

UNIVERSIDAD POLITÉCNICA DE MADRID

E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

PROYECTO FIN DE GRADO

GRADO EN INGENIERÍA DEL SOFTWARE

# EduWordle: Una Herramienta Web para la Gamificación de Contenidos Educativos



Universidad  
Politécnica  
de Madrid

ETSI SISTEMAS  
INFORMÁTICOS

**Desarrollado por:** Elena Campanero Belda y Miriam  
Sánchez Saiz

**Dirigido por:** Daniel López Fernández

Madrid, 27 de junio de 2025

# Agradecimientos

---

“Quiero dar las gracias a mi familia, amigos y pareja por el apoyo que me han dado durante todo este tiempo, sin ellos nada de esto habría sido igual. En especial a mi madre que durante estos últimos meses ha estado a mi lado para animarme cuando todo parecía torcerse.

Agradecer también a mi tutor Daniel por su cercanía, confianza, sus ganas y su entusiasmo por enseñar ha sido de gran motivación para llevar a cabo este proyecto.

Y por supuesto, a mi compañera Elena, por ser una gran compañera de TFG, ha sido fundamental para la realización de este trabajo. Sin ella, no habría sido posible.”

Miriam Sánchez Saiz

Quiero agradecer a todas las personas que han formado parte de mi vida en un momento u otro de este camino. A mi tutor Daniel, por el apoyo y cercanía, siempre valorando nuestro esfuerzo y elogiando nuestros avances. A mi compañera Miriam, por hacer más ameno el trayecto y por afrontar este reto conmigo; estoy muy orgullosa de todo lo que hemos logrado.

A mi pareja, por su apoyo incondicional y por siempre tener una respuesta para cada problema que surgía. A mis amigos, a los que estuvieron y a los que todavía están, gracias por caminar a mi lado y por todos los recuerdos que me hacen hoy ser quien soy.

Gracias a mi familia, en especial a mi hermano pequeño por ser el mejor cómplice y sin apenas darse cuenta, ser una de mis mayores fuentes de apoyo y aprendizaje.

A la niña que fui, por siempre seguir adelante y luchar por tus sueños. Gracias por creer, aunque a veces dudaras, que podrías hacer de cualquier camino el mío. Este logro también es tuyo.

En definitiva, a cada persona que ha compartido un trocito de su vida conmigo, gracias.

Elena Campanero Belda

# Resumen

---

Este Trabajo de Fin de Grado ha consistido en la concepción, diseño e implementación de EduWordle, una aplicación web educativa interactiva. Su fin primordial es fomentar y reforzar el aprendizaje de temáticas especificadas por el profesorado a través de la gamificación, utilizando una dinámica similar al popular juego *Wordle* e incorporando un componente de evaluación formativa mediante preguntas asociadas.

En el desarrollo se ha tenido en cuenta el tipo de usuarios que trabajarán con la herramienta por lo que se ha priorizado una interfaz de usuario intuitiva y fácil de manejar para los profesores ya que éstos no han de requerir de conocimiento técnico para usarla. Los principales objetivos del sistema giran en torno a la simplificación de creación de contenido educativo personalizado, la gestión eficiente de grupos de alumnos y el seguimiento detallado de su progreso.

Respecto al alumnado, se ha aprovechado su familiaridad con las tecnologías digitales para ofrecerles un enfoque atractivo y dinámico para el estudio dando como resultado una transformación del aprendizaje en una actividad lúdica. Por ello EduWordle presenta una estrategia efectiva para mejorar la retención de conocimientos y la motivación.

El desarrollo ha consistido en la implementación de una arquitectura en tres capas favoreciendo la modularidad, escalabilidad y mantenibilidad del código. El sistema se ha construido usando HTML, JavaScript y CSS en el *frontend* para proporcionar una experiencia de usuario fluida y reactiva. El *backend* se ha desarrollado con Node.js y Express.js garantizando un procesamiento eficiente de las peticiones. La persistencia de datos se gestiona mediante MySQL y ORM Sequelize mientras que la seguridad se ha abordado con autenticación JWT y cifrado de contraseñas con Bcrypt. Finalmente se ha optado por una contenerización en Docker para asegurar la portabilidad y facilidad de despliegue del sistema en diversos entornos.

**Palabras clave:** Gamificación, Educación, Aplicación Web, Wordle, Node.js, Express.js, MySQL

# Abstract

---

This Final Degree Project has involved the conception, design and implementation of EduWordle, an interactive educational web application. Its primary aim is to encourage and reinforce the learning of a specific topic defined by educators through gamification. EduWordle uses a similar dynamic to the popular game Wordle and incorporates a formative assessment component via associated questions.

During development, the tool's target users, teachers and students, were carefully considered. For example, an intuitive and easy-to-use user interface was developed, to ensure all teachers, including those without technical knowledge, could easily operate the application. The system's main objectives revolve around simplifying the creation of personalized educational content, efficient management of student groups and detailed tracking of their progress.

Students' familiarity with digital technologies has been leveraged to offer an attractive and dynamic approach to study, resulting in the transformation of learning into a recreational activity. Therefore, EduWordle offers teachers and students an effective strategy for improving knowledge retention and motivation.

EduWordle's development consisted of the implementation of a three layered architecture which supports code modularity, scalability and maintainability. The system's *frontend* was built using HTML, JavaScript and CSS to provide a fluid and reactive user experience. The *backend* was developed with Node.js and Express.js, resulting in an efficient request processing. Data persistence is managed using MySQL and Sequelize ORM, while security has been addressed with JWT authentication and Bcrypt password encryption. Finally, Docker containerization was chosen to ensure the system's portability and ease of deployment across various environments.

**Keywords:** Gamification, Education, Web Application, Wordle, Node.js, Express.js, MySQL.

# Índice general

---

<b>1. Introducción.....</b>	<b>13</b>
1.1. Motivación.....	14
1.2. Objetivos.....	15
<b>2. Estado de la cuestión.....</b>	<b>17</b>
2.1. Gamificación en la Educación.....	17
2.1.1. Elementos de la Gamificación.....	18
2.1.2. Beneficios y Desafíos.....	19
2.2. Juegos de palabras educativos.....	20
2.2.1. Wordle.....	20
2.2.2. Ejemplos actuales de soluciones similares.....	21
<b>3. Ecosistema tecnológico.....</b>	<b>23</b>
3.1. MySQL.....	23
3.2. Docker.....	24
3.3. GitHub.....	24
3.4. Visual Studio Code.....	25
3.5. Node.js.....	25
3.6. Express.js.....	26
3.7. Sequelize CLI.....	26
3.8. JSON Web token.....	27
3.9. Bcrypt.js.....	27
3.10. Nodemailer.....	27
3.11. JavaScript.....	28
3.12. CSS.....	29
3.13. Swagger.....	29
<b>4. Proceso de construcción de EduWordle.....</b>	<b>30</b>
4.1. Modelo de Proceso Software.....	30
4.1.1. Reunión inicial.....	31
4.1.2. Planificación y Gestión de Hitos.....	31
4.2. Extracción de Requisitos.....	32
4.2.1. Entrevista.....	32

4.2.2. Primer prototipo.....	33
4.2.3. Requisitos funcionales.....	34
Gestión de Usuarios.....	34
Gestión de Wordles.....	36
Gestión de Grupos.....	42
Gestión de Rankings.....	46
Gestión de Archivos.....	46
4.2.4. Explicación del funcionamiento del juego.....	47
4.2.5. Diagramas de Casos de Uso.....	50
Gestión de Usuarios.....	51
Gestión de Wordles.....	52
Gestión de Grupos.....	54
4.3. Análisis.....	57
4.3.1. Modelo de Datos.....	58
4.3.2. Diagramas de clases.....	59
4.3.3. Diagramas de secuencia.....	60
4.4. Diseño.....	62
4.4.1. Diagrama de clases de diseño.....	63
4.4.2. Patrones de diseño.....	63
Singleton.....	63
Factory Method.....	64
4.4.3. Arquitectura de la solución.....	64
Patrón arquitectónico.....	64
Estilo arquitectónico.....	65
Flujo de interacción entre capas.....	66
4.4.4. Atributos de Calidad.....	66
4.5. Implementación.....	68
4.5.1. Hito 1: Desarrollo Frontend.....	68
Uso de placeholders y plantillas.....	69
Gestión de elementos asíncronos.....	69
Uso de live-server y mock de backend.....	70
4.5.2. Hito 2: Desarrollo Backend.....	71
Implementación de la base de datos.....	73
4.5.3. Hito 3: Dockerización.....	75
<b>5. EduWordle.....</b>	<b>80</b>

5.1. Manual de Usuario y flujo de navegación.....	80
5.1.1. Inicio de Sesión.....	80
5.1.2. Gestión de Contraseñas.....	83
5.1.3. Creación y Edición de Wordles.....	85
5.1.4. Gestión de Grupos.....	91
5.1.5. Visualización de Resultados.....	98
5.1.6. Jugar Wordle.....	100
5.2. Validación : pruebas de aceptación.....	104
5.2.1. Introducción de Datos de Semilla.....	105
5.2.2. Escenarios de demostración.....	105
Escenario 1: Profesor.....	105
Escenario 2: Estudiante.....	106
5.3. Despliegue.....	107
5.3.1. Recursos del Proyecto.....	107
Acceso Web.....	107
Acceso Repositorio.....	108
<b>6. Conclusiones.....</b>	<b>109</b>
<b>7. Aspectos sociales, ambientales, éticos y profesionales.....</b>	<b>111</b>
<b>8. Líneas Futuras.....</b>	<b>113</b>
<b>9. Bibliografía.....</b>	<b>115</b>
<b>10. Anexos.....</b>	<b>118</b>

# Índice de figuras

---

Figura 3.1: Diagrama de las herramientas principales de Desarrollo.....	23
Figura 3.2: Flujo de trabajo en GitHub.....	25
Figura 3.3: Ejemplo de Correo de registro de Alumno.....	28
Figura 4.1: Modelo en cascada con un enfoque de prototipado evolutivo.....	31
Figura 4.2: Prototipo inicial.....	34
Figura 4.3: Diagrama de actividad del funcionamiento del juego original.....	48
Figura 4.4: Diagrama de actividad del juego de EduWordle con capa educativa.....	50
Figura 4.5: Diagrama caso de uso Gestión de Usuarios.....	52
Figura 4.6: Diagrama caso de uso Gestión de Wordles.....	54
Figura 4.7: Diagrama caso de uso Gestión de grupos.....	57
Figura 4.8: Modelo Entidad-Relación.....	58
Figura 4.9: Diagrama paso a tablas.....	59
Figura 4.10: Diagrama de clases.....	60
Figura 4.11: Diagrama de secuencia Petición de Login.....	61
Figura 4.12: Diagrama de secuencia Petición genérica.....	62
Figura 4.13: Diagrama de secuencia Petición datos de juego y guardado de resultados.....	62
Figura 4.14: Diagrama clases de Diseño. Enlace directo.....	63
Figura 4.15: Diagrama de Arquitectura final del Proyecto.....	65
Figura 4.16: Diagrama de Secuencia de la Interacción Cliente-Servidor.....	66
Figura 4.17: Fragmento de game.html - placeholder del tablero.....	69
Figura 4.18: Fragmento de game.js - Ejemplo de lógica asíncrona.....	70
Figura 4.19: Fragmento del servidor mock server.js.....	71
Figura 4.20: Fragmento de game.js - Llamada fetch al servidor simulado.....	71
Figura 4.21: Fragmento de gameService.js - Ejemplo de rollback.....	72
Figura 4.22: Fragmento de studentRoutes.js - Rutas protegidas y manejo de resultados de juego.....	73
Figura 4.23: Fragmento de gameResult.js - Definición del modelo de partida.....	74
Figura 4.24: Ingeniería inversa: diagrama de la Base de Datos.....	75
Figura 4.25: Fragmento de Dockerfile - Definición del servicio de backend.....	76

Figura 4.26: Fragmento de entrypoint.sh - Script de espera.....	77
Figura 4.27: Fragmento de docker-compose.yml.....	78
Figura 5.1: Página de inicio.....	81
Figura 5.2: Página de login.....	81
Figura 5.3: Forzado de cambio de contraseña.....	82
Figura 5.4: Dashboard Profesor.....	82
Figura 5.5: Dashboard Alumno.....	83
Figura 5.6: Login erróneo.....	83
Figura 5.7: Camino hacia configuración desde Dashboard de Profesor.....	84
Figura 5.8: Página de cambio de contraseña.....	84
Figura 5.9: Página de cambio de contraseña con mensaje de éxito.....	85
Figura 5.10: Camino hacia Lista de Wordles desde Dashboard de Profesor.....	86
Figura 5.11: Camino hacia creación de nuevo Wordle desde Lista de Wordles.....	86
Figura 5.12: Página de creación de Wordle.....	87
Figura 5.13: Mensaje de éxito de creación de Wordle.....	88
Figura 5.14: Camino a visualización de Wordle desde Lista de Wordles.....	88
Figura 5.15: Botón de Editar en página de visualización de Wordle.....	89
Figura 5.16: Página de edición de Wordle.....	89
Figura 5.17: Botón de eliminar en la pantalla de visualizar Wordle.....	90
Figura 5.18: Mensaje de confirmación de eliminación de Wordle.....	91
Figura 5.19: Camino hacia Lista de grupos desde Dashboard de Profesor.....	92
Figura 5.20: Camino hacia creación de nuevo Grupo desde Lista de Grupos.....	92
Figura 5.21: Página de creación de Grupo.....	93
Figura 5.22: Mensaje de éxito de creación de Grupo.....	94
Figura 5.23: Camino a visualización de Grupo desde Lista de Grupos.....	94
Figura 5.24: Botón de Editar en página de visualización de Grupo.....	95
Figura 5.25: Página de edición de Grupo.....	96
Figura 5.26: Botón de eliminar en la pantalla de visualizar Grupo.....	97
Figura 5.27: Mensaje de confirmación de eliminación de Grupo.....	97
Figura 5.28: Camino hacia Clasificaciones desde Dashboard de Profesor.....	98
Figura 5.29: Menú de clasificaciones.....	99
Figura 5.30: Pantalla de Clasificaciones por Grupo.....	99
Figura 5.31: Pantalla de Clasificaciones por Wordle.....	100
Figura 5.32: Camino hacia Lista de Wordles desde Dashboard de Alumno.....	101
Figura 5.33: Lista de Wordles de Alumno.....	101

Figura 5.34: Pantalla de Juego.....	102
Figura 5.35: Popup de Pregunta simple.....	102
Figura 5.36: Mensaje de éxito y pista.....	103
Figura 5.37: Popup de varias preguntas.....	103
Figura 5.38: Coloración de letras acertadas.....	104
Figura 5.39: Mensaje de éxito al acertar una palabra.....	104
Figura 5.40: Video de ejemplo de Eduwordle Profesor. Enlace directo.....	106
Figura 5.41: Video de ejemplo de Eduwordle Estudiante. Enlace directo.....	107
Figura 5.42: Acceso a Eduwordle. Enlace directo.....	108
Figura 5.43: Acceso al repositorio en GitHub. Enlace directo.....	108
Figura A.1: Documento de Entrevista inicial. Enlace directo.....	118

# Índice de tablas

---

Tabla 1: Requisitos de Gestión de Usuarios.....	36
Tabla 2: Requisitos de Gestión de Wordles.....	41
Tabla 3: Requisitos de Gestión de Grupos.....	45
Tabla 4: Requisitos de Gestión de Rankings.....	46
Tabla 5: Requisitos de Gestión de Archivos.....	47
Tabla 6: Tabla de trazabilidad entre Requisitos Funcionales y Casos de Uso Gestión Usuarios.....	51
Tabla 7: Tabla de trazabilidad entre Requisitos Funcionales y Casos de Uso Gestión Wordles.....	53
Tabla 8: Tabla de trazabilidad entre Requisitos Funcionales y Casos de Uso Gestión Grupos.....	56
Tabla 9: Atributos de Calidad.....	68
Tabla 10: Tabla LOS Retrospectiva 1.....	124
Tabla 11: Tabla SSK Retrospectiva 1.....	124
Tabla 12: Tabla LOS Retrospectiva 2.....	126
Tabla 13: Tabla SSK Retrospectiva 2.....	126
Tabla 14: Tabla LOS Retrospectiva 3.....	128
Tabla 15: Tabla SSK Retrospectiva 3.....	128
Tabla 16: Tabla LOS Retrospectiva 4.....	130
Tabla 17: Tabla SSK Retrospectiva 4.....	130
Tabla 18: Tabla LOS Retrospectiva 5.....	132
Tabla 19: Tabla SSK Retrospectiva 5.....	132
Tabla 20 Tabla LOS Retrospectiva 6.....	134
Tabla 21: Tabla SSK Retrospectiva 6.....	135
Tabla 22: Tabla LOS Retrospectiva 7.....	137
Tabla 23: Tabla SSK Retrospectiva 7.....	137
Tabla 24: Semilla: cuentas de usuario con rol profesor.....	138
Tabla 25: Semilla: palabras Wordle 1.....	140
Tabla 26: Semilla: preguntas Wordle 1.....	144
Tabla 27: Semilla: palabras Wordle 2.....	145
Tabla 28: Semilla: preguntas Wordle 2.....	148

Tabla 29: Semilla: palabras Wordle 3.....	149
Tabla 30: Semilla: preguntas Wordle 3.....	152
Tabla 31: Semilla: palabras Wordle 4.....	154
Tabla 32: Semilla: preguntas Wordle 4.....	158
Tabla 33: Semilla: palabras Wordle 5.....	159
Tabla 34: Semilla: preguntas Wordle 5.....	163

# 1. Introducción

---

La tecnología ha tenido un impacto positivo en numerosos aspectos de nuestra vida cotidiana, siendo la educación uno de los más destacados. Ha transformado profundamente la manera en que aprendemos y accedemos a la información, facilitando nuevas formas de enseñanza y aprendizaje. [1]

En la actualidad, el sistema educativo se enfrenta a múltiples desafíos que deben abordarse aprovechando al máximo las oportunidades que ofrece la tecnología. Vivimos en una era caracterizada por un acceso sin precedentes a la información y al conocimiento, así como por una amplia disponibilidad de recursos y medios digitales.[2]

En este contexto, promover el uso de tecnologías en el ámbito educativo ha dado lugar al desarrollo de nuevas estrategias de aprendizaje, adaptadas a una generación de estudiantes inmersa en un entorno tecnológico. Estas estrategias priorizan metodologías innovadoras, entre ellas las de carácter lúdico, que buscan estimular la participación y el interés del alumnado.

La motivación es un factor clave en el proceso de aprendizaje y resulta determinante para la obtención de buenos resultados académicos. Sin embargo, mantener altos niveles de motivación representa un desafío para el profesorado, especialmente ante los cambios generacionales y el avance tecnológico. Como respuesta a esta necesidad, han surgido metodologías como la *gamificación*.

La *gamificación* es una técnica de aprendizaje que comenzó a popularizarse en el ámbito del *marketing* en torno al año 2008, con el objetivo de fomentar la interacción de los clientes con las marcas. En el campo educativo, esta técnica adapta la mecánica de los juegos a contextos de enseñanza-aprendizaje con el propósito de mejorar la adquisición de conocimientos, potenciar habilidades específicas y aumentar la motivación del alumnado.[3] [4]

Esta metodología permite interiorizar contenidos de manera más amena y divertida, lo que genera una experiencia positiva para el estudiante y promueve su implicación activa y el deseo de superación.

Algunas de las técnicas de *gamificación* más comunes incluyen:

- **Acumulación de puntos:** Asignación de valores cuantitativos a acciones específicas.
- **Obtención de premios o recompensas:** Reconocimiento tras alcanzar determinados objetivos.
- **Desafíos:** Competencias entre estudiantes para superar metas concretas.
- **Clasificaciones o *rankings*:** Ordenación del alumnado según los puntos obtenidos, fomentando un ambiente de sana competencia.

En este contexto, el juego *Wordle* se presenta como una herramienta potencial para la gamificación del aprendizaje. Se trata de un juego en el que los jugadores deben descubrir una palabra oculta en un máximo de seis intentos. Tras cada intento, el sistema proporciona retroalimentación mediante colores: verde si la letra está en la posición correcta, amarillo si la letra está en la palabra pero en una posición incorrecta, y gris si la letra no forma parte de la palabra. [5]

La adaptación de *Wordle* como recurso educativo *gamificado* constituye el eje central de este Trabajo de Fin de Grado (TFG), con el objetivo de contribuir a la innovación tecnológica en el ámbito educativo y ofrecer una alternativa lúdica y motivadora para el proceso de enseñanza-aprendizaje.

## 1.1. Motivación

La creación de este TFG surge de la necesidad de abordar ciertos desafíos del ámbito educativo, como la falta de motivación por parte del alumnado actual y la escasa integración de herramientas tecnológicas con un enfoque lúdico en el aula educativa.

En este momento el alumnado de cualquier nivel educativo vive inmerso en un mundo digitalizado que va a seguir avanzando en esta línea; sin embargo, hay muchas aulas que aún no han integrado de forma adecuada y segura la gran cantidad de recursos tecnológicos actualmente disponibles. En la mayor parte de las aulas siguen predominando métodos tradicionales de aprendizaje que no

motivan ni atraen al alumnado hacia la adquisición de aprendizajes, especialmente en edades tempranas. Esto provoca una brecha entre el uso cotidiano de la tecnología por parte del alumnado y su uso en el aula.

El uso de la *gamificación* abre la puerta a la mejora de adquisición de aprendizajes con experiencias dinámicas y atractivas que favorezcan y estimulen la participación activa, el aprendizaje autónomo, la motivación, la superación personal y a la vez refuercen capacidades y habilidades como la memoria.

En este trabajo, se propone una plataforma *gamificada* basada en el popular juego *Wordle*, adaptada al ámbito educativo, con ella el alumnado practicará vocabulario, definiciones o conceptos claves relacionados con las asignaturas impartidas en el aula. EduWordle, una plataforma lúdica y sencilla para el alumnado, además de una herramienta intuitiva, fácil y adaptable a las necesidades pedagógicas de cualquier profesorado, tenga el nivel competencial digital que tenga, lo que la hace una plataforma muy versátil.

Con este TFG se pretende facilitar una solución accesible que combina tecnología, juego y aprendizaje. Pretendemos facilitar la integración de la tecnología en el aula, hacer de ésta un lugar motivador para las nuevas generaciones a las que se enfrenta el profesorado actual y con ella contribuir a la mejora de los procesos de enseñanza-aprendizaje actuales.

## 1.2. Objetivos

El presente Trabajo de Fin de Grado tiene como principal objetivo desarrollar una plataforma educativa basada en *Wordle*, diseñada para mejorar el aprendizaje mediante juegos interactivos. Esta plataforma presenta las siguientes funcionalidades:

- Creación y gestión de *Wordles* por parte del profesorado, con preguntas y palabras relacionadas con el contenido curricular.
- Creación y gestión de grupos de alumnos por parte de los docentes, pudiendo asignar a cada grupo los *Wordles* correspondientes.

- Posibilidad de que los alumnos jueguen a los *Wordles* asignados a los grupos a los que pertenecen, fomentando así la participación activa y el aprendizaje lúdico.
- Seguimiento del rendimiento del alumnado a través de un sistema de clasificación o *ranking*, tanto por *Wordle* como por grupo.
- Visualización, por parte del alumnado, de los grupos en los que están inscritos.

## 2. Estado de la cuestión

---

Con el objetivo de fundamentar teóricamente el desarrollo de la aplicación EduWordle, en este apartado se analiza el estado actual de la investigación y las prácticas relacionadas con la *gamificación* educativa. Para ello, se revisan los conceptos clave que sustentan esta metodología, los elementos que la componen, sus beneficios y desafíos, así como su aplicación concreta en el contexto de los juegos de palabras. Además, se estudia el caso del juego Wordle como referente lúdico de gran impacto y se presentan algunas soluciones similares disponibles en el mercado.

### 2.1. Gamificación en la Educación

En las últimas dos décadas la *gamificación* se ha convertido en una estrategia consolidada en el ámbito académico. A diferencia de los métodos tradicionales de enseñanza, esta incorpora dinámicas y elementos de juegos con el fin de motivar al alumnado, así como aumentar su participación en el aula. Su uso está extendido entre los distintos niveles académicos, desde la educación primaria hasta la universidad.

Aunque su origen está vinculado con el *marketing*, donde se empleaba para fidelizar a los clientes mediante mecánicas lúdicas, su potencial ha sido reconocido por la comunidad educativa, en este ámbito está orientado a la adquisición o refuerzo de nuevos conocimientos y a la motivación de los alumnos.

Diversas investigaciones respaldan esta metodología. Sarabia-Guevara y Bowen-Mendoza, en una revisión sistemática de estudios entre 2015 y 2022, concluyen que la *gamificación* mejora significativamente el interés y la retención de conocimientos en estudiantes universitarios de carreras de ingeniería. Además, los autores subrayan que los beneficios se ven potenciados cuando su uso se alinea con los objetivos curriculares. Estos hallazgos, centrados en el ámbito de la ingeniería,

confirman su versatilidad como herramientas en las distintas áreas de conocimiento. [6]

No obstante, su aplicación no está exenta de retos. Su eficacia depende de la calidad del diseño didáctico, la contextualización y la preparación del docente. Si estos aspectos no se abordan adecuadamente, existe el riesgo de que su uso pierda la finalidad educativa y se limite a entretener.

### 2.1.1. Elementos de la Gamificación

La *gamificación* en contextos educativos se apoya en una serie de elementos fundamentales que permiten transformar la experiencia de aprendizaje en una dinámica más atractiva y motivadora. Estos componentes, no solo estimulan el interés del alumnado, sino que también refuerzan la adquisición de conocimientos y habilidades a través de la práctica y el refuerzo positivo.

Según Sánchez-Pacheco (2019) [7], entre los elementos más efectivos en entornos educativos se encuentran:

- **Recompensas:** La obtención de puntos, insignias o trofeos funciona como incentivo para el esfuerzo y el logro. Estas recompensas, al estar asociadas a metas específicas, ayudan a mantener el compromiso del estudiante y a reforzar conductas deseadas dentro del proceso de aprendizaje.
- **Libertad de elección:** Ofrecer al alumnado la posibilidad de seleccionar actividades o rutas dentro de un entorno *gamificado* estimula la autonomía y el sentido de control. La percepción de libertad incrementa la implicación y favorece un mejor aprendizaje.
- **Desafíos y niveles de dificultad:** Proponer retos progresivos permite mantener el interés y evitar tanto la monotonía como la frustración. El diseño de niveles escalonados facilita la adaptación a los distintos ritmos de aprendizaje y genera una sensación de logro constante.
- **Clasificaciones o *rankings*:** Estos sistemas fomentan la competitividad positiva entre los estudiantes y visibilizan el esfuerzo y la evolución individual. No obstante, su aplicación debe realizarse con cautela para no generar comparaciones contraproducentes o desmotivación en quienes se sitúan en los últimos puestos.

En el diseño de una herramienta *gamificada* como EduWordle, la integración de estos elementos permite no solo captar la atención del alumnado, sino también estructurar la actividad de forma que cada mecánica tenga un propósito educativo claro. Al aprovechar la lógica de los juegos para organizar el contenido, se genera una experiencia más activa, personalizada y motivadora para los estudiantes.

### 2.1.2. Beneficios y Desafíos

La incorporación de la *gamificación* en el entorno educativo ha generado un interés entre los docentes debido a su capacidad para mejorar la experiencia de aprendizaje. Bien aplicada, puede influir positivamente en los aspectos clave que hemos comentado anteriormente del proceso educativo, la motivación, la participación activa y la retención de contenidos.

Entre los beneficios más destacados se encuentra el aumento de la motivación del alumnado. Al integrar dinámicas propias del juego, se transforma el aprendizaje en una actividad más atractiva, lo que facilita el compromiso sostenido y el interés por los contenidos curriculares. La participación activa es otro efecto positivo ampliamente documentado, ya que el estudiante deja de tener un rol pasivo para convertirse en protagonista de su propio proceso de aprendizaje.

La retroalimentación inmediata, característica común en muchas plataformas *gamificadas*, permite a los alumnos identificar errores y aciertos en tiempo real, favoreciendo la autorregulación y el aprendizaje autónomo. Asimismo, la estructura de niveles, desafíos o recompensas progresivas estimula la superación personal y el establecimiento de metas individuales, lo que refuerza el sentido de logro.

No obstante, también existen desafíos y limitaciones que deben tenerse en cuenta al implementar la *gamificación* en el aula. Uno de los principales riesgos es que el componente lúdico desplace los objetivos educativos si no se planifica adecuadamente. Cuando se prioriza la competición o la recompensa por encima del aprendizaje, puede perderse el enfoque pedagógico y reducir la eficacia de la actividad.

Además, el uso de *rankings* o sistemas de puntuación puede resultar contraproducente si genera ansiedad, frustración o comparaciones negativas entre estudiantes.

Otro desafío importante es la formación del profesorado. No todos los docentes disponen de las competencias digitales necesarias para diseñar o aplicar experiencias *gamificadas* eficaces, lo que limita su inclusión. Asimismo, muchas herramientas existentes en el mercado presentan barreras técnicas o interfaces complejas, que dificultan su integración real en la práctica docente diaria.

En este contexto, resulta fundamental disponer de plataformas accesibles, intuitivas y adaptables, que permitan a cualquier docente, independientemente de su nivel técnico. La propuesta de EduWordle responde precisamente a esta necesidad, ofreciendo un recurso que combina simplicidad de uso con un diseño pedagógico basado en los elementos clave de la *gamificación*.

## 2.2. Juegos de palabras educativos

Tras haber revisado los fundamentos teóricos de la *gamificación*, sus elementos principales y los beneficios y desafíos de su implementación educativa, resulta pertinente analizar cómo se han materializado estas ideas en herramientas concretas. En este sentido, los juegos de palabras representan un recurso didáctico con amplia tradición en el ámbito educativo, especialmente útil para reforzar el aprendizaje de vocabulario, ortografía y competencias lingüísticas.

Entre los distintos formatos existentes, el juego Wordle ha alcanzado una enorme popularidad por su sencillez y efectividad, convirtiéndose en un modelo de referencia para aplicaciones *gamificadas*. A continuación, se expone brevemente su origen y evolución, así como un análisis de otras plataformas similares actualmente disponibles en el mercado, con el fin de contextualizar y justificar la propuesta desarrollada en este TFG.

### 2.2.1. Wordle

El juego Wordle fue creado en octubre de 2021 por el ingeniero de *software* Josh Wardle como un pasatiempo personal. Su estructura simple, adivinar una palabra de cinco letras en seis intentos, unida a una estética minimalista y la posibilidad de compartir los resultados en redes sociales, propició su viralización. A principios de

2022, el juego fue adquirido por The New York Times, consolidando su estatus como fenómeno cultural global. [8]

El éxito de Wordle se puede explicar por varios factores: su mecánica es accesible, no requiere instalación ni registro, ofrece una experiencia diaria limitada (una palabra por día), y proporciona un tipo de reto que combina deducción, lenguaje y lógica. Además, su sistema de retroalimentación mediante colores (verde, amarillo, gris) facilita el aprendizaje por ensayo y error, y potencia la motivación intrínseca.

Estas características han favorecido la aparición de múltiples adaptaciones temáticas (como Worldle, Nerdle o Mathler) y han inspirado iniciativas educativas que aprovechan su dinámica para trabajar contenidos escolares. Su estructura simple, clara y repetitiva lo convierte en una excelente base para diseñar actividades *gamificadas* centradas en la adquisición de vocabulario, conceptos disciplinares o competencias lingüísticas.

### 2.2.2. Ejemplos actuales de soluciones similares

En el panorama actual existen diversas plataformas que combinan tecnología y *gamificación* con fines educativos. Algunas de las más reconocidas son:

*Kahoot*: centrada en concursos de preguntas multijugador, muy utilizada en contextos escolares por su dinamismo y facilidad de uso. [9]

*Quizizz*: similar a Kahoot, pero con mayor énfasis en el trabajo autónomo y el seguimiento del rendimiento del alumnado. [10]

*Wordwall*: permite generar juegos de palabras interactivos (como emparejar, completar o clasificar) aunque con opciones más limitadas en personalización y control de grupo. [11]

Si bien estas herramientas han supuesto un avance notable en la incorporación del juego al aula, muchas presentan limitaciones como la rigidez en el tipo de contenidos, la escasa personalización, o la falta de seguimiento individualizado del rendimiento del estudiante.

En este contexto, EduWordle se presenta como una solución innovadora que combina las ventajas del formato Wordle con funcionalidades diseñadas

específicamente para el ámbito educativo: asignación de juegos a grupos, edición de contenido por parte del docente, retroalimentación inmediata, y seguimiento del progreso mediante *rankings*. Todo ello desde una interfaz intuitiva y accesible, que facilita su integración en el aula sin necesidad de conocimientos técnicos avanzados por parte del profesorado.

# 3. Ecosistema tecnológico

Para el desarrollo del proyecto, se ha seleccionado un conjunto de herramientas para abarcar desde la gestión de la base de datos hasta el entorno de desarrollo y control de versiones. En esta sección se detallan las principales herramientas empleadas y su rol en la construcción del sistema.

La siguiente figura representa de manera visual un esquema de las herramientas utilizadas y su implicación y relación dentro del sistema.

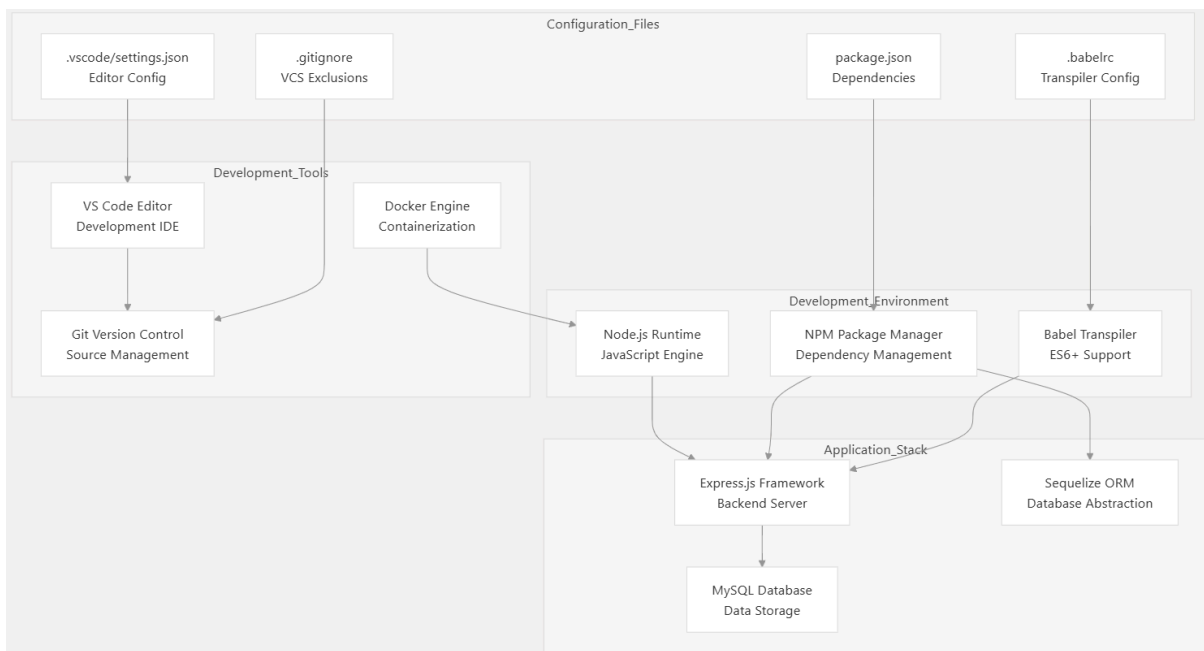


Figura 3.1: Diagrama de las herramientas principales de Desarrollo

## 3.1. MySQL

MySQL es un sistema de gestión de bases de datos relacionales de código abierto, reconocido por su rendimiento, fiabilidad y facilidad de uso. Fue creado en 1994 y actualmente es uno de los sistemas de bases de datos más populares del mundo,

utilizado por grandes empresas y proyectos de código abierto para gestionar datos estructurados [12].

Para el desarrollo de este proyecto, se ha decidido usar para almacenar toda la información relevante de la aplicación. Su naturaleza relacional ha permitido modelar de manera eficiente las complejas relaciones entre entidades, asegurando la integridad y consistencia de los datos.

## 3.2. Docker

Docker es una plataforma de virtualización ligera que permite empaquetar aplicaciones y sus dependencias en contenedores, garantizando una ejecución uniforme en cualquier entorno. [13]

En consonancia con ello, se ha hecho uso de Docker Compose para orquestar el correcto funcionamiento de aplicaciones que hacen uso de varios contenedores Docker y se coordinan usando un único archivo YAML. Así se optimiza el proceso de desarrollo, pruebas y despliegue sin necesidad de gestionar manualmente cada contenedor.

Ha sido una herramienta fundamental para estandarizar el entorno de desarrollo ya que ha permitido ejecutar el sistema en diferentes dispositivos a la hora de desarrollar el proyecto y ha eliminado los problemas de inconsistencia. Además, para poder ejecutarlo en el servidor de la UPM, era un requisito que el proyecto estuviera *dockerizado*.

## 3.3. GitHub

GitHub es una plataforma basada en la nube que permite el alojamiento de repositorios y control de versiones, facilitando la colaboración en proyectos de *software* y la gestión de código fuente [14]. Fundada en 2008, es actualmente la plataforma de desarrollo colaborativo más grande del mundo, esencial para proyectos de código abierto y equipos de desarrollo profesionales.

Esta herramienta ha permitido llevar un registro detallado de cada cambio realizado, vital para localizar errores derivados por dichos cambios, y facilitar la colaboración. La siguiente figura ilustra cómo se trabaja en local y con ayuda de Git se sube al repositorio online en GitHub los cambios y se trabaja simultáneamente.

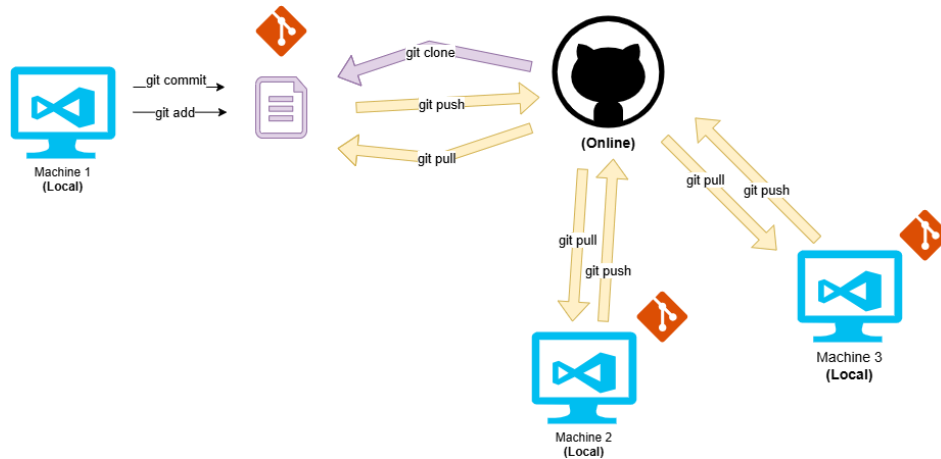


Figura 3.2: Flujo de trabajo en GitHub

## 3.4. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, compatible con múltiples lenguajes de programación y extensible mediante plugins. [15]

Esta herramienta ha sido el entorno de desarrollo principal para la codificación de EduWordle. Su riqueza en extensiones para JavaScript, Node.js, HTML, CSS y SQL junto con sus capacidades integradas para Git, depuración y terminal, han optimizado significativamente la productividad durante todo el ciclo de desarrollo.

## 3.5. Node.js

Es un entorno de ejecución de JavaScript de código abierto, multiplataforma, que permite ejecutar código fuera del navegador.

Node.js es el pilar del *backend* de EduWordle. Se utiliza para construir la API REST que maneja todas las solicitudes del *frontend*, gestiona la lógica de negocio, interactúa con la base de datos y provee los servicios necesarios para el funcionamiento de la aplicación. Su naturaleza asíncrona y basada en eventos lo hace ideal para manejar múltiples conexiones concurrentes de manera eficiente.

## 3.6. Express.js

Express.js es un *framework* de aplicaciones web minimalista y flexible para Node.js, que proporciona un conjunto robusto de características para construir aplicaciones web y APIs.

Facilita la definición de los endpoints que comunican el *frontend* con el *backend*, la gestión de solicitudes HTTP, la aplicación de *middleware* para la autenticación y validación de datos, y la estructuración del código del servidor de manera modular y escalable.

## 3.7. Sequelize CLI

Sequelize CLI es una herramienta de línea de comandos asociada al ORM (Object-Relational Mapper) Sequelize, que simplifica las tareas relacionadas con la gestión de bases de datos. Desarrollada como parte del ecosistema Sequelize, se utiliza comúnmente para automatizar y estandarizar la creación de migraciones y *seeders*, esenciales en el ciclo de vida de una aplicación con base de datos.

En el proyecto se ha utilizado para generar y ejecutar migraciones que permiten evolucionar el esquema de la base de datos de manera controlada y versionada. Asimismo, se emplea para la creación de *seeders* que facilitan la inserción de datos iniciales y de prueba, lo cual es muy útil para la configuración del entorno de desarrollo y las demostraciones.

## 3.8. JSON Web token

JWT es un estándar abierto para crear *tokens* de acceso y en este caso, el mecanismo principal de autenticación y autorización de EduWordle. Tras el inicio de sesión exitoso de un usuario (profesor o estudiante), se genera un JWT que se envía al cliente. Este *token* es luego utilizado en cada solicitud subsiguiente a la API para verificar la identidad del usuario y sus permisos, asegurando que solo los usuarios autenticados y autorizados puedan acceder a los recursos y funcionalidades correspondientes.

## 3.9. Bcrypt.js

Bcrypt.js es una librería para JavaScript que implementa el algoritmo de *hashing* de contraseñas Bcrypt, diseñado para ser computacionalmente lento y resistente a ataques de fuerza bruta.

En este proyecto se usa para realizar un *hash* sobre las contraseñas previamente a su almacenamiento en la base de datos.

## 3.10. Nodemailer

Nodemailer es un módulo para Node.js que simplifica el envío de correos electrónicos desde aplicaciones Node.js, compatible con una variedad de servicios de correo y protocolos [16].

Nodemailer se integra en el *backend* de EduWordle para gestionar el envío de correos electrónicos al registrar a un alumno por parte del profesor.

Esta figura muestra un ejemplo del correo enviado a los alumnos nuevos una vez el profesor guarde el nuevo grupo o los cambios en un grupo existente. La imagen es solo un ejemplo de estructura, texto y aspecto del mismo, sin embargo los mensajes actualmente están en español, no en inglés.

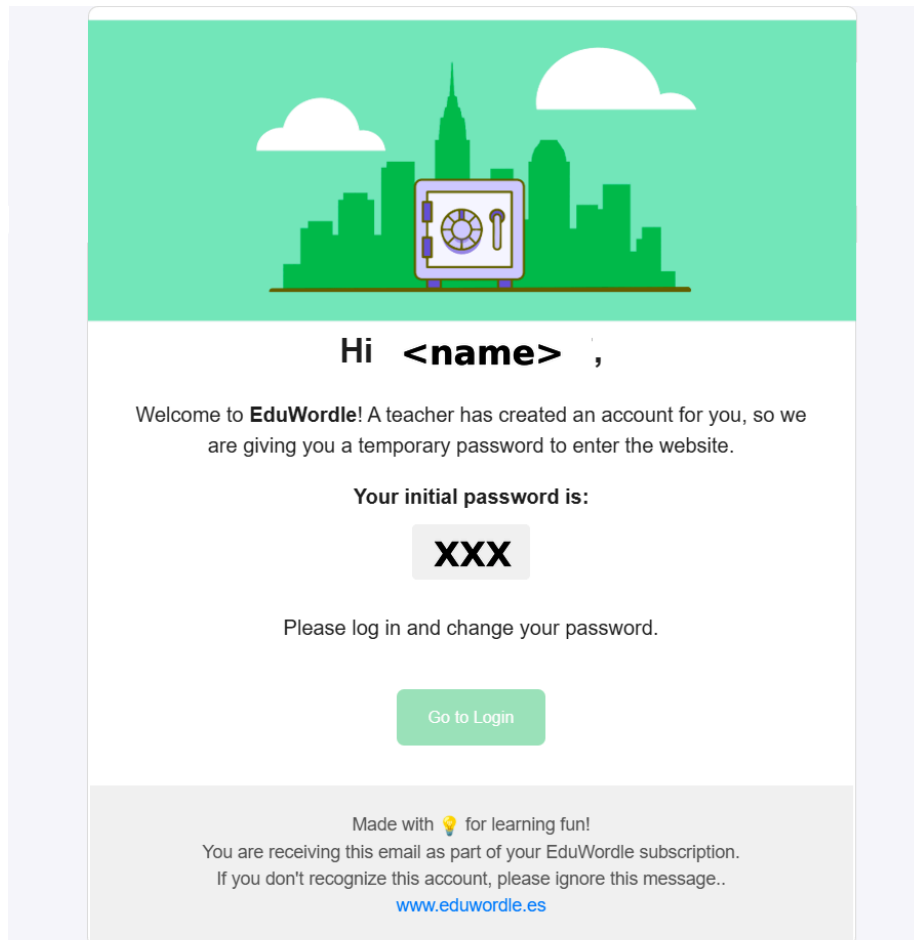


Figura 3.3: Ejemplo de Correo de registro de Alumno

## 3.11. JavaScript

Es un lenguaje de programación de alto nivel, interpretado y multiparadigma, fundamental para el desarrollo web.

JavaScript es el lenguaje principal utilizado en todo el *stack* de EduWordle. En el *frontend* gestiona la interactividad de la interfaz de usuario, las peticiones a la API y la lógica de presentación. En el *backend*, JavaScript implementa la lógica de negocio, el manejo de la base de datos y la gestión de la API REST.

## 3.12. CSS

CSS es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en un lenguaje de marcado como HTML.

Es esencial para el diseño visual y la experiencia de usuario de la interfaz de EduWordle. Se utiliza para aplicar estilos a todos los elementos HTML generados por el *frontend*, definiendo la tipografía, los colores, el espaciado, la disposición de los elementos y la responsividad para asegurar que la aplicación se vea atractiva y funcione correctamente en diferentes dispositivos y tamaños de pantalla.

## 3.13. Swagger

Swagger (basado en la OpenAPI Specification) es un estándar abierto para definir, crear, consumir y visualizar APIs REST, proporcionando un formato legible por humanos y por máquinas para describir las capacidades de una API.

Se ha implementado en el proyecto para generar una documentación interactiva de la API REST. Esto sirve como una referencia clara y concisa de todos los endpoints, parámetros de entrada, estructuras de respuesta y códigos de error, facilitando el desarrollo del *frontend* y la comprensión de la API por parte de cualquier persona que necesite interactuar con el *backend*. Además, su interfaz web permite realizar pruebas directas sobre la API. [\[17\]](#)

# 4. Proceso de construcción de EduWordle

---

En este apartado se detalla el desarrollo de EduWordle, una plataforma web educativa donde los profesores pueden crear *Wordles* personalizados, permitiéndoles presentar su materia de manera dinámica y divertida, los estudiantes podrán jugar a adivinar las palabras especificadas por sus profesores teniendo que responder preguntas del temario para obtener resultados.

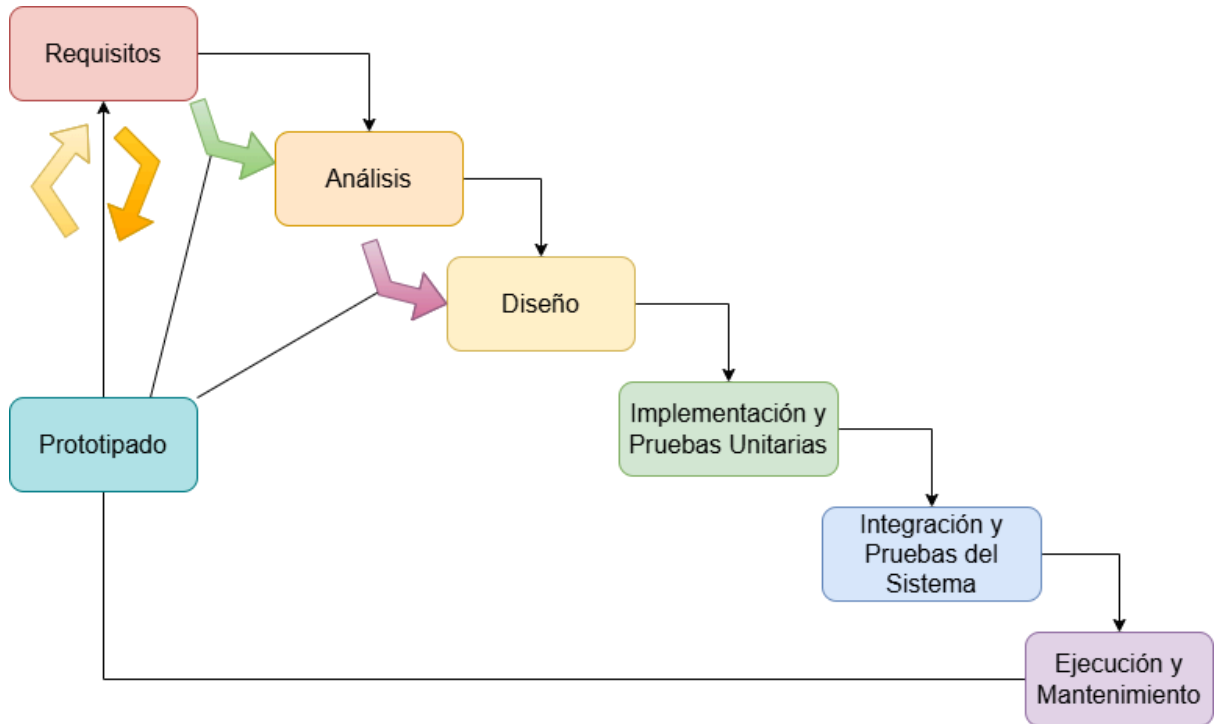
## 4.1. Modelo de Proceso Software

Para la gestión del ciclo de vida del *software* de EduWordle, se ha adoptado una estrategia híbrida que combina el **Modelo en Cascada** con un enfoque de **Prototipado Evolutivo**. Esta metodología ha permitido estructurar el desarrollo a través de fases secuenciales, mientras se integra la flexibilidad necesaria para incorporar retroalimentación y mejoras continuas.

Inicialmente, se ha procedido a través de las fases clásicas del Modelo en Cascada (análisis de requisitos, diseño, implementación y pruebas) hasta la obtención de un prototipo funcional de valor. Posteriormente, este prototipo ha servido como base para ciclos iterativos adicionales. En cada iteración, se han introducido mejoras y nuevas funcionalidades sobre el prototipo existente, repitiendo las fases del ciclo de vida del *software* hasta alcanzar el producto final deseado.

Se ha elegido este modelo porque permite ir verificando el avance del proyecto de forma más estructurada en cada fase, genera un producto de valor desde etapas tempranas y permite identificar y abordar posibles problemas de manera más fácil.

Para una mejor comprensión del proceso de desarrollo seguido, se presenta a continuación la siguiente figura a modo de ejemplo:



**Figura 4.1:** Modelo en cascada con un enfoque de prototipado evolutivo

#### 4.1.1. Reunión inicial

Al ser EduWordle un trabajo derivado de la unión de dos proyectos independientes, una reunión inicial estratégica fue un elemento clave para sentar las bases de su desarrollo. El acta detallada de esta sesión se encuentra disponible en el [Anexo 2: “Acta Reunión Inicial”](#)

El objetivo de esta reunión fue la alineación de objetivos, la unificación de requisitos y la definición de una visión común para EduWordle. Fruto de esta fase, se estableció un conjunto inicial de funcionalidades y un alcance del proyecto que servirían como base para la planificación futura.

#### 4.1.2. Planificación y Gestión de Hitos

La implementación del código se abordó dividiendo el conjunto del sistema en tres hitos principales de desarrollo:

1. Desarrollo del prototipo del *frontend*

2. Desarrollo del *backend*
3. Unión de ambas partes y *dockerización*
4. Despliegue
5. Gestión de errores y mantenimiento

Para la gestión y el control del progreso de cada hito, la carga de trabajo fue organizada en paquetes de trabajo. Se establecieron reuniones de seguimiento periódicas en las que se analizaba el resultado obtenido en el periodo anterior y se validaba el avance hacia los objetivos del hito en curso. Esta metodología de trabajo permitió una eficaz distribución de tareas entre los miembros del equipo y una gestión optimizada del tiempo, contribuyendo significativamente al cumplimiento de los hitos del proyecto.

Las actas de estas reuniones se pueden encontrar en el [Anexo 3: "Actas de Reuniones"](#).

Por último, cada vez que un hito era completado, se organizaba una reunión con el tutor para presentar los avances, recibir *feedback* y, si fuera necesario, realizar ajustes al alcance previamente definido del proyecto.

## 4.2. Extracción de Requisitos

La fase inicial del desarrollo de EduWordle comenzó con un proceso de elicitación de requisitos fundamental para establecer las bases y el alcance del proyecto. Para ello, se llevaron a cabo diferentes métodos que permitieron dar una idea más completa del sistema y una mejor comprensión a la hora de la implementación.

### 4.2.1. Entrevista

El primer método que se llevó a cabo fue una entrevista directa con el tutor del proyecto, en la cual se definieron los objetivos principales y las fronteras iniciales del sistema. Los detalles de esta entrevista y los requisitos preliminares extraídos se encuentran en el [Anexo 1: "Entrevista para la Definición de Requisitos"](#).

Es importante señalar que, dada la naturaleza evolutiva del proyecto y los ajustes realizados posteriormente, ciertos requisitos inicialmente identificados en esta etapa fueron modificados, expandidos o, en algunos casos, suprimidos a lo largo del proceso. Esta adaptabilidad ha permitido que el producto final se haya alineado de manera más precisa con las necesidades emergentes y el enfoque conjunto del proyecto.

#### 4.2.2. Primer prototipo

Como parte de esta etapa, también se llevó a cabo el desarrollo de un prototipo visual detallado. Este prototipo fue concebido con el objetivo de mostrar el aspecto de una posible adaptación del sistema a dispositivos móviles, así como de visualizar el flujo de interacción del usuario.

A pesar de que este aspecto no se consolidó como un requisito final, la creación de este prototipo resultó ser fundamental y de gran valor. Sirvió como una base sólida para la definición posterior de la interfaz de usuario de la aplicación web, permitiendo refinar la estética general, la disposición de los elementos y los caminos de navegación antes de la implementación final. Esta aproximación facilitó la anticipación de desafíos de diseño y la toma de decisiones informadas.

En la siguiente figura se presenta el prototipo visual clave que ilustra estos conceptos iniciales de diseño:

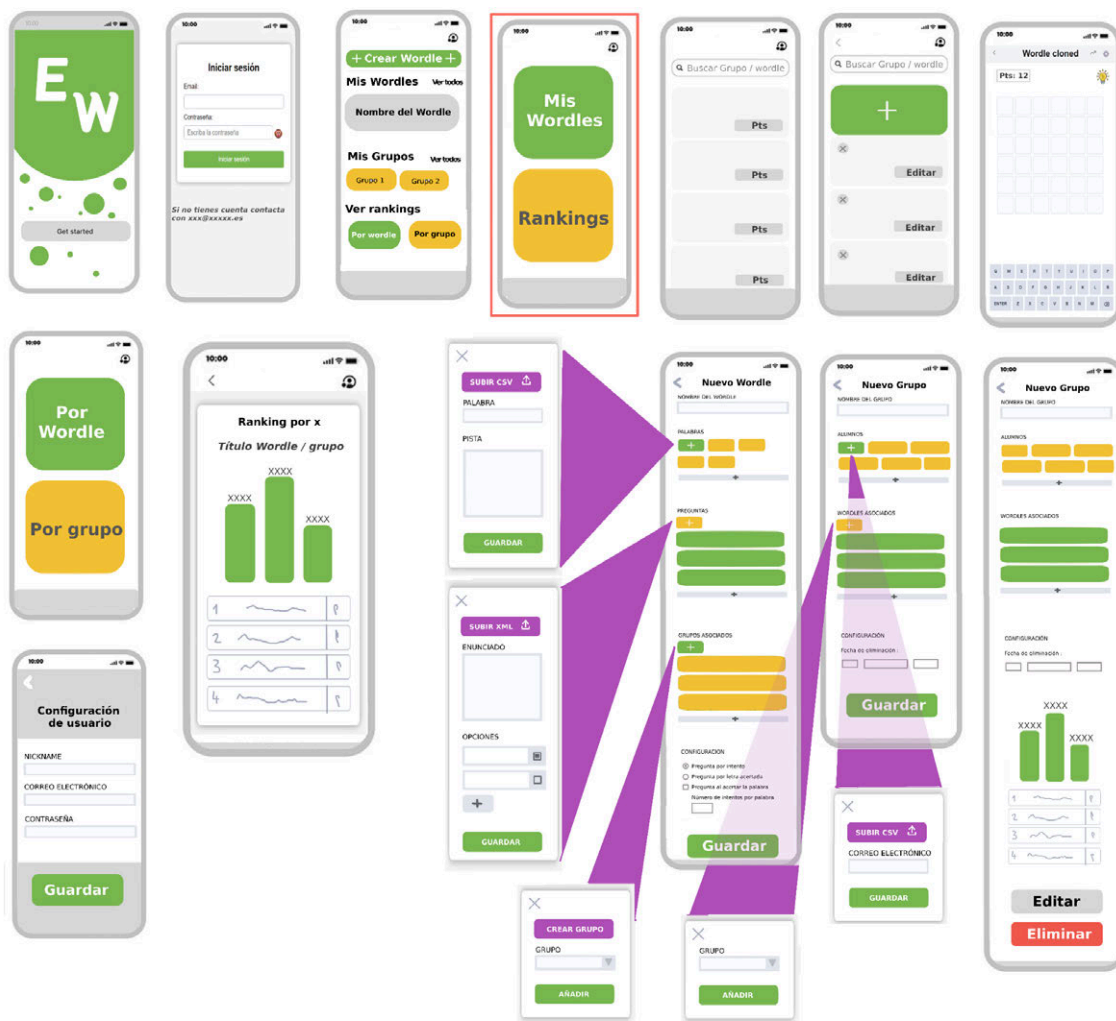


Figura 4.2: Prototipo inicial

### 4.2.3. Requisitos funcionales

#### Gestión de Usuarios

Nombre de la RF	Descripción	Acceptance Criteria
RF1: Iniciar sesión	Como <b>usuario</b> , quiero iniciar sesión con mi	<b>AC1.1: Dada</b> una dirección de correo electrónico válida y una contraseña correcta para un usuario registrado, <b>cuando</b> el usuario intente iniciar sesión, <b>entonces</b> el sistema guardará la información de sesión (incluyendo

	<p>correo electrónico y contraseña para acceder de forma segura a la plataforma y a las funcionalidades correspondientes a mi rol.</p>	<p>el <i>token</i> de autorización) de forma segura en el almacenamiento local del navegador y redirigirá al usuario al <i>dashboard</i> correspondiente a su rol</p> <p><b>AC1.2: Dada</b> una dirección de correo electrónico no válida o una contraseña incorrecta, <b>cuando</b> el usuario intente iniciar sesión, <b>entonces</b> el sistema mostrará un mensaje de error indicando “<i>Credenciales Incorrectas</i>”. No se muestran errores específicos de contraseña o correo inválidos para aumentar la seguridad</p>
<p><b>RF2:</b> Cerrar sesión</p>	<p>Como <b>usuario</b>, quiero poder cerrar sesión para proteger mi cuenta.</p>	<p><b>AC2.1: Cuando</b> el usuario pulse el botón de “<i>cerrar sesión</i>”, <b>entonces</b> el sistema eliminará toda la información sesión y el <i>token</i> de autorización del almacenamiento local del navegador</p> <p><b>AC2.2: Luego</b>, el usuario será redirigido automáticamente a la página de login</p> <p><b>AC2.3: Una vez redirigido a la página de inicio</b>, el usuario no podrá acceder a ninguna pantalla protegida (anterior a la sesión) utilizando las funciones de navegación del navegador</p>
<p><b>RF3:</b> Modificar contraseña</p>	<p>Como <b>usuario</b>, quiero poder modificar mi contraseña existente desde mi perfil para actualizar mis credenciales de acceso de forma segura.</p>	<p><b>AC3.1: Dado</b> un usuario autenticado, <b>cuando</b> pulse el botón de “<i>configuración</i>” desde el <i>dashboard</i>, <b>entonces</b> el sistema presentará un formulario para introducir la contraseña actual y una nueva contraseña. Las contraseñas nueva y antigua no pueden coincidir</p> <p><b>AC3.2: Cuando</b> el usuario introduzca la contraseña actual correctamente y una nueva contraseña que cumpla los criterios de seguridad y confirme la nueva contraseña, <b>entonces</b> el sistema actualizará la contraseña del usuario en la base de datos y mostrará un mensaje de éxito</p>

		<p><b>AC3.3:</b> Si la contraseña actual es incorrecta, <b>entonces</b> el sistema mostrará un mensaje de error específico indicando “La contraseña actual es incorrecta”</p>
		<p><b>AC3.4:</b> Si las nuevas contraseñas introducidas no coinciden o no cumplen los criterios de seguridad, <b>entonces</b> el sistema mostrará un mensaje de error adecuado</p>
		<p><b>AC3.5:</b> Dado que un alumno inicia sesión por primera vez (su contraseña coincide con la otorgada inicialmente por el sistema), <b>cuando</b> acceda a la plataforma, <b>entonces</b> el sistema lo obligará a cambiar su contraseña por una nueva y válida.</p>

Tabla 1: Requisitos de Gestión de Usuarios

## Gestión de Wordles

Nombre de la RF	Descripción	Acceptance Criteria
<b>RF4:</b> Visualizar Lista de Wordles accesibles (alumno)	Como <b>alumno</b> , quiero visualizar una lista de Wordles disponibles para mí para poder elegir y jugar cualquiera de ellos.	<p><b>AC4.1:</b> Dado un alumno autenticado desde el <i>dashboard</i>, <b>cuando</b> accede a “Mis Wordles”, <b>entonces</b> el sistema mostrará automáticamente una lista de Wordles a los que tiene acceso</p>
		<p><b>AC4.2:</b> Cuando el alumno seleccione un Wordle de esta lista, <b>entonces</b> el sistema iniciará una nueva partida de ese Wordle</p>

<b>RF5:</b> Visualizar Lista de <i>Wordles</i> propios (profesor)	Como <b>profesor</b> , quiero visualizar una lista de los <i>Wordles</i> que he creado para poder acceder a ellos, editarlos y eliminarlos.	<b>AC5.1: Dado</b> un profesor autenticado, <b>cuando</b> acceda a la sección “ <i>Mis Wordles</i> ” desde el <i>dashboard</i> , <b>entonces</b> el sistema mostrará una lista de todos los <i>Wordles</i> que el profesor ha creado
		<b>AC5.2: El profesor</b> podrá usar la barra de búsqueda para filtrar la lista de <i>Wordles</i> por nombre
		<b>AC5.3: Cuando</b> el profesor seleccione un <i>Wordle</i> de la lista, <b>entonces</b> el sistema le redirigirá a la pantalla de “ <i>Visualizar información de un Wordle</i> ” (RF6)
<b>RF6:</b> Visualizar información de un <i>Wordle</i>	Como <b>profesor</b> , quiero visualizar la información de un <i>Wordle</i> específico que he creado, incluyendo sus componentes y configuración, para poder gestionarlo.	<b>AC6.1: Dado</b> que el profesor ha seleccionado un <i>Wordle</i> de su lista (RF5), <b>cuando</b> acceda a la pantalla de visualización, <b>entonces</b> se mostrará una pestaña que contendrá: <ul style="list-style-type: none"> <li>- Nombre del <i>Wordle</i></li> <li>- Dificultad del <i>Wordle</i></li> <li>- Lista de Palabras asociadas</li> <li>- Lista de Preguntas asociadas</li> <li>- Lista de grupos que tienen acceso al <i>Wordle</i></li> <li>- Fecha de creación</li> </ul>
		<b>AC6.2: En esta pantalla</b> , el profesor tendrá visible un botón para “ <i>Editar Wordle</i> ” (RF8) que le permitirá acceder a su modificación
		<b>AC6.3: En esta pantalla</b> , el profesor tendrá visible un botón para “ <i>Eliminar Wordle</i> ” (RF9) que le permitirá borrar el <i>Wordle</i>
<b>RF7: Crear</b> <i>Wordle</i>	Como <b>profesor</b> , quiero poder crear nuevos <i>Wordles</i> configurando sus componentes y asignándole	<b>AC7.1: Cuando</b> el profesor inicie el proceso de creación de un <i>Wordle</i> , <b>entonces</b> el sistema le presentará un formulario que obligará a ingresar: <ul style="list-style-type: none"> <li>- Un nombre único para el <i>Wordle</i></li> <li>- Al menos una pregunta</li> <li>- Al menos una palabra.</li> </ul>
		<b>AC7.2: El profesor</b> podrá añadir nuevas palabras al <i>Wordle</i> pulsando el botón “+” de la sección de Palabras. <b>Entonces</b> se abrirá un <i>popup</i> con un campo

un nombre, para que mis alumnos puedan jugar.	de texto para palabra y otro para pista. Este segundo es opcional rellenarlo.
	<b>AC7.3:</b> Si al añadir una palabra se cumplen los requisitos, <b>entonces</b> se mostrará un mensaje de éxito
	<b>AC7.4:</b> Si al añadir una palabra no se cumplen los requisitos o ya existe en el <i>Wordle</i> , <b>entonces</b> se mostrará un mensaje de error y no dejará añadir la palabra
	<b>AC7.5:</b> El <b>profesor</b> podrá eliminar palabras que previamente asociadas al <i>Wordle</i> durante el proceso de creación
	<b>AC7.6:</b> El <b>profesor</b> podrá añadir nuevas preguntas al <i>Wordle</i> pulsando el botón “+” en la sección de Preguntas. <b>Entonces</b> se abrirá un <i>popup</i> en el que será obligatorio introducir: <ul style="list-style-type: none"> <li>- Enunciado</li> <li>- Al menos dos opciones</li> <li>- Al menos una opción correcta</li> </ul>
	<b>AC7.7:</b> Si al añadir una pregunta se cumplen los requisitos, <b>entonces</b> se mostrará un mensaje de éxito
	<b>AC7.8:</b> Si al añadir una pregunta no se cumplen los requisitos o ya existe en el <i>Wordle</i> , <b>entonces</b> se mostrará un mensaje de error y no se añadirá
	<b>AC7.9:</b> El <b>profesor</b> podrá eliminar preguntas que haya asociado al <i>Wordle</i> previamente durante el proceso de creación
	<b>AC7.10:</b> El <b>profesor</b> podrá vincular grupos existentes al <i>Wordle</i> pulsando el botón “+” en la sección de <i>Wordles</i> . <b>Entonces</b> se abrirá un <i>popup</i> en el que podrá seleccionar uno de los grupos que él ha creado.
	<b>AC7.11:</b> Si la vinculación de grupo ha sido exitosa,

		<p><b>entonces</b> el sistema mostrará un mensaje de éxito</p>
		<p><b>AC7.12: El profesor</b> podrá desvincular Grupos que haya asociado al <i>Wordle</i> previamente durante el proceso de creación</p>
		<p><b>AC7.13: Mientras</b> el profesor añade palabras, <b>entonces</b> el sistema visualizará en tiempo real la lista de palabras y preguntas asociadas al <i>Wordle</i></p>
		<p><b>AC7.14: Cuando</b> el profesor complete los campos obligatorios y guarde el <i>Wordle</i>, <b>entonces</b> el sistema validará los datos y mostrará un mensaje de éxito indicando “<i>Wordle creado correctamente</i>”</p>
		<p><b>AC7.15: Luego</b> el profesor será redirigido al <i>dashboard</i></p>
<p><b>RF8: Editar <i>Wordle</i></b></p>	<p>Como <b>profesor</b>, quiero poder editar los <i>Wordles</i> que he creado para modificar las preguntas, las palabras, la configuración y los grupos asociados.</p>	<p><b>AC8.1: Dado</b> que el profesor ha accedido a la pantalla de edición de un <i>Wordle</i> (desde RF6), <b>cuando</b> se cargue el formulario de edición, <b>entonces</b> mostrará los datos actuales del <i>Wordle</i></p>
		<p><b>AC8.2: El profesor</b> podrá modificar el nombre del <i>Wordle</i> en el formulario por cualquier cosa que no sean caracteres vacíos (que no sea null)</p>
		<p><b>AC8.3: El profesor</b> podrá añadir nuevas palabras al <i>Wordle</i> pulsando el botón “+” de la sección de Palabras. <b>Entonces</b> se abrirá un <i>popup</i> con un campo de texto para palabra y otro para pista. Este segundo es opcional rellenarlo.</p>
		<p><b>AC8.4: Si</b> al añadir una palabra se cumplen los requisitos, <b>entonces</b> se mostrará un mensaje de éxito</p>
		<p><b>AC8.5: Si</b> al añadir una palabra no se cumplen los requisitos o ya existe en el <i>Wordle</i>, <b>entonces</b> se mostrará un mensaje de error y no se añadirá</p>
		<p><b>AC8.6: El profesor</b> podrá eliminar palabras ya añadidas</p>
		<p><b>AC8.7: El profesor</b> podrá añadir nuevas preguntas al</p>

		<p><i>Wordle</i> pulsando el botón “+” en la sección de Preguntas. <b>Entonces</b> se abrirá un <i>popup</i> en el que será obligatorio introducir:</p> <ul style="list-style-type: none"> <li>- Enunciado</li> <li>- Al menos dos opciones</li> <li>- Al menos una opción correcta</li> </ul>
		<p><b>AC8.8:</b> Si al añadir una pregunta se cumplen los requisitos, <b>entonces</b> se mostrará un mensaje de éxito</p>
		<p><b>AC8.9:</b> Si al añadir una pregunta no se cumplen los requisitos o ya existe en el <i>Wordle</i>, <b>entonces</b> se mostrará un mensaje de error y no se añadirá</p>
		<p><b>AC8.10:</b> El <b>profesor</b> podrá eliminar preguntas ya añadidas</p>
		<p><b>AC8.11:</b> El <b>profesor</b> podrá vincular grupos existentes al <i>Wordle</i> pulsando el botón “+” en la sección de <i>Wordles</i>. <b>Entonces</b> se abrirá un <i>popup</i> en el que podrá seleccionar uno de los grupos que él ha creado.</p>
		<p><b>AC8.12:</b> Si la vinculación de grupo ha sido exitosa, <b>entonces</b> el sistema mostrará un mensaje de éxito</p>
		<p><b>AC8.13:</b> El <b>profesor</b> podrá desvincular Grupos que haya asociado al <i>Wordle</i> previamente</p>
		<p><b>AC8.14:</b> Mientras el profesor añade palabras, <b>entonces</b> el sistema visualizará en tiempo real la lista de palabras y preguntas asociadas al <i>Wordle</i></p>
		<p><b>AC8.15:</b> Cuando el profesor intente guardar los cambios, <b>entonces</b> el sistema validará que el <i>Wordle</i> contenga como mínimo una pregunta y una palabra asociadas. Si no se cumple, se mostrará un mensaje de error específico</p>
		<p><b>AC8.16:</b> Si los cambios son válidos, <b>cuando</b> el profesor guarde, <b>entonces</b> el sistema mostrará un mensaje de</p>

		<p>éxito indicando “Cambios guardados correctamente”</p> <p><b>AC8.17:</b> Luego, el profesor será redirigido al <i>dashboard</i></p>
<p><b>RF9:</b> Eliminar <i>Wordle</i></p>	<p>Como <b>profesor</b>, quiero poder eliminar los <i>Wordles</i> que he creado para administrar mis <i>Wordles</i>.</p>	<p><b>AC9.1:</b> Dado que el profesor está visualizando la información de un <i>Wordle</i> (RF6), <b>cuando</b> pulse el botón “Eliminar”, <b>entonces</b> el sistema solicitará una confirmación</p>
		<p><b>AC9.2:</b> Si el profesor confirma la eliminación, <b>entonces</b> el sistema eliminará el <i>Wordle</i> y todos sus elementos asociados de la base de datos</p>
		<p><b>AC9.3:</b> Luego, el sistema mostrará un mensaje de éxito</p>
		<p><b>AC9.4:</b> Finalmente, el profesor será redirigido al <i>dashboard</i></p>
<p><b>RF10:</b> Jugar <i>Wordle</i></p>	<p>Como <b>alumno</b> quiero poder jugar a los <i>wordles</i> que me han asignado</p>	<p><b>AC10.1:</b> Dado que el alumno está visualizando la lista de <i>Wordles</i> (RF4), <b>cuando</b> pulse sobre uno de ellos, comenzará la partida.</p>
		<p><b>AC10.2:</b> El alumno debe poder introducir un intento de palabra, con un máximo de 6 intentos.</p>
		<p><b>AC10.3:</b> En cada intento, dependiendo del nivel de aprendizaje se mostrarán una o más preguntas en cada intento, como alumno debo poder responder a ellas</p>
		<p><b>AC10.4:</b> Luego, después de 6 intentos, si el <i>wordle</i> contiene más palabras, aparecerá una nueva, en caso contrario mostrará un mensaje de fin de juego y mostrará un botón para redirigir al <i>dashboard</i></p>
		<p><b>AC10.5</b> Finalmente, el alumno será redirigido al <i>dashboard</i></p>

Tabla 2: Requisitos de Gestión de *Wordles*

## Gestión de Grupos

Nombre de la RF	Descripción	Acceptance Criteria
<p><b>RF11:</b> Visualizar información de un Grupo (alumno)</p>	<p>Como <b>alumno</b>, quiero visualizar la información de un grupo al que pertenezco, incluyendo los <i>Wordles</i> a los que tiene acceso y los compañeros que lo componen.</p>	<p><b>AC11.1: Dado</b> que el alumno ha seleccionado un grupo de la lista de grupos a los que pertenece (RF13), <b>cuando</b> acceda a la pantalla de visualización del grupo, <b>entonces</b> se mostrará una pantalla que contendrá:</p> <ul style="list-style-type: none"> <li>- Nombre de grupo</li> <li>- Lista de alumnos asociados</li> <li>- Lista de <i>Wordles</i> a los que tiene acceso el grupo</li> <li>- Fecha de inicio</li> <li>- Fecha de fin</li> </ul>
<p><b>RF12:</b> Visualizar información de un Grupo (profesor)</p>	<p>Como <b>profesor</b>, quiero visualizar la información de un grupo específico de los que he creado para administrar los alumnos que lo componen y los <i>Wordles</i> a los que tiene acceso</p>	<p><b>AC12.1: Dado</b> que el profesor ha seleccionado un grupo de la lista de grupo que ha creado (RF14), <b>cuando</b> acceda a la pantalla de visualización del grupo, <b>entonces</b> se mostrará una pestaña que contendrá:</p> <ul style="list-style-type: none"> <li>- Nombre de grupo</li> <li>- Lista de alumnos que componen el grupo</li> <li>- Lista de <i>Wordles</i> a los que tiene acceso el grupo</li> <li>- Fecha de inicio</li> <li>- Fecha de fin</li> </ul> <p><b>AC12.2: En esta pantalla</b>, el profesor tendrá visible un botón para “<i>Editar Grupo</i>” (RF16) que le permitirá acceder a su modificación</p> <p><b>AC12.3: En esta pantalla</b>, el profesor tendrá visible un botón para “<i>Eliminar Grupo</i>” (RF17) que le permitirá borrar el <i>Grupo</i></p>
<p><b>RF13:</b> Visualizar Lista de</p>	<p>Como <b>alumno</b>, quiero visualizar todos los grupos a los</p>	<p><b>AC13.1: Dado</b> un alumno autenticado que accede a la sección de “<i>Mis Grupos</i>” desde el <i>dashboard</i>, <b>entonces</b> el sistema mostrará una lista de todos los grupos a los que pertenece el alumno</p>

Grupos accesibles (alumno)	que pertenezco para poder acceder a su información.	<b>AC13.2:</b> El <b>alumno</b> podrá utilizar una barra de búsqueda para filtrar la lista de grupos por nombre
		<b>AC13.2:</b> Cuando el alumno seleccione un grupo de la lista, <b>entonces</b> el sistema le redirigirá a la pantalla de “Visualizar Información de un Grupo” (RF12)
<b>RF14:</b> Visualizar Lista de Grupos propios (profesor)	Como <b>profesor</b> , quiero poder visualizar todos los grupos que he creado para poder acceder a ellos.	<b>AC14.1:</b> Dado un profesor autenticado que accede a la sección de “Mis Grupos” desde el <i>dashboard</i> , <b>entonces</b> el sistema mostrará una lista de todos los grupos activos que el profesor ha creado
		<b>AC14.2:</b> El <b>profesor</b> podrá utilizar una barra de búsqueda para filtrar la lista de grupos por nombre
		<b>AC14.3:</b> Cuando el profesor seleccione un grupo de la lista, <b>entonces</b> el sistema le redirigirá a la pantalla de “Visualizar Información de un Grupo” (RF12)
<b>RF15:</b> Crear Grupo	Como <b>profesor</b> , quiero poder crear nuevos grupos, asignarles un nombre y asociar alumnos y <i>Wordles</i> para organizar mis clases y actividades.	<b>AC15.1:</b> Cuando el profesor inicie el proceso de creación de un Grupo, <b>entonces</b> el sistema le presentará un formulario que obligará a ingresar: <ul style="list-style-type: none"> <li>- Un nombre único para el Grupo</li> <li>- Una fecha de inicio (por defecto se pone la fecha de creación)</li> </ul>
		<b>AC15.2:</b> El <b>profesor</b> podrá añadir nuevos alumnos al Grupo pulsando el botón “+” de la sección de Alumnos. <b>Entonces</b> se abrirá un <i>popup</i> con un campo de texto para el correo electrónico del alumno.
		<b>AC15.3:</b> Si al añadir un alumno se introduce un correo válido <b>entonces</b> se guardará y se mostrará un mensaje de éxito
		<b>AC15.4:</b> Si al añadir un alumno, este ya está vinculado al grupo, <b>entonces</b> se mostrará un mensaje de error y no se añadirá
		<b>AC15.5:</b> El <b>profesor</b> podrá desvincular alumnos previamente asociados al grupo durante el proceso de creación
		<b>AC15.6:</b> El <b>profesor</b> podrá asociar <i>Wordles</i> existentes

		<p>al grupo pulsando el botón de “+” en la sección de <i>Wordles</i>.</p> <p>Para asociarlo, aparecerá un <i>popup</i> con un selector de la lista de <i>Wordles</i> que ha creado el profesor</p> <p><b>AC15.7:</b> Si al añadir un <i>Wordle</i>, éste ya está vinculado previamente al Grupo, aparecerá un mensaje de advertencia y no permitirá vincular el <i>Wordle</i></p> <p><b>AC15.8:</b> El profesor podrá desvincular <i>Wordles</i> que ya ha asociado previamente al grupo</p> <p><b>AC15.9:</b> Mientras el profesor añade palabras, entonces el sistema visualizará en tiempo real la lista de alumnos y <i>Wordles</i> asociados al Grupo</p> <p><b>AC15.10:</b> Cuando el profesor guarde el nuevo grupo, entonces el sistema mostrará un mensaje de éxito indicando "<i>Grupo creado correctamente</i>"</p> <p><b>AC15.11:</b> Luego, el profesor será redirigido al <i>dashboard</i></p>
<p><b>RF16:</b>          Editar          Grupo</p>	<p>Como profesor, quiero poder editar un grupo para poder gestionar los alumnos que lo componen y los <i>Wordles</i> a los que tiene acceso.</p>	<p><b>AC16.1:</b> Dado que el profesor ha accedido a la pantalla de edición de un Grupo (desde RF12), cuando se cargue el formulario de edición, entonces mostrará los datos actuales del Grupo</p> <p><b>AC16.2:</b> El profesor podrá modificar el nombre del Grupo en el formulario por cualquier cosa que no sean caracteres vacíos (que no sea null)</p> <p><b>AC16.3:</b> El profesor podrá añadir nuevos alumnos al Grupo pulsando el botón “+” de la sección de Alumnos. Entonces se abrirá un <i>popup</i> con un campo de texto para el correo electrónico del alumno</p> <p><b>AC16.4:</b> Si al añadir una alumno se introduce un correo válido entonces se guardará y se mostrará un mensaje de éxito</p> <p><b>AC16.5:</b> Si al añadir un alumno, este ya está vinculado al grupo, entonces se mostrará un mensaje de error y no se añadirá</p>

		<p><b>AC16.6:</b> El <b>profesor</b> podrá desvincular alumnos del grupo</p>
		<p><b>AC16.7:</b> <b>Mientras</b> el profesor añade palabras, <b>entonces</b> el sistema visualizará en tiempo real la lista de alumnos y <i>Wordles</i> asociados al Grupo</p>
		<p><b>AC16.8:</b> El <b>profesor</b> podrá asociar <i>Wordles</i> existentes al grupo pulsando el botón de “+” en la sección de <i>Wordles</i>. Para asociarlo, aparecerá un <i>popup</i> con un selector de la lista de <i>Wordles</i> que ha creado el profesor</p>
		<p><b>AC16.9:</b> Si al añadir un <i>Wordle</i>, éste ya está vinculado previamente al Grupo, aparecerá un mensaje de advertencia y no permitirá vincular el <i>Wordle</i></p>
		<p><b>AC16.10:</b> El <b>profesor</b> podrá desvincular <i>Wordles</i> que ya ha asociado previamente al grupo</p>
		<p><b>AC16.11:</b> <b>Cuando</b> el profesor guarde el nuevo grupo, <b>entonces</b> el sistema mostrará un mensaje de éxito indicando "<i>Grupo actualizado correctamente</i>"</p>
		<p><b>AC16.12:</b> <b>Luego</b>, el profesor será redirigido al <i>dashboard</i></p>
<p><b>RF17:</b> Eliminar Grupo</p>	<p>Como <b>profesor</b>, quiero poder eliminar un grupo para administrar mis grupos</p>	<p><b>AC17.1:</b> <b>Dado</b> que el profesor está visualizando la información de un Grupo (RF12), <b>cuando</b> pulse el botón “<i>Eliminar</i>”, <b>entonces</b> el sistema solicitará una confirmación</p>
		<p><b>AC17.2:</b> Si el profesor confirma la eliminación, <b>entonces</b> el sistema eliminará el <i>Wordle</i> y todos sus elementos asociados de la base de datos</p>
		<p><b>AC17.3:</b> <b>Luego</b>, el sistema mostrará un mensaje de éxito</p>
		<p><b>AC17.4:</b> <b>Finalmente</b>, el profesor será redirigido al <i>dashboard</i></p>

Tabla 3: Requisitos de Gestión de Grupos

## Gestión de Rankings

Nombre de la RF	Descripción	Acceptance Criteria
<b>RF18:</b> Ver Ranking de Wordle	Como <b>profesor</b> quiero poder ver las puntuaciones de mis alumnos obtenidas para un wordle para poder calificar su desempeño	<b>AC18.1: Dado</b> que el profesor está en la pestaña de "Clasificaciones", <b>cuando</b> seleccione la opción de "Por Wordle", <b>entonces</b> el sistema le mostrará una tabla por cada Wordle que haya creado y en cada tabla, una entrada de cada alumno que ha jugado ese wordle con su mejor puntuación. La lista estará ordenada de mayor a menor puntuación
<b>RF19:</b> Ver Ranking de Grupo	Como <b>profesor</b> quiero poder ver las puntuaciones de mis alumnos obtenidas para los wordles asociados a su grupo para poder calificar su desempeño	<b>AC19.1: Dado</b> que el profesor está en la pestaña de "Clasificaciones", <b>cuando</b> seleccione la opción de "Por Grupo", <b>entonces</b> el sistema le mostrará una tabla por cada Grupo que haya creado el profesor y en cada tabla, una entrada de cada alumno del grupo con la suma de las puntuaciones obtenidas en los wordles asociados al grupo. La lista estará ordenada de mayor a menor puntuación

Tabla 4: Requisitos de Gestión de Rankings

## Gestión de Archivos

Nombre de la RF	Descripción	Acceptance Criteria
<b>RF20:</b> Subir CSV de Alumnos, Preguntas y Palabras	Como <b>profesor</b> quiero subir un archivo CSV para para importar	<b>AC20.1: Dado</b> que el profesor está en el contexto de creación/edición de un grupo o <i>Wordle</i> , <b>cuando</b> seleccione la opción de "Importar desde CSV", <b>entonces</b> el sistema le permitirá seleccionar un archivo CSV desde su sistema local

<p>masivamente una lista de alumnos a un grupo o una lista de palabras/preguntas a un <i>Wordle</i>, simplificando el proceso de creación de contenido</p>	<p><b>AC20.2: Cuando</b> el profesor intente subir un archivo, <b>entonces</b> el sistema verificará el formato y la estructura del archivo CSV en base a las especificaciones predefinidas</p>
	<p><b>AC20.3: Si</b> el archivo no es compatible o tiene un formato incorrecto, <b>entonces</b> el sistema mostrará un mensaje de error detallado indicando la razón</p>
	<p><b>AC20.4: Si</b> el archivo es compatible y cumple con el formato, <b>entonces</b> el sistema lo procesará y añadirá los elementos al elemento a editar/ crear en local. Entonces mostrará un mensaje de éxito</p>
	<p><b>AC20.5: Después de la importación,</b> la lista de alumnos/palabras/preguntas en el formulario de creación/edición se actualizará para reflejar los nuevos elementos.</p>

*Tabla 5: Requisitos de Gestión de Archivos*

#### 4.2.4. Explicación del funcionamiento del juego

La mecánica de juego de EduWordle es inspiración del juego original "Wordle", pero adaptada y extendida para un contexto educativo. El objetivo principal es que los alumnos adivinen una palabra oculta en un número limitado de intentos. En cada intento, el jugador introduce una palabra del tamaño de la palabra a adivinar y por cada letra, se comprueba si está en la palabra original y si está en la posición correcta. Para una letra en la correcta posición, la letra se torna verde y para una letra correcta pero en posición incorrecta, la letra se torna amarilla. En el modo daltonismo los colores son diferentes. En el siguiente diagrama se muestra la funcionalidad:

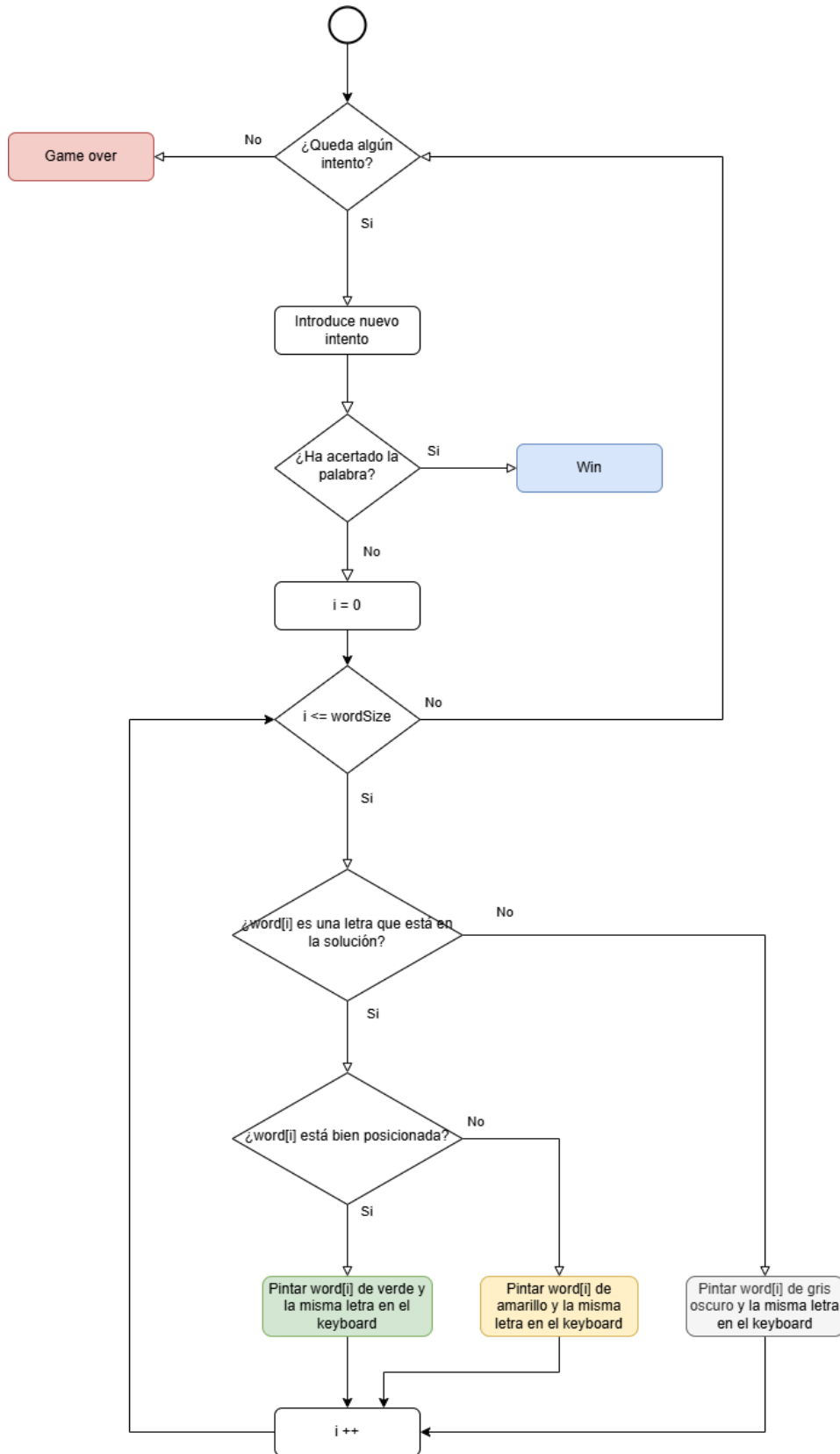


Figura 4.3: Diagrama de actividad del funcionamiento del juego original

La capa educativa consiste en un sistema de preguntas que pueden ser configuradas para aparecer al introducir una palabra o por cada letra que se acierte según dictamine el profesor en la configuración del *Wordle*. Los aciertos frente a estas preguntas determinan si la palabra se aceptará como entrada o si simplemente se perderá ese intento. Cuando la dificultad es alta, los aciertos determinarán las letras que se revelarán como amarillas o verdes.

Para mejorar el entendimiento de esta funcionalidad, se presenta la siguiente figura:

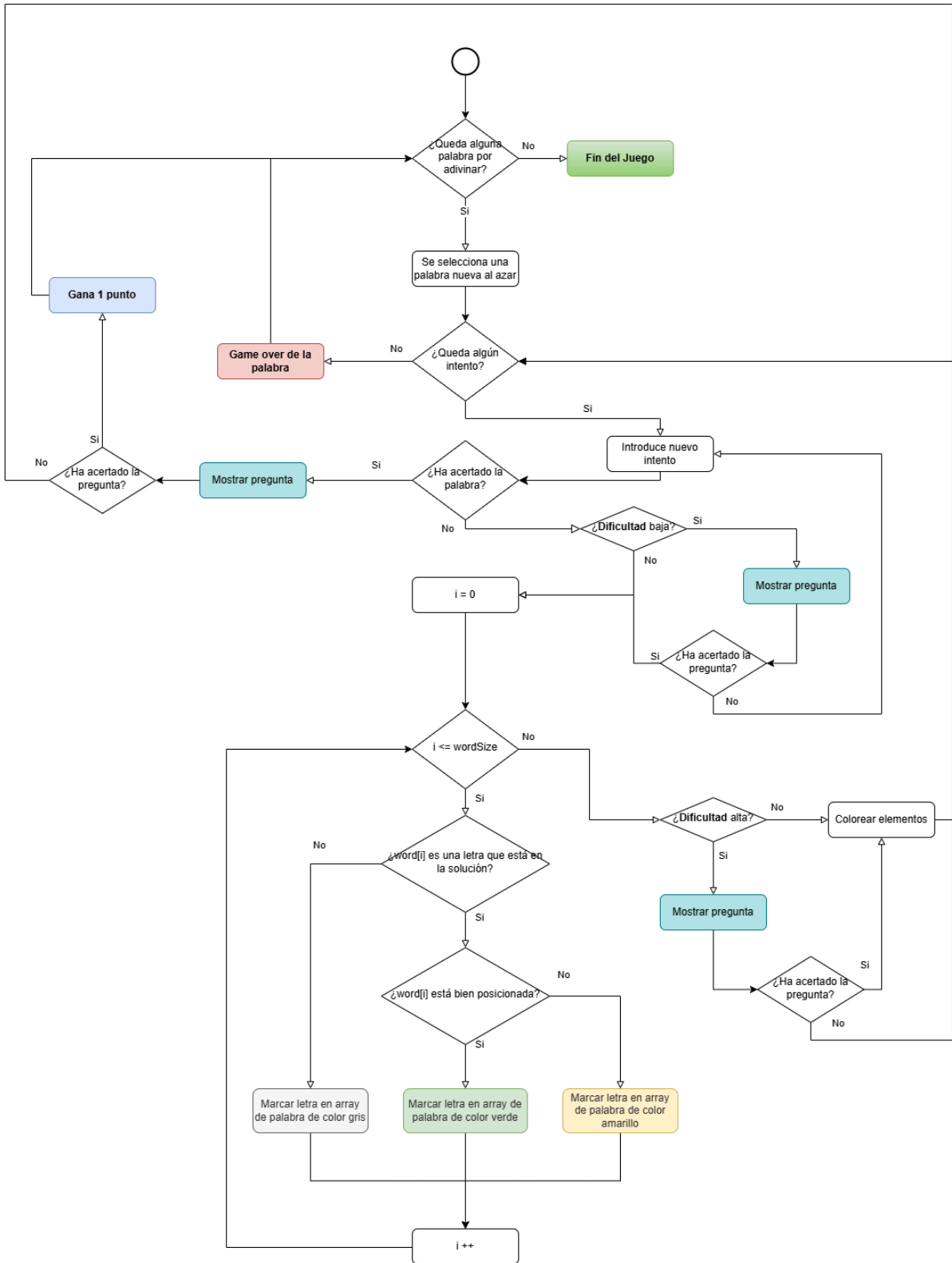


Figura 4.4: Diagrama de actividad del juego de EduWordle con capa educativa

#### 4.2.5. Diagramas de Casos de Uso

Para una mejor comprensión del sistema y mayor facilidad en la validación de requisitos, se generaron diagramas de casos de uso que modelan y representan las funcionalidades del sistema desde la perspectiva de los usuarios que interactúan con él. Estos casos de uso están numerados de tal forma que la numeración corresponde con los requisitos funcionales del apartado anterior y para mejor comprensión, en cada apartado se aporta una tabla de trazabilidad que relaciona ambos elementos.

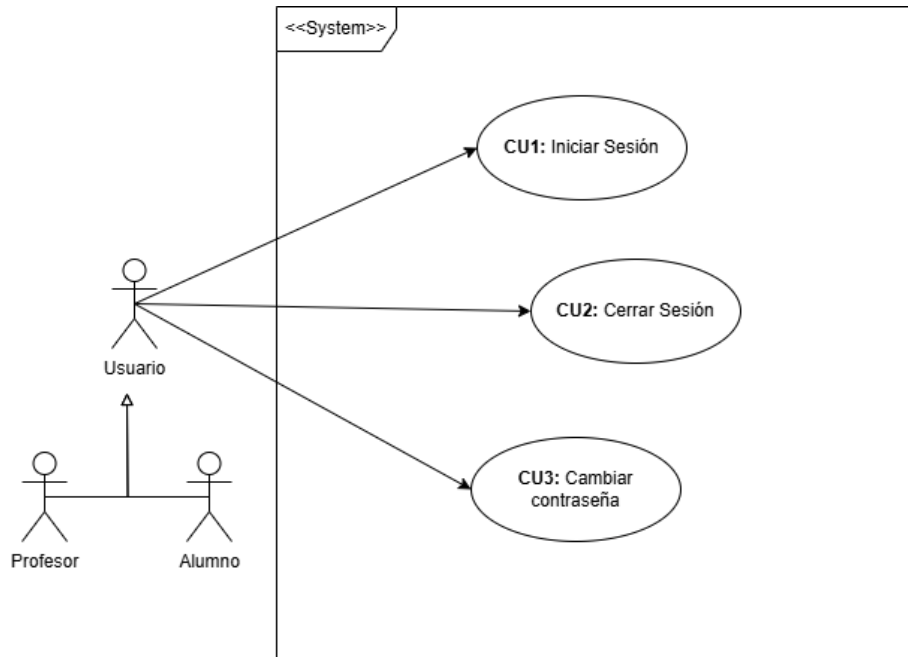
En esta sección se presentan diagramas que agrupan los casos de uso por funcionalidades.

## Gestión de Usuarios

En este diagrama se establecen los casos de uso relacionados con el acceso seguro a la aplicación y con la personalización de los datos de perfil. Esas funcionalidades son comunes entre profesores y estudiantes.

Requisito Funcional	Casos de uso relacionados
RF1: Iniciar Sesión	CU1: Iniciar Sesión
RF2: Cerrar Sesión	CU2: Cerrar Sesión
RF3: Modificar contraseña	CU3: Cambiar contraseña

*Tabla 6: Tabla de trazabilidad entre Requisitos Funcionales y Casos de Uso Gestión Usuarios*



**Figura 4.5:** Diagrama caso de uso Gestión de Usuarios

## Gestión de Wordles

Este diagrama describe la creación, gestión y juego de *Wordles* educativos por ello se detallan todos los requisitos específicos al ciclo de vida de los mismos dentro de la plataforma. Estas funcionalidades son fundamentales ya que permiten la personalización de los “juegos” que conforman la naturaleza educativa del sistema.

Requisito Funcional	Casos de uso relacionados
RF4: Visualizar Lista de <i>Wordles</i> accesibles (alumno)	CU4: Visualizar lista wordles accesibles CU10: Jugar wordle
RF5: Visualizar Lista de <i>Wordles</i> propios (profesor)	CU5: Visualizar Lista de <i>Wordles</i> propios

RF6: Visualizar información de un <i>Wordle</i>	CU6: Visualizar información de un <i>Wordle</i> CU8: <i>Editar wordle</i> CU9: <i>Eliminar wordle</i>
RF7: Crear <i>Wordle</i>	CU7: Crear <i>Wordle</i> CU23: Añadir pregunta CU21: Subir preguntas CSV CU24: Eliminar pregunta CU25: Añadir palabra CU22: Subir palabras CSV CU26: Eliminar palabra CU27: Asociar grupo CU15: Ver lista grupos propios
RF8: Editar <i>Wordle</i>	CU8: Editar <i>Wordle</i> CU23: Añadir pregunta CU21: Subir preguntas CSV CU24: Eliminar pregunta CU25: Añadir palabra CU22: Subir palabras CSV CU26: Eliminar palabra CU27: Asociar grupo CU15: Ver lista grupos propios
RF9: Eliminar <i>Wordle</i>	CU9: Eliminar <i>Wordle</i>
RF10: Jugar <i>Wordle</i>	CU10: Jugar <i>wordle</i>
RF18: Ver Ranking de <i>Wordle</i>	CU18: Ver Ranking de <i>Wordle</i>

*Tabla 7: Tabla de trazabilidad entre Requisitos Funcionales y Casos de Uso Gestión Wordles*

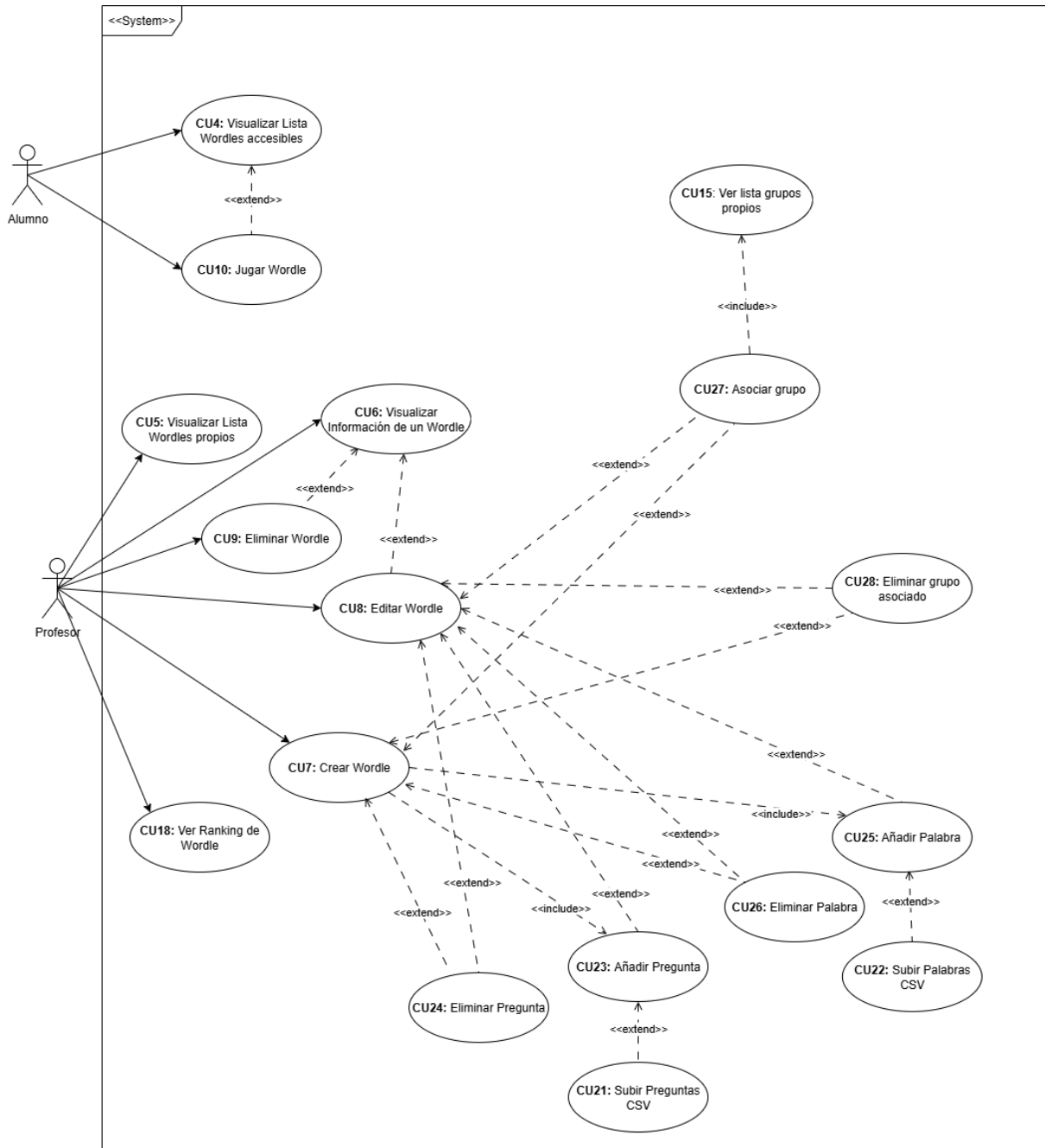


Figura 4.6: Diagrama caso de uso Gestión de Wordles

## Gestión de Grupos

Para facilitar la organización de estudiantes y la asignación de *Wordles*, EduWordle incorpora un sistema de gestión de grupos. Este abarca la creación, modificación y eliminación de los mismos, así como la asignación de alumnos y la asociación de *Wordles* para permitir su acceso. Esto último es crucial para controlar la

disponibilidad de los “juegos” y para vincular el desempeño de los alumnos de forma organizada.

Requisito Funcional	Casos de uso relacionados
RF11: Visualizar información de un Grupo (alumno)	CU11: Visualizar información de un Grupo (alumno)
RF12: Visualizar información de un Grupo (profesor)	CU12: Visualizar información de un Grupo (profesor) CU17: Eliminar Grupo CU16: Editar Grupo
RF13: Visualizar Lista de Grupos accesibles (alumno)	CU13: Visualizar Lista de Grupos accesibles
RF14: Visualizar Lista de Grupos propios (profesor)	CU14: Visualizar Lista de Grupos propios
RF15: Crear Grupo	CU15: Crear Grupo CU29: Añadir alumno CU20: Subir CSV alumnos CU30: Eliminar alumno CU31: Asociar wordle CU5: Visualizar Lista de Wordles propios CU32: Eliminar Wordle asociado
RF16: Editar Grupo	CU17: Eliminar Grupo CU29: Añadir alumno CU20: Subir CSV alumnos CU30: Eliminar alumno CU31: Asociar wordle CU5: Visualizar Lista de Wordles propios

	CU32: Eliminar Wordle asociado
RF17: Eliminar Grupo	CU17: Eliminar Grupo
RF19: Ver Ranking de Grupo	CU19: Ver Ranking de Wordle

**Tabla 8:** Tabla de trazabilidad entre Requisitos Funcionales y Casos de Uso Gestión Grupos

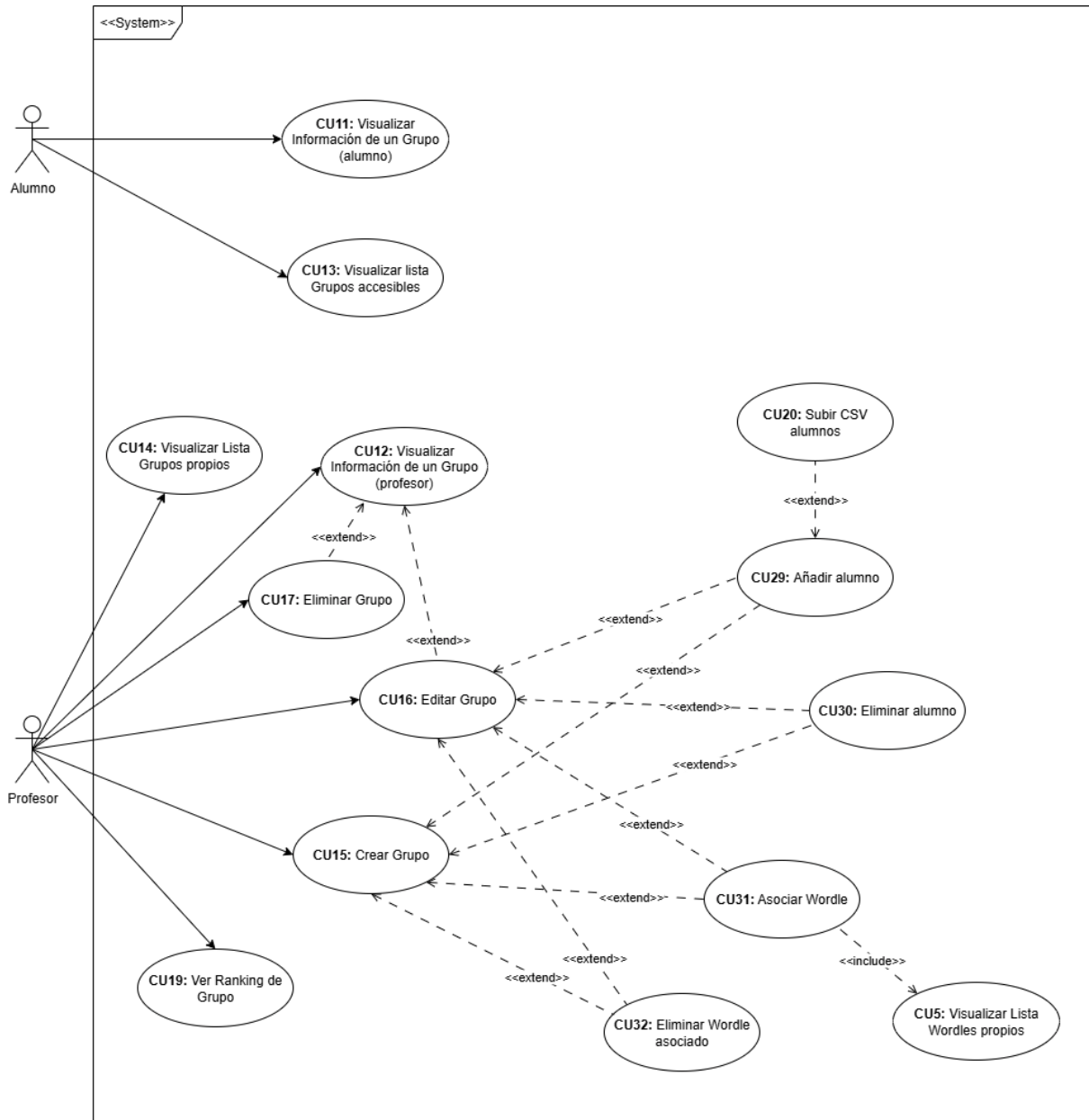


Figura 4.7: Diagrama caso de uso Gestión de grupos

### 4.3. Análisis

Una vez completada la fase de elicitación, se procedió con el análisis de requisitos identificados para el sistema EduWordle. Esta etapa fue crucial para interpretar, organizar y validar las necesidades de los usuarios y las funcionalidades deseadas, transformándolas en especificaciones detalladas y consistentes.

Este apartado pretende dar una comprensión más sencilla y visual del sistema a través de diversos diagramas.

### 4.3.1. Modelo de Datos

En esta etapa se empleó un diagrama de Entidad-Relación como una herramienta fundamental para representar gráficamente los componentes de datos esenciales de EduWordle y sus interconexiones. Este apartado detalla el modelo desarrollado, que ilustra las entidades, sus atributos y las relaciones que existen entre ellas, reflejando la complejidad de los datos del proyecto y sirviendo de guía para la posterior implementación física de la base de datos.

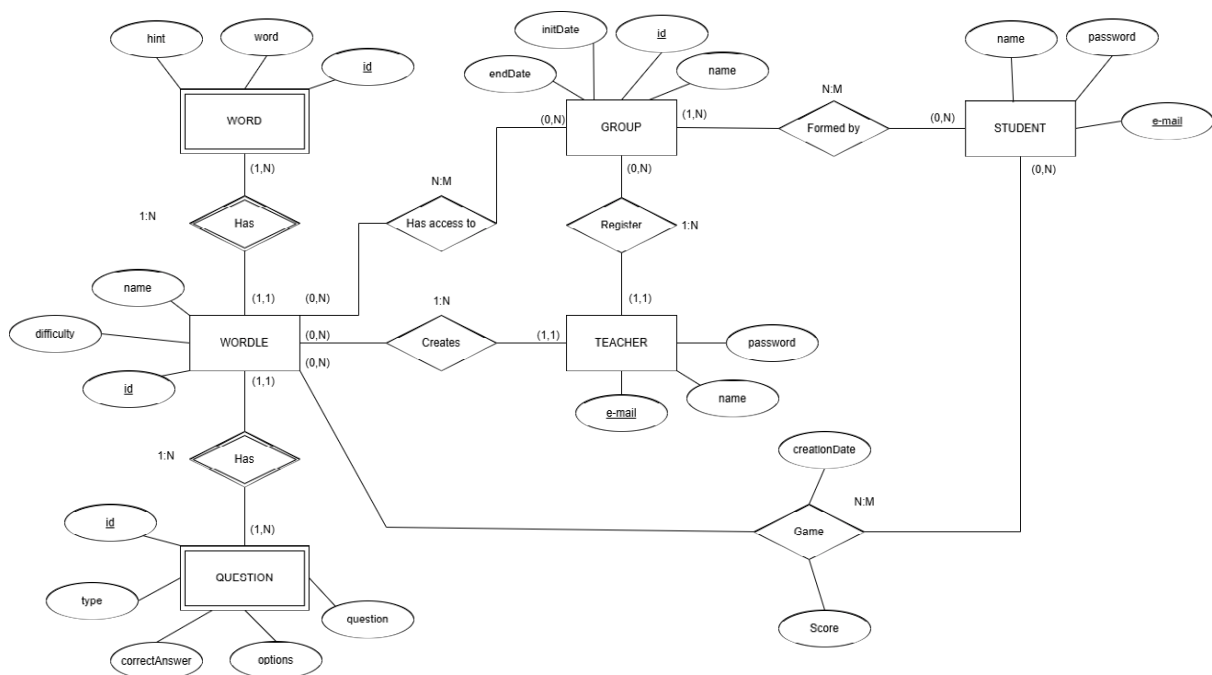


Figura 4.8: Modelo Entidad-Relación

Además, se realizó un diagrama para el paso a tablas que justifica las tablas de soporte derivadas de las relaciones  $N:M$ . Este diagrama se detalla en la siguiente figura:



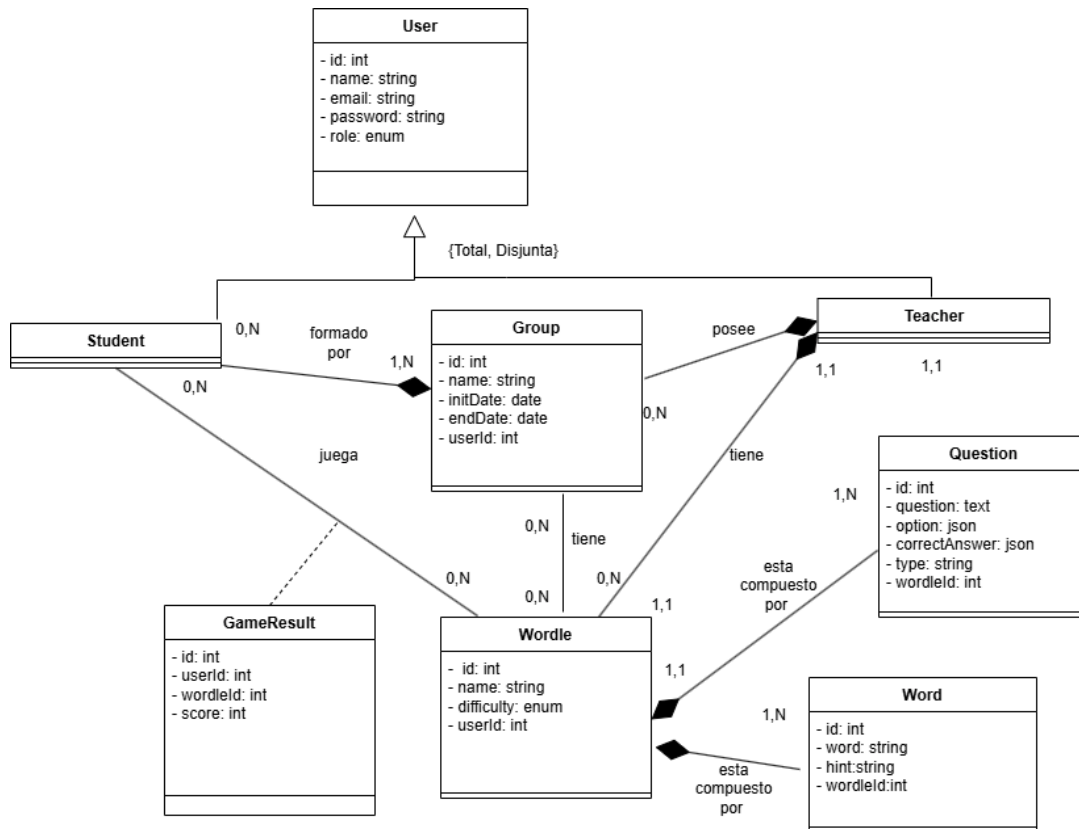


Figura 4.10: Diagrama de clases

### 4.3.3. Diagramas de secuencia

En este apartado se detalla a través de diagramas de secuencia cómo funciona la autenticación y envío de *token JWT* y cualquier petición que se produzca desde el *frontend*. Además, se ha considerado añadir un diagrama que muestre el funcionamiento del juego y el guardado de resultados.

#### Petición de Login:

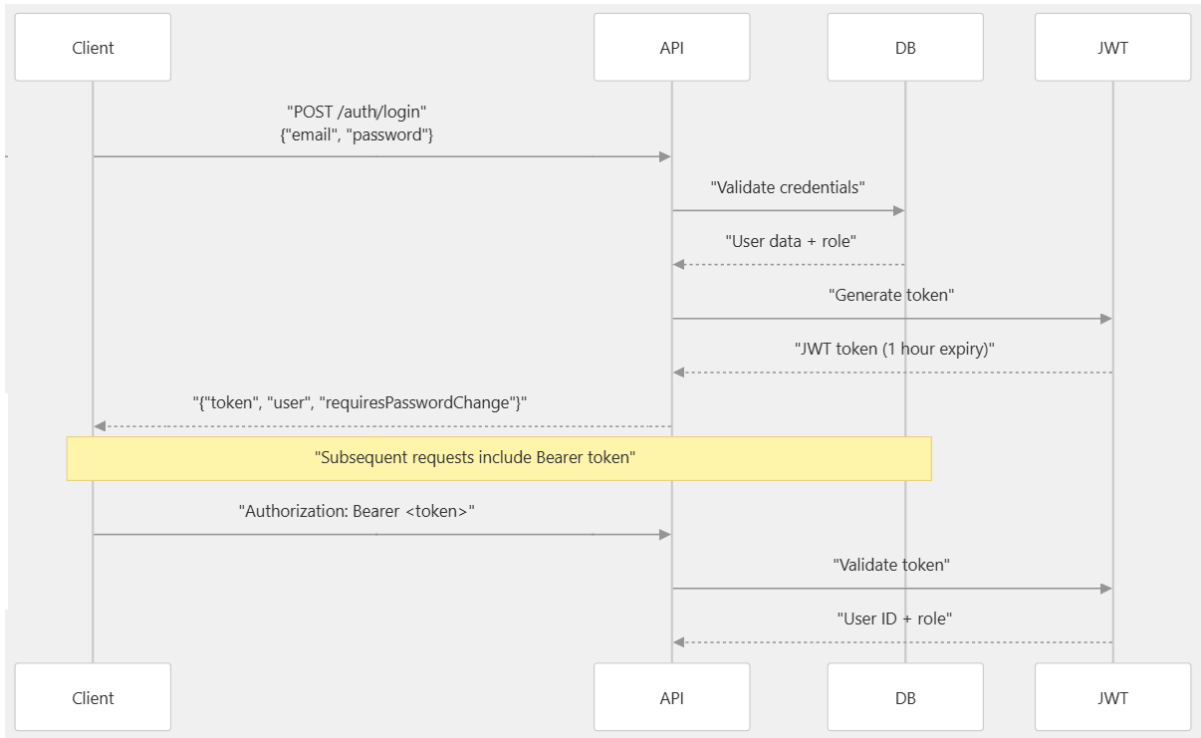


Figura 4.11: Diagrama de secuencia Petición de Login

### Petición genérica:

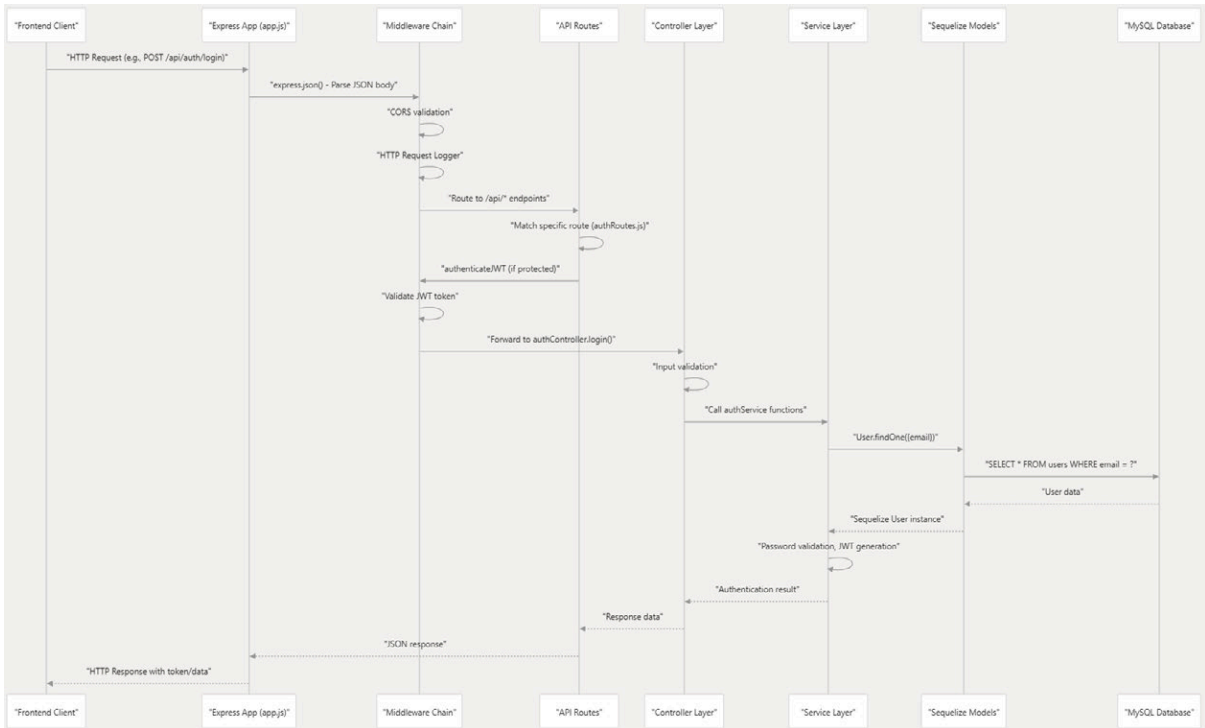


Figura 4.12: Diagrama de secuencia Petición genérica

### Petición de datos de juego y guardado de resultados:

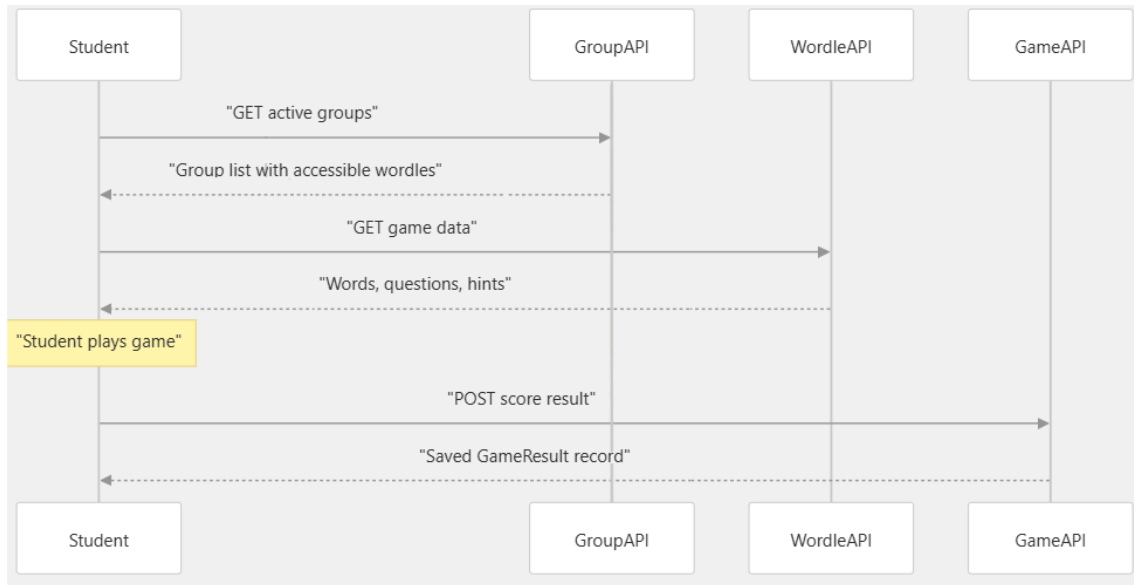


Figura 4.13: Diagrama de secuencia Petición datos de juego y guardado de resultados

## 4.4. Diseño

En este apartado se describen las decisiones técnicas adoptadas a nivel de diseño. Se presenta la arquitectura general del sistema, detallando la distribución en capas y la organización de los principales componentes. Asimismo, se analizan los patrones de diseño y estilos arquitectónicos empleados, explicando su papel en la construcción de una solución escalable, mantenible y coherente con los objetivos del proyecto.

### 4.4.1. Diagrama de clases de diseño

Este apartado presenta el diseño detallado de las clases del sistema, derivado directamente del diagrama de clases de análisis. Representa la estructura estática del *software* especificando las clases, sus atributos, operaciones y las relaciones entre ellas. Debido a su tamaño, el diagrama completo se adjunta en la siguiente figura



Figura 4.14: Diagrama clases de Diseño. [Enlace directo](#)

### 4.4.2. Patrones de diseño

Durante el desarrollo de EduWordle se han aplicado los siguientes patrones de diseño.

#### Singleton

Se trata de un patrón de creación empleado para garantizar que una clase tenga una única instancia accesible desde cualquier parte del sistema. En este proyecto se ha utilizado en:

- *apiService.js* del *frontend*, donde se define un único objeto para gestionar todas las llamadas HTTP.
- La conexión de Sequelize en el *backend*, que se configura una sola vez y se reutiliza en todos los servicios y modelos

## Factory Method

El Factory Method también es un patrón de creación, pero este se emplea para centralizar en una clase constructora la creación de objetos que sean subtipos de un tipo determinado. En este trabajo se aplica en el *backend*, dentro de *errorHandler.js*. Esta clase implementa métodos como *badRequest()*, *unauthorized()*, *notFound()*, etc., que devuelven errores personalizados preconfigurados.

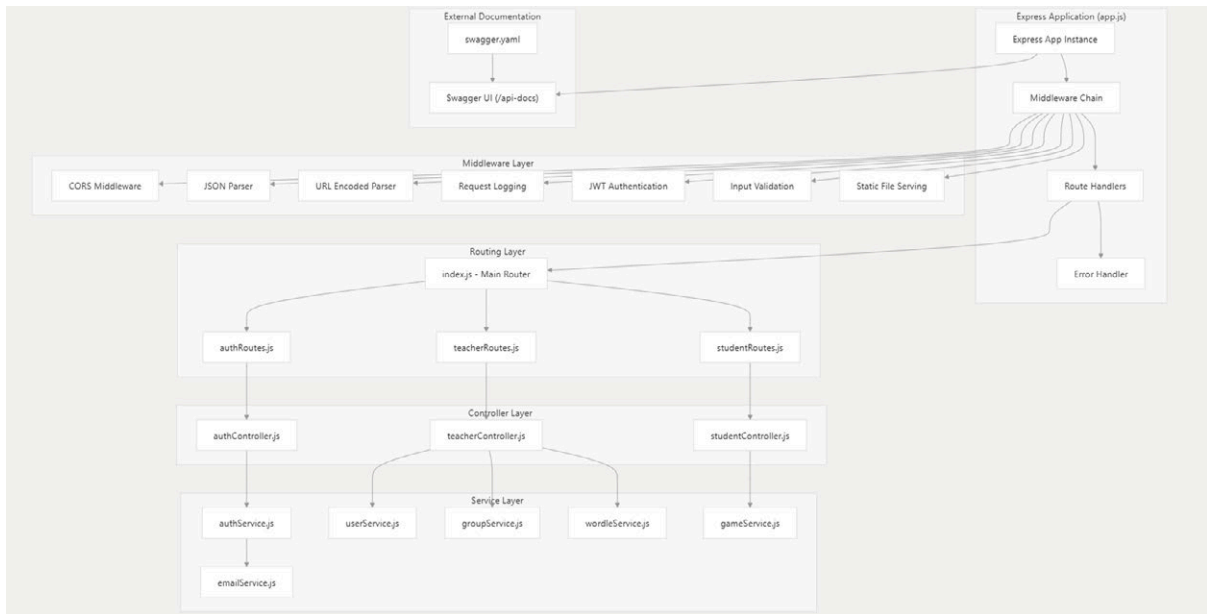
### 4.4.3. Arquitectura de la solución

#### Patrón arquitectónico

EduWordle sigue una arquitectura Three-Tier, esta divide la aplicación en tres capas bien definidas:

- Capa de presentación (*Frontend*): construida con HTML, CSS y JavaScript. Esta capa se ejecuta en el navegador y se encarga de la interacción con el usuario.
- Capa de lógica de aplicación (*Backend*): implementada en Node.js con Express. Se encarga de procesar las solicitudes del cliente, aplicar la lógica de negocio y servir datos.
- Capa de persistencia (Base de datos): gestionada con Sequelize, facilita la interacción con la base de datos relacional subyacente.

Para mejorar la comprensión de la arquitectura, se ha generado un diagrama que visualmente representa la arquitectura mencionada.



**Figura 4.15:** Diagrama de Arquitectura final del Proyecto.

## Estilo arquitectónico

EduWordle incorpora varios estilos arquitectónicos que complementan su estructura:

- *Client-Server* (cliente-servidor): El sistema sigue el modelo tradicional donde el cliente realiza peticiones HTTP al servidor (*backend*), que responde con datos o resultados.
- *Main Program and Subroutines* (programa principal y subrutinas): El flujo de ejecución tanto en el *backend* como en el *frontend* responde a este estilo, en el que una función principal invoca subrutinas auxiliares que realizan tareas concretas.
- *Object-Oriented* (Orientado a Objetos): Aunque JavaScript no está estrictamente orientado a objetos, Sequelize permite definir modelos como clases con atributos y métodos, encapsulando la lógica de cada entidad.

## Flujo de interacción entre capas

Para ilustrar el funcionamiento interno y la interacción dinámica entre los distintos componentes de EduWordle, se ha elaborado un diagrama de secuencia que representa el flujo de control de una operación típica. Abarca desde la solicitud del cliente hasta la persistencia en la base de datos, detallando cómo las peticiones HTTP son procesadas a través del *middleware*, son manejadas por los controladores, procesadas por la capa de servicio y finalmente gestionadas por el ORM Sequelize para interactuar con la base de datos MySQL. Con ello se resalta la modularidad del diseño y la separación de responsabilidades.

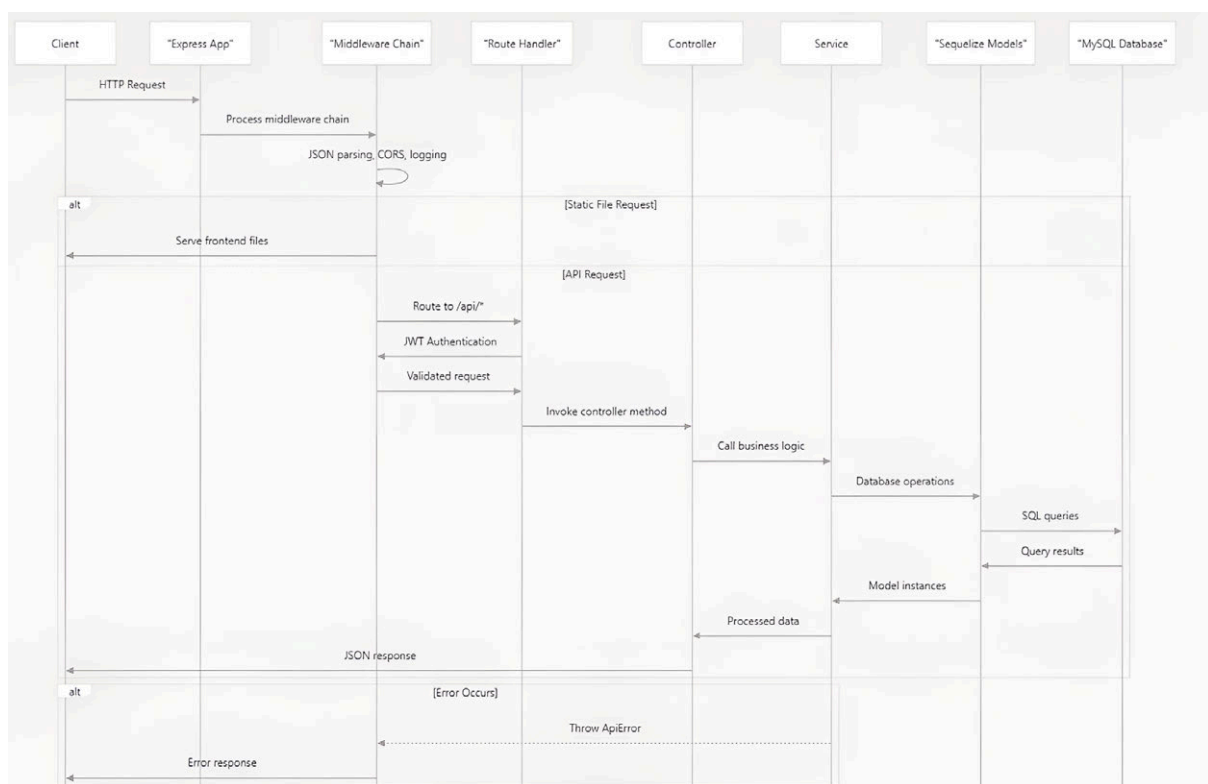


Figura 4.16: Diagrama de Secuencia de la Interacción Cliente-Servidor.

### 4.4.4. Atributos de Calidad

Nombre del Requisito	Descripción
Usabilidad	El sistema EduWordle debe ser intuitivo, fácil de aprender y eficiente de usar tanto para profesores como para alumnos,

	<p>minimizando el esfuerzo cognitivo y los errores. La interfaz debe permitir una navegación clara y una interacción fluida en las tareas de creación, gestión y juego de <i>Wordles</i>.</p>
Seguridad	<p>El sistema debe proteger la información sensible de los usuarios y garantizar que solo los usuarios autenticados y autorizados puedan acceder a las funcionalidades y datos correspondientes a su rol. Se deben prevenir accesos no autorizados y manipulaciones de datos.</p>
Accesibilidad	<p>El sistema debe ser utilizable por personas con diversas capacidades, asegurando que la información y las funcionalidades clave sean perceptibles, operables, comprensibles y robustas para todos los usuarios.</p>
Escalabilidad	<p>El sistema debe ser capaz de manejar un incremento en el número de usuarios concurrentes, más <i>Wordles</i> y más partidas sin una degradación significativa del rendimiento. Para lograrlo se ha hecho uso de Node.js/Express para el <i>backend</i> y su modelo asíncrono y no bloqueante ya que permite manejar un alto volumen de conexiones concurrentes de manera eficiente. La arquitectura de rutas, servicios y modelos promueve la modularidad y facilita la distribución de la carga. La base de datos MySQL, al ser un sistema de gestión de bases de datos relacionales robusto y ampliamente utilizado, permite la optimización de consultas y la escalabilidad vertical/horizontal.</p>
Mantenibilidad	<p>La arquitectura del sistema, el código y la documentación son claros, modulares y fáciles de entender, lo que permite realizar modificaciones, añadir nuevas funcionalidades y corregir errores con un esfuerzo y riesgo mínimos.</p> <p>La arquitectura en capas fomenta una clara separación de responsabilidades, lo que facilita la localización y modificación de funcionalidades específicas. La modularidad del código, el uso de Sequelize para la interacción con la base de datos, y las convenciones de Node.js/Express contribuyen a un código más organizado y fácil de mantener.</p>

Portabilidad (Docker)	<p>El sistema debe ser capaz de ser desplegado y ejecutarse de manera consistente en diferentes entornos operativos y plataformas de servidor con un mínimo de reconfiguración.</p> <p>La contenerización con Docker encapsula la aplicación y sus dependencias en un entorno aislado y reproducible, garantizando un comportamiento idéntico independientemente del sistema host subyacente.</p>
--------------------------	---

*Tabla 9: Atributos de Calidad*

## 4.5. Implementación

En este capítulo se detalla el proceso técnico de construcción de la aplicación, ejemplificado a través de la implementación de la funcionalidad principal: **el Juego**. Se seguirá una narrativa cronológica que refleja la evolución de la funcionalidad a través de los hitos del proyecto: desarrollo del *frontend*, implementación del *backend* y unificación con la API y finalmente la *dockerización* del servicio.

Se destacarán las decisiones que fueron claves para resolver los problemas surgidos durante cada etapa. El código fuente completo y el resto de funcionalidades se encuentran disponibles en el repositorio de GitHub del proyecto que se puede encontrar en el link definido en el apartado de Recursos del Proyecto: [Acceso Repositorio](#).

### 4.5.1. Hito 1: Desarrollo Frontend

La fase inicial se centró en construir una interfaz de juego interactiva puramente en el cliente. La prioridad era validar la mecánica de juego y la experiencia de usuario de forma aislada, utilizando una estrategia de prototipado rápido con un servidor simulado (live-server).

## Uso de placeholders y plantillas

La estructura del fichero *game.html* se diseñó con un enfoque minimalista, definiendo "contenedores vacíos" destinados a ser manipulados dinámicamente por JavaScript. En lugar de codificar interfaces complejas, se adoptó un patrón de plantillas `<template>` para componentes como los *popups*.

```
<!-- Placeholder para el tablero, será rellenado por JS -->
<div id="game-board"></div>

<!-- Plantilla para el popup de final de partida -->
<template id="game">
  <div class="endGame-container">
    <h1>FIN DEL JUEGO</h1>
    <p>Has obtenido <span id="points">0</span> puntos !</p>
    <button id="endButton" class="save-button">Volver al
inicio</button>
  </div>
</template>
```

Figura 4.17: Fragmento de *game.html* - placeholder del tablero.

Este diseño desacopla la estructura HTML de la lógica de la interfaz, permitiendo una mayor mantenibilidad y facilitando la creación dinámica de elementos como el tablero de juego a través de la manipulación del DOM en *game.js*.

## Gestión de elementos asíncronos

El reto técnico más significativo del *frontend* fue la gestión de la asincronía. El flujo del juego requiere pausar la ejecución para esperar la respuesta del usuario a una pregunta emergente. Este problema se resolvió mediante el uso de "async/await" y el uso de variables globales de estado para rastrear el progreso, como *guessesRemaining*, *currentGuess* y *isProcessingGuess*.

```
async function checkGuess() {
  // ... lógica de validación del intento ...

  if (guessString === rightGuessString) {
    // La ejecución se pausa aquí hasta que el usuario responda en
```

```

el popup
  const answer = await popup_quest();

  if (answer) {
    // Se espera a que la animación termine antes de proceder
    await animateRow(row, letterColor, guessString);
    loadWord(); // Se carga la siguiente palabra
  }
}

isProcessingGuess = false;

// ...
}

```

**Figura 4.18:** Fragmento de game.js - Ejemplo de lógica asíncrona

Esta implementación garantiza que la secuencia (intento -> pregunta -> resultado -> siguiente) se respete rigurosamente, evitando condiciones de carrera y manteniendo una interfaz fluida.

## Uso de live-server y mock de backend

Para que el *frontend* pudiera consumir datos realistas desde el principio, se tomó la decisión de crear un servidor temporal simulado con Node.js y Express. Este servidor simulaba la futura *API REST*, exponiendo endpoints básicos que devolvían datos estáticos o de una base de datos de prueba.

```

// Simula el endpoint que devuelve la lista de palabras
app.get('/words', (req, res) => {
  // En un futuro, estos datos vendrían de una consulta a la BD
  real
  const mockWords = [{ word: 'TEST', hint: 'Prueba' }, ...];
  res.json(mockWords);
});

```

**Figura 4.19:** Fragmento del servidor mock server.js

El *frontend*, a su vez, realizaba las llamadas a este servidor local usando la *API fetch*, tratando la respuesta como si proviniera del *backend* definitivo.

```
async function fetchWords() {
  try {
    // La URL apunta al servidor mock local durante esta fase
    const response = await fetch('http://localhost:3000/words');
    const words = await response.json();
    return words;
  } catch (error) {
    console.error("Error al obtener palabras:", error);
    return [];
  }
}
```

**Figura 4.20:** Fragmento de game.js - Llamada fetch al servidor simulado

Esta estrategia de prototipado de *API* fue fundamental para desacoplar los equipos de trabajo, definir los contratos de datos (qué información se espera en cada petición) y poder avanzar en la lógica de manejo de datos del *frontend* mucho antes de que el *backend* final esté completado.

## 4.5.2. Hito 2: Desarrollo Backend

En esta fase, el servidor simulado fue sustituido por la *API RESTful* completa, implementada con Node.js, Express y Sequelize. Se desarrollaron los modelos, controladores y rutas para gestionar la lógica de negocio real, incluyendo la autenticación, la gestión de roles, y el acceso a los datos de la base de datos MySQL definitiva.

Las llamadas *fetch* en el *frontend*, que antes apuntaban a *localhost:5000*, fueron actualizadas para apuntar a los *endpoints* de producción de la nueva *API* y gracias al prototipado inicial, esta transición se realizó de forma fluida y con mínimas refactorizaciones.

Siguiendo el diseño utilizado, la lógica de negocio se encapsuló en los archivos de controladores. Estos ficheros son llamados por la capa de enrutación tras procesar

las peticiones HTTP, y son los encargados de filtrar los errores de validación y llamar a los servicios, quienes finalmente interactúan con los modelos de Sequelize para leer o escribir en la base de datos, y formular la respuesta.

Para garantizar la integridad de los datos, toda la lógica de negocio que implicaba múltiples operaciones de base de datos se encapsuló en la capa de Servicios y se gestionó mediante transacciones. Si cualquier operación dentro de la transacción falla, se ejecuta un *rollback* para deshacer todos los cambios, manteniendo la base de datos en un estado consistente.

```
const saveGameResult = async (userId, wordleId, score) => {
  const transaction = await sequelize.transaction();
  try {
    // Todas las operaciones de BD usan la misma transacción
    const gameResult = await GameResult.findOne({ where: {...},
transaction });

    // ... lógica para crear o actualizar el resultado ...
    await gameResult.save({ transaction });

    // Si todo va bien, se confirman los cambios
    await transaction.commit();
    return gameResult;
  } catch (error) {
    // Si algo falla, se deshacen todos los cambios
    await transaction.rollback();
    throw new Error('Failed to save game result');
  }
};
```

**Figura 4.21:** Fragmento de gameService.js - Ejemplo de rollback

Finalmente, las rutas de la *API* definidas, actúan como el punto de entrada que conecta las solicitudes HTTP con los controladores correspondientes. Se utiliza el Router de Express para organizar los puntos finales de forma modular.

Un aspecto crucial en esta capa es la implementación de *middleware* para la seguridad. Cada ruta protegida pasa primero por un *middleware* de autenticación que valida el token JWT del usuario, y luego por uno de autorización que verifica si el usuario tiene el rol adecuado para acceder a ese recurso. Por último, las rutas que

lo requieren también pasan el mensaje de *request* por un middleware que verifica que los datos enviados tengan la estructura esperada.

```
router.use(authenticateJWT);
router.use(authorizeRole(['student']));

// ...
// Route to get game data for a specific wordle
// GET /api/student/wordles/:wordleId/game-data
router.get('/wordles/:wordleId/game-data',
studentController.getWordleGameData);

// Route to save the result of a completed game
// POST /api/student/games/:wordleId/save-result
router.post('/games/:wordleId/save-result',
  validateSaveGameResult,
  studentController.saveGameResult
);

module.exports = router;
```

**Figura 4.22:** Fragmento de studentRoutes.js - Rutas protegidas y manejo de resultados de juego

## Implementación de la base de datos

En este momento, la persistencia se basaba en modelos de datos de cada una de las tablas y se usó Sequelize para definir las. Un ejemplo de estas tablas es *gameResult* en donde se guardaban los resultados de las partidas

```
const { DataTypes } = require('sequelize');
const sequelize = require('../config/database');

const GameResult = sequelize.define('GameResult', {
  id: {
    type: DataTypes.INTEGER,
    autoIncrement: true,
    primaryKey: true
  },
  userId: { // Clave foránea que referencia al estudiante
```

```

    type: DataTypes.INTEGER,
    references: { model: 'user', key: 'id' }
  },
  wordleId: { // Clave foránea que referencia al Wordle jugado
    type: DataTypes.INTEGER,
    references: { model: 'wordle', key: 'id' }
  },
  score: {
    type: DataTypes.INTEGER,
    defaultValue: 0
  }
}, {
  tableName: 'game',
  timestamps: true,
  indexes: [
    // Índice único para asegurar una sola partida por alumno y
    Wordle
    { unique: true, fields: ['userId', 'wordleId'] }
  ]
});

module.exports = GameResult;

```

**Figura 4.23:** Fragmento de gameResult.js - Definición del modelo de partida

Para la fase de implementación del sistema, se procedió a la materialización del diseño físico de la base de datos. Por ello, este diseño representa la estructura final que alberga los datos de EduWordle en MySQL, optimizada para la eficiencia en el desarrollo y la gestión de la información.

Cabe destacar una decisión de diseño relevante en la gestión de usuarios ya que, a diferencia del modelo conceptual de datos presentado en el apartado de Análisis donde se contemplaban entidades separadas para "Student" y "Teacher", se optó por una estrategia de herencia de tabla única. De este modo se implementó una única tabla "User" que incluye el atributo "role" para distinguir entre estudiantes y profesores. Esta consolidación se justificó por la superposición de funcionalidades y atributos entre ambos tipos de usuarios, simplificando significativamente el desarrollo y la gestión de autenticación y autorización.

La siguiente figura ilustra el diseño final de la base de datos de EduWordle, tal como fue implementada. Este diagrama ha sido generado mediante un proceso de ingeniería inversa directamente desde la base de datos real en MySQL.

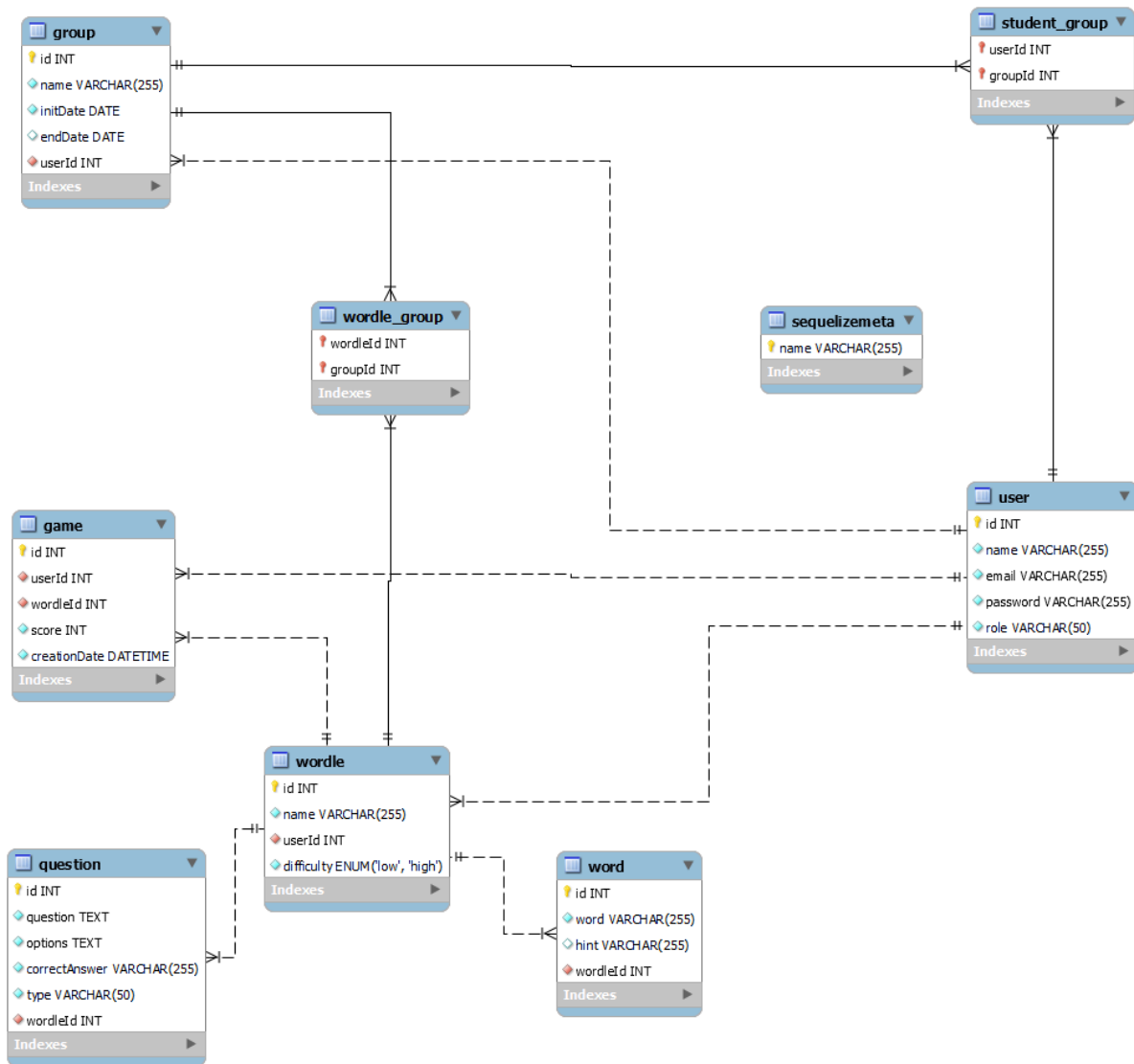


Figura 4.24: Ingeniería inversa: diagrama de la Base de Datos

Al final de este hito el sistema ya se podía considerar completo.

### 4.5.3. Hito 3: Dockerización

La última fase del proyecto consistió en encapsular la aplicación completa en contenedores Docker para asegurar la portabilidad, simplificar el entorno de

desarrollo y estandarizar el proceso de despliegue. El objetivo era crear un sistema que pudiera levantarse con un solo comando, con todos sus servicios (*Frontend*, *Backend*, Base de Datos) interconectados y preconfigurados.

Se creó un *Dockerfile* para definir la imagen del servicio de *backend*. Este archivo automatiza la creación de un entorno Node.js, la instalación de dependencias y la configuración inicial necesaria para ejecutar la aplicación.

```
# Se usa una imagen base oficial de Node.js con Alpine para un
tamaño reducido
FROM node:18-alpine

# Se establece el directorio de trabajo dentro del contenedor
WORKDIR /app/backend

# Se copian los archivos de dependencias, se reinician y se
instalan de nuevo
COPY backend/package.json ./
RUN npm cache clean --force && npm install
RUN npm install -g sequelize-cli

# Se copia el resto del código de la aplicación
COPY backend/. ./
COPY backend/docker-entrypoint.sh
/usr/local/bin/docker-entrypoint.sh
RUN chmod +x /usr/local/bin/docker-entrypoint.sh

# Se expone el puerto en el que correrá el backend
EXPOSE 3000

# Se define un entrypoint para asegurar que la base de datos esté
lista
# antes de iniciar el servidor Node.js.
ENTRYPOINT ["docker-entrypoint.sh"]
CMD ["npm", "start"]
```

**Figura 4.25:** Fragmento de Dockerfile - Definición del servicio de backend

Una decisión de diseño importante fue la inclusión de un script *entrypoint.sh*. Este *script* ejecuta una comprobación en bucle para asegurar que la base de datos

MySQL esté completamente iniciada y aceptando conexiones antes de lanzar el servidor de Node.js, evitando así errores de conexión durante el arranque.

```
#!/bin/sh
echo "Waiting for MySQL database..."
# El comando 'nc' comprueba si el puerto está abierto en el host
de la BD
until nc -z -w 5 db 3306; do
    echo "MySQL is unavailable - sleeping"
    sleep 2
done

echo "MySQL is up - executing migrations"
// ...
# Ejecuta las migraciones y seeders
/usr/local/bin/npm exec sequelize-cli db:migrate

echo "Executing database seeds..."
/usr/local/bin/npm exec sequelize-cli db:seed:all

echo "Starting Node.js application"
# Asegúrate de que NODE_ENV se establece aquí si no está en .env o
si quieres sobrescribir
export NODE_ENV=production

# Ejecutar el script prestart.
node ./src/utils/generateFrontendConfig.js
npm run prestart
# Iniciar la aplicación Node.js
exec npm start
```

**Figura 4.26:** Fragmento de entrypoint.sh - Script de espera

Para gestionar los servicios interdependientes (*backend* [que a su vez aloja a *frontend*] y base de datos), se utilizó Docker Compose. El archivo *docker-compose.yml* define cómo estos tres contenedores deben construirse, configurarse e interconectarse.

```
services:
  # Servicio de la base de datos
  db:
```

```

image: mysql:8.0
restart: always
environment:
  MYSQL_DATABASE: ${DB_NAME}
  MYSQL_USER: ${DB_USER}
  MYSQL_PASSWORD: ${DB_PASSWORD}
volumes:
  - db-data:/var/lib/mysql
# ...otras variables...

# Servicio del backend
backend:

  # ...otras variables...
  depends_on:
    - db # Asegura que el contenedor de la BD se inicie antes
que el backend

volumes:
  db-data:

```

**Figura 4.27:** Fragmento de docker-compose.yml

En esta configuración, se utiliza un volumen (db-data) para persistir los datos de la base de datos incluso si los contenedores se retiran o se eliminan, lo cual es fundamental para un entorno de producción. Además, la comunicación interna entre el *backend* y la base de datos se realiza de forma segura a través de la red interna de Docker, utilizando el nombre del servicio (db) como *hostname*.

Un desafío clave al trabajar con Docker es que el esquema de la base de datos debe ser creado y poblado automáticamente cuando el sistema se inicia por primera vez. No es viable confiar en que Sequelize (*sequelize.sync()*) cree las tablas, ya que este método es propenso a errores y no gestiona cambios estructurales en una base de datos ya existente.

Para solucionar esto, se adoptó un enfoque basado en migraciones y *seeders* de Sequelize.

Este flujo de trabajo da como resultado un proyecto consistente, reproducible y fácilmente desplegable, terminando así el ciclo de desarrollo completo de la aplicación EduWordle.

# 5. EduWordle

---

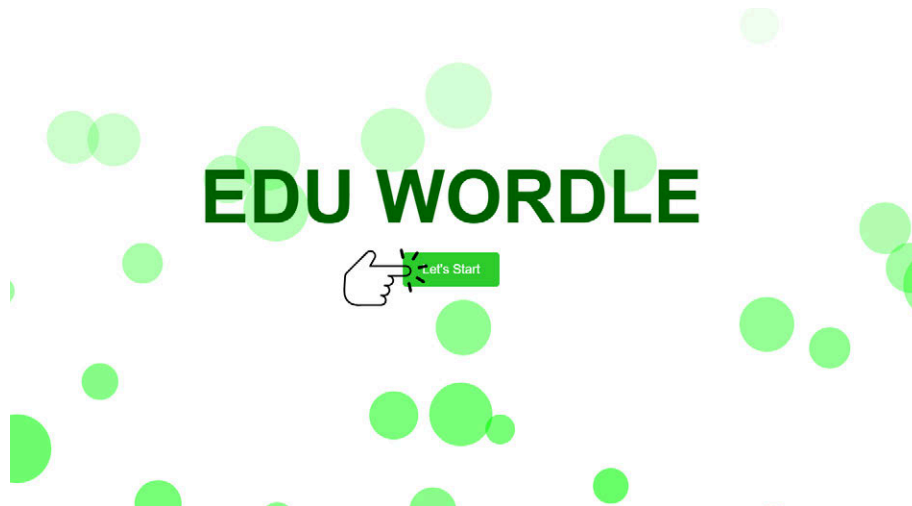
## 5.1. Manual de Usuario y flujo de navegación

En este apartado se expone el recorrido visual de los usuarios a través de las principales funcionalidades de la plataforma EduWordle. Para facilitar su comprensión, se han incluido capturas de pantalla acompañadas de elementos gráficos que muestran los pasos que realiza el usuario dentro de cada funcionalidad.

Esto permite observar la lógica de navegación implementada, tanto desde el perfil del profesorado como del alumnado. La interacción con la plataforma, tal como se presenta en este apartado, se complementa con una demostración en vídeo incluida, que muestra en tiempo real el funcionamiento completo del sistema.

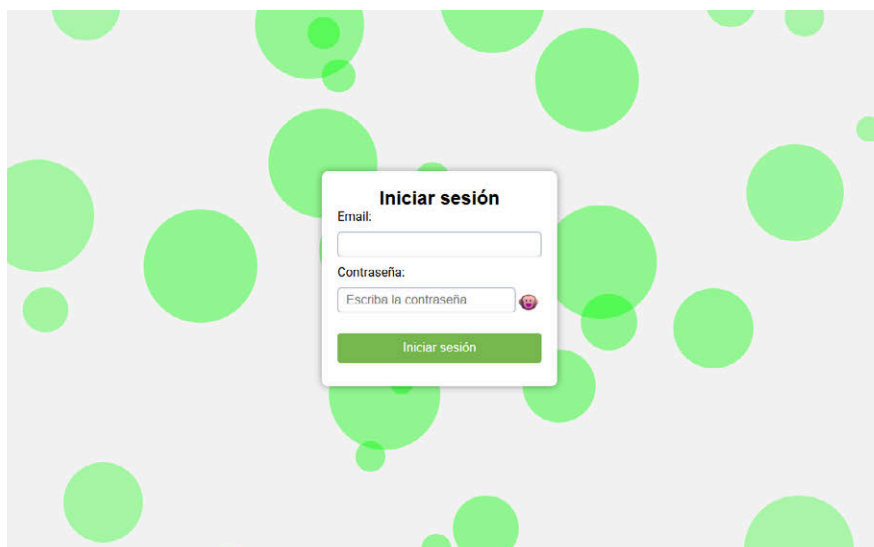
### 5.1.1. Inicio de Sesión

El primer paso del flujo de navegación comienza con la pantalla de acceso al sistema.



**Figura 5.1:** Página de inicio

Segundo paso:

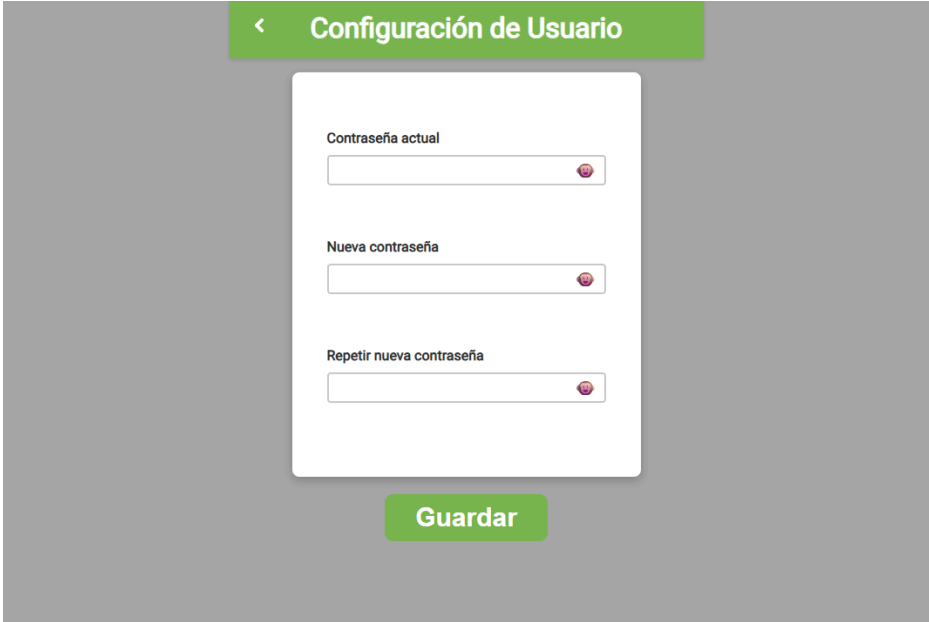


**Figura 5.2:** Página de login

Desde esta vista, tanto docentes como alumnos introducen sus credenciales para ingresar a la plataforma.

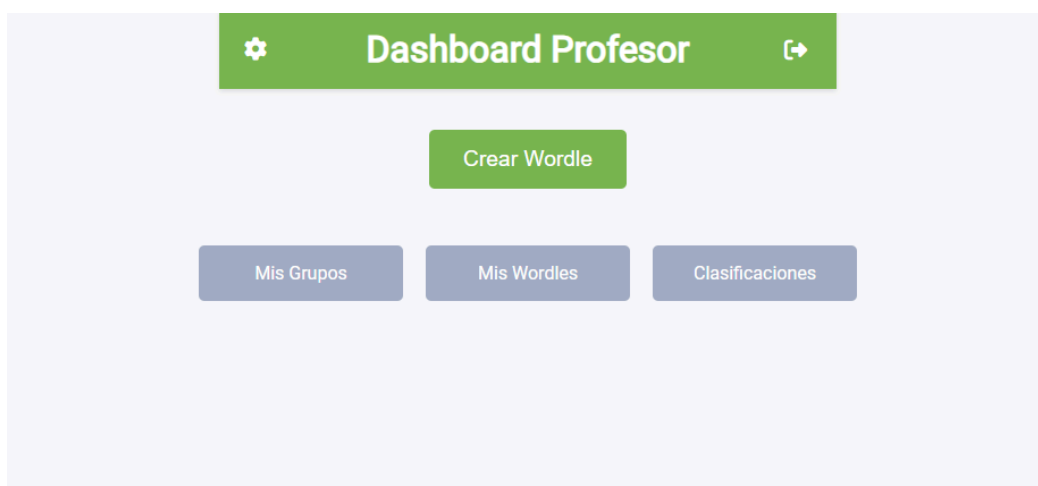
Una vez completado el formulario de inicio de sesión, el sistema verifica las credenciales. En caso de ser correctas, si es la primera vez que se inicia sesión, el sistema redirigirá a la pantalla de configuración donde se le pedirá al usuario (profesor o alumno) que modifique su contraseña. La contraseña tendrá que tener

una estructura específica para ser aceptada. En específico una contraseña fuerte deberá de tener al menos 8 caracteres, una letra mayúscula, una letra minúscula, un dígito y un carácter especial.

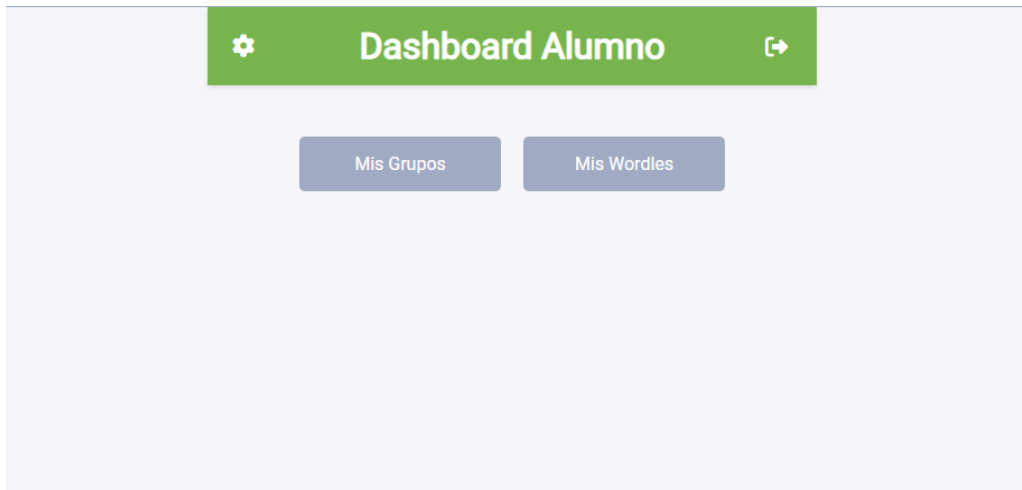


**Figura 5.3:** Forzado de cambio de contraseña

Si no es la primera vez que se inicia sesión, dirige al usuario a su *dashboard* correspondiente (profesor o alumno).

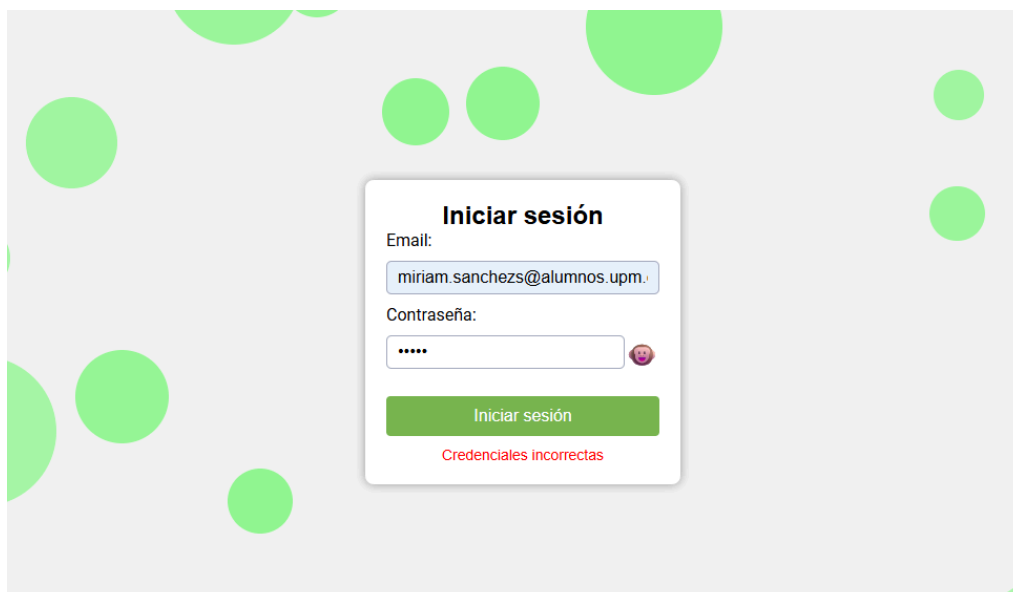


**Figura 5.4:** Dashboard Profesor



**Figura 5.5:** *Dashboard Alumno*

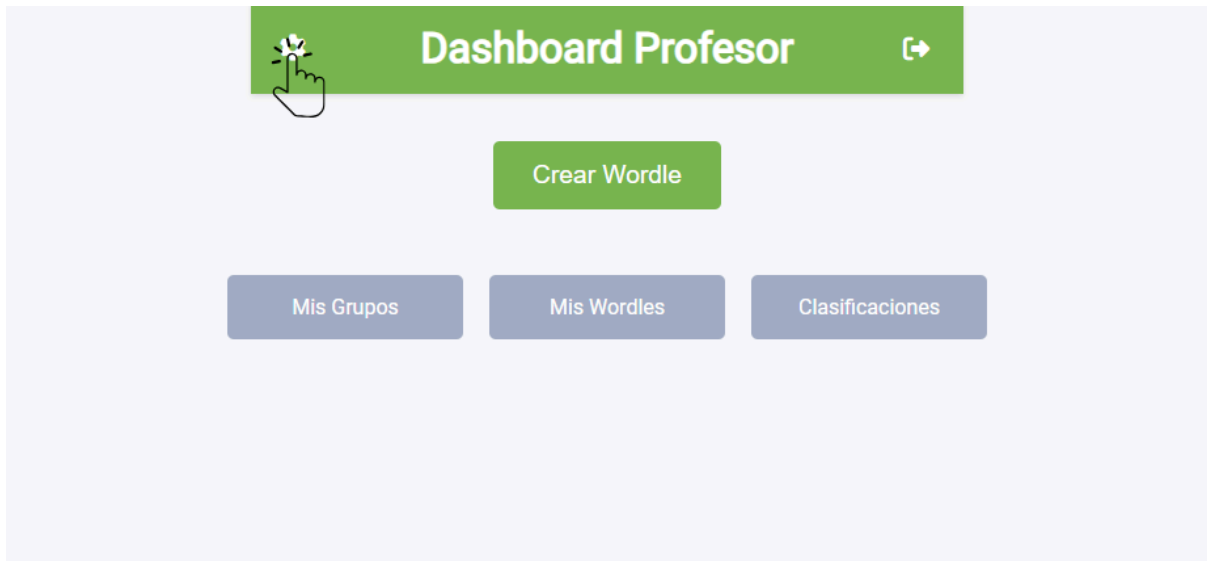
Si los datos no son válidos, se muestra un mensaje de error indicando que el nombre de usuario o la contraseña son incorrectos.



**Figura 5.6:** Login erróneo

### 5.1.2. Gestión de Contraseñas

En caso de que el usuario quiera modificar su contraseña, puede acceder a la rueda/engranaje de configuración ubicada en la pantalla arriba a la izquierda.

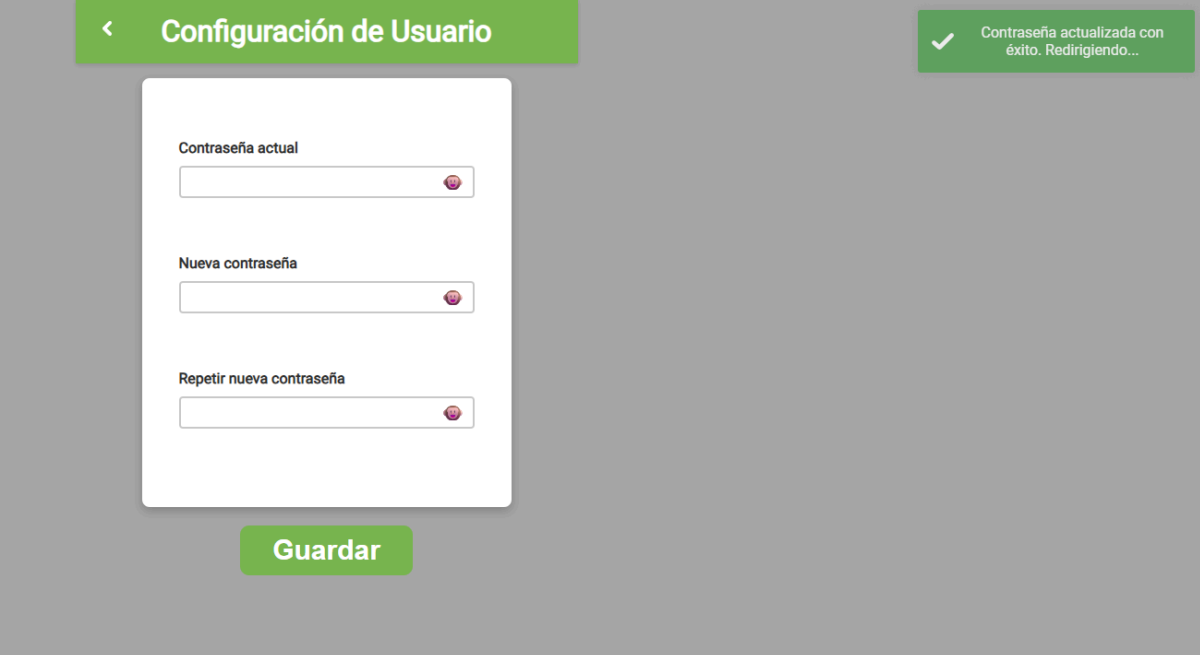


**Figura 5.7:** Camino hacia configuración desde *Dashboard* de Profesor

Al hacer clic, se muestra un formulario donde debe introducir su contraseña actual y la nueva. Al igual que se ha explicado anteriormente en la explicación del [Inicio de Sesión](#), la contraseña debe de cumplir los requisitos de una contraseña fuerte para ser aceptada por el sistema.

**Figura 5.8:** Página de cambio de contraseña

Una vez actualizada, muestra un mensaje de éxito indicando que la contraseña se ha actualizado correctamente y redirige a la página inicial.

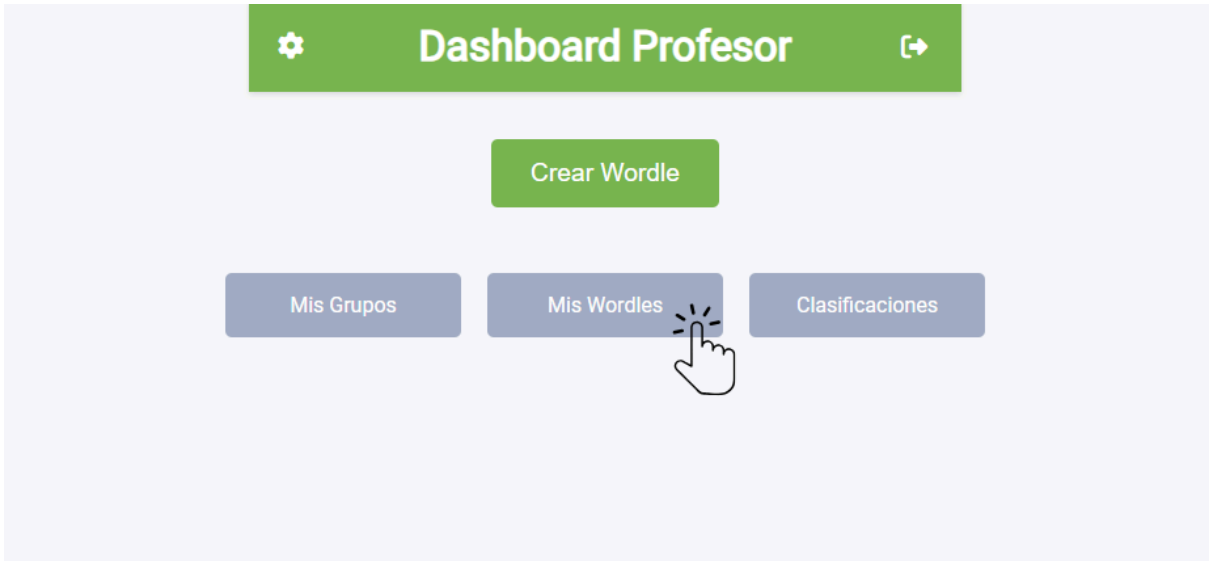


The screenshot shows a user configuration page titled "Configuración de Usuario". It features three input fields for password management: "Contraseña actual", "Nueva contraseña", and "Repetir nueva contraseña". Each field has a red eye icon to toggle visibility. A green "Guardar" button is positioned below the fields. In the top right corner, a green notification box with a checkmark icon displays the message: "Contraseña actualizada con éxito. Redirigiendo..."

**Figura 5.9:** Página de cambio de contraseña con mensaje de éxito

### 5.1.3. Creación y Edición de *Wordles*

Desde el panel del docente, se accede al apartado de "Mis *Wordles*", donde es posible crear nuevas actividades o editar las ya existentes, además desde el botón "Crear *Wordle*" también se pueden crear nuevos.



**Figura 5.10:** Camino hacia Lista de *Wordles* desde *Dashboard* de Profesor



**Figura 5.11:** Camino hacia creación de nuevo *Wordle* desde Lista de *Wordles*

← **Crear Wordle**

**Nombre del Wordle:**  
Introduce el nombre del Wordle

**Dificultad** ⓘ  
Bajo Alto

**Palabras** ⓘ  
+

**Preguntas** ⓘ  
+

**Grupos Asociados**  
+

**Guardar**

Figura 5.12: Página de creación de *Wordle*

El profesorado puede configurar cada *Wordle* con preguntas, palabras, un nivel de dificultad y grupos ya creados previamente. Cada elemento añadido tiene unos criterios de validación que si no se cumplen, no permitirá añadir al elemento al *Wordle*. En primer lugar, las preguntas deben de tener todos los campos cubiertos con al menos dos opciones de respuesta y al menos una de ellas ser correcta. Las palabras permiten opcionalmente añadir una pista sobre la misma y los grupos deben ser elegidos de la lista de grupos creados por el profesor. El *Wordle* per se necesita obligatoriamente al menos una palabra, una pregunta y un nombre.

Al darle a guardar aparece un mensaje de éxito y redirige al *dashboard*.

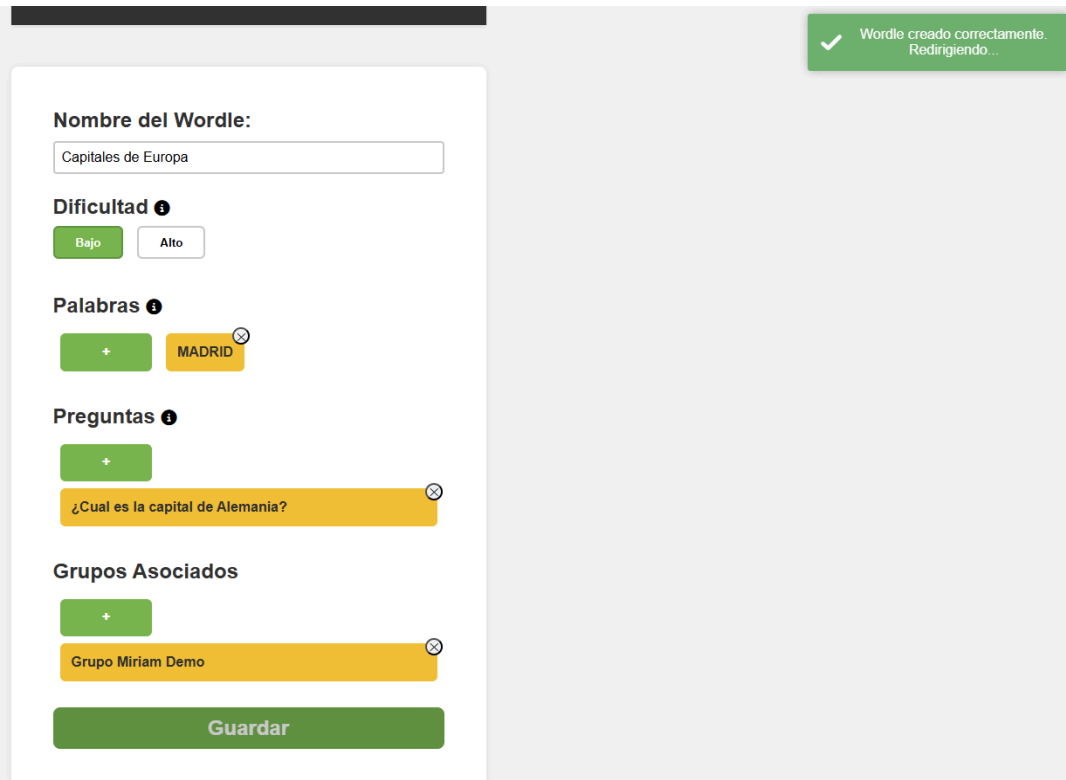


Figura 5.13: Mensaje de éxito de creación de *Wordle*

También es posible editar o eliminar actividades creadas previamente. Esta funcionalidad permite adaptar el contenido a diferentes asignaturas o niveles educativos.

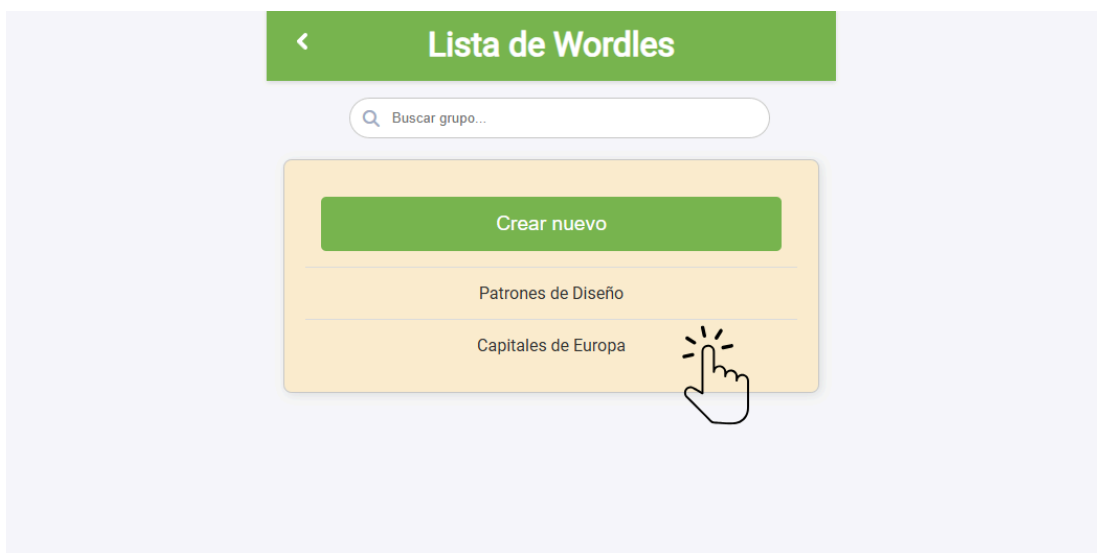


Figura 5.14: Camino a visualización de *Wordle* desde Lista de Wordles

Editar:

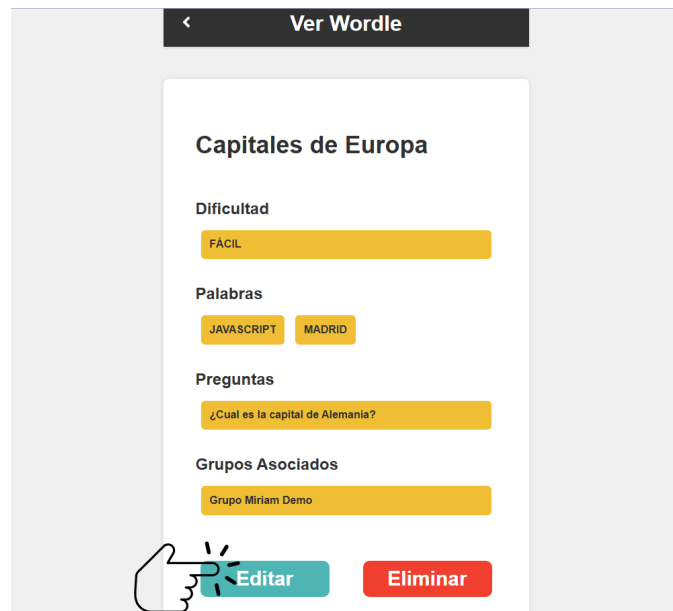


Figura 5.15: Botón de Editar en página de visualización de *Wordle*

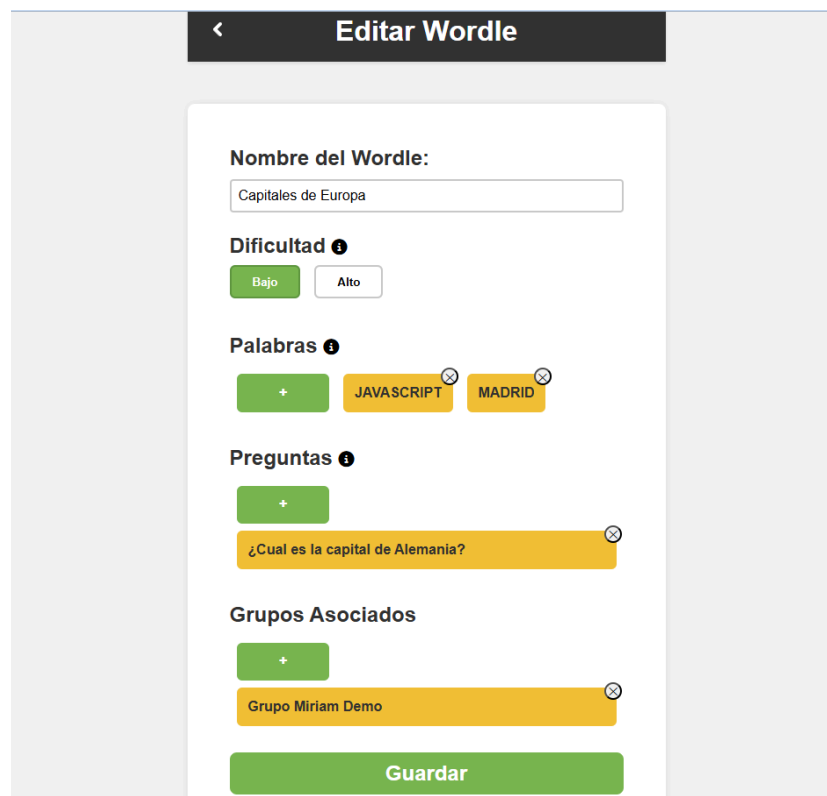


Figura 5.16: Página de edición de *Wordle*

Se pueden añadir/eliminar tanto palabras, preguntas y grupos, así como modificar su nivel de dificultad. El *Wordle* posee las mismas restricciones de validación que en la creación del mismo, se podrán aplicar cambios, pero siempre se debe de mantener al menos una palabra, una pregunta y el nombre.

Al igual que ocurre al crear, al darle al botón “*Guardar*”, lanza un mensaje de éxito y redirige al *dashboard*.

### Eliminar:

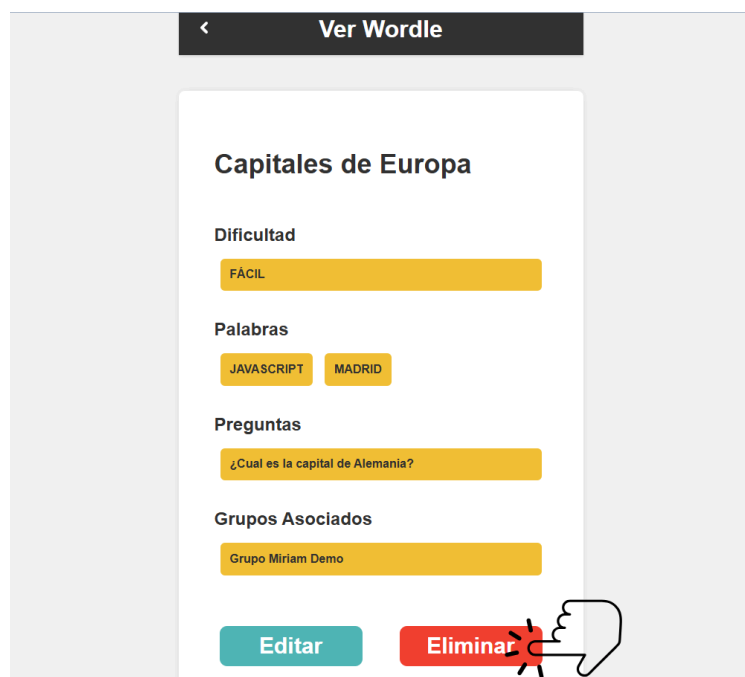


Figura 5.17: Botón de eliminar en la pantalla de visualizar *Wordle*

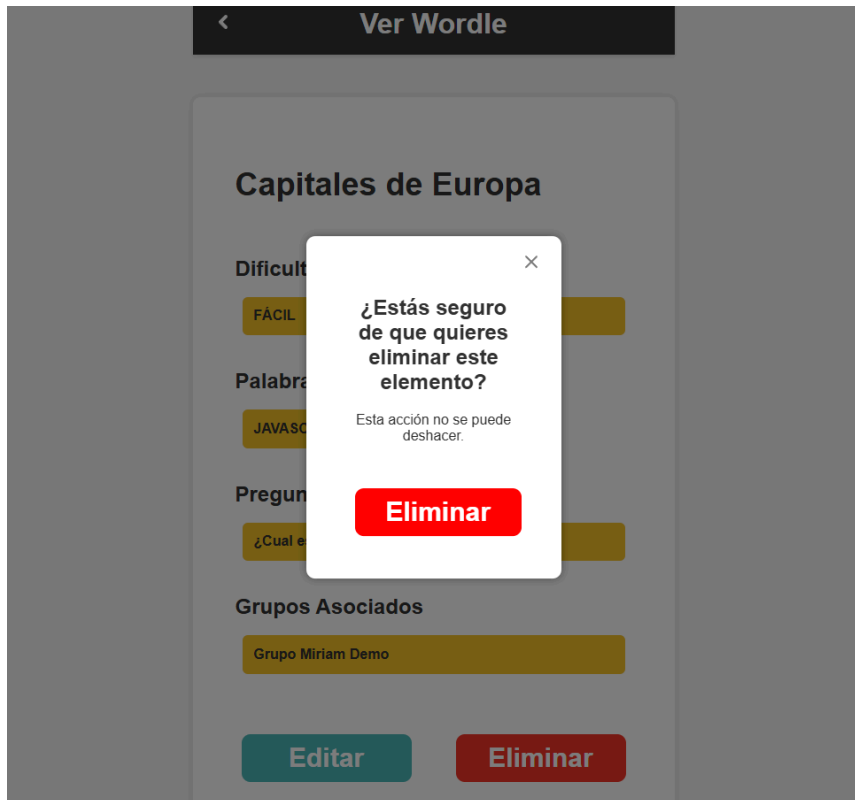
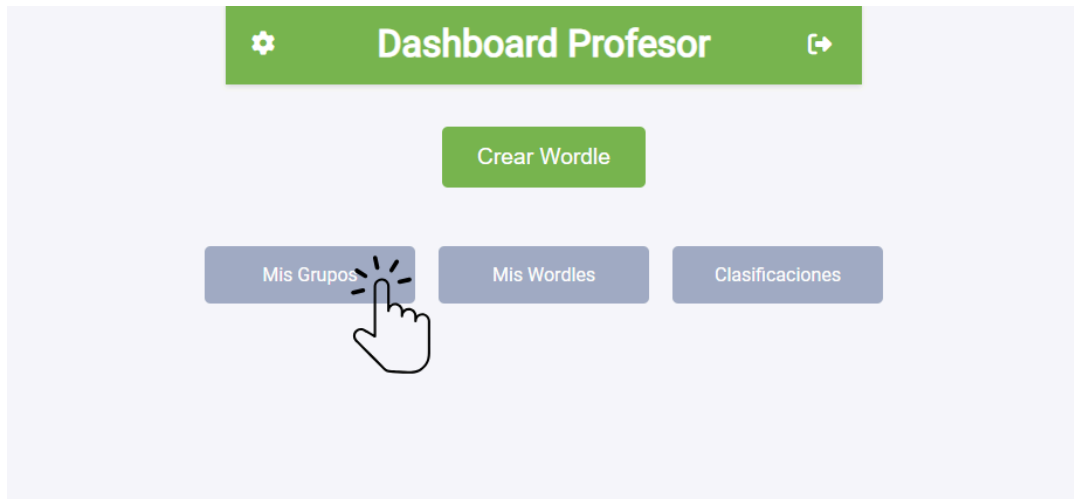


Figura 5.18: Mensaje de confirmación de eliminación de Wordle

Una vez que se le da al botón “Eliminar” se muestra un mensaje de éxito y se redirige al *dashboard*.

#### 5.1.4. Gestión de Grupos

Desde el menú principal, se accede al apartado “Mis grupos” donde el profesorado puede crear y administrar grupos de estudiantes.



**Figura 5.19:** Camino hacia Lista de grupos desde *Dashboard* de Profesor

**Crear nuevo grupo:**



**Figura 5.20:** Camino hacia creación de nuevo Grupo desde Lista de Grupos

En esta pantalla, introducimos los datos del nuevo grupo, y le damos al botón de guardar. Existen dos restricciones a la hora de crear un Grupo, darle un nombre y

una fecha de inicio, esta última se pone automáticamente a la fecha actual si no se define lo contrario.

**Crear Grupo**

**Nombre del Grupo:**  
Introduce el nombre del Grupo

**Alumnos** ⓘ  
+

**Wordles Asociados**  
+

**Configuración**  
Fecha de inicio :  
17/06/2025

Fecha de fin :  
 No definir fecha de fin  
 Definir fecha de fin  
dd/mm/aaaa

**Guardar**

**Figura 5.21:** Página de creación de Grupo

Una vez le hemos dado a “*Guardar*”, se muestra un mensaje de éxito y se redirige al *dashboard* principal.

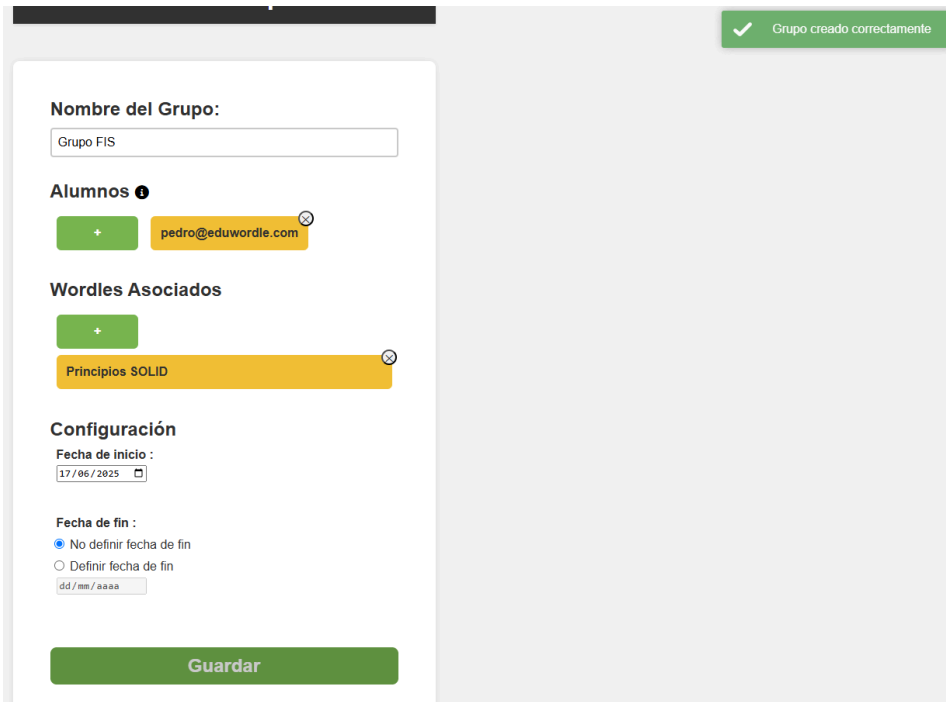
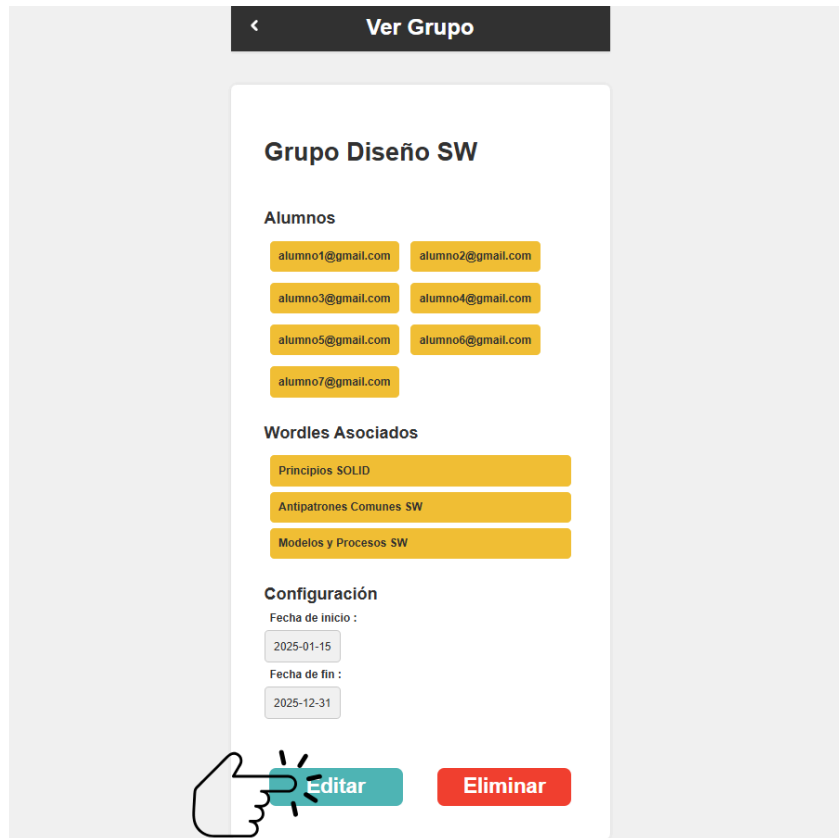


Figura 5.22: Mensaje de éxito de creación de Grupo

### Editar grupo:



Figura 5.23: Camino a visualización de Grupo desde Lista de Grupos



**Figura 5.24:** Botón de Editar en página de visualización de Grupo

En esta pantalla se pueden eliminar/añadir tanto alumnos, como *Wordles* asociados, así como modificar la fecha de inicio y fin del grupo. Del mismo modo que ocurre con la edición de *Wordles*, mantiene las restricciones del apartado de creación.

**Editar Grupo**

**Nombre del Grupo:**  
Grupo Diseño SW

**Alumnos**

- + alumno1@gmail.com
- alumno2@gmail.com
- alumno3@gmail.com
- alumno4@gmail.com
- alumno5@gmail.com
- alumno6@gmail.com
- alumno7@gmail.com

**Wordles Asociados**

- + Principios SOLID
- Antipatrones Comunes SW
- Modelos y Procesos SW

**Configuración**

Fecha de inicio : 15/09/2025

Fecha de fin :  
 No definir fecha de fin  
 Definir fecha de fin 15/12/2025

**Guardar**

**Figura 5.25:** Página de edición de Grupo

Al darle al botón de “*Guardar*”, se muestra un mensaje de éxito y se redirige al *dashboard*.

**Eliminar grupo:**

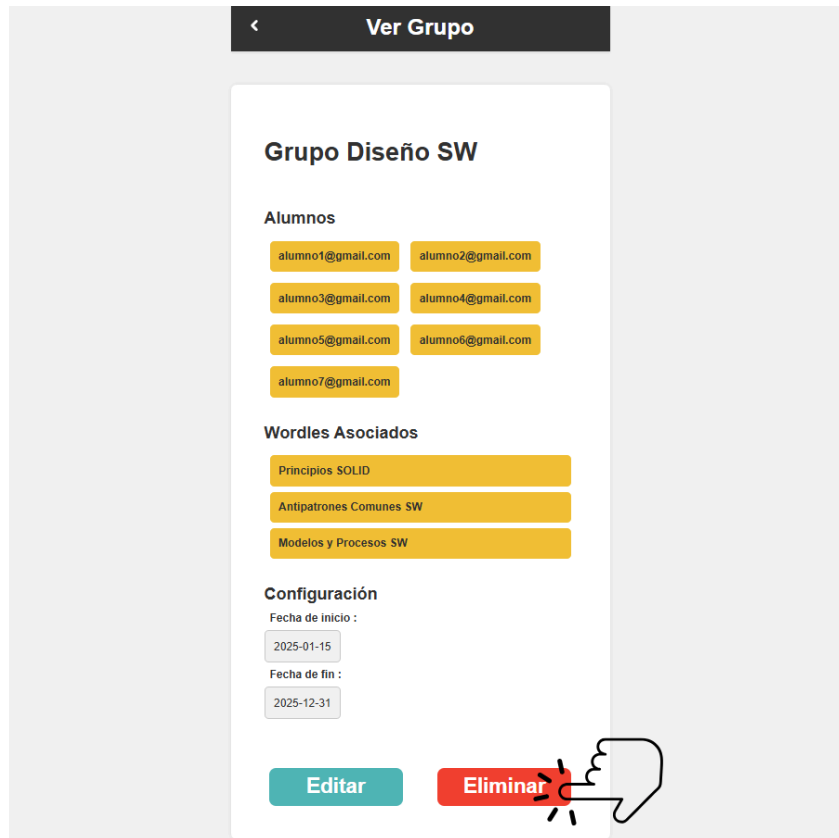


Figura 5.26: Botón de eliminar en la pantalla de visualizar Grupo

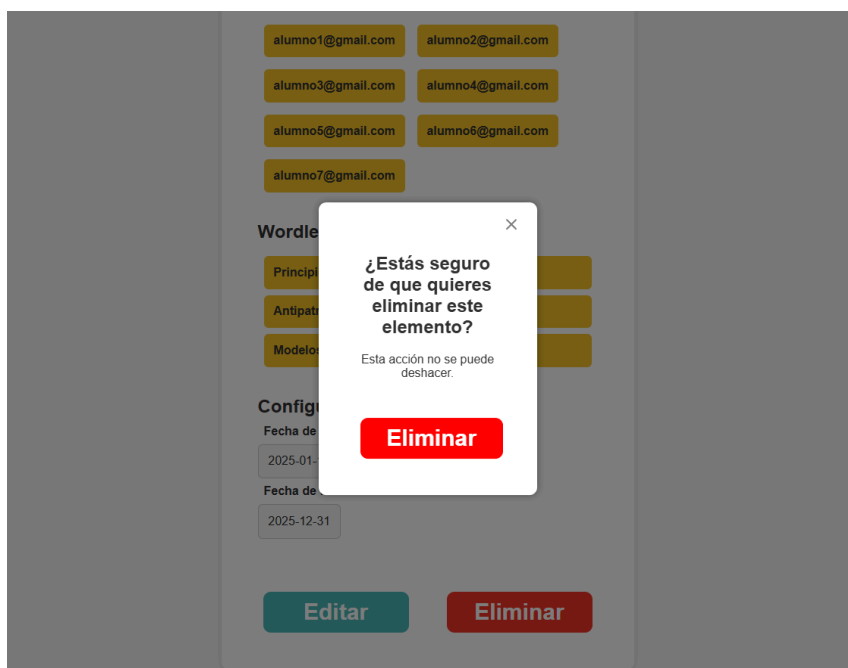


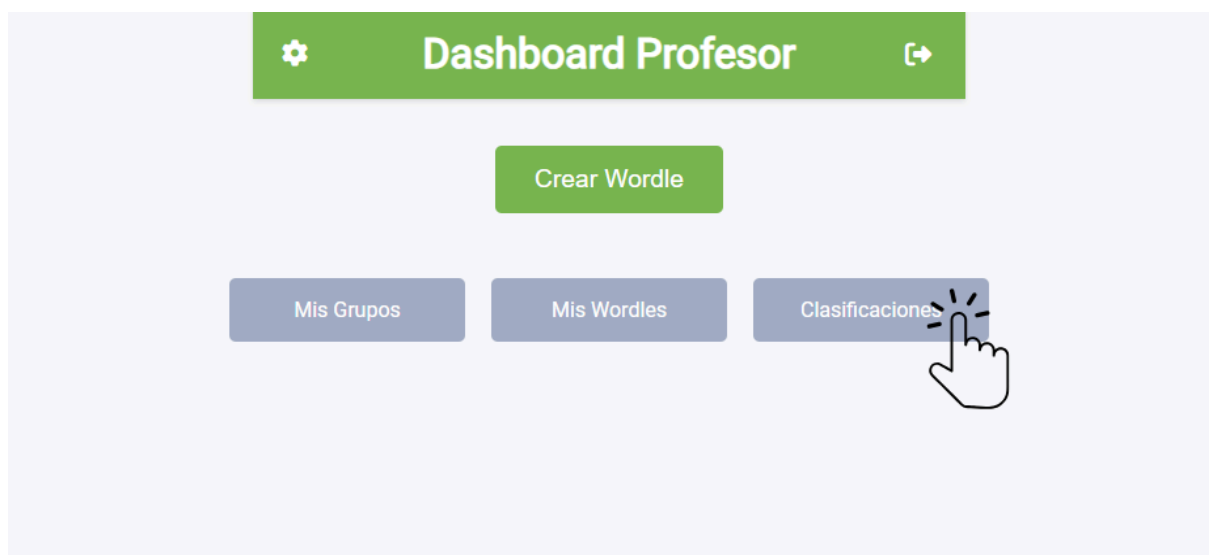
Figura 5.27: Mensaje de confirmación de eliminación de Grupo

Al eliminar un grupo, se muestra un mensaje de éxito y se redirige al *dashboard* inicial.

Cada grupo puede tener *Wordles* asignados, facilitando así la organización del alumnado y la personalización del aprendizaje. Esta función también permite segmentar resultados por clase o nivel.

### 5.1.5. Visualización de Resultados

Los docentes pueden acceder a los resultados de los *Wordle* o de los grupos. Se muestran puntuaciones individuales, número de intentos y posición en el *ranking*.



**Figura 5.28:** Camino hacia Clasificaciones desde *Dashboard* de Profesor



Figura 5.29: Menú de clasificaciones

Por grupo:

Clasificaciones

Por Grupo Por Wordle

**Clasificaciones por Grupo**

**Grupo: Grupo Diseño SW**

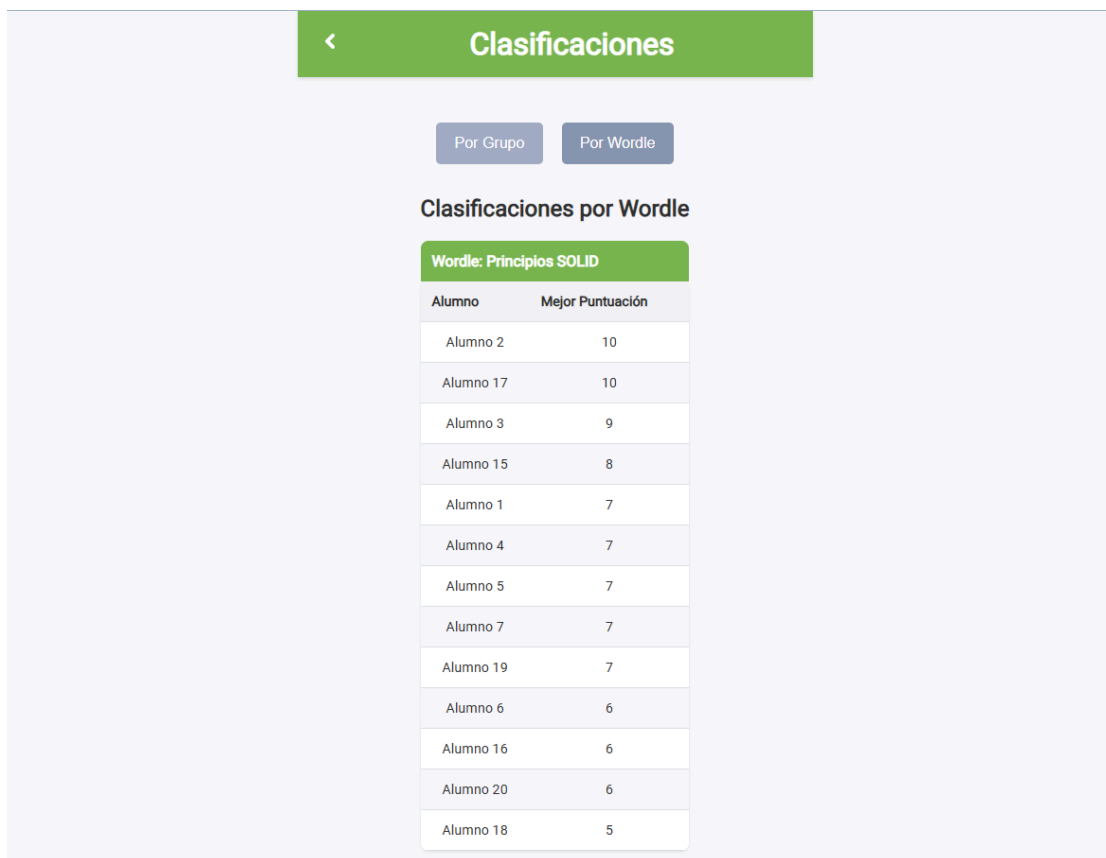
Alumno	Total Puntos
Alumno 1	26
Alumno 2	25
Alumno 4	25
Alumno 6	25
Alumno 3	24
Alumno 5	23
Alumno 7	19

**Grupo: Grupo Pruebas y Calidad**

Alumno	Total Puntos
Alumno 12	18
Alumno 8	16
Alumno 13	16
Alumno 9	14

Figura 5.30: Pantalla de Clasificaciones por Grupo

Por Wordle:



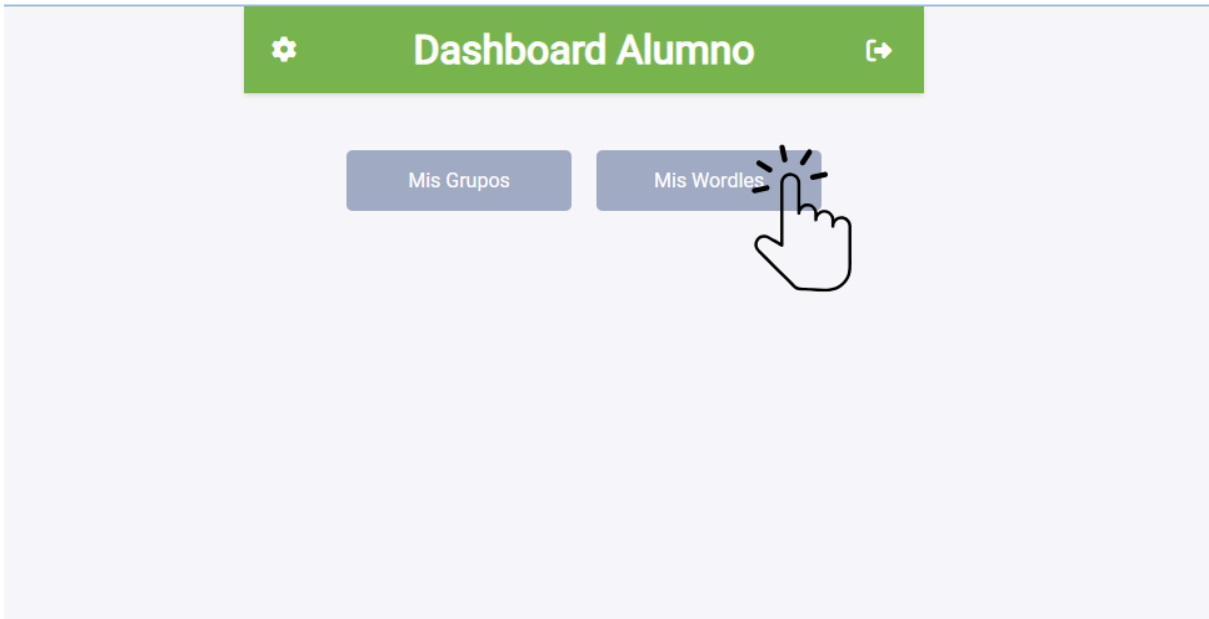
Wordle: Principios SOLID	
Alumno	Mejor Puntuación
Alumno 2	10
Alumno 17	10
Alumno 3	9
Alumno 15	8
Alumno 1	7
Alumno 4	7
Alumno 5	7
Alumno 7	7
Alumno 19	7
Alumno 6	6
Alumno 16	6
Alumno 20	6
Alumno 18	5

Figura 5.31: Pantalla de Clasificaciones por *Wordle*

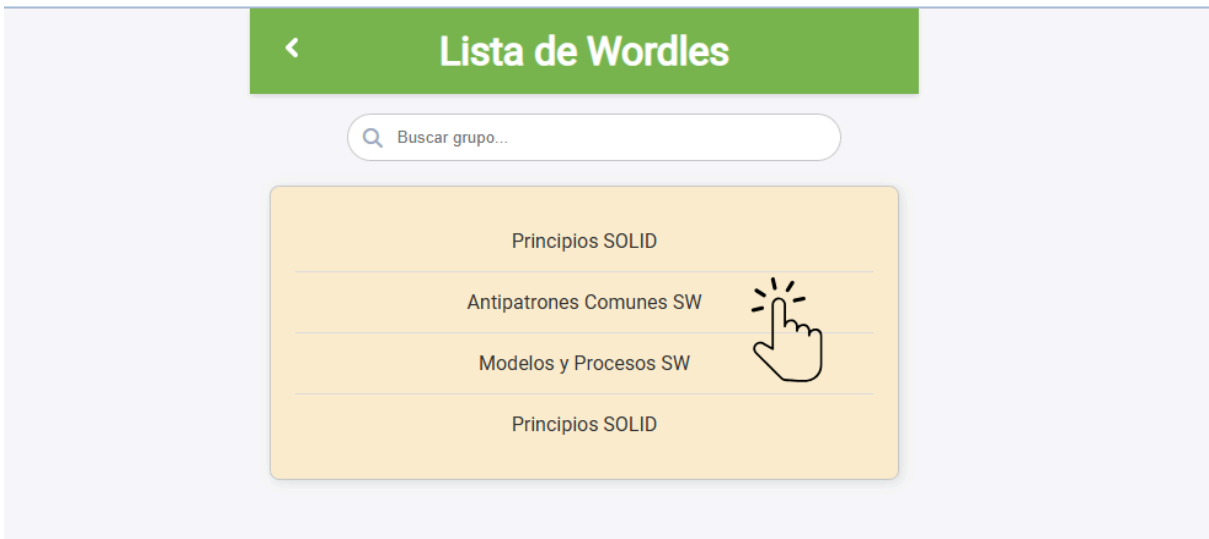
Esta información permite al profesorado analizar el rendimiento de forma rápida y detectar posibles dificultades de aprendizaje en su alumnado.

### 5.1.6. Jugar *Wordle*

El alumnado puede acceder al listado de *Wordles* disponibles desde su panel principal. Tras seleccionar uno, comienza el juego.



**Figura 5.32:** Camino hacia Lista de *Wordles* desde *Dashboard* de Alumno



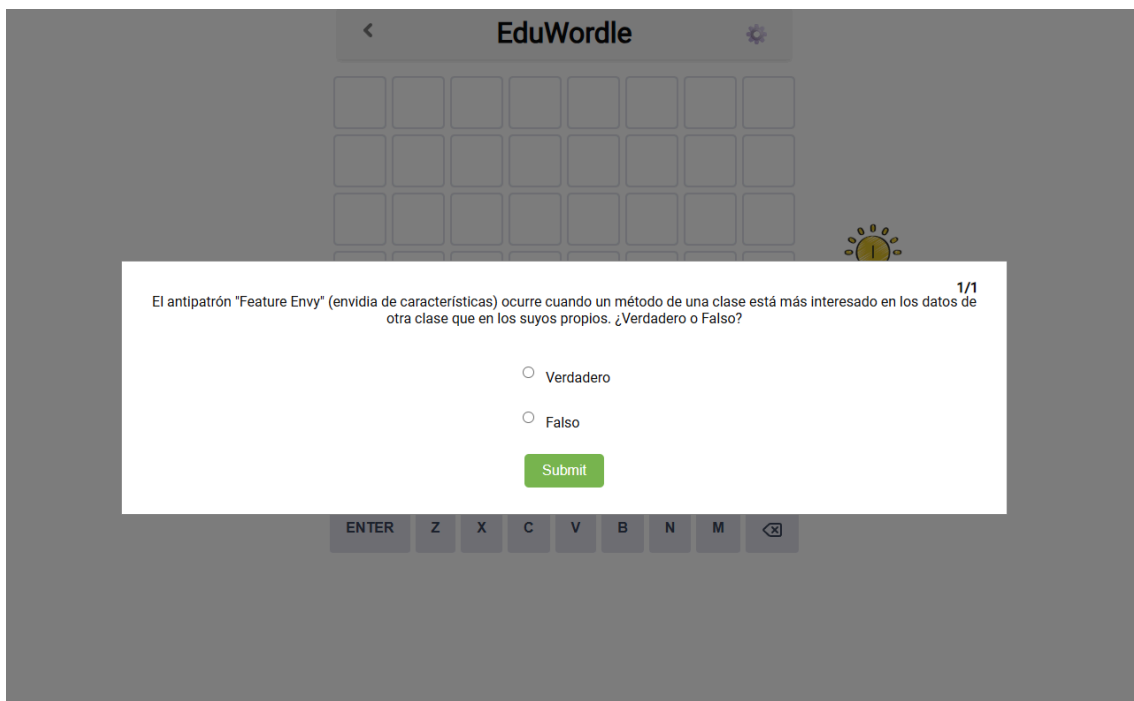
**Figura 5.33:** Lista de *Wordles* de Alumno



**Figura 5.34:** Pantalla de Juego

El juego consiste en adivinar la palabra en seis intentos, para descubrir la palabra el docente ha podido incluir una pista.

Si se pide una pista, habrá que pulsar sobre el icono de la bombilla y el sistema lanzará una pregunta aleatoria sobre el temario, si esta se acierta, devolverá la pista.



**Figura 5.35:** *Popup* de Pregunta simple



Figura 5.36: Mensaje de éxito y pista

Cuando introduces un intento, se mostrará una pregunta si es de nivel bajo o varias preguntas en función de las letras acertadas si el *Wordle* es de nivel alto.

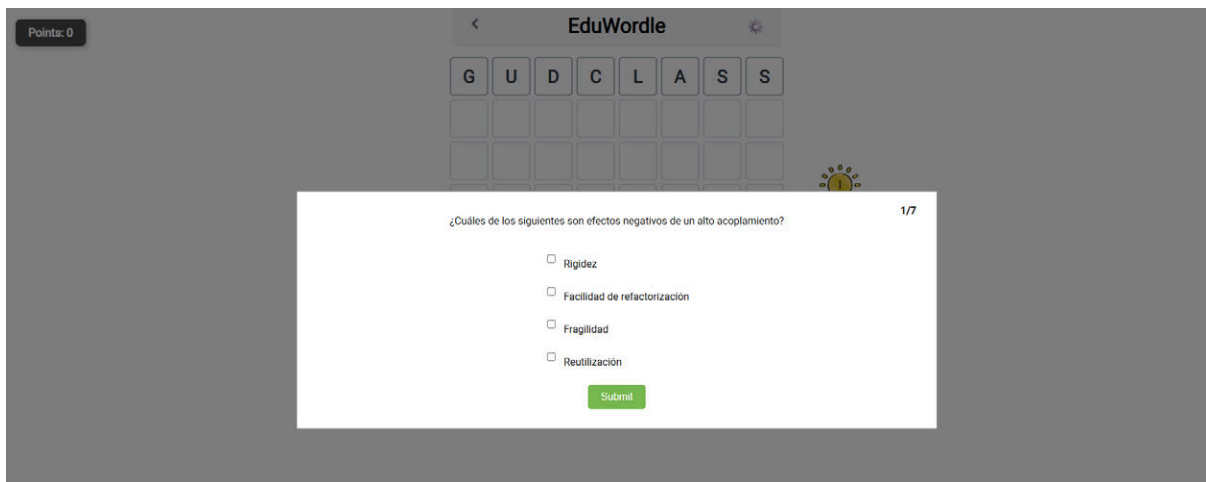
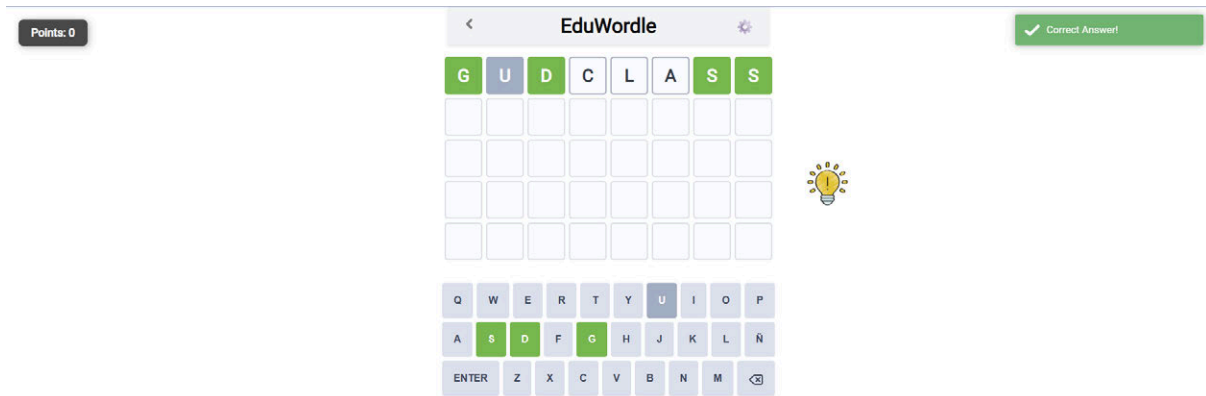


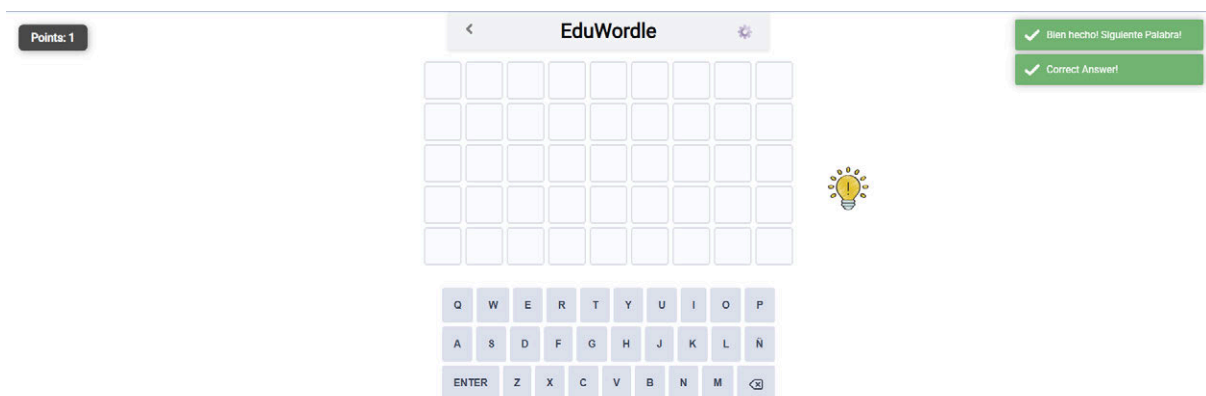
Figura 5.37: Popup de varias preguntas

El sistema ofrece retroalimentación visual en cada intento: verde si la letra está bien colocada, amarillo si está en otra posición, y gris si no pertenece a la palabra.



**Figura 5.38:** Coloración de letras acertadas

Una vez que sepamos cual es la palabra correcta y la introducimos, aparecerán preguntas como hemos indicado antes, pero después de contestarlas, la palabra se mostrará verde y a continuación se indicará que es correcta y se sumará un punto al marcador, así hasta resolver todas las palabras de ese *Wordle*.



**Figura 5.39:** Mensaje de éxito al acertar una palabra

## 5.2. Validación : pruebas de aceptación

Para asegurar el cumplimiento de los requisitos funcionales y no funcionales del sistema, se ha llevado a cabo un proceso de validación. En este apartado se presenta la preparación de las pruebas, un vídeo demostrativo de cómo funciona la web y escenarios de demostración para verificar todos los escenarios posibles.

## 5.2.1. Introducción de Datos de Semilla

Para facilitar las pruebas, validación y demostración de las funcionalidades de EduWordle, se ha implementado un proceso de carga inicial de datos de semilla en la base de datos. Estos datos, diseñados específicamente para cubrir los casos de uso definidos y simular un entorno de uso realista, permiten presentar la plataforma con información pre-existente sin necesidad de una configuración manual extensa durante las demostraciones.

La semilla de datos incluye una variedad de usuarios de ambos roles alumnos y profesores, grupos que agrupan a los alumnos, *Wordles* temáticos con sus respectivas palabras y preguntas y un conjunto de resultados de juego simulados. Esto asegura que todas las características clave del sistema como la gestión de usuarios, la asignación de *Wordles* a grupos, el seguimiento de resultados y los *rankings*, puedan ser demostradas de manera efectiva y coherente.

Para una descripción detallada de los datos de semilla cargados en el sistema incluyendo credenciales de acceso y configuración de *Wordles*, se puede referir al [Anexo 4: Datos de Semilla para la demostración.](#)

## 5.2.2. Escenarios de demostración

Para ilustrar el funcionamiento de EduWordle, se han preparado dos escenarios diferenciados que muestran la experiencia de uso desde las perspectivas del profesorado y del alumnado. A través de esta demo se pretende visualizar el flujo de trabajo habitual y las funcionalidades principales de la plataforma.

### Escenario 1: Profesor

En este primer caso, se muestra cómo un docente accede a la plataforma y realiza las acciones esenciales para gestionar su grupo de estudiantes y sus actividades. Entre las funcionalidades que se visualizan se incluyen:

- Inicio de sesión con perfil de profesor.
- Creación de un nuevo grupo.

- Creación de un nuevo *Wordle*, incluyendo preguntas, palabras con pista y nivel de dificultad.
- Asignación del *Wordle* al grupo previamente creado.
- Visualización de los resultados obtenidos por el alumnado en cada *Wordle*, tanto de forma individual como en el *ranking* del grupo.

Este recorrido representa las tareas habituales que realiza un docente en el uso didáctico de la plataforma.

En la siguiente figura se puede acceder a un video donde se visualiza el escenario definido anteriormente:



Figura 5.40: Video de ejemplo de Eduwordle Profesor. [Enlace directo](#)

## Escenario 2: Estudiante

A continuación, se muestra la experiencia desde el punto de vista del estudiante. Este usuario accede a su cuenta y realiza las siguientes acciones:

- Inicio de sesión con perfil de alumno/a.
- Visualización de los grupos a los que ha sido asignados por su profesor/a.
- Visualización de los *Wordles* disponibles.
- Selección y resolución de un *Wordle*.

Este segundo escenario permite observar cómo el estudiante interactúa con el contenido *gamificado* y cómo recibe la retroalimentación que facilita su aprendizaje.

En la siguiente figura se puede acceder a un video donde se visualiza el escenario definido anteriormente:



**Figura 5.41:** Video de ejemplo de Eduwordle Estudiante. [Enlace directo](#)

## 5.3. Despliegue

El despliegue se realizó en un servidor propiedad de la escuela y para ello, fue fundamental la colaboración con el personal técnico responsable para llevar a cabo los trámites correspondientes. Concretamente, se le proporcionó acceso al código fuente del proyecto y se solicitó la configuración de las variables de entorno (.env) necesarias en el servidor de producción. Este ajuste fue crucial, por ejemplo, para asegurar que las URL generadas por la aplicación (como las incluidas en los correos electrónicos) apuntaran correctamente a la instancia desplegada.

### 5.3.1. Recursos del Proyecto

En esta sección, se proporcionan los enlaces directos al producto final de EduWordle y al repositorio de código fuente del proyecto. Estos recursos permiten el acceso y la exploración tanto de la aplicación web en funcionamiento como de la implementación técnica y su historial de desarrollo.

#### Acceso Web

La siguiente figura muestra un QR de acceso a la web de EduWordle.

Invitamos al lector a escanear el QR y probar las funcionalidades del sistema. Para iniciar sesión, se puede utilizar los datos de usuario especificados en la sección de [“Introducción de Datos de Semilla”](#).

Debido a que otros usuarios podrían haber accedido a estos perfiles y modificado sus contraseñas como parte de sus pruebas, es posible que algunas de las credenciales originales ya no sean válidas.



**Figura 5.42:** Acceso a Eduwordle. [Enlace directo](#)

## Acceso Repositorio

Este código QR proporciona acceso directo al repositorio donde se encuentra todo el código fuente del proyecto. En el archivo README.md encontrará instrucciones detalladas para la inicialización e implementación local de la aplicación.



**Figura 5.43:** Acceso al repositorio en GitHub. [Enlace directo](#)

## 6. Conclusiones

Este Trabajo de Fin de Grado ha culminado con el desarrollo de *EduWordle*, una aplicación web educativa que cumple los objetivos propuestos: fomentar el aprendizaje lúdico en el aula y motivar al alumnado mediante una herramienta digital accesible, intuitiva y funcional. A lo largo del proyecto se han abordado diversos aspectos pedagógicos y técnicos, lo que ha permitido crear una solución potente y alineada con las necesidades del entorno educativo actual.

La principal aportación de *EduWordle* reside en su capacidad para transformar una dinámica lúdica conocida en una experiencia de aprendizaje significativa. Gracias a la personalización de contenidos, el uso de preguntas vinculadas al temario y el sistema de puntuación, la plataforma convierte cada partida en una oportunidad para reforzar conocimientos de forma motivadora. Esta integración entre juego y contenido curricular ofrece al profesorado una herramienta didáctica eficaz y flexible.

*EduWordle* ha sido concebida como una herramienta polivalente, pensada para docentes de cualquier nivel educativo, desde primaria hasta estudios universitarios, y de cualquier área de conocimiento. Su interfaz sencilla y su sistema de carga de datos mediante archivos CSV hacen posible su uso por parte de profesores sin formación técnica avanzada.

La plataforma puede integrarse en el aula como complemento a las metodologías tradicionales. Recomendamos su uso en sesiones de introducción de nuevos contenidos, como actividad de repaso antes de una evaluación o como tarea gamificada que fomente el estudio autónomo. Además, la creación de competiciones sanas entre grupos puede reforzar la motivación del alumnado y fomentar el trabajo colaborativo, adaptándose a distintos ritmos de aprendizaje.

Desde el punto de vista técnico, el desarrollo de *EduWordle* ha supuesto un desafío enriquecedor. La aplicación se ha construido sobre una arquitectura sólida, modular y escalable, siguiendo una metodología en cascada con prototipado evolutivo. Se han utilizado tecnologías como Node.js, Express, MySQL, Sequelize y Docker, además de herramientas de apoyo como Swagger o Nodemailer. Todo ello ha

contribuido a la implementación de una solución funcional, mantenible y preparada para su evolución futura.

Este proyecto nos ha permitido aplicar de forma práctica los conocimientos adquiridos durante la carrera, enfrentándonos a problemas reales de diseño, desarrollo e implementación. Hemos reforzado nuestras capacidades de planificación, trabajo en equipo y toma de decisiones técnicas, y hemos ganado experiencia con herramientas ampliamente utilizadas en el sector. Ha sido una experiencia completa que nos ha permitido entender de forma integral el ciclo de vida del desarrollo de software.

En definitiva, *EduWordle* puede ser mucho más que un proyecto académico: es una propuesta con potencial real para mejorar el proceso de enseñanza-aprendizaje a través de la gamificación. Gracias a su diseño flexible y su arquitectura, la aplicación ofrece una base sólida para futuras mejoras. Creemos firmemente que esta herramienta puede seguir creciendo y adaptándose a las necesidades de la educación del presente y del futuro.

## 7. Aspectos sociales, ambientales, éticos y profesionales.

---

El desarrollo de EduWordle responde a un compromiso ético y profesional con la mejora del sistema educativo mediante el uso responsable e innovador de la tecnología mediante una herramienta educativa. Esta propuesta, se alinea con el Objetivo de Desarrollo Sostenible número 4: Educación de calidad, cuyo objetivo es garantizar una educación de calidad e inclusiva, además de asegurar el acceso a oportunidades de aprendizaje para todos.

En este sentido, la aplicación busca no solo motivar al alumnado mediante el juego, sino facilitar el trabajo del profesorado, ofreciendo una solución sencilla, eficaz y aplicable en cualquier grupo, independientemente del nivel académico del mismo. Además, su diseño accesible garantiza que el profesorado pueda implementarla en el aula independientemente de su nivel de competencia digital, reduciendo así las desigualdades en el acceso a herramientas digitales innovadoras.

Desde el punto de vista social, EduWordle apuesta por una educación participativa y centrada en el estudiante, que respete los distintos ritmos y estilos de aprendizaje. La posibilidad de personalizar los Wordles en función del contenido curricular y el nivel del grupo favorece la inclusión de todo el alumnado, promoviendo una enseñanza más justa y cercana a la realidad. Además, la plataforma incorpora un modo adaptado para personas con daltonismo, que permite distinguir los colores utilizados en el juego mediante tonos alternativos, contribuyendo así a una experiencia accesible y equitativa para todos los usuarios.

En cuanto al plano ético y profesional, el proyecto ha sido desarrollado siguiendo buenas prácticas de programación, con especial atención en la seguridad del sistema, y la claridad en la documentación técnica. Esto garantiza la responsabilidad profesional en el desarrollo de *software* y su viabilidad a largo plazo.

Aunque su impacto ambiental es limitado, debido a su carácter digital, EduWordle contribuye de forma indirecta con el Objetivo de Desarrollo Sostenible número 11: Producción y consumo responsables, evitando el consumo de materiales físicos y fomentando prácticas de enseñanza más eficientes y respetuosas con el medio ambiente.

Por todo ello, EduWordle representa una aportación ética, social y profesionalmente responsable a la transformación educativa, demostrando cómo la tecnología puede ser utilizada para mejorar la calidad del aprendizaje y favorecer la inclusión, en coherencia con los principios del desarrollo sostenible.

## 8. Líneas Futuras

---

El desarrollo de EduWordle actual, sienta las bases para una plataforma educativa con gran potencial de crecimiento. Si bien se han cumplido gran cantidad de los objetivos iniciales y se ha entregado un producto funcional completo, existen diversas vías para su mejora y evolución. Las siguientes líneas de trabajo representan posibles nuevas funcionalidades que mejorarían significativamente la experiencia de usuario y la utilidad pedagógica del sistema:

1. Exportación de Clasificaciones a CSV: Permitir a los profesores descargar los resultados de alumnos tanto para grupos como para *Wordles* en formato CSV para conformar una métrica de clasificación del desempeño del alumno frente a la materia.
2. Mejora del filtrado de la visualización de *Wordles* y Grupos para permitir una búsqueda más específica.
3. Visualización de *rankings* por parte del alumno: Permitir a los alumnos acceder a las puntuaciones personales para comprobar su propio desempeño.
4. Nuevas modalidades de juego: Introducir variaciones en la dinámica de juego de *Wordle* para mantener el interés y adaptarse a diferentes niveles de dificultad o estrategias de aprendizaje. Ejemplos de ellos serían:
  - a. Cronómetro.
  - b. Dinámica de vidas: Un número limitado de errores antes de perder.
  - c. Número de intentos por palabra variable : Configurable por el profesor.
  - d. Pregunta a cambio de una letra: Un sistema de ayuda basado en la resolución de preguntas adicionales
5. Compatibilidad con otro tipo de preguntas: Ampliar los formatos de preguntas para incluir opciones más variadas, por ejemplo preguntas que muestren imágenes, preguntas de rellenar campos de emparejamiento...
6. Compatibilidad con ficheros de subida de datos con otros formatos, por ejemplo *moodleXML* para las preguntas.

7. Configuración avanzada de *Wordles*: Ofrecer la posibilidad de que el profesor configure el *Wordle* para ser temporal, por ejemplo semanal y así dinamizar la disposición de los mismos.
8. Accesibilidad y Personalización: Implementar la capacidad de que la interfaz presente varios idiomas, ampliando así la audiencia a la que iría dirigido el sistema. Otro ejemplo de personalización sería ofrecer la opción de cambiar entre tema claro y oscuro o incluso ajustarse en función del tema del sistema.
9. Fomento de la Comunidad y Competitividad: Desarrollar un sistema de competencia entre diferentes grupos para promover el trabajo y mejora del equipo para alcanzar posiciones mejores.
10. Adaptar la versión móvil visionada en un inicio.

Estas líneas futuras no solo consolidarían EduWordle como una herramienta educativa más versátil, sino que también podrían demostrar el impacto potencial del proyecto en el ámbito educativo.

Desde una perspectiva de mejora interna del sistema se propone una optimización de la arquitectura del código, lo que implicaría una revisión y depuración de la distribución de carpetas y la estructura de herencia de clases.

Un ejemplo concreto de esta mejora sería la centralización de los estilos del *frontend* para evitar redundancias y facilitar la mantenibilidad. Además, sería fundamental implementar una política de retención de datos con lógica de eliminación automática de la información de los usuarios tras un periodo de tiempo predefinido, asegurando así el cumplimiento con la Ley Orgánica de Protección de Datos (LOPD) y el Reglamento General de Protección de Datos (RGPD).

Finalmente, el código de EduWordle se ha liberado como un proyecto de código abierto, accesible en GitHub. De esta forma, el software está completamente preparado para que futuras generaciones de estudiantes no puedan solo beneficiarse de su estructura y funcionalidad para el aprendizaje, sino que también continuarán activamente su desarrollo y expansión.

## 9. Bibliografía

---

- [1] **Axel Abraham V.**, «*El impacto de la tecnología en la sociedad actual*», LinkedIn, 2022. Disponible en: <https://es.linkedin.com/pulse/el-impacto-de-la-tecnolog%C3%ADa-en-sociedad-actual-axel-abraham-v>
- [2] **Elizabeth, Garcés Suárez, E. Suárez, y Orly David Alcívar Fajardo.**, «*Las Tecnologías de la Información en el cambio de la Educación Superior en el siglo XXI: reflexiones para la práctica*». Revista Universidad y Sociedad 2016. Disponible en: [http://scielo.sld.cu/scielo.php?pid=S2218-36202016000400023&script=sci\\_arttext&tlng=en](http://scielo.sld.cu/scielo.php?pid=S2218-36202016000400023&script=sci_arttext&tlng=en)
- [3] **Educativa**, «*Gamificación: el aprendizaje divertido*», Educativa.com, s.f. Disponible en: <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>
- [4] **Ministerio de Educación y Formación Profesional de España**, «*Gamificación en la enseñanza*», Madrid, 2022. Disponible en: <https://www.educacionfpydeportes.gob.es/biblioteca-central/blog/2022/gamificacion-en-la-ensenanza.html>
- [5] **Educación 3.0**, «*Wordle, juego de palabras*», s.f. Disponible en: <https://www.educaciontrespuntocero.com/noticias/wordle-juego-de-palabras>
- [6] **Sarabia-Guevara, D. A., & Bowen-Mendoza, L. E.** «*Uso de la gamificación en el proceso de enseñanza aprendizaje en carreras de ingeniería: revisión sistemática.*», EPISTEME KOINONIA, 2023. Disponible en: [https://ve.scielo.org/scielo.php?script=sci\\_arttext&pid=S2665-0282202300](https://ve.scielo.org/scielo.php?script=sci_arttext&pid=S2665-0282202300)

[0200020](#)

- [7] **Sanchez-Pacheco, Carlos, y Unidad Educativa Fiscal Réplica Guayaquil**, «ELEMENTOS DE LA GAMIFICACION Y SUS IMPACTOS EN LA ENSEÑANZA Y EL APRENDIZAJE», Identidad Bolivariana , 2019 Disponible en:  
<https://identidadbolivariana.itb.edu.ec/index.php/identidadbolivariana/article/view/56>
- [8] **Perazo, Cintia**, «La historia del Wordle, el juego que nació como un regalo y se convirtió en un fenómeno global», La Nación (Buenos Aires), 27 mayo 2024. Disponible en:  
<https://www.lanacion.com.ar/juegos/historias/la-historia-del-wordle-nid27052024/>
- [9] **INTEF**, «Kahoot: ¿evaluamos o jugamos?», s.f. Disponible en:  
[https://intef.es/observatorio\\_tecno/kahoot-evaluamos-o-jugamos/](https://intef.es/observatorio_tecno/kahoot-evaluamos-o-jugamos/)
- [10] **INTEF**, «Quizizz», s. f. Disponible en:  
[https://intef.es/observatorio\\_tecno/quizizz/](https://intef.es/observatorio_tecno/quizizz/)
- [11] **INTEF**, «Wordwall: jugando en el aula», s.f. Disponible en:  
[https://intef.es/observatorio\\_tecno/wordwall-jugando-en-el-aula/](https://intef.es/observatorio_tecno/wordwall-jugando-en-el-aula/)
- [12] **Oracle Corporation**, «10 razones para elegir MySQL para las aplicaciones web de la próxima generación», Santa Clara, s.f. Disponible en:  
<https://www.mysql.com/why-mysql/white-papers/10-razones-para-elegir-mysql-para-las-aplicaciones-web-de-la-proxima-generacion/>
- [13] **Red Hat**, «¿Qué es Docker?», s.f. Disponible en:  
<https://www.redhat.com/es/topics/containers/what-is-docker>
- [14] **GitHub**, «About GitHub and Git», s.f. Disponible en:  
<https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>

- [15] **Microsoft**, «*Visual Studio Code*», s. f. Disponible en: <https://code.visualstudio.com>
- [16] **Nodemailer Team**, «*Nodemailer*», s. f. Disponible en: <https://nodemailer.com>
- [17] **Swagger**, «*Swagger*», s. f. Disponible en: <https://swagger.io>

# 10. Anexos

---

## Anexo 1

### *“Entrevista para la Definición de Requisitos”*

Este anexo contiene el Acta completa de la entrevista realizada con el tutor del proyecto, Daniel López Fernández, el 23/10/2024. Esta reunión tuvo como objetivo la elicitación de los requisitos preliminares y la definición del alcance inicial del proyecto EduWordle. La información aquí contenida sirvió como punto de partida para las primeras fases de análisis y diseño, aunque, como se menciona en el apartado de [Extracción de Requisitos](#), algunos de estos requisitos fueron modificados o suprimidos en fases posteriores del desarrollo debido a la evolución del proyecto.

El documento de la entrevista se puede visualizar en el siguiente QR



Figura A.1: Documento de Entrevista inicial. [Enlace directo](#)

## Anexo 2

### “Acta Reunión Inicial”

---

#### **Información General de la Reunión**

- **Fecha:** 06/03/25
- **Hora de Inicio:** 18:20
- **Hora de Finalización:** 19:30
- **Lugar/Plataforma:** Teams
- **Objetivo de la Reunión:** Alinear los requisitos de dos proyectos de TFG independientes para fusionarlos en EduWordle y definir un alcance inicial.

#### **Asistentes**

- **Tutor/Supervisor:**
  - Daniel López Fernández (como Cliente)
- **Miembros del Equipo:**
  - Elena Campanero Belda
  - Miriam Sánchez Saiz

#### **Puntos de la Agenda y Discusión**

##### **1. Presentación y Situación Inicial de los Proyectos**

- Se ha expuesto que inicialmente se habían asignado dos TFGs con enfoques similares pero requisitos dispares. En este momento ya se habían generado los siguientes elementos para el proyecto inicial de Elena :
  - Elicitación de Requisitos
  - Esquemas de casos de Uso
  - Diagrama Entidad-Relación y paso a tablas
  - Interfaces de Inicio, Login y Juego
  - Dinámica de Juego (tanto *frontend* como unión parcial al live server provisional) con dinámica de pistas y compatibilidad con preguntas de *single* y *multiple selection*
  - Prototipo visual para versión móvil

##### **2. Objetivo de Unificación y Visión Común de EduWordle**

- Se ha acordado seguir el proyecto con la mayoría de los requisitos definidos en la Entrevista entre Daniel y Elena. La visión se ha centrado en una herramienta *gamificada* para profesores y alumnos, que permita la creación de contenido educativo adaptado y un seguimiento del progreso.

### 3. Requisitos Consolidados y MVP

- Se han analizado los requisitos y se ha decidido abordar la gestión de usuarios (registro, roles), creación de contenido (*Wordles*, palabras, preguntas), dinámica de juego, gestión de grupos y seguimiento de resultados.
- Se priorizaron las funcionalidades mínimas viables para una primera versión y se ha definido el objetivo para una primera entrega con los siguientes elementos
  - Desarrollo de un prototipo para:
    - Inicio de Sesión
    - Registro (función de generación de contraseñas y creación de usuario para desarrollo y para los estudiantes al añadirlos el profesor a un grupo)
    - Profesor:
      - Pantalla de creación de *Wordle*
      - Pantalla de creación de Grupo
      - Asociación de Alumno a Grupo
      - Asociación de Grupo a *Wordle*
    - Alumno:
      - Ver lista de *Wordles*
      - Jugar un *Wordle*
      - Dinámica de puntos dentro de una partida (*Wordles* con más de una palabra)
  - Documentación:
    - Estudiar una posible arquitectura de la solución
  - Características de cambio sobre las versiones iniciales:
    - Desaparece la figura del Admin, la adición de profesores se hará directamente sobre el sistema
    - Solo se guarda la partida con mayor puntuación para cada tupla de alumno, *Wordle*
    - Grupos sin fecha de eliminación automática

- En la partida, el número de intentos para adivinar una palabra deja de ser variable y pasa a estar fijo en 5 intentos
- Si se sale a media partida de un juego, se pierden los datos
- La configuración de un *Wordle* se limita
- Hay que añadir una manera de búsqueda en las listas de *Wordle* y Grupo por nombre
- Necesidad de añadir una capa de Controladores

#### 4. Propuesta de solución

- Se propuso una solución con Node.js en el *backend*, HTML, JavaScript y CSS en el *frontend* para seguir con la parte del proyecto anterior ya desarrollada y MySQL como base de datos . El tutor ha dado su visto bueno a la pila tecnológica propuesta y solicitó una solución orientada a una futura *dockerización*.
- Al ser la primera entrega un prototipo, se ha decidido usar la librería de live-server en VSCode como mockup del futuro *backend*.

#### 5. Planificación Inicial e Hitos

- Se ha acordado una fase inicial de diseño detallado de la base de datos, desarrollo del *frontend* y adaptación al live-server para facilitar la posterior unión al *backend*.

#### 6. Próximos Pasos

- Elaborar el Product Backlog inicial detallado en Jira
- Diseñar el esquema de la base de datos
- Comenzar con el desarrollo de la aplicación

### **Conclusiones y Acuerdos Principales**

- Unificación exitosa de los proyectos
- Visión clara de EduWordle como plataforma educativa *gamificada*
- Definición de un alcance inicial y funcionalidades prioritarias
- Aprobación de la pila tecnológica propuesta
- Definición de primeros pasos

## Anexo 3

### “Actas de Reuniones”

#### Acta Reunión de planificación iteración 1

---

##### 1. Información General

- **Fecha:** 13/03/25
- **Hora de Inicio:** 17:15
- **Hora de Finalización:** 18:15
- **Lugar/Plataforma:** Discord
- **Asistentes:** Miriam Sánchez Saiz, Elena Campanero Belda
- **Objetivo:** *Completar el desarrollo de todas las pantallas de frontend, generar las tablas de la BD en MySQL y conectarlo para el login con live-server*

##### 2. Planificación Detallada de Tareas

- **Frontend** (Responsable: Elena)
  - Desarrollo de página de “List”, tanto para Wordles como para Grupos
  - Desarrollo de *Dashboard* de Profesor y Alumno
  - Desarrollo de pantallas de Visualización, tanto de *Wordle* como de Grupo
  - Desarrollo de pantalla de Edición, tanto de *Wordle* como de Grupo
  - Adaptación de Pantalla de Juego para puntuación
- **Backend** (Responsable: Miriam)
  - Creación de Tablas de Preguntas y Palabras según diagrama entidad-relación
  - Lógica de Inicio de Sesión
  - Generar archivo de conexión de MySQL con live-server/nuestro sistema

##### 3. Identificación de Posibles Bloqueos/Dependencias

- Dependencia entre la implementación del *backend* de registro y la conexión desde el *frontend*.

#### 4. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta Iteración será de **2 semanas**.

## Acta Reunión de planificación iteración 2

---

### 1. Retrospectiva

#### i. Revisión

Logros	Obstáculos	Sorpresas
Trabajar en paralelo	Tiempo disponible	La implementación de filtros de búsqueda no era viable sin la conexión a la BD así que se dejó para más adelante
Seguimiento de objetivos periódico	Dependencias del <i>frontend</i> con la BD (a la hora de actualizar la pestaña de juego, se necesitaba la conexión de BD para obtener las preguntas y palabras y solo la tenía Miriam)	
	Al programar el login, hubo una falta de comunicación ya que no se guardaba la información de sesión en LocalStorage y se necesitaba para funcionalidades de la aplicación	

**Tabla 10:** Tabla LOS Retrospectiva 1

ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing
Tener una visión más general del proyecto para tener perspectiva de nuestra velocidad	Retrasar las fechas de fin de Iteración	Ir haciendo tests para generar elementos bug para solucionar en la siguiente Iteración
Medir mejor nuestra capacidad para no sobrecargarnos	Abarcar mucho	

**Tabla 11:** Tabla SSK Retrospectiva 1

iii. Elementos conseguidos

1. Desarrollar bases sólidas de todas las pantallas HTML
2. Generar tablas de Preguntas y Palabras y conectar correctamente la BD
3. Lógica de Login terminada y comprobada con archivos locales de usuarios registrados

iv. Elementos aplazados

1. Guardar los datos de sesión en Local Storage
2. Modificar la dinámica de partida para poder jugar más de una palabra
3. Añadir mensajes de éxito y error al trabajar con la BD
4. Añadir doble confirmación a la hora de eliminar un *Wordle* o un Grupo

## 2. Información General

- **Fecha:** 26/03/25
- **Hora de Inicio:** 10:00
- **Hora de Finalización:** 11:00
- **Lugar/Plataforma:** Discord
- **Asistentes:** Miriam Sánchez Saiz y Elena Campanero Belda

- **Objetivo:** Definición de Persistencia formal, mejorar htmls, creación de modelos de datos, tomar decisiones sobre arquitectura y herramientas

### 3. Planificación Detallada de Tareas

- Persistencia
  - *Elemento aplazado:* Guardar sesión en Local Storage (Responsable: Miriam)
  - Creación de archivos de Persistencia temporal (mockup hasta la creación del *backend* final) (Responsable: Miriam)
  - Conexión HTML con persistencia y mensajes de éxito (Responsable: Miriam)
- Adicciones HTML (Responsable: Elena)
  - *Elemento aplazado:* Pop Ups de doble confirmación de borrado de Wordle o Grupo
  - *Elemento aplazado:* Añadir la dinámica de Juego de más de una palabra
  - Pop Ups para pestañas de Edición (añadir palabras, preguntas, grupos, alumnos y wordles)
  - Revisión de funcionalidades ya implementadas
  - Añadir botones de eliminar elementos dentro de *groupEditor* y *wordleEditor*
- Utils
  - Crear modelos de datos para facilitar el envío y recepción de información posteriormente al servidor (Responsable: Elena)
- Documentación (Responsable: Ambas)
  - Definición de Arquitectura del *backend*
  - Decisión sobre mantener la herramienta de Jira

### 4. Identificación de Posibles Bloqueos/Dependencias

- Se mantiene la dependencia de la BD pero ahora es con todas las tablas, no solo con las de Preguntas y Palabras

### 5. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta Iteración será de **2 semanas**.

## Acta Reunión de planificación iteración 3

---

### 1. Retrospectiva

#### i. Revisión

Logros	Obstáculos	Sorpresas
Finalización del primer hito	Disponibilidad	Una situación ajena al proyecto provocó un retraso en una de las partes y la recuperación del mismo conllevó el retraso de fechas pre establecidas

*Tabla 12: Tabla LOS Retrospectiva 2*

#### ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing
Cambiar la manera de dividir las tareas	Tomar decisiones individuales	Seguir el seguimiento del desarrollo con revisiones periódicas

*Tabla 13: Tabla SSK Retrospectiva 2*

#### iii. Elementos conseguidos

1. Todos los cambios propuestos del HTML
2. Creación de Utils
3. Archivos de Persistencia (parcial)
4. Conexiones BD y mensajes de éxito
5. **FIN DEL PRIMER HITO (frontend)**

- iv. Elementos aplazados
  - 1. Guardar sesión en Local Storage
  - 2. Persistencia de Grupos, Wordles y Alumnos
  - 3. Generar documentación de arquitectura del *backend*

## 2. Información General

- **Fecha:** 10/04/25
- **Hora de Inicio:** 11:30
- **Hora de Finalización:** 12:30
- **Lugar/Plataforma:** Discord
- **Asistentes:** Miriam Sánchez Saiz y Elena Campanero Belda
- **Objetivo:** Desarrollo de últimos cambios en el *frontend* antes de la entrega formal

## 3. Planificación Detallada de Tareas

- Detalles *frontend* (Responsable: Elena)
  - Añadir al HTML el display de puntos de partida para mostrar al alumno sus puntos en la partida actual
- Detalles base de datos (Responsable: Miriam)
  - Generar los archivos de Persistencia restantes

## 4. Identificación de Posibles Bloqueos/Dependencias

- Las tareas se han podido dividir a lo largo del tiempo para no producir dependencias

## 5. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta iteración será de **2 semanas**.

## Acta Reunión de planificación iteración 4

---

### 1. Retrospectiva

#### i. Revisión

Logros	Obstáculos	Sorpresas
	Periodo Vacacional (Semana Santa)	

Tabla 14: Tabla LOS Retrospectiva 3

#### ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing
Mejorar la comunicación	Abarcar mucho	Mantener las revisiones periódicas
	No mantener las fechas de entrega	

Tabla 15: Tabla SSK Retrospectiva 3

#### iii. Elementos conseguidos

1. Todos los cambios propuestos del HTML
2. Parte de la arquitectura de *backend*

#### iv. Elementos aplazados

1. Guardar sesión en LocalStorage: por falta, se mantendrá por parámetros hasta implementar el *backend*
2. Persistencia de Grupos, Wordles y Alumnos: se terminó por omitir ya que la siguiente fase era ya el desarrollo del *backend* real

### 2. Información General

- Fecha: 25/04/25
- Hora de Inicio: 11:07

- **Hora de Finalización:** 13:10
- **Lugar/Plataforma:** Discord
- **Asistentes:** Miriam Sánchez Saiz y Elena Campanero Belda
- **Objetivo:** Planificación del siguiente hito y del desarrollo del *backend*

### 3. Planificación Detallada de Tareas

- *Backend* (Responsable: Elena)
  - Desarrollo de las capas: *controllers, middlewares, models, routes, services*
  - Desarrollo de archivos de pruebas unitarias y funcionales
  - Creación de una semilla para las pruebas
  - Creación de plantilla de correos electrónicos para nuevos usuarios
- Documentación (Responsable: Ambas)
  - Añadir los avances a Jira y definir las nuevas tasks

### 4. Identificación de Posibles Bloqueos/Dependencias

- En esta etapa no hubo dependencias

### 5. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta iteración será de **1 mes**.

## Acta Reunión de planificación iteración 5

---

### 1. Retrospectiva

#### i. Revisión

Logros	Obstáculos	Sorpresas
Tener una base del	Periodo Vacacional	

segundo hito a tiempo		
-----------------------	--	--

**Tabla 16:** Tabla LOS Retrospectiva 4

ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing
Plasmar todas las tareas y distribuirlas eficientemente		Mantener las revisiones periódicas

**Tabla 17:** Tabla SSK Retrospectiva 4

iii. Elementos conseguidos

1. Desarrollo de una base semi completada para el *backend*
2. Cumplidos todos los objetivos respecto al *backend*
3. Añadida la creación de un archivo swagger (requisito extra no contemplado en la Iteración anterior)

iv. Elementos aplazados

1. Terminar de detallar e investigar el *emailService*
2. Seguir trabajando en los tests
3. Tomar decisión sobre la capa donde realizar el *hash* de contraseñas

## 2. Información General

- **Fecha:** 06/05/25
- **Hora de Inicio:** 18:00
- **Hora de Finalización:** 19:00
- **Lugar/Plataforma:** Teams
- **Asistentes:** Daniel López Fernández, Miriam Sánchez Saiz y Elena Campanero Belda
- **Objetivo:** Adición de nuevos requisitos de EduWordle y refinamiento de *backend*

### 3. Planificación Detallada de Tareas

- *Elemento aplazado*: Terminar de detallar e investigar el *emailService* (Responsable: Elena)
- *Elemento aplazado*: Seguir trabajando en los tests (Responsable: Elena)
- *Elemento aplazado*: Tomar decisión sobre la capa donde realizar el *hash* de contraseñas (Responsable: Elena)
- Nuevos Requisitos:
  - Que la aplicación obligue a cambiar la contraseña al usuario si es la primera vez que inicia sesión
    - Desarrollo funcionalidad en *frontend* (Responsable: Miriam)
    - Desarrollo funcionalidad en *backend*: generación de contraseñas iniciales (Responsable: Elena)
  - Añadir “dificultad” a la configuración de *Wordle* para permitir ajustar la cantidad de preguntas que aparecen por intento
    - Desarrollo funcionalidad en *frontend* (Responsable: Miriam)
    - Desarrollo modificaciones en *backend* (Responsable: Elena)
  - Permitir subir archivos CSV de carga masiva de datos (Responsable: Miriam)
  - Clasificaciones por *Wordle* y por Grupo
    - Desarrollo de *frontend* de la nueva página (Responsable: Miriam)
    - Desarrollo de las funciones de apoyo en *backend* (Responsable: Elena)

### 4. Identificación de Posibles Bloqueos/Dependencias

- En esta etapa hubo una dependencia a la hora de trabajar con los *Wordles* ya que se necesitaba la propagación del cambio en todas las capas para que funcionaran correctamente

### 5. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta iteración será de **2 semanas**.

## Acta Reunión de planificación iteración 6

---

### 1. Retrospectiva

#### i. Revisión

Logros	Obstáculos	Sorpresas
Mantener el ritmo de avance	Comprensión de código	

*Tabla 18: Tabla LOS Retrospectiva 5*

#### ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing
Con cada nuevo cambio, se debe de explicar minuciosamente las decisiones y la funcionalidad de los mismos	Empezar con nuevas funcionalidades antes de acabar las modificaciones de las más prioritarias	Mantener las revisiones periódicas
		Aportar ideas creativas cuando los cambios pueden perjudicar el proyecto

*Tabla 19: Tabla SSK Retrospectiva 5*

#### iii. Elementos conseguidos

1. Añadido el nuevo atributo de “dificultad”
2. Se han pulido detalles de los HTML
3. Procesado de documentos: Se ha configurado en *front* la carga masiva de alumnos y palabras
4. *Email*: se ha conectado el sistema a la plataforma de Resend y funciona correctamente
5. *Hash* de contraseñas: se decidió solo mantener el *hash* previo al

guardado y se confió que la conexión HTTPS cifraría los mensajes entre cliente y servidor.

6. Se ha añadido la pantalla de clasificaciones y las funcionalidades en *backend*

iv. Elementos aplazados

1. Procesado de documentos CSV para el cargado masivo de preguntas

v. Decisiones sobre elementos

1. Se ha decidido eliminar los archivos de test debido a falta de tiempo para su desarrollo completo

## 2. Información General

- **Fecha:** 19/05/25
- **Hora de Inicio:** 20:00
- **Hora de Finalización:** 21:30
- **Lugar/Plataforma:** Discord
- **Asistentes:** Miriam Sánchez Saiz y Elena Campanero Belda
- **Objetivo:** Fusión de *Frontend* y *Backend*

## 3. Planificación Detallada de Tareas

- **Elemento aplazado:** Procesado de documentos CSV para el cargado masivo de preguntas (Responsable: Miriam)
- Iniciar unión de front y back (Responsable: Ambas)
  - Conexión de archivos en *frontend* de *dashboard*, *login*, *classifications*, *list* y configuración (pestaña de cambio de contraseña) (Responsable: Elena)
  - Conexión de archivos en *frontend* de edición, creación y visualización de Wordles y Grupos (Responsable: Miriam)
  - Centralización de peticiones al *backend* desde *frontend* en *apiService* (Responsable: Elena)
- **Dockerización** de la aplicación (Responsable: Elena)
  - Definición de archivos de contenerización (*docker-compose*, *dockerfile*...)

- Definición de archivo de migración
- Definición de archivo de Semilla

#### 4. Identificación de Posibles Bloqueos/Dependencias

- En esta etapa hay muchas dependencias ya que al empezar a conectar funciones entre *frontend* y *backend* y eliminar el *live-server* (que anteriormente había hecho de soporte para el *frontend*), van a dejar de funcionar correctamente todas las funcionalidades hasta su correcta conexión.

#### 5. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta Iteración será de **1 semana**.

## Acta Reunión de planificación iteración 7

---

### 1. Retrospectiva

#### i. Revisión

Logros	Obstáculos	Sorpresas
Aportación de soluciones creativas	Tiempo: fecha límite muy próxima	
Se han podido abordar requisitos no esperados		

*Tabla 20 Tabla LOS Retrospectiva 6*

#### ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing

Trabajar todos <i>fullstack</i> , tanto en <i>frontend</i> como en <i>backend</i>		Mantener las revisiones periódicas
		Aportar ideas creativas cuando los cambios pueden perjudicar el proyecto
		Solucionar los problemas con la misma eficiencia

**Tabla 21:** Tabla SSK Retrospectiva 6

iii. Elementos conseguidos

1. Unión parcial de front y back
2. Lógica en back para obligar a cambiar la contraseña: generación de contraseñas iniciales débiles y filtrado en login de las mismas. Si son débiles, envía por la respuesta HTTP el elemento “*forcePasswordChange*”
3. Centralizar las funciones de *backend* en *frontend* en el archivo *apiService*
4. *Elemento no esperado:* Generar y añadir el icono favicon (Responsable: Elena)
5. *Elemento no esperado:* Añadido *popup* de mensaje de expiración de sesión (Responsable: Elena)

iv. Elementos aplazados

1. Terminar la unión de *frontend* y *backend* (Edición, Creación y Visualización de Wordles )
2. Arreglar problema de palabras duplicadas
3. Añadir funcionalidad de cerrar *popups* tras subir documentos de carga masiva

## 2. Información General

- **Fecha:** 27/05/25

- **Hora de Inicio:** 20:41
- **Hora de Finalización:** 21:40
- **Lugar/Plataforma:** Discord
- **Asistentes:** Miriam Sánchez Saiz y Elena Campanero Belda
- **Objetivo:** Solución de errores y refinamiento de la aplicación final

### 3. Planificación Detallada de Tareas

- *Elemento aplazado:* Terminar la unión de *frontend* y *backend* (Edición, Creación y Visualización de Wordles) (Responsable: Miriam)
- Arreglar problema de palabras duplicadas (Responsable: Miriam)
- Añadir funcionalidad de cerrar *popups* tras subir documentos de carga masiva (Responsable: Elena)
- Hacer tests de funcionamiento y arreglar errores (Responsable: Elena)
- Actualizar Swagger (Responsable: Elena)
- Centralizar Errores en errorHandler (Responsable: Elena)
- Adición de información a las pestañas de creación y edición de Wordles y Grupos (Responsable: Ambas)
- Generación de Semilla para futura demo (Responsable: Elena)

### 4. Identificación de Posibles Bloqueos/Dependencias

- En esta etapa hubo una dependencia a la hora de trabajar con los Wordles ya que se necesitaba la propagación del cambio en todas las capas para que funcionaran correctamente

### 5. Próximos Pasos y Acuerdos

- Iniciar el desarrollo de las tareas asignadas.
- Se ha acordado que la **duración** de esta iteración será de **1 semana**.

## Acta Retrospectiva 09/06/25

---

## 1. Retrospectiva

### i. Revisión

Logros	Obstáculos	Sorpresas
	Se ha tenido que retrasar la fecha de entrega por no poder llegar a los objetivos en el tiempo acordado	Ha surgido un problema de prestación de servicios gratuitos con la aplicación Resend

Tabla 22: Tabla LOS Retrospectiva 7

### ii. Dinámica de Equipo y Proceso

Start doing	Stop doing	Keep doing
		Trabajar <i>fullstack</i>

Tabla 23: Tabla SSK Retrospectiva 7

### iii. Elementos conseguidos

1. Finalización de unión de *frontend* y *backend*
2. Solución de problemas y errores
  - a. Duplicación de palabras
  - b. Errores de unión y conexiones
  - c. Preparación de archivos para despliegue en servidor
  - d. Errores y detalles generales
3. Creación y conexión del *errorHandler*
4. Actualización final del archivo *swagger*
5. Actualización de *Readme*
6. Adición de iconos, *popups* y *tooltips* a las pestañas de creación y edición de *Wordles* y *Grupos*
7. *Elemento inesperado*: Cambio de soporte para el envío de *emails* : finalmente el envío de *emails* se realizará por *Gmail* y *nodemailer* en vez de por *Resend*

## Anexo 4

### “Datos de Semilla para la demostración”

Este anexo describe la estructura y contenido de los datos de semilla precargados en la base de datos de EduWordle. Estos datos han sido diseñados para facilitar las demostraciones de todas las funcionalidades desarrolladas en el presente TFG.

#### 1. Cuentas de Usuario

Se han generado varias cuentas de profesores y alumnos con la misma estructura genérica.

##### 1.1. Cuentas de Profesor

Rol	Descripción
Profesor Demo (Principal)	Cuenta más completa. Ideal para demostrar la creación de Wordles, gestión de grupos y seguimiento de alumnos. Es el propietario de los grupos y Wordles temáticos
Profesores Genéricos	Cuentas adicionales sin grupos ni Wordles precreados. Útiles para mostrar el flujo de un profesor nuevo en la plataforma. (se han generado 19 cuentas)

*Tabla 24: Semilla: cuentas de usuario con rol profesor*

## 1.2. Cuentas de Alumno

Se han creado 20 cuentas de alumno vinculadas a los grupos de los profesores para permitirles jugar a los Wordles asignados.

## 2. Grupos creados

El “Profesor Demo” ha creado los siguientes grupos temáticos que organizan a los alumnos genéricos.

- Grupo de Diseño SW
- Grupo Pruebas y Calidad
- Grupo Arquitectura y Patrones

## 3. Wordles jugables

Se han creado 5 Wordles principales, pertenecientes a “Profesor Demo”, cada uno con su temática, palabras y preguntas específicas relacionadas con la asignatura de Fundamentos de Ingeniería del *Software*.

### 3.1. Principios SOLID

**Dificultad:** Alta

Palabra	Pista
---------	-------

Responsabilidad	El principio S de SOLID: una clase debe tener solo una razón para cambiar.
Abierto	El principio O de SOLID: abierto para extensión, cerrado para modificación.
Liskov	El principio L de SOLID: los objetos de un tipo base deben poder ser sustituidos por objetos de un subtipo.
Interfaz	El principio I de SOLID: mejor tener muchas interfaces pequeñas y específicas que una grande.
Dependencia	El principio D de SOLID: los módulos de alto nivel no deben depender de los de bajo nivel, sino de abstracciones.
Cohesión	Mide cuán relacionadas y enfocadas están las responsabilidades de un módulo. Alta es deseable.
Acoplamiento	Mide la interdependencia entre módulos. Bajo es deseable para facilitar el mantenimiento.

*Tabla 25: Semilla: palabras Wordle 1*

Nº	Pregunta	Opciones	Respuesta correcta
----	----------	----------	--------------------

1	¿Qué principio de SOLID dice que una clase debe tener solo una razón para cambiar?	<ul style="list-style-type: none"> <li>- Responsabilidad Única</li> <li>- Abierto/Cerrado</li> <li>- Sustitución Liskov</li> </ul>	Responsabilidad Única
2	El principio que indica que las entidades de <i>software</i> deben estar abiertas para extensión, pero cerradas para modificación, es el de:	<ul style="list-style-type: none"> <li>- Liskov Substitution</li> <li>- Open/Closed</li> <li>- Dependency Inversion</li> </ul>	Open/Closed
3	Según el principio de Sustitución de Liskov, si S es un subtipo de T, ¿pueden los objetos de tipo T ser reemplazados por objetos de tipo S sin alterar las propiedades deseables del programa?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Verdadero
4	¿Qué principio de SOLID sugiere que es mejor tener muchas interfaces pequeñas y específicas que una interfaz grande y general?	<ul style="list-style-type: none"> <li>- Interface Segregation</li> <li>- Dependency Inversion</li> <li>- Single Responsibility</li> </ul>	Interface Segregation
5	El principio de Inversión de Dependencia establece que los módulos de alto nivel no deben depender de módulos de bajo nivel, sino de:	<ul style="list-style-type: none"> <li>- Implementaciones concretas</li> <li>- Abstracciones</li> <li>- Clases base</li> </ul>	Abstracciones
6	¿Qué letra del acrónimo SOLID se relaciona con la frase "Build abstractions on stable interfaces, not volatile implementations"?	<ul style="list-style-type: none"> <li>- S</li> <li>- O</li> <li>- D</li> </ul>	D
7	Un módulo debe ser fácil de extender sin modificar	<ul style="list-style-type: none"> <li>- Abierto/Cerrado</li> </ul>	Abierto/Cerrado

	su código fuente. ¿A qué principio SOLID se refiere?	<ul style="list-style-type: none"> <li>- Responsabilidad Única</li> <li>- Inversión de Dependencia</li> </ul>	
8	Si una clase cambia por más de un motivo, ¿qué principio de SOLID se está violando?	<ul style="list-style-type: none"> <li>- Single Responsibility</li> <li>- Liskov Substitution</li> <li>- Interface Segregation</li> </ul>	Single Responsibility
9	¿Cuál de los principios SOLID es el más difícil de entender y aplicar según algunos desarrolladores?	<ul style="list-style-type: none"> <li>- S (SRP)</li> <li>- L (LSP)</li> <li>- D (DIP)</li> </ul>	D (DIP)
10	Un buen diseño SOLID lleva a un <i>software</i> más:	<ul style="list-style-type: none"> <li>- Acoplado</li> <li>- Flexible</li> <li>- Rígido</li> </ul>	Flexible
11	¿Qué principio de SOLID busca reducir el acoplamiento y aumentar la cohesión?	<ul style="list-style-type: none"> <li>- Responsabilidad Única</li> <li>- Interface Segregation</li> <li>- Inversión de Dependencia</li> </ul>	Inversión de Dependencia
12	El uso excesivo de herencia de implementación puede llevar a la violación de qué principio SOLID?	<ul style="list-style-type: none"> <li>- Open/Closed</li> <li>- Liskov Substitution</li> <li>- Interface Segregation</li> </ul>	Liskov Substitution
13	Si una interfaz tiene demasiados métodos, y las clases que la implementan no necesitan todos	<ul style="list-style-type: none"> <li>- Single Responsibility</li> <li>- Interface Segregation</li> </ul>	Interface Segregation

	ellos, ¿qué principio SOLID se viola?	- Dependency Inversion	
14	Las dependencias deben ir de lo concreto a lo abstracto. ¿Verdadero o Falso?	- Verdadero - Falso	Falso
15	Cuando un módulo de alto nivel depende de un módulo de bajo nivel concreto, se viola el principio de:	- Responsabilidad Única - Inversión de Dependencia - Abierto/Cerrado	Inversión de Dependencia
16	¿Qué principios de SOLID están relacionados con la mejora del acoplamiento y la cohesión en el diseño de <i>software</i> ?	- Single Responsibility Principle (SRP) - Liskov Substitution Principle (LSP) - Interface Segregation Principle (ISP) - Dependency Inversion Principle (DIP)	- Single Responsibility Principle (SRP) - Interface Segregation Principle (ISP) - Dependency Inversion Principle (DIP)
17	¿Qué principios de SOLID se consideran los pilares de un diseño de <i>software</i> flexible y extensible?	- Open/Closed Principle (OCP) - Liskov Substitution Principle (LSP) - Interface Segregation Principle (ISP)	- Open/Closed Principle (OCP) - Liskov Substitution Principle (LSP) - Interface Segregation

			Principle (ISP)
--	--	--	-----------------

*Tabla 26: Semilla: preguntas Wordle 1*

### 3.2. Patrones de Diseño

**Dificultad:** Baja

Palabra	Pista
Fábrica	Patrón creacional que encapsula la lógica de creación de objetos complejos
Builder	Patrón creacional que construye objetos complejos paso a paso, permitiendo diferentes representaciones
Singleton	Patrón creacional que asegura que una clase tenga una sola instancia y proporciona un punto de acceso global a ella
Prototype	Patrón creacional que permite crear nuevos objetos clonando instancias existentes sin acoplarse a sus clases concretas
Abstract	Este adjetivo precede al patrón creacional que crea familias de objetos relacionados

Factoría	Variante del patrón Factory Method donde se usa un método estático para crear objetos
Complejos	Los patrones creacionales son útiles para la creación de objetos de este tipo

*Tabla 27: Semilla: palabras Wordle 2*

Nº	Pregunta	Opciones	Respuesta correcta
1	¿Qué patrón asegura que una clase tenga una sola instancia y proporciona un punto de acceso global a ella?	<ul style="list-style-type: none"> <li>- Factory Method</li> <li>- Builder</li> <li>- Singleton</li> </ul>	Singleton
2	¿Qué patrón creacional se usa para construir objetos complejos paso a paso?	<ul style="list-style-type: none"> <li>- Factory Method</li> <li>- Builder</li> <li>- Prototype</li> </ul>	Builder
3	¿Qué patrón creacional permite crear nuevos objetos clonando instancias existentes?	<ul style="list-style-type: none"> <li>- Abstract Factory</li> <li>- Prototype</li> <li>- Singleton</li> </ul>	Prototype
4	¿Qué patrón creacional define una interfaz para crear un objeto, pero deja que las subclasses decidan qué clase instanciar?	<ul style="list-style-type: none"> <li>- Abstract Factory</li> <li>- Factory Method</li> <li>- Builder</li> </ul>	Factory Method

5	¿Qué patrón creacional proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas?	<ul style="list-style-type: none"> <li>- Abstract Factory</li> <li>- Factory Method</li> <li>- Builder</li> </ul>	Abstract Factory
6	El patrón que oculta la lógica de inicialización del objeto es un patrón:	<ul style="list-style-type: none"> <li>- Estructural</li> <li>- Comportamental</li> <li>- Creacional</li> </ul>	Creacional
7	¿Es el patrón "Lazy Initialization" un patrón creacional?	<ul style="list-style-type: none"> <li>- Sí</li> <li>- No</li> </ul>	Si
8	¿Qué problema intentan resolver los patrones creacionales?	<ul style="list-style-type: none"> <li>- Comunicación entre objetos</li> <li>- Estructura de objetos</li> <li>- Creación de objetos</li> </ul>	Creación de objetos
9	El patrón "Object Pool" es un patrón creacional. ¿Verdadero o Falso?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Verdadero
10	¿Qué patrón se usaría para crear una configuración global en una aplicación web?	<ul style="list-style-type: none"> <li>- Builder</li> <li>- Singleton</li> <li>- Factory</li> </ul>	Singleton
11	Cuando la creación de objetos es muy costosa, ¿qué patrón creacional es útil?	<ul style="list-style-type: none"> <li>- Factory Method</li> <li>- Prototype</li> <li>- Abstract Factory</li> </ul>	Prototype
12	Un patrón creacional que permite que una clase	<ul style="list-style-type: none"> <li>- Singleton</li> </ul>	Factory Method

	delegue la creación de objetos a sus subclases es...	<ul style="list-style-type: none"> <li>- Factory Method</li> <li>- Builder</li> </ul>	
13	¿Qué patrón facilita la creación de variantes de un mismo producto en diferentes plataformas (ej: Button para Windows y MacOS)?	<ul style="list-style-type: none"> <li>- Builder</li> <li>- Abstract Factory</li> <li>- Factory Method</li> </ul>	Abstract Factory
14	El patrón que desacopla la construcción de un objeto de su representación es:	<ul style="list-style-type: none"> <li>- Factory</li> <li>- Builder</li> <li>- Prototype</li> </ul>	Builder
15	Este patrón es útil cuando el proceso de creación de un objeto es muy complejo y puede tener diferentes pasos.	<ul style="list-style-type: none"> <li>- Singleton</li> <li>- Abstract Factory</li> <li>- Builder</li> </ul>	Builder
16	¿Cuáles de los siguientes son patrones de diseño creacionales del "Gang of Four" (GoF)?	<ul style="list-style-type: none"> <li>- Adapter</li> <li>- Factory Method</li> <li>- Singleton</li> <li>- Strategy</li> <li>- Builder</li> </ul>	<ul style="list-style-type: none"> <li>- Factory Method</li> <li>- Singleton</li> <li>- Builder</li> </ul>
17	Un patrón creacional es útil cuando:	<ul style="list-style-type: none"> <li>- Se desea ocultar la lógica de creación de objetos</li> <li>- Se necesita una única instancia de una clase</li> <li>- La creación de objetos es</li> </ul>	<ul style="list-style-type: none"> <li>- Se desea ocultar la lógica de creación de objetos</li> <li>- Se necesita</li> </ul>

		un proceso simple y directo	una única instancia de una clase
18	Si necesitas crear diferentes versiones de un producto, cada una con un conjunto diferente de componentes, ¿qué patrones creacionales podrías considerar?	<ul style="list-style-type: none"> <li>- Builder</li> <li>- Abstract Factory</li> <li>- Singleton</li> <li>- Observer</li> </ul>	<ul style="list-style-type: none"> <li>- Builder</li> <li>- Abstract Factory</li> </ul>

*Tabla 28: Semilla: preguntas Wordle 2*

### 3.3. Antipatrones Comunes en el Software

**Dificultad:** Alta

Palabra	Pista
Godclass	Antipatrón donde una clase acumula demasiadas responsabilidades. También llamada "Clase Dios"
Espagueti	Antipatrón que describe código con un flujo de control caótico y difícil de seguir
BigBallOfMud	Antipatrón arquitectónico que describe un sistema sin estructura discernible, que crece de forma ad hoc
Lockin	Problema de dependencia excesiva de un proveedor o

	tecnología específica, impidiendo la migración
Duplication	Antipatrón donde el mismo código o lógica se repite en múltiples lugares
Rigidez	Antipatrón donde el <i>software</i> es difícil de cambiar debido a un diseño inflexible, a menudo resultado de un alto acoplamiento
Fragilidad	Antipatrón donde un pequeño cambio en un lugar inesperadamente causa fallos en muchas otras áreas del sistema
Oro	Se refiere al antipatrón "Martillo de Oro", donde una solución o tecnología se aplica a todos los problemas

*Tabla 29: Semilla: palabras Wordle 3*

Nº	Pregunta	Opciones	Respuesta correcta
1	¿Cómo se llama el antipatrón donde una clase tiene demasiadas responsabilidades y se vuelve un punto central de la aplicación?	<ul style="list-style-type: none"> <li>- Clase Dios</li> <li>- Acoplamiento Fuerte</li> <li>- Código Espagueti</li> </ul>	Clase Dios
2	Un sistema con una estructura indistinta, mal	<ul style="list-style-type: none"> <li>- Arquitectura de Capas</li> </ul>	Bola de Barro

	documentada y sin patrones reconocibles se conoce como:	<ul style="list-style-type: none"> <li>- Bola de Barro Gigante</li> <li>- Diseño Estructurado</li> </ul>	Gigante
3	¿Qué antipatrón describe una base de código donde el flujo de control es caótico y difícil de seguir?	<ul style="list-style-type: none"> <li>- Código Ladrillo</li> <li>- Código Chicle</li> <li>- Código Espagueti</li> </ul>	Código Espagueti
4	El antipatrón "Golden Hammer" se refiere al uso indiscriminado de una tecnología o herramienta en todos los problemas. ¿Verdadero o Falso?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Verdadero
5	Cuando los cambios en una parte del código afectan a muchas otras partes no relacionadas, estamos ante un problema de:	<ul style="list-style-type: none"> <li>- Baja Cohesión</li> <li>- Alto Acoplamiento</li> <li>- Duplicación de Código</li> </ul>	Alto Acoplamiento
6	¿Qué antipatrón de diseño se evita aplicando el principio de Responsabilidad Única?	<ul style="list-style-type: none"> <li>- Clase Dios</li> <li>- Conexión Directa</li> <li>- Bloqueo Mutuo</li> </ul>	Clase Dios
7	El antipatrón "Copy-Paste Programming" lleva a una violación directa de qué principio de diseño?	<ul style="list-style-type: none"> <li>- DRY (Don't Repeat Yourself)</li> <li>- YAGNI (You Aren't Gonna Need It)</li> <li>- KISS (Keep It Simple, Stupid)</li> </ul>	DRY (Don't Repeat Yourself)
8	Si un desarrollador se niega a refactorizar un	<ul style="list-style-type: none"> <li>- Rigidez</li> </ul>	Rigidez

	código por miedo a romperlo, ¿qué antipatrón podría estar sufriendo el proyecto?	<ul style="list-style-type: none"> <li>- Fragilidad</li> <li>- Inmovilidad</li> </ul>	
9	¿Qué antipatrón es el opuesto a un buen diseño modular?	<ul style="list-style-type: none"> <li>- Clase Dios</li> <li>- Interfaz Inflada</li> <li>- Principio de Cierre</li> </ul>	Clase Dios
10	Cuando un componente cambia muy a menudo, y eso causa cambios en otros componentes, se le llama:	<ul style="list-style-type: none"> <li>- Acoplamiento</li> <li>- Cohesión</li> <li>- Fragilidad</li> </ul>	Fragilidad
11	El antipatrón "Conmutación por error (Fail-Safe)" es un error de seguridad donde un sistema falla de forma insegura. ¿Verdadero o Falso?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Verdadero
12	¿Cómo se llama la situación en la que dos módulos A y B se llaman entre sí, creando una dependencia circular?	<ul style="list-style-type: none"> <li>- Acoplamiento temporal</li> <li>- Acoplamiento transitivo</li> <li>- Acoplamiento cíclico</li> </ul>	Acoplamiento cíclico
13	El antipatrón "Feature Envy" (envidia de características) ocurre cuando un método de una clase está más interesado en los datos de otra clase que en los suyos propios. ¿Verdadero o Falso?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Verdadero
14	¿Qué antipatrón se describe como una clase que tiene demasiadas dependencias y es difícil de	<ul style="list-style-type: none"> <li>- Clase de servicio</li> <li>- Clase de utilidad</li> </ul>	Clase de Dios

	probar y reutilizar?	- Clase de Dios	
15	¿Qué características son comunes en un "Big Ball of Mud"?	<ul style="list-style-type: none"> <li>- Buena documentación</li> <li>- Falta de estructura discernible</li> <li>- Fácil de mantener</li> <li>- Crecimiento ad hoc</li> </ul>	<ul style="list-style-type: none"> <li>- Falta de estructura discernible</li> <li>- Crecimiento ad hoc</li> </ul>
16	El "Código Espagueti" se caracteriza por:	<ul style="list-style-type: none"> <li>- Flujo de control caótico</li> <li>- Alto acoplamiento funcional</li> <li>- Baja cohesión</li> <li>- Manejo de errores centralizado</li> </ul>	<ul style="list-style-type: none"> <li>- Flujo de control caótico</li> <li>- Baja cohesión</li> </ul>
17	¿Cuáles de los siguientes son efectos negativos de un alto acoplamiento?	<ul style="list-style-type: none"> <li>- Rigidez</li> <li>- Facilidad de refactorización</li> <li>- Fragilidad</li> <li>- Reutilización</li> </ul>	<ul style="list-style-type: none"> <li>- Rigidez</li> <li>- Fragilidad</li> </ul>

*Tabla 30: Semilla: preguntas Wordle 3*

### 3.4. Tipos y Técnicas de Pruebas Software

**Dificultad: Baja**

<b>Palabra</b>	<b>Pista</b>
Unitarias	Pruebas que verifican el comportamiento de componentes individuales de forma aislada
Integración	Pruebas que verifican la interacción entre diferentes módulos o sistemas
CajaNegra	Tipo de prueba que ignora la estructura interna del código y se enfoca en la funcionalidad externa
CajaBlanca	Tipo de prueba que evalúa la estructura interna y el funcionamiento del código
Regresión	Pruebas que se realizan para asegurar que los cambios recientes no han introducido nuevos defectos en funcionalidades existentes
Aceptación	Pruebas finales para validar que el sistema cumple con los requisitos del usuario o del cliente
Sistema	Pruebas que evalúan el sistema completo e integrado para verificar que cumple los requisitos funcionales y no funcionales
Cobertura	Métrica que indica la proporción de código ejecutado

	durante las pruebas
Mock	Objeto de prueba que simula el comportamiento de un componente y registra las interacciones para verificación

*Tabla 31: Semilla: palabras Wordle 4*

Nº	Pregunta	Opciones	Respuesta correcta
1	¿Qué tipo de pruebas verifican el comportamiento de componentes individuales de forma aislada?	<ul style="list-style-type: none"> <li>- Unitarias</li> <li>- Integración</li> <li>- Sistema</li> </ul>	Unitarias
2	Las pruebas que verifican la interacción entre diferentes módulos o sistemas son:	<ul style="list-style-type: none"> <li>- Funcionales</li> <li>- De Integración</li> <li>- De Carga</li> </ul>	De Integración
3	¿Qué tipo de pruebas se realizan para asegurar que los cambios recientes no han introducido nuevos defectos en funcionalidades existentes?	<ul style="list-style-type: none"> <li>- De Humo</li> <li>- De Regresión</li> <li>- De Sanidad</li> </ul>	De Regresión
4	Las pruebas de caja negra se centran en:	<ul style="list-style-type: none"> <li>- La estructura interna del código</li> <li>- La funcionalidad desde la perspectiva del usuario</li> </ul>	La funcionalidad desde la perspectiva del usuario

		<ul style="list-style-type: none"> <li>- Los caminos de ejecución del código</li> </ul>	
5	¿Qué técnica de caja blanca se asegura de que cada línea de código sea ejecutada al menos una vez?	<ul style="list-style-type: none"> <li>- Cobertura de Decisión</li> <li>- Cobertura de Sentencia</li> <li>- Cobertura de Condición</li> </ul>	Cobertura de Sentencia
6	¿Cuál es el objetivo principal de las pruebas de aceptación?	<ul style="list-style-type: none"> <li>- Encontrar errores internos del código</li> <li>- Validar que el sistema cumple los requisitos del usuario</li> <li>- Medir el rendimiento bajo carga</li> </ul>	Validar que el sistema cumple los requisitos del usuario
7	El "Test Driven Development" (TDD) es una práctica que prioriza la escritura de pruebas _____ antes de la escritura de código.	<ul style="list-style-type: none"> <li>- De Integración</li> <li>- Unitarias</li> <li>- De Sistema</li> </ul>	Unitarias
8	Las pruebas de _____ intentan descubrir defectos explotando las vulnerabilidades del sistema.	<ul style="list-style-type: none"> <li>- Usabilidad</li> <li>- Seguridad</li> <li>- Rendimiento</li> </ul>	Seguridad
9	Cuando un sistema se somete a un volumen muy alto de peticiones para ver su comportamiento, estamos hablando de pruebas de:	<ul style="list-style-type: none"> <li>- Estrés</li> <li>- Carga</li> <li>- Integración</li> </ul>	Carga

10	¿Es el "Exploratory Testing" un tipo de prueba formal?	<ul style="list-style-type: none"> <li>- Sí</li> <li>- No</li> </ul>	No
11	¿Qué tipo de prueba se realiza con el propósito de garantizar que el <i>software</i> se puede desplegar y operar en un nuevo entorno sin problemas?	<ul style="list-style-type: none"> <li>- Pruebas de Usabilidad</li> <li>- Pruebas de Migración</li> <li>- Pruebas de Recuperación</li> </ul>	Pruebas de Migración
12	Una prueba de caja blanca se basa en la estructura interna del <i>software</i> . ¿Verdadero o Falso?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Verdadero
13	¿Qué se utiliza para simular el comportamiento de componentes aún no desarrollados en pruebas unitarias?	<ul style="list-style-type: none"> <li>- Stubs</li> <li>- Drivers</li> <li>- Monitores</li> </ul>	Stubs
14	El proceso de encontrar la causa raíz de un defecto se conoce como:	<ul style="list-style-type: none"> <li>- Debugging</li> <li>- Testing</li> <li>- Verification</li> </ul>	Debugging
15	Las pruebas de ____ se centran en verificar que el sistema es fácil de usar y aprender.	<ul style="list-style-type: none"> <li>- Rendimiento</li> <li>- Seguridad</li> <li>- Usabilidad</li> </ul>	Usabilidad
16	¿Qué tipo de prueba se realiza después de un parche o actualización para asegurar que el sistema sigue funcionando como se espera?	<ul style="list-style-type: none"> <li>- Sanity Test</li> <li>- Smoke Test</li> <li>- Regression Test</li> </ul>	Regression Test
17	¿Qué principio de pruebas sugiere que no es	<ul style="list-style-type: none"> <li>- Principio de Pareto</li> </ul>	Exhaustividad es

	posible probar exhaustivamente todas las combinaciones de entrada?	<ul style="list-style-type: none"> <li>- Paradoja del Pesticida</li> <li>- Exhaustividad es imposible</li> </ul>	imposible
18	¿Qué elementos son característicos de las pruebas de caja negra?	<ul style="list-style-type: none"> <li>- Basadas en la especificación de requisitos</li> <li>- Ignoran la estructura interna del código</li> <li>- Requieren acceso al código fuente <ul style="list-style-type: none"> <li>- Se enfocan en el comportamiento externo</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Basadas en la especificación de requisitos</li> <li>- Ignoran la estructura interna del código</li> <li>- Se enfocan en el comportamiento externo</li> </ul>
19	¿Cuáles de los siguientes son tipos de pruebas no funcionales?	<ul style="list-style-type: none"> <li>- Pruebas de Usabilidad</li> <li>- Pruebas de Integración</li> <li>- Pruebas de Rendimiento</li> <li>- Pruebas de Regresión</li> </ul>	<ul style="list-style-type: none"> <li>- Pruebas de Usabilidad</li> <li>- Pruebas de Rendimiento</li> </ul>
20	Las pruebas unitarias son beneficiosas porque:	<ul style="list-style-type: none"> <li>- Ayudan a encontrar errores temprano <ul style="list-style-type: none"> <li>- Facilitan la refactorización</li> </ul> </li> <li>- Miden la satisfacción del</li> </ul>	<ul style="list-style-type: none"> <li>- Ayudan a encontrar errores temprano</li> <li>- Facilitan la</li> </ul>

		usuario - Aseguran el rendimiento del sistema	refactorización
--	--	--	-----------------

*Tabla 32: Semilla: preguntas Wordle 4*

### 3.5. Modelos y Procesos Software

**Dificultad:** Baja

Palabra	Pista
Cascada	Modelo de ciclo de vida del <i>software</i> lineal y secuencial, donde cada fase debe completarse antes de que comience la siguiente
Ágil	Conjunto de metodologías de desarrollo de <i>software</i> que enfatizan la colaboración, la entrega incremental y la adaptación al cambio
Scrum	Framework ágil iterativo e incremental para gestionar proyectos de desarrollo de <i>software</i>
Kanban	Método visual para gestionar el flujo de trabajo, centrado en limitar el trabajo en progreso y maximizar la eficiencia

Sprint	Período de tiempo fijo y corto (generalmente 1-4 semanas) durante el cual un equipo de Scrum trabaja para completar un conjunto de trabajo.
Backlog	Listas de tareas o requisitos pendientes en metodologías ágiles
Iterativo	Característica de los procesos ágiles donde el desarrollo se realiza en ciclos repetitivos
Incremental	Característica de los procesos ágiles donde el producto se construye en pequeñas partes funcionales y se entrega regularmente
Devops	Cultura y práctica que unifica el desarrollo de <i>software</i> (Dev) y las operaciones de <i>software</i> (Ops)
XP	Una de las metodologías ágiles, conocida por prácticas como programación en parejas y TDD

Tabla 33: Semilla: palabras Wordle 5

Nº	Pregunta	Opciones	Respuesta correcta
1	¿Qué modelo de ciclo de vida del <i>software</i> se caracteriza por etapas secuenciales y rigurosas, con poco retorno a fases anteriores?	<ul style="list-style-type: none"> <li>- Espiral</li> <li>- Ágil</li> <li>- Cascada</li> </ul>	Cascada

2	En Scrum, ¿cómo se llama el período de tiempo fijo y corto durante el cual un equipo trabaja para completar un incremento de producto?	<ul style="list-style-type: none"> <li>- Fase</li> <li>- Sprint</li> <li>- Iteración</li> </ul>	Sprint
3	¿Qué rol en Scrum es responsable de maximizar el valor del producto y gestionar el Product Backlog?	<ul style="list-style-type: none"> <li>- Scrum Master</li> <li>- Development Team</li> <li>- Product Owner</li> </ul>	Product Owner
4	Un proceso de desarrollo que se adapta a los cambios y enfatiza la colaboración con el cliente es de tipo:	<ul style="list-style-type: none"> <li>- Lineal</li> <li>- Predictivo</li> <li>- Ágil</li> </ul>	Ágil
5	¿Qué modelo de ciclo de vida del <i>software</i> incorpora gestión de riesgos en cada iteración?	<ul style="list-style-type: none"> <li>- Cascada</li> <li>- Espiral</li> <li>- V-Model</li> </ul>	Espiral
6	La frase "Las personas y sus interacciones sobre los procesos y herramientas" pertenece al manifiesto:	<ul style="list-style-type: none"> <li>- XP</li> <li>- Scrum</li> <li>- Ágil</li> </ul>	Ágil
7	¿Qué artefacto de Scrum describe el trabajo que el equipo realizará durante el Sprint?	<ul style="list-style-type: none"> <li>- Product Backlog</li> <li>- Sprint Backlog</li> <li>- Incremento</li> </ul>	Sprint Backlog
8	El concepto de "Minimum Viable Product" (MVP) se asocia comúnmente con metodologías:	<ul style="list-style-type: none"> <li>- Cascada</li> <li>- Ágiles</li> <li>- En V</li> </ul>	Ágiles

9	¿Qué ceremonia de Scrum se realiza al final de cada Sprint para inspeccionar el incremento y adaptar el Product Backlog?	<ul style="list-style-type: none"> <li>- Daily Scrum</li> <li>- Sprint Planning</li> <li>- Sprint Review</li> </ul>	Sprint Review
10	¿Cuál es el principal enfoque del "DevOps"?	<ul style="list-style-type: none"> <li>- Desarrollo rápido</li> <li>- Integración y despliegue continuos</li> <li>- Documentación exhaustiva</li> </ul>	Integración y despliegue continuos
11	En un modelo en cascada, las fases de diseño y desarrollo se superponen para permitir retroalimentación temprana. ¿Verdadero o Falso?	<ul style="list-style-type: none"> <li>- Verdadero</li> <li>- Falso</li> </ul>	Falso
12	¿Qué proceso se enfoca en entregar <i>software</i> en pequeñas partes funcionales de forma regular?	<ul style="list-style-type: none"> <li>- Cascada</li> <li>- Iterativo e Incremental</li> <li>- Big Bang</li> </ul>	Iterativo e Incremental
13	¿Qué tipo de planificación es característica de las metodologías ágiles, centrándose en períodos cortos?	<ul style="list-style-type: none"> <li>- Planificación a largo plazo</li> <li>- Planificación adaptativa</li> <li>- Planificación predictiva</li> </ul>	Planificación adaptativa
14	El objetivo de la retrospectiva en Scrum es:	<ul style="list-style-type: none"> <li>- Planificar el siguiente Sprint</li> <li>- Revisar el incremento del producto</li> </ul>	Mejorar el proceso de trabajo del equipo

		<ul style="list-style-type: none"> <li>- Mejorar el proceso de trabajo del equipo</li> </ul>	
15	¿Qué modelo de desarrollo se conoce por su forma de "V" al representar las fases de verificación y validación?	<ul style="list-style-type: none"> <li>- Espiral</li> <li>- Cascada</li> <li>- V-Model</li> </ul>	V-Model
16	¿Qué artefactos principales se utilizan en Scrum?	<ul style="list-style-type: none"> <li>- Diagrama de Gantt</li> <li>- Product Backlog</li> <li>- Sprint Backlog</li> <li>- Plan de Proyecto Detallado</li> <li>- Incremento</li> </ul>	<ul style="list-style-type: none"> <li>- Product Backlog</li> <li>- Sprint Backlog</li> <li>- Incremento</li> </ul>
17	Las ventajas de los modelos ágiles de desarrollo incluyen:	<ul style="list-style-type: none"> <li>- Rigidez en el cambio de requisitos</li> <li>- Mayor colaboración con el cliente</li> <li>- Entrega temprana y continua de <i>software</i></li> <li>- Menor necesidad de comunicación</li> </ul>	<ul style="list-style-type: none"> <li>- Mayor colaboración con el cliente</li> <li>- Entrega temprana y continua de <i>software</i></li> </ul>
18	¿Qué roles principales existen en el marco de trabajo Scrum?	<ul style="list-style-type: none"> <li>- Jefe de Proyecto</li> <li>- Product Owner</li> <li>- Scrum Master</li> <li>- Equipo de Desarrollo</li> </ul>	<ul style="list-style-type: none"> <li>- Product Owner</li> <li>- Scrum Master</li> <li>- Equipo de Desarrollo</li> </ul>

		- Analista de Negocio	
--	--	-----------------------	--

*Tabla 34: Semilla: preguntas Wordle 5*

#### 4. Vinculación de Wordles a Grupos

Los Wordles creados por el "Profesor Demo" están asignados a los siguientes grupos:

- **Principios SOLID:** Vinculado a "Grupo Diseño SW" y "Grupo Arquitectura y Patrones".
- **Patrones de Diseño:** Vinculado a "Grupo Arquitectura y Patrones".
- **Antipatrones Comunes SW:** Vinculado a "Grupo Diseño SW".
- **Tipos y Técnicas de Pruebas SW:** Vinculado a "Grupo Pruebas y Calidad".
- **Modelos y Procesos SW:** Vinculado a "Grupo Diseño SW" y "Grupo Pruebas y Calidad".

#### 5. Resultados de Juego Simulados

Se han simulado resultados de juego para todos los 20 alumnos genéricos. Cada alumno tiene entre 1 y 3 resultados registrados para cada uno de los 5 *Wordles* temáticos.

- **Puntuaciones:** Asignadas aleatoriamente entre 100 y 600 puntos.

- **Fechas de Juego:** Registradas en los últimos 30 días previos a la generación de la semilla, para simular un historial de actividad reciente.

Estos resultados permiten la demostración de las funcionalidades de *rankings* individuales y grupales, así como el seguimiento del rendimiento de los alumnos por *Wordle* y por grupo.

