TECNOLOGÍAS DE INFORMACIÓN Y CIBERSEGURIDAD

Inacap

INFORME TÉCNICO

Plataforma de Gestión de Cursos Online

GRUPO	04
NOMBRES	Sebastián Araujo, Gabriela Lettieri, Antonio Morales Laura Núñez, Francisco Palacios, Joaquín Sepúlveda
CARRERA	Ingeniería Informática
ASIGNATURA	[TI3V43] Ingeniería de Software
PROFESOR	Eugenio Alberto Bravo García
FECHA	28-05-25

Índice

1. Introducción **5**

2. Análisis del Contexto y Alcance del Problema **6**
- 2.1. Contexto . . . 6
- 2.2. Objetivo . . . 6
- 2.3. Alcance . . . 6
- 2.4. Funcionalidades principales . . . 6

3. Proceso de Cocreación **7**
- 3.1. Marco Metodológico . . . 7
- 3.2. Participantes y Perfil de Usuario . . . 7
- 3.3. Fases del Proceso de Cocreación . . . 7
- 3.4. Impacto en el Diseño del Sistema . . . 8
- 3.5. Integración con Metodología de Desarrollo . . . 9
- 3.6. Métricas de Validación . . . 9
- 3.7. Conclusiones de la Cocreación . . . 9

4. Metodología y Planificación **11**
- 4.1. Roles y Responsabilidades . . . 11
- 4.2. Estructura del Proyecto en 3 Fases . . . 12
- 4.3. Plan de Tareas Detallado (WBS) . . . 12
- 4.4. Cronograma y Estimaciones . . . 22
- 4.5. Ceremonias Scrum . . . 22
- 4.6. Ruta Crítica del Proyecto . . . 23
- 4.7. Criterios de Éxito y Control de Calidad . . . 24

5. Requerimientos **25**
- 5.1. Historias de Usuario (HU) . . . 25
- 5.2. Requerimientos Funcionales . . . 26
- 5.3. Requerimiento No Funcionales . . . 28

6. Diagramas UML **30**
- 6.1. Diagrama de Caso de Uso . . . 30
- 6.2. Diagrama de Clases . . . 32
- 6.3. Diagramas de Secuencia . . . 34
- 6.4. Diagrama de Actividad: Flujo de curso online . . . 57

7. Patrones de Diseño Aplicados **59**
- 7.1. Patrón Observer - Sistema de Notificaciones . . . 59
- 7.2. Patrón Singleton - Gestión de Configuraciones . . . 59
- 7.3. Patrón Factory - Creación de Contenido Multimedia . . . 60
- 7.4. Justificación de Patrones No implementados . . . 60

8. Estándares de Calidad **61**
8.1. ISO/IEC 12207 - Procesos del Ciclo de Vida del Software 61
8.2. ISO/IEC 25000 (SQuaRE) - Atributos de Calidad 61

9. Matriz de Trazabilidad (RTM) **63**
9.1. Identificación de Riesgos 65
9.2. Matriz 3x3 67

10. Conclusiones **68**

11. Bibliografía **69**

Índice de figuras

1. Diagrama de Caso de Uso 31
2. Diagrama de Clases 33
3. Diagrama de Secuencia - Usuario (Registrarse) 35
4. Diagrama de Secuencia - Usuario (Iniciar Sesión) 36
5. Diagrama de Secuencia - Usuario (Recuperar Contraseña) 37
6. Diagrama de Secuencia - Usuario (Confirmar Nueva Contraseña) 38
7. Diagrama de Secuencia - Usuario (Cerrar Sesión) 39
8. Diagrama de Secuencia - Estudiante (Buscar Curso) 40
9. Diagrama de Secuencia - Estudiante (Inscribir Curso) 41
10. Diagrama de Secuencia - Estudiante (Realizar Quiz) 42
11. Diagrama de Secuencia - Estudiante (Completar Curso) 43
12. Diagrama de Secuencia - Docente (Crear Curso) 44
13. Diagrama de Secuencia - Docente (Agregar Módulo al Curso) 45
14. Diagrama de Secuencia - Docente (Subir material al Módulo) 46
15. Diagrama de Secuencia - Docente (Crear Evaluaciones) 47
16. Diagrama de Secuencia - Docente (Ver Estudiantes) 48
17. Diagrama de Secuencia - Docente (Revisar y Calificar Evaluaciones) 49
18. Diagrama de Secuencia - Administrador (Ver Usuarios) 50
19. Diagrama de Secuencia - Administrador (Crear Usuario) 51
20. Diagrama de Secuencia - Administrador (Editar Usuario) 52
21. Diagrama de Secuencia - Administrador (Solicitar Reporte del Sistema) 53
22. Diagrama de Secuencia - Administrador (Generar Reporte de Usuarios) 54
23. Diagrama de Secuencia - Administrador (Generar Reporte de Cursos) 55
24. Diagrama de Secuencia - Administrador (Generar Reporte de Certificados) . . . 56
25. Diagrama de Actividad 58
26. Matriz 3×3 de Evaluación de Riesgos del Proyecto 67

Índice de cuadros

1. Perfil de Participantes en Cocreación 7
2. Modificaciones Derivadas del Proceso de Cocreación 8
3. Métricas de Validación del Proceso de Cocreación 9
4. Distribución de Roles y Cargas de Trabajo 11
5. Work Breakdown Structure Detallado 21
6. Distribución de Carga de Trabajo 22
7. Ceremonias Scrum por Fase 22
8. Dependencias Críticas del WBS 23
9. Historias de Usuario 25
10. Requerimientos Funcionales 28
11. Requerimientos No Funcionales 29
12. Procesos de Implementación en el Proyecto 61
13. Características de Calidad del Software 62
14. Requisitos y Diagramas Relacionados 64
15. Análisis de Riesgos del Proyecto 66

1 Introducción

El presente informe tiene como propósito documentar de manera estructurada el desarrollo de una plataforma de gestión de cursos online, diseñada en el marco de la asignatura de Ingeniería de Software. Este trabajo busca aplicar los contenidos vistos en clase, integrando prácticas de levantamiento de requerimientos, planificación ágil, modelamiento UML, patrones de diseño y estándares de calidad, siguiendo las mejores prácticas establecidas en la literatura especializada (Sommerville, 2016).

La plataforma está orientada a estudiantes y docentes, y tiene como objetivo facilitar la inscripción a cursos, el acceso a materiales de aprendizaje, la evaluación académica y la emisión de certificados digitales verificables. Como valor agregado, incorpora un módulo de inteligencia artificial para la generación de resúmenes automáticos de clases y el registro del historial de aprendizaje, aplicando principios modernos de ingeniería de software (Pressman & Maxim, 2014).

El informe se organiza en apartados que abarcan desde el análisis del contexto y alcance del problema, pasando por la metodología y planificación del proyecto, el levantamiento de requerimientos, la representación mediante diagramas UML, la aplicación de patrones de diseño, la implementación de estándares de calidad y la matriz de trazabilidad, hasta concluir con el análisis de riesgos y las conclusiones generales.

2 Análisis del Contexto y Alcance del Problema

2.1 Contexto

Una institución educativa requiere una plataforma web de gestión de cursos online que permita a los estudiantes inscribirse, acceder a contenidos académicos y obtener certificados al finalizar. El sistema debe garantizar seguridad, disponibilidad y facilidad de uso, entregando soporte tanto a estudiantes como a docentes. En el contexto actual de la educación digital (Aparicio et al., 2016), las plataformas de aprendizaje online se han convertido en herramientas fundamentales para la gestión educativa moderna. Adicionalmente, se considera como valor agregado una funcionalidad de inteligencia artificial para generar resúmenes automáticos de clases y mantener un historial de aprendizaje.

2.2 Objetivo

Desarrollar una plataforma que facilite la gestión integral de cursos, materiales, evaluaciones y certificaciones, mejorando la experiencia académica de los estudiantes y optimizando el trabajo de los docentes, aplicando metodologías reconocidas de desarrollo de software (Bourque & Fairley, 2014).

2.3 Alcance

- Registro e inicio de sesión mediante cuenta institucional, Google u otros servicios.
- Acceso a distintos tipos de cursos con la capacidad de registrar avances en los mismos finalizando con un examen para optar a certificación.
- Generar certificados habilitando descarga de los mismos y envío por correo electrónico.
- Soporte de una inteligencia artificial que brinde resúmenes y un historial de clases.
- Garantizar ligereza, seguridad y facilidad de uso tanto para estudiantes como para docentes.

2.4 Funcionalidades principales

- **Estudiantes:** Autenticación, inscripción a cursos, acceso a materiales, evaluaciones y notas, progreso, certificados, notificaciones, historial y resúmenes IA.
- **Docentes:** Creación y gestión de cursos, subida de materiales, evaluaciones y notas, emisión de certificados, notificaciones.
- **Administradores:** Gestión de usuarios, roles, contenidos, parámetros de certificación, monitoreo básico.

3 Proceso de Cocreación

La cocreación constituye una metodología fundamental para garantizar que los sistemas educativos respondan a necesidades reales de los usuarios finales (Sanders & Stappers, 2008). En este proyecto se implementó un proceso estructurado de cocreación que involucró a estudiantes, docentes y administradores desde las etapas tempranas del diseño.

3.1 Marco Metodológico

Se adoptó el modelo de cocreación de Sanders y Stappers (Sanders & Stappers, 2008), estructurado en tres niveles de participación:

- **Hacer (Making):** Talleres prácticos de prototipado colaborativo
- **Decir (Saying):** Entrevistas estructuradas y focus groups
- **Usar (Using):** Pruebas de usabilidad y observación contextual

3.2 Participantes y Perfil de Usuario

El proceso involucró representantes de los tres perfiles principales de usuario:

Perfil	Participantes	Representación
Estudiantes	8-12	Diferentes niveles académicos y experiencia tecnológica
Docentes	6-8	Diversas áreas disciplinares y años de experiencia
Administradores	2-3	Personal de coordinación académica y sistemas

Cuadro 1: Perfil de Participantes en Cocreación

3.3 Fases del Proceso de Cocreación

Fase 1: Levantamiento de Necesidades

Métodos aplicados:

- Entrevistas semi-estructuradas individuales
- Mapeo de experiencias de usuario (User Journey Mapping)
- Identificación de puntos de dolor y oportunidades de mejora

Resultados: Identificación de requerimientos funcionales prioritarios y restricciones de usabilidad críticas que informaron el diseño posterior.

Fase 2: Co-diseño de Soluciones

Métodos aplicados:

- Talleres de ideación colaborativa (Design Thinking)
- Prototipado de papel y wireframing participativo
- Validación temprana de conceptos mediante votación estructurada

Resultados: Generación de conceptos de diseño validados por usuarios y priorización de funcionalidades según valor percibido.

Fase 3: Validación y Refinamiento

Métodos aplicados:

- Testing de usabilidad con prototipos funcionales
- Evaluación mediante System Usability Scale (SUS)
- Iteración basada en feedback estructurado

Resultados: Validación de decisiones de diseño y identificación de ajustes necesarios antes de la implementación.

3.4 Impacto en el Diseño del Sistema

Modificaciones a Requerimientos

El proceso de cocreación resultó en ajustes significativos a los requerimientos iniciales:

Tipo	Requerimiento	Justificación
Modificado	RF5: Visualización de progreso	Inclusión de elementos visuales motivacionales según feedback de usuarios
Modificado	RF8: Sistema de notificaciones	Implementación de priorización inteligente vs. notificaciones genéricas
Agregado	RF18: Dashboard personalizable	Demanda consistente de personalización según rol y preferencias
Agregado	RF19: Autoguardado en evaluaciones	Prevención de pérdida de progreso identificada como riesgo crítico

Cuadro 2: Modificaciones Derivadas del Proceso de Cocreación

Decisiones de Diseño Informadas

Las siguientes decisiones de diseño se basaron directamente en insights del proceso de cocreación:

- **Navegación adaptativa:** Interfaz que se ajusta según el rol del usuario y tareas más frecuentes

- **Progreso visual:** Sistema de indicadores de avance con elementos de gamificación básica
- **Flujo de inscripción simplificado:** Reducción de pasos basada en testing de usabilidad
- **Dashboard contextual:** Información relevante priorizada según momento del semestre académico

3.5 Integración con Metodología de Desarrollo

El proceso de cocreación se integró con la metodología Scrum mediante:

- **Sprint 1-2:** Incorporación de insights de cocreación en Product Backlog inicial
- **Sprints intermedios:** Validación iterativa de funcionalidades con subset de usuarios
- **Sprint Reviews:** Participación de usuarios en demostraciones y feedback
- **Retrospectivas:** Evaluación de efectividad del proceso colaborativo

3.6 Métricas de Validación

Se establecieron métricas para evaluar la efectividad del proceso de cocreación:

Métrica	Objetivo	Método de Medición
Tasa de completitud de tareas	$\geq 85\,\%$	Testing de usabilidad con escenarios reales
System Usability Scale (SUS)	≥ 70 puntos	Cuestionario estandarizado post-testing
Alineación con necesidades	$\geq 80\,\%$ funcionalidades validadas	Mapeo de requerimientos vs. feedback de usuarios
Satisfacción del proceso	$\geq 4/5$ puntos	Evaluación de metodología de cocreación

Cuadro 3: Métricas de Validación del Proceso de Cocreación

3.7 Conclusiones de la Cocreación

El proceso de cocreación implementado demostró su valor en:

1. **Reducción de riesgos:** Identificación temprana de problemas de usabilidad y aceptación
2. **Alineación de producto:** Funcionalidades validadas directamente por usuarios objetivo
3. **Calidad de diseño:** Decisiones de UX/UI informadas por evidencia empírica
4. **Facilidad de adopción:** Mayor probabilidad de aceptación por participación en diseño

Esta metodología aseguró que el desarrollo técnico posterior se basara en necesidades reales documentadas, reduciendo el riesgo de retrabajos y mejorando la probabilidad de éxito en la implementación final del sistema.

4 Metodología y Planificación

Para el desarrollo de la plataforma se adopta el marco ágil Scrum (Schwaber & Sutherland, 2020), estructurado en 12 sprints organizados en tres fases que garantizan un desarrollo incremental estructurado del producto. La planificación se basa en un Work Breakdown Structure (WBS) detallado que incluye 101 tareas específicas con estimaciones precisas, responsables asignados y dependencias claramente definidas, siguiendo las mejores prácticas de gestión de proyectos ágiles (Rubin, 2012).

Esta aproximación granular permite un control estricto sobre la calidad y el cumplimiento de requerimientos, facilitando la gestión de riesgos sprint a sprint y asegurando que cada iteración entregue funcionalidades concretas y medibles. El proyecto contempla un total estimado de 650+ horas de desarrollo distribuidas equitativamente entre los 12 sprints, implementando principios del desarrollo lean (Poppendieck & Poppendieck, 2003).

4.1 Roles y Responsabilidades

La estructura del equipo se mantiene constante durante los 12 sprints, con roles específicos definidos para optimizar la productividad y mantener la calidad del código, aplicando las mejores prácticas documentadas en la literatura sobre equipos ágiles (Kniberg, 2015).

Rol	Responsabilidades Principales	Tareas Específicas en WBS	Carga Estimada
PO/UX	Product Owner + UX Research	Levantamiento HU, feedback usuarios, refinamiento backlog	80-100h total
Backend Lead	Arquitectura y Backend Core	Setup inicial, modelos, APIs, seguridad, integración IA	140-160h total
Backend	Desarrollo Backend Específico	Endpoints, lógica de negocio, evaluaciones, reportes	120-140h total
Frontend	Interfaces y UX/UI	Navegación, panels, móvil, accesibilidad, componentes	130-150h total
Fullstack	Integración y DevOps	Testing, CI/CD, backups, integración completa, datos demo	140-160h total
QA	Quality Assurance	Testing manual/automatizado, UAT, validaciones	60-80h total

Cuadro 4: Distribución de Roles y Cargas de Trabajo

4.2 Estructura del Proyecto en 3 Fases

El proyecto se organiza en tres fases coherentes basadas en el WBS detallado, permitiendo entregas progresivas y construcción incremental de funcionalidades.

Fase 1: Fundación y Core (Sprints 1-4)

Objetivo: Establecer la base técnica sólida y implementar funcionalidades core operativas.

Fase 2: Funcionalidades Avanzadas (Sprints 5-8)

Objetivo: Incorporar características diferenciadas y capacidades avanzadas del sistema.

Fase 3: Calidad y Producción (Sprints 9-12)

Objetivo: Asegurar calidad enterprise, rendimiento y preparación para producción.

4.3 Plan de Tareas Detallado (WBS)

La tabla presenta la descomposición del proyecto en 12 sprints de 1 semana. Cada fila corresponde a una tarea derivada de historias de usuario y requisitos priorizados.

- Las filas **"SPRINT X – . . . "** son solo separadores visuales.
- La columna **Hrs** expresa esfuerzo en horas de foco (no días) y se revisa en cada Sprint Planning.
- La columna **Responsable** indica el rol líder de la entrega, sin excluir colaboración de otros roles.
- La columna **Depende** registra pre-requisitos técnicos/funcionales para ordenar el trabajo, identificar ruta crítica y permitir paralelización.

Todas las semanas incluyen actividades de descubrimiento/feedback con usuarios para refinar criterios de aceptación y mantener la trazabilidad HU $\rightarrow$ requisitos $\rightarrow$ tareas $\rightarrow$ sprint.

Esta planificación sirve como base para seguimiento (burndown/velocidad), control de riesgos y verificación de cumplimiento (DoD/DoR) sin alterar la duración fija de cada sprint.

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
FASE 1: FUNDACIÓN Y CORE						
SPRINT 1 - Base del Sistema y descubrimiento inicial						
1	S1-01	Levantamiento inicial y mapeo HU v1	Entrevistas rápidas, priorización de HU y definición de criterios de aceptación del MVP	PO/UX	6	—
1	S1-02	Setup inicial	Proyecto Django, repo Git, CI básico y estructura	Backend Lead	4	S1-01
1	S1-03	Modelos básicos	Usuario, Curso, Módulo, Inscripción	Backend Lead	8	S1-02
1	S1-04	Autenticación simple	Login/registro con email y contraseña	Backend	6	S1-03
1	S1-05	CRUD cursos básico	Crear, editar, listar cursos (solo docentes)	Backend	8	S1-03
1	S1-06	Catálogo público	Vista de cursos disponibles para estudiantes	Frontend	6	S1-05
1	S1-07	Inscripción básica	Estudiante se inscribe a curso	Backend	6	S1-05
1	S1-08	Navegación base	Menú, rutas principales, layout responsivo	Frontend	8	S1-04
1	S1-09	Subir contenido PDF	Docente sube archivos PDF por módulo	Fullstack	8	S1-05
1	S1-10	Visualizar contenido	Estudiante ve PDFs y enlaces de sus cursos	Frontend	6	S1-09

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
1	S1-11	Datos de prueba	Usuarios, cursos y contenidos demo	Fullstack	4	S1-07
SPRINT 2 - Evaluación y Progreso						
2	S2-01	Modelo de quiz	Preguntas opción múltiple, respuestas	Backend	8	S1-03
2	S2-02	Crear quiz	Docente crea evaluaciones simples	Backend	8	S2-01
2	S2-03	Rendir quiz	Estudiante responde preguntas	Frontend	10	S2-01
2	S2-04	Calificación automática	Corrección y cálculo de nota	Backend	6	S2-03
2	S2-05	Progreso del estudiante	% de módulos vistos y nota promedio	Fullstack	8	S2-04
2	S2-06	Panel estudiante	Ver cursos inscritos, progreso y notas	Frontend	8	S2-05
2	S2-07	Panel docente básico	Ver estudiantes inscritos y sus notas	Frontend	8	S2-04
2	S2-08	Validaciones y seguridad básica	Permisos básicos, validación de archivos	Fullstack	6	Varias
2	S2-09	Testing básico	Pruebas manuales de flujos principales	QA	8	Varias
2	S2-10	Feedback usuarios v2	Mini pruebas con 3–5 usuarios y refinamiento de HU	PO/UX	6	Varias

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
SPRINT 3 - Certificados y Administración básica						
3	S3-01	Reglas de aprobación	Configurar umbrales (nota/progreso)	Fullstack	6	S2-05
3	S3-02	Generar certificado PDF + QR	Emisión automática al aprobar	Backend	10	S3-01
3	S3-03	Verificación y revocación	URL pública, revocar/validar certificados	Backend	8	S3-02
3	S3-04	Notificaciones básicas	Panel/email para eventos clave	Backend	8	S2-04
3	S3-05	Panel admin básico	Listar/crear/editar usuarios	Frontend	8	S1-04
3	S3-06	Gestión cursos admin	Activar/desactivar, estadísticas	Frontend	6	S3-05
3	S3-07	Testing certificados	E2E flujo completo	QA	8	S3-03
3	S3-08	Documentación usuarios v1	Guías básicas	PO/UX	4	Varias
SPRINT 4 - Historial, Búsqueda y Roles						
4	S4-01	Sistema de roles avanzado	Permisos granulares	Backend	10	S1-04
4	S4-02	Historial académico detallado	Registro completo de actividades del estudiante	Backend	8	S2-05
4	S4-03	Búsqueda avanzada catálogo	Filtros, etiquetas y ordenamiento múltiple	Frontend	8	S1-06

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
4	S4-04	Prerrequisitos y cupos de curso	Validación automática de requisitos y límites	Backend	6	S1-07
4	S4-05	Lista de espera automática	Gestión automática cuando se agota cupo	Backend	6	S4-04
4	S4-06	Reportes básicos admin	Reportes de usuarios, cursos y certificados	Backend	8	S3-05
4	S4-07	Interfaz historial estudiante	Vista cronológica de logros y progreso	Frontend	8	S4-02
4	S4-08	Testing roles y permisos	Verificación de control de acceso	QA	6	S4-01
FASE 2: FUNCIONALIDADES AVANZADAS						
SPRINT 5 - Móvil, Accesibilidad y Medios						
5	S5-01	Optimización móvil completa	Responsive avanzado y touch gestures	Frontend	10	S1-08
5	S5-02	Accesibilidad WCAG 2.1 AA básica	Cumplimiento estándares de accesibilidad	Frontend	8	S5-01
5	S5-03	Player de video integrado	Reproductor con controles y subtítulos	Frontend	8	S1-10
5	S5-04	Gestión multimedia avanzada	Soporte múltiples formatos y compresión	Fullstack	8	S1-09
5	S5-05	Compresión y optimización archivos	Reducción automática de tamaño de archivos	Backend	6	S5-04
5	S5-06	PWA básica	Funcionalidad offline parcial	Frontend	8	S5-01

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
5	S5-07	Testing móvil y accesibilidad	Pruebas en dispositivos y herramientas de accesibilidad	QA	6	S5-02
SPRINT 6 - Autenticación Avanzada						
6	S6-01	OAuth Google/Microsoft	Integración con proveedores externos	Backend	10	S1-04
6	S6-02	Two-Factor Authentication (2FA)	Autenticación de dos factores SMS/email	Backend	8	S6-01
6	S6-03	Single Sign-On (SSO) institucional	Integración con sistemas institucionales	Backend	10	S6-01
6	S6-04	Gestión sesiones avanzada	Control de sesiones en múltiples dispositivos	Backend	6	S6-02
6	S6-05	Interfaz configuración cuenta	Panel de seguridad y privacidad para usuarios	Frontend	8	S6-02
6	S6-06	Logs de seguridad y auditoría	Registro detallado de eventos de autenticación	Backend	6	S6-04
6	S6-07	Testing seguridad autenticación	Pruebas de penetración y seguridad	QA	8	S6-03
SPRINT 7 - Observabilidad y Reportes Avanzados						
7	S7-01	Dashboard admin completo	Métricas en tiempo real y KPIs	Frontend	10	S4-06
7	S7-02	Sistema de logging avanzado	Logging estructurado y centralizado	Backend	8	S6-06

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
7	S7-03	Métricas y analytics	Uso, progreso, engagement y abandono	Backend	8	S7-02
7	S7-04	Reportes exportables	Generación PDF, Excel, CSV automática	Backend	8	S7-01
7	S7-05	Alertas automáticas admin	Notificaciones de errores y problemas	Backend	6	S7-03
7	S7-06	Monitoreo de rendimiento básico	APM básico y métricas de performance	Fullstack	6	S7-05
7	S7-07	Testing observabilidad	Verificación de métricas y alertas	QA	6	S7-04
SPRINT 8 - IA de Resúmenes e Historial Enriquecido						
8	S8-01	Integración API de IA	Conexión con OpenAI/Claude para procesamiento	Backend	10	S1-10
8	S8-02	Generación resúmenes automáticos	IA genera resúmenes de contenido de clases	Backend	8	S8-01
8	S8-03	Interfaz resúmenes estudiante	Visualización de resúmenes en interfaz	Frontend	6	S8-02
8	S8-04	Historial de aprendizaje IA	Patrones de aprendizaje y recomendaciones personalizadas	Backend	8	S8-02
8	S8-05	Cache y optimización llamadas IA	Reducción de costos y latencia de IA	Backend	6	S8-04

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
8	S8-06	Fallback manual si IA falla	Sistema de respaldo cuando IA no está disponible	Fullstack	4	S8-05
8	S8-07	Testing integración IA	Pruebas de calidad y disponibilidad de IA	QA	8	S8-03
FASE 3: CALIDAD Y PRODUCCIÓN						
SPRINT 9 - Rendimiento y Escalabilidad Intermedia						
9	S9-01	Optimización base de datos	Índices, consultas optimizadas y particionado	Backend	8	S7-03
9	S9-02	CDN para archivos multimedia	Distribución global de contenido estático	Fullstack	6	S5-04
9	S9-03	Caching avanzado	Implementación Redis/Memcached para performance	Backend	8	S9-01
9	S9-04	Load testing	Pruebas con 500+ usuarios concurrentes	QA	8	S9-03
9	S9-05	Optimización frontend	Lazy loading, code splitting y chunks optimizados	Frontend	8	S9-02
9	S9-06	Monitoreo de performance	Métricas de rendimiento en tiempo real	Fullstack	6	S9-04
SPRINT 10 - Seguridad, Backups y Privacidad						

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
10	S10-01	Auditoría de seguridad completa	Revisión exhaustiva de vulnerabilidades	Backend	10	S6-07
10	S10-02	Sistema de backups automatizado	Backups incrementales y restauración automática	Fullstack	8	S10-01
10	S10-03	Política de privacidad y GDPR básico	Cumplimiento regulaciones de privacidad	PO/UX	6	S10-01
10	S10-04	Encriptación datos sensibles	Cifrado end-to-end de información crítica	Backend	8	S10-02
10	S10-05	Recuperación ante desastres	Plan de continuidad y recuperación (DR)	Fullstack	6	S10-02
10	S10-06	Testing seguridad y backups	Verificación de procedimientos de seguridad	QA	8	S10-04
SPRINT 11 - Calidad, CI/CD y UAT						
11	S11-01	CI/CD completo	Automatización completa de deployment	Fullstack	10	S10-05
11	S11-02	Testing automatizado E2E completo	Suite completa de pruebas end-to-end	QA	10	S11-01
11	S11-03	User Acceptance Testing (UAT)	Pruebas de aceptación con usuarios reales	PO/UX	8	S11-02
11	S11-04	Corrección bugs críticos UAT	Resolución de issues encontrados en UAT	Todo el equipo	8	S11-03

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Sprint	ID	Tarea	Descripción	Responsable	Hrs	Depende
11	S11-05	Documentación técnica completa	Documentación de arquitectura y APIs	Fullstack	6	S11-04
11	S11-06	Guías de usuario finales	Manuales de usuario y administrador	PO/UX	4	S11-05
SPRINT 12 - Hardening Final y Release						
12	S12-01	Hardening final seguridad	Penetration testing y fortificación final	Backend	8	S11-04
12	S12-02	Optimización final performance	Ajustes finales de rendimiento y memoria	Todo el equipo	8	S12-01
12	S12-03	Preparación entorno producción	Configuración y despliegue en producción	Fullstack	6	S12-02
12	S12-04	Testing final integrado	Batería final de pruebas integrales	QA	6	S12-03
12	S12-05	Release candidato y deploy	Release final y puesta en producción	Fullstack	4	S12-04
12	S12-06	Documentación de entrega y handover	Documentos de entrega y transferencia	PO/UX	4	S12-05

Cuadro 5: Work Breakdown Structure Detallado

4.4 Cronograma y Estimaciones

Distribución de Horas por Sprint

Sprint	Enfoque Principal	Horas Estimadas	Hrs/Persona
1-4	Fundación y Core	280 horas	14h/semana
5-8	Funcionalidades Avanzadas	260 horas	13h/semana
9-12	Calidad y Producción	210 horas	10.5h/semana
Total	**Proyecto Completo**	**750 horas**	**12.5h/semana/persona**

Cuadro 6: Distribución de Carga de Trabajo

4.5 Ceremonias Scrum

Para mantener la disciplina ágil durante los 12 sprints, se implementan ceremonias Scrum con enfoques específicos según la fase del proyecto:

Ceremonia	Frecuencia	Duración	Enfoque por Fase
Planning	Inicio sprint	2-3 horas	Fase 1: Arquitectura; Fase 2: Funcionalidades; Fase 3: Calidad
Daily	Lunes a viernes	15 min	Bloqueos técnicos, dependencias críticas, progreso WBS
Review	Final sprint	1-1.5 horas	Demo funcional con criterios aceptación del WBS
Retrospectiva	Final sprint	45 min	Mejoras proceso, optimización estimaciones WBS
Review de Fase	Sprints 4, 8, 12	2-3 horas	Validación completa con stakeholders y usuarios

Cuadro 7: Ceremonias Scrum por Fase

4.6 Ruta Crítica del Proyecto

La ruta crítica identifica las dependencias que no pueden sufrir retrasos sin afectar la entrega final, basándose en el análisis detallado del WBS.

Dependencias Críticas Principales

Dependencia	Descripción	Impacto Crítico
S1-03 → S2-01	Modelos básicos → Modelo quiz	Bloquea sistema evaluación
S2-04 → S3-01	Calificación automática → Reglas aprobación	Bloquea certificados
S3-02 → S8-01	Certificados → IA	Bloquea funcionalidades avanzadas
S7-03 → S9-01	Analytics → Optimización BD	Bloquea escalabilidad
S11-02 → S12-01	Testing E2E → Hardening final	Bloquea release

Cuadro 8: Dependencias Críticas del WBS

Ruta Crítica Principal

Ruta Crítica - WBS de 12 Sprints

FASE 1: S1-01 → S1-03 → S2-01 → S2-04 → S3-02

↓

FASE 2: S6-01 → S7-03 → S8-01 → S8-02

↓

FASE 3: S9-01 → S11-02 → S12-01 → S12-05

Duración: 12 semanas — Margen seguridad: 3-5 días por fase

4.7 Criterios de Éxito y Control de Calidad

Por Fase del Proyecto

Fase 1 (Sprints 1-4): Fundación y Core

- Sistema de autenticación seguro operativo (S1-04, S6-01)
- CRUD completo de cursos con roles (S1-05, S4-01)
- Sistema de evaluación y certificados básico (S2-04, S3-02)
- Panel administrativo funcional (S3-05, S4-06)

Fase 2 (Sprints 5-8): Funcionalidades Avanzadas

- Experiencia móvil optimizada con accesibilidad (S5-01, S5-02)
- Autenticación OAuth y 2FA operativos (S6-01, S6-02)
- Sistema de observabilidad completo (S7-01, S7-03)
- Integración IA con resúmenes automáticos (S8-01, S8-02)

Fase 3 (Sprints 9-12): Calidad y Producción

- Performance validado con 500+ usuarios (S9-04)
- Seguridad auditada y backups automatizados (S10-01, S10-02)
- CI/CD completo con UAT satisfactorio (S11-01, S11-03)
- Release candidato listo para producción (S12-05)

Esta planificación detallada basada en el WBS de 101 tareas garantiza un desarrollo estructurado, trazable y de alta calidad, manteniendo la flexibilidad ágil mientras se asegura el cumplimiento de todos los requerimientos técnicos y de negocio establecidos.

5 Requerimientos

Para asegurar que la plataforma responda a las necesidades de estudiantes, docentes y administradores, se realizó un proceso de levantamiento y validación de requerimientos siguiendo las mejores prácticas del IEEE (IEEE Computer Society, 1998). Este se basó en entrevistas con usuarios clave, la elaboración de historias de usuario y la observación de plataformas similares. Como resultado, se identificaron tanto requerimientos funcionales (acciones que el sistema debe realizar) como requerimientos no funcionales (atributos de calidad y restricciones técnicas), los cuales servirán de base para el diseño, desarrollo y validación del sistema.

5.1 Historias de Usuario (HU)

Con el fin de reflejar las necesidades reales de estudiantes, docentes y administradores, los requerimientos se documentaron inicialmente en forma de historias de usuario, siguiendo la metodología propuesta en (Cohn, 2004). Estas historias permiten expresar, en un lenguaje sencillo y centrado en el usuario, las funcionalidades que el sistema debe ofrecer y sirven como punto de partida para detallar los requerimientos funcionales y no funcionales en las tablas posteriores.

Historia Usuario	Descripción
HU1	Como estudiante, quiero registrarme e iniciar sesión con mi correo, para acceder a mis cursos.
HU2	Como docente, quiero crear cursos con materiales, para que mis estudiantes puedan inscribirse y estudiar.
HU3	Como estudiante, quiero inscribirme a cursos disponibles, para organizar mi aprendizaje.
HU4	Como estudiante, quiero acceder a videos y documentos en PDF, para estudiar el contenido en cualquier momento.
HU5	Como docente, quiero subir evaluaciones y notas, para retroalimentar a los estudiantes. / Como estudiante, quiero revisar mis notas.
HU6	Como estudiante, quiero recibir un certificado al aprobar un curso, para validar mis logros académicos.
HU7	Como estudiante, quiero recibir notificaciones de nuevas evaluaciones, cursos o certificados, para mantenerme informado.
HU8	Como estudiante, quiero usar la plataforma desde mi celular, para acceder en cualquier lugar.
HU9	Como administrador, quiero administrar cuentas y contenidos de la plataforma.

Cuadro 9: Historias de Usuario

Estas historias de usuario proporcionan una visión clara de las expectativas y necesidades

de los distintos actores del sistema. A partir de ellas, se definieron los requerimientos funcionales y no funcionales, que permiten transformar estas necesidades en especificaciones técnicas medibles y verificables, asegurando su correcta implementación y validación en el proyecto.

5.2 Requerimientos Funcionales

ID	Descripción	Técnica	Criterio de Aceptación
RF1	Registro e inicio de sesión con correo/clave y OAuth (Google/Microsoft) para estudiantes/docentes.	Entrevista + Historia de usuario	El usuario puede crear cuenta o iniciar sesión en 10s y acceder a su panel personal.
RF2	Creación, edición, publicación y eliminación de cursos por docentes.	Entrevista	Un docente puede gestionar un curso y este queda visible/invisible en catálogo.
RF3	Subida, edición y borrado de materiales por curso (videos, PDFs, enlaces externos).	Entrevista	El material queda disponible en línea con validación de formato y tamaño.
RF4	Inscripción del estudiante a cursos disponibles, con validación de cupo y prerrequisitos.	Historia de usuario	El estudiante se inscribe y aparece en el listado de inscritos.
RF5	Visualización del progreso por curso (lecciones completadas, evaluaciones rendidas, notas).	Historia de usuario	El sistema muestra % de avance y nota acumulada en tiempo real.
RF6	Creación y gestión de evaluaciones por docentes; registro de notas y retroalimentación.	Entrevista + Observación	Estudiantes reciben calificaciones visibles en su perfil.
RF7	Emisión automática de certificados al aprobar un curso (60 % nota final + 100 % módulos vistos).	Entrevista + Observación	El estudiante recibe un PDF con QR verificable en 10s tras aprobar.

Continúa en la siguiente página...

Tabla – continuación de la página anterior

ID	Descripción	Técnica	Criterio de Aceptación
RF8	Envío de notificaciones a estudiantes (nuevos cursos, evaluaciones, certificados).	Observación	Notificación en panel y/o correo llega en 1 min tras el evento.
RF9	Administración de cuentas, roles, contenidos y certificaciones por parte de administrador.	Observación	El administrador puede crear, editar, desactivar usuarios y cursos desde un panel.
RF10	Generación de historial académico del estudiante (cursos inscritos, progreso, certificados).	Historia de usuario	El estudiante visualiza lista cronológica de sus cursos, calificaciones y logros.
RF11	Funcionalidad de IA para generar resúmenes automáticos de clases.	Observación + Entrevista	El estudiante accede a un resumen textual generado desde el material de cada clase.
RF12	Auditoría básica: registro de accesos, inscripciones y emisión de certificados.	Observación	El sistema almacena logs consultables por administradores con fecha/hora/usuario.
RF13	Verificación y revocación de certificados.	Observación	Cada certificado tiene URL pública; admin puede revocarlo y se refleja en 1 min.
RF14	Gestión de intentos y ponderaciones de evaluación.	Entrevista	Se configuran intentos máximos y ponderaciones; nota final queda trazada en historial.
RF15	Cupos, lista de espera y prerrequisitos de cursos.	Observación	Si el cupo está lleno, el alumno queda en lista de espera; se valida prerrequisito.
RF16	Gestión segura de archivos.	Observación	Se validan formatos, tamaño máximo (ej. 200MB video), y escaneo antivirus.

Continúa en la siguiente página...

Tabla – continuación de la página anterior

ID	Descripción	Técnica	Criterio de Aceptación
RF17	El catálogo permite buscar por título/etiquetas y muestra resultados paginados (ej. 20 ítems/página).	Observación	Dashboards con métricas p95, tasa de error ¡1 % y alertas automáticas.

Cuadro 10: Requerimientos Funcionales

5.3 Requerimiento No Funcionales

ID	Descripción	Técnica	Criterio de Aceptación
RNF1	Seguridad: cifrado de datos (HTTPS/TLS), autenticación con roles y hash seguro.	Entrevista + Observación	100 % del tráfico en HTTPS; contraseñas almacenadas con bcrypt/Argon2.
RNF2	Disponibilidad: sistema operativo 99.5 % mensual, con alertas en caídas.	Observación	Tiempo de inactividad 3,5h al mes.
RNF3	Rendimiento: tiempo de respuesta 3 s en el 95 % de las solicitudes.	Historia de usuario	Pruebas de carga demuestran cumplimiento en p95.
RNF4	Usabilidad: interfaz responsiva y usable en móviles.	Historia de usuario	Puntaje 90 en auditoría Lighthouse (mobile).
RNF5	Portabilidad: compatibilidad con navegadores modernos y móviles iOS/Android.	Observación	Funciona correctamente en Chrome, Firefox, Edge, Safari y dispositivos móviles.
RNF6	Mantenibilidad: cobertura de pruebas 70 % en capa de negocio.	Observación	Reportes de CI muestran cobertura mínima alcanzada.
RNF7	Accesibilidad: cumplimiento de estándares WCAG 2.1 nivel AA.	Entrevista + Observación	Todas las pantallas pasan validación de accesibilidad.

Continúa en la siguiente página...

Tabla – continuación de la página anterior

ID	Descripción	Técnica	Criterio de Aceptación
RNF8	Escalabilidad: capacidad de soportar al menos 500 usuarios concurrentes.	Observación + Benchmark	Pruebas de estrés confirman soporte 500 usuarios sin degradar rendimiento.
RNF9	Confiabilidad: recuperación ante fallos mediante backups automáticos diarios.	Observación	Restauración de base de datos verificada en 30 min en pruebas de recuperación.
RNF10	Trazabilidad: requerimientos vinculados a casos de uso y pruebas.	Observación + Documentación	Matriz de trazabilidad actualizada y revisada en cada Sprint Review.
RNF11	Privacidad y retención de datos.	Entrevista + Observación	Política de retención de 12 meses en logs; derecho al borrado en 30 días.
RNF12	Observabilidad y monitoreo.	Observación	Dashboards con métricas p95, tasa de error ¡1 % y alertas automáticas.

Cuadro 11: Requerimientos No Funcionales

El levantamiento y la validación de requerimientos garantizan que la plataforma responda tanto a las necesidades funcionales de los usuarios como a los atributos de calidad exigidos en un entorno académico. La combinación de historias de usuario, requerimientos funcionales y no funcionales con criterios de aceptación claros asegura una base sólida para el diseño UML, la planificación del desarrollo y la posterior evaluación del sistema, evitando ambigüedades y asegurando trazabilidad en todo el ciclo de vida del proyecto.

6 Diagramas UML

La definición detallada de los requerimientos permitió identificar los principales procesos y entidades que debe soportar la plataforma. Para representar gráficamente estas interacciones y la estructura del sistema, se elaboraron distintos diagramas UML siguiendo las metodologías establecidas en la literatura (Fowler, 2003; Larman, 2004), que cubren de manera exhaustiva todos los aspectos funcionales y estructurales del sistema. Estos diagramas traducen los requerimientos en modelos visuales, facilitando la comprensión de los flujos, las relaciones entre los actores y la lógica interna del sistema, y asegurando que el diseño técnico se mantenga alineado con las necesidades levantadas en las etapas anteriores.

6.1 Diagrama de Caso de Uso

Presenta una vista integral del sistema con tres actores.

- El *Usuario* genérico maneja el sistema completo de autenticación incluyendo registro, inicio de sesión, recuperación de contraseña y cierre de sesión seguro.
- El *Estudiante* tiene acceso a un ecosistema educativo completo que incluye búsqueda y navegación del catálogo de cursos, inscripción a cursos con validación de prerrequisitos, acceso estructurado a contenidos organizados por módulos, seguimiento detallado de progreso académico, realización de evaluaciones en línea, y obtención de certificados digitales verificables.
- El *Docente* cuenta con herramientas completas de gestión académica para crear y editar cursos, estructurar contenidos en módulos, subir materiales multimedia diversos, diseñar y configurar evaluaciones, realizar seguimiento detallado de estudiantes inscritos, calificar evaluaciones de manera eficiente, y gestionar el ciclo de vida de los cursos (activación/desactivación).
- El *Administrador del Sistema* tiene control total sobre la plataforma incluyendo gestión completa de usuarios y roles, supervisión y monitoreo de todos los cursos, generación de reportes comprehensivos del sistema, y configuración de parámetros globales de la plataforma.

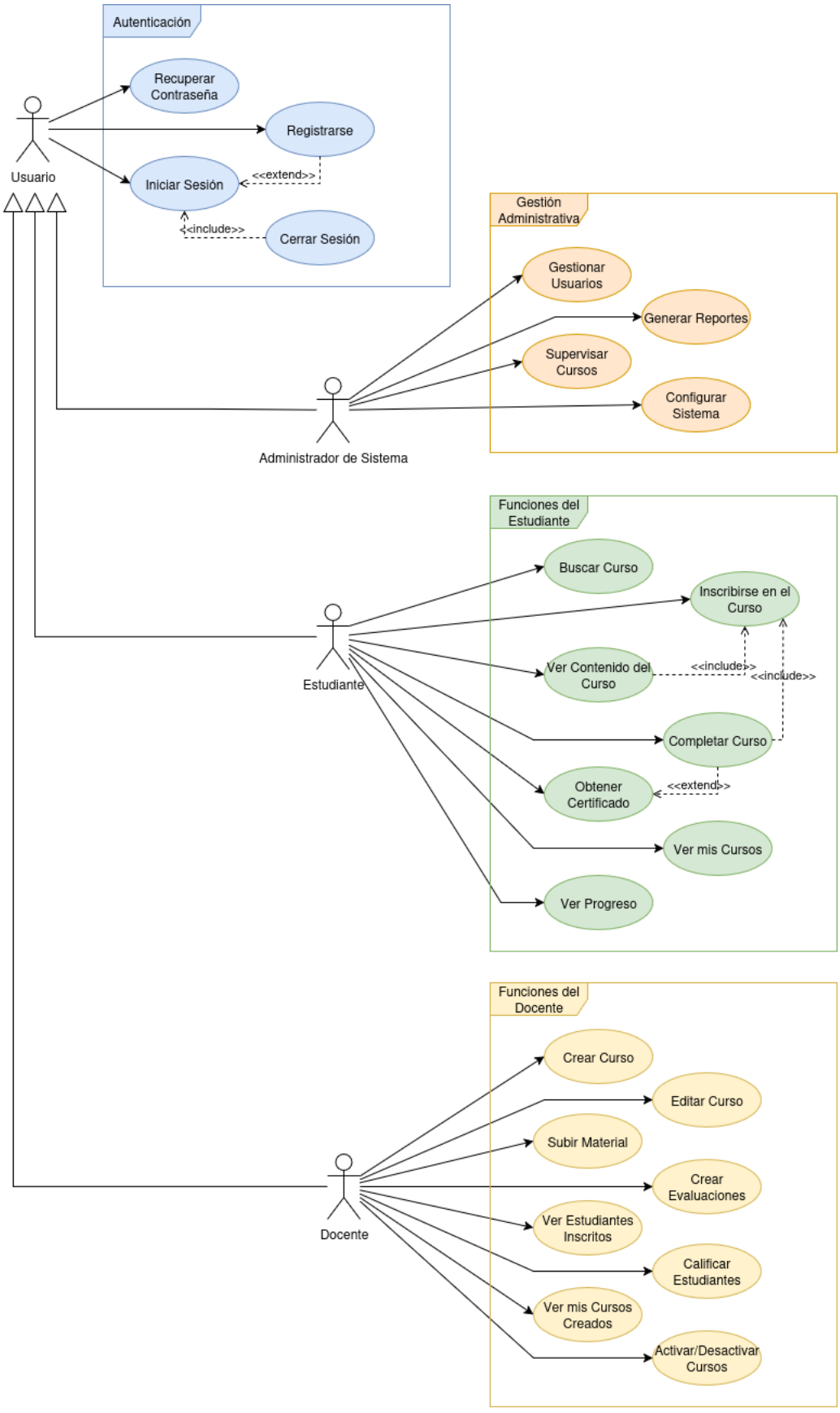


Figura 1: Diagrama de Caso de Uso

6.2 Diagrama de Clases

Modela la arquitectura del sistema mediante un conjunto estructurado de entidades interrelacionadas.

- La clase central *Usuario* implementa herencia con las especializaciones *Estudiante*, *Docente* y *Administrador*, cada una con métodos específicos para sus funcionalidades particulares.
- La entidad *Curso* actúa como contenedor principal que se relaciona con *Módulo* mediante una composición fuerte, donde cada módulo puede contener múltiples *Material*es de diferentes tipos (videos, PDFs, enlaces).
- El sistema de evaluación se modela a través de las clases *Quiz* y *Pregunta* con una relación de composición, donde cada quiz contiene múltiples preguntas con sus respectivas opciones de respuesta.
- La clase *Inscripción* funciona como entidad de asociación entre *Estudiante* y *Curso*, manteniendo el estado del progreso académico, las calificaciones obtenidas, y generando automáticamente instancias de *Certificado* cuando se cumplen los criterios de aprobación.
- El modelo incluye también la clase *Rol* que implementa un sistema de control de acceso basado en roles (RBAC) para gestionar permisos de manera granular.

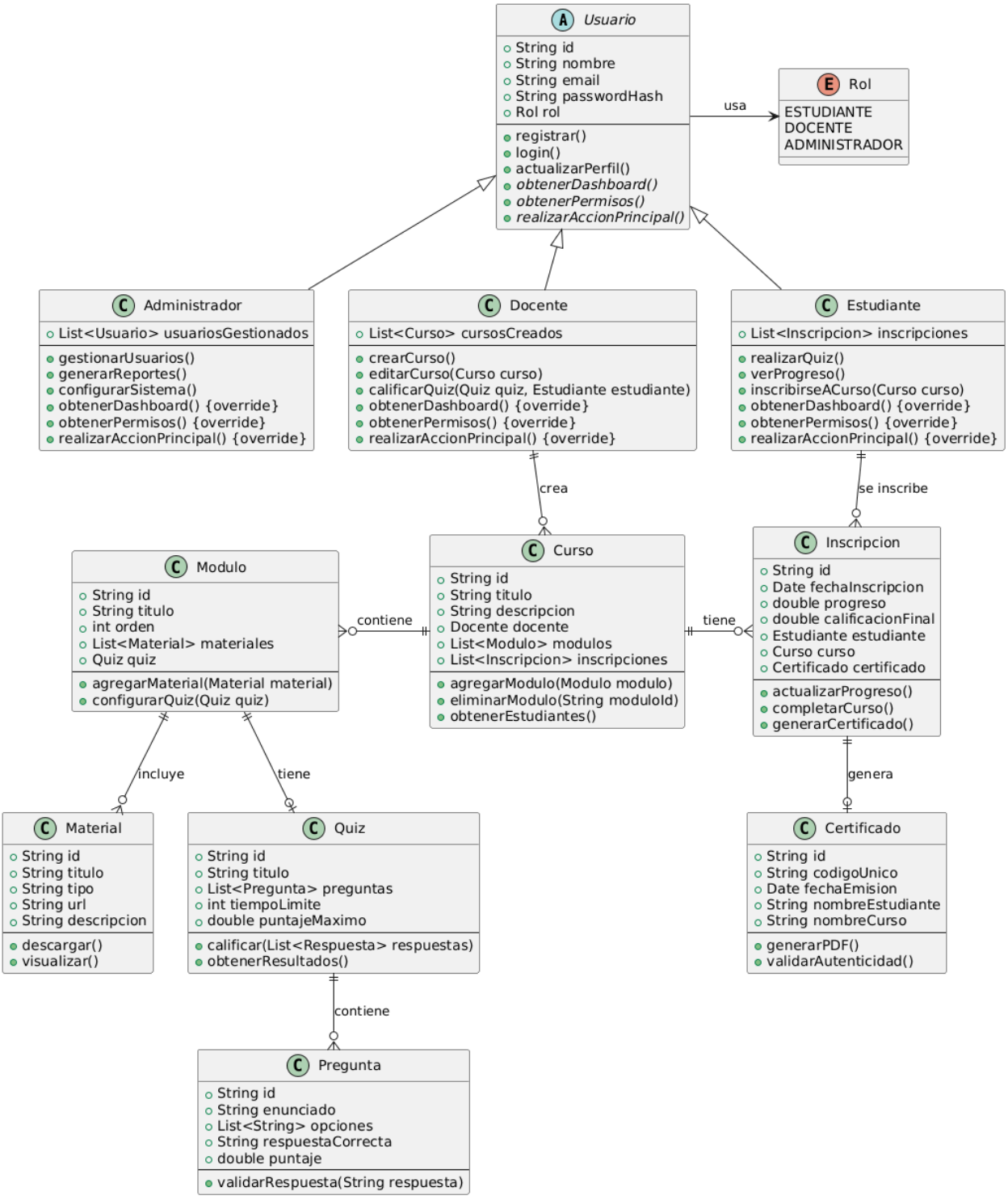


Figura 2: Diagrama de Clases

6.3 Diagramas de Secuencia

Se desarrollaron 22 diagramas de secuencia que cubren de manera exhaustiva todas las interacciones críticas del sistema organizados en cuatro categorías principales.

- Los *procesos de autenticación y gestión de usuarios* (5 diagramas) incluyen el registro de nuevos usuarios con validación de datos y envío de emails de confirmación, el inicio de sesión con múltiples métodos de autenticación y manejo de tokens JWT, la recuperación segura de contraseñas mediante tokens temporales, la confirmación y actualización de credenciales, y el cierre de sesión con invalidación de tokens.
- Los *procesos del estudiante* (4 diagramas) abarcan la búsqueda avanzada en el catálogo de cursos con filtros múltiples, la inscripción a cursos con validación de cupos y prerrequisitos, la realización de evaluaciones con cálculo automático de calificaciones y actualización de progreso, y la verificación de completitud de cursos con generación automática de certificados.
- Los *procesos del docente* (6 diagramas) incluyen la creación de cursos con validación de permisos docentes, la estructuración de contenidos mediante la adición de módulos, la gestión de materiales multimedia con validación de formatos y tamaños, la creación y configuración de evaluaciones con múltiples tipos de preguntas, la visualización detallada del progreso de estudiantes inscritos, y la calificación manual de evaluaciones con retroalimentación.
- Los *procesos del administrador* (7 diagramas) cubren la visualización comprehensiva de usuarios del sistema con estadísticas, la creación de nuevos usuarios con asignación de roles, la edición de información de usuarios existentes, la solicitud de reportes del sistema con métricas en tiempo real, y la generación especializada de reportes de usuarios, cursos y certificados con exportación a múltiples formatos.

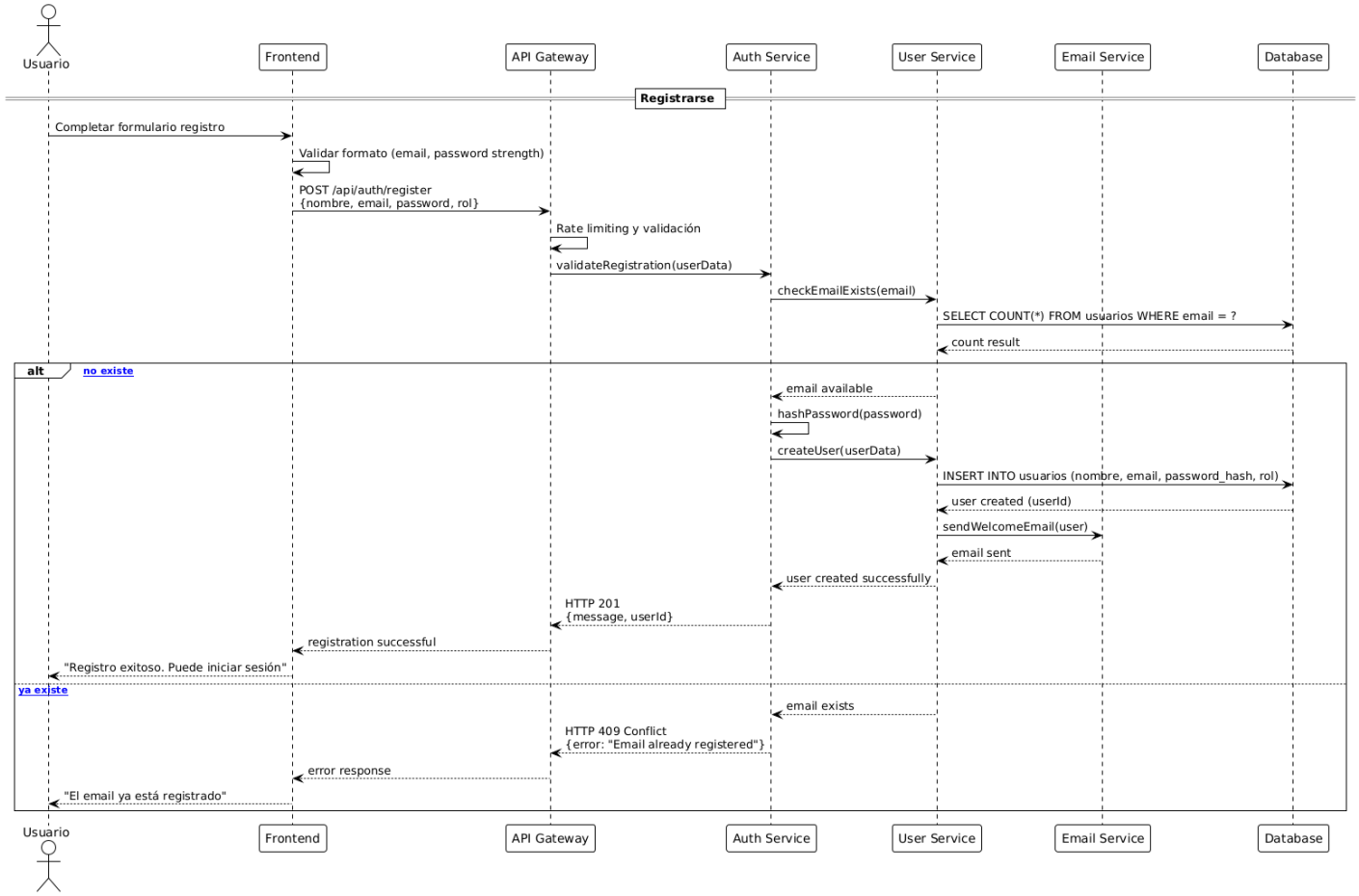


Figura 3: Diagrama de Secuencia - Usuario (Registrarse)

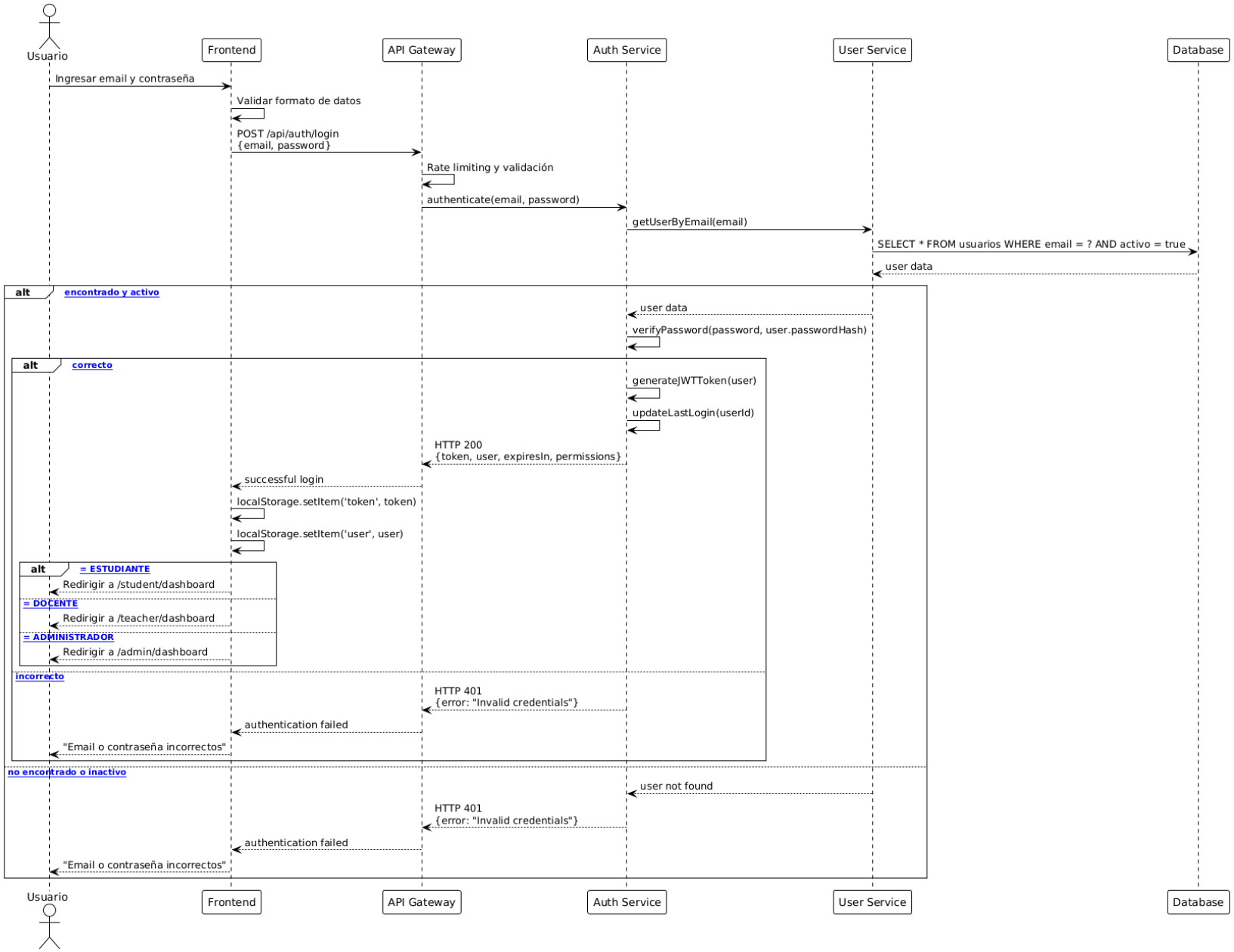


Figura 4: Diagrama de Secuencia - Usuario (Iniciar Sesión)

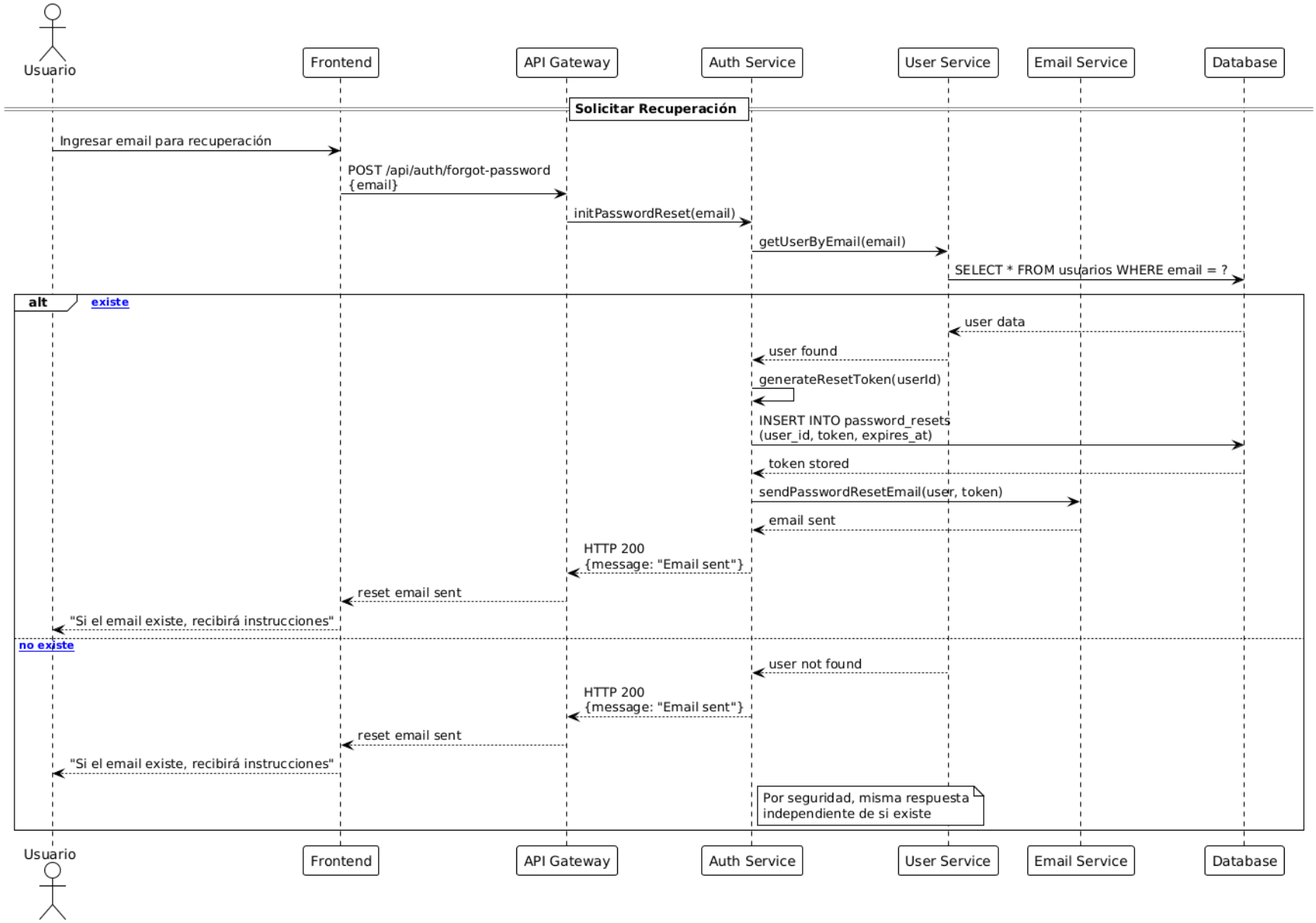


Figura 5: Diagrama de Secuencia - Usuario (Recuperar Contraseña)

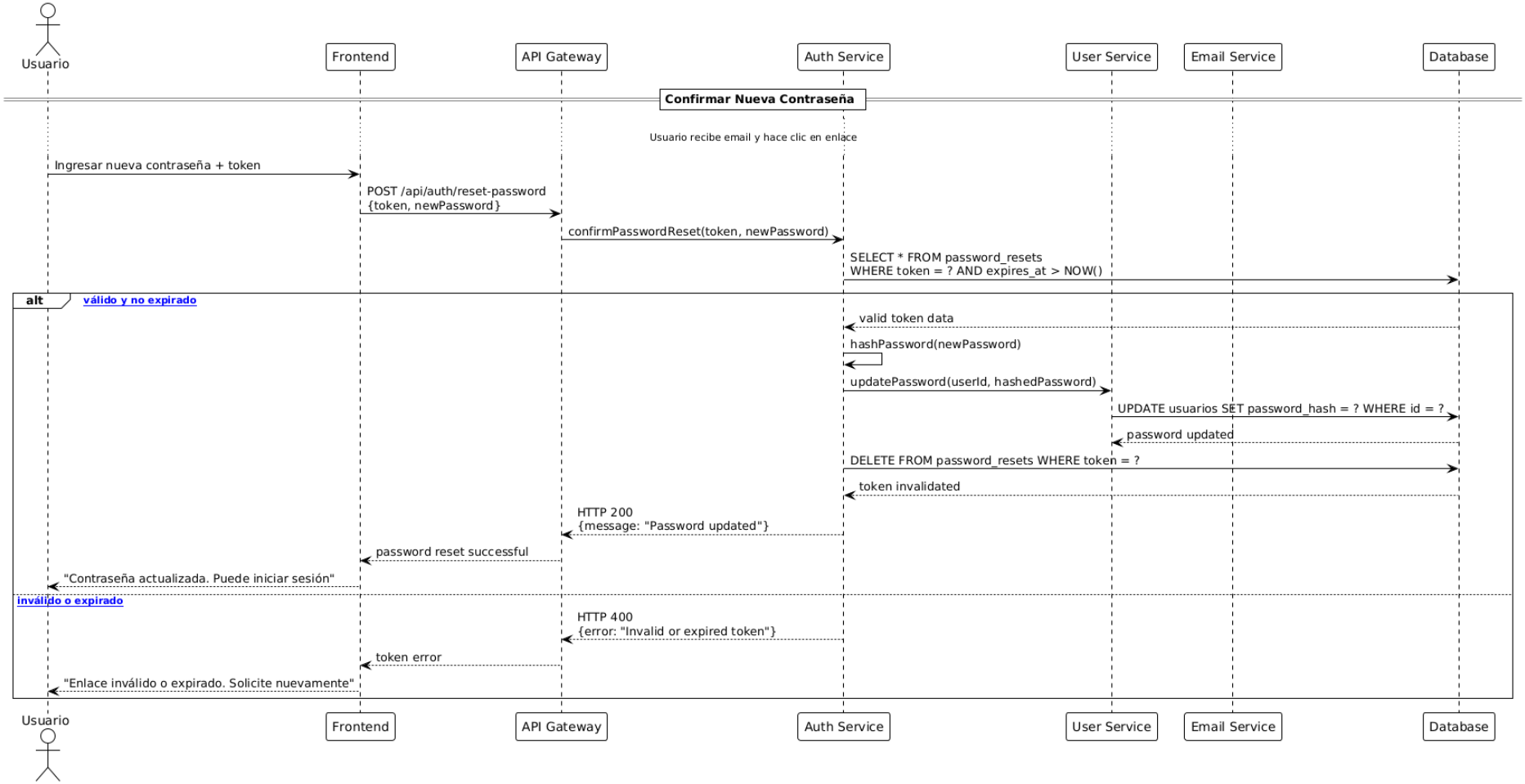


Figura 6: Diagrama de Secuencia - Usuario (Confirmar Nueva Contraseña)

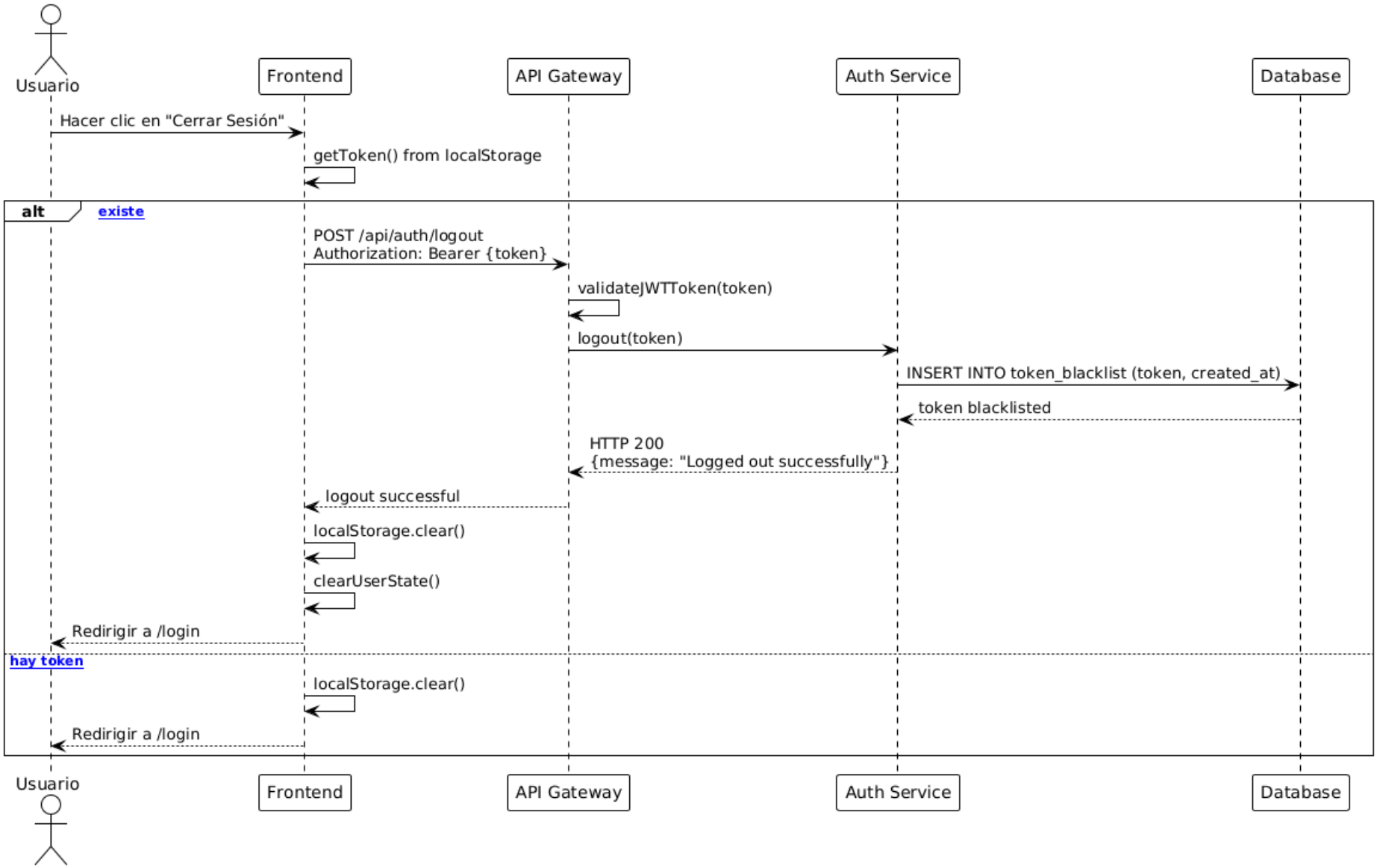


Figura 7: Diagrama de Secuencia - Usuario (Cerrar Sesión)

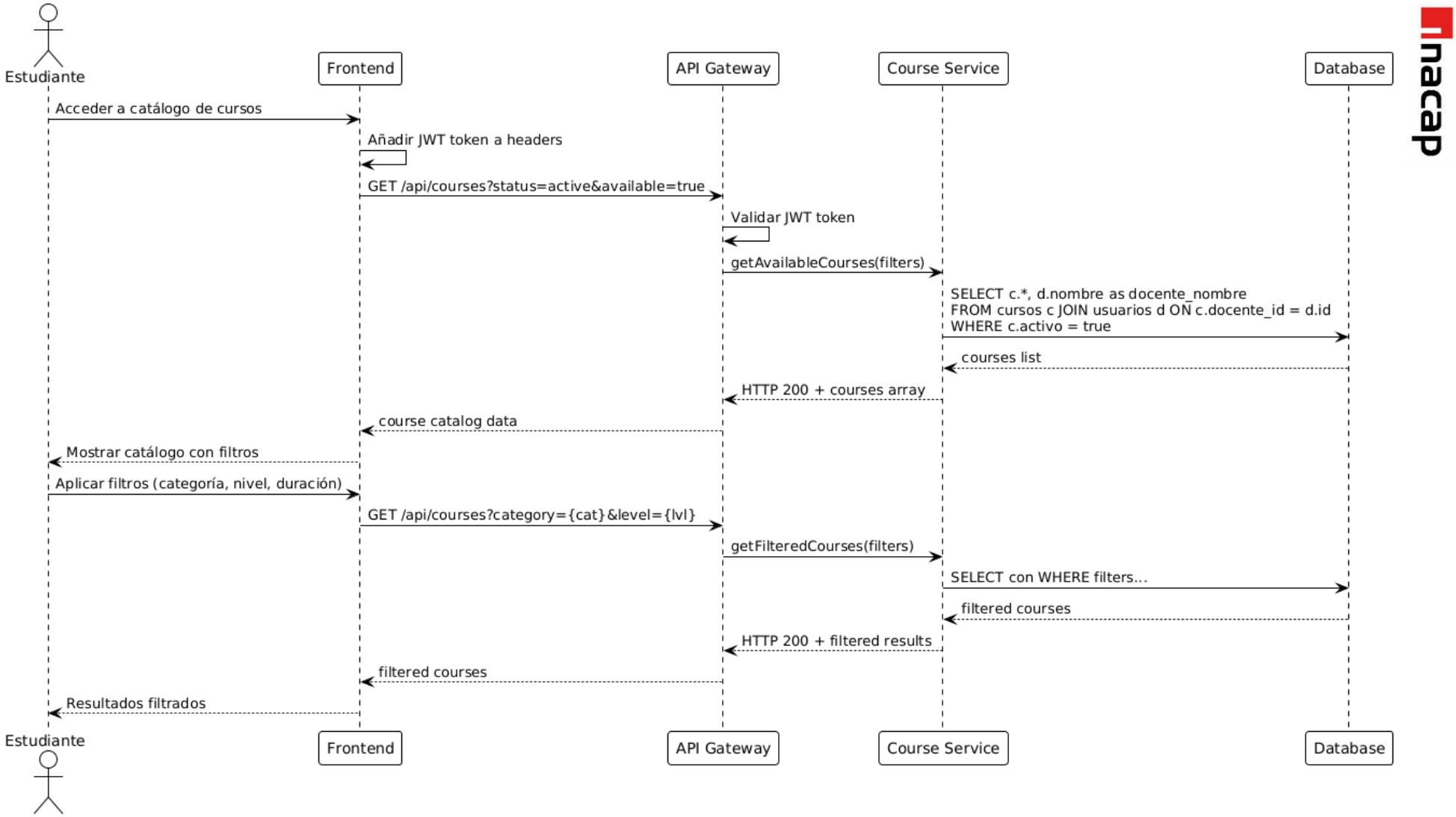


Figura 8: Diagrama de Secuencia - Estudiante (Buscar Curso)

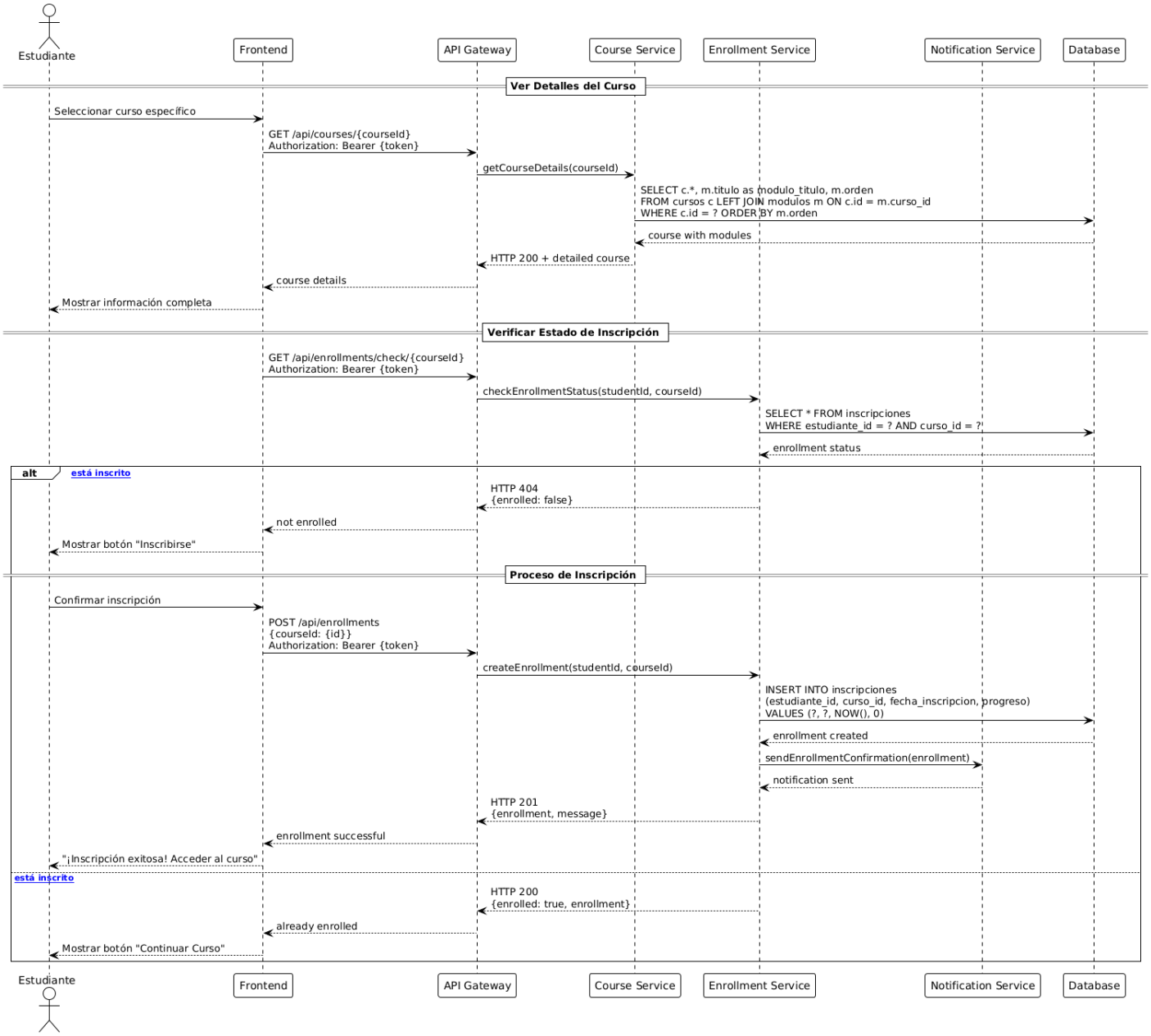


Figura 9: Diagrama de Secuencia - Estudiante (Inscribir Curso)

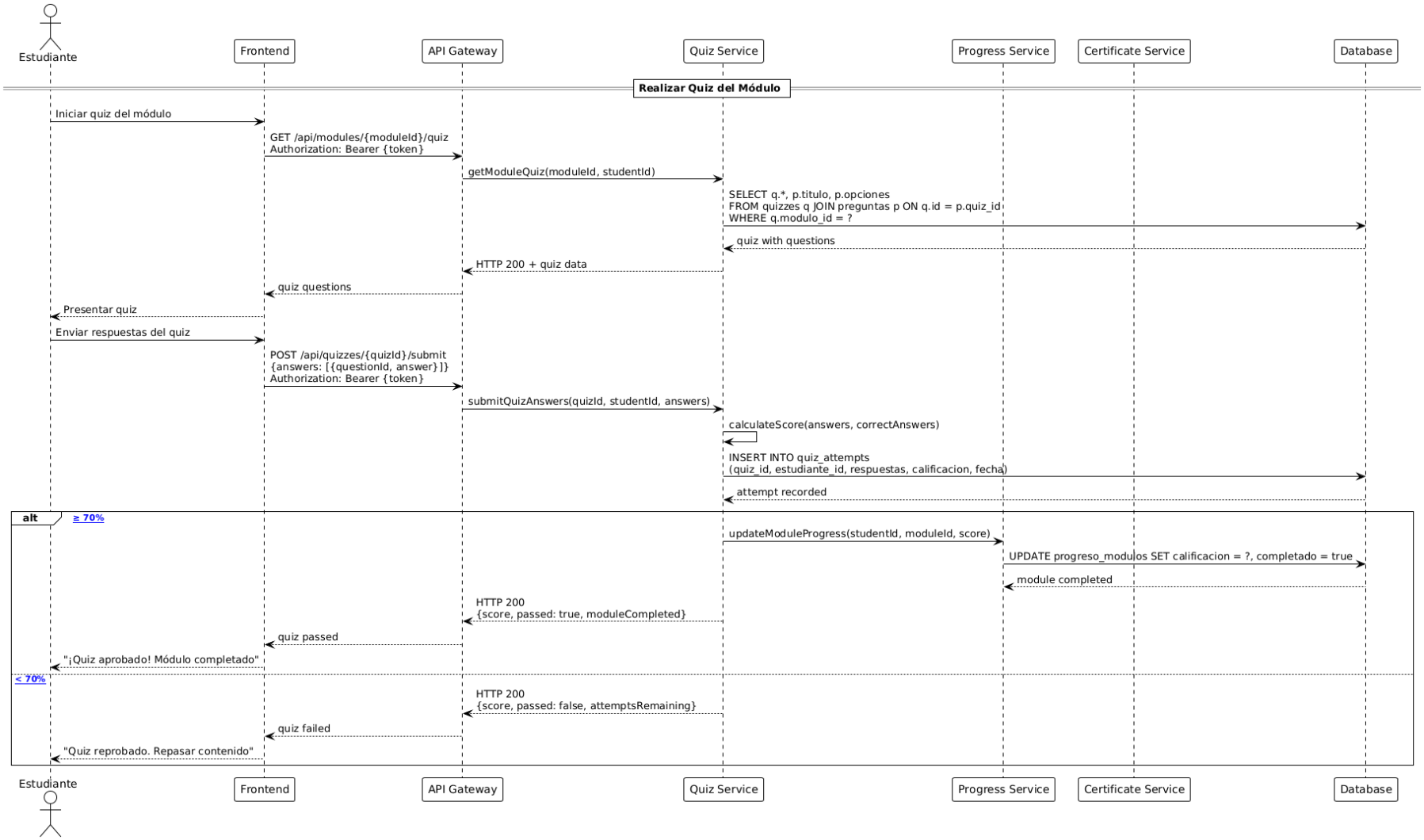


Figura 10: Diagrama de Secuencia - Estudiante (Realizar Quiz)

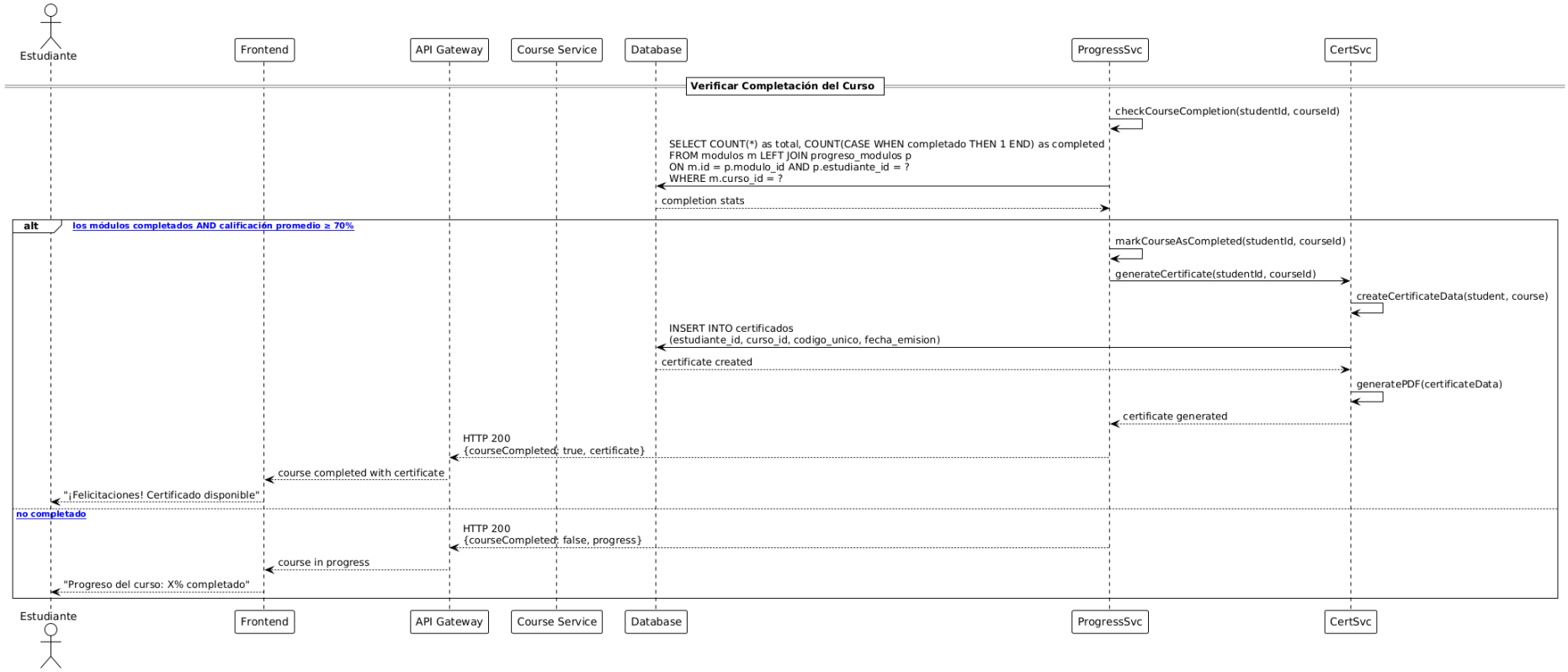


Figura 11: Diagrama de Secuencia - Estudiante (Completar Curso)

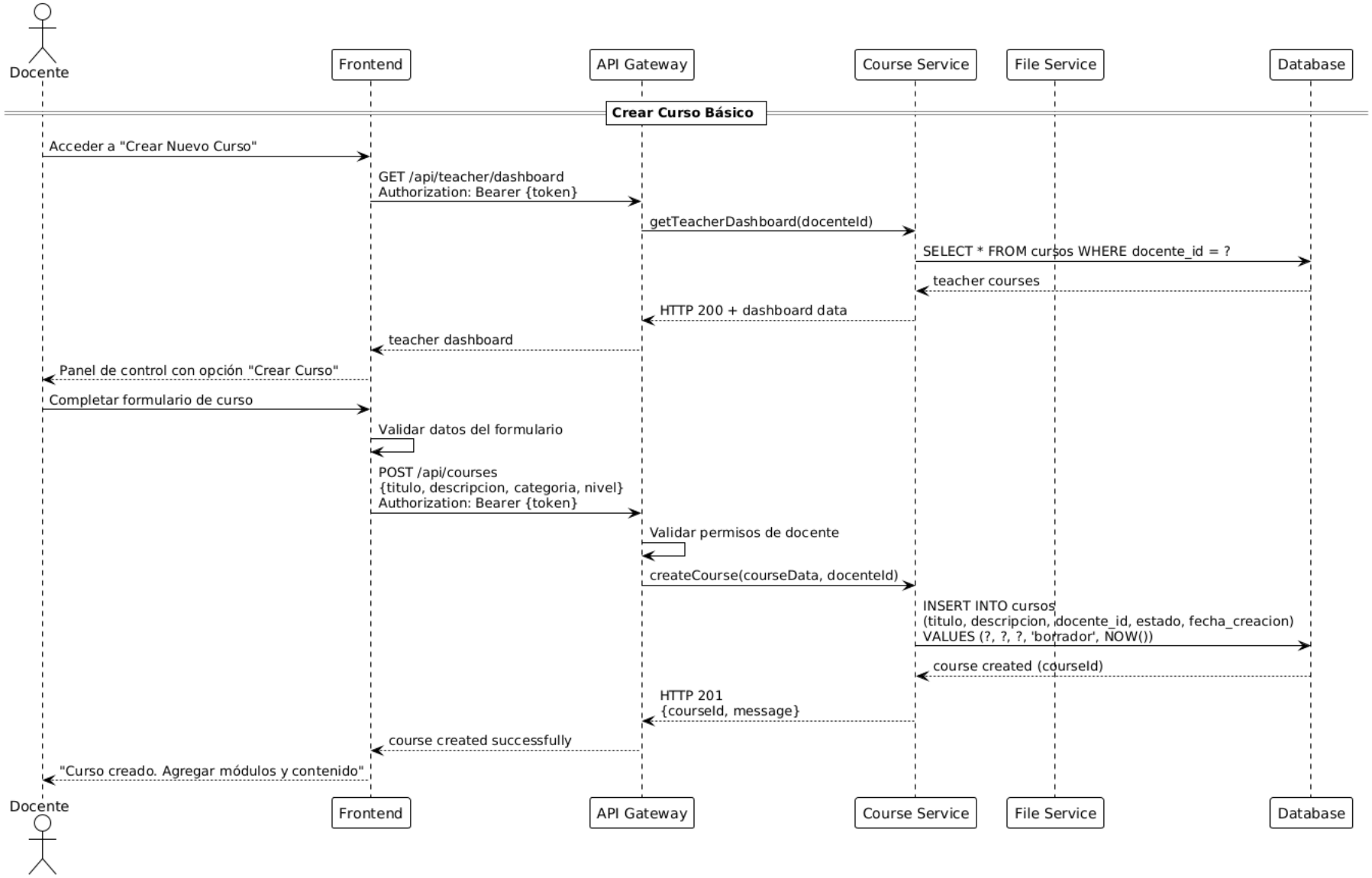


Figura 12: Diagrama de Secuencia - Docente (Crear Curso)

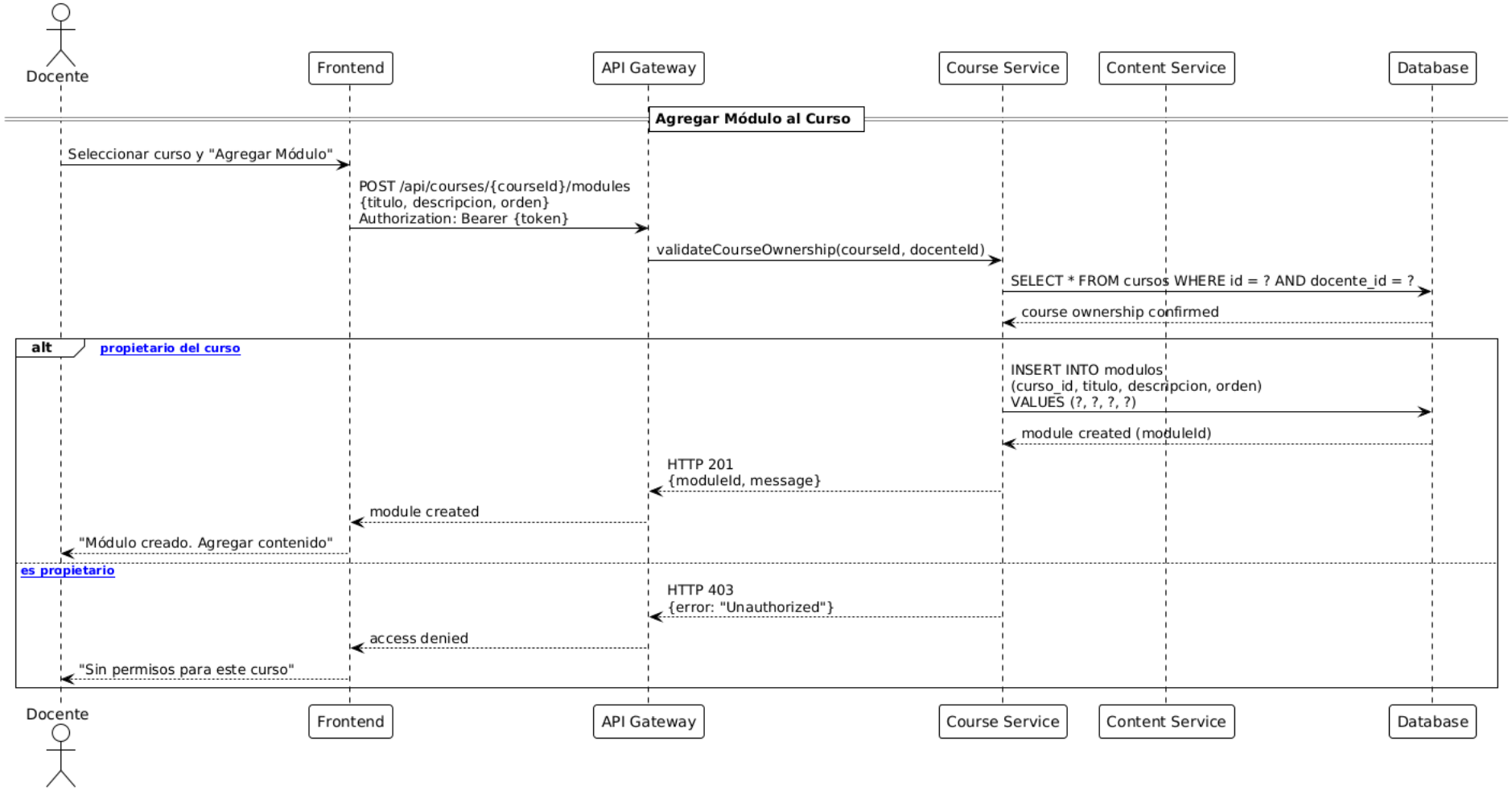


Figura 13: Diagrama de Secuencia - Docente (Agregar Módulo al Curso)

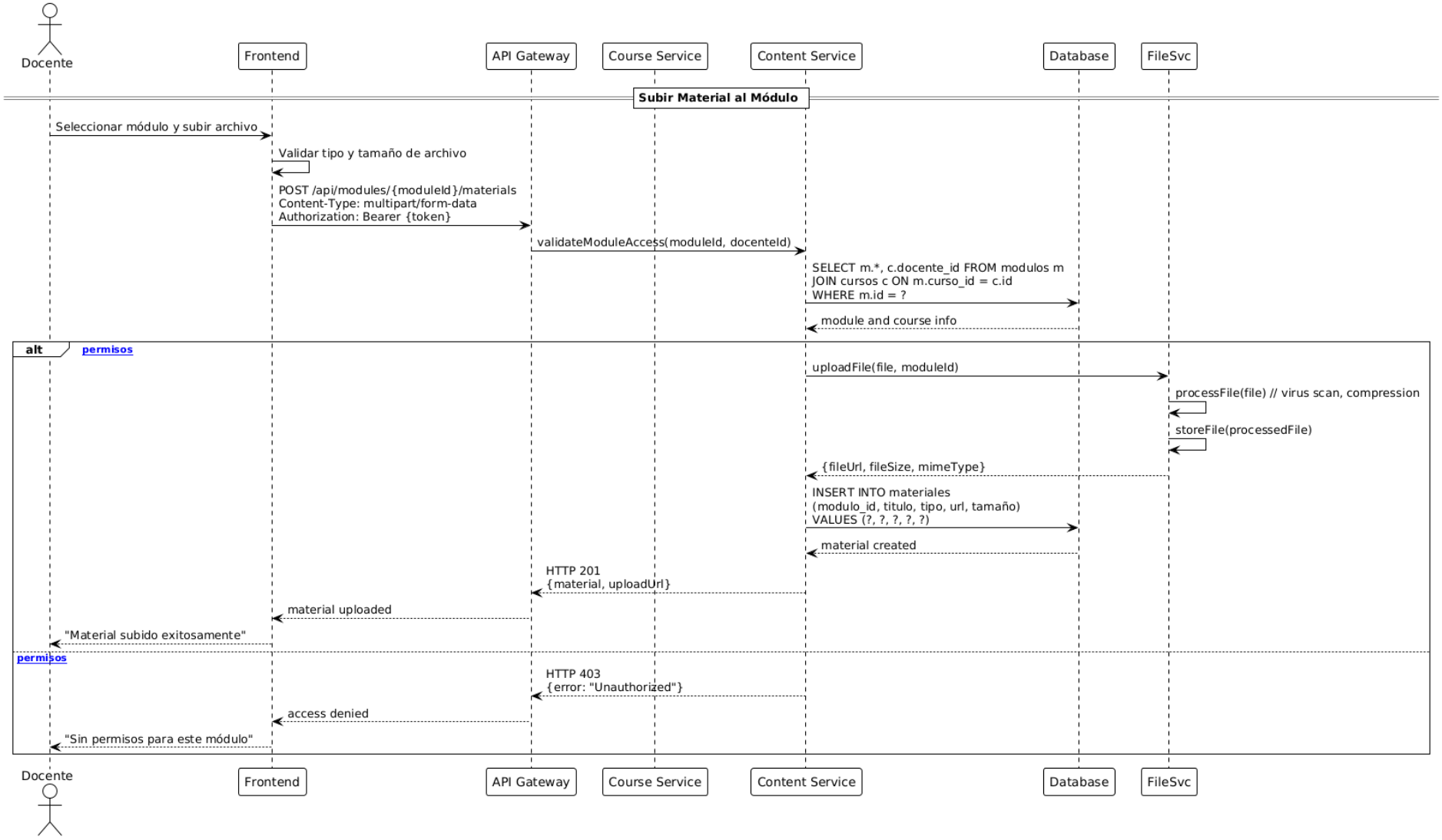


Figura 14: Diagrama de Secuencia - Docente (Subir material al Módulo)

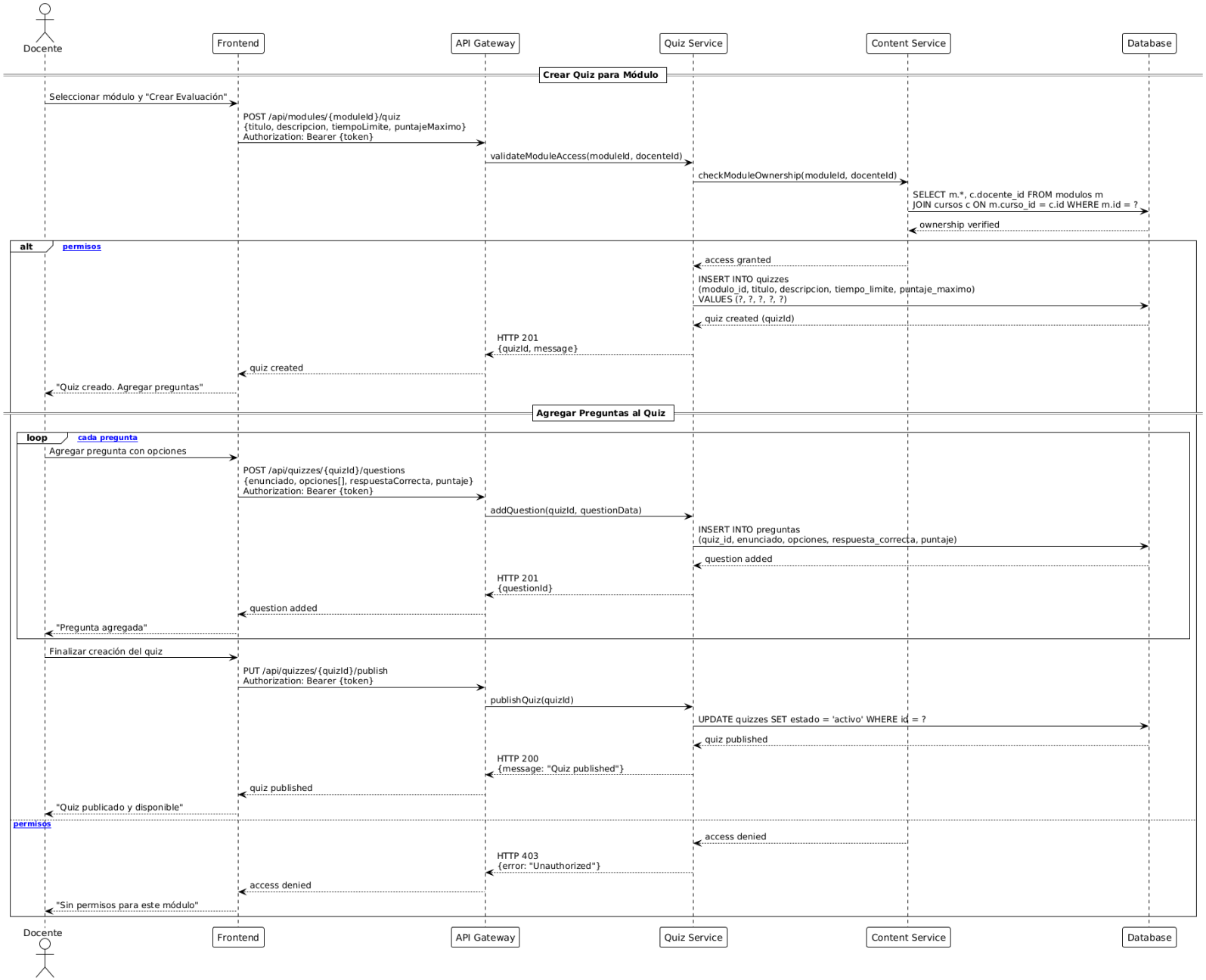


Figura 15: Diagrama de Secuencia - Docente (Crear Evaluaciones)

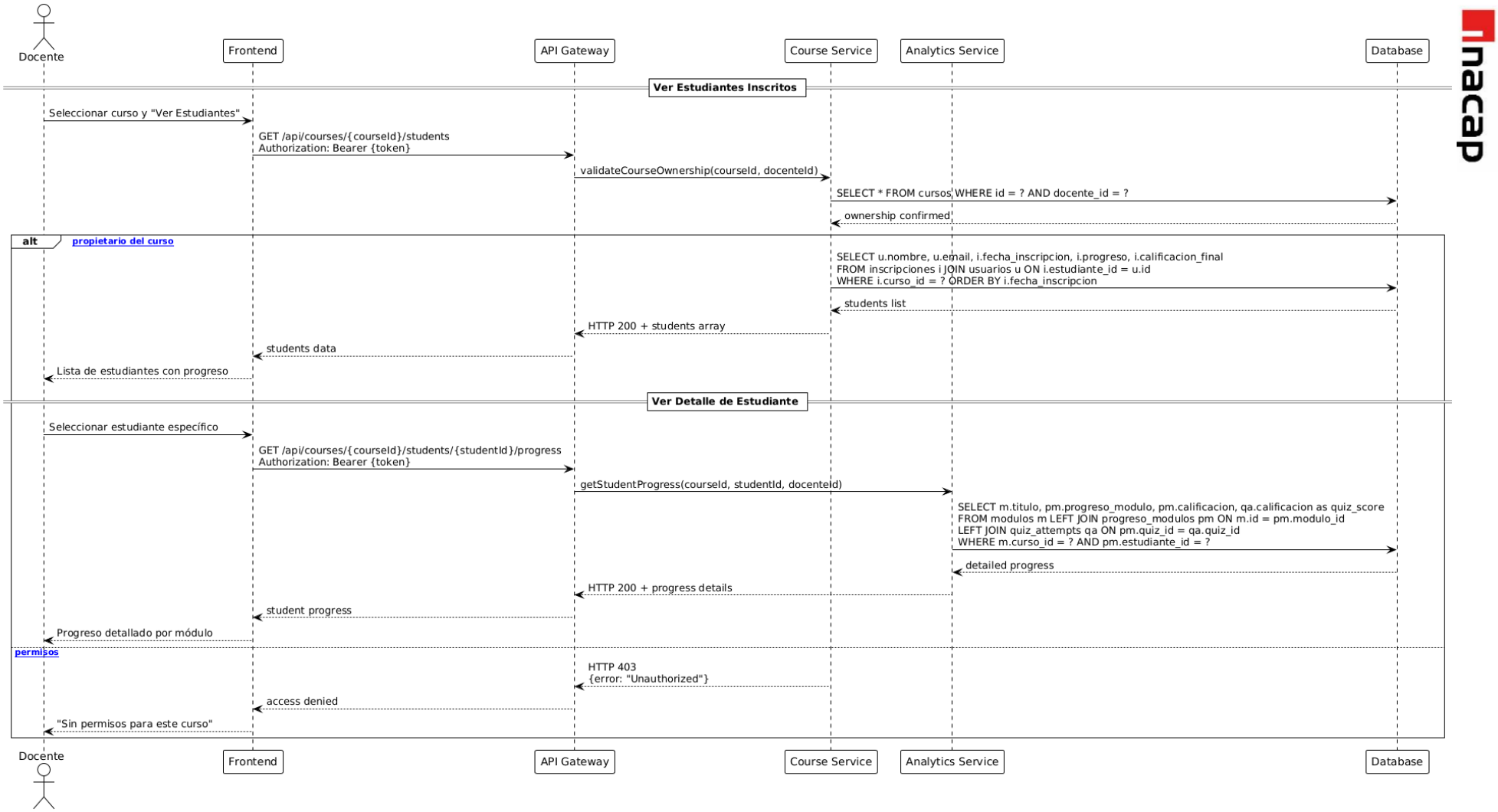


Figura 16: Diagrama de Secuencia - Docente (Ver Estudiantes)

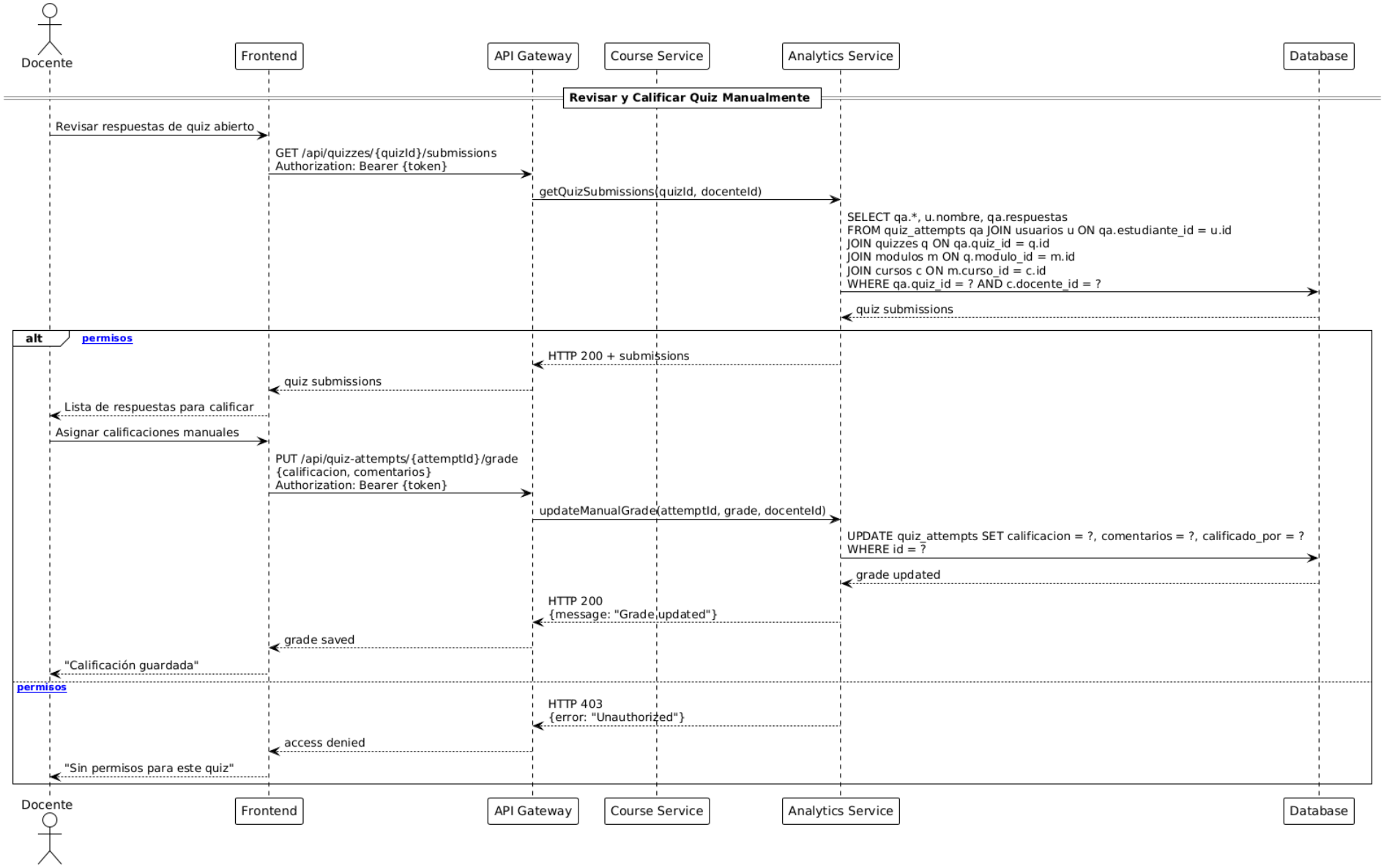


Figura 17: Diagrama de Secuencia - Docente (Revisar y Calificar Evaluaciones)

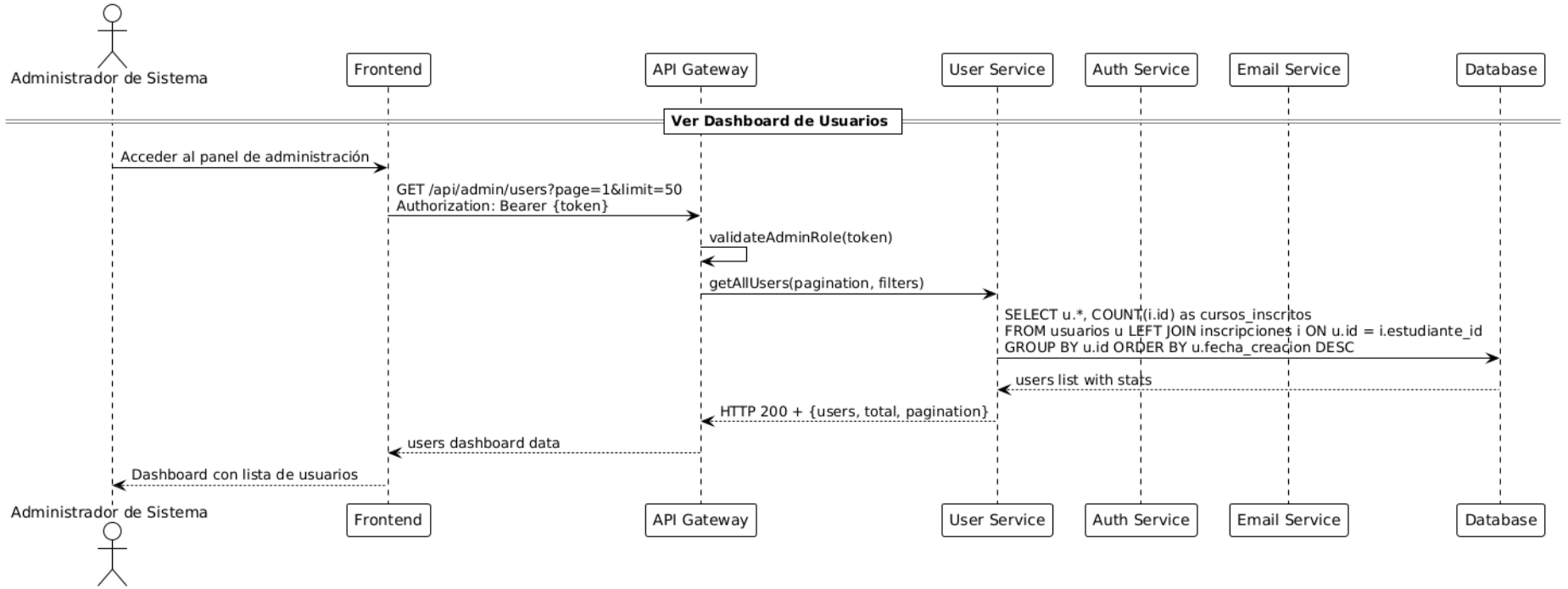


Figura 18: Diagrama de Secuencia - Administrador (Ver Usuarios)

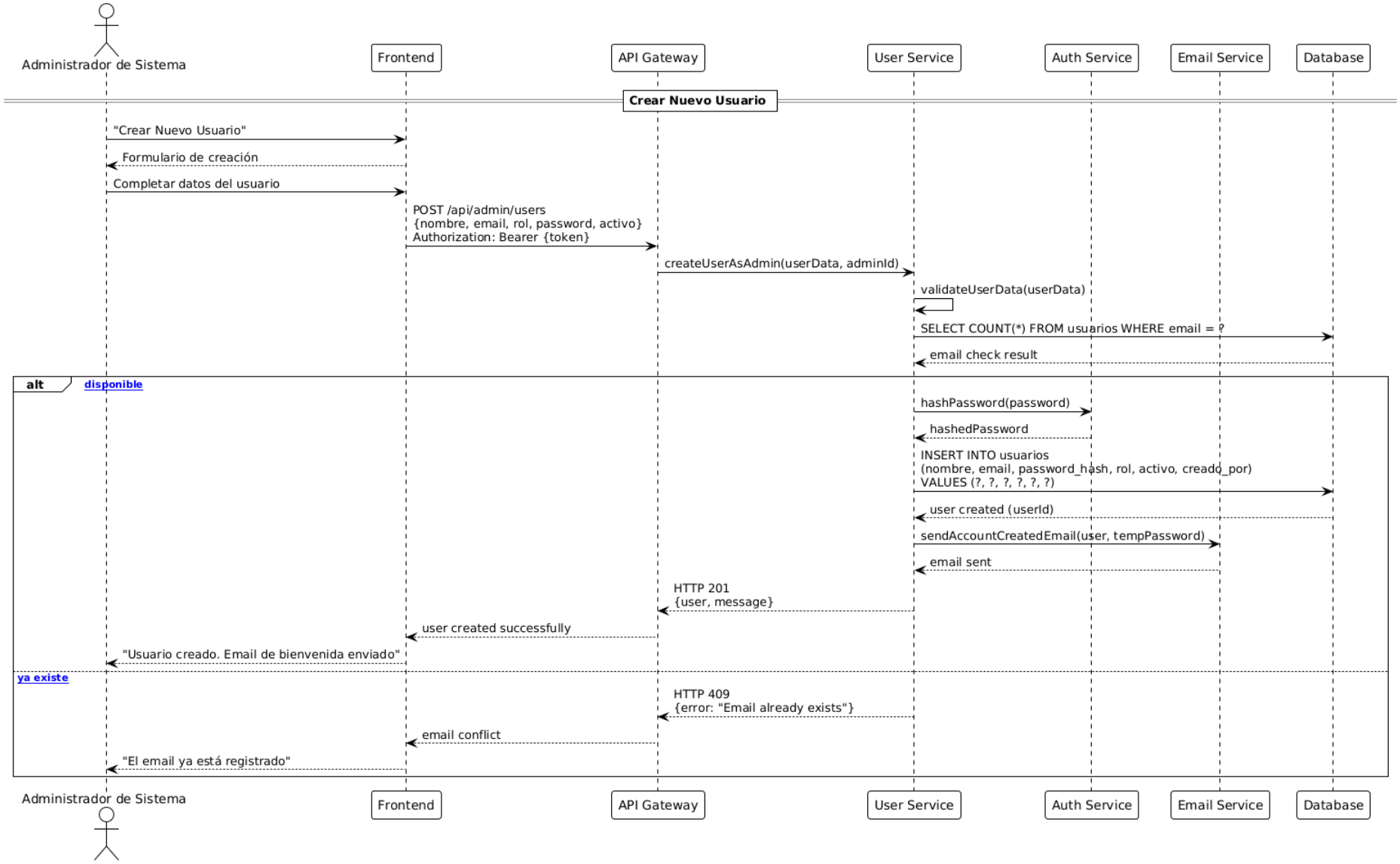


Figura 19: Diagrama de Secuencia - Administrador (Crear Usuario)

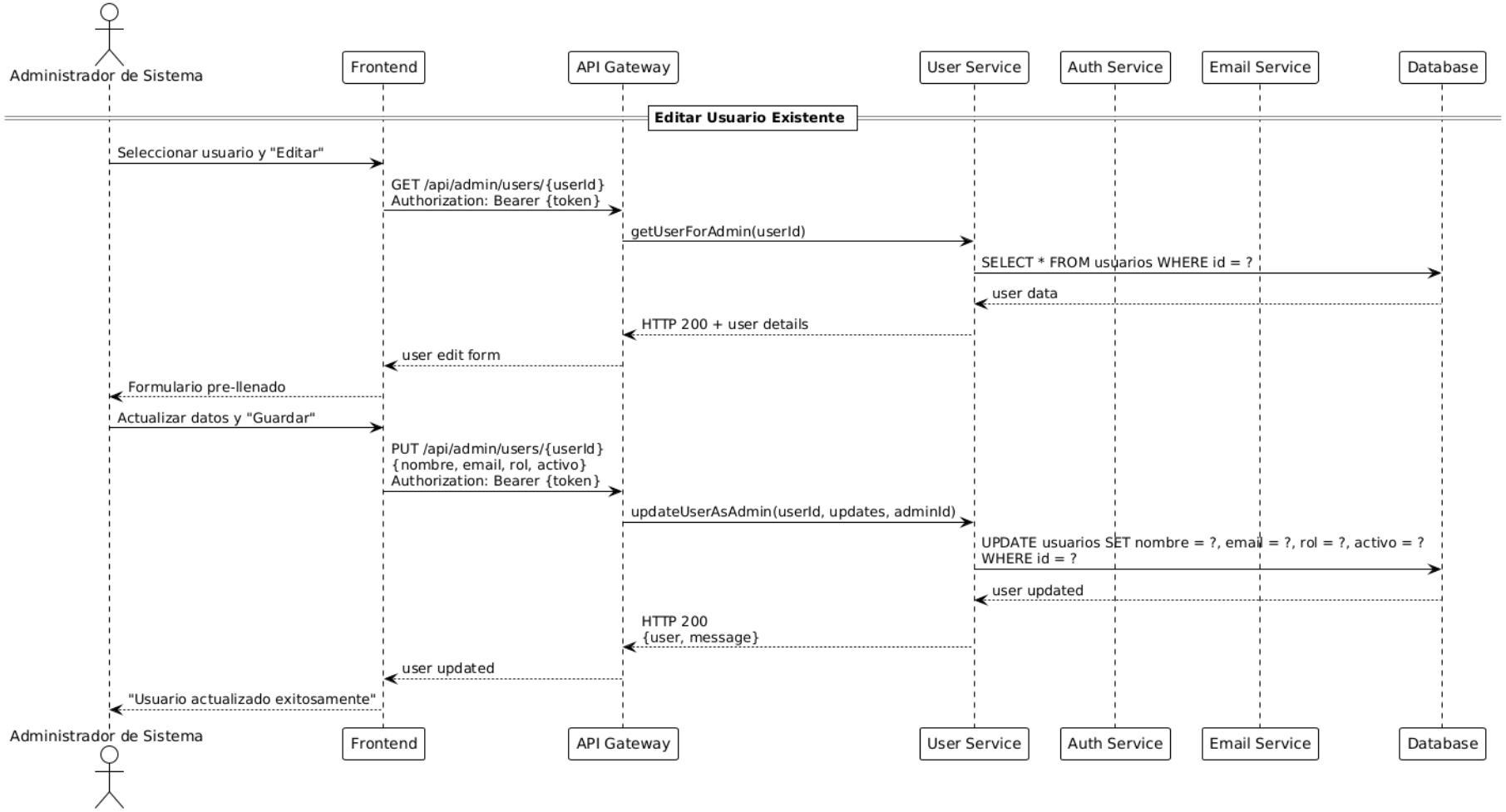


Figura 20: Diagrama de Secuencia - Administrador (Editar Usuario)

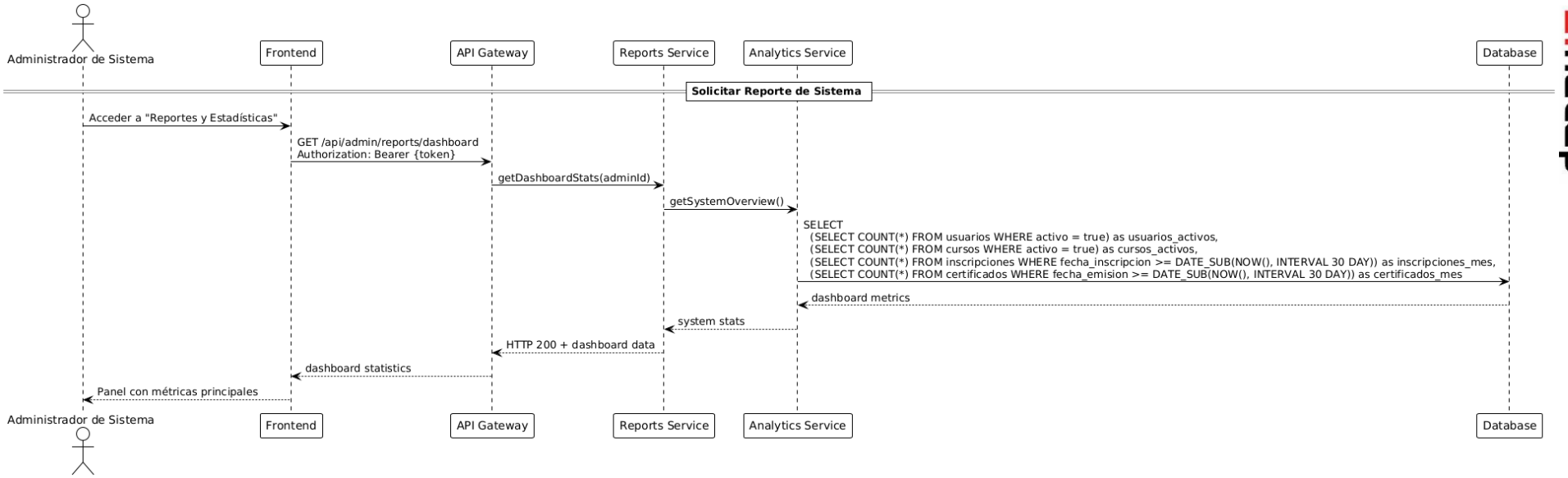


Figura 21: Diagrama de Secuencia - Administrador (Solicitar Reporte del Sistema)

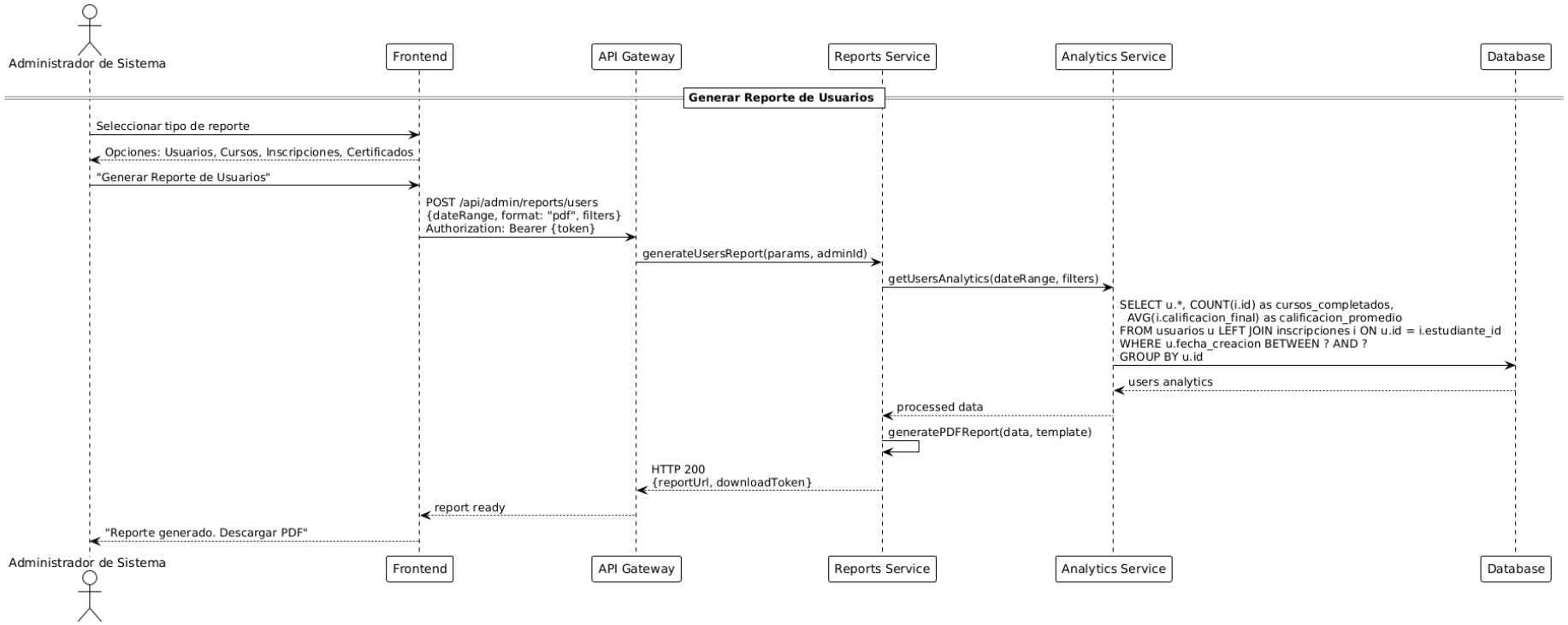


Figura 22: Diagrama de Secuencia - Administrador (Generar Reporte de Usuarios)

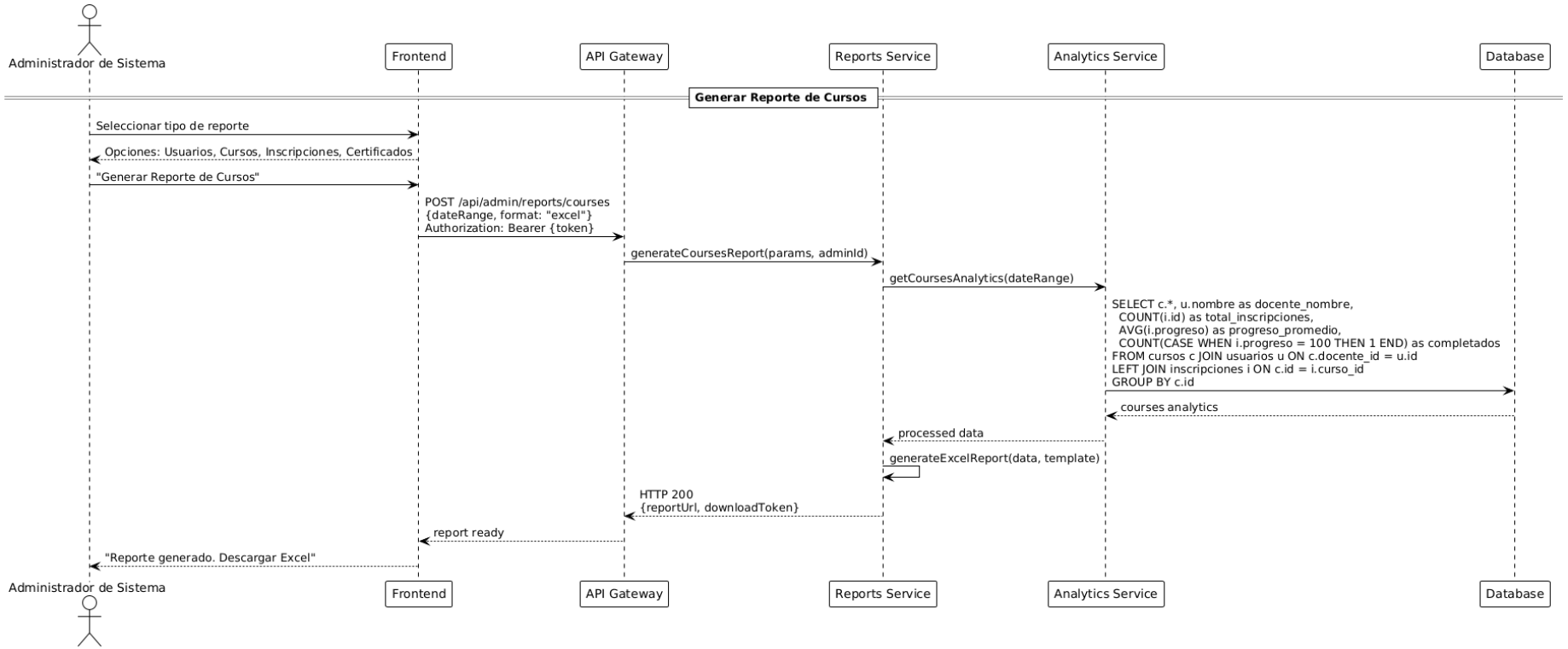


Figura 23: Diagrama de Secuencia - Administrador (Generar Reporte de Cursos)

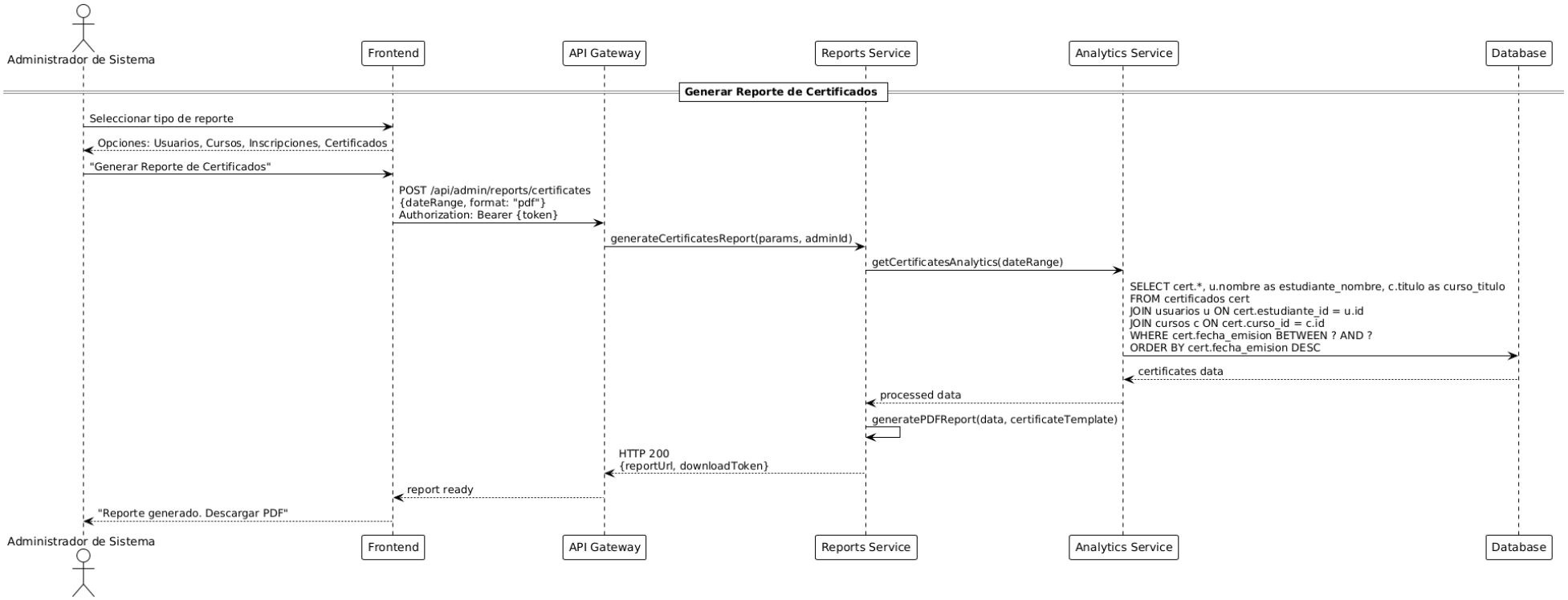


Figura 24: Diagrama de Secuencia - Administrador (Generar Reporte de Certificados)

6.4 Diagrama de Actividad: Flujo de curso online

Modela el flujo lógico completo que experimenta un estudiante desde el acceso inicial hasta la obtención del certificado final.

- El proceso inicia con la autenticación del usuario y la selección de cursos disponibles en el catálogo.
- Tras la inscripción exitosa, el estudiante accede a una estructura modular de contenidos donde puede consumir materiales de aprendizaje de forma paralela (visualización de videos, descarga de documentos PDF, y participación en actividades prácticas).
- El diagrama incluye decisiones clave como la verificación de completitud de módulos, la evaluación mediante quizzes con posibilidad de re-intentos según configuración, y la validación de criterios de aprobación.
- El flujo incorpora actividades concurrentes de actualización de progreso y cálculo de calificaciones en tiempo real.
- La culminación exitosa del proceso resulta en la generación automática de certificados digitales con códigos QR únicos, el almacenamiento seguro en el sistema, y la notificación al estudiante con múltiples opciones de entrega (descarga directa y envío por correo electrónico).



Figura 25: Diagrama de Actividad

7 Patrones de Diseño Aplicados

La implementación de patrones de diseño en el desarrollo de software es una práctica fundamental que permite crear sistemas más mantenibles, escalables y robustos (Gamma et al., 1995). En este proyecto se aplicaron tres patrones principales que abordan diferentes aspectos arquitectónicos del sistema.

7.1 Patrón Observer - Sistema de Notificaciones

Problema Resuelto: Necesidad de ejecutar acciones automáticas cuando ocurren eventos específicos en el sistema (ej: enviar email cuando un estudiante se inscribe, actualizar estadísticas cuando se completa un curso).

Justificación: Django proporciona un sistema de señales (signals) que implementa naturalmente el patrón Observer (Django Software Foundation, 2023), permitiendo que múltiples componentes reaccionen a eventos sin acoplamiento directo.

Beneficios:

- **Desacoplamiento:** Los modelos no necesitan conocer qué acciones se ejecutan tras sus cambios.
- **Extensibilidad:** Fácil agregar nuevas funcionalidades sin modificar código existente.
- **Mantenibilidad:** Cada observer maneja una responsabilidad específica.
- **Testabilidad:** Cada observer puede ser testeado independientemente.

7.2 Patrón Singleton - Gestión de Configuraciones

Problema Resuelto: Garantizar que ciertas configuraciones y recursos del sistema tengan una única instancia global, como configuraciones de integración con servicios externos o managers de conexión.

Justificación: En una plataforma educativa, necesitamos garantizar que configuraciones críticas como límites de archivos, configuraciones de email, o conexiones de APIs externas se manejen de forma centralizada y consistente, siguiendo las recomendaciones establecidas (Gamma et al., 1995).

Beneficios:

- **Consistencia:** Una sola fuente de verdad para configuraciones críticas.
- **Eficiencia:** Evita múltiples instancias de recursos costosos.
- **Control de acceso:** Punto único de acceso a configuraciones sensibles.
- **Facilidad de testing:** Configuraciones centralizadas son más fáciles de mockear.

7.3 Patrón Factory - Creación de Contenido Multimedia

Problema Resuelto: La plataforma debe manejar diferentes tipos de contenido (videos, PDFs, presentaciones, quizzes) donde cada tipo requiere procesamiento y validaciones específicas.

Justificación: En lugar de tener múltiples constructores o lógica condicional dispersa, el patrón Factory centraliza la creación de objetos complejos basándose en parámetros de entrada, como recomiendan los expertos en diseño de software (Gamma et al., 1995).

Beneficios:

- **Extensibilidad:** Fácil agregar nuevos tipos de contenido
- **Encapsulación:** La lógica de creación está centralizada
- **Consistencia:** Todos los objetos se crean siguiendo el mismo patrón
- **Mantenibilidad:** Cambios en la creación se hacen en un solo lugar

7.4 Justificación de Patrones No implementados

Adapter Pattern: Aunque útil para integrar sistemas externos, en esta fase del proyecto no tenemos suficientes integraciones complejas que justifiquen su implementación. Las integraciones actuales (sistema de pagos, email) son suficientemente simples para ser manejadas directamente.

Estos patrones de diseño complementan la arquitectura general basada en Django, operando a diferentes niveles de abstracción para crear un sistema robusto y mantenible.

8 Estándares de Calidad

La calidad del software debe ser planificado, medido y controlado desde las etapas tempranas del proyecto. Para garantizar que la Plataforma de Gestión de Cursos Online cumpla con los niveles de calidad esperados, se utilizan estándares internacionales reconocidos que proporcionan marcos de trabajo estructurados para la gestión de procesos y la evaluación de atributos de calidad.

8.1 ISO/IEC 12207 - Procesos del Ciclo de Vida del Software

El estándar ISO/IEC 12207 ("ISO/IEC 12207:2017 - Systems and software engineering – Software life cycle processes", 2017) establece un marco común para los procesos del ciclo de vida del software, proporcionando una terminología bien definida que puede ser utilizada por la industria del software.

Relevancia para el proyecto: Este estándar es especialmente relevante para nuestro proyecto ya que define procesos que van desde la conceptualización hasta el retiro del software, asegurando que todas las actividades necesarias sean consideradas y ejecutadas de manera sistemática.

Procesos Principales Implementados

Proceso	Implementación en el Proyecto
Adquisición	Definición de requerimientos con stakeholders, evaluación de proveedores de cloud
Suministro	Entrega incremental de funcionalidades cada sprint, documentación de APIs
Desarrollo	Aplicación de metodología Scrum, desarrollo dirigido por pruebas (TDD)
Operación	Monitoreo continuo, sistema de logging, métricas de performance
Mantenimiento	Plan de actualizaciones, versionado semántico, gestión de bugs

Cuadro 12: Procesos de Implementación en el Proyecto

8.2 ISO/IEC 25000 (SQuaRE) - Atributos de Calidad

El estándar ISO/IEC 25000, conocido como SQuaRE ("ISO/IEC 25000:2014 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE", 2014) proporciona guías para el uso de las nuevas series de estándares internacionales, reemplazando a las antiguas ISO/IEC 9126 e ISO/IEC 14598.

Importancia para la plataforma educativa: En el contexto de una plataforma de cursos online, la calidad del software impacta directamente en la experiencia de aprendizaje de los estudiantes y la eficiencia de los instructores (Wahyudin et al., 2020). SQuaRE nos permite definir métricas objetivas y medibles para estos atributos de calidad.

Enfoque de medición: Se ha adoptado un enfoque preventivo donde la calidad se mide desde las etapas tempranas del desarrollo, utilizando métricas automatizadas cuando es posible y estableciendo objetivos cuantitativos para cada característica de calidad.

Características de calidad medibles

Característica	Sub Característica	Métrica	Objetivo
Funcionalidad	Completitud funcional	% funciones implementadas	$> 95\,\%$
	Corrección funcional	% test cases pasados	$> 98\,\%$
	Pertinencia funcional	SUS Score	> 80
Eficiencia	Tiempo de respuesta	Latencia promedio	< 2 seg
	Utilización de recursos	CPU / Memory usage	$< 70\,\%$
Compatibilidad	Coexistencia	Navegadores soportados	5 principales
	Interoperabilidad	APIs conformes REST	$100\,\%$
Usabilidad	Reconocibilidad	Tiempo para identificar función	< 30 seg
	Aprendizaje	Tiempo para completar tareas	< 5 min
Confiabilidad	Madurez	MTBF (Mean Time Between Failures)	> 720 hrs
	Disponibilidad	Uptime	$> 99.5\,\%$
Seguridad	Confidencialidad	Datos encriptados	$100\,\%$
	Integridad	Checksum validation	$100\,\%$
Mantenibilidad	Modularidad	Acoplamiento entre módulos	Bajo
	Reusabilidad	% código reutilizable	$> 60\,\%$
Portabilidad	Adaptabilidad	Plataformas soportadas	3+ SO
	Instalabilidad	Tiempo de setup	< 30 min

Cuadro 13: Características de Calidad del Software

9 Matriz de Trazabilidad (RTM)

La Matriz de Trazabilidad de Requerimientos (RTM) se elaboró con el propósito de garantizar la coherencia entre los requerimientos levantados, los diagramas UML y los criterios de aceptación definidos, siguiendo las mejores prácticas establecidas en la ingeniería de software (Petersen et al., 2018). Este instrumento permite mantener un control claro sobre qué funcionalidades están cubiertas, cómo se representan en los modelos y de qué manera se validará su cumplimiento durante el desarrollo y las pruebas.

ID Req	Requisito	Diagrama Relacionado	Criterio de Aceptación
RF1	Registro e inicio de sesión	Caso de uso: Autenticarse	Usuario inicia sesión en $\leq$10s con correo/clave o OAuth válido.
RF2	Gestión de cursos	Caso de uso: Administrar cursos	El docente crea/edita un curso y aparece en catálogo.
RF3	Subida de materiales	Caso de uso: Gestionar contenidos	El material se sube con validación de formato/tamaño.
RF4	Inscripción en cursos	Caso de uso: Inscribirse a curso	Estudiante inscrito y aparece en listado.
RF5	Visualización de progreso	Diagrama de clases (Inscripción)	% y nota final actualizados en tiempo real.
RF6	Evaluaciones en línea	Caso de uso: Rendir evaluación	Estudiante completa quiz, profesor recibe nota.
RF7	Emisión de certificados	Secuencia: Quiz $\rightarrow$ Certificado	PDF con QR generado y verificable en $\leq$10s.
RF8	Notificaciones	Caso de uso: Notificación	Mensaje llega al panel/correo en $\leq$1 min.
RF9	Administración de cuentas y roles	Diagrama de clases (Usuario/Admin)	Admin crea/edita usuarios y roles desde panel.
RF10	Historial académico	Caso de uso: Consultar historial	Estudiante visualiza cursos, notas y certificados.
RF11	Resúmenes IA	Caso de uso: Acceder a material	Estudiante ve resumen textual de la clase.

Continúa en la siguiente página...

Tabla – continuación de la página anterior

ID Req	Requisito	Diagrama Relacionado	Criterio de Aceptación
RF12	Auditoría básica	Diagrama de clases (Logs)	Admin consulta accesos e inscripciones con fecha/hora.
RF13	Verificación/revocación certificados	Caso de uso: Verificar Certificado	URL pública refleja estado (válido/revocado).
RF14	Intentos y ponderaciones	Caso de uso: Rendir evaluación	Nota final calculada según reglas configuradas.
RF15	Cupos y prerrequisitos	Caso de uso: Inscribirse a curso	Sistema bloquea inscripción si cupo lleno o requisito no cumplido.
RF16	Gestión segura de archivos	Caso de uso: Gestionar contenidos	Archivos verificados (formato, tamaño, antivirus).
RF17	Búsqueda/paginación	Diagrama de actividad (Catálogo)	Catálogo lista 20 resultados por página y permite búsqueda.
RNF1	Seguridad	Diagrama de clases (Usuario/Roles)	Todo tráfico en HTTPS, hash seguro, RBAC activo.
RNF4	Usabilidad	Mockups UI / Prototipo	Puntaje $\geq$90 en Lighthouse móvil.
RNF7	Accesibilidad	Prototipo / UI	Cumplimiento WCAG 2.1 AA validado.
RNF8	Escalabilidad	Diagrama de despliegue	Respuesta $\leq$3s con 500 usuarios concurrentes.
RNF9	Confiabilidad	Caso de uso: Restaurar BD	Backup restaurado en $\leq$30 min.
RNF12	Observabilidad	Dashboard de monitoreo	Alertas automáticas si error $>$1 % o p95 $>$3s.

Cuadro 14: Requisitos y Diagramas Relacionados

La RTM asegura la trazabilidad completa de los requerimientos y sirve como base para identificar los riesgos asociados al cumplimiento de cada funcionalidad y atributo de calidad.

9.1 Identificación de Riesgos

La identificación de riesgos permite anticipar posibles problemas que podrían afectar el desarrollo y la operación de la plataforma, aplicando metodologías reconocidas de gestión de riesgos en proyectos ágiles (Boehm & Turner, 2005). Para cada riesgo se evaluó la probabilidad, el impacto y el nivel resultante, además de definir medidas de mitigación que reduzcan su efecto en el proyecto.

Riesgo ID	Riesgo identificado	Probabilidad	Impacto	Nivel	Medidas de mitigación
R1	Fallas en autenticación (login/OAuth)	Alta	Alto	Crítico	Autenticación con tokens JWT, pruebas de estrés, recuperación segura de contraseña.
R2	Pérdida de datos académicos	Media	Alto	Alto	Backups automatizados diarios, redundancia en BD, replicación geográfica.
R3	Ataques de seguridad (SQLi, XSS)	Alta	Alto	Crítico	Validación de entradas, uso de ORM, pruebas de penetración, sanitización de datos.
R4	Baja adopción por usabilidad	Media	Medio	Medio	Pruebas de usabilidad con usuarios reales, capacitación inicial, documentación clara.
R5	Sobrecarga con usuarios simultáneos	Alta	Medio	Alto	Load balancing, auto-scaling, pruebas de carga, optimización de consultas.
R6	Caída del servidor principal	Baja	Alto	Medio	Infraestructura en nube con failover automático, monitoreo 24/7.
R7	Retrasos en la entrega del proyecto	Baja	Medio	Bajo	Planificación ágil, revisiones quincenales, gestión proactiva de dependencias.

Continúa en la siguiente página...

Tabla – continuación de la página anterior

Riesgo ID	Riesgo identificado	Probabilidad	Impacto	Nivel	Medidas de mitigación
R8	Vulneración de datos personales	Media	Alto	Alto	HTTPS obligatorio, encriptación en reposo, control de accesos RBAC, auditorías.
R9	Fallos en verificación de certificados QR	Baja	Medio	Bajo	Pruebas funcionales automatizadas, fallback de validación manual, logs detallados.
R10	Dependencia de proveedores externos (OAuth, cloud)	Media	Bajo	Bajo	Múltiples métodos de autenticación, proveedores cloud alternativos, SLAs claros.
R11	Errores tipográficos en interfaz de usuario	Baja	Bajo	Bajo	Revisión ortográfica automatizada, proceso de QA, corrección rápida mediante hotfixes.
R12	Problemas de compatibilidad entre navegadores	Alta	Baja	Medio	Testing cross-browser automatizado, polyfills, documentación de navegadores soportados.

Cuadro 15: Análisis de Riesgos del Proyecto

La identificación y análisis de riesgos proporciona una visión clara de los factores críticos del proyecto y facilita su gestión.

9.2 Matriz 3x3

Para complementar la identificación de riesgos, se elaboró una Matriz 3x3 que cruza la probabilidad y el impacto de cada evento. Esta representación gráfica facilita la priorización, destacando los riesgos críticos que requieren mayor atención y control durante el desarrollo del proyecto.

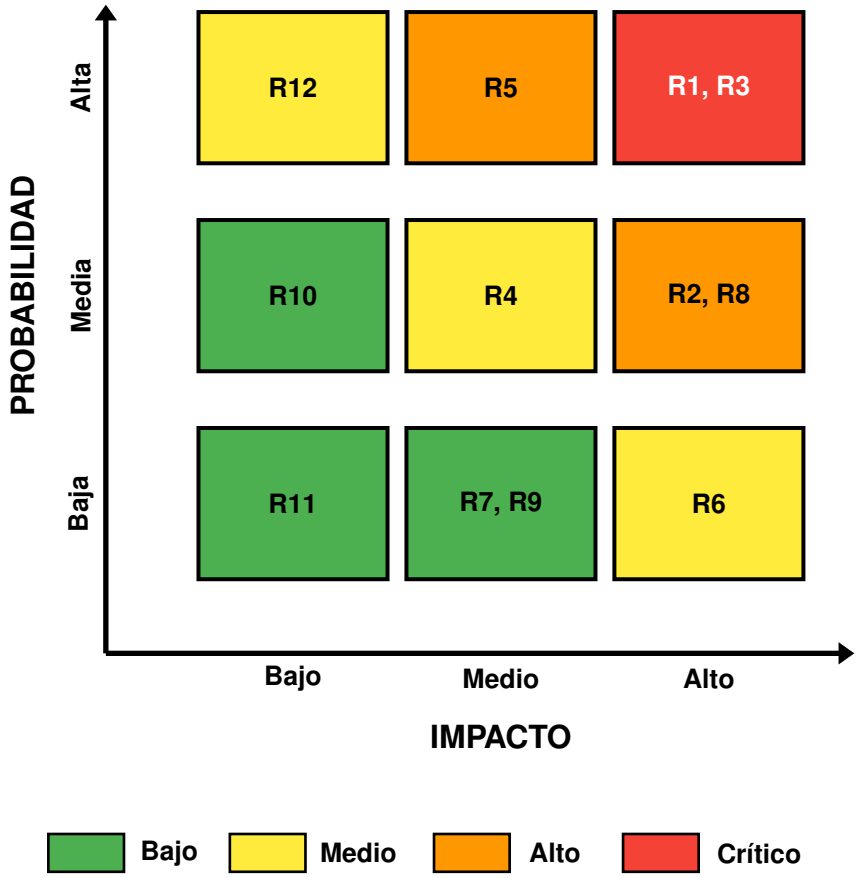


Figura 26: Matriz 3×3 de Evaluación de Riesgos del Proyecto

La matriz permite visualizar de forma clara y rápida la concentración de riesgos en cada nivel, apoyando la toma de decisiones sobre las acciones de mitigación más adecuadas.

10 Conclusiones

El desarrollo de la plataforma de gestión de cursos online permitió aplicar de manera práctica los conceptos fundamentales de la Ingeniería de Software, integrando desde el levantamiento de requerimientos hasta la planificación ágil, el modelamiento UML, el uso de patrones de diseño y la incorporación de estándares de calidad. El trabajo realizado evidencia la importancia de contar con una metodología clara (Scrum), requerimientos bien definidos y trazables, así como con diagramas y patrones que orienten la implementación técnica de forma ordenada y mantenible.

La solución propuesta responde a las necesidades de estudiantes, docentes y administradores, ofreciendo un sistema capaz de gestionar cursos, materiales, evaluaciones y certificados digitales verificables, junto con un módulo de inteligencia artificial que enriquece la experiencia de aprendizaje. Además, la aplicación de los estándares internacionales ISO/IEC 12207 y ISO/IEC 25000 garantiza que el sistema no solo cumpla con las funcionalidades esperadas, sino que también mantenga atributos de calidad medibles como rendimiento, seguridad, usabilidad y confiabilidad.

En definitiva, este proyecto constituye un ejemplo de cómo la correcta integración de técnicas, herramientas y buenas prácticas de la Ingeniería de Software permite construir soluciones tecnológicas que son funcionales, escalables y alineadas con las expectativas de los usuarios y los objetivos académicos.

11 Bibliografía

Aparicio, M., Bacao, F., & Oliveira, T. (2016). An e-Learning Theoretical Framework. *Educational Technology & Society*, *19*(1), 292-307. https://www.jstor.org/stable/jeductechsoci.19.1.292

Boehm, B., & Turner, R. (2005). Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software*, *22*(5), 30-39. https://doi.org/10.1109/MS.2005.129

Bourque, P., & Fairley, R. E. (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). *IEEE Computer Society*. https://www.computer.org/education/bodies-of-knowledge/software-engineering

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional. https://www.mountaingoatsoftware.com/books/user-stories-applied

Django Software Foundation. (2023). Django Documentation. https://docs.djangoproject.com/en/4.2/

Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3.ª ed.). Addison-Wesley Professional. https://martinfowler.com/books/uml.html

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional. https://www.oreilly.com/library/view/design-patterns-elements/0201633612/

IEEE Computer Society. (1998). *IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications* (inf. téc.). Institute of Electrical y Electronics Engineers. New York, NY. https://standards.ieee.org/ieee/830/1223/

ISO/IEC 12207:2017 - Systems and software engineering – Software life cycle processes. (2017). International Organization for Standardization. https://www.iso.org/standard/63712.html

ISO/IEC 25000:2014 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. (2014). International Organization for Standardization. https://www.iso.org/standard/64764.html

Kniberg, H. (2015). *Scrum and XP from the Trenches* (2.ª ed.). InfoQ. https://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2/

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (3.ª ed.). Prentice Hall. https://www.pearson.com/us/higher-education/program/Larman-Applying-UML-and-Patterns-An-Introduction-to-Object-Oriented-Analysis-and-Design-and-Iterative-Development-3rd-Edition/PGM13068.html

Petersen, K., Wohlin, C., & Baca, D. (2018). The Waterfall Model in Large-Scale Development. *Information and Software Technology*, *98*, 68-82. https://doi.org/10.1016/j.infsof.2018.02.002

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional. https://www.amazon.com/Lean-Software-Development-Agile-Toolkit/dp/0321150783

Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8.ª ed.). McGraw-Hill Education. https://www.mheducation.com/highered/product/software-engineering-practitioner-s-approach-pressman-maxim/M9780078022128.html

Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional. https://www.amazon.com/Essential-Scrum-Practical-Popular-Process/dp/0137043295

Sanders, E. B.-N., & Stappers, P. J. (2008). *Co-creation and the new landscapes of design*. Taylor & Francis. https://www.taylorfrancis.com/books/mono/10.4324/9780203855928/co-creation-new-landscapes-design-elizabeth-sanders-pieter-stappers

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum*. Scrum.org. https://scrumguides.org/scrum-guide.html

Sommerville, I. (2016). *Software Engineering* (10.ª ed.). Pearson. https://www.pearson.com/us/higher-education/program/Sommerville-Software-Engineering-10th-Edition/PGM58925.html

Wahyudin, D., Abdullah, A. G., Morales, M. P., & Virtanen, M. A. (2020). Systematic Literature Review of Learning Management System Features for Academic Information System. *International Journal of Distance Education Technologies*, *18*(4), 1-22. https://doi.org/10.4018/IJDET.2020100101