Universidad Autónoma del Cauca

INGENIERÍA DE SOFTWARE II

PROYECTO DE INGENIERÍA DE SOFTWARE

Evaluación Docente

Autores:

Thomas Montoya Magon

Juan Daniel Bravo

Alejandro Martínez Salazar

Daniel Rivas Agredo

Luisa Julieth Joaqui

Viernes 14 de marzo del 2025

Índice

1. Introducción **2**

2. Objetivos del Proyecto **2**
2.1. Objetivo General . . . 2
2.2. Objetivos Específicos . . . 2

3. Justificación y Alcance **3**
3.1. Justificación . . . 3
3.2. Alcance . . . 3

4. Metodología de Desarrollo **3**

5. Arquitectura del Sistema **3**

6. Cronograma y Plan de Trabajo **4**

7. Roles y Responsabilidades **4**

8. Estrategia de Pruebas y QA **4**

9. Requisitos del Sistema **4**
9.1. Revisión General de Requisitos Funcionales . . . 5
9.2. Revisión General de Requisitos No Funcionales . . . 6

10. Análisis de Riesgos y Plan de Contingencia **6**

11. Limitaciones y Trabajo Futuro **7**
11.1. Limitaciones . . . 7
11.2. Trabajo Futuro . . . 7

12. Impacto en la Calidad Educativa **7**

13. Anexos y Referencias **7**
13.1. Anexos . . . 7
13.2. Sitio Web del Proyecto . . . 8
13.3. Referencias . . . 8

14. Conclusiones **8**

15. Anexos - Diagramas de Caso de Uso **9**

1 Introducción

El proyecto de **Evaluación Docente** consiste en el desarrollo de una aplicación web integral destinada a gestionar y optimizar el proceso de evaluación del desempeño de los docentes en la institución. En esta fase se ha desarrollado únicamente el **frontend** del sistema, utilizando HTML, CSS y JavaScript con datos simulados para demostrar la funcionalidad de los módulos. La integración con un **backend** (usando PHP, Laravel y MySQL) se planificará en futuras iteraciones, permitiendo la persistencia, autenticación avanzada y manejo de datos en producción.

La aplicación contempla tres tipos de evaluaciones:

- **Evaluación Estudiantil:** Los estudiantes evaluarán a sus docentes de forma anónima, asignando calificaciones y dejando comentarios. (Actualmente, esta funcionalidad se simula en el frontend).
- **Evaluación Administrativa:** El Decano o Coordinador podrá evaluar aspectos administrativos y el cumplimiento de obligaciones del docente, aunque en esta fase solo se cuenta con la interfaz de seguimiento y alerta.
- **Autoevaluación Docente:** Se planifica permitir que los docentes se autoevalúen, aunque en la versión actual este módulo no está implementado.

La combinación de estas evaluaciones dará lugar a una nota final (de 0 a 5), determinando si el docente deberá ingresar en un proceso de mejora y, de ser necesario, en un proceso de sanción o retiro.

2 Objetivos del Proyecto

2.1 Objetivo General

Desarrollar una aplicación web que permita gestionar integralmente el proceso de evaluación docente, garantizando confidencialidad, transparencia y mejora continua del desempeño académico. La versión actual se centra en el frontend, con integración futura del backend para la persistencia y procesamiento de datos reales.

2.2 Objetivos Específicos

- Permitir evaluaciones anónimas de los docentes por parte de los estudiantes (simuladas en el frontend).
- Facilitar la evaluación administrativa a cargo del Decano o Coordinador mediante interfaces de seguimiento.
- Planificar un mecanismo de autoevaluación para que los docentes puedan reflexionar sobre su desempeño (pendiente de implementación).
- Combinar las evaluaciones simuladas para generar una nota final, identificando áreas de mejora.
- Implementar un proceso de seguimiento (y eventualmente sanciones) para docentes con bajo rendimiento.
- Garantizar altos estándares de seguridad, rendimiento y disponibilidad, a completar en la fase de backend.

3 Justificación y Alcance

3.1 Justificación

El sistema de Evaluación Docente es esencial para:

- Mejorar la calidad educativa mediante la identificación de áreas de oportunidad en la enseñanza.
- Proporcionar retroalimentación objetiva y anónima que incentive el desarrollo profesional de los docentes.
- Cumplir con las normativas del Ministerio de Educación y las políticas internas de la institución.

3.2 Alcance

La versión actual del proyecto abarca:

- El desarrollo del **frontend** con interfaces de usuario para la autenticación, evaluaciones (simuladas), generación de actas y seguimiento.
- La visualización de reportes, dashboards y alertas basados en datos de ejemplo.
- La planificación para la integración futura del **backend** (PHP, Laravel, MySQL) que permita la persistencia y procesamiento real de datos.

Mejoras adicionales y la integración con otros sistemas institucionales se implementarán en futuras fases.

4 Metodología de Desarrollo

Se utilizará una metodología ágil (por ejemplo, Scrum) para iterar y entregar versiones funcionales del sistema. Los aspectos clave son:

- Planificación y definición de sprints.
- Reuniones diarias de seguimiento.
- Revisión y retrospectiva de cada sprint.
- Integración continua y despliegue automatizado (CI/CD) para la futura fase de backend.

5 Arquitectura del Sistema

La arquitectura del sistema se dividirá en dos capas principales:

- **Frontend:** Desarrollo de la interfaz web utilizando HTML, CSS, JavaScript y frameworks modernos (en esta versión se utiliza una solución basada en tecnologías front-end con datos simulados).
- **Backend (Futuro):** Servicios y API REST que se implementarán en PHP con el framework Laravel, conectados a una base de datos MySQL para el almacenamiento y procesamiento real de la información.

La integración entre ambas capas se realizará mediante una arquitectura modular, facilitando la extensión y mantenimiento del sistema.

6 Cronograma y Plan de Trabajo

El plan de trabajo se estructura en varias fases:

1. **Planificación:** Definición de requisitos, cronograma y elaboración de prototipos.
2. **Diseño:** Diseño de la arquitectura del sistema y elaboración de prototipos de interfaz.
3. **Desarrollo del Frontend:** Implementación de las interfaces de usuario y simulación de datos.
4. **Pruebas:** Ejecución de pruebas unitarias y de integración en el frontend.
5. **Despliegue y Validación:** Implementación en un entorno de prueba, con vistas a la futura integración del backend.
6. **Desarrollo del Backend (Futuro):** Integración de la lógica de negocio, persistencia y servicios.

7 Roles y Responsabilidades

El equipo se ha dividido en los siguientes roles:

- **Frontend:** *Thomas Montoya Magon* y *Juan Daniel Bravo* — Desarrollo de la interfaz de usuario y experiencia interactiva.
- **Backend (Futuro):** *Alejandro Martínez Salazar* y *Luisa Julieth Joaqui* — Implementación de la lógica de negocio y servicios API.
- **Base de Datos:** *Daniel Rivas Agredo* y *Luisa Julieth Joaqui* — Diseño y gestión del almacenamiento de datos (a implementar en el backend).
- **Seguridad:** *Thomas Montoya Magon* — Implementación de mecanismos de autenticación, cifrado y control de acceso (por ahora básicos en el frontend, con mejoras en el backend).

8 Estrategia de Pruebas y QA

Para garantizar la calidad del sistema se implementarán:

- **Pruebas Unitarias:** Validar la funcionalidad de cada componente del frontend.
- **Pruebas de Integración:** Asegurar la correcta interacción entre los módulos de la interfaz.
- **Pruebas End-to-End:** Validar los flujos críticos (por ejemplo, el proceso de evaluación y generación de actas) en el entorno simulado.
- **Revisiones de Código:** Mantener la calidad y consistencia del desarrollo.
- **Pruebas de Usabilidad:** Con usuarios reales para optimizar la experiencia.

En futuras fases, se integrará un proceso formal de QA (incluyendo CI/CD) para el backend.

9 Requisitos del Sistema

El sistema se define mediante dos conjuntos de requisitos:

- **Requisitos Funcionales (RF):** Incluyen módulos de autenticación, evaluación (estudiantil, administrativa, autoevaluación), cálculo de promedios, generación de actas, seguimiento, reportes, gestión de roles y permisos, alertas, auditoría, integración con sistemas externos, etc.
- **Requisitos No Funcionales (RNF):** Abarcan seguridad, disponibilidad, rendimiento, usabilidad, mantenibilidad, escalabilidad, accesibilidad, integridad de datos, interoperabilidad, continuidad ante desastres y aseguramiento de la calidad.

Actualmente, varios de estos requisitos se encuentran en desarrollo parcial o pendientes de integración con el backend. A continuación se presenta un resumen de su estado.

9.1 Revisión General de Requisitos Funcionales

Se han identificado **17 Requisitos Funcionales (RF01 a RF017)**. A continuación, se muestra un *resumen* de cada uno, indicando su importancia, urgencia y estado actual (implementado, parcial, pendiente). Para mayor detalle, consultar la tabla extensa en el Anexo **??**.

- **RF01 - Autenticación e inicio de sesión de Usuarios:** *Alta prioridad.* Implementado parcialmente (autenticación básica, roles fijos). Prevista encriptación y uso de Laravel.
- **RF02 - Datos de Evaluación Estudiantil:** *Alta prioridad.* No implementado. Requiere integración con base de datos real.
- **RF03 - Datos de Evaluación del Decano/Coordinador:** *Alta prioridad.* No implementado. Falta formulario y almacenamiento de datos reales.
- **RF04 - Cálculo de Estadísticas (Decano):** *Alta prioridad.* Implementado parcialmente. Requiere datos de RF02 y RF03.
- **RF05 - Cálculo de Estadísticas (Docente):** *Alta prioridad.* Implementado parcialmente. Faltan datos reales y backend.
- **RF06 - Generación de Acta de Compromiso:** *Alta prioridad.* Implementado (frontend). Falta guardar historial en backend.
- **RF07 - Seguimiento a Plan de Mejora:** *Alta prioridad.* Implementado (frontend). Falta persistencia y notificaciones reales.
- **RF08 - Reportes e Informes de Evaluación:** *Media prioridad.* Implementado (frontend). Faltan exportaciones y datos auténticos.
- **RF09 - Alertas de Bajo Desempeño:** *Alta prioridad.* Implementado (solo en panel Decano). Falta envío de alertas fuera del sistema.
- **RF010 - Visualización de Comentarios (Ciego):** *Alta prioridad.* Implementado (datos ficticios). Pendiente decisión final de confidencialidad.
- **RF011 - Historial de Evaluaciones:** *Media prioridad.* No implementado. Requiere almacenamiento histórico.
- **RF012 - Gestión de Roles y Permisos:** *Alta prioridad.* Implementado parcialmente (interfaz). Falta persistencia y control granular.
- **RF013 - Notificaciones y Alertas Generales:** *Media prioridad.* No implementado (más allá de alertas de bajo desempeño).
- **RF014 - Auditoría y Registro de Eventos:** *Alta prioridad.* No implementado. Pendiente en backend para trazabilidad.

- **RF015 - Módulo de Soporte y Ayuda:** *Baja prioridad.* No implementado. Se planea tras completar la integración principal.
- **RF016 - Panel de Indicadores y Dashboard:** *Media prioridad.* Implementado parcialmente. Usa datos simulados, requiere backend.
- **RF017 - Proceso de Sanciones / Retiro Docente:** *Alta prioridad.* Implementado parcialmente. Falta notificar al docente y formalizar criterios.

9.2 Revisión General de Requisitos No Funcionales

Se han identificado **9 Requisitos No Funcionales (RNF01 a RNF09)**. En general, la mayoría están en fase inicial o pendiente de un backend robusto para su implementación completa. Para más detalle, véase el Anexo **??**.

- **RNF01 - Seguridad y Confidencialidad:** *Alta prioridad.* Parcial. Sin cifrado real ni control avanzado de accesos. Planeado con Laravel/MySQL.
- **RNF02 - Rendimiento:** *Media prioridad.* Parcial. Sin pruebas de carga formales.
- **RNF03 - Usabilidad:** *Media prioridad.* Implementado parcialmente. Faltan pruebas de accesibilidad.
- **RNF04 - Mantenibilidad:** *Media prioridad.* Parcial. Falta documentación formal y pruebas unitarias extensas.
- **RNF05 - Escalabilidad:** *Media prioridad.* No implementado. Se planea en la arquitectura backend.
- **RNF06 - Cumplimiento Normativo:** *Alta prioridad.* No implementado. Pendiente validación legal (protección de datos).
- **RNF07 - Interoperabilidad y Extensibilidad:** *Media prioridad.* Parcial. Falta integración real con otros sistemas (Moodle, etc.).
- **RNF08 - Plan de Continuidad y Recuperación:** *Alta prioridad.* No implementado. Solo se menciona uso de GitHub para copias.
- **RNF09 - Aseguramiento de la Calidad (QA y Testing):** *Media prioridad.* Parcial. Faltan pruebas automatizadas y CI/CD.

10 Análisis de Riesgos y Plan de Contingencia

Se han identificado los siguientes riesgos:

- **Riesgo Técnico:** Dificultades en la integración entre frontend y el futuro backend.
  Mitigación: Pruebas de integración y uso de CI/CD en la fase de backend.
- **Riesgo de Seguridad:** Vulnerabilidades por falta de cifrado y mecanismos robustos de autenticación en la versión actual.
  Mitigación: Mejorar la seguridad en la fase backend, siguiendo estándares OWASP.
- **Riesgo de Disponibilidad:** Interrupción del servicio al integrar la solución en producción.
  Mitigación: Uso de infraestructura en la nube con redundancia y planes de contingencia.
- **Riesgo de Usabilidad:** La interfaz podría no ser completamente intuitiva.
  Mitigación: Realizar pruebas de usabilidad y ajustar el diseño basado en retroalimentación de usuarios.

11 Limitaciones y Trabajo Futuro

11.1 Limitaciones

- **Integración con sistemas externos:** La versión inicial se limita al frontend; la integración completa se realizará en el backend.
- **Escalabilidad:** El diseño actual corresponde a un MVP, ampliable en futuras iteraciones.
- **Interfaz de Usuario:** Mejoras en el diseño se implementarán conforme se reciba retroalimentación.

11.2 Trabajo Futuro

- Integrar el backend usando PHP, Laravel y MySQL para la persistencia de datos.
- Implementar módulos de evaluación real (estudiantil, administrativa y autoevaluación) con almacenamiento seguro.
- Desarrollar un sistema robusto de notificaciones, auditoría y monitorización.
- Ampliar la integración con otros sistemas académicos y administrativos.
- Desarrollar aplicaciones móviles o adaptaciones para dispositivos móviles.

12 Impacto en la Calidad Educativa

El sistema de Evaluación Docente tendrá un impacto positivo al:

- Proporcionar retroalimentación objetiva y anónima que incentive el desarrollo profesional de los docentes.
- Facilitar la toma de decisiones estratégicas para la mejora continua de la enseñanza.
- Asegurar el cumplimiento de normativas y políticas institucionales.
- Promover una cultura de transparencia y rendición de cuentas en el ámbito académico.

13 Anexos y Referencias

13.1 Anexos

- **Diagramas UML:** Representación gráfica de la arquitectura y flujos de datos.
- **Tablas Detalladas de Requisitos Funcionales (RF) y No Funcionales (RNF):**
  - **Edición en línea:** Editar.
- **Prototipos de Interfaz:** Capturas y esquemas del diseño de la interfaz de usuario.

13.2 Sitio Web del Proyecto

Para más información y para ver el proyecto en acción, visita: https://slarkz01.github.io/Proyecto_Evaluacion_Docente/

Repositorio del proyecto en GitHub: https://github.com/SLarkZ01/Proyecto_Evaluacion_Docente

13.3 Referencias

Las principales fuentes consultadas son:

- Normativas del Ministerio de Educación.
- Políticas de seguridad y estándares (OWASP, WCAG).
- Manual de estilo de la Universidad Autónoma del Cauca.

14 Conclusiones

El desarrollo de la plataforma de Evaluación Docente, aun en su fase de frontend, sienta las bases para una gestión integral y transparente del desempeño académico. La versión actual, con funcionalidades simuladas, permite validar la interfaz y la experiencia de usuario, mientras que la futura integración del backend aportará la persistencia, seguridad y procesamiento real de los datos. Este proyecto, al combinar altos estándares de calidad y una metodología ágil, contribuirá significativamente a la mejora continua de la calidad educativa en la universidad.

15 Anexos - Diagramas de Caso de Uso

A continuación se presentan los diagramas de caso de uso que describen la interacción de los distintos roles:

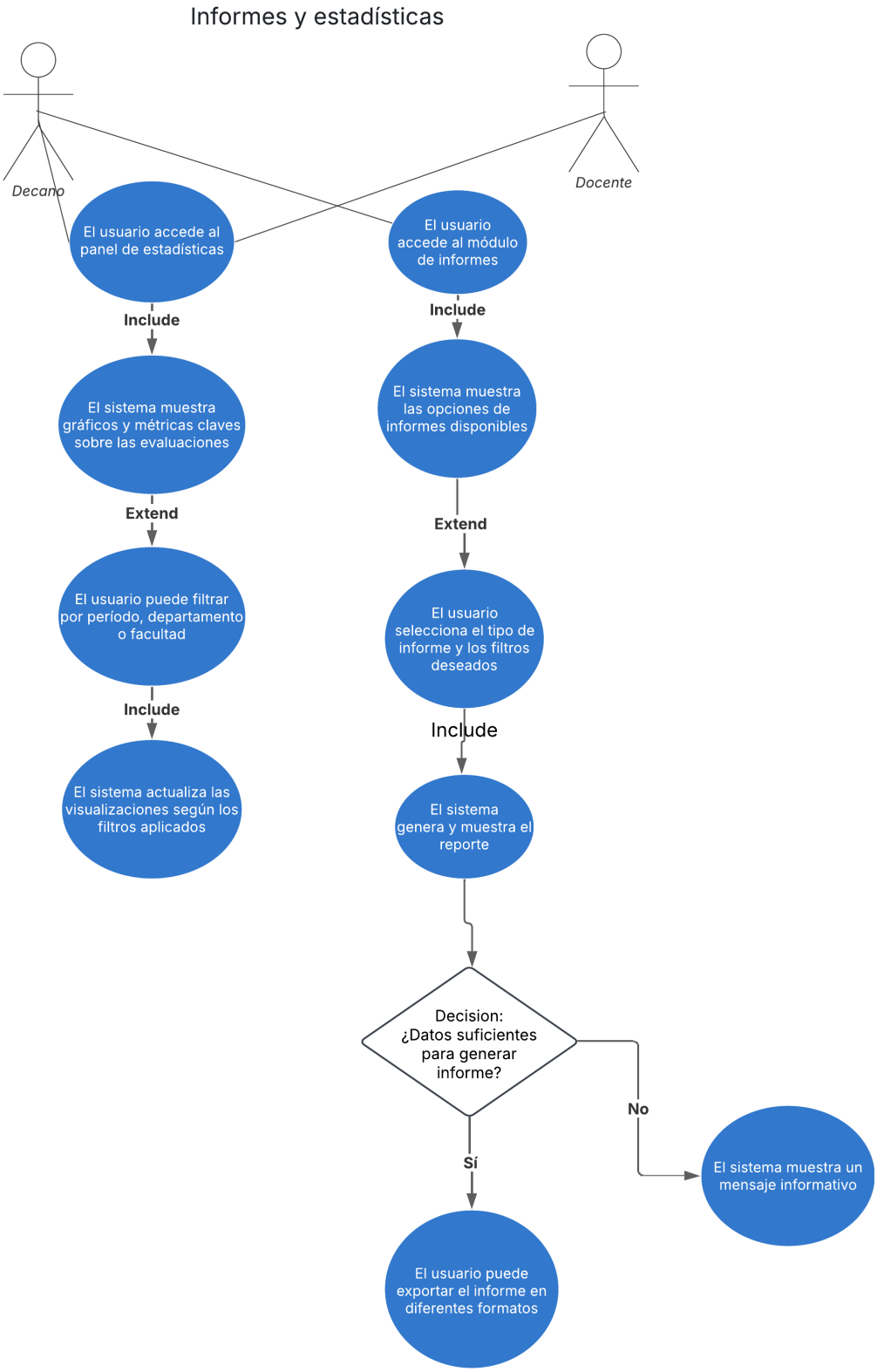

Figura 1: Informes y estadisticas.

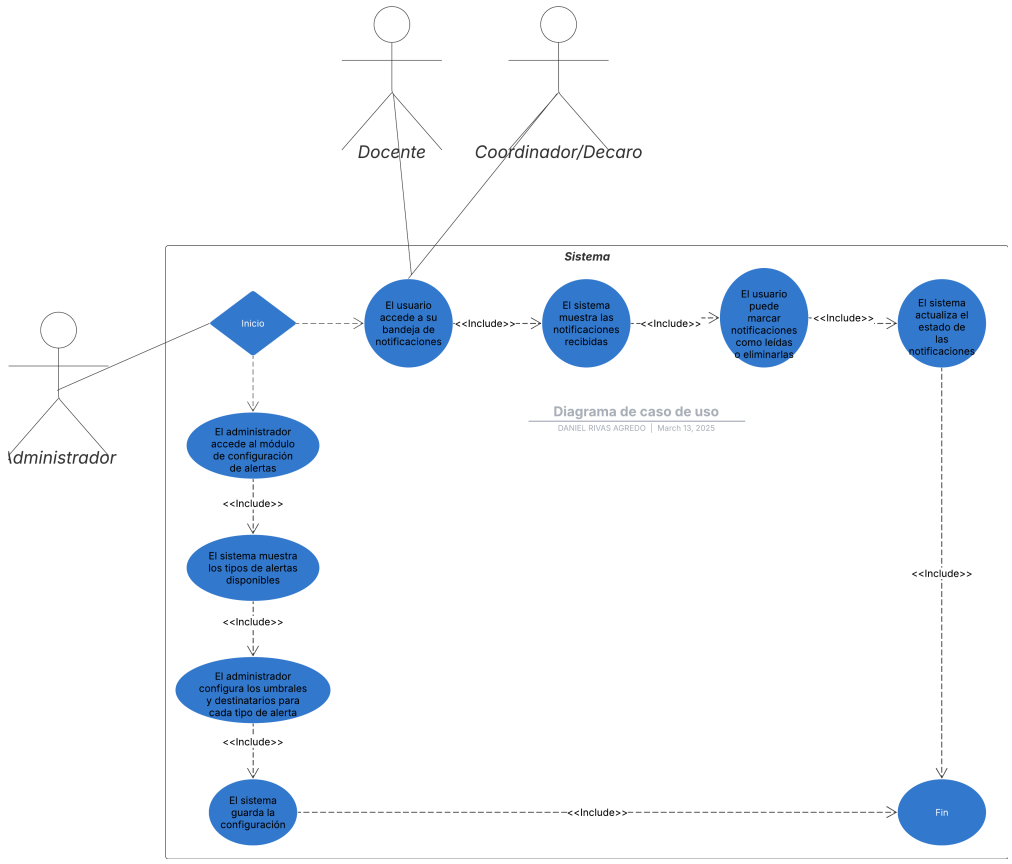

Figura 2: Gestion de alertas y notificaciones

Gestión de Planes de Mejora

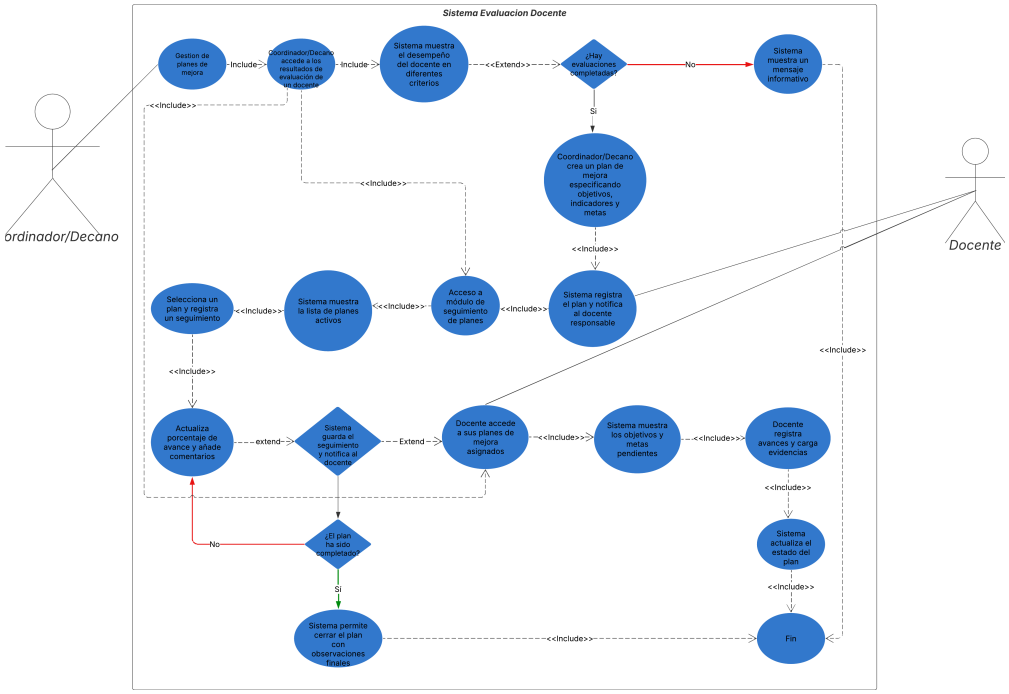

Figura 3: Gestion de planes de mejora

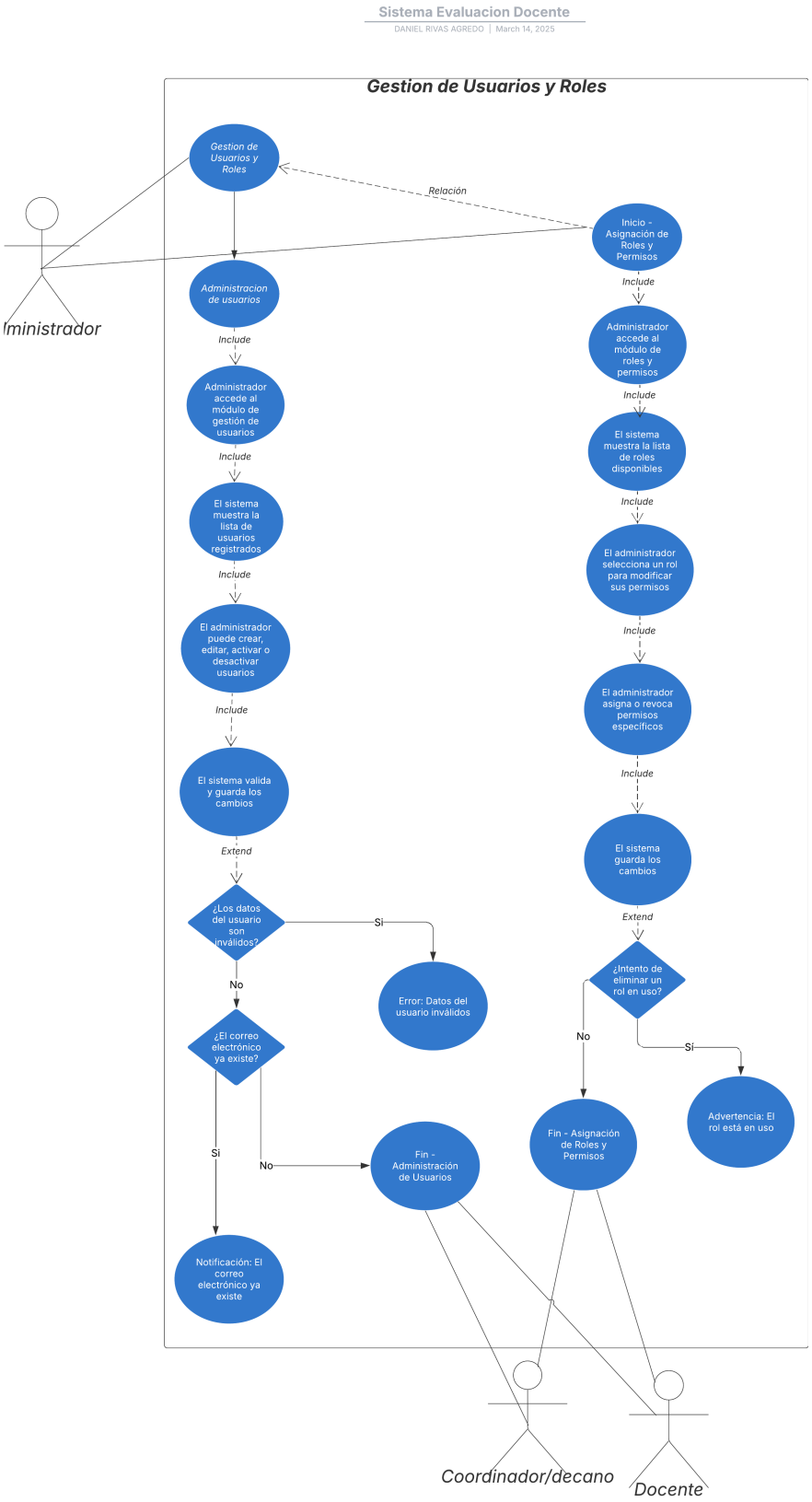

Figura 4: Gestion de roles y informes

Actas de compromiso

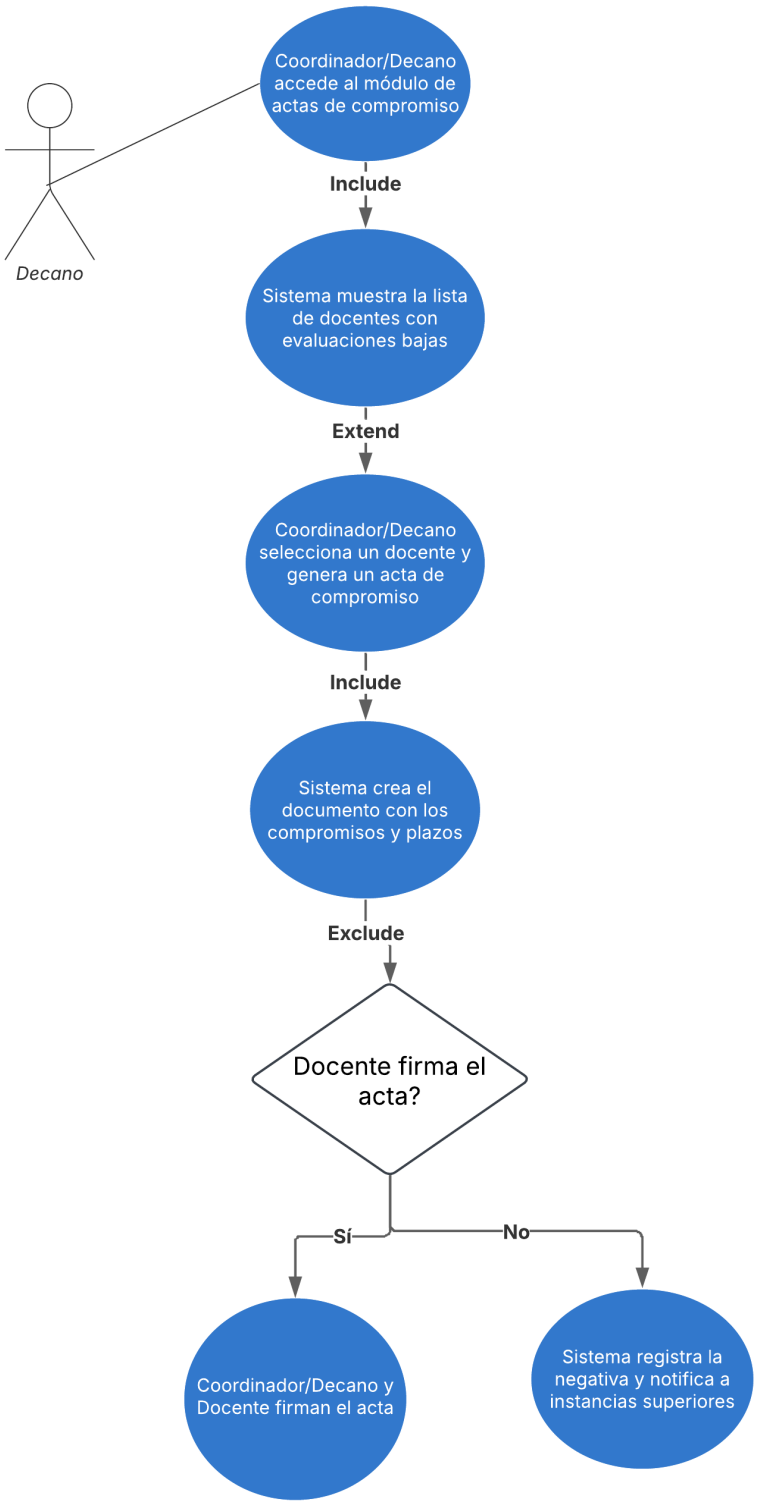

Figura 5: Acta de compromiso