



**Universidad
Zaragoza**

UNIVERSIDAD DE ZARAGOZA

Trabajo Fin de Grado

Grado en Ingeniería Informática

Diseño e implementación de un sistema web para la
optimización de vías clínicas mediante métodos
formales

*Design and implementation of a web system for the
optimization of clinical pathways using formal methods*

Autor

Miguel Aréjula Aísa

Directores

Cristian Mahulea Poleuca
José Javier Merseguer Hernaiz

Escuela de Ingeniería y Arquitectura

2025

Resumen

El presente Trabajo de Fin de Grado se enmarca en el proceso de transformación digital del sistema sanitario y tiene como objetivo principal desarrollar una herramienta informática que facilite la digitalización, estructuración y monitorización de vías clínicas en procedimientos médicos complejos. Esta necesidad surge a partir de una problemática real identificada en el servicio de Traumatología y Cirugía Ortopédica del Hospital Clínico Universitario “Lozano Blesa”, donde los procedimientos clínicos se registran actualmente de forma manual o con herramientas poco estructuradas, dificultando el seguimiento de los pacientes y la evaluación objetiva de las prácticas médicas.

Como respuesta a este contexto, se ha diseñado y desarrollado un sistema web que permite crear nuevas vías clínicas a partir de fases y tareas definidas, representarlas mediante redes de Petri, utilizarlas como guía durante el seguimiento del paciente y registrar el proceso realizado para cada paciente durante sus operaciones. El sistema ofrece una visualización estructurada del cumplimiento de dichas vías, facilitando así la detección de desviaciones, el análisis de su efectividad y la mejora progresiva basada en datos reales. Además, permite una adaptación dinámica a distintos procedimientos médicos, lo que lo convierte en una herramienta versátil para distintos servicios hospitalarios.

Este trabajo ha sido desarrollado en el marco del proyecto nacional de investigación *TED2021-130449B-I00* y está alineado con su quinto objetivo: *Identificar desviaciones de las vías clínicas mediante la monitorización de pacientes*. Para garantizar la adecuación de la herramienta a las necesidades reales del entorno clínico, se ha mantenido una colaboración estrecha con el personal médico del hospital, mediante reuniones periódicas para la recolección de requisitos, validación de prototipos y verificación funcional del sistema. Como base inicial, se tomaron como referencia dos vías clínicas reales; las correspondientes a artroplastia de cadera y de rodilla, que sirvieron como modelo representativo para la definición de funcionalidades.

Durante el desarrollo del sistema se adoptó un enfoque sistemático, dividido en fases bien definidas (análisis de requisitos, diseño, implementación, pruebas e integración), que permitió aplicar de forma práctica los conocimientos adquiridos durante la formación. La metodología empleada fue iterativa e incremental, con revisiones frecuentes basadas en el *feedback* del personal clínico, lo que garantizó que el sistema evolucionara en consonancia con sus necesidades reales. A nivel técnico, se optó por una arquitectura cliente-servidor modular, con microservicios independientes para funcionalidades críticas, una base de datos diseñada conforme a los flujos de trabajo médicos, y una interfaz centrada en la claridad, la eficiencia y la coherencia con la práctica clínica diaria.

En definitiva, este trabajo ha supuesto una oportunidad única para aplicar conocimientos teóricos a un contexto real, con impacto directo en la práctica clínica hospitalaria. La construcción de una herramienta funcional, validada por profesionales y alineada con sus flujos de trabajo, ha sido el resultado de un proceso riguroso de análisis, diseño, implementación y validación. Más allá de la solución técnica desarrollada, el aprendizaje obtenido a lo largo de todo el proyecto, desde la interacción con los usuarios hasta la toma de decisiones arquitectónicas, representa un crecimiento significativo tanto a nivel profesional como personal. Este proyecto no solo demuestra la viabilidad técnica de la propuesta, sino también su potencial de escalabilidad y mejora continua en entornos reales donde la tecnología puede marcar una diferencia tangible.

Agradecimientos

Quiero comenzar agradeciendo sinceramente a mis tutores, Cristian Mahulea y José Merseguer. Más allá de su conocimiento y experiencia, que han sido clave para sacar este trabajo adelante, les estoy especialmente agradecido por su cercanía, su disponibilidad constante y por haber confiado en mí desde el principio. Siempre estuvieron ahí cuando me surgían dudas, bloqueos o inseguridades, aportando claridad y motivación. He aprendido muchísimo gracias a ellos, no solo sobre este proyecto, sino también sobre cómo enfrentar con criterio y rigor los desafíos del mundo profesional.

También quiero dar las gracias a los médicos del hospital que han colaborado conmigo durante este proyecto. Su implicación, su tiempo y su disposición para compartir conocimientos han sido esenciales para entender las necesidades reales del entorno clínico. Gracias por su paciencia al explicarme con detalle conceptos que me eran ajenos, por sus sugerencias constructivas y por su compromiso con mejorar el sistema desde una perspectiva humana y profesional. Su visión ha sido clave para dar sentido y dirección al trabajo que hay detrás de este proyecto.

Tampoco puedo dejar de mencionar a mis amigos, que han sido un verdadero salvavidas en todo este viaje. A los de siempre, que me han acompañado desde mucho antes de empezar esta aventura y que han sabido darme aliento, risas y perspectiva cuando más lo necesitaba. A de la universidad, con quienes compartí aprendizajes, retos y muchas experiencias que enriquecieron este proceso. Y, por supuesto, a los amigos que hice durante el Erasmus: gracias por enseñarme a disfrutar del presente, por los planes improvisados y por hacer que unos pocos meses se sintieran como un pedazo de vida.

Por último, pero sin duda lo más importante, gracias a mi familia por estar ahí en cada paso del camino. Gracias por la paciencia, por los ánimos, por hacerme sentir que podía con todo incluso cuando yo dudaba. Quiero hacer una mención muy especial a mi hermano, que ha sido mi mayor referente y apoyo en estos cuatro años. Ha estado ahí para guiarme, para escucharme, para aconsejarme con calma y sabiduría. Ha sido mi brújula en más de una ocasión y no tengo palabras suficientes para agradecerle todo lo que ha hecho por mí. Sin él, este camino habría sido mucho más difícil y sin duda, mucho menos bonito.

Índice general

1.	Introducción	5
1.1.	Problema abordado	5
1.2.	Contexto del proyecto	6
1.3.	Estructura de la memoria	6
2.	Especificación funcional	7
2.1.	Diccionario de términos	7
2.2.	<i>Stakeholders</i>	8
2.3.	Requisitos funcionales de los médicos	8
2.4.	Requisitos funcionales del administrador-médico	9
2.5.	Requisitos no funcionales	9
2.6.	Flujos de trabajo del administrador-médico	9
2.7.	Resumen de la especificación funcional	10
3.	Prototipo de la interfaz	13
4.	Arquitectura del sistema	15
4.1.	Descomposición en microservicios	16
4.2.	Tecnologías utilizadas	17
5.	Modelo de datos	18
5.1.	Modelo Entidad-Relación	18
5.2.	Modelo relacional de la base de datos	19
6.	Metodología de desarrollo	22
6.1.	Análisis de requisitos	22
6.2.	Diseño del sistema	23
6.3.	Implementación del sistema	23
6.4.	Validación	24
7.	Producto final	25
8.	Conclusiones	32

8.1.	Trabajo Futuro	33
8.2.	Problemas encontrados	34
8.3.	Lecciones aprendidas	34
	Bibliografía	36
	Glosario	38
	Acrónimos	39
	A. Detalle de los requisitos del sistema	42
	B. Prototipo completo	45
	C. Diagrama de Gantt	56

1. Introducción

La transformación digital del sistema sanitario representa un reto urgente y necesario, especialmente en aquellos entornos donde los procesos clínicos aún se gestionan de forma manual o con herramientas poco estructuradas. Este Trabajo de Fin de Grado surge con el propósito de contribuir a esa modernización, abordando la necesidad de una herramienta que permita digitalizar, estructurar y monitorizar las vías clínicas utilizadas en procedimientos médicos complejos. A lo largo de esta sección se describe en detalle el problema identificado en un contexto hospitalario real, así como los objetivos planteados para dar respuesta a dicha necesidad mediante el desarrollo de una solución tecnológica adaptada al entorno clínico. Además, se establece el marco de investigación en el que se plantea este proyecto.

1.1. Problema abordado

Vivimos en una era marcada por la transformación digital, donde sectores de todo tipo buscan modernizar sus procesos mediante herramientas tecnológicas que aumenten la eficiencia, la trazabilidad y la capacidad de análisis. Esta digitalización ha demostrado ser clave no solo para optimizar recursos, sino también para mejorar la calidad del servicio ofrecido. Sin embargo, esta transición no se produce de forma uniforme en todas las áreas. Existen aún numerosos entornos en los que los procesos continúan llevándose a cabo manualmente, con herramientas poco estructuradas, lo que limita considerablemente el aprovechamiento de los datos generados.

El ámbito sanitario no es ajeno a esta realidad. A pesar de los avances tecnológicos disponibles, muchos hospitales siguen gestionando procedimientos clínicos críticos mediante métodos manuales o semi-estructurados, como hojas de cálculo. Este es el caso del servicio de Traumatología del Hospital Clínico Universitario (HCU) “Lozano Blesa”, donde actualmente el seguimiento de pacientes y la documentación de los procedimientos realizados durante una operación se lleva a cabo sin el apoyo de un sistema digital especializado.

En este contexto, el presente proyecto aborda dos problemas principales. Por un lado, la necesidad de optimizar la recolección y gestión de datos clínicos relevantes que permitan tanto el seguimiento de pacientes como la evaluación objetiva de las prácticas médicas. Por otro, la falta de una herramienta que facilite la creación, edición y visualización de vías clínicas de forma estructurada y adaptable. Las vías clínicas son fundamentales para estandarizar procedimientos médicos complejos, y su correcta implementación requiere de herramientas que no solo permitan su definición formal, sino también su actualización dinámica a partir de datos reales.

Para resolver estos desafíos, se propone el desarrollo de un sistema web que permita crear nuevas vías clínicas a partir de la definición de fases y tareas, representarlas mediante modelos de redes de Petri [14], y utilizarlas como guía para registrar el progreso de los pacientes. Además, el sistema debe facilitar la visualización estructurada del cumplimiento de dichas vías, permitiendo así identificar desviaciones, evaluar su efectividad y alimentar procesos de mejora continua [7]. De este modo, la herramienta no solo digitaliza procesos actualmente manuales, sino que proporciona una base sólida para avanzar hacia una medicina más estructurada, medible y basada en datos.

1.2. Contexto del proyecto

La herramienta desarrollada en este proyecto forma parte del marco de investigación de la Universidad de Zaragoza, concretamente del proyecto de I+D+i con referencia **TED2021-130449B-I00**, titulado *Evaluación y optimización de vías clínicas mediante métodos formales*. Se trata de un proyecto de ámbito nacional orientado a avanzar en la digitalización del sistema sanitario, mediante el uso de técnicas de minería de procesos, modelado formal y análisis de datos clínicos.

Este trabajo se enmarca específicamente dentro del quinto objetivo del proyecto (**O5**): *Identificar desviaciones de las vías clínicas mediante la monitorización de pacientes*. En este contexto, el software desarrollado tiene como propósito registrar de forma estructurada y fiable los procedimientos médicos realizados durante las distintas fases de una operación, con el fin de evaluar el cumplimiento real de las vías clínicas previamente definidas. Además, el sistema persigue la mejora continua de dichas vías mediante su actualización progresiva, utilizando los datos recopilados durante su aplicación clínica. El objetivo final es lograr versiones cada vez más refinadas y eficientes de las vías clínicas, ajustadas a la realidad asistencial, al contexto específico de cada paciente y a la experiencia acumulada por los profesionales sanitarios.

Para alcanzar este fin, ha sido esencial la colaboración directa con el equipo médico del HCU, mediante entrevistas y reuniones. En las mismas se ha planteado el problema, se han recogido los requisitos funcionales, se han discutido los prototipos y se ha validado el sistema. Este enfoque colaborativo ha permitido diseñar una herramienta adaptada a las necesidades reales del entorno clínico, con una interfaz intuitiva, funcionalidades específicas para cada perfil de usuario, y una base de datos capaz de reflejar la complejidad del flujo clínico.

Como punto de partida, durante las reuniones iniciales con el equipo médico, se tomaron como referencia dos vías clínicas reales, las correspondientes a operaciones de artroplastia de cadera y de rodilla. Estas sirvieron como ejemplos representativos para definir los primeros requisitos funcionales del sistema. Sin embargo, el objetivo del software va mucho más allá del simple registro de estos casos específicos: se ha diseñado como una herramienta flexible y extensible que permite al personal médico y administrativo crear nuevas vías clínicas desde cero, definir sus fases y tareas asociadas, y posteriormente monitorizar su cumplimiento en operaciones reales. De este modo, el sistema busca facilitar no solo la digitalización del proceso clínico, sino también su evolución dinámica a partir de la experiencia asistencial recogida en el día a día.

Además, en etapas anteriores de dicho proyecto, se desarrolló un algoritmo capaz de generar automáticamente vías clínicas a partir de datos clínicos estructurados¹. Este algoritmo permite construir vías clínicas personalizadas y estandarizadas, basándose en la información médica disponible, con el fin de facilitar la toma de decisiones clínicas y promover la eficiencia en la atención al paciente. Desde el punto de vista técnico, el algoritmo ha requerido ciertas modificaciones para adaptarse a una mayor variabilidad en las estructuras de entrada y el formato de la salida.

1.3. Estructura de la memoria

La presente memoria se organiza de forma lógica y progresiva, con el objetivo de reflejar todas las fases que han intervenido en el desarrollo del sistema, desde su concepción hasta su implementación

¹El código fuente del algoritmo se encuentra disponible en <https://gitlab.crans.org/bleizi/arpe-identification>

final. En primer lugar, se expone la especificación funcional del sistema, en la que se definen los requisitos funcionales y no funcionales. Estos fueron recopilados a lo largo de varias reuniones con el personal clínico. Los requisitos completos se encuentran detallados en el anexo A.

Posteriormente, se describe en detalle la arquitectura del sistema, diseñada siguiendo un modelo cliente-servidor con un enfoque modular. Esta sección presenta los componentes principales que conforman la solución y analiza cómo se comunican entre sí mediante Application Program Interface (API) Representational State Transfer (REST) y protocolos seguros. Asimismo, se profundiza en las decisiones de diseño adoptadas y se explican las tecnologías empleadas, destacando su relevancia para la escalabilidad, mantenibilidad y evolución futura del sistema.

Seguidamente, se presenta el modelo de datos, que describe la estructura de la información manejada por el sistema. Esta sección incluye tanto el diagrama entidad-relación como su correspondiente modelo relacional, lo que permite comprender cómo se almacena y relaciona la información clínica dentro de la aplicación.

A continuación, se introduce la metodología empleada para el desarrollo del proyecto. En esta sección se explican las decisiones adoptadas durante todo el proceso, así como las prácticas utilizadas para garantizar el éxito en el proyecto.

Finalmente, la memoria concluye con la presentación del producto final y las conclusiones generales del trabajo. En estos apartados se recogen los resultados alcanzados, junto con una valoración crítica del sistema construido. Además, se plantean posibles líneas de mejora y evolución futura, tanto desde el punto de vista funcional como técnico, con el objetivo de guiar su continuidad en próximas fases del proyecto de investigación.

2. Especificación funcional

En un proceso de ingeniería de software [8], una de las etapas esenciales es la definición clara y precisa de los requisitos funcionales del sistema. Esta fase resulta crucial para garantizar que el desarrollo se alinee con las necesidades reales de los usuarios y con los objetivos del proyecto. En este caso, la información recogida en esta sección ha sido obtenida mediante un proceso iterativo de reuniones y entrevistas con los médicos especialistas que participan en el proyecto, quienes aportaron su conocimiento sobre el flujo de trabajo clínico y los requerimientos operativos. Este proceso se explica en profundidad en la Sección 6. Gracias a esta colaboración, se ha podido construir un vocabulario común, identificar a los distintos actores involucrados (*stakeholders*) y detallar de forma estructurada los requisitos funcionales y no funcionales del sistema. Esta especificación servirá como base para el diseño técnico y la validación de la solución desarrollada.

2.1. Diccionario de términos

En todo proceso de desarrollo de software orientado a un dominio específico, especialmente en contextos tan sensibles y especializados como el entorno clínico, es fundamental establecer un lenguaje común entre los desarrolladores y los usuarios finales. El diccionario de términos constituye una herramienta clave para garantizar esta comprensión mutua, ya que permite formalizar los conceptos esenciales que serán utilizados a lo largo del sistema.

- **Paciente (T1)**, persona que va a ser sometida a una operación de la cual se guarda su número de historia clínica, nombre, apellidos, edad, sexo y opcionalmente el índice de masa corporal en el momento de la operación.
- **Operación (T2)**, procedimiento médico que se realiza sobre un paciente para tratar una condición médica.
- **Vía clínica (T3)**, conjunto estructurado y estandarizado de prácticas y procedimientos que se deben seguir para la realización de una operación. Se compone de una secuencia de acciones que deben llevar acabo los médicos durante el ciclo completo de una operación, desde la planificación de la operación hasta el post-operatorio.
- **Tareas (T4)**, acciones que realizan los médicos para llevar acabo una vía clínica.

2.2. Stakeholders

La identificación de los *stakeholders* es una etapa crítica dentro del proceso de especificación funcional, ya que permite delimitar quiénes son los actores relevantes en el uso y la gestión del sistema, y qué necesidades específicas deben ser consideradas en su diseño. En este proyecto, los *stakeholders* principales son profesionales del ámbito sanitario con distintos niveles de responsabilidad y acceso a la información clínica.

- **Médicos**, son aquellos que van a hacer uso del sistema registrando los procedimientos realizados durante la vía clínica.
- **Administrador-médico**, se encarga de gestionar los usuarios y el acceso de estos a las distintas vías de los pacientes. Además es el único con acceso a todos los datos del sistema mediante la exportación de los mismos.
- **Pacientes**, son el foco principal de las vías clínicas. Aunque no interactúan directamente con el sistema, su información y evolución médica son gestionadas a través de él, por lo que sus necesidades y privacidad deben ser consideradas.
- **Desarrolladores**, son responsables de construir, mantener y evolucionar el sistema. Su función es implementar las funcionalidades requeridas por los usuarios, garantizar la calidad del software y adaptar el sistema a las necesidades clínicas y técnicas.
- **Equipo IT del hospital**, se encargará de proporcionarnos la infraestructura tecnológica existente en el hospital, así como de definir las condiciones técnicas necesarias para el despliegue y la integración del sistema dentro del entorno institucional.

2.3. Requisitos funcionales de los médicos

Los médicos constituyen el grupo de usuarios principal del sistema, con un conjunto de funcionalidades enfocadas en la gestión clínica diaria. A través de la plataforma, pueden registrar y consultar pacientes, visualizar sus operaciones, crear nuevas operaciones asociadas a vías clínicas predefinidas y registrar el progreso de las tareas asociadas a cada tratamiento. Además, cuentan con herramientas para editar datos clínicos, buscar pacientes por número de historia, y completar la documentación de seguimiento.

Estas funcionalidades han sido diseñadas para cubrir los procesos habituales que se realizan actualmente de forma manual, optimizando la trazabilidad, reduciendo errores y facilitando un seguimiento estructurado. El detalle completo de los requisitos funcionales asociados a este perfil se encuentra en el Anexo A.

2.4. Requisitos funcionales del administrador-médico

El rol del administrador-médico incorpora funciones clave de gestión dentro del sistema, incluyendo la administración de usuarios, la configuración de vías clínicas y la supervisión de los datos generados. En un primer momento, no se contemplaba esta diferenciación de perfiles y todos los médicos iban a tener acceso completo a las funcionalidades de edición y configuración. Sin embargo, en una de las primeras reuniones con los profesionales clínicos, propuse y recomendé establecer esta separación de roles para mejorar la seguridad del sistema, preservar la integridad de las vías clínicas definidas y evitar modificaciones accidentales por parte de usuarios sin responsabilidades administrativas. A partir de esa decisión, se definieron requisitos específicos para el perfil de administrador-médico, que permiten controlar de forma centralizada tanto la creación y edición de vías clínicas como la asignación de tareas y el análisis de los modelos generados.

El perfil administrador-médico dispone de funcionalidades avanzadas que permiten la creación, edición y validación de vías clínicas. Desde la interfaz, estos usuarios pueden definir nuevas vías a partir de ficheros CSV, ejecutar el algoritmo de minería de procesos, visualizar el grafo que define la red de Petri generada y realizar su validación estructural. Una vez validadas, pueden asignar a las distintas fases las tareas y sus tipos de respuesta. Así como modificar vías ya registradas o etiquetar casos clínicos según su estructura.

Estas funcionalidades están orientadas a facilitar la generación estructurada de modelos clínicos reutilizables y adaptados a distintos contextos hospitalarios. Los requisitos funcionales específicos de este perfil se detallan en el Anexo A.

2.5. Requisitos no funcionales

Además de las funcionalidades orientadas al usuario, se han definido varios requisitos no funcionales que garantizan la usabilidad, interoperabilidad y calidad técnica del sistema. Por ejemplo, los datos exportados están disponibles en formato CSV para su integración con herramientas externas, y toda la interfaz se encuentra en español clínico, facilitando su uso diario en un entorno hospitalario real.

Para más información, puede consultarse el listado completo de requisitos no funcionales en el Anexo A.

2.6. Flujos de trabajo del administrador-médico

En esta sección se presentan dos diagramas de actividades que ilustran los procesos principales del sistema desde la perspectiva del administrador-médico.

El diagrama de la Figura 1 describe el flujo de trabajo que sigue el administrador-médico al registrar una nueva vía clínica en el sistema. El proceso se inicia con la introducción de los datos necesarios, tras lo cual se ejecuta un algoritmo que genera automáticamente una red de Petri como

representación estructurada del flujo clínico. Esta red es visualizada por el administrador en forma de grafo para simplificar su lectura, quien evalúa si es satisfactoria. En caso de serlo, se procede con la asignación o creación de tareas para cada fase, la definición del tipo de respuesta requerida, la configuración de etiquetas y, finalmente, el almacenamiento de la nueva vía. Si la red no cumple con los criterios esperados, el administrador puede modificar los datos y reiniciar el proceso de generación.

El diagrama de la Figura 2 representa las acciones disponibles para el administrador-médico al editar una vía clínica previamente registrada. El proceso comienza con la selección de la vía a modificar, tras lo cual se muestra su contenido actual. El administrador puede optar por editar el nombre, modificar tareas existentes o generar una nueva red de Petri basada en los datos actuales. Si se genera una nueva red, esta es evaluada visualmente. Si resulta satisfactoria, se asignan o ajustan tareas, se define el tipo de respuesta y se editan las etiquetas. Finalmente, la vía modificada se guarda, concluyendo el proceso de edición.

2.7. Resumen de la especificación funcional

La especificación funcional presentada en esta sección constituye la base estructural del sistema propuesto, integrando de manera coherente las necesidades operativas identificadas durante el análisis de requisitos. A través del diccionario de términos, se ha establecido un lenguaje común que permite evitar ambigüedades y asegurar una comunicación efectiva entre los desarrolladores y los profesionales sanitarios. La identificación y caracterización de los distintos *stakeholders* ha facilitado una comprensión profunda del entorno de uso, permitiendo definir funcionalidades específicas para cada perfil, con especial atención a los médicos y al administrador-médico.

Los requisitos funcionales reflejan las capacidades mínimas que el sistema debe ofrecer para satisfacer las demandas del entorno clínico en términos de gestión de pacientes, operaciones y vías clínicas. Del mismo modo, los requisitos no funcionales garantizan aspectos clave de calidad como la interoperabilidad, el uso adecuado del idioma y la estructuración de los datos exportados. Finalmente, los flujos de trabajo modelados mediante diagramas de actividad proporcionan una representación clara y detallada de los procesos más relevantes desde la perspectiva del administrador-médico, sirviendo como guía tanto para la implementación técnica como para la formación de los futuros usuarios del sistema.

En conjunto, esta especificación funcional ofrece un marco sólido y completo para el diseño, desarrollo y posterior validación de la solución, asegurando que esta responda adecuadamente a las necesidades reales del contexto clínico y se integre de forma efectiva en la dinámica del entorno hospitalario.

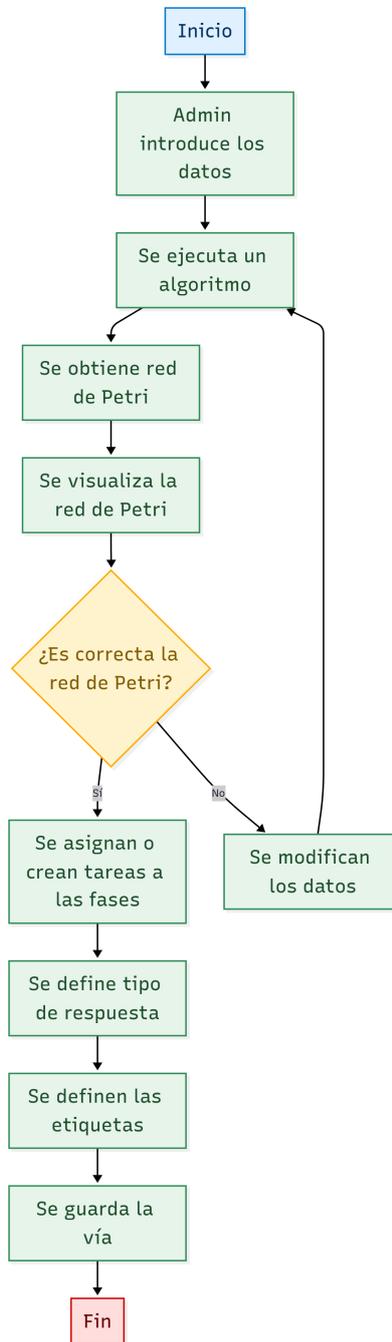


Figura 1: Proceso de creación de una nueva vía clínica

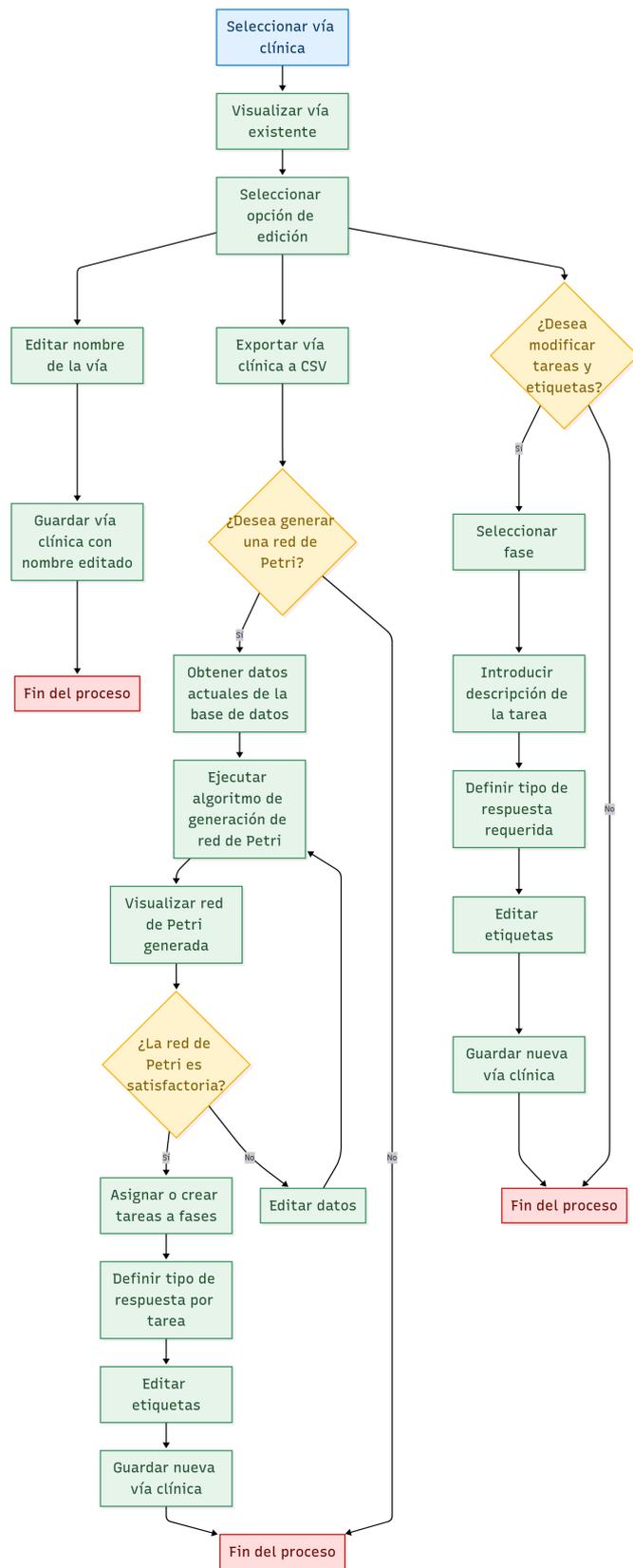


Figura 2: Proceso de edición de una vía clínica existente

3. Prototipo de la interfaz

Una vez definida la especificación funcional del sistema, se procedió al diseño de un primer prototipo de la interfaz de usuario con el objetivo de validar de forma temprana la organización de las pantallas, los flujos de interacción y la usabilidad general del sistema. Para ello, se empleó la herramienta *Figma*², ampliamente utilizada en el ámbito del diseño de interfaces por su versatilidad, colaboración en línea y capacidad para crear prototipos interactivos de alta fidelidad sin necesidad de programación.

Este prototipo sirvió como base para establecer una visión compartida con los expertos clínicos, permitiendo identificar incoherencias, proponer mejoras y asegurar que el diseño respondiera adecuadamente a las necesidades funcionales del proyecto. Como se describe en la sección de metodología (véase la Sección 6), esta fase de validación visual permitió reducir la incertidumbre en etapas posteriores del desarrollo y alinear la experiencia de usuario con los objetivos clínicos y técnicos definidos.

Además, el prototipo ayudó a definir la estructura visual de la aplicación y guió el proceso de implementación, minimizando ambigüedades en los requisitos y garantizando una coherencia estética en toda la interfaz. Cada pantalla fue diseñada siguiendo principios de simplicidad, claridad y jerarquía visual, de modo que el usuario pudiera localizar rápidamente las acciones más relevantes y comprender el flujo de trabajo sin necesidad de formación técnica. Asimismo, se reutilizaron componentes y estilos para reforzar la uniformidad de la experiencia [1].

El prototipo fue presentado al personal médico del HCU “Lozano Blesa”, que participó activamente en la definición de los requisitos del sistema. Durante las sesiones de revisión se recogió *feedback* valioso sobre la disposición de elementos, la comprensión de los formularios y la lógica de navegación. Algunas de estas observaciones llevaron a rediseñar ciertas pantallas antes del desarrollo final, asegurando así una mejor adecuación a las necesidades reales del entorno clínico.

Las Figuras 3 y 4 presentan, a modo de ejemplo, dos de las pantallas más representativas del prototipo desarrollado, ya que ilustran las funcionalidades esenciales del sistema: la creación de una nueva vía y el registro de las tareas asociadas a una operación. Cada una se acompaña de una breve descripción de su funcionalidad. El conjunto completo de pantallas de la aplicación puede consultarse en el Anexo B.

La Figura 3 muestra la interfaz de creación de una nueva vía clínica tras la validación de una red de Petri. En ella, el usuario puede asignar tareas a distintas fases, especificar su tipo (inicial, intermedia o final) y revisar el diseño general. Se priorizó una representación gráfica clara, editable y de uso intuitivo para facilitar su adopción por parte del personal clínico.

La Figura 4 representa la vista destinada al registro de tareas realizadas durante una operación asociada a una vía clínica. Esta pantalla permite actualizar en tiempo real el estado del paciente, comparando el progreso con la vía predefinida y ofreciendo una visión clara del cumplimiento de cada fase.

En conjunto, este prototipo tuvo un papel clave durante la fase de desarrollo, ya que sirvió como guía visual para los programadores y facilitó la toma de decisiones relacionadas con el diseño y la navegación. Su validación se realizó de forma cualitativa en una reunión con personal clínico, donde se presentó el funcionamiento de la aplicación y se recogieron sus opiniones, sugerencias y comentarios. Esta retroalimentación permitió alinear mejor el diseño con sus prácticas habituales

²Figma es un editor de gráficos vectorial y una herramienta de generación de prototipos, <https://www.figma.com>

y necesidades reales, reduciendo errores de interpretación y posibilitando ajustes antes de incurrir en costes de implementación. Como resultado, muchas de las estructuras y flujos definidos en esta etapa se mantuvieron en la versión final del producto.

Figura 3: Prototipo de la pantalla para la creación de una vía clínica

Figura 4: Prototipo de la pantalla de registro de las tareas de una operación

4. Arquitectura del sistema

La arquitectura del sistema intenta reflejar un diseño moderno orientado a la modularidad, la interoperabilidad y la evolución continua [7]. Se ha buscado construir una solución capaz de adaptarse a distintos contextos tecnológicos y de integrarse fácilmente con herramientas externas, sin comprometer la cohesión interna ni la calidad del software. Esta aproximación facilita tanto el desarrollo incremental como el mantenimiento a largo plazo, al tiempo que permite distribuir la carga computacional y organizativa entre distintos componentes especializados.

El sistema se ha diseñado siguiendo un enfoque basado cliente-servidor. A su vez, el servidor se ha implementado siguiendo una arquitectura de microservicios [15], lo que permite que cada componente cumpla una función concreta y se desarrolle de forma independiente. En su conjunto, la solución se compone de cinco elementos principales interconectados. La Figura 5 ilustra la arquitectura lógica del sistema, mostrando los principales módulos y los flujos de información entre ellos.

Por un lado, se encuentra el cliente web o *frontend* que actúa como interfaz de usuario y punto de acceso a la aplicación. En el núcleo de la lógica del sistema se sitúa el servidor *backend* que gestiona la lógica de negocio, la autenticación y coordina la comunicación entre el cliente, la base de datos y los servicios auxiliares. Uno de estos servicios es el *microservicio generador*, responsable de la creación de las vías clínicas. De forma complementaria, el *microservicio analizador* se encarga del análisis de las redes de Petri generadas, evaluando la presencia de bloqueos. Finalmente, una base de datos relacional gestiona el almacenamiento persistente de toda la información relevante del sistema, garantizando integridad y rendimiento.

La comunicación entre los distintos componentes se realiza principalmente a través del protocolo HTTPS, utilizando API REST para garantizar una integración desacoplada y segura. La conexión con la base de datos se establece mediante consultas SQL [11] desde el *backend*.

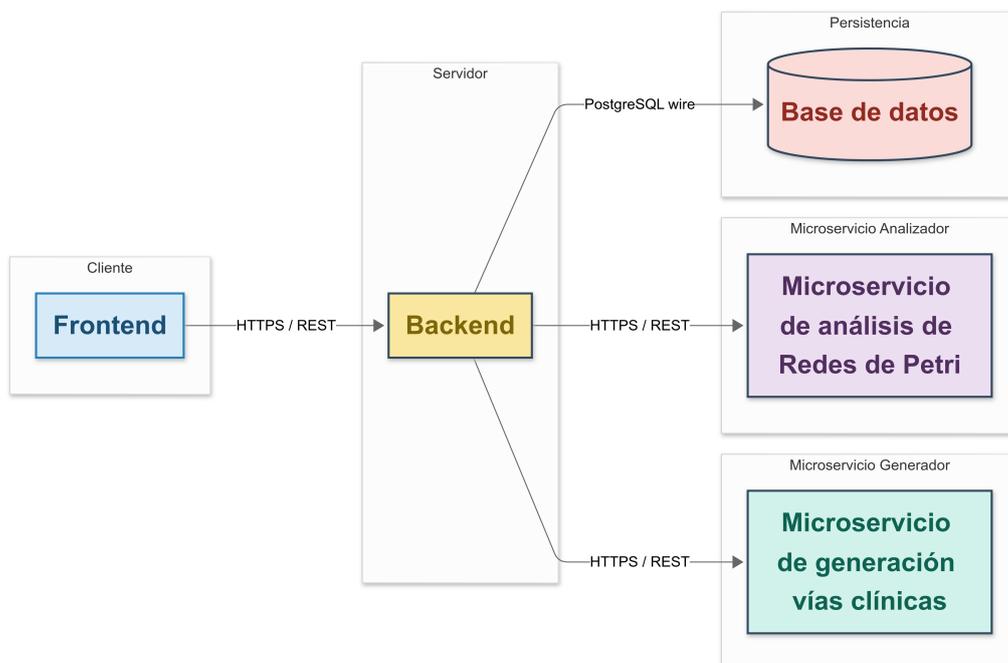


Figura 5: Arquitectura del sistema

4.1. Descomposición en microservicios

Una de las decisiones arquitectónicas clave fue optar por una arquitectura basada en microservicios, lo que permitió dividir el sistema en componentes especializados, cada uno encargado de una funcionalidad concreta. Esta estrategia mejora la mantenibilidad, facilita el escalado individual de servicios y permite una evolución independiente de cada módulo, reduciendo el acoplamiento entre partes del sistema.

4.1.1. Diseño del microservicio generador

Este componente tiene como responsabilidad central la generación de redes de Petri a partir de los datos introducidos por el usuario, reflejando la estructura de una posible vía clínica. El diseño se centra en tres aspectos principales: modularidad del núcleo algorítmico, robustez en la validación de datos de entrada y eficiencia en la respuesta.

El microservicio está estructurado como un único controlador que reciben peticiones, valida el contenido semántico y estructural de los datos, y canaliza la ejecución del algoritmo generador.

Además, si en futuras interacciones se contemplará mejorar o modificar el algoritmo, esta arquitectura basada en microservicios permite incluir múltiples estrategias generadoras bajo una misma interfaz común. Esta previsión amplía las capacidades del sistema y refuerza su escalabilidad funcional.

4.1.2. Diseño del microservicio analizador

El segundo microservicio se encarga del análisis formal de las redes generadas, validando propiedades fundamentales como la vivacidad o la ausencia de bloqueos. Para ello, actúa como puente entre el sistema y herramientas externas de análisis.

Su diseño contempla una capa de abstracción que encapsula la interacción con las herramientas de análisis, permitiendo su sustitución o extensión sin modificar la lógica principal. Esta separación también facilita la gestión de errores.

La arquitectura modular del microservicio permite incorporar nuevos tipos de análisis o motores de evaluación en el futuro. Así, se garantiza la evolución de las capacidades analíticas sin comprometer la estabilidad del sistema existente.

4.1.3. Integración e interoperabilidad

Ambos microservicios se integran con el backend mediante llamadas HTTP asincrónicas. Los datos se transmiten en formato JSON, lo que asegura la interoperabilidad entre componentes desarrollados con distintos lenguajes y tecnologías. Esta estrategia refuerza la independencia tecnológica de cada servicio, a la vez que facilita el testeado individual y la monitorización.

El diseño orientado a servicios con responsabilidades bien delimitadas permite distribuir la carga computacional de forma eficiente, gestionar errores de manera localizada y escalar de forma selectiva según las necesidades del sistema.

4.2. Tecnologías utilizadas

El sistema se ha construido empleando tecnologías modernas, ampliamente utilizadas en la industria y seleccionadas no solo por su madurez, sino también por su adecuación a los requerimientos funcionales de cada componente, la experiencia previa del equipo y la facilidad de integración entre ellas. A continuación, se detallan las tecnologías elegidas y el motivo de su selección frente a otras alternativas:

- **Frontend:** se ha utilizado React [21], acompañado de Tailwind CSS [22] para la maquetación y el diseño de interfaces. React fue elegido por su enfoque basado en componentes reutilizables, su gran rendimiento en aplicaciones dinámicas y la existencia de una amplia comunidad que proporciona bibliotecas y soluciones para una gran variedad de casos de uso. En comparación con otras opciones como Angular o Vue, React ofrecía una curva de aprendizaje más favorable dada mi experiencia previa, además de una mayor flexibilidad para integrarse con herramientas de diseño modernas como Tailwind CSS.
- **Backend:** implementado con Node.js [16] y el framework Express [10], permitiendo desarrollar una API rápida y escalable utilizando JavaScript, el mismo lenguaje que en el *frontend*. Esta elección favorece la consistencia tecnológica a lo largo del *stack* y reduce el coste de aprendizaje. Frente a otras opciones como Django (Python) o Spring Boot (Java), Node.js ofrece mayor ligereza, rapidez de desarrollo y una comunidad muy activa para proyectos web modernos.
- **Microservicio generador:** desarrollado en Python [19] utilizando el framework FastAPI [20]. La elección de Python responde, en primer lugar, a una cuestión de deuda técnica: el algoritmo ya se encontraba previamente implementado en este lenguaje como parte de un trabajo anterior, y reescribirlo en otro lenguaje como JavaScript habría supuesto un esfuerzo considerable, sin aportar mejoras funcionales significativas. Dado que el objetivo de este proyecto no era optimizar dicho algoritmo, sino integrarlo eficazmente dentro de una arquitectura moderna, se optó por conservar su implementación original para reducir los costes de desarrollo y enfocar el trabajo en los objetivos clave del sistema. FastAPI fue seleccionado por ser un framework moderno, eficiente y altamente compatible con los estándares actuales de desarrollo de APIs REST. Su sintaxis declarativa, su soporte nativo para OpenAPI y su rapidez de implementación lo convierten en una opción ideal para construir servicios web ligeros y bien documentados en poco tiempo.
- **Microservicio analizador:** también desarrollado en Python con FastAPI, este componente se encarga de analizar las redes de Petri generadas previamente, evaluando la presencia de bloqueos y otras propiedades mediante la herramienta externa LoLA [25], instalada en un entorno Ubuntu. Python fue la opción más adecuada para esta tarea debido a su excelente capacidad de integración con herramientas del sistema operativo, scripts de automatización y utilidades de línea de comandos, lo cual facilita la ejecución controlada de procesos externos. Además, el uso compartido de Python entre los distintos microservicios analíticos permite mantener la coherencia tecnológica, favoreciendo el mantenimiento y reduciendo la complejidad del sistema. Al igual que en el microservicio generador, FastAPI proporciona una base eficiente y ligera para exponer estas funcionalidades como servicios web, accesibles desde el backend principal mediante llamadas HTTP estructuradas y seguras.
- **Base de datos:** se ha optado por PostgreSQL [17], un sistema de gestión de bases de datos relacional de código abierto que destaca por su fiabilidad, soporte a transacciones ACID

y potentes capacidades de consulta. En comparación con otras opciones como MySQL o MongoDB, PostgreSQL ofrecía un equilibrio óptimo entre robustez, rendimiento y soporte para estructuras de datos complejas, siendo especialmente adecuado para la integridad de los datos clínicos manejados.

En conjunto, esta arquitectura orientada a servicios permite una evolución progresiva del sistema, facilitando la incorporación de nuevas funcionalidades sin impactar en el núcleo existente.

5. Modelo de datos

El modelo de datos constituye un componente esencial del sistema, ya que define la estructura lógica de la información, las entidades clave y las relaciones entre ellas. Un diseño adecuado garantiza la integridad, coherencia y eficiencia en el acceso a los datos, a la vez que facilita el mantenimiento y evolución del sistema. A lo largo del desarrollo del proyecto, este modelo ha sido refinado iterativamente para adaptarse a los requisitos funcionales detectados.

5.1. Modelo Entidad-Relación

El diagrama Entidad-Relación (ER), que aparece en la Figura 6 ilustra de forma conceptual las entidades principales del sistema y las asociaciones existentes entre ellas. Este modelo permite comprender de manera clara cómo se organiza la información y cómo interactúan los distintos componentes de la base de datos.

Las entidades más relevantes del sistema:

- **Médico:** contiene atributos como **usuario**, **contraseña**, **nombre** y **esAdmin**, e interactúa con múltiples pacientes.
- **Paciente:** identificado por el atributo **numHist**, dispone de campos como **nombre** y **sexo**, y puede ser tratado por uno o varios médicos.
- **Operación:** representa un procedimiento clínico realizado a un paciente, registrando datos como **fecha**, **edad**, **índice de masa corporal** y sus etiquetas asociadas. Cada operación está vinculada a una única vía clínica.
- **Vía clínica:** estructura el tratamiento del paciente en fases sucesivas. Cada vía contiene varias fases y está asociada a una o más operaciones.
- **Fase:** unidad estructural dentro de una vía clínica. Cada fase puede estar compuesta por múltiples tareas.
- **Tarea:** unidad de acción o evaluación clínica, definida por atributos como **descripción** y **tipo**. Puede incluir una o varias opciones como posibles respuestas o acciones asociadas.
- **Opción:** elemento que forma parte de una tarea seleccionable, representando alternativas concretas que pueden marcarse como completadas o no.
- **Etiqueta:** permite clasificar operaciones con criterios clínicos, analíticos u organizativos. Su asociación con las operaciones facilita la posterior explotación de los datos.

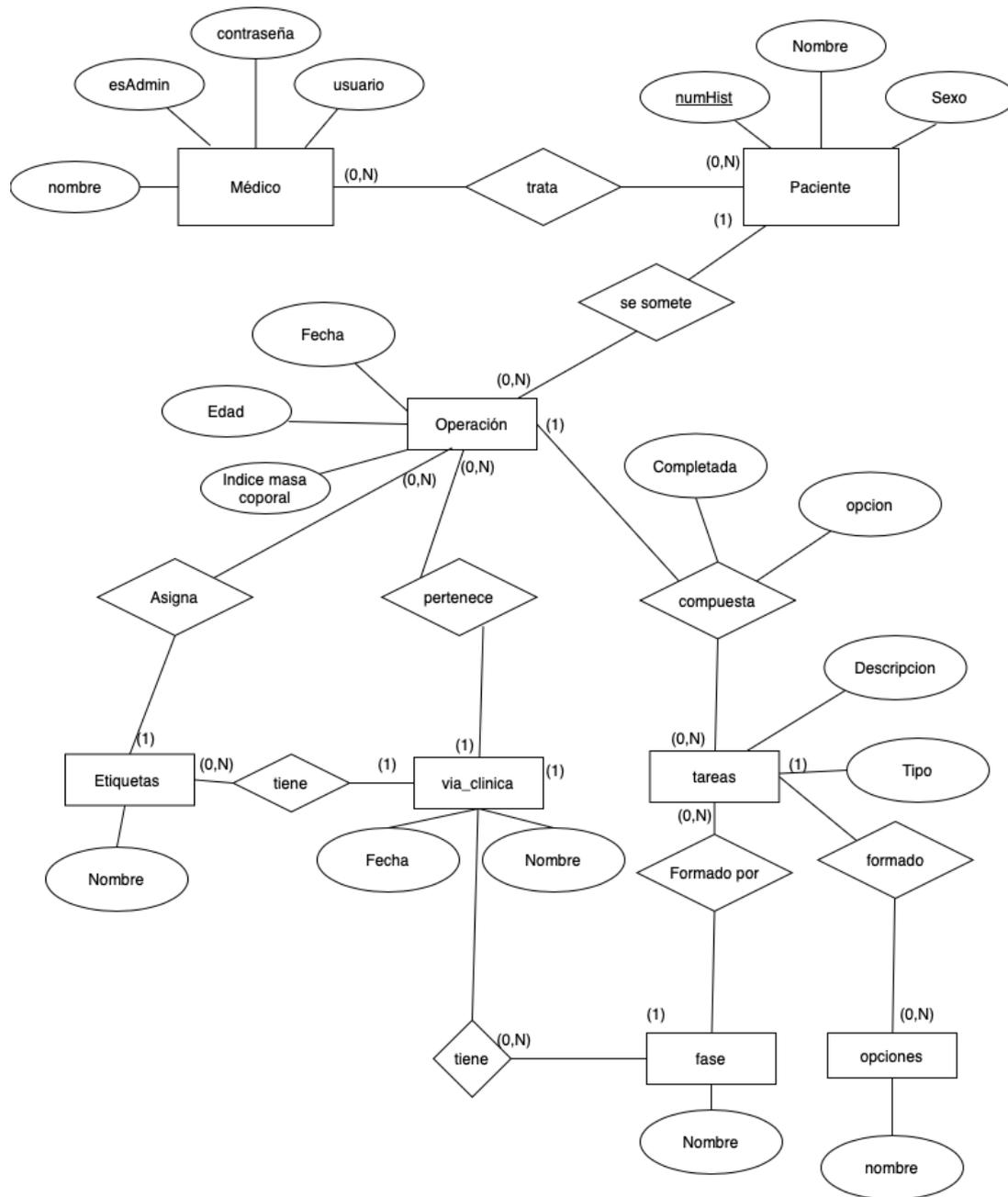


Figura 6: Modelo Entidad-Relación del sistema

Este modelo ofrece una visión estructurada del flujo clínico y de la lógica del sistema, permitiendo representar la trazabilidad de las acciones realizadas y la organización jerárquica de las vías clínicas y sus componentes.

5.2. Modelo relacional de la base de datos

A partir del modelo conceptual, se ha generado el correspondiente modelo relacional, el cual traduce las entidades y relaciones en tablas, claves primarias y claves foráneas. Este modelo es el que se implementa directamente en el sistema de gestión de bases de datos (PostgreSQL en este caso), y su correcta normalización garantiza una gestión eficiente y segura de los datos. El mismo

se muestra en la Figura 7.

El diseño ha sido documentado y visualizado utilizando el lenguaje Database Markup Language (DBML) [9] a través de la herramienta *dbdiagram.io*³, lo que ha facilitado tanto su validación como su integración con herramientas de desarrollo. En este modelo relacional:

- Las entidades conceptuales se corresponden con tablas: *Medico*, *Paciente*, *Operacion*, *Via*, *Fase*, *Tarea*, *Opcion_tarea*, *Etiqueta*, etc.
- Las relaciones N:M se han descompuesto en tablas intermedias como *Medico_de_paciente*, que asocia médicos con pacientes tratados.
- Se han definido claves primarias para identificar de forma única cada fila, y claves foráneas para mantener la integridad referencial entre las entidades.
- Atributos adicionales como *completada* o *valor* permiten registrar el estado de las tareas dentro de una operación concreta.

El resultado es una base de datos relacional normalizada, preparada para soportar eficientemente las operaciones del sistema, consultas clínicas y futuras ampliaciones.

³Herramienta para dibujar diagramas relacionales, <https://dbdiagram.io/home>

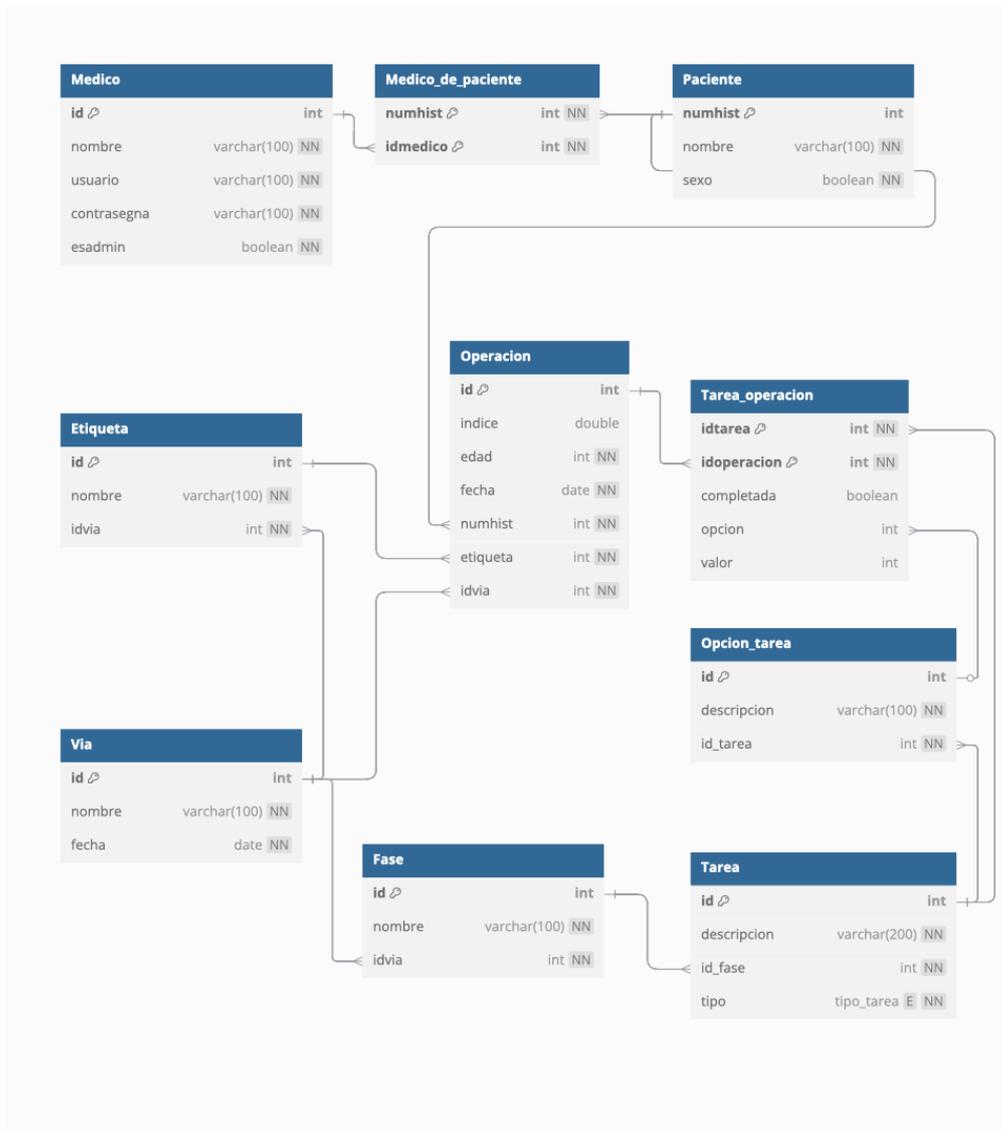


Figura 7: Modelo relacional de la base de datos

6. Metodología de desarrollo

A lo largo de estos cuatro años nos han inculcado la importancia de aplicar metodologías adecuadas para planificar, desarrollar y evaluar proyectos de forma rigurosa y eficiente. En este trabajo, he procurado trasladar dichos principios a la práctica, adoptando un enfoque sistemático que ha guiado las distintas etapas del desarrollo. Esta experiencia ha permitido consolidar los conocimientos adquiridos y comprobar su utilidad en un entorno real de diseño y construcción de software.

El proyecto se ha dividido en varias fases bien diferenciadas: análisis de requisitos, diseño de la arquitectura, implementación, pruebas e integración.

6.1. Análisis de requisitos

La fase de análisis de requisitos se estructuró en torno a una serie de reuniones con los profesionales médicos, quienes actuaron como clientes y usuarios finales del sistema. En una primera reunión se llevó a cabo una toma de contacto en la que me expusieron el funcionamiento del sistema actual, las dinámicas habituales de trabajo y las principales necesidades que debían ser abordadas. Este primer intercambio permitió establecer un marco de comprensión compartido y detectar las limitaciones del proceso existente que el nuevo sistema debía resolver.

Durante una segunda reunión, se presentó una propuesta preliminar de requisitos, aún sin formalizar, con el objetivo de contrastar ideas y orientar adecuadamente la dirección del proyecto. Gracias a la retroalimentación obtenida en esta sesión, se refinaron varias funcionalidades clave y surgieron nuevos aspectos que no se habían considerado en el encuentro inicial, como la necesidad de incorporar un sistema de autenticación de usuarios.

También se planteó una cuestión relevante relacionada con los pacientes: ¿debían estar visibles para todos los médicos o solo para quienes los atendieran? El equipo médico explicó que cada profesional trabaja siempre con sus propios pacientes, siendo el único responsable de su atención. Por tanto, solicitaron que el buscador de pacientes mostrara únicamente aquellos asignados a cada médico.

En esta sesión de trabajo, se reconoció la posibilidad de que existieran excepciones, como cambios de médico o situaciones especiales. Ante esto, se propuso incorporar una funcionalidad que permitiera al administrador asignar pacientes a diferentes médicos, garantizando así la flexibilidad del sistema ante cualquier eventualidad.

La tercera reunión tuvo como objetivo validar tanto la versión formalizada de los requisitos como un prototipo inicial de la interfaz de usuario. En esta sesión se recogió nuevo *feedback*, como la necesidad de condicionar el avance entre fases del proceso clínico a la finalización previa de todas las tareas de la fase actual y la necesidad de crear etiquetas para cada una de las vías, lo cual les ayuda a ellos en su día a día. Este tipo de aportaciones resultaron fundamentales para consolidar la lógica de navegación del sistema y garantizar una experiencia coherente con los flujos de trabajo reales del entorno médico.

6.2. Diseño del sistema

El proceso de diseño del sistema no fue inmediato, sino el resultado de una evolución progresiva en función de las necesidades técnicas que se iban identificando y de las lecciones aprendidas durante el análisis de requisitos. Inicialmente, se planteó una arquitectura monolítica, dada su simplicidad, la facilidad de implementación y el tamaño relativamente acotado del sistema. Esta opción permitía un desarrollo más directo, especialmente en fases tempranas del proyecto, cuando la complejidad parecía manejable desde un único núcleo de control.

Sin embargo, pronto surgió un primer punto de inflexión: el algoritmo de generación de vías clínicas, que estaba desarrollado en Python, mientras que el servidor principal se planteaba en JavaScript (Node.js). Esta diferencia de tecnologías abrió dos caminos posibles: adaptar el algoritmo al entorno del servidor, o mantenerlo en su lenguaje original y desacoplarlo funcionalmente. Se optó por esta última alternativa, ya que preservaba la fiabilidad del algoritmo y permitía una integración más limpia. Esto supuso una transición hacia una arquitectura cliente-servidor, en la que el *backend* comenzaba a asumir una estructura orientada a microservicios [15].

Posteriormente, se identificó un nuevo requisito funcional: la validación formal de las redes de Petri [14] generadas. Esta validación permitiría asegurar propiedades clave, como la ausencia de bloqueos, antes de que el médico las revisara, reduciendo su carga de trabajo y mejorando la fiabilidad del sistema. Para ello, era necesario incorporar una herramienta externa especializada en este tipo de análisis. La mejor forma de integrar esta funcionalidad sin introducir dependencias tecnológicas fuertes fue encapsularla como un microservicio independiente, accesible a través de una API.

Este conjunto de decisiones llevó a consolidar un diseño basado en microservicios para aquellas funcionalidades que requerían autonomía tecnológica o que podían evolucionar de forma separada. De este modo, se logró desacoplar el núcleo del sistema de componentes externos y se garantizó que futuras modificaciones (como el reemplazo de una herramienta de análisis) no impactaran directamente sobre el resto de la arquitectura. El resultado final es un sistema modular, flexible y preparado para evolucionar sin comprometer su cohesión interna.

6.3. Implementación del sistema

El proceso de implementación se abordó de forma incremental, priorizando inicialmente los requisitos más próximos a la interacción directa con los profesionales médicos. De este modo, los primeros componentes desarrollados fueron aquellos relacionados con el registro de tareas en una operación quirúrgica y el sistema de autenticación. Esta funcionalidad se implementó tanto en el cliente web como en el servidor, así como en la base de datos, donde se gestionan las credenciales y la persistencia de los registros clínicos.

Dado que los médicos proporcionaron dos ejemplos concretos de vías clínicas, artroplastia de cadera y artroplastia de rodilla, la parte correspondiente del *frontend* se implementó inicialmente de forma estática (*hardcoded*), representando visualmente los pasos definidos para cada caso. Una vez completada esta primera versión, se grabó y envió a los médicos una demostración en vídeo, permitiéndoles validar la funcionalidad y confirmar que el sistema seguía alineado con sus necesidades y expectativas.

Con su visto bueno, se abordó la segunda gran fase: la implementación de los requisitos relacionados con la creación automática de vías clínicas. Para ello, el primer paso fue comprender en

profundidad el funcionamiento del algoritmo ya existente, desarrollado en Python, y qué tipo de salida producía. Inicialmente se trabajó con una versión *mockeada* de esa salida, lo que permitió avanzar en paralelo en el desarrollo del *frontend*, el *backend* y el microservicio generador sin depender todavía de la ejecución real del algoritmo. Esto facilitó la integración inicial, simulando la respuesta del sistema como si fuera completamente funcional.

Una vez establecida esta estructura, se adaptó el algoritmo para que aceptara cualquier fichero CSV como entrada y generara únicamente la información relevante para el sistema. Esta versión ya operativa se integró completamente en el microservicio generador, expuesto mediante FastAPI.

La última etapa de la implementación se centró en la validación formal de las redes de Petri generadas. Este proceso presentó numerosos retos. Inicialmente se valoró el uso de herramientas como Renew [24] o TimeNET [23], que operaban en entornos Windows y se ejecutaban mediante conexión remota (SSH). Sin embargo, su integración resultó compleja y poco robusta, por lo que se exploraron alternativas más adecuadas. Finalmente, se optó por utilizar LoLA, una herramienta ligera y ejecutable desde terminal en sistemas Linux, capaz de analizar propiedades fundamentales de las redes de Petri, como la existencia de bloqueos.

Este microservicio analizador se implementó como un componente independiente, capaz de comunicarse con el servidor *backend* mediante una API REST basada en FastAPI. Paralelamente, se modificó el microservicio generador para que su salida incluyera la red de Petri en un formato compatible con LoLA. También fue necesario realizar ajustes en el algoritmo para asegurar que las redes de Petri generadas fueran *vivas*, es decir, que desde cualquier estado alcanzable fuera siempre posible activar alguna transición. Esta propiedad es clave para garantizar que el modelo representado por la red no llegue a un estado de inactividad irreversible, lo que se traduciría en un fallo de diseño desde el punto de vista clínico.

En conjunto, la implementación avanzó en iteraciones coordinadas y validadas por los usuarios finales, permitiendo construir una solución funcional, robusta y adaptada a los requisitos reales del entorno médico.

6.4. Validación

Una vez implementadas todas las funcionalidades principales del sistema, se procedió a su validación técnica mediante distintas estrategias. En primer lugar, se realizaron pruebas de integración de los distintos microservicios y de los puntos de comunicación entre cliente y servidor. Para ello, se utilizó la herramienta Postman [18], que permitió enviar peticiones personalizadas al API REST del sistema y verificar que las respuestas fueran correctas en diferentes escenarios de entrada, tanto válidos como erróneos.

En paralelo, se llevaron a cabo pruebas manuales sobre el cliente web para comprobar el correcto funcionamiento de la interfaz, el flujo de navegación, y la ausencia de errores visibles en la interacción con el usuario. Esta fase permitió detectar pequeños fallos visuales o de lógica que fueron corregidos antes de considerar el sistema estable.

Tal y como se detalla en la sección 8.1, la validación con usuarios finales (médicos del HCU) constituye el siguiente hito importante. Su participación será clave para evaluar no solo la funcionalidad, sino también la usabilidad, claridad y alineación del sistema con sus flujos de trabajo reales.

Aunque las pruebas actuales aseguran un comportamiento correcto desde el punto de vista

técnico, aún no se ha completado una evaluación formal en el entorno clínico. Por tanto, el sistema se encuentra en un estado funcionalmente sólido pero en fase previa a su despliegue real. La validación con usuarios permitirá detectar posibles limitaciones no evidentes durante el desarrollo y servirá como base para futuras mejoras y ajustes.

7. Producto final

El resultado del desarrollo llevado a cabo es una aplicación web funcional diseñada para apoyar la gestión y seguimiento de vías clínicas personalizadas dentro del entorno hospitalario. Este producto representa la convergencia entre los objetivos clínicos planteados inicialmente y las soluciones técnicas implementadas para darles respuesta.

La herramienta ofrece funcionalidades adaptadas a dos perfiles de usuario diferenciados: médicos y administradores-médico. Esta separación, introducida tras las reuniones con los profesionales del ámbito clínico, ha resultado esencial para garantizar la seguridad, el control y la trazabilidad de los procesos definidos en el sistema.

Funcionalidades para médicos

El perfil de médico está diseñado para facilitar la gestión de pacientes, operaciones y el seguimiento clínico de manera eficiente, respetando el flujo de trabajo real en el entorno hospitalario. A continuación, se describen algunas de las principales funcionalidades disponibles para este perfil, acompañadas de capturas de pantalla representativas de la interfaz.

Una de las primeras funcionalidades clave es la búsqueda de pacientes registrados en el sistema mediante su nombre o su número de historia clínica, lo que permite un acceso rápido a la información necesaria.

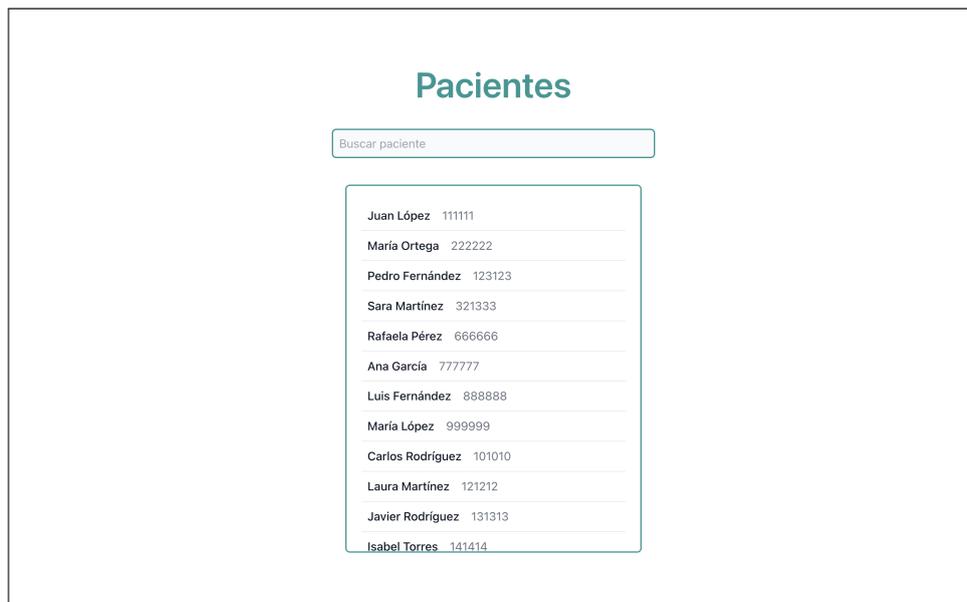


Figura 8: Interfaz de búsqueda de pacientes por número de historia clínica

Una vez localizado el paciente, se accede a una vista detallada de su información clínica, inclu-

yendo los datos personales y sus operaciones.

The screenshot shows a web interface with a home icon in the top left. The main content is divided into two sections: 'Información del Paciente' and 'Operaciones'. The patient information section displays: 'Número de historia: 111111', 'Nombre: Juan López', and 'Sexo: Hombre'. The operations section contains two entries, each with a 'Nueva operación' button to its right. The first entry is for 'Rodilla' (knee) with date '20/6/2025', BMI 'Índice masa corporal: 23', age 'Edad: 43', and label 'Etiqueta: Artroplastia primera izquierda'. The second entry is for 'Cadera' (hip) with the same date, BMI, age, and label 'Etiqueta: Artroplastia primera derecha'.

Figura 9: Visualización de la información de un paciente y su historial de operaciones

Desde esta vista, el médico puede crear una nueva operación para el paciente, seleccionando una de las vías clínicas previamente definidas. Esta acción inicia el seguimiento estructurado del proceso clínico.

The screenshot shows the 'Nueva operación' form. It includes input fields for 'Índice masa corporal' (23), 'Edad' (43), and 'Vía seleccionada' (Cadera). There is a radio button group for 'Etiqueta' with options: 'Artroplastia primera derecha' (selected), 'Artroplastia primera izquierda', 'Artroplastia de recambio derecha', and 'Artroplastia de recambio izquierda'. A 'Buscar vía' section shows a search box and a table with two rows: 'Cadera' with date '20/06/2025' and 'Rodilla' with date '20/06/2025'. A 'Guardar' button is at the bottom.

Figura 10: Creación de una nueva operación asociada a una vía clínica

Una vez creada la operación, el sistema muestra las acciones de la vía clínica correspondiente, permitiendo marcar tareas como completadas, según el progreso del paciente, así como registrar los valores asociados cuando se requiera una respuesta numérica o categórica.

Planificación Ingreso Quirófano Post Cirugía Primer Día Segundo Día Post Operatorio

Planificación Guardar

Número: 11111
 Nombre: Juan López
 Operación: Cadera
 Fecha: 20/6/2025
 Edad: 63
 Índice masa: 23
 Etiqueta: Artroplastia primera derecha

Verificar documentación completa en historia clínica electrónica (HCE)	<input type="radio"/> Sí <input checked="" type="radio"/> No
Consentimiento informado	<input checked="" type="radio"/> Sí <input type="radio"/> No
Planificar fecha de la intervención y revisión por anestesia	<input checked="" type="radio"/> Sí <input type="radio"/> No
Verificar que ha sido valorada por anestesia y posibles incidencias	<input type="radio"/> Sí <input checked="" type="radio"/> No
Verificar medicación habitual del paciente y comorbilidades	<input checked="" type="radio"/> Sí <input type="radio"/> No
Solicitar pruebas complementarias y/o interconsultas interservicio	<input type="radio"/> Sí <input checked="" type="radio"/> No
Evaluar hemograma y pautar si procede de prevención de transfusión	<input checked="" type="radio"/> Sí <input type="radio"/> No
Solicitar material	<input type="radio"/> Sí <input checked="" type="radio"/> No
Solicitud de banco de tejidos si procede	<input checked="" type="radio"/> Sí <input type="radio"/> No
Revisión de consentimiento informado	<input checked="" type="radio"/> Sí <input type="radio"/> No
Actualización de pruebas complementarias si procede: Rx de control	<input type="radio"/> Sí <input checked="" type="radio"/> No

Figura 11: Registro de tareas completadas en la vía clínica de una operación

Aunque en esta sección se han mostrado únicamente algunas de las funcionalidades implementadas, estas representan las de mayor importancia y relevancia. El conjunto completo de funcionalidades desarrolladas cubre de forma exhaustiva todos los requisitos funcionales definidos para el perfil médico, asegurando una digitalización robusta, coherente y plenamente alineada con las necesidades del entorno clínico real.

Funcionalidades para el administrador-médico

El perfil de administrador-médico integra funcionalidades avanzadas orientadas a la creación, validación y actualización estructural de las vías clínicas. A continuación, se ilustran algunas de las funcionalidades clave desarrolladas para este tipo de usuario.

En primer lugar, el sistema ofrece una interfaz que permite buscar y consultar todas las vías clínicas registradas, así como acceder directamente a su edición. Desde esta misma vista, el administrador puede iniciar el proceso de creación de una nueva vía clínica, lo que da lugar a una secuencia estructurada de configuración asistida.

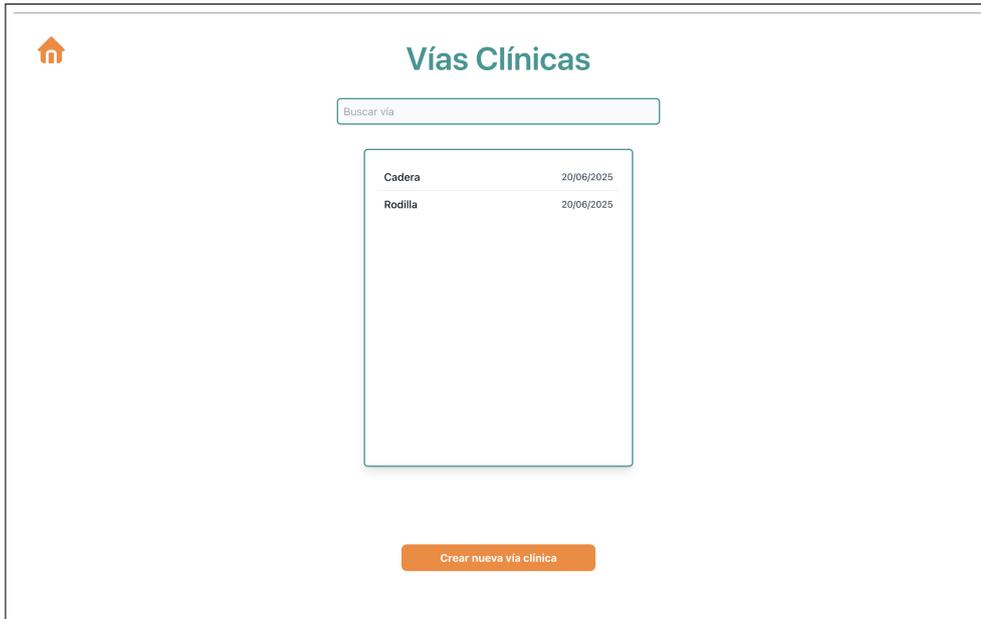


Figura 12: Listado y búsqueda de vías clínicas registradas. Opción de crear una nueva vía.

Una vez introducidos el nombre y el archivo CSV correspondiente, el sistema genera automáticamente una red de Petri y su representación gráfica mediante el algoritmo implementado. Esta red es analizada por el microservicio analizador. En caso de validación positiva, se presenta visualmente al administrador para su revisión estructural.

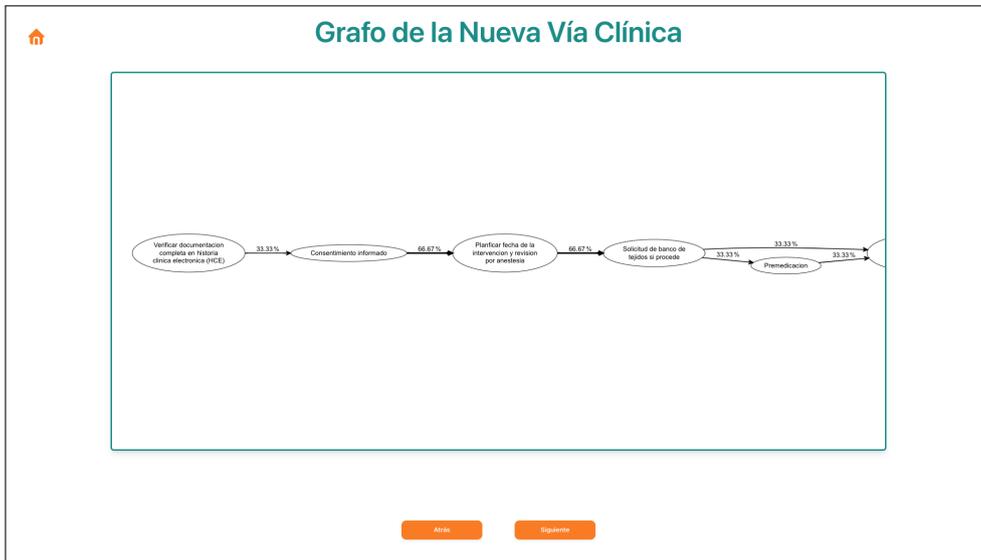


Figura 13: Visualización del grafo generado y la red de Petri resultante para su validación.

A continuación, el sistema solicita la asignación de tareas a cada fase de la vía clínica generada, permitiendo al administrador especificar el tipo de respuesta esperada para cada una: valor numérico, opción múltiple o dicotómica. Este paso es esencial para estructurar el flujo clínico y garantizar la integridad semántica del modelo.

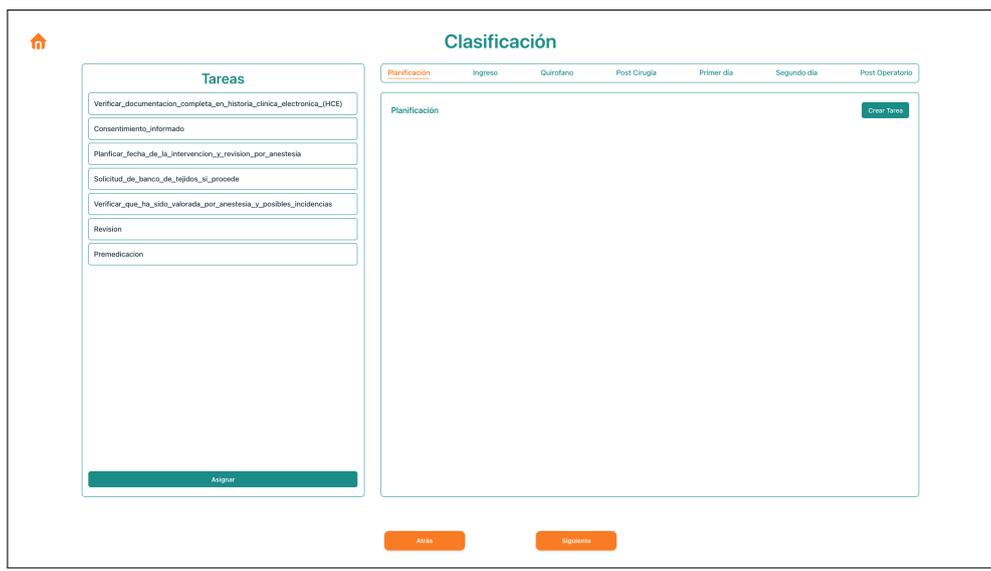


Figura 14: Asignación de tareas por fase y definición del tipo de respuesta asociada.

Por último, el sistema ofrece una vista de comprobación final en la que se resumen todos los datos introducidos: nombre, etiquetas, fases definidas y tareas asignadas, junto con sus tipos. Esta revisión previa permite validar el modelo antes de ser almacenado en la base de datos y puesto en producción.

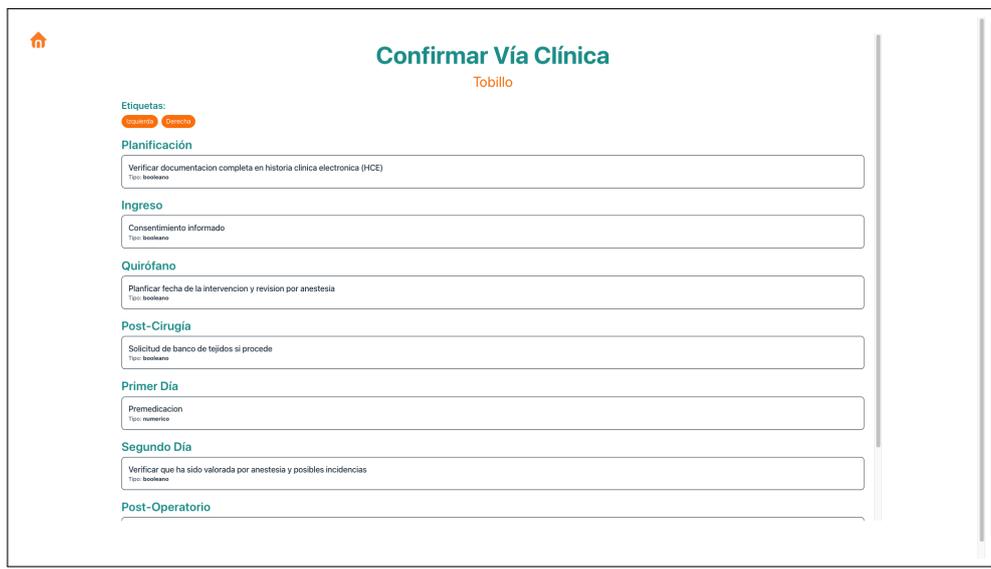


Figura 15: Pantalla de comprobación final de la vía clínica antes de su validación.

Estas funcionalidades reflejan las principales capacidades ofrecidas al perfil administrador-médico. Aunque no se recogen en esta sección de forma exhaustiva, el conjunto total de acciones implementadas cubre íntegramente los requisitos funcionales definidos para este rol, permitiendo un control estructurado, seguro y escalable de las vías clínicas dentro del sistema.

Proceso de creación de una vía clínica: diagrama de secuencia

El proceso de creación y validación de una vía clínica en el sistema implica la interacción entre varios componentes distribuidos: el cliente web, el servidor principal, dos microservicios independientes (uno encargado de generar la red de Petri y otro de analizarla) y la base de datos.

En la Figura 16 se muestra un diagrama de secuencia que resume este flujo de manera estructurada. El proceso comienza cuando el usuario (administrador-médico) introduce el nombre de la nueva vía clínica y sube un archivo CSV que contiene las fases y tareas clínicas. Este archivo es enviado al servidor, que a su vez lo deriva al microservicio encargado de ejecutar el algoritmo de generación de la red.

Este microservicio construye tanto un grafo intermedio como la red de Petri asociada, que posteriormente devuelve al servidor. El servidor envía dicha red al microservicio de análisis, que valida su estructura para detectar posibles bloqueos u otras inconsistencias.

En caso de que la red generada sea correcta, el cliente muestra visualmente la red de Petri para que el usuario pueda continuar con la definición detallada de la vía clínica, incluyendo la asignación de tareas, tipos de respuesta y etiquetas. Finalmente, la vía clínica se guarda en la base de datos.

Si, por el contrario, el análisis detecta bloqueos, el sistema notifica al usuario para que revise y modifique los datos iniciales antes de volver a ejecutar el algoritmo.

Documentación de las APIs

Con el objetivo de garantizar la mantenibilidad, escalabilidad y correcta integración del sistema, se ha generado documentación técnica detallada para todas las interfaces de programación de aplicaciones (APIs) expuestas, tanto por el servidor principal como por los microservicios auxiliares.

La API del servidor web, responsable de gestionar las operaciones principales del sistema (pacientes, operaciones, vías clínicas y autenticación de usuarios), ha sido documentada empleando el estándar *OpenAPI*, permitiendo así una descripción estructurada y navegable de los distintos endpoints. Esta documentación incluye métodos disponibles, parámetros requeridos, tipos de respuesta, códigos de estado y ejemplos de uso, lo cual facilita tanto el desarrollo como futuras tareas de integración o ampliación del sistema. En la figura 17 se muestra la documentación obtenida.

Por otro lado, los microservicios, específicamente, el generador de redes de Petri y el analizador de bloqueos. También exponen sus propias APIs internas. Aunque su uso está restringido al servidor principal y no directamente accesible para clientes externos, estas interfaces han sido igualmente documentadas para facilitar su comprensión, pruebas unitarias y mantenimiento independiente. La documentación de estos microservicios se mantiene en formato *Markdown* [2] y se complementa con ejemplos concretos de peticiones y respuestas.

Toda esta documentación se encuentra centralizada en un repositorio técnico del proyecto, lo que permite una consulta rápida y facilita la futura incorporación de nuevos módulos o integraciones con sistemas hospitalarios existentes.

Modificaciones al algoritmo original

Durante el desarrollo del sistema, fue necesario realizar una serie de adaptaciones al algoritmo encargado de generar la red de Petri a partir de datos clínicos. Estas modificaciones tuvieron

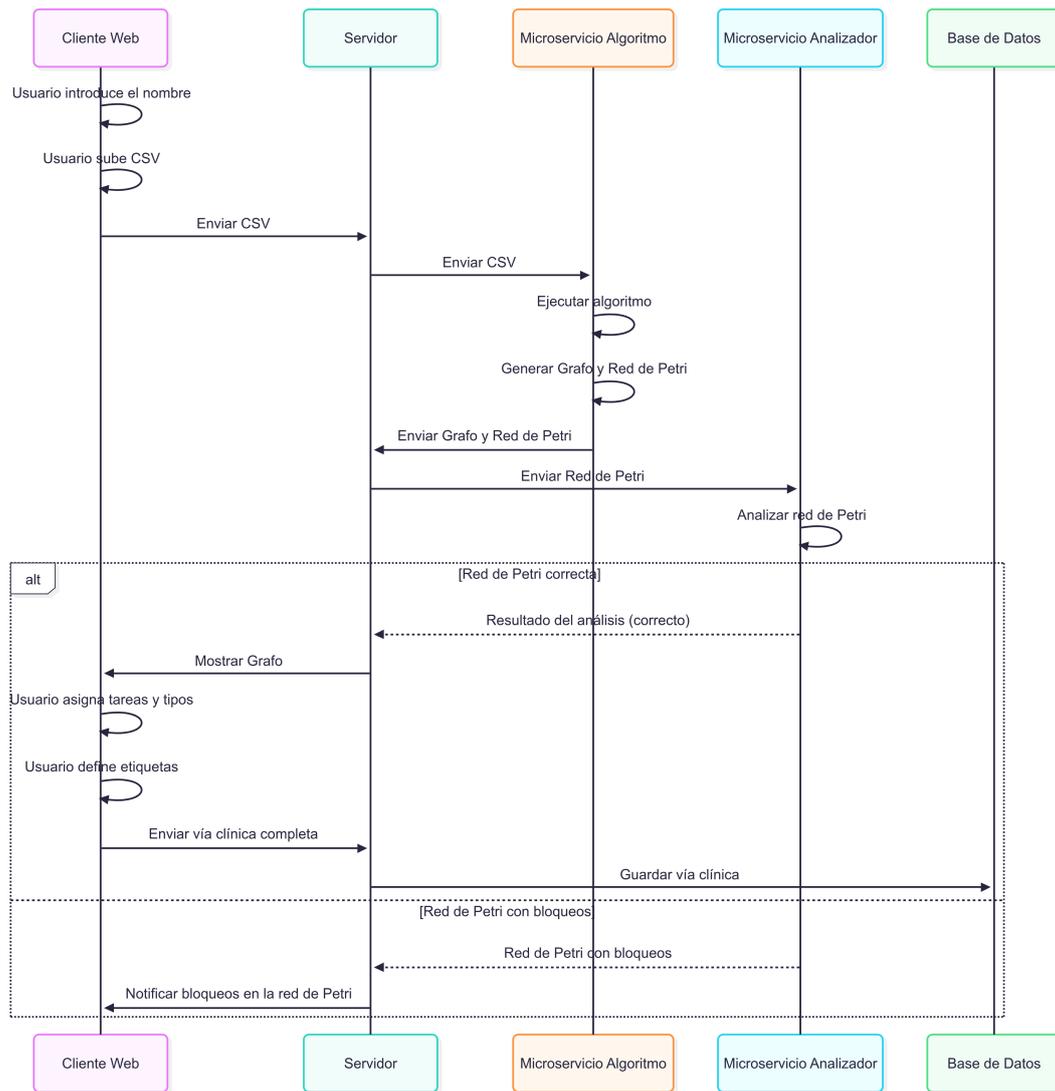


Figura 16: Diagrama de secuencia del proceso de creación y validación de una vía clínica

como objetivo mejorar la flexibilidad del proceso, permitir su integración dentro de la arquitectura basada en microservicios y asegurar la compatibilidad con herramientas de análisis posteriores.

En primer lugar, se implementó la posibilidad de que el algoritmo recibiera como parámetro el fichero de entrada en formato CSV. A diferencia de la versión original, que operaba sobre archivos estáticos definidos previamente, esta modificación permite la lectura dinámica de cualquier conjunto de datos estructurado, facilitando su uso en un entorno de producción donde los ficheros pueden variar entre ejecuciones.

En segundo lugar, se añadió una salida específica compatible con la herramienta LoLA [25]. Esta herramienta requiere una sintaxis concreta para representar redes de Petri, por lo que se desarrolló un proceso de transformación que traduce la estructura generada por el algoritmo al formato requerido. Esta compatibilidad permite aplicar análisis de verificación formal sobre las redes obtenidas, asegurando que no presenten bloqueos antes de su validación final en el sistema.

Por último, se redujo la salida generada por el algoritmo para adaptarla a los requisitos funcio-

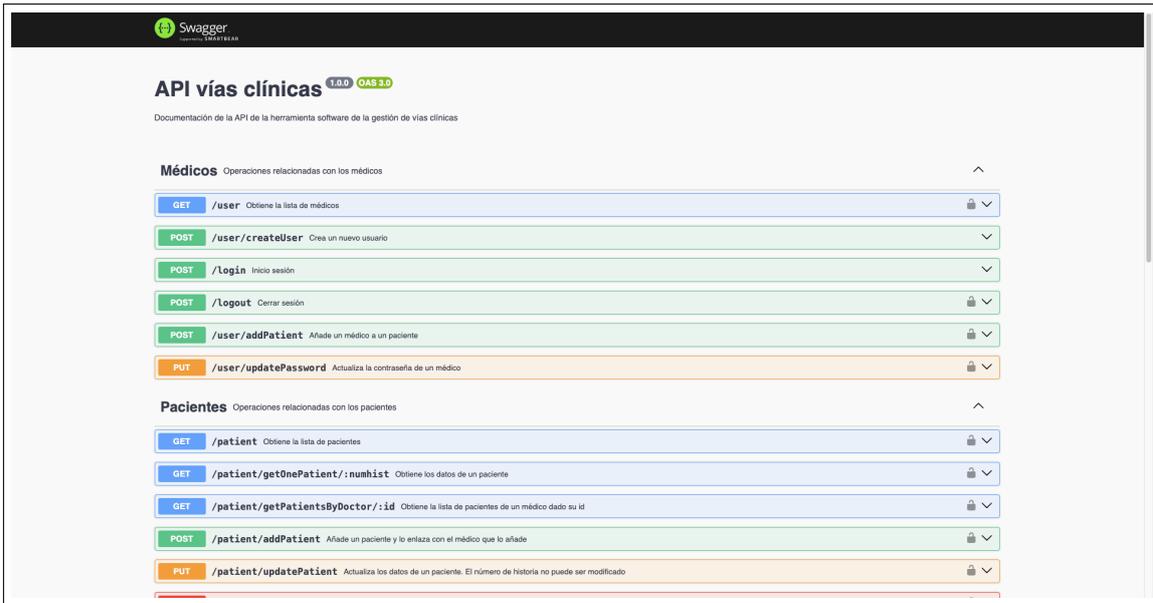


Figura 17: Documentación *Swagger* del servidor

nales del sistema. Mientras que la versión original incluía múltiples archivos intermedios y datos auxiliares, se optó por mantener únicamente la información esencial: el grafo resultante y la red de Petri correspondiente en formato LoLA. Esta simplificación mejora el rendimiento del sistema y reduce el volumen de datos transmitido entre microservicios.

Estas modificaciones han permitido que el algoritmo se integre de forma eficiente dentro del flujo general de la aplicación, garantizando tanto su reutilización como su interoperabilidad con herramientas externas de análisis.

8. Conclusiones

La realización de este Trabajo de Fin de Grado ha supuesto una oportunidad valiosa para aplicar conocimientos técnicos en un contexto real, enfrentando desafíos tanto de ingeniería como de comunicación con profesionales del ámbito sanitario. A lo largo del proyecto, se han abordado problemas relacionados con la definición precisa de requisitos, el diseño de una arquitectura modular basada en microservicios y la adaptación de herramientas algorítmicas al flujo clínico. Esta experiencia ha permitido no solo afianzar competencias técnicas clave, como el desarrollo *backend* y la integración de servicios distribuidos, sino también adquirir habilidades transversales fundamentales, como la colaboración con un cliente real, la toma de decisiones de diseño con impacto directo en el usuario final y la capacidad de adaptación frente a cambios en las necesidades del proyecto.

El resultado final ha sido una aplicación funcional que permite gestionar y personalizar vías clínicas de forma digital, cubriendo los principales requisitos definidos por los profesionales médicos. Su uso facilita el seguimiento estructurado de las intervenciones clínicas, aporta trazabilidad a los procesos hospitalarios y mejora la organización del trabajo médico. La distinción entre perfiles de usuario, junto con la validación automática de los modelos generados, contribuye a una mayor seguridad y eficiencia en la toma de decisiones clínicas. De este modo, el proyecto no solo cumple

con sus objetivos técnicos, sino que también ofrece un impacto real y positivo en el día a día del personal sanitario, al reducir tareas manuales, minimizar errores y estructurar de manera clara el flujo asistencial.

Durante las sesiones con el personal clínico, se evidenció el potencial impacto que una herramienta como esta puede tener en su práctica diaria. Los médicos nos trasladaron que, de media, atienden a unos 20 pacientes al día, lo que genera una gran cantidad de información y decisiones clínicas que deben ser gestionadas con precisión. En este contexto, la aplicación desarrollada ofrece una solución tangible para estructurar y organizar esa carga asistencial, mejorando la fluidez del trabajo, facilitando el acceso rápido a la información y promoviendo una mayor coherencia en el seguimiento de cada paciente. Además, al estandarizar los procesos mediante vías clínicas digitalizadas, se potencia la mejora continua de la calidad asistencial, permitiendo detectar ineficiencias y optimizar progresivamente la atención médica.

8.1. Trabajo Futuro

El desarrollo realizado hasta el momento constituye una base sólida para un sistema que facilita la creación, seguimiento y análisis de vías clínicas. Sin embargo, existen múltiples líneas de trabajo futuras necesarias para completar y consolidar su despliegue y utilización real en entornos clínicos.

En primer lugar, es fundamental llevar a cabo una fase de validación exhaustiva del sistema en colaboración con el personal médico. Aunque el prototipo inicial y la interfaz fueron validados parcialmente durante el desarrollo, resulta necesario realizar pruebas con datos reales en un entorno clínico controlado. Estas pruebas permitirán identificar posibles errores funcionales, evaluar la usabilidad del sistema y recoger sugerencias directamente de los profesionales sanitarios que lo utilizarán en su práctica diaria.

Este proceso se desarrollará a lo largo de los próximos meses e incluirá varias sesiones de trabajo. En una primera reunión, se presentará la versión actual del sistema, ya completamente funcional, y se explicará detalladamente su uso. A continuación, se espera que el personal médico utilice la herramienta durante un periodo de prueba de aproximadamente dos semanas. Tras este tiempo, se organizará una segunda reunión centrada en la recogida de *feedback*, con el objetivo de identificar problemas críticos que puedan dificultar su uso, así como proponer mejoras orientadas a su adopción efectiva en el entorno hospitalario.

Además, el despliegue del sistema en un entorno de producción presenta importantes desafíos tanto técnicos como legales. Al tratarse de una herramienta que gestiona información clínica sensible, cualquier decisión relativa a su infraestructura debe garantizar el cumplimiento estricto de los principios establecidos por el Reglamento General de Protección de Datos (GDPR) [6] y la Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD) [12] en España.

En este contexto, soluciones de infraestructura en la nube como Amazon Web Services (AWS) [3] ofrecen claras ventajas en cuanto a escalabilidad, disponibilidad, mantenimiento y facilidad de despliegue. No obstante, su adopción en el ámbito sanitario debe ser cuidadosamente evaluada, ya que el tratamiento de datos clínicos en entornos externos puede contravenir políticas internas del hospital o exigir medidas adicionales de protección, como el cifrado de extremo a extremo, la pseudonimización de datos o la ubicación de los servidores dentro del Espacio Económico Europeo.

Una alternativa viable sería adoptar una arquitectura híbrida, en la que la base de datos, que

contiene la información clínica, se mantenga en servidores internos del propio centro hospitalario, mientras que el resto de la aplicación (servidor web, cliente *frontend* y microservicios) se despliegue en una plataforma en la nube como AWS. Esta solución permitiría combinar el control directo sobre los datos sensibles con la flexibilidad y potencia de la infraestructura en la nube. Sin embargo, también requeriría asegurar una conexión segura y robusta entre ambos entornos, así como definir claramente las responsabilidades del tratamiento de datos entre las partes implicadas.

Antes de tomar una decisión al respecto, será imprescindible mantener una reunión con el departamento de informática del hospital para conocer el estado actual de su infraestructura, evaluar la posibilidad de integrar nuestro sistema dentro de ella y definir qué restricciones o requisitos legales aplican al almacenamiento y procesamiento de datos clínicos. Esta conversación será clave para establecer la viabilidad técnica y legal del despliegue, así como para valorar si es posible realizar una instalación piloto en un entorno controlado del hospital.

En resumen, el sistema desarrollado constituye un primer paso hacia la digitalización y análisis formal de vías clínicas, pero su consolidación requiere un proceso iterativo de validación, despliegue seguro, acompañamiento al usuario y mejora continua. La colaboración con el equipo médico será esencial para garantizar que la herramienta no solo cumpla con los requisitos técnicos, sino que también aporte valor real en la práctica asistencial.

8.2. Problemas encontrados

A lo largo del desarrollo del proyecto surgieron diversos desafíos técnicos, siendo uno de los más significativos la necesidad de integrar y adaptar un algoritmo previamente desarrollado por terceros. Este componente, esencial para la generación automática de modelos de vías clínicas, carecía de una documentación extensa, lo que dificultó su comprensión inicial y supuso una inversión considerable de tiempo en análisis de código, pruebas y validación de resultados. Fue necesario modificar su comportamiento para aceptar entradas dinámicas y ajustarlo a las necesidades específicas del sistema desarrollado.

Otro obstáculo importante fue la elección de la tecnología adecuada para llevar a cabo el análisis de las redes de Petri generadas. Inicialmente se exploraron opciones como Renew [24] o TimeNet [23], herramientas académicas conocidas en este ámbito. Sin embargo, su integración resultó inviable debido a limitaciones técnicas, falta de soporte o complejidad en su uso automatizado dentro de una arquitectura web. Finalmente, se optó por utilizar LoLA [25], una herramienta más versátil y fácilmente integrable, que permitió automatizar la verificación de los modelos generados con resultados satisfactorios.

8.3. Lecciones aprendidas

A lo largo del desarrollo de este proyecto, han surgido numerosos aprendizajes que resultan especialmente valiosos para futuros trabajos técnicos. Uno de los aspectos más relevantes ha sido la importancia de mantener una organización clara desde las primeras fases. Registrar de forma sistemática cada avance o decisión tomada ha demostrado ser fundamental, especialmente cuando se retoma una tarea tras un periodo de inactividad. La ausencia de anotaciones precisas puede generar confusión y ralentizar el progreso.

Otro punto crítico ha sido la necesidad de revisar en profundidad el modelo de datos desde las fases iniciales del proyecto. Una planificación incompleta o poco detallada en este aspecto puede

derivar en inconsistencias difíciles de resolver una vez avanzado el desarrollo. En este sentido, una validación temprana y frecuente del diseño del sistema puede evitar retrabajos y errores posteriores.

Además, he comprendido el valor de mantener un estilo de programación coherente. Seguir convenciones claras, como el uso uniforme de nombres en camel case o una nomenclatura consistente de las variables, facilita tanto el desarrollo como la posterior comprensión del código. En este proyecto, ciertas decisiones poco rigurosas, como alternar entre identificadores como *id_via* e *iope-racion*, provocaron pérdidas de tiempo evitables que, con una disciplina mayor desde el principio, podrían haberse evitado.

Por último, este proyecto ha supuesto una valiosa inmersión en el proceso completo de ingeniería de software dentro de un entorno real y colaborativo. Desde las primeras reuniones con el equipo médico, hasta la definición conjunta de requisitos, el diseño arquitectural, la toma de decisiones tecnológicas y la validación continua del producto, he podido experimentar de forma práctica cada una de las etapas clave del ciclo de vida de un sistema software. Esta experiencia me ha permitido comprender la importancia del diálogo constante con los usuarios finales, así como la necesidad de adaptar las soluciones técnicas a las verdaderas necesidades del cliente. Más allá de los aspectos puramente técnicos, he reforzado habilidades esenciales como la comunicación, la planificación iterativa y la toma de decisiones en contextos de incertidumbre, todo ello imprescindible para el desarrollo exitoso de productos útiles y sostenibles.

Bibliografía

- [1] Julio Abascal, Ignacio Aedo, José J. Cañas, Miguel Gea, Ana Belén Gil, Jesús Lorés, Ana Belén Martínez, Manuel Ortega, Pedro Valero y Manuel Vélez. *La interacción persona-ordenador*. Jesús Lorés, Editor, 2001. ISBN: 84-607-2255-4. URL: <https://aipo.es/wp-content/uploads/2022/02/LibroAIP0.pdf>.
- [2] Adobe. *Cómo utilizar Markdown para escribir documentación técnica*. Último acceso: 2025-06-20. 2024. URL: <https://experienceleague.adobe.com/es/docs/contributor-contributor-guide/writing-essentials/markdown>.
- [3] Amazon Web Services, Inc. *Amazon Web Services (AWS) – Cloud Computing Services*. Último acceso: 2025-06-23. 2025. URL: <https://aws.amazon.com>.
- [4] ECMA International. *ECMA-262, ECMAScript® 2026 Language Specification*. Último acceso: 2025-06-18. 2023. URL: <https://tc39.es/ecma262/>.
- [5] ECMA International. *The JSON Data Interchange Syntax*. Último acceso: 2025-06-19. 2017. URL: <https://ecma-international.org/wp-content/uploads/ECMA-404.pdf>.
- [6] El Parlamento Europeo y el consejo de la Unión Europea. *REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO*. Último acceso: 2025-06-23. 2016. URL: <https://www.boe.es/doue/2016/119/L00001-00088.pdf>.
- [7] A. Fox y D.A. Patterson. *Engineering Software as a Service: An Agile Approach Using Cloud Computing*. Strawberry Canyon LLC, 2013. ISBN: 9780984881246. URL: <https://books.google.es/books?id=3kqjmwEACAAJ>.
- [8] Carlo Ghezzi, Mehdi Jazayeri y Dino Mandrioli. *Fundamentals of software engineering*. USA: Prentice-Hall, Inc., 1991. ISBN: 013820432.
- [9] Holistics. *DBML - Database Markup Language*. Último acceso: 2025-06-20. 2025. URL: <https://dbml.dbdiagram.io/home/>.
- [10] TJ Holowaychuk y Contributors. *Express – Fast, unopinionated, minimalist web framework for Node.js*. Último acceso: 2025-06-18. 2023. URL: <https://expressjs.com/>.
- [11] ISO/IEC. *Information technology — Database languages SQL*. Último acceso: 2025-06-18. 2016. URL: <https://www.iso.org/standard/76583.html>.
- [12] Jefatura del Estado. *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. Último acceso: 2025-06-23. 2018. URL: <https://www.boe.es/eli/es/lo/2018/12/05/3>.
- [13] Le Moigne, Cristian Mahulea, Grégory Faraut y Jérémie Saives. *ARPE identification*. Último acceso: 2025-06-23. 2023. URL: <https://gitlab.crans.org/bleizi/arpe-identification>.
- [14] Tadao Murata. “Petri nets: Properties, analysis and applications”. En: *Proceedings of the IEEE* 77.4 (2002), págs. 541-580.

- [15] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. 1st. O'Reilly Media, feb. de 2015, pág. 280. ISBN: 978-1491950357.
- [16] Node.js Foundation. *Node.js JavaScript runtime*. Último acceso: 2025-06-18. 2023. URL: <https://nodejs.org/>.
- [17] PostgreSQL Global Development Group. *PostgreSQL Documentation*. Último acceso: 2025-06-18. 2023. URL: <https://www.postgresql.org>.
- [18] Postman, Inc. *Postman*. Herramienta para pruebas de APIs. Último acceso: 2025-06-19. 2025. URL: <https://www.postman.com>.
- [19] Python Software Foundation. *Python Programming Language*. Último acceso: 2025-06-18. 2023. URL: <https://www.python.org/>.
- [20] Sebastián Ramírez. *FastAPI – Modern, fast (high-performance), web framework for building APIs with Python*. Último acceso: 2025-06-18. 2023. URL: <https://fastapi.tiangolo.com/>.
- [21] React. *React, The library for web and native user interfaces*. Último acceso: 2025-06-18. 2023. URL: <https://react.dev>.
- [22] Tailwind Labs. *Tailwind CSS: A utility-first CSS framework*. Último acceso: 2025-06-18. 2025. URL: <https://tailwindcss.com>.
- [23] Technische Universität Ilmenau. *TimeNET – Stochastic Petri Net Analyzer*. Último acceso: 2025-06-19. 2025. URL: <https://timenet.tu-ilmenau.de/>.
- [24] University of Hamburg. *Renew – The Reference Net Workshop*. Último acceso: 2025-06-19. 2025. URL: <http://www.renew.de>.
- [25] Karsten Wolf y Niels Lohmann. *LoLA, A Low Level Petri Net Analyser*. Último acceso: 2025-06-18. 2016. URL: <https://theo.informatik.uni-rostock.de/en/theo-forschung/tools/lola/>.

Glosario

artroplastia Procedimiento quirúrgico que tiene como objetivo reemplazar, total o parcialmente, una articulación dañada por una prótesis artificial. 1, 6, 23

Acrónimos

API Application Program Interface. 7, 15, 17, 23, 24, 30

AWS Amazon Web Services. 33

CSV Comma Separated Values. 9, 24, 28, 30, 31, 44

DBML Database Markup Language. 20

ER Entidad-Relación. 18

HCU Hospital Clínico Universitario. 1, 5, 6, 13, 24

HTTPS Protocolo de Transferencia de Hipertexto Seguro. 15

JSON JavaScript Object Notation. 16

REST Representational State Transfer. 7, 15, 17, 24

SSH Secure SHell. 24

Lista de Figuras

1.	Proceso de creación de una nueva vía clínica	11
2.	Proceso de edición de una vía clínica existente	12
3.	Prototipo de la pantalla para la creación de una vía clínica	14
4.	Prototipo de la pantalla de registro de las tareas de una operación	14
5.	Arquitectura del sistema	15
6.	Modelo Entidad-Relación del sistema	19
7.	Modelo relacional de la base de datos	21
8.	Interfaz de búsqueda de pacientes por número de historia clínica	25
9.	Visualización de la información de un paciente y su historial de operaciones	26
10.	Creación de una nueva operación asociada a una vía clínica	26
11.	Registro de tareas completadas en la vía clínica de una operación	27
12.	Listado y búsqueda de vías clínicas registradas. Opción de crear una nueva vía.	28
13.	Visualización del grafo generado y la red de Petri resultante para su validación.	28
14.	Asignación de tareas por fase y definición del tipo de respuesta asociada.	29
15.	Pantalla de comprobación final de la vía clínica antes de su validación.	29
16.	Diagrama de secuencia del proceso de creación y validación de una vía clínica	31
17.	Documentación <i>Swagger</i> del servidor	32
B.1.	Pantalla de inicio de sesión del prototipo	45
B.2.	Pantalla de inicio para el perfil médico	46
B.3.	Pantalla de cambio de contraseña	46
B.4.	Pantalla de creación de un nuevo paciente	47
B.5.	Pantalla de búsqueda de pacientes	47
B.6.	Pantalla de información del paciente	48
B.7.	Pantalla de registro de nueva operación	48
B.8.	Pantalla de visualización de la vía clínica	49
B.9.	Pantalla de inicio del perfil administrador	49

B.10.Pantalla de creación de nuevo usuario	50
B.11.Pantalla de asignación de médico a paciente	50
B.12.Pantalla de selección de vía clínica	51
B.13.Definición del nombre de la nueva vía clínica	51
B.14.Visualización del grafo generado por el algoritmo	52
B.15.Asignación de las tareas en la creación de una nueva vía clínica	52
B.16.Selección de la etapa en la creación de una nueva vía clínica	53
B.17.Definir de una nueva tarea en la creación de una nueva vía clínica	53
B.18.Visualización de la red de Petri generada	54
B.19.Pantalla de vía clínica seleccionada	54
B.20.Pantalla para actualizar una vía clínica	55
B.21.Pantalla de carga de datos desde archivo Excel	55
C.1. Cronograma con las actividades y tiempos del proyecto	56

Apéndice A

Detalle de los requisitos del sistema

Requisitos funcionales de los médicos

- **RFM-1:** El sistema debe permitir la visualización de las acciones de la vía clínica (T3) de una operación (T2) de un paciente (T1).
- **RFM-2:** El sistema debe permitir la modificación de los datos de una vía clínica (T3) de una operación (T2) de un paciente (T1).
- **RFM-3:** El sistema debe permitir crear un paciente (T1).
- **RFM-4:** El sistema debe permitir editar un paciente (T1).
- **RFM-5:** El sistema debe permitir buscar un paciente (T1) mediante su número de historia clínica.
- **RFM-6:** El sistema debe permitir visualizar las operaciones (T2) de un paciente (T1).
- **RFM-7:** El sistema debe permitir visualizar los datos de un paciente (T1).
- **RFM-8:** El sistema debe permitir visualizar los datos de un paciente (T1) para una operación (T2).
- **RFM-9:** El sistema debe permitir editar los datos de un paciente (T1) para una operación (T2).
- **RFM-10:** El sistema debe permitir seleccionar la vía clínica (3) al crear una nueva operación (T2).
- **RFM-11:** El sistema debe mostrar la vía clínica (T3) correspondiente a la operación (T2).
- **RFM-12:** El sistema debe permitir marcar como completadas las tareas (T4) de una operación (T2) asociada a una vía clínica (T3).

Requisitos funcionales del administrador-médico

- **RFA-1:** El sistema debe permitir la creación de cuentas para los médicos.

- **RFA-2:** El sistema debe permitir añadir un médico a una vía clínica (T3) de una operación (T2) de un paciente (T1).
- **RFA-3:** El sistema debe permitir exportar los datos.
- **RFA-4:** El sistema debe permitir visualizar el listado de todas las vías clínicas (T3) registradas.
- **RFA-5:** El sistema debe permitir seleccionar una vía clínica (T3) específica para consultar su contenido o editarla.
- **RFA-6:** El sistema debe mostrar el detalle completo de una vía clínica (T3) seleccionada, incluyendo sus fases y tareas (T4).
- **RFA-7:** El sistema debe permitir exportar los datos recopilados de una vía clínica (T3) seleccionada, incluyendo únicamente las tareas (T4) marcadas como realizadas en operaciones (T2) anteriores vinculadas a dicha vía.
- **RFA-8:** El sistema debe permitir al administrador iniciar el proceso de creación de una nueva vía clínica (T3) mediante la introducción del nombre.
- **RFA-9:** Tras la introducción de los datos, el sistema debe ejecutar un algoritmo que genere automáticamente un modelo de red de Petri [14] representando el flujo de la vía clínica (T3).
- **RFA-10:** El sistema debe analizar la red de Petri obtenida y rechazarla si presenta bloqueos.
- **RFA-11:** El sistema debe mostrar de forma visual la red de Petri generada para su evaluación por parte del administrador.
- **RFA-12:** El sistema debe permitir al administrador aceptar o rechazar la red de Petri generada:
 - **RFA-12.1:** En caso de rechazo, el administrador podrá modificar los datos y volver a ejecutar el algoritmo.
 - **RFA-12.2:** En caso de aceptación, se continuará con la asignación de tareas (T4).
- **RFA-13:** El sistema debe permitir la asignación de tareas (T4) específicas a cada fase de la vía clínica (T3).
- **RFA-14:** El sistema debe permitir añadir nuevas tareas (T4) de manera manual.
- **RFA-15:** El sistema debe permitir definir el tipo de respuesta requerida para cada tarea (T4): dicotómicas, opción múltiple o valor numérico.
- **RFA-16:** El sistema debe permitir añadir etiquetas a una vía clínica (T3) para identificar las operaciones (T2) posteriores.
- **RFA-17:** Una vez completadas y validadas todas las etapas, el sistema debe guardar la vía clínica (T3) y los datos asociados en la base de datos para su posterior utilización.
- **RFA-18:** El sistema debe permitir al administrador modificar una vía clínica (T3) previamente registrada.
- **RFA-19:** Las modificaciones posibles deben incluir:
 - **RFA-19.1:** Edición del nombre y descripción de la vía clínica (T3).

- **RFA-19.2:** Gestión de tareas (T4) existentes (editar, eliminar o añadir).
- **RFA-19.3:** Generación de una nueva red de Petri basada en los datos actuales de la vía clínica (T3), recopilados a partir de operaciones (T2) reales asociadas a la misma.
- **RFA-20:** El sistema debe permitir realizar un análisis de la Red de Petri obtenida.
 - **RFA-20.1:** El sistema debe descartar la vía clínica si el análisis de la Red de Petri ha sido desfavorable.

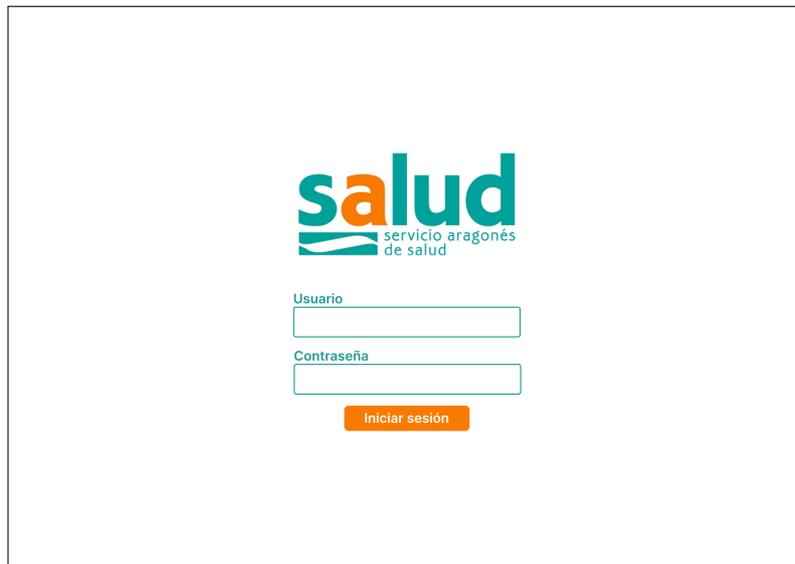
Requisitos no funcionales

- **RNF-1:** Los datos exportados deben ser en formato Comma Separated Values (CSV).
- **RNF-2:** El sistema debe estar disponible en idioma español, con terminología coherente con el uso clínico habitual.

Apéndice B

Prototipo completo

Pantalla de inicio de sesión



The image shows a login screen for the 'salud' service. At the top center is the logo for 'salud servicio aragonés de salud', with 'salud' in large blue and orange letters and 'servicio aragonés de salud' in smaller blue text below it. Below the logo are two input fields: the first is labeled 'Usuario' and the second is labeled 'Contraseña'. Below these fields is an orange button with the text 'Iniciar sesión'.

Figura B.1: Pantalla de inicio de sesión del prototipo

La Figura B.1 muestra la pantalla de inicio de sesión, que se presenta cuando el usuario accede a la aplicación sin haber iniciado sesión previamente. Desde aquí, puede autenticarse para acceder al sistema.

Pantalla de inicio del perfil médico



Figura B.2: Pantalla de inicio para el perfil médico

Tras iniciar sesión aparecerá la figura B.2, los médicos acceden a esta pantalla principal. Desde ella pueden crear nuevos pacientes, buscar entre los ya registrados o cambiar su contraseña.

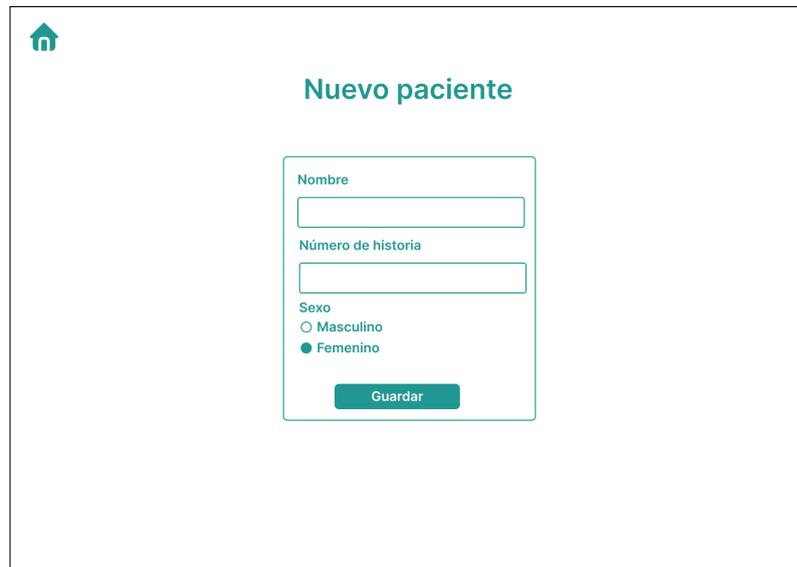
Pantalla de cambio de contraseña



Figura B.3: Pantalla de cambio de contraseña

La pantalla de la figura B.3 permite al médico modificar su contraseña de acceso al sistema. Solo es necesario introducir la nueva contraseña.

Pantalla de creación de un nuevo paciente



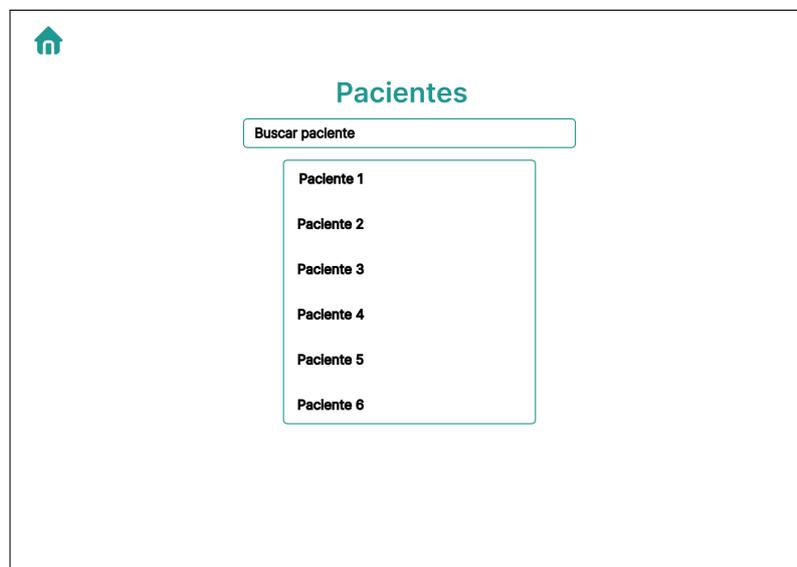
The screenshot shows a web interface for creating a new patient. At the top left is a home icon. The main heading is "Nuevo paciente". Below it is a form with the following elements:

- Field: Nombre
- Field: Número de historia
- Section: Sexo
 - Radio button: Masculino
 - Radio button: Femenino (selected)
- Button: Guardar

Figura B.4: Pantalla de creación de un nuevo paciente

Desde la pantalla de la figura B.4, el médico puede registrar un nuevo paciente introduciendo sus datos personales y clínicos básicos.

Pantalla de búsqueda de pacientes



The screenshot shows a web interface for searching patients. At the top left is a home icon. The main heading is "Pacientes". Below it is a search bar labeled "Buscar paciente". Underneath the search bar is a list of six patients:

- Paciente 1
- Paciente 2
- Paciente 3
- Paciente 4
- Paciente 5
- Paciente 6

Figura B.5: Pantalla de búsqueda de pacientes

En la figura B.5 el médico puede buscar pacientes registrados mediante filtros como el nombre o el número de historia clínica, y acceder a su información.

Pantalla de información del paciente

Información del paciente

Número de historia: 123123
Nombre: Jaime Perez
Sexo: Hombre

Operaciones Nueva operacion

Operación: vía rodilla
Fecha: 23/2/2025
Índice masa corporal: 21
Edad: 22
Etiqueta: artroplastia primera izquierda

Figura B.6: Pantalla de información del paciente

La figura B.6 Permite visualizar información detallada de un paciente previamente seleccionado, incluyendo operaciones registradas y vías clínicas asociadas.

Pantalla de nueva operación

Nueva operación

Índice masa corporal

Edad

Etiquetas

Buscar vía

Vía 1
Vía 2
Vía 3
Vía 4

Crear

Figura B.7: Pantalla de registro de nueva operación

Desde esta pantalla (Figura B.7), el médico puede registrar una nueva intervención quirúrgica asociada a un paciente.

Pantalla de visualización de vía clínica

Tarea	Sí	No
Verificar documentación completa en historia clínica electrónica (HCE)	<input checked="" type="radio"/>	<input type="radio"/>
Consentimiento informado	<input type="radio"/>	<input checked="" type="radio"/>
Planificar fecha de la intervención y revisión por anestesia	<input checked="" type="radio"/>	<input type="radio"/>
Verificar que ha sido valorada por anestesia y posibles incidencias	<input type="radio"/>	<input type="radio"/>
Verificar medicación habitual del paciente y comorbilidades	<input type="radio"/>	<input type="radio"/>
Solicitar pruebas complementarias y/o interconsultas interservicio	<input type="radio"/>	<input type="radio"/>
Evaluar hemograma y pautar si procede de prevención de transfusión	<input type="radio"/>	<input type="radio"/>
Planificación preIQ:		
Solicitar material	<input type="radio"/>	<input type="radio"/>
Solicitud de banco de tejidos si procede	<input type="radio"/>	<input type="radio"/>
Revisión de consentimiento informado	<input type="radio"/>	<input type="radio"/>
Actualización de pruebas complementarias si procede: Rx de control	<input type="radio"/>	<input type="radio"/>

Figura B.8: Pantalla de visualización de la vía clínica

La pantalla de la imagen B.8 corresponde a una de las etapas del proceso clínico, donde el usuario debe completar el registro de las tareas realizadas en cada operación.

Pantalla de inicio del perfil administrador

salud
servicio aragonés
de salud

Bienvenido al sistema

Vías clínicas

Nuevo Usuario

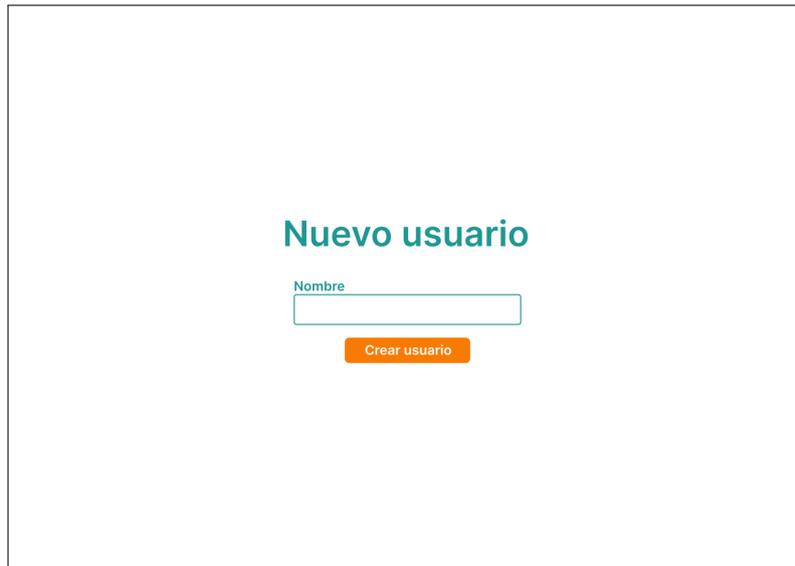
Añadir médico a paciente

Cerrar sesión

Figura B.9: Pantalla de inicio del perfil administrador

La figura B.9 representa la pantalla principal para los usuarios con perfil administrador, desde la cual pueden gestionar usuarios y las vías clínicas.

Pantalla de creación de nuevo usuario



Nuevo usuario

Nombre

Crear usuario

Figura B.10: Pantalla de creación de nuevo usuario

La figura B.10 permite al administrador registrar nuevos usuarios en el sistema, obteniendo su nombre de usuario y contraseña.

Pantalla de asignación de médico a paciente



Añadir médico a paciente

Usuario del médico

Número de historia del paciente

Añadir

Figura B.11: Pantalla de asignación de médico a paciente

La figura B.11 representa la funcionalidad para vincular un médico a un paciente determinado.

Pantalla de selección de vía clínica



Figura B.12: Pantalla de selección de vía clínica

La figura B.12 permite al médico seleccionar la vía clínica que desea visualizar.

Pantallas de creación de una nueva vía clínica

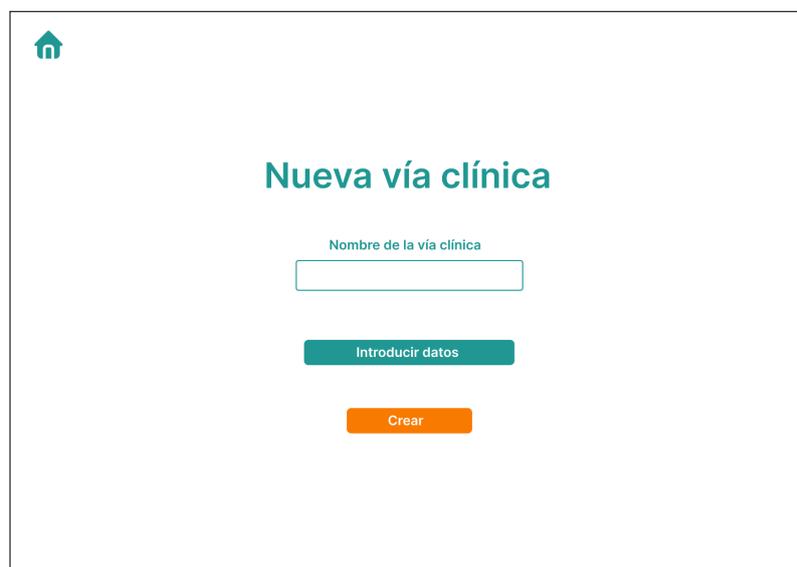


Figura B.13: Definición del nombre de la nueva vía clínica

En esta pantalla (Figura B.13) el usuario introduce el nombre que identificará la nueva vía clínica que se va a crear, estableciendo así la base para su configuración posterior.



Figura B.14: Visualización del grafo generado por el algoritmo

Aquí (Figura B.14) se muestra una representación gráfica del grafo generado automáticamente por el algoritmo, que refleja la estructura inicial y las relaciones entre las diferentes etapas de la vía clínica.



Figura B.15: Asignación de las tareas en la creación de una nueva vía clínica

Esta pantalla (Figura B.15) permite asignar y organizar las tareas específicas que conforman la vía clínica, vinculándolas a las diferentes etapas para definir el flujo de trabajo.



Figura B.16: Selección de la etapa en la creación de una nueva vía clínica

En esta figura B.16 el usuario selecciona la etapa clínica a la que corresponde cada tarea, permitiendo ordenar cronológicamente el proceso asistencial.



Figura B.17: Definir de una nueva tarea en la creación de una nueva vía clínica

Esta pantalla (Figura B.17) permite introducir los detalles específicos de una nueva tarea, incluyendo nombre, descripción y otros atributos necesarios para su correcta definición dentro de la vía clínica.

Pantalla de generación y visualización de red de Petri



Figura B.18: Visualización de la red de Petri generada

En la figura B.18 se presenta la red de Petri generada automáticamente a partir de la definición de la vía clínica, facilitando su análisis formal y validación del flujo asistencial.

Pantalla de vía clínica seleccionada



Figura B.19: Pantalla de vía clínica seleccionada

Esta pantalla muestra la vía clínica seleccionada, permitiendo al usuario consultar detalles específicos y acceder a las funcionalidades de gestión disponibles.

Pantalla de actualización de vía clínica

Home icon

Vía X

Planificación Ingreso Quirofano Post Cirugía Primer día Segundo día Post Operatorio

Planificación

Añadir

Tarea 1 Si/No

Tarea 2 Numérico

Tarea 3

Opción 1

Opción 2

Opción 3

Opciones

Generar Red de Petri

Guardar

Figura B.20: Pantalla para actualizar una vía clínica

Figura B.20, interfaz destinada a modificar y actualizar los parámetros y tareas de una vía clínica existente, facilitando la incorporación de cambios necesarios.

Pantalla de carga de datos desde Excel



Figura B.21: Pantalla de carga de datos desde archivo Excel

Esta pantalla (Figura B.21) permite importar datos desde un archivo Excel para automatizar y agilizar la creación o actualización de vías clínicas, evitando la entrada manual de datos.

Apéndice C

Diagrama de Gantt

ACTIVIDADES	FEBRERO				MARZO				ABRIL				MAYO				JUNIO				TOTAL
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	
A1: ANÁLISIS DE REQUISITOS	10	5	15	5						5											40
A2: DISEÑO DEL SISTEMA				10	10	10				5											35
A3: IMPLEMENTACIÓN						10	20	10	10	10	5	15	10	20	10					120	
A4: VALIDACIÓN									5		5						5				15
A5: DOCUMENTACIÓN			5	10										5	5		5	20	25	5	80
TOTAL	10	5	20	25	10	20	20	10	15	10	15	5	15	15	25	10	10	20	25	5	290

Figura C.1: Cronograma con las actividades y tiempos del proyecto