

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UNA  
HERRAMIENTA GRÁFICA PARA AYUDA A LA  
DECISIÓN DE UN SISTEMA DE OPTIMIZACIÓN  
DE PROCESOS EN CENTROS DE  
HEMODONACIÓN

PABLO RAMOS IZQUIERDO

JUNIO 2020



# Trabajo Fin de Grado

**Título:** DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA  
GRÁFICA PARA AYUDA A LA DECISIÓN DE UN SISTEMA DE  
OPTIMIZACIÓN DE PROCESOS EN CENTROS  
DE HEMODONACIÓN

**Autor:** PABLO RAMOS IZQUIERDO

**Tutor:** DR. JOSUÉ PAGÁN ORTIZ

**Departamento:** DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

## Miembros del tribunal

**Presidente:** D.

**Vocal:** D.

**Secretario:** D.

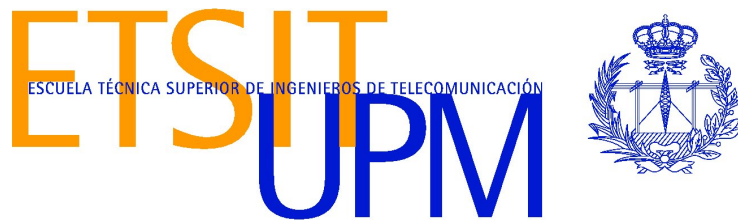
**Suplente:** D.

Madrid, a        de        de



UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

**Diseño e implementación de una  
herramienta gráfica para ayuda  
a la decisión de un sistema de  
optimización de procesos en  
centros de hemodonación**

PABLO RAMOS IZQUIERDO

18 DE JUNIO DE 2020



# Resumen del proyecto

## Resumen

La sangre es una sustancia principal para la supervivencia del ser humano. Al no ser posible su sintetización artificial a día de hoy, la única posibilidad de generación es de forma natural por el cuerpo. Sin embargo, hay ocasiones que se necesita una transfusión de sangre de una persona sana para un paciente para que pueda sobrevivir o para la generación de vacunas o medicamentos capaces de combatir enfermedades.

Para la generación de estas vacunas o medicamentos se necesita el plasma que tiene la sangre. La forma de conseguir este plasma se realiza por dos medios distintos: por plasmaféresis, proceso por el cual se extrae el plasma directamente, o por fragmentación de los elementos de la sangre para obtener el plasma. El primer método es caro, requiere de un material muy específico y de personal cualificado para la tarea. El segundo método, por el contrario, es el más empleado pero menos eficiente.

Debido a esta necesidad de obtener plasma para salvar vidas, existen centros de hemodonación que su trabajo consiste en la realización de campañas de donación de sangre para que las personas, en un acto de solidaridad, donen su propia sangre sana para ayudar a quien lo necesite. Este proceso requiere de una planificación previa para poder visitar un lugar en particular donde realizar la campaña apropiada donde existen muchos factores como la elección de una fecha en particular donde pueda haber más donantes, la necesidad de contar con cierto número de personal para la tarea, publicitar a los donantes de la zona los días anteriores de nuestra intención de visitar el lugar que frecuentan o el desplazamiento a una zona con el material y equipo necesario.

Es por ello que este TFG, se ha desarrollado de un software que se empleará como una interfaz gráfica, para comunicación con un sistema de optimización, donde se puedan visualizar las planificaciones que debe realizar nuestro administrador del centro de hemodonación para facilitarle el trabajo y por ende, mejorar la eficiencia.

## Abstract

Blood is a main substance for human survival. Since its artificial synthesis is not possible today, the only way of generation is naturally by the body. However, there are occasions when a healthy person's blood transfusion is needed for a patient in order to survive or for the generation of vaccines or drugs capable of fighting diseases.

For the generation of these vaccines or medicines, the plasma that is in the blood is needed. The way to obtain this plasma is carried out by two different ways: by plasmapheresis, a process by which the plasma is extracted directly, or by fragmentation of the elements of the blood to obtain the plasma. The first method is expensive, requires very specific material and qualified personnel for the task. The second method, on the contrary, is the most used but least efficient.

Due to this need to obtain plasma to save lives, there are hemodialysis centers whose job is to carry out blood donation campaigns so that people, in an act of solidarity, donate their own healthy blood to help those who need it. . This process requires prior planning in order to visit a particular place where to carry out the appropriate campaign where there are many factors such as choosing a particular date where there may be more donors, the need to have a certain number of personnel for the task, publicize to the donors of the area the previous days of our intention to visit the place they frequent or the displacement to an area with the necessary material and equipment.

In this bachelor's thesis it has been developed of a software that will be used as a graphical user interface that communicates with an optimization system to visualize the scheduling of a blood donation center to facilitate the work to the center's administrator and thus improve the efficiency of the blood donation center.

## Palabras clave

Intefaz gráfica, Centro hemodonación, planificación, optimización, eficiencia, campaña de donación, donantes

## Keywords

Graphic User Interface, GUI, Blood donation center, planification, optimization, effiency, donation campaign, donnors



# Agradecimientos

*Quiero agradecer este Trabajo de Fin de Carrera a todas las personas que me acompañan día a día y que forman parte de mi vida. En mención especial a mis padres Jesús y María Paula y a mi hermano Víctor, ya que soy la persona que soy gracias a ellos. A mis abuelos Ejé, Carmen y María, mis tíos, primos, amigos y en particular a mi tutor Josué por darme la oportunidad de participar en dicho proyecto del que tanto he aprendido, disfrutado, cabreado y todo tipo de sentimientos.*

*Gracias.*







# Lista de Acrónimos

**BBDD** Base de Datos

**CSV** Comma-Separated Values

**EBSN** Event-Based Social Networks

**GUI** Graphical User Interface (Interfaz Gráfica de Usuario)

**HTML** HyperText Markup Language

**HTTP** HyperText Transfer Protocol

**IDE** Integrated Development Environment

**JSON** JavaScript Object Notation

**PDF** Portable Document Format

**PDMP** Plasma Derived Medicinal Products

**RRSS** Redes Sociales

**SQL** Structured Query Language

**TFG** Trabajo Fin de Grado

**XLSX** Excel Microsoft Office Open XML Format Spreadsheet file



# Índice general

<b>Resumen del proyecto</b>	<b>I</b>
<b>Agradecimientos</b>	<b>III</b>
<b>Lista de Acrónimos</b>	<b>VII</b>
<b>1. Introducción y objetivos</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Objetivos . . . . .	3
<b>2. Estado del arte</b>	<b>5</b>
<b>3. Metodología</b>	<b>9</b>
3.1. Arquitectura . . . . .	12
3.2. Preparación de los datos . . . . .	14
3.3. Construcción y llenado de la BBDD . . . . .	17
3.4. Implementación de la GUI . . . . .	19
3.5. Comunicación entre Sistemas . . . . .	22
<b>4. Implementación</b>	<b>27</b>
4.1. BBDD . . . . .	27
4.2. GUI . . . . .	29
<b>5. Resultados</b>	<b>31</b>
5.1. Proceso de instalación . . . . .	31
5.2. Graphical User Interface - GUI . . . . .	34

5.2.1. Inicio . . . . .	35
5.2.2. Planificación . . . . .	39
5.2.3. Publicidad . . . . .	41
5.2.4. Hoja de firmas . . . . .	42
5.2.5. Otras pestañas . . . . .	45
 <b>6. Conclusiones y líneas de futuro</b>	 <b>47</b>
 <b>Bibliografía</b>	 <b>49</b>
 <b>I Apéndices</b>	 <b>51</b>
 <b>A. Campos de Ficheros necesarios para la Instalación</b>	 <b>53</b>
 <b>B. Campos de las Tablas de Datos</b>	 <b>59</b>
 <b>C. Aspectos éticos, económicos, sociales y ambientales</b>	 <b>69</b>
 <b>D. Presupuesto económico</b>	 <b>71</b>



# Índice de figuras

3.1. Etapas del desarrollo del proceso de recolección . . . . .	9
3.2. Arquitectura del sistema . . . . .	13
4.1. Arquitectura de la implementación del sistema . . . . .	27
4.2. Logo del sistema gestor MariaDB . . . . .	28
4.3. Logo de la herramienta phpMyAdmin . . . . .	28
4.4. Logo del IDE Eclipse . . . . .	29
4.5. Logo del lenguaje Java . . . . .	29
4.6. Herramienta de desarrollo para GUI en Eclipse . . . . .	30
5.1. Esquema del proceso de Instalación de la herramienta . . . . .	32
5.2. Interfaz del buscador de ficheros . . . . .	33
5.3. Contenido de la tabla locations de la BBDD . . . . .	34
5.4. Contenido de la tabla vehicles de la BBDD . . . . .	34
5.5. Captura de pantalla pestaña Inicio de la GUI . . . . .	35
5.6. Captura de pantalla de JSON enviado por el optimizador . . . . .	36
5.7. Captura de pantalla información de BBDD en GUI . . . . .	38
5.8. Captura de pantalla pestaña Planificación Interfaz Gráfica . . . . .	39
5.9. Captura de pantalla de pestaña <i>Planificación</i> del evento Forzar Ir y Editar . . . . .	40
5.10. Captura de pantalla del evento del botón <i>No Ir</i> en la pestaña Planificación . . . . .	41
5.11. Captura de pantalla de pestaña <i>Planificación</i> del evento Bloquear y Validar . . . . .	41
5.12. Captura de pantalla pestaña Publicidad Interfaz Gráfica . . . . .	42

5.13. Ejemplo de cartel publicitario devuelto por sistema optimizador . . .	43
5.14. Captura de pantalla del evento del botón <i>ENVIAR TELEGRAMS</i> en la pestaña Publicidad . . . . .	44
5.15. Captura de pantalla pestaña Hoja de firmas GUI . . . . .	44

# Índice de cuadros

D.1. Coste de los recursos materiales . . . . .	71
D.2. Coste de mano de obra . . . . .	72
D.3. Coste total del proyecto . . . . .	72
D.4. Gastos generales y beneficio industrial . . . . .	72
D.5. Presupuesto total . . . . .	73



# CAPÍTULO 1

## Introducción y objetivos

### 1.1. Introducción

Hoy en día, la sangre es una sustancia que no podemos sintetizar de forma artificial y es necesaria para la supervivencia del ser humano. Se calcula que una persona tiene de media entre 4,5 y 6,5 litros de sangre en el cuerpo, representando en consecuencia el 7,7 % del peso corporal total. Para conseguir dicha sustancia, se realizan donaciones de sangre de personas sanas con el fin de poder salvar las vidas de las personas. El uso de esta sangre recogida de los donantes se puede emplear para realizar transfusiones de sangre, para dársela a un paciente que se encuentre en niveles de sangre inferiores a la necesaria para sobrevivir y su propio cuerpo no pueda generarla rápidamente de forma natural, o, para poder conseguir desarrollar medicamentos o vacunas para luchar contra las enfermedades que existen en nuestro día a día como, lamentablemente, estamos viviendo en esta etapa donde el mundo entero se ha volcado en la lucha de poner fin a la COVID-19.

Cuando se procede a realizar una extracción voluntaria de sangre del donante, existen unos requisitos que la persona que va a donar debe cumplir. Para que un donante pueda hacerlo debe tener entre 16 y 65 años, pesar más de 50 kg, o que no haya padecido determinadas enfermedades, como la hepatitis A, B o C, para no poner en riesgo la salud del enfermo que se le hace la transfusión, o en el caso de el desarrollo de vacunas o fármacos, para que no se alteren los resultados de las pruebas. Cuando una persona dona, se le extraen de media 450 ml de sangre.

Añadido al problema de no poder sintetizar sangre de manera artificial, se suma el suceso de que la demanda de los productos médicos derivados del plasma (PMDP) [3] se ha disparado de forma considerable en la industria farmacéutica, con lo que conlleva a un problema serio en los centros de hemodonación porque deben incrementar la recolección del plasma. Éste se puede extraer de dos formas distintas:

1. Por plasmaféresis: Extracción directa del plasma. Este proceso es caro, lento y requiere de un material muy específico para su realización, por lo que es el menos utilizado.

2. Por donaciones de sangre total: Este método es el que usan los centros de hemodonación para la recolección de dicho plasma, siendo un método mucho más barato aunque menos eficiente que el anterior, ya que, solo el 45 % de la sangre extraída corresponde al plasma deseado.

Al haber más demanda, los centros de hemodonación intensifican las campañas para recolectar sangre aunque las necesidades de transfusión para pacientes no sea tan elevada, lo que trae consigo un exceso en las existencias que procederá al descarte de productos debido a su caducidad. Si nos encontramos en la situación contraria, donde la necesidad de la recolección de la sangre es debida a la necesidad de transfusión para pacientes, hallamos la suma de dos problemas, que es la captación de donantes suficientes y necesarios para ambas necesidades.

Para poder realizar las campañas de hemodonación no siempre se hace de forma eficiente debido a que se encargar de realizar dicha planificación, o ruta diaria, una persona administrativa que debe tener en cuenta diversas variables. Para conseguir esta optimización se ha desarrollado un modelo de optimización primigenio integrado en un sistema — backend — optimizador. Este sistema está siendo probado con datos reales del Centro Regional de Hemodonación de Murcia, pero para que pueda entrar en la rutina de trabajo habitual del administrativo del centro, se necesita desarrollar una Interfaz Gráfica (GUI, Graphical User Interface) amigable que sirva de intermediario entre este el administrativo del centro y el sistema optimizador que corre en un servidor.

Además, desde el grupo de investigación, se están planteando nuevos modelos de optimización integrados al anterior que permitan, por ejemplo, realizar campañas de promoción eficientes, y analizar el impacto humano en las pérdidas de donación de sangre; tarea ahora imposible porque el registro de errores o rechazos solo se realiza en papel, y su información no se digitaliza ni se analiza. Actualmente el trabajo de recolección de unidades de sangre se divide en 4 procesos: Ofrecimiento, Donación, Transporte y Almacenamiento/Uso. Para simplificar la distribución de tareas a realizar, nos dispondremos inicialmente a cambiar el planteamiento de esos 4 procesos para expandirlos en 11 [4] más concisos. Estos procesos con los que vamos a operar son los siguientes:

1. Inicio: Selección de la fecha de visita del Punto de Colecta.
2. Planificación: Lista de las características que tiene cada Punto de Colecta, con número de trabajadores, lugar de visita, bolsas esperadas. . .
3. Publicidad: Medios para publicitar a las personas que frecuentan el lugar nuestra intención de ir a realizar la campaña de donación de sangre.
4. Transporte: Desplazamiento del equipo de trabajo al Punto de Colecta.
5. Ofrecimientos: Acciones que debe realizar el donante cuando hace acto de presencia en el punto de colecta.
6. Donaciones: Proceso que abarca desde que el donante se presenta en el punto de colecta hasta que la sangre queda embolsada.

7. Transporte: Desplazamiento de retorno del equipo de trabajo al centro de hemodonación.
8. Procesamiento: Proceso que tiene en cuenta el control de las condiciones sanitarias en las que se encuentra el material y el fraccionamiento en sí de la sangre en subproductos.
9. Almacenamiento: Proceso que mide el nivel de aprovisionamiento de los almacenes calculadas mediante la capacidad y la ocupación.
10. Distribución: Transporte de las bolsas de sangre desde el almacén hasta el receptor.
11. Hoja de firmas: Herramienta que emplean los trabajadores del centro para indicar anotaciones de cómo se ha llevado el proceso de extracción una vez visitado el lugar.

Cada uno de estos procesos puede ser optimizado [6] en alguno de sus aspectos. Además de las propuestas dadas por el sistema optimizador [4] al administrativo del centro, este empleado también puede querer realizar modificaciones en las soluciones dadas. Debemos dar la posibilidad al administrador de poder realizar peticiones de visita a lugares de colecta, editar las propuestas que se nos han dado o incluso eliminarlas. Para poder recolectar las donaciones de los pacientes, se dispone de un equipo formado por varios expertos, como médicos o enfermeros, que se encargarán de la extracción de la sangre, del correcto almacenamiento y del transporte entre diversas tareas.

## 1.2. Objetivos

Nuestro proyecto consistirá en el desarrollo de una GUI capaz de servir de intermediador entre el administrador del centro de hemodonación y el sistema optimizador. Para ello, debemos implementar un sistema de comunicación sencillo y escalable cliente-servidor, que sea capaz de enviar información de manera simple entre la GUI y el sistema optimizador con las propuestas enviadas para las rutas que debe hacer nuestro equipo de trabajo.

Las características que deberá tener nuestra GUI son las siguientes:

- Nuestra interfaz debe tener una apariencia amigable y no estar sobrecargada de opciones para la visualización de datos relativos al sistema de predicción y optimización.
- Debe ser capaz de realizar consultas a bases de datos (BBDD), para obtener información de interés para el operador que maneje la interfaz.
- Tiene que poder recoger y manejar restricciones deseadas por los responsables del centro de hemodonación y para la optimización de las propuestas de planificación de colectas que son recibidas.

- La interfaz debe poder ejecutarse en un sistema independiente del sistema operativo de la máquina cliente.
- Debe tener la capacidad de comunicarse con el backend de optimización para hacer test y pruebas de concepto en entorno real.

Además del desarrollo de la GUI mencionada, deberemos desarrollar una BBDD que contenga toda la información necesaria para que se puedan realizar las planificaciones de colectas correspondientes. En previsión de que esta herramienta pueda ser usada en otros centros de hemodonación, incluso extranjeros, la BBDD debe dar soporte multi-lenguaje al almacenar y presentar su información.

En este Capítulo 1 se ha descrito una introducción previa del problema existente a resolver y los objetivos que debemos cumplir para resolverlos. En la estructura del TFG tendremos dividida por capítulos donde el Capítulo 4 será el capítulo encargado de explicar las soluciones que podemos encontrar en la actualidad y cómo vamos a mejorarlas, el Capítulo 3 se encargará de explicar la metodología que tenemos que aplicar para cumplir los objetivos, el Capítulo 4 donde contaremos cómo tenemos que implementar el programa a desarrollar. En los capítulos finales, en el Capítulo 5 mostraremos al lector los resultados obtenidos al realizar nuestro trabajo y para terminar el Capítulo 6 donde procederemos a explicar las conclusiones finales y las líneas futuras que pueden existir.

Este TFG (Trabajo Fin de Grado) se realiza en colaboración con el equipo de investigación del Centro de Hemodonación de la Región de Murcia, quienes proponen el comportamiento, las características y el aspecto de la herramienta para ser utilizada en un entorno real.



## CAPÍTULO 2

# Estado del arte

En este capítulo se va a realizar un análisis de las soluciones que existen en la actualidad para satisfacer los objetivos del proyecto. Como hemos explicado en el Capítulo 1, nuestro TFG se encargará, entre muchas tareas, del desarrollo de una interfaz gráfica que pueda mostrar de forma sencilla al usuario que emplee nuestro programa las propuestas de planificación que deben realizar los centros de hemodonación para optimizar los recursos del centro y así aumentar su eficiencia de bolsas de sangre recogidas [4]. Además de estas propuestas, el sistema optimizador también es capaz de mostrar los procesos que se producen en el acto de recolección de bolsas de sangre. El número de procesos totales en los que se ha dividido la carga de trabajo es de 11 etapas que hemos explicado sus características en el Capítulo 1.

A día de hoy no existen software de administración de centros de hemodonación para optimizar los recursos y las colectas de donación para aumentar la eficiencia de este trabajo. Lo que podemos encontrar en el mercado son gestores de los bancos de sangre—entendido el banco como el seguimiento del donante y la sangre *per se*—como puede ser el caso del software *Hematos IIG* [5] o del programa *ePROGESA* [7]. Utilizaremos estos dos ejemplos para conseguir información de interés de ellos que nos puedan dar ideas a la hora de desarrollar nuestro TFG. La desventaja de estos programas es que funcionan como gestores del banco, como hemos dicho anteriormente, en vez de administradores del centro—la parte operativa—, como desearíamos, por lo que no nos servirán para cumplir el objetivo principal del trabajo.

Para facilitar la comprensión del lector iremos describiendo las tecnologías de referencia en las que nos basaremos para cada una de las etapas de la división de trabajo en las que nos centraremos en este proyecto.

### Inicio

El reto principal que se enfrenta la etapa de *Inicio* es tener la capacidad de seleccionar la información que se necesita en cada una de las demás etapas. Como su labor será la de seleccionar fechas, no podremos tomar de referencia cualquiera

de los dos programas mencionados antes ya que ellos solo gestionan los bancos de sangre.

En esta ocasión podemos basarnos en otras herramientas que hay en la web para la selección de fechas. En ese caso podemos coger de referencia el calendario de Google <sup>1</sup> para poder desarrollar la herramienta de selección.

## Planificación

La etapa de *Planificación* será en la que se centre nuestro proyecto prioritariamente. En él se mostrarán las planificaciones existentes [4] de colectas comprendidas en periodos de un mes de duración, sin embargo, al apreciar que las referencias que tenemos no son capaces de realizar las funciones de planificación no se ha realizado exploración en ese sentido, si no que nos hemos basado en la necesidad real que existe en el Centro de Hemodonación de la Región de Murcia, para que el diseño sea lo más amigable posible y sea de agrado para los empleados que vayan a manipular nuestro programas.

## Publicidad

La publicidad desde siempre es necesaria para poder hacer llegar a la población información sobre tareas que se van a realizar o para las empresas, su medio de salvación para llegar sus productos a los compradores. La organización de una buena campaña de comunicación es otro de los grandes retos que se presenta cualquier empresa o colectivo. Dirigir la campaña de forma precisa y establecer los medios adecuados que mejor se adapten a las circunstancias y necesidades de cada empresa son incógnitas que los expertos deben manejar para cumplir con sus objetivos.

El artículo de Baggi et al. [1] presenta una comparación entre los resultados obtenidos tras una campaña que se han realizado mensajes publicitarios a la población frente a los resultados obtenidos mediante el boca a boca. Para realizar la comparación modela las dos alternativas y se produce a realizar una simulación en un entorno virtualizado. Para el caso de la campaña de envío de mensajes utilizan el Modelo Magnético Eden, propuesto por Eden (1961) para describir el crecimiento de colonias bacterianas. La formulación matemática realizada en base a la generalización y adaptación del Modelo de Eden a situaciones estocásticas, sirve para fundamento para futuras investigaciones en las que se compruebe el rendimiento de campañas de este tipo. En lo relacionado al efecto boca a boca en el sector publicitario se estudia bajo un modelo de difusión epidemiológica que nos confirma que esta segunda opción es más efectiva que la primera.

En la publicación de Negahban [8] de 2013, se propone un marco de simulación de dos distintos niveles para el problema de gestión de inventario. En dicha simulación, se basa en agentes para modelar el efecto de la intensidad publicitaria y el efecto del boca a boca sobre la demanda para estimar la distribución de la demanda bajo niveles de intensidad publicitaria. Los resultados del modelo basado en agentes

---

<sup>1</sup><https://gsuite.google.com/intl/es/products/calendar>

se conectan a un modelo de simulación de Monte Carlo para tomar la decisión final sobre la intensidad de publicidad óptima y la cantidad de orden económica con el objetivo de maximizar el beneficio esperado.

En los tiempos actuales, el uso de las RRSS (Redes Sociales) ya está estandarizado entre toda la población y cada vez se usan de forma más frecuente para buscar información de interés o noticias de la actualidad. Para incluir en nuestro proyecto las campañas en el ámbito de las RRSS, Du et al [2] desarrolla un modelo para predecir la asistencia a cierta actividad mediante un algoritmo de descomposición de valor singular con vecindad de factores múltiples (SVD-MFN), explotando así los comportamientos individuales en EBSN (Event-Based Social Networks).

Sapp et al. [9] proponen su “Digital Advertisinf System Simulation” como una posible solución de modelado de la publicidad y su impacto en el comportamiento del público objetivo. Trata de una propuesta genérica que ofrece funcionalidades interesantes dentro del marketing digital, como la optimización de la campaña publicitaria.

En el artículo de Sherstennikov [10] presenta un modelo matemático universal para una campaña publicada para empresas que disponen de un único producto. El objetivo del artículo es el cálculo del presupuesto óptimo para la campaña en base a una predicción de la demanda. Ésta herramienta le será de gran utilidad al sistema optimizador para realizar sus propuestas propias.

## **Hoja de firmas**

Esta etapa final es una tarea que se realiza una vez ya realizada la visita al Punto de Colecta y haber completado las anteriores etapas. Su principal función es la de realizar un registro de los datos más importantes de la visita como puede ser el número de donantes que se han presentado o el tiempo que ha llevado en realizar toda la operación. El contenido íntegro de este proceso será explicado en el Capítulo 5 en la Sección 5.2.4.

Esta parte del trabajo viene de una petición personal del propio Centro de Hemodonación de la Región de Murcia para facilitar el trabajo del administrador, ya que, actualmente, esta labor se realiza de forma no informatizada por lo que su rendimiento siempre va a ser mejorable en cualquiera de los aspectos. Al ser una petición para incluir en nuestro proyecto no se ha procedido a la investigación de herramientas que puedan servir para realizar dicha tarea.

## **Procesos restantes**

Además de los principales 4 procesos que hemos mencionado en este mismo capítulo, no hemos indagado en los demás debido a que no es objetivo de nuestro estudio. Al haber realizado varias divisiones del trabajo general en procesos más simples, se puede realizar un estudio de cada uno de estas etapas para optimizar

cada una de ellas y así mejorar la eficiencia total del trabajo que en etapas futuras tengan un reflejo en nuestra interfaz gráfica.

## CAPÍTULO 3

# Metodología

En este capítulo vamos a explicar la metodología empleada para conseguir los objetivos propuestos en el Capítulo 1. Como ya hemos mencionado anteriormente, este proyecto nace de una necesidad real del centro de hemodonación de la Región de Murcia donde se quiere optimizar la colecta de bolsas de sangre de los donantes.

Hoy en día, la forma de trabajar que tiene el administrador del centro consiste en pasar horas delante de un calendario, realizando llamadas, diseñando carteles, recogiendo hojas de firmas de cada día, elaborando tablas de los turnos que tiene que tener cada empleado... Todo este trabajo realizado sin ayuda de un programa específico puede afectar a su eficiencia, debido a que se debe contar con diversos parámetros al realizar la planificación. Para mejorar la recolección debemos usar un administrador capaz de informarnos de cuáles son las rutas que debemos seguir diariamente y con qué número de empleados. En la actualidad, como hemos contado en Capítulo 2, no existe un administrador de centros de hemodonación para realizar planificaciones más eficientes que puedan ahorrar tiempo al administrador, solo se usan gestores de bancos de sangre, por lo que nuestro equipo de proyecto utilizará un sistema optimizador ya desarrollado que se encargará de dicha administración.

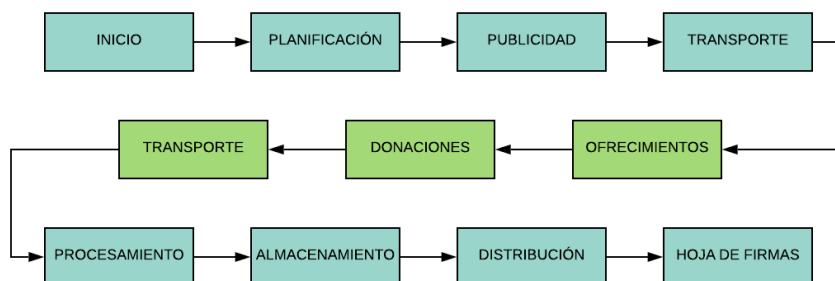


Figura 3.1: Etapas del desarrollo del proceso de recolección de unidades de sangre.

El proceso de recolección de unidades de sangre actualmente se realiza en 4 etapas diferentes. En nuestra propuesta, queremos eliminar esas 4 etapas y en su lugar realizar 11 procesos más simples y fácilmente diferenciables como podemos ver

en la Figura 3.1.

#### **Proceso 1: Inicio**

La primera etapa que hemos definido en nuestra división de la carga de trabajo, consiste en la selección de la fecha en particular que nosotros queremos visitar un punto de colecta. Este proceso será el núcleo debido a que la información de cada proceso beberán de este.

#### **Proceso 2: Planificación**

La siguiente etapa del trabajo realizado en el proceso de recolección de bolsas de sangre corresponde a la organización de visita de cada Punto de Colecta, con número de trabajadores, lugar de desplazamiento, bolsas esperadas. . . Gracias a esta información, se pueden conocer el material y los recursos que vamos a necesitar en el día.

#### **Proceso 3: Publicidad**

Etapa relacionada con los medios para notificar a las personas que frecuentan el lugar nuestra intención de ir a realizar la campaña de donación de sangre. Los medios que se emplean en la actualidad para publicitar son el empleo de carteles informativos (en formato PDF) o mediante mensajes de teléfono mediante la aplicación de *Telegram*.

#### **Proceso 4: Transporte**

Una vez publicitado el punto de colecta con la intención de visitar el lugar, la siguiente etapa del trabajo consistirá en el desplazamiento de los equipos de trabajo al sitio correspondiente.

#### **Proceso 5: Ofrecimientos**

Esta etapa contiene las acciones que debe realizar el donante cuando hace acto de presencia en el punto de colecta y entrega los documentos personales legales necesarios para su posterior revisión por el administrativo a cargo.

#### **Proceso 6: Donaciones**

Etapa que abarca desde que el donante se presenta en el punto de colecta hasta que la sangre queda embolsada.

**Proceso 7: Transporte**

Etapa donde se indica el desplazamiento de retorno del equipo de trabajo al centro de hemodonación.

**Proceso 8: Procesamiento**

Etapa que tiene en cuenta el control de las condiciones sanitarias en las que se encuentra el material y el fraccionamiento en sí de la sangre en subproductos.

**Proceso 9: Almacenamiento**

Etapa que mide el nivel de aprovisionamiento de los almacenes calculadas mediante la capacidad y la ocupación.

**Proceso 10: Distribución**

Etapa de transporte de las bolsas de sangre desde el almacén hasta el receptor.

**Proceso 11: Hoja de firmas**

Etapa final donde se toman notas de cómo se ha llevado el proceso de extracción una vez visitado el lugar.

Este planteamiento en 11 pasos es una propuesta innovadora del equipo de investigación donde cada etapa se verá reflejada en la interfaz. El administrador que interactuará con la GUI deberá navegar por la ella seleccionando la etapa en concreto que desea ver para simplemente visualizar cierta información o modificarla. Para llevar a cabo dicho proyecto daremos uso de un administrador backend que denominaremos “Sistema Optimizador” desarrollado por un compañero del equipo de investigación, Nacho Uranga Herrandorena, cuyo sistema de optimización será el encargado, entre otras cosas, de proponer los calendarios correspondientes de los destinos, personal, vehículos y material necesario que debemos llevar con nosotros a las colectas.

Nuestra GUI tendrá que tener ciertas funcionalidades que debemos incorporar a para poder realizar las tareas oportunas. Deberá tener las funciones de:

- Seleccionar cualquier fecha. Para ello se usaría un calendario interactivo, con el cual nosotros podamos navegar por él para seleccionar la fecha de los datos que queramos visualizar o modificar.
- Enviar propuestas al sistema optimizador para que las tenga en cuenta cuando nos dé una nueva solución.
- Modificar información almacenada en la BBDD.

- Facilitar el uso de los medios publicitarios, ya sea cargando documentos de publicidad o enviando mensajes a los donantes.
- Digitalizar la hoja de firmas para poder realizar futuras optimizaciones.

En adición a esta funcionalidad, tenemos que tener claros unos requisitos que debe tener nuestra GUI:

- Realización de una interfaz gráfica amigable para la visualización de datos relativos al sistema de predicción y optimización.
- Recogida y manejo de restricciones deseadas por los responsables del Centro de Hemodonación y adecuación para optimización.
- Desarrollo de un sistema de comunicación sencillo y escalable Cliente-Servidor.
- Desarrollo del backend de almacenamiento de datos.
- Sistema independiente del sistema operativo de la máquina cliente.
- Test y pruebas de concepto en entorno real.

En las siguientes secciones de este capítulo, el lector podrá encontrarse con una descripción de la arquitectura de nuestro sistema para mayor comprensión y visualización, seguido del desarrollo de los objetivos a alcanzar.

### 3.1. Arquitectura

En esta sección, explicaremos de forma más clara cómo debe ser nuestro sistema que vamos a desarrollar y las relaciones que deben existir entre los diferentes elementos del mismo.

Como podemos observar en la Figura 3.2, la arquitectura de nuestro sistema está compuesta por diferentes equipos.

- **Interfaz Gráfica de Usuario (GUI):** Herramienta de trabajo central del sistema. Es la encargada de realizar la comunicación entre los demás equipos y de mostrar o editar información relacionada de la BBDD o el backend de optimización.
- **Empleado:** Persona del centro de hemodonación que manejará la GUI para administrar las salidas y rutas que deben hacer los equipos de recolección de bolsas de sangre en las campañas de donación.
- **Base de datos (BBDD):** Almacenamiento en la nube de los datos que son de especial interés para el cálculo de las propuestas ofrecidas por el sistema de optimización.



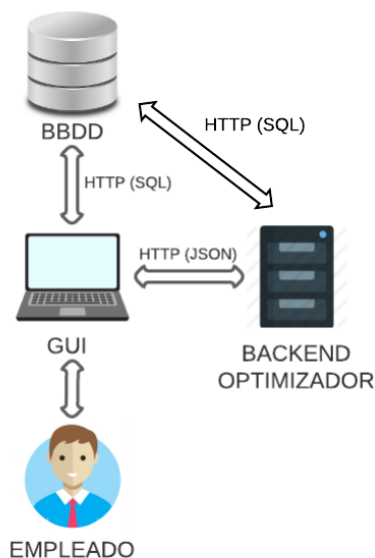


Figura 3.2: Esquema de la arquitectura del sistema.

- **Backend de optimización:** Herramienta diseñada por el equipo de investigación quien ofrecerá las soluciones posibles de planificación para una mayor eficiencia a la hora de realizar las recolecciones de muestras.

Al fijarnos en las relaciones entre los sistemas en la Figura 3.2, nuestro empleado será el encargado de controlar la GUI para realizar las consultas que él necesite. Deberá existir una comunicación con la BBDD mediante peticiones SQL (Structured Query Language) sobre HTTP, por la cual podremos mostrar en nuestra GUI la información relevante que haya almacenada y se desee consultar. También debemos ser capaces de editar esa misma información, así como eliminarla o añadir nueva como: puntos de extracción, personal o vehículos disponibles para la recolección de bolsas de sangre.

Además de poder disponer de esa comunicación de con la BBDD, también debemos ser capaces de comunicarnos con nuestro sistema optimizador. En esta ocasión, los mensajes de comunicación entre nuestra GUI y el sistema optimizador deberán ser mensajes JSON (JavaScript Object Notation)—el mismo que emplea el sistema optimizador con la BBDD para realizar sus correspondientes cálculos—del tipo HTTP también. Esta arquitectura de comunicación está pensada para que toda la lógica pueda ser trasladada a una interfaz web si en un futuro se requiere, lo que hace que sea más sencilla su implementación.

Una vez definidas las diferentes partes de las que está compuesto nuestro sistema, debemos desarrollar nuestra GUI desde el proceso de instalación, hasta su uso del día a día. En primer lugar tendríamos que lanzar un archivo ejecutable

para la instalación que consistiría en la preparación de todos los elementos para su posterior ejecución. Este fichero ejecutable, instalaría la BBDD y la GUI, y, en un futuro, el sistema optimizador. Actualmente este sistema se está realizando en Java por lo que aún no están conectados, pero cuando esté disponible se podrá integrar en nuestro instalador. A continuación, mostraremos los pasos que debemos seguir para realizar nuestro proyecto:

- Preparación de los datos.
- Construcción y llenado de la BBDD.
- Implementación de la GUI.
- Comunicación entre los sistemas.

### 3.2. Preparación de los datos

En primer lugar, en lo relacionado a la instalación del programa, para realizar el perfecto llenado de la BBDD necesitaremos disponer de unos archivos adicionales con los datos que nosotros queremos almacenar, tratar y operar con ellos. Es importante remarcar, que estos archivos deben tener unas cabeceras concretas de datos para poder tratarlos y almacenarlos de forma correcta. A continuación contaremos qué tipos de ficheros se deben de preparar con las cabeceras en el orden indicado.

#### Fichero 1: Localizaciones de Puntos de Colecta

En este fichero podremos encontrar información relativa a los puntos de colecta que visitamos cuando realizamos las campañas de donación de sangre. El contenido de este fichero de ser:

1. **Código identificación (formato text):** Número de identificación del punto de colecta. Se almacena en formato de texto debido a que un punto de colecta puede tener el caracter “A o B” para indicar si se visita el lugar por la mañana o por la tarde respectivamente.
2. **Colecta (formato text):** Nombre de referencia del punto de colecta.
3. **Ciudad (formato text):** Nombre de la ciudad a la que pertenece el punto de colecta.
4. **Modulo (formato text):** Indicador de la zona a la que pertenece el punto de colecta. En nuestro caso real de la región de Murcia se dispone de 4 módulos: A, B, C y D.
5. **Turno (formato text):** Indicador de si se debe visitar un sitio por la mañana, por la tarde o en cualquiera de los dos momentos.

6. **Tipo (formato text):** Texto que informa de qué tipo es la zona del punto de colecta.
7. **Frecuencia de visita (formato text):** Texto que nos informará de cuántas veces podemos visitar el punto de colecta al año o en qué momentos podremos ir a realizar las campañas. Las opciones que se pueden dar para especificar este campo vendrán referenciadas en el Anexo: 1.
8. **Día específico (formato text):** Texto que nos informará de si hay que visitar el sitio en cuestión un día específico de la semana.

### Fichero 2: Histórico de los Puntos de Colecta

En este fichero podremos encontrar información relativa al histórico de datos estadísticos de interés en los puntos de colecta para su optimización. El contenido de este fichero de ser:

1. **Comarca (formato text):** Texto explicativo de la región en la que se encuentra la recolección realizada.
2. **Código identificación (formato text):** Número de identificación del punto de colecta. El mismo campo utilizado en el *Fichero 1*.
3. **Colecta (formato text):** Nombre de referencia del punto de colecta.
4. **Fecha (formato date [mm/DD/yyyy]):** Fecha de realización de la campaña de donación.
5. **Donantes previstos (formato int):** Número de donantes esperados al iniciar la recolección.
6. **Donantes presentados (formato int):** Número de donantes presentados en la recolección.
7. **Donaciones aceptadas (formato int):** Número de donaciones de donantes no rechazados en la selección previa.
8. **Diferencia (formato int):** Resta entre el número de donantes previstos y las donaciones aceptadas.
9. **Número de habitantes (formato int):** Número de habitantes de la zona.

### Fichero 3: Empleados

En este fichero podremos encontrar información relativa a los empleados que realizarán las campañas de donación con su cargo e información correspondiente. El contenido de este fichero de ser:

1. **Apellido 1 (formato text):** Primer apellido del empleado.

2. **Apellido 2 (formato text):** Segundo apellido del empleado.
3. **Nombre (formato text):** Nombre del empleado.
4. **Entidad (formato text):** Nombre de la entidad a la que pertenece el empleado.
5. **Teléfono de contacto (formato int):** Número de teléfono laboral del empleado.
6. **e-Mail (formato text):** E-Mail de contacto del empleado.
7. **Puesto (formato text):** Text explicativo del puesto de trabajo del empleado. En el Anexo 1 se especificará los diferentes puestos de trabajo.
8. **Contrato (formato text):** Tipo de contrato del empleado. Puede ser “Permanente” o “Externo”.
9. **Turno (formato text):** Turno de trabajo del empleado.
10. **Fecha de alta (formato date [mm/DD/yyyy]):** Fecha de alta laboral del empleado.
11. **Fecha de baja (formato date [mm/DD/yyyy]):** Fecha de baja laboral del empleado.
12. **Estado (formato text):** Texto indicador de si el empleado está trabajando en activo o se encuentra de baja. Debemos indicar en este apartado si el empleado está *Disponible* o *No disponible*.
13. **Observaciones (formato text):** Texto adicional para dejar las anotaciones propias correspondientes sobre el empleado.

#### Fichero 4: Vehículos

En este fichero podremos encontrar información relativa a los vehículos que disponemos para los desplazamientos. El contenido de este fichero de ser:

1. **Nombre (formato text):** Nombre del referencia del vehículo.
2. **Entidad (formato text):** Nombre de la entidad a la que pertenece el vehículo.
3. **Pasajeros (formato int):** Número de pasajeros capaces de transportar el vehículo.
4. **Capacidad máxima (formato int):** Capacidad máxima de unidades de sangre que puede transportar el vehículo.
5. **Capacidad máxima ampliada (formato int):** Capacidad máxima adicional de unidades de sangre que puede transportar el vehículo mediante el uso de un elemento de aumento de carga.

6. **Estado (formato text):** Texto indicador de si el vehículo se encuentra disponible o sufre alguna avería. Debemos indicar en este apartado si el vehículo está *Disponible* o *No disponible*.
7. **Observaciones (formato text):** Texto adicional para dejar las anotaciones propias correspondientes sobre el vehículo.

#### Fichero 5: Ciudades

En este fichero podremos encontrar información relativa a las ciudades que se encuentran en la zona de operación. El contenido de este fichero de ser:

1. **Nombre (formato text):** Nombre de la ciudad.
2. **Identificador chat (formato int):** Identificador del chat correspondiente para el envío de notificaciones a los donantes.

#### Fichero 6: Entidades

En este fichero podremos encontrar información relativa a las entidades que se encuentran en la zona de operación. El contenido de este fichero de ser:

1. **Nombre (formato text):** Nombre de la entidad correspondiente.
2. **Dirección (formato text):** Dirección de la localización de la entidad.
3. **Administrador (formato text):** Nombre del administrador de la entidad.
4. **Año inicio (formato int):** Año que empezó su labor la entidad.
5. **Api Token (formato text):** Identificador token para el uso de notificaciones a los donantes por mensajería.

La explicación de los valores que deben tomar algunos campos de los ficheros vendrá explicada en el Apéndice A. Estos archivos que usaremos para importar los datos y subirlos a nuestra BBDD pueden tener diferentes extensiones, aunque nosotros nos centraremos en los formatos más clásicos para la agrupación de datos del mismo tipo como son las extensiones CSV (Comma-Separated Values) o XLSX (Excel Microsoft Office Open XML Format Spreadsheet file).

### 3.3. Construcción y llenado de la BBDD

Una vez teniendo disponibles los archivos que se necesitan para el llenado de la BBDD, procederíamos a la creación de las tablas de datos correspondientes. Para la construcción de estas tablas, usaremos unos ficheros ya definidos, del tipo SQL,

donde estarán definidos los parámetros correspondientes de cada tabla, además de las relaciones que puedan existir entre unas tablas y otras. Las tablas que se crearán con los ficheros SQL serán las siguientes:

- *cities*: Tabla con información de las ciudades que se encuentran en nuestra zona de operación y de su identificador para los chats correspondientes para el envío de publicidad mediante Telegram.
- *entities*: Tabla con información de las entidades que se encuentran en nuestra zona de operación. En esta tabla podemos encontrar información de la entidad como el año que empezó a operar, su dirección o el Api Token adecuado para complementar con el identificador de chat usado en la tabla *cities* para los mensajes vía Telegram.
- *locations*: Tabla con información de las localizaciones de los lugares que se visitarán para la recolección de las donaciones. Los principales campos que podemos encontrar en la tabla son el identificador del Punto de Colecta, el módulo y la ciudad al que pertenece y el turno de cuándo se puede ir a visitar el sitio en cuestión.
- *personnel*: Tabla con información de los empleados encargados de realizar dichas tareas. En ella podemos encontrar información como el teléfono de contacto del empleado, su dirección de correo electrónico o la labor que ejerce en las campañas de donación.
- *vehicles*: Tabla con información de los transportes de personal, material médico y demás. En la tabla podremos encontrar información relativa a los vehículos como por ejemplo si el transporte sufre una avería o de cuál es su capacidad para el transporte de las unidades de sangre.
- *statistics*: Tabla con información del histórico de los lugares donde se han realizado recolectas anteriormente. Principalmente los datos a tratar son los extraídos del fichero de *Histórico de los Puntos de Extracción*, donde usaremos los datos del fichero para calcular diferentes cálculos estadísticos que son: la media aritmética, el primer, segundo y tercer cuartil de datos, desviación estándar, diferencia entre los datos esperados y los reales y las visitas que se han realizado al Punto de Colecta.
- *data\_type\_info*: Tabla relacionada con la tabla de datos *statistics* donde se especifica qué tipo de dato es el almacenado de los listados.
- *days\_info*: Tabla con información de los días de la semana. Esta tabla se proporciona para una posible futura implementación de un idioma adicional.
- *types\_info*: Tabla relacionada con la tabla de datos *locations* donde se especifica el tipo del punto de extracción.
- *jobs\_info*: Tabla relacionada con la tabla de datos *personnel* donde se especifica los diferentes oficios de los empleados.

- `signature_sheet`: Tabla con información sobre las notas realizadas en la hoja de firmas como por ejemplo los empleados que han visitado el Punto de Colecta, el número de donantes presentados, donantes atendido u horas de trabajo dedicadas entre otros parámetros.
- `contraindications`: Tabla con información de las contraindicaciones existentes para situar en la hoja de firmas. En la tabla podremos encontrar el código de la contraindicación que deseamos marcar, el tipo, su nombre explicativo y su duración.
- `rejections`: Tabla con información de los rechazos existentes para situar en la hoja de firmas. En la tabla podemos encontrar datos del tipo el identificador del trabajador, el lugar donde se ha producido o el tipo de contraindicación producida.
- `jobs_functions.info`: Tabla con información de los trabajos desempeñados por los distintos empleados del centro, para poder indicarlo en la hoja de firmas y que sirva para el optimizador.

La información completa del contenido de las tablas de datos y de las relaciones entre ellas será explicada en el Apéndice B.

Si el centro de hemodonación no dispone de un acceso a una BBDD propia por los motivos que fueren, el mismo centro podría contactar con nosotros para proporcionarle el acceso a una BBDD remota en MySQL, ya que el uso de una nube de información es clave para el completo funcionamiento del sistema optimizador.

Cuando tenemos seguro dónde vamos a tener nuestra BBDD, procederemos a la creación de las tablas con los ficheros SQL. Estos ficheros además de contener las instrucciones de creación y los campos de los que disponen cada una de ellas, tendrán relaciones entre unas tablas y otras. Una vez creadas, el instalador leerá los 6 ficheros de datos que hemos tenido que preparar para su tratamiento y subida subida a la BBDD.

### 3.4. Implementación de la GUI

La siguiente parte de nuestro programa a desarrollar deberá ser una GUI cuya función primordial trabajar de intermediador entre el administrador del centro de hemodonación, la BBDD y el sistema optimizador. En la interfaz, podremos visualizar y editar cualquier flujo de datos que se le dará en cada uno de los 11 procesos.

La GUI deberá ser intuitiva, amigable y no muy sobrecargada de opciones para que cualquier usuario que la utilice no le genere rechazo en su interacción con ella. Nuestro objetivo es facilitar el trabajo del administrador ya que actualmente todas las operaciones que debe realizar se hacen a mano en papel, y le resultará de gran ayuda esta herramienta.

Como nuestra GUI debe mostrar diferentes tipos de datos—ya sea respecto a la BBDD o al sistema optimizador—deberemos dividir nuestra interfaz en varias pestañas donde cada una de ellas se especialice en cada uno de los 11 diferentes procesos optimizables que hemos dividido la cadena de trabajo.

1. Inicio: Seleccionar la fecha que más nos interese para saber su planificación.
2. Planificación: Modificar los parámetros de la organización de las rutas a realizar.
3. Publicidad: Elegir de entre los medios existentes la forma de publicitar nuestra intención de visita.
4. Transporte: Aumentar la eficiencia a la hora de desplazarnos al punto de colecta.
5. Ofrecimientos: Seleccionar mejor a los donantes que realizan las acciones correspondientes cuando hacen acto de presencia en el punto de colecta.
6. Donaciones: Reducir el tiempo que transcurre desde que el donante se presenta en el punto de colecta hasta que la sangre queda embolsada.
7. Transporte: Aumentar la eficiencia a la hora de desplazarnos de vuelta al centro de hemodonación.
8. Procesamiento: Aumentar el control de las condiciones sanitarias en las que se encuentra el material.
9. Almacenamiento: Optimizar el nivel de aprovisionamiento de los almacenes.
10. Distribución: Mejorar el transporte de bolsas de sangre enviadas a los receptores.
11. Hoja de firmas: Controlar de forma eficiente las anotaciones de los lugares visitados.

Nuestro proyecto se centrará principalmente en el proceso de *Planificación*. Aunque en adición a ese proceso, también daremos uso a las etapas de *Inicio*, *Publicidad* y *Hoja de firmas* para darle más funcionalidad a la interfaz. Para futuros, nuestro equipo de trabajo se encargará de optimizar cada uno de los procesos restantes.

Como ya hemos explicado, la etapa de *Inicio* es el núcleo del programa, por lo que todas las pestañas que podamos implementar en nuestra GUI correspondientes a los restantes 10 procesos, tendrán un botón de navegación que nos permitirá volver a nuestro proceso de *Inicio* en la interfaz.

En cuanto al proceso que se centra este TFG, la etapa de *Planificación* dispondrá de unas funciones que sirvan para enviarle las peticiones que nosotros consideremos adecuadas al sistema optimizador. Estas funcionalidades se realizarán mediante eventos que se produzcan al accionar botones de la pestaña. Los botones que se encargarían de realizar los eventos serían los siguientes:



- **Forzar Ir:** Al seleccionar este botón nos dará la posibilidad de seleccionar un destino entre los que podemos visitar para indicarle al sistema optimizador que queremos ir a un lugar en particular el día que seleccionemos en el calendario.
- **Editar:** Este botón no daría la posibilidad de editar las propuestas que nos manda el sistema optimizador. El funcionamiento es el siguiente: en primer lugar debemos utilizar botones de selección para indicar qué destino es el que deseamos realizar alguna modificación. Una vez accionado el botón, se abrirá una ventana donde podremos cambiar parámetros de la propuesta enviada por el sistema optimizador.
- **No Ir:** Este botón sirve para indicar al sistema optimizador que alguna de las propuestas que nos manda no nos convence o es impracticable. Por defecto, las propuestas recibidas no están bloqueadas ni validadas, cuyo significado procederemos a explicar cuando desarrollemos el funcionamiento de los dos botones restantes. El funcionamiento de este botón es el siguiente: en primer lugar debemos utilizar los botones de selección—como con el botón *Editar*—para indicar qué destino es el que deseamos realizar la modificación. Una vez accionado el botón, se abrirá la ventana informativa donde responderemos si deseamos no ir al destino seleccionado, lo que provocará que desbloqueemos e invalidemos ese destino.
- **Bloquear:** Este botón se utiliza para indicar al sistema optimizador que la propuestas que nos manda nos gusta. El funcionamiento de este botón es el siguiente: seleccionaremos el destino que queremos bloquear de los propuestos y una vez seleccionado accionaremos el botón que abrirá una ventana informativa donde responderemos si deseamos bloquear al destino seleccionado. Al bloquear el destino y enviarse esa información al backend optimizador, el sistema tendrá en cuenta dicha petición para no modificar nuestra planificación cuando nos envíe nuevas propuestas.
- **Validar:** Este botón sirve para indicar al sistema optimizador que la propuestas que nos manda nos gusta y queremos dejarla fija. El funcionamiento de este botón es el mismo que el botón de *Bloquear* solo que dándonos la posibilidad de validar el destino. Un destino validado para el backend optimizador tiene, a efectos prácticos, el mismo tratamiento que un destino bloqueado. La diferencia es la confirmación que tiene del lugar el administrador de la GUI. En primer lugar un destino se bloquea para indicar que la propuesta nos interesa y una vez que se ha confirmado que se puede ir a dicho lugar en dicha fecha, el usuario podrá validar el destino para conocimiento de él que se realizará la visita indicada. Como hemos explicado anteriormente en el botón *No Ir*, esta validación puede ser eliminada pulsando ese botón por si ocurriera una emergencia de cualquier tipo que nos impediría visitar el sitio.

Todos las modificaciones que realicemos con los botones de *Forzar Ir*, *Editar*, *No Ir*, *Bloquear* y *Validar*, se verán reflejadas al instante en la Interfaz Gráfica para que el usuario pueda apreciar los cambios que ha realizado. Sin embargo, dichas modificaciones no se han enviado al sistema optimizador.

Para enviar nuestras peticiones al sistema, si queremos guardar las últimas modificaciones realizadas, en primer lugar debemos accionar un botón que denominaremos *Guardar* que pondrá en marcha un evento para salvar los cambios efectuados y si por el contrario queremos retractarnos de los cambios que hemos hecho deberemos pulsar un botón denominado *Cancelar* o simplemente volver a nuestra pestaña de Inicio. Una vez guardados los cambios que queremos enviar al sistema optimizador y somos conscientes que queremos enviar dichas peticiones, deberemos pulsar un botón denominado *Optimizar* situado en la pestaña de *Inicio* que generará un archivo del tipo JSON que procederemos a pasarle al optimizador para que tenga en cuenta las modificaciones que nosotros queremos que haga.

En cuanto al funcionamiento de la pestaña *Publicidad*, esta se encarga de avisar a los donantes de sangre del sitio en concreto que vamos a visitar para la recolección de bolsas de sangre. De esta forma podemos asegurar en cierta modo que los donantes habituales estarán pendientes de cuándo deben ir y su índice de rechazo disminuya debido a una concienciación previa antes de realizar las donaciones.

### 3.5. Comunicación entre Sistemas

Una vez desarrollada nuestra GUI, nuestro siguiente paso sería establecer las comunicaciones necesarias entre nuestro programa con la BBDD y el sistema optimizador. Como pudimos relatar de forma resumida en la sección de Sección 3.1, existen dos tipos de comunicaciones en nuestro programa.

#### Comunicación BBDD

La primera consiste en la comunicación con la BBDD donde tendremos almacenados los lugares que visitar para realizar las donaciones, el histórico de las extracciones anteriores para comprobar la afluencia de donantes de cada sitio y en qué fecha, los vehículos registrados en los que nos desplazaremos a los destinos y otros datos que nos serán útiles para el perfecto funcionamiento de nuestro proyecto.

Desde nuestra interfaz, el administrador podrá acceder a parte de esa información que hay almacenada en la nube mediante mensajes HTTP (SQL) para realizar algunas acciones. Los elementos que se le dará permiso para modificar son: los datos de los Puntos de Colecta, el personal en plantilla que formará los equipos de trabajo y los vehículos que se ocuparán del transporte de los equipos. Para esta información que podemos acceder y modificar, podemos realizar las siguientes acciones:

1. Añadir contenido: Nuestra BBDD estará inicializada con unos datos que le hemos pasado durante el proceso de instalación del programa. Somos conscientes que las campañas de donación pueden ampliar los lugares de colecta para visitar zonas nuevas de interés. Además si se contrata un nuevo empleado o se dispone de un vehículo nuevo, deberemos ser capaces de añadir sus datos para que el sistema optimizador cuente con esta nueva información.

2. Editar contenido: También podremos ser capaces, no solo de leer el contenido de las tablas mencionadas, si no de editar la información que hay guardada. De esa forma si se producen variaciones del tipo el turno que se puede visitar un lugar de extracción ha cambiado, o un vehículo deja de estar disponible por una avería, o una persona cambia de número de contacto, entre otros muchos parámetros, tendremos que poder modificar de forma sencilla cada uno de estos datos de cada tabla.
3. Eliminar contenido: La última acción que podremos hacer con la información de la BBDD es la eliminación de información obsoleta o que ya no se dará uso. Si apreciamos que el rendimiento de un Punto de Colecta ha bajado, el sistema optimizador podría optar por visitar otros lugares más “rentables” por lo que a la larga se dejaría de visitar. En lo relacionado a los empleados, también podremos eliminar algún trabajador que ya no trabaje para el centro o desechar un vehículo que ya no se utilice.

### Comunicación Sistema Optimizador

El último tipo de comunicación que nos encontraremos en nuestra GUI sería el paso de mensajes con el sistema optimizador. Este, se encarga de enviarnos mensajes del tipo HTTP (JSON) con las propuestas que ha considerado óptimas para visitar los lugares para la recolección de las bolsas de sangre. Nosotros con nuestra GUI deberemos ser capaces no solo de leer y mostrar dichas propuestas, si no deberemos ser capaces de decirle al sistema optimizador si nos han gustado las soluciones que ha sugerido, o si no nos gustan, o si nos gustan parcialmente con ciertas modificaciones que podamos hacer. De esta forma, podremos enviarle esas peticiones al backend y así tener unas restricciones a la hora de calcular nuevas propuestas. Los elementos que se encargarán de pasar esta información al sistema optimizador serán las mencionadas en Sección 3.4.

En la comunicación del sistema optimizador con la GUI, el sistema nos enviará propuestas que procederemos a guardar en una carpeta creada en el proceso de instalación. Además de esta carpeta se dará uso a otras adicionales que cumplan ciertas funciones que contaremos a continuación.

- Carpeta *receivedJSON*: En esta carpeta se guardarán las propuestas que nos envía el sistema optimizador. El nombre que tendrá este tipo de ficheros será “YYYYxMMx\_scheduler\_rec\_v00.json” donde se indicarán en YYYY y en MM el año y el mes correspondiente a las propuestas que está enviando y en los números finales la versión enviada. Nuestra GUI se encargará de mostrar los datos en pantalla de las propuestas más recientes, por lo que si se dispone de más de una versión, seleccionará la más nueva.
- Carpeta *sentJSON*: Esta carpeta será donde se guarden las peticiones que nosotros hagamos al sistema optimizador. De forma análoga a los ficheros de *receivedJSON*, el nombre que tendrán estos ficheros será “YYYYxMMx\_scheduler\_sent\_v00.json”. Como podemos observar la estructura

es la misma salvo que contiene el subnombre “sent”. El sistema optimizador tendrá en cuenta las peticiones que nosotros tengamos, con el mismo comportamiento que tiene la GUI con las propuestas enviadas, de solo tener en cuenta el archivo más reciente si se encuentran archivos con la misma fecha en su nombre.

- Carpeta *localViewJSON*: Por último, en esta carpeta se guardarán los cambios que queremos ver reflejados al instante en nuestra interfaz pero no hemos procedido a enviarle al sistema optimizador. El nombre de estos ficheros será de la forma “YYYYxMMx\_scheduler\_loc\_v00.json”, donde sigue la misma estructura en el nombre que en los ficheros recibidos y enviados al sistema optimizador, donde nuestra GUI se encargará de mostrar por pantalla los datos que tengan la versión más reciente.

En cuanto a la estructura que tendrán los ficheros JSON contarán con dos campos a resaltar: *generalInfo* y *scheduling*.

- **generalInfo**

Información básica del mensaje. La información que proporciona es: nombre de referencia, número de versión, número de compilación, año y mes correspondiente a la propuesta que se ha enviado.

- **scheduling**

Array de todas las propuestas proporcionadas por el sistema optimizador. Los campos que podemos ver en este array son:

1. date: Fecha de cuándo se debe realizar la campaña de donación.
2. id: Identificador del Punto de Colecta a visitar.
3. name: Nombre de referencia del Punto de Colecta.
4. city: Nombre de la ciudad a la que pertenece el Punto de Colecta.
5. business: Indicador para definir si es una fecha laboral o no laboral.
6. shift: Turno de visita al Punto de Colecta.
7. vehicle: Nombre del vehículo que se debe usar.
8. forecastTBUmean: Media aritmética de la previsión del número de bolsas de sangre se van a recoger.
9. forecastTBUstd: Desviación típica de la previsión del número de bolsas de sangre se van a recoger.

10. forecastDonnors: Previsión del número de donantes que se van a presentar en el Punto de Colecta.
11. drivers: Número de conductores necesarios para el trabajo.
12. admin: Número de administradores necesarios para el trabajo.
13. nurses: Número de enfermeros o enfermeras necesarios para el trabajo.
14. doctors: Número de doctores necesarios para el trabajo.
15. blocked: Dato boolean para indicar si el destino está bloqueado.
16. blocked\_modified: Dato boolean para indicar si el destino está bloqueado.
17. validated: Dato boolean para indicar si el destino está bloqueado.
18. validated\_modified: Dato boolean para indicar si el destino está bloqueado.

El comportamiento entre un dato blocked o blocked\_modified, al igual que con validated o validated\_modified, es similar entre ambos campos, con la diferencia de que el subnombre “modified” sirve para referenciar que nos gusta solo una parte de la propuesta enviada por el sistema optimizador pero que queremos realizar algunos cambios.



## CAPÍTULO 4

# Implementación

En este capítulo del proyecto, vamos a explicar al lector las tecnologías que vamos a emplear para desarrollar los dos programas que necesitamos, nuestro programa Instalador y la GUI.

La Figura 4.1 corresponde al mismo esquema que se presentó en el Capítulo 3. En esta ocasión, lo que se ha querido representar en los diferentes elementos del diagrama— BBDD y GUI—son las tecnologías que se han utilizado.

### 4.1. BBDD

Para el desarrollo e implementación de la base de datos se utilizará MariaDB; una base de datos relacional.

MariaDB (Figura 4.2) es un sistema de gestión de BBDD derivado de MySQL con licencia GPL (General Public License). Este gestor, fundado por Michael (Monty) Widenius, es considerado el más popular del mundo gracias a

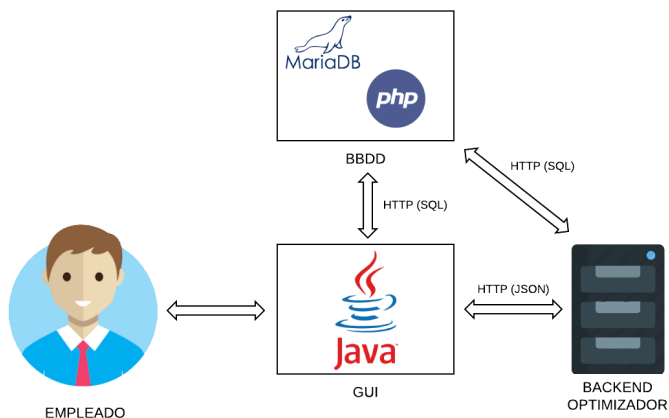


Figura 4.1: Arquitectura de la implementación del sistema.



Figura 4.2: Logo del sistema gestor MariaDB.

la particularidad de ser un software libre de código abierto. La elección de esta tecnología se ve debido a las características que presenta:

- Velocidad: Es una tecnología muy escalable, capaz de manejar decenas de miles de tablas y miles de millones de filas de datos.
- Seguridad: Cuando se producen problemas críticos se distribuye una nueva versión de la misma.
- Funciones integradas: Capaz de manipular y formatear texto, cálculos empresariales y estadísticos o registrar información cronológica entre muchas funciones.

La versión que nosotros usaremos para desarrollar nuestro proyecto en MariaDB será la versión 10.0. Para realizar la gestión nuestra BBDD hemos utilizado la herramienta *phpMyAdmin*:



Figura 4.3: Logo de la herramienta phpMyAdmin.

*phpMyAdmin* (Figura 4.3) es una herramienta escrita en PHP para manejar la administración de MySQL a través de páginas web, utilizando un navegador web. La versión de PHP que usaremos será la versión 7.2.14 y la versión 4.8.4 para phpMyAdmin. Las principales características de esta herramienta son:

- Creación y eliminación de bases de datos.
- Creación, modificación y eliminación de tablas de datos.
- Adición, edición y eliminación de campos.
- Ejecución de sentencias SQL.
- Administración de claves en campos y privilegios.
- Manejo de datos en varios formatos.

En las tablas que hemos estado desarrollando, varias de ellas están preparadas para una posibilidad futura de incluir otros idiomas que no sean el castellano.



## 4.2. GUI

Una vez explicadas las tecnologías que necesitamos usar para la generación y manejo de la base de datos, procederemos a explicar qué herramientas se han usado para realizar nuestro proyecto.



Figura 4.4: Logo del IDE Eclipse.

Eclipse (Figura 4.4) es un Entorno de Desarrollo Integrado (IDE, Integrated Development Environment), una plataforma software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma, capaz de utilizar diferentes lenguajes de programación, usado para desarrollar aplicaciones de cliente enriquecido. En nuestro caso, utilizaremos la versión Eclipse IDE for Enterprise Java Developers en su versión 4.10.0 (2018-12). Ya que es multilenguaje, en nuestro caso utilizaremos el lenguaje Java (Figura 4.5) para el desarrollo de nuestra GUI.



Figura 4.5: Logo del lenguaje Java.

Eclipse contiene una gran variedad de herramientas específicas denominadas clases, procedentes de librerías estándar de Java. Además de estas clases internas que contiene Eclipse, en el entorno también existe la posibilidad de importar librerías externas donde se especialicen en funciones determinadas. La versión de Java que utilizaremos será Java 8 Update 211 (64-bits) y Java SE Development Kit 8 Update 191 (64-bits).

Para el desarrollo de la GUI, además de utilizar el IDE Eclipse y el lenguaje Java, para que nos sea más sencillo e intuitivo el desarrollo de la misma, usaremos una herramienta característica de Eclipse denominada WindowsBuilder Editor.

Como podemos ver en la imagen Figura 4.6, esta herramienta nos ayuda de forma gráfica a desarrollar el diseño de nuestra GUI, dándonos la posibilidad de cambiar la vista entre la herramienta gráfica y la herramienta clásica, donde alberga el código del programa, para poder especificar de forma más precisa parámetros de la misma.

En cuanto a las librerías externas a nuestro IDE pero que debemos usar para realizar todas las funciones que necesitamos, nos vemos en la obligación de importar

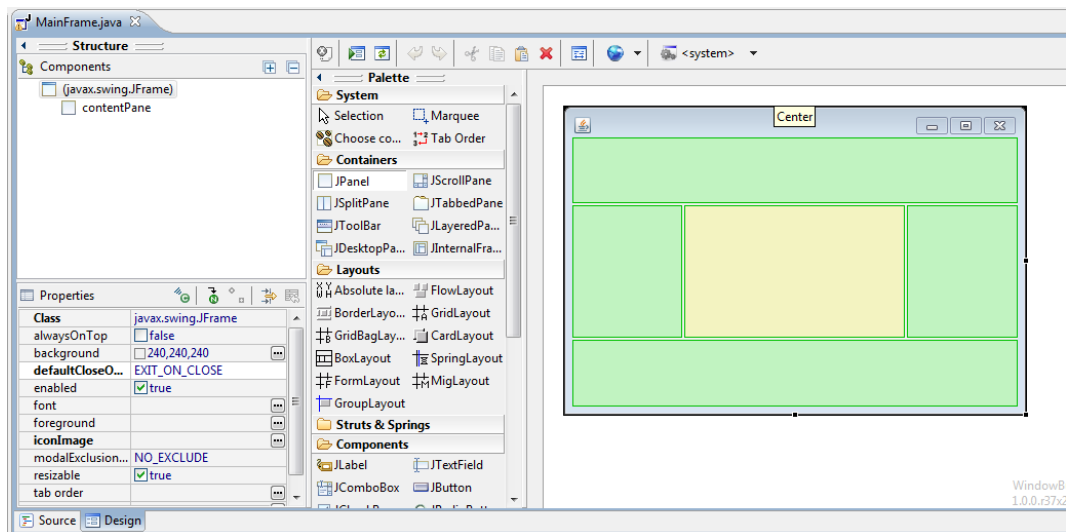


Figura 4.6: Captura de pantalla de la herramienta de desarrollo para GUI en Eclipse.

las siguientes:

- opencsv-4.6.jar: Librería que nos permitirá leer los archivos de extensión CSV.
- poi-4.1.0.jar, poi-examples-4.1.0.jar, poi-excelant-4.1.0.jar, poi-ooxml-4.1.0.jar, poi-ooxml-schemas-4.1.0.jar, poi-scratchpad-4.1.0.jar: Librerías que nos permitirán leer los archivos de extensión XLS y XLSX.
- mysql-connector-java-5.1.47-bin.jar: Librería que nos permitirá realizar conexiones y peticiones a nuestra BBDD.
- gson-2.8.6.jar: Librería que nos permitirá el envío de mensajes del tipo HTTP (JSON) con el backend de optimización de nuestro sistema.

# CAPÍTULO 5

## Resultados

En este capítulo vamos a mostrar al lector los resultados obtenidos al realizar el proyecto empleando la metodología explicada en el Capítulo 3 con la implementación explicada en el Capítulo 4. Como hemos explicado en la metodología, vamos a necesitar seguir los pasos indicados para la realización del TFG.

### 5.1. Proceso de instalación

Como podemos visualizar en la Figura 5.1, cuando instalamos nuestro programa ejecutará diferentes funciones para dejar preparadas todas las herramientas y datos que necesitará usar nuestra GUI. A continuación, procederemos a explicar cada una de las etapas por las que pasa nuestra instalación desde que lo lanzamos hasta que podemos ya ejecutar la GUI.

#### Carga de Parámetros BBDD

En primer lugar, cuando lanzamos el archivo de instalación, el primer paso que dará será la carga de parámetros necesarios para establecer una comunicación con la BBDD que vamos a usar. Antes de proceder con ello debemos conocer si el centro de hemodonación dispone de un acceso a una base de datos. Si resulta que sí, nuestro programa pedirá las credenciales para acceder a dicha base y almacenar allí los datos, o si por el contrario no disponen libremente de esa ventaja, proporcionaremos acceso al centro de hemodonación a través de una base de datos MySQL.

La información para el acceso a las tablas de datos, si tenemos nosotros que proporcionarla, vendrá dada en un fichero del tipo Properties. Este fichero tendrá en su contenido varios parámetros de interés que iremos explicando a lo largo de la sección cuando debamos realizar una consulta a dicho archivo. En lo relacionado a nuestra BBDD, dentro del fichero podemos encontrar información acerca del nombre de la BBDD, su dirección de acceso, el nombre de usuario y la contraseña que hemos de necesitar para poder acceder a ella. Estos datos no son modificables.

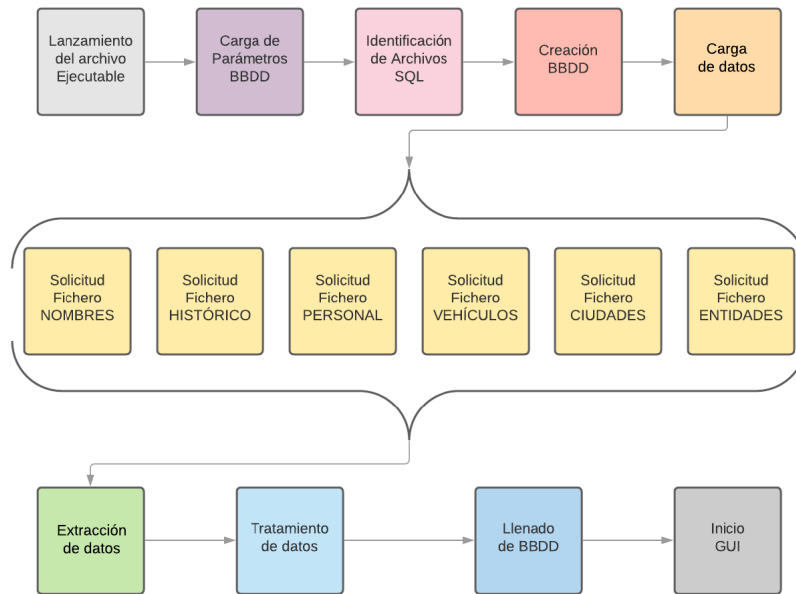


Figura 5.1: Esquema del proceso a seguir en la Instalación de la herramienta.

## Identificación de archivos SQL

Una vez cargados los parámetros necesarios para acceder a la BBDD gracias al fichero *Properties*, el siguiente paso que se realizará en nuestro proceso de instalación será seleccionar los ficheros necesarios donde vendrán los comandos necesarios para la creación de las BBDD. Estos ficheros son del tipo SQL que procederemos a su ejecución en la siguiente etapa de nuestro programa.

## Creación BBDD

Habiendo identificado los ficheros SQL necesarios para la creación de las tablas de la base de datos. En primer lugar, se utilizarán los parámetros de la BBDD que extrajimos de nuestro fichero *Properties* para realizar la conexión con el servidor. Una vez establecida la conexión, se ejecutará el fichero SQL que tendrá las instrucciones necesarias para crear la base de datos—cuyo nombre será proporcionado también por el fichero *Properties*—y que será la que contenga todas las tablas de datos necesarias. Habiéndose realizado con éxito la creación de la base de datos, el siguiente paso de nuestro programa será el llenado de las tablas de datos.

El proceso de instalación recibirá de nuestro fichero *Properties*, los nombres de los ficheros SQL que debe ejecutar. Entre los ficheros que contienen el código necesario para crear las tablas, también se encontrará entre ellos un archivo que defina la relación que debe existir entre las tablas. Este archivo se ejecutará en último lugar cuya información de las relaciones vendrá dada en el Anexo...(aref).

## Carga de datos

El siguiente paso de la instalación, una vez terminada la etapa de creación de todas las tablas de datos en nuestra BBDD, será el llenado de estas tablas. Para ello, nuestro programa nos pedirá que seleccionemos los ficheros que necesita para extraer los datos correspondientes en cada situación.

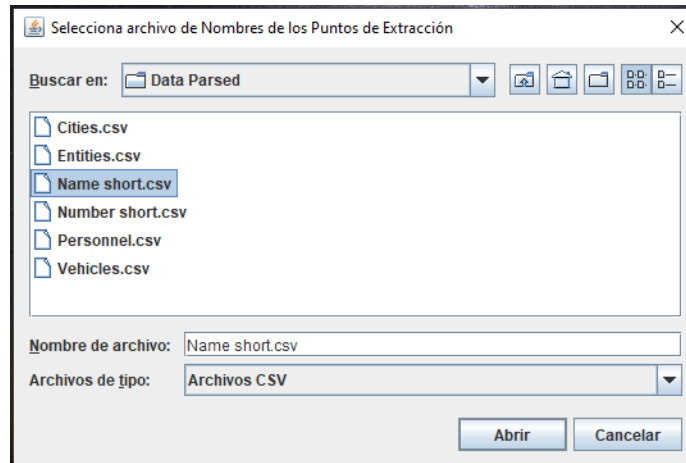


Figura 5.2: Interfaz que nos proporcionará el instalador para la selección de los ficheros a extraer los datos

Como podemos apreciar en la Figura 5.2, durante la instalación se nos abrirán varias ventanas que nos indicarán en el título el archivo necesario para su extracción de datos. Además, podemos ver en la misma figura que el tipo de ficheros que es capaz de extraer los datos son de 3 tipos diferentes, siendo estas extensiones del tipo CSV o XLSX de Microsoft. En la interfaz debemos seleccionar el tipo de archivo que tendrá los datos almacenados donde se podrá visualizar un desplegable con las diferentes extensiones que soporta nuestro programa. Una vez seleccionado el tipo del archivo, deberemos navegar por el buscador de ficheros hasta encontrar el archivo que necesita nuestro programa. El programa nos pedirá que le proporcionemos los 6 archivos diferentes de datos que hemos explicado en el Capítulo 3.

## Extracción y tratamiento de los datos

Después de cargar en nuestro programa la dirección de cada uno de los ficheros de los cuales vamos a extraer los datos, durante la instalación se leerá cada uno de los ficheros y se extraerán los datos como una conjunto de objetos. Una vez extraídos todos los datos de cada uno de los ficheros, estos serán tratados para su posterior subida al servidor que aloja la base de datos.

## Llenado de la BBDD

La última etapa que se realizará durante la instalación antes de poder ejecutar nuestra GUI, será la subida de los datos que hemos extraído y tratado en nuestra

code	name_place	city	region	module	turn	type	freq_times_month	specific_day_week	observations
10000	BARRIO SAN ANDRES	1		A	T	9	not_defined	NULL	NULL
10001	HOSPITAL LA VEGA	1		A	M	5	solicited	NULL	NULL
10002	REAL MURCIA CF SAD (Estadio N Condomina)	1		A	M	1	solicited	NULL	NULL
10003	FRIPOZO	1		B	M	4	solicited	NULL	NULL

Figura 5.3: Captura de pantalla del contenido de la tabla de datos *locations* de la BBDD después de subir el contenido de los ficheros de datos.

id	name_vehicle	entity	passengers	cap_max	cap_max_amp	state	observations
1	Vito		4	100	120	not_avaliabile	NULL
2	Mercedes		5	100	100	not_avaliabile	NULL
3	Renault		4	80	120	not_avaliabile	NULL
4	Seat		5	100	120	not_avaliabile	NULL

Figura 5.4: Captura de pantalla del contenido de la tabla de datos *vehicles* de la BBDD después de subir el contenido de los ficheros de datos.

ya creada BBDD.

Una vez realizado la subida de los datos a la BBDD se avisará al usuario mediante un mensaje en pantalla que se ha completado dicha acción e informando si no se han producido errores durante el proceso.

Un ejemplo cómo quedaría la tabla de datos de *locations* sería lo correspondiente a la Figura 5.3. En la figura podemos apreciar los datos guardados que hubo en los ficheros que le indicamos a nuestro programa para que los leyera y subiera a la nube sobre los puntos de colecta.

Además del ejemplo de la tabla de *locations*, también podemos ver en la Figura 5.4 los datos de la tabla *vehicles* con la información que extrajimos de su fichero correspondiente durante la instalación.

## 5.2. Graphical User Interface - GUI

El objetivo principal de nuestro proyecto consiste en facilitar a los encargados de los Centros de Hemodonación la planificación que ellos deben hacer para satisfacer la demanda de bolsas de sangre, mediante una interfaz gráfica. Al disponer de un sistema optimizador capaz de proporcionarnos propuestas de planificación, debemos desarrollar una interfaz gráfica que nos facilite la visualización de esos datos entre otras muchas funciones que podremos realizar.

Esta GUI debe ser intuitiva, amigable y no sobrecargada de opciones para que cualquier usuario que la utilice no le genere rechazo. La distribución de ella

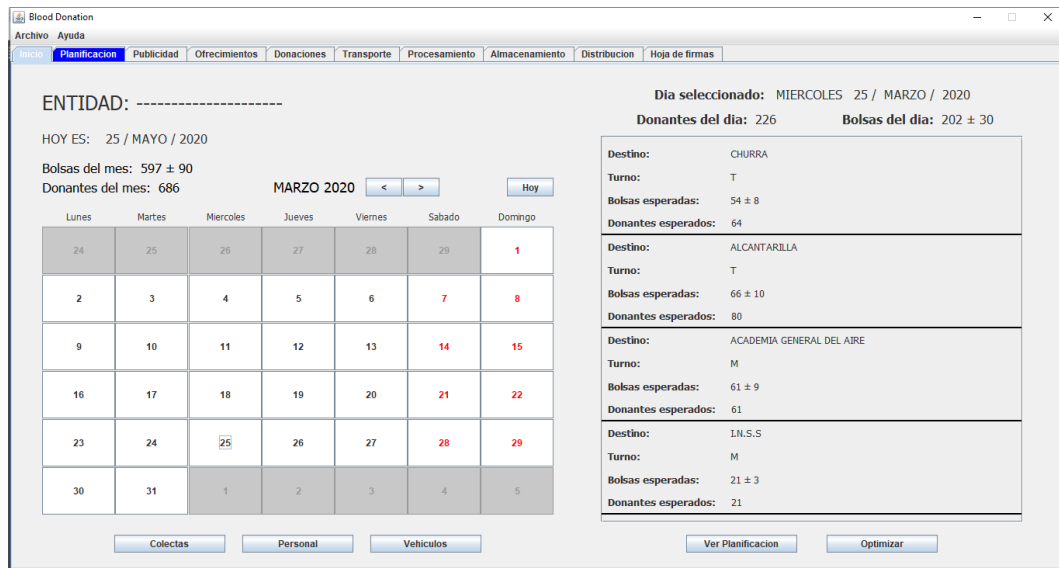


Figura 5.5: Pantalla capturada de la pestaña Inicio de la GUI para el tratamiento de datos almacenados en la BBDD.

estará dividida en pestañas donde cada una mostrará un fase concreta del proceso de recolección de las donaciones. Cada una de las pestañas corresponderán a las ya citadas en el Capítulo 3, correspondiente a los 11 procesos que definimos que vamos a dividir el trabajo de las campañas de donación de sangre. En la información que daremos al lector a continuación mostraremos el resultado gráfico de la interfaz tras su implementación.

### 5.2.1. Inicio

Nuestra pantalla principal será la denominada pantalla de Inicio. Sirve para el control y tratamiento de todos los datos almacenados. Como podemos apreciar en la Figura 5.5, está compuesta por varios elementos. En su parte superior izquierda, destaca el nombre de la entidad que se encarga de administrar la recogida de bolsas de sangre de los donantes. Debajo de la entidad, nos encontramos la fecha actual para tenerla de referencia, pero, el alma de esta pestaña es el calendario. Este calendario será la herramienta necesaria para mostrar los datos que nosotros estamos buscando, para conocer los destinos donde ir para la recogida de las donaciones, equipo necesario, donantes esperados, *etc.*

Su funcionamiento plantea ser amigable e intuitivo—como objetivo del proyecto—para facilitar la labor del trabajador que se encargue de gestionar las salidas a los puntos de extracción. El usuario que maneje la GUI deberá seleccionar con el ratón en el calendario el día que desee conocer la propuesta generada por el sistema optimizador del backend. Al hacerlo, se podrá ver en la zona derecha de la misma pestaña un breve resumen de los destinos que se propone visitar ese día. Además del destino, también se ofrece información adicional de interés al usuario como el turno de visita—si es mañana (M) o tarde (T)—y los donantes y bolsas

```

{
  "generalInfo":
  {
    "barName": "home",
    "version": "v1.0",
    "compilationID": 20200207171933,
    "year": 2020,
    "month": 2
  }
},
"scheduling": [
{
  "date": 20200324,
  "business": 1,
  "loc_id": "3000",
  "name": "CENTRO REGIONAL DE HEMODONACION",
  "shift": "M",
  "type": "Clinic",
  "module": "A",
  "vehicle": "DTN",
  "forecast_TBU_mean": 30,
  "forecast_TBU_std": 5,
  "forecast_donors": 41,
  "drivers": 1,
  "admin": 1,
  "doctors": 1,
  "nurses": 2,
  "blocked": false,
  "blocked_modified": false,
  "validated": false,
  "validated_modified": false
},
{
  "date": 20200324,
  "business": 1,
  "loc_id": "10005".
}

```

Figura 5.6: Captura de pantalla de un contenido de ejemplo de archivo JSON enviado por el sistema optimizador con la propuesta calculada.

esperadas en dicho punto de extracción. En la zona superior donde se muestra el resumen del día, podremos ver la fecha que hemos seleccionado en el calendario además del total de donantes y bolsas de sangre esperadas en total en el día.

Al ejecutar la GUI, el calendario se iniciará con el mes correspondiente a la fecha del día que se está ejecutando dicho programa. Si el empleado desea conocer las propuestas que el sistema optimizador le ha dado para una cierta fecha que no se encuentra en el mes actual, no tiene más que navegar por ese calendario utilizando los botones < y > situados en la zona superior del calendario. Estos botones nos permitirán cambiar el mes seleccionado a uno anterior o a uno posterior de forma respectiva hasta fijar el mes que se está buscando. Si navegando por el calendario nos encontramos en un mes distinto al actual, en ocasiones resulta tedioso y poco práctico tener que viajar por varios meses para volver al mes actual, para evitar dicho desgaste mental existe el botón de “Hoy”. Este botón situado en la parte superior derecha del calendario, nos servirá de atajo para volver de forma rápida al mes que nos encontramos hoy en día. Si por el contrario observamos la zona superior izquierda de nuestro calendario, podremos ver el cálculo total del mes de donantes y bolsas esperadas en el que estamos navegando.

Cada vez que ejecutamos nuestro programa, la GUI busca la propuesta más reciente que ha enviado el backend optimizador. Estas propuestas se realizarán



mediante mensajes HTTP (JSON) donde parte de esos mensajes hemos mostrado en la Figura 5.6. Como podemos ver en la misma imagen, en las propuestas enviadas por el sistema optimizador encontramos campos con la información de la fecha a la que van indicados, el nombre del punto de colecta, su identificación y otros parámetros de interés que ya contamos en el Sección 3.5.

En nuestra pestaña *Inicio* podemos encontrar además unos botones en la parte inferior, tres de ellos en la zona inferior del calendario y dos de ellos situados debajo del resumen del día seleccionado. Los tres primeros botones que tienen el texto *Colectas*, *Personal* y *Vehículos* son los encargados de mostrar la información que hay guardada en nuestra BBDD de los lugares de puntos de extracción, el personal y los vehículos registrados respectivamente.

En la Figura 5.7a podemos observar una lista con los lugares, colectas, que almacenamos inicialmente en la BBDD gracias a los ficheros importados por nuestro Instalador. Si deseamos ver la información completa almacenada de un destino en particular, solo tendremos que seleccionar el botón de *+ Info* y así abrir un mensaje *pop up* que nos dará la información completa del punto de extracción. Además, también podremos editar la información que hay guardada del destino, eliminarla o incluso añadir una nueva localización. En esta última acción, debemos tener cuidado al incluir el nuevo destino porque tenemos que tener en cuenta que los códigos de identificación de los destinos son únicos y si queremos añadir un nuevo destino con un identificador ya en uso, el mismo programa nos avisará de que ya existe una colecta con esa identificación y deberemos usar un código diferente.

En cuanto a las ventanas para visualizar el personal y los vehículos registrados en la BBDD, el funcionamiento es el mismo que en la ventana de colectas salvo que la información guardada está más relacionada a los datos propios. En esta ocasión, entre los datos guardados en personal podremos ver la información que tenemos para contactar con el empleado, además de otros muchos datos, y en los datos de los vehículos podemos ver sus características como la capacidad máxima de bolsas que puede transportar o el número de pasajeros que caben en el vehículo. El contenido de las columnas de todas las tablas de la BBDD vendrá en el Apéndice B para interés del lector. Ambas tablas tienen una particularidad común que es la columna de estado, en ese campo podemos apreciar si el empleado está de baja o se encuentra trabajando activamente, o, en el caso del vehículo, si ese transporte sufre alguna avería por la cual no podemos disponer de él en ese mismo momento, dato importante para el sistema optimizador para saber si puede contar con el vehículo en sus propuestas o no. También podemos añadir nuevos empleados y vehículos a nuestras BBDD al igual que ocurría con las colectas, solo que en esta ocasión no es necesario que asignemos un identificador propio a cada empleado o vehículo ya que la BBDD le asignará uno que será único.

En lo relacionado a los dos botones que nos queda por explicar en la pestaña *Inicio* de la Figura 5.5, *Optimizar* y *Ver Planificación*, el segundo nos redirige a la pestaña de *Planificación* (la cual también es accesible seleccionando su pestaña en la parte superior de la ventana) donde podremos ver de forma más detallada los destinos que ha sugerido el sistema optimizador. El contenido de esta nueva pestaña será explicada a continuación.

Colecta

Localizaciones registradas:

Codigo	Nombre	Comarca	Modulo	Turno	Frecuencia veces al mes	Dia de la semana			
10000	BARRIO SAN ANDRES	1	not_defined	not_defined	not_defined	not_defined	+ Info	Editar	Borrar
10001	HOSPITAL LA VEGA	1	not_defined	not_defined	not_defined	not_defined	+ Info	Editar	Borrar
10002	REAL MURCIA CF SAD (Estado N Condomina)	1	not_defined	not_defined	not_defined	not_defined	+ Info	Editar	Borrar
10003	FRUPOZO	1	not_defined	not_defined	not_defined	not_defined	+ Info	Editar	Borrar

Añadir Volver al Inicio

(a) Captura de pantalla información de Puntos Extracción de la BBDD

Personal

Personal registrado:

ID	Apellido 1	Apellido 2	Nombre	Entity	Categoria	Turno	Estado			
1	Perez	Camona	Juan Antonio		conductor	M	not_available	+ Info	Editar	Borrar
2	Martinez	Lorenzo	Laura		medico	T	not_available	+ Info	Editar	Borrar

Añadir Volver al Inicio

(b) Captura de pantalla información del Personal almacenado

Vehiculos

Vehiculos registrados:

ID	Nombre	Entity	Pasajeros	C. Maxima	C. Maxima Ampliada	Estado			
1	Vito		4	100	120	not_available	+ Info	Editar	Borrar
2	Mercedes		5	100	100	not_available	+ Info	Editar	Borrar
3	Renault		4	80	120	not_available	+ Info	Editar	Borrar
4	Seat		5	100	120	not_available	+ Info	Editar	Borrar

Añadir Volver al Inicio

(c) Captura de pantalla información de los Vehículos almacenados

Figura 5.7: Pantalla capturada en la pestaña Inicio de la Interfaz Gráfica para la visualización de los datos almacenados en la BBDD sobre los Puntos de Colecta, Personal y Vehículos.

Donantes del día: 226      Bolsas del día: 202 ± 30

Destino n°1	Destino n°2	Destino n°3	Destino n°4
Destino 1: CHURRA	Destino 2: ALCANTARILLA	Destino 3: ACADEMIA GENERAL DEL AIRE	Destino 4: I.N.S.S
ID Localización: 10460	ID Localización: 10500	ID Localización: 10650	ID Localización: 10850
Negocio: CS Municipios	Negocio: CS Municipios	Negocio: CS Municipios	Negocio: CS Municipios
Turno: T	Turno: T	Turno: M	Turno: M
Tipo: Others	Tipo: Others	Tipo: Others	Tipo: Others
Modulo: A	Modulo: A	Modulo: C	Modulo: A
Vehiculo: RÁPIDA	Vehiculo: FXT	Vehiculo: DTN	Vehiculo: FXT
Bolsas esperadas: 54 ± 8	Bolsas esperadas: 66 ± 10	Bolsas esperadas: 61 ± 9	Bolsas esperadas: 21 ± 3
Donantes esperados: 64	Donantes esperados: 80	Donantes esperados: 61	Donantes esperados: 21
Nº conductores: 1	Nº conductores: 1	Nº conductores: 1	Nº conductores: 1
Nº administradores: 1	Nº administradores: 1	Nº administradores: 1	Nº administradores: 1
Nº doctores: 1	Nº doctores: 1	Nº doctores: 1	Nº doctores: 1
Nº enfermeras: 3	Nº enfermeras: 4	Nº enfermeras: 3	Nº enfermeras: 2
Bloqueado: false	Bloqueado: false	Bloqueado: false	Bloqueado: false
Bloqueado modificado: false	Bloqueado modificado: false	Bloqueado modificado: false	Bloqueado modificado: false
Validado: false	Validado: false	Validado: false	Validado: false
Validado modificado: false	Validado modificado: false	Validado modificado: false	Validado modificado: false

Figura 5.8: Pantalla capturada de la pestaña Planificación de la Interfaz Gráfica para el tratamiento de datos almacenados en la BBDD.

### 5.2.2. Planificación

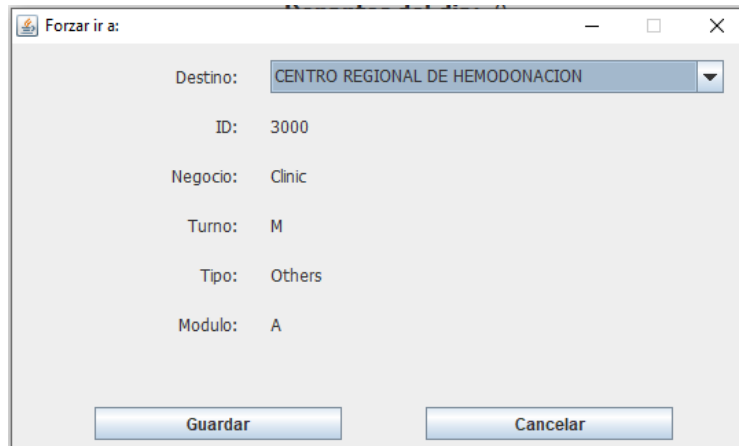
La segunda pestaña de nuestra GUI es la encargada de mostrarnos en detalle los destinos del día seleccionado en la pestaña de Inicio, cuyo contenido se encontraba en el fichero JSON enviado por el sistema optimizador. En el contenido de esta pestaña podemos apreciar en la parte superior información familiar que ya mostrábamos en la pestaña de *Inicio*.

En la zona superior podemos conocer el día que hemos seleccionado en nuestro calendario de *Inicio* junto con el total de bolsas y donantes esperados en dicho día. En la parte inferior de la pestaña podemos encontrar los botones de *Guardar*, *Cancelar* y el botón de *Volver al Inicio*. En la zona central podremos ver la información proporcionada por el sistema optimizador de forma detallada y encima de ella unos botones de selección, *RadioButtons*, que servirán para dar funcionalidad a los botones hallados a la izquierda de la información que estamos mostrando. Estos botones que funcionan junto con los botones de selección, son: *Forzar Ir*, *Editar*, *No Ir*, *Bloquear* y *Validar* cuyo funcionamiento ya hemos explicado en el Capítulo 3.

Como podemos ver en la Figura 5.9a, cuando pulsamos en el botón *Forzar Ir* podemos ver esa ventana que se nos abre donde nos saldrá un desplegable de los posibles sitios de visita que hay guardados en la BBDD para enviarle la petición al sistema optimizador de que tenemos intención de ir.

Al igual sucede cuando pulsamos el botón *Editar*, se nos abrirá la ventana de la Figura 5.9b donde podemos cambiar el vehículo que utilizamos o el destino que nos propone el optimizador.

Cuando accionamos el botón de *No Ir*, en esta ocasión se nos abrirá la ventana vista en la Figura 5.10 donde nos saldrá un mensaje informativo preguntándonos si



Forzar ir a:

Destino: CENTRO REGIONAL DE HEMODONACION

ID: 3000

Negocio: Clinic

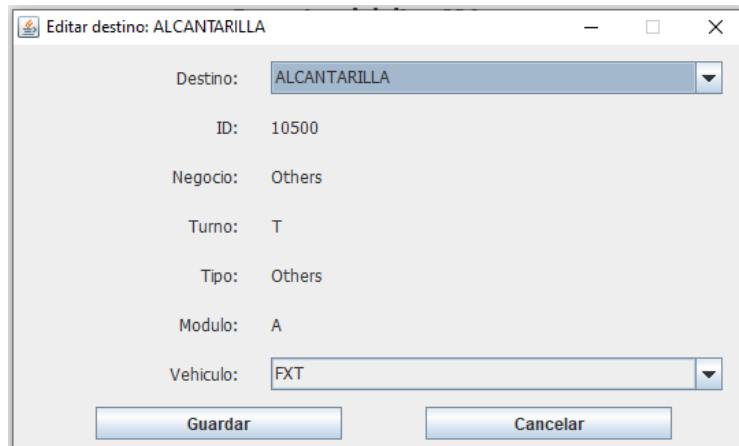
Turno: M

Tipo: Others

Modulo: A

Guardar Cancelar

(a) Captura de pantalla del evento del botón *Forzar Ir* en la pestaña Planificación



Editar destino: ALCANTARILLA

Destino: ALCANTARILLA

ID: 10500

Negocio: Others

Turno: T

Tipo: Others

Modulo: A

Vehiculo: FXT

Guardar Cancelar

(b) Captura de pantalla del evento del botón *Editar* en la pestaña Planificación

Figura 5.9: Pantalla capturada en la pestaña Planificación de la Interfaz Gráfica de los eventos producidos cuando interaccionamos los botones *Forzar Ir* y *Editar*.

estamos seguros de que queremos indicarle al sistema optimizador que no queremos visitar el lugar seleccionado la fecha concreta.

Cuando pulsamos el botón de *Bloquear*, se nos abrirá la ventana informativa de la Figura 5.11a preguntándonos si estamos seguros de bloquear el destino para enviárselo al sistema optimizador.

De forma análoga sucede cuando accionamos el botón *Validar*, en esta ocasión se abrirá la ventana de la Figura 5.11b para preguntarnos si estamos seguros de validar el destino para enviárselo posteriormente al sistema optimizador.

Debemos recordar que toda interacción con estos botones de *Forzar Ir*, *Editar*, *No Ir*, *Bloquear* y *Validar* no produce que se envíe nada al sistema optimizador, para ello debemos interaccionar el botón de *Optimizar* situado en la pestaña *Inicio* como vemos en la Figura 5.5.

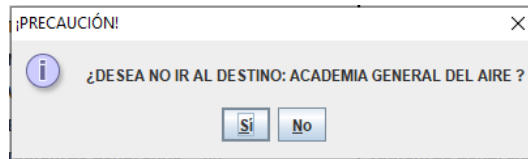
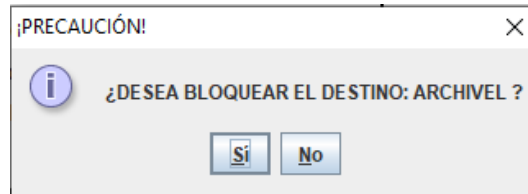
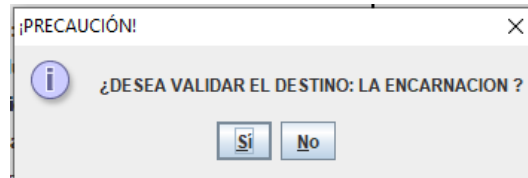


Figura 5.10: Captura de pantalla del evento del botón *No Ir* en la pestaña Planificación



(a) Captura de pantalla del evento del botón *Bloquear* en la pestaña Planificación



(b) Captura de pantalla del evento del botón *Validar* en la pestaña Planificación

Figura 5.11: Pantalla capturada en la pestaña Planificación de la Interfaz Gráfica de los eventos producidos cuando interaccionamos los botones *Bloquear* y *Validar*.

### 5.2.3. Publicidad

En esta ocasión, nuestra pestaña *Publicidad* tiene una estructura muy similar a la pestaña de *Inicio*. Como podemos ver en la Figura 5.12, disponemos de un calendario unilateralmente dependiente al calendario del *Inicio*. Este, nos servirá para que seleccionemos el día que queremos publicitar. Esta dependencia unilateral se produce cuando navegamos por el calendario del *Inicio*, nuestro calendario de *Publicidad* se moverá a la par, sin embargo, si navegamos por en el calendario de *Publicidad*, el calendario de *Inicio* se mantendrá inmóvil, dejando fijas las fechas que habían sido seleccionadas. En la parte derecha de la pestaña, podremos encontrarnos otro cuadro resumen de los lugares que se van a visitar en el día seleccionado con las distintas maneras de avisar a los donantes.

La forma de notificar a las personas que frecuentan el lugar donde se va para realizar las donaciones de sangre, pueden ser de dos formas. La primera es con carteles, como el de la Figura 5.13, avisando a los donantes que se va a ir cierto día y en cierto momento a realizar las donaciones. Cuando nosotros pulsamos el botón de *ABRIR PDF*, lo que hará nuestro programa será realizar un búsqueda en una carpeta que se genera durante la instalación y en el que se guardarán los resultados del sistema optimizador del cartel anunciante correspondiente del lugar que visitaremos filtrando por el mes que se hará dicha visita.

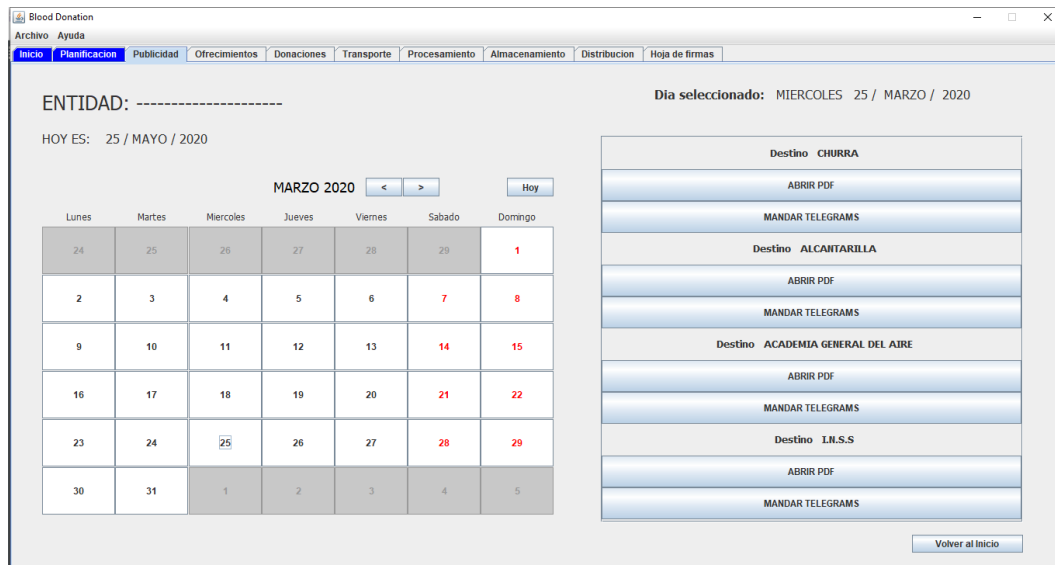


Figura 5.12: Pantalla capturada de la pestaña Publicidad de la Interfaz Gráfica para el tratamiento de datos almacenados en la BBDD.

La segunda vía de publicidad es enviando un mensaje vía Telegram a los donantes de la ciudad. Si pulsamos el botón de *ENVIAR TELEGRAMS*, en esta ocasión se nos abrirá la ventana de la Figura 5.14 para escribir en ella el mensaje que queremos hacer llegar a los donantes inscritos en el grupo de Telegram de la zona que visitaremos.

Además de estos botones, podemos encontrar un último botón en la zona inferior derecha de la pestaña con la inscripción *Volver al Inicio*.

#### 5.2.4. Hoja de firmas

La última pestaña de nuestra Interfaz Gráfica realiza la función de una *Hoja de firmas* del Centro de Hemodonación de Murcia, donde en ella podemos notificar y almacenar alguna incidencia o suceso en particular que haya ocurrido en el lugar de extracción. Servirá para realizar estadísticos de los fallos más comunes, tener un control del personal y en futuras versiones indicar con exactitud los errores de las bolsas de sangre. Hoy en día el uso de estadísticos en la hoja es impracticable debido a que las demás hojas de firmas se realizan en papel y su función en el momento es simplemente para el control de asistencia al trabajo y de las horas utilizadas.

Como podemos ver en la Figura 5.15, en ella podemos ver el resultado de esta pestaña y de los campos para rellenar nuestra hoja. Estos campos son:

- Localidad: Desplegable donde tenemos que seleccionar el destino que vamos a realizar la anotación. Al seleccionarlo, a la derecha del nombre de la localidad, podremos ver el ID correspondiente de ese lugar de extracción.



Figura 5.13: Ejemplo de cartel publicitario devuelto por el sistema optimizador.

- Fecha: Fecha de cuándo se visitó la localidad seleccionada que debe estar escrita en el formato dd/MM/yyyy para la correcta extracción de los datos.
- Vehículo: Vehículo que se ha utilizado para el desplazamiento.
- Hora salida centro, hora final colecta, hora regreso centro: Dichas horas deben estar escritas con el formato HH:mm para la correcta extracción de los datos. En estas secciones debemos indicar la hora hemos salido del Centro de Hemodonación, la hora que hemos terminado la recolección de donaciones y la hora de vuelta al centro. Cuando hayamos rellenado dichos datos, a la derecha de ellos podemos ver el tiempo que ha llevado dicho trabajo en la zona de *HORAS TOTALES COLECTA*. Si ha habido un error al introducir los datos, aparecerá un mensaje en esta sección explicando la fuente de este error.
- Extracciones previstas: Número de donaciones previstas antes de visitar el punto de extracción.

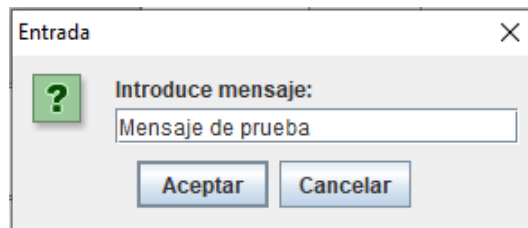


Figura 5.14: Captura de pantalla del evento del botón *ENVIAR TELEGRAMS* en la pestaña Publicidad.

Figura 5.15: Pantalla capturada de la pestaña Hoja de firmas de la GUI para el tratamiento de datos almacenados en la BBDD.

- Donantes atendidos: Número de donantes atendidos.
- Donantes rechazados: Número de donantes rechazados.
- Donantes nuevos: Número de donantes que donan por primera vez.
- Número total de bolsas: Número total de bolsas de sangre recolectadas.
- Selección de personal: Selección del trabajador específico de su puesto de trabajo junto con la función que ha empleado.
- Incidencias: Texto para indicar la incidencia que se ha producido, si hay alguna.
- Sugerencias: Texto para indicar alguna sugerencia o comentario de la colecta.

A la hora de seleccionar a los empleados que han ido al punto de extracción, podremos seleccionar en total a: un médico, dos conductores, dos administrativos y diez enfermeros.



En la parte inferior de nuestra pestaña nos encontramos tres botones distintos, dos en la zona izquierda y uno en la derecha. El botón de la derecha con la inscripción *Volver al Inicio* realizará la misma tarea que hacían los mismos botones en las pestañas de *Planificación* y *Publicidad*. Este, nos permitirá cambiar de pestaña de la actual a la inicial. En cuanto a la zona inferior izquierda de la pestaña, nos encontramos los botones *Enviar* y *Limpiar*. El primer botón dará la orden de recoger todos los datos que hemos escrito en nuestra *Hoja de firmas* y enviarlos a la BBDD, mostrándonos un mensaje informándonos de si el almacenamiento se ha realizado con éxito, o no. Y por último, el botón de *Limpiar* realizará la tarea de eliminar todos los campos que hayan sido modificados en la *Hoja de firmas* y sobrescribirlos con los datos precargados al principio.

#### 5.2.5. Otras pestañas

En esta sección hemos agrupado las siguientes pestañas en un mismo apartado debido a que no forman parte de objetivo principal al desarrollar nuestro proyecto, pero dejamos definidas dichas pestañas para la futura implementación de cualquier interesado que quiera darle la utilidad adecuada.

La labor de las pestañas corresponde a los procesos restantes de las 11 etapas de las que dividimos la carga de trabajo. Estos procesos son: *Ofrecimientos*, *Donaciones*, *Transporte*, *Procesamiento*, *Almacenamiento* y *Distribución*.



## CAPÍTULO 6

# Conclusiones y líneas de futuro

### Conclusiones

En este proyecto tiene como objetivos principales (i) el desarrollo de una herramienta capaz de servir de intercomunicador entre diferentes sistemas, con el fin de para facilitar el trabajo de un administrador en un centro de hemodonación encargado de realizar planificaciones de rutas para visitar Puntos de Colecta en las campañas de donación de sangre, y (ii) la realización de una base de datos que contenga toda la información necesaria para optimizar las planificaciones

Como resultado de estas necesidades se ha desarrollado una Interfaz Gráfica sencilla que sustituya el actual modo de operación de realizar dichas planificaciones en papel y bolígrafo, por un programa que haga más fácil el trabajo del administrador. En cuanto a la necesidad de disponer de una BBDD, se ha desarrollado una con las tablas necesarias para el almacenamiento de los datos. El desarrollo de estos sistemas ha sido considerando la metodología explicada en el Capítulo 3 y llevada a cabo gracias a las herramientas mencionadas en el Capítulo 4.

El conjunto completo del que participan nuestros sistemas está formado por 3 diferentes elementos: la GUI, la BBDD y un sistema optimizador. El sistema optimizador será una herramienta ya implementada en nuestro equipo de trabajo que será el encargado de proporcionarnos las soluciones adecuadas para la optimización de recursos del centro.

En cuando al desarrollo de la interfaz ha sido implementada con éxito. Gracias a ella somos capaces de seleccionar una fecha de la que queremos visualizar la planificación que nos ha proporcionado el sistema optimizador. Podremos ver un pequeño resumen de la organización que se va a llevar en el día o, si lo necesitamos, visualizar esa organización íntegra para apreciar si la propuesta nos resulta de interés o queremos desecharla para que el sistema optimizador nos proporcione una nueva solución. Además, podremos enviarle peticiones personales de interés para las propuestas al optimizador para que las tenga en cuenta al realizar nuevas búsquedas de soluciones.

Además de esta particularidad, nuestra GUI tiene un par de funciones adicionales como la posibilidad de enviar mensajes publicitarios a los donantes que usen la aplicación de Telegram y así que sean conscientes de que se va a realizar una campaña de donación en el lugar que ellos frecuenten. Y por último, su otra función, por petición del centro de hemodonación de la región de Murcia, consistirá en el llenado de un acta que haga las funciones de control y de almacenar datos de interés de la realizada campaña de donación en un Punto de Colecta.

Para el desarrollo de la BBDD se ha implementado con éxito siendo esta capaz de almacenar todo tipo de datos que son necesarios para el cálculo de la planificación realizada por el sistema optimizador. En lo relacionado a la BBDD y nuestra GUI, entre ambos sistemas se pueden enviar mensajes para consultar datos y mostrarlos en la interfaz o incluso añadir, modificar o eliminar información almacenada en la BBDD a través de la GUI.

Una vez explicados los resultados que hemos obtenido en el proyecto, se han dejado preparados unas vías para un interés futuro de añadir nuevas funcionalidades que se pueden implementar en nuestro proyecto para aumentar su productividad y eficiencia que pasaremos a contar a continuación.

## Líneas futuras

Nuestro TFG se ha centrado principalmente en una parte de un proyecto más grande. Como se ha explicado en el Capítulo 1, existen diferentes procesos en los que hemos dividido la carga de trabajo que pueden ser optimizables y pasaremos a contar a continuación.

- Optimización de los demás procesos: El trabajo de las campañas de donación de sangre la hemos dividido en 11 etapas de trabajo más reducidas, con el fin de poder optimizar cada una de estas etapas y mejorar la eficiencia total del trabajo. En el proyecto nos hemos centrado en los procesos de *Inicio*, *Planificación*, *Publicidad* y *Hoja de firmas*, dejando sin dar uso a los procesos restantes de *Transporte*, *Ofrecimientos*, *Donaciones*, *Procesamiento*, *Almacenamiento* y *Distribución* que pueden ser optimizables en futuras implementaciones.
- Resolución del problema de la enfermera: Este problema existente en los centros de hemodonación consiste en que cada empleado tiene un turno en concreto de trabajo. El administrador encargado de planificar las rutas que se deben hacer en el día debe seleccionar qué empleado tiene que ir a cierto lugar cierto día. Esto genera conflicto y una organización extra que debe hacer el administrador pero optimizable. Se podría incluir otro sistema optimizador, o una funcionalidad extra al que usamos, para que también nos proporcionara una planificación de qué empleado debe ir a cada Punto de Colecta.

# Bibliografía

- [1] Rodolfo Baggio, Chris Cooper, Noel Scott, and Magda Antonioli Corigliano. Advertising and word of mouth in tourism, a simulation study. A. Fyall, M. Kozak, L. Andreu, J. Gnoth, & S. Lebe, Marketing innovations for sustainable destinations, pages 13–22, 2009.
- [2] Rong Du, Zhiwen Yu, Tao Mei, Zhitao Wang, Zhu Wang, and Bin Guo. Predicting activity attendance in event-based social networks: Content, context and social influence. In Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, pages 425–434, 2014.
- [3] Albert Farrugia and Daniela Scaramuccia. The dynamics of contract plasma fractionation. Biologicals, 46:159–167, 2017.
- [4] Eduardo Fernández García. Diseño e implementación de una herramienta de modelado y optimización para ayudar a la planificación de colectas de un centro de hemodonación. 2019.
- [5] MEDINFO International Hemoservice. Software de gestión para banco de sangre hematos iig. descripción general. 2020.
- [6] Iñaki Uranga Herrandonera. Desarrollo e implementación de un simulador de hemovigilancia para optimización y análisis de una nueva descripción de los procesos de la cadena transfusional. 2020.
- [7] MAK-SYSTEM. Software de gestión para banco de sangre eprogesa. descripción general. 2020.
- [8] Ashkan Negahban. A hybrid simulation framework for the newsvendor problem with advertising and viral marketing. In 2013 Winter Simulations Conference (WSC), pages 1613–1624. IEEE, 2013.
- [9] Stephanie Sapp, Jon Vaver, Minghui Shi, and Neil Bathia. Dass: Digital advertising system simulation. 2016.
- [10] Yuriy Sherstennikov, Tatyana Rudyanova, Liliia Barannyk, Victoriia Datsenko, and Lyudmyla Novikova. Management of an advertising campaign based on the model of the enterprise’s logistic system. Eastern-European Journal of Enterprise Technologies, 2(3):40–49.



Parte I

Apéndices





## APÉNDICE A

# Campos de Ficheros necesarios para la Instalación

En este Anexo vamos a explicar el contenido de cada fichero y de los campos necesarios. Para realizar la instalación correcta de nuestro programa, durante este proceso se nos pide que seleccionemos unos ficheros con unas características concretas para la correcta extracción de los datos.

### Fichero 1: Localizaciones de Puntos de Colecta

Las cabeceras de este fichero deberán ser:

1. **Código identificación (formato text):** Número de identificación del punto de colecta. Se almacena en formato de texto debido a que un punto de colecta puede tener el caracter “A o B” para indicar si se visita el lugar por la mañana o por la tarde respectivamente.
2. **Colecta (formato text):** Nombre de referencia del punto de colecta.
3. **Ciudad (formato text):** Nombre de la ciudad a la que pertenece el punto de colecta.
4. **Modulo (formato text):** Indicador de la zona a la que pertenece el punto de colecta. En nuestro caso real de la región de Murcia se dispone de 4 módulos: A, B, C y D. En caso de incluir un parámetro de los no definidos se pondrá el valor NOT\_DEFINED.
5. **Turno (formato text):** Indicador de si se debe visitar un sitio por la mañana, por la tarde o en cualquiera de los dos momentos. Para este parámetro debemos incluir el texto M (turno de mañana), T (turno de tarde), BOTH (disponibilidad de los dos turnos) o si no se define nada se le asignará el valor de NOT\_DEFINED.

## 54APÉNDICE A. CAMPOS DE FICHEROS NECESARIOS PARA LA INSTALACIÓN

6. **Tipo (formato text):** Texto que informa de qué tipo es la zona del punto de colecta. Los tipos que soporta nuestro programa son: colectivos, es municipios, cyfse, empresas, hospitales, ies-fp, maratones/especiales, universidad y, si no se define nada, se asignará el valor NOT\_DEFINED.
7. **Frecuencia de visita (formato text):** Texto que nos informará de cuántas veces podemos visitar el punto de colecta al año o en qué momentos podremos ir a realizar las campañas. Las opciones que se pueden dar para especificar este campo se explicarán con su significado cuando se acaben de definir cada uno de los campos de este fichero.
8. **Día específico (formato text):** Texto que nos informará de si hay que visitar el sitio en cuestión un día específico de la semana. Los valores que puede tomar son: lunes, martes, miércoles, jueves, viernes, sabado, domingo y, si no se define nada, se asignará el valor NOT\_DEFINED.

Los datos relacionados a la frecuencia de visita son:

- esporadica: Sin frecuencia definida.
- jl.a: Periodo de visita julio y agosto.
- periodica 1/1: Una visita al mes.
- periodica 1/12: Una visita al año.
- periodica 1/4: Una visita cada 4 meses.
- periodica 1/6: Una visita cada 6 meses.
- periodica 2/1: Dos visitas al mes.
- periodica 2/4: Dos visitas cada 4 meses.
- periodica 2/4+1/2: Dos visitas al mes y una cada dos meses.
- periodica 3/4: Tres visitas cada 4 meses.
- periodica 3/4+1/2: Tres visitas al mes y una cada dos meses.
- periodica 4/4: Cuatro visitas cada 4 meses.
- periodica 4/4+1/2: Cuatro visitas cada 4 meses y una cada dos meses.
- solicitada: Frecuencia solicitada.

### Fichero 2: Histórico de los Puntos de Colecta

En este fichero podremos encontrar información relativa al histórico de datos estadísticos de interés en los puntos de colecta para su optimización. El contenido de este fichero de ser:

1. **Comarca (formato text):** Texto explicativo de la región en la que se encuentra la recolección realizada.
2. **Código identificación (formato text):** Número de identificación del punto de colecta. El mismo campo utilizado en el *Fichero 1*.
3. **Colecta (formato text):** Nombre de referencia del punto de colecta.
4. **Fecha (formato date [mm/DD/yyyy]):** Fecha de realización de la campaña de donación.
5. **Donantes previstos (formato int):** Número de donantes esperados al iniciar la recolección.
6. **Donantes presentados (formato int):** Número de donantes presentados en la recolección.
7. **Donaciones aceptadas (formato int):** Número de donaciones de donantes no rechazados en la selección previa.
8. **Diferencia (formato int):** Resta entre el número de donantes previstos y las donaciones aceptadas.
9. **Número de habitantes (formato int):** Número de habitantes de la zona.

### Fichero 3: Empleados

En este fichero podremos encontrar información relativa a los empleados que realizarán las campañas de donación con su cargo e información correspondiente. El contenido de este fichero de ser:

1. **Apellido 1 (formato text):** Primer apellido del empleado.
2. **Apellido 2 (formato text):** Segundo apellido del empleado.
3. **Nombre (formato text):** Nombre del empleado.
4. **Entidad (formato text):** Nombre de la entidad a la que pertenece el empleado.
5. **Teléfono de contacto (formato int):** Número de teléfono laboral del empleado.
6. **e-Mail (formato text):** E-Mail de contacto del empleado.
7. **Puesto (formato text):** Text explicativo del puesto de trabajo del empleado. Los valores que debemos incluir en este campo son: conductor, enfermero, medico, administrativo, jefe y, si no se define nada, se asignará el valor NOT\_DEFINED.
8. **Contrato (formato text):** Tipo de contrato del empleado. Puede ser “Permanente” o “Externo”.

## 56 APÉNDICE A. CAMPOS DE FICHEROS NECESARIOS PARA LA INSTALACIÓN

9. **Turno (formato text):** Turno de trabajo del empleado. Para este parámetro debemos incluir el texto M (turno de mañana), T (turno de tarde), BOTH (disponibilidad de los dos turnos) o si no se define nada se le asignará el valor de NOT\_DEFINED.
10. **Fecha de alta (formato date [mm/DD/yyyy]):** Fecha de alta laboral del empleado.
11. **Fecha de baja (formato date [mm/DD/yyyy]):** Fecha de baja laboral del empleado.
12. **Estado (formato text):** Texto indicador de si el empleado está trabajando en activo o se encuentra de baja. Debemos indicar en este apartado si el empleado está *Disponible* o *No disponible*.
13. **Observaciones (formato text):** Texto adicional para dejar las anotaciones propias correspondientes sobre el empleado.

### Fichero 4: Vehículos

En este fichero podremos encontrar información relativa a los vehículos que disponemos para los desplazamientos. El contenido de este fichero de ser:

1. **Nombre (formato text):** Nombre del referencia del vehículo.
2. **Entidad (formato text):** Nombre de la entidad a la que pertenece el vehículo.
3. **Pasajeros (formato int):** Número de pasajeros capaces de transportar el vehículo.
4. **Capacidad máxima (formato int):** Capacidad máxima de unidades de sangre que puede transportar el vehículo.
5. **Capacidad máxima ampliada (formato int):** Capacidad máxima adicional de unidades de sangre que puede transportar el vehículo mediante el uso de un elemento de aumento de carga.
6. **Estado (formato text):** Texto indicador de si el vehículo se encuentra disponible o sufre alguna avería. Debemos indicar en este apartado si el vehículo está *Disponible* o *No disponible*.
7. **Observaciones (formato text):** Texto adicional para dejar las anotaciones propias correspondientes sobre el vehículo.

### Fichero 5: Ciudades

En este fichero podremos encontrar información relativa a las ciudades que se encuentran en la zona de operación. El contenido de este fichero de ser:

1. **Nombre (formato text):** Nombre de la ciudad.
2. **Identificador chat (formato int):** Identificador del chat correspondiente para el envío de notificaciones a los donantes.

#### **Fichero 6: Entidades**

En este fichero podremos encontrar información relativa a las entidades que se encuentran en la zona de operación. El contenido de este fichero de ser:

1. **Nombre (formato text):** Nombre de la entidad correspondiente.
2. **Dirección (formato text):** Dirección de la localización de la entidad.
3. **Administrador (formato text):** Nombre del administrador de la entidad.
4. **Año inicio (formato int):** Año que empezó su labor la entidad.
5. **Api Token (formato text):** Identificador token para el uso de notificaciones a los donantes por mensajería.



## APÉNDICE B

# Campos de las Tablas de Datos

En este Anexo vamos a explicar el contenido de cada tabla de datos y de los campos que tienen cada uno. Para realizar la instalación correcta de nuestro programa, durante este proceso se leerán unos ficheros SQL que vendrán ya incluidos en la instalación unos campos concretos.

### **Tabla 1: cities**

El contenido de esta tabla será:

1. ID (formato int): Identificador único de la ciudad.
2. Nombre (formato text): Nombre de la ciudad.
3. Identificador chat (formato int): Identificador del chat correspondiente para el envío de notificaciones a los donantes.

### **Tabla 2: entities**

El contenido de esta tabla será:

1. ID (formato int): Identificador único de la entidad.
2. Nombre (formato text): Nombre de la entidad correspondiente.
3. Dirección (formato text): Dirección de la localización de la entidad.
4. Administrador (formato text): Nombre del administrador de la entidad.
5. Año inicio (formato int): Año que empezó su labor la entidad.
6. Api Token (formato text): Identificador token para el uso de notificaciones a los donantes por mensajería.

**Tabla 3: locations**

El contenido de esta tabla será:

1. Código identificación (formato text): Número de identificación del punto de colecta. Se almacena en formato de texto debido a que un punto de colecta puede tener el caracter “A o B” para indicar si se visita el lugar por la mañana o por la tarde respectivamente.
2. Colecta (formato text): Nombre de referencia del punto de colecta.
3. Ciudad (formato int): Identificador relacionado con la tabla “cities” referenciando a la ciudad correspondiente.
4. Comarca (formato text): Texto explicativo de la región en la que se encuentra el Punto de Colecta.
5. Modulo (formato text): Indicador de la zona a la que pertenece el punto de colecta. En nuestro caso real de la región de Murcia se dispone de 4 módulos: A, B, C y D. En caso de incluir un parámetro de los no definidos se pondrá el valor NOT\_DEFINED.
6. Turno (formato text): Indicador de si se debe visitar un sitio por la mañana, por la tarde o en cualquiera de los dos momentos. Para este parámetro debemos incluir el texto M (turno de mañana), T (turno de tarde), BOTH (disponibilidad de los dos turnos) o si no se define nada se le asignará el valor de NOT\_DEFINED.
7. Tipo (formato int): Indicador relacionado con la tabla “types\_info”, explicada más adelante, con la información del tipo correspondiente.
8. Frecuencia de visita (formato text): Texto que nos informará de cuántas veces podemos visitar el punto de colecta al año o en qué momentos podremos ir a realizar las campañas. Las opciones que se pueden dar para especificar este campo están explicadas en el Apéndice A.
9. Día específico (formato text): Texto que nos informará de si hay que visitar el sitio en cuestión un día específico de la semana. Los valores que puede tomar son: lunes, martes, miércoles, jueves, viernes, sabado, domingo y, si no se define nada, se asignará el valor NOT\_DEFINED.

**Tabla 4: personnel**

El contenido de esta tabla será:

1. ID (formato int): Identificador único del trabajador.
2. Apellido 1 (formato text): Primer apellido del empleado.
3. Apellido 2 (formato text): Segundo apellido del empleado.



4. Nombre (formato text): Nombre del empleado.
5. Entidad (formato text): Nombre de la entidad a la que pertenece el empleado.
6. Teléfono de contacto (formato int): Número de teléfono laboral del empleado.
7. e-Mail (formato text): E-Mail de contacto del empleado.
8. Puesto (formato int): Indicador relacionado con la tabla “jobs\_info”, explicada más adelante, con la información del tipo correspondiente.
9. Contrato (formato text): Tipo de contrato del empleado. Puede ser “Permanente” o “Externo”. Si no se especifica nada se le pondrá el valor Externo por defecto.
10. Turno (formato text): Turno de trabajo del empleado. Para este parámetro debemos incluir el texto M (turno de mañana), T (turno de tarde), BOTH (disponibilidad de los dos turnos) o si no se define nada se le asignará el valor de NOT\_DEFINED.
11. Fecha de alta (formato date [mm/DD/yyyy]): Fecha de alta laboral del empleado.
12. Fecha de baja (formato date [mm/DD/yyyy]): Fecha de baja laboral del empleado.
13. Estado (formato text): Texto indicador de si el empleado está trabajando en activo o se encuentra de baja. Debemos indicar en este apartado si el empleado está *Disponible* o *No disponible*. Si no se asigna un valor inicial, se pondrá el valor por defecto *No Disponible*.
14. Observaciones (formato text): Texto adicional para dejar las anotaciones propias correspondientes sobre el empleado.

#### **Tabla 5: vehicles**

El contenido de esta tabla será:

1. ID (formato int): Identificador único del vehículo.
2. Nombre (formato text): Nombre del referencia del vehículo.
3. Entidad (formato text): Nombre de la entidad a la que pertenece el vehículo.
4. Pasajeros (formato int): Número de pasajeros capaces de transportar el vehículo.
5. Capacidad máxima (formato int): Capacidad máxima de unidades de sangre que puede transportar el vehículo.

6. Capacidad máxima ampliada (formato int): Capacidad máxima adicional de unidades de sangre que puede transportar el vehículo mediante el uso de un elemento de aumento de carga.
7. Estado (formato text): Texto indicador de si el vehículo se encuentra disponible o sufre alguna avería. Debemos indicar en este apartado si el vehículo está *Disponible* o *No disponible*.
8. Observaciones (formato text): Texto adicional para dejar las anotaciones propias correspondientes sobre el vehículo.

**Tabla 6: statistics**

El contenido de esta tabla será:

1. Código identificación (formato text): Número de identificación del punto de colecta. El mismo campo utilizado en el *Fichero 1*.
2. Tipo de dato (formato int): Indicador relacionado con la tabla “data\_type\_info”, explicada más adelante, con la información del tipo correspondiente.
3. Dato (formato int): Valor guardado.
4. Mes (formato int): Mes del año al que pertenece el dato.

**Tabla 7: data\_type\_info**

El contenido de esta tabla es la siguiente (int, text):

- (1, 'mean\_planned'): Media aritmética de donaciones previstas.
- (2, 'q1\_planned'): Primer cuartil de donaciones previstas.
- (3, 'q2\_planned'): Segundo cuartil de donaciones previstas.
- (4, 'q3\_planned'): Tercer cuartil de donaciones previstas.
- (5, 'sta\_dev\_planned'): Desviación típica de donaciones previstas.
- (6, 'mean\_presented'): Media aritmética de los donantes presentados.
- (7, 'q1\_presented'): Primer cuartil de los donantes presentados.
- (8, 'q2\_presented'): Segundo cuartil de los donantes presentados.
- (9, 'q3\_presented'): Tercer cuartil de los donantes presentados.
- (10, 'sta\_dev\_presented'): Desviación típica de los donantes presentados.
- (11, 'mean\_donations'): Media aritmética de las donaciones realizadas.

- (12, 'q1\_donations'): Primer cuartil de las donaciones realizadas.
- (13, 'q2\_donations'): Segundo cuartil de las donaciones realizadas.
- (14, 'q3\_donations'): Tercer cuartil de las donaciones realizadas.
- (15, 'sta\_dev\_donations'): Desviación típica de las donaciones realizadas.
- (16, 'difference'): Diferencia entre los datos previstos y las donaciones realizadas.
- (17, 'visits'): Número de veces que se ha visitado el Punto de Colecta.

#### **Tabla 7: days\_info**

El contenido de esta tabla es la siguiente (int, text):

- (1, 'lunes')
- (2, 'martes')
- (3, 'miercoles')
- (4, 'jueves')
- (5, 'viernes')
- (6, 'sabado')
- (7, 'domingo')
- (8, 'not\_defined')

#### **Tabla 7: types\_info**

El contenido de esta tabla es la siguiente (int, text):

- (1, 'colectivos')
- (2, 'cs\_municipios')
- (3, 'cyfse')
- (4, 'empresas')
- (5, 'hospitales')
- (6, 'ies\_fp')
- (7, 'maratones\_especiales')
- (8, 'universidad')
- (9, 'not\_defined')

**Tabla 8: jobs\_info**

El contenido de esta tabla es la siguiente (int, text):

- (1, 'enfermero')
- (2, 'medico')
- (3, 'administrativo')
- (4, 'conductor')
- (5, 'jefe')
- (6, 'not\_defined')

**Tabla 9: signature\_sheet**

El contenido de esta tabla es la siguiente:

- Localidad (formato text): Identificador del Punto de Colecta.
- Fecha (formato date [dd/MM/yyyy]): Fecha de visita al Punto de Colecta.
- departure\_time (formato time): Hora de salida del centro de hemodonación.
- donation\_end\_time (formato time): Hora de finalización de la campaña de donación.
- arrival\_time (formato time): Hora de vuelta al centro de hemodonación.
- expected\_donations (formato int): Número de donantes esperados.
- attended\_donors (formato int): Número de donantes atendidos.
- total\_blood\_bags (formato int): Número de bolsas recolectadas.
- rejected\_donors (formato int) Número de donantes rechazados.
- new\_donors (formato int): Número de nuevos donantes.
- vehicle\_id (formato int): Identificador del vehículo utilizado para ir al Punto de Colecta.
- doctor\_id (formato int): Identificador del médico que ha visitado el Punto de Colecta.
- doctor\_job\_function (formato int): Identificador de la labor del médico al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- driver1\_id (formato int): Identificador del conductor que ha visitado el Punto de Colecta.

- driver1\_job\_function (formato int): Identificador de la labor del conductor al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- driver2\_id (formato int): Identificador del conductor que ha visitado el Punto de Colecta.
- driver2\_job\_function (formato int): Identificador de la labor del conductor al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- admin1\_id (formato int): Identificador del administrador que ha visitado el Punto de Colecta.
- admin1\_job\_function (formato int): Identificador de la labor del administrador al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- admin12\_id (formato int): Identificador del administrador que ha visitado el Punto de Colecta.
- admin2\_job\_function (formato int): Identificador de la labor del administrador al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- nurse1\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse1\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- nurse2\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse2\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- nurse3\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse3\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- nurse4\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse4\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- nurse5\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse5\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions.info”.
- nurse6\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.

- nurse6\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions\_info”.
- nurse7\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse7\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions\_info”.
- nurse8\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse8\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions\_info”.
- nurse9\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse9\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions\_info”.
- nurse10\_id (formato int): Identificador del enfermero que ha visitado el Punto de Colecta.
- nurse10\_job\_function (formato int): Identificador de la labor del enfermero al realizar el trabajo relacionado con la tabla de datos “jobs\_functions\_info”.
- incidences (formato text): Texto indicativo si ha sucedido alguna incidencia.
- material\_suggestions (formato text): Texto indicativo de las sugerencias pedidas.

#### **Tabla 10: contraindications**

El contenido de esta tabla es la siguiente:

- code\_id (formato text): Identificador de la contraindicación.
- type (formato text): Tipo de la contraindicación (T o D).
- name (formato text): Nombre de la contraindicación.
- duration\_days (formato int): Tiempo de duración de la contraindicación.

#### **Tabla 11: rejections**

El contenido de esta tabla es la siguiente:

- location\_code (formato text): Identificador del Punto de Colecta.
- date (formato date): Fecha de referencia.

- interviewer\_id (formato int): Identificador del trabajador.
- contraindication\_name (formato text): Nombre de la contraindicación.
- comment (formato int): Indicador del comentario.
- rejection\_id (formato int): Número de identificador.

**Tabla 12: jobs\_functions\_info**

El contenido de esta tabla es la siguiente (int, int, text), el primer parámetro hace referencia al identificador de la tabla, el segundo parámetro al trabajo que está relacionado de la tabla “jobs\_info”.

- (1, 2, 'entrevista')
- (2, 2, 'laboratorio')
- (3, 2, 'entrevista y laboratorio')
- (4, 1, 'entrevista')
- (5, 1, 'laboratorio')
- (6, 1, 'entrevista y laboratorio')
- (7, 1, 'laboratorio y extracción')
- (8, 1, 'entrevista, laboratorio y extracción')
- (9, 4, 'transporte')
- (10, 4, 'publicidad')
- (11, 4, 'transporte y publicidad')
- (12, 3, 'ordenador')





## APÉNDICE C

# Aspectos éticos, económicos, sociales y ambientales

En este apéndice se analizan los aspectos éticos, económicos, sociales y ambientales de este TFG.

### Introducción

Con el desarrollo de las tecnologías de la información este trabajo podrá ser consultado desde cualquier parte del mundo, por lo tanto podrá tener un impacto global.

El desarrollo de este trabajo genera un ahorro monetario para los centros de hemodonación y puede servir de inspiración para el desarrollo de otras plataformas similares adaptadas a las necesidades de cada centro.

Esperamos que la realización de este trabajo tenga un impacto notable en la eficiencia y en la comodidad de los administradores de los centros de hemodonación, para facilitar su labor adaptado a las nuevas tecnologías.

Por todo esto, podemos concluir que busca mejorar la vida laboral de los empleados en dichos centros.

### Responsabilidad ética y profesional

Al tratar con datos estadísticos de las recolecciones en las campañas de donación, en este trabajo no hemos necesitado cumplir normas de protección y privacidad que marca la Unión Europea, ya que se emplean datos reales pero numéricos. Además, se ha buscado cumplir con los requisitos de propiedad intelectual de los recursos de terceros usados y a su vez, que estos recursos, contasen con licencia de código abierto.

## **Impacto económico**

Como indicamos en las conclusiones del Capítulo 6, este trabajo pretende generar una optimización de los recursos de los centros de hemodonación de forma directa, propiciando el ahorro de material, personal y temporal. Además, se espera, que este desarrollo se pueda expandir a más estudios como este.

## **Impacto social**

El trabajo va dirigido en especial a los trabajadores de los centros de hemodonación con la función de mejorar su vida laboral. Fomentando así la eficiencia en el trabajo como la felicidad del mismo trabajador.

## **Impacto ambiental**

Siendo la electricidad y los equipos electrónicos los únicos recursos que se emplean para el desarrollo de este trabajo, podemos decir que el desarrollo de este trabajo no tiene un impacto ambiental directo pero sí indirecto. Tendrá un ahorro de recursos como combustible o la necesidad de menor equipo médico lo que conlleva a una menor contaminación producida por el equipo de trabajo y la generación de menos residuos.

## **Conclusiones**

Este proyecto ayudará a mejorar la eficiencia y la vida laboral de los empleados de los centros de hemodonación, como de un ahorro de los recursos utilizados por los mismos centros. De esta forma, queremos ayudar en mejorar la vida de los trabajadores empleando tan importante labor.

## APÉNDICE D

# Presupuesto económico

El presupuesto consta de los siguientes conceptos:

- Coste de recursos materiales
- Costes de mano de obra
- Costes totales del proyecto
- Gastos generales y beneficio industrial
- Presupuesto total

A continuación se detalla cada uno de ellos.

### Costes de recursos materiales

El coste en recursos materiales hace referencia a los recursos utilizados por el desarrollador (ordenador, software y demás). A continuación, en el Tabla D.1, encontramos los recursos materiales utilizados a lo largo del proyecto.

Cuadro D.1: Coste de los recursos materiales

Concepto	Coste	Amortización	Coste real
Ordenador personal (software incluido)	1000,00€	20 %	200,00€
Coste de la web	15,00€		15,00€
Servidor AWS	60,00€		60,00€
Coste del SSL	15,00€		15,00€
<b>Coste total:</b>			<b>290,00€</b>

## Costes de mano de obra

Para la realización del proyecto es necesario el trabajo de un FullStack Developer encargado del desarrollo de la plataforma web. Este perfil se enmarca dentro del grupo de cotización 1 del Régimen General de Seguridad Social, con una jornada laboral de 8 horas al día, 40 horas a la semana y 21 días al mes. El desglose completo se puede consultar en el Cuadro D.2.

Cuadro D.2: Coste de mano de obra

Puesto	Horas	Precio/Hora	Total
FullStack Developer	420	16€	6.720,00€
Seguridad Social	420	5,60€	2.352,00€
<b>Coste total:</b>			<b>9.072,00€</b>

## Coste total del proyecto

El coste total de los recursos está formado por el coste de recursos materiales y de mano de obra. Su desglose lo podemos ver en el Cuadro D.3.

Cuadro D.3: Coste total del proyecto

Concepto	Coste
Recursos materiales	290,00€
Mano de obra	9.072,00€
<b>Total</b>	<b>9.362,00€</b>

## Gastos generales y beneficio industrial

Aquí se incluyen todos aquellos gastos indirectos derivados de la utilización de instalaciones, amortizaciones, gastos fiscales, etc. Con esto, el presupuesto para gastos y beneficio quedan como vemos en el Cuadro D.4.

Cuadro D.4: Gastos generales y beneficio industrial

Concepto	Coste
Presupuesto de ejecución material	9.362,00€
Gastos generales (16 % del P.E.M.)	1.497,92€
Beneficio industrial (6 % del P.E.M.)	561,72€
<b>Total gastos generales y beneficio industrial</b>	<b>11.421,44€</b>

## Presupuesto total

Aplicando el 21 % del Impuesto sobre el Valor Añadido (IVA), se obtiene el presupuesto total del Cuadro D.5. El presupuesto total del proyecto asciende a diez mil seiscientos treinta y un euros y cincuenta y nueve céntimos.

Cuadro D.5: Presupuesto total

Concepto	Coste
Presupuesto Subtotal	11.421,44€
I.V.A. (21 %)	2.398,50€
<b>Presupuesto total</b>	<b>13.819,94€</b>

Madrid, 18 de junio de 2020

**Fdo.:** Pablo Ramos Izquierdo