

Aplicación Web Asistida por Inteligencia Artificial para la Gestión de Indicadores y Generación de Reportes en el Proceso de Referencia y Contrarreferencia de una EPS Indígena

Caso de estudio: DUSAKAWI EPSI

Hector Armando Rincón Farelo
Facultad de Ingeniería
Universidad Popular del Cesar
Valledupar, Colombia
harmandorincon@unicesar.edu.co

Diego Edinson Tique Ramírez
Facultad de Ingeniería
Universidad Popular del Cesar
Valledupar, Colombia
detique@unicesar.edu.co

Adith Bismarck Pérez Orozco
Facultad de Ingeniería
Universidad Popular del Cesar
Valledupar, Colombia
Director del proyecto

Resumen—La fragmentación de la información y la persistencia de procesos manuales en el flujo de referencia y contrarreferencia (RyC) representan una barrera significativa para la continuidad de la atención en las entidades promotoras de salud (EPS) de Colombia. Este artículo presenta el diseño, desarrollo y validación de una aplicación web asistida por inteligencia artificial (IA) orientada a automatizar el cálculo de indicadores de gestión (KPI) y la generación de reportes administrativos en DUSAKAWI EPSI, una entidad promotora de salud indígena. La solución se construyó bajo una arquitectura cliente-servidor de tres capas, integrando un cliente web en Next.js/React, un backend REST en Node.js/Express/TypeScript desplegado en Oracle Cloud Infrastructure, una base de datos relacional PostgreSQL institucional y un motor de generación de análisis en lenguaje natural basado en modelos Gemini (Google AI Studio), con OpenRouter como mecanismo de respaldo (*fallback*). Se documentan de forma exhaustiva los modelos de arquitectura, componentes, despliegue, casos de uso, comportamiento dinámico (actividades y secuencia) y el algoritmo de generación de análisis mediante IA que sustentan el diseño del sistema. Se aplicó un enfoque de investigación mixto, con un modelo de prototipado evolutivo y un proceso de aseguramiento de calidad basado en análisis estático iterativo con SonarQube conforme a ISO/IEC 25002:2024 e ISO/IEC/IEEE 29119-1:2022. Los resultados de validación evidenciaron una reducción del tiempo de generación de indicadores y reportes de un rango de una a dos semanas (proceso manual) a aproximadamente cuatro minutos con el sistema desarrollado, así como una correspondencia prácticamente exacta entre los cálculos automatizados y los cálculos manuales de referencia (diferencia de 0.03 horas, atribuible al redondeo), una reducción del 97.9% en la tasa de error frente al proceso manual sobre una muestra de 200 registros, y un cumplimiento del 98.0% en la validación de calidad de las interpretaciones generadas por el componente de IA mediante una rúbrica de cinco criterios aplicada a diez casos. La evaluación de calidad del software mostró calificaciones A en seguridad, confiabilidad y mantenibilidad, con coberturas de pruebas de 84.8% (backend) y 81.4% (frontend) en la versión final, y la encuesta de satisfacción aplicada a la usuaria principal confirmó una percepción favorable de la herramienta. Los hallazgos sugieren que la integración de modelos de lenguaje de gran escala con arquitecturas de datos híbridas, bajo un principio explícito de minimización de datos, constituye una estrategia viable para fortalecer la gestión de

información en instituciones de salud con recursos tecnológicos limitados.

Palabras Clave—salud indígena, interoperabilidad, inteligencia artificial, gestión hospitalaria, indicadores de desempeño, transformación digital, procesamiento de lenguaje natural, modelos de lenguaje de gran escala, arquitectura de software, ingeniería de requisitos.

Abstract—Information fragmentation and the persistence of manual processes in referral and counter-referral (R&C) workflows represent a significant barrier to continuity of care within healthcare provider entities (EPS) in Colombia. This article presents the design, development, and validation of an artificial intelligence (AI)-assisted web application aimed at automating management indicator (KPI) calculations and administrative report generation at DUSAKAWI EPSI, an indigenous healthcare provider entity. The solution was built under a three-tier client-server architecture, integrating a web client in Next.js/React, a REST backend in Node.js/Express/TypeScript deployed on Oracle Cloud Infrastructure, an institutional PostgreSQL relational database, and a natural language analysis generation engine based on Gemini models (Google AI Studio), with OpenRouter as a fallback mechanism. The models for architecture, components, deployment, use cases, dynamic behavior (activity and sequence diagrams), and the AI-driven analysis generation algorithm underlying the system design are comprehensively documented. A mixed-methods research approach was applied, incorporating an evolutionary prototyping model and a quality assurance process based on iterative static analysis with SonarQube in compliance with ISO/IEC 25002:2024 and ISO/IEC/IEEE 29119-1:2022. Validation results demonstrated a reduction in indicator and report generation time from a range of one to two weeks (manual process) to approximately four minutes using the developed system, as well as a virtually exact alignment between automated calculations and manual reference calculations (0.03-hour difference, attributable to rounding), a 97.9% reduction in the error rate compared to the manual process over a sample of 200 records, and a 98.0% compliance score in the quality validation of AI-generated interpretations evaluated through a five-criterion rubric across ten test cases. Software quality evaluation yielded Grade A ratings in security, reliability, and maintainability, with test coverage reaching 84.8% (backend)

and 81.4% (frontend) in the final version, while the satisfaction survey administered to the primary user confirmed a favorable perception of the tool. The findings suggest that integrating large language models with hybrid data architectures, under an explicit data minimization principle, constitutes a viable strategy to strengthen information management in healthcare institutions operating under limited technological resources.

Keywords—Indigenous health, interoperability, artificial intelligence, hospital management, performance indicators, digital transformation, natural language processing, large language models, software architecture, requirements engineering.

I. INTRODUCCIÓN

La gestión de la información constituye un factor determinante para la sostenibilidad operativa de las entidades promotoras de salud (EPS) en el contexto de la transformación digital del sector sanitario. En Colombia, la Ley Estatutaria 1751 de 2015 [23] establece la continuidad del cuidado como un componente esencial del derecho fundamental a la salud, lo que exige la articulación efectiva entre los distintos niveles de atención mediante procesos de referencia y contrarreferencia (RyC) oportunos y confiables. Sin embargo, la desconexión tecnológica entre dichos niveles continúa siendo una limitación estructural en numerosas instituciones, particularmente en aquellas que atienden poblaciones dispersas o con condiciones de vulnerabilidad, como es el caso de las EPS de régimen subsidiado indígena.

Las EPS indígenas colombianas operan bajo condiciones particularmente exigentes: atienden poblaciones geográficamente dispersas en territorios rurales y de difícil acceso, deben articular sus procesos con la red pública y privada de prestadores en distintos municipios y departamentos, y con frecuencia disponen de equipos técnicos reducidos y presupuestos limitados para la adquisición o el desarrollo de sistemas de información propios. En este escenario, la gestión de indicadores de gestión (*Key Performance Indicators*, KPI) del proceso de RyC —oportunidad de la notificación, aceptación e ingreso de la remisión, efectividad de la atención, cumplimiento de metas institucionales, entre otros— suele depender de hojas de cálculo dispersas, mantenidas manualmente por personal administrativo sin apoyo de automatización, lo que introduce riesgos considerables de inconsistencia, duplicidad y desactualización de la información utilizada para la toma de decisiones directivas y para el reporte a los entes de control.

En este contexto se desarrolló el presente proyecto, orientado a evaluar la manera en que una aplicación web asistida por inteligencia artificial impacta la reducción de errores y el tiempo de generación de indicadores y reportes de referencia y contrarreferencia en DUSAKAWI EPSI, entidad ubicada en Valledupar, Cesar, Colombia. Durante el diagnóstico institucional se evidenció que el proceso de consolidación de indicadores dependía en gran medida de hojas de cálculo dispersas y de la digitación manual de datos, lo que generaba inconsistencias, duplicidad de registros y retrasos considerables en la disponibilidad de la información para la toma de decisiones. El levantamiento inicial, realizado junto con la Coordinadora de Autorizaciones, permitió estimar que el ciclo

completo de consolidación, cálculo y elaboración de un reporte gerencial de indicadores demandaba habitualmente entre una y dos semanas de trabajo manual, con una probabilidad no despreciable de error humano en la transcripción y el cálculo de fórmulas dentro de las hojas de cálculo.

El objetivo general del proyecto consistió en desarrollar una aplicación web asistida por inteligencia artificial para la gestión de indicadores y generación de reportes en el proceso de referencia y contrarreferencia de DUSAKAWI EPSI, con el fin de reducir los errores y el tiempo de procesamiento de la información. De este objetivo general se desprendieron cinco objetivos específicos: (a) caracterizar el proceso actual de generación de indicadores y reportes, identificando tiempos de procesamiento, fuentes de error y limitaciones en la calidad de los datos; (b) diseñar una solución web para la gestión de registros clínicos asistida con IA, estableciendo los requerimientos funcionales, no funcionales y los diagramas de arquitectura; (c) desarrollar una aplicación web asistida por técnicas de procesamiento de lenguaje natural (NLP) que automatice la gestión de indicadores y la generación de reportes; (d) implementar la aplicación en el entorno operativo de la institución; y (e) evaluar el impacto de la solución mediante la medición del tiempo de generación de reportes en comparación con el proceso manual.

La contribución de este trabajo puede resumirse en cuatro aportes concretos. En primer lugar, se documenta el diseño de una arquitectura híbrida que combina el procesamiento local de archivos CSV/XLSX con la consulta directa a una base de datos institucional PostgreSQL, integrando un motor de cálculo de KPI configurable y un mecanismo de generación de reportes en lenguaje natural con redundancia entre proveedores de IA (Gemini y OpenRouter). En segundo lugar, se presenta un conjunto completo de vistas arquitectónicas —arquitectura global, componentes, despliegue, casos de uso, modelo de dominio, actividades, secuencia y especificación algorítmica del módulo de IA— que documentan de manera reproducible las decisiones de diseño adoptadas, un nivel de detalle escasamente reportado en la literatura relacionada. En tercer lugar, se describe un principio explícito de minimización de datos aplicado a la capa de inteligencia artificial, mediante el cual únicamente indicadores agregados y anonimizados son enviados al proveedor de IA, lo que constituye una respuesta operativa concreta a las preocupaciones de privacidad en el uso de modelos de lenguaje de gran escala en salud. Finalmente, se aporta evidencia empírica —derivada de la comparación entre cálculo manual y automatizado, de un proceso de aseguramiento de calidad de software basado en SonarQube y de una encuesta de aceptación tecnológica— sobre el impacto cuantificable de este tipo de soluciones en un entorno real de salud con recursos limitados.

El resto del artículo se organiza de la siguiente manera. La Sección II presenta el estado del arte relacionado con sistemas de información en salud, IA aplicada a procesos clínicos e indicadores de gestión hospitalaria. La Sección III describe la metodología de investigación y desarrollo, así como el entorno tecnológico empleado. La Sección IV detalla la arquitectura

del sistema, incluyendo las vistas estática (componentes, despliegue) y estructural (casos de uso, modelo de dominio) del software. La Sección V describe la implementación de los módulos funcionales y el comportamiento dinámico del sistema mediante diagramas de actividades, de secuencia y la especificación del algoritmo de generación de análisis con IA. La Sección VI presenta los resultados de validación funcional, precisión de indicadores, calidad de software y aceptación tecnológica. La Sección VII discute los hallazgos y, finalmente, la Sección VIII presenta las conclusiones y el trabajo futuro.

II. TRABAJOS RELACIONADOS

A. Sistemas de información y continuidad del cuidado

La literatura reciente coincide en señalar la fragmentación de la información como uno de los principales obstáculos para garantizar la continuidad de la atención en los procesos de referencia y contrarreferencia en América Latina. Huauya-Cancharí y Medina-Sotelo [1] identificaron inconsistencias en las políticas de derivación sanitaria entre distintos niveles de atención en la región, mientras que Barrios y Méndez [2] evidenciaron que la digitalización del proceso de RyC en la red pública de salud chilena mejoró la trazabilidad del paciente y la eficiencia institucional, aunque persistieron dificultades de interoperabilidad. En el ámbito de los sistemas de información estadísticos, Magallón y González [3] destacaron que el uso de mecanismos manuales y sistemas estadísticos básicos compromete la confiabilidad de los indicadores institucionales, mientras que Saleyvis et al. [4] relacionaron directamente la calidad de los indicadores de gestión con la eficiencia organizacional hospitalaria. Estos antecedentes convergen en un diagnóstico común: la ausencia de mecanismos automatizados de consolidación de datos clínicos y administrativos limita tanto la oportunidad de la respuesta asistencial como la confiabilidad de los indicadores utilizados para la toma de decisiones directivas.

B. Inteligencia artificial aplicada a procesos clínicos y administrativos

Respecto a la incorporación de tecnología en el sector salud, Quezada [5] y Kazi et al. [6] argumentan que la automatización mediante inteligencia artificial mitiga errores derivados de la intervención humana y facilita el manejo de grandes volúmenes de información. Desde la perspectiva del procesamiento de lenguaje natural, Zhang y Boulos [7] discuten las oportunidades y desafíos de la IA generativa en medicina, y Liu et al. [8] demostraron que las sugerencias generadas por modelos de lenguaje de gran escala (LLM) pueden optimizar los sistemas de soporte a la decisión clínica. Kuwaiti et al. [9] realizaron una revisión sistemática sobre el papel de la IA en el sector salud, en tanto que Barbieri et al. [10] documentaron la transición desde registros electrónicos básicos hacia sistemas de gestión clínica avanzados, y Prabhu et al. [11] propusieron la integración de IA, historias clínicas electrónicas y dispositivos vestibles para soporte de decisiones centrado en el paciente. En conjunto, estos trabajos sustentan

TABLE I
COMPARACIÓN CON LOS TRABAJOS RELACIONADOS

Trabajo	Proceso RyC	IA gen.	Validación real	Redund. IA
Barrios y Méndez [2]	Sí	No	Sí	No
Saleyvis et al. [4]	No	No	Sí	No
Liu et al. [8]	No	Sí	Sí	No
Prabhu et al. [11]	No	Sí	No	No
Rojas-Villa et al. [20]	Sí	No	Sí	No
Este trabajo	Sí	Sí	Sí	Sí

la viabilidad técnica de emplear modelos de lenguaje de gran escala como capa de interpretación sobre datos clínicos ya estructurados, siempre que se resguarden adecuadamente los principios de privacidad y minimización de datos sensibles.

C. Indicadores de gestión y adopción organizacional de IA

Collins et al. [12] realizaron una revisión sistemática sobre IA en sistemas de información, mientras que Zambrano y Valdiviezo [13] analizaron herramientas de mejora de procesos logísticos en el sector hospitalario ecuatoriano. Esmaeilzadeh [14] identificó los principales retos organizacionales para la adopción de IA a gran escala en instituciones de salud, en tanto que Wang et al. [15] y Sallam [16] exploraron, respectivamente, metodologías de aprendizaje automático en salud y la utilidad de ChatGPT en educación e investigación sanitaria. En el campo específico del procesamiento de retroalimentación textual, Khanna et al. [17] aplicaron NLP al análisis de experiencia del paciente mediante una revisión sistemática, hallazgos consistentes con la revisión estructurada de Secinaro et al. [18] sobre el rol de la IA en salud. Christie et al. [19] documentaron el uso del sistema Epic para investigación en salud, y Rojas-Villa et al. [20] estudiaron la percepción del usuario externo sobre la oportunidad de respuesta en los servicios de referencia y contrarreferencia.

D. Síntesis y brecha de investigación

La Tabla I resume el posicionamiento del presente trabajo frente a los antecedentes revisados, a partir de cuatro dimensiones: si el estudio aborda directamente el proceso de RyC, si incorpora un componente de IA generativa, si el instrumento fue validado en un entorno operativo real con datos institucionales, y si documenta explícitamente un mecanismo de redundancia entre proveedores de IA.

A diferencia de los antecedentes citados, que abordan de manera independiente la digitalización de procesos de RyC, la analítica de indicadores de gestión o la aplicación de NLP en salud, el presente trabajo integra estos tres componentes en una única plataforma operativa, validada en un entorno real con datos institucionales, e incorpora explícitamente un mecanismo de redundancia entre proveedores de IA y un proceso de aseguramiento de calidad de software basado en normas ISO/IEC vigentes, aspectos escasamente reportados de forma conjunta en la literatura revisada.

III. MATERIALES Y MÉTODOS

A. Enfoque y método de investigación

La investigación se desarrolló bajo un enfoque mixto que integró técnicas cualitativas —observación directa y entrevista semiestructurada— y cuantitativas —tiempo de generación de reportes, cantidad de errores detectados, calidad de la información y nivel de satisfacción de la usuaria—. El método adoptado correspondió a la investigación-acción con orientación al desarrollo tecnológico, complementado con un modelo de prototipado evolutivo que permitió ajustar iterativamente los requerimientos del sistema y los componentes de IA a partir de la retroalimentación obtenida en cada fase. Cada iteración del prototipo se sometió a una sesión de validación funcional con la usuaria clave, cuyos hallazgos retroalimentaron directamente el backlog de la iteración siguiente, en un ciclo típico de cuatro a seis semanas por versión.

B. Población, muestra y fuentes de información

Debido a la naturaleza aplicada del proyecto, no se realizó un muestreo probabilístico. Se empleó un muestreo intencional por criterio, seleccionando como informante clave a la Coordinadora de Autorizaciones de DUSAKAWI EPSI, responsable directa de la gestión, seguimiento e interpretación de los indicadores del proceso de RyC. Las fuentes primarias correspondieron a entrevistas, observación directa y actividades de validación con dicha coordinadora; las fuentes secundarias incluyeron documentación institucional, bases de datos internas, normativa vigente del sector salud y literatura científica especializada.

C. Hipótesis de trabajo

Se planteó como hipótesis que la implementación de una aplicación web asistida por IA para la gestión de indicadores y generación de reportes permitiría disminuir el tiempo requerido para la elaboración de reportes y reducir los errores asociados al procesamiento manual de la información en el proceso de RyC de DUSAKAWI EPSI.

D. Requerimientos del sistema

A partir del levantamiento de información se definieron siete requerimientos funcionales principales (Tabla II) y doce requerimientos no funcionales agrupados en las categorías de rendimiento, usabilidad, disponibilidad, seguridad, escalabilidad y compatibilidad.

Las decisiones de diseño arquitectónico y de selección tecnológica se orientaron a satisfacer, en particular, los requerimientos no funcionales de rendimiento (procesamiento en segundos), disponibilidad (despliegue *cloud*), seguridad (control de acceso y minimización de datos) y escalabilidad (incorporación de nuevos indicadores sin modificar la estructura principal de la aplicación). Los requerimientos no funcionales restantes —usabilidad, mantenibilidad y compatibilidad entre navegadores— se verificaron de forma transversal durante las pruebas de aceptación descritas en la Sección VI.

TABLE II
REQUERIMIENTOS FUNCIONALES PRINCIPALES

ID	Requerimiento	Descripción
RF01	Carga de datos	Cargar archivos CSV, conectarse a bases de datos, validar estructura y mostrar errores.
RF02	Limpieza y transformación	Estandarizar formatos, eliminar duplicados y generar alertas de datos inválidos.
RF03	Procesamiento de datos	Clasificar registros de RyC y preparar datos para el cálculo de indicadores.
RF04	Cálculo de indicadores	Calcular y actualizar automáticamente los indicadores tras nuevas cargas.
RF05	Visualización	Mostrar KPIs mediante gráficos, con filtrado por fechas y criterios.
RF06	Generación de reportes con IA	Generar reportes automáticos en lenguaje natural, exportables a Excel.
RF07	Exportación	Exportar indicadores y datos procesados en formatos CSV y Excel.

TABLE III
HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

Capa	Herramientas
Frontend	Next.js (App Router), React, Tailwind CSS, React Context API
Backend	Node.js, Express, TypeScript
Base de datos	PostgreSQL (institucional, solo consulta)
IA generativa	Google AI Studio (Gemini 2.5/3 Flash), OpenRouter (respaldo)
Despliegue	Vercel (frontend), Oracle Cloud Infrastructure (backend)
Calidad de software	SonarQube, Playwright (pruebas automatizadas)
Control de versiones	Git
Seguridad	JSON Web Token (JWT), API Keys con <i>hash</i> irreversible, HTTPS/TLS

E. Herramientas y entorno tecnológico

La Tabla III resume las principales herramientas y tecnologías empleadas en el desarrollo, agrupadas por capa arquitectónica. La selección de cada componente respondió a criterios de madurez del ecosistema, disponibilidad de soporte gratuito para instituciones con presupuesto limitado y facilidad de despliegue sobre infraestructura ya disponible en la institución.

F. Procedimiento de validación

La validación de la solución se estructuró en cinco frentes: (i) pruebas funcionales de cada módulo; (ii) pruebas de calidad y consistencia de los datos procesados; (iii) comparación entre el cálculo automatizado de indicadores y el cálculo manual de referencia, realizada en conjunto con el Área de Calidad de la institución; (iv) pruebas de usabilidad y una encuesta de satisfacción y aceptación tecnológica basada en escala de Likert de cinco niveles aplicada a la usuaria principal; y (v) evaluación de la calidad interna del software mediante análisis estático iterativo con SonarQube. Cada frente de validación se relaciona directamente con uno o más de los objetivos específicos definidos en la Sección I, lo que permitió trazar una correspondencia explícita entre requerimientos, artefactos de diseño y evidencia de cumplimiento.

IV. ARQUITECTURA DEL SISTEMA

A. Visión general

La arquitectura fue diseñada bajo un modelo cliente-servidor organizado en tres capas: (a) cliente web, (b) backend y servicios API, y (c) base de datos y procesamiento. Este diseño permitió desacoplar el procesamiento de información, la lógica de negocio y la visualización, favoreciendo la escalabilidad y el mantenimiento independiente de cada componente. La solución fue concebida como un módulo integrado dentro del ecosistema tecnológico corporativo ya existente en DUSAKAWI EPSI, por lo que la autenticación de usuarios no es gestionada por el módulo, sino delegada al sistema institucional principal. Para documentar de manera trazable las decisiones arquitectónicas, se elaboró un conjunto de vistas complementarias —arquitectura global, componentes, despliegue, casos de uso y modelo de dominio— que se describen en las siguientes subsecciones.

B. Arquitectura global de la solución

La Fig. 1 presenta la vista global de la solución. El usuario institucional (Coordinadora de Autorizaciones) interactúa exclusivamente con el cliente web, desplegado sobre Vercel Cloud Platform, el cual opera bajo dos modos de operación complementarios: (1) procesamiento local de archivos CSV/XLSX cargados por el usuario, gestionado íntegramente en el navegador mediante el componente *CSV Uploader*; y (2) consulta de información institucional histórica mediante el componente *Backend Connector*, que se comunica vía HTTPS con la infraestructura institucional de DUSAKAWI EPSI. En ambos modos, el estado de la aplicación se centraliza en *DataContext*, un contenedor de estado en memoria que alimenta de manera uniforme al motor de cálculo de indicadores (*KPI Calculator* y *KPI Engine*), al módulo de visualización (*Dashboard* y *Filtros*) y al exportador de reportes (*XLSX Exporter*).

Dentro de la infraestructura institucional, el *Backend Express* expone los servicios REST `GET /api/referrals` y `GET /api/kpi-results`, protegidos mediante un *Authentication Middleware* basado en *API Key* y una capa adicional de *JWT Validation*. La lógica de negocio (*Business Logic Layer*) coordina el acceso de solo consulta a la base de datos corporativa PostgreSQL, el registro de eventos en *Application Logs* y la invocación del *AI Analysis Service*, el cual delega en un *Prompt Builder* la construcción de las instrucciones enviadas al *LLM Client*. Este último gestiona el consumo de los modelos de IA externos —Google AI Studio Gemini como proveedor principal y OpenRouter como mecanismo de respaldo (*fallback*)— garantizando la continuidad operativa del módulo de generación de análisis ante fallos o límites de uso del proveedor principal. El diagrama resalta expresamente que la información clínica y administrativa nunca abandona la infraestructura institucional: únicamente indicadores agregados y minimizados son enviados a los servicios de IA, y toda operación queda registrada mediante *logs* de auditoría en formato JSON con fines de trazabilidad.

C. Diagrama de componentes

La Fig. 2 descompone la arquitectura global en los componentes de software que la integran, organizados en tres paquetes principales: el cliente Next.js desplegado en Vercel, el backend Express desplegado en la infraestructura institucional, y el Sistema Corporativo DUSAKAWI EPSI, responsable de la autenticación mediante un *JWT Manager* con tokens de una hora de vigencia.

En el cliente, la capa de *Presentación* agrupa los componentes visuales *Sidebar*, *FileUploader*, *KpiDetailModal*, *GlobalFilters* y *KpiTechnicalSheetModal*, que a su vez consumen la capa de *Aplicación* (*CounterReferenceAnalyzer*, *DashboardView*, *ReferralMonitoring* y el *hook useKpiExport*). Toda esta capa se apoya en *DataContext* como fuente única de estado, el cual a su vez delega en la capa de *Dominio* —compuesta por *kpi-calculator*, *kpi-engine*, *kpi-config*, *segment-by-time*, *types*, *DTOs*, *Validators* y *Mappers*— la lógica pura de cálculo de indicadores, independiente de la fuente de datos. La capa de *Integración Local* (*CSVParser*, *external-data*) procesa los archivos cargados por el usuario, mientras que *APIClient* centraliza las llamadas HTTP hacia el backend. En el servidor, los servicios *KPIService*, *ReferralService*, *AIAnalysisService* y *DatabaseService* implementan la lógica de negocio sobre la API REST, apoyados por *PromptBuilder* y *LogService*; la capa de *Seguridad* (*JWT Validation*) protege ambos *endpoints* expuestos, y *LLMClient* gestiona el consumo de Gemini y OpenRouter con conmutación automática ante fallas (*failover*).

D. Diagrama de despliegue

La Fig. 3 describe la topología física de la solución. El frontend Next.js se despliega en Vercel Cloud Platform y se comunica mediante HTTPS con el navegador del usuario, el cual también puede procesar localmente archivos CSV/XLSX sin necesidad de conexión al backend. Las solicitudes que requieren datos institucionales viajan por Internet hacia la infraestructura institucional DUSAKAWI EPSI, delimitada como una intranet corporativa que aloja una *Virtual Cloud Network* (VCN) con el servidor backend Express.js y el servidor PostgreSQL.

Un aspecto crítico de esta topología es que el servidor PostgreSQL no es accesible directamente desde Internet: todo acceso a la información institucional se realiza exclusivamente a través del backend Express, que valida el JWT y ejecuta las consultas SQL de solo lectura sobre el motor Node.js Runtime. Este mismo entorno de ejecución conecta con el *AI Analysis Service*, que registra sus operaciones en *Application Logs* y consume los modelos de IA (Gemini como principal, OpenRouter como respaldo) mediante llamadas salientes autenticadas por *API Key*. Esta segregación de red garantiza que la base de datos corporativa permanezca dentro del perímetro

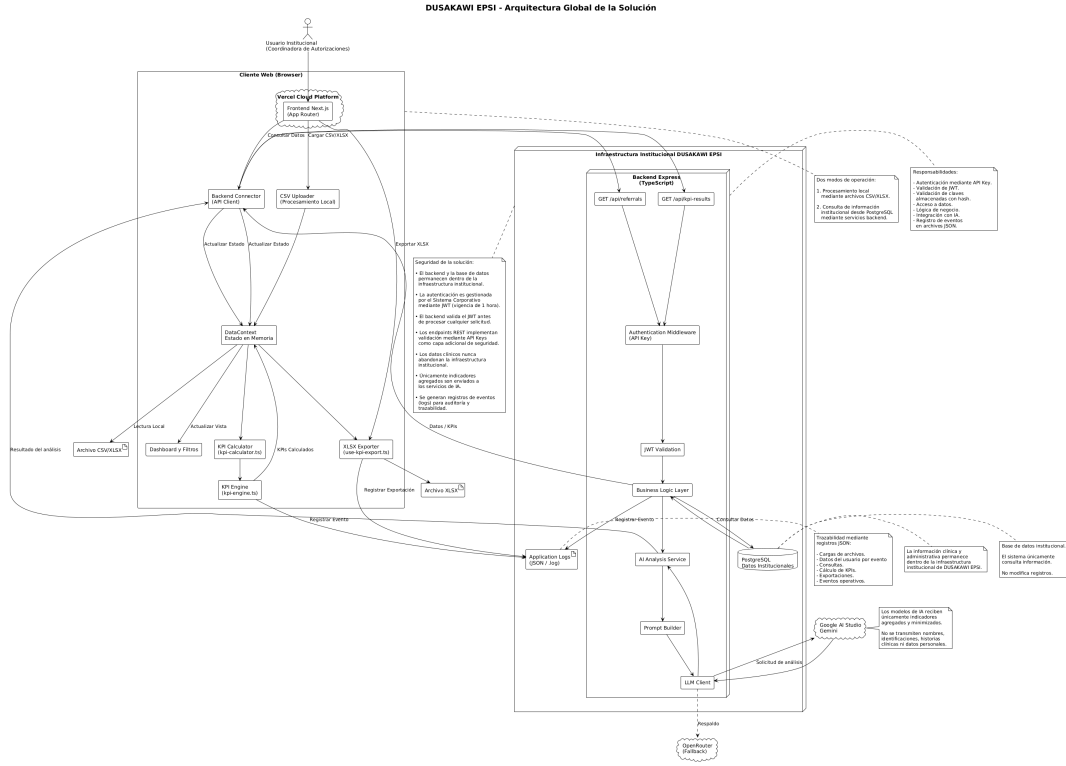


Fig. 1. Arquitectura global de la solución DUSAKAWI EPSI: interacción entre el cliente web, la infraestructura institucional y los proveedores externos de inteligencia artificial.

institucional en todo momento, reduciendo la superficie de exposición de los datos clínicos y administrativos.

E. Casos de uso del sistema

La Fig. 4 presenta el diagrama de casos de uso correspondiente al actor principal del sistema, la Coordinadora de Autorizaciones. El diagrama documenta dos opciones de entrada de datos —Opción 1: carga de archivo CSV/XLSX con procesamiento local, y Opción 2: consulta institucional al backend conectado a PostgreSQL— ambas relacionadas mediante una dependencia <<include>> con el caso de uso *Analizar remisiones y calcular KPIs*.

A partir de los KPI calculados, el caso de uso *Generar análisis con IA (mensual/trimestral)* extiende opcionalmente el flujo principal mediante una relación <<extend>> hacia *Exportar ficha técnica XLSX*, reflejando que la generación de análisis en lenguaje natural es un paso complementario y no obligatorio del proceso de exportación. Adicionalmente, el diagrama documenta los casos de uso *Analizar tendencias para la toma de decisiones* —que incluye *Filtrar por año, periodo (mensual, trimestral, semestral) y territorio*—, *Revisar alertas de cumplimiento de metas*, *Consultar detalle técnico de KPI*, *Monitorear referencia y contrarreferencia* y *Exportar matriz de indicadores (XLSX)*. Esta vista funcional fue el insumo principal para la definición de los requerimientos funcionales de la Tabla II y para la priorización de las iteraciones del prototipado evolutivo.

F. Modelo de dominio

La Fig. 5 integra el modelo de dominio clínico con la arquitectura de infraestructura, evidenciando la trazabilidad entre las entidades de negocio y los componentes de software que las manipulan. En el dominio clínico, un *Paciente* genera una o varias *Referencias*, cada una caracterizada por su estado y su código CIE-10; estas referencias alimentan los objetos *KPIResultado* (valor numérico y bandera de cumplimiento de meta), los cuales son evaluados en el contexto de una *IPS* (identificada por NIT y su condición de adscripción).

En el frontend, el objeto de valor *FiltroKPI* (año, periodo, territorio) configura el *Boundary DashboardUI*, que solicita datos al servicio *DataContext*; este, a su vez, consulta al servicio *APIClient* mediante *fetch* bajo el modo híbrido ya descrito. En el backend, el controlador *BackendAPI* expone los tres *endpoints* REST del sistema y delega en la capa de seguridad *JWT Validation* antes de invocar los servicios *AIAnalysisService*, *ReferralService* y *KPIService*, los cuales acceden en modo de solo consulta a PostgreSQL y registran cada operación en *Application Logs*. El servicio *AIAnalysisService* delega en *PromptBuilder* la construcción del contexto textual —explícitamente advertido en el diagrama como libre de datos personales— antes de invocar a *LLMClient*, que gestiona el consumo de Gemini como proveedor principal y OpenRouter como respaldo.

DUSAKAWI EPSI - Casos de Uso de la Coordinadora de Autorizaciones

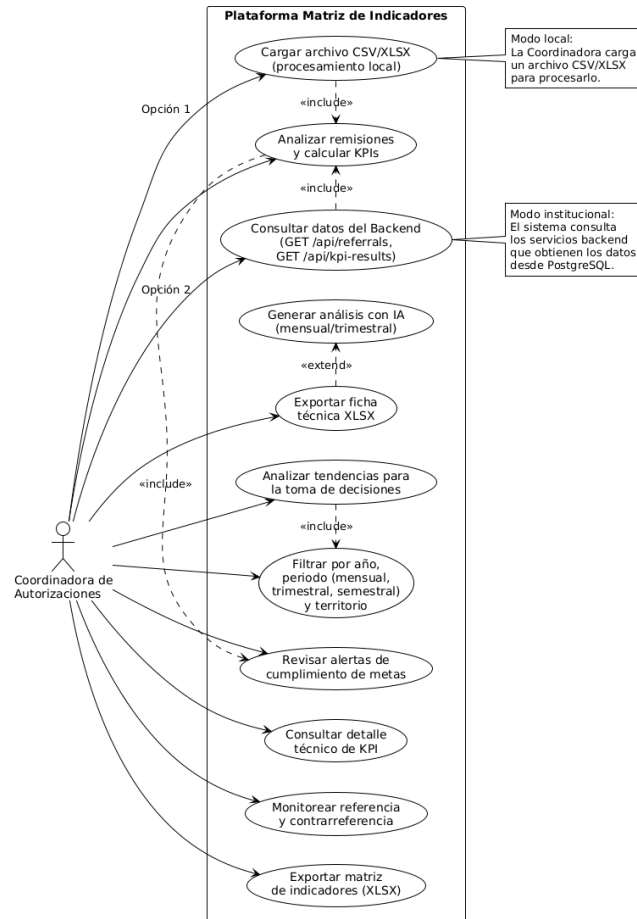


Fig. 4. Diagrama de casos de uso de la Coordinadora de Autorizaciones sobre la Plataforma Matriz de Indicadores.

de respuesta, costo y disponibilidad de capa gratuita para el desarrollo y las pruebas. OpenRouter se incorporó como proveedor secundario debido a que ofrece acceso unificado a múltiples modelos (Llama, DeepSeek, entre otros) mediante una única interfaz API, permitiendo implementar un mecanismo de contingencia sin modificar la lógica de la aplicación: ante errores de conexión, indisponibilidad temporal o límites de uso del proveedor principal, las solicitudes se redirigen automáticamente hacia el proveedor de respaldo, preservando la continuidad operativa de la generación de reportes. Este mecanismo de *failover* se documenta en detalle, a nivel algorítmico, en la Sección V-C.

K. Seguridad, privacidad y trazabilidad

El diseño de seguridad se alineó con la Ley 1581 de 2012 sobre protección de datos personales [21]. Todas las comunicaciones entre componentes emplean HTTPS/TLS. El acceso a los servicios backend se controla mediante *API keys* almacenadas con funciones criptográficas de *hash* irreversible. El frontend se despliega en infraestructura cloud (Vercel), mientras que el backend y la base de datos permanecen dentro

de la intranet institucional, reduciendo la exposición directa de los datos clínicos y administrativos.

Un aspecto particular del diseño es el principio de minimización de datos aplicado a la capa de IA: el sistema procesa inicialmente los registros institucionales y calcula los indicadores de gestión mediante algoritmos locales; únicamente los resultados agregados y estadísticos —sin nombres, identificaciones ni historias clínicas— son enviados al modelo Gemini para la generación de análisis en lenguaje natural. De esta manera, los servicios de IA no reciben en ningún momento información identificable de pacientes, principio que se documenta de manera explícita en la especificación algorítmica presentada en la Fig. 11.

V. IMPLEMENTACIÓN

A. Diagrama de actividades

La Fig. 9 modela el flujo operativo completo del sistema, desde el acceso del usuario hasta la descarga del reporte final, integrando las dos fuentes de datos posibles (carga local de CSV/XLSX o consulta al backend institucional).

Cuando el usuario opta por el modo local, el archivo se valida en cuanto a formato y estructura; si es válido,

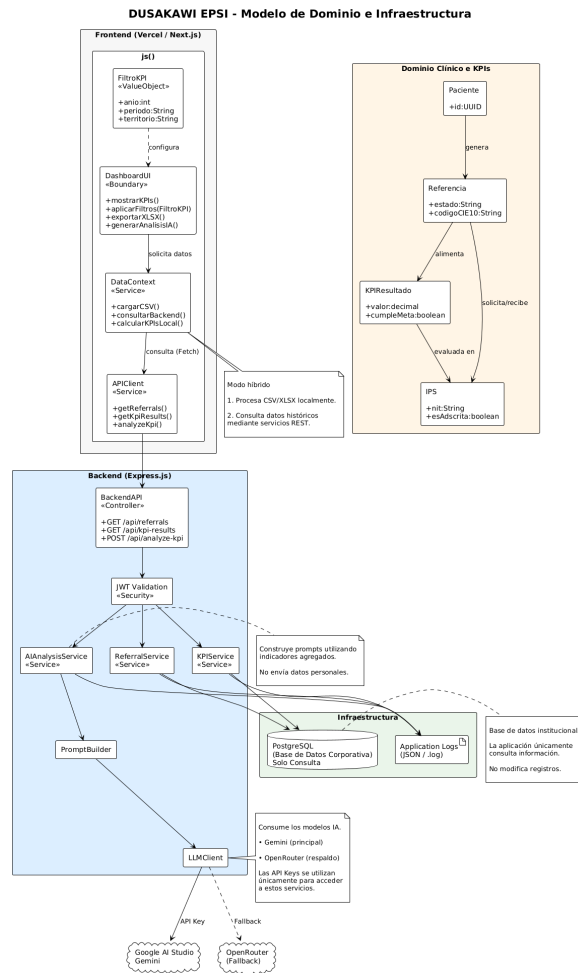


Fig. 5. Modelo de dominio e infraestructura: relación entre las entidades clínicas (Paciente, Referencia, KPIResultado, IPS) y los componentes de software del frontend y el backend.



Fig. 6. Módulo de carga de información por rango de fechas o archivo CSV/XLSX.

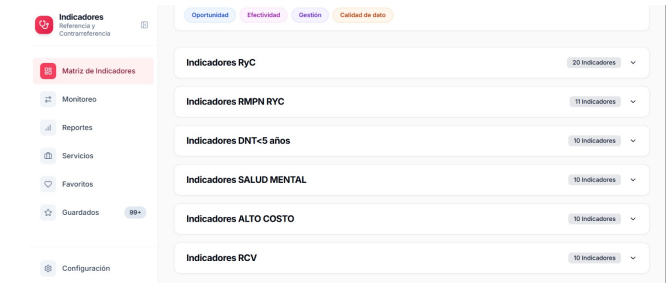


Fig. 7. Categorización funcional de los 71 indicadores del sistema por dominio de gestión.

DataContext lo almacena, lo parsea a un arreglo de filas tipadas y normaliza fechas y campos de riesgo. Cuando el usuario opta por consultar el backend, el frontend construye los parámetros de consulta a partir de los filtros seleccionados y ejecuta GET /api/referrals; el backend valida el JWT y, si es válido, consulta PostgreSQL y retorna los resultados, que el frontend mapea al esquema interno. Ambos flujos convergen en un nodo de sincronización común, a partir del cual

el sistema enriquece los datos con el catálogo CIE-10/RIAS, calcula los KPI mediante los módulos kpi-calculator y kpi-engine, actualiza DataContext y renderiza el dashboard junto con las alertas de cumplimiento de metas. Si el usuario decide generar un análisis con IA, el sistema segmenta los indicadores por mes, prepara un payload minimizado, lo envía mediante POST /api/analyze-kpi, y

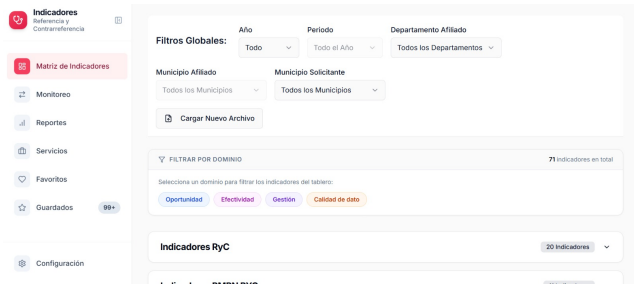


Fig. 8. Filtros globales y clasificación por dominio (oportunidad, efectividad, gestión, calidad de dato).

—tras la validación del JWT y la construcción del *prompt*— consulta a Gemini (con OpenRouter como respaldo) antes de incorporar el análisis generado al reporte. El flujo finaliza, en cualquiera de los dos casos, con la generación del archivo XLSX y su descarga por parte del usuario.

B. Diagrama de secuencia

La Fig. 10 detalla, a nivel de interacción entre objetos, el escenario de consulta trimestral con generación de análisis mediante IA, diferenciando explícitamente los dos modos de operación del sistema.

En la Opción 1 (procesamiento local), el usuario selecciona año y trimestre en `GlobalFilters`, que notifica a `DashboardPage`; este obtiene el conjunto de datos en memoria desde `DataContext` y ejecuta `calculateKpisFromCsv()`, el cual recorre en un ciclo cada indicador invocando a `kpi-calculator.calculateKpi()` hasta obtener el conjunto completo de KPI calculados, que finalmente se renderizan en el *dashboard*. En la Opción 2 (consulta institucional), `APIClient` construye los parámetros de consulta y ejecuta `GET /api/referrals` contra el *Backend Express*; si el JWT es válido, el backend consulta PostgreSQL y retorna el conjunto de datos, que el frontend usa para actualizar el *dashboard*; si el JWT es inválido, el backend retorna 401 Unauthorized y el frontend informa al usuario que su sesión ha expirado. De manera opcional, el frontend puede consultar KPI ya precalculados mediante `GET /api/kpi-results`, siguiendo el mismo esquema de validación de JWT. La generación de análisis con IA se modela como un fragmento opcional adicional: el usuario solicita el análisis, el backend construye el *prompt*, anonimiza los indicadores y solicita el análisis a Gemini; si Gemini no está disponible, el backend reintenta automáticamente mediante OpenRouter antes de retornar el resultado al frontend, que lo integra al reporte analítico final. El diagrama documenta también el caso de borde en que no existen resultados para los filtros seleccionados, mostrando un mensaje explícito al usuario en lugar de un *dashboard* vacío.

C. Especificación del algoritmo de generación del análisis de KPI mediante IA

La Fig. 11 presenta la especificación algorítmica completa del módulo de exportación con análisis de IA (`use-kpi-export`), organizada en cinco bloques de responsabilidad: cliente, *Backend Express*, *LLM Client*, validación de respuesta y post-procesamiento en el cliente.

El algoritmo inicia cuando el usuario selecciona un KPI para exportar. Según el origen de los datos (CSV local u obtención mediante `GET /api/kpi-results` con validación de JWT), el sistema toma el indicador seleccionado y los datos filtrados, segmenta los valores por mes y calcula el estado de cumplimiento (*Cumple/No Cumple*) frente a la meta institucional, para finalmente construir un *payload* con los indicadores agregados. Si existen datos suficientes, el *Backend Express* recibe la solicitud mediante `POST /api/analyze-kpi`, valida el JWT, construye el *prompt* y valida la estructura del *payload* antes de invocarlo al *LLM Client*. Este último envía la solicitud a Gemini; si el modelo responde correctamente, recibe el análisis directamente, mientras que ante un error o un *timeout* reintenta automáticamente mediante OpenRouter como mecanismo de respaldo, garantizando la continuidad operativa. Una vez recibido el análisis, el backend valida que la respuesta corresponda a una estructura JSON válida —retornando un error controlado en caso contrario— y la entrega al cliente, que mapea el análisis a nivel mensual y trimestral y lo integra con los indicadores agregados antes de generar el archivo XLSX final. Si no existen datos suficientes, o si el análisis no pudo obtenerse, el sistema continúa el flujo de exportación sin el bloque de análisis narrativo, evitando que una falla del proveedor de IA bloquee la generación del reporte. Esta especificación constituye el nivel de detalle más fino de la arquitectura dinámica del sistema y fue la referencia directa para la implementación del módulo descrito a continuación.

D. Procesamiento y calidad del dato

El módulo de ingesta admite archivos CSV/XLSX o consulta directa a PostgreSQL, y ejecuta automáticamente un *pipeline* de (a) validación de estructura, (b) limpieza de datos inconsistentes, (c) eliminación de duplicados, (d) normalización de formatos (particularmente de fecha/hora), (e) clasificación de registros por tipo de remisión y (f) preparación de variables para el cálculo de indicadores. El diagnóstico inicial había identificado, sobre archivos históricos, registros incompletos, valores nulos, duplicidad, heterogeneidad de formatos de fecha, errores tipográficos e información desactualizada; estas observaciones definieron las reglas de validación implementadas y se reflejan directamente en las ramas de decisión *¿Formato válido?* y *¿Existen datos?* del diagrama de actividades de la Fig. 9.

E. Motor de cálculo de indicadores

Se implementó un módulo de configuración de indicadores que permite crear, consultar, editar y eliminar KPI mediante interfaz gráfica, sin modificar el código fuente de la aplicación

DUSAKAWI EPSI - Diagrama de Actividades
Carga de Datos -> Procesamiento KPI -> IA -> Exportación

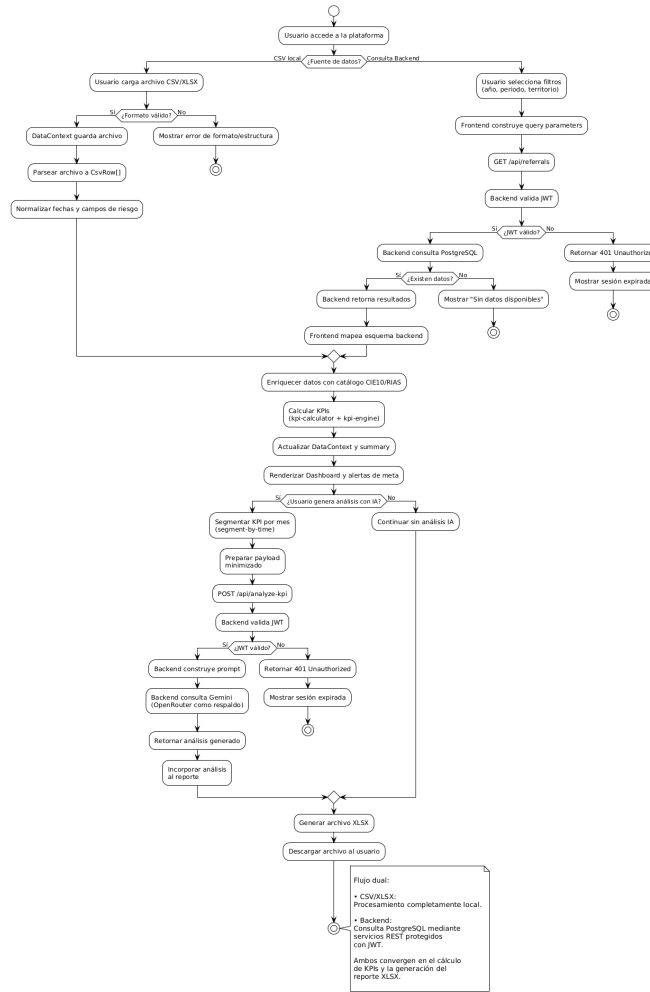


Fig. 9. Diagrama de actividades del flujo Carga de Datos → Procesamiento KPI → IA → Exportación.

(Fig. 14). Cada indicador se define en tres pestañas: datos básicos (nombre, categoría RIAS, dominio, meta sugerida y frecuencia; Fig. 15), fórmula (variables con filtros de capa 1 y expresión matemática de capa 2 sobre dichas variables, con vista previa en tiempo real; Fig. 16) y descripción (grupo/categoría, objetivo, y descripción del numerador y del denominador; Fig. 17). Esta separación por capas —variables filtradas y fórmula sobre variables— permite construir indicadores tipo razón, porcentaje o promedio temporal sin necesidad de intervención del equipo de desarrollo, incrementando la mantenibilidad y escalabilidad del sistema ante nuevos requerimientos institucionales, y materializa directamente el requerimiento no funcional de escalabilidad enunciado en la Sección III-D.

Adicionalmente, se implementaron mecanismos de control para las modificaciones realizadas sobre las fórmulas de los indicadores. El módulo restringe estas operaciones de acuerdo con los permisos del usuario y registra las acciones realizadas mediante mecanismos de auditoría. Asimismo, se incorporó un historial de versiones que permite conservar las config-

uraciones anteriores de las fórmulas y restaurar una versión previa cuando se identifica una modificación incorrecta, como se observa en las Fig. 12 y 13.

Los resultados calculados se presentan mediante tarjetas KPI con codificación visual del estado frente a la meta (cumple/alerta), como se observa para los indicadores del dominio RyC en la Fig. 18, y mediante un módulo de monitoreo operativo que agrega tiempos de notificación, aceptación e ingreso (Fig. 19), tendencias mensuales del volumen de remisiones (Fig. 20), distribución por especialidad y tipo de traslado (Fig. 21) y distribución geográfica por municipio de origen del afiliado (Fig. 22).

F. Generación de reportes asistida por IA

El flujo de generación de reportes envía las métricas ya agregadas hacia el modelo Gemini mediante llamadas API estructuradas con plantillas de *prompt engineering* diseñadas para estandarizar la interpretación de los indicadores institucionales, en correspondencia directa con el algoritmo descrito en la Sección V-C. Cuatro tipos de reporte exportable en

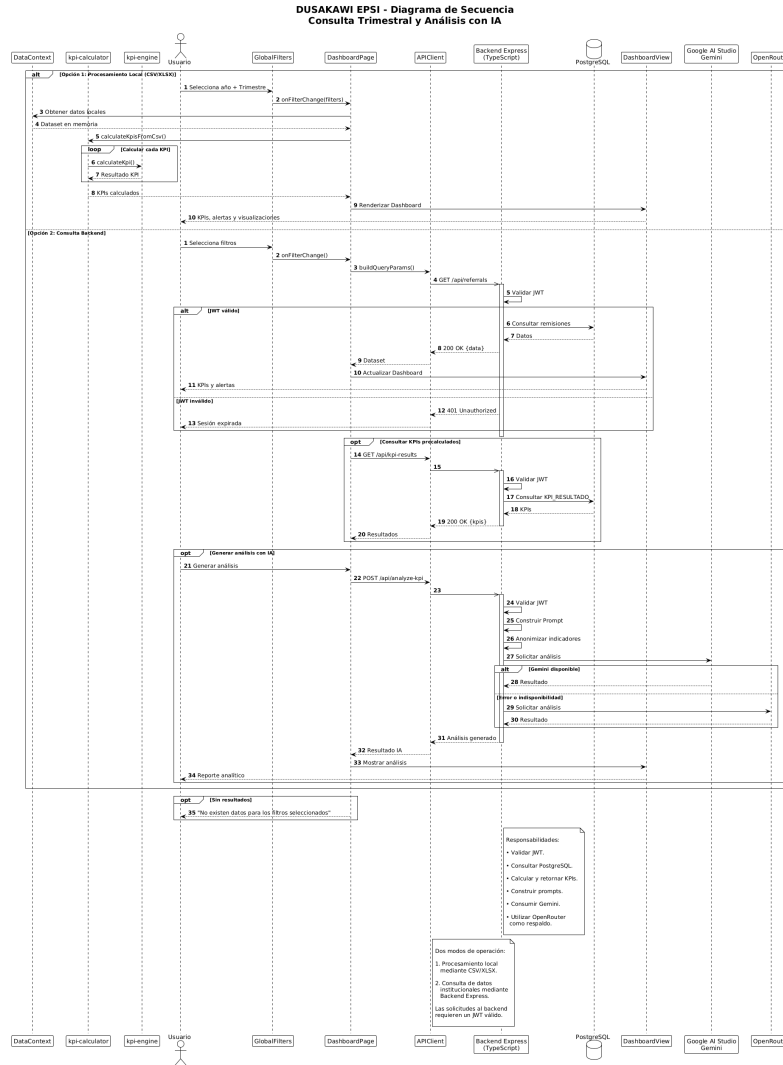


Fig. 10. Diagrama de secuencia: consulta trimestral de indicadores y generación de análisis con IA, en modo local y en modo institucional.

formato XLSX están disponibles (Fig. 23): matriz consolidada de indicadores, ficha técnica individual por KPI, guía técnica (diccionario de indicadores con fórmula, fuente de datos y meta) y bitácora completa de remisiones. Ante fallos de conexión, límites de uso o indisponibilidad del proveedor principal, el sistema redirige automáticamente la solicitud hacia OpenRouter, garantizando la continuidad del servicio sin intervención del usuario.

VI. RESULTADOS

A. Trazabilidad entre diseño e implementación

Como síntesis de las secciones IV y V, la Tabla V presenta la correspondencia entre cada requerimiento funcional, el artefacto de diseño que lo especifica y el módulo de implementación que lo materializa, evidenciando la cobertura completa del ciclo de diseño-implementación del sistema.

TABLE V
TRAZABILIDAD ENTRE REQUERIMIENTOS, DISEÑO E IMPLEMENTACIÓN

Req.	Artefacto de diseño	Módulo implementado
RF01	Casos de uso (Fig. 4), Actividades (Fig. 9)	FileUploader, CSVParser
RF02–	Actividades (Fig. 9)	DataContext, Validators,
RF03		Mappers
RF04	Modelo de dominio (Fig. 5)	kpi-calculator, kpi-engine
RF05	Componentes (Fig. 2)	DashboardView, GlobalFilters
RF06	Secuencia (Fig. 10), Algoritmo (Fig. 11)	AIAnalysisService, PromptBuilder, LLMClient
RF07	Casos de uso (Fig. 4)	useKpiExport

B. Calidad de los datos procesados

La evaluación de los criterios de calidad sobre los archivos procesados (Tabla VI) confirmó el correcto funcionamiento de

```

graph TD
    Start([Inicio]) --> Client[Cliente (tele-kt+experts)]
    Client --> SelectKPH[Usuario selecciona KPH para respuesta]
    SelectKPH --> C1{C1}
    C1 -- "Clasificación de KPH" --> BackEnd[Backend]
    C1 -- "Otras disponibles en SAP/Consult" --> End([Fin])
    BackEnd --> GetResults[GET nKPH+results]
    GetResults --> ValidJRT{Backend valida JRT}
    ValidJRT -- "No" --> GetIndicator[Obtener indicadores desde Backend]
    ValidJRT -- "Si" --> ShowScreen[Mostrar screen expreso]
    ShowScreen --> End
    GetIndicator --> C2{C2}
    C2 -- "Si" --> SelectData[Tomar KPH selecciona y datos filtrados]
    C2 -- "No" --> End
    SelectData --> Segment[Segmentar indicadores por mes]
    Segment --> CalcRisk[Calcular riesgo: Cumpli No Cumpli]
    CalcRisk --> Control[Controlar puntos con indicadores agregados]
    Control --> C3{C3}
    C3 -- "Existen datos sustraídos" --> End
    C3 -- "No" --> BackEndExpress[Backend Express]
    BackEndExpress --> POST[POST /api/analyze-kph]
    POST --> ValidJRT2{Validar JRT}
    ValidJRT2 -- "No" --> ControlPrompts[Control Prompts]
    ControlPrompts --> ValidStructure[Validar estructura del payload]
    ValidStructure --> LLMClient[LLM Client]
    LLMClient --> SendToGemini[Enviar solicitud a Gemini]
    SendToGemini -- "Si" --> GetResponse[Recibir análisis]
    SendToGemini -- "No" --> GetResponse
    GetResponse --> C4{C4}
    C4 -- "Si" --> GetResponse
    C4 -- "No" --> GetResponse
    GetResponse --> C5{C5}
    C5 -- "Si" --> GetResponse
    C5 -- "No" --> GetResponse
    GetResponse --> C6{C6}
    C6 -- "Si" --> GetResponse
    C6 -- "No" --> GetResponse
    GetResponse --> C7{C7}
    C7 -- "Si" --> GetResponse
    C7 -- "No" --> GetResponse
    GetResponse --> C8{C8}
    C8 -- "Si" --> GetResponse
    C8 -- "No" --> GetResponse
    GetResponse --> C9{C9}
    C9 -- "Si" --> GetResponse
    C9 -- "No" --> GetResponse
    GetResponse --> C10{C10}
    C10 -- "Si" --> GetResponse
    C10 -- "No" --> GetResponse
    GetResponse --> C11{C11}
    C11 -- "Si" --> GetResponse
    C11 -- "No" --> GetResponse
    GetResponse --> C12{C12}
    C12 -- "Si" --> GetResponse
    C12 -- "No" --> GetResponse
    GetResponse --> C13{C13}
    C13 -- "Si" --> GetResponse
    C13 -- "No" --> GetResponse
    GetResponse --> C14{C14}
    C14 -- "Si" --> GetResponse
    C14 -- "No" --> GetResponse
    GetResponse --> C15{C15}
    C15 -- "Si" --> GetResponse
    C15 -- "No" --> GetResponse
    GetResponse --> C16{C16}
    C16 -- "Si" --> GetResponse
    C16 -- "No" --> GetResponse
    GetResponse --> C17{C17}
    C17 -- "Si" --> GetResponse
    C17 -- "No" --> GetResponse
    GetResponse --> C18{C18}
    C18 -- "Si" --> GetResponse
    C18 -- "No" --> GetResponse
    GetResponse --> C19{C19}
    C19 -- "Si" --> GetResponse
    C19 -- "No" --> GetResponse
    GetResponse --> C20{C20}
    C20 -- "Si" --> GetResponse
    C20 -- "No" --> GetResponse
    GetResponse --> C21{C21}
    C21 -- "Si" --> GetResponse
    C21 -- "No" --> GetResponse
    GetResponse --> C22{C22}
    C22 -- "Si" --> GetResponse
    C22 -- "No" --> GetResponse
    GetResponse --> C23{C23}
    C23 -- "Si" --> GetResponse
    C23 -- "No" --> GetResponse
    GetResponse --> C24{C24}
    C24 -- "Si" --> GetResponse
    C24 -- "No" --> GetResponse
    GetResponse --> C25{C25}
    C25 -- "Si" --> GetResponse
    C25 -- "No" --> GetResponse
    GetResponse --> C26{C26}
    C26 -- "Si" --> GetResponse
    C26 -- "No" --> GetResponse
    GetResponse --> C27{C27}
    C27 -- "Si" --> GetResponse
    C27 -- "No" --> GetResponse
    GetResponse --> C28{C28}
    C28 -- "Si" --> GetResponse
    C28 -- "No" --> GetResponse
    GetResponse --> C29{C29}
    C29 -- "Si" --> GetResponse
    C29 -- "No" --> GetResponse
    GetResponse --> C30{C30}
    C30 -- "Si" --> GetResponse
    C30 -- "No" --> GetResponse
    GetResponse --> C31{C31}
    C31 -- "Si" --> GetResponse
    C31 -- "No" --> GetResponse
    GetResponse --> C32{C32}
    C32 -- "Si" --> GetResponse
    C32 -- "No" --> GetResponse
    GetResponse --> C33{C33}
    C33 -- "Si" --> GetResponse
    C33 -- "No" --> GetResponse
    GetResponse --> C34{C34}
    C34 -- "Si" --> GetResponse
    C34 -- "No" --> GetResponse
    GetResponse --> C35{C35}
    C35 -- "Si" --> GetResponse
    C35 -- "No" --> GetResponse
    GetResponse --> C36{C36}
    C36 -- "Si" --> GetResponse
    C36 -- "No" --> GetResponse
    GetResponse --> C37{C37}
    C37 -- "Si" --> GetResponse
    C37 -- "No" --> GetResponse
    GetResponse --> C38{C38}
    C38 -- "Si" --> GetResponse
    C38 -- "No" --> GetResponse
    GetResponse --> C39{C39}
    C39 -- "Si" --> GetResponse
    C39 -- "No" --> GetResponse
    GetResponse --> C40{C40}
    C40 -- "Si" --> GetResponse
    C40 -- "No" --> GetResponse
    GetResponse --> C41{C41}
    C41 -- "Si" --> GetResponse
    C41 -- "No" --> GetResponse
    GetResponse --> C42{C42}
    C42 -- "Si" --> GetResponse
    C42 -- "No" --> GetResponse
    GetResponse --> C43{C43}
    C43 -- "Si" --> GetResponse
    C43 -- "No" --> GetResponse
    GetResponse --> C44{C44}
    C44 -- "Si" --> GetResponse
    C44 -- "No" --> GetResponse
    GetResponse --> C45{C45}
    C45 -- "Si" --> GetResponse
    C45 -- "No" --> GetResponse
    GetResponse --> C46{C46}
    C46 -- "Si" --> GetResponse
    C46 -- "No" --> GetResponse
    GetResponse --> C47{C47}
    C47 -- "Si" --> GetResponse
    C47 -- "No" --> GetResponse
    GetResponse --> C48{C48}
    C48 -- "Si" --> GetResponse
    C48 -- "No" --> GetResponse
    GetResponse --> C49{C49}
    C49 -- "Si" --> GetResponse
    C49 -- "No" --> GetResponse
    GetResponse --> C50{C50}
    C50 -- "Si" --> GetResponse
    C50 -- "No" --> GetResponse
    GetResponse --> C51{C51}
    C51 -- "Si" --> GetResponse
    C51 -- "No" --> GetResponse
    GetResponse --> C52{C52}
    C52 -- "Si" --> GetResponse
    C52 -- "No" --> GetResponse
    GetResponse --> C53{C53}
    C53 -- "Si" --> GetResponse
    C53 -- "No" --> GetResponse
    GetResponse --> C54{C54}
    C54 -- "Si" --> GetResponse
    C54 -- "No" --> GetResponse
    GetResponse --> C55{C55}
    C55 -- "Si" --> GetResponse
    C55 -- "No" --> GetResponse
    GetResponse --> C56{C56}
    C56 -- "Si" --> GetResponse
    C56 -- "No" --> GetResponse
    GetResponse --> C57{C57}
    C57 -- "Si" --> GetResponse
    C57 -- "No" --> GetResponse
    GetResponse --> C58{C58}
    C58 -- "Si" --> GetResponse
    C58 -- "No" --> GetResponse
    GetResponse --> C59{C59}
    C59 -- "Si" --> GetResponse
    C59 -- "No" --> GetResponse
    GetResponse --> C60{C60}
    C60 -- "Si" --> GetResponse
    C60 -- "No" --> GetResponse
    GetResponse --> C61{C61}
    C61 -- "Si" --> GetResponse
    C61 -- "No" --> GetResponse
    GetResponse --> C62{C62}
    C62 -- "Si" --> GetResponse
    C62 -- "No" --> GetResponse
    GetResponse --> C63{C63}
    C63 -- "Si" --> GetResponse
    C63 -- "No" --> GetResponse
    GetResponse --> C64{C64}
    C64 -- "Si" --> GetResponse
    C64 -- "No" --> GetResponse
    GetResponse --> C65{C65}
    C65 -- "Si" --> GetResponse
    C65 -- "No" --> GetResponse
    GetResponse --> C66{C66}
    C66 -- "Si" --> GetResponse
    C66 -- "No" --> GetResponse
    GetResponse --> C67{C67}
    C67 -- "Si" --> GetResponse
    C67 -- "No" --> GetResponse
    GetResponse --> C68{C68}
    C68 -- "Si" --> GetResponse
    C68 -- "No" --> GetResponse
    GetResponse --> C69{C69}
    C69 -- "Si" --> GetResponse
    C69 -- "No" --> GetResponse
    GetResponse --> C70{C70}
    C70 -- "Si" --> GetResponse
    C70 -- "No" --> GetResponse
    GetResponse --> C71{C71}
    C71 -- "Si" --> GetResponse
    C71 -- "No" --> GetResponse
    GetResponse --> C72{C72}
    C72 -- "Si" --> GetResponse
    C72 -- "No" --> GetResponse
    GetResponse --> C73{C73}
    C73 -- "Si" --> GetResponse
    C73 -- "No" --> GetResponse
    GetResponse --> C74{C74}
    C74 -- "Si" --> GetResponse
    C74 -- "No" --> GetResponse
    GetResponse --> C75{C75}
    C75 -- "Si" --> GetResponse
    C75 -- "No" --> GetResponse
    GetResponse --> C76{C76}
    C76 -- "Si" --> GetResponse
    C76 -- "No" --> GetResponse
    GetResponse --> C77{C77}
    C77 -- "Si" --> GetResponse
    C77 -- "No" --> GetResponse
    GetResponse --> C78{C78}
    C78 -- "Si" --> GetResponse
    C78 -- "No" --> GetResponse
    GetResponse --> C79{C79}
    C79 -- "Si" --> GetResponse
    C79 -- "No" --> GetResponse
    GetResponse --> C80{C80}
    C80 -- "Si" --> GetResponse
    C80 -- "No" --> GetResponse
    GetResponse --> C81{C81}
    C81 -- "Si" --> GetResponse
    C81 -- "No" --> GetResponse
    GetResponse --> C82{C82}
    C82 -- "Si" --> GetResponse
    C8
```

Historial de Versiones

Indicador #1 — Porcentaje de remisiones finalizadas

Total de versiones guardadas: 2

Orden: más reciente a más antigua

Rev #2

UPDATE

Hector 2026-08-23 16:55:19

Tipo de fórmula: Flexible v2

Meta: $\geq 95\%$

Expresión: $(\text{estado_remision_num} / \text{estado_remision_den}) * 1000$

Efectividad • Indicadores RyC

Restaurar esta versión

Rev #1

UPDATE

Hector 2026-08-23 16:53:52

Tipo de fórmula: Flexible v2

Meta: $\geq 95\%$

Expresión: $(\text{estado_remision_num} / \text{estado_remision_den}) * 100$

Efectividad • Indicadores RyC

Restaurar esta versión

A screenshot of a web application interface. At the top, there is a header bar with a logo on the left and the text '¿Restaurar versión anterior?' in the center. Below the header, there is a red-bordered box containing the text: 'Estás a punto de restaurar la configuración del indicador #1 a la Revisión #1 guardada el 2026-08-23 16:53:52 por Hector.' Below this box, there is a light gray rectangular area containing the text: 'Tranquilidad: la versión actual no se perderá; se guardará automáticamente una nueva revisión antes de aplicar la restauración.' At the bottom of the dialog, there are two buttons: a white button with the text 'Cancelar' and an orange button with the text 'Sí, Restaurar Versión'.

tos y duplicados, la normalización de fechas y la consistencia de los resultados generados.

Como evidencia representativa del proceso de validación se documentó el indicador *Tiempo promedio de aceptación de remisión*, definido como $\Delta t = t_{\text{aceptación}} - t_{\text{notificación}}$, promediado en horas sobre las remisiones con estado distinto

la validación de estructura, la detección de registros incomple-

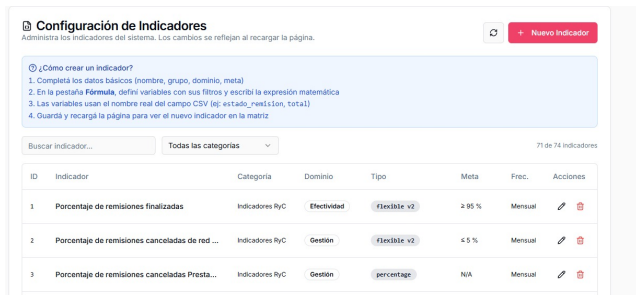


Fig. 14. Panel de configuración de indicadores: 71 de 74 KPI activos, con metadatos de categoría, dominio, tipo, meta y frecuencia.

Nuevo Indicador
Completá los datos del indicador. Usá las pestañas para alternar entre datos básicos y fórmula.

Nombre del indicador *
Ej: Porcentaje de remisiones finalizadas

Categoría (RIAS)
Seleccionar categoría...

Dominio
Seleccionar...

Meta sugerida
Seleccionar meta...

Frecuencia
Mensual

[Cancelar] [Crear Indicador]

Fig. 15. Creación de indicador – pestaña de datos básicos.

Nuevo Indicador
Completá los datos del indicador. Usá las pestañas para alternar entre datos básicos y fórmula.

Variables Capa 1
Define las variables que usarás en la expresión matemática.
total1 COUNT [Agregar variable]

Fórmula Capa 2
Escribe la expresión matemática usando los nombres de las variables.
(total1/5)*100

Formato de salida
Decimales: 0 Unidad: %

Vista previa: 44300

[Cancelar] [Crear Indicador]

Fig. 16. Creación de indicador – definición de variables (Capa 1) y fórmula (Capa 2), con vista previa de resultado.

de *Cancelado* e igual a *Finalizado* (Fig. 24). Sobre una muestra de 6.748 registros con fechas completas (suma total de 57.089,1 horas), el cálculo manual realizado en Microsoft Excel por la auditora del Área de Calidad arrojó un promedio de 8,4684 horas (Fig. 25), mientras que la plataforma generó automáticamente un resultado de 8,5 horas (Fig. 26), con una diferencia de apenas 0,03 horas atribuible al redondeo aplicado en la capa de presentación (Tabla VII). La correspondencia fue confirmada por el Área de Calidad de la institución sobre múltiples indicadores adicionales, siguiendo exactamente la

Nuevo Indicador
Completá los datos del indicador. Usá las pestañas para alternar entre datos básicos y fórmula.

Básicos **Fórmula** **Descripción**

Grupo / Categoría
Ej: Referencia, Contrarreferencia

Objetivo
Describí el objetivo del indicador...

Descripción del numerador
Descripción del denominador

[Cancelar] [Crear Indicador]

Fig. 17. Creación de indicador – pestaña de descripción (objetivo, numerador y denominador).

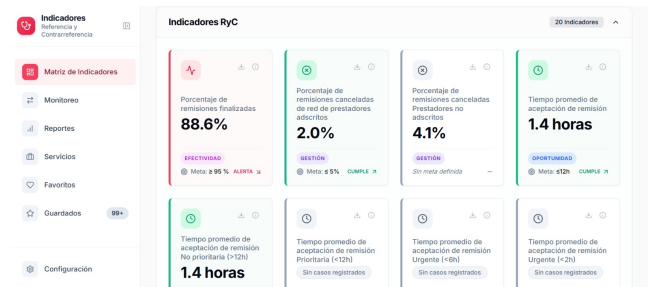


Fig. 18. Tarjetas de indicadores del dominio RyC con meta, estado de cumplimiento y clasificación por dimensión.



Fig. 19. Panel de monitoreo operativo: tiempos agregados de notificación, aceptación e ingreso sobre 8.167 remisiones filtradas.



Fig. 20. Tendencia mensual del volumen total de remisiones.

lógica de agregación descrita en el algoritmo de la Fig. 11.

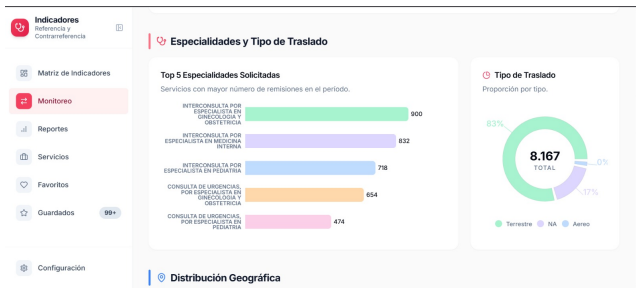


Fig. 21. Top 5 especialidades solicitadas y proporción por tipo de traslado (terrestre, aéreo, no aplica).



Fig. 22. Top 10 municipios por origen del afiliado en el período filtrado.

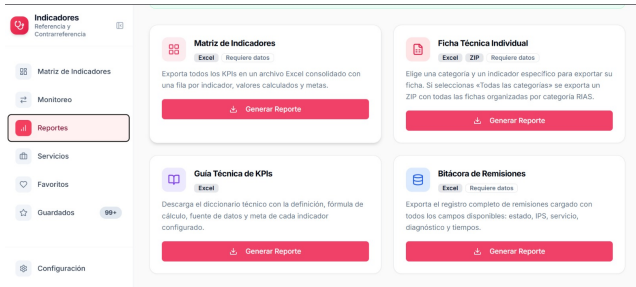


Fig. 23. Módulo de generación de reportes: matriz de indicadores, ficha técnica individual, guía técnica de KPI y bitácora de remisiones (todos exportables a Excel).

TABLE VI
RESULTADOS DEL ANÁLISIS DE CALIDAD DE LOS DATOS

Criterio evaluado	Resultado
Validación de estructura del archivo CSV/XLSX	Correcta
Detección de registros incompletos	Correcta
Identificación de registros duplicados	Correcta
Normalización de fechas y campos requeridos	Correcta
Procesamiento de registros válidos	Correcto
Generación automática de indicadores	Correcta
Consistencia de los resultados obtenidos	Correcta

TABLE VII
COMPARACIÓN ENTRE CÁLCULO MANUAL Y CÁLCULO DEL SISTEMA

Indicador	Manual (Excel)	Sistema	Dif.
Tiempo promedio de aceptación de remisión	8,4684 h	8,5 h	0,03 h



Fig. 24. Definición formal del indicador: fórmula de cálculo, filtro aplicado y trazabilidad (suma total y número de registros).

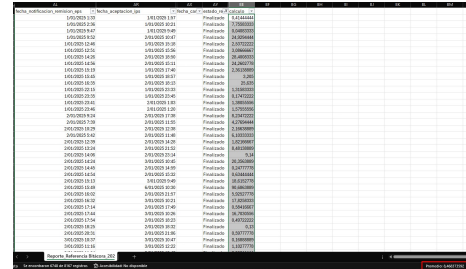


Fig. 25. Cálculo manual de referencia en Microsoft Excel sobre 6.748 de 8.167 registros (promedio: 8,468373592 h).



Fig. 26. Resultado calculado automáticamente por el sistema (8,5 horas), con cumplimiento de la meta institucional (≤ 12 h).

D. Reducción de la tasa de error

Con el fin de cuantificar objetivamente la reducción de errores derivada de la automatización, se realizó una comparación entre el proceso manual (Excel) y la aplicación web sobre un conjunto de 200 registros, contabilizando los registros que presentaron errores en cada proceso (Tabla VIII). El proceso manual presentó 48 registros con errores (tasa de error del 24,0%), mientras que la aplicación web presentó un único registro con error sobre el mismo conjunto (tasa de error del 0,5%). La reducción relativa de la tasa de error se calculó mediante la expresión $\text{Reducción} = [(24,0\% - 0,5\%) / 24,0\%] \times 100 = 97,91\%$, lo que evidencia una disminución significativa y cuantificable en la tasa de error del proceso frente al procedimiento manual utilizado anteriormente.

E. Pruebas de integración con inteligencia artificial

Las pruebas confirmaron la coherencia de los textos generados, tiempos de respuesta adecuados de la API y la correcta

TABLE VIII
COMPARACIÓN DE LA TASA DE ERROR (200 REGISTROS)

Métrica	Manual (Excel)	Aplicación web
Total de registros procesados	200	200
Registros con errores	48	1
Registros sin errores	152	199
Tasa de error	24,0%	0,5%
Tasa de registros correctos	76,0%	99,5%

TABLE IX
PRUEBA DE CONTROL SOBRE LA MODIFICACIÓN DE UN INDICADOR
(SÍNTESIS)

Elemento evaluado	Resultado
Autenticación y verificación del rol del usuario	Cumple
Registro de usuario, fecha y hora de la modificación	Cumple
Registro en <i>log</i> de auditoría	Cumple
Conservación de la fórmula anterior y de la nueva	Cumple
Recuperación (restauración) de una fórmula anterior	Cumple
Confirmación previa antes de aplicar el cambio	Cumple
Validación posterior del resultado del indicador	Cumple

Nota. Catorce elementos evaluados en total; 14/14 cumplieron el resultado esperado.

adaptación de las respuestas mediante *prompts* personalizados. Se verificó explícitamente el mecanismo de redundancia mediante OpenRouter descrito en la Fig. 11, demostrando continuidad operativa ante caídas simuladas del proveedor principal: en todos los escenarios de prueba en que se forzó artificialmente la indisponibilidad de Gemini, el sistema completó satisfactoriamente el ciclo de generación de análisis mediante el proveedor de respaldo, sin requerir intervención manual ni generar errores visibles para el usuario final.

F. Prueba de control de modificaciones de fórmulas de indicadores

Con el fin de verificar el control aplicado sobre las modificaciones realizadas a las fórmulas utilizadas para el cálculo de los indicadores, se realizó una prueba específica sobre el módulo de administración de indicadores. La evaluación contempló los mecanismos de autenticación, control de acceso, registro de auditoría, conservación de versiones y recuperación de configuraciones anteriores; los catorce elementos evaluados—entre ellos la verificación del rol del usuario, el registro del autor y de la fecha/hora de cada modificación, la conservación de la fórmula anterior y de la nueva, la disponibilidad de una opción de restauración y la existencia de un modal de confirmación previo a aplicar el cambio— cumplieron satisfactoriamente el resultado esperado (Tabla IX).

Como evidencia adicional del mecanismo de control implementado, la Fig. 12 presenta el historial de versiones asociado a un indicador, donde se registran las modificaciones realizadas, el usuario responsable y la fecha y hora de cada cambio. Asimismo, la Fig. 13 muestra el mecanismo de confirmación utilizado antes de restaurar una versión anterior

TABLE X
EVALUACIÓN DE LAS INTERPRETACIONES GENERADAS POR IA (10 CASOS)

Caso	Indicador evaluado	Puntaje
1	% de remisiones finalizadas	5/5
2	% de remisiones canceladas – red de prestadores adscritos	5/5
3	% de remisiones canceladas – prestadores no adscritos	4/5
4	Tiempo promedio de aceptación de remisión	5/5
5	Tiempo prom. de aceptación – no prioritaria (>12h)	5/5
6	Tiempo prom. de aceptación – prioritaria (<12h)	5/5
7	Tiempo prom. de aceptación – urgente (<6h)	5/5
8	Tiempo prom. de aceptación – urgente (<2h)	5/5
9	Tiempo promedio de ubicación de remisión	5/5
10	Tiempo prom. de ubicación – no prioritaria (>12h)	5/5
Cumplimiento global (49/50 criterios)		98,0%

del indicador, el cual solicita la confirmación del usuario antes de aplicar la restauración y permite mantener la trazabilidad de la operación mediante el registro de una nueva revisión. En conjunto, los resultados evidencian que el módulo de administración de indicadores cuenta con mecanismos de autenticación y autorización para controlar quién puede modificar las fórmulas, registros de auditoría para identificar las operaciones realizadas, e historial de versiones para conservar y restaurar configuraciones anteriores ante una modificación incorrecta.

G. Validación de la calidad de las interpretaciones generadas por IA

Además de las pruebas funcionales de integración descritas previamente, se realizó una evaluación orientada a determinar objetivamente la calidad de las interpretaciones generadas por el componente de inteligencia artificial. Para ello, se seleccionaron diez casos de análisis asociados a diferentes indicadores institucionales y se evaluó cada interpretación mediante una rúbrica compuesta por cinco criterios: correspondencia con los datos suministrados, identificación correcta de la tendencia, coherencia de la conclusión, ausencia de información no presente en los datos y adecuación al contexto del indicador. Cada criterio se valoró de manera binaria (1 si se cumplía, 0 en caso contrario), de modo que cada interpretación podía obtener una puntuación máxima de 5 puntos (Tabla X).

De los diez casos evaluados, nueve obtuvieron la puntuación máxima de 5/5, mientras que el Caso 3 obtuvo 4/5 al incorporar información adicional no presente explícitamente en los datos suministrados, incumpliendo el criterio de ausencia de información no presente en los datos. El porcentaje global de cumplimiento se calculó mediante la expresión $\text{Cumplimiento} = (49/50) \times 100 = 98,0\%$. Estos resultados evidencian un alto nivel de correspondencia entre las interpretaciones generadas y los datos y el contexto de los indicadores evaluados; no obstante, el caso con observación resalta la importancia de mantener una revisión humana de las interpretaciones generadas antes de utilizarlas como único soporte para la toma de decisiones administrativas.



Fig. 27. Resultados de la encuesta de satisfacción aplicada a la usuaria principal del sistema (n=1, usuaria clave del proceso).

H. Usabilidad, satisfacción y aceptación tecnológica

Se aplicó una encuesta de satisfacción a la Coordinadora de Autorizaciones —usuaria final del sistema— sobre cinco dimensiones (facilidad de uso, claridad de la información, velocidad del sistema, facilidad para exportar y utilidad general), calificadas en una escala de 1 a 5. La totalidad de las dimensiones evaluadas obtuvo la calificación máxima (5/5, 100% de las respuestas), como se ilustra en la Fig. 27. Complementariamente, una escala de Likert de cinco niveles orientada a la aceptación tecnológica evidenció una percepción favorable respecto a la reducción del tiempo de análisis, la facilidad de consulta mediante filtros dinámicos, la confiabilidad de los resultados y la utilidad de la generación automática de análisis mediante IA.

I. Calidad interna del software

El aseguramiento de la calidad del código se desarrolló mediante análisis estático iterativo con SonarQube bajo el enfoque *Clean as You Code*, complementado con pruebas unitarias, pruebas automatizadas con Playwright y asistencia de modelos de IA (Claude Opus, Kimi K3 y Claude Fable) para la identificación de oportunidades de refactorización, siguiendo los lineamientos de ISO/IEC 25002:2024 [25] e ISO/IEC/IEEE 29119-1:2022 [26]. Todas las recomendaciones generadas por los asistentes de IA fueron revisadas y validadas manualmente antes de su incorporación.

La Tabla XI resume la evolución de las métricas de calidad a través de tres iteraciones. En la primera versión, ambos

componentes superaban el *Quality Gate*, pero con oportunidades relevantes de mejora: 63,7% de cobertura y 5,1% de duplicación en backend, y 51,1% de cobertura y 4,7% de duplicación en frontend, con calificación de confiabilidad C en el frontend. Tras dos ciclos de refactorización asistida, la versión final alcanzó calificaciones A en seguridad, confiabilidad y mantenibilidad en ambos componentes, cero vulnerabilidades y *security hotspots*, cobertura de 84,8% (backend) y 81,4% (frontend) —ambas por encima del 75% mínimo institucional—, y duplicación de 1,9% y 2,1% respectivamente.

J. Comparación entre proceso manual y sistema web

La Tabla XII resume el efecto agregado de la solución sobre las principales dimensiones operativas evaluadas. El hallazgo más relevante consiste en la reducción del tiempo de procesamiento: mientras el proceso manual tradicional requería entre una y dos semanas para consolidar, validar y analizar la información institucional, la aplicación desarrollada realizó el mismo proceso en aproximadamente cuatro minutos.

VII. ANÁLISIS DE RESULTADOS Y DISCUSIÓN

El análisis de resultados permitió evaluar el impacto de la aplicación desarrollada frente a los objetivos específicos, los requerimientos definidos y la metodología aplicada. Durante la validación se evidenció que la plataforma automatizó tareas antes realizadas mediante hojas de cálculo y archivos dispersos, reduciendo inconsistencias y facilitando la organización de los registros institucionales, lo que sustenta la hipótesis de trabajo planteada y resulta consistente con lo reportado por Barrios y Méndez [2] sobre el impacto de la digitalización en procesos de RyC en Latinoamérica, y con la relación documentada por Saleyvis et al. [4] entre calidad de indicadores y eficiencia organizacional hospitalaria.

El hallazgo más relevante fue la reducción del tiempo de generación de indicadores y reportes, de un rango de una a dos semanas (proceso manual) a aproximadamente cuatro minutos, sin comprometer la exactitud de los resultados. De forma complementaria, la comparación sobre 200 registros evidenció una reducción relativa del 97,9% en la tasa de error (24,0% manual frente a 0,5% en la aplicación web; Sección VI-D), respaldando cuantitativamente el aporte de los mecanismos automatizados de validación, limpieza y transformación de datos.

La integración de IA mediante Google AI Studio y los modelos Gemini permitió complementar el análisis administrativo con reportes en lenguaje natural. Más allá de la verificación funcional, la evaluación objetiva de la calidad de las interpretaciones (rúbrica de cinco criterios sobre diez casos, Sección VI-G) arrojó un cumplimiento global del 98,0% (49/50), evidencia cuantitativa adicional sobre la fiabilidad de los análisis generados. El caso con observación identificado —relacionado con información no presente explícitamente en los datos— confirma la conveniencia, señalada también por Liu et al. [8] en sistemas de soporte a la decisión clínica, de mantener revisión humana de las interpretaciones antes de usarlas como soporte de decisiones. La incorporación de

TABLE XI
EVOLUCIÓN DE LA CALIDAD DEL CÓDIGO (SONARQUBE)

Métrica	Backend			Frontend		
	V1	V2	V3	V1	V2	V3
Coverage	63,7%	77,8%	84,8%	51,1%	76,4%	81,4%
Duplicación	5,1%	3,6%	1,9%	4,7%	3,4%	2,1%
Maint. Issues	123	64	61	87	87	57
Security	A	A	A	A	A	A
Reliability	A	A	A	C	A	A
Quality Gate	Passed	Passed	Passed	Passed	Passed	Passed

TABLE XII
COMPARACIÓN ENTRE PROCESO MANUAL Y SISTEMA WEB

Proceso	Método manual	Sistema web
Consolidación	Hojas de cálculo dispersas	Carga y procesamiento centralizado
Indicadores	Cálculo manual	Cálculo automático (KPI)
Reportes	Elaboración manual	Generación automatizada con IA
Datos	Distribuidos	Centralizados y estructurados
Errores	Validación manual	Validación y limpieza automática
Tiempo	1–2 semanas	~4 minutos
Actualización	Consolidación periódica	Automática tras cada carga
Visualización	Tablas y documentos	Dashboards interactivos
Exportación	Manual	Automatizada (Excel)
Detección de errores	Validación manual (tasa de error 24,0%)	Validación y limpieza automática (tasa de error 0,5%)
Calidad de interpretaciones	No dispone de generación automatizada	Generación mediante IA con 98,0% de cumplimiento

OpenRouter como proveedor alternativo mejoró además la disponibilidad de la plataforma, mientras que las técnicas de *prompt engineering* facilitaron la interpretación de métricas institucionales y agilizaron la elaboración de reportes.

La estrategia de minimización de datos —procesar la información identificable solo en la capa local y enviar al modelo únicamente métricas agregadas (Fig. 11)— responde de forma concreta a las preocupaciones de privacidad señaladas por Esmailzadeh [14] como barrera para la adopción de IA en salud. El mecanismo de redundancia entre Gemini y OpenRouter (Figs. 1, 10 y 11) aborda, asimismo, la dependencia de un único proveedor de IA como punto único de falla, riesgo habitualmente subestimado en la literatura.

Desde el punto de vista tecnológico, la arquitectura basada en Next.js, Node.js/Express, PostgreSQL y Oracle Cloud Infrastructure permitió una solución modular, escalable y estable, reforzada por autenticación JWT en el backend. La mejora sostenida de las métricas de SonarQube a través de las tres iteraciones —cobertura de 51,1%–63,7% a 81,4%–84,8%— evidencia el valor del análisis estático continuo, en línea con el modelo SQuaRE (ISO/IEC 25002:2024) [25], mientras que

los registros de auditoría en JSON y `.log` fortalecieron la trazabilidad de las operaciones. La incorporación de asistentes de IA (Claude Opus, Kimi K3, Claude Fable) como apoyo al proceso de refactorización sugiere un patrón de trabajo híbrido humano-IA replicable en equipos reducidos, habituales en instituciones de salud de mediana escala.

La documentación completa de las vistas estática, estructural y dinámica (Secciones IV y V) mantuvo una correspondencia directa entre los requerimientos levantados en campo (Tabla II) y los módulos implementados (Tabla V), reduciendo el riesgo de desviación diseño-construcción propio del prototipado evolutivo sin documentación explícita. El mecanismo de control de versiones sobre las fórmulas de los indicadores (Sección V-E), con cumplimiento total en su prueba de catorce elementos, aporta una capa de gobernanza de datos poco documentada en los antecedentes: conservar el historial y permitir restauraciones reduce el riesgo de que un cambio erróneo se propague silenciosamente a los reportes. Los *dashboards* interactivos y visualizadores KPI, por su parte, facilitaron la comprensión de las métricas mediante gráficos, tablas dinámicas y filtros personalizados.

En conjunto, los resultados evidencian que las herramientas web asistidas con IA pueden optimizar procesos administrativos de RyC en entidades de salud. No obstante, deben señalarse limitaciones: la validación de usabilidad se realizó con una única usuaria clave ($n=1$), lo que limita la generalización estadística; la comparación de precisión se documentó explícitamente para un indicador representativo, aunque confirmada cualitativamente por el Área de Calidad para otros indicadores; y, al ser un estudio de caso único, los hallazgos no son directamente generalizables a otras EPS indígenas sin validación adicional que considere diferencias de infraestructura, volumen de datos y madurez organizacional.

VIII. CONCLUSIONES

El desarrollo de la aplicación web asistida con IA permitió optimizar la gestión de indicadores y la generación de reportes en DUSAKAWI EPSI. La caracterización inicial identificó duplicidad de registros, errores de digitación, retrasos en la consolidación y falta de centralización de la información, diagnóstico que orientó los requerimientos funcionales y tecnológicos de la solución.

La arquitectura modular (Next.js, Express/TypeScript, PostgreSQL, autenticación JWT) facilitó la escalabilidad, organización y mantenimiento del sistema; la documentación

explícita de las vistas de arquitectura global, componentes, despliegue, casos de uso, modelo de dominio, actividades, secuencia y especificación algorítmica constituye un aporte metodológico replicable para proyectos similares con equipos técnicos reducidos. Los registros de auditoría en JSON y .log, junto con los módulos de carga, validación y limpieza de datos, fortalecieron la trazabilidad y la calidad de la información utilizada para el cálculo de indicadores.

El módulo de administración de indicadores —con creación, edición, configuración de fórmulas y mecanismos de control de versiones y auditoría (Sección V-E)— incrementó la flexibilidad y mantenibilidad de la solución, permitiendo adaptarla a nuevos requerimientos de forma segura y sin modificar el código fuente. La integración de Gemini y OpenRouter como respaldo, junto con técnicas de PLN, contribuyó a la generación automatizada de reportes: la evaluación mediante rúbrica (diez casos, cinco criterios) evidenció un cumplimiento global del 98,0% (49/50), respaldando la utilidad de la IA como apoyo al análisis administrativo, sin sustituir la revisión humana.

La validación confirmó una reducción del tiempo de generación de indicadores de una a dos semanas a aproximadamente cuatro minutos, y una reducción del 97,9% en la tasa de error (24,0% a 0,5% sobre 200 registros), evidenciando el aporte de los mecanismos automatizados de validación y transformación de datos. El Área de Calidad de la institución confirmó, además, la precisión de los indicadores frente a los cálculos manuales de referencia, y los *dashboards* interactivos y filtros personalizados mejoraron la presentación y consulta de la información. La evaluación mediante SonarQube evidenció que frontend y backend superaron el *Quality Gate*, con calificaciones A en seguridad, confiabilidad y mantenibilidad y coberturas superiores al mínimo institucional. En conjunto, los objetivos planteados fueron alcanzados mediante la caracterización del proceso, el diseño de la arquitectura, el desarrollo de la plataforma, la automatización del procesamiento de información, la administración configurable de indicadores y la integración de IA para la generación de reportes.

IX. RECOMENDACIONES

Se recomienda continuar fortaleciendo la digitalización de los procesos de RyC, reduciendo la dependencia de procesos manuales y ampliando progresivamente la integración de nuevas fuentes de información institucional. Se sugiere implementar procesos periódicos de validación y actualización de los datos, dado que la automatización reduce significativamente los errores pero no elimina inconsistencias en los datos de origen, así como capacitaciones continuas al personal encargado de la plataforma para fortalecer su apropiación tecnológica.

Se recomienda mantener mecanismos de revisión humana sobre los reportes e interpretaciones generados mediante IA: pese al cumplimiento del 98,0% obtenido, el caso con observación identificado confirma que estos resultados deben considerarse un apoyo al análisis administrativo y no un sustituto de la validación del personal responsable. Asimismo,

se recomienda mantener controles de autorización sobre las configuraciones del módulo de indicadores almacenadas en la base de datos institucional, verificando que toda modificación, incorporación o eliminación de indicadores, fórmulas o metas sea previamente aprobada por los responsables del proceso.

Se recomienda conservar la estrategia de aseguramiento de calidad mediante pruebas unitarias, pruebas automatizadas y análisis estático con SonarQube, dado que el proyecto alcanzó coberturas superiores al mínimo institucional y superó el *Quality Gate*; mantener estas prácticas evita la degradación de la seguridad, confiabilidad y mantenibilidad del software. Se aconseja, además, un monitoreo continuo de los registros de auditoría (JSON y .log) para facilitar la identificación temprana de incidentes o comportamientos anómalos.

Finalmente, se recomienda explorar futuras integraciones con otros sistemas institucionales de salud para mejorar la interoperabilidad, considerando siempre los controles de seguridad, autorización y protección de datos de la institución. Como líneas de trabajo futuro complementarias se identifican: (i) ampliar la validación cuantitativa de precisión a la totalidad de los 71 indicadores; (ii) incorporar versionamiento e historial de aprobación adicionales en el módulo de configuración de indicadores; (iii) desarrollar módulos predictivos orientados a la identificación temprana de riesgos operativos; (iv) ampliar los estudios de usabilidad a otros roles institucionales; y (v) explorar la interoperabilidad con sistemas de información en salud a nivel departamental y nacional, en línea con la Ley 1751 de 2015 [23].

REFERENCES

- [1] L. Huauya-Cancharí y C. G. Medina-Sotelo, "Políticas de referencia y contrarreferencia en salud pública: comparación entre niveles de atención en América Latina," *Prohominum. Revista de Ciencias Sociales y Humanas*, vol. 7, pp. 307–319, 2025.
- [2] R. L. Barrios y D. M. Méndez, "Impacto de la digitalización en la continuidad del cuidado: el caso del proceso de referencia y contrarreferencia de la red pública de salud chilena," *Revista Médica de Chile*, vol. 153, pp. 857–870, 2025.
- [3] R. M. B. Magallón y E. A. M. González, "Sistemas de información estadísticos e indicadores de salud," *Revista Varianza*, pp. 1–15, 2022.
- [4] A. M. Saleyvis, M. Valdés, J. F. Pérez, E. V. Contreras, P. Sebastián, y H. Ritchie, "Contribución de los indicadores de gestión en la eficiencia organizacional y la administración hospitalaria en instituciones de salud," *UNESUM – Ciencias. Revista Científica Multidisciplinaria*, vol. 5, pp. 109–122, 2021.
- [5] P. Quezada, "El rol de la tecnología en la gestión de la salud en hospitales públicos," *SAGAS. Revista Científica Multidisciplinaria*, vol. 2, pp. 1–21, 2025.
- [6] M. Kazi, S. Uddin, K. Md, y R. Hossan, "A review of implementing AI-powered data warehouse solutions to optimize big data management and utilization," *Academia Journal on Business Administration, Innovation & Sustainability*, vol. 4, 2024.
- [7] B. Zhang y M. N. K. Boulous, "Generative AI in medicine and healthcare: Promises, opportunities and challenges," *Future Internet*, vol. 15, no. 9, p. 286, 2023.
- [8] S. Liu *et al.*, "Using AI-generated suggestions from ChatGPT to optimize clinical decision support," *J. Am. Med. Inform. Assoc.*, vol. 30, no. 7, pp. 1237–1245, 2023.
- [9] A. Kuwaiti *et al.*, "A review of the role of artificial intelligence in healthcare," *J. Pers. Med.*, vol. 13, no. 6, p. 951, 2023.
- [10] C. Barbieri, L. Neri, S. Storti, F. Mari, y J. D. Martín-Guerrero, "From electronic health records to clinical management systems: how the digital transformation can support healthcare services," *Clin. Kidney J.*, vol. 16, pp. 1878–1884, 2023.

- [11] D. F. Prabhu, V. Gurupur, A. Stone, y E. Trader, "Integrating artificial intelligence, electronic health records, and wearables for predictive, patient-centered decision support in healthcare," *Healthcare*, vol. 13, no. 21, p. 2753, 2025.
- [12] C. Collins, D. Dennehy, K. Conboy, y P. Mikalef, "Artificial intelligence in information systems research: A systematic literature review and research agenda," *Int. J. Inf. Manage.*, vol. 60, p. 102383, 2021.
- [13] E. J. Z. Zambrano y M. T. F. Valdiviezo, "Herramientas de mejora para los procesos logísticos en el sector hospitalario ecuatoriano," *Uniandes Episteme*, vol. 12, pp. 237–254, 2025.
- [14] P. Esmailzadeh, "Challenges and strategies for wide-scale artificial intelligence (AI) deployment in healthcare practices: A perspective of healthcare organizations," *Artif. Intell. Med.*, vol. 102, p. 101865, 2020.
- [15] J. Wang *et al.*, "Machine learning for healthcare: Methodologies and applications," *Meta-Radiology*, vol. 4, p. 100190, 2026.
- [16] M. Sallam, "ChatGPT utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns," *Healthcare*, vol. 11, no. 3, p. 323, 2023.
- [17] M. Khanna, P. Anand, J. Symons, K. Flott, A. Darzi, y E. Mayer, "Applying natural language processing and machine learning to patient experience feedback: a systematic review," *BMJ Health Care Inform.*, vol. 28, no. 1, e100262, 2021.
- [18] S. Secinaro, D. Calandra, A. Secinaro, V. Muthurangu, y P. Biancone, "The role of artificial intelligence in healthcare: a structured literature review," *BMC Med. Inform. Decis. Mak.*, vol. 21, p. 125, 2021.
- [19] J. Christie *et al.*, "Use of Epic electronic health record system for health care research: Scoping review," *J. Med. Internet Res.*, vol. 25, e51003, 2023.
- [20] C. Rojas-Villa, P. Ormeño-Vásquez, y N. Rojas-Villa, "Percepción del usuario externo sobre la atención sanitaria en el servicio de referencias y contrarreferencias," *Revista INFODIR*, vol. 44, 2025.
- [21] Corte Constitucional de Colombia, "Ley 1581 de 2012," Gestor Normativo, Función Pública, 2012. [En línea]. Disponible: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>
- [22] Congreso de Colombia, "Ley 1438 de 2011: Por medio de la cual se reforma el Sistema General de Seguridad Social en Salud y se dictan otras disposiciones," Ministerio de Salud y Protección Social, 2011.
- [23] Gobernación del Huila, "Ley Estatutaria 1751 de 2015," Sede Electrónica del Departamento del Huila, 2015.
- [24] Ministerio de Salud y Protección Social, "Resolución 4331 de 2012," Minsalud, 2012.
- [25] ISO/IEC 25002:2024, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality model overview and usage*, International Organization for Standardization, 2024.
- [26] ISO/IEC/IEEE 29119-1:2022, *Software and systems engineering – Software testing – Part 1: General concepts*, International Organization for Standardization, 2022.