

Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Ingeniería en Electricidad y Computación

**EVALUACIÓN INTEGRAL DE LA PLATAFORMA OBSERVACIONES
LITORALES DEL ECUADOR DESDE UN ENFOQUE DE
ARQUITECTURA EMPRESARIAL**

Proyecto de Titulación

Previo la obtención del Título de:

Magíster en Sistemas de Información Gerencial

Presentado por:

Luis Antonio Palomino Romero

David William Jácome Rodríguez

Guayaquil - Ecuador

Año: 2025

Dedicatoria

Dedico este trabajo a mí madre por su amor infinito, por creer en mí, por su perseverancia, por haber sido la mujer que me enseñó a jamás rendirme y quien me acompañó en cada paso mientras pudo. También a mis hermanos por ser mi compañía y apoyo en los días más difíciles.

(William Jácome Rodríguez)

Dedico este trabajo a mis queridas hijas y a mi esposa, quienes con su amor, apoyo incondicional y alegría me han dado las fuerzas para continuar; a mi madre, por enseñarme el verdadero valor del esfuerzo, la perseverancia, el amor y la disciplina; y a mi padre, por su sabiduría y constante aliento, siendo siempre un ejemplo de esfuerzo y dedicación. Este logro es gracias a su amor y dedicación.

(Luis Palomino Romero)

Agradecimientos

Expresamos nuestro más sincero agradecimiento a nuestro tutor Mgtr. José Luis Asencio por su orientación, seguimiento y apoyo otorgado, además a nuestros profesores y compañeros por el conocimiento y enseñanzas que contribuyeron a nuestra formación académica.

Finalmente agradecemos a ESPOL por ser una institución de excelencia y forjarnos a ser mejores líderes en el ámbito profesional.

Declaración Expresa

Nosotros David William Jácome Rodríguez y Luis Antonio Palomino Romero acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 15 de agosto del 2025.

Luis Antonio Palomino
Romero

David William Jácome
Rodríguez

Evaluadores

Mgtr. José Luis Asencio Mera

Tutor de proyecto

PhD. Juan Carlos García

Revisor

RESUMEN

El proyecto evalúa la plataforma "Observaciones Litorales del Ecuador" con el fin de consolidar su viabilidad técnica y su alineación con los objetivos institucionales. Se propone como objetivo perfeccionar su estructura tecnológica, mejorar su operación y garantizar su progreso a largo plazo, fundamentándose en la necesidad de vencer las restricciones de su estructura monolítica y su elevada deuda técnica. Se utilizó el marco TOGAF para el desarrollo, incorporando sus fases para el diagnóstico, diseño y planificación de la transición hacia un modelo fundamentado en microservicios y microfrontends. Se emplearon instrumentos de modelado arquitectónico, métodos de integración y entrega continua, y estrategias de gestión del cambio. Los hallazgos indican la detección de fallos críticos en la arquitectura vigente, la elaboración de una propuesta de mejora completa y la presentación de una propuesta de mejora integral. El rediseño propuesto permite una mayor flexibilidad, escalabilidad e interoperabilidad con otros sistemas, garantizando que la plataforma continúe siendo un soporte clave para la gestión y monitoreo costero.

Palabras clave: Arquitectura empresarial, microservicios, sostenibilidad tecnológica, TOGAF, monitoreo costero.

ABSTRACT

The project evaluates the "Coastal Observations of Ecuador" platform to consolidate its technical viability and alignment with institutional objectives. Its objective is to perfect its technological structure, improve its operation, and ensure its long-term progress, based on the need to overcome the constraints of its monolithic structure and high technical debt. The TOGAF framework was used for development, incorporating its phases for diagnosis, design, and planning of the transition to a model based on microservices and microfrontends. Architectural modeling tools, continuous integration and delivery methods, and change management strategies were employed. The findings indicate the detection of critical flaws in the current architecture, the development of a comprehensive improvement proposal, and the presentation of a comprehensive improvement proposal. The proposed redesign allows for greater flexibility, scalability, and interoperability with other systems, ensuring that the platform continues to be a key support for coastal management and monitoring.

Keywords: Enterprise architecture, microservices, technological sustainability, TOGAF, coastal monitoring.

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICES DE FIGURAS	VII
ÍNDICES DE TABLAS	VIII
CAPÍTULO 1	1
1. INTRODUCCIÓN.....	1
1.1 Descripción del Problema	1
1.2 Justificación del Problema	1
1.3 Objetivos	2
1.3.1 Objetivo general	2
1.3.2 Objetivos específicos	2
1.4 Marco Teórico.....	2
1.4.1 Arquitectura empresarial	2
1.4.2 Sistemas de información.....	4
1.4.3 Metodologías de evaluación	4
1.4.4 Evaluación centrada en el usuario	5
CAPÍTULO 2	6
2. METODOLOGÍA.....	6
2.1 Levantamiento de Información	6
2.1.1 Fase A – Visión de arquitectura	8
2.1.2 Fase B – Arquitectura de negocio	9
2.1.3 Fase C – Arquitectura de sistemas de información.....	10
2.1.4 Fase D – Arquitectura de tecnología.....	11
2.2 Diagnóstico de Situación Actual.....	12
2.2.1 Evaluación de la plataforma web	12
2.2.2 Evaluación de la aplicación móvil	12
2.3 Análisis de Brechas.....	14
2.3.1 Fase E – Oportunidades y soluciones.....	15
2.4 Consideraciones Éticas y Legales	16

2.5	Consideraciones Finales.....	16
CAPÍTULO 3		17
3.	RESULTADOS Y ANÁLISIS	17
3.1	Fase F – Planificación de la Migración.....	17
3.2	Fase G – Gobernanza de la Implementación	17
3.3	Fase H – Gestión del Cambio	18
3.4	Estimación del Impacto Esperado del Plan de Mejora.....	19
CAPÍTULO 4		21
4.	CONCLUSIONES Y RECOMENDACIONES	21
4.1	Conclusiones	21
4.2	Recomendaciones	21
Referencias		22

ABREVIATURAS

CENAREC	Centro Nacional de Recursos Costeros
OLE	Observaciones Litorales del Ecuador
ADM	Método de Desarrollo de Arquitectura
ESPOL	Escuela Superior Politécnica del Litoral
API	Interfaz de Programación de Aplicaciones
REST	Transferencia de Estado Representacional
CI	Integración Continua
CD	Despliegue Continuo
HTTP	HyperText Transfer Protocol
CSP	Content Security Policy
HSTS	HTTP Strict Transport Security
XSS	Cross-Site Scripting
IoT	Internet de las Cosas

SIMBOLOGÍA

Seg Segundos

ÍNDICES DE FIGURAS

Fig. 1. Comparación entre marcos de trabajo de arquitectura empresarial	3
Fig. 2. Método de desarrollo de arquitectura (ADM) [1]	3
Fig. 3. Proceso de análisis.....	6
Fig. 4. Vista de interesados	8
Fig. 5. Matriz de Poder- Interés	9
Fig. 6. Roles y procesos	10
Fig. 7. Situación actual: Arquitectura de Sistemas de Información	11
Fig. 8. Situación actual: Tecnología	11
Fig. 9. Análisis de calidad de código Backend	12
Fig. 10. Análisis de calidad de código Frontend	13
Fig. 11. OLE Arquitectura TO BE	15
Fig. 12. Fases de implementación	17
Fig. 13. Gobernanza de la implementación	18
Fig. 14. Plan de gestión del cambio	19

ÍNDICES DE TABLAS

Tabla I. Fases del ADM y su aplicación en la plataforma OLE	7
Tabla II. Evaluación de aspectos de la aplicación.....	13
Tabla III. Comparación de alternativas arquitectónicas para la plataforma OLE.....	14
Tabla IV. Análisis de mejora esperada	19

CAPÍTULO 1

1. INTRODUCCIÓN

La tecnología desempeña un rol fundamental en la gestión de información ambiental, optimizando la toma de decisiones en la conservación costera. En la oceanografía, herramientas como sensores IoT y plataformas de datos en tiempo real permiten monitorear variables costeras, fortaleciendo la planificación ambiental. En zonas rurales como Manglaralto, aplicaciones móviles con capacidad offline facilitan la recolección de datos oceanográficos, superando limitaciones de conectividad. La estabilidad de estos sistemas es crucial para garantizar la integridad de los datos y evitar pérdidas críticas, apoyando decisiones informadas. En este contexto, la plataforma Observaciones Litorales del Ecuador (OLE) monitorea cambios costeros y presenta datos accesibles para los tomadores de decisiones en Manglaralto. Sin embargo, enfrenta limitaciones técnicas, como una arquitectura monolítica, el uso de tecnologías obsoletas, y procesos de desarrollo no estandarizados, que restringen su escalabilidad y alineación con objetivos institucionales, justificando una evaluación integral basada en arquitectura empresarial.

1.1 Descripción del Problema

La plataforma “Observaciones Litorales del Ecuador” es muy importante para manejar la información ambiental en la parroquia Manglaralto, pero ahora mismo tiene varios problemas que complican que funcione bien y que pueda seguir creciendo. Entre los problemas identificados se incluyen tecnologías obsoletas, procesos de desarrollo cambiantes debido a múltiples desarrolladores en etapas distintas, lo que genera código inconsistente, y limitaciones en escalabilidad por la estructura monolítica.

Por ello, la plataforma presenta dificultades para crecer, mantenerse actualizada tecnológicamente y responder plenamente a las necesidades de la organización. Aunque cuenta con un alto potencial para contribuir a la gestión ambiental y a la toma de decisiones estratégicas, estas limitaciones dificultan el aprovechamiento pleno de las capacidades que un sistema de esta categoría puede ofrecer.

1.2 Justificación del Problema

La plataforma OLE constituye un elemento clave para la gestión ambiental en Manglaralto; sin embargo, sus limitaciones actuales impiden maximizar su impacto. No se trata únicamente de un aspecto técnico, ya que también representa una herramienta fundamental para comprender la dinámica y el estado de la zona costera. Esta herramienta resulta esencial para mejorar la planificación, fortalecer la protección ambiental y responder de manera oportuna ante la ocurrencia de eventos naturales. Al mejorarla, se beneficia la institución y las comunidades que necesitan esos datos para tomar decisiones. La ausencia de una evaluación integral resalta la necesidad de analizar la plataforma desde un enfoque de arquitectura empresarial para identificar fallas técnicas y alinearla con objetivos institucionales.

En este proyecto se requiere aplicar el análisis de la arquitectura empresarial, donde se revisará de manera integral el sistema, con el objetivo de mejorar su funcionalidad y componentes, así como la estructura, procesos o el uso de los datos. Esto servirá para que el sistema trabaje eficientemente y aporte a la transformación digital. El proyecto tiene muchas razones para ser viable. Por un lado, desde lo técnico, se tiene acceso completo a la plataforma y a su documentación, así que se puede usar métodos reconocidos sin gastar en cosas nuevas. Por otra parte, el equipo encargado del desarrollo del proyecto posee amplia experiencia en análisis institucional y en la gestión de soluciones tecnológicas. Ha planificado cada etapa de manera realista y cuenta con el respaldo de las personas clave involucradas. Asimismo, la iniciativa se encuentra alineada con los objetivos institucionales orientados a la mejora continua y a la digitalización de procesos, lo que garantiza el apoyo necesario. Al no requerir una inversión significativa ni la incorporación de nueva infraestructura, el proyecto puede avanzar sin contratiempos económicos ni organizativos.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un plan de mejora basado en principios de arquitectura empresarial considerando la estructura tecnológica, la gestión de datos, la experiencia del usuario y la alineación con los objetivos estratégicos de la institución para la plataforma OLE.

1.3.2 Objetivos específicos

1. Levantar información sobre la situación actual de la plataforma en aspectos tecnológicos, funcionales y organizacionales.
2. Diseñar la arquitectura empresarial aplicando el Método de Desarrollo de Arquitectura (ADM) del marco de trabajo TOGAF.
3. Elaborar un plan de mejora integral que consolide las recomendaciones derivadas del diagnóstico arquitectónico y técnico de la plataforma, incorporando acciones estructuradas orientadas al mejoramiento de su rendimiento, mantenibilidad, experiencia de usuario y alineación estratégica.

1.4 Marco Teórico

1.4.1 Arquitectura empresarial

La arquitectura empresarial (AE) es un enfoque estratégico que alinea los objetivos de una organización con sus capacidades tecnológicas. Su finalidad es comprender de forma integral el funcionamiento de la entidad para planificar, ejecutar y controlar los cambios, algo especialmente relevante en un entorno caracterizado por la rápida evolución tecnológica y la constante transformación organizativa [1], [2].

La arquitectura empresarial se puede ver como una hoja de ruta que muestra dónde está la organización ahora y hacia dónde quiere llegar, incluyendo sus procesos, sistemas digitales y tecnologías [1]. Contar con una visión clara hace que sea mucho más fácil tomar buenas decisiones, impulsar nuevas ideas y avanzar en la transformación digital. Según lo que mencionan en [3], la arquitectura empresarial es como el punto donde se encuentran las metas de la empresa con la tecnología que tienen, para que todo funcione bien junto. Es como asegurarse de que ambos caminen juntos y sin tropiezos. Esto ayuda a sacar mejor provecho de los recursos disponibles y a que los diferentes sistemas puedan “entenderse” y trabajar juntos sin problemas.

Hay varias metodologías y marcos que ayudan a guiar cómo implementar la arquitectura empresarial. Algunos de los más conocidos se encuentran TOGAF, el Marco de Zachman, FEAF y el marco de Gartner. TOGAF, que es uno de los más usados en todo el mundo. Su base es un método llamado ADM, que es un ciclo que se repite y que incluye etapas como definir la visión, diseñar la arquitectura del negocio, los sistemas de información y la tecnología, planear cómo hacer la migración, y también la gobernanza e innovación [4], [5]. Por otro lado, está el Marco de Zachman, que es como una tabla organizada alrededor de seis preguntas clave: qué, cómo, dónde, quién, cuándo y por qué. Estas preguntas se aplican desde distintos puntos de vista dentro de la organización para entender mejor su estructura [2].

Luego está FEAF, un marco creado por el gobierno de Estados Unidos, que busca que las agencias públicas manejen sus tecnologías de forma estandarizada, para que todo sea más eficiente y los sistemas puedan trabajar juntos sin problemas [2].

Y también está el marco de Gartner, que pone énfasis en los resultados para el negocio, con un enfoque es ágil y flexible, siempre buscando aportar valor y mejorar de forma continua [2].

Comparación entre Marcos de trabajo de Arquitectura Empresarial				
	Enfocado en	Fortalezas	Limitaciones	Ideal para
TOGAF	Proceso de desarrollo de arquitectura	Método integral (ADM) adoptado ampliamente	Complejo y necesita entrenamiento	Organizaciones grandes que buscan estructura
Zachman	Clasificación y vistas	Muy estructurado, bueno para la documentación	No es metodología, faltan guías de implementación	Organizaciones que necesitan estándares sólidos de documentación
FEAF	Alineación de Tecnologías de información Gubernamental	Estandarización entre agencias gubernamentales	Centrado en el gobierno de EE.UU	Sector público y agencias federales
Gartner	Resultados estratégicos	Fuerte alineación entre negocios y T.I., adaptable	Más conceptual	Organizaciones enfocadas en resultados estratégicos

Fig. 1. Comparación entre marcos de trabajo de arquitectura empresarial

Basado en el cuadro comparativo de la Fig. 1, se considerará que el marco de referencia de AE a utilizar en este trabajo será TOGAF, por su capacidad para ofrecer un enfoque estructurado y detallado en el desarrollo y gestión de arquitecturas empresariales complejas. Mediante su ADM, proporciona un ciclo de vida claro para la creación de arquitecturas alineadas con los objetivos estratégicos de la organización. Lo bueno de su flexibilidad es que se adapta bien a lo que necesita OLE, un proyecto público donde participan muchas personas y procesos que están muy conectados. Por lo que, ayuda a pensar a largo plazo, para que la arquitectura pueda crecer y cambiar conforme el entorno y los usuarios lo requieran. Como muchas organizaciones gubernamentales la usan y hay una comunidad fuerte de expertos detrás, se cuenta con recursos y buenas prácticas para apoyar el proyecto. De esta manera, no solo se consigue que la plataforma esté en sintonía con los objetivos estratégicos, sino que también se mantenga funcionando bien y pueda sostenerse a lo largo del tiempo.

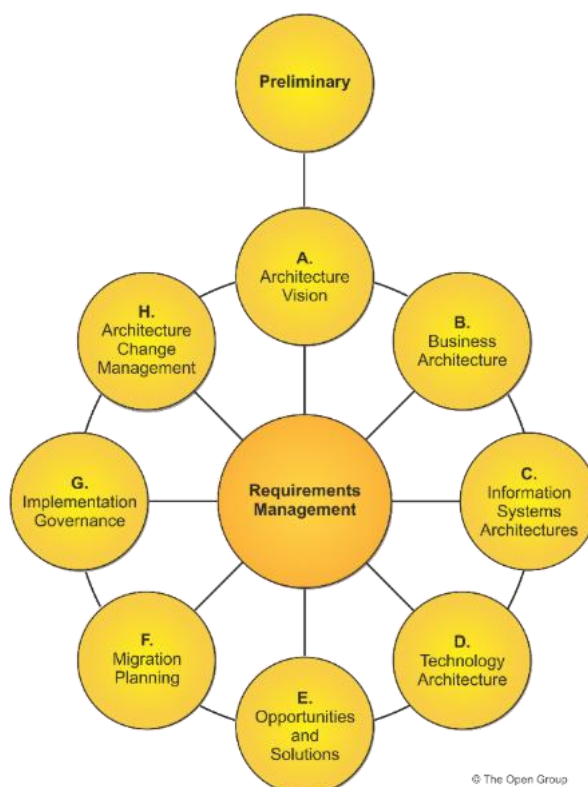


Fig. 2. Método de desarrollo de arquitectura (ADM) [1]

El ADM del marco de referencia TOGAF detalla un método estructurado y cíclico para el desarrollo de la arquitectura empresarial en las organizaciones, tal como se observa en la Fig. 2.

El punto de partida es la fase preliminar, en la cual se establecen los principios, marcos de referencia y capacidades necesarias para llevar a cabo la arquitectura. Luego, en la fase A, básicamente se trata de entender qué es la

arquitectura, qué se quiere conseguir, hasta dónde llega el proyecto y quiénes están involucrados, las personas claves. Luego, en la fase B, se organiza la Arquitectura de Negocio, es decir, cómo funcionan las tareas, procesos, servicios y la estructura de la empresa que apoyan la estrategia. La fase C se divide en dos partes: una que se encarga de la Arquitectura de Datos y otra de la Arquitectura de Aplicaciones, para asegurarse de que la información y los sistemas estén alineados con los procesos de la empresa. Finalmente, en la fase D, se describe la Arquitectura de Tecnología, detallando el hardware, el software y las redes que hacen que todo funcione correctamente [6], [7].

Cuando ya se tienen claras las arquitecturas, en la fase E se empiezan a buscar oportunidades y soluciones, y se traza un plan claro para llevarlas a cabo. Después, en la fase F, se crea un plan detallado para hacer la migración, organizando paso a paso cómo se hará la transición para que todo sea lo más suave posible. La fase G trata de supervisar la gobernanza de la implementación, de modo que los diseños mantengan los modelos arquitectónicos. En último lugar, la fase H acomoda el continuo cambio y permite modular la construcción para ser fácil de cambiar ahora cuando el entorno o necesidades surgen. Durante todo el proceso también permanece operativa la fase de Gestión de Requerimientos, tratando de mantener rastreables y adecuadas las soluciones a las necesidades del negocio. Este método iterativo y sencillo ayuda a las organizaciones a avanzar de forma coherente y establecida [6], [7].

1.4.2 Sistemas de información

Un sistema de información se define como un conjunto de elementos o componentes que interactúan entre sí para alcanzar objetivos comunes. Estos sistemas funcionan mediante mecanismos de entrada, procesamiento, salida y retroalimentación, lo que permite transformar datos en información útil para apoyar decisiones y operaciones organizativas. Cada componente del sistema interactúa con los demás, formando un todo integrado orientado a metas específicas [2].

- Entrada: actividad de recopilación y captura de datos.
- Procesamiento: transformación de entradas en salidas mediante operaciones como el cómputo.
- Salida: producción de información útil (por ejemplo, informes o documentos) que puede servir como entrada para otros sistemas.
- Retroalimentación: información que se utiliza para modificar o ajustar las entradas o el tratamiento dentro del sistema.

Los sistemas de información, como parte de la arquitectura empresarial, ofrecen una visión completa de cómo funcionan los procesos dentro de una organización, junto con la infraestructura tecnológica y los datos que se manejan. Esto facilita tomar decisiones estratégicas que sean coherentes y que puedan crecer a medida que la organización avanza en su transformación digital [3].

Los sistemas de información son como el corazón de una organización. No solo sirven para planificar y manejar el trabajo diario, sino que también se aseguran de que todo marche en conjunto y sin problemas, algo que es muy importante en el sector público, donde es vital ser eficientes y claros con la información.

TOGAF es como ese mapa que ayuda a que todos los sistemas, la tecnología y los procesos estén alineados con lo que la institución realmente quiere lograr. Así, no hay confusión y todos trabajan en la misma dirección. Según lo que cuentan en [4], cuando las instituciones públicas usan TOGAF, logran que todas las partes importantes, como los datos, las aplicaciones y la tecnología, trabajen en conjunto sin problemas, funcionando de forma coordinada y armoniosa. Eso hace que los sistemas “hablen el mismo idioma”, se manejen con orden y ayuden a tomar decisiones basadas en información que se puede confiar. TOGAF no solo ayuda a que todo esté bien organizado y fácil de seguir, sino que también hace posible que los sistemas de información se mantengan útiles y se ajusten con el tiempo dentro de las instituciones públicas.

1.4.3 Metodologías de evaluación

Al evaluar un sistema de información, es importante realizar un análisis integral. No basta con verificar su velocidad de respuesta o flexibilidad; también debe considerarse la experiencia del usuario y la calidad global del software. Por eso, como dice [5], revisar y mejorar el código de vez en cuando, lo que llamamos refactorizar, contribuye a optimizar el rendimiento del sistema, evitar la acumulación de deuda técnica y garantizar su capacidad de crecimiento sin comprometer la estabilidad ni generar errores. Además, cambiar de un sistema monolítico, que es como una pieza muy grande y complicada, a un conjunto de microservicios más pequeños y organizados es una forma inteligente de avanzar

en tecnología. Según el estudio de [6], al separar el sistema en partes más pequeñas y bien definidas, con responsabilidades claras y un diseño pensado, se vuelve mucho más sencillo mantenerlo, hacerlo crecer y adaptarlo cuando surgen nuevas necesidades o se quieren hacer mejoras.

Hay ejemplos reales que lo muestran muy bien, como el caso de Netflix que menciona [7], cuando cambiaron a microservicios, su sistema se volvió más fuerte y funcionó mejor con sus objetivos, porque podían hacer cambios frecuentes sin problemas y el equipo tenía más libertad para actuar rápido. También, en [8] analizan mercados virtuales y muestran que los sistemas con microservicios crecen más fácil, trabajan mejor y son menos costosos de mantener a largo plazo.

Por otro lado, el estudio de [9] aboga por un sistema denominado “Microservices Integrated Performance and Reliability Testing” (MIPaRT), que incluye una combinación de análisis simultáneo de medidas importantísimas como tiempos de respuesta, resistencia al fallo y comportamiento en condiciones distribuidas. Esta etapa de pruebas completas es muy útil para asegurarse de que al usar microservicios se logren los objetivos planteados y se mantenga la calidad técnica que se espera.

El estudio [10] subraya lo clave que es detectar, de forma automática, los 'code smells' y los problemas de arquitectura ya en las primeras fases del desarrollo. Cuando usamos herramientas que analizan tanto el código como la estructura del sistema, podemos identificar estas cuestiones temprano, lo que eleva la calidad del software, favorece un diseño más sólido y facilita que las mejoras y correcciones sean más rápidas y eficaces. Así se reducen los problemas técnicos y se logra que los sistemas se mantengan sanos y funcionen bien a largo plazo.

1.4.4 Evaluación centrada en el usuario

Hoy en día, cuando se diseña una aplicación o una plataforma, lo primero que hay que preguntarse es: ¿esto le va a servir a la gente? ¿Le va a resultar fácil de usarlo? ¿Va a sentirse cómoda interactuando con ella? Cada vez más, el foco está en la experiencia real de las personas que usan los sistemas. Y eso va mucho más allá de que el software funcione sin errores: se trata de que sea intuitivo, agradable y que no se convierta en un dolor de cabeza. Por eso, evaluar la experiencia del usuario se ha vuelto una parte esencial del proceso. Es la forma de saber si lo que se está creando realmente está cumpliendo su propósito.

Este tipo de evaluación ha tomado fuerza, sobre todo con la llegada de tecnologías como la realidad mixta o los hologramas. En un estudio realizado por [11], en el que, con juegos tipo rompecabezas en realidad mixta, por ejemplo, se observaron aspectos como qué tan fluida era la interacción o si la gente se divertía de verdad. Eso ayudó a ajustar detalles en la interfaz y mejorar la forma en que el usuario se relacionaba con el entorno. En otro caso expuesto por [12], una revisión sobre hologramas táctiles mostró que no basta con medir si algo es técnicamente viable: también hay que mirar cómo se siente usarlo, qué tan cómodo resulta, si tiene sentido para quien lo prueba.

Pero esto no se queda solo en las tecnologías más modernas. También se ha aplicado en sistemas más cotidianos. Un buen ejemplo es el de [13], una plataforma que se diseñó en Indonesia para gestionar inventarios en situaciones de emergencia. Las evaluaciones de usabilidad permitieron detectar problemas que, a simple vista, podrían haberse pasado por alto. Ajustes en la forma de navegar el sistema o en cómo se presentaba la información marcaron una gran diferencia, sobre todo en un contexto donde el tiempo y la claridad pueden salvar vidas. También se han propuesto métodos más específicos para analizar con detalle lo que pasa cuando una persona usa un sistema. Hay estudios que se fijan, por ejemplo, en si el sistema responde bien a las acciones del usuario, si mantiene coherencia o si da la sensación de control. Todo esto ayuda a pulir los diseños, no desde la teoría, sino desde lo que realmente experimentan las personas al usar una herramienta digital [14].

Y en el ámbito educativo, esta forma de pensar también ha dado resultados valiosos. En una plataforma de realidad virtual pensada para enseñar ingeniería de software, se midió cómo afectaba al aprendizaje el hecho de estar inmerso en un entorno virtual. Se vio que la forma en que se presenta la información, y cómo se interactúa con ella, influye directamente en la motivación y el aprovechamiento de los estudiantes [15]. En otras palabras: cuando una herramienta está bien diseñada pensando en el usuario, también enseña mejor.

CAPÍTULO 2

2. METODOLOGÍA.

En este capítulo se explica cómo se llevó a cabo la evaluación de la plataforma “Observaciones Litorales del Ecuador” (OLE). Para esto, se combinó una serie de métodos de auditoría técnica, análisis detallado de la estructura del software y una revisión basada en la experiencia real de quienes usan la plataforma. La idea fue obtener una imagen completa de cómo está funcionando OLE, no solo desde el punto de vista técnico, sino también en su uso diario, con el fin de proponer mejoras que sean prácticas, viables y que respondan a lo que realmente necesita la institución.

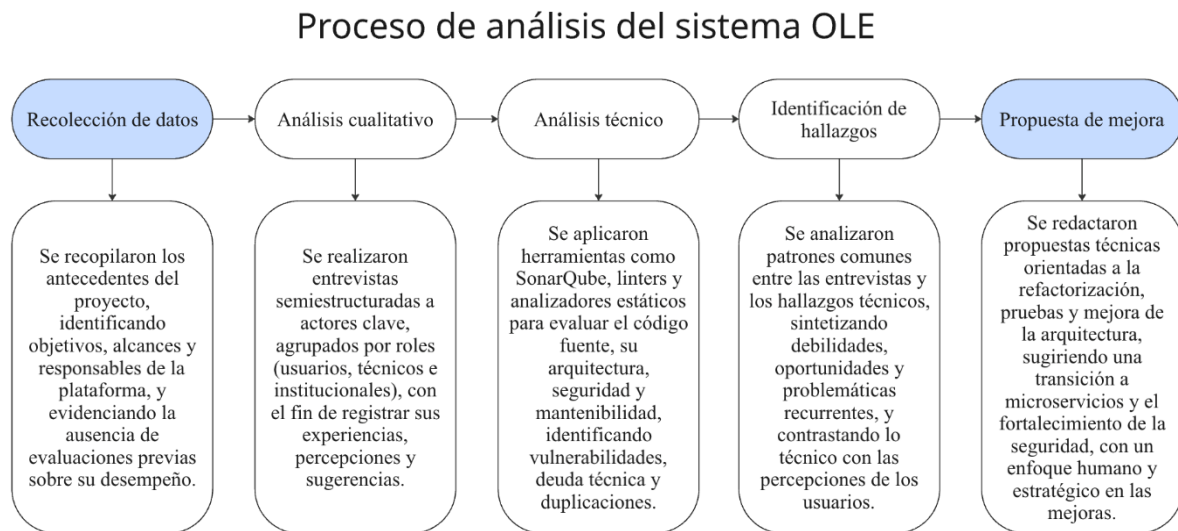


Fig. 3. Proceso de análisis

El estudio surgió con la intención de conocer a fondo el desempeño de OLE en un momento concreto de su desarrollo. Más que hacer un análisis superficial centrado solo en el software, se buscó comprender la herramienta desde varios ángulos: cómo funciona técnicamente, cómo se usa en la práctica, qué opinan quienes la manejan y cuáles son sus principales desafíos tanto a nivel funcional como estratégico. Como no había estudios previos que abordaran todos estos aspectos específicos de esta plataforma, este trabajo se convirtió en un esfuerzo novedoso para integrar lo técnico, lo humano y lo organizacional en un solo enfoque.

Se adoptó un enfoque metodológico de tipo descriptivo, no experimental y de corte transversal. Fue descriptivo en tanto se centró en caracterizar el estado actual de la plataforma, identificando sus componentes, fortalezas, debilidades y riesgos. No se intervinieron los entornos de ejecución ni se alteró el comportamiento del sistema, lo que valida su naturaleza no experimental. Asimismo, se aplicó una perspectiva transversal al captar datos en un único momento temporal, permitiendo establecer una fotografía diagnóstica del sistema tal como operaba en ese periodo.

La metodología combinó técnicas cualitativas y cuantitativas para integrar la información. Se llevaron a cabo entrevistas semiestructuradas con usuarios de distintos perfiles para recoger sus opiniones, identificar los problemas que enfrentan y entender sus expectativas. Por otro, se efectuó un análisis técnico detallado del backend y frontend de la plataforma, apoyado en herramientas especializadas como SonarQube, revisión manual de código, lectura estructural de carpetas y validación de funcionalidades desde la interfaz.

2.1 Levantamiento de Información

Para captar la dimensión humana del sistema, se aplicó entrevistas semiestructuradas a tres grupos clave: usuarios institucionales, como investigadores o técnicos que operan la plataforma en el campo o desde laboratorios; desarrolladores y mantenedores del sistema; y gestores responsables de su planificación, continuidad y sostenibilidad.

Las entrevistas se realizaron de forma remota, con grabación previa autorización, y fueron adaptadas según el perfil del entrevistado para favorecer una conversación fluida y obtener información relevante desde distintas perspectivas. El procesamiento de estas entrevistas contempló su transcripción, codificación y análisis cualitativo, lo que permitió organizar la información y relacionarla con aspectos técnicos previamente identificados. Este enfoque garantizó que las voces de los diferentes actores quedaran reflejadas en el estudio, proporcionando un insumo sólido para comprender las dinámicas de interacción con OLE y alimentar la interpretación de los datos técnicos con un componente humano.

Tabla I. Fases del ADM y su aplicación en la plataforma OLE

Fase	Descripción General	Aplicación en el Proyecto OLE
Preliminar	Establece los principios, marcos de referencia y capacidades necesarias para desarrollar la arquitectura empresarial, definiendo el alcance y los recursos del proyecto.	Se definieron los principios de arquitectura empresarial para OLE, identificando a los interesados (investigadores, gestores, comunidad) y los recursos disponibles en ESPOL, como acceso a la plataforma y documentación técnica.
A: Visión de Arquitectura	Define la visión estratégica, el alcance del proyecto, los objetivos y los interesados clave, asegurando alineación con las metas institucionales.	Se estableció una visión para modernizar OLE, enfocada en seguridad, escalabilidad y soporte al monitoreo costero, con una matriz de poder-interés para gestionar interesados
B: Arquitectura de Negocio	Describe los procesos, servicios y estructuras organizativas que apoyan la estrategia, identificando roles y funciones clave.	Se analizaron procesos como recolección y validación de datos costeros, identificando roles (investigadores, técnicos, comunidad) y sus interacciones
C: Arquitectura de Sistemas de Información	Detalla la arquitectura de datos y aplicaciones, asegurando que la información y los sistemas estén alineados con los procesos de negocio.	Se documentaron las funciones del backend (Django) y frontend (Flutter) de OLE, identificando módulos como autenticación y validación de datos, y limitaciones como código repetido
D: Arquitectura de Tecnología	Describe la infraestructura tecnológica (hardware, software, redes) que soporta los sistemas de información.	Se revisaron componentes tecnológicos de OLE (bases de datos, frameworks), proponiendo actualizaciones como Docker y CI/CD para un entorno robusto
E: Oportunidades y Soluciones	Identifica soluciones para abordar brechas arquitectónicas, definiendo oportunidades para mejorar el sistema.	Se analizaron alternativas (mejorar el monolito, reconstrucción total, microservicios), seleccionando una migración progresiva a microservicios para mayor escalabilidad
F: Planificación de la Migración	Diseña un plan detallado para implementar las soluciones, organizando la transición de manera estructurada.	Se elaboró un plan de migración hacia microservicios, con fases de implementación y cronograma para minimizar interrupciones
G: Gobernanza de la Implementación	Supervisa la implementación para asegurar que los diseños se adhieran a los modelos arquitectónicos.	Se definieron mecanismos de gobernanza, como revisiones técnicas y auditorías, para garantizar la calidad durante la implementación
H: Gestión del Cambio	Gestiona cambios continuos, adaptando la arquitectura a nuevas necesidades o entornos.	Se propusieron capacitaciones, planes de comunicación y mecanismos de retroalimentación para gestionar cambios en OLE
Gestión de Requerimientos	Asegura que los requerimientos del negocio sean rastreables y se mantengan alineados con las soluciones.	Se implementó un proceso iterativo para capturar y priorizar requerimientos de usuarios (investigadores, comunidad), integrándolos en todas las fases del ADM.

2.1.1 Fase A – Visión de arquitectura

Se definió una visión estratégica centrada en la necesidad de modernizar la plataforma OLE para que sea segura, escalable, mantenible y capaz de soportar los procesos de monitoreo costero y gestión ambiental de manera eficiente. Esta visión sirve de guía para orientar todas las fases posteriores del diseño arquitectónico, garantizando coherencia entre los objetivos institucionales y las decisiones tecnológicas.

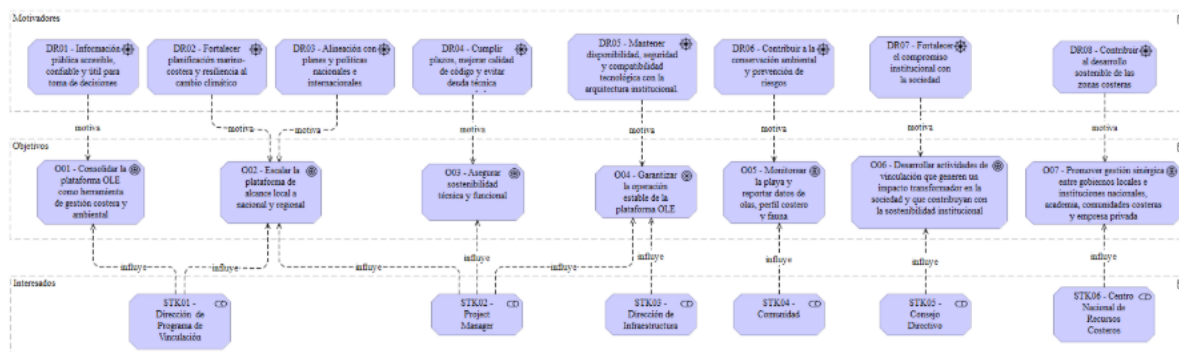


Fig. 4. Vista de interesados

La Fig. 4 nos permitió estructurar de forma precisa la información sobre cada interesado, incluyendo sus objetivos y motivadores. Esto facilitó la priorización de la gestión de interesados, asignar estrategias de comunicación adecuadas y asegurar que cada parte reciba el nivel de atención necesario según su poder e interés. Además, tener esta información consolidada agiliza la toma de decisiones entre áreas y evita omisiones de actores relevantes. En proyecto como OLE, disponer de estos puntos clave fortalece la alineación estratégica y aumenta las probabilidades de éxito y sostenibilidad del sistema.

En la Fig. 5, los interesados se ubican según su capacidad de influencia y el grado de implicación en el proyecto. En el cuadrante de gestionar de cerca se encuentra la Líder de vinculación comunitaria, ya que combina alto poder de decisión con un fuerte interés en el desarrollo y resultados de la plataforma. En mantener satisfechos se ubican el Consejo directivo y el Líder de infraestructura, pues poseen alta capacidad de influencia institucional, pero su interés es puntual y centrado en el cumplimiento de lineamientos. En mantener informados se encuentran el Project manager y la Comunidad, quienes tienen un rol activo en la operación o uso de la plataforma, pero sin poder de decisión estratégico. Esta distribución permite asignar estrategias de gestión diferenciadas para optimizar la comunicación y la colaboración.

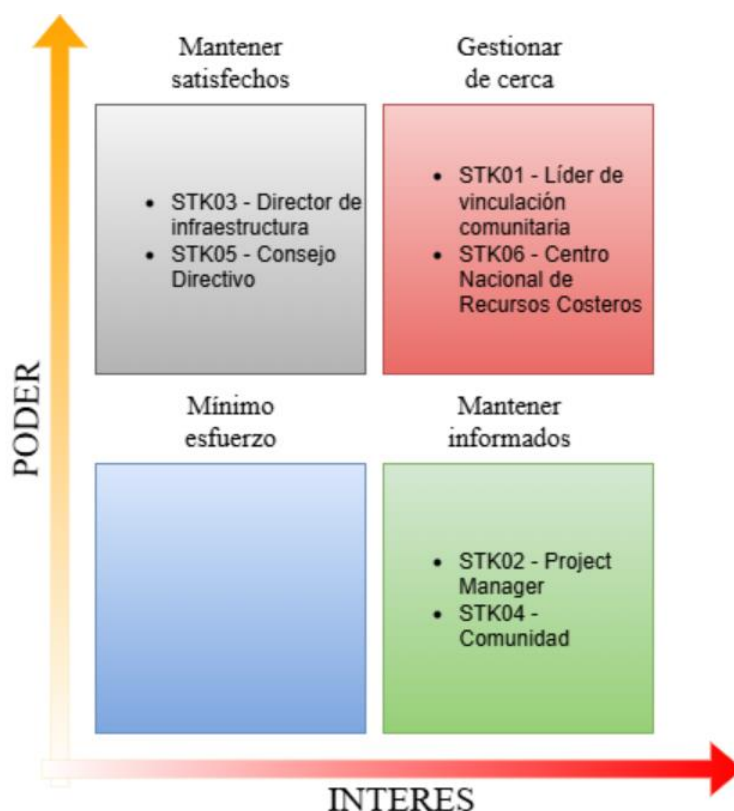


Fig. 5. Matriz de Poder- Interés

En base al levantamiento de información inicial, se definió el alcance de la Arquitectura Empresarial para este ejercicio con base a 4 dimensiones:

- **Amplitud:** Se abarcó las funciones y componentes relacionados directamente con la operación de la aplicación. Se incluyeron las funcionalidades de registro de observaciones, visualización geoespacial de datos, gestión de usuarios, recolección de datos ambientales y generación de reportes para la comunidad y la academia.
- **Nivel de detalle:** La arquitectura cubrió los cuatro dominios definidos por TOGAF. Negocio, procesos de recolección de datos, validación científica, interacción con actores comunitarios y académicos. Datos, modelos de datos ambientales, geolocalización, almacenamiento de series temporales y metadatos. Aplicaciones, estructura modular de la aplicación web y móvil, componentes de visualización y formularios de ingreso. Tecnología, infraestructura actual de la universidad, servidores, conectividad y posibles servicios en la nube.
- **Tiempo:** El análisis y rediseño arquitectónico se enmarcó en un periodo de 2 meses, con un enfoque de corto a mediano plazo. Este periodo incluyó diagnóstico de la situación actual y rediseño de la arquitectura.
- **Extensión organizacional:** El estudio se limitó a las unidades académicas responsables de la aplicación, en especial aquellas vinculadas con proyectos de investigación costera y programas de vinculación con la sociedad. Involucra también a los usuarios comunitarios locales (investigadores ciudadanos, pescadores, estudiantes), sin extenderse a otros sistemas institucionales no relacionados.

2.1.2 Fase B – Arquitectura de negocio

Se analizaron los procesos clave que la plataforma apoya, identificando a quienes participan activamente: investigadores, técnicos, gestores institucionales y miembros de la comunidad. Se detallaron las funciones principales, como la recolección y validación de datos costeros, la administración de las estaciones de monitoreo y la creación de reportes para facilitar la toma de decisiones ambientales. Esto ayudó a entender mejor cómo funciona el sistema en la práctica y el contexto en el que se usa.

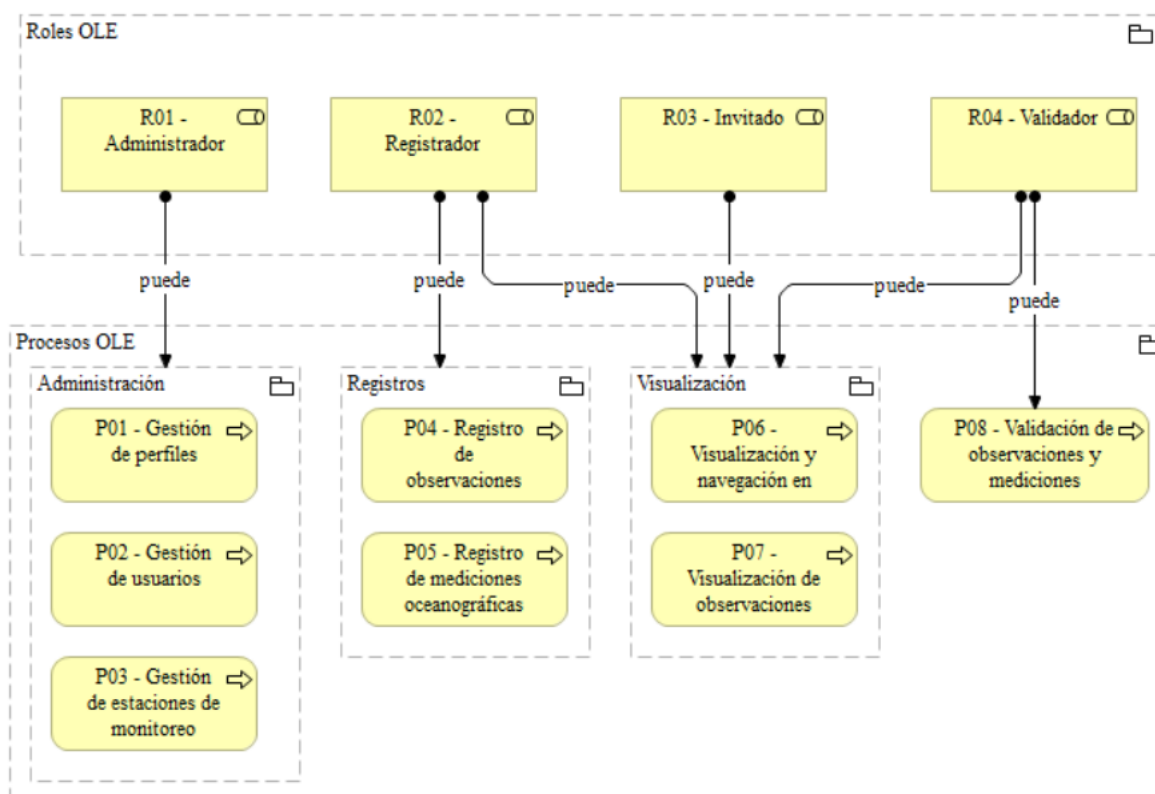


Fig. 6. Roles y procesos

En la Fig. 6, se identificó los roles y procesos que interactúan mediante la aplicación, lo que nos permitió establecer una base sólida para el diseño de una arquitectura alineada con los objetivos del sistema y las necesidades de los usuarios. Este enfoque facilitó la definición clara de responsabilidades y aseguró que cada funcionalidad esté vinculada a un actor específico, lo que contribuyó a una mejor eficiencia operativa. Además, desde la perspectiva de la arquitectura empresarial, pudimos conocer estos elementos desde etapas tempranas y permitió modelar adecuadamente los dominios de negocio y aplicación, optimizando el desarrollo, la escalabilidad y el mantenimiento del sistema a largo plazo.

2.1.3 Fase C – Arquitectura de sistemas de información

Se registraron las funciones que la plataforma tiene hoy en día y cómo fluye la información entre sus diferentes partes, como se muestra en la Fig. 7. El backend, construido con Django, y el frontend móvil desarrollado en Flutter, soportan varios módulos importantes como autenticación, registro y validación de observaciones, mediciones y gestión de usuarios. Aunque la plataforma cumple con sus funciones, se notaron algunos detalles que complican su mejora. Por ejemplo, hay bastante código repetido, no se hacen muchas pruebas automáticas y no existen procesos claros que faciliten el trabajo de desarrollo. Todo esto hace que avanzar y hacer cambios en el sistema sea más complicado de lo que debería.

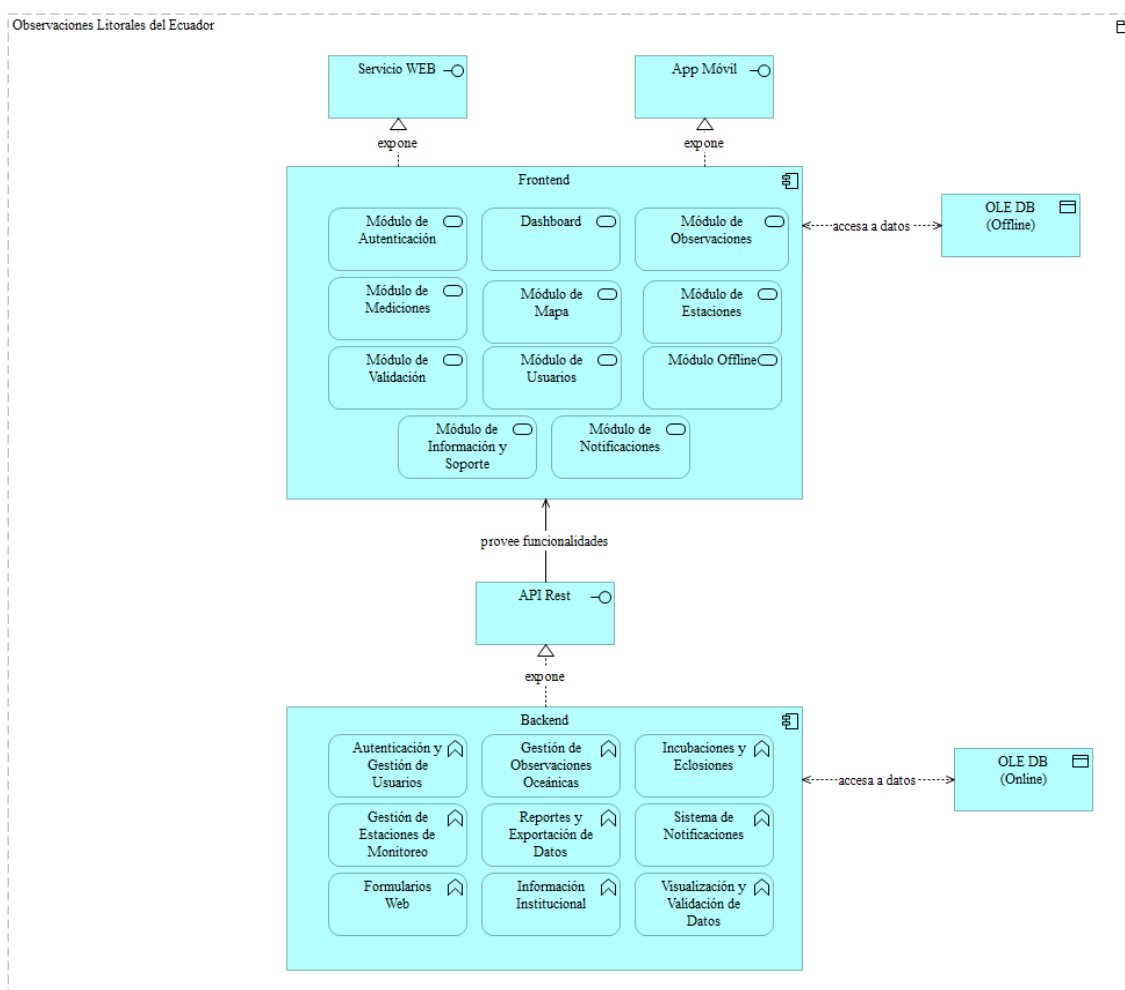


Fig. 7. Situación actual: Arquitectura de Sistemas de Información

2.1.4 Fase D – Arquitectura de tecnología

Se revisaron todos los componentes tecnológicos que usa la plataforma, desde las bases de datos hasta los frameworks, librerías y las herramientas para desarrollar y desplegar. Quedó claro que es necesario actualizar la tecnología que se emplea, incorporando arquitecturas más modernas, el uso de contenedores como Docker, la orquestación con Kubernetes y procesos automatizados de integración continua (CI/CD), asegurando un entorno robusto, seguro y flexible.

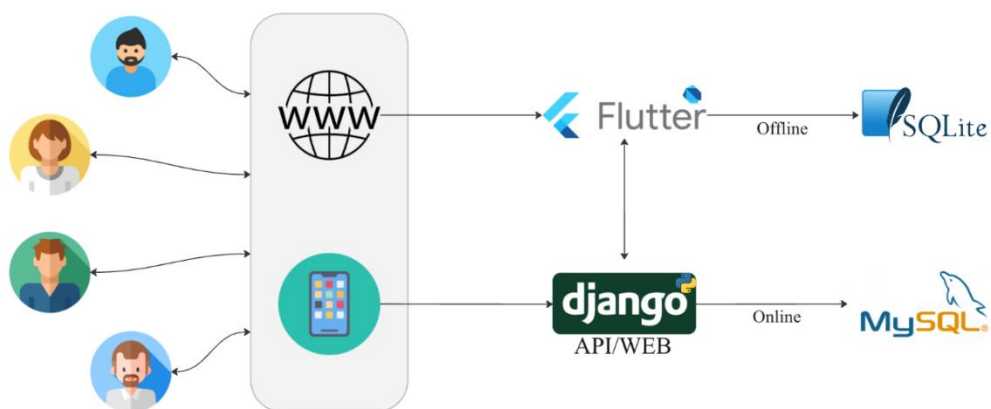


Fig. 8. Situación actual: Tecnología

2.2 Diagnóstico de Situación Actual

Como parte central de la metodología, se realizó un diagnóstico técnico detallado de la plataforma "Observaciones Litorales del Ecuador" (OLE), abordando tanto su versión web como móvil. El objetivo fue evaluar la calidad interna del software, su arquitectura, nivel de seguridad, mantenibilidad y grado de preparación para escalabilidad futura.

2.2.1 Evaluación de la plataforma web

Se empleó la herramienta SonarQube para realizar un escaneo estático del código fuente backend desarrollado en Django. Este análisis arrojó métricas clave como la cobertura de pruebas, complejidad ciclomática, duplicación de código, violaciones a estándares de codificación y niveles de deuda técnica acumulada. De manera complementaria, se realizó una revisión manual de la estructura de carpetas, vistas, modelos, controladores y archivos de configuración para detectar problemas de arquitectura y organización del proyecto.

Los resultados evidenciaron un estado crítico. El sistema se ejecuta sobre Django 3.1, una versión obsoleta que ya no recibe actualizaciones de seguridad, y contiene bibliotecas con vulnerabilidades conocidas. El modo de depuración se encuentra activo en producción, lo que representa un riesgo serio de exposición de información sensible. Asimismo, no se han configurado cabeceras de seguridad HTTP como CSP, HSTS o X-Frame-Options, lo que lo hace susceptible a ataques comunes como XSS o clickjacking.

Desde el punto de vista de la arquitectura, el backend está construido como un monolito bastante rígido, donde la lógica del negocio está concentrada en archivos muy extensos. Los módulos están poco conectados entre sí, pero hay cerca de un 20 % de código repetido. Este tipo de estructura hace que mantener y mejorar el sistema sea todo un reto. Las pruebas para asegurarse de que todo funciona bien se vuelven complicadas, y escalar o adaptar el sistema con el tiempo se vuelve muy difícil. Además, al no contar con pruebas automáticas ni procesos que ayuden a actualizar el software de forma constante, se pierde un buen control sobre la calidad y las diferentes versiones del sistema, lo que puede traer problemas a futuro.

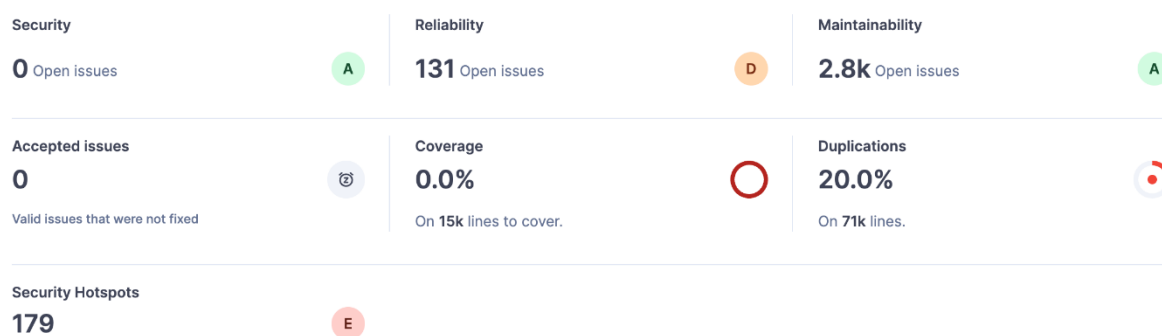


Fig. 9. Análisis de calidad de código Backend

Aunque la plataforma OLE ofrece una gran variedad de funciones, con más de 25 páginas web, más de 50 endpoints REST y módulos completos para autenticación, validación científica, generación de reportes y gestión de usuarios y estaciones, esta riqueza funcional no está acompañada por una base técnica fuerte. La arquitectura no es lo suficientemente robusta ni se siguen prácticas modernas de desarrollo, lo que crea una brecha importante entre lo que la plataforma puede hacer y la calidad real del software que la sostiene.

2.2.2 Evaluación de la aplicación móvil

En cuanto al frontend, desarrollado con Flutter, se hizo una revisión completa de cómo está organizado el proyecto. Se encontró que la estructura es bastante modular, con las carpetas bien divididas según funciones específicas como vistas, widgets y servicios. También se implementó correctamente la persistencia local usando SQLite, lo que permite que la aplicación funcione sin conexión y luego sincronice los datos cuando haya internet. Esto es especialmente útil para quienes usan la plataforma en zonas costeras donde la conexión puede ser inestable. Sin embargo, también se detectaron deficiencias técnicas: ausencia total de pruebas unitarias y de integración, uso de dependencias desactualizadas, carencia de linters y herramientas de formateo automático, y falta de un proceso

de CI/CD que automatice el despliegue o valide la calidad del código. Tampoco se encontró documentación técnica suficiente que respalde el trabajo futuro de mantenimiento o incorporación de nuevas funcionalidades.

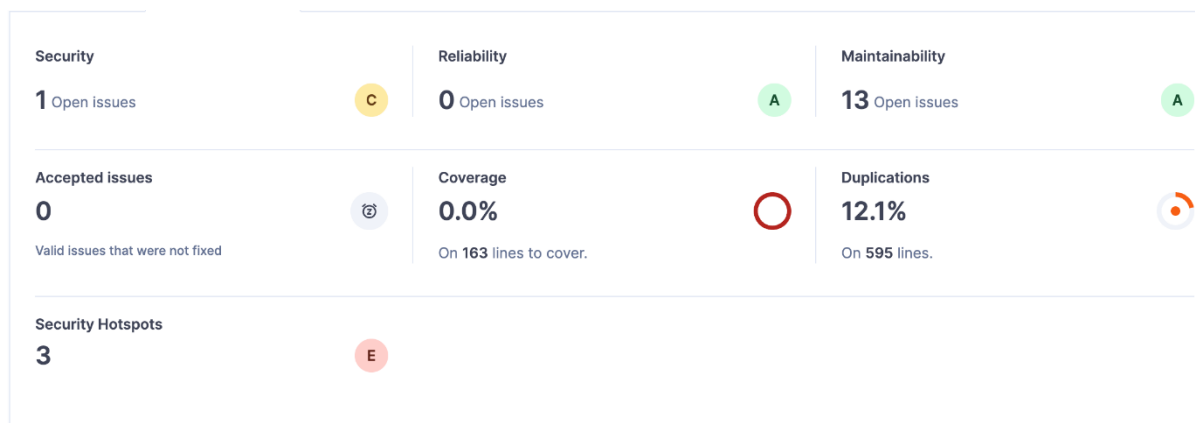


Fig. 10. Análisis de calidad de código Frontend

Esto resulta problemático si se considera la complejidad funcional que la app maneja: la posibilidad de realizar observaciones georreferenciadas, registrar variables oceanográficas, consultar mapas, validar datos científicos, administrar usuarios y recibir notificaciones automáticas. Las funciones cambian según el tipo de usuario que esté usando la plataforma, ya sea un administrador, un validador, alguien que registra datos o un invitado. Esto requiere que el sistema maneje de manera bastante cuidadosa y detallada cómo se muestran las opciones y cómo se mueve cada persona dentro de la aplicación.

Ambos entornos presentan una carencia significativa de prácticas de aseguramiento de calidad. No contar con pruebas automáticas, revisiones claras ni una buena documentación, ni tampoco con procesos que ayuden a integrar cambios de forma ágil, hace que el sistema sea más vulnerable a errores cuando está en uso. Además, esto complica que se pueda mejorar o ampliar fácilmente más adelante, lo que pone en riesgo que funcione bien a largo plazo. Que estos problemas se acumulen, sobre todo en el backend, es algo que preocupa bastante porque podría poner en riesgo la estabilidad del sistema si no se toman medidas a tiempo.

Tabla II. Evaluación de aspectos de la aplicación

Aspecto Evaluado	Web (Django)	Móvil (Flutter)	Estado Global
Arquitectura	Monolítica	Modular	Crítico
Pruebas automatizadas	No	No	Regular
Duplicación de código	20%	12.1%	Crítico
Seguridad	Vulnerable	Baja	Crítico
CI/CD	Ausente	Ausente	Crítico
Documentación técnica	Parcial	Inexistente	Regular
Deuda técnica	Muy alta	Alta	Crítico

Para la valoración cualitativa global de los aspectos técnicos evaluados en la plataforma OLE, se definió una escala simplificada con tres niveles:

- **Bueno:** Aspectos que cumplen con los estándares y no representan riesgos significativos para la operación y mantenimiento del sistema.
- **Regular:** Aspectos con deficiencias que afectan la calidad, seguridad o funcionalidad, pero cuya corrección es viable con recursos adecuados.
- **Crítico:** Aspectos con problemas graves que afectan la estabilidad, seguridad o continuidad del sistema y que requieren atención prioritaria para evitar impactos mayores.

2.3 Análisis de Brechas

Del diagnóstico realizado se desprende que la arquitectura monolítica actual representa uno de los mayores cuellos de botella del sistema. Su estructura rígida dificulta que los módulos crezcan por separado, hace más lento el desarrollo de nuevas funciones y crea una fuerte dependencia entre las diferentes partes del sistema.

Frente a esta situación, se analizaron tres rutas posibles: mantener y mejorar el monolito existente, reconstruir el sistema desde cero o migrar progresivamente hacia una arquitectura de microservicios y microfrontends.

Tabla III. Comparación de alternativas arquitectónicas para la plataforma OLE

Aspecto	Mejora del Monolito	Reconstrucción Total	Migración a Microservicios (Elegida)
Arquitectura	Monolito modular	Nuevo sistema desde cero	Microservicios + microfrontends modulares, migración gradual
Backend	Django 4.x, MySQL	Stack nuevo (FastAPI, Node.js, etc.)	Django + FastAPI/Flask por servicio
Frontend	Flutter optimizado	Flutter o nueva tecnología	Microfrontends en Angular/Flutter desacoplado por dominio
Base de Datos	MySQL central	Base de Datos rediseñada	BD por dominio o MySQL compartido
CI/CD	GitHub Actions parcial	Pipeline completo desde cero	CI/CD progresivo por módulo
Contenedores	Opcional	Docker + Kubernetes obligatorio	Docker gradual por servicio
Orquestación	No aplica	Kubernetes desde el inicio	Docker Compose → Kubernetes
Escalabilidad	Limitada	Alta	Alta, por componente
Desarrollo paralelo	Difícil	Complejo	Fácil por separación de servicios
Impacto en usuarios	Bajo	Alto	Bajo, controlado por fases
Costo inicial	Bajo	Muy alto	Medio, escalonado
Riesgo técnico	Medio, puede arrastrar errores	Alto, posible fracaso	Bajo, mitigado por etapas
Sostenibilidad	Limitada a corto plazo	Alta si se ejecuta bien	Alta y adaptable
Justificación	Solución temporal	Muy costosa y riesgosa	Equilibrio entre continuidad y modernización

Tras evaluar ventajas, costos y riesgos, se optó por la tercera alternativa, ya que ofrece un balance adecuado entre sostenibilidad, escalabilidad y continuidad operativa. Esta opción no solo pone al día la parte técnica del sistema, sino que también lo hace mucho más fácil de cuidar, mejorar y mantener estable cuando surgen problemas. Al usar microservicios, el sistema se divide en pequeñas partes que funcionan por separado, cada una encargada de una tarea específica, lo que hace que todo sea más flexible y sencillo de manejar. Esto facilita que varios equipos trabajen al mismo tiempo, que las nuevas funciones se implementen poco a poco y que el sistema siga funcionando incluso si alguna parte falla.

En paralelo, la implementación de microfrontends utilizando Angular facilitará el desacoplamiento del frontend, permitiendo que diferentes módulos (por ejemplo, registro de observaciones, validación, visualización de datos) sean desarrollados, desplegados y mantenidos por separado. Esta estrategia será especialmente útil para incorporar nuevas funcionalidades sin afectar el núcleo del sistema, mejorar la experiencia de usuario de manera segmentada, y escalar el sistema visualmente sin comprometer la estabilidad de toda la interfaz.

A partir de esta decisión, se definió un conjunto de requerimientos funcionales y no funcionales que orientarán el rediseño progresivo de la plataforma. Entre las funciones que se necesitan están dividir los procesos en módulos claros, como el registro, la validación y la visualización de datos; manejar de forma separada a los distintos usuarios y sus roles; y crear paneles de análisis específicos para cada área o dominio. En el ámbito no funcional, se priorizan metas como la reducción de los tiempos de carga a menos de dos segundos, el aumento de la disponibilidad mediante contenedores y despliegues controlados, el aseguramiento de la trazabilidad con sistemas de monitoreo y logging estructurado, la portabilidad del sistema mediante contenerización progresiva y la mantenibilidad técnica a través de pruebas automatizadas, integración continua y separación de responsabilidades por módulo. Estos requerimientos ayudarán a saber si vamos por buen camino en cada etapa del proceso y a asegurarnos de que la nueva arquitectura realmente responda a las metas estratégicas, técnicas y operativas que buscamos para la plataforma OLE.

Estas decisiones técnicas fueron fundamentadas en el análisis detallado realizado y servirán como línea base para evaluar la evolución del sistema una vez aplicadas las fases del plan de mejora. Además, refuerzan el compromiso con una arquitectura sostenible, alineada con principios modernos de ingeniería de software y capaz de responder de forma efectiva a los desafíos futuros de la plataforma OLE.

2.3.1 Fase E – Oportunidades y soluciones

Después de evaluar las distintas opciones y sus riesgos, se decidió optar por la tercera alternativa porque ofrece un buen equilibrio entre sostenibilidad, escalabilidad y continuidad operativa. No solo renueva la base técnica del sistema, sino que también crea un entorno más favorable para que sea más fácil de mantener, evolucionar y que resista mejor frente a problemas.

Al usar microservicios, el sistema se fragmentará en partes independientes en el backend, donde cada una se encargará de una función concreta. Esto facilitará que varios equipos trabajen al mismo tiempo, que se puedan agregar funcionalidades poco a poco y que el sistema soporte mejor los posibles fallos.

Al mismo tiempo, el uso de microfrontends con Angular permitirá separar el frontend en módulos independientes, como el registro de observaciones, la validación o la visualización de datos. Cada uno podrá ser desarrollado, actualizado y mantenido sin afectar al resto. Esto ayudará a incorporar nuevas funciones sin poner en riesgo la estabilidad general, además de mejorar la experiencia de usuario de forma segmentada y permitir que la interfaz crezca sin problemas.

Estas decisiones técnicas se basan en un análisis detallado y serán la referencia para medir cómo evoluciona la plataforma OLE una vez se implemente el plan de mejora. Además, reflejan el compromiso con una arquitectura moderna y sostenible, alineada con buenas prácticas de ingeniería de software, preparada para enfrentar los retos futuros que la plataforma pueda tener, tal como se ilustra en la Fig. 11.

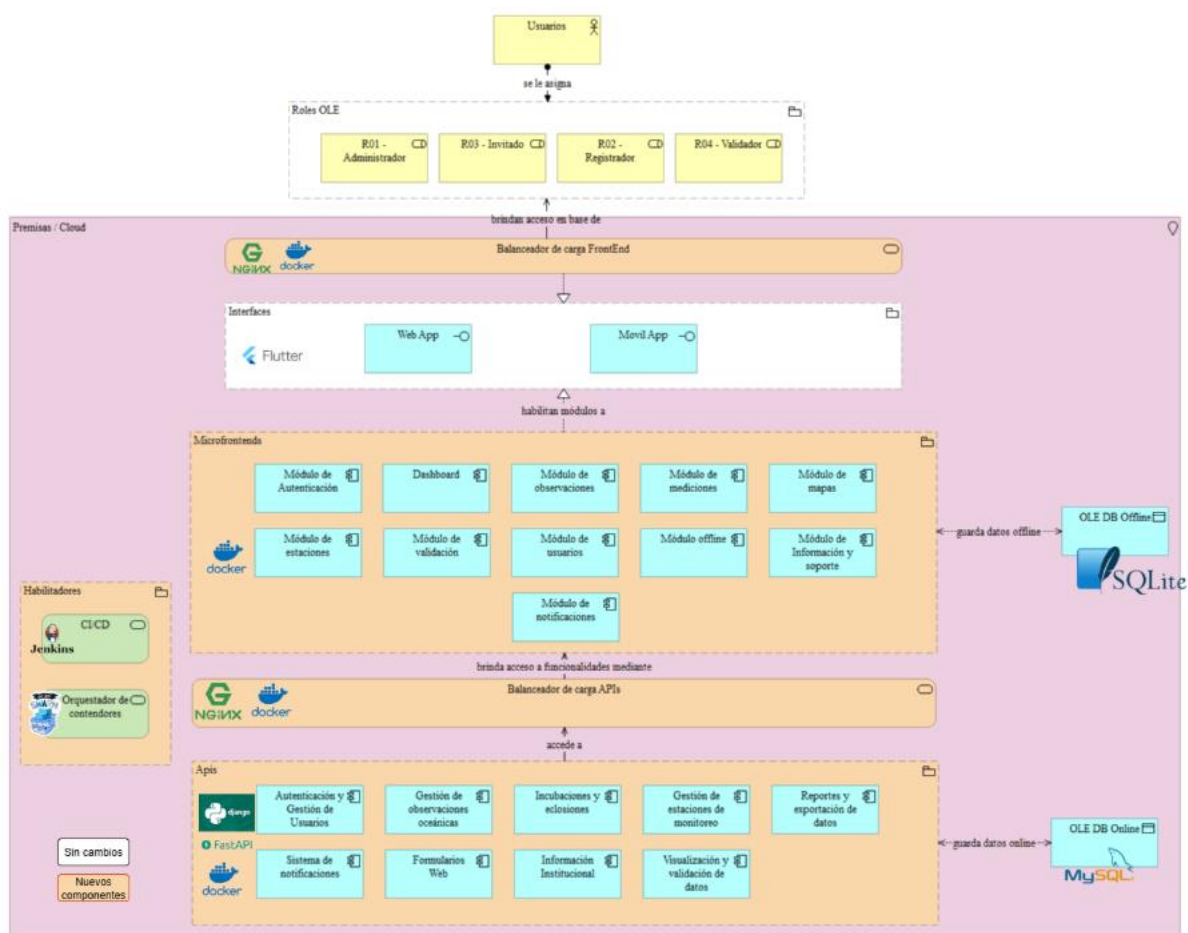


Fig. 11. OLE Arquitectura TO BE

En esta Fig. 11, se representó las diferencias de la situación actual respecto a la propuesta, destacando la implementación de microfrontends, microservicios y la adopción de tecnologías open source para las prácticas de integración continua (CI) y Despliegue Continuo (CD) que permitieron automatizar el proceso de desarrollo de software, desde la integración del código hasta su despliegue en producción, ayudando a los equipos de desarrollo a liberar software de forma más rápida y frecuente, con mayor fiabilidad.

2.4 Consideraciones Éticas y Legales

Durante todo el proceso de auditoría y al elaborar el plan de mejora, se puso mucho énfasis en cuidar aspectos éticos como la confidencialidad, la profesionalidad y el uso responsable de la información técnica. No se tocó ningún dato personal ni información de los usuarios finales. Además, se respetaron los acuerdos internos sobre quién puede acceder y usar la plataforma, asegurando que los resultados obtenidos solo se usen para mejorar la institución.

Por otro lado, desde el punto de vista legal, el análisis siguió las reglas básicas de seguridad de la información, protección de infraestructuras clave y buenas prácticas para desarrollar software seguro, basándose en normas y estándares reconocidos internacionalmente. Estas consideraciones éticas y legales forman parte esencial del proceso metodológico propuesto, asegurando que la intervención técnica sea no solo efectiva, sino también responsable y alineada con los valores institucionales.

2.5 Consideraciones Finales

Es importante destacar la relevancia de aplicar de forma estratégica las fases F G y H del ADM de TOGAF que corresponden a la planificación de la migración la implementación y la gestión del cambio arquitectónico para asegurar la mejora y sostenibilidad de la plataforma “Observaciones Litorales del Ecuador”. En la fase F es importante definir una hoja de ruta clara que ayude a priorizar las acciones necesarias para modernizar el sistema desde el punto de vista técnico. En la fase G es importante asegurarse de que los proyectos se realicen siguiendo las pautas establecidas, con contratos claros y controles que permitan una correcta ejecución. Luego, en la fase H, es clave hacer un seguimiento constante y ajustar las cosas poco a poco. De este modo, los cambios se manejan con cautela y la plataforma se mantiene conectada con lo que realmente necesita la institución. Si se trabaja con orden en estas fases, la transición será mucho más suave, se reducirán los problemas en el día a día y se sentarán las bases para que el sistema siga siendo útil y estable a largo plazo.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En este capítulo se presentan los hallazgos del análisis técnico y funcional de la plataforma “Observaciones Litorales del Ecuador”, en sus versiones web y móvil. El estudio evidenció debilidades relevantes que comprometen la estabilidad, escalabilidad y sostenibilidad del sistema, así como inconsistencias en la cobertura de pruebas, calidad del código y alineación entre roles, procesos y funcionalidades clave.

A partir de estos resultados, se definió un plan de mejora gradual orientado a optimizar el rendimiento técnico, fortalecer la gestión y coordinación del sistema, y asegurar su valor como herramienta de monitoreo ambiental costero. Los hallazgos se organizan siguiendo los dominios de negocio, datos, aplicaciones y tecnología establecidos en el marco TOGAF.

3.1 Fase F – Planificación de la Migración

Esta fase derivó en un plan de transición gradual para transformar el sistema monolítico actual en un conjunto de microservicios funcionales independientes. Este plan prioriza la migración de los módulos de mayor impacto operativo, incorpora progresivamente servicios desacoplados y prevé la implementación de pipelines automatizados que faciliten las pruebas y el despliegue continuo.

Se estima que la ejecución completa del plan requerirá aproximadamente 8,6 meses, con posibilidad de acortarse si se amplía el equipo. La hoja de ruta se organizó en pasos progresivos para minimizar riesgos y garantizar la continuidad del servicio durante la migración.

Los insumos obtenidos de entrevistas con responsables institucionales y usuarios de OLE revelaron que la plataforma, aunque cumple una función comunitaria, se utiliza principalmente para la captura de datos oceanográficos y con baja frecuencia. Asimismo, se identificó que los usuarios mayores enfrentan mayores barreras de uso, lo que refuerza la necesidad de que la migración contemple mejoras de accesibilidad y programas de capacitación continua.

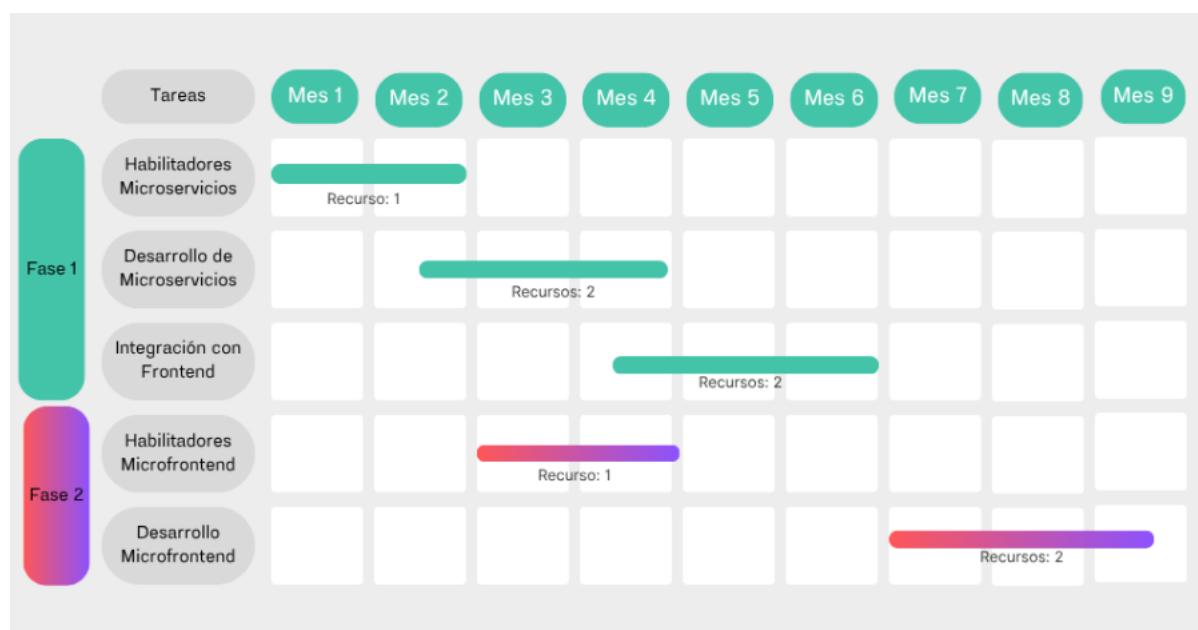


Fig. 12. Fases de implementación

3.2 Fase G – Gobernanza de la Implementación

Esta fase resultó en la propuesta de la creación de un equipo de gobernanza técnica que se encargue de velar por la calidad, la seguridad y la evolución de la plataforma. Este grupo tendría la tarea de establecer estándares técnicos, dar seguimiento a métricas de desempeño y servir de puente de comunicación entre desarrolladores, gestores y usuarios, como se muestra en la Fig. 13.

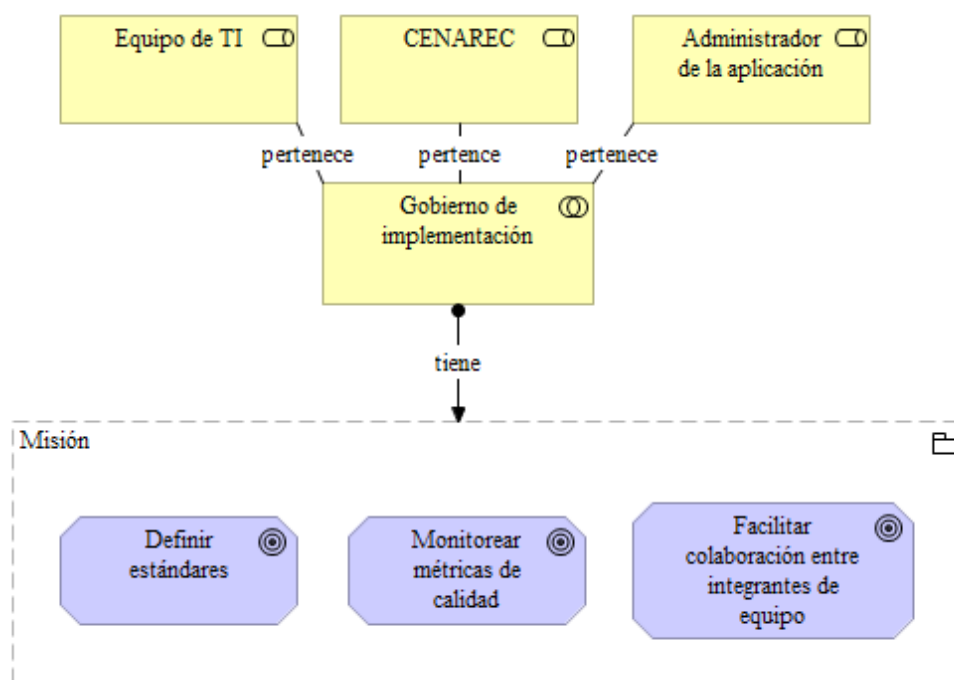


Fig. 13. Gobernanza de la implementación

Según comentaron los miembros del equipo de desarrollo y mantenimiento entrevistados, la comunicación interna ha sido en general buena. Sin embargo, mencionaron que no existe una documentación formal del código y que, en ocasiones, gestionar los despliegues resulta complicado, sobre todo en la aplicación móvil, por depender de procesos externos como los de Play Store. También hicieron hincapié en la importancia de fortalecer las prácticas y herramientas para mantener un flujo claro y ordenado de requerimientos y retroalimentación, algo que contribuiría a reforzar la gobernanza propuesta.

3.3 Fase H – Gestión del Cambio

Esta fase consolidó un plan para garantizar que la arquitectura implementada en el proyecto OLE pudiera evolucionar y ajustarse a nuevas demandas, fomentando una gestión activa del cambio a nivel organizacional, técnico y operativo. Este momento fue clave para que la nueva arquitectura tecnológica se adoptara de manera sólida y estuviera siempre en sintonía con las necesidades de la institución y las capacidades reales del sistema. Por lo que, se diseñó un Plan de Gestión del Cambio que se basa en tres aspectos fundamentales:

- Capacitaciones en las que se establecieron programas dirigidos a usuarios operativos, equipo técnico y validadores/administradores de datos, combinando sesiones presenciales, recursos virtuales y guías prácticas, con periodicidad trimestral para asegurar transferencia efectiva de conocimiento.
- Plan de comunicación en la que se propuso un esquema de comunicación bidireccional, con boletines informativos, reuniones periódicas y un módulo de notificaciones en la plataforma para mantener informados a los usuarios internos y comunitarios.
- Mecanismos de retroalimentación en la que se definieron herramientas para recoger y analizar de forma continua la opinión de los usuarios, mediante formularios digitales, espacios de revisión y paneles de control con métricas de rendimiento.

Este enfoque busca minimizar la resistencia al cambio, detectar oportunidades de mejora en tiempo real y mantener la evolución de la plataforma alineada con los principios de arquitectura empresarial y mejora continua del marco TOGAF.

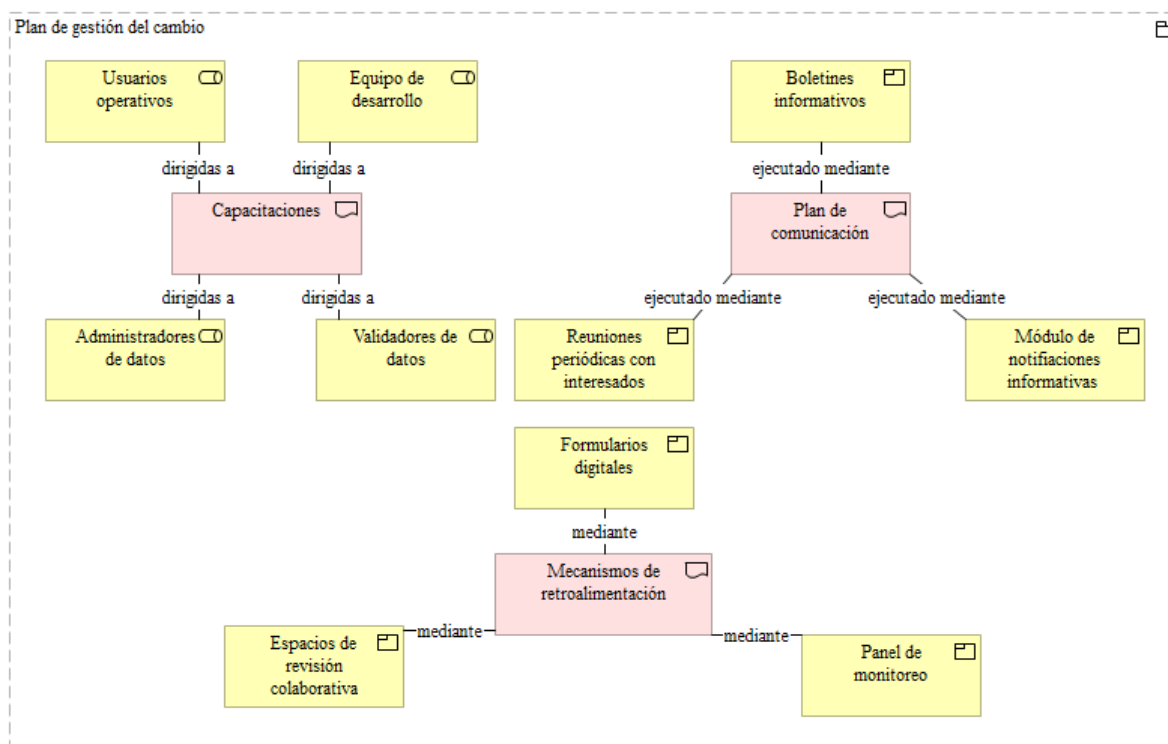


Fig. 14. Plan de gestión del cambio

3.4 Estimación del Impacto Esperado del Plan de Mejora

Con base en el diagnóstico técnico, funcional e institucional presentado, se proyecta que la implementación del plan de mejora que incluye la migración progresiva a microservicios, la adopción de microfrontends en Angular y la introducción de buenas prácticas de desarrollo y pruebas tendrá los siguientes efectos estimados sobre la plataforma OLE:

Tabla IV. Análisis de mejora esperada

Dimensión	Estado actual observado	Mejora esperada	Tipo de medición
Rendimiento	Tiempos de carga lentos (>4 seg/página)	Reducción a <2 seg mediante separación de servicios	Cuantitativa
Mantenibilidad	Código duplicado, sin pruebas, sin CI/CD	Módulos independientes, pruebas unitarias, CI/CD	Cualitativa
Experiencia de usuario	Navegación poco intuitiva, fallas de acceso	Fluidez, carga modular, interfaz modernizada	Cualitativa
Alineación estratégica	Plataforma desconectada de la planificación institucional	Integración con objetivos ambientales y territoriales	Cualitativa
Escalabilidad	Estructura monolítica, difícil de ampliar funcionalidad o recursos	Arquitectura desacoplada y modular con capacidad horizontal de escalado	Cualitativa/Cuantitativa
Disponibilidad	Caídas frecuentes al desplegar cambios o mantener el sistema	Uso de contenedores, despliegues controlados, infraestructura redundante	Cuantitativa
Seguridad	Sin autenticación robusta ni control granular de accesos	Implementación de OAuth2, roles, logs de auditoría y cifrado de datos sensibles	Cualitativa/Cuantitativa
Portabilidad	Difícil de desplegar en otros entornos, dependencias no documentadas	Contenerización (Docker), despliegues replicables y versionados en múltiples entornos	Cualitativa
Interoperabilidad	Mínima capacidad de integración con sistemas externos	Diseño basado en API RESTful, posibilidad de integración con otras plataformas	Cualitativa
Trazabilidad	Sin monitoreo ni registro de acciones o métricas	Incorporación de logs estructurados, panel de monitoreo, métricas de uso	Cuantitativa

En cuanto a la capacidad de crecimiento, la forma en que está construida la plataforma ahora limita mucho su desarrollo. El sistema principal, que usa Django 3.1, está hecho como un bloque único y rígido, lo que hace difícil agregar nuevas funciones o hacer mantenimiento sin complicaciones. Además, hay problemas con las bases de datos y cómo se sincronizan los datos, algo que los desarrolladores han notado. Todo esto afecta la estabilidad general del sistema y ha creado una deuda técnica que necesita arreglarse a fondo.

Cambiar a una arquitectura con microservicios y microfrontends permitiría dividir el sistema en partes independientes. Así, cada parte puede crecer y cambiar según lo que se necesite, sin afectar las demás. Esto facilitaría agregar nuevas funciones poco a poco, manteniendo el sistema estable. Este cambio es importante para una institución que sigue creciendo y donde los usuarios necesitan más flexibilidad.

Sobre la disponibilidad, la plataforma actual sufre interrupciones inesperadas cuando se hacen actualizaciones o mantenimientos, lo que afecta a los usuarios. La aplicación móvil, que está hecha con Flutter, funciona bien y puede usarse sin conexión, lo cual es muy útil en zonas con internet limitado. Sin embargo, no tiene suficientes pruebas automáticas ni procesos que faciliten las actualizaciones, lo que puede causar errores. Para mejorar esto, se recomienda usar tecnologías que automaticen y controlen mejor estos procesos, para que el sistema esté siempre disponible y con menos problemas para los usuarios.

En cuanto a la seguridad, la plataforma tiene algunas fallas importantes. No protege bien los datos sensibles ni controla de forma detallada quién puede acceder a qué, lo que puede poner en riesgo la información y los procesos de la institución. Aunque usa un sistema básico para controlar el acceso, no hay reglas claras ni formas fuertes de revisar lo que hacen los usuarios. Usar estándares modernos y establecer roles claros mejoraría la seguridad y la confianza en el sistema.

Sobre la portabilidad, el sistema tiene dificultades para instalarse o usarse en diferentes lugares porque el código está muy ligado y no hay ambientes separados para pruebas y producción. Además, la documentación que explica cómo funciona no es suficiente. Aunque ya se ha avanzado al usar contenedores para facilitar esto, aún hay que mejorar para que sea más fácil mantener el sistema y colaborar con otras instituciones.

En cuanto a la interoperabilidad, la plataforma funciona de forma aislada y no se conecta con otros sistemas del gobierno o la institución. Esto limita su utilidad como una herramienta completa para monitorear las zonas costeras. Crear conexiones mediante APIs permitiría que la plataforma se comuniqué con otros sistemas y mejore su alcance e impacto.

Por último, en cuanto a la trazabilidad, la falta de registros claros y sistemas de monitoreo hace difícil saber qué cambios se hicieron, analizar cómo se usa el sistema o detectar problemas rápidamente. Aunque hay algunos registros, no es suficiente para actuar a tiempo. Contar con paneles y sistemas de monitoreo ayudaría a tener una visión clara y tomar mejores decisiones para mejorar la plataforma.

Desde el punto de vista de los usuarios, reconocen que la plataforma cumple funciones importantes como registrar observaciones del océano, pero piden que se agreguen más funciones para cubrir más especies y procesos, y que haya capacitación constante para usarla mejor. Los desarrolladores, por su parte, indican que la gestión de las solicitudes de mejora es irregular y que se beneficiaría de procesos más organizados y herramientas adecuadas para asegurar un mejor trabajo en equipo y mantenimiento.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Tras aplicar las fases necesarias de la propuesta, se obtienen las siguientes conclusiones primordiales:

- La evaluación integral de la plataforma “Observaciones Litorales del Ecuador” dejó en evidencia varios puntos de mejora en su arquitectura actual. Se detectó que funciona bajo una estructura monolítica, con una elevada deuda técnica y sin contar con pruebas automatizadas que respalden su correcto funcionamiento.
- Como resultado del proceso de diseño, la aplicación del ADM de TOGAF facilitó la construcción de una arquitectura empresarial alineada con los objetivos institucionales. Este enfoque permitió modelar de manera estructurada los dominios de negocio, datos, aplicaciones y tecnología, lo cual respondió al objetivo de diseñar una arquitectura integral y estratégicamente orientada.
- En función del diagnóstico y del diseño arquitectónico propuesto, se definió un plan de mejora escalonado y sostenible. Este plan contempló la migración progresiva hacia una arquitectura desacoplada basada en microservicios y microfrontends, complementada con prácticas modernas como integración y entrega continua (CI/CD) y contenerización, asegurando la viabilidad técnica y la sostenibilidad a largo plazo.
- Finalmente, se abordó la alineación institucional y la gestión del cambio mediante el desarrollo de las fases F, G y H del marco TOGAF. Estas fases permitieron establecer una hoja de ruta realista, definir mecanismos de gobernanza y diseñar un plan de gestión del cambio que incluyó acciones formativas, estrategias de comunicación y mecanismos de retroalimentación. Todo ello fortaleció la vinculación entre la arquitectura tecnológica propuesta y los objetivos estratégicos de la organización, garantizando una transición ordenada y efectiva hacia el nuevo modelo.

4.2 Recomendaciones

Tras culminar lo planificado en la propuesta, se obtienen las siguientes recomendaciones primordiales:

- Se recomienda que el rediseño arquitectónico no solo aborde los problemas actuales, sino que también vaya incorporando poco a poco nuevas funcionalidades, como herramientas de análisis predictivo, generación automática de indicadores ambientales y una integración más fluida con otros sistemas institucionales. Así, la plataforma ganará en utilidad para respaldar la gestión del territorio y la planificación de la zona costera, ajustándose con flexibilidad a los cambios y nuevas demandas que vayan apareciendo en su entorno.
- Se aconseja implementar pilotos de despliegue progresivo, priorizando los módulos críticos, como el registro de las observaciones. Así, se pueden evitar problemas mayores, controlar mejor el funcionamiento y hacer ajustes antes de usar todo el sistema.
- Se sugiere también darle mucha importancia a mejorar la documentación técnica y los procesos de mantenimiento. Es fundamental que cada cambio o actualización del sistema venga acompañado de manuales actualizados, diagramas claros, guías prácticas para los desarrolladores, información útil para los usuarios y sistemas que permitan un monitoreo constante. Todo esto ayudará a que el sistema sea más fácil de mantener y pueda crecer sin problemas a mediano y largo plazo.
- Se recomienda implementar un programa de acompañamiento continuo que combine capacitación práctica con soporte ágil. Organizar talleres regulares y habilitar canales de apoyo directo para que los usuarios se sientan cómodos y seguros al usar el sistema.
- En una perspectiva más estratégica, se sugiere explorar cómo hacer crecer la plataforma y conectar con otras iniciativas similares a través de alianzas con universidades y organismos gubernamentales que trabajan en el monitoreo costero. Con esta estrategia, la plataforma podría consolidarse como un punto de referencia en la observación ambiental tanto a nivel nacional como regional.
- Finalmente, se propone realizar evaluaciones periódicas de sostenibilidad, estableciendo una política de revisión técnica y estratégica semestral que garantice que la evolución del sistema se mantenga alineada con los objetivos institucionales, las necesidades de los usuarios y los estándares tecnológicos vigentes.

Referencias

- [1] R. van de Wetering, S. Kurnia, y S. Kotusev, “The Role of Enterprise Architecture for Digital Transformations”, *Sustainability*, vol. 13, núm. 4, 2021, doi: 10.3390/su13042237.
- [2] D. Anny, “The Role of Enterprise Architecture in Digital Transformation Initiatives”, 2025.
- [3] P. Makovhololo, M. Makovhololo, y T. Sekgweleo, “The significance of Enterprise Architecture in driving Digital Transformation on Public sectors”, *International Journal of Applied Mathematics Electronics and Computers*, vol. 9, núm. 3, pp. 35–42, 2021, doi: 10.18100/ijamec.949442.
- [4] A. Abdallah, A. Abran, y M. A. Khasawneh, “Enterprise architecture measurement: A systematic literature review”, *J Theor Appl Inf Technol*, vol. 99, núm. 6, pp. 1257–1268, 2021.
- [5] R. S. B. Cokro y G. Wang, “Improving Architecture of Legal System by using TOGAF-ADM”, *International Journal*, vol. 12, núm. 5, 2023.
- [6] The Open Group, “The TOGAF® Standard”, abril de 2022. [En línea]. Disponible en: <https://pubs.opengroup.org/togaf-standard/>
- [7] M. L. Pasiak y A. W. R. Emanuel, “Enterprise Architecture Planning (EAP) Using TOGAF-ADM at Fuel Supplier”, en 2021 13th International Conference on Information & Communication Technology and System (ICTS), 2021, pp. 73–77. doi: 10.1109/ICTS52701.2021.9608396.
- [8] L. Zemmouchi-Ghomari, “Basic concepts of information systems”, *Contemporary Issues in Information Systems-A Global Perspective*, 2021.
- [9] H. Saputra, I. Gustiana, y E. Soeryanto Soegoto, “Design of enterprise information system using TOGAF framework for public service agency”, *Journal of Engineering Science and Technology*, vol. 17, núm. 4, 2022.
- [10] A. Yadav, M. Usman, A. Sati, y S. Jain, “Revolutionizing Software Development: Enhancing Quality and Performance Through Code Refactoring”, en *Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing*, en IC3-2024. New York, NY, USA: Association for Computing Machinery, 2024, pp. 715–725. doi: 10.1145/3675888.3676139.
- [11] M. Chebrolu, “BUILDING SCALABLE APPLICATIONS: TRANSITIONING FROM MONOLITHIC TO MICROSERVICES WITH MODULARIZED DESIGN”, *International Research Journal of Modernization in Engineering Technology and Science*, vol. 07, pp. 2582–5208, mar. 2025, doi: 10.56726/IRJMETS70797.
- [12] C. Henríquez, J. D. R. Valencia, y G. S. Torres, “Architectural Evolution at Netflix: A Case Study on Microservices and the Transformation from Monolithic to Scalable Systems.”, *Prospectiva*, vol. 23, núm. 1, 2025.
- [13] M. De Luca, S. Di Meglio, A. R. Fasolino, L. L. L. Starace, y P. Tramontana, “Automatic Assessment of Architectural Anti-patterns and Code Smells in Student Software Projects”, en *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, en EASE '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 565–569. doi: 10.1145/3661167.3661290.
- [14] M. Camilli, A. Guerriero, A. Janes, B. Russo, y S. Russo, “Microservices integrated performance and reliability testing”, en *Proceedings of the 3rd ACM/IEEE International Conference on Automation of Software Test*, en AST '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 29–39. doi: 10.1145/3524481.3527233.
- [15] N. N. Nayim, A. Karmakar, M. R. Ahmed, M. Saifuddin, y Md. H. Kabir, “Performance Evaluation of Monolithic and Microservice Architecture for an E-commerce Startup”, en 2023 26th International Conference on Computer and Information Technology (ICCIT), dic. 2023, pp. 1–5. doi: 10.1109/ICCIT60459.2023.10441241.
- [16] T. P. de Campos, M. da S. Hounsell, E. F. Damasceno, y N. M. C. Valentim, “Evaluating Usability, User Experience, and Playability of a Puzzle Game in Mixed Reality”, en *Proceedings of the 26th Symposium on Virtual and Augmented Reality*, en SVR '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 61–70. doi: 10.1145/3691573.3691584.
- [17] T. P. De Campos, E. F. Damasceno, y N. M. C. Valentim, “Usability and User Experience Evaluation of Touchable Holographic Solutions: A Systematic Mapping Study”, en *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems*, en IHC '23. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3638067.3638071.

- [18] N. U. Handayani, W. D. Kusuma, Z. F. Rosyada, Y. Widharto, y A. Hanifah, “Usability Evaluation of ‘Inventory Information System’ Design of Disaster Management in Yogyakarta Province - Indonesia”, en Proceedings of the 2020 International Conference on Engineering and Information Technology for Sustainable Industry, en ICONETSI '20. New York, NY, USA: Association for Computing Machinery, 2020. doi: 10.1145/3429789.3429843.
- [19] B. A. Santos, E. G. Q. Palmeira, F. F. Mendes, y A. B. de Sales, “A Method for Evaluating Usability Mechanisms in Software Systems”, en Proceedings of the XXIII Brazilian Symposium on Human Factors in Computing Systems, en IHC '24. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3702038.3702063.
- [20] F. A. Fernandes, D. C. B. Castro, C. S. C. Rodrigues, y C. M. L. Werner, “Evaluating User Experience of a Software Engineering Education Virtual Environment”, en Proceedings of the 24th Symposium on Virtual and Augmented Reality, en SVR '22. New York, NY, USA: Association for Computing Machinery, 2024, pp. 137–141. doi: 10.1145/3604479.3604516.

Apéndice A

Cuestionarios de preguntas de los grupos de usuarios entrevistados

Usuarios Institucionales de la Plataforma OLE

Objetivo: Conocer la experiencia de uso, percepción de utilidad, problemas funcionales y recomendaciones desde el punto de vista de quienes utilizan la plataforma en su trabajo diario.

Sección A: Datos Generales

1. Cargo o rol institucional: _____
2. Tiempo de uso de la plataforma OLE:
 - ☐ Menos de 6 meses
 - ☐ 6 meses a 1 año
 - ☐ 1 a 2 años
 - ☐ Más de 2 años

Sección B: Usabilidad y Experiencia

1. ¿Con qué frecuencia utiliza la plataforma?
 - ☐ A diario
 - ☐ Semanalmente
 - ☐ Mensualmente
 - ☐ Esporádicamente
2. ¿Cómo califica la facilidad de uso general de la plataforma?
 - ☐ Muy fácil
 - ☐ Fácil
 - ☐ Difícil
 - ☐ Muy difícil
3. ¿Qué funcionalidades utiliza con mayor frecuencia?
(respuesta abierta)
4. ¿Ha tenido problemas al acceder o navegar en la plataforma?
 - ☐ Sí
 - ☐ No
 - ☐ En caso afirmativo, ¿cuáles? (respuesta abierta)
5. ¿Qué mejoras cree que serían prioritarias?
(respuesta abierta)

Usuarios Desarrolladores o Equipo Técnico

Objetivo: Obtener una visión técnica desde quienes han trabajado o trabajan con el código de OLE (web o móvil), sobre su estructura, mantenibilidad y herramientas disponibles.

Sección A: Perfil técnico

1. ¿En qué parte del sistema ha trabajado?
 - ☐ Backend (Django)
 - ☐ Frontend móvil (Flutter)
 - ☐ Infraestructura/DevOps
 - ☐ Otro: _____
2. ¿Durante cuánto tiempo ha participado en el desarrollo o mantenimiento del sistema?
 - ☐ Menos de 6 meses
 - ☐ 6 meses a 1 año
 - ☐ Más de 1 año

Sección B: Diagnóstico técnico

1. ¿Considera que el sistema actual es fácil de mantener?
 - ☐ Sí
 - ☐ No
 - ☐ ¿Por qué? (respuesta abierta)
2. ¿Qué limitaciones técnicas ha identificado en la arquitectura actual? (respuesta abierta)
3. ¿Existen pruebas automatizadas en los módulos que usted revisó o desarrolló?
 - ☐ Sí
 - ☐ No
 - ☐ Parcialmente
4. ¿Qué herramientas o prácticas considera que deberían incorporarse para mejorar la calidad del software? (respuesta abierta)
5. ¿Estaría de acuerdo con una migración hacia microservicios u otra arquitectura moderna?
 - ☐ Sí
 - ☐ No
 - ☐ ¿Por qué? (respuesta abierta)
6. ¿Qué riesgos o ventajas ves en dicha migración? (Respuesta abierta)
7. ¿Qué tan viable consideras técnicamente una transición hacia microservicios en el caso de OLE?
 - ☐ Muy viable
 - ☐ Viable con ajustes
 - ☐ Difícil de implementar
 - ☐ No viable
 - ☐ ¿Por qué? (respuesta abierta)

Usuarios Responsables del Proyecto

Objetivo: Identificar la visión estratégica y organizacional de la plataforma, percepción sobre su sostenibilidad, problemas recurrentes y prioridades institucionales.

Sección A: Rol institucional

1. ¿Cuál es su relación con el sistema OLE?
 - ☐ Coordinación técnica
 - ☐ Gestión institucional
 - ☐ Supervisión operativa
 - ☐ Otro: _____
2. ¿Con qué frecuencia recibe reportes o incidentes relacionados con el sistema?
 - ☐ Semanalmente
 - ☐ Mensualmente
 - ☐ Esporádicamente
 - ☐ Casi nunca

Sección B: Evaluación de sostenibilidad

1. ¿Considera que la plataforma cumple actualmente con los objetivos institucionales?
 - ☐ Sí
 - ☐ Parcialmente
 - ☐ No
 - ☐ ¿Por qué? (respuesta abierta)
2. ¿Qué aspectos del sistema considera más críticos para su sostenibilidad?
(respuesta abierta)
3. ¿Existen recursos asignados para el mantenimiento y mejora del sistema?
 - ☐ Sí
 - ☐ No
 - ☐ Parcialmente
4. ¿Qué nivel de prioridad tiene actualmente la plataforma dentro de la institución?
 - ☐ Alta
 - ☐ Media
 - ☐ Baja
5. ¿Qué expectativas tiene sobre el futuro de la plataforma OLE?
(respuesta abierta)

Responsable de la Infraestructura

Objetivo: Identificar el estado actual de la infraestructura tecnológica que soporta la plataforma OLE, comprender los desafíos técnicos y operativos relacionados con su mantenimiento, escalabilidad y seguridad, así como evaluar los recursos disponibles y las prioridades para su mejora y optimización.

Sección A: Alineación Estratégica y Expectativas (Dominio Negocio – TOGAF Fase B)

1. ¿Cuáles son las expectativas del área de Infraestructura respecto a la futura operación y soporte de esta plataforma?
2. ¿Qué condiciones o criterios mínimos deberían cumplirse para aceptar la plataforma en el entorno institucional?
3. ¿Qué preocupaciones tiene el área sobre la sostenibilidad técnica de esta solución en el mediano y largo plazo?

Sección B: Requisitos Técnicos de Aplicaciones (Fase C – Arquitectura de Aplicaciones)

1. ¿Existen lineamientos definidos sobre los lenguajes, frameworks o servidores de aplicaciones permitidos institucionalmente?
2. ¿Qué requisitos de integración deberían considerarse con otras plataformas existentes o servicios externos?
3. ¿Cómo debería gestionarse el versionamiento, despliegue y mantenimiento del código fuente desde infraestructura?

Sección C: Consideraciones de Datos (Fase C – Arquitectura de Datos)

1. ¿Qué tipo de base de datos se recomienda institucionalmente para nuevos desarrollos?
2. ¿Existen políticas vigentes para el almacenamiento, respaldo y recuperación de datos?
3. ¿Qué criterios de calidad de datos considera críticos para aceptar un sistema en producción?

Sección D: Arquitectura Tecnológica (Fase D – Tecnología)

1. ¿Cuál es la arquitectura para nuevas aplicaciones: on-premise, cloud o híbrida?
2. ¿Qué limitaciones actuales tiene el entorno tecnológico que podrían afectar el despliegue?
3. ¿Qué infraestructura está disponible para ambientes de desarrollo, pruebas y producción?
4. ¿Cuáles son las recomendaciones sobre balanceo de carga, alta disponibilidad o escalabilidad?

Sección E: Seguridad, Gobierno y Gestión Operativa

1. ¿Qué requisitos mínimos de seguridad debe cumplir la aplicación antes de pasar a producción (autenticación, cifrado, trazabilidad)?
2. ¿Qué herramientas o procesos usa infraestructura para el monitoreo de aplicaciones en producción?
3. ¿Cuál es el procedimiento institucional para recibir una nueva aplicación en ambientes oficiales?

Sección F: Proyección y Preparación para el Soporte

1. ¿Qué métricas o indicadores de desempeño se esperan que la plataforma entregue una vez esté operativa?
2. ¿Qué documentación técnica debe entregarse junto con la plataforma para facilitar su gestión futura?