



Arquitectura Hexagonal y Microservicios para la Digitalización de Procesos Administrativos en una Institución Pública: Caso de Gestión de Vacaciones y Permisos

Hexagonal Architecture and Microservice for the Digitalization of Administrative Processes in a Public Institution: Case of Vacation and Leave Management

Autores

*Britney Paulette Tenecela Tocto¹

✉ btenece1@utmachala.edu.ec



Kevin Joel Zuñiga Iñiguez¹

✉ kzuniga2@utmachala.edu.ec



Bertha Mazon-Olivo¹

✉ bmazon@utmachala.edu.ec



Damar Alejandra Taylor Vivanco²

✉ dtaylor@bmv.edu.ec



¹Universidad Técnica de Machala, Facultad de Ingeniería Civil, Carrera de Tecnologías de la Información, Machala, El Oro, Ecuador.

²Instituto Superior Tecnológico Bolívar Madero Vargas, Carrera de Desarrollo de Software, Machala, El Oro, Ecuador.

*Autor para correspondencia

Comó citar el artículo:

Tenecela Tocto, B.P., Zuñiga Iñiguez, K.J., Mazon-Olivo, B. & Taylor Vivanco, D.A. (2026). Arquitectura Hexagonal y Microservicios para la Digitalización de Procesos Administrativos en una Institución Pública: Caso de Gestión de Vacaciones y Permisos. *Informática y Sistemas*, 10(1), pp. 79-97. <https://doi.org/10.33936/isrtic.v10i1.8441>

Enviado: 18/05/2026

Aceptado: 11/06/2026

Publicado: 11/06/2026

Resumen

Gestionar vacaciones, licencias y permisos con hojas de cálculo genera errores, duplicados y baja trazabilidad. Para atender esta problemática, este trabajo documenta el diseño, desarrollo y evaluación del sistema GIATH, orientado a centralizar y automatizar solicitudes en Talento Humano de una entidad pública ecuatoriana denominada ISE por confidencialidad, cuyo volumen histórico alcanzó 1.997 trámites de 626 funcionarios entre enero de 2024 y septiembre de 2025. La solución integra metodológicamente el modelo As-Is/To-Be, que rediseñó los flujos administrativos, con la metodología Extreme Programming, que orientó el desarrollo iterativo; ambos enfoques se materializaron en una arquitectura de microservicios con patrón hexagonal implementada en Java 21, Spring Boot, Angular, TypeScript, SQL Server 2022 y MongoDB. Tres estrategias de validación respaldan los resultados obtenidos. Las pruebas funcionales confirmaron integridad y consistencia de datos en todos los escenarios. Las pruebas de rendimiento ejecutadas con Apache JMeter en cinco niveles de carga evidenciaron tiempos promedio de 110 ms con 20 usuarios y 472 ms con 70 usuarios concurrentes, sin ningún error (tasa 0,0 %) y con un throughput máximo de 95,3 req/s. Frente al proceso manual, que demandaba cerca de diez minutos por solicitud, el sistema lo reduce a menos de 1,5 segundos: una mejora superior al 99,7 %. La encuesta a 22 expertos en TI, basada en ISO/IEC 25010 con un Alfa de Cronbach de 0,87, arrojó promedios superiores a 4,4 sobre 5 en todas las dimensiones. GIATH constituye un modelo técnico-metodológico transferible para digitalizar y validar trámites administrativos análogos en el sector público.

Palabras claves: Arquitectura hexagonal; Microservicios; As-Is/To-Be; Extreme Programming; Talento Humano.

Abstract

Gestionar vacaciones, licencias y permisos con hojas de cálculo genera errores, duplicados y escasa trazabilidad. Managing vacations, licenses, and permits with spreadsheets generates errors, duplicates, and low traceability. To address this issue, this work documents the design, development, and evaluation of the GIATH system, aimed at centralizing and automating requests in Human Talent of a public entity in Ecuador called ISE for confidentiality, whose historical volume reached 1,997 transactions from 626 officials between January 2024 and September 2025. The solution methodologically integrates the As-Is/To-Be model, which redesigned the administrative flows, with the Extreme Programming methodology, which guided the iterative development; both approaches were materialized in a microservices architecture with a hexagonal pattern implemented in Java 21, Spring Boot, Angular, TypeScript, SQL Server 2022, and MongoDB. Three validation strategies support the results obtained. Functional tests confirmed data integrity and consistency in all scenarios. Performance tests conducted with Apache JMeter at five load levels showed average times of 110 ms with 20 users and 472 ms with 70 concurrent users, with no errors (0.0% rate) and a maximum throughput of 95.3 req/s. Compared to the manual process, which took about ten minutes per request, the system reduces it to less than 1.5 seconds: an improvement of over 99.7%. The survey of 22 IT experts, based on ISO/IEC 25010 with a Cronbach's Alpha of 0.87, yielded averages above 4.4 out of 5 in all dimensions. GIATH constitutes a transferable technical-methodological model for digitizing and validating analogous administrative procedures in the public sector.

Keywords: Hexagonal architecture; Microservices; As-Is/To-Be; Extreme Programming; Human resources.



1. Introducción

La administración de recursos humanos en las entidades públicas demanda procesos confiables, auditables y conformes a la normativa vigente. En Ecuador, la Ley Orgánica del Servicio Público LOSEP (2010) rige todo lo relacionado con vacaciones, licencias y permisos de los funcionarios. En ese orden de ideas, la consulta a información consistente y rastreable es necesaria para asegurar la continuidad operativa, la conformidad con los requerimientos normativos y la transparencia administrativa. Análogamente, Espinoza Beltrán y Cachipundo Vásquez (2024) exponen procedimientos de planificación, evaluación, formación y supervisión de licencias para el desarrollo del quehacer institucional. Del mismo modo, Pinargote Párraga y Pico Macías (2023) afirman que la gestión estratégica del personal contribuye al crecimiento del cuerpo organizativo porque vincula las metas institucionales con los resultados. En esa misma línea, Alarcón Quinapanta y Hernández Junco (2024) señalan la retención y el desarrollo del talento como aspectos significativos para la sostenibilidad laboral. También se utilizan herramientas digitales para la renovación de la administración y la facilitación de los procesos internos (Proaño Ponce & Capurro Choez, 2023); y aquí se sitúan aquellas aplicaciones web y bases de datos con las que se gestiona la información y se resguardan procedimientos administrativos (Juma Alba et al., 2024).

En ese sentido, se han desarrollado diferentes soluciones tecnológicas para administrar el talento humano. Wang (2023) desarrolló una plataforma de gestión universitaria basada en computación en la nube, con arquitectura multicapa J2EE, con tecnologías como Hibernate para guardar datos, HDFS y HBase para almacenamiento distribuido. En una línea similar, Juma Alba et al. (2024) implementaron un sistema con PHP y CodeIgniter junto con MySQL para el control de asistencia y evaluación de desempeño laboral. Así mismo, Caciano Arroyo et al. (2024) implementaron un sistema web para la gestión de ventas y del personal de un supermercado, utilizando la metodología ágil Scrum, el framework Laravel y pruebas de rendimiento con JMeter, lo que evidenció una adecuada sinergia entre los equipos de trabajo y los procesos operativos de la organización. Estas propuestas evidencian mejoras en la automatización y el manejo de información, pero se centran en intenciones específicas o prototipos sin explorar arquitecturas orientadas al bajo acoplamiento, escalabilidad, interoperabilidad y trazabilidad completa de operaciones.

El problema científico que los trabajos previos no han resuelto radica en la vulnerabilidad del software ante la evolución normativa y la pérdida de trazabilidad operativa debido al alto acoplamiento arquitectónico. Cuando se producen cambios en

la legislación laboral como la LOSEP, es necesario actualizar las reglas de negocio del sistema. En arquitecturas tradicionales como multicapa o MVC monolítico, las modificaciones generan efectos en cascada que obligan a alterar bases de datos e interfaces de usuario, comprometiendo la estabilidad del sistema y la fidelidad de las auditorías. De acuerdo con Durán Delgado y Rojas Sánchez (2023), la poca integración entre aplicaciones impacta en la calidad de la gestión y la visibilidad de los procesos relacionados con el recurso humano. En esa misma línea, Carvajal-Pérez (2021) afirma que la prevalencia de procesos manuales limita la innovación administrativa y obstruye el avance moderno de la organización.

La revisión comparativa de la Tabla 1 revela tres brechas científico-técnicas no resueltas:

- **Brecha 1:** Desarticulación entre rediseño de procesos y desarrollo de software. Los trabajos que emplean As-Is/To-Be se concentran en el modelado de procesos sin documentar la implementación técnica ni la arquitectura resultante. Ferreira et al. (2026) proponen la mejora del proceso de reclutamiento mediante BPMN e inteligencia artificial, pero no especifican cómo se construyó técnicamente el sistema ni qué arquitectura lo soporta. Esta tendencia hacia la digitalización y la incorporación de inteligencia artificial en los sistemas judiciales de Ecuador y América Latina evidencia un interés por la modernización sectorial, aunque frecuentemente se relega a un segundo plano el sustento arquitectónico y la automatización de los flujos administrativos básicos de sus propios funcionarios (Palate-Ayme et al., 2024; Pérez-Pacheco & Simón Castellano, 2025). En sentido contrario, las propuestas estrictamente tecnológicas parten del desarrollo sin analizar el proceso administrativo subyacente (Wang, 2023; Juma Alba et al., 2024; Caciano Arroyo et al., 2024), lo que limita la alineación entre la solución informática y las necesidades institucionales reales.

- **Brecha 2:** Ausencia de arquitecturas desacopladas orientadas a auditoría y evolución normativa. Las soluciones documentadas emplean arquitecturas multicapa tradicionales (Wang, 2023) o MVC monolítico (Juma Alba et al., 2024; Caciano Arroyo et al., 2024), sin separar la lógica de dominio de la infraestructura de persistencia y presentación, lo que impide modificar las reglas de negocio de forma aislada. Una arquitectura bien estructurada permite que el sistema sea mantenible y evolucionable de forma independiente (Cárdenas-Gutiérrez et al., 2022). La arquitectura hexagonal permite aislar la lógica de negocio de los adaptadores externos como bases de datos, interfaces y servicios, mientras que los microservicios posibilitan el despliegue independiente y la evolución diferenciada de componentes (Celis Crisostomo et al., 2025; Rodríguez & Cedeño, 2023). Esta combinación

arquitectónica resulta pertinente en contextos donde las reglas de negocio cambian frecuentemente por actualizaciones normativas y donde cada operación debe ser auditable.

- **Brecha 3:** Evaluación limitada de criterios de calidad técnica. La mayoría de los trabajos revisados validan sus propuestas mediante pruebas funcionales o de rendimiento específicas como JMeter en Caciano Arroyo et al. (2024), pero no aplican marcos integrales de evaluación de calidad de software. La ausencia de validación bajo estándares como ISO/IEC 25010 limita la evidencia sobre atributos críticos como mantenibilidad, seguridad, eficiencia y usabilidad del sistema desarrollado.

las principales diferencias en términos de acoplamiento, adaptabilidad normativa, auditoría y escalabilidad.

Los resultados de la comparación muestran que la combinación de microservicios y arquitectura hexagonal presenta ventajas en desacoplamiento, adaptabilidad normativa, trazabilidad y escalabilidad respecto a las arquitecturas tradicionales. Estas características son especialmente relevantes en entornos públicos donde los cambios regulatorios afectan continuamente las reglas de negocio y donde la auditoría de operaciones constituye un requisito institucional.

La adopción de microservicios en este entorno institucional

Tabla 1. Comparación de enfoques metodológicos y arquitectónicos en sistemas de gestión administrativa y talento humano.

Fuente: Los autores

Autor(es)	Dominio	Metodología de desarrollo	Arquitectura	Aspectos relevantes
Wang (2023)	Gestión universitaria.	Desarrollo directo	Multicapa J2EE, computación en nube.	Arquitectura monolítica/multicapa con alto acoplamiento sin análisis previo del proceso.
Juma Alba et al. (2024)	Desempeño laboral.	Desarrollo directo	MVC con PHP CodeIgniter	Prototipo funcional sin estrategia de escalabilidad ni auditoría de operaciones.
Caciano Arroyo et al. (2024)	Gestión comercial y RRHH.	Scrum	MVC con Laravel.	Lógica de negocio acoplada a la infraestructura, orientada a un caso específico.
Ferreira et al. (2026)	Reclutamiento empresarial.	As-Is/To-Be con BPMN	Solución impulsada por IA	Orientado a modelado de procesos sin detalles de implementación arquitectónica.
Trabajo propuesto	Gestión de vacaciones, licencias y permisos en el sector público.	As-Is/To-Be + XP	Hexagonal + Microservicios	Integración de As-Is/To-Be, XP, arquitectura hexagonal y microservicios, validación (pruebas funcionales, rendimiento y evaluación de calidad basada en ISO/IEC 25010).

Para fundamentar la elección técnica de este estudio frente a paradigmas arquitectónicos convencionales, se comparan arquitecturas monolíticas, multicapa tradicionales y microservicios sin patrón hexagonal. La Tabla 2 resume

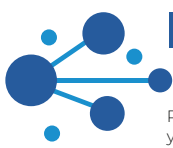
de escala moderada responde a la complejidad normativa del dominio. Un monolito modular podría procesar el caudal de solicitudes documentado con menor overhead operativo; sin embargo, la LOSEP regula mediante reglas diferenciadas cada

Tabla 2. Evidencia comparativa de la propuesta frente a paradigmas arquitectónicos tradicionales.

Fuente: Los autores

Criterio técnico	Arquitectura monolítica	Arquitectura multicapa tradicional	Microservicios sin patrón hexagonal	Propuesta: microservicios + arquitectura hexagonal
Acoplamiento de reglas de negocio	Alto (lógica dispersa en base de datos e interfaz).	Alto (dependencia secuencial estricta entre capas).	Media (aislado por servicio, pero acoplado a la infraestructura del microservicio).	Mínimo (el dominio es agnóstico a la base de datos, frameworks y contratos externos).
Adaptabilidad a cambios normativos	Baja (requiere compilar y desplegar toda la aplicación).	Baja (el cambio impacta hacia arriba y hacia abajo en la pila de capas).	Media (afecta solo a un microservicio, pero altera su persistencia).	Alta (las reglas de negocio pueden modificarse de forma aislada en el núcleo del dominio).
Granularidad de auditoría	Compleja (centralizada en una única base de datos transaccional).	Limitada (dependiente de disparadores o capas de persistencia).	Descentralizada (cada servicio audita, pero la lógica puede duplicarse).	Alta (trazabilidad encapsulada en cada microservicio).
Escalabilidad operativa	Media (involucra replicar todo el monolito).	Media (escala la capa completa, consumiendo recursos ineficientemente).	Alta (escalabilidad independiente por servicios funcionales).	Alta (escalabilidad independiente de servicios funcionales con bajo impacto sobre otros dominios).





tipo de trámite. Vacaciones, licencias con y sin remuneración y permisos poseen criterios de cálculo, plazos y niveles de aprobación que evolucionan de forma independiente. Esta heterogeneidad normativa convierte cada dominio funcional en un candidato natural para un microservicio autónomo, ya que permite modificar las reglas de un tipo de solicitud sin afectar los demás, mantener trazas de auditoría independientes por dominio y acotar el alcance de las pruebas de regresión ante cada cambio regulatorio. Aunque esta arquitectura introduce una mayor complejidad operativa y de despliegue respecto a un monolito modular, dicha complejidad se considera justificable debido a los beneficios obtenidos en mantenibilidad, trazabilidad y adaptación normativa.

Para que la arquitectura tecnológica responda fielmente a la realidad institucional, el desarrollo de software no puede ejecutarse de forma aislada de la ingeniería de procesos. La digitalización efectiva en el sector público requiere rediseñar los flujos administrativos antes de automatizarlos (Quiñónez Rentería & Segarra Jaime, 2025). Aunque estudios previos han utilizado modelos As-Is/To-Be para optimizar procesos organizacionales, estos no describen la transición hacia una arquitectura de software concreta (Ferreira et al., 2026). Por otro lado, las propuestas centradas exclusivamente en el desarrollo tecnológico suelen automatizar procesos sin una revisión organizacional previa (Wang, 2023; Juma Alba et al., 2024). En este contexto, la contribución científica de la presente investigación consiste en la validación empírica de una propuesta de software para la gestión de vacaciones, licencias y permisos en el sector público, desarrollada mediante la integración de As-Is/To-Be, Extreme Programming, Arquitectura Hexagonal y Microservicios. El modelo As-Is/To-Be permite rediseñar los procesos conforme a la normativa vigente, XP facilita la implementación iterativa y adaptable de dichos procesos, la Arquitectura Hexagonal protege la lógica de negocio frente a cambios tecnológicos y los Microservicios posibilitan la evolución independiente de cada dominio funcional. Esta integración constituye una propuesta metodológica orientada al desarrollo de sistemas administrativos públicos donde la trazabilidad, la auditoría y la adaptación normativa son requisitos críticos.

Además de los criterios de mantenibilidad y evolución normativa, la arquitectura propuesta incorpora mecanismos orientados a la protección de información sensible y al cumplimiento de requisitos de auditoría. Al gestionar datos sensibles de servidores públicos, el diseño prioriza la privacidad desde la arquitectura. El patrón hexagonal blindo el dominio donde residen las reglas de negocio frente a accesos externos, asegurando el aislamiento de la lógica de negocio respecto a interfaces de red, bases de datos y servicios externos.

En el presente trabajo se estudiaron las solicitudes de vacaciones, licencias y permisos de los empleados en una agencia gubernamental, que por razones de protección de identidad se denominará Institución Sujeta a Estudio (ISE). El sistema empleado anteriormente funcionaba con limitantes, que van desde errores en los cálculos, hasta la duplicidad de registros, además de retrasos en la consolidación de los datos y poca visibilidad sobre el estado de las peticiones. También la ausencia de integración entre herramientas dificultaba la comprobación de saldos, la identificación de cruces de turnos y la supervisión en tiempo real del procedimiento administrativo. Además, fueron revisados formularios de acciones del personal para permisos generados desde enero de 2024 a septiembre de 2025 y se destacaron 1.997 registros que contenían este tipo de diligencias, correspondientes a 626 funcionarios públicos particulares. Este número equivale a unas 95 solicitudes por mes relacionadas con períodos vacacionales, cierres administrativos, y demandas operativas de las distintas oficinas. Estos resultados muestran que existen múltiples áreas que representan una alta carga de trabajo para el área de Talento Humano y, en consecuencia, que es fundamental encontrar una solución informática que permita centralizar toda esta información, automatizar las validaciones, controlar los estados de las solicitudes y mantener la trazabilidad de las operaciones.

A partir de esta problemática, se plantea la siguiente pregunta de investigación:

¿En qué medida el desarrollo y la validación de un sistema informático, construido mediante el modelo As-Is/To-Be, la metodología Extreme Programming y una arquitectura de microservicios con patrón hexagonal, contribuyen a mejorar el control, la validación y la trazabilidad en la gestión de vacaciones, licencias y permisos en una institución pública?

Coincidiendo con este interrogante, la finalidad del estudio fue desarrollar un sistema informático para la gestión centralizada de solicitudes que permitiera también automatizar el cálculo de los saldos, validar cruces de turnos y registrar la trazabilidad del proceso administrativo. Como respuesta a esta necesidad, se desarrolló GIATH (Gestión Inteligente Automatizada de Talento Humano), una propuesta informática orientada a mejorar la administración de vacaciones, licencias y permisos en la ISE. La solución fue desarrollada mediante la metodología XP y una arquitectura de microservicios con patrón hexagonal. El rediseño de los procesos institucionales se realizó utilizando el modelo As-Is/To-Be. La aplicación fue construida con Java 21, Spring Boot, Angular, TypeScript, SQL Server 2022 y MongoDB.

En la Figura 1(a) se presenta el ciclo de desarrollo de Extreme Programming (XP), compuesto por las fases de planificación, diseño, codificación, prueba y lanzamiento. En la Figura 1(b) se muestra el modelo As-Is/To-Be, utilizado para diagnosticar el proceso actual, identificar brechas, definir el proceso objetivo y evaluar las mejoras propuestas (Ferreira et al., 2026). La integración de ambos enfoques permitió estructurar la propuesta GIATH a partir del análisis del proceso institucional existente y desarrollar la solución mediante iteraciones cortas con validación continua de los requerimientos (Muñoz Solórzano et al., 2024; Pambudi & Apriandari, 2023).

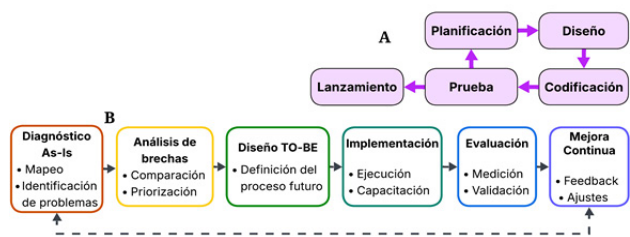


Figura 1. Representación de metodologías de desarrollo de software.

(a) Metodología XP; (b) Modelo As-Is/To-Be.

Fuente: Elaboración con base en (Ferreira et al., 2026; Merchán-Narváez et al., 2024)

La aplicación de As-Is/To-Be permitió definir el proceso objetivo para la gestión de vacaciones, licencias y permisos, mientras que XP proporcionó el marco de desarrollo iterativo utilizado para construir la solución propuesta (Cañizares Larco et al., 2025; Muñoz Solórzano et al., 2024; Shrivastava et al., 2021).

2. Materiales y Métodos

Este estudio se realizó en el área de Talento Humano de la Institución Sujeta a Estudio (ISE), ubicada en la ciudad de Machala, Ecuador, durante el lapso de septiembre de 2025 a febrero de 2026. Se analizó el procedimiento actual de administración de vacaciones, licencias y permisos del personal, así como los requerimientos de mejora y automatización, con el fin de diseñar y desarrollar una propuesta tecnológica que permitiera superar las limitaciones detectadas. Como parte del proceso, se contempló una evaluación cuantitativa del producto para determinar su funcionalidad, su rendimiento técnico y su aporte al proceso administrativo.

Para el desarrollo del sistema se aplicó una estrategia metodológica que integró el modelo As-Is/To-Be y la metodología Extreme Programming (XP), implementada mediante una arquitectura hexagonal fundamentada en microservicios. La decisión de adoptar dicha combinación se basó en la complementariedad de ambos enfoques: por un lado, el modelo As-Is/To-Be permitió analizar el estado actual del proceso y proponer un rediseño sustentado en evidencia (Bosco et al., 2024); por otro, XP facilitó el desarrollo iterativo, incremental y adaptativo del software

(Bautista-Villegas, 2022). Esta integración resultó pertinente para un entorno institucional caracterizado por requerimientos operativos y normativos susceptibles de cambio, así como por la necesidad de gestionar información sensible bajo estrictos protocolos de privacidad.

La Figura 2 consolida las actividades de los enfoques descriptivo y prescriptivo. Las fases del modelo As-Is/To-Be identificadas con etiquetas color lila. En este contexto, la fase As-Is se centró en el levantamiento del proceso vigente, la marcación de problemas y la detección de oportunidades de mejora, mientras que la fase To-Be sirvió para definir el rediseño de los flujos, las reglas de negocio y la arquitectura del sistema. Por su parte, las actividades propias de la metodología XP (etiquetas de color rojo) abarcaron la planificación, el diseño, la codificación y las pruebas, utilizando iteraciones cortas, retroalimentación continua y validaciones funcionales progresivas (Merchán-Narváez et al., 2024). En conjunto, estas fases permitieron articular el inicio, diseño, desarrollo y validación de la solución tecnológica dentro de un esquema iterativo. De este modo, se garantizó la construcción de un sistema alineado con las necesidades del área de Talento Humano, estructurado para someterse a evaluaciones métricas de mantenibilidad, escalabilidad y trazabilidad operativa.

La elección de una arquitectura de microservicios con patrón hexagonal se definió en contraposición a la opción de un monolito modular. Aunque un modelo modular habría proporcionado una separación lógica de los componentes, carecía de los beneficios operacionales críticos para este proyecto, el aislamiento de fallos, la escalabilidad específica por dominio y la protección robusta de la lógica de negocio siendo aspectos que el área técnica determinó como fundamentales para garantizar la evolución y sostenibilidad del sistema en el tiempo.

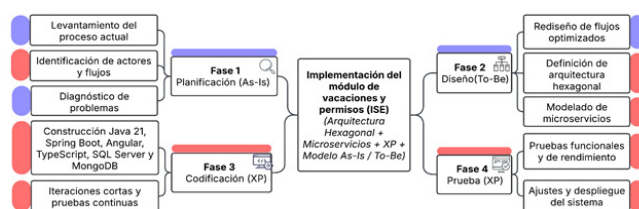


Figura 2. Fases y actividades del modelo As-Is/To-Be (color lila) e integración con la metodología XP (color rojo). Esta metodología fusionada se aplicó en el desarrollo de la aplicación propuesta de vacaciones y permisos del talento humano (GIATH).

Fuente: Los autores.

2.1 Fase de Planificación y levantamiento As-Is

Durante la planificación se identificaron los actores involucrados en el proceso de solicitudes (Tabla 3), estableciendo sus responsabilidades específicas dentro del sistema GIATH en apego a los perfiles de seguridad institucional.

Tabla 3. Actores Involucrados en el Sistema GIATH.
 Fuente: Los autores

Actor	Descripción
Servidor Judicial (solicitante)	Registra, adjunta documentos y envía solicitudes de permisos, vacaciones y licencias.
Analista de Talento Humano (TH)	Recibe, revisa, valida y tramita las solicitudes; comunica resultados al solicitante.
Coordinador de Recursos Humanos (RRHH)	Decide autorizaciones excepcionales y firma aprobaciones finales cuando corresponde.
Auditor / Control interno	Consulta trazabilidad, verifica cumplimientos y genera reportes para control y auditoría.
Administrador técnico	Gestiona catálogos, usuarios, roles y brinda soporte técnico y despliegue del sistema.

Elaborado a partir de reuniones de trabajo entre los actores involucrados, en la Tabla 4 se expone una historia de usuario global que sintetiza la funcionalidad principal que se espera del sistema. Esta historia fue el ancla para el plan de iteraciones dentro de XP.

De la historia de usuario general y de la información obtenida durante las sesiones de requerimientos se generó un modelo del proceso actual (As-Is). El diagrama resultante muestra las actividades de los diferentes actores, así como las interacciones y validaciones durante la gestión de solicitudes (Figura 3). Se analizaron para identificar distintas ineficiencias, tales como validaciones manuales, búsquedas fragmentadas de información relativa a servidores, normativa aplicable a permisos y licencias, y también registros de solicitudes incompletos o inconsistentes.

Tabla 4. Historia de usuario principal para la gestión de permisos y vacaciones.
 Fuente: Los autores

Elemento	Detalle
Como	Servidor Judicial, Analista de Talento Humano y Coordinador/a de Recursos Humanos.
Cuando	Cada vez que un servidor gestione licencias (remuneradas/no remuneradas), vacaciones o permisos por horas.
Necesito	Un sistema parametrizable y seguro que centralice la gestión y validación (motor de reglas) de solicitudes de vacaciones, licencias (remuneradas/no remuneradas) y permisos por horas, y que registre un historial inmutable de acciones.
Para	Estandarizar el registro único de solicitudes, automatizar y auditar el cálculo de saldos, prevenir cruces de turnos, simplificar la gestión y facilitar la adaptación a cambios normativos.

Las insuficiencias identificadas en el proceso As-Is se tradujeron en requisitos funcionales para el GIATH. En la Tabla 5 se muestra la matriz de requisitos priorizados del sistema, donde se indica el identificador, la prioridad y el criterio de aceptación que valida el requisito.

Tabla 5. Matriz de requisitos funcionales priorizados del sistema.
 Fuente: Los autores

ID	Requisito breve	Prioridad	Criterio de aceptación
R-01	Registro único de solicitudes	Alta	El sistema genera un folio único y asigna estado inicial "Nuevo".
R-02	Validación de saldo y turnos	Alta	Bloqueo automático si hay saldo insuficiente o cruce de turnos.
R-03	Motor de reglas parametrizable	Media	El administrador puede crear o modificar reglas sin necesidad de despliegue.
R-04	Seguimiento de estados	Alta	La solicitud avanza por los estados definidos: "Nuevo", "En revisión", "Aprobado", "Rechazado".
R-05	Notificaciones automáticas	Media	El sistema informa al solicitante y analista cuando existe actualización o revisión pendiente.
R-06	Bitácora inmutable de transacciones	Alta	Registra usuario, fecha y hora de cada acción (no editable).
R-07	Gestión de adjuntos en solicitudes	Media	El sistema valida y almacena archivos adjuntos según tipo de solicitud.
R-08	Acceso diferenciado por rol (seguridad)	Alta	Los permisos del usuario dependen de su perfil (solicitante, analista, auditor, etc.).

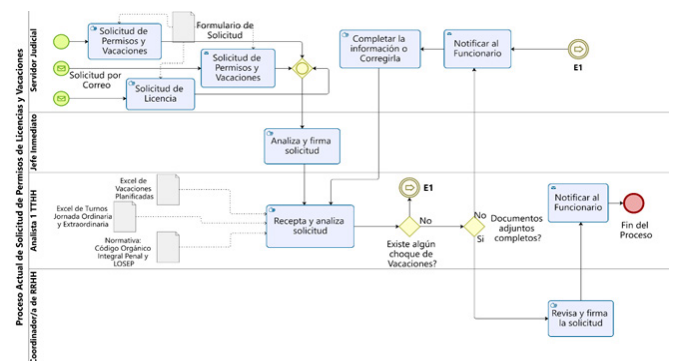


Figura 3. Diagrama del proceso de solicitudes de permisos y vacaciones en el departamento de Talento Humano (As-Is).
 Fuente: Los autores.

Como resultado de esta fase se elaboró el diagrama de casos de uso (Figura 4), el cual muestra la interacción de los actores con las funcionalidades de la plataforma. El sistema GIATH central se compone de módulos internos comunicados a través de interfaces definidas: gestión de solicitudes (registro y manejo de adjuntos); soporte y auditoría (registro en bitácora inmutable, administración de usuarios, reportes y notificaciones); validaciones y reglas (motor de reglas normativas, validación de saldos y cruces de turnos); y tramitación y autorización (revisión y aprobación/rechazo de solicitudes).



Figura 4. Diagrama de casos de uso de la plataforma GIATH.
Fuente: Los autores.

2.2 Fase de Diseño y elaboración To-Be

En cuanto a la etapa de diseño, basados en los requerimientos, se definió el flujo del proceso To-Be de GIATH. La Figura 5 muestra la propuesta que concentra la gestión en validaciones automáticas y reglas parametrizadas. El esquema propone reemplazar tareas manuales por procesos digitales sincronizados entre los participantes del sistema.

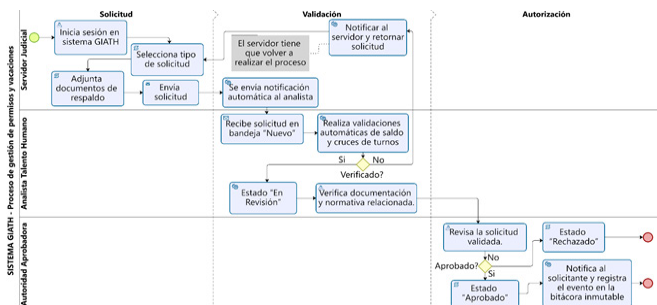


Figura 5. Diagrama de flujo del proceso de gestión de permisos y vacaciones propuesto (To-Be).
Fuente: Los autores.

La Tabla 6, muestra el CRC (Clase, Responsabilidades y Colaboraciones) de solicitud, detallando las responsabilidades y las clases colaboradoras que gestionan el registro, validación, manejo de estados, notificaciones y la rastreabilidad de las operaciones del sistema.

Al final de la fase se estableció la arquitectura de GIATH basada en microservicios y soportada por el patrón hexagonal (Figura 6). El acceso al sistema se realiza en la capa de presentación mediante una aplicación frontend desarrollada en Angular y

Tabla 6. Tarjeta CRC – Clase Solicitud Permiso Vacaciones.
Fuente: Los autores

Clase: Solicitud Permiso Vacaciones	
Responsabilidad	Colaboración
Registrar nuevas solicitudes (vacaciones, permisos o licencias).	SolicitudRepository – Persiste la solicitud
Asociar documentos adjuntos según el tipo de solicitud.	DocumentService – Gestiona archivos adjuntos ValidadorDocumentos – Verifica tipos permitidos
Validar saldo disponible y cruces de turnos mediante el motor de reglas.	MotorReglas – Aplica validaciones automáticas ServicioSaldos – Consulta saldos actuales
Actualizar estados del flujo (Nuevo, En revisión, Aprobado, Rechazado).	BitacoraTransacciones – Registra cambio de estado WorkflowService – Gestiona transiciones
Enviar notificaciones automáticas a los actores involucrados.	NotificacionService – Comunica cambios de estado TemplateService – Genera mensajes
Registrar transacciones en la bitácora inmutable.	BitacoraTransacciones – Mantiene trazabilidad AuditoriaService – Gestiona auditoría
Permitir la consulta del historial completo por servidor o analista.	ReportService – Genera consultas

TypeScript. Desde esta interfaz, las peticiones se transmiten al backend a través de protocolos HTTP/REST. Para garantizar la seguridad de la información y el cumplimiento normativo en cuanto a protección de datos, se implementó un esquema de autenticación y control de acceso basado en OIDC (OpenID Connect) y tokens JWT (JSON Web Tokens).

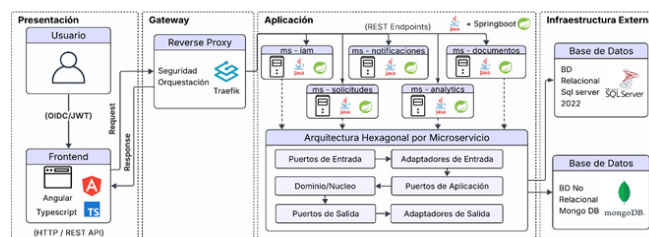


Figura 6. Modelo de la Arquitectura Hexagonal basada en microservicios del sistema GIATH.
Fuente: Los autores.

Las solicitudes pasan a ser atendidas a través de una capa gateway implementada con Traefik, el cual actúa como reverse proxy. Este módulo actúa como un punto único de acceso al sistema, provee enrutamiento para las peticiones y permuta cada solicitud al microservicio adecuado. Por lo tanto, el frontend no habla directamente con un individual servicio, en vez de que habla a un punto de entrada limitado.

Los microservicios que son el núcleo del sistema se encuentran en

esta capa de aplicación. El servicio ms-iam se ocupa de usuarios, roles, permisos y control de acceso; ms-solicitudes gestiona el alta, validación y proceso de las solicitudes; y ms-documentos gestiona el almacenamiento y consulta de anexos. También se contempla ms-analytics como servicio adicional para tableros e indicadores, con integración desacoplada al flujo central del sistema.

Cada microservicio es responsable de su propagación basada en la arquitectura hexagonal. En esta arquitectura, los adaptadores de entrada son los clientes de las peticiones REST. Los puertos de la aplicación son los casos de uso, el dominio es donde residen las reglas de negocio y los puertos de salida son los que se comunican con bases de datos u otros servicios externos. Esta división permite mantener la lógica principal lejos de la interfaz y la infraestructura de los detalles de los métodos.

Finalmente, la infraestructura de persistencia se clasificó en dos tipos de almacenamiento, aplicando políticas de seguridad de la información y protección de datos. La información transaccional relacional (usuarios, contraseñas encriptadas mediante algoritmos de hash seguros como Bcrypt, permisos, estados y trazabilidad de auditoría) se almacena en SQL Server 2022, asegurando integridad referencial y soporte para encriptación de datos en reposo. Por otro lado, los archivos adjuntos que contienen información sensible y privada de los servidores públicos se gestionan en MongoDB a través de GridFS. Esto optimiza el manejo de documentos no estructurados sin sobrecargar la base relacional, restringiendo su descarga y visualización estrictamente a través de las validaciones de seguridad del backend.

2.3 Fase de Codificación

La fase de codificación del backend de la plataforma se ejecutó aplicando los principios ágiles de XP. Se adoptó una arquitectura orientada a microservicios, con el patrón hexagonal en el interior, teniendo como resultado una solución desacoplada, mantenible y escalable. Utilizando Java 21 y Spring Boot, se consolidó una estructura modular que distingue claramente la lógica de negocios del acceso a los datos y de las interfaces para la comunicación. Para garantizar el cumplimiento normativo y la gestión de privacidad, se integró el framework Spring Security. Este componente intercepta todas las peticiones entrantes, valida la firma criptográfica de los tokens JWT y aplica el Control de Acceso Basado en Roles (RBAC), asegurando que solo los perfiles autorizados (ej. Analistas o Coordinadores) puedan ejecutar acciones sensibles dentro de los microservicios. Adicionalmente, cada microservicio operó con su propio repositorio y esquema de base de datos.

Tal como se muestra en la Figura 7, cada microservicio se

estructuró en capas con funciones definidas en cada una. La capa “domain” concentra el núcleo del negocio e incluye modelos, servicios y excepciones que representan las reglas e invariantes del sistema sin depender de frameworks ni de tecnologías concretas. La capa “application” se ocupa de los casos de uso y coordina los flujos a través de puertos de entrada y salida, por lo que la lógica del dominio está desacoplada de la persistencia y de los servicios externos. La capa “infrastructure” provee las implementaciones concretas de persistencia, entidades JPA y manejo global de excepciones, permitiendo actualizar tecnologías subyacentes sin alterar la normativa institucional. Paralelamente, la capa “web” expone la funcionalidad vía API REST mediante contratos de comunicación estrictamente validados. La carpeta resources aloja la configuración por ambientes y las migraciones versionadas a través de Flyway.

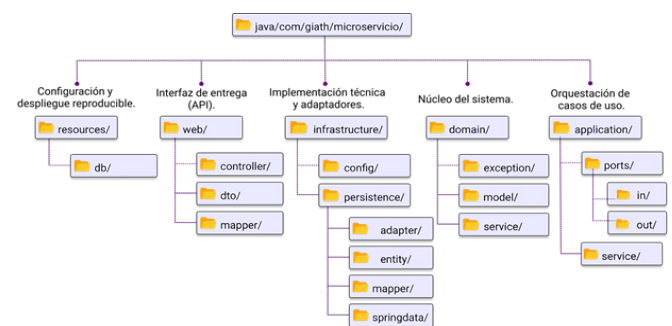


Figura 7. Estructura de carpetas del proyecto GIATH.

Fuente: Los autores.

En este punto, el modelo de datos que apoyó la lógica de negocio estaba en su lugar. La implementación presentó servicios de aplicación, adaptadores de persistencia y endpoints REST para manejar los ciclos de vida de las peticiones, de acuerdo con el modelo relacional del sistema. El modelo de datos se orienta a la normalización, separación de entidades de negocio transaccionales, catálogos y elementos paramétricos junto a la definición de las relaciones con claves primarias y foráneas para permitir un bajo acoplamiento, integridad referencial, consistencia y facilidad de mantenimiento. Las reglas de negocio relacionadas con balances, validaciones y requisitos fueron resueltas en componentes separados, que entonces no fueron incorporados en el código.

Además, se añadieron mecanismos de trazabilidad y control a través de la consignación de balances previos y posteriores a cada operación, identificadores únicos y marcas temporales de creación y actualización. La migración del esquema fue gestionada con migraciones versionadas por medio de Flyway, integradas al proceso de construcción y despliegue

del microservicio, lo que le permitió propagar los cambios a través de los ambientes y mantener un historial de cambio en el esquema. Con el desarrollo del backend y del modelo de datos que sustenta, el sistema logra un nivel de madurez funcional suficiente para probar el comportamiento, la aplicación de las reglas de negocio y la interacción de las diferentes capas y componentes construidos.

2.4 Fase de Pruebas

La Figura 8 sintetiza el proceso de validación del sistema mediante tres tipos de pruebas:

- **P1-Prueba funcional**, orientada a comprobar el comportamiento funcional del sistema de gestión de solicitudes, la aplicación de las reglas de negocio, el control de estados y la persistencia de la información. Este procedimiento permitió contrastar los resultados obtenidos con los criterios de aceptación definidos, asegurando la coherencia del flujo operativo y la correcta aplicación de las validaciones establecidas.

- **P2-Prueba de rendimiento**, destinada al rendimiento frente a varias cargas concurrentes. Esto hizo posible conocer el comportamiento del sistema ante aumentos de demanda y comprobar su estabilidad en ambientes similares a los institucionales.

- **P3-Prueba UI/UX**, destinada a la evaluación de la interfaz y la experiencia de uso de los expertos. Esta encuesta fue un cuestionario digital aplicado al finalizar la interacción con el sistema en los escenarios específicos.

Para evaluar el logro de los objetivos en las pruebas técnicas P1 y P2 se emplearon herramientas especializadas. La validación funcional se apoyó Swagger UI, lo cual permitió inspeccionar y verificar los endpoints OpenAPI con base en la matriz de requisitos de la Tabla 5. Para las pruebas de rendimiento se utilizó Apache JMeter, ejecutando pruebas concurrentes para observar el comportamiento del backend; el criterio de aceptación técnico se fijó en un tiempo medio de respuesta menor a 1 segundo con

una tasa de error del 0% para todos los escenarios (Papalkar & Alvi, 2024; Pilicita Garrido et al., 2020).

Respecto a la prueba P3, el cuestionario contiene 10 ítems valorados mediante una escala Likert de cinco niveles (1 – Totalmente en desacuerdo, 2 – En desacuerdo, 3 – Indiferente, 4 – De acuerdo y 5 – Totalmente de acuerdo). La estructura de la encuesta (Tabla 7) toma como base el modelo de calidad de la norma ISO/IEC 25010 y abarca las dimensiones de Idoneidad Funcional (IF), Eficiencia de Desempeño (ED), Usabilidad (US), Fiabilidad (FI) y Seguridad (SE). El umbral de aceptación para este análisis exige una puntuación media grupal superior a 4,0 (“De acuerdo”) en cada métrica evaluada (Cortés Rojas, 2024).

Para la recolección estructurada de información se implementó un formulario digital a través de Google Forms (Manggabernani & Darlis, 2024). Previo a su aplicación, el instrumento fue validado mediante juicio de expertos para garantizar la claridad y pertinencia de los ítems, un procedimiento metodológico indispensable para asegurar el diseño técnico de cuestionarios organizados y la obtención de datos institucionales precisos (Bedolla Solano et al., 2026). Asimismo, para fortalecer el rigor del análisis estadístico, se determinó evaluar la consistencia interna y confiabilidad del cuestionario mediante el coeficiente Alfa de Cronbach, estableciendo un umbral mínimo aceptable de 0,7.

La muestra cuantitativa estuvo conformada por 22 profesionales del área de Tecnologías de la Información e Ingeniería de Software. La captación se realizó bajo un muestreo no probabilístico intencional. Los criterios de inclusión exigieron un mínimo de tres años de experiencia en desarrollo o aseguramiento de calidad (QA) y un dominio intermedio-avanzado en arquitecturas de sistemas institucionales. La participación mantuvo carácter voluntario y anónimo. Los datos recolectados se sometieron a estadística descriptiva para perfilar la percepción técnica sobre prevención de errores, tiempos de carga y restricciones de acceso.

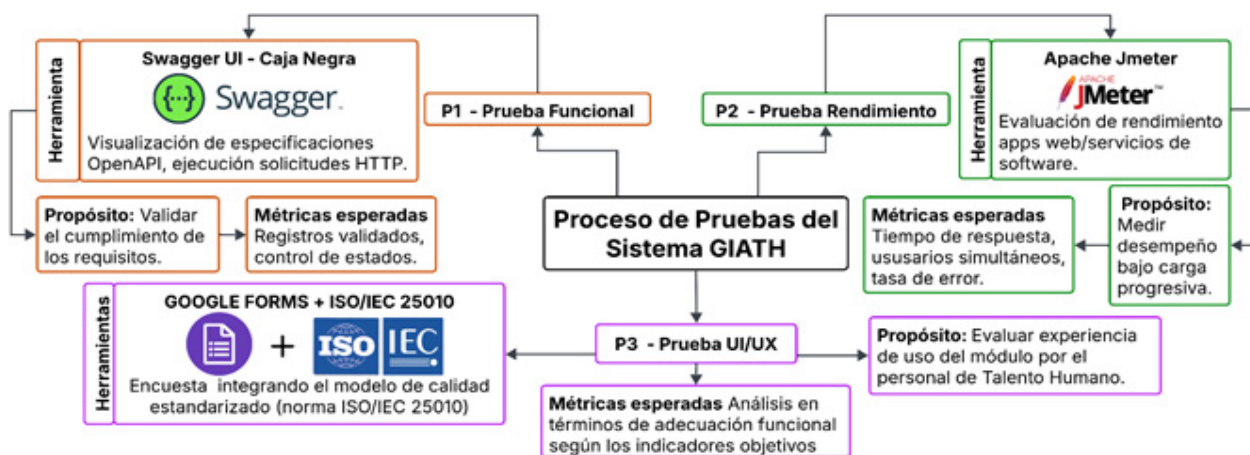


Figura 8. Pruebas al sistema GIATH.

Fuente: Los autores.

Tabla 7. Cuestionario de encuesta basado en ISO/IEC 25010.
Fuente: Los autores

Métrica	Ítem	Enunciado	1	2	3	4	5
AF – Completitud funcional	P1	El sistema permite gestionar completamente una solicitud.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AF – Corrección funcional	P2	El sistema aplica correctamente las validaciones y cálculos.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AF – Pertinencia funcional	P3	Las funciones son apropiadas para el proceso institucional.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ED – Comportamiento temporal	P4	El sistema responde en tiempos adecuados.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ED – Capacidad	P5	El sistema funciona bien con varios usuarios al mismo tiempo.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
US – Facilidad de Aprendizaje	P6	Es fácil registrar y revisar una solicitud.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
US – Protección contra errores	P7	El sistema previene errores al ingresar información.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
US – Presentación de interfaz	P8	La información se muestra de forma clara y ordenada.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FI – Madurez	P9	El sistema funciona sin fallos durante su uso.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SE – Control de acceso	P10	El acceso está restringido según el rol del usuario.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Resultados y Discusión

3.1 Resultados

3.1.1 Resultados de encuesta a expertos

El análisis de confiabilidad del instrumento arrojó un coeficiente Alfa de Cronbach de 0,87 lo cual confirma una alta consistencia interna de la escala utilizada en el estudio. La Figura 11 presenta una combinación de los resultados porcentuales obtenidos en cada pregunta del cuestionario de la Tabla 7. Los datos muestran una concentración en las categorías De acuerdo y Totalmente de acuerdo sin registros en las opciones de rechazo. Esta tendencia respalda la aceptación técnica del sistema por parte de los evaluadores. En la pregunta P10 referida al control de acceso basado en roles el 100 % de la muestra seleccionó Totalmente de acuerdo confirmando la eficacia del esquema de permisos implementado. Respecto a la claridad en la presentación de la información evaluada en P8 el 90,9 % marcó Totalmente de acuerdo y el 9,1 % De acuerdo. En los ítems P6 y P7 sobre la facilidad de registro y la prevención de errores el 81,8 % optó por Totalmente de acuerdo y el 18,2 % por De acuerdo. Las cifras evidencian que el diseño de la interfaz y las validaciones del sistema facilitan una interacción controlada y comprensible.

En las métricas correspondientes a las preguntas P1 y P4 el 72,7 % de los participantes indicó Totalmente de acuerdo y el 27,3 % señaló (De acuerdo) lo que refleja que el flujo principal de la aplicación se encuentra bien realizado y la agilidad en los tiempos

de respuesta. Para los ítems P2 P5 y P9 la máxima valoración alcanzó el 90,9 %. Por su parte la pregunta P3 obtuvo un 81,8 % de respuestas afirmativas y concentró el porcentaje restante en la categoría Indiferente. Este hallazgo sugiere que, aunque la plataforma cumple con los requisitos institucionales básicos existen oportunidades de mejora en la pertinencia de ciertas funciones. Las métricas verifican que el aplicativo satisface las necesidades operativas y optimiza el trabajo administrativo mediante validaciones automáticas y trazabilidad de acciones.

ye una validación técnica preliminar al aplicarse a profesionales de TI/Software y no a los usuarios finales del área administrativa. Para mitigar esta limitación y cumplir con el principio de participación continua del cliente de la metodología XP, el proyecto integró a dos ingenieros delegados por el departamento de Talento Humano. Estos profesionales actuaron como expertos del dominio a través de reuniones semanales. En estos encuentros suministraron la información operativa necesaria y verificaron directamente el funcionamiento del aplicativo en cada iteración. Esta dinámica de trabajo garantizó que el flujo del sistema y las reglas de negocio se ajusten a los procesos reales de la institución manteniendo la estricta confidencialidad de la información interna. El proceso permitió asegurar la utilidad de la plataforma y derivó la ejecución de encuestas al resto del personal para futuras fases de implementación.

Los promedios mostrados en la Figura 12 permiten estimar cómo se comportaría el sistema con las métricas de calidad de la ISO/

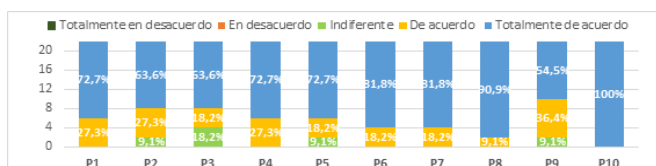


Figura 9. Distribución de respuestas por ítem (escala Likert).

Fuente: Los autores.

IEC 25010. Todos los valores evaluados (por encima de 4,4 de 5) indican un grado alto en cada uno de los criterios, sin ninguna desviación que modifique su funcionamiento en general. La máxima puntuación para control de acceso indica que los mecanismos de seguridad son lo suficientemente fuertes como para un entorno institucional, mientras que las puntuaciones de usabilidad indican una interacción simple y constante en las pruebas.

Los indicadores un poco más bajos no significan que el sistema es no funcional, pero ayuda a encontrar áreas de mejora conforme se conoce más del sistema. Este comportamiento es normal en los sistemas o aplicaciones recién estrenados; sin embargo, conforme avanza su uso en cuanto a experiencia práctica, se realizan ajustes de forma paulatina en sus funcionalidades. Estos resultados demostraron un alto grado de coherencia entre los subsistemas y una respuesta estable a las demandas del área de recursos humanos.

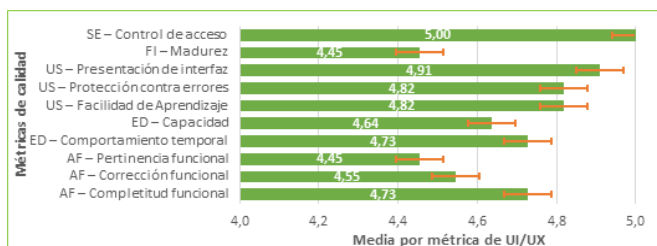


Figura 10. Promedio por métrica de calidad (ISO/IEC 25010).

Fuente: Los autores.

Tabla 8. Pruebas funcionales del módulo de solicitudes (endpoints evaluados, códigos HTTP y evidencias).

Fuente: Los autores

Aspecto validado	Método y endpoint	Escenario de prueba	Código HTTP	Evidencia registrada	Resultado
Creación de solicitud con datos válidos	POST /api/solicitudes/solicitud	Envío de solicitud con datos completos y rango de fechas permitido	201	Identificador generado (id = 1002), folio y fechaCreacion en el registro de auditoría	Registro exitoso
Control de integridad de datos (validación de rangos de fecha)	POST /api/solicitudes/solicitud	Envío de solicitud con fechaHasta menor que fechaDesde	400	Mensaje de validación: "fechaHasta no puede ser menor que fechaDesde"	Solicitud rechazada por inconsistencia en las fechas.
Actualización de estado dentro del flujo de tramitación	PUT /api/solicitudes/solicitud/{id}/estado	Cambio de estado de una solicitud existente (id = 1002)	200	Actualización de campos de auditoría (usuarioModificacion, fechaModificacion) y estado	Estado actualizado

3.1.2 Resultados de funcionalidad

Los resultados presentados en la Tabla 8 muestra casos de prueba del módulo de solicitudes. En creación válida (POST /api/solicitudes/solicitud con datos completos), el servidor retorna 201 y graba el ID generado, folio y fechaCreacion en la auditoría, verificando el éxito del registro. Ante fechas incorrectas (mismo endpoint, fechaHasta < fechaDesde), devuelve 400 con el mensaje de validación correspondiente, evitando la persistencia. Al modificar el estado de una solicitud (PUT /api/solicitudes/solicitud/1002), se devuelve 200 y además se modifican los campos usuarioModificacion, fechaModificacion junto con el estado, representando la transición. En cada situación registra el endpoint, código y audit data para probar que el sistema mantiene los datos consistentes y falla como se espera.

3.1.3 Resultados de pruebas de rendimiento

El sistema fue sometido a pruebas de rendimiento mediante pruebas de estrés con Apache JMeter en el microservicio de peticiones (ms-solicitudes) y en el microservicio de identidades (ms-iam). Probada en local con Java 21, Spring Boot y base de datos [SQL Server 2022 + MongoDB] bajo SO Windows 11 Pro, procesador AMD Ryzen 5 7520U y 16 GB de RAM. La carga se construyó en cinco escenarios con 20, 40, 70, 100 y 350 usuarios concurrentes, 10 iteraciones por usuario virtual y ramp-up progresivos, tal como se muestra en la Tabla 9.

Tabla 9. Configuración de las pruebas de carga.

Fuente: Los autores

Prueba	Usuarios concurrentes	Ramp-up (s)	Iteraciones por usuario	Muestras
1	20	5	10	200
2	40	10	10	400
3	70	15	10	700
4	100	20	10	1000
5	350	30	10	3500

Para el análisis, se consideraron el promedio de tiempo de respuesta, la mediana, los percentiles 90 y 95, la desviación estándar, el throughput y la tasa de error, y los resultados se muestran en la Tabla 10. Dichas métricas hacen posible evaluar el rendimiento temporal del servicio y su procesamiento a diferentes niveles de concurrencia.

Los resultados en la Tabla 10 revelan que el tiempo promedio varió de 110 ms a 210 ms para 20 y 40 usuarios simultáneos, con medianas entre 60 ms y 140 ms; el percentil 95 no superó los 680 ms en estos casos, el throughput alcanzó hasta 74.2 req/s, y no hubo errores al ejecutar. Bajo la configuración de 70 usuarios concurrentes, el tiempo promedio subió a 472 ms y el percentil 95 fue de 1439 ms, mientras que con 100 y 350 usuarios el promedio se elevó a 780 ms y 3200 ms, respectivamente, y el percentil 95 alcanzó 2350 ms y 6800 ms, indicando una mayor demanda hacia los recursos del sistema bajo carga pesada. La tasa de error se mantuvo en 0.0 % en todos los escenarios, por lo que las solicitudes fueron procesadas sin incidencias. La Figura 13 puede utilizarse para comparar estos resultados con el promedio, la mediana, el percentil 95 y el throughput. Para complementar

estas métricas, la Tabla 11 presenta el consumo de recursos del servidor durante las mismas pruebas.

Se observa que la utilización de CPU crece de manera directamente proporcional a la carga, alcanzando un promedio del 98 % y picos del 100 % con 350 usuarios, lo que señala al procesador como el principal cuello de botella en el entorno de prueba local. La memoria RAM se mantiene dentro de los 16 GB disponibles, con un máximo de 5100 MB, sin síntomas de agotamiento ni intercambio excesivo. El tráfico de red, por su parte, escala linealmente con el número de peticiones, sin saturaciones anómalas. Estos indicadores confirman que, aun bajo estrés severo, el sistema permanece estable, aunque la latencia se eleva por la saturación de la CPU.

Finalmente, la Figura 11 desglosa el tiempo de respuesta del flujo de solicitudes discriminando la contribución de cada microservicio (ms-solicitudes, ms-iam) y el total combinado, tanto para el promedio como para el percentil 95, en los escenarios de 70, 100 y 350 usuarios.

Tabla 10. Resultados de rendimiento por escenario de carga.

Fuente: Los autores

#	Tiempo promedio (ms)	Usuarios concurrentes	Mediana (ms)	Percentil 90 (ms)	Percentil 95 (ms)	Desviación estándar (ms)	Throughput (req/s)	Tasa de error (%)
1	20	110	60	290	360	125.0	68.5	0.0
2	40	210	140	520	680	210.0	74.2	0.0
3	70	472	237	1158	1439	515.9	73.6	0.0
4	100	780	420	1850	2350	820.0	78.1	0.0
5	350	3200	2600	5400	6800	2100.0	95.3	0.0

Tabla 11. Consumo de recursos del servidor durante las pruebas de carga.

Fuente: Los autores

#	Usuarios	CPU (%) promedio	CPU (%) pico	RAM (MB) promedio	RAM (MB) pico	Red (KB/s) promedio	Red (KB/s) pico
1	20	38	65	820	980	510	890
2	40	55	82	1150	1380	840	1420
3	70	78	95	1720	2050	1820	2200
4	100	90	99	2400	2800	2580	3500
5	350	98	100	4200	5100	7200	9600

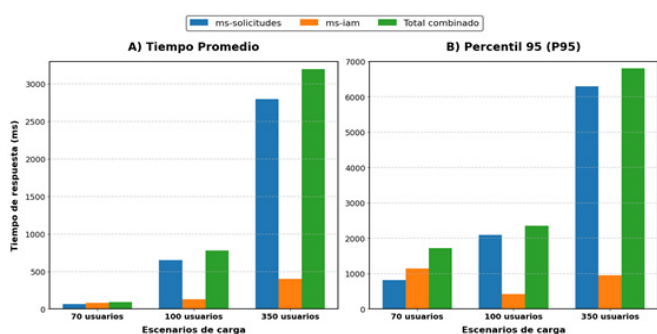


Figura 11. Desglose del tiempo de respuesta del flujo de solicitudes. (A) Tiempo promedio y (B) percentil 95 (P95), discriminado por microservicio (ms-solicitudes, ms-iam) y total combinado, para cargas de 70, 100 y 350 usuarios concurrentes. Fuente: Apache JMeter.

En la Figura 11 se observa que ms-solicitudes concentra la mayor parte de la latencia total, en particular en los percentiles 95 de 350 usuarios (6300 ms), mientras que ms-iam se mantiene por debajo de 1150 ms en P95 en todos los casos. Esta diferencia valida la decisión arquitectónica de desacoplar la gestión de identidades y señala a ms-solicitudes como el componente prioritario para futuras acciones de escalado horizontal o balanceo de carga.

Considerando los resultados de la Tabla 10, con usuarios concurrentes desde 20 hasta 100 el tiempo medio de respuesta del sistema se situó entre 110 ms y 780 ms (mediana de 60 ms a 420 ms), con un percentil 95 de hasta 2350 ms y un throughput que alcanzó 78,1 req/s. Para 350 usuarios, el tiempo promedio ascendió a 3200 ms (P95 = 6800 ms) y el throughput llegó a 95,3 req/s. No se produjeron errores en ningún escenario (tasa de error 0,0 %), lo que indica que el sistema mantiene la disponibilidad bajo carga, aunque la latencia aumenta debido a la saturación de la CPU observada en la Tabla 11.

3.1.4 Evidencias de Software

La digitalización del proceso de vacaciones y permisos mediante la arquitectura propuesta permitió superar las limitaciones operativas y los cuellos de botella característicos del modelo tradicional (As-Is). Para evidenciar el nivel de desacoplamiento de la solución (To-Be), la Figura 12 expone la orquestación de eventos en el backend a través de un diagrama de secuencia.

A diferencia del procedimiento previo, donde el servidor público debía ingresar la totalidad de sus datos de forma manual, el sistema delega la responsabilidad de validación a la interacción entre microservicios. Al originarse una petición, el microservicio de peticiones (ms-solicitudes) consulta directamente al gestor de identidades (ms-iam) para validar la jerarquía, los roles de acceso (RBAC) y determinar al jefe inmediato. Esta integración garantiza la inmutabilidad de la información institucional desde la creación del registro.

Asimismo, el patrón hexagonal aísla el cálculo automatizado de días y horas en la capa de dominio, independizándolo de la persistencia de datos y de la vista. Una vez que las reglas de negocio aprueban la transacción, el sistema persiste la asignación y dispara eventos asíncronos para notificar a los responsables en la cadena de aprobación. Este rediseño arquitectónico no solo reduce el ingreso manual de datos, sino que previene inconsistencias transaccionales, asegurando una trazabilidad auditable de cada decisión tomada en el flujo.

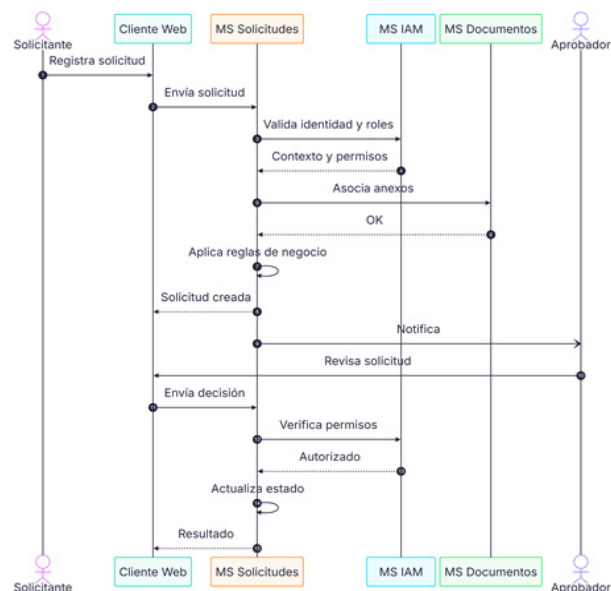


Figura 12. Diagrama de secuencia formal del flujo de solicitud de vacaciones y permisos. Se evidencia la orquestación de eventos entre la interfaz cliente, el microservicio de solicitudes y el gestor de identidades para la validación asilada de las reglas de negocio.

Fuente: Los autores.

Para dimensionar el impacto real de este rediseño arquitectónico, se contrastó el desempeño del nuevo flujo automatizado frente al procedimiento manual que reemplaza. Tal como se evidenció en los resultados de rendimiento (sección 3.1.3), el sistema procesa y valida las reglas de negocio de una solicitud en tiempos inferiores a 1.5 segundos, incluso en escenarios de alta concurrencia. La Tabla 12 recoge la comparación directa entre ambos modelos operativos.

Tabla 12. Comparación del tiempo total de registro de una solicitud entre el proceso manual y el sistema GIATH.

Método	Tiempo promedio por solicitud	Actividades principales involucradas
Manual (anterior)	~10 minutos (600 000 ms)	Descargar el formulario institucional, llenar a mano datos personales y fechas, firmar, escanear y enviar por correo electrónico o en físico.
GIATH (automatizado)	< 1.5 segundos (1500 ms)	Ingresar tipo, subtipo y motivo; el sistema precarga la identidad, calcula automáticamente los días u horas hábiles y notifica al jefe inmediato.

La comparación evidencia una reducción superior al 99.7% en el tiempo de registro operativo. Esta optimización es consecuencia directa de la eliminación de la captura manual de datos, delegando el cálculo de días hábiles y la notificación asíncrona a la lógica de los microservicios ilustrada en la Figura 12. De este modo, los tiempos de procesamiento obtenidos y el modelado lógico del sistema validan la transición hacia un esquema capaz de absorber la carga operativa de la institución.

3.2 Discusión

El presente estudio tuvo por objetivo desarrollar y evaluar un sistema informático para la administración de vacaciones, permisos y licencias, mediante una estrategia que integra el rediseño de procesos administrativos con una arquitectura de software desacoplada. Los resultados funcionales detallados en la Tabla 8 confirman el cumplimiento estricto de las reglas de negocio y la consistencia en el manejo de la información, un problema crítico en los flujos administrativos manuales. Esta verificación anticipada constituye una práctica estándar en la ingeniería de software para mitigar los riesgos de integración y reducir los costos de refactorización en etapas avanzadas del ciclo de vida del sistema.

La integración del análisis As-Is/To-Be con la metodología Extreme Programming permitió transformar un proceso institucional con validaciones manuales en un flujo digital organizado por reglas, estados y roles. Como evidencia cuantitativa de esta transición, la Tabla 12 de la sección 3.1.4 mostró que el tiempo de registro por solicitud se redujo desde aproximadamente 10 minutos en el proceso manual (que incluía descarga de formularios, llenado a mano, firma, escaneo y envío físico o por correo) hasta menos de 1.5 segundos en GIATH, lo que representa una reducción superior al 99.7%. Adicionalmente, la precarga automática de datos del empleado eliminó los errores de digitación en campos como nombres, cédula, cargo y dependencia, y el cálculo automático de días hábiles evitó inconsistencias en las fechas. Si bien la cuantificación exacta de errores y registros duplicados en el modelo As-Is no fue posible debido a la dispersión de la información en hojas de cálculo no centralizadas (lo que en sí mismo evidencia una limitación del proceso anterior), el análisis de los 1.997 registros históricos identificó ausencia de validaciones automáticas de saldos y cruces de turnos, así como la imposibilidad de rastrear el estado de las solicitudes en tiempo real. Estas brechas, documentadas en la fase de levantamiento, son las que el modelo To-Be resuelve mediante validaciones integradas en el motor de reglas, registro inmutable de cada acción y control de acceso diferenciado por rol. La cuantificación del ahorro en errores e inconsistencias

queda sujeta a medición en una fase de despliegue real, como se plantea en el trabajo futuro.

Desde un punto de vista técnico se optó por una arquitectura basada en microservicios con patrón hexagonal que desacopla la lógica de dominio de los detalles de infraestructura tales como la interfaz de usuario y la persistencia de datos. Esta segregación ofrece un buen soporte para el mantenimiento evolutivo y minimiza la posibilidad de que modificaciones en la interfaz, en la base de datos, o en servicios externos puedan tener impacto directo en las reglas centrales del procedimiento. El sistema resuelve los problemas de rigidez de sistemas monolíticos tradicionales y acoplamiento entre objetos. Un resultado importante de este diseño es el desempeño del motor de reglas (requisito R-03); siendo parametrizable, posibilita cambiar los umbrales por cantidades de días y los flujos de aprobación ante modificaciones en la legislación sin necesidad de modificar, compilar o desplegar el código fuente del dominio. Además, la separación de servicios independientes permite asegurar reproducibilidad de procesos claves con altísima concurrencia tales como cálculo automático de saldos y detección de cruces en turnos -estos procesos corren con recursos de cómputo aislados- y así optimiza tolerancia a fallos a nivel de infraestructura.

La Tabla 13 presenta una comparación entre GIATH y estudios previos, considerando el tipo de aporte, contexto, finalidad, validación, alcance operativo y aplicabilidad. Esta comparación permite ubicar el aporte del presente trabajo frente a investigaciones que también emplean soluciones tecnológicas para resolver problemas de gestión, automatización o control de información.

En relación con Santacruz-Erraez y Poma-Japón (2025), coincide en el empleo de microservicios para potenciar la escalabilidad, seguridad y una mejor separación de los componentes. Su trabajo propone una arquitectura para sistemas transaccionales financieros basada en tecnologías como Spring Boot, Spring Cloud, Eureka, Keycloak y Docker, validada mediante pruebas de rendimiento y seguridad. Ambos trabajos coinciden en que una arquitectura orientada a servicios facilita superar las limitaciones de los sistemas monolíticos, pero difieren en el dominio de aplicación: mientras el estudio financiero se enfoca en transacciones de alta demanda y autenticación centralizada, GIATH aborda la administración de talento humano, donde el mayor desafío es gestionar solicitudes, validar saldos, analizar documentos, aplicar estados de aprobación y mantener la trazabilidad institucional. En contraste con los resultados reportados por Santacruz-Erraez y Poma-Japón, que no detallan tiempos de respuesta por escenario, GIATH demostró en las pruebas de carga (Tabla 10) que puede procesar hasta 95.3 req/s y mantener una tasa de error del 0%

incluso con 350 usuarios concurrentes, con tiempos promedio que van de 110 ms (20 usuarios) a 3200 ms (350 usuarios). Esta capacidad de sostener la estabilidad bajo estrés, combinada con una evaluación formal de calidad mediante la norma ISO/IEC 25010 (puntuaciones superiores a 4.4/5), constituye un nivel de evidencia más estructurado que el reportado por los estudios previos comparados.

por microservicio (Figura 11). La evaluación de usabilidad de GIATH mediante ISO/IEC 25010 constituye un punto de contraste metodológico, ya que los otros estudios no aplican un estándar internacional de calidad de software.

Un aspecto particular de GIATH que no ha sido analizado en estudios comparativos es que la privacidad y la seguridad de la información están claramente tratadas desde su diseño. Porque el

Tabla 13. Comparación de GIATH con estudios previos en criterios clave (tipo de aporte, contexto y validación).

Fuente: Los autores

Criterio	(Santacruz-Eraez & Poma-Japón, 2025)	(Sanchez-Galan & Castro Rodríguez, 2022)	(Rojano Guamaní et al., 2022)	Trabajo propuesto
Tipo de aporte	Prototipo de arquitectura de microservicios	Prototipo tecnológico	Aplicación móvil	Diseño arquitectónico y metodológico unificado
Contexto	Sistemas transaccionales financieros	Control administrativo de abastecimiento	Monitoreo agrícola	Gestión de vacaciones y permisos
Finalidad principal	Mejorar seguridad, escalabilidad e interoperabilidad transaccional	Automatizar control de suministro	Automatizar medición y seguimiento	Estructurar y controlar la gestión de solicitudes
Validación	Pruebas de seguridad, rendimiento, escalabilidad, integración y cumplimiento	Funcionamiento del prototipo	Métricas de precisión	Pruebas funcionales, rendimiento y encuesta ISO/IEC 25010
Alcance operativo	Financiero y transaccional	Específico	Específico	Administrativo e institucional
Aplicabilidad	Sistemas financieros y fintech	Sectorial	Sectorial	Transferible a organizaciones con procesos similares

Respecto al trabajo de Sanchez-Galan y Castro Rodríguez (2022), la semejanza está en la automatización de un trámite administrativo mediante una solución tecnológica. Su prototipo ciberfísico integra software, hardware, Arduino, códigos QR y un sistema experto para controlar el suministro de bebidas. GIATH, en cambio, no posee componente físico ni se ocupa de inventarios, sino de la gestión digital de requerimientos internos sometidos a reglas, roles y aprobaciones. A diferencia de la validación puramente funcional del prototipo de Sanchez-Galan, GIATH contrastó la calidad de su arquitectura mediante métricas estandarizadas de la norma ISO/IEC 25010, obteniendo evaluaciones de expertos superiores a 4.4/5 en todas las dimensiones, lo que otorga un respaldo metodológico más riguroso al artefacto.

En cuanto a Rojano Guamaní et al. (2022), coincide en el uso de una metodología ágil y pruebas de validación. Su aplicación, basada en Mobile-D y OpenCV, mide la longitud y diámetro del tallo de maíz, evaluando precisión con RMSE. GIATH no persigue precisión métrica sobre variables físicas, sino calidad funcional, rendimiento y aceptación por expertos. Frente al análisis de error físico de Rojano Guamaní, GIATH enfoca su éxito cuantitativo en el comportamiento del sistema bajo carga concurrente, demostrando un 0 % de tasa de errores bajo estrés y ofreciendo un análisis analítico de la latencia discriminada

sistema trata datos sensibles de servidores públicos (identidad, roles, documentos adjuntos y trazas completas del proceso de aprobación), el diseño se hizo con mecanismos alineados con la normativa ecuatoriana de protección de datos. La autenticación se implementó mediante el protocolo OIDC con tokens JWT validados criptográficamente en cada petición, mientras que el Control de Acceso Basado en Roles (RBAC), aplicado a través de Spring Security, restringe las operaciones a los perfiles autorizados. Las contraseñas se guardan con algoritmo Bcrypt y la información transaccional se persiste en SQL Server 2022 con soporte para cifrado de datos en reposo, los archivos adjuntos sensibles se gestionan mediante MongoDB GridFS, cuyo acceso está mediado exclusivamente por el backend. Adicionalmente, la bitácora inmutable registra cada acción (usuario, fecha, hora) sin posibilidad de modificación, garantizando la trazabilidad auditada de cada decisión. Estos mecanismos están alineados con los principios de responsabilidad proactiva y minimización de riesgos propios de los marcos normativos de protección de datos personales vigentes, aportan una dimensión de gobernanza de la información que resulta crítica en el contexto del sector público y que los sistemas comparados, orientados a dominios no institucionales, no requirieron abordar con el mismo nivel de exigencia.

De esta comparación, la contribución específica de GIATH





radica en la integración de un motor de reglas parametrizable, una bitácora inmutable y una arquitectura hexagonal basada en microservicios para administrar vacaciones, permisos y licencias. Esta combinación no se encontró en la literatura revisada, donde las soluciones se orientan a sistemas financieros, control de suministros o monitoreo agrícola. En GIATH, las reglas sobre saldos, cruce de turnos, requisitos documentales, estados del trámite y permisos por rol se gestionan fuera de la interfaz de usuario y de la infraestructura de persistencia, lo que dota al sistema de flexibilidad normativa y auditabilidad.

El sistema fue también validado a través de una estrategia de evaluación formal que comprendió pruebas funcionales de caja negra, simulaciones de rendimiento y un juicio experto basado en la norma ISO/IEC 25010. Esto prueba la viabilidad técnica de la solución y su capacidad para manejar reglas del negocio institucional antes de un potencial despliegue en producción. Por ello, GIATH no solo atiende el caso específico de la ISE, sino que se configura como un modelo tecnológico transferible a otras instituciones que necesiten controlar, validar y auditar procesos de gestión interna análogos.

Sin embargo, los resultados deben ser interpretados considerando que la evaluación se realizó en un ambiente controlado. Se anticipa que futuras investigaciones puedan basarse en evaluaciones longitudinales para capturar el impacto organizacional del sistema, su desempeño bajo condiciones reales de concurrencia, la disminución de errores administrativos y su integración con otros sistemas institucionales, como nómina, gestión documental u otros sistemas relacionados a la planificación del talento humano.

4. Conclusiones

Este estudio presenta GIATH, un sistema informático diseñado para digitalizar la gestión de vacaciones y permisos en la administración pública, permitiendo la transición de un flujo operativo manual hacia un entorno automatizado, auditable y alineado a la normativa de la Ley Orgánica del Servicio Público (LOSEP). Para su fundamentación se adoptó un enfoque metodológico y tecnológico integrado que unifica el modelado estructural de procesos (As-Is/To-Be) y los ciclos de desarrollo iterativos del marco ágil Extreme Programming (XP) con una arquitectura moderna basada en microservicios desacoplados y el patrón hexagonal. Este acoplamiento metodológico permitió traducir las reglas institucionales en componentes de software independientes (frontend en Angular/TypeScript, gateway con Traefik, backend en Java 21/Spring Boot, bases de datos SQL Server 2022 y MongoDB), ofreciendo condiciones favorables

para la escalabilidad horizontal y una separación efectiva de responsabilidades, si bien se requieren pruebas adicionales en entornos productivos para confirmar su escalabilidad plena.

Las pruebas realizadas en el entorno experimental (hasta 350 usuarios concurrentes en ambiente local controlado) evidencian que la solución alcanza los objetivos de funcionalidad, rendimiento y usabilidad definidos por el estándar ISO/IEC 25010, con resultados que confirman la aplicación consistente de las reglas de negocio y la estabilidad del sistema bajo las condiciones de carga evaluadas.

Los resultados cuantitativos indican que la arquitectura de GIATH aplica las reglas de negocio de manera consistente en todos los escenarios evaluados. El test funcional mostró un registro exitoso de solicitudes válidas (HTTP 201) y bloqueo de entradas erróneas (HTTP 400), asegurando integridad y trazabilidad en cada transacción. Las simulaciones de rendimiento revelaron que el sistema puede sostener un throughput de hasta 95.3 req/s, con tiempos promedio que oscilan entre 110 ms (con 20 usuarios) y 3200 ms (con 350 usuarios), sin registrar fallos (tasa de error 0 %). El incremento de latencia observado a partir de 70 usuarios (472 ms) y con cargas mayores es técnicamente razonable e identifica el umbral idóneo para el monitoreo y escalado en entornos de producción.

Estos resultados son relevantes ya que demuestran que un sistema basado en una arquitectura hexagonal y microservicios puede gestionar exitosamente el proceso de vacaciones y permisos en un entorno digital. GIATH separa la lógica de negocio (dominio) de las capas de persistencia y presentación, facilitando la evolución futura del software, y la parametrización de reglas permite adaptarse a cambios normativos sin modificar el código. Asimismo, el modelo desarrollado es transferible a otras instituciones con procesos administrativos análogos, como validación de permisos, aprobaciones jerárquicas y control de saldos.

En cuanto a las limitaciones, debe considerarse que las pruebas se realizaron en un entorno local y controlado; por tanto, los resultados reflejan el comportamiento del sistema durante la validación técnica, mas no su desempeño en una operación institucional continua. Para mitigar esto, la robustez del artefacto fue evaluada por un panel de 22 profesionales expertos del área de tecnologías de la información e ingeniería de software, quienes participaron en la revisión del sistema. Los resultados de esta evaluación bajo el estándar ISO/IEC 25010 arrojaron medias superiores a 4.4/5 en todas las dimensiones, concentrando las valoraciones entre “de acuerdo” y “totalmente de acuerdo”, con un 100 % de conformidad para el control de acceso por roles (P10), 90,9 % para la claridad de la interfaz (P8) y 81,8 % para la

facilidad de registro y prevención de errores (P6 y P7).

A partir de estos hallazgos, se plantea como trabajo futuro evaluar GIATH en escenarios reales de uso tras su despliegue institucional, medir su efecto en los tiempos de atención, en la reducción de errores operativos y en la satisfacción del personal, así como analizar su comportamiento bajo cargas mayores de tráfico. También se considera pertinente estudiar su interoperabilidad con sistemas institucionales core como nómina, planificación de talento humano o gestión documental, con el fin de ampliar su alcance dentro de un ecosistema digital integrado. No obstante, es preciso reconocer una limitación metodológica: la evaluación basada en la norma ISO/IEC 25010 se aplicó exclusivamente a profesionales de TI, no a los usuarios finales del área de Talento Humano. En consecuencia, las valoraciones de usabilidad y pertinencia funcional reflejan únicamente la perspectiva técnica, la cual podría diferir de la experiencia cotidiana de los servidores públicos. Para mitigar esta limitación durante el desarrollo, se integraron a dos ingenieros delegados por el departamento de Talento Humano como parte del principio de participación continua del cliente (XP), quienes verificaron el funcionamiento del sistema en cada iteración.

Agradecimientos

Agradecemos a la ISE por facilitar el desarrollo de la investigación y al personal participante por su colaboración durante el levantamiento de información. Asimismo, expresamos nuestro agradecimiento a nuestros tutores por el acompañamiento académico y técnico brindado, y a nuestras familias por su apoyo durante el desarrollo de este trabajo.

Contribución de los autores

Britney Paulette Tenecela Tocto: Redacción, Investigación, Metodología y Software. **Kevin Joel Zuñiga Iñiguez:** Redacción, Investigación, Metodología y Software. **Bertha Eugenia Mazon-Olivo:** Validación, Redacción, revisión y edición del artículo. **Damar Alejandra Taylor Vivanco:** Dirección de desarrollo práctico.

Conflictos de interés

Los autores declaran no tener ningún conflicto de interés.

Referencias bibliográficas

- Alarcón Quinapanta, C., & Hernández Junco, V. (2024). Modelos de Gestión de Talento Humano. Análisis Crítico: Human Talent Management Models. *Critical Analysis. REVISTA CIENTÍFICA ECOCIENCIA*, 11(1), 1-17. <https://doi.org/10.21855/ecociencia.111.847>
- Bautista-Villegas, E. (2022). Metodologías ágiles XP y Scrum,

empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel. *Revista Amazonía Digital*, 1(1), e168. <https://doi.org/10.55873/rad.v1i1.168>

- Bedolla Solano, J. J., Miranda Esteban, A., Bedolla Solano, R., & Castellanos Meza, C. (2026). Diseño y validación de instrumento para evaluar un modelo automatizado de sostenibilidad ambiental como estrategia de calidad científica y sostenible. *RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 16(32), e1039. <https://doi.org/10.23913/ride.v16i32.2833>
- Bosco, G., D'Amore, R., Sciarrone, A., & Barile, S. (2024). Managerial and Organizational Implications Arising from the Implementation of Blockchain Technology in Supply Chains: An AS-IS and To-Be Analysis. *Administrative Sciences*, 14(6), 120. <https://doi.org/10.3390/admsci14060120>
- Caciano Arroyo, M. E., Vasquez Cabrera, A. F., Santos Fernández, J. P., Boy Chavil, L. E., & Córdova Otero, J. L. (2024). Sistema Web para mejorar la gestión comercial y de talento humano utilizando la metodología Scrum. *Innovation and Software*, 5(1), 125-140. <https://doi.org/10.48168/innosoft.s15.a147>
- Cañizares Larco, H. F., Camacho Castillo, J. D., Castillo Camacho, T. A., & Canchignia Vasco, A. J. (2025). Desarrollo de una aplicación web para la gestión integral de la salud física y nutrición en adultos mayores en el centro gerontológico Guano. *Tesla Revista Científica*, 5(2), e492. <https://doi.org/10.55204/trc.v5i2.e492>
- Cárdenas-Gutiérrez, J. A., Barrientos-Monsalve, E. J., & Molina-Salazar, L. (2022). Arquitectura de Software para el desarrollo de herramienta Tecnológica de Costos, Presupuestos y Programación de obra. *I+D Revista de Investigaciones*, 17(1), 85-95. <https://doi.org/10.33304/revinv.v17n1-2022007>
- Carvajal-Pérez, A. L. (2021). Gestión Actual del talento humano: Contexto universitario. *Revista Nacional de Administración*, 12(2), e3914. <https://doi.org/10.22458/rna.v12i2.3914>
- Celis Crisostomo, M. A., Hernández López, F. M., Cárdenas Magaña, J. A., & Vega Negrete, E. (2025). Implementación de microservicios en proyectos de IoT con Arduino. *Ingenius*, (34), 9-19. <https://doi.org/10.17163/ings.n34.2025.01>
- Cortés Rojas, H. F. (2024). Uso de la norma ISO 25010 para establecer requerimientos de calidad en el diseño de un prototipo tecnológico educativo basado en realidad aumentada para la enseñanza de programación básica en estudiantes de educación media. *Arandu UTIC*, 11(2), 2603-2621. <https://doi.org/10.69639/arandu.v11i2.452>



- Durán Delgado, J. E., & Rojas Sánchez, H. A. (2023). Sistema de gestión del talento como medio de optimización del rendimiento de cada empleado. *Polo del Conocimiento*, 8(3), 485-495. <https://doi.org/10.23857/pc.v8i3.5314>
- Espinoza Beltrán, V., & Cachipundo Vásquez, M. V. (2024). El Sistema Integrado de Administración del Talento Humano, un enfoque en el Sector Público Ecuatoriano. *PODIUM*, (45), 33-52. <https://doi.org/10.31095/podium.2024.45.3>
- Ferreira, A., Salvadorinho, J., & Teixeira, L. (2026). Enhancing recruitment process for early-graduate employees in business consulting: A BPMN case study with AI-driven To-Be solution. *Procedia Computer Science*, 7th International Conference on Industry of the Future and Smart Manufacturing (former International Conference on Industry 4.0 and Smart Manufacturing), 277, 1983-1991. <https://doi.org/10.1016/j.procs.2026.02.236>
- Juma Alba, A. P., Pusda Aguilera, A. R., & Alvarez Varona, F. (2024). Gestión del talento humano: Aplicativo web para el análisis del desempeño laboral. *Revista Científica Dejando Huellas*, (2), 63-76. <https://doi.org/10.65100/recidh/12>
- Ley Orgánica de Servicio Público, LOSEP. (2010, octubre 6). *LEXIS*. <https://www.lexis.com.ec/biblioteca/losep>
- Manggaberani, A. A., & Darlis, A. M. (2024). The effectiveness of Google Forms in assessing and evaluating online learning outcomes: Meta-analysis study. *Jurnal Indonesia Sosial Teknologi*, 5(10), 5328-5337. <https://doi.org/10.59141/jist.v5i10.5305>
- Merchán-Narváez, N. J., Palma-Peralta, E. E., & Poma-Japón, D. X. (2024). Comparación de metodologías ágiles para el desarrollo de software. *MQRInvestigar*, 8(1), 5052-5074. <https://doi.org/10.56048/MQR20225.8.1.2024.5052-5074>
- Muñoz Solórzano, S. D., Cadena Vinuesa, P. A., Suquillo Guijarro, E., & Pastrano Jimbo, S. A. (2024). Continuous Improvement of Process Management Considering the «AS - IS» and «TO - BE» Criteria. *International Journal of Religion*, 5(11), 7410-7419. <https://doi.org/10.61707/5bddxr35>
- Palate-Ayme, N. N., Ramírez-Coque, J. F., Espinosa-Pico, P. E., & Alfonso-González, I. (2024). Implementación y regulación sobre el uso de la inteligencia artificial en el sistema judicial [Implementation and regulation of the use of artificial intelligence in the judicial system]. *Verdad y Derecho. Revista Arbitrada de Ciencias Jurídicas y Sociales*, 3(especial_Ambato), 176-182. <https://doi.org/10.62574/kqd4t363>
- Pambudi, A., & Apriandari, W. (2023). An Extreme Programming Approach for Instructor Performance Evaluation System Development. *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, 5(2), 126-135. <https://doi.org/10.20895/inista.v5i2.1050>
- Papalkar, R. R., & Alvi, A. S. (2024). Enhancing IoT security: A Creative Swagger Optimization algorithm for DDOS defence. *Network: Computation in Neural Systems*, 1-39. <https://doi.org/10.1080/0954898X.2024.2443605>
- Pérez-Pacheco, Y., & Simón Castellano, P. (2025). La inteligencia artificial en los sistemas judiciales de América Latina: Una revisión sobre desafíos y oportunidades. *A&C - Revista de Derecho Administrativo & Constitucional*, 25(100), 95-129. <https://doi.org/10.21056/aec.v25i100.2027>
- Pilicita Garrido, A., Borja López, Y., & Gutiérrez Constante, G. (2020). Rendimiento de MariaDB y PostgreSQL. *Revista Científica y Tecnológica UPSE*, 7(2), 09-16. <https://doi.org/10.26423/rctu.v7i2.538>
- Pinargote Párraga, J. E., & Pico Macías, M. E. (2023). Modelo de Gestión de Talento Humano como factor del desarrollo en centros de educación superior: Revisión bibliográfica. *RECIMUNDO*, 7(2), 117-131. [https://doi.org/10.26820/recimundo/7.\(2\).jun.2023.117-131](https://doi.org/10.26820/recimundo/7.(2).jun.2023.117-131)
- Proaño Ponce, W. P., & Capurro Choez, R. E. (2023). Gestión administrativa y el talento humano: Empresa pública de Agua Potable y Alcantarillado, Cantón Jipijapa. *Ciencia y Desarrollo*, 26(3), 65-73. <https://doi.org/10.21503/cyd.v26i3.2492>
- Quiñónez Rentería, M. A., & Segarra Jaime, H. P. (2025). La digitalización documental y su incidencia en la eficiencia de procesos administrativos de la cooperativa de ahorro y crédito grupo número tres limitada.: Document digitization and its impact on the efficiency of administrative processes at the savings and credit cooperative group number three limited. *Revista Científica Multidisciplinaria G-nerando*, 6(1), 6414-6437. <https://doi.org/10.60100/rcmg.v6i1.710>
- Rodríguez, B., & Cedeño, D. (2023). Diseño e implementación de una arquitectura de microservicios orientada a trabajar con transacciones distribuidas. *I+D Tecnológico*, 19(1), 113-118. <https://doi.org/10.33412/idt.v19.1.3783>

Rojano Guamaní, V. J., Jaramillo Tenezaca, G. L., Cantuña Flores, K. S. C., Sandoval Ruilova, G. A., & Bengochea Guevara, J. M. (2022). Desarrollo de una aplicación móvil para el monitoreo de la fenometría vegetativa del maíz amarillo, en la sierra central ecuatoriana. *Informática y Sistemas*, 6(1), 71-77. <https://doi.org/10.33936/isrtic.v6i1.4447>

Sanchez-Galan, J., & Castro Rodríguez, A. A. (2022). Desarrollo de un prototipo de sistema ciber-físico para el control administrativo del suministro de bebidas y refrescos: Development of a Expert System and a Prototype of a Cyber-physical System for the Administrative Control of the Supply of Beverages and Soft Drinks. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 6(2), 23-31. <https://doi.org/10.33936/isrtic.v6i2.4777>

Santacruz-Erreaez, L. F., & Poma-Japón, D. X. (2025). Prototipo de Arquitectura de Microservicios para Sistemas Transaccionales Financieros con Keycloak. *MQRInvestigar*, 9(1), e189-e189. <https://doi.org/10.56048/MQR20225.9.1.2025.e189>

Shrivastava, A., Jaggi, I., Katoch, N., Gupta, D., & Gupta, S. (2021). A Systematic Review on Extreme Programming. *Journal of Physics: Conference Series*, 1969(1), 012046. <https://doi.org/10.1088/1742-6596/1969/1/012046>

Wang, L. (2023). Innovation of Administrative Management System of Universities Based on Cloud Computing. *Applied Mathematics and Nonlinear Sciences*, 9(1), 1-16. <https://doi.org/10.2478/amns.2023.2.01495>

