



**Implementación y estandarización en microservicios para altos consumos de lectura en
sistemas transaccionales**

Juan José Urrego Rios

Práctica empresarial para optar al título de Ingeniero en Sistemas otorgador por la UdeA

Orientadores

Sandra Patricia Zabala Orrego, Ingeniera Informática Especialista en Gerencia

Kelly Johana Pulgarín Cañaveral, Ingeniera en Sistemas

Universidad de Antioquia

Facultad de Ingeniería

Pregrado en Ingeniería en Sistemas presencial

Medellín

2024

Cita	Urrego Rios [1]
Referencia	[1] J. J. Urrego Rios, "Implementación y estandarización en microservicios para altos consumos de lectura en sistemas transaccionales", Pregrado en Ingeniería en Sistemas presencial UdeA, Universidad de Antioquia, Medellín, 2024.

Estilo IEEE (2020)



Centro de Documentación de Ingeniería -CENDOI-

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga Molina.

Jefe departamento: Danny Alejandro Múnera Ramírez

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

RESUMEN	7
ABSTRACT	8
I. INTRODUCCIÓN	9
II. PLANTEAMIENTO DEL PROBLEMA	10
III. JUSTIFICACIÓN	11
IV. OBJETIVOS	12
V. MARCO TEÓRICO	13
VI. METODOLOGÍA	16
VII. RESULTADOS	17
VIII. CONCLUSIONES	19
REFERENCIAS	20

LISTA DE TABLAS

TABLA I RESULTADOS DEL TEST

17

LISTA DE FIGURAS

Fig. 1. Arquitectura de sistema Horus	13
Fig. 2. Implementación arquitectura Hexagonal	14

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

IEEE	Institute of Electrical and Electronics Engineers
S.A	Sociedad Anónima
S.A.S	Sociedad por Acciones Simplificada
UdeA	Universidad de Antioquia
AWS	Amazon Web Services
EKS	Amazon Elastic Kubernetes Service
RDS	Amazon Relational Database Service

RESUMEN

Este proyecto se enfoca en mejorar la eficiencia del sistema transaccional Horus de Bancolombia Fiduciaria, una entidad destacada en servicios fiduciarios en Colombia y América Latina. Horus, implementado con PostgreSQL en una infraestructura en la nube de AWS y utilizando tecnologías como Angular y Spring Boot, ha enfrentado problemas significativos de rendimiento, especialmente en la lectura de datos desde la base de datos y en la gestión de grandes volúmenes de información en tiempo real. Estos problemas afectan negativamente la experiencia del usuario y la productividad del sistema, que está destinado a sustituir a la aplicación legada GESTOR.

El objetivo principal del proyecto es optimizar la eficiencia y escalabilidad del sistema Horus mediante la implementación de microservicios especializados para gestionar altos volúmenes de lectura de datos. Se propone realizar un análisis detallado de los patrones de acceso a datos, diseñar nuevas herramientas y metodologías, y desarrollar microservicios con mejores prácticas arquitectónicas. La documentación exhaustiva de estos procesos garantizará la mejora continua y la adaptabilidad del sistema. Este enfoque busca resolver los cuellos de botella actuales y mejorar la experiencia del usuario al permitir un procesamiento más ágil y confiable de las transacciones.

Palabras clave — Transaccional, GESTOR, infraestructura, Bancolombia.

ABSTRACT

This project aims to enhance the performance of Bancolombia Fiduciaria's transactional system, Horus, a key component in their fiduciary services. Built with PostgreSQL on AWS cloud infrastructure, and employing Angular and Spring Boot, Horus has faced significant performance issues, notably with data retrieval and handling large datasets in real-time. These issues negatively impact user experience and system productivity, as Horus is designed to replace the legacy application GESTOR.

The project's primary goal is to improve the efficiency and scalability of Horus by implementing and standardizing microservices tailored for high data read loads. It involves a detailed analysis of data access patterns, the design of new tools and methodologies, and the development of microservices adhering to best architectural practices. Comprehensive documentation will support future projects and enhancements. This approach seeks to resolve existing bottlenecks, ensuring faster and more reliable processing of transactions, thus enhancing user experience.

***Keywords* — Transactional, GESTOR, infrastructure, Bancolombia.**

I. INTRODUCCIÓN

El presente informe tiene como objetivo presentar una solución eficiente y confiable para el sistema transaccional Horus, implementado por Bancolombia Fiduciaria, una entidad líder en servicios fiduciarios en Colombia. Fundada en 1945, Bancolombia Fiduciaria se destaca por su sólida trayectoria y experiencia en el mercado. Ofrece una amplia gama de servicios fiduciarios tanto para personas naturales como jurídicas, y opera en varios países de América Latina, incluyendo Perú, Panamá, Puerto Rico y El Salvador. Su red de sucursales y base de clientes en Colombia, que supera los millones, la posiciona como una de las fiduciarias más grandes y relevantes en el país. Su compromiso con la innovación y la excelencia en el servicio la ha consolidado como una institución de confianza en el panorama financiero colombiano.

Horus, el nuevo sistema transaccional de Bancolombia Fiduciaria, es clave para la innovación y el desarrollo en el ámbito fiduciario. Este sistema, diseñado para gestionar grandes volúmenes de transacciones, requiere una alta eficiencia en cada una de sus funcionalidades. No obstante, a pesar de su desarrollo durante casi tres años, se han identificado desventajas significativas en el procesamiento de consultas. Estas deficiencias afectan la experiencia del usuario y, dado que Horus pretende reemplazar el sistema legad GESTOR, es esencial que sus usuarios se adapten eficazmente al nuevo sistema y experimenten una mejora en la productividad.

Por consiguiente, el presente informe abordará los antecedentes del problema, así como la metodología utilizada para su resolución. El problema principal radica en la ineficiencia en el procesamiento de consultas, que ralentiza el sistema y afecta la experiencia del usuario. En respuesta, se establecerán objetivos para mejorar la eficiencia y escalabilidad de Horus, mediante la implementación de nuevas tecnologías y metodologías. El objetivo es garantizar que Horus no solo cumpla con las expectativas actuales, sino que también se posicione como un sistema transaccional de vanguardia en el sector fiduciario.

II. PLANTEAMIENTO DEL PROBLEMA

El sistema transaccional Horus, desarrollado con PostgreSQL y desplegado en la infraestructura en la nube de AWS, utilizando Angular y Spring Boot, enfrenta problemas significativos de rendimiento. Desde su implementación, se han reportado numerosos incidentes y fallas, con un foco particular en la lentitud en la lectura de datos desde la base de datos. Este problema se manifiesta especialmente en las bandejas, que son tablas que contienen grandes volúmenes de información y que, debido a su paginación y a los complejos criterios de búsqueda, generan cuellos de botella.

El desafío principal radica en la dificultad de manejar grandes cantidades de datos en tiempo real sin comprometer el rendimiento del sistema. Esta ineficiencia en la lectura de datos afecta de manera crítica la velocidad y la fluidez de la aplicación, deteriorando la experiencia del usuario y limitando la productividad esperada del sistema.

Adicionalmente, la integración de Spring Security para la autenticación añade una capa de complejidad, en particular cuando se implementan microservicios reactivos. Estos microservicios requieren configuraciones y manejos especializados, distintos de los servicios tradicionales, lo que complica aún más la resolución de los problemas de rendimiento.

En síntesis, el problema central es la ineficiencia en el procesamiento de datos en la base de datos, que repercute negativamente en el rendimiento global de la aplicación. Es esencial identificar y aplicar soluciones que optimicen la gestión de grandes volúmenes de datos, garantizando así la agilidad y la fiabilidad del sistema transaccional Horus.

III. JUSTIFICACIÓN

Este estudio se enfoca en optimizar el rendimiento del sistema transaccional Horus de Bancolombia Fiduciaria, crucial para la eficiencia operativa y la satisfacción del usuario en un entorno financiero moderno. Dado que Horus reemplaza al sistema legado GESTOR, es esencial abordar los problemas de lentitud en la lectura de datos y los cuellos de botella en el procesamiento, que afectan tanto la productividad como la experiencia del usuario.

La elección de este tema responde a la necesidad de mejorar el manejo de grandes volúmenes de datos y la integración de tecnologías como Spring Security en un entorno de microservicios. Las soluciones propuestas no solo resolverán problemas específicos de Horus, sino que también ofrecerán metodologías y enfoques aplicables a otros sistemas transaccionales similares.

Este trabajo contribuye a la ciencia al proporcionar estrategias prácticas para la optimización de sistemas transaccionales complejos y ofrece una base para futuras investigaciones en la ingeniería de sistemas y la informática aplicada, extendiendo su utilidad más allá del contexto de Bancolombia Fiduciaria.

IV. OBJETIVOS

A. Objetivo general

Mejorar la eficiencia y escalabilidad de los sistemas transaccionales a través de la implementación y estandarización de microservicios diseñados para manejar altos consumos de lectura.

B. Objetivos específicos

- Realizar un análisis detallado de los patrones de acceso a datos en los microservicios para identificar las operaciones de lectura más frecuentes y críticas para el negocio.
- Diseñar nuevas herramientas y metodologías que puedan ser usadas en cualquier microservicio que presente Horus.
- Desarrollar microservicios siguiendo las mejores prácticas arquitectónicas y de diseño, asegurando su escalabilidad y capacidad para manejar cargas de lectura intensivas.
- Documentar exhaustivamente los procesos de análisis, diseño, desarrollo e implementación, proporcionando recursos y guías para futuros proyectos y mejoras en sistemas transaccionales.

V. MARCO TEÓRICO

Arquitectura de Sistemas Transaccionales:

Los sistemas transaccionales están diseñados para manejar operaciones críticas en tiempo real, garantizando la integridad y disponibilidad de los datos. Estos sistemas deben gestionar grandes volúmenes de transacciones simultáneas y asegurar que los datos sean procesados de manera eficiente y precisa. La arquitectura típica de un sistema transaccional incluye componentes como bases de datos, servidores de aplicaciones, y mecanismos de recuperación ante fallos, todos optimizados para operaciones de alta carga.

El sistema Horus está desplegado en Amazon Web Services (AWS), lo que proporciona una infraestructura escalable y flexible. Los usuarios acceden al sistema a través de un cliente, que redirige las solicitudes a un clúster de Kubernetes (EKS). Este clúster gestiona múltiples microservicios que, a su vez, interactúan con una base de datos gestionada por RDS cluster aurora como lo muestra la Figura 1. Esta configuración permite un alto grado de escalabilidad y disponibilidad, clave para soportar las operaciones transaccionales de Bancolombia Fiduciaria.

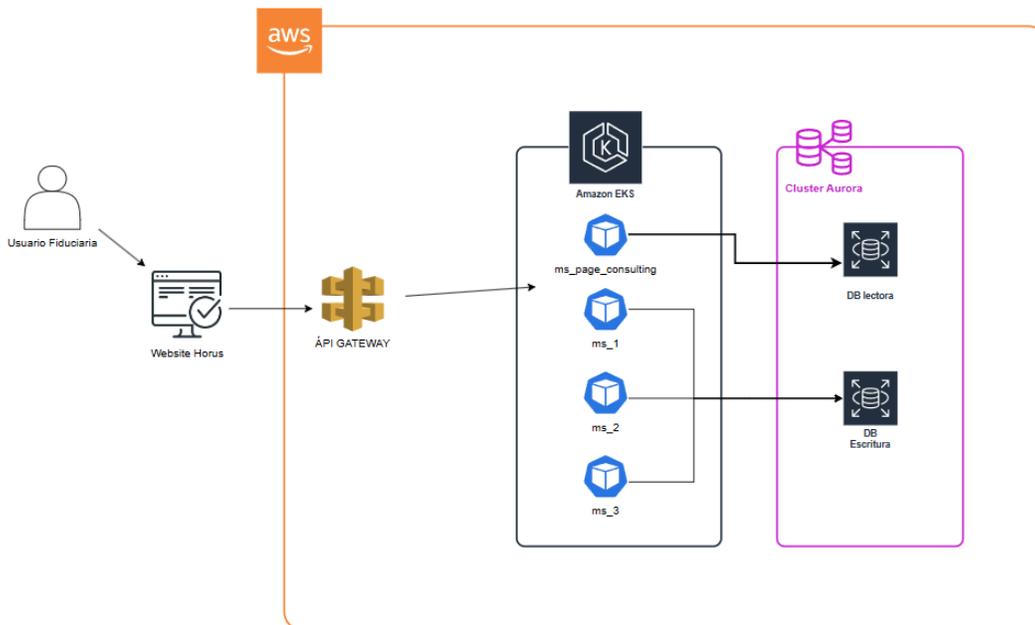


Fig. 1. Arquitectura de sistema Horus

Nota. Autoría propia

Microservicios y Arquitectura Hexagonal:

La arquitectura de microservicios descompone una aplicación en servicios independientes que pueden ser desarrollados, desplegados y escalados de manera autónoma. Este enfoque mejora la flexibilidad y la capacidad de mantenimiento del sistema al permitir cambios y actualizaciones en servicios individuales sin afectar al sistema completo.

La arquitectura de un microservicio, en este caso, el microservicio "PageConsulting" que la implementación realizada para la solución, es basada en la arquitectura hexagonal también conocida como arquitectura de puertos y adaptadores, se enfoca en separar la lógica de negocio del resto del sistema, tal como muestra en la Figura 2. En el modelo hexagonal, el núcleo del microservicio está aislado de las interfaces externas, lo que facilita la adaptación a diferentes tecnologías y herramientas. Este enfoque es crucial para manejar la complejidad de las consultas y mejorar el rendimiento del sistema, permitiendo una gestión más eficiente de las operaciones de lectura y la integración con otros componentes del sistema.

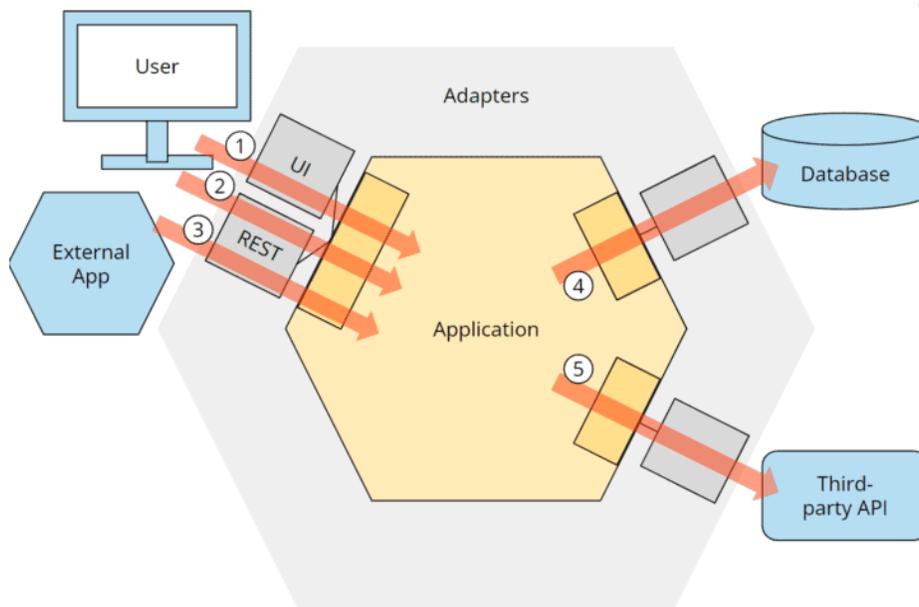


Fig. 2. Implementación arquitectura Hexagonal

Nota. Imagen recuperada de [5]

Optimización de Consultas en Bases de Datos

La optimización de consultas es fundamental en sistemas que manejan grandes volúmenes de datos. Implementar índices en las bases de datos y utilizar técnicas como la paginación eficiente mejora el tiempo de respuesta de las consultas. En el contexto de Horus, el uso de una base de datos lectora ayuda a distribuir la carga de trabajo, aliviando la presión sobre la base de datos principal y mejorando el rendimiento general del sistema.

Spring WebFlux y Arquitectura Reactiva

Spring WebFlux es un framework para construir aplicaciones reactivas, que maneja de manera eficiente operaciones I/O intensivas mediante un modelo basado en eventos. Esta tecnología es particularmente útil para aplicaciones que requieren alta escalabilidad y rendimiento. En el sistema Horus, Spring WebFlux permite gestionar operaciones de manera más eficiente, mejorando la capacidad de respuesta del sistema y su rendimiento global.

Replicación y Seguridad de Bases de Datos

La replicación de bases de datos implica crear una copia secundaria de la base de datos principal para operaciones de lectura, lo que ayuda a distribuir la carga y mejorar la velocidad de las consultas. Además, la integración de Spring Security con microservicios reactivos garantiza que el sistema mantenga un alto nivel de seguridad y protección de datos, adaptándose a los requisitos específicos de la arquitectura de microservicios.

VI. METODOLOGÍA

Continuamos utilizando la metodología ágil, específicamente la modalidad de Scrum, para guiar nuestro proceso de desarrollo. Dado el éxito y la eficacia de esta metodología en fases anteriores, mantendremos la estructura de sprints de 15 días calendario (equivalentes a 10 días hábiles de trabajo efectivo). Durante cada sprint, seguiremos realizando reuniones diarias (dailys) para monitorear el progreso del equipo y abordar cualquier obstáculo de manera oportuna. En esta fase, las actividades se enfocarán en la implementación y optimización de las nuevas pantallas y funcionalidades críticas dentro del microservicio "PageConsulting". Continuaremos iterando y mejorando el producto en cada ciclo de desarrollo, asegurando que cada incremento aporte valor significativo y se alinee con los objetivos del proyecto. La metodología Scrum nos permitirá mantener una adaptación constante y una mejora continua, maximizando la eficiencia y efectividad del equipo.

VII RESULTADOS

La implementación del microservicio "PageConsulting" ha sido completamente adoptada y los resultados han sido excepcionales. Se logró una integración efectiva en todas las áreas críticas de la aplicación, lo que ha permitido un rendimiento superior y una notable mejora en la eficiencia del sistema. Gracias a esta implementación, se redujo los tiempos de espera de 20-50 segundos a menos de 0.5 segundos para obtener la misma información, pruebas: estos resultados se muestran en la **TABLA I**. Esta mejora ha sido consistente a lo largo de todas las pruebas y validaciones, consolidando la estabilidad y efectividad del microservicio. El estado actual del proyecto es altamente prometedor. La solución implementada no solo ha mostrado resultados positivos, sino que también ha mejorado significativamente la experiencia del usuario final y la eficiencia del sistema transaccional en su conjunto. Con estas mejoras, la aplicación está preparada para futuras expansiones y optimizaciones continuas.

TABLA I
RESULTADOS DEL TEST

Condición	Implementación	Tiempos
Suspendido	Antigua	50s
En Línea	Antigua	24s
Suspendido	Nueva	3s
En Línea	Nueva	0.5s

Nota: variación de resultados con implementación propuesta e implementación antigua.

VIII. CONCLUSIONES

- La completa adopción e implementación del microservicio "PageConsulting" ha demostrado ser una solución altamente efectiva para mejorar el rendimiento y la eficiencia del sistema transaccional.
- La reducción drástica en los tiempos de respuesta, de 20-50 segundos a menos de 0.5 segundos, ha superado nuestras expectativas y ha mejorado significativamente la experiencia del usuario final.
- A pesar de los desafíos operativos y técnicos, como la migración de lógica antigua y la integración con Spring Security, hemos logrado estabilizar el microservicio y asegurar su funcionamiento óptimo.
- La adopción de buenas prácticas, estándares de desarrollo, y una documentación exhaustiva han sido claves para este éxito.
- Con la implementación continua de nuevas pantallas y la migración de funcionalidades críticas al microservicio, estamos bien posicionados para mantener y mejorar el rendimiento del sistema.
- La capacitación continua del equipo y el monitoreo constante aseguran que estaremos preparados para enfrentar cualquier desafío futuro y aprovechar al máximo las capacidades de nuestra arquitectura mejorada.

REFERENCIAS

- [1] Astros, Ivan Jose Turnero. “Sistema de información transaccional: Teoría y aplicación.” *Monografias.com*, 13 July 2017, <https://www.monografias.com/docs114/sistema-informacion-transaccional-teoria-y-aplicacion/sistema-informacion-transaccional-teoria-y-aplicacion>.
- [2] “Grupo.” *Bancolombia*, <https://www.grupobancolombia.com/>. Accessed 29 Aug. 2024.
- [3] “Grupo.” https://www.sic.gov.co/sites/default/files/files/integracion_empresarial/pdf/2019/julio/BANCOLOMBIA%20-%20DAVIVIENDA%20-%20BANCO%20DE%20BOGOT%C3%81.pdf. Accessed 29 Aug. 2024.
- [4] *Org.Co*, https://www.asofiduciarias.org.co/wp-content/uploads/2017/11/ABCDELAFIDUCIAFiscaliaFINAL_comprimido.pdf. Accessed 29 Aug. 2024.
- [5] Ferreira, Diogo. “How to Structure Your Code? Hexagonal Architecture!” *DataDrivenInvestor*, 20 Feb. 2024, <https://medium.datadriveninvestor.com/how-to-structure-your-code-hexagonal-architecture-261bec59ef64>.