienestar	Bajo	Medio	Alto
λ_1	B1. Equidad de carga	10	50	200
λ_2	B2. Equidad de fines de semana	10	50	200
λ_3	B3. Suavizado de rachas largas	5	20	50
λ_4	B4. Refuerzo de preferencias	1	3	10
λ_5	B5. Equidad del turno penoso	10	50	200
λ_6	B6. Fines de semana completos	5	20	50

Cuadro 8.3: Perfiles combinados (nivel medio y alto de cada término).

Perfil	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
Moderado	50	50	20	3	50	20
Alto	200	200	50	10	200	50

Anexo D. Resultados completos de los experimentos

Las tablas siguientes recogen las 63 ejecuciones del estudio, agrupadas por instancia. Para cada configuración se indica el valor de eficiencia Z_A , los indicadores de bienestar en unidades naturales (rachas largas, desigualdad de carga, desigualdad del turno penoso, fines de semana incompletos y tasa de satisfacción de preferencias), el estado del solver y el tiempo de cálculo en segundos. La abreviatura “n.a.” indica que el término no aplica (instancia con un solo tipo de turno).

Cuadro 8.4: Resultados completos. Instance1 (8 enfermeras, 1 turno).

Configuración	Z_A	Rachas	Carga	Penoso	F. inc.	Pref.	Estado	$t(s)$
baseline	607	7	2	n.a.	2	0.808	Ópt.	0.7
carga=10	607	8	2	n.a.	2	0.808	Ópt.	1.5
carga=50	607	8	2	n.a.	2	0.808	Ópt.	2.2
carga=200	716	5	0	n.a.	3	0.692	Ópt.	0.6
finde=10	607	8	2	n.a.	2	0.808	Ópt.	0.5
finde=50	607	7	2	n.a.	2	0.808	Ópt.	0.5
finde=200	607	10	2	n.a.	2	0.846	Ópt.	0.5
rachas=5	613	4	2	n.a.	0	0.731	Ópt.	1.3
rachas=20	613	4	2	n.a.	0	0.731	Ópt.	0.8
rachas=50	613	4	2	n.a.	0	0.731	Ópt.	0.3
pref=1	607	10	2	n.a.	2	0.846	Ópt.	0.6
pref=3	607	10	2	n.a.	2	0.846	Ópt.	0.5
pref=10	607	10	2	n.a.	2	0.846	Ópt.	0.3
penoso=10	607	7	2	n.a.	2	0.808	Ópt.	0.5
penoso=50	607	7	2	n.a.	2	0.808	Ópt.	0.5
penoso=200	607	7	2	n.a.	2	0.808	Ópt.	0.6
findeinc=5	608	6	2	n.a.	0	0.808	Ópt.	0.4
findeinc=20	608	6	2	n.a.	0	0.808	Ópt.	0.3
findeinc=50	608	6	2	n.a.	0	0.808	Ópt.	0.4
moderado	613	4	2	n.a.	0	0.731	Ópt.	2.6
alto	811	2	0	n.a.	1	0.769	Ópt.	1.0

Cuadro 8.5: Resultados completos. Instance2 (14 enfermeras, 2 turnos).

Configuración	Z_A	Rachas	Carga	Penoso	F. inc.	Pref.	Estado	$t(s)$
baseline	828	15	5	8	1	0.710	Fact.	30.0
carga=10	828	16	5	8	1	0.710	Fact.	30.0
carga=50	828	18	5	8	1	0.726	Fact.	30.0
carga=200	828	15	5	8	1	0.710	Fact.	30.0
finde=10	828	15	5	8	1	0.710	Fact.	30.0
finde=50	828	15	5	8	1	0.710	Fact.	30.0
finde=200	828	16	5	8	1	0.710	Fact.	30.0
rachas=5	835	10	5	8	1	0.661	Fact.	30.0
rachas=20	835	10	5	9	1	0.677	Fact.	30.0
rachas=50	934	5	5	8	2	0.661	Fact.	30.0
pref=1	828	15	5	8	1	0.710	Fact.	30.0
pref=3	828	15	5	8	1	0.710	Fact.	30.0
pref=10	828	15	5	8	1	0.710	Fact.	30.0
penoso=10	828	16	5	8	1	0.710	Fact.	30.0
penoso=50	828	15	5	8	1	0.710	Fact.	30.0
penoso=200	923	18	5	7	2	0.742	Fact.	30.0
findeinc=5	828	15	5	9	1	0.710	Fact.	30.0
findeinc=20	828	16	5	8	1	0.710	Fact.	30.0
findeinc=50	828	15	5	8	1	0.710	Fact.	30.0
moderado	835	10	5	8	1	0.677	Fact.	30.0
alto	1325	1	4	7	2	0.726	Fact.	30.0

Cuadro 8.6: Resultados completos. Instance3 (20 enfermeras, 3 turnos).

Configuración	Z_A	Rachas	Carga	Penoso	F. inc.	Pref.	Estado	$t(s)$
baseline	1001	25	5	5	0	0.984	Fact.	30.0
carga=10	1003	23	5	5	0	0.969	Fact.	30.0
carga=50	1003	22	5	5	0	0.969	Fact.	30.0
carga=200	1001	26	5	5	0	0.984	Fact.	30.0
finde=10	1001	26	5	5	0	0.984	Fact.	30.0
finde=50	1001	26	5	5	0	0.984	Fact.	30.0
finde=200	1003	22	5	5	0	0.969	Fact.	30.0
rachas=5	1005	19	5	5	0	0.938	Fact.	30.0
rachas=20	1023	17	5	5	0	0.844	Fact.	30.0
rachas=50	1219	12	5	5	2	0.859	Fact.	30.0
pref=1	1001	26	5	5	0	0.984	Fact.	30.0
pref=3	1003	23	5	5	0	0.969	Fact.	30.0
pref=10	1003	23	5	5	0	0.969	Fact.	30.0
penoso=10	1004	22	5	4	0	0.953	Fact.	30.0
penoso=50	1003	22	5	4	0	0.969	Fact.	30.0
penoso=200	1004	23	5	4	0	0.953	Fact.	30.0
findeinc=5	1001	28	5	5	0	0.984	Fact.	30.0
findeinc=20	1001	25	5	5	0	0.984	Fact.	30.0
findeinc=50	1001	26	5	5	0	0.984	Fact.	30.0
moderado	1010	19	5	4	0	0.906	Fact.	30.0
alto	1006	20	5	4	0	0.938	Fact.	30.0