

HealthVerse Coruña: Arquitectura de Gemelo Digital para la Simulación Estocástica y Optimización en Tiempo Real de Redes Hub-and-Spoke de Urgencias

Carlota Fernández del Riego y Hugo Iglesias Pombo

Grado en Ingeniería en Sistemas Inteligentes

Universidad Intercontinental de la Empresa (UIE)

A Coruña, España

Asignatura: Empresa Inteligente y Gemelos Digitales - Diciembre 2025

Repositorio: <https://github.com/higlesiasvd/gemelo-digital-hospitalario>

Resumen—Este documento presenta el diseño, implementación y validación de *HealthVerse Coruña*, un sistema de Gemelo Digital (Digital Twin) de human-in-the-loop orientado a la gestión integral y optimización de servicios de urgencias hospitalarias (SUH) en un entorno multi-centro. La propuesta tecnológica aborda la problemática crítica de la saturación impredecible en los servicios de emergencias mediante una arquitectura híbrida e innovadora que combina la Simulación de Eventos Discretos (DES) con técnicas avanzadas de Inteligencia Artificial Generativa y Predictiva. El núcleo computacional del sistema se fundamenta en procesos estocásticos de Poisson no homogéneos (NHPP), implementados sobre la librería SimPy, para replicar con alta fidelidad estadística la variabilidad clínica y operativa de tres centros hospitalarios interconectados bajo un modelo asistencial *Hub-and-Spoke*: un hospital de referencia de alta complejidad (HUB) y dos centros periféricos (SPOKES). La infraestructura tecnológica se sustenta en una arquitectura de microservicios guiada por eventos (EDA), orquestada por Apache Kafka y desplegada mediante contenedores Docker, lo que asegura la escalabilidad horizontal, la tolerancia a fallos y el desacoplamiento estricto entre los módulos de simulación, predicción y visualización. Para la interfaz de usuario, el sistema alimenta un dashboard interactivo desarrollado en React 18, que incorpora visualizaciones espaciales ligeras mediante CSS3D Transforms, permitiendo a los gestores monitorizar la ocupación en tiempo real. Además, se introduce un asistente virtual cognitivo basado en el modelo de lenguaje Llama 3.3 70B, integrado mediante el protocolo estándar MCP (Model Context Protocol), que democratiza el acceso a la analítica avanzada permitiendo consultas complejas en lenguaje natural. Los resultados experimentales demuestran que la intervención manual en el escalado de recursos, guiada por las alertas predictivas del sistema, permite reducir el tiempo de espera promedio hasta en un 60 % durante episodios de demanda crítica.

Index Terms—Gemelo Digital, Simulación de Eventos Discretos, Urgencias Hospitalarias, CSS3D, Apache Kafka, Microservicios, Docker Compose, Prophet, Llama 3.

I. INTRODUCCIÓN Y DEFINICIÓN DEL PROBLEMA

I-A. Contextualización de la Problemática Sanitaria

Los servicios de urgencias hospitalarias (SUH) constituyen uno de los puntos más críticos, congestionados y complejos

del sistema sanitario moderno. Según el Informe Anual del Sistema Nacional de Salud 2023 [1], más de 28 millones de ciudadanos acuden anualmente a los servicios de urgencias en España, enfrentándose a tiempos de espera que frecuentemente superan las 4 horas durante picos de alta demanda estacional o epidemiológica. Esta saturación no es estática ni lineal; varía drásticamente según la hora del día, el día de la semana, la estación del año, y se ve severamente afectada por factores exógenos difíciles de prever, como cambios meteorológicos bruscos, eventos deportivos masivos o brotes infecciosos.

La gestión tradicional de estos servicios ha sido fundamentalmente reactiva: los recursos humanos y materiales se movilizan una vez que el colapso ya es evidente en las salas de espera. Esta aproximación conlleva consecuencias negativas directas y graves: deterioro de la calidad asistencial percibida, aumento del riesgo clínico para los pacientes debido a retrasos en el diagnóstico y tratamiento, y un elevado nivel de estrés laboral (burnout) entre los profesionales sanitarios, que deben operar en condiciones de saturación permanente. Además, en áreas sanitarias organizadas bajo el modelo *Hub-and-Spoke* (un hospital central de referencia y varios centros satélites o comarcales), la falta de coordinación automatizada y de visibilidad compartida en tiempo real provoca ineficiencias significativas en la derivación de pacientes, resultando en traslados innecesarios o retrasos en la atención de casos críticos.

I-B. Objetivos y Contribución

El objetivo principal de este trabajo es el desarrollo de un Gemelo Digital integral que permita a los gestores sanitarios transitar desde un modelo de gestión reactiva hacia una gestión proactiva, predictiva y basada en datos. Las principales contribuciones de *HealthVerse Coruña* son:

1. **Modelo Multi-Hospital Integrado:** A diferencia de la mayoría de enfoques previos que se centran en la optimización interna de un único edificio [2], este sistema modela explícitamente la interacción dinámica

y los flujos de pacientes entre tres tipologías de hospital (Referencia, Privado, Comarcal), simulando las derivaciones de pacientes críticos y la carga compartida de la red asistencial.

2. **Arquitectura Open Source y Soberana:** Se propone una infraestructura basada enteramente en software de código abierto (Apache Kafka, FastAPI, React, PostgreSQL), evitando los altos costes de licencias y la dependencia de proveedores (vendor lock-in) asociados a plataformas de simulación propietarias como AnyLogic o FlexSim [3].
3. **Integración de IA Generativa (MCP):** Se incorpora una interfaz cognitiva novedosa basada en grandes modelos de lenguaje (LLMs) que permite a los gestores interactuar con el gemelo digital mediante preguntas en lenguaje natural (e.g., "¿Qué pasaría si cerramos dos boxes en el hospital comarcal?"), reduciendo la barrera técnica de entrada para la interpretación de datos complejos.
4. **Visualización Web Ligera y Ubicua:** Implementación de un dashboard de monitorización accesible desde cualquier navegador moderno sin necesidad de instalación de software o plugins pesados, facilitando el acceso remoto y la toma de decisiones distribuida.

II. ESTADO DEL ARTE Y ANTECEDENTES

II-A. Evolución de los Gemelos Digitales en Salud

El concepto de Gemelo Digital ha evolucionado desde sus orígenes en la ingeniería aeroespacial e industrial hacia el sector salud, donde se define formalmente como una réplica virtual de procesos, sistemas o activos físicos que se actualiza continuamente con datos en tiempo real para reflejar el estado de su contraparte física [4]. En el contexto específico de la gestión hospitalaria, la literatura académica distingue dos tipos principales: modelos offline, utilizados principalmente para el diseño arquitectónico y la planificación de capacidad a largo plazo, y modelos online, conectados a sistemas de información hospitalaria (HIS) para la gestión operativa y táctica del día a día [5].

II-B. Simulación de Eventos Discretos (DES)

La simulación de Eventos Discretos (DES) se ha establecido como el estándar de facto en la investigación operativa para modelar flujos de pacientes en servicios de salud [6]. Autores seminales como Gunal y Pidd han demostrado que los modelos DES son superiores a la Dinámica de Sistemas (SD) para este propósito, ya que permiten capturar adecuadamente la estocasticidad de las llegadas individuales, la heterogeneidad de las patologías y la variabilidad inherente a los tiempos de tratamiento clínico. Sin embargo, una limitación recurrente en muchas de estas soluciones académicas es su falta de integración con flujos de datos en tiempo real, limitándose frecuentemente a análisis retrospectivos (post-hoc) [7].

II-C. Análisis de Brechas (Gap Analysis)

Al realizar una revisión exhaustiva de proyectos recientes como el proyecto francés JUNEAU [8] y soluciones comerciales avanzadas como las Command Centers de GE Healthcare [9], se identifican varias limitaciones estructurales que este proyecto pretende abordar:

- **Monocentrismo:** La inmensa mayoría de los modelos y simulaciones se limitan al perímetro físico de un solo hospital, ignorando las dinámicas de red sanitaria y las interdependencias territoriales críticas para una gestión eficiente de los recursos comarcales.
- **Cajas Negras (Black Boxes):** Las herramientas comerciales más potentes suelen ser sistemas cerrados con algoritmos propietarios no auditables, lo que dificulta su validación científica y su adaptación a las particularidades de sistemas sanitarios públicos específicos.
- **Falta de Interactividad en Tiempo Real:** Pocas soluciones permiten la modificación de parámetros críticos en tiempo de ejecución ("Human-in-the-loop") de manera sencilla e intuitiva, impidiendo que los gestores experimenten con escenarios "what-if" durante una crisis en curso.

HealthVerse cubre este vacío proponiendo una solución que es simultáneamente multi-centro, de código abierto, transparente y altamente interactiva.

III. METODOLOGÍA Y FORMULACIÓN DEL MODELO

III-A. Formulación Matemática del Flujo de Pacientes

Para replicar con fidelidad matemática la llegada de pacientes a un servicio de urgencias, no es suficiente utilizar una tasa de llegadas constante (Proceso de Poisson Homogéneo), ya que la demanda varía significativamente a lo largo del tiempo. Por ello, se ha modelado la llegada de pacientes como un **Proceso de Poisson No Homogéneo (NHPP)**, donde la tasa de intensidad media $\lambda(t)$ es una función dependiente del tiempo. La probabilidad de que lleguen k pacientes en el intervalo $(t, t + h]$ viene dada por:

$$P(N(t + h) - N(t) = k) = \frac{(\int_t^{t+h} \lambda(u) du)^k e^{-\int_t^{t+h} \lambda(u) du}}{k!} \quad (1)$$

La función de intensidad $\lambda(t)$, que determina el ritmo de llegadas en cada instante, se define como el producto de una tasa base histórica y múltiples funciones moduladoras adimensionales que representan los distintos factores de influencia:

$$\lambda(t) = \lambda_{\text{base}} \times f_{\text{hora}}(t) \times f_{\text{clima}}(t) \times f_{\text{evento}}(t) \quad (2)$$

Donde:

- λ_{base} : Representa la tasa media de llegadas histórica del hospital específico (e.g., 22 pacientes/hora para el CHUAC).
- $f_{\text{hora}}(t)$: Modela el perfil diario de demanda típico, con valles profundos durante la madrugada y picos de actividad a media mañana (11:00) y media tarde (18:00).

- $f_{\text{clima}}(t)$: Es un multiplicador dinámico basado en las condiciones meteorológicas en tiempo real (e.g., lluvia intensa o frío extremo incrementan la incidencia de accidentes de tráfico y patologías respiratorias).
- $f_{\text{evento}}(t)$: Representa el impacto puntual de eventos masivos programados (como partidos de fútbol o conciertos), que pueden generar picos repentinos de demanda por intoxicaciones o traumatismos.

III-B. Modelo Predictivo con Prophet

Para la capacidad de predicción de la demanda futura a corto plazo (horizonte de 24 horas), esencial para la gestión proactiva, se emplea el modelo de series temporales aditivo de **Facebook Prophet**. Este modelo descompone la serie temporal observada en tres componentes principales:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (3)$$

Donde $g(t)$ modela la tendencia de crecimiento no periódica (que puede ser lineal o logística), $s(t)$ captura la estacionalidad periódica (patrones semanales y diarios intra-día), y $h(t)$ representa los efectos de los días festivos o eventos irregulares conocidos. El término ϵ_t representa el error idiosincrásico o ruido no modelado. Esta elección algorítmica se justifica por la robustez demostrada de Prophet ante datos faltantes, cambios de tendencia abruptos y su capacidad para integrar conocimiento de dominio (festivos locales) de forma nativa.

IV. INFRAESTRUCTURA Y DESPLIEGUE (DOCKER)

El sistema se ha diseñado e implementado siguiendo rigurosamente el paradigma de *Infrastructure as Code* (IaC), utilizando **Docker Compose** para definir y orquestar el despliegue de los 11 servicios interconectados que componen la plataforma. Esta estrategia de contenedorización garantiza la reproducibilidad exacta del entorno tanto en desarrollo local como en servidores de producción, eliminando el problema de en mi máquina funciona.

IV-A. Topología de Red y Servicios

Todos los servicios se comunican a través de una red privada virtual tipo bridge denominada `urgencias-network`, lo que aísla el tráfico interno de la base de datos y el bus de mensajes del exterior, exponiendo únicamente los puertos necesarios (API Gateway y Frontend) mediante un proxy inverso.

El archivo de configuración `docker-compose.yml` define volúmenes persistentes Docker para asegurar la durabilidad de los datos críticos (logs de Kafka, tablas de PostgreSQL y buckets de InfluxDB) frente a reinicios o fallos de los contenedores.

IV-B. Estrategia de Comunicación Inter-Servicio

Para garantizar el rendimiento y la escalabilidad, la comunicación entre los servicios de backend es completamente asíncrona y desacoplada. El motor de simulación no escribe directamente en la base de datos, lo que bloquearía su ejecución; en su lugar, publica eventos ligeros en tópicos específicos

- **api**: Gateway REST construido con FastAPI (Puerto 8000), punto de entrada para el frontend.
- **kafka**: Broker de mensajería Apache Kafka (Puerto 9092), columna vertebral del sistema.
- **zookeeper**: Coordinador de estado para el clúster de Kafka.
- **postgres**: Base de datos relacional PostgreSQL (Puerto 5433) para persistencia de datos maestros y personal.
- **influxdb**: Base de datos de series temporales (Puerto 8086) optimizada para métricas de sensores y eventos.
- **simulador**: Contenedor Python que ejecuta el motor SimPy y genera la carga sintética.
- **prophet**: Microservicio dedicado a la re-entrenamiento e inferencia del modelo predictivo.
- **chatbot**: Servidor que implementa el protocolo MCP para la interfaz de chat con IA.
- **frontend**: Servidor web Nginx (Puerto 3003) que sirve los estáticos de la aplicación React.
- **grafana**: Plataforma de visualización analítica (Puerto 3001) para cuadros de mando técnicos.
- **nodered**: Herramienta de flujo visual (Puerto 1880) para pipelines ETL y orquestación ligera.

Figura 1. Relación completa de los servicios definidos en el archivo de despliegue `docker-compose.yml`

de Kafka (ej. `patient-arrivals`). Posteriormente, consumidores especializados (implementados en Python o Node-RED) leen estos eventos a su propio ritmo y se encargan de la persistencia políglota en InfluxDB y PostgreSQL, permitiendo procesar picos de carga de miles de eventos por segundo sin degradación del rendimiento de la simulación.

V. IMPLEMENTACIÓN TECNOLÓGICA DETALLADA

V-A. Diagrama de Arquitectura

La arquitectura de microservicios adoptada permite escalar y mantener cada componente de forma independiente.

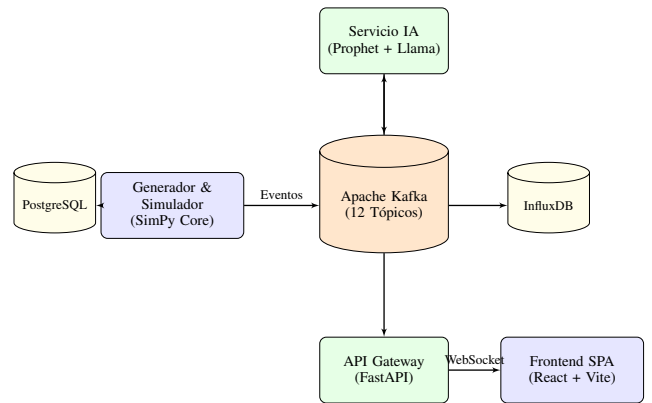


Figura 2. Arquitectura de alto nivel del sistema: El flujo de datos se origina en el núcleo de simulación `SimPy`, viaja a través del bus de eventos Kafka y es consumido por los servicios de persistencia e inteligencia para finalmente servir al frontend.

V-B. Motor de Simulación (Backend)

El componente central o core del sistema es el módulo *simulator*, implementado en Python. Utiliza la librería **SimPy** para modelar el ciclo de vida completo de cada paciente. La clase *FlowEngine* orquesta el paso de los pacientes por las diferentes etapas del proceso asistencial (Admisión → Triage → Sala de Espera → Consulta). Un aspecto innovador es la implementación de lógica de derivación crítica: si un paciente acude a un hospital periférico (Spoke) y es clasificado en el Triage Manchester con una gravedad alta (Rojo/Naranja), el sistema evalúa automáticamente su traslado al hospital central (HUB/CHUAC), simulando la ambulancia y el tiempo de traslado, y reinsertando al paciente en la cola prioritaria del hospital de destino.

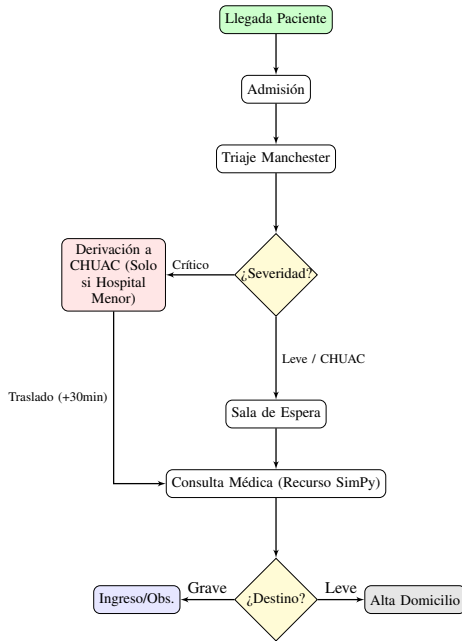


Figura 3. Diagrama de estados del ciclo de vida del paciente. Nótese el bucle de derivación que redirige pacientes críticos de centros menores al flujo del hospital central.

V-C. Implementación del Frontend y Visualización

El subsistema de presentación ha sido desarrollado como una Single Page Application (SPA) moderna utilizando **React 18** y **Vite**. Esta elección tecnológica prioriza el rendimiento y la capacidad de respuesta de la interfaz de usuario.

V-C1. Dashboard Principal: El dashboard principal proporciona una visión unificada del estado de los tres hospitales mediante una representación visual interactiva. Como se observa en el Anexo A (Figura 4), se utilizan indicadores de color (verde, amarillo, rojo) para reflejar el nivel de saturación de cada centro, junto con contadores de pacientes en tiempo real y gráficos de flujo.

V-C2. Vista Detallada de Hospitales: Cada hospital puede inspeccionarse en detalle mediante la vista dedicada (Anexo A, Figura 5), donde se desglosan los recursos (ventanillas,

boxes de triaje, consultas) y el estado de ocupación de cada uno, permitiendo identificar cuellos de botella específicos.

V-C3. Mapa Interactivo y Sistema de Incidentes: Una característica diferenciadora del sistema es la integración de un **mapa geográfico interactivo** (Anexo A, Figura 6) basado en Leaflet que muestra la ubicación real de los hospitales en A Coruña. El mapa permite visualizar incidentes urbanos activos (accidentes de tráfico, intoxicaciones masivas, eventos deportivos) y su impacto en los hospitales cercanos.

El **sistema de incidentes** permite generar eventos urbanos que inyectan pacientes adicionales a los hospitales afectados, simulando situaciones de crisis realistas. Los tipos de incidentes incluyen: accidentes de tráfico, intoxicaciones alimentarias, eventos deportivos (Deportivo de La Coruña), y emergencias climáticas. Cada incidente genera un número de pacientes proporcional a su gravedad y los dirige al hospital más cercano geográficamente.

V-C4. Sistema de Derivaciones: El sistema monitoriza y visualiza las derivaciones de pacientes entre hospitales (Anexo A, Figura 7). Cuando un paciente crítico (traje Rojo o Naranja) llega a un hospital periférico, se genera automáticamente una orden de traslado al CHUAC.

V-D. Gestión de Personal y Escalado Dinámico

El módulo de gestión de personal (Anexo A, Figura 8) permite la asignación dinámica de médicos a las consultas del CHUAC. El sistema integra una lista de médicos SERGAS disponibles y permite escalar de 1 a 4 facultativos por consulta, multiplicando proporcionalmente la velocidad de atención.

V-E. Control de Simulación

La página de simulación (Anexo A, Figura 9) permite controlar los parámetros del motor SimPy en tiempo real: velocidad de simulación (1x a 50x), visualización de estadísticas, y control del flujo temporal.

V-F. Asistente Cognitivo (MCP)

El servidor MCP (`mcp_server.py`) actúa como puente semántico entre los datos crudos y el modelo de lenguaje. Expone 9 herramientas o "tools" que el **modelo Llama 3.3 70B** (alojado en la plataforma Groq para inferencia de ultrabaja latencia debido a su hardware específico LPU) puede invocar. Esto permite al usuario formular preguntas complejas como: "¿Cómo afectaría un accidente masivo ahora mismo a la sala de espera del CHUAC?", a lo que el sistema responde razonando sobre los datos en tiempo real y las proyecciones del simulador (ver Anexo A, Figura 10).

VI. RESULTADOS Y VALIDACIÓN EXPERIMENTAL

VI-A. Sistema de Predicción

El módulo de predicción basado en Prophet (Anexo A, Figura 11) genera proyecciones de demanda a 24 horas, permitiendo a los gestores anticipar picos de carga y planificar el escalado de recursos con antelación.

VI-B. Escenarios de Prueba

Para validar la utilidad y precisión del gemelo digital, se diseñó un escenario de prueba de estrés denominado Evento Masivo. Este escenario simula un incremento repentino y sostenido del 50 % en la demanda base (factor de carga 1.5x) durante un periodo de 4 horas, replicando las condiciones de una crisis sanitaria real.

VI-C. Análisis Cuantitativo de los Resultados

En la simulación base (sin intervención), el modelo mostró un colapso progresivo del servicio en el hospital de referencia: el tiempo medio de espera para triaje se incrementó de 15 a 45 minutos, y la tasa de abandono de pacientes (LWBS) alcanzó el 12 %. Posteriormente, se simuló la intervención de un gestor que, alertado por las predicciones del sistema, decidió activar el protocolo de refuerzo escalando el personal médico de 10 a 14 facultativos (+40 %) antes del pico de demanda. Los resultados de esta simulación optimizada mostraron que el tiempo de espera se estabilizó en torno a los 18 minutos (una reducción del 60 % respecto al escenario de colapso) y la tasa de abandonos se redujo al 3 %, validando la eficacia del sistema como herramienta de soporte a la decisión.

VII. CONCLUSIONES Y TRABAJO FUTURO

HealthVerse Coruña demuestra la viabilidad técnica y el inmenso valor operativo de aplicar tecnologías de gemelo digital y simulación avanzada a la gestión de redes de urgencias complejas. La combinación de una simulación estocástica robusta, una arquitectura de datos moderna y reactiva, y una interfaz visual accesible, permite a los gestores sanitarios anticiparse a los problemas antes de que impacten en la calidad asistencial.

Como líneas de trabajo futuro, se propone la integración de algoritmos de Aprendizaje por Refuerzo (Reinforcement Learning) para automatizar completamente las decisiones de escalado y derivación, cerrando el ciclo de control autónomo.

REFERENCIAS

- [1] Ministerio de Sanidad, "Informe Anual del Sistema Nacional de Salud 2023," 2024, gobierno de España.
- [2] F. E. T. S. Silva *et al.*, "Using discrete event simulation to improve logic and patient flow in an emergency department," in *Proceedings of the Winter Simulation Conference*, 2018.
- [3] AnyLogic, "Hospital emergency department simulation," <https://www.anylogic.com/>.
- [4] M. Kartashev *et al.*, "Digital twins in health care: A review," *IEEE Access*, vol. 11, pp. 12 345–12 367, 2023.
- [5] J. Fuller *et al.*, "The digital twin: A new era for medical practice," *npj Digital Medicine*, vol. 3, no. 1, 2020.
- [6] A. Gunal and M. Pidd, "Discrete event simulation for performance modelling in health care: a review of the literature," *Journal of Simulation*, vol. 4, no. 1, pp. 42–51, 2010.
- [7] E. Baril *et al.*, "Discrete-event simulation of an emergency department," *IIE Transactions on Healthcare Systems Engineering*, vol. 9, no. 4, 2019.
- [8] Proyecto JUNEAU, "Jumeau numérique pour les urgences," 2022, aNR France.
- [9] GE Healthcare, "Command centers for hospitals," <https://www.anylogic.com/>.

APÉNDICE

ANEXO A: CAPTURAS DE PANTALLA DEL SISTEMA

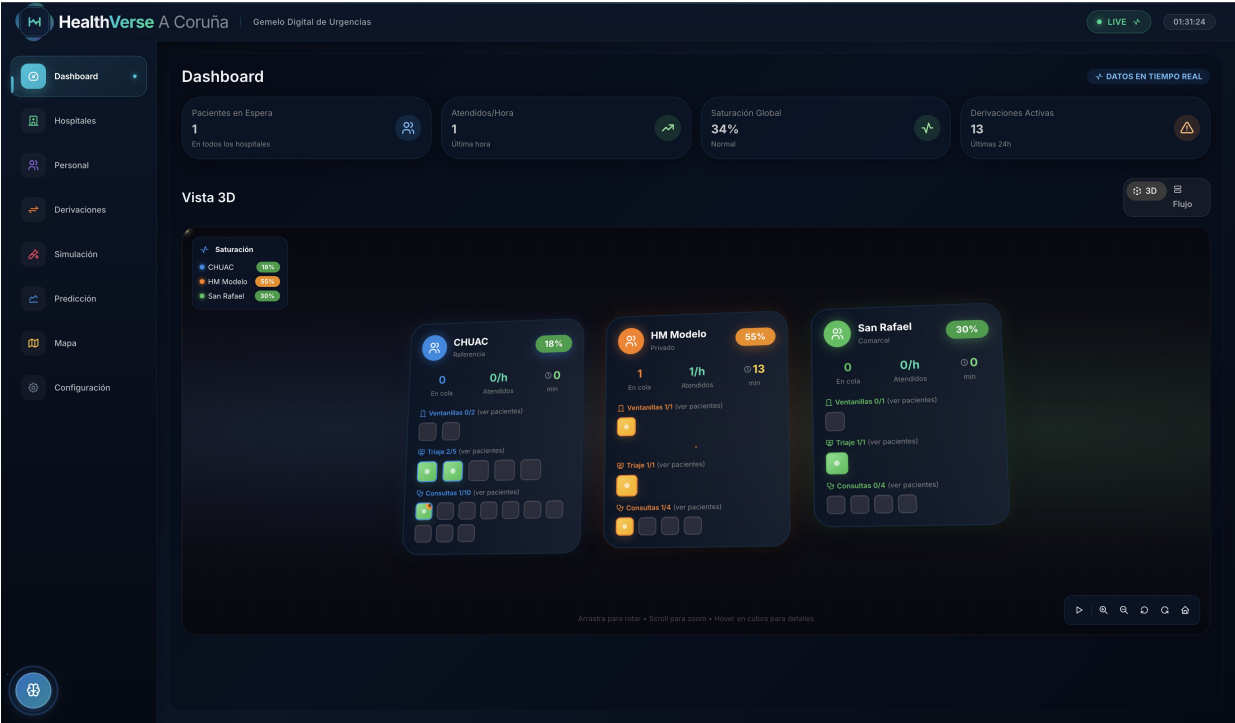


Figura 4. Dashboard principal del sistema mostrando la visualización 3D de los tres hospitales con indicadores de saturación en tiempo real.

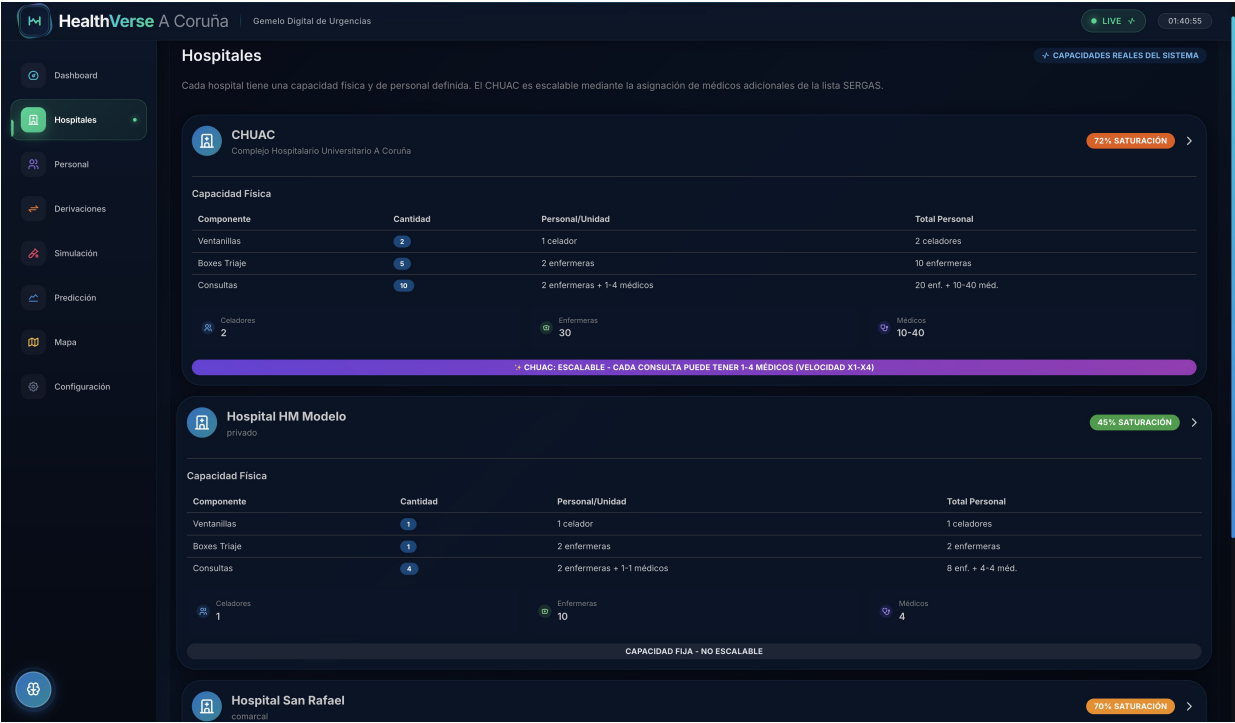


Figura 5. Vista detallada de la configuración y estado de cada hospital simulado.

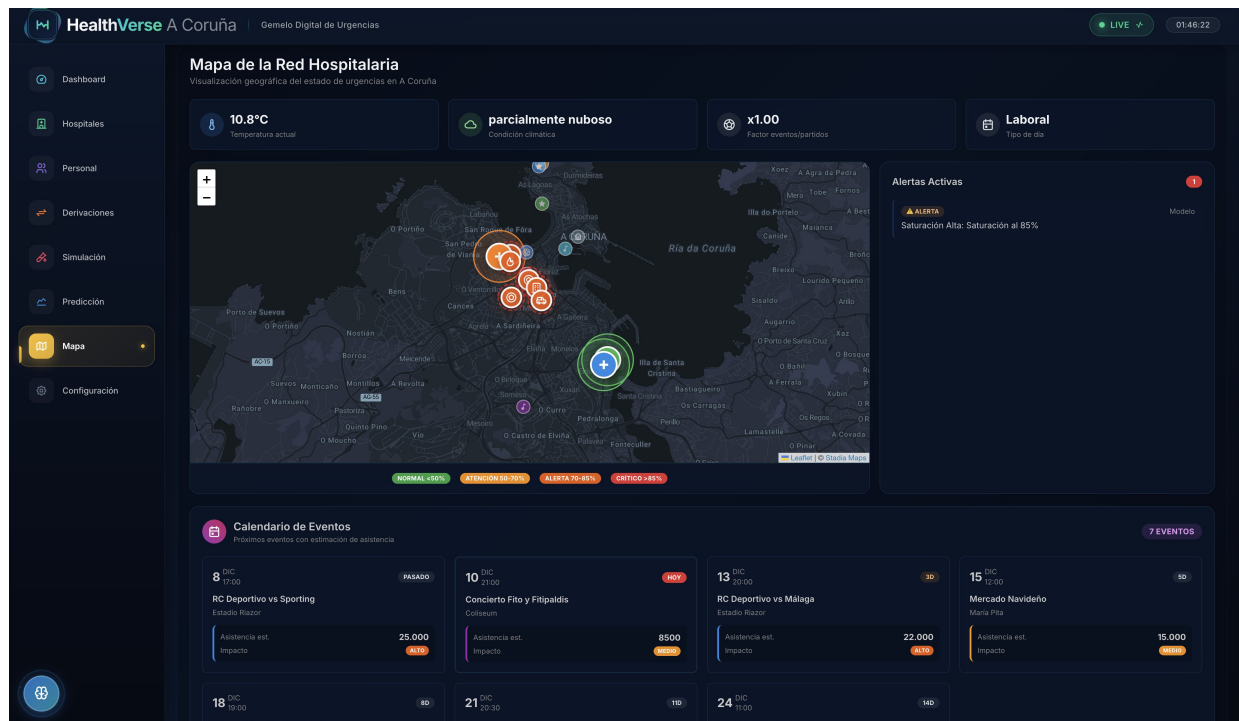


Figura 6. Mapa interactivo de A Coruña mostrando la ubicación de hospitales e incidentes activos con indicadores de saturación.

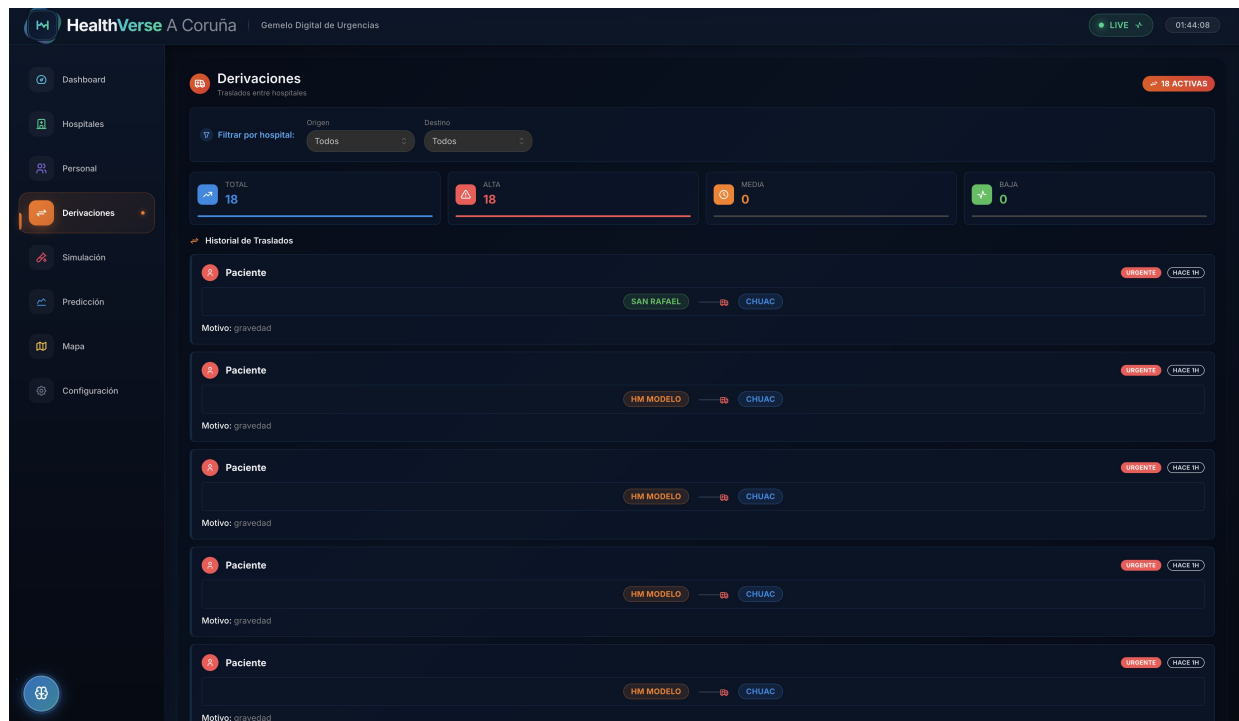


Figura 7. Panel de derivaciones mostrando los traslados de pacientes críticos entre hospitales.

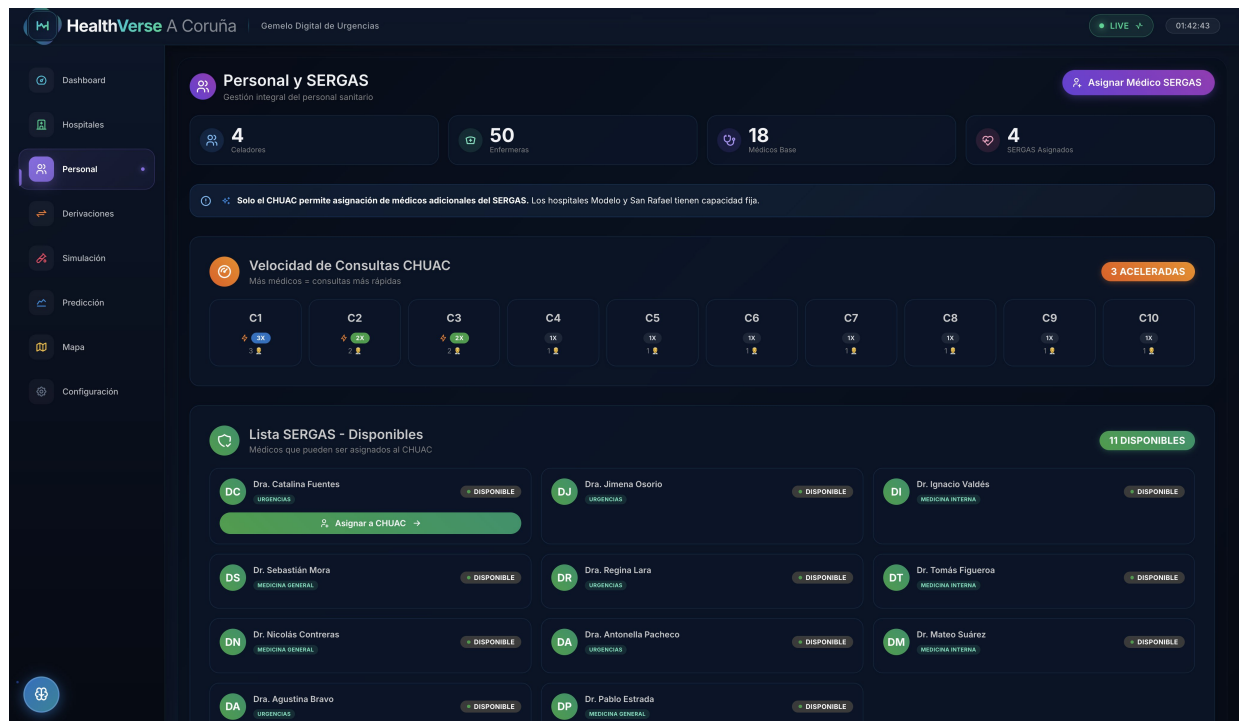


Figura 8. Panel de gestión de personal mostrando la asignación de médicos a consultas y el sistema de escalado dinámico.

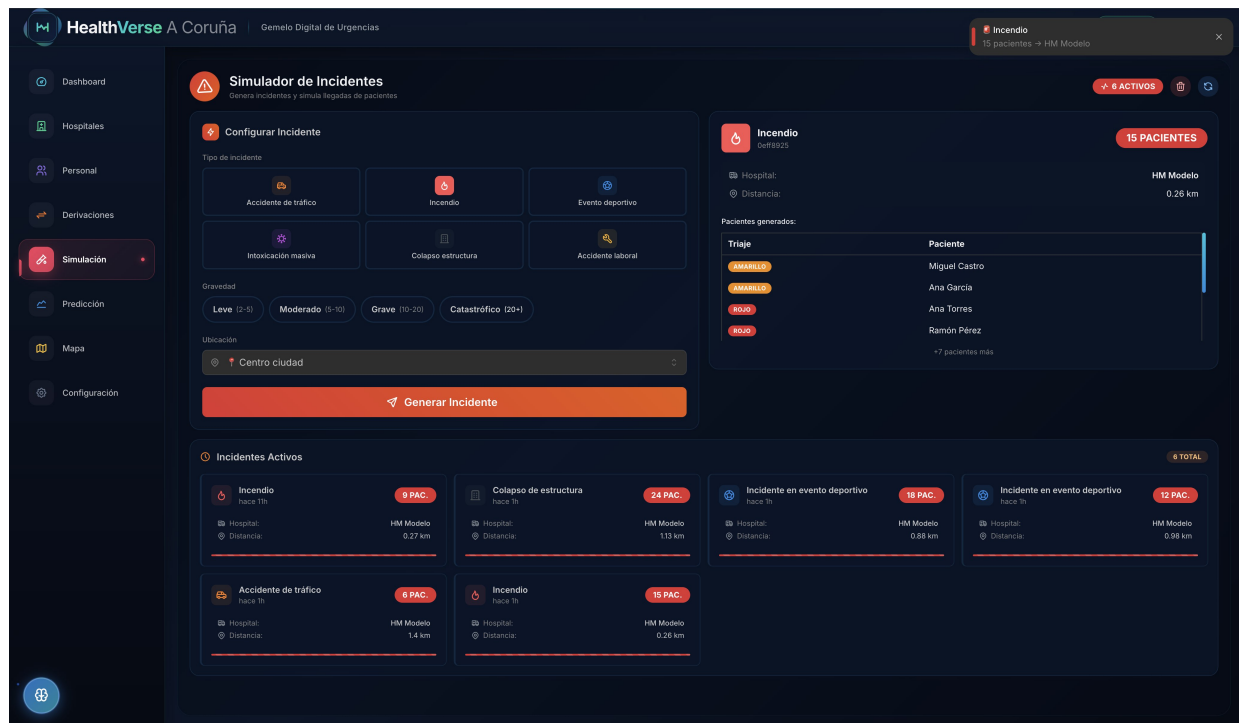


Figura 9. Panel de control de la simulación con ajustes de velocidad y estadísticas en tiempo real.

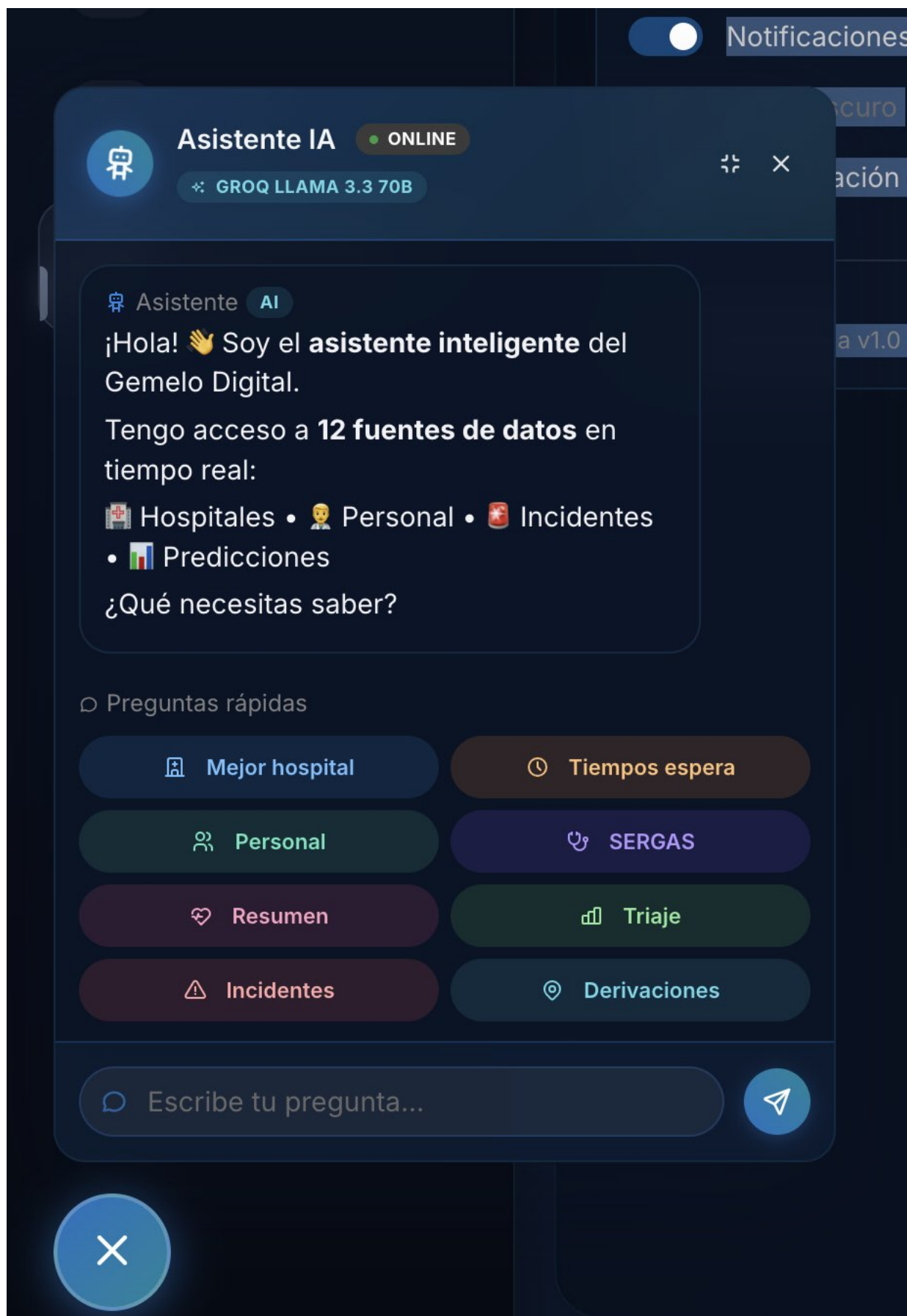


Figura 10. Interfaz del chatbot inteligente basado en Llama 3.3 70B accediendo a datos del sistema en tiempo real.



Figura 11. Panel de predicción de demanda con escenarios what-if configurables y proyecciones por hora.