

UNIVERSIDAD NACIONAL DE EDUCACIÓN  
A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
INFORMÁTICA

MÁSTER UNIVERSITARIO EN INGENIERÍA  
INFORMÁTICA - CURSO 2023/24

*Título del Trabajo Final de Máster*

***Aplicación de análisis de las  
interacciones y modelado del  
estudiante***

Autor: Juan Manuel Sánchez Villarejo

Tutores: Antonio Rodríguez Anaya

Manuel Luque Gallego

Febrero 2024



# Agradecimientos

*A mis tutores Antonio Rodríguez Anaya y Manuel Luque Gallego por ayudarme a elegir el tema del trabajo y a buscar una solución cuando no conseguía avanzar. Por ayudarme a elegir el contenido para la memoria y solventar todas las dudas sobre la teoría y la organización de la misma.  
Muchas gracias.*



# Dedicatoria

*Me gustaría dedicar este trabajo a David por ser un gran apoyo durante todos estos años y por estar siempre ahí. Este logro es también tuyo. Gracias.*



# Resumen

Las plataformas de aprendizaje online son a día de hoy, herramientas imprescindibles en cualquier actividad educativa. Estas plataformas ofrecen, entre otros servicios, el de foros, que son utilizados para que los estudiantes se comuniquen con el equipo docente o entre sí.

El objetivo del proyecto es continuar con el desarrollo de una aplicación que permita incorporar los datos de los foros asociados a los cursos virtuales de la UNED, gestionar y manipular dichos datos desde la propia aplicación y generar información de salida que facilite al responsable de la asignatura el proceso de análisis de la interacciones y modelado del estudiante, y así poder establecer procesos de mejora.

El proyecto amplía la funcionalidad y profundiza en los métodos de minería de datos orientados al entorno educativo y en la integración de la información generada por la aplicación con las herramientas más avanzadas actualmente en materia de gestión, así como representación de gráficos. Se realiza un proceso de modelado del estudiante, sobre todo de su componente social, con la finalidad de ofrecer información útil para la gestión de actividades de aprendizaje colaborativo.

Las tecnologías empleadas para llevar a cabo el proyecto han sido entre otras Spring Tool Suite, el cual ha permitido crear toda la aplicación a través del lenguaje de programación Java, y utilizándose tecnologías y lenguajes como HTML, PhpMyAdmin, Spring JPA y Thymeleaf para la creación de la aplicación y para la gestión de bases de datos. Por último, se ha usado una base de datos MariaDB y tecnología de contenedores Docker para encapsular toda la aplicación y facilitar el despliegue y ejecución de la misma.

## Palabras clave

OpenWebinars, Gephi, ALF, Spring Tool Suite, MariaDB, Maven, Análisis de redes sociales, OpenMarkov, Agora, Evaluación entre pares





# Abstract

Online learning platforms are essential tools in today's educational activities. These platforms offer various services, including forums, which are used for students to communicate with the teaching team or among themselves.

The project's objective is to continue developing an application that allows the incorporation of data from forums associated with virtual courses at UNED, manage and manipulate this data within the application, and generate output information to facilitate the subject coordinator's process of analyzing interactions and modeling student behavior. This aims to establish improvement processes.

The project expands functionality and deepens into data mining methods tailored to the educational environment, integrating information generated by the application with cutting-edge tools in graph management and representation. A student modeling process, particularly focusing on the social component, is conducted to provide useful information for managing collaborative learning activities.

Technologies employed for the project include, among others, Spring Tool Suite, enabling the creation of the entire application using the Java programming language. Technologies and languages such as HTML, PhpMyAdmin, Spring JPA, and Thymeleaf are utilized for application development and database management. Finally, a MariaDB database and Docker container technology are employed to encapsulate the entire application, simplifying deployment and execution.

## Key words

OpenWebinars, Gephi, ALF, Spring Tool Suite, MariaDB, Maven, Social Network Analysis, OpenMarkov, Agora, Peer Assessment.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Ágora y aLF . . . . .	1
1.2. Los foros en plataformas de enseñanza online . . . . .	3
1.3. Análisis de redes sociales . . . . .	4
1.4. Evaluación entre pares . . . . .	5
1.5. Objetivos del proyecto . . . . .	5
1.6. Estructura de la memoria . . . . .	7
<b>2. Especificaciones y gestión del proyecto</b>	<b>9</b>
2.1. Metodología de trabajo . . . . .	9
2.2. Fases del trabajo . . . . .	10
2.3. Especificaciones del trabajo . . . . .	11
2.4. Cronograma . . . . .	12
2.5. Herramientas . . . . .	13
2.6. Repositorio del código y recursos . . . . .	18
<b>3. Estado de la cuestión</b>	<b>19</b>
3.1. Aplicación base . . . . .	19
3.1.1. Inicio . . . . .	19
3.1.2. Contacto . . . . .	20
3.1.3. Gestión de estudios . . . . .	20
3.1.4. Importar datos . . . . .	21
3.1.5. Generar grafo . . . . .	22
3.1.6. Conclusión . . . . .	23
<b>4. Análisis del sistema</b>	<b>25</b>
4.1. Alcance del sistema . . . . .	25
4.2. Clases del proyecto . . . . .	26
4.2.1. Modelo Vista Controlador (MVC) . . . . .	26
4.2.2. Módulos, clases y diagramas . . . . .	27
4.3. Casos de uso . . . . .	36

4.4.	Descripción de la interfaz . . . . .	39
4.4.1.	Diagrama de flujo . . . . .	40
4.4.2.	Estructura de las pantallas . . . . .	41
<b>5.</b>	<b>Diseño del sistema</b>	<b>45</b>
5.1.	Diseño de la base de datos . . . . .	45
5.2.	Diseño de la interfaz . . . . .	52
5.2.1.	Gestión de Tareas de Revisor . . . . .	52
5.2.2.	Generar grafo . . . . .	56
5.2.3.	Revisión . . . . .	64
<b>6.</b>	<b>Conclusiones y trabajos futuros</b>	<b>73</b>
6.1.	Conclusiones . . . . .	73
6.2.	Trabajos futuros . . . . .	74
<b>A.</b>	<b>Instalación y acceso a la aplicación</b>	<b>81</b>

# Índice de figuras

1.1. Ejemplo de plataforma de la asignatura de máster UNED. . . . .	3
1.2. Esquema general del trabajo del que se parte. . . . .	6
2.1. Diagrama de metodología agile. . . . .	9
2.2. Diagrama del Modelo en cascada. . . . .	10
2.3. Cronograma del proyecto final de máster. . . . .	13
2.4. Logo de la plataforma Open Webinars. . . . .	13
2.5. Logo de la plataforma GitHub. . . . .	14
2.6. Logo del IDE Spring Tool. . . . .	15
2.7. Logo del motor de plantillas Thymeleaf. . . . .	15
2.8. Logo de la herramienta PhpMyAdmin. . . . .	16
2.9. Logo del gestor de bases de datos MariaDB. . . . .	16
2.10. Logo de la herramienta Gephi. . . . .	17
2.11. Logo del Integrated Development Environment (IDE) Eclipse. . . . .	17
2.12. Logo de la plataforma Overleaf. . . . .	17
3.1. Aplicación base: Inicio. . . . .	20
3.2. Aplicación base: Asignaturas. . . . .	20
3.3. Aplicación base: Convocatorias. . . . .	21
3.4. Aplicación base: Personas. . . . .	21
3.5. Aplicación base: Importar datos. . . . .	22
3.6. Aplicación base: Generar grafo. . . . .	22
3.7. Aplicación base: Esquema general del proyecto. . . . .	23
4.1. Aplicación desarrollada para el análisis de interacciones y mo- delado del estudiante. . . . .	25
4.2. Nueva opción de Gestión de tareas de revisor. . . . .	26
4.3. Esquema de un modelo Modelo-Vista-Controlador MVC. . . . .	27
4.4. Paquetes o módulos creados para la aplicación. . . . .	28
4.5. Diagrama de clases. . . . .	29
4.6. Diagrama UML de la clase TareaRevisor. . . . .	33
4.7. Ejemplo de template para diseño web. . . . .	34

4.8. Código para añadir dependencia de la API de Gephi. . . . .	36
4.9. Configuración del fichero application.properties. . . . .	36
4.10. Casos de uso de la aplicación. . . . .	37
4.11. Página de inicio de la aplicación. . . . .	39
4.12. Diagrama de flujo de la aplicación. . . . .	40
4.13. Diseño de la página de Tareas de Revisión. . . . .	41
4.14. Diseño de la página de Generar Grafo. . . . .	42
4.15. Diseño de la página de Revisión. . . . .	43
5.1. Tabla Convocatoria de la base de datos. . . . .	45
5.2. Tabla Asignatura de la base de datos. . . . .	45
5.3. Tabla Materia de la base de datos. . . . .	46
5.4. Tabla Matrícula de la base de datos. . . . .	46
5.5. Tabla Foro de la base de datos. . . . .	47
5.6. Tabla Mensaje de la base de datos. . . . .	47
5.7. Tabla Persona de la base de datos. . . . .	48
5.8. Tabla Tablas de la base de datos. . . . .	49
5.9. Tabla Revisión de la base de datos. . . . .	51
5.10. Acceso a Gestión de tareas de revisor. . . . .	52
5.11. Listado de tareas de revisor. . . . .	53
5.12. Formulario para crear una nueva tarea de revisor. . . . .	53
5.13. Formulario para asignar orden a los indicadores. . . . .	54
5.14. Tabla de probabilidad. . . . .	55
5.15. Tabla de utilidades de <i>Activity Compatibility</i> . . . . .	56
5.16. Matriz de adyacencia del foro. . . . .	57
5.17. Mensajes intercambiados entre los nodos 242 y 241. . . . .	57
5.18. Grafo generado con los datos de un foro por la aplicación. . . . .	59
5.19. Tabla de indicadores del grafo. . . . .	61
5.20. Tabla con los resultados de Gephi. . . . .	62
5.21. Gráfica de Cercanía. . . . .	62
5.22. Gráfica de Intermediación. . . . .	63
5.23. Gráfica de Cercanía de Vector Propio. . . . .	63
5.24. Gráfica de Modularidad. . . . .	64
5.25. Esquema de diseño del método de asignación de revisores. . . . .	65
5.26. Listado de tareas de revisor. . . . .	65
5.27. Selección de una tarea de revisor. . . . .	66
5.28. Selección del número de matrices de n revisores. . . . .	67
5.29. Matriz de utilidades aleatoria de 10 estudiantes. . . . .	68
5.30. Matriz de N revisores. . . . .	68
5.31. Matriz de 7 revisores. . . . .	69
5.32. Matriz Producto. . . . .	70

5.33. Matriz Producto de 7 revisores. . . . .	71
5.34. Valor Utilidad Total que muestra la aplicación. . . . .	72
A.1. Git LFS. . . . .	81
A.2. Contenedor con la aplicación y la base de datos corriendo. . .	82
A.3. Ejemplo de la aplicación desde navegador. . . . .	83





# Índice de tablas

5.1. Valores por defecto de la tabla Probabilidad. . . . .	50
5.2. Valores por defecto de la tabla Utilidad. . . . .	50
5.3. Ejemplo de tarea de 1 revisor. . . . .	66
5.4. Ejemplo de tarea de 2 revisores. . . . .	67
5.5. Matriz aleatoria. . . . .	70
5.6. ejemplo de matriz de 1 revisor. . . . .	70
5.7. Ejemplo de resultado de matriz producto. . . . .	70
5.8. Ejemplo de resultado de matriz producto para cálculo de Utilidad Total. . . . .	71



# Siglas

**ID** Influence Diagram. 64

**IDE** Integrated Development Environment. 14, 17, 27, 58, 73, XI

**LMS** Learning Management System. 1

**MVC** Model-View-Controller. 26, 28

**OPA** Online Peer Assessment. 5, 7, 12, 38, 50, 64, 66, 73

**SNA** Social Network Analysis. 4, 6, 10, 12, 22, 26, 38, 50, 58, 60, 64

**SQL** Structured Query Language. 15, 16, 48

**UML** Unified Modeling Language. 32

**UNED** Universidad Nacional de Educación a Distancia. 1, 2, 46, 47



# Capítulo 1

## Introducción

La Universidad Nacional de Educación a Distancia (UNED) es una institución educativa, especializada en la educación a distancia y la enseñanza online. La (UNED, 2024c) tiene como objetivo principal, ofrecer oportunidades educativas a personas que, por diversas razones, no pueden asistir a clases presenciales en una universidad convencional.

La UNED utiliza métodos de enseñanza a distancia, como materiales impresos, recursos multimedia y plataformas virtuales, para facilitar el aprendizaje de los estudiantes que pueden acceder a los contenidos desde cualquier lugar.

La flexibilidad en el horario de estudio y la posibilidad de aprender a distancia han convertido a la UNED en una opción popular para los más de 150.000 alumnos (UNED, 2024a) que buscan continuar su educación mientras enfrentan limitaciones geográficas o de tiempo.

Ágora (UNED, 2022), es la plataforma de gestión del aprendizaje basada en el software de código abierto Open LMS, que a su vez se basa en Moodle, y que permite un aprendizaje activo, adaptativo y personalizado.

### 1.1. Ágora y aLF

La UNED, lanzó el pasado curso académico 2022-2023 (UNED, 2022) una nueva plataforma virtual llamada Ágora, la cual está siendo implantada de forma paulatina en las diferentes comunidades virtuales para atender a estudiantes, docentes y profesores-tutores. Está basada en la tecnología de código abierto Moodle, y va a reemplazar a la plataforma aLF que hasta ahora se venía usando, respondiendo así a la demanda creciente de la comunidad universitaria y apostando por la innovación tecnológica en su metodología online y semipresencial.

Las características de un curso virtual en la UNED en las que se han llevado a cabo todas las asignaturas de este máster universitario pueden variar según la naturaleza de cada curso o asignatura. Sin embargo, en términos generales, los cursos virtuales en la UNED suelen incorporar las siguientes características:

- **Plataforma online:** Los cursos virtuales en la UNED se imparten a través de una plataforma educativa en línea. En este caso, la nueva plataforma Ágora descrita anteriormente.
- **Contenido multimedia:** Los cursos incluyen material multimedia como documentos, presentaciones, vídeos y recursos interactivos normalmente suministrados por los docentes, que respaldan el aprendizaje a distancia.
- **Interacción:** Dado que la UNED se especializa en educación a distancia, los cursos virtuales a menudo incluyen actividades y discusiones que permiten la participación de los estudiantes en momentos que se ajusten a sus horarios, es decir, de manera asíncrona.
- **Evaluación en línea:** Los estudiantes pueden completar evaluaciones y exámenes en línea a través de la plataforma. Esto puede incluir cuestionarios, tareas, proyectos y exámenes.
- **Foros de discusión:** Se fomenta la participación y la interacción entre estudiantes y profesores a través de foros de discusión en línea. Estos permiten el intercambio de ideas y la resolución de dudas.
- **Seguimiento del progreso:** La plataforma suele ofrecer herramientas para que los estudiantes y profesores puedan realizar un seguimiento del progreso académico. Esto puede incluir informes de actividad y/o calificaciones.
- **Soporte técnico y pedagógico:** La UNED proporciona soporte técnico y pedagógico para ayudar a los estudiantes a utilizar la plataforma y abordar cualquier pregunta relacionada con el contenido del curso.
- **Experiencia móvil:** Las plataformas se diseñan para ser accesibles desde dispositivos móviles, facilitando a los estudiantes acceder al contenido y participar en el aprendizaje desde cualquier lugar.

En la figura 1.1, se muestra una captura de parte de la plataforma de una asignatura de Trabajo de Fin de Máster en Ingeniería Informática de la Universidad Nacional de Educación a Distancia (UNED) donde se

muestran la evidencia de algunas de las características anteriormente mencionadas.



Figura 1.1: Ejemplo de plataforma de la asignatura de máster UNED.

## 1.2. Los foros en plataformas de enseñanza online

Los foros en plataformas de enseñanza virtual desempeñan un papel importante en la educación enriqueciendo la experiencia del estudiantes y de los docentes. Algunos de los puntos clave que ofrecen los foros para ambas partes y que resultan de interés para este trabajo son los siguientes:

1. **Interacción y colaboración:** Los foros permiten la interacción y colaboración entre estudiantes y profesores. Los participantes pueden discutir temas, compartir ideas, hacer preguntas y responderlas, promoviendo un ambiente de aprendizaje colaborativo.
2. **Aclaración de dudas:** Los estudiantes pueden plantear preguntas en los foros cuando tengan dudas sobre el contenido del curso. Los profesores y otros estudiantes pueden proporcionar respuestas, lo que contribuye a la resolución colectiva de problemas y al intercambio de conocimientos.
3. **Debate y reflexión:** Los foros ofrecen un espacio para el debate y la reflexión sobre temas específicos del curso. Los estudiantes pueden expresar sus opiniones, debatir ideas y profundizar en la comprensión de los conceptos a través de la discusión.
4. **Archivo de discusiones:** Los foros sirven como un archivo de discusiones que los estudiantes pueden revisar en cualquier momento. Esto puede

ser útil para repasar conceptos, repasar discusiones anteriores y consolidar el aprendizaje.

5. Fomento de la participación: Los foros pueden utilizarse como parte de la evaluación de la participación, incentivando a los estudiantes a involucrarse activamente en las discusiones y contribuir al aprendizaje colectivo.

En resumen, los foros en plataformas de enseñanza ofrecen un espacio interactivo y colaborativo que mejora la comunicación, la participación y la construcción de conocimiento en el entorno educativo en línea.

### 1.3. Análisis de redes sociales

El análisis de redes sociales (por sus siglas en inglés SNA) es una metodología que estudia las interconexiones entre individuos (Sinha, 2014). Utiliza técnicas y herramientas para analizar la estructura de una red (Scott, 2017) y ha sido aplicada en varios campos (Aggarwal, 2011; Wasserman and Faust, 1994). El objetivo principal es comprender la dinámica de las interacciones sociales y cómo la información fluye a través de la red, proporcionando información valiosa sobre la estructura social y los vínculos entre los actores o nodos en una red dada.

El SNA es utilizado en el ámbito educativo (De-Marcos et al., 2016) y es aplicable en los foros de plataformas de enseñanza online (Dawson, 2008), donde la estructura de la red se construye mediante un grafo, donde los nodos son los participantes (estudiantes, profesores, tutores, etc.) y las aristas que conectan los nodos indican relaciones o interacciones entre los participantes.

Diferentes investigaciones ya han utilizado SNA en entorno educativos:

- La relación específica entre las métricas y el comportamiento de los estudiantes para que los profesores puedan monitorizar a los estudiantes en el curso en línea. (Saqr et al., 2018)
- Propuesta de un conjunto de métricas de SNA para caracterizar a individuos y redes en su totalidad (Saxena et al., 2019).
- En el contexto educativo, los foros son el principal medio para conectar a los participantes y son la principal fuente para construir las redes para realizar el SNA (Saqr et al., 2020).
- La relación entre los resultados de los estudiantes y sus métricas de SNA ha sido demostrada (Saqr et al., 2020) (Liu et al., 2018).



## 1.4. Evaluación entre pares

La evaluación entre pares en la enseñanza online OPA es una estrategia pedagógica en la que los estudiantes evalúan el trabajo de sus compañeros. Esta práctica ofrece varias ventajas y desafíos, y puede implementarse de diversas maneras. Algunos de los puntos clave de OPA, son las ventajas (Uto and Ueno, 2016) que ofrecen en cuanto a capacitación para el estudiante:

1. Responsabilidad
2. Autonomía
3. Autorreagulación

Sin embargo, las tareas de OPA encuentran algunas dificultades (Hanrahan and Isaacs, 2001) o desafíos como pueden ser la falta de compromiso y participación por parte de los alumnos (Meek et al., 2016). La implementación efectiva de la evaluación entre pares en la enseñanza en línea requiere cuidado en el diseño del proceso y una atención continua para abordar desafíos potenciales.

El trabajo sobre *Asignación automática de revisores en una tarea de evaluación basado en las interacciones sociales de los alumnos* (Anaya et al., 2019) es la base para el desarrollo de este proyecto porque es la guía para poner en práctica todo esto que se ha desarrollado en estos primeros apartados, es decir, lograr una asignación automática para una evaluación entre pares, basados en el análisis de redes sociales que son los foros de plataformas de enseñanza virtual.

## 1.5. Objetivos del proyecto

El presente proyecto plantea una serie de objetivos, donde el principal de ellos trata del desarrollo de una aplicación que permita al usuario, en este caso un docente, obtener lo que se ha mencionado en la sección anterior 1.4, es decir, una **asignación automática para una evaluación entre pares**, basados en el **análisis de redes sociales** que son los foros de plataformas de enseñanza virtual.

Dicho de otra forma, el docente conseguirá que la aplicación le diga qué estudiante/s deberá/n evaluar el trabajo de otro alumno, todo ello basado en las interacciones que hay en los foros.

El presente proyecto, parte de un trabajo previo que realizó un alumno de la UNED (Martín, 2022) para su proyecto fin de grado, en el que desarrolló una aplicación que facilitaba la labor de los profesores para procesar

y analizar la información generada en los diferentes estudios y asignaturas que impartían, así como ayudar a la hora de identificar acciones de mejora sobre dichos estudios. Dicha aplicación será mostrada más en profundidad en el capítulo 4.



Figura 1.2: Esquema general del trabajo del que se parte.

Utilizando los mensajes intercambiados por los alumnos y docentes en los foros de las asignaturas (en la figura 1.2 la figura identificada con azul) , esta aplicación (en la figura 1.2 la figura identificada con naranja) , generaba un documento en formato gml (Gephi, 2024b) (en la figura 1.2 la figura identificada con verde) que el docente podría utilizar en aplicaciones avanzadas de análisis de grafos.

Además del objetivo principal de crear tal aplicación, el proyecto tiene como objetivos los que se definen a continuación:

1. Análisis del trabajo previo.
  - a) Leer el documento del proyecto final de grado y analizar la estructura y código fuente creado.
  - b) La forma en que se organizó la información de los foros en el anterior proyecto y cómo se implementó la teoría de grafos.
2. Implementación de Social Network Analysis (SNA).
  - a) Conocer las principales métricas de un grafo.
  - b) Investigar librerías que puedan ser integradas para creación de grafos y diagramas de influencia sobre los datos de los foros.
  - c) Investigar librerías que puedan ser integradas para la extracción de métricas de grafos.
  - d) Presentar al usuario el resultado del SNA como grafos o gráficas.
3. Implementar asignación de evaluación entre pares.
  - a) Crear en la aplicación un apartado para el usuario, en este caso el docente, para introducir criterios como experto para las tareas de revisión.

- b)* Mostrar resultados de asignación aleatoria de revisores para la evaluación entre pares.
  - c)* Generar nueva información para la Online Peer Assessment (OPA) que pueda ser integrada por las herramientas más avanzadas actualmente en materia de gestión y representación de modelos de gráficos probabilistas. Como por ejemplo OpenMarkov (Arias and Díez, 2008; Arias et al., 2019).
- 4. Diseñar de la aplicación usable.
  - a)* Conservar el diseño de una arquitectura modular (Deitel, 2017) y extensible que cubra las diferentes fases del proceso (procesado, análisis y presentación de información) y permita la ampliación futura de funcionalidades.
  - b)* Crear una aplicación que sea ejecutable en cualquier ordenador.
- 5. Elaborar un documento para el proyecto siguiendo las directrices que marca la UNED (UNED, 2024b).

## 1.6. Estructura de la memoria

Además de esta sección introductoria, en el presente documento podrán encontrarse los siguientes capítulos:

En el Capítulo 2 Se habla de la metodología y especificaciones seguidas, así como la gestión de recursos y tareas aplicadas.

En el Capítulo 3 Se trata del estado de la cuestión o estado del arte, donde se detallan los apartados de la aplicación desarrollada de la que se parte para este trabajo.

En el Capítulo 4 Se especifica el análisis y diseño de la aplicación desarrollada.

En el Capítulo 5 Se habla del conocimiento adquirido durante el proceso de todo el proyecto.

En el Capítulo 6 Se detallan las conclusiones y desarrollos aplicables en futuros trabajos.

En el Anexo 1 A Se describen los pasos para la instalación y uso de la aplicación.



# Capítulo 2

## Especificaciones y gestión del proyecto

### 2.1. Metodología de trabajo

Para la consecución de los objetivos descritos en el capítulo 1, se han llevado a cabo una serie de procedimientos y procesos que se describen a continuación:

- Definición de los objetivos y alcance del proyecto. A través de reuniones frecuentes mantenidas con lo tutores, se han ido describiendo las herramientas que se usarían para alcanzar los objetivos marcados y qué requisitos y funcionalidades que debería cumplir el trabajo. Metodología *agile*.

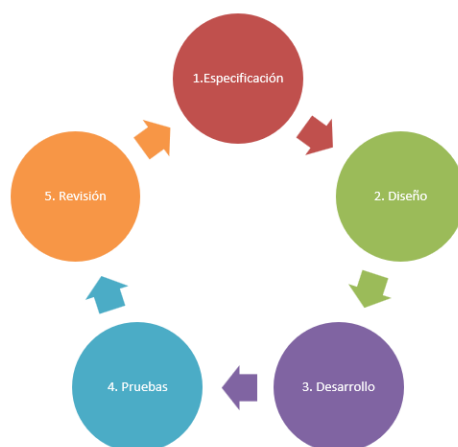


Figura 2.1: Diagrama de metodología agile.

Estas sesiones han sido coordinadas mediante la plataforma Microsoft Teams con el propósito de interactuar con los tutores con respecto a las últimas especificaciones y metas del proyecto. Durante estas reuniones, se discuten los progresos logrados hasta la fecha, así como las inquietudes o dudas que hayan surgido. Asimismo, la herramienta Microsoft SharePoint ha sido empleada como un repositorio para la compartición de documentación, ya sea para la revisión de documentos científicos o para la creación de material como el cuaderno de bitácora.

- Análisis documental de la literatura existente relacionada con el tema elegido. En su mayoría aportado por los tutores.
- Estudio preliminar de las herramientas descritas en el Capítulo 1. Este material es aportado por las propias empresas distribuidoras de software como es Spring Boot (Spring, 2024a), Gephi (Gephi, 2024a) o Thymeleaf (Thymeleaf, 2024) o por la comunidad de usuarios que los utilizan.
- Anotación de los trabajos realizados diariamente en un cuaderno de bitácora que permita ver el avance, problemas y soluciones encontradas a lo largo del proyecto.
- Implementación del software respetando cuestiones éticas como son el uso de librerías de Gephi para el SNA o el gestor de dependencias Maven (Maven, 2024).
- Redacción de la memoria en la que se plasma el trabajo realizado, con los resultados y conclusiones extraídas. Para la redacción de la memoria se ha utilizado la herramienta Overleaf (Overleaf, 2024).

## 2.2. Fases del trabajo

En esta sección se trata de explicar cuales han sido las principales fases para llevar a cabo este proyecto.

El modelo en cascada ha sido el modelo elegido para la elaboración.



Figura 2.2: Diagrama del Modelo en cascada.

En el siguiente apartado 2.3 se describen los requisitos o especificaciones del sistema.

En el capítulo 4 se trata el Análisis del sistema.

En el capítulo 5 se trata el Diseño del Sistema.

La implementación ha sido el desarrollo de la aplicación cuyo código fuente será posible obtener en los detalles aportados en el A.

Algunas de las pruebas realizadas al software son descritas en el capítulo 5, donde se comprueban los nuevos requisitos implementados para este proyecto y donde se han llevado a cabo mediante pruebas unitarias.

## 2.3. Especificaciones del trabajo

El proyecto desarrollado parte de unas especificaciones funcionales y técnicas hechas por otro alumno. Parte de estas especificaciones se siguen manteniendo como base para este proyecto y son la que se indican a continuación:

- **Gestión de asignaturas.** El sistema dispone de un módulo para el alta, baja y modificación de las diferentes asignaturas.
- **Gestión de convocatorias.** El sistema dispone de un módulo para el alta, baja y modificación de las diferentes convocatorias comprometidas por la universidad.
- **Gestión integral de personas.** El sistema dispone de un módulo para el alta, baja y modificación de la información de personas (alumnos y profesores).
- **Gestión integral de materias.** El sistema dispone de un módulo para el alta, baja y modificación de las diferentes materias gestionadas por un profesor.
- **Gestión integral de matrículas.** El sistema dispone de un módulo para el alta, baja y modificación de las diferentes matrículas a las que está suscrito un alumno.
- **Incorporación de datos al sistema.** Funcionalidad que permite seleccionar un fichero con los datos de los foros de una asignatura y una determinada convocatoria, y los incorpora a la base de datos del sistema. Los datos son repartidos y alojados en las diferentes tablas del modelo entidad relación.

## 12 CAPÍTULO 2. ESPECIFICACIONES Y GESTIÓN DEL PROYECTO

Otras especificaciones han sido tomadas como base del trabajo anteriormente realizado, pero modificadas para ser adaptadas al nuevo diseño del proyecto y son las siguientes:

- **Menús de navegación y hojas de estilos.** El sistema dispone de un menú de navegación que dirige al usuario a las funcionalidades requeridas, con un estilo uniforme y sencillo.
- **Exportación de información y generación de fichero de grafo.** Además de la información que aportaba el trabajo previo (número de nodos de un grafo, aristas de un grafo y un fichero con extensión gml de grafo), se exporta nueva información resultado de SNA y de los criterios de los expertos para la OPA.
- **Integración de SNA.** La novedad en este proyecto es que se utilizan librerías externas de Gephi para evitar así tener que utilizar tal aplicación de terceros para obtener resultados.
- **Sistema auto contenido.** La aplicación se desarrolla bajo paradigma de contenedores. Esto significa que para instalar la aplicación y acceder a sus funcionalidades, únicamente se requiere la ejecución de un script que realiza todo el trabajo. No se requiere la instalación de ningún software o motor de bases de datos adicional.

Las nuevas especificaciones desarrolladas en este proyecto son las siguientes:

- **Gestión de tareas de revisión.** El sistema dispone de un módulo para el alta, baja y modificación de tareas de revisión entre pares.
- **Gestión de criterios de expertos.** El experto, en este caso el profesor que controle la aplicación, será el encargado de generar información de entrada al programa para obtener resultados óptimos.

### 2.4. Cronograma

En la figura que se muestra a continuación, se muestra la tabla o cronograma con las principales actividades que se han realizado para el desarrollo de este trabajo:



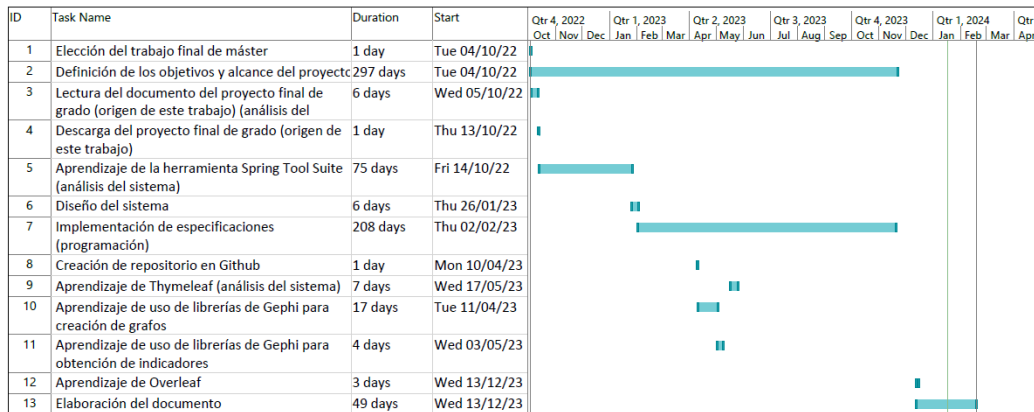


Figura 2.3: Cronograma del proyecto final de máster.

## 2.5. Herramientas

A continuación se describe brevemente las principales herramientas utilizadas para llevar a cabo todo el proyecto:

- **Open Webinars**

### DEFINICIÓN

OpenWebinars (OpenWebinars, 2024) es una plataforma de formación en línea que ofrece cursos y webinars relacionados con tecnología, desarrollo de software, programación y otros temas relacionados con la informática y la tecnología de la información.

### FUNCIONALIDAD

Plataforma utilizada para aprendizaje de Spring Boot y Spring MVC.

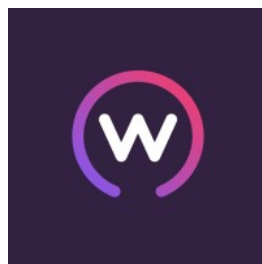


Figura 2.4: Logo de la plataforma Open Webinars.

- **Github**

## 14 CAPÍTULO 2. ESPECIFICACIONES Y GESTIÓN DEL PROYECTO

### DEFINICIÓN

GitHub (GitHub, 2024) es una plataforma en línea que utiliza Git para el control de versiones. Facilita la colaboración de desarrolladores en proyectos de software, permite el seguimiento de cambios en el código, gestión de problemas, proyectos y fusiones de código. Es ampliamente utilizado en la comunidad de desarrollo de software.

### FUNCIONALIDAD

Plataforma utilizada como repositorio para almacenamiento, como medio para llevar un control de versiones y como medio para compartir código con los tutores.

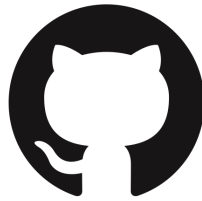


Figura 2.5: Logo de la plataforma GitHub.

### ■ **Spring Tool**

#### DEFINICIÓN

Spring Tool (Spring, 2024b) es un entorno de desarrollo integrado (IDE) basado en Eclipse que está diseñado específicamente para el desarrollo de aplicaciones con el framework Spring. Spring es un marco de desarrollo de aplicaciones Java que simplifica y agiliza el desarrollo al proporcionar una infraestructura completa para construir aplicaciones empresariales.

#### FUNCIONALIDAD

IDE utilizado para el desarrollo de la aplicación en lenguajes Java para las partes de control del software y en html para la parte de visualización de la misma.



Figura 2.6: Logo del IDE Spring Tool.

- **Thymeleaf**

DEFINICIÓN

Thymeleaf (Thymeleaf, 2024) es un motor de plantillas para el desarrollo de aplicaciones web en Java. Se utiliza para crear y renderizar vistas en el lado del servidor. Thymeleaf es especialmente popular en entornos de desarrollo web que utilizan el framework Spring.

FUNCIONALIDAD

Estas plantillas se utilizan para la creación de la parte visual (html) de nuestra aplicación integradas en Spring Tool.



Figura 2.7: Logo del motor de plantillas Thymeleaf.

- **PhPMyAdmin**

DEFINICIÓN

PhpMyAdmin (MySQL, 2024a) es una herramienta de administración de bases de datos MySQL (MySQL, 2024b) basada en web. Permite a los usuarios gestionar sus bases de datos MySQL a través de una interfaz gráfica de usuario, en lugar de tener que interactuar directamente con comandos Structured Query Language (SQL) en la línea de comandos.

FUNCIONALIDAD

## 16 CAPÍTULO 2. ESPECIFICACIONES Y GESTIÓN DEL PROYECTO

Esta herramienta se utiliza para la visualización, edición, borrado y creación de tablas y registros de estas mediante sentencias SQL.



Figura 2.8: Logo de la herramienta PhpMyAdmin.

### ■ MariaDB

#### DEFINICIÓN

MariaDB (Foundation, 2024) es un sistema de gestión de bases de datos relacional de código abierto, surgido como una bifurcación de MySQL. Mantiene compatibilidad con MySQL, permitiendo una migración sencilla de aplicaciones y datos entre ambas plataformas.

#### FUNCIONALIDAD

Se usa este sistema como gestor de la base de datos creada.



Figura 2.9: Logo del gestor de bases de datos MariaDB.

### ■ Gephi

#### DEFINICIÓN

Gephi (Gephi, 2024a) es una herramienta de código abierto para visualizar y analizar redes, útil en ciencia de datos y disciplinas similares. Permite explorar y comprender la estructura y relaciones en conjuntos de datos de red.

#### FUNCIONALIDAD

Esta herramienta se utiliza como base para adquirir conocimiento del funcionamiento de sus librerías, así como para contrastar resultados obtenidos.



Figura 2.10: Logo de la herramienta Gephi.

- **Eclipse**

DEFINICIÓN

Eclipse (Eclipse, noun) es un IDE de código abierto utilizado principalmente para programación en Java, aunque admite varios otros lenguajes de programación mediante el uso de complementos. Proporciona herramientas y características avanzadas para facilitar el desarrollo de software.

FUNCIONALIDAD

Esta plataforma se utiliza para la creación del Diagrama UML que se presenta en el siguiente capítulo.



Figura 2.11: Logo del IDE Eclipse.

- **Overleaf**

DEFINICIÓN

Overleaf (Overleaf, 2024) es una plataforma en línea que facilita la colaboración y la creación de documentos científicos, técnicos y académicos en  $\text{\LaTeX}$ .

FUNCIONALIDAD

Esta plataforma se utiliza para la elaboración del presente documento.



Figura 2.12: Logo de la plataforma Overleaf.

## **2.6. Repositorio del código y recursos**

Para poder ver al código fuente de la aplicación desarrollada, puede acceder al repositorio (Sánchez, 2024).

Puede descargar el proyecto mediante un cliente git de línea de comandos con la sentencia

```
git clone https://github.com/juanmanuelanvi/tfm-jsv.git
```

Para más información sobre la instalación y acceso a la aplicación, consultar A.

# Capítulo 3

## Estado de la cuestión

Previamente, se han tratado dos puntos importantes en el capítulo 1 Introducción:

1. Análisis de redes sociales en la sección 1.3.
2. Evaluación entre pares en la sección 1.4.

En ellos han sido nombrados algunos artículos y estudios que ha habido hasta la fecha de y los cuales son la base para el desarrollo de este proyecto. Estos serán descritos en los capítulos siguientes. En este capítulo, se describe la aplicación desarrollada previamente por un alumno de la UNED que sirve como base desde la que se parte para el desarrollo del presente proyecto. El título de la siguiente sección es el nombre del proyecto final de grado que este alumno realizó.

### 3.1. Aplicación base

Como podrá verse, la aplicación desarrollada para este trabajo tiene un entorno web con un menú que permite al usuario moverse por diferentes apartados y funcionalidades, a continuación, se irán describiendo cada uno de ellos.

#### 3.1.1. Inicio

Este apartado contiene información sobre la propia aplicación, describe qué es, qué opciones tiene disponibles y un enlace al repositorio donde puede descargarse todo el proyecto para su uso y recursos de utilidad.

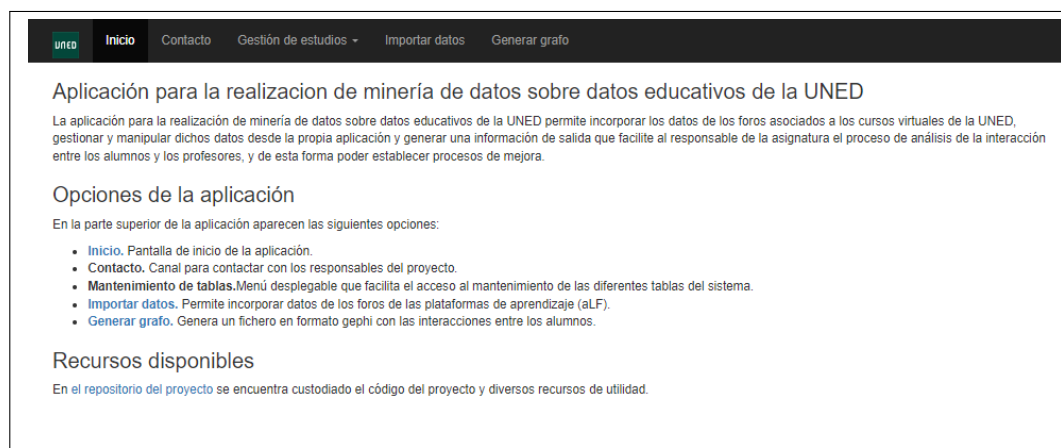


Figura 3.1: Aplicación base: Inicio.

### 3.1.2. Contacto

Este apartado no ha sido desarrollado por el alumno. Pretendía ser un formulario de contacto con el autor del trabajo. Se dejará tal cual en este proyecto.

### 3.1.3. Gestión de estudios

En este apartado se diseña para el usuario diferentes apartados para poder gestionar:

- Asignaturas: para el alta, baja y modificación de las diferentes asignaturas gestionadas.



Figura 3.2: Aplicación base: Asignaturas.



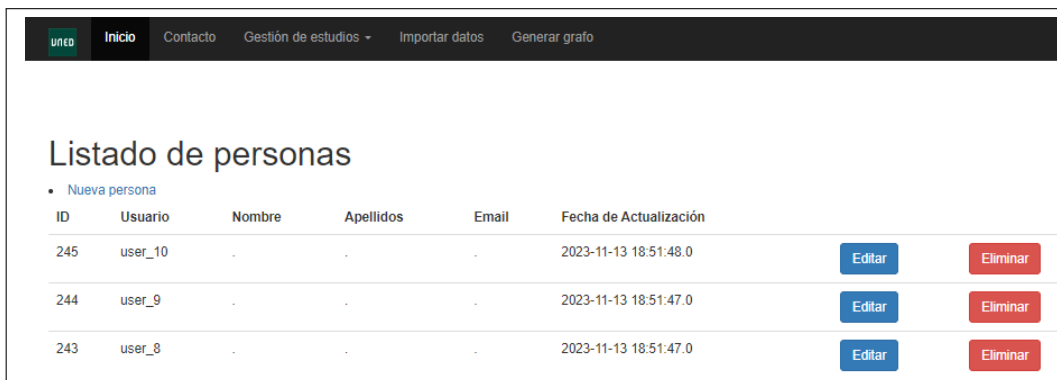
- Convocatorias: para el alta, baja y modificación de las diferentes convocatorias.



ID	Convocatoria	Fecha de Actualización		
7	JUNIO-2020	2023-10-19 18:53:02.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>

Figura 3.3: Aplicación base: Convocatorias.

- Personas: para el alta, baja y modificación de la información de personas (alumnos y profesores).



ID	Usuario	Nombre	Apellidos	Email	Fecha de Actualización		
245	user_10	.	.	.	2023-11-13 18:51:48.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>
244	user_9	.	.	.	2023-11-13 18:51:47.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>
243	user_8	.	.	.	2023-11-13 18:51:47.0	<a href="#">Editar</a>	<a href="#">Eliminar</a>

Figura 3.4: Aplicación base: Personas.

### 3.1.4. Importar datos

Dentro de las especificaciones del trabajo descritas en la sección 2.3, la incorporación de datos al sistema era una de ellas. En este punto, el alumno permite la carga de un fichero en formato csv o txt que contenga la información de interacciones entre alumnos y profesores de una determinada asignatura. Este fichero debe contener una estructura concreta para que el programa pueda agregar toda esta información de interacciones en una base de datos.

Figura 3.5: Aplicación base: Importar datos.

### 3.1.5. Generar grafo

En este apartado la aplicación ofrece información útil para el usuario. Una vez realizada la importación de datos, la aplicación la ofrece al usuario tres datos importantes:

- El número de nodos del grafo, que se corresponde con el número de personas de que dispone la base de datos.
- Número de aristas del grafo, que se corresponde con el número de mensajes totales intercambiados en el grafo.
- Documento con formato gml para su descarga. Este documento servirá como dato de entrada para realizar el SNA con una aplicación externa como es Gephi.

Proceso de generación de grafo	
El grafo se ha generado satisfactoriamente. A continuación se detallan los elementos que conforman dicho grafo, así como al posibilidad de descargarse el fichero con toda la información, para su integración en herramientas de explotación analítica.	
	<b>Valor</b>
<b>Número de nodos.</b> Los elementos que conforman el grafo, en este caso formados por los alumnos y profesores de las asignaturas	13
<b>Número de aristas.</b> Las conexiones existentes entre los diferentes nodos, y que en este caso representan los mensajes enviados desde un nodo A a un nodo B.	128
<b>Fichero gephi.</b> Archivo en formato gephi con el contenido completo del grafo.	<a href="#">Grafo dirigido</a>

Figura 3.6: Aplicación base: Generar grafo.

### 3.1.6. Conclusión

Una vez descritos de forma resumida los diferentes apartados y funcionalidades del proyecto tomado como base para el trabajo actual, conviene resaltar que tal aplicación no profundiza en minería de datos o en análisis de redes sociales, sino que prepara esta aplicación para desarrollar estas tareas en futuros proyectos.

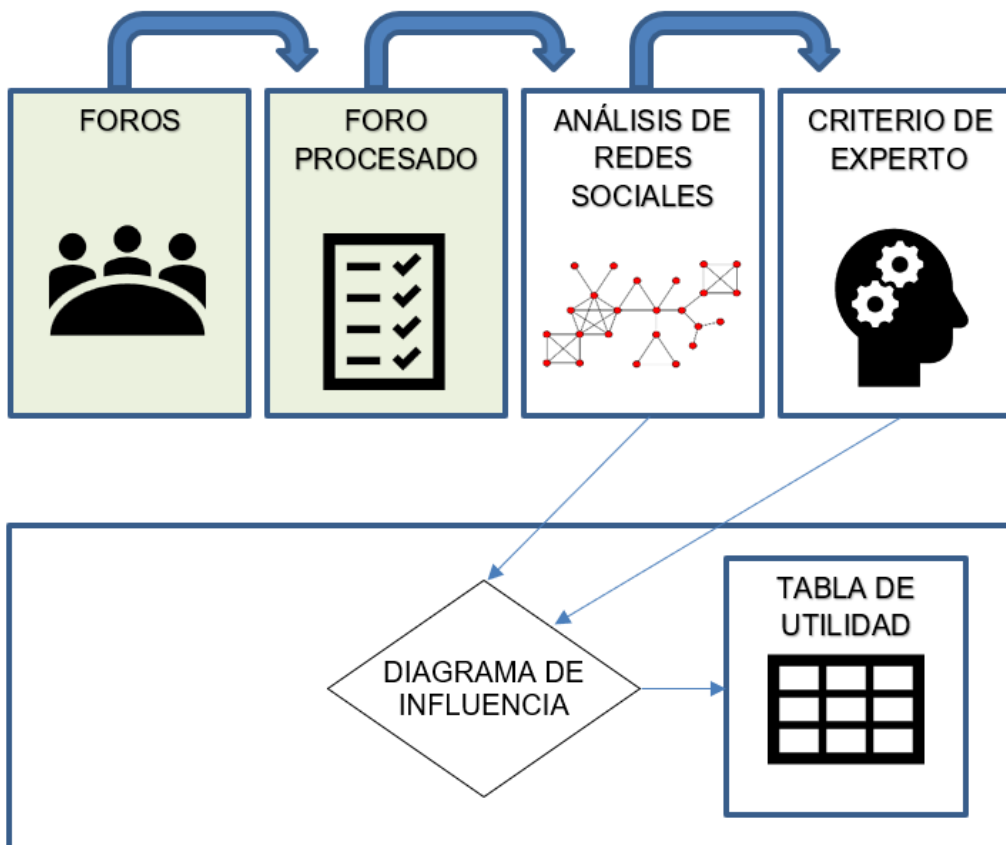


Figura 3.7: Aplicación base: Esquema general del proyecto.

En la figura 3.7 puede verse el esquema que plantea la base para el desarrollo de este proyecto. En verde, se muestran los apartados ya implementados y sobre lienzo blanco los que se pretenden realizar en este trabajo y que será detallado en profundidad en el capítulo 4.

- Foros: Se trata de la información obtenida de los foros de un curso académico.

- Foro procesado: Los datos del foro son procesados por la aplicación creada. La aplicación toma como dato de entrada la información del foro y la procesa para el posterior análisis de la misma.
- Análisis de redes sociales: La aplicación de la que se parte generaba información de salida para realizar el análisis de redes sociales con aplicaciones. El presente proyecto pretende integrar este apartado.
- Criterio de experto: Se tendrá en cuenta el criterio de los docentes para conseguir mejores resultados en la asignación de revisores.
- Diagrama de influencia (Howard and Matheson, 1984): Es una estructura de datos tipo grafo dirigido que toma como dato de entrada el análisis de redes sociales y el criterio de los expertos para obtener una tabla de utilidad.
- Tabla de utilidad: Es el resultado de mejor asignación para una tarea de evaluación entre pares.

# Capítulo 4

## Análisis del sistema

En este capítulo se explica el análisis del sistema. Una vez conocidos los requisitos que se piden y que han sido descritos en el apartado 2.3, se trata ahora el alcance del sistema, las clases implementadas, los casos de uso, la interfaz y el plan de pruebas.

### 4.1. Alcance del sistema

En este apartado se describe el alcance del sistema donde se describe brevemente qué se pretende abarcar en este proyecto.

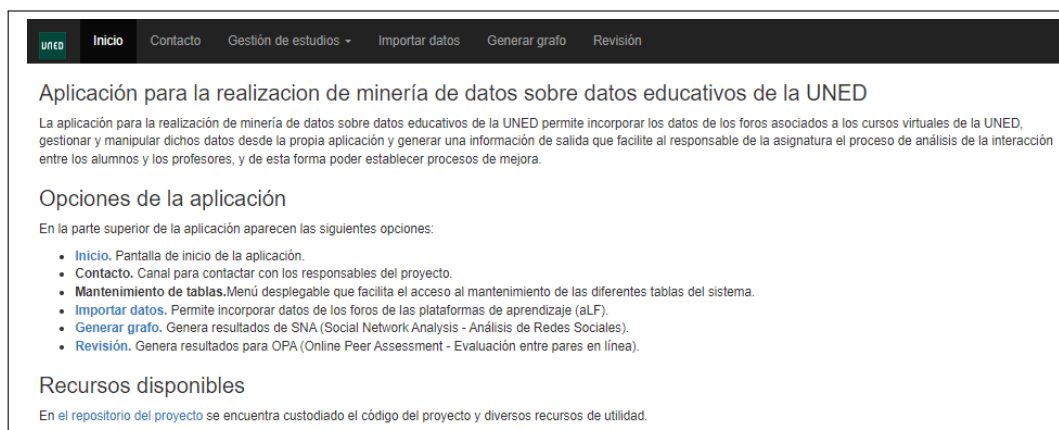


Figura 4.1: Aplicación desarrollada para el análisis de interacciones y modelado del estudiante.

Como puede observarse en la figura 4.1 de arriba, hay alguna pequeña diferencia respecto a la figura mostrada en el capítulo 3 donde se mostraba

en la figura 3.1 una página de inicio donde se aprecian principalmente los siguientes cambios y que son alcance del proyecto:

- **Gestión de tareas de revisión:** Nuevo apartado en la opción Gestión de estudios llamado Gestión de tareas de revisor.



Figura 4.2: Nueva opción de Gestión de tareas de revisor.

Desde este apartado se cumple con la especificación de *Gestión de criterios de expertos* y *Gestión de tareas de revisor*.

- **Generar grafo:** El apartado aparentemente es el mismo, pero en él se le ofrece al usuario nueva información resultado del Análisis de Redes Sociales.
- **Revisión:** Nueva página para que, una vez cargados los datos de un foro en el sistema, generado los grafos y el SNA, el sistema genere información para la asignación de revisores de una tarea.

## 4.2. Clases del proyecto

En esta sección se describe la estructura organizativa de las clases creadas en el sistema. Antes de hablar de las clases creadas, se habla de la arquitectura en la que se basa el sistema.

### 4.2.1. Modelo Vista Controlador (MVC)

El patrón de arquitectura de software conocido como Model-View-Controller (MVC) (Contributors, 2024), es el patrón que se ha seguido para la creación de esta aplicación. Se basa en separar los elementos fundamentales de una aplicación en tres componentes distintos: el Modelo, la Vista y el Controlador.

- **Modelo:** Contiene toda la lógica del negocio, representando el sistema en su totalidad. Son por tanto todos los datos contenidos en las bases de datos llevados a cabo mediante el gestor MariaDB.

- **Vista:** Representa la interfaz de usuario y todas las herramientas que el usuario utiliza para interactuar con el programa. Esta parte es la desarrollada con la ayuda del motor de plantillas Thymeleaf integrada en el IDE Spring Tool.
- **Controlador:** Responde a las interacciones (eventos) realizadas por el usuario en la Vista y realiza solicitudes al Modelo. Es la parte intermedia entre ambas partes. Esta parte se desarrolla también en el IDE Spring Tool.

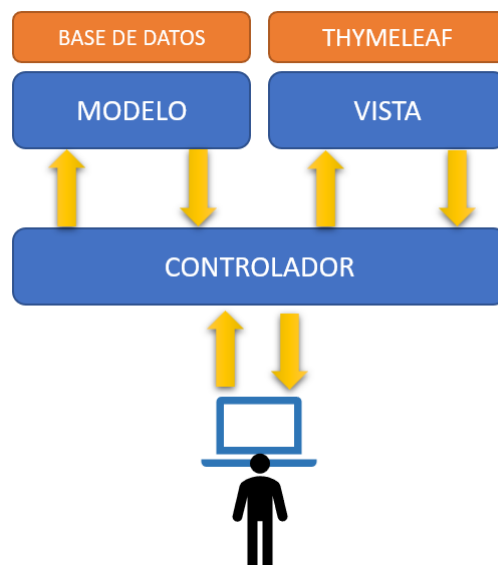


Figura 4.3: Esquema de un modelo Modelo-Vista-Controlador MVC.

#### 4.2.2. Módulos, clases y diagramas

En este apartado se describe la estructura modular (Deitel, 2017) seguida para llevar a cabo la aplicación desarrollada en el IDE de Spring Tool Suite.

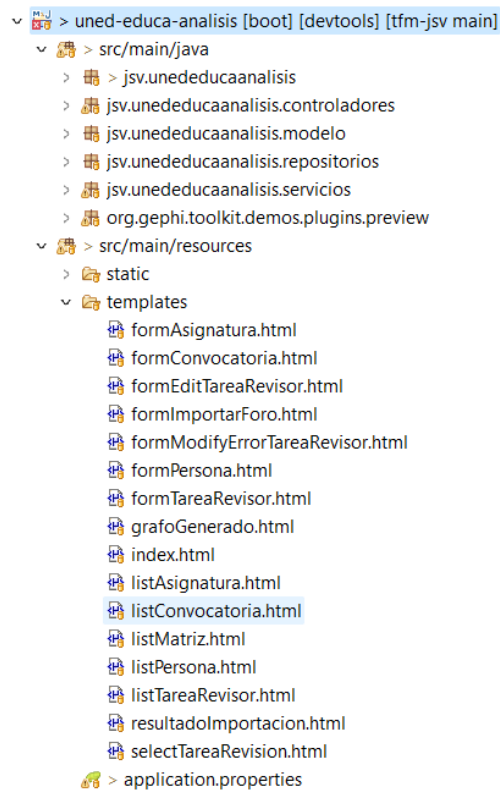


Figura 4.4: Paquetes o módulos creados para la aplicación.

En la figura 4.4 de arriba, existen 4 paquetes o módulos: controladores, modelo, repositorios y servicios. Estos 4 módulos más las plantillas de presentación de la vista son los componentes que permiten crear nuestra estructura MVC.

En la siguiente figura 4.5, se muestra un diagrama ejemplo para la obtención de una tarea de revisor diseñada, donde el usuario desde su navegador hace una llamada a la aplicación para obtener la lista de tareas de revisor que hay creadas en la aplicación (consulta):



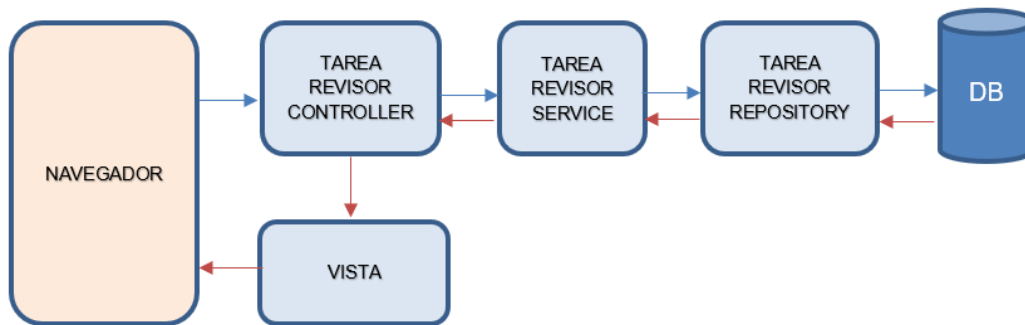


Figura 4.5: Diagrama de clases.

El contenido de cada uno de los módulos del sistema es el que se describe a continuación:

- **Modelo** (dentro de la carpeta del proyecto uned-educa-analisis/src/main/java).

Dentro de esta carpeta se definen las clases desarrolladas en java cuyas instancias podrán ser almacenadas en la bases de datos.

- Arista.java: Almacena información de una arista de un grafo que representa un mensaje de un foro.
- Asignatura.java: Almacena información de una asignatura de un curso.
- Convocatoria.java: Almacena información de un curso académico.
- Foro.java: Almacena los datos que identifican un foro.
- Grafo.java: Almacena información de aristas y personas que lo componen.
- Indicador.java: Almacena los valores que se obtienen de las métricas resultado del análisis de redes sociales.
- InfoProcesoForo.java: Almacena los resultados de haber importado un nuevo foro.
- Materia.java: Almacena datos de la asignatura, convocatoria y profesor que la imparte.
- Mensaje.java: Almacena información de un mensaje escrito por un usuario en el foro.
- Persona.java: Almacena información de un usuario (estudiante o docente) que interactúe en un foro.

- TareaRevisor.java: Almacena información del criterio de un experto (docente y usuario de la aplicación).

- **Controladores** (dentro de la carpeta del proyecto uned-educa-analisis/src/main/java)

Dentro de esta carpeta se definen aquellas clases desarrolladas en java, que se encargan de las interacciones entre el usuario y el programa. La aplicación trabaja haciendo llamadas a métodos de las plantillas web que el usuario ve. Estas llamadas son por lo general de tipo Post y Get y las principales clases son:

- AsignaturaController.java: Esta clase permite la creación, modificación, consulta y eliminación de Asignaturas.
- ConvocatoriaController.java: Esta clase permite la creación, modificación, consulta y eliminación de Convocatorias.
- ForoAlumnosController.java: Esta clase permite importar desde fichero (con una estructura determinada) datos de foros y mostrar los resultados de la importación.
- PersonaController.java: Esta clase permite la creación, modificación, consulta y eliminación de Personas.
- RevisiónController.java: Esta clase le ofrece al usuario información de asignación aleatoria de revisores.
- TareaRevisorController.java: Esta clase permite la creación, modificación, consulta y eliminación de TareasRevisor.

- **Repositorios** (dentro de la carpeta del proyecto uned-educa-analisis/src/main/java)

Dentro de esta carpeta se definen aquellas clases desarrolladas en java, que se encargan de las llamadas a las bases de datos para realizar consultas de forma sencilla. Todas ellas son clases que extienden de JPRepository (Data, 2024). Las principales clases definidas en este proyecto son:

- AristaRepository.java
- AsignaturaRepository.java
- ConvocatoriaRepository.java
- ForoRepository.java
- MateriaRepository.java

- MensajeRepository.java
- PersonaRepository.java
- RevisorRepository.java

- **Servicios** (dentro de la carpeta del proyecto uned-educa-analisis/src/main/java)

Dentro de esta carpeta se definen dos tipos de archivos, uno de *Interface* y otro de la clase que lo implementa desarrolladas en java ambas dos. Dichas clases son las que dan atienden llamadas de un controlador para llegar a las bases de datos a través de las clases repositorio. Las clases definidas en este paquete son las siguientes:

- AristaService.java
- AristaServiceImpl.java
- AsignaturaService.java
- AsignaturaServiceImpl.java
- ConvocatoriaService.java
- ConvocatoriaServiceImpl.java
- ForoAlumnosService.java
- ForoAlumnosServiceImpl.java
- ForoService.java
- ForoServiceImpl.java
- GrafoService.java
- GrafoServiceImpl.java
- MateriaService.java
- MateriaServiceImpl.java
- MensajeService.java
- MensajeServiceImpl.java
- PersonaService.java
- PersonaServiceImpl.java
- RevisionService.java
- RevisionServiceImpl.java
- TareaRevisorService.java
- TareaRevisorServiceImpl.java

A continuación, se muestra a modo de ejemplo un diagrama Unified Modeling Language (UML) de una de las clases desarrolladas en la aplicación, la de *TareaRevisor*, donde pueden verse las clases de cada uno de los módulos anteriormente mencionados.

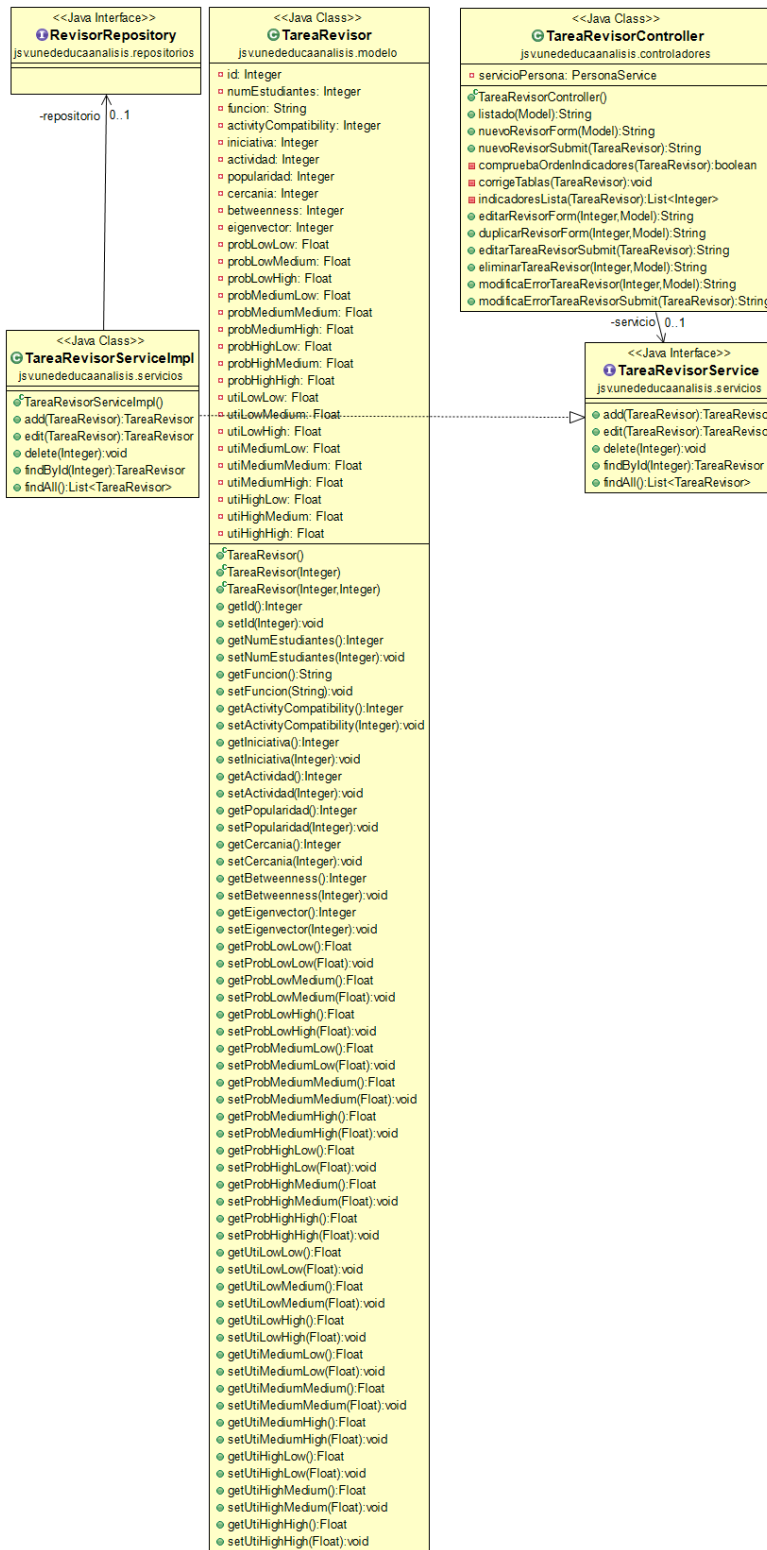


Figura 4.6: Diagrama UML de la clase TareaRevisor.

Por otro lado, el sistema cuenta con otro tipo de ficheros que permiten la configuración de las vistas que se le presentan al usuario en la carpeta *Templates* que contiene ficheros en formato html, y con otros dos documentos importantes que nos ayudan por un lado a la carga de paquetes (documento *pom.xml*) gracias a (Maven, 2024) y por otro a la configuración de nuestra aplicación (*application.properties*).

- **Templates** (dentro de la carpeta del proyecto uned-educa-analysis/src/main/resources). Dentro de esta carpeta se almacenan los archivos con formato HTML. En la figura 4.7 puede verse un ejemplo referente a la plantilla diseñada para crear un formulario para que el usuario cree una nueva *Tarea de Revisor*.

```

formTareaRevisor.html x
1 <!DOCTYPE html>
2 <html lang="es" xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
8 <meta name="description" content="">
9 <meta name="author" content="">
10
11 <title>Starter Template for Bootstrap</title>
12
13 <!-- Bootstrap core CSS -->
14 <link href="/webjars/bootstrap/css/bootstrap.min.css" rel="stylesheet">
15
16 <!-- Custom styles for this template -->
17 <link href="/css/starter-template.css" rel="stylesheet">
18
19
20 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
21 <!--[if lt IE 9]>
22 <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
23 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
24 <![endif]-->
25 </head>
26
27 <body>
28 <!-- Incorporo el menú de navegación de la página principal -->
29 <nav th:insert="index.html :: nav"> </nav>
30
31 <div class="container">
32 <div class="starter-template">
33 <h2>Revisor</h2>
34 <div class="row">
35 <div class="col-md-offset-2 col-md-8">
36

```

Figura 4.7: Ejemplo de template para diseño web.

Las plantillas elaboradas para cada una de las páginas son las que se indican a continuación:

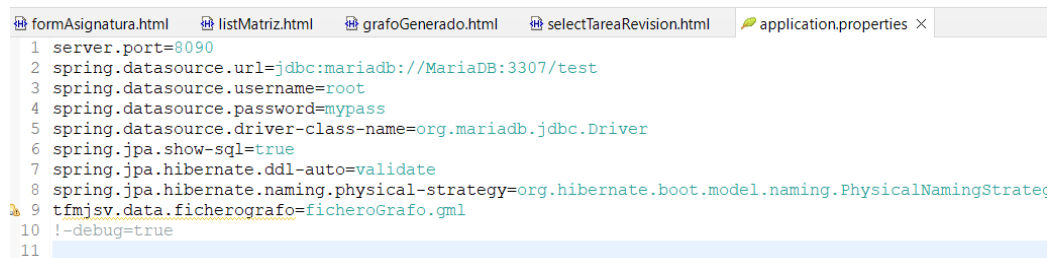
- *formAsignatura.html*: Muestra un formulario para recoger datos de una nueva Asignatura.
- *formConvocatoria.html*: Muestra un formulario para recoger datos de una nueva Convocatoria.
- *formEditTareaRevisor.html*: Muestra un formulario para editar una TareaRevisor ya creada en la base de datos.

- `formImportarForo.html`: Muestra un formulario para insertar un fichero de datos de un foro para guardarlo en la base de datos.
  - `formModifyErrorTareaRevisor.html`: Muestra un formulario para subsanar errores cometidos al crear o modificar una `TareaRevisor`.
  - `formPersona.html`: Muestra un formulario para crear datos de una nueva `Persona`.
  - `formTareaRevisor.html`: Muestra un formulario para crear una nueva `TareaRevisor`.
  - `grafoGenerado.html`: Muestra información al usuario del resultado de haber cargado los datos de un foro y haber aplicado análisis de redes sociales: tabla de adyacencia, métricas, gráficas y grafo.
  - `index.html`: Muestra de la página inicial de la aplicación.
  - `listAsignatura.html`: Muestra al usuario un listado de todas las `Asignaturas` de la base de datos. Permitirá elegir además la opción de crear nuevas y modificar o eliminar las ya creadas.
  - `listConvocatoria.html`: Muestra al usuario un listado de todas las `Convocatorias` de la base de datos. Permitirá elegir además la opción de crear nuevas y modificar o eliminar las ya creadas.
  - `listMatriz.html`: Muestra al usuario los resultado de una asignación aleatoria de revisores en forma de matrices además de la utilidad total de la mejor matriz producto lograda.
  - `listPersona.html`: Muestra al usuario un listado de todas las `Personas` de la base de datos. Permitirá elegir además la opción de crear nuevas y modificar o eliminar las ya creadas.
  - `listTareaRevisor.html`: Muestra al usuario un listado de todas las `TareaRevisor` de la base de datos. Permitirá elegir además la opción de crear nuevas y modificar o eliminar las ya creadas.
  - `resultadoImportacion.html`: Muestra al usuario el resultado de la carga de datos de un foro a través de fichero de texto.
  - `selectTareaRevision.html`: Muestra al usuario un formulario para que elija una `TareaRevisor` e indique un número de iteraciones que hará el sistema.
- **pom.xml** Fichero para la integración de APIs de diferentes compañías que permiten hacerlo de manera sencilla incluyendo dependencias. Figura 4.8.

```
65 <dependency>
66   <groupId>org.gephi</groupId>
67   <artifactId>statistics-api</artifactId>
68   <version>0.10.0</version>
69 </dependency>
```

Figura 4.8: Código para añadir dependencia de la API de Gephi.

- **application.properties** (dentro de la carpeta uned-educa-analisis/src/main/resources) Fichero donde se configuran las propiedades de la aplicación como por ejemplo los datos para la conexión a la base de datos o el puerto de servicio. En la figura 4.9 puede verse parte de la configuración del sistema para esta aplicación:



```
formAsignatura.html listMatriz.html grafoGenerado.html selectTareaRevision.html application.properties x
1 server.port=8090
2 spring.datasource.url=jdbc:mariadb://MariaDB:3307/test
3 spring.datasource.username=root
4 spring.datasource.password=mypass
5 spring.datasource.driver-class-name=org.mariadb.jdbc.Driver
6 spring.jpa.show-sql=true
7 spring.jpa.hibernate.ddl-auto=validate
8 spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStratec
9 tfmjvs.data.ficherografo=ficheroGrafo.gml
10 !-debug=true
11
```

Figura 4.9: Configuración del fichero application.properties.

### 4.3. Casos de uso

Una vez familiarizado con la interfaz desarrollada, en la figura 4.10 pueden observarse los casos de uso de más alto nivel definidos dentro de este proyecto para la aplicación. A continuación, se describen cada uno de ellos:



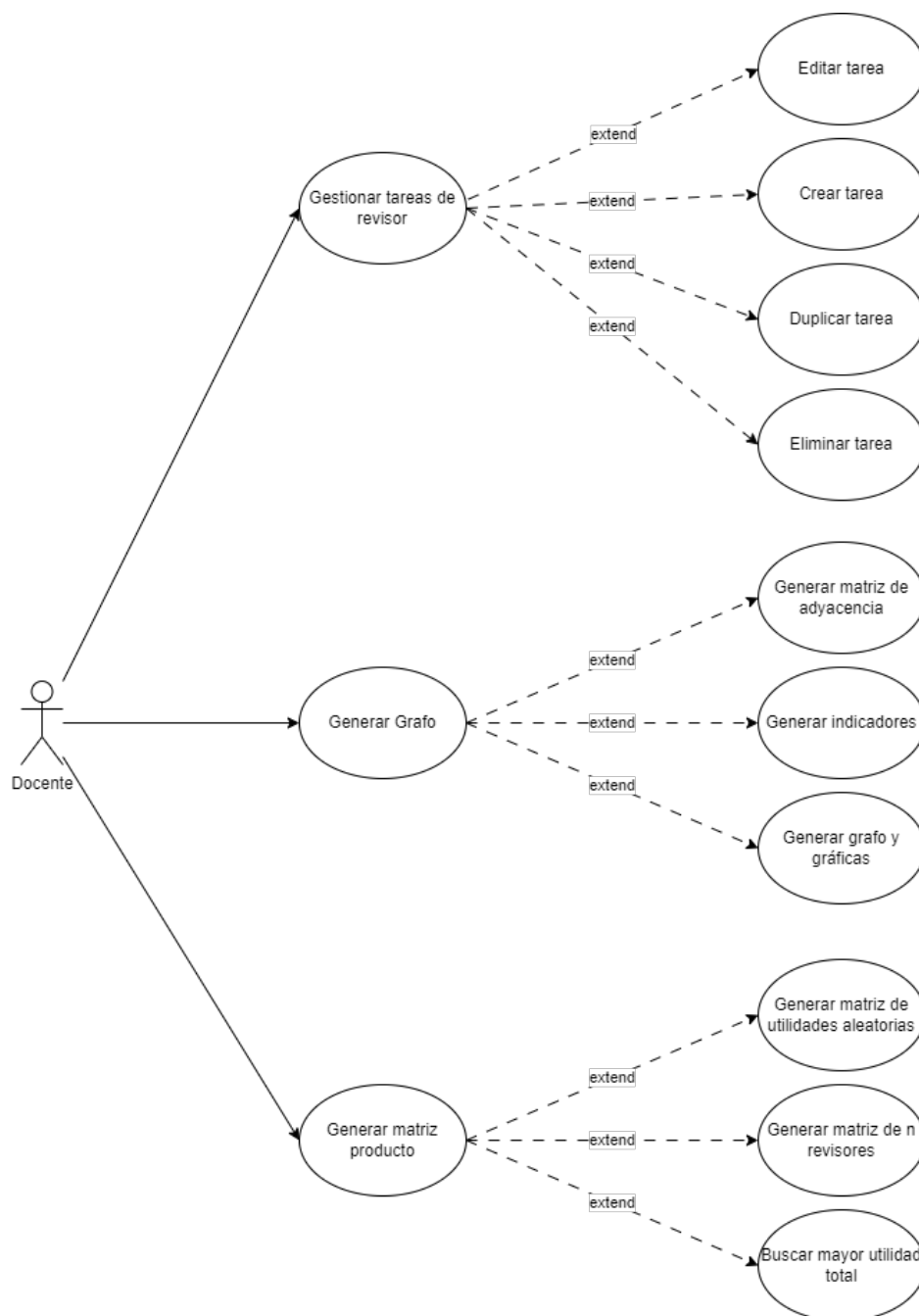


Figura 4.10: Casos de uso de la aplicación.

- **Caso de uso 1:** Gestionar tareas de revisor. El docente dispondrá de un apartado para poder gestionar todos aquellos criterios que, como experto, se utilizarán para el análisis de las relaciones de estudiantes.

Estas tareas de revisor y lo que engloba serán descritas en secciones futuras.

- **Caso de uso 2:** Editar tarea. El usuario podrá editar una tarea de revisor ya creada.
- **Caso de uso 3:** Crear tarea. El usuario podrá crear una nueva tarea de revisor y permitir que esta sea almacenada.
- **Caso de uso 4:** Duplicar tarea. Se facilitará al usuario la duplicidad de tareas de revisor para mejorar la usabilidad de la aplicación.
- **Caso de uso 5:** Generar Grafo. El usuario obtendrá información útil, alguna de esta información ya fue implementada en el trabajo previo, las nuevas se desglosan en los siguientes casos de uso.
- **Caso de uso 6:** Generar matriz de adyacencia. Mediante una tabla se podrá observar cuántos mensajes ha escrito un estudiante a cada uno de los otros estudiantes.
- **Caso de uso 7:** Generar indicadores. Implementación del SNA para evaluar el grafo generado.
- **Caso de uso 8:** Generar grafo y gráficas. Permite generar imágenes de grafos y gráficas resultado de los dos casos de uso anteriores.
- **Caso de uso 9:** Generar matriz producto. Será el resultado de implementar OPA. El programa mostrará una tabla o matriz resultado del producto de las celdas que ocupan la misma posición de las matrices que se describen en los casos de uso siguientes.
- **Caso de uso 10:** Generar matriz de utilidades aleatorias. El programa muestra al usuario una tabla de  $N \times N$  generada de forma automática, siendo  $N$  el número de estudiantes registrados de una asignatura, donde se rellenan cada una de las celdas con valores aleatorios entre 0 y 100, siendo siempre la diagonal igual a 0.
- **Caso de uso 11:** Generar matriz de  $n$  revisores. Se genera una tabla o matriz de  $N \times N$  de forma automática, donde  $N$  es el número de estudiantes registrados de una asignatura, y las celdas podrán tomar valor 0 o 1, donde 1 significa que un estudiante revisa a otro estudiante.  $N$  revisores significa que cada estudiante revisará a  $n$  estudiantes y un estudiante será revisado por  $n$  estudiantes y nunca por él mismo.

- **Caso de uso 12:** Buscar mayor utilidad total. Se genera un valor que indica la utilidad total máxima encontrada por el programa. Esta utilidad será mayor cuanto mayor sea el número de iteraciones que realice el programa. El número de iteraciones que realice la aplicación será indicado por el usuario. Un mayor número de iteraciones va en detrimento de la rapidez en la obtención de resultado.

## 4.4. Descripción de la interfaz

La interfaz diseñada parte, como se ha dicho, de una ya creada en un anterior proyecto y se pretende seguir con la misma filosofía y estilo. Una vez que el usuario accede a la aplicación se muestra una página principal o inicio como la que se muestra en la siguiente figura 4.11.

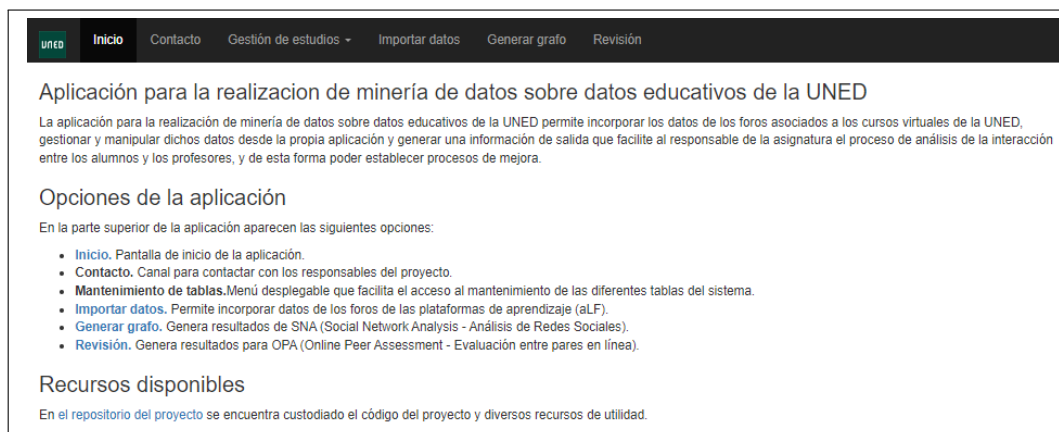


Figura 4.11: Página de inicio de la aplicación.

En la parte superior se muestra un menú con las siguientes opciones:

- Inicio
- Contacto
- Gestión de estudios
- Importar datos
- Generar grafo
- Revisión

#### 4.4.1. Diagrama de flujo

Conocidas las opciones del menú, se presenta el diagrama de flujo de la aplicación en la figura 4.12.

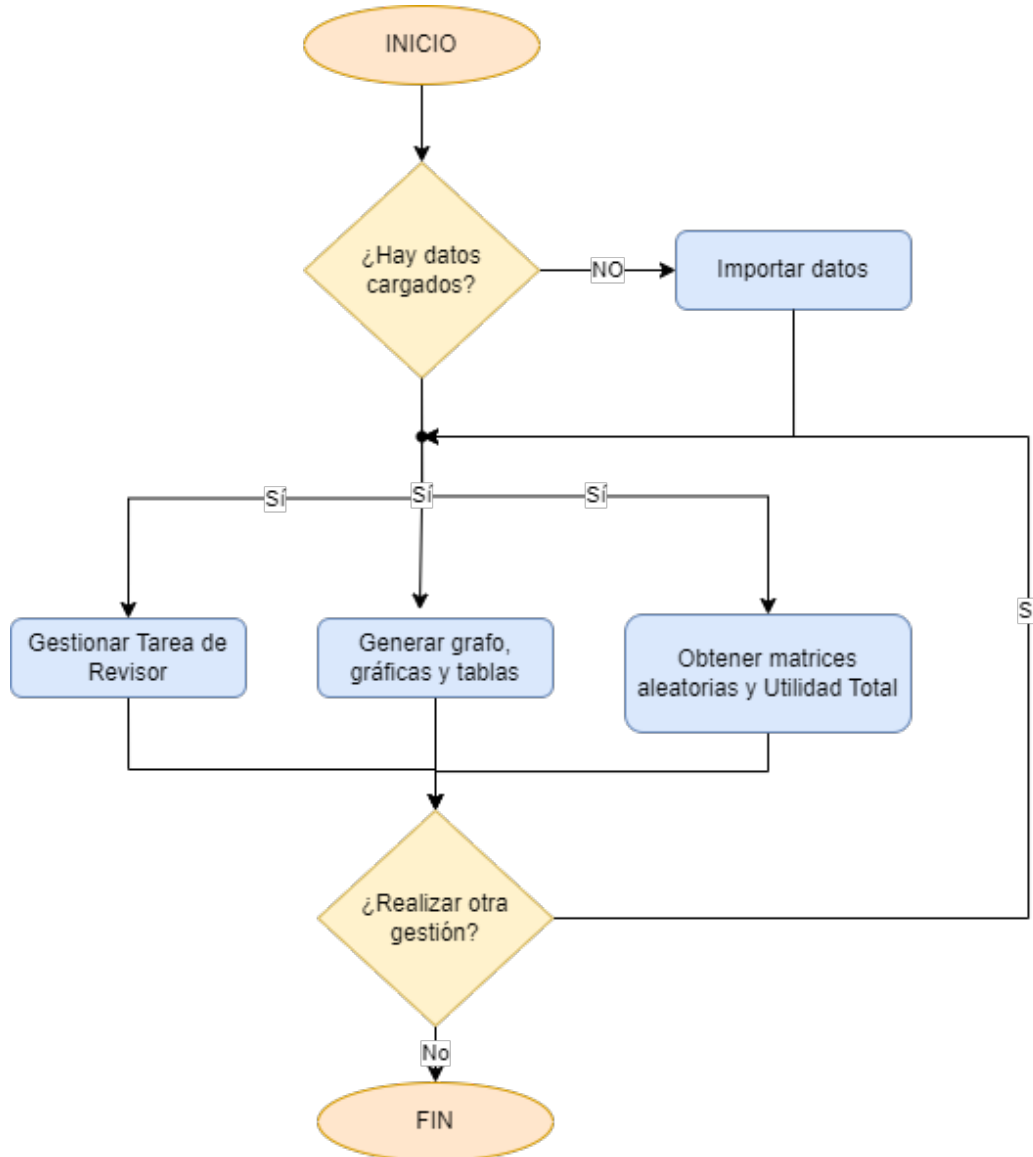


Figura 4.12: Diagrama de flujo de la aplicación.

En el diagrama pueden verse la siguientes acciones:

- Importar datos: Puede realizarse a través de la opción del menú *Importar datos*.

- Gestionar Tarea de Revisor: Puede realizarse desde la opción del menú *Gestión de estudios*.
- Generar grafo, gráficas y tablas: Puede hacerse desde la opción del menú *Generar grafo*.
- Obtener matrices aleatorias y Utilidad Total: Puede hacerse a través de la opción *Revisión*.

#### 4.4.2. Estructura de las pantallas

A continuación se muestran unos bocetos creados con la aplicación Mock-Flow (MockFlow, 2024), donde muestran los diseños que se especifican para la interfaz. En todos ellos se muestra un menú superior. Marcado en verde en la 4.13 está el apartado *Gestión de estudios*, desde donde el usuario puede acceder a la gestión de *Tareas de Revisor* desde las que podrá crear, modificar o borrar una *lista de tareas*.

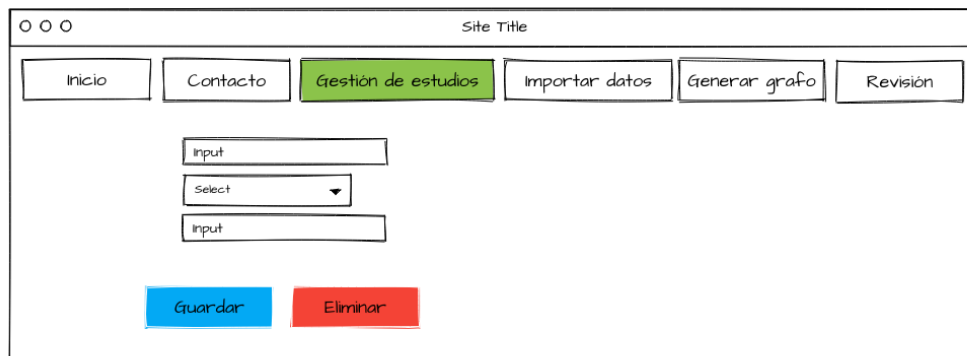


Figura 4.13: Diseño de la página de Tareas de Revisión.

Desde el menú superior también el usuario puede acceder al apartado *Generar grafo* desde el cual el usuario obtendrá de forma automática valores resultado de análisis de redes sociales en una página como la que se muestra en la figura 4.14

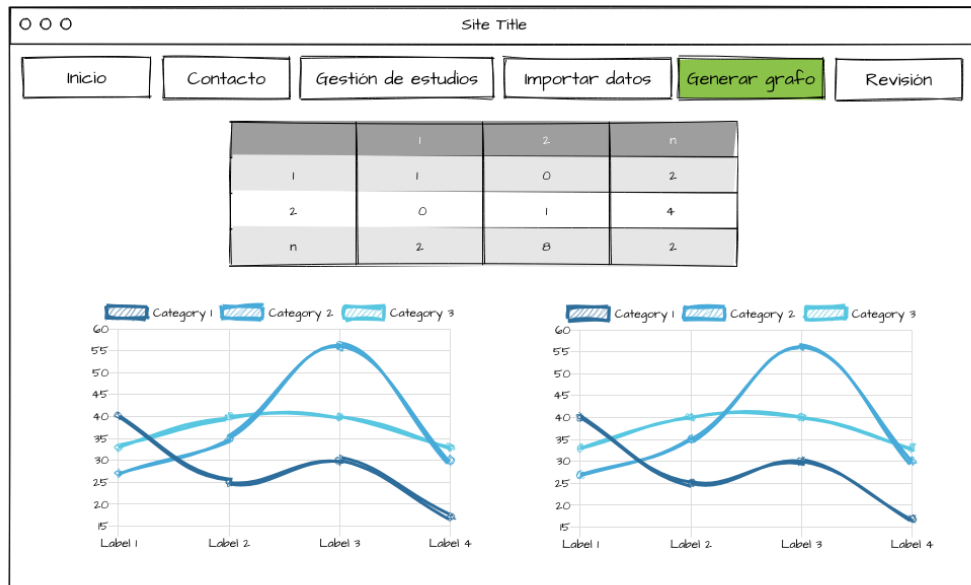


Figura 4.14: Diseño de la página de Generar Grafo.

Por último, el usuario puede acceder desde el menú superior al apartado *Revisión* desde el cual se obtienen resultados de asignación aleatoria de revisores por pares para el usuario. En la figura 4.15 puede verse el sketch diseñado.



Figura 4.15: Diseño de la página de Revisión.

En todo momento se tendrá en cuenta que toda la información que se obtenga (datos, tablas, imágenes, etc.) se pretende que sean descargables para el usuario de forma sencilla.





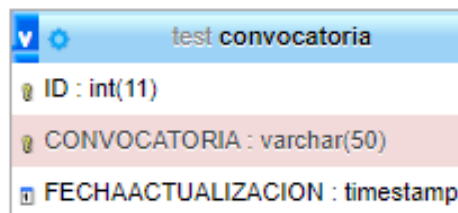
# Capítulo 5

## Diseño del sistema

### 5.1. Diseño de la base de datos

La base de datos fue diseñada con la intención de almacenar la información extraída de los foros de la plataforma de aprendizaje. Las tablas diseñadas originalmente son las que se describen a continuación:

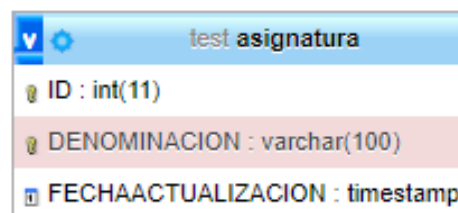
1. **Convocatoria:** Tabla encargada de gestionar las diferentes convocatorias anuales.



test convocatoria	
ID	int(11)
CONVOCATORIA	varchar(50)
FECHAACTUALIZACION	timestamp

Figura 5.1: Tabla Convocatoria de la base de datos.

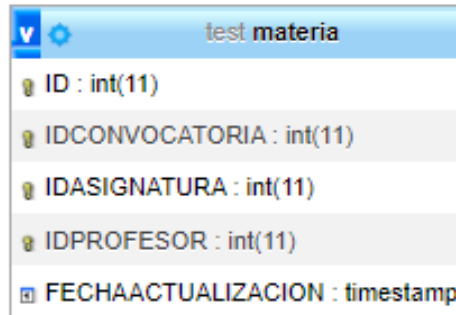
2. **Asignatura:** Tabla que se encarga de gestionar la información relevante de las diferentes asignaturas.



test asignatura	
ID	int(11)
DENOMINACION	varchar(100)
FECHAACTUALIZACION	timestamp

Figura 5.2: Tabla Asignatura de la base de datos.

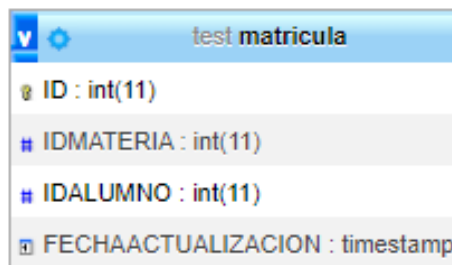
3. **Materia:** Tabla encargada de gestionar las diferentes materias. Representa una materia impartida, por un profesor determinado, en un curso determinado, y de una asignatura concreta.



test materia	
ID	int(11)
IDCONVOCATORIA	int(11)
IDASIGNATURA	int(11)
IDPROFESOR	int(11)
FECHAACTUALIZACION	timestamp

Figura 5.3: Tabla Materia de la base de datos.

4. **Matrícula:** Tabla donde se gestionan las diferentes materias en las cuales se ha matriculado un alumno. Es una tabla que básicamente relaciona información de materias y alumnos.



test matricula	
ID	int(11)
IDMATERIA	int(11)
IDALUMNO	int(11)
FECHAACTUALIZACION	timestamp

Figura 5.4: Tabla Matrícula de la base de datos.

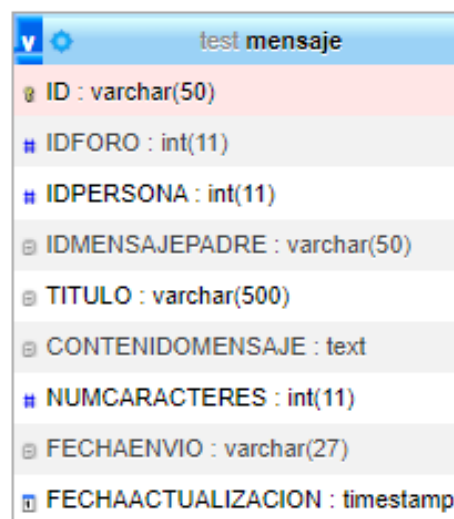
5. **Foro:** Entidad que gestiona los diferentes foros existentes en la UNED. Un foro está asignado a una materia. Una materia puede disponer de diferentes foros (foro de alumnos, tablón de anuncios...).



test foro	
ID	int(11)
IDMATERIA	int(11)
FECHACREACION	timestamp
DENOMINACION	varchar(100)
DESCRIPCION	varchar(150)
FECHAACTUALIZACION	timestamp

Figura 5.5: Tabla Foro de la base de datos.

6. **Mensaje:** Tabla donde se gestionan los diferentes mensajes generados en la plataforma.



test mensaje	
ID	varchar(50)
IDFORO	int(11)
IDPERSONA	int(11)
IDMENSAJEPADRE	varchar(50)
TITULO	varchar(500)
CONTENIDOMENSAJE	text
NUMCARACTERES	int(11)
FECHAENVIO	varchar(27)
FECHAACTUALIZACION	timestamp

Figura 5.6: Tabla Mensaje de la base de datos.

7. **Persona:** Tabla donde se gestionan las diferentes personas que forman parte de materias (ya sea como alumno o como profesor) de la UNED.



The image shows a screenshot of a database table structure for a table named 'test persona'. The table has seven columns with the following data types: ID (int(11)), USUARIO (varchar(100)), APELLIDOS (varchar(100)), NOMBRE (varchar(50)), EMAIL (varchar(100)), and FECHAACTUALIZACION (timestamp). The 'ID' column is highlighted in red, indicating it is the primary key.

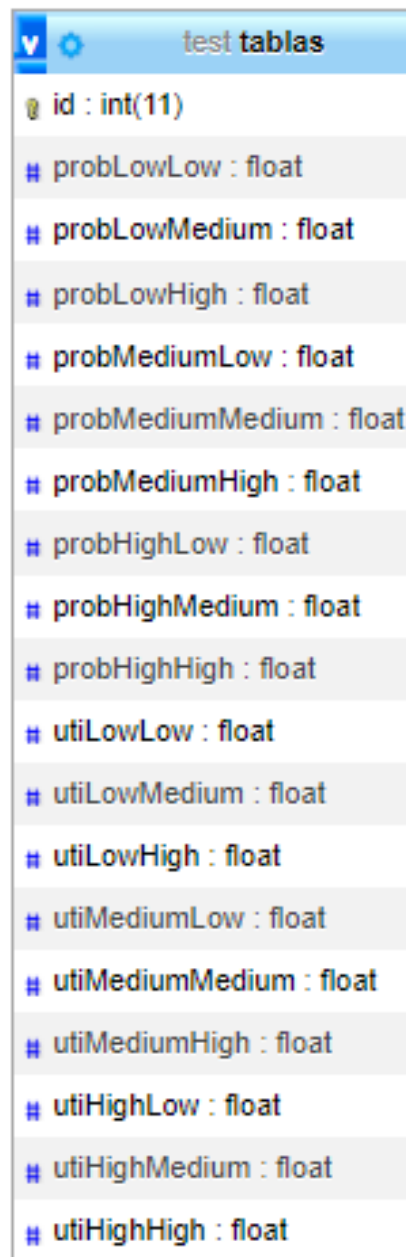
Column Name	Data Type
ID	int(11)
USUARIO	varchar(100)
APELLIDOS	varchar(100)
NOMBRE	varchar(50)
EMAIL	varchar(100)
FECHAACTUALIZACION	timestamp

Figura 5.7: Tabla Persona de la base de datos.

La estructura de estas siete tablas descritas no han sido modificadas. Para este proyecto, tan solo ha sido necesaria la creación de dos nuevas tablas llamadas *Tablas* y *Revisión* respectivamente y son las que se muestra en las figuras 5.8 y 5.9.

Las sentencias SQL utilizadas para la creación de todas las tablas que contiene la aplicación, están disponibles para el lector en la carpeta *BBDD*.

8. **Tablas:** Gestiona los valores por defecto que contendrán las tablas de probabilidad y de utilidad.



test tablas	
id	int(11)
probLowLow	float
probLowMedium	float
probLowHigh	float
probMediumLow	float
probMediumMedium	float
probMediumHigh	float
probHighLow	float
probHighMedium	float
probHighHigh	float
utiLowLow	float
utiLowMedium	float
utiLowHigh	float
utiMediumLow	float
utiMediumMedium	float
utiMediumHigh	float
utiHighLow	float
utiHighMedium	float
utiHighHigh	float

Figura 5.8: Tabla Tablas de la base de datos.

La idea es que esta tabla no sufra alteraciones, por lo que contendrá valores por defecto que son los que se muestran en la tabla 5.1 y 5.2 respectivamente.

9. **Revision:** En esta tabla se gestionan todos los datos que son almace-

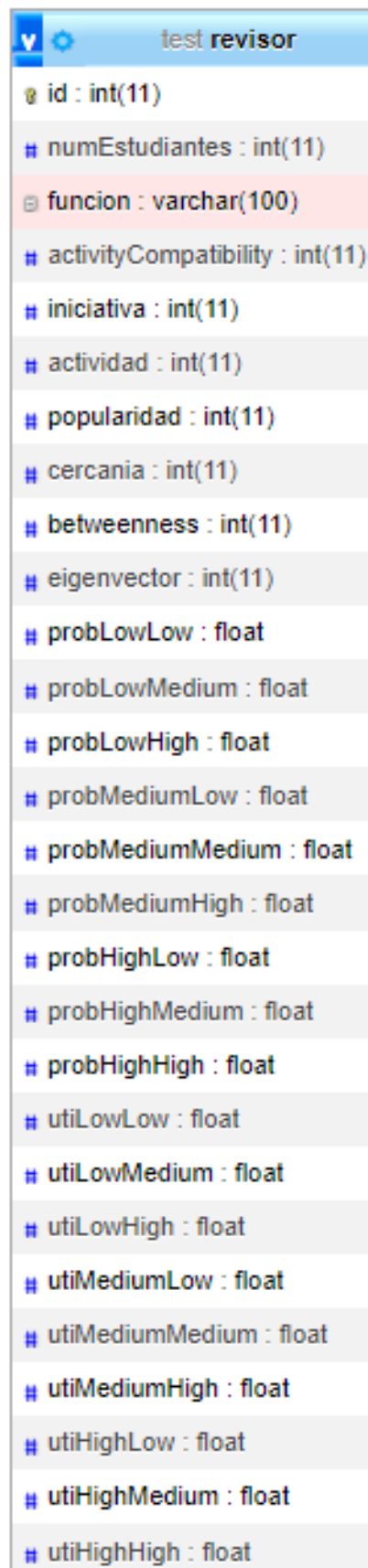
Tabla 5.1: Valores por defecto de la tabla Probabilidad.

Probabilidad	Low	Medium	High
Low	0.7	0.2	0.1
Medium	0.15	0.7	0.15
High	0.1	0.2	0.7

Tabla 5.2: Valores por defecto de la tabla Utilidad.

Utilidad	Low	Medium	High
Low	0.6	0.25	0.15
Medium	0.2	0.6	0.2
High	0.15	0.25	0.6

nados para una Tarea de revisión, que engloba datos extraídos de SNA y relacionados con OPA.



The image shows a screenshot of a database table definition for a table named 'test revisor'. The table has the following columns:

Column Name	Data Type
id	int(11)
numEstudiantes	int(11)
funcion	varchar(100)
activityCompatibility	int(11)
iniciativa	int(11)
actividad	int(11)
popularidad	int(11)
cercania	int(11)
betweenness	int(11)
eigenvector	int(11)
probLowLow	float
probLowMedium	float
probLowHigh	float
probMediumLow	float
probMediumMedium	float
probMediumHigh	float
probHighLow	float
probHighMedium	float
probHighHigh	float
utiLowLow	float
utiLowMedium	float
utiLowHigh	float
utiMediumLow	float
utiMediumMedium	float
utiMediumHigh	float
utiHighLow	float
utiHighMedium	float
utiHighHigh	float

Figura 5.9: Tabla Revisión de la base de datos.

## 5.2. Diseño de la interfaz

Esta sección describe el diseño creado para la interfaz de usuario de la aplicación. El diseño se desglosa en tres grandes bloques:

- Gestión de Tareas de Revisor.
- Generar grafo.
- Revisión.

### 5.2.1. Gestión de Tareas de Revisor

Desde el menú superior de la aplicación se puede acceder a la gestión de tareas de revisor desde el apartado *Gestión de estudios*, seleccionando la opción *Gestión de tareas de revisor*, tal y como puede verse en la figura 5.10.



Figura 5.10: Acceso a Gestión de tareas de revisor.

Llamamos tareas de revisor a una serie de atributos que el docente configura, es decir, son los criterios que el experto introduce para obtener mejores resultados en la asignación de revisores.

### Listado de Tareas de Revisor

Cuando el usuario accede a la *Gestión de tareas de revisor*, ve un listado con las tareas de revisor creadas previamente. Además, se le da la opción de crear una nueva y también de modificar o eliminar alguna de las ya existentes. En la figura 5.11 se muestra un ejemplo de un listado de tareas de revisor.



unco Inicio Contacto Gestión de estudios - Importar datos Generar grafo Revisión

### Listado de tareas de revisor

[Nueva tarea](#)

Identificador	Número de revisores	Función	ActivityCompatibility	Iniciativa	Actividad	Popularidad	Cercanía	Betweenness	Eigenvector			
1	3	Lineal	1	2	3	4	5	6	7	<a href="#">Editar</a>	<a href="#">Duplicar</a>	<a href="#">Eliminar</a>
2	4	Lineal	1	2	3	4	5	6	7	<a href="#">Editar</a>	<a href="#">Duplicar</a>	<a href="#">Eliminar</a>
3	1	Cuadrática	1	3	2	4	5	7	6	<a href="#">Editar</a>	<a href="#">Duplicar</a>	<a href="#">Eliminar</a>

Figura 5.11: Listado de tareas de revisor.

### Modificación y creación de Tarea de Revisor

En este apartado se describe cada uno de los atributos que el docente y usuario de la aplicación introducirá cuando cree o modifique una tarea de revisor.

- **Número de revisores:** Es un valor numérico que indica a cuantos alumnos revisará la tarea un revisor. Este campo podrá tomar valor entre 1 y  $n-1$ , siendo  $n$  el número de estudiantes totales cargados en el sistema.
- **Función:** Existen tres tipos de funciones: lineal, cuadrática y raíz cuadrada. El usuario seleccionará uno de ellos.

En la aplicación, cuando se crea una nueva *Tarea de Revisor*, el usuario debe indicar valores para estos atributos. La figura 5.12 muestra estos campos *Número de revisores* y *Selecciona función* del formulario.

### Revisor

Identificador (se genera de forma automática)

Número de revisores

Selecciona función

Figura 5.12: Formulario para crear una nueva tarea de revisor.

- Orden de los indicadores:** El experto debe indicar la importancia que tiene cada uno de los indicadores, siendo 1 el valor máximo para un indicador y 7 el de menor importancia. En la figura 5.13 se muestra la forma en que el usuario dará prioridades a las métricas. La aplicación a su vez controlará que ningún valor puede ser repetido ni menor que 1 o mayor que 7.

Asignar orden a las métricas:	
Activity/Compatibility	<input type="text" value="1"/>
Iniciativa	<input type="text" value="2"/>
Actividad	<input type="text" value="3"/>
Popularidad	<input type="text" value="4"/>
Cercanía	<input type="text" value="5"/>
Betweenness	<input type="text" value="6"/>
Eigenvector	<input type="text" value="7"/>

Figura 5.13: Formulario para asignar orden a los indicadores.

- Tabla de Parámetros de Probabilidad:**

Se necesita obtener los valores numéricos correspondientes a los parámetros de probabilidad y utilidad.

Para la tabla de probabilidad, serán generados 9 parámetros donde solo seis de ellos son independientes. Al usuario se le habilitan 6 casillas para indicar la probabilidad de actividad entre estudiantes agrupada por valores (low, medium y high, es decir, bajo, medio o alto). Las casillas que se muestran sombreadas (referidos a los parámetros que no son independientes) son autocalculadas en el momento que el usuario pulsa el botón *Guardar*.

En la figura 5.14 puede verse cómo se muestra el formulario para rellenar estos datos al usuario.

**Tabla 1: Parámetros de P( ActividadEstudiante | MensajesEstudiante )**

\* Los valores sombreados se autorrecalculan al guardar los cambios

Mensajes de estudiante	Actividad de estudiante	Probabilidad
Low	Low	<input type="text" value="0,7"/>
Low	Medium	<input type="text" value="0,2"/>
Low	High	<input type="text" value="0,1"/>
Medium	Low	<input type="text" value="0,15"/>
Medium	Medium	<input type="text" value="0,7"/>
Medium	High	<input type="text" value="0,15"/>
High	Low	<input type="text" value="0,1"/>
High	Medium	<input type="text" value="0,2"/>
High	High	<input type="text" value="0,7"/>

Figura 5.14: Tabla de probabilidad.

- **Tabla de utilidad de compatibilidad de actividad:**

La tabla de utilidad del nodo *Activity Compatibility* contiene nueve parámetros, pero solo seis de ellos son independientes y deben ser insertados. La figura 5.15 muestra las 6 celdas donde el usuario introducirá los valores que considere oportuno. Al finalizar la edición de la tarea de revisor, el programa recalcula los valores que tomarán los parámetros independiente.

**Tabla 2: ActivityCompatibility**  
\* Los valores sombreados se autorrecalculan al guardar los cambios

Actividad del estudiante	Actividad del revisor	Utility
Low	Low	0,6
Low	Medium	0,25
Low	High	0,15
Medium	Low	0,2
Medium	Medium	0,6
Medium	High	0,2
High	Low	0,15
High	Medium	0,25
High	High	0,6

Figura 5.15: Tabla de utilidades de *Activity Compatibility*.

### 5.2.2. Generar grafo

Seleccionando la opción *Generar grafo* del menú superior de la aplicación, el usuario obtiene automáticamente los resultados de aplicar análisis de redes sociales sobre la información almacenada de los foros.

En esta sección, se le ofrece al usuario nueva información desglosada en los siguiente bloques:

- Tabla de adyacencia
- Grafo
- Tabla de valores de indicadores
- Gráficas de indicadores

#### Tabla de adyacencia

La tabla o matriz de adyacencia muestra el número de mensajes intercambiados entre los alumnos. Esta información se obtiene de los foros que

hay cargado en la base de datos. A continuación, en la figura 5.16 se muestra un ejemplo de matriz de adyacencia.

Matriz de adyacencia										
	245	244	243	242	241	240	239	238	237	236
245	0	1	0	0	1	1	0	0	1	2
244	1	0	1	0	1	2	0	0	1	1
243	0	0	2	0	1	1	0	0	1	5
242	1	1	0	0	4	1	0	1	0	3
241	2	0	1	0	4	2	0	0	1	8
240	1	1	1	0	0	0	0	0	1	8
239	0	0	0	0	0	1	0	0	0	8
238	0	0	0	0	1	1	0	0	0	4
237	0	2	0	0	1	2	0	0	3	6
236	1	0	5	2	3	5	4	1	7	7

Figura 5.16: Matriz de adyacencia del foro.

En la parte izquierda de la tabla pueden verse los identificadores de los alumnos con los valores que van desde el 245 al 236. En la parte superior de la tabla observamos los mismos valores identificativos. En la figura 5.17 se muestra un ejemplo para la comprensión de la matriz: marcado con un rectángulo rojo en la columna izquierda está el alumno 242 y en la parte superior el 241, uniendo estos dos nodos en la tabla aparece rodeado en azul el valor 4 que indica que el alumno 242 ha respondido a 4 mensajes del alumno 241.

Matriz de adyacencia										
	245	244	243	242	241	240	239	238	237	236
245	0	1	0	0	1	1	0	0	1	2
244	1	0	1	0	1	2	0	0	1	1
243	0	0	2	0	1	1	0	0	1	5
242	1	1	0	0	4	1	0	1	0	3
241	2	0	1	0	4	2	0	0	1	8
240	1	1	1	0	0	0	0	0	1	8
239	0	0	0	0	0	1	0	0	0	8
238	0	0	0	0	1	1	0	0	0	4
237	0	2	0	0	1	2	0	0	3	6
236	1	0	5	2	3	5	4	1	7	7

Figura 5.17: Mensajes intercambiados entre los nodos 242 y 241.

De esta forma, puede verse de una manera simple el número de mensajes

intercambiados en el foro, mejorando la visualización del grafo de la figura 5.18.

## Grafo

Para poder implementar SNA en el proyecto, la herramienta Gephi ofrece desde su página web (Gephi, 2024a) la descarga de una API para poder generar estos indicadores que se han descrito.

Para usarla, se instala en el IDE Spring Tool. Por un lado, se crea la carpeta *libs* en el proyecto y se pega en ellas el toolkit descargado y por último, en las propiedades del proyecto, en la sección *Java Build Path*, en la pestaña *Libraries*, se añade desde el botón *Add external Jars* el mismo fichero descargado (.jar).

Además de esto, será necesario agregar la dependencia de Gephi con la versión que se considere oportuna. En este caso se ha utilizado la versión 10.0.

Una vez conocida la forma y herramientas utilizadas para la obtención de los indicadores, se muestra la parte visual diseñada para mostrar la información obtenida al usuario. Para ello, una vez cargados valores de un foro en la base de datos, la API de Gephi permite crear un grafo, donde un nodo de ese grafo representa a un alumno o un profesor y donde un mensaje del foro se expresa como una arista de dicho grafo.

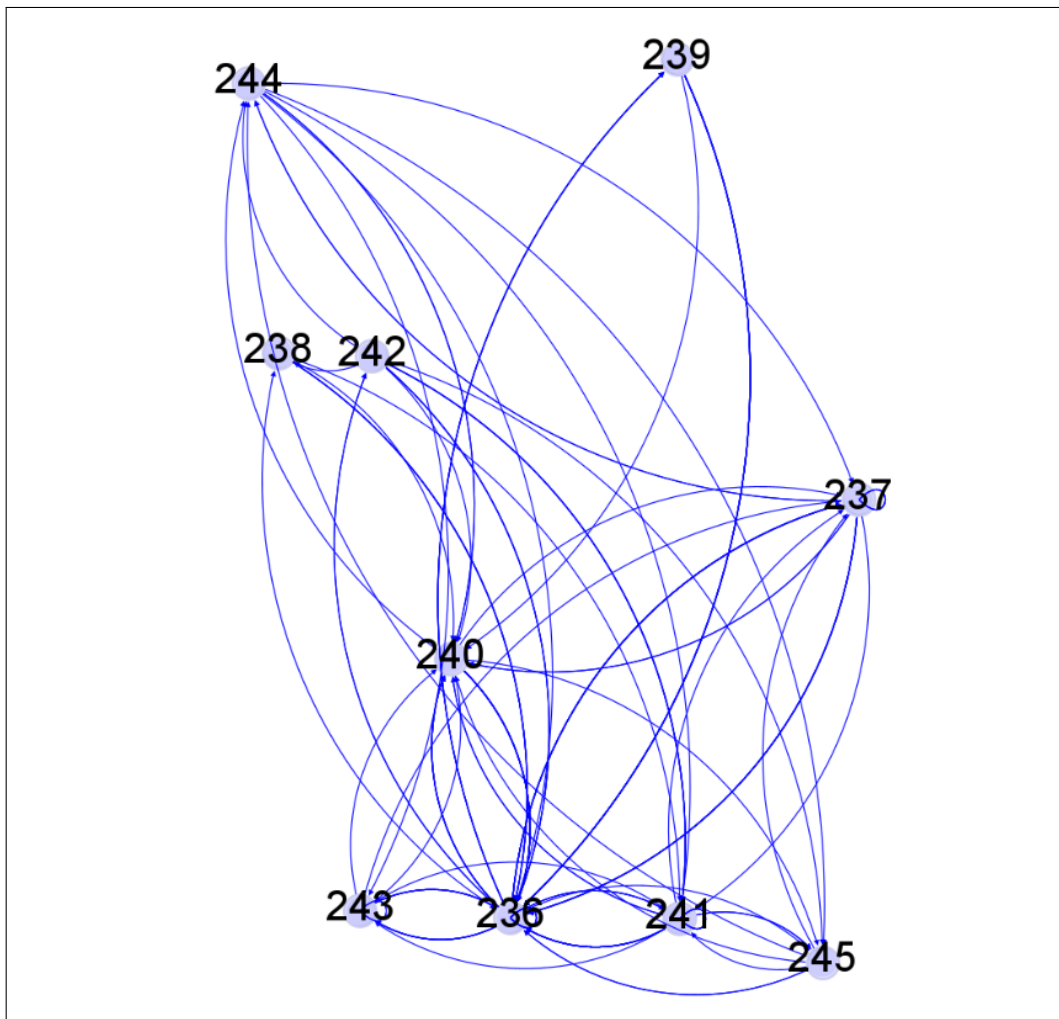


Figura 5.18: Grafo generado con los datos de un foro por la aplicación.

En la figura 5.18 se muestra un grafo generado por la aplicación creada en este proyecto. En ella puede verse que con valores numéricos se nombran cada uno de los nodos del grafo, siendo cada uno de ellos el identificador de un alumno. Con flechas o aristas, se muestran todos los mensajes intercambiados entre los alumnos.

Como ya se indicó en la sección 3.1.5, se mantienen las funcionalidades hechas en la versión inicial. Es decir, se mantiene para el usuario la posibilidad de descargar un fichero con extensión *.gml* para poder utilizarlo en la herramienta Gephi. Pero, para implementar los nuevos requisitos de este proyecto, se ha utilizado la API de Gephi, para la creación interna de una estructura grafo, de forma que se automatizan y agilizan los procesos para la obtención de grafos como el de la figura 5.18 o los indicadores que se indican

a continuación.

### Tabla de valores de indicadores

Dentro de SNA existen unas métricas que ayudan a caracterizar el comportamiento social de los alumnos a partir de sus interacciones en los foros. De esta forma, logramos otro de los objetivos del proyecto que es el de implementar métodos para el modelado del comportamiento social de los estudiantes. Estas variables o indicadores son las que se definen a continuación:

- **Iniciativa**

Es el número de conversaciones que un alumno ha iniciado en los foros, un valor alto en este atributo indica que un alumno a menudo inicia nuevas conversaciones en los foros.

- **Actividad** (*Outdegree*)

Es el número de mensajes que escribe un alumno, es también llamado grado de salida de un nodo.

- **Popularidad** (*Popularity*)

Es el número de respuestas a los mensajes enviados por un alumno. Si un nodo posee un valor alto de este atributo indica que el alumno es muy popular.

- **Cercanía** (*Closeness Centrality*)

Este valor indica lo cerca que está un alumno de otros alumnos, cuanto mayor sea su centralidad de cercanía, más rápido accederá a otros alumnos.

- **Intermediación** (*Betweenness Centrality*)

Cuando un alumno actúa con mayor frecuencia como infomediario (una persona que difunde información entre vecinos) entre otros dos alumnos, ese alumno tendrá una mayor intermediación.

- **Centralidad del vector propio** (*Eigenvector centrality*)

Este valor es una medida que evalúa la importancia de un alumno considerando no solo la cantidad de mensajes directos que intercambia, sino también la importancia de los alumnos con los que intercambia mensajes.



- **Modularidad** (*Modularity*)

Este valor es una medida que evalúa la calidad de la partición de alumnos en el foro. Busca identificar grupos de alumnos que tengan más conexiones internas entre sí que las conexiones esperadas al azar.

Los atributos *Popularidad*, *Vector Propio*, *Intermediación*, *Iniciativa* y *Cercanía*, representan atributos sociales de los estudiantes. Por otro lado, la *Modularidad* indica la pertenencia del estudiante a un grupo de estudiantes relacionados. Todos los estos indicadores son considerador atributos cuantitativos.

Existe además de los descritos anteriormente otro atributo al que llamamos *Compatibilidad de la actividad* (en la aplicación desarrollada podrá verse como *Activity Compatibility*), que es un atributo de tipo cualitativo que expresa la compatibilidad entre la actividad de los estudiantes.

A continuación, se muestra la forma en que se le ofrecen al usuario los resultados del análisis de redes sociales, más concretamente el de los indicadores, y el programa lo hace en forma de tabla tal y como puede verse en la siguiente figura 5.19:

Indicadores							
	Iniciativa	Actividad	Popularidad	Cercanía (Closeness Centrality)	Intermediación (Betweenness Centrality)	Centralidad del vector propio(Eigenvector)	Modularidad
245	0	6	6	0.6923076923076923	0.4502164502164502	0.11162500626316664	0
244	0	7	5	0.75	0.41978021978021973	0.0809784854928759	0
243	3	10	10	0.6428571428571429	0.03571428571428571	0.3277708479445223	0
242	2	11	2	0.75	0.09090909090909091	0.10185064983057664	0
241	3	18	16	0.6923076923076923	1.21999666999667	0.2918315039448697	0
240	2	12	16	0.6923076923076923	3.777695564313022	0.3862051848488964	0
239	0	9	4	0.5625	0.0	0.20370129966115327	0
238	1	6	2	0.6	0.0	0.056385754271101485	0
237	4	14	15	0.6428571428571429	1.6720779220779223	0.4934106631637031	0
236	5	35	52	0.9	34.33360979699234	1.0	0

Figura 5.19: Tabla de indicadores del grafo.

Para la obtención de estos resultados, ha sido necesario contrastar que los resultados logrados por la aplicación desarrollada para este proyecto coinciden con los resultados que ofrece la propia herramienta de Gephi. Para ello, se utiliza el fichero gml que se descarga desde esta aplicación y se toma como dato de entrada para Gephi. Se debe tener en cuenta que Gephi, a la hora de crear el proyecto, debe seleccionarse la opción de no combinar arista, pues los resultados no coincidirían.

usuario	Closeness Centrality	Betweenness Centrality	Eigenvector Centrality	Modularity Class
user_10	0.692308	0.4502164502164502	0.111625	0
user_9	0.75	0.41978	0.080978	0
user_8	0.642857	0.035714	0.327771	0
user_7	0.75	0.090909	0.101851	0
user_6	0.692308	1.219997	0.291832	0
user_5	0.692308	3.777696	0.386205	0
user_4	0.5625	0.0	0.203701	0
user_3	0.6	0.0	0.056386	0
user_2	0.642857	1.672078	0.493411	0
user_1	0.9	34.33361	1.0	0

Figura 5.20: Tabla con los resultados de Gephi.

### Gráficas de indicadores

Por último, otra de las opciones que se ofrece al usuario es la de poder ver o descargar los gráficos de los indicadores *Centralidad*, *Intermediación*, *Centralidad del vector propio* y *Modularidad*, tal y como se ve en las figuras 5.21, 5.22, 5.23 y 5.24.

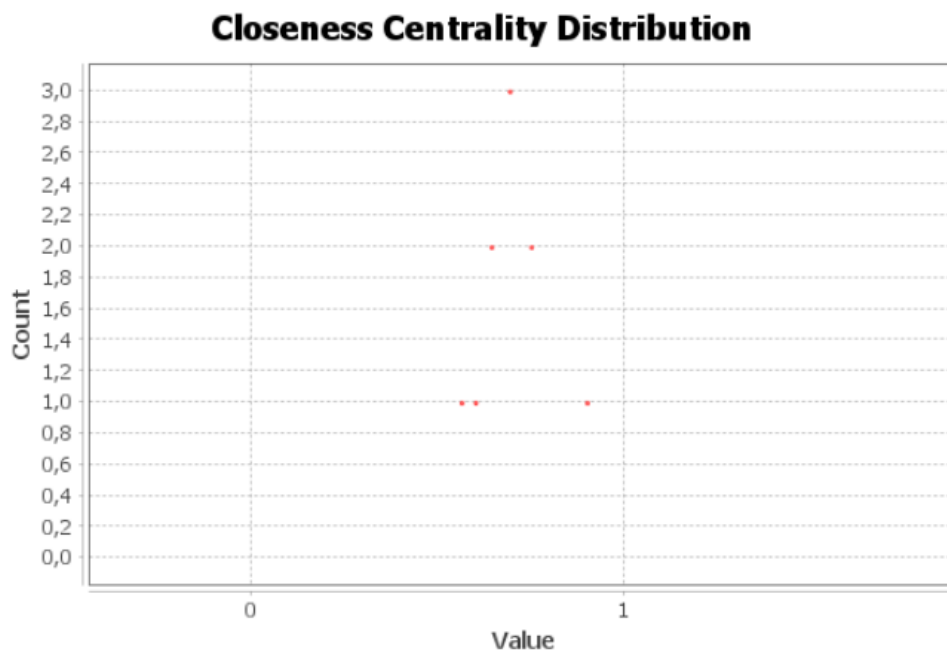


Figura 5.21: Gráfica de Cercanía.

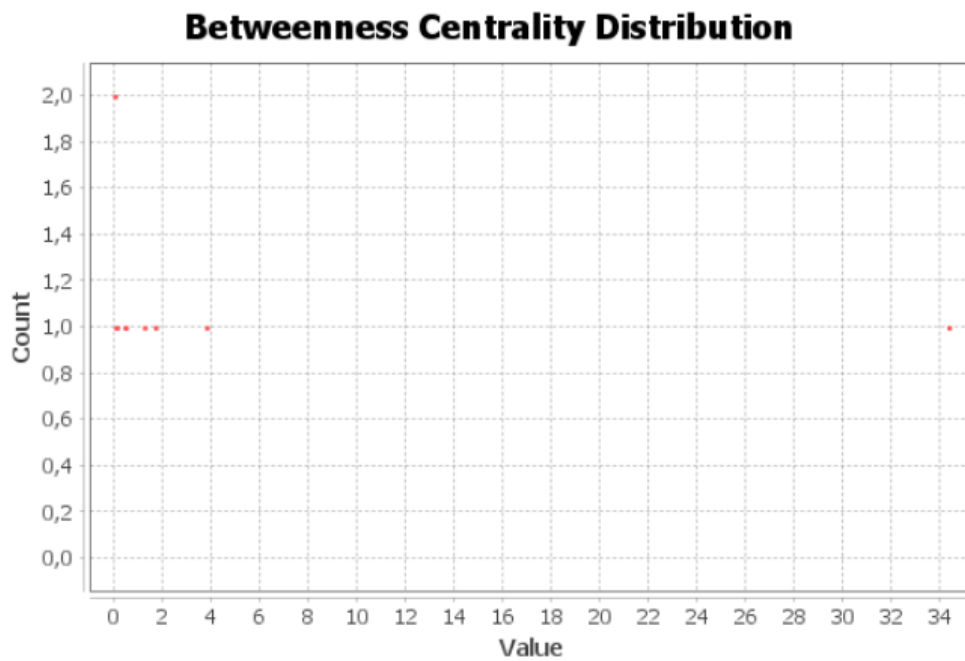


Figura 5.22: Gráfica de Intermediación.

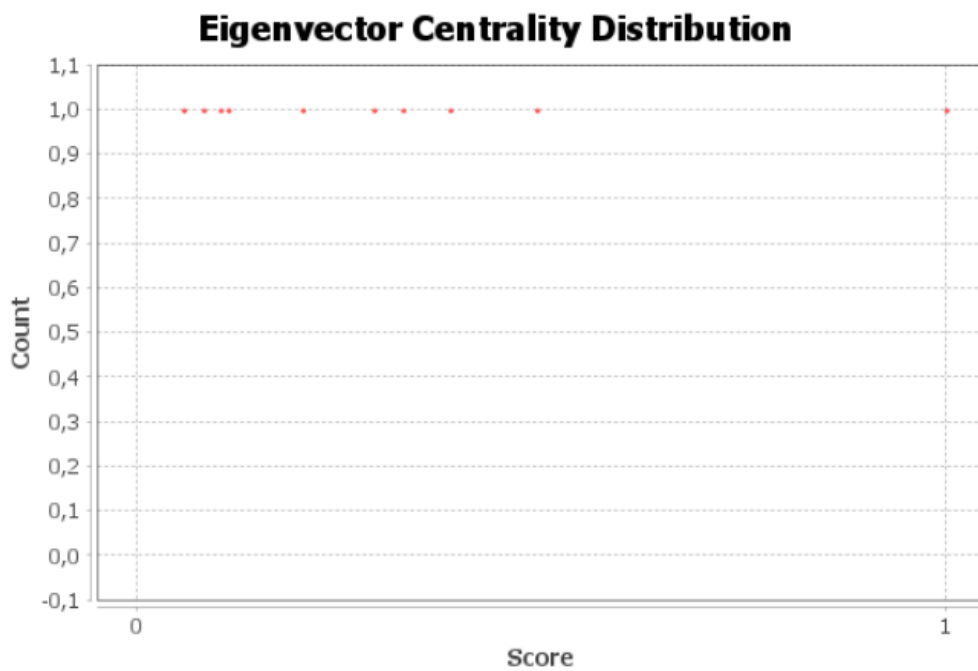


Figura 5.23: Gráfica de Cercanía de Vector Propio.



Figura 5.24: Gráfica de Modularidad.

### 5.2.3. Revisión

En la sección 1.4 se ha descrito la estrategia de evaluación entre pares y se ha hablado de las ventajas que aporta a la enseñanza online.

En las plataformas de cursos en línea que apoyan las tareas de OPA, los candidatos, es decir, los propios alumnos, son seleccionados al azar como revisores y se les pide que revisen el trabajo de otros compañeros de curso (Kao, 2013).

Se ha descrito en las dos secciones anteriores 5.2.1 y 5.2.2, cómo obtener información para lograr un Influence Diagram (ID) a través de los criterios de los expertos y de los resultados de SNA y evitar el azar en la selección. De momento, esta aplicación no crea este diagrama, pero sí permite exportar dicha información y utilizarla en aplicaciones externas como OpenMarkov.

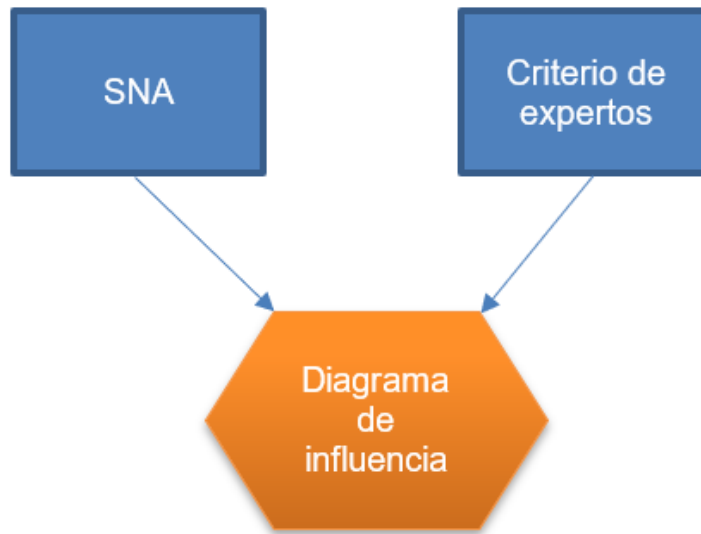


Figura 5.25: Esquema de diseño del método de asignación de revisores.

En esta sección, se ha diseñado para el usuario una asignación aleatoria de revisores con los siguientes apartados:

- Matriz de utilidades aleatorias.
- Matriz de  $n$  revisores.
- Matriz producto.
- Valor Utilidad Total.

Desde el menú superior de la aplicación, el usuario debe seleccionar la opción *Revisión*. El usuario va a encontrar un listado con las *Tareas de revisor* que hay cargadas en la base de datos. En la figura 5.26 se muestra un ejemplo.

Listado de tareas de revisor									
A continuación, se listan las tareas creadas por el usuario. De entre ellas debe elegir solo una (su identificador), para generar una matriz de $n$ revisores, siendo $n$ el número que se indica en cada tarea de revisión									
Identificador	Número de revisores	Función	ActivityCompatibility	Iniciativa	Actividad	Popularidad	Cercanía	Betweenness	Eigenvector
29	1	Lineal	1	2	3	4	5	6	7
28	7	Lineal	7	1	3	4	5	6	2
27	1	Lineal	1	2	3	4	5	6	7

Figura 5.26: Listado de tareas de revisor.

El listado que recoge la figura 5.26 es una ayuda para que el docente para que elija el identificador o id de una de estas tareas. El valor del campo *Número de revisores* va a ser utilizado para los resultados de este apartado.

En la figura 5.27 se muestra el desplegable que se le ofrece al usuario para escoger identificador *Tarea de revisor*:

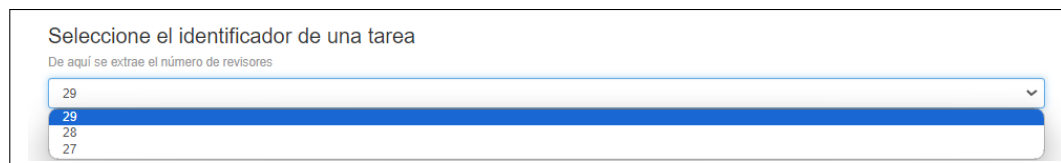


Figura 5.27: Selección de una tarea de revisor.

En OPA, un alumno revisará el trabajo de otro alumno, pero puede que también se quiera proponer que un alumno revise el trabajo de varios alumnos. A continuación, se ve con unos ejemplos:

Tabla 5.3: Ejemplo de tarea de 1 revisor.

	1	2	3	4
1			x	
2	x			
3				x
4		x		

En la tabla 5.3, puede verse una matriz en la que están representados los 4 posibles alumnos de un curso numerados del 1 al 4. En este primer ejemplo se contempla la opción de que un alumno solo revise el trabajo de otro alumno, de esta forma en la fila del alumno 1 vemos que este revisará el trabajo del alumno 3. El alumno 2 revisará el trabajo del alumno 1 y así sucesivamente. Se tiene en cuenta además dos aspectos importantes:

- Nunca un alumno puede revisarse así mismo, de ahí que está sombreada la diagonal.
- El trabajo de un alumno no puede ser revisado dos veces por un alumno.

En la tabla 5.4, se muestra una tabla de 2 revisores, donde en este caso un alumno evalúa el trabajo de 2 alumnos. Así, el alumno 1 revisará los trabajos de los alumnos 2 y 3. En este caso, también se tendrán en cuenta las reglas anteriormente mencionadas del ejemplo anterior.

Como puede observarse, la combinación de revisores aleatorios es amplia y puede aumentar exponencialmente cuando el número de alumnos crece.

Una vez propuesta en el desarrollo de esta aplicación esta opción de poder aceptar más de un revisor, el siguiente paso para el usuario es indicar al

Tabla 5.4: Ejemplo de tarea de 2 revisores.

	1	2	3	4
1		x	x	
2	x			x
3	x			x
4		x	x	

programa el número de pruebas o matrices que desea que el programa genere para encontrar una solución lo más óptima posible.

En el apartado 5.2.3 se explica con mayor detalle el uso de este valor. De momento, lo que se le pide al usuario es indicar el número de combinaciones posibles diferentes quiere el usuario que genere el programa, tal y como se muestra en la figura 5.28.

Número de matrices de n revisores

Indique cuántas Matrices Revisor diferentes quiere que el programa genere para las pruebas (Tenga en cuenta que un número elevado puede suponer elevado tiempo de cálculo)

Figura 5.28: Selección del número de matrices de n revisores.

Finalmente, una vez seleccionada la *Tarea de revisor* y el número de prueba o matrices que se desea que el programa genere, el usuario pulsará el botón *Crear matriz* para obtener los resultados finales que se describen a continuación.

### Matriz de utilidades aleatoria

Lo primero que se genera en el programa como resultado de *Revisión* es una matriz de  $N \times N$  a la que se llama *Matriz de utilidades aleatorias*, siendo  $N$  el número de estudiantes. La tabla muestra en los encabezados tanto vertical como horizontal los identificadores de cada estudiante. Como dice su propio nombre, esta matriz es creada por el programa automáticamente con valores aleatorios entre 0 y 100, en cuya diagonal siempre estará el valor 0.

Esta matriz se utiliza para obtener los resultados de la **Matriz producto** explicada más adelante.

En la figura 5.29, se observa una tabla o matriz de utilidades aleatorias de 10 estudiantes, el resto de información son valores aleatorios entre 0 y 100 con la diagonal siempre igual a 0.

Matriz de utilidades aleatorias										
Matriz generada con valores aleatorios entre 0 y 100										
	245	244	243	242	241	240	239	238	237	236
245	0	78	45	14	83	56	76	13	72	2
244	70	0	92	51	94	33	92	39	35	20
243	16	22	0	72	25	11	13	100	54	85
242	68	3	39	0	63	2	7	39	82	91
241	61	21	11	59	0	82	10	80	1	45
240	84	75	36	91	41	0	67	30	98	18
239	72	2	20	51	85	60	0	24	50	10
238	24	27	56	9	91	60	19	0	77	76
237	23	50	52	7	8	14	75	1	0	44
236	35	12	35	81	83	84	96	74	63	0

Figura 5.29: Matriz de utilidades aleatoria de 10 estudiantes.

### Matriz de N revisores

La *Matriz de N revisores* es una matriz también de tamaño  $N \times N$ , siendo  $N$  el número de estudiantes registrados en la base de datos.

Es la matriz resultado que ofrece el programa y cuya creación es la que se ha descrito en los ejemplos de las tablas 5.3 y 5.4.

Un valor 1 en una celda significa que el alumno de esa fila revisa la tarea del alumno de esa columna.

En la figura 5.30, se ve el resultado de una matriz devuelta por el programa de 10 estudiantes también y con 1 solo revisor. En esta figura se observa que el alumno 245 revisa el trabajo del alumno 239.

Matriz de n revisores										
Matriz generada con n revisores por fila y n revisores por columna. 1 significa que el nodo de esa fila revisa al de esa columna. Un mismo nodo no puede revisarse a sí mismo.										
	245	244	243	242	241	240	239	238	237	236
245	0	0	0	0	0	0	1	0	0	0
244	0	0	0	0	1	0	0	0	0	0
243	1	0	0	0	0	0	0	0	0	0
242	0	0	0	0	0	0	0	0	1	0
241	0	0	0	0	0	1	0	0	0	0
240	0	0	0	0	0	0	0	0	0	1
239	0	0	0	0	0	0	0	1	0	0
238	0	0	1	0	0	0	0	0	0	0
237	0	1	0	0	0	0	0	0	0	0
236	0	0	0	1	0	0	0	0	0	0

Figura 5.30: Matriz de N revisores.



Otro ejemplo es el de la figura 5.31, en él se ve el resultado de una matriz de la misma dimensión que la anterior, pero en este caso de 7 revisores.

**Matriz de n revisores**

Matriz generada con n revisores por fila y n revisores por columna. 1 significa que el nodo de esa fila revisa al de esa columna. Un mismo nodo no puede revisarse a sí mismo.

	245	244	243	242	241	240	239	238	237	236
245	0	1	1	1	1	1	1	0	1	0
244	0	0	1	1	0	1	1	1	1	1
243	1	1	0	0	1	1	1	0	1	1
242	1	1	0	0	1	1	1	1	0	1
241	1	1	1	1	0	1	1	1	0	0
240	1	1	0	0	1	0	1	1	1	1
239	0	0	1	1	1	1	0	1	1	1
238	1	0	1	1	1	1	0	0	1	1
237	1	1	1	1	1	0	0	1	0	1
236	1	1	1	1	0	0	1	1	1	0

Figura 5.31: Matriz de 7 revisores.

## Matriz producto

La *Matriz producto* es una matriz de la misma dimensión siempre que las 2 anteriores (matriz aleatoria y matriz de N revisores), donde las celdas contienen el producto de las mismas posiciones de las dos matrices anteriores.

A continuación, se muestran tres tablas que representan tres matrices, la matriz aleatoria 5.5, la matriz de N revisores 5.6 y la matriz producto 5.7, y se observa cómo se insertan valores en la última (matriz producto). En rojo se marca el valor 9 de la matriz aleatoria que multiplicado por el valor 1 de la celda que ocupa la misma posición de la matriz de N revisores, da como resultado un valor 9 en la misma posición de la matriz producto.

Continuando con el ejemplo de la aplicación, se muestra el resultado de la *Matriz producto* en la figura 5.32.

Tabla 5.5: Matriz aleatoria.

	1	2	3	4
1		2	9	4
2	4		5	7
3	8	10		11
4	3	7	1	

Tabla 5.6: ejemplo de matriz de 1 revisor.

	1	2	3	4
1			1	
2	1			
3				1
4		1		

Tabla 5.7: Ejemplo de resultado de matriz producto.

	1	2	3	4
1			9	
2	4			
3				11
4		7		

**Matriz producto**  
Matriz resultado de multiplicar las celdas que ocupan la misma posición respecto de las dos matrices anteriores

	245	244	243	242	241	240	239	238	237	236
245	0	0	0	0	0	0	76	0	0	0
244	0	0	0	0	94	0	0	0	0	0
243	16	0	0	0	0	0	0	0	0	0
242	0	0	0	0	0	0	0	0	82	0
241	0	0	0	0	0	82	0	0	0	0
240	0	0	0	0	0	0	0	0	0	18
239	0	0	0	0	0	0	0	24	0	0
238	0	0	56	0	0	0	0	0	0	0
237	0	50	0	0	0	0	0	0	0	0
236	0	0	0	81	0	0	0	0	0	0

Figura 5.32: Matriz Producto.

En la figura 5.33, se muestra el resultado de la matriz producto generada de la anterior matriz de 7 revisores:

**Matriz producto**  
Matriz resultado de multiplicar las celdas que ocupan la misma posición respecto de las dos matrices anteriores

	245	244	243	242	241	240	239	238	237	236
245	0	37	21	32	60	87	29	0	82	0
244	0	0	12	75	0	70	34	6	85	81
243	55	16	0	0	18	64	11	0	96	74
242	28	19	0	0	52	10	21	74	0	14
241	2	43	33	12	0	40	25	40	0	0
240	30	35	0	0	63	0	68	63	95	69
239	0	0	40	1	100	66	0	80	30	58
238	49	0	5	82	51	30	0	0	89	76
237	11	9	29	83	54	0	0	28	0	56
236	85	70	54	56	0	0	69	85	84	0

Figura 5.33: Matriz Producto de 7 revisores.

### Valor Utilidad Total

Finalmente, el programa genera un valor llamado **Utilidad Total**, un dato numérico cuyo valor oscila entre 0 y 100 y cuyo resultado se obtiene con la fórmula 5.1.

$$UtilidadTotal = \frac{SumaValoresMatrizProducto}{NumeroEstudiantes * NumeroRevisores} \quad (5.1)$$

Siguiendo el ejemplo anterior de una matriz de 4 estudiantes (tabla 5.8), se calcula el valor *Utilidad Total* tal y como indica la fórmula 5.2.

Tabla 5.8: Ejemplo de resultado de matriz producto para cálculo de Utilidad Total.

	1	2	3	4
1			9	
2	4			
3				11
4		7		

$$UtilidadTotal = \frac{9 + 4 + 11 + 7}{4 * 1} = 7,75 \quad (5.2)$$

Este valor *Utilidad Total*, se calcula tantas veces como se ha indicado en el campo *Número de matrices de n revisores* por el usuario (5.28). En el ejemplo que mostraba esta figura, el usuario ha seleccionado solo generar una matriz, de forma que el programa solo ha calculado un único valor *Utilidad Total*. Si este valor hubiese sido 10 en lugar de 1, el programa hubiese generado internamente 10 *Matrices de n revisores* distintas, siempre utilizando la **misma *Matriz aleatoria***, obteniendo por tanto 10 valores *Utilidad Total* distintos. La aplicación muestra finalmente la *Matriz de n revisores* que mayor valor *Utilidad Total* haya logrado.

Es lógico pensar que un número elevado en el campo *Número de matrices de n revisores*, va a generar un mejor resultado, y aunque es cierto, el problema puede darse en el tiempo que le lleve al ordenador lograr tales cálculos. Por tanto, dependerá del equipo, del número de revisores o de la dimensión del grafo, el hecho de que el tiempo de cálculo de este valor sea mayor o menor.

En la aplicación, el usuario obtiene el valor *Utilidad Total* como se muestra en la figura 5.34.



Figura 5.34: Valor *Utilidad Total* que muestra la aplicación.

# Capítulo 6

## Conclusiones y trabajos futuros

### 6.1. Conclusiones

En este trabajo se ha continuado con la implementación de una aplicación que facilite la labor de la enseñanza a los docentes a la hora de organizar a los alumnos para llevar a cabo tareas de evaluación entre los estudiantes.

Para conocer el punto de partida para el desarrollo de este trabajo, se ha llevado a cabo un estudio a base de una lectura de documentación (Objetivo 1*a*) y revisión de código fuente (Objetivo 1*b*) que ha permitido establecer un punto de partida para implantar nuevos requisitos. Como consecuencia del trabajo previo, se han mantenido ciertas herramientas que desconocía como Spring Boot o Thymeleaf para el diseño de la aplicación.

Una vez conocido el alcance del antiguo proyecto, se observa que hay que implementar en el programa análisis de redes sociales (Objetivo 2), para lo cual se ha tenido que retomar teoría de grafos y profundizar en el uso de la herramienta Gephi. Además de esto, se investiga en cómo pueden integrarse las APIs de esta herramienta con el IDE utilizado y conocer las funciones interface a las que era necesario llamar (Objetivos 2*b* y 2*c*). El resultado final, ha sido mayor del esperado ya que estas librerías han permitido generar, además de las métricas esperadas, más información útil para el usuario como son los grafos o gráficas.

Apoyado en el diseño de la aplicación inicial, se crea el apartado para que el docente pueda guardar en una base de datos información resultado de las métricas de análisis de redes sociales (Objetivo 3*a*) y disponer el sistema para poder utilizar esta información cuando se integren librerías para la creación de diagramas de influencia (Objetivo 3*c*). La OPA que se diseña para el usuario es aleatoria (Objetivo 3*c*), donde se ofrece información suficiente de cómo se genera y además permite al usuario tomar estos datos y exportarlos

a tablas.

Por último, para la presentación de este trabajo, se han utilizado tres herramientas que se desconocían, pero que han permitido, por un lado, desarrollar este documento ((Overleaf, 2024)) (Objetivo 5) y por otro, preparar una aplicación que permita la modularidad (Spring, 2024b) (Objetivo 4a) y que sea exportable (Docker, nown) (Objetivo 4b).

## 6.2. Trabajos futuros

Teniendo en cuenta que esta aplicación está diseñada para un tipo de enseñanza online donde han sido utilizados los datos de los foros, para aplicar en ellos análisis de redes sociales, podría implementarse como futura mejora el análisis de los mensajes intercambiado entre los estudiantes y no quedarse solo con el número de mensajes intercambiados.

Gracias al criterio de selección de los docentes y al resultado del análisis de redes sociales, se puede crear un diagrama de influencia que es la información que finalmente dice qué alumno es recomendable que analice el trabajo de otro alumno y cuya implementación no ha sido posible en este trabajo, por tanto tiene cabida como mejora, integrar este apartado.

Por otro lado, el método propuesto en este proyecto de asignación de revisores basado en el análisis social podría adaptarse a otros criterios, ya que estos pueden variar según el contexto educativo o la experiencia de los docentes. Esto podría aplicarse también a los indicadores utilizados (iniciativa, actividad, centralidad, etc.) teniendo en cuenta que la API de Gephi ofrece la opción de obtener otras métricas no implementadas aquí. Además, podría adaptarse no solo a actividades de evaluación entre pares sino a cualquier actividad colaborativa que surja.

Se ha tratado de desarrollar una aplicación sencilla en uso para el usuario, y una posible mejora es poder realizar un diseño centrado en el usuario tal y como se mostró en la asignatura impartida en este máster *Diseño Centrado en el Usuario de Sistemas Informáticos*, teniendo en cuenta que detrás de la aplicación desarrollada puede haber usuarios que necesiten adaptaciones especiales para su facilidad de uso. Además, tampoco se ha tenido en cuenta en este diseño el uso de otros dispositivos que no sea un ordenador personal, por tanto, también es parte del diseño poder mejorar extensibilidad también a dispositivos móviles por ejemplo.

En cuanto a la arquitectura del proyecto, se ha continuado con un diseño modular y extensible para que pueda seguir creciendo en un futuro, ya que podrían ser integradas herramientas que permitan la creación de modelos gráficos probabilistas como por ejemplo OpenMarkov al igual que se ha hecho

con Gephi.

Finalmente, aunque se ha marcado como objetivo pero finalmente no se ha cumplido, un trabajo a futuro sería la integración de una librería que permita la creación de plantilla para diagramas de influencia que generen redes de diagramas de influencia y poder así comparar los resultados de una asignación basada en los criterios de un experto y análisis de foros con la asignación aleatoria.





# Bibliografía

- Aggarwal, C. C. (2011). An introduction to social network data analytics. In Aggarwal, C. C., editor, *Social Network Data Analytics*, page 1–15. Springer US, Boston, MA.
- Anaya, A. R., Luque, M., Letón, E., and del Olmo, F. H. (2019). Automatic assignment of reviewers in an online peer assessment task based on social interaction. *Expert Systems*, 36:e12405.
- Arias, M. and Díez, F. J. (2008). Carmen: An open source project for probabilistic graphical models. In Jaeger, M. and Nielsen, T. D., editors, *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM'08)*, page 25–32, Hirtshals, Denmark.
- Arias, M., Pérez-Martín, J., Luque, M., and Díez, F. J. (2019). OpenMarkov, an open-source tool for probabilistic graphical models. In Kraus, S., editor, *Proceedings of the Twenty-eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, page 6485–6487, Macao, China. Morgan Kaufmann.
- Contributors, W. (2024). Model-view-controller. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller#Bibliography>(accessed January 31, 2024). Online; accessed 23-January-2024.
- Data, S. (2024). Jpa repository. <https://docs.spring.io/spring-data/jpa/docs/current/api/org/springframework/data/jpa/repository/JpaRepository.html>. Online; accessed 05-February-2024.
- Dawson, S. (2008). A study of the relationship between student social networks and sense of community. *Educational Technology and Society*, 11:224–238.
- De-Marcos, L., García-López, E., García-Cabot, A., Medina-Merodio, J. A., Domínguez, A., Martínez-Herráiz, J. J., and Díez-Folledo, T. (2016). Social

- network analysis of a gamified e-learning course: Small-world phenomenon and network metrics as predictors of academic performance. *Computers in Human Behavior*, 60:312–321.
- Deitel, P. (2017). Understanding java 9 modules. <https://www.oracle.com/corporate/features/understanding-java-9-modules.html>. Online; accessed 31-January-2024.
- Docker (publication date unknown). Docker. <https://www.docker.com/>. Online; accessed 15-February-2024.
- Eclipse (publication date unknown). Eclipse ide. <https://eclipseide.org/>. Online; accessed 05-February-2024.
- Foundation, M. (2024). Mariadb. <https://mariadb.org/>. [Online; accessed 05-February-2024].
- Gephi (2024a). Gephi. <https://gephi.org/toolkit/>. Online; accessed 17-January-2024.
- Gephi (2024b). Gml format. <https://gephi.org/users/supported-graph-formats/gml-format/>. [Online; accessed 02-February-2024].
- Git (2024). Git. <https://git-scm.com/downloads>. [Online; accessed 16-February-2024].
- GitHub (2024). Github. <https://github.com/>. [Online; accessed 05-February-2024].
- GitLFS (2024). Gitlfs. <https://git-lfs.com/>. [Online; accessed 16-February-2024].
- Hanrahan, S. J. and Isaacs, G. (2001). Assessing self-and peer-assessment: The students' views. *Higher Education Research & Development*, 20:53–70.
- Howard, R. A. and Matheson, J. E. (1984). Influence diagrams. In Howard, R. A. and Matheson, J. E., editors, *Readings on the Principles and Applications of Decision Analysis*, page 719–762. Strategic Decisions Group, Menlo Park, CA.
- Kao, G. Y.-M. (2013). Enhancing the quality of peer review by reducing student “free riding”: Peer assessment with positive interdependence. *British Journal of Educational Technology*, 44:112–124.

- Liu, Z., Kang, L., Domanska, M., Liu, S., Sun, J., and Fang, C. (2018). Social network characteristics of learners in a course forum and their relationship to learning outcomes. In *CSEdu (1)*, page 15–21.
- Martín, A. (2022). Aplicación para la realización de minería de datos sobre datos educativos de la uned. Master's thesis, E.T.S Informática, UNED, Madrid, Spain.
- Maven (2024). Apache maven project. <https://maven.apache.org/>. Online; accessed 17-January-2024.
- Meek, S. E. M., Blakemore, L., and Marks, L. (2016). Is peer review an appropriate form of assessment in a MOOC? Student participation and performance in formative peer review. *Assessment & Evaluation in Higher Education*, 42:1000–1013.
- MockFlow (2024). Mockflow. <https://mockflow.com/>. Online; accessed 05-February-2024.
- MySQL (2024a). Bringing mysql to the web. <https://www.phpmyadmin.net/>. Online; accessed 05-February-2024.
- MySQL (2024b). Mysql. <https://www.mysql.com/>. Online; accessed 05-February-2024.
- OpenWebinars (2024). Openwebinars. <https://openwebinars.net/>. Online; accessed 05-February-2024.
- Overleaf (2024). Overleaf. <https://es.overleaf.com/>. Online; accessed 17-January-2024.
- Saqr, M., Fors, U., Tedre, M., and Nouri, J. (2018). How social network analysis can be used to monitor online collaborative learning and guide an informed intervention. *PLoS One*, 13:e0194777.
- Saqr, M., Nouri, J., Vartiainen, H., and Malmberg, J. (2020). What makes an online problem-based group successful? a learning analytics study using social network analysis. *BMC Medical Education*, 20:1–11.
- Saxena, A., Saxena, P., Reddy, H., and Gera, R. (2019). A survey on studying the social networks of students. *arXiv preprint arXiv:1909.05079*.
- Scott, J. (2017). *Social network analysis*. Sage.

- Sinha, T. (2014). Together we stand, together we fall, together we win: Dynamic team formation in massive open online courses. In *Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the*, page 107–112. IEEE.
- Spring (2024a). Spring. <https://spring.io/>. Online; accessed 17-January-2024.
- Spring (2024b). Spring tool 4. <https://spring.io/tools>. Online; accessed 05-February-2024.
- Sánchez, J. M. (2024). Repositorio del trabajo final de máster de ingeniería informática de juan manuel sánchez villarejo. xxx. Online; accessed 16-February-2024.
- Thymeleaf (2024). Thymeleaf documentation. <https://www.thymeleaf.org/documentation.html>. Online; accessed 17-January-2024.
- UNED (2022). La uned estrena plataforma educativa virtual. <https://comunicacion.uned.es/news/show/86809/la-uned-estrena-plataforma-educativa-virtual.html>. Online; accessed 28-July-2023.
- UNED (2024a). Portal estadístico - uned. <https://app.uned.es/evacaldos/>. Online; accessed 19-December-2023.
- UNED (2024b). Regulación de los trabajos de fin de máster en las enseñanzas conducentes al título oficial de máster de la uned. [https://descargas.uned.es/publico/pdf/guias/posgrados/normativa/TRABAJO\\_FIN\\_DE\\_MASTER.pdf](https://descargas.uned.es/publico/pdf/guias/posgrados/normativa/TRABAJO_FIN_DE_MASTER.pdf). Online; accessed 23-January-2024.
- UNED (2024c). Uned - conócenos. <https://www.uned.es/universidad/inicio/conocenos.html>. Online; accessed 19-December-2023.
- Uto, M. and Ueno, M. (2016). Item response theory for peer assessment. *IEEE Transactions On Learning Technologies*, 9:157–170.
- Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and Applications*, volume 8. Cambridge University Press.

# Apéndice A

## Instalación y acceso a la aplicación

Como se ha comentado en el apartado 2.6, se puede descargar el proyecto mediante un cliente git de línea de comandos con la sentencia

```
git clone https://github.com/juanmanuelanvi/tfm-jsv.git
```

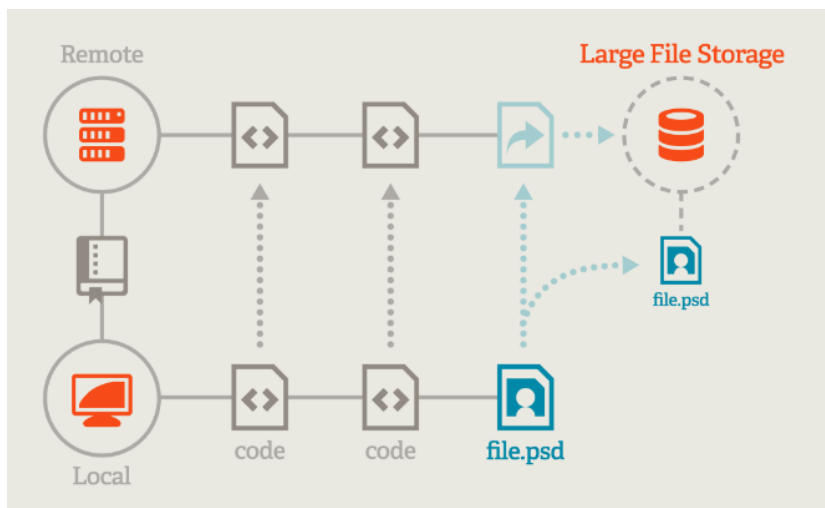


Figura A.1: Git LFS.

Si se descarga directamente desde GitHub, el fichero uned-educa-analisis-0.0.2-SNAPSHOT.jar no será correctamente descargado, ya que al ocupar más de 100 MB realmente se aloja en otro servidor siguiendo la filosofía

observada en la figura A.1 <sup>1</sup>, donde puede verse que los datos de mayor peso, son almacenados realmente en otro servidor distinto nombrado como *Large File Storage*, guardando en GitHub una llamada a este fichero.

Si no se dispone de la herramienta git para la clonación, podrá instalarla con la aplicación gratuita (Git, 2024).

La aplicación funciona a través de contenedores, para lo cual se aconseja tener instalado en su equipo una aplicación que permita trabajar con ellos como es (Docker, nown).

Una vez clonado el trabajo, conviene chequear dos cosas:

1. El puerto para el acceso a la base de datos es el 3306. Por tanto, se aconseja que esté libre.
2. El puerto para el acceso a la aplicación es el 8090. Por tanto, también se aconseja que esté desocupado.

Para lanzar la aplicación al contenedor, únicamente debe ejecutarse el fichero *instalar.bat* y una vez cargada la aplicación en el contenedor, el usuario verá que tanto la base de datos (mariadb) como la aplicación (uned-educa-analisis) habrán sido cargadas.

Se aconseja ejecutar en docker primero la base de datos y una vez establecida, arrancar la aplicación. En la figura A.2 se ve un ejemplo del correcto resultado.

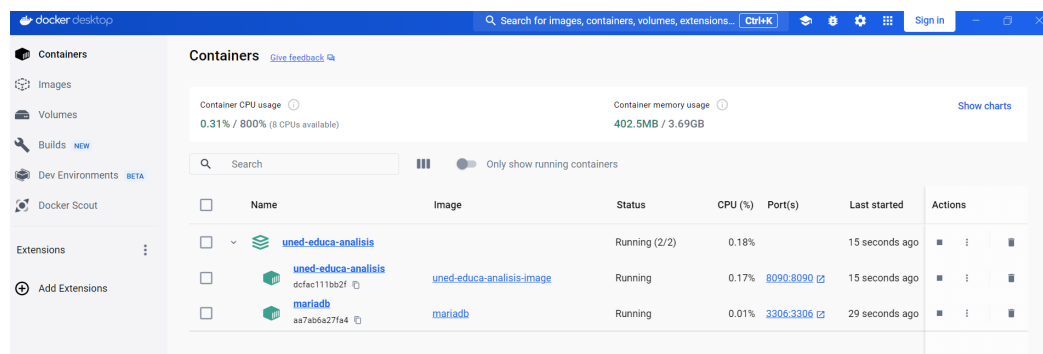


Figura A.2: Contenedor con la aplicación y la base de datos corriendo.

Para utilizar la aplicación, debe estar ejecutado el contenedor anterior y una vez hecho, puede abrirse un navegador cualquiera y acceder a ellas a través de *localhost:8090*, tal y como se muestra en la figura A.3.

<sup>1</sup>Figura tomada de (GitLFS, 2024) Git Large File Storage

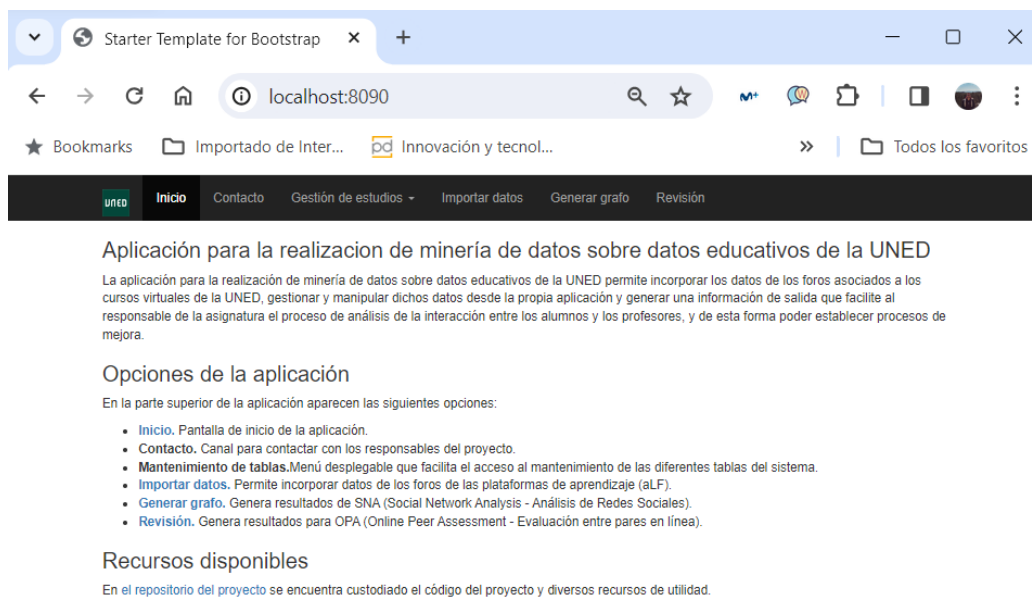


Figura A.3: Ejemplo de la aplicación desde navegador.

Por último, indicar que para el desarrollo de este proyecto se han utilizado herramientas respetando en todas esas su licencia de uso:

- **Maven:** A través de Maven se han podido añadir todas las dependencias que ha necesitado el programa y las cuales pueden verse en el fichero pom.xml.
- **Gephi:** Su toolkit ha permitido el cálculo de análisis de redes sociales así como la generación de grafos y gráficas.