



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una Plataforma Web para la Preparación de
Estudiantes de Segundo de Bachillerato para la EVAU

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Molina Gandía, Roberto

Tutor/a: Valderas Aranda, Pedro José

CURSO ACADÉMICO: 2023/2024

Resumen

Descripción: El proyecto tratará sobre el desarrollo de una aplicación web llamada 'PrepEVAU' que incluirá varias funcionalidades orientadas a la ayuda de estudiantes que se estén preparando para los exámenes de acceso a la universidad. Entre estas se encontrarán distintos problemas de nivel de bachillerato, así como sus soluciones y distintas lecciones y vídeos recomendados sobre realización de problemas de matemáticas y física.

Palabras clave: prepEVAU, servidor, usuario, base de datos, selectividad, matemáticas, física, ejercicios, exámenes

Abstract

Description: The project will deal with the development of a web application called 'PrepEVAU' that will include several functionalities aimed at helping students who are preparing for university entrance exams. Among these you will find different high school level problems, as well as their solutions and different recommended lessons and videos on completing mathematics and physics problems.

Key words: prepEVAU, server, user, database, selectivity, maths, physics, exercises, exams

Índice general

Índice general	II
Índice de figuras	IV
<hr/>	
1 Introducción	1
1.1 Objetivos	1
1.2 Estructura de la memoria	2
2 Estado del arte	3
2.1 Plataformas web similares.	3
3 Metodología	5
3.1 Diseño Centrado en el Usuario (DCU).	5
4 Análisis de necesidades	7
4.1 Investigación Cualitativa	7
4.1.1 Resultados del Cuestionario	8
4.2 Técnica de Personas	14
4.2.1 Lucas Martínez Almeida, 18 años	14
4.3 Escenarios	15
4.3.1 Escenario 1. Búsqueda de exámenes, ejercicios y documentación.	15
4.3.2 Escenario 2. Uso de la biblioteca.	15
5 Análisis conceptual y Diseño	16
5.1 Diagrama de clases	16
5.2 Modelo de la BD que use el servidor	18
5.3 Bocetos de las interfaces de la app	20
6 Desarrollo de la solución	27
6.1 Arquitectura	27
6.1.1 Frontend.	28
6.1.2 Backend.	29
6.2 Contexto tecnológico	31
6.2.1 Lenguajes	32
6.2.2 Programas y tecnologías	32
6.3 Ejemplos de código	33
7 Producto desarrollado	37
7.0.1 Escenario 1	37
7.0.2 Escenario 2	40
8 Validación	42
8.1 Metodología de Evaluación Heurística	42
8.2 Resultados	43
8.3 Conclusiones de la validación	45
9 Conclusiones	46
Bibliografía	48
<hr/>	
Apéndice	

A Relación con las ODS	49
A.1 Objetivos de Desarrollo Sostenible (ODS)	49
A.1.1 Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS)	49
A.1.2 Reflexión sobre la relación del TFG/TFM con los ODS	49

Índice de figuras

1.1	Logotipo de PrepEVAU	1
2.1	Plataforma educaciongratuita.es	3
2.2	Plataforma selectividad.com	4
3.1	Fases del proceso DCU.	5
4.1	Género	8
4.2	Curso	8
4.3	Rama	8
4.4	Manejo de las tecnologías	9
4.5	Dispositivo para estudiar	9
4.6	Uso de las tecnologías	9
4.7	Redes sociales	10
4.8	Perfil propio o invitado	10
4.9	Mayores desafíos	10
4.10	Recursos usados.	11
4.11	Ayuda adicional	11
4.12	Recursos adicionales	12
4.13	Valoración de recursos adicionales	12
4.14	Uso de aplicaciones de organización del curso	12
4.15	Aplicaciones utilizadas	13
4.16	Realización de ejercicios prácticos	13
4.17	Asignatura de más tiempo de estudio	13
5.1	Diagrama de clases.	17
5.2	Diseño de la base de datos.	18
5.3	Creación de un índice en Xampp.	19
5.4	Interfaz de Xampp para edición de tablas.	19
5.5	Interfaz inicial de la aplicación.	20
5.6	Diseño de la interfaz de Registro.	21
5.7	Interfaz final de Registro.	21
5.8	Interfaz del menu principal.	22
5.9	Interfaz del modo exámen.	22
5.10	Diseño de interfaz de visualización de ejercicios.	23
5.11	Diseño de interfaz de visualización de temario.	23
5.12	Interfaz final de visualización de temario.	24
5.13	Diseño de interfaz de biblioteca I.	24
5.14	Diseño de interfaz de biblioteca II.	25
5.15	Interfaz final de biblioteca.	25
5.16	Diseño de la interfaz de modificación del perfil.	26
6.1	Modelo basado en 3 capas.	28
6.2	Ejemplo del formulario del modo exámen.	28

6.3	NodeJS + Express.	29
6.4	Logotipo de Nodemon.	30
6.5	Fragmento de código del método iniciar sesión.	33
6.6	Fragmento de código del frontend del modo examen.	34
6.7	Fragmento de código del backend del modo examen.	34
6.8	Fragmento de código HTML del menú principal.	35
6.9	Fragmento de código de carga de los datos de la sesión.	36
7.1	Inicio de sesión en modo invitado.	37
7.2	Modo invitado.	38
7.3	Menu principal de la aplicación	38
7.4	Modo exámen	38
7.5	Modo entrenamiento.	39
7.6	Modo entrenamiento (Solución).	39
7.7	Modo visualizar temario	39
7.8	Registro de usuario	40
7.9	Modo entrenamiento (Usuario Registrado)	40
7.10	Inicio de sesión de usuario registrado	41
7.11	Biblioteca del usuario	41
7.12	Modificación del perfil de un usuario	41
8.1	Error añadiendo ejercicio a la biblioteca	43
8.2	Mensaje de usuario incorrecto	44
8.3	Mensaje de usuario incorrecto	44

CAPÍTULO 1

Introducción

El proyecto del que trata este trabajo de fin de grado consiste en el desarrollo de una aplicación web diseñada para ayudar a los estudiantes de bachillerato en la preparación de la Evaluación para el Acceso a la Universidad (EVAU). La EVAU es la prueba más importante para un alumno que aspire a estudiar una carrera, ya que su nota le permitirá o no estudiar lo que le gusta en un futuro. Muchos estudiantes enfrentan dificultades al organizarse y encontrar recursos adecuados para su estudio, por lo que es necesaria una aplicación que aporte las herramientas necesarias para ayudar al alumno a conseguir la mayor nota posible.



Figura 1.1: Logotipo de PrepEVAU

El nombre de la aplicación web es 'PrepEVAU' (figura 1.1) y estará desarrollada inicialmente como apoyo a las asignaturas de matemáticas y física. Las funcionalidades que implementará serán:

- Exámenes y soluciones: Será una parte donde el usuario podrá encontrar cualquier examen de EVAU en 'pdf' con soluciones filtrando por año, convocatoria o comunidad autónoma.
- Generación de ejercicios aleatorios: Funcionalidad que permite al usuario generar un ejercicio aleatorio de la temática que seleccione, pudiendo acceder a su solución y guardar el ejercicio en su biblioteca para revisar en cualquier momento.
- Visualización de temario: El usuario podrá acceder a distintos documentos (teoría, formularios, etc) de la temática que seleccionen.

1.1 Objetivos

El principal objetivo de este proyecto es desarrollar una aplicación web funcional y accesible que ayude a los estudiantes de bachillerato a estudiar y preparar sus exáme-

nes de la EVAU de manera mucho más eficiente. Para lograr esto, se han establecido los siguientes objetivos específicos:

- Proporcionar una herramienta interactiva y fácil de usar: Diseñar una aplicación intuitiva que permita a los estudiantes acceder a exámenes y recursos de estudio de manera sencilla.
- Facilitar la organización del estudio: Ofrecer funcionalidades como la simulación de exámenes y la posibilidad de guardar ejercicios en la biblioteca para ayudar a los estudiantes a gestionar su tiempo y estudiar de forma eficiente.
- Garantizar una experiencia de usuario optimizada: Utilizar principios de usabilidad y realizar una evaluación heurística para asegurar que la aplicación sea fácil de navegar y utilizar para todos los estudiantes.

1.2 Estructura de la memoria

La memoria del proyecto se estructura en los siguientes nueve capítulos:

1. Introducción: Es el capítulo actual. Se presenta el proyecto, los objetivos a conseguir y la estructura del resto de la memoria.
2. Estado del arte: Se realiza un análisis de las herramientas existentes que ofrezcan un producto similar.
3. Metodología: Explicación de la metodología utilizada para el desarrollo del proyecto, que será el Diseño Centrado en el Usuario (DCU).
4. Análisis de necesidades: Evaluación de usuarios potenciales mediante cuestionarios, elaboración de la técnica de personas y creación de escenarios de uso de la aplicación.
5. Análisis conceptual y diseño: Se describe la estructura de la base de datos mediante el diagrama de clases y el diagrama entidad-relación. También se expondrán los bocetos de las interfaces de la aplicación.
6. Desarrollo de la solución: Explicación de las partes de la arquitectura del sistema, definiendo los lenguajes y tecnologías utilizados, además de algunos ejemplos de código.
7. Producto desarrollado: Aquí se realizará una guía de uso siguiendo los escenarios creados en la fase de análisis de necesidades (4). Se acompañará con capturas de pantalla de las interfaces del producto final.
8. Validación: Se realiza una evaluación heurística para comprobar que la aplicación cumple con todos los aspectos de usabilidad y para corregir algún problema que se haya podido encontrar.
9. Conclusión: Resumen de los resultados obtenidos y opinión personal del proyecto desarrollado.

CAPÍTULO 2

Estado del arte

En la actualidad se puede encontrar en internet infinidad de información y recursos que ayudan al alumno a la hora de estudiar y prepararse para sus exámenes. En los últimos años se han ido incrementando la cantidad de webs o aplicaciones que ofrecen este tipo de servicios, de una forma o de otra. El objetivo de esta sección es analizar algunas de estas plataformas web, en especial aquellas que se enfocan en la preparación de la EVAU.

Tras la revisión de estas aplicaciones, no solo ayudará al correcto desarrollo de Pre-EVAU, sino que se permitirá identificar las limitaciones que tienen para poder construir una herramienta mucho más efectiva y con funcionalidades más adaptadas a lo que necesita el alumno de bachillerato hoy en día para estudiar correctamente la EVAU.

2.1 Plataformas web similares.

Una de las páginas web más populares para documentación en internet es 'educaciongratuita.es' (Ver figura 2.1). En esta aplicación puedes encontrar una gran cantidad de información de todos los cursos de la educación desde primaria hasta bachillerato. También te explica todo lo relacionado a cómo cursar los distintos tipos de cursos y qué temarios te vas a encontrar. Se trata de una plataforma muy general, la cual está destinada a un grupo mucho más grande de usuarios que Pre-EVAU, lo cual hace que tenga ciertas limitaciones en las funcionalidades que ofrecen para la preparación de exámenes, ya que, aunque cuentan con una amplia base de datos con temario de todas las asignaturas y de exámenes de otros años, se echa en falta más preparación con un enfoque práctico. Esto lo consigue Pre-EVAU con sus funcionalidades de mostrar los exámenes y ejercicios con soluciones para que el alumno pruebe por sí mismo a hacer los ejercicios que se encontrará en sus exámenes.



Figura 2.1: Plataforma educaciongratuita.es

Otra de las páginas web que puedes encontrar es 'selectividad.com' (Ver figura 2.2). En este caso si que se trata de una aplicación más enfocada a la EVAU, ya que ofrece una gran cantidad de exámenes de todas las comunidades autónomas y de asignaturas, además de que te da la posibilidad de guardar apuntes y documentación. La gran limitación que tiene es que no cuenta con exámenes y ejercicios resueltos para que los usuarios puedan comprobar sus ejercicios por ellos mismos. Esta web tiene otro defecto, no te deja acceder a ninguna de sus funcionalidades si no inicias sesión, algo que podría ser un factor importante, ya que hay usuarios que lo encuentran una tarea innecesaria y quieren acceder a la información más fácilmente, por eso en PrepEVAU se ofrece la opción de acceder como invitado para poder disfrutar de las funcionalidades principales de una manera mucho más sencilla.



**EXÁMENES DE
SELECTIVIDAD.
TODOS.**

Tenemos más de 17.000 exámenes de selectividad de todas las comunidades autónomas, de todos los años, de todas las asignaturas.

23.623 personas ya nos han utilizado

Q Buscar exámenes →

Figura 2.2: Plataforma selectividad.com

CAPÍTULO 3

Metodología

Para el desarrollo de esta aplicación web se ha utilizado la metodología de Desarrollo Centrado en el Usuario (DCU), basada en comprender las necesidades, deseos y limitaciones de los usuarios finales de nuestro producto a lo largo de todo el proceso de creación. La premisa fundamental de esta forma de trabajo es que diseñar con empatía hacia los usuarios conduce a productos más efectivos, útiles y satisfactorios.

3.1 Diseño Centrado en el Usuario (DCU).

El enfoque DCU persigue asegurar la consecución de un producto con la funcionalidad adecuada para usuarios concretos. De acuerdo con lo establecido por [12] el proceso de desarrollo utilizando esta metodología se puede desgranar en cuatro fases (Ver figura 3.1)(<https://www.nosolousabilidad.com/manual/3.htm>):

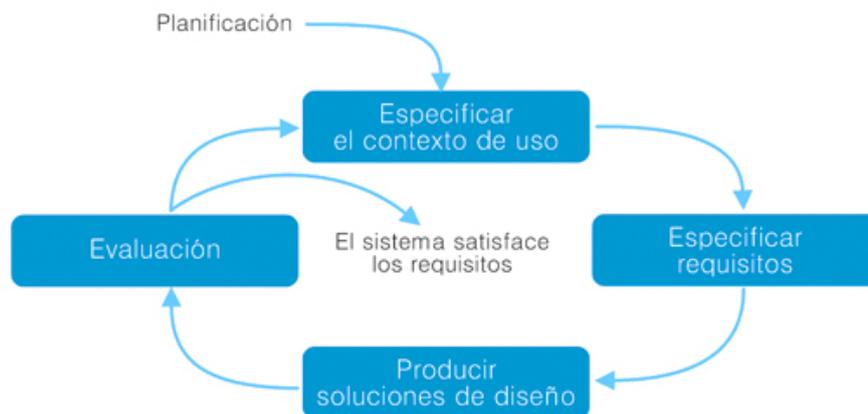


Figura 3.1: Fases del proceso DCU.

- **Entender y especificar el contexto de uso:** El objetivo de la aplicación es ayudar a los usuarios a preparar mejor el examen de acceso a la universidad. Para especificar el contexto de uso existen diferentes técnicas, entre las que destacan el 'Eye Tracking' [3], que consiste en estudiar los componentes de la interfaz que más llaman la atención al usuario. Otra de las técnicas más famosas es el test de usuarios, que consiste en dejar al usuario que interactúe con la aplicación poniéndole un contexto de uso, para ver la facilidad con la que encuentra las cosas y consigue realizar las tareas. En el caso de este proyecto se ha usado un cuestionario, que se ha enviado a una serie de personas que se sabe que realizarán la EVAU este año, por lo que

son usuarios potenciales. De esta forma sacaremos información necesaria para las siguientes etapas del diseño.

- **Especificar requisitos:** Durante esta fase se analiza la información sobre los usuarios recogida mediante las técnicas anteriores y documentarla. En el caso del proyecto, hay que crear una documentación de investigación con usuarios y para ello se ha llevado a cabo la técnica de personas y escenarios. De esta forma con la información del cuestionario se crean perfiles de personas ficticias que juntan las características más comunes de las encuestas. Tras la elaboración de perfiles, se crearán escenarios que reflejen cómo esas personas pueden utilizar la aplicación para cumplir un objetivo.
- **Producir soluciones de diseño:** Esta fase ya engloba el desarrollo del producto, en el que se toman decisiones sobre dónde situar ciertas partes de la interfaz, cómo estructurar la información o cuáles son las posibilidades interactivas del sitio. Algunos de los documentos más populares para registrar este tipo de decisiones son patrones y diagramas de interacción, en los que se representa cómo la aplicación responde a las acciones del usuario. Otra de las formas es mediante mapas de sitios web, en los que se representa una jerarquía de elementos y su relación, representados mediante simples índices o diagramas en forma de árbol. En este caso usaremos wireframes [4], que son esquemas donde aparece cómo deberá ser la distribución, ordenación y relación de los diferentes elementos en cada una de las páginas (o tipos de páginas) del sitio web.
- **Evaluación:** La más importante del proceso, donde se validan las soluciones de diseño y se comprueba que el sistema satisface los requisitos o por el contrario se detectan problemas de usabilidad. Esta validación normalmente se hace a través de tests con usuarios, pero hay otras como pruebas de usabilidad y entrevistas directas con usuarios.

CAPÍTULO 4

Análisis de necesidades

Debido a que se va a seguir la metodología explicada en el capítulo anterior, debemos analizar a los potenciales usuarios de la aplicación. El objetivo de esta parte es investigar los hábitos de las personas que utilizarán la app y a partir de ahí identificar sus necesidades para saber qué funcionalidades debemos desarrollar.

El análisis de necesidades se divide en tres fases, las cuales se complementan entre sí para proporcionar una comprensión completa de las expectativas de los usuarios finales y de esta manera, formar un enfoque integral para informar y guiar el diseño y desarrollo de la solución.

- **Investigación cualitativa:** Se realiza un cuestionario online donde se sacará información sobre qué tecnologías o métodos de estudio usan los alumnos y por qué lo hacen. De esta manera podremos ver los problemas y necesidades que tienen y deberíamos tener en cuenta para desarrollar la solución.
- **Técnica Personas:** Con los resultados obtenidos de la fase anterior se modelarán los potenciales usuarios de la aplicación. Estas serán personas ficticias que se crean a partir de las necesidades reales de los usuarios.
- **Escenarios:** Con la información de la fase de investigación cualitativa se diseñarán varios escenarios en los que, la persona ficticia modelada en la fase anterior, usará las funcionalidades de la aplicación en diferentes contextos de su vida cotidiana.

A continuación veremos una información más detallada y el resultado final de las tres fases.

4.1 Investigación Cualitativa

El cuestionario se ha realizado online mediante la herramienta que proporciona Google Forms, distribuyéndolo en su totalidad a alumnos que este año tienen que realizar la EVAU y están preparándose. Contiene 17 preguntas en las que se intenta recoger información sobre la forma de estudio y los hábitos que tienen los estudiantes, así como datos necesarios para posteriormente modelar un perfil de usuario acorde a los resultados obtenidos. Se han recogido respuestas de 17 personas.

4.1.1. Resultados del Cuestionario

1. Inicialmente con los datos obtenidos sabemos que la mayoría de personas son hombres que están cursando 2° de bachillerato. (ver figuras 4.1 y 4.2). Además en la figura 4.3 vemos que hay un empate entre alumnos de ciencias y de letras.

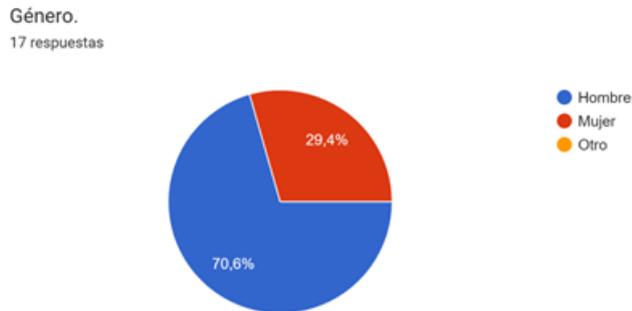


Figura 4.1: Género

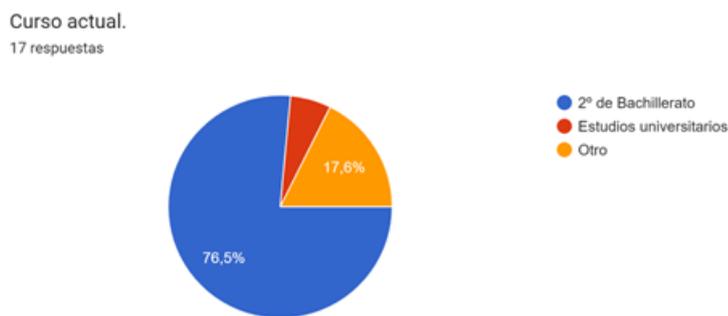


Figura 4.2: Curso

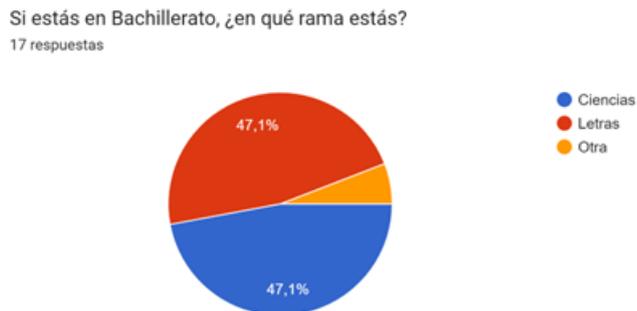


Figura 4.3: Rama

2. Las siguientes preguntas son acerca del uso de las tecnologías. La mayoría de usuarios afirma que se le dan bien las tecnologías en general, teniendo algunos expertos. (Figura 4.4). Además En las figuras ?? y 4.6 observamos que para estudiar el dispositivo más usado para estudiar es el ordenador, aunque en lo que más tiempo se gasta con las tecnologías son con las redes sociales.

¿Cómo te manejas con las tecnologías? (Uso del móvil, ordenador...)
17 respuestas

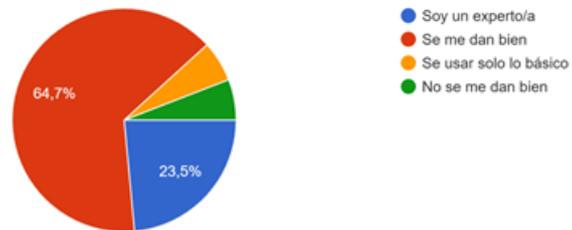


Figura 4.4: Manejo de las tecnologías

¿Si usas algún dispositivo para estudiar, cuál utilizas?
17 respuestas

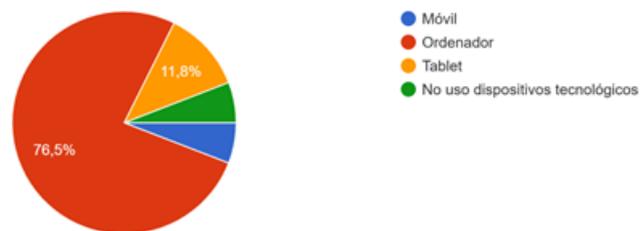


Figura 4.5: Dispositivo para estudiar

¿En qué gastas más tiempo con las tecnologías que utilizas?
17 respuestas

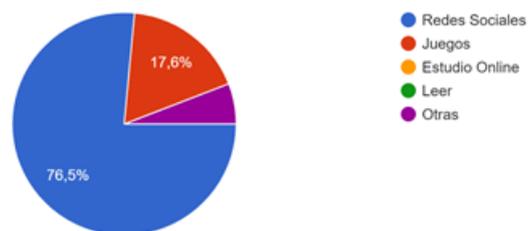


Figura 4.6: Uso de las tecnologías

Mirando la figura 4.7, las redes sociales más utilizadas son Whatsapp, TikTok e Instagram, siendo ésta última la preferida por los usuarios.



Figura 4.7: Redes sociales

Los usuarios tienen opiniones diversas respecto a registrarse en las páginas web o usarlas como invitados. Aún así hay más a favor del acceso sin registro. (Ver figura 4.8).



Figura 4.8: Perfil propio o invitado

3. Respecto a la preparación para la EVAU, los alumnos encuentran la gestión del tiempo como el mayor desafío, seguido por la sobrecarga de información. (Ver figura 4.9).

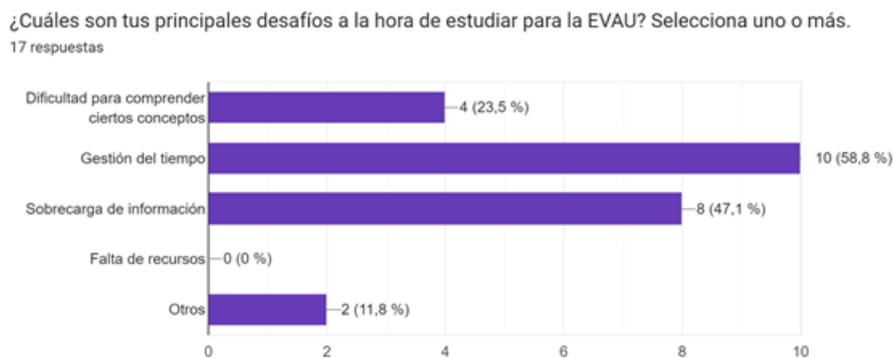


Figura 4.9: Mayores desafíos

En las figuras 4.10 y 4.11 observamos que los recursos más usados para el estudio son libros de texto y que a los usuarios les gustaría disponer de recursos multimedia de ayuda y de ejercicios prácticos con soluciones.



Figura 4.10: Recursos usados.

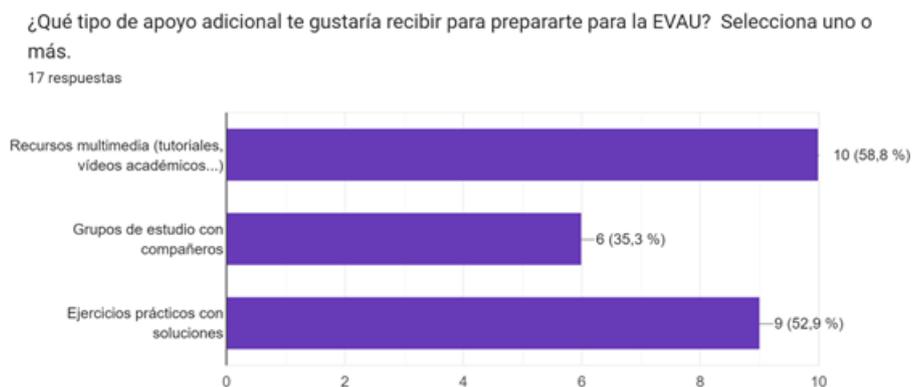


Figura 4.11: Ayuda adicional

4. En cuanto al uso de tecnologías para estudiar, la más popular son los vídeos de Youtube de tutoriales. (Ver figura 4.12). Además, como podemos ver en la figura 4.13 los factores que consideran más importantes en un recurso de apoyo son la claridad de la información y la facilidad de acceso y uso.

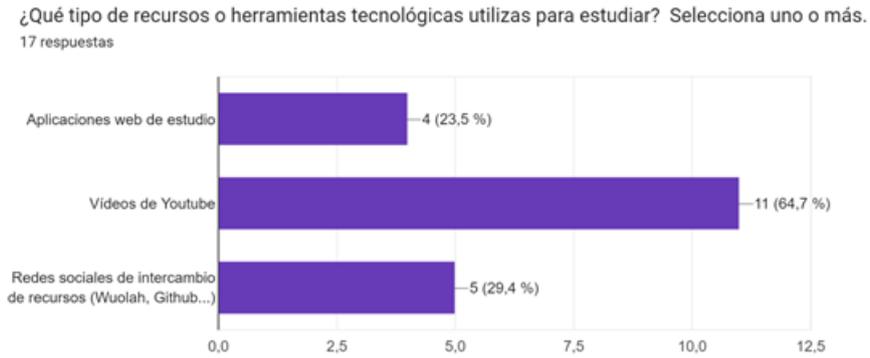


Figura 4.12: Recursos adicionales

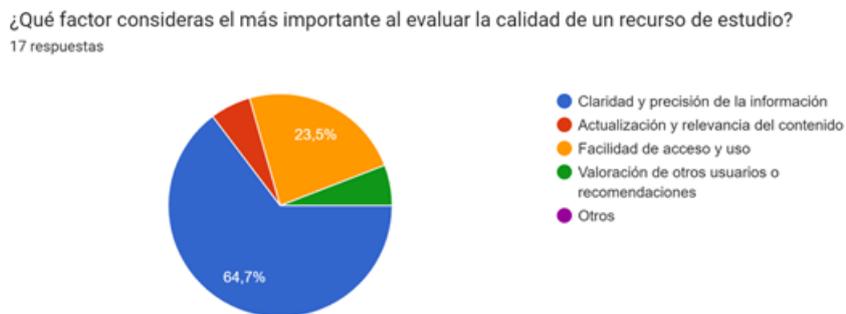


Figura 4.13: Valoración de recursos adicionales

5. Según los resultados de la encuesta, solo hay dos alumnos que utilizan actualmente una aplicación para organizarse su curso. Estas son Calendars y Moodle. (Ver figuras 4.14 y 4.15).

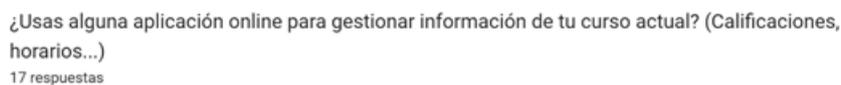


Figura 4.14: Uso de aplicaciones de organización del curso

¿Usas alguna aplicación online para gestionar información de tu curso actual? (Calificaciones, horarios...)
17 respuestas

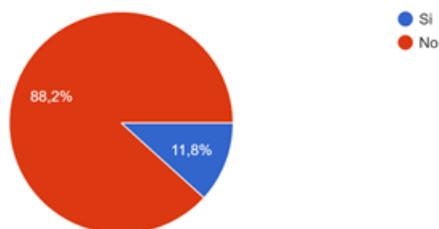


Figura 4.15: Aplicaciones utilizadas

6. Finalmente, en las figuras 4.16 y 4.17 observamos que al realizar ejercicios prácticos, los usuarios mayoritariamente los hacen sin mirar las soluciones, para luego comprobarlos posteriormente. También destacamos que hay variedad de opiniones respecto a la asignatura a la que más tiempo de estudio dedicar, pero ganan las asignaturas del bloque principal (Historia, Literatura y Matemáticas).

Cuando haces ejercicios prácticos, ¿cómo sueles practicar?
17 respuestas

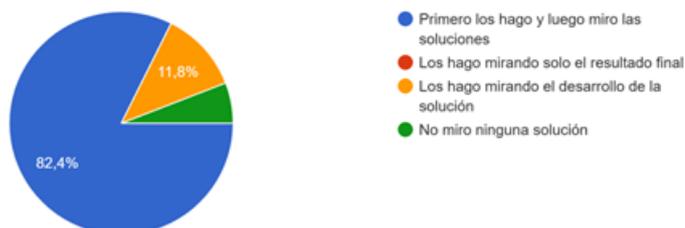


Figura 4.16: Realización de ejercicios prácticos

¿Cuál es la asignatura a la que más tiempo de estudio le vas a dedicar para la EVAU?
17 respuestas

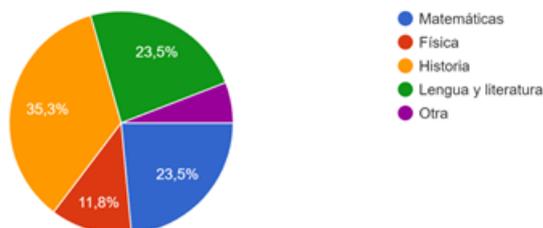


Figura 4.17: Asignatura de más tiempo de estudio

4.2 Técnica de Personas

Una vez hecho el análisis cualitativo por medio del cuestionario, ya se pueden crear perfiles potenciales de personas que vayan a usar la aplicación. En este caso se ha creado a Lucas, un estudiante de segundo de bachillerato que va a realizar la EVAU este año y necesita prepararse.

En la técnica de personas, se pueden realizar los perfiles que se requieran pero para el caso del desarrollo de la aplicación PrepEVAU solo se hará una persona, ya que todos los usuarios potenciales de la app serán de un perfil muy parecido, el de alumnos que se quieren preparar para el examen de acceso a la universidad o cualquier otro de las asignaturas de matemáticas o física de bachillerato.

4.2.1. Lucas Martínez Almeida, 18 años



Biografía

Lucas es un estudiante de segundo de bachillerato de la rama de ciencias tecnológicas, en el centro IES San José de Cuenca. Este año tiene pensado presentarse al examen de acceso a la universidad para estudiar alguna ingeniería, aunque no tiene muy claro cuál. Sus asignaturas favoritas son matemáticas y física. Prefiere estudiar realizando ejercicios prácticos parecidos a los que se encontrará en los exámenes. Además, es un chico deportista que juega en un equipo de fútbol por las tardes.

Perfil Tecnológico

- A Lucas no le gustan los productos de Apple, tiene un móvil de gama media Android y un buen ordenador portátil, que usa para estudiar y jugar a videojuegos.
- Toma apuntes en clase a mano y por las tardes los pasa a formato digital con el ordenador.
- Para prepararse la EVAU está guardándose los exámenes de otros años en formato PDF para poder hacerlos más tarde.
- Suele usar el móvil en su tiempo libre para chatear con amigos y usar las redes sociales, en especial le gusta mucho TikTok.

Necesidades y Objetivos

- Su objetivo es sacar la máxima nota para poder cursar estudiar la ingeniería que quiera.
- Quiere más facilidad a la hora de acceder a ejercicios prácticos resueltos, ya que es su forma de estudio habitual.
- Le gustaría tener toda la información (documentación, ejercicios...) en un mismo sitio, para no tener que buscar cada cosa en una web distinta.

4.3 Escenarios

4.3.1. Escenario 1. Búsqueda de exámenes, ejercicios y documentación.

Lucas vuelve a casa después de un día de clase en el que le han dicho que esa semana harán un simulacro de examen de EVAU de matemáticas. Hoy no tiene entrenamiento porque ya tuvo ayer, así que tiene toda la tarde libre para prepararse. Una vez está con el ordenador no sabe por dónde empezar a buscar exámenes y ejercicios de otros años, entonces recuerda que uno de sus amigos le ha dicho que conoce una aplicación que te muestra exámenes solucionados de la comunidad y año que quieras con solo un click, además de una amplia documentación sobre los temas que están dando en clase. Lucas ve que además no hace falta registrarse para utilizarla así que empieza a hacer ejercicios prácticos y se da cuenta de que la web le permite aprender mucho más rápido de forma sencilla, así que se la enseña a otro amigo de clase para que pueda practicar también.

Tras un día completo de estudio y práctica, Lucas se va a descansar y a jugar a algún juego del ordenador con la tranquilidad de que ha sido muy productivo y se siente bien preparado por si la prueba fuese al día siguiente.

4.3.2. Escenario 2. Uso de la biblioteca.

Lucas tiene un examen de física mañana y esta semana no le ha dado mucho tiempo a estudiar porque ha estado haciendo un trabajo de lengua y literatura. Cuando acaba de comer se sube a su habitación y se pone a practicar con la aplicación ejercicios prácticos. Empieza con los más fáciles y los hace sin problemas, pero cuando cambia de temática de problemas, hay algunos que se le complican porque no tiene los conceptos muy claros. Lucas ve que la aplicación te deja guardarte ejercicios en favoritos para poder revisar más tarde, pero para eso necesita registrarse y tener un perfil. Como sabe que esa tarde tiene entrenamiento de fútbol, decide crearse un usuario en la aplicación para guardarse los problemas que no le salen bien. Esto lo hace para poder estudiarse bien la teoría y después volver a probar el ejercicio y entender por qué se hace de esa manera. Lucas, después de mirarse bien la teoría, se prepara para ir a su entrenamiento. Al volver a casa, entra a la aplicación con el usuario que se creó anteriormente para ver los ejercicios que guardó en favoritos y poder rehacerlos.

Al final del día Lucas consigue entender bien los problemas que le entrarán en el examen de mañana, así que se va a dormir tranquilo y preparado para sacar una buena nota.

CAPÍTULO 5

Análisis conceptual y Diseño

En este capítulo se explicará con detalle cómo se ha realizado y estructurado la base de datos del servidor, donde se almacena la información necesaria para el correcto funcionamiento de la aplicación y para hacer posible el correcto desarrollo de las funcionalidades propuestas. Entre las distintas opciones que se podían usar para crear la base de datos, la más adecuada era usar la herramienta Xampp ya que mediante sql y phpMyAdmin permite la creación y organización de tablas y relaciones de una forma muy sencilla [2]. En la primera sección (5.1) se explica detalladamente el diagrama de clases utilizado para el desarrollo de la solución, mientras que en la segunda (5.2) se verá el modelado de la base de datos con Xampp, mostrando gráficamente las tablas, columnas y relaciones existentes.

En la última sección del capítulo (5.3) se detallan los distintos bocetos iniciales de la interfaz de usuario y la comparativa con el resultado final, con su explicación de los cambios realizados.

5.1 Diagrama de clases

Un diagrama de clases es una representación gráfica de las clases que forman parte de tu aplicación y sus relaciones entre ellas. Cada clase corresponderá a un objeto, que en la base de datos se representa como una tabla y sus atributos serán sus columnas. En el caso de esta aplicación se han creado 6 tablas:

- **Usuario:** Puede entrar como invitado y estar formado por un id único, que será su clave primaria, o registrarse. En este caso tendrá el id, un nombre de usuario, una contraseña y un avatar (que es la ruta a una imagen en el servidor).
- **Asignatura:** Solo tendrá dos filas haciendo referencia a las dos asignaturas que trata la aplicación, matemáticas y física (escalable a más asignaturas si se quisiera ampliar la app). Cuentan con un id y un nombre.
- **Examen:** Consta de un código de examen, un año, una convocatoria y una ruta al examen en el servidor. Un examen pertenece a una asignatura, por lo que tendrá una relación con esa tabla.
- **Ejercicio:** Formada por un código de ejercicio, una temática, la ruta al ejercicio en el servidor (formato imagen png) y la ruta a su solución. Un ejercicio también pertenece a una asignatura.

- Biblioteca: Tabla a la que solo tendrá acceso un usuario registrado, donde guardará sus ejercicios favoritos. Está formada por un código de biblioteca y por relaciones con la tabla usuario y ejercicio, almacenando sus códigos correspondientes.
- Documentación: Tabla que contendrá un código de documentación, una temática y una ruta a un documento que está alojado en el servidor. La documentación también está relacionada con la tabla asignatura.

A continuación en la figura 5.1 se detalla el esquema de clases gráficamente:

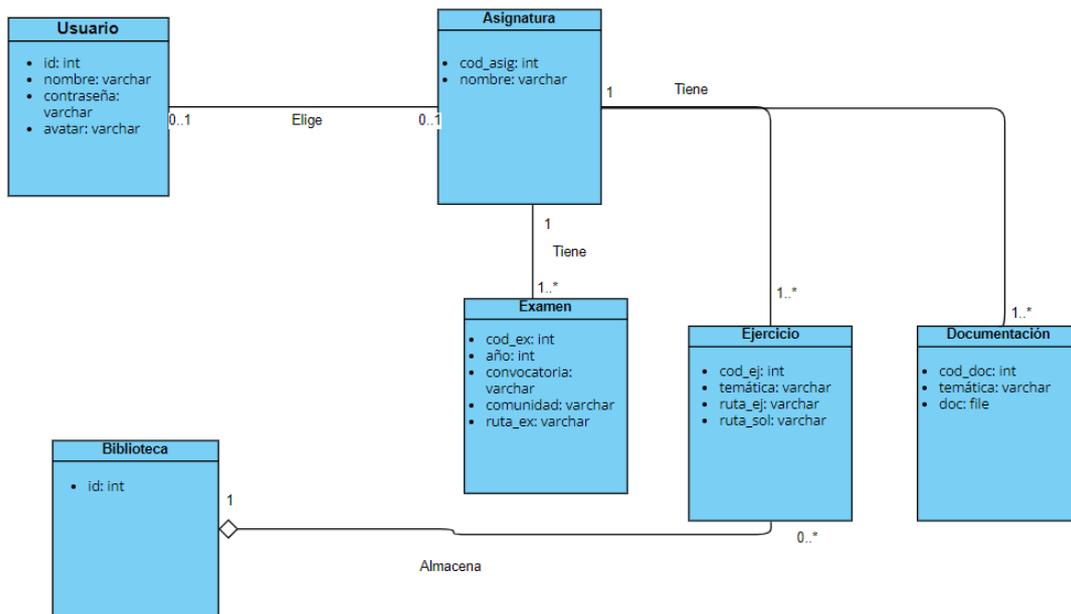


Figura 5.1: Diagrama de clases.

En el gráfico se aprecian todas las relaciones que tienen las tablas y sus atributos junto al tipo de dato que son. En el siguiente apartado se explican con más detenimiento las claves primarias y ajenas de cada objeto.

5.2 Modelo de la BD que use el servidor

El modelo de la base de datos muestra cómo estos están organizados y almacenados en el servidor. A continuación se detalla el diagrama entidad-relación que la herramienta de trabajo Xampp nos ofrece:

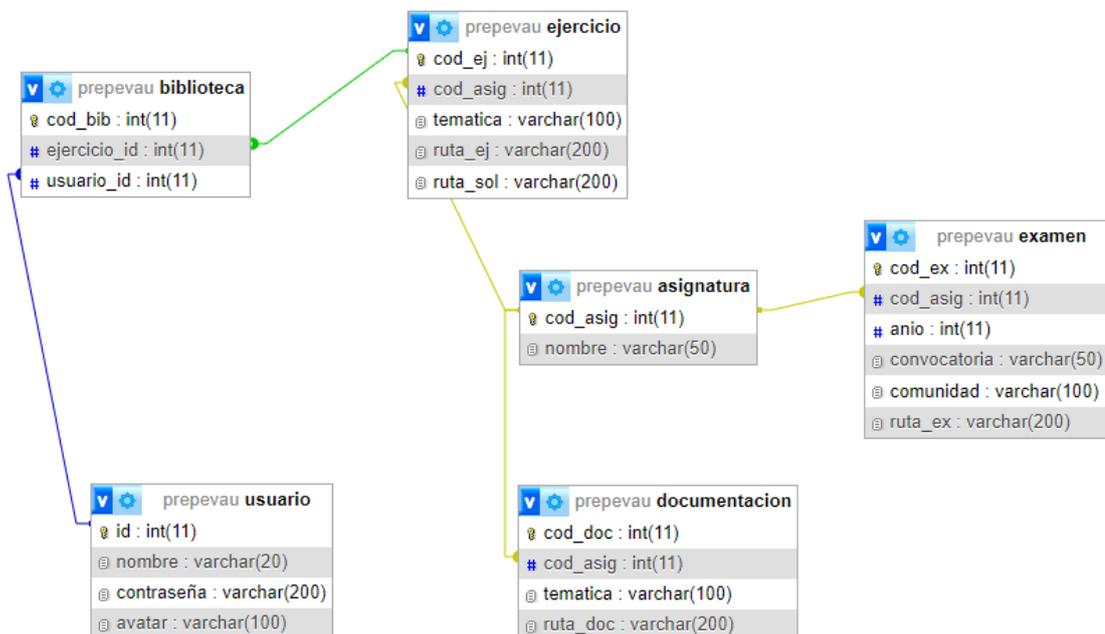


Figura 5.2: Diseño de la base de datos.

En la figura 5.2 se aprecian todos los tipos de datos y sus claves primarias y ajenas que presentan las tablas. Los atributos que se representan con el símbolo de una llave son las claves primarias y los que tienen el símbolo del 'Hastag' son las claves ajenas.

Para crear las claves ajenas o foráneas previamente se tienen que crear unos índices, los cuales nos permitirán relacionar los atributos de las tablas. Xampp nos ofrece la posibilidad de crearlos muy fácilmente utilizando su interfaz (Ver figura 5.3), aunque también se puede hacer mediante código. Una de las ventajas más importantes de usar esta herramienta es que cada cambio o transacción que hagas en tu base de datos te muestra su equivalente en código.

Nombre del índice:

Opción de índice: INDEX

Opciones avanzadas

Columna	Tamaño
<input type="text" value="cod_asig [int(11)]"/>	<input type="text"/>

Figura 5.3: Creación de un índice en Xampp.

Y el código equivalente a la figura 5.3:

```
CREATE INDEX idx_nombre_usuario
ON Usuarios (nombre);
```

Además la interfaz nos ofrece la posibilidad de crear, modificar o eliminar tanto tablas como filas o columnas. La vista de las tablas una vez creada toda la base de datos se puede ver en la figura 5.4.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> asignatura	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> biblioteca	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> documentacion	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	42	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> ejercicio	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	86	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> examen	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	78	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> usuario	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	8	InnoDB	utf8mb4_general_ci	32.0 KB	-
6 tablas	Número de filas	223	InnoDB	utf8mb4_general_ci	208.0 KB	0 B

Figura 5.4: Interfaz de Xampp para edición de tablas.

5.3 Bocetos de las interfaces de la app

Una de las partes más importantes a la hora de desarrollar una aplicación web es la fase de diseño y la elaboración de bocetos, los cuales servirán como guía para poder desarrollar correctamente el producto. En este caso para crear las interfaces se ha utilizado la herramienta de Figma, que proporciona una gran cantidad de combinaciones de elementos para diseñar en distintos entornos. Una de las grandes ventajas que nos proporciona Figma es la posibilidad de crear formas, las cuales puedes exportar como imagen y posteriormente utilizar en tu desarrollo de la interfaz. En esta sección veremos como esto se ha aplicado en varias páginas de PrepEVAU con las formas de elipses y la forma de 'olas' de la página inicial.

Es muy común que durante el diseño de bocetos para la web se creen prototipos de interfaces que luego difieran del resultado final. En esta aplicación en la mayoría de las páginas se ha seguido prácticamente igual el diseño, pero ha habido algunos casos en los que por limitaciones o por cambios en la arquitectura del proyecto se han tenido que cambiar componentes, sin embargo, los cambios no han sido extremadamente distintos al diseño original.

A continuación se mostrarán algunos ejemplos del diseño en Figma y en los casos en los que haya habido cambios con respecto al original se explicarán con más detalle.



Figura 5.5: Interfaz inicial de la aplicación.

La primera interfaz que se muestra en la figura 5.5 es la página inicial de la aplicación, en la que el usuario elegirá si quiere registrarse, iniciar sesión o entrar como invitado. Aquí no ha habido cambios respecto al resultado final.



Figura 5.6: Diseño de la interfaz de Registro.



Figura 5.7: Interfaz final de Registro.

En cuanto a las vistas del registro de usuarios en la figura 5.6 se ve el prototipo de diseño, el cual si comparamos con la interfaz final (ver figura 5.7) se pueden apreciar leves cambios. Los motivos que dieron lugar a estos cambios fueron simplemente estéticos, por ejemplo, en la vista final cuando el usuario cambia de avatar, este se visualiza justo a la derecha y puede elegir el que más le agrade.



Figura 5.8: Interfaz del menú principal.

En el menú principal de la aplicación (ver figura 5.8) están las tres funcionalidades principales de PrepEVAU, el modo examen, entrenamiento y ver temario. El resultado final solo tiene una pequeña diferencia, se ha eliminado el texto de explicación de los modos. La razón es estética y que los modos ya son suficientemente intuitivos y fáciles de entender, por lo que una explicación de cada funcionalidad era innecesaria.



Figura 5.9: Interfaz del modo exámen.

En la figura 5.9 se ve la interfaz del modo examen, en la que un usuario filtra por año, convocatoria y comunidad autónoma y pulsando en buscar aparecen los resultados de su búsqueda de exámenes. Al pulsar sobre un examen, este se abrirá en una pastaña nueva del navegador. En este caso el producto final no difiere del diseño original.

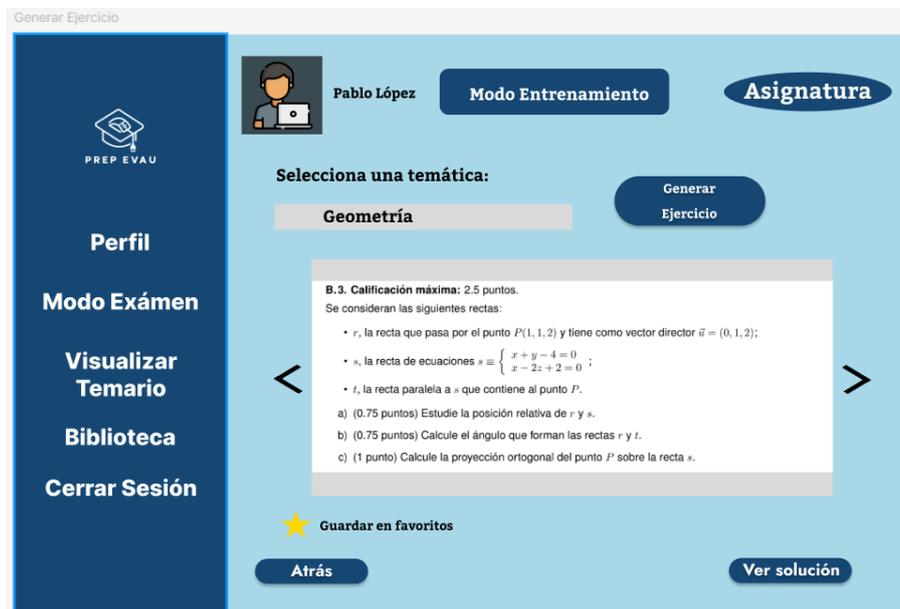


Figura 5.10: Diseño de interfaz de visualización de ejercicios.

La figura 5.10 muestra la vista de diseño del modo entrenamiento, en la que el usuario elige una temática de la asignatura en la que se encuentre y muestra dentro de la aplicación el ejercicio en formato imagen. Esta página no tiene cambios significativos respecto al resultado final tras el desarrollo, salvo alguna modificación de botones.

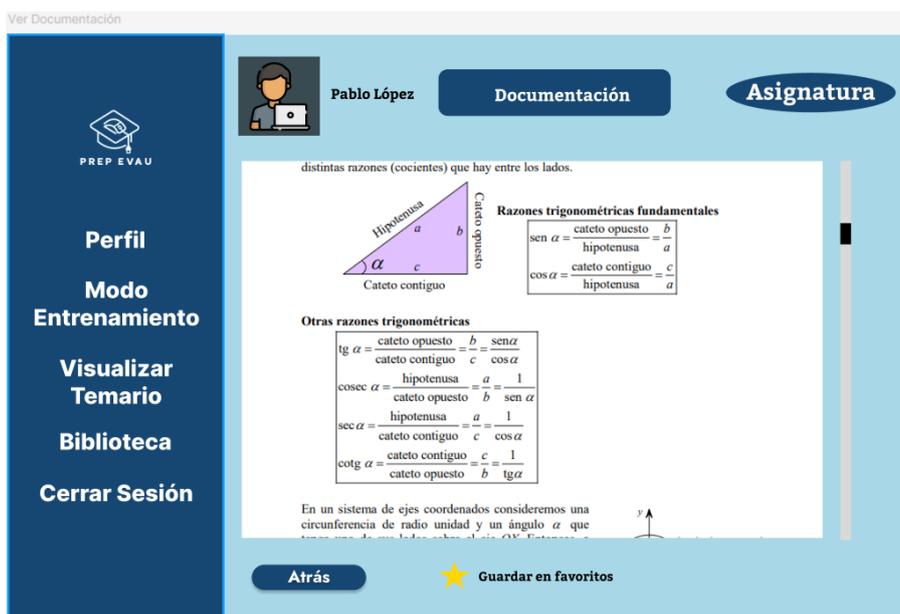


Figura 5.11: Diseño de interfaz de visualización de temario.

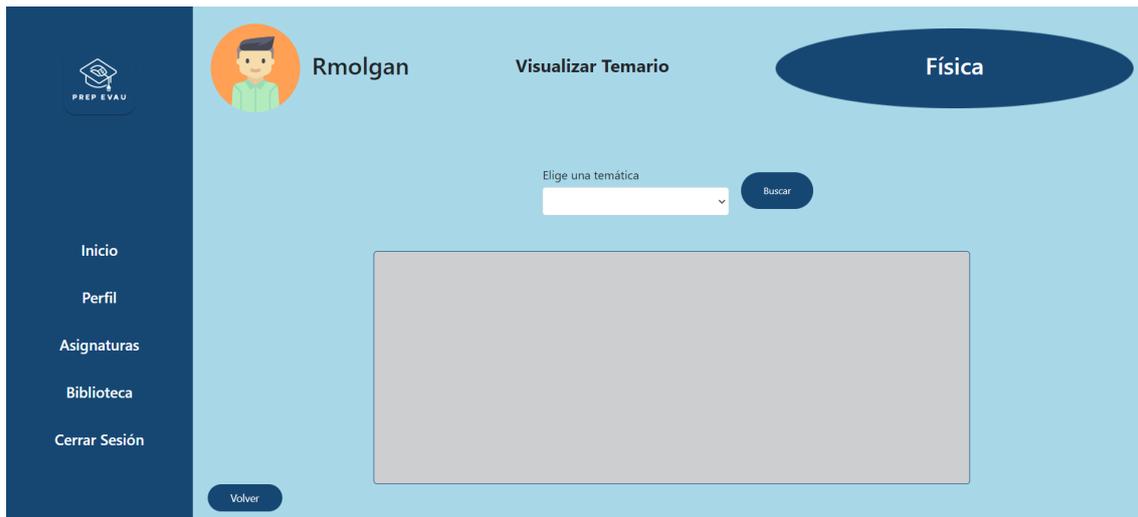


Figura 5.12: Interfaz final de visualización de temario.

En cuanto a la interfaz de visualizar temario si que ha habido cambios significativos durante el desarrollo. En la figura 5.11 hace referencia al primer diseño, en el que el documento se muestra dentro de la misma aplicación, pudiendo hacer 'scroll' para navegar por él. En la figura 5.12 se observa cómo queda la interfaz final, que es muy parecida a la del modo exámen. En esta página el usuario filtra por temática y le aparecen una lista de documentos del temario que elija pero al pinchar sobre estos, se abrirán en una nueva pestaña del navegador, al igual que pasa en la vista del modo exámen. Esto se hizo porque simplificaba el diseño y hacía que fuese mucho más cómodo ver los documentos.



Figura 5.13: Diseño de interfaz de biblioteca I.



Figura 5.14: Diseño de interfaz de biblioteca II.

La funcionalidad de la biblioteca, a la que solo tiene acceso el usuario registrado, inicialmente se diseñó dividida en dos partes, dependiendo de qué tipo de documento se quiera mostrar (ver figuras 5.13 y 5.14). Durante el desarrollo hubo un cambio importante en cuanto a las funcionalidades que estaban pensadas en un principio, y es que se eliminó que el usuario pudiese guardar en su biblioteca exámenes y documentos, sólo podría almacenar ejercicios y soluciones. Esto se debe a que la manera de mostrar en la aplicación tanto los exámenes como los documentos del temario es en una pestaña nueva del navegador, lo que ya le permite al usuario descargar y guardar el pdf en su ordenador, sin embargo, los ejercicios se muestran en formato imagen y no se da la opción de descargar, así que en la biblioteca sólo se podrán guardar ejercicios que el usuario considere interesantes para poder revisarlos más tarde.

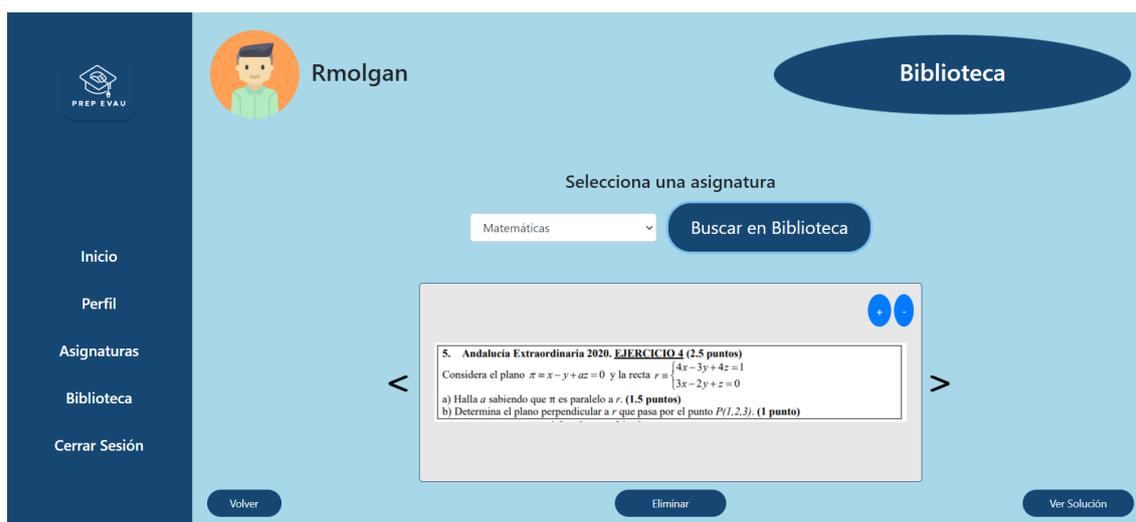


Figura 5.15: Interfaz final de biblioteca.

Como se puede ver en la figura 5.15 la interfaz final de la biblioteca es muy parecida a la del modo ejercicio, mostrando los ejercicios que el usuario tenga guardados eligiendo la asignatura deseada.



Figura 5.16: Diseño de la interfaz de modificación del perfil.

Por último, en la figura 5.16 se muestra el diseño de la interfaz de modificación de perfil. Esta es prácticamente igual que el diseño final, salvo por el mismo cambio que en la de registro, en la parte de elección del avatar, que será igual que en la figura 5.7. También se elimina la parte superior derecha que muestra la asignatura actual, ya que en esta página no tiene importancia.

CAPÍTULO 6

Desarrollo de la solución

Una vez hecho el análisis conceptual, el diseño de las interfaces y creada la base de datos, se puede pasar a la siguiente fase, la de desarrollo. Todo el proyecto ha sido desarrollado en Visual Studio Code, un editor de código que ofrece la posibilidad de trabajar en multitud de lenguajes de programación y la instalación de herramientas y extensiones que facilitan el desarrollo de la solución. Para explicar con detalle esta parte del proyecto se va a dividir el capítulo en varios apartados en los que se mostrará la arquitectura que sigue la aplicación (6.1), se explicarán las herramientas y lenguajes utilizados en el proceso (6.2) y finalizará con una breve demostración de partes importantes del código del proyecto (6.3).

6.1 Arquitectura

La aplicación se basa en un modelo Cliente-Servidor basado en 3 capas [1]. Se llama así porque está formado por tres componentes que se comunican entre si, el cliente, el servidor y la base de datos (ver figura 6.1). Estos componentes forman las 3 capas de la aplicación:

- **Capa de Presentación:** Se trata del cliente que realiza peticiones para recibir datos o para modificarlos. Esta capa está desarrollada en el frontend de la aplicación, el cuál se explicará con detalle en el siguiente apartado.
- **Capa de Aplicación:** Es el servidor, que actúa como intermediario entre el cliente y la base de datos. Se encarga de procesar las peticiones y devolver los datos requeridos por el cliente. Esta capa está desarrollada en el backend de la aplicación.
- **Capa de Datos:** Es la base de datos. Se encarga de gestionarlos y modificarlos según las consultas que le haga el servidor.

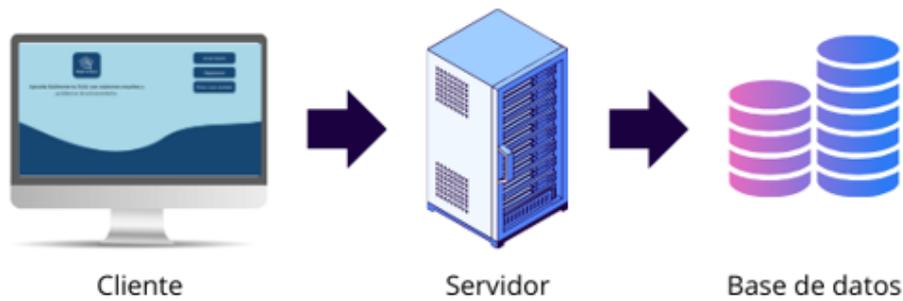


Figura 6.1: Modelo basado en 3 capas.

A continuación se explicará cómo se ha desarrollado cada parte de la aplicación, dividiéndola en dos partes: El frontend (Cliente) y el backend (servidor y base de datos).

6.1.1. Frontend.

El frontend se encarga de la interacción con el usuario y está desarrollado utilizando HTML, CSS, Bootstrap y JavaScript [9]. Todos los estilos utilizados en el proyecto se alojan en el archivo `styles.css`. Bootstrap se ha utilizado para asegurar un diseño 'responsive' y estéticamente agradable, adaptando la aplicación a diferentes cambios en el tamaño de la pantalla. La funcionalidad dinámica, como la actualización de contenido sin recargar la página, se maneja con JavaScript.

Como el editor de código utilizado para el desarrollo de la aplicación es Visual Studio Code, existe una extensión que se ha utilizado para crear las páginas HTML y comprobar los estilos CSS cada vez que se detecta un cambio. Esta extensión es LiveServer y es de gran ayuda a la hora de desarrollar el frontend.

La mayoría de peticiones al servidor se hacen con formularios que el usuario rellena en el frontend (ejemplo en la figura 6.2), ya sea para obtener un recurso de la base de datos o para añadir o modificar, pero también se hacen peticiones nada más cargar la página, por ejemplo, para añadir dinámicamente los datos del usuario que está conectado mediante el objeto `session` que ofrece Express. En la siguiente sección (6.2) se detallarán más los lenguajes y tecnologías utilizados durante el desarrollo.

Miguel_05 Modo Exámen Matemáticas

Año Convocatoria Comunidad Autónoma Buscar

Figura 6.2: Ejemplo del formulario del modo exámen.

6.1.2. Backend.

El Backend de la aplicación se encarga de procesar las peticiones del cliente, interactuar con la base de datos y devolver las respuestas adecuadas. Las herramientas utilizadas para el desarrollo han sido Node.js como entorno de ejecución de JavaScript y Express.js como framework que proporcione las herramientas necesarias para el desarrollo (figura 6.3). El archivo principal del servidor es `app.js`, donde se encuentran las rutas a todas las páginas y donde se importan los archivos donde se procesan las peticiones que se envían desde el frontend. A continuación se detallarán las partes más importantes del backend:

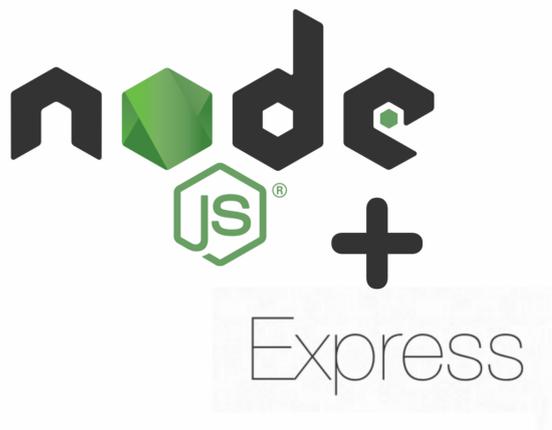


Figura 6.3: NodeJS + Express.

Conexión con la base de datos

La base de datos también está desarrollada en un entorno local (`localhost`) y desde el servidor hay que configurar su conexión para poder realizar las consultas necesarias. De esto se encarga el archivo `db.js`, donde se especifica el nombre de la base de datos 'prepevau' y se crea la conexión guardándola en una variable 'db', que se exportará y será utilizada en otros archivos manejadores de peticiones para realizar consultas SQL.

Middlewares

Para el desarrollo también se han utilizado algunos middlewares para que actúen como intermediarios entre las peticiones y el servidor y realicen alguna tarea específica antes de que se procese la petición. Estos se han definido en el archivo principal de la aplicación `app.js`.

Se han utilizado tres middlewares para el procesado de peticiones y respuestas en formato JSON. Además, Express.js nos ofrece un objeto 'session' [6] (librería `express-session` instalable), el cuál se ha usado en la aplicación para cargar los datos del usuario registrado y de la asignatura al cargar las páginas HTML. El objeto `session` consta de seis atributos:

- `userId`: Identificador del usuario registrado.
- `username`: Nombre de usuario. En caso de entrar como invitado este valor es null.
- `avatar`: Ruta del avatar del usuario registrado.

- `asignaturaId`: Identificador de la asignatura que se haya elegido.
- `asignatura`: Nombre de la asignatura seleccionada.
- `id`: Identificador de la sesión. (Por defecto establecido por Express.js)

Los tres primeros atributos se establecen al iniciar sesión o registrarse en la aplicación, de esta forma aparecen en la página de selección de asignatura. Los datos de la asignatura se actualizan al elegir una, de forma que en la pantalla del menú, si eres un usuario registrado, te deben salir los datos de la sesión cargados automáticamente al entrar. Para todo el control de la sesión se utiliza el archivo `session.js`, que maneja las peticiones del frontend para obtener los datos.

Herramientas y Librerías

Durante el proceso de desarrollo se han instalado algunas librerías que ayudaban en ciertas funciones del proyecto. Una de ellas ha sido **Nodemon** [8] (Figura 6.4), una herramienta que reinicia automáticamente la aplicación cada vez que detecta cambios en los archivos del proyecto. La instalación se realiza mediante la terminal de Visual Studio Code y tras finalizar, aparece en el archivo `package.json`. Con esta herramienta al ejecutar el script creado `'npm run start'` (al haber creado el script `start`, se añade la parte de `nodemon app.js`) se reiniciará al detectar cambios en el código, lo que hace que el desarrollo sea mucho más rápido.



Figura 6.4: Logotipo de Nodemon.

Otra de las herramientas utilizadas ha sido **bcrypt** [7], una librería que se utiliza para la encriptación de contraseñas, aumentando considerablemente la seguridad de los usuarios de la web. Esta herramienta nos permite generar hashes de contraseñas para almacenar en la base de datos, para evitar que un atacante con acceso pueda descifrarlas fácilmente. Bcrypt utiliza un tipo de cifrado simétrico llamado Blowfish [10], que usa bloques de 64 bits y claves que van desde los 32 hasta los 448 bits.

Para manejar el encriptado de las contraseñas se han creado dos funciones en el archivo `auth.js`. Una es `hashPassword`, que recibe una contraseña y devuelve un hash de esta, y la otra es `checkPassword`, que recibe la contraseña hasheada y la original y compara si son la misma. Estas son funciones asíncronas, por lo que para su correcto funcionamiento se deben utilizar las expresiones de `'async'` y `'await'` (esta última es usada en el manejo de peticiones). `hashPassword` se utiliza en la petición de `/Registro`, mientras que `checkPassword` se usa en `/IniciarSesion`. (Para ver fragmentos de código de manejo de las peticiones ir a la sección 6.3).

API Rest y manejo de peticiones

En la carpeta `'routes'` se encuentran los archivos `auth`, `session`, `ejercicio`, `examen`, `biblioteca` y `documentacion`. Estos archivos forman la API Rest y manejan todas las peti-

ciones que se hacen desde el frontend, comunicándose con la base de datos mediante consultas SQL. Las rutas se han dividido en varios archivos por comodidad a la hora de encontrar los métodos.

- Archivo `auth.js`: Es el encargado de procesar los métodos de autenticación y el encriptado de contraseñas. En él están definidos los métodos de 'login' (`/IniciarSession`), de registro (`/Registro`) y de modificar perfil (`/modificarPerfil`).
- Archivo `session.js`: Como se ha explicado anteriormente, aquí se manejan todas las peticiones relacionadas con la sesión. Está formado por los métodos `/user-data`, utilizado al cargar los datos de la sesión en las páginas de la aplicación, `/seleccionasignatura` para añadir la asignatura a la sesión y `/asignaturaInvitado`, que añade la asignatura elegida a la sesión de un usuario invitado.
- Archivo `examen.js`: Maneja la petición que se realiza en el frontend en el modo examen (`/examen`), en la que según los filtros que introduzca el usuario consulta la base de datos y devuelve una lista de exámenes que los cumplen.
- Archivo `ejercicio.js`: Aquí se encuentran todos los métodos del modo entrenamiento, que incluyen el `/generarEjercicio` para generar un ejercicio aleatorio dependiendo de la temática elegida por el usuario, y `/verSolucion` para mostrar la solución de ese ejercicio.
- Archivo `biblioteca.js`: Aquí se manejan la inserción de un ejercicio en la biblioteca de un usuario mediante `/addEjercicioABiblioteca`, el borrado de uno de ellos con `/deleteEjercicioBiblioteca` y el método para mostrar los ejercicios guardados de un usuario dependiendo de la temática elegida.
- Archivo `documentacion.js`: En este archivo se define el método de ver temario `/getDocumentacion`, que devuelve una lista de documentos relacionados con la temática elegida por el usuario.

Rutas de las páginas HTML

En el archivo principal `app.js` también se establecen las rutas que permiten a la aplicación ir navegando por todas sus páginas. Para ello se hace uso del módulo nativo de Node.js 'Path' para hacer que el código sea más robusto y portátil y para evitar escribir todas las rutas manualmente. Esto ha permitido que el acceso a todos los archivos estáticos de la carpeta `public` (archivo `css`, imágenes, exámenes, etc) sean accesibles más fácilmente.

Inicialización del servidor

El código más importante, ya que sin él no funcionaría nada de lo que se ha visto anteriormente es la inicialización del servidor. Se hace en el archivo `app.js` y se encarga de establecer el puerto donde se escucharán las peticiones que lleguen del frontend, que al ser en entorno local será `'localhost:3000'`.

6.2 Contexto tecnológico

En esta sección se especificarán los lenguajes y herramientas utilizadas en todo el proceso de creación del proyecto, explicando por qué se han elegido esas opciones frente a otras que se podían haber usado en su lugar.

6.2.1. Lenguajes

Para el desarrollo del proyecto se han tenido que utilizar diferentes lenguajes de programación dependiendo de la parte a realizar. Estos son los siguientes:

- HTML
- CSS
- Bootstrap
- JavaScript
- SQL

Para el frontend se ha utilizado HTML como lenguaje base para estructurar el contenido de la aplicación web. Se ha elegido esta opción porque es muy fácil de escribir y cualquier navegador moderno admite HTML.

En el diseño del frontend se han utilizado CSS y BOOTSTRAP. Uno se encarga del estilo visual de la aplicación, configurando colores, tipografías o márgenes y el otro se encarga de hacer que las interfaces sean adaptables a distintos tamaños de pantalla, además de ofrecer componentes listos para usar, como botones, menús... Ambos se han elegido porque se complementan muy bien en el diseño y son fáciles de usar.

El lenguaje más utilizado en el proyecto es JavaScript, ya que se usa tanto en el frontend como en el backend. En el lado del cliente sirve para modificar dinámicamente componentes de la página y para hacer peticiones al servidor. En el backend se utiliza para manejar las peticiones, además de para configurar todos los elementos del proyecto como configurar la conexión con la base de datos, configurar el servidor o servir las rutas para las páginas HTML.

Para crear la base de datos se ha utilizado lenguaje SQL, ya que es un lenguaje basado en un modelo relacional, lo que era esencial para el correcto desarrollo de la aplicación. Además, SQL ofrece un lenguaje de consultas a la base de datos muy potente.

6.2.2. Programas y tecnologías

En este apartado se explican los programas utilizados para todas las partes del desarrollo, desde el inicio del proyecto hasta la finalización del mismo.

- Visual Studio Code: Es el editor de código en el que está hecha toda la aplicación, contiene la parte del frontend (archivos HTML y CSS) y la parte del backend (archivos JS del servidor).
- Figma: Es una página web que te permite hacer diseños de interfaces muy completos y de forma gratuita. Con esta herramienta están hechos los bocetos de las interfaces de la aplicación, que se pueden ver en el apartado 5.3.
- Visual Paradigm (Visual-paradigm.com): Es una herramienta web de modelado visual para crear diagramas UML, el cual sirve para planificar la estructura del software y la base de datos.
- Smashing Logo (smashinglogo.com): Se trata de una plataforma web para la creación de logotipos con inteligencia artificial. Con ella se ha creado el logo de PrepE-VAU, el cuál se puede ver en todas las páginas de la aplicación.

- Node.js: Es un entorno de ejecución de JavaScript utilizado para el desarrollo del servidor. Ha sido utilizado para hacer el backend del sistema, manejando las peticiones de los usuarios y comunicándose con la base de datos.
- Express.js: Es un framework de backend para Node.js utilizado para la creación de aplicaciones web y APIs. Ha ayudado a gestionar las rutas y las peticiones HTTP en el servidor, además de permitir utilizar middlewares como los de sesión o autenticación de usuarios.
- Xampp: Es un software gratuito que sirve para crear servidores virtuales locales en tu propia máquina [11]. Se ha utilizado para crear la base de datos MySQL y gestionarla a través de la interfaz que ofrece phpMyAdmin.

6.3 Ejemplos de código

En esta sección se mostrarán algunos de los fragmentos de código más representativos de cada parte de la aplicación, explicando la función que cumplen en cada caso.

En el primer fragmento (figura 6.5) se puede ver la API de inicio de sesión. El cliente envía una petición con los datos de usuario y contraseña y el método busca su nombre de usuario en la base de datos. Si este existe, busca su contraseña (hasheada) y la compara con la introducida mediante el método `checkPassword()` (explicado en el apartado 6.1.2) y en caso de coincidir actualiza los datos del usuario en la sesión.

```
router.post('/IniciarSesion', async (req, res) => {
  const data = req.body;
  const username = data.username;
  const password = data.password;

  // Consulta para verificar las credenciales del usuario en la base de datos
  const query = 'SELECT * FROM usuario WHERE nombre = ?';
  db.query(query, [username], async (err, results) => {
    if (err) throw err;

    if (results.length > 0) {
      // El usuario existe, comprobamos la contraseña
      const passHashed = results[0].contraseña;
      const coincide = await checkPassword(password, passHashed);
      if (coincide){
        req.session.username = username;
        req.session.avatar = results[0].avatar;
        req.session.userId = results[0].id;
        console.log(req.session.userId);
        res.json({ success: true });
      }
      else{
        res.json({ success: false, message: 'Contraseña incorrecta' });
      }
    } else {
      res.json({ success: false, message: 'Nombre de usuario incorrecto' });
    }
  });
});
```

Figura 6.5: Fragmento de código del método iniciar sesión.

En los dos siguientes fragmentos (figuras 6.6 y 6.7) se muestra todo el procesado del modo examen desde la parte del cliente y la parte del servidor.

```

document.getElementById('examen-form').addEventListener('submit', function(event) {
  event.preventDefault();

  const anio = document.getElementById('anio').value || null;
  const convocatoria = document.getElementById('convocatoria').value || null;
  const comunidad = document.getElementById('comunidad').value || null;

  const data = {
    anio: anio,
    convocatoria: convocatoria,
    comunidad: comunidad
  };

  fetch('/examen', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(data),
  })
  .then(response => response.json())
  .then(examenes => {
    // Mostrar los exámenes en la interfaz de usuario
    const listaExámenes = document.getElementById('lista');
    listaExámenes.innerHTML = '';

    examenes.forEach(examen => {
      const li = document.createElement('li');
      const enlace = document.createElement('a');
      enlace.href = examen.ruta_ex;
      enlace.textContent = examen.nombreArchivo;
      enlace.target = '_blank';
      li.appendChild(enlace);
      listaExámenes.appendChild(li);
    });
  })
  .catch(error => {
    console.error('Error:', error);
  });
});

```

Figura 6.6: Fragmento de código del frontend del modo examen.

En la figura 6.6 se muestra la parte del cliente, en la cual se recogen los datos del filtro de exámenes y se hace una petición al servidor. En caso de que alguno de los filtros del formulario no se haya seleccionado, aparecerá como null y no se tendrá en cuenta a la hora de elegir exámenes en el servidor.

```

router.post('/examen', (req, res) => {
  const data = req.body;
  console.log(data);
  const anio = data.anio;
  const convocatoria = data.convocatoria;
  const comunidad = data.comunidad;
  const cod_asig = req.session.asignaturaId;

  console.log(anio);
  console.log(cod_asig);
  let query = 'SELECT * FROM examen WHERE cod_asig = ?';
  const params = [cod_asig];

  if (anio) {
    query += ' AND anio = ?';
    params.push(anio);
  }
  if (convocatoria) {
    query += ' AND convocatoria = ?';
    params.push(convocatoria);
  }
  if (comunidad) {
    query += ' AND comunidad = ?';
    params.push(comunidad);
  }

  db.query(query, params, (err, results) => {
    if (err) throw err;

    // Solo devolver el nombre del archivo sin la extensión .pdf
    const examenes = results.map(examen => {
      const nombreArchivo = examen.ruta_ex.split('/').pop().replace('.pdf', '');
      return { ...examen, nombreArchivo };
    });

    res.json(examenes);
  });
});

```

Figura 6.7: Fragmento de código del backend del modo examen.

En la figura 6.7 se muestra el manejo de la petición del cliente, que recibe los datos de filtrado y según el usuario haya elegido va creando una consulta a la base de datos. Una vez se han encontrado los exámenes que satisfacen las condiciones se realiza un método para mostrar el nombre del documento (sacado de su ruta) para que al usuario le sea más fácil verlo en la lista de la interfaz.

A continuación en las siguientes figuras (6.8 y 6.9) se mostrará un ejemplo de como es la estructura de la mayoría de las páginas HTML del proyecto (la página del menú principal) y la forma en la que se cargan los datos del usuario al entrar.

```

<div class="menuAsig-wrapper d-flex">
  <!-- Sidebar -->
  <div class="asignaturas-sidebar bg-custom-blue d-flex flex-column align-items-center">
    <div class="sidebar-heading text-center mt-3">
      
    </div>
    <div class="mt-auto mb-auto">
      <div class="list-group list-group-flush w-100 text-center">
        <a href="/menuAsig" class="list-group-item list-group-item-action bg-custom-blue text-white py-3 h4">Inicio</a>
        <a href="/perfil" class="list-group-item list-group-item-action bg-custom-blue text-white py-3 h4">Perfil</a>
        <a href="/asignatura" class="list-group-item list-group-item-action bg-custom-blue text-white py-3 h4">Asignaturas</a>
        <a href="/biblioteca" class="list-group-item list-group-item-action bg-custom-blue text-white py-3 h4">Biblioteca</a>
        <a href="/" class="list-group-item list-group-item-action bg-custom-blue text-white py-3 h4">Cerrar Sesión</a>
      </div>
    </div>
  </div>
  <!-- Page Content -->
  <div class="menuAsig-content d-flex flex-column p-4 bg-custom-light-blue w-100 position-relative">
    <div class="menuAsig-top d-flex justify-content-between align-items-center w-100 mb-5">
      <div class="user-info d-flex align-items-center">
        <img src="" alt="User Image" class="img-fluid rounded-circle mr-3">
        <h2></h2>
      </div>
      <div class="ellipse-container">
        
        <div class="ellipse-text">
          <h2></h2>
        </div>
      </div>
    </div>
    <div class="menuAsig-center-content w-100">
      <h4>Elige una opción</h4>
      <div class="btn-container mt-4 d-flex flex-column align-items-center">
        <a href="/examen" class="btn btn-custom btn-lg mb-3">Modo Examen</a>
        <a href="/ejercicio" class="btn btn-custom btn-lg mb-3">Modo Entrenamiento</a>
        <a href="/visualizarTemario" class="btn btn-custom btn-lg">Visualizar Temario</a>
      </div>
    </div>
  </div>
</div>

```

Figura 6.8: Fragmento de código HTML del menú principal.

En la figura 6.8 se muestra el código HTML del menú principal, donde el usuario elegirá una de las tres funcionalidades que tiene la aplicación. Se divide en la parte del menú izquierdo (Sidebar) y el contenido de la página donde están los tres botones con los distintos modos (Page Content).

```
document.addEventListener('DOMContentLoaded', function() {
  fetch('/user-data')
    .then(response => response.json())
    .then(data => {
      if (data.username) {
        document.querySelector('.user-info h2').textContent = data.username;
        document.querySelector('.user-info img').src = data.avatar;
        document.querySelector('.ellipse-text h2').textContent = data.asignatura;
      } else {
        document.querySelector('.user-info h2').textContent = 'Invitado';
        document.querySelector('.user-info img').src = data.avatar;
        document.querySelector('.ellipse-text h2').textContent = data.asignatura;
        //Cambia las opciones del menu de la izquierda para invitados
        const divElement = document.querySelector('.list-group');
        const enlaces = divElement.getElementsByTagName('a');
        enlaces[1].textContent = "Iniciar Sesión";
        enlaces[1].href = "/IniciarSesion";
        enlaces[2].textContent = "Asignaturas";
        enlaces[2].href = "/Invitado";
        enlaces[3].textContent = "Registrar";
        enlaces[3].href = "/Registro";
        enlaces[4].textContent = "Salir";
        enlaces[4].href = "/";
      }
    })
    .catch(error => console.error('Error al obtener los datos del usuario:', error));
});
```

Figura 6.9: Fragmento de código de carga de los datos de la sesión.

En la figura 6.9 se muestra cómo se procesan los datos de la sesión. Se manda una petición al servidor, el cual nos devolverá los datos del usuario. Existen dos casos, que el usuario esté registrado, con lo que se pondrán sus datos en la parte superior y que sea un invitado, con lo cual se modificará el menú izquierdo (eliminando las opciones a las que un usuario invitado no puede acceder) y se pondrá en la parte superior el nombre y avatar de invitado.

CAPÍTULO 7

Producto desarrollado

En este capítulo se mostrará la aplicación final mediante capturas de pantalla de las interfaces finales y se realizará una guía de uso con las funcionalidades que aparecen en los escenarios descritos en la sección 4.3.

7.0.1. Escenario 1

Inicio de Sesión en modo invitado

En el primer escenario, el usuario creado mediante la técnica personas utiliza la aplicación en modo invitado para utilizar todas sus funcionalidades principales (el modo examen, el modo entrenamiento y el modo de visualización de temario y documentación). Para ello lo primero será elegir la opción de entrar como invitado en el inicio de la aplicación (ver figura 7.1).



Figura 7.1: Inicio de sesión en modo invitado.

Una vez dentro, aparecerá la página de la figura 7.2, que será la página bienvenida y donde tendrá que elegir la asignatura que quiera, además de las opciones de registrarse o iniciar sesión.



Figura 7.2: Modo invitado.

Cuando se elija la asignatura se redirigirá automáticamente al menú principal (ver figura 7.3). Aquí el usuario deberá elegir una de las tres funcionalidades de la aplicación.



Figura 7.3: Menu principal de la aplicación

Modo Exámen

Si eliges el modo exámen te aparecerá inicialmente la lista vacía y cuando pongas los filtros, si pulsas en buscar aparecerán en la lista los que coincidan con el formulario. (Ver figura 7.4)



Figura 7.4: Modo exámen

Modo Entrenamiento

En el modo entrenamiento (figura 7.5) lo que se muestran son ejercicios aleatorios de la temática elegida por el usuario sacados de exámenes de EVAU, dando además la posibilidad de mostrar la solución al pulsar el botón de solución (ver figura 7.6).



Figura 7.5: Modo entrenamiento.

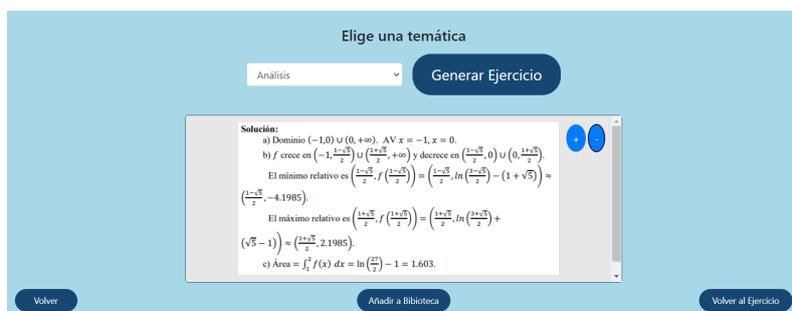


Figura 7.6: Modo entrenamiento (Solución).

Modo Visualizar Temario

Este modo es parecido al modo examen en cuanto a la estructura de la página. El usuario elige una temática y al buscar aparecerán una lista de documentos con diferente información sobre ella. (Ver figura 7.7).



Figura 7.7: Modo visualizar temario

7.0.2. Escenario 2

En el escenario dos, Lucas cambia de asignatura a física y elige registrarse en la aplicación para poder guardarse los ejercicios en su biblioteca y así poder revisarlos más tarde.

Registro de usuario

Lucas se registra en PrepEVAU eligiendo un nombre de usuario, contraseña y avatar (figura 7.8) y entra a la aplicación. Después elige la asignatura de física y va al modo entrenamiento para practicar con unos ejercicios de electromagnetismo. (Ver figura 7.9).

Figura 7.8: Registro de usuario

Esta vez a lucas le ha costado mucho un ejercicio y quiere guardárselo en la biblioteca para poder revisarlo más tarde, así que pulsa el botón de añadir a biblioteca que hay debajo del ejercicio en la figura 7.9.

Figura 7.9: Modo entrenamiento (Usuario Registrado)

Inicio de sesión y Biblioteca

Al cerrar sesión y al iniciarla otra vez más tarde (figura 7.10) se puede acceder al apartado biblioteca, donde estarán los ejercicios que el usuario haya guardado y donde puede ver también la solución (ver figura 7.11).

Figura 7.10: Inicio de sesión de usuario registrado

Figura 7.11: Biblioteca del usuario

Modificación del perfil

En caso de que el usuario quiera cambiar su nombre de usuario, contraseña o avatar, puede hacerlo en el apartado de Perfil que encontrará en el menú izquierdo. (Ver figura 7.12).

Figura 7.12: Modificación del perfil de un usuario

CAPÍTULO 8

Validación

El objetivo de este capítulo es evaluar la usabilidad de la aplicación web desarrollada. No se ha realizado una validación con usuarios reales debido a que en las fechas en las que se desarrolla el proyecto (Agosto/Septiembre) no hay exámenes de EVAU próximos y los nuevos alumnos de bachillerato aún no tienen clases, por lo tanto se ha optado por llevar a cabo una evaluación heurística, utilizando las 10 heurísticas de usabilidad de Jakob Nielsen. Estas heurísticas permiten identificar posibles problemas de usabilidad en la interfaz de usuario, que podrían afectar a la experiencia final del usuario.

8.1 Metodología de Evaluación Heurística

Para realizar la evaluación heurística, se siguieron las 10 heurísticas de Nielsen [5], que son los principios generales de diseño utilizados para identificar problemas comunes en las interfaces de usuario. Cada componente clave de la aplicación (inicio de sesión, navegación, modo examen, biblioteca de ejercicios, etc.) se ha evaluado en función de estas heurísticas. A continuación, se describen brevemente las 10 heurísticas utilizadas:

1. Visibilidad del estado del sistema: El sistema debe mantener al usuario informado de lo que está ocurriendo, proporcionando una retroalimentación adecuada.
2. Coincidencia entre el sistema y el mundo real: La interfaz debe utilizar un lenguaje que sea familiar para los usuarios, con palabras y conceptos que ellos entiendan fácilmente.
3. Control y libertad del usuario: Los usuarios deben tener la posibilidad de deshacer y rehacer acciones de manera sencilla.
4. Coherencia y estándares: Los diseños de las páginas deben ser comunes y seguir un mismo patrón a lo largo de toda la aplicación.
5. Prevención de errores: La interfaz debe minimizar la posibilidad de errores por parte del usuario.
6. Reconocimiento antes que memoria: La interfaz debe mostrar elementos reconocibles que no requieran al usuario recordar información entre diferentes partes del sistema.
7. Flexibilidad y eficiencia de uso: La interfaz debe ofrecer métodos abreviados para usuarios avanzados, sin que interfieran con los usuarios novatos.
8. Estética y diseño minimalista: La interfaz no debe incluir información irrelevante o innecesaria.

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores: Los mensajes de error deben ser claros y proponer soluciones.
10. Ayuda y documentación: El sistema debe proporcionar ayuda accesible y fácil de entender.

8.2 Resultados

Visibilidad del estado del sistema

Evaluación: La aplicación proporciona retroalimentación clara al usuario sobre el estado del sistema. Ejemplos:

- Al iniciar sesión, si las credenciales no son correctas aparece un mensaje de error que te informa del problema.
- Al registrarse, si ya existe un nombre de usuario igual en la base de datos también se comunica al usuario mediante un mensaje de error.
- Al añadir un ejercicio a la biblioteca, salta un mensaje del navegador avisando de si la acción se ha llevado a cabo correctamente. En caso de que un usuario invitado intente guardar un ejercicio en la biblioteca, le aparecerá un mensaje de error (ver figura 8.1).



Figura 8.1: Error añadiendo ejercicio a la biblioteca

Resultado: No se identificaron problemas en este aspecto. La visibilidad del estado es adecuada.

Coincidencia entre el sistema y el mundo real

Evaluación: El lenguaje utilizado en la interfaz es claro y coincide con las expectativas del usuario. Muchos de los términos como 'examen', 'temática' y 'solución' son familiares para los estudiantes de bachillerato y todos los botones dan explicaciones claras de sus funcionalidades.

Resultado: Sin problemas detectados. El sistema utiliza términos comprensibles y coherentes.

Control y libertad del usuario

Evaluación: Se ha implementado la opción de retroceder en casi todas las páginas acciones. El usuario tiene total control sobre la navegación y las funcionalidades.

Resultado: En el menú principal de la aplicación no está el botón de retroceder, que redirigiría a la página de elegir asignatura. En este caso no se ha implementado porque ya existe un botón en el menú izquierdo que te lleva a esa página.

Coherencia y estándares

Evaluación: La interfaz mantiene un diseño consistente en todas las páginas (Para ver las interfaces de usuario finales ir al apartado 5.3. Los elementos comunes, como los botones y menús, siguen un mismo estilo, colocación y color a lo largo de la aplicación.

Resultado: Cumple con los estándares de consistencia, sin errores identificados.

Prevención de errores

Evaluación: Se han implementado mecanismos de validación en los formularios de entrada para prevenir errores de usuario. Por ejemplo, en el inicio de sesión, hay un mensaje de error diferente dependiendo de si lo que falla es el nombre de usuario (no existe en la base de datos) o si la contraseña no es correcta (ver figuras 8.2 y 8.3).



The screenshot shows a login form titled "Iniciar Sesión" on a light blue background. The form has two input fields: "Nombre de Usuario:" with the value "Lucas11" and "Contraseña:" with masked characters ".....". Below the username field, a red error message reads "Nombre de usuario incorrecto". At the bottom, there are two blue buttons: "Atrás" and "Iniciar". A link at the very bottom says "¿No tienes cuenta? Regístrate."

Figura 8.2: Mensaje de usuario incorrecto



The screenshot shows the same login form as Figure 8.2. The "Nombre de Usuario:" field now contains "Lucas_01". The "Contraseña:" field is masked with ".....". A red error message below the password field reads "Contraseña incorrecta". The "Atrás" and "Iniciar" buttons and the "¿No tienes cuenta? Regístrate." link are also present.

Figura 8.3: Mensaje de usuario incorrecto

Resultado: No se identificaron errores. La prevención de errores es adecuada.

Reconocimiento antes que memoria

Evaluación: La aplicación ofrece listas desplegables y menús con opciones predefinidas que facilitan la selección de funcionalidades y navegación por las páginas de manera sencilla.

Resultado: Sin problemas detectados, la aplicación ayuda eficazmente a los usuarios a reconocer las opciones.

Flexibilidad y eficiencia de uso

Evaluación: La aplicación permite tanto a usuarios invitados como registrados utilizarla de manera eficiente. Aunque no se han implementado accesos rápidos o atajos para usuarios expertos (debido a la facilidad de uso), la aplicación es funcional para el público objetivo.

Resultado: No se identifican errores graves, pero se podría considerar añadir más opciones de personalización o atajos en futuras versiones.

Estética y diseño minimalista

Evaluación: El diseño es sencillo y sin elementos innecesarios, cada parte o botón de la aplicación cumple su función. Las páginas están estructuradas de manera clara y accesible.

Resultado: No se detectaron problemas en este principio. La interfaz es limpia y funcional.

Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores

Evaluación: Los mensajes de error son claros y ofrecen soluciones específicas. Por ejemplo, en el registro de usuarios, si no escribes la misma contraseña en los dos campos que hay, se informa al usuario con un mensaje preciso. Otro ejemplo es si el usuario se equivoca de asignatura, tendrá tan fácil volver a elegirla con tan solo dos clicks.

Resultado: No se identificaron problemas. La ayuda para la recuperación de errores es efectiva.

Ayuda y documentación

Evaluación: Aunque la aplicación es bastante intuitiva y se ha pensado para que cualquier estudiante de bachillerato pueda utilizarla sin necesidad de ninguna guía de uso. La ayuda que proporciona la aplicación está presente cuando el usuario comete algún error.

Resultado: No se identifican problemas.

8.3 Conclusiones de la validación

Tras realizar la evaluación heurística de la aplicación, los resultados muestran que la mayor parte de los principios de usabilidad han sido correctamente implementados y se cumplen con las expectativas. Aunque no se encontraron errores graves, se podrían realizar mejoras para optimizar la experiencia de usuarios más avanzados, por ejemplo, mediante la implementación de atajos o más opciones de personalización en futuras versiones, además de aumentar la cantidad de exámenes, ejercicios y documentación en la base de datos.

CAPÍTULO 9

Conclusiones

El desarrollo de este proyecto ha sido un proceso de aprendizaje constante, que me ha permitido entender y aplicar conocimientos teóricos y prácticos en la creación de una aplicación web. A lo largo del proyecto, he aprendido sobre las distintas áreas del desarrollo de software, desde el análisis conceptual y el diseño, hasta la implementación y la validación del producto final. Este proceso me ha permitido consolidar habilidades técnicas, aprender nuevas tecnologías y enfrentar problemas que, a su vez, me han proporcionado importantes aprendizajes.

La elección de la metodología de trabajo basada en el Diseño Centrado en el Usuario (DCU) ha sido clave para el éxito del proyecto. Gracias a la investigación previa y a los cuestionarios realizados a los usuarios potenciales, se definieron las funcionalidades más relevantes de la aplicación.

En términos de arquitectura, la decisión de utilizar una estructura cliente-servidor de tres capas con Node.js y Express.js para el backend, y HTML, CSS, Bootstrap y JavaScript para el frontend, ha demostrado ser la más adecuada para el tipo de proyecto. El uso de XAMPP y MySQL para la gestión de la base de datos facilitó la implementación de las funcionalidades principales de la aplicación.

Para la validación del software se optó por realizar una evaluación heurística. Esta técnica permitió identificar potenciales problemas de usabilidad y asegurarse de que la aplicación cumpliera con los principios clave de Nielsen. Los resultados de la evaluación fueron en su mayoría positivos, lo que refuerza la idea de que la aplicación está bien diseñada y optimizada.

Desde una perspectiva personal, este proyecto ha sido una excelente oportunidad para mejorar mis capacidades en desarrollo web y gestión de proyectos. Me ha permitido aprender a enfrentarme a la resolución de problemas complejos y trabajar con tecnologías que no conocía, pero que son muy utilizadas en el mundo profesional. Además, me ha hecho más consciente de la importancia de la experiencia de usuario y de cómo pequeñas decisiones en el diseño pueden tener un gran impacto en la utilidad y eficiencia de una aplicación.

Durante el grado he cursado muchas asignaturas que me han ayudado a entender los conceptos y desarrollar aplicaciones web. La más relacionada con este proyecto es la asignatura DCU, la cuál me ha enseñado muchas cosas acerca de la metodología de diseño centrado en el usuario, la cuál se ha utilizado para este proyecto. Otra asignatura que me ha ayudado ha sido DEW (Desarrollo Web), con la que he aprendido las bases de cómo se desarrolla una plataforma web. Para la gestión de bases de datos, TBD (Tecnología de Bases de Datos) me ha enseñado a crear y manejar estructuras de datos mediante MySQL, lo cuál ha sido muy importante en este proyecto.

Posibles mejoras

El proyecto podría ampliarse incorporando más funcionalidades y aumentando la escalabilidad mediante la inclusión de más asignaturas, exámenes, ejercicios y documentación. También sería interesante explorar la posibilidad de crear una versión móvil de la aplicación para adaptarse a las necesidades actuales de los estudiantes, quienes cada vez utilizan más sus dispositivos móviles para estudiar y organizarse.

Otra idea de actualización, a parte de las ya mencionadas, sería incluir una plataforma online en la cuál los profesores puedan subir sus propios exámenes y apuntes de las asignaturas para que cualquiera pueda acceder a esos recursos.

Bibliografía

- [1] Juan Cabero. Arquitectura cliente/servidor: modelo de 3 capas. <https://iberasync.es/arquitectura-cliente-servidor-modelo-de-3-capas/>, 2024. Último acceso: Agosto 2024.
- [2] Casamayor Ródenas J. C. Mota Herranz L Celma Giménez, M. *Bases de datos relacionales*. Prentice Hall, 2003.
- [3] Maria Coppola. ¿qué es el eye tracking, cómo funciona y para qué sirve? <https://blog.hubspot.es/website/eye-tracking>, 2024. Último acceso: Abril 2024.
- [4] Miro. ¿qué es un wireframe? <https://miro.com/es/wireframe/que-es-wireframe/>, 2024. Último acceso: Abril 2024.
- [5] Jakob Nielsen. *Usabilidad: diseño de sitios web*. Pearson Educacion, Madrid, 2000.
- [6] NPM. express-session. <https://www.npmjs.com/package/express-session>, 2024. Último acceso: Junio 2024.
- [7] NPM. node.bcrypt.js. <https://www.npmjs.com/package/bcrypt>, 2024. Último acceso: Junio 2024.
- [8] Geek Nómada. Qué es nodemon en node.js. <https://geeknomada.blog/nodejs/que-es-nodemon-en-node-js/>, 2024. Último acceso: Julio 2024.
- [9] Thomas Powell. *HTML CSS: The complete reference*. McGraw-Hill, 2010.
- [10] Wikipedia. Blowfish. <https://es.wikipedia.org/wiki/Blowfish>, 2024. Último acceso: Agosto 2024.
- [11] Wikipedia. Xampp. <https://es.wikipedia.org/wiki/XAMPP>, 2024. Último acceso: Junio 2024.
- [12] Yusef Hassan Montero y Sergio Ortega Santamaría. Diseño centrado en el usuario. *NSU*, 2009. <https://www.nosolousabilidad.com/manual/3.htm>.

APÉNDICE A

Relación con las ODS

A.1 Objetivos de Desarrollo Sostenible (ODS)

A.1.1. Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS)

A continuación, se muestra el grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.			x	
ODS 2. Hambre cero.			x	
ODS 3. Salud y bienestar.		x		
ODS 4. Educación de calidad.	x			
ODS 5. Igualdad de género.		x		
ODS 6. Agua limpia y saneamiento.				x
ODS 7. Energía asequible y no contaminante.			x	
ODS 8. Trabajo decente y crecimiento económico.			x	
ODS 9. Industria, innovación e infraestructuras.			x	
ODS 10. Reducción de las desigualdades.		x		
ODS 11. Ciudades y comunidades sostenibles.			x	
ODS 12. Producción y consumo responsables.			x	
ODS 13. Acción por el clima.	x			
ODS 14. Vida submarina.			x	
ODS 15. Vida de ecosistemas terrestres.	x			
ODS 16. Paz, justicia e instituciones sólidas.			x	
ODS 17. Alianzas para lograr objetivos.			x	

Tabla A.1: Grado de relación del trabajo con los ODS

A.1.2. Reflexión sobre la relación del TFG/TFM con los ODS

El desarrollo de esta aplicación tiene como fin facilitar el acceso a recursos de estudio, como exámenes de años anteriores, documentación y ejercicios de manera totalmente online. Es por ello que está ciertamente relacionada con los Objetivos de Desarrollo Sostenible propuestos por las Naciones Unidas en la Agenda 2030.

En la tabla anterior A.1 se puede ver el grado de relación del trabajo con las ODS y la conclusión es que en los aspectos que más presente está el trabajo realizado son la educación de calidad, la acción por el clima y la vida de ecosistemas terrestres.

- Educación de calidad: Este ODS busca garantizar una educación inclusiva, equitativa y de calidad, promoviendo oportunidades de aprendizaje para todos. Hoy en día la mayoría de alumnos tienen acceso a un ordenador con internet, ya sea propio o prestado por su centro educativo, por lo que todos tienen la oportunidad de utilizar la aplicación de PrepEVAU. De esta forma se facilita un aprendizaje autónomo y organizado para todos los alumnos.
- Acción por el clima: Tiene como objetivo tomar medidas urgentes para combatir el cambio climático y sus efectos. Su relación con este proyecto está en los aspectos de sostenibilidad y eficiencia energética que se han considerado en el desarrollo de la aplicación. Además, PrepEVAU fomenta el uso de la educación digital, lo que ayuda a reducir la necesidad de contar con materiales impresos por los centros educativos. De esta forma se consigue una menor deforestación y un menor uso de papel.
- Vida de ecosistemas terrestres: Este objetivo busca proteger, restaurar y promover el uso sostenible de los ecosistemas terrestres. Esto está relacionado con el ODS anterior, ya que la deforestación de los bosques es un problema para el ecosistema de muchas especies. Al promover el uso de tecnologías digitales se está reduciendo la cantidad de papel necesario y, por lo tanto, se ayuda a conservar el ecosistema de muchas especies que habitan en los bosques.

ETS Enginyeria Informàtica
Camí de Vera, s/n. 46022. València
T +34 963 877 210
F +34 963 877 219
etsinf@upvnet.upv.es - www.inf.upv.es