



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Una aplicación web para la formación de equipos de
estudiantes

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Alfaro Moscardó, Daniel

Tutor/a: Alberola Oltra, Juan Miguel

Cotutor/a: Sánchez Anguix, Víctor

CURSO ACADÉMICO: 2023/2024

Resumen

Este Trabajo de Fin de Grado propone el desarrollo de una aplicación web para facilitar la agrupación de alumnos en proyectos utilizando la metodología Belbin, que clasifica a los individuos en roles basados en sus fortalezas y debilidades. La herramienta está diseñada para asistir a profesores y alumnos en la formación de equipos efectivos, mejorando así el aprendizaje colaborativo en el entorno educativo. La implementación de la herramienta se realiza mediante Angular 17, Nodejs y MongoDB, una combinación de tecnologías que buscan proporcionar una interfaz intuitiva y una experiencia de usuario eficiente, asegurando que los profesores puedan gestionar sus actividades y grupos de manera flexible y adaptada a las necesidades de cada profesor.

Palabras clave: Trabajo en equipo, Metodología Belbin, Agrupación de alumnos, Herramienta web, Aprendizaje colaborativo, Desarrollo de software, Angular, Node.js, MongoDB.

Abstract

This Bachelor's Thesis proposes the development of a web tool to facilitate the grouping of students in projects using the Belbin methodology, which classifies individuals into roles based on their strengths and weaknesses. The tool is designed to assist teachers and students in forming effective teams, thus enhancing collaborative learning in the educational environment. The implementation of the tool involves a combination of modern frontend and backend technologies, and a matching algorithm programmed in Python. The application aims to provide an intuitive interface and an efficient user experience, ensuring that teachers can manage activities and groups flexibly and tailored to students' needs.

Keywords: *Teamwork, Belbin Methodology, Student Grouping, Web Tool, Collaborative Learning, Software Development, Angular, Node.js, MongoDB.*

Índice general

Resumen	I
Índice general	II
Índice de figuras	IV
1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	2
2. Marco Teórico	3
2.1. Trabajo en equipo	3
2.2. Métodos de Agrupación	4
3. Análisis del problema	8
3.1. Propósito	8
3.2. Casos de uso	9
4. Propuesta	11
4.1. Tecnologías y frameworks	11
4.2. Funcionalidades de la Aplicación	14
5. Diseño	16
6. Desarrollo de la propuesta	21
6.1. Configuración del Entorno de Desarrollo	21
6.2. Desarrollo del Backend	21
6.3. Desarrollo del Frontend	26
7. Conclusiones	38
7.1. Trabajos futuros	42

Bibliografía

43

Índice de figuras

2.1. Roles de la metodología Belbin	5
2.2. Rasgos de personalidad según la metodología MBTI	7
3.1. Casos de uso para un Usuario anónimo	9
3.2. Casos de uso para un Alumno	9
3.3. Casos de uso para un Profesor	10
5.1. Mockups iniciales de la aplicación	17
5.2. Relación entre las tecnologías utilizadas	19
6.1. Diagrama de la aplicación en Angular	22
6.2. Diagrama de la aplicación	26
6.3. Registro del profesor	27
6.4. Iniciar sesión	28
6.5. Olvido de contraseña	28
6.6. Dashboard del profesor	29
6.7. Crear una actividad	29
6.8. Detalle de una actividad	30
6.9. Formulario para añadir estudiantes a la actividad	30
6.10. Estudiante pendiente	31
6.11. Grupos creados en la actividad	32
6.12. Invitación por correo electrónico	33
6.13. Enlace que recibe un estudiante.	33
6.14. Registro del alumno	33
6.15. Login del alumno	34
6.16. Dashboard del estudiante	34
6.17. Enter Caption	35
6.18. Notificación del estudiante	35
6.19. Detalle de actividad del estudiante	36

6.20. Grupos creados en la actividad	36
6.21. Detalle de grupo del estudiante	37

1 Introducción

El **trabajo en equipo** es una habilidad fundamental en el ámbito educativo y profesional, ya que permite a los individuos combinar sus fortalezas y habilidades para alcanzar objetivos comunes. En el contexto universitario, los proyectos colaborativos se han convertido en una práctica común, brindando a los estudiantes la oportunidad de aplicar de manera conjunta sus conocimientos y habilidades. Sin embargo, la formación de grupos efectivos presenta numerosos desafíos para los profesores, quienes deben considerar diversos factores como las habilidades, intereses y personalidades de los estudiantes.

La dificultad de **crear equipos equilibrados y eficaces** radica en la necesidad de equilibrar habilidades complementarias y diversidad de perspectivas, garantizando que cada miembro del grupo pueda contribuir de manera significativa al proyecto. Los **métodos tradicionales** de formación de equipos a menudo resultan insuficientes para satisfacer estas necesidades complejas, lo que puede afectar negativamente tanto el proceso de aprendizaje como la calidad de los proyectos realizados.

El presente Trabajo de Fin de Grado propone el desarrollo de una **aplicación web** que utilice diversas metodologías para automatizar el proceso de agrupación de alumnos en proyectos universitarios, optimizando así la formación de equipos, potenciar el aprendizaje colaborativo y ofrecer una solución integral y accesible para mejorar la dinámica de trabajo en equipo en el entorno universitario.

1.1 Motivación

Este concepto de formar equipos más eficientes que los creados mediante métodos aleatorios es una idea que he tenido en mente durante algún tiempo y quería aprovechar mis conocimientos para crear una aplicación capaz de llevarlo a cabo. Además, actualmente existe la necesidad real de una herramienta que automatice y facilite este proceso para los profesores, quienes, debido a sus múltiples responsabilidades, se beneficiarían enormemente de una aplicación que no solo ahorre tiempo, sino que también optimice la formación de equipos, asegurando que cada grupo tenga el equilibrio necesario para trabajar eficientemente. Con esto, los alumnos se beneficiarían al ser parte de equipos donde sus habilidades son complementarias y bien aprovechadas, incrementando así su motivación y rendimiento en los proyectos. Estas son mejoras que me hubiera gustado tener durante mi carrera académica y que, aunque se han intentado implementar, todavía no existía una manera fácil de lograrlo.

1.2 Objetivos

Desarrollar una herramienta web diseñada para facilitar el proceso de agrupación de alumnos en proyectos colaborativos utilizando la metodología Belbin. La aplicación tiene como objetivo principal ayudar tanto a profesores como a estudiantes en la formación de equipos equilibrados y efectivos, optimizando así el aprendizaje y el rendimiento en proyectos académicos.

- Creación de una aplicación web para la creación y gestión de grupos de estudiantes en proyectos académicos. Los profesores podrán crear actividades, añadir alumnos y utilizar un algoritmo de emparejamiento para formar grupos equilibrados.
- Obtener una aplicación intuitiva, que permita los profesores gestionar de manera eficiente la creación y modificación de equipos de trabajo en sus asignaturas.
- Integración del algoritmo de emparejamiento de grupos proporcionado, implementando un sistema de colas para gestionar el procesamiento asíncrono del mismo.
- Proveer a los alumnos una plataforma donde puedan visualizar en qué actividades y grupos están asignados, promoviendo la transparencia y la organización. La herramienta permitirá a los alumnos unirse a actividades mediante invitaciones enviadas por los profesores.

2 Marco Teórico

En este capítulo se exploran los conceptos fundamentales del trabajo en equipo y los métodos de agrupación en contextos educativos. Se analiza cómo la colaboración entre individuos con habilidades y roles complementarios puede impulsar el logro de objetivos comunes, fomentando el desarrollo de competencias personales y profesionales. Además, se examinan las características esenciales del trabajo en equipo eficaz, incluyendo la interdependencia, la responsabilidad compartida y la sinergia, así como las habilidades clave necesarias para su éxito. Finalmente, se presentan y evalúan diversos métodos de agrupación, destacando sus ventajas y desventajas en función del contexto y los objetivos del equipo.

2.1 Trabajo en equipo

El trabajo en equipo puede definirse como un proceso colaborativo en el que un grupo de individuos con habilidades y roles complementarios se unen para alcanzar un objetivo común. Este enfoque es esencial en diversos contextos, especialmente en el educativo, donde fomenta el desarrollo de competencias personales y profesionales entre los estudiantes. El trabajo en equipo permite una mayor sinergia, creatividad y resolución de problemas, superando muchas veces las capacidades individuales. (Gündüz, 2008)

Como se ha comentaba en la sección anterior, los proyectos colaborativos en el ámbito académico son cada vez más frecuentes, promueven el aprendizaje activo y permiten a los alumnos poner en común sus conocimientos con sus compañeros. Esto se consigue creando un ambiente que minimice los obstáculos a los que estudiantes se deban enfrentar trabajando en grupo, para ello es necesario crear equipos eficaces, teniendo en cuenta la compatibilidad de personalidades y una distribución equilibrada de habilidades, los cuales se ha demostrado que estos equipos acaban siendo más exitosos que los generados con otros tipos de métodos (Alberola et al., 2016)

Las características esenciales del trabajo en equipo incluyen la interdependencia, la responsabilidad compartida y la sinergia en la contribución de cada miembro. El desarrollo de un equipo atraviesa diversas etapas, conocidas como formación, tormenta, normativización, desempeño y disolución, cada una con sus propios desafíos y dinámicas. (Bonebright, 2010)

También son necesarias ciertas habilidades clave para que el trabajo en equipo sea eficaz, estas son la comunicación clara y efectiva, la capacidad de colaboración, la resolución de conflictos y el liderazgo. (Conde et al., 2016) La comunicación permite la alineación de objetivos y la comprensión mutua. La colaboración fomenta la cohesión y el apoyo entre miembros; la resolución de conflictos facilita la superación de desacuerdos constructivamente, y el liderazgo guía al equipo hacia el cumplimiento de sus metas. Además, el trabajo en equipo efectivo requiere que los miembros del grupo tengan de manera individual habilidades interpersonales, como la empatía, la comunicación asertiva y la resolución de conflictos.

2.2 Métodos de Agrupación

Existen diversos métodos para la formación de equipos, cada uno con sus propias ventajas y desventajas. (Del Val et al., 2014). Los métodos más comunes incluyen:

- **Asignación Aleatoria:** Los equipos se forman al azar, lo que puede fomentar la diversidad pero no garantiza un equilibrio en habilidades y personalidades.
- **Selección por Afinidad:** Los estudiantes eligen sus propios equipos basándose en afinidades personales, lo cual puede mejorar la cohesión pero podría resultar en una falta de diversidad de habilidades y perspectivas.
- **Agrupación Basada en Roles:** Se forman equipos considerando las habilidades individuales de los estudiantes, buscando un balance que maximice el rendimiento del grupo.
- **Agrupación por Personalidad:** Se forman equipos teniendo en cuenta los rasgos de personalidad de los estudiantes, con el objetivo de optimizar la dinámica de grupo y la colaboración efectiva.

Cada método presenta ventajas y desventajas que deben ser evaluadas en función del contexto y los objetivos específicos del equipo. A continuación, se detallan algunas de las metodologías más reconocidas.

2.2.1 Metodología BELBIN®

La metodología Belbin es un enfoque desarrollado para analizar los roles que las personas asumen en equipos de trabajo. Esta metodología se centra en identificar las fortalezas y debilidades de los miembros de un equipo y cómo estas pueden afectar la dinámica y el rendimiento del equipo en su conjunto (Batenburg et al., 2013). Esta clasifica a los individuos en nueve roles principales.

Coordinador: Lidera y motiva al equipo.

Investigador de recursos: Busca información y recursos para el proyecto.

Implementador: Convierte las ideas en acción.

Evaluador: Analiza las ideas y toma decisiones.

Especialista: Aporta conocimiento y experiencia en un área específica.

Monitor-evaluador: Observa y evalúa el progreso del equipo.

Finalizador: Se asegura de que el proyecto se complete a tiempo y con calidad.

Promotor: Entusiasma y motiva al equipo.

Colaborador: Busca el consenso y la armonía en el equipo.



Figura 2.1: Roles de la metodología Belbin

Según la tesis de Joan Anton Ros Guash (2006), se destacan dos importantes contribuciones en el análisis de Belbin:

- (I) Se reconoce que las fortalezas humanas suelen estar acompañadas de ciertas debilidades.
- (II) La combinación de estos roles aumenta significativamente la probabilidad de formar un equipo exitoso en comparación con otras alternativas. Esto se debe a que algunos equipos pueden lograr una productividad complementaria, mientras que otros podrían caer en una competencia interna.

Evaluación de Roles

La metodología Belbin incluye herramientas para evaluar los roles naturales de los miembros del equipo. Esto se hace mediante cuestionarios que los individuos y sus compañeros completan para proporcionar una visión completa de cómo se perciben a sí mismos y cómo los ven los demás.

2.2.2 Indicador de Myers-Brigg

El Myers-Briggs Type Indicator (MBTI, 1917) es otro indicador popular utilizado para la agrupación de equipos basada en la personalidad. Desarrollado a partir de la teoría de tipos psicológicos de Carl Jung, el MBTI clasifica a los individuos en 16 tipos de personalidad basados en cuatro dimensiones:

Extraversión (E) vs. Introversión (I): Indica la fuente de energía y enfoque de atención de una persona.

Sensación (S) vs. Intuición (N): Describe cómo una persona percibe la información.

Pensamiento (T) vs. Sentimiento (F): Indica cómo una persona toma decisiones.

Juicio (J) vs. Percepción (P): Describe el enfoque de una persona hacia el mundo exterior, ya sea organizado o adaptable.



Figura 2.2: Rasgos de personalidad según la metodología MBTI

Cada combinación de estas dimensiones produce un tipo de personalidad único que puede influir en cómo una persona trabaja en equipo, se comunica y resuelve problemas.

Utilizar el test MBTI para la formación de equipos puede ayudar a equilibrar las personalidades dentro de un mismo grupo, mejorando la comunicación y optimizando la colaboración de los integrantes.

3 Análisis del problema

Al abordar el desarrollo de los diversos componentes de un proyecto, se dispone de una amplia gama de soluciones. En este contexto, es esencial investigar y comprender las distintas alternativas para tratar cada aspecto y determinar las estrategias de resolución adecuadas. En el presente capítulo, se examinarán requisitos en los cuales se fundamentar la aplicación y los casos de uso de esta.

3.1 Propósito

- **Facilitar la Tarea de los profesores:** Permitir a los profesores gestionar de manera eficiente la creación y modificación de equipos de trabajo en sus asignaturas. Los profesores podrán crear actividades, añadir alumnos y utilizar el algoritmo de emparejamiento basado en la metodología Belbin para formar grupos equilibrados. Además, tendrán la capacidad de realizar ajustes manuales a los grupos generados automáticamente.
- **Optimizar la Formación de equipos:** Utilizar un algoritmo que asocie a los alumnos en función de sus roles principales según la metodología Belbin, garantizando que cada equipo cuente con una combinación equilibrada de habilidades y personalidades. Con esto, se espera que mejore la cohesión del equipo y aumente la probabilidad de éxito en los proyectos.
- **Fomentar el Aprendizaje colaborativo:** Al facilitar la formación de equipos equilibrados, se busca potenciar el aprendizaje colaborativo, permitiendo a los alumnos experimentar y desarrollar habilidades de trabajo en equipo que son esenciales tanto en el ámbito académico como en el profesional.
- **Mejorar la Experiencia del alumno:** Proveer a los alumnos una plataforma donde puedan visualizar en qué actividades y grupos están asignados, promoviendo

la transparencia y la organización. La herramienta permitirá a los alumnos unirse a actividades mediante invitaciones enviadas por los profesores.

3.2 Casos de uso

En el contexto del desarrollo de una aplicación, es fundamental definir y comprender los casos de uso que guiarán la interacción entre los usuarios y el sistema. Los diagramas UML proporcionan una visión detallada de las funcionalidades clave de la plataforma, destacando las interacciones críticas entre los usuarios y el sistema. A continuación, se analizarán en profundidad los casos de uso representados en las figuras mencionadas, especificando los requisitos que fundamentarán el desarrollo de la aplicación.

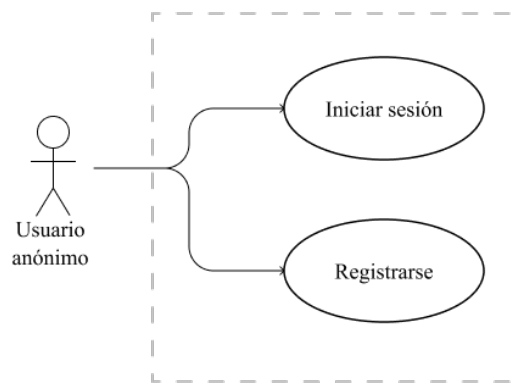


Figura 3.1: Casos de uso para un Usuario anónimo

La [figura 3.1](#) muestra los casos de uso relacionados con la autenticación. Tanto alumnos como profesores deben poder iniciar sesión, recuperar contraseñas olvidadas y cerrar sesión. Estos casos de uso son esenciales para garantizar la seguridad y accesibilidad de la plataforma.

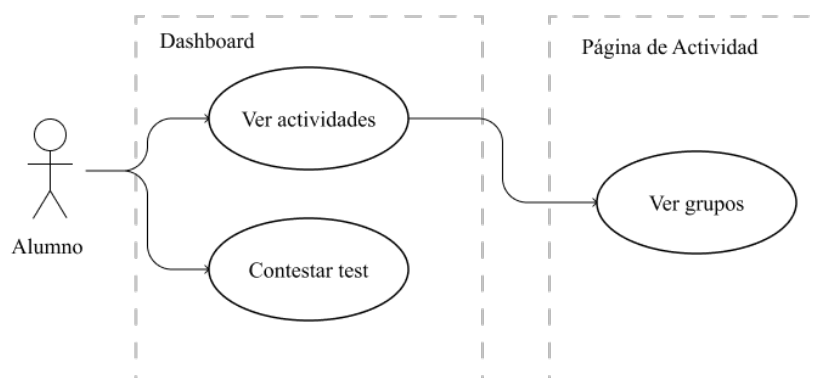


Figura 3.2: Casos de uso para un Alumno

En la [figura 3.2](#) se puede observar como los alumnos deben poder registrarse en el sistema, iniciar sesión, hacer cuestionarios y ver las actividades en las que están inscritos y ver en que grupos están asignados dentro de estas actividades. Estas funcionalidades aseguran que los alumnos puedan interactuar de manera eficiente con la plataforma y colaborar en la creación de los equipos.

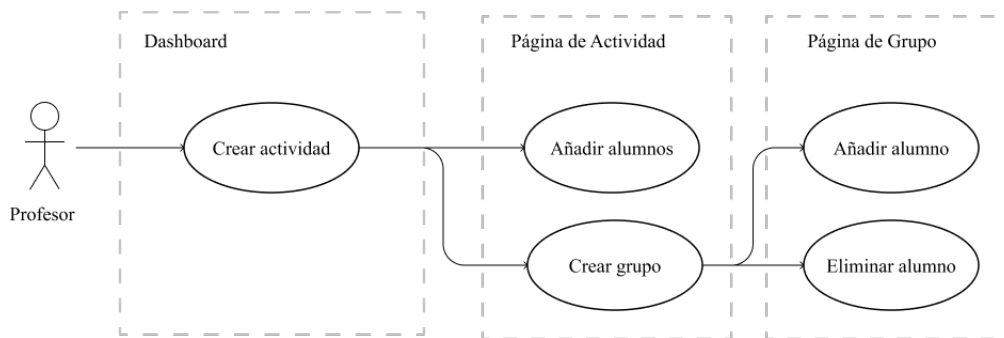


Figura 3.3: Casos de uso para un Profesor

Por último, como se puede ver en la [figura 3.3](#), los profesores pueden iniciar sesión, crear nuevas actividades y gestionar grupos dentro de estas actividades. Estas acciones permiten a los profesores organizar y supervisar eficazmente las actividades educativas realizadas.

4 Propuesta

Una vez definidos los propósitos y los objetivos, se propone una herramienta para facilitar el proceso de agrupación de los alumnos en grupos para un determinado proyecto, haciendo uso de un algoritmo que asocia a los alumnos basándose en sus roles principales según la metodología Belbin. La herramienta se trata de una aplicación web, donde se distinguen dos tipos de usuario, el profesor y el alumno. Cada profesor puede crear una actividad, donde podrá añadir a los alumnos de la asignatura. Los alumnos a su vez, pueden realizar el cuestionario que define sus roles, además de ver en qué actividades y grupos se encuentran.

4.1 Tecnologías y frameworks

Para desarrollar la aplicación web, se emplearán las siguientes tecnologías, las cuales permitirán desarrollar una aplicación web moderna, eficiente que cumpla con los objetivos propuestos. La implementación de la plataforma educativa se ha llevado a cabo utilizando una combinación de tecnologías modernas que abarcan tanto el frontend como el backend, así como la gestión de bases de datos y herramientas de integración y desarrollo. A continuación, se describen en detalle los componentes y tecnologías utilizadas en cada una de estas áreas, junto con algunas alternativas consideradas.

4.1.1 Frontend

El frontend de la aplicación se ha desarrollado con una combinación de tecnologías que permiten crear una interfaz de usuario atractiva y funcional:

- **HTML:** Utilizado para estructurar el contenido de las páginas web, HTML es la columna vertebral del frontend, proporcionando la base sobre la cual se construyen todos los elementos visuales y de interacción. Alternativas como Pug o Markdown

pueden ofrecer diferentes enfoques para la generación de HTML, pero HTML sigue siendo el estándar más ampliamente utilizado y compatible.

- **CSS:** Empleado para estilizar y mejorar la apariencia visual de la aplicación, CSS permite definir el diseño, colores, fuentes y disposición de los elementos en la página, asegurando una experiencia de usuario coherente y atractiva. Alternativas como Sass y Less extienden las capacidades de CSS con características adicionales, como variables y anidamiento, facilitando la escritura y mantenimiento de estilos complejos, pero no se han utilizado en este proyecto.
- **JavaScript:** Este lenguaje de programación se utiliza para implementar la lógica del lado del cliente y la interacción dinámica. JavaScript permite que la aplicación responda a las acciones del usuario sin necesidad de recargar la página, mejorando así la interactividad y la fluidez de la experiencia del usuario. Alternativas como TypeScript (un superconjunto de JavaScript con tipado estático) y CoffeeScript pueden mejorar la legibilidad y mantenimiento del código, pero JavaScript sigue siendo el lenguaje principal del ecosistema web.
- **Angular:** Este framework facilita la creación de aplicaciones web de una sola página (SPA) mediante componentes reutilizables y una gestión eficiente del estado. Angular permite desarrollar aplicaciones modulares y escalables, lo que facilita el mantenimiento y la expansión futura de la plataforma. Alternativas como React y Vue.js también ofrecen soluciones robustas para la creación de SPAs, cada una con su propia filosofía y ventajas. Sin embargo, Angular ha sido elegido por su integridad y soluciones completas en un solo paquete, además de las mejoras introducidas en su reciente versión 17.

4.1.2 Backend

El backend de la aplicación se ha desarrollado utilizando tecnologías que permiten gestionar la lógica del negocio, la autenticación y la gestión de datos de manera eficiente:

- **Node.js:** Este entorno de ejecución de JavaScript del lado del servidor permite manejar la lógica del negocio, gestionar la autenticación de usuarios y procesar las solicitudes de datos. Node.js es conocido por su eficiencia y capacidad para manejar múltiples conexiones simultáneamente. Alternativas como Python con Django o Ruby on Rails también ofrecen soluciones completas para el desarrollo del backend, pero Node.js ha sido seleccionado por su rendimiento y la ventaja de utilizar JavaScript en ambos lados del stack.
- **Express:** Este framework para Node.js simplifica la creación de rutas, middleware y controladores. Express facilita la estructuración del código del servidor y la gestión

de las solicitudes y respuestas HTTP, permitiendo una implementación más rápida y ordenada de las funcionalidades del backend. Alternativas como Koa y NestJS ofrecen diferentes enfoques y características adicionales, pero Express sigue siendo la opción más popular y ampliamente soportada en el ecosistema Node.js.

Base de datos

Para la gestión de datos, se ha optado por una base de datos NoSQL que ofrece flexibilidad y escalabilidad:

- **MongoDB:** Esta base de datos almacena los datos en formato JSON, lo que facilita su integración con aplicaciones desarrolladas en JavaScript. MongoDB se utiliza para gestionar la información de usuarios, actividades y grupos, permitiendo un acceso rápido y eficiente a los datos. Alternativas como PostgreSQL (una base de datos relacional) y Firebase (una base de datos en tiempo real) también ofrecen soluciones robustas para la gestión de datos, pero MongoDB ha sido elegido por su flexibilidad y facilidad de uso con estructuras de datos no estructuradas.

4.1.3 Integración y desarrollo

Para la integración y desarrollo del proyecto, se han utilizado herramientas que facilitan la gestión del código y el entorno de desarrollo:

- **Git:** Este sistema de control de versiones permite gestionar el código fuente del proyecto, facilitando la colaboración entre los desarrolladores y el seguimiento de los cambios realizados. Git asegura que todas las versiones del código estén bien documentadas y que cualquier cambio pueda ser revertido si es necesario. Alternativas como Mercurial y Subversion también ofrecen capacidades de control de versiones, pero Git ha sido elegido por su robustez y popularidad.
- **Visual Studio Code:** Este editor de código fuente ofrece soporte para depuración y control de versiones, proporcionando un entorno de desarrollo robusto y eficiente. Visual Studio Code incluye múltiples extensiones y herramientas que mejoran la productividad de los desarrolladores y la calidad del código producido. Alternativas como Atom y Sublime Text también son opciones populares, pero Visual Studio Code ha sido seleccionado por su conjunto de características integradas, su fuerte comunidad de soporte y plugins.

4.2 Funcionalidades de la Aplicación

Para cumplir con los objetivos detallados anteriormente, la aplicación web debe incluir las siguientes funcionalidades:

- **Registro, login y recuperación de contraseña para profesores.** Los profesores pueden registrarse en la plataforma mediante un formulario de inscripción, crear una cuenta de usuario y establecer una contraseña. Una vez registrados, pueden iniciar sesión utilizando sus credenciales. En caso de olvido de contraseña, los profesores pueden solicitar la recuperación de la misma a través de un enlace enviado a su correo electrónico registrado, lo que permite restablecer su acceso a la plataforma.
- **Creación y gestión de clases.** Los profesores pueden dar de alta nuevas clases introduciendo información relevante como el nombre de la titulación y la asignatura correspondiente. Además, tienen la opción de cargar un fichero con los correos electrónicos de los alumnos para inscribirlos automáticamente en la clase. Este proceso simplifica la administración de grupos grandes de estudiantes y facilita su incorporación a las clases.
- **Activación de cuentas de alumnos.** Al registrar una nueva clase, la aplicación envía automáticamente un correo electrónico a cada alumno con un enlace de activación. Este enlace permite a los alumnos crear su usuario y contraseña para acceder a la plataforma. Si un alumno ya está registrado en otra clase, se le notifica su adición a la nueva clase sin necesidad de crear una nueva cuenta, manteniendo la simplicidad y eficiencia en la gestión de usuarios.
- **Creación de proyectos dentro de las clases.** Los profesores pueden crear proyectos específicos dentro de una clase, definiendo un nombre y una duración para cada proyecto. Por ejemplo, un proyecto podría titularse “Proyecto de página web de turismo” y tener una duración predeterminada. Esta funcionalidad facilita la organización y seguimiento de las actividades académicas y proyectos de los estudiantes.
- **Definición y gestión de agrupaciones de alumnos.** Los profesores pueden establecer grupos de trabajo dentro de los proyectos. Tienen la capacidad de definir qué alumnos se agruparán, seleccionando manualmente a algunos estudiantes o utilizando métodos automáticos como la agrupación aleatoria (random) o basada en el método Belbin. Además, pueden establecer restricciones específicas como evitar que ciertos alumnos coincidan en el mismo grupo, o limitar el número de veces que ciertos alumnos pueden coincidir en diferentes grupos. También pueden establecer que ciertos alumnos deben estar juntos en un grupo, garantizando flexibilidad y personalización en la formación de equipos.

- **Creación y modificación de grupos a través de un servicio web.** La aplicación cuenta con una API que envía la información de agrupación a un web service y recibe los grupos formados. Este proceso no es inmediato, por lo que la aplicación notifica a los profesores cuando el proceso de agrupación esté completado. Los profesores también pueden realizar modificaciones manuales en los grupos creados si es necesario. Una vez formados los grupos, se envía un correo electrónico a cada alumno notificándole su grupo asignado, asegurando una comunicación eficaz.
- **Visualización de cuestionarios y perfiles Belbin.** Los profesores tienen acceso a un dashboard donde pueden visualizar los resultados de los cuestionarios de Belbin completados por los alumnos. Este dashboard proporciona una vista detallada de los roles de cada alumno según el perfil Belbin, facilitando la comprensión de las dinámicas de grupo y la asignación de roles en proyectos.
- **Gestión de cuentas y recuperación de contraseñas para alumnos.** Los alumnos, al recibir una invitación por primera vez, deben crear un usuario y una contraseña mediante un enlace enviado a su correo electrónico. También pueden recuperar su contraseña en caso de olvido, siguiendo un proceso similar al de los profesores. Esta funcionalidad garantiza que los alumnos puedan acceder a la plataforma de manera segura y recuperen su acceso en caso de problemas.
- **Visualización de clases actuales y pasadas.** Los alumnos pueden ver las clases en las que están inscritos, tanto las actuales como las pasadas. Esta funcionalidad permite a los alumnos tener un registro completo de su historial académico y acceder a la información de las clases en las que han participado.
- **Rellenar cuestionarios pendientes.** Los alumnos tienen acceso a los cuestionarios que necesitan completar, como el cuestionario de Belbin. Pueden ver y completar estos cuestionarios directamente en la plataforma, facilitando la recopilación de información necesaria para la organización de grupos y proyectos.
- **Visualización del perfil Belbin.** Los alumnos pueden acceder a un dashboard donde se detalla su perfil según los resultados del cuestionario de Belbin. Este perfil proporciona información valiosa sobre sus roles y habilidades, ayudando a los alumnos a entender mejor su contribución potencial en proyectos de grupo.
- **Visualización de grupos de trabajo asignados.** Los alumnos pueden ver el grupo de trabajo al que han sido asignados y los miembros del mismo. Esta funcionalidad asegura que los alumnos estén informados sobre sus equipos y facilita la coordinación y colaboración en los proyectos asignados.

5 Diseño

El diseño en una aplicación es una fase crucial que abarca desde la conceptualización visual hasta la arquitectura de datos y la integración de tecnologías. Esta sección detalla el proceso de creación de los mockups, el diseño de la base de datos en MongoDB utilizando JSON Schemas, y la relación entre las diversas tecnologías empleadas

El diseño de la herramienta web se centrará en proporcionar una interfaz de usuario intuitiva y accesible tanto para profesores como para alumnos, garantizando una experiencia de usuario eficiente y agradable. Se describen a continuación los principales componentes y funcionalidades del diseño.

5.0.1 Mockups

Antes de proceder con la implementación de la herramienta web, se han realizado mockups detallados para **visualizar y validar la interfaz** de las diferentes páginas y componentes de la aplicación. Los mockups son representaciones visuales que permiten planificar la estructura, el diseño y la funcionalidad de la interfaz de usuario, asegurando que todos los elementos se integren de manera coherente.

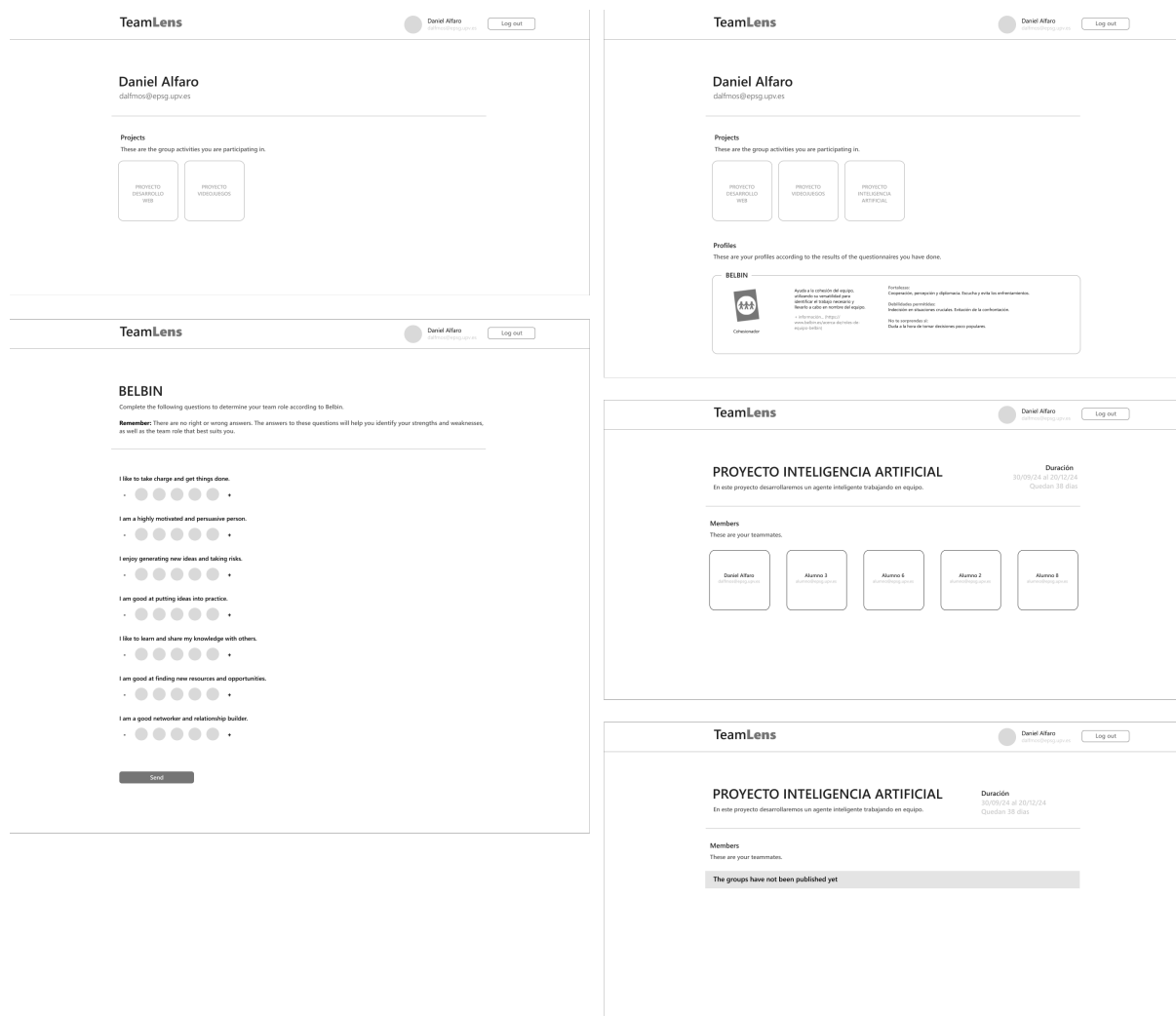


Figura 5.1: Mockups iniciales de la aplicación

5.0.2 Esquema de la Base de Datos

La base de datos está diseñada para almacenar y gestionar información relacionada con actividades, grupos, usuarios y cuestionarios. Cada colección dentro de la base de datos tiene un esquema específico que define la estructura y las restricciones de los documentos que puede contener. A continuación, se describen los esquemas de las principales colecciones de la base de datos: *activities*, *groups*, *users* y *questionnaires*.

En MongoDB, los esquemas de cada colección se definen utilizando JSON Schema, especificando los tipos de datos, los campos requeridos y las propiedades adicionales permitidas. Esta estructura permite asegurar la integridad y la consistencia de los datos almacenados. Se muestran en forma de tabla para una mejor visualización de la información.

Collección 'users'

Campo	Tipo	Requerido
_id	objectId	Sí
name	string	Sí
email	string	Sí
password	string	Sí
role	string	Sí
profile	string	No
invitationToken	string	No
resetToken	string	No
askedQuestionnaires	array	Sí
askedQuestionnaires.items	object	No
askedQuestionnaires.items.questionnaire	objectId	Sí
askedQuestionnaires.items.result	string	Sí
activities	array	No
activities.items	objectId	No
groups	array	No
groups.items	objectId	No
notifications	array	No
notifications.items	object	No
notifications.items.title	string	No
notifications.items.description	string	No
notifications.items.date	string	No
notifications.items.link	string	No

Collección 'activities'

Campo	Tipo	Requerido
_id	objectId	Sí
title	string	Sí
description	string	Sí
teacher	objectId	Sí
students	array	No
students.items	objectId	No
groups	array	No
groups.items	objectId	No
algorithmStatus	string	No

Collección 'groups'

Campo	Tipo	Requerido
_id	objectId	Sí
name	string	Sí
students	array	Sí
students.items	objectId	No
activity	objectId	No

Collección 'questionnaires'

Campo	Tipo	Requerido
_id	objectId	Sí
title	string	Sí
description	string	Sí
enabled	boolean	Sí
questionnaireType	string	Sí
questions	array	Sí
questions.items	object	No
questions.items.question	string	Sí
questions.items.type	string	Sí
questions.items.options	array	No

5.0.3 Relación entre las Tecnologías Utilizadas

La integración de diversas tecnologías es clave para el funcionamiento óptimo de la aplicación. Cada tecnología seleccionada cumple un propósito específico y se integra de manera cohesiva con las demás:

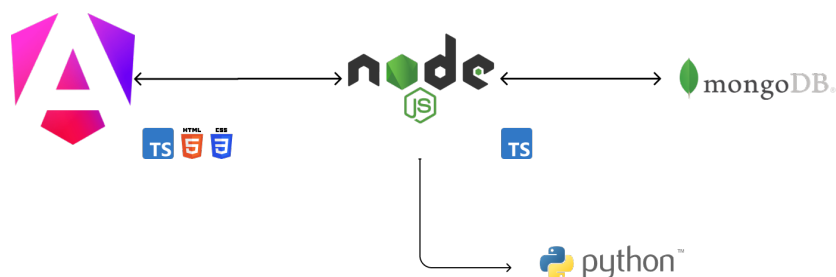


Figura 5.2: Relación entre las tecnologías utilizadas

Node.js para el Backend: Node.js, junto con Express, proporciona una plataforma robusta para construir la API REST que manejará las operaciones del servidor, como la autenticación de usuarios, la gestión de actividades y la comunicación con la base de datos.

Angular para el Frontend: Angular se utilizará para desarrollar una interfaz de usuario dinámica y receptiva. Este framework permite crear componentes reutilizables y servicios que interactúan con la API del backend, ofreciendo una experiencia de usuario fluida y eficiente.

MongoDB para la Base de Datos: MongoDB almacena los datos de la aplicación de manera flexible, permitiendo una fácil escalabilidad. Los JSON Schemas garantizan que los datos se mantengan estructurados y válidos, facilitando su manipulación y consulta.

Python para el Algoritmo de Emparejamiento: El algoritmo de emparejamiento basado en la metodología Belbin está desarrollado en Python. Este algoritmo será ejecutado desde el backend de Node.js mediante la integración de procesos hijo, como se detalla más adelante (6.2.1), asegurando una comunicación eficiente y una correcta aplicación de la lógica de emparejamiento.

La combinación de estas tecnologías permite construir una aplicación web moderna, eficiente y escalable, que facilita la agrupación de alumnos en proyectos de manera óptima.

6 Desarrollo de la propuesta

En esta sección se desarrolla el proceso de implementación de la herramienta web, explicando las partes que la componen, así como la integración con el algoritmo de agrupación, el cual ya está implementado (). La implementación de la aplicación se lleva a cabo en varias fases, asegurando que cada componente se desarrolla, prueba e integra correctamente. A continuación, se describe cada fase en detalle para proporcionar una visión clara del proceso de desarrollo.

6.1 Configuración del Entorno de Desarrollo

Primero, se prepara el entorno de desarrollo para asegurar que todas las herramientas y dependencias necesarias están disponibles. Esto incluye la instalación de Node.js y npm, los cuales son esenciales para la gestión de dependencias y la ejecución del backend. También se instala Angular CLI (Command Line Interface) para facilitar la creación y gestión del frontend de Angular. Para la base de datos, se instala y configura MongoDB, asegurando que esté correctamente configurado para aceptar conexiones desde la aplicación Node.js. Además, se configura un entorno virtual para Python, necesario para el desarrollo del algoritmo de emparejamiento.

6.2 Desarrollo del Backend

Con el entorno preparado, se comienza con la configuración inicial del backend. Se crea un proyecto Node.js e instala Express, un framework minimalista que facilita la creación de aplicaciones web y APIs. También se configura la conexión a la base de datos de MongoDB.

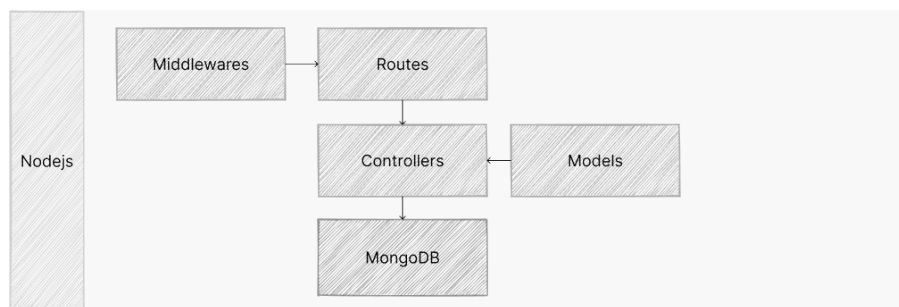


Figura 6.1: Diagrama de la aplicación en Angular

La API REST se desarrolla para manejar las diferentes funcionalidades de la aplicación. Se implementa un sistema de autenticación y autorización utilizando JWT (JSON Web Tokens), permitiendo el registro e inicio de sesión de usuarios, así como la protección de rutas sensibles.

Para la gestión de actividades, se crean endpoints que permiten a los profesores añadir, editar, listar y eliminar actividades. La gestión de grupos incluye endpoints para generar, modificar y visualizar grupo.

Los grupos serán creados manualmente o mediante el algoritmo de emparejamiento. Este algoritmo se integra en Node.js utilizando del módulo *child_process*, permitiendo la ejecución del script asíncronamente de Python desde un entorno en Javascript.

6.2.1 Implementación de colas e hilos para el procesamiento de las peticiones

El algoritmo se demora unas decenas de minutos, lo cual puede entorpecer el tráfico del servidor si no se maneja correctamente. La implementación de una cola para gestionar las peticiones entrantes y el uso de (*workers*) para procesar tareas en segundo plano permite manejar eficientemente las solicitudes sin sobrecargar el sistema. Este enfoque es crucial para aplicaciones que requieren procesamiento intensivo de datos, asegurando que el sistema se mantenga operativo.

activity.router.ts

```
const MAX_WORKERS = 10;
let activeWorkers = 0;
const taskQueue: any[] = [];
activitiesRouter.post(
  "/:id/create-algorithm", async (
    req: Request, res:
    Response) => {
```

```
    const id = req?.params
      ?.id;
    console.log(req.body);
    const data = {
      activityId: id,
      algorithmData: {
        ...req.body
      }
    };
    console.log(data);
    try {
      if (activeWorkers <
        MAX_WORKERS) {
        startAlgorithmWorker
          (data);
      } else {
        taskQueue.push(
          data);
      }
      res.status(200)
        .send({
          message: 'Tarea iniciada en segundo plano'
        });
    } catch (error: any) {
      res.status(500)
        .send({
          message: error
            .message
        });
    }
  });
const startAlgorithmWorker = (data:
any) => {
  activeWorkers++;
  const worker = new Worker(path
    .join(__dirname,
      './scripts/algorithm-req-worker.js'
    ), {
    workerData: data
  });
  worker.on('message', async (
    message) => {
    console.log(
      'Mensaje del trabajador: ${message}',
      data
        .activityId
    );
  });
  worker.on('exit', (code) => {
```

```

        activeWorkers--;
        if (taskQueue
            .length > 0) {
            const nextTask =
                taskQueue
                    .shift();
            startAlgorithmWorker
                (nextTask);
        }
        if (code !== 0) {
            console.error(
                'El trabajador se detuvo con el código de salida ${code}'
            );
        }
    });
};

```

El código presentado en *activity.router.ts* (6.2.1) implementa una cola para gestionar las peticiones al endpoint `/create-algorithm`, garantizando una administración eficiente de las solicitudes y evitando la sobrecarga del sistema. Inicialmente, se definen variables clave: `MAX_WORKERS`, que establece el número máximo de trabajadores simultáneos (configurado en 10), `activeWorkers`, que lleva un conteo de los trabajadores activos, y `taskQueue`, una cola para almacenar las tareas cuando se alcanza el límite de trabajadores activos.

El endpoint POST `/:id/create-algorithm` recibe solicitudes para ejecutar un algoritmo específico para una actividad. Extrae el id de la actividad de los parámetros de la solicitud y combina los datos del cuerpo de la solicitud en un objeto `data`. Si el número de trabajadores activos es menor que `MAX_WORKERS`, se inicia un nuevo worker mediante la función `startAlgorithmWorker(data)`. Si no, la tarea se agrega a `taskQueue` para su procesamiento posterior. La función responde con un mensaje de éxito si la tarea se maneja correctamente o con un mensaje de error si ocurre algún problema.

La función `startAlgorithmWorker` incrementa el conteo de `activeWorkers` y crea un nuevo trabajador utilizando un worker-thread de node llamado *algorithm-req-worker* (6.2.1), pasando los datos de la tarea. Los eventos `message` y `exit` del trabajador se manejan para registrar el progreso y decrementar `activeWorkers` al finalizar, respectivamente. Si hay tareas en `taskQueue` al finalizar un trabajador, se inicia la siguiente tarea.

`/algorithm-req-worker.js`

```

const {
    parentPort,
    workerData
} = require('worker_threads');

```

```
const {
  exec
} = require('child_process');
const path = require('path');
const jsonString = JSON.stringify(
  workerData.algorithmData
).replace(
  /\//g, '\\');
const scriptPath = path.join(__dirname,
  'algorithm.py');
const command =
  `py "${scriptPath}" ${jsonString}`;
// console.log('Comando a ejecutar: ${command}');
exec(command, (error, stdout,
stderr) => {
  if (error) {
    console.error(
      'Error ejecutando el script: ${error.message}'
    );
    return;
  }
  if (stderr) {
    console.error(
      'Error en el script: ${stderr}'
    );
    return;
  }
  parentPort.postMessage(
    stdout);
});
```

6.3 Desarrollo del Frontend

El frontend se desarrollará utilizando Angular 17, creando un proyecto inicial y configurando el enrutamiento de la aplicación. Se configurarán servicios en Angular para consumir la API REST del backend, manejando la autenticación en el frontend utilizando tokens y guards para proteger las rutas de usuarios no identificados, además de asegurar una comunicación fluida con el backend mediante *Services* (6.3) e *Interceptors*(6.3).

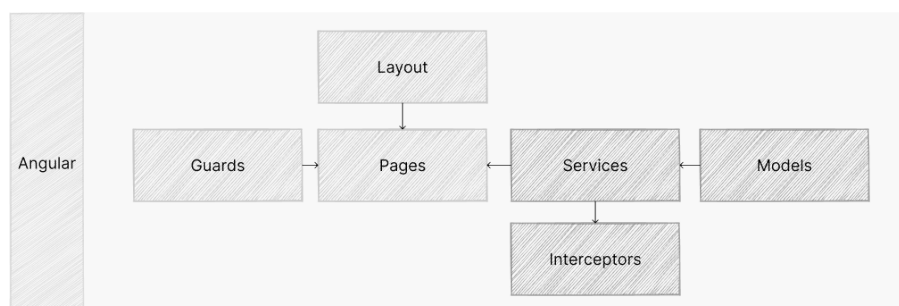


Figura 6.2: Diagrama de la aplicación

Services

Entre los elementos fundamentales de la arquitectura de Angular se encuentran los servicios, que desempeñan un papel crucial en la organización y funcionamiento de las aplicaciones. Un servicio es una clase que se encarga de proporcionar funcionalidades compartidas entre diferentes componentes de la aplicación. Estos servicios se utilizan para encapsular lógica que no pertenece directamente a la presentación de datos, como la comunicación con APIs externas, el manejo de datos, la autenticación de usuarios, y cualquier otra lógica que se desee reutilizar a lo largo de la aplicación. La principal ventaja de los servicios es la separación de responsabilidades. Al mover la lógica de negocio fuera de los componentes y centralizarla en servicios, se logra un código más limpio, modular y fácil de mantener. Esta separación también facilita la realización de pruebas unitarias, ya que la lógica de negocio se encuentra desacoplada de la lógica de la interfaz de usuario.

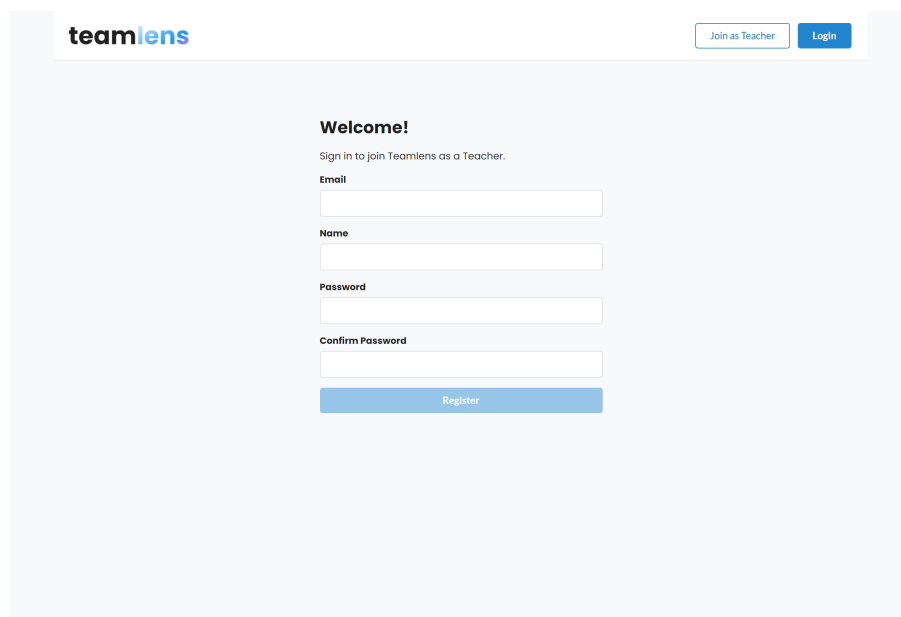
Interceptors

Los interceptores en Angular son una característica avanzada del módulo `HttpClient` que permiten interceptar y manipular las solicitudes y respuestas HTTP. Actúan como middleware que puede modificar, añadir, o manejar errores de las solicitudes antes de que lleguen al servidor y de las respuestas antes de que lleguen a la aplicación

6.3.1 Páginas

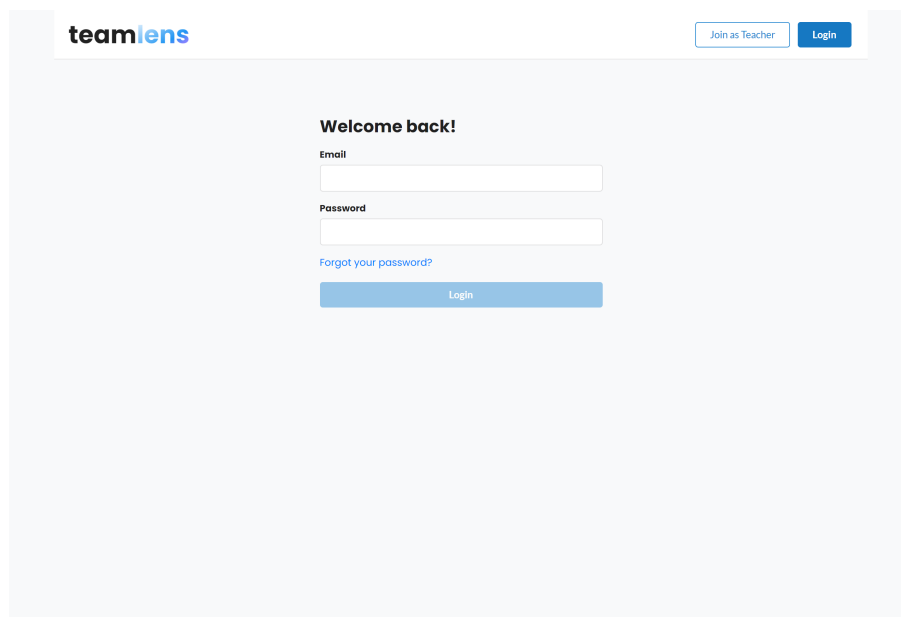
Una vez diseñados se desarrollarán varios componentes, comenzando con los formularios de inicio de sesión y registro. Estos formularios manejarán validaciones y enviarán datos al backend para autenticar a los usuarios.

Páginas de Autenticación



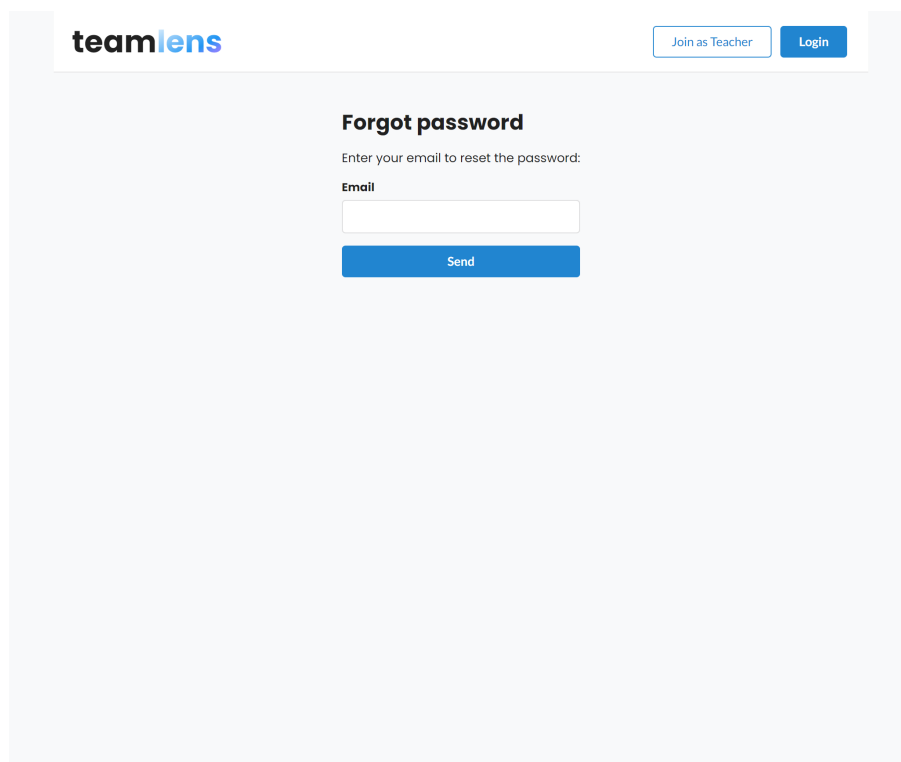
The screenshot shows a web page for Teamlens. At the top left is the 'teamlens' logo. At the top right are two buttons: 'Join as Teacher' and 'Login'. The main content area is titled 'Welcome!' and contains the text 'Sign in to join Teamlens as a Teacher.' Below this are four input fields labeled 'Email', 'Name', 'Password', and 'Confirm Password'. At the bottom of the form is a blue 'Register' button.

Figura 6.3: Registro del profesor



The screenshot shows the login page for teamlens. At the top left is the teamlens logo. At the top right are two buttons: "Join as Teacher" and "Login". The main content area is titled "Welcome back!". Below the title are two input fields: "Email" and "Password". Below the "Password" field is a link "Forgot your password?". At the bottom of the form is a blue "Login" button.

Figura 6.4: Iniciar sesión



The screenshot shows the forgot password page for teamlens. At the top left is the teamlens logo. At the top right are two buttons: "Join as Teacher" and "Login". The main content area is titled "Forgot password". Below the title is the instruction "Enter your email to reset the password:". Below this is an "Email" input field. At the bottom of the form is a blue "Send" button.

Figura 6.5: Olvido de contraseña

Dashboard del Profesor

El dashboard del profesor será una vista principal donde podrá ver y crear todas sus actividades.

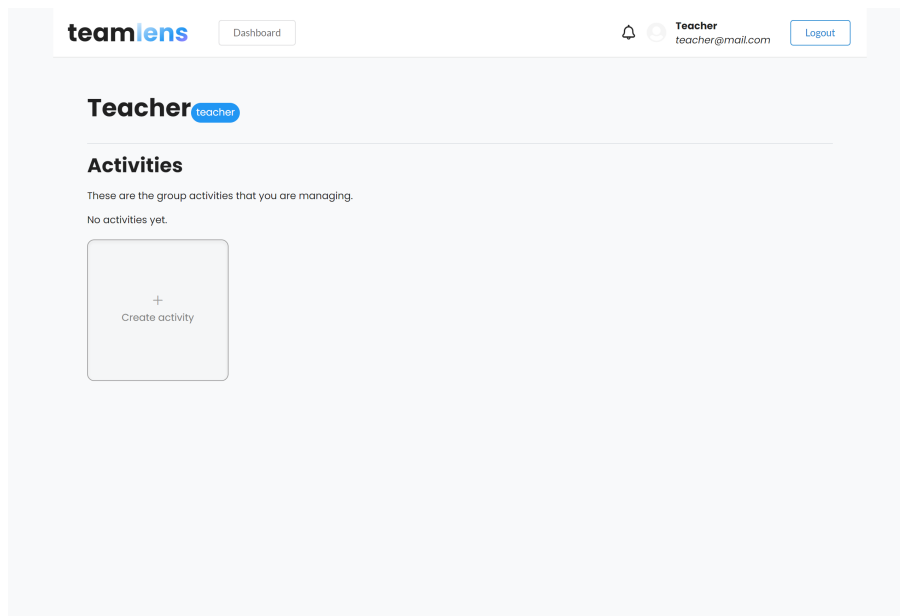


Figura 6.6: Dashboard del profesor

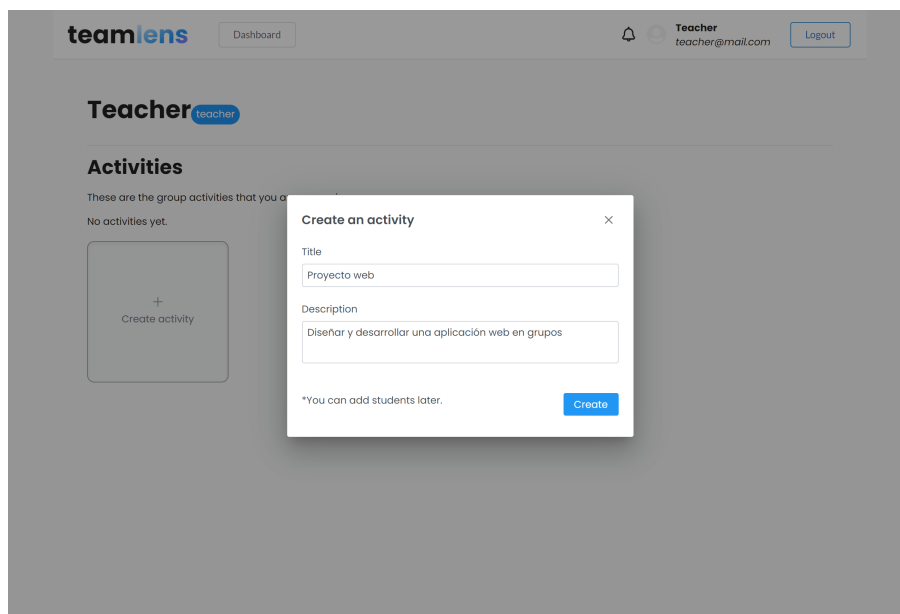


Figura 6.7: Crear una actividad

Detalle Actividad desde la vista del Profesor

En esta página, el profesor podrá gestionar la actividad, añadiendo alumnos y creando grupos.

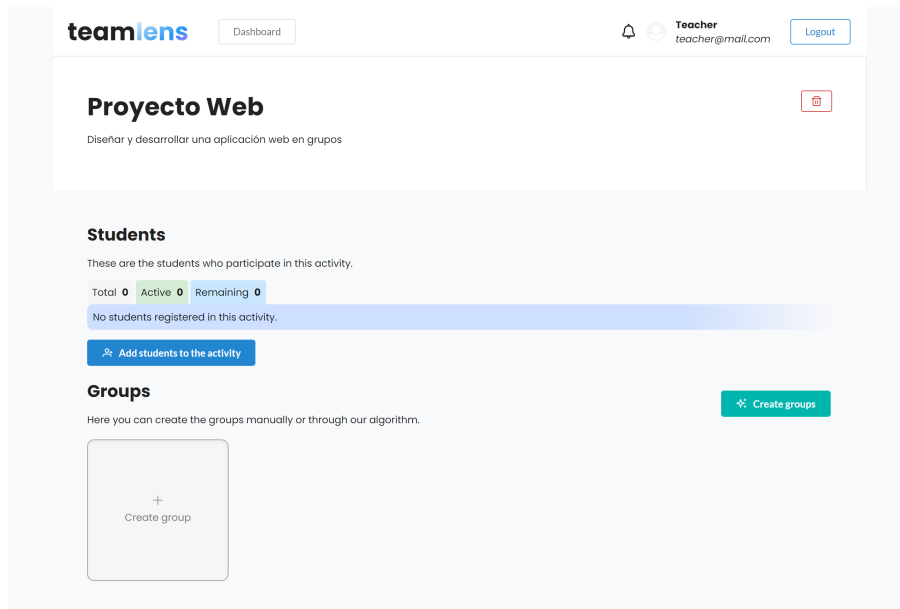


Figura 6.8: Detalle de una actividad

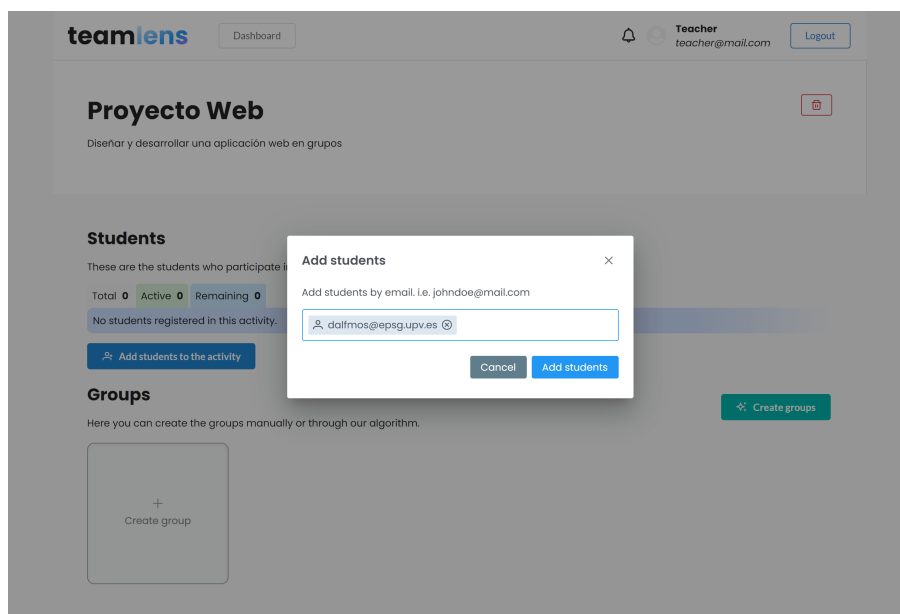


Figura 6.9: Formulario para añadir estudiantes a la actividad

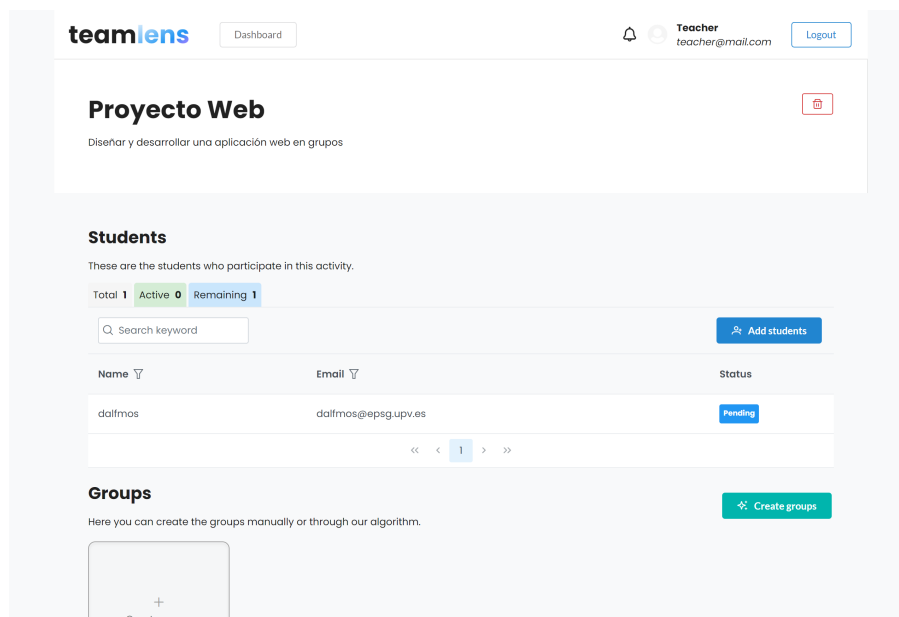


Figura 6.10: Estudiante pendiente

Los alumnos añadidos que no estén registrados, recibirán una **invitación** por correo electrónico, permaneciendo en estado '*Pendiente*' hasta que se registren en la plataforma, como se puede observar en 6.10.

Gestionar grupos

Una vez hayan suficientes alumnos en la actividad, el profesor podrá crear grupos manualmente, haciendo click en Crear Grupo o automáticamente, haciendo uso del algoritmo de emparejamiento, haciendo click en el boton destacado Crear grupos.

The screenshot displays the TeamLens interface for a project titled "Proyecto Web". At the top, there is a navigation bar with the "teamlens" logo, a "Dashboard" button, a notification bell, a user profile for "Teacher" (teacher@mail.com), and a "Logout" button. The main content area is divided into several sections:

- Project Header:** "Proyecto Web" with a subtitle "Diseñar y desarrollar una aplicación web en grupos" and a trash icon.
- Students Section:** A heading "Students" followed by the text "These are the students who participate in this activity." Below this, a summary shows "Total 4", "Active 3", and "Remaining 1". There is a search bar labeled "Search keyword" and an "Add students" button.
- Student List Table:** A table with columns for "Name", "Email", and "Status".

Name	Email	Status
Dani	dani@mail.com	Active
student	student@mail.com	Active
Daniel	dalfmos@epsg.upves	Active
infdaifamosni	infdaifamosni@mail.com	Pending
- Groups Section:** A heading "Groups" followed by the text "Here you can create the groups manually or through our algorithm." and a "Create groups" button. Below this, there are three boxes: "Grupo 1" (2 members), "Grupo 2" (2 members), and a "Create group" button.
- Questionnaires Section:** A heading "Questionnaires" followed by a card for "BELBIN Test". The card includes the text "Trata de responder teniendo en cuenta tu comportamiento real ..." and "7 questions.", along with a "Send to students" button.

Figura 6.11: Grupos creados en la actividad

Registro del Alumno

Como se comentaba en el apartado anterior, el Alumno solo podrá registrarse mediante invitación, que recibirá por correo electrónico cuando un profesor le invite a una actividad por primera vez. El contenido de este es un enlace que le permitirá rellenar sus datos.

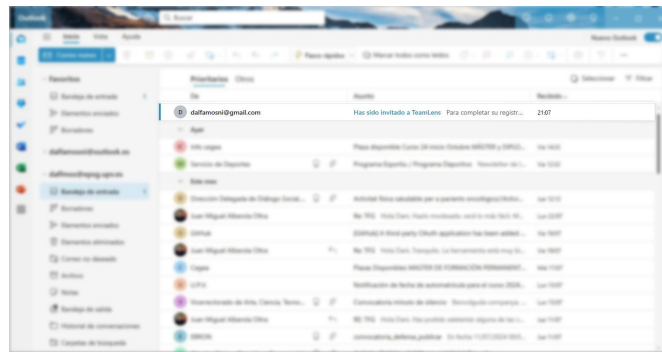


Figura 6.12: Invitación por correo electrónico

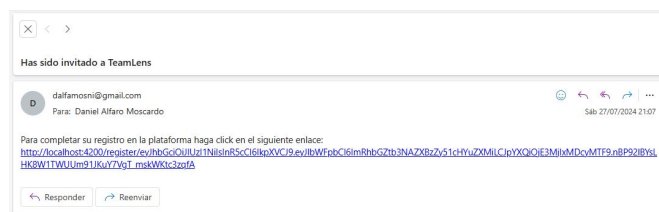


Figura 6.13: Enlace que recibe un estudiante.

 A screenshot of the TeamLens registration page. The page has a 'Welcome!' heading and a sub-heading 'Sign in to join TeamLens as a Student'. There are input fields for 'Email' (with 'dafamosi@gmail.com' entered), 'Name' (with 'Daniel' entered), 'Password', and 'Confirm Password'. A blue 'Register' button is at the bottom.

Figura 6.14: Registro del alumno

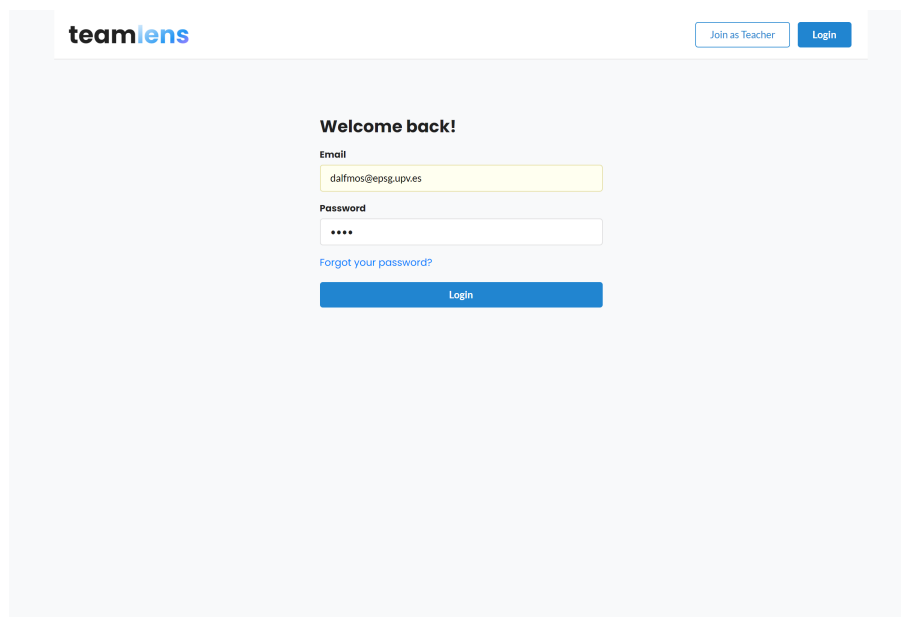


Figura 6.15: Login del alumno

Dashboard del Alumno

El dashboard del alumno permitirá a los estudiantes ver sus actividades y grupos asignados. Se desarrollarán componentes para listar estas actividades y mostrar detalles de los grupos, proporcionando una visión clara de su participación en los proyectos.

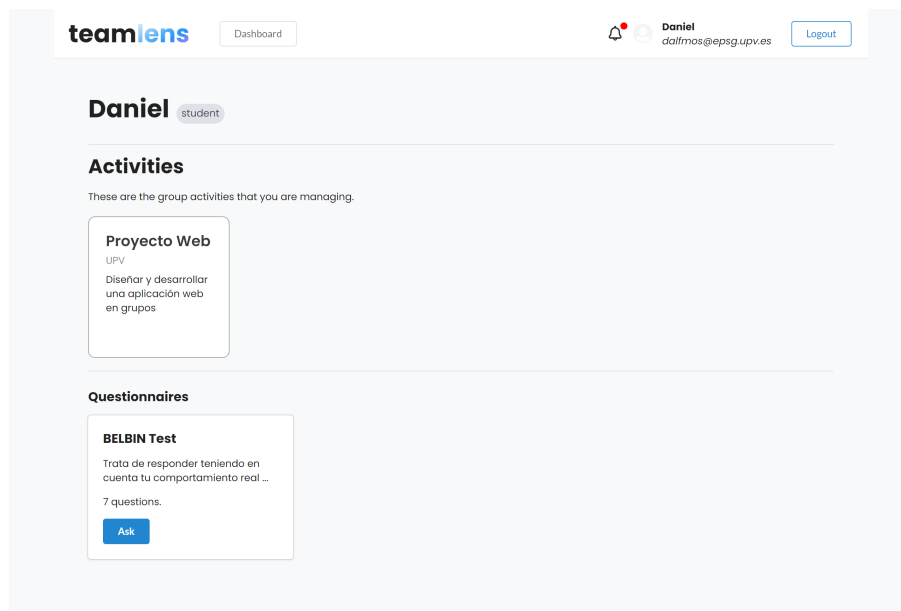


Figura 6.16: Dashboard del estudiante

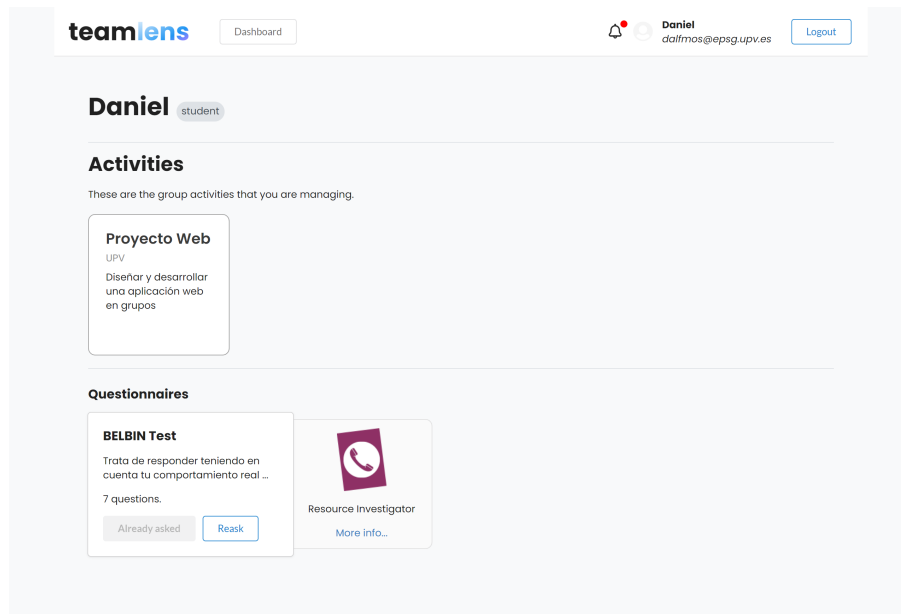


Figura 6.17: Enter Caption

El estudiante podrá relizar el cuestionario que asignará su rol Belbin, haciendo click en el boton de responder en esta misma página, lo que le redirigirá al cuestionario. (6.16)

Una vez realizado el cuestionario, podrá visualizar su rol dominante en la pagina anterior (6.17)

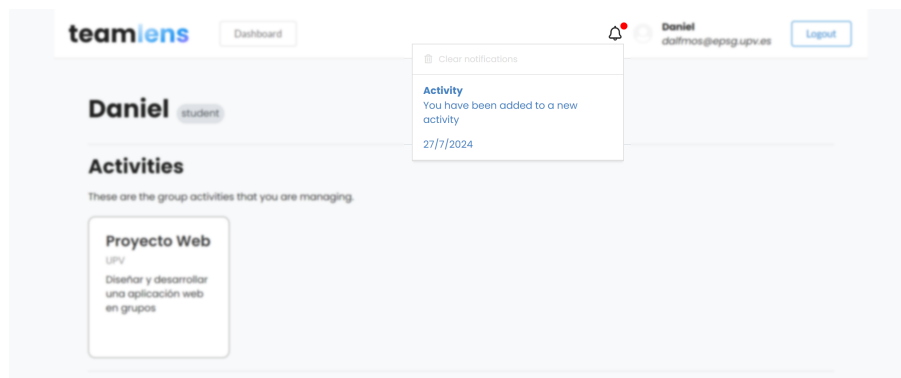
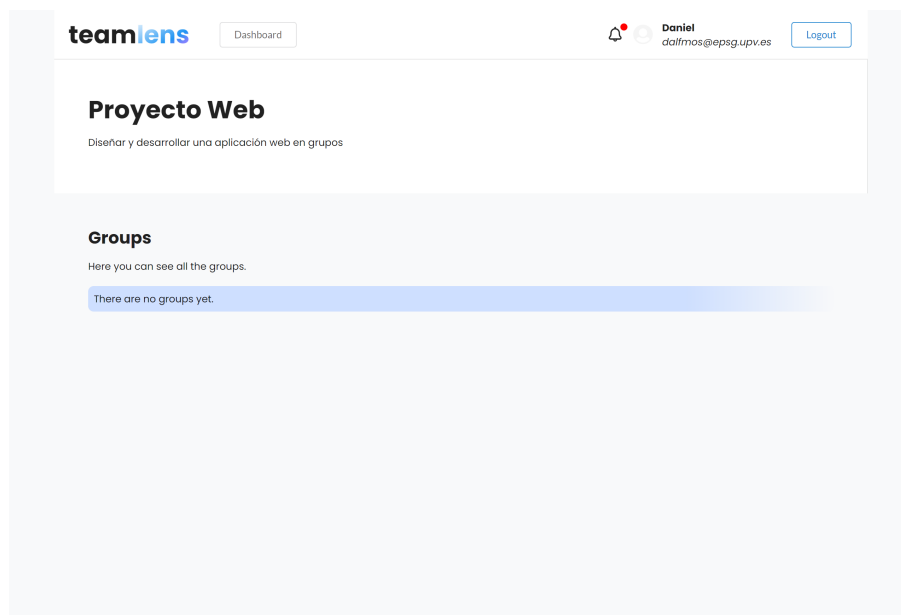
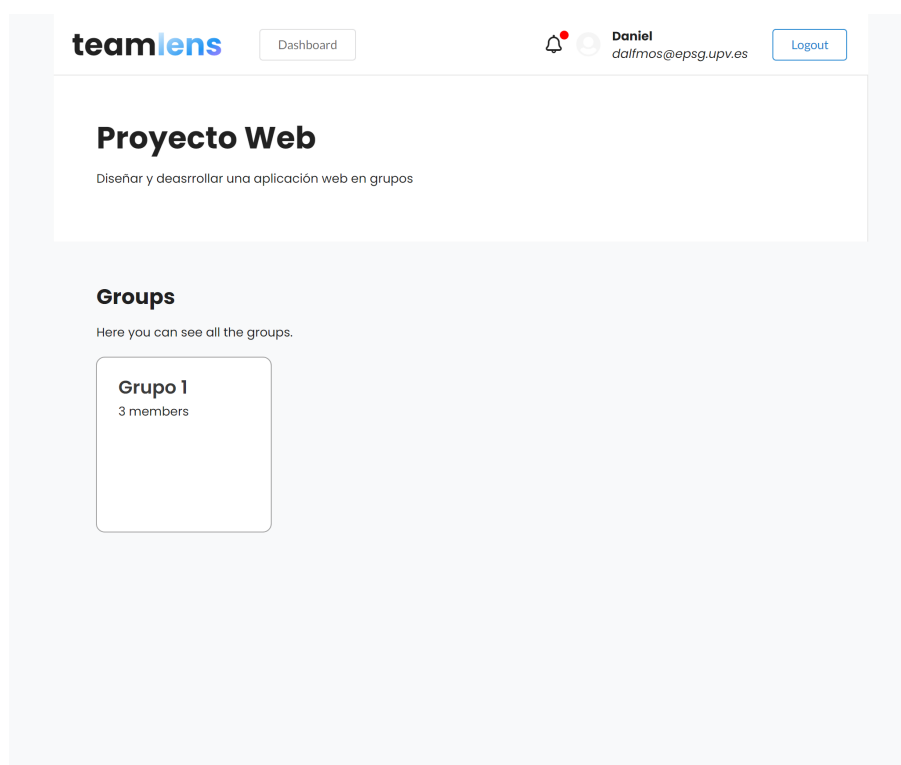


Figura 6.18: Notificación del estudiante

Detalle de Actividad desde la vista del Alumno**Figura 6.19:** Detalle de actividad del estudiante**Figura 6.20:** Grupos creados en la actividad

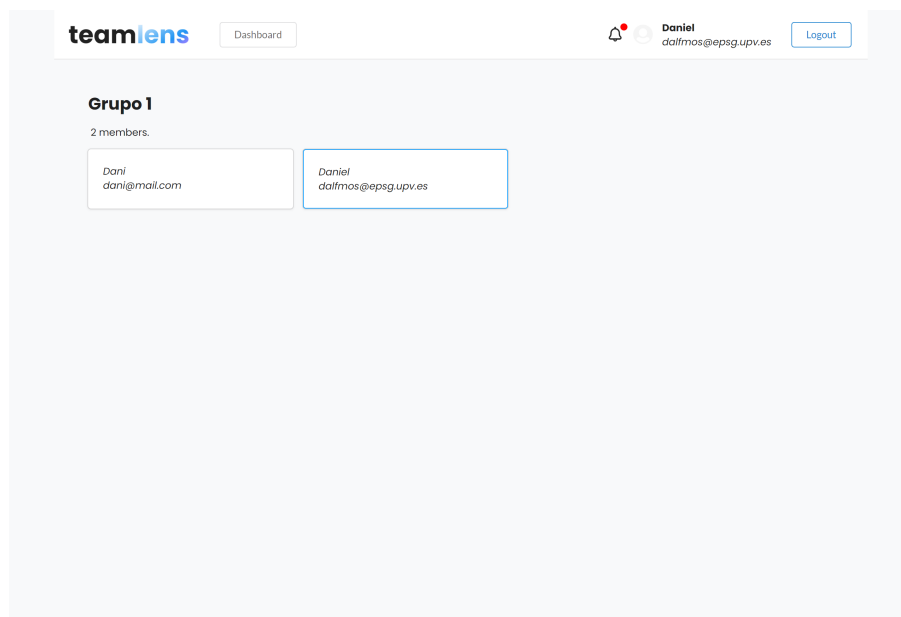


Figura 6.21: Detalle de grupo del estudiante

Una vez creados los grupos, el alumno podrá ver en qué grupo ha sido asignado, además de ver todos los grupos de la actividad y que alumnos los componen (6.21).

7 Conclusiones

En este apartado se resumen los logros y aprendizajes obtenidos a través del desarrollo de la herramienta web para la gestión de actividades y formación de equipos en el entorno educativo. Se destaca la eficiencia lograda para los profesores, la optimización en la formación de equipos, el fomento del aprendizaje colaborativo y la mejora en la experiencia del alumno. Además, se valoran los desafíos enfrentados y las tecnologías utilizadas, subrayando las bases establecidas para futuras mejoras y expansiones de la aplicación.

La herramienta web desarrollada logra cumplir con el objetivo principal de facilitar la tarea de los profesores en la gestión de actividades y la formación de equipos de trabajo. Al integrar un sistema de gestión eficiente y automatizado, los profesores pueden concentrarse más en la enseñanza y menos en las tareas administrativas. La implementación de funcionalidades como la creación de actividades, la adición de alumnos y la capacidad de realizar ajustes manuales a los grupos generados automáticamente proporciona un nivel de flexibilidad y control que es crucial en el entorno educativo.

Optimización en la Formación de Equipos

La utilización del algoritmo de emparejamiento basado en la metodología Belbin garantiza la formación de equipos equilibrados, lo que es esencial para la cohesión y el éxito de los proyectos. Al considerar los roles principales de los alumnos, el sistema no solo mejora la dinámica del grupo, sino que también potencia el aprendizaje colaborativo. Esto, a su vez, prepara a los estudiantes para situaciones del mundo real donde la diversidad de habilidades y personalidades es fundamental para el éxito en el trabajo en equipo. La aplicación no solo facilita la formación de equipos, sino que también promueve un entorno de aprendizaje colaborativo. Los alumnos, al formar parte de equipos equilibrados, tienen la oportunidad de experimentar y desarrollar habilidades esenciales para el trabajo en equipo. Esta experiencia es valiosa tanto en el ámbito académico como en el profesional, preparando a los estudiantes para colaborar eficazmente en sus futuras carreras. La

plataforma mejora significativamente la experiencia del alumno al permitirle visualizar sus actividades y grupos, promoviendo una mayor organización y transparencia.

Uso de Tecnologías Modernas

La elección de tecnologías y frameworks modernos como Node.js, Angular, y MongoDB asegura que la aplicación sea robusta, escalable y fácil de mantener. La combinación de estas tecnologías permite un desarrollo ágil y eficiente, además de garantizar que la plataforma pueda evolucionar y adaptarse a nuevas necesidades en el futuro. La integración del algoritmo de emparejamiento en Python mediante *child_process* demuestra una solución efectiva para ejecutar lógica compleja dentro de un entorno desde el lado del servidor en un entorno de Javascript sin comprometer su velocidad.

7.0.1 *Desafíos y Aprendizajes*

El desarrollo del proyecto también ha enfrentado desafíos, como la integración de diferentes tecnologías y la implementación del algoritmo de emparejamiento. Sin embargo, estos desafíos han proporcionado valiosos aprendizajes que han contribuido a la mejora continua del proyecto.

En conclusión, la herramienta web desarrollada cumple con éxito los objetivos propuestos, proporcionando una solución integral para la gestión de actividades y la formación de equipos en el entorno educativo. La combinación de tecnología, metodología y diseño centrado en el usuario resulta en una plataforma que no solo facilita el trabajo de los profesores, sino que también enriquece la experiencia educativa de los alumnos.

7.0.2 *Relación del Trabajo desarrollado con los Estudios cursados*

El desarrollo de la herramienta web presentada en este proyecto se vincula directamente con los conocimientos adquiridos en varias de las asignaturas estudiadas. Este apartado destaca cómo las competencias y habilidades obtenidas en dichas asignaturas han sido fundamentales para la realización exitosa del proyecto.

En la asignatura de Diseño de Interfaces Web y el Proyecto web realizado, se estudiaron los principios y técnicas para crear interfaces de usuario intuitivas y efectivas. Estos conocimientos fueron aplicados en la creación del frontend de la aplicación, utilizando tecnologías como HTML, CSS y Angular. La experiencia adquirida en esta asignatura permitió diseñar una interfaz que no solo es visualmente atractiva, sino también funcional y fácil de usar, mejorando así la experiencia del usuario tanto para profesores como para alumnos.

La diversas asignaturas de programación proporcionaron una base sólida en varios lenguajes de programación y paradigmas, incluyendo JavaScript y Node.js, que fueron esenciales para el desarrollo del backend de la aplicación. La implementación de la lógica de negocio, la gestión de la base de datos con MongoDB y la creación de APIs RESTful son ejemplos de cómo los conceptos aprendidos en programación se aplicaron en el desarrollo de la plataforma.

Además, la integración de un algoritmo de emparejamiento basado en Python demuestra la capacidad de combinar y utilizar múltiples lenguajes de programación y tecnologías en un solo proyecto, reflejando la enseñanza interdisciplinaria del grado. La gestión eficiente de recursos del servidor mediante el uso de colas y trabajadores para manejar peticiones concurrentes también se fundamenta en los principios de programación concurrente y asíncrona.

En resumen, el proyecto desarrollado no solo consolida los conocimientos teóricos adquiridos durante la carrera, sino que también los pone en práctica en un contexto real y aplicable, demostrando la relevancia y aplicabilidad de las asignaturas cursadas para la creación de soluciones tecnológicas innovadoras.

7.0.3 Relación de la Aplicación con los Objetivos de Desarrollo Sostenible (ODS)

La aplicación desarrollada para la gestión de proyectos académicos y la organización de grupos de trabajo no solo optimiza la eficiencia en el ámbito educativo, sino que también contribuye a varios Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. A continuación, se detalla cómo cada funcionalidad de la aplicación puede alinearse y apoyar específicamente algunos de estos objetivos:

ODS 4: Educación de Calidad. La aplicación desarrollada para la gestión de proyectos académicos y la organización de grupos de trabajo no solo optimiza la eficiencia en el ámbito educativo, sino que también contribuye a varios Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. A continuación, se detalla cómo de la aplicación se alinea y apoya específicamente algunos de estos objetivos:

ODS 5: Igualdad de Género. La aplicación juega un papel crucial en la promoción de la igualdad de género en el entorno educativo. La funcionalidad de definición y gestión de agrupaciones de alumnos permite que los profesores formen grupos de trabajo basados en criterios objetivos y pedagógicos, sin distinciones de género. En particular, el uso del método Belbin para la formación de equipos se basa en roles y características individuales de los estudiantes, como su capacidad de trabajo en equipo, liderazgo, y habilidades técnicas, sin considerar el género. Esto asegu-

ra que todos los estudiantes, independientemente de su género, tengan las mismas oportunidades de participar y destacarse en los proyectos académicos. Además, la posibilidad de realizar modificaciones manuales en los grupos creados permite a los profesores asegurar un entorno de trabajo justo y equilibrado, en el que se valoran las habilidades y competencias de cada estudiante sin prejuicios ni discriminaciones de género.

ODS 8: Trabajo Decente y Crecimiento Económico. La aplicación facilita la preparación de los estudiantes para el mundo laboral mediante la creación de proyectos y la gestión de grupos de trabajo. Estas experiencias académicas permiten a los alumnos desarrollar habilidades cruciales como la colaboración, la gestión de proyectos y la resolución de problemas, que son esenciales para su futura empleabilidad y contribución al crecimiento económico. Al trabajar en proyectos reales y estructurados, los estudiantes adquieren competencias prácticas que les preparan para desempeñarse en entornos laborales diversos y dinámicos.

ODS 10: Reducción de las Desigualdades. La funcionalidad de activación de cuentas de alumnos y la gestión de cuentas aseguran que todos los estudiantes tengan las mismas oportunidades de acceso a la plataforma y sus beneficios educativos. La aplicación también permite a los profesores monitorizar el progreso de los estudiantes y proporcionar soporte adicional a aquellos que lo necesiten, ayudando a reducir las desigualdades en el rendimiento académico. Al ofrecer una plataforma inclusiva y accesible, se promueve un entorno educativo donde todos los estudiantes pueden alcanzar su máximo potencial, independientemente de sus antecedentes socioeconómicos.

ODS 16: Paz, Justicia e Instituciones Sólidas La transparencia y la eficiencia en la gestión de proyectos y grupos de trabajo fomentadas por la aplicación contribuyen a la creación de instituciones educativas más sólidas y justas. La posibilidad de realizar modificaciones manuales de los grupos creados garantiza que los profesores puedan intervenir cuando sea necesario para asegurar la equidad y justicia en la distribución de tareas y responsabilidades. Al promover un entorno educativo basado en la equidad y la justicia, la aplicación contribuye a la formación de ciudadanos responsables y comprometidos con los valores democráticos.

ODS 17: Alianzas para Lograr los Objetivos La creación de grupos demuestra la importancia de las alianzas y la colaboración en el desarrollo de soluciones. Esta aplicación puede servir como un ejemplo de cómo las tecnologías pueden ser utilizadas para promover objetivos globales a través de la colaboración entre instituciones educativas y tecnológicas.

7.1 Trabajos futuros

La herramienta desarrollada sienta las bases para futuras mejoras y expansiones. La recolección de feedback de los usuarios será crucial para identificar áreas de mejora y desarrollar nuevas funcionalidades.

Se considera la posibilidad de incorporar nuevos cuestionarios, proporcionando al profesor una mayor flexibilidad en la generación automática de grupos. Esta funcionalidad adicional permitiría adaptar los cuestionarios a diferentes necesidades pedagógicas y dinámicas de grupo según los objetivos de cada actividad.

Otra de las ampliaciones que se podría realizar es otorgar más complejidad a los grupos, por ejemplo, dotándolos de funcionalidades más sofisticadas, donde los alumnos puedan interactuar con sus compañeros dejando comentarios positivos/negativos sobre estos o compartir enlaces a repositorios.

Estas son solo algunas sugerencias que podrían explorarse en futuros desarrollos para mejorar y ampliar las funcionalidades del sistema. La evolución constante de las tecnologías y las cambiantes necesidades de los usuarios abren numerosas oportunidades para enriquecer y actualizar esta aplicación.

7.1.1 Enlaces de interés

 **Repositorio de Github del frontend.**

https://github.com/danialfaro/frontend_teamlens

 **Repositorio de Github del backend.**

https://github.com/danialfaro/backend_teamlens

Bibliografía

- Alberola, J. M., Del Val, E., Sanchez-Anguix, V., Palomares, A., & Teruel, M. D. (2016). An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowledge-Based Systems*, 101, 1-14 (vid. pág. 3).
- Batenburg, R., van Walbeek, W., & in der Maur, W. (2013). Belbin role diversity and team performance: is there a relationship? *Journal of Management Development*, 32(8), 901-913 (vid. pág. 5).
- Bonebright, D. A. (2010). 40 years of storming: a historical review of Tuckman's model of small group development. *Human Resource Development International*, 13(1), 111-120 (vid. pág. 4).
- Conde, M. Á., Rodríguez-Sedano, F. J., Sánchez-González, L., Fernández-Llamas, C., Rodríguez-Lera, F. J., & Matellán-Olivera, V. (2016). Evaluation of teamwork competence acquisition by using CTMTC methodology and learning analytics techniques. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturalism*, 787-794 (vid. pág. 4).
- Del Val, E., Alberola, J. M., Sanchez-Anguix, V., Palomares, A., & Teruel, M. D. (2014). A team formation tool for educational environments. *Trends in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, 173-181 (vid. pág. 4).
- Google. (s.f.). Angular. <https://angular.dev/>

Gündüz, H. B. (2008). An Evaluation on Belbin's Team Roles Theory (The Case of Sakarya Anatolian Professions High School, Professions High School and Vocational High School for Industry). *World Applied Sciences Journal*, 4(3), 460-469 (vid. pág. 3).

MongoDB. (s.f.). <https://www.mongodb.com/docs/drivers/node/current/>

Node.js®. (s.f.). NodeJS. <https://nodejs.org/docs/latest/api/>