

Máster en Ingeniería de Sistemas y Control



UNIVERSIDAD  
**COMPLUTENSE**  
MADRID

---

---

# Gestión logística optimizada de la recolección de bolsas de hemodonación

---

---

**Trabajo de Fin de Máster**  
**Curso 2022–2023**

**Autor**

Eduardo Abreu Pessoa de Mendonça

**Directores**

José Luis Risco Martín  
Josué Pagán Ortiz

Junio de 2023



# Gestión logística optimizada de la recolección de bolsas de hemodonación

Trabajo de Fin de Máster en  
Ingeniería de Sistemas y Control

**Autor**

Eduardo Abreu Pessoa de Mendonça

**Directores**

José Luis Risco Martín  
Josué Pagán Ortiz

Facultad de Informática  
Universidad Complutense de Madrid

09 de junio de 2023



# Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

*Dedicado a mi hermana Letícia, que con coraje  
abrazó su vocación de salvar vidas.*

# Agradecimientos

Quiero dar mi agradecimiento a José Luis Risco Martín, por presentarme ese apasionante proyecto, por sus buenas ideas y por el tiempo que me ha dedicado.

Agradezco también a Josué Pagán Ortiz, por compartir su conocimiento y por sus innumerables aportaciones para mejorar este trabajo.

Debo agradecer a Eva Besada Portas, por “conectar los puntos” y por mostrarme el camino.

Por último, agradezco al equipo del Centro Regional de Hemodonación de Murcia por proveer la información que permitió la elaboración de este trabajo.

# Resumen

## Gestión logística optimizada de la recolección de bolsas de hemodonación

La disponibilidad de sangre y sus componentes es crucial para garantizar una pronta respuesta a los tratamientos médicos que pueden salvar vidas. Las donaciones son actualmente la única forma de obtener sangre humana, pero existen desafíos para obtener donaciones frecuentes y los componentes sanguíneos tienen una vida útil limitada. La recolección de bolsas de donación de sangre es una actividad importante en la cadena de suministro de sangre y está lejos de ser trivial, ya que la variabilidad y la aleatoriedad son inherentes al problema. El objetivo de este trabajo es definir un modelo matemático e implementar una herramienta basada en optimización que considera mejores pronósticos de demanda para apoyar la gestión logística de la recolección de sangre del Centro Regional de Donación de Sangre de la Región de Murcia. Se proponen tres modelos de programación lineal entera mixta (MILP) y se realizan posteriormente pruebas exhaustivas para comprobar la funcionalidad. El modelo inicial se amplió para detectar posibles no factibilidades y también para incluir las existencias de sangre en el centro. El optimizador puede ayudar al planificador a crear planes de recolección óptimos y es una base sólida para futuras mejoras de la herramienta.

## Palabras clave

Optimización, OPL, CPLEX, sangre, donación, hemodonación, planificación, colectas, modelo matemático, programación lineal entera mixta



# Abstract

## Optimized logistics management of bloodmobile collection

The availability of blood and its components is crucial to guarantee prompt response to life-saving medical treatments. Donations are currently the only way to obtain human blood, but there are challenges to obtain frequent donations and blood components have a limited shelf life. The collection of blood donation bags is an important activity in the blood supply chain and is far from trivial, as variability and randomness are inherent to the problem. The goal of this work is to define a mathematical model and implement an optimization based decision-making tool that considers improved demand forecasts to support the logistics management of blood collection of Murcia's Regional Blood Donation Center. Three mixed integer linear programming (MILP) models are proposed, and extensive testing was conducted to test functionality. The initial model was extended to detect potential infeasibilities and to include blood inventory at the center. The optimizer can assist the planner to build optimal collection plans and is a solid foundation for further improvements.

## Keywords

Optimization, OPL, CPLEX, blood, donation, planning, collections, mathematical model, mixed integer linear programming

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Gestión logística en hemodonación . . . . .	1
1.2. Motivación . . . . .	4
1.3. Objetivos . . . . .	5
1.4. Estructura del documento . . . . .	5
1.5. Código del proyecto . . . . .	5
<b>2. Estado de la cuestión</b>	<b>6</b>
2.1. Revisión de la literatura . . . . .	6
2.2. Contribuciones del trabajo . . . . .	7
<b>3. Metodología</b>	<b>9</b>
3.1. Herramientas . . . . .	9
3.2. Datos de entrada y salida . . . . .	11
3.2.1. Datos de entrada . . . . .	11
3.2.2. Datos de salida . . . . .	12
3.3. Análisis de los datos de entrada . . . . .	13
3.3.1. Frecuencias de visitas . . . . .	13
3.3.2. Estimación de la demanda . . . . .	15
3.3.3. <i>Stock</i> en el centro . . . . .	16
3.3.4. Costes de recolección . . . . .	16
<b>4. Modelado</b>	<b>18</b>
4.1. Descripción general del problema . . . . .	18
4.2. Modelo inicial . . . . .	20
4.2.1. Conjuntos . . . . .	20
4.2.2. Variables de decisión . . . . .	20
4.2.3. Variables auxiliares . . . . .	21
4.2.4. Parámetros del problema . . . . .	21
4.2.5. Función objetivo . . . . .	22
4.2.6. Restricciones . . . . .	22
4.3. Modelo con <i>stock</i> . . . . .	24
4.3.1. Variables auxiliares . . . . .	24
4.3.2. Parámetros . . . . .	24
4.3.3. Función objetivo . . . . .	24
4.3.4. Restricciones . . . . .	24
4.4. Diagnóstico de no factibilidad . . . . .	25
4.4.1. Variables auxiliares . . . . .	26
4.4.2. Parámetro . . . . .	26
4.4.3. Restricción Suave . . . . .	26
4.4.4. Función objetivo . . . . .	26

<b>5. Experimentos</b>	<b>27</b>
5.1. Datos de referencia . . . . .	27
5.2. Experimentos . . . . .	28
5.2.1. Experimento con visitas libres . . . . .	29
5.2.2. Experimento con bloqueo de visitas . . . . .	31
5.2.3. Experimento con optimización de 6 meses . . . . .	32
5.2.4. Experimento con modelo con <i>stock</i> . . . . .	34
<b>6. Conclusiones y Trabajo Futuro</b>	<b>36</b>
<b>7. Introduction</b>	<b>38</b>
<b>8. Conclusions and Future Work</b>	<b>43</b>
<b>Bibliografía</b>	<b>45</b>
<b>Apéndice</b>	<b>46</b>
<b>A. Modelo OPL</b>	<b>47</b>

# Índice de figuras

1.1. Red de centros de transfusión en España. . . . .	2
1.2. Procesos en la cadena transfusional. . . . .	2
1.3. Donaciones por año en Murcia. . . . .	3
1.4. Visión general del sistema de planificación. . . . .	4
3.1. Tablas de entrada y sus relaciones. . . . .	12
3.2. Tablas de salida. . . . .	13
3.3. Cantidad de puntos de extracción según la frecuencia de visita. . . . .	14
3.4. Ejemplo de frecuencia: “Periódica $4/4 + 1/2$ ”. . . . .	15
3.5. Demanda diaria en el año 2018 y promedio calculado. . . . .	15
3.6. Estimación de salidas . . . . .	16
3.7. Costes de personal según módulos y previsión de donaciones. . . . .	17
4.1. Proceso general de recolección de bolsas . . . . .	18
4.2. Ubicación del centro y de los puntos de extracción en Murcia. . . . .	19
5.1. Cantidad de visitas por día y turno en el histórico . . . . .	28
5.2. Resultado del experimento con visitas libres . . . . .	30
5.3. Mensaje de diagnóstico de no factibilidad. . . . .	30
5.4. Página del informe: necesidad de personal . . . . .	31
5.5. Página del informe: visitas por turno . . . . .	32
5.6. Resultados del experimento con visitas bloqueadas . . . . .	33
5.7. Resultados del experimento con <i>stock</i> . . . . .	35
5.8. Fichero con mensajes del posprocesado. . . . .	35

# Índice de tablas

3.1. Tablas de entrada. . . . .	11
5.1. Resumen del histórico del mes de enero de 2018. . . . .	27
5.2. Desglose del histórico por tipología de puntos visitados. . . . .	28
5.3. Resumen del plan optimizado . . . . .	29
5.4. Desglose del plan optimizado por tipología de puntos visitados. . . . .	31
5.5. Resultados para 6 meses consecutivos . . . . .	33
5.6. Análisis de cadencia . . . . .	34

# Lista de Acrónimos

<b>API</b>	Interfaz de programación de aplicaciones, del inglés <i>Application Programming Interface</i>
<b>CRHM</b>	Centro regional de hemodonación de Murcia
<b>CSV</b>	Valores separados por comas, del inglés <i>Comma-Separated Values</i>
<b>CYFSE</b>	Cuerpos y fuerzas de seguridad del Estado
<b>DES</b>	Simulación de eventos discretos, del inglés <i>Discrete Event Simulation</i>
<b>IDE</b>	Entorno de desarrollo integrado, del inglés <i>Integrated Development Environment</i>
<b>MILP</b>	Programación lineal entera mixta, del inglés <i>Mixed Integer Linear Programming</i>
<b>OPL</b>	Lenguaje de programación de optimización, del inglés <i>Optimization Programming Language</i>

# Capítulo 1

## Introducción

### 1.1. Gestión logística en hemodonación

La sangre y sus componentes — como los hematíes, los leucocitos, las plaquetas y el plasma —, se utilizan en una amplia gama de tratamientos médicos. Una transfusión puede ser necesaria, por ejemplo, cuando un paciente pierde mucha sangre debido a una lesión, cuando un paciente es sometido a determinadas intervenciones quirúrgicas, o cuando una persona tiene algún trastorno sanguíneo hereditario como la talasemia o un trastorno hemorrágico causado por problemas con factores de coagulación.

Cuando se piensa en sangre, se suele pensar en la sangre total, pero los hemoderivados son también muy importantes para la industria farmacéutica. Las terapias derivadas del plasma son tratamientos que pueden salvar la vida de pacientes con enfermedades poco frecuentes y complejas. Existen muchos medicamentos cuyo principio activo proviene del plasma de donantes humanos sanos a través de un proceso de fraccionamiento y purificación adecuado, tales como los siguientes[1]:

- Albúmina humana;
- Inmunoglobulina humana;
- Complejo de factor VIII de la coagulación y factor von Willebrand humanos;
- Factor IX de la coagulación;
- Inmunoglobulina humana;
- Concentrado de alfa-1-antitripsina (AAT);
- Antitrombina humana de alta pureza.

Es fácil notar que es de vital importancia que esos productos médicos estén prontamente disponibles para garantizar la actividad asistencial. Actualmente las donaciones son la única manera de obtener sangre humana, por lo que el correcto suministro de sangre es vital ya que se requieren donaciones frecuentes y los componentes sanguíneos tienen un tiempo de almacenamiento limitado.

En España, los Centros de Transfusión (CT) son los responsables de garantizar la suficiencia en componentes sanguíneos. Estos centros deben conocer en todo momento las reservas existentes y actuar para evitar posibles situaciones de escasez. Para eso ellos mantienen comunicación regular con otros centros, con los servicios

de transfusión hospitalarios y además promueven regularmente campañas para incentivar la donación.

Cada una de las comunidades autónomas tiene uno o más de estos centros, formando la red del Sistema Transfusional en España presentada en la figura 1.1 [2].



Figura 1.1: Red de centros de transfusión en España.  
(reproducido de: Ministerio de Sanidad [2])

La labor del CT es importante dentro de la cadena transfusional (ver figura 1.2), con actividades que van desde la selección del donante hasta la extracción, procesamiento, almacenamiento, preparación y distribución de componentes [3].

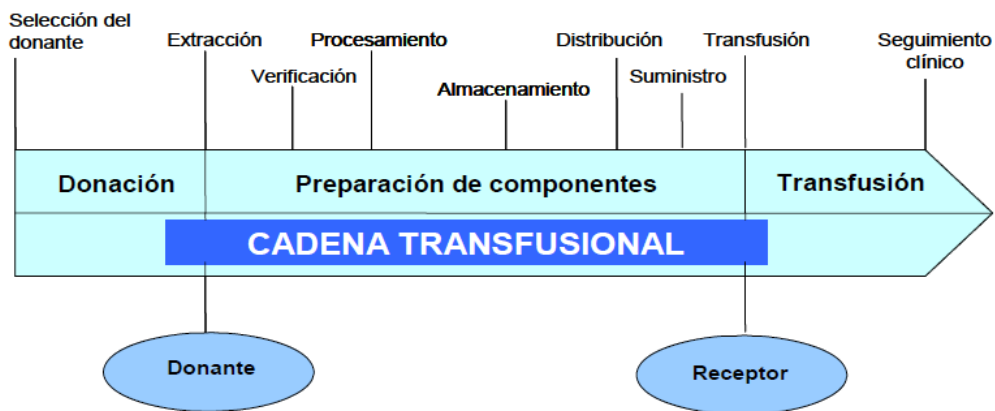


Figura 1.2: Procesos en la cadena transfusional.  
(reproducido de: Ministerio de Sanidad [3])



El centro de interés en ese trabajo es el Centro Regional de Hemodonación de Murcia (CRHM), responsable por la cobertura de las necesidades de donación de toda esta comunidad autónoma. En 2020 el CRHM gestionó un total de 46.978 donaciones de sangre total [4], dato este que es inferior a la cantidad de los años anteriores [5, 6], como se puede observar en la figura 1.3.

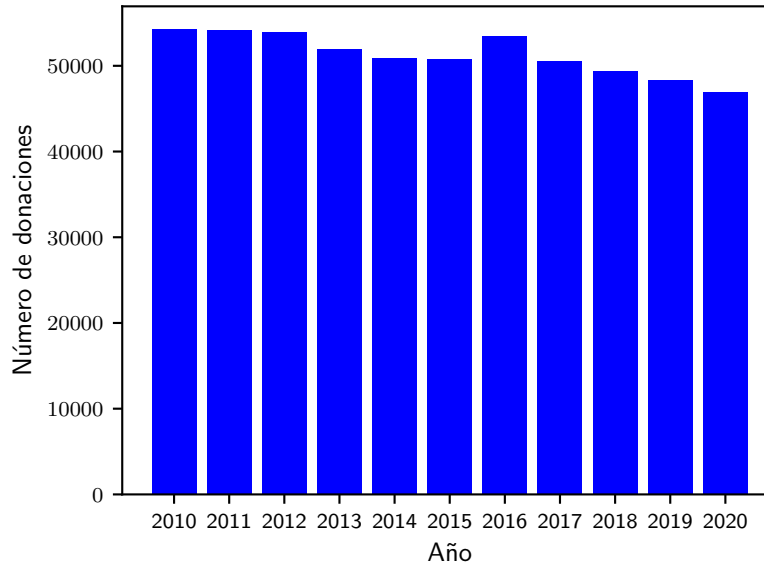


Figura 1.3: Donaciones por año en Murcia.  
Adaptado de: Centro de Hemodonación [7].

Esa reducción se debe al impacto de la pandemia de la COVID-19 en todo el país. Felizmente no hubo mayores consecuencias, pues afortunadamente los centros han podido responder a las demandas hospitalarias, ya que también hubo una reducción de la demanda en ese período [7]. Lo que se busca mostrar con ese dato es que la cantidad de donaciones es variable, y depende de múltiples factores sociales que influyen en el comportamiento de los donantes.

La **gestión logística de la recolección de sangre** es complicada y la tarea de planificación no es trivial. Como se puede imaginar, hay una variabilidad que es inherente al problema y son muchas las restricciones que deben ser llevadas en consideración para construir un plan factible, tales como la frecuencia de visita a cada punto de extracción, los horarios de funcionamiento de cada punto, la capacidad de los vehículos, y otras restricciones operativas que se juzguen necesarias.

Por todo lo expuesto, se cree que la mejor manera de elaborar el plan de recolección para el centro es utilizando una solución informática, con el fin de reducir el tiempo de planificación y aumentar la calidad del plan, mejorando así el aprovechamiento de los recursos disponibles. Lo que se busca en este trabajo es desarrollar una herramienta que auxilie en la tarea de definir un plan óptimo para la recolección de bolsas de sangre en los puntos de extracción con el objetivo de mantener el *stock* del CRHM en un nivel deseado. En este trabajo se presenta un modelo de programación lineal entera mixta para cumplir ese objetivo.

## 1.2. Motivación

Actualmente la planificación en el CRHM se realiza en base a la experiencia del profesional responsable, pero sin ninguna herramienta específica para ayudarlo. Por ese motivo, el profesional puede necesitar hasta algunas semanas para elaborar el plan de recolección de bolsas de sangre. Este trabajo es parte de un proyecto más amplio elaborado junto al CRHM para desarrollar un sistema en línea que permita optimizar los procesos de la cadena transfusional y asegurar el autoabastecimiento, reduciendo las alertas por insuficiencia de sangre de algún tipo. El sistema reduciría el tiempo de planificación sin reemplazar a ningún trabajador. Esta herramienta ayudaría a la toma de decisiones, liberando al planificador un tiempo valioso para centrarse en aspectos más estratégicos de su trabajo.

El proyecto abarca múltiples actividades para el desarrollo del sistema propuesto, que se esquematiza en la figura 1.4, tales como la definición de una base de datos, el desarrollo de un módulo para preprocesar los datos, el desarrollo del optimizador, además del tratamiento de los resultados y puesta en marcha de toda la interfaz gráfica para interacción con el usuario final. La aportación de este trabajo está centrada principalmente en el desarrollo del modelo matemático para el optimizador, que es el núcleo del sistema.

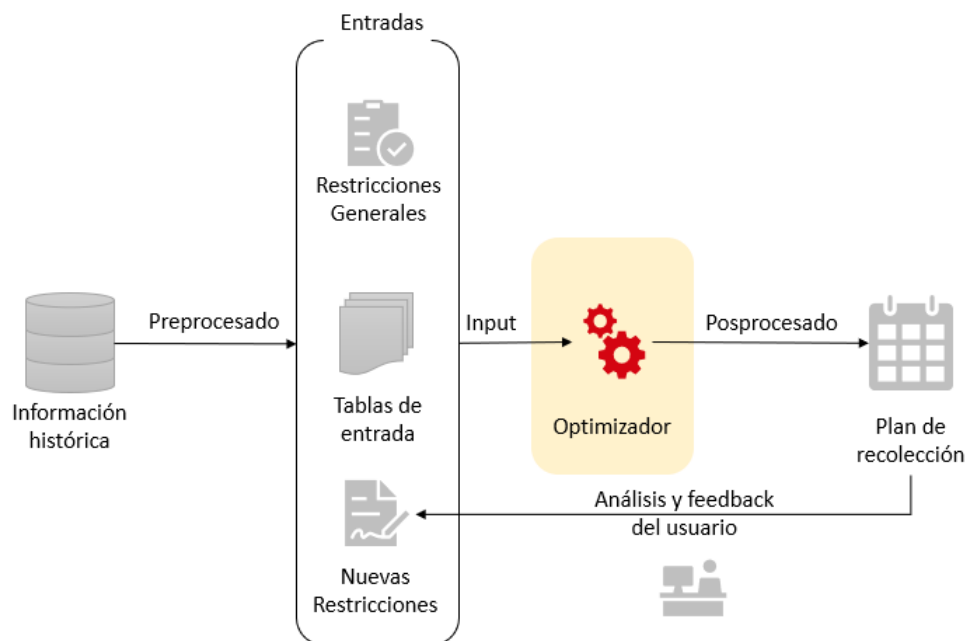


Figura 1.4: Visión general del sistema de planificación.

### 1.3. Objetivos

La colaboración con el CRHM empezó hace algún tiempo y desde entonces el grupo de investigación pudo avanzar con un trabajo previo de recopilación de datos y de mapeo de las restricciones necesarias para el modelo. Esa colaboración incluso ya generó la publicación de un trabajo académico en años anteriores. Este trabajo es una continuación de [8], definiendo un modelo matemático más genérico y abstracto, y con previsiones de demanda mejoradas. En particular, se definen los siguientes objetivos específicos:

- Desarrollar un modelo matemático abstracto para la planificación mensual de visitas a puntos de extracción.
- Tener en consideración en el modelo que el operador de la herramienta pueda incluir nuevas restricciones que posibiliten la optimización iterativa de la solución.
- Realizar experimentos con los datos disponibles que demuestren el correcto funcionamiento de la herramienta.
- Hacer una comparativa entre los datos de planificación reales y los resultados obtenidos con el optimizador.
- Desarrollar un modelo matemático de depuración para ayudar al usuario final a diagnosticar problemas en los datos que puedan tornar el problema no factible.

### 1.4. Estructura del documento

Se presenta a continuación la estructura del TFM.

- En el Capítulo 2, se muestra el estado del arte de la planificación de recolección de bolsas de sangre.
- En el Capítulo 3, un breve resumen de los datos de entrada es presentado junto con la descripción de las herramientas utilizadas.
- El Capítulo 4 elabora los distintos modelos matemáticos desarrollados.
- En el Capítulo 5 se muestran los resultados obtenidos con los modelos desarrollados, haciendo una comparativa con la planificación realizada por el CRHM entre distintos casos de uso.
- Por último, en el Capítulo 6 se presentan las conclusiones y las sugerencias para trabajos futuros.

### 1.5. Código del proyecto

El código desarrollado en este trabajo y un extracto de los datos de entrada y salida de los experimentos realizados se encuentran accesibles bajo demanda en el siguiente enlace:

[https://github.com/greenlsi/hemoglobulab\\_opt](https://github.com/greenlsi/hemoglobulab_opt)

# Capítulo 2

## Estado de la cuestión

Este capítulo muestra cómo ha sido tratado el tema en publicaciones científicas y lista las aportaciones de este trabajo.

### 2.1. Revisión de la literatura

La cadena de suministro de sangre humana es compleja, pero parece tener similitudes en la manera que es gestionada en distintos países. En la literatura encontramos descripciones de redes muy similares a la de España, o sea, redes en que el territorio es subdividido y cada región tiene uno o más centros responsables de las actividades de hemodonación y de producción y reparto de hemoderivados en su área de influencia, como es el caso por ejemplo de la red de Francia [9] o la de Colombia [10].

A grandes rasgos, la cadena transfusional puede ser dividida en 4 eslabones: colecta, producción, stock y distribución. En general las publicaciones tratan de una única parte de la cadena y muy pocas trabajan con un modelo integrado [11]. Un estudio de 2012 muestra que el principal enfoque es la gestión de *stock*, mientras que la planificación de colectas estaba en último lugar en términos de cantidad de publicaciones ([12], como se citó en [9]). Este dato es corroborado por otro estudio del 2017, donde solamente 11 % de un total de 218 artículos disponibles en la literatura tratan de las colectas [13].

Incluso dentro de un mismo tópico, el enfoque de las publicaciones varía, pues algunas desarrollan modelos más estratégicos, mientras otras optan por un enfoque más operativo. Según la clasificación de Osorio, Brailsford y Smith (2015) [11], usualmente las siguientes decisiones son tomadas en cada nivel para la etapa de colecta:

- Estratégico: ubicación de los puntos de colecta, definición de la capacidad y del personal necesario a largo plazo.
- Táctico: definición de políticas de atención a donantes, planificación de campañas de donación y asignación de recursos a medio plazo.
- Operativo: programación de colectas y otras decisiones diarias tales como rutas para el transporte, manera de extraer la sangre, y tipo de bolsas a usar para cada donante, etc.

La metodología utilizada también es variada. Algunas publicaciones trabajan con modelos de simulación de eventos discretos (DES, del inglés *Discrete Event*

*Simulation*) [10, 14], otras con teoría de colas [15], mientras otras se enfocan en modelos de predicción [16] o incluso en el diseño organizacional [17].

Teniendo en cuenta que las publicaciones son muy diversas, se seleccionaron 3 artículos que son más cercanos al planteamiento de este trabajo:

1. El artículo de Alfonso *et al* [9] desarrolla 3 modelos de programación entera mixta para la colecta: dos modelos para la planificación anual, siendo uno para frecuencias fijas de visitas y otro para frecuencias variables, y uno para planificación semanal. Estos modelos se ejecutan de manera secuencial, o sea, los resultados del modelo anual alimentan el modelo semanal, que decide la fecha exacta de visita y los recursos humanos necesarios.
2. El artículo de Osorio *et al* [10] desarrolla un modelo integrado de colecta y producción que tiene en cuenta detalles adicionales interesantes, como por ejemplo: los diferentes tipos sanguíneos, las cantidades de hemoderivados que pueden ser obtenidos a partir de una bolsa de sangre, y penalizaciones para productos expirados o desabastecimientos.
3. Por último, otro artículo de Osorio *et al* [18] desarrolla un modelo determinista de optimización multiobjetivo que busca minimizar el coste de colecta y la cantidad necesaria de donantes. Una de las variables del modelo es la cantidad de máquinas necesarias de cada tipo, pues la extracción puede ser directamente de la sangre total o de componentes por separado por aféresis, que requiere una máquina especial que separa el componente en concreto y devuelve los demás al torrente sanguíneo del donante.

Ninguno de esos planteamientos es directamente aplicable a la realidad del CRHM, necesitando un desarrollo para que el modelo pueda ser utilizado en la planificación de colectas en Murcia. Por citar algunos ejemplos de diferencias entre estos modelos en la literatura y la realidad del CRHM, el coste de colecta es considerado en la función objetivo en muchos de estos modelos, pero es considerado un objetivo secundario para el CRHM. También la transferencia de material entre centros no se aplica al caso de Murcia y el CRHM trabaja con un horizonte de planificación mensual.

Las soluciones de estos trabajos son rígidas en el sentido de que son puramente prescriptivas y no consideran la intervención humana. Este enfoque no es realista, pues hay mucha variabilidad en todo el proceso, como por ejemplo los frecuentes cambios en la disponibilidad de los puntos de extracción, que pueden impactar tanto la cantidad de donantes cuanto las fechas de las siguientes visitas para mantenerse la cadencia. Esto significa que la actividad requiere un sistema iterativo donde el responsable pueda proporcionar su juicio y experiencia para conseguir resultados más fiables.

Por último, vale recalcar que el óptimo que se busca en el CRHM no es el óptimo global para el mes en el sentido estrictamente matemático, sino la solución más factible considerando las múltiples restricciones adicionales que la realidad impone al problema en el día a día.

## 2.2. Contribuciones del trabajo

Por todo lo expuesto en el apartado anterior, la aportación del trabajo es la de desarrollar un modelo adaptado a la realidad del CRHM con los siguientes puntos:

- Construir un modelo de planificación a lo largo de un año con discretización mensual.
- Incluir el *stock* del centro en el modelo de planificación.
- Considerar el bucle de retroalimentación, permitiendo al responsable de la planificación hacer cambios (por ejemplo, fijando fechas de visitas a determinados puntos de extracción, prohibiendo ciertas variantes, intensificando la demanda, etc.).
- Incluir la posibilidad de trabajar con restricciones suaves para que el modelo no se vuelva no factible, lo que deriva en un modelo alternativo para depurar restricciones.
- Integrar el modelo en un sistema de control en lazo cerrado, de forma que el responsable de la planificación pueda repetir optimizaciones hasta encontrar la configuración adecuada. Porque la ventana de optimización es de un mes, el modelo está pensado para trabajar con él a diario, ya que el procesado de datos está optimizado para que la ejecución no dure más de un minuto.
- Presentar las soluciones al usuario de manera amigable, incluyendo el cálculo de costes.

# Capítulo 3

## Metodología

En este capítulo se listan las herramientas utilizadas y se describen los datos de entrada y salida del modelo. En el último apartado se presenta cómo los datos originales fueron trabajados para generar los datos de entrada del modelo.

### 3.1. Herramientas

#### CPLEX

El optimizador elegido para el presente estudio fue CPLEX de la empresa IBM. El *ILOG CPLEX Optimization Studio* es un paquete de software que está disponible comercialmente y es capaz de leer los datos de un problema iterativamente desde ficheros en formatos específicos, resolver el problema y devolver la solución en una interfaz gráfica o en ficheros de texto. Entre los problemas que puede resolver están los de programación lineal entera mixta (MILP, del inglés *Mixed Integer Linear Programming*), que es justamente el tipo al que pertenece la formulación del problema de la recolección de bolsas de sangre del presente trabajo.

#### OPL

OPL (del inglés *Optimization Programming Language*) es el lenguaje de IBM usado en CPLEX Studio para modelar problemas de optimización. Este lenguaje permite declarar las variables de decisión y expresar las restricciones y la función objetivo de una manera intuitiva para el usuario, pues la sintaxis se aproxima a las expresiones matemáticas necesarias para describir este tipo de problema.

CPLEX también provee extensiones a OPL que permiten hacer tareas que no tienen que ver con el modelado propiamente dicho, o sea, tareas tales como preprocesado, pos-procesado y control del flujo, en conjunto llamadas en inglés de *scripting*. Estas extensiones también han sido usadas en este trabajo para, leer los datos, manipular las soluciones y cambiar parámetros del optimizador cuando es aplicable, respectivamente.

#### Python

Los ficheros recibidos del CRHM se procesan con un algoritmo desarrollado en Python para crear los datos de entrada del optimizador. Se realizan cálculos estadísticos de la cantidad de bolsas de sangre para recolección en cada punto de extracción basado en el histórico de las donaciones pasadas. Además de calcular el

promedio, desviación estándar, máximos y mínimos para la cantidad de bolsas, el algoritmo también considera la posibilidad de mover las donaciones a fechas distintas a las pasadas, adoptando un intervalo que da un cierto margen para que se realice la recolección, lo que permite al optimizador obtener soluciones realistas.

## Power BI

Los resultados obtenidos son presentados al usuario final y comparados con los datos reales para cada periodo en un informe desarrollado en MS Power BI. Eso no quiere decir que la herramienta que el usuario final tenga que utilizar sea necesariamente Power BI, pues resultados parecidos pueden ser obtenidos con otras herramientas que permitan construir un cuadro de mando con las métricas de la solución. Se puede usar incluso MS Excel o alternativas como Google Sheets, o de software libre y gratuito como LibreOffice. Sin embargo, se sugiere una herramienta de BI (del inglés *Business Intelligence*) para que el usuario pueda trabajar con visualizaciones interactivas y así poder filtrar y analizar mejor los datos como desee. Alternativas conocidas de herramientas de BI son Tableau, Qlickview y Tibco Jaspersoft, pero se puede incorporar a la página web del sistema incluso un informe personalizado desarrollado en Python con marcos de trabajo (del inglés *framework*) como Plotly Dash o Streamlit.



## 3.2. Datos de entrada y salida

### 3.2.1. Datos de entrada

Los datos de entrada del problema fueron estandarizados en ocho tablas distintas, como listado en la tabla 3.1.

#	Tabla	Descripción
1	<i>Structural</i>	Lista todos los puntos de extracción y sus características (dirección, tipo, prioridad, etc.).
2	<i>Bloodmobiles</i>	Lista las unidades móviles, que son cada uno de los vehículos de los que dispone el centro, y sus características (capacidades mínima y máxima de bolsas).
3	<i>Statistics</i>	Contiene la información estadística de la cantidad de bolsas que pueden ser recogidas en cada punto y en cada día de la semana para un mes dado.
4	<i>Visits</i>	Almacena la información de la periodicidad de los puntos de extracción, bien como del número de visitas que se ha de realizar en un mes dado cuando se elige un determinado punto.
5	<i>Alpha</i>	Tabla que permite al usuario definir una visita de una unidad móvil a un determinado punto de extracción en un determinado día y turno. Sirve como restricción para el modelo.
6	<i>Shift</i>	Lista todos los turnos de funcionamiento de cada punto de extracción: mañana, tarde o jornada completa.
7	<i>Demand</i>	Tabla que lista los días laborables de cada semana en un determinado mes y permite informar la demanda diaria de bolsas del centro. Contiene campos adicionales que permiten controlar la cantidad de visitas. Además, permite informar el <i>stock</i> objetivo diario en el centro.
8	<i>Parameters</i>	Tabla auxiliar que almacena el mes y año, además de informar si se trata de un mes lectivo y cuál es el valor objetivo mínimo del indicador de visitas a centros educativos. Contiene campos adicionales aplicables al modelo con <i>stock</i> que permiten configurar los parámetros deseados para control de la cantidad de bolsas almacenadas en el centro.

Tabla 3.1: Tablas de entrada.

Los campos de año y mes en la tabla *Parameters* podrían ser eliminados ya que OPL permite trabajar con fechas directamente, pero se optó por simplificar el código y usar solamente la información del día en las demás tablas.

Aplicando el concepto de base de datos relacional, tenemos la siguiente vista de las tablas y algunas de sus relaciones en la figura 3.1:

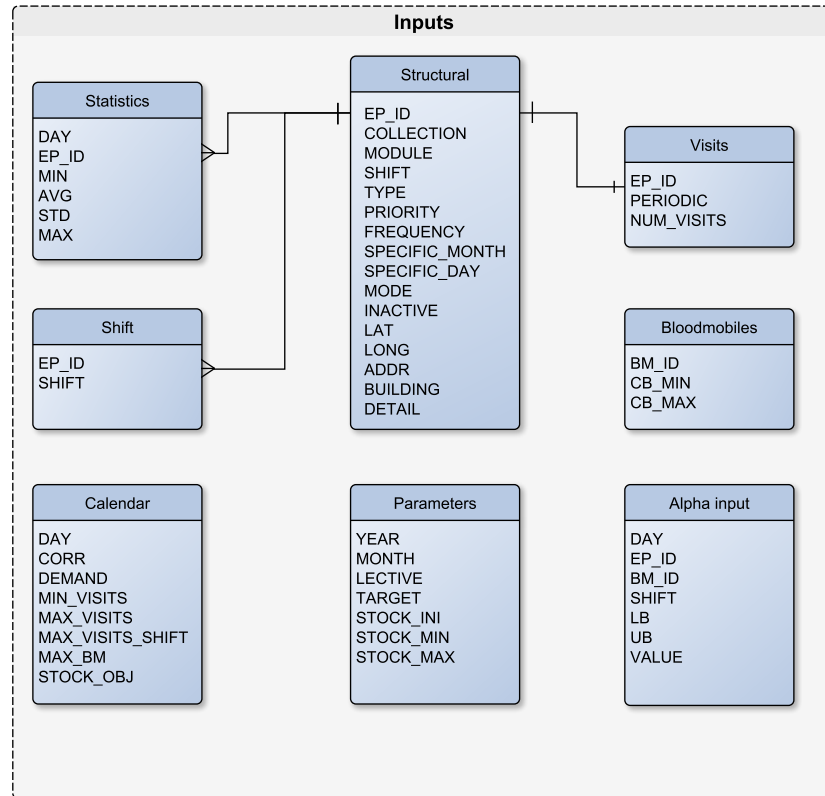


Figura 3.1: Tablas de entrada y sus relaciones.

Las relaciones entre las tablas *Statistics* y *Structural*, y entre entre las tablas *Shift* y *Structural* son del tipo “1:N”, pues un mismo punto puede tener un promedio distinto de donaciones cada día y puede estar disponible para recolección en más de un turno. La relación entre las tablas *Visits* y *Structural* es del tipo “1:1”, pues se presenta al optimizador la cantidad total de visitas al punto para un determinado mes. Se opta en este trabajo por considerar que las unidades móviles están disponibles todos los días, pero se puede análogamente añadir una tabla adicional para controlar la disponibilidad de las unidades móviles cada día y turno.

### 3.2.2. Datos de salida

Los principales datos de salida se almacenan en un conjunto de ficheros CSV con los valores obtenidos por el optimizador para cada variable del problema, como presentado en la figura 3.2.

De manera resumida, tenemos entonces los siguientes datos en cada una de las tablas:

- *Alpha*: puntos de extracción visitados cada día, turno y por cual vehículo.
- *Beta*: cantidad diaria de bolsas recogidas por vehículo.
- *Gamma*: cantidad diaria de bolsas por punto de extracción, con desglose por turno y vehículo.

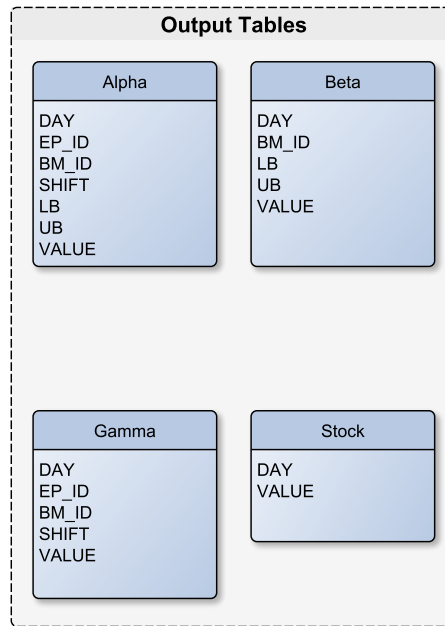


Figura 3.2: Tablas de salida.

- *Stock*: cantidad diaria de bolsas almacenadas en el centro.

Estos ficheros alimentan un informe desarrollado en Power BI para presentar los resultados de una manera más amigable al usuario. Este informe contiene también los cálculos de los costes de la solución obtenida.

Por último, un fichero en formato texto es generado cada vez que se ejecuta una optimización para guardar los mensajes del optimizador y mostrarlos al usuario. Este fichero es generado en la etapa de posprocesado con el resumen de los datos de la solución obtenida.

### 3.3. Análisis de los datos de entrada

Los datos en crudo son complejos de procesar, porque vienen en el formato del centro y no están pensados para usarlos en un algoritmo de optimización. Para dar una idea de la complejidad del análisis se esbozan algunos ejemplos del procesamiento de los datos de entrada.

#### 3.3.1. Frecuencias de visitas

La frecuencia de visita es un dato importante para la planificación de las recolecciones, pues los donantes necesitan un mínimo de dos meses entre donaciones y en un periodo de doce meses no pueden donar más de tres veces las mujeres ni más de cuatro veces los hombres [19]. Se suele tener en cuenta donantes de sangre voluntarios habituales, que son fidelizados y no remunerados, una vez que es beneficioso para la seguridad de la sangre y para control de los costes de captación, ya que un donante fidelizado cuesta menos que captar un donante nuevo [20].

La frecuencia de visita, la oferta diaria de cada punto de extracción y también la demanda diaria estimada del CRHM fueron obtenidas de un análisis estadístico de los datos históricos proporcionados para los años de 2007 hasta 2019.

Primeramente, se eliminaron los puntos que se encuentran inactivos y también algunos pocos puntos que son visitados diariamente. Los demás fueron clasificados según la frecuencia de visitas, como se puede observar en la figura 3.3.

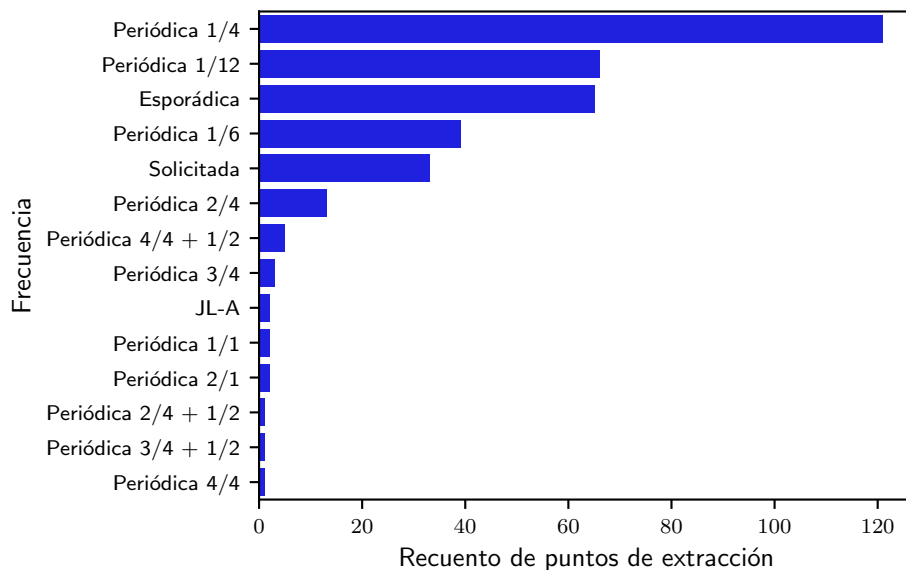


Figura 3.3: Cantidad de puntos de extracción según la frecuencia de visita.

Algunos locales se visitan solamente de manera esporádica, mientras otros son visitados con una periodicidad determinada. En el caso de las visitas periódicas, la codificación está basada en dos números, siendo el primero la cantidad mensual de visitas y el segundo el número de meses para que se repita el patrón. Así, “periódica 1/4” significa que se debe visitar el punto 1 vez al mes cada 4 meses.

Solamente por poner un ejemplo de un caso quizás más complicado, la clasificación “periódica 4/4 + 1/2” presentada en la figura 3.4 sería la de un punto con periodos compuestos, donde además de ser visitado 4 veces al mes cada 4 meses, también se visita 1 vez al mes cada 2 meses.

Cabe recalcar que el código desarrollado para la etapa de preprocesado también considera cierta flexibilidad con relación a la semana específica para la realización de una visita. Sería demasiado estricto y poco realista considerar siempre exactamente las mismas semanas para visitar un determinado punto, pues incluso en el histórico se nota que este no es el caso, habiendo una considerable variabilidad a lo largo de los años. Se optó entonces por considerar un margen que nos permite movernos algunas semanas por delante o por detrás para identificar las semanas candidatas, pero siempre respetando la cadencia, o sea, la periodicidad.

Este mismo concepto se utiliza para calcular la cantidad de bolsas que se espera recoger en un punto. Lo que se hizo fue establecer bloques que dividen el año proporcionalmente, las llamadas “semanas de influencia”, y hacer el cálculo del promedio de bolsas recolectadas utilizando estos intervalos.

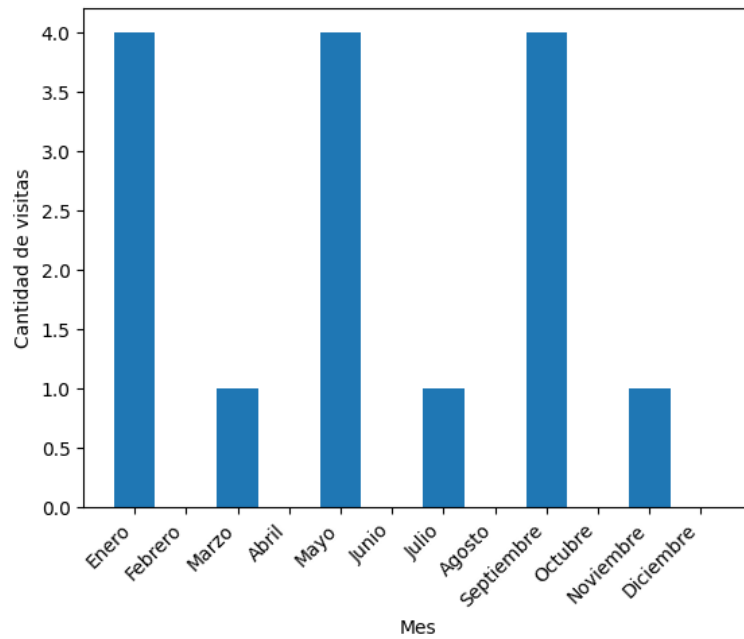


Figura 3.4: Ejemplo de frecuencia: “Periódica  $4/4 + 1/2$ ”.

### 3.3.2. Estimación de la demanda

La demanda diaria del centro es un dato de entrada que debe ser informado por el usuario. En este trabajo la demanda fue calculada a partir del histórico, más específicamente del año 2018. Como se indica en la figura 3.5, el promedio es de  $165 \pm 43$  bolsas de sangre al día considerando solamente los días laborables y sin considerar los puntos de extracción fijos.

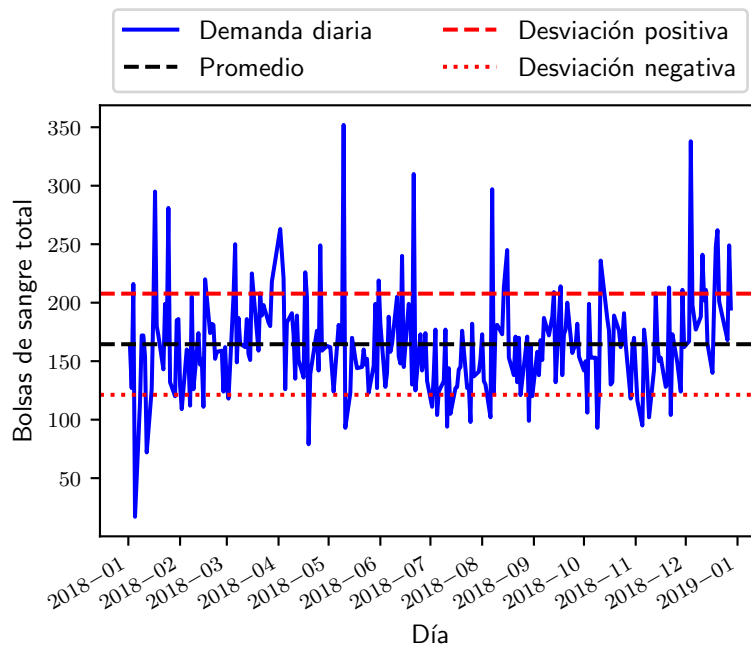


Figura 3.5: Demanda diaria en el año 2018 y promedio calculado.

### 3.3.3. *Stock* en el centro

El *stock* en el centro no es un dato que está inmediatamente disponible. El CRHM compartió datos de la demanda por tipo de componente para el período entre enero de 2007 y noviembre de 2016, o sea, las salidas del centro. Sin embargo, el proceso de transformar las bolsas de sangre total en componentes puede tardar hasta 18 horas y la cantidad de productos sanguíneos obtenidos puede variar, además de poder haber pérdidas por caducidad. De esta manera, se opta por hacer una estimación sencilla basada en los datos de salidas de hematíes del año de 2016, considerando que se necesita de una bolsa de sangre total para producir una bolsa de hematíes y calculando el promedio para los días laborables, ya que la demanda en los fines de semana es insignificante comparada con la demanda durante los días de semana.

Los resultados obtenidos están en la figura 3.6, donde se puede verificar que el promedio de  $181 \pm 56$  bolsas de hematíes al día se encuentra en un intervalo parecido al promedio de entradas de sangre total en el centro en 2018.

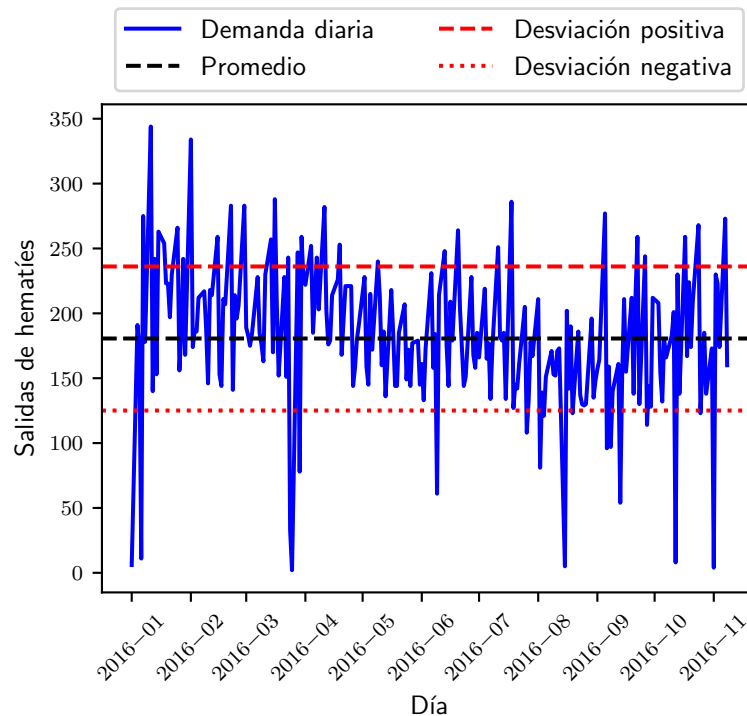


Figura 3.6: Estimación de salidas diarias en 11 meses del año 2016.

De esta manera, para los experimentos de este trabajo se define la demanda del centro con valores próximos a los promedios calculados y adicionalmente se define de manera arbitraria un *stock* inicial y un *stock* máximo en el centro sin pérdida de generalidad.

### 3.3.4. Costes de recolección

En este trabajo, el coste de cada colecta depende de tres factores: de la distancia al punto de extracción, del avituallamiento o refrigerio, que es el alimento que se le da a los donantes para reparar las fuerzas después de una donación, y del equipo de trabajo necesario.

El coste de avituallamiento puede ser considerado trivial, pues se puede asumir un coste simbólico de 1 euro por bolsa donada. El coste asociado al transporte también puede ser simplificado con un coste unitario por kilómetro para cada tipo de vehículo en base al consumo de combustible de cada uno, pues todas las distancias a los puntos son conocidas o pueden ser fácilmente obtenidas.

El coste del equipo de trabajo es un poco más elaborado y se calcula en base a la cantidad de profesionales de cuatro categorías distintas (enfermeros, administrativos, médicos y conductores) que son necesarios para atender la previsión de extracción de bolsas. Esta necesidad está tabulada, bien como los costes unitarios de cada categoría profesional, que varían según el módulo al que el punto a ser visitado pertenece. Los módulos se detallan más adelante, pero son clasificaciones predominantemente geográficas que implican en costes de personal mayores para puntos de extracción más lejanos. Cruzando estas dos tablas compartidas por el CHRM, se obtienen cuatro funciones lineales por partes, como se muestra en la figura 3.7

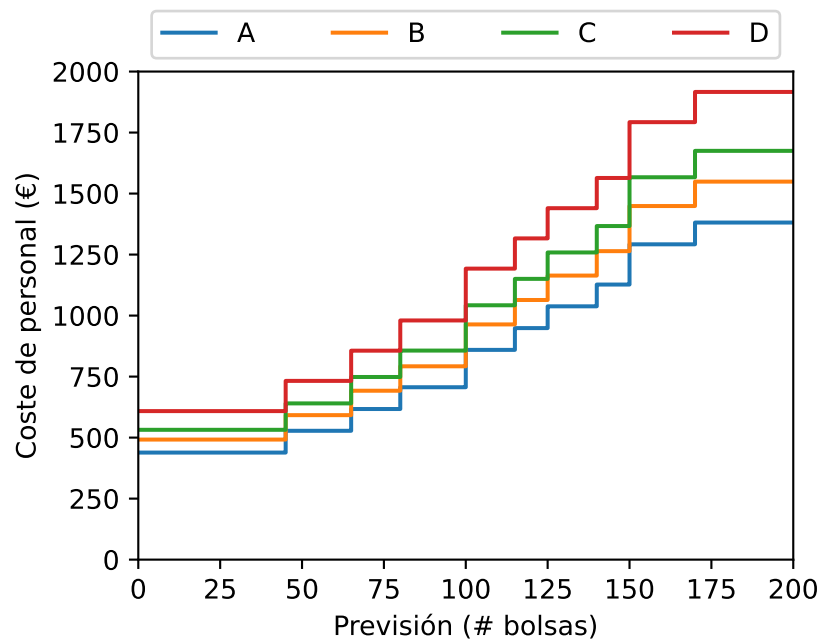


Figura 3.7: Costes de personal según módulos y previsión de donaciones.

Estas funciones fueron incluidas en el modelo solamente para el calculo de métricas que dan al operador más información sobre los resultados, pero en un futuro los costes también pueden ser considerados en una optimización multiobjetivo.

# Capítulo 4

## Modelado

### 4.1. Descripción general del problema

El problema es definir un plan óptimo de recolección de bolsas de sangre en los puntos de extracción para mantener el *stock* de un centro regional de donación en un nivel deseado, como se muestra en la figura 4.1.



Figura 4.1: Proceso general de recolección de bolsas de sangre.  
(creada en la versión educativa de icograms.com)

Con este sencillo enunciado, podemos ver que el correcto modelado del problema necesita cubrir las características de las siguientes entidades y sus interrelaciones:

- Producto: bolsas de sangre,
- Proveedores: puntos de extracción, donde se realizan las colectas de donantes de sangre,
- Destino: centro regional de donación, donde se almacenan las bolsas para posterior envío a los hospitales en los cuales sea necesaria una transfusión sanguínea,
- Medio de transporte: por carretera con una flota de vehículos dedicados que empiezan las rutas de recogida siempre en el CRHM.

Inicialmente el modelo no tiene en cuenta el tipo sanguíneo de cada bolsa de sangre. Este es un dato importante, pero actualmente no está disponible de inmediato, por lo que se decidió no incluirlo en el modelo. El problema actual informado por



el CRHM es la falta de sangre en general, independiente del tipo, por lo que nuestro sistema puede ser usado de inmediato y esta información puede ser incorporada añadiendo más restricciones en el futuro.

En la figura 4.2 se representan los puntos de extracción. Son alrededor de 360 y se encuentran dispersos en toda la provincia de Murcia. Están divididos en 4 zonas geográficas, llamadas módulos, que son codificadas por las letras “A” (en azul), “B” (en verde), “C” (en naranja) y “D” (en rojo). Son de distintos tipos, tales como centros de salud, centros educativos, empresas, etc., y pueden ser puntos de donación fijos o móviles. Es importante recalcar que para los puntos móviles se tiene que coordinar que un equipo de trabajo compuesto de profesionales sanitarios vaya al destino, lo que conlleva costes adicionales. Obviamente existen en cada uno de esos puntos restricciones en cuanto al horario de funcionamiento (e.g. turnos) y frecuencia de visitas (e.g. se colecta con periodicidad definida o bajo demanda).

El centro regional está ubicado en la ciudad de Murcia. Es importante mencionar que dentro del centro se procesan las bolsas de sangre para garantizar la calidad del material y también para obtener subproductos como hematíes, plaquetas, plasma, etc. Los procesos internos en el centro no están en el alcance del modelo presentado en ese trabajo.

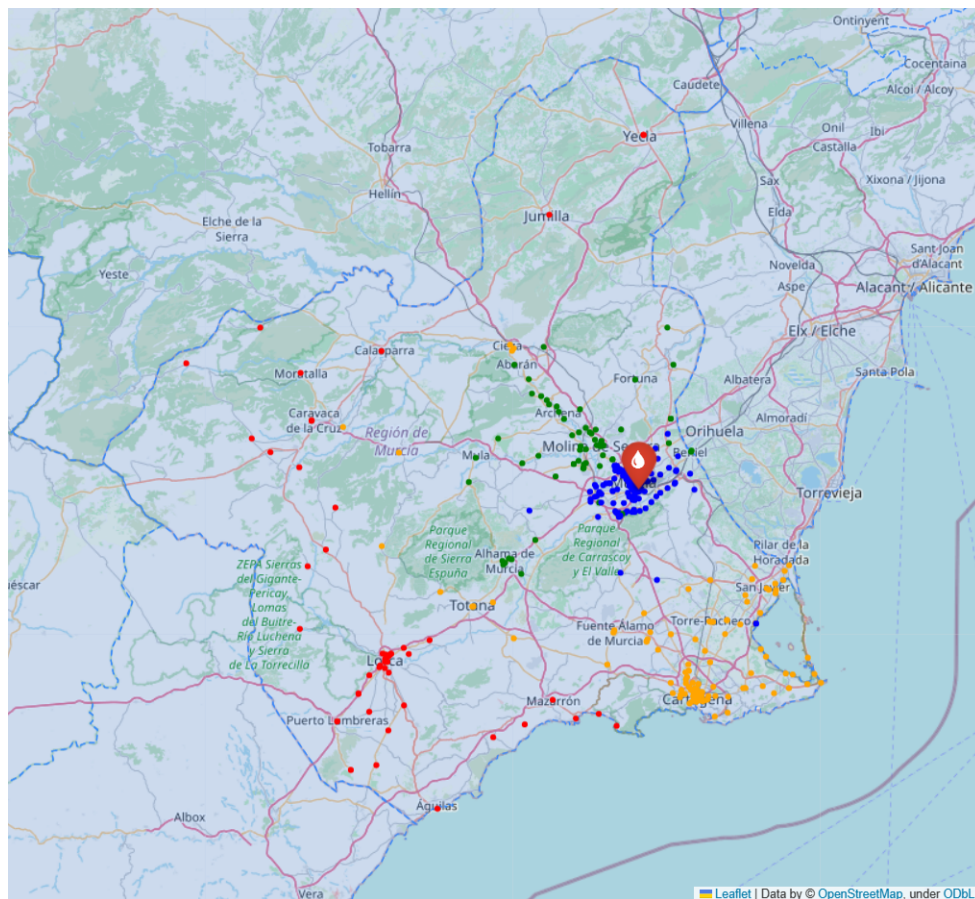


Figura 4.2: Ubicación del centro y de los puntos de extracción en Murcia. Centro en rojo y puntos de extracción en círculos con distintos colores que indican el módulo al que pertenecen (A - azul, B - verde, C - naranja y D - rojo).

El centro dispone de cuatro vehículos para hacer las recogidas. Estos vehículos poseen distintas capacidades y están debidamente equipados para controlar la temperatura y mantener la integridad del producto hasta el destino final.

El horizonte de planificación de este trabajo es mensual y se dispone de la información de los festivos que afectan a cada una de las localidades. Esa información es importante, pues se debe compensar la fluctuación en el stock regional aumentando la actividad de recolección en las vísperas de los festivos.

A continuación se presentan 3 modelos matemáticos desarrollados en este trabajo. En primer lugar, un modelo denominado de “Inicial”. Este modelo se encarga de considerar todas las restricciones generales requeridas en la planificación. En segundo lugar, un modelo denominado de “*Stock*” que altera el modelo inicial e incluye restricciones adicionales relacionadas con la existencia de bolsas en el centro. Finalmente, un modelo denominado “Diagnóstico de no factibilidad”, que incluye variables adicionales para detección de restricciones que son violadas.

Dividimos la presentación de los modelos en 6 partes: los conjuntos que se usan como índices, las variables de decisión, las variables auxiliares, los parámetros del problema, la función objetivo y las restricciones.

## 4.2. Modelo inicial

### 4.2.1. Conjuntos

A continuación están listados los conjuntos utilizados en el modelo.

- D: conjunto de días laborables del horizonte de planificación.

$$D = \{d_1, d_2, \dots, d_n\}, d_i \in \{1, 2, \dots, 31\}$$

- E: conjunto de puntos de extracción.

$$E = \{e_1, e_2, \dots, e_n\}$$

- V: conjunto de vehículos, también llamados unidades móviles.

$$V = \{Rápida, Vito, DTN, FXT\}$$

- S: conjunto de turnos de colecta - mañana, tarde o todo el día.

$$S = \{M, T, 2\}$$

A lo largo de este documento, se adopta la convención de utilizar la letra minúscula que corresponda para los índices de los elementos de cada conjunto, o sea:

$$d \in D, e \in E, v \in V, s \in S$$

### 4.2.2. Variables de decisión

- $\alpha$ : variable binaria que toma el valor 1 para representar que una unidad móvil visita un punto de extracción en un determinado día y turno.

$$\alpha_{d,e,v,s} \in [0, 1]$$

- $\gamma$ : variable real que representa la cantidad de bolsas recogida por cada vehículo en un día y turno en cada punto de extracción.

$$\gamma_{d,e,v,s} \in \mathbb{R}^+$$

### 4.2.3. Variables auxiliares

- $\tau$ : cantidad total de bolsas recolectadas diariamente

$$\tau_d \in \mathbb{R}^+$$

- $\beta$ : representa la cantidad diaria de bolsas de sangre recolectadas por cada vehículo.

$$\beta_{d,v} \in \mathbb{R}^+$$

### 4.2.4. Parámetros del problema

- $\eta$ : necesidad diaria de bolsas en el centro.

$$\eta_d \in \mathbb{R}^+$$

- $\mu$ : promedio de bolsas recolectadas en un punto de extracción en un día según el estudio estadístico.

$$\mu_{d,e} \in \mathbb{R}^+$$

- $\sigma$ : desviación estándar de la cantidad de bolsas recolectadas en un punto de entrega en un día según el estudio estadístico.

$$\sigma_{d,e} \in \mathbb{R}^+$$

- $Q^{min}$ : cantidad mínima de bolsas que un vehículo puede transportar.

$$Q_v^{Min} \in \mathbb{Z}^+$$

- $Q^{max}$ : cantidad máxima de bolsas que un vehículo puede transportar.

$$Q_v^{Max} \in \mathbb{Z}^+$$

- $N$ : número de visitas para el punto de extracción en el período.

$$N_e \in \mathbb{Z}^+$$

- $\pi$ : porcentaje mínimo de visitas a centros educativos en el mes.

$$\pi = \begin{cases} 10\% & , \text{ si es un mes lectivo} \\ 5\% & , \text{ si el mes no es lectivo} \end{cases}$$

Los meses que no son considerados lectivos en ese trabajo son: junio, julio, agosto y septiembre.

- $P$ : prioridad asignada a un punto de extracción, que depende del tipo del punto (*p. ej.* hospitales, escuelas, empresas o cuerpos y fuerzas de seguridad del estado).

$$P_e = \begin{cases} 0 & , \text{ si el punto es un centro de salud} \\ 1 & , \text{ si el punto es un centro educativo} \\ 2 & , \text{ si el punto es una empresa o CYFSE} \end{cases}$$

- $F$ : indica si un punto se colecta con periodicidad definida o bajo demanda.

$$F_e = \begin{cases} 1 & , \text{ si el punto tiene una periodicidad de visitas definida} \\ 0 & , \text{ si se visita un punto solamente bajo demanda} \end{cases}$$

### 4.2.5. Función objetivo

Minimizar la diferencia entre las bolsas recogidas y la necesidad diaria.

$$\min \sum_d |\eta_d - \tau_d| \quad (4.1)$$

A pesar del valor absoluto ser una función no lineal, CPLEX es capaz de transformar el problema en lineal, por lo que no se presenta en este trabajo ninguna sustitución de variables para mantener la linealidad del modelo.

### 4.2.6. Restricciones

- Definición de la variable auxiliar  $\tau$ : es la cantidad total recolectada cada día por todos los vehículos en todos los puntos de extracción y turnos.

$$\tau_d = \sum_e \sum_v \sum_s \gamma_{d,e,v,s}, \quad \forall d \in D \quad (4.2)$$

- Definición de la variable  $\beta$ : es la suma de la cantidad de bolsas recolectadas por un vehículo en un día determinado.

$$\beta_{d,v} = \sum_e \sum_s \gamma_{d,e,v,s}, \quad \forall d \in D, \forall v \in V \quad (4.3)$$

- Cantidad de bolsas: lo que se recoge en un punto de extracción tiene que estar dentro del intervalo estadístico de los datos históricos. No se recoge nada si un punto no es visitado.

$$\alpha_{d,e,v,s}(\mu_{d,e} - \sigma_{d,e}) \leq \gamma_{d,e,v,s} \leq \alpha_{d,e,v,s}(\mu_{d,e} + \sigma_{d,e}) \quad (4.4)$$

- Capacidad de los vehículos: cada unidad móvil deberá llevar, como poco, el mínimo y, como mucho, el máximo de su capacidad.

$$Q_v^{\min} \leq \beta_{d,v} \leq Q_v^{\max} \quad (4.5)$$

- Cantidad de visitas por punto: los puntos de extracción que tienen visitas periódicas deben recibir exactamente la cantidad de visitas definida para el período, mientras que los demás reciben, como mucho, la cantidad definida.

$$\sum_d \sum_v \sum_s \alpha_{d,e,v,s} = N_e, \quad \forall e \in E : F_e = 1 \quad (4.6)$$

$$\sum_d \sum_v \sum_s \alpha_{d,e,v,s} \leq N_e, \quad \forall e \in E : F_e = 0 \quad (4.7)$$

- Cantidad de visitas por vehículo: las unidades móviles van, como mucho, a un punto de extracción por día.

$$\sum_e \sum_s \alpha_{d,e,v,s} \leq 1, \quad \forall d \in D, \forall v \in V \quad (4.8)$$

- Cantidad de visitas diarias: cada día se deben visitar al menos dos y como mucho  $|V| - 1$  puntos de extracción diferentes.

$$2 \leq \sum_e \sum_v \sum_s \alpha_{d,e,v,s} \leq |V| - 1, \quad \forall d \in D \quad (4.9)$$

- Porcentaje de visitas a escuelas: los centros educativos son considerados prioritarios, entonces la cantidad de visitas a ellos tiene que ser mayor o igual que el porcentaje objetivo fijado para el mes.

$$\frac{\sum_d \sum_x \sum_v \sum_s \alpha_{d,e,v,s}}{\sum_d \sum_e \sum_v \sum_s \alpha_{d,e,v,s}} \geq \pi, \quad (4.10)$$

$$x = \{e \in E : P_e = 1\}$$

- Visitas matinales: para cada día laborable debe haber por lo menos 2 salidas en el turno de la mañana.

$$\sum_e \sum_v \alpha_{d,e,v,s} \geq 2, \quad \forall d \in D, s \in S : s = "M" \quad (4.11)$$

- Visitas diarias por punto de extracción: cada punto debe ser visitado, como mucho, una vez y por un único vehículo en un día.

$$\sum_v \sum_s \alpha_{d,e,v,s} \leq 1, \quad \forall d \in D, \forall e \in E \quad (4.12)$$

Es importante recalcar que fue posible eliminar algunas restricciones del modelo directamente en la etapa de preprocesado. Actualmente se trabaja solamente en días laborables, por ejemplo, por lo tanto el modelo considera que el conjunto  $D$  contiene solamente los días laborables, no siendo necesario incluir una restricción para forzar que  $\alpha = 0$  en los festivos y finales de semana. Eso reduce las dimensiones del problema y hace con que el tiempo de ejecución sea menor.

Otro punto importante es que el modelo considera que la necesidad diaria de bolsas en el centro  $\eta$  ya considera el ajuste mencionado para las vísperas de los festivos, además de restar la cantidad relacionada a los puntos de extracción fijos, pues el objetivo es optimizar la recolección en los puntos de extracción móviles a los que se acude.

### 4.3. Modelo con *stock*

Las variables y restricciones son las mismas presentadas anteriormente en el apartado 4.2. La diferencia en ese modelo es que se incluyen nuevas variables, parámetros y restricciones relacionadas al *stock* en el centro y también se cambia la función objetivo, como se muestra a continuación.

#### 4.3.1. Variables auxiliares

- $I$ : cantidad total de bolsas en *stock* en el centro en un determinado día.

$$I_d \in \mathbb{R}^+$$

Otras variables que influyen el *stock* pueden ser añadidas en caso de ser necesario, por ejemplo para incorporar al modelo cambios como transferencias de bolsas para otros centros o pérdidas de *stock* por caducidad. Estas posibilidades se considerarán parte del desarrollo en un trabajo futuro.

#### 4.3.2. Parámetros

- $I_0$ : cantidad inicial de bolsas en el *stock* del centro.

$$I_0 \in \mathbb{R}^+$$

- $I^{min}$ : *stock* de seguridad definido para el centro.

$$I^{min} \in \mathbb{R}^+$$

- $I^{max}$ : *stock* máximo definido para el centro.

$$I^{max} \in \mathbb{R}^+$$

- $I^{obj}$ : *stock* de referencia diario definido por el centro.

$$I_d^{obj} \in \mathbb{R}^+$$

#### 4.3.3. Función objetivo

Minimizar la diferencia entre el *stock* diario y el *stock* objetivo.

$$\min \sum_d |I_d - I_d^{obj}| \quad (4.13)$$

#### 4.3.4. Restricciones

- Definición del *stock* inicial.

$$I_1 = I_0 \quad (4.14)$$

- Balance de flujo.

$$I_{d+1} = I_d + \tau_d - \eta_d, \quad \forall d \in D \quad (4.15)$$

- El *stock* diario tiene que estar dentro de los límites definidos.

$$I^{Min} \leq I_d \leq I^{Max}, \quad \forall d \in D \quad (4.16)$$

Vale recalcar que la variable  $\eta$  en este modelo ya no corresponde a la necesidad del centro, sino a las salidas del *stock*.

## 4.4. Diagnóstico de no factibilidad

Hay múltiples maneras de detectar cuando una restricción es violada. CPLEX posee opciones de calcular qué restricciones están en conflicto y relajarlas de forma que se obtenga un modelo factible. Otra manera es añadir variables adicionales binarias al problema, una para cada restricción que se desee detectar, y aplicar una relación lógica tal como la ecuación (4.17), en que  $y = 1$  si la restricción genérica  $\sum_j a_j x_j \leq b$  no se cumple. Es posible linealizar este tipo de restricción fácilmente, pero CPLEX también acepta relaciones lógicas como esta directamente en el modelo.

$$\sum_j a_j x_j > b \implies y = 1, \forall y \in \mathbb{B} \quad (4.17)$$

Sin embargo, se adopta otra estrategia en este trabajo, que es la de alterar el modelo para que sea siempre factible y que eventuales violaciones sean informadas al usuario en la etapa de posprocesado. Se considera importante para una adopción temprana del sistema que el usuario no reciba muchos mensajes de errores, a no ser que sean errores críticos. Así, las violaciones que se busca detectar son aquellas de las restricciones para las cuales hay casos en los que es aceptable que no se cumplan.

La alteración necesaria para transformar un modelo no factible en factible es la introducción de variables adicionales positivas de holgura o de exceso (*slack* o *surplus* en inglés, respectivamente). Se suele usar esta técnica en programación lineal para transformar las restricciones de desigualdad en restricciones de igualdad, pero lo que se busca aquí es mantener las desigualdades añadiendo una variable artificial que puede tomar valores positivos para que el problema vuelva a ser factible. Para que el valor de estas variables sea lo más próximo posible a cero, se añaden estas variables también a la función objetivo con una penalización para que sean minimizadas.

De esta manera, se trabaja con algunas restricciones blandas (*soft constraints* en inglés) que pueden ser violadas, mientras las demás son estrictas y permanecen inalteradas (*hard constraints* en inglés). Con el resultado de la optimización en manos, se puede fácilmente detectar las violaciones mirando los valores de las variables artificiales del modelo, pues siempre que haya alguna con valor positivo es porque no se respetó la restricción correspondiente.

Para dar un ejemplo práctico, algunas restricciones en este modelo pueden hacer que el modelo sea no factible para meses específicos a depender de cómo los datos de entrada son generados. Para contornar este problema, se opta por transformar algunas restricciones específicas del modelo en restricciones blandas. El razonamiento fue el siguiente:

- Puede haber días en que la demanda no sea suficiente como para justificar el mínimo de 2 visitas a puntos de extracción durante el día o en un determinado turno. Tratamos esta posibilidad añadiendo variables adicionales a las restricciones (4.9) y (4.11).
- Por otro lado, puede que la cantidad deseada de visitas a puntos clasificados como periódicos sea demasiado alta, provocando un conflicto con la restricción que limita la cantidad máxima permitida de visitas al día. Para tratar esta posibilidad, se permite que parte de las visitas a los puntos periódicos no sea realizada utilizando una variable adicional en la restricción (4.6).
- Por último, la restricción (4.10) también fue transformada en blanda, pues puede que no haya visitas suficientes a escuelas para cumplir con el porcentaje

mínimo deseado. Cuando eso ocurre, lo que se busca es por lo menos se aproximar al máximo del objetivo una vez que se minimiza la variable adicional correspondiente en la función objetivo.

A continuación se presentan las alteraciones necesarias en el modelo para el caso de la restricción (4.9). Los demás cambios siguen la misma lógica, pero no serán detallados por una cuestión de brevedad.

#### 4.4.1. Variables auxiliares

- $\delta$ : cantidad ficticia de visitas al día.

$$\delta_d \in \mathbb{R}^+$$

#### 4.4.2. Parámetro

- $Z$ : penalización por infringir la restricción blanda de la cantidad diaria de visitas.

$$Z \in \mathbb{R}^+$$

#### 4.4.3. Restricción Suave

Cantidad de visitas diarias: cada día se deben visitar al menos 2 puntos de extracción diferentes.

$$\sum_e \sum_v \sum_s \alpha_{d,e,v,s} + \delta_d \geq 2, \quad \forall d \in D \quad (4.18)$$

#### 4.4.4. Función objetivo

Minimizar la diferencia entre el *stock* diario y el *stock* objetivo, pero con un término adicional para la penalización.

$$\min \sum_d |I_d - I_d^{Obj}| + Z \sum_d \delta_d \quad (4.19)$$

Vale recalcar que los pesos correspondientes a cada variable artificial pueden ser distintos. En este trabajo se optó por aplicar un peso único a todas las variables artificiales.



# Capítulo 5

## Experimentos

En este capítulo se muestran los resultados obtenidos con los modelos desarrollados. Cada modelo es aplicado a un caso de uso específico con el objetivo de demostrar que el optimizador cumple con los requisitos. Como un recordatorio, se listan de manera resumida los objetivos enumerados en el Capítulo 2: construir un modelo de planificación mensual con o sin *stock* que considere retroalimentación y que, además de presentar los resultados de manera amigable al usuario, pueda también generar alertas caso una instancia sea no factible.

Se detallan los datos de referencia en la primera sección para luego presentar los resultados de cada experimento realizado en la sección siguiente.

### 5.1. Datos de referencia

Se elige el mes de enero de 2018 como referencia para todos los experimentos por ser un mes representativo en cantidad de visitas y donaciones. Primeramente, se analiza la solución del histórico y, como se puede observar en la tabla 5.1, un total de 51 puntos de extracción distintos fueron visitados con un total de 65 visitas en el mes. La mayoría de las visitas ocurren en puntos periódicos y notamos también que se visitó un punto no-periódico dos veces en enero. El promedio es de 165,3 bolsas cada día laborable.

Periódico	# Puntos	# Visitas	# Bolsas
Sí	49	62	3.586
No	2	3	51
<b>Total</b>	<b>51</b>	<b>65</b>	<b>3.637</b>

Tabla 5.1: Resumen del histórico del mes de enero de 2018.

En la tabla 5.2 se presenta el desglose por tipo de punto de extracción. La mayoría de los puntos visitados son centros de salud (prioridad 0). Enero es un mes lectivo y la proporción de visitas a centros educativos (prioridad 1) es de 13,85 %, cumpliendo con el mínimo deseado.

Por último, se presenta en la figura 5.1 cómo se realizaron las visitas a lo largo del mes en los distintos turnos. Claramente no se cumplió con el mínimo de dos visitas en el turno de la mañana, tampoco con realizar al menos dos y como mucho tres visitas al día. Esto nos permite mostrar que algunas de las restricciones del modelo muchas veces no son respetadas en la práctica. Esto refuerza que algunas

Tipo	# Puntos	# Visitas	# Bolsas
<b>Prioridad 0</b>	<b>35</b>	<b>47</b>	<b>2.870</b>
CS Municipio	34	46	2.853
Hospitales	1	1	17
<b>Prioridad 1</b>	<b>8</b>	<b>9</b>	<b>469</b>
IES-FP	6	7	266
Universidad	2	2	203
<b>Prioridad 2</b>	<b>8</b>	<b>9</b>	<b>298</b>
Colectivos	1	1	63
CYFSE	1	1	68
Empresas	6	7	167
<b>Total</b>	<b>51</b>	<b>65</b>	<b>3.637</b>

Tabla 5.2: Desglose del histórico por tipología de puntos visitados.

restricciones se deben considerar blandas y se espera que el optimizador indique al usuario cuándo se infringe alguna de ellas.

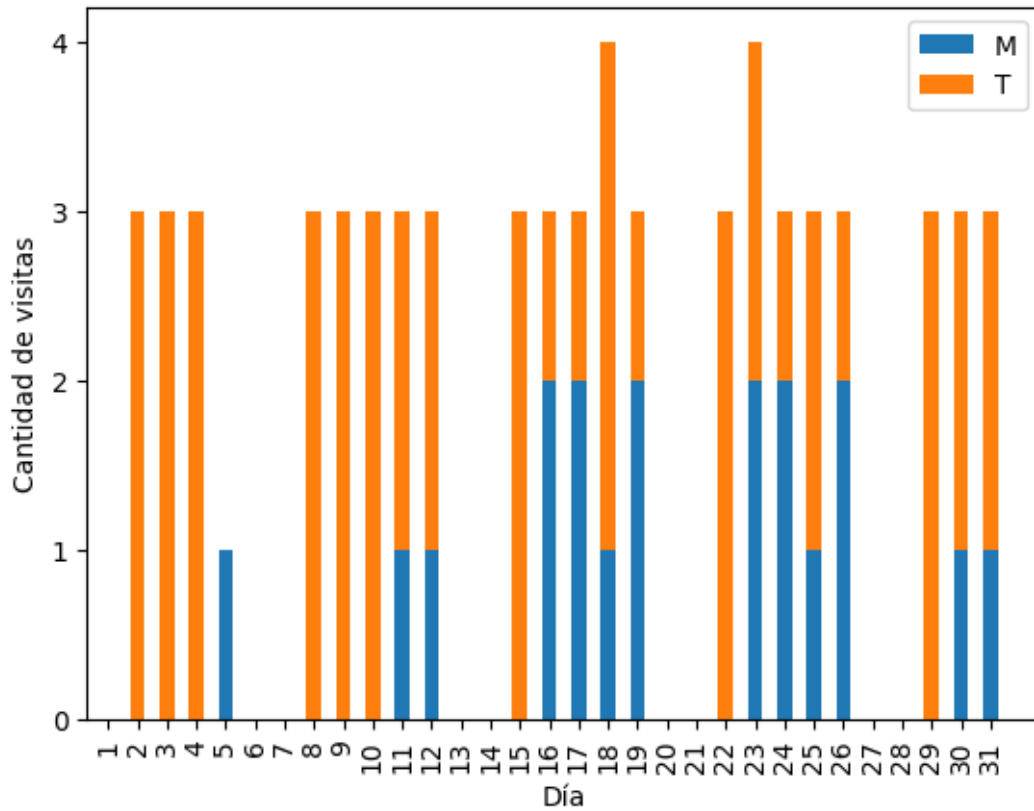


Figura 5.1: Cantidad de visitas por día y turno en el histórico (“M” - mañana y “T” - tarde).

## 5.2. Experimentos

Se realizan un total de cuatro experimentos para comprobar que el modelo cumple los requerimientos. En los tres primeros se trabaja con el modelo inicial y en el

último experimento se trabaja con el modelo con *stock*. Lo que se busca es mostrar cómo el usuario puede incluir sus *inputs*, cómo puede hacer iteraciones sucesivas con el optimizador y qué mensajes recibe cuando un problema no es factible.

### 5.2.1. Experimento con visitas libres

En el primer experimento, se presentan al optimizador los datos de entrada con exactamente los mismos puntos periódicos del histórico y con todos los puntos no-periódicos que el algoritmo de preprocesado sugiere para este mes. Se define una demanda de 200 bolsas al día, valor 21 % superior al promedio realizado en este mes. Los objetivos de este experimento son: comparar los resultados del optimizador con el histórico y mostrar la funcionalidad de diagnóstico de no factibilidad.

En la figura 5.2 se presenta un calendario con las fechas de visita posibles para cada punto en verde y la fecha elegida por el optimizador en rojo. Es fácil notar que las visitas ocurren solamente en las fechas disponibles. En el eje vertical se nota que hay un mínimo de dos y un máximo de cuatro visitas al día, respetando las restricciones del problema. En el eje horizontal la cantidad de visitas realizadas en cada punto de extracción, que en el caso de los puntos periódicos es exactamente igual a la cantidad solicitada.

Tras un análisis del patrón de visitas, es posible notar una posible oportunidad de mejora del modelo. Tómese como ejemplo el punto “12700”, que recibe cuatro visitas en el mes. Las tres primeras son en días consecutivos, pero puede que sea importante en determinadas situaciones controlar el espaciado entre visitas a un mismo punto de extracción. Esta posibilidad queda como sugerencia para desarrollo en un trabajo futuro.

Sin embargo, cuando se analiza la cantidad de visitas realizadas cada día en el turno de la mañana, se nota que la solución no cumple con la restricción blanda correspondiente. Esto es debidamente señalado al usuario en la salida del *script* de pos-procesado que se presenta en la figura 5.3.

Comparando los resultados en la tabla 5.4 con el histórico en la tabla 5.1 anterior, se nota que se presentaron los mismos 49 puntos periódicos al optimizador y se realizaron las mismas 62 visitas a ellos. Sin embargo, se recolectaron más bolsas y se visitaron más puntos no-periódicos en el mes.

Periódico	# Puntos	# Visitas	# Bolsas
Sí	49	62	3.926
No	14	15	510,1
<b>Total</b>	<b>63</b>	<b>77</b>	<b>4.436,1</b>

Tabla 5.3: Resumen del plan optimizado del mes de enero de 2018.

Como el modelo es determinístico y no estocástico, lo que permite que la cantidad de bolsas sea mayor es principalmente la posibilidad de recolectar más bolsas en una visita, siempre dentro del rango permitido del parámetro de desviación estándar. Otro hecho que permite incrementar el total recolectado son las visitas a puntos no-periódicos, pues el optimizador es capaz de evaluar combinaciones adicionales de visitas que respetan las restricciones del modelo, como es el caso de las segundas visitas a puntos del módulo “A” en el turno de la tarde.

Por último, algunos de los puntos no-periódicos adicionales presentados son centros educativos y el optimizador los elige para visitar. Esto resulta en un porcentaje

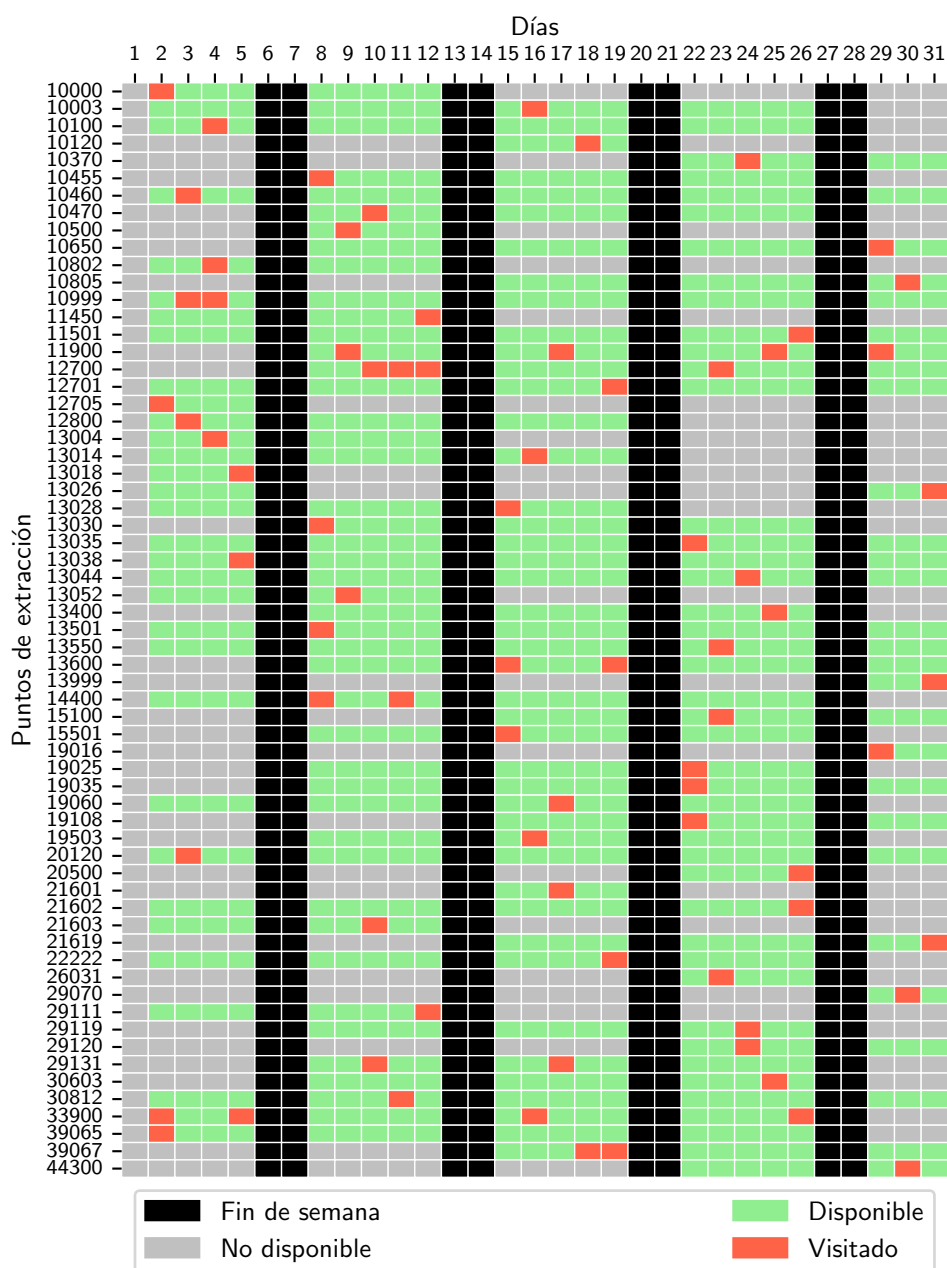


Figura 5.2: Resultado del experimento con visitas libres para el mes de enero de 2018.

```

-----
DIAGNÓSTICO DE NO FACTIBILIDAD:

Cantidad de días sin el mínimo de visitas por la mañana: 6
La solución obtenida viola 01 restriccion(es) blanda(s) del modelo.
-----

```

Figura 5.3: Mensaje de diagnóstico de no factibilidad.

de visitas a escuelas de 15,58%, valor este superior al mínimo deseado, como se puede observar en la tabla 5.4.

Se puede también analizar la solución en detalle importando los ficheros de salida

Tipo	# Puntos	# Visitas	# Bolsas
Centros Educativos	11	12	512,8
Otros	52	65	3923,3
<b>Total</b>	<b>63</b>	<b>77</b>	<b>4.436,1</b>

Tabla 5.4: Desglose del plan optimizado por tipología de puntos visitados.

en el informe desarrollado en *Power BI*. La figura 5.4 muestra la cantidad de enfermeros, administrativos, médicos y conductores necesarios por día, juntamente con el coste total de personal y el coste medio por bolsa recolectada. El usuario puede filtrar un punto de extracción específico, turno o unidad móvil para actualizar la gráfica y los totales.

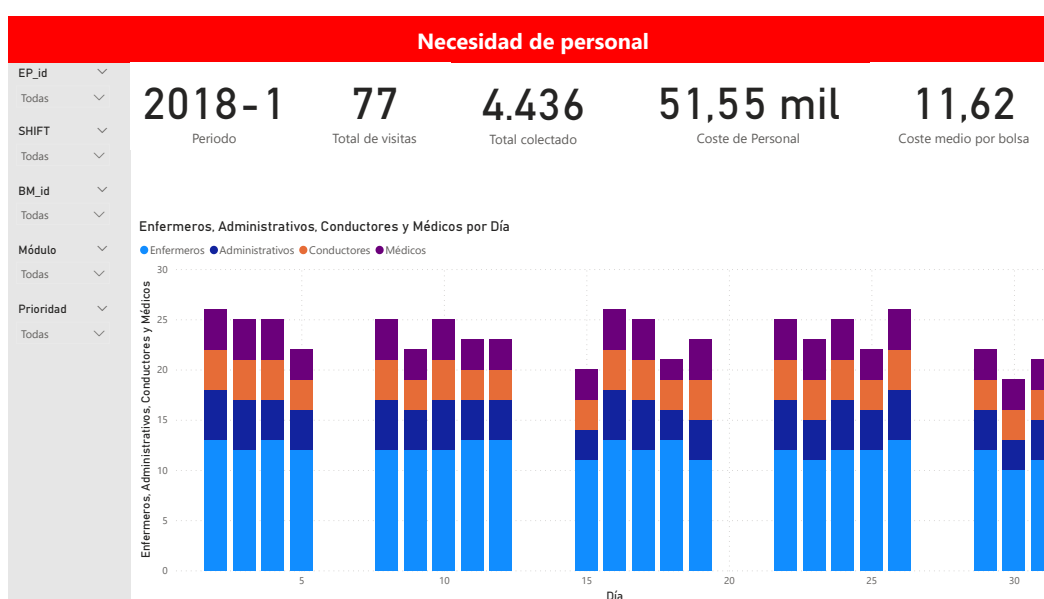


Figura 5.4: Vista de una página del informe con el detalle de la necesidad diaria de personal.

Otro análisis del informe es el de las visitas, donde se puede ver los detalles de cantidad diaria de visitas por turno, juntamente con el indicador de visitas a centros educativos, como se presenta en la figura 5.5.

Por una cuestión de brevedad, no se presentan en este trabajo las demás páginas del informe, pero vale recalcar que incluyen toda la información presentada anteriormente, tales como las visitas por turno, por tipo de punto de extracción o el indicador de visitas a centros educativos, entre otros.

### 5.2.2. Experimento con bloqueo de visitas

En el segundo experimento, se simula un caso de uso en que el usuario fija las visitas ya acordadas a puntos de extracción en días y turnos específicos, eligiendo también la unidad móvil de su preferencia para cada visita, y pide entonces al optimizador que proponga un plan óptimo para las demás visitas del mes que todavía están en proceso de confirmación. El objetivo es mostrar que el modelo está integrado

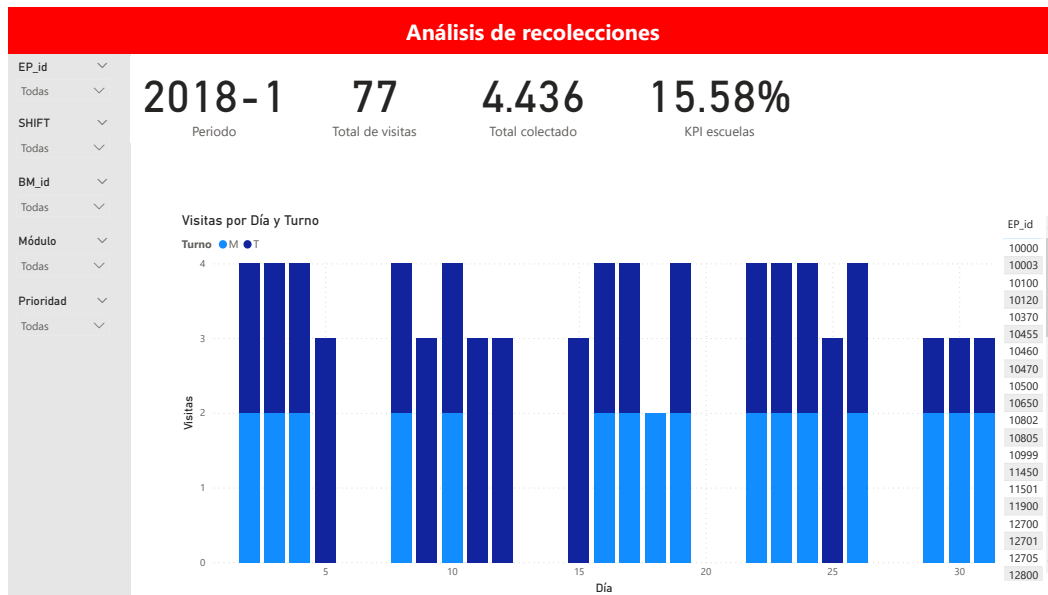


Figura 5.5: Vista de la página de análisis de visitas por turno, con múltiples posibilidades de filtros.

en un lazo de control cerrado, permitiendo al responsable de la planificación repetir optimizaciones hasta encontrar la combinación de visitas adecuada.

La figura 5.6 muestra el resultado de optimizaciones sucesivas en las que cada escenario consiste en bloquear las visitas de la semana anterior. En este caso de uso, el responsable de la planificación ya conoce cuales fueron los puntos de extracción visitados cada día y turno en la semana anterior, qué cantidades fueron colectadas en cada uno y utilizando qué vehículo. El plan es revisado entonces para optimizar solamente las visitas en las semanas que están libres.

Como se puede observar, se parte de un escenario en que todas las visitas son libres, con 4.436,1 bolsas recogidas, y se llega a un escenario con las mismas visitas y cantidades de bolsas del histórico, con 3.637 bolsas recogidas. Existe un comportamiento asintótico que se explica por el hecho de que cuantas más visitas se bloqueen con cantidades de bolsas iguales al valor histórico del mes, menor es el grado de libertad del optimizador para obtener mejores resultados.

### 5.2.3. Experimento con optimización de 6 meses

En el tercer experimento, se aplica el modelo con *stock* a los datos de entrada generados para los seis primeros meses del año de 2018. El objetivo es mostrar que el algoritmo de preprocesado lleva en consideración las optimizaciones de meses anteriores para sugerir al usuario los puntos que deben ser visitados en un determinado mes.

Primeramente, cabe recalcar que los datos del mes de enero son distintos de los experimentos anteriores, pues se deja que el algoritmo genere los datos libremente. Los demás meses siempre llevan en consideración el resultado de las optimizaciones de hasta 12 meses anteriores.

La tabla 5.5 muestra los resultados obtenidos. Lo primero que se nota es que el algoritmo incluye más visitas a puntos periódicos en enero que lo que las restricciones del problema permiten. Sin embargo, el optimizador es capaz de calcular el plan

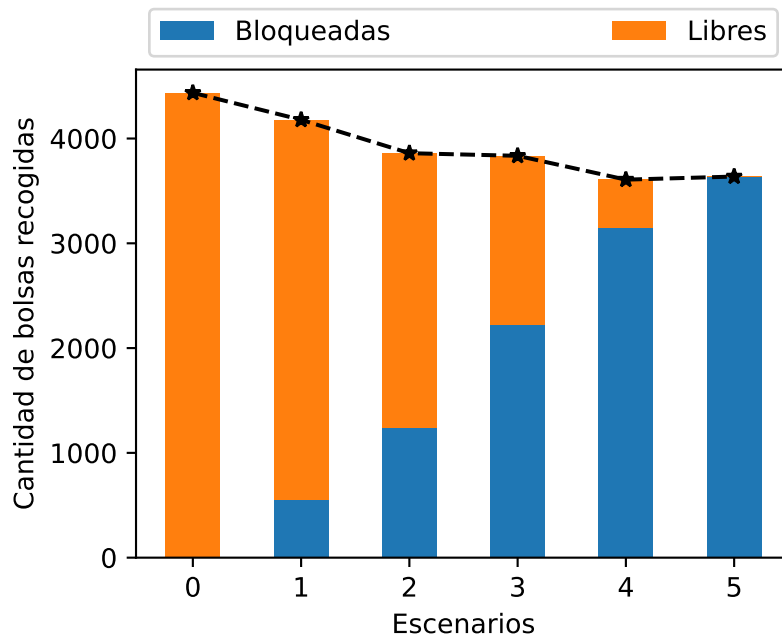


Figura 5.6: Resultados del experimento con visitas bloqueadas para el mes de enero de 2018

óptimo y cumplir con la demanda y con el indicador de visitas a centros educativos. Lo mismo ocurre en el mes de febrero, pero se nota que la cantidad de puntos de extracción sugeridos es menor que en el mes anterior. Cuando se generan los datos de marzo se nota un problema, pues el algoritmo no es capaz de sugerir más puntos de manera que mantenga la cadencia. Este problema mejora poco a poco en los meses siguientes, una vez que más puntos vuelven a estar disponibles para visitación, pero el impacto es que el optimizador no es capaz de calcular un plan que cumpla con la demanda con tan pocos puntos de extracción.

Indicadores	ene	feb	mar	abr	may	jun
Días laborables	23	20	22	21	23	21
Puntos presentados	103	91	21	33	49	50
Visitas a puntos periódicos	95	90	13	39	46	55
Visitas periódicas no atendidas	6	12	0	4	0	3
Demanda histórica	3.637	3.090	3.478	3.472	3.561	3.579
Demanda total	4.600	4.000	4.400	4.200	4.600	4.200
Total de bolsas recogidas	4.625	4.040	1.226	2.933	3.303	2.987
Visitas a centros educativos	11	8	0	1	12	5
Porcentaje de visitas a centros educativos	11,96	10,00	0,00	2,63	22,22	8,93

Tabla 5.5: Resultado del experimento con optimizaciones consecutivas considerando histórico.

Después de un análisis detallado, se puede decir con seguridad que es necesario mejorar el algoritmo de preprocesado, pues se detectan algunas inconsistencias. Un problema es que los puntos que aparecen en marzo en el histórico son distintos de los

que aparecen en enero y febrero, entonces la razón que menos bolsas son recogidas se debe a que menos puntos son presentados al optimizador y no al hecho de que se presenta una mayor demanda en los meses anteriores. Por poner otro ejemplo práctico, de los cinco puntos listados en la tabla 5.6, solamente dos cumplen con la frecuencia deseada en el periodo estudiado.

Sin embargo, el experimento es importante para mostrar que el optimizador es capaz de producir resultados incluso cuando los datos de entrada son inconsistentes. Se detectan las oportunidades de mejora debido al modelado con restricciones blandas y también al *script* de posprocesado que presenta de manera clara al usuario un resumen tanto de los datos de entrada cuanto de los resultados obtenidos.

Puntos de extracción	ene	feb	mar	abr	may	jun
10000	1	0	0	0	1	0
10100	1	0	0	0	1	0
10122	1	0	0	0	0	0
10140	1	0	0	0	0	0
10332	1	0	0	0	0	1

Tabla 5.6: Análisis de cadencia para puntos de extracción con frecuencia “Periódica 1/4” (1 - Visita realizada, 0 - Punto no visitado).

#### 5.2.4. Experimento con modelo con *stock*

En el cuarto y último experimento, se aplica el modelo con *stock* a los datos de entrada generados para el mes de enero. El objetivo es mostrar que este modelo extendido también funciona y puede ser utilizado de manera análoga al modelo inicial para controlar las existencias en el CRHM.

Los datos de entrada fueron cambiados arbitrariamente para simular un escenario en que la demanda de bolsas aumentase cada semana y también la cantidad de bolsas almacenadas en el centro variase a lo largo del mes para verificar como eso impacta las visitas y extracciones. Se fija el *stock* inicial en 450 bolsas, el mínimo en 400 y el máximo en 700 unidades. La figura 5.7 muestra el resultado del experimento y se puede observar que las existencias en el centro siguen la curva del *stock* objetivo. Haciendo una analogía a un sistema de control, la señal de salida (*stock*) sigue la señal de consigna (*stock* objetivo) en este experimento.

Otro punto importante que puede ser observado en la figura 5.7 es que el *stock* no supera el límite máximo permitido, aunque la consigna esté intencionalmente por encima de este valor en el día 19. Eso prueba que se cumple la restricción (4.16).

Se presenta en la figura 5.8 el fichero con los mensajes generados en la etapa de posprocesado para esta solución. El coste total de personal es calculado con las funciones lineales por partes incluidas en el modelo para cálculo de estadísticas.



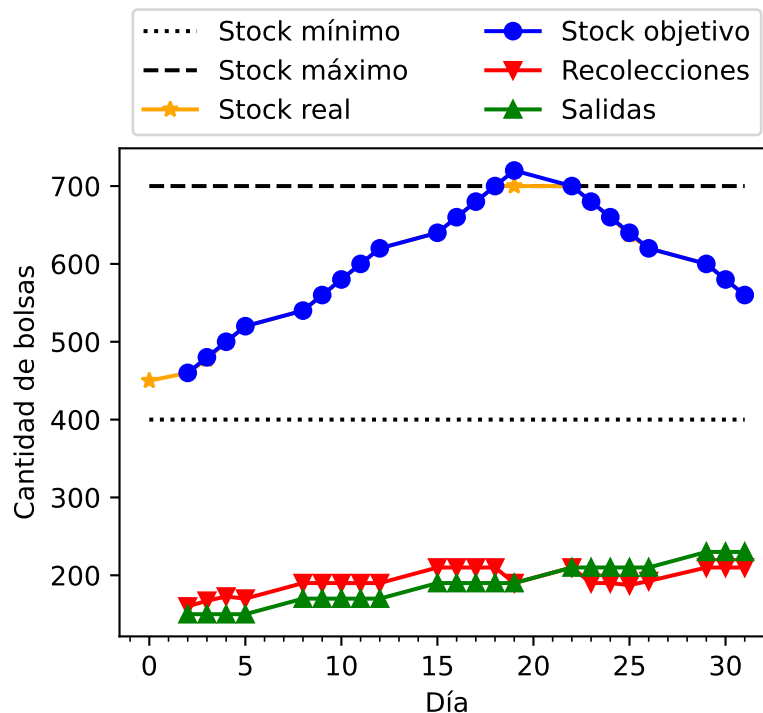


Figura 5.7: Resultados del experimento con *stock*

-----
INFORMACIONES GENERALES:
Período: 2018-1
Lectivo: 1
Días: 22
Puntos de extracción: 64
Unidades móviles: 4
Turnos: 2
Visitas a puntos periódicos: 62
-----
DIAGNÓSTICO DE NO FACTIBILIDAD:
Cantidad de días sin el mínimo de visitas por la mañana: 6
La solución obtenida viola 01 restriccion(es) blanda(s) del modelo.
-----
ESTADÍSTICAS:
Demanda total: 4140
Total de bolsas recogidas: 4250
Diferencia: 2.66%
Total de visitas: 77
Total de visitas a centros educativos: 12
Porcentaje de visitas a escuelas: 15.58%
Coste total de personal: 50265.42

Figura 5.8: Fichero con mensajes del posprocesado.

# Capítulo 6

## Conclusiones y Trabajo Futuro

Los resultados de los experimentos comprueban que la herramienta cumple todos los requisitos. Además de ser capaz de producir planes de recolección óptimos, es también capaz de detectar cuándo se viola alguna de las restricciones blandas y lo informa al usuario de manera amigable. El responsable de la planificación puede cambiar los datos de entrada y hacer sucesivas iteraciones hasta obtener las combinaciones que juzgue aceptables en un lazo de control cerrado.

Además del modelo en OPL, también se presenta una versión preliminar de un informe desarrollado en Power BI para que el usuario pueda analizar detalles de la solución y visualizar métricas tales como la necesidad de personal y costes totales calculados.

Por último, se presenta una extensión del modelo inicial incorporando el *stock* en el CRHM. Se trata de un modelo integrado que puede ayudar al responsable de la planificación controlar las existencias en el centro siempre que una buena previsión de la demanda esté disponible y los parámetros objetivos de la política de *stock* estén debidamente definidos.

Es necesario continuar el desarrollo del algoritmo de preprocesado para que pueda considerar las optimizaciones pasadas y sugerir las visitas posibles de una manera más fiable. Sin embargo, los experimentos realizados muestran que el optimizador es capaz de producir buenos resultados incluso cuando los datos de entrada presentan inconsistencias.

Para trabajos futuros, se proponen algunos puntos de mejora, que se pueden clasificar en cuatro tipos: automatizar la ejecución del optimizador; modificar los datos de entrada y salida; agregar restricciones adicionales al modelo o eliminar restricciones del modelo; y, por último, cambiar la función objetivo.

La primera oportunidad de mejora que se sugiere es la de implementar el modelo directamente Python con la API de CPLEX, sin necesidad de usar el IDE. Esto puede automatizar la ejecución del optimizador y hacer que sea más fácil para el usuario ejecutar optimizaciones sucesivas.

Con relación a los datos de entrada y salida, la primera modificación propuesta se refiere a la información del día. Actualmente se trabaja con un número entero, pero se sugiere que se codifiquen en todas las tablas las fechas completas en el formato reconocido por OPL (“YYYY/DD/MM”) en lugar de solamente el día. Esta modificación es sencilla y elimina la necesidad de almacenar el mes y año en la tabla auxiliar *Parameters*.

Todavía con relación a los datos de entrada y de salida, otra mejora posible es cambiar la estructura de las tablas. El código OPL puede ser simplificado al cambiar dónde se almacena o cómo se presenta la información al optimizador. Por poner un ejemplo, se utiliza la tabla de entrada *Visits* para informar la cantidad de

visitas que se ha de realizar en un mes, pero también se informa en esta tabla si un punto es periódico, lo que es redundante, ya que esta información también está en la tabla *Structural*. Es posible combinar estas dos tablas en una única tabla de entrada cambiando el código del preprocesado, incluso si la información en el sistema se almacena en tablas distintas. También es posible combinar las tablas de salida para las variables *alpha* y *gamma* en una sola, pues ambas presentan información complementaria. Basta con añadir una columna adicional para la cantidad de bolsas recolectadas a la tabla de salida correspondiente a la variable *alpha* para que toda la información de las visitas esté en un único fichero.

Las mejoras anteriores simplifican el código, pero no cambian el modelo. Los cambios que se sugiere que se incorporen al modelo son los siguientes:

- Añadir la restricción para controlar el espaciado entre visitas sucesivas a un mismo punto de extracción;
- Adaptar el modelo para permitir la retirada de bolsas del centro durante los fines de semana;
- Detallar la cantidad de bolsas por tipo sanguíneo;
- Considerar pérdidas por caducidad en el *stock*;

Controlar el espaciado entre visitas sucesivas a un punto añade una restricción adicional al modelo, pero parece ser algo lógico y deseable, ya que cuando se quiere, por ejemplo, cuatro visitas a un punto de extracción en un mes, se suele hacer una visita cada semana. La adaptación para permitir retirada de bolsas del centro durante los fines de semana deja el modelo más general. Detallar la cantidad de bolsas por tipo sanguíneo añade una dimensión extra al modelo, pero se cree que es un dato importante para el CRHM una vez que la demanda es distinta para cada tipo sanguíneo. Por último, por tratarse de productos perecederos, considerar pérdidas por caducidad en el *stock* resulta en un modelo más realista.

El último cambio propuesto en este trabajo requiere más esfuerzo de desarrollo, pero se cree que es un buen tema para futuras investigaciones. Se trata de transformar el modelo en multiobjetivo añadiendo una función objetivo adicional para el coste total de las visitas. Como ya se calculan los costes de personal de cada visita en el modelo con una función lineal por partes, se puede añadir al modelo una segunda función objetivo utilizando la función “staticLex” de OPL. De esta manera, el optimizador buscaría minimizar primero el objetivo actual, considerado el más importante, y en segundo lugar buscaría minimizar el coste total de la solución, o sea, en orden lexicográfico. En caso de que este enfoque no sea efectivo, se podría estudiar el uso de heurísticas en el modelo multiobjetivo.

# Chapter 7

## Introduction

### Logistics management in blood donation

Blood and its components — such as red blood cells, leukocytes, platelets and plasma —, are used in a wide range of medical treatments. A transfusion may be needed, for example, when a patient bleeds too much after an injury or during certain surgeries, or when a person has an inherited blood disorder such as thalassemia or a bleeding disorder caused by lack of blood clotting factors.

The first thing that comes to mind when one thinks about blood is usually whole blood, but blood products are also very important to the pharmaceutical industry. Plasma-derived therapies are potentially life-saving treatments for patients with rare and complex diseases. There are many medicines whose active principle comes from the plasma of healthy human donors through a proper fractionation and purification process, such as the following [1]:

- Human albumin;
- Human immune globulin;
- Coagulation factor VIII and human von Willebrand factor;
- Coagulation factor IX;
- Alpha-1-Antitrypsin (AAT) concentrate;
- High-purity human antithrombin.

It is of vital importance that these medical products are promptly available to guarantee healthcare activity. Donations are currently the only way to obtain human blood, so a proper blood supply is crucial as frequent donations are required and blood components have a limited shelf life.

In Spain, Transfusion Centers (TC) are responsible for guaranteeing sufficiency in blood components. These centers must always monitor the existing stocks in the blood bank and act to avoid possible shortages. To achieve this goal, they maintain regular communication with other centers, with hospital transfusion services and also regularly promote campaigns to encourage donation.

Each of the autonomous communities has one or more of these centers, forming the network of the Transfusion System in Spain presented in the figure 7.1 [2].



Figure 7.1: Transfusion center network in Spain.  
(Reproduced from: Spanish Ministry of Health [2])

The work of the TC is important within the transfusion chain, or more broadly the blood supply chain, with activities ranging from the selection of the donor to the extraction, processing, storage, preparation and distribution of components, as shown in figure 7.2[3].

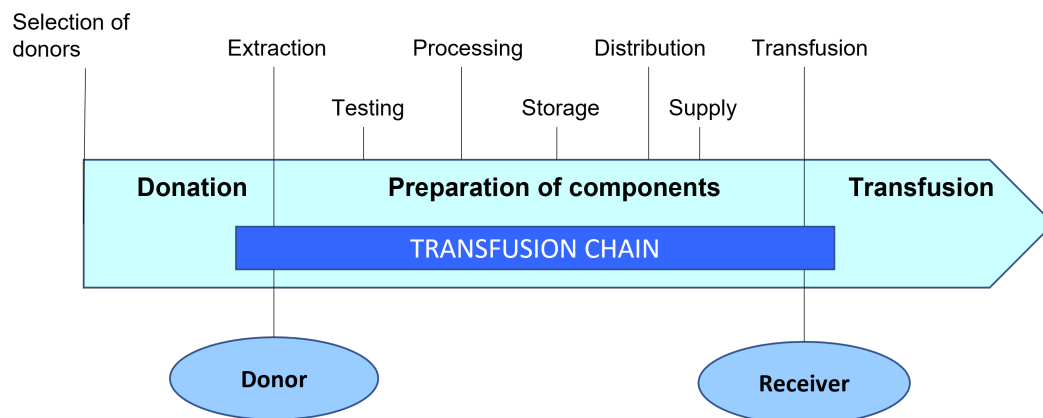


Figure 7.2: Processes in the transfusion chain.  
(Adapted from: Spanish Ministry of Health [3])

The center of interest in this work is the Murcia Regional Blood Donation Center (in Spanish, CRHM), responsible for covering the donation needs of this entire autonomous community. In 2020, the CRHM managed a total of 46,978 whole blood donations [4], which is a less than the amount of previous years [5, 6], as shown in figure 7.3.

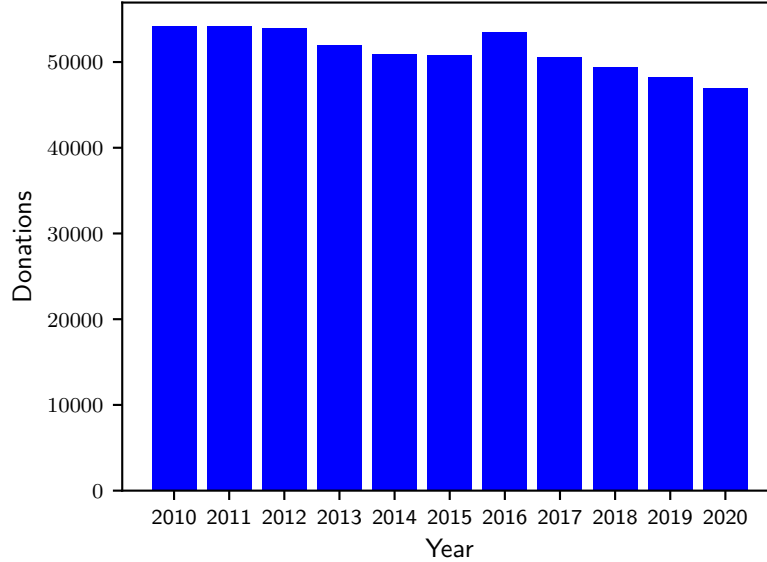


Figure 7.3: Blood donations per year in Murcia.  
Adapted from: Murcia Regional Blood Donation Center [7].

This reduction is due to the impact of the COVID-19 pandemic throughout the country. Fortunately, there were no major consequences, since fortunately the centers have been able to respond to hospital demands, and because there was a reduction in demand in this period [7]. What we want to emphasize with this data is that the amount of donations is variable and depends on multiple social factors that influence the behavior of donors.

**Logistic management of blood collection** is challenging and the planning task is not trivial. Variability and randomness are inherent to this problem and there are many restrictions that must be taken into consideration to build a feasible plan, such as the frequency of visits to each extraction point, the operating hours of each point, the capacity of the vehicles, and other operational restrictions that are deemed necessary.

Based on the aforementioned reasons, we believe that the best way to prepare the collection plan for the center is by using a computer solution. Such solution would reduce planning time and increase the quality of the plan, thus improving the use of available resources. The objective of this work is to develop a tool to support the task of defining an optimal plan for the collection of blood bags at extraction points in order to maintain the blood stock of the CRHM at a desired level. In this paper, a mixed integer linear programming model is presented to meet this objective.

## Motivation

The planning activity is currently carried out in the CRHM based on the experience of the responsible professional, but with no support from any specific computer software. For this reason, it was reported that the planner may need up to a few weeks to develop the blood collection plan. Our work is part of a larger project developed together with the CRHM to develop an online system that allows optimizing the processes of the blood supply chain, ensuring self-sufficiency and reducing potential occurrences of shortages.

The project covers multiple activities for the development of the proposed system, which are shown in figure 7.4, such as the database definition, the development of a module to preprocess the data, the development of the optimizer, in addition to the treatment of the results and implementation of the end user graphic interface. The contribution of this work is mainly focused on the development of the mathematical model for the optimizer, which is the core of the system.

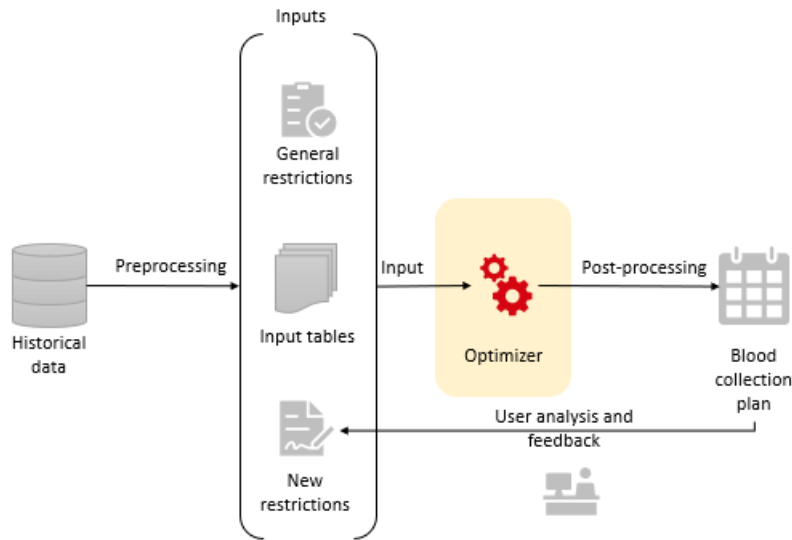


Figure 7.4: Schematic overview of the planning system.

## Objectives

The collaboration with the CRHM began some time ago and since then the research group has been able to gather the necessary data and map the required constraints for the model. This collaboration has already generated the publication of an academic paper in previous years. This work is a continuation of [8], defining a more generic and abstract mathematical model and considering improved demand forecasts. In particular, the following specific objectives are defined for this final master's degree project:

- Develop an improved abstract mathematical model for the monthly planning of visits to extraction points.
- Take into account in the model that the user can include new restrictions and iteratively optimize the solution.

- Carry out experiments with the available data that demonstrate the correct functioning of the tool.
- Make a comparison between the actual planning data and the results obtained with the optimizer.
- Develop a debugging mathematical model to help the end user diagnose problems in the data that may make the problem infeasible.

## Document structure

The remainder of this document is organized as follows:

- A review of the relevant literature is presented in chapter 2.
- Chapter 2, a brief summary of the input data is presented together with the description of the tools used.
- Chapter 3 is dedicated to define the problem and formulate the different mathematical models considered in this study.
- The results obtained with the developed models are shown in chapter 5, making a comparison with the planning carried out by the CRHM between different use cases.
- Lastly, conclusions and suggestions for future work are presented in chapter 6.

## Project code

The code developed in this work together with an extract of the input data and the output of the experiments performed are accessible on demand in the following link:

[https://github.com/greenlsi/hemoglobulab\\_opt](https://github.com/greenlsi/hemoglobulab_opt)



# Chapter 8

## Conclusions and Future Work

The results of the experiments prove that the optimizer fulfils all the requirements. In addition to being able to produce optimal collection plans, it is also capable of detecting when any of the soft constraints are violated and informs the planner in a user-friendly way. The person in charge of the planning can change the input data and make successive iterations in a closed control loop until obtaining the combinations considered acceptable.

In addition to the OPL model, a preliminary version of a dashboard developed in Power BI is also presented so that the user can analyze details of the solution and view metrics such as the headcount to perform the collections and total calculated personnel costs.

Finally, an extension of the initial model is presented incorporating the stock at the CRHM. This is an integrated model that can help the planner control stocks at the site when a good demand forecast is available and the objective parameters of the inventory policy are properly defined.

It is necessary to continue the development of the preprocessing algorithm so that it can consider past optimizations and suggest possible visits in a more reliable way. However, the experiments carried out show that the optimizer can produce good results even when the input data presents inconsistencies.

For future work, some improvement points are proposed, which can be classified into four types: automate the execution of the optimizer; modify the input and output data; add additional constraints to the model or remove constraints from the model; and finally, change the objective function.

The first suggested improvement opportunity is to implement the model directly using the Python API of CPLEX, without using the IDE. This can automate the execution of the optimizer and make it easier for the user to run successive optimizations.

Regarding the input and output data, the first proposed modification refers to the information of the day. Currently the day is an integer number in the input tables, but it can be encoded as a full date in the format recognized by OPL (“YYYY/DD/MM”) instead of just the day. This modification is simple and eliminates the need to store the month and year in the *Parameters* auxiliary table.

Still in regard to the input and output data, another possible improvement is to change the structure of the tables. The OPL code can be simplified by changing where the information is stored or how it is presented to the optimizer. For example, the input table *Visits* is used to report the number of visits to be made in a month, but it is also reported in this table if a point is periodic, which is redundant, since that this information is also in the *Structural* table. It is possible to combine these two tables into a single input table by changing the preprocessing code, even if

the information in the system is stored in different tables. It is also possible to combine the output tables for the variables *alpha* and *gamma* into one, since both present complementary information. It is enough to add an additional column for the number of bags collected to the output table corresponding to the variable *alpha* so that all the information on the visits is in a single file.

The above enhancements simplify the code, but do not change the model. The following points could be incorporated into the model for improvement:

- Add the restriction to control the spacing between successive visits to the same extraction point;
- Adapt the model to allow outbound flows from the center during weekends;
- Detail the number of bags by blood type;
- Consider losses due to expiration in the stock;

Controlling the spacing between successive visits to a point adds an additional restriction to the model, but it seems to be something logical and desirable, since when there are four visits to an extraction point in a month for example, it is usually accomplished with one visit every week. The adaptation to allow the removal of bags from the center during weekends results in a more general model. Detailing the number of bags by blood type adds an extra dimension to the model, but it is an important information for the CRHM since the demand is different for each blood type. Lastly, since these are perishable products, considering expiration losses in the stock results in a more realistic model.

The last proposed change requires more development effort but is an exciting topic for further research. The modification is about transforming the model into a multi-objective one by adding an additional objective function for the total cost of visits. Since personnel costs for each visit are already calculated in the model with a piecewise linear function, a second objective function can be added to the model using the OPL function “staticLex”. In this way, the optimizer would first seek to minimize the current objective, considered the most important, and secondly it would seek to minimize the total cost of the solution, that is, in lexicographical order. In case this approach is not effective, the use of heuristics to the multiobjective model could be studied.

# Bibliografía

- [1] Banc de Sang i Textits, “Derivados del plasma.” [https://www.bancsang.net/professionals/productes-serveis/components-sanguinis/es\\_derivats-plasma/](https://www.bancsang.net/professionals/productes-serveis/components-sanguinis/es_derivats-plasma/), Accedido el 05/03/2023.
- [2] Ministerio de Sanidad, “Estructura de la red transfusional en España.” <https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/redTransfuEspana/RedTransEspana.htm>, Accedido el 24/07/2022.
- [3] Ministerio de Sanidad, “Informe hemovigilancia,” 2020. <https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/hemovigilancia/docs/Informe2020.pdf>, Accedido el 24/07/2022.
- [4] Ministerio de Sanidad, “Centros de transfusión sanguínea de la región de murcia.” <https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/ComunidadesAutonomas/murcia.htm>, Accedido el 24/07/2022.
- [5] Centro de hemodonación de Murcia, “Estadísticas.” <https://www.murciasalud.es/crh.php?iddoc=6929&idsec=202>, Accedido el 24/07/2022.
- [6] Ministerio de Sanidad, “Hemovigilancia año 2019.” <https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/hemovigilancia/docs/Informe2019.pdf>, Accedido el 14/05/2023.
- [7] Ministerio de Sanidad, “Informe actividad de centros y servicios de transfusión,” 2020. [https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/indicadores/docs/Informe\\_Actividad2020.pdf](https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/indicadores/docs/Informe_Actividad2020.pdf), Accedido el 24/07/2022.
- [8] E. Fernández García, “Diseño e implementación de una herramienta de modelado y optimización para ayudar a la planificación de colectas de un centro de hemodonación,” Master’s thesis, Universidad Politécnica de Madrid, 2019.
- [9] E. Alfonso, V. Augusto, and X. Xie, “Mathematical programming models for annual and weekly bloodmobile collection planning,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 96–105, 2014.
- [10] A. F. Osorio, S. C. Brailsford, H. K. Smith, S. P. Forero-Matiz, and B. A. Camacho-Rodríguez, “Simulation-optimization model for production planning in the blood supply chain,” *Health Care Management Science*, pp. 548–564, 12 2017.

- [11] A. F. Osorio, S. C. Brailsford, and H. K. Smith, “A structured review of quantitative models in the blood supply chain: A taxonomic framework for decision-making,” *International Journal of Production Research*, vol. 53, pp. 7191–7212, 12 2015.
- [12] J. Beliën and H. Forcé, “Supply chain management of blood products: A literature review,” *European Journal of Operational Research*, vol. 217, pp. 1–16, 02 2012.
- [13] S. Baş Güre, G. Carello, E. Lanzarone, and S. Yalçındağ, “Unaddressed problems and research perspectives in scheduling blood collection from donors,” *Production Planning & Control*, vol. 29, no. 1, pp. 84–90, 2018.
- [14] E. Alfonso, X. Xie, and V. Augusto, “A simulation-optimization approach for capacity planning and appointment scheduling of blood donors based on mathematical programming representation of event dynamics,” 2015.
- [15] J. D. Michaels, J. Brennan, B. L. Golden, M. Fu, and A. R. Cross, “A simulation study of donor scheduling systems for the american red cross,” 1993.
- [16] A. M. Baart, S. Fontana, A. Tschaggelar, M. W. Heymans, and W. L. D. Kort, “Generalizability of dutch prediction models for low hemoglobin deferral: A study on external validation and updating in swiss whole blood donors,” *Transfusion Medicine and Hemotherapy*, vol. 43, pp. 407–414, 11 2016.
- [17] “Traceability project of a blood supply chain,” *Supply Chain Forum*, vol. 17, pp. 15–25, 2016.
- [18] A. F. Osorio, S. C. Brailsford, and H. K. Smith, “Whole blood or apheresis donations? a multi-objective stochastic optimization approach,” *European Journal of Operational Research*, vol. 266, pp. 193–204, 4 2018.
- [19] Servicio Madrileño de Salud, “Donación de sangre.” <https://www.comunidad.madrid/servicios/salud/donacion-sangre>, Accedido el 04/06/2023.
- [20] Ministerio de Sanidad, “Manual de gestión de donantes,” 2011. [https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/publicaciones/docs/Manual\\_Gestion\\_Donantes.pdf](https://www.sanidad.gob.es/profesionales/saludPublica/medicinaTransfusional/publicaciones/docs/Manual_Gestion_Donantes.pdf), Accedido el 04/06/2023.
- [21] Centro de Transfusión, Tejidos y Células de Granada y Almería, “Productos sanguíneos.” <http://transfusion.granada-almeria.org/actividad/productos-sanguineos>, Accedido el 21/05/2023.
- [22] A. F. Osorio, S. C. Brailsford, H. K. Smith, and J. Blake, “Designing the blood supply chain: how much, how and where?,” *Vox Sanguinis*, vol. 113, pp. 760–769, 11 2018.

# Apéndice A

## Modelo OPL

### Lectura de datos

Como detallado en el Capítulo 3, los datos de entrada del modelo están en archivos con el formato CSV. La lectura de estos archivos también es parte del código OPL.

### Declaración de tuplas y variables auxiliares

```
1 // DEMAND: (en[D])
2 tuple CalendarTuple {
3     int week;
4     key int day;
5     float corr;
6     float demand;
7     int minVisitsPerDay;
8     int maxVisitsPerDay;
9     int maxVisitsPerShift;
10    int maxBM;
11 };
12
13 // BLOOD MOBILES: bm \in BM
14 tuple BloodmobilesTuple {
15     key string bm_id;
16     int cb_min;
17     int cb_max;
18 }
19
20 // STATISTICS: (DELTA)
21 tuple StatisticsTuple {
22     string ep;
23     int week;
24     int day;
25     int school;
26     float corr;
27     float _min;
28     float avg;
29     float std;
30     float _max;
31     int num;
32     int period_len;
33 }
34
35 // STRUCTURAL: ep \in EP:
```

```

36 tuple StructuralTuple {
37     key string ep_id;
38     string collection;
39     string module;
40     string type;
41     float priority;
42     string frequency;
43     string specific_month;
44     string specific_day;
45 };
46
47 // NUM_VISITS:
48 tuple VisitsTuple {
49     key string ep;
50     int periodic;
51     int num_visits;
52 }
53
54 // SHIFTS:
55 tuple ShiftTuple {
56     string ep_id;
57     string shift;
58 };
59
60 // ALPHA:
61 tuple AlphaTuple {
62     int day;
63     string ep_id;
64     string bm_id;
65     string shift;
66     int lb;
67     int ub;
68     int value;
69 };
70
71 // AlphaIndex: auxiliar tuple
72 tuple AlphaIndexTuple {
73     int day;
74     string ep_id;
75     string bm_id;
76     string shift;
77 };

```

## Preprocesado

```

78
79 //Variables auxiliares
80 int YEAR;
81 int MONTH;
82 int SCHOOL;
83 float TARGET;
84 float STOCK_INI;
85 float STOCK_MIN;
86 float STOCK_MAX;
87
88 //Carga datos de la tabla PERIOD_INFO con el mes y el parametro que
    indica si es lectivo
89 //Si el mes es lectivo, minimo de 10% y 5% si no es lectivo
90 execute LOAD_PERIOD_INFO {
91     var f = new IloOplInputFile("../data/opl_data/input/
        period_info.csv");

```

```

92 f.readline(); // Saltamos los encabezados
93 var data = f.readline();
94 if(data.length>0) {
95     data = data.split(",");
96     YEAR = Op1.intValue(data[0]);
97     MONTH = Op1.intValue(data[1]);
98     SCHOOL = Op1.intValue(data[2]);
99     TARGET = Op1.floatValue(data[3]);
100     if(data[4]!=null) STOCK_INI = Op1.floatValue(data[4]);
101     if(data[5]!=null) STOCK_MIN = Op1.floatValue(data[5]);
102     if(data[6]!=null) STOCK_MAX = Op1.floatValue(data[6]);
103 }
104 }
105
106 //Carga los datos de la tabla CALENDARIO (ya elimina festivos y
    finales de semana)
107 {CalendarTuple} CalendarTable = {};
108 execute LOAD_DEMAND {
109     var f = new IloOp1InputFile("../data/opl_data/input/demand.csv
        ");
110     f.readline(); // Saltamos los encabezados
111     while (!f.eof) {
112         var data = f.readline();
113         if(data.length>0) {
114             data = data.split(",");
115             //Carga la fila solamente si es un dia laborable y con
            demanda (corr y demanda > 0)
116             if(Op1.floatValue(data[2])>0 && Op1.floatValue(data[3])>0) {
117                 CalendarTable.add(Op1.intValue(data[0]) //week
                    ,Op1.intValue(data[1]) //day
                    ,Op1.floatValue(data[2]) //corr
                    ,Op1.floatValue(data[3]) //demand
                    ,Op1.intValue(data[4]) //min visits per day
                    ,Op1.intValue(data[5]) //max visits per day
                    ,Op1.intValue(data[6]) //max visits per shift
                    ,Op1.intValue(data[7])); //max bm per day
118             }
119         }
120     }
121 }
122 }
123 }
124
125 // DAYS SET d \in D: considera exactamente los días con demanda
    informada para el centro
126 {int} D = { row.day | row in CalendarTable};
127
128 // Lado derecho de las restricciones
129 int MIN_VISITS[D];
130 int MAX_VISITS[D];
131 int MAX_VISITS_SHIFT[D];
132 int MAX_BM[D];
133 execute READ_RHS {
134     for( var row in CalendarTable ) {
135         MIN_VISITS[row.day] = row.minVisitsPerDay;
136         MAX_VISITS[row.day] = row.maxVisitsPerDay;
137         MAX_VISITS_SHIFT[row.day] = row.maxVisitsPerShift;
138         MAX_BM[row.day] = row.maxBM;
139     }
140 }
141
142 //Calcula la demanda diaria de bolsas en el centro
143 float en[D];

```

```

149 execute CALC_ADJUSTED_DEMAND {
150     for(var row in CalendarTable) {
151         en[row.day] = row.demand * row.corr;
152     }
153 }
154
155 //Carga los datos de la tabla BLOODMOBILES (vehículos)
156 {BloodmobilesTuple} BloodmobilesTable = {};
157 execute LOAD_BLOODMOBILES{
158     var f = new IloOplInputFile("../data/opl_data/input/
159         bloodmobiles.csv");
159     f.readline(); // Saltamos los encabezados
160     while (!f.eof) {
161         var data = f.readline();
162         if(data.length>0) {
163             data = data.split(",");
164             BloodmobilesTable.add(data[0]
165                 ,Opl.intValue(data[1])
166                 ,Opl.intValue(data[2]));
167         }
168     }
169 }
170
171 //Conjunto de vehículos
172 {string} BM = { row.bm_id | row in BloodmobilesTable};
173
174 //Carga los datos de la tabla STATISTICS
175 {StatisticsTuple} StatisticsTable = {};
176 execute LOAD_STATISTICS {
177     var f = new IloOplInputFile("../data/opl_data/input/statistics
178         .csv");
178     f.readline(); // Saltamos los encabezados
179     while (!f.eof) {
180         var data = f.readline();
181         if(data.length>0) {
182             data = data.split(",");
183             //Carga la fila solamente si es un dia laborable y avg > 0
184             if(D.contains(Opl.intValue(data[2])) && Opl.floatValue(data[6]
185 ])>0) {
186                 StatisticsTable.add(data[0]//id
187                     ,Opl.intValue(data[1])//semana
188                     ,Opl.intValue(data[2])//dia
189                     ,Opl.intValue(data[3])//school
190                     ,Opl.floatValue(data[4])//corr
191                     ,Opl.floatValue(data[5])//min
192                     ,Opl.floatValue(data[6])//avg
193                     ,Opl.floatValue(data[7])//std
194                     ,Opl.floatValue(data[8])//max
195                     ,Opl.intValue(data[9])//num
196                     ,Opl.intValue(data[10]));//period_len
197             }//if avg > 0
198             }//if data.length > 0
199         }//while
200     }//execute
201
202 //Conjunto de puntos de extracción
203 {string} EP = { row.ep | row in StatisticsTable};
204
205 float DELTA_mu[D][EP];
206 float DELTA_sigma[D][EP];
207 execute DELTAS {

```



```

207     for( var row in StatisticsTable ) {
208         DELTA_mu[row.day][row.ep] = row.avg;
209         DELTA_sigma[row.day][row.ep] = row.std;
210     }
211 }
212
213 //Carga los datos de la tabla STRUCTURAL
214 {StructuralTuple} StructuralTable = {};
215 execute LOAD_STRUCTURAL {
216     var f = new IloOplInputFile("../data/opl_data/input/structural
217     .csv");
218     f.readline(); // Saltamos los encabezados
219     while (!f.eof) {
220         var data = f.readline();
221         if(data.length>0) {
222             data = data.split(",");
223             //Carga solamente los puntos de extraccion que estan en la
224             tabla statistics
225             if(EP.contains(data[0])) {
226                 StructuralTable.add(data[0] //id
227                                     ,data[1] //collection (name)
228                                     ,data[2] //module
229                                     ,data[3] //type
230                                     ,Opl.floatValue(data[4]) //priority
231                                     ,data[5] //frequency
232                                     ,data[6] //specific month
233                                     ,data[7]); //specific day
234             }
235         }
236     }
237 }
238
239 //{string} EP_id = { id | <id, collection, priority> in EP_table};
240
241 //Carga la tabla VISITS
242 {VisitsTuple} VisitsTable = {};
243 execute LOAD_VISITS {
244     var f = new IloOplInputFile("../data/opl_data/input/
245     must_visit_if_periodic.csv");
246     f.readline(); // Saltamos los encabezados
247     while (!f.eof) {
248         var data = f.readline();
249         if(data.length>0) {
250             data = data.split(",");
251             //Carga solamente los puntos de extraccion que estan en la
252             tabla statistics
253             if(EP.contains(data[0])) {
254                 VisitsTable.add(data[0] //ep
255                                ,Opl.intValue(data[1]) //periodic
256                                ,Opl.intValue(data[2])); //num_visits
257             }
258         }
259     }
260 }
261
262 //
263 int PERIODIC[EP];
264 int NUM_VISITS[EP];
265 execute PERIODIC_VISITS {
266     for(var row in VisitsTable) {
267         PERIODIC[row.ep] = row.periodic;
268     }
269 }

```

```

264     NUM_VISITS[row.ep] = row.num_visits;
265 }
266 }
267
268 //Carga los datos de la tabla SHIFT
269 {ShiftTuple} ShiftTable = {};
270 execute LOAD_SHIFT {
271     var f = new IloOplInputFile("../data/opl_data/input/shift.csv"
272 );
273 f.readline(); // Saltamos los encabezados
274 while (!f.eof) {
275     var data = f.readline();
276     if(data.length>0) {
277         data = data.split(",");
278         //Carga solamente los puntos de extraccion que estan en la
279         tabla statistics
280         if(EP.contains(data[0])) {
281             ShiftTable.add(data[0] //ep
282                 ,data[1]); //shift
283         }
284     }
285 }
286
287 //Conjunto de turnos
288 {string} SHIFT = { row.shift | row in ShiftTable};
289
290 //Carga los datos de la tabla ALPHA
291 {AlphaTuple} AlphaTable = {};
292 execute LOAD_ALPHA {
293     var f = new IloOplInputFile("../data/opl_data/input/var_alpha.
294 csv");
295 f.readline(); // Saltamos los encabezados
296 while (!f.eof) {
297     var data = f.readline();
298     if(data.length>0) {
299         data = data.split(",");
300         //Carga solamente los puntos de extraccion que estan en la
301         tabla statistics
302         // y si es un dia laborable. Ademas, interesan solamente las
303         filas donde lb = ub
304         if(EP.contains(data[1]) && D.contains(Opl.intValue(data[0]))
305         && Opl.intValue(data[4])==Opl.intValue(data[5])) {
306             AlphaTable.add(Opl.intValue(data[0]) //day
307                 ,data[1] //ep
308                 ,data[2] //bm
309                 ,data[3] //shift
310                 ,Opl.intValue(data[4]) //lb
311                 ,Opl.intValue(data[5]) //ub
312                 ,Opl.intValue(data[6])); // value
313         }
314     }
315 }
316
317 // Crea el conjunto de indices de alfa solamente con las
318 posibilidades de los ficheros de entrada
319 {AlphaIndexTuple} AlphaIndex = {<stats.day,stats.ep,bm.bm_id,shift.
320 shift> | stats in StatisticsTable,
321     shift in ShiftTable, bm in BloodmobilesTable:
322     stats.ep == shift.ep_id};

```

# Modelo OPL

Esta sección detalla la implementación del modelo inicial en OPL. En este apartado están las variables de decisión, las variables auxiliares, las restricciones y la función objetivo.

```
316 // *****
317 // DECLARACIÓN DE VARIABLES AUXILIARES
318 // *****
319
320 //Lectura de datos: de momento directamente del fichero .dat
321 //puede ser cambiado para leer ficheros CSV, incluso necesidad de
    personal
322 {string} Modulos = ...;
323 {string} Prevision = ...;
324
325 float costeVisita[Prevision][Modulos] = ...;
326
327 //variables auxiliares
328 int nbSegments = card(Prevision)-2;
329 float x [i in 0..nbSegments]= ...;
330 float yA[i in 0..nbSegments+1]=costeVisita[item(Prevision,i)][ "A"];
331 float yB[i in 0..nbSegments+1]=costeVisita[item(Prevision,i)][ "B"];
332 float yC[i in 0..nbSegments+1]=costeVisita[item(Prevision,i)][ "C"];
333 float yD[i in 0..nbSegments+1]=costeVisita[item(Prevision,i)][ "D"];
334
335 //declara finalmente las funciones lineales por partes, una por
    módulo
336 stepFunction fA = stepwise(i in 0..nbSegments){yA[i]->x[i];yA[
    nbSegments+1]};
337 stepFunction fB = stepwise(i in 0..nbSegments){yB[i]->x[i];yB[
    nbSegments+1]};
338 stepFunction fC = stepwise(i in 0..nbSegments){yC[i]->x[i];yC[
    nbSegments+1]};
339 stepFunction fD = stepwise(i in 0..nbSegments){yD[i]->x[i];yD[
    nbSegments+1]};
340
341 // *****
342 // DECLARACIÓN DE VARIABLES DE DECISIÓN
343 // *****
344
345 dvar int+ alpha[D][EP][BM][SHIFT] in 0..1; // Puntos seleccionados
    entre los elegibles
346 dvar int+ assetAssignment[D][BM] in 0..1; // Variable para
    indicar si se usa una unidad móvil un día o no.
347 dvar float+ gamma[D][EP][BM][SHIFT]; // Cantidad de bolsas
    recogidas diariamente por punto de extracción, turno y unidad
    móvil
348
349 // *****
350 // DECLARACIÓN DE VARIABLES ARTIFICIALES - SOFT CONSTRAINTS
351 // *****
352
353 // variables artificiales (soft constraints)
354 dvar int+ slackVisitsPeriodic[EP]; //C09
355 dvar int+ slackMinVisitsPerDay[D] in 0..2; //C10
356 dvar float+ slackSchoolVisits; //C12yC13
357 dvar int+ slackMorningVisits[D] in 0..2; //C14
358
359
360 // *****
```

```

361 // DECLARACIÓN DE EXPRESIONES DE DECISIÓN
362 // *****
363
364 //-----
365 // EXPRESIONES DEL MODELO
366 //-----
367
368 // Cantidad de bolsas recogidas diariamente por unidad móvil
369 dexpr float beta[d in D][bm in BM] = sum(ep in EP, shift in SHIFT)
    gamma[d][ep][bm][shift];
370 // Total de bolsas recogidas por día
371 dexpr float wbb[d in D] = sum(bm in BM)beta[d][bm];
372
373 //-----
374 // CÁLCULO DE PENALIDADES
375 //-----
376
377 // Cantidad de visitas no atendidas para puntos de extracción
    periódicos
378 dexpr int penaltyVisitPeriodic = sum(ep in EP)slackVisitsPeriodic[
    ep];
379
380 // Cantidad de días sin el mínimo de visitas por la mañana
381 dexpr int penaltyMorningVisits = sum(d in D)(slackMorningVisits[d
    ]>=1);
382
383 // Cantidad de días sin el mínimo de visitas
384 dexpr int penaltyMinVisitsPerDay = sum(d in D)(slackMinVisitsPerDay
    [d]>=1);
385
386 //-----
387 // CÁLCULO DE ESTADÍSTICAS
388 //-----
389
390 // Visitas a escuelas en el mes
391 dexpr int schoolvisits = sum(d in D, row in StructuralTable, bm in
    BM, shift in SHIFT : row.priority==1)alpha[d][row.ep_id][bm][
    shift];
392 // Total de visitas realizadas en el mes
393 dexpr int totalvisits = sum(d in D, ep in EP, bm in BM, shift in
    SHIFT)alpha[d][ep][bm][shift];
394
395
396 // Total de bolsas demandadas en el mes
397 dexpr float totalDemand = sum(row in CalendarTable)row.demand;
398 // Total de bolsas recogidas en el mes
399 dexpr float totalCollected = sum(d in D)wbb[d];
400
401 // Cantidad máxima de unidades móviles disponibles por día
402 //dexpr int maxbm = max1(card(BM_id)-1,ftoi(ceil(minvisits/card(D))
    ));
403 // Cantidad de visitas a puntos de extracción periódicos
404 dexpr int minvisits = sum(ep in EP)NUM_VISITS[ep]*PERIODIC[ep];
405 // Cantidad de visitas por punto de extracción
406 dexpr int visits[ep in EP] = sum(d in D, bm in BM, shift in SHIFT)
    alpha[d][ep][bm][shift];
407 // Cantidad de visitas por punto de extracción
408 dexpr float visitsA = sum(row in AlphaIndex, row2 in
    StructuralTable : row.ep_id==row2.ep_id && row2.module== "A")fA(
    gamma[row.day][row.ep_id][row.bm_id][row.shift]);

```

```

409 dexpr float visitsB = sum(row in AlphaIndex, row2 in
    StructuralTable : row.ep_id==row2.ep_id && row2.module== "B")fB(
    gamma[row.day][row.ep_id][row.bm_id][row.shift]);
410 dexpr float visitsC = sum(row in AlphaIndex, row2 in
    StructuralTable : row.ep_id==row2.ep_id && row2.module== "C")fC(
    gamma[row.day][row.ep_id][row.bm_id][row.shift]);
411 dexpr float visitsD = sum(row in AlphaIndex, row2 in
    StructuralTable : row.ep_id==row2.ep_id && row2.module== "D")fD(
    gamma[row.day][row.ep_id][row.bm_id][row.shift]);
412 dexpr float visitCosts = visitsA + visitsB + visitsC + visitsD;
413
414 // *****
415 // FUNCIÓN OBJETIVO
416 // *****
417
418 minimize
419 // sum(d in D, ep in EP, bm in BM, shift in SHIFT)alpha[d][ep][bm
    ][shift];
420 sum(d in D)abs(en[d]-wbb[d]) + 10000*(penaltyMinVisitsPerDay +
    penaltyMorningVisits +
421     penaltyVisitPeriodic + slackSchoolVisits);
422
423 // *****
424 // RESTRICCIONES
425 // *****
426 subject to{
427
428     //-----
429     // UTILITARIAS
430     //-----
431
432     // Considera el input del usuario para alfa
433     forall(row in AlphaTable : row.lb == row.ub){
434         cstAlphaUserInput: alpha[row.day][row.ep_id][row.bm_id][row.
            shift] == row.ub;
435     }
436
437     //-----
438     // LÓGICAS Y DE CONTROL
439     //-----
440
441     // C1: Las combinaciones que no se aplican tendrán alfa = 0.
442     forall(d in D, ep in EP, bm in BM, shift in SHIFT: <d,ep,bm,shift
        > not in AlphaIndex) {
443         cstSparcityControl: alpha[d][ep][bm][shift] == 0;
444     }
445
446     // C2: Los puntos de extracción que no son elegibles (no han sido
        visitados
447     // en el pasado por estas fechas) tendrán alfa=0.
448     forall(d in D, ep in EP : DELTA_mu[d][ep] == 0) {
449         forall(bm in BM, shift in SHIFT) {
450             cstNotApplicable: alpha[d][ep][bm][shift] == 0;
451         }
452     }
453
454     //Se puede hacer una visita solamente si hay una unidad movil
        assignada
455     forall(d in D, bm in BM, ep in EP, shift in SHIFT){
456         cstBMAssignment: alpha[d][ep][bm][shift] <= assetAssignment[d
            ][bm];

```

```

457 }
458
459 // C8: Una unidad móvil va solamente a un EP por turno
460 forall(d in D, bm in BM, shift in SHIFT) {
461     cstMaxBMVisitsPerShift:
462         sum(ep in EP)(alpha[d][ep][bm][shift]) <= 1;
463 }
464
465 // C15New: Puede haber solamente 1 visita por EP al día
466 forall(d in D, ep in EP) {
467     cstMaxVisitsPerExtractionPoint:
468         sum(bm in BM, shift in SHIFT)(alpha[d][ep][bm][shift]) <= 1;
469 }
470
471 //-----
472 // FÍSICAS (capacidad)
473 //-----
474
475 // C3 y C4: Definición de gamma
476 forall(d in D, ep in EP, bm in BM, shift in SHIFT) {
477     cstMinQuantityCollected:
478         gamma[d][ep][bm][shift] >= (alpha[d][ep][bm][shift]*(DELTA_mu
479         [d][ep]-DELTA_sigma[d][ep]));
480     cstMaxQuantityCollected:
481         gamma[d][ep][bm][shift] <= (alpha[d][ep][bm][shift]*(DELTA_mu
482         [d][ep]+DELTA_sigma[d][ep]));
483 }
484
485 // C5 y C6: Capacidad de las unidades móviles
486 forall(row in BloodmobilesTable, d in D) {
487     cstBMCapacity: row.cb_min <= beta[d][row.bm_id] <= row.cb_max;
488 }
489
490 //-----
491 // REGLAS DEL NEGÓCIO
492 //-----
493
494 // C9: Cada unidad móvil bm visitará cada punto de extracción ep
495 // numvisits veces
496 // en todos los días d del mes.
497 forall(ep in EP : NUM_VISITS[ep]>0 && PERIODIC[ep]==1) {
498     cstNumberOfVisitsPeriodic:
499         sum(d in D, bm in BM, shift in SHIFT)alpha[d][ep][bm][shift]
500         + slackVisitsPeriodic[ep]== NUM_VISITS[ep];
501 }
502
503 forall(ep in EP : NUM_VISITS[ep]>0 && PERIODIC[ep]==0) {
504     cstNumberOfVisitsNotPeriodic:
505         sum(d in D, bm in BM, shift in SHIFT)alpha[d][ep][bm][shift]
506         <= NUM_VISITS[ep];
507 }
508
509 // C10: Cada da laborable needCorrFactor se deben visitar, al
510 // menos, 2 puntos
511 // de extracción ep diferentes --> transformada en soft
512 // constraint
513 forall(d in D) {
514     cstMinVisitsPerDay: sum(ep in EP, bm in BM, shift in SHIFT)
515     alpha[d][ep][bm][shift] + slackMinVisitsPerDay[d] >= MIN_VISITS[
516     d]; //2
517 }
518

```

```

509 // C11a: Cada día laborable needCorrFactor se deben visitar como
      mucho BM
510 // puntos de extracción ep diferentes.
511 forall(d in D) {
512     cstMaxEPVisitsPerDay: sum(ep in EP, bm in BM, shift in SHIFT)
      alpha[d][ep][bm][shift] <= MAX_VISITS[d]; //card(BM)
513 }
514 // C11b: Cada turno en un día laborable needCorrFactor se deben
      visitar como mucho BM-1
515 // puntos de extracción ep diferentes.
516 forall(d in D, shift in SHIFT) {
517     cstMaxEPVisitsPerShift: sum(ep in EP, bm in BM)alpha[d][ep][bm
      ][shift] <= MAX_VISITS_SHIFT[d]; //card(BM)-1
518 }
519
520 // C11c: Cada día laborable needCorrFactor se deben usar como
      mucho BM-1 unidades
521 // móviles.
522 forall(d in D) {
523     cstMaxBMPerDay: sum(bm in BM)assetAssignment[d][bm]<= MAX_BM[d
      ];//card(BM)-1;
524 }
525
526 // C8tarde: las unidades móviles pueden hacer una segunda visita
      si el EP del turno de tarde
527 // está en un módulo A
528 forall(d in D, bm in BM) {
529     cstAfternoonSecondVisit:
530         sum(ep in EP, shift in SHIFT : shift=="M")alpha[d][ep][bm][
      shift] +
531         sum(row in StructuralTable, shift in SHIFT : shift=="T" &&
      row.module == "A")alpha[d][row.ep_id][bm][shift] <= 2;
532 }
533
534 // C12yC13: La proporción de visitas a centros educativos/(todos
      los días
535 // lectivos) debe ser >= 10%. Los puntos de extracción ep
      relativos a
536 // centros educativos (priority=1) tienen que tener, para los
      meses
537 // lectivos, una proporción de al menos high=10% y para los meses
      no lectivos
538 // una proporción de al menos low=5%.
539     cstSchoolVisitsPercentage: schoolvisits + slackSchoolVisits >=
      TARGET*totalvisits;
540
541 // C14: Cada día d debe haber, al menos, 2 salidas en el turno
      shift de mañana.
542 forall(d in D, shift in SHIFT : shift=="M") {
543     cstMorningVisits: sum(ep in EP, bm in BM)alpha[d][ep][bm][shift
      ] + slackMorningVisits[d] >= 2;
544 }
545
546 }

```

## Posprocesado

Esta sección muestra cómo se presentan los resultados al usuario y cómo se generan los archivos de salida en formato CSV.

```

547 // *****
548 //  SCRIPTS
549 //  *****
550
551 //  ALFA
552 execute WRITE_ALPHA {
553     var f = new IloOplOutputFile("../data/opl_data/output/
554         var_alpha.csv");
555     f.writeln("D,EP_id,BM_id,SHIFT,lb,ub,value");//guardando sin lb y
556         ub
557     for(var row in AlphaIndex){
558         f.writeln(row.day + "," + row.ep_id + "," + row.bm_id + "," +
559             row.shift + ",0,1" + "," + alpha[row.day][row.ep_id][row.bm_id][
560                 row.shift]);
561     }
562     f.close();
563 }
564
565 //  BETA
566 execute WRITE_BETA {
567     var f = new IloOplOutputFile("../data/opl_data/output/var_beta
568         .csv");
569     f.writeln("D,BM_id,lb,ub,value");//guardando con lb y ub
570     for(var d in D)
571         for(var bm in BloodmobilesTable) {
572             f.writeln(d + "," + bm.bm_id + "," + bm.cb_min + "," + bm.
573                 cb_max + "," + beta[d][bm.bm_id]);
574         }
575     f.close();
576 }
577
578 //  GAMMA
579 execute WRITE_GAMMA {
580     var f = new IloOplOutputFile("../data/opl_data/output/
581         var_gamma.csv");
582     f.writeln("D,EP_id,BM_id,SHIFT,value");
583     for(var row in AlphaIndex){
584         f.writeln(row.day + "," + row.ep_id + "," + row.bm_id + "," +
585             row.shift + "," + gamma[row.day][row.ep_id][row.bm_id][row.shift
586                 ]);
587     }
588     f.close();
589 }
590
591 // Sacar los resultados por pantalla de manera amigable
592 execute DISPLAY{
593     var f = new IloOplOutputFile("../data/opl_data/output/
594         posprocesado.txt");
595
596     // Info del período
597     f.writeln("INFORMACIONES GENERALES: ")
598     f.writeln("");
599     f.writeln("Período: " + YEAR + "-" + MONTH);
600     f.writeln("Lectivo: " + SCHOOL);
601     f.writeln("Días: " + D.size);
602     f.writeln("Puntos de extracción: " + EP.size);
603     f.writeln("Unidades móviles: " + BM.size);
604     f.writeln("Turnos: " + SHIFT.size);
605     f.writeln("Visitas a puntos periódicos: " + minvisits);
606     f.writeln("-----");
607     f.writeln("DIAGNÓSTICO DE NO FACTIBILIDAD: ")

```



```

598 f.writeln("");
599 var infactible = false;
600 var nbConstraints = 0;
601 // Verifica si cumple con la restricción de visitas a puntos
    periódicos
602 if (penaltyVisitPeriodic > 0){
603     f.writeln("Visitas periódicas no atendidas: " +
        penaltyVisitPeriodic);
604     infactible = true;
605     nbConstraints = nbConstraints + 1;
606 }
607 // Verifica si cumple con la restricción de mínimo de visitas por
    la mañana
608 if (penaltyMorningVisits > 0){
609     f.writeln("Cantidad de días sin el mínimo de visitas por la
        mañana: " + penaltyMorningVisits);
610     infactible = true;
611     nbConstraints = nbConstraints + 1;
612 }
613 // Verifica si cumple con la restricción de mínimo de visitas por
    día
614 if (penaltyMinVisitsPerDay > 0){
615     f.writeln("Cantidad de días sin el mínimo de visitas por día: "
        + penaltyMinVisitsPerDay);
616     infactible = true;
617     nbConstraints = nbConstraints + 1;
618 }
619 // Verifica si cumple con la restricción de visitas a escuelas
620 if (slackSchoolVisits > 0){
621     f.writeln("Diferencia de visitas a escuelas: " +
        slackSchoolVisits);
622     infactible = true;
623     nbConstraints = nbConstraints + 1;
624 }
625 // Verifica si cumple con la restricción de visitas a escuelas
626 if (infactible == true){
627     f.writeln("La solución obtenida viola " + nbConstraints + "
        restriccion(es) blanda(s) del modelo.");
628 } else {
629     f.writeln("La solución obtenida satisface todas las
        restricciones del modelo.");
630 }
631
632 // Función para redondear y controlar la cantidad de dígitos
    presentados
633 function roundTo(x,digits){
634     x = x*Opl.pow(10,digits);
635     x=Opl.round(x);
636     x=x/Opl.pow(10,digits);
637     return x;
638 }
639
640 f.writeln("-----");
641 f.writeln("ESTADÍSTICAS: ")
642 f.writeln("");
643 // Valor de la función objetivo
644 f.writeln("Valor de la función objetivo: " + cplex.getObjValue())
    ;
645 // Demanda total
646 f.writeln("Demanda total: " + totalDemand);
647 // Total de bolsas recogidas

```

```

648 f.writeln("Total de bolsas recogidas: " + totalCollected);
649 // Diferencia
650 var gap = ((totalCollected-totalDemand)/totalDemand)*100
651 f.writeln("Diferencia: " + roundTo(gap,2) + "%");
652 // Total de visitas
653 f.writeln("Total de visitas: " + totalvisits);
654 // Visitas a escuelas
655 f.writeln("Total de visitas a centros educativos: " +
    schoolvisits);
656 // Porcentaje de visitas a escuelas
657 var kpi = (schoolvisits/totalvisits)*100;
658 f.writeln("Porcentaje de visitas a escuelas: " + roundTo(kpi,2) +
    "%");
659 // Coste total de personal
660 f.writeln("Coste total de personal: " + roundTo(visitCosts,2));
661 f.close();
662 }

```