

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Reingeniería del Software aplicada al Sistema de Evaluación Docente de la Universidad
Industrial de Santander.

Jorge Julián García Hernández y Erika Yamile Lache Blanco

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Jathinson Meneses Mendoza

Magister en Gestión y Desarrollo de Software

Escuela de Ingeniería de Sistemas e Informática.

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2026

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Dedicatoria

A mi madre, Georgina Hernández Luque,

Por ser el pilar más importante de mi vida, por tu amor incondicional y tu entrega sin límites en cada momento. Gracias por creer en mí incluso cuando yo dudaba, por cada sacrificio silencioso que hiciste para verme crecer, y por enseñarme con tu ejemplo que con esfuerzo y perseverancia todo es posible.

Cada logro que alcanzo lleva tu nombre, porque sin tu apoyo constante, tanto en lo personal como en lo profesional, este camino no hubiera sido el mismo. Este trabajo es el resultado de todo lo que me has dado: valores, dedicación y la fortaleza para nunca rendirme.

Este triunfo es tuyo, mamá.

Jorge Julián Garcia Hernández

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

A mi madre, Elive Blanco Esteban,

y a mi nona, Gerardina Esteban de Blanco,

Aunque la vida no les permitió estar presentes para ver culminado este logro, su amor y su fe en mí siguen siendo el motor que impulsa cada uno de mis pasos. Cuando estuvieron a mi lado, me brindaron un apoyo incondicional y me enseñaron a creer en mis capacidades cuando yo más lo necesitaba.

Hoy, al cerrar este capítulo de mi vida, las llevo conmigo. Son el recuerdo vivo de que puedo lograr todo lo que me proponga, porque ustedes siempre lo supieron antes que yo.

Las llevo en el corazón, y este triunfo también es de ustedes.

Erika Yamile Lache Blanco

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Agradecimientos

Culminar este trabajo de grado no habría sido posible sin el apoyo de quienes, de una u otra manera, estuvieron presentes durante este proceso. A todas esas personas e instituciones, queremos dedicarles estas palabras de gratitud.

A la Universidad Industrial de Santander (UIS), por ser nuestra casa de formación y por proporcionarnos los medios, el conocimiento y el entorno necesarios para desarrollarnos como profesionales íntegros.

Al profesor Jathinson Meneses Mendoza, nuestro director de proyecto, por creer en nuestra propuesta desde el inicio, por cada observación que nos ayudó a mejorar y por su disposición constante para orientarnos en los momentos de mayor dificultad. Su acompañamiento fue clave para que este proyecto tomara forma.

A la Dirección de Tecnologías de la Información y las Comunicaciones (DTIC) de la UIS, por el soporte técnico e institucional que permitió que las decisiones de implementación del sistema estuvieran alineadas con la infraestructura real de la universidad.

A quienes estuvieron cerca durante este recorrido — familia, amigos y compañeros — gracias por el aliento, la paciencia y la confianza depositada en nosotros.

Este logro es el reflejo de todo lo aprendido y de cada persona que caminó a nuestro lado.

Índice general

Introducción	18
1 Generalidades.....	18
1.1 Planteamiento y Justificación del Problema	18
1.2 Alcance	20
2 Objetivos.....	22
2.1 Objetivo General.....	22
2.2 Objetivos Específicos.....	22
3 Marco teórico	23
3.1 Reingeniería de Software.....	23
3.1.1 Concepto de Reingeniería.....	24
3.1.2 Objetivos de la Reingeniería.....	24
3.1.3 Fases de la Reingeniería.....	25
3.1.3.1 Inventario y Evaluación del Sistema.....	25
3.1.3.2 Ingeniería Inversa.....	25
3.1.3.3 Reestructuración.	25
3.1.3.4 Reconstrucción o Rediseño.....	26
3.1.3.5 Validación y Pruebas.....	26
3.1.4 Modelos de Reingeniería de Software	26
3.1.4.1 Modelo de Reingeniería de Roger S. Pressman.....	26
3.1.4.2 Modelo de Ian Sommerville.....	27
3.1.4.3 Modelo de “Big Bang Replacement” (Reemplazo total).....	27

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.1.4.4	Modelo de Reingeniería Basada en Arquitectura (Architecture-Driven Reengineering).....	28
3.1.4.5	Modelo de Reingeniería Orientada a Componentes.	28
3.1.5	Comparación de Modelos	29
3.1.6	Modelo Seleccionado para el Proyecto y Justificación.....	30
3.1.7	Criterios para Decidir entre Reutilización y Reconstrucción	31
3.2	Enfoque Arquitectura Dirigida.....	32
3.2.1	ATAM – Metodo de Análisis de Trade-Offs	33
3.3	Evaluación Docente	34
3.4	Métodos de Evaluación.....	34
3.4.1	Encuesta a Estudiantes	34
3.4.2	Autoevaluación	35
3.4.3	Evaluación por Directivos.....	35
3.5	Metodología de Desarrollo	35
3.5.1	Prototipado Evolutivo	35
3.5.2	Scrum	36
3.6	Arquitectura de Software	36
3.6.1	Arquitectura Orientada a Microservicios.....	37
3.6.2	Arquitectura Hexagonal	37
3.6.3	Principios SOLID.....	38
3.7	Tecnologías de Desarrollo.....	39
3.7.1	Angular (Frontend)	39
3.7.2	Spring Boot (Backend)	39

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.8	Tecnologías de Persistencia	39
3.8.1	Oracle	39
3.8.2	MongoDB	40
3.9	Tecnologías de Integración y Despliegue	40
3.9.1	RabbitMQ	40
3.9.2	Docker	41
4	Metodología	41
4.1	Modelo de Reingeniería Adoptado	42
4.2	Aplicación del Modelo de Reingeniería en el Sistema EVADOC	42
4.2.1	Ingeniería Inversa Aplicada	42
4.2.2	Análisis y Diagnóstico del Sistema	43
4.2.3	Rediseño Arquitectónico	43
4.2.4	Reconstrucción e Implementación	43
4.3	Enfoque Arquitectura Dirigida	43
4.4	Alternativas Tecnológicas Consideradas	44
4.5	Fases de Desarrollo del Nuevo Sistema (Scrum DTIC)	46
4.5.1	Levantamiento y Priorización de Requerimientos	47
4.5.2	Diseño del Sistema	47
4.5.3	Implementación	47
4.5.4	Pruebas	48
4.5.5	Despliegue	48
4.6	Herramientas y Tecnologías Utilizadas	48
4.6.1	Tecnologías de Desarrollo	48

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

4.6.2	Herramientas de Gestión y Control.....	49
4.6.3	Tecnologías de Integración y Comunicación	49
4.6.4	Justificación Tecnológica.....	49
5	Resultados.....	50
5.1	Requerimientos	50
5.1.1	Ingeniería Inversa del Proceso EVADOC.....	51
5.1.1.1	Identificación de Actores y Responsabilidades.....	51
5.1.1.2	Modelado del Sistema Actual.	54
5.1.1.3	Reconstrucción del Flujo Operativo.	58
5.1.1.4	Hallazgos y Resultados de la Ingeniería Inversa.	60
5.1.2	Requerimientos no Funcionales.....	61
5.1.3	Requerimientos Funcionales.....	63
5.1.3.1	Épicas.....	65
5.1.3.2	Historias de Usuario.....	66
5.2	Diseño.	67
5.2.1	ATAM Aplicado al Sistema EVADOC	67
5.2.1.1	Árbol de Utilidad.	67
5.2.1.2	Análisis de Trade-Offs – Decisiones Arquitectónicas.....	68
5.2.2	Arquitectura Propuesta – Sistema EVADOC.....	69
5.2.2.1	Justificación de la Arquitectura frente a los RNF.	73
5.2.2.2	Diagrama de Arquitectura.	73
5.2.3	Diagrama de Procesos del Sistema EVADOC Propuesto	74

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

5.3	Implementación del Sistema Fase 1	76
5.3.1	Organización del Trabajo Realizado	76
5.3.2	Versionamiento del Proyecto Implementado	80
5.3.3	Resultados de Implementación del Backend	88
5.3.4	Resultados de Implementación del Frontend	93
5.3.5	Resultado de Despliegue	98
5.4	Pruebas y Análisis de Resultados	101
5.4.1	Pruebas Unitarias e Integración del Backend	101
5.4.2	Pruebas de Flujo sobre el Sistema Integrado	102
5.4.3	Casos de Prueba	105
5.4.4	Pruebas Funcionales	107
5.4.5	Validación del Sistema con Muestra de Población	110
5.5	Errores Detectados y Resultados	113
5.5.1	Detección mediante SonarQube	114
5.5.2	Detección mediante Pipelines CI/CD	114
5.5.3	Resultado del Proceso de Corrección	114
6	Conclusiones	114
7	Trabajo Futuro	117
	Referencias Bibliográficas	119
	Apéndices	121

Índice de tablas

Tabla 1 Comparación de Modelos de Reingeniería de Software.	29
Tabla 2 Alternativa 1: Base de Datos Mongo vs Oracle	44
Tabla 3 Alternativa 2: Sistema de Mensajería Kafka vs RabbitMQ	45
Tabla 4 Alternativo 3: Estilo Arquitectónico Monolítico vs Microservicios	46
Tabla 5 Actividades Identificadas en el Proceso Actual de Evaluación Docente	59
Tabla 6 Hallazgos Identificados en el Sistema Actual de Evaluación Docente	60
Tabla 7 Definición y Priorización de los Requerimientos no Funcionales.	62
Tabla 8 Requerimientos Funcionales Identificados en el Sistema.	63
Tabla 9 Resumen de Historias de Usuario Organizadas por Épicas	66
Tabla 10 Decisiones Arquitectónicas, Análisis de Beneficios y Sacrificios.	69
Tabla 11 Interfaces de Usuario, Tecnologías Utilizadas y Funcionalidades.	70
Tabla 12 Microservicios, Responsabilidades y Fuentes de Datos.....	71
Tabla 13 Relación entre los RNF y su Solución en la Arquitectura.....	73
Tabla 14 Módulos Implementados en la Fase 1 del Sistema EVADOC	76
Tabla 15 Distribución de Endpoints por Controlador Academic Service	88
Tabla 16 Distribución de Endpoints por Controlador Assessment Service.....	90
Tabla 17 Resultados de Pruebas Unitarias — Fase 1 EVADOC.....	102
Tabla 18 Resumen de Casos de Prueba del Sistema EVADOC	105

Índice de figuras

Figura 1	Modelo de reingeniería de software según Pressman.	27
Figura 2	Modelo de reingeniería de software basada en la arquitectura.	28
Figura 3	Configuración e inicio del proceso de evaluación.....	52
Figura 4	Recolección de evaluaciones.....	52
Figura 5	Autoevaluación docente.	53
Figura 6	Generación de reportes y consulta de resultados.....	54
Figura 7	Flujo general del proceso de evaluación docente.....	55
Figura 8	Proceso de evaluación del profesor por parte del estudiante.....	56
Figura 9	Proceso de recepción y gestión de solicitudes de ajuste.	57
Figura 10	Proceso de generación de reportes y cierre.	58
Figura 11	Árbol de utilidad del método ATAM.	68
Figura 12	Modelo de datos basado en arquitectura de microservicio.....	72
Figura 13	Arquitectura de microservicios del sistema.....	74
Figura 14	Diagrama BPMN del proceso propuesto de evaluación docente.	75
Figura 15	Análisis de calidad del academic service	77
Figura 16	Análisis de calidad del assessment service.....	78
Figura 17	Análisis de calidad del answer service.	79
Figura 18	Análisis de calidad del form service.....	80
Figura 19	Estructura raíz del repositorio academic service	82
Figura 20	Organización de paquetes src del academic service.....	82
Figura 21	Estructura raíz del repositorio answer service.....	83
Figura 22	Organización de paquetes src del answer service	83

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 23	Estructura raíz del repositorio assessment service	84
Figura 24	Organización de paquetes src del assessment service	84
Figura 25	Estructura raíz del repositorio form service	85
Figura 26	Organización de paquetes src del form service	85
Figura 27	Merge requests academic service	86
Figura 28	Merge requests answer service	86
Figura 29	Merge requests assessment service	87
Figura 30	Merge requests form service	87
Figura 31	Endpoints courses y teachers del academic service	89
Figura 32	Endpoints education cycles y generic users del academic service	89
Figura 33	Endpoints students y organizational units del academic service	90
Figura 34	Endpoints assessments del assessment service	91
Figura 35	Endpoints evaluation progress y locks del assessment service	91
Figura 36	Endpoints answers del answer service	92
Figura 37	Endpoints forms del form service	92
Figura 38	Vista principal del manager UI con menú de navegación	93
Figura 39	Vista de periodos académicos	94
Figura 40	Vista de cursos	94
Figura 41	Vista de docentes	95
Figura 42	Vista de estudiantes	95
Figura 43	Vista de unidades organizacionales	96
Figura 44	Vista de proceso evaluativo	96
Figura 45	Vista de gestión de formularios	97

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 46	Vista de evaluaciones disponibles para el estudiante	97
Figura 47	Vista del formulario de evaluación en proceso de diligenciamiento	98
Figura 48	Vista de omisión de evaluación con justificación.....	98
Figura 49	Environments academic service: develop success	99
Figura 50	Environments answer service: develop success	99
Figura 51	Environments assessment service: develop success.....	100
Figura 52	Environments form service: develop success.....	100
Figura 53	Environments manager UI: develop success.....	100
Figura 54	Environments catchment UI: develop success	101
Figura 55	Historial de pipelines academic service	103
Figura 56	Historial de pipelines answer service	103
Figura 57	Historial de pipelines assessment service.....	104
Figura 58	Historial de pipelines form service.....	104
Figura 59	Formulario implementado, sección entrada de datos	111
Figura 60	Formulario implementado, sección selección con opciones	111
Figura 61	Progreso de evaluaciones	112
Figura 62	Dashboard metabase EVADOC — resultados de evaluación	113
Figura 63	Gráfica histórica de code smells resueltos-academic	142
Figura 64	Detalle de issues-academic.....	142
Figura 65	Gráfica histórica de code smells resueltos-answer.....	143
Figura 66	Detalle de issues- answer	143
Figura 67	Gráfica histórica de code smells resueltos-assessment	144
Figura 68	Detalle de issues-assessment.....	144

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 69 Gráfica histórica de code smells resueltos-form	145
Figura 70 Detalle de issues-form	145
Figura 71 Métricas de pruebas unitarias	146
Figura 72 Quality gate academic service	146
Figura 73 Overall code academic service	147
Figura 74 Quality gate answer service.....	147
Figura 75 Overall code answer service.....	148
Figura 76 Quality gate assessment service	148
Figura 77 Overall Code assessment service	149
Figura 78 Quality gate form service	149
Figura 79 Overall code form service	150

Índice de Apéndices

Apéndice A: Historias de usuario	121
Apéndice B: Escenarios de atributos de calidad – ATAM	127
Apéndice C: Casos de prueba	129
Apéndice D: Evidencia de Calidad y Corrección de Errores.....	141

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Resumen

Título: Reingeniería del Software aplicada al Sistema de Evaluación Docente de la Universidad Industrial de Santander.

Autores: Jorge Julián García Hernández y Erika Yamile Lache Blanco

Palabras claves: Sistemas legados, Arquitectura de microservicios, Ingeniería inversa, Arquitectura dirigida por atributos de calidad.

Descripción: El sistema de evaluación docente de la Universidad Industrial de Santander (EVADOC), implementado desde 1996, presenta limitaciones estructurales asociadas a una arquitectura monolítica con alto acoplamiento, ausencia de seguimiento en tiempo real y baja interoperabilidad con los sistemas institucionales, afectando su mantenimiento, escalabilidad y evolución tecnológica. Para intervenir esta situación, el presente trabajo aplica una estrategia de reingeniería de software basada en el modelo de Roger S. Pressman, complementada con un enfoque de arquitectura dirigida, prototipado evolutivo y metodología Scrum, siguiendo los lineamientos de la Dirección de Tecnologías de Información y Comunicación (DTIC) de la UIS. Mediante ingeniería inversa se reconstruyó formalmente el funcionamiento del sistema actual, identificando sus principales limitaciones técnicas y justificando la decisión de realizar una reconstrucción arquitectónica. A partir de ello se definieron diez requerimientos no funcionales y veinte funcionales, evaluados y priorizados mediante el método ATAM. Posteriormente, se diseñó e implementó la Fase 1 del nuevo sistema bajo una arquitectura de microservicios, incorporando cuatro microservicios backend (Academic, Assessment, Answers y Forms) y dos interfaces de usuario (Manager UI y Catchment UI), desarrollados con Spring Boot, Angular y Oracle. Finalmente, la validación mediante catorce casos de prueba aprobados confirmó la viabilidad técnica y funcional de la solución propuesta para la modernización del sistema EVADOC.

* Trabajo de Grado

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Jathinson Meneses Mendoza.

Abstract

Title: Software Reengineering Applied to the Faculty Evaluation System of the Universidad Industrial de Santander.

Authors: Jorge Julián García Hernández and Erika Yamile Lache Blanco

Keywords: Legacy Systems, Microservices Architecture, Reverse Engineering, Quality Attribute-Driven Architecture.

Description: The teaching evaluation system of the Industrial University of Santander (EVADOC), implemented since 1996, presents structural limitations associated with a monolithic architecture with high coupling, lack of real-time monitoring, and low interoperability with institutional systems, affecting its maintenance, scalability, and technological evolution. To address this situation, this project applies a software reengineering strategy based on Roger S. Pressman's model, complemented by an architecture-driven approach, evolutionary prototyping, and the Scrum methodology, following the guidelines established by the Information and Communication Technologies Department (DTIC) of UIS. Through reverse engineering, the current system's operation was formally reconstructed, identifying its main technical limitations and justifying the decision to carry out an architectural reconstruction. Based on these findings, ten non-functional requirements and twenty functional requirements were defined, evaluated, and prioritized using the ATAM method. Subsequently, Phase 1 of the new system was designed and implemented under a microservices architecture, incorporating four backend microservices (Academic, Assessment, Answers, and Forms) and two user interfaces (Manager UI and Catchment UI), developed with Spring Boot, Angular, and Oracle technologies. Finally, validation through fourteen approved test cases confirmed the technical and functional feasibility of the proposed solution for the modernization of the EVADOC system.

* Bachelor Thesis

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Jathinson Meneses Mendoza.

Introducción

En el contexto actual de transformación digital y modernización de los procesos institucionales, las universidades enfrentan el reto de adaptar sus sistemas de información para garantizar una gestión eficiente, automatizada y orientada a la toma de decisiones basadas en datos. La Universidad Industrial de Santander (UIS), comprometida con la mejora continua de sus procesos académicos, ha identificado la necesidad de optimizar el sistema de evaluación docente, una herramienta clave para asegurar la calidad educativa y el desarrollo profesional del cuerpo docente.

El presente trabajo de grado, desarrollado bajo la modalidad de práctica empresarial, propone la aplicación de una estrategia de reingeniería de software al sistema de evaluación docente de la UIS. Esta estrategia busca rediseñar, modernizar y optimizar el sistema actual a partir de un diseño arquitectónico estructurado, basado en estándares actuales de desarrollo, metodologías ágiles y tecnologías como Angular y Spring Boot. El objetivo es mejorar la funcionalidad, usabilidad, interoperabilidad y capacidad de evolución del sistema frente a los desafíos institucionales futuros.

1 Generalidades

1.1 Planteamiento y Justificación del Problema

El sistema de evaluación docente de la UIS fue implementado en 1996 como mecanismo de autorregulación y mejora continua, bajo un entorno web que permitió consolidar información relevante para la toma de decisiones académicas. La Vicerrectoría Académica es la responsable de este proceso, reglamentado en el Acuerdo 63 de 1994 del Consejo Superior, específicamente en

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

los artículos 57 y 58, que establecen su finalidad, políticas, períodos, instrumentos y procedimientos.

Con el paso del tiempo, las exigencias institucionales, el crecimiento del cuerpo docente y los avances tecnológicos han evidenciado limitaciones significativas en el sistema actual:

- Ausencia de seguimiento en tiempo real: el sistema no permite consultar el avance de las encuestas durante el periodo de recolección, lo que impide una gestión proactiva del proceso.
- Procesamiento manual de datos: al finalizar cada periodo, los datos deben exportarse y procesarse manualmente para generar informes estadísticos, lo que implica un alto esfuerzo operativo y escasa automatización.
- Falta de integración institucional: el sistema no cuenta con mecanismos de interoperabilidad con otras plataformas institucionales, lo que genera duplicación de esfuerzos, fragmentación de la información y limitaciones en la calidad del análisis.
- Arquitectura tecnológica obsoleta: la arquitectura monolítica del sistema presenta alto acoplamiento, baja mantenibilidad y dificultades para integrarse con el ecosistema digital que la UIS está construyendo en su proceso de modernización tecnológica.

Ante esta situación, la reingeniería de software representa la estrategia más adecuada para intervenir el sistema. Este enfoque no implica únicamente un reemplazo tecnológico, sino un análisis técnico riguroso del sistema existente — identificando actores, flujos de información, restricciones normativas y criterios de calidad — como fundamento para una solución arquitectónica moderna, escalable e integrada. Desde una perspectiva académica, el proyecto integra áreas del conocimiento propias de la Ingeniería de Sistemas: auditoría de sistemas,

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

modelado de procesos, levantamiento de requisitos, diseño arquitectónico e implementación con metodologías ágiles.

La UIS se encuentra en un proceso activo de modernización de sus plataformas institucionales, lo que exige que todos los sistemas nuevos o intervenidos cumplan estándares de interoperabilidad, escalabilidad, seguridad y experiencia de usuario. Intervenir el sistema de evaluación docente bajo el enfoque de la reingeniería representa, por tanto, una oportunidad estratégica para alinearlo con la visión tecnológica institucional y garantizar su sostenibilidad a largo plazo.

1.2 Alcance

El presente proyecto aborda la reingeniería del sistema de evaluación docente de la Universidad Industrial de Santander desde un análisis global del sistema (que comprende la ingeniería inversa del proceso actual, la definición de requerimientos, el diseño de la arquitectura y la planificación de todas las funcionalidades) hasta la implementación y validación de la Fase 1, que constituye el alcance de entrega del presente trabajo.

La Fase 1 incluye el desarrollo de cuatro microservicios (Academic Service, Assessment Service, Answers Service y Forms Service) y dos interfaces de usuario (Manager UI y Catchment UI), que materializan ocho de las veinte historias de usuario definidas para el sistema:

- HU-01: Cargar información académica del periodo
- HU-03: Crear proceso de evaluación docente
- HU-04: Habilitar proceso de evaluación
- HU-05: Gestionar cierre parcial por unidad académica
- HU-07: Consultar profesores pendientes por evaluar

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- HU-08: Diligenciar formulario de evaluación docente
- HU-09: Omitir evaluación con justificación
- HU-20: Gestionar formularios dinámicos

Quedan fuera del alcance de la Fase 1 las doce historias de usuario restantes (HU-02, HU-06, HU-10 a HU-19) y los casos de prueba asociados a funcionalidades de fases posteriores. Estas líneas de trabajo se describen en el Capítulo 6 Trabajo Futuro.

2 Objetivos

2.1 Objetivo General

Diseñar e implementar una solución tecnológica para optimizar el sistema de evaluación docente de la Universidad Industrial de Santander, mediante la aplicación de reingeniería de software, asegurando su viabilidad técnica, funcional, su integración con otros sistemas institucionales y su adaptabilidad a futuros requerimientos tecnológicos.

2.2 Objetivos Específicos

1. Diagnosticar el sistema actual de Evaluación Docente de la Universidad Industrial de Santander, identificando sus limitaciones técnicas, funcionales y de integración, como base para definir el alcance del proceso de reingeniería.
2. Analizar el estado del arte en metodologías y técnicas de reingeniería de software, identificando los conceptos, modelos y métodos existentes que puedan aplicarse al sistema de evaluación docente
3. Definir los requisitos funcionales y no funcionales del sistema Evaluación Docente, modelando sus procesos, de los flujos de evaluación y la gestión de encuestas, para garantizar su alineación con las necesidades actuales de la Universidad y su integración con otros sistemas institucionales.
4. Diseñar la arquitectura tecnológica del sistema de Evaluación Docente, abarcando el desarrollo del frontend, backend y la base de datos, asegurando el cumplimiento de los estándares establecidos por la DTIC y su compatibilidad con los sistemas institucionales existentes.
5. Implementar los componentes del Sistema de Evaluación Docente, definido como una plataforma integral para la gestión de instrumentos de evaluación, que abarque la creación

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

y configuración de formularios dinámicos, la captura y el análisis de respuestas, así como la generación de informes y visualizaciones de resultados, utilizando las tecnologías seleccionadas y cumpliendo con los lineamientos y estándares establecidos por la DTIC.

6. Realizar pruebas funcionales, de integración y usabilidad del gestor de instrumentos de evaluación, aplicándolas con una muestra de la población estudiantil y utilizando tanto el instrumento actual como el nuevo propuestos, con el fin de validar los requerimientos del producto.

3 Marco teórico

El presente capítulo establece el marco conceptual y tecnológico que fundamenta el desarrollo del proyecto. Se presentan los conceptos teóricos de reingeniería de software y los modelos metodológicos analizados, el enfoque de arquitectura dirigida y el método ATAM utilizado para la evaluación arquitectónica, el proceso de evaluación docente como dominio del problema, las metodologías de desarrollo adoptadas y las tecnologías seleccionadas para la construcción del nuevo sistema.

3.1 Reingeniería de Software

La reingeniería de software se define como una disciplina dentro de la ingeniería de sistemas que se ocupa del análisis, rediseño y transformación de sistemas existentes con el fin de mejorar su rendimiento, mantenibilidad, escalabilidad y alineación con los objetivos organizacionales. Este enfoque surge como respuesta a la necesidad de actualizar aplicaciones que, a pesar de ser funcionales en su momento, se han vuelto obsoletas frente a las nuevas exigencias tecnológicas y operativas.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.1.1 Concepto de Reingeniería

La reingeniería de software es una disciplina orientada al análisis y transformación de sistemas existentes para mejorar su calidad estructural, mantenibilidad y alineación con los objetivos organizacionales, sin partir desde cero.

Pressman la define como el proceso de examinar y alterar un sistema existente para reconstruirlo preservando su funcionalidad esencial mientras se optimizan sus componentes internos. Por su parte, Sommerville la describe como un conjunto de actividades para mejorar un sistema sin modificar su comportamiento externo, actuando sobre su estructura interna y arquitectura.

En entornos institucionales, la reingeniería resulta especialmente relevante cuando los sistemas legacy presentan arquitecturas obsoletas que dificultan su integración con plataformas modernas, buscando garantizar su escalabilidad y evolución futura.

3.1.2 Objetivos de la Reingeniería

La reingeniería de software busca mejorar la calidad de los sistemas existentes, prolongar su vida útil y facilitar su mantenimiento y evolución, sin perder el conocimiento funcional acumulado. Pressman (s.f.) destaca como objetivo central reducir la complejidad y la deuda técnica de los sistemas legacy, recuperando el control sobre su estructura mediante análisis y rediseño sistemático. Sus principales objetivos son:

- Mejorar la mantenibilidad y comprensión del software.
- Optimizar la arquitectura promoviendo modularidad y buenas prácticas de diseño.
- Reducir la dependencia de tecnologías obsoletas.
- Incrementar la escalabilidad e integración con otros sistemas.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- Preservar el conocimiento funcional de procesos institucionales consolidados.

En instituciones de educación superior, estos objetivos son especialmente relevantes, ya que permiten alinear los sistemas con estándares tecnológicos actuales y fortalecer la toma de decisiones basada en datos.

3.1.3 Fases de la Reingeniería

La reingeniería de software se desarrolla en fases estructuradas que permiten analizar, transformar y modernizar un sistema existente de forma controlada. Pressman y Sommerville coinciden en que estas fases pueden ejecutarse de forma iterativa según el alcance del proyecto.

3.1.3.1 Inventario y Evaluación del Sistema. Consiste en identificar los componentes del sistema actual, su estado, nivel de documentación y relevancia organizacional. Su objetivo es determinar la viabilidad de intervenir el sistema y definir si conviene reutilizar, reestructurar o reconstruir sus componentes.

3.1.3.2 Ingeniería Inversa. Implica analizar el sistema para comprender su arquitectura, flujos de información y comportamiento funcional, generando representaciones de mayor abstracción (diagramas de procesos, modelos de datos, casos de uso). Es fundamental cuando la documentación original es insuficiente o inexistente.

3.1.3.3 Reestructuración. Reorganiza la estructura interna del sistema sin alterar su funcionalidad externa, mejorando su claridad, modularidad y mantenibilidad. Cuando las limitaciones son profundas (obsolescencia tecnológica o arquitectura inadecuada), puede derivar en una reconstrucción parcial o total.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.1.3.4 Reconstrucción o Rediseño. Define una nueva arquitectura alineada con estándares actuales y necesidades organizacionales, pudiendo implicar migración tecnológica y redefinición de componentes funcionales para lograr mayor escalabilidad, interoperabilidad y seguridad.

3.1.3.5 Validación y Pruebas. El sistema reingenierizado se somete a pruebas funcionales, de integración, rendimiento y usabilidad para garantizar el cumplimiento de requisitos y la preservación de la funcionalidad esencial.

3.1.4 Modelos de Reingeniería de Software

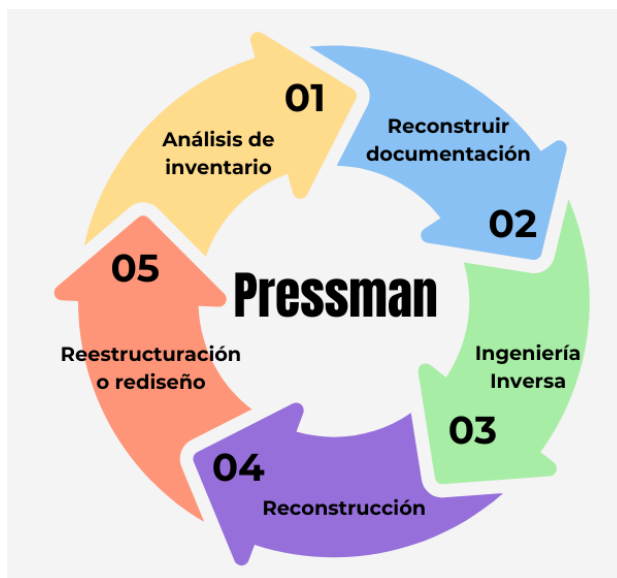
La reingeniería de software se aborda desde distintos modelos metodológicos que estructuran el proceso de análisis, transformación y modernización de sistemas existentes. La selección del modelo más adecuado depende del estado del sistema legacy, el contexto institucional y los objetivos del proyecto.

3.1.4.1 Modelo de Reingeniería de Roger S. Pressman. Pressman propone un proceso estructurado que parte de un diagnóstico técnico detallado antes de cualquier intervención. Sus actividades principales son: inventario del software, ingeniería inversa, reestructuración y reconstrucción o rediseño. Este modelo es especialmente pertinente en entornos institucionales donde los sistemas legacy son aún funcionales, pero presentan limitaciones estructurales, ya que permite una transición controlada preservando el conocimiento funcional acumulado.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 1

Modelo de reingeniería de software según Pressman.



Nota: Modelo de reingeniería propuesto por Roger S. Pressman (2014).

3.1.4.2 Modelo de Ian Sommerville. Sommerville concibe la reingeniería como una estrategia para prolongar la vida útil del software mejorando su estructura interna sin modificar su comportamiento externo. Sus actividades incluyen: análisis del sistema, reestructuración del código, reestructuración de datos y redocumentación. Este enfoque es adecuado cuando la arquitectura es relativamente estable y la tecnología subyacente sigue siendo compatible con los estándares organizacionales.

3.1.4.3 Modelo de “Big Bang Replacement” (Reemplazo total). Propone la sustitución completa del sistema existente por una nueva solución desarrollada desde cero, con migración total de datos y eliminación de dependencias obsoletas. Aunque elimina la deuda técnica acumulada, implica riesgos operativos significativos durante la transición. Es adecuado cuando el sistema presenta obsolescencia severa que hace inviable su reestructuración.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.1.4.4 Modelo de Reingeniería Basada en Arquitectura (Architecture-Driven Reengineering). Centra la transformación en el rediseño de la arquitectura del sistema, migrando desde estructuras monolíticas hacia arquitecturas desacopladas y modulares (microservicios, SOA, arquitectura en capas). Sus etapas son: evaluación de la arquitectura existente, definición de la arquitectura objetivo, plan de migración e implementación con validación. Es pertinente cuando el problema principal radica en la estructura arquitectónica y no únicamente en el código.

Figura 2

Modelo de reingeniería de software basada en la arquitectura.



Nota: Modernización arquitectónica y migración de sistemas legacy.

3.1.4.5 Modelo de Reingeniería Orientada a Componentes. Promueve una transformación progresiva mediante la intervención selectiva de módulos del sistema. Sus actividades incluyen: identificación de componentes, evaluación de calidad, reestructuración o sustitución selectiva, e integración y validación. Permite conservar los componentes de calidad

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

aceptable y rediseñar solo los deficientes, reduciendo costos y riesgos frente a una reconstrucción total. Su efectividad se limita en sistemas con arquitecturas altamente acopladas o monolíticas.

3.1.5 Comparación de Modelos

La selección del modelo de reingeniería adecuado depende del nivel de deterioro del sistema, su arquitectura y los objetivos estratégicos de la organización. La siguiente tabla resume las características principales de los modelos descritos:

Tabla 1

Comparación de Modelos de Reingeniería de Software.

Modelo	En que consiste	Ventajas	Desventajas	Pertinencia para el sistema UIS
Pressman	Análisis, ingeniería inversa, reestructuración y reconstrucción del sistema.	Metodología estructurada y académicamente sólida.	Requiere mayor tiempo y esfuerzo inicial.	Alta. Permite diagnóstico profundo y reconstrucción arquitectónica.
Sommerville	Evolución progresiva mediante refactorización y transformación incremental.	Menor riesgo e impacto operativo.	No es ideal cuando la arquitectura es obsoleta.	Media. No es el más adecuado si se decide reconstrucción total.
Reemplazo (Big Bang)	Sistema completamente nuevo desplegado en un solo momento.	Solución limpia, arquitectura moderna desde cero.	Alto riesgo si no hay buen análisis previo.	Media-Alta. Aplicable pero carece de sustento metodológico formal.
Basada en Arquitectura	Rediseño de la arquitectura hacia modelos modernos (microservicios, capas, hexagonal).	Alta escalabilidad, alineación con estándares modernos.	Requiere rediseño completo y experiencia técnica.	Alta. Se ajusta a la arquitectura propuesta para el sistema.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Orientado a Componentes	Identificación y reemplazo selectivo de módulos críticos.	Reduce costos si existen componentes estables	No viable si el sistema está altamente acoplado.	Baja. El sistema presenta limitaciones estructurales significativas.
--------------------------------	---	---	--	--

Nota: Literatura de ingeniería de software, aplicado al contexto del sistema de Evaluación Docente de la UIS.

El análisis evidencia que los modelos conservadores son adecuados cuando la arquitectura es relativamente estable, mientras que enfoques como Pressman o la reingeniería basada en arquitectura son más pertinentes ante obsolescencia tecnológica significativa. La decisión final debe equilibrar el nivel de intervención requerido con los riesgos operativos y consideraciones de sostenibilidad institucional a largo plazo.

3.1.6 Modelo Seleccionado para el Proyecto y Justificación

A partir del análisis comparativo, se determinó que el enfoque más adecuado para el proyecto es el modelo de Roger S. Pressman, complementado con reingeniería basada en arquitectura.

Esta selección responde a las características del sistema de evaluación docente de la UIS, un sistema legacy con más de dos décadas de operación que, aunque funcional, presenta limitaciones arquitectónicas, dificultad de integración con otros sistemas institucionales, dependencia de procesos manuales y escasa escalabilidad.

Los demás modelos fueron descartados por las siguientes razones: Sommerville resultó insuficiente al centrarse en reestructuración interna sin atender las limitaciones estructurales profundas; Big Bang Replacement, aunque viable, carece de sustento metodológico formal que

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

requiere el proyecto; y el modelo orientado a componentes no es aplicable dado el alto acoplamiento de la arquitectura existente.

Se adopta Pressman por su énfasis en ingeniería inversa, diagnóstico técnico riguroso y toma de decisiones fundamentada entre reestructuración o reconstrucción. Se complementa con reingeniería basada en arquitectura, dado que el principal desafío identificado es estructural, permitiendo alinear la solución con estándares modernos, promover la modularidad y garantizar la interoperabilidad institucional a largo plazo.

3.1.7 Criterios para Decidir entre Reutilización y Reconstrucción

La decisión de reutilizar, reestructurar o reconstruir un sistema debe basarse en un análisis técnico riguroso, no en preferencias tecnológicas. Los factores determinantes son: nivel de deuda técnica, obsolescencia tecnológica, falta de documentación, bajo nivel de modularidad, dificultad de integración y riesgos de seguridad. Cuando estos factores afectan significativamente la mantenibilidad y escalabilidad del sistema, la reconstrucción puede resultar más viable que la reutilización parcial. Los criterios clave para esta evaluación son:

- **Código legacy:** Código desarrollado bajo tecnologías o arquitecturas obsoletas que acumula alta complejidad, dependencias rígidas y escasa claridad estructural, dificultando su modificación y evolución.
- **Deuda técnica:** Costo futuro derivado de decisiones que priorizaron soluciones rápidas sobre estructuras sostenibles. Cuando alcanza niveles elevados, el esfuerzo de mantener el sistema puede superar el costo de reconstruirlo.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- **Arquitectura obsoleta:** Limita la integración con nuevas plataformas y la escalabilidad. Las arquitecturas monolíticas altamente acopladas requieren rediseño estructural completo, no solo refactorización.
- **Falta de documentación:** Incrementa la dependencia del conocimiento tácito y eleva el riesgo de errores. En estos casos, la ingeniería inversa se convierte en una fase obligatoria antes de cualquier intervención.
- **Baja mantenibilidad:** Alto acoplamiento y baja modularidad hacen que cada modificación implique riesgo de afectar funcionalidades existentes, reduciendo la agilidad organizacional.
- **No alineación con estándares institucionales (DTIC):** Cuando el sistema no cumple los lineamientos institucionales en arquitectura, seguridad o interoperabilidad, la reconstrucción se convierte en una necesidad estratégica.

En escenarios donde predominan la obsolescencia, la arquitectura rígida y la deuda técnica acumulada, la reconstrucción arquitectónica controlada es la alternativa que mejor garantiza la sostenibilidad e integración futura del sistema.

3.2 Enfoque Arquitectura Dirigida

El enfoque arquitectura-dirigida (Architecture Driven Development) establece la arquitectura del sistema como eje central que orienta todas las decisiones de diseño, implementación y evolución del software, definiéndola explícitamente antes de su construcción para garantizar coherencia técnica y alineación con atributos de calidad previamente establecidos.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

La arquitectura de software se entiende como la estructura organizacional del sistema sus componentes, relaciones, patrones y principios de interacción que condiciona aspectos críticos como escalabilidad, mantenibilidad, interoperabilidad, seguridad y rendimiento.

En procesos de reingeniería, este enfoque es especialmente relevante cuando el sistema presenta limitaciones estructurales que no pueden resolverse mediante refactorización. Dentro del contexto institucional implica definir una arquitectura objetivo antes de la reconstrucción, establecer separación clara de responsabilidades, reducir el acoplamiento entre componentes, facilitar la integración con otros sistemas y alinear la solución con los lineamientos de la DTIC.

En el presente proyecto, el enfoque arquitectura-dirigida actúa como complemento estratégico del modelo de Pressman, permitiendo que las fases de ingeniería inversa, análisis y reconstrucción se desarrollen bajo una visión arquitectónica claramente definida, garantizando que la solución propuesta pueda adaptarse a futuros requerimientos institucionales sin comprometer su estabilidad estructural.

3.2.1 ATAM – Metodo de Análisis de Trade-Offs

El ATAM (Architecture Tradeoff Analysis Method) es un método de evaluación arquitectónica que permite determinar en qué medida una arquitectura de software satisface los atributos de calidad del sistema y qué compromisos (trade-offs) implica cada decisión de diseño. El método se estructura en cuatro componentes principales:

- **Árbol de utilidad:** organiza y prioriza los atributos de calidad del sistema según su importancia para el negocio y su dificultad técnica.
- **Escenarios de atributos de calidad:** describen situaciones concretas que el sistema debe manejar correctamente para cumplir cada atributo de calidad.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- Análisis de trade-offs: identifica qué beneficios y sacrificios implica cada decisión arquitectónica frente a los atributos de calidad definidos.
- Puntos de sensibilidad y riesgo: identifican qué decisiones arquitectónicas tienen mayor impacto sobre los atributos de calidad prioritarios.

El ATAM es especialmente útil en proyectos de reingeniería donde se deben justificar formalmente las decisiones de diseño frente a restricciones técnicas, institucionales o normativas.

3.3 Evaluación Docente

La evaluación del desempeño docente constituye una herramienta fundamental en las instituciones de educación superior para garantizar la calidad del proceso educativo, promover la mejora continua y respaldar la toma de decisiones académicas. Su propósito es valorar el cumplimiento de las funciones docentes, identificar fortalezas y debilidades en la práctica pedagógica y fomentar el desarrollo profesional del profesor. En la UIS, este proceso está reglamentado por el Acuerdo 63 de 1994, que establece los lineamientos, períodos, instrumentos y procedimientos asociados a la evaluación institucional del cuerpo docente.

3.4 Métodos de Evaluación

Existen múltiples enfoques para evaluar el desempeño de los profesores, los cuales pueden combinarse para obtener una visión más completa y objetiva del proceso. Entre los más utilizados se encuentran:

3.4.1 Encuesta a Estudiantes

Permiten conocer la percepción de los estudiantes sobre la metodología, la claridad, el compromiso y la actitud del docente en el aula. Es el método principal implementado en el sistema EVADOC de la UIS.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.4.2 Autoevaluación

Invita al docente a reflexionar críticamente sobre su propia práctica, identificando logros y áreas de mejora. El sistema EVADOC contempla este mecanismo en su diseño como parte de las épicas funcionales a implementar en fases posteriores.

3.4.3 Evaluación por Directivos

Se basa en criterios institucionales relacionados con el cumplimiento de funciones, participación en actividades institucionales y desarrollo profesional del docente.

3.5 Metodología de Desarrollo

El proyecto adoptó dos metodologías complementarias: el prototipado evolutivo como enfoque general y Scrum bajo los lineamientos de la DTIC para la fase de implementación. Su combinación permitió abordar la complejidad del proyecto desde la fase de análisis hasta la entrega del prototipo funcional de la Fase 1.

3.5.1 Prototipado Evolutivo

El prototipado evolutivo es una metodología de desarrollo de software en la que el sistema se construye de forma incremental mediante prototipos que son evaluados y refinados progresivamente. A diferencia del prototipado desechable, en el modelo evolutivo los prototipos no se descartan: cada iteración incorpora los cambios derivados de la retroalimentación obtenida, hasta consolidar la solución definitiva. Este enfoque resulta especialmente pertinente en proyectos con requerimientos inicialmente ambiguos o incompletos, donde la comprensión del sistema se construye de forma iterativa a medida que avanza el análisis. En el presente proyecto, el prototipado evolutivo orientó las etapas de análisis, levantamiento de requerimientos y diseño,

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

permitiendo refinar progresivamente la comprensión del sistema de evaluación docente existente y la propuesta de solución.

3.5.2 Scrum

Scrum es un marco de trabajo ágil para la gestión y desarrollo de software complejo, basado en ciclos iterativos denominados sprints, cuya duración típica oscila entre una y cuatro semanas. Al finalizar cada sprint se entrega un incremento funcional del producto, permitiendo inspeccionar el avance y adaptar el plan según los resultados obtenidos. Scrum define tres roles principales: Product Owner (responsable del backlog y la priorización), Scrum Master (facilitador del proceso) y Development Team (equipo de desarrollo autoorganizado). Sus eventos clave son: Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective. En el presente proyecto, Scrum se adoptó bajo los lineamientos de la DTIC de la UIS para estructurar la fase de implementación del nuevo sistema, organizando el trabajo de desarrollo en sprints y garantizando la entrega continua de valor funcional.

3.6 Arquitectura de Software

La arquitectura de software define la estructura organizacional de un sistema: sus componentes, relaciones, patrones de interacción y principios de diseño. Las decisiones arquitectónicas condicionan atributos críticos del sistema como escalabilidad, mantenibilidad, interoperabilidad, seguridad y rendimiento. En el presente proyecto, la arquitectura del nuevo sistema EVADOC se sustenta en tres pilares: la organización basada en microservicios, el patrón de arquitectura hexagonal y los principios SOLID de diseño orientado a objetos.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

3.6.1 Arquitectura Orientada a Microservicios

Es un enfoque de diseño de software que organiza una aplicación como un conjunto de servicios pequeños e independientes, cada uno responsable de una funcionalidad específica y comunicándose con los demás a través de interfaces bien definidas, comúnmente mediante APIs REST o mecanismos de mensajería. Este tipo de arquitectura ofrece ventajas significativas frente a los modelos monolíticos tradicionales: mayor escalabilidad (cada servicio puede escalar de forma independiente), facilidad de mantenimiento (los cambios en un servicio no afectan al sistema en su conjunto) y tolerancia a fallos (el fallo de un servicio no compromete el funcionamiento global). En el sistema EVADOC, la arquitectura de microservicios resuelve las principales limitaciones del sistema legacy, que presentaba alto acoplamiento y baja capacidad de integración con otros sistemas institucionales.

3.6.2 Arquitectura Hexagonal

Conocida también como arquitectura de puertos y adaptadores, su propósito es separar la lógica central de la aplicación (dominio) de los elementos externos con los que interactúa, como bases de datos, interfaces de usuario o servicios externos. Esta separación se logra mediante puertos, que definen las interfaces del dominio, y adaptadores, que implementan la conexión con el entorno exterior. Gracias a esta estructura, los componentes externos pueden integrarse o reemplazarse sin afectar el funcionamiento interno del sistema, facilitando el mantenimiento, las pruebas automatizadas y la evolución independiente del núcleo de negocio. En el sistema EVADOC, la arquitectura hexagonal se aplica internamente en cada microservicio, garantizando la separación entre la lógica de negocio y las tecnologías de persistencia o comunicación utilizadas.

3.6.3 Principios SOLID

Los principios SOLID constituyen un conjunto de cinco directrices de diseño orientado a objetos formuladas por Robert C. Martin, que promueven la construcción de sistemas mantenibles, extensibles y robustos:

- S - Responsabilidad Única (SRP): cada clase o componente debe tener una única razón para cambiar, lo que promueve la cohesión y reduce el acoplamiento.
- O - Abierto/Cerrado (OCP): las entidades de software deben estar abiertas para extensión, pero cerradas para modificación, permitiendo añadir funcionalidad sin alterar el código existente.
- L - Sustitución de Liskov (LSP): los objetos de una clase derivada deben poder sustituir a los de su clase base sin alterar el comportamiento esperado del sistema.
- I - Segregación de Interfaces (ISP): es preferible tener varias interfaces específicas que una interfaz general de múltiples propósitos, evitando que los clientes dependan de métodos que no utilizan.
- D - Inversión de Dependencias (DIP): los módulos de alto nivel no deben depender de módulos de bajo nivel; ambos deben depender de abstracciones.

En el sistema EVADOC, estos principios orientaron el diseño interno de cada microservicio, garantizando la cohesión funcional, el bajo acoplamiento entre componentes y la facilidad de extensión para incorporar nuevos requerimientos en fases futuras del proyecto.

3.7 Tecnologías de Desarrollo

3.7.1 *Angular (Frontend)*

Angular es un framework de desarrollo web de código abierto, respaldado por Google, utilizado para la construcción de aplicaciones de una sola página (SPA). Está desarrollado sobre TypeScript, que aporta robustez mediante tipado estático y herramientas avanzadas de desarrollo. Angular organiza la aplicación en módulos, componentes y servicios, promoviendo la separación de responsabilidades y la reutilización de código. En el sistema EVADOC, Angular fue seleccionado por los lineamientos de la DTIC para el desarrollo de las interfaces Manager UI y Catchment UI.

3.7.2 *Spring Boot (Backend)*

Spring Boot es un framework de desarrollo backend perteneciente al ecosistema Spring (Java), diseñado para facilitar la construcción de aplicaciones web y servicios RESTful con configuración mínima. Ofrece una estructura que permite desarrollar soluciones robustas, seguras y escalables mediante el uso de "starters" que integran automáticamente dependencias comunes. Su compatibilidad con arquitecturas de microservicios y su amplia adopción en entornos corporativos lo convierten en una herramienta idónea para el desarrollo del backend del sistema EVADOC. Fue seleccionado bajo los lineamientos tecnológicos de la DTIC.

3.8 Tecnologías de Persistencia

3.8.1 *Oracle*

Oracle es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation, ampliamente utilizado en entornos empresariales por su alto rendimiento, escalabilidad y capacidad para gestionar transacciones complejas de manera segura. Funciona

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

mediante el lenguaje SQL y ofrece funcionalidades avanzadas como replicación de información, respaldo automatizado, recuperación ante fallos y soporte para entornos en la nube. En el sistema EVADOC, Oracle fue seleccionado como tecnología de persistencia por contar con soporte institucional, licenciamiento disponible y herramientas de administración en la DTIC de la UIS.

3.8.2 MongoDB

MongoDB es un sistema de gestión de bases de datos NoSQL, orientado al manejo de documentos en formato BSON (variante de JSON), que ofrece un esquema flexible y escalabilidad horizontal eficiente. Resulta especialmente adecuado para aplicaciones con datos no estructurados o estructuras variables, como formularios dinámicos. En el contexto del proyecto, MongoDB fue evaluado como alternativa técnica para el almacenamiento de formularios de evaluación; sin embargo, fue descartado al no contar con soporte institucional en la DTIC. El análisis comparativo detallado se presenta en la sección 3.4.

3.9 Tecnologías de Integración y Despliegue

3.9.1 RabbitMQ

RabbitMQ es un broker de mensajería de código abierto que implementa el protocolo AMQP (Advanced Message Queuing Protocol), utilizado para la comunicación asíncrona entre componentes de software distribuido. Funciona mediante un modelo de colas: los productores envían mensajes a un exchange, que los enruta hacia una o más colas según reglas de binding, donde los consumidores los procesan de forma independiente. Este mecanismo permite el desacoplamiento entre servicios, garantizando que el emisor no dependa de la disponibilidad inmediata del receptor. En el sistema EVADOC, RabbitMQ gestiona la comunicación asíncrona bidireccional entre el Assessment Service y el Academic Service y en un solo sentido entre el

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Answers Service y Assessment Service, asegurando la entrega confiable de mensajes sin generar dependencias directas entre microservicios. Fue seleccionado sobre Apache Kafka por contar con soporte institucional disponible en la DTIC (ver sección 3.4).

3.9.2 *Docker*

Docker es una plataforma de código abierto para el desarrollo, despliegue y ejecución de aplicaciones mediante contenedores. Un contenedor encapsula una aplicación junto con todas sus dependencias en una unidad portátil y reproducible, garantizando que el software se ejecute de forma idéntica en cualquier entorno. A diferencia de las máquinas virtuales, los contenedores comparten el núcleo del sistema operativo anfitrión, lo que los hace más ligeros y eficientes en el uso de recursos. En el sistema EVADOC, Docker se utiliza para empaquetar y desplegar cada microservicio de forma independiente, facilitando la orquestación de servicios y la alineación con la infraestructura institucional de la DTIC.

4 Metodología

El presente proyecto se desarrolló bajo un enfoque metodológico aplicado, con orientación práctica y fundamentado en principios de la ingeniería de software y la reingeniería de sistemas legacy. Para su desarrollo se adoptaron dos metodologías complementarias: el prototipado evolutivo como enfoque general del proyecto, y Scrum bajo los lineamientos de la División de Tecnologías de Información y Comunicaciones (DTIC) para la fase de implementación. Como base de análisis se adoptó el modelo de reingeniería propuesto por Roger S. Pressman, complementado con un enfoque de arquitectura dirigida.

Desde el punto de vista investigativo, el proyecto puede clasificarse como un estudio de tipo descriptivo y propositivo: descriptivo porque analiza el estado actual del sistema de evaluación

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

docente identificando sus debilidades técnicas y estructurales; propositivo porque plantea una solución arquitectónica moderna que responde a las necesidades institucionales actuales y futuras. De esta manera, el enfoque metodológico adoptado garantiza coherencia entre el sustento teórico presentado en el Capítulo 2 y la aplicación práctica desarrollada en el presente trabajo.

4.1 Modelo de Reingeniería Adoptado

El proyecto adopta el modelo de reingeniería de Roger S. Pressman, que plantea un proceso estructurado basado en ingeniería inversa, reestructuración, redocumentación y reconstrucción del sistema. Su elección se justifica por las características del sistema legacy de evaluación docente de la UIS (más de dos décadas de operación, con limitaciones de mantenibilidad, escalabilidad e integración) que requieren un análisis técnico riguroso antes de definir la estrategia de intervención. Tras el diagnóstico realizado, se determinó que la reconstrucción arquitectónica era más viable que la reutilización del código existente, dada su obsolescencia tecnológica y no alineación con los estándares de la DTIC.

4.2 Aplicación del Modelo de Reingeniería en el Sistema EVADOC.

La aplicación del modelo de Pressman, articulada con un enfoque arquitectura-dirigida, se organizó en cuatro etapas para analizar el sistema legacy, diagnosticar sus limitaciones y definir una estrategia de reconstrucción alineada con los estándares institucionales.

4.2.1 Ingeniería Inversa Aplicada

Se analizaron los módulos funcionales, la base de datos, los procesos institucionales y la documentación técnica disponible. Mediante modelado BPMN y levantamiento de requerimientos implícitos se recuperó el conocimiento funcional del sistema, identificando componentes críticos y limitaciones arquitectónicas.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

4.2.2 *Análisis y Diagnóstico del Sistema*

El diagnóstico técnico evidenció: tecnologías desactualizadas, arquitectura monolítica con alto acoplamiento, baja modularidad, dificultad de integración con otros sistemas institucionales y dependencia de procesos manuales. La magnitud de la obsolescencia tecnológica hizo inviable una simple reestructuración, determinando la reconstrucción arquitectónica controlada como estrategia.

4.2.3 *Rediseño Arquitectónico*

Se definió una arquitectura objetivo basada en separación de responsabilidades (presentación, lógica de negocio y persistencia), modularidad, bajo acoplamiento, exposición de servicios mediante API e integración con otros sistemas institucionales, cumpliendo los estándares de la DTIC.

4.2.4 *Reconstrucción e Implementación*

La reconstrucción se gestionó con metodología ágil Scrum en sprints iterativos, desarrollando nuevos módulos funcionales, una base de datos optimizada y mecanismos de seguridad y control de acceso. El proceso concluyó con pruebas de validación técnica y funcional que garantizaron el correcto funcionamiento del sistema.

4.3 *Enfoque Arquitectura Dirigida*

De manera transversal al modelo de Pressman, el proyecto se desarrolló bajo un enfoque arquitectura dirigida, en el cual la estructura del sistema se definió previamente como marco orientador del diseño e implementación. Se adoptó una arquitectura basada en microservicios, principios de arquitectura hexagonal, comunicación mediante API REST y separación clara entre presentación, lógica de negocio y persistencia. Este enfoque garantiza mayor mantenibilidad,

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

escalabilidad horizontal, facilidad de integración con el ecosistema institucional y cumplimiento de los lineamientos de la DTIC, asegurando la sostenibilidad técnica del sistema a largo plazo.

4.4 Alternativas Tecnológicas Consideradas

Para el diseño del sistema EVADOC, se evaluaron distintas alternativas tecnológicas en tres componentes clave la base de datos, el sistema de mensajería y el estilo arquitectónico. La selección final estuvo determinada por una combinación de criterios técnicos y lineamientos institucionales establecidos por la División de Tecnologías de Información y comunicaciones (DTIC) de la UIS.

Tabla 2

Alternativa 1: Base de Datos Mongo vs Oracle

Criterio	MongoDB	Oracle relacional
Modelo de datos	Documental (flexible, sin esquema fijo)	Relacional (esquema estructurado)
Escalabilidad horizontal	Alta	Media
Flexibilidad para formularios dinámicos	Alta - ideal para estructuras variables	Media - requiere diseño previo del esquema
Integridad referencial	Limitada	Alta
Soporte institucional DTIC	No disponible	Disponible
Herramientas de administración institucional	No disponible	Disponible

Nota: Análisis de características técnicas de MongoDB y Oracle en el contexto del sistema de evaluación docente de la UIS.

MongoDB fue la alternativa técnicamente propuesta por el equipo debido a su flexibilidad para almacenar formularios de evaluación con estructura variable y su capacidad de escalabilidad horizontal. Sin embargo, al no contar con soporte ni licenciamiento disponible en la DTIC, se

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

adoptó Oracle relacional como tecnología de base de datos, garantizando así el respaldo institucional, la integridad de los datos y la alineación con la infraestructura tecnológica existente en la UIS.

Tabla 3

Alternativa 2: Sistema de Mensajería Kafka vs RabbitMQ

Criterio	Apache Kafka	RabbitMQ
Modelo de mensajería	Log distribuido (alta retención)	Cola de mensajes (broker tradicional)
rendimiento en alto volumen	Muy alto	Alto
Persistencia de mensajes	Alta - ideal para auditoría	Media
Complejidad y configuración	Alta	Media
Soporte institucional DTIC	No disponible	Disponible
Casos de uso ideales	Streaming de eventos, Big Data	Comunicación asíncrona entre servicios

Nota: Análisis de características de Apache Kafka y RabbitMQ, considerando criterios de rendimiento, persistencia y soporte institucional para su aplicación en el sistema de evaluación docente de la UIS.

Apache Kafka fue considerado como la opción técnicamente superior para la comunicación asíncrona entre microservicios, especialmente por su capacidad de retención de mensajes y su rendimiento bajo cargas altas. No obstante, al igual que MongoDB, Kafka no contaba con soporte disponible en la DTIC, por lo que se adoptó RabbitMQ como broker de mensajería, el cual cumple con los requerimientos de comunicación asíncrona del sistema y cuenta con respaldo institucional.

Tabla 4*Alternativo 3: Estilo Arquitectónico Monolítico vs Microservicios*

Criterio	Arquitectura Monolítica	Arquitectura de Microservicios
Mantenibilidad	Baja - cambios afectan todo el sistema	Alta - cambios aislados por servicio
Escalabilidad	Limitada	Alta - escala por servicios independiente
Complejidad de despliegue	Baja	Media - requiere orquestación (Docker)
Tolerancia a fallos	Baja - fallo total entra un error	Alta - fallo aislado por servicio
Alineación con RNF del sistema	No cumple mantenibilidad ni escalabilidad	Cumple todos los RNF prioritarios

Nota: Análisis comparativo entre arquitectura monolítica y arquitectura de microservicios, considerando criterios de mantenibilidad, escalabilidad y alineación con los requerimientos no funcionales del sistema.

Dado que el sistema legacy de EVADOC presentaba precisamente los problemas asociados a una arquitectura monolítica (alto acoplamiento, baja mantenibilidad, dificultad de integración), se adoptó una arquitectura de microservicios como estrategia de modernización, alineada con los objetivos de calidad definidos en el ATAM.

4.5 Fases de Desarrollo del Nuevo Sistema (Scrum DTIC)

La fase de reconstrucción del sistema EVADOC se ejecutó bajo la metodología Scrum, adaptada a los lineamientos establecidos por la División de Tecnologías de Información y Comunicaciones (DTIC) de la UIS. Este marco ágil estructuró el trabajo de desarrollo en ciclos iterativos e incrementales, priorizando la entrega de valor funcional verificable al finalizar cada

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

sprint y garantizando el cumplimiento de los estándares tecnológicos institucionales a lo largo del proceso.

4.5.1 Levantamiento y Priorización de Requerimientos

A partir de los hallazgos de la ingeniería inversa y el diagnóstico técnico del sistema legacy, se construyó el product backlog del sistema EVADOC, conformado por veinte historias de usuario organizadas en siete épicas funcionales. La priorización se realizó considerando el impacto funcional de cada historia, su dependencia con otros componentes del sistema y los lineamientos de la DTIC. Para la Fase 1 se seleccionaron ocho historias de usuario correspondientes a los módulos de carga de información académica (HU-01), gestión del proceso de evaluación (HU-03, HU-04, HU-05), captura y registro de respuestas (HU-08, HU-09), consulta de pendientes (HU-07) y gestión de formularios dinámicos (HU-20).

4.5.2 Diseño del Sistema

Con base en el backlog definido y la arquitectura objetivo, se llevó a cabo el diseño técnico de los componentes a implementar en cada sprint: contratos de API REST entre microservicios, modelo de datos por servicio y flujo de interacción entre las interfaces de usuario y el backend. El diseño estuvo guiado en todo momento por los atributos de calidad priorizados en el ATAM, asegurando que cada decisión técnica respondiera a la arquitectura de microservicios definida previamente.

4.5.3 Implementación

El desarrollo se organizó en sprints iterativos, al final de cada uno de los cuales se entregó una versión funcional y verificable de los componentes trabajados. En la Fase 1 se implementaron cuatro microservicios (Academic Service, Assessment Service, Answers Service y Forms Service)

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

y dos interfaces de usuario (Manager UI y Catchment UI), materializando las ocho historias de usuario priorizadas. Cada microservicio fue desarrollado de forma independiente con su propia base de datos, siguiendo los principios de arquitectura hexagonal y bajo acoplamiento.

4.5.4 Pruebas

Al concluir cada sprint se ejecutaron los casos de prueba funcionales asociados a las historias de usuario implementadas, verificando el cumplimiento de los criterios de aceptación definidos en el backlog. Para la Fase 1 se diseñaron treinta casos de prueba, de los cuales catorce correspondían a las historias de usuario desarrolladas y fueron ejecutados sobre el prototipo funcional, obteniendo resultado aprobado en su totalidad.

4.5.5 Despliegue

El sistema fue desplegado mediante Docker, empaquetando cada microservicio en contenedores independientes. Este enfoque garantizó la reproducibilidad del entorno de ejecución, el aislamiento entre servicios y la alineación con la infraestructura de despliegue institucional de la DTIC.

4.6 Herramientas y Tecnologías Utilizadas

El desarrollo del sistema EVADOC requirió la combinación de herramientas de modelado, desarrollo, control de versiones e integración. Su selección estuvo orientada por los lineamientos institucionales de la DTIC y por la idoneidad técnica de cada herramienta frente a los requerimientos del sistema.

4.6.1 Tecnologías de Desarrollo

El backend de los cuatro microservicios fue desarrollado en Spring Boot (Java), framework que facilita la construcción de servicios RESTful robustos y escalables con configuración mínima.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Las interfaces de usuario (Manager UI y Catchment UI) fueron desarrolladas con Angular (TypeScript), framework de desarrollo frontend, seleccionado por los lineamientos de la DTIC. Como sistema de gestión de bases de datos se utilizó Oracle, tecnología relacional con soporte y licenciamiento institucional disponible en la UIS.

4.6.2 Herramientas de Gestión y Control

El control de versiones del código fuente se gestionó con Git, lo que permitió el trabajo colaborativo entre los integrantes del equipo, el seguimiento detallado de cambios a lo largo del desarrollo y la posibilidad de revertir modificaciones cuando fue necesario.

4.6.3 Tecnologías de Integración y Comunicación

La comunicación asíncrona entre microservicios se implementó mediante RabbitMQ, broker de mensajería con soporte institucional disponible en la DTIC, que permite el intercambio de mensajes desacoplado entre servicios sin dependencias directas entre ellos. El despliegue del sistema se realizó con Docker, plataforma de contenedores que garantiza el aislamiento de cada microservicio y la reproducibilidad del entorno de ejecución en diferentes infraestructuras.

4.6.4 Justificación Tecnológica

La selección de estas tecnologías respondió a dos criterios fundamentales: la disponibilidad de soporte y licenciamiento en la DTIC, que garantiza la sostenibilidad institucional del sistema, y la idoneidad técnica frente a los requerimientos no funcionales definidos. El análisis comparativo de las alternativas tecnológicas evaluadas (MongoDB vs Oracle, Apache Kafka vs RabbitMQ, arquitectura monolítica vs microservicios) se presenta en la sección 3.4, donde se detalla la justificación de cada decisión adoptada.

5 Resultados

El presente capítulo expone la ejecución práctica del proceso de reingeniería aplicado al sistema de evaluación docente de la Universidad Industrial de Santander. A partir del modelo metodológico definido en el capítulo anterior, se describen las actividades desarrolladas en cada fase del proyecto, evidenciando el análisis del sistema existente y la construcción de la nueva solución tecnológica.

En esta sección se presentan los resultados obtenidos durante la ingeniería inversa, el diagnóstico técnico, el rediseño arquitectónico y la reconstrucción del sistema. Asimismo, se detallan las decisiones técnicas adoptadas, las herramientas utilizadas y los criterios que orientaron el desarrollo del nuevo sistema bajo un enfoque de arquitectura dirigida y gestión ágil mediante Scrum.

El objetivo de este capítulo es demostrar la coherencia entre el sustento teórico y metodológico previamente expuesto y su aplicación práctica en el contexto institucional, evidenciando la transformación estructural del sistema y la materialización de la nueva arquitectura propuesta.

5.1 Requerimientos

Los resultados presentados en esta sección son producto de un proceso continuo e iterativo desarrollado a lo largo de las fases del proyecto. A medida que se avanzaba en el análisis del sistema existente, el levantamiento de información, la validación de los hallazgos y los artefactos generados fueron ajustados y refinados progresivamente. Lo que se expone a continuación representa la versión consolidada y definitiva de cada componente, resultado de múltiples ciclos de revisión coherentes con el enfoque de prototipado evolutivo adoptado en el proyecto.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

5.1.1 Ingeniería Inversa del Proceso EVADOC

Esta fase tuvo como propósito reconstruir el funcionamiento real del proceso de evaluación docente de la UIS, identificando actores, actividades e interacciones dentro del flujo institucional para formalizar su estructura y detectar oportunidades de mejora. Para ello, se recopiló información mediante entrevistas con el personal encargado, observación directa del procedimiento, revisión de formatos y documentos del proceso, y análisis de actividades por periodo académico. Esto permitió identificar las tareas ejecutadas, su secuencia y los puntos de intervención humana dentro del proceso.

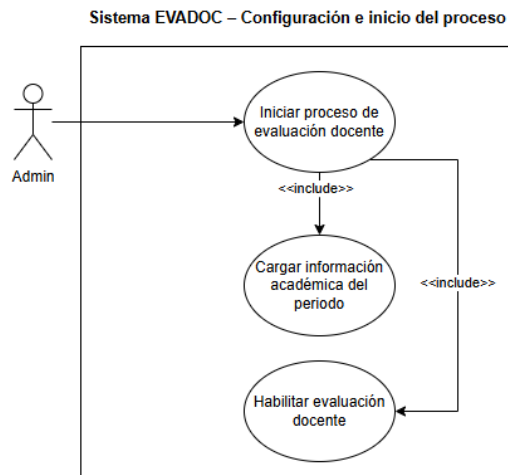
5.1.1.1 Identificación de Actores y Responsabilidades. Se identificaron cuatro actores principales: estudiantes (diligencian las encuestas), docentes (sujetos de evaluación, coevaluación y autoevaluación), personal administrativo (habilita, supervisa y consolida la información) y dependencias académicas (utilizan los resultados). Para representar sus interacciones se elaboraron diagramas de casos de uso organizados en cuatro paquetes funcionales:

1. Configuración e inicio: carga de información académica y habilitación del proceso.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 3

Configuración e inicio del proceso de evaluación.

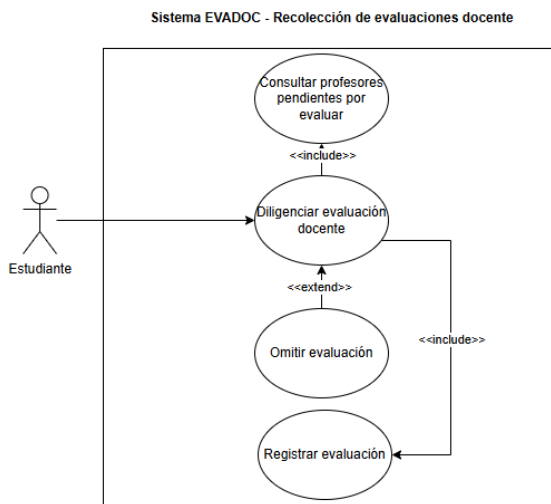


Nota: Figura elaborada a partir del análisis funcional del sistema EVADOC.

2. Recolección de evaluaciones: acceso del estudiante, consulta de profesores pendientes, diligenciamiento del formulario y registro de justificaciones ante NS/NR u omisión.

Figura 4

Recolección de evaluaciones.



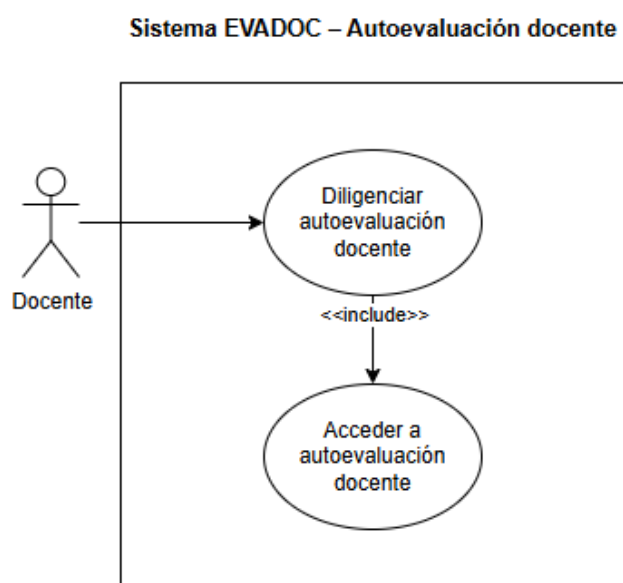
REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Nota: Diagrama de casos de uso construido con base en el estudio del proceso de evaluación docente implementado en el sistema EVADOC.

3. Autoevaluación docente: acceso y diligenciamiento del instrumento de autoevaluación por parte del docente.

Figura 5

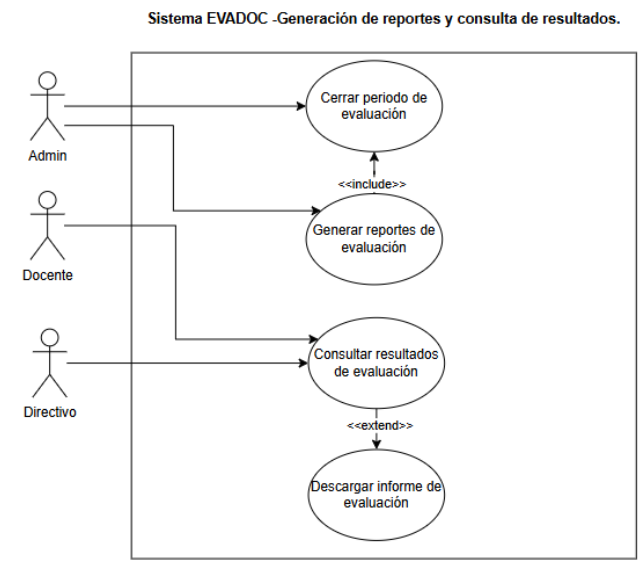
Autoevaluación docente.



Nota: Análisis del sistema EVADOC.

4. Generación de reportes: cierre del periodo, procesamiento de información y consulta de resultados según perfil de acceso.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 6*Generación de reportes y consulta de resultados.*

Nota: Representación derivada del análisis del sistema EVADOC.

5.1.1.2 Modelado del Sistema Actual. Con el fin de formalizar la comprensión obtenida, se elaboraron diagramas de procesos mediante notación BPMN. Esta representación permitió estructurar el flujo de actividades, identificar eventos de inicio y finalización, representar decisiones y bifurcaciones, y detectar oportunidades de mejora estructural, facilitando la transición desde un conocimiento operativo informal hacia una documentación técnica estructurada que sirvió como base para el diagnóstico posterior. Se modelaron cuatro flujos que cubren las etapas principales del proceso de evaluación docente:

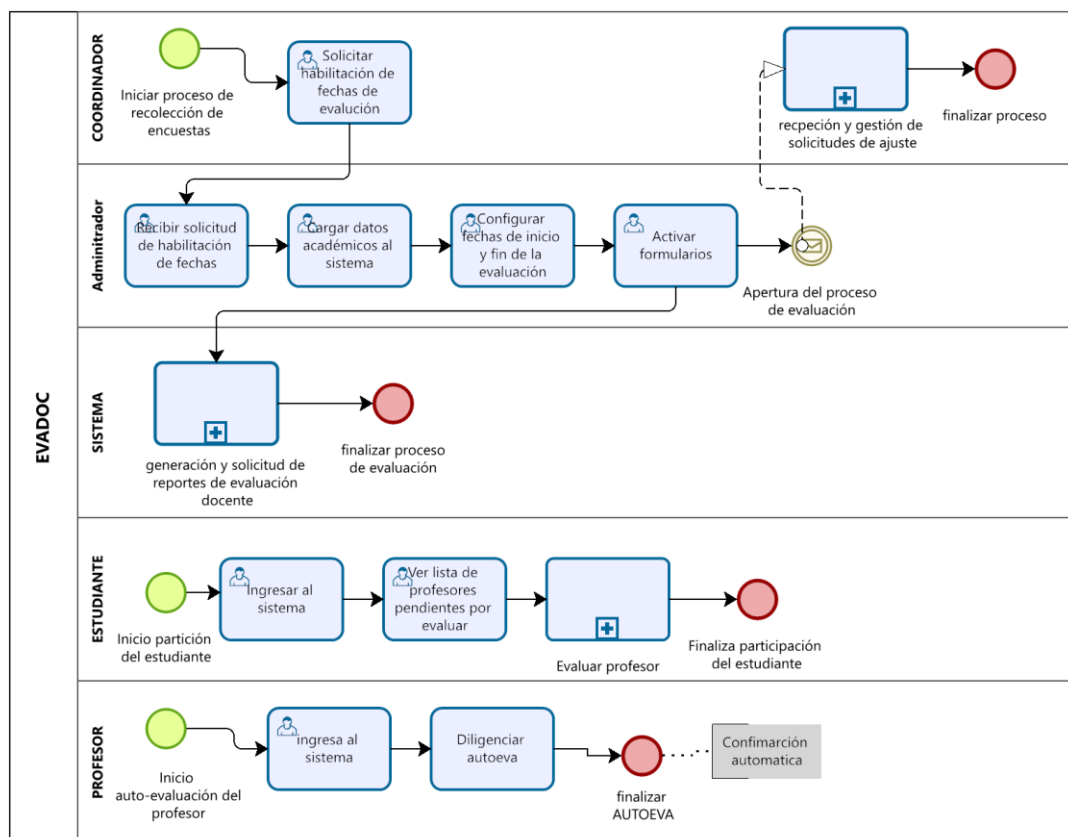
Flujo general del proceso EVADOC: El flujo general (Figura 7) describe el ciclo completo de evaluación docente. El proceso inicia cuando el coordinador habilita el periodo de recolección, cargando la información académica del semestre y activando los formularios para los estudiantes. Durante el periodo activo, los estudiantes diligencian las evaluaciones de sus docentes mientras

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

estos realizan su autoevaluación en paralelo. El sistema gestiona el seguimiento del avance y las solicitudes de ajuste que puedan surgir. Al alcanzar la fecha de cierre, el sistema consolida la información recopilada y genera los informes disponibles para las dependencias académicas según sus permisos de acceso.

Figura 7

Flujo general del proceso de evaluación docente.



Nota: Análisis del proceso institucional de evaluación docente.

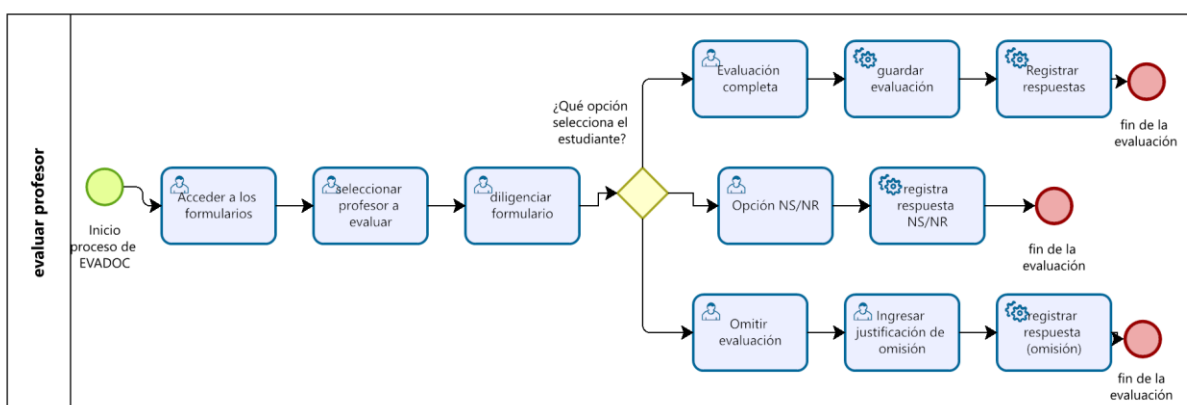
Flujo de evaluación del profesor por parte del estudiante: El flujo (Figura 8) describe el proceso desde la perspectiva del estudiante. Una vez habilitado el periodo, el estudiante accede al sistema, consulta la lista de profesores pendientes por evaluar y selecciona el docente a evaluar. Durante el diligenciamiento del formulario, el sistema contempla tres escenarios: completar la

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

evaluación en su totalidad, seleccionar la opción NS/NR (No sabe / No responde) u omitir la evaluación. En el último caso, el sistema exige el registro de una justificación antes de continuar. En todos los escenarios, al finalizar el proceso el docente desaparece de la lista de pendientes del estudiante.

Figura 8

Proceso de evaluación del profesor por parte del estudiante.



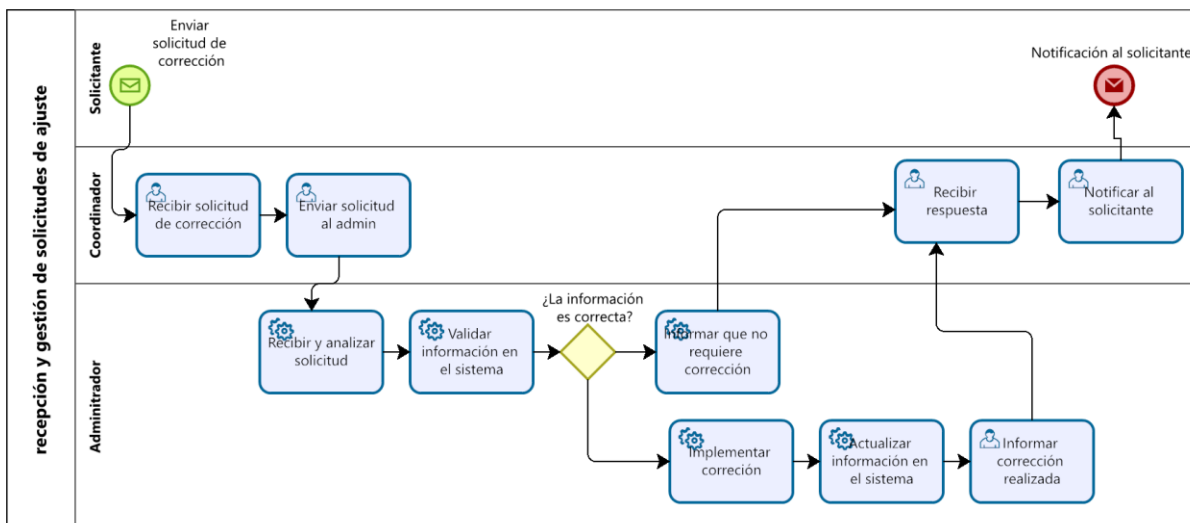
Nota: Análisis del proceso de evaluación docente.

Flujo de recepción y gestión de solicitudes de ajuste: El flujo (Figura 9) representa el procedimiento mediante el cual se gestionan las solicitudes de corrección al proceso evaluativo. El solicitante inicia el proceso enviando una solicitud de corrección; el coordinador la recibe y la traslada al administrador para su análisis. El administrador valida la información en el sistema y determina si la corrección es procedente: en caso de que la información sea correcta, informa que no se requiere ajuste; de lo contrario, implementa la corrección, actualiza la información en el sistema e informa el resultado. En ambos casos, el coordinador recibe la respuesta y notifica al solicitante, garantizando trazabilidad sobre el estado de su petición.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 9

Proceso de recepción y gestión de solicitudes de ajuste.

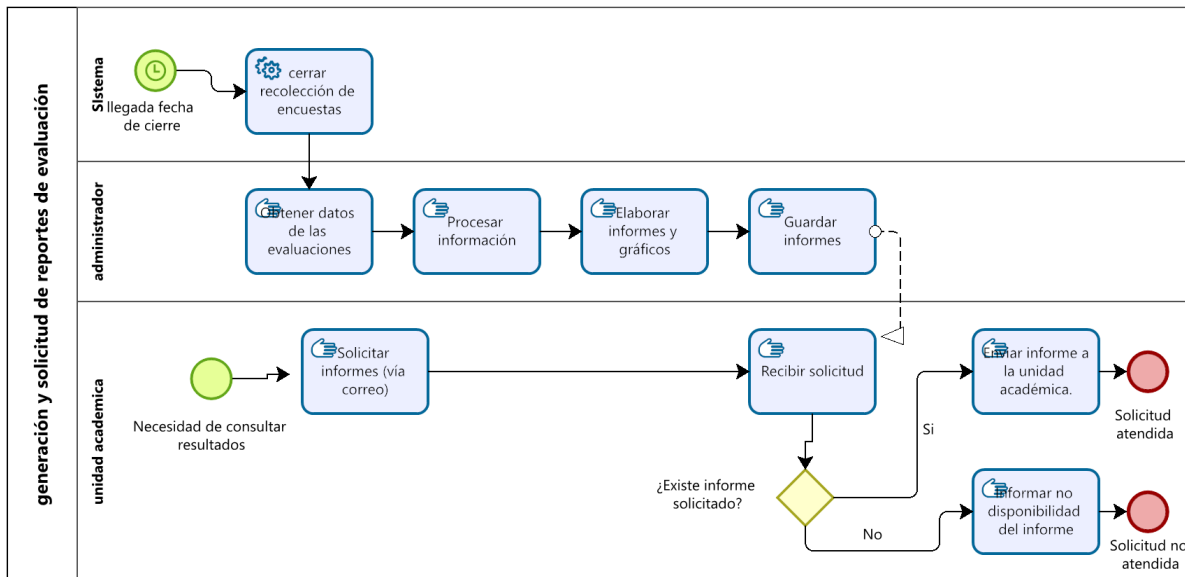


Nota: Análisis del proceso institucional de evaluación docente.

Flujo de cierre del proceso y generación de reportes: El flujo (Figura 10) describe las actividades finales del ciclo evaluativo distribuidas entre tres participantes. Al alcanzar la fecha de cierre configurada, el sistema cierra automáticamente la recolección de encuestas. A partir de ese momento, el administrador asume de forma manual la obtención de los datos de evaluación, el procesamiento de la información, la elaboración de informes y gráficos, y el almacenamiento de los resultados. En paralelo, cuando una unidad académica requiere consultar los resultados, realiza la solicitud vía correo electrónico; el administrador recibe dicha solicitud y, dependiendo de la disponibilidad del informe, procede a enviarlo a la unidad académica o a informar su no disponibilidad, determinando así si la solicitud queda atendida o no atendida.

Figura 10

Proceso de generación de reportes y cierre.



Nota: Diagrama desarrollado con base en el análisis del funcionamiento actual del sistema EVADOC.

5.1.1.3 Reconstrucción del Flujo Operativo. A partir de la información recopilada y los diagramas BPMN elaborados, se procedió a descomponer el flujo completo del proceso de evaluación docente, desde la habilitación del periodo hasta la consolidación y publicación de resultados. Este análisis permitió identificar las principales tareas ejecutadas por cada actor, distinguir las actividades manuales de las automáticas, detectar procesos secuenciales dependientes de validaciones previas, y localizar puntos críticos con riesgo de demoras o errores susceptibles de automatización.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Tabla 5*Actividades Identificadas en el Proceso Actual de Evaluación Docente*

N°	Actividad	Responsable	Tipo
1	Cargar información académica del periodo (docentes, asignaturas, estudiantes)	Administrador	Manual
2	Habilitar proceso de evaluación docente	Administrador	Manual
3	Acceder al sistema EVADOC	Estudiante	Manual
4	Consultar profesores pendientes por evaluar	Estudiante	Automático
5	Seleccionar profesor a evaluar	Estudiante	Manual
6	Diligenciar formulario de evaluación docente	Estudiante	Manual
7	Registrar justificación cuando se selecciona NS/NR u omitir evaluación	Estudiante	Manual
8	Registrar evaluación docente en el sistema	Sistema	Automático
9	Acceder al sistema para realizar autoevaluación	Docente	Manual
10	Diligenciar autoevaluación docente	Docente	Manual
11	Recibir solicitud de corrección o ajuste del proceso	Administrador	Manual
12	Validar solicitud de ajuste	Administrador	Manual
13	Aplicar corrección dentro del sistema	Administrador / Sistema	Manual
14	Cerrar periodo de recolección de encuestas	Administrador / Sistema	Manual
15	Procesar información recopilada	Sistema EVADOC	Manual
16	Generar reportes de evaluación docente	Sistema EVADOC	Manual
17	Consultar resultados de evaluación	Docente / Directivo / Unidad académica	Manual

Nota: Análisis del proceso de evaluación docente y de los diagramas BPMN desarrollados durante la fase de ingeniería inversa.

Como se observa, el proceso involucra actividades tanto manuales como automáticas. Las manuales corresponden a las acciones directas de los actores (estudiantes, docentes y

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

coordinadores), mientras que las automáticas son ejecutadas por el sistema en etapas de procesamiento, generación de reportes y gestión del flujo. Esta descomposición facilitó la identificación de oportunidades de mejora que se describen en la sección siguiente.

5.1.1.4 Hallazgos y Resultados de la Ingeniería Inversa. La aplicación de la ingeniería inversa permitió documentar formalmente el proceso de evaluación docente e identificar limitaciones funcionales, operativas y de gestión que afectan su eficiencia. Estos hallazgos constituyen el punto de partida para el diagnóstico del sistema actual y proporcionan los insumos necesarios para la definición de requerimientos y el diseño de la nueva arquitectura.

Tabla 6

Hallazgos Identificados en el Sistema Actual de Evaluación Docente

N°	Hallazgo	Descripción	Impacto en el proceso	Oportunidades de mejora
1	Dependencia de participación obligatoria del estudiante	La evaluación se realiza principalmente al final del periodo porque el sistema académico exige su diligenciamiento para permitir la matrícula del siguiente semestre.	Concentración de respuestas en un corto periodo y posible disminución en la calidad de la información recolectada.	Recordatorios progresivos y visualización del estado de avance durante el periodo.
2	Falta de notificación de apertura del proceso	El sistema no cuenta con mecanismos automáticos que informen a los estudiantes cuando el proceso ha sido habilitado.	Los estudiantes pueden desconocer el inicio del proceso hasta el final del periodo.	Notificaciones automáticas por correo o dentro de la plataforma al habilitar el proceso.
3	No se guardan avances parciales en las evaluaciones	El sistema no permite almacenar respuestas parciales; si el estudiante abandona el formulario, las respuestas se pierden.	Riesgo de pérdida de información y desmotivación para completar la evaluación.	Guardado automático o manual de avances parciales para permitir la continuidad del proceso.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

4	Gestión manual de solicitudes de ajuste	Las solicitudes de corrección requieren intervención manual del coordinador sin un flujo definido dentro del sistema.	Mayor carga administrativa y tiempo de gestión ante inconsistencias.	Automatización parcial de solicitudes con trazabilidad dentro del sistema.
5	Limitaciones en la visualización de resultados	Los resultados solo son consultables una vez finalizado el proceso, según los permisos de cada rol.	Acceso limitado a la información durante el periodo evaluativo.	Reportes más dinámicos y accesibles según el rol del usuario.
6	Dependencia del cierre manual del proceso	El cierre del periodo de evaluación depende de la intervención administrativa del coordinador.	Riesgo de retrasos en la generación de reportes y consolidación de resultados.	Cierre automático del proceso basado en fechas configuradas dentro del sistema.

Nota: Análisis del proceso de evaluación docente, del modelado BPMN y de la descomposición de actividades realizada durante la fase de ingeniería inversa.

Los hallazgos evidencian que el sistema actual cumple su función principal de recolección de información; sin embargo, presenta limitaciones en la gestión del proceso, la interacción con los usuarios y la administración de la información. Estas oportunidades de mejora sirvieron como base para la definición de los requerimientos del sistema propuesto en las etapas siguientes del proyecto.

5.1.2 *Requerimientos no Funcionales*

A partir de los hallazgos obtenidos en la fase de ingeniería inversa y considerando los atributos de calidad prioritarios para el sistema, se definieron los requerimientos no funcionales del sistema EVADOC. Estos requerimientos establecen las condiciones de calidad que debe cumplir el sistema en términos de rendimiento, seguridad, disponibilidad e interoperabilidad, y constituyen la base sobre la cual se evaluaron y justificaron las decisiones arquitectónicas mediante el método ATAM.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Tabla 7*Definición y Priorización de los Requerimientos no Funcionales.*

Requerimiento no funcional	Atributo	Descripción	Prioridad EVADOC
RNF-01	Rendimiento	El sistema debe responder eficientemente cuando múltiples usuarios acceden simultáneamente, especialmente durante el periodo de evaluación.	Alta
RNF-02	Disponibilidad	El sistema debe estar disponible durante los periodos académicos activos sin interrupciones no planificadas, dado que la evaluación está vinculada al proceso de matrícula	Alta
RNF-03	Escalabilidad	La arquitectura debe soportar el crecimiento de usuarios, formularios y periodos académicos sin degradar el rendimiento ni requerir rediseño.	Media
RNF-04	Seguridad	El sistema debe garantizar autenticación, control de acceso por roles (administrador, docente, estudiante, directivo) y protección de la información evaluativa.	Alta
RNF-05	Mantenibilidad	El sistema debe permitir realizar modificaciones en módulos específicos sin afectar el funcionamiento general del sistema.	Alta
RNF-06	Interoperabilidad	El sistema debe integrarse con los sistemas de información institucionales de la UIS (datos académicos, información de docentes y estudiantes por periodo) sin requerir duplicación manual de información.	Alta
RNF-07	Trazabilidad	El sistema debe registrar las acciones del proceso evaluativo para garantizar auditoría y control.	Media
RNF-08	Usabilidad	Las interfaces deben ser intuitivas y accesibles para todos los actores, minimizando errores durante el diligenciamiento.	Media
RNF-09	Confiabilidad	El sistema debe garantizar que las respuestas registradas no se pierdan y que los reportes sean íntegros y consistentes.	Alta

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

RNF-10	Portabilidad	El sistema debe poder desplegarse en diferentes entornos institucionales mediante contenedorización (Docker)	Media
---------------	--------------	--	-------

Nota: Identificación, descripción y priorización de los requerimientos no funcionales del sistema de evaluación docente (EVADOC).

5.1.3 *Requerimientos Funcionales*

A partir del análisis del sistema de evaluación docente de la UIS realizado en la fase de ingeniería inversa, y en el marco de la metodología de prototipado evolutivo adoptada en el proyecto, se definieron los requerimientos funcionales del sistema EVADOC. Estos requerimientos describen las capacidades que el sistema debe ofrecer para satisfacer las necesidades de cada actor del proceso evaluativo, y constituyen la base sobre la cual se formularon las historias de usuario para su implementación mediante Scrum, siguiendo los lineamientos de la DTIC.

Nota: Los requerimientos marcados con (*) corresponden a las funcionalidades desarrolladas en la Fase 1 del proyecto.

Tabla 8

Requerimientos Funcionales Identificados en el Sistema.

ID	DESCRIPCIÓN DEL REQUERIMIENTO	HU
RF-01*	El sistema debe cargar la información académica del periodo (docentes, estudiantes, asignaturas) desde los sistemas institucionales de la UIS, notificando al administrador al completar o fallar el proceso	HU-01
RF-02	El sistema debe permitir consultar la información académica cargada por periodo, con filtros por docente, asignatura o unidad organizacional	HU-02

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

RF-03*	El sistema debe permitir crear un proceso de evaluación configurando tipo, periodo académico, fechas, unidades organizacionales, tipo de evaluador, tipo de evaluado y formulario asociado, validando la coherencia de los datos antes de guardar	HU-03
RF-04*	El sistema debe habilitar el proceso de evaluación para todos los participantes de forma simultánea, cambiando su estado a "activo" y permitiendo el acceso con credenciales institucionales	HU-04
RF-05*	El sistema debe permitir registrar y modificar el cierre parcial del proceso por unidad académica sin afectar el proceso de las demás unidades, notificando al administrador en cada acción	HU-05
RF-06	El sistema debe enviar una notificación automática al estudiante cuando el proceso de evaluación sea habilitado, permitiendo el acceso directo desde la notificación	HU-06
RF-07*	El sistema debe mostrar al estudiante únicamente los profesores pendientes por evaluar en el periodo activo, actualizando la lista al completar u omitir cada evaluación	HU-07
RF-08*	El sistema debe cargar el formulario de evaluación desde el Form Service y registrar las respuestas del estudiante de forma anónima, confirmando el envío y actualizando la lista de pendientes	HU-08
RF-09*	El sistema debe permitir al estudiante omitir la evaluación de un profesor seleccionando una justificación obligatoria de un listado desplegable, registrando la omisión con fecha	HU-09
RF-10	El sistema debe permitir al docente acceder y diligenciar su formulario de autoevaluación únicamente cuando el proceso esté activo, confirmando el registro exitoso al finalizar	HU-10
RF-11	El sistema debe mostrar al docente el estado de su autoevaluación (pendiente o completada) con la fecha de registro, sin permitir modificaciones una vez enviada	HU-11
RF-12	El sistema debe registrar las solicitudes de ajuste con fecha, hora y datos del solicitante en estado "pendiente", notificando automáticamente al administrador	HU-12
RF-13	El sistema debe permitir al administrador aprobar o rechazar solicitudes de ajuste, registrando la resolución con fecha y notificando al solicitante el resultado	HU-13
RF-14	El sistema debe mantener un historial inmutable de solicitudes de ajuste por periodo, consultable con filtros por estado, fecha y solicitante	HU-14
RF-15	El sistema debe generar reportes consolidados de resultados de evaluación agrupados por unidad organizacional, disponibles desde el Dashboard	HU-15

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

RF-16	El sistema debe generar reportes individuales de evaluación por docente, incluyendo resultados de evaluación estudiantil y autoevaluación, accesibles solo para roles autorizados	HU-16
RF-17	El sistema debe filtrar la información de resultados visible según el rol del usuario, garantizando que cada actor acceda únicamente a la información que le corresponde	HU-17
RF-18	El sistema debe almacenar los reportes al cerrar cada proceso y permitir su consulta histórica por periodo y rol, con opción de exportación en PDF	HU-18
RF-19	El sistema debe publicar informes generales con datos agregados a nivel institucional al cerrar el proceso, accesibles sin autenticación desde el sitio web o intranet institucional	HU-19
RF-20*	El sistema debe permitir al administrador crear, editar y gestionar formularios dinámicos con secciones, preguntas y opciones de respuesta, impidiendo la eliminación de formularios asociados a procesos activos	HU-20

Nota: Análisis del sistema de evaluación docente.

5.1.3.1 Épicas. Los requerimientos funcionales del sistema EVADOC se organizaron en siete épicas que agrupan las funcionalidades según su naturaleza y propósito dentro del proceso evaluativo. La épica EP-01 Gestión de Información Académica contempla la carga y consulta de los datos institucionales del periodo, como docentes, estudiantes y asignaturas, provenientes de los sistemas de la UIS. La épica EP-02 Configuración del Proceso de Evaluación agrupa las funcionalidades relacionadas con la creación, habilitación y gestión del cierre parcial de los procesos evaluativos. La épica EP-03 Captación de Respuestas cubre el flujo del estudiante durante la evaluación: consulta de profesores pendientes, diligenciamiento del formulario y registro de omisiones. La épica EP-04 Autoevaluación Docente contempla el acceso y diligenciamiento del instrumento de autoevaluación por parte del docente. La épica EP-05 Gestión de Solicitudes de Ajuste agrupa las funcionalidades de registro, resolución y seguimiento de solicitudes de corrección al proceso evaluativo. La épica EP-06 Generación de Reportes comprende la producción y consulta de resultados por unidad organizacional, por docente y por rol, incluyendo

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

el historial de periodos anteriores y el acceso público a informes institucionales. Finalmente, la épica EP-07 Gestión de Usuarios y Roles contempla la administración de los formularios dinámicos de evaluación utilizados en el proceso.

5.1.3.2 Historias de Usuario. A partir de los hallazgos obtenidos en la fase de ingeniería inversa, los requerimientos no funcionales establecidos y los requerimientos funcionales definidos, se formularon las historias de usuario del sistema EVADOC. Estas historias representan, desde la perspectiva de cada actor, las funcionalidades que el sistema debe ofrecer para responder a las necesidades del proceso de evaluación docente de la UIS. A continuación, se presenta un resumen de las veinte historias de usuario distribuidas en las siete épicas definidas, indicando para cada una su identificador, la épica a la que pertenece, el título y la prioridad asignada. La descripción detallada de cada historia de usuario, incluyendo sus criterios de aceptación, se encuentra en los anexos del presente documento.

Tabla 9

Resumen de Historias de Usuario Organizadas por Épicas

ID	Épica	Título	Prioridad
HU-01	EP-01	Cargar información académica del periodo	Alta
HU-02	EP-01	Consultar información académica cargada	Media
HU-03	EP-02	Crear proceso de evaluación docente	Alta
HU-04	EP-02	Habilitar proceso de evaluación	Alta
HU-05	EP-02	Gestionar cierre parcial del proceso por unidad académica	Alta
HU-06	EP-03	Recibir notificación de apertura del proceso	Alta
HU-07	EP-03	Consultar profesores pendientes por evaluar	Alta
HU-08	EP-03	Diligenciar formulario de evaluación docente	Alta
HU-09	EP-03	Omitir evaluación con justificación	Media
HU-10	EP-04	Acceder al formulario de autoevaluación	Alta
HU-11	EP-04	Consultar estado de autoevaluación	Media
HU-12	EP-05	Recibir solicitud de ajuste	Alta
HU-13	EP-05	Validar y resolver solicitud de ajuste	Alta

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

HU-14	EP-05	Consultar historial de solicitudes	Media
HU-15	EP-06	Generar reporte por unidad organizacional	Alta
HU-16	EP-06	Generar reporte individual por docente	Alta
HU-17	EP-06	Consultar resultados según rol	Alta
HU-18	EP-06	Consultar historial de reportes de periodos anteriores	Alta
HU-19	EP-06	Acceso público a informes generales de la universidad	Media
HU-20	EP-07	Gestionar formularios dinámicos	Alta

Nota: La tabla presenta las historias de usuario definidas para el sistema EVADOC.

5.2 Diseño.

Con base en los requerimientos funcionales y no funcionales definidos, se procedió a la fase de diseño del sistema EVADOC. Esta etapa comprende la evaluación arquitectónica mediante el método ATAM, la definición de la arquitectura propuesta bajo un enfoque de microservicios, y el modelado del proceso mejorado mediante notación BPMN. Las decisiones de diseño adoptadas en esta fase responden directamente a los atributos de calidad priorizados y a los lineamientos institucionales de la DTIC, garantizando coherencia entre los requerimientos establecidos y la solución arquitectónica construida.

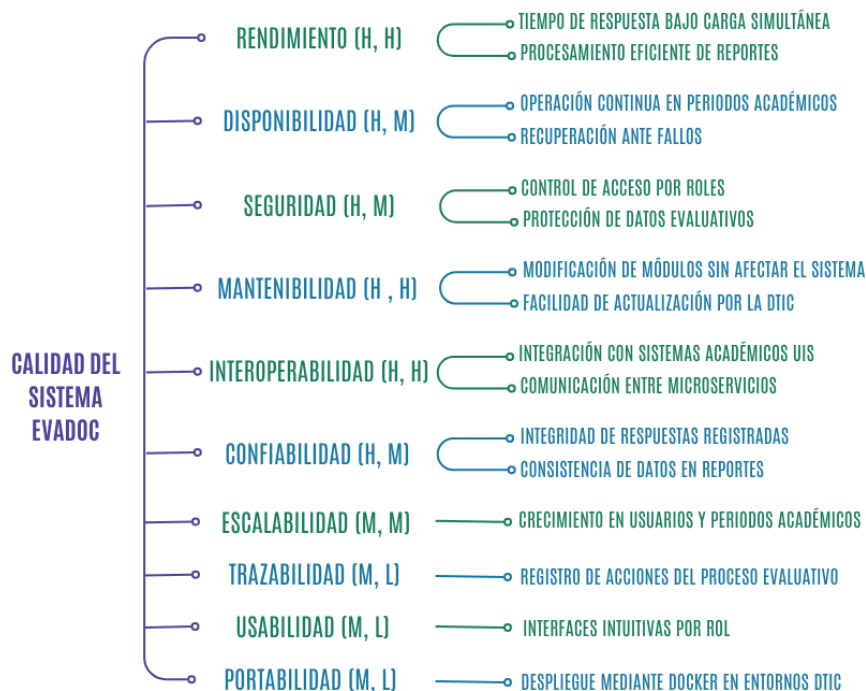
5.2.1 ATAM Aplicado al Sistema EVADOC

5.2.1.1 Árbol de Utilidad. El árbol de utilidad organiza los atributos de calidad del sistema por importancia, clasificando cada uno con dos criterios: importancia para el negocio (H=Alta, M=Media, L=Baja) y dificultad técnica de lograrla (H=Alta, M=Media, L=Baja). Esta priorización permite identificar qué atributos deben ser atendidos con mayor rigor en las decisiones arquitectónicas del sistema.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 11

Árbol de utilidad del método ATAM.



Nota: Aplicación del método ATAM para la identificación y priorización de atributos de calidad del sistema EVADOC.

Los requerimientos no funcionales detallados, así como los escenarios de atributos de calidad asociados a cada nodo del árbol, se presentan en los Anexos del presente documento. Allí se describe para cada atributo el estímulo, contexto, respuesta esperada y medida de cumplimiento, permitiendo una evaluación formal y trazable de la arquitectura propuesta frente a los objetivos de calidad del sistema.

5.2.1.2 Analisis de Trade-Offs – Decisiones Arquitectónicas. El análisis de trade-offs permite identificar los compromisos implícitos en cada decisión arquitectónica adoptada para el sistema EVADOC. Cada decisión fue evaluada considerando los beneficios que aporta a los

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

atributos de calidad prioritarios y los sacrificios que implica, determinando si la arquitectura resultante responde adecuadamente a los objetivos del sistema y a los lineamientos institucionales de la DTIC.

Tabla 10

Decisiones Arquitectónicas, Análisis de Beneficios y Sacrificios.

Decisión	Beneficio	Sacrificio	Razón
Oracle en lugar de MongoDB	Integridad relacional, soporte institucional	Menor flexibilidad para datos documentales	Lineamiento institucional DTIC
RabbitMQ en lugar de Kafka	Soporte institucional, mensajería asíncrona funcional	Menor rendimiento en volúmenes muy altos	Lineamiento institucional DTIC
Microservicios	Mantenibilidad y escalabilidad por módulo	Mayor complejidad de despliegue y comunicación	Decisión técnica del equipo
Docker	Portabilidad y estandarización de entornos	Requiere conocimiento de contenedores en DTIC	Decisión técnica del equipo

Nota: Análisis de decisiones arquitectónicas del sistema EVADOC, considerando beneficios, sacrificios y lineamientos institucionales.

5.2.2 *Arquitectura Propuesta – Sistema EVADOC*

El sistema EVADOC se diseñó bajo una arquitectura de microservicios, estructurada en tres niveles: interfaces de usuario, capa de servicios y capa de datos, complementada con un sistema de mensajería asíncrona y contenedorización para el despliegue.

1. Interfaces de usuario: El sistema EVADOC cuenta con tres interfaces de usuario diferenciadas por rol y propósito, diseñadas para cubrir las necesidades específicas de cada actor del proceso evaluativo. Esta separación garantiza que cada usuario acceda únicamente

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

a las funcionalidades correspondientes a su rol, contribuyendo al control de acceso, la usabilidad y la claridad operativa del sistema.

Tabla 11

Interfaces de Usuario, Tecnologías Utilizadas y Funcionalidades.

Interfaz	Tecnología	Función
Evadoc Manager UI	Angular	Interfaz administrativa para la configuración del proceso evaluativo: CRUD de formularios dinámicos, creación de procesos de evaluación y cierre selectivo de periodos
Evadoc Catchment UI	Angular	Interfaz para la captación de respuestas: acceso de estudiantes, visualización de profesores pendientes por evaluar y gestión de omisiones
Evadoc Dashboard UI	Metabase	Interfaz de visualización y análisis de resultados: estadísticas por unidad organizacional, estadísticas individuales e informes consolidados

Nota: Definición de las interfaces de usuario y tecnologías empleadas en el sistema

EVADOC.

2. Capa de microservicios: Constituyen el núcleo funcional del sistema EVADOC. Cada servicio es independiente, tiene una responsabilidad única y cuenta con su propia base de datos, lo que garantiza el aislamiento entre dominios, facilita el mantenimiento individual de cada módulo y permite escalar de forma selectiva según la carga del sistema. La comunicación entre servicios se realiza de forma asíncrona mediante RabbitMQ, evitando dependencias directas que puedan comprometer la disponibilidad del sistema ante fallos parciales.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Tabla 12*Microservicios, Responsabilidades y Fuentes de Datos.*

Microservicio	Responsabilidad	Fuente de datos
Academic Service	Gestiona la información académica del periodo (docentes, estudiantes, asignaturas). Se integra directamente con los sistemas de datos institucionales de la UIS	Datos UIS + Oracle
Assessment Service	Gestiona la creación, habilitación y cierre de los procesos de evaluación docente.	Oracle
Answer Service	Gestiona el registro y almacenamiento de las respuestas de los estudiantes durante el proceso evaluativo	Oracle
Forme Service	Gestiona la estructura u configuración de los formularios de evaluación	Oracle

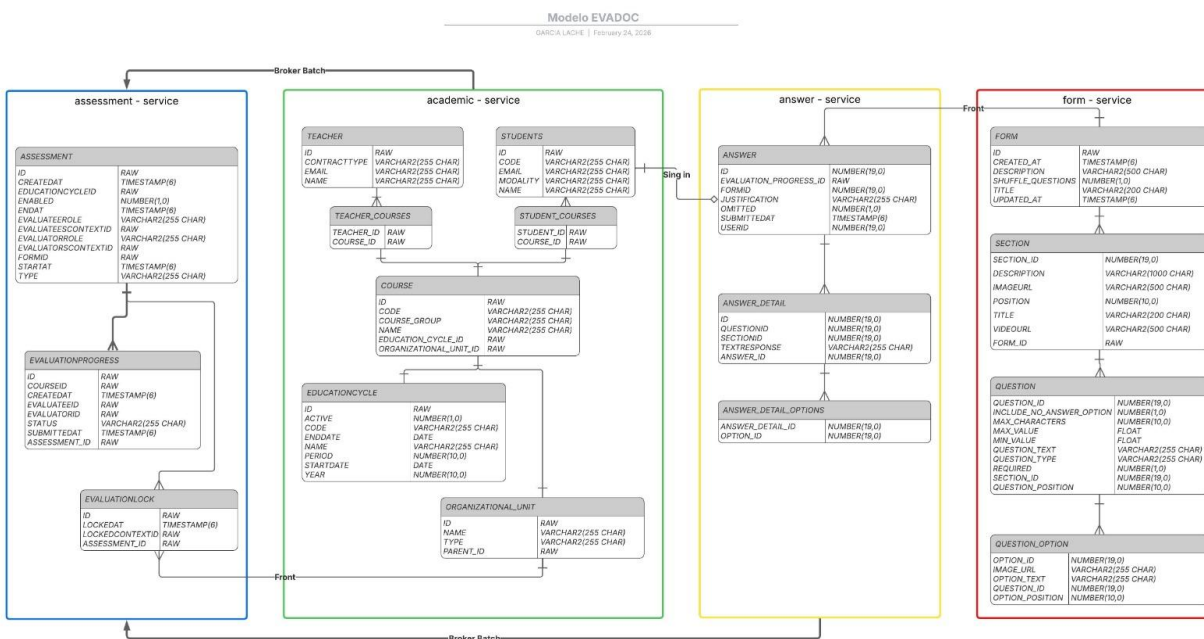
Nota: Definición de la arquitectura de microservicios del sistema EVADOC.

3. Capa de datos: cada microservicio cuenta con su propia base de datos Oracle relacional, garantizando el aislamiento de datos entre servicios y la integridad referencial dentro de cada dominio. Esta decisión, aunque difiere de la propuesta técnica inicial (MongoDB), responde a los lineamientos de soporte institucional establecidos por la DTIC. El siguiente diagrama presenta el modelo de base de datos del sistema EVADOC, ilustrando la estructura relacional de cada microservicio, las entidades que lo componen y las relaciones entre ellas. Este diseño refleja el principio de aislamiento de datos entre servicios, donde cada microservicio gestiona su propio esquema de manera independiente.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 12

Modelo de datos basado en arquitectura de microservicio



Nota: Diseño del modelo de datos del sistema EVADOC, organizado por microservicios.

4. Sistema de mensajería – RabbitMQ: La comunicación asíncrona entre los microservicios Academic, Assessment y Answer se realiza mediante RabbitMQ a través de topics. Este mecanismo permite desacoplar los servicios, garantizar la entrega de mensajes ante fallos temporales y mantener la consistencia del proceso evaluativo sin dependencias directas entre servicios.
5. Contenedorización Docker: todos los componentes del sistema se despliegan mediante Docker, lo que garantiza la portabilidad entre entornos, estandariza la configuración del sistema y facilita su instalación y mantenimiento por parte de la DTIC.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

5.2.2.1 Justificación de la Arquitectura frente a los RNF. La arquitectura propuesta para el sistema EVADOC fue diseñada teniendo en cuenta los requerimientos no funcionales definidos previamente. La siguiente tabla muestra la correspondencia entre cada atributo de calidad y el elemento arquitectónico que lo satisface, evidenciando que las decisiones de diseño adoptadas responden de forma directa a los objetivos de calidad del sistema.

Tabla 13

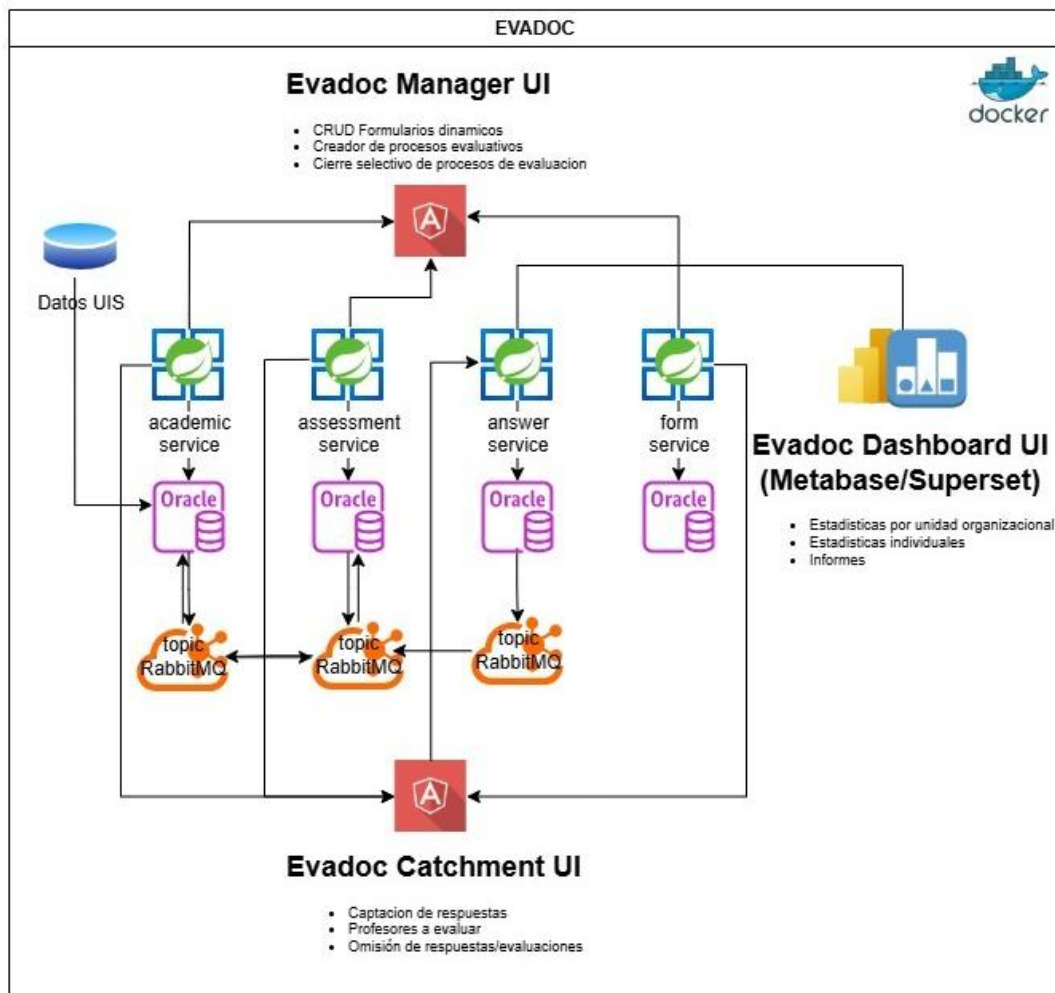
Relación entre los RNF y su Solución en la Arquitectura.

RNF	Cómo lo resuelve la arquitectura
Rendimiento	Microservicios permiten escalar cada servicio de forma independiente según la carga.
Disponibilidad	El fallo de un microservicio no afecta los demás.
Seguridad	Control de acceso diferenciado por interfaz y rol.
Mantenibilidad	Cada microservicio puede modificarse de forma independiente.
Interoperabilidad	Academic Service se integra directamente con los sistemas UIS.
Confiabilidad	RabbitMQ garantiza entrega de mensajes ante fallos temporales.
Escalabilidad	Cada microservicio puede escalar horizontalmente de forma independiente.
Portabilidad	Docker permite despliegue de cualquier entorno institucional.

Nota: Relación entre los requerimientos no funcionales y las decisiones arquitectónicas implementadas en el sistema EVADOC.

5.2.2.2 Diagrama de Arquitectura. A continuación, se presenta el diagrama de arquitectura del sistema EVADOC, el cual ilustra la organización general de sus componentes, las relaciones entre ellos y los mecanismos de comunicación adoptados. Este diagrama permite visualizar de manera integral la distribución de responsabilidades entre las interfaces de usuario, los microservicios, las bases de datos y la infraestructura de mensajería y despliegue.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 13*Arquitectura de microservicios del sistema.*

Nota: Diseño de la arquitectura de microservicios del sistema EVADOC.

5.2.3 Diagrama de Procesos del Sistema EVADOC Propuesto

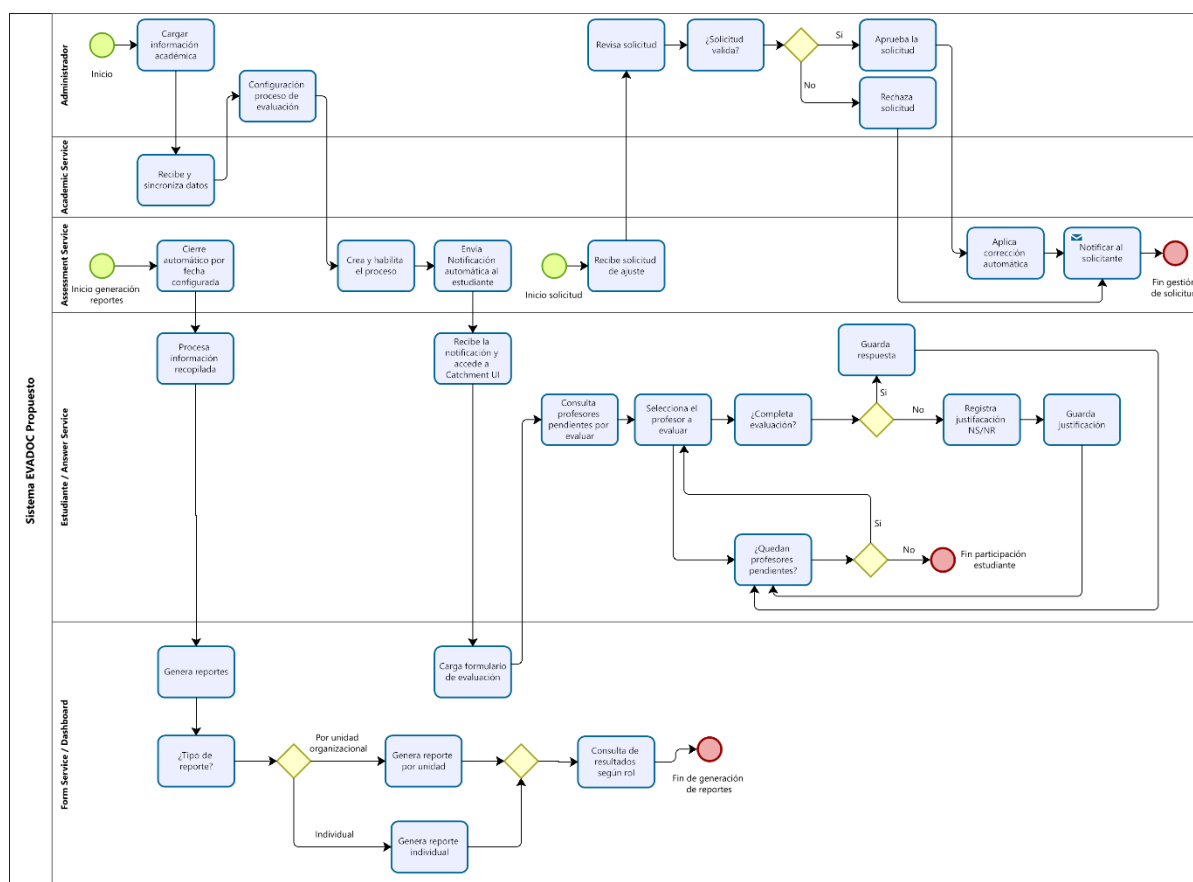
El siguiente diagrama representa el flujo general del proceso de evaluación docente en el sistema EVADOC propuesto, modelado mediante notación BPMN. A diferencia del sistema actual, el nuevo proceso incorpora mejoras como el cierre automático por fecha configurada, notificaciones automáticas a los estudiantes, guardado de respuestas parciales y automatización en

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

la gestión de solicitudes de ajuste. El diagrama se organiza en cinco carriles que representan los actores e infraestructura del sistema: Administrador, Academic Service, Assessment Service, Estudiante/Answer Service y Form Service/Dashboard.

Figura 14

Diagrama BPMN del proceso propuesto de evaluación docente.



Nota: El diagrama modela el proceso propuesto de evaluación docente, identificando actores, decisiones y mejoras funcionales del sistema

5.3 Implementación del Sistema Fase 1

5.3.1 Organización del Trabajo Realizado

La implementación de la Fase 1 del sistema EVADOC se organizó mediante la metodología Scrum bajo los lineamientos de la DTIC de la Universidad Industrial de Santander. El trabajo se distribuyó en sprints iterativos, priorizando las ocho historias de usuario definidas para esta fase: HU-01, HU-03, HU-04, HU-05, HU-07, HU-08, HU-09 y HU-20. Cada sprint incluyó actividades de planificación, desarrollo, revisión de código mediante Merge Requests y validación funcional sobre el prototipo.

El sistema se organizó en seis repositorios independientes alojados en la instancia GitLab de la UIS, bajo el grupo Gestor de evaluaciones docentes de la Secretaría General: cuatro repositorios para los microservicios del backend (back-evadoc-academic-service, back-evadoc-assessment-service, back-evadoc-answer-service, back-evadoc-forms-service) y dos para las interfaces de usuario (front-evadoc-manager y front-evadoc-catchment).

La siguiente tabla resume los módulos implementados, las historias de usuario asociadas y las interfaces involucradas:

Tabla 14

Módulos Implementados en la Fase 1 del Sistema EVADOC

Módulo	Tipo	HU implementada	Descripción
Academic Service	Microservicio	HU-01	Recibe y almacena la información académica del periodo (docentes, estudiantes, asignaturas) para ponerla disponible en el sistema
Assessment Service	Microservicio	HU-03, HU-04, HU-05	Gestiona la creación, habilitación y cierre parcial de los procesos de evaluación; obtiene

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

las relaciones evaluador-evaluado desde el Academic Service			
Answer Service	Microservicio	HU-08, HU-09	Captura y almacena las respuestas de los estudiantes y el registro de omisiones con justificación
Form Service	Microservicio	HU-20	Gestiona la estructura y configuración de los formularios de evaluación dinámicos
Manager UI	Interfaz (Angular)	HU-20	Permite al administrador crear, editar y gestionar formularios de evaluación desde una interfaz web
Catchment UI	Interfaz (Angular)	HU-07, HU-08, HU-09	Permite al estudiante consultar los profesores pendientes por evaluar, diligenciar el formulario de evaluación y registrar omisiones con justificación

Nota: Descripción de los módulos del sistema, su tipo, historias de usuario asociadas y funcionalidad.

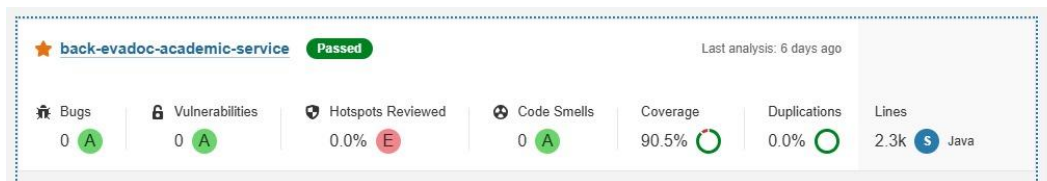
A continuación, se describe la funcionalidad implementada en cada módulo:

1. Academic Service: tiene como responsabilidad recibir y almacenar la información académica correspondiente al periodo de evaluación. Para ello, el administrador inicia el proceso de carga desde el sistema, el cual obtiene los datos de docentes, estudiantes y asignaturas desde los sistemas institucionales de la UIS y los almacena en la base de datos del servicio. Esta información queda disponible para que el Assessment Service pueda construir las relaciones evaluador-evaluado necesarias al momento de configurar un proceso de evaluación. El servicio notifica al administrador al completar o fallar el proceso de carga, garantizando trazabilidad sobre el estado de la información del periodo.

Figura 15

Análisis de calidad del academic service

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

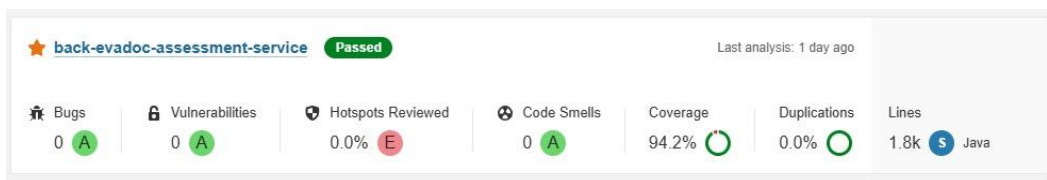


Nota: La imagen presenta los resultados del análisis de calidad del servicio, incluyendo métricas de errores, vulnerabilidades, cobertura de pruebas y duplicación de código, evidenciando el estado técnico del Academic Service.

2. **Assessment Service:** centraliza la lógica de gestión de los procesos de evaluación docente. A partir de la información almacenada por el Academic Service, este microservicio construye las relaciones entre evaluadores (estudiantes) y evaluados (docentes) según el tipo de proceso configurado. En la Fase 1 se implementaron tres funcionalidades principales: la creación del proceso de evaluación (HU-03), que permite al administrador configurar el tipo, periodo, fechas, unidades organizacionales y formulario asociado; la habilitación del proceso (HU-04), que activa el acceso simultáneo para todos los participantes; y la gestión del cierre parcial por unidad académica (HU-05), que permite cerrar el proceso de forma selectiva sin afectar las demás unidades activas. La comunicación entre el Assessment Service y el Academic Service se realiza de forma asíncrona mediante RabbitMQ.

Figura 16

Análisis de calidad del assessment service



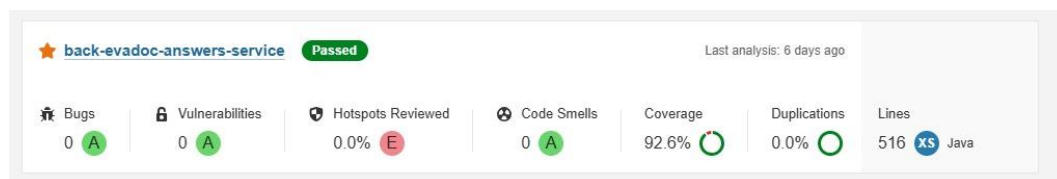
REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Nota: Métricas de calidad del servicio, incluyendo cobertura, errores y duplicación de código.

3. Answer Service: es el responsable de capturar y persistir las respuestas de los estudiantes durante el proceso de evaluación. Cuando el estudiante diligencia y envía el formulario de evaluación (HU-08), el servicio registra las respuestas de forma anónima y actualiza el estado del evaluador en la tabla del Assessment Service. Asimismo, cuando el estudiante decide omitir la evaluación de un profesor (HU-09), el Answer Service registra la omisión junto con la justificación seleccionada y la fecha, garantizando que el evento quede trazado en el sistema.

Figura 17

Análisis de calidad del answer service.

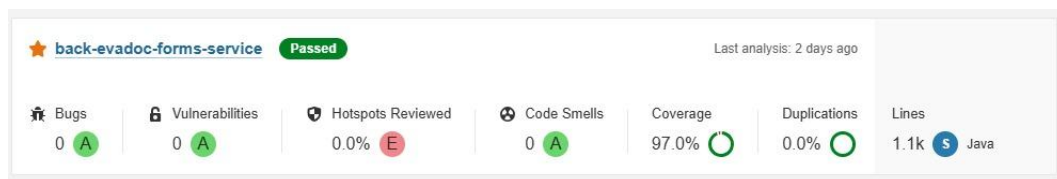


Nota: Resultados del análisis de calidad del servicio durante la gestión de respuestas del sistema.

4. Form Service: gestiona la estructura y configuración de los formularios de evaluación utilizados en el proceso. Permite al administrador crear formularios dinámicos con secciones, preguntas y opciones de respuesta desde el Manager UI, editarlos antes de asociarlos a un proceso activo e impedir su eliminación cuando ya están en uso. Los formularios almacenados en este servicio son consumidos por la Catchment UI al momento de presentarlos al estudiante durante la evaluación.

Figura 18

Análisis de calidad del form service.



Nota: Resultados del análisis de calidad del servicio en la gestión de formularios.

5. Manager UI: El Manager UI es la interfaz administrativa del sistema EVADOC, desarrollada en Angular. En la Fase 1 se implementaron los módulos de gestión de participantes (docentes, estudiantes, unidades organizacionales, periodos académicos y cursos), gestión de formularios dinámicos y configuración de procesos evaluativos.
6. Catchment UI: es la interfaz de captación de respuestas del sistema EVADOC, desarrollada en Angular y orientada al evaluador (estudiante). En la Fase 1 se implementaron tres módulos: la consulta de profesores pendientes por evaluar (HU-07), que presenta al estudiante la lista de docentes asociados a sus asignaturas en el periodo activo; el diligenciamiento del formulario de evaluación (HU-08), que carga el instrumento desde el Form Service y registra las respuestas a través del Answer Service; y el registro de omisiones con justificación (HU-09), que permite al estudiante indicar formalmente que no puede evaluar a un docente, seleccionando obligatoriamente una justificación del listado disponible.

5.3.2 *Versionamiento del Proyecto Implementado*

El control de versiones del proyecto se gestionó mediante GitLab, bajo el grupo Gestor de evaluaciones docentes de la Secretaría General de la UIS. El proyecto quedó organizado en 6

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

repositorios independientes: 4 repositorios para el backend y 2 repositorios para el frontend, tal como se detalla a continuación:

Repositorios backend: back-evadoc-academic-service, back-evadoc-assessment-service, back-evadoc-answers-service, back-evadoc-forms-service

Repositorios frontend: front-evadoc-manager, front-evadoc-catchment*

Cada repositorio cuenta con una rama principal (main) y una rama de desarrollo (develop), sobre la cual se integraron los cambios mediante Merge Requests (MR) como mecanismo formal de revisión de código, siguiendo la política de ramas de la DTIC. A lo largo de la Fase 1 se registraron MRs relacionados con correcciones de calidad (sonar smells), manejo global de excepciones (global handler), corrección de entidades (entities fix) y migración de mensajería RabbitMQ

Cada microservicio sigue la arquitectura hexagonal en su organización interna de paquetes, con las capas: config, controller, dto, mapper, model, repository y service, garantizando la separación de responsabilidades y facilitando el mantenimiento independiente de cada módulo.

Las siguientes figuras muestran la estructura del repositorio y la organización de paquetes del código fuente para cada microservicio:

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 19

Estructura raíz del repositorio academic service

The screenshot shows the root directory of the 'back-evadoc-academic-service' repository. The breadcrumb path is 'RSI / Secretaría General / Gestor de evaluaciones docentes / back-evadoc-academic-service'. The repository name is 'back-evadoc-academic-service' and the current branch is 'develop'. A merge commit 'Merge branch 'feature/01-deploy' into 'develop'' by Jorge Julian Garcia Hernández is shown. The commit hash is 'bd157748'. The repository statistics include 95 Commits, 4 Branches, 20 Tags, 20 Releases, and 4 Environments. The project information shows a coverage of 90.10% and Swagger API Documentation. The repository was created on September 22, 2025.

Name	Last commit	Last update
.gitlab	refactor(01): pipe settings	2 months ago
._mvn/wrapper	Initial commit	6 months ago
src	feat(01): rabbit tests	7 hours ago
.dockerignore	refactor(01): dockerignore	2 months ago
.gitattributes	Initial commit	6 months ago
.gitignore	feat(01): logs, env vars	1 week ago
.gitlab-ci.yml	feat(01): ci project name	2 months ago
Dockerfile	feat(01): docker entry point	2 months ago
README.md	feat(01): Readme	1 week ago
ci_settings.xml	refactor(01): pipe settings	2 months ago
dockerfile.local	refactor(01): ci/cd	4 months ago
mvnw	Initial commit	6 months ago
mvnw.cmd	Initial commit	6 months ago

Figura 20

Organización de paquetes src del academic service

The screenshot shows the directory structure of the 'src' folder in the 'back-evadoc-academic-service' repository. The breadcrumb path is 'RSI / Secretaría General / Gestor de evaluaciones docentes / back-evadoc-academic-service / src / main / java / co / edu / uis / academic'. The current branch is 'develop'. A merge commit 'feat(01): rabbit producer' by Jorge Julian Garcia Hernández is shown. The commit hash is '98f63b73'. The repository statistics include 95 Commits, 4 Branches, 20 Tags, 20 Releases, and 4 Environments. The project information shows a coverage of 90.10% and Swagger API Documentation. The repository was created on September 22, 2025.

Name	Last commit	Last update
..		
config	feat(01): rabbit exchange	8 hours ago
controller	feat(01): global handler	1 week ago
dto	feat(01): rabbit producer	8 hours ago
mapper	feat(01): sonar smells	1 week ago
model	feat(01): rabbit lib	2 months ago
repository	feat(01): Javadoc, RSI exceptions, Swagger	2 weeks ago
service	feat(01): rabbit producer	8 hours ago
AcademicServiceApplication.java	feat(01): rabbit lib	2 months ago

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 21

Estructura raíz del repositorio answer service

The screenshot shows the root directory of a GitHub repository named 'back evadoc answers service'. The repository is on the 'develop' branch. A recent merge of the 'feature/01-deploy' branch into 'develop' is shown, authored by Jorge Julian Garcia Hernández 13 minutes ago. The repository has 0 stars and 0 forks.

Project information:

- coverage: 92.50%
- Swagger API Documentation
- 64 Commits
- 4 Branches
- 13 Tags
- 13 Releases
- 1 Environment
- README
- CI/CD configuration
- Add Wiki
- Created on September 22, 2025

Name	Last commit	Last update
.gitlab	feat(01): profile properties	2 months ago
.mvn/wrapper	initial commit	6 months ago
src	feat(01): tests	20 minutes ago
.dockerignore	feat(01): profile properties	2 months ago
.gitattributes	initial commit	6 months ago
.gitignore	feat(01): logs, env vars	1 week ago
.gitlab-ci.yml	feat(01): profile properties	2 months ago
Dockerfile	Merge branch 'develop' of https://gitlab...	2 months ago
README.md	feat(01): Readme	1 week ago
ci_settings.xml	feat(01): profile properties	2 months ago
dockerfile.local	refactor(01): ci/cd	4 months ago
mvnw	initial commit	6 months ago

Figura 22

Organización de paquetes src del answer service

The screenshot shows the directory structure of the 'src' folder in the 'back evadoc answers service' repository. The repository is on the 'develop' branch. A recent commit 'feat(01): answer mapper' by Jorge Julian Garcia Hernández is shown, authored 47 minutes ago.

Name	Last commit	Last update
..		
config	feat(01): cors config	5 hours ago
controller	feat(01): handler	1 week ago
dto	feat(01): handler	1 week ago
mapper	feat(01): answer mapper	47 minutes ago
model	feat(01): answer mapper	47 minutes ago
repository	feat(01): rabbitMQ	2 months ago
service	feat(01): answer mapper	47 minutes ago
AnswersServiceApplication.java	feat(01): rabbitMQ	2 months ago

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 23

Estructura raíz del repositorio assessment service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc assessment service

B back evadoc assessment service

deveLop back-evadoc-assessment-service

Merge branch 'feature/01-deploy' into 'deveLop' ✓ 227beee1 History

Name	Last commit	Last update
.gitlab	feat(01): profile properties	2 months ago
.mvn/wrapper	Initial commit	6 months ago
src	feat(01): reprocess assessment	10 hours ago
.dockerignore	feat(01): profile properties	2 months ago
.gitattributes	Initial commit	6 months ago
.gitignore	feat(01): logs, env vars	1 week ago
.gitlab-ci.yml	feat(01): profile properties	2 months ago
Dockerfile	Edit dockerfile	2 months ago
README.md	feat(01): Readme	1 week ago
ci_settings.xml	feat(01): profile properties	2 months ago
dockerfile.local	refactor(01): ci/cd	4 months ago
mvnw	Initial commit	6 months ago

Project information

Coverage 91.50%

Swagger API Documentation

- 88 Commits
- 5 Branches
- 20 Tags
- 20 Releases
- 1 Environment

README

CI/CD configuration

+ Add Wiki

Created on September 22, 2025

Figura 24

Organización de paquetes src del assessment service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc assessment service

deveLop back-evadoc-assessment-service / src / main / java / co / edu / uis / assessment

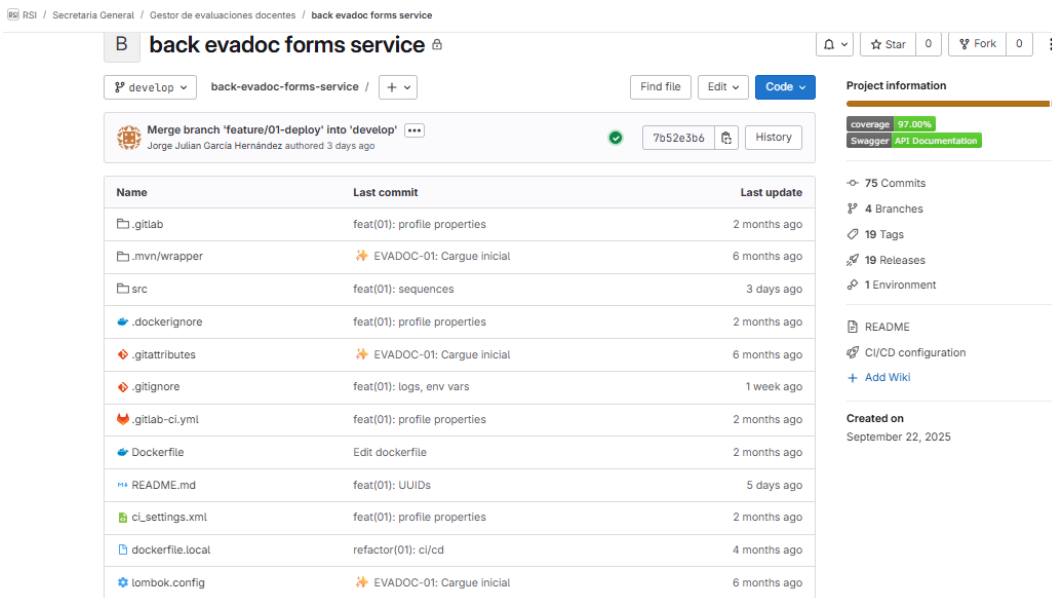
feat(01): reprocess assessment ✓ cee55e2f History

Name	Last commit	Last update
...		
config	feat(01): security config	1 week ago
controller	feat(01): reprocess assessment	10 hours ago
dto	feat(01): reprocess assessment	10 hours ago
mapper	feat(01): entities fix	5 days ago
model	feat(01): reprocess assessment	10 hours ago
repository	feat(01): sonar smells	3 days ago
service	feat(01): reprocess assessment	10 hours ago
AssessmentServiceApplication.java	feat(01): rabbitMQ	2 months ago

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 25

Estructura raíz del repositorio form service



RSI / Secretaría General / Gestor de evaluaciones docentes / back-evadoc-forms-service

back-evadoc-forms-service

Merge branch 'feature/01-deploy' into 'develop'
 Jorge Julian Garcia Hernández authored 3 days ago

Name	Last commit	Last update
.gitlab	feat(01): profile properties	2 months ago
.mvn/wrapper	EVADOC-01: Cargue inicial	6 months ago
src	feat(01): sequences	3 days ago
.dockerignore	feat(01): profile properties	2 months ago
.gitattributes	EVADOC-01: Cargue inicial	6 months ago
.gitignore	feat(01): logs, env vars	1 week ago
.gitlab-ci.yml	feat(01): profile properties	2 months ago
Dockerfile	Edit dockerfile	2 months ago
README.md	feat(01): UUIDs	5 days ago
ci_settings.xml	feat(01): profile properties	2 months ago
dockerfile.local	refactor(01): ci/cd	4 months ago
lombok.config	EVADOC-01: Cargue inicial	6 months ago

Project information

coverage: 97.00%
Swagger API Documentation

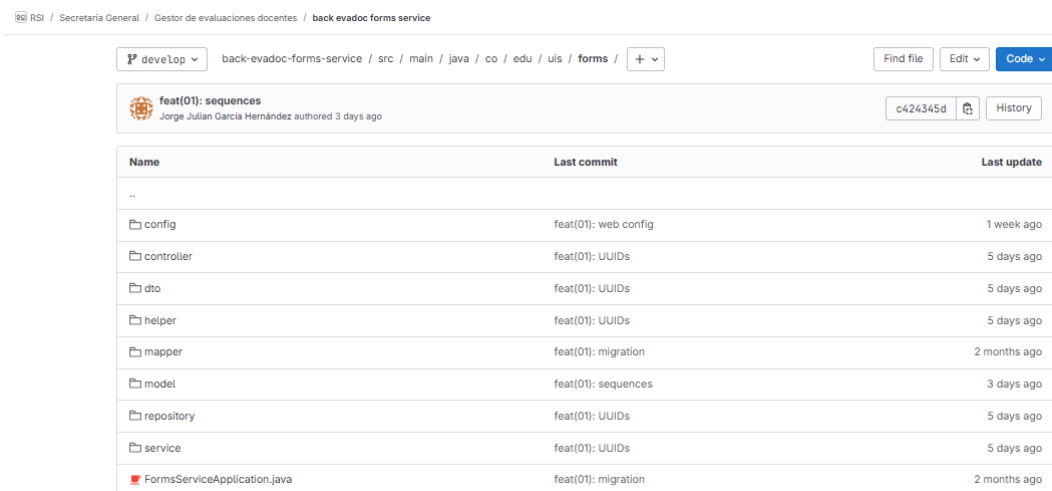
- 75 Commits
- 4 Branches
- 19 Tags
- 19 Releases
- 1 Environment

README
CI/CD configuration
Add Wiki

Created on
September 22, 2025

Figura 26

Organización de paquetes src del form service



RSI / Secretaría General / Gestor de evaluaciones docentes / back-evadoc-forms-service

back-evadoc-forms-service / src / main / java / co / edu / uis / forms

feat(01): sequences
 Jorge Julian Garcia Hernández authored 3 days ago

Name	Last commit	Last update
..		
config	feat(01): web config	1 week ago
controller	feat(01): UUIDs	5 days ago
dto	feat(01): UUIDs	5 days ago
helper	feat(01): UUIDs	5 days ago
mapper	feat(01): migration	2 months ago
model	feat(01): sequences	3 days ago
repository	feat(01): UUIDs	5 days ago
service	feat(01): UUIDs	5 days ago
FormsServiceApplication.java	feat(01): migration	2 months ago

Las siguientes imágenes muestran el historial de Merge Requests por microservicio:

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 27

Merge requests academic service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc academic service / Merge requests

Open 1 Merged 27 Closed 1 All 29

Bulk edit New merge request

Search or filter results... Created date

feat(01): security config 0 of 6 checklist items completed !29 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): logs, env vars 0 of 6 checklist items completed !28 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): sonar smells 0 of 6 checklist items completed !27 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): global handler 0 of 6 checklist items completed !26 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): Readme 0 of 6 checklist items completed !25 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
Feature/01 deploy 0 of 6 checklist items completed !24 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): Javadoc, RSI exceptions, Swagger 0 of 6 checklist items completed !23 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): Javadoc, RSI exceptions, Swagger 0 of 6 checklist items completed !22 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago

Figura 28

Merge requests answer service

Open 1 Merged 15 Closed 1 All 17

Bulk edit New merge request

Search or filter results... Created date

feat(01): security config 0 of 6 checklist items completed !17 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
Feature/01 deploy 0 of 6 checklist items completed !16 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
Feature/01 deploy 0 of 6 checklist items completed !15 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): Readme 1 of 6 checklist items completed !14 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): Javadoc, RSI exceptions, Swagger 0 of 6 checklist items completed !13 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): Javadoc, RSI exceptions, Swagger 0 of 6 checklist items completed !12 - created 1 week ago by Jorge Julian Garcia Hernández	Merged updated 1 week ago
feat(01): pom proyect name 0 of 6 checklist items completed !11 - created 3 weeks ago by Jorge Julian Garcia Hernández	Merged updated 2 weeks ago
feat(01): librerías 0 of 6 checklist items completed !10 - created 3 weeks ago by Jorge Julian Garcia Hernández	Merged updated 3 weeks ago

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 29

Merge requests assessment service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc assessment service / Merge requests

Open 1 Merged 22 Closed 1 All 24 Bulk edit New merge request ⋮

🔍 Search or filter results... Q Created date ↓ ⌵

feat(01): sonar smells 0 of 6 checklist items completed !24 - created 2 days ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 2 days ago
feat(01): uuidGenerator 0 of 6 checklist items completed !23 - created 4 days ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 4 days ago
feat(01): entities fix 0 of 6 checklist items completed !22 - created 4 days ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 4 days ago
feat(01): models sync 0 of 6 checklist items completed !21 - created 6 days ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 6 days ago
feat(01): security config 0 of 6 checklist items completed !20 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): logs, env vars 0 of 6 checklist items completed !19 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): sonar smell 0 of 6 checklist items completed !18 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): position tests 0 of 6 checklist items completed !17 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago

Figura 30

Merge requests form service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc forms service / Merge requests

Open 1 Merged 23 Closed 1 All 25 Bulk edit New merge request ⋮

🔍 Search or filter results... Q Created date ↓ ⌵

feat(01): sequences 0 of 6 checklist items completed !25 - created 3 days ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 3 days ago
Feature/01 deploy 0 of 6 checklist items completed !24 - created 4 days ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 4 days ago
feat(01): web config 0 of 6 checklist items completed !23 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): security config 0 of 6 checklist items completed !22 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): security config 0 of 6 checklist items completed !21 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): logs, env vars 0 of 6 checklist items completed !20 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
Feature/01 deploy 0 of 6 checklist items completed !19 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago
feat(01): force sync 0 of 6 checklist items completed !18 - created 1 week ago by Jorge Julian Garcia Hernández	Merged ! ⊗ updated 1 week ago

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

5.3.3 Resultados de Implementación del Backend

Los cuatro microservicios del sistema EVADOC exponen sus funcionalidades a través de APIs REST documentadas con Swagger/OpenAPI. A continuación se presenta el inventario de endpoints implementados por cada servicio en la Fase 1.

Academic Service: implementa 37 endpoints distribuidos en 6 controladores REST, cubriendo las entidades del dominio académico necesarias para la construcción de los procesos de evaluación.

Tabla 15

Distribución de Endpoints por Controlador Academic Service

Controlador	Descripción	Endpoints
Courses	Gestión de cursos académicos	6
Teachers	Gestión de docentes	5
Education Cycles	Gestión de ciclos educativos	7
Generic Users	Gestión de usuarios genéricos	5
Students	Gestión de estudiantes	7
Organizational Units	Gestión de unidades organizacionales	7
Total		37

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 31*Endpoints courses y teachers del academic service*

Servers
http://10.0.44.19:30500/academic-service - Generated server url

Courses Gestión de cursos académicos ^

GET	/api/academic/courses/{id}	Obtener curso por ID
PUT	/api/academic/courses/{id}	Actualizar curso
DELETE	/api/academic/courses/{id}	Eliminar curso
GET	/api/academic/courses	Obtener todos los cursos
POST	/api/academic/courses	Crear curso
GET	/api/academic/courses/filter	Filtrar cursos

Teachers Gestión de docentes ^

GET	/api/academic/teachers/{id}	Obtener docente por ID
PUT	/api/academic/teachers/{id}	Actualizar docente
DELETE	/api/academic/teachers/{id}	Eliminar docente
GET	/api/academic/teachers	Obtener todos los docentes
POST	/api/academic/teachers	Crear docente

Figura 32*Endpoints education cycles y generic users del academic service*

Education Cycles Gestión de ciclos educativos ^

GET	/api/academic/education-cycles/{id}	Obtener ciclo educativo por ID
PUT	/api/academic/education-cycles/{id}	Actualizar ciclo educativo
DELETE	/api/academic/education-cycles/{id}	Eliminar ciclo educativo
GET	/api/academic/education-cycles	Obtener todos los ciclos educativos
POST	/api/academic/education-cycles	Crear ciclo educativo
GET	/api/academic/education-cycles/code/{code}	Obtener ciclo educativo por código
GET	/api/academic/education-cycles/active	Obtener ciclos educativos activos

Generic Users Gestión de usuarios genéricos ^

GET	/api/academic/generic-users/{id}	Obtener usuario por ID
PUT	/api/academic/generic-users/{id}	Actualizar usuario genérico
DELETE	/api/academic/generic-users/{id}	Eliminar usuario genérico
GET	/api/academic/generic-users	Obtener todos los usuarios
POST	/api/academic/generic-users	Crear usuario genérico

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 33*Endpoints students y organizational units del academic service*

Students <small>Gestión de estudiantes</small>	
GET	/api/academic/students/{id} Obtener estudiante por ID
PUT	/api/academic/students/{id} Actualizar estudiante
DELETE	/api/academic/students/{id} Eliminar estudiante
GET	/api/academic/students Obtener todos los estudiantes
POST	/api/academic/students Crear estudiante
GET	/api/academic/students/{id}/courses Obtener cursos de un estudiante
GET	/api/academic/students/code/{code} Obtener estudiante por código
Organizational Units <small>Gestión de unidades organizacionales</small>	
GET	/api/academic/organizational-units/{id} Obtener unidad organizacional por ID
PUT	/api/academic/organizational-units/{id} Actualizar unidad organizacional
DELETE	/api/academic/organizational-units/{id} Eliminar unidad organizacional
GET	/api/academic/organizational-units Obtener todas las unidades organizacionales
POST	/api/academic/organizational-units Crear unidad organizacional
GET	/api/academic/organizational-units/{id}/descendants Obtener descendientes de una unidad organizacional
GET	/api/academic/organizational-units/type/{type} Obtener unidades organizacionales por tipo

Assessment Service: implementa 24 endpoints en 3 controladores, cubriendo la creación y gestión de procesos evaluativos, el seguimiento del progreso por evaluador y el control de bloqueos.

Tabla 16*Distribución de Endpoints por Controlador Assessment Service*

Controlador	Descripción	Endpoints
Assessments	Gestión de evaluaciones	8
Evaluation Progress	Gestión del progreso de evaluaciones	10
Evaluation Locks	Gestión de bloqueos de evaluaciones	6
Total		24

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 34*Endpoints assessments del assessment service*

Server:

Assessments <small>Gestión de evaluaciones</small>	
GET	/api/assessments/{id} Obtener evaluación por ID
PUT	/api/assessments/{id} Actualizar evaluación
GET	/api/assessments Listar evaluaciones
POST	/api/assessments Crear evaluación
POST	/api/assessments/{id}/reprocess
GET	/api/assessments/search Búsqueda paginada de evaluaciones
GET	/api/assessments/evaluator/{evaluatorscontextId} Obtener evaluaciones por evaluador
GET	/api/assessments/evalutee/{evaluteescontextId} Obtener evaluaciones por evaluado

Figura 35*Endpoints evaluation progress y locks del assessment service*

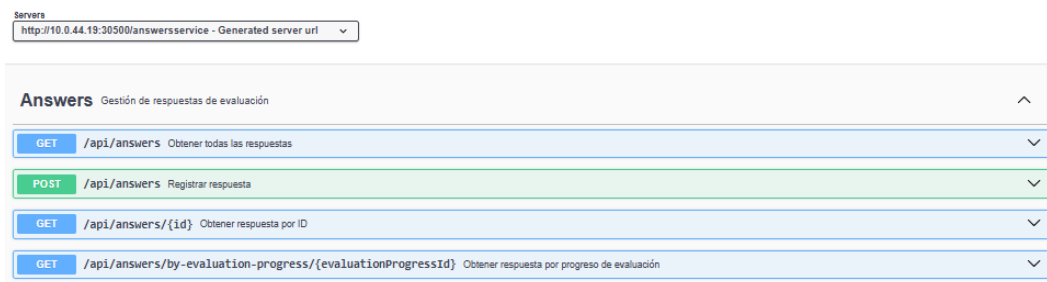
Evaluation Progress <small>Gestión del progreso de evaluaciones</small>	
GET	/api/evaluation-progress/{id} Obtener progreso por ID
PUT	/api/evaluation-progress/{id} Actualizar progreso
DELETE	/api/evaluation-progress/{id} Eliminar progreso
GET	/api/evaluation-progress Listar todos los progresos
POST	/api/evaluation-progress Crear progreso
GET	/api/evaluation-progress/summary/{assessmentId} Obtener resumen de evaluación
GET	/api/evaluation-progress/exists Validar existencia
GET	/api/evaluation-progress/evaluator/{evaluatorId} Buscar por evaluador
GET	/api/evaluation-progress/assessment/{assessmentId} Buscar por evaluación
GET	/api/evaluation-progress/assessment/{assessmentId}/evaluator/{evaluatorId} Buscar por evaluación y evaluador
Evaluation Locks <small>Gestión de bloqueos de evaluaciones</small>	
POST	/api/evaluation-locks Crear lock
DELETE	/api/evaluation-locks Eliminar lock por evaluación y contexto
POST	/api/evaluation-locks/bulk Crear múltiples locks
GET	/api/evaluation-locks/is-locked Verificar si un contexto está bloqueado
GET	/api/evaluation-locks/assessment/{assessmentId} Obtener locks por evaluación
DELETE	/api/evaluation-locks/{lockId} Eliminar lock por ID

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Answer Service: implementa 4 endpoints en un controlador, cubriendo el registro y consulta de respuestas de evaluación.

Figura 36

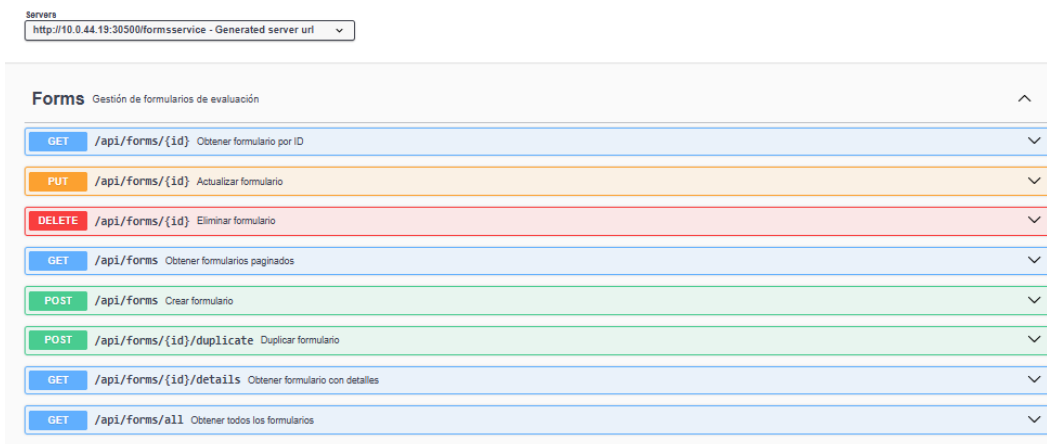
Endpoints answers del answer service



Form Service: implementa 8 endpoints en un controlador, cubriendo el ciclo completo de gestión de formularios de evaluación.

Figura 37

Endpoints forms del form service



En total, la Fase 1 del sistema EVADOC implementa 73 endpoints REST distribuidos en los cuatro microservicios del backend.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

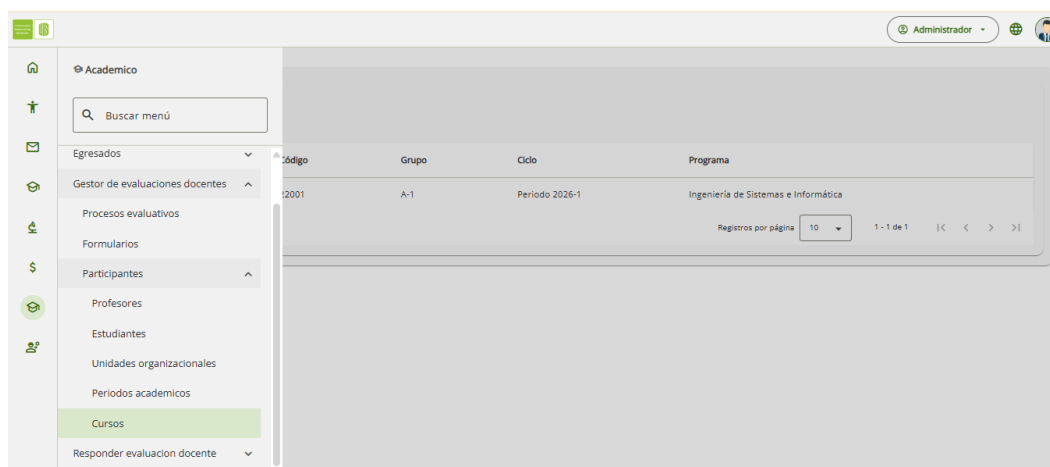
5.3.4 Resultados de Implementación del Frontend

El sistema EVADOC cuenta con dos interfaces de usuario desarrolladas en Angular: el Manager UI, orientado al administrador, y el Catchment UI, orientado al estudiante evaluador.

Manager UI: centraliza las funciones administrativas del sistema. En la Fase 1 se implementaron los módulos de gestión de participantes (docentes, estudiantes, unidades organizacionales, periodos académicos y cursos), gestión de formularios dinámicos y configuración de procesos evaluativos.

Figura 38

Vista principal del manager UI con menú de navegación



REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 39*Vista de periodos académicos*

Administrador

Períodos Académicos

Código	Nombre	Periodo	Año	Fechas	Activo
2026-1	Periodo 2026-1	1	2026	Jan 15, 2026 - Jun 15, 2026	✓

Registros por página 10 1 - 10 de 0 |< < > >|

RSI/AV 2.0

Figura 40*Vista de cursos*

Administrador

Cursos

Buscar por nombre 🔍

Nombre	Código	Grupo	Ciclo	Programa
Programación I	22001	A-1	Periodo 2026-1	Ingeniería de Sistemas e Informática

Registros por página 10 1 - 1 de 1 |< < > >|

RSI/AV 2.0

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 41*Vista de docentes*

Docentes

Buscar por nombre o ...

Nombre	Email	Contrato
Carlos Manuel	carlos@uis.edu.co	PLANTA

Registros por página 5 1 - 1 de 1 |< < > >|

RSIEV 2.0

Figura 42*Vista de estudiantes*

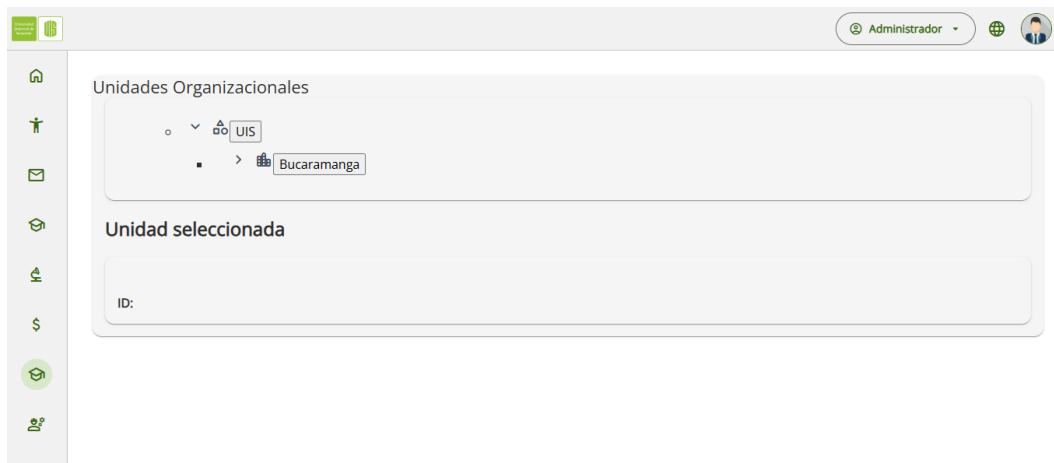
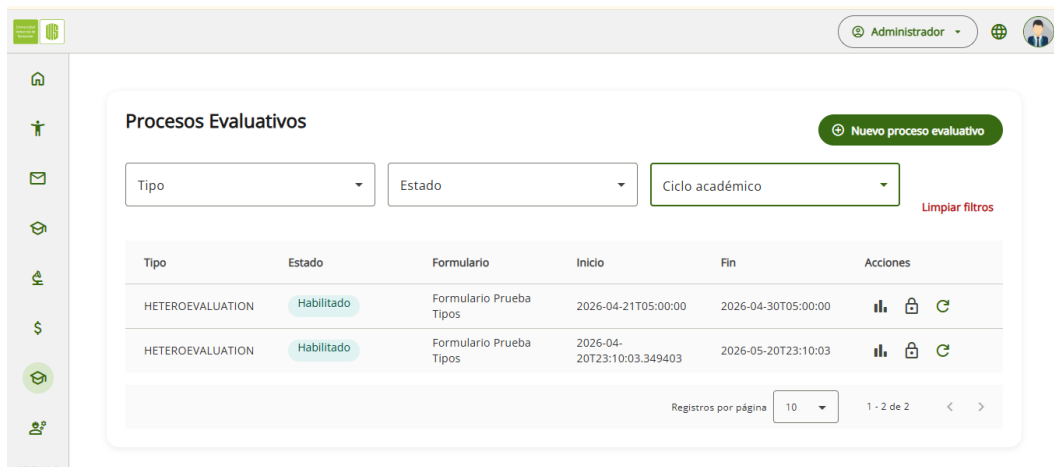
Estudiantes

Buscar estudiante

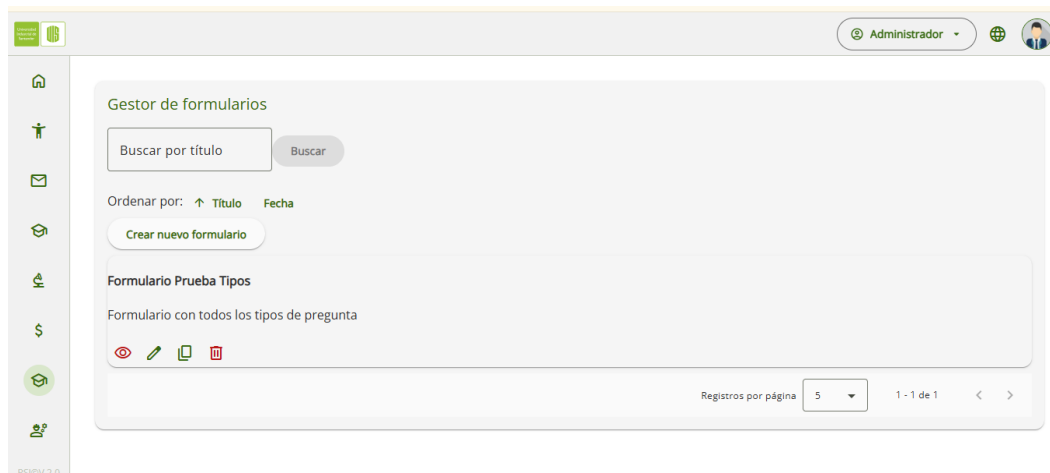
Código	Nombre	Email	Modalidad	Acciones
COD1	Estudiante 1	est1@mail.com	PRESENCIAL	
COD2	Estudiante 2	est2@mail.com	PRESENCIAL	
COD3	Estudiante 3	est3@mail.com	PRESENCIAL	
COD4	Estudiante 4	est4@mail.com	PRESENCIAL	
COD5	Estudiante 5	est5@mail.com	PRESENCIAL	

Registros por página 5 1 - 5 de 21 |< < > >|

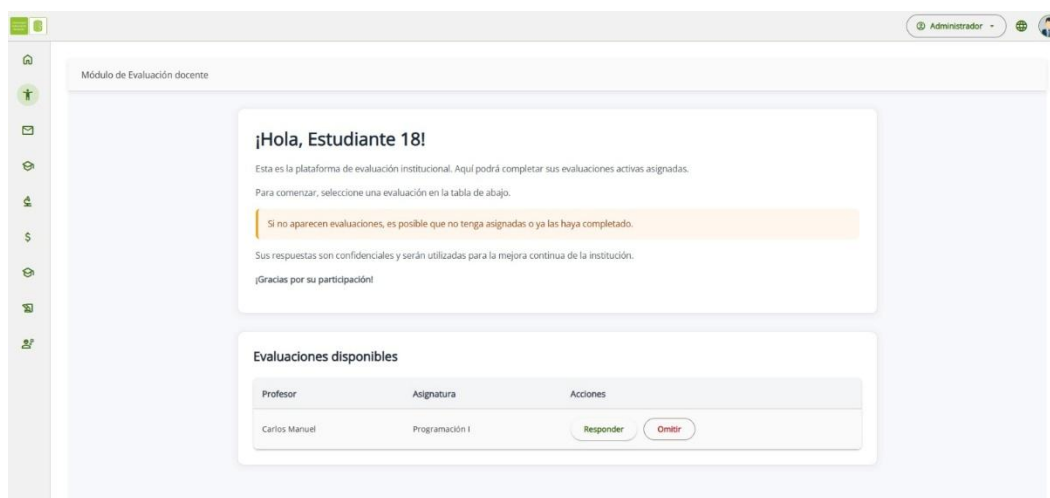
RSIEV 2.0

Figura 43*Vista de unidades organizacionales***Figura 44***Vista de proceso evaluativo*

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 45*Vista de gestión de formularios*

Catchment UI: es la interfaz de captación de respuestas utilizada por el estudiante durante el proceso de evaluación. En la Fase 1 se implementaron la consulta de evaluaciones disponibles, el diligenciamiento del formulario y la opción de omisión con justificación.

Figura 46*Vista de evaluaciones disponibles para el estudiante*

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 47*Vista del formulario de evaluación en proceso de diligenciamiento*

The screenshot shows a web interface for an evaluation form. At the top right, there is a user profile icon and the text 'Administrador'. The main content area is titled 'Formulario Prueba Tipos' and 'Formulario con todos los tipos de pregunta'. Below this, there are three main sections:

- 'Califica el curso de 1 a 5 *' with a 'Respuesta numérica' input field.
- 'Describe tu experiencia en el curso *' with a larger 'Respuesta' text area.
- '¿Qué aspectos te gustaron?' with three checkboxes: 'Contenido', 'Docente', and 'Metodología'.

 A left sidebar contains navigation icons for home, search, messages, courses, reports, settings, and user profile.

Figura 48*Vista de omisión de evaluación con justificación*

The screenshot shows a web interface for the 'Módulo de Evaluación docente'. A modal dialog box titled 'Omitir evaluación' is open in the center. The dialog contains the following text:

- 'Está a punto de omitir la evaluación de Carlos Manuel en Programación I.'
- 'Seleccione una justificación:'
- A list of radio button options:
 - No cursé esta asignatura durante el periodo evaluado
 - Cancele la asignatura antes de finalizar el curso
 - El docente listado no fue quien impartió la asignatura
 - No tuve interacción académica suficiente para evaluar
 - La asignatura no se desarrolló con normalidad
 - Otra
- Buttons for 'Cancelar' and 'Omitir' at the bottom.

 The background interface shows a table of 'Evaluaciones disponibles' with columns for 'Profesor' and 'Asignatura'. One entry is visible: 'Carlos Manuel' for 'Programación I'.

5.3.5 Resultado de Despliegue

El despliegue del sistema EVADOC se realizó mediante contenedores Docker, siguiendo los lineamientos de infraestructura de la DTIC. Cada repositorio cuenta con su propio Dockerfile y dockerfile.local para los ambientes de producción y desarrollo local respectivamente. La automatización del proceso de construcción y despliegue se configuró mediante el archivo .gitlab-

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

ci.yml de cada repositorio, integrando las etapas de compilación, análisis de calidad y despliegue dentro del pipeline CI/CD de GitLab.

La evidencia del despliegue exitoso se encuentra en la sección Environments de cada repositorio en GitLab, donde los seis componentes del sistema (cuatro microservicios del backend y dos interfaces del frontend) registran estado Success con el tag Latest Deployed en el ambiente de desarrollo.

Figura 49

Environments academic service: develop success

The screenshot shows the GitLab Environments page for the 'academic service' project. The breadcrumb trail is: RSI / Secretaria General / Gestor de evaluaciones docentes / back evadoc academic service / Environments. At the top, there are tabs for 'Active 4' and 'Stopped 4', and a 'New environment' button. A search bar is present with the placeholder 'Search by environment name'. Below this is a table with columns 'Name', 'Deployments', and 'Actions'.

Name	Deployments	Actions
develop	<ul style="list-style-type: none"> Success Latest Deployed Merge branch 'feature/01-deploy' into 'develop' -> bd157748 Triggered by @jjgarciah on April 22, 2026 at 2:40:59 AM GMT-5 	Open
predev	There are no deployments for this environment yet. Learn more about setting up deployments.	
preprod	There are no deployments for this environment yet. Learn more about setting up deployments.	
production	There are no deployments for this environment yet. Learn more about setting up deployments.	

Figura 50

Environments answer service: develop success

The screenshot shows the GitLab Environments page for the 'answer service' project. The breadcrumb trail is: RSI / Secretaria General / Gestor de evaluaciones docentes / back evadoc answers service / Environments. At the top, there are tabs for 'Active 1' and 'Stopped 2', and a 'New environment' button. A search bar is present with the placeholder 'Search by environment name'. Below this is a table with columns 'Name', 'Deployments', and 'Actions'.

Name	Deployments	Actions
develop	<ul style="list-style-type: none"> Success Latest Deployed Merge branch 'feature/01-deploy' into 'develop' -> d35b6c6f Triggered by @jjgarciah on April 22, 2026 at 12:32:18 PM GMT-5 	Open

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 51*Environments assessment service: develop success*

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc assessment service / Environments

Active 1 Stopped 2 New environment

Q Search by environment name

Name	Deployments	Actions
develop	<ul style="list-style-type: none"> Success Latest Deployed <ul style="list-style-type: none"> Merge branch 'feature/01-deploy' into 'develop' -> 227bee1 Triggered by @jjgarciah on April 22, 2026 at 12:09:58 AM GMT-5 	Open

Figura 52*Environments form service: develop success*

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc forms service / Environments

Active 1 Stopped 2 New environment

Q Search by environment name

Name	Deployments	Actions
develop	<ul style="list-style-type: none"> Success Latest Deployed <ul style="list-style-type: none"> Merge branch 'feature/01-deploy' into 'develop' -> 7b52e3b6 Triggered by @jjgarciah on April 18, 2026 at 7:12:18 PM GMT-5 	Open

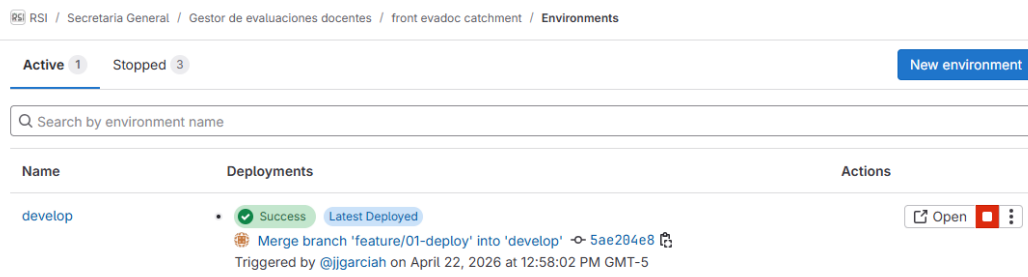
Figura 53*Environments manager UI: develop success*

RSI / Secretaría General / Gestor de evaluaciones docentes / front evadoc manager / Environments

Active 1 Stopped 4 New environment

Q Search by environment name

Name	Deployments	Actions
develop	<ul style="list-style-type: none"> Success Latest Deployed <ul style="list-style-type: none"> Merge branch 'feature/01-deploy' into 'develop' -> 4e57d5c8 Triggered by @jjgarciah on April 22, 2026 at 3:44:43 AM GMT-5 	Open

Figura 54*Environments catchment UI: develop success***5.4 Pruebas y Análisis de Resultados****5.4.1 Pruebas Unitarias e Integración del Backend**

Las pruebas unitarias fueron desarrolladas para cada microservicio utilizando el framework de pruebas de Spring Boot (JUnit 5 y Mockito). El Academic Service alcanzó una cobertura de código del 90.50%, evidenciada en el badge de cobertura del repositorio GitLab. Los resultados de calidad del código fueron monitoreados continuamente mediante SonarQube, cuya integración con el pipeline CI/CD permitió detectar y resolver de forma temprana los Code Smells identificados en cada microservicio. Al cierre de la Fase 1, los cuatro microservicios obtuvieron Quality Gate Passed con 0 bugs, 0 vulnerabilidades y 0 Code Smells activos.

La siguiente tabla consolida los resultados de pruebas unitarias por microservicio, obtenidos desde SonarQube (Measures - Coverage). El detalle completo por servicio se presenta en el Anexo D.

Tabla 17*Resultados de Pruebas Unitarias — Fase 1 EVADOC*

Microservicio	Unit Tests	Cobertura	Errores	Fallos	Omitidas	Resultado
Academic Service	94	90.1%	0	0	0	100% Success
Answer Service	17	92.9%	0	0	0	100% Success
Assessment Service	87	97.0%	0	0	0	100% Success
Form Service	26	91.5%	0	0	0	100% Success
Total	224		0	0	0	100% Success

5.4.2 Pruebas de Flujo sobre el Sistema Integrado

Las pruebas de flujo verificaron el correcto funcionamiento del sistema de extremo a extremo, validando la comunicación entre los microservicios a través de RabbitMQ y las llamadas REST entre servicios. Los pipelines CI/CD de GitLab ejecutaron automáticamente las etapas de compilación, análisis y pruebas de integración en cada Merge Request, generando evidencia del proceso de verificación continua. A lo largo del desarrollo se registraron pipelines con estado Failed, Blocked y Warning que identificaron problemas resueltos en iteraciones posteriores, hasta alcanzar el estado Passed en todos los servicios.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 57

Historial de pipelines assessment service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc assessment service / Pipelines

All 79 Finished Branches Tags [View analytics](#) [New pipeline](#)

Filter pipelines

Status	Pipeline	Created by	Stages	Actions
Passed 00:04:01 2 days ago	Merge branch 'feature/01-deploy' into 'devel...' #91842 latest branch			
Warning 00:05:59 2 days ago	feat(01): sonar smells #91841 latest merge request			
Passed 00:03:43 4 days ago	Merge branch 'feature/01-deploy' into 'devel...' #91745 branch			
Warning 00:05:10 4 days ago	feat(01): uuidGenerator #91744 latest merge request			
Passed 00:03:54 4 days ago	Merge branch 'feature/01-deploy' into 'devel...' #91741 branch			

Figura 58

Historial de pipelines form service

RSI / Secretaría General / Gestor de evaluaciones docentes / back evadoc forms service / Pipelines

Status	Pipeline	Created by	Stages	Actions
Passed 00:03:55 3 days ago	Merge branch 'feature/01-deploy' into 'devel...' #91837 latest branch			
Warning 00:05:28 3 days ago	feat(01): sequences #91836 latest merge request			
Passed 00:03:52 4 days ago	Merge branch 'feature/01-deploy' into 'devel...' #91748 branch			
Warning 00:05:25 4 days ago	feat(01): UUIDs #91747 latest merge request			
Passed 00:03:47 1 week ago	Merge branch 'feature/01-deploy' into 'devel...' #91478 branch			
Warning 00:05:06 1 week ago	feat(01): web config #91477 latest merge request			

5.4.3 Casos de Prueba

Con el sistema implementado y los requerimientos funcionales validados a través de las historias de usuario, se procedió al diseño y ejecución de los casos de prueba del sistema EVADOC. Los casos de prueba constituyen el mecanismo formal mediante el cual se verifica que cada funcionalidad implementada cumple con los criterios de aceptación definidos en las historias de usuario correspondientes. Para la Fase 1 del proyecto se diseñaron treinta casos de prueba distribuidos en las siete épicas del sistema, de los cuales catorce corresponden a las ocho historias de usuario desarrolladas en esta fase y obtuvieron resultado aprobado tras su ejecución sobre el prototipo funcional. Los casos de prueba restantes quedan documentados para su ejecución en fases posteriores del proyecto. A continuación, se presenta la tabla resumen con el identificador de cada caso de prueba, la historia de usuario relacionada, la descripción y el estado obtenido. El detalle completo de cada caso de prueba se encuentra en los anexos del presente documento.

Tabla 18

Resumen de Casos de Prueba del Sistema EVADOC

ID	HU	TITULO	ESTADO
CP-01	HU-01	Carga exitosa de información académica del periodo	Aprobado
CP-02	HU-01	Notificación de fallo al ocurrir un error en la carga	Aprobado
CP-03	HU-02	Consulta exitosa de información académica filtrada	Pendiente para próxima fase del proyecto
CP-04	HU-03	Creación exitosa de proceso de evaluación con datos válidos	Aprobado
CP-05	HU-03	Rechazo de proceso con fechas incoherentes	Aprobado
CP-06	HU-04	Habilitación exitosa del proceso para todos los participantes	Aprobado

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

CP-07	HU-05	Cierre parcial exitoso del proceso para una unidad académica	Aprobado
CP-08	HU-05	Los resultados de una unidad quedan disponibles al registrar su cierre parcial	Aprobado
CP-09	HU-06	Estudiante recibe notificación automática al habilitarse el proceso	Pendiente para próxima fase del proyecto
CP-10	HU-06	Estudiante accede al sistema directamente desde la notificación	Pendiente para próxima fase del proyecto
CP-11	HU-07	Lista correcta de profesores pendientes para el estudiante	Aprobado
CP-12	HU-07	Evaluación completada desaparece de la lista de pendientes	Aprobado
CP-13	HU-08	Estudiante completa formulario y la respuesta queda registrada	Aprobado
CP-14	HU-09	Omisión de evaluación registrando justificación	Aprobado
CP-15	HU-09	El sistema impide omitir sin registrar justificación	Aprobado
CP-16	HU-10	Docente accede y completa su formulario de autoevaluación	Pendiente para próxima fase del proyecto
CP-17	HU-10	Formulario de autoevaluación no disponible con proceso inactivo	Pendiente para próxima fase del proyecto
CP-18	HU-11	Docente consulta estado de autoevaluación completada	Pendiente para próxima fase del proyecto
CP-19	HU-12	Solicitud de ajuste registrada y notificada al administrador	Pendiente para próxima fase del proyecto
CP-20	HU-13	Administrador aprueba una solicitud de ajuste	Pendiente para próxima fase del proyecto
CP-21	HU-13	Administrador rechaza una solicitud de ajuste	Pendiente para próxima fase del proyecto
CP-22	HU-14	Consulta de historial de solicitudes filtrado por periodo	Pendiente para próxima fase del proyecto
CP-23	HU-15	Administrador consulta reporte por unidad organizacional desde el Dashboard	Pendiente para próxima fase del proyecto

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

CP-24	HU-16	Reporte individual por docente disponible en el Dashboard para roles autorizados	Pendiente para próxima fase del proyecto
CP-25	HU-17	Docente accede únicamente a sus propios resultados según su rol	Pendiente para próxima fase del proyecto
CP-26	HU-17	Directivo visualiza solo los resultados de su unidad organizacional	Pendiente para próxima fase del proyecto
CP-27	HU-18	Administrador consulta reporte histórico de semestre anterior	Pendiente para próxima fase del proyecto
CP-28	HU-19	Usuario externo accede a informes generales sin autenticación	Pendiente para próxima fase del proyecto
CP-29	HU-20	Administrador crea formulario con secciones y preguntas	Aprobado
CP-30	HU-20	El sistema impide eliminar un formulario en uso	Aprobado

Nota: Resumen de los casos de prueba del sistema EVADOC, su relación con las historias de usuario y el estado de ejecución. Detalle completo en anexos.

5.4.4 Pruebas Funcionales

La presente sección documenta los resultados obtenidos tras la ejecución de los casos de prueba sobre el sistema EVADOC implementado en la Fase 1 del proyecto. Las pruebas fueron realizadas directamente sobre el prototipo funcional, verificando el cumplimiento de los criterios de aceptación definidos para cada historia de usuario. Se incluyen únicamente las historias de usuario correspondientes a la Fase 1: HU-01, HU-03, HU-04, HU-05, HU-07, HU-08, HU-09 y HU-20.

HU-01 Cargar información académica del periodo: Se ejecutaron los casos de prueba CP-01 y CP-02 sobre el sistema implementado. CP-01 verificó la carga exitosa de información académica del periodo, confirmando que el sistema integra correctamente los datos de docentes, estudiantes y asignaturas provenientes del Academic Service, genera la notificación de carga

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

completada al administrador y deja los datos disponibles para la configuración del proceso evaluativo. CP-02 verificó el comportamiento del sistema ante un fallo en el servicio de carga, confirmando que el sistema detiene el proceso, notifica al administrador con el detalle del error y preserva la integridad de los datos previamente cargados. Ambos casos de prueba obtuvieron resultado aprobado, evidenciando el cumplimiento total de los criterios de aceptación de esta historia de usuario.

HU-03 Crear proceso de evaluación docente: Se ejecutaron los casos de prueba CP-04 y CP-05. CP-04 verificó la creación exitosa de un proceso de evaluación con datos válidos, confirmando que el sistema permite configurar el tipo de evaluación, el periodo académico, las fechas, las unidades organizacionales, los tipos de evaluador y evaluado, y el formulario asociado, validando la coherencia de los datos antes de guardar y dejando el proceso disponible para ser habilitado. CP-05 verificó el rechazo de un proceso con fechas incoherentes, confirmando que el sistema muestra el mensaje de validación correspondiente sin guardar el proceso. Ambos casos de prueba obtuvieron resultado aprobado.

HU-04 Habilitar proceso de evaluación: Se ejecutó el caso de prueba CP-06, que verificó la habilitación exitosa del proceso de evaluación para todos los participantes de forma simultánea. Se confirmó que el proceso cambia correctamente a estado "activo" y que los estudiantes pueden acceder al formulario mediante sus credenciales institucionales. El caso de prueba obtuvo resultado aprobado.

HU-05 Gestionar cierre parcial del proceso por unidad académica: Se ejecutaron los casos de prueba CP-07 y CP-08. CP-07 verificó el cierre parcial exitoso del proceso para una unidad académica específica, confirmando que el cierre aplica únicamente a la unidad seleccionada sin afectar las demás unidades activas y que el sistema genera la confirmación correspondiente al

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

administrador. CP-08 verificó que al registrar el cierre parcial de una unidad, los resultados de esa unidad quedan disponibles para su visualización sin interrumpir el proceso de las demás unidades. Ambos casos de prueba obtuvieron resultado aprobado.

HU-07 Consultar profesores pendientes por evaluar: Se ejecutaron los casos de prueba CP-11 y CP-12. CP-11 verificó que el sistema muestra correctamente la lista de profesores asociados a las asignaturas del estudiante en el periodo activo, todos en estado pendiente de evaluación. CP-12 verificó que al completar y enviar la evaluación de un profesor, este desaparece de la lista de pendientes, y que al finalizar todas las evaluaciones el sistema muestra el mensaje de confirmación correspondiente. Ambos casos de prueba obtuvieron resultado aprobado.

HU-08 Diligenciar formulario de evaluación docente: Se ejecutó el caso de prueba CP-13, que verificó el diligenciamiento exitoso del formulario de evaluación. Se confirmó que el formulario carga correctamente desde el Form Service, que las respuestas quedan registradas, que el sistema muestra el mensaje de confirmación de completitud y que el profesor evaluado desaparece de la lista de pendientes. El caso de prueba obtuvo resultado aprobado.

HU-09 Omitir evaluación con justificación: Se ejecutaron los casos de prueba CP-14 y CP-15. CP-14 verificó que el estudiante puede omitir la evaluación de un profesor seleccionando una justificación del listado desplegable, confirmando que la omisión queda registrada con fecha y justificación y que el profesor desaparece de la lista de pendientes. CP-15 verificó que el sistema impide confirmar la omisión cuando no se ha seleccionado una justificación, mostrando el mensaje de campo obligatorio correspondiente. Ambos casos de prueba obtuvieron resultado aprobado.

HU-20 Gestionar formularios dinámicos: Se ejecutaron los casos de prueba CP-29 y CP-30. CP-29 verificó la creación exitosa de un formulario dinámico con secciones, preguntas y

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

opciones de respuesta desde el Manager UI, confirmando que el formulario queda disponible para ser asociado a un proceso de evaluación. CP-30 verificó que el sistema impide eliminar un formulario que se encuentra asociado a un proceso activo, mostrando el mensaje de restricción correspondiente. Ambos casos de prueba obtuvieron resultado aprobado.

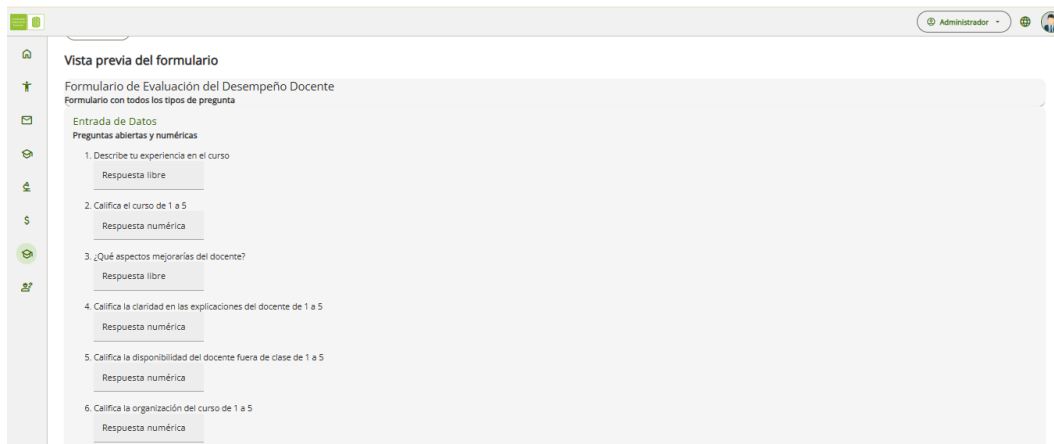
Nota: Los casos de prueba correspondientes a las historias de usuario de fases posteriores (CP-03, CP-09, CP-10, CP-16 a CP-28) quedan pendientes para su ejecución en la siguiente fase del proyecto.

5.4.5 Validación del Sistema con Muestra de Población

Con el fin de validar el funcionamiento integral del sistema EVADOC en condiciones reales de uso, se ejecutó una prueba de flujo completo utilizando una muestra de 21 evaluadores sobre el docente del curso Programación I, correspondiente al periodo académico 2026-1. Para esta validación se creó el formulario "Formulario de Evaluación del Desempeño Docente", compuesto por 12 preguntas distribuidas en dos secciones: entrada de datos (preguntas abiertas y numéricas) y selección (preguntas con opciones de respuesta). Las siguientes imágenes muestran el formulario utilizado en la validación, tal como fue configurado desde el Manager UI:

Figura 59

Formulario implementado, sección entrada de datos



Vista previa del formulario

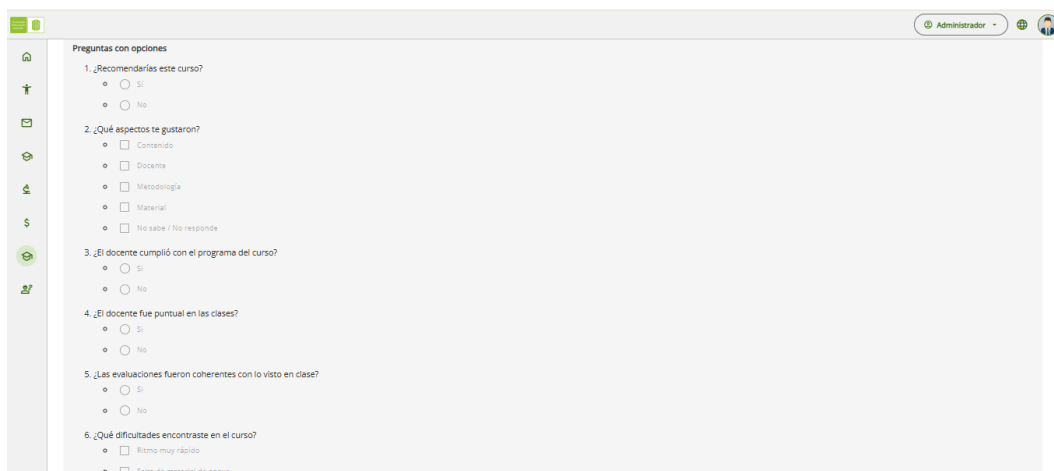
Formulario de Evaluación del Desempeño Docente
Formulario con todos los tipos de pregunta

Entrada de Datos
Preguntas abiertas y numéricas

1. Describe tu experiencia en el curso
Respuesta libre
2. Califica el curso de 1 a 5
Respuesta numérica
3. ¿Qué aspectos mejorarías del docente?
Respuesta libre
4. Califica la claridad en las explicaciones del docente de 1 a 5
Respuesta numérica
5. Califica la disponibilidad del docente fuera de clase de 1 a 5
Respuesta numérica
6. Califica la organización del curso de 1 a 5
Respuesta numérica

Figura 60

Formulario implementado, sección selección con opciones



Preguntas con opciones

1. ¿Recomendarías este curso?
 Sí
 No
2. ¿Qué aspectos te gustaron?
 Contenido
 Docente
 Metodología
 Material
 No sabe / No responde
3. ¿El docente cumplió con el programa del curso?
 Sí
 No
4. ¿El docente fue puntual en las clases?
 Sí
 No
5. ¿Las evaluaciones fueron coherentes con lo visto en clase?
 Sí
 No
6. ¿Qué dificultades encontraste en el curso?
 Ritmo muy rápido
 Falta de material de apoyo

Una vez habilitado el proceso evaluativo, los 21 evaluadores accedieron al sistema a través del Catchment UI y diligenciaron el formulario. El sistema registró el progreso en tiempo real a través del Assessment Service, obteniendo el siguiente resultado:

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 61*Progreso de evaluaciones*

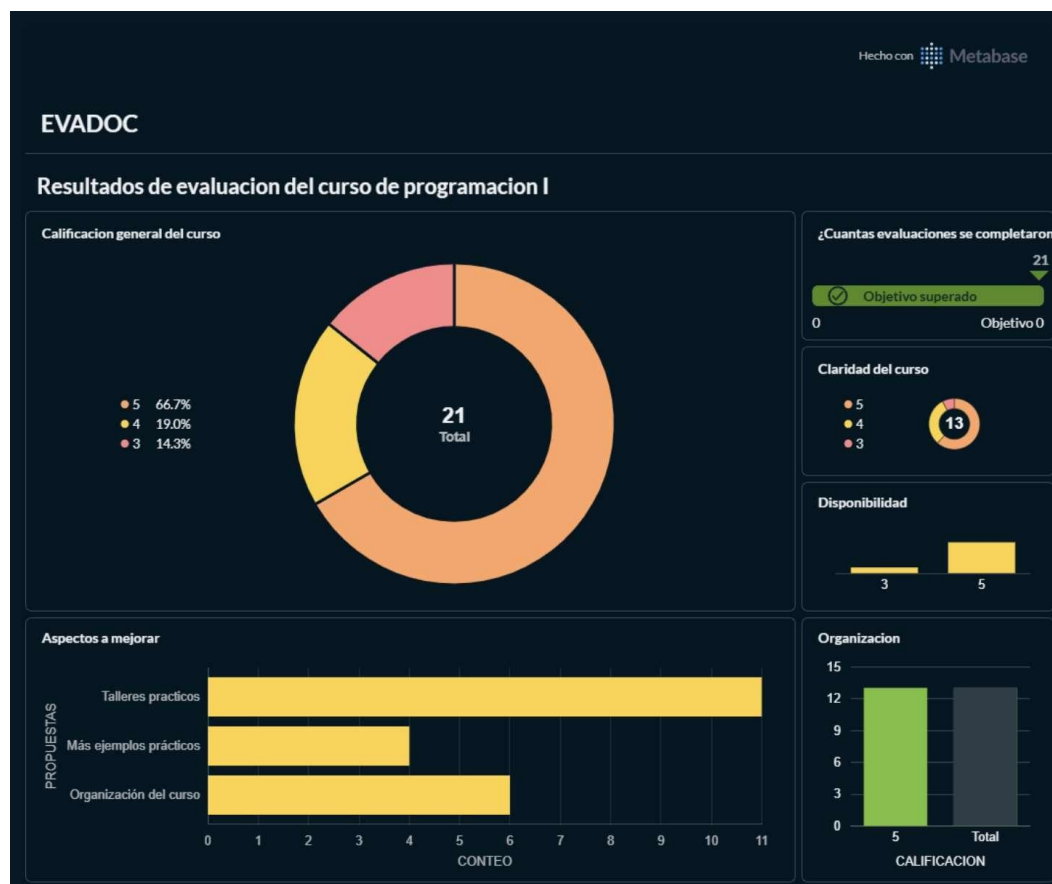
Evaluador	Evaluado	Estado
4F16c45-0335-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-0338-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-0330-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-033e-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-0341-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-032f-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-0332-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-0334-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-0337-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada
4F16c45-033a-0c29-e064-00144f8b01a1	4F16c45-032d-0c29-e064-00144f8b01a1	Completada

Resumen: Total: 21, Completadas: 21, Omitidas: 0, Pendientes: 0

Registros por página: 10, 1 - 10 de 21

El 100% de las evaluaciones asignadas fueron completadas exitosamente, lo que confirma el correcto funcionamiento del flujo evaluativo de extremo a extremo.

Finalmente, las respuestas recopiladas fueron visualizadas a través del Dashboard de Metabase, donde se proyectaron los resultados consolidados de la evaluación del curso Programación I.

Figura 62*Dashboard metabase EVADOC — resultados de evaluación*

Esta validación demuestra que el sistema EVADOC cumple de manera satisfactoria con el flujo completo de evaluación docente: creación del formulario, habilitación del proceso, captación de respuestas y visualización de resultados, integrando de forma coherente los cuatro microservicios implementados en la Fase 1.

5.5 Errores Detectados y Resultados

Durante el desarrollo de la Fase 1 del sistema EVADOC, el proceso de implementación fue acompañado de un monitoreo continuo de calidad mediante dos mecanismos complementarios: el análisis estático de código a través de SonarQube y la ejecución automatizada de pipelines CI/CD

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

en GitLab. Ambas herramientas permitieron identificar errores de forma temprana y garantizar que cada microservicio alcanzara los estándares de calidad requeridos antes de su integración a la rama principal de desarrollo.

5.5.1 Detección mediante SonarQube

Los análisis de SonarQube sobre los cuatro microservicios identificaron principalmente Code Smells problemas de mantenibilidad que representan deuda técnica. En todos los servicios se registraron MRs denominados sonar smells como evidencia de las correcciones aplicadas. El detalle de los reportes por microservicio se presenta en el Anexo D.

5.5.2 Detección mediante Pipelines CI/CD

Los pipelines de GitLab ejecutaron automáticamente las etapas de compilación, pruebas y análisis de calidad en cada MR. A lo largo del desarrollo se registraron pipelines con estado Failed, Blocked y Warning que señalaron problemas resueltos en iteraciones posteriores. El detalle por microservicio se presenta en el Anexo D.

5.5.3 Resultado del Proceso de Corrección

Al finalizar la Fase 1, los cuatro microservicios obtuvieron Quality Gate Passed en SonarQube, con 0 bugs, 0 vulnerabilidades y 0 Code Smells activos, y sus pipelines de integración finalizaron en estado Passed. Este resultado evidencia que los errores detectados durante el desarrollo fueron gestionados de forma sistemática y resuelta antes de la entrega del sistema.

6 Conclusiones

El presente capítulo recoge las conclusiones derivadas del proceso de reingeniería aplicado al sistema de evaluación docente de la Universidad Industrial de Santander, así como del desarrollo

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

e implementación de la Fase 1 del sistema EVADOC. Cada conclusión sintetiza los hallazgos más relevantes obtenidos a lo largo del proyecto, evidenciando el cumplimiento de los objetivos propuestos y el aporte técnico y metodológico de la solución construida.

1. La aplicación de la ingeniería inversa sobre el sistema de evaluación docente de la UIS permitió reconstruir formalmente su funcionamiento, identificar sus limitaciones estructurales y justificar técnicamente la decisión de reconstrucción arquitectónica. Sin este análisis previo, no hubiera sido posible definir con precisión los requerimientos del nuevo sistema ni garantizar que la solución propuesta respondiera a las necesidades reales del proceso institucional.
2. La combinación del modelo de reingeniería de Roger S. Pressman con un enfoque de arquitectura dirigida demostró ser adecuada para proyectos en los que el sistema legacy presenta obsolescencia tecnológica significativa. Pressman aportó la estructura metodológica para el diagnóstico y la toma de decisiones, mientras que el enfoque arquitectura dirigida garantizó que todas las decisiones de diseño e implementación respondieran a una visión arquitectónica previamente definida.
3. La adopción del prototipado evolutivo como metodología general del proyecto resultó especialmente pertinente dado que el sistema existente presentaba documentación insuficiente. Este enfoque permitió construir el conocimiento del sistema de forma iterativa, refinando progresivamente los requerimientos hasta consolidar una propuesta técnica coherente con las necesidades institucionales.
4. El método ATAM permitió evaluar y justificar formalmente las decisiones arquitectónicas del sistema EVADOC, identificando los compromisos entre atributos de calidad. Este análisis evidenció que la arquitectura de microservicios propuesta satisface los

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- requerimientos no funcionales prioritarios del sistema, en particular rendimiento, disponibilidad, mantenibilidad e interoperabilidad.
5. La arquitectura de microservicios propuesta resuelve las principales limitaciones del sistema legacy: el alto acoplamiento, la baja mantenibilidad y la dificultad de integración con otros sistemas institucionales. Al distribuir la lógica del sistema en servicios independientes (Academic, Assessment, Answer y Form Service), cada módulo puede evolucionar, mantenerse y escalarse de forma autónoma sin comprometer el funcionamiento global del sistema.
 6. Las decisiones tecnológicas adoptadas en el proyecto estuvieron condicionadas por los lineamientos institucionales de la DTIC. La selección de Oracle sobre MongoDB y de RabbitMQ sobre Apache Kafka, aunque implicó sacrificar algunas ventajas técnicas, garantizó el respaldo institucional, la sostenibilidad del sistema a largo plazo y su alineación con la infraestructura tecnológica existente en la UIS.
 7. La implementación de la Fase 1 demostró la viabilidad técnica del sistema propuesto. Los catorce casos de prueba ejecutados sobre las ocho historias de usuario desarrolladas obtuvieron resultado aprobado, validando que los módulos implementados cumplen con los criterios de aceptación definidos y que la arquitectura de microservicios funciona correctamente en condiciones reales de operación.
 8. El presente proyecto sienta las bases técnicas y documentales para el desarrollo de las fases posteriores del sistema EVADOC. La arquitectura definida, los requerimientos levantados y el modelo de datos establecido constituyen un punto de partida sólido para continuar con la implementación de las doce historias de usuario restantes, avanzando hacia un sistema

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

de evaluación docente moderno, escalable e integrado con el ecosistema institucional de la UIS.

7 Trabajo Futuro

El desarrollo de la Fase 1 del sistema EVADOC constituye el punto de partida de un proceso evolutivo de construcción que continuará en fases posteriores. A continuación, se describen las principales líneas de trabajo identificadas para el avance del proyecto.

1. Implementación de las historias de usuario pendientes: De las veinte historias de usuario definidas para el sistema EVADOC, doce quedaron fuera del alcance de la Fase 1 y serán desarrolladas en las siguientes fases del proyecto. Estas historias cubren funcionalidades críticas que complementan lo ya implementado, incluyendo la consulta de información académica cargada, las notificaciones automáticas al estudiante, la autoevaluación docente, la gestión de solicitudes de ajuste y la generación de reportes por unidad organizacional, por docente y a nivel institucional.
2. Ejecución de los casos de prueba pendientes: Los dieciséis casos de prueba que no pudieron ejecutarse en la Fase 1 por corresponder a historias de usuario fuera de su alcance deberán ser ejecutados a medida que se implementen las funcionalidades correspondientes en fases posteriores. Adicionalmente, el CP-28 (acceso público a informes generales sin autenticación) ha sido seleccionado para su ejecución en la siguiente fase del proyecto.
3. Mejoras identificadas en la ingeniería inversa: Durante la fase de ingeniería inversa se identificaron oportunidades de mejora sobre el proceso actual que no forman parte de las historias de usuario definidas pero que representan valor para el sistema: el guardado automático de avances parciales durante el diligenciamiento del formulario, el cierre

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

automático del proceso basado en fechas configuradas y la incorporación de mecanismos de recordatorio progresivo para incentivar la participación temprana de los estudiantes en el proceso evaluativo.

Referencias Bibliográficas

- Universidad Industrial de Santander. (s.f.). *WikiRSI: Documentación institucional de arquitectura, tecnologías y metodologías de desarrollo*. <https://wikirsi.uis.edu.co/en/home>
- Boehm, B. W. (1984). *Software Engineering Economics*. Prentice Hall.
- Bradshaw, C., & Dixon, C. (2017). *Angular 4 for Enterprise-Ready Web Applications*. Packt Publishing.
- Chodorow, K., & Dirolf, M. (2017). *MongoDB: The Definitive Guide*. O'Reilly Media.
- Fischer, H., & Zogaj, S. (2017). "Software reengineering in higher education." *Proceedings of the European Conference on Software Engineering*.
- Gunderloy, M. (2015). *Spring Boot in Action*. Manning Publications. Pohl, K., & Broy, M. (2009). *Software Engineering: Modern Approaches*. Springer.
- Pressman, R. S. (2014). *Ingeniería del software: un enfoque práctico* (8th ed.). McGraw-Hill.
- Pozo, C., Bretones, B., Martos, J., & Alonso, E. (2011). "Evaluación de la actividad docente en el Espacio Europeo de Educación Superior." *Revista Española de Pedagogía*, LXIX (248), 145-163.
- Rivas, M. (2018). *Evaluación y Calidad Educativa*. Editorial Universitaria.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modeling Language Reference Manual*. Addison-Wesley.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. Scrum.org. Recuperado de: <https://www.scrumguides.org>

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley. González, A., Sánchez, L., & Pérez, M. (2020). Modernización del sistema de gestión académica de la UNAM. Universidad Nacional Autónoma de México. Recuperado de: <https://www.unam.mx>
- Martínez, F., & López, J. (2019). Reingeniería del sistema de inscripciones en la UBA: Un enfoque para mejorar la experiencia del usuario. Universidad de Buenos Aires. Recuperado de: <https://www.uba.ar>
- Anaya, Y., (2022). Vicerrectoría de investigación y extensión presupuesto. Universidad Industrial de Santander. Recuperado de: <https://uis.edu.co/wp-content/uploads/2022/08/TARIFAS-INSTITUCIONALES-2022-2.pdf>

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Apéndices

Apéndice A: Historias de usuario

Se presentan las historias de usuario del sistema EVADOC, incluyendo su descripción y criterios de aceptación, con el fin de orientar el desarrollo de las funcionalidades requeridas.

id	HU-01
Épica	EP-01 Gestión de información académica
título	Cargar información académica del periodo
Como	Administrador
Quiero	Cargar la información académica del periodo (docentes, estudiantes, asignaturas) desde los sistemas de la UIS
Para	Habilitar el proceso de evaluación con datos actualizados sin intervención manual
Criterios de aceptación	El sistema carga exitosamente la información de docentes, estudiantes y asignaturas El administrador recibe notificación de carga completada. Los datos quedan disponibles para la configuración del proceso.
Prioridad	Alta

ID	HU-02
Épica	EP-01 Gestión de información académica
título	Consultar información académica cargada
Como	Administrador
Quiero	Consultar la información académica cargada por periodo
Para	Verificar que los datos de docentes, estudiantes y asignaturas sean correctos antes de habilitar la evaluación
Criterios de aceptación	El sistema muestra la información organizada por periodo académico. Se puede filtrar por docente, asignatura o unidad organizacional. Consultar reportes de evaluaciones.
Prioridad	Media

ID	HU-03
Épica	EP-02 Configuración del proceso de evaluación
título	Crear proceso de evaluación docente
Como	Administrador
Quiero	Configurar un proceso de evaluación general y/o por unidad académica
Para	Gestionar calendarios diferenciados según las particularidades de cada unidad

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Criterios de aceptación	El administrador puede configurar el proceso de evaluación de forma general. El administrador define una fecha de inicio y asocia el formulario de evaluación correspondiente. El sistema valida que los datos ingresados sean coherentes con el periodo académico antes de guardar.
Prioridad	Alta
ID	HU-04
Épica	EP-02 Configuración del proceso de evaluación
título	Habilitar proceso de evaluación
Como	Administrador
Quiero	Habilitar el proceso de evaluación para que todos los participantes puedan acceder al mismo tiempo
Para	Iniciar formalmente el periodo de recolección de evaluaciones de manera general
Criterios de aceptación	El proceso cambia a estado "activo" para todos los participantes. Los estudiantes pueden acceder al formulario con su usuario y contraseña institucional.
Prioridad	Alta
ID	HU-05
Épica	EP-02 Configuración del proceso de evaluación
título	Gestionar cierre parcial del proceso por unidad académica
Como	Administrador
Quiero	Cerrar el proceso de forma parcial por unidad académica según se requiera y modificar ese cierre cuando sea necesario
Para	Permitir que cada unidad cierre en el momento que lo necesite, por ejemplo para realizar contrataciones, sin afectar el proceso de las demás unidades
Criterios de aceptación	El administrador puede cerrar el proceso de forma parcial para una unidad académica específica en cualquier momento durante el proceso activo. - El cierre parcial de una unidad no afecta ni interrumpe el proceso de las demás unidades. - El administrador puede modificar un cierre parcial previamente registrado si las condiciones cambian. - El cierre definitivo del proceso ocurre cuando el administrador lo cierra de forma general al finalizar el semestre en la sede principal. - El sistema notifica al administrador cuando se registra o modifica un cierre parcial.
Prioridad	Alta
ID	HU-06
Épica	EP-03 captación de respuestas
título	Recibir notificación de apertura del proceso
Como	Estudiante
Quiero	Recibir una notificación automática cuando el proceso de evaluación sea habilitado

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Para	Conocer oportunamente el inicio del proceso sin esperar al final del semestre
Criterios de aceptación	El sistema envía la notificación automáticamente al habilitar el proceso. El estudiante puede acceder directamente al sistema desde la notificación
Prioridad	Alta

ID	HU-07
Épica	EP-03 captación de respuestas
título	Consultar profesores pendientes por evaluar
Como	Estudiante
Quiero	Ver la lista de profesores que tengo pendientes por evaluar
Para	Saber cuántas evaluaciones me faltan completar en el periodo
Criterios de aceptación	El sistema muestra únicamente los profesores asociados a las asignaturas del estudiante en el periodo activo. La evaluación desaparece de la lista cuando el estudiante la completa. El administrador puede consultar el estado de evaluaciones de todos los estudiantes.
Prioridad	Alta

ID	HU-08
Épica	EP-03 captación de respuestas
título	Diligenciar formulario de evaluación docente
Como	Estudiante
Quiero	Diligenciar el formulario de evaluación para cada profesor asignado
Para	Registrar mi valoración sobre el desempeño docente del periodo
Criterios de aceptación	El formulario carga desde el Form Service. Las respuestas quedan registradas de forma anónima; el profesor no puede identificar al estudiante. Al enviar, desaparece de pendientes. El sistema muestra mensaje de confirmación de completitud.
Prioridad	Alta

ID	HU-09
Épica	EP-03 captación de respuestas
título	Omitir evaluación con justificación
Como	Estudiante
Quiero	Omitir la evaluación de un profesor registrando una justificación
Para	Indicar formalmente que no puedo evaluar a ese docente en el periodo
Criterios de aceptación	El sistema permite seleccionar la opción NS/NR u omitir evaluación. Es obligatorio registrar una justificación antes de confirmar la omisión. La omisión queda registrada en el sistema con fecha y justificación.
Prioridad	media

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

ID	HU-10
Épica	EP-04 Autoevaluación docente
título	Acceder al formulario de autoevaluación
Como	Docente
Quiero	Acceder al formulario de autoevaluación desde el sistema EVADOC
Para	Completar mi autoevaluación dentro del periodo académico activo
Criterios de aceptación	El formulario solo está disponible cuando el proceso de evaluación está activo. El docente solo puede acceder a su propio formulario de autoevaluación. El sistema confirma el registro exitoso al finalizar.
Prioridad	Alta

ID	HU-11
Épica	EP-04 Autoevaluación docente
título	Consultar estado de autoevaluación
Como	Docente
Quiero	Consultar si ya completé mi autoevaluación en el periodo activo
Para	Verificar que mi participación en el proceso evaluativo quedó registrada
Criterios de aceptación	El sistema muestra el estado de la autoevaluación (pendiente o completada). Si está completada, muestra la fecha de registro. El docente no puede modificar una autoevaluación ya enviada.
Prioridad	Media

ID	HU-12
Épica	EP-05 Gestión de solicitudes de ajuste
título	Recibir solicitud de ajuste
Como	Administrador
Quiero	Recibir en el sistema las solicitudes de corrección o ajuste relacionadas con el proceso evaluativo
Para	Gestionar de forma centralizada y trazable las solicitudes sin intervención manual externa
Criterios de aceptación	El sistema registra la solicitud con fecha, hora y datos del solicitante. El administrador recibe notificación automática de nueva solicitud. La solicitud queda en estado "pendiente" hasta ser revisada.
Prioridad	Alta

ID	HU-13
Épica	EP-05 Gestión de solicitudes de ajuste
título	Validar y resolver solicitud de ajuste
Como	Administrador
Quiero	Revisar, aprobar o rechazar una solicitud de ajuste dentro del sistema

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Para	Aplicar correcciones al proceso evaluativo de forma controlada y con trazabilidad
Criterios de aceptación	El administrador puede aprobar o rechazar la solicitud. - El sistema notifica al solicitante el resultado de su solicitud. - La solicitud queda registrada con su resolución y fecha.
Prioridad	Alta

ID	HU-14
Épica	EP-05 Gestión de solicitudes de ajuste
título	Consultar historial de solicitudes
Como	Administrador
Quiero	Consultar el historial completo de solicitudes de ajuste por periodo
Para	Auditar las correcciones realizadas y garantizar trazabilidad del proceso
Criterios de aceptación	El sistema muestra todas las solicitudes con su estado, fecha y resolución. Se puede filtrar por periodo, estado y solicitante. El historial no puede ser modificado una vez registrado.
Prioridad	Media

ID	HU-15
Épica	EP-06 Generación de reportes
título	Generar reporte por unidad organizacional
Como	Administrador
Quiero	Generar reportes de resultados de evaluación agrupados por unidad organizacional.
Para	Analizar el desempeño docente por facultad o departamento
Criterios de aceptación	Muestra estadísticas consolidadas por unidad organizacional. El reporte puede ser consultado desde el Dashboard.
Prioridad	Alta

ID	HU-16
Épica	EP-06 Generación de reportes
título	Generar reporte individual por docente
Como	Administrador
Quiero	Generar reportes individuales de evaluación por docente
Para	Que cada docente pueda consultar sus resultados del periodo evaluativo.
Criterios de aceptación	El reporte incluye los resultados de evaluación estudiantil y autoevaluación. Solo el docente y roles autorizados pueden acceder al reporte individual. El reporte está disponible en el Dashboard.
Prioridad	Alta

ID	HU-17
-----------	--------------

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Épica	EP-06 Generación de reportes
título	Consultar resultados según rol
Como	Docente / Directivo / Unidad académica
Quiero	Consultar los resultados de evaluación según mi perfil de acceso
Para	Acceder únicamente a la información que me corresponde según mi rol institucional
Criterios de aceptación	El sistema filtra la información visible según el rol del usuario. El docente solo ve sus propios resultados. El directivo ve los resultados de su unidad organizacional. Los reportes son de solo lectura.
Prioridad	Alta

ID	HU-18
Épica	EP-06 Generación de reportes
título	Consultar historial de reportes de periodos anteriores
Como	Docente / Administrador / Directivo
Quiero	Consultar reportes de evaluación de semestres anteriores
Para	Hacer seguimiento histórico sin depender de archivos físicos o externos
Criterios de aceptación	El sistema almacena automáticamente los reportes al cerrar cada proceso. Cada actor consulta según su rol. Exportables en PDF. Disponible desde el primer periodo en el nuevo sistema.
Prioridad	Alta

ID	HU-19
Épica	EP-06 Generación de reportes
título	Acceso público a informes generales de la universidad
Como	Usuario externo / público general
Quiero	Consultar los informes generales de evaluación docente de la UIS
Para	Garantizar transparencia institucional
Criterios de aceptación	Publica resultados generales a nivel institucional al cerrar el proceso. Solo datos agregados no identifica docentes individuales. Acceso sin autenticación desde el sitio web o intranet institucional.
Prioridad	media

ID	HU-20
Épica	EP-07 Gestión de usuarios y roles
título	Gestionar formularios dinámicos
Como	Administrador
Quiero	Crear, editar y eliminar formularios de evaluación desde el Manager UI
Para	Adaptar los instrumentos de evaluación a las necesidades de cada periodo académico

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Criterios de aceptación	El administrador puede crear formularios con secciones, preguntas y opciones de respuesta. Los formularios pueden ser editados antes de habilitar el proceso. Un formulario en uso no puede ser eliminado.
Prioridad	Alta

Apéndice B: Escenarios de atributos de calidad – ATAM

Los siguientes escenarios describen situaciones concretas que el sistema EVADOC debe manejar para cumplir cada atributo de calidad definido en el árbol de utilidad.

Rendimiento	
Campo	Descripción
Estímulo	500 estudiantes acceden simultáneamente al sistema para diligenciar evaluaciones
Contexto	Cierre del periodo académico, evaluación habilitada
Respuesta esperada	El sistema procesa las solicitudes sin degradación del servicio
Medida	Tiempo de respuesta menor a 3 segundos por transacción

Disponibilidad	
Campo	Descripción
Estímulo	Falla en uno de los microservicios durante el periodo de evaluación
Contexto	Proceso de evaluación activo con usuarios conectados
Respuesta esperada	Los demás microservicios continúan operando sin interrupción general
Medida	Disponibilidad del 99% durante periodos académicos activos

Seguridad	
Campo	Descripción
Estímulo	Un usuario intenta acceder a información de otro rol sin autorización
Contexto	Sistema en operación normal
Respuesta esperada	El sistema rechaza el acceso y registra el intento
Medida	0% de accesos no autorizados exitosos

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Mantenibilidad

Campo	Descripción
Estímulo	La DTIC requiere modificar el módulo de formularios por cambio normativo
Contexto	Sistema en producción
Respuesta esperada	El cambio se realiza en el microservicio correspondiente sin afectar los demás
Medida	Modificación implementada sin tiempo de inactividad en otros servicios

Interoperabilidad

Campo	Descripción
Estímulo	Inicio de un nuevo periodo académico con nueva información de docentes y estudiantes
Contexto	Carga de datos desde los sistemas académicos de la UIS
Respuesta esperada	El academic service recibe e integra la información sin intervención manual
Medida	Sincronización completada en menos de 5 minutos sin errores de integridad

Confiabilidad

Campo	Descripción
Estímulo	El estudiante completa y envía su evaluación
Contexto	Sistema bajo carga normal o alta
Respuesta esperada	La respuesta queda registrada de forma íntegra y sin duplicados
Medida	0% de pérdida o corrupción de datos registrados

Escalabilidad

Campo	Descripción
Estímulo	Se incorpora una nueva facultad con 1.000 estudiantes adicionales al proceso evaluativo
Contexto	Sistema en producción durante un periodo académico activo
Respuesta esperada	El sistema absorbe la carga adicional sin modificar su arquitectura
Medida	Sin degradación del rendimiento con hasta el doble de usuarios actuales

Trazabilidad

Campo	Descripción
--------------	--------------------

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Estímulo	El coordinador requiere auditar quién realizó una corrección en el proceso evaluativo
Contexto	cierre del periodo de evaluación
Respuesta esperada	El sistema presenta el registro completo de acciones con usuario, fecha y hora
Medida	100% de acciones administrativas registradas y consultables

Usabilidad

Campo	Descripción
Estímulo	Un estudiante accede por primera vez al sistema para diligenciar una evaluación
Contexto	Sin capacitación previa
Respuesta esperada	El estudiante completa el proceso sin asistencia externa
Medida	Tasa de completitud del formulario mayor al 90% sin errores de navegación

Portabilidad

Campo	Descripción
Estímulo	La DTIC requiere desplegar el sistema en un nuevo servidor institucional
Contexto	Migración de entorno
Respuesta esperada	El sistema se despliega correctamente mediante los contenedores Docker existentes
Medida	Despliegue completado en menos de 2 horas sin modificaciones al código

Apéndice C: Casos de prueba

se incluyen los casos de prueba del sistema EVADOC utilizados para validar las funcionalidades implementadas.

campo	Detalle
ID	CP-01
HU relacionada	HU-01
Descripción	Carga exitosa de información académica del periodo
Precondiciones	El administrador ha iniciado sesión. Existe información académica disponible en el Academic Service para el periodo activo.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Pasos	1. El administrador accede al módulo de carga de información académica. 2. Selecciona el periodo académico. 3. Inicia la carga desde el Academic Service. 4. Espera la notificación del sistema.
Resultado esperado	El sistema carga exitosamente la información de docentes, estudiantes y asignaturas. El administrador recibe notificación de carga completada. Los datos quedan disponibles para la configuración del proceso.
Resultado obtenido	El sistema cargó exitosamente la información de docentes, estudiantes y asignaturas del periodo seleccionado. Se generó la notificación de carga completada para el administrador y los datos quedaron disponibles en el módulo de configuración del proceso.
Estado	aprobado

Campo	Detalle
ID	CP-02
HU relacionada	HU-01
Descripción	Notificación de fallo al ocurrir un error en la carga
Precondiciones	El administrador ha iniciado sesión. El Academic Service no está disponible o presenta un error de conexión.
Pasos	1. El administrador accede al módulo de carga. 2. Inicia manualmente la carga. 3. El Academic Service falla durante el proceso.
Resultado esperado	El sistema detiene la carga y notifica al administrador con el detalle del error. Los datos previos no se ven alterados.
Resultado obtenido	El sistema detuvo el proceso de carga al detectar el fallo en el Academic Service y notificó al administrador con el detalle del error. Los datos previamente cargados no fueron alterados.
Estado	aprobado

Campo	Detalle
ID	CP-03
HU relacionada	HU-02
Descripción	Consulta exitosa de información académica filtrada por unidad organizacional
Precondiciones	El administrador ha iniciado sesión. Se ha cargado información académica para el periodo activo.
Pasos	1. El administrador accede al módulo de consulta de información académica. 2. Selecciona el periodo académico. 3. Aplica filtro por unidad organizacional. 4. Consulta los resultados.
Resultado esperado	El sistema muestra la información filtrada por la unidad seleccionada, organizada por periodo. Los datos reflejan lo cargado desde la UIS.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-04
HU relacionada	HU-03
Descripción	Creación exitosa de proceso de evaluación con datos válidos
Precondiciones	El administrador ha iniciado sesión. La información académica del periodo ha sido cargada.
Pasos	1. El administrador accede al módulo de configuración del proceso. 2. Selecciona el tipo de evaluación (heteroevaluación o coevaluación). 3. Selecciona el periodo académico. 4. Define la fecha de evaluación. 5. Indica si la evaluación estará habilitada por unidades organizacionales del evaluador y del evaluado. 6. Selecciona el tipo de evaluador y el tipo de evaluado. 7. Asocia el formulario de evaluación correspondiente. 8. Guarda el proceso.
Resultado esperado	El sistema guarda el proceso correctamente. El sistema valida que los datos sean coherentes con el periodo académico. El proceso queda disponible para ser habilitado.
Resultado obtenido	El sistema guardó el proceso de evaluación correctamente con todos los parámetros configurados. Los datos fueron validados con coherencia respecto al periodo académico y el proceso quedó disponible en estado "configurado" para ser habilitado.
Estado	aprobado
Campo	Detalle
ID	CP-05
HU relacionada	HU-03
Descripción	Rechazo de proceso con fecha de cierre anterior a la de inicio
Precondiciones	El administrador ha iniciado sesión y accede al módulo de configuración.
Pasos	1. El administrador accede al módulo de configuración del proceso. 2. Ingresa una fecha de inicio y una fecha de cierre anterior a la de inicio.
Resultado esperado	El sistema rechaza el proceso y muestra un mensaje de validación indicando que las fechas no son coherentes con el periodo académico.
Resultado obtenido	El sistema rechazó la creación del proceso y mostró un mensaje de validación indicando que la fecha de cierre no puede ser anterior a la fecha de inicio.
Estado	aprobado
Campo	Detalle

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

ID	CP-06
HU relacionada	HU-04
Descripción	Habilitación exitosa del proceso para todos los participantes al mismo tiempo
Precondiciones	El administrador ha iniciado sesión. Existe un proceso en estado "configurado".
Pasos	1. El administrador accede al módulo de habilitación. 2. Habilita el proceso. 3. Confirma la habilitación.
Resultado esperado	El proceso cambia a estado "activo" para todos los participantes al mismo tiempo. Los estudiantes pueden acceder al formulario con su usuario y contraseña institucional.
Resultado obtenido	El proceso cambió a estado "activo" simultáneamente para todos los participantes. Los estudiantes pudieron acceder al formulario de evaluación mediante sus credenciales institucionales.
Estado	aprobado

Campo	Detalle
ID	CP-07
HU relacionada	HU-05
Descripción	Cierre parcial exitoso del proceso para una unidad académica específica
Precondiciones	El proceso está activo. El administrador ha iniciado sesión.
Pasos	1. El administrador accede al módulo de gestión de cierres. 2. Selecciona la unidad académica a cerrar. 3. Registra el cierre parcial. 4. Confirma la acción.
Resultado esperado	El proceso cierra para la unidad académica seleccionada. Las demás unidades continúan con el proceso activo sin verse afectadas. El sistema muestra una confirmación al administrador que confirma el cierre parcial registrado.
Resultado obtenido	El proceso cerró correctamente para la unidad académica seleccionada. Las demás unidades continuaron con el proceso activo sin interrupciones. El sistema mostró un mensaje de confirmación del cierre parcial registrado al administrador.
Estado	aprobado

Campo	Detalle
ID	CP-08
HU relacionada	HU-05
Descripción	Los resultados de una unidad quedan disponibles al registrar su cierre parcial.
Precondiciones	El proceso está activo. La unidad académica tiene el 100% de sus evaluaciones respondidas.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Pasos	1. El administrador accede al módulo de gestión de cierres. 2. Selecciona la unidad académica que completó todas sus evaluaciones. 3. Registra el cierre parcial para esa unidad.
Resultado esperado	El sistema actualiza el cierre parcial con los nuevos datos. El proceso de las demás unidades no se ve afectado. El sistema notifica la modificación al administrador.
Resultado obtenido	El sistema registra el cierre parcial de la unidad. Los resultados de esa unidad quedan disponibles para su visualización. Las demás unidades continúan con el proceso activo sin verse afectadas.
Estado	aprobado

Campo	Detalle
ID	CP-09
HU relacionada	HU-06
Descripción	Estudiante recibe notificación automática al habilitarse el proceso de evaluación
Precondiciones	El proceso de evaluación ha sido habilitado por el administrador. El estudiante tiene correo institucional registrado.
Pasos	1. El administrador habilita el proceso de evaluación. 2. El sistema envía la notificación automáticamente. 3. El estudiante accede al sistema desde la notificación.
Resultado esperado	El estudiante recibe la notificación automáticamente al habilitarse el proceso. Puede acceder directamente al sistema desde la notificación.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

Campo	Detalle
ID	CP-10
HU relacionada	HU-06
Descripción	Estudiante accede al sistema directamente desde el enlace de la notificación
Precondiciones	El proceso ha sido habilitado. El estudiante ha recibido la notificación.
Pasos	1. El estudiante abre la notificación recibida. 2. Hace clic en el enlace de acceso al sistema incluido en la notificación.
Resultado esperado	El sistema carga correctamente y el estudiante puede iniciar el proceso de evaluación sin necesidad de buscar el acceso manualmente.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Campo	Detalle
ID	CP-11
HU relacionada	HU-07
Descripción	Lista correcta de profesores pendientes para el estudiante
Precondiciones	El proceso del grupo del estudiante está activo. El estudiante tiene asignaturas registradas con profesores en el periodo.
Pasos	1. El estudiante inicia sesión en el sistema. 2. Accede al módulo de evaluación. 3. Consulta la lista de profesores pendientes.
Resultado esperado	El sistema muestra únicamente los profesores asociados a las asignaturas del estudiante en el periodo activo, todos en estado "pendiente".
Resultado obtenido	El sistema mostró únicamente los profesores asociados a las asignaturas matriculadas por el estudiante en el periodo activo, todos en estado "pendiente" de evaluación.
Estado	aprobado

Campo	Detalle
ID	CP-12
HU relacionada	HU-07
Descripción	La evaluación desaparece de la lista al ser completada
Precondiciones	El estudiante tiene al menos un profesor pendiente por evaluar.
Pasos	1. El estudiante completa y envía la evaluación de un profesor. 2. El sistema muestra el mensaje de confirmación. 3. El estudiante regresa a la lista de pendientes. 4. El estudiante completa y envía las evaluaciones de todos los profesores restantes.
Resultado esperado	El profesor evaluado ya no aparece en la lista de pendientes del estudiante. Cuando no quedan evaluaciones pendientes, el sistema muestra un mensaje indicando que todas las evaluaciones han sido contestadas satisfactoriamente.
Resultado obtenido	El profesor evaluado desapareció de la lista de pendientes tras completar y enviar el formulario. Al finalizar todas las evaluaciones, el sistema mostró un mensaje indicando que todas habían sido contestadas satisfactoriamente.
Estado	aprobado

Campo	Detalle
ID	CP-13
HU relacionada	HU-08
Descripción	Estudiante completa formulario y la respuesta queda registrada
Precondiciones	El proceso está activo. El estudiante tiene profesores pendientes por evaluar.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Pasos	1. El estudiante selecciona un profesor pendiente. 2. Carga el formulario desde el Form Service. 3. Responde todas las preguntas. 4. Envía el formulario.
Resultado esperado	El sistema registra la evaluación exitosamente. El sistema muestra el mensaje de confirmación de completitud. La evaluación desaparece de la lista de pendientes.
Resultado obtenido	El sistema registró la evaluación de forma anónima y exitosa. Se mostró el mensaje de confirmación de completitud al estudiante y el profesor evaluado desapareció de la lista de pendientes.
Estado	aprobado

Campo	Detalle
ID	CP-14
HU relacionada	HU-09
Descripción	Estudiante omite evaluación registrando justificación
Precondiciones	El proceso está activo. El estudiante tiene al menos un profesor pendiente.
Pasos	1. El estudiante selecciona la opción de omitir evaluación para un profesor. 2. El sistema solicita justificación. 3. El estudiante selecciona una opción del listado desplegable. 4. Confirma la omisión.
Resultado esperado	La omisión queda registrada con fecha y justificación. El profesor desaparece de la lista de pendientes.
Resultado obtenido	La omisión quedó registrada en el sistema con la fecha y la justificación seleccionada por el estudiante. El profesor omitido desapareció de la lista de evaluaciones pendientes.
Estado	aprobado

Campo	Detalle
ID	CP-15
HU relacionada	HU-09
Descripción	El sistema impide omitir una evaluación sin registrar justificación
Precondiciones	El proceso está activo. El estudiante intenta omitir sin ingresar justificación.
Pasos	1. El estudiante selecciona omitir evaluación para un profesor. 2. El sistema solicita justificación. 3. El estudiante deja el campo vacío. 4. Intenta confirmar la omisión.
Resultado esperado	El sistema no permite confirmar la omisión y muestra un mensaje indicando que la justificación es obligatoria.
Resultado obtenido	El sistema bloqueó la confirmación de la omisión y mostró un mensaje indicando que el campo de justificación es obligatorio para continuar.
Estado	aprobado

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Campo	Detalle
ID	CP-16
HU relacionada	HU-10
Descripción	Docente accede y completa su formulario de autoevaluación
Precondiciones	El proceso de evaluación está activo. El docente tiene autoevaluación pendiente.
Pasos	1. El docente inicia sesión en el sistema. 2. Accede al módulo de autoevaluación. 3. Completa el formulario. 4. Envía la autoevaluación.
Resultado esperado	El sistema registra exitosamente la autoevaluación del docente y confirma el registro con un mensaje.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

Campo	Detalle
ID	CP-17
HU relacionada	HU-10
Descripción	Docente no puede acceder al formulario cuando el proceso no está activo
Precondiciones	El proceso de evaluación está en estado "configurado" o "cerrado"
Pasos	1. El docente inicia sesión. 2. Intenta acceder al módulo de autoevaluación.
Resultado esperado	El sistema no muestra el formulario de autoevaluación e indica que el proceso no está activo.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

Campo	Detalle
ID	CP-18
HU relacionada	HU-11
Descripción	Docente consulta estado de autoevaluación ya completada
Precondiciones	El docente ha completado y enviado su autoevaluación en el periodo activo.
Pasos	1. El docente inicia sesión. 2. Accede al módulo de consulta de estado de autoevaluación.
Resultado esperado	El sistema muestra el estado "completada" con la fecha de registro. El docente no tiene opción de modificar la autoevaluación.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Campo	Detalle
ID	CP-19
HU relacionada	HU-12
Descripción	Solicitud de ajuste registrada y notificada al administrador
Precondiciones	El proceso está activo o cerrado. El usuario tiene acceso al módulo de solicitudes.
Pasos	1. El usuario accede al módulo de solicitudes de ajuste. 2. Ingresa los datos de la solicitud. 3. Envía la solicitud.
Resultado esperado	El sistema registra la solicitud con fecha, hora y datos del solicitante en estado "pendiente". El administrador recibe notificación automática.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-20
HU relacionada	HU-13
Descripción	Administrador aprueba una solicitud de ajuste
Precondiciones	Existe al menos una solicitud en estado "pendiente".
Pasos	1. El administrador accede al módulo de gestión de solicitudes. 2. Selecciona una solicitud pendiente. 3. Revisa el detalle. 4. Aprueba la solicitud.
Resultado esperado	La solicitud cambia a estado "aprobada" con fecha de resolución. El sistema notifica al solicitante el resultado.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-21
HU relacionada	HU-13
Descripción	Administrador rechaza una solicitud de ajuste
Precondiciones	Existe al menos una solicitud en estado "pendiente".
Pasos	1. El administrador accede al módulo de gestión de solicitudes. 2. Selecciona una solicitud pendiente. 3. Rechaza la solicitud.
Resultado esperado	La solicitud cambia a estado "rechazada" con fecha de resolución. El sistema notifica al solicitante el rechazo.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-22
HU relacionada	HU-14
Descripción	Administrador consulta historial de solicitudes filtrado por periodo
Precondiciones	Existen solicitudes registradas en periodos anteriores.
Pasos	1. El administrador accede al módulo de historial de solicitudes. 2. Aplica filtro por periodo académico. 3. Consulta los resultados.
Resultado esperado	El sistema muestra todas las solicitudes del periodo seleccionado con su estado, fecha y resolución. El historial no puede ser modificado.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-23
HU relacionada	HU-15
Descripción	Administrador consulta reporte por unidad organizacional desde el Dashboard
Precondiciones	El proceso de evaluación ha sido cerrado. Existen respuestas registradas.
Pasos	1. El administrador accede al Dashboard. 2. Selecciona la unidad organizacional. 3. Consulta el reporte de resultados.
Resultado esperado	El sistema muestra estadísticas consolidadas por unidad organizacional desde el Dashboard.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-24
HU relacionada	HU-16
Descripción	Reporte individual por docente disponible en el Dashboard para roles autorizados
Precondiciones	El proceso ha sido cerrado. Existen evaluaciones registradas para el docente.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Pasos	1. El administrador accede al módulo de reportes individuales en el Dashboard. 2. Selecciona el docente. 3. Consulta el reporte.
Resultado esperado	El reporte incluye los resultados de evaluación estudiantil y autoevaluación. Solo el docente y roles autorizados pueden acceder al reporte individual. El reporte está disponible en el Dashboard.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

Campo	Detalle
ID	CP-25
HU relacionada	HU-17
Descripción	Docente accede únicamente a sus propios resultados según su rol
Precondiciones	El proceso ha sido cerrado. El docente tiene resultados disponibles.
Pasos	1. El docente inicia sesión. 2. Accede al módulo de resultados.
Resultado esperado	El sistema muestra únicamente los resultados del docente. No puede visualizar resultados de otros docentes. Los reportes son de solo lectura.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

Campo	Detalle
ID	CP-26
HU relacionada	HU-17
Descripción	Directivo accede únicamente a los resultados de su unidad organizacional
Precondiciones	El proceso ha sido cerrado. El directivo tiene sesión activa.
Pasos	1. El directivo inicia sesión. 2. Accede al módulo de resultados.
Resultado esperado	El sistema muestra únicamente los resultados de los docentes pertenecientes a la unidad organizacional del directivo. No puede acceder a resultados de otras unidades.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto

Campo	Detalle
ID	CP-27
HU relacionada	HU-18

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Descripción	Administrador consulta reporte histórico de un semestre anterior
Precondiciones	Existen reportes almacenados de al menos un periodo anterior.
Pasos	1. El administrador accede al módulo de historial de reportes. 2. Selecciona el periodo anterior de interés. 3. Consulta el reporte. 4. Exporta en PDF.
Resultado esperado	El sistema muestra el reporte del periodo seleccionado. El archivo puede exportarse en PDF correctamente.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-28
HU relacionada	HU-19
Descripción	Usuario externo accede a informes generales sin iniciar sesión
Precondiciones	El proceso ha sido cerrado y los informes generales han sido publicados.
Pasos	1. Un usuario externo accede al sitio web o intranet institucional. 2. Navega a la sección de informes de evaluación docente. 3. Consulta los informes generales sin autenticarse.
Resultado esperado	El sistema permite el acceso y muestra únicamente datos agregados a nivel universitario sin información que identifique docentes individuales. No se requiere inicio de sesión.
Resultado obtenido	
Estado	pendiente para próxima fase del proyecto
Campo	Detalle
ID	CP-29
HU relacionada	HU-20
Descripción	Administrador crea un formulario con secciones y preguntas
Precondiciones	El administrador ha iniciado sesión. No existe un proceso activo que use este formulario.
Pasos	1. El administrador accede al módulo de gestión de formularios en el Manager UI. 2. Crea un nuevo formulario. 3. Agrega secciones, preguntas y opciones de respuesta. 4. Guarda el formulario.
Resultado esperado	El sistema almacena el formulario correctamente. Queda disponible para ser asociado a un proceso de evaluación.

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Resultado obtenido	El sistema almacenó el formulario con sus secciones, preguntas y opciones de respuesta. El formulario quedó disponible en el listado para ser asociado a un proceso de evaluación.
Estado	aprobado
Campo	Detalle
ID	CP-30
HU relacionada	HU-20
Descripción	El sistema impide eliminar un formulario que está siendo usado en un proceso activo
Precondiciones	Existe un formulario asociado a un proceso de evaluación activo o configurado.
Pasos	1. El administrador accede al módulo de gestión de formularios. 2. Selecciona el formulario en uso. 3. Intenta eliminarlo.
Resultado esperado	El sistema rechaza la eliminación y muestra un mensaje indicando que el formulario está en uso y no puede ser eliminado.
Resultado obtenido	El sistema rechazó la eliminación y mostró un mensaje indicando que el formulario se encuentra asociado a un proceso de evaluación activo y no puede ser eliminado.
Estado	aprobado

Apéndice D: Evidencia de Calidad y Corrección de Errores

El presente anexo consolida la evidencia del proceso de detección y corrección de errores registrado durante el desarrollo de la Fase 1 del sistema EVADOC, obtenida mediante SonarQube y los pipelines CI/CD de GitLab.

Análisis de Calidad, SonarQube

- Academic Service

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 63

Gráfica histórica de code smells resueltos-academic

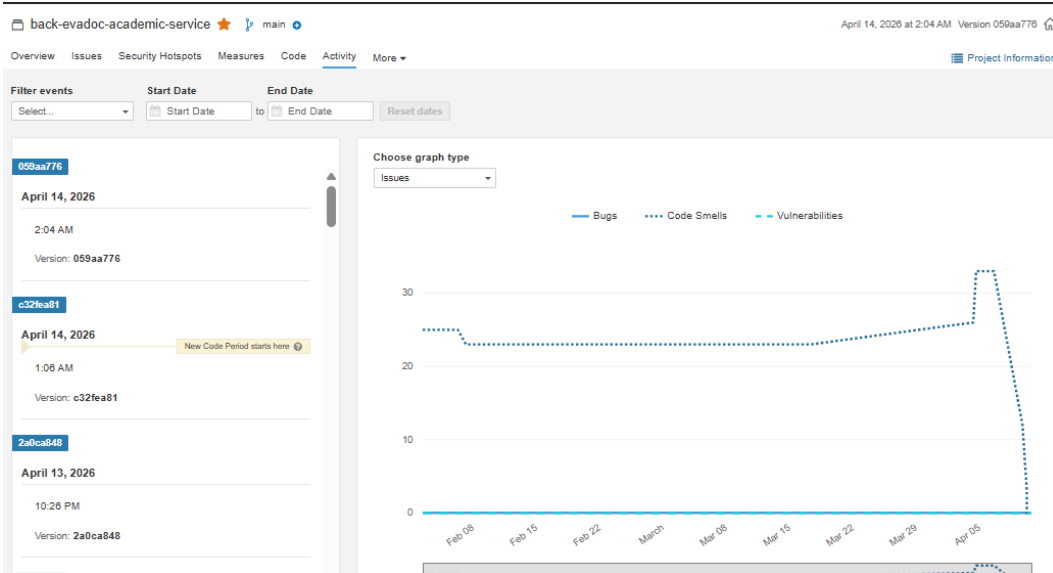
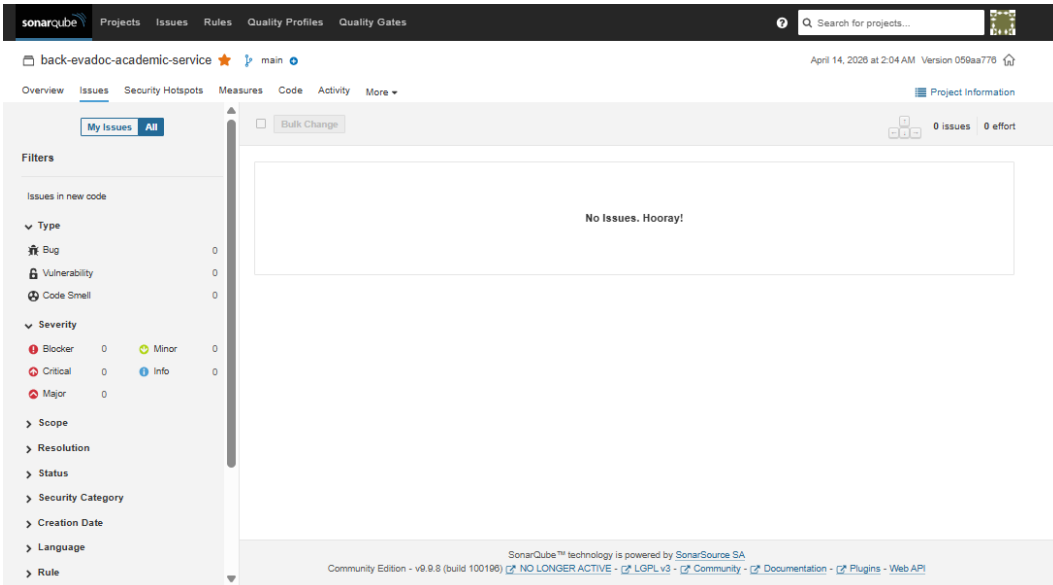


Figura 64

Detalle de issues-academic



REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- Answer Service

Figura 65

Gráfica histórica de code smells resueltos-answer

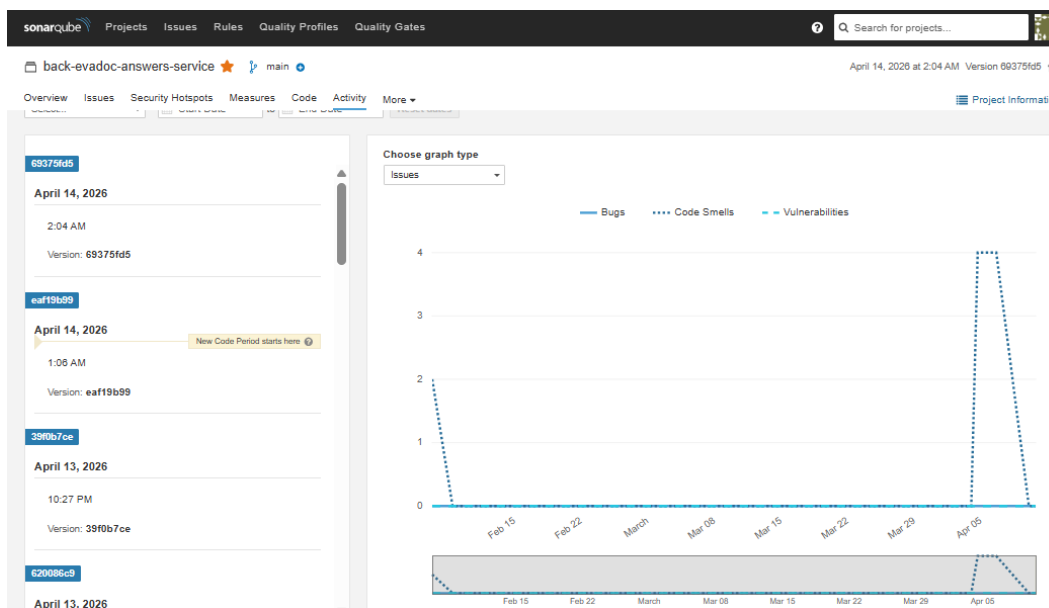
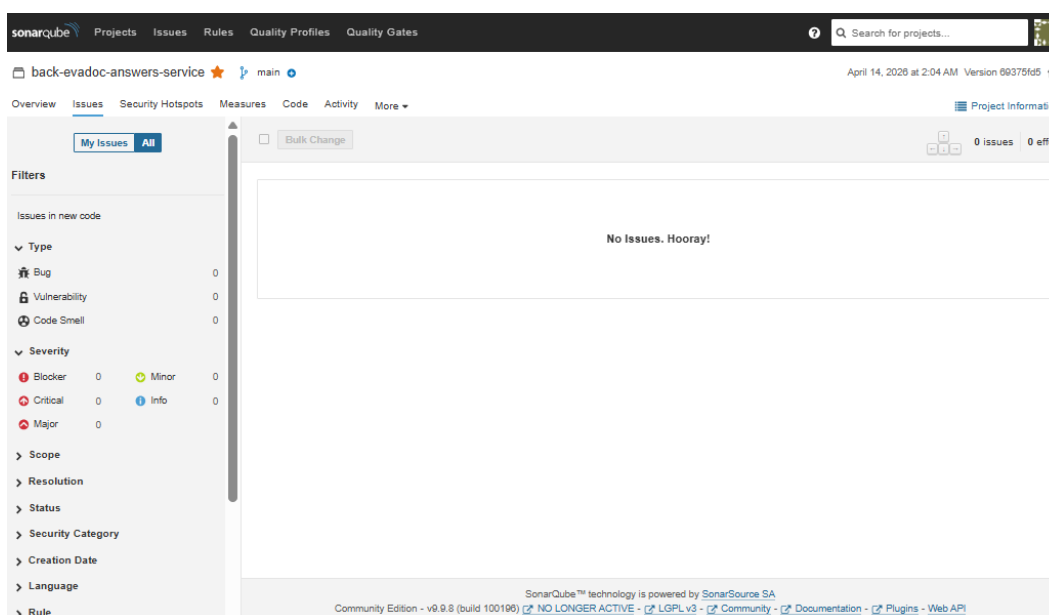


Figura 66

Detalle de issues- answer



REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

- Assessment Service

Figura 67

Gráfica histórica de code smells resueltos-assessment

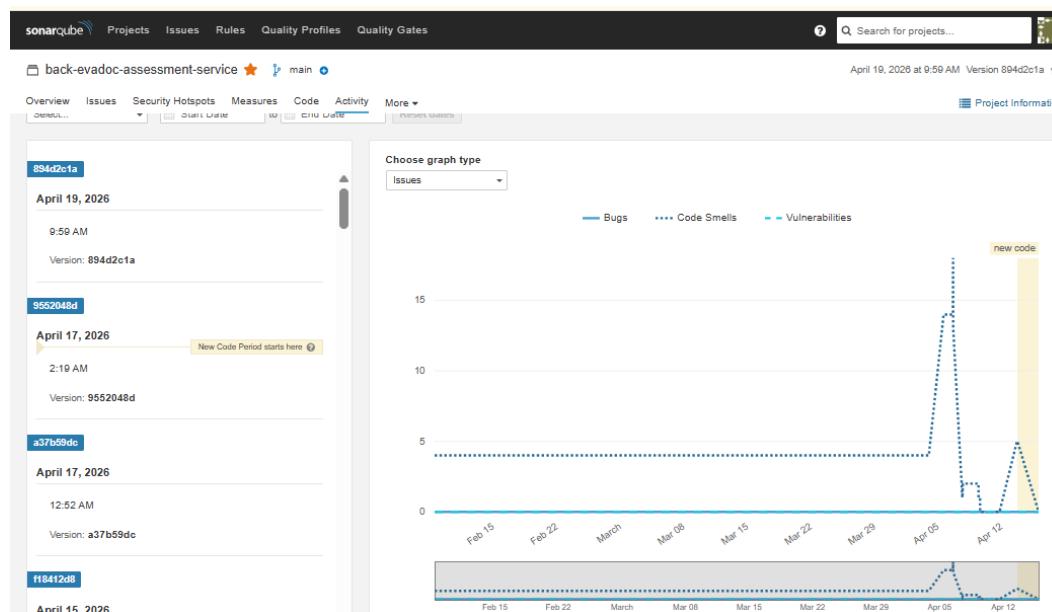
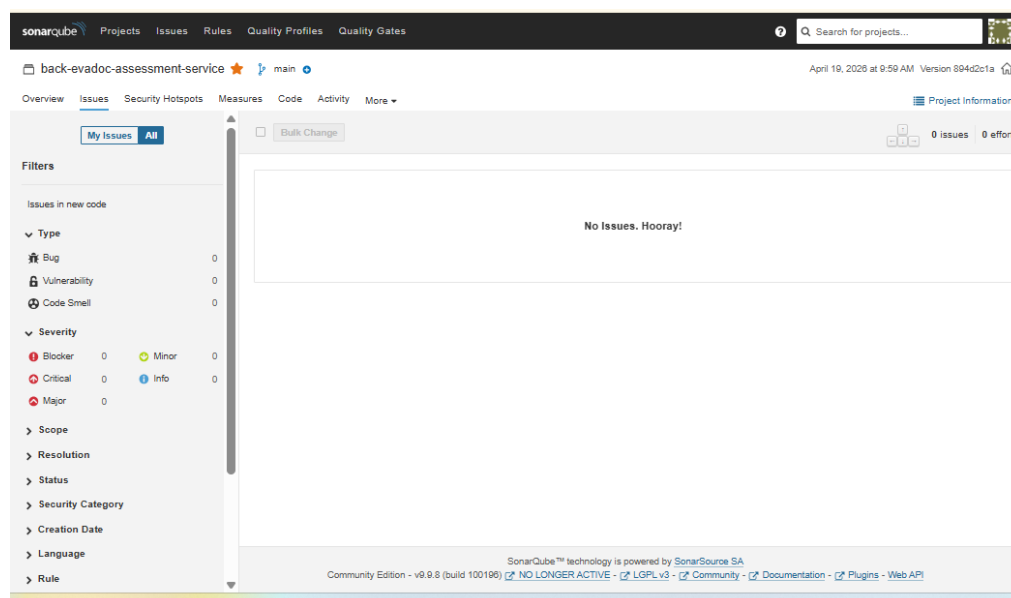


Figura 68

Detalle de issues-assessment

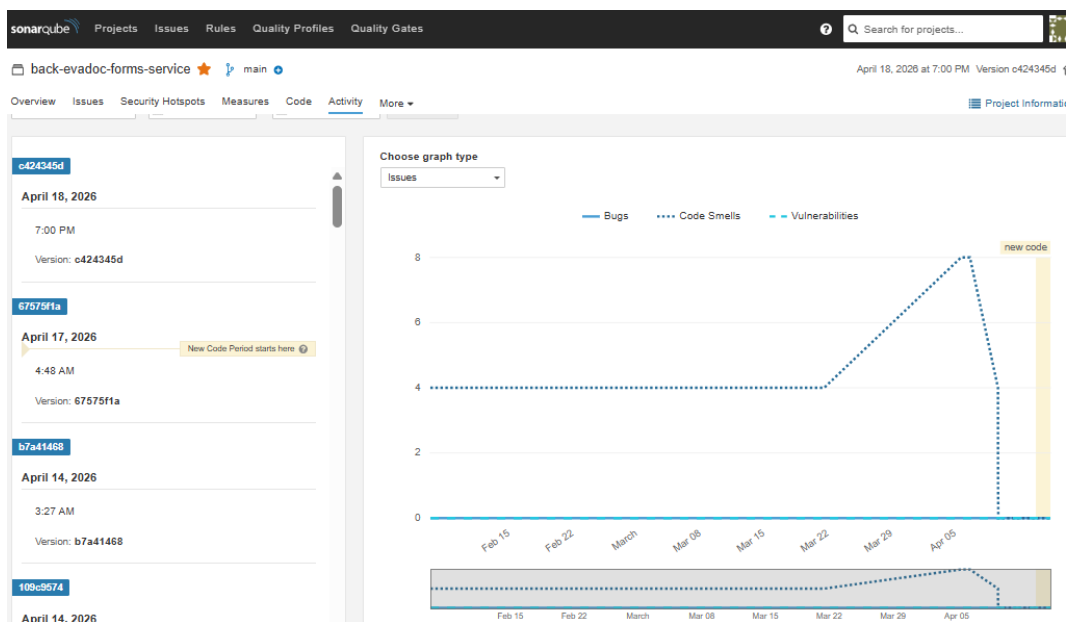


REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

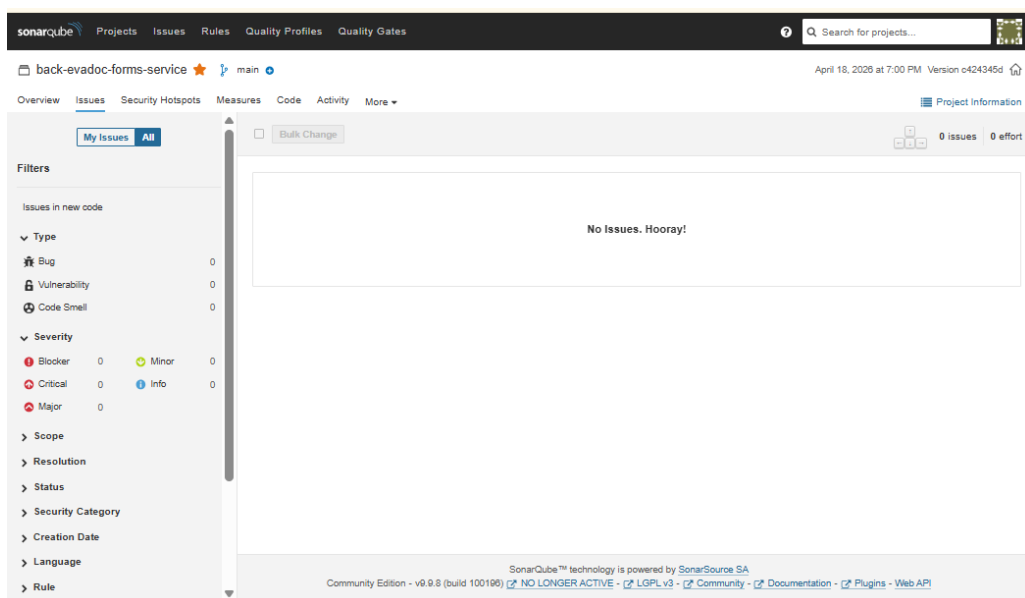
- Form Service

Figura 69

Gráfica histórica de code smells resueltos-form

**Figura 70**

Detalle de issues-form



REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura sobre Pruebas Unitarias e Integración del Backend

Figura 71

Métricas de pruebas unitarias

Metric	back-evadoc-academic-service	back-evadoc-answers-service	back-evadoc-assessment-service	back-evadoc-forms-service
Uncovered Conditions	0	0	0	0
Overall Coverage	90.1%	92.9%	91.5%	97.0%
Lines to Cover	369	50	269	131
Uncovered Lines	28	5	13	1
Line Coverage	92.4%	90.0%	95.2%	99.2%
Conditions to Cover	86	20	72	34
Uncovered Conditions	17	0	16	4
Condition Coverage	80.2%	100%	77.8%	88.2%
Tests				
Unit Tests	94	17	87	26
Errors	0	0	0	0
Failures	0	0	0	0
Skipped	0	0	0	0
Success	100%	100%	100%	100%
Duration	2s	2s	2s	2s

Figura 72

Quality gate academic service

Passed
All conditions passed.

The quality gate used by this project does not comply with Clean as You Code. Fixing this quality gate will help you achieve a Clean Code state. [Learn why](#)

Metric	Value	Status
New Bugs	0	Reliability (A)
New Vulnerabilities	0	Security (A)
New Security Hotspots	0	Security Review (A)
Added Debt	0	Maintainability (A)
New Code Smells	0	Maintainability (A)
Coverage on New Lines to cover	0	
Duplications on New Lines	10	0.0%

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 73

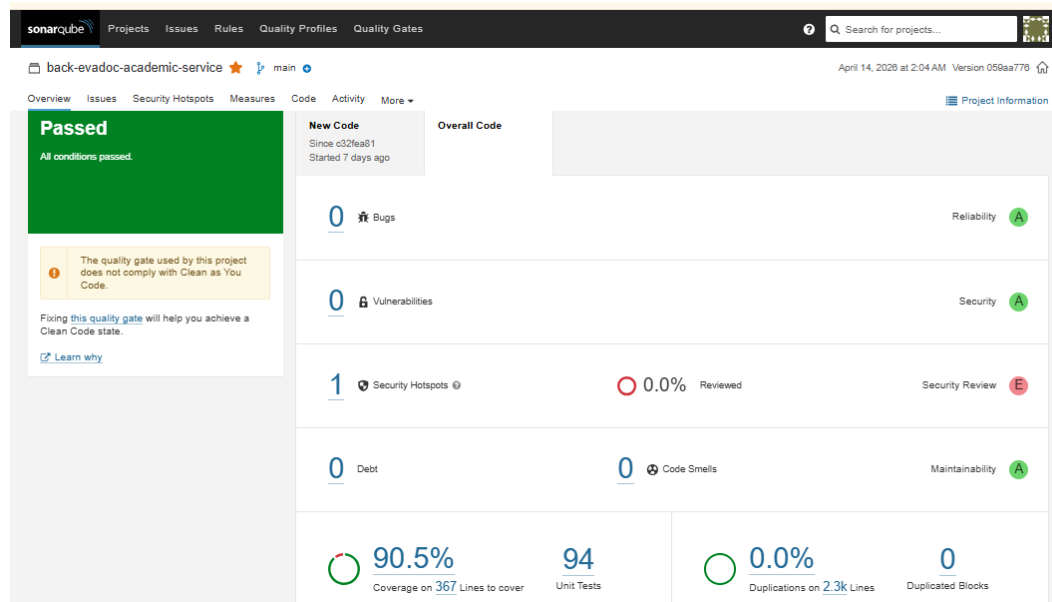
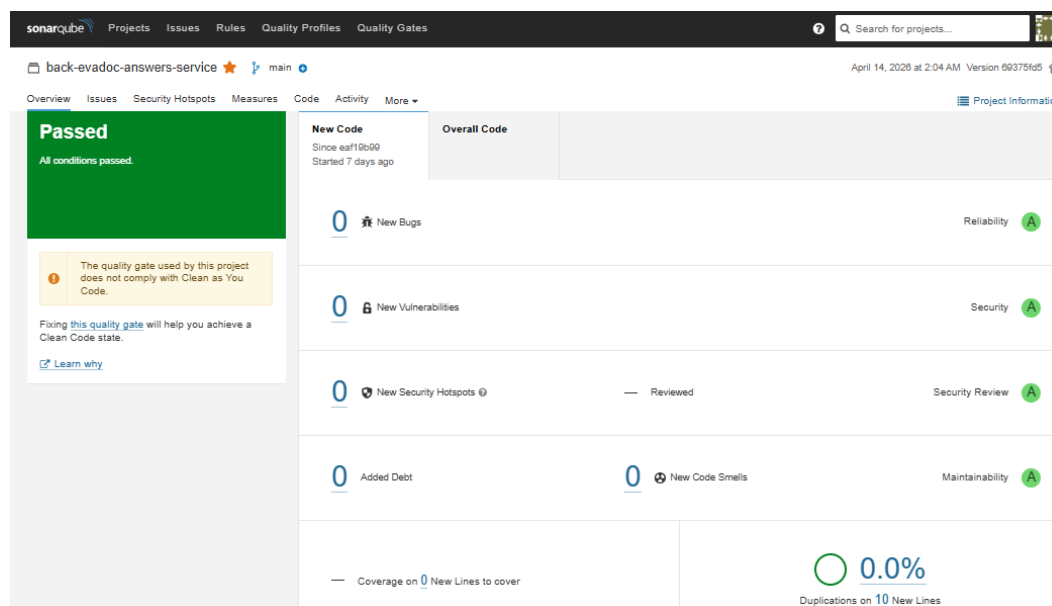
Overall code academic service

Figura 74

Quality gate answer service

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 75

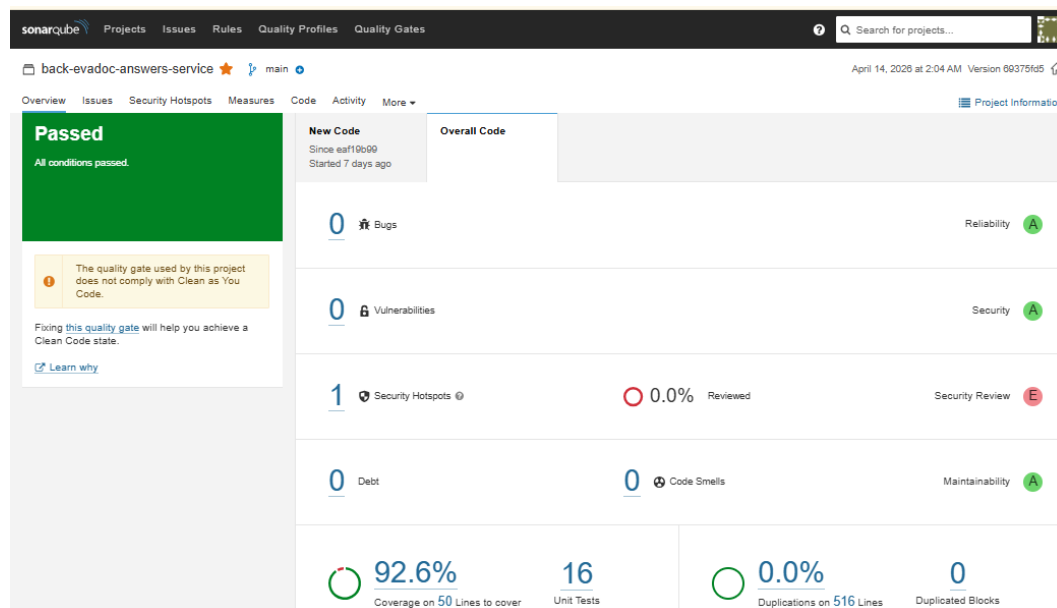
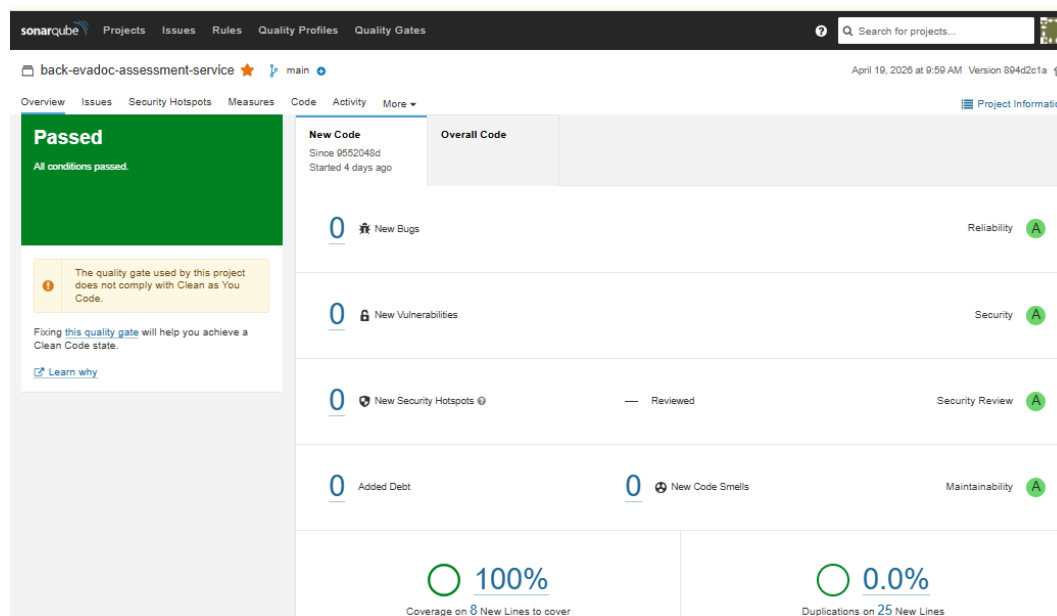
Overall code answer service

Figura 76

Quality gate assessment service

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 77

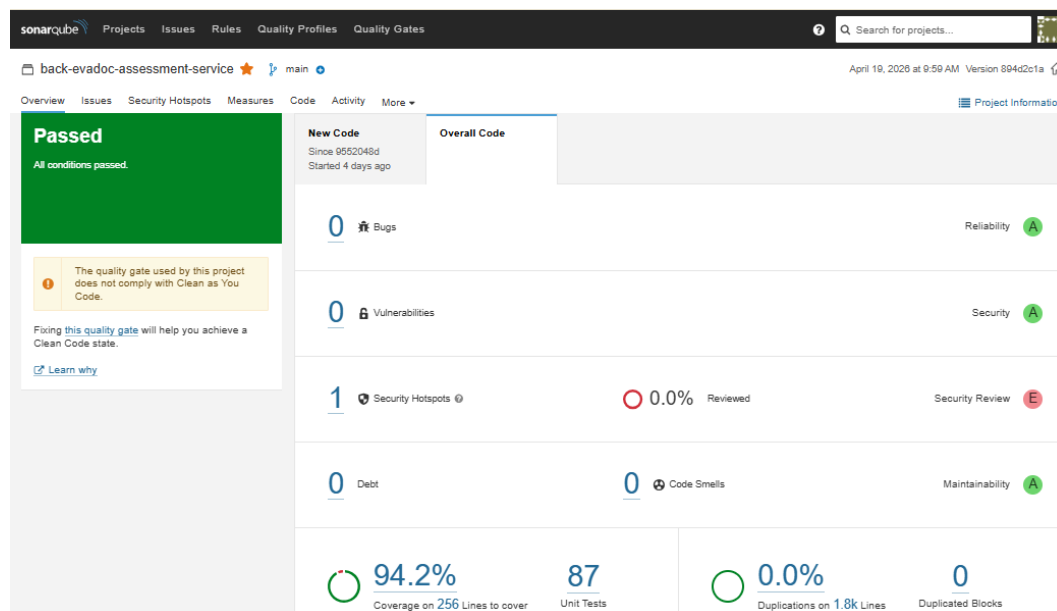
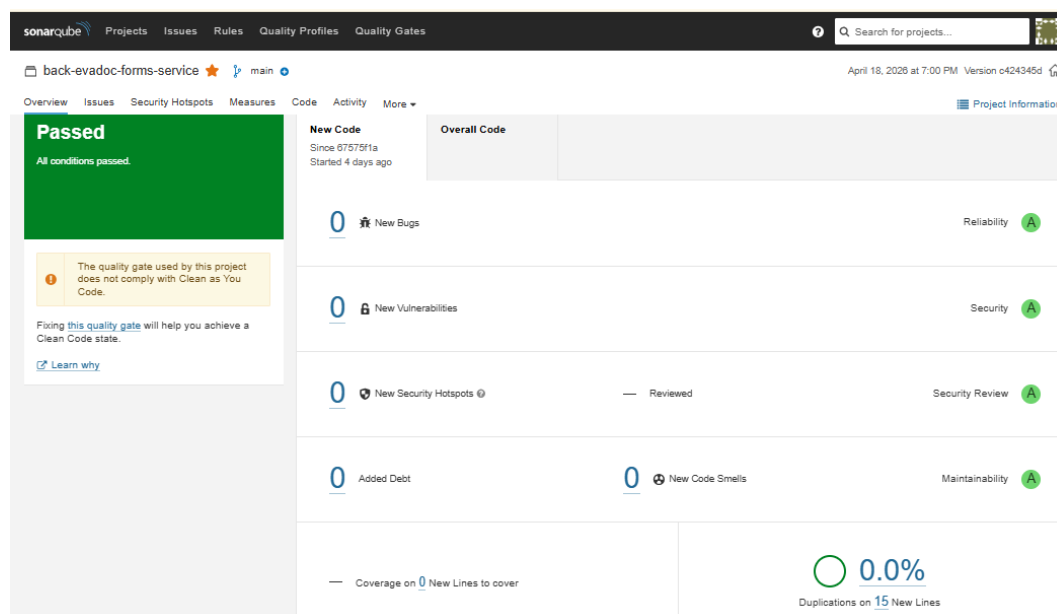
Overall Code assessment service

Figura 78

Quality gate form service

REINGENIERÍA DEL SISTEMA DE EVALUACIÓN DOCENTE

Figura 79

Overall code form service