Análisis de lectura de textos

Text reading analysis

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:

Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),

Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

A mis padres y abuelos,
por su amor, apoyo incondicional
y por siempre creer en mí.
Este logro es tan suyo como mío.

Análisis de lectura de textos
Memoria

Trabajo de Fin de Grado
GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:
Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),
Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

Índice general

Resumen **IX**

English Abstract **XI**

1. Introducción **1**

1.1. Estructura de la memoria 2

1.2. Anexos 2

2. Objetivos **5**

2.1. Objetivos del sistema 5

2.2. Objetivos personales 6

3. Conceptos teóricos **7**

3.1. Patrones de Diseño 7

3.2. Conceptos Relacionados 9

4. Métodos y herramientas **11**

4.1. Métodos 11

4.2. Herramientas 11

5. Aspectos relevantes del desarrollo del proyecto **15**

5.1. Marco de trabajo y metodología 15

5.2. Estimación de la duración del proyecto 16

5.3. Planificación temporal 17

5.4. Especificación de requisitos 21

5.4.1. Participantes 21

5.4.2. Objetivos del Sistema 21

5.4.3. Requisitos de Información 23

5.4.4. Requisitos Funcionales 24

5.4.5. Requisitos No Funcionales 29

5.5. Análisis del sistema 30

5.5.1. Modelo de dominio 31

5.5.2. Realización de casos de uso 31

5.5.3. Clases de Análisis y Descripción de la Arquitectura 32

5.6. Diseño del sistema 33

5.6.1. Patrón arquitectónico 33

5.6.2. Subsistemas de diseño 34

5.6.3. Clases de diseño 34

5.6.4. Realización de Casos de Uso 35

5.6.5. Base de Datos 35

5.7. Funcionalidad del sistema 36

5.7.1. Usuario No Identificado 36

5.7.2. Usuario Identificado 38

5.7.3. Participante 40

5.7.4. Entrenador 42

5.8. Administrador 45

5.8.1. Gestión de Usuarios 45

6. Limitaciones del Sistema **47**

7. Líneas de Trabajo Futuras **49**

8. Conclusiones **51**

Bibliografía **52**

Índice de figuras

1. Diagrama del proceso unificado. 16
2. Estimación temporal. 17
3. Diagrama de Gantt de la vista global de la planificación temporal. 18
4. Planificación global. 19
5. Estadísticas de la planificación temporal. 20
6. Diagrama de paquetes. 25
7. Actores. 26
8. Autentificación de usuarios. 27
9. Diagrama de dominio. 30
10. Diagrama de secuencia de iniciar sesión. 31
11. Diagrama de paquetes de análisis. 32
12. Patrón cliente-servidor. 33
13. Subsistema de diseño. 34
14. Diagrama de secuencia: registrar usuario. 35
15. Base de datos. 35
16. Vista de inicio de Usuario No Identificado. 36
17. Vista general de la pantalla de inicio de sesión. 37

18. Vista de la pantalla de registro. 37
19. Vista del inicio del usuario identificado. 38
20. Formulario de modificación del perfil. 39
21. Botón para cerrar la sesión. 39
22. Vista del apartado de actividades por el usuario. 40
23. Vista de las sesiones sin finalizar. 41
24. Vista de grabación de entrenamiento con texto. 41
25. Vista de gestión de sesiones de entrenamiento. 42
26. Vista de gestión de actividades. 43
27. Vista de gestión de ítems. 44
28. Vista de gestión de resultados en el apartado de subir tareas. 44
29. Vista de gestión de usuarios por el administrador. 45

Índice de tablas

1. Especificación de participantes. 21
2. OBJ-0001 Gestión de actividades. 23
3. IRQ-001: Información sobre usuarios. 24
4. ACT-01 Usuario no identificado. 27
5. UC-001: Registrar usuario. 28
6. NFR-001: Usabilidad. 29

Resumen

Este proyecto tiene como objetivo el desarrollo de una plataforma que facilite la creación y edición de un banco de pruebas para sesiones de actividades de lectura. Estas sesiones, realizadas por los usuarios, generan resultados que pueden ser analizados mediante tareas específicas cargadas al sistema por terceros, permitiendo obtener análisis detallados.

El enfoque del proyecto se centra en la gestión eficiente de los datos generados durante las actividades de lectura, utilizando un modelo de desarrollo incremental. Este enfoque permite la evolución continua de la plataforma y la incorporación progresiva de nuevas funcionalidades. Se han seleccionado herramientas y frameworks modernos para asegurar un manejo adecuado de los datos y garantizar la escalabilidad futura del sistema, manteniendo su operatividad desde las primeras fases del desarrollo.

Un ejemplo de uso sería emplear la plataforma para evaluar las habilidades de lectura de los estudiantes. El profesor crea una sesión con actividades de lectura, y los estudiantes generan audios al leer en voz alta. Estos audios se cargan en el sistema, que analiza aspectos seleccionados por el profesor, como pronunciación y fluidez. Con los resultados del análisis de cada estudiante, el profesor puede identificar áreas de mejora y adaptar la enseñanza de manera más efectiva.

English Abstract

This project proposes the development of a platform primarily aimed at enabling the creation and editing of a test bank for reading activity sessions. These sessions, conducted by users, generate results that can be analyzed using specific tasks uploaded to the system by third parties, allowing for detailed analysis.

The project focuses on the efficient management of data generated during reading activities, employing an incremental development model. This approach facilitates the continuous evolution of the platform and the progressive integration of new features. Modern tools and frameworks have been selected to ensure proper data handling and future scalability of the system, maintaining its operability from the early stages of development.

An example of use would be employing the platform to assess students' reading skills. The teacher creates a session with reading activities, and students generate audio recordings of themselves reading aloud. These recordings are uploaded to the system, which analyzes selected aspects such as pronunciation and fluency. With the analysis results for each student, the teacher can identify areas for improvement and adjust the teaching approach more effectively.

1 Introducción

El análisis de lectura de texto se refiere al estudio y evaluación de cómo los usuarios interactúan con los textos escritos, con el objetivo de identificar dificultades y áreas de mejora. Aunque está relacionado con el procesamiento del lenguaje natural (PLN) [1], el análisis de lectura de texto se centra específicamente en la interpretación y evaluación del comportamiento lector, lo que lo convierte en una herramienta valiosa en el ámbito educativo y de la evaluación cognitiva.

Este proyecto se enfoca en el desarrollo de una plataforma para el procesamiento del lenguaje natural aplicada a la lectura. La plataforma permitirá la creación y edición de un banco de pruebas compuesto por sesiones de actividades de lectura, generando resultados que podrán ser analizados mediante tareas específicas cargadas por terceros. Esta funcionalidad proporciona un marco para evaluar de manera detallada las capacidades de lectura de los usuarios a través de análisis específicos realizados por tareas.

En colaboración con la empresa Glifing[30], especializada en desarrollar métodos para mejorar las habilidades de lectura en niños, este proyecto tiene como objetivo principal apoyar el aprendizaje de la lectura de manera efectiva. Glifing se dedica a crear herramientas que permiten a los niños aprender a leer correctamente, y esta plataforma se diseña para complementar y potenciar sus metodologías, permitiendo un análisis detallado y una personalización de las actividades de lectura según las necesidades individuales de los usuarios.

En el contexto educativo y formativo, la capacidad de analizar los resultados obtenidos de las actividades de lectura es fundamental para mejorar la eficacia de los métodos de enseñanza. Este proyecto tiene como objetivo desarrollar una herramienta que gestione de manera eficiente los datos relacionados con la lectura, garantizando su escalabilidad y robustez para facilitar una evaluación precisa y continua del aprendizaje.

1.1 Estructura de la memoria

Para obtener una visión global y clara de los principales aspectos del proyecto, esta memoria se ha dividido en los siguientes capítulos:

- **Objetivos del proyecto**: se especifican los objetivos generales y específicos del proyecto, con un enfoque en los resultados esperados.
- **Conceptos teóricos**: se explican los fundamentos teóricos necesarios para entender el proyecto.
- **Técnicas y herramientas**: se presentan las metodologías y herramientas empleadas en el desarrollo del proyecto, justificando las decisiones tomadas.
- **Aspectos Relevantes del Desarrollo del Proyecto**: se detallan los elementos clave del desarrollo del software, incluyendo decisiones de diseño, implementación y desafíos encontrados.
- **Conclusiones y líneas de trabajo futuras**: se resumen las conclusiones derivadas del proyecto y se proponen posibles mejoras y extensiones para futuras investigaciones.

Además, se han incluido varios anexos que documentan aspectos específicos del proyecto.

1.2 Anexos

Cada anexo incluye un índice específico para facilitar la localización de los distintos apartados.

- **A1. Plan del proyecto software**: este anexo detalla el plan del proyecto, incluyendo la planificación temporal, asignación de recursos y estimaciones de esfuerzo.
- **A2. Especificación de requisitos de software**: describe los requisitos funcionales y no funcionales del sistema, junto con los casos de uso principales.
- **A3. Análisis de requisitos**: en este anexo se presenta un análisis exhaustivo de los requisitos necesarios para el desarrollo del sistema, identificando tanto las necesidades del usuario como las restricciones del sistema. Se detallan los requisitos funcionales y no funcionales, además de los casos de uso que guían el desarrollo.
- **A4. Diseño del sistema de software**: proporciona un análisis detallado del diseño del sistema, incluyendo diagramas de arquitectura, modelos de base de datos y especificaciones de interfaces.

- **A5. Manuales de usuario**: proporciona guías de usuario tanto para administradores como para usuarios finales de la plataforma, explicando cómo utilizar las funcionalidades del sistema.

2 Objetivos

En este apartado se identifican tanto los objetivos personales como los objetivos del sistema que se deben alcanzar para un desarrollo exitoso del proyecto. El objetivo principal es desarrollar una plataforma integral que permita la gestión eficiente de usuarios, actividades de lectura, sesiones de entrenamiento y el almacenamiento y análisis de resultados de los entrenamientos, todo ello en un entorno digital seguro y escalable.

2.1 Objetivos del sistema

El objetivo principal de este proyecto es desarrollar una plataforma integral que facilite la recolección, gestión y análisis de datos de actividades de lectura, apoyando tanto a investigadores como a educadores en la comprensión y mejora de las habilidades de lectura de los usuarios. Para alcanzar este objetivo, se han definido los siguientes objetivos específicos del sistema:

- **Desarrollo de una plataforma de análisis de actividades de lectura:** crear una plataforma que permita la creación, edición y gestión de un banco de pruebas compuesto por sesiones de actividades de lectura. Estas actividades generan resultados diversos a partir de las tareas, que pueden ser analizados y utilizados para obtener métricas y estadísticas. Esta capacidad facilita el estudio detallado de los datos recolectados y su aplicación en contextos educativos y formativos.
- **Gestión de usuarios y datos:** la plataforma debe permitir la creación, modificación, visualización y eliminación de usuarios, así como la gestión eficiente de los datos recolectados de las actividades de lectura.
- **Almacenamiento y procesamiento de datos a gran escala:** utilizar herramientas y frameworks modernos como Django REST Framework y Angular, para asegurar un manejo adecuado de los datos, garantizando la escalabilidad y robustez del sistema.

- **Interoperabilidad con códigos externos:** habilitar el sistema para recibir e integrar tareas codificadas por terceros, permitiendo la ejecución y análisis de resultados generados por estos códigos externos. Esta capacidad aumenta la flexibilidad y adaptabilidad del sistema para distintos contextos educativos.

2.2 Objetivos personales

En cuanto a los objetivos personales, se destacan los siguientes:

- **Desarrollo del trabajo de fin de grado.** Realizar el desarrollo del Trabajo de Fin de Grado de manera efectiva, aplicando los conocimientos adquiridos durante la carrera, y beca curricular realizada en la empresa Glifing SL., enfrentando el desafío de gestionar un proyecto de software integral de forma individual.
- **Crecimiento profesional.** Ampliar las habilidades en el desarrollo web y la gestión de proyectos de software mediante el aprendizaje de nuevas tecnologías y herramientas relevantes, como Django para backend, así como técnicas avanzadas de gestión de bases de datos y análisis de datos.
- **Gestión del tiempo y tareas.** Mejorar las habilidades de gestión del tiempo y organización para compaginar el desarrollo del proyecto con otras responsabilidades académicas y profesionales, aplicando una metodología de trabajo ágil para asegurar un avance constante y significativo en el proyecto.
- **Contribución al ámbito educativo.** Desarrollar una plataforma que no solo cumpla con los objetivos técnicos, sino que también aporte valor al ámbito educativo al mejorar las herramientas disponibles para el entrenamiento de habilidades de lectura, contribuyendo así al desarrollo de nuevas metodologías educativas basadas en tecnología.

3 Conceptos teóricos

Para comprender el desarrollo de la plataforma, es crucial conocer algunos conceptos teóricos clave relacionados con la tecnología utilizada. A continuación, se detallan estos conceptos, estructurados de manera que se explique cómo se implementan en el proyecto.

3.1 Patrones de Diseño

Patrones de Diseño: Cliente-Servidor y MVC. En el desarrollo de la plataforma, se han aplicado dos patrones de diseño fundamentales: el modelo-vista-controlador (MVC) y la arquitectura cliente-servidor. Estos patrones permiten una separación clara de responsabilidades y una organización eficiente del código.

- **Arquitectura Cliente-Servidor:** Este patrón divide el sistema en dos componentes principales: el cliente y el servidor. El cliente solicita servicios o datos al servidor, que luego procesa la solicitud y devuelve la respuesta correspondiente. Esta arquitectura facilita la escalabilidad y permite que diferentes partes del sistema se desarrollen y desplieguen de manera independiente.
- **Modelo-Vista-Controlador (MVC):** Este patrón organiza una aplicación en tres componentes principales:
    - **Modelo:** Representa la capa de persistencia de datos. En nuestro proyecto, la persistencia se maneja a través de una base de datos relacional MySQL. El modelo se encarga de gestionar los datos, realizar consultas y mantener la integridad de la información.
    - **Vista:** Es la capa encargada de la presentación de la información al usuario. En esta plataforma, la vista está implementada con Angular. Esta tecnología permite construir interfaces de usuario interactivas y dinámicas, proporcionando una experiencia de usuario fluida y moderna.

- **Controlador:** Actúa como intermediario entre el modelo y la vista. En nuestro caso, el controlador está implementado con Django, que gestiona la lógica de la aplicación y la interacción con la base de datos. Django maneja las solicitudes del usuario, procesa los datos y actualiza la vista en consecuencia.

- **Modelo-Vista-VistaModelo (MVVM):** Angular, en el desarrollo del frontend, sigue el patrón MVVM, que permite una separación clara entre la lógica de la vista (View) y la lógica de presentación (ViewModel), manteniendo los datos y la lógica del negocio en el modelo (Model).
  En este patrón:

  - **Modelo (Model):** Similar al patrón MVC, el modelo contiene los datos y la lógica de negocio de la aplicación.
  - **Vista (View):** Representa la interfaz gráfica que interactúa con el usuario, mostrando la información al usuario y recibiendo sus interacciones.
  - **VistaModelo (ViewModel):** Este componente se comunica con la Vista a través de técnicas como el "data binding", enlazando automáticamente los datos del modelo con la interfaz gráfica. El ViewModel también puede manejar la lógica de presentación e interactúa con el modelo para obtener y actualizar los datos [35].

  La principal ventaja del patrón MVVM en Angular es la claridad que proporciona al código mediante la programación orientada a componentes, combinada con el uso del *data binding* y la técnica del *code-behind*, que separa la lógica de la interfaz de usuario del código subyacente. Esto mejora la mantenibilidad y legibilidad del código, facilitando la reutilización de componentes y la modularización de la aplicación.
  La inyección de dependencias también es una característica clave en Angular, que permite que los componentes, servicios y otras piezas de la aplicación sean gestionados y reutilizados de manera flexible, lo que simplifica tanto el desarrollo como las pruebas [26].

Además, en el desarrollo del frontend, se ha adoptado el patrón de diseño de **programación orientada a componentes**, especialmente utilizando Angular. Este patrón organiza la interfaz de usuario en componentes reutilizables e independientes, lo que facilita el mantenimiento y escalabilidad del sistema. Cada componente en Angular encapsula la lógica y el diseño, lo que permite un desarrollo modular y eficiente.

3.2 Conceptos Relacionados

- **Interfaz Web:** Una interfaz web es una estructura formada por elementos gráficos y componentes interactivos que permiten al usuario acceder y manipular la información de un sitio web. Su objetivo principal es proporcionar un entorno visual que facilite la comunicación entre el usuario y la aplicación [25].
- **Arquitectura de Aplicaciones Web:** La arquitectura de una aplicación web se divide en dos partes fundamentales:
  - **Frontend:** Es la parte de la aplicación con la que interactúan directamente los usuarios. Incluye todo el código necesario para la interfaz del sitio web, como la tipografía, los colores y los estilos. En este proyecto, el frontend se ha desarrollado con Angular, que es un framework de desarrollo de aplicaciones web dinámicas basado en componentes [26].
  - **Backend:** Es la capa que gestiona la lógica de la aplicación y el procesamiento de datos. En este proyecto, el backend se ha desarrollado utilizando Django y Django REST Framework, que se encargan de manejar las solicitudes, procesar datos y garantizar la seguridad del sitio. Django es un framework de Python que sigue el patrón MVC, y está diseñado para facilitar el desarrollo rápido y seguro de aplicaciones web [27].
- **Application Programming Interface (API):** Una API es un conjunto de reglas y protocolos que permite la comunicación entre diferentes programas. En esta plataforma, las APIs facilitan la gestión de usuarios, la creación de sesiones de lectura y la recopilación de datos para el análisis de actividades [34].
- **REST (Representational State Transfer):** REST es un estilo de arquitectura de software que define cómo realizar el intercambio y manejo de datos a través de servicios web. Basado en el protocolo HTTP, las APIs RESTful utilizan formatos estándar como JSON y métodos HTTP (GET, POST, PUT, DELETE) para la manipulación de recursos [14].

4 Métodos y herramientas

4.1 Métodos

- **Herramienta de gestión de proyectos**.Trello ha sido utilizado para planificar iteraciones y gestionar el proyecto, basándose en metodologías ágiles como SCRUM [31]. Su interfaz visual facilita la organización de tareas en tableros y listas, permitiendo una colaboración en tiempo real y un seguimiento eficiente de los sprints. Esto ha permitido implementar un enfoque incremental en el desarrollo, escalando y añadiendo nuevas funcionalidades de manera continua. La integración de Trello con SCRUM ha facilitado la gestión de iteraciones, así como el mantenimiento adaptativo y correctivo del proyecto [21].
- **Desarrollo incremental**. La plataforma se ha desarrollado de manera incremental, permitiendo iteraciones sucesivas y la incorporación gradual de funcionalidades. Este enfoque ha facilitado la validación continua del sistema y ajustes basados en la retroalimentación de los usuarios. Al utilizar metodologías ágiles y herramientas como Trello, el proceso de desarrollo ha sido más flexible y adaptativo, alineándose con los principios del desarrollo ágil y SCRUM.

4.2 Herramientas

Han sido diversas las herramientas utilizadas para el desarrollo del proyecto:

- **Lenguajes de programación**:
  - **Python**: lenguaje de programación orientado a objetos ampliamente utilizado por su semántica dinámica y estructuras de datos de alto nivel, ideal para desarrollo backend y procesamiento de datos, incluyendo IA [10].

  - **JavaScript/TypeScript**: utilizados para frontend con Angular, permitiendo experiencias de usuario dinámicas e interactivas en la web. Su popularidad se debe a su versatilidad y rapidez de ejecución [11].
  - **HTML**: lenguaje de marcado estándar para la creación de páginas web, fundamental para la estructura y contenido básico [12].
  - **CSS**: lenguaje de estilo para definir la presentación visual de documentos HTML, esencial para el diseño y experiencia del usuario en la web [13].

- **Frameworks y Bibliotecas**:
  - **Django REST Framework**: framework utilizado para la creación de APIs RESTful, proporcionando una base robusta para el backend.
  - **Angular**: framework para desarrollo de interfaces de usuario modernas e intuitivas.
  - **Bootstrap**: utilizado para diseño web adaptativo (responsive), ofreciendo componentes reutilizables y personalizables [15].

- **Bases de Datos**:
  - **MySQL**: sistema de gestión de bases de datos relacional, elegido por su robustez y capacidad para manejar datos estructurados de manera eficiente [18].
  - **DBeaver**: herramienta utilizada para la gestión y administración de la base de datos MySQL, facilitando la visualización, edición y consulta de datos de forma gráfica y eficiente.

- **Editor de texto**:
  - **Visual Studio Code**: editor de texto fuente con soporte para depuración, control de versiones, y múltiples lenguajes de programación, ampliamente utilizado durante el desarrollo del sistema [19].

- **Servidor**:
  - **Node.js**: entorno de ejecución para JavaScript orientado a eventos asíncronos, permitiendo operaciones en paralelo sin cambios de contexto [16].
  - **NPM**: sistema de gestión de paquetes para Node.js, facilitando la instalación y gestión de dependencias [17].

- **Control de versiones**:
  - **Git**: sistema de control de versiones para gestionar cambios en el código y facilitar la colaboración.
  - **GitHub**: plataforma basada en Git para alojar y gestionar código en la nube.
  - **Google drive**: herramienta de almacenamiento en la nube que permite la gestión y control de versiones de documentos ofimáticos desde múltiples dispositivos.

- **Herramientas CASE y gestión de proyectos**:
  - **Draw.io**: herramienta de modelado UML y diagramación utilizada para crear diagramas UML y otros tipos de diagramas, facilitando la documentación visual del proyecto [23].
  - **EZestimate**: herramienta de estimación de costos y tiempos de proyectos, utilizada para realizar estimaciones precisas y gestionar presupuestos de manera eficiente [7].
  - **Microsoft Project**: software de gestión de proyectos que permite planificar, ejecutar y hacer seguimiento a proyectos, facilitando la gestión de recursos y cronogramas [24].

5 Aspectos relevantes del desarrollo del proyecto

5.1 Marco de trabajo y metodología

En este proyecto se ha seguido el Proceso Unificado como marco de trabajo, que ofrece un enfoque iterativo e incremental. Esto permite que el desarrollo del sistema sea flexible y evolutivo, adaptándose a los cambios y mejorando a medida que avanza el proyecto. El Proceso Unificado se caracteriza por estar:

- **Conducido por casos de uso:** se basa en casos de uso para definir las funcionalidades del sistema.
- **Centrado en la arquitectura:** el diseño y la implementación se estructuran a partir de una arquitectura bien definida.
- **Iterativo e incremental:** el desarrollo se realiza en varias iteraciones, mejorando de forma constante con cada ciclo.

El Proceso Unificado se divide en cuatro fases principales, cada una con varias disciplinas que se realizan a lo largo de múltiples iteraciones:

- **Inicio.** Se desarrollan los casos de negocio y se define el alcance del proyecto.
- **Elaboración.** Se planifica el proyecto, se estiman los esfuerzos y se diseñan los casos de uso más importantes, así como la arquitectura del sistema.
- **Construcción.** Se implementa el sistema.
- **Transición.** Se ajusta el sistema a una versión funcional, corrigiendo problemas y refinando las características.

La Figura 1 presenta un diagrama que muestra las diferentes fases y disciplinas del Proceso Unificado, distribuidas a lo largo del ciclo de vida del proyecto.

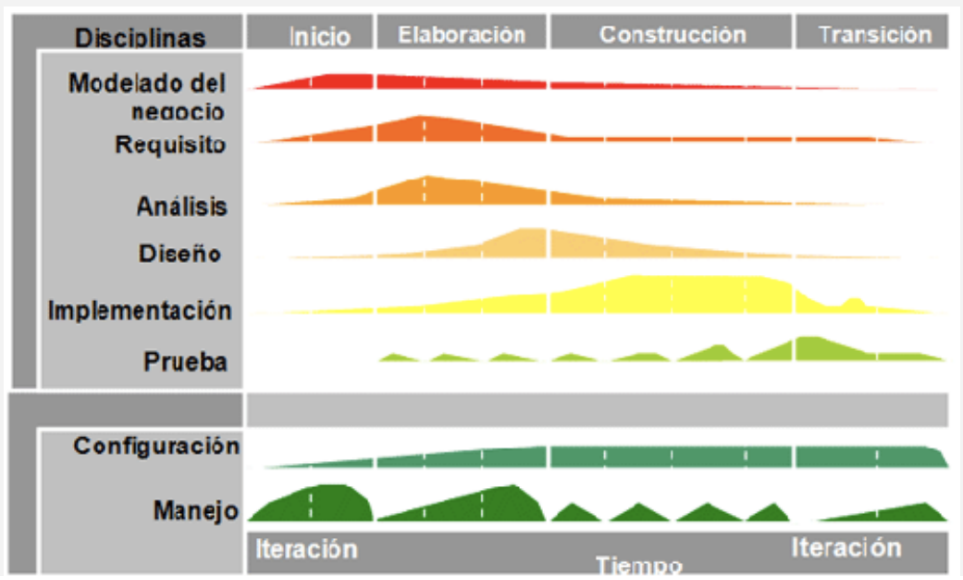


Figura 1: Diagrama del proceso unificado.

5.2 Estimación de la duración del proyecto

La estimación de la duración del proyecto se realiza una vez que se han definido los casos de uso, ya que estos son necesarios para su cálculo. Con esta estimación, es posible planificar el tiempo dedicado a cada una de las tareas del proyecto.

Para obtener información más detallada sobre la estimación de la duración, consulte el **Anexo I: Plan del Proyecto**. Los parámetros introducidos y la estimación obtenida se pueden observar en la Figura 2 .

La estimación del tiempo necesario para el desarrollo del proyecto se ha llevado a cabo utilizando la herramienta **EZEstimate**. Para obtener esta estimación, se deben introducir en la aplicación los factores que influyen en el desarrollo, como la complejidad técnica y del entorno, así como la valoración de la complejidad de los actores y los requisitos funcionales[7].

La estimación temporal calculada para el proyecto es de aproximadamente 758,52 horas de trabajo. Esta cifra se basa en el análisis de los casos de uso y otros factores definidos en los apartados previos.

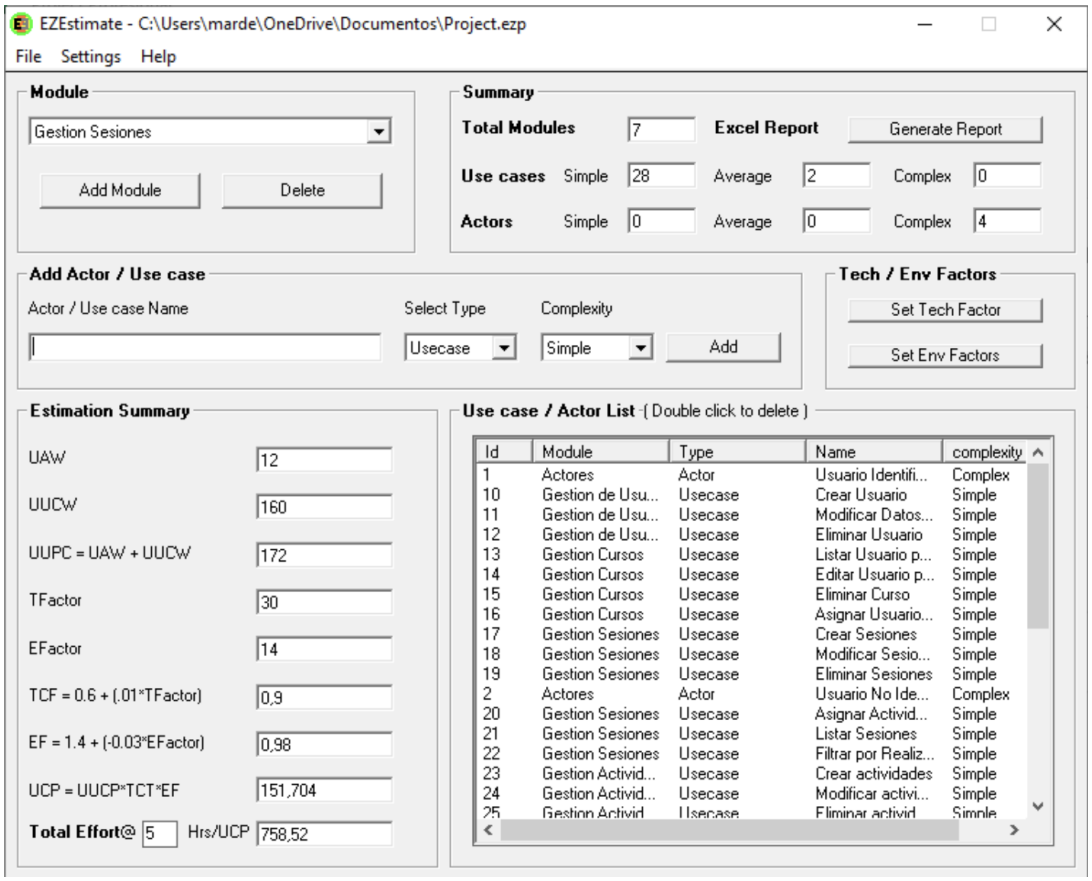


Figura 2: Estimación temporal.

Para interpretar mejor la duración del proyecto en términos prácticos, consideramos una jornada laboral estándar de 8 horas diarias. Según esta premisa, el proyecto requeriría aproximadamente 95 días laborables. Esto se traduce en un período de aproximadamente 3 meses y 5 días laborables. En términos de días naturales, esto equivale a cerca de 4 meses y 3 días naturales

5.3 Planificación temporal

La planificación temporal del proyecto se ha llevado a cabo después de obtener una estimación inicial de la duración del desarrollo, la cual se calcula en base a las tareas y recursos necesarios para completar el ciclo de vida del sistema. Esta planificación permite identificar de manera clara todas las tareas que deben realizarse, así como los recursos y tiempo necesarios para cada una de ellas. Para el desarrollo del sistema se ha utilizado el marco de trabajo del Proceso Unificado, lo que ha permitido organizar las tareas en diferentes disciplinas asociadas

a cada fase.

Las disciplinas del Proceso Unificado incluyen:

- **Modelo de negocio:** se realiza una investigación y análisis de los recursos necesarios para llevar a cabo el proyecto.
- **Requisitos:** se establecen los objetivos y requisitos que el proyecto debe cumplir.
- **Análisis:** se lleva a cabo el análisis de los requisitos previamente definidos.
- **Diseño:** se define el funcionamiento y la estructura de los diferentes componentes del sistema.
- **Implementación:** se desarrolla el código del sistema según el diseño planteado.
- **Pruebas:** se realizan pruebas unitarias e integrales para verificar el correcto funcionamiento del sistema y corregir posibles errores.

En la Figura 3, se presenta un diagrama de Gantt que proporciona una vista global de la planificación temporal del proyecto, facilitando la comprensión de la distribución de tareas y la secuencia temporal en la que se llevarán a cabo.

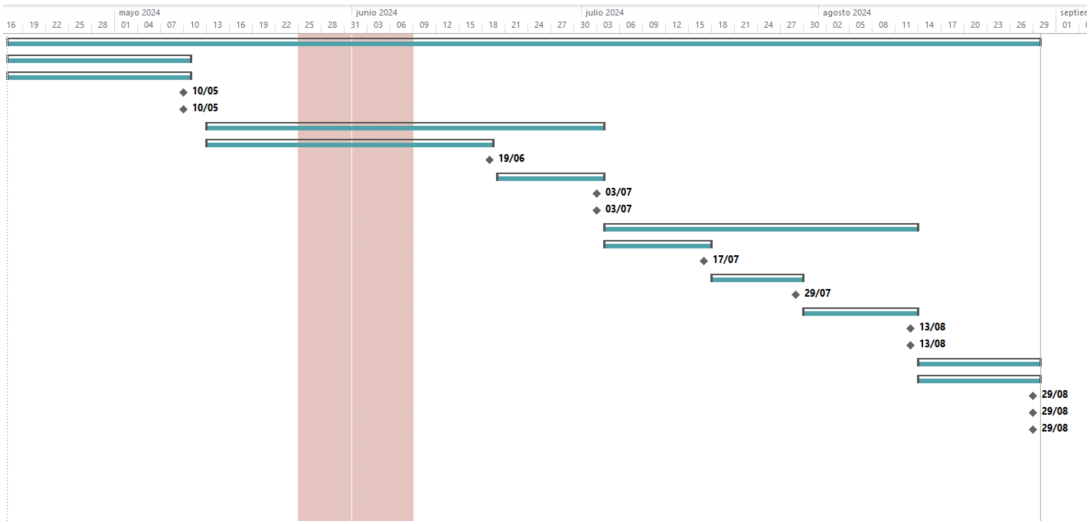


Figura 3: Diagrama de Gantt de la vista global de la planificación temporal.

El desarrollo del sistema se ha dividido en diferentes fases, según las especificaciones del *Proceso Unificado*. Este proceso es incremental e iterativo, lo que implica que cada fase se compone de un número determinado de iteraciones. Para este proyecto, las fases y sus iteraciones se han distribuido de la siguiente manera:

- **Inicio:** 1 iteraciones

- **Elaboración:** 2 iteraciones
- **Construcción:** 3 iteraciones
- **Transición:** 1 iteraciones

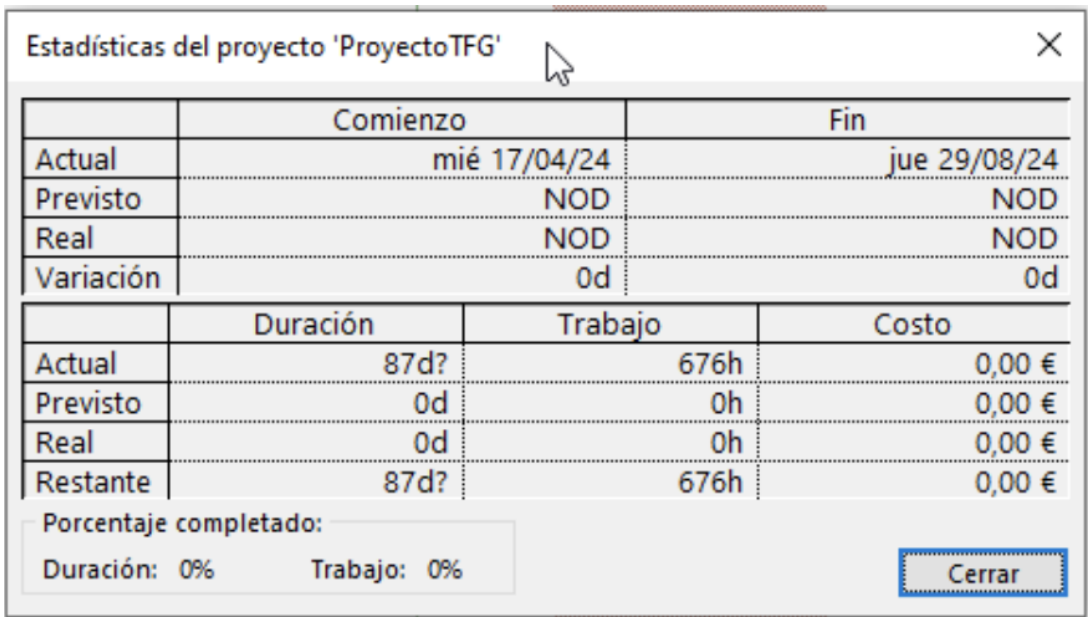
Modo de	Nombre de tarea	Duración	Comienzo	Fin
	Proyecto	**87 días**	**mié 17/04/24**	**jue 29/08/24**
	Inicio	**18 días**	**mié 17/04/24**	**vie 10/05/24**
	Iteración 1	**18 días**	**mié 17/04/24**	**vie 10/05/24**
	Fin Iteración I	0 días	vie 10/05/24	vie 10/05/24
	Fin Inicio	0 días	vie 10/05/24	vie 10/05/24
	Elaboración	**28 días**	**lun 13/05/24**	**mié 03/07/24**
	Iteración I	**17,5 días**	**lun 13/05/24**	**mié 19/06/24**
	Fin Iteración I	0 días	mié 19/06/24	mié 19/06/24
	Iteración II	**10 días**	**jue 20/06/24**	**mié 03/07/24**
	Fin Iteración II	0 días	mié 03/07/24	mié 03/07/24
	Fin Elaboración	0 días	mié 03/07/24	mié 03/07/24
	Construcción	**29 días**	**jue 04/07/24**	**mar 13/08/24**
	Iteración I	**10 días**	**jue 04/07/24**	**mié 17/07/24**
	Fin Iteración I	0 días	mié 17/07/24	mié 17/07/24
	Iteración II	**8 días**	**jue 18/07/24**	**lun 29/07/24**
	Fin Iteración II	0 días	lun 29/07/24	lun 29/07/24
	Iteración III	**11 días**	**mar 30/07/24**	**mar 13/08/24**
	Fin Iteración III	0 días	mar 13/08/24	mar 13/08/24
	Fin contrucción	0 días	mar 13/08/24	mar 13/08/24
	Transición	**12 días**	**mié 14/08/24**	**jue 29/08/24**
	Iteración I	**12 días**	**mié 14/08/24**	**jue 29/08/24**
	Fin Iteración I	0 días	jue 29/08/24	jue 29/08/24
	Fin Transición	0 días	jue 29/08/24	jue 29/08/24
	Fin Planificación Temporal	0 días	jue 29/08/24	jue 29/08/24

Figura 4: Planificación global.

La planificación temporal también desempeña un papel crucial en la descomposición del proyecto en elementos más manejables, permitiendo una constante identificación de tareas completadas y una estimación precisa de la duración total del proyecto [8]. Esta etapa se ejecuta al comienzo del proyecto, brindando una visión clara de las tareas que deben realizarse en cada fase y permitiendo la supervisión continua del cumplimiento de los plazos establecidos. De esta manera, se evita cualquier posible retraso en la ejecución del proyecto.

Para facilitar la planificación, se ha utilizado la herramienta **Microsoft Project**, Figura 4, con la cual se ha configurado un calendario de trabajo ajustado a las necesidades del proyecto. En este caso, el calendario comprende una jornada laboral de 8 horas diarias, dividida en dos periodos: de 9:00 a 14:00 y de 15:00 a 18:00, de lunes a viernes. Además, se han definido una semana sin realizar el proyecto debido a exámenes.

En conclusión, los resultados de la Figura 5 indican una duración total de aproximadamente 87 días de trabajo, lo cual está en línea con las estimaciones previas que eran en torno a 95 días. Este enfoque estratégico asegura la alineación con las estimaciones iniciales y establece una base sólida para futuras implementaciones y ajustes en el proceso de desarrollo del proyecto.



Estadísticas del proyecto 'ProyectoTFG'

	Comienzo	Fin
Actual	mié 17/04/24	jue 29/08/24
Previsto	NOD	NOD
Real	NOD	NOD
Variación	0d	0d

	Duración	Trabajo	Costo
Actual	87d?	676h	0,00 €
Previsto	0d	0h	0,00 €
Real	0d	0h	0,00 €
Restante	87d?	676h	0,00 €

Porcentaje completado:

Duración: 0% Trabajo: 0%

Cerrar

Figura 5: Estadísticas de la planificación temporal.

5.4 Especificación de requisitos

En esta sección se detallan los requisitos del software para el sistema en desarrollo. Para la elaboración de estos, se sigue la metodología propuesta por Durán y Bernárdez[2]. Esta metodología sirve de base para las siguientes fases del ciclo de vida del sistema, ya que define claramente las necesidades del proyecto.

Para obtener información más detallada sobre la especificación de requisitos, consulte el **Anexo II: Especificación de Requisitos del Sistema**. En los apartados siguientes se presentan ejemplos que ilustran cada etapa del proceso de especificación de requisitos.

5.4.1 Participantes

En el desarrollo del proyecto participan los siguientes integrantes:

- **Marta de Castro Leira**
- **Jesús Manuel Nieto Carracedo**
- **Ángelica González Arrieta**

A continuación, se muestra un ejemplo de la especificación de participantes.

Campo	Valor
Participante	Marta de Castro Leira
Organización	Universidad de Salamanca (USAL)
Rol	Desarrolladora
Es desarrollador	Sí
Es cliente	No
Es usuario	Sí
Comentarios	-

Tabla 1: Especificación de participantes.

5.4.2 Objetivos del Sistema

Los objetivos identificados para resolver las necesidades planteadas del sistema son los siguientes:

- **Gestión de Actividades**
  - **OBJ-0001**: crear y editar actividades de lectura, y almacenar los resultados y métricas asociadas para su análisis externo.
  - **OBJ-0004**: permitir la asignación de actividades de lectura a los usuarios y gestionar su progreso en estas actividades.
- **Gestión de Usuarios**
  - **OBJ-0002**: almacenar y gestionar la información básica de los usuarios, como ID, nombre, email y otros datos relevantes.
  - **OBJ-0003**: gestionar un historial de sesiones de trabajo de los usuarios, incluyendo ejercicios de lectura.
- **Gestión de Resultados y Métricas**
  - **OBJ-0005**: obtener y almacenar los resultados y las tareas importadas por terceros para analizar estos resultados.
- **Gestión de Sesiones**
  - **OBJ-0006**: administrar las sesiones asignadas a los usuarios, incluyendo la creación, seguimiento y actualización del estado de las tareas.
- **Interfaz de Usuario**
  - **OBJ-0007**: diseñar una interfaz de usuario intuitiva y amigable que permita una navegación eficiente y fácil acceso a las funcionalidades del sistema.
- **Gestión de Cursos**
  - **OBJ-0008**: administrar los cursos asignados a los usuarios, incluyendo la creación, seguimiento y actualización.

A continuación se muestra un ejemplo de la especificación de objetivos.

OBJ-0001	Gestión de Actividades
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Crear y editar actividades de lectura y almacenar los resultados y métricas asociadas para su análisis externo.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 2: OBJ-0001 Gestión de actividades.

5.4.3 Requisitos de Información

Los requisitos de información definen los datos que el sistema debe manejar y almacenar. Los requisitos identificados son:

- Información sobre Usuarios
- Información sobre Cursos
- Información sobre Entrenamientos
- Información sobre Actividades
- Información sobre Resultados
- Información sobre Items

A continuación, se muestra un ejemplo de la especificación de requisitos de información.

IRQ-001	Información sobre Usuarios
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0002] Gestión de Usuarios
Descripción	El sistema debe almacenar la información básica de los usuarios registrados en la plataforma, incluyendo su identificador único, nombre, credenciales y datos relacionados con su cuenta de Glifing.
Datos específicos	Identificador único (ID) Usuario: formato nombre.apellido Usuario Glifing: Nombre de usuario en Glifing Nombre Apellidos Correo electrónico Rol: entrenador, entrenado, administrador Password: Clave de acceso cifrada Avatar: Imagen asociada al usuario
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 3: IRQ-001: Información sobre usuarios.

5.4.4 Requisitos Funcionales

Los requisitos funcionales describen las funcionalidades específicas que el sistema debe ofrecer para cumplir así los objetivos establecidos en el proyecto. Estos se han agrupado en diferentes paquetes según su funcionalidad. Esta división se puede observar en el diagrama de paquetes:

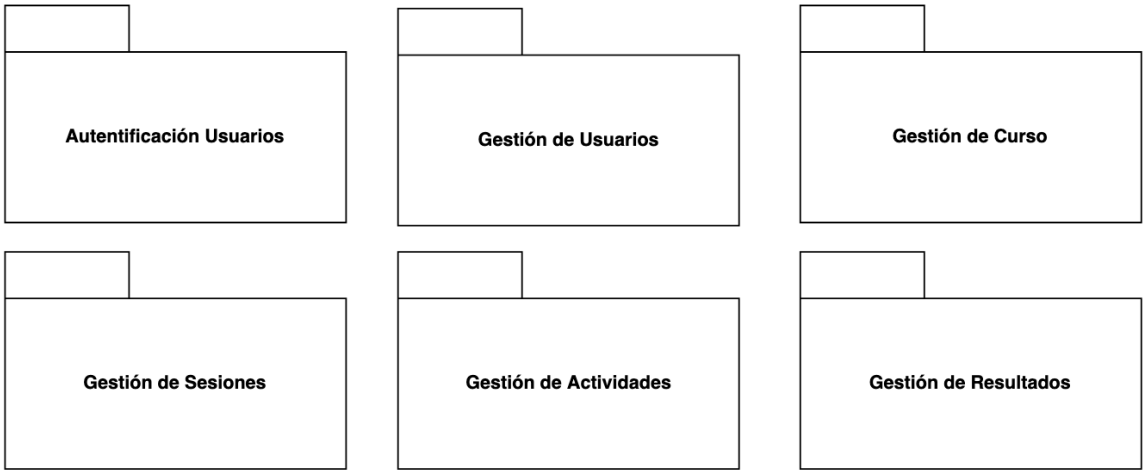


Figura 6: Diagrama de paquetes.

- **Gestión de Usuarios**
  - Funcionalidades para que los usuarios puedan modificar y actualizar sus propios datos, como información de contacto, contraseña, y avatar.
- **Autenticación de Usuarios**
  - Requisitos para la autenticación segura de usuarios, incluyendo la verificación de credenciales y la gestión de sesiones.
  - Implementación de mecanismos de autenticación de múltiples factores si es necesario.
- **Gestión de Cursos**
  - Requisitos para la administración de cursos asignados a los usuarios, incluyendo la creación, seguimiento y actualización de los cursos.
  - Funcionalidades para asociar actividades y sesiones a los cursos y gestionar el progreso del usuario en estos cursos.
- **Gestión de Sesiones**
  - Requisitos para la administración de sesiones asignadas a los usuarios, incluyendo la creación, seguimiento y actualización del estado de las tareas.
  - Funcionalidades para gestionar el historial de sesiones y las actividades realizadas durante cada sesión.
- **Gestión de Resultados**
  - Requisitos para la administración de resultados de las sesiones de entrenamiento.

- **Gestión de Actividades**
  - Requisitos para la creación, edición y gestión de actividades relacionadas con el procesamiento del lenguaje natural.
  - Asignación y seguimiento de tareas para los usuarios (participantes) dentro del sistema.
  - Funcionalidades para la gestión de paquetes de tareas y su asociación con usuarios específicos.

A continuación se detallan los actores del sistema y ejemplos de casos de uso.

Figura 7: Actores.

- **Actores del Sistema:**
  - Usuario No Identificado
  - Usuario Identificado
  - Participante
  - Entrenador
  - Administrador

Finalmente, se definen y detallan los casos de uso para ilustrar el comportamiento del sistema. Los diagramas de casos de uso se han creado de forma independiente para cada uno de los paquetes previamente indicados. A continuación se presenta un ejemplo de diagrama de casos de uso.

ACT-01	Usuario no identificado
Versión	1.0
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Usuario que interactúa con el sistema sin haberse identificado o iniciado sesión.
Comentarios	Tiene acceso limitado a las funcionalidades del sistema y no puede realizar tareas administrativas o de edición.

Tabla 4: ACT-01 Usuario no identificado.

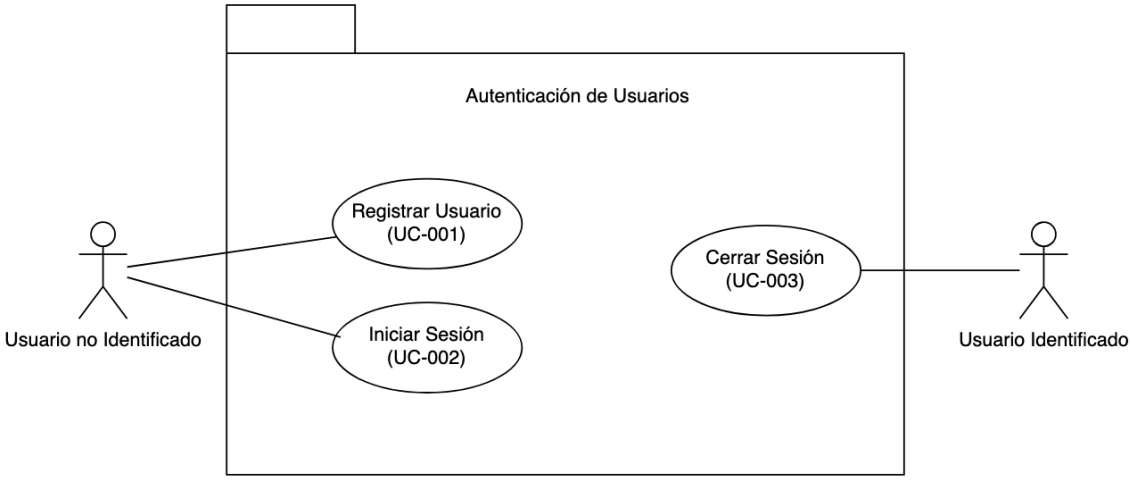


Figura 8: Autentificación de usuarios.

A continuación, se muestra un ejemplo de la especificación de un caso de uso.

UC-001	**Registrar Usuario**
Versión	1.0 (03/07/2023)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El sistema solicita al usuario que introduzca sus datos personales y posteriormente se crea su cuenta.
Precondición	El usuario no ha iniciado sesión y no deben tener cuenta creada.
Secuencia Normal	1. El actor Usuario No Identificado solicita el registro de su cuenta. 2. El sistema solicita al usuario que introduzca sus datos personales. 3. El actor Usuario No Identificado introduce los datos requeridos. 4. El sistema comprueba que los datos introducidos presentan el formato correcto, que se han introducido todos los obligatorios, y que el nombre del usuario no existe.
Postcondición	El usuario se crea en el sistema con los datos introducidos, y por defecto como Usuario logeado [ACT-002]
Excepciones	3. El actor Usuario No Identificado cancela el registro, este caso de uso queda sin efecto. 4. Los datos introducidos son incorrectos o los campos nombre de usuario y/o correo electrónico ya están registrados en el sistema, el sistema devuelve un mensaje de error y el caso de uso queda sin efecto.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 5: UC-001: Registrar usuario.

5.4.5 Requisitos No Funcionales

Los requisitos no funcionales establecen las características y propiedades del sistema que no están directamente relacionadas con la funcionalidad, pero que son esenciales para garantizar su calidad y eficiencia. Estos requisitos se centran en aspectos como la usabilidad, rendimiento, seguridad y otros atributos importantes para el sistema.

En nuestro proyecto de procesamiento de lenguaje natural, los requisitos no funcionales identificados son:

- Usabilidad
- Compatibilidad
- Fiabilidad
- Implementación
- Privacidad

A continuación, se muestra un ejemplo de la especificación de un requisito no funcional.

NFR-001	**Usabilidad**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	Ninguna
Descripción	El sistema debe presentar una interfaz intuitiva y amigable, permitiendo a los usuarios llevar a cabo sus tareas con facilidad y sin necesidad de formación extensiva.
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 6: NFR-001: Usabilidad.

5.5 Análisis del sistema

El análisis de requisitos es una etapa fundamental en el desarrollo de sistemas, dedicada a examinar, refinar y organizar los requisitos del software. Este proceso busca entender de manera exhaustiva las expectativas y necesidades del sistema, utilizando modelos conceptuales como el diagrama de clases para representar las entidades y sus relaciones. Se incluye el **Anexo II: Análisis del sistema Software**, para más detalles sobre este proceso y su aplicación, en el que se proporciona información adicional y técnicas específicas del análisis orientado a objetos.

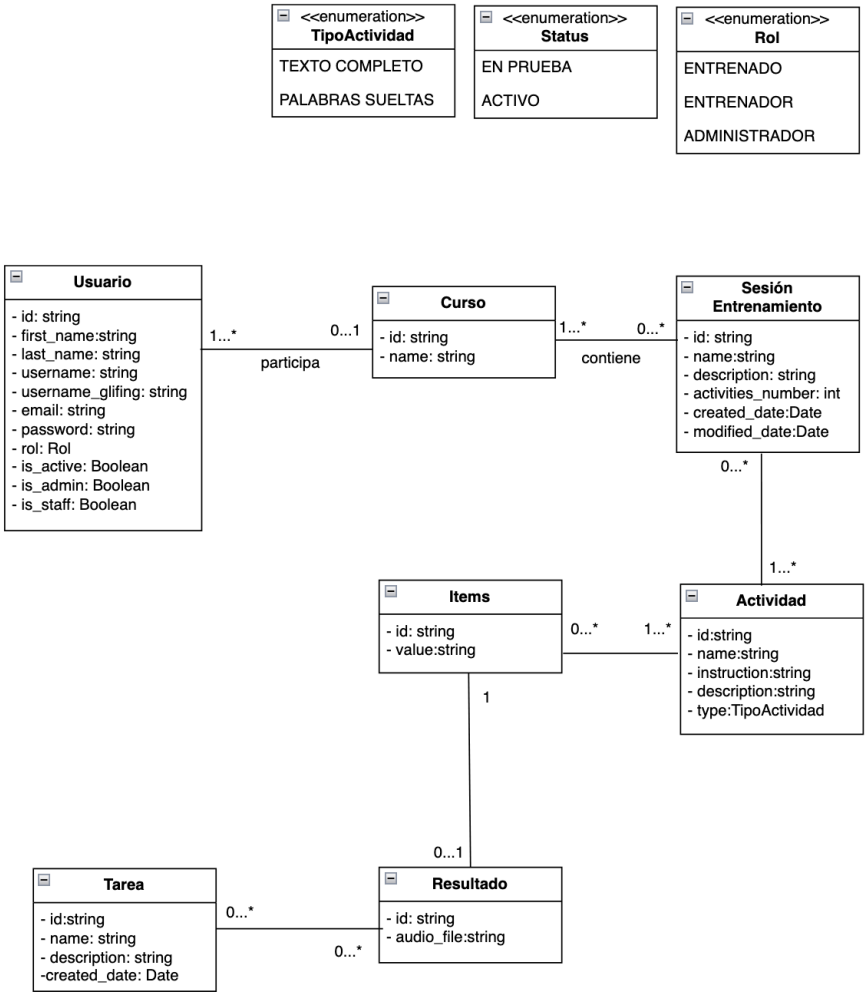


Figura 9: Diagrama de dominio.

5.5.1 Modelo de dominio

El modelo del dominio Figura 9 es una representación de la estructura de los conceptos y entidades clave relaciones con el problema que aborda la plataforma a desarrollar. Este se usa para comprender y representar los requisitos y la lógica del negocio del sistema.

5.5.2 Realización de casos de uso

A partir del diagrama de entidad-relación y considerando las relaciones entre los diversos objetos, se detalló la funcionalidad de los subsistemas mediante los diagramas de secuencia correspondientes a los casos de uso del sistema.

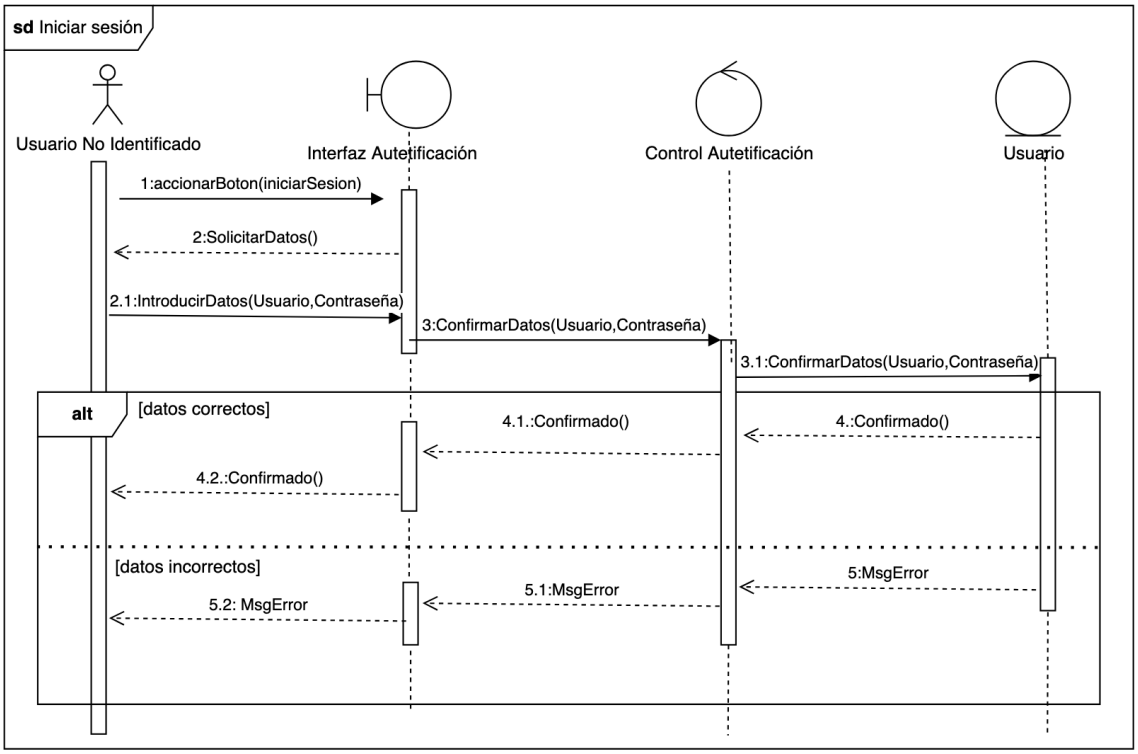


Figura 10: Diagrama de secuencia de iniciar sesión.

5.5.3 Clases de Análisis y Descripción de la Arquitectura

Una clase de diseño es un tipo de clasificador que agrupa objetos con especificaciones, restricciones y significados similares. En el modelo de análisis, se distinguen tres categorías principales de clases:

- **Clases de Entidad**: representan objetos de negocio que son persistentes y se almacenan en el sistema.
- **Clases de Control**: se encargan de gestionar y procesar la información dentro del sistema.
- **Clases de Interfaz**: son responsables de presentar la información al usuario y de facilitar la comunicación con los distintos componentes del sistema.

A continuación se muestra la arquitectura del modelo de análisis:

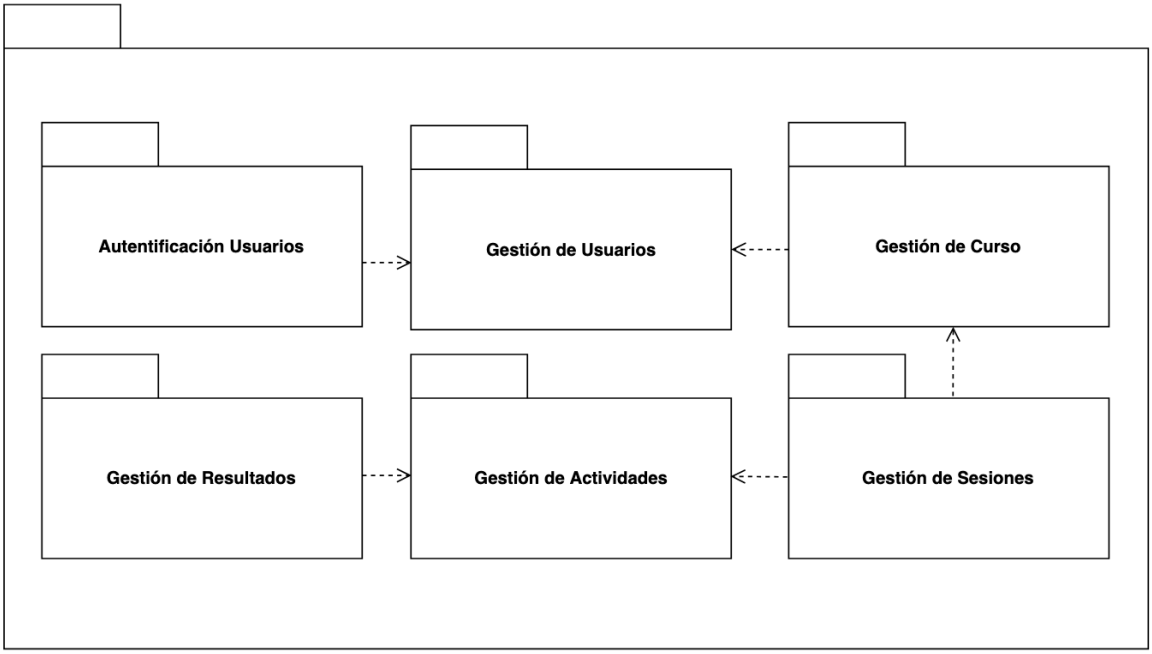


Figura 11: Diagrama de paquetes de análisis.

5.6 Diseño del sistema

En esta sección se presenta el diseño orientado a objetos del sistema propuesto, describiendo cómo se materializan los casos de uso. Este diseño es una aproximación a la implementación real, por lo que los nombres de las clases, métodos y atributos se ajustarán a los detalles específicos del desarrollo. Para más información detallada sobre el diseño del sistema, consulte el **Anexo IV: Diseño del sistema software**.

5.6.1 Patrón arquitectónico

El sistema está estructurado bajo el patrón arquitectónico cliente-servidor representado en la Figura 12, donde se divide principalmente en dos subsistemas: el frontend y el backend.

- **Frontend:** Implementado con Angular, este subsistema es responsable de la interacción con los usuarios. Sigue el patrón Modelo-Vista-ViewModel (MVVM) para facilitar la separación entre la lógica de presentación y la lógica de negocio. Cada componente dentro del frontend gestiona una parte específica de la interfaz de usuario, mientras que los servicios facilitan la comunicación con el backend a través de la API RESTful.
- **Backend:** Desarrollado con Django, el backend maneja la lógica de negocio y expone una API RESTful para la interacción con el frontend. Sigue el patrón Modelo-Vista-Controlador (MVC), donde los modelos representan la estructura de los datos y gestionan su persistencia en la base de datos, las vistas procesan las solicitudes HTTP y los controladores coordinan la interacción con los modelos.

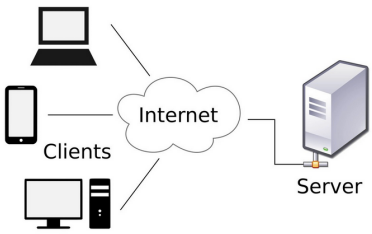


Figura 12: Patrón cliente-servidor.

Este enfoque asegura una clara separación de responsabilidades, permitiendo que cada subsistema evolucione de manera independiente, manteniendo la compatibilidad a través de la API.

5.6.2 Subsistemas de diseño

El diseño del sistema se ha dividido en dos subsistemas principales: el frontend y el backend. Esta separación permite una gestión más eficiente del desarrollo y facilita la escalabilidad del sistema. A continuación en la Figura 13 se presenta un diagrama que ilustra la arquitectura general de estos subsistemas.

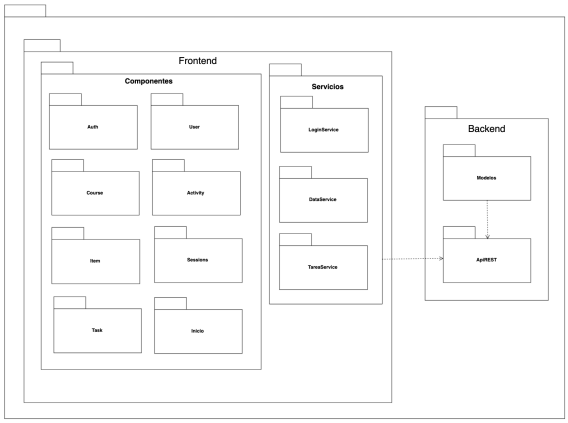


Figura 13: Subsistema de diseño.

5.6.3 Clases de diseño

Las clases de diseño representan la implementación concreta de los casos de uso del sistema, resultando en un conjunto de clases que se agrupan dentro de los subsistemas mencionados. A continuación se destacan algunas de las clases:

- **Clase ActividadComponent (Frontend):** Responsable de gestionar las actividades dentro de las sesiones de lectura. Esta clase interactúa con los servicios para obtener los datos necesarios desde el backend y mostrar la información relevante al usuario.
- **Clase AuthComponent (Frontend):** Contiene los componentes para la autenticación y el registro de usuarios. Gestiona el acceso a los recursos del sistema, permitiendo que los usuarios inicien sesión y se registren en la plataforma.
- **Modelos Django (Backend):** Las clases del backend, como las relacionadas con las actividades y usuarios, definen la estructura de los datos y su persistencia en la base de datos. Estas clases interactúan con las vistas y los controladores para realizar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos.

Este diseño modular y orientado a objetos no solo facilita la comprensión y mantenibilidad del sistema, sino que también promueve la escalabilidad y la facilidad para realizar futuras ampliaciones y mejoras en la plataforma.

5.6.4 Realización de Casos de Uso

Los diagramas de secuencia se utilizan para ilustrar con mayor detalle las interacciones y mensajes entre los objetos del sistema, en comparación con los diagramas realizados durante la fase de análisis. Un ejemplo de esto es el diagrama de secuencia para el caso de uso "Registrar Usuario" de la Figura 14.

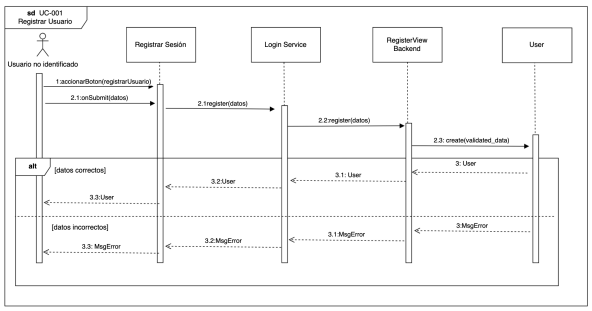


Figura 14: Diagrama de secuencia: registrar usuario.

5.6.5 Base de Datos

Aquí en la Figura 15 se muestran las relaciones entre las distintas tablas de mi base de datos.

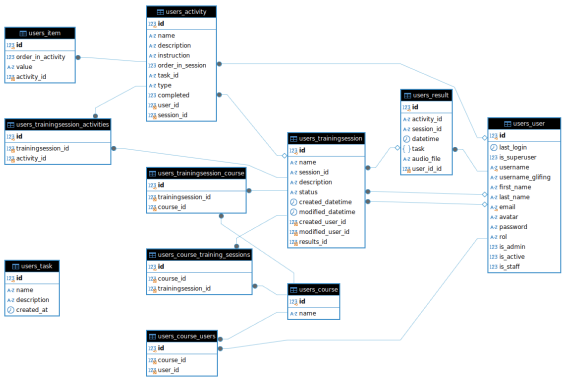


Figura 15: Base de datos.

5.7 Funcionalidad del sistema

En esta sección se ofrece una descripción general del funcionamiento de la aplicación web del sistema propuesto, con el objetivo de facilitar su comprensión y uso por parte del usuario. Se destacan las principales funcionalidades del sistema que están a disposición del usuario, teniendo en cuenta que existen 5 roles distintos con diferentes opciones: Usuario No Identificado, Usuario Identificado, Participante, Entrenador y Administrador. La información está organizada en función de estos roles para mayor claridad. Para una guía más exhaustiva sobre el uso del sistema, consulte el *Anexo VI – Manual de Usuario.*

5.7.1 Usuario No Identificado

Un usuario no identificado tiene acceso restringido y limitado. Este tipo de usuario puede:

- **Iniciar Sesión**: ingresar sus credenciales para autenticar su cuenta.
- **Registro**: completar un formulario para crear una nueva cuenta.

La Figura 16 muestra la vista general para un usuario no identificado al acceder a la página web.

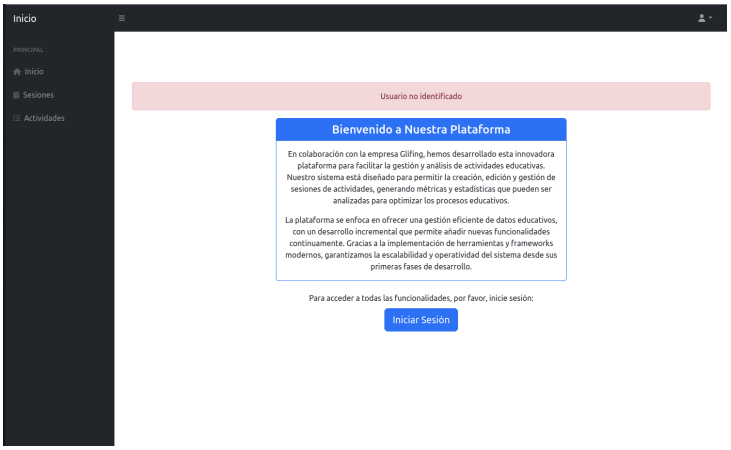


Figura 16: Vista de inicio de Usuario No Identificado.

Si intenta acceder a "Actividades" o "Sesiones", será redirigido a la página de inicio de sesión, ya que estas secciones requieren autenticación.

Iniciar Sesión

La interfaz de "Iniciar Sesión", Figura 17, permite al usuario ingresar su nombre de usuario y contraseña. La siguiente imagen muestra la vista general de esta interfaz:

Figura 17: Vista general de la pantalla de inicio de sesión.

Registro

La interfaz de "Registro", Figura 18, permite crear una nueva cuenta introduciendo los datos requeridos. La siguiente imagen muestra la vista de la pantalla de registro:

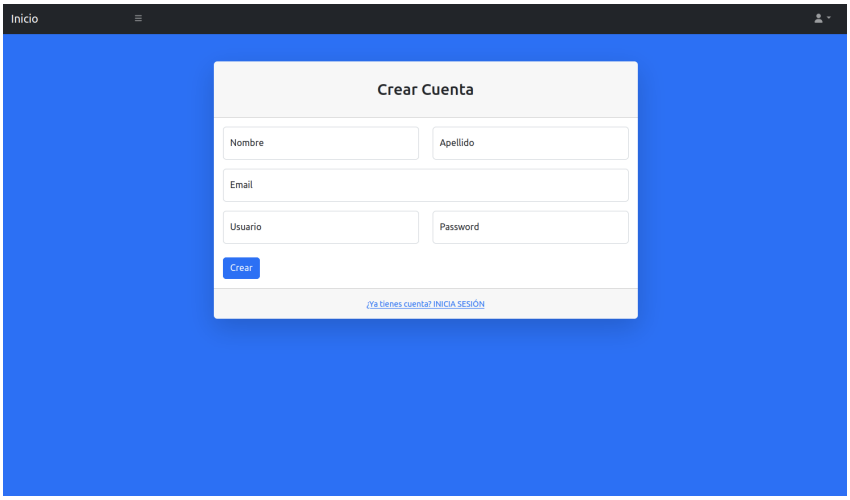


Figura 18: Vista de la pantalla de registro.

5.7.2 Usuario Identificado

Un usuario identificado tiene acceso a funcionalidades adicionales después de autenticarse. Al acceder a la página web, Figura 19, se muestra la vista inicial para el usuario identificado.

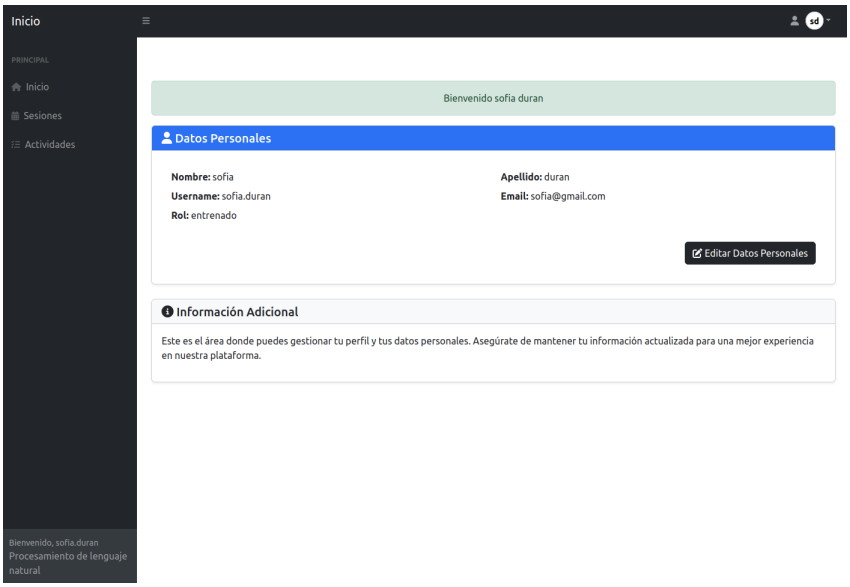


Figura 19: Vista del inicio del usuario identificado.

Las funcionalidades disponibles para un usuario identificado incluyen:

- **Gestión de Perfil**: permite visualizar y actualizar los datos personales, cambiar la foto de perfil y modificar la contraseña.
- **Cierre de Sesión**: permite finalizar la sesión de manera segura para proteger la cuenta del usuario.

Gestión de Perfil

En la sección de Gestión de Perfil, el usuario puede actualizar su información personal. La siguiente imagen muestra el formulario de modificación del perfil:

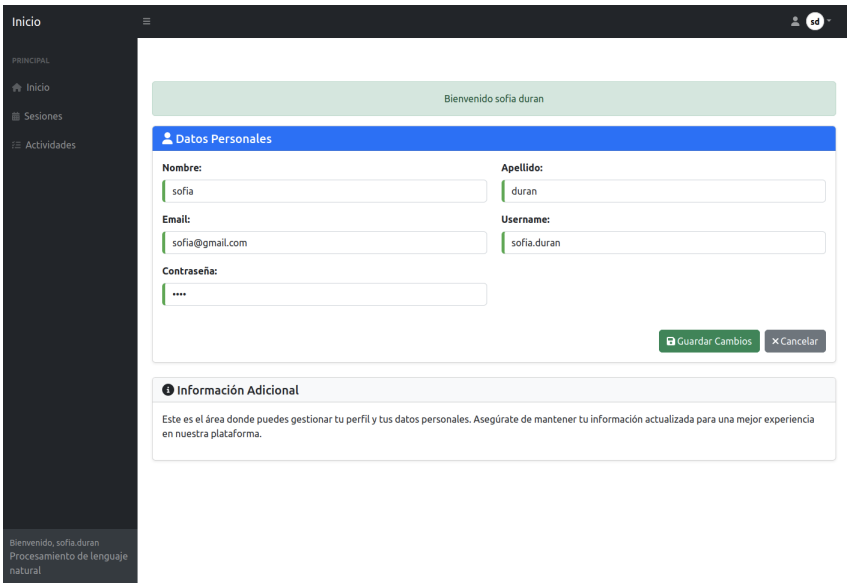


Figura 20: Formulario de modificación del perfil.

Cierre de Sesión

Para cerrar la sesión de manera segura, el usuario debe hacer clic en el botón "Cerrar Sesión", ubicado en la parte superior derecha de la pantalla. La Figura 21 muestra este botón.

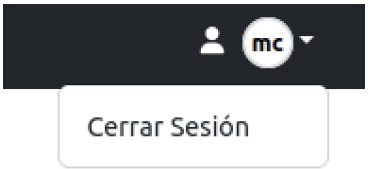


Figura 21: Botón para cerrar la sesión.

5.7.3 Participante

Un usuario con el rol de Participante tiene acceso a funcionalidades específicas para interactuar con sus entrenamientos asignados. Las principales funcionalidades disponibles son:

- **Gestión de Actividades**: permite visualizar las actividades asignadas y las sesiones a las que pertenecen. Esta funcionalidad se accede a través de la sección de actividades en la barra lateral.
- **Gestión de Sesiones**: permite visualizar y gestionar las sesiones en las que el usuario está involucrado, diferenciando entre sesiones con actividades finalizadas y pendientes.

Gestión de Actividades

En la sección de Gestión de Actividades, el usuario puede ver las actividades asignadas y las sesiones correspondientes tal y como se muestra en la Figura 22.

Figura 22: Vista del apartado de actividades por el usuario.

Gestión de Sesiones

En la sección de Gestión de Sesiones, Figura 23, el usuario puede observar las sesiones a las que pertenece y distinguir entre las finalizadas y las no finalizadas. En ella se ilustran las sesiones en distintos estados.

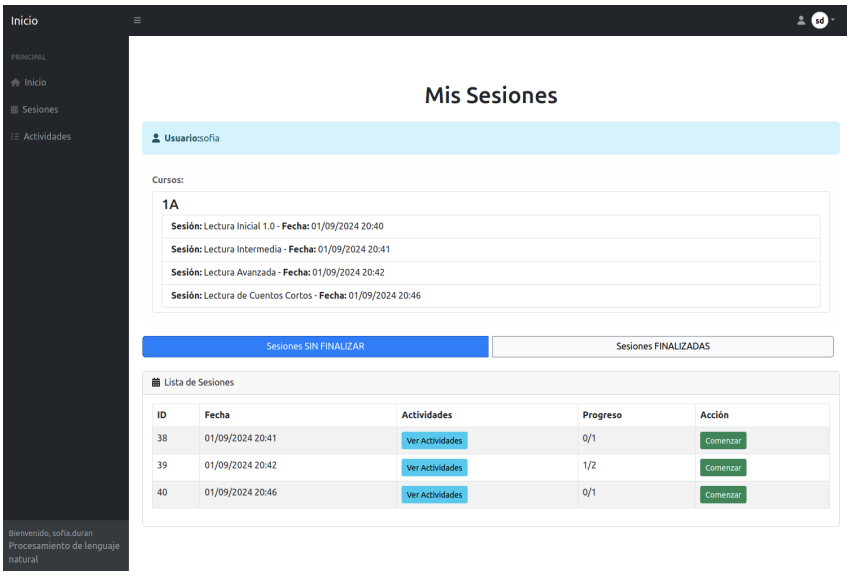


Figura 23: Vista de las sesiones sin finalizar.

Para las sesiones no finalizadas, el usuario puede comenzar las actividades, que pueden ser de tipo *texto completo* o *palabras sueltas*. La Figura 24 muestra la vista de grabación para las actividades de entrenamiento.

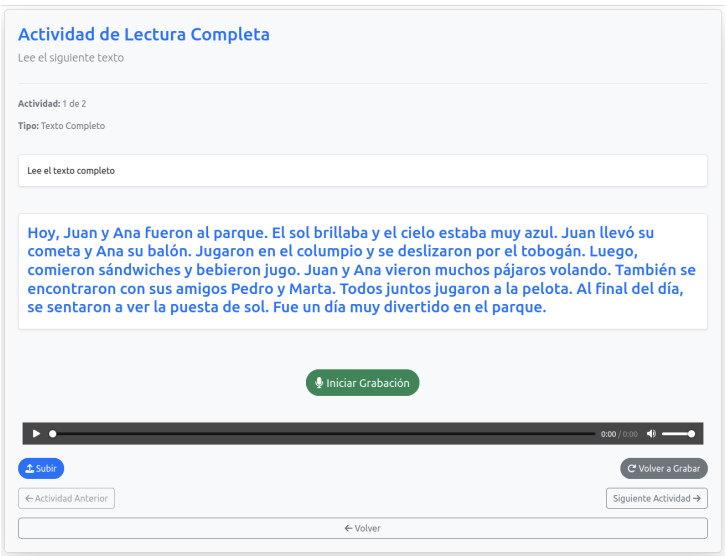


Figura 24: Vista de grabación de entrenamiento con texto.

Cada vez que se sube un archivo de audio, el sistema solicita confirmación para proceder, ya que no se puede modificar una vez subido. El usuario también puede abandonar el entrenamiento en cualquier momento.

5.7.4 Entrenador

Los entrenadores tienen acceso a funcionalidades avanzadas que les permiten gestionar sesiones de entrenamiento, actividades, cursos, ítems y resultados. A continuación se resumen las principales funcionalidades disponibles para este rol:

- **Gestión de Sesiones de Entrenamiento**: crear, modificar y eliminar sesiones de entrenamiento, y asignar cursos y actividades a cada sesión.
- **Gestión de Actividades**: crear, editar y eliminar actividades dentro de los cursos.
- **Gestión de Cursos**: crear, editar y eliminar cursos, y gestionar la asignación de usuarios a estos cursos.
- **Gestión de Ítems**: crear ítems y vincularlos con las actividades correspondientes.
- **Gestión de Resultados**: crear y eliminar tareas, y analizar archivos de audio asociados a estas tareas.

Gestión de Sesiones de Entrenamiento

El entrenador puede gestionar las sesiones de entrenamiento, incluyendo la creación y edición de sesiones así como la asignación de cursos y actividades. La Figura 25 muestra la vista de la gestión de sesiones de entrenamiento.

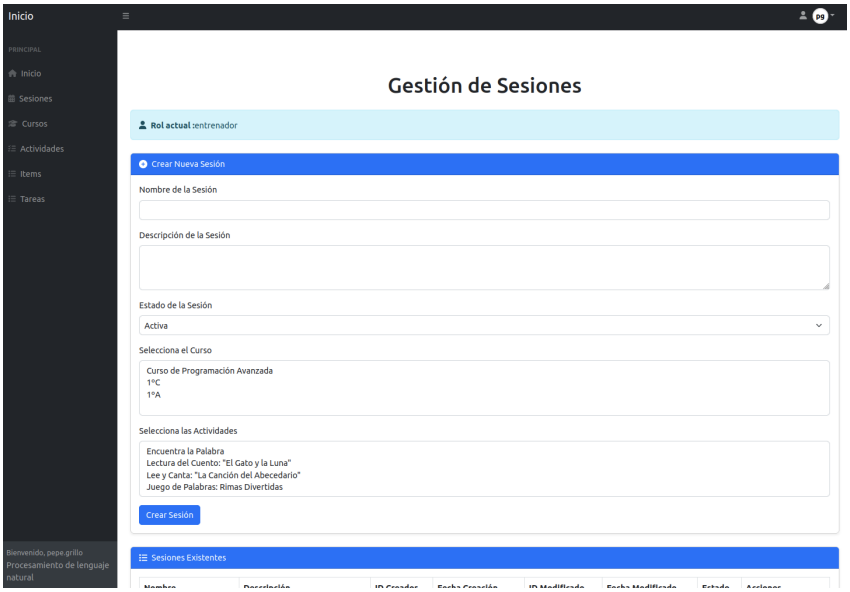


Figura 25: Vista de gestión de sesiones de entrenamiento.

Gestión de Actividades

En esta sección, el entrenador puede crear y gestionar actividades, eligiendo entre diferentes tipos de actividades como *palabras sueltas* o *texto completo*. La Figura 26 ilustra la vista de gestión de actividades.

Figura 26: Vista de gestión de actividades.

El entrenador puede crear y gestionar cursos, asignar usuarios y editar detalles de los cursos, tal y como se muestra en la Figura **??** en la que se puede observar un ejemplo para la gestión de cursos.

Gestión de Ítems

El entrenador tiene la capacidad de crear y asignar ítems a actividades. a modo de ejemplificación se observa en la Figura 27 como la aplicación los gestiona.

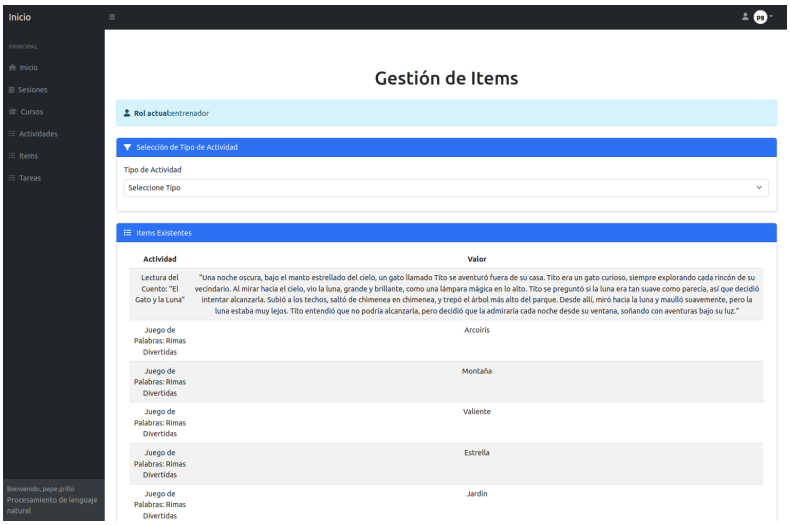


Figura 27: Vista de gestión de ítems.

Gestión de Resultados

El entrenador puede subir y analizar tareas, así como los resultados de los análisis de audio. Figura 28.

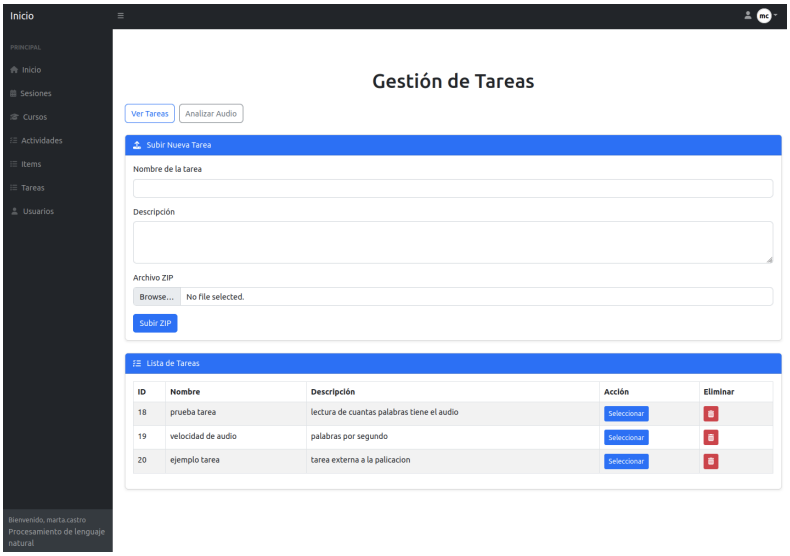


Figura 28: Vista de gestión de resultados en el apartado de subir tareas.

5.8 Administrador

El rol de Administrador incluye las funcionalidades del rol de Entrenador y añade la capacidad de gestionar usuarios. La principal funcionalidad por lo tanto es:

- **Gestión de Usuarios**: creación, modificación y eliminación de cuentas de usuario, así como la asignación de los diferentes roles.

5.8.1 Gestión de Usuarios

El administrador puede gestionar las cuentas de usuario de manera integral, incluyendo su creación, modificación y eliminación. Además, puede asignar diferentes roles y permisos a los usuarios. En la Figura 29 se muestra la vista general de la gestión de usuarios por el administrador:

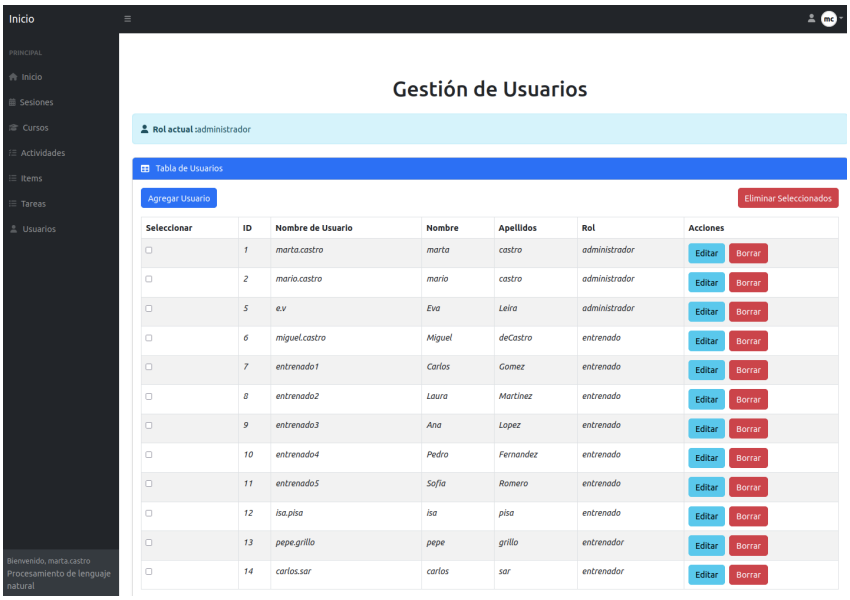


Figura 29: Vista de gestión de usuarios por el administrador.

6 Limitaciones del Sistema

El sistema presenta algunas limitaciones que pueden afectar su funcionamiento y rendimiento. A continuación, se describen las principales restricciones y consideraciones:

- **Problemas con la Calidad del Audio**: el sistema no es capaz de analizar audios que sean incomprensibles o que contengan interferencias significativas. La calidad del análisis depende en gran medida de la claridad del audio proporcionado. Audios con ruido de fondo o distorsiones pueden no ser procesados adecuadamente.
- **Formato Específico para Tareas**: las tareas que se suben al sistema para el análisis de audios deben seguir un formato estricto. Cada tarea debe ser importada a través de un archivo `.zip` que contenga una estructura fija de archivos y carpetas, incluyendo:
  - `info.json`: un diccionario JSON con información sobre la tarea, como el autor, la descripción y los parámetros esperados.
  - `task_code.py`: el código fuente de la tarea, que debe adaptarse a la entrada y salida especificadas en `info.json`.
  - `requeriments.txt`: un archivo que lista las librerías necesarias para ejecutar la tarea.
  - `libraries.zip`: un archivo que contiene librerías adicionales necesarias para la tarea, si las hay.
  - `doc.pdf` y `doc.html`: documentación de la tarea, tanto en formato PDF como HTML.

  Cualquier desviación del formato requerido puede derivar en errores que se produzcan durante la importación o el análisis de la tarea.
- **Dependencia de Recursos Computacionales**: el procesamiento de audios puede ser intensivo en recursos, especialmente cuando se analizan archivos grandes o se realizan tareas complejas. Esto puede requerir una infraestructura computacional robusta para mantener el rendimiento y evitar retrasos en el análisis.
- **Requisitos del Formato de Archivo de Audio**: aunque el sistema está diseñado para

trabajar con archivos de audio en formato `.wav`, todos los archivos en el análisis deben de convertirse al formato específico `WAV PCM` para ser analizados correctamente.

- **Formato WAV vs. WAV PCM**:
  - **WAV (Waveform Audio File Format)**: es un formato de archivo de audio que puede contener datos de audio en varios formatos de codificación. Puede ser un archivo sin comprimir (como PCM) o comprimido. Es una estructura de archivo que define cómo se almacenan los datos de audio en un archivo.
  - **WAV PCM (Pulse Code Modulation)**: es una subcategoría del formato WAV en la que los datos de audio están sin comprimir y representados en una forma de modulación de código de pulso. PCM es el formato de audio más simple y sin pérdida, y representa el audio en una serie de muestras de amplitud a intervalos regulares. Este formato garantiza que los datos de audio sean de alta calidad y sin alteraciones.

  Para asegurar una compatibilidad total y la correcta ejecución de las tareas de análisis, es necesario convertir todos los archivos de audio al formato WAV PCM, ya que el sistema está optimizado para trabajar únicamente con este formato sin comprimir.

7 Líneas de Trabajo Futuras

En este apartado se proponen algunas posibles líneas de trabajo futuras para la evolución del proyecto. Aunque el sistema ha alcanzado sus objetivos, hay varias áreas que podrían beneficiarse de futuras mejoras y desarrollos:

- **Mejora en el Análisis de Audio con Interferencias**: implementar técnicas avanzadas de procesamiento de señales para mejorar el análisis de audios con interferencias o baja calidad. Esto podría incluir el uso de algoritmos de eliminación de ruido o filtros adaptativos.
- **Expansión de Formatos de Audio**: ampliar el soporte para formatos de audio adicionales, además de WAV PCM, para permitir la integración con una gama más amplia de fuentes de audio.
- **Incorporación de Machine Learning para el Análisis de Audio**: explorar el uso de técnicas de aprendizaje automático para mejorar la precisión del análisis de audio. Esto podría incluir la implementación de modelos de reconocimiento de patrones o la utilización de redes neuronales para interpretar audios de manera más efectiva.
- **Comparación del Audio con el Ítem de Lectura**: desarrollar una funcionalidad para comparar el audio grabado por el usuario con el ítem de lectura correspondiente. Esto permitirá evaluar si el contenido ha sido leído correctamente y proporcionar retroalimentación precisa sobre la exactitud de la lectura.
- **Descarga y Modificación de Tareas**: implementar una funcionalidad que permita a los usuarios con persisos de entrenador descargar tareas subidas por otros, con el objetivo de mejorarlas y reutilizarlas. Esto fomentará la colaboración y el intercambio de recursos entre los usuarios del sistema, facilitando la personalización y la adaptación de las tareas según las necesidades específicas.
- **Automatización del Análisis de Audios en Función de Actividades**: permitir que los entrenadores seleccionen previamente las actividades cuyos audios de respuesta deben analizarse con tareas específicas. De esta manera, cuando un usuario grabe un audio relacionado con una actividad seleccionada, el sistema realizará automáticamente

el análisis correspondiente y ofrecerá una respuesta inmediata sobre el resultado de ese análisis. Esta funcionalidad mejoraría la eficiencia y la rapidez en la evaluación del desempeño del usuario.

Estas líneas de trabajo futuras pueden añadir valor al sistema y mejorar su funcionalidad, haciendo que sea más robusto y adaptado a las necesidades cambiantes de los usuarios.

8 Conclusiones

Tras la finalización del desarrollo del sistema, se ha logrado cumplir de manera exitosa tanto los objetivos técnicos del proyecto como los objetivos personales planteados al inicio.

En primer lugar, se ha desarrollado de manera efectiva una plataforma integral para la gestión de actividades de lectura. Esta herramienta permite la creación, edición y administración de un banco de pruebas compuesto por sesiones de lectura, generando datos que pueden ser analizados para ofrecer resultados útiles. Dichos resultados proporcionan una base sólida para un análisis detallado, cumpliendo así con el objetivo de ofrecer una herramienta valiosa para investigadores y educadores en contextos educativos y formativos.

La gestión de usuarios y datos ha sido implementada satisfactoriamente, permitiendo un manejo eficiente de las operaciones de creación, modificación, visualización y eliminación de usuarios. Además, la plataforma gestiona eficazmente los datos recolectados de las actividades de lectura, cumpliendo con el objetivo de administrar y procesar información a gran escala.

El sistema ha sido diseñado utilizando herramientas modernas como Django REST Framework y Angular, lo que garantiza su escalabilidad y robustez. Esta arquitectura asegura que la plataforma pueda manejar un gran volumen de datos y adaptarse a futuras necesidades, consolidando su relevancia y utilidad a largo plazo.

En cuanto a la interoperabilidad, el sistema ha sido diseñado para aceptar y ejecutar tareas codificadas por terceros, lo que permite su integración con nuevos códigos externos para el análisis de resultados. Esta capacidad amplía la funcionalidad del sistema, incrementando su flexibilidad y versatilidad para su uso en diversos contextos educativos.

Respecto a los objetivos personales, el desarrollo de este Trabajo de Fin de Grado ha permitido aplicar de manera efectiva los conocimientos adquiridos durante la carrera, afrontando el desafío de gestionar de forma individual un proyecto de software integral. Este proceso ha contribuido al fortalecimiento de habilidades en desarrollo web, gestión de proyectos de software, así como en la organización y gestión del tiempo. Además, la plataforma desarrollada

ofrece un valor añadido al ámbito educativo, al proporcionar herramientas avanzadas para el entrenamiento de habilidades de lectura, apoyando el desarrollo de nuevas metodologías educativas basadas en tecnología digital.

En síntesis, el desarrollo del sistema ha sido exitoso en la consecución de los objetivos propuestos, estableciendo una base sólida para futuros avances y mejoras en el análisis de actividades de lectura. Las experiencias adquiridas durante el proyecto han sido invaluables, contribuyendo significativamente al crecimiento profesional y académico.

Bibliografía

[1] Procesamiento de Lenguaje Natural. *PLN*. Recuperado de https://www.ibm.com/es-es/topics/natural-language-processing.

[2] Durán Toro, A., & Bernárdez Jiménez, B. (2000). *Metodología para la Elicitación de Requisitos de Sistemas Software* (Versión 2.1). Informe Técnico LSI–2000–10. Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática y Estadística, Sevilla.

[3] Python. *Python Overview*. Recuperado de https://builtin.com/software-engineering-perspectives/python.

[4] JavaScript. *JavaScript*. Recuperado de https://en.wikipedia.org/wiki/JavaScript.

[5] HTML. *HTML*. Recuperado de https://es.wikipedia.org/wiki/HTML.

[6] CSS. *CSS*. Recuperado de https://en.wikipedia.org/wiki/CSS.

[7] Moreno García, M. N. *Transparencias de Gestión de Proyectos, Práctica 1 – Estimación del esfuerzo*.

[8] Moreno García, M. N. *Transparencias de Gestión de Proyectos, Práctica 2 – Planificación temporal*.

[9] Microsoft. *Software de Gestión de Proyectos*. Recuperado de https://www.microsoft.com/es-es/microsoft-365/project/project-management-software.

[10] Python. *Python Overview*. Recuperado de https://builtin.com/software-engineering-perspectives/python.

[11] JavaScript. *JavaScript*. Recuperado de https://en.wikipedia.org/wiki/JavaScript.

[12] HTML. *HTML*. Recuperado de https://es.wikipedia.org/wiki/HTML.

[13] Wikipedia. (2024). *CSS*. Recuperado de https://es.wikipedia.org/wiki/CSS

[14] EstiloWeb. (n.d.). *Métodos HTTP: POST, GET, PUT y DELETE*. Recuperado de https://estilow3b.com/metodos-http-post-get-put-delete/

[15] Bootstrap. *Bootstrap*. Recuperado de https://getbootstrap.com/.

[16] Node.js. *Node.js*. Recuperado de https://nodejs.org/.

[17] NPM. *NPM*. Recuperado de https://www.npmjs.com/.

[18] MySQL. *MySQL*. Recuperado de https://www.mysql.com/.

[19] Visual Studio Code. *Visual Studio Code*. Recuperado de https://code.visualstudio.com/.

[20] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Recuperado de https://www.scrumguides.org/scrum-guide.html.

[21] Trello. (n.d.). *Trello*. Recuperado de https://trello.com.

[22] Boehm, B. W. (1988). *A Spiral Model of Software Development and Enhancement*. ACM SIGSOFT Software Engineering Notes, 11(4), 14-24. Recuperado de https://dl.acm.org/doi/10.1145/74334.74337.

[23] Draw.io. (n.d.). *Draw.io*. Recuperado de https://keepcoding.io/blog/que-es-drawio/.

[24] Microsoft. (n.d.). *Microsoft Project*. Recuperado de https://www.microsoft.com/en-us/microsoft-365/project/project-management-software.

[25] Nielsen, J. (2000). *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing.

[26] Angular. (n.d.). *Angular Documentation*. Recuperado de https://angular.io/docs

[27] Django Software Foundation. (n.d.). *Django Documentation*. Recuperado de https://docs.djangoproject.com/

[28] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine.

[29] Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR.

[30] Glifing. (n.d.). *Método Glifing: Ayuda a tus hijos a mejorar su lectura*. Recuperado el 4 de septiembre de 2024, de https://www.glifing.com/

[31] Atlassian. (n.d.). *¿Qué es Scrum?*. Recuperado el 4 de septiembre de 2024, de `https://www.atlassian.com/es/agile/scrum`

[32] Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux Journal*, 2014(239), 2.

[33] Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress. Recuperado de `https://git-scm.com/book/en/v2`.

[34] Fielding, R. T., & Taylor, R. N. (2000). Architectural Styles and the Design of Network-based Software Architectures. *Doctoral dissertation*, University of California, Irvine. Recuperado de `https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm`.

[35] Jose, D. (2022). MVVM Architecture in Angular: A Complete Guide. Recuperado de `https://deepakjosecodes.com/mvvm-architecture-in-angular-guide-with-code-examples/`.

Análisis de lectura de textos
Anexo I. Plan del proyecto de software

Trabajo de Fin de Grado
GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:
Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),
Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

Índice general

1. Introducción **1**

2. Estimación del esfuerzo **3**

2.1. Componentes de la métrica UCP 3

2.2. Complejidad de los actores 4

2.3. Complejidad de los requisitos funcionales 5

2.4. Factores implicados en la estimación temporal 6

2.4.1. Factores técnicos 6

2.4.2. Factores del entorno 9

2.5. Interpretación de resultados 10

3. Planificación temporal **13**

3.1. Calendario de trabajo 13

3.2. Identificación de tareas y subtareas 14

3.3. Resultados 18

Bibliografía **19**

Índice de figuras

1. Proceso Unificado. 1
2. Estimación temporal. 10
3. Calendario. 13
4. Diagrama de Gantt. 14
5. Planificación temporal. 15
6. Fase inicio, primera iteración. 15
7. Fase elaboración, primera iteración. 16
8. Fase elaboración, segunda iteración. 16
9. Fase construcción, primera iteración. 17
10. Fase construcción, segunda iteración. 17
11. Fase construcción, tercera iteración. 17
12. Fase transición, primera iteración. 18
13. Estadísticas de la planificación temporal. 18
14. Carga de trabajo. 19

Índice de tablas

1. Peso asignado a la complejidad en la interacción de actores. 4

2. Peso asignado a los actores. 5

4. Factores de complejidad técnica. 8

5. Factores del entorno. 9

1 Introducción

En este anexo se presenta una visión detallada de la planificación temporal y la estimación del esfuerzo necesario para llevar a cabo el proyecto. Una planificación adecuada es esencial para garantizar el éxito de un proyecto de software, ya que permite establecer un cronograma claro, definir tareas, asignar recursos y fijar plazos [1]. Esto facilita una mejor gestión del tiempo y de los recursos disponibles y ayuda a identificar y mitigar posibles riesgos de retrasos, optimizando así la productividad durante todo el proceso de desarrollo.

El desarrollo de este proyecto se llevará a cabo siguiendo el marco de desarrollo software del Proceso Unificado. Este marco permite un enfoque iterativo e incremental, proporcionando la flexibilidad necesaria para adaptar el proyecto a medida que evoluciona. El Proceso Unificado (Figura 1) se estructura en varias fases solapadas, cada una contribuyendo al avance progresivo del proyecto. Estas fases incluyen: modelado de negocio, elicitación de requisitos, análisis de requisitos, diseño del sistema, implementación y pruebas. Este anexo se divide en dos apartados:

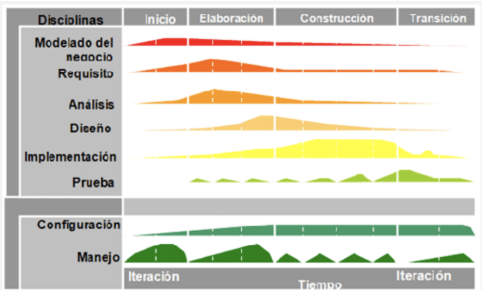


Figura 1: Proceso Unificado.

- **Estimación del esfuerzo**: en esta sección se detallan las tareas y los recursos humanos asignados, así como el tiempo estimado para cada una de ellos.
- **Planificación temporal**: aquí se establece un cronograma detallado que guía la ejecución del proyecto, minimizando riesgos y asegurando un seguimiento eficaz.

2 Estimación del esfuerzo

En este apartado se presenta la estimación temporal basada en la complejidad técnica y la implementación de los requisitos funcionales, utilizando los actores y requisitos descritos en el **Anexo II: Especificación de requisitos software**. Para llevar a cabo esta estimación, se utiliza la métrica UCP (Use Case Points), la cual evalúa la funcionalidad representada en forma de casos de uso. Los UCP son fundamentales para estimar el esfuerzo de desarrollo y consideran diversos factores que incluyen actores, escenarios y factores técnicos y del entorno [1].

2.1 Componentes de la métrica UCP

La métrica UCP se descompone en los siguientes componentes:

- **UUCP (Unadjusted Use Case Points)**: es una medida básica que se calcula sumando el peso no ajustado de todos los casos de uso y actores refleja la complejidad de estos.
  - **UUCW (Unadjusted Use Case Weight)**: considera el número y complejidad de los casos de uso.
  - **UAW (Unadjusted Actor Weight)**: considera el número y complejidad de los actores.
- **TCF (Technical Complexity Factor)**: factor de Complejidad Técnica que se calcula a partir de una serie de factores técnicos que afectan el desarrollo del proyecto.
- **ECF (Environmental Complexity Factor)**: factor de Complejidad del Entorno que se calcula a partir de una serie de factores del entorno que afectan el desarrollo del proyecto.

La fórmula general para calcular los **UCP** es:

$$UCP = UUCP \times TCF \times ECF$$

Donde:

- **UUCP** es el valor no ajustado de los casos de uso.
- **TCF** es el Factor de Complejidad Técnica.
- **ECF** es el Factor de Complejidad del Entorno.

La aplicación de esta métrica proporciona una estimación más precisa del esfuerzo requerido para el desarrollo del software, considerando tanto la complejidad técnica como las condiciones del entorno en que se llevará a cabo el proyecto.

2.2 Complejidad de los actores

En la gestión de proyectos, se crea una tabla que asigne un peso a los actores que interactúan con el sistema, esto permite determinar la importancia relativa de cada interacción. Los pesos asignados reflejan la complejidad y el esfuerzo necesarios para implementar estas iteraciones.

Tipo de interacción	**Peso asignado**
Simple (API)	1
Media (Protocolo)	2
Compleja (Interfaz gráfica)	3

Tabla 1: Peso asignado a la complejidad en la interacción de actores.

Las iteraciones simples, como las realizadas mediante API, tienen un peso de 1, indicando baja complejidad; las iteraciones de complejidad media, a través de protocolos, tienen un peso de 2 y las iteraciones más complejas, realizadas mediante interfaz gráfica, tienen un peso de 3.

Actor	Peso asignado
Usuario no identificado	3
Usuario identificado	3
Participante	3
Entrenador	3
Administrador	3

Tabla 2: Peso asignado a los actores.

2.3 Complejidad de los requisitos funcionales

Asignar estos pesos proporciona una base sólida para la asignación de recursos y la planificación temporal del proyecto, facilitando la priorización de tareas y la correcta distribución del tiempo necesario para implementar cada interacción.

Para determinar la complejidad de cada caso de uso, se asigna una categoría en función del número de transacciones. Una transacción comienza cuando un usuario inicia una acción y finaliza cuando el sistema responde. La clasificación utilizada se muestra aquí:

- **Simple**: entre 0 y 3 transacciones.
- **Medio**: entre 4 y 7 transacciones.
- **Complejo**: más de 7 transacciones.

Caso de Uso	Transacciones	Complejidad
UC-001: Registrar usuario	3	Simple
UC-002: Iniciar sesión	2	Simple
UC-003: Cerrar sesión	1	Simple
UC-004: Consultar datos personales	1	Simple
UC-005: Modificar datos personales	3	Simple
UC-006: Crear usuario	3	Simple
UC-007: Modificar datos de un usuario	3	Simple
UC-008: Eliminar usuario	3	Simple
UC-009: Crear cursos	3	Simple
UC-010: Listar usuarios por curso	1	Simple
UC-011: Editar usuarios de curso	3	Simple
UC-012: Eliminar curso	3	Simple

Caso de Uso	Transacciones	Complejidad
UC-013: Asignar usuarios a cursos	2	Simple
UC-014: Crear sesiones	3	Simple
UC-015: Modificar sesiones	3	Simple
UC-016: Eliminar sesiones	3	Simple
UC-017: Asignar actividades	2	Simple
UC-018: Asignar cursos	2	Simple
UC-019: Listar sesiones	1	Simple
UC-020: Filtrar por realización	2	Simple
UC-021: Iniciar entrenamiento	3	Simple
UC-022: Grabar entrenamiento	3	Simple
UC-023: Crear actividades	3	Simple
UC-024: Modificar actividades	3	Simple
UC-025: Eliminar actividades	3	Simple
UC-026: Crear items	3	Simple
UC-027: Asignar items	2	Simple
UC-028: Ver actividades	1	Simple
UC-029: Importar tareas	4	Medio
UC-030: Analizar audios	4	Medio
UC-031: Actividades completadas	2	Simple

Tabla de casos de uso con complejidad asignada, basada en el número de transacciones.

2.4 Factores implicados en la estimación temporal

Para obtener una estimación temporal precisa, es esencial considerar los diversos factores que influyen en el desarrollo del sistema. Estos factores se dividen en dos grupos: **factores técnicos** y **factores del entorno**. Cada factor tiene un valor asignado que afecta la estimación del tiempo requerido para completar el proyecto.

2.4.1 Factores técnicos

Son elementos que influyen directamente en la complejidad y el esfuerzo necesario para llevar a cabo un proyecto. Estos factores pueden incluir aspectos como la arquitectura del

sistema, el rendimiento, la escalabilidad, la facilidad de uso... La correcta identificación y evaluación de estos permite anticipar posibles desafíos y optimizar los recursos y el tiempo de desarrollo.

En el contexto de nuestra aplicación, es crucial evaluar estos factores para asegurar que la solución sea robusta, eficiente y fácil de usar. A continuación, se detallan los factores técnicos específicos considerados y cómo cada uno de ellos contribuye a la complejidad técnica del proyecto. Tras realizarlo, se calcula TCF [1] mediante la expresión:

$$\mathrm{TCF} = C_1 + C_2 \sum_{i=1}^{13} W_i F_i$$

Factor	Valor	Descripción
Sistemas distribuidos	1	Actualmente no está distribuido, por lo que la complejidad técnica en este área es baja.
Rendimiento	3	El sistema debe cumplir un rendimiento medio- alto para que las iteraciones de los usuarios sean fluidas y comprendidas rápidamente.
Eficiencia del usuario final	1	La aplicación está diseñada para ser utilizada por niños, lo que implica que la interfaz debe ser extremadamente sencilla e intuitiva. El número de pasos que un usuario necesita para completar una tarea debe ser mínimo.
Procesamiento interno complejo	3	Descomprimir archivos ZIP y analizar su contenido añade complejidad técnica. Además de las operaciones CRUD básicas, esto implica un procesamiento más avanzado, como manejo de archivos y validación de datos.
Reusabilidad	2	La aplicación puede tener algunos componentes reutilizables, pero en general, la reutilización es limitada debido a la especificidad de la aplicación.
Facilidad de instalación	0	El sistema consta de una aplicación web accesible desde cualquier navegador.
Facilidad de uso	4	La facilidad de uso es fundamental, especialmente para niños, por lo que la aplicación debe ser muy intuitiva.
Portabilidad	3	La aplicación necesita funcionar en múltiples dispositivos y sistemas operativos, lo cual añade cierta complejidad técnica.
Facilidad a cambio	3	El sistema se diseñará para permitir la adicción de nuevas funcionalidades.
Concurrencia	3	El sistema se desarrolla para tener la capacidad de ser concurrente.
Características especiales de seguridad	2	El sistema debe garantizar la privacidad de la información almacenada en él y únicamente accesible para aquellos usuarios autorizados.
Acceso directo a terceras partes	0	El sistema se despliega en un entorno cerrado, es decir, solo es accesible por aquellos usuarios que forman parte de él.
Entrenamiento especial del usuario	0	El sistema que se plantea no necesita incluir ninguna ayuda especial para los usuarios al comienzo de su uso.

Tabla 4: Factores de complejidad técnica.

2.4.2 Factores del entorno

Los factores del entorno son condiciones externas que pueden influir en la planificación y duración del proyecto.

Tras realizarlo, se calcula EFC [1] mediante la expresión:

$$\mathrm{ECF} = C_1 + C_2 \sum_{i=1}^{8} W_i F_i$$

Los factores del entorno, sus valores y descripciones, son los citados en la Tabla 5.

Factor	**Valor**	**Descripción**
Familiaridad con el UML	2	Se ha estudiado UML y se ha trabajado sobre el proceso unificado en trabajos de práctica.
Trabajadores a tiempo parcial	0	No se incorporan trabajadores a tiempo parcial, ya que el proyecto lo realiza un único desarrollador.
Capacidad de los analistas	0	El desarrollador no presenta ninguna experiencia previa en alguna técnica de análisis.
Experiencia en la aplicación	1	El desarrollador es nuevo en el desarrollo de aplicaciones educativas, pero está aprendiendo rápidamente y se adapta a las nuevas tecnologías.
Experiencia en orientación a objetos	2	El desarrollador ha completado proyectos pequeños utilizando programación orientada a objetos, pero esta es su primera aplicación completa que requiere un enfoque orientado a objetos sólido.
Motivación	5	El desarrollador presenta una gran motivación debido a que afronta su primer proyecto completo.
Dificultad del lenguaje de programación	2	Los lenguajes de programación a utilizar son de alto nivel, pero en alguno de ellos no se tiene ninguna experiencia previa.
Estabilidad de los requisitos	3	Los requisitos han sido fijados después de haber sido revisados en numerosas ocasiones

Tabla 5: Factores del entorno.

2.5 Interpretación de resultados

En este punto, hay que definir el factor de conversión que se va a usar, este valor se corresponde con el número de horas de personas por UCP y se va a dar el valor de 5. La evaluación del tiempo requerido para completar el proyecto se ha realizado utilizando la herramienta EZEstimate. Para calcular esta estimación, basta con ingresar en la aplicación la información detallada en las secciones previas.

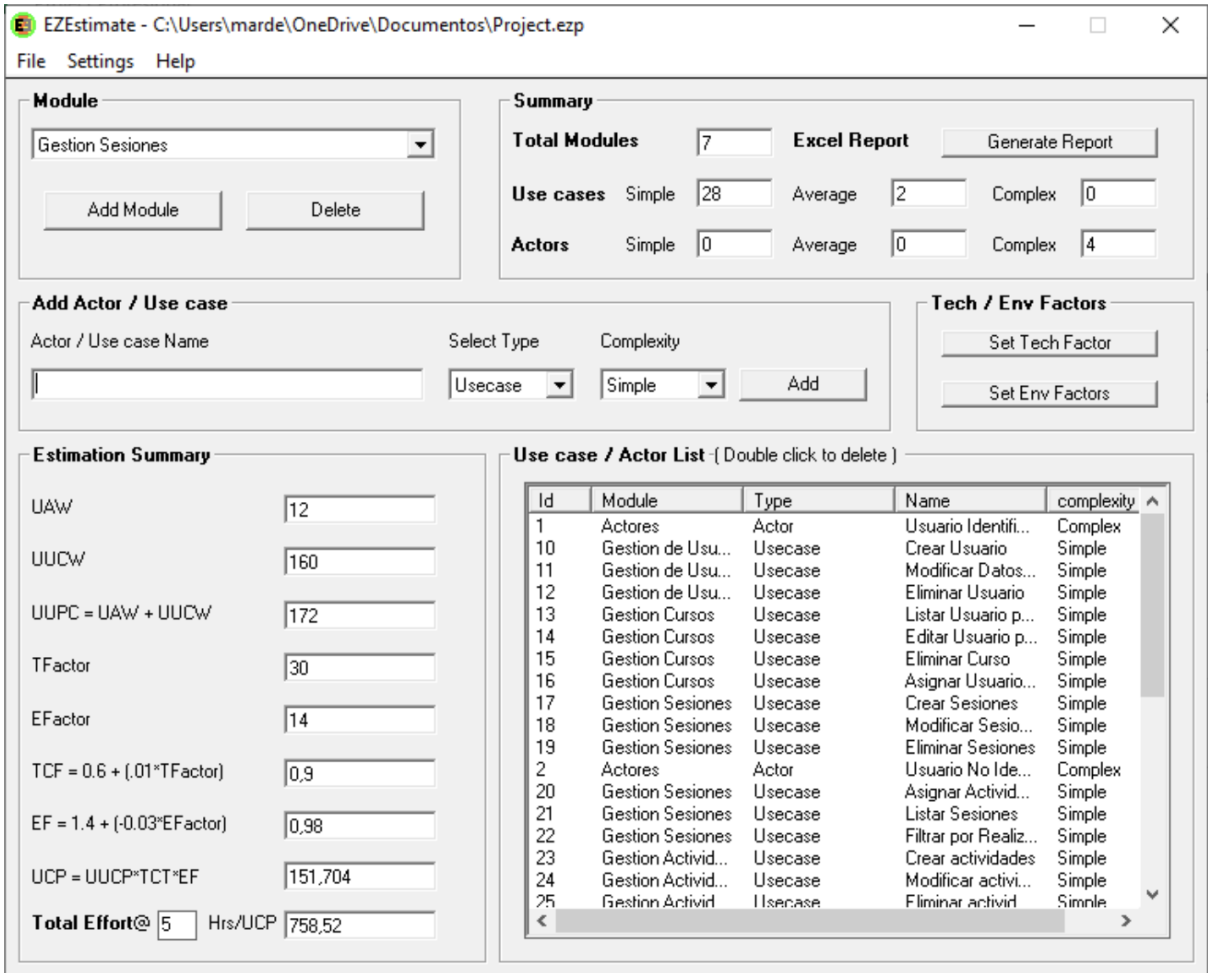


Figura 2: Estimación temporal.

En la Figura 2 la estimación temporal obtenida para el proyecto es de aproximadamente 758,52 horas de trabajo.

Si consideramos una jornada laboral de 8 horas diarias, el proyecto requeriría aproximadamente 95 días laborables para completarse. Esto se traduce en un período de aproximadamente

3 meses y 5 días naturales. En términos de días laborables, esto equivale a cerca de 4 meses y 3 días, calculado teniendo en cuenta el período laboral expuesto en el siguiente apartado.

- **Horas estimadas:** 758,52 horas.
- **Horas diarias de trabajo:** 8 horas.
- **Días laborables necesarios:** 95 días.
- **Período en meses laborables:** Aproximadamente 3 meses y 5 días.
- **Período en días naturales:** Aproximadamente 127 días (4 meses y 3 días).

3 Planificación temporal

En esta sección se detalla el procedimiento empleado para llevar a cabo la planificación temporal[2], lo cual es crucial para el desarrollo eficiente del sistema de software. Para realizar esta planificación, se ha utilizado la herramienta Microsoft Project[3].

3.1 Calendario de trabajo

La planificación de mi trabajo para el desarrollo del proyecto establece una jornada de 8 horas diarias de lunes a viernes, distribuidas de la siguiente manera: 5 horas en la mañana, de 9:00 a 14:00, y 3 horas en la tarde, de 15:00 a 18:00. Adicionalmente, hay un período de pausa previsto para los exámenes, que abarca desde el 27 de mayo hasta el 9 de junio. El proyecto da inicio el 17 de abril y apartir de ahí dará comienzo la estimación de tiempo.

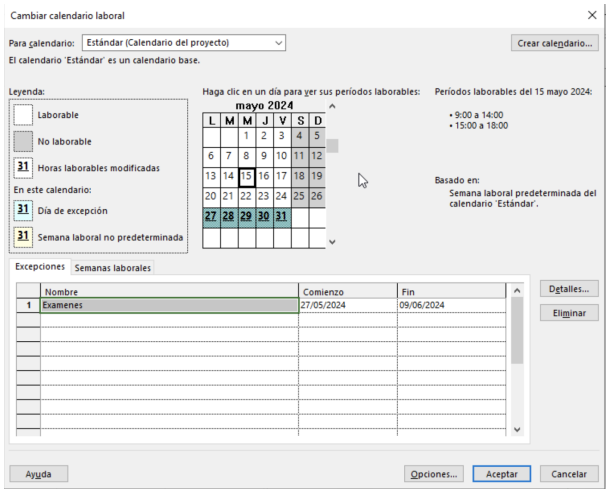


Figura 3: Calendario.

3.2 Identificación de tareas y subtareas

Con el calendario de trabajo configurado, se procede a definir las tareas necesarias para el desarrollo del proyecto. Para estructurar estas actividades, se ha utilizado el enfoque del Proceso Unificado, que organiza el trabajo en cuatro fases principales: inicio, elaboración, construcción y transición. Este enfoque permite una gestión ordenada del desarrollo del software, dividiendo el trabajo en diferentes áreas de enfoque.

El Proceso Unificado divide el desarrollo del software en varias disciplinas: modelado del negocio, requisitos, análisis, diseño, implementación y pruebas del sistema. Cada una de estas fases está segmentada en iteraciones, y cada iteración conlleva una carga de trabajo específica. Para este proyecto, la distribución de las iteraciones por fase es la siguiente:

- **Inicio:** 1 iteración
- **Elaboración:** 2 iteraciones
- **Construcción:** 3 iteraciones
- **Transición:** 1 iteración

Como se observa en la Figura 1, Proceso Unificado, el enfoque de trabajo varía en función de la fase y la iteración correspondiente. Las actividades de desarrollo se asignan a diferentes disciplinas para garantizar un desarrollo completo y equilibrado del software. Es relevante mencionar que este esquema sirve como una referencia general y se ajusta durante el desarrollo del proyecto, dependiendo de las necesidades específicas que surjan para alcanzar los objetivos establecidos.

Mi planificación temporal se muestra en la Figura 5, y su diagrama de Gantt en la Figura 4.

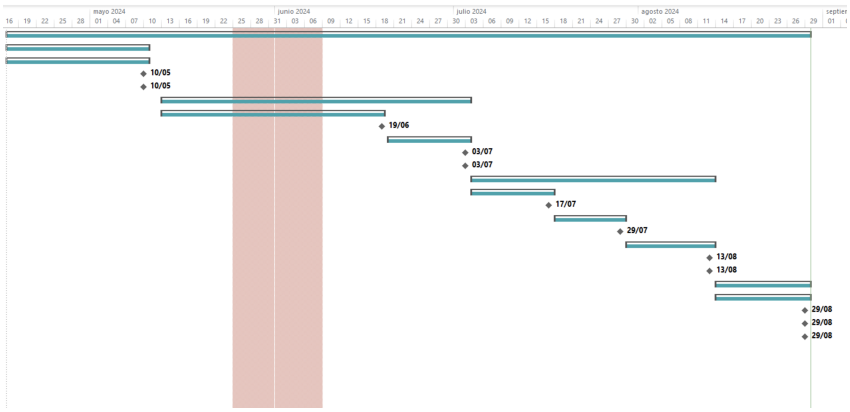


Figura 4: Diagrama de Gantt.

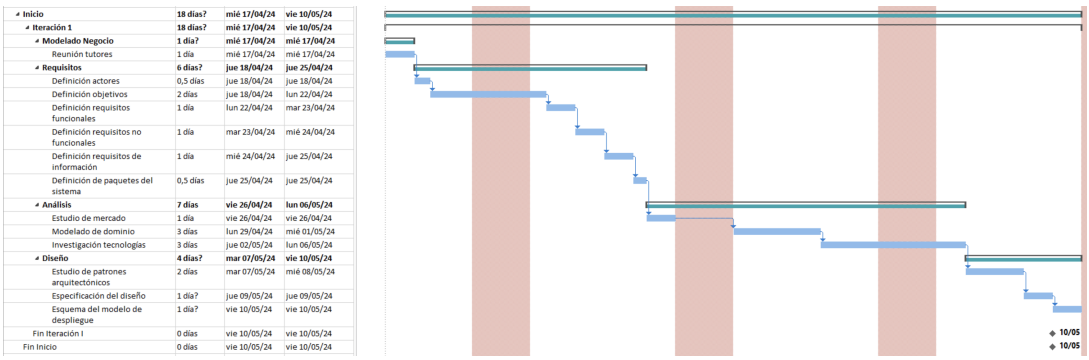
	Modo de	Nombre de tarea	Duración	Comienzo	Fin
		Proyecto	**87 días**	**mié 17/04/24**	**jue 29/08/24**
		Inicio	**18 días**	**mié 17/04/24**	**vie 10/05/24**
		Iteración 1	**18 días**	**mié 17/04/24**	**vie 10/05/24**
		Fin Iteración I	0 días	vie 10/05/24	vie 10/05/24
		Fin Inicio	0 días	vie 10/05/24	vie 10/05/24
		Elaboración	**28 días**	**lun 13/05/24**	**mié 03/07/24**
		Iteración I	**17,5 días**	**lun 13/05/24**	**mié 19/06/24**
		Fin Iteración I	0 días	mié 19/06/24	mié 19/06/24
		Iteración II	**10 días**	**jue 20/06/24**	**mié 03/07/24**
		Fin Iteración II	0 días	mié 03/07/24	mié 03/07/24
		Fin Elaboración	0 días	mié 03/07/24	mié 03/07/24
		Construcción	**29 días**	**jue 04/07/24**	**mar 13/08/24**
		Iteración I	**10 días**	**jue 04/07/24**	**mié 17/07/24**
		Fin Iteración I	0 días	mié 17/07/24	mié 17/07/24
		Iteración II	**8 días**	**jue 18/07/24**	**lun 29/07/24**
		Fin Iteración II	0 días	lun 29/07/24	lun 29/07/24
		Iteración III	**11 días**	**mar 30/07/24**	**mar 13/08/24**
		Fin Iteración III	0 días	mar 13/08/24	mar 13/08/24
		Fin contrucción	0 días	mar 13/08/24	mar 13/08/24
		Transición	**12 días**	**mié 14/08/24**	**jue 29/08/24**
		Iteración I	**12 días**	**mié 14/08/24**	**jue 29/08/24**
		Fin Iteración I	0 días	jue 29/08/24	jue 29/08/24
		Fin Transición	0 días	jue 29/08/24	jue 29/08/24
		Fin Planificación Temporal	0 días	jue 29/08/24	jue 29/08/24

Figura 5: Planificación temporal.

Una vez visualizada la vista global, se muestra cada iteracion de cada fase:

Inicio

Iteración I:



Inicio	**18 días?**	**mié 17/04/24**	**vie 10/05/24**
Iteración 1	**18 días?**	**mié 17/04/24**	**vie 10/05/24**
Modelado Negocio	**1 día?**	**mié 17/04/24**	**mié 17/04/24**
Reunión tutores	1 día	mié 17/04/24	mié 17/04/24
Requisitos	**6 días?**	**jue 18/04/24**	**jue 25/04/24**
Definición actores	0,5 días	jue 18/04/24	jue 18/04/24
Definición objetivos	2 días	jue 18/04/24	lun 22/04/24
Definición requisitos funcionales	1 día	lun 22/04/24	mar 23/04/24
Definición requisitos no funcionales	1 día	mar 23/04/24	mié 24/04/24
Definición requisitos de información	1 día	mié 24/04/24	jue 25/04/24
Definición de paquetes del sistema	0,5 días	jue 25/04/24	jue 25/04/24
Análisis	**7 días**	**vie 26/04/24**	**lun 06/05/24**
Estudio de mercado	1 día	vie 26/04/24	vie 26/04/24
Modelado de dominio	3 días	lun 29/04/24	mié 01/05/24
Investigación tecnologías	3 días	jue 02/05/24	lun 06/05/24
Diseño	**4 días?**	**mar 07/05/24**	**vie 10/05/24**
Estudio de patrones arquitectónicos	2 días	mar 07/05/24	mié 08/05/24
Especificación del diseño	1 día?	jue 09/05/24	jue 09/05/24
Esquema del modelo de despliegue	1 día?	vie 10/05/24	vie 10/05/24
Fin Iteración I	0 días	vie 10/05/24	vie 10/05/24
Fin Inicio	0 días	vie 10/05/24	vie 10/05/24

Figura 6: Fase inicio, primera iteración.

Elaboración

Iteración I:

Elaboración	**29 días?**	**lun 13/05/24**	**jue 04/07/24**
Iteración I	**17,52 días?**	**lun 13/05/24**	**mié 19/06/24**
Modelado de Negocio	**5 días**	**lun 13/05/24**	**vie 17/05/24**
Reunión tutores	1 día	lun 13/05/24	lun 13/05/24
Estimación del esfuerzo	2 días	mar 14/05/24	mié 15/05/24
Planificación temporal	2 días	jue 16/05/24	vie 17/05/24
Requisitos	**5 días?**	**lun 20/05/24**	**vie 24/05/24**
Repasar casos de uso	0,5 días	lun 20/05/24	lun 20/05/24
Documentación actores	0,5 días	lun 20/05/24	lun 20/05/24
Documentación de los requisitos de información	0,5 días	mar 21/05/24	mar 21/05/24
Documentación de los requisitos de información	0,5 días	mar 21/05/24	mar 21/05/24
Documentación de los casos de u	1,5 días?	mié 22/05/24	jue 23/05/24
Diagramas de casos de uso	1,5 días	jue 23/05/24	vie 24/05/24
Análisis	**3 días?**	**lun 10/06/24**	**mié 12/06/24**
Paquete de análisis	1,5 días?	lun 10/06/24	mar 11/06/24
Realización de casos de uso. Gestión Usuarios	0,5 días	mar 11/06/24	mar 11/06/24
Realización de casos de uso. Gestión Cursos	0,5 días	mié 12/06/24	mié 12/06/24
Realización de casos de uso. Autentificación Usuario	0,5 días	mié 12/06/24	mié 12/06/24
Diseño	**2 días**	**jue 13/06/24**	**vie 14/06/24**
Estudio de patrones de diseño	1 día	jue 13/06/24	jue 13/06/24
Realización de clases de diseño	1 día	vie 14/06/24	vie 14/06/24
Implementación	**2 días**	**lun 17/06/24**	**mar 18/06/24**
Montar el servidor	2 días	lun 17/06/24	mar 18/06/24
Pruebas	**0,5 días?**	**mié 19/06/24**	**mié 19/06/24**
Pruebas en el servidor	0,5 días	mié 19/06/24	mié 19/06/24
Fin Iteración I	0 días	mié 19/06/24	mié 19/06/24
Iteración II	**10 días?**	**jue 20/06/24**	**mié 03/07/24**

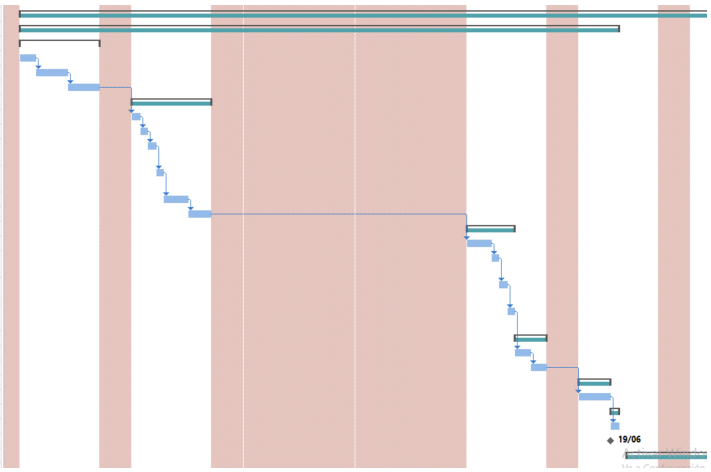


Figura 7: Fase elaboración, primera iteración.

Iteración II:

Iteración II	**10 días?**	**jue 20/06/24**	**mié 03/07/24**
Modelado de Negocio	**1 día**	**jue 20/06/24**	**jue 20/06/24**
Reunión tutores	1 día	jue 20/06/24	jue 20/06/24
Análisis	**1,5 días?**	**vie 21/06/24**	**lun 24/06/24**
Realización de casos de uso. Gestión Actividades	0,5 días	vie 21/06/24	vie 21/06/24
Realización de casos de uso. Gestión Sesiones	0,5 días	vie 21/06/24	vie 21/06/24
Realización de casos de uso. Gestión Resultados	0,5 días	lun 24/06/24	lun 24/06/24
Diseño	**3 días**	**lun 24/06/24**	**jue 27/06/24**
Análisis de casos de uso. Autentificación usuarios	0,5 días	lun 24/06/24	lun 24/06/24
Análisis de casos de uso. Gestión Usuarios	1 día	mar 25/06/24	mar 25/06/24
Análisis de casos de uso. Gestión Cursos	1 día	mié 26/06/24	mié 26/06/24
Implementación	**2 días**	**jue 27/06/24**	**vie 28/06/24**
Construcción componente: Autentificación usuarios	0,5 días	jue 27/06/24	jue 27/06/24
Construcción componente: Gestión usuarios	0,5 días	jue 27/06/24	jue 27/06/24
Construcción componente: Gestión Cursos	1 día	vie 28/06/24	vie 28/06/24
Pruebas	**3 días?**	**lun 01/07/24**	**mié 03/07/24**
Pruebas del componente: Autentificación usuarios	1 día?	lun 01/07/24	lun 01/07/24
Pruebas del componente: Gestión usuarios	1 día?	mar 02/07/24	mar 02/07/24
Pruebas del componente: Gestión Cursos	1 día?	mié 03/07/24	mié 03/07/24
Fin Iteración II	0 días	mié 03/07/24	mié 03/07/24
Fin Elaboración	0 días	mié 03/07/24	mié 03/07/24

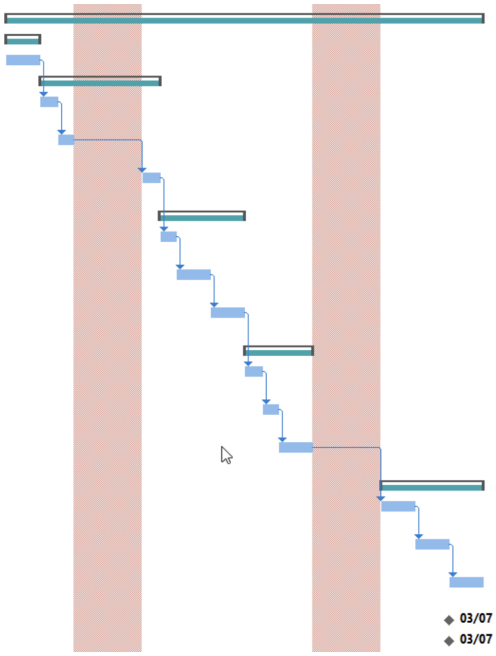


Figura 8: Fase elaboración, segunda iteración.

Construcción

Iteración I:

Construcción	**30,63 días?**	**jue 04/07/24**	**jue 15/08/24**
Iteración I	**10 días?**	**jue 04/07/24**	**mié 17/07/24**
Modelado de Negocio	**1 día**	**jue 04/07/24**	**jue 04/07/24**
Reunión con los tutoe	1 día	jue 04/07/24	jue 04/07/24
Diseño	**3 días**	**vie 05/07/24**	**mar 09/07/24**
Análisis de casos de uso. Gestión actividades	1 día	vie 05/07/24	vie 05/07/24
Análisis de casos de uso. Gestión Sesiones	1 día	lun 08/07/24	lun 08/07/24
Análisis de casos de uso. Gestión Resultados	1 día	mar 09/07/24	mar 09/07/24
Implementación	**3 días**	**mié 10/07/24**	**vie 12/07/24**
Construcción componente: Gestión Actividades	1 día	mié 10/07/24	mié 10/07/24
Construcción componente: Gestión Sesiones	1 día	jue 11/07/24	jue 11/07/24
Construcción componente: Gestión Resultados	1 día	vie 12/07/24	vie 12/07/24
Pruebas	**3 días**	**lun 15/07/24**	**mié 17/07/24**
Pruebas del componente: Gestión Actividades	1 día	lun 15/07/24	lun 15/07/24
Pruebas del componente: Gestión Sesiones	1 día	mar 16/07/24	mar 16/07/24
Pruebas del componente: Gestión Resultados	1 día	mié 17/07/24	mié 17/07/24
Fin Iteración I	0 días	mié 17/07/24	mié 17/07/24
Iteración II	**10 días?**	**jue 18/07/24**	**mié 31/07/24**
Fin Iteración II	0 días	lun 29/07/24	lun 29/07/24

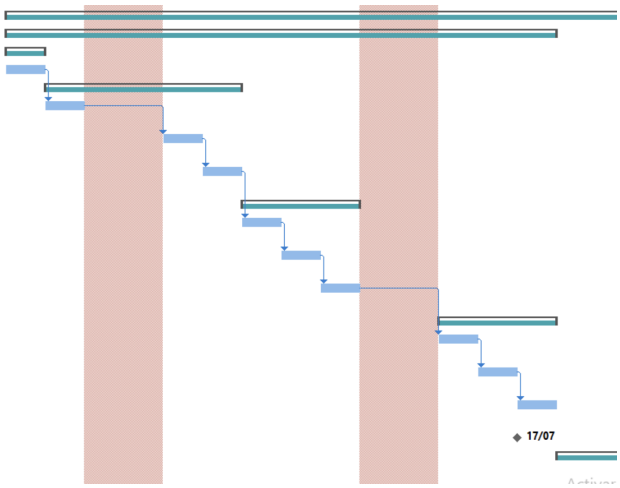


Figura 9: Fase construcción, primera iteración.

Iteración II:

Iteración II	**8 días?**	**jue 18/07/24**	**lun 29/07/24**
Modelado de Negocio	**1 día**	**jue 18/07/24**	**jue 18/07/24**
Reunión con los tutores	0,5 días	jue 18/07/24	jue 18/07/24
Revisar Documentación	0,5 días	jue 18/07/24	jue 18/07/24
Diseño	**1 día?**	**vie 19/07/24**	**vie 19/07/24**
Modelo de despliegue	1 día?	vie 19/07/24	vie 19/07/24
Implementación	**4 días**	**lun 22/07/24**	**jue 25/07/24**
Despliegue del sistema	4 días	lun 22/07/24	jue 25/07/24
Pruebas	**2 días**	**vie 26/07/24**	**lun 29/07/24**
Pruebas de integración de los módulos	1 día	vie 26/07/24	vie 26/07/24
Pruebas del sistema completo	1 día	lun 29/07/24	lun 29/07/24
Fin Iteración II	0 días	lun 29/07/24	lun 29/07/24

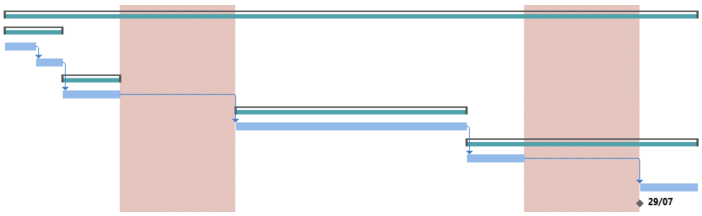


Figura 10: Fase construcción, segunda iteración.

Iteración III:

Iteración III	**11 días?**	**mar 30/07/24**	**mar 13/08/24**
Modelado de Negocio	**1 día?**	**mar 30/07/24**	**mar 30/07/24**
Reunión con los tutores	1 día	mar 30/07/24	mar 30/07/24
Requisitos	**2 días**	**mié 31/07/24**	**jue 01/08/24**
Hacer Anexo II	2 días	mié 31/07/24	jue 01/08/24
Análisis	**2 días**	**vie 02/08/24**	**lun 05/08/24**
Hacer anexo III	2 días	vie 02/08/24	lun 05/08/24
Diseño	**3 días**	**mar 06/08/24**	**jue 08/08/24**
Hacer anexo IV	3 días	mar 06/08/24	jue 08/08/24
Implementación	**2 días**	**vie 09/08/24**	**lun 12/08/24**
Comprobar errores en el sistema	2 días	vie 09/08/24	lun 12/08/24
Pruebas	**1 día?**	**mar 13/08/24**	**mar 13/08/24**
Pruebas para solventar los fallos encontrados	1 día?	mar 13/08/24	mar 13/08/24
Fin Iteración III	0 días	mar 13/08/24	mar 13/08/24
Fin contrucción	0 días	mar 13/08/24	mar 13/08/24

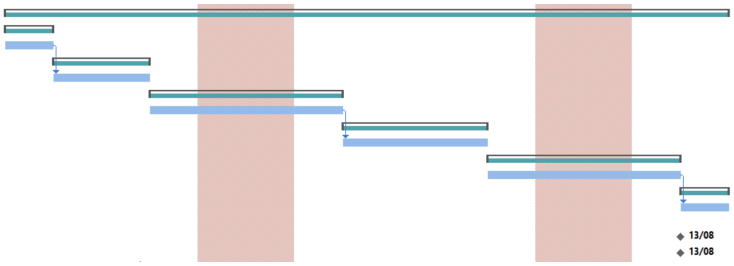


Figura 11: Fase construcción, tercera iteración.

Transición

Iteración I:

Transición	12 días?	mié 14/08/24	jue 29/08/24
Iteración I	12 días?	mié 14/08/24	jue 29/08/24
Modelado de Negocio	1 día?	mié 14/08/24	mié 14/08/24
Reunión con los tutores	1 día?	mié 14/08/24	mié 14/08/24
Requisitos	2 días	jue 15/08/24	vie 16/08/24
Terminar Anexo II	2 días	jue 15/08/24	vie 16/08/24
Análisis	2 días?	lun 19/08/24	mar 20/08/24
Terminar anexo III	2 días	lun 19/08/24	mar 20/08/24
Diseño	2 días?	mié 21/08/24	jue 22/08/24
Terminar anexo IV	2 días	mié 21/08/24	jue 22/08/24
Implementación	3 días?	vie 23/08/24	mar 27/08/24
Repasar código	3 días	vie 23/08/24	mar 27/08/24
Pruebas	2 días?	mié 28/08/24	jue 29/08/24
Pruebas finales del sistema completo	2 días	mié 28/08/24	jue 29/08/24
Fin Iteración I	0 días	jue 29/08/24	jue 29/08/24
Fin Transición	0 días	jue 29/08/24	jue 29/08/24

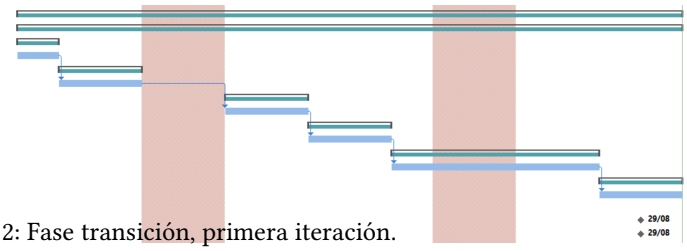


Figura 12: Fase transición, primera iteración.

3.3 Resultados

Tras llevar a cabo la planificación temporal, se ha determinado la duración del proyecto. El análisis muestra que el desarrollo del proyecto se completará en 676 horas, trabajando aproximádamente 8 horas cada dia.

Estadísticas del proyecto 'ProyectoTFG'

	Comienzo	Fin
Actual	mié 17/04/24	jue 29/08/24
Previsto	NOD	NOD
Real	NOD	NOD
Variación	0d	0d

	Duración	Trabajo	Costo
Actual	87d?	676h	0,00 €
Previsto	0d	0h	0,00 €
Real	0d	0h	0,00 €
Restante	87d?	676h	0,00 €

Porcentaje completado:

Duración: 0% Trabajo: 0%

Cerrar

Figura 13: Estadísticas de la planificación temporal.

Tras presentar la planificación temporal, se analiza la distribución de carga del proyecto, y se observa que el recurso presente (Marta de Castro) realiza el total de las horas de trabajo.

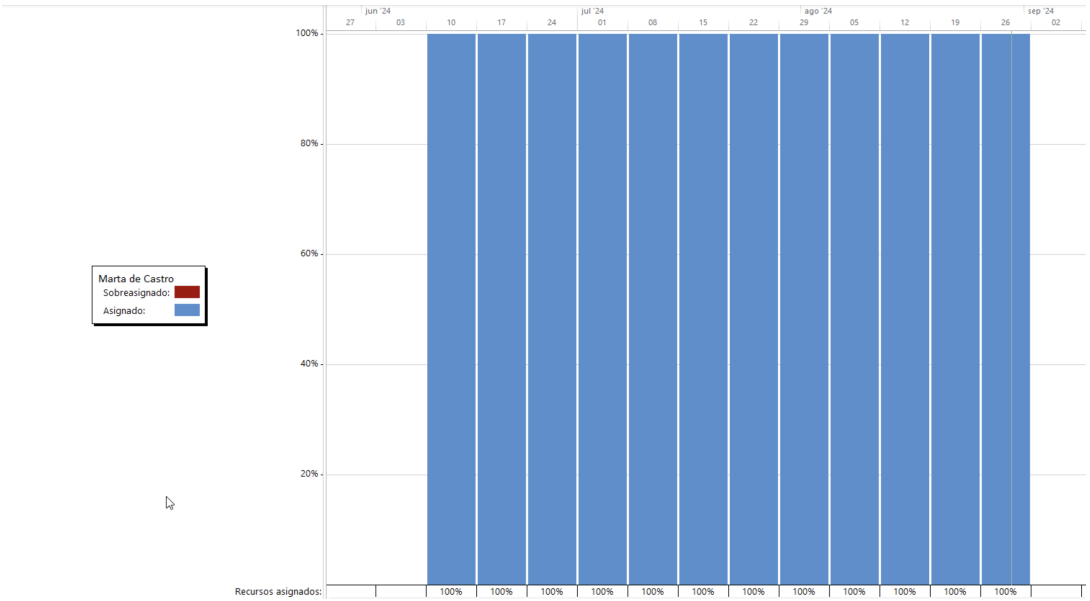


Figura 14: Carga de trabajo.

Bibliografía

[1] Moreno García, M. N. *Transparencias de Gestión de Proyectos, Práctica 1 – Estimación del esfuerzo.*

[2] Moreno García, M. N. *Transparencias de Gestión de Proyectos, Práctica 2 – Planificación temporal.*

[3] Microsoft. *Software de Gestión de Proyectos.* Recuperado de `https://www.microsoft.com/es-es/microsoft-365/project/project-management-software`.

Análisis de lectura de textos
Anexo II. Especificación de requisitos de software

Trabajo de Fin de Grado
GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:
Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),
Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

Índice general

1. Introducción **1**

2. Organización y participantes **3**

2.1. Organización 3

2.2. Participantes 4

3. Objetivos del proyecto **5**

4. Catálogo de requisitos del sistema **9**

4.1. Requisitos de información 9

4.2. Requisitos funcionales 15

4.2.1. Diagrama de paquetes 15

4.2.2. Definición de actores 15

4.2.3. Casos de uso del sistema 18

4.3. Requisitos no funcionales 55

5. Matriz de rastreabilidad **59**

Bibliografía **60**

Índice de figuras

1. Diagrama de paquetes. 15
2. Definicion de actores. 15
3. Paquete de autentificación de usuarios. 18
4. Paquete de gestión de usuarios. 22
5. Paquete de gestión de cursos. 28
6. Paquete de gestión de sesiones. 34
7. Paquete de gestión de actividades. 44
8. Paquete de gestión de resultados. 51

Índice de tablas

1. Organización. 3
2. Participantes. 4
3. OBJ-0001 Gestión de actividades. 5
4. OBJ-0002 Gestión de usuarios. 5
5. OBJ-0003 Gestión de historial de usuarios. 6
6. OBJ-0004 Asignación de actividades. 6
7. OBJ-0005 Almacenamiento de resultados y métricas. 6
8. OBJ-0006 Gestión de sesiones. 7
9. OBJ-0007 Interfaz de usuario. 7
10. OBJ-0008 Gestión de cursos. 7
11. IRQ-001 Información sobre usuarios. 10
12. IRQ-002 Información sobre cursos. 11
13. IRQ-003 Información sobre entrenamientos. 12
14. IRQ-004 Información sobre resultados. 13
15. IRQ-005 Información sobre actividades. 13
16. IRQ-006 Información sobre items. 14

17. ACT-01 Usuario no identificado. 16
18. ACT-02 Usuario identificado. 16
19. ACT-03 Participante. 16
20. ACT-04 Entrenador. 17
21. ACT-05 Administrador. 17
22. UC-001: Registrar usuario. 19
23. UC-002: Iniciar sesión. 20
24. UC-003: Cerrar sesión. 21
25. UC-004: Consultar datos personales. 23
26. UC-005: Modificar datos personales. 24
27. UC-006: Crear usuario. 25
28. UC-007: Modificar datos de un usuario. 26
29. UC-008: Eliminar usuario. 27
30. UC-009: Crear cursos. 29
31. UC-010: Listar usuarios por curso. 30
32. UC-011: Editar usuarios de curso. 31
33. UC-012: Eliminar curso. 32
34. UC-013: Asignar usuarios a cursos. 33
35. UC-014: Crear sesiones. 35
36. UC-015: Modificar sesiones. 36
37. UC-016: Eliminar sesiones. 37
38. UC-017: Asignar actividades. 38
39. UC-018: Asignar cursos. 39
40. UC-019: Listar sesiones. 40

41. UC-020: Filtrar por realización. 41
42. UC-021: Iniciar entrenamiento. 42
43. UC-022: Grabar entrenamiento. 43
44. UC-023: Crear actividades. 45
45. UC-024: Modificar actividades. 46
46. UC-025: Eliminar actividades. 47
47. UC-026: Crear items. 48
48. UC-027: Asignar items. 49
49. UC-028: Ver actividades. 50
50. UC-029: Importar tareas. 52
51. UC-030: Analizar audios. 53
52. UC-031: Actividades completadas. 54
53. NFR-001 Usabilidad. 55
54. NFR-002 Compatibilidad. 55
55. NFR-003 Fiabilidad. 56
56. NFR-004 Implementación. 56
57. NFR-005 Privacidad. 57

1 Introducción

En este anexo se describe de forma detallada la especificación de requisitos software [1] del sistema. Para ello, se sigue la metodología de Durán y Bernárdez [2] para la elicitación de requisitos del sistema, cuya estructura incluye:

- Organizaciones en el proyecto: se identifican y describen a las organizaciones involucradas en el desarrollo y uso del sistema, como las responsables del proyecto y las entidades beneficiarias.
- Participantes en el proyecto: se listan las personas y roles clave en el proyecto, como desarrolladores, gerentes de proyectos y usuarios finales, indicando sus responsabilidades.
- Objetivos del sistema: se definen los objetivos principales del sistema, describiendo qué se espera lograr con su implementación y cuáles son las metas a largo plazo.
- Catálogo de requisitos del sistema:
  - Requisitos de información: se detallan los datos y la información que el sistema deberá manejar, almacenar y procesar.
  - Requisitos funcionales:
    - Diagrama de paquetes: se muestra una visión general de los componentes del sistema y sus relaciones.
    - Definición de actores: se identifican los actores que interactúan con el sistema, como los usuarios y otros sistemas.
    - Casos de uso: se describen las interacciones típicas entre los actores y el sistema para cumplir con los requisitos funcionales.
  - Requisitos no funcionales: se especifican las restricciones y características que debe cumplir el sistema en términos de rendimiento, seguridad, usabilidad, etc.
- Matriz de rastreabilidad: se crea una matriz que permitirá rastrear la correspondencia entre los requisitos del sistema y los elementos del diseño, asegurando que todos los requisitos estén cubiertos por la implementación.

2 Organización y participantes

En este apartado se describen los diferentes participantes en el desarrollo del Trabajo de Fin de Grado. Han intervenido la estudiante del Grado de Ingeniería Informática con un tutor de la empresa Glifing SL y una tutora de la Universidad de Salamanca.

2.1 Organización

Campo	Valor
Organización	Universidad de Salamanca (USAL)
País	España
Provincia	Salamanca
Dirección	Plaza de los Caídos, s/n, 37008
Teléfono	923 29 44 50
Comentarios	-

Campo	Valor
Organización	Glifing SL
País	España
Provincia	Barcelona
Dirección	C/ Buenos Aires, 13, 08036 Barcelona
Teléfono	933 19 66 77
Comentarios	-

Tabla 1: Organización.

2.2 Participantes

Campo	Valor
Participante	Marta de Castro Leira
Organización	Universidad de Salamanca (USAL)
Rol	Desarrolladora
Es desarrollador	Sí
Es cliente	No
Es usuario	Sí
Comentarios	-

Campo	Valor
Participante	Jesús Manuel Nieto Carracedo
Organización	Glifing SL
Rol	Tutor del proyecto
Es desarrollador	No
Es cliente	No
Es usuario	No
Comentarios	-

Campo	Valor
Participante	Ángelica González Arrieta
Organización	Universidad de Salamanca (USAL)
Rol	Tutor del proyecto
Es desarrollador	No
Es cliente	No
Es usuario	No
Comentarios	-

Tabla 2: Participantes.

3 Objetivos del proyecto

OBJ-0001	Gestión de Actividades
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Crear y editar actividades de lectura y almacenar los resultados y métricas asociadas para su análisis externo.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 3: OBJ-0001 Gestión de actividades.

OBJ-0002	Gestión de Usuarios
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Almacenar y gestionar la información básica de los usuarios, como ID, nombre, email y otros datos relevantes.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 4: OBJ-0002 Gestión de usuarios.

OBJ-0003	Gestión de Historial de Usuarios
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Gestionar un historial de sesiones de trabajo de los usuarios, incluyendo ejercicios de lectura.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 5: OBJ-0003 Gestión de historial de usuarios.

OBJ-0004	Asignación de Actividades
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Permitir la asignación de actividades de lectura a los usuarios y gestionar su progreso en estas actividades.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 6: OBJ-0004 Asignación de actividades.

OBJ-0005	Almacenamiento de Resultados y Métricas
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Obtener y almacenar las métricas y estadísticas relacionadas con las sesiones de lectura para realizar un análisis detallado posterior.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 7: OBJ-0005 Almacenamiento de resultados y métricas.

OBJ-0006	Gestión de Sesiones
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Administrar las sesiones asignadas a los usuarios, incluyendo la creación, seguimiento y actualización del estado de las tareas.
Importancia	Media
Urgencia	Media
Comentarios	Ninguno

Tabla 8: OBJ-0006 Gestión de sesiones.

OBJ-0007	Interfaz de Usuario
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Diseñar una interfaz de usuario intuitiva y amigable que permita una navegación eficiente y fácil acceso a las funcionalidades del sistema.
Importancia	Alta
Urgencia	Alta
Comentarios	Ninguno

Tabla 9: OBJ-0007 Interfaz de usuario.

OBJ-0008	Gestión de Cursos
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Administrar los cursos asignado a los usuarios la creación, seguimiento y actualización.
Importancia	Media
Urgencia	Media
Comentarios	Ninguno

Tabla 10: OBJ-0008 Gestión de cursos.

4 Catálogo de requisitos del sistema

En este apartado se exponen los diferentes requisitos de información y funcionales necesarios para que el sistema cumpla con los objetivos establecidos.

4.1 Requisitos de información

A continuación, se detallan los requisitos de información que el sistema debe almacenar para gestionar adecuadamente los datos de usuarios, actividades y resultados.

IRQ-001	Información sobre Usuarios
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0002] Gestión de Usuarios
Descripción	El sistema debe almacenar la información básica de los usuarios registrados en la plataforma, incluyendo su identificador único, nombre, credenciales y datos relacionados con su cuenta de Glifing.
Datos específicos	Identificador único (ID) Usuario: formato nombre.apellido Nombre Apellidos Correo electrónico Rol: entrenador, entrenado, administrador Password: clave de acceso cifrada Avatar: imagen asociada al usuario
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 11: IRQ-001 Información sobre usuarios.

IRQ-002	Información sobre Cursos
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0001] Gestión de Actividades
Descripción	El sistema debe almacenar información relacionada con los cursos a los que tiene acceso el usuario entrenado, incluyendo las sesiones de entrenamiento asociadas.
Datos específicos	Identificador único (ID) Nombre Descripción
Importancia	Alta
Urgencia	Media
Estado	Validado
Estabilidad	Alta

Tabla 12: IRQ-002 Información sobre cursos.

IRQ-003	Información sobre Entrenamientos
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0001] Gestión de Actividades [OBJ-0003] Gestión de Historial de Usuarios [OBJ-0004] Asignación de Actividades [OBJ-0006] Gestión de Sesiones
Descripción	El sistema debe almacenar información sobre las sesiones de entrenamiento de los usuarios, incluyendo las actividades realizadas y sus resultados.
Datos específicos	Identificador único (ID) Nombre Número de actividades completadas Número total de actividades ID del usuario creador ID del usuario editor Fecha de creación Fecha de edición
Importancia	Alta
Urgencia	Media
Estado	Validado
Estabilidad	Alta

Tabla 13: IRQ-003 Información sobre entrenamientos.

IRQ-004	Información sobre Resultados
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0001] Gestión de Actividades [OBJ-0005] Almacenamiento de Resultados y Métricas
Descripción	El sistema debe almacenar los resultados generados por cada actividad realizada por los usuarios en las sesiones de entrenamiento.
Datos específicos	Identificador único (ID)
Importancia	Alta
Urgencia	Media
Estado	Validado
Estabilidad	Alta

Tabla 14: IRQ-004 Información sobre resultados.

IRQ-005	Información sobre Actividades
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0001] Gestión de Actividades [OBJ-0004] Asignación de Actividades [OBJ-0006] Gestión de Sesiones
Descripción	El sistema debe almacenar información sobre las actividades de lectura asignadas a las sesiones, detallando la configuración de cada actividad y su relación con los resultados de los usuarios.
Datos específicos	Identificador único (ID) Nombre Descripción Instrucción Tipo: texto completo o palabras sueltas
Importancia	Alta
Urgencia	Media
Estado	Validado
Estabilidad	Alta

Tabla 15: IRQ-005 Información sobre actividades.

IRQ-006	Información sobre Items
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-0001] Gestión de Actividades
Descripción	El sistema debe almacenar información sobre las actividades de lectura asignadas a las sesiones, detallando la configuración de cada actividad y su relación con los resultados de los usuarios.
Datos específicos	Identificador único (ID) Valor
Importancia	Alta
Urgencia	Media
Estado	Validado
Estabilidad	Alta

Tabla 16: IRQ-006 Información sobre items.

4.2 Requisitos funcionales

A continuación, se exponen los requisitos funcionales que el sistema debe cumplir para asegurar su correcto funcionamiento.

4.2.1 Diagrama de paquetes

En un proyecto de software, un diagrama de paquetes ayuda a dividir el sistema en componentes lógicos o subsistemas que agrupan clases relacionadas en función de su funcionalidad y responsabilidades.

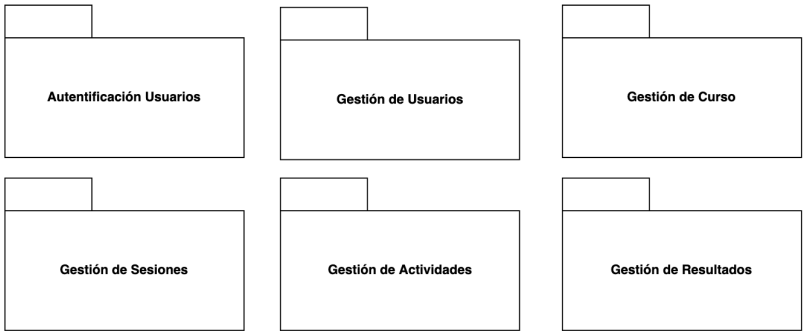


Figura 1: Diagrama de paquetes.

4.2.2 Definición de actores

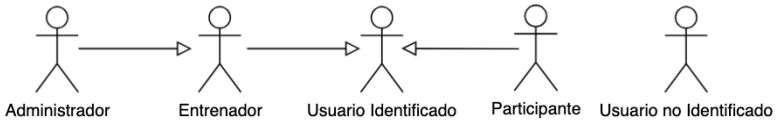


Figura 2: Definicion de actores.

ACT-01	Usuario no identificado
Versión	1.0
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Usuario que interactúa con el sistema sin haberse identificado o iniciado sesión.
Comentarios	Tiene acceso limitado a las funcionalidades del sistema y no puede realizar tareas administrativas o de edición.

Tabla 17: ACT-01 Usuario no identificado.

ACT-02	Usuario identificado
Versión	1.0
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Usuario que ejecuta las cargas de trabajo que le han sido asignadas, siguiendo un orden específico en una lista de tareas.
Comentarios	No tiene permisos de edición o administración, solo de ejecución de actividades asignadas.

Tabla 18: ACT-02 Usuario identificado.

ACT-03	Participante
Versión	1.0
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Usuario que puede ejecutar las sesiones y pertenecer a cursos.
Comentarios	No tiene permisos de edición o administración, solo de ejecución de actividades asignadas.

Tabla 19: ACT-03 Participante.

ACT-04	Entrenador
Versión	1.0
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Usuario capaz de administrar los paquetes, sesiones y actividades en la plataforma, así como asignar cargas de trabajo a los entrenados.
Comentarios	No puede gestionar usuarios, pero tiene control sobre la creación y asignación de actividades.

Tabla 20: ACT-04 Entrenador.

ACT-05	Administrador
Versión	1.0
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Descripción	Administrador que puede gestionar completamente a otros usuarios, acceder y revisar configuraciones en la plataforma, y acceder a los resultados de los usuarios.
Comentarios	Tiene permisos completos para la administración de la plataforma y usuarios.

Tabla 21: ACT-05 Administrador.

4.2.3 Casos de uso del sistema

Los casos de uso detallan las interacciones entre los actores y el sistema para alcanzar objetivos específicos. En esta sección, se presentan los diagramas de casos de uso y sus descripciones, que ilustran las funcionalidades del sistema y las acciones de los distintos tipos de usuarios.

Los casos de uso se organizan y describen utilizando el formato propuesto por Durán y Bernárdez [1], proporcionando una visión clara y estructurada de las capacidades del sistema. Los diagramas ofrecen una representación visual de las interacciones, mientras que las descripciones detallan los flujos de trabajo y operaciones clave.

Autentificación del usuario

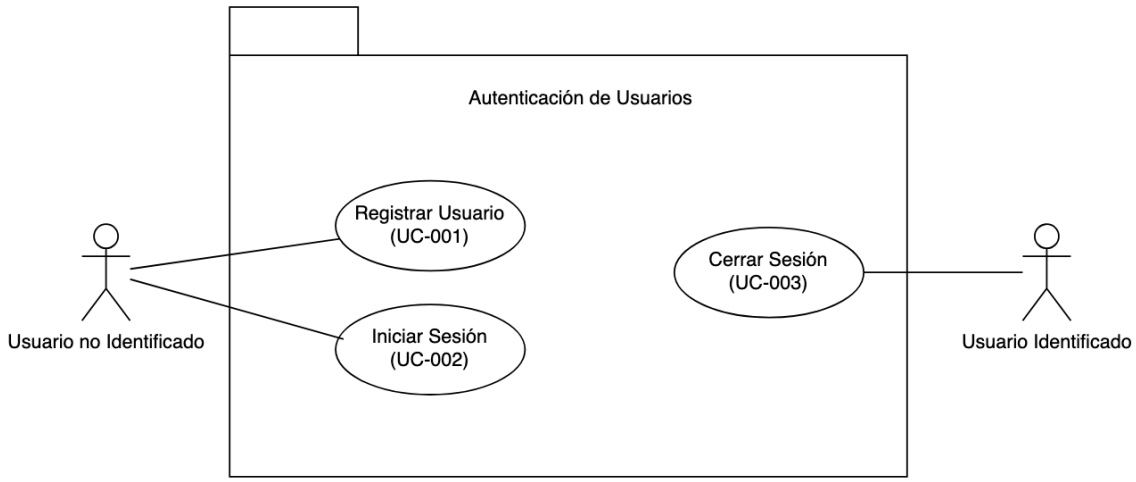


Figura 3: Paquete de autentificación de usuarios.

UC-001	**Registrar Usuario**
Versión	1.0 (03/07/2023)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El sistema solicita al usuario que introduzca sus datos personales y posteriormente se crea su cuenta.
Precondición	El usuario no ha iniciado sesión y no deben tener cuenta creada.
Secuencia normal	1. El actor Usuario No Identificado solicita el registro de su cuenta. 2. El sistema solicita al usuario que introduzca sus datos personales. 3. El actor Usuario No Identificado introduce los datos requeridos. 4. El sistema comprueba que los datos introducidos presentan el formato correcto, que se han introducido todos los obligatorios, y que el nombre del usuario no existe.
Postcondición	El usuario se crea en el sistema con los datos introducidos, y por defecto como Usuario Identificado
Excepciones	3. El actor Usuario No Identificado cancela el registro, este caso de uso queda sin efecto. 4. Los datos introducidos son incorrectos o los campos nombre de usuario y/o correo electrónico ya están registrados en el sistema, el sistema devuelve un mensaje de error y el caso de uso queda sin efecto.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 22: UC-001: Registrar usuario.

UC-002	**Iniciar Sesión**
Versión	1.0 (03/07/2023)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El sistema solicita al usuario que introduzca sus datos de autenticación e inicia sesión.
Precondición	El usuario no ha iniciado sesión pero tenga una cuenta creada.
Secuencia normal	1. El actor Usuario No Identificado solicita el inicio de sesión. 2. El sistema solicita al usuario que introduzca su usuario y contraseña. 3. El actor Usuario No Identificado introduce los datos requeridos. 4. El sistema comprueba que los datos introducidos son correctos y que el usuario existe en el sistema.
Postcondición	El actor Usuario No Identificado se transforma en el actor Usuario Identificado o Administrador, dependiendo de su rol.
Excepciones	3. El actor Usuario No Identificado cancela el inicio de sesión, este caso de uso queda sin efecto. 3. El actor Usuario No Identificado no completa alguno de los campos muestra un mensaje de error, por falta de datos. 4. Los datos introducidos son incorrectos o el usuario no existe en el sistema, el sistema devuelve un mensaje de error y el caso de uso queda sin efecto.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 23: UC-002: Iniciar sesión.

UC-003	**Cerrar Sesión**
Versión	1.0 (03/07/2023)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El sistema permite al usuario cerrar su sesión actual, terminando la autenticación en la aplicación.
Precondición	El usuario ha iniciado sesión en el sistema.
Secuencia normal	1. El actor Usuario Identificado solicita cerrar su sesión. 2. El sistema confirma la acción y procede a cerrar la sesión. 3. El sistema invalida la sesión actual del usuario. 4. El actor es redirigido a la página de inicio sin estar Identificado.
Postcondición	El usuario es desconectado del sistema y vuelve a ser considerado como Usuario No Identificado.
Excepciones	2. El actor Usuario Identificado decide no cerrar sesión, este caso de uso queda sin efecto.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 24: UC-003: Cerrar sesión.

Gestión de usuarios

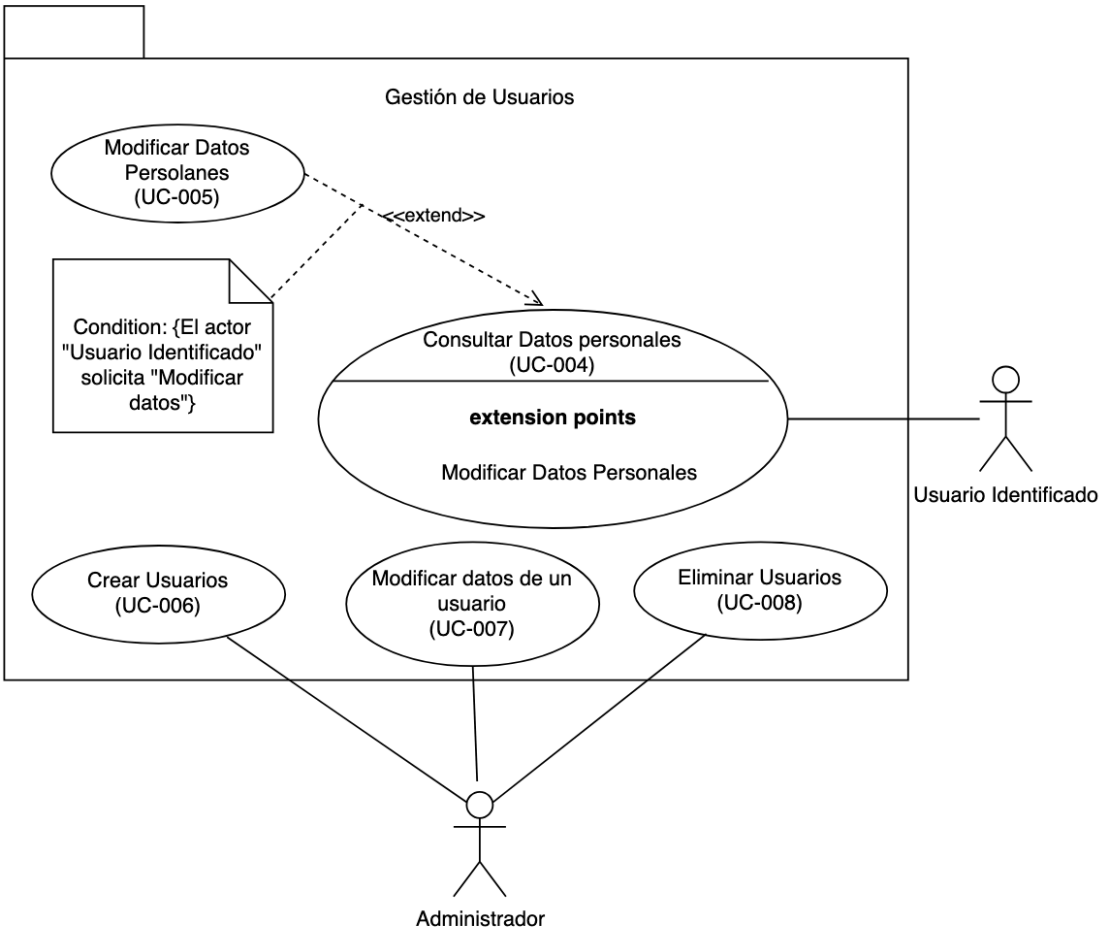


Figura 4: Paquete de gestión de usuarios.

UC-004	**Consultar Datos Personales**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El usuario puede consultar sus datos personales almacenados en el sistema.
Precondición	El usuario debe haber iniciado sesión.
Secuencia normal	1. El actor Usuario Identificado solicita consultar sus datos personales. 2. El sistema muestra los datos personales del usuario. 3. El sistema permite la modificación de los datos mediante la posible realización del caso de uso UC-005: modificar datos personales.
Postcondición	El usuario visualiza sus datos personales.
Excepciones	Ninguna.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 25: UC-004: Consultar datos personales.

UC-005	**Modificar Datos Personales**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El usuario puede modificar sus datos personales en el sistema.
Precondición	El usuario debe haber iniciado sesión.
Secuencia normal	1. El actor Usuario Identificado solicita modificar sus datos personales. 2. El sistema permite editar los datos personales. 3. El usuario guarda los cambios realizados. 4. El sistema actualiza los datos.
Postcondición	Los datos personales del usuario han sido actualizados en el sistema.
Excepciones	2. Si la información facilitada no cumple los requisitos establecidos, el sistema informará al usuario para que cambie los datos introducidos 3. El usuario cancela la modificación, no se realizan cambios.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 26: UC-005: Modificar datos personales.

UC-006	**Crear Usuario**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	El sistema debe permitir al usuario administrador, dar de alta nuevos usuarios en la plataforma.
Precondición	El usuario debe ser un Administrador.
Secuencia normal	1. El actor Administrador solicita crear un nuevo usuario. 2. El sistema solicita los datos del nuevo usuario. 3. El administrador ingresa los datos requeridos. 4. El sistema crea el nuevo usuario y lo almacena en la base de datos.
Postcondición	El nuevo usuario ha sido creado y está disponible en el sistema.
Excepciones	2. Si los datos no siguen el formato requerido o los usuarios ya existen se avisa al Administrador para que los corrija. 3. El administrador cancela el proceso, no se realiza la creación del usuario.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 27: UC-006: Crear usuario.

UC-007	**Modificar Datos de un Usuario**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	Un administrador puede modificar los datos de un usuario existente.
Precondición	El usuario debe ser un Administrador.
Secuencia normal	1. El actor Administrador selecciona un usuario del listado de usuarios y visualiza sus datos. 2. El actor Administrador solicita modificar los datos de un usuario. 3. El sistema permite editar los datos del usuario. 4. El administrador guarda los cambios realizados. 5. El sistema actualiza los datos del usuario en la base de datos.
Postcondición	Los datos del usuario han sido modificados y actualizados en el sistema.
Excepciones	3. Si los datos no siguen el formato requerido o los usuarios ya existen se avisa al Administrador para que los corrija. 4. El administrador cancela el proceso, no se realiza la creación del usuario.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 28: UC-007: Modificar datos de un usuario.

UC-008	**Eliminar Usuario**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-002] Gestión de Usuarios [OBJ-007] Interfaz de Usuario [IRQ-001] Información sobre Usuarios
Descripción	Un administrador puede eliminar usuarios del sistema.
Precondición	El usuario debe ser un Administrador.
Secuencia normal	1. El actor Administrador selecciona un usuario del listado de usuarios y visualiza sus datos. 2. El actor Administrador solicita eliminar a un usuario. 3. El sistema solicita confirmación para eliminar al usuario. 4. El administrador confirma la eliminación. 5. El sistema elimina al usuario de la base de datos.
Postcondición	El usuario ha sido eliminado del sistema.
Excepciones	4. El administrador cancela la eliminación, no se realizan cambios.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 29: UC-008: Eliminar usuario.

Gestión de cursos

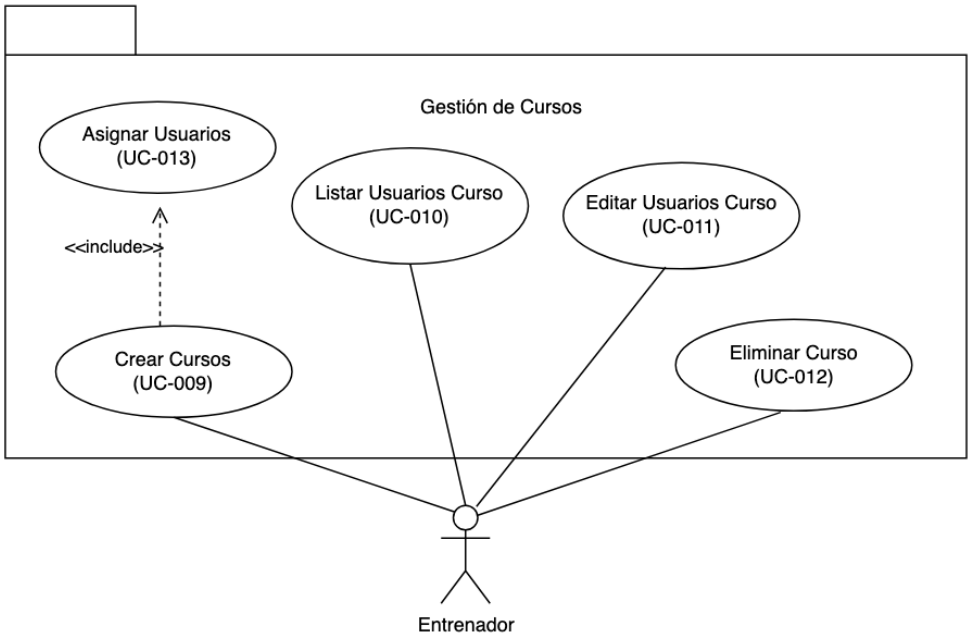


Figura 5: Paquete de gestión de cursos.

UC-009	**Crear Cursos**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-007] Interfaz de Usuario [OBJ-008] Gestión de Cursos [IRQ-002] Información sobre Cursos
Descripción	Un administrador puede crear nuevos cursos en el sistema.
Precondición	El usuario debe ser un Entrenador o Administrador.
Secuencia normal	1. El actor Administrador/Entrenador solicita la creación de un nuevo curso. 2. El sistema solicita los datos del nuevo curso. 3. El administrador/entrenador ingresa los datos requeridos. 4. El sistema realiza el caso de uso UC-013: asignar Usuarios a Cursos 5. El sistema crea el nuevo curso y lo almacena en la base de datos.
Postcondición	El nuevo curso ha sido creado y está disponible en el sistema.
Excepciones	2. Si los datos no siguen el formato requerido o si el curso ya existe se avisa al usuario que solicita la creación para que los corrija. 3. El actor cancela el proceso, no se realiza la creación del curso.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 30: UC-009: Crear cursos.

UC-010	**Listar Usuarios por Curso**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-003] Gestión de Historial de Usuarios [OBJ-007] Interfaz de Usuario [OBJ-008] Gestión de Cursos [IRQ-002] Información sobre Cursos
Descripción	Un administrador puede consultar la lista de usuarios asignados a un curso específico.
Precondición	El usuario debe ser un Entrenador o Administrador y debe existir al menos un curso
Secuencia normal	1. El actor solicita ver la lista de usuarios asignados a un curso. 2. El sistema muestra los cursos disponibles y para cada curso los usuarios que hay.
Postcondición	La lista de usuarios asignados al curso ha sido mostrada al administrador.
Excepciones	2. Un curso que no tiene usuarios asignados aparece con el hueco para los usuarios en blanco.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 31: UC-010: Listar usuarios por curso.

UC-011	**Editar Usuarios de Curso**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-007] Interfaz de Usuario [OBJ-008] Gestión de Cursos [IRQ-002] Información sobre Cursos
Descripción	Un administrador puede modificar la lista de usuarios asignados a un curso.
Precondición	El usuario debe ser un Entrenador o Administrador y debe existir al menos un curso creado
Secuencia normal	1. El actor solicita Editar los usuarios asignados a un curso. 2. El sistema muestra los usuarios actualmente asignados al curso. 3. El actor agrega o elimina usuarios de la lista asignada. 4. El sistema actualiza la lista de usuarios asignados al curso.
Postcondición	La lista de usuarios asignados al curso ha sido actualizada en el sistema.
Excepciones	3. El administrador cancela la modificación, no se realizan cambios.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 32: UC-011: Editar usuarios de curso.

UC-012	**Eliminar Curso**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-007] Interfaz de Usuario [OBJ-008] Gestión de Cursos [IRQ-002] Información sobre Cursos
Descripción	Un administrador puede eliminar cursos del sistema.
Precondición	El usuario debe ser un Entrenador o Administrador y debe existir al menos un curso creado.
Secuencia normal	1. El actor solicita eliminar un curso. 2. El sistema solicita confirmación para eliminar el curso. 3. El actor confirma la eliminación. 4. El sistema elimina el curso de la base de datos.
Postcondición	El curso ha sido eliminado del sistema.
Excepciones	3. El administrador cancela la eliminación, no se realizan cambios.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 33: UC-012: Eliminar curso.

UC-013	**Asignar Usuarios a Cursos**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-004] Asignación de actividades [OBJ-007] Interfaz de Usuario [OBJ-008] Gestión de Cursos [IRQ-002] Información sobre Cursos
Descripción	Un administrador puede asignar usuarios a los cursos creados en el sistema.
Precondición	El usuario debe ser un Entrenador o Administrador y debe estar Creando un Curso
Secuencia normal	1. El usuario solicita asignar usuarios a un curso. 2. El sistema muestra los usuarios Identificados que pueden formar parte del curso. 3. El Entrenador selecciona los usuarios que serán asignados. 4. El sistema asigna a los usuarios seleccionados al curso correspondiente.
Postcondición	Los usuarios han sido asignados al curso seleccionado.
Excepciones	3. El administrador cancela la asignación, no se realizan cambios.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 34: UC-013: Asignar usuarios a cursos.

Gestión de sesiones

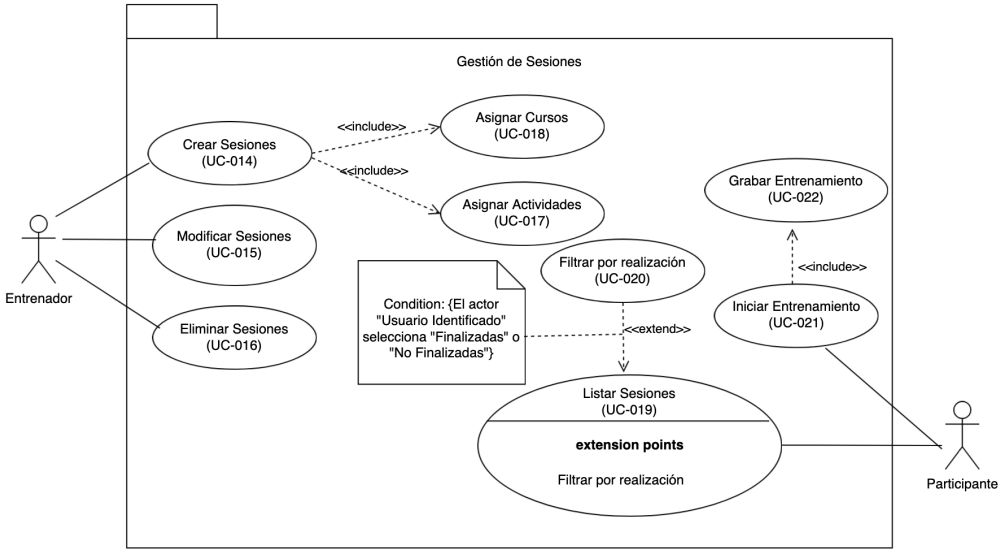


Figura 6: Paquete de gestión de sesiones.

UC-014	**Crear Sesiones**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos
Descripción	El entrenador puede crear nuevas sesiones de entrenamiento con las actividades necesarias.
Precondición	El usuario debe ser un entrenador identificado.
Secuencia normal	1. El actor Entrenador accede a la opción de crear sesiones en el sistema. 2. El sistema solicita los datos necesarios para la creación de la sesión (nombre, fecha, actividades, etc.). 3. El entrenador completa los datos requeridos. 4. El sistema realiza el caso de uso UC-017: asignar Actividades 5. El sistema registra la nueva sesión y la almacena en la base de datos. 6. El entrenador recibe una confirmación de la creación exitosa.
Postcondición	La sesión es creada y almacenada en el sistema.
Excepciones	3. El actor Entrenador cancela la creación de la sesión, no se realiza ningún cambio. 3. Los datos introducidos son incompletos o incorrectos, el sistema muestra un mensaje de error.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 35: UC-014: Crear sesiones.

UC-015	**Modificar Sesiones**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos
Descripción	El entrenador puede modificar una sesión de entrenamiento previamente creada, actualizando su información.
Precondición	El usuario debe ser un entrenador identificado.
Secuencia normal	1. El actor Entrenador selecciona una sesión previamente creada para modificarla. 2. El sistema muestra los detalles actuales de la sesión. 3. El entrenador edita la información deseada (nombre, fecha, actividades, etc.). 4. El sistema actualiza la sesión con los nuevos datos ingresados. 5. El entrenador recibe una confirmación de la modificación exitosa.
Postcondición	La sesión es modificada con éxito y los nuevos datos son guardados en el sistema.
Excepciones	3. El actor Entrenador cancela la modificación de la sesión, no se realiza ningún cambio. 3. Los datos introducidos son incompletos o incorrectos, el sistema muestra un mensaje de error.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 36: UC-015: Modificar sesiones.

UC-016	**Eliminar Sesiones**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos
Descripción	El entrenador puede eliminar una sesión de entrenamiento existente del sistema.
Precondición	El usuario debe ser un entrenador (ACT-004) identificado.
Secuencia normal	1. El actor Entrenador selecciona una sesión para eliminar. 2. El sistema solicita confirmación para proceder con la eliminación. 3. El entrenador confirma la eliminación de la sesión. 4. El sistema elimina la sesión seleccionada de la base de datos. 5. El entrenador recibe una confirmación de la eliminación exitosa.
Postcondición	La sesión seleccionada es eliminada del sistema.
Excepciones	3. El actor Entrenador cancela la eliminación de la sesión, no se realiza ningún cambio. 4. Ocurre un error en la eliminación de la sesión, el sistema muestra un mensaje de error.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 37: UC-016: Eliminar sesiones.

UC-017	Asignar Actividades
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-004] Asignación de Actividades [OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos [UC-014] Crear Sesiones
Descripción	El entrenador puede asignar actividades a una sesión existente.
Precondición	El usuario debe ser un entrenador (ACT-004) identificado y haber creado una sesión.
Secuencia normal	1. El actor Entrenador selecciona una sesión previamente creada para asignar actividades. 2. El sistema muestra las actividades disponibles para la asignación. 3. El entrenador selecciona las actividades que desea asignar a la sesión. 4. El sistema asocia las actividades seleccionadas con la sesión correspondiente. 5. El entrenador recibe una confirmación de la asignación exitosa.
Postcondición	Las actividades seleccionadas son asignadas a la sesión.
Excepciones	3. El actor Entrenador cancela la asignación, no se realiza ningún cambio. 4. Ocurre un error durante la asignación de actividades, el sistema muestra un mensaje de error.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 38: UC-017: Asignar actividades.

UC-018	**Asignar Curso**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [OBJ-008] Gestión de Cursos [IRQ-003] Información sobre Entrenamientos [UC-014] Crear Sesiones
Descripción	El entrenador puede asignar cursos a una sesión existente.
Precondición	El usuario debe ser un entrenador (ACT-004) identificado y haber creado una sesión.
Secuencia normal	1. El actor Entrenador selecciona una sesión previamente creada para asignar cursos. 2. El sistema muestra los cursos disponibles para la asignación. 3. El entrenador selecciona los cursos que desea asignar a la sesión. 4. El sistema asocia los cursos seleccionados con la sesión correspondiente. 5. El entrenador recibe una confirmación de la asignación exitosa.
Postcondición	Los cursos seleccionados son asignados a la sesión.
Excepciones	3. El actor Entrenador cancela la asignación, no se realiza ningún cambio. 4. Ocurre un error durante la asignación de cursos, el sistema muestra un mensaje de error.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 39: UC-018: Asignar cursos.

UC-019	**Listar Sesiones**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-003] Gestión de Historial de Usuarios [OBJ-005] Almacenamiento de Resultados y Métricas [OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos
Descripción	Los usuarios pueden consultar un listado de todas las sesiones creadas en el sistema.
Precondición	El usuario debe ser un Participante.
Secuencia normal	1. El Participante solicita ver el listado de sesiones. 2. El sistema muestra una lista con todas las sesiones disponibles en el sistema. 3. El usuario revisa las sesiones. 4. El sistema permite seleccionar una sesión para más detalles.
Postcondición	El usuario ha consultado el listado de sesiones.
Excepciones	2. No hay sesiones disponibles para mostrar, el sistema informa de la situación.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	La opción de filtrar las sesiones se incluye como un punto de extensión (UC-020).

Tabla 40: UC-019: Listar sesiones.

UC-020	**Filtrar por Realización**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos [UC-019] Listar Sesiones
Descripción	Los usuarios pueden filtrar las sesiones por su estado de realización, mostrando finalizadas o no finalizadas.
Precondición	El usuario debe haber solicitado el listado de sesiones (UC-019).
Secuencia normal	1. El actor Participante selecciona la opción para filtrar las sesiones por estado de realización. 2. El sistema muestra las opciones de filtro: "Finalizadas" y "No Finalizadas". 3. El usuario selecciona el filtro deseado. 4. El sistema actualiza la lista de sesiones según el filtro aplicado.
Postcondición	La lista de sesiones se actualiza según el filtro seleccionado.
Excepciones	4. Ocurre un error en la aplicación del filtro, el sistema muestra un mensaje de error.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 41: UC-020: Filtrar por realización.

UC-021	**Iniciar Entrenamiento**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos
Descripción	El usuario identificado puede iniciar un entrenamiento basado en las sesiones disponibles en la plataforma.
Precondición	El usuario ser Participante y haber seleccionado una sesión de entrenamiento.
Secuencia normal	1. El usuario selecciona la opción de iniciar entrenamiento. 2. El sistema inicia el entrenamiento y registra el estado de la sesión. 3. El usuario realiza el entrenamiento según las actividades de la sesión seleccionada.
Postcondición	El entrenamiento se completa y el sistema queda listo para registrar los resultados de la sesión.
Excepciones	2. Si ocurre un error al iniciar el entrenamiento, el sistema muestra un mensaje de error y no inicia la sesión.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Este caso de uso incluye la opción de grabar el entrenamiento, lo cual se detalla en [UC-022: Grabar Entrenamiento].

Tabla 42: UC-021: Iniciar entrenamiento.

UC-022	**Grabar Entrenamiento**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-006] Gestión de Sesiones [OBJ-007] Interfaz de Usuario [IRQ-003] Información sobre Entrenamientos [UC-021] Iniciar Entrenamiento
Descripción	El sistema graba el entrenamiento realizado por el usuario, registrando todos los datos pertinentes para su posterior análisis.
Precondición	El usuario debe haber iniciado un entrenamiento (UC-021).
Secuencia normal	1. El usuario Participante selecciona la opción de grabar el entrenamiento. 2. El sistema registra los datos del entrenamiento en curso. 3. El entrenamiento se almacena en el sistema para su posterior análisis y revisión.
Postcondición	El entrenamiento queda registrado y disponible en el sistema para su revisión y análisis futuros.
Excepciones	
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 43: UC-022: Grabar entrenamiento.

Gestión de actividades

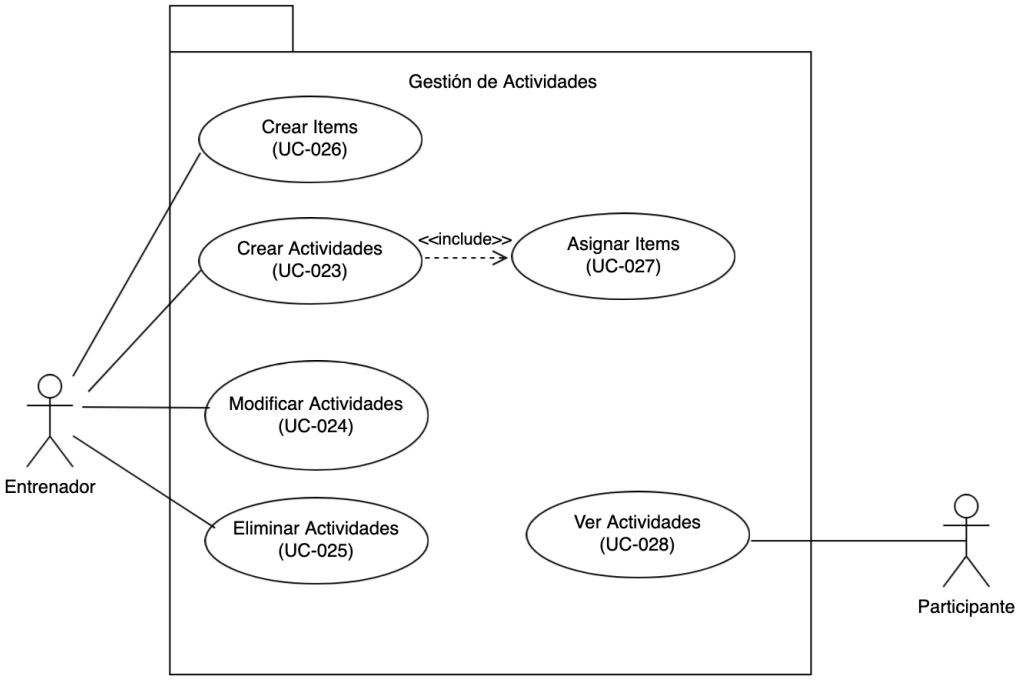


Figura 7: Paquete de gestión de actividades.

UC-023	Crear Actividades
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-007] Interfaz de Usuario [IRQ-005] Información sobre Actividades
Descripción	El entrenador crea nuevas actividades en el sistema, asignando los detalles necesarios.
Precondición	El actor debe ser un entrenador identificado.
Secuencia normal	1. El actor Entrenador accede a la opción de crear actividades en el sistema. 2. El sistema solicita los datos necesarios para la creación de la actividad. 3. El entrenador completa los datos requeridos. 4. El sistema realiza el caso de uso UC-017: asignar Items. 5. El sistema valida los datos e inserta la nueva actividad en el sistema.
Postcondición	La actividad queda registrada en el sistema y lista para ser modificada o asignada.
Excepciones	4. Si hay un error en los datos, el sistema muestra un mensaje de error y la actividad no se crea.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 44: UC-023: Crear actividades.

UC-024	Modificar Actividades
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-007] Interfaz de Usuario [IRQ-005] Información sobre Actividades [UC-023] Crear Actividades
Descripción	El entrenador puede modificar las actividades creadas previamente en el sistema.
Precondición	El actor debe ser un entrenador identificado (ACT-003) y la actividad debe existir.
Secuencia normal	1. El entrenador selecciona la actividad a modificar. 2. El sistema muestra los detalles actuales de la actividad. 3. El entrenador modifica los detalles necesarios y envía los cambios. 4. El sistema valida los datos y actualiza la actividad en el sistema.
Postcondición	La actividad queda actualizada en el sistema.
Excepciones	4. Si hay un error en los datos, el sistema muestra un mensaje de error y no actualiza la actividad.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 45: UC-024: Modificar actividades.

UC-025	**Eliminar Actividades**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-007] Interfaz de Usuario [IRQ-005] Información sobre Actividades [UC-023] Crear Actividades
Descripción	El entrenador puede eliminar actividades del sistema.
Precondición	El actor debe ser un entrenador identificado (ACT-003) y la actividad debe existir.
Secuencia normal	1. El entrenador selecciona la actividad a eliminar. 2. El sistema solicita confirmación de eliminación. 3. El entrenador confirma la eliminación. 4. El sistema elimina la actividad del sistema.
Postcondición	La actividad se elimina del sistema.
Excepciones	4. Si hay un error durante la eliminación, el sistema muestra un mensaje de error.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 46: UC-025: Eliminar actividades.

UC-026	**Crear Items**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-007] Interfaz de Usuario [IRQ-006] Información sobre Items
Descripción	El entrenador puede crear nuevos items en el sistema, necesarios para las actividades.
Precondición	El actor debe ser un entrenador identificado.
Secuencia normal	1. El entrenador solicita crear un nuevo item. 2. El sistema presenta un formulario para introducir los detalles del item. 3. El entrenador rellena el formulario y lo envía. 4. El sistema valida los datos e inserta el nuevo item en el sistema.
Postcondición	El item queda registrado en el sistema y listo para ser asignado a actividades.
Excepciones	4. Si hay un error en los datos, el sistema muestra un mensaje de error y el item no se crea.
Importancia	Alta
Urgencia	Alta
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 47: UC-026: Crear items.

UC-027	**Asignar Items**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-007] Interfaz de Usuario [IRQ-006] Información sobre Items [UC-026] Crear Items
Descripción	El entrenador asigna items a actividades específicas en el sistema.
Precondición	El actor debe ser un entrenador identificado (ACT-003) y los items deben estar creados.
Secuencia normal	1. El entrenador selecciona la actividad a la cual asignar items. 2. El sistema muestra los items disponibles. 3. El entrenador asigna los items deseados a la actividad. 4. El sistema valida los cambios y actualiza la actividad con los nuevos items asignados.
Postcondición	Los items quedan asignados a la actividad correspondiente.
Excepciones	4. Si ocurre un error en la asignación, el sistema muestra un mensaje de error y no realiza los cambios.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 48: UC-027: Asignar items.

UC-028	**Ver Actividades**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-001] Gestión de Actividades [OBJ-007] Interfaz de Usuario [IRQ-005] Información sobre Actividades
Descripción	Los usuarios identificados pueden ver una lista de las actividades disponibles.
Precondición	El usuario debe ser Participante.
Secuencia normal	1. El usuario solicita ver las actividades. 2. El sistema muestra la lista de actividades disponibles. 3. El usuario visualiza las actividades.
Postcondición	El usuario obtiene la información de las actividades disponibles.
Excepciones	2. Si ocurre un error en la carga de actividades, el sistema muestra un mensaje de error.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 49: UC-028: Ver actividades.

Gestión de resultados

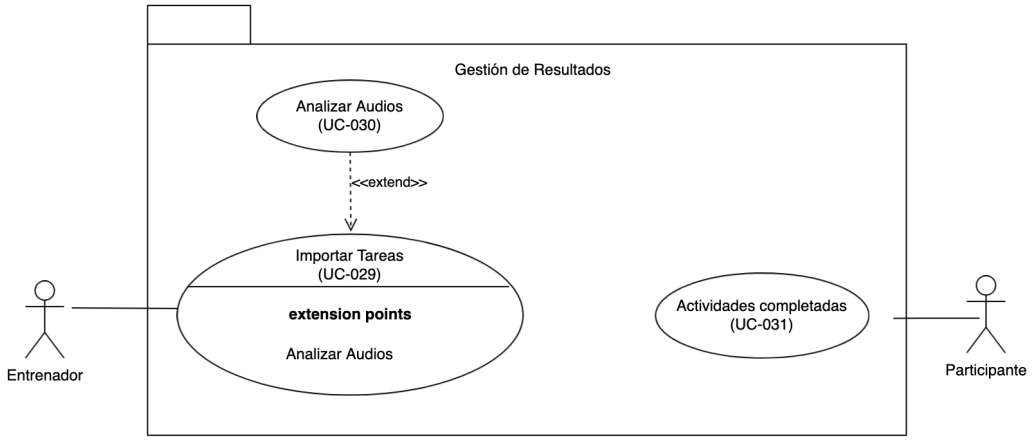


Figura 8: Paquete de gestión de resultados.

UC-029	Importar Tareas
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-005] Almacenamiento de Resultados y Métricas [OBJ-007] Interfaz de Usuario [IRQ-004] Información sobre Resultados
Descripción	El entrenador importa tareas al sistema desde una fuente externa.
Precondición	El actor debe ser un entrenador identificado (ACT-003).
Secuencia normal	1. El entrenador selecciona la opción de importar tareas. 2. El sistema solicita el archivo o la fuente desde donde importar las tareas. 3. El entrenador proporciona el archivo o fuente. 4. El sistema procesa y registra las tareas en el sistema. 5. El sistema permite la ejecución de estas tareas para el análisis de los audios almacenados mediante la posible realización del caso de uso UC-030: análisis de Audios.
Postcondición	Las tareas quedan registradas y listas para ser utilizadas.
Excepciones	4. Si hay un error en el archivo o en la fuente, el sistema muestra un mensaje de error y no importa las tareas.
Importancia	Alta
Urgencia	Media
Estabilidad	Alta
Comentarios	Este caso de uso puede ser extendido por el caso de uso [UC-030] Analizar Audios.

Tabla 50: UC-029: Importar tareas.

UC-030	**Analizar Audios**
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-005] Almacenamiento de Resultados y Métricas [OBJ-007] Interfaz de Usuario [IRQ-004] Información sobre Resultados [UC-029] Importar Tareas
Descripción	El entrenador puede analizar audios importados como tareas en el sistema.
Precondición	El actor debe ser un entrenador identificado (ACT-003) y deben existir tareas con audios importados.
Secuencia normal	1. El entrenador selecciona la opción de analizar audios. 2. El sistema tiene la opción de importar un audio o mostrar la lista de audios disponibles. 3. El entrenador selecciona un audio para analizar. 4. El sistema realiza el análisis del audio resultado de la tarea importada en el caso de uso UC-029 y muestra los resultados.
Postcondición	El entrenador obtiene el análisis del audio seleccionado.
Excepciones	4. Si hay un error en el análisis, el sistema muestra un mensaje de error y no completa el análisis.
Importancia	Alta
Urgencia	Alta
Estabilidad	Media
Comentarios	Este caso de uso extiende a [UC-029] Importar Tareas.

Tabla 51: UC-030: Analizar audios.

UC-031	Actividades Completadas
Versión	1.0 (29/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	[OBJ-005] Almacenamiento de Resultados y Métricas [OBJ-007] Interfaz de Usuario [IRQ-004] Información sobre Resultados
Descripción	El usuario identificado puede ver una lista de las actividades que ha completado.
Precondición	El usuario debe ser Participante.
Secuencia normal	1. El usuario solicita ver las actividades completadas. 2. El sistema muestra la lista de actividades completadas por el usuario. 3. El usuario visualiza las actividades completadas.
Postcondición	El usuario obtiene la información de las actividades completadas.
Excepciones	2. Si ocurre un error en la carga de actividades, el sistema muestra un mensaje de error.
Importancia	Media
Urgencia	Media
Estabilidad	Alta
Comentarios	Ninguno.

Tabla 52: UC-031: Actividades completadas.

4.3 Requisitos no funcionales

Los requisitos no funcionales son aspectos del sistema que no están directamente relacionados con las funciones específicas que el sistema debe realizar, sino que se refieren a las propiedades que el sistema debe tener. Estos requisitos incluyen características como la usabilidad, la seguridad, el rendimiento y la compatibilidad, y son esenciales para asegurar que el sistema funcione correctamente en diferentes entornos y situaciones.

NFR-001	**Usabilidad**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	Ninguna
Descripción	El sistema debe presentar una interfaz intuitiva y amigable, permitiendo a los usuarios llevar a cabo sus tareas con facilidad y sin necesidad de formación extensiva.
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 53: NFR-001 Usabilidad.

NFR-002	**Compatibilidad**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	Ninguna
Descripción	La aplicación web debe ser compatible con los principales navegadores web (Chrome, Firefox, Safari, Edge) y versiones actuales.
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 54: NFR-002 Compatibilidad.

NFR-003	Fiabilidad
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	Ninguna
Descripción	El sistema debe ser capaz de recuperarse de fallos y caídas, asegurando la consistencia de la información y minimizando el tiempo de inactividad.
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 55: NFR-003 Fiabilidad.

NFR-004	Implementación
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	Ninguna
Descripción	El sistema debe ofrecer una aplicación web accesible sin necesidad de instalación local, facilitando su acceso desde cualquier navegador compatible.
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 56: NFR-004 Implementación.

NFR-005	**Privacidad**
Versión	1.0 (26/08/2024)
Autores	Marta de Castro Leira
Fuentes	Jesús Manuel Nieto Carracedo Ángelica González Arrieta
Dependencias	Ninguna
Descripción	El sistema debe cumplir con la Ley Orgánica 3/2018 de Protección de Datos Personales y garantizar los derechos digitales de los usuarios.
Importancia	Alta
Urgencia	Alta
Estado	Validado
Estabilidad	Alta

Tabla 57: NFR-005 Privacidad.

5 Matriz de rastreabilidad

La matriz de rastreabilidad es una herramienta clave para garantizar que todos los requisitos del proyecto se cumplan de manera efectiva. Esta matriz proporciona una visión general de cómo cada caso de uso (UC) está vinculado a los objetivos específicos del proyecto (OBJ). A continuación, se presenta la matriz de rastreabilidad, donde se muestran los casos de uso y su relación con los diferentes objetivos del proyecto. Cada celda de la tabla indica si un caso de uso particular está asociado con un objetivo específico.

UC	**OBJ-01**	**OBJ-02**	**OBJ-03**	**OBJ-04**	**OBJ-05**	**OBJ-06**	**OBJ-07**	**OBJ-08**
UC-01		X					X	
UC-02		X					X	
UC-03		X					X	
UC-04		X					X	
UC-05		X					X	
UC-06		X					X	
UC-07		X					X	
UC-08		X					X	
UC-09							X	X
UC-10			X				X	X
UC-11							X	X
UC-12							X	X
UC-13				X			X	X
UC-14						X	X	
UC-15						X	X	
UC-16						X	X	
UC-17	X			X		X	X	
UC-18						X	X	X
UC-19			X		X	X	X	

UC	OBJ-01	OBJ-02	OBJ-03	OBJ-04	OBJ-05	OBJ-06	OBJ-07	OBJ-08
UC-20						X	X	
UC-21						X	X	
UC-22						X	X	
UC-23	X						X	
UC-24	X						X	
UC-25	X						X	
UC-26	X						X	
UC-27	X						X	
UC-28	X						X	
UC-29					X		X	
UC-30					X		X	
UC-31					X		X	

Tabla 58: Matriz de rastreabilidad.

Bibliografía

[1] García Peñalvo, F. J., Moreno García, M. N. *Transparencias de Ingeniería del Software I, UML.*

[2] Durán Toro, A., & Bernárdez Jiménez, B. (2000). *Metodología para la Elicitación de Requisitos de Sistemas Software* (Versión 2.1). Informe Técnico LSI–2000–10. Departamento de Lenguajes y Sistemas Informáticos, Facultad de Informática y Estadística, Sevilla.

VNiVERSiDAD
D SALAMANCA

Análisis de lectura de textos
Anexo III. Análisis del sistema software

Trabajo de Fin de Grado
GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:
Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),
Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

Índice general

1. Introducción **1**

2. Modelo de dominio **3**

3. Paquete de análisis y de servicio **5**

3.1. Clases de análisis 6

4. Realización de casos de uso en el modelo de análisis **11**

4.1. Paquete de gestión de autenticación de usuarios 11

4.2. Paquete de análisis de gestión de usuarios 13

4.3. Paquete de análisis de gestión de cursos 18

4.4. Paquete de análisis de gestión de sesiones 23

4.5. Paquete de análisis de gestión de actividades. 29

4.6. Paquete de análisis de gestión de resultados 34

5. Propuesta de la arquitectura **37**

Bibliografía **37**

Índice de figuras

1. Modelo de dominio. 3

2. Diagrama de paquetes de análisis. 5

3. Paquete de análisis de gestión de autenticación de usuarios 6

4. Paquete de análisis de gestión de usuarios. 7

5. Paquete de análisis de gestión de sesiones. 7

6. Paquete de análisis de gestión de cursos. 8

7. Paquete de análisis de gestión de actividades. 8

8. Paquete de análisis de gestión de resultados. 9

9. Diagrama de secuencia de registrar usuario 11

10. Diagrama de secuencia de iniciar sesión. 12

11. Diagrama de secuencia de cerrar sesión. 12

12. Diagrama de secuencia de consultar datos personales. 13

13. Diagrama de secuencia de modificar datos personales. 14

14. Diagrama de secuencia de crear usuarios. 15

15. Diagrama de secuencia de modificar datos de un usuario. 16

16. Diagrama de secuencia de eliminar usuario 17

17. Diagrama de secuencia de crear cursos. 18
18. Diagrama de secuencia de listar usuarios por curso. 19
19. Diagrama de Secuencia de Editar Usuarios por Curso 20
20. Diagrama de secuencia de eliminar curso. 21
21. Diagrama de secuencia de asignar usuarios a curso. 22
22. Diagrama de secuencia de crear sesión. 23
23. Diagrama de secuencia de modificar sesión. 24
24. Diagrama de secuencia de eliminar sesión. 25
25. Diagrama de secuencia de asignar actividad a sesiones. 26
26. Diagrama de secuencia de asignar curso a sesiones. 26
27. Diagrama de secuencia de listar sesiones. 27
28. Diagrama de secuencia de filtrar por realización. 27
29. Diagrama de secuencia de iniciar entrenamiento. 28
30. Diagrama de secuencia de grabar entrenamiento. 28
31. Diagrama de secuencia de crear actividad. 29
32. Diagrama de secuencia de modificar actividades. 30
33. Diagrama de secuencia de eliminar actividades. 31
34. Diagrama de secuencia de crear items. 32
35. Diagrama de secuencia de asignar items. 33
36. Diagrama de secuencia de ver actividades. 33
37. Diagrama de secuencia de importar tareas. 34
38. Diagrama de secuencia de analizar audio. 35
39. Diagrama de secuencia de actividades completadas. 36
40. Descripción de la arquitectura del modelo de análisis. 37

1 Introducción

Este documento ofrece una visión detallada del análisis y diseño del sistema, basándose en los requisitos especificados en el **Anexo II: Especificación de requisitos de software**.

El contenido del documento se organiza de la siguiente manera:

1. **Modelo de dominio**: se describe el modelo del dominio del sistema, derivado de los requisitos funcionales especificados. Este modelo representa las entidades principales y sus relaciones en el contexto del sistema.
2. **Paquete de análisis y de servicio**: se incluyen la descripción y los diagramas asociados a las clases de análisis y su organización en paquetes. Se divide en:
   - **Clases de análisis**: presenta las clases de entidad, control e interfaz que definen el comportamiento del sistema.
   - **Diagrama de paquetes**: ofrecen una vista general de los paquetes de análisis y su interrelación.
3. **Realización de casos de uso en el modelo de análisis**: se detallan los casos de uso implementados en el modelo de análisis. Se incluyen los diagramas de secuencia por paquetes de gestión, ilustrando el flujo de mensajes y la interacción entre los actores y el sistema. Esto cubre los siguientes paquetes:
   - Paquete de gestión de autenticación de usuarios
   - Paquete de análisis de gestión de usuarios
   - Paquete de análisis de gestión de cursos
   - Paquete de análisis de gestión de sesiones
   - Paquete de análisis de gestión de actividades
   - Paquete de análisis de gestión de resultados
4. **Propuesta de la arquitectura**: se ofrece una visión general de la arquitectura propuesta del sistema, incluyendo la estructura de las clases y su organización en el sistema

basado en el modelo de análisis. Se detalla la arquitectura del sistema para proporcionar una visión clara del diseño estructural.

Este documento proporciona una visión integral del análisis orientado a objetos del sistema, empleando técnicas y metodologías para una comprensión precisa de los requisitos y la estructura del sistema [1].

2 Modelo de dominio

El modelo de dominio proporciona una representación conceptual de las principales entidades, conceptos y sus relaciones dentro del sistema. Este modelo es fundamental para entender los requisitos y características del software, así como para identificar las necesidades de almacenamiento e información del sistema.

En la Figura 1 se describe el modelo de dominio del sistema.

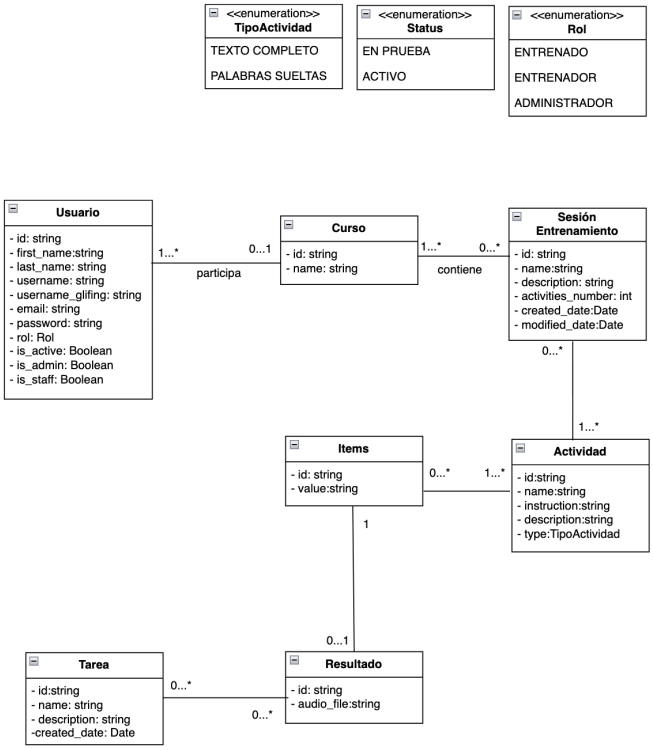


Figura 1: Modelo de dominio.

3 Paquete de análisis y de servicio

En esta sección, se presenta la descomposición del sistema en varios paquetes de análisis y servicio, en la Figura 2 se muestra el diagrama de paquetes correspondiente al análisis.

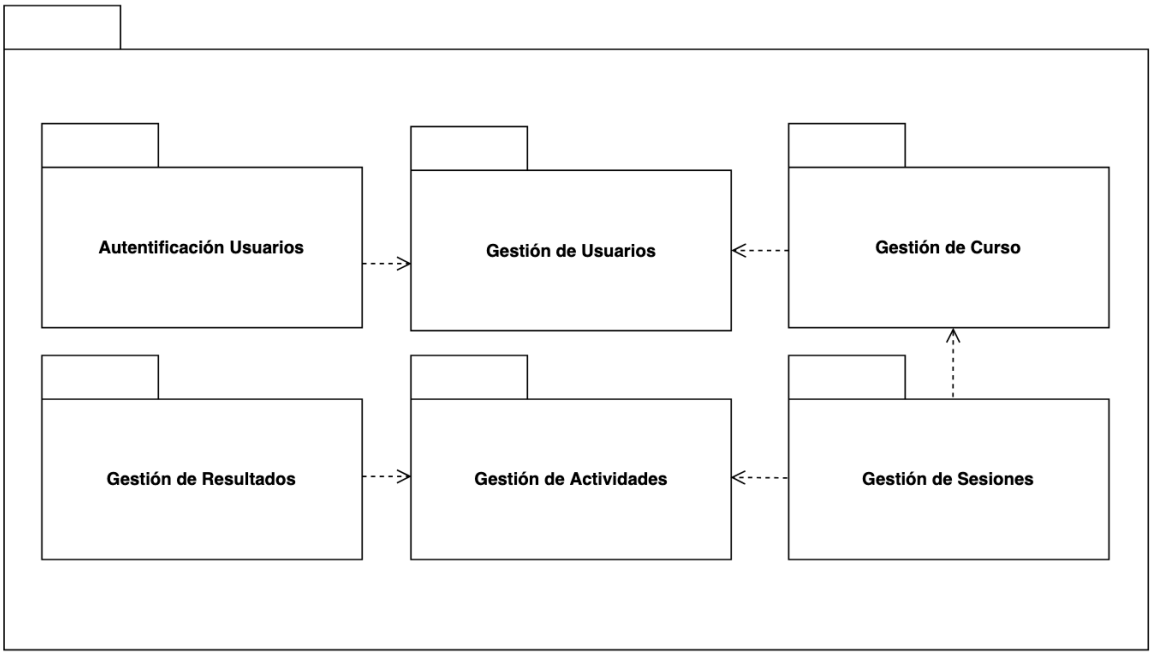


Figura 2: Diagrama de paquetes de análisis.

A continuación, se proporciona una breve descripción de cada uno de los paquetes de análisis:

- **Gestión de autenticación de usuarios**: este paquete se encarga de gestionar todo lo relacionado con la autenticación de los usuarios en el sistema.
- **Gestión de usuarios**: paquete responsable, entre otras funciones, de la administración y modificación de la información relacionada con los usuarios.

- **Gestión de actividades**: se ocupa de la gestión de actividades en el sistema, incluyendo la creación, visualización y eliminación de información relacionada con estas.
- **Gestión de sesiones**: maneja la administración de sesiones, controlando los aspectos relacionados con el inicio, mantenimiento y finalización de sesiones dentro del sistema.
- **Gestión de resultados**: se centra en la gestión de los resultados de las actividades de cada entrenamiento.

3.1 Clases de análisis

Para definir de manera más precisa el comportamiento de los paquetes de análisis, se utilizan tres tipos de clases de análisis:

- **Clases de entidad**: representan objetos de negocio generalmente persistentes que se almacenan en el sistema. Estas clases suelen corresponderse con las clases conceptuales definidas previamente.
- **Clases de control**: se encargan del procesamiento de la información.
- **Clases de interfaz**: son responsables de presentar y comunicar la información al usuario o a otros sistemas.

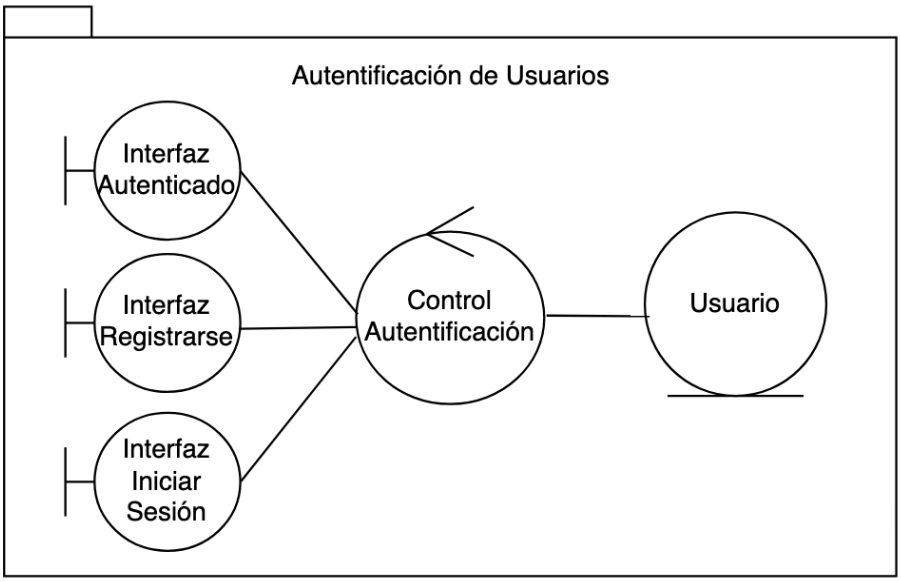


Figura 3: Paquete de análisis de gestión de autenticación de usuarios

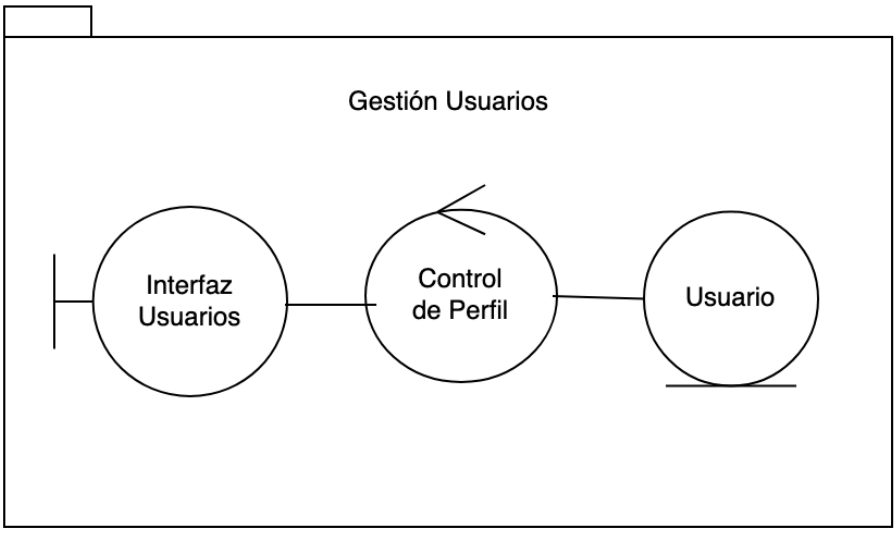


Figura 4: Paquete de análisis de gestión de usuarios.

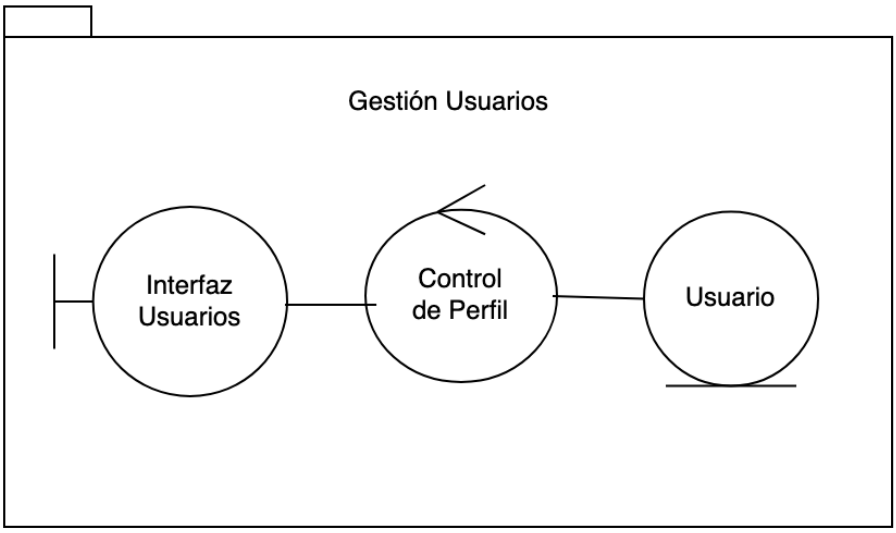


Figura 5: Paquete de análisis de gestión de sesiones.

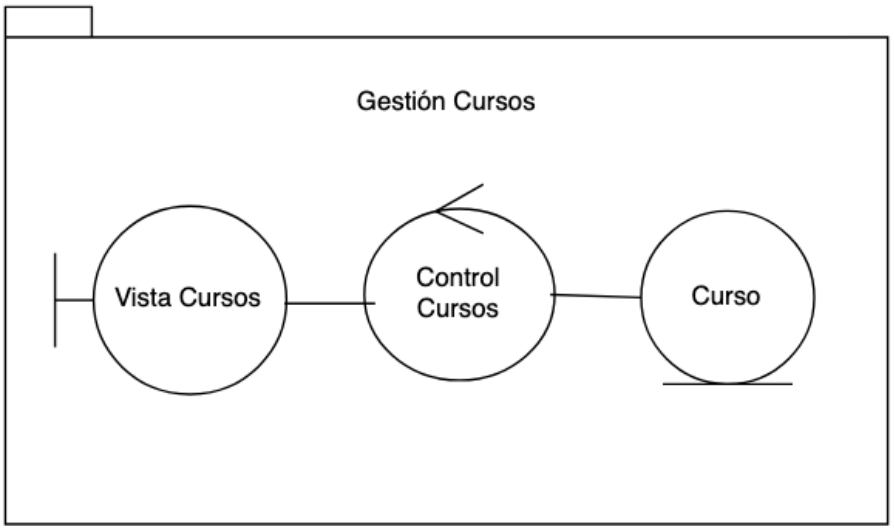


Figura 6: Paquete de análisis de gestión de cursos.

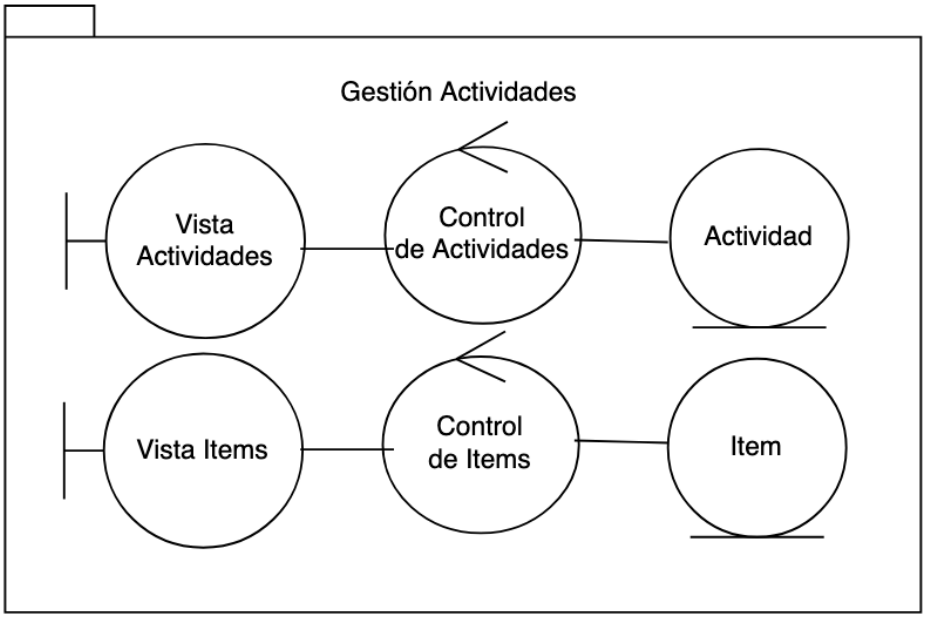


Figura 7: Paquete de análisis de gestión de actividades.

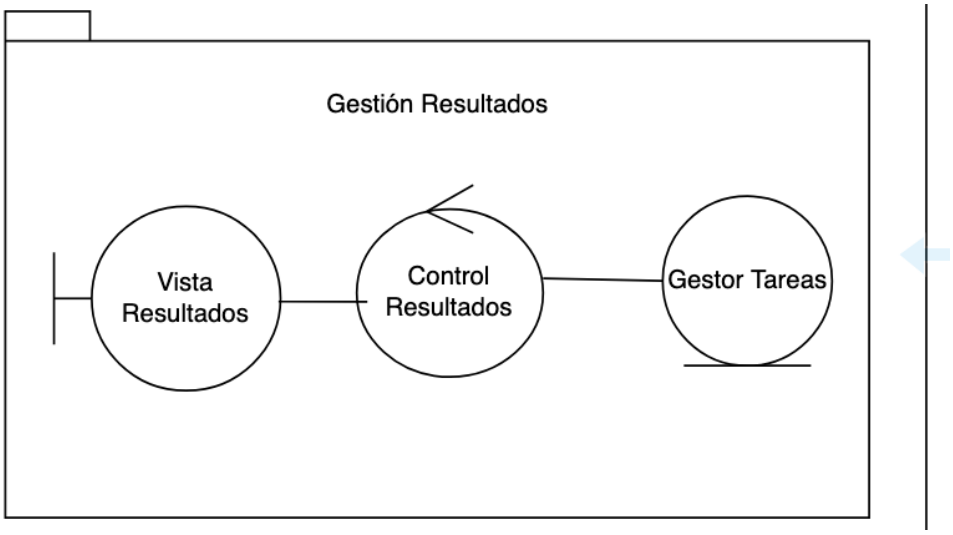


Figura 8: Paquete de análisis de gestión de resultados.

4 Realización de casos de uso en el modelo de análisis

En esta sección se presentan las vistas de interacción de los casos de uso mediante diagramas de secuencia.

4.1 Paquete de gestión de autenticación de usuarios

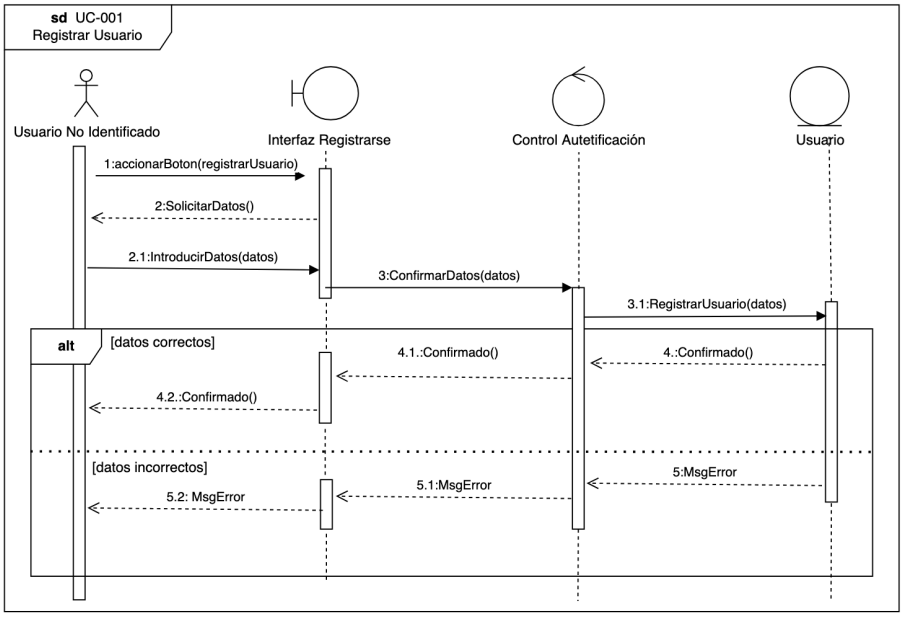


Figura 9: Diagrama de secuencia de registrar usuario

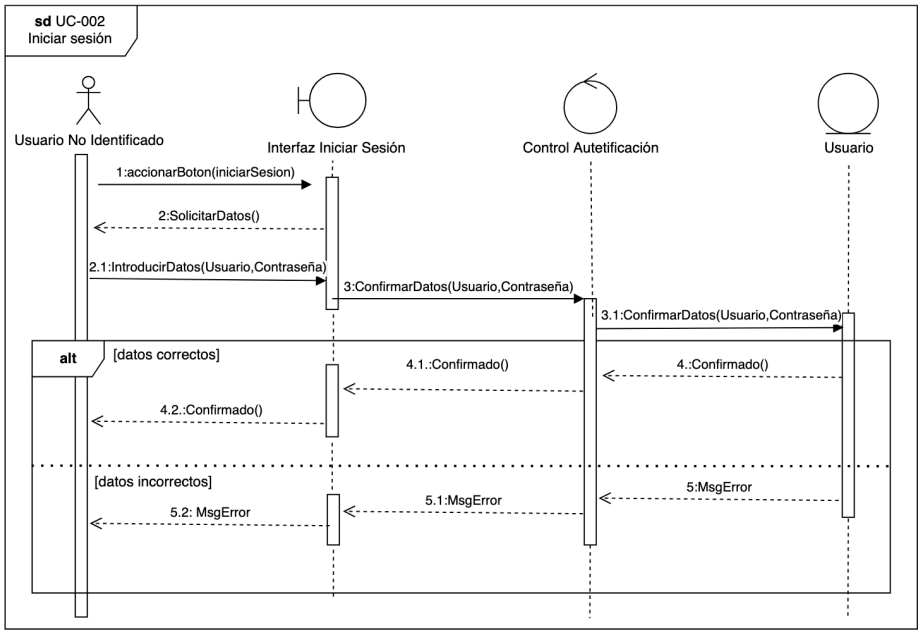


Figura 10: Diagrama de secuencia de iniciar sesión.

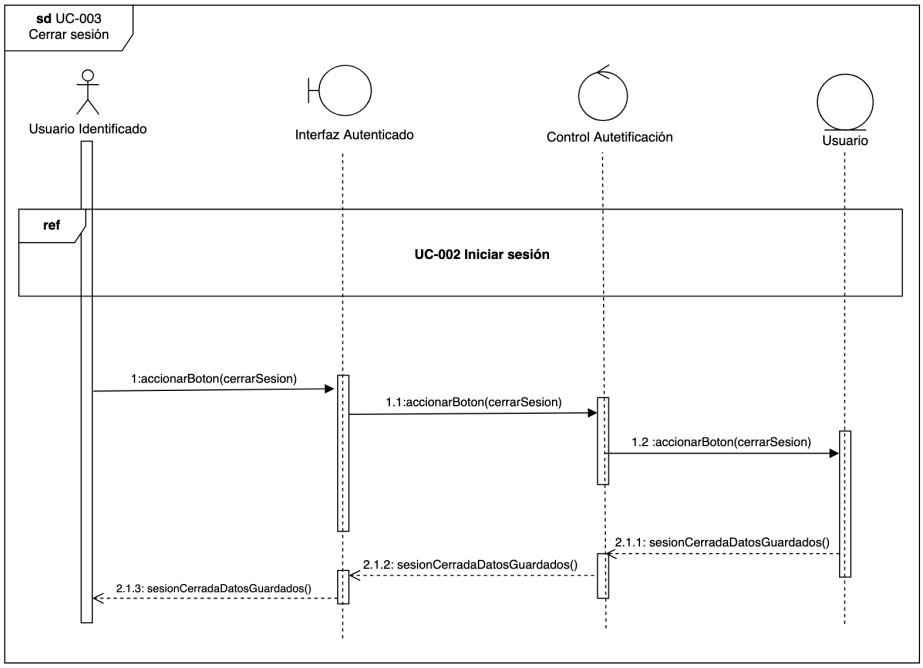


Figura 11: Diagrama de secuencia de cerrar sesión.

4.2 Paquete de análisis de gestión de usuarios

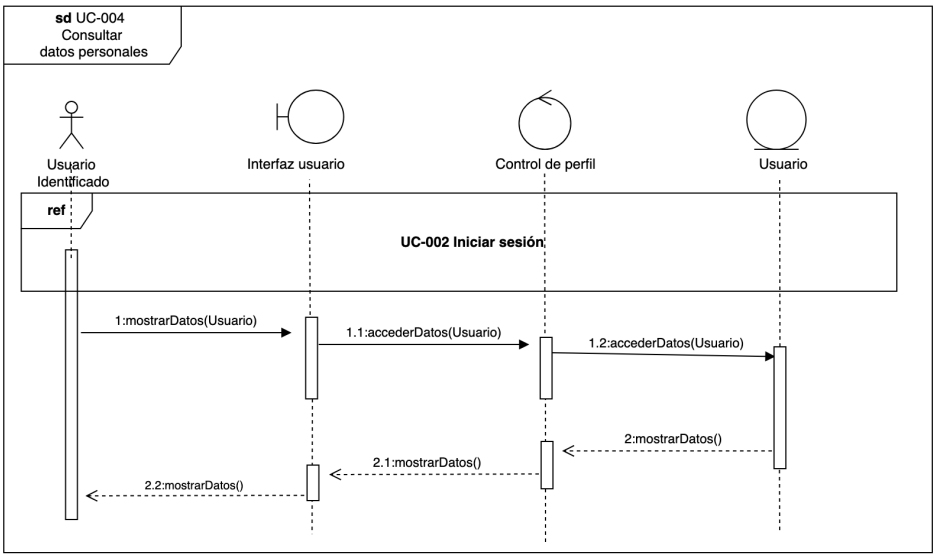


Figura 12: Diagrama de secuencia de consultar datos personales.

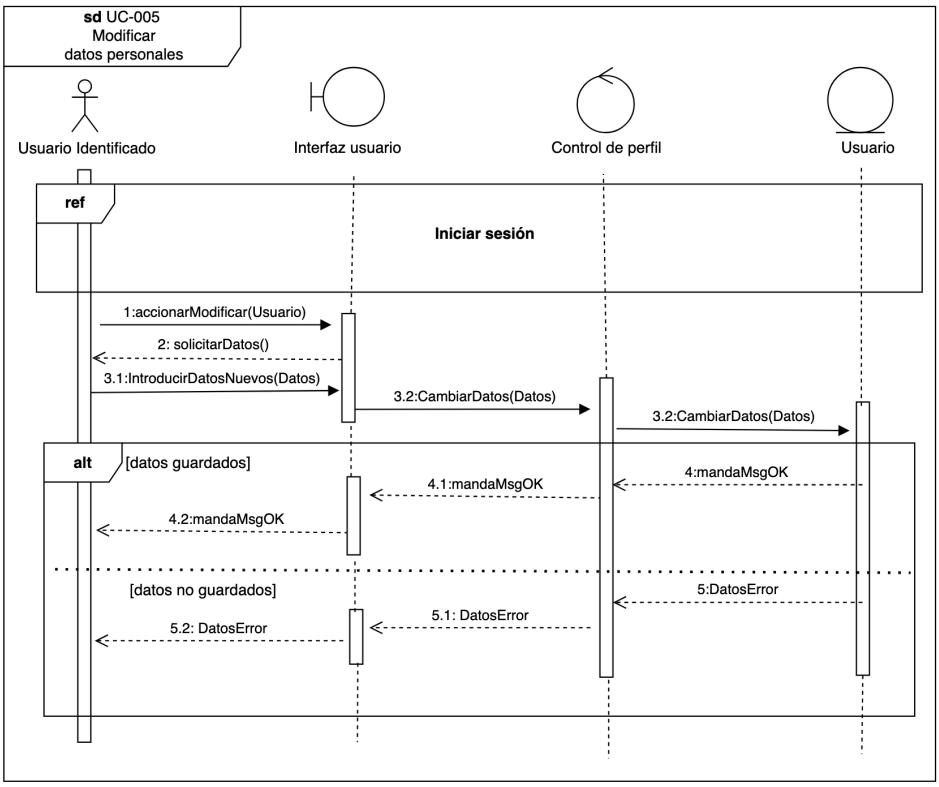


Figura 13: Diagrama de secuencia de modificar datos personales.

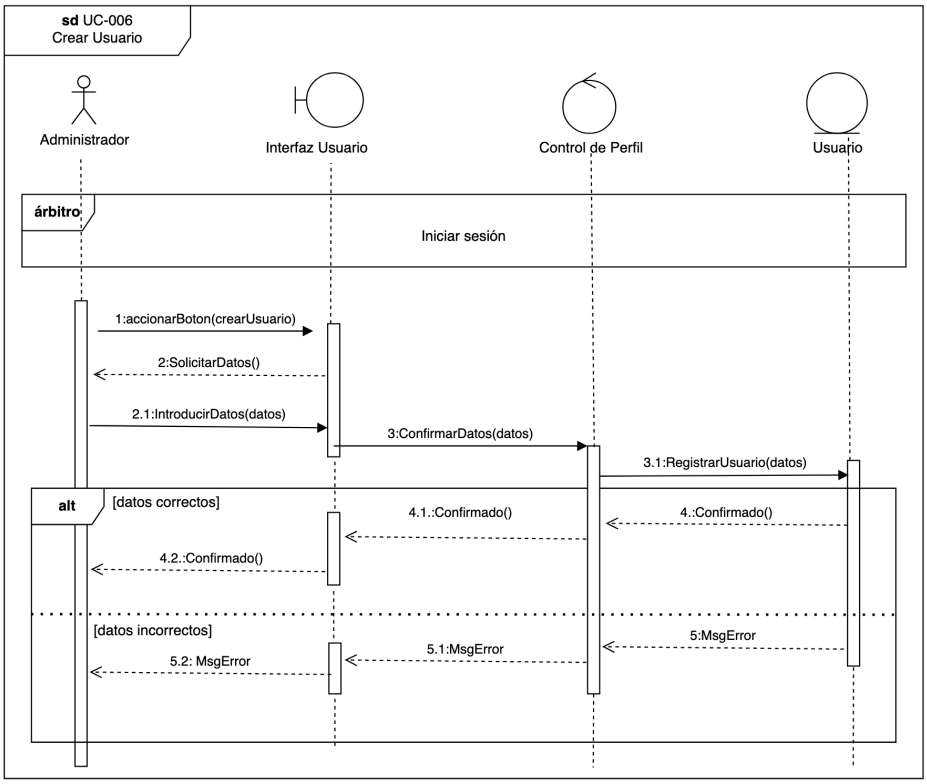


Figura 14: Diagrama de secuencia de crear usuarios.

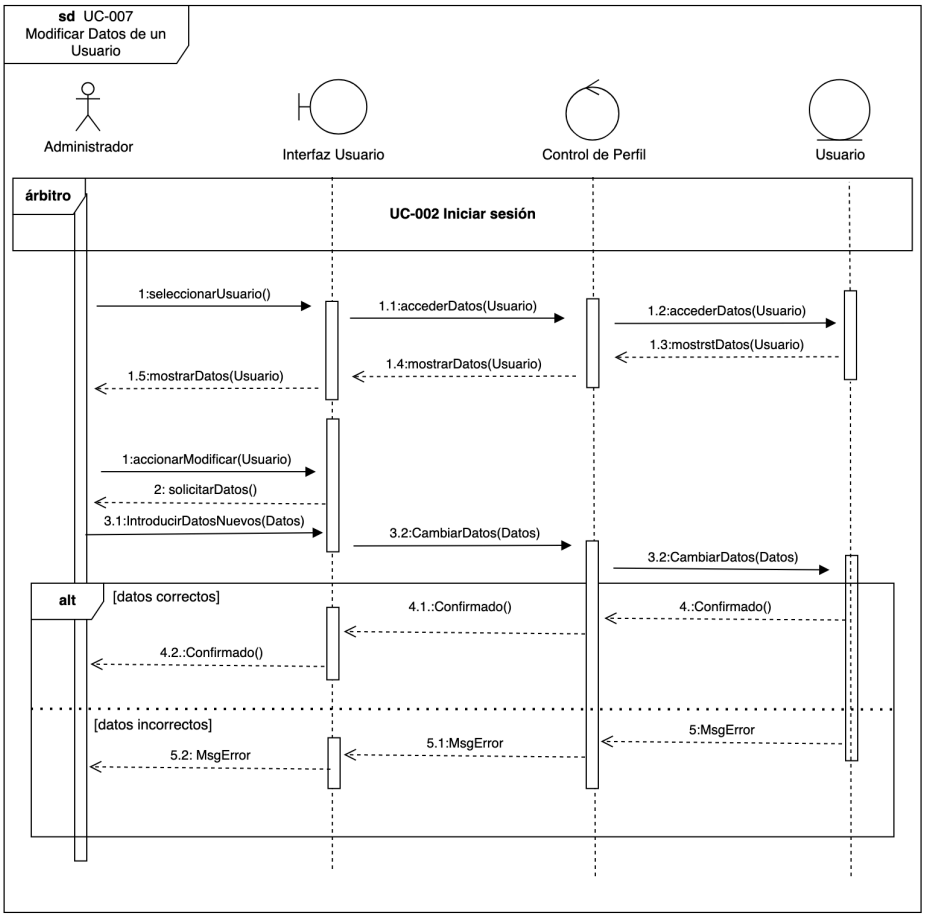


Figura 15: Diagrama de secuencia de modificar datos de un usuario.

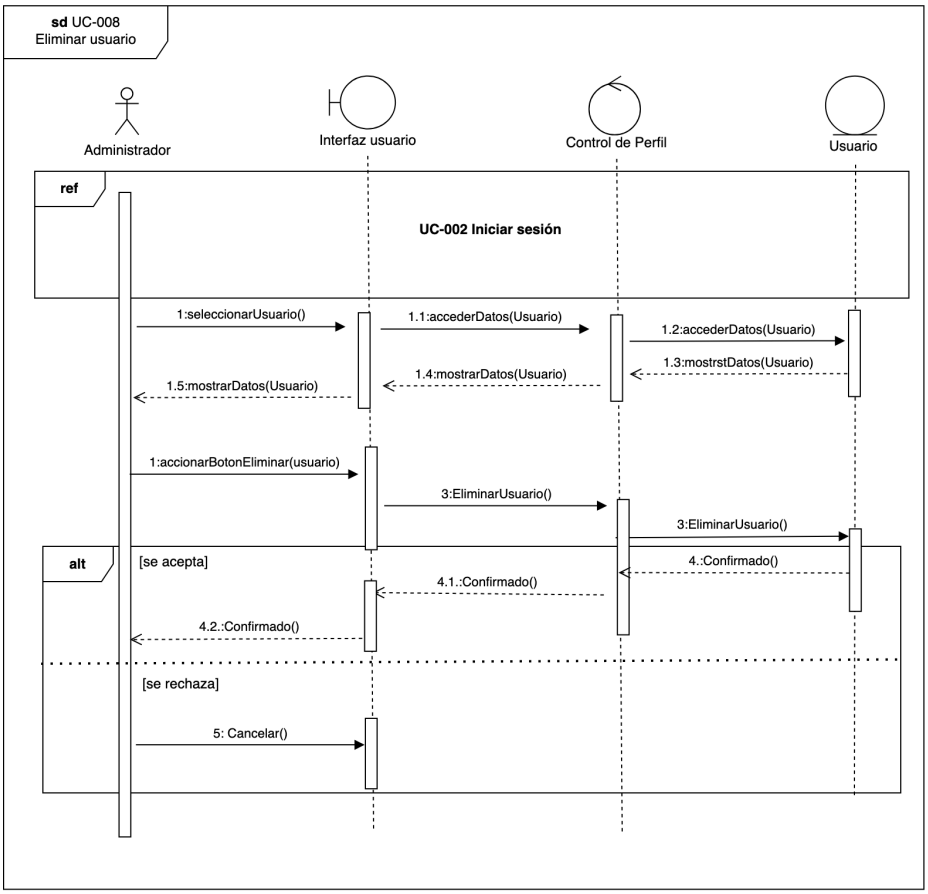


Figura 16: Diagrama de secuencia de eliminar usuario

4.3 Paquete de análisis de gestión de cursos

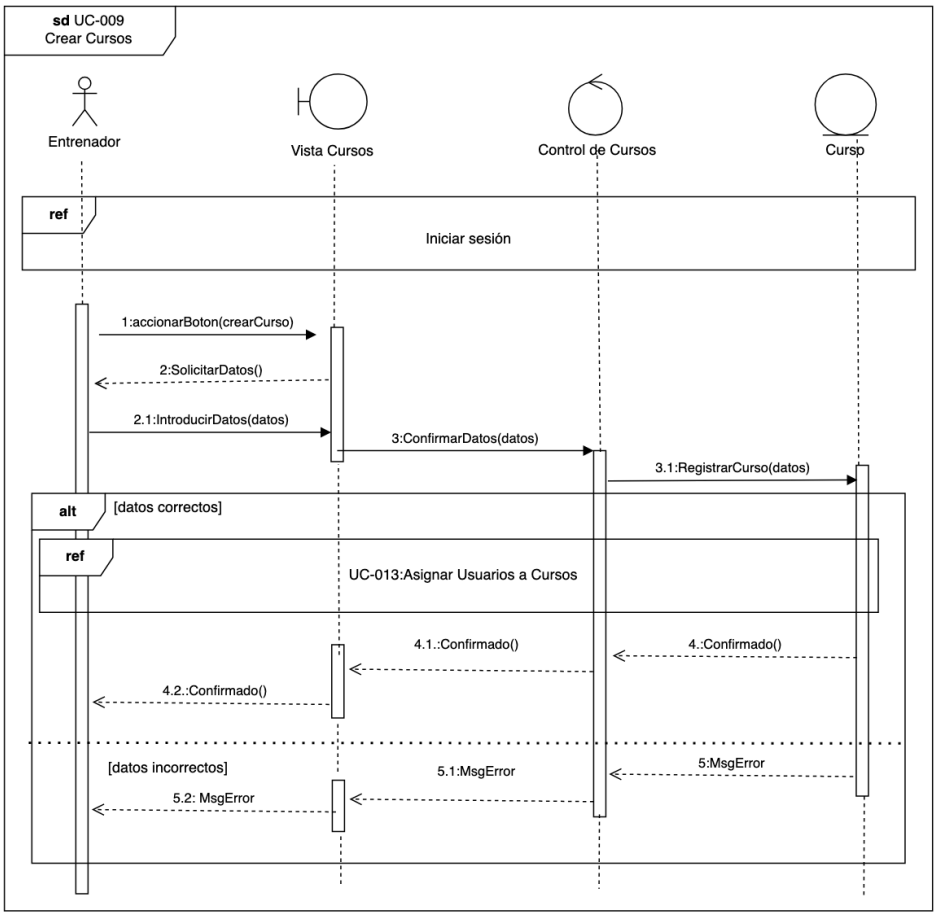


Figura 17: Diagrama de secuencia de crear cursos.

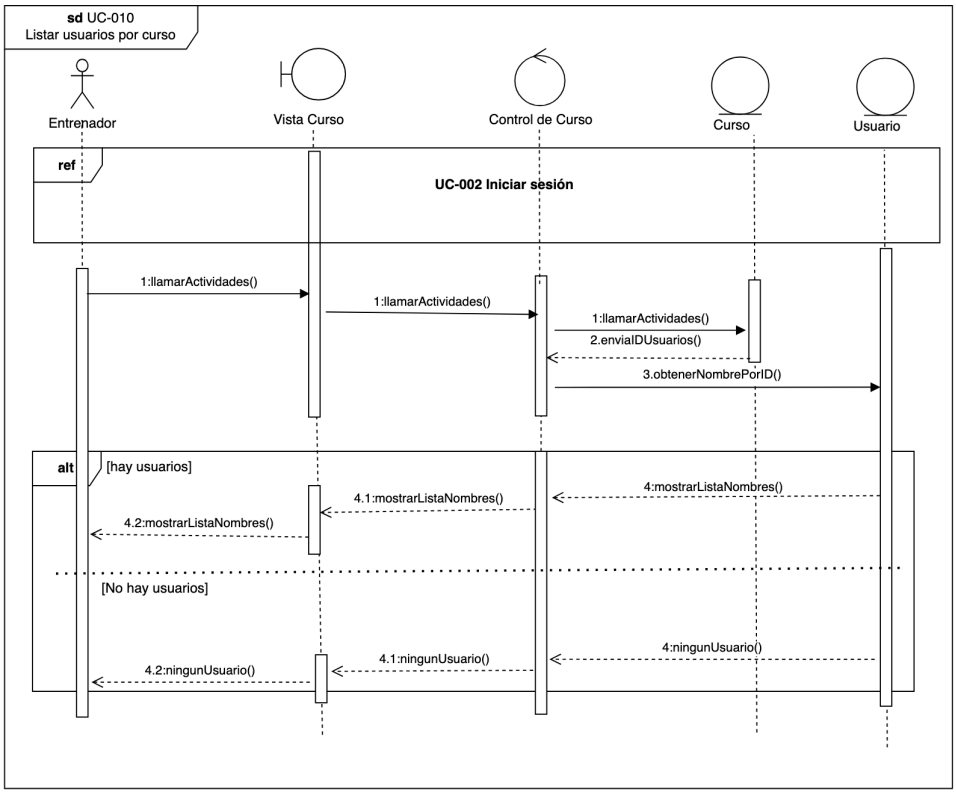


Figura 18: Diagrama de secuencia de listar usuarios por curso.

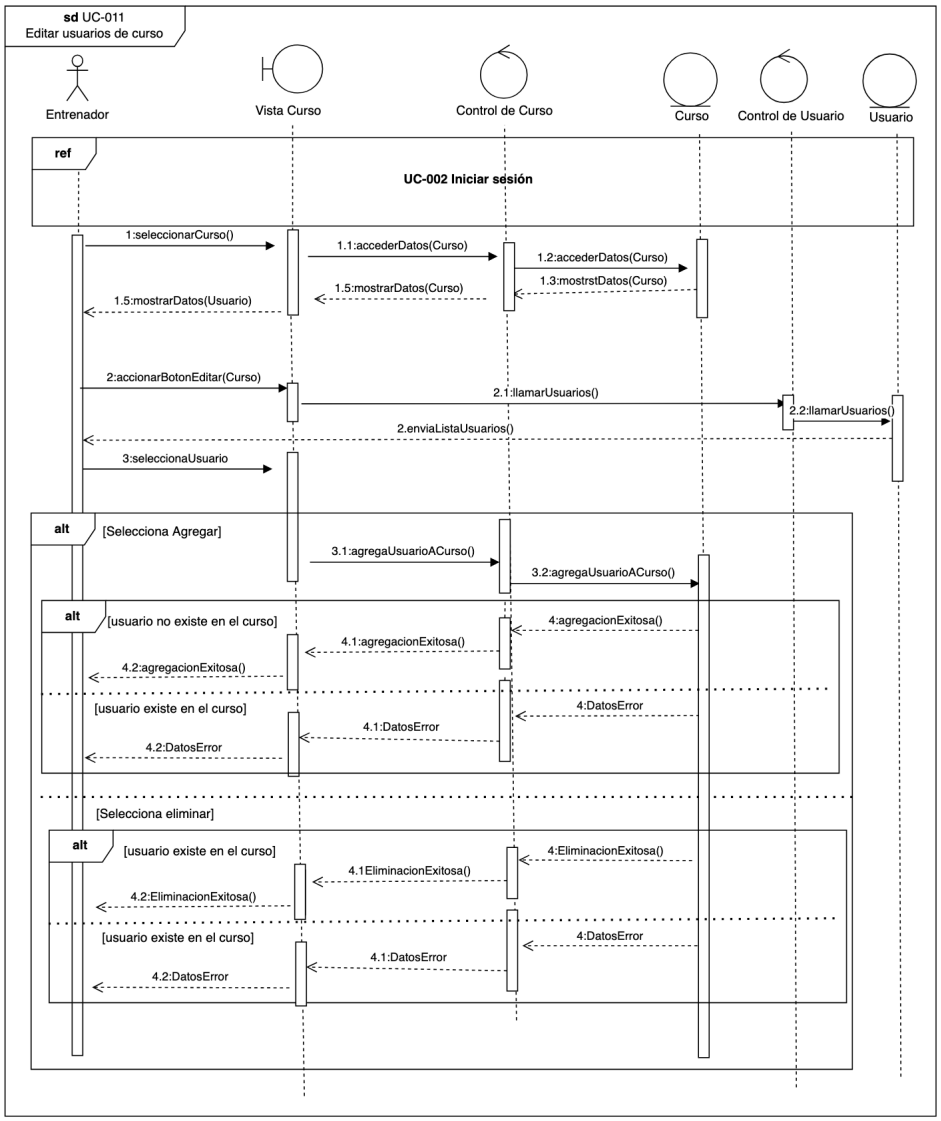


Figura 19: Diagrama de Secuencia de Editar Usuarios por Curso

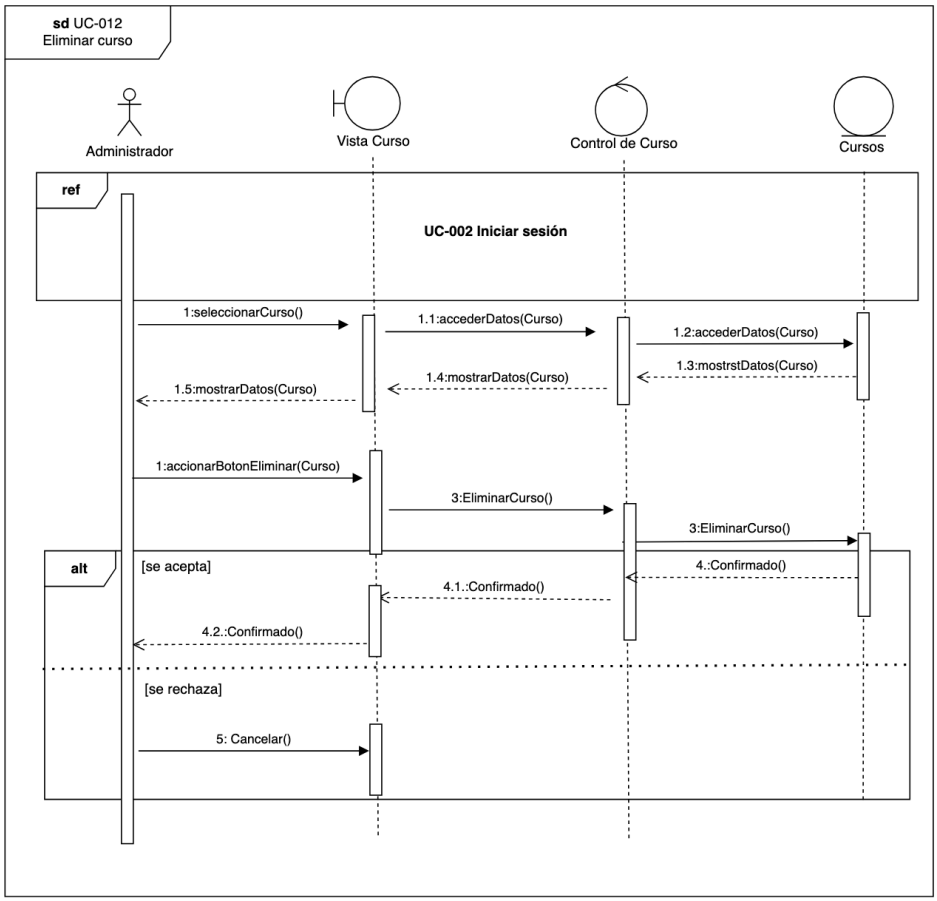


Figura 20: Diagrama de secuencia de eliminar curso.

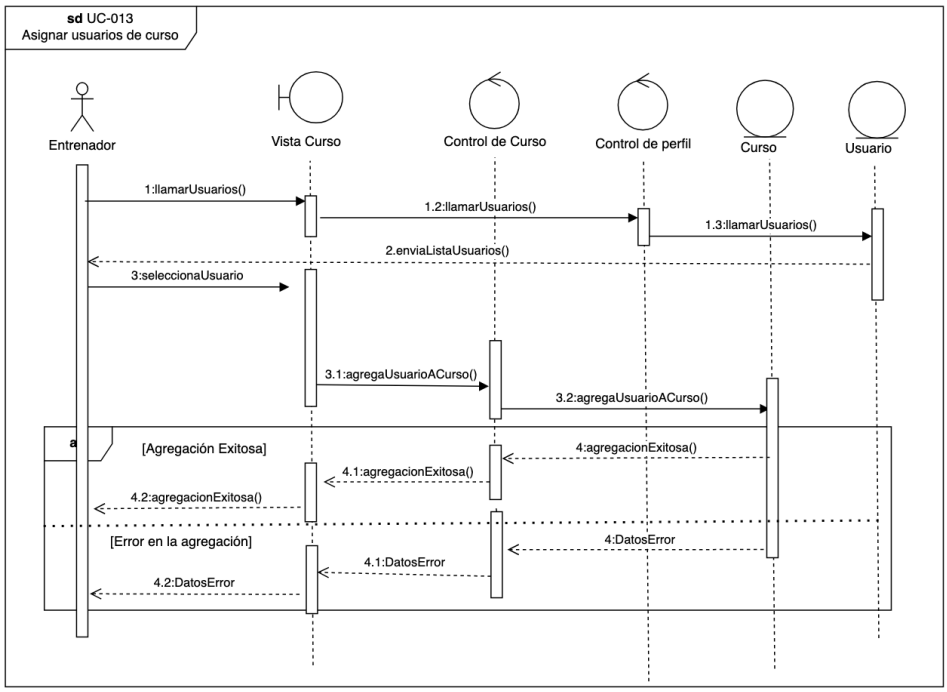


Figura 21: Diagrama de secuencia de asignar usuarios a curso.

4.4 Paquete de análisis de gestión de sesiones

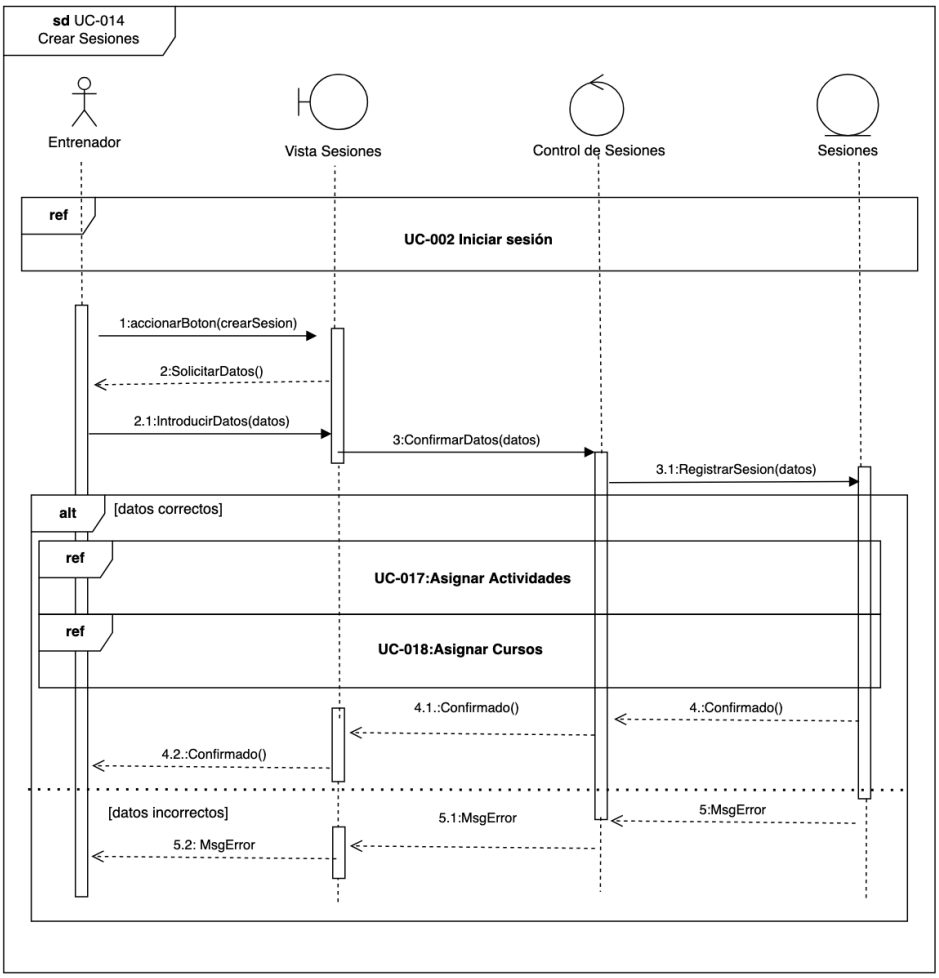


Figura 22: Diagrama de secuencia de crear sesión.

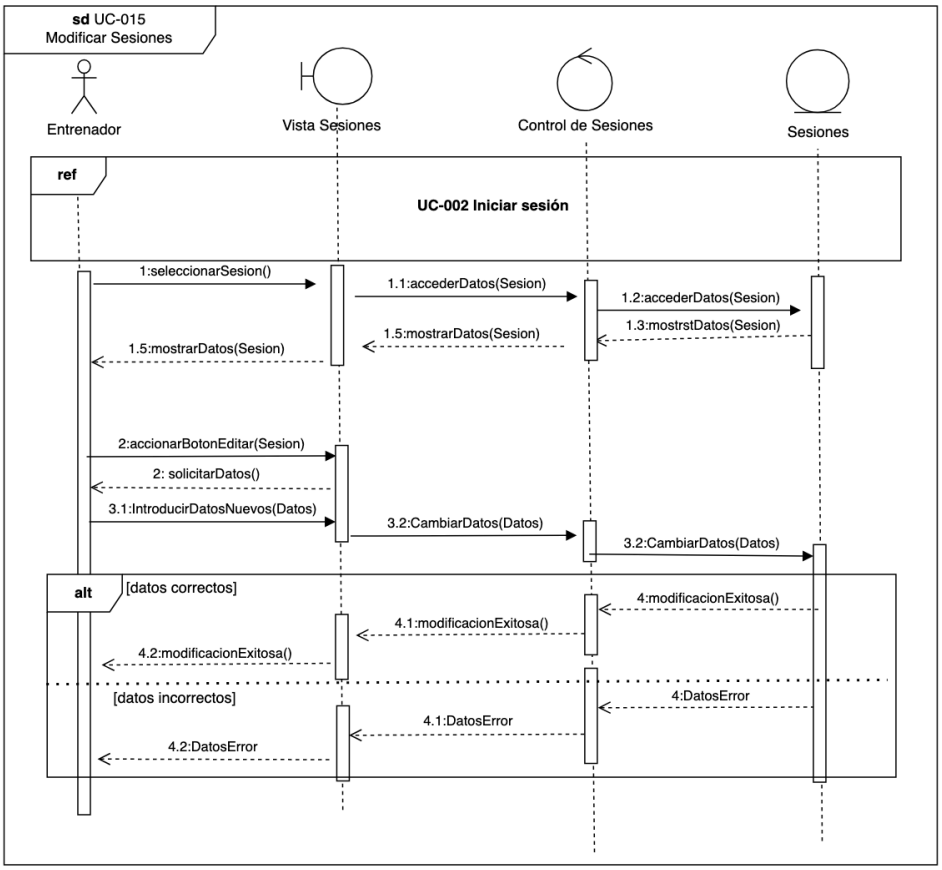


Figura 23: Diagrama de secuencia de modificar sesión.

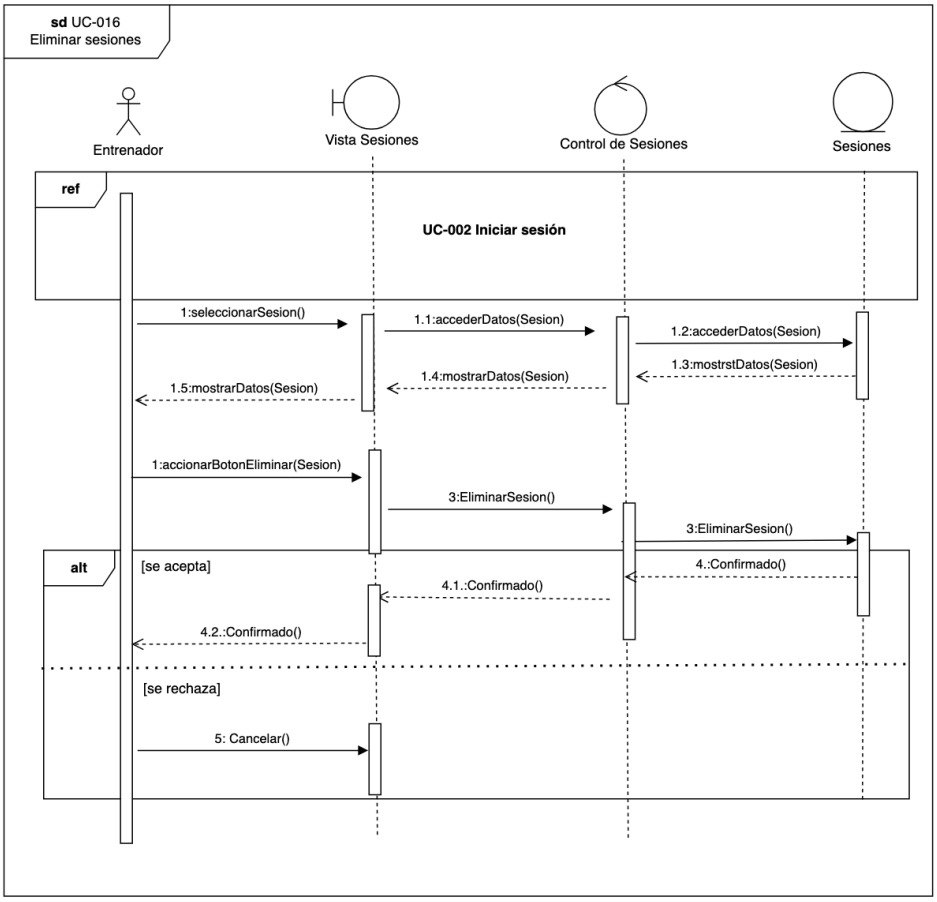


Figura 24: Diagrama de secuencia de eliminar sesión.

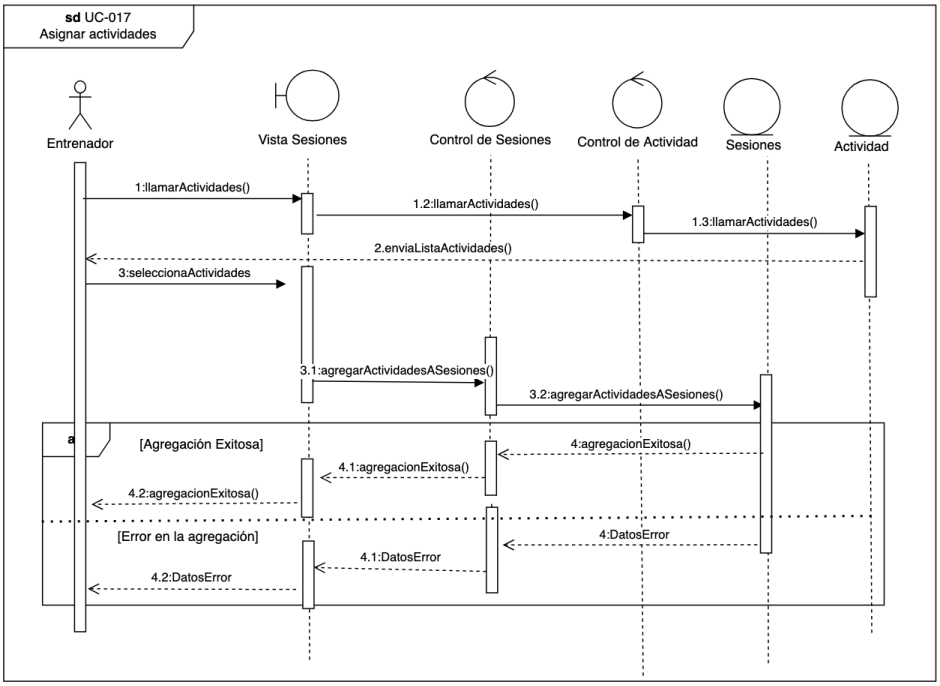


Figura 25: Diagrama de secuencia de asignar actividad a sesiones.

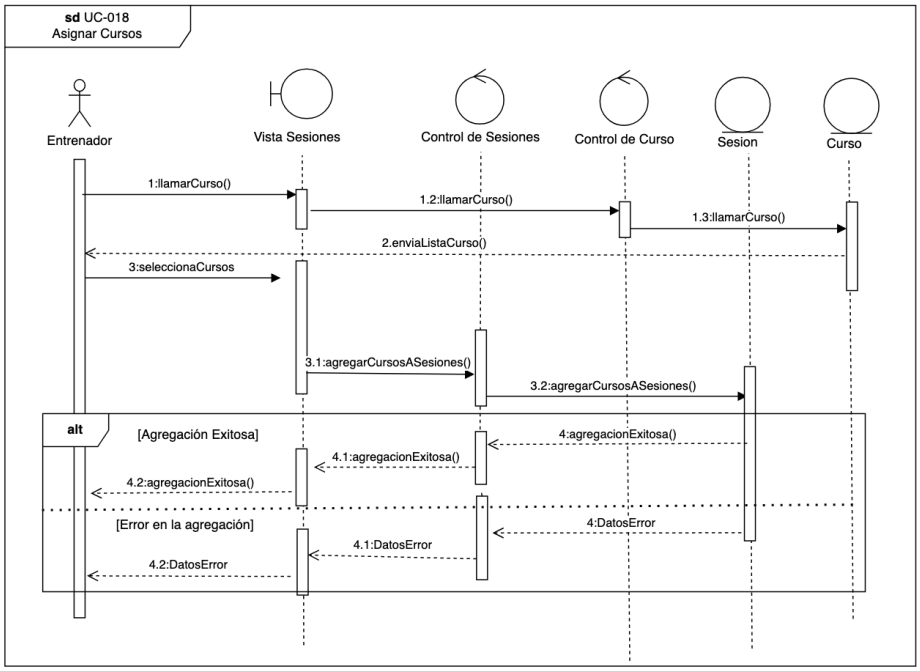


Figura 26: Diagrama de secuencia de asignar curso a sesiones.

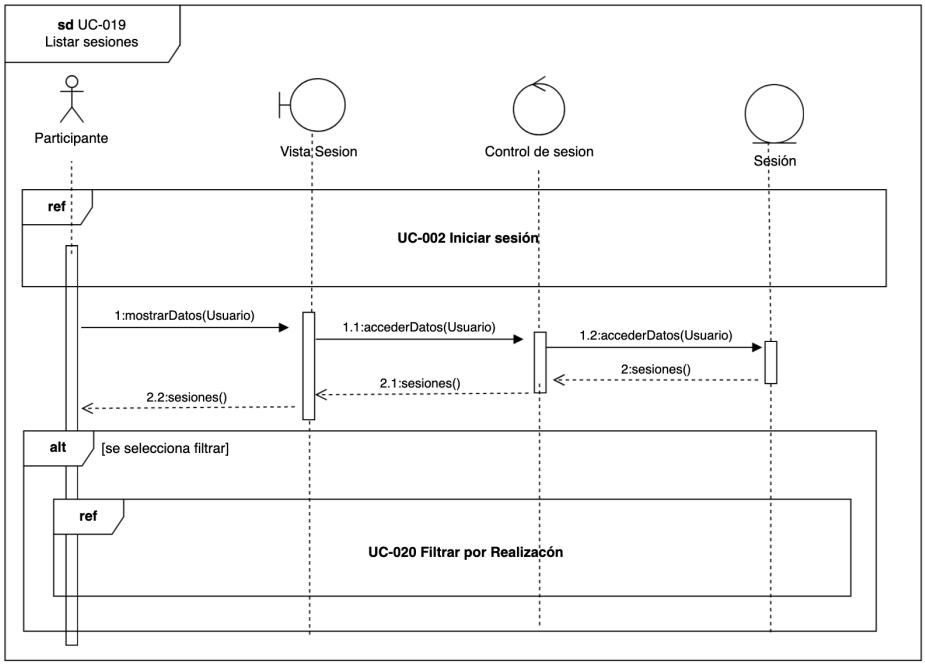


Figura 27: Diagrama de secuencia de listar sesiones.

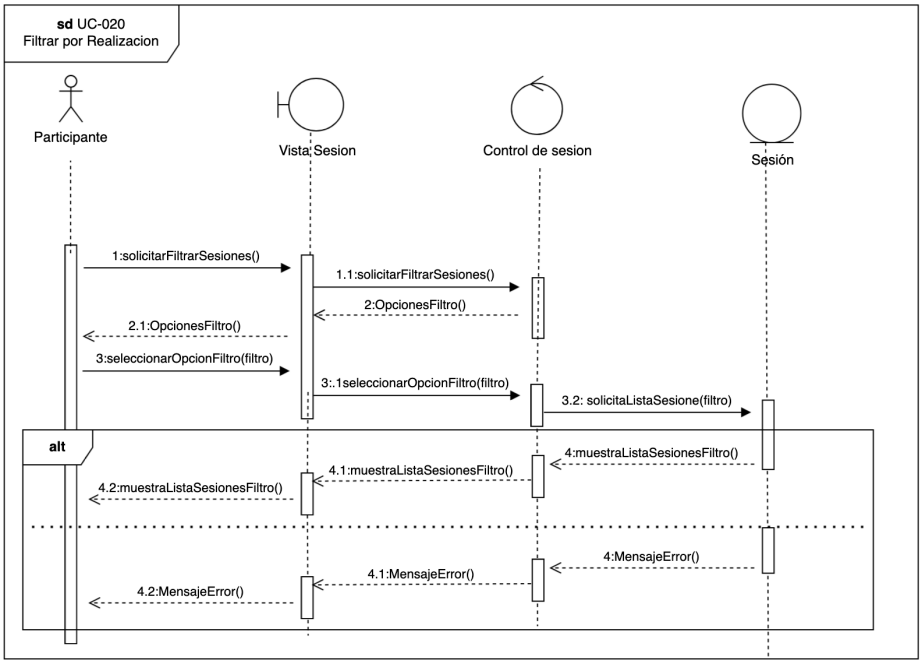


Figura 28: Diagrama de secuencia de filtrar por realización.

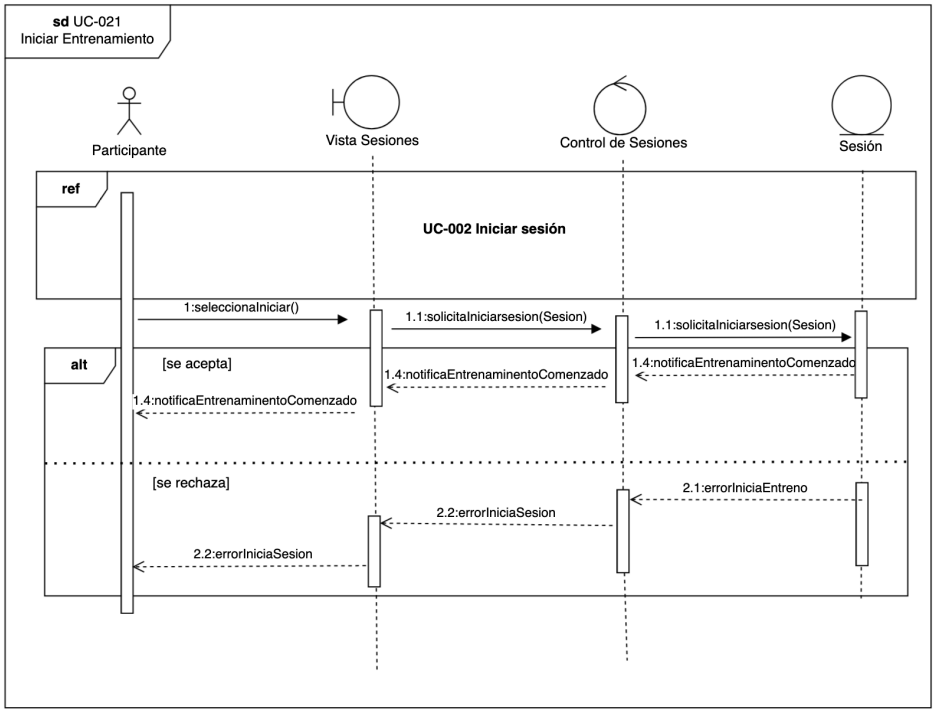


Figura 29: Diagrama de secuencia de iniciar entrenamiento.

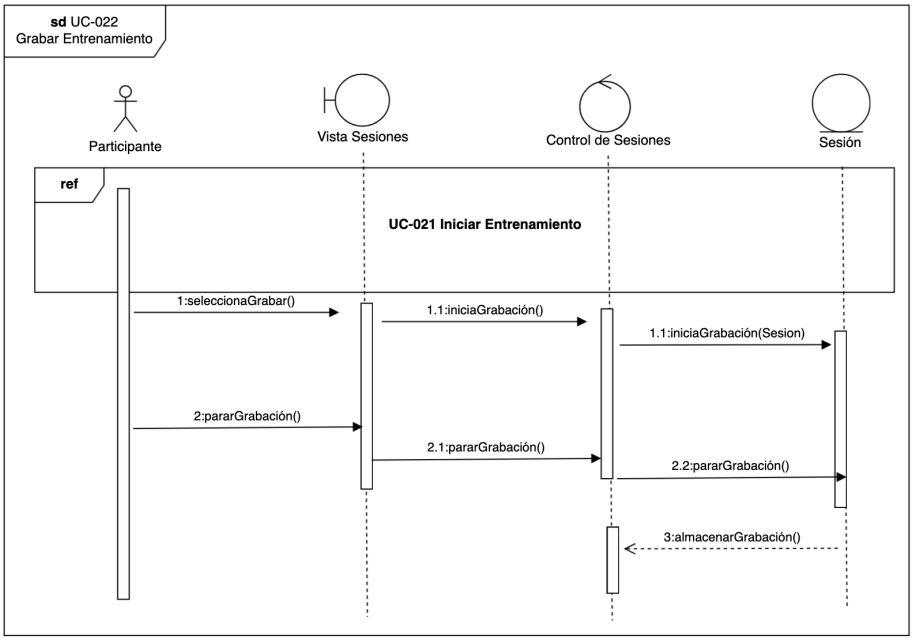


Figura 30: Diagrama de secuencia de grabar entrenamiento.

4.5 Paquete de análisis de gestión de actividades.

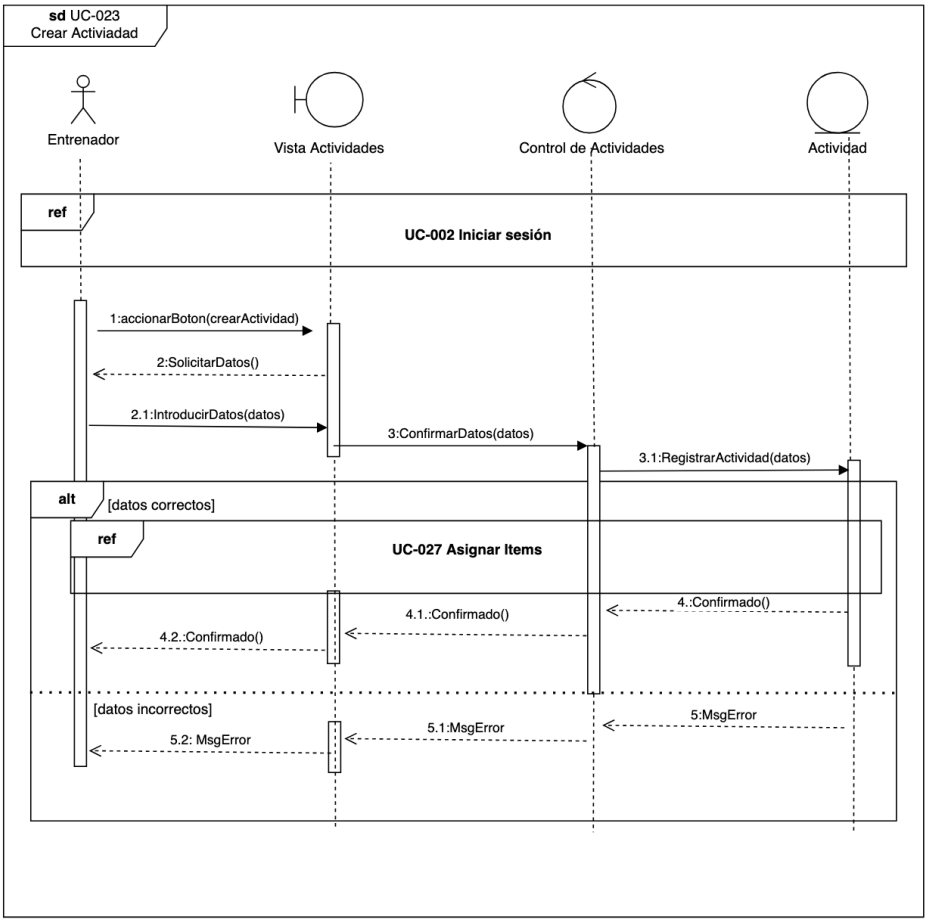


Figura 31: Diagrama de secuencia de crear actividad.

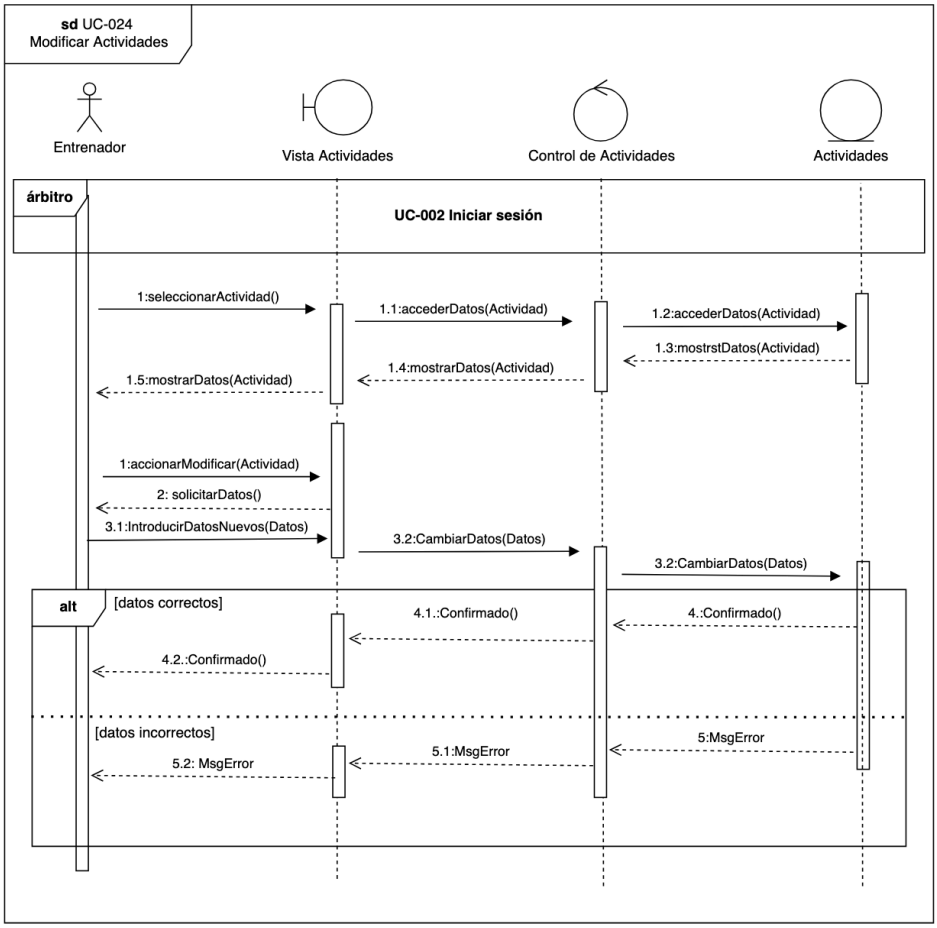


Figura 32: Diagrama de secuencia de modificar actividades.

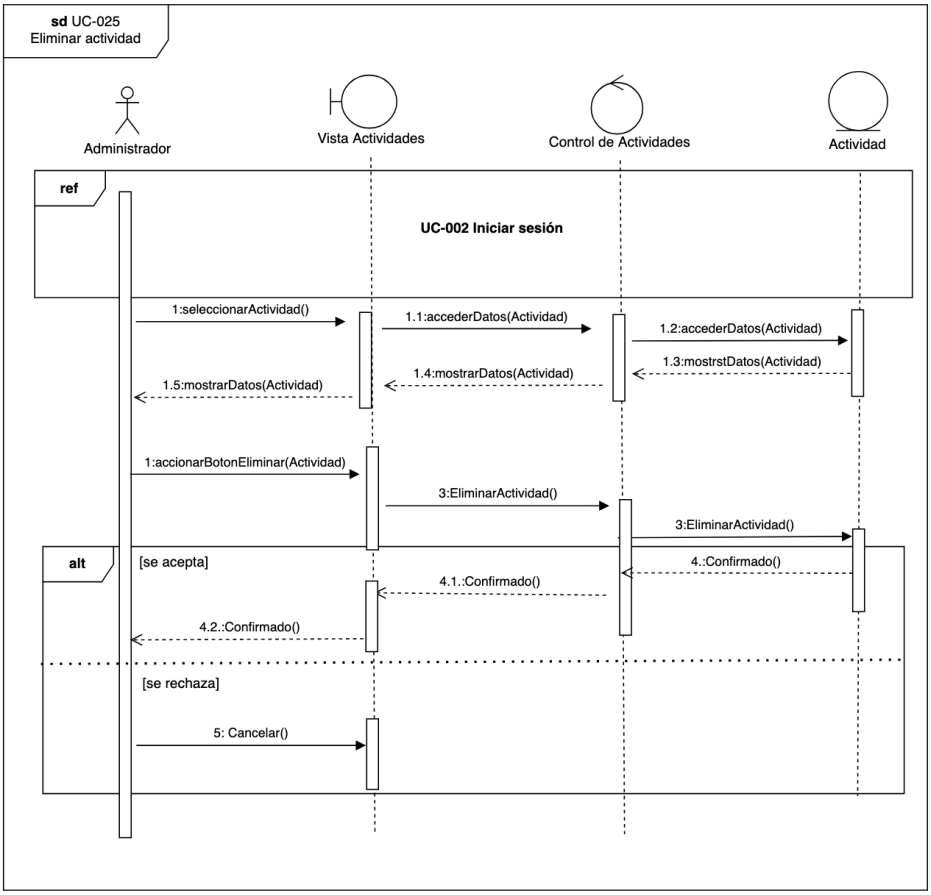


Figura 33: Diagrama de secuencia de eliminar actividades.

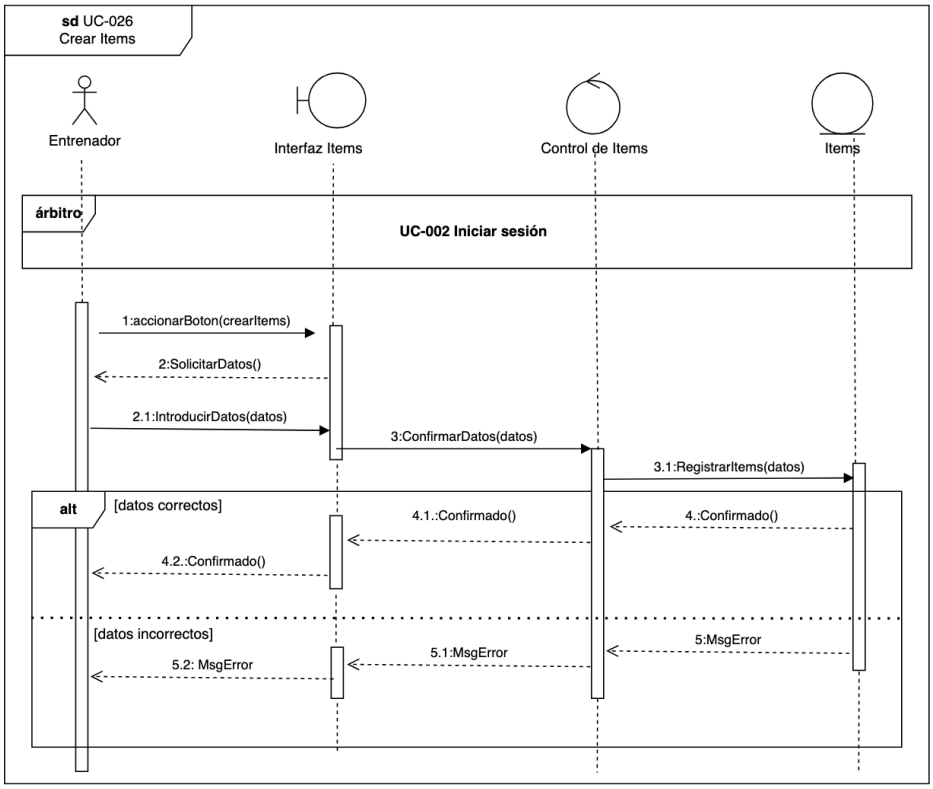


Figura 34: Diagrama de secuencia de crear items.

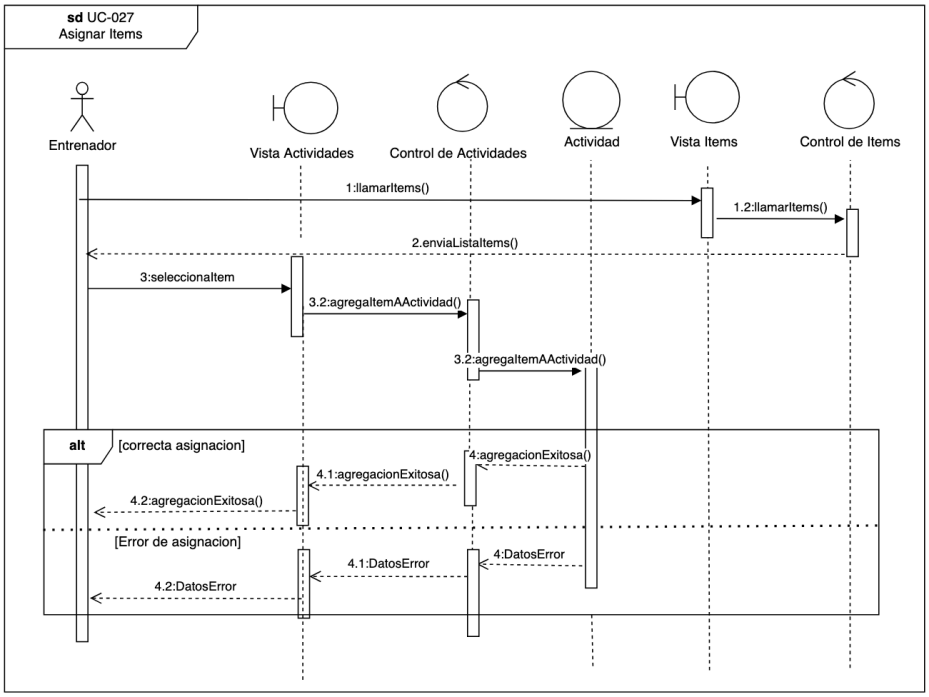


Figura 35: Diagrama de secuencia de asignar items.

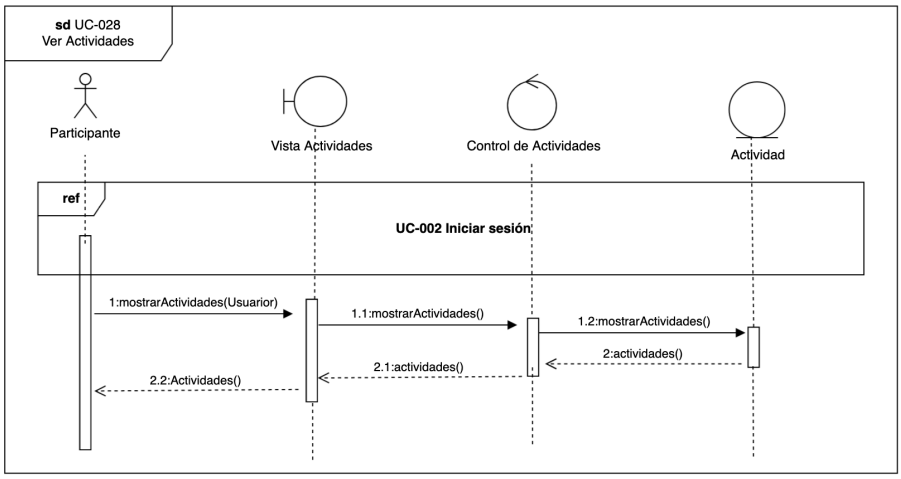


Figura 36: Diagrama de secuencia de ver actividades.

4.6 Paquete de análisis de gestión de resultados

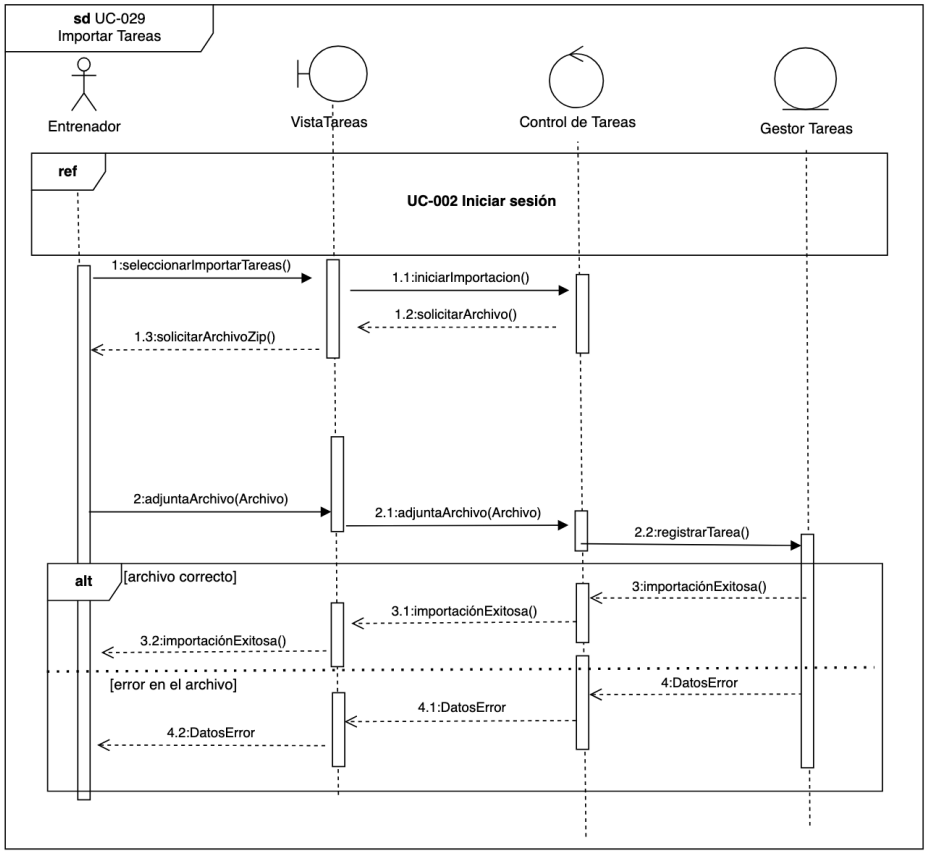


Figura 37: Diagrama de secuencia de importar tareas.

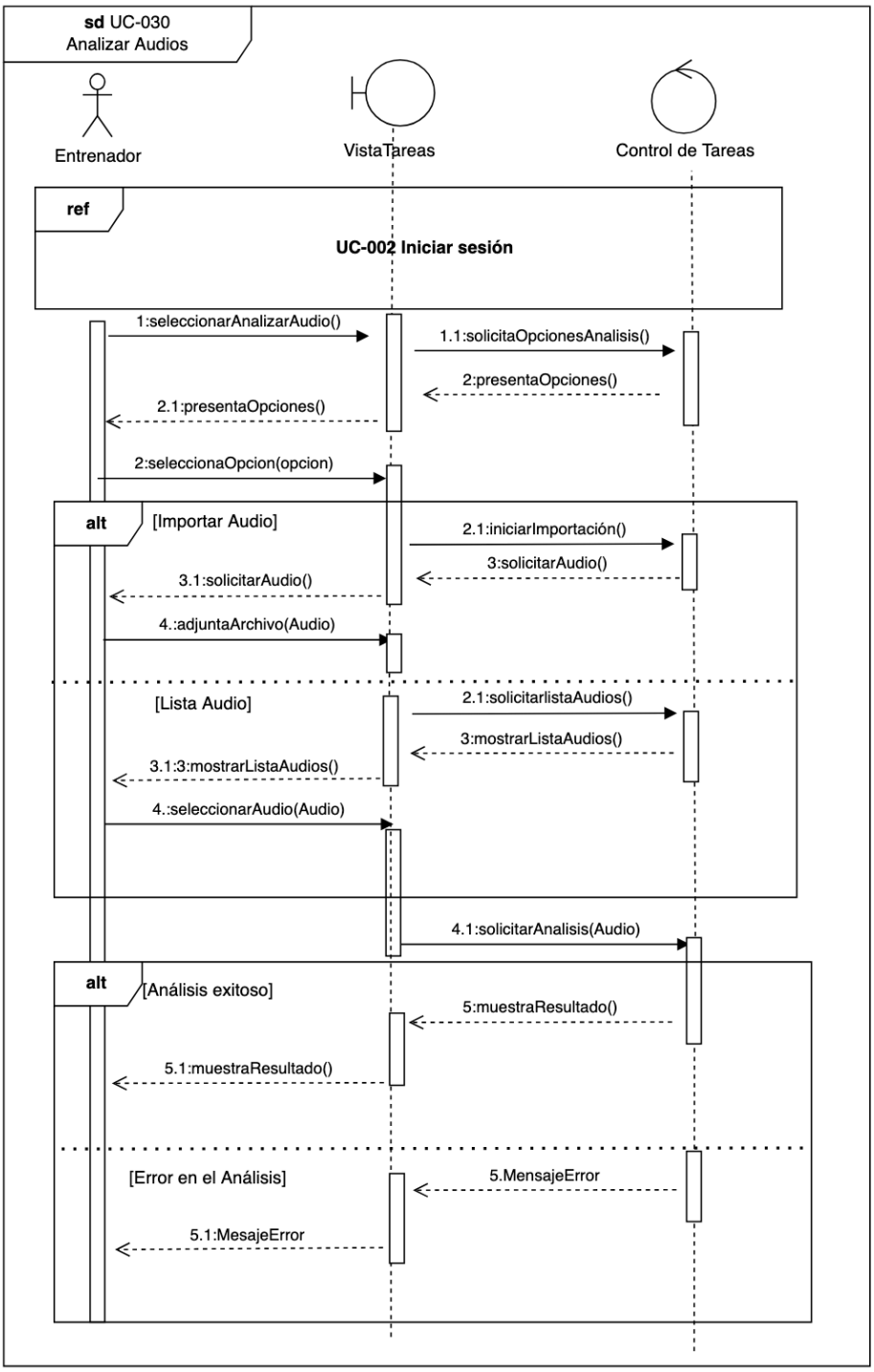


Figura 38: Diagrama de secuencia de analizar audio.

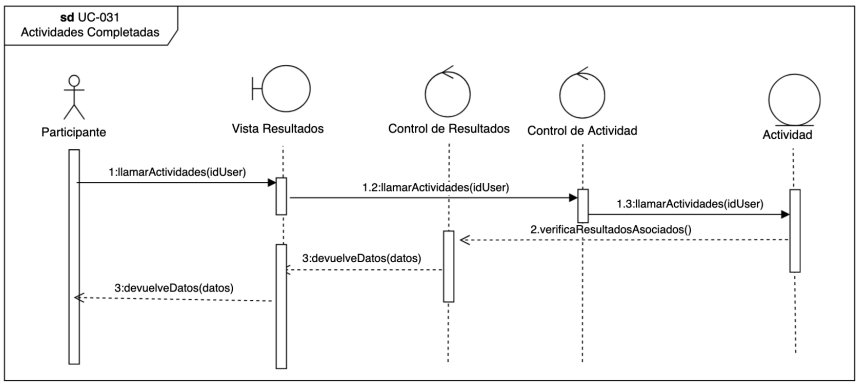


Figura 39: Diagrama de secuencia de actividades completadas.

5 Propuesta de la arquitectura

En este apartado se observa arquitectura definida en el sistema.

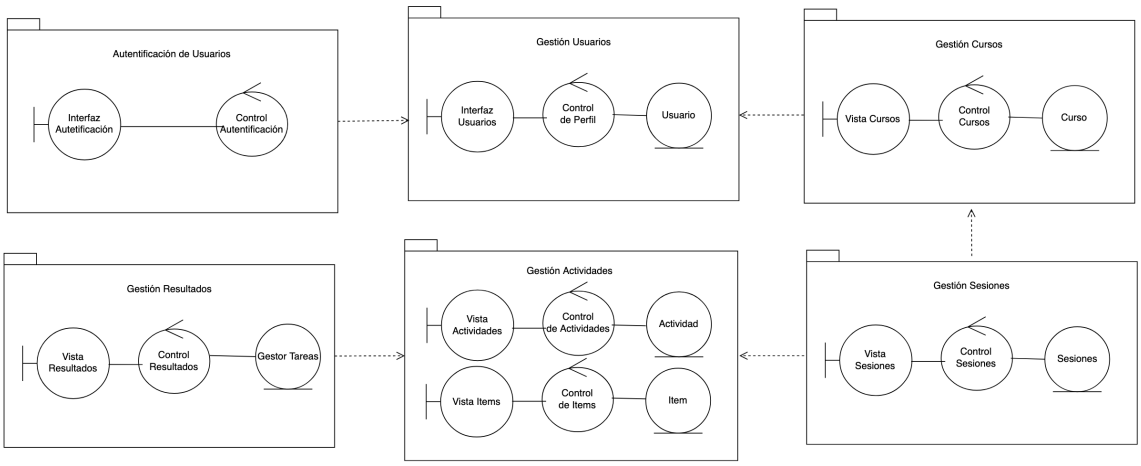


Figura 40: Descripción de la arquitectura del modelo de análisis.

Bibliografía

[1] García Peñalvo, F. J., & Moreno García, M. N. *Transparencias de la asignatura Ingeniería del Software I (Tema 7).*

Análisis de lectura de textos
Anexo IV. Diseño del sistema software

Trabajo de Fin de Grado
GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:
Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),
Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

Índice general

1. Introducción **1**

2. Modelo de diseño **3**

2.1. Patrón arquitectónico 3

3. Subsistemas de diseño **7**

4. Clases de diseño **9**

4.1. Frontend 9

4.2. Backend 16

5. Realización de casos de uso **19**

5.1. Gestión de autentificación de usuarios 19

5.2. Gestión de usuarios 21

5.3. Gestión de cursos 23

5.4. Gestión de sesiones 26

5.5. Gestión de actividades 30

5.6. Gestión de resultados 33

6. Base de datos **35**

7. Modelo de despliegue **37**

Bibliografía **37**

Índice de figuras

1. Modelo-Vista-ViewModel. 4
2. Modelo-Vista-Controlador. 5
3. Patrón cliente-servidor. 5
4. Subsistemas de diseño. 8
5. Clase de diseño actividad. 9
6. Clase de diseño Auth. 10
7. Clase de diseño Inicio. 10
8. Clase de diseño Items. 11
9. Clase de diseño Sesiones. 11
10. Clase de diseño Cursos. 12
11. Clase de diseño Task. 13
12. Clase de diseño User. 14
13. Clase de diseño Servicios. 15
14. Clase de diseño apiREST. 16
15. Diagrama de las clases de diseño de modelos. 17
16. Diagrama de secuencia: registrar usuario. 19

17. Diagrama de secuencia: iniciar sesión. 20
18. Diagrama de secuencia: cerrar sesión. 20
19. Diagrama de secuencia: consultar datos personales. 21
20. Diagrama de secuencia: modificar datos personales. 21
21. Diagrama de secuencia: crear usuario. 22
22. Diagrama de secuencia: modificar datos de un usuario. 22
23. Diagrama de secuencia: eliminar usuario. 23
24. Diagrama de secuencia: crear curso. 23
25. Diagrama de secuencia: listar usuarios por curso. 24
26. Diagrama de secuencia: editar usuarios de curso. 24
27. Diagrama de secuencia: eliminar curso. 25
28. Diagrama de secuencia: asignar usuarios a curso. 25
29. Diagrama de secuencia: crear sesión. 26
30. Diagrama de secuencia: modificar sesión. 26
31. Diagrama de secuencia: eliminar sesión. 27
32. Diagrama de secuencia: asignar actividades a sesiones. 27
33. Diagrama de secuencia: asignar cursos a sesiones. 28
34. Diagrama de secuencia: listar sesiones. 28
35. Diagrama de secuencia: filtrar por realización. 29
36. Diagrama de secuencia: iniciar entrenamiento. 29
37. Diagrama de secuencia: grabar entrenamiento. 30
38. Diagrama de secuencia: crear actividad. 30
39. Diagrama de secuencia: modificar actividad. 31
40. Diagrama de secuencia: eliminar actividad. 31

41. Diagrama de secuencia: crear items. 32

42. Diagrama de secuencia: asignar items a actividad. 32

43. Diagrama de secuencia: ver actividades. 33

44. Diagrama de secuencia: importar tareas. 33

45. Diagrama de secuencia: analizar audios. 34

46. Base de datos con DBeaver. 35

47. Modelo de despliegue. 37

1 Introducción

En este anexo, se presenta el diseño del sistema software desarrollado para una plataforma destinada al análisis de actividades de lectura. El objetivo principal de este diseño es proporcionar una estructura detallada y bien fundamentada que permita la implementación de un sistema robusto, escalable y flexible. Este diseño se fundamenta en un modelo orientado a objetos, que describe cómo se realizarán físicamente los casos de uso definidos en los requisitos del sistema.

El diseño del sistema se ha elaborado considerando tanto los requisitos funcionales como los no funcionales, con un enfoque en la eficiencia, la escalabilidad y la facilidad de mantenimiento del sistema. A lo largo de este anexo, se detallan los diferentes componentes del modelo de diseño, incluidos los patrones arquitectónicos y de diseño seleccionados, los subsistemas identificados, las clases de diseño principales y la realización de los casos de uso. Además, se presenta el diseño de la base de datos del sistema y el modelo de despliegue que ilustra la distribución física del software en diferentes nodos.

La estructura del anexo es la siguiente:

- **Patrones arquitectónicos y de diseño:** se presentan los patrones utilizados en el desarrollo del sistema, justificando su selección en función de las necesidades del proyecto.
- **Subsistemas de diseño:** se describe la descomposición del sistema en subsistemas, facilitando la gestión de su desarrollo e implementación.
- **Clases de diseño:** se exponen las clases de diseño que forman la base del sistema, explicando su papel y relación con los subsistemas.
- **Realización de casos de uso:** se detalla cómo se implementarán los casos de uso del sistema, utilizando diagramas de secuencia y descripciones detalladas.
- **Diseño de la base de datos:** se describe el esquema de la base de datos, incluyendo las tablas principales y las relaciones entre ellas.
- **Modelo de despliegue:** se proporciona una descripción del despliegue físico, mostrando cómo los diferentes componentes se distribuyen en los nodos de la red.

2 Modelo de diseño

El modelo de diseño de este sistema está enfocado en la representación modular y escalable que refleja cómo los casos de uso se realizan en la aplicación. Vamos a ver este diseño general en cada una de sus partes de forma detallada.

- **Frontend (Cliente):** implementado con Angular, se encarga de la presentación y la interacción con el usuario. Está estructurado en módulos y componentes que manejan distintas funcionalidades.
- **Backend (Servidor):** implementado con Django, maneja la lógica de negocio, el acceso a datos, y expone una API RESTful. Django se encarga de la validación, autenticación, y persistencia de datos en la base de datos.

2.1 Patrón arquitectónico

El sistema está diseñado con un enfoque modular y escalable que se refleja en una división entre el frontend y el backend, en donde cada uno utiliza patrones arquitectónicos específicos para optimizar el desarrollo y la gestión del sistema.

El **frontend**, implementado con Angular, sigue el patrón Modelo-Vista-ViewModel (MVVM) en la Figura 1. Este patrón [1] facilita una clara separación entre la lógica de presentación y la lógica de negocio. En este contexto:

- *Modelo*: Representa los datos y la lógica de negocio a través de clases TypeScript y servicios que interactúan con el backend.
- *Vista*: Compuesta por componentes y plantillas HTML que definen la interfaz de usuario y cómo se presenta la información.

- *ViewModel*: Representado por los componentes de Angular, actúa como un intermediario que maneja la lógica de presentación y actualiza la vista en respuesta a los cambios en el modelo.

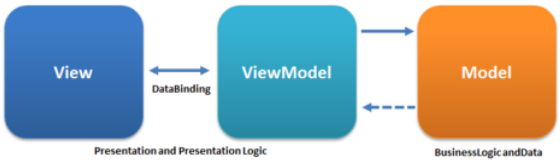


Figura 1: Modelo-Vista-ViewModel.

El uso de **data binding** en Angular sincroniza automáticamente los datos entre el modelo y la vista, facilitando la actualización en tiempo real sin código adicional. La técnica del **code behind** separa la lógica de la aplicación del diseño de la interfaz, mejorando la claridad y mantenibilidad del código. Juntas, estas técnicas no solo mejoran la claridad y legibilidad del código, sino que también facilitan la sincronización automática entre la vista y el modelo, promoviendo una separación de responsabilidades que hace el código más mantenible.

Por otro lado, el **backend** está desarrollado con Django y sigue el patrón Modelo-Vista-Controlador (MVC) en la Figura 2. En este patrón:

- *Modelo*: Define la estructura de los datos y las reglas de negocio a través de los modelos en Django, que se encargan de gestionar la interacción con la base de datos.
- *Vista*: Maneja las solicitudes HTTP y devuelve respuestas en formato JSON. En Django, las vistas procesan la lógica de negocio y coordinan la interacción con los modelos.
- *Controlador*: Se implementa mediante la configuración de URLs, que mapean las solicitudes a las vistas correspondientes.

Este enfoque [2] facilita una separación eficiente entre la lógica de datos, la lógica de negocio y el manejo de solicitudes, promoviendo un desarrollo más organizado y modular que permite una fácil escalabilidad del sistema.

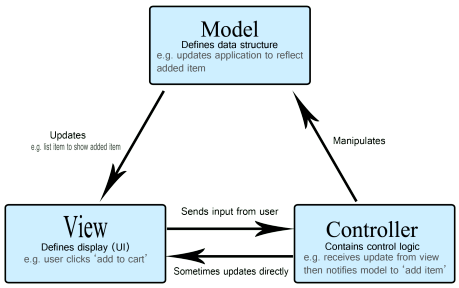


Figura 2: Modelo-Vista-Controlador.

El sistema diseñado sigue un patrón **cliente-servidor** que se ilustra en la Figura 3, y divide la arquitectura en dos partes principales: el cliente y el servidor. El cliente, en este caso, es la aplicación frontend implementada con Angular, que maneja la interfaz de usuario y las interacciones del usuario. El servidor, implementado con Django, procesa las solicitudes del cliente, maneja la lógica de negocio y gestiona el acceso a los datos. Esta separación permite que el cliente y el servidor evolucionen de manera independiente, siempre que se mantenga la compatibilidad con la API que los conecta.

El **backend** y el **frontend** se comunican a través de una API RESTful, que utiliza métodos HTTP estándar (GET, POST, PUT, DELETE) para intercambiar datos en formato JSON. Esta arquitectura de API RESTful proporciona una interfaz simple y coherente para la comunicación entre el cliente y el servidor, facilitando la integración y el mantenimiento del sistema. Además, mejora la eficiencia de la comunicación y asegura que las distintas partes del sistema puedan evolucionar y escalar de manera independiente.

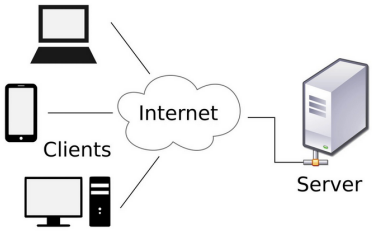


Figura 3: Patrón cliente-servidor.

En conjunto, este enfoque modular y la clara separación entre las distintas capas del sistema aseguran un diseño flexible y mantenible, capaz de adaptarse a futuros cambios y mejoras en la funcionalidad y en la tecnología utilizada.

3 Subsistemas de diseño

Para facilitar la comprensión del diseño, el sistema se ha dividido en subsistemas más específicos denominados subsistemas de diseño. La principal división se ha realizado entre el frontend y el backend [4], cada uno con sus responsabilidades y funciones particulares:

- **Frontend:** implementado con Angular, el frontend se encarga de la interfaz de usuario y de la interacción con el usuario final. Está estructurado en módulos y componentes que manejan diferentes funcionalidades.
  - **Componentes:** definen la estructura y el comportamiento de la interfaz de usuario. Cada componente está asociado a una vista específica y maneja la lógica de presentación de esa vista.
  - **Servicios:** encargados de la comunicación con el backend a través de la API RESTful. Gestionan la lógica para obtener y enviar datos entre el cliente y el servidor, facilitando la integración entre los componentes y la API.
- **Backend:** implementado con Django, el backend maneja la lógica de negocio, el acceso a datos y expone una API RESTful. Se encarga de la validación, autenticación y persistencia de los datos en la base de datos.
  - **API RESTful:** permite la interacción entre el frontend y el backend utilizando métodos HTTP estándar (GET, POST, PUT, DELETE). Las respuestas se envían en formato JSON, lo que facilita la comunicación entre los diferentes componentes del sistema.
  - **Modelos Django:** definen la estructura de los datos y gestionan la interacción con la base de datos. Cada modelo representa una tabla en la base de datos y define los campos y las relaciones entre ellos.

Teniento en cuenta todo esto, la Figura 4 representa los subsistemas de diseño.

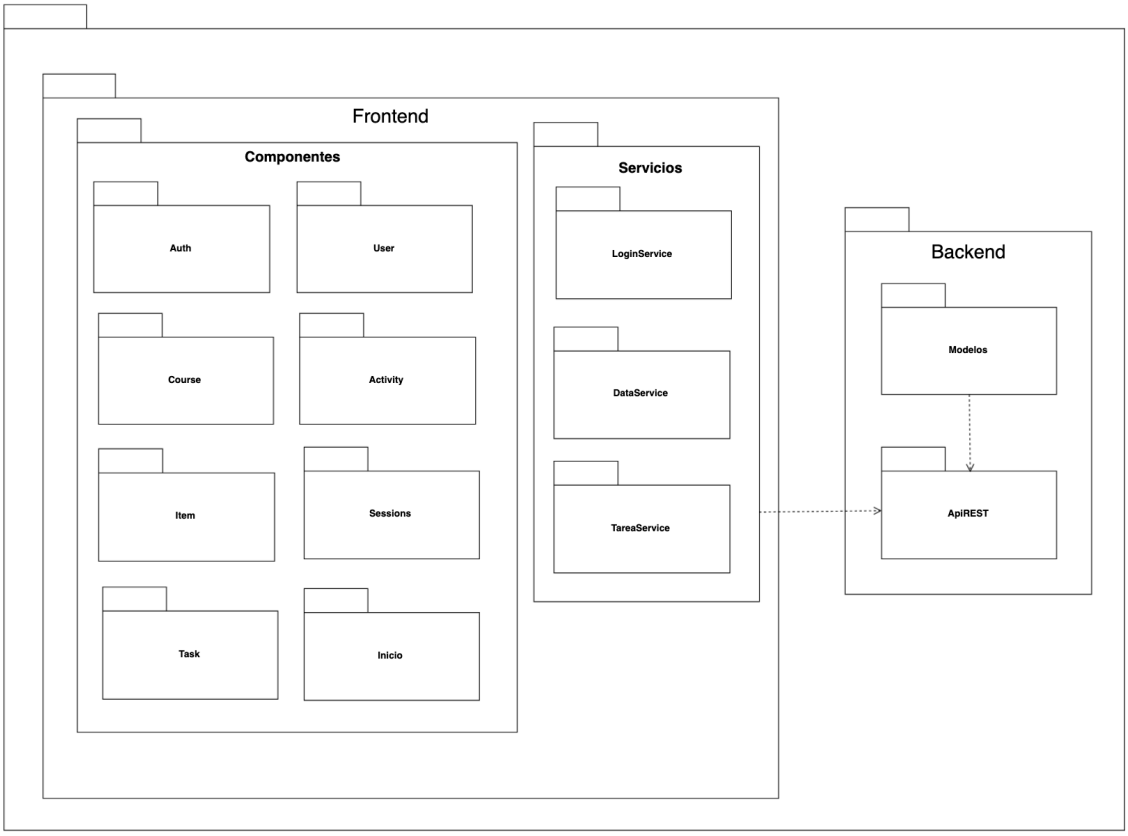


Figura 4: Subsistemas de diseño.

4 Clases de diseño

En este apartado se describen las clases de diseño resultantes de la evolución de las clases de análisis al adaptarse a la implementación concreta de los casos de uso. Las diferentes clases de diseño se exponen dentro del contexto de su subsistema para facilitar su estructuración y comprensión:

4.1 Frontend

Clase diseño ActividadComponent

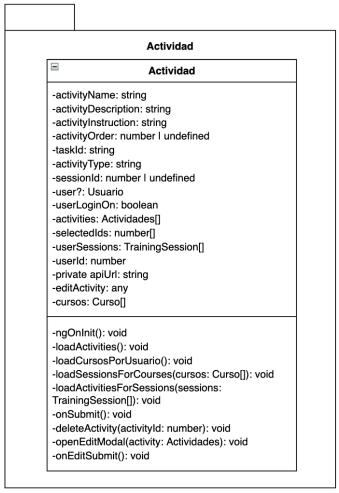


Figura 5: Clase de diseño actividad.

La clase **Actividad** gestiona las actividades que forman parte de las sesiones de entrenamiento.

Clase Diseño AuthComponent

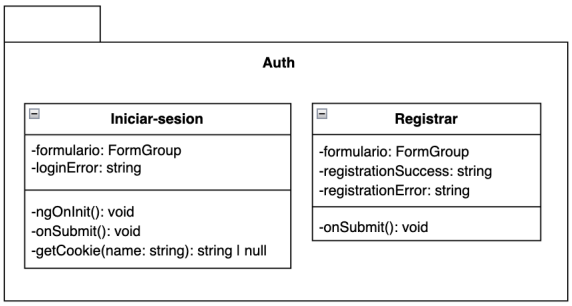


Figura 6: Clase de diseño Auth.

La clase **Auth** contiene 2 componentes el de iniciar sesión y registrar que son los componentes claves para la autenticación y autorización de usuarios, gestionando el acceso a los recursos del sistema.

Clase diseño InicioComponent

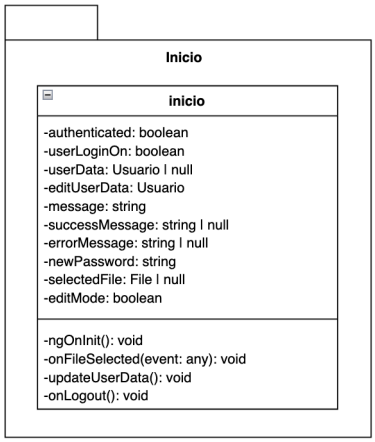


Figura 7: Clase de diseño Inicio.

La clase **Inicio** gestiona el estado inicial del sistema y la carga de datos necesarios para el funcionamiento de la aplicación.

Clase diseño ItemsComponent

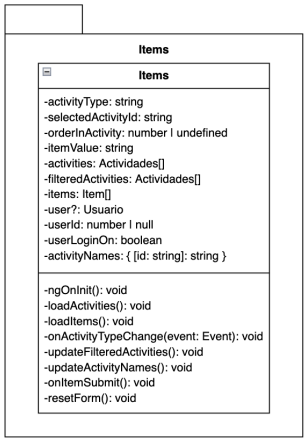


Figura 8: Clase de diseño Items.

La clase **Items** gestiona los ítems asociados a cada actividad, incluyendo la validación y el almacenamiento de los mismos.

Clase diseño SesionesComponent

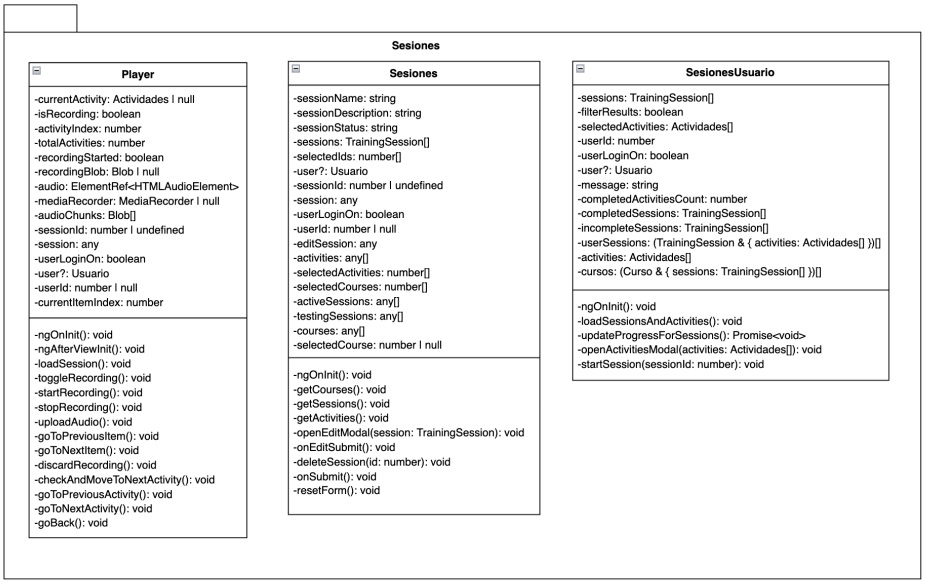


Figura 9: Clase de diseño Sesiones.

Esta carpeta también contiene varios componentes, las sesiones y el player que es el componente con en que interactua el usuario cuando está haciendo las actividades.La clase **Sesiones** gestiona las sesiones de entrenamiento, incluyendo la asociación con actividades y cursos.

Clase diseño CursosComponent

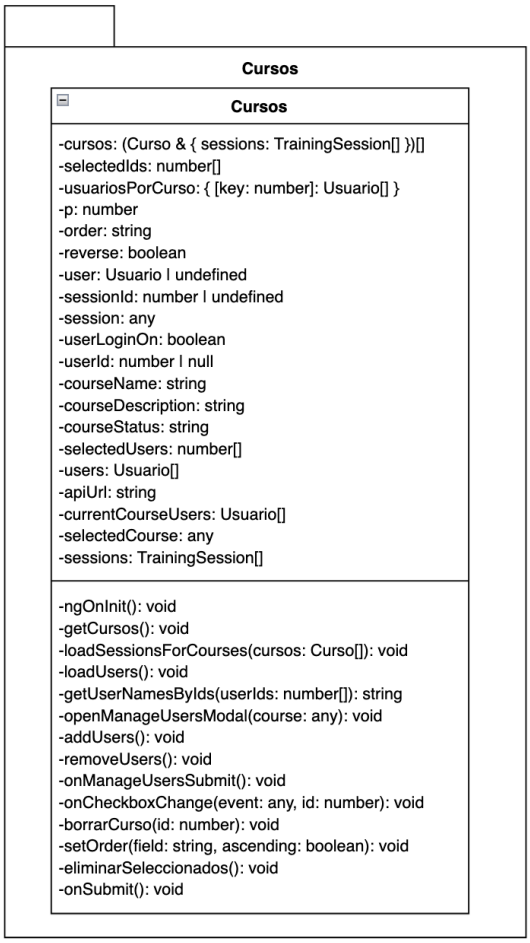


Figura 10: Clase de diseño Cursos.

La clase **Cursos** organiza los cursos de entrenamiento y gestiona la relación con las sesiones y los usuarios inscritos.

Clase diseño TaskComponent

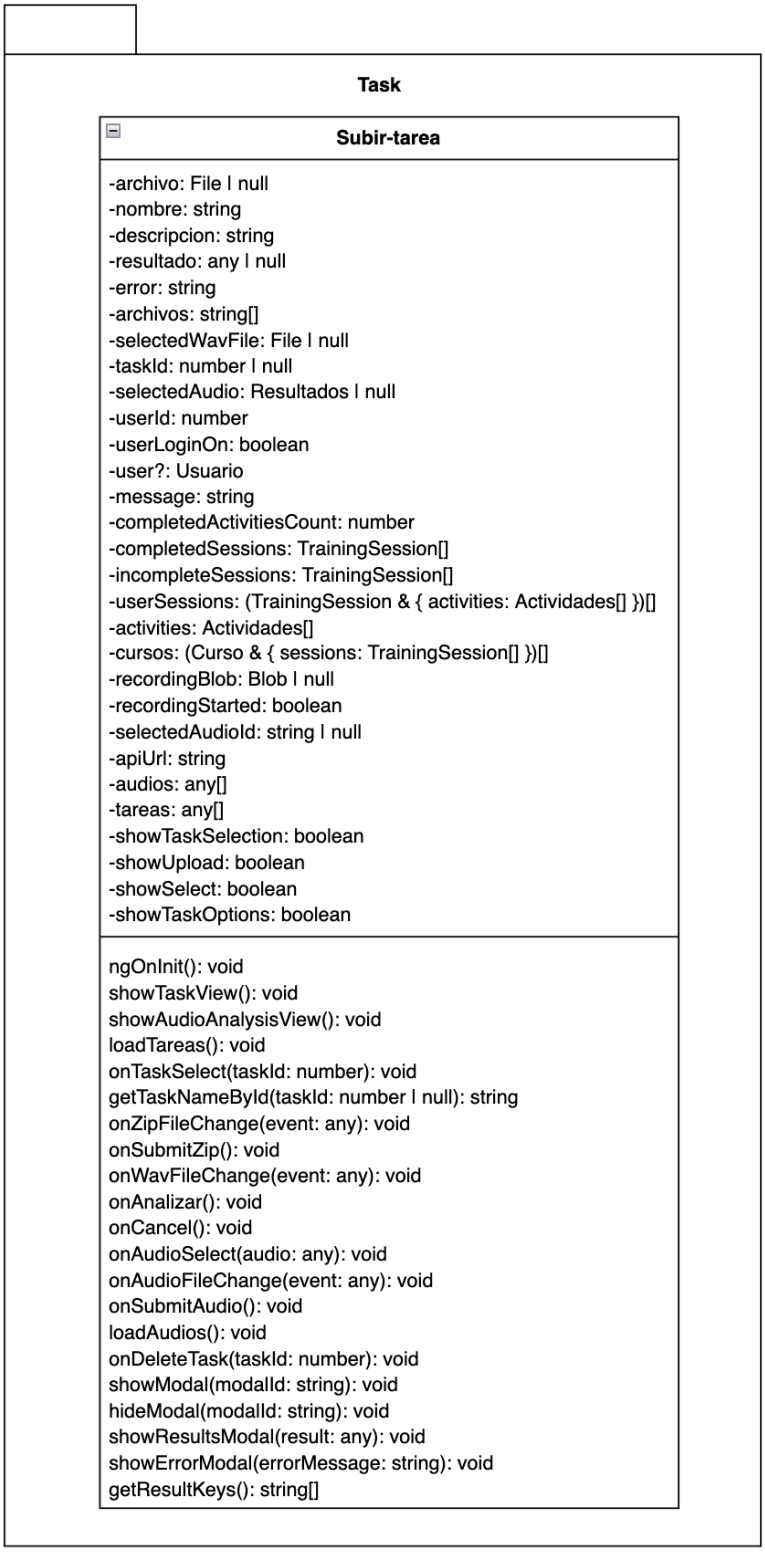


Figura 11: Clase de diseño Task.

La clase **Task** gestiona las tareas que forman parte de las actividades, incluyendo la descripción y el estado de cada tarea.

Clase diseño UserComponent

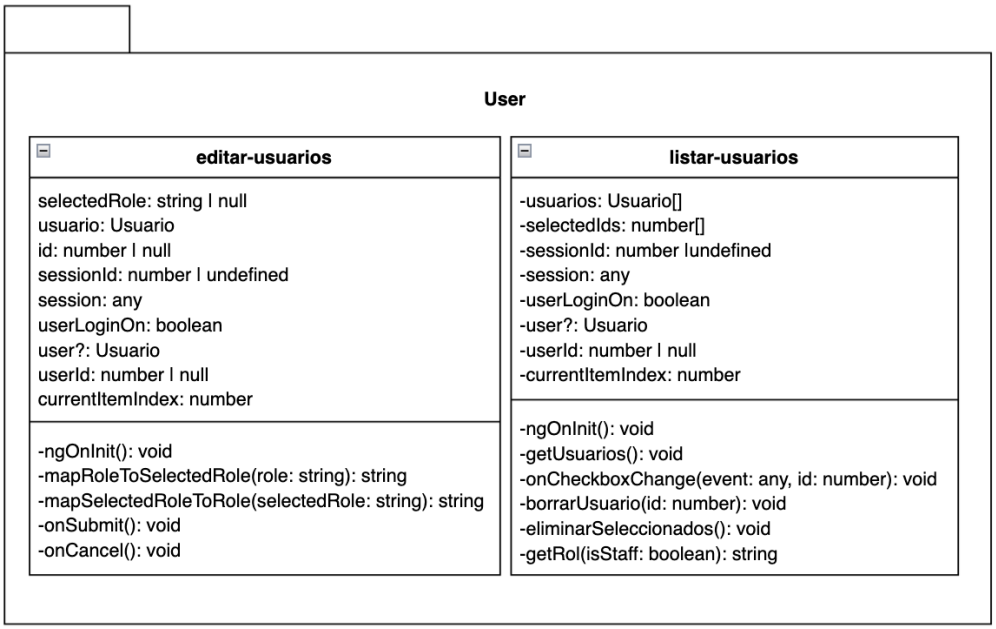


Figura 12: Clase de diseño User.

La clase **User** gestiona la información de los usuarios, incluyendo credenciales, rol y perfil. También incluye la opcion de editar los usuarios por el Administrador.

Clase diseño Servicios

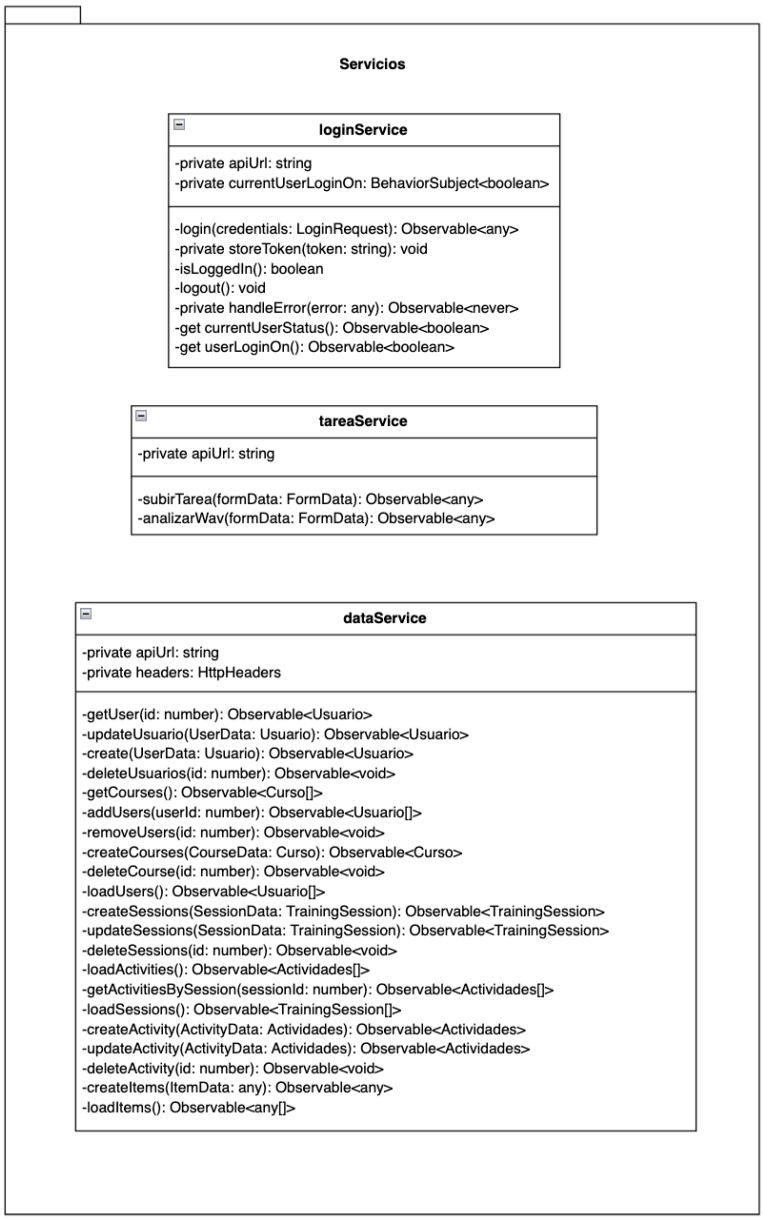


Figura 13: Clase de diseño Servicios.

La clase **Servicios** maneja la lógica de negocio y la comunicación con el backend a través de la API RESTful en el frontend.

4.2 Backend

Clase diseño apiREST

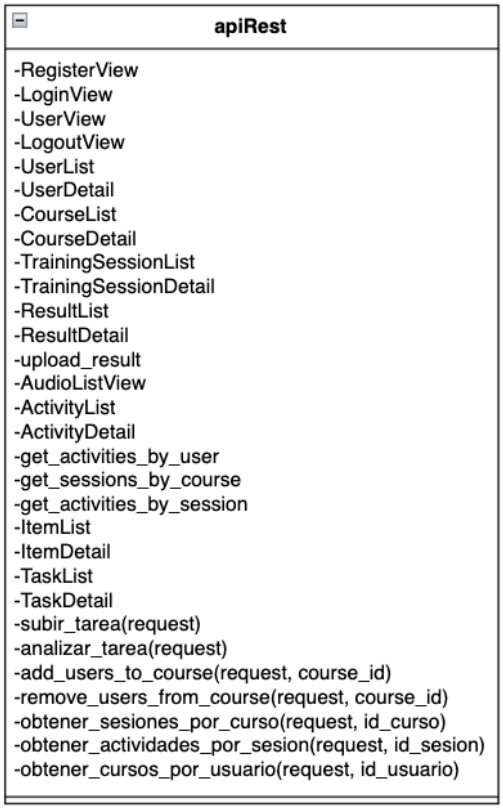


Figura 14: Clase de diseño apiREST.

La clase **apiREST** define la interfaz de la API RESTful que permite la interacción entre el frontend y el backend.

Clases de diseño de modelos

En esta sección se describen las clases de diseño correspondientes a los modelos del backend en Django. Estas clases son fundamentales para definir la estructura de los datos y ges-

tionar su persistencia en la base de datos. En la figura 15 se muestran los diferentes modelos utilizados en el sistema.

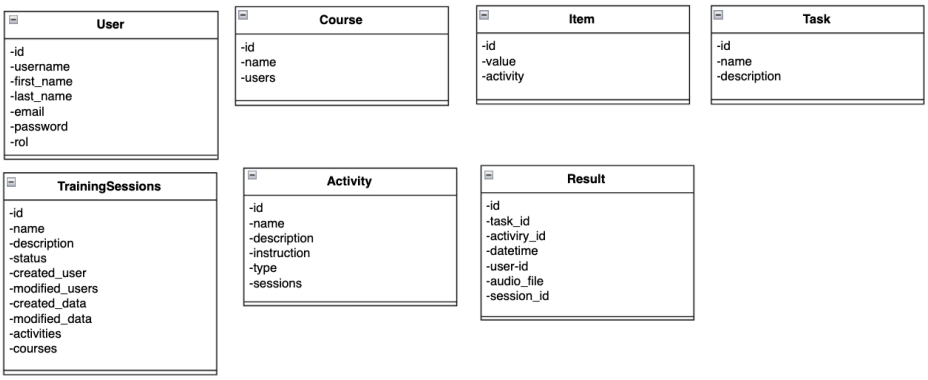


Figura 15: Diagrama de las clases de diseño de modelos.

A continuación, se proporciona una breve descripción de cada uno de estos modelos:

- **User Model:** representa a los usuarios del sistema. Es crucial para la autenticación y gestión de usuarios.
- **Actividad Model:** define las actividades dentro de las sesiones de entrenamiento. Incluye atributos como el nombre, la descripción y el tipo de actividad.
- **Sesiones Model:** gestiona las sesiones de entrenamiento y su relación con las actividades y los resultados asociados.
- **Cursos Model:** organiza los cursos de entrenamiento y gestiona las relaciones con las sesiones y los usuarios inscritos.
- **Items Model:** maneja los ítems asociados a cada actividad, incluyendo su validación y almacenamiento.
- **Task Model:** define las tareas dentro de las actividades, incluyendo descripción y estado.
- **Result Model:** gestiona los resultados de las actividades, incluyendo datos como el archivo de audio y la tarea realizada.

Cada uno de estos modelos juega un papel esencial en la gestión y manipulación de los datos en el sistema, y su interrelación asegura la integridad y la eficacia en el manejo de la información.

5 Realización de casos de uso

Ahora se define como se implementar cada uno de los casos de uso obtenidos en la etapa de captación de requisitos.

5.1 Gestión de autentificación de usuarios

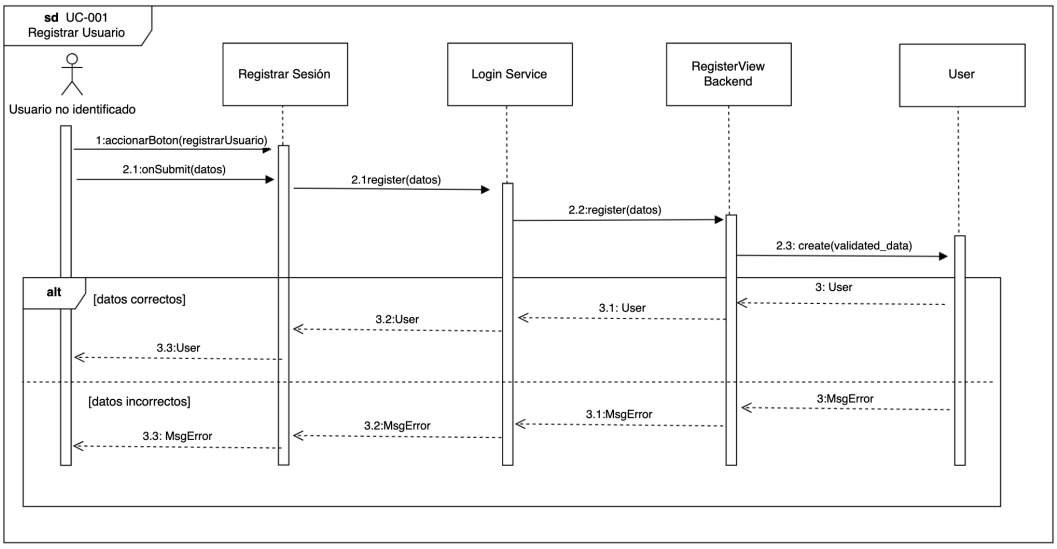


Figura 16: Diagrama de secuencia: registrar usuario.

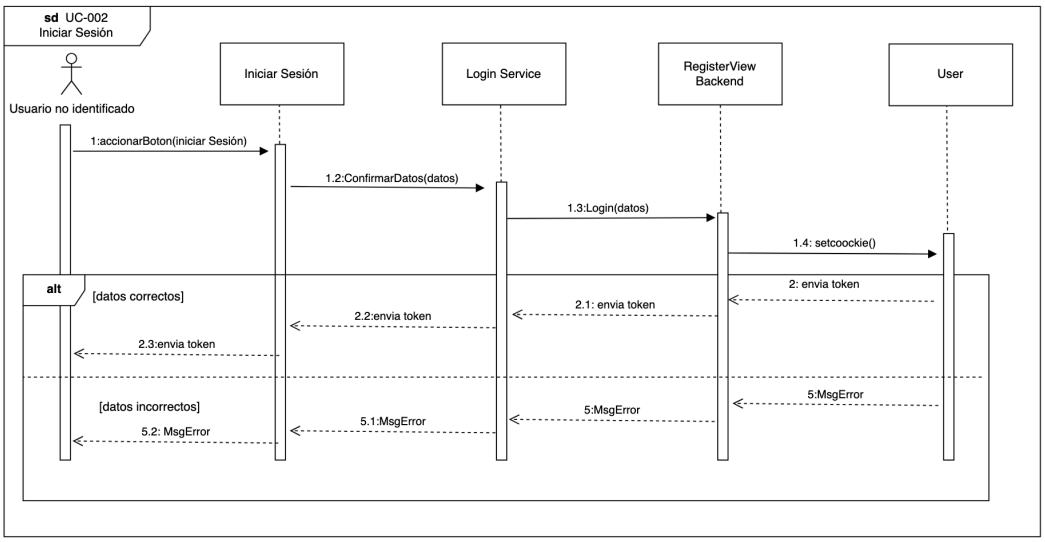


Figura 17: Diagrama de secuencia: iniciar sesión.

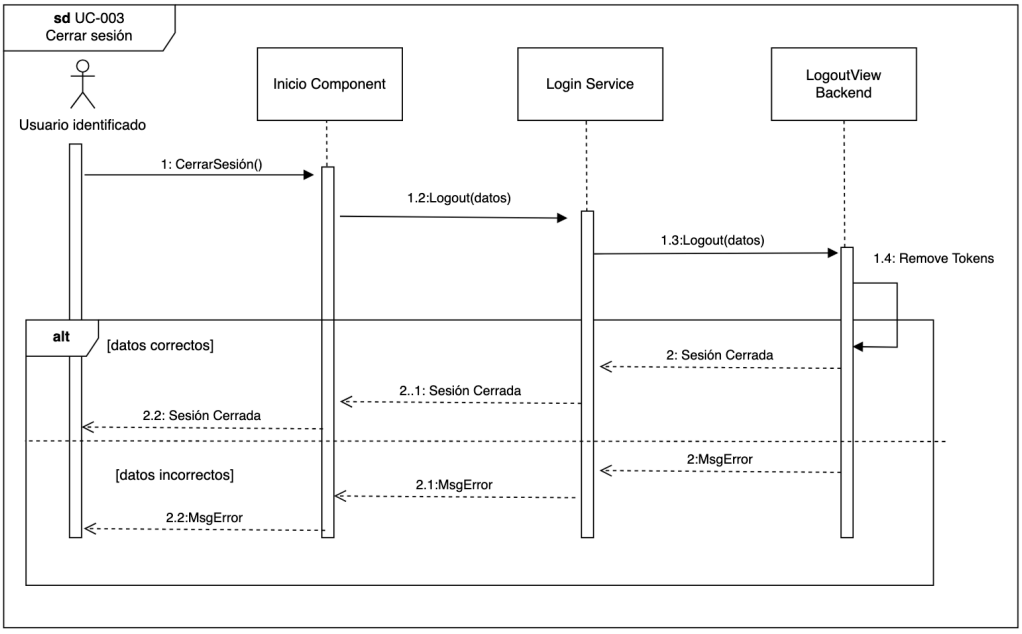


Figura 18: Diagrama de secuencia: cerrar sesión.

5.2 Gestión de usuarios

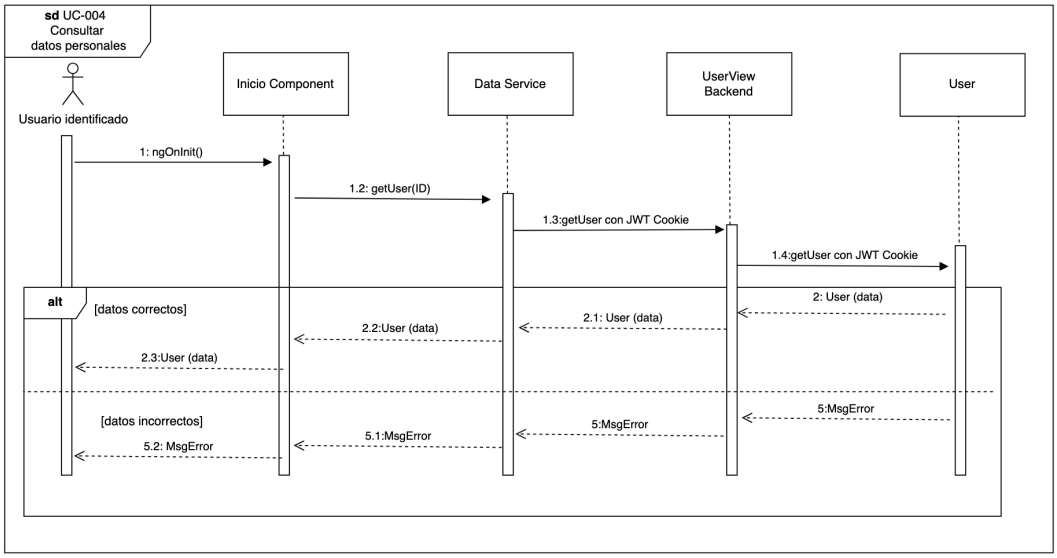


Figura 19: Diagrama de secuencia: consultar datos personales.

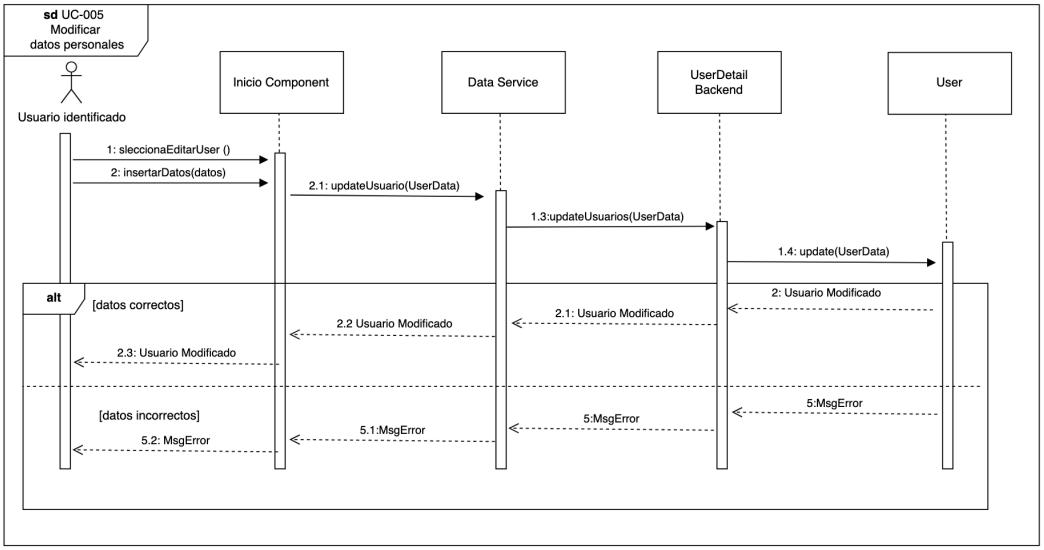


Figura 20: Diagrama de secuencia: modificar datos personales.

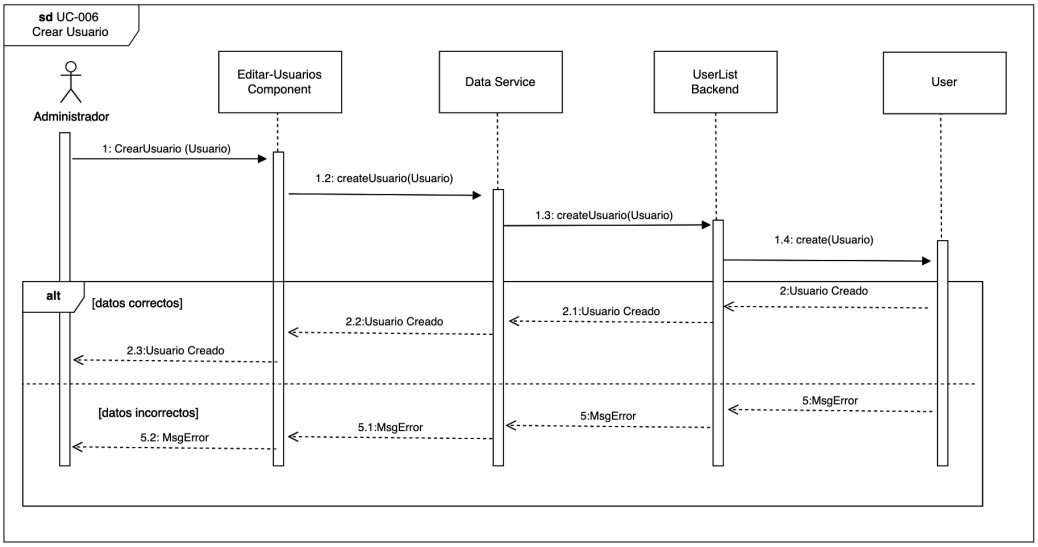


Figura 21: Diagrama de secuencia: crear usuario.

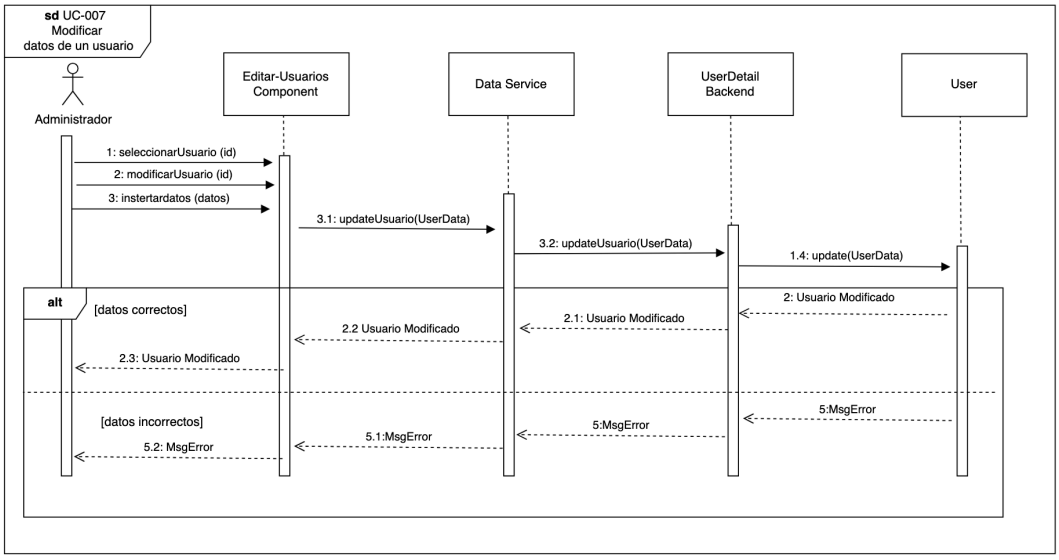


Figura 22: Diagrama de secuencia: modificar datos de un usuario.

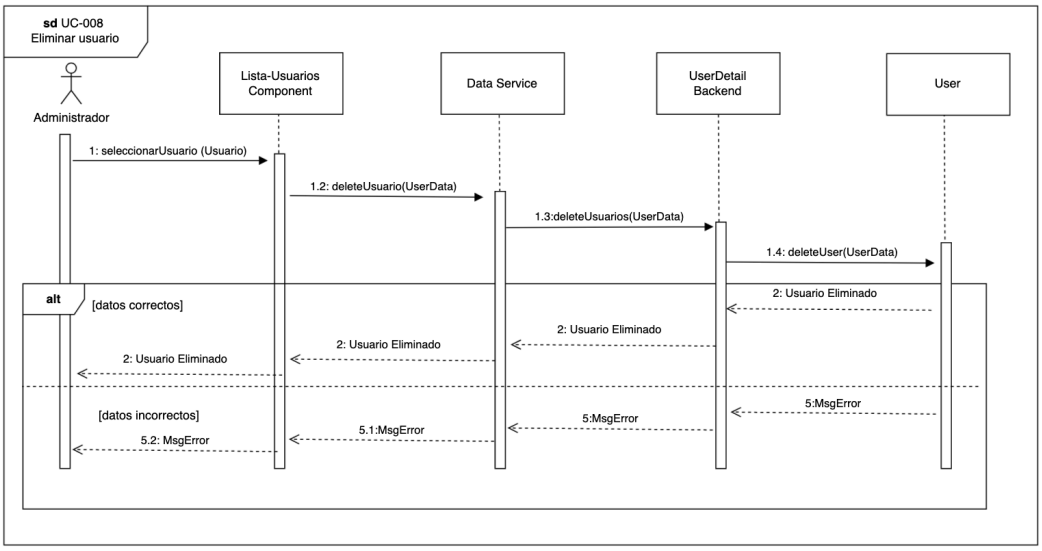


Figura 23: Diagrama de secuencia: eliminar usuario.

5.3 Gestión de cursos

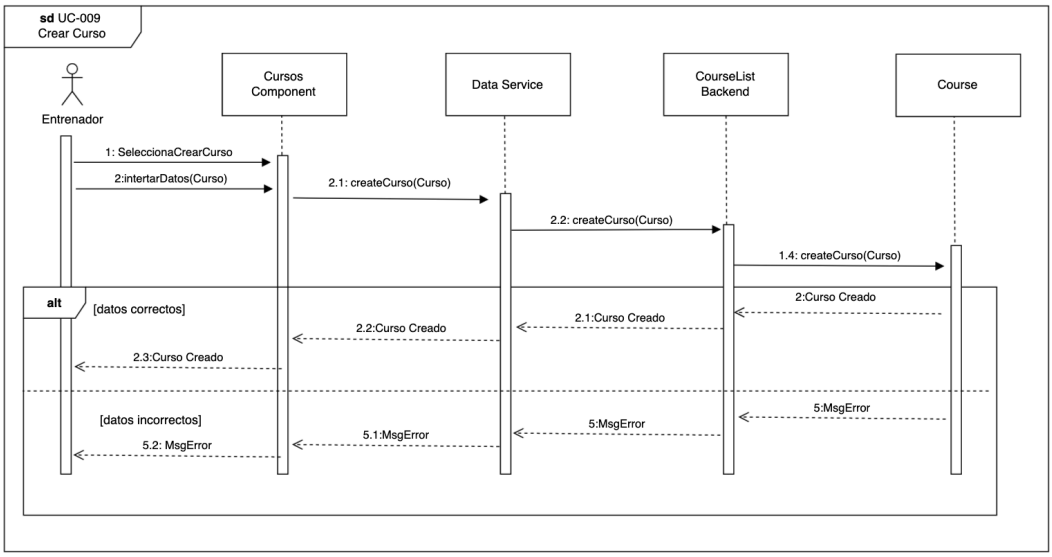


Figura 24: Diagrama de secuencia: crear curso.

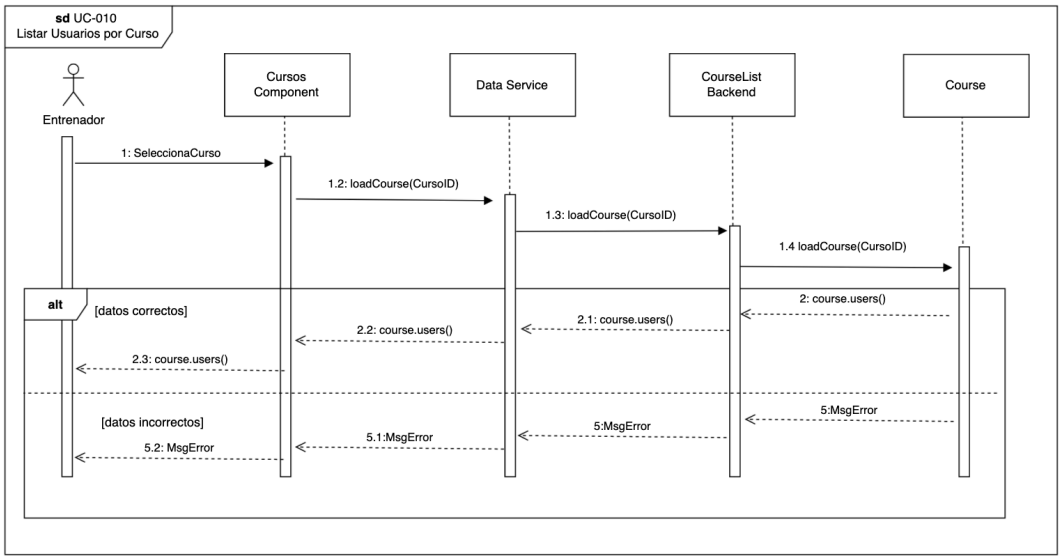


Figura 25: Diagrama de secuencia: listar usuarios por curso.

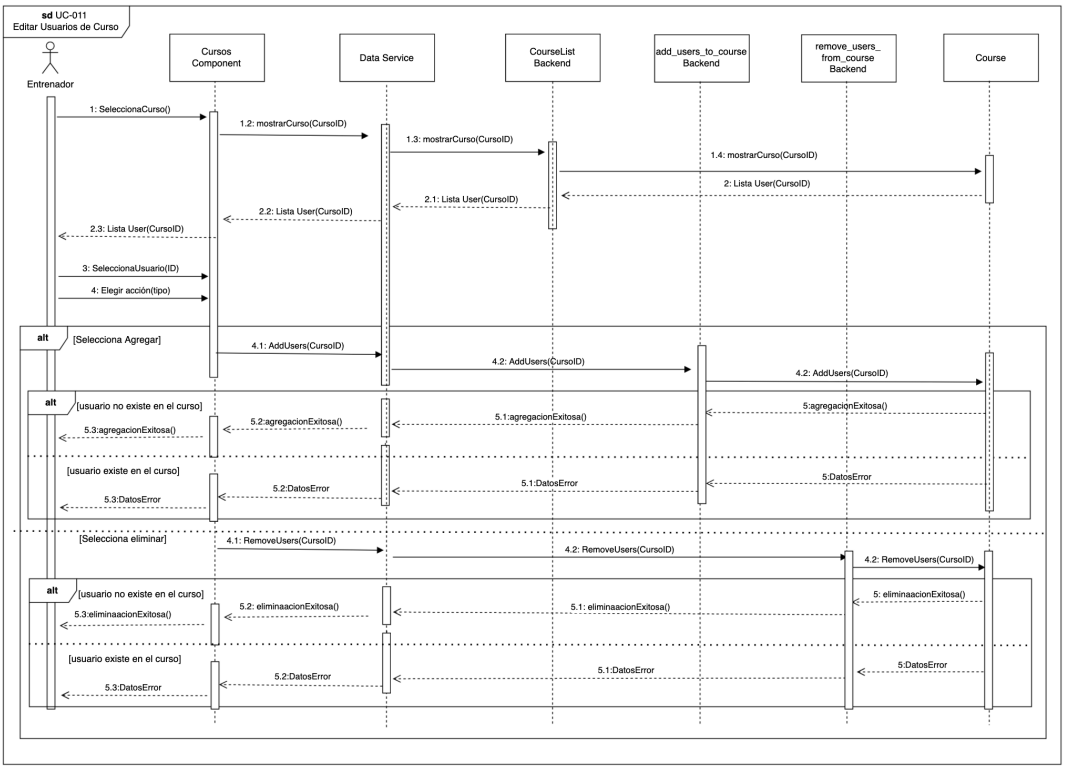


Figura 26: Diagrama de secuencia: editar usuarios de curso.

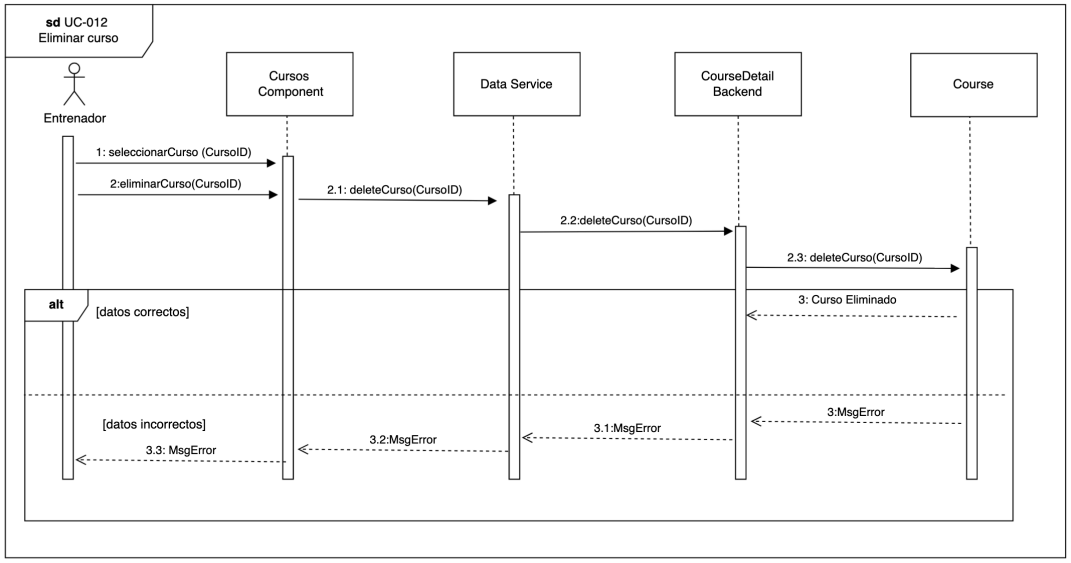


Figura 27: Diagrama de secuencia: eliminar curso.

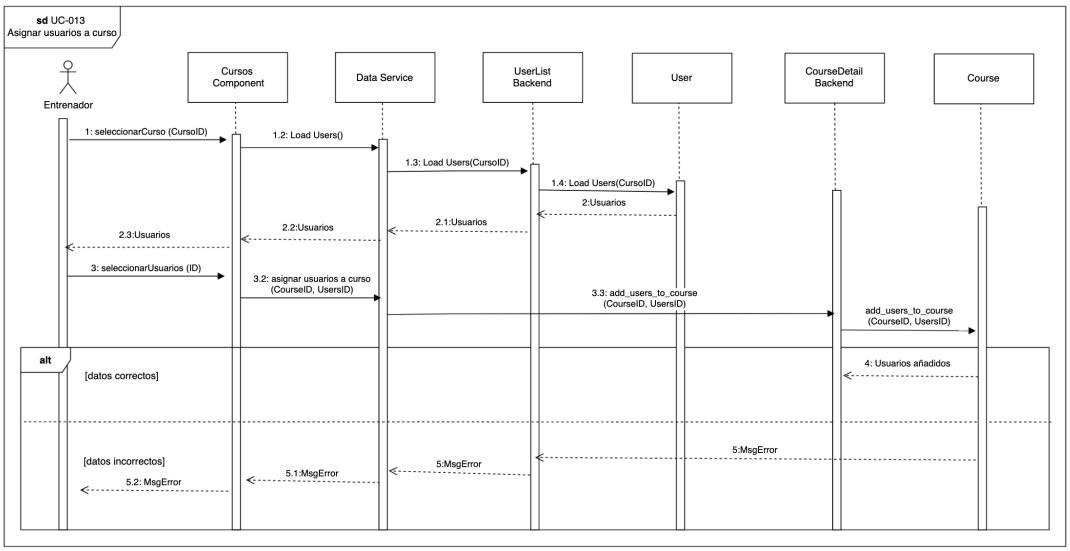


Figura 28: Diagrama de secuencia: asignar usuarios a curso.

5.4 Gestión de sesiones

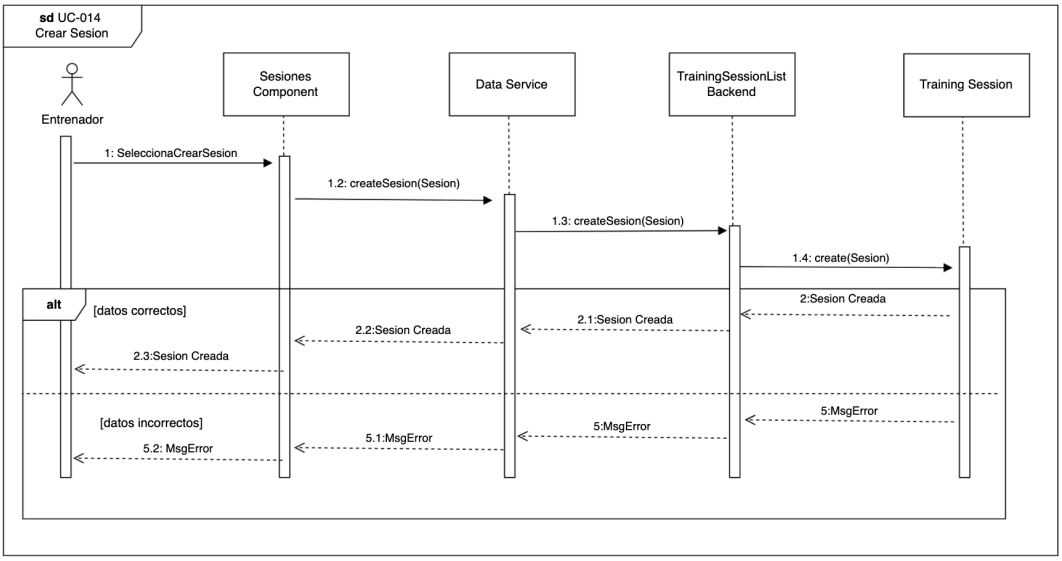


Figura 29: Diagrama de secuencia: crear sesión.

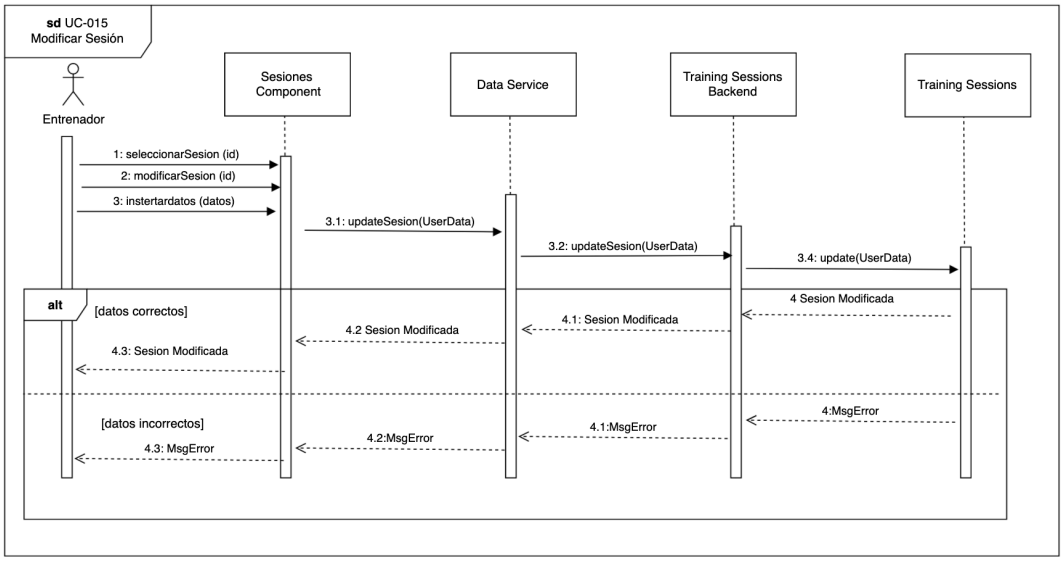


Figura 30: Diagrama de secuencia: modificar sesión.

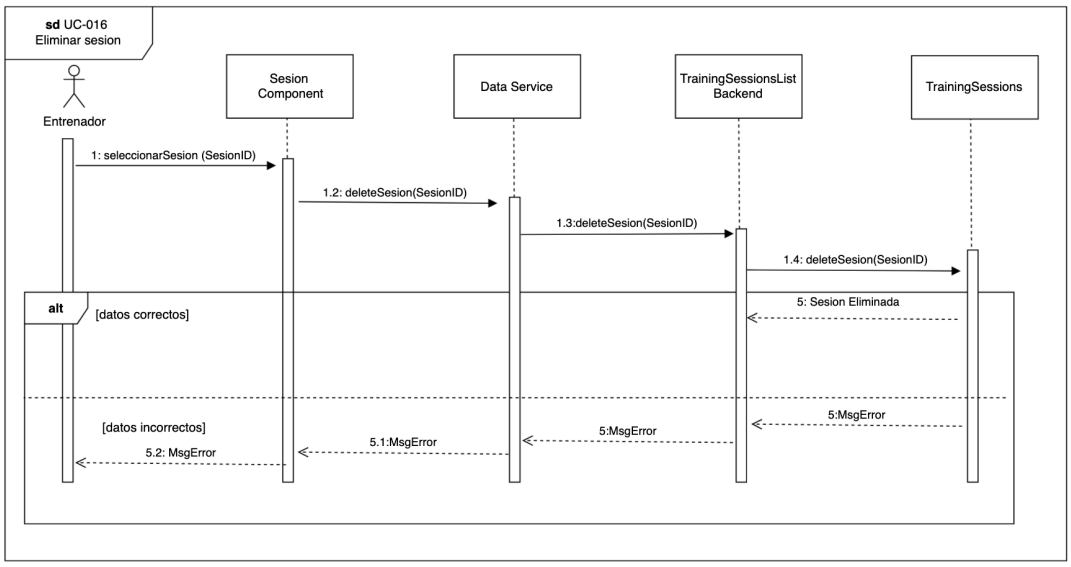


Figura 31: Diagrama de secuencia: eliminar sesión.

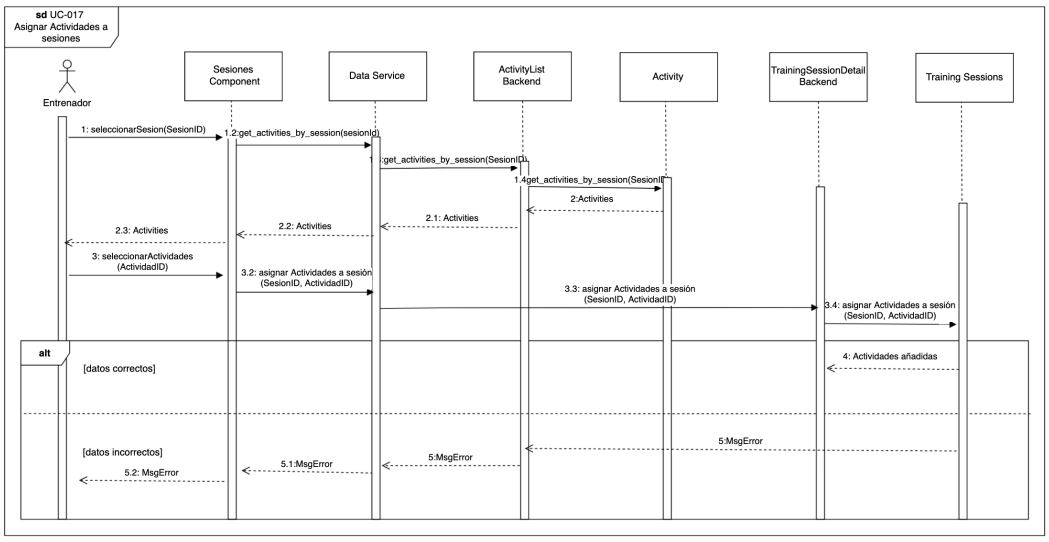


Figura 32: Diagrama de secuencia: asignar actividades a sesiones.

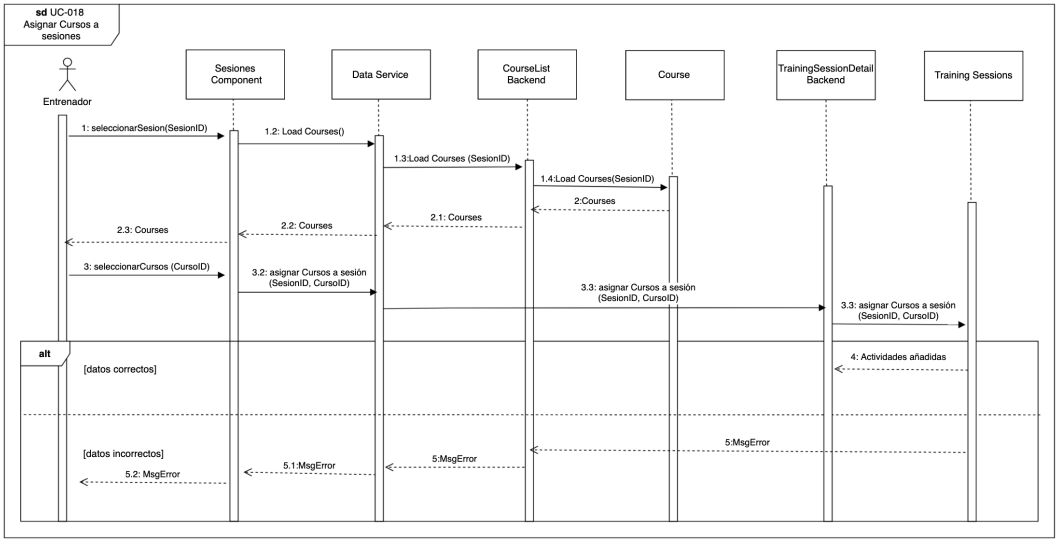


Figura 33: Diagrama de secuencia: asignar cursos a sesiones.

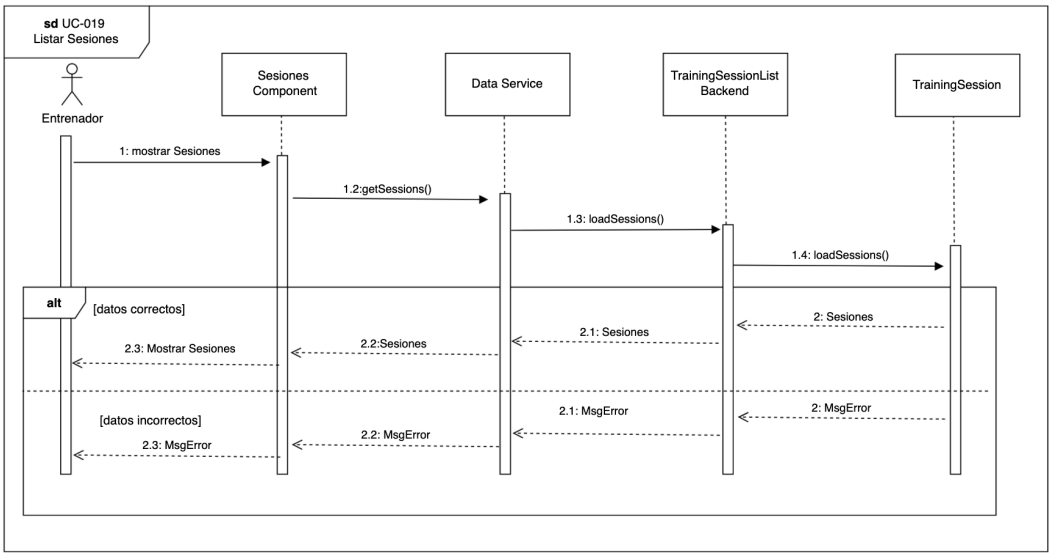


Figura 34: Diagrama de secuencia: listar sesiones.

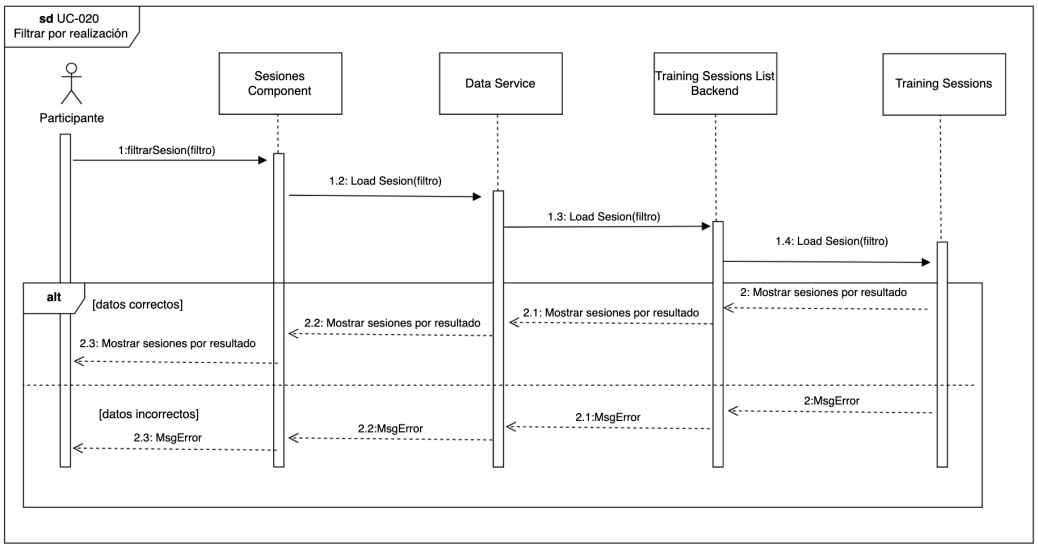


Figura 35: Diagrama de secuencia: filtrar por realización.

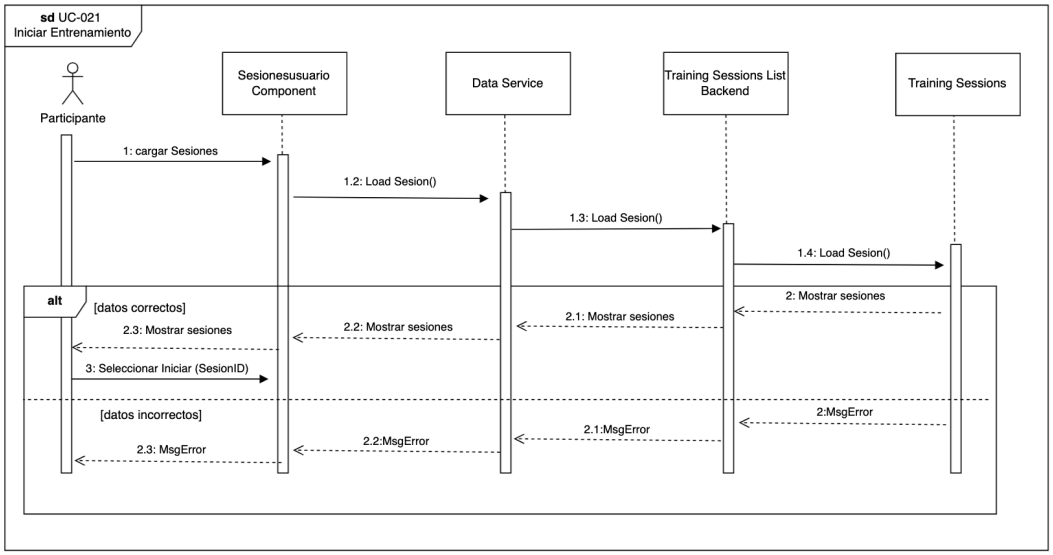


Figura 36: Diagrama de secuencia: iniciar entrenamiento.

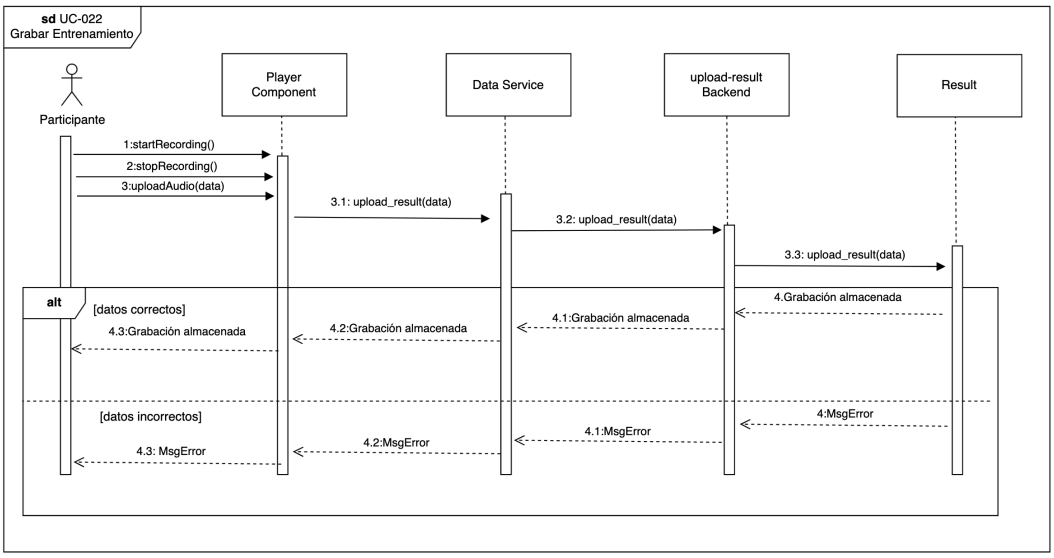


Figura 37: Diagrama de secuencia: grabar entrenamiento.

5.5 Gestión de actividades

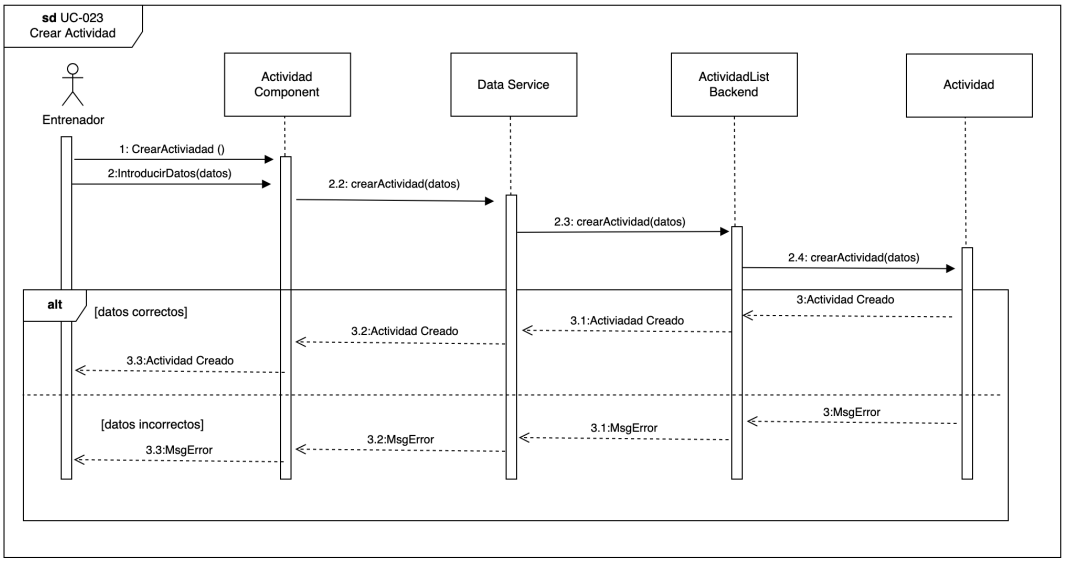


Figura 38: Diagrama de secuencia: crear actividad.

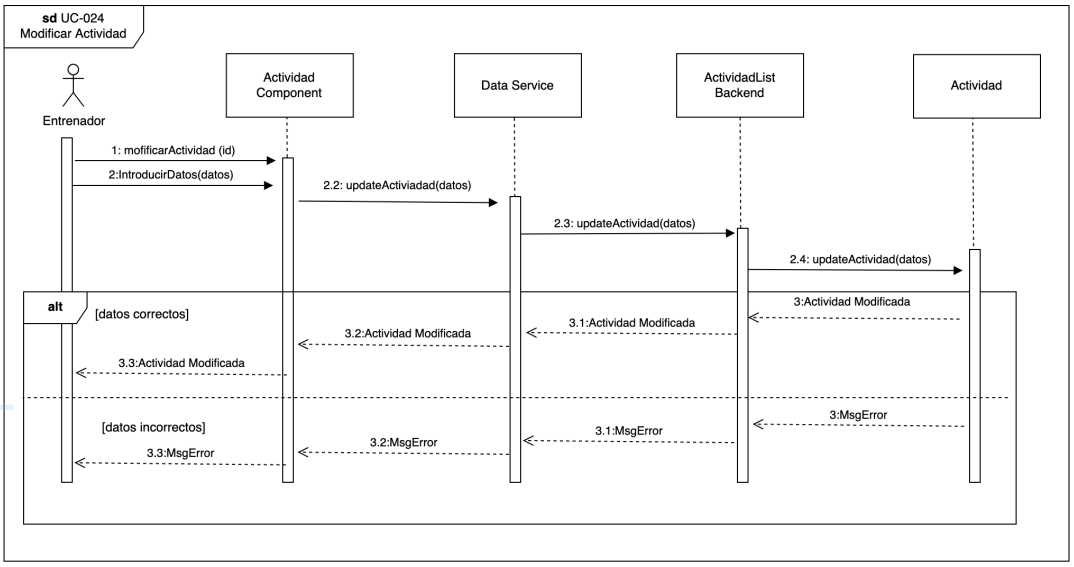


Figura 39: Diagrama de secuencia: modificar actividad.

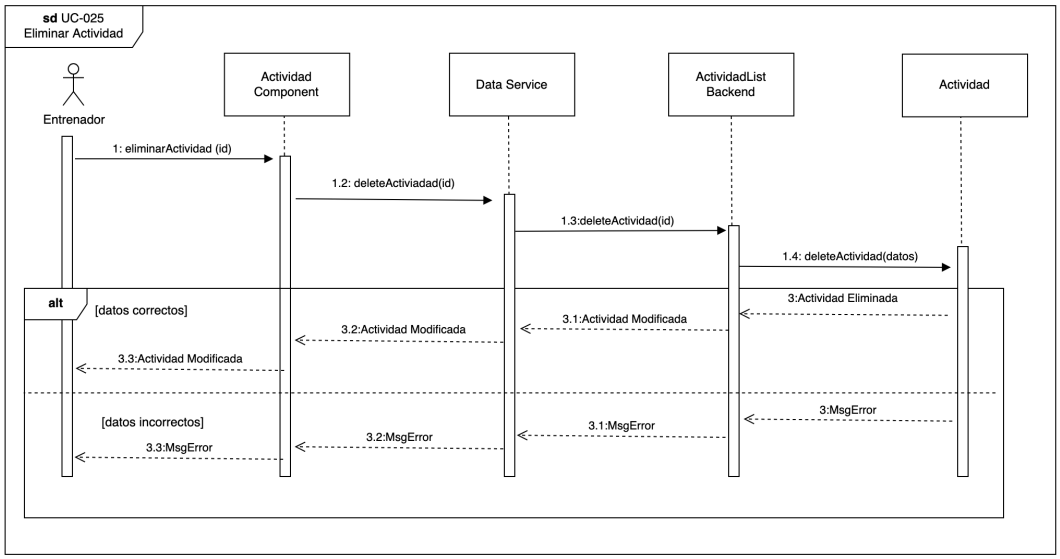


Figura 40: Diagrama de secuencia: eliminar actividad.

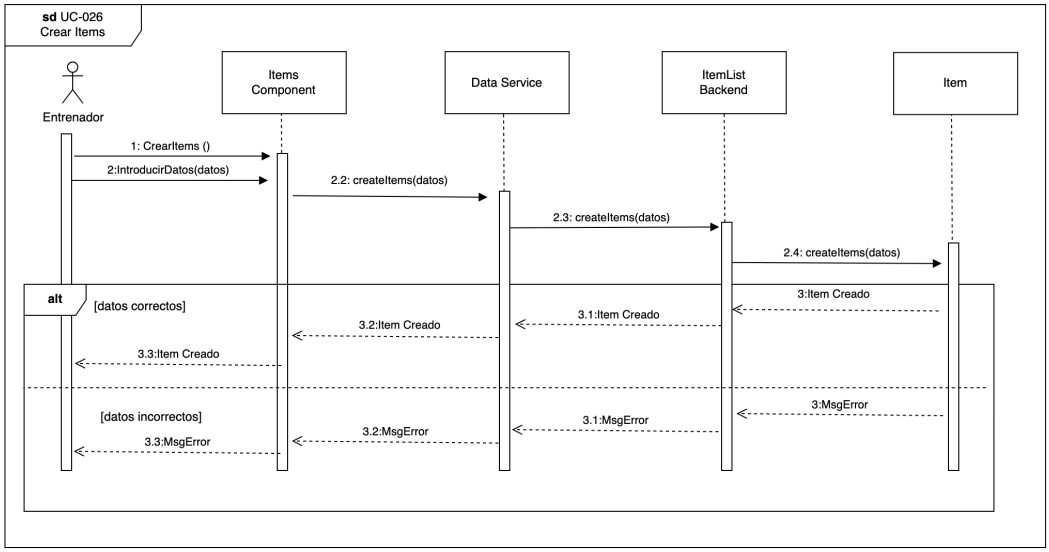


Figura 41: Diagrama de secuencia: crear items.

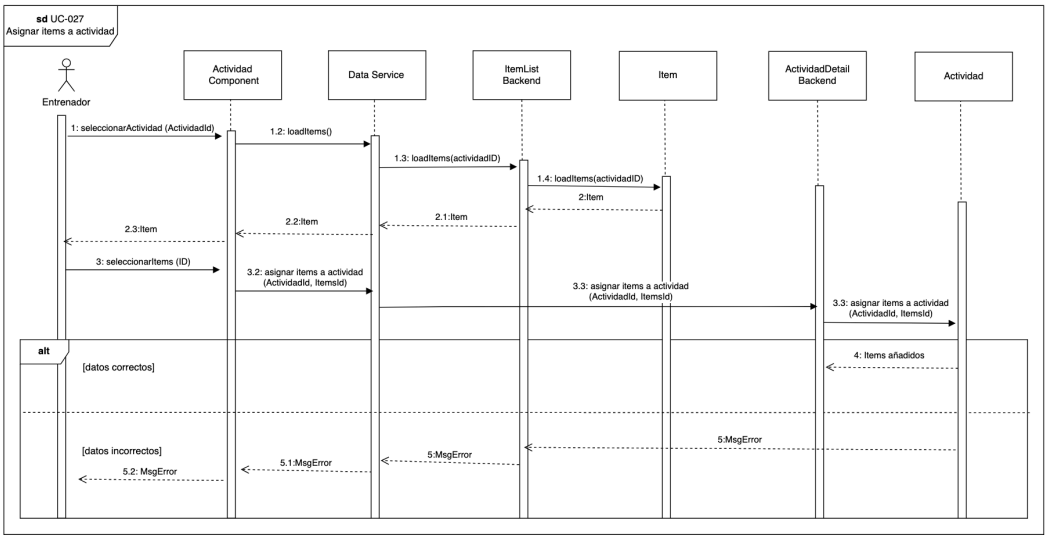


Figura 42: Diagrama de secuencia: asignar items a actividad.

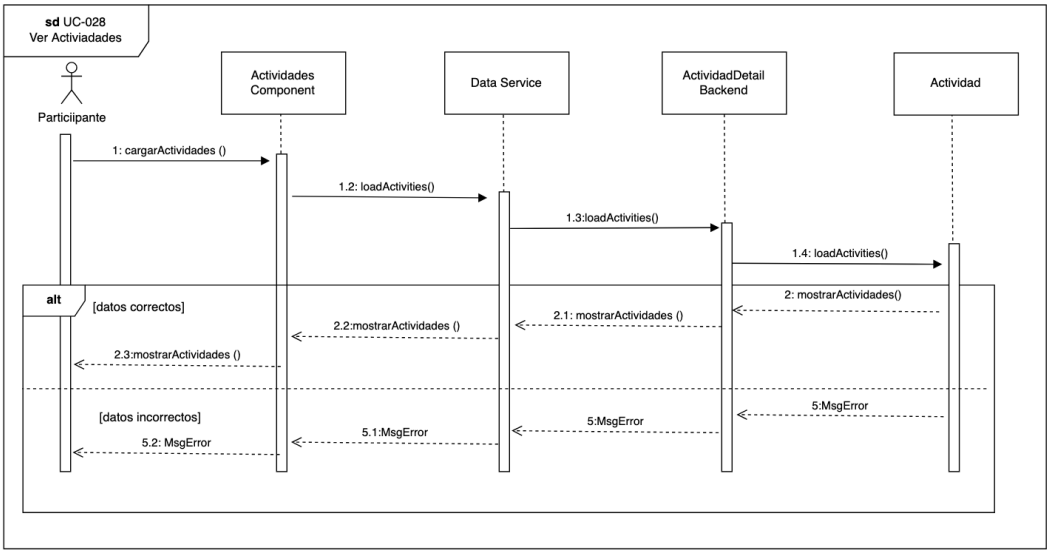


Figura 43: Diagrama de secuencia: ver actividades.

5.6 Gestión de resultados

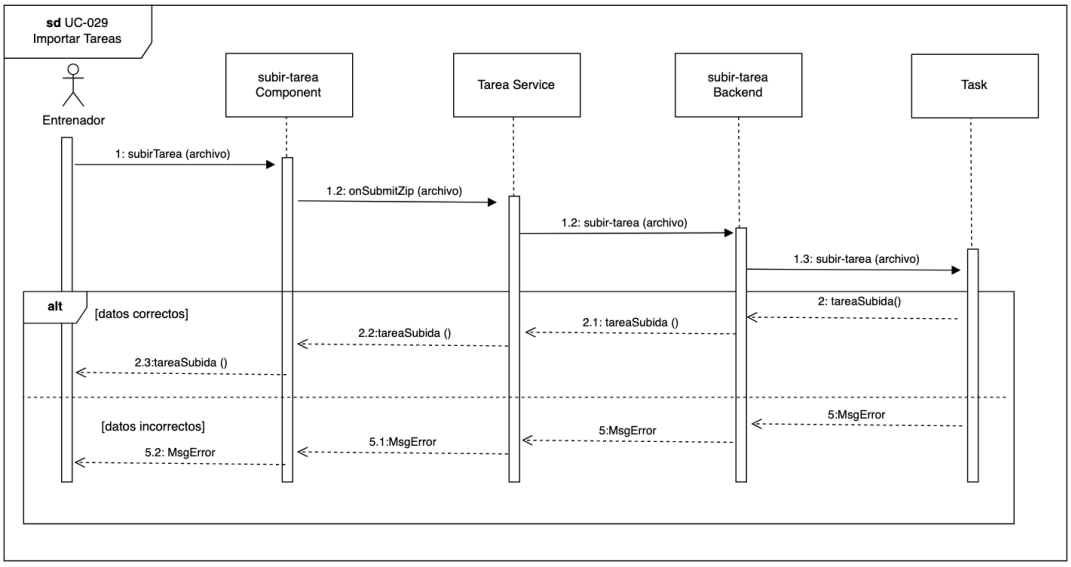


Figura 44: Diagrama de secuencia: importar tareas.

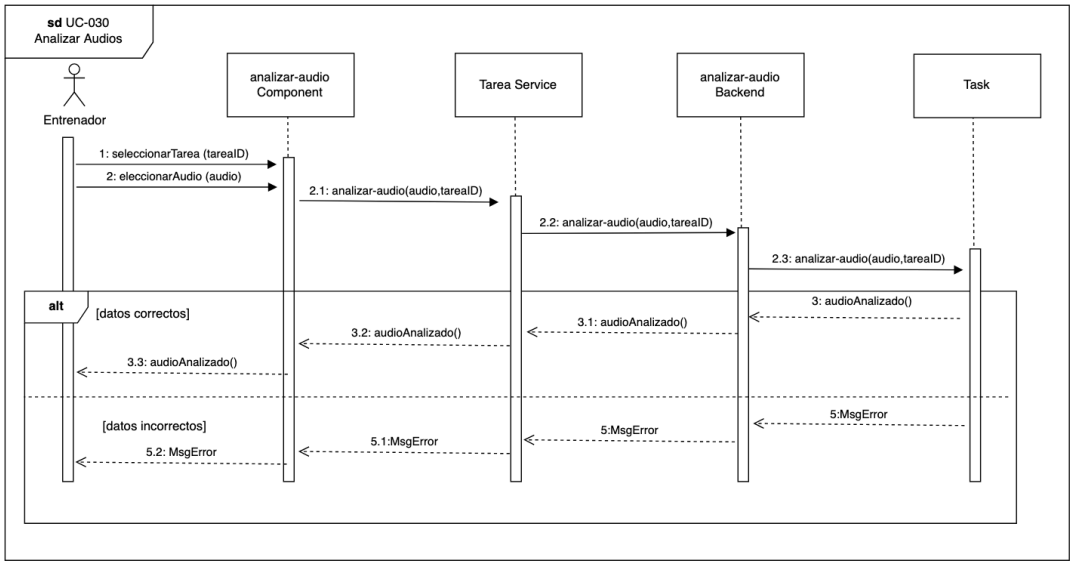


Figura 45: Diagrama de secuencia: analizar audios.

6 Base de datos

Esta es la base de Datos que muestra DBeaver al introducir los datos de mi base de datos.

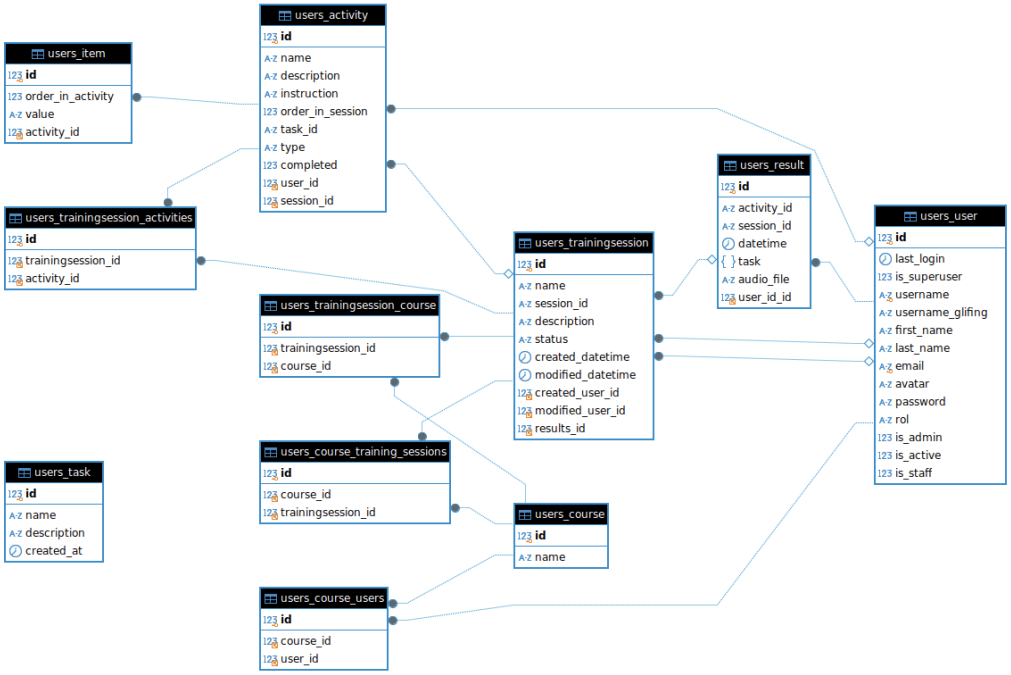


Figura 46: Base de datos con DBeaver.

7 Modelo de despliegue

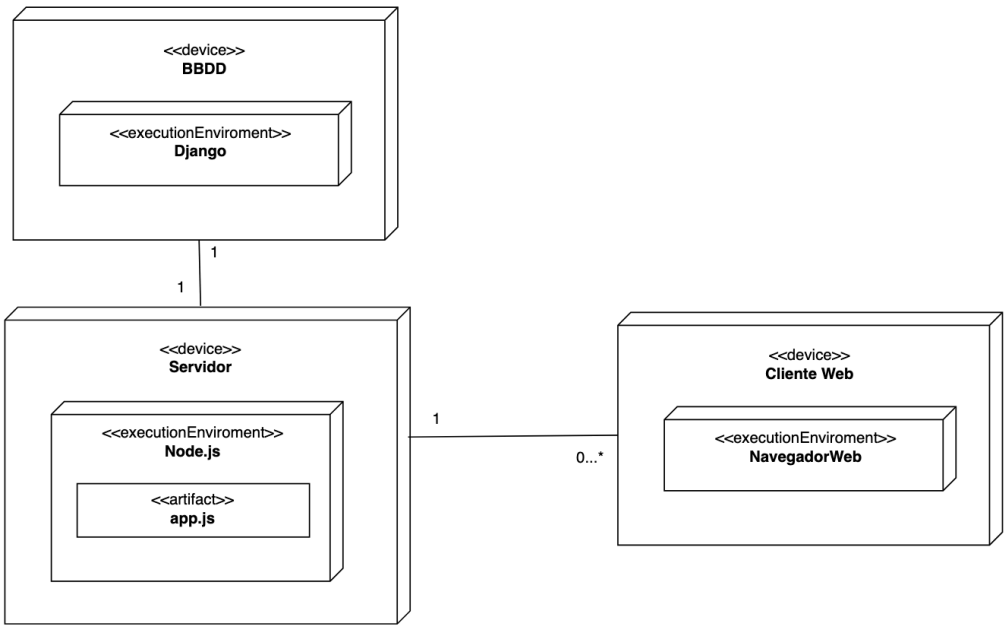


Figura 47: Modelo de despliegue.

Bibliografía

[1] Jose, D. (2022). MVVM Architecture in Angular: A Complete Guide. Recuperado de https://deepakjosecodes.com/mvvm-architecture-in-angular-guide-with-code-examples/.

[2] Modelo-Vista-Controlador. *Wikipedia, la enciclopedia libre*. Recuperado de https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador.

[3] Todo sobre la arquitectura cliente-servidor. *Arsys*. Recuperado de https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor.

[4] ¿Qué es backend y frontend?. *Descubre Comunicación*. Recuperado de https://descubrecomunicacion.com/que-es-backend-y-frontend/.

Análisis de lectura de textos
Anexo V. Manual de usuario

Trabajo de Fin de Grado
GRADO EN INGENIERÍA INFORMÁTICA

Autor: Marta de Castro Leira

Tutores:
Prof. Tutor Jesús Manuel Nieto Carracedo (Glifing S.L.),
Prof. Tutor Angélica González Arrieta

Fecha: Septiembre 2024

Índice general

1. Introducción **1**

2. Usuario no identificado **3**

2.1. Iniciar sesión 4

2.2. Registro 5

3. Usuario identificado **9**

3.1. Gestión de perfil 10

3.2. Cierre de sesión 12

4. Participante **13**

4.1. Gestión de actividades 13

4.2. Gestión de sesiones 14

5. Entrenador **19**

5.1. Gestión de sesiones de entrenamiento 19

5.2. Gestión de actividades 21

5.3. Gestión de cursos 23

5.4. Gestión de ítems 25

5.5. Gestión de resultados 27

6. Administrador **31**

6.1. Gestión de usuarios 31

Índice de figuras

1. Vista de inicio de usuario no identificado. 3

2. Vista general de la pantalla de inicio de sesión. Muestra el formulario donde los usuarios pueden ingresar su nombre de usuario y contraseña. 4

3. Vista de la pantalla de inicio de sesión con los campos de usuario y contraseña vacíos. Indica que el formulario aún no ha sido completado por el usuario. . . 4

4. Vista de la pantalla de inicio de sesión después de ingresar datos incorrectos. Se muestra un mensaje de error indicando que las credenciales proporcionadas son incorrectas. 5

5. Vista de la pantalla de registro. 6

6. Vista de la pantalla de registro cuando falta completar algún campo del formulario. 6

7. Vista de la pantalla de registro cuando el usuario ya existe. 7

8. Vista de la pantalla de registro con el mensaje de registro completado. 7

9. Vista del inicio del usuario identificado. 9

10. Formulario de modificación del perfil. 10

11. Formulario de modificación del perfil con campos incompletos. 11

12. Notificación de actualización del perfil. 11

13. Botón para cerrar la sesión. 12

14. Vista del apartado de actividades por el usuario. 14
15. Sesiones sin finalizar. 15
16. Sesiones finalizadas. 15
17. Vista de las actividades e ítems de una sesión de entrenamiento. 16
18. Vista de grabar entrenamiento con texto. 17
19. Botón de grabación accionado. 17
20. Vista de grabar entrenamiento con palabras sueltas. 18
21. Vista de gestión de sesiones de entrenamiento. 20
22. Tabla de sesiones de entrenamiento. 20
23. Modal de edición de sesión de entrenamiento. 21
24. Vista de gestión de actividades. 22
25. Modal de edición de actividad. 22
26. Vista de gestión de cursos. 23
27. Tabla de cursos con sus usuarios y sesiones. 24
28. Modal de edición de curso. 24
29. Vista de gestión de ítems. 25
30. Primer paso en la creación de un ítem. 25
31. Segundo paso en la creación de ítems. 26
32. Tabla de ítems con su valor correspondiente. 26
33. Vista de gestión de resultados en el apartado de subir tareas. 27
34. Acceso al análisis de audio tras seleccionar una tarea. 28
35. Ejemplos de resultado de análisis de audio correctamente. 28
36. Ejemplos de resultado de análisis de audio con errores. 29

37. Vista de gestión de usuarios por el administrador. 31

38. Vista de creación de usuarios. 32

39. Creación de usuarios requiere la introducción de todos los datos necesarios. . 32

1 Introducción

El objetivo de este documento es proporcionar una guía clara y detallada sobre las funcionalidades disponibles en la aplicación para los diferentes tipos de usuarios: **Administrador**, **Entrenador**, **Usuario Identificado**, y **Usuario no identificado**. Este anexo se organiza en varias secciones que corresponden a las funcionalidades específicas asociadas con cada tipo de usuario y los distintos módulos de gestión que incluye la aplicación.

Las secciones se estructuran de la siguiente manera:

- **Usuario no identificado**: descripción de las funcionalidades limitadas disponibles para un usuario que aún no se ha autenticado en el sistema.
- **Usuario identificado**: explicación detallada de las funcionalidades generales disponibles para un usuario que ha iniciado sesión.
- **Participante**: funcionalidades disponibles para un usuario que ha sido asignado como alumno de un curso, incluyendo la realización de sus sesiones de trabajo.
- **Entrenador**: funcionalidades especializadas disponibles para usuarios con el rol de entrenador, incluyendo la gestión de sesiones de entrenamiento y actividades relacionadas.
- **Administrador**: descripción de las capacidades avanzadas de gestión y administración del sistema, que incluyen la gestión de usuarios, cursos, sesiones, actividades y resultados.

2 Usuario no identificado

Un usuario no identificado es aquel que aún no se ha autenticado en el sistema. Este tipo de usuario tiene acceso limitado a ciertas funcionalidades básicas necesarias para registrarse o autenticarse. Cuando este accede a la pagina web esta es la vista que tiene:

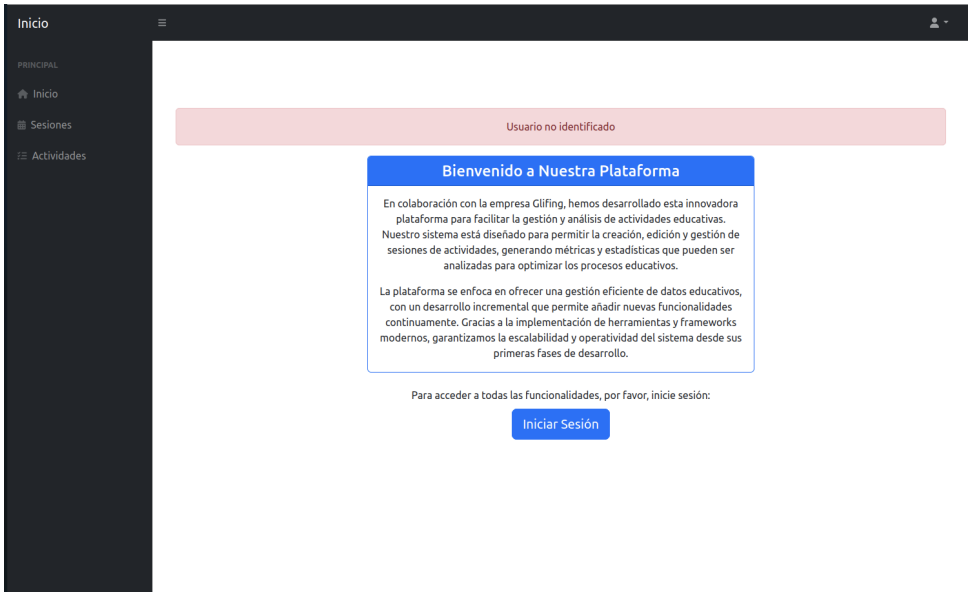


Figura 1: Vista de inicio de usuario no identificado.

Y si intenta acceder a "Actividades" o "Sesiones" se redirecciona a la ventana de Iniciar Sesión ya que requiere que el usuario este identificado para acceder a ellos.

Las ventanas accesibles para un usuario no identificado incluyen:

- **Iniciar sesión**: permite al usuario ingresar sus credenciales para acceder al sistema.
- **Registro**: proporciona un formulario para que el usuario pueda crear una cuenta nueva en el sistema.

2.1 Iniciar sesión

La interfaz de "Iniciar Sesión" permite a los usuarios ingresar su nombre de usuario y contraseña para autenticarse en la aplicación. A continuación se presentan las distintas vistas de la interfaz en diferentes estados: Figura 2, Figura 3 y Figura 4.

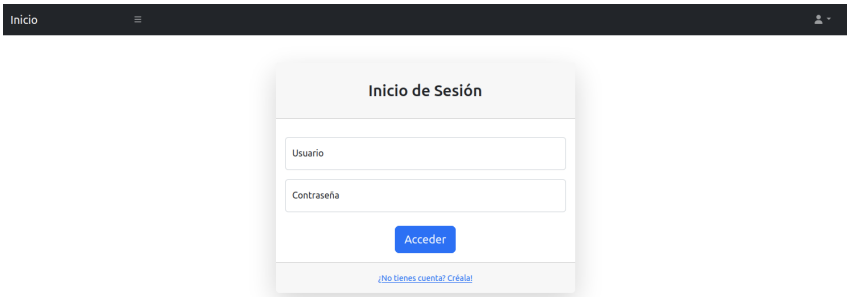


Figura 2: Vista general de la pantalla de inicio de sesión. Muestra el formulario donde los usuarios pueden ingresar su nombre de usuario y contraseña.

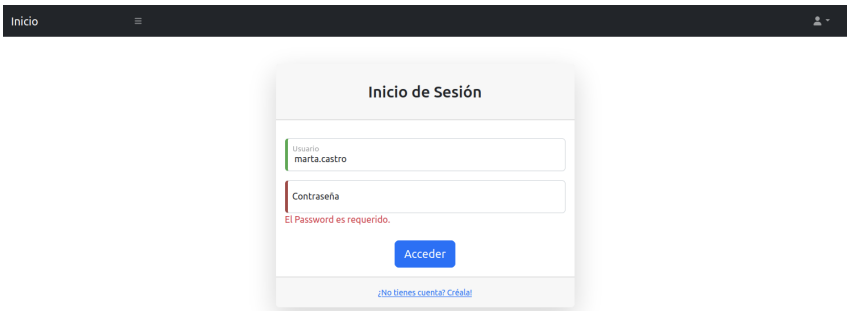


Figura 3: Vista de la pantalla de inicio de sesión con los campos de usuario y contraseña vacíos. Indica que el formulario aún no ha sido completado por el usuario.

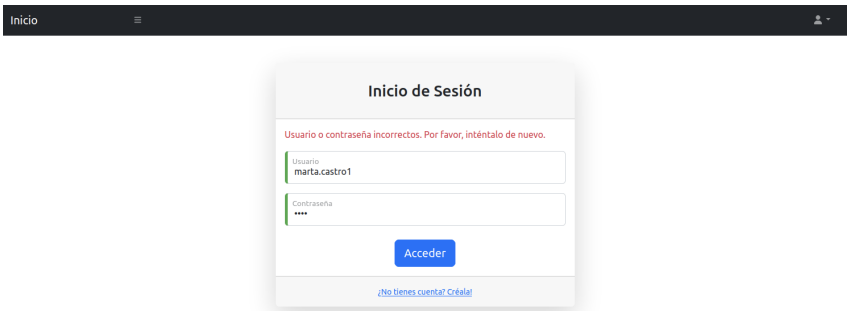


Figura 4: Vista de la pantalla de inicio de sesión después de ingresar datos incorrectos. Se muestra un mensaje de error indicando que las credenciales proporcionadas son incorrectas.

2.2 Registro

La interfaz de "Registro" permite al usuario crear una nueva cuenta introduciendo los datos requeridos en el formulario. En caso de que haya algún error en los datos proporcionados, como campos ausentes o un usuario ya existente, se muestra un mensaje de advertencia adecuado.

En la Figura 5 se muestra la vista de la pantalla de registro, donde el usuario puede introducir su información personal para crear una nueva cuenta.

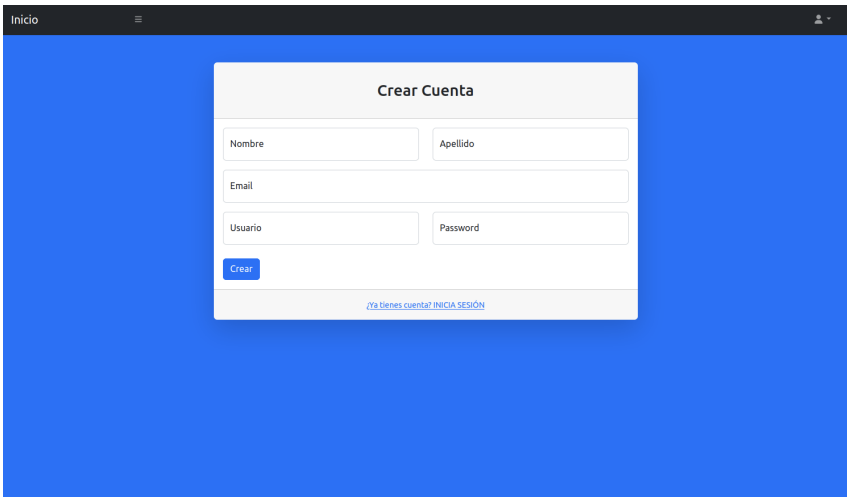


Figura 5: Vista de la pantalla de registro.

La Figura 6 ilustra la pantalla de registro cuando falta completar algún campo del formulario. En esta situación, se muestra un mensaje de advertencia para indicar qué campos están incompletos o incorrectos.

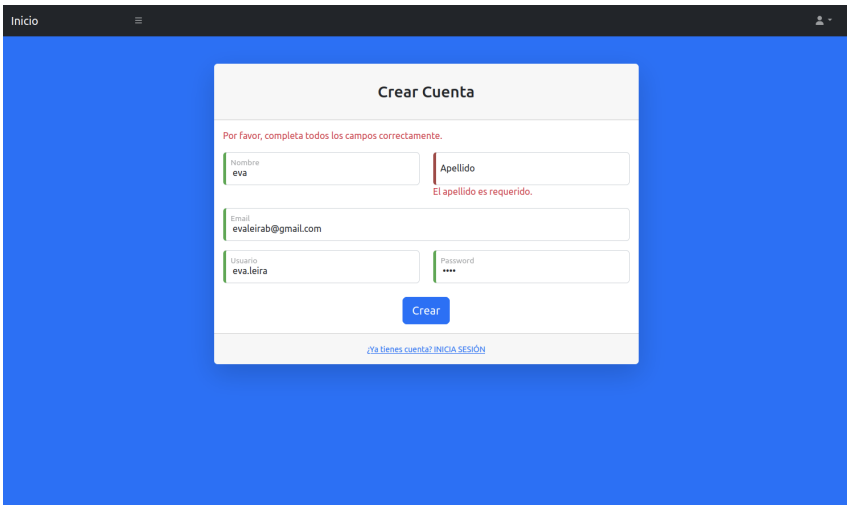


Figura 6: Vista de la pantalla de registro cuando falta completar algún campo del formulario.

La Figura 7 muestra la pantalla de registro cuando se intenta registrar un usuario que ya existe en el sistema. Se presenta un mensaje de error que indica que el usuario ya está registrado.

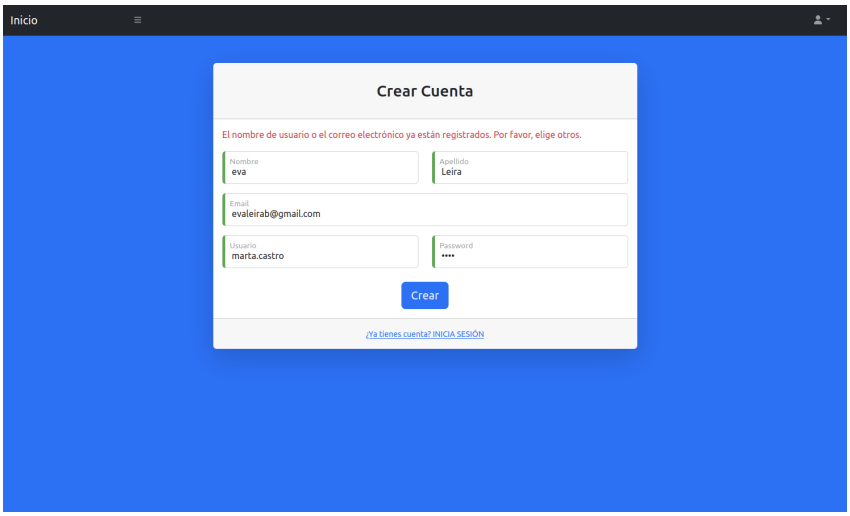


Figura 7: Vista de la pantalla de registro cuando el usuario ya existe.

Si el registro del usuario se realiza satisfactoriamente, la Figura 8 muestra la vista de la pantalla con el mensaje de éxito. Este mensaje aparece durante 3 segundos, tras lo cual la página redirige automáticamente a la pantalla de "Iniciar Sesión".

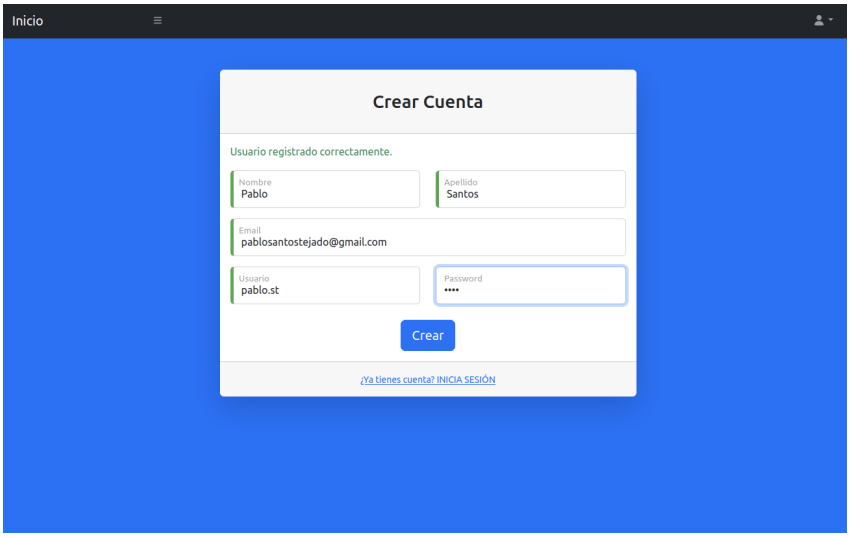


Figura 8: Vista de la pantalla de registro con el mensaje de registro completado.

3 Usuario identificado

Una vez autenticado, el usuario tiene acceso a varias funcionalidades adicionales organizadas en diferentes módulos dentro de la aplicación. La Figura 9 muestra la vista una vez que se accede a la página web.

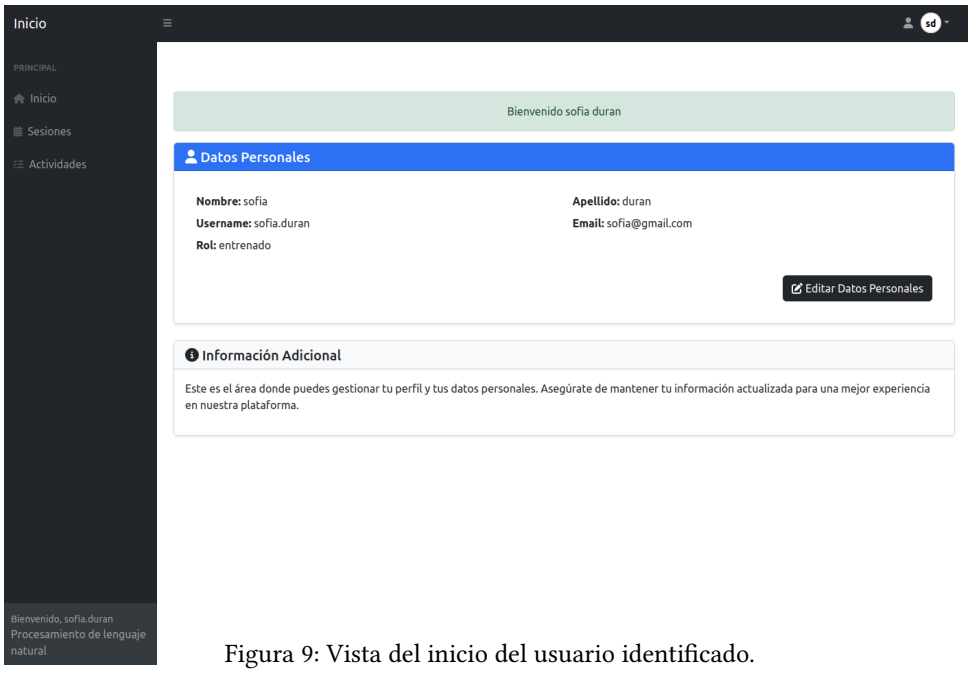


Figura 9: Vista del inicio del usuario identificado.

- **Gestión de perfil**: permite al usuario visualizar y actualizar sus datos personales, cambiar su foto de perfil y modificar su contraseña.
- **Cierre de sesión**: permite al usuario finalizar su sesión de manera segura, asegurando que ningún dato personal quede accesible a terceros y evitando el acceso no autorizado a su cuenta.

3.1 Gestión de perfil

Esta sección permite a los usuarios actualizar su información personal. Cada una de las acciones está acompañada de mensajes de confirmación o error para guiar al usuario a través del proceso.

En la Figura 10 se muestra la vista del formulario de modificación del perfil. Este formulario permite a los usuarios actualizar su información personal, incluyendo nombre, apellido, email, y otros detalles relevantes.

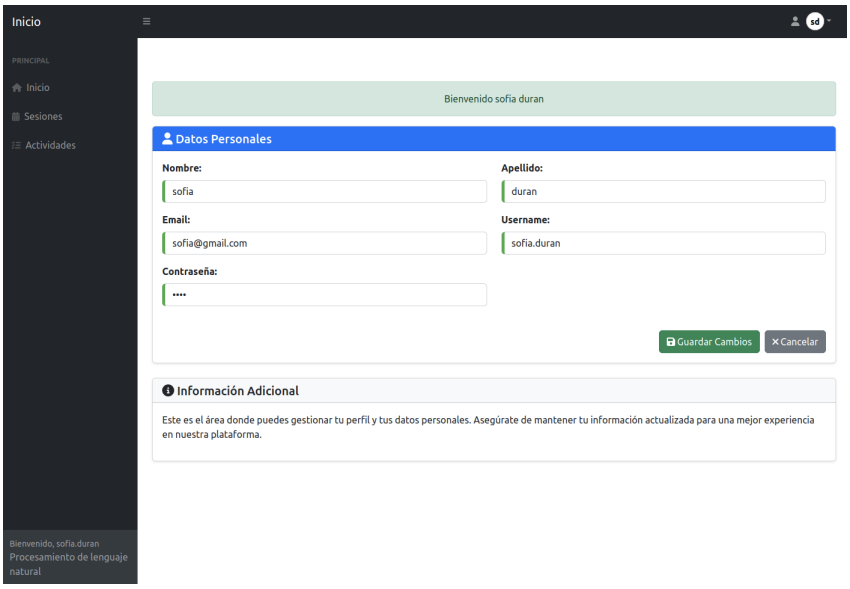


Figura 10: Formulario de modificación del perfil.

La Figura 11 ilustra el formulario de modificación del perfil cuando no se han completado todos los campos necesarios. En este caso, el botón de guardar permanece desactivado hasta que todos los campos obligatorios estén correctamente completados, garantizando que no se puedan guardar datos incompletos.

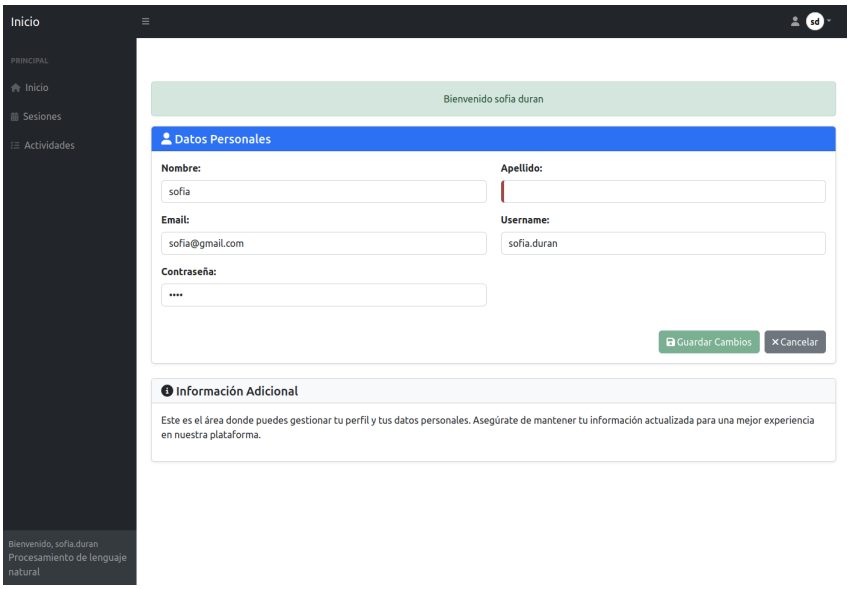


Figura 11: Formulario de modificación del perfil con campos incompletos.

Finalmente, la Figura 12 muestra la vista del inicio del usuario con una notificación de actualización exitosa que aparece por 10 segundos. Esta notificación confirma al usuario que los cambios en su perfil se han guardado correctamente.

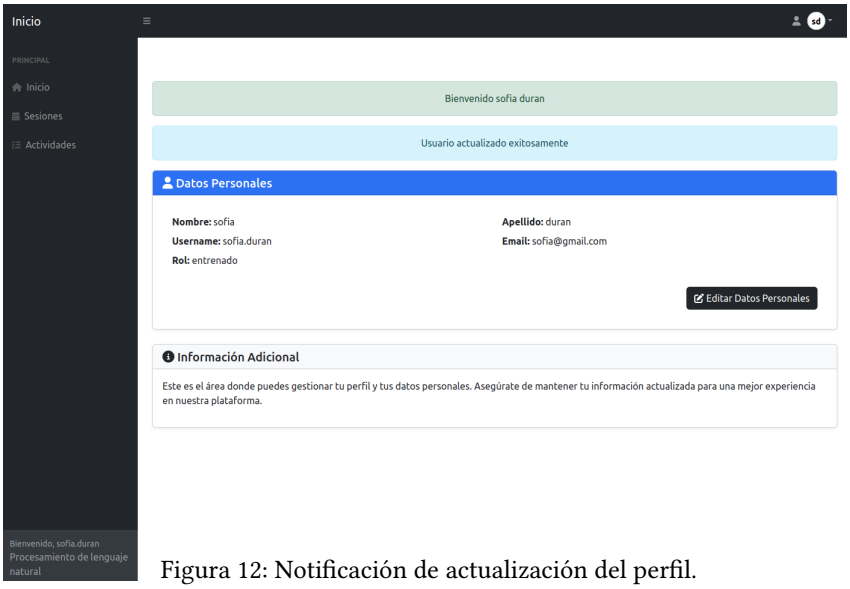


Figura 12: Notificación de actualización del perfil.

3.2 Cierre de sesión

Para cerrar la sesión de manera segura, el participante debe hacer clic en el botón "Cerrar Sesión", ubicado en la parte superior derecha de la pantalla. Al hacerlo, se garantiza que la cuenta del usuario queda protegida y que no habrá acceso no autorizado a su información personal.

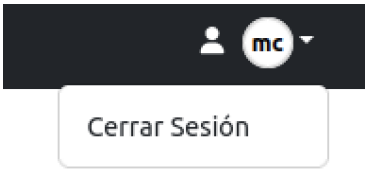


Figura 13: Botón para cerrar la sesión.

4 Participante

Un usuario con el rol de Participante tiene acceso a funcionalidades específicas dentro del sistema, que le permiten interactuar y participar en sus entrenamientos asignados.

- **Gestión de actividades**: permite la visualización de las actividades asignadas al usuario y las sesiones a las que pertenecen dichas actividades. Los usuarios pueden acceder a esta funcionalidad a través de la sección de actividades en la barra lateral.
- **Gestión de sesiones**: permite la visualización y gestión de las sesiones a las que pertenece el usuario. Los usuarios pueden diferenciar entre las sesiones con todas las actividades finalizadas y aquellas que aún tienen actividades pendientes.

4.1 Gestión de actividades

Si los usuarios acceden a la sección de actividades de la barra de lateral accederán a este apartado. En esta sección, los usuarios pueden simplemente visualizar las actividades que tienen asignadas y las sesiones a las que pertenecen dichas actividades

Figura 14: Vista del apartado de actividades por el usuario.

4.2 Gestión de sesiones

En esta sección, los usuarios pueden visualizar las sesiones que tienen asignadas y diferenciar entre aquellas en las que todas las actividades están finalizadas y las que aún no están completas. Además, los usuarios pueden observar las actividades junto con sus ítems por sesión, y también verificar el número de actividades pendientes en caso de que alguna sesión esté incompleta, como se muestra en la Figura 15.

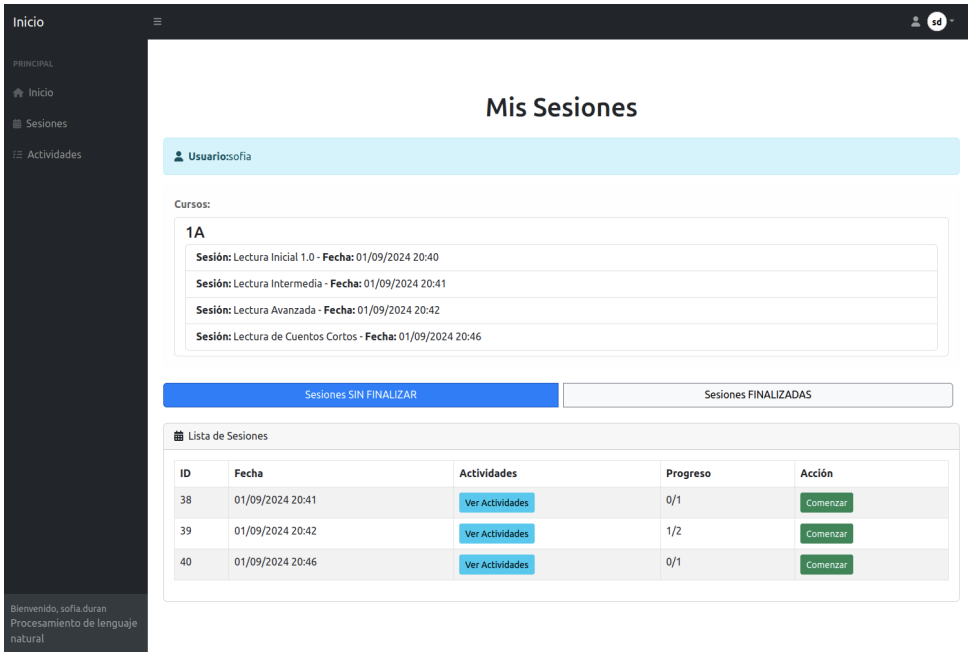


Figura 15: Sesiones sin finalizar.

Por otro lado, en la Figura 16, se muestran las sesiones que ya han sido completadas. Dentro de cada sesión, los usuarios tienen la capacidad de revisar las actividades y los ítems de la sesión, como se puede observar en la Figura 17.

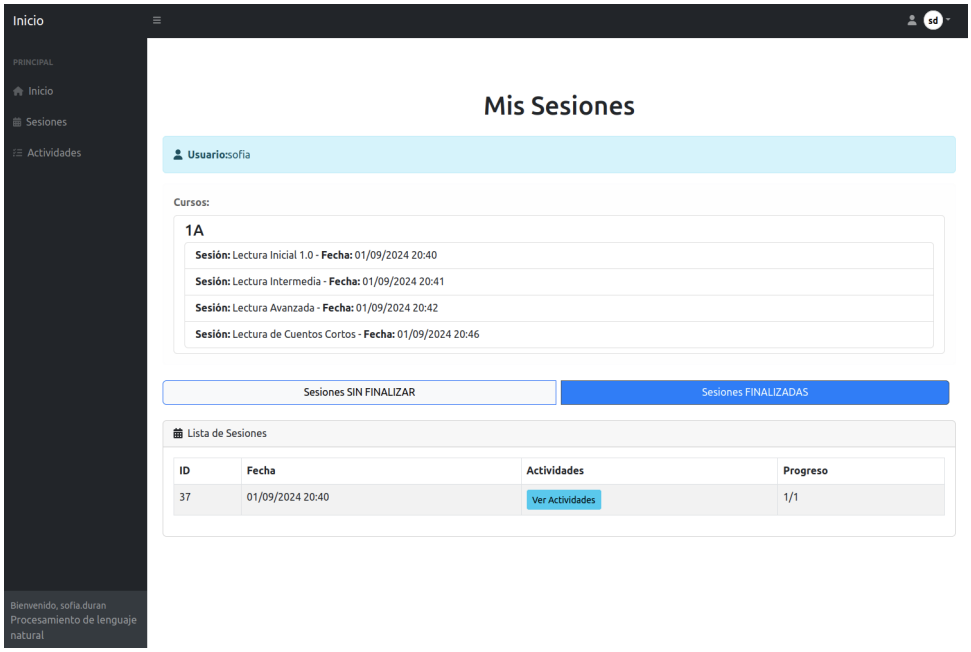


Figura 16: Sesiones finalizadas.

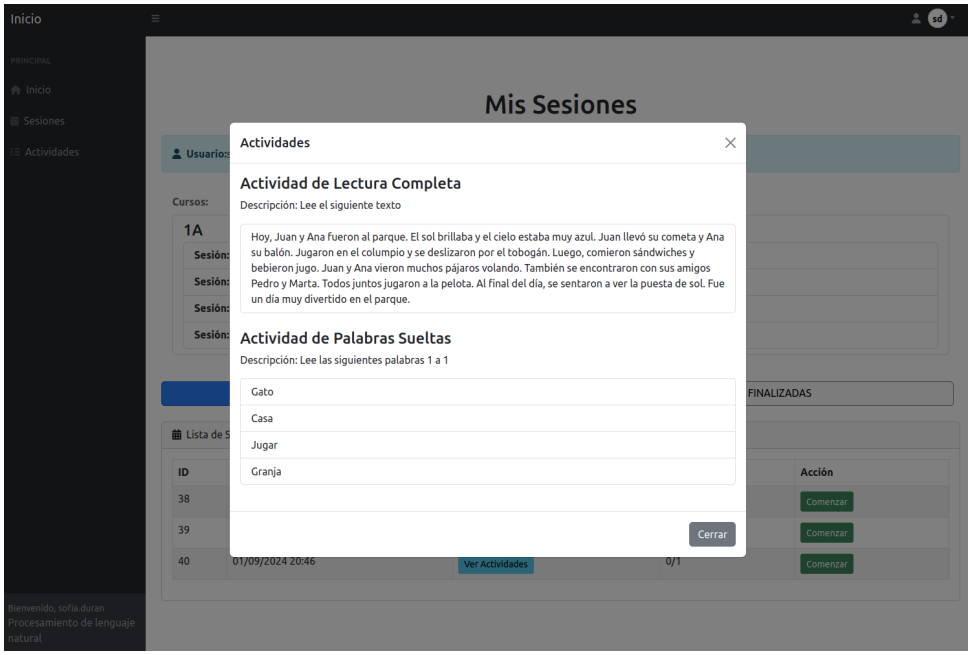


Figura 17: Vista de las actividades e ítems de una sesión de entrenamiento.

Para las sesiones no finalizadas, los usuarios tienen la opción de comenzar las actividades. En las sesiones de tipo *texto completo*, se presenta un texto que debe ser grabado y leído, como se ilustra en la Figura 18. Por otro lado, en las sesiones de *palabras sueltas*, se muestra una serie de palabras que deben ser leídas una a una, como se muestra en la Figura 20. Durante el proceso de grabación, al accionar el botón de grabación, el sistema comienza a registrar el audio, como se aprecia en la Figura 19. Finalmente, para sesiones con *palabras sueltas*, la interfaz permite grabar una palabra a la vez, como se muestra en la Figura 20.

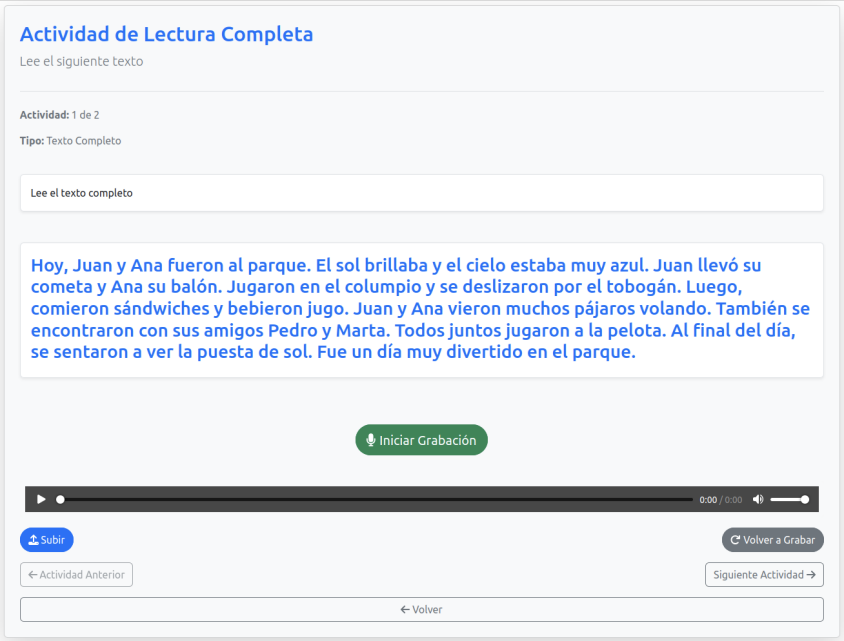


Figura 18: Vista de grabar entrenamiento con texto.

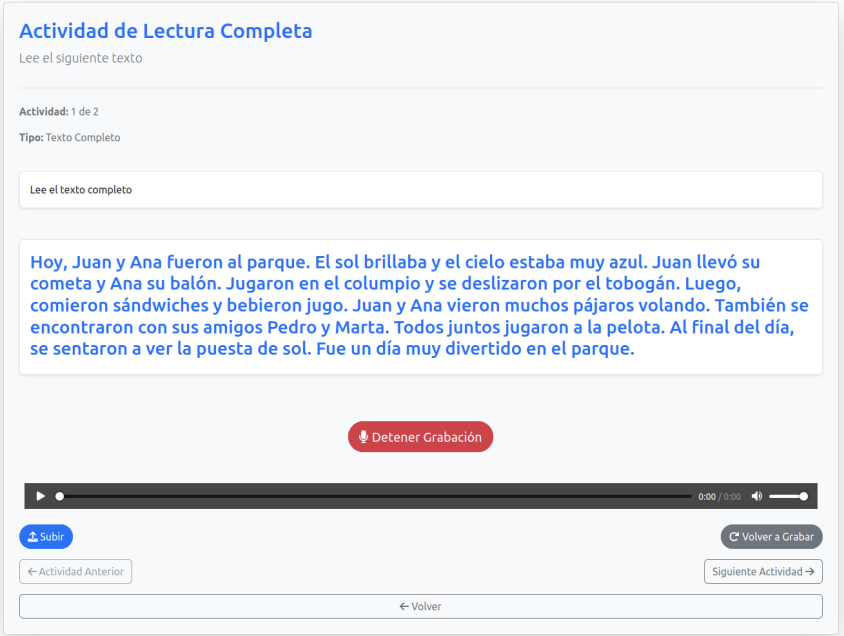


Figura 19: Botón de grabación accionado.

Cada vez que se procede a subir un archivo de audio, el sistema solicita confirmación para asegurar que el usuario desea realizar esa acción, ya que una vez que se ha subido el archivo,

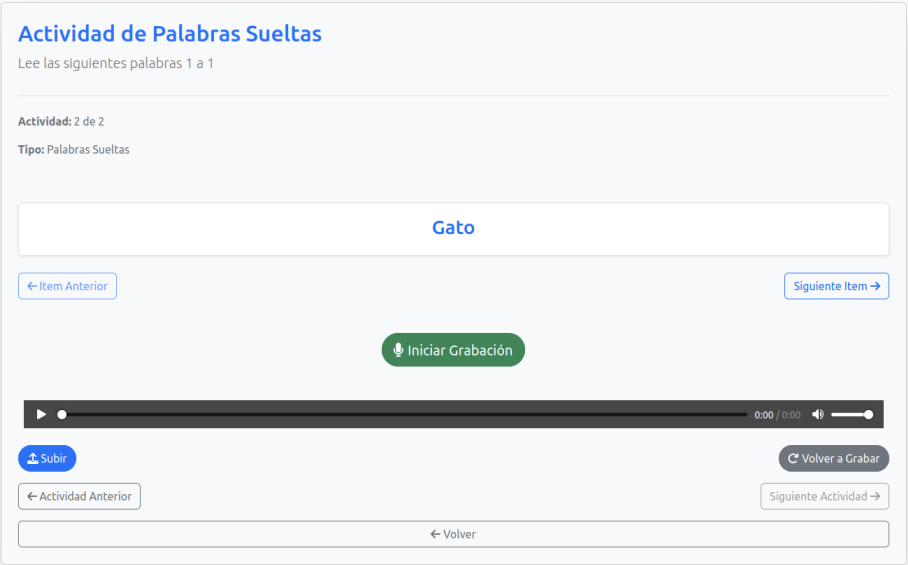


Figura 20: Vista de grabar entrenamiento con palabras sueltas.

no se puede modificar. Además, el usuario tiene la opción de abandonar el entrenamiento en cualquier momento.

5 Entrenador

Los entrenadores tienen acceso a funcionalidades avanzadas que les permiten gestionar de manera efectiva las sesiones de entrenamiento, actividades, cursos, ítems y tareas. Estas funcionalidades incluyen:

- **Gestión de sesiones de entrenamiento**: Permite al entrenador crear, modificar y eliminar sesiones de entrenamiento, así como asignar cursos y actividades a cada sesión.
- **Gestión de actividades**: facilita al entrenador la creación, edición y eliminación de actividades dentro de los cursos.
- **Gestión de cursos**: permite al entrenador crear, editar y eliminar cursos, así como asignar y retirar usuarios de los mismos.
- **Gestión de ítems**: permite al entrenador crear ítems y vincularlos con la actividad correspondiente a la que pertenecen.
- **Gestión de resultados**: permite al entrenador crear y eliminar tareas que el sistema descomprime al ser subidas. Además, facilita el análisis de archivos de audio tanto importados como almacenados en el sistema.

5.1 Gestión de sesiones de entrenamiento

El entrenador tiene la capacidad de crear nuevas sesiones de entrenamiento, asignar cursos y actividades a estas sesiones, y gestionar los detalles específicos de cada una. En la Figura 21 se muestra la vista de esta gestión, y en la Figura 22 un ejemplo de las sesines que puede haber.

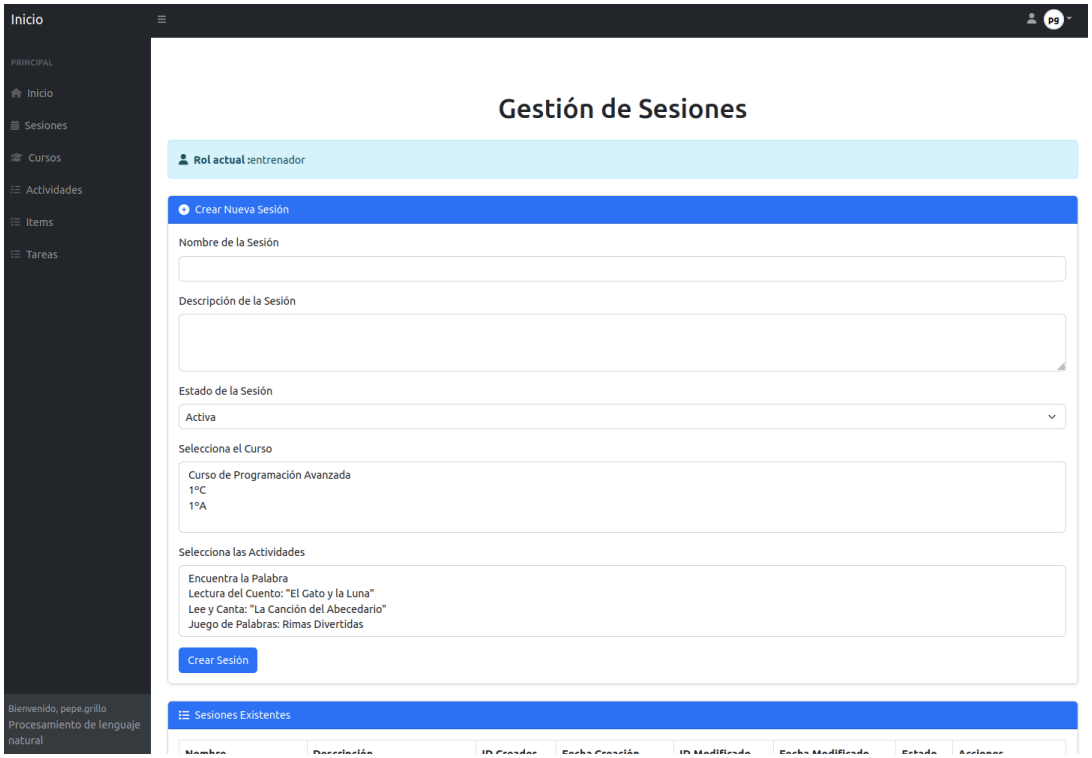


Figura 21: Vista de gestión de sesiones de entrenamiento.

Sesiones Existentes

Nombre	Descripción	ID Creador	Fecha Creación	ID Modificado	Fecha Modificado	Estado	Acciones
Nombre de la sesión	*Descripción de la sesió*	*1*	*02/09/2024 10:54*	*1*	*02/09/2024 14:11*	*activo*	Editar Borrar
Prueba	*Prueba de sesión de lectura*	*1*	*02/09/2024 12:55*	*No disponible*	*02/09/2024 12:55*	*activo*	Editar Borrar

Figura 22: Tabla de sesiones de entrenamiento.

Si en la vista de esta gestión se seleciona editar en una de las sesiones se abre un modal con los campos a editar, esto se ve en la Figura 23

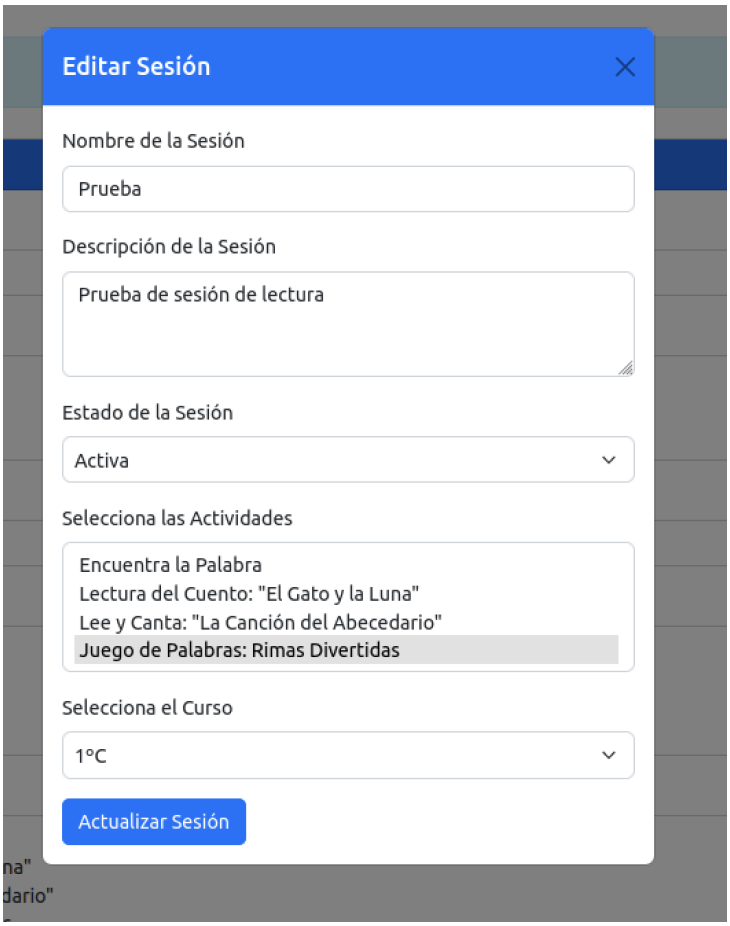


Figura 23: Modal de edición de sesión de entrenamiento.

5.2 Gestión de actividades

El entrenador puede crear nuevas actividades y seleccionar el tipo de cada una: *palabras sueltas* o *texto completo*. Además, puede eliminar y editar actividades existentes. Las actividades del tipo *palabras sueltas* solo pueden contener ítems de palabras individuales, mientras que las actividades de tipo *texto completo* pueden tener únicamente un ítem que contenga un texto. Las vistas de esta gestión se muestran en la Figura 24, y en la Figura 25.

Figura 24: Vista de gestión de actividades.

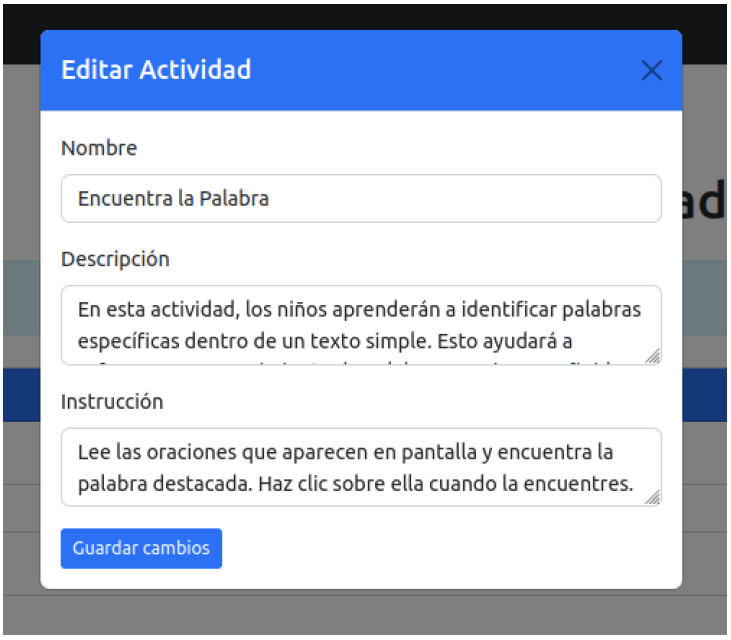


Figura 25: Modal de edición de actividad.

5.3 Gestión de cursos

El entrenador puede crear nuevos cursos y asignarles los usuarios que formarán parte de cada curso. Durante la edición de los cursos, se permite agregar o eliminar usuarios según sea necesario. En la tabla de cursos se muestran los nombres de los usuarios inscritos en cada curso, así como las sesiones asignadas a estos. Si no se muestra ninguna información en el apartado de sesiones, significa que aún no hay sesiones asignadas. Las vistas de esta gestión se muestran en la Figura 26, en la Figura 27 y en la Figura 28.

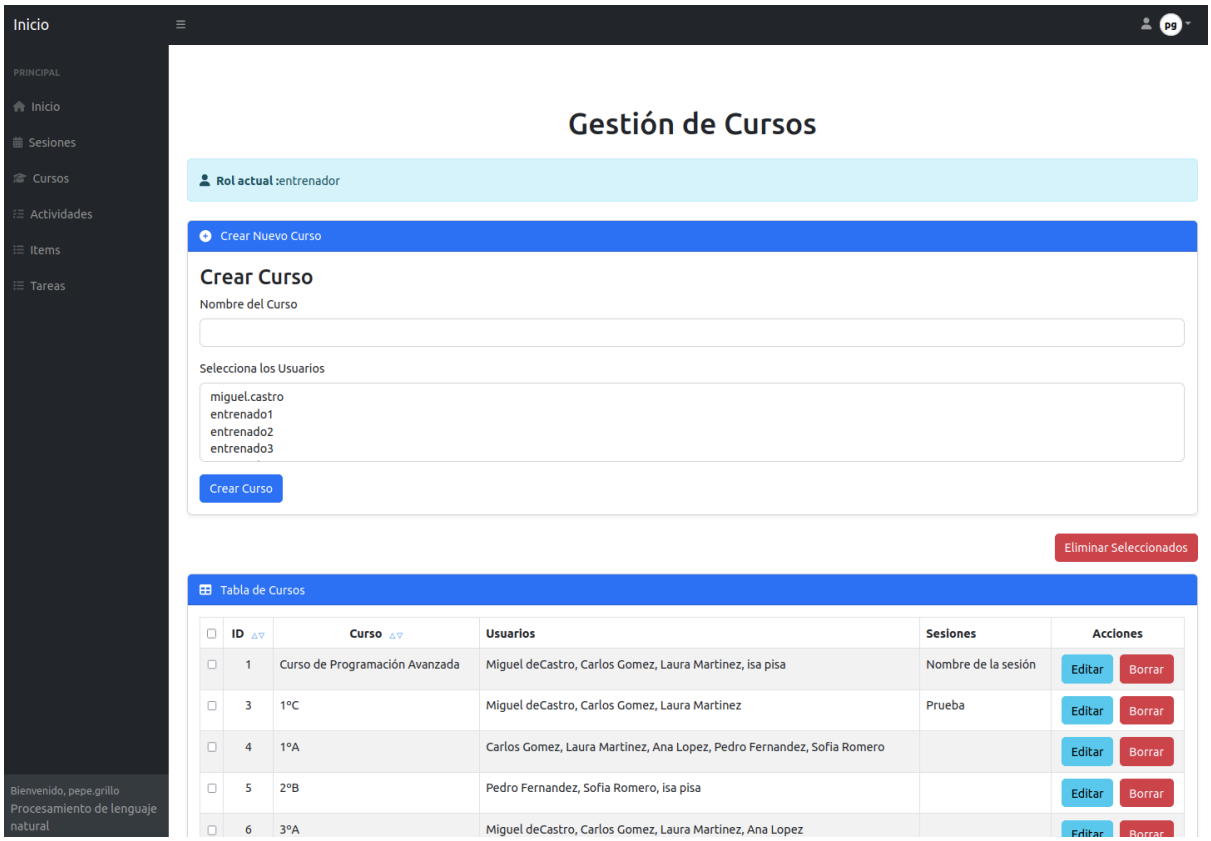


Figura 26: Vista de gestión de cursos.

Eliminar Seleccionados

Tabla de Cursos

	ID	Curso	Usuarios	Sesiones	Acciones
	1	Curso de Programación Avanzada	Miguel deCastro, Carlos Gomez, Laura Martinez, isa pisa	Nombre de la sesión	Editar Borrar
	6	3ºA	Miguel deCastro, Carlos Gomez, Laura Martinez, Ana Lopez		Editar Borrar
	5	2ºB	Pedro Fernandez, Sofia Romero, isa pisa		Editar Borrar
	3	1ºC	Miguel deCastro, Carlos Gomez, Laura Martinez	Prueba	Editar Borrar
	7	1ºB	Carlos Gomez, Laura Martinez, Ana Lopez		Editar Borrar

« Previous 1 2 Next »

Figura 27: Tabla de cursos con sus usuarios y sesiones.

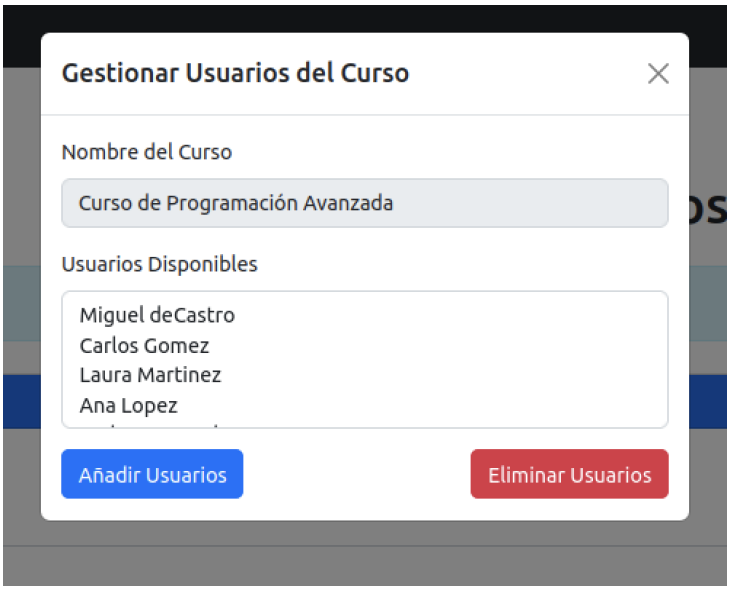


Figura 28: Modal de edición de curso.

5.4 Gestión de ítems

El entrenador puede crear nuevos ítems y asignarles la actividad correspondiente a la que pertenecen. Durante la creación, el entrenador selecciona inicialmente el tipo de ítem que desea crear: ya sea de *palabras sueltas* o de *texto completo*. Según el tipo seleccionado, el ítem se podrá asignar a diferentes actividades o tareas.

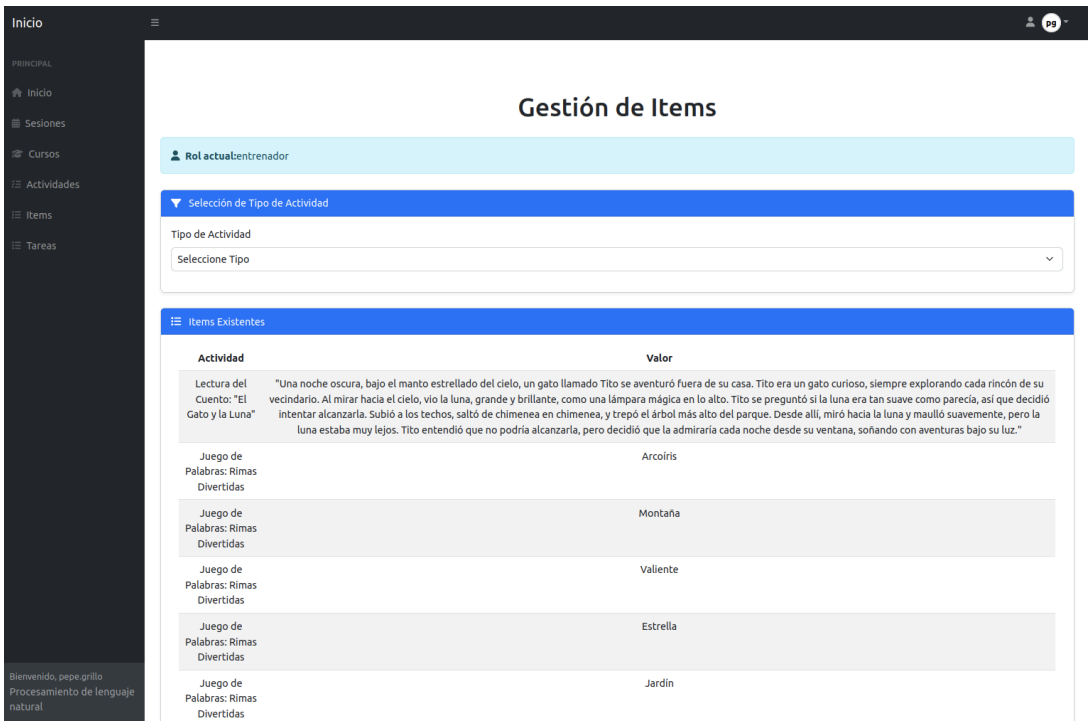


Figura 29: Vista de gestión de ítems.

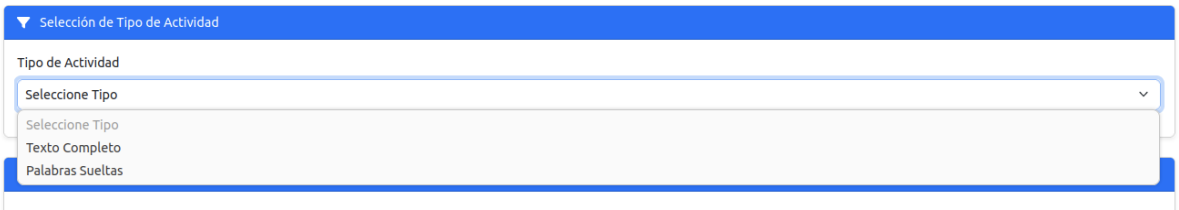


Figura 30: Primer paso en la creación de un ítem.

Figura 31: Segundo paso en la creación de ítems.

Items Existentes

Actividad	Valor
Lectura del Cuento: "El Gato y la Luna"	"Una noche oscura, bajo el manto estrellado del cielo, un gato llamado Tito se aventuró fuera de su casa. Tito era un gato curioso, siempre explorando cada rincón de su vecindario. Al mirar hacia el cielo, vio la luna, grande y brillante, como una lámpara mágica en lo alto. Tito se preguntó si la luna era tan suave como parecía, así que decidió intentar alcanzarla. Subió a los techos, saltó de chimenea en chimenea, y trepó el árbol más alto del parque. Desde allí, miró hacia la luna y maulló suavemente, pero la luna estaba muy lejos. Tito entendió que no podría alcanzarla, pero decidió que la admiraría cada noche desde su ventana, soñando con aventuras bajo su luz."
Juego de Palabras: Rimas Divertidas	Arcoíris
Juego de Palabras: Rimas Divertidas	Montaña
Juego de Palabras: Rimas Divertidas	Valiente
Juego de Palabras: Rimas Divertidas	Estrella
Juego de Palabras: Rimas Divertidas	Jardín
Lee y Canta: "La Canción del Abecedario"	"Una mañana soleada, Ana y Luis decidieron adentrarse en el bosque cercano a su casa. Armados con mochilas llenas de bocadillos y una brújula, comenzaron su aventura. El bosque era un lugar misterioso, lleno de altos árboles que susurraban con el viento. Mientras caminaban, encontraron un arroyo cristalino que reflejaba la luz del sol como un espejo. Ana, curiosa como siempre, propuso seguir el arroyo para ver a dónde los llevaba. Después de un rato, llegaron a un claro donde un ciervo los observaba desde lejos. Luis, emocionado, sacó su cámara para tomar una foto, pero el ciervo desapareció entre los árboles. La aventura continuó hasta que encontraron una cueva. ¿Entrarían? Ana tenía miedo, pero Luis la convenció de que juntos serían valientes. Dentro de la cueva, descubrieron murales antiguos pintados en las paredes, contando historias de tiempos pasados. La cueva era un tesoro de misterios esperando ser descubiertos."
Encuentra la Palabra	"Había una vez en un pequeño pueblo llamado Valledelrío, un río al que los habitantes llamaban 'El Río Encantado'. Cada vez que alguien se acercaba al río, podía escuchar una melodía suave que parecía venir del agua. Los aldeanos estaban convencidos de que el río tenía poderes mágicos y que la música era un canto de bienvenida de los espíritus del agua. Un día, una niña llamada Clara, conocida por su valentía y curiosidad, decidió investigar el misterio del río. Se preparó con una linterna, una cuerda y un diario para tomar notas. Al llegar al río, Clara se detuvo a escuchar la melodía. Notó que la música cambiaba de tono cuando se acercaba más al agua. Siguiendo el sonido, Clara descubrió una pequeña cueva oculta detrás de una cascada. Decidió entrar y, para su sorpresa, encontró un grupo de ranas cantando en coro, acompañadas por el sonido del agua que caía. Las ranas le explicaron que eran las guardianas del río y que su música tenía el poder de calmar a cualquiera que la escuchara. Clara, emocionada por su descubrimiento, prometió mantener el secreto y volvió al pueblo con una sonrisa, sabiendo que había resuelto el misterio del Río Encantado."

Figura 32: Tabla de ítems con su valor correspondiente.

5.5 Gestión de resultados

El entrenador tiene la capacidad de subir nuevas tareas al sistema y eliminarlas si es necesario. Al seleccionar una tarea, se le otorga permiso para analizar los audios asociados, ejecutándolos con la tarea correspondiente. Los audios a analizar pueden ser tanto los importados por el entrenador como aquellos seleccionados de la lista de audios registrados en el sistema. Una vez finalizado el análisis de un audio, se muestra en pantalla el resultado obtenido, como se ilustra en la Figura 33.

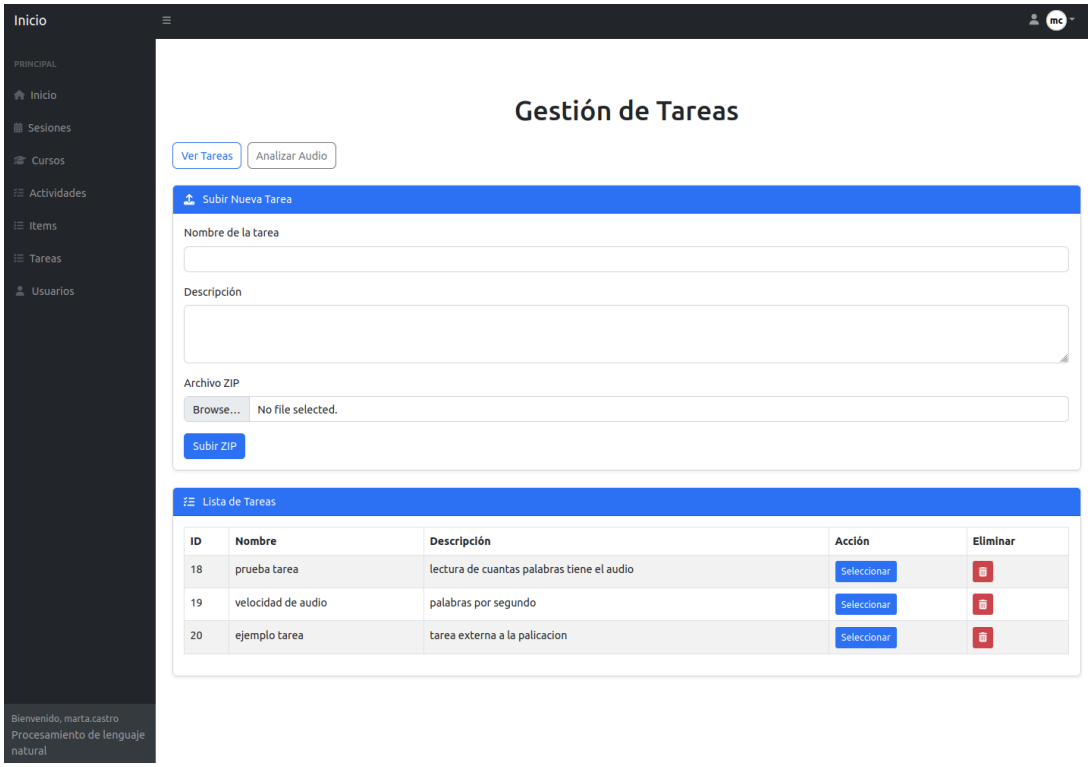


Figura 33: Vista de gestión de resultados en el apartado de subir tareas.

En la Figura 33, se muestra la vista de gestión de resultados correspondiente al proceso de subir tareas. Tras seleccionar una tarea, se tiene acceso al análisis de audio asociado, como se observa en la Figura 34. En esta etapa, se realiza el análisis del audio vinculado a la tarea.

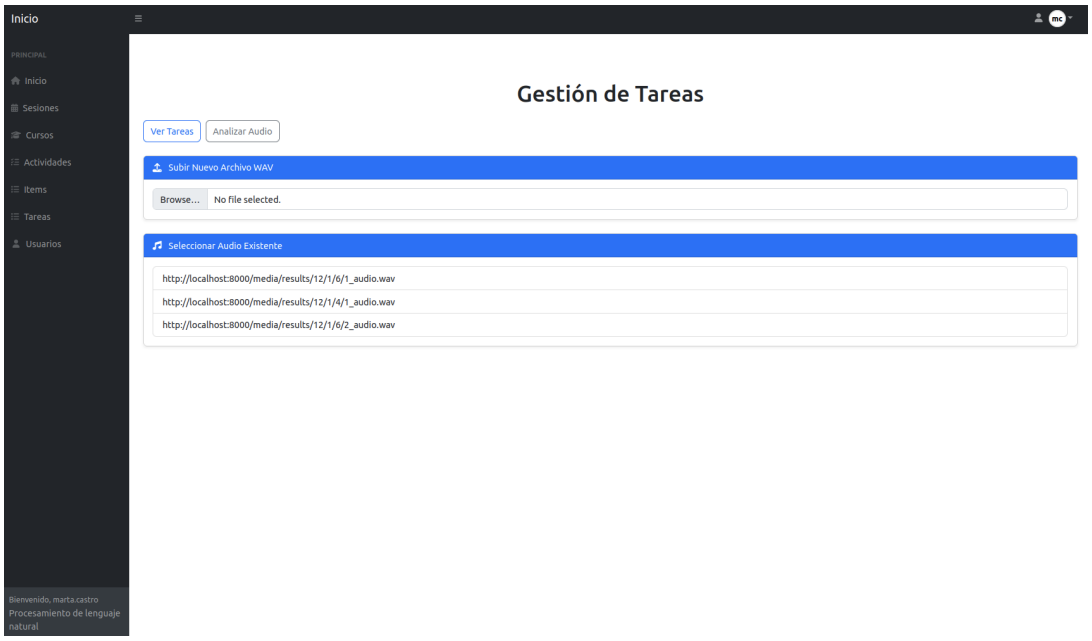


Figura 34: Acceso al análisis de audio tras seleccionar una tarea.

Los resultados del análisis de los audios se visualizan de manera detallada. En la Figura 35, se presentan ejemplos de resultados obtenidos correctamente tras el análisis del audio. Por otro lado, en la Figura 36, se muestran ejemplos de resultados donde se detectaron errores o problemas durante el análisis del audio.

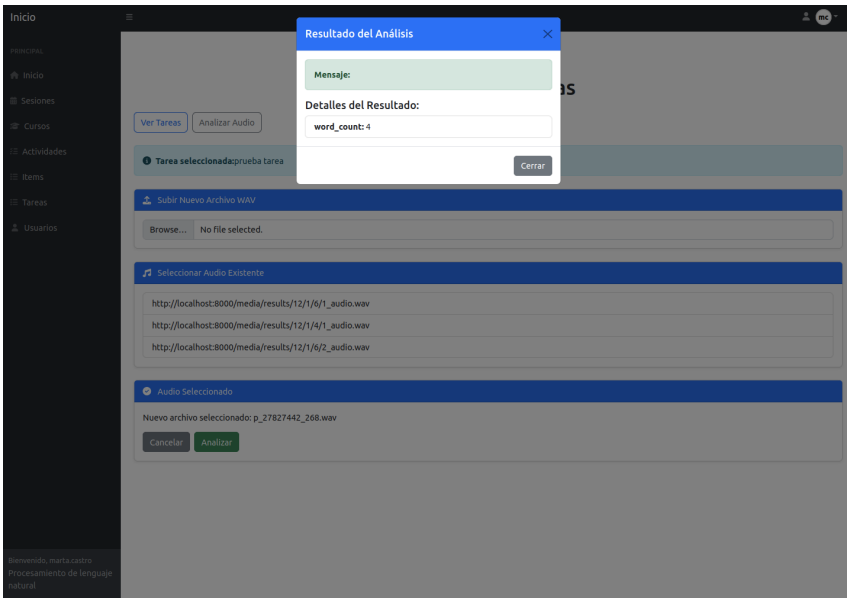


Figura 35: Ejemplos de resultado de análisis de audio correctamente.

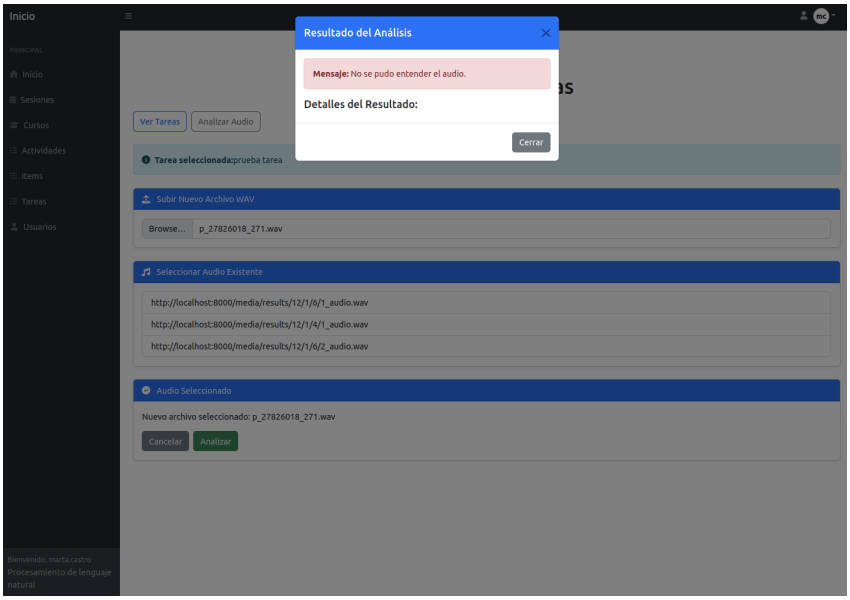


Figura 36: Ejemplos de resultado de análisis de audio con errores.

6 Administrador

El rol de administrador posee todas las capacidades del entrenador, además de funciones adicionales, como:

- **Gestión de usuarios**: creación, modificación y eliminación de cuentas de usuario, así como la asignación de roles y permisos.

6.1 Gestión de usuarios

El administrador tiene la capacidad de gestionar cuentas de usuario mediante la creación, modificación y eliminación de dichas cuentas se muestra en la Figura 37 la vista de esta gestión.

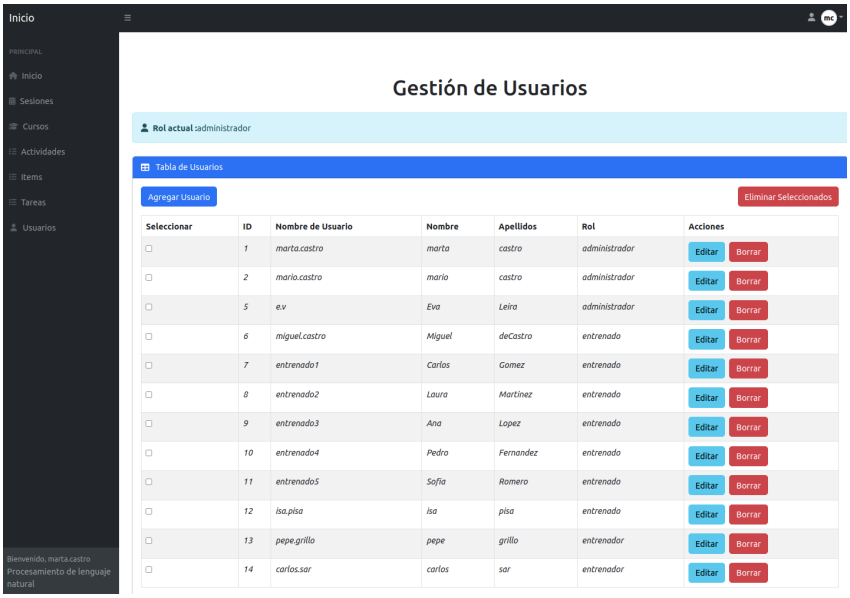


Figura 37: Vista de gestión de usuarios por el administrador.

En la Figura 38 se muestra el formulario para dar de alta a un usuario y hay q cubrir todos los datos para poder crearlo, como se muestre en la Figura 39.

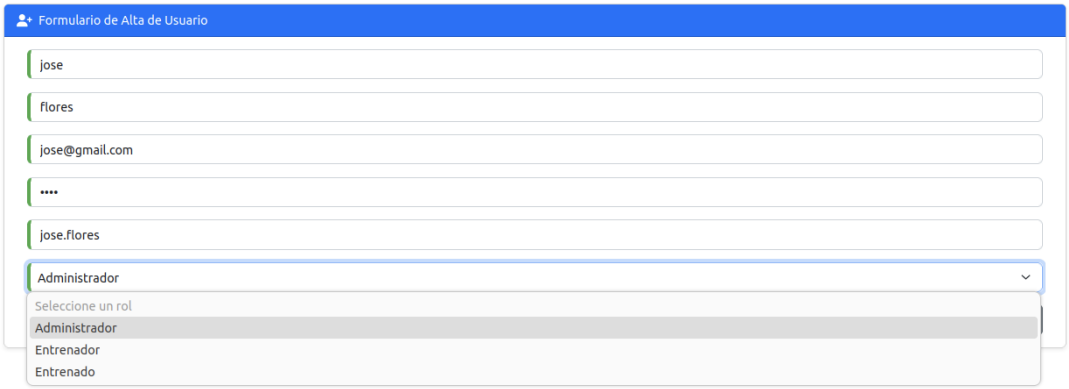


Figura 38: Vista de creación de usuarios.

Figura 39: Creación de usuarios requiere la introducción de todos los datos necesarios.