

# Entorno Virtual de Aprendizaje



## Informe Final

ADÁN - Grupo 6

### Integrantes:

- Eduardo Alvarez
- Marcelo Bravo
- Iván Castro
- Bruno Menendez
- Maia Oldak
- Manuel Sainz

Tutor: Ing. Federico Gómez Frois

<b>1. Introducción.....</b>	<b>4</b>
<b>2. Objetivos planteados y resultados esperados.....</b>	<b>5</b>
<b>3. Estado del arte.....</b>	<b>6</b>
3.1 Introducción.....	6
3.2 Marco Teórico.....	6
3.3 Plataformas analizadas.....	8
3.4 Análisis comparativo.....	13
3.5 Conclusiones.....	14
<b>4. Análisis y Diseño.....</b>	<b>17</b>
4.1 Perfiles de usuario.....	17
4.2 Requisitos del Sistema.....	18
4.3 Requerimientos específicos.....	18
4.4 Modelo de Casos de Uso.....	21
4.5 Modelo de dominio.....	25
4.6 Arquitectura del sistema.....	27
4.7 Diseño del sistema.....	29
<b>5. Gestión del proyecto.....</b>	<b>36</b>
5.1 Metodología de trabajo.....	36
5.2 Distribución de Roles.....	37
5.3 Planificación.....	38
5.4 Horas dedicadas.....	40
5.5 Problemas encontrados.....	41
<b>6. Implementación.....</b>	<b>43</b>
6.1 Tecnologías aplicadas.....	43
6.2 Aplicación desarrollada.....	44
<b>7. Testing y validación.....</b>	<b>47</b>
7.1 Plan de testing.....	47
7.2 Resultados obtenidos.....	48
7.3 Conclusión.....	54
<b>8. Conclusiones y trabajo a futuro.....</b>	<b>55</b>
8.1 Conclusiones finales.....	55
8.2 Trabajo a futuro.....	56
<b>9. Referencias.....</b>	<b>57</b>
<b>10. Anexos.....</b>	<b>59</b>
10.1 Documento de Alcance.....	59
10.2 Documento de Arquitectura.....	59
10.3 Documento de Casos de Uso.....	59
10.4 Documento de Diseño.....	59
10.5 Documento de Requerimientos.....	59
10.6 Documento de Pruebas de Concepto.....	60
10.7 Documento de Verificación.....	60
10.8 Documento de Modelos de Datos.....	60
10.9 Documento de Modelo de Dominio.....	60
10.10 Glosario.....	60



## Resumen

El objetivo de este documento es centralizar y organizar la información relacionada con el desarrollo de la plataforma “ADÁN”, creada en el marco de la asignatura Proyecto de la carrera Tecnólogo en Informática. ADÁN es un Entorno Virtual de Aprendizaje (EVA) que no se dirige a un nivel educativo particular ni a un medio de distribución específico, ya que ofrece herramientas útiles para todos los niveles de enseñanza y para usuarios que accedan tanto desde la web como desde dispositivos móviles.

La plataforma incorpora tres perfiles de usuario, cada uno cuenta con diferentes funcionalidades y permisos:

- **Administrador:** Es el perfil con mayor autoridad dentro del sistema. Puede gestionar cursos y usuarios, y además tiene acceso a la totalidad de los cursos disponibles.
- **Profesor:** Puede gestionar sus cursos asignados, incluyendo la administración de su contenido y de los estudiantes matriculados. Puede calificar estudiantes y tareas y puede participar en los foros.
- **Estudiante:** Solo puede visualizar sus cursos asignados y sus recursos, sin capacidad de gestión. Puede realizar tareas y participar en los foros.

Los tres perfiles pueden acceder a su perfil y modificarlo. Además, los perfiles de profesor y estudiante comparten algunas funciones, como recibir notificaciones y el uso del sistema de mensajería interna.

Para el desarrollo de la plataforma se emplearon tecnologías modernas y se definió una arquitectura escalable, con el objetivo de simplificar su mantenimiento a futuro.

En conclusión, ADÁN es una plataforma sólida e intuitiva, diseñada para atender las necesidades de distintos tipos de usuarios y brindarles herramientas que acompañen su actividad académica, ya sean profesores o estudiantes de cualquier nivel educativo.

## Palabras Clave

ADÁN, Entorno Virtual de Aprendizaje, EVA, enseñanza, profesor, estudiante, web, móvil, Java, Spring Boot, React, Kotlin.



# 1. Introducción

Los Entornos Virtuales de Aprendizaje nacen con el propósito de facilitar la difusión del conocimiento y apoyar el trabajo académico tanto dentro como fuera del aula. Funcionan como una especie de aula virtual asincrónica, donde los estudiantes pueden acceder a contenidos, tareas e información relevante proporcionada por sus docentes o por las instituciones educativas a las que pertenecen.

Estos entornos han adquirido aún mayor relevancia en los últimos años debido al creciente acceso a dispositivos móviles y computadoras por parte de la población en general, así como por situaciones excepcionales como la pandemia de COVID-19. Entre 2020 y 2021, la emergencia sanitaria global imposibilitó la asistencia presencial a las aulas, obligando a recurrir a alternativas como los EVA para asegurar la continuidad del proceso educativo durante ese periodo.

Dentro de este contexto surge la plataforma ADÁN, cuyo propósito es ofrecer herramientas que faciliten la gestión de actividades educativas en distintos niveles de enseñanza.

En este documento se detalla el proceso de desarrollo de la plataforma, enfocándose en los siguientes puntos:

- **Objetivos planteados y resultados esperados**  
Se describen los objetivos a completar y el alcance del proyecto.
- **Estado del arte**  
Se presenta un análisis de distintas plataformas existentes, comparando sus características, similitudes y diferencias.
- **Análisis y diseño**  
Se describe todo lo relacionado con el análisis y el diseño del sistema.
- **Gestión del proyecto**  
Se describe el enfoque de gestión adoptado durante el desarrollo del proyecto.
- **Implementación**  
Se detallan las tecnologías utilizadas durante el desarrollo del proyecto.
- **Testing y validación**  
Se muestra el proceso de verificación y prueba de las distintas funcionalidades del sistema.
- **Conclusiones y trabajo a futuro**  
Se muestran los resultados obtenidos, las metas alcanzadas y se proponen posibles mejoras y ampliaciones del sistema.



## 2. Objetivos planteados y resultados esperados

El presente trabajo se realiza en el marco de la asignatura Proyecto de la carrera Tecnólogo en Informática, impartida por la Facultad de Ingeniería de la Universidad de la República y la Dirección General de Educación Técnico Profesional, siendo el objetivo de esta carrera la formación de profesionales técnicos capacitados para diseñar, desarrollar y mantener soluciones informáticas.

El proyecto se enfoca en el análisis, planificación, documentación, desarrollo, implementación y verificación del Entorno Virtual de Aprendizaje ADÁN. Esta plataforma tiene como objetivo proporcionar un entorno virtual que facilite el aprendizaje remoto y complemente la enseñanza presencial en distintos niveles educativos, proporcionando una interfaz clara e intuitiva, herramientas para la gestión de cursos y sus contenidos, medios de interacción entre docentes y estudiantes, y disponibilidad tanto en versión web como en dispositivos móviles.

Para llevar adelante el desarrollo, se definieron las siguientes etapas:

- **Análisis del contexto y soluciones ya existentes**  
Se analizaron diversas plataformas ya existentes con el objetivo de identificar sus funcionalidades, fortalezas y limitaciones. También se analizó el contexto de uso para relevar las necesidades de los usuarios finales.
- **Definición de requerimientos y selección de tecnologías**  
Se determinaron las necesidades básicas que debía cubrir el sistema y se investigaron las tecnologías adecuadas para el desarrollo, teniendo en cuenta su utilidad para el desarrollo, sus capacidades y el nivel de experiencia del equipo con dichas tecnologías.
- **Diseño de la solución y planificación de la implementación**  
Se realizó el diseño general del sistema, incluyendo su arquitectura, modelo de dominio, modelo de datos, casos de uso y pruebas de concepto. Asimismo, se planificaron las funcionalidades a desarrollar, organizándolas en dos sprints según su grado de relevancia dentro del proyecto.
- **Desarrollo e implementación de la solución**  
Se desarrolló la plataforma de manera ordenada y sistemática, basándose en el diseño y planificación previamente establecidos.
- **Verificación y testing**  
Se realizaron pruebas de validación para comprobar que la plataforma cumpliera correctamente con los requerimientos definidos.



### 3. Estado del arte

#### 3.1 Introducción

Los Entornos Virtuales de Aprendizaje (EVA) han transformado la forma en que instituciones educativas, docentes y estudiantes interactúan, constituyéndose en una herramienta fundamental de la educación contemporánea al posibilitar nuevas modalidades de enseñanza y aprendizaje.

En los últimos años, se han desarrollado diversas soluciones digitales orientadas a la gestión de cursos, la comunicación entre profesores y alumnos, y la creación de espacios de interacción académica en entornos virtuales.

Antes de avanzar en el diseño de una propuesta propia, resulta imprescindible analizar el estado del arte, es decir, sistemas actualmente disponibles y las tecnologías predominantes, con el fin de identificar fortalezas, debilidades y tendencias que sirvan como base para la plataforma a implementar en el presente proyecto.

#### 3.2 Marco Teórico

En los últimos años, los avances tecnológicos, el acceso masivo a internet y la creciente disponibilidad de dispositivos digitales han transformado radicalmente el panorama educativo. Este cambio, junto con la necesidad generada por factores externos, han impulsado la expansión de los Entornos Virtuales de Aprendizaje (EVA), herramientas que en la actualidad se consideran esenciales para los procesos pedagógicos.

Inicialmente concebidos como un complemento a la educación presencial, los EVA han evolucionado hasta constituirse en un pilar central de la enseñanza y la formación, impulsados por la demanda creciente de modelos de aprendizaje flexibles y personalizados. En este contexto, la integración de estos sistemas en las instituciones educativas se ha vuelto prácticamente imprescindible para responder a las exigencias del panorama educativo contemporáneo.

Los EVA son plataformas digitales diseñadas específicamente para mediar y enriquecer los procesos de enseñanza y aprendizaje. Su función va más allá de ser simples repositorios de archivos, ya que operan como sistemas integrales en los que convergen aspectos pedagógicos, tecnológicos y sociales, con el fin de optimizar la experiencia educativa. [1]

Entre sus principales características se destacan [2]:

- **Gestión académica:** administración eficiente de cursos, matrículas, seguimiento del progreso y evaluación de los estudiantes.
- **Comunicación e interacción:** foros de discusión, mensajería interna, chats y videoconferencias que promueven comunidades de aprendizaje.



- **Acceso a recursos didácticos:** disponibilidad de materiales educativos en múltiples formatos (documentos, presentaciones, audios, videos, enlaces web).
- **Flexibilidad y accesibilidad:** acceso permanente a contenidos y actividades desde distintos dispositivos (computadoras, tabletas y teléfonos móviles).
- **Actividades interactivas:** herramientas para cuestionarios, wikis, tareas y ejercicios prácticos que fomentan la aplicación del conocimiento y la retroalimentación.

De esta forma, los EVA se consolidan como entornos educativos integrales que articulan elementos pedagógicos, tecnológicos y sociales orientados a mejorar la experiencia formativa.

En Uruguay, las instituciones educativas, tanto públicas como privadas y en todos los niveles de enseñanza (primaria, secundaria y terciaria), han incorporado de manera progresiva el uso de plataformas virtuales de aprendizaje. Estas herramientas, que pueden ser de código abierto o de licenciamiento comercial, ofrecen distintas funcionalidades y servicios, pero comparten un mismo propósito: facilitar la interacción entre docentes y estudiantes, apoyando los procesos de enseñanza y aprendizaje en entornos digitales.

En el sector público, una de las iniciativas más relevantes es la plataforma Crea, desarrollada en el marco del Plan Ceibal [3], que desde su implementación ha permitido extender el acceso a recursos digitales a estudiantes de primaria y secundaria en todo el país. Basada en Schoology, esta solución integra herramientas de gestión de cursos, foros de discusión, entrega de tareas y evaluaciones en línea, constituyéndose en un pilar de la educación pública uruguaya.

En el ámbito universitario, diversas facultades y centros de enseñanza terciaria utilizan plataformas como Moodle [4], debido a su carácter open source, su flexibilidad y la amplia comunidad de desarrollo que la respalda. Algunas instituciones privadas han optado por Google Classroom, aprovechando su integración con el ecosistema de Google (Drive, Meet, Calendar, Docs), lo que facilita la gestión de clases y materiales educativos en línea. Otras han implementado soluciones comerciales como Blackboard o Canvas, que si bien implican costos de licencia, ofrecen un entorno robusto, escalable y con soporte profesional.

De esta forma, se observa que el ecosistema uruguayo en materia de Entornos Virtuales de Aprendizaje es diverso, con opciones tanto gratuitas como de pago, todas orientadas a garantizar la continuidad educativa, fomentar la colaboración y responder a las necesidades de estudiantes y docentes en un contexto cada vez más digitalizado.



### 3.3 Plataformas analizadas

En esta sección se examinan diferentes Entornos Virtuales de Aprendizaje (EVA), poniendo especial atención en las plataformas más utilizadas en el ámbito educativo y en las funcionalidades que cada una ofrece. Concretamente, se analizarán las siguientes plataformas: Moodle, Schoology, Google Classrooms y Canva.

El análisis se centrará en seis características clave que permiten comparar de forma coherente las plataformas y extraer conclusiones útiles para el desarrollo del proyecto. Dichas características son:

- **Matriculación/Desmatriculación:** La manera en que los estudiantes son matriculados a los distintos cursos que ofrece una institución particular.
- **Seguimiento del avance personal:** Las funciones que faciliten al estudiante seguir su progreso dentro de un curso y/o carrera específicos. Esto puede incluir calificaciones numéricas, devoluciones, presentismo, etc.
- **Alojamiento de recursos:** La manera en que las distintas plataformas ofrecen distintos recursos para el uso de los estudiantes, ya sean imágenes, videos, audios, tareas, etc.
- **Comunicación estudiante - docente:** Las facilidades que la plataforma presta para facilitar la comunicación entre uno o más estudiantes y uno o más docentes en particular.
- **Comunicación institucional:** Las facilidades prestadas para la comunicación entre los estudiantes y la institución.
- **Disponibilidad móvil:** La disponibilidad de la plataforma en dispositivos móviles.

El análisis de estas características permite identificar fortalezas, debilidades y tendencias de cada plataforma, brindando un marco de referencia sólido para fundamentar la elección de la arquitectura tecnológica más adecuada para el proyecto.



## Moodle



Moodle es un LMS (Learning Management System) gratuito y open source cuya primera versión pública fue liberada en 2001, creado por Martin Dougiamas, fundador y presidente de Moodle. Además de los dos ya mencionados, tiene también como beneficios el ser modular, personalizable, responsive y escalable. Tiene soporte constante de la organización Moodle que con frecuencia libera versiones nuevas y de la comunidad, que desarrolla complementos o plug-ins para uso público [5].

Un ejemplo de aplicación de Moodle es el Campus Virtual perteneciente a la DGETP-UTU (Dirección General de Educación Técnico Profesional y Universidad del Trabajo del Uruguay) que brinda entornos virtuales de aprendizaje basados en la anteriormente mencionada plataforma, para dar soporte a estudiantes y docentes de todo el país en carreras de nivel superior pertenecientes a estas entidades desde el año 2008.

A continuación se presenta el análisis en base a los 6 puntos antes mencionados:

Función	Descripción
Matriculación / desmatriculación	Se proveen métodos de matriculación manual al estudiante, así como matriculación manual por parte del docente o automaticación respondiendo a distintos criterios, por ejemplo si el estudiante pertenece a una cierta carrera.
Seguimiento del avance personal	Se cuenta con un Libro de Calificaciones, donde se registran las calificaciones de los estudiantes referidas a una actividad particular, ya sea automáticamente al finalizar la actividad o manualmente por parte del docente. Asimismo se cuenta con un sistema de competencias relacionadas a cada curso en las que los estudiantes son evaluados. Los estudiantes pueden acceder a ambas herramientas, así como a las condiciones de finalización de cada curso para tener una idea clara de su progreso.
Alojamiento de recursos	Puede hospedar archivos de texto, video, audio e imágenes para ser utilizados por los estudiantes. Se pueden publicar tareas de varios tipos, como cuestionarios embebidos en la página, hasta tareas que requieran la entrega de soluciones en forma de archivos, los cuales los estudiantes pueden subir al sitio y el docente puede corregir para dar una devolución y/o calificación.
Comunicación estudiante - docente	Permite la creación de foros específicos a las materias, así como foros específicos a temas particulares. Se incorpora un sistema de chat con el que los estudiantes y docentes pueden interactuar.
Comunicación institucional	En la página principal se muestra un apartado de novedades donde se publican comunicados de las distintas instituciones.
Disponibilidad móvil	Se puede acceder a través de navegadores en dispositivos móviles.



## Schoology



Schoology es un LMS de la compañía norteamericana Powerschool, basado en la nube utilizando el modelo de Software as a Service(SaaS) y enfocado principalmente a la educación K-12, o entre años 5 y 18, con la idea de brindarle al estudiante una única plataforma para su uso en su vida académica comprendida entre estos años. Al mismo tiempo busca presentarle al docente un lugar donde centralizar los distintos aspectos de su labor [6]. Esta plataforma fue puesta en funcionamiento a finales del año 2009 [7], siendo adquirida por su actual propietario (Powerschool) en el año 2019 [8].

Un ejemplo de aplicación de Schoology es la plataforma uruguaya CREA (siendo CREA2 su denominación oficial [9]) utilizada por docentes y estudiantes comprendidos desde nivel inicial hasta bachillerato de todo el país.

A continuación se presenta el análisis en base a los 6 puntos antes mencionados:

Función	Descripción
Matriculación / desmatriculación	La matriculación de los cursos es automática, a cargo de la institución.
Seguimiento del avance personal	Cuenta con calificaciones generales y de tareas específicas, además de un historial de calificaciones de cursos anteriores del estudiante.
Alojamiento de recursos	Se hospedan materiales, recursos y tareas a los que los estudiantes pueden acceder.
Comunicación estudiante - docente	Cuenta con un sistema de chat que estudiantes y docentes pueden utilizar para interactuar entre sí.
Comunicación institucional	Se muestran comunicados y noticias de interés tanto de cada institución como en general en la página principal.
Disponibilidad móvil	Se accede a la plataforma a través de la app de Schoology, con funcionalidades reducidas principalmente para el docente.



## Google Classroom



Google Classroom es una plataforma de gestión del aprendizaje desarrollada por Google y lanzada en 2014 como parte del ecosistema Google Workspace for Education. Se ofrece de manera gratuita a instituciones educativas y docentes individuales, y permite integrarse con otros servicios de Google, como Drive, Docs, Sheets, Slides, Calendar y Meet, lo que facilita la organización de clases, la distribución de materiales, la comunicación y la evaluación de los estudiantes [10].

La plataforma es utilizada por millones de usuarios en todo el mundo, con especial presencia en educación primaria y secundaria, aunque también se emplea en el ámbito universitario. Su facilidad de uso, gratuidad y rápida implementación la convierten en una herramienta altamente valorada por colegios y docentes a nivel global.

A continuación se presenta el análisis en base a los 6 puntos antes mencionados:

Función	Descripción
Matriculación / desmatriculación	Permite inscribirse en cursos mediante invitación del docente o código del curso, los estudiantes también pueden abandonar un curso si lo desean.
Seguimiento del avance personal	Permite ver tareas asignadas, entregadas y calificadas, y un estudiante puede seguir su progreso en cada curso. No ofrece un seguimiento completo de toda la carrera.
Alojamiento de recursos	Los docentes pueden subir materiales, enlaces, videos y tareas, y los estudiantes pueden entregarlas directamente en la plataforma.
Comunicación estudiante - docente	Mediante comentarios en tareas, anuncios del curso y correo integrado, los estudiantes pueden comunicarse con docentes.
Comunicación institucional	No está diseñado como red social institucional, por lo tanto no permite comunicados “generales” de la institución orientados al público.
Disponibilidad móvil	Cuenta con una aplicación móvil, con funcionalidades limitadas para el docente.



## Canvas



Canvas es un sistema de gestión del aprendizaje desarrollado por la empresa estadounidense Instructure, Inc., la misma fué fundada en 2008 con sede en Salt Lake City, Utah. La primera versión de Canvas fue lanzada en 2011, con un enfoque en la usabilidad, accesibilidad y escalabilidad en la nube [11].

Canvas es una solución comercial disponible principalmente bajo modalidad SaaS (Software as a Service), aunque también está disponible en modalidad on-premise. Es compatible con estándares internacionales de e-learning que garantizan la interoperabilidad y el aprovechamiento de recursos educativos, entre ellos se destacan SCORM, que permite empaquetar y reutilizar contenidos creados en distintas herramientas manteniendo el seguimiento de calificaciones, LTI, que facilita la integración segura de aplicaciones externas como Kahoot, Zoom o Google Drive sin necesidad de desarrollos adicionales, y xAPI, una evolución de SCORM que amplía el registro de experiencias de aprendizaje fuera del propio EVA, por ejemplo en aplicaciones móviles.

Además, Canvas ofrece un alto nivel de personalización, ya que cuenta con un conjunto de APIs abiertas que posibilitan el desarrollo de integraciones a medida con sistemas institucionales, como bibliotecas digitales. Del mismo modo, dispone de un amplio soporte de módulos y extensiones externas, que enriquecen la experiencia educativa y permiten adaptarla a las necesidades específicas de cada institución.

La plataforma ha sido adoptada por instituciones educativas de gran prestigio internacional, algunas de ellas son la Universidad de Harvard, la Universidad de Yale, la Universidad de Stanford, la Universidad de Michigan. [12]

A continuación se presenta el análisis en base a los 6 puntos antes mencionados:

Función	Descripción
Matriculación / desmatriculación	Permite la inscripción automática o manual en cursos, ya sea por códigos, autoinscripción o gestión administrativa. Los estudiantes también pueden abandonar cursos, y los administradores pueden gestionar matrículas masivas.
Seguimiento del avance personal	Provee un panel de progreso donde el estudiante puede ver tareas pendientes, calificaciones, porcentaje de avance y módulos completados Permite ver el progreso de la carrera.
Alojamiento de recursos	Permite que tanto los docentes como estudiantes suban archivos, enlaces, tareas, etc.
Comunicación estudiante - docente	Incluye mensajería interna, comentarios en tareas, anuncios y foros.
Comunicación	Permite mensajes a grupos o toda la institución, integración con



institucional	correo, notificaciones push y foros institucionales.
Disponibilidad móvil	Cuenta con tres aplicaciones móviles para tres perfiles de usuario distintos: docentes, estudiantes y padres.

### 3.4 Análisis comparativo

Se presenta a continuación un análisis comparativo de las distintas aplicaciones analizadas. El propósito de este es identificar las particularidades de cada plataforma y resaltar sus características más significativas.

Función	Moodle	Schoology	Google Classroom	Canvas
Matriculación / desmatriculación	Automática y manual	Automática	Manual	Automática y manual
Seguimiento del avance personal	No seguimiento de cursos ni de carreras	Seguimiento de cursos	Seguimiento de cursos	Seguimiento de cursos y carreras
Alojamiento de recursos	Si	Si	Si	Si
Comunicación estudiante - docente	Chat, foros, anuncios y comentarios en tareas	Chat	Anuncios, comentarios en tareas y correo interno	Chat, foros, anuncios y comentarios en tareas
Comunicación institucional	Anuncios en página principal	Anuncios en página principal	No	Mensajes a grupos, correos y foros
Disponibilidad móvil	Disponible a través de navegador	Disponible a través de app Schoology	Disponible a través de app dedicada	Disponible a través de app dedicada



### 3.5 Conclusiones

El análisis realizado evidencia que los Entornos Virtuales de Aprendizaje constituyen hoy en día un componente esencial de la educación moderna, permitiendo la continuidad pedagógica, la flexibilidad en los procesos y una mayor interacción entre docentes, estudiantes e instituciones.

En términos comparativos se observa que:

- No hay un estándar referido a la automatización de la matriculación y desmatriculación, quedando la implementación a criterio de cada plataforma y/o institución.
- Todas ofrecen acceso desde dispositivo móvil, ya sea desde aplicaciones dedicadas o desde versiones adaptadas del sitio web, lo que reafirma la importancia del acceso permanente a la información.
- Existen diferencias en el seguimiento académico, unas se limitan a mostrar calificaciones de tareas o cursos individuales, otras permiten un control más integral del progreso, incluyendo trayectorias completas.
- Todas proveen la capacidad al docente de subir recursos para ser consumidos por sus estudiantes, mostrando la necesidad absoluta de la inclusión de esta funcionalidad.
- Todas las plataformas analizadas ofrecen recursos y mecanismos para la comunicación estudiante - docente aunque con diferentes niveles de integración.
- La mayoría presenta la capacidad para la comunicación entre instituciones y estudiantes, generalmente a través de anuncios.

Teniendo en cuenta estas similitudes observadas en las distintas plataformas, se pueden concluir ciertas necesidades a la hora de proveer un servicio de esta clase:

**1-** Es necesario contar con la funcionalidad para subir, visualizar y descargar archivos, siendo esta la manera más elemental en la que los estudiantes pueden acceder a conocimientos.

**2-** Se necesita contar con algún medio de comunicación directa entre estudiantes y docentes dentro de la plataforma ya que la guía del docente es fundamental para el proceso de aprendizaje del estudiante al momento de interactuar con los recursos que el docente pone a disposición. La mayoría de las plataformas opta por implementar un sistema de mensajería como medio principal para facilitar esta comunicación.

**3-** Es imperativo contar con acceso desde dispositivos móviles. Esto responde a la situación actual de la sociedad uruguaya, en la que si bien el 94% de los hogares cuenta con acceso a dispositivos capaces de conectarse a internet, menos del 50% generalmente cuenta con



un ordenador, ya sea portátil o de escritorio mientras que alrededor del 87% cuenta con un teléfono celular [13].

Así mismo, de las diferencias entre las plataformas, también se pueden sacar conclusiones, siendo estas las siguientes:

**1-** La matriculación y desmatriculación es algo que puede quedar a criterio de cada plataforma o institución, pero ambos enfoques presentan beneficios y perjuicios, principalmente administrativos, que se deben de tener en cuenta al momento de elegir uno de los dos.

**2-** Si bien no todas las plataformas cuentan con un seguimiento del historial académico del estudiante a largo plazo (lo cual puede responder al nivel educativo al que la plataforma esté orientada), la mayoría cuenta con funcionalidades para seguir el desempeño del estudiante dentro de un curso en particular. Ambos tipos de seguimiento resultan tremadamente útiles para la referencia del estudiante, por lo que es beneficioso(si no casi imperativo) implementarlos.

**3-** Las plataformas que no dan prestaciones para la comunicación entre estudiantes e institución están en la minoría, siendo en este caso solamente google classroom. Esta comunicación es fundamental para que el estudiante esté al tanto de sucesos de importancia en el día a día de la institución y si bien se asume que las plataformas que no cuentan con facilidades para esto han de contar con otros medios de comunicación en los que informar a su cuerpo estudiantil, no se observan perjuicios aparentes al contar con esta funcionalidad dentro de la plataforma, por lo que la falta de esta funcionalidad será considerada una carencia de la plataforma.

En conclusión, en la actualidad los Entornos Virtuales de Aprendizaje no deben ser considerados sólo herramientas de apoyo para el estudiante, sino como componentes estratégicos para su formación, por ende se debe de poner énfasis en que estos pongan al servicio del estudiante funcionalidades beneficiosas para su vida académica, asimismo, dadas la proliferación masiva de los dispositivos móviles, es de vital importancia que las plataformas provean sus servicios a través de los mismos.

En el caso de la plataforma ADÁN, basados en el análisis anteriormente presentado, se consideró incorporar las siguientes funciones:

- Que los profesores puedan subir archivos en diferentes formatos (documentos, imágenes, videos, archivos de audio) y crear tareas espacios de tareas que puedan incluir estos archivos, para que los estudiantes las resuelvan y sean calificados.
- Matriculación manual a cargo de profesores, ya sea individual o en masa. Presentándose también la opción de matriculación manual a cargo del estudiante en los cursos en que se habilite la opción. Esto responde a la idea de cursos irregulares y/u optativos.



- Presentar un historial de calificaciones del estudiante, dentro de un curso específico (referente tanto a tareas específicas, como a desempeño general), así como de las calificaciones generales de los cursos que el estudiante haya cursado anteriormente.
- Incluir la funcionalidad de foros para temas específicos, como foros generales dentro de los cursos, creados y administrados por el o los docentes encargados de cada curso, con la idea de brindar al estudiante un espacio donde resolver dudas, así como intercambiar ideas con sus docentes y pares.
- Proveer también de un sistema de mensajería privada dentro de la plataforma donde un estudiante pueda interactuar directamente con un docente con el fin de resolver situaciones o dudas más particulares a la situación específica del estudiante.
- Poner a disposición de la institución un apartado donde publicar anuncios y noticias, así como de una manera de notificar en masa a estudiantes para tenerlos al tanto de distintas noticias referidas a la vida diaria de la institución que puedan ser de su interés.
- La plataforma será accesible por estudiantes desde dispositivos móviles, contando con todas sus funcionalidades, obedeciendo a la necesidad de brindar esta alternativa en una sociedad donde el método principal de acceso a la web de un número significativo de estudiantes es a través de dispositivos móviles.



## 4. Análisis y Diseño

En esta sección se detalla la especificación de los requerimientos del sistema, abarcando tanto los funcionales como los no funcionales, y se define el alcance del proyecto. Se presentan los tipos de usuarios y las funcionalidades asociadas a cada perfil.

### 4.1 Perfiles de usuario

A continuación se presentan los distintos perfiles de usuario del sistema, junto con un resumen de las funcionalidades asociadas a cada uno:

- **Administrador:** Es el perfil con mayor nivel de control dentro del sistema; administra usuarios y cursos, además de disponer de acceso a la mayoría de las funciones generales. Este usuario accede únicamente a través de la web.
- **Usuario común:** Es el usuario final del sistema. Puede acceder únicamente a los cursos en los que participa. Solo estos usuarios pueden recibir notificaciones y utilizar el sistema de mensajería interna. Dentro de cada curso puede adoptar uno de los dos roles posibles:
  - **Profesor:** Es el responsable de gestionar el contenido y los participantes de sus cursos. Puede matricular, desmatricular y calificar estudiantes. También tiene la capacidad de organizar el curso en secciones que pueden incluir distintos tipos de recursos, como páginas temáticas, materiales, tareas y foros. Todas estas herramientas están disponibles únicamente desde la web.
  - **Estudiante:** Cuenta con funcionalidades básicas para interactuar con el contenido del curso. Puede visualizar las páginas temáticas, descargar materiales, participar en los foros y realizar entregas de tareas. Sus funcionalidades están disponibles tanto en la versión web como en dispositivos móviles.

Esta subdivisión responde a una realidad común en el ámbito terciario, donde un mismo usuario puede desempeñarse como profesor en un curso y como estudiante en otro.

Cabe señalar que la funcionalidad de mensajería está limitada a la comunicación entre profesores y estudiantes pertenecientes a un mismo curso. Es decir, un profesor solo puede contactar a estudiantes matriculados en sus cursos, y un estudiante únicamente puede comunicarse con los profesores de los cursos en los que participa. La comunicación entre profesores o entre estudiantes no es posible.



## 4.2 Requisitos del Sistema

A continuación se describen los requisitos mínimos para el funcionamiento de la plataforma.

### **Requisitos de Hardware**

Para la aplicación web, es necesario contar con una computadora con por lo menos 2 GB de memoria RAM. Es recomendable utilizar sistemas operativos actualizados. Para la aplicación móvil, es necesario un dispositivo con un sistema operativo android, 4 GB de memoria ram y 63 MB de memoria interna disponibles. En ambos casos es necesario una conexión a internet estable.

### **Requisitos de Software**

En el caso de la aplicación web, se necesita contar con un navegador web actualizado para asegurar la compatibilidad con las funcionalidades de la aplicación. Se recomienda utilizar Google Chrome, Mozilla Firefox o Microsoft Edge en sus versiones más recientes. En caso de la aplicación móvil es necesario un sistema operativo android en su versión 10 o superior.

## 4.3 Requerimientos específicos

En esta sección se describen los requerimientos específicos de la plataforma, ofreciendo una visión general de sus funcionalidades y presentando un conjunto de requerimientos no funcionales considerados esenciales para su correcto funcionamiento. Se recomienda consultar el anexo Documento de Requerimientos para una descripción detallada de cada requerimiento funcional.

### **Requerimientos Funcionales**

#### **Funcionalidades comunes entre los tres perfiles:**

- Iniciar sesión.
- Cerrar sesión.
- Editar perfil.
- Cambiar contraseña.
- Recuperar contraseña.
- Buscar cursos.
- Buscar participantes de un curso.
- Descargar materiales.
- Crear hilo en el foro de consultas.
- Publicar mensaje en el foro de consultas.
- Editar mensaje propio publicado en el foro de consultas.
- Eliminar mensaje propio publicado en el foro de consultas.



### **Funcionalidades comunes entre profesor y estudiante:**

- Enviar mensaje privado.
- Leer mensaje privado.

### **Funcionalidades comunes entre profesor y administrador:**

- Matricular estudiante a un curso.
- Matricular estudiantes a un curso de forma masiva.
- Desmatricular estudiante.
- Desmatricular estudiantes de un curso de forma masiva.
- Crear sección dentro de un curso.
- Editar sección dentro de un curso.
- Eliminar sección dentro de un curso.
- Publicar anuncio en cartelera de curso.
- Editar anuncio en cartelera de curso.
- Eliminar anuncio en cartelera de curso.
- Crear página temática.
- Editar página temática.
- Eliminar página temática.
- Subir material a un curso.
- Eliminar material de un curso.
- Crear foro de consultas.
- Editar foro de consultas.
- Eliminar foro de consultas.
- Editar cualquier mensaje del foro de consultas.
- Eliminar cualquier mensaje del foro de consultas.
- Eliminar hilo de un foro.
- Crear tarea con fecha límite.
- Editar tarea.
- Eliminar tarea.
- Enviar aviso masivo de proximidad de plazo límite de entrega.
- Descargar entrega de estudiante.
- Descargar entregas de forma masiva.
- Calificar entrega.
- Calificar entregas de forma masiva.
- Publicar calificación de forma individual.
- Publicar calificaciones de forma masiva.

### **Funcionalidades del administrador:**

- Crear usuario individual.
- Crear usuarios de forma masiva.
- Buscar usuarios dentro del sistema.
- Visualizar resumen de actividad de usuario.
- Eliminar usuario.
- Crear curso individual.
- Crear cursos de forma masiva.
- Editar curso.



- Eliminar curso.
- Asignar profesor a un curso.
- Desasignar profesor a un curso.

#### **Funcionalidades del estudiante:**

- Subir entrega de una tarea.
- Visualizar calificaciones publicadas por el profesor.

### **Requerimientos No Funcionales**

#### **Usabilidad**

Se hizo hincapié en la facilidad de uso de la interfaz de usuario, prefiriendo fuentes claras y de buen tamaño, colores con buen contraste pero no agresivos a la vista y botones claros. Logrando una interfaz clara e intuitiva para el usuario final.

#### **Portabilidad**

La plataforma es accesible desde los navegadores web de mayor uso, así como en dispositivos tanto de escritorio como portátiles. Esto garantiza que la aplicación tenga gran disponibilidad para el usuario.

#### **Escalabilidad**

El sistema es fácilmente escalable, tanto vertical como horizontalmente, logrando poder funcionar bajo distintos niveles de carga de usuarios, así como pudiéndose agregar funcionalidades a futuro sin mayor complicación.

#### **Seguridad**

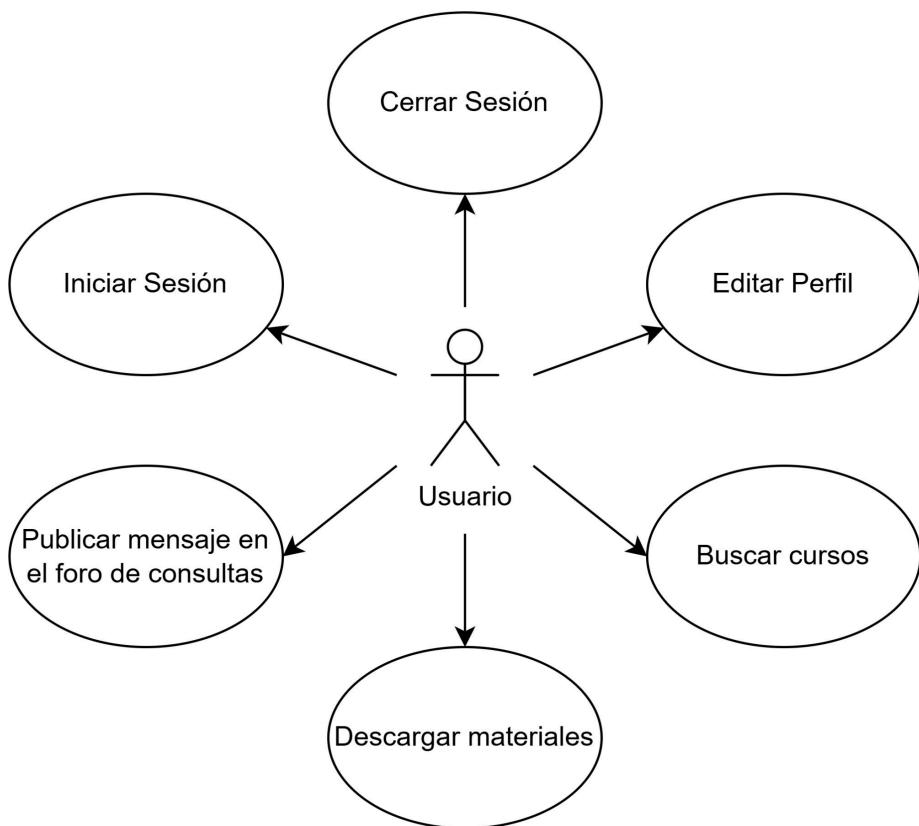
Las contraseñas son encriptadas de manera segura utilizando algoritmos como Bcrypt, asimismo, las validaciones se hacen a nivel del backend garantizando que ningún usuario pueda acceder a funcionalidades o datos a los que no deba poder acceder.



## 4.4 Modelo de Casos de Uso

En esta sección se muestra un resumen de los casos de uso críticos del sistema. Se consideran críticos aquellos casos de uso que influyen directamente en el modelo de la arquitectura del sistema o que resultan indispensables para la ejecución de otros casos de uso. Para información más detallada de cada uno, se recomienda consultar el Documento de Casos de Uso.

### Casos de uso comunes a todos los tipos de usuario



*Imagen 1: Diagrama de casos de uso comunes para todos los usuarios*

**Iniciar sesión:** Permite a cualquier usuario registrado acceder a la plataforma mediante el ingreso de sus credenciales (correo electrónico y contraseña).

**Cerrar sesión:** Permite a un usuario autenticado finalizar su sesión y salir de la plataforma de forma segura.

**Editar perfil:** El usuario puede modificar información personal (nombres, apellidos, foto de perfil y fecha de nacimiento).

**Buscar cursos:** El sistema muestra los cursos disponibles y permite filtrarlos por nombre del curso, código, año, turno y nombre de profesores asociados.

**Descargar materiales:** Permite a los usuarios obtener en su dispositivo los archivos asociados a un curso.



**Publicar mensajes en el foro de consultas de un curso:** Los usuarios pueden publicar mensajes en los distintos foros de consulta de los cursos. Estos mensajes pueden crear un hilo nuevo si el usuario así lo desea, o pueden formar parte de un hilo existente.

### Casos de uso para usuario Administrador

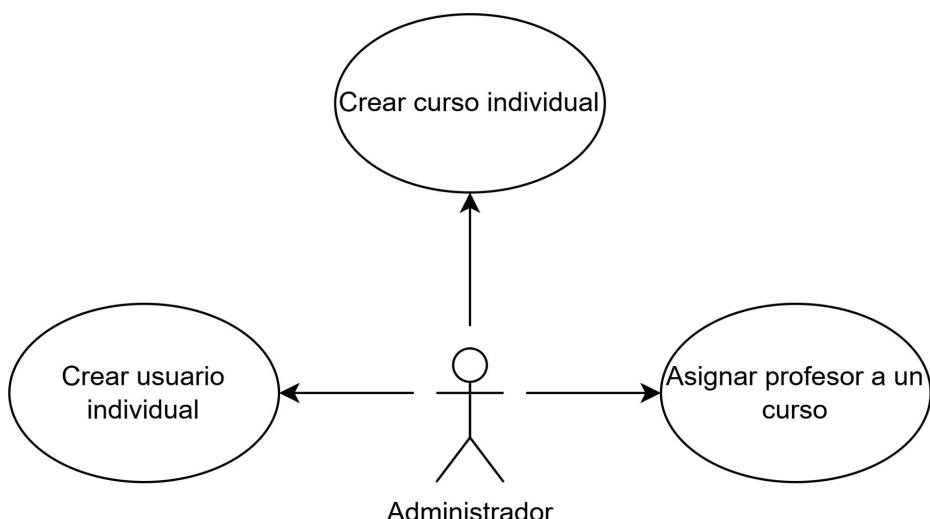


Imagen 2: Diagrama de casos de uso del perfil Administrador

**Crear usuario individual:** Posibilidad de registrar manualmente a un nuevo usuario en el sistema. Los datos mínimos para el registro varían levemente en base al tipo de usuario a registrar:

- Estudiante y profesor: cédula, nombre, apellido y correo electrónico.
- Administrador: correo electrónico.

Al ser creado un usuario, su contraseña será autogenerada y deberá ser cambiada por el usuario a futuro para su propia conveniencia.

**Crear curso individual:** Posibilidad de generar un curso nuevo con datos básicos ingresados manualmente. Los datos mínimos requeridos son nombre del curso, código, año y turno, y al dar de alta el curso éste se crea automáticamente con una sección inicial denominada “Cartelera de Novedades”.

**Asignar profesor a un curso:** Permite asignarle a un usuario registrado en el sistema el rol de profesor dentro de un curso existente.



## Casos de uso para usuario Profesor

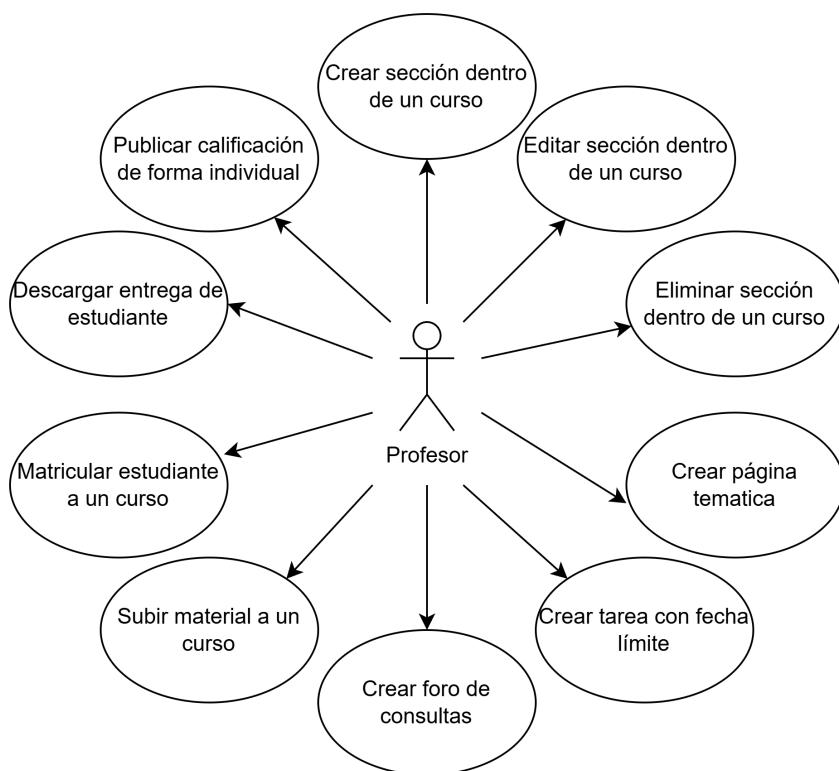


Imagen 3: Diagrama de casos de uso del perfil Profesor

**Crear sección dentro de un curso:** Permite a un profesor crear una nueva sección dentro de un curso específico. La sección es un apartado que contiene un título y puede contener varios recursos (foros, materiales, tareas o páginas temáticas). Pueden existir múltiples secciones dentro de un curso. La cartelera de novedades es una sección especial, siendo esta creada junto con el curso y apareciendo al tope de la página del curso.

**Editar sección dentro de un curso:** Permite modificar los datos de título y visibilidad de la sección.

**Eliminar sección dentro de un curso:** Permite eliminar una sección específica dentro de un curso. Al eliminar la sección, todos los recursos dentro de la misma también son eliminados.

**Crear página temática:** El profesor podrá crear páginas de contenido temático dentro de una sección específica de un curso a través de un editor de texto enriquecido. La página de contenido temático tendrá un título y podrá contener imágenes, texto y links. Una vez finalizada la edición, la página generada se guardará como HTML.

**Crear tarea con fecha límite:** Permite la creación de un recurso de tipo tarea ingresando nombre, descripción, fecha de inicio y fecha de fin. La tarea habilita por un período de tiempo específico (delimitado por fecha de inicio y fecha de fin) la posibilidad de que los estudiantes suban un único archivo en formato zip con el contenido requerido en la propuesta de la tarea.



**Crear foro de consultas:** El profesor puede crear foros de consulta en las distintas secciones de un curso. Una sección puede tener más de un foro. En dichos foros se crean hilos de discusión cuando un mensaje sea publicado que no responda a un mensaje anterior.

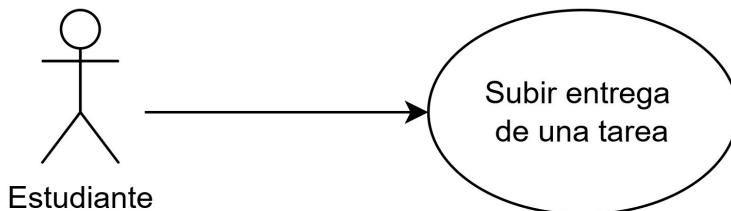
**Subir material a un curso:** Permite cargar archivos a las secciones de un curso. El tipo de archivo a subir es libre.

**Matricular estudiante a un curso:** Permite inscribir manualmente a un usuario registrado en el sistema como estudiante en un curso. Al matricular un estudiante a un curso se genera una calificación final vacía asociada al estudiante y curso específicos.

**Descargar entrega de estudiante:** El profesor, luego de crear una tarea tiene acceso al contenido del directorio en el que se guardan los archivos que entregan los estudiantes, pudiendo ver y corregir las entregas cuando desee. La descarga de tareas es de forma individual.

**Publicar calificación de forma individual:** Permite subir notas finales del curso para un estudiante matriculado. El tipo de calificación, ya sea numérica, en forma de letra (A,B,C, etc.) o de otro tipo que utilice caracteres alfanuméricos queda a criterio de cada profesor.

### **Casos de uso para usuario Estudiante**



*Imagen 4: Diagrama de casos de uso del perfil Estudiante*

**Subir entrega de una tarea:** El estudiante puede realizar la entrega de tareas dentro de un curso accediendo al recurso correspondiente dentro del plazo permitido, subiendo un archivo .zip para luego ser calificado por el profesor.



## 4.5 Modelo de dominio

A continuación se presenta el modelo de dominio del proyecto. En él se describen las entidades, sus atributos y las relaciones que las vinculan, ofreciendo una visión estructural del funcionamiento general de la realidad correspondiente a la plataforma a desarrollar.

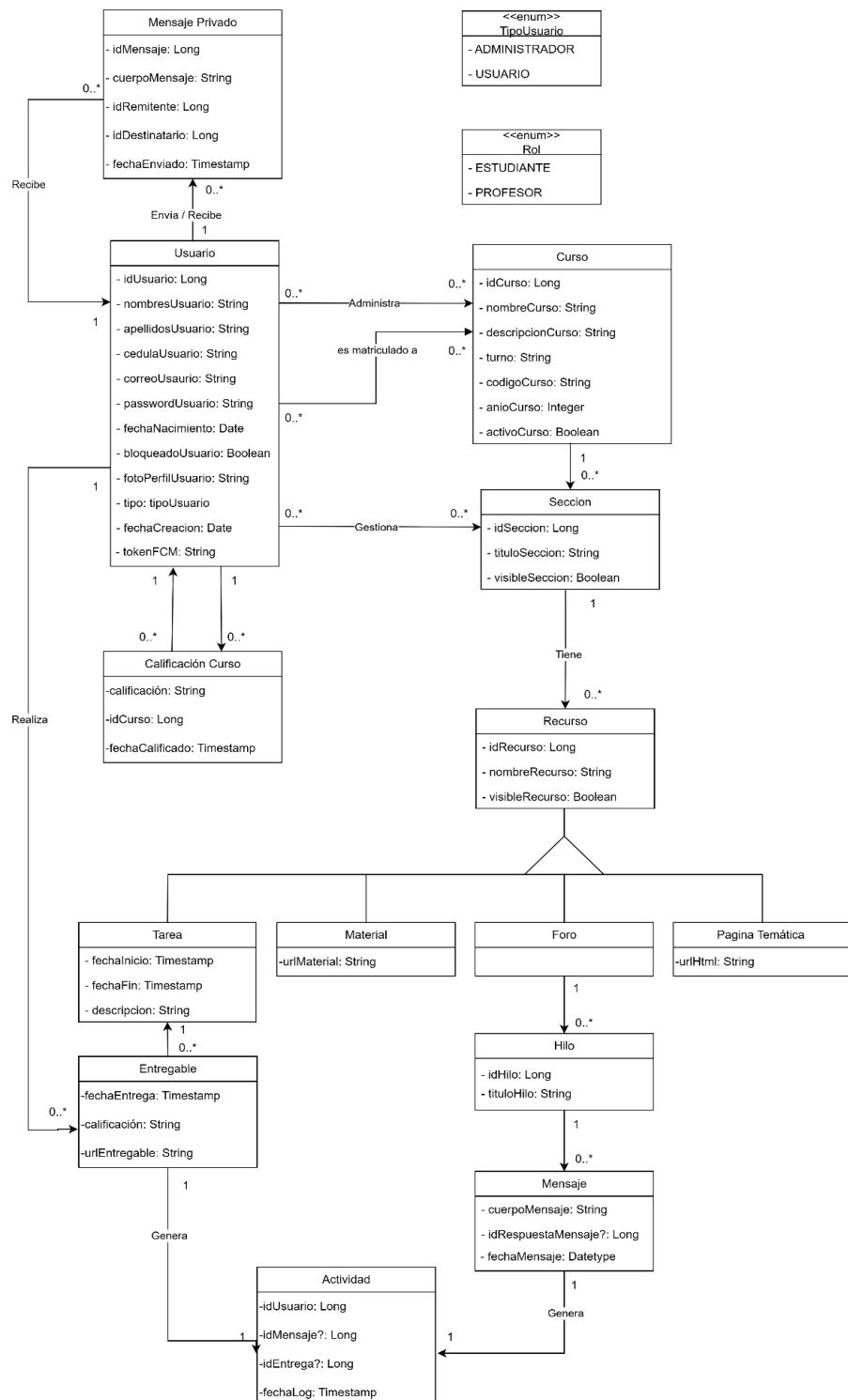


Imagen 5: Diagrama de modelo de dominio



El diagrama del modelo de dominio representa un sistema de gestión de cursos virtuales, orientado a la administración de usuarios, la organización de cursos y secciones, la entrega de tareas, la interacción en foros y el envío de mensajes privados.

En la parte central se encuentra la clase **Usuario**, que modela a todas las personas que interactúan con el sistema. Incluye atributos como nombres, apellidos, cédula, correo electrónico, contraseña, fecha de nacimiento, tipo de usuario (Administrador, Usuario), rol (Estudiante, Profesor), estado (bloqueado / desbloqueado) y un token de autenticación para la gestión de notificaciones.

Los administradores pueden crear, matricular y también gestionar los cursos. La clase **Curso** contiene información básica como nombre, descripción, código y estado (activo o inactivo).

Los cursos se dividen en **Secciones**, que poseen un título y visibilidad (habilitada / deshabilitada). Cada sección está compuesta por uno o varios **Recursos**, los cuales pueden adoptar distintas formas:

- **Tarea**: actividad práctica con fecha de inicio y fin, que los estudiantes deben resolver.
- **Material**: archivo disponible mediante una URL.
- **Foro**: espacio de discusión donde los usuarios intercambian mensajes.
- **Página Temática**: recurso con contenido en formato HTML estático.

Cada **Tarea** permite que los usuarios suban un **Entregable**, que contiene la fecha de entrega, calificación y un enlace al archivo entregado. A su vez, cada entrega se registra en la entidad **Actividad**, que refleja las acciones del usuario en el sistema (mensajes publicados en foros y entregas realizadas).

Los cursos también registran una **Calificación Curso**, que indica la nota final obtenida por un estudiante en una materia específica.

En paralelo, el sistema gestiona la comunicación entre usuarios:

- A través de **Mensajes Privados**, que son enviados directamente de un estudiante a un profesor o viceversa, con su fecha de envío y contenido.
- **Foros**: los mensajes en este espacio están organizados mediante la entidad **Hilo**, que agrupa publicaciones bajo un tema central. Cada **Mensaje** en un foro contiene cuerpo, fecha y puede responder a otro mensaje dentro del mismo hilo.

Este modelo busca garantizar una gestión adecuada de los usuarios, cursos, entregas y recursos educativos, favoreciendo tanto la comunicación como la evaluación dentro del entorno virtual de aprendizaje.



## 4.6 Arquitectura del sistema

A continuación se presenta la arquitectura del sistema, describiendo sus componentes principales y la forma en que interactúan entre sí para soportar los distintos procesos de la aplicación.

### Modelo de implementación

En este proyecto se adoptó una arquitectura basada en tres capas principales: la capa de presentación, la capa de negocio y servicios, y la capa de acceso a datos. Este enfoque permite una separación clara de responsabilidades, facilitando el mantenimiento, la escalabilidad y la evolución del sistema.

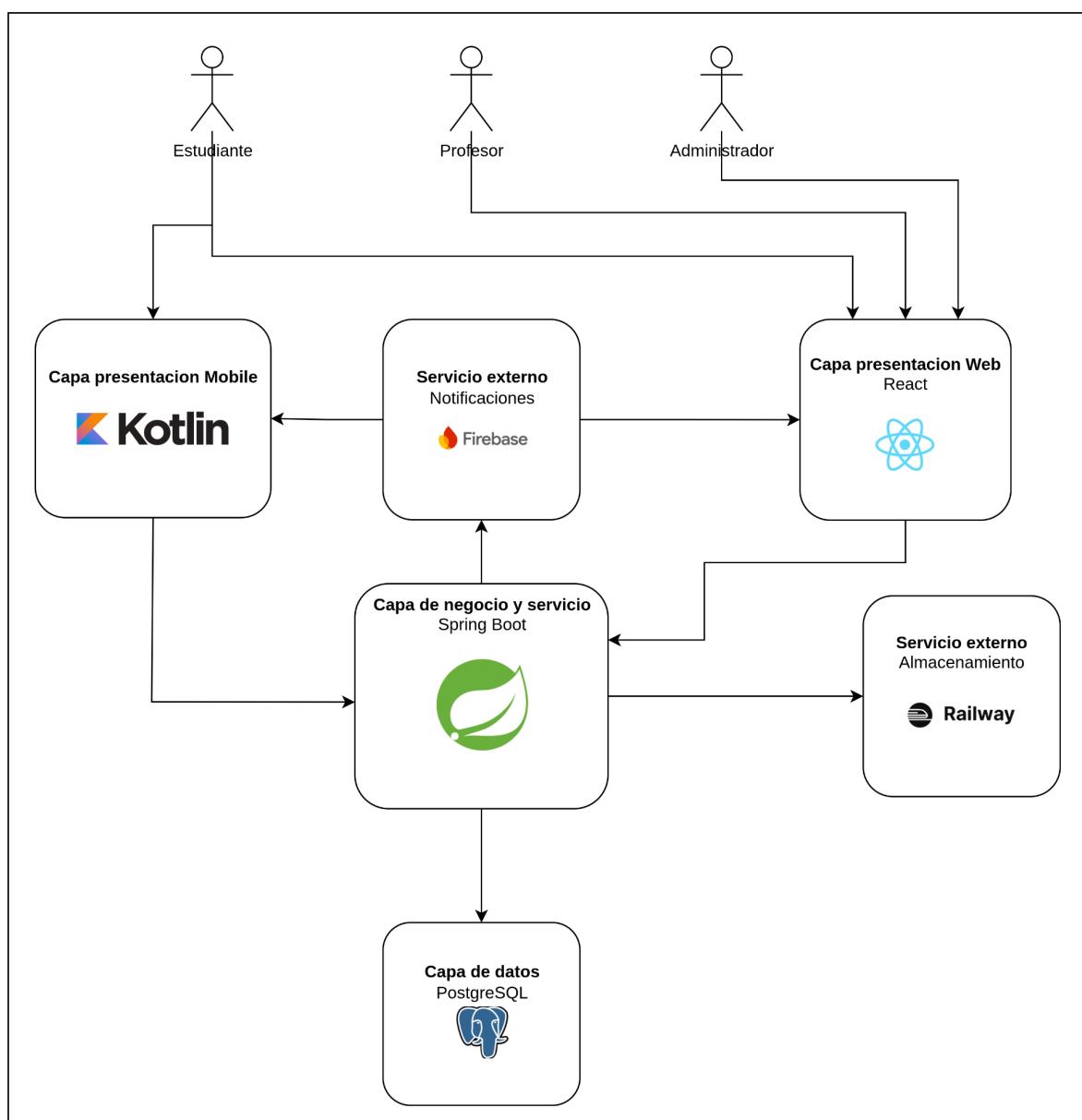


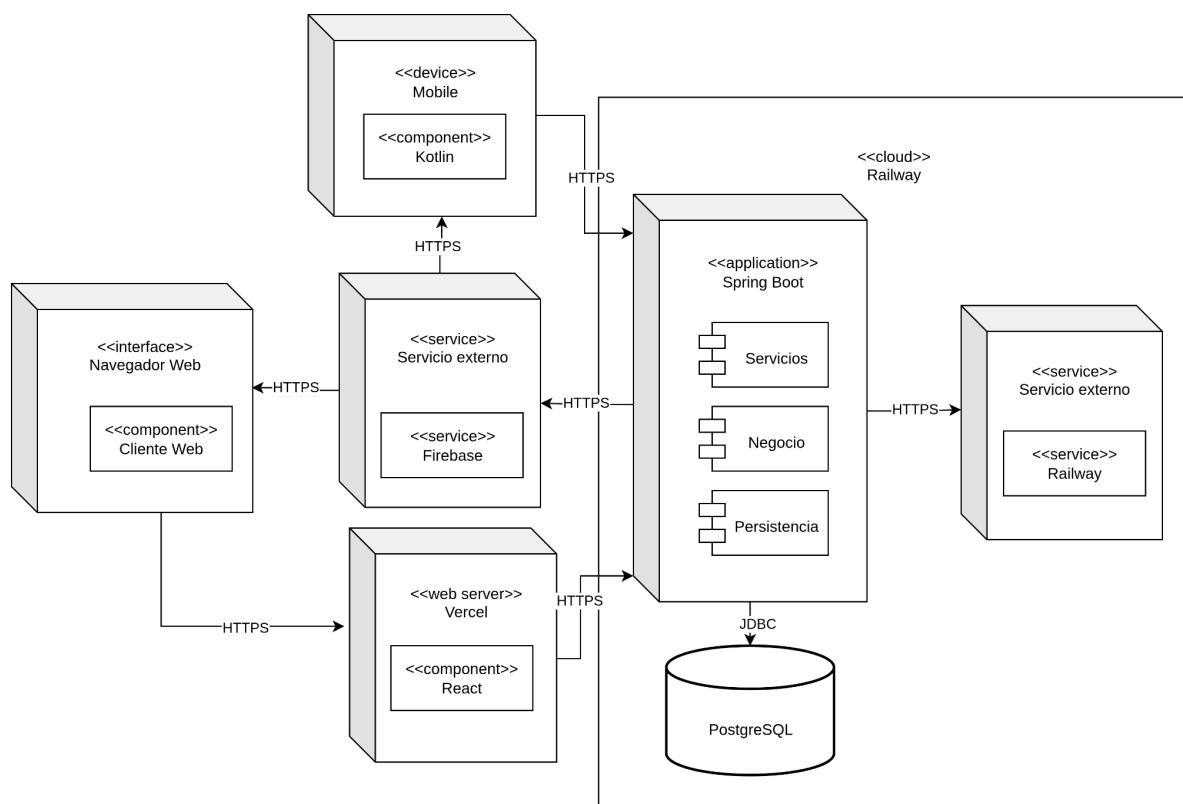
Imagen 6: Diagrama de modelo de implementación



Con este enfoque, se logra separar las responsabilidades del sistema en capas que brindan servicios a la capa que tienen por encima; La capa de datos provee el acceso a los datos de la base de datos a la capa de negocio y servicios; esta maneja la lógica de negocio y provee servicios a la capa de presentación; esta última comprende las interfaces de usuario tanto web como mobile y las proveen al usuario final. Finalmente, se utilizaron los servicios externos de Railway (para el alojamiento de imágenes y archivos) y Firebase (para el manejo de notificaciones push).

## Modelo de Distribución

En esta sección se presenta la distribución física de los componentes del sistema, así como la forma en que estos se comunican e interactúan dentro de la infraestructura definida.



*Imagen 7: Diagrama de modelo de distribución*

El backend, desarrollado en Spring Boot, se aloja en la plataforma Railway y se organiza bajo un esquema de tres capas principales:

- Servicios (API): expone los endpoints que son consumidos por las interfaces web y móvil.
- Negocio: contiene la lógica central de la aplicación y coordina la interacción entre los distintos módulos.
- Persistencia: gestiona la comunicación con la base de datos a través de JPA/Hibernate sobre JDBC, garantizando la integridad de los datos.

El sistema cuenta con dos interfaces de usuario: la web, desarrollada en React y desplegada en Vercel, y la móvil, desarrollada en Kotlin. Ambas aplicaciones consumen los



servicios REST expuestos por el backend, lo que garantiza una experiencia coherente entre plataformas y una comunicación unificada con la lógica del sistema. Toda la información del sistema se almacena en una base de datos relacional PostgreSQL, también desplegada en Railway. Esta plataforma se utiliza además para el almacenamiento de archivos (por ejemplo, materiales de cursos y tareas enviadas por los estudiantes). Para la funcionalidad de notificaciones y mensajería, se emplea Firebase, que permite enviar mensajes en tiempo real a los clientes web y móviles. La comunicación entre todos los componentes, frontend, backend, base de datos y servicios externos, se realiza de forma segura y encriptada mediante el protocolo HTTPS, garantizando la confidencialidad, integridad y disponibilidad de la aplicación en todos los entornos.

## 4.7 Diseño del sistema

Esta sección describe las decisiones adoptadas para el diseño del sistema, abarcando la estructura de clases y los métodos que lo conforman. Para más detalle, se recomienda consultar el anexo Documento de Diseño.

### Diagrama de clases

En esta sección se presenta el diagrama que modela las clases principales del backend y las relaciones entre ellas. Este diseño constituye la base estructural del sistema, ya que define los servicios y la lógica necesarios para soportar los casos de uso críticos y asegurar un funcionamiento coherente, extensible y mantenable.

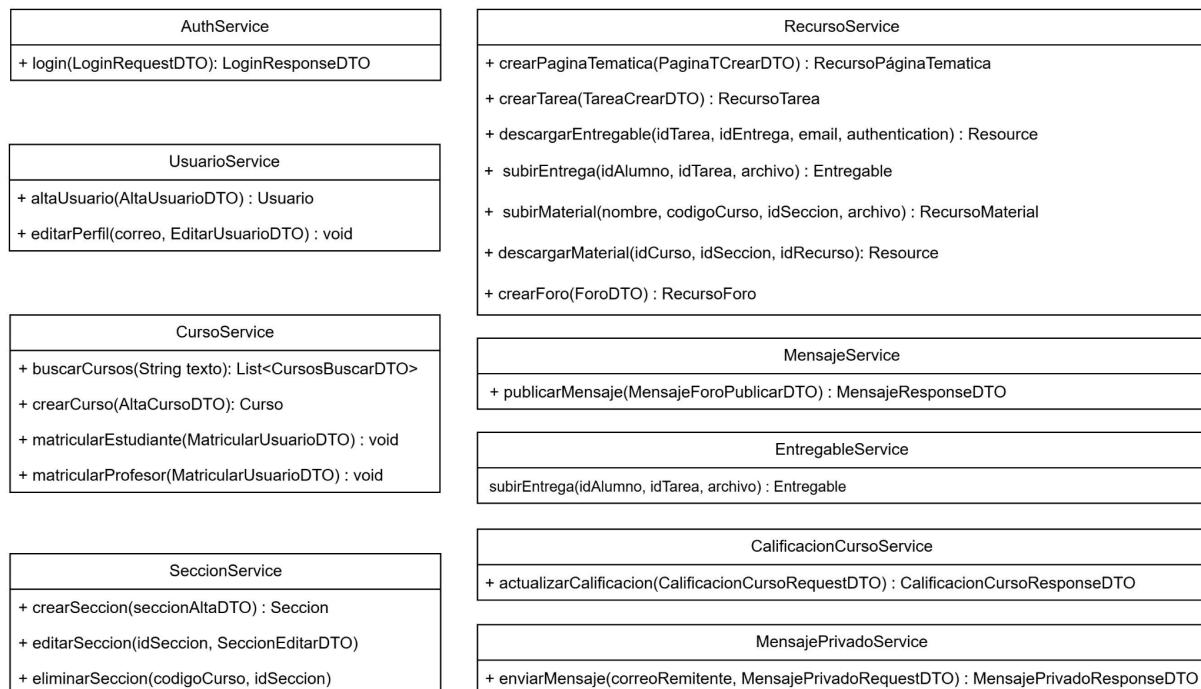


Imagen 8: Diagrama de clases



Estas clases de tipo Service manejan la lógica de negocio, la encapsulan y la proveen a las clases de tipo Controller, siendo estas últimas las encargadas de recibir y responder a las solicitudes del usuario enviadas a través del frontend de la aplicación. Asimismo, son los Service los que tienen comunicación con las clases Repository, las cuales son las que gestionan el acceso a la capa de datos.

Este es un resumen de la funcionalidad de cada clase ilustrada en el diagrama:

**AuthService:** Responsable de la autenticación de los usuarios, brindando funcionalidades como el inicio de sesión.

**UsuarioService:** Gestiona todas las operaciones relacionadas con los usuarios, incluyendo alta, baja, modificación y demás acciones asociadas.

**CursoService:** Administra los cursos del sistema, permitiendo su creación, eliminación, modificación y búsqueda. También gestiona la matriculación de los estudiantes en los distintos cursos.

**SeccionService:** Administra las secciones de los cursos, permitiendo su creación, eliminación y modificación.

**RecursoService:** Gestiona los recursos asociados a cada curso, como materiales, foros, tareas entregables y páginas temáticas de sus secciones.

**MensajeService:** Maneja el sistema de mensajes dentro de los foros, incluyendo la publicación, edición y eliminación de los mismos.

**EntregableService:** Administra los entregables asociados a las tareas de los cursos.

**CalificacionCursoService:** Gestiona las calificaciones finales de los estudiantes, habilitando a los profesores a generarlas y modificarlas.

**MensajePrivadoService:** Gestiona la mensajería interna entre usuarios, incluyendo el envío, recepción y administración de mensajes privados.



## Diagrama de Secuencia de casos de uso Críticos

En esta sección se incluyen, a modo de ejemplo, los diagramas de secuencia correspondientes a algunos de los casos de uso más relevantes del sistema. Para consultar la totalidad de los diagramas elaborados se puede acudir al documento anexo “Documento de Diseño”. Estos diagramas permiten visualizar cómo se intercambia la información entre los distintos componentes, objetos y métodos involucrados en cada interacción descrita.

### Crear curso individual

crearCurso(AltaCursoDTO): Curso

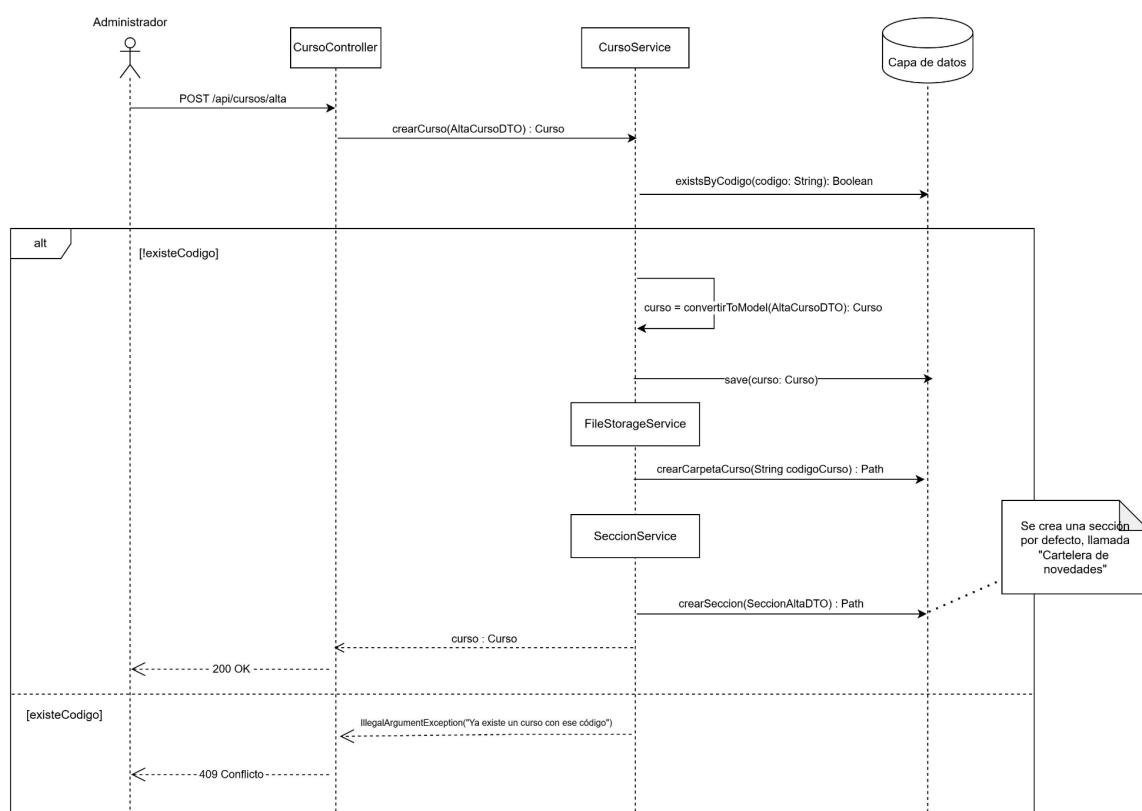


Imagen 9: Diagrama de secuencia CU Crear curso individual



## Descargar entrega de estudiante

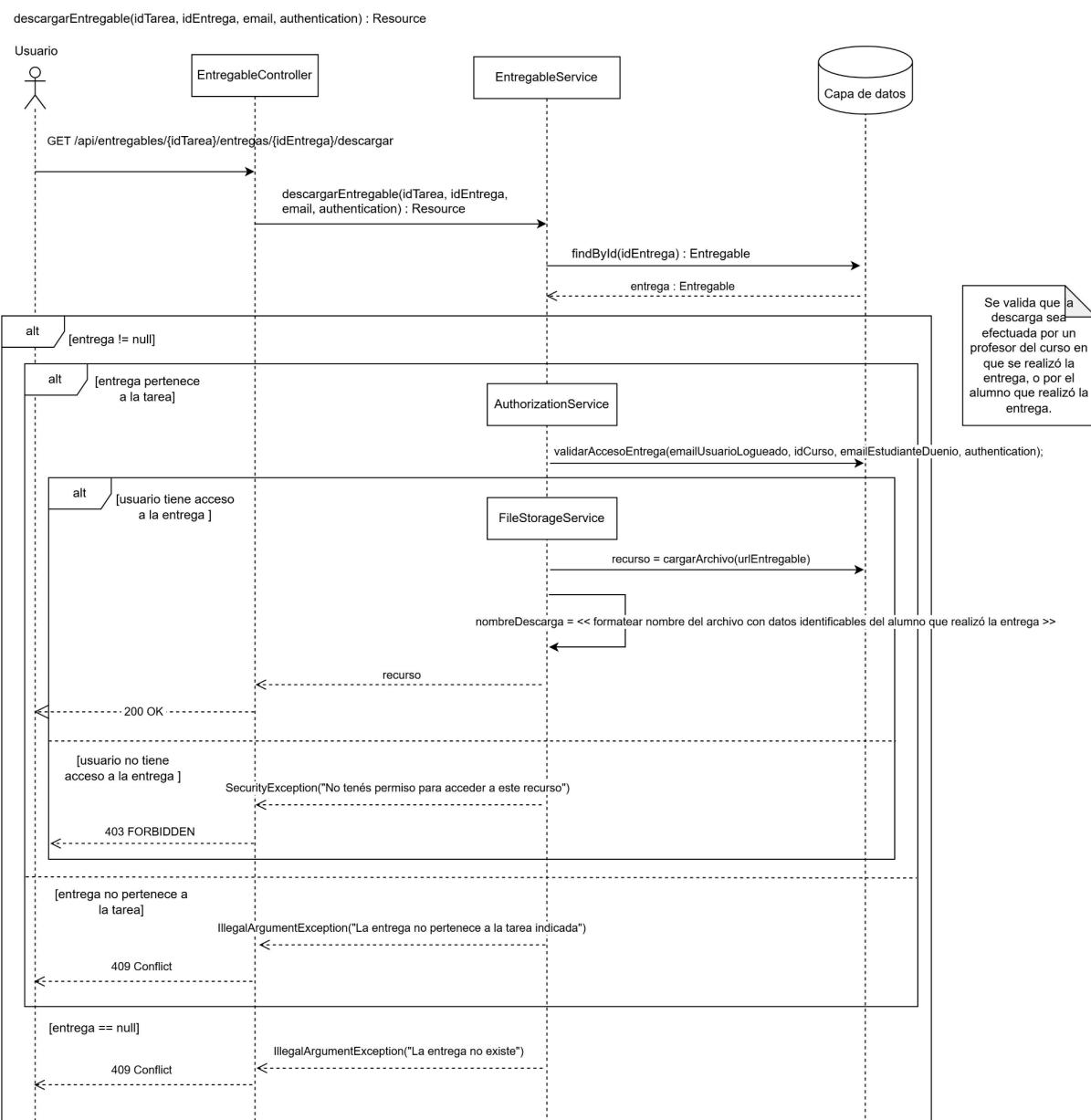


Imagen 10: Diagrama de secuencia CU Descargar entrega de estudiante



## Crear página temática

crearPaginaTematica(PaginaTCrearDTO) : RecursoPáginaTematica

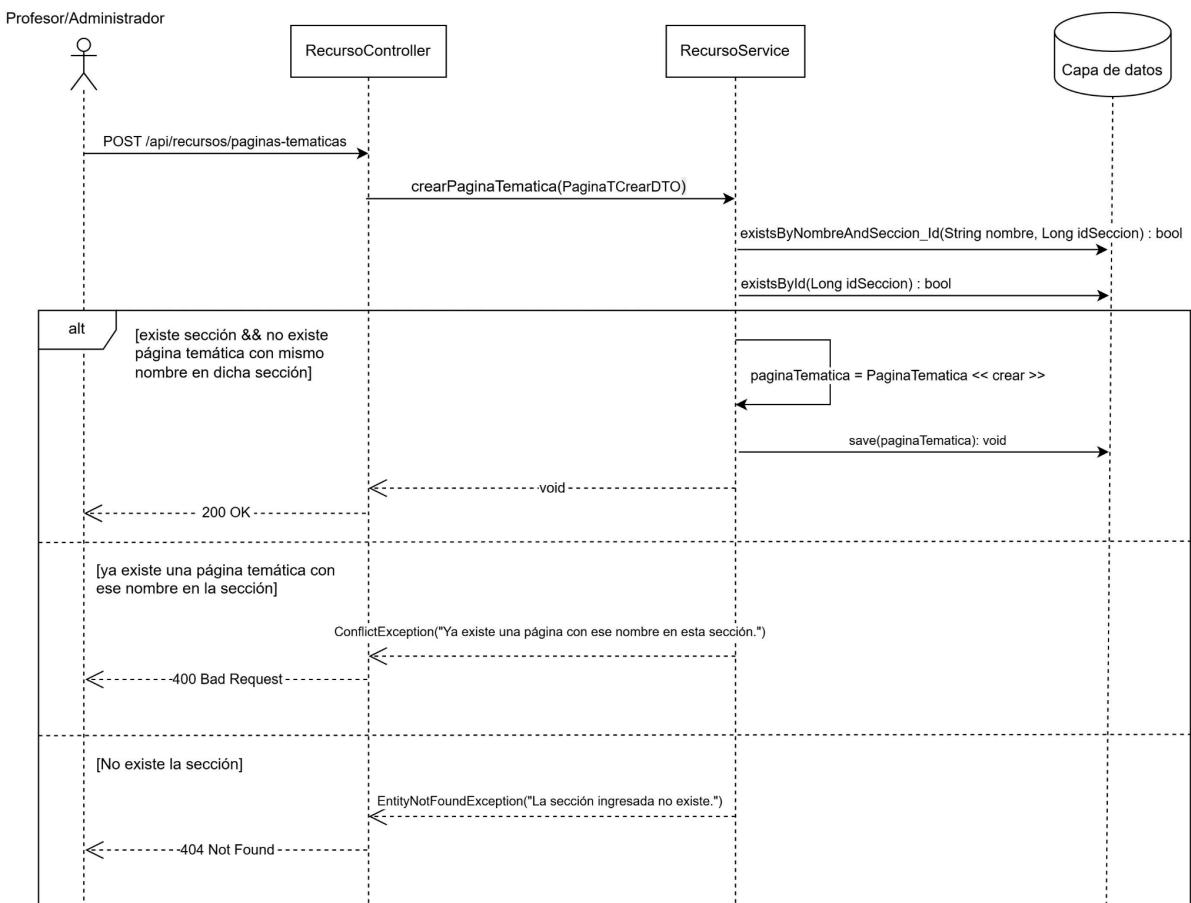


Imagen 11: Diagrama de secuencia CU Crear página temática



## Matricular estudiante a un curso

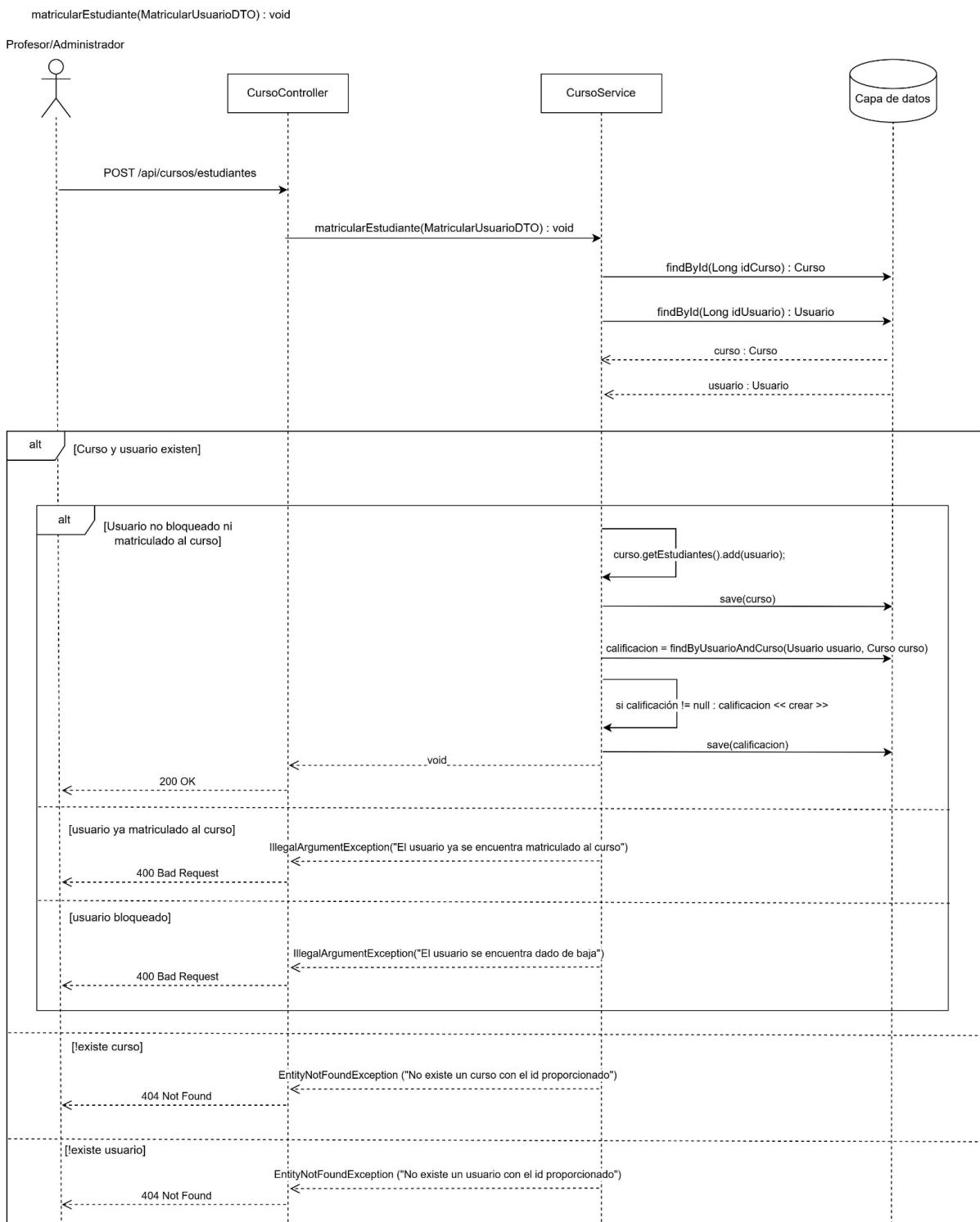


Imagen 12: Diagrama de secuencia CU Matricular estudiante a un curso



## Crear usuario individual

altaUsuario(AltaUsuarioDTO): Usuario

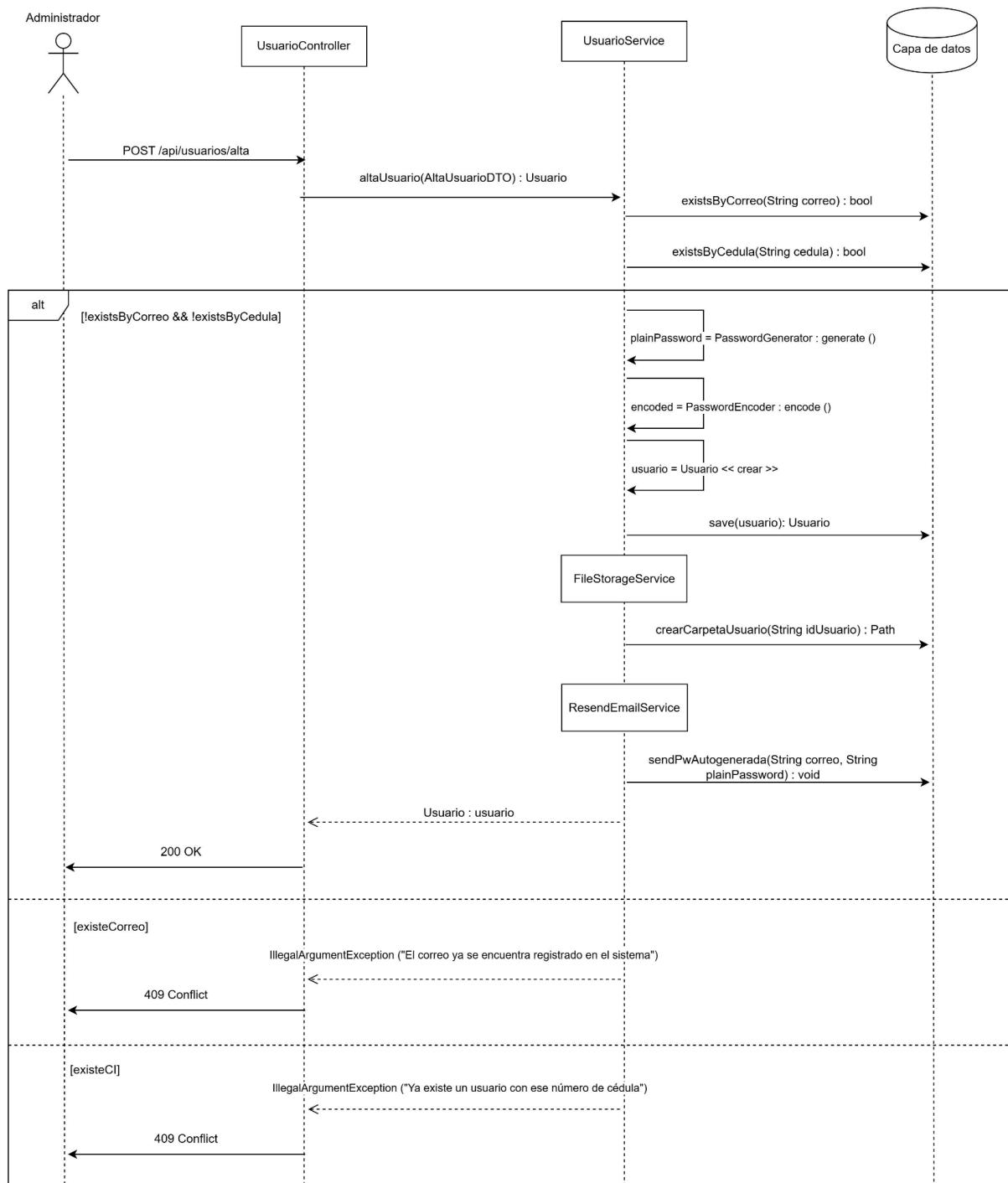


Imagen 13: Diagrama de secuencia CU Crear usuario individual



## 5. Gestión del proyecto

En este apartado se describe detalladamente el proceso de trabajo llevado a cabo durante el transcurso del proyecto. Se detalla la metodología del trabajo empleada, la planificación del trabajo y la distribución de los roles en el equipo.

### 5.1 Metodología de trabajo

El desarrollo se llevó a cabo en el transcurso de cuatro meses, realizándose en cada uno de ellos un conjunto de tareas específicas.

En el primer mes, se llevó a cabo la distribución de roles en el equipo. Seguidamente se realizaron en paralelo la investigación del contexto en el que se desarrolla el sistema y las soluciones ya disponibles en el mercado, junto con la investigación de las tecnologías a utilizar y pruebas de concepto de las mismas, las cuales fueron plasmados en el documento de Pruebas de Concepto.

El segundo mes se utilizó para el análisis y diseño de la aplicación, dividiendo el mes en dos mitades. La primera fue dedicada a definir el alcance del proyecto, junto con los requerimientos, para luego escribir los casos de uso críticos y realizar el modelado de dominio. En la segunda mitad del mes se produjeron los documentos de Arquitectura, Diseño y Modelado de datos, y se planificó la etapa de testing, la cual se concretó posteriormente.

En el tercer mes se llevó a cabo la implementación de la solución utilizando los documentos y planificación producidos en los dos meses anteriores. Este mes también se dividió en dos mitades, siendo la primera dedicada al primer sprint, en el cual se desarrollaron las funcionalidades consideradas por el equipo como críticas para el funcionamiento de la plataforma; y la segunda mitad fue dedicada al segundo sprint, en el que se implementaron el resto de las funcionalidades. También se dio inicio a la etapa de validación y testing.

Finalmente, en el cuarto mes se completó la validación y testing del sistema, así como la preparación del informe final y la presentación.

Para la coordinación y el trabajo se hicieron uso de las siguientes herramientas:

**Whatsapp:** Para todo lo referente a la coordinación del equipo.

**Discord:** Para realizar las reuniones previamente coordinadas, ya fueran semanales o otras que el equipo considerara necesarias.

**Google Docs:** Para el reparto de las tareas en el equipo y la producción de los documentos.

**Draw.io:** Para la producción de esquemas y diagramas para su uso en los distintos documentos.

**Figma:** Para el desarrollo de mockups de la interfaz de usuario.

**Google Drive:** Para el compartir de materiales.

**Github:** Para la gestión de recursos y el versionado.



## 5.2 Distribución de Roles

En esta sección se detalla la distribución de los roles en el equipo de trabajo. El reparto de roles se basó en preferencias individuales, así como en experiencia previa con el trabajo a realizar, teniendo cada integrante por lo menos dos roles. Estos roles son los siguientes:

**Desarrollador Backend:** Encargado de desarrollar el backend o la parte de la plataforma que maneja la lógica de negocio, los datos y la comunicación con servicios externos para y que brinda funcionalidad al frontend.

**Desarrollador Frontend:** Encargado de desarrollar el frontend, la parte de la solución que interactúa directamente con el usuario final, comprendiendo esta la interfaz del usuario o UI además de lógica ligera en el manejo de los datos y el consumo de servicios provistos por el backend.

**Desarrollador Mobile:** Encargado de desarrollar la aplicación Mobile. Esta aplicación consume los servicios brindados por el backend para proveer al usuario de tipo estudiante las funcionalidades del mismo en dispositivos móviles.

**DevOps:** Encargado de la automatización y la integración entre el desarrollo, el despliegue y las pruebas. Es el encargado de asegurarse que la aplicación funcione correctamente en el entorno esperado.

**Tester:** Encargado de detectar problemas, errores y aspectos a cambiar para un mejor funcionamiento del sistema; esto lo hace a través de la ejecución de casos de prueba para las distintas funcionalidades de la aplicación.

**Coordinador:** Encargado de gestionar el tiempo de desarrollo y tomar decisiones respecto a que parte del desarrollo priorizar y cuando. También se encarga de verificar los documentos y organizar el trabajo del equipo.

Estos roles fueron asignados de la siguiente manera:

**Backend:**

Iván Castro  
Bruno Menendez  
Maia Oldak

**Frontend:**

Marcelo Bravo  
Manuel Sainz  
Eduardo Alvarez  
Maia Oldak

**Mobile:**

Manuel Sainz  
Eduardo Alvarez



**Testing:**

Marcelo Bravo  
Bruno Menendez

**DevOps:**

Iván Castro

**Coordinación:**

Bruno Menendez  
Eduardo Alvarez

## 5.3 Planificación

En esta sección se detalla la planificación del trabajo realizado en el tercer mes del proyecto. El trabajo fue dividido en dos sprints de dos semanas. En el primer sprint se implementaron las funcionalidades críticas de la plataforma, mientras que en el segundo sprint se implementó el resto de las funcionalidades. Este enfoque dio como resultado que al final del primer sprint se tuviera un producto con funcionalidades básicas que le diera al equipo una base sobre la que se pudiera re evaluar el producto y en caso de ser necesario, realizar ajustes al diseño y sobre el que se pudiera iterar en el segundo sprint.

El reparto de las funcionalidades entre ambos fue el siguiente:

### Primer Sprint

#### Semana 1:

- Iniciar sesión.
- Cerrar sesión.
- Crear usuario individual.
- Editar perfil.
- Crear curso individual.
- Asignar profesor a un curso.
- Matricular estudiante a un curso.
- Crear sección dentro de un curso.
- Crear tarea con fecha límite.
- Subir entrega de una tarea.
- Subir material a un curso.

#### Semana 2:

- Buscar cursos.
- Descargar entrega de estudiante.
- Descargar materiales.
- Crear página temática.
- Publicar calificación de forma individual.
- Crear foro de consultas.
- Crear hilo en el foro de consultas.
- Enviar mensaje privado.



- Publicar mensaje en el foro de consultas.
- Eliminar sección dentro de un curso.
- Editar sección dentro de un curso.

## **Segundo Sprint**

### **Semana 3:**

- Cambiar contraseña.
- Recuperar contraseña.
- Leer mensaje privado.
- Editar mensaje propio publicado en el foro de consultas.
- Editar cualquier mensaje del foro de consultas.
- Eliminar usuario.
- Crear usuarios de forma masiva.
- Buscar usuarios dentro del sistema.
- Visualizar resumen de actividad de usuario.
- Crear cursos de forma masiva.
- Eliminar curso.
- Editar curso.
- Desasignar profesor a un curso.
- Buscar participantes de un curso.
- Matricular estudiantes a un curso de forma masiva.
- Desmatricular estudiante.
- Desmatricular estudiantes de un curso de forma masiva.
- Eliminar página temática.

### **Semana 4:**

- Calificar entrega.
- Calificar entregas de forma masiva.
- Eliminar material de un curso.
- Publicar anuncio en cartelera de curso.
- Editar anuncio en cartelera de curso.
- Eliminar anuncio en cartelera de curso.
- Eliminar mensaje propio publicado en el foro de consultas.
- Eliminar cualquier mensaje del foro de consultas.
- Enviar aviso masivo de proximidad de plazo límite de entrega.
- Publicar calificaciones de forma masiva.
- Eliminar tarea.
- Editar tarea.
- Eliminar foro de consultas.
- Editar foro de consultas.
- Eliminar hilo de un foro.
- Visualizar calificaciones publicadas por el profesor.
- Editar página temática.
- Descargar entregas de forma masiva.



## 5.4 Horas dedicadas

La siguiente tabla muestra el estimativo inicial de horas a dedicar a cada tarea del proyecto, junto con las horas reales dedicadas a las mismas.

Actividad	Estimación (horas)	Efectivas(horas)	Diferencia
Monitoreos	6	7.5	+1.5
Documento Estado de Arte	30	25	-5
Documento de Alcance	10	10	0
Documento de Requerimientos	30	35	+5
Documento Casos de uso	70	80	+10
Documento de Arquitectura	10	10	0
Documento de Diseño	60	75	+15
Glosario	5	3	-2
Pruebas de Concepto	30	50	+20
Modelo de Dominio	30	25	-5
Modelo de Datos	30	25	-5
Prototipo funcional	500	625	+125
Testing	60	60	0
Informe final y demostración	70	80	+10
<b>Total</b>	<b>941</b>	<b>1110.5</b>	<b>+169.5</b>

Si bien la cantidad de horas estimadas por tarea fue cercana a la efectivamente realizada, se observó una desviación que resultó en un incremento del 18% sobre el total previsto. Esta diferencia se explica, en gran medida, por la falta de experiencia del equipo en varias de las áreas abordadas durante el proyecto.

En lo referente a la documentación, dicha inexperiencia hizo necesario realizar múltiples revisiones para corregir problemas de trazabilidad entre los distintos documentos,



especialmente en relación con el Documento de Requerimientos. Este proceso de ajustes sucesivos demandó más tiempo del inicialmente contemplado.

Por otro lado, en la etapa de implementación, la desviación horaria se originó principalmente por dos factores. En primer lugar, surgieron detalles no contemplados en el diseño inicial de la aplicación que se evidenciaron al realizar las primeras implementaciones. Esto obligó a revisar y modificar el diseño, así como a re-implementar partes de la solución conforme a la nueva estructura. En segundo lugar, se detectaron diferencias entre el entorno empleado durante las pruebas de concepto y el entorno real de desarrollo, lo cual requirió sustituir ciertas tecnologías planificadas y adaptar parte de la solución a dichas modificaciones.

## 5.5 Problemas encontrados

En esta sección se describen los problemas y desafíos a los que el equipo se enfrentó a la hora de implementar la solución. Si bien la cantidad de problemas encontrados no fue elevada gracias a una extensiva cantidad de pruebas de concepto y documentación detallada producida en las fases anteriores a la implementación, es menester documentar las siguientes instancias en las que el equipo se vió desafiado.

### General

Durante la implementación surgieron inconvenientes derivados de pequeños detalles en la estructura de las entidades y sus relaciones, lo que obligó a reconsiderar parte del diseño original. Estos aspectos fueron pasados por alto en la etapa de diseño y análisis, y fueron descubiertos en la etapa de implementación.

Un ejemplo es el de la estructura de los foros. Inicialmente, cada foro contenía uno o más hilos, y cada hilo estaba compuesto por un mensaje inicial al cual se le encadenaban respuestas de forma lineal, de modo que cada mensaje sólo podía tener un único mensaje hijo. Este modelo resultó limitado al momento de representar conversaciones reales, ya que no contemplaba múltiples ramas de interacción.

### Backend

Los problemas encontrados en el backend estuvieron principalmente relacionados con las diferencias entre el entorno utilizado en las pruebas de concepto y el entorno real de desarrollo y despliegue.

Inicialmente, se trabajó con Java 21, lo que generó incompatibilidades con la infraestructura de Railway, ya que dicha plataforma no ofrecía soporte para esa versión hasta etapas avanzadas de la implementación del proyecto. La solución fue migrar la aplicación a Java 17, y dado que esto estabilizó el entorno, el equipo no consideró suficientemente beneficioso volver a Java 21 una vez que Railway añadió soporte.

Al realizar las pruebas de concepto, se utilizó la librería Spring Mail de Spring Boot para la funcionalidad de envíos de mails. Esto ocasionó problemas al realizar el deploy de la aplicación en Railway, ya que Spring Mail requiere el uso de un puerto SMTP, pero Railway solo permite el uso de puertos HTTPS. Esto se solucionó utilizando un servicio externo llamado Resend [14] y comprando el dominio adan.website.



## **Frontend**

En el desarrollo del frontend, el principal inconveniente surgió durante la implementación de las notificaciones push. La causa estuvo asociada a la ausencia de pruebas de concepto previas con Firebase y a la limitada experiencia del equipo en esta tecnología. La dificultad se resolvió mediante la dedicación de tiempo adicional a la lectura de documentación oficial y a la realización de pruebas unitarias, lo que permitió comprender mejor el funcionamiento del servicio e integrarlo correctamente.

## **Mobile**

En la aplicación mobile se presentó el mismo desafío relacionado con la implementación de notificaciones push. La causa fue la misma que en el frontend, y se resolvió aplicando el mismo conjunto de acciones mencionadas previamente.

## **Documentación**

Durante la elaboración de la documentación, el principal desafío fue mantener una trazabilidad consistente entre los distintos documentos, especialmente en lo referente a la denominación y cantidad de funcionalidades de la aplicación. Esta dificultad se debió tanto a la inexperiencia del equipo en proyectos que requieren una producción documental extensa como al enfoque adoptado para el trabajo, basado en la división de los documentos entre sub-equipo. Esta estrategia, si bien permitió avanzar en paralelo, generó pequeñas inconsistencias entre los documentos, que luego debieron ser corregidas para asegurar coherencia y alineación global.



## 6. Implementación

En este capítulo se detalla la información relevante a la implementación de la plataforma. Esto abarca información sobre las tecnologías utilizadas y mockups o prototipos de las pantallas que sirvieron como referencia a la hora de desarrollar la interfaz gráfica de usuario, tanto web como mobile.

### 6.1 Tecnologías aplicadas

A continuación se describen las tecnologías utilizadas en el desarrollo de la plataforma:

#### **Backend**

El lenguaje utilizado para la programación del backend fue Java [15] en su versión 17. En conjunción, fue utilizado Spring Boot [16] por su facilidad de integración con seguridad, persistencia y servicios externos, así como por la experiencia previa del equipo con el uso de este framework.

#### **Frontend**

Para el desarrollo del frontend, se eligió utilizar React.js [17], siendo los dos factores de más peso a la hora de su elección la experiencia previa del equipo con la tecnología y el amplio soporte de la comunidad para esta librería.

#### **Mobile**

En el caso de la aplicación móvil, la elección de Kotlin [18] obedece a la extensa compatibilidad de este lenguaje con la plataforma Android, siendo este el lenguaje recomendado por Google para el desarrollo en Android. Asimismo, Kotlin reduce el código repetitivo y ofrece sintaxis moderna, clara y concisa, lo que mejora la productividad frente a Java tradicional. La falta de experiencia extensa del equipo con el lenguaje, no se tomó como un detrimento, sino como una oportunidad para profundizar los conocimientos sobre el mismo.

#### **Base de datos**

En tanto a la persistencia de datos, se eligió PostgreSQL [19] ya que este sistema de manejo de bases datos ofrece robustez y fiabilidad, así como gran compatibilidad con Spring Boot.

#### **Servidor de aplicaciones**

Para el despliegue de las aplicaciones, se seleccionó Railway [20] en el caso de la aplicación web y Vercel [21] en el caso de mobile. Railway ofrece velocidad a la hora del despliegue y facilidad de uso, además de tener fácil integración con Github y PostgreSQL, así como un sistema de precios adaptado a las necesidades de aplicaciones en fase de crecimiento. Vercel mientras tanto, ofrece un despliegue casi instantáneo de la aplicación, y buen rendimiento para aplicaciones estáticas y herramientas para el monitoreo de las mismas.



## Testing

En el caso del testing, se eligió Karate DSL [22] para el testeo de las APIs ya que su sintaxis declarativa facilita la creación de tests robustos y reproducibles. Para la UI, se eligió Playwright [23] ya que permite automatizar flujos completos de la interfaz de usuario, validar compatibilidad cross-browser, y generar reportes visuales con capturas de pantalla y videos, asegurando que la experiencia de usuario se mantenga consistente y confiable.

## 6.2 Aplicación desarrollada

En esta sección se presentan, a modo de ejemplo, algunos de los mockups elaborados durante la etapa previa a la implementación. Estos diseños preliminares sirvieron como referencia para la construcción de las interfaces gráficas de los frontends web y mobile. La aplicación desarrollada será mostrada en la demostración durante la defensa del proyecto.

### Inicio de sesión

En este mockup se muestra la vista web del inicio de sesión de los usuarios del sistema, siendo esta la que utilizarán los Administradores, Profesores y Estudiantes. También se muestra la opción para recuperar la contraseña del usuario.

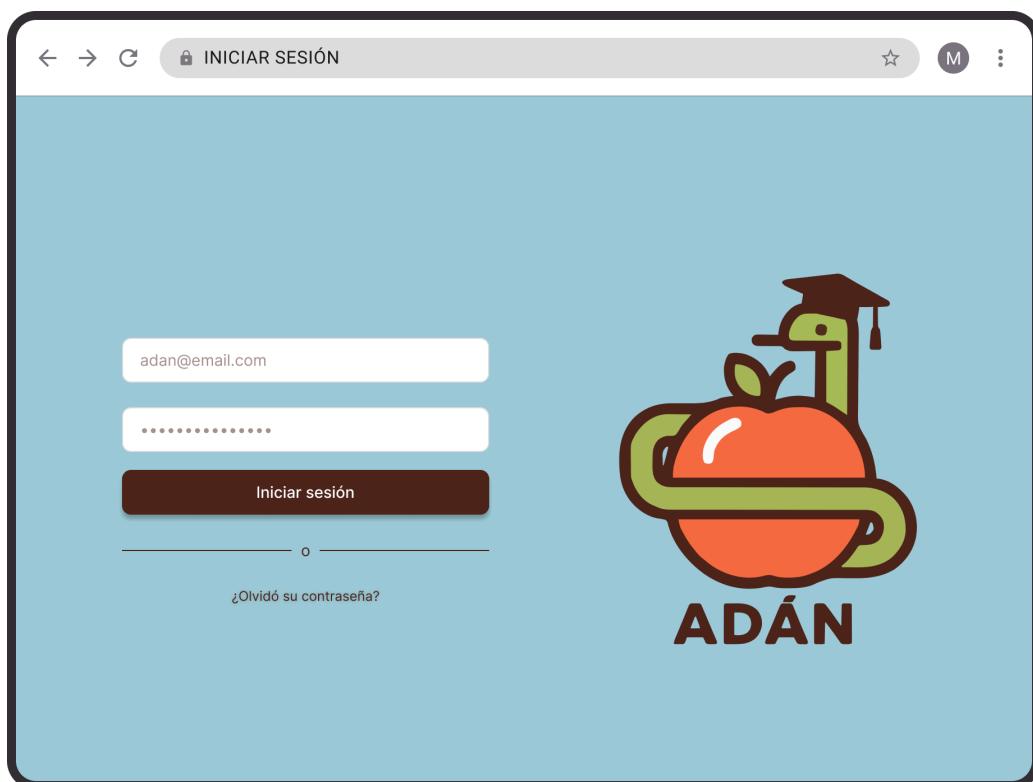


Imagen 14: Mockup de inicio de sesión web



## Vista principal de un curso

En este mockup se muestra la vista web de la página principal de un curso. En este caso específico, se pueden apreciar los botones para agregar, modificar y eliminar recursos y secciones. Si bien todos los usuarios comparten esta vista, los Estudiantes no tienen acceso a estas funcionalidades, por lo que estos botones no les son visibles.

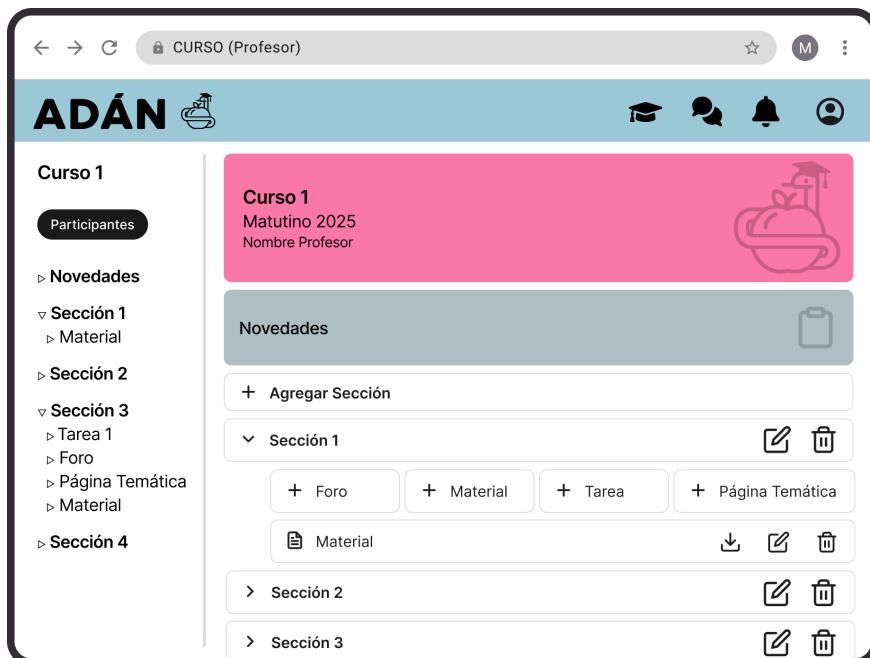


Imagen 15: Mockup de vista principal de un curso

## Creación de un curso individual

En este mockup se muestra la vista de creación de un curso individual. Esta funcionalidad solo está disponible para usuarios de tipo Administrador.

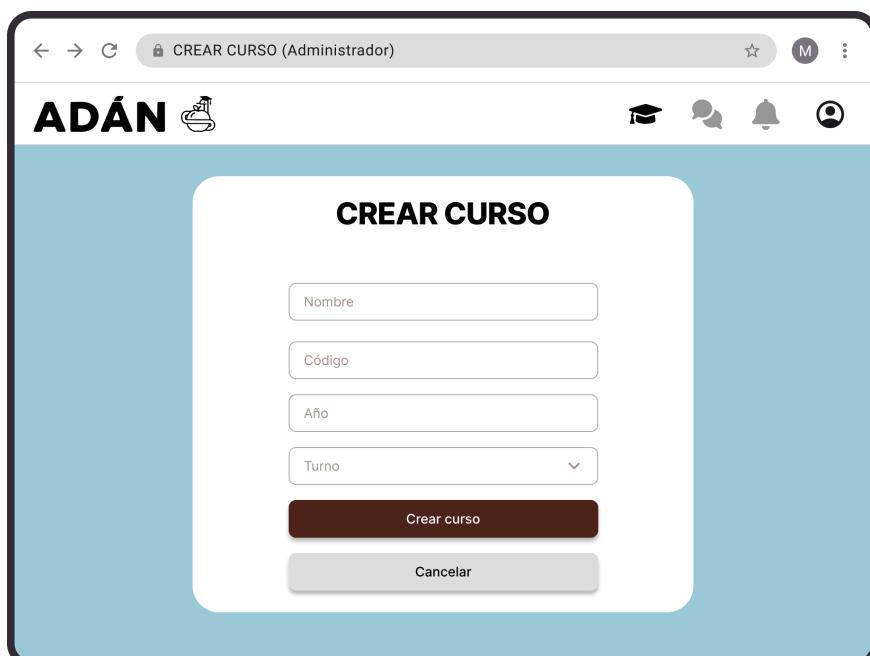


Imagen 16: Mockup de creación de curso individual



## Mobile

A continuación se muestran tres mockups de la aplicación mobile. Las siguientes son Inicio de sesión (imagen 17), vista y filtrado de los cursos del estudiante (imagen 18) y vista del chat con un profesor (imagen 19).

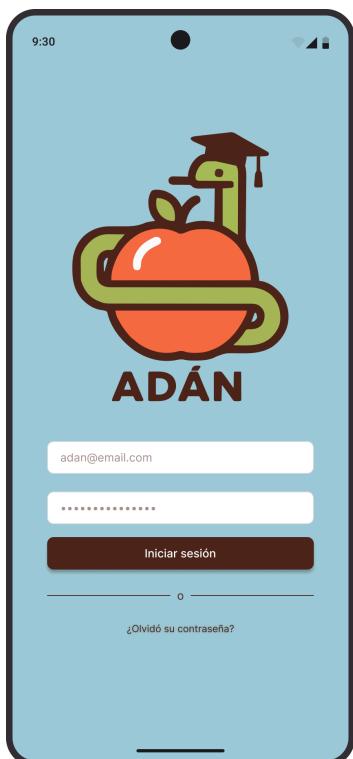


Imagen 17: Mockup de Inicio de sesión

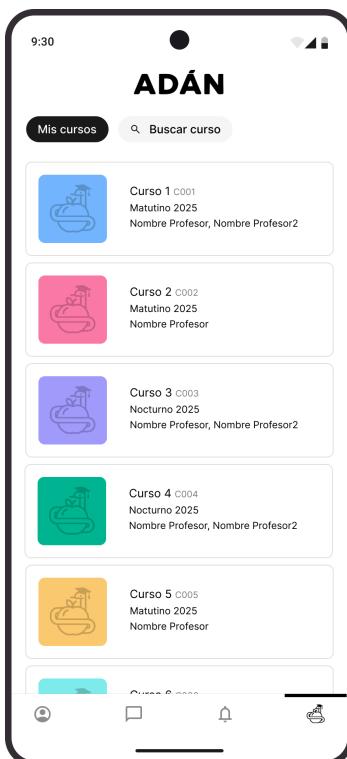


Imagen 18: Mockup de cursos del estudiante

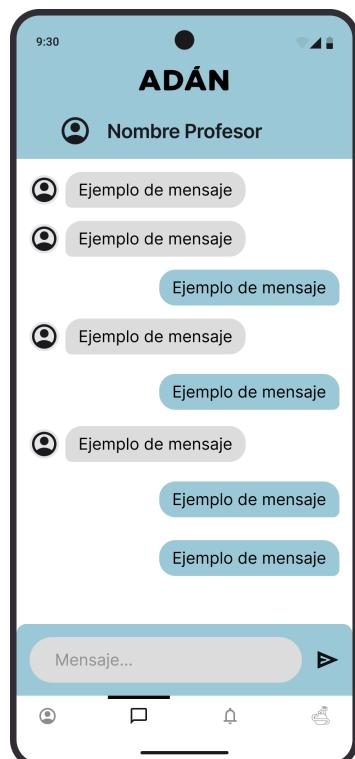


Imagen 19: Mockup de chat



## 7. Testing y validación

En este capítulo se describen las estrategias de prueba aplicadas durante el desarrollo del proyecto, junto con ejemplos de casos de prueba y sus resultados. Para información más detallada, se recomienda consultar el anexo “Documento de Verificación”, donde se profundiza en cada uno de los aspectos tratados en el presente capítulo.

### 7.1 Plan de testing

Con el objetivo de garantizar el correcto funcionamiento del sistema se definió un plan de pruebas compuesto por dos enfoques principales: pruebas manuales y pruebas automatizadas.

#### **Pruebas manuales**

Las pruebas manuales consisten en la ejecución directa de validaciones por parte de un desarrollador o tester sin el uso de herramientas automatizadas. Este enfoque permite interactuar de forma real con la aplicación, evaluando su comportamiento mediante la verificación de flujos completos, validaciones de datos y respuestas ante diferentes escenarios.

#### **Pruebas automatizadas**

Las pruebas automatizadas complementan al testeo manual, permitiendo ejecutar validaciones repetitivas de forma rápida, segura y confiable. En el marco de este proyecto se implementaron pruebas unitarias y pruebas de integración.

#### **Pruebas unitarias**

Las pruebas unitarias se enfocan en evaluar funciones o métodos individuales del sistema de forma aislada, garantizando que cada unidad de código cumpla con su comportamiento esperado. Este tipo de pruebas permite identificar fallos en el desarrollo de un módulo sin depender de otros componentes del sistema. Para lograr este aislamiento suelen utilizarse técnicas como el uso de mocks o datos simulados, lo que facilita reproducir diferentes interacciones entre módulos o capas del sistema y asegurar que el método evaluado responda de manera consistente. Gracias a este enfoque, es posible detectar errores tempranos antes de que la funcionalidad se integre con el resto de la aplicación.

#### **Pruebas de integración**

Las pruebas de integración se utilizan para verificar que los diferentes módulos y servicios del sistema funcionen de forma correcta. A diferencia de las pruebas unitarias este tipo de verificación recorre flujos completos que involucran diferentes componentes, asegurando que los datos se procesen correctamente desde el inicio hasta el final del recorrido. Este enfoque permite identificar fallos que no se manifiestan al probar componentes de forma aislada, y muestra un comportamiento más cercano al uso real de la aplicación.



## 7.2 Resultados obtenidos

A lo largo del proceso de verificación se aplicaron distintos tipos de pruebas (manuales, automatizadas unitarias y de integración) con el objetivo de evaluar la calidad del sistema a través de diferentes enfoques de prueba. En esta sección se presentan los resultados obtenidos para cada tipo de prueba.

### **Pruebas manuales**

Las pruebas manuales se realizaron de forma continua a lo largo del desarrollo del sistema con el objetivo de validar cada funcionalidad a medida que era incorporada. Este proceso permitió detectar de forma temprana errores vinculados a la interacción entre módulos, fallos en las validaciones de datos y comportamientos inesperados en los distintos flujos de uso.

Gracias a esta verificación temprana y reiterada, fue posible corregir inconsistencias antes de avanzar hacia etapas posteriores, asegurando que cada funcionalidad cumpliera con los criterios funcionales y de calidad definidos para el proyecto.

### **Pruebas unitarias**

Durante esta etapa se desarrolló un conjunto de pruebas unitarias orientadas a validar el comportamiento individual de los servicios expuestos por el sistema. Estas pruebas permitieron verificar que cada componente respondiera correctamente ante diferentes escenarios, asegurando la estabilidad y confiabilidad de la lógica interna. Al realizar estas validaciones de manera aislada, fue posible identificar errores específicos antes de integrar nuevas funcionalidades o modificar las existentes dentro del sistema. Esto permitió resolver problemas en etapas tempranas, reduciendo la complejidad del trabajo posterior y contribuyendo a mejorar la calidad general del software.

Las pruebas unitarias fueron ejecutadas utilizando Karate DSL, lo que permitió automatizar la validación del comportamiento individual de los servicios del sistema de manera consistente y repetible.

A continuación, se presentan capturas de pantalla que ilustran la ejecución completa de un ciclo de pruebas unitarias.



✓ ✓ KarateRunner (karate)	1 min 3 sec
✓ ✓ runAllTests()	1 min 3 sec
✓ ✓ karate/auth/login	4 sec 620 ms
✓ [1:3] Obtener token de usuario administrador	4 sec 620 ms
✓ ✓ karate/auth/loginProfesor	611 ms
✓ [1:3] Obtener token de usuario PROFESOR	611 ms
✓ ✓ karate/auth/loginValidoInvalido	1 sec 252 ms
✓ [1:3] Login con credenciales válidas	626 ms
✓ [2:12] Login con credenciales inválidas	626 ms
✓ ✓ karate/calificacion/calificar	1 sec 191 ms
✓ [1:7] Calificar curso	1 sec 191 ms
✓ ✓ karate/calificacion/obtenerCalificacionesCurso	1 sec 80 ms
✓ [1:7] Obtener calificaciones de curso	1 sec 80 ms
✓ ✓ karate/calificacion/obtenerCalificacionesEntregables	1 sec 109 ms
✓ [1:7] Obtener calificaciones de entregables	1 sec 109 ms
✓ ✓ karate/curso/bajaCurso	2 sec 633 ms
✓ [1:35] Baja de curso	2 sec 633 ms
✓ ✓ karate/curso/crearCurso	1 sec 78 ms
✓ [1:7] Crear curso	1 sec 78 ms
✓ ✓ karate/curso/modificarCurso	2 sec 630 ms
✓ [1:35] Modificar curso	2 sec 630 ms
✓ ✓ karate/curso/obtenerCursos	1 sec 453 ms
✓ [1:7] Obtener cursos	1 sec 453 ms
✓ ✓ karate/entregables/borrarEntrega	1 sec 97 ms
✓ [1:7] Borrar entrega	1 sec 97 ms
✓ ✓ karate/entregables/calificarEntregable	1 sec 60 ms
✓ [1:7] Calificar entregable	1 sec 60 ms

Imagen 20: Pruebas unitarias



✓ ✓ karate/entregables/obtenerEntregablesTarea	1 sec 55 ms
✓ [1:7] Obtener entregables de tarea	1 sec 55 ms
✓ ✓ karate/entregables/obtenerEntregablesUsuario	1 sec 43 ms
✓ [1:7] Obtener entregable de usuario	1 sec 43 ms
✓ ✓ karate/entregables/subirEntrega	1 sec 148 ms
✓ [1:7] Subir entregable	1 sec 148 ms
✓ ✓ karate/foro/crearForo	1 sec 645 ms
✓ [1:8] Crear foro	1 sec 645 ms
✓ ✓ karate/foro/eliminarForo	2 sec 854 ms
✓ [1:8] Eliminar foro	2 sec 854 ms
✓ ✓ karate/mensaje/modificarMensaje	1 sec 76 ms
✓ [1:7] Modificar mensaje	1 sec 76 ms
✓ ✓ karate/mensaje/obtenerMensajes	1 sec 865 ms
✓ [1:7] Obtener mensajes	1 sec 865 ms
✓ ✓ karate/mensaje/publicarMensaje	1 sec 37 ms
✓ [1:7] Publicar mensaje	1 sec 37 ms
✓ ✓ karate/mensaje/responderMensaje	1 sec 77 ms
✓ [1:7] Responder mensaje	1 sec 77 ms
✓ ✓ karate/mensaje_privado/enviarMensaje	1 sec 28 ms
✓ [1:7] Enviar mensaje privado	1 sec 28 ms
✓ ✓ karate/mensaje_privado/obtenerConversacion	1 sec 634 ms
✓ [1:7] Obtener conversacion	1 sec 634 ms
✓ ✓ karate/mensaje_privado/obtenerInterlocutores	1 sec 39 ms
✓ [1:7] Obtener interlocutores	1 sec 39 ms
✓ ✓ karate/paginaTematica/crearPaginaTematica	1 sec 607 ms
✓ [1:8] Crear página temática	1 sec 607 ms
✓ ✓ karate/paginaTematica/eliminarPaginaTematica	2 sec 730 ms
✓ [1:8] Eliminar página temática	2 sec 730 ms

Imagen 21: Pruebas unitarias



✓ ✓ karate/recursos/obtenerRecusos	1 sec 225 ms
✓ [1:8] Obtener recursos (foros, páginas, etc)	1 sec 225 ms
✓ ✓ karate/seccion/altaSeccion	1 sec 886 ms
✓ [1:7] Alta de sección	1 sec 886 ms
✓ ✓ karate/seccion/eliminarSeccion	3 sec 22 ms
✓ [1:7] Eliminar sección	3 sec 22 ms
✓ ✓ karate/seccion/modificarSeccion	3 sec 18 ms
✓ [1:7] Modificar sección	3 sec 18 ms
✓ ✓ karate/seccion/obtenerSecciones	1 sec 143 ms
✓ [1:7] Obtener secciones	1 sec 143 ms
✓ ✓ karate/tarea/crearTarea	1 sec 616 ms
✓ [1:8] Crear tarea	1 sec 616 ms
✓ ✓ karate/tarea/eliminarTarea	2 sec 699 ms
✓ [1:8] Eliminar tarea	2 sec 699 ms
✓ ✓ karate/usuario/alta	1 sec 331 ms
✓ [1:7] Alta de usuario	1 sec 331 ms
✓ ✓ karate/usuario/bloquearDesbloquear	3 sec 631 ms
✓ [1:35] Bloquear/desbloquear usuario	3 sec 631 ms
✓ ✓ karate/usuario/obtenerPerfil	1 sec 57 ms
✓ [1:7] Obtener perfil de usuario	1 sec 57 ms
✓ ✓ karate/usuario/obtenerUsuarios	1 sec 562 ms
✓ [1:7] Obtener usuarios	1 sec 562 ms

*Imagen 22: Pruebas unitarias*

### Pruebas de integración

En esta etapa se desarrollaron pruebas de integración orientadas a verificar la correcta funcionalidad de los diferentes módulos y servicios del sistema a través de flujos completos. A diferencia de las pruebas unitarias, este tipo de verificación simula un comportamiento más cercano al uso real de la aplicación.

Para la verificación de la integración y la experiencia de usuario extremo a extremo (E2E), se implementó un conjunto de pruebas automatizadas utilizando Playwright. Este marco simula la interacción completa del usuario (incluyendo autenticación, navegación y manipulación de datos), asegurando que los flujos críticos de negocio operen correctamente y mantengan la integridad de la información.

A continuación, se presentan capturas de la ejecución de estas pruebas de integración, que sugiere la robustez del sistema en sus flujos críticos.



<input type="text" value="Q"/>	All 41	<span style="color: green;">✓</span> Passed 41	Failed 0	Flaky 0	Skipped 0	
Project: chromium						5/12/2025, 3:17:39 p. m. Total time: 42.6s
<b>altaforo.spec.ts</b>						
✓ <b>crear foro ok</b> <small>chromium</small>						5.9s
altaforo.spec.ts:9						
✓ <b>crear foro duplicado</b> <small>chromium</small>						5.9s
altaforo.spec.ts:18						
✓ <b>crear foro vacío</b> <small>chromium</small>						4.9s
altaforo.spec.ts:25						
<b>altapagina.spec.ts</b>						
✓ <b>crear pagina ok</b> <small>chromium</small>						6.0s
altapagina.spec.ts:10						
✓ <b>crear pagina duplicada</b> <small>chromium</small>						4.8s
altapagina.spec.ts:20						
✓ <b>crear pagina vacío</b> <small>chromium</small>						2.8s
altapagina.spec.ts:28						
<b>altaseccion.spec.ts</b>						
✓ <b>crear sección ok</b> <small>chromium</small>						3.6s
altaseccion.spec.ts:11						
✓ <b>crear sección duplicada</b> <small>chromium</small>						3.6s
altaseccion.spec.ts:20						
✓ <b>crear sección vacía</b> <small>chromium</small>						2.6s
altaseccion.spec.ts:27						
<b>altatarea.spec.ts</b>						
✓ <b>crear tarea ok</b> <small>chromium</small>						3.9s
altatarea.spec.ts:11						
✓ <b>crear tarea duplicada</b> <small>chromium</small>						3.8s
altatarea.spec.ts:23						
✓ <b>crear tarea vacía</b> <small>chromium</small>						2.9s
altatarea.spec.ts:33						
✓ <b>crear tarea con fechas incorrectas</b> <small>chromium</small>						2.8s
altatarea.spec.ts:39						
<b>asignarprofesor.spec.ts</b>						
✓ <b>asignar profesor</b> <small>chromium</small>						10.3s
asignarprofesor.spec.ts:8						
<b>buscarcurso.spec.ts</b>						
✓ <b>buscar con registros</b> <small>chromium</small>						4.6s
buscarcurso.spec.ts:9						
<b>calificarcsv.spec.ts</b>						
✓ <b>crear calificaciones ok</b> <small>chromium</small>						4.2s
calificarcsv.spec.ts:10						
✓ <b>crear calificaciones sin archivo</b> <small>chromium</small>						2.6s
calificarcsv.spec.ts:21						
<b>crearcursito.spec.ts</b>						
✓ <b>crear curso ok</b> <small>chromium</small>						3.5s
crearcursito.spec.ts:9						
✓ <b>crear curso duplicado</b> <small>chromium</small>						3.5s
crearcursito.spec.ts:22						
✓ <b>crear curso con campos incompletos</b> <small>chromium</small>						2.7s
crearcursito.spec.ts:33						

Imagen 23: Pruebas de integración



✓ <b>crearcursocsv.spec.ts</b>	
✓ <b>crear cursos ok</b> chromium	4.8s
crearcursocsv.spec.ts:10	
✓ <b>crear cursos sin archivo</b> chromium	2.5s
crearcursocsv.spec.ts:20	
✓ <b>crearhilo.spec.ts</b>	
✓ <b>crear hilo ok</b> chromium	3.7s
crearhilo.spec.ts:10	
✓ <b>crear hilo vacío</b> chromium	2.7s
crearhilo.spec.ts:17	
✓ <b>crearusuario.spec.ts</b>	
✓ <b>crear admin ok</b> chromium	4.5s
crearusuario.spec.ts:9	
✓ <b>crear usuario normal ok</b> chromium	4.5s
crearusuario.spec.ts:20	
✓ <b>crear usuario duplicado correo</b> chromium	3.5s
crearusuario.spec.ts:37	
✓ <b>crear usuario duplicado cedula</b> chromium	3.5s
crearusuario.spec.ts:45	
✓ <b>crear usuario con campos incompletos</b> chromium	2.7s
crearusuario.spec.ts:57	
✓ <b>crearusuarioscsv.spec.ts</b>	
✓ <b>crear usuarios ok</b> chromium	4.9s
crearusuarioscsv.spec.ts:10	
✓ <b>crear usuarios sin archivo</b> chromium	2.8s
crearusuarioscsv.spec.ts:20	
✓ <b>login.spec.ts</b>	
✓ <b>login ok</b> chromium	3.5s
login.spec.ts:3	
✓ <b>login campos incompletos</b> chromium	603ms
login.spec.ts:11	
✓ <b>login con credenciales erroneas</b> chromium	2.5s
login.spec.ts:18	
✓ <b>matricularestudiantes.spec.ts</b>	
✓ <b>matricular estudiantes ok</b> chromium	4.8s
matricularestudiantes.spec.ts:10	
✓ <b>matricular estudiantes sin archivo</b> chromium	2.6s
matricularestudiantes.spec.ts:20	
✓ <b>publicaranuncio.spec.ts</b>	
✓ <b>crear anuncio ok</b> chromium	3.7s
publicaranuncio.spec.ts:11	
✓ <b>crear anuncio duplicado</b> chromium	3.8s
publicaranuncio.spec.ts:22	
✓ <b>crear anuncio vacío</b> chromium	2.8s
publicaranuncio.spec.ts:30	
✓ <b>publicarmensaje.spec.ts</b>	
✓ <b>crear mensaje ok</b> chromium	4.4s
publicarmensaje.spec.ts:17	
✓ <b>crear mensaje vacío</b> chromium	2.6s
publicarmensaje.spec.ts:25	

Imagen 24: Pruebas de integración



## 7.3 Conclusión

Tras completar el proceso de verificación mediante pruebas manuales y automatizadas se concluye que el sistema funciona conforme a lo esperado y no presenta fallos relevantes en los flujos principales. Las validaciones realizadas permitieron asegurar la estabilidad de las funcionalidades implementadas y el cumplimiento de los requisitos definidos.



## 8. Conclusiones y trabajo a futuro

En este capítulo se documentan las conclusiones finales obtenidas por el equipo a partir del proceso de desarrollo de la plataforma, así como funcionalidades aún no implementadas y posibles mejoras a futuro.

### 8.1 Conclusiones finales

Si bien la naturaleza de la plataforma a desarrollar era familiar a los miembros del equipo, siendo los EVAs la forma más común en la que estos interactúan con la institución educativa a la que pertenecen, el diseño e implementación de una plataforma de este tipo implicó un desafío diferente y estimulante. A su vez, el proceso de desarrollo con relativa rigidez estructural y alto énfasis en la documentación también significó un cambio de paradigma para el equipo, pero significó una oportunidad de crecimiento personal como desarrolladores y miembros de un equipo de trabajo.

Inicialmente, se buscó desarrollar una solución que atendiera las carencias identificadas por el equipo en las plataformas existentes, por lo que se hizo un extenso trabajo de investigación del estado del arte. Luego, durante la selección de tecnologías, se evaluaron factores como la facilidad de implementación, las ventajas que cada tecnología ofrecía y el nivel de experiencia del equipo. Esto permitió alcanzar un equilibrio entre el uso de herramientas conocidas y la incorporación de tecnologías nuevas que pudieran contribuir al crecimiento del grupo. Específicamente, si bien se utilizó React para el frontend web, se prefirió alejarse de React Native, eligiendo en su lugar Kotlin para el desarrollo del frontend mobile.

En la etapa siguiente, siendo esta la de análisis y diseño, el obstáculo más grande fue mantener la trazabilidad entre los distintos documentos, requiriendo extensivo trabajo de revisión y modificación para llegar a los altos estándares de calidad y detalle que fueron requeridos, y siendo esto una experiencia desafiante, pero altamente enriquecedora y provechosa, ya que el trabajo realizado en esta etapa proporcionó una base robusta y clara para la etapa de implementación.

Dicha etapa de implementación, no fue ajena a problemas y sorpresas, pero, como fue mencionado anteriormente, el trabajo realizado en el diseño facilitó en gran medida el desarrollo del sistema y resultó en un producto funcional y de una calidad que el equipo encuentra satisfactoria, cumpliendo con los requisitos funcionales planteados inicialmente así como con las expectativas del equipo.

Para finalizar, este proceso representó una valiosa oportunidad de aprendizaje que el equipo considera haber aprovechado plenamente. No solo permitió profundizar conocimientos técnicos en áreas ya conocidas, sino también incorporar nuevas tecnologías y adoptar distintas formas de trabajo. Se destaca especialmente la experiencia de trabajar dentro de un equipo organizado y con una dirección clara, junto con los beneficios y desafíos que esto implicó.



## 8.2 Trabajo a futuro

A continuación se presentan posibles líneas de trabajo a futuro, correspondientes a funcionalidades que el equipo considera de alto valor para la evolución del producto.

### **Autenticación externa**

Permitir el inicio de sesión mediante credenciales de otras plataformas simplificaría el proceso de autenticación, facilitando el acceso con un solo clic.

### **Autenticación 2FA**

Incorporar métodos de doble factor reforzaría la seguridad de las cuentas y la protección de los datos de los usuarios.

### **Soporte para IOS**

Desarrollar una versión de la aplicación para dispositivos iOS aumentaría su disponibilidad, considerando que este es el segundo sistema operativo móvil más utilizado.

### **Opciones de accesibilidad**

Funciones como lectura de texto (text-to-speech) o configuraciones para usuarios con discapacidades visuales mejorarían la inclusión y accesibilidad de la plataforma.

### **Cuestionarios**

Agregar cuestionarios como nuevo tipo de recurso dentro de los cursos permitiría recolectar feedback de los estudiantes y ofrecer actividades evaluativas de forma más dinámica y personalizada.

### **Sistema de cálculo de la calificación final**

Automatizar el cálculo de la calificación final facilitaría el trabajo docente y aportaría mayor transparencia al proceso de evaluación.

### **Videos embebidos**

Habilitar la incorporación de vídeos embebidos en los recursos de un curso ampliaría las posibilidades de contenido, permitiendo ofrecer materiales más dinámicos y enriquecedores para los estudiantes.

### **Cartelera de novedades institucional**

Incorporar un espacio destinado a comunicar novedades institucionales brindaría a los usuarios acceso directo a información relevante de su institución.

### **Mejoras en la seguridad**

Si bien algunas funcionalidades ya cuentan con medidas de seguridad, se podrían incorporar estos controles al resto del sistema, con el fin de fortalecer la protección general y evitar accesos no autorizados.



## 9. Referencias

[1]

[¿Qué es un entorno virtual de aprendizaje? | Red Educa](#) - ¿Qué es un entorno virtual de aprendizaje? (Septiembre 2025)

[2]

<https://www.uv.es/bellochc/pedagogia/EVA3.pdf> - Características de los EVA (Septiembre 2025)

[3]

[La plataforma educativa CREA del Plan Ceibal en el top de las tendencias de Google](#) - La plataforma educativa CREA del Plan Ceibal en el top de las tendencias de Google (Agosto 2025)

[4]

[Moodle – Departamento de Apoyo Técnico Académico](#) - Entornos Virtuales de Aprendizaje (EVA) de la UDELAR (Septiembre 2025)

[5]

<https://moodle.com/es/acerca-de/> - Acerca de: Moodle (Agosto 2025)

[6]

<https://www.powerschool.com/solutions/personalized-learning/schoology-learning/> - Acerca de: Schoology (Septiembre 2025)

[7]

<https://web.archive.org/web/20130324102126/https://www.schoology.com/history.php> - Historia Schoology (Septiembre 2025)

[8]

<https://www.powerschool.com/company/> - Historia Powerschool (Septiembre 2025)

[9]

<https://tecnologiaeducativa.cfe.edu.uy/index.php/home/95-nuevo-distrito-schoology-para-cfe> - Nuevo distrito de Schoology para CFE (Septiembre 2025)

[10]

[About Classroom - Classroom Help](#) - Acerca de: Google Classroom (Agosto 2025)

[11]

<https://www.instructure.com/about> - Acerca de: Canvas (Agosto 2025)

[12]

[University LMS - Eanes ISD](#) - Uso de LMS en Universidades (Septiembre 2025)



[13]

<https://www.usi.org.uy/files.php/noticias/2461/estudio-usos-de-internet-en-uruguay.pdf> - Los usos de Internet en el Uruguay. (Agosto 2025)

[14]

[Send emails with Railway - Resend](#) - Send emails with Railway (Septiembre 2025)

[15]

[Java | Oracle](#) - Java (Septiembre 2025)

[16]

<https://spring.io/projects/spring-boot> - Spring Boot (Septiembre 2025)

[17]

[React](#) - React.js (Septiembre 2025)

[18]

[Kotlin Docs | Kotlin Documentation](#) - Kotlin (Septiembre 2025)

[19]

[PostgreSQL: Documentation](#) - PostgreSQL (Septiembre 2025)

[20]

[Railway Docs](#) - Railway (Septiembre 2025)

[21]

[Vercel Documentation](#) - Vercel (Septiembre 2025)

[22]

[Test Automation Made Simple - Karate Framework | Karate](#) - Karate DSL (Octubre 2025)

[23]

[Installation | Playwright](#) - Playwright (Octubre 2025)

[Imagen base para el logo del proyecto]

[https://www.flaticon.es/icono-gratis/serpiente\\_3805822](https://www.flaticon.es/icono-gratis/serpiente_3805822) - 'Serpiente icono gratis' Darius Dan



## 10. Anexos

En esta sección se presentan brevemente los documentos adjuntos a este informe, los cuales contienen información detallada correspondiente a cada uno de sus capítulos.

### 10.1 Documento de Alcance

El documento de alcance tiene como finalidad establecer los límites, metas y entregables del proyecto. En este documento se especifican las funcionalidades que formarán parte del sistema y la planificación general de las etapas de trabajo.

### 10.2 Documento de Arquitectura

El propósito del documento de arquitectura es presentar la visión técnica general del sistema, detallando el enfoque arquitectónico adoptado y los componentes principales que lo conforman. Este documento describe cómo se organizan dichos componentes dentro del estilo arquitectónico seleccionado.

### 10.3 Documento de Casos de Uso

El documento de casos de uso especifica las interacciones entre los actores y el sistema, organizando los requerimientos funcionales en escenarios claros y comprensibles. Cada caso de uso detalla los pasos necesarios para lograr un objetivo del usuario.

### 10.4 Documento de Diseño

El documento de diseño se encarga de describir la forma en que se implementarán las funcionalidades definidas, explicando su estructura interna y el funcionamiento de los distintos módulos del sistema. Incluye diagramas de secuencia, diagramas de clases y mockups.

### 10.5 Documento de Requerimientos

El objetivo del documento de requerimientos es recopilar y detallar las necesidades funcionales y no funcionales que el sistema debe satisfacer.



## 10.6 Documento de Pruebas de Concepto

El documento de pruebas de concepto tiene como fin registrar el análisis técnico realizado para determinar qué tecnologías resultan más adecuadas para el proyecto. Incluye evaluaciones, comparativas y pruebas necesarias para validar la viabilidad de cada opción tecnológica.

## 10.7 Documento de Verificación

Este documento describe el plan de verificación implementado para asegurar que el sistema cumple con los requerimientos y el diseño establecidos.

## 10.8 Documento de Modelos de Datos

El documento de modelo de datos tiene como finalidad presentar la estructura de la información que maneja el sistema. Describe las tablas, relaciones, claves y restricciones que conforman la base de datos.

## 10.9 Documento de Modelo de Dominio

El documento de modelo de dominio representa las entidades más relevantes del sistema y las relaciones entre ellas. Mediante diagramas UML y descripciones textuales se especifican los atributos, reglas y vínculos que forman parte del núcleo conceptual del proyecto.

## 10.10 Glosario

El glosario reúne definiciones de términos y conceptos mencionados a lo largo de toda la documentación.

