



UNIVERSIDAD COMPLUTENSE MADRID

Proyecto de Innovación

Convocatoria 2023/2024

Nº de proyecto 469

Creación y desarrollo de una aplicación móvil como soporte al aprendizaje sobre energías renovables y sostenibilidad para uso de los estudiantes de la UCM

Responsable del Proyecto: M.^a Yolanda Martínez Solana

Facultad de Ciencias de la información

Departamento: Periodismo y Comunicación Global

Contenido

RESUMEN	3
1. Objetivos propuestos en la presentación del proyecto	4
2. Objetivos alcanzados	6
3. Metodología empleada en el proyecto	8
4. Recursos Humanos	9
5. Desarrollo de las actividades	10
6. Anexos	12

RESUMEN

El proyecto se enfocó para completar la campaña de comunicación sobre las energías renovables como garantía de un futuro sostenible para la comunidad universitaria complutense, que se elaboró para la convocatoria Innova 2020-2021 (nº 281), por lo que continúa existiendo un déficit en el conocimiento pragmático de lo que supone el uso de fuentes de energía que no respetan el medio ambiente. Se ha creado una aplicación móvil ad hoc para que la campaña se haga realidad en el uso del estudiantado, ya que supondría una mejora de la calidad docente, la aportación de herramientas de conocimiento adaptadas a las nuevas tecnologías de la información que ya son de uso cotidiano.

El proyecto se centra en fomentar la conciencia y el conocimiento sobre las energías renovables de forma accesible, ágil y entretenida. Para llevarlo a cabo, se ha diseñado una aplicación web en la que, a través de un juego basado en desafíos diarios, las personas usuarias podrán aprender los conceptos más importantes sobre las energías renovables dedicando solo unos minutos al día.

El juego consiste en un desafío diario en el que las personas que acceda al mismo, tendrán la posibilidad de descubrir un nuevo término relacionado con las energías renovables cada día. El usuario tendrá que descubrir la “palabra del día” basándose únicamente en unas reglas sencillas y su propia deducción. Una vez que el utilizador de la aplicación haya sido capaz de descubrir cuál era la palabra, se desplegará una pantalla en la que se mostrará información detallada sobre ese término, tales como: definición, funcionamiento o su relevancia en el contexto de las energías renovables.

Del mismo modo, también se mostrarán consejos o buenos hábitos relacionados con la palabra del día y que, sin requerir demasiado esfuerzo, muestren a los usuarios nuevas formas de aumentar el ahorro energético y de contribuir al cuidado del medio ambiente. Para el desarrollo de esta aplicación web se ha necesitado una combinación de técnicas de programación web y diseño de interfaces de usuario, así como, la colaboración entre Facultades para tener disponibles los contenidos educativos necesarios sobre energías renovables.

Dicha iniciativa no busca solo promover la conciencia ambiental entre los usuarios, sino también difundir información relevante sobre las energías renovables de una forma sencilla y atractiva.

1. Objetivos propuestos en la presentación del proyecto

Como se comunicó en el anterior informe, el punto de partida para la elaboración de este proyecto ha sido la Guía para Periodistas “Comunicar la sostenibilidad” elaborado por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) en 2008. El medio ambiente debería representar lo mismo para todo el planeta.

El principal propósito de este proyecto es diseñar una aplicación web que promueva la conciencia y el conocimiento sobre las energías renovable, utilizando un método atractivo para los usuarios que les permita adquirir los conocimientos propuestos sin tener que utilizar una gran cantidad de esfuerzo o tiempo. Para llegar a cumplir este objetivo vamos a determinar los principales puntos en los que queremos centrarnos:

- la obtención de la información sobre energías renovables que se va a incluir en la aplicación, cuyo propósito es potenciar la divulgación y comunicación de los beneficios ecológicos y socioeconómicos del patrimonio natural.
- se ha llevado a cabo un estudio sobre las distintas energías renovables y su contribución en la lucha contra el cambio climático en la Facultad de Ciencias de la Información. Este estudio se ha utilizado para obtener la información clave que se muestra en la aplicación.
- para proceder a testar la propia aplicación, se diseñará un sistema de puntuaciones a través de un ranking, mostrando tanto las puntuaciones individuales como las de cada Facultad en su conjunto. De este modo, se pretende atraer a la mayor cantidad de usuarios posibles y animarlos a intentar realizar cada reto diariamente.
- para la implementación de la aplicación web, diseñaremos una web sencilla y atractiva al usuario, buscando la facilidad de uso desde el primer momento, y siendo perfectamente operativo, desde un ordenador hasta un móvil. Para ello se diseñarán varios mockups y se probarán distintas interfaces de usuarios hasta dar con la más adecuada.
- otro de los temas más importantes que hemos tratado es la forma en la que vamos a diseñar la aplicación para que los usuarios puedan aprender sobre las energías renovables de una forma amena y entretenida. Después de contrastar varias posibilidades, hemos decidido diseñar un juego basado en retos diarios con el que los usuarios puedan aprender un concepto nuevo cada día.

- otro de los objetivos sería potenciar la divulgación y comunicación de los beneficios ecológicos y socioeconómicos del patrimonio natural. Así como, la integración de las necesidades de conservación y uso sostenible de la biodiversidad en el desarrollo y aplicación de un importante número de políticas sectoriales, a fin de lograr un desarrollo equitativo, inclusivo, justo y duradero.
- como objetivo adicional, queremos que este proyecto sirva como ejemplo de cómo la colaboración entre equipos de ciencia, ingeniería y comunicación, pueden promover la creación de nuevos modelos de enseñanza y de elaboración de contenidos científicos.
- el resultado que queremos obtener se basa en conseguir una competencia transversal con visión global, desde la perspectiva de la comunicación y de las energías renovables y sostenibilidad del medio ambiente. El proyecto responde a la necesidad de aplicar métodos más eficaces para dar a conocer las medidas a adoptar en relación con la materia que queremos difundir, el alumnado y los docentes e investigadores de la Universidad Complutense de Madrid.

2. Objetivos alcanzados

Hacer realidad una campaña de comunicación sobre las energías renovables garantiza de un futuro sostenible para la comunidad universitaria complutense por medio del desarrollo de una aplicación web, estimando que era el medio más accesible para el público objetivo. Para el diseño de esta web se ha intentado conseguir una interfaz con una navegación sencilla y amigable para el usuario, para lo que ha hecho falta refinarla por medio de la elaboración de diferentes mockups y reuniones para mostrar los avances y diferentes puntos de vista.

Una vez diseñada la web, se pasó a la implementación de una serie de juegos haciendo uso de las tecnologías descritas en secciones anteriores. Estos juegos servirían para atraer a los usuarios y conseguir que aprendan conceptos clave sobre energías renovables de una forma divertida. De esta forma, una vez el juego haya finalizado, se mostrará una interfaz que proporcionará información importante sobre la palabra del día, así como consejos que puedan resultar útiles al usuario con respecto al papel de dicha palabra, en el uso de energías renovables. De esta forma, se pretende ayudar a la formación continua del usuario en su día a día para la contribución de un mundo más sostenible.

Para realizar un enfoque más indirecto en la interacción entre los usuarios por medio del uso de un ranking. Este ranking les permitirá ver sus posiciones respecto a otros jugadores y resolver los retos diariamente para mejorar su puntuación y subir puestos en el ranking, incentivando de esta forma el uso regular de la aplicación por parte de las personas usuarias de la aplicación, y por consiguiente, la introducción de conocimientos acerca de las energías renovables de una forma continuada.

De forma esquemática el grupo ha logrado:

- Completar la campaña de comunicación sobre las energías renovables ya que continúa existiendo un déficit en el conocimiento pragmático de lo que supone el uso de fuentes de energía que no respeten el medio ambiente.
- La creación de una aplicación móvil adhoc permite que los estudiantes adquieran un hábito de consulta en lo referido a estas energías.
- El diseño y materialización del reto digital se ha realizado a través de un Trabajo Fin de Grado elaborado por tres estudiantes de 4º de ingeniería informática. De esta forma hemos abierto una línea, que creemos pionera, en la que la innovación docente se convierte en las prácticas elaboradas por estudiantes que

obtienen de esa forma, el Título de Grado con un trabajo teórico práctico interdisciplinar.

- Materializar el concepto de transversalidad, integrando en el equipo a investigadores de las áreas de Informática, Políticas y Sociología y Periodismo.
- Se promueve la toma de conciencia frente a retos que requieren acciones globales pero también acciones locales.
- Se logra una sinergia entre científicos y comunicadores (profesionales e investigadores de la Comunicación) que es imprescindible para que la universidad cumpla su papel de transmisión de conocimientos a la sociedad y aplicar las tecnologías más avanzadas del audiovisual a la enseñanza universitaria, promoviendo nuevos modelos de enseñanzas y de creación de contenidos científicos.
- Se han materializado las sinergias científico-comunicativas a través de un trabajo interdisciplinar y de la utilización de tecnologías audiovisuales aplicadas a la enseñanza.
- La experiencia ha servido de aprendizaje presencial, generando un producto científico que permite su explotación para, al mismo tiempo, aplicarlo a una enseñanza online/ semi presencial a través del audiovisual.
- La realización de este Proyecto a las energías renovables y sostenibilidad del Medio ambiente se trata de cumplir con el concepto de transferencia de conocimiento.
- El proyecto permite la creación de unos materiales que pueden ser empleados para la transferencia de conocimiento de una forma eficaz y amena, como sólo el audiovisual puede permitir, llegando a un amplísimo público, especializado o no especializado; un concepto que, creemos, le da un valor añadido al proyecto.

3. Metodología empleada en el proyecto

La metodología del equipo de trabajo se ha basado en un método colaborativo a través de un espacio virtual para la celebración de reuniones presenciales para la toma de acuerdos hasta la consecución del proyecto final INNOVA.

El mencionado espacio virtual sirve como repositorio de toda la documentación generada, incluidas las actas de las reuniones que se han celebrado, existiendo un canal de comunicación permanente y de participación; para el desarrollo de la estrategia de trabajo siguiendo un protocolo para la creación de una aplicación móvil Android.

Se han aplicado enfoques cualitativos para comprender mejor las necesidades y preferencias de los usuarios, así como para evaluar la experiencia de usuario de manera más detallada.

En cuanto a la metodología propia de la realización de la aplicación:

1.- FASE DE PREPRODUCCIÓN

En esta fase tendríamos los siguientes procesos:

- Reunión inicial de los estudiantes de los campus integrados para el planteamiento del tema a investigar.
- Reuniones de coordinación periódicas - cada 15/20 días - de los estudiantes registrados.

Estas reuniones y sus avances han sido supervisados por el profesorado integrantes del presente Proyecto.

2.- FASE DE POSTPRODUCCIÓN

Se han celebrado briefing de seguimiento de objetivos para realizar una adaptación creativa a las necesidades operativas de la tecnología a utilizar.

Se ha realizado una proyección de todos los materiales preelaborados incluyendo a estudiantes que no han participado en el proyecto y se ha pasado una encuesta para obtener inputs de aceptación o mejora de los trabajos.

La última fase de elaboración de materia se han integrado todas las piezas obtenidas (resultantes de las fases descritas) en los soportes indicados y objetivos del presente proyecto de comunicación con base informática.

4. Recursos Humanos

Equipo

Responsable **MARIA YOLANDA MARTINEZ SOLANA** PDI Complutense mymartin@ucm.es

Miembro **SERGIO BERNABE GARCIA** PDI Complutense sebernab@ucm.es

Miembro **TERESA MARTÍN GARCIA** PDI Universidad de Salamanca teresam@usal.es

Miembro **LUCILA FINKEL MORGENSTERN** PDI Complutense lfinkelm@ucm.es

Miembro **M.^a DEL CARMEN BAÑUELOS MADERA** Estudiante madelban@ucm.es

Miembro **DAVID RUIZ LUJÁN** Estudiante davru03@ucm.es

Miembro **ALVARO GOMEZ MORAN** Estudiante alvgom14@ucm.es

Miembro **CARLA MATOS MEJÍAS** Estudiante carmatos@ucm.es

Miembro **JOSE MARIA MORENO GARCIA** Estudiante jomore10@ucm.es

Miembro **DANIEL PRIETO REMACHA** Estudiante daniprie@ucm.es

Miembro **MIGUEL ZAYAS BOIZA** Estudiante mzayas@ucm.es

5. Desarrollo de las actividades

Fases del proyecto:

- **Análisis sobre las distintas tecnologías que se han contemplado utilizar:** para el desarrollo de la aplicación, comparando las distintas ventajas e inconvenientes que presenta cada una de ellas y concluyendo con una explicación de la decisión final que se ha tomado para cada tecnología utilizada.
- **Diseño de la Aplicación:** se detalla el proceso de diseño de la aplicación, en la que se incluye la arquitectura general, la estructura usada en la base de datos, el análisis de los distintos requisitos y el diseño de los casos de uso, así como, la creación de los prototipos de la interfaz de usuario de la página web y del juego propuesto.
- **Estructuración del Código:** se detalla la organización jerárquica de las carpetas dentro del proyecto, explicando el propósito y el contenido de cada una de ellas. Se utiliza como base para entender mejor la implementación del proyecto.
- **Implementación de la Aplicación:** se detalla la implementación del diseño establecido anteriormente mediante las tecnologías escogidas, analizando en profundidad todo el proceso de desarrollo seguido tanto en front-end como en back-end.
- **Conclusiones** que hemos obtenido tras la realización de este proyecto. Se comentan las distintas implicaciones derivadas del uso de la aplicación, así como posibles mejoras que se puedan incluir en próximas versiones.

Para definir de una forma más visual el tiempo que se estimó dedicar a cada tarea, se diseñó un diagrama de Gantt (ver Figura 1 adjunta) que muestra las distintas tareas con respecto al tiempo total del proyecto.

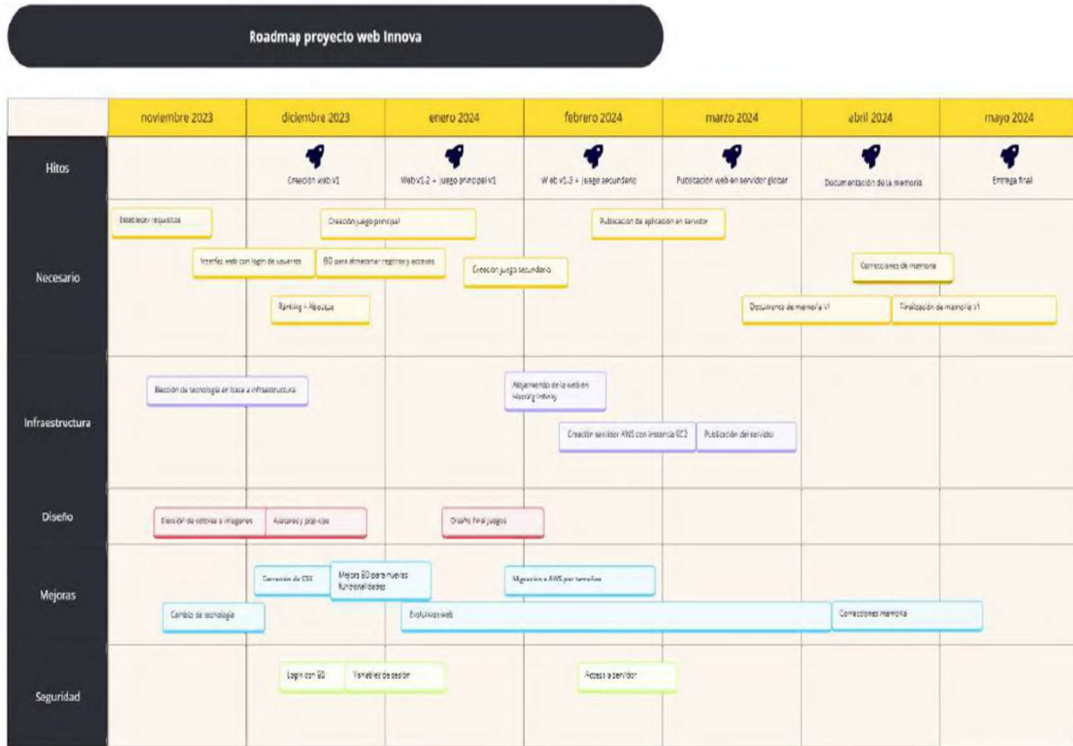


Figura 1: Diagrama de Gantt con la planificación estimada

En resumen, el uso de esta herramienta ha sido fundamental para la planificación y ejecución eficiente del proyecto, proporcionando una estructura clara y bien definida para alcanzar nuestros objetivos.

6. Anexos

Flujo de uso y demo de la Aplicación

En este anexo, se mostrará todas las funcionalidades de la web mediante un flujo de uso normal de un usuario.

A partir de aquí se dividirán en secciones las diferentes acciones que puede realizar el usuario, con qué elementos de la página puede interactuar, y se mostrará cuál es el comportamiento que se ha pensado para que desarrolle el usuario de forma diaria. Se ha optado por una mecánica y diseño sencillo e intuitivo, el cual pretende que el usuario quiera interactuar diariamente con la aplicación.

Registro

En primer lugar, para acceder a la aplicación deberás estar registrado en ella, ya que se necesitará guardar el progreso que hayas conseguido, además de poder posicionarte en el ranking. Para ello simplemente se hace clic en la parte superior derecha de la pantalla en el botón inicio de sesión para acceder al formulario de registro que aparece en la Figura 2.

Como dato a destacar, deberemos registrarnos con nuestro correo de la universidad para poder acceder.

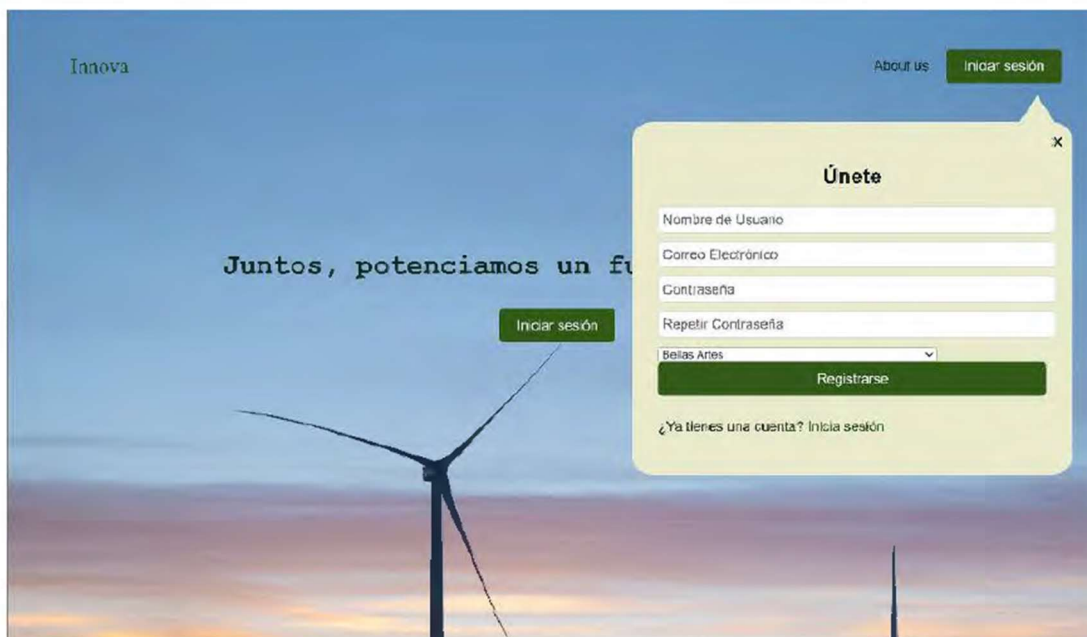


Figura 20: Registro de nuevos usuarios

Inicio de sesión

Si ya tenemos un usuario creado únicamente deberemos de hacer clic en el botón alojado en el centro de la página, y añadir nuestras credenciales.



Figura 3: Inicio de sesión

Modificación de cuenta

Una vez hemos accedido con nuestras credenciales tendremos acceso al ranking y a modificar algunos aspectos de nuestra cuenta en la parte superior izquierda, como por ejemplo cambiar el avatar de usuario, como se muestra en la Figura 4, modificar nuestro nombre de usuario, o darnos de baja Figura 5.

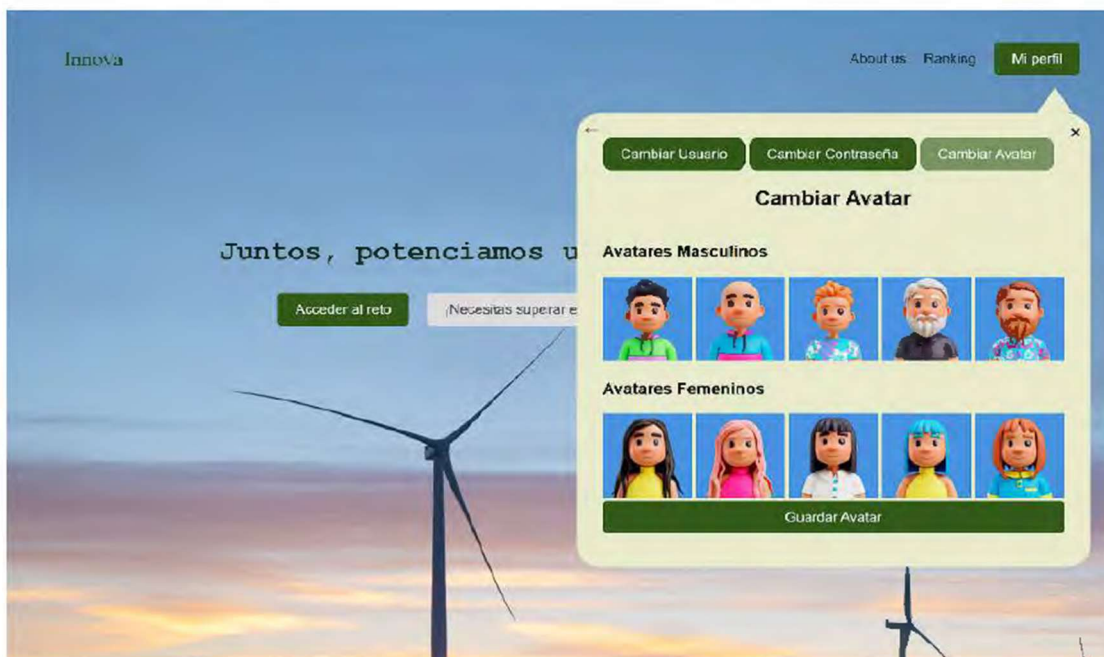


Figura 4: Cambio de avatar en el usuario/a

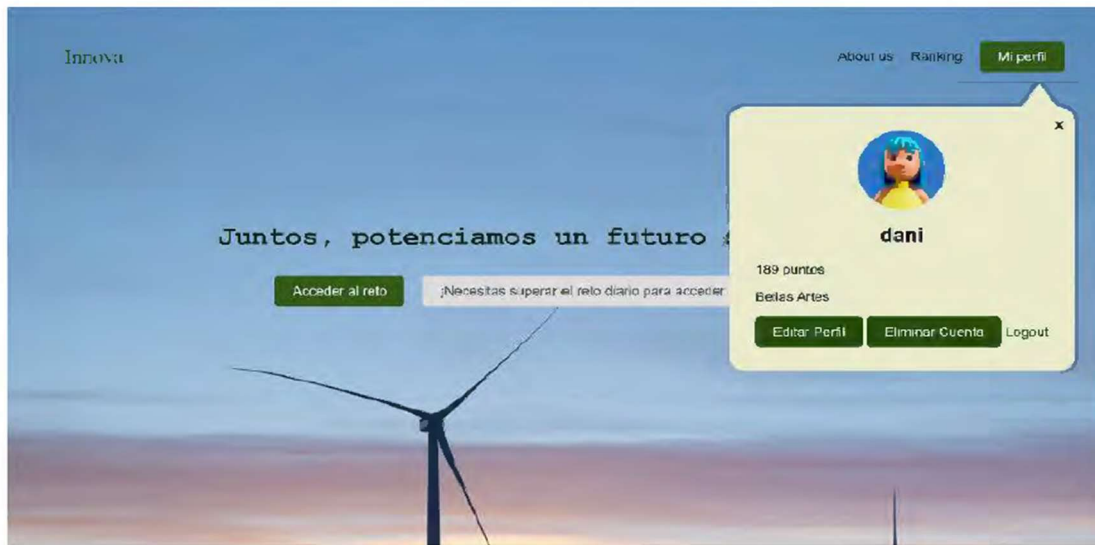


Figura 5: Opciones de usuario/a

Reto diario

Una vez dentro de la página con nuestro usuario, tendremos acceso al reto diario mediante los botones del inicio. Al acceder vemos la página donde poder jugar al reto diario Figura 6, y tras haberlo completado nos aparecerá un mensaje con nuestra puntuación en caso de haberlo superado Figura 7.



Figura 6: Reto diario con varios intentos

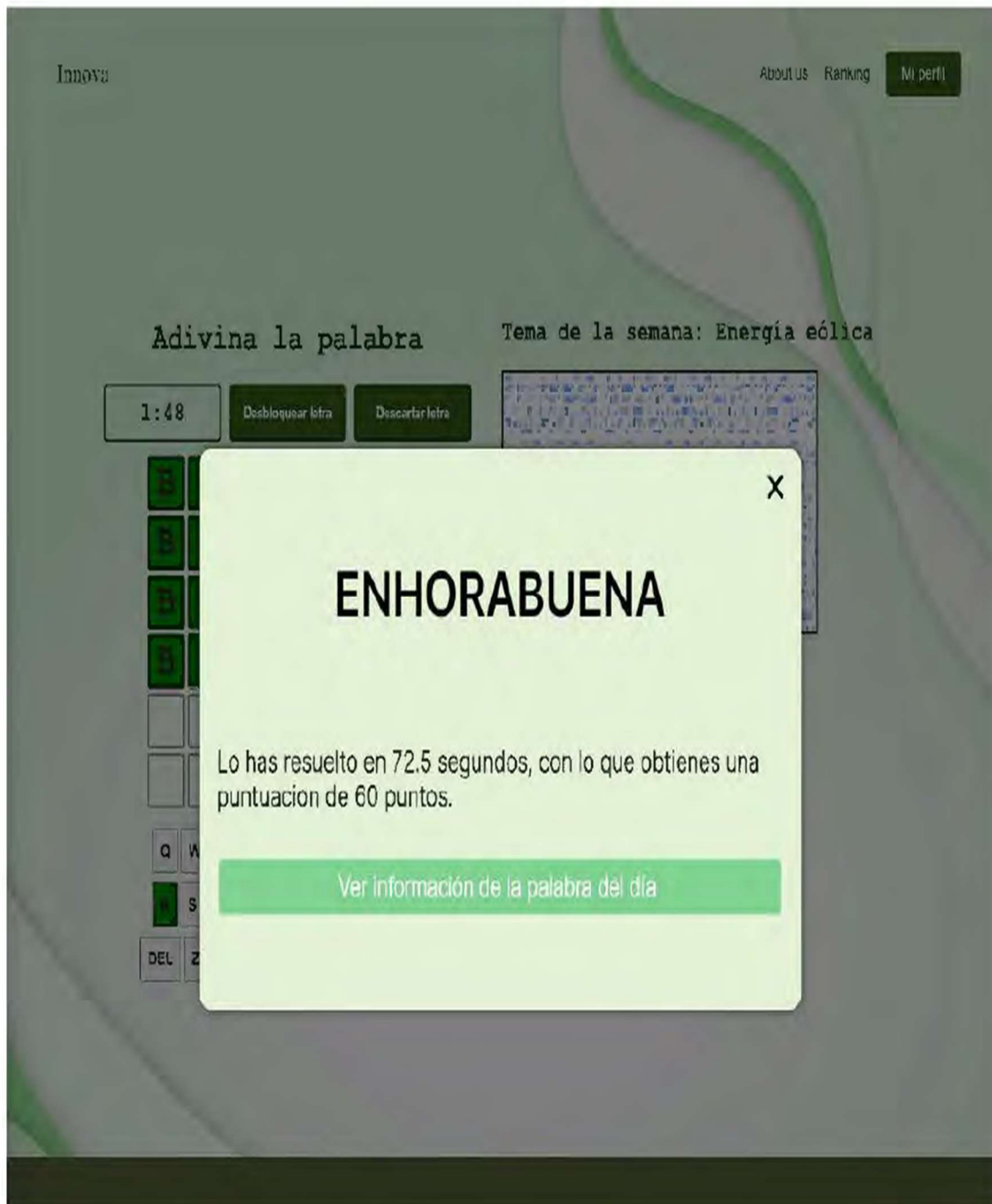


Figura 7: Reto diario logrado con una puntuación de 60 puntos.

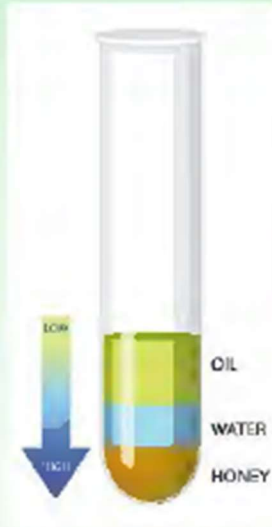
Información palabra del día

Tras haber completado el reto nos redigirá mediante un mensaje a la página con la información de la palabra diaria Figura.8.

DENSIDAD

¿Qué significa?

La densidad es una propiedad física que se define como la relación de masa dividida por el volumen (volumen/masa). Representa la cantidad de materia que ocupa un espacio (g/cm³). Se expresa por diferentes unidades (g/cm³, kg/m³, etc.) para facilitar su uso. Cuando una sustancia se encuentra en un espacio determinado, su densidad nos permite determinar y diferenciar sustancias que ocupan el espacio de una manera que no es obvia. Es un indicador que nos ayuda a determinar si una sustancia es más pesada o más ligera que otra sustancia.



¿Qué necesito saber sobre esta palabra?

¿Qué necesito saber sobre esta palabra?

Esta propiedad física nos ayuda a determinar si una sustancia es más pesada o más ligera que otra. La densidad es una propiedad física que se define como la relación de masa dividida por el volumen (volumen/masa). Representa la cantidad de materia que ocupa un espacio (g/cm³). Se expresa por diferentes unidades (g/cm³, kg/m³, etc.) para facilitar su uso. Cuando una sustancia se encuentra en un espacio determinado, su densidad nos permite determinar y diferenciar sustancias que ocupan el espacio de una manera que no es obvia. Es un indicador que nos ayuda a determinar si una sustancia es más pesada o más ligera que otra sustancia.

¿Cómo puedo aplicarlo en mi día a día?

En la vida cotidiana, la densidad tiene aplicaciones prácticas. Por ejemplo, al elegir un aceite para cocinar, es importante saber su densidad para asegurarse de que sea el adecuado. También, al elegir un líquido para beber, es importante saber su densidad para asegurarse de que sea el adecuado. La densidad es una propiedad física que se define como la relación de masa dividida por el volumen (volumen/masa). Representa la cantidad de materia que ocupa un espacio (g/cm³). Se expresa por diferentes unidades (g/cm³, kg/m³, etc.) para facilitar su uso. Cuando una sustancia se encuentra en un espacio determinado, su densidad nos permite determinar y diferenciar sustancias que ocupan el espacio de una manera que no es obvia. Es un indicador que nos ayuda a determinar si una sustancia es más pesada o más ligera que otra sustancia.



¿Quieres conseguir puntos extra?

¡Participa en el reto diario y gana puntos extra! Cada día hay un reto nuevo y puedes ganar puntos extra por completarlos. ¡Participa hoy mismo y gana puntos extra!

¡Gana puntos!

Figura 8: Visualización de la página que contiene la información de la palabra del reto diario.

Juego infinito

Si hemos completado el juego diario, se nos dará la opción de aumentar nuestra puntuación a través de un juego infinito, el cual sumará la mejor de nuestras puntuaciones a nuestra puntuación general Figura 9. Hay que destacar que este juego cuenta con un pad que permite jugar desde el móvil sin necesidad de teclado.

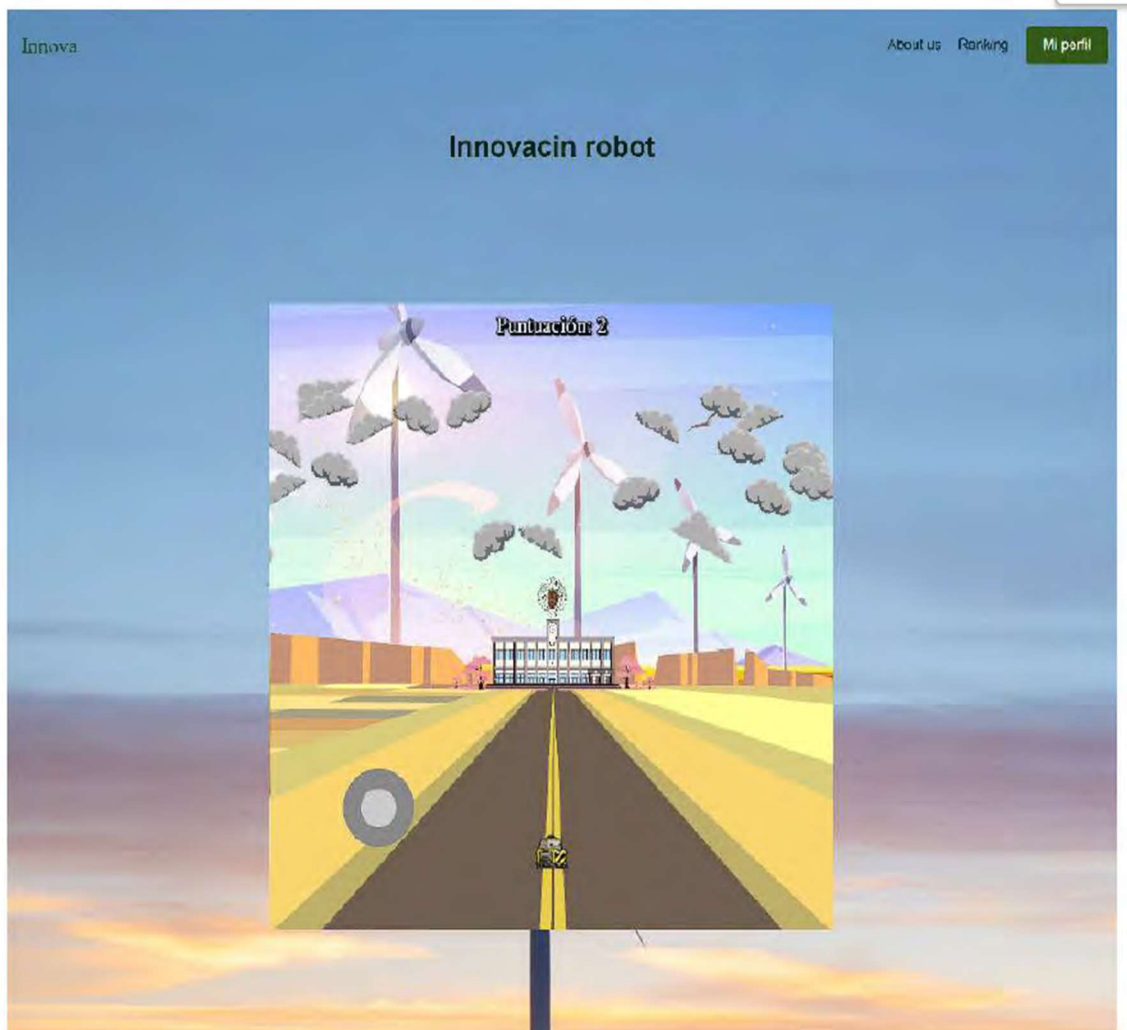


Figura 9: Juego infinito desde PC

Ranking

Desde nuestra cuenta, se puede acceder a un ranking donde podemos ver las puntuaciones con respecto a otros usuarios Figura 10 y a otras facultades Figura 11.



Figura 10: Ranking de usuarias/os

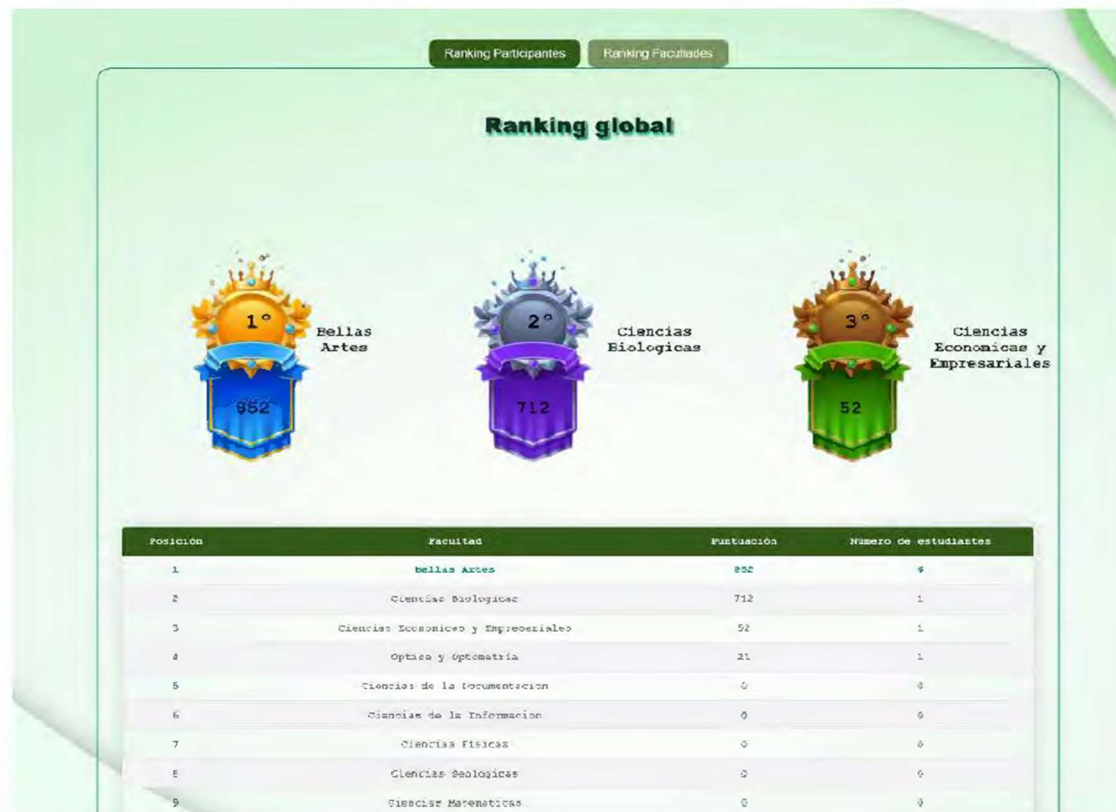


Figura 11: Ranking de las puntuaciones de todas las facultades

Acerca de

En esta página encontramos la información acerca del proyecto Figura 12 y 13.

The image shows a webpage layout for 'Innova'. At the top left is the logo 'Innova'. At the top right are the links 'About us' and 'Iniciar sesión'. The main content area has a light green background. On the left, there is a section titled 'NUESTRO OBJETIVO' with a sub-header 'Diviértete aprendiendo'. Below this is a paragraph of text in a typewriter font: 'Nuestra misión es educar y entretener a las personas a través de una experiencia interactiva y divertida que promueva un cambio positivo en nuestros hábitos energéticos diarios.' This is followed by another paragraph: 'Esta aplicación web tiene como objetivo principal la promoción del uso de energías renovables y la conciencia sobre su importancia en la protección del medio ambiente y la sostenibilidad del planeta.' To the right of this text is a vertical image of a hand holding a lightbulb where the filament is a globe of the Earth. Below the text on the left is an illustration of four people standing around various renewable energy symbols: a sun, a wind turbine, a water drop, and a solar panel. On the right side of the page, there is a section titled 'CÓMO FUNCIONA' with a sub-header 'Completa el reto para desbloquear la palabra del día'. Below this is a paragraph: 'Con Innova, hemos creado una plataforma que combina la emoción de los juegos con la importancia de la energía renovable.' This is followed by another paragraph: 'Nuestro juego te permitirá aprender los principales conceptos sobre los distintos tipos de energías renovables al mismo tiempo que te diviertes compitiendo por la primera posición en el ranking.'

Figura 12: Página completa Acerca de

NUESTRA ASPIRACIÓN

Conseguir un mundo más limpio

A través de nuestro minijuego, queremos inspirar a todos los estudiantes de la Universidad Complutense a tomar medidas concretas para reducir su huella de carbono y adoptar prácticas más sostenibles en sus vidas cotidianas. Creemos firmemente que cada pequeña acción cuenta y que juntos podemos marcar una gran diferencia en la protección de nuestro planeta para las generaciones futuras.



Únete a nosotros en nuestro viaje hacia un futuro más limpio y sostenible. ¡Descubre, aprende y diviértete mientras te conviertes en un defensor activo del uso de energías renovables!

Vamos!

Innova

[Inicio](#) [About Us](#)

© 2023 - 2024 Innova. Todos los derechos reservados.

[Política de Privacidad](#) | [Términos de Servicio](#)

Imágenes diseñadas por Freepik

Figura 13: Página completa acerca de

Desarrollo de una aplicación enfocada a la concienciación del ahorro energético para la comunidad universitaria

Development of an application focused on raising awareness of energy savings for the university community

Álvaro Gómez Morán - Grado en Ingeniería Informática
Daniel Prieto Remacha - Grado en Ingeniería Informática
Miguel Zayas Boíza - Grado en Ingeniería Informática

FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo de Fin de Grado en Ingeniería Informática

Madrid, 26 de mayo de 2024

Directores:

Sergio Bernabé García
María Yolanda Martínez Solana

Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a todas las personas que nos han apoyado a lo largo de este proyecto.

En primer lugar, a nuestros tutores, Sergio Bernabé García y María Yolanda Martínez Solana, por su inestimable guía, paciencia y sabios consejos durante todo el proceso de desarrollo de este trabajo.

A nuestros padres y familiares, por su apoyo incondicional y por creer en nosotros, proporcionándonos el ánimo necesario para seguir adelante. Finalmente, a todos los profesores de la facultad de ingeniería informática, cuyos conocimientos y enseñanzas han sido fundamentales para nuestra formación académica y profesional. Sin toda esta ayuda, este logro no habría sido posible. Gracias.

Índice general

Índice	I
Índice de figuras	V
Índice de tablas	VII
Resumen	IX
Abstract	XI
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Plan de trabajo	3
1.4. Organización de la memoria	5
2. Estado del Arte	7
2.1. Tecnologías para la creación del juego	7
2.1.1. Alternativas contempladas	7
2.1.2. Alternativa seleccionada: Phaser	8
2.2. Tecnologías para la creación de la web	9
2.2.1. Front-end	9
2.2.2. Back-end	9
2.2.3. Base de Datos	10
2.2.4. Entorno elegido	11
2.3. Aplicaciones web relacionadas	13

3. Análisis y diseño de la aplicación	17
3.1. Arquitectura general	17
3.2. Estructura de la Base de Datos	19
3.3. Análisis de requisitos	21
3.3.1. Requisitos funcionales	21
3.3.2. Requisitos no funcionales	23
3.4. Diseño de Casos de Uso	24
3.5. Prototipos de la Interfaz de Usuario	31
4. Estructuración del código web	37
4.1. Front-end o lado del cliente	39
4.1.1. Página principal	39
4.1.2. Carpeta “includes”	41
4.1.3. Carpeta “popups”	42
4.1.4. aboutUs.php	43
4.1.5. ranking.php	44
4.2. Infinity-game.php	46
4.3. Juego diario (index.php)	46
4.4. Back-end o Lado de servidor	48
4.4.1. Carpeta “ <i>DAO</i> ”	48
4.4.2. Carpeta “domains”	49
4.4.3. Carpeta “services”	50
4.4.4. Carpeta “controller”	51
4.4.5. Carpeta “Infinity-game”	53
4.4.6. Carpeta “wordle”	54
5. Implementación de la Aplicación	56
5.1. Arquitectura de la aplicación	56

5.2. Base de datos: construcción del modelo de datos	60
5.3. Implementación de funcionalidades: back-end	61
5.3.1. Juego Palabra Diaria	62
5.3.2. Juego Infinito	67
5.4. Rendimiento de la aplicación	73
6. Conclusiones y Trabajo Futuro	77
6.1. Conclusiones	77
6.2. Trabajo futuro	79
Bibliography	82
A. Introduction	83
A.1. Motivation	83
A.2. Objectives	84
A.3. Work Plan	85
A.4. Organization of this report	87
B. Conclusions and Future Work	89
B.1. Conclusions	89
B.2. Future Work	91
C. Flujo de uso y demo de la Aplicación	92
C.1. Registro	93
C.2. Inicio de sesión	94
C.3. Modificación de cuenta	94
C.4. Reto diario	96
C.5. Información palabra del día	98
C.6. Juego infinito	99
C.7. Ranking	100

C.8. About us	102
D. Aportaciones	104
D.1. Gómez Morán, Álvaro	104
D.2. Prieto Remacha, Daniel	106
D.3. Zayas Boíza, Miguel	107

Índice de figuras

1.1.	Diagrama de Gantt con planificación estimada	4
2.1.	Logo del juego World Rescue	14
2.2.	Logo del juego Climate Kids	15
2.3.	Logo del juego Energy Remix	16
3.1.	Diagrama del patrón Modelo-Vista-Controlador (MVC) con el patrón DAO .	19
3.2.	Diagrama relacional de la base de datos.	20
3.3.	Diagrama de Casos de Uso	25
3.4.	Mockup de la página principal previa al inicio de sesión del usuario	32
3.5.	Mockup de la página con el reto diario	33
3.6.	Mockup de la página que contiene el ranking	34
3.7.	Mockup de los popups de inicio de sesión y registro	35
3.8.	Mockups de los popups de perfil y editar perfil	36
4.1.	Estructura del proyecto representada en diagrama de zonas	38
4.2.	Estructura de carpetas del front-end	39
4.3.	Página principal una vez el usuario ha iniciado sesión	40
4.4.	Footer presente en todas las páginas de la aplicación	41
4.5.	Header presente en todas las páginas de la aplicación	41
4.6.	Ejemplo de ventanas modales para visualizar el perfil y cambiar la contraseña respectivamente	43
4.7.	Visualización de la puntuación del usuario y la tabla de ranking	45
4.8.	Visualización del reto diario durante su ejecución	47

5.1. Arquitectura empleada para el funcionamiento del servidor	57
5.2. Implementación final del reto diario	62
5.3. Visualización de la página que contiene la información de la palabra del reto diario	64
5.4. Sprite correspondiente al personaje principal del juego infinito	73
5.5. Tiempo de carga y recursos consumidos por defecto de la web.	74
5.6. Rendimiento durante la ejecución del reto diario	75
5.7. Rendimiento página juego infinito	76
A.1. Gantt Chart with Estimated Planning	86
C.1. Registro de nuevos usuarios	93
C.2. Inicio de sesión	94
C.3. Cambio de avatar en el usuario	95
C.4. Opciones de usuario	95
C.5. Reto diario con varios intentos	96
C.6. Reto diario logrado con una puntuación de 60 puntos	97
C.7. Visualización de la página que contiene la información de la palabra del reto diario	98
C.8. Juego inifinito desde PC	99
C.9. Ranking de usuarios	100
C.10. Ranking de las puntuaciones de todas las facultades	101
C.11. Página completa About Us	102
C.12. Página completa About Us	103

Índice de tablas

2.1. Comparación de las distintas tecnologías contempladas para el desarrollo del juego	8
2.2. Comparación de las distintas tecnologías contempladas para el desarrollo del front-end	11
2.3. Comparación de las distintas tecnologías contempladas para el desarrollo del back-end	12
2.4. Comparación de las distintas tecnologías contempladas para el desarrollo de la base de datos	13
2.5. Comparación de las páginas web similares con respecto a nuestra propuesta .	16
3.1. Caso de uso 01. Registrarse.	26
3.2. Caso de uso 02. Iniciar sesión.	27
3.3. Caso de uso 03. Cerrar sesión.	27
3.4. Caso de uso 04. Editar perfil.	28
3.5. Caso de uso 05. Eliminar cuenta.	28
3.6. Caso de uso 06. Reto diario.	29
3.7. Caso de uso 07. Juego extra.	30
3.8. Caso de uso 08. Ver ranking.	30

Resumen en castellano

En el ámbito de las energías renovables, la gran mayoría de las personas cuentan con un conocimiento general sobre qué son y para qué sirven, pero en muchos casos no cuentan con información suficiente, lo que les puede llevar a tener algunas conductas que no fomentan el ahorro energético y el cuidado del medio ambiente y que no tendrían si contaran con esa información. Una de las principales razones por las que esto puede ocurrir es el hecho de que la energía renovable es un tema de gran complejidad, lo que puede dar lugar a que la gente no se interese inicialmente por él.

Debido a esto, nuestro proyecto se centra en fomentar la conciencia y el conocimiento sobre las energías renovables de forma sencilla y entretenida. Para llevarlo a cabo hemos diseñado una aplicación web en la que, a través de un juego basado en desafíos diarios, los usuarios podrán aprender los conceptos más importantes sobre las energías renovables dedicando solo unos minutos al día.

El juego consiste en un desafío diario en el que los usuarios tendrán la posibilidad de descubrir un nuevo término relacionado con las energías renovables cada día. El usuario tendrá que descubrir la “palabra del día” basándose únicamente en unas reglas sencillas y su propia deducción. Una vez el usuario haya sido capaz de descubrir cuál era la palabra, se desplegará una pantalla en la que se mostrará información detallada sobre ese término, tal como su definición, funcionamiento o su relevancia en el contexto de las energías renovables.

Del mismo modo, también se mostrarán consejos o buenos hábitos relacionados con la palabra del día y que, sin requerir demasiado esfuerzo, muestren a los usuarios nuevas formas de aumentar el ahorro energético y de contribuir al cuidado del medio ambiente.

Para el desarrollo de esta aplicación web se ha necesitado una combinación de técnicas de programación web y diseño de interfaces de usuario, así como la colaboración con otras Facultades para tener disponibles los contenidos educativos necesario sobre energías renovables.

Nuestra iniciativa no busca solo promover la conciencia ambiental entre los usuarios, sino también difundir información relevante sobre las energías renovables de una forma sencilla y atractiva.

Palabras clave

Energías renovables, ecosistema, aplicación web, juego, educación, innova, ucm, informática, javascript, php.

Abstract

In the field of renewable energies, the vast majority of people have general knowledge about what they are and what they are for, but in many cases they do not have enough information, which can lead them to have some behaviors that do not encourage energy savings and care for the environment and that they would not have if they had that information. One of the main reasons why this may occur is the fact that renewable energy is a highly complex topic, which may result in people not being initially interested in it.

Because of this, our project focuses on promoting awareness and knowledge about renewable energies in a simple and entertaining way. To carry it out, we have designed a web application in which, through a game based on daily challenges, users will be able to learn the most important concepts about renewable energies spending just a few minutes a day.

The game consists of a daily challenge in which users will have the chance to discover a new term related to renewable energy every day. The user will have to discover the “word of the day” based only on some simple rules and their own deduction. Once the user has been able to discover what the word was, a screen will be displayed showing detailed information about that term, such as its definition, operation or its relevance in the context of renewable energies.

In the same way, tips or good habits related to the word of the day will also be shown and, without requiring too much effort, show users new ways to increase energy savings and contribute to caring for the environment.

The development of this web application has required a combination of web programming techniques and user interface design, as well as collaboration with other Faculties to have the necessary educational content on renewable energies available.

Our initiative not only seeks to promote environmental awareness among users, but also to disseminate relevant information about renewable energies in a simple and attractive way.

Keywords

Renewable energies, ecosystem, web application, game, education, Innova, computer science, javascript, php

Capítulo 1

Introducción

1.1. Motivación

Los efectos del cambio climático se han vuelto cada vez más presentes, pudiéndose apreciar un aumento a nivel mundial tanto en las emisiones de dióxido de carbono¹ como en la temperatura terrestre y oceánica². Estos efectos, en gran medida, son causados por los métodos que se utilizan actualmente para generar energía, basados en la quema de combustibles fósiles.

Durante los últimos años, se han encontrado nuevas formas de generar energía limpia por medio de energías renovables que no producen gases de efecto invernadero ni emisiones contaminantes. El problema es que, al ser un descubrimiento relativamente reciente, la mayor parte de las personas todavía no son completamente conscientes de la importancia que tienen y del papel que pueden desempeñar en el futuro.

Por eso hemos decidido, ayudándonos de los conocimientos que hemos adquirido durante el estudio de este grado, diseñar una aplicación web que ayude a las personas a comprender sin esfuerzo qué son las energías renovables, cómo nos pueden ayudar y qué pueden hacer para contribuir al mantenimiento del medio ambiente.

¹CO₂: emisiones mundiales 1995-2023

²Superficie terrestre y oceánica: variaciones en la temperatura 1880-2023

1.2. Objetivos

El principal propósito de este proyecto es diseñar una aplicación web que promueva la conciencia y el conocimiento sobre las energías renovables utilizando un método atractivo para los usuarios y que les permita adquirir los conocimientos propuestos sin tener que utilizar una gran cantidad de esfuerzo o tiempo. Para llegar a cumplir este objetivo vamos a determinar los principales puntos en los que queremos centrarnos.

Uno de los temas fundamentales a tener en cuenta inicialmente es la obtención de la información sobre energías renovables que se va a incluir en la aplicación. Con respecto a esto, es importante destacar que partimos de un Proyecto Innova-Docencia cuyo propósito es potenciar la divulgación y comunicación de los beneficios ecológicos y socioeconómicos del patrimonio natural, así como la integración de necesidades de conservación y uso sostenible de la biodiversidad en el desarrollo y aplicación de un importante número de políticas sectoriales, a fin de lograr un desarrollo justo, equitativo y duradero. Para conseguir esto, la Facultad de Ciencias de la Información ha llevado a cabo un estudio sobre las distintas energías renovables y su contribución en la lucha contra el cambio climático. Este estudio será el que utilizaremos para obtener la información clave que se mostrará en la aplicación, con el fin de concienciar de una forma clara y precisa a la comunidad universitaria.

Otro de los temas más importantes que hemos tratado es la forma en la que vamos a diseñar la aplicación para que los usuarios puedan aprender sobre las energías renovables de una forma amena y entretenida. Después de contrastar varias posibilidades, hemos decidido diseñar un juego basado en retos diarios con el que los usuarios puedan aprender un concepto nuevo cada día.

Relacionado con el punto anterior, también se diseñará un sistema de puntuaciones que se presentará en un ranking, mostrando tanto las puntuaciones individuales como las de cada Facultad en conjunto. De este modo se pretende atraer a la mayor cantidad de usuarios posibles y animarles a intentar realizar cada reto diario.

Para la implementación de la aplicación web, nuestro propósito será diseñar una web

con un diseño sencillo pero atractivo para el usuario, buscando la facilidad de uso desde el primer momento, y siendo perfectamente operativo, desde un ordenador hasta un móvil. Para ello se diseñarán varios mockups y se probarán distintas interfaces de usuarios hasta dar con la más adecuada.

Como objetivo adicional, dado que este proyecto está realizado en colaboración con la Facultad de Ciencias de la Información, queremos que este proyecto sirva como ejemplo de cómo la colaboración entre científicos, ingenieros y comunicadores puede promover la creación de nuevos modelos de enseñanza y de creación de contenidos científicos.

1.3. Plan de trabajo

Para el desarrollo de este proyecto hemos seguido una metodología ágil, manteniendo reuniones muy frecuentes entre los integrantes del grupo y realizando reuniones con los tutores de forma regular cada tres semanas con el objetivo de demostrar los avances conseguidos en el proyecto y detectar posibles desviaciones según lo planificado.

Para definir de una forma más visual el tiempo que se espera dedicar a cada tarea, hemos diseñado un diagrama de Gantt (ver Figura 1.1) que muestra las distintas tareas con respecto al tiempo total que contamos para terminar el proyecto.

El diagrama de Gantt no solo facilita la gestión del tiempo, sino que también ayuda a todos los involucrados a tener una comprensión clara de las etapas del proyecto y sus respectivos plazos. Esto asegura que el equipo pueda anticipar posibles problemas y ajustar los recursos según sea necesario para cumplir con los hitos establecidos. En resumen, el uso de esta herramienta ha sido fundamental para la planificación y ejecución eficiente del proyecto, proporcionando una estructura clara y bien definida para alcanzar nuestros objetivos.

Roadmap proyecto web Innova

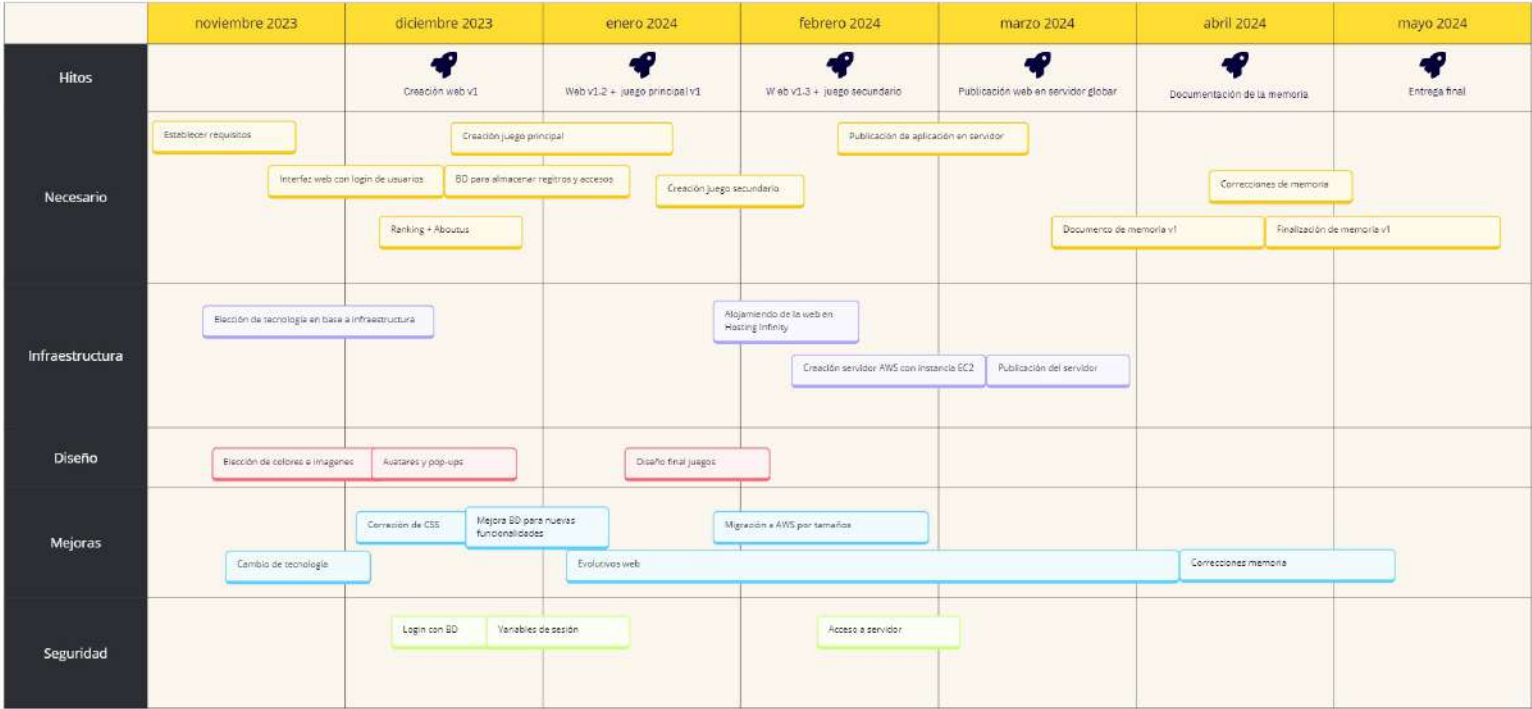


Figura 1.1: Diagrama de Gantt con planificación estimada

1.4. Organización de la memoria

La memoria del proyecto se estructura en 6 capítulos, cada uno de los cuales aborda uno de los aspectos relacionados con el desarrollo de la aplicación propuesta. A continuación se detalla la organización de cada capítulo:

- **Capítulo 1. Introducción:** este capítulo muestra la visión general del TFG, describiendo las motivaciones que nos han llevado al desarrollo de esta aplicación, los objetivos que pretendemos conseguir con este proyecto y el plan de trabajo que vamos a seguir para cumplir dichos objetivos.
- **Capítulo 2. Estado del Arte:** en este capítulo se realiza un análisis sobre las distintas tecnologías que hemos contemplado usar para el desarrollo de la aplicación, comparando las distintas ventajas e inconvenientes que presenta cada una de ellas y concluyendo con una explicación de la decisión final que hemos tomado para cada tecnología utilizada.
- **Capítulo 3. Diseño de la Aplicación:** en esta sección se detalla el proceso de diseño de la aplicación, en la que se incluye la arquitectura general, la estructura usada en la base de datos, el análisis de los distintos requisitos y el diseño de los casos de uso, así como la creación de los prototipos de la interfaz de usuario de la página web y del juego propuesto.
- **Capítulo 4. Estructuración del Código:** en este capítulo se detallará la organización jerárquica de las carpetas dentro del proyecto, explicando el propósito y el contenido de cada una de ellas. Se utilizará como base para entender mejor la implementación del proyecto que se detalla en el capítulo posterior.
- **Capítulo 5. Implementación de la Aplicación:** en este capítulo se detalla la implementación del diseño establecido en el Capítulo 3 mediante las tecnologías escogidas

en el Capítulo 2, analizando en profundidad todo el proceso de desarrollo seguido tanto en front-end como en back-end.

- **Capítulo 6. Conclusiones y Trabajo futuro:** en este último capítulo se incluyen las principales conclusiones que hemos obtenido tras la realización de este proyecto. Se comentan las distintas implicaciones derivadas del uso de la aplicación, así como posibles mejoras que se puedan incluir en próximas versiones

Capítulo 2

Estado del Arte

En esta sección se expondrán algunas de las tecnologías consideradas para la implementación del proyecto. A través de un análisis de las características de cada una de estas opciones, se proporcionará una explicación fundamentada de la elección final.

2.1. Tecnologías para la creación del juego

2.1.1. Alternativas contempladas

- **MelonJS**^[2]: es un framework de código abierto destinado al desarrollo de juegos en el entorno HTML5. Está diseñado para ofrecer una solución simple y efectiva, destacando por su enfoque modular y fácil aprendizaje. Cuenta con una gran compatibilidad, ya que opera como una biblioteca independiente que solo necesita un navegador compatible con HTML5. Un aspecto a tener en cuenta es que al estar enfocado para el desarrollo de juegos simples y enfocado a personas con poca experiencia en este campo, no cuenta con tantas herramientas como otras opciones.
- **PixiJS**^[8]: se trata de un framework de código abierto en Javascript, basado en HTML5. PixiJS utiliza *WebGL*¹ y, de manera opcional, utiliza la presentación en lienzo de HTML5 para ofrecer un rendimiento óptimo en una amplia gama de dispositivos. Destaca por su capacidad para facilitar el desarrollo de experiencias interactivas

¹Página oficial de *WebGL*

y visuales envolventes. Proporciona herramientas para la carga eficiente de recursos, así como un sistema de animación robusto que simplifica la creación de movimientos fluidos y transiciones visuales en los juegos.

- **Phaser**^[5]: se considera un framework muy accesible y versátil para el desarrollo de videojuegos. Se trata de un motor de juegos 2D construido en javascript, compatible tanto con *WebGL* como con *Canvas* de manera automática. Ofrece una amplia gama de características prácticas, como funciones de física, sprites, animaciones o incluso elementos como cámaras para implementar juegos con scroll o soporte de sonidos.

2.1.2. Alternativa seleccionada: Phaser

	MelonJS	PixiJS	Phaser
Curva de aprendizaje moderada	✓	✓	X
Documentación completa	X	✓	✓
Comunidad activa	X	✓	✓
Muy buen rendimiento en navegadores	X	X	✓
Físicas integradas	✓	X	✓
Soporte para animaciones	✓	✓	✓

Tabla 2.1: Comparación de las distintas tecnologías contempladas para el desarrollo del juego

La elección de Phaser como plataforma de desarrollo para el proyecto se basó principalmente en la experiencia previa que teníamos algunos de los integrantes del equipo con esta tecnología. Aunque al principio consideramos MelonJS o PixiJS como opciones viables, después de analizar más detenidamente las características de cada una de las tecnologías, como se muestra en la Tabla 2.1, decidimos optar por Phaser. Es cierto que PixiJS también ofrece una curva de aprendizaje moderada y una documentación completa, pero un aspecto muy importante para nuestro proyecto es conseguir el mejor rendimiento posible en los navegadores, y por este lado PixiJS se quedaba atrás. Por otro lado, MelonJS nos resultó muy atractivo al principio debido a que era el que tenía una menor curva de aprendizaje,

pero dado que parte del equipo ya contaba con experiencia en el uso de Phaser, decidimos que no valía la pena renunciar a las ventajas que este aportaba si lo único que ganábamos al hacerlo era una pequeña disminución en la dificultad de aprendizaje.

2.2. Tecnologías para la creación de la web

2.2.1. Front-end

- **React**^[19]: es una biblioteca de JavaScript ampliamente utilizada para construir interfaces de usuario interactivas y eficientes. Su enfoque se centra en la creación de componentes reutilizables, lo que facilita la construcción y mantenimiento de aplicaciones complejas. React cuenta con un modelo de representación virtual que logra un rendimiento superior. Además, utiliza un paradigma de programación declarativa que permite gestionar los cambios de una forma más eficiente.
- **Angular**^[9]: es un framework de código abierto basado en Typescript². Una de sus características fundamentales es su enfoque en la arquitectura de una página (SPA³), que permite la carga de contenido de una forma eficiente sin recargar la página completa. Del mismo modo que en el caso anterior, este framework permite la creación de componentes reutilizables y cuenta con directivas que simplifican la manipulación del DOM⁴.

2.2.2. Back-end

- **Python (framework Django)**^[21]: python con Django es una combinación para el desarrollo web que ha ganado popularidad debido a su simplicidad y productividad. Este framework utiliza el patrón Modelo-Vista-Plantilla y proporciona un conjunto completo de herramientas y características listas para usar, lo que acelera el desarrollo y facilita la construcción de aplicaciones web robustas.

²Página oficial de Typescript

³SPA: Single-Page Application

⁴DOM: Document Object Model

- **Java (framework Spring)**[¹¹]: spring es un framework que utiliza el patrón Modelo-Vista-Controlador. Ofrece una función de inyección de dependencias que permite que los objetos definan sus propias dependencias que el contenedor de Spring luego inyecta en ellos. Esto permite crear aplicaciones modulares que constan de componentes poco acoplados ideales para aplicaciones web.
- **PHP puro (patrón DAO)**[¹³]: el lenguaje de programación PHP ha sido diseñado específicamente para el desarrollo web, lo que lo convierte en una opción natural para la construcción de aplicaciones en este ámbito. Al optar por usar PHP puro se prescinde de frameworks y librerías externas, lo que resulta en un código mucho más ligero y fácil de mantener. Además, la implementación del patrón DAO facilita el mantenimiento y escalabilidad de la aplicación, al permitir cambios en la capa de acceso a datos sin afectar otras partes del sistema. Adicionalmente, la encapsulación de la lógica de acceso a datos en objetos DAO facilita la reutilización del código en diferentes partes de la aplicación.

2.2.3. Base de Datos

- **MySQL**[⁶]: es un sistema de gestión de bases de datos puramente relacional. Proporciona una plataforma robusta y escalable para el almacenamiento, gestión y recuperación de datos de manera eficiente y confiable. MySQL destaca por su facilidad de uso, rendimiento optimizado, alta disponibilidad y seguridad avanzada. Es conocido por su velocidad y capacidad de manejar grandes volúmenes de datos, así como por su soporte multiplataforma, lo que lo hace compatible con una amplia gama de sistemas operativos y entornos de desarrollo.
- **PostgreSQL**[¹⁸]: es un sistema de gestión de bases de datos relacionales que sobresale por su capacidad de manejar tanto datos convencionales como objetos complejos. Además de admitir todos los tipos estándar de MySQL, incorpora tipos de datos más avanzados que permiten almacenar y manipular información de manera más sofisticada.

da. Una de sus características más destacadas es su control de concurrencia MVCC⁵, que garantiza la consistencia de los datos al permitir la creación de copias duplicadas de registros para realizar lecturas y actualizaciones de manera segura y concurrente.

- **MongoDB**[²⁰]: es un sistema de gestión de bases de datos NoSQL, diseñado para el almacenamiento y recuperación eficiente de datos no estructurados o semiestructurados. Cuenta con un modelo de datos flexible basado en JSON, lo que le permite almacenar y consultar datos de forma dinámica y escalable. Este sistema incluye un lenguaje de consulta rico y expresivo, indexación avanzada para consultas rápidas y soporte para operaciones complejas.

2.2.4. Entorno elegido

Front-end

	React	Angular	JavaScript puro
Curva de aprendizaje moderada	X	X	✓
Ligereza	✓	X	✓
Comunidad activa	✓	✓	✓
Tamaño	✓	X	✓
Flexibilidad	✓	X	✓

Tabla 2.2: Comparación de las distintas tecnologías contempladas para el desarrollo del front-end

Para la implementación del front-end, en un primer momento pensamos en utilizar React o Angular, e incluso llegamos a la conclusión de que React sería la mejor opción después de analizar ambas opciones, debido a la flexibilidad en el desarrollo y la integración con otra librería que permite [1]. Sin embargo, teniendo en cuenta las características de la aplicación que vamos a desarrollar, también quisimos valorar el uso de JavaScript puro en lugar de usar un framework.

Tras analizar las alternativas disponibles, decidimos que la mejor opción era emplear Javascript prescindiendo del uso de cualquier framework. Esta elección se basa en distintos

⁵MVCC: Multiversion Concurrency Control

aspectos. En primer lugar, ninguno de los miembros de nuestro equipo contaba con experiencia previa en el manejo de frameworks de este tipo, por lo que considerando la significativa curva de aprendizaje asociada a estos frameworks, concluimos que no sería buena idea aventurarse a usarlos. Además, dadas las especificaciones de la aplicación, no creímos que fuera realmente necesario recurrir a un framework para alcanzar los objetivos establecidos. Otro aspecto que nos llevó a optar por Javascript fue el hecho de que las aplicaciones desarrolladas con esta tecnología tienden a presentar un bundle inicial más reducido en comparación con aquellas que hacen uso de frameworks, proporcionando así mejoras en los tiempos de carga, lo que consideramos un aspecto muy beneficioso para la experiencia del usuario.

Back-end

	Python(Django)	Java(Spring)	PHP puro
Curva de aprendizaje moderada	✓	X	✓
Comunidad activa	✓	✓	✓
Escalabilidad	✓	✓	✓
Experiencia previa	X	X	✓
Rendimiento en aplicaciones web	✓	X	✓

Tabla 2.3: *Comparación de las distintas tecnologías contempladas para el desarrollo del back-end*

Para la implementación del backend, al igual que en el caso anterior, revisamos las posibles alternativas y cuál podría ser la mejor para nuestro proyecto[[22](#)]. Aunque Python parecía una opción prometedora, como solo uno de los integrantes del grupo contaba con experiencia en este lenguaje, al final decidimos descartar esa opción. Además, dado que todo el equipo poseía experiencia previa en el desarrollo de aplicaciones web utilizando PHP, esta tecnología nos parecía una opción más familiar y cómoda. Otro factor determinante a favor del uso de PHP fue la falta de experiencia en el uso del resto de frameworks contemplados, ya que cualquiera de ellos contaba con una gran curva de aprendizaje que habría requerido demasiado tiempo. Tras investigar un poco en profundidad los mismos, PHP puro con el

patrón DAO parecía la opción más adecuada, considerando las capacidades que ofrece.

Base de Datos

	MySQL	PostgreSQL	MongoDB
Curva de aprendizaje moderada	✓	✓	X
Experiencia previa	✓	X	✓
Comunidad activa	✓	✓	✓
Tasa de transacciones elevada	✓	✓	X
Seguridad de los datos	✓	✓	X

Tabla 2.4: *Comparación de las distintas tecnologías contempladas para el desarrollo de la base de datos*

Por último, para la gestión de la base de datos, decidimos optar por el uso de MySQL. Descartamos MongoDB debido a que, dadas las características específicas de la base de datos contemplada, llegamos a la conclusión de que una base de datos SQL sería más adecuada, ya que permite manejar una tasa alta de transacciones que puede ser muy útil para una aplicación de estas características y es más seguro que MongoDB a la hora de manejar datos confidenciales [3]. Aunque PostgreSQL era una opción tentadora, consideramos que, dado el hecho de que todos los integrantes del grupo tenían experiencia previa con MySQL y ninguno con PostgreSQL, así como el reconocimiento de una curva de aprendizaje mayor con PostgreSQL, era más prudente optar por MySQL.

2.3. Aplicaciones web relacionadas

En este apartado vamos a analizar algunas de las páginas web que hemos visitado para buscar referencias antes de comenzar el desarrollo del diseño de nuestra aplicación. Hemos intentado comprender los distintos enfoques que siguen cada una de estas páginas para así poder identificar las áreas que podríamos aprovechar para mejorar la experiencia del usuario en nuestra aplicación.

World Rescue

Esta es una plataforma educativa que combina juegos interactivos con contenido educativo sobre energías renovables y sostenibilidad. Está desarrollado por el Programa de las Naciones Unidas para el Medio Ambiente (PNUMA) y es un juego que se centra en enseñar a los jugadores sobre los desafíos globales relacionados con el cambio climático y cómo pueden afectar al medio ambiente las diferentes decisiones que se tomen.

Consideramos esta plataforma un muy buen ejemplo de cómo se puede conseguir aprender sobre temas tan importantes como este sin necesidad de dejar a un lado la diversión. Sin embargo, en relación con nuestro proyecto, esta plataforma ha desarrollado un juego muy extenso que incluso hace falta descargar y que los usuarios puedan utilizar durante mucho tiempo. En nuestro caso, buscamos desarrollar una aplicación a la que cualquier usuario pueda acceder muy fácilmente y en la que solo utilizando unos minutos al día, pueda aprender sobre conceptos clave relacionados con las energías renovables.



Figura 2.1: *Logo del juego World Rescue*

Climate Kids

Climate Kids es una plataforma interactiva diseñada para educar a los niños sobre el cambio climático de una forma divertida y accesible. Ofrece una gran variedad de juegos, videos y recursos educativos relacionados con temas como el calentamiento global, la conservación de energía, la biodiversidad y la meteorología.

Esta plataforma representa un modelo similar al tipo de aplicación que queremos desarro-

llar, centrándose en juegos de fácil acceso y corta duración. Sin embargo, nosotros buscamos que nuestra aplicación, además de incluir eso, también incentive a los usuarios a visitar la plataforma diariamente, facilitando así el aprendizaje continuo de nuevos conceptos.



Figura 2.2: Logo del juego *Climate Kids*

Energy Remix

Esta web es una plataforma interactiva creada por *Enel Green Power* que ofrece un juego interactivo centrado en energías renovables y sostenibilidad ambiental. A través de una experiencia de juegos inmersiva, los jugadores pueden explorar diferentes aspectos relacionados con la generación de recursos naturales y la mitigación del cambio. El juego combina secciones con preguntas tipo test y secciones con pequeños terrenos donde los jugadores deberán decidir qué tipo de energía renovable es más conveniente extraer.

Esta plataforma tiene un gran potencial y dentro de las distintas páginas que hemos visitado es una de las más profesionales y con más funcionalidades. Sin embargo, como hemos mencionado anteriormente, a nuestra aplicación web queremos darle un enfoque distinto, de forma que en vez de tener un juego muy extenso que el jugador pueda jugar durante mucho tiempo, en nuestro caso buscamos desarrollar un juego muy corto, pero que anime al jugador a seguir jugando todos los días.



Figura 2.3: *Logo del juego Energy Remix*

A continuación se muestra una tabla con la que se pretende comparar de una forma gráfica las distintas aplicaciones web que hemos analizado y la aplicación web que pretendemos desarrollar para este proyecto:

	World Rescue	Climate Kids	Energy Remix	Nuestro proyecto
Facilidad de acceso a la aplicación	X	✓	✓	✓
Tiempo que necesita el usuario	X	✓	X	✓
Complejidad del juego suficiente	✓	X	✓	✓
Uso continuo de la aplicación	X	X	X	✓

Tabla 2.5: *Comparación de las páginas web similares con respecto a nuestra propuesta*

Capítulo 3

Análisis y diseño de la aplicación

En este capítulo explicaremos con detalle todos los pasos que hemos seguido a la hora de diseñar la aplicación. Partiremos de la definición de la arquitectura general de la aplicación, donde mostraremos las distintas opciones que contemplamos hasta llegar a la arquitectura que tomamos como definitiva, y mostraremos los distintos componentes que forman esta arquitectura. Seguidamente, trataremos temas como la estructura de la base de datos elegida para el proyecto y las razones que nos han llevado a escogerla, el análisis de requisitos del sistema y las decisiones que tomamos en consecuencia a los mismos, y el diseño de los casos de uso que servirán como guía para la funcionalidad de la aplicación. Además, también mostraremos los distintos prototipos de la interfaz de usuario creados, y las modificaciones que hemos realizado para cumplir todos los nuevos requisitos que han aparecido a lo largo de la creación del proyecto.

3.1. Arquitectura general

Después de evaluar las distintas opciones que podrían ser útiles para desarrollar este proyecto, para la arquitectura de la aplicación nos decantamos por seguir el patrón Modelo-Vista-Controlador (MVC¹), ya que este nos permitía tener una separación clara entre las distintas partes del proyecto y facilitaba mucho el trabajo a la hora de añadir nuevas funcionalidades, ya fueran relacionadas con el front-end o con el back-end.

¹MVC: Modelo-Vista-Controlador

Por otro lado, como método de comunicación con la base de datos, optamos por seguir el patrón Data Access Object (DAO²), ya que nos permitía tener en una misma zona y de forma muy controlada todas las operaciones realizadas en la base de datos. Además, en combinación con el patrón MVC, permite crear un entorno muy intuitivo a la hora de trabajar y con una gran escalabilidad.

De esta forma, la arquitectura final a la que llegamos consta de las siguientes capas:

- **Modelo:** esta capa se encarga de representar la lógica de negocio, incluyendo la creación de clases para representar las entidades de la aplicación y el tratamiento de información de la base de datos. Todos los accesos a la base de datos se hacen por medio de clases DAO que sirven como intermediarios entre el modelo y la capa de persistencia de datos.
- **Vista:** en esta capa hemos incluido todo lo relacionado con la presentación de la información y las interacciones de la aplicación con el usuario. Esta capa se encarga de mostrar los datos proporcionados por el modelo y envía las acciones del usuario al controlador.
- **Controlador:** esta capa actúa como intermediario entre el modelo y la vista. En esta capa hemos incluido todas las clases que se encargan de procesar las solicitudes del usuario, interactuar con el modelo y actualizar la vista según dicha interacción.

²DAO: Data Access Object

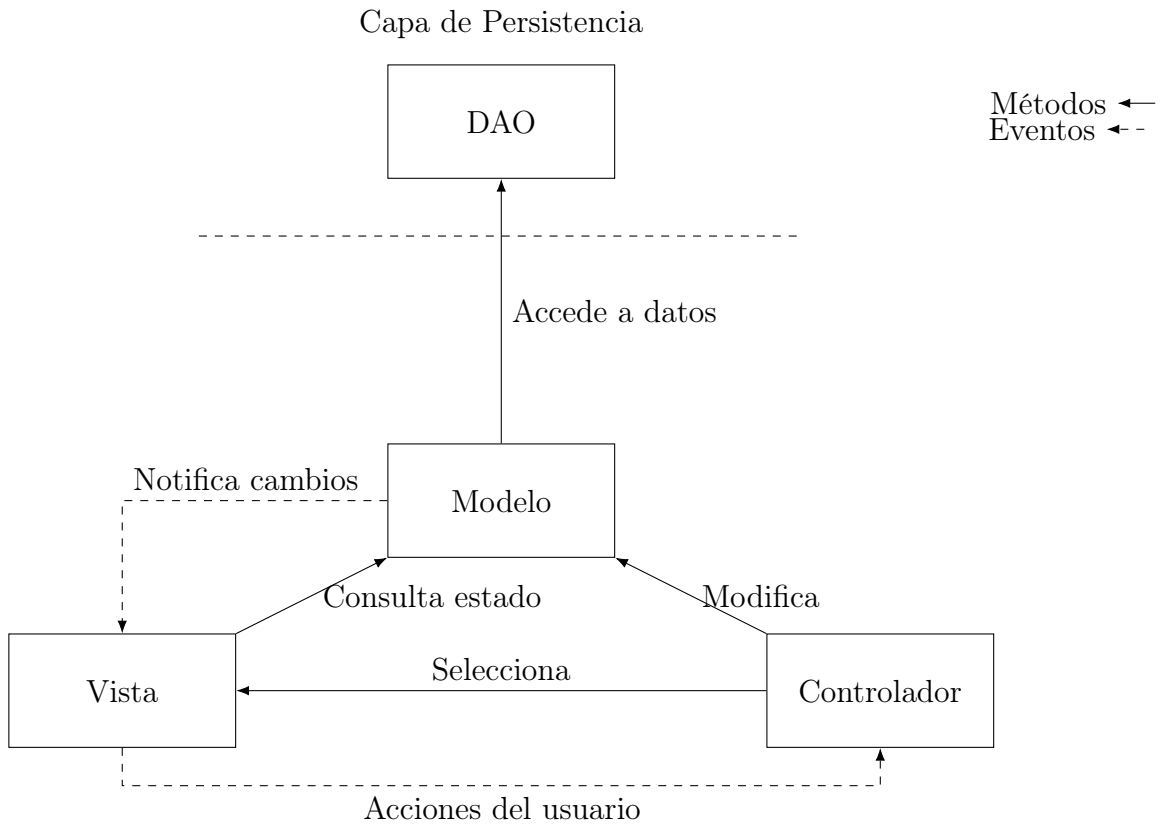


Figura 3.1: Diagrama del patrón Modelo-Vista-Controlador (MVC) con el patrón DAO

3.2. Estructura de la Base de Datos

Después de analizar los distintos requisitos que presentaba el proyecto, hemos intentado diseñar una base de datos que, a pesar de ser simple, cumpla con todos los requerimientos y permita recuperar los datos de una forma eficiente.

De esta forma, nuestra base de datos final está formada por las siguientes tablas:

- **Tabla Facultad:** esta tabla almacena la información sobre las distintas Facultades, incluyendo un identificador de la Facultad, el número total de puntos sumado por todos los estudiantes de dicha Facultad y el número total de estudiantes de esa Facultad que están participando en el juego.
- **Tabla Usuario:** esta tabla recoge toda la información sobre los usuarios del sistema,

incluyendo, entre otros campos, su nombre de usuario, dirección de correo electrónico, una contraseña cifrada y la cantidad de puntos acumulada por el usuario tanto en el reto diario como en el juego extra. Además, para controlar los accesos del usuario al reto diario y al juego extra, hemos añadido el campo “winner” para controlar si el usuario ha ganado el reto diario, y el campo “lastPlay” para guardar la fecha del último reto diario que realizó el usuario. También se incluye un campo con la Facultad a la que pertenece el usuario, lo que nos permite relacionar esta tabla con la Facultad mencionada anteriormente. Por último, contiene un campo “avatar” que contendrá la ruta a la imagen que haya elegido el usuario como foto de perfil.

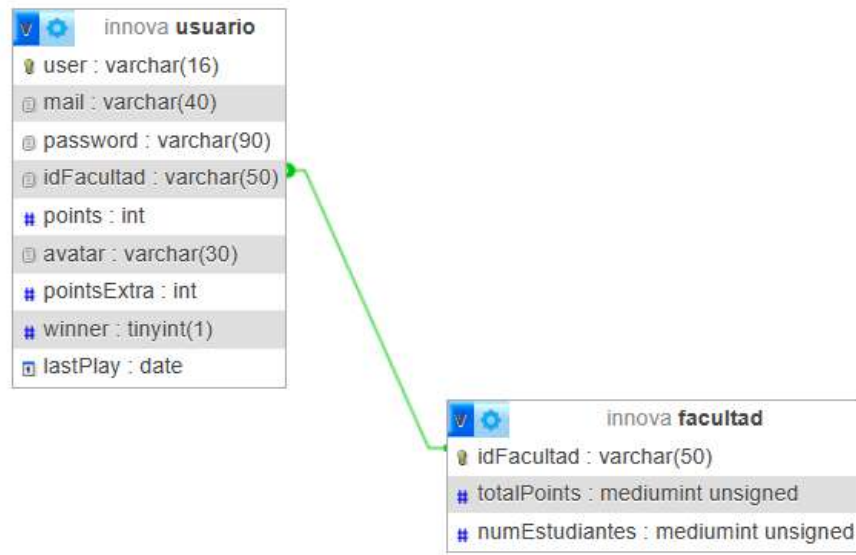


Figura 3.2: Diagrama relacional de la base de datos.

Además de estas tablas, serán necesarios disparadores en la tabla Usuario para automatizar ciertas acciones y mantener actualizada la información en relación con las Facultades.

3.3. Análisis de requisitos

Para clasificar los distintos requisitos que buscábamos que cumpliera nuestro proyecto, los hemos separado en dos grupos principales: requisitos funcionales y requisitos no funcionales.

3.3.1. Requisitos funcionales

■ Registro de usuario:

- Se permitirá registrarse a cualquier usuario que todavía no disponga de cuenta y que cuente con una dirección de correo electrónico de la Universidad Complutense.
- Para registrarse será suficiente con que se indique el nombre de usuario, el correo electrónico y la Facultad a la que se pertenece.
- Para registrarse será necesario añadir una contraseña que contenga por lo menos 8 caracteres de los cuales uno tendrá que ser una letra mayúscula, otro una letra minúscula y otro un carácter especial.

■ Inicio de sesión:

- Se permitirá que los usuarios registrados inicien sesión en sus cuentas utilizando su nombre de usuario y contraseña.
- Se verificarán las credenciales del usuario antes de darle acceso a la cuenta.

■ Perfil de usuario:

- Se mostrará la información del usuario, incluyendo el nombre de usuario, la Facultad a la que pertenece y la puntuación actual que tiene.
- Se proporcionarán opciones para que el usuario pueda editar su nombre de perfil y usuario y pueda elegir entre los distintos avatares predefinidos que se proporcionan.

- Se proporcionará la opción de eliminar la cuenta, siempre comprobando la identidad del usuario antes de hacerlo.

■ **Juego diario:**

- Los usuarios deberán poder acceder al juego diario y participar en él.
- Cada día se mostrará una palabra nueva relacionada con las energías renovables que los usuarios tendrán que adivinar.
- Los usuarios tendrán un número limitado de intentos para adivinar la palabra diaria.
- Después de cada intento se proporcionará retroalimentación sobre la palabra adivinada, indicando las letras correctas en la posición correcta y las letras correctas en la posición incorrecta. Además, se mostrará una imagen pixelada, de forma que después de cada intento de adivinar la palabra, la imagen irá mejorando su calidad.
- Existirán dos botones para proporcionar una ayuda a cambio de perder unos segundos del temporizador. Uno de los botones permitirá desbloquear una letra aleatoria de la palabra y el otro botón permitirá descartar del teclado una letra que no esté en la palabra diaria.

■ **Juego extra:**

- Los usuarios solo podrán tener acceso a este juego en el caso de haber ganado el reto diario.
- Los usuarios podrán repetir este juego tantas veces como quieran.
- La mejor puntuación obtenida de entre todos los intentos será la que se suma a la puntuación base del usuario.
- Cuando el usuario finalice un intento tendrá disponible un botón para realizar otro intento directamente.

- **Sistema de ranking:**

- Se mantendrá un ranking de usuarios basado en su rendimiento en el juego diario y su mejor puntuación en el juego extra.
- Los usuarios conseguirán una mayor puntuación cuanto más rápido adivinen la palabra diaria y menos utilicen los botones de ayudas.
- Se implementará un ranking por Facultades que mostrará el rendimiento total de los usuarios asociados a cada Facultad.

3.3.2. Requisitos no funcionales

- **Rendimiento:**

- La aplicación deberá funcionar de forma eficiente, respondiendo rápidamente a las interacciones del usuario y reduciendo todo lo posible los tiempos de carga y procesamiento.

- **Seguridad:**

- **Control de acceso**

- Se controlará que solo las personas con una cuenta de la Universidad Complutense tengan la posibilidad de registrarse.

- **Requisitos de contraseña**

- Se requerirá una contraseña que cumpla ciertos criterios para poder registrarse en la aplicación.
- La contraseña tendrá que cumplir las recomendaciones de seguridad [10].

- **Cifrado de contraseñas**

- Las contraseñas se guardarán cifradas en la base de datos utilizando la función `password_hash()` con el algoritmo `PASSWORD_BCRYPT`[7].

- **Usabilidad:**
 - **Diseño de interfaz**
 - La interfaz de usuario deberá tener un diseño atractivo y ser intuitiva, facilitando cualquier interacción del usuario.
 - **Compatibilidad de dispositivos**
 - La aplicación se podrá visualizar tanto en pantallas de ordenador como en dispositivos móviles.
 - **Dificultad para recordar**
 - Las contraseñas pueden ser más difíciles de recordar para los usuarios, lo que puede llevar a problemas de usabilidad.
 - **Posibles frustraciones de los usuarios**
 - Los usuarios podrían sentirse frustrados al tener que cumplir los criterios de las contraseñas.

3.4. Diseño de Casos de Uso

En esta sección presentaremos todos los casos de uso que hemos diseñado con el fin de que la aplicación lleve a cabo todas las funcionalidades necesarias. Primero mostraremos un diagrama con las distintas relaciones entre los usuarios y los casos de uso creados, para luego pasar a explicar de forma detallada cada uno de ellos.

Nuestro diagrama incluye todos los casos de uso necesarios para gestionar las cuentas de los usuarios, tratando tanto el registro de cuentas como el inicio y cierre de sesión.

También se incluyen casos de uso para funcionalidades típicas de este tipo de aplicaciones web, como la edición del perfil o la eliminación de la cuenta del usuario. Cabe destacar que tanto para el inicio de sesión como para la eliminación de una cuenta se incluye un caso de uso que se encarga de verificar las credenciales del usuario.

Por último se añaden casos de uso para funcionalidades más concretas de nuestro proyecto, incluyendo el reto diario y la visualización del ranking. En relación con el reto diario, hemos decidido dividir el sistema del juego en tres casos de uso: el primero se encargará de gestionar el reto de la palabra diaria, el segundo se encargará de mostrar toda la información relacionada con la palabra diaria una vez se haya adivinado, y el tercero gestionará el juego extra, al cual se permitirá jugar una vez el usuario haya completado el reto diario.

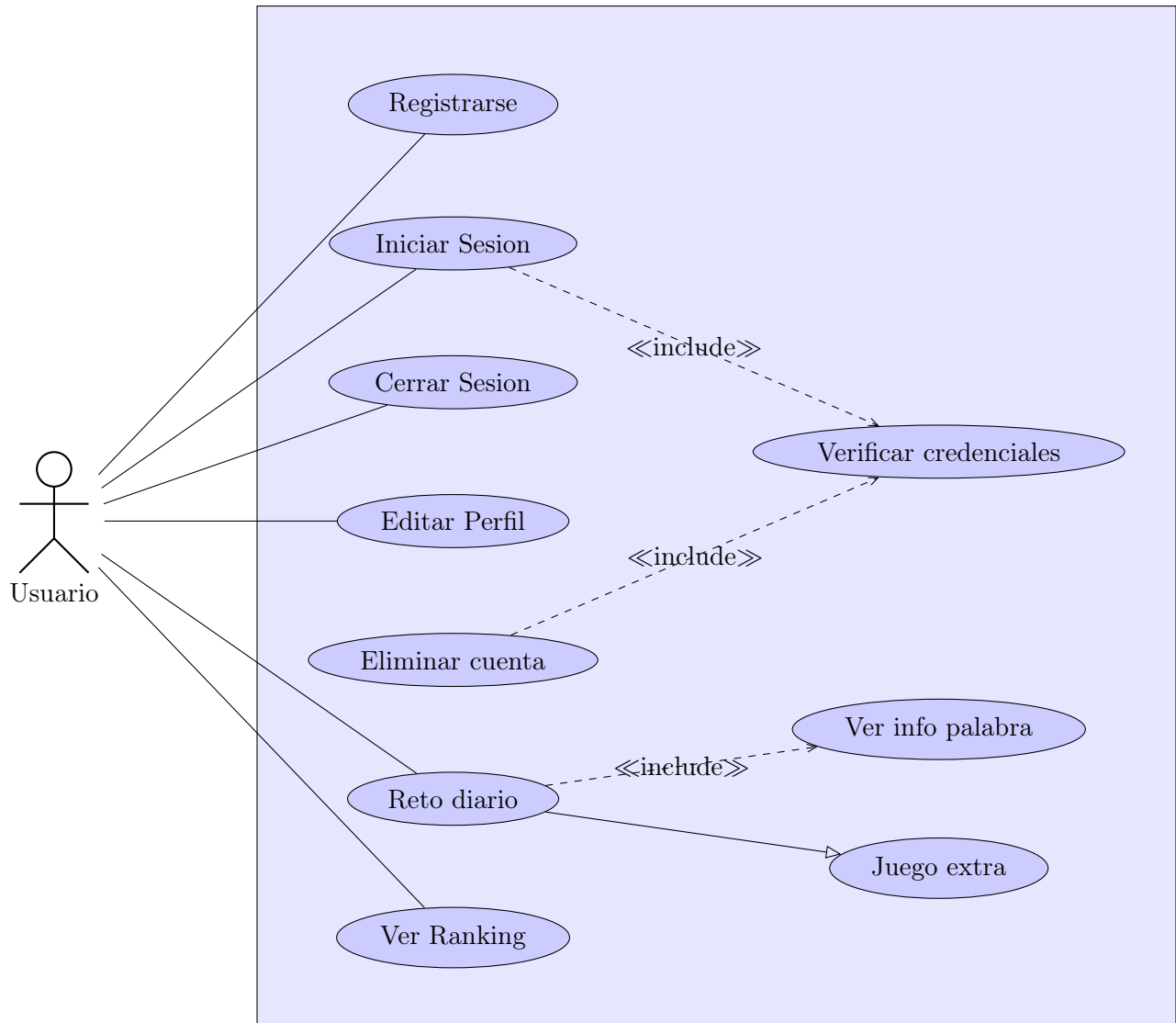


Figura 3.3: *Diagrama de Casos de Uso*

01 Registrarse	
Actores	Usuario.
Descripción	El usuario se crea una cuenta usando su correo de la Universidad Complutense.
Precondición	El usuario no puede crear la cuenta usando un correo que ya esté registrado.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Iniciar sesión». 2. Se abre una ventana en cuya parte inferior contiene un botón «Regístrate aquí». 3. Al pulsar el botón se abre una ventana con todos los campos que hace falta introducir para crear la cuenta (nombre de usuario, correo electrónico, contraseña, confirmación, contraseña y Facultad a la que pertenece). 4. Una vez rellenados todos los campos, el usuario pulsa en el botón «Registrarse» para terminar el proceso.
Postcondición	Se guarda en un registro de la base de datos la nueva cuenta con todos los datos.
Excepciones	<ol style="list-style-type: none"> 1. Si el formato del correo electrónico introducido no pertenece a la Universidad Complutense, se informará al usuario del error y no se permitirá finalizar el registro hasta cambiarlo por un correo válido. 2. Si la contraseña introducida no cumple con todos los requisitos necesarios, se informará al usuario y no se permitirá finalizar el registro hasta introducir una contraseña válida.
Comentarios	

Tabla 3.1: *Caso de uso 01. Registrarse.*

02 Iniciar sesión	
Actores	Usuario.
Descripción	El usuario accede a su cuenta usando su nombre de usuario y contraseña
Precondición	El usuario debe haberse registrado previamente.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Iniciar sesión». 2. Se abre una ventana en la que tendrá que introducir el nombre de usuario y la contraseña. 3. Una vez introducidos los datos, se pulsa el botón «Iniciar sesión». De esta forma se pasa a verificar las credenciales y si son correctas se procede al inicio de sesión.
Postcondición	Se crea una sesión para controlar los estados.
Excepciones	Si no se han podido verificar las credenciales correctamente, no se permite el acceso y se notifica al usuario
Comentarios	

Tabla 3.2: *Caso de uso 02. Iniciar sesión.*

03 Cerrar sesión	
Actores	Usuario.
Descripción	El usuario cierra la sesión de su cuenta.
Precondición	El usuario debe haber iniciado sesión previamente.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Mi Perfil». 2. Se despliega una ventana en cuya esquina inferior derecha contendrá un botón «Cerrar sesión». 3. En cuanto se pulse el botón se cerrará la cuenta del usuario y se recargará la página para que el mismo u otro usuario pueda iniciar sesión.
Postcondición	El usuario pierde el acceso a la cuenta hasta que vuelva a iniciar sesión.
Excepciones	
Comentarios	

Tabla 3.3: *Caso de uso 03. Cerrar sesión.*

04 Editar perfil	
Actores	Usuario.
Descripción	El usuario puede modificar cualquier de los campos de su ventana de perfil.
Precondición	El usuario debe haberse registrado y haber iniciado sesión.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Mi Perfil». 2. Se despliega la ventana de perfil con los datos del usuario, y en la parte inferior contiene un botón «Editar perfil». 3. En cuanto se pulse el botón se abrirá una ventana nueva con tres pestañas: la primera para cambiar el nombre de usuario, la segunda para cambiar la contraseña y la tercera para cambiar la imagen de avatar entre las disponibles.
Postcondición	Se actualizan en la base de datos los registros correspondientes a los datos modificados por el usuario.
Excepciones	Si el usuario intenta cambiar la contraseña por una que no cumple con los requisitos mínimos, se notificará al usuario.
Comentarios	

Tabla 3.4: *Caso de uso 04. Editar perfil.*

05 Eliminar cuenta	
Actores	Usuario.
Descripción	El usuario elimina toda la información de su cuenta y cierra la cuenta.
Precondición	El usuario debe haberse registrado y haber iniciado sesión.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Mi Perfil». 2. Se despliega la ventana de perfil con los datos del usuario, y en la parte inferior contiene un botón «Eliminar cuenta». 3. En cuanto se pulse el botón se abrirá una ventana nueva en la que habrá que introducir el nombre de usuario y contraseña y pulsar de nuevo el botón «Eliminar cuenta». 4. Una vez se hayan verificado las credenciales introducidas en la ventana anterior, se procederá a eliminar la cuenta y se recargará la página para que otro usuario pueda acceder.
Postcondición	Se borran de la base de datos los registros asociados a la cuenta del usuario.
Excepciones	Si las credenciales introducidas por el usuario no se pueden verificar, se notificará al usuario.
Comentarios	

Tabla 3.5: *Caso de uso 05. Eliminar cuenta.*

06 Reto diario	
Actores	Usuario.
Descripción	El usuario accede al juego diario en el que tendrá que adivinar la palabra del día para acceder a la información sobre esa palabra.
Precondición	El usuario debe haberse registrado y haber iniciado sesión.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Acceder al reto». 2. Se muestra una página en la que el usuario puede ver el tema de la semana, una imagen pixelada y un tablero en el que tendrá que ir probando combinaciones de letras hasta dar con la palabra del día. 3. A cada combinación de letras que se introduzca, la imagen pixelada mejorará un poco la calidad de la imagen. 4. El usuario dispone de dos botones que dan una pista a cambio de disminuir el tiempo disponible para acabar el reto. 5. Una vez el reto haya finalizado, se mostrará el tiempo usado y los puntos obtenidos en el juego, y se dará acceso a la página con la información sobre la palabra del día.
Postcondición	Se actualizará en la base de datos la cantidad de puntos obtenida por el usuario.
Excepciones	
Comentarios	

Tabla 3.6: *Caso de uso 06. Reto diario.*

07 Juego extra	
Actores	Usuario.
Descripción	Si el usuario consigue adivinar la palabra del día, desbloqueará el acceso al juego extra, con el que podrá aumentar su puntuación.
Precondición	El usuario debe haber ganado el reto diario.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Acceder al juego extra». 2. Se redirige al usuario a la página en la que se encuentra el juego extra. 3. En este juego el usuario deberá utilizar las teclas “awsd” o el pad, en caso de estar utilizando un dispositivo móvil, para esquivar obstáculos durante la mayor cantidad de tiempo posible. 4. Si el usuario tiene el juego extra desbloqueado, podrá jugarlo tantas veces como quiera. 5. La mejor puntuación obtenida en el juego extra será la utilizada para sumar a la puntuación obtenida en el reto diario por el usuario.
Postcondición	Se actualizará en la base de datos la cantidad de puntos obtenida por el usuario.
Excepciones	
Comentarios	

Tabla 3.7: *Caso de uso 07. Juego extra.*

08 Ver ranking	
Actores	Usuario.
Descripción	El usuario tendrá acceso al ranking individual y por Facultades.
Precondición	El usuario debe haber ganado el reto diario.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de «Ranking». 2. Se muestra una página en la que en la parte superior el usuario puede ver su puntuación y posición en los rankings y en la parte inferior puede ver las tablas completas de ranking. 3. El usuario puede pulsar entre las dos pestañas disponibles en las tablas de ranking para alternar entre la tabla de clasificación por puntuación individual y la clasificación por Facultades.
Postcondición	
Excepciones	
Comentarios	

Tabla 3.8: *Caso de uso 08. Ver ranking.*

3.5. Prototipos de la Interfaz de Usuario

De forma adicional, antes de comenzar el desarrollo de la aplicación web realizamos un mockup de la interfaz de usuario, utilizando la herramienta Figma³, para utilizarla como base a la hora de empezar a implementar el proyecto [17]. A la hora de realizar el mockup tuvimos en cuenta los requisitos que tenía que cumplir el proyecto, la disposición de los botones, la paleta de colores y la disposición de todos los elementos para crear una interfaz tan intuitiva y atractiva como fuera posible.

A continuación, podemos encontrar todas las pantallas del mockup:

- **Página de inicio:**

- Propósito: Proporcionar una descripción general de la aplicación y acceso a funciones clave.
- Elementos destacados:
 - Menú de navegación superior e inferior para acceder a diferentes partes de la aplicación.
 - Botón de inicio de sesión en el centro de la pantalla.
 - Botones de RRSS relacionadas con el proyecto.

³Mockup en Figma



Figura 3.4: *Mockup de la página principal previa al inicio de sesión del usuario*

■ **Página de Reto:**

- Propósito: Proporcionar la experiencia de jugar donde intentan adivinar una palabra oculta en un número limitado de intentos.
- Elementos destacados:
 - Presenta una interfaz simple y minimalista que se enfoca en la esencia del juego: adivinar la palabra oculta.
 - Retroalimentación a través de colores, los cuales muestran las letras correctas en la posición correcta, las letras correctas en la posición incorrecta y las letras incorrectas.

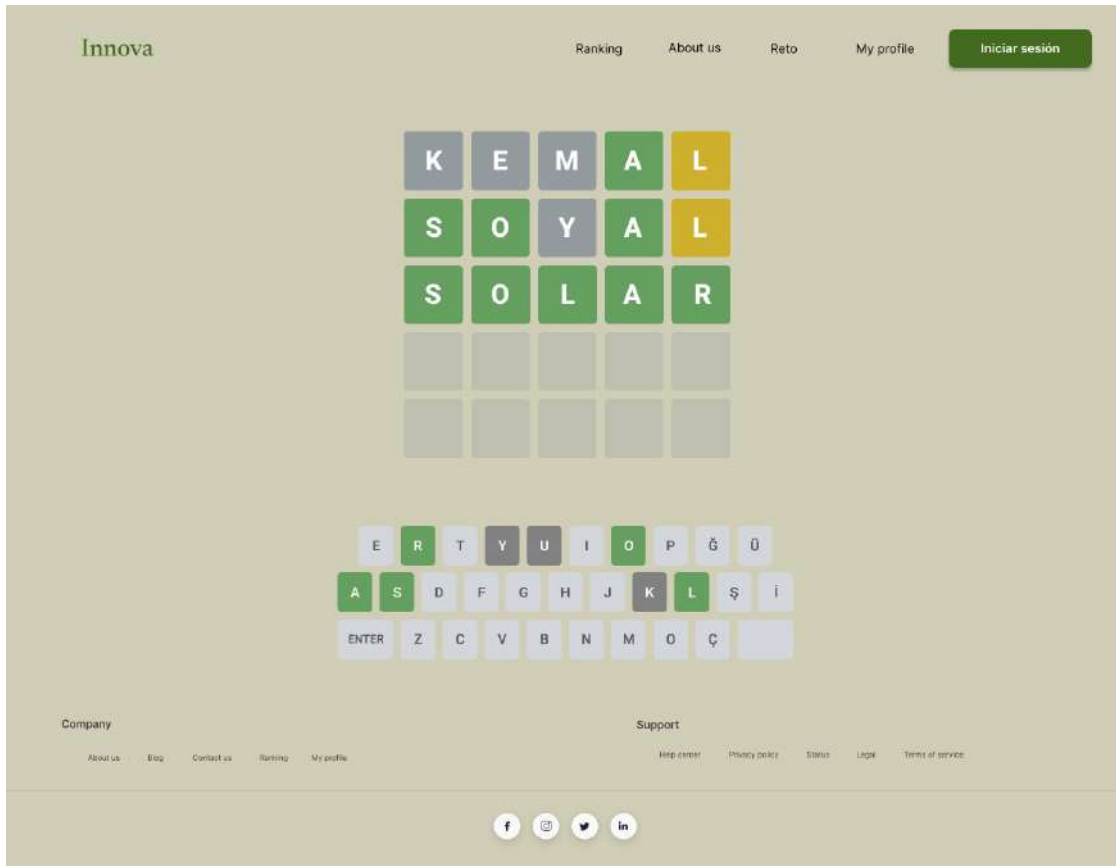


Figura 3.5: *Mockup de la página con el reto diario*

■ **Página de Ranking:**

- Propósito: Proporcionar una visión del ranking de los usuarios y las Facultades. De esta manera hacer que esto motive a los jugadores a competir y mejorar en el reto.
- Elementos destacados:
 - Permite ver a los tres primeros en la clasificación.
 - Presentan botones para cambiar la clasificación, ya sea para ver la clasificación de usuarios o de Facultades.

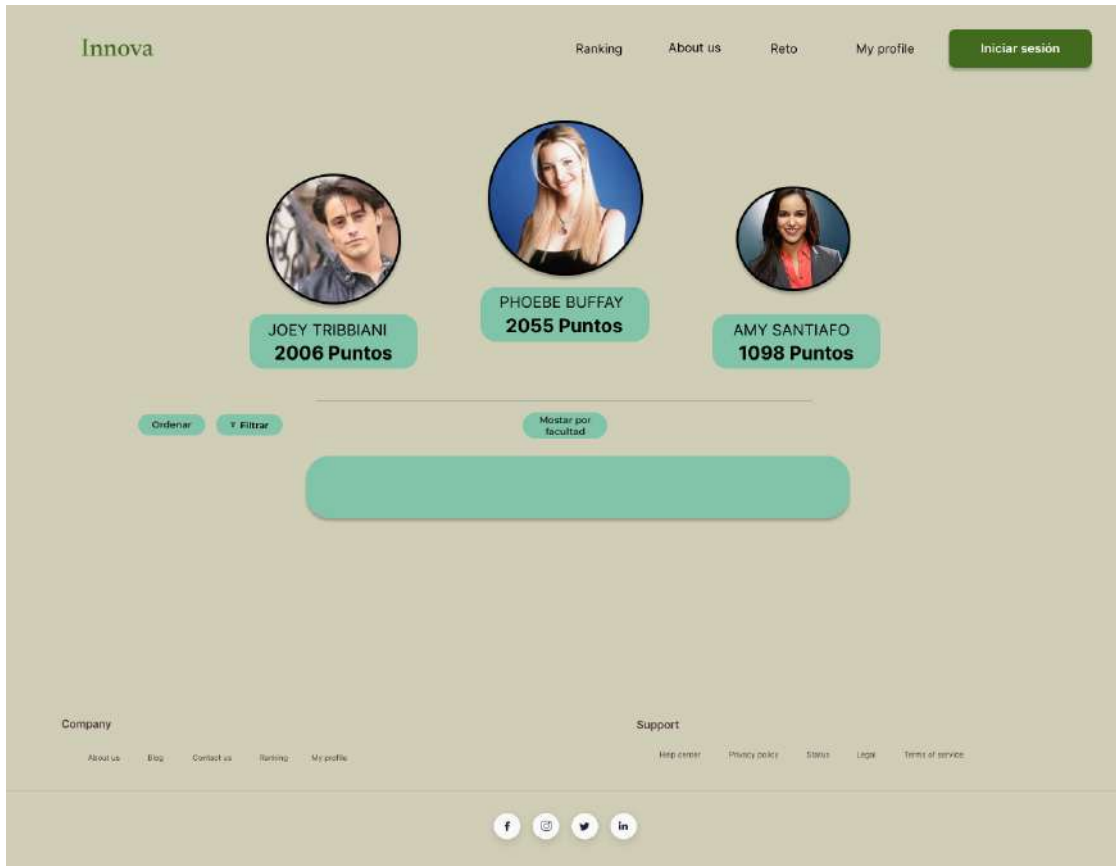


Figura 3.6: *Mockup de la página que contiene el ranking*

■ **Popups de inicio de sesión y registro:**

- Propósito: Permitir a los usuarios iniciar sesión en sus cuentas o registrarse.
- Elementos destacados:
 - Campos de entrada para los datos del usuario.
 - Botones de inicio de sesión o de registro.
 - Enlace para registrarse si aún no tienen una cuenta o iniciar sesión si ya tienen cuenta.

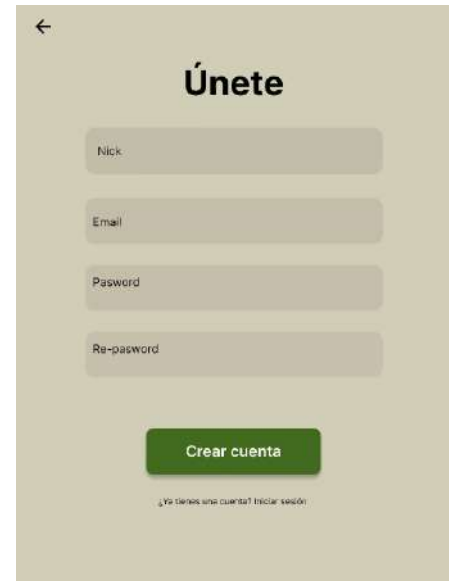
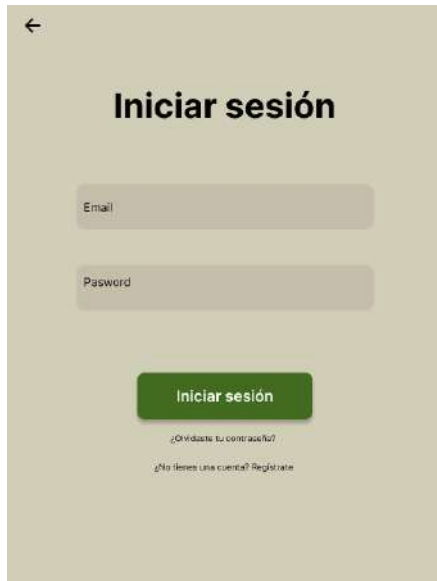


Figura 3.7: *Mockup de los popups de inicio de sesión y registro*

■ **Popups de perfil y editar perfil:**

- Propósito: Permitir a los usuarios ver los datos de su perfil y editarlos.
- Elementos destacados:
 - Foto de avatar del usuario y todos los datos del usuario.
 - Botones de editar perfil o de cerrar sesión.
 - Campos de entrada para los datos del usuario que quiere editar.

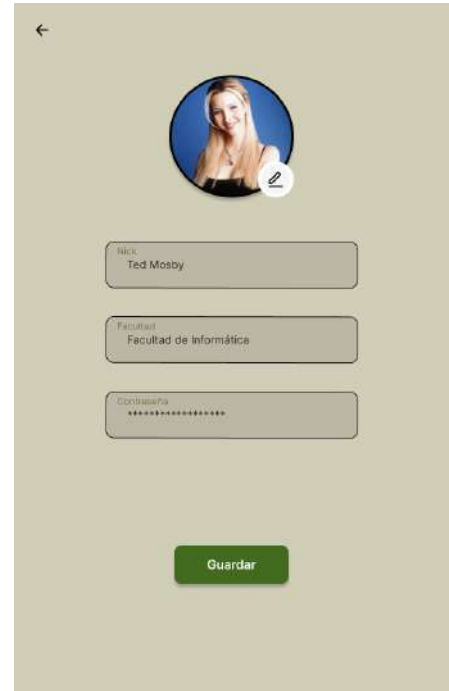


Figura 3.8: *Mockups de los popups de perfil y editar perfil*

Capítulo 4

Estructuración del código web

En este apartado se detallan las tecnologías utilizadas para la aplicación web y los elementos creados para el servidor, detallando la estructura e implementación de todos los elementos que intervienen a más bajo nivel.

Para una visión más general de la estructura del proyecto, en la Figura 4.1 se puede ver un gráfico generado mediante *Visualizing a Codebase*¹ y que muestra el tipo de lenguaje y la cantidad de código que hay en cada zona del proyecto, lo que permite comprender con mayor facilidad dónde se encuentra cada elemento y qué relevancia tiene con respecto a la aplicación.

La estructura que se va a comentar a continuación se basa en la organización que hemos seguido a la hora de diseñar el proyecto. Se ha seguido un patrón Modelo-Vista-Controlador que nos ha permitido mantener una visión clara y ordenada de todos los componentes del sistema, facilitando así su desarrollo y mantenimiento.

Para el desarrollo del proyecto se ha trabajado con tecnologías de Git, concretamente *GitHub*, para crear el repositorio en el que se encuentra actualmente el proyecto y al cual se puede acceder desde el siguiente enlace: <https://github.com/dprieto96/TFG/tree/main>.

¹Visualizing a Codebase

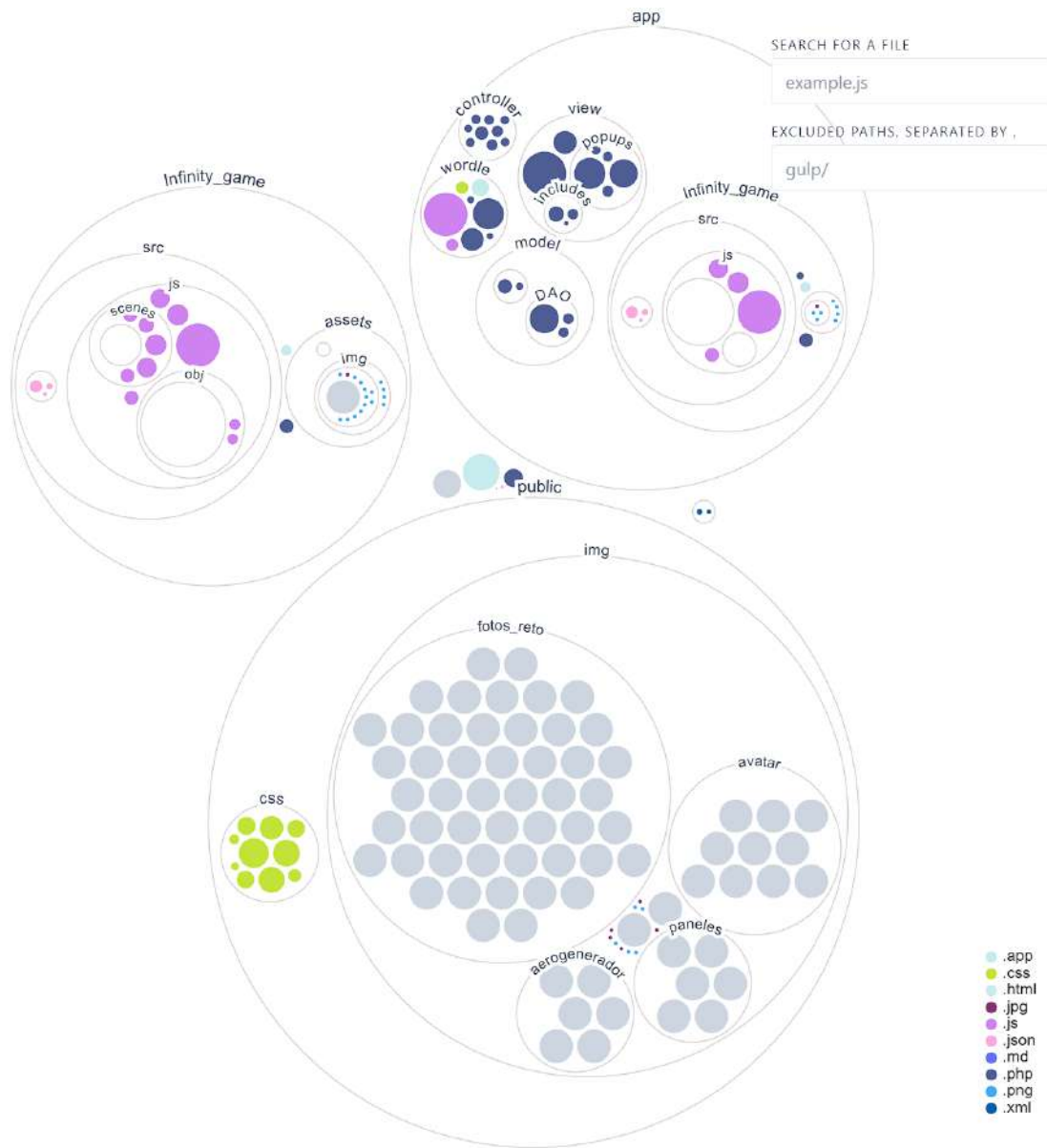


Figura 4.1: Estructura del proyecto representada en diagrama de zonas

4.1. Front-end o lado del cliente

En esta sección, describiremos cómo hemos organizado el front-end del proyecto, en varias secciones clave. La estructura del front-end está diseñada para ser modular y fácil de mantener, permitiendo una navegación intuitiva y una experiencia de usuario coherente. A continuación, se detalla, la página principal, que actúa como el punto de entrada al sitio web, la carpeta “includes”, que contiene el *header* y *footer*, que se irán incorporando a cada página, la carpeta “popups”, que gestiona las ventanas emergentes, y los archivos *aboutUs.php* y *ranking.php*, que proporcionan información sobre el proyecto y las clasificaciones respectivamente.

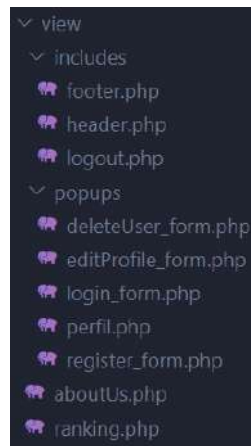


Figura 4.2: *Estructura de carpetas del front-end*

4.1.1. Página principal

Antes de empezar a analizar la implementación de los principales archivos que constituyen el front-end de nuestro proyecto, vamos a analizar el archivo *index* que representa la página principal de nuestra aplicación, el cual se puede ver en la Figura 4.3.

Dado que nuestra aplicación está destinada a funcionar tanto en ordenadores como en dispositivos móviles, hemos optado por mantener el número de elementos en la página principal lo más reducido posible para evitar saturar la pantalla del usuario. Hemos diseñado una página que representa un fondo de pantalla en el que además del *header* y el *footer* el

usuario solo verá el lema de la aplicación y dos botones ubicados justo en el centro de la pantalla: uno para acceder al reto diario y otro para acceder al juego extra.

Ambos botones los hemos implementado de forma que en caso de no cumplirse ciertos requisitos, el usuario no pueda acceder a la sección correspondiente.

En el caso del botón del reto diario, hemos registrado previamente para cada usuario la fecha en la que realizó por última vez el reto diario. De esta forma, si el usuario ya ha accedido al reto en un día determinado, le aparecerá el botón bloqueado informando de que ya no tiene acceso durante ese día.

Del mismo modo, en el caso del botón del juego extra se verificará la misma fecha que hemos mencionado anteriormente para confirmar que el usuario ha completado el reto del día, pero además se verificará también que el usuario haya ganado dicho juego. De esta forma, solo los usuarios que hayan sido capaces de ganar el reto de ese día verán el botón desbloqueado.



Figura 4.3: *Página principal una vez el usuario ha iniciado sesión*

4.1.2. Carpeta “includes”

En esta carpeta hemos incluido todos los elementos de la aplicación web que se distribuyen en todos los puntos de la aplicación web y a los que los usuarios deben tener acceso en todo momento.

- **footer.php**: para la implementación del footer decidimos dividir el contenido en columnas, dejando en una columna todos los accesos a las redes sociales relacionadas con el proyecto y en otra columna toda la información relacionada con los derechos de autor, las políticas legales y las atribuciones de diseño de la aplicación.



Figura 4.4: *Footer presente en todas las páginas de la aplicación*

- **header.php**: para la implementación del header hemos optado por mantener una estructura simplificada, mantenido visible solo los elementos indispensables. Para organizar estos elementos, se ha seguido la misma lógica utilizada en el pie de página, dividiendo el contenido en dos columnas. En la columna derecha, en el caso de que el usuario aún no haya iniciado sesión, estarán disponibles el enlace a la sección “Sobre Nosotros” y el botón para acceder al formulario de inicio de sesión. Por otro lado, si el usuario ya ha iniciado sesión, también podrá ver el enlace al ranking y el botón para desplegar su perfil. En la columna izquierda hemos añadido un *escape hatch* para que el usuario pueda volver a la página principal de forma rápida desde cualquier punto de la aplicación.



Figura 4.5: *Header presente en todas las páginas de la aplicación*

- **logout.php**: este fichero es el encargado de gestionar el cierre de la sesión de un usuario, destruyendo la sesión, liberando las variables de sesión y redirigiendo al usuario a la página principal para mantener una experiencia fluida.

4.1.3. Carpeta “popups”

En esta carpeta hemos añadido todos los componentes de la interfaz de usuario que se presentan en forma de ventanas emergentes o modales, ver Figura 4.6.

- **deleteUser_form.php**: este fichero se encarga de verificar la contraseña del usuario antes de proceder a eliminar la cuenta y una vez hecho, la envía por medio de un formulario utilizando una petición POST al controlador que se encarga de borrar la cuenta de la base de datos.
- **editProfile_form.php**: este fichero contiene todas las operaciones relacionadas con la edición del perfil del usuario. Cabe destacar que en la sección de cambio de contraseña, es en este fichero en el que se verifica que la nueva contraseña cumpla los criterios necesarios antes de pasarla al controlador que la cambiará en la base de datos. De esta forma se reduce el número de peticiones POST innecesarias. Otro punto a destacar es en cuanto a la edición del avatar. En un principio pensamos implementar esta función de forma que el usuario pudiera subir una imagen de perfil desde su dispositivo, pero después de analizar las implicaciones de seguridad y usabilidad que esto podría tener, optamos por guardar unos avatares predeterminados entre los cuales los usuarios pueden elegir.
- **login_form.php**: este fichero se encarga de verificar el nombre y contraseña del usuario y los envía al controlador que se encarga de gestionar el inicio de sesión y crear las variables de sesión necesarias para las distintas operaciones.
- **perfil.php**: este fichero se encarga de recuperar y mostrar los principales datos del usuario por medio del uso de variables de sesión. Del mismo modo, se incluyen botones

que permiten acceder a la edición de los datos del usuario y a la eliminación de la cuenta si fuera necesario.

- **register_form.php**: este fichero contiene un formulario que pide al usuario todos los datos necesarios para crearle una cuenta. Estos datos, del mismo modo que en los casos anteriores, se validan internamente por medio de funciones para posteriormente enviarse por medio de una petición POST al controlador encargado de crear la nueva cuenta.



Figura 4.6: Ejemplo de ventanas modales para visualizar el perfil y cambiar la contraseña respectivamente

4.1.4. aboutUs.php

El fichero aboutUs.php es un componente muy importante a pesar de ser un fichero muy sencillo en lo que a implementación se refiere, ya que es la página que debe servir para que cualquier persona que visite la web se interese por el proyecto y decida registrarse. Para conseguir esto en esta página hemos incluido los principales objetivos que buscamos conseguir con este proyecto, una idea básica del funcionamiento de la aplicación, centrada en el aprendizaje por medio de juegos; y las aspiraciones que tenemos de cara a lo que podamos obtener en un futuro gracias a la aplicación. Además, al final de este documento

hemos añadido un enlace directo a la página principal para que cualquier persona que esté interesada en participar en el proyecto pueda registrarse de forma rápida y sencilla.

4.1.5. ranking.php

El fichero ranking.php despliega una página web que muestra el ranking de los usuarios y facultades en nuestra plataforma en función de los puntos que han obtenido en los retos diarios y en el juego extra.

Nuestra implementación se centra en garantizar la seguridad y la autenticación del usuario, de forma que si alguien que no ha iniciado sesión intenta acceder se le redirigirá a la página principal. Además, hemos utilizado un controlador dedicado para gestionar la obtención de datos del ranking, lo que nos ha permitido mantener claramente separadas las responsabilidades del back-end y del front-end.

Hemos estructurado la página buscando que sea lo más intuitiva posible. Como se puede ver en la Figura 4.7 lo primero que hemos añadido es un cuadro en el que se incluyen los datos principales del usuario y las puntuaciones personal y de Facultad que ha obtenido hasta el momento. A continuación hemos añadido los rankings completos organizados en forma de tablas y de forma que el usuario pueda alternar entre la tabla de puntuaciones individuales y la de puntuaciones de las facultades. La clasificación se actualiza dinámicamente y se ordena según la puntuación obtenida. Además, para que el usuario que ha iniciado sesión pueda identificarse con facilidad dentro de las tablas de ranking, hemos implementado una función que destaca visualmente la fila correspondiente a dicho usuario, modificando el color del borde y las letras de la fila de la tabla.

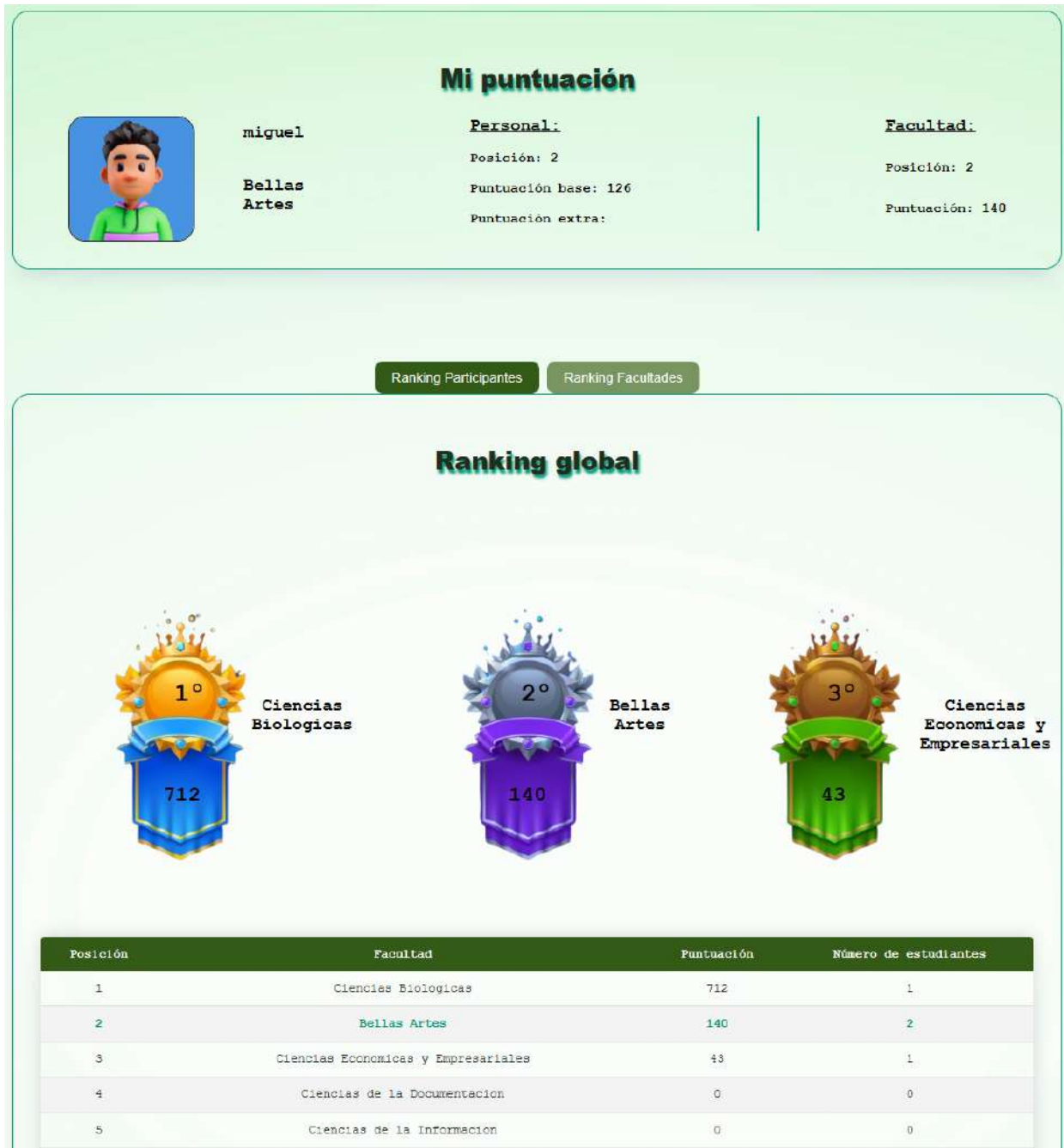


Figura 4.7: Visualización de la puntuación del usuario y la tabla de ranking

4.2. Infinity-game.php

Esta página es la que se encarga de alojar el juego infinito, dentro de la página principal, comprobando que el juego esté siempre a la resolución necesaria para poder ser jugado correctamente. Cuenta además, con un pequeño título, en la parte superior, que muestra el nombre del juego y un recuadro en la parte inferior que contiene la información necesaria para jugar, incluyendo los controles que se utilizarán y el objetivo del juego.

Este juego será accesible durante todo el día, siempre que se haya superado el reto diario, de tal forma que permita al usuario jugar todas las veces que quiera en ese tiempo y le ayude a mejorar su puntuación.

La puntuación se calcula, en función del tiempo que el jugador permanezca sin colisionar con los elementos de polución, de forma que cuanto más aguante el jugador esquivando los obstáculos, más puntos podrá sumar a la puntuación que haya obtenido en el reto diario.

Al principio se contempló la opción de sumar todas las puntuaciones que se obtuvieran en el juego infinito, pero dado que esto podría dar lugar a puntuaciones muy dispares, se ha optado por coger la máxima puntuación obtenida por el usuario en el juego infinito de ese día. De esta forma se consigue que las puntuaciones estén más equilibradas, y anima a los usuarios a seguir jugando.

4.3. Juego diario (index.php)

Dentro de la carpeta *app/wordle* encontramos este archivo, que se encarga de mostrar el juego principal. Para ello se ha optado por un teclado en pantalla que se ha ido creando con elementos html, los cuales han sido la solución más efectiva que se ha encontrado, ya que al ser directamente una letra alojada en un recuadro ganamos algo en rendimiento, sin tener que incluir ni llamar a otras clases. La idea es dar todos los elementos en un golpe de vista en la página, sin la necesidad de que el jugador tenga que desplazarse ni realizar ninguna otra acción para verlos en su totalidad.

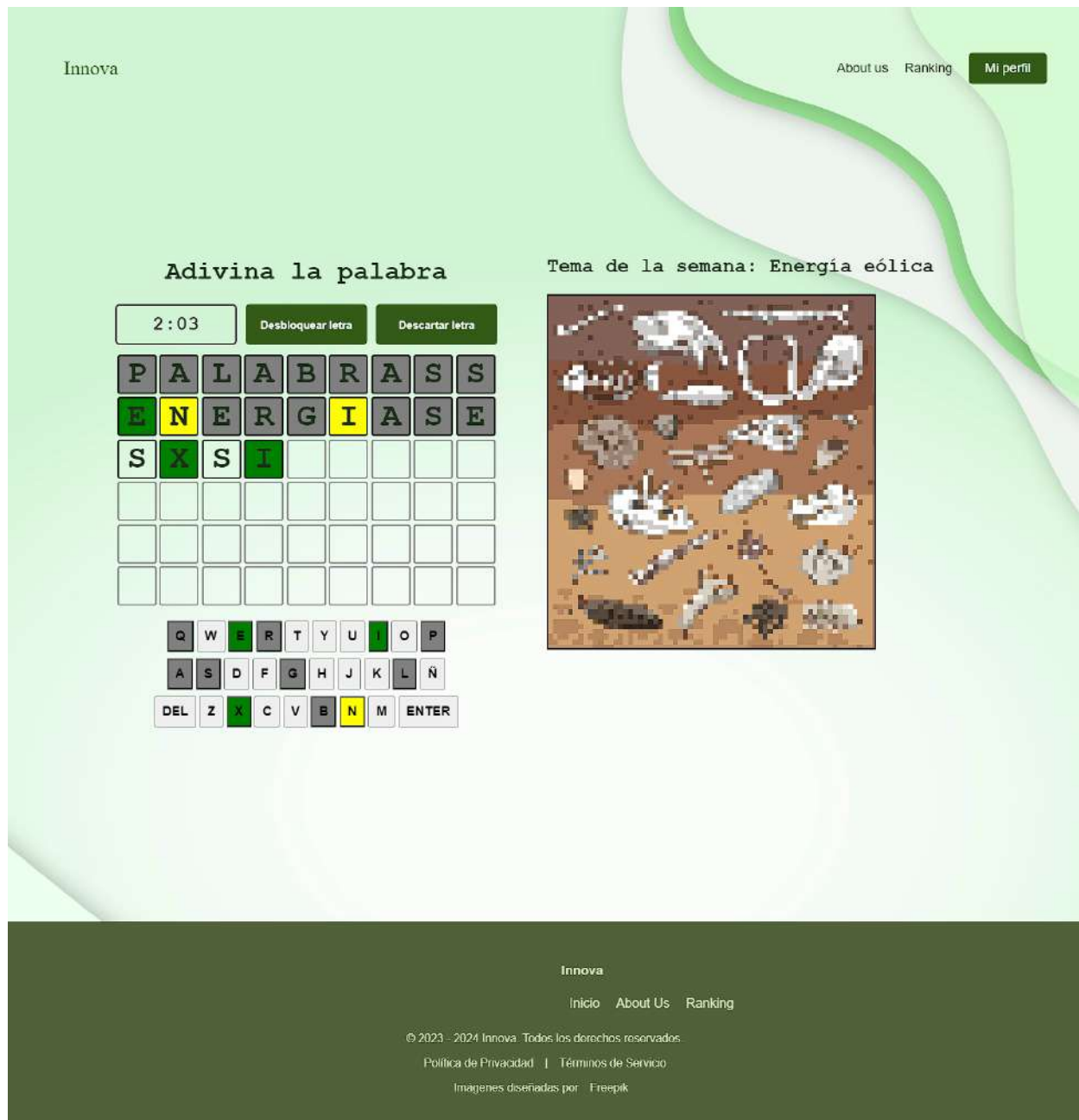


Figura 4.8: *Visualización del reto diario durante su ejecución*

Además del elemento footer y header explicado anteriormente, podemos ver que disponemos de un contador a partir del cual, según transcurra su tiempo, irá disminuyendo nuestra puntuación. Debajo se encuentra el tablero donde poder introducir nuestras palabras, bien por teclado o bien por pantalla haciendo clic en la letra. Según vayamos gastando intentos,

la imagen de la derecha se des-pixelará y con ello se podrá obtener una ayuda para resolver el acertijo diario. Del mismo modo, los dos botones superiores, “Desbloquear letra” y “Descartar letra”, nos proporcionan otra ayuda. Una vez finalizado el juego obtendremos un popup con nuestro resultado.

4.4. Back-end o Lado de servidor

El Back-end de nuestro proyecto lo hemos estructurado en cuatro secciones principales, cada una dedicada a un objetivo distinto. A continuación vamos a detallar la implementación de cada una de las secciones:

4.4.1. Carpeta “*DAO*”

- **conexion.php**: Este fichero define una clase que obtiene una conexión a la base de datos utilizando el patrón Singleton^[12]. La primera vez que se llama al método `getConnection()`, se crea una instancia de esta clase y se establece una conexión con la base de datos. En llamadas posteriores, se devuelve la misma instancia sin volver a crearla, lo que garantiza que solo haya una conexión activa en cualquier momento.
- **userDAO.php**: Este fichero define una clase llamada `userDAO` que actúa como un objeto de acceso a datos para manipular usuarios en la base de datos. Algunas de las características principales de este código son:
 - Conexión a la base de datos: Establece una conexión a la base de datos utilizando un objeto de conexión (`$db`) obtenido a través de la clase `DB::getInstance()`, que puede implementar el patrón Singleton para garantizar que la instancia de conexión sea única en toda la aplicación.
 - Funciones de registro e inicio de sesión: La función de registro es responsable de registrar un nuevo usuario en la base de datos, mientras que la función de inicio de sesión permite al usuario iniciar sesión en el sistema. Ambas funciones

verifican la existencia del usuario en la base de datos y validan las credenciales proporcionadas.

- Función de edición de usuario: Proporciona la función de editar diferentes atributos de los usuarios, como nombre de usuario, contraseña y avatar. Estas funciones realizan operaciones de actualización en la base de datos después de validar la información proporcionada.
- Función de eliminación y actualización de puntajes: Además, existe una función para eliminar usuarios y actualizar puntajes en el sistema, ya sean puntajes regulares o puntajes adicionales.
- Función para marcar ganador o perdedor: el sistema incluye una función para marcar a un usuario como ganador o perdedor.

En resumen, `userDao` proporciona una interfaz para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en usuarios dentro de la base de datos, manteniendo las operaciones seguras mediante el uso de contraseñas *hash* y consultas parametrizadas para evitar la inyección de SQL.

- **rankingDAO.php**: Este fichero define una clase llamada `RankingDAO` que actúa como un objeto de acceso a datos para leer los usuarios y Facultades de la base de datos. Ambos métodos `getUserRanking()` y `getFacultyRanking()` obtienen todos los usuarios y todas las Facultades, facilitando así la obtención de los rankings correspondientes.

4.4.2. Carpeta “domains”

- **tUser.php**: Este fichero define una clase llamada `tUser` que representa un usuario en la aplicación. Algunas de las características principales de este código son:
 - Encapsulación de datos: La clase `tUser` usa variables privadas para almacenar los datos del usuario. Esto garantiza la protección de los datos y la abstracción de estos mismos.

- Simplicidad: El código es simple, centrándose en la representación y manipulación de datos de usuario. Esto facilita su comprensión y mantenimiento.

En resumen, `tUser` contiene información del usuario y ofrece métodos para obtener estos datos de manera segura. Otras clases, como `userSA` o `userDAO`, la utilizan para gestionar usuarios dentro de la aplicación.

4.4.3. Carpeta “services”

- **userSA.php**: Este fichero define una clase llamada `UserSA` (User Service Access), que actúa como una interfaz de alto nivel para acceder a las funcionalidades relacionadas con usuarios en la aplicación. Algunas de las características principales de este código son:
 - Dependencia del Objeto de Acceso a Datos (DAO): La clase `UserSA` posee una dependencia del objeto `UserDAO`, el cual ofrece métodos para interactuar con la base de datos y ejecutar acciones CRUD que tienen relación con los usuarios.
 - Métodos de Interfaz Pública: La clase `UserSA` proporciona métodos públicos que sirven como una interfaz para realizar diversas acciones relacionadas con los usuarios, como iniciar sesión, registrar un nuevo usuario, editar información del usuario, eliminar un usuario, entre otros. Estos métodos encapsulan la lógica de negocio y delegan las operaciones específicas al objeto `UserDAO`.
 - Uso de la Clase `tUser`: Algunos métodos de la clase `UserSA` utilizan objetos de la clase `tUser` para representar y manipular la información de los usuarios.

En resumen, `UserSA` ofrece una capa de abstracción que simplifica el acceso y la manipulación de los datos de usuarios en la aplicación, facilitando la modularidad y mantenibilidad del código.

- **rankingSA.php**: Este fichero define una clase llamada `rankingSA` (Ranking Service Access), que actúa como una interfaz de alto nivel para acceder a las funcionalidades

relacionadas con los rankings en la aplicación. Algunas de las características principales de este código son:

- **Dependencia del Objeto de Acceso a Datos (DAO):** La clase `rankingSA` depende del objeto `RankingDAO`, el cual proporciona métodos para interactuar con la base de datos y obtener los rankings de usuarios y Facultades.
- **Métodos de Interfaz Pública:** La clase `rankingSA` ofrece métodos públicos que sirven como una interfaz para obtener los rankings de usuarios y Facultades. Estos métodos simplemente delegan las operaciones al objeto `RankingDAO`, el cual se encarga de realizar las consultas a la base de datos.

En resumen, `rankingSA` proporciona una capa de abstracción que simplifica el acceso a los rankings de usuarios y Facultades en la aplicación, facilitando la modularidad y mantenibilidad del código.

4.4.4. Carpeta “controller”

- **`processDeleteUser.php`:** Este fichero recibe el usuario y la contraseña mediante el formulario POST. Luego, intenta eliminar al usuario llamando al método `deleteUser` del servicio `UserSA`. Dependiendo del resultado de este método, redirecciona a la página de inicio con o sin mensaje de error.
- **`processEditAvatar.php`:** Este fichero recibe la URL del avatar mediante el formulario POST. Luego, actualiza el avatar del usuario en la base de datos utilizando el método `editAvatar` del servicio `UserSA`. Si tiene éxito, actualiza la sesión con el nuevo avatar y redirecciona al usuario a la página principal. Si falla, redirecciona con un mensaje de error.
- **`processEditPassword.php`:** Este fichero recibe la contraseña antigua y las nuevas contraseñas mediante el formulario POST. Luego, llama al método `editPassword` del servicio `UserSA` para cambiar la contraseña del usuario en la base de datos. Si tiene

éxito, redirecciona al usuario a la página principal. Si falla, redirecciona con un mensaje de error.

- **processEditUser.php**: Este fichero recibe el nuevo nombre de usuario mediante el formulario POST. Después, llama al método editUser del servicio UserSA para actualizar el nombre de usuario en la base de datos. Si el método tiene éxito, actualiza la sesión con el nuevo nombre de usuario y redirecciona al usuario a la página principal. Si falla, redirecciona a la pantalla de inicio con un mensaje de error.
- **processLogin.php**: Este fichero recibe el nombre de usuario y la contraseña mediante el formulario POST. Después, inicia sesión llamando al método loginUser del servicio UserSA. Si el método tiene éxito, actualiza la sesión con la información del usuario y redirecciona a la página principal. Si no tiene éxito, redirecciona a la página de inicio y muestra un mensaje de error.
- **processRegistration.php**: Este fichero recibe el nombre de usuario, correo electrónico, contraseña y Facultad mediante el formulario POST. Tras recibir los datos, llama al método registerNewUser del servicio UserSA para registrar al nuevo usuario en la base de datos. Si el registro de usuario tiene éxito, actualiza la sesión con la información del nuevo usuario y redirecciona a la página principal. Si no tiene éxito, redirecciona a la página de inicio y muestra un mensaje de error. Marca al usuario como ganador del reto diario a través del método winner() y como perdedor gracias al método loser().
- **processWinner.php**: Este fichero define una clase llamada processWinner, la cual se encarga de procesar el estado de un usuario como ganador o perdedor del reto diario.
- **rankingController.php**: Este fichero define una clase llamada RankingController, la cual se encarga de obtener los rankings de usuarios y Facultades mediante los métodos getUserRanking() y getFacultyRanking() respectivamente.
- **pointsController.php**: Este fichero define una clase llamada pointsController, la cual

se encarga de manejar los puntos de los usuarios. Esto lo hace gracias a los métodos `addScore($score)` y `addExtraScore($extraScore)`.

4.4.5. Carpeta “Infinity-game”

Esta carpeta contiene la lógica del juego secundario de la web, al cual se tiene acceso únicamente superado el reto diario y al que se tiene acceso durante ese día únicamente, con lo que añade algo más de complicación a la hora de obtener una mayor puntuación. El juego se compone además de los elementos visuales mencionados anteriormente:

- **phaser.js**: Este fichero es la consecuencia de haber quitado las dependencias de `no-de.js`, a partir de las cuales hemos conseguido que en un único fichero se integre toda la lógica de esta tecnología, haciendo que el tamaño de los archivos disminuya considerablemente. Contiene toda la lógica necesaria para que Phaser pueda funcionar correctamente.
- **game.js**: Aquí es donde se establece la configuración inicial que requiere phaser para cualquier juego creado. A partir de ella es desde donde se lanza toda la ejecución lógica del juego secundario de nuestra aplicación web.
- **settings.js**: Se encarga de establecer unas macros para hacer más legible el código en los elementos implementados posteriormente.
- **levelSelector.js**: Inicializador de niveles que se encarga de que las escenas sean llamadas en un orden concreto y de garantizar que estas tengan todos los elementos necesarios en el momento de la visualización.
- **progressScene.js**: Escena que muestra una barra de carga al comienzo del juego y pretende cargar todos los elementos del juego, ya sean imágenes, audio, sprites y assets, para que el nivel no aparezca con algún elemento de carga.
- **nivelVertical.js**: Escena donde se implementa toda la lógica del juego en sí. Añade el personaje del jugador, así como los enemigos, la posición de caída y velocidades de

estos, y actualiza la posición y estado del jugador en todo momento. Esto quiere decir que es el encargado de controlar el número de vidas y la finalización del juego.

- **player.js**: Objeto que se encarga de cargar el personaje del jugador junto con sus imágenes de movimiento.
- **gameOverScene.js**: Escena final que muestra un mensaje de partida finalizada y cómo volver a reiniciar el juego, ya que ella se encarga de reiniciar los valores iniciales del juego.
- **/assets.js**: En este directorio se encuentran todas las imágenes que necesita el juego, bien para los movimientos de los jugadores, las imágenes de fondo, creación de enemigos...

4.4.6. Carpeta “wordle”

En esta carpeta se encuentran todos elementos que se utilizan en el juego principal, el cual consiste en averiguar la palabra del día mediante una serie de pistas y retos. La carpeta se compone de:

- **index.php**: Vista general de la web y comentada en el apartado anterior
- **script-wordle**: Script que contiene toda la lógica principal del funcionamiento del juego. En ella se comprueba que letras han sido colocadas y en qué orden, para mostrar así un código de colores para que el usuario pueda solucionar el reto y adivinar la palabra. Además, es la encargada de calcular la puntuación basándose en un temporizador y proporcionar las ayudas.
- **lista-palabras.js**: Fichero con una variable constante que contiene todas las palabras que van a ser objeto de adivinar. Esta lista está implementada en otro fichero para mejorar su legibilidad y conseguir que sea más sencillo de escalar.

- **daily-info.php**: Fichero que contiene la información a mostrar una vez adivinada la palabra del día.

Capítulo 5

Implementación de la Aplicación

5.1. Arquitectura de la aplicación

A la hora de crear el servidor, se ha alojado toda la aplicación web¹ en los servicios de AWS, haciendo uso de la tecnología *Amazon Elastic Compute Cloud* (Amazon EC2), que nos permite crear instancias de máquinas bajo demanda.

Arquitectura

En cuanto a la arquitectura, nos guiaremos a través de la Figura 5.1 que se muestra a continuación. Todos los servicios de la máquina están alojados en los servicios de AWS, accesibles únicamente a través de Amazon *API Gateway*. Este servicio traduce las solicitudes desde una IP pública a una IP privada de AWS, asociada a una VPC (Virtual Private Cloud). En esta red, tenemos todos los servicios que utilizamos. En este caso, el único servicio es EC2 (*Amazon Elastic Compute Cloud*). Si necesitamos otros servicios como Secret Manager para la gestión de contraseñas, deberán estar dentro de esta red para poder interconectarse.

En cuanto a la interacción del usuario, se realiza una conexión a internet a través de una red pública, que es consultada en los servidores DNS, los cuales redirigen a los servicios de AWS. Al acceder desde la API Gateway, el usuario no interactúa directamente con la máquina, sino únicamente con los servicios del servidor que se han instalado previamente.

¹Acceso a la aplicación web

Desde el lado del desarrollador, el acceso se realiza mediante una clave SSH (*Secure Shell*), sin necesidad de contraseñas. Esto verifica ambas partes y permite el acceso a los servicios de la máquina vía FTP (*File Transfer Protocol*), proporcionando autenticación extremo a extremo y segura. ²

En la Figura 5.1 vemos en naranja los servicios que nos proporciona AWS, y en azul los servicios que están asociados a esa VPC (*Virtual Private Cloud*). Cabe destacar que para modificar cualquier parámetro del EC2, será únicamente vía Consola de AWS.

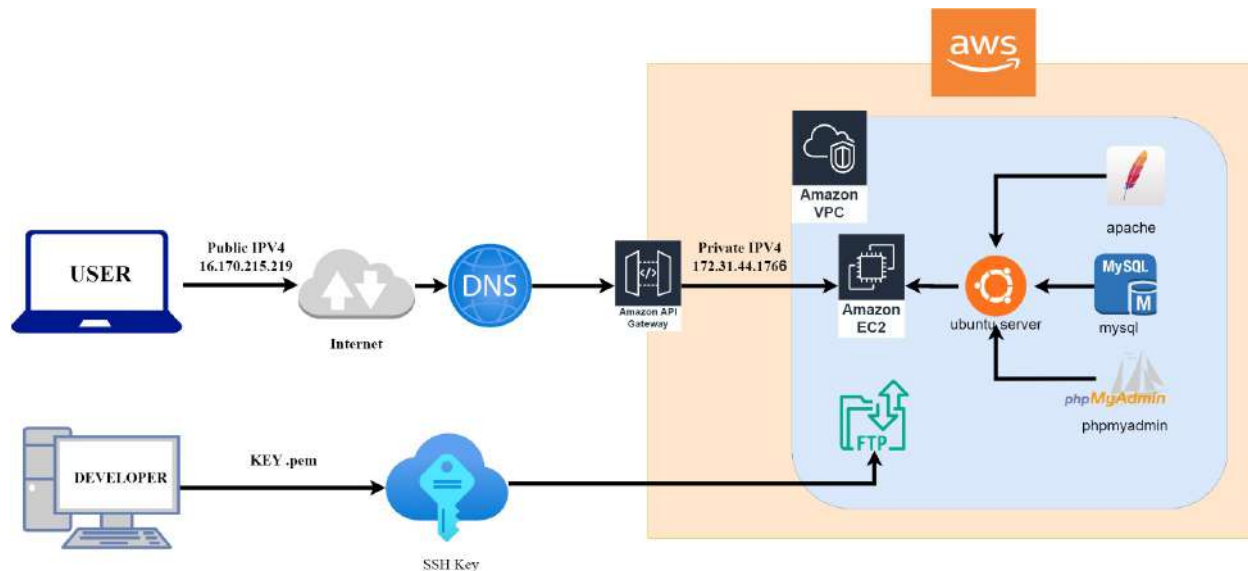


Figura 5.1: Arquitectura empleada para el funcionamiento del servidor

Información General

- Nombre del Servidor: innova php server
- ID de Instancia EC2: i-0dcd9a634a0993845
- Región de AWS: Europa-Estocolmo(eu-north-1)

²Tecnologías mencionadas: Amazon EC2, Amazon API Gateway, Amazon VPC, AWS Secrets Manager, SSH Protocol, FTP Protocol

- **Tipo de Instancia:** t3.micro
- **Sistema Operativo:** Linux/UNIX -(Ubuntu)

Recursos de Computación

- **Cantidad de CPU:** 2 vCPU asignadas al servidor
- **Memoria RAM:** 1 GB de memoria RAM
- **Almacenamiento Principal:** 5 GB

Red y Conectividad

- **Dirección IP Pública:** 16.170.215.219
- **Dirección IP Privada:** 172.31.44.176
- **ID de grupo de seguridad:** sg-0eb94ba47b056fea5
- **ID de subred:** subnet-0dfde313e98b5f76b
- **ID de VPC:** vpc-0bc4723c3cd6128ee
- **Seguridad de Red:** launch-wizard-3 created 2024-04-27T10:15:34.136Z

Gestión de Acceso

- **IAM Roles:** -
- **Claves :** fichero innova.ppk

Costos y Facturación

- **Costos de Instancia:** 0,0104 USD/hora* *Licencia educativa, gratuita

Software y Componentes

El servidor está configurado con una instancia de una máquina Ubuntu-server ³, configurada de tal modo que sea accesible desde direcciones de IPs públicas mediante protocolo HTTP y HTTPS, para lo cual se ha tenido que activar el firewall de la máquina y permitir el tráfico entrante para establecer estas conexiones.

Se ha elegido esta instancia, ya que para nuestra aplicación web era necesario tener los servicios de Apache, MySQL y phpmyAdmin, los cuales pueden ser instalados mediante paquetes.

Esta máquina contiene los paquetes y, por lo tanto, la seguridad de acceso a todos los servicios mencionados anteriormente. Es por ello que para poder acceder, además de hacerlo mediante nuestra cuenta de AWS, necesitamos una clave. Esta se ha generado en el momento de la instancia y no puede replicarse. Se ha obtenido en formato .pem, lo cual ha hecho que sea más sencillo acceder mediante Xserver, para posteriormente convertirse a .ppk para poder añadir ficheros mediante FTP.

Estado y disponibilidad

El servidor se encuentra siempre activo gracias a que AWS garantiza una total disponibilidad, encargándose de que la instancia permanezca siempre levantada, de tal forma que siempre va a ser accesible por el usuario. Podemos ver en su documentación:

Service Commitment

“AWS” will use commercially reasonable efforts to make Amazon EC2 and Amazon EBS each available with a Monthly Uptime Percentage (defined below) of at least 99.99%, in each case during any monthly billing cycle (the “Service Commitment”). In the event Amazon EC2 or Amazon EBS does not meet the Service Commitment, you will be eligible to receive a Service Credit as described below.

³Ubuntu-server

“Monthly Uptime Percentage” is calculated by subtracting from 100 % the percentage of minutes during the month in which Amazon EC2 or Amazon EBS, as applicable, was in the state of “Region Unavailable.” Monthly Uptime Percentage measurements exclude downtime resulting directly or indirectly from any Amazon EC2 SLA Exclusion (defined below). “Region Unavailable” and “Region Unavailability” mean that more than one Availability Zone in which you are running an instance, within the same Region, is “Unavailable” to you. “Unavailable” and “Unavailability” mean: For Amazon EC2, when all of your running instances have no external connectivity. For Amazon EBS, when all of your attached volumes perform zero read write IO, with pending IO in the queue. A “Service Credit” is a dollar credit, calculated as set forth below, that we may credit back to an eligible account.

Fuente: AWS

Por todos estos motivos, AWS, y concretamente sus servicios EC2, han sido elegidos para alojar la infraestructura de nuestro servidor, ya que nos da una disponibilidad prácticamente total y nos permite hacer uso de instancias Linux, a las que poder instalar prácticamente cualquier paquete y además hacerlas accesibles desde cualquier lugar de internet a un coste muy bajo y con un alta seguridad.

5.2. Base de datos: construcción del modelo de datos

Hemos desarrollado una base de datos que se adapta a los requisitos del proyecto, manteniendo la simplicidad y la eficiencia en la recuperación de datos [15]. El modelo de datos consta de dos tablas principales:

- **Tabla usuario:** Esta tabla guarda toda la información necesaria de cada usuario. Tiene una clave foránea que referencia a la Facultad a la que pertenece.
- **Tabla Facultad:** Esta tabla contiene todas las Facultades de la UCM, los estudiantes

registrados de cada Facultad y los puntos que suman todos ellos.

Adicionalmente, hemos implementado disparadores en la tabla Usuario para automatizar ciertas acciones y mantener la coherencia de los datos relacionados con las Facultades:

- **Actualizar puntos de Facultad:** Se activa tras la actualización de un registro en la tabla Usuario para actualizar el total de puntos de la Facultad correspondiente.
- **Eliminar usuario de Facultad:** Se activa después de eliminar un usuario de la tabla Usuario para reducir los puntos totales de la Facultad afectada.
- **Eliminar usuario:** Reduce el número de estudiantes de la Facultad correspondiente después de eliminar un usuario.
- **Insertar usuario:** Se activa tras la inserción de un nuevo usuario para incrementar el número de estudiantes de la Facultad correspondiente.

Este diseño nos permite gestionar eficientemente los datos de los usuarios y las Facultades, manteniendo la integridad y la consistencia de la información en la base de datos.

5.3. Implementación de funcionalidades: back-end

A continuación, procederemos a detallar todo el proceso de desarrollo llevado a cabo para implementar tanto el reto diario como el juego infinito. Analizaremos cada una de las fases relevantes, explicando en profundidad en qué consisten, los métodos utilizados y cómo se implementaron. Abordaremos desde la planificación inicial hasta las pruebas finales, describiendo los desafíos enfrentados y las soluciones adoptadas. Además, se explorarán las tecnologías empleadas y las razones detrás de su elección, proporcionando una visión completa del trabajo realizado para lograr estos objetivos.

5.3.1. Juego Palabra Diaria

Para desarrollar el reto diario inicialmente consideramos el uso de Phaser. Sin embargo, al comenzar el desarrollo nos dimos cuenta de que Phaser está más orientado a la creación de juegos que se basan en una serie de escenas progresivas, por lo que teniendo en cuenta que para este juego no hacía falta más de una escena, decidimos prescindir de ella y utilizar en su lugar Javascript puro.

Reservamos Phaser para el desarrollo del juego extra, ya que las características que proporcionaba se ajustaban mucho mejor a este tipo de juego.

Para la implementación del juego creamos un archivo PHP que sirviera como página web base en la que ir agregando todos los elementos del juego y los scripts necesarios para su funcionamiento. En la Figura 5.2 se puede ver como se presenta esta página principal.

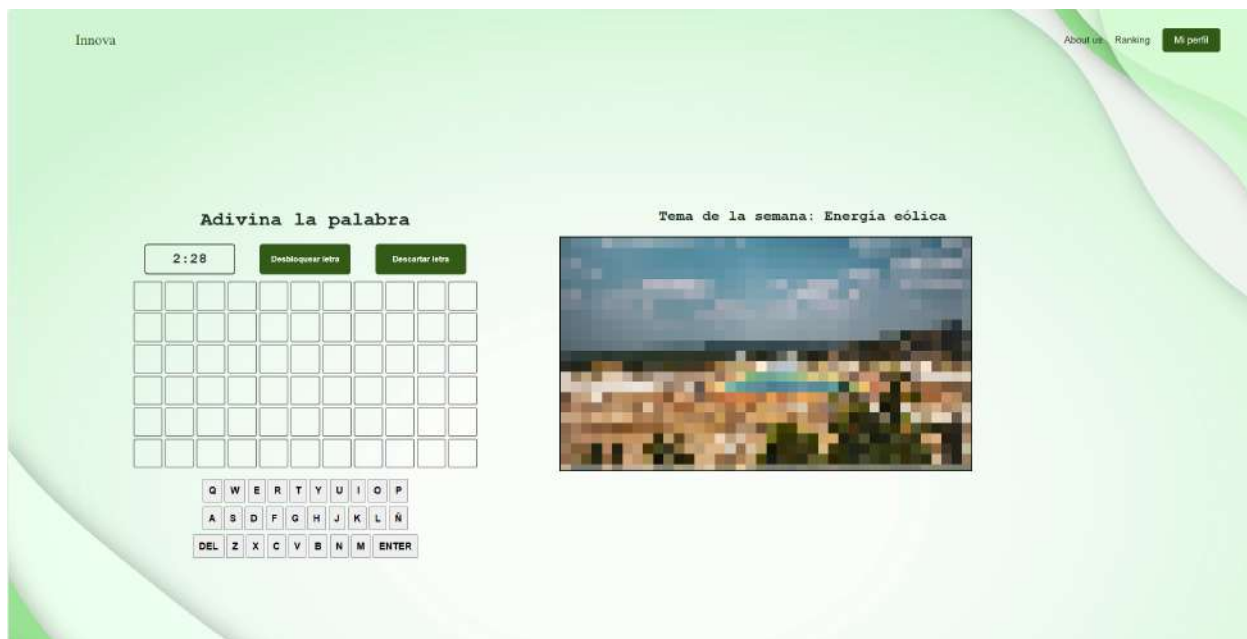


Figura 5.2: *Implementación final del reto diario*

Lista de palabras

Después de analizar varias alternativas para almacenar la lista de palabras que los usuarios deberán adivinar diariamente, optamos por la creación de un archivo Javascript que simplemente contuviera un array en formato JSON con una palabra de la lista en cada posición del mismo. La principal razón por la que decidimos escoger este formato es la facilidad de uso que presenta, ya que incluso aunque en el futuro se cambie el entorno, esta estructura simple permite una integración sin dificultad.

Un aspecto importante de este archivo es que la primera posición del array contiene la fecha en la que el juego pasará a estar disponible para los usuarios. Decidimos implementarlo de esta forma debido a que desde el archivo Javascript que contiene el desarrollo principal del juego se utiliza esta referencia para determinar cuántos días han transcurrido desde el lanzamiento del juego, y así establecer qué palabra corresponde al reto de ese día.

Información de la palabra diaria

Una vez que el usuario haya completado el reto diario, se le proporcionará acceso a una página donde podrá aprender los principales conceptos relacionados con la palabra que ha adivinado. Para llevar a cabo la implementación de esta página nos hemos guiado en el diseño que usamos para la página de la sección “Sobre Nosotros”, ya que al igual que en esa sección, nuestro objetivo aquí también es aportar la mayor cantidad de información posible sin abrumar al usuario con una cantidad excesiva de contenido.

Para obtener los datos necesarios para poder mostrar la información hemos utilizado varias funciones encargadas de leer el JSON para obtener la fecha de lanzamiento del juego y la palabra del día correspondiente. Además, al final de este documento hemos añadido un botón que solo será visible para los usuarios que hayan conseguido ganar el reto diario y que les permitirá acceder directamente al juego extra.

PALABRA DEL DÍA

DENSIDAD

DENSIDAD

¿Qué significa?

La densidad es una propiedad física que se define como la cantidad de masa contenida en un volumen determinado, expresada en unidades como kilogramos por metro cúbico (kg/m³) o gramos por centímetro cúbico (g/cm³). Es una medida esencial para entender cómo se distribuye la materia en un espacio determinado. La densidad nos permite identificar y diferenciar materiales; por ejemplo, el oro es más denso que la madera, lo que significa que una pequeña cantidad de oro pesa más que una mayor cantidad de madera del mismo volumen.

DENSIDAD

¿Qué necesito saber sobre este término?

Para comprender bien este término, es importante saber que la densidad es una propiedad intrínseca de los materiales y no depende de la cantidad de materia presente, sino de la relación entre masa y volumen. Esto significa que dos objetos del mismo material tendrán la misma densidad, sin importar su tamaño. Además, la densidad se ve afectada por la temperatura y la presión; en general, los materiales se expanden y su densidad disminuye al calentarse y se comprimen, aumentando su densidad al enfriarse.

DENSIDAD

¿Cómo puedo aplicarlo en mi día a día?

En la vida cotidiana, la densidad tiene múltiples aplicaciones prácticas. En la cocina, por ejemplo, saber que la densidad del agua es de 1 g/ml puede ayudar a conectar fácilmente entre masa y volumen cuando se mide un ingrediente. En transporte y logística, calcular la densidad de los objetos permite optimizar el uso del espacio y reducir costos de envío. También es crucial en la salud y el fitness, donde se utiliza para calcular la composición corporal y evaluar el porcentaje de grasa corporal. Además, la comprensión de la densidad es fundamental para prever si un objeto flotará o se hundirá en un líquido, lo que puede ser útil tanto en experimentos científicos como en situaciones prácticas diarias.

¿Quiéres conseguir puntos extra?

¡Felicitades! Has ganado el reto diario y ahora tienes acceso al juego extra para acumular más puntos y subir en el ranking. ¡No pierdas más tiempo y accede al juego pulsando el botón que hay justo debajo!

[Accede al juego extra](#)

Figura 5.3: Visualización de la página que contiene la información de la palabra del reto diario

Fichero con procesamiento principal

Este fichero abarca todo el funcionamiento del juego y es, por tanto, el más extenso de todos los relacionados con el reto diario. Para describirlo de manera estructurada presentaremos una lista de las principales funcionalidades que hemos implementado:

- **Obtención de la palabra diaria:** como ya explicamos en el apartado anterior, utilizamos un fichero JSON para almacenar la lista completa de palabras para el reto y la fecha de lanzamiento del juego. Basándonos en esto, para determinar la palabra a mostrar en un reto diario específico, creamos una función que comprueba el día actual y calcula la cantidad de días transcurridos desde la fecha de lanzamiento del juego, utilizando esta cantidad como índice para acceder a la lista de palabras.
- **Temporizador:** para implementar esta funcionalidad decidimos utilizar la librería Luxon^[4] debido a la amplia documentación que presentaba y a su facilidad de uso. Además, esta librería es compatible con *ECMAScript*, lo que garantiza su funcionamiento en entornos Javascript modernos.
- **Sistema de pixelado:** para crear un sistema que pixelara gradualmente la imagen relacionada con la palabra del día, inicialmente intentamos pixelar manualmente las imágenes, creando 6 imágenes para cada palabra cada una con un tamaño de pixel mayor. Sin embargo, esta alternativa resultaba poco eficiente debido a la gran cantidad de espacio que se utilizaba al necesitar 6 fotos para cada palabra. Después de analizar varias opciones, optamos por utilizar la librería Pixelate^[14], que nos permitía pixelar la imagen original desde el propio código y controlando el tamaño de pixel. De esta forma podíamos guardar una sola imagen para cada palabra y pixelarla según fuera necesario, reduciendo mucho el espacio de almacenamiento utilizado con respecto a nuestra idea inicial.
- **Funcionamiento principal:** para la implementación del juego principal creamos una matriz para representar el tablero del juego, utilizando ceros para las casillas no adivi-

nadas y unos para las casillas adivinadas. Siguiendo esta idea, cada vez que el usuario rellena una fila con caracteres, se pasa a comprobar cada posición para dar un color de fondo u otro a la casilla del tablero y se comprueba si el usuario ha adivinado la palabra.

También implementamos ventanas emergentes para informar sobre distintos eventos tales como que al usuario se le acabe el tiempo del temporizador o que se intente comprobar una solución sin haber utilizado todas las casillas de la fila. De la misma forma, cuando se completa el reto se muestra una ventana emergente con la puntuación obtenida y un enlace a la página que contiene la información sobre la palabra. Para la creación de estas ventanas emergentes utilizamos la librería `Popup-js`^[16], ya que permitía la personalización y adición de las ventanas emergentes de una forma muy intuitiva.

Cabe destacar que a la hora de actualizar la puntuación del usuario hacía falta acceder a la base de datos. Después de analizar varias opciones, decidimos realizar esta operación utilizando AJAX. De esta forma, en el momento en el que se muestra la ventana emergente con la puntuación obtenida por el usuario, se envía utilizando AJAX un mensaje a un fichero PHP encargado solamente de recibir ese mensaje y, a través de una instancia del controlador adecuado, llamar a la función encargada de actualizar la base de datos a través del DAO. Esto se realiza de forma análoga para actualizar el campo del usuario que controla si el usuario ya ha realizado el reto de ese día.

- **Botones para desbloquear letras:** para la implementación de estos botones utilizamos javascript puro, intentando conseguir que se realizaran las operaciones de la forma más eficiente posible. En ambos casos, se comprueban los requisitos que se deben cumplir para que el usuario pueda utilizar el botón, se obtiene una letra que cumpla con las condiciones indicadas por el botón, se actualiza la interfaz de usuario y se resta el tiempo correspondiente al temporizador.

5.3.2. Juego Infinito

Para este juego se ha hecho uso de los conocimientos adquiridos en la asignatura DVI(Desarrollo de videojuegos mediante tecnologías web), además de la reutilización y refactorización de ejercicios pedidos para esta asignatura. De esta forma se ha desarrollado un juego con la tecnología Phaser 3.0:

Phaser is a fast, free, and fun, open source HTML5 game framework that offers WebGL and Canvas rendering across desktop and mobile web browsers. Games can be compiled to iOS, Android and native apps by using 3rd party tools. You can use JavaScript or TypeScript for development.

Fuente: Phaser Doc

Configuraciones iniciales

Normalmente, Phaser suele ser implementado con la tecnología **Node.js**, sin embargo, en este caso se ha refactorizado parte del código y buscando una mayor optimización y eficiencia, se ha llevado todo a un archivo, “**phaser.js**”, el cual contiene todo el motor de Phaser. Únicamente al extender este archivo sobre los demás se consigue su funcionamiento, evitando así hacer uso de tecnologías demasiado pesadas e innecesarias.

Para poder hacer uso de Phaser, en primer lugar, se ha tenido que añadir un canvas con la relación de aspecto adecuada en nuestra web donde poder ejecutar el script, que en nuestro caso cuenta con unas dimensiones de 752 x 752 píxeles.

Una vez proporcionado el lugar de la web, Phaser requiere de un fichero, “**game.js**”, el cual contiene una configuración inicial donde debemos destacar principalmente el estilo de juego que se desea implementar; en este caso un diseño de “PIXELART”, ya que nos centramos principalmente en las visualizaciones 2D.

Otros aspectos de esta clase de configuración, son las físicas y movimientos que va a seguir nuestro juego:

```
physics: {
  default: 'arcade',
  arcade: {
    gravity: { y: 100 },
    debug: false
  },
  checkCollision: {
    up: true,
    down: true,
    left: true,
    right: true
  }
}
```

Con esta configuración establecemos por defecto unas físicas arcade, las cuales son más apropiadas para nuestro juego, ya que se requiere un motor más ágil y con menor dificultad para el usuario. También podemos observar que se han activado todos los movimientos. Por último, la variable **debug** sirve para facilitar el trabajo del desarrollador. Al estar activada, podemos ver los *colliders* y vectores, entre otras cosas, para ajustar así el juego con mayor facilidad. Una vez se ha terminado de desarrollar el juego y se pone a disponibilidad del usuario, esta debe estar deshabilitada.

Escenas

Phaser es un motor que viene con funciones predefinidas que ayudan al programador a desarrollar sus juegos, y está basado en escenas. Esto quiere decir que para el correcto funcionamiento del juego Infinito debemos de predefinir qué escenas se van a mostrar. En este caso las escenas son:

- **ProgressScene:** La cual es una escena que únicamente contiene una barra de carga al comienzo del juego. Esta escena se encarga de inicializar todos los elementos del

juego, especialmente las imágenes, que al ser de mayor peso pueden tener una demora de tiempo a la hora de cargarse en la web. Con esto nos aseguramos que siempre que se inicie el juego van a estar todos los recursos disponibles y además el usuario va a tener una mejor experiencia.

- **LevelSelector:** Esta escena es la que se encarga de inicializar todo el juego.
- **GameOverScene:** La cual muestra una imagen junto con un sonido en el que indica que la partida se ha completado y da la oportunidad de resetear todos los valores, para así poder volver a jugar.

Por último, a nivel de configuración mencionar que existen algunos otros ficheros, los cuales contienen funciones y variables muy utilizadas, introducidas en estos ficheros, para así mejorar su legibilidad y comprensión a la hora de programar.

Hasta ahora hemos mencionado configuraciones y ficheros de carácter general, por lo que a continuación pasaremos a ver como está estructurado el juego en sí.

Dentro de la carpeta scenes, es donde tenemos toda la información de las escenas comentadas anteriormente, sin embargo, hay otros ficheros que le añaden funcionalidad, es decir, las escenas son las encargadas de elegir qué elementos se ejecutan y en qué momento, pero la lógica se encuentra en los demás ficheros.

Lógica juego Infinito a bajo nivel

Para mencionarlo brevemente, al igual que scenes, hay una carpeta llamada player, dentro de obj, la cual contiene la información del personaje que estamos utilizando como nuestro personaje controlable del juego, además de los enemigos.

El fichero “**NivelVertical.js**”, es el que realmente contiene toda la lógica, además de los movimientos, colisiones con olas y nubes de polución, y lo que se entiende como el juego en sí mismo.

- **preload():** Traemos todos los recursos necesarios para este nivel, además de crear aquellos elementos que vayan a ser usados en la pantalla como por ejemplo nuestro personaje. Como dato reseñable, destacamos el pad, el cual ha sido añadido gracias a un repositorio de git de código abierto, el cual nos permite que sea accesible desde dispositivos. Esta librería es muy útil y muy ligera, ya que para la creación de los pads hace uso de las funciones de phaser para crear imágenes, en este caso círculos, y así evitar el uso de imágenes que demoren o hagan más ineficiente la aplicación web.
- **create():** Coloca los elementos donde les corresponda. Por ejemplo, la imagen del fondo la coloca como último elemento y el pad como el primero para que se visualice correctamente.
- **colision():** Comprueba que el personaje haya tocado alguno de los *colliders* de los enemigos y lleva a cabo la acción que corresponda, en este caso terminar el juego.
- **updateTimer():** Refresca los valores del temporizador que se está utilizando para calcular la puntuación del juego, de forma que a cada segundo se le suma un punto.
- **generateEnemy():** función que se encarga de generar los enemigos, teniendo en cuenta tanto la elección aleatoria del enemigo, ya que según el tamaño hay varios tipos diferentes, como la posición en la que van a apareciendo en la pantalla. Para conseguirlo se hace uso de una funcionalidad de Phaser que permite generar objetos fuera del canvas, es decir, las nubes y enemigos se generan por encima de la pantalla de juego de forma aleatoria. Al caer, colisionan unas entre otras, haciendo que cuando lleguen al canvas del juego la trayectoria y la posición sea totalmente aleatoria, mejorando así la experiencia de juego.
- **hitPlayer(enemy, player):** Comprueba que el jugador haya sido o no golpeado para permitirle seguir jugando.

- **handleMovement():** Se encarga de realizar los movimientos del jugador, tanto los accedidos desde el teclado como los establecidos mediante el pad.
- **update():** función que Phaser tiene asociada a su biblioteca que se encarga de actualizar periódicamente los elementos que se encuentren dentro del juego. Por tanto, todo aquello que vaya a sufrir algún tipo de modificaciones deberá aparecer en este método. De estas comprobaciones, la más destacable es la condición del usuario, de forma que si este ha sido golpeado, el estado será pasado a “eliminado”, cambiando de escena y parando todos los elementos de esta escena.

Efectos y animaciones

Por último, es importante mencionar cómo se cogen los assets o imágenes del personaje. La gestión de assets en Phaser se realiza principalmente a través de la clase **Loader**, que proporciona métodos para cargar diferentes tipos de archivos. Durante el proceso de desarrollo, los assets se precarga típicamente en la fase inicial del juego, en una función llamada `preload()`. En esta función, el desarrollador especifica los assets que el juego necesitará y Phaser se encarga de cargarlos en la memoria para su uso posterior.

Por ejemplo, para cargar una imagen, se usaría un código como:

```
this.load.image('nombreDeLaImagen', 'ruta/al/archivo.png');
```

Una vez que los assets están cargados, se pueden crear y manipular dentro del juego usando diversas clases de Phaser, como **Sprite** para imágenes o **Sound** para audio.

En Phaser, un sprite no es más que una imagen o un conjunto de imágenes que usamos para representar cosas en el juego, como personajes u objetos. Cuando tienes una imagen grande que contiene varios sprites, a eso lo llamamos una “sprite sheet”. Es como una hoja llena de imágenes en la que puedes seleccionar cuáles quieres usar. En tu sprite sheet, podrías tener varias imágenes del personaje: una parada, otra en pleno paso con el pie derecho, otra con el pie izquierdo, y así sucesivamente. Para hacer que tu personaje se mueva de forma creíble en el juego, necesitarás elegir solo esas imágenes específicas que representan los

movimientos al caminar.

En este caso, el personaje utilizado ha sido el que se muestra en la Figura 5.4, a partir del cual en los ficheros de configuración .json, se han ido eligiendo las imágenes que más nos interesaban, mediante funciones como:

```
createAnimations(){
this.create_Anim("standingRight", 0, 3, IDLE_FRAME_RATE);
this.create_Anim("standingLeft", 4, 7, IDLE_FRAME_RATE);
this.create_Anim("runningRight", 8, 11, MOVI_FRAME_RATE);
this.create_Anim("runningLeft", 12, 15, MOVI_FRAME_RATE);
this.create_Anim("upRight", 8, 8, 1);
this.create_Anim("upLeft", 12, 12, 1);
}

constructor(scene, x, y){
//load of graphics:
this.scene.load.spritesheet(this.graphicName, Utils.getImgH(this.graphicName), {
frameWidth: SPRITE_WIDTH,
frameHeight: SPRITE_HEIGHT
});
}
```

Con estos elementos podemos seleccionar las zonas de la imagen donde están los frames que queremos utilizar y asignar dichos frames a los movimientos deseados. El fichero origen con todos los Sprites, ha sido el siguiente:

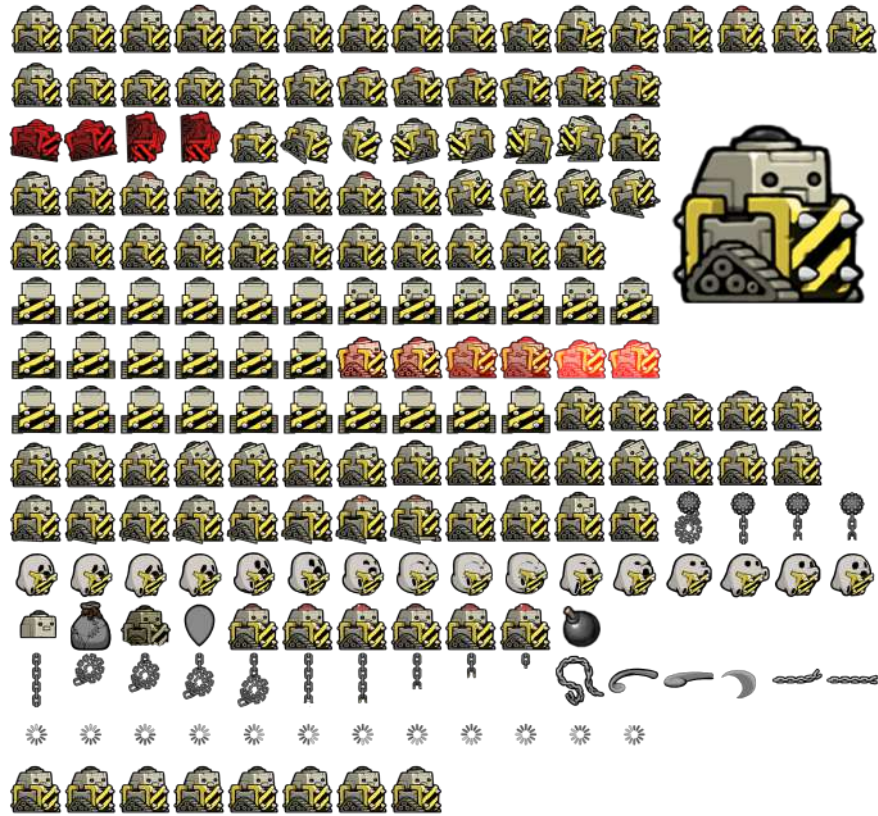


Figura 5.4: *Sprite correspondiente al personaje principal del juego infinito*

En Phaser, esto se maneja fácilmente. Primero, cargas tu sprite sheet en el juego durante la fase de `preload`. Luego, cuando creas un sprite, especificas de qué parte de esa hoja grande, quieres extraer cada imagen. Con Phaser también puedes decirle que tome esas imágenes y las reproduzca en un ciclo, creando así una animación de tu personaje caminando.

5.4. Rendimiento de la aplicación

En este apartado, se mostrará el coste de tiempo y recursos durante la ejecución de la aplicación web. Ya que estos datos dependerán de la página en la que nos encontremos, se hará especial énfasis en las que más recursos consumen, que son el juego infinito y el reto diario. Además, se mostrarán aquellos recursos que siempre se incluyen en todas las páginas web de la aplicación.

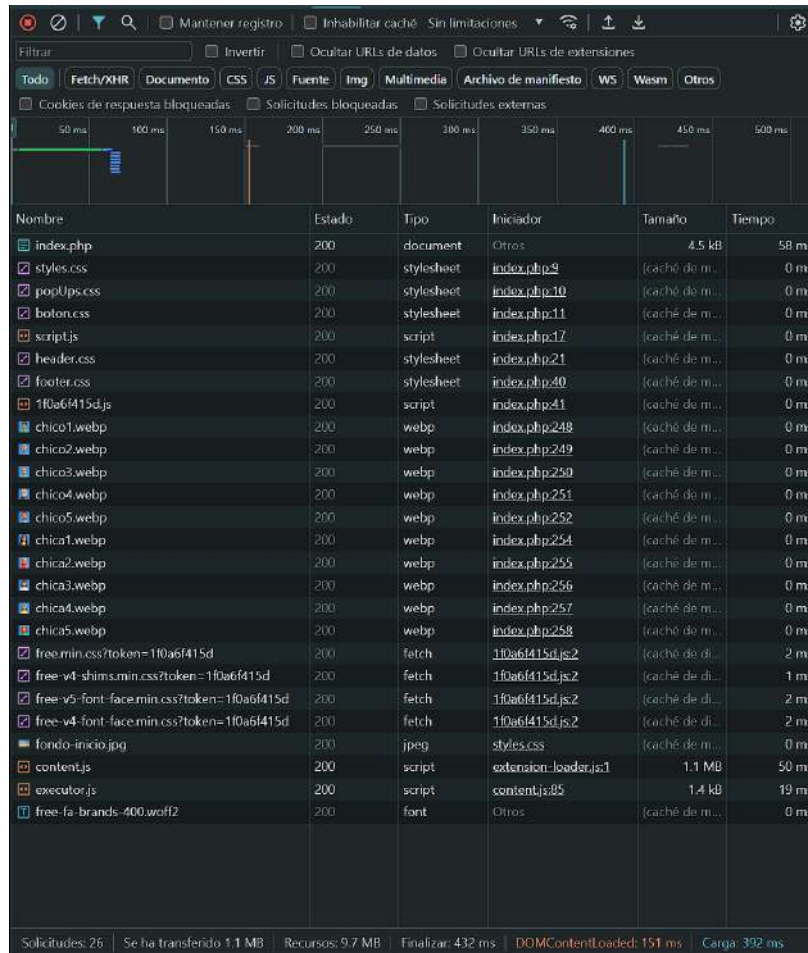


Figura 5.5: Tiempo de carga y recursos consumidos por defecto de la web.

Elementos por defecto

El coste de las páginas siempre van a tener un mínimo y estos son los archivos por defecto que hacen que la página siempre tenga la misma estructura, estos archivos, son por ejemplo el footer, los elementos de css, que hacen que la disposición de los elementos sea la correcta, la carga de los elementos del header ... Como vemos en la Figura 5.5, la página tarda unos 432 ms y se descargan alrededor de unos 10 MB, para una visualización correcta. Esto dependerá en gran medida de nuestra conexión. En términos generales, podemos afirmar que el rendimiento de la web es óptimo para los usuarios que van a interactuar con ella.

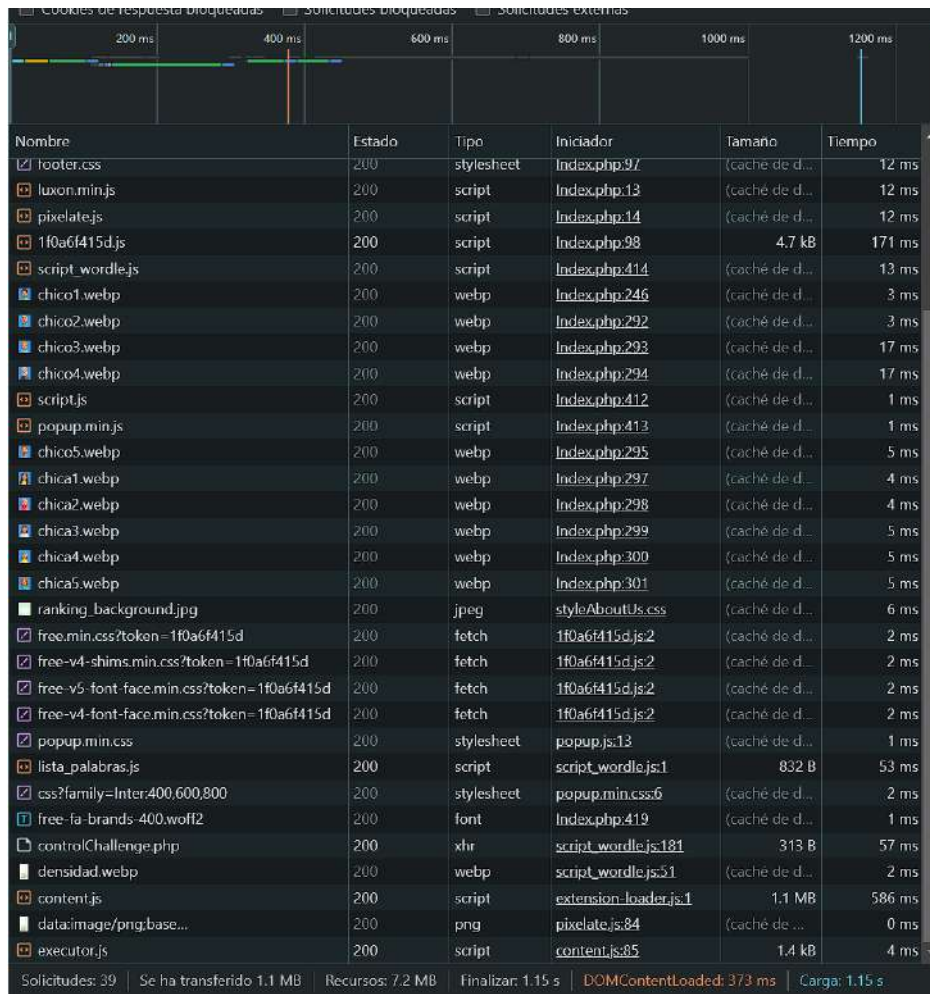


Figura 5.6: Rendimiento durante la ejecución del reto diario

Rendimiento reto diario

El coste de acceder a la página web del reto diario, es algo mayor al del apartado anterior, esto es debido a que carga todos los elementos por defecto, indicados en la sección anterior, además de la lógica y la imagen a des-pixelar.

Sin embargo, a pesar de que la ejecución en tiempo es mayor, alrededor, de 1 segundo, se puede observar en la Figura 5.6, que el tamaño de los archivos descargados es menor, se reduce de unos 10 MB a 7 MB aproximadamente. Esto se debe a la imagen de fondo, que aparece por defecto. En esta página, se cambió la imagen de fondo, para mejorar el rendimiento.

revirtualjoystickplugin.min.js	200	xhr	phaser.js:122932	4.7 kB	290
verticalLevelElements.json	200	xhr	phaser.js:122932	(caché de d...	2
background.jpg	200	xhr	phaser.js:122932	853 kB	345
blob:http://16.170.215.219/a441a5e2-5b55-...	200	jpeg	phaser.js:5071	0 B	1
Solicitudes: 84 Se ha transferido 3.6 MB Recursos: 19.0 MB Finalizar: 2.35 s DOMContentLoaded: 1.14 s Carga: 1.44 s					

Figura 5.7: *Rendimiento página juego infinito*

Rendimiento juego infinito

Como se ha mencionado en apartados anteriores, para conseguir un correcto funcionamiento de Phaser, se creó un archivo javascript, con todas las dependencias, para evitar incluir otras tecnologías y mejorar así el rendimiento. Gracias a esto, podemos ver en la Figura 5.7, que aunque el tamaño de los archivos es el mayor de todas las páginas, alrededor de unos 20 MB, vemos que el tiempo de carga sigue siendo bajo, teniendo en cuenta la cantidad de elementos que interaccionan.

Capítulo 6

Conclusiones y Trabajo Futuro

Para finalizar la memoria de este proyecto, se explicarán las conclusiones obtenidas durante la realización del mismo, así como algunos evolutivos del proyecto.

6.1. Conclusiones

El objetivo de este TFG es poder hacer realidad una campaña de comunicación sobre las energías renovables mes a mes como garantía de un futuro sostenible para la comunidad universitaria complutense.

La decisión final, ha sido desarrollar una aplicación web, ya que algunos de los integrantes del grupo tenían conocimientos acerca de creación de páginas webs, y se estimó que era el medio más accesible para el público objetivo. Para el diseño de esta web se ha intentado conseguir una interfaz con una navegación sencilla y amigable para el usuario, para lo que ha hecho falta refinarla por medio de la elaboración de diferentes mockups y reuniones para mostrar los avances y diferentes puntos de vista.

Una vez diseñada la web, se pasó a la implementación de una serie de juegos haciendo uso de las tecnologías descritas en secciones anteriores. Estos juegos servirían para atraer a los usuarios y conseguir que aprendan conceptos clave sobre energías renovables de una forma divertida. De esta forma, una vez el juego haya finalizado, se mostrará una interfaz que proporcionará información importante sobre la palabra del día, así como consejos que puedan resultar útiles al usuario con respecto al papel de la palabra del día en el uso de

energías renovables. De esta forma se pretende ayudar a la formación continua del usuario en su día a día para la contribución de un mundo más sostenible.

Como idea inicial se propuso la creación de un foro donde pudieran interactuar los usuarios. Sin embargo, ya que los comentarios debían ser revisados para asegurar que los mensajes eran adecuados para la comunidad, se decidió prescindir de su uso. Por tanto, se ha buscado un enfoque más indirecto para la interacción entre los usuarios por medio del uso de un ranking. Este ranking permitirá a los usuarios ver sus posiciones respecto a otros jugadores y así querer ir resolviendo los retos diariamente para mejorar su puntuación y subir puestos en el ranking, incentivando de esta forma el uso regular de la aplicación por parte del usuario, y por consiguiente, la introducción de conocimientos acerca de las energías renovables de una forma continua.

Gran parte de este desarrollo ha sido posible gracias a los conocimientos aprendidos durante estos años en nuestros estudios de informática, tales como:

- **Aplicaciones Web:** gracias a la que hemos adquirido los conocimientos necesarios para el desarrollo de la página web.
- **Bases de Datos:** gracias a la que hemos sido capaces de implementar la base de datos e interactuar con ella por medio de consultas SQL para obtener los datos y mejorar la experiencia de los usuarios.
- **Desarrollo de Videojuegos mediante tecnologías web:** gracias a la que contábamos con experiencia previa en el uso de Phaser, que se ha empleado para el desarrollo del juego infinito basado en escenas.
- **Ingeniería del Software:** gracias a la que se ha podido estructurar correctamente el proyecto y llegar a la fecha de entrega con el proyecto finalizado.
- **Redes, Redes y Seguridad y Ampliación de Sistemas Operativos y Redes:** gracias a las cuales hemos aprendido lo necesario para poder alojar todo el trabajo en

un servidor levantado desde cero por parte de los estudiantes, haciendo uso de las tecnologías AWS, pero creando una máquina Linux accesible por cualquier otra máquina mediante internet, y llevando a cabo todas las medidas de seguridad necesarias.

6.2. Trabajo futuro

Aquí señalaremos algunas ideas que se podrían ir añadiendo en futuras versiones o evolutivos de la página web y del proyecto, pero que no se han podido desarrollar completamente por falta de tiempo o recursos:

- Asignar un nombre de dominio a nuestro servidor, para hacerlo más accesible por parte de los usuarios y más fácil de acceder, ya que ahora mismo se compone de una dirección IP.
- Añadir una comprobación de los registros de usuarios con correos de la Universidad, ya que ahora mismo para registrarse solo es necesario introducir un correo que contenga “@ucm”, pero no se comprueba que sea el que corresponde al usuario que se está registrando. Se estudió la posibilidad de mandar un mail de confirmación a los usuarios, pero por costes se desestimó.
- Establecer un límite en la cantidad de veces que el usuario puede emplear los botones de ayuda en el reto diario.
- Mejorar los controles del juego infinito, ya que a pesar de detectar correctamente los movimientos en el juego portátil, se añade una dificultad al jugar en modo móvil ya que los movimientos son algo más limitados.
- Añadir la posibilidad de compartir la puntuación obtenida a otros usuarios mediante redes sociales, fomentando así el futuro uso de la aplicación por nuevos usuarios.

Bibliografía

- [1] Vanessa Marely Aristizabal Angel. Angular vs. react, dos herramientas poderosas en la web. <https://medium.com/notasdeangular/angular-vs-react-dos-herramientas-poderosas-en-la-web-d5e9abc568ab>, 2023. Actualizado por última vez el 21 de abril del 2023.
- [2] Olivier Biot. Documentación melonjs. <https://github.com/melonjs/melonJS>, 2024. [Online; accedido el 22 de abril de 2024].
- [3] Viviana Bruno. Mysql vs mongodb. <https://blog.infranetworking.com/mysql-vs-mongodb/>, 2018. Actualizado por última vez el 29 de octubre de 2018.
- [4] Isaac Cambron. Documentación luxon. <https://github.com/moment/luxon/>, 2024. [Online; accedido el 13 de marzo de 2024].
- [5] Richard Davey. Documentación phaser. <https://github.com/phaserjs/phaser>, 2024. [Online; accedido el 22 de abril de 2024].
- [6] Allan Larsson y Michael Widenius David Axmark. Guía mysql. <https://downloads.mysql.com/docs/mysql-tutorial-excerpt-8.0-en.a4.pdf>, 2024. Actualizado por última vez el 24 de agosto de 2018.
- [7] The PHP Group. Manual de php - funciones de hashing de contraseñas. <https://www.php.net/manual/es/function.password-hash.php>, 2012. Actualizado por última vez el 29 de noviembre de 2012.
- [8] Mat Groves. Documentación pixijs, 2024. [Online; accedido el 18 de septiembre de 2024].
- [9] Miško Hevery. Documentación angular, 2024. [Online; accedido el 10 de mayo de 2024].

- [10] Incibe. Políticas de privacidad para la pyme: contraseñas. <https://www.incibe.es/sites/default/files/contenidos/politicas/documentos/contrasenas.pdf>, 2018. Actualizado por última vez el 6 de febrero de 2018.
- [11] Rod Jhonson. Documentación java spring. <https://github.com/spring-projects/spring-framework>, 2024. [Online; accedido el 12 de mayo de 2024].
- [12] Yannik Evain Laurent Debrauwer, Sébastien Ferrandez. *Design Patterns en PHP*. 2021. ISBN:978-2-409-03343-8.
- [13] Rasmus Lerdorf. Documentación php. <https://www.php.net/manual/es/index.php>, 2024. Actualizado por última vez el 9 de junio de 2017.
- [14] Miguel Mota. Documentación pixelate. <https://github.com/miguelmota/pixelate>, 2017. Online; accedido por última vez el 11 de junio de 2017.
- [15] Olivier Rollet. *Aprender a desarrollar un sitio web con PHP y MySQL*. 2019. ISBN:978-2-409-01803-9.
- [16] Simón. Documentación popup-js. <https://github.com/SimonDMC/popup-js>, 2024. Online; accedido por última vez el 23 de marzo de 2024.
- [17] Fabio Staiano. *Designing and Prototyping interfaces with Figma*. 2022. ISBN:978-1800564183.
- [18] Michael Stonebraker. Documentación postgresql. <https://www.postgresql.org/docs/current/index.html>, 2024. Actualizado por última vez el 25 de noviembre de 2020.
- [19] Jordan Walke. Documentación react. <https://github.com/facebook/react?tab=readme-ov-file#readme>, 2024. [Online; accedido el 10 de mayo de 2024].

- [20] Dwight Merriman y Eliot Horowitz. Documentación mongodb. <https://www.mongodb.com/docs/drivers/php-laravel-mongodb/current/>, 2024. Actualizado por última vez el 15 de enero de 2024.
- [21] Adrian Holovaty y Simon Willison. Documentación django. <https://docs.djangoproject.com/en/5.0/>, 2024. [Actualizado por última vez el 7 de agosto de 2020].
- [22] Rahul Yadav. Django vs php-mvc vs spring: What are the differences? <https://cracode.com/laravelphp-vs-springjava-vs-djangopython/>, 2022. Actualizado por última vez el 23 de diciembre de 2022.

Apéndice A

Introduction

A.1. Motivation

The effects of climate change have become increasingly present, with a global increase in carbon dioxide emissions being seen¹ and land and ocean temperatures². These effects, to a large extent, are caused by the methods currently used to generate energy, based on the burning of fossil fuels.

In recent years, new ways have been found to generate clean energy through renewable energies that do not produce greenhouse gases or polluting emissions. The problem is that, being a relatively recent discovery, most people are still not fully aware of the importance they have and the role they can play in the future.

That is why we have decided, using the knowledge we have acquired during the study of this degree, to design a web application that helps people to effortlessly understand what renewable energies are, how they can help us and what they can do to contribute to the maintenance . enviroment.

¹CO2: global emissions 1995-2023

²Land and ocean surface: variations in temperature 1880-2023

A.2. Objectives

The main purpose of this project is to design a web application that promotes awareness and knowledge about renewable energies using an attractive method for users and that allows them to acquire the proposed knowledge without having to use a large amount of effort or time. To achieve this objective we are going to determine the main points on which we want to focus.

One of the fundamental issues to take into account initially is obtaining the information on renewable energies that will be included in the application. Regarding this, it is important to highlight that we start from an Innova-Teaching Project whose purpose is to enhance the dissemination and communication of the ecological and socioeconomic benefits of the natural heritage, as well as the integration of conservation needs and sustainable use of biodiversity in the development and application of a significant number of sectoral policies, in order to achieve fair, equitable and lasting development. To achieve this, the Faculty of Information Sciences has carried out a study on the different renewable energies and their contribution to the fight against climate change. This study will be the one that we will use to obtain the key information that will be shown in the application, in order to raise awareness in a clear and precise way among the university community.

Another of the most important topics that we have discussed is the way in which we are going to design the application so that users can learn about renewable energies in a pleasant and entertaining way. After contrasting several possibilities, we have decided to design a game based on daily challenges whose users can use to learn a new concept every day.

Related to the previous point, a scoring system will also be designed that will be presented in a ranking showing both the individual scores and those of each Faculty as a whole. In this way, the aim is to attract as many users as possible and encourage them to try to complete each daily challenge.

For the implementation of the web application, our purpose will be to design a website

with a simple but attractive design for the user, seeking ease of use from the first moment, and being perfectly operational from a computer to a mobile. To do this, several mockups will be designed and different user interfaces will be tested until the most suitable one is found.

As an additional objective, given that this project is carried out in collaboration with the Faculty of Information Sciences, we want this project to serve as an example of how collaboration between scientists, engineers and communicators can promote the creation of new teaching and creation models of scientific content.

A.3. Work Plan

For the development of this project we have followed an agile methodology, holding very frequent meetings between the group members and holding meetings with the tutors regularly every three weeks with the aim of demonstrating the progress made in the project and detecting possible deviations from the original plan.

To define in a more visual way the time expected to be spent on each task, we have designed a Gantt chart (see Figure [A.1](#)) that shows the different tasks with respect to the total time we have to finish the project.

Roadmap proyecto web Innova

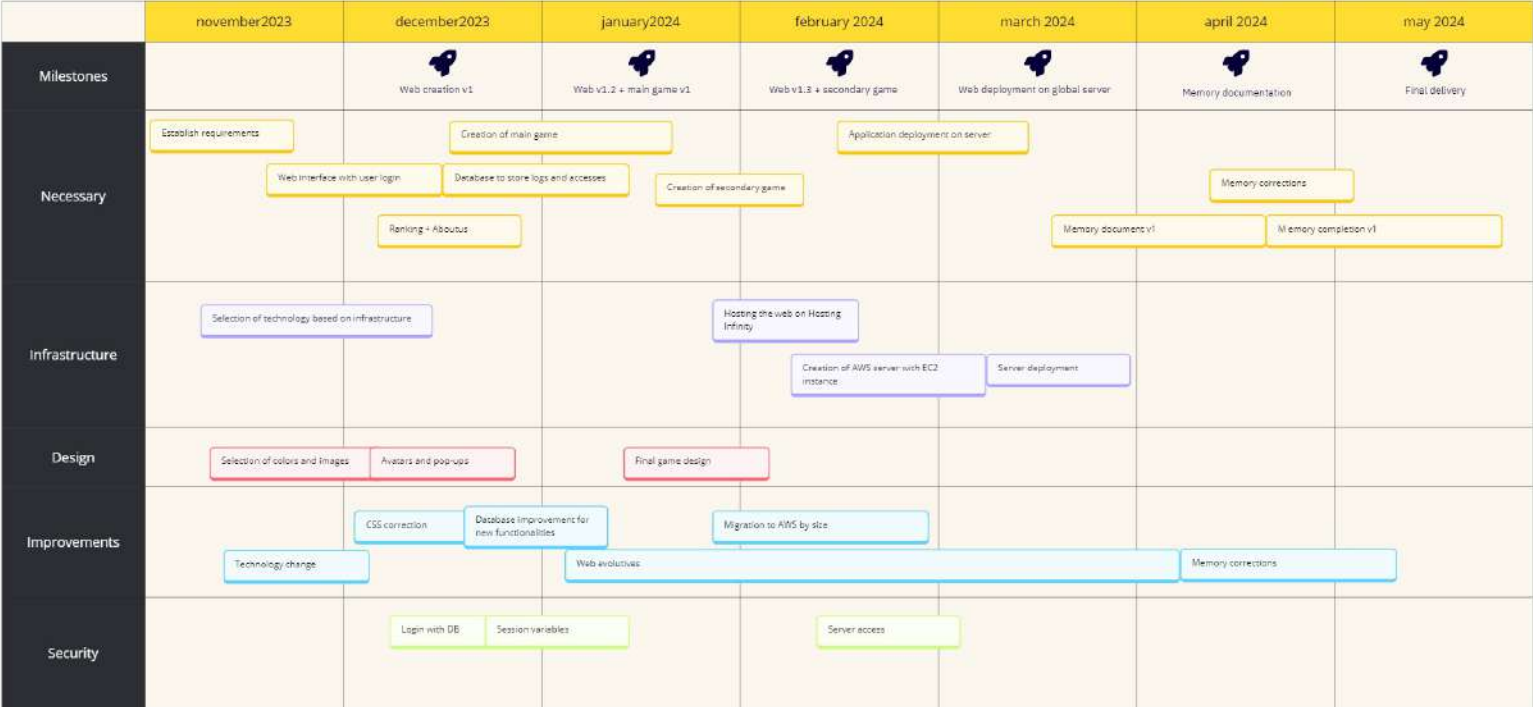


Figura A.1: *Gantt Chart with Estimated Planning*

A.4. Organization of this report

The project report is structured into 6 chapters, each of which addresses one of the aspects related to the development of the proposed application. The organization of each chapter is detailed below:

- **Chapter 1. Introduction:** this chapter shows the general vision of the TFG, describing the motivations that have led us to the development of this application, the objectives that we intend to achieve with this project and the work plan that we are going to follow to meet these objectives.
- **Chapter 2. State of the Art:** this chapter analyzes the different technologies that we have considered using for the development of the application, comparing the different advantages and disadvantages that each of them presents and concluding with an explanation of the final decision that we have made for each technology used.
- **Chapter 3. Application Design:** this section details the application design process, which includes the general architecture, the structure used in the database, the analysis of the different requirements and the design of the use cases, as well as the creation of the prototypes of the user interface of the website and the proposed game.
- **Chapter 4. Code Structuring:** this chapter will detail the hierarchical organization of the folders within the project, explaining the purpose and content of each of them. It will be used as a basis to understand better the implementation of the project detailed in the subsequent chapter.
- **Chapter 5. Implementation of the Application:** this chapter details the implementation of the design established in Chapter 3 using the technologies chosen in Chapter 2, analyzing in depth the entire development process followed both in front-end and in back-end .

- **Chapter 6. Conclusions and Future Work:** this last chapter includes the main conclusions that we have obtained after carrying out this project. The different implications derived from the use of the application are discussed, as well as possible improvements that may be included in future versions.

Apéndice B

Conclusions and Future Work

To conclude the report of this project, some conclusions obtained during its implementation will be explained, as well as some evolution of the project.

B.1. Conclusions

The objective of this TFG is to make a communication campaign on renewable energies a reality month by month as a guarantee of a sustainable future for the Complutense university community.

The final decision was to develop a web application, since some of the group members had knowledge about creating web pages, and it was estimated that it was the most accessible medium for the target audience. For the design of this website, an attempt has been made to achieve an interface with simple and user-friendly navigation, for which it has been necessary to refine it by creating different mockups and meetings to show the progress and different points of view.

Once the website was designed, we moved on to implementing a series of games using the technologies described in previous sections. These games would serve to attract users and get them to learn key concepts about renewable energy in a fun way. In this way, once the game has finished, an interface will be displayed that will provide important information about the word of the day, as well as tips that may be useful to the user regarding the role of the word of the day in the use of renewable energy. In this way, it is intended to help the

user's continuous training in their daily life to contribute to a more sustainable world.

As an initial idea, the creation of a forum where users could interact was proposed. However, since the comments had to be reviewed to ensure that the messages were appropriate for the community, it was decided to dispense with their use. Therefore, a more indirect approach has been sought for the interaction between users through the use of a ranking. This ranking will allow users to see their positions with respect to other players and thus want to solve the challenges daily to improve their score and climb positions in the ranking, thus encouraging regular use of the application by the user, and consequently , the introduction of knowledge about renewable energies in a continuous way.

Much of this development has been possible thanks to the knowledge learned during these years in our computer science studies, such as:

- **Web Application:** Thanks to which we have acquired the necessary knowledge for the development of the website.
- **Data Bases:** Thanks to which we have been able to implement the database and interact with it through SQL queries to obtain the data and improve the user experience.
- **Video game development through web technologies:** thanks to which we had previous experience in using Phaser, which has been used for the development of the infinite scene-based game.
- **Software engineering:** thanks to which it has been possible to correctly structure the project and reach the delivery date with the project completed.
- **Networks, Networks and Security and Expansion of Operating Systems and Networks:** Thanks to which we have learned what is necessary to be able to host all the work on a server built from scratch by the students, using AWS technologies, but creating a Linux machine accessible by any other machine via the Internet, and carrying out all necessary security measures.

B.2. Future Work

Here we will point out some ideas that could be added in future or evolutionary versions of the website and the project, but that have not been able to be fully developed due to lack of time or resources:

- Assign a domain name to our server, to make it more accessible to users and easier to access, since right now it consists of an IP address.
- Add a verification of user registrations with University emails, since right now to register it is only necessary to enter an email that contains “@ucm”, but it is not verified that it is the one that corresponds to the user who is registering . The possibility of sending a confirmation email to users was studied, but due to costs it was rejected.
- Set a limit on the number of times the user can use the help buttons in the daily challenge.
- Improve the controls of the infinite game, since despite correctly detecting movements in the portable game, difficulty is added when playing in mobile mode since the movements are somewhat more limited.
- Add the possibility of sharing the score obtained with other users through social networks, thus encouraging future use of the application by new users.

Apéndice C

Flujo de uso y demo de la Aplicación

En este anexo, se mostrará todas las funcionalidades de la web mediante un flujo de uso normal de un usuario.

A partir de aquí se dividirán en secciones las diferentes acciones que puede realizar el usuario, con qué elementos de la página puede interaccionar, y se mostrará cuál es el comportamiento que se ha pensado para que desarrolle el usuario de forma diaria. Se ha optado por una mecánica y diseño sencillo e intuitivo, el cual pretende que el usuario quiera interactuar diariamente con la aplicación.

C.1. Registro

En primer lugar, para acceder a la aplicación deberás estar registrado en ella, ya que se necesitará guardar el progreso que hayas conseguido, además de poder posicionarte en el ranking. Para ello simplemente se hace clic en la parte superior derecha de la pantalla en el botón inicio de sesión para acceder al formulario de registro que aparece en la Figura C.1. Como dato a destacar, deberemos registrarnos con nuestro correo de la universidad para poder acceder.

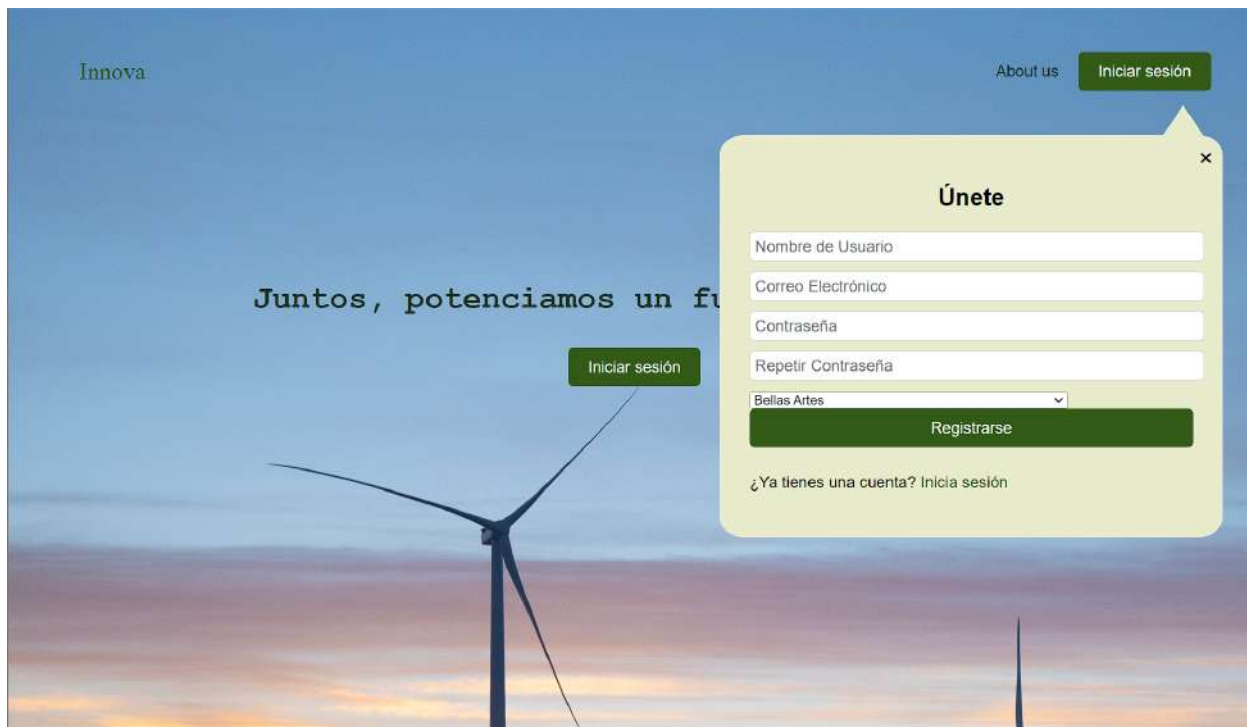


Figura C.1: Registro de nuevos usuarios

C.2. Inicio de sesión

Si ya tenemos un usuario creado únicamente deberemos de hacer clic en el botón alojado en el centro de la página, y añadir nuestras credenciales.

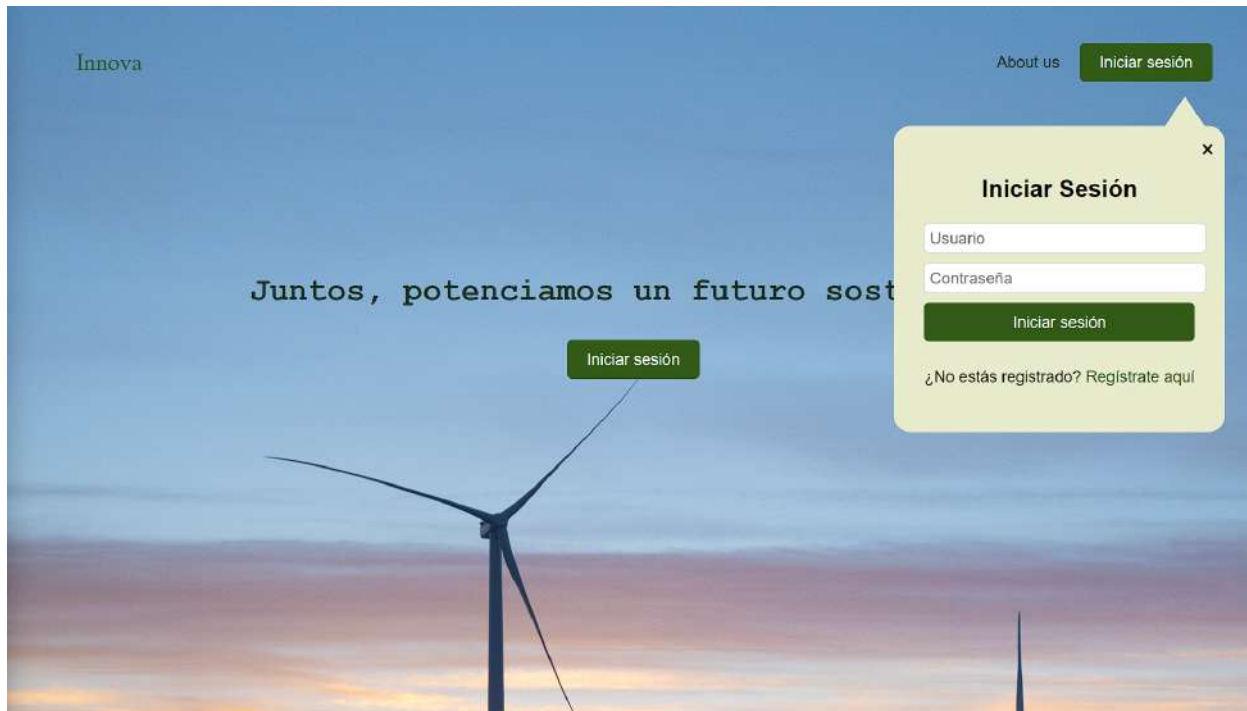


Figura C.2: Inicio de sesión

C.3. Modificación de cuenta

Una vez hemos accedido con nuestras credenciales tendremos acceso al ranking y a modificar algunos aspectos de nuestra cuenta en la parte superior izquierda, como por ejemplo cambiar el avatar de usuario, como se muestra en la Figura C.3, modificar nuestro nombre de usuario, o darnos de baja C.4.



Figura C.3: Cambio de avatar en el usuario



Figura C.4: Opciones de usuario

C.4. Reto diario

Ya dentro de la página con nuestro usuario, tendremos acceso al reto diario mediante los botones del inicio. Al acceder vemos la página donde poder jugar al reto diario [C.5](#), y tras haberlo completado nos aparecerá un mensaje con nuestra puntuación en caso de haberlo superado [C.6](#).

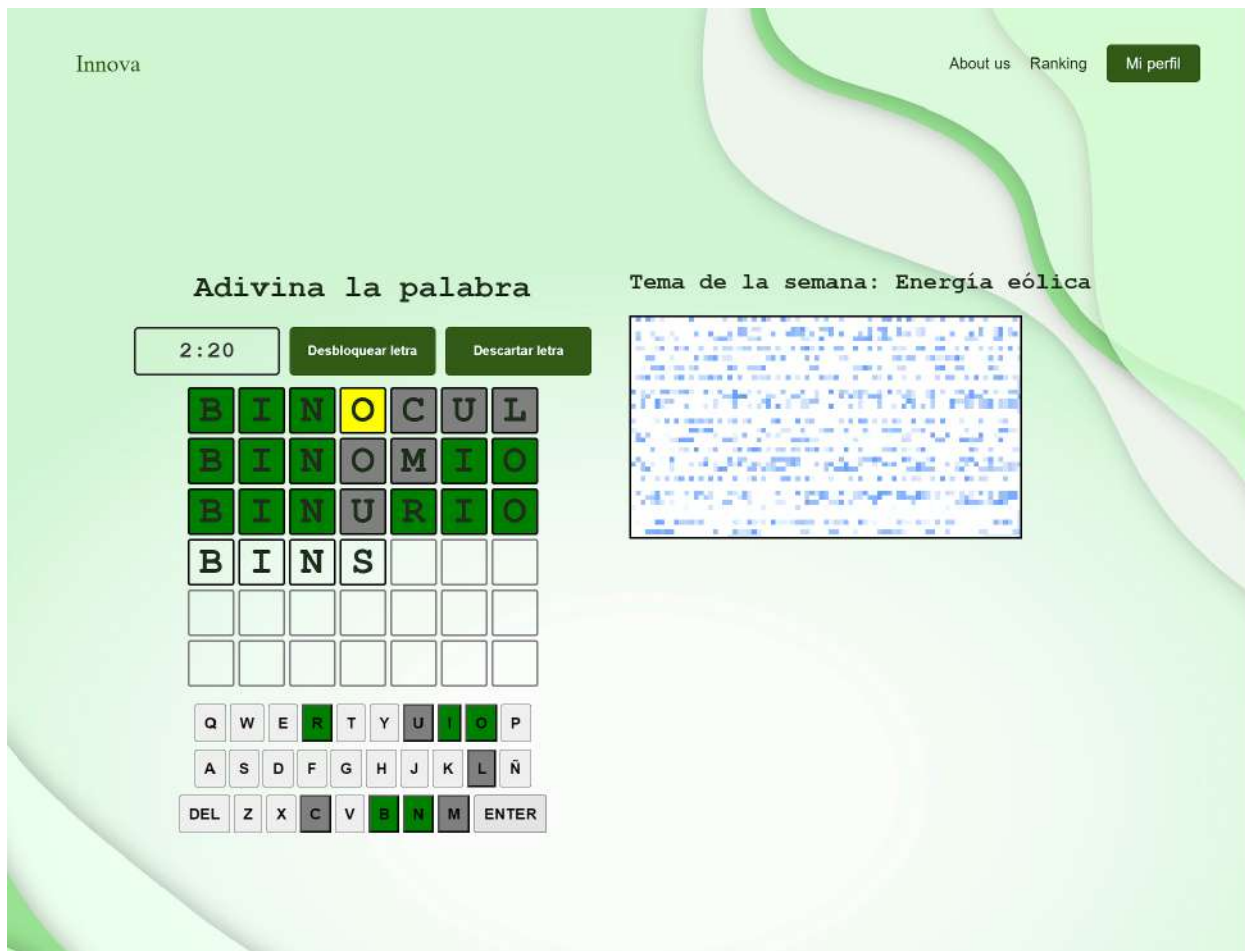


Figura C.5: Reto diario con varios intentos

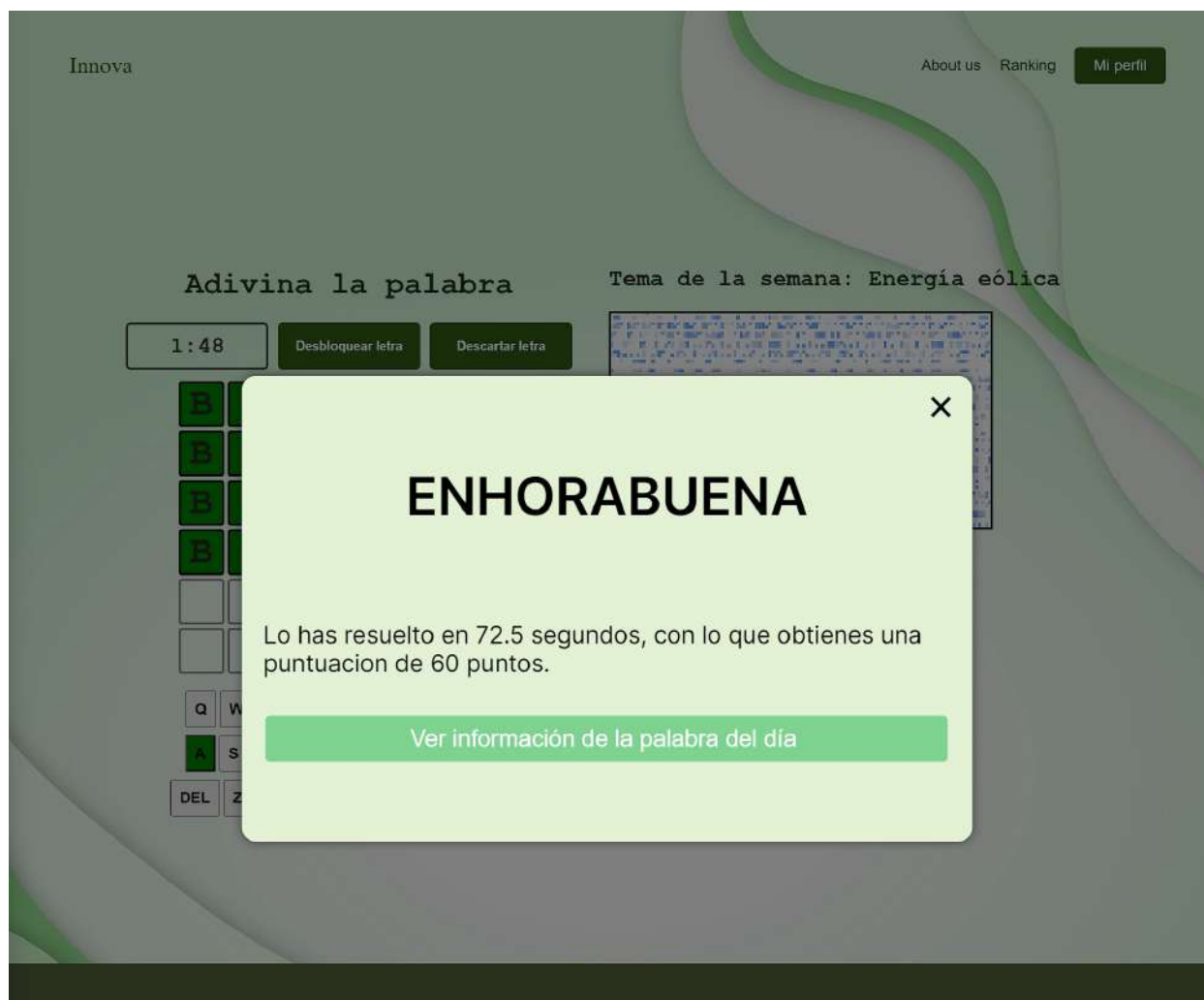


Figura C.6: Reto diario logrado con una puntuación de 60 puntos

C.5. Información palabra del día

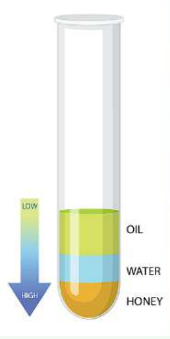
Tras haber completado el reto nos redigirá mediante un mensaje a la página con la información de la palabra diaria **C.7** .

PALABRA DEL DÍA
DENSIDAD

DEFINICIÓN

¿Qué significa?


La densidad es una propiedad física que se define como la relación de masa contenida en un volumen determinado. Expresada en unidades más utilizadas por estos países (kg/m³) se expresa por sencillez como g/cm³. Es una medida esencial para entender cómo se distribuye la materia en un espacio determinado. La densidad nos permite identificar y diferenciar sustancias, por ejemplo, el oro es más denso que la madera, lo que significa que una pequeña cantidad de oro pesa más que una mayor cantidad de madera del mismo volumen.



DEFINICIÓN

¿Qué necesito saber sobre este término?


Para comprender bien este término, es importante saber que la densidad es una propiedad inherente de los materiales y no depende de la cantidad de materia presente, sino de la relación entre masa y volumen. Esto significa que dos objetos del mismo material tendrán la misma densidad, sin importar su tamaño. Además, la densidad se ve afectada por la temperatura y la presión (en gases), por eso cuando se calienta o se comprime, aumentamos su densidad, en sólidos.



DEFINICIÓN

¿Cómo puedo aplicarlo en mi día a día?

En la vida cotidiana, la densidad tiene múltiples aplicaciones prácticas. En la cocina, por ejemplo, sabes que la densidad del agua es de 1 g/cm³ puede ayudar a reconocer fácilmente entre agua y aceite cuando se está cocinando. En transporte y logística, conocer la densidad de los objetos permite optimizar el uso del espacio y reducir costos de envío. También es crucial en la salud y el fitness, donde se utiliza para calcular la composición corporal y evaluar el porcentaje de grasa corporal. Además, la comprensión de la densidad es fundamental para entender si un objeto flota o se hunde en un líquido. Lo que puede ser útil tanto en experimentos científicos como en situaciones prácticas diarias.



¿Quiéres conseguir puntos extra?

¡Felicidades! Has ganado el reto diario y ahora tienes acceso al juego extra para acumular más puntos y subir en el ranking. ¡No pierdas más tiempo y accede al juego pulsando el botón que hay justo debajo!

[Accede al juego extra!](#)

Figura C.7: Visualización de la página que contiene la información de la palabra del reto diario

C.6. Juego infinito

Si hemos completado el juego diario, se nos dará la opción de aumentar nuestra puntuación a través de un juego infinito, el cual sumará la mejor de nuestras puntuaciones a nuestra puntuación general [C.8](#). Destacar que este juego cuenta con un pad que permite jugar desde el móvil sin necesidad de teclado.

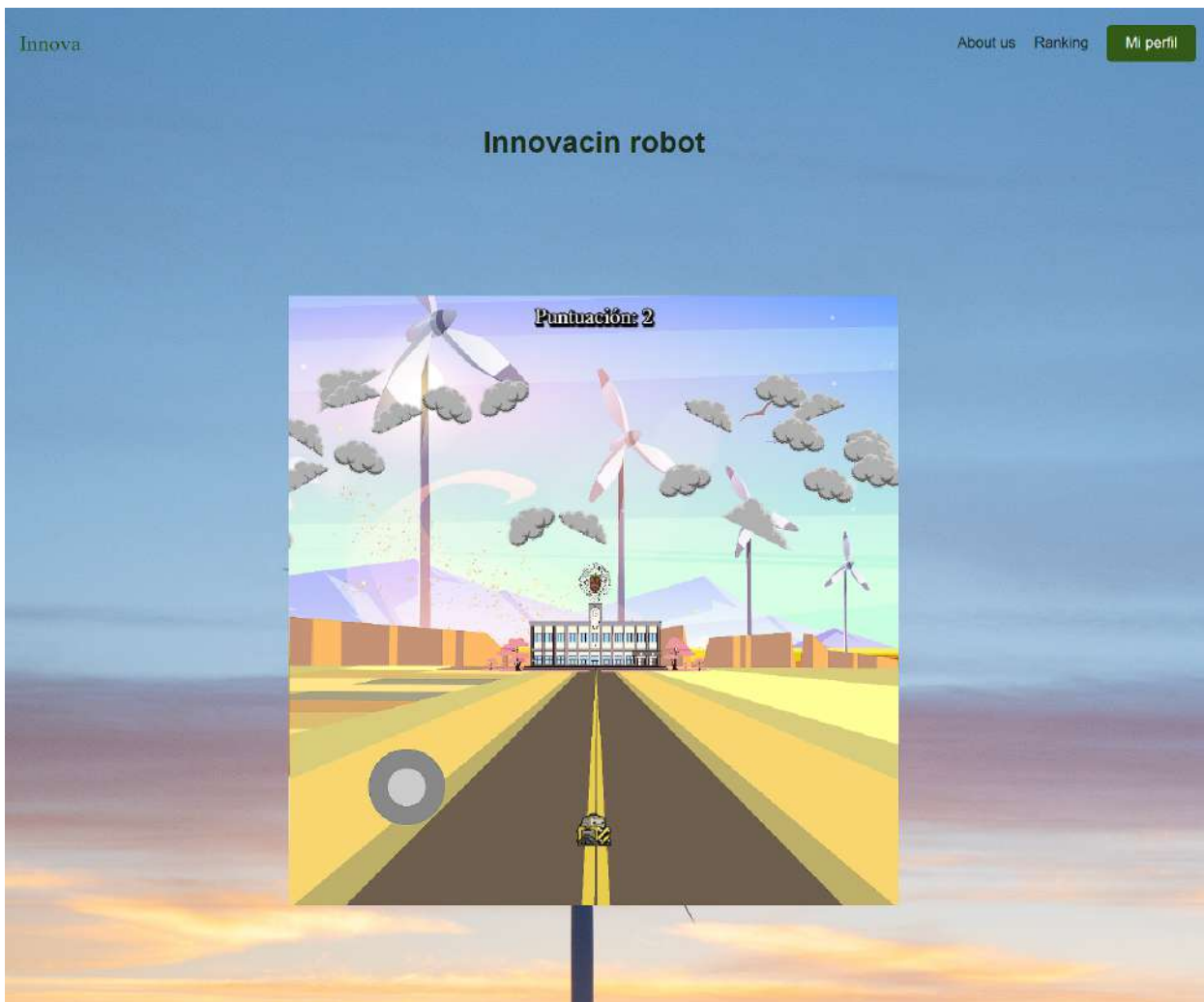


Figura C.8: *Juego inifinito desde PC*

C.7. Ranking

Desde nuestra cuenta, se puede acceder a un ranking donde podemos ver las puntuaciones con respecto a otros usuarios [C.9](#) y a otras facultades [C.12](#)

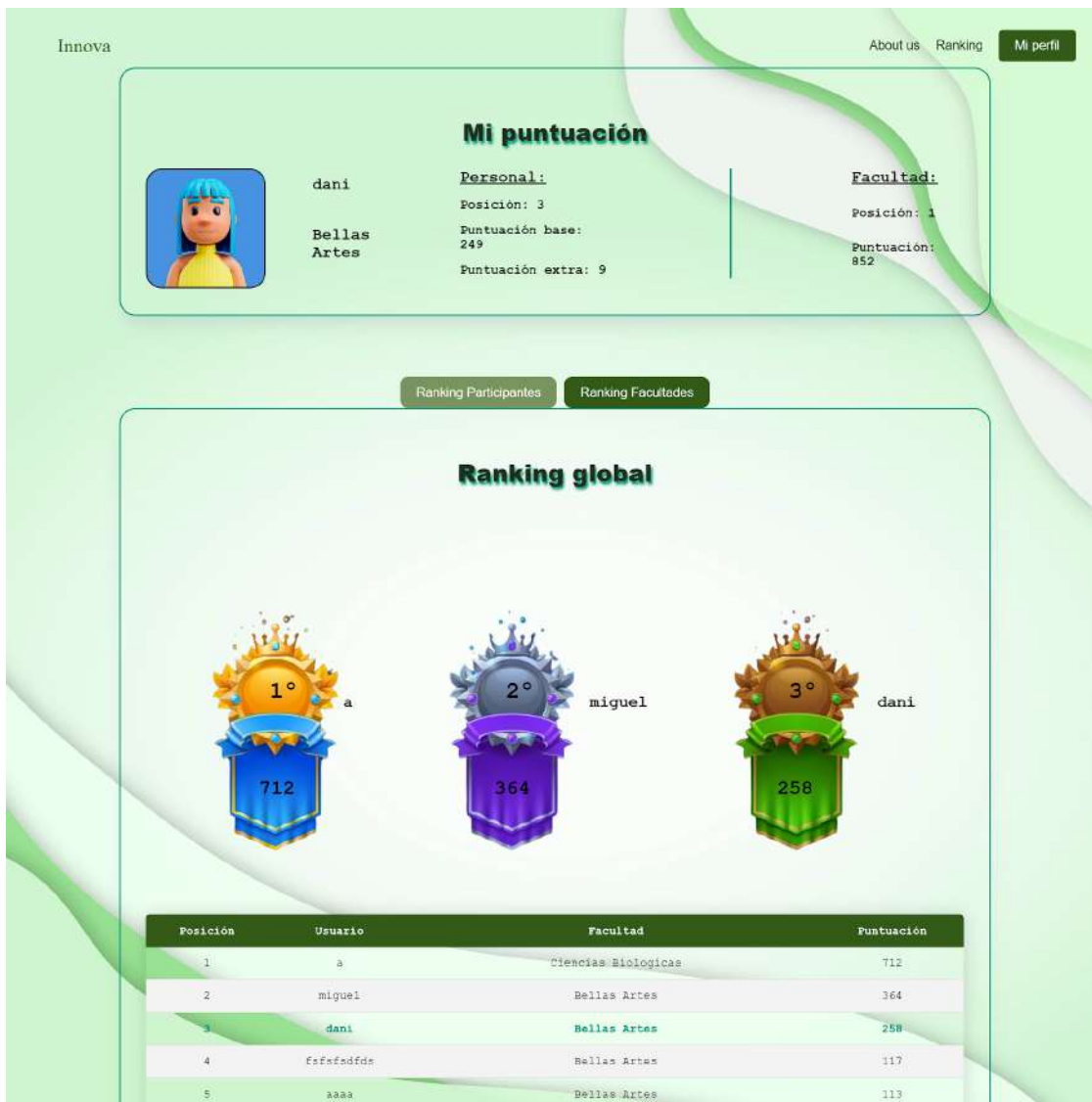


Figura C.9: *Ranking de usuarios*

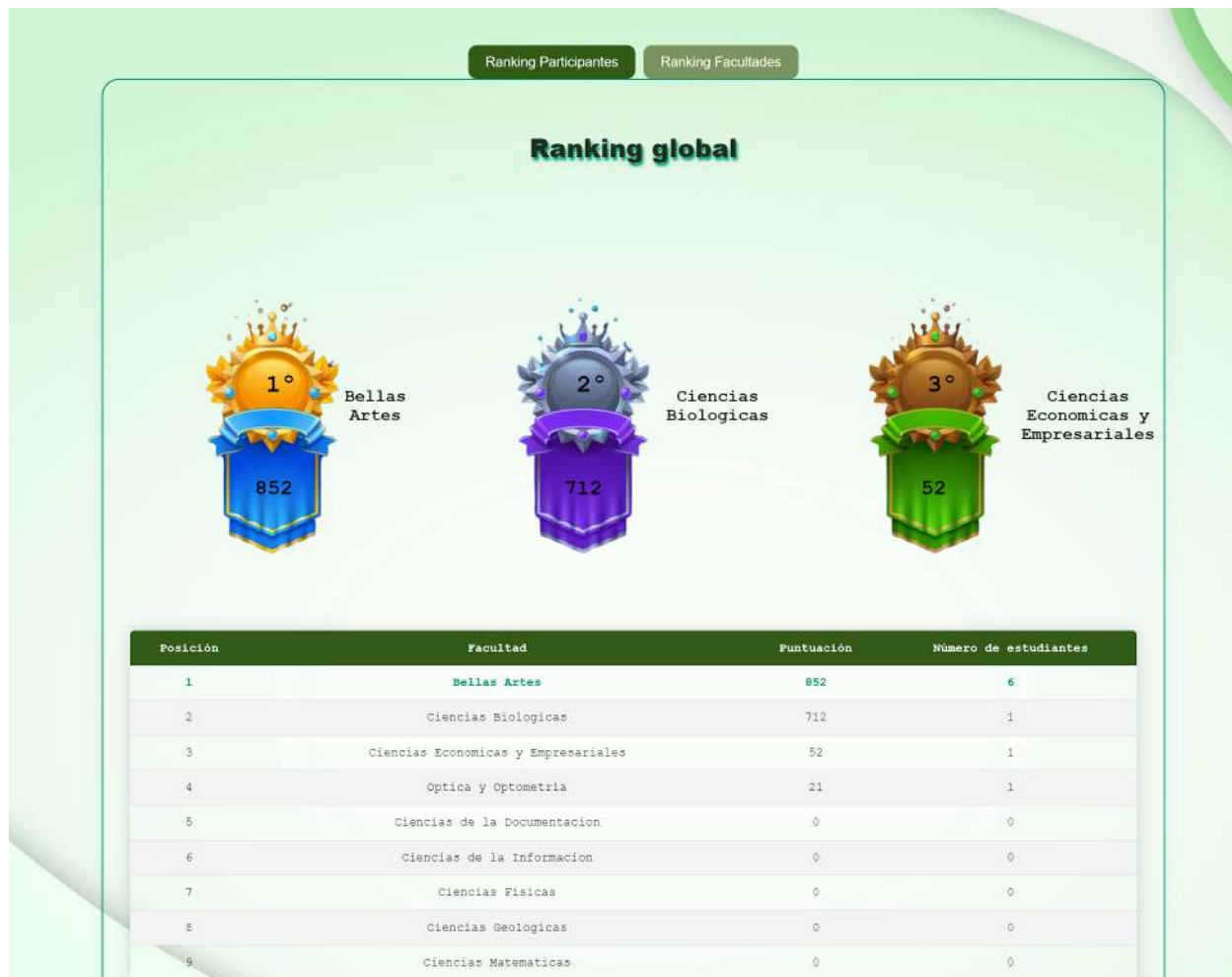


Figura C.10: *Ranking de las puntuaciones de todas las facultades*

C.8. About us

En esta página encontramos la información acerca del proyecto.



Figura C.11: *Página completa About Us*

NUESTRA ASPIRACIÓN

Conseguir un mundo más limpio

A través de nuestro minijuego, queremos inspirar a todos los estudiantes de la Universidad Complutense a tomar medidas concretas para reducir su huella de carbono y adoptar prácticas más sostenibles en sus vidas cotidianas. Creemos firmemente que cada pequeña acción cuenta y que juntos podemos marcar una gran diferencia en la protección de nuestro planeta para las generaciones futuras.



Únete a nosotros en nuestro viaje hacia un futuro más limpio y sostenible. ¡Descubre, aprende y diviértete mientras te conviertes en un defensor activo del uso de energías renovables!

Vamos!

Innova

[Inicio](#) [About Us](#)

© 2023 - 2024 Innova. Todos los derechos reservados.

[Política de Privacidad](#) | [Términos de Servicio](#)

Imágenes diseñadas por Freepik

Figura C.12: *Página completa About Us*

Apéndice D

Aportaciones

En este apéndice, discutiremos cómo hemos dividido el trabajo y las metodologías y dinámicas que hemos empleado para mantener un ritmo de trabajo eficiente. Hemos seguido una metodología ágil para asignar las tareas, lo que nos ha permitido adaptarnos a imprevistos y cambios necesarios, a pesar de tener una planificación inicial. Las tareas fueron distribuidas de manera consensuada y se ajustaron según las necesidades, permitiéndonos colaborar y, en algunos casos, intercambiar responsabilidades por diferentes motivos. Esta flexibilidad ha sido clave para nuestra capacidad de respuesta y eficiencia en el proyecto.

D.1. Gómez Morán, Álvaro

Diseñó y creó un mockup utilizando la herramienta Figma, que sirvió como punto de partida para la implementación del diseño final del proyecto. También fue el responsable de diseñar e implementar los elementos clave de dicha interfaz, incluyendo el header, el footer y la página de inicio, implementándolos de manera responsiva para mejorar la experiencia del usuario.

Utilizó el Patrón DAO (Objeto de Acceso a Datos) para encapsular el acceso a los datos de usuario, separando la lógica de negocio de la lógica de acceso a datos. Esto se llevó a cabo mediante la creación de clases dedicadas (userDAO) y objetos de transferencia de datos (tUser) para manipular la información de usuario de manera eficiente y modular, promoviendo la escalabilidad y la mantenibilidad del sistema.

Creó y desarrolló diversos pop-ups para acciones del usuario, como inicio de sesión, registro, edición de perfil (incluyendo la edición de información como usuario, contraseña y avatar), y eliminación de cuenta. Estos pop-ups no solo se diseñaron visualmente, sino que también se implementaron con funcionalidades de validación de datos, utilizando JavaScript en la capa de presentación, asegurando la integridad de la información del usuario.

Además, implementó los procesos mediante los cuales los datos ingresados por el usuario se reciben a través de solicitudes POST y se envían utilizando los métodos proporcionados por el objeto de servicio de usuario (userSA), garantizando una gestión eficiente y segura de dichos datos a lo largo de la aplicación.

También, desarrolló la gestión de errores para los distintos procesos, como:

- Inicio de sesión fallido debido a nombre de usuario o contraseña incorrectos.
- Registro fallido debido a correo electrónico o nombre de usuario ya existentes.
- Edición de usuario o contraseña fallida.
- Errores generales de MySQL.

Esta gestión de errores se realizó mediante JavaScript en la capa de presentación, proporcionando mensajes claros al usuario y abriendo los pop-ups correspondientes según fuera necesario.

Desarrolló el código necesario para establecer una conexión segura y eficiente con una base de datos MySQL, utilizando el patrón de diseño Singleton, para garantizar que solo existiera una instancia de conexión en todo el ciclo de vida de la aplicación. Esta implementación aseguró la coherencia y la gestión óptima de recursos en el acceso a la base de datos.

Adicionalmente, diseñó las tablas que componen la base de datos, definiendo las relaciones entre ellas, incluyendo en este proceso la identificación de las entidades principales y sus atributos, así como la normalización del esquema para minimizar la redundancia de datos y mantener la integridad de la información almacenada.

D.2. Prieto Remacha, Daniel

Este miembro participó en las primeras reuniones del proyecto, donde se discutieron diversas funcionalidades y requisitos de la aplicación web, además de las tecnologías que se iban a utilizar. Una de ellas era Phaser, ya que Daniel tenía experiencia con esta herramienta y resultaba útil para el desarrollo de los juegos.

Comenzó investigando el uso de Phaser sin necesidad de usar Node.js. Debido a su elevado volumen, era necesario prescindir de esta tecnología, para poder desplegarlo en un servidor sin afectar su funcionamiento. Tras consultar la documentación, logró migrar todas las dependencias a un archivo JavaScript, lo que permitía ejecutar Phaser con mayor rapidez.

Daniel se encargó de crear la primera versión del reto diario, inicialmente planeado para realizarse con Phaser. Sin embargo, tras estudiarlo, y hacer algunas pruebas, descartó esta idea porque el juego del reto diario no requería de “escenas”. Por ello, optó por realizar todo el reto con JavaScript. Una vez terminada la primera versión, sirvió para generar nuevas ideas y funcionalidades en reuniones futuras, haciendo el reto diario más completo.

Desarrolló el juego infinito, aprovechando su investigación inicial para implementar Phaser sin usar Node.js. Primero, planificó el número de “escenas” y su orden en el juego, que consistía en una pantalla inicial de carga (que, dependiendo del navegador o la conexión a internet, es imperceptible), una escena donde se desarrollaría el juego y una final donde se enviaría la puntuación y se ofrecería la oportunidad de reintentar. Para este desarrollo, utilizó la documentación del curso *Desarrollo de videojuegos mediante tecnologías web* y ejercicios propuestos.

Investigó para encontrar un lugar donde alojar la página web a bajo costo y que proporcionase soporte para Apache y MySQL. Tras consultar diversos servicios, optó por usar un servidor ya configurado, en este caso, InfinityFree. Después de realizar las modificaciones correspondientes en cuanto a rutas y dependencias, este servicio no pudo ejecutar el reto diario correctamente debido al tamaño de algunos archivos.

Tras todos estas limitaciones, este miembro, probó otros servidores para alojar la apli-

cación web, pero sin éxito, debido a limitaciones o costos. Finalmente, tuvo que aprender a trabajar con las tecnologías de AWS, que aparentemente podían cumplir los requisitos necesarios.

Finalmente, levantó una instancia de EC2, con los servicios de una máquina Ubuntu, a la cual instaló Apache y MySQL. Tras hacer el servidor accesible a través de internet, comprobó que todos los elementos funcionaban correctamente, y dejó la página alojada en internet, para que cualquier usuario pudiese acceder a ella.

D.3. Zayas Boíza, Miguel

Antes de comenzar el desarrollo del proyecto, Miguel se dedicó a repasar sus conocimientos previos de PHP, Javascript y MySQL para poder trabajar de una forma eficiente durante la elaboración de la aplicación.

Comenzó su trabajo diseñando y desarrollando la página “Sobre Nosotros” con la introducción a la web y la página de ranking que muestra a los usuarios su puntuación actual en el reto diario y en el juego extra y la de la facultad a la que pertenece. Para llevar esto a cabo se encargó tanto de la obtención de datos desde la base de datos como de la implementación de la página web que mostrara toda la información.

Una vez se comenzó el desarrollo del reto diario, dado que su compañero Daniel se estaba encargando de la creación del juego, se dedicó a añadir el resto de elementos necesarios en la página para el funcionamiento del reto. En primer lugar, revisó la documentación aportada por la Facultad de Ciencias de la Información para diseñar la lista de términos a emplear en los retos diarios. Seguidamente, se encargó de la implementación del temporizador y los eventos que este podía lanzar, para lo que hizo falta investigar varias librerías hasta encontrar la librería Luxon. A continuación se dedicó a la implementación de los botones de ayuda para el reto diario. Por último, se dedicó al desarrollo del sistema de pixelado de imágenes del reto diario. En un principio se implementó de tal forma que para cada término diario se guardarán 6 imágenes con distinto tamaño de píxel, de forma que se pasara a la

siguiente imagen cada vez que el usuario introdujera un intento en el tablero. Sin embargo, a pesar de que funcionaba correctamente, esta opción requería más recursos de los que se disponían, por lo que se decidió buscar una librería que permitiera pixelar las imágenes en el momento. Después de probar varias librerías distintas, encontró Pixelate, que permitía realizarlo de forma sencilla.

Una vez estaba terminado el procesamiento principal del reto diario, pasó a implementar la página dedicada a mostrar la información sobre el término usado en el reto diario. Algo a destacar es que durante la implementación de esta página surgieron dificultades para obtener el término y la información a mostrar en la página, por lo que Miguel tuvo que estudiar varias opciones hasta conseguir solucionar el problema con el uso de AJAX.

Siguiendo la misma técnica usando AJAX que para el paso anterior, se dedicó al diseño del control de acceso de los usuarios al reto diario y al juego extra y al control de puntuaciones de ambos juegos para que se actualizaran para cada usuario tanto en la base de datos como en el ranking. Para realizar esto tuvo que añadir algunos campos nuevos en la base de datos y gestionar el back-end para almacenar tres valores nuevos: la mejor puntuación obtenida en el juego extra, la fecha del último reto diario realizado y un booleano indicando si el reto se había ganado o no.

Por último, con relación al desarrollo de la memoria del proyecto, tanto él como sus compañeros tuvieron que utilizar la plataforma Overleaf para la redacción de la misma, por lo que, al no con experiencia previa, se vieron obligados a aprender desde cero.