



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Entrenador de sistemas secuenciales

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Octavia Daniela Buliman

Tutor: Antonio Martí Campoy

2020-2021

Agradecimientos

Me gustaría agradecer de todo corazón a mi tutor Antonio Martí Campoy por mantenerme siempre con una visión positiva durante todo el desarrollo de este TFG, por su paciencia, guía y apoyo.

A todos los profesores que me han ayudado durante mi etapa académica, sobre todo, a mis profesores de segundo Soledad Valero Cubas y Josep Francesc Silva Galiana. Gracias por haberme dado esperanzas y un rumbo a seguir en mi carrera y entre mis profesores de la rama de software, a Patricio Letelier Torre y Jorge Eliseo Marzal Calatayud quiero agradecerles por mantener la llama ardiendo.

A mi madre por estar siempre a mi lado, educarme, quererme y por haber sufrido conmigo cada vez que tocaba hacer algún examen.

A mi querida y mejor amiga María por siempre confiar en mí.

A todos mis compañeros y amigos de universidad con los que hemos pasado juntos por alegrías y dificultades y para todas aquellas personas que han estado allí o han expresado alguna vez su preocupación por mí, especialmente a Jesús.

Resumen

En este trabajo se ha desarrollado una herramienta que contribuye al aprendizaje del alumnado de primer año del Grado de Ingeniería Informática de la Universitat Politècnica de València cursando la asignatura de “Fundamentos de computadores”.

Se ha desarrollado una página web que permite acceder a ejercicios para resolver unas ecuaciones y un cronograma interactivo asociados a un circuito secuencial síncrono con biestables. Se ha optado por desarrollar una aplicación web para evitar problemas de compatibilidad y facilitar el acceso. Esto ha permitido incluir y manejar de forma más sencilla una gestión de usuarios y un registro de los ejercicios realizados por los alumnos.

Esta memoria detalla las fases de análisis y especificación, diseño, implementación, la puesta en marcha y las pruebas que verifican el correcto funcionamiento de la herramienta, además de las tecnologías adoptadas.

Palabras clave: entrenador, ejercicio, biestable, cronograma, usuario, alumno, aplicación web, documento web.

Abstract

In this work, a tool has been developed that contributes to the learning of first-year students of the Computer Engineering Degree of the Universitat Politècnica de València, taking the subject of "Computer Fundamentals".

A web page has been developed that allows access to exercises to solve equations and an interactive time diagram associated with a synchronous sequential circuit with flip-flops. It has been chosen to develop a web application to avoid compatibility problems and facilitate access. This has made it possible to include and manage in a simpler way a user management and a record of the exercises carried out by the students.

This report details the phases of analysis and specification, design, implementation, commissioning and tests that verify the correct operation of the tool, in addition to the technologies adopted.

Keywords : trainer, exercise, flip flop, time diagram, user, student, web application, web document.

Índice de contenidos

Índice de figuras	8
Índice de tablas.....	10
Definiciones, abreviaturas y acrónimos	12
1. Introducción.....	15
1.1. Contexto.....	15
1.2. Estado del arte	15
1.3. Motivación	16
1.4. Objetivos	16
1.5. Plan temporal.....	17
1.6. Presupuesto	18
1.7. Estructura de la memoria	19
2. Conceptos generales.....	21
3. Metodología.....	23
4. Análisis y Especificación de requisitos.....	25
4.1. Glosario de términos.....	25
4.2. Diagrama de contexto	26
4.3. Modelo de dominio	27
4.4. Casos de uso	27
4.4.1. Gestión de usuarios	28
4.4.2. Funcionalidades del administrador	30
4.4.3. Funcionalidades del profesor.....	34
4.4.4. Funcionalidades del alumno	36
4.4.5. Funcionalidades del sistema	46
4.4.6. Requisitos no funcionales	48
5. Diseño.....	49
5.1. Arquitectura de tres capas	49
5.1.1. Capa de presentación	49
5.1.2. Capa de negocio.....	50
5.1.3. Capa de datos.....	50
5.2. Mockups.....	51
5.2.2. Página principal.....	52
5.2.3. Página de ejercicios	55
5.2.4. Página de ayuda.....	56
6. Tecnologías y herramientas	57

6.1.	JavaScript	57
6.2.	PHP	57
6.3.	HTML.....	58
6.4.	CSS.....	58
6.5.	Visual Studio Code.....	59
6.6.	WAMP.....	60
6.7.	Herramientas de diseño.....	60
7.	Implementación	63
7.1.	Diagrama de bloques	63
7.2.	Estructura del proyecto	64
7.2.1.	Backend	64
7.2.2.	CSS.....	65
7.2.3.	Ejercicios	65
7.2.4.	Scripts.....	66
7.2.5.	Imágenes	68
7.2.6.	Plantillas	68
7.3.	Refactorización	72
8.	Pruebas.....	75
8.1.	Usabilidad en la web	75
8.2.	Despliegue.....	77
8.2.1.	Pruebas de funcionalidad y compatibilidad	82
8.2.2.	Pruebas de seguridad	82
9.	Conclusiones	85
9.1.	Valoración personal	85
9.2.	Trabajos futuros.....	85
10.	Bibliografía	87

Índice de figuras

Ilustración 1. Símbolo lógico de biestable D activo por flanco de subida.	21
Ilustración 2. Representación de flancos de subida y bajada en una señal de reloj.	21
Ilustración 3. Circuito secuencial síncrono con biestables tipo D activos por flanco de subida.	22
Ilustración 4. Cronograma de un circuito con 3 salidas y una entrada.....	22
Ilustración 5. Representación del modelo en cascada.	23
Ilustración 6. Diagrama de contexto.	26
Ilustración 7. Diagrama del modelo de dominio.	27
Ilustración 8. Diagrama de casos de uso para la gestión de usuarios.	28
Ilustración 9. Diagrama de casos de uso para las funcionalidades del administrador. .	30
Ilustración 10. Diagrama de casos de uso para las funcionalidades del profesor.	34
Ilustración 11. Diagrama de casos de uso para las funcionalidades del alumno.....	36
Ilustración 12. Diagrama de casos de uso para las funcionalidades del sistema.	47
Ilustración 13. Esquema de la arquitectura en 3 capas.	49
Ilustración 14. Diagrama estructural de la base de datos.	50
Ilustración 15. Prototipo página de acceso.	52
Ilustración 16. Prototipo página principal del administrador – configuración datos de profesores.	53
Ilustración 17. Prototipo página principal del administrador – configuración datos de registro.....	53
Ilustración 18. Prototipo página principal del profesor.	54
Ilustración 19. Prototipo página principal del alumno.	54
Ilustración 20. Prototipo página de ejercicios.....	55
Ilustración 21. Prototipo página de un ejercicio X – Circuito y ecuaciones.....	55
Ilustración 22. Prototipo página de un ejercicio X – Cronograma.	56
Ilustración 23. Prototipo página de ayuda.	56
Ilustración 24. Logotipo de JavaScript.	57
Ilustración 25. Logotipo de PHP.	58
Ilustración 26. Logotipo de HTML.....	58
Ilustración 27. Logotipo de CSS.	59
Ilustración 28. Logotipo de Visual Studio Code.....	59
Ilustración 29. Logotipo de WAMP.....	60
Ilustración 30. Logotipo de Canva.	60
Ilustración 31. Logotipo de GenMyModel.....	61
Ilustración 32. Logotipo de Creately.	61
Ilustración 33. Logotipo de Balsamiq.....	61
Ilustración 34. Diagrama de bloques.	63
Ilustración 35. Estructura global del proyecto.	64
Ilustración 36. Contenido de la carpeta Backend.....	65
Ilustración 37. Contenido de la carpeta css.....	65
Ilustración 38. Contenido de la carpeta ejercicios.	66
Ilustración 39. Contenido de la carpeta scripts.....	66
Ilustración 40. Interfaz gráfica de la página de acceso – formulario login.....	67
Ilustración 41. Interfaz gráfica de la página de acceso – formulario registro.....	67

Ilustración 42. Contenido de la carpeta imágenes.	68
Ilustración 43. Contenido de la carpeta plantillas.	68
Ilustración 44. Archivo PlantillaEjercicio.php – descripción del ejercicio.	69
Ilustración 45. Archivo PlantillaEjercicio.php – inserción de la plantilla en JavaScript.	69
Ilustración 46. Archivo JavaScript – inserción de ecuación del circuito.	70
Ilustración 47. Archivo PlantillaEjercicio.php – descripción de la nomenclatura de las ecuaciones.....	70
Ilustración 48. Archivo JavaScript – función solución del cronograma.....	70
Ilustración 49. Ejemplo implementación de la solución de un circuito con biestables tipo D activo por flanco de bajada.	71
Ilustración 50. Ejemplo implementación de la solución de un circuito con biestables tipo JK activo por flanco de subida.	72
Ilustración 51. Archivo ejerciciosPage.php – enlace a la plantilla php de un ejercicio...72	
Ilustración 52. Mensaje indicando el correcto registro de un usuario.....76	
Ilustración 53. Campos estadísticos de aciertos y fallos.76	
Ilustración 54. Mensaje aleatorio al intentar mostrar solución.76	
Ilustración 55. Accesos directos a las diferentes secciones de la página de ayuda.77	
Ilustración 56. Accesos directos a la página de ejercicios y el ejemplo tutorial.77	
Ilustración 57. Barra de opciones de la aplicación del entrenador.77	
Ilustración 58. Solicitud de alta de un servicio de publicación avanzado. 78	
Ilustración 59. Panel de control de la web de secuenciales.....79	
Ilustración 60. Inserción y extracción de los archivos de la aplicación.79	
Ilustración 61. Creación de una base de datos. 80	
Ilustración 62. Panel de control de la base de datos.81	
Ilustración 63. Importación de un archivo a la base de datos.....81	
Ilustración 64. Acceso a la página de la aplicación del entrenador..... 82	
Ilustración 65. Mensaje de restricción de acceso a la página del administrador..... 83	



Índice de tablas

Tabla 1. Plan de trabajo	17
Tabla 2. Presupuesto estimado del proyecto.....	18
Tabla 3. Tabla de funcionamiento de un biestable D activo por flanco de subida.....	21
Tabla 4. Plantilla de casos de uso	28
Tabla 5. Caso de uso: Registrar alumno	29
Tabla 6. Caso de uso: Inicio sesión.....	29
Tabla 7. Caso de uso: Cerrar sesión.....	29
Tabla 8. Caso de uso: Registrar profesor.....	31
Tabla 9. Caso de uso: Asignar profesor	31
Tabla 10. Caso de uso: Borrar profesor	31
Tabla 11. Caso de uso: Borrar titulación	32
Tabla 12. Caso de uso: Añadir titulación	32
Tabla 13. Caso de uso: Borrar asignatura.....	32
Tabla 14. Caso de uso: Añadir asignatura	33
Tabla 15. Caso de uso: Borrar grupo.....	33
Tabla 16. Caso de uso: Añadir grupo	33
Tabla 17. Caso de uso: Borrar curso	34
Tabla 18. Caso de uso: Añadir curso.....	34
Tabla 19. Caso de uso: Filtrar alumnos	35
Tabla 20. Caso de uso: Acceder página principal.....	37
Tabla 21. Caso de uso: Acceder página de ejercicios	37
Tabla 22. Caso de uso: Acceder página ayuda.....	37
Tabla 23. Caso de uso: Acceder ejercicio básico/ejemplo	38
Tabla 24. Caso de uso: Acceder ejercicio.....	38
Tabla 25. Caso de uso: Acceso D Subida	38
Tabla 26. Caso de uso D Bajada	39
Tabla 27. Caso de uso: Acceso T Subida	39
Tabla 28. Caso de uso: Acceso T Bajada.....	40
Tabla 29. Caso de uso: Acceso JK Subida.....	40
Tabla 30. Caso de uso: Acceso JK Bajada	41
Tabla 31. Caso de uso: Ayuda biestable D	41
Tabla 32. Caso de uso: Ayuda biestable T	41
Tabla 33. Caso de uso: Ayuda biestable JK.....	42
Tabla 34. Caso de uso: Ayuda circuito JK	42
Tabla 35. Caso de uso: Ayuda circuito T.....	42
Tabla 36. Caso de uso: Ayuda circuito D.....	43
Tabla 37. Caso de uso: Dibujar nivel alto	43
Tabla 38. Caso de uso: Dibujar nivel bajo	43
Tabla 39. Caso de uso: Comprobar cronograma	44
Tabla 40. Caso de uso: Mostrar solución cronograma	44
Tabla 41. Caso de uso: Nuevo ejercicio.....	45
Tabla 42. Caso de uso: Comprobar ecuación	45
Tabla 43. Caso de uso: Mostrar solución ecuación	46
Tabla 44. Caso de uso: Despejar campos ecuación	46

Tabla 45. Caso de uso: Introducir ecuaciones	46
Tabla 46. Caso de uso: Enviar estadísticas.....	47
Tabla 47. Caso de uso: Generar cronograma.....	47



Definiciones, abreviaturas y acrónimos

Software: conjunto de instrucciones lógicas y rutinas, que permiten ofrecer servicios a un usuario o realizar unas tareas determinadas.

HTML: cuyas siglas son HyperText Markup Language y se refieren a un lenguaje de marcado para el desarrollo de documentos web.

Script: es un archivo con una secuencia de comandos utilizada para ejecutar operaciones o escribir un programa.

HTTP: también conocido como HyperText Transfer Protocol, representa un protocolo que permite intercambiar documentos hipertexto, como por ejemplo ficheros HTML.

MySQL: se trata de un sistema gestor de bases de datos relacionales que permite administrar consultas y almacenar información mediante llamadas a una base de datos.

SQL: es un lenguaje de dominio específico orientado a diseñar y representar la gestión de información en bases de datos relacionales.

JavaScript: es un lenguaje de programación principalmente enfocado en páginas web.

JS: es una abreviatura para referirse a JavaScript.

Canvas: este término hace referencia a un elemento de HTML que permite la generación de gráficas de forma dinámica mediante el uso de scripts.

CSS: del inglés, Cascading Style Sheet, referencia un lenguaje de estilos empleado para modificar la presentación de un documento HTML.

Backend: se trata de la parte que desarrolla la lógica de una aplicación y que no es accesible de forma directa por un usuario.

PHP: conocido como Hypertext Preprocessor, es un lenguaje especialmente adaptado para el desarrollo web.

URL: conocido del inglés como Uniform Resource Locator, representa una dirección única a algún recurso disponible alojado en Internet.

Windows: es un sistema operativo desarrollado por Microsoft.

Linux: es un sistema operativo libre y de código abierto.

Apache: se trata de un software de servidor web HTTP de código abierto y gratuito para diferentes plataformas.

TFG: cuyas siglas son Trabajo Final de Grado.

Feedback: se traduce como retroalimentación, y en este contexto consiste en proporcionar información sobre el resultado de una acción a quién la ha realizado.

Mockup: es un prototipo o una maqueta donde se puede representar diferentes funcionalidades y evaluar el diseño de un producto a desarrollar.

LOPD: también conocida como *Ley orgánica de protección de datos*, cuyo principal objetivo es reconocer y proteger el derecho de las personas en la recogida y el tratamiento de sus datos.

Toolbar: también conocida en español como barra de herramientas. Se trata de un componente de una interfaz gráfica que permite el acceso a diferentes funciones. Generalmente se muestra en forma de un bloque o menú con botones e iconos.

UML: Es un lenguaje gráfico empleado para describir y especificar métodos, procesos y esquematizar las funcionalidades de un sistema.

WAMP: es una herramienta que permite ejecutar un servidor Web, SQL y PHP sobre Windows.

Página/Sitio web: Es un documento web accesible desde cualquier navegador web con internet, que puede incluir diferentes tipos de datos, como puede ser textos, hipervínculos, imágenes, enlaces y programas, entre otros.

Footer: es conocido como pie de página y es una sección fija en la parte inferior de las páginas web donde generalmente se visualiza la información de contacto de una empresa, botones a las redes sociales o enlaces de navegación.

Java: lenguaje de programación orientado a objetos.

Hosting: se trata de un servicio que provee alojamiento de sitios web y sus contenidos, permitiendo el acceso desde Internet.

UPV: representan las siglas de la Universitat Politècnica de València.

ASIC: también conocido como Área de Sistemas de la información y las Comunicaciones. Es una entidad perteneciente a la UPV que presta y gestiona servicios TIC.

ASP: o Active Server Pages, es una tecnología desarrollada por Microsoft enfocada para la creación de sitios web dinámicos.

Plesk: es un software comercial para hosting que se ejecuta en los sistemas operativos Windows y Linux.

Backup: se trata de una copia de seguridad o una copia de respaldo con el fin de poder recuperar datos en caso de su pérdida.

SHA 512: se trata de un algoritmo criptográfico de 512 bits de longitud, del inglés Secure Hashing Algorithm.

SSS: las siglas simbolizan Sistemas Secuencial Síncrono.

1. Introducción

1.1. Contexto

En el primer año del Grado de Ingeniería Informática de la Escuela Técnica Superior de Ingeniería Informática de la Universitat Politècnica de València se imparte la asignatura “Fundamentos de computadores”. En esta asignatura se pretende ayudar al alumnado a comprender el funcionamiento básico de un procesador, así como el diseño de sus componentes, entre otros objetivos.

Una de las competencias que los alumnos deben adquirir es el análisis de sistemas secuenciales síncronos (SSS). Para facilitar dicha tarea, se ha diseñado una solución cómoda para que un alumno pueda evaluar sus capacidades de análisis y comprensión. Se plantea una página web que incorpore unos ejercicios acerca de los diferentes tipos de biestables aprendidos en la asignatura y las posibles combinaciones y conexiones que se puedan dar entre ellos, junto a una representación gráfica de la evolución en el tiempo por parte de las componentes involucradas (conocida comúnmente como un cronograma). Por simplicidad, se utilizará el nombre de entrenador para referirse a la aplicación web objetivo de este trabajo.

1.2. Estado del arte

En la actualidad, se muestra atención por diferentes enfoques de aprendizaje mediante herramientas digitales para facilitar e incrementar el interés de los estudiantes. Algunos de estos métodos ya existen en el ámbito tecnológico y se están implementando en instituciones de enseñanza, como es el caso de programas software para la simulación de circuitos como Crocodile, AutoCAD, etc. Estas herramientas pueden ser gratuitas o de pago. Sin embargo, el mayor desafío reside en la complejidad de su manejo, requiriendo una curva de aprendizaje que no es asumible en el primer curso del grado.

De esta manera, los profesores acordaron emplear el apoyo de herramientas digitales diseñadas exclusivamente para el uso y la capacitación de los estudiantes de informática. Estas herramientas serán rigurosas en su componente tecnológica pero sencillas en el aprendizaje de su uso. Hasta ahora se han conseguido varias soluciones, tales como entrenadores para la construcción de mapas de Karnaugh [1] con soluciones y proceso guiado, para la representación de números enteros [2], para operaciones bitwise [3] o realización de ejercicios IEEE754 [4], entre otros.

Este trabajo es una evolución de un trabajo final de grado anterior, consistente en un entrenador gráfico de biestables [5]. La solución propuesta se basa en la lógica del entrenador de biestables, ampliada y ajustada para la realización de ejercicios de análisis de sistemas secuenciales síncronos.

1.3. Motivación

Los estudiantes pueden encontrar dificultades en esta etapa temprana de aprendizaje al tratar de resolver circuitos con tecnologías y componentes recién vistas, como es el caso de los biestables. Por esa razón, una metodología que les pueda apoyar y guiar en el proceso de resolución es interesante, sobre todo desde el punto de vista de una alumna que ya ha cursado la asignatura.

El tema de este trabajo resultó atractivo y alentador, en particular llevar a la práctica los conocimientos adquiridos a partir de las asignaturas de desarrollo de proyectos de la rama de software. Fue una buena oportunidad y una nueva experiencia poder realizar el desarrollo de un producto desde la gestión de requisitos, la metodología de trabajo, el alcance, la dedicación de tiempo y la separación de tareas, aplicar los conocimientos técnicos en las fases de programación, conocer y comprender como aportar creatividad y calidad a un producto software, cuya finalidad justamente va a ser de utilidad a otros estudiantes en el futuro.

1.4. Objetivos

El objetivo principal es desarrollar una página web donde un alumno pueda resolver ejercicios de análisis de circuitos secuenciales síncronos implementados con biestables y puertas lógicas.

Entre otros objetivos, se desea conseguir:

- Acceso a diferentes tipos de ejercicios de circuitos secuenciales síncronos con biestables de tipo D, JK y T.
- Interacción por parte del usuario con un cronograma, permitiéndole establecer el nivel lógico de cada semiperiodo.
- Permitir resolver y comprobar un sistema de ecuaciones de las entradas de los biestables de los circuitos planteados.
- Permitir resolver y comprobar el cronograma del circuito.
- Mostrar al usuario la solución correcta de las ecuaciones y el cronograma del circuito
- Recibir *feedback* en formato estadístico acerca de los fallos y aciertos conseguidos en la resolución interactiva con el cronograma.
- Control y gestión de usuarios para los diferentes perfiles analizados.
- Obtener una plantilla para permitir a los profesores añadir nuevos ejercicios.

1.5. Plan temporal

Este apartado define el plan de trabajo previsto para el desarrollo de la aplicación web de este TFG.

Para la especificación de este plan se ha considerado el trabajo a media jornada de la alumna en las prácticas de empresa. De esta forma, la dedicación de tiempo sería entre 4 y 5 horas diarias.

En primer lugar, se han establecido tres semanas para la fase de análisis y especificación de requisitos. En la primera semana se analizará y definirá el comportamiento, los requisitos funcionales y no funcionales de la aplicación web y se esperará la aprobación por parte del profesor. A partir de la segunda semana se empezará con la especificación de los casos de uso y los diagramas necesarios.

En las próximas tres semanas comenzaría la etapa de diseño, estudiando y determinando la arquitectura más conveniente según las tecnologías necesarias para el desarrollo de la aplicación web, además de definir y diseñar los prototipos de la aplicación basados en los requisitos aprobados previamente.

Una vez acreditada y aceptada la propuesta, se llevará a cabo la fase de implementación, que consistirá en siete semanas de programación y que exigirá una fundamental implicación por parte del profesor para realizar revisiones. Por otro lado, se comprobarán las diferentes funcionalidades que se irán añadiendo a la aplicación, por lo que la etapa de pruebas tendría una duración repartida entre las mismas semanas de programación.

Finalmente, en la última semana se procede a desplegar la aplicación y probar el acceso por parte de diferentes usuarios y el correcto funcionamiento.

La Tabla 1 muestra la planificación realizada.

		Semanas												
Tareas	Duración	1	2	3	4	5	6	7	8	9	10	11	12	13
Análisis y Especificación de requisitos	12 días	■	■	■										
Diseño	11 días			■	■	■								
Implementación	45 días						■	■	■	■	■	■	■	
Pruebas	19 días						■	■	■	■	■	■	■	
Despliegue	4 días													■

Tabla 1. Plan de trabajo

1.6. Presupuesto

El desarrollo de un proyecto software es un proceso extenso y laborioso que conlleva el uso de diferentes recursos, ya sea el personal, el material o los servicios imprescindibles para poder llevarlo a cabo.

Con el fin de cubrir el consumo y la gestión de estos recursos, se ha realizado un análisis previo para estimar los gastos necesarios en la elaboración de la aplicación web. Para ello se han considerado los siguientes recursos: el trabajo realizado por la alumna, la colaboración del tutor y el material empleado.

Del estudio realizado por el Consejo de Colegios de Ingeniería Informática [6] se desprende que el salario de un recién titulado oscila entre 12.000€ y 30.000€ al año, mientras que el de un profesor universitario puede estar entre los 34 000 € y 39 000 € al año. Teniendo en cuenta estos números, se puede estimar un coste por hora de 12 € para un ingeniero recién egresado y 24 € por hora para un profesor universitario.

En cuanto al coste del material, se ha empleado un equipo de gama media-alta, con un coste de adquisición entre 800 € y 900 € y un tiempo de vida útil de 4 a 5 años, lo que representa un coste de adquisición aproximado de 0,11 € por hora. El equipo dispone de una fuente de alimentación de 300 W, lo que equivale a un consumo eléctrico de 0,26 kWh y se ha considerado un precio medio de 0,18 € por kWh.

Para desplegar la aplicación en un servidor web o tener almacenada una base de datos, se añadiría el precio conveniente según el servicio que requiera el cliente. Puesto que se trata de una aplicación que alojará la propia universidad, no se ha visto necesario tener en consideración dichos gastos.

Finalmente, se considera el tiempo dedicado por parte de los diferentes recursos en horas y se calcula el coste total del proyecto, como se puede observar en la Tabla 2, estimado en **4610,24 €**.

Recurso	Horas	Coste/hora	
Personal			
Alumna	320	12 €	3840 €
Tutor	30	24 €	720 €
Material			
Adquisición equipo	320	0,11 €	35,2 €
Consumo eléctrico	320	0,047 €	15,04 €
Coste total			4610,24 €

Tabla 2. Presupuesto estimado del proyecto

1.7. Estructura de la memoria

En este documento se detallan las diferentes etapas llevadas a cabo para el desarrollo de este TFG. Las etapas se dividen en los siguientes capítulos:

- **Capítulo 1:** se introduce el problema planteado, el estado del arte, la motivación, se listan los objetivos que se desean alcanzar, el plan temporal y el presupuesto.
- **Capítulo 2:** se repasa brevemente definiciones y conceptos que se consideran importantes y significativas dentro del contexto de este trabajo [7].
- **Capítulo 3:** se indica la metodología de trabajo empleada y se introducen las diferentes etapas que la componen.
- **Capítulo 4:** se describe la fase de análisis y especificación de requisitos, donde se detalla las diferentes funcionalidades, junto a diagramas y modelos que componen el sistema.
- **Capítulo 5:** se expone la arquitectura utilizada explicando el rol de cada una de sus capas y se señalan los prototipos de las interfaces gráficas de usuario.
- **Capítulo 6:** se presentan las tecnologías y herramientas empleadas para el desarrollo de este proyecto.
- **Capítulo 7:** se profundiza en el proceso de desarrollo y la solución alcanzada, además de la aplicación de algunas técnicas de refactorización de código.
- **Capítulo 8:** se describen los diferentes tipos de pruebas que se han llevado a cabo, además del despliegue de la aplicación.
- **Capítulo 9:** se comenta brevemente los objetivos alcanzados, una valoración personal acerca del trabajo realizado y unas posibles mejoras de cara al futuro.
- **Capítulo 10:** este apartado muestra las referencias bibliográficas que se han utilizado para la elaboración de este TFG.



2. Conceptos generales

Biestado: Es un circuito secuencial, es decir, con capacidad de almacenar información. Tiene dos estados posibles (0 y 1) y está constituido por dos salidas complementarias que se pueden modificar mediante señales aplicadas a sus entradas. El comportamiento de un biestado se describe mediante su tabla de excitación o de funcionamiento.

Según el tipo de entradas se distinguen entre biestados *asíncronos* (solamente tienen entradas de control) o *síncronos* (cuentan además con una entrada sincronizada de reloj). Estos últimos son los más utilizados y dependiendo de la configuración del circuito serán de tipo D, T o JK.

En la Ilustración 1 se muestra un ejemplo de biestados síncrono de tipo D y su tabla de funcionamiento en la Ilustración 1:

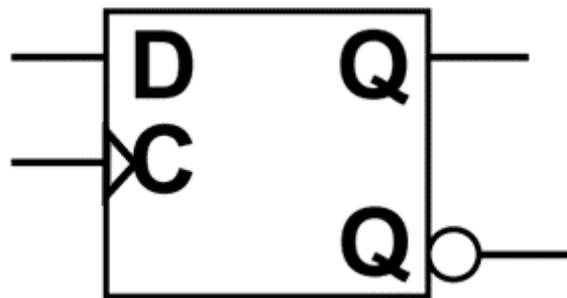


Ilustración 1. Símbolo lógico de biestado D activo por flanco de subida.

C	D	Q(t+1)	/Q(t+1)
0/1/↓	X	Q(t)	/Q(t)
↑	0	0	1
↑	1	1	0

C = entrada de reloj
D = entrada de control
Q = estado actual
Q(t+1) = estado siguiente

Tabla 3. Tabla de funcionamiento de un biestado D activo por flanco de subida.

Flanco: se trata de una transición de un nivel bajo a alto (flanco de subida) o de un nivel alto a bajo (flanco de bajada) en una señal digital como se muestra en la Ilustración 2 .

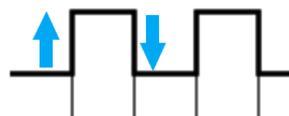


Ilustración 2. Representación de flancos de subida y bajada en una señal de reloj.

Circuitos secuenciales de biestables: Representa una agrupación de varios tipos de biestables que conectan sus salidas y entradas mediante puertas lógicas y activados por un flanco de reloj. Los valores de salida de estos circuitos pueden depender de entradas externas y/o estados anteriores (Ilustración 3). Se dice que es un circuito síncrono si todos los biestables reciben la misma señal de reloj y son activos por el mismo flanco. En caso contrario se llama asíncrono.

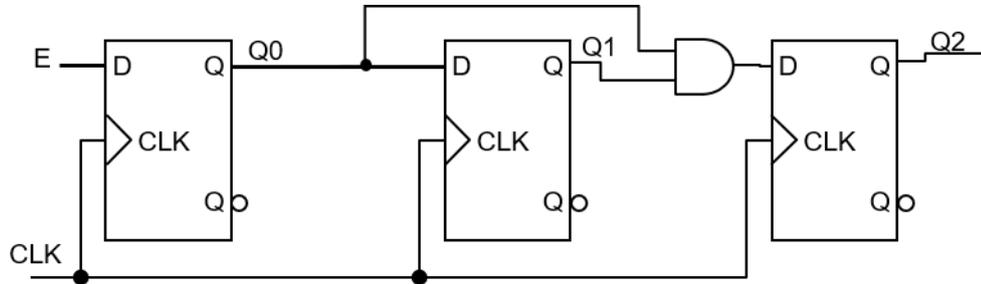


Ilustración 3. Circuito secuencial síncrono con biestables tipo D activos por flanco de subida.

Cronograma: Es una representación gráfica de la evolución en el tiempo de las entradas y salidas de un circuito. Se muestra un ejemplo con una señal de entrada, otra de reloj y tres salidas en la Ilustración 4.

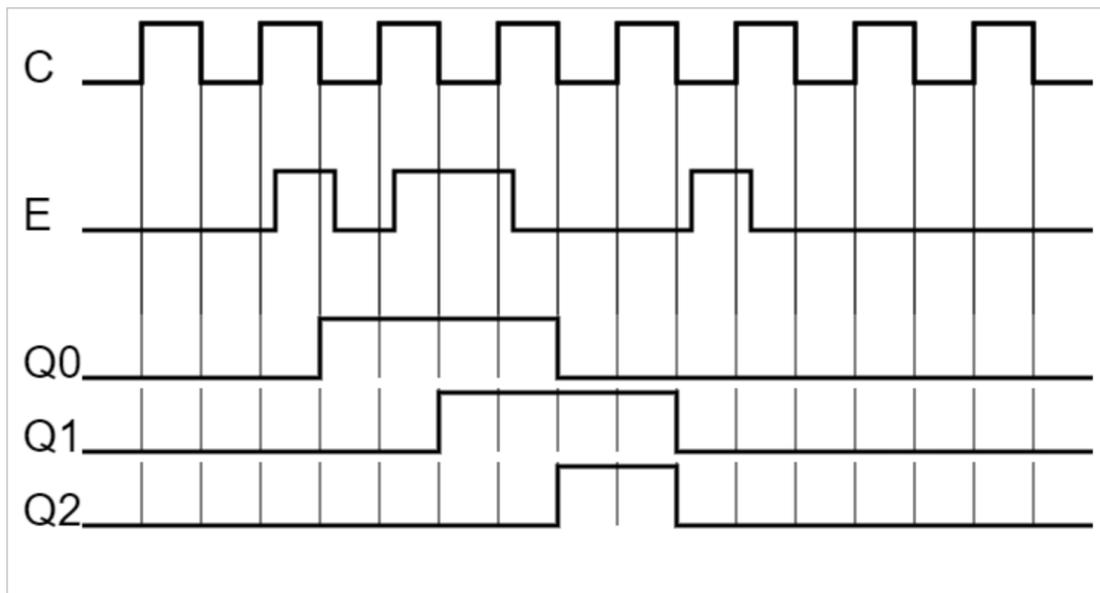


Ilustración 4. Cronograma de un circuito con 3 salidas y una entrada.

Ecuación: Se trata de una expresión algebraica que representa el valor de las entradas de control de los biestables. Al juntar varios biestables en un circuito se obtiene un sistema de ecuaciones con dependencias que facilita la resolución de los valores de estado y salida. Con estas ecuaciones se puede dibujar las señales de salida del circuito en un cronograma. A continuación, se muestra las ecuaciones de la Ilustración 3:

$$\begin{aligned} D_0 &= E \\ D_1 &= Q_0 \\ D_2 &= Q_0 * Q_1 \end{aligned}$$

3. Metodología

Es importante definir y seguir un proceso en la planificación y constitución de un producto o proyecto software, ya que aporta una mejor organización y comprensión del alcance y la evolución del sistema a realizar.

Para gestionar el ciclo de vida de la aplicación se ha seguido un modelo en cascada [8]. Esta metodología es un enfoque de desarrollo de forma lineal, en donde se han ordenado las diferentes etapas a llevar a cabo, como se muestra en la Ilustración 5:

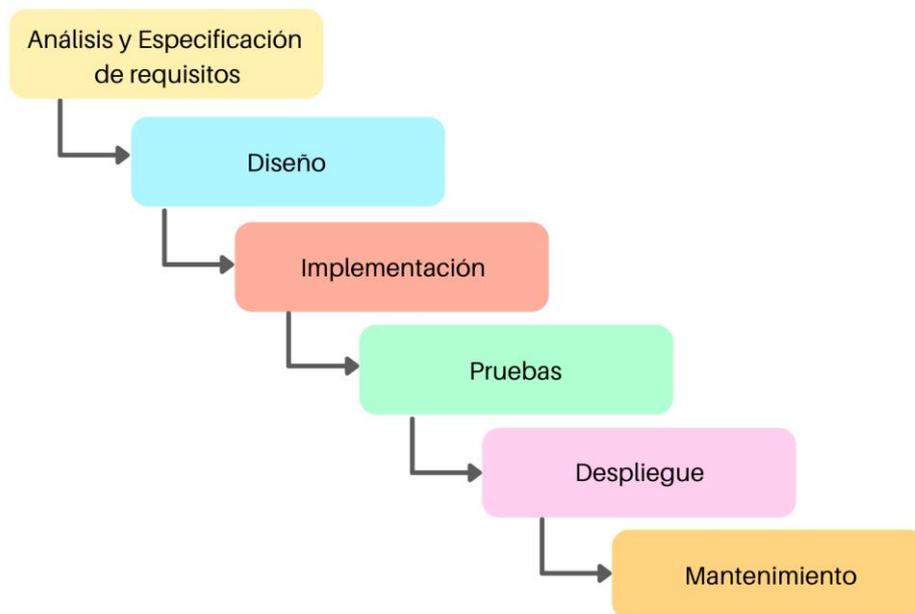


Ilustración 5. Representación del modelo en cascada.

A continuación, se describe brevemente cada proceso que constituye esta metodología:

- **Análisis y especificación de requisitos:** en esta etapa se realiza un estudio sobre las diferentes características y funcionalidades que se desean para la aplicación del entrenador.
- **Diseño:** tras declarar las funcionalidades que se quieren conseguir en la etapa anterior, se plantea una estructura arquitectónica y se esbozan unos prototipos iniciales para las interfaces gráficas del entrenador.
- **Implementación:** esta etapa requiere programar los requisitos funcionales planteados anteriormente con el apoyo de las herramientas y tecnologías seleccionadas.
- **Pruebas:** al finalizar la parte de implementación, se comprueba que la aplicación funciona correctamente y que además cumple con los objetivos planteados en la fase de análisis y especificación de requisitos.
- **Despliegue:** esta etapa se encarga de publicar la aplicación en un servidor para que los usuarios puedan acceder desde Internet.

- **Mantenimiento:** esta última etapa tiene como objetivo el uso posterior por parte de los profesores, que se encargarán de actualizar y mantener la aplicación funcional ante cualquier fallo o cambio en alguna de las tecnologías empleadas.

Estas fases se han repartido durante los siguientes capítulos de este documento, a excepción del mantenimiento, que consistirá más bien en el sostenimiento y persistencia que tendrá en el futuro.

4. Análisis y Especificación de requisitos

Este apartado representa una fase fundamental en el desarrollo y posterior mantenimiento de un proyecto. La calidad del software se ve afectada por la falta de documentación y problemas en el entendimiento del propósito y los objetivos con el cual el sistema software fue concebido originalmente.

En consecuencia, se plantea analizar y comprender el dominio del problema, capturar y especificar los requisitos funcionales y no funcionales, así como validar y verificar la correcta comprensión de las partes implicadas en el desarrollo e implementación del sistema.

A continuación, se muestra más en profundidad, el análisis realizado durante la fase inicial del proyecto, que ha servido como base para afianzar y modelar los aspectos funcionales de la aplicación entrenador. Para ello, se ha usado como modelo el estándar IEEE-830 [9].

4.1. Glosario de términos

Ejercicio: Se referencia al sitio web donde se puede encontrar y resolver las ecuaciones y el cronograma de un circuito secuencial síncrono del entrenador.

Actor: Establece el rol o papel ejecutado por una persona o entidad que interacciona con el sistema.

Página de acceso: Denominación a la página web donde se encuentran los formularios de registro e inicio de sesión. Se necesita registrar o iniciar sesión para acceder al contenido de la aplicación.

Página principal: Se designa a las diferentes páginas iniciales de un usuario autenticado.

Persona: Es un usuario no registrado en el sistema.

Usuario: Indica una cuenta dada de alta con un perfil determinado (administrador, profesor o alumno).

Titulación: Representa el grado o la carrera donde un profesor puede estar asociado o un alumno cursarla.

Asignatura: Se trata de la materia que imparte un profesor y a la que asiste un alumno.

Curso: Coincide con el año escolar.

Grupo: Indica el número de la clase que un alumno puede estar asignado y donde enseña un profesor.

Periodo: En una señal periódica describe la duración desde un flanco dado que atraviesa los dos estados (0 y 1), denominados en una gráfica como niveles, hasta el siguiente flanco, coincidente con el anterior.

Semiperiodo: En una señal periódica representa la duración desde un flanco que atraviesa un solo estado para alcanzar el flanco opuesto.

4.2. Diagrama de contexto

El diagrama de contexto permite una visión de alto nivel de la aplicación web, donde se definen los límites entre el sistema y los actores o entidades externas que interactúan con él.

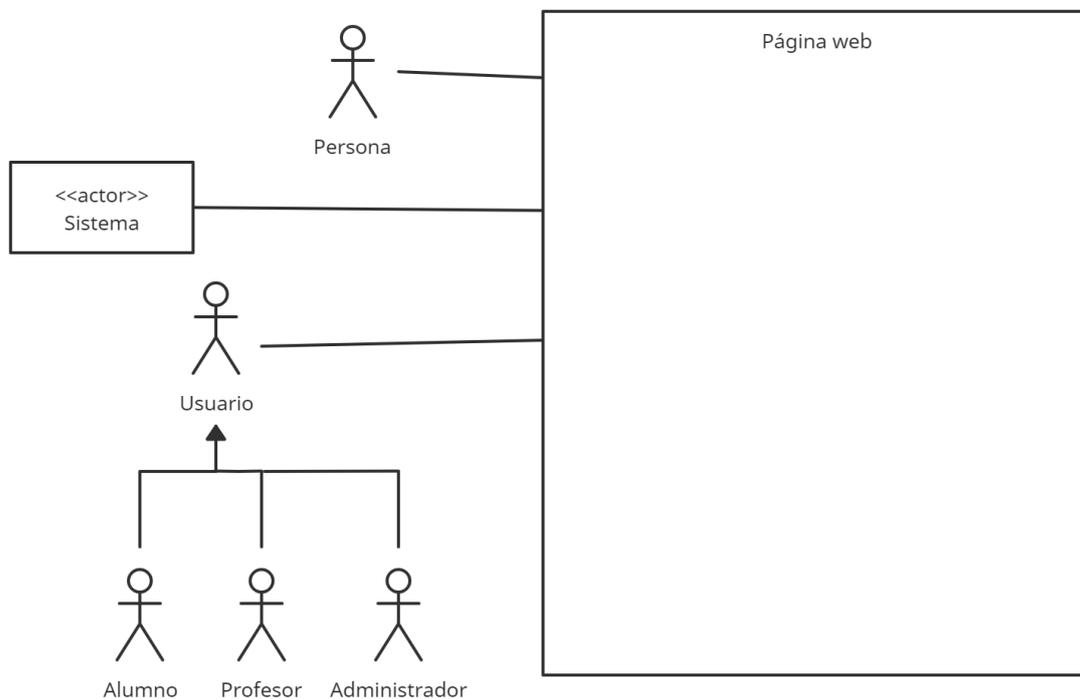


Ilustración 6. Diagrama de contexto.

Como se puede observar en la Ilustración 6, el contexto del programa software es la página web y los actores que se pueden encontrar son:

- Una persona cualquiera que no sea un usuario identificado en el sistema.
- Varios tipos de usuarios que comparten ciertas características de forma general y luego de forma individual tienen acceso a servicios exclusivos de su perfil.
- El actor sistema es una entidad que realiza las ejecuciones internas y administra servicios disponibles para los usuarios.

4.3. Modelo de dominio

El modelo de dominio representa un glosario gráfico para identificar conceptos en el contexto del sistema a desarrollar. Su objetivo principal es facilitar la comunicación entre clientes y desarrolladores estableciendo un lenguaje común. No se centra en aspectos de diseño, sin embargo, puede evolucionar a un modelo de análisis que represente la estructura de la aplicación (véase la Ilustración 14 e Ilustración 34).

Se distinguen y determinan los conceptos más importantes del dominio del problema: entidades que registran información o la manipulan (profesor, alumno, historial...), eventos que puedan ocurrir en el contexto del sistema (cronograma, ecuaciones...) y las relaciones existentes entre ellos (crear, modificar, resolver, pertenece a...).

En la Ilustración 7 se puede observar el diagrama del modelo de dominio del entrenador de circuitos secuenciales representado en notación UML como un diagrama de clases.

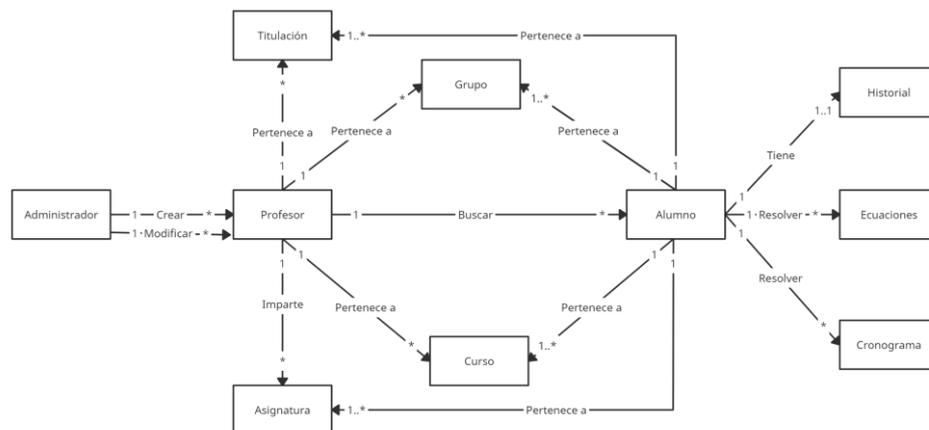


Ilustración 7. Diagrama del modelo de dominio.

4.4. Casos de uso

En este apartado se detallan los diferentes requisitos funcionales del sistema. Cabe señalar que se han descompuesto en varias características y representado gráficamente mediante diagramas con sus respectivos casos de uso: la gestión de usuarios y las funcionalidades de los perfiles de usuario disponibles (administrador, profesor, alumno), además de las acciones que realiza el sistema.

Para cada caso de uso asociado se ha definido una plantilla con las siguientes características:

Nombre del caso de uso	
Descripción	Explicación breve que indica el propósito de esta funcionalidad.
Actor	Entidad que realiza la acción del caso de uso.
Relaciones	Conexión o vínculo que pueda tener con uno o varios casos de uso.
Precondición	Requisito previo necesario para realizar la funcionalidad descrita por el caso de uso.
Secuencia	Flujo de eventos o pasos que se tienen que realizar para la ejecución correcta del caso de uso.
Postcondición	Condiciones que se deben cumplir al finalizar la ejecución de la secuencia.
Comentarios	Observaciones y valoraciones que se pueden o deben tener en cuenta para el cumplimiento del requisito.

Tabla 4. Plantilla de casos de uso

4.4.1. Gestión de usuarios

La Ilustración 8 muestra el diagrama de casos de uso con las funcionalidades de una persona y un usuario cualquiera y desde la Tabla 5 a Tabla 7 se especifican cada uno de ellos, según la plantilla indicada anteriormente en la Tabla 4:

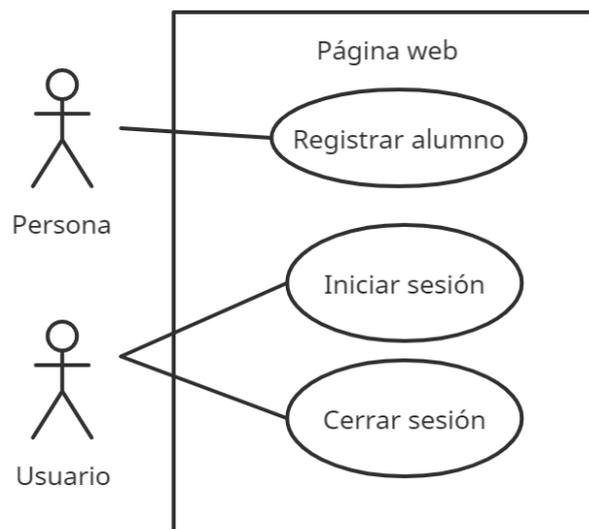


Ilustración 8. Diagrama de casos de uso para la gestión de usuarios.

Caso de uso 1: Registrar alumno	
Descripción	Crear una cuenta como alumno para acceder a las funcionalidades de la página web.
Actor	Persona
Relaciones	-
Precondición	Estar en la página de acceso de la aplicación web.
Secuencia	1. Rellenar los campos de registro: nombre completo, correo electrónico, nombre de usuario, contraseña, titulación, asignatura, grupo y el curso. 2. Pulsar el botón “Registrar”.
Postcondición	Si todos los datos del formulario están completados y son correctos, el usuario se registrará en la base de datos y será redirigido a la página principal del perfil de alumno.
Comentarios	En la página de acceso solamente se podrán dar de alta perfiles de alumnos. Se puede emplear menús desplegables para los valores de titulación, asignatura, grupo y curso. Se comprobará que el usuario no esté dado de alta en el sistema con el nombre de usuario y correo indicado.

Tabla 5. Caso de uso: Registrar alumno

Caso de uso 2: Inicio sesión	
Descripción	Acceder a las funcionalidades de un usuario. Dependiendo del tipo de usuario, la página principal será diferente.
Actor	Usuario
Relaciones	-
Precondición	Estar en la página de acceso y tener un usuario dado de alta con el perfil que se desea acceder.
Secuencia	1. Rellenar los campos de usuario y contraseña. 2. Pulsar el botón “Iniciar sesión”.
Postcondición	Si los datos son correctos, se iniciará sesión con el usuario en cuestión y se redirigirá a la página principal del perfil asociado.
Comentarios	El campo usuario aceptará tanto nombre de usuarios como el correo electrónico. Los diferentes perfiles de usuario solo tendrán acceso a sus propias páginas principales.

Tabla 6. Caso de uso: Inicio sesión

Caso de uso 3: Cerrar sesión	
Descripción	Finalizar la sesión de un usuario autenticado en la aplicación web.
Actor	Usuario
Relaciones	-
Precondición	Estar logado con alguna cuenta de usuario.
Secuencia	1. Pulsar el botón “Cerrar sesión”.
Postcondición	El usuario dejará de tener una sesión activa y se le habrá redirigido a la página de acceso.
Comentarios	

Tabla 7. Caso de uso: Cerrar sesión

4.4.2. Funcionalidades del administrador

La Ilustración 9 muestra el diagrama con las funcionalidades de un administrador y a continuación, desde la Tabla 8 a Tabla 18 se especifican cada una de ellas:

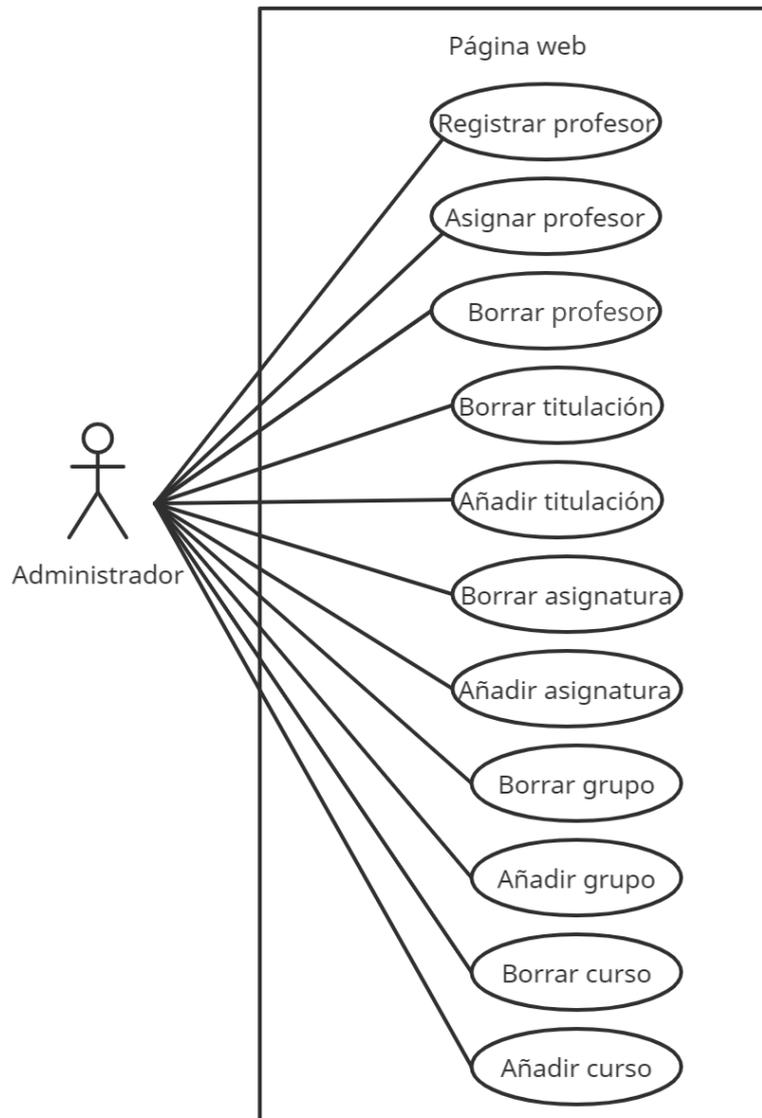


Ilustración 9. Diagrama de casos de uso para las funcionalidades del administrador.

Caso de uso 4: Registrar profesor	
Descripción	Dar de alta un perfil de usuario de tipo profesor.
Actor	Administrador
Relaciones	-
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Rellenar los campos de nombre completo, correo electrónico, nombre de usuario y contraseña. 2. Pulsar el botón “Registrar”.
Postcondición	Se debería almacenar en la base de datos el nuevo perfil de profesor con los datos indicados. Se le mantendrá al usuario en la página principal.
Comentarios	

Tabla 8. Caso de uso: Registrar profesor

Caso de uso 5: Asignar profesor	
Descripción	Asociar un profesor existente a un curso, asignatura, titulación y grupo existente.
Actor	Administrador
Relaciones	-
Precondición	Estar logado con un perfil de administrador. Tener algún profesor dado de alta en la base de datos.
Secuencia	1. Seleccionar un profesor existente, un curso, una titulación, asignatura y un grupo. 2. Pulsar el botón “Asignar”.
Postcondición	Se asociará al profesor correspondiente con los valores elegidos y se guardarán los valores en la base de datos.
Comentarios	Un profesor podrá estar vinculado a más de un curso, asignatura, titulación y grupo. Sin embargo, no podrá relacionarse dos veces con todos los mismos valores.

Tabla 9. Caso de uso: Asignar profesor

Caso de uso 6: Borrar profesor	
Descripción	Borrar un profesor existente de la base de datos.
Actor	Administrador
Relaciones	-
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Seleccionar un profesor dado de alta. 2. Pulsar el botón “Borrar”
Postcondición	Se eliminará en la base de datos el usuario del profesor escogido y todas las asociaciones que tenga (titulaciones, asignaturas, cursos, grupos)
Comentarios	

Tabla 10. Caso de uso: Borrar profesor

Caso de uso 7: Borrar titulación	
Descripción	Borrar una titulación existente de la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Seleccionar una titulación existente. 2. Pulsar el botón “Borrar”.
Postcondición	Se eliminará en la base de datos el valor de la titulación escogida y los usuarios que la tenían relacionada cambiará el valor de su campo a NULL.
Comentarios	

Tabla 11. Caso de uso: Borrar titulación

Caso de uso 8: Añadir titulación	
Descripción	Añadir una nueva titulación en la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Rellenar el campo de titulación con un nuevo valor. 2. Pulsar el botón “Añadir”.
Postcondición	Se añadirá el nuevo valor en la base de datos de titulaciones.
Comentarios	

Tabla 12. Caso de uso: Añadir titulación

Caso de uso 9: Borrar asignatura	
Descripción	Borrar una asignatura existente de la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Seleccionar asignatura existente. 2. Pulsar el botón “Borrar”.
Postcondición	Se eliminará en la base de datos el valor de la asignatura escogida y los usuarios que la tenían relacionada cambiará el valor de su campo a NULL.
Comentarios	

Tabla 13. Caso de uso: Borrar asignatura

Caso de uso 10: Añadir asignatura	
Descripción	Añadir una nueva asignatura en la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Rellenar el campo de asignatura con un nuevo valor. 2. Pulsar el botón “Añadir”.
Postcondición	Se añadirá el nuevo valor en la base de datos de asignaturas.
Comentarios	

Tabla 14. Caso de uso: Añadir asignatura

Caso de uso 11: Borrar grupo	
Descripción	Borrar un grupo existente de la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Seleccionar un grupo existente. 2. Pulsar el botón “Borrar”.
Postcondición	Se eliminará en la base de datos el valor del grupo escogido y los usuarios que lo tenían relacionado cambiará el valor de su campo a NULL.
Comentarios	

Tabla 15. Caso de uso: Borrar grupo

Caso de uso 12: Añadir grupo	
Descripción	Añadir un nuevo grupo en la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Rellenar el campo de grupo con un nuevo valor. 2. Pulsar el botón “Añadir”.
Postcondición	Se añadirá el nuevo valor en la base de datos de grupos.
Comentarios	

Tabla 16. Caso de uso: Añadir grupo

Caso de uso 13: Borrar curso	
Descripción	Borrar un curso existente de la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Seleccionar un curso existente. 2. Pulsar el botón “Borrar”.
Postcondición	Se eliminará en la base de datos el valor del curso escogido y los usuarios que lo tenían relacionado cambiará el valor de su campo a NULL.
Comentarios	

Tabla 17. Caso de uso: Borrar curso

Caso de uso 14: Añadir curso	
Descripción	Añadir un nuevo curso en la base de datos.
Actor	Administrador
Relaciones	
Precondición	Estar logado con un perfil de administrador.
Secuencia	1. Rellenar el campo de curso con un nuevo valor. 2. Pulsar el botón “Añadir”.
Postcondición	Se añadirá el nuevo valor en la base de datos de cursos.
Comentarios	

Tabla 18. Caso de uso: Añadir curso

4.4.3. Funcionalidades del profesor

La Ilustración 10 muestra el diagrama de casos de uso del profesor y en la Tabla 19 se describe la funcionalidad:

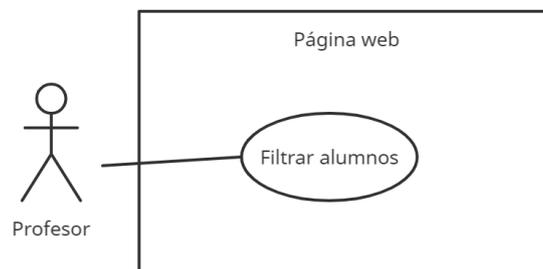


Ilustración 10. Diagrama de casos de uso para las funcionalidades del profesor.

Caso de uso 15: Filtrar alumnos	
Descripción	Buscar y mostrar los alumnos de la base de datos basados en los criterios indicados en el filtro.
Actor	Profesor
Relaciones	-
Precondición	Estar logado con un perfil de profesor y tener algún alumno dado de alta en la base de datos en los criterios respectivos.
Secuencia	<ol style="list-style-type: none"> 1. Seleccionar un valor para los campos de titulación, curso, asignatura, grupo y tipo de ejercicio. 2. Pulsar el botón “Buscar”.
Postcondición	Se mostrará en una tabla todos los alumnos de la base de datos que coinciden con los valores seleccionados.
Comentarios	Las columnas de la tabla serán: nombre del alumno, correo electrónico, titulación, curso, asignatura, grupo, tipo de ejercicio, numero de aciertos y fallos totales conseguidos para el tipo de ejercicio elegido.

Tabla 19. Caso de uso: Filtrar alumnos

4.4.4. Funcionalidades del alumno

La Ilustración 11 muestra el diagrama de casos de uso con las funcionalidades de un alumno y a partir de la Tabla 20 hasta la Tabla 45 se detallan cada una de ellas:

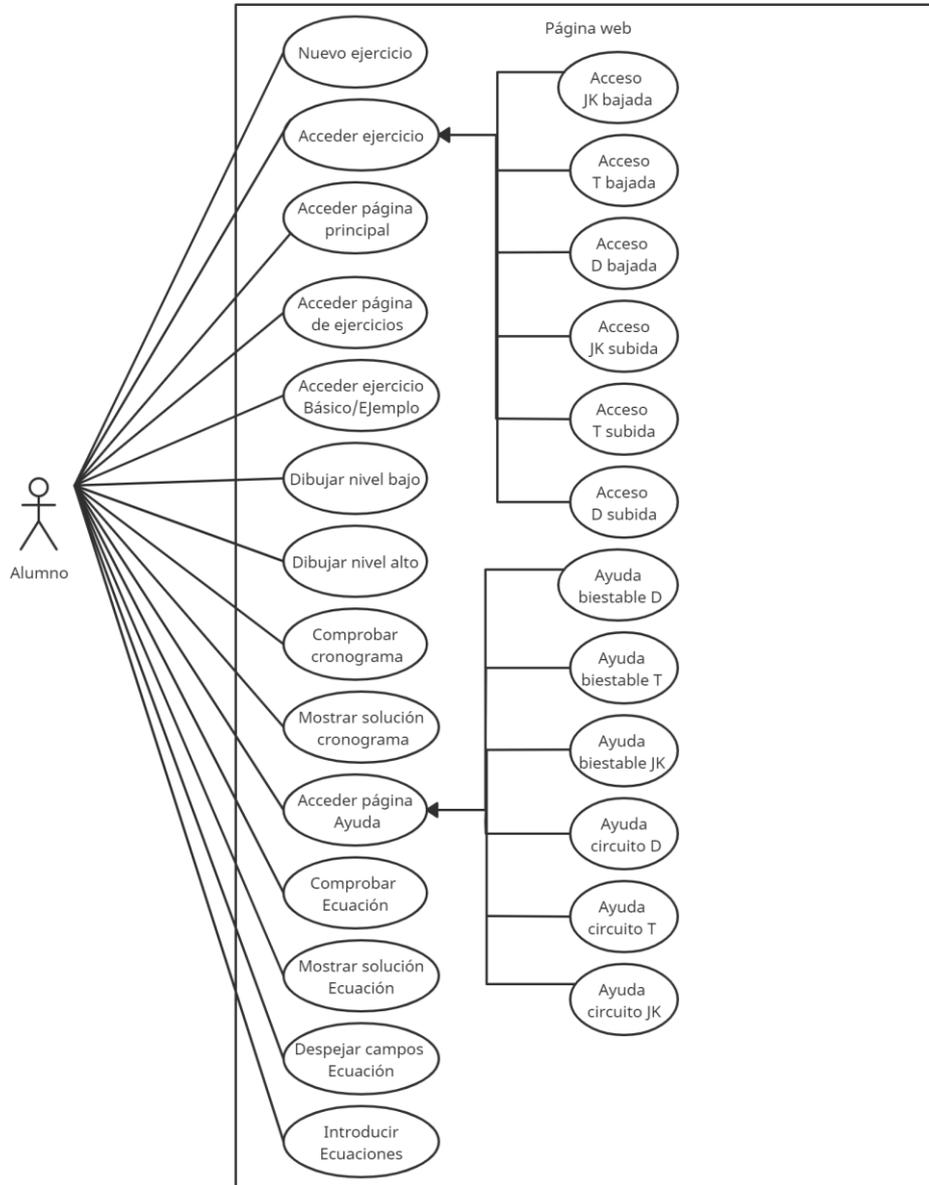


Ilustración 11. Diagrama de casos de uso para las funcionalidades del alumno.

Caso de uso 16: Acceder página principal	
Descripción	Se trata de un acceso directo a la página principal del perfil del alumno.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno.
Secuencia	1. Pulsar sobre el botón “Página inicio”.
Postcondición	Llevará al usuario a la página principal.
Comentarios	Este acceso directo se encontrará de manera estática en una <i>toolbar</i> que estará disponible en todos los sitios web que puede acceder un estudiante.

Tabla 20. Caso de uso: Acceder página principal

Caso de uso 17: Acceder página de ejercicios	
Descripción	Se trata de un acceso directo a la página de la lista de ejercicios existentes del perfil del alumno.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno.
Secuencia	1. Pulsar sobre el botón “Ejercicios”.
Postcondición	Llevará al usuario a una página donde se mostrará una lista de los ejercicios disponibles para ser realizados por el alumno.
Comentarios	Este acceso directo se encontrará de manera estática en una <i>toolbar</i> que estará disponible en todos los sitios web que puede acceder un estudiante.

Tabla 21. Caso de uso: Acceder página de ejercicios

Caso de uso 18: Acceder página ayuda	
Descripción	Se trata de un acceso directo a la página de ayuda.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno.
Secuencia	1. Pulsar sobre el botón “Ayuda”.
Postcondición	Redireccionará a una página donde se mostrarán definiciones e información acerca de los diferentes tipos de biestables y los circuitos que se pueden formar con ellos.
Comentarios	Este acceso directo se encontrará de manera estática en una <i>toolbar</i> que estará disponible en todos los sitios web que puede acceder un estudiante.

Tabla 22. Caso de uso: Acceder página ayuda



Caso de uso 19: Acceder ejercicio básico/ejemplo	
Descripción	Se accede a un tipo de ejercicio ejemplo. Es una opción que sirve como tutorial para enseñarle al estudiante la funcionalidad del entrenador.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y situado en la página principal.
Secuencia	1. Pulsar sobre el botón “Ejercicio tutorial”.
Postcondición	Redireccionará al usuario a una página donde se mostrará un ejercicio de un circuito secuencial de un biestable cuyos valores de entrada sean fijos.
Comentarios	

Tabla 23. Caso de uso: Acceder ejercicio básico/ejemplo

Caso de uso 20: Acceder ejercicio	
Descripción	Este caso de uso es una generalización de los diferentes tipos de ejercicios a los que podrá acceder un alumno.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno.
Secuencia	-
Postcondición	-
Comentarios	-

Tabla 24. Caso de uso: Acceder ejercicio

Caso de uso 21: Acceso D Subida	
Descripción	Acceder al ejercicio del entrenador de circuitos secuenciales con biestables de tipo D activos por flanco de subida.
Actor	Alumno
Relaciones	Acceder ejercicio
Precondición	Estar logado con un perfil de alumno y haber accedido a la página con el listado de los ejercicios existentes.
Secuencia	Dos alternativas: <ol style="list-style-type: none"> 1. Pulsar sobre el botón “Circuito de biestable D (flanco de subida)”. 2. Pulsar sobre la imagen que indica la configuración del circuito.
Postcondición	Redireccionará al usuario a una página donde se muestre un ejercicio de un circuito secuencial de biestable D por flanco de subida y cuyos valores de entrada serán aleatorios según se recarga la página o se pida un nuevo ejercicio.
Comentarios	Todas las páginas de ejercicios tendrán el mismo formato, con las diferencias en las imágenes, las ecuaciones o en el cronograma que podrá tener una o dos entradas aleatorias.

Tabla 25. Caso de uso: Acceso D Subida

Caso de uso 22: Acceso D Bajada	
Descripción	Acceder al ejercicio del entrenador de circuitos secuenciales con biestables de tipo D activos por flanco de bajada.
Actor	Alumno
Relaciones	Acceder ejercicio
Precondición	Estar logado con un perfil de alumno y haber accedido a la página con el listado de los ejercicios existentes.
Secuencia	Dos alternativas: 1. Pulsar sobre el botón “Circuito de biestable D (flanco de bajada)”. 2. Pulsar sobre la imagen que indica la configuración del circuito.
Postcondición	Redireccionará al usuario a una página donde se muestre un ejercicio de un circuito secuencial de biestable D por flanco de bajada y cuyos valores de entrada serán aleatorios según se recarga la página o se pida un nuevo ejercicio.
Comentarios	Todas las páginas de ejercicios tendrán el mismo formato, con las diferencias en las imágenes, las ecuaciones o en el cronograma que podrá tener una o dos entradas aleatorias.

Tabla 26. Caso de uso D Bajada

Caso de uso 23: Acceso T Subida	
Descripción	Acceder al ejercicio del entrenador de circuitos secuenciales con biestables de tipo T activos por flanco de subida.
Actor	Alumno
Relaciones	Acceder ejercicio
Precondición	Estar logado con un perfil de alumno y haber accedido a la página con el listado de los ejercicios existentes.
Secuencia	Dos alternativas: 1. Pulsar sobre el botón “Circuito de biestable T (flanco de subida)”. 2. Pulsar sobre la imagen que indica la configuración del circuito.
Postcondición	Redireccionará al usuario a una página donde se muestre un ejercicio de un circuito secuencial de biestable T por flanco de subida y cuyos valores de entrada serán aleatorios según se recarga la página o se pida un nuevo ejercicio.
Comentarios	Todas las páginas de ejercicios tendrán el mismo formato, con las diferencias en las imágenes, las ecuaciones o en el cronograma que podrá tener una o dos entradas aleatorias.

Tabla 27. Caso de uso: Acceso T Subida

Caso de uso 24: Acceso T Bajada	
Descripción	Acceder al ejercicio del entrenador de circuitos secuenciales con biestables de tipo T activos por flanco de bajada.
Actor	Alumno
Relaciones	Acceder ejercicio
Precondición	Estar logado con un perfil de alumno y haber accedido a la página con el listado de los ejercicios existentes.
Secuencia	Dos alternativas: <ol style="list-style-type: none"> 1. Pulsar sobre el botón “Circuito de biestable T (flanco de bajada)”. 2. Pulsar sobre la imagen que indica la configuración del circuito.
Postcondición	Redireccionará al usuario a una página donde se muestre un ejercicio de un circuito secuencial de biestable T por flanco de bajada y cuyos valores de entrada serán aleatorios según se recarga la página o se pida un nuevo ejercicio.
Comentarios	Todas las páginas de ejercicios tendrán el mismo formato, con las diferencias en las imágenes, las ecuaciones o en el cronograma que podrá tener una o dos entradas aleatorias.

Tabla 28. Caso de uso: Acceso T Bajada

Caso de uso 25: Acceso JK Subida	
Descripción	Acceder al ejercicio del entrenador de circuitos secuenciales con biestables de tipo JK activos por flanco de subida.
Actor	Alumno
Relaciones	Acceder ejercicio
Precondición	Estar logado con un perfil de alumno y haber accedido a la página con el listado de los ejercicios existentes.
Secuencia	Dos alternativas: <ol style="list-style-type: none"> 1. Pulsar sobre el botón “Circuito de biestable JK (flanco de subida)”. 2. Pulsar sobre la imagen que indica la configuración del circuito.
Postcondición	Redireccionará al usuario a una página donde se muestre un ejercicio de un circuito secuencial de biestable JK por flanco de subida y cuyos valores de entrada serán aleatorios según se recarga la página o se pida un nuevo ejercicio.
Comentarios	Todas las páginas de ejercicios tendrán el mismo formato, con las diferencias en las imágenes, las ecuaciones o en el cronograma que podrá tener una o dos entradas aleatorias.

Tabla 29. Caso de uso: Acceso JK Subida

Caso de uso 26: Acceso JK Bajada	
Descripción	Acceder al ejercicio del entrenador de circuitos secuenciales con biestables de tipo JK activos por flanco de bajada.
Actor	Alumno
Relaciones	Acceder ejercicio
Precondición	Estar logado con un perfil de alumno y haber accedido a la página con el listado de los ejercicios existentes.
Secuencia	Dos alternativas: 1. Pulsar sobre el botón “Circuito de biestable JK (flanco de bajada)”. 2. Pulsar sobre la imagen que indica la configuración del circuito.
Postcondición	Redireccionará al usuario a una página donde se muestre un ejercicio de un circuito secuencial de biestable JK por flanco de bajada y cuyos valores de entrada serán aleatorios según se recarga la página o se pida un nuevo ejercicio.
Comentarios	Todas las páginas de ejercicios tendrán el mismo formato, con las diferencias en las imágenes, las ecuaciones o en el cronograma que podrá tener una o dos entradas aleatorias.

Tabla 30. Caso de uso: Acceso JK Bajada

Caso de uso 27: Ayuda biestable D	
Descripción	Acceder a la sección de ayuda acerca del biestable D.
Actor	Alumno
Relaciones	Acceder página Ayuda
Precondición	Estar logado con un perfil de alumno y en la página principal.
Secuencia	1. Pulsar sobre la opción “Ver más” del biestable D.
Postcondición	Redireccionará al usuario a la página de ayuda y en concreto a la sección donde se describe el funcionamiento del biestable D.
Comentarios	Todos los enlaces de “Ver más” de la página principal llevarán al sitio web de ayuda, solo que en diferentes secciones.

Tabla 31. Caso de uso: Ayuda biestable D

Caso de uso 28: Ayuda biestable T	
Descripción	Acceder a la sección de ayuda acerca del biestable T.
Actor	Alumno
Relaciones	Acceder página Ayuda
Precondición	Estar logado con un perfil de alumno y en la página principal.
Secuencia	1. Pulsar sobre la opción “Ver más” del biestable T.
Postcondición	Redireccionará al usuario a la página de ayuda y en concreto a la sección donde se describe el funcionamiento del biestable T.
Comentarios	Todos los enlaces de “Ver más” de la página principal llevarán al sitio web de ayuda, solo que en diferentes secciones.

Tabla 32. Caso de uso: Ayuda biestable T

Caso de uso 29: Ayuda biestable JK	
Descripción	Acceder a la sección de ayuda acerca del biestable JK.
Actor	Alumno
Relaciones	Acceder página Ayuda
Precondición	Estar logado con un perfil de alumno y en la página principal.
Secuencia	1. Pulsar sobre la opción “Ver más” del biestable JK.
Postcondición	Redireccionará al usuario a la página de ayuda y en concreto a la sección donde se describe el funcionamiento del biestable JK.
Comentarios	Todos los enlaces de “Ver más” de la página principal llevarán al sitio web de ayuda, solo que en diferentes secciones.

Tabla 33. Caso de uso: Ayuda biestable JK

Caso de uso 30: Ayuda circuito JK	
Descripción	Acceder a la sección de ayuda acerca del funcionamiento de los circuitos secuenciales con biestables JK.
Actor	Alumno
Relaciones	Acceder página Ayuda
Precondición	Estar logado con un perfil de alumno y en la página principal.
Secuencia	1. Pulsar sobre la opción “Ver más” de los circuitos secuenciales con biestables JK.
Postcondición	Redireccionará al usuario a la página de ayuda y en concreto a la sección donde se comenta acerca de cómo resolver un circuito secuencial con biestables JK.
Comentarios	Todos los enlaces de “Ver más” de la página principal llevarán al sitio web de ayuda, solo que en diferentes secciones.

Tabla 34. Caso de uso: Ayuda circuito JK

Caso de uso 31: Ayuda circuito T	
Descripción	Acceder a la sección de ayuda acerca del funcionamiento de los circuitos secuenciales con biestables T.
Actor	Alumno
Relaciones	Acceder página Ayuda
Precondición	Estar logado con un perfil de alumno y en la página principal.
Secuencia	1. Pulsar sobre la opción “Ver más” de los circuitos secuenciales con biestables T.
Postcondición	Redireccionará al usuario a la página de ayuda y en concreto a la sección donde se comenta acerca de cómo resolver un circuito secuencial con biestables T.
Comentarios	Todos los enlaces de “Ver más” de la página principal llevarán al sitio web de ayuda, solo que en diferentes secciones.

Tabla 35. Caso de uso: Ayuda circuito T

Caso de uso 32: Ayuda circuito D	
Descripción	Acceder a la sección de ayuda acerca del funcionamiento de los circuitos secuenciales con biestables D.
Actor	Alumno
Relaciones	Acceder página Ayuda
Precondición	Estar logado con un perfil de alumno y en la página principal.
Secuencia	1. Pulsar sobre la opción “Ver más” de los circuitos secuenciales con biestables D.
Postcondición	Redireccionará al usuario a la página de ayuda y en concreto a la sección donde se comenta acerca de cómo resolver un circuito secuencial con biestables D.
Comentarios	Todos los enlaces de “Ver más” de la página principal llevarán al sitio web de ayuda, solo que en diferentes secciones.

Tabla 36. Caso de uso: Ayuda circuito D

Caso de uso 33: Dibujar nivel alto	
Descripción	Se dibuja un nivel alto en el cronograma de los ejercicios del entrenador.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar en el área del canvas en algunas de las salidas Q donde este dibujado un nivel bajo o ninguna señal.
Postcondición	Como resultado, deberá dibujarse un semiperiodo a nivel alto. Se ajustarán adecuadamente los flancos del semiperiodo.
Comentarios	Cuando se acceda a un ejercicio del entrenador de circuitos, el cronograma inicialmente no tendrá trazada ninguna señal. Se esperará a un evento causado por el clic del ratón para mostrar la señal de una salida. Cuando se pulse sobre un semiperiodo, este será el que cambie de estado “0” a “1”.

Tabla 37. Caso de uso: Dibujar nivel alto

Caso de uso 34: Dibujar nivel bajo	
Descripción	Se dibuja un nivel bajo en el cronograma de los ejercicios del entrenador.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar en el área del canvas en algunas de las salidas Q donde este dibujado un nivel alto.
Postcondición	Como resultado, deberá dibujarse un semiperiodo a nivel bajo. Se ajustarán adecuadamente los flancos del semiperiodo.
Comentarios	Cuando se pulse sobre un semiperiodo, este será el que cambie de estado “1” a “0” .

Tabla 38. Caso de uso: Dibujar nivel bajo

Caso de uso 35: Comprobar cronograma	
Descripción	Se comprueba la solución dibujada en el cronograma de un ejercicio.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar el botón “Comprobar”.
Postcondición	<p>Primer posible escenario:</p> <ul style="list-style-type: none"> • La solución dibujada en la gráfica es incorrecta, por tanto, se debe sumar un fallo y mostrarse al usuario visualmente mediante algún elemento en la interfaz gráfica, por ejemplo, mediante campos de texto. • La gráfica del cronograma se puede seguir modificando hasta encontrar la solución <p>Segundo posible escenario:</p> <ul style="list-style-type: none"> • La solución dibujada en la gráfica es correcta, por tanto, se debe sumar un acierto y mostrarse al usuario visualmente mediante algún elemento en la interfaz gráfica, por ejemplo, mediante campos de texto. <p>Tercer posible escenario:</p> <ul style="list-style-type: none"> • Si no se ha dibujado sobre ninguna salida, se deberá indicar al usuario de la situación y como solucionarla mediante un cuadro de dialogo.
Comentarios	Se podrá seguir dibujando y comprobando las veces que hagan falta.

Tabla 39. Caso de uso: Comprobar cronograma

Caso de uso 36: Mostrar solución cronograma	
Descripción	Se muestra la solución al cronograma de un ejercicio.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	<ol style="list-style-type: none"> 1. Pulsar sobre el botón “Mostrar Solución”. 2. Se muestra un mensaje de ánimo aleatorio y dos opciones “Seguir Intentando” y “Mostrar Solución”. 3. Dependiendo de la que se escoge pasará a un escenario u otro.
Postcondición	<p>Escenario “Seguir intentando” se obtiene:</p> <ul style="list-style-type: none"> • El cronograma mantendrá el estado actual. <p>Escenario “Mostrar Solución” se obtiene:</p> <ul style="list-style-type: none"> • El cronograma se alterará y se mostrará dibujada la solución del circuito sobre el cronograma.
Comentarios	En el caso de que previamente se haya dibujado la solución correcta y se pulsa sobre “Mostrar Solución”, el cronograma permanecerá inalterable.

Tabla 40. Caso de uso: Mostrar solución cronograma

Caso de uso 37: Nuevo ejercicio	
Descripción	Se genera un nuevo ejercicio con el circuito secuencial del biestable dado.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar sobre el botón “Nuevo ejercicio”.
Postcondición	Se recargará el cronograma y la o las entradas que pueda tener se volverán a generar de forma aleatoria.
Comentarios	Cada vez que se accederá a la página o se pulsa el botón se producirá una nueva entrada al azar.

Tabla 41. Caso de uso: Nuevo ejercicio

Caso de uso 38: Comprobar ecuación	
Descripción	Se comprueba que las ecuaciones de las entradas de los biestables escritas por el usuario coinciden con las ecuaciones correctas del ejercicio.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar el botón “Comprobar”.
Postcondición	<p>Primer escenario posible:</p> <ul style="list-style-type: none"> • Las ecuaciones introducidas no coinciden con las del circuito, por tanto, la solución será incorrecta y se notificará al usuario a través de un mensaje de error. <p>En el segundo escenario se obtiene:</p> <ul style="list-style-type: none"> • Las ecuaciones introducidas coinciden con las del circuito, por tanto, la solución será correcta y se notificará al usuario mediante un mensaje de felicitación.
Comentarios	<p>Cada ejercicio tendrá una imagen asociada donde el alumno podrá identificar visualmente los componentes y la unión de estos. En base a estos deberá encontrar la respuesta.</p> <p>La resolución de ecuaciones no alterará el estado del cronograma o de los contadores de aciertos y fallos.</p>

Tabla 42. Caso de uso: Comprobar ecuación

Caso de uso 39: Mostrar solución ecuación	
Descripción	Se muestra la solución con las ecuaciones de las entradas de los biestables del circuito.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar sobre el botón “Mostrar solución”.
Postcondición	Se mostrará en la pantalla mediante algún elemento gráfico como puede ser un campo de texto, la solución del circuito que se visualiza en la imagen del ejercicio.
Comentarios	

Tabla 43. Caso de uso: Mostrar solución ecuación

Caso de uso 40: Despejar campos ecuación	
Descripción	Borra los valores de los campos de escribir ecuaciones y mostrar la solución.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Pulsar sobre el botón “Clear”.
Postcondición	Se despejarán cualquier valor mostrado en el campo de ecuaciones o el área donde se muestra la solución.
Comentarios	El campo de ecuaciones deberá poder volver a modificarse.

Tabla 44. Caso de uso: Despejar campos ecuación

Caso de uso 41: Introducir ecuaciones	
Descripción	El alumno introduce las ecuaciones del circuito en un campo de texto.
Actor	Alumno
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	1. Escribir en un campo de texto.
Postcondición	
Comentarios	El alumno podrá modificar el contenido de este campo cuantas veces quiera.

Tabla 45. Caso de uso: Introducir ecuaciones

4.4.5. Funcionalidades del sistema

La Ilustración 12 muestra el diagrama de casos de uso del sistema y en la Tabla 46 y Tabla 47 se detallan cada una de ellas:

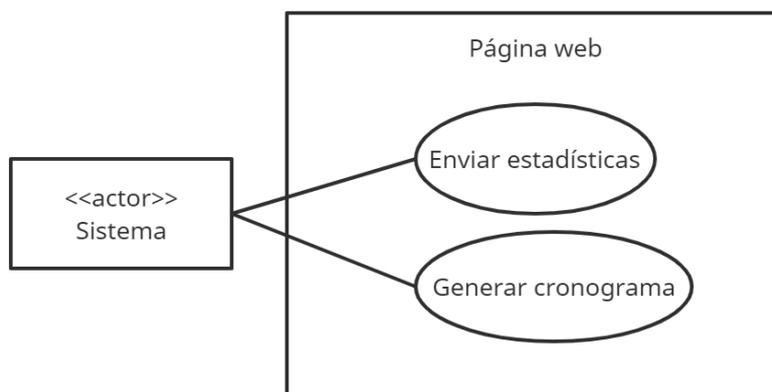


Ilustración 12. Diagrama de casos de uso para las funcionalidades del sistema.

Caso de uso 42: Enviar estadísticas	
Descripción	Funcionalidad que se encarga de almacenar los resultados de aciertos y fallos obtenidas por el usuario en los ejercicios.
Actor	Sistema
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio. Secuencia realizada previamente por el usuario: 1. Resolver el cronograma. 2. Pulsar el botón “Comprobar”.
Secuencia	-
Postcondición	Almacenará los datos obtenidos acerca de los aciertos y fallos en el historial de dicho alumno en la base de datos.
Comentarios	-

Tabla 46. Caso de uso: Enviar estadísticas

Caso de uso 43: Generar cronograma	
Descripción	Funcionalidad que se encarga de generar el cronograma de un ejercicio.
Actor	Sistema
Relaciones	-
Precondición	Estar logado con un perfil de alumno y en la página de algún ejercicio.
Secuencia	-
Postcondición	Se generará y se dibujará un cronograma con: <ul style="list-style-type: none"> • Una señal de reloj, cuadrada con semiperiodos al 50%, de X periodos de tiempo. • Una o varias señales de entrada dependiendo del tipo de biestable con X periodos de tiempo. Cambia en cada ejercicio de forma aleatoria. • El número de salidas indicadas con X periodos de tiempo.
Comentarios	-

Tabla 47. Caso de uso: Generar cronograma

4.4.6. Requisitos no funcionales

En este apartado se describen algunas cualidades y atributos del sistema por medio de ciertos protocolos y se establecen restricciones sobre las funciones y servicios que no se pueden clasificar como requisitos funcionales:

- El sistema estará protegido frente a accesos no autorizados.
- Se mantendrá la sesión de un usuario una vez logado en la página web.
- La aplicación web se desarrollará para un navegador web con acceso a internet y compatible con HTML 5.
- El sistema encriptará las contraseñas de los usuarios usando el algoritmo de *hashing* SHA512 [10].
- El sistema no registrará ninguna información personal salvo el nombre y el correo electrónico, respetando lo establecido por la LOPD.
- La aplicación web será implementada en HTML5 [11] simple, sin *framework* de desarrollo, PHP [12] para la comunicación con una base de datos MySQL, JavaScript [13] para programar las funcionalidades que se quieran aplicar a elementos o servicios y CSS [14] para mejorar el diseño gráfico y la presentación de la página web.
- Realizar una plantilla parametrizada para añadir ejercicios en el futuro, de forma clara y sencilla.
- Alojamiento en un servidor web donde se puedan ejecutar las tecnologías planteadas.
- Mensajes de error, indicadores y cuadros de dialogo que señalen al usuario si se ha equivocado o ha introducido algún valor erróneo en alguno de los campos disponibles.
- Los circuitos estarán compuestos como máximo de tres biestables, pudiendo combinar diferentes tipos de biestables.
- El cronograma debería tener al menos 8 periodos de tiempo.

5. Diseño

En esta sección se describe el modelo arquitectónico empleado en el desarrollo de la aplicación web y se indican los prototipos elaborados a raíz de los requisitos funcionales del sistema analizados en la fase anterior.

5.1. Arquitectura de tres capas

Se ha utilizado una arquitectura separada en tres capas [15] de modo que facilite el desacoplamiento entre los diferentes componentes. Se puede observar el esquema del modelo en la Ilustración 13 en las que se ha añadido las tecnologías utilizadas en cada capa.

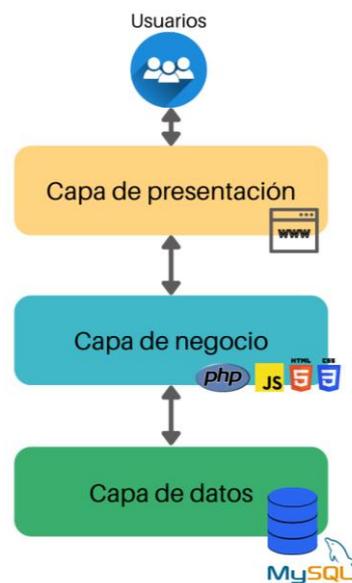


Ilustración 13. Esquema de la arquitectura en 3 capas.

La ventaja de emplear este diseño reside en la distribución en diferentes niveles, donde cada uno puede emplear una lógica diferente y ser desarrollada en sus respectivas tecnologías. Ante un cambio en el código o la estructura de datos de algún nivel, no será necesario adaptar la lógica de los demás niveles para soportar las modificaciones realizadas. Esto facilitará el mantenimiento y la escalabilidad de la aplicación web, además de contribuir a una solución más flexible y con posibilidad de reutilización en otros proyectos o en la evolución de este.

5.1.1. Capa de presentación

Este primer nivel consiste en mostrarle información al usuario final a través de las interfaces gráficas de la aplicación web y permite pasar las respuestas obtenidas durante la interacción con los diferentes elementos a la capa de negocio.

5.1.2. Capa de negocio

El segundo nivel gestiona la lógica de la aplicación y la comunicación entre los otros dos niveles. Esta capa se encarga de recibir y procesar la acción realizada por el usuario, gestionarla internamente ejecutando el servicio correspondiente y comunicarse con la capa de datos para transmitir, almacenar o recuperar información. Aquí se encuentran las funciones encargadas de resolver el cronograma, las ecuaciones y las estadísticas de los ejercicios del entrenador de circuitos secuenciales.

Adicionalmente, se han desarrollado funciones para mejorar aspectos de calidad y usabilidad del software, tales como la verificación y validación de información introducida por el usuario, así como aspectos relacionados con el diseño de las interfaces gráficas.

5.1.3. Capa de datos

Este último nivel se ocupa del almacenar o devolver los datos requeridos por la capa de negocio. Esta capa dispone de una conexión con una base de datos relacional que se dispone a guardar información relacionada con los usuarios y las estadísticas de los diferentes tipos de ejercicios, además de retornar información en forma de consultas. En el desarrollo de la aplicación el sistema de gestión de base de datos empleado ha sido MySQL.

A continuación, se presenta un diagrama UML con el esquema planteado para la base de datos relacional con sus componentes y relaciones (Ilustración 14)

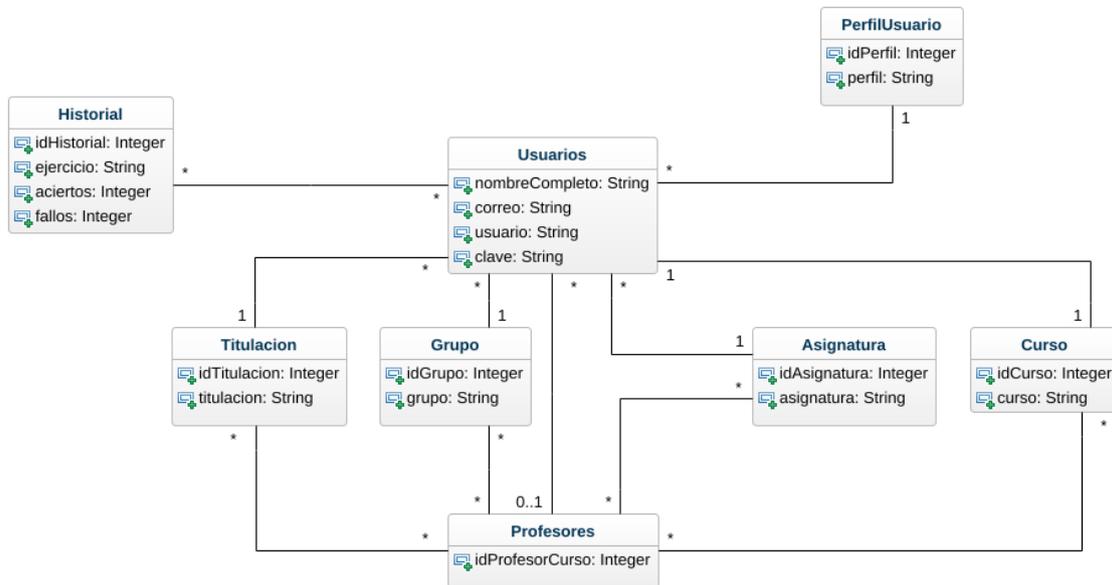


Ilustración 14. Diagrama estructural de la base de datos.

El esquema muestra los conceptos analizados anteriormente en el modelo de dominio durante la fase de análisis y especificación de requisitos, ajustados de forma práctica a la solución. Se han retratado los diferentes componentes mediante clases con atributos, cada una con su clave primaria asociada y las relaciones se han representado mediante una línea con una numeración que indica las restricciones entre las tablas.

Se describe la funcionalidad de cada uno de los componentes a continuación:

- **Usuarios:** almacenará la información relativa a los usuarios del sistema. Como norma común todos los usuarios deben registrarse con los atributos nombre completo, correo electrónico, nombre de usuario y contraseña.
- **PerfilUsuario:** se trata de una componente utilizada como clase enumeración, que describe los tipos de usuarios que se pueden dar en la aplicación web: administrador, profesor, alumno.
- **Historial:** esta clase se encarga de almacenar para cada alumno los resultados de aciertos y fallos de los diferentes ejercicios.
- **Profesores:** guardará información acerca de los diferentes profesores y los cursos, titulaciones, asignaturas o grupos a los que pueden asociarse.
- **Titulación:** representa una clase enumeración, que describe las titulaciones que se requieran, por ejemplo, una posible titulación podría ser Grado en Ingeniería Informática o empleando sus siglas, GII.
- **Grupo:** conserva la información con respecto a los tipos de grupos, como puede ser A, E o F. También es utilizada como una clase enumerada.
- **Asignatura:** es una clase enumeración que guarda los tipos de asignaturas. Por ejemplo, el caso de las asignaturas de Fundamentos de computadores y Tecnología de computadores, o por sus siglas conocidas en la universidad como FCO y TCO.
- **Curso:** se almacenan los datos acerca de las fechas de los cursos escolares, al igual que en las demás clases enumeradas, como un tipo en el formato que se decida. Por ejemplo, se podría dar “2020-2021” o “20/21”.

5.2. Mockups

Para facilitar la interacción del usuario con la aplicación web, se han diseñado prototipos de las diferentes funcionalidades identificadas aplicando algunos de los principios de usabilidad web: disponibilidad, mostrar la información clave y estructurar el contenido de forma sencilla, compacta e intuitiva para que el usuario no se sienta confundido en ningún momento.

De este modo, se ha optado por diferenciar de forma común a todas las páginas web de la aplicación, las siguientes zonas:

- **Menú superior:** se muestra el logo de la universidad y las acciones que puedan efectuar los usuarios autenticados, tales como el cierre de sesión o navegabilidad entre las diferentes páginas.
- **Zona central:** En ella se visualiza el contenido de los ejercicios, la ayuda, las configuraciones o consultas que se pueden realizar.
- **Footer:** Esta zona presenta los créditos y referencias mediante enlaces a la universidad y al TFG del entrenador de biestables en el que se encuentra basado.

A continuación, se desglosa las diferentes partes y se muestra el diseño de los prototipos para la aplicación web:

5.2.1. Página de acceso

La Ilustración 15 muestra el boceto de la página web donde se realizará el registro o inicio de sesión para acceder al entrenador.

Google Chrome

https://

Icono UPV

Iniciar sesión

Usuario o Correo

Contraseña

Entrar

Registrar alumno

Nombre complet

Correo

Titulación

Asignatura

Curso

Grupo

Usuario

Contraseña

Registrar

Ilustración 15. Prototipo página de acceso.

5.2.2. Página principal

Este apartado distingue las páginas de inicio de los diferentes perfiles de usuario. Dependiendo del perfil con el que se acceda en la página de acceso, se redirigirá al usuario a una interfaz u otra.

Administrador: En la Ilustración 16 e Ilustración 17 se muestran las maquetas con las diferentes funcionalidades correspondientes al perfil de administrador.

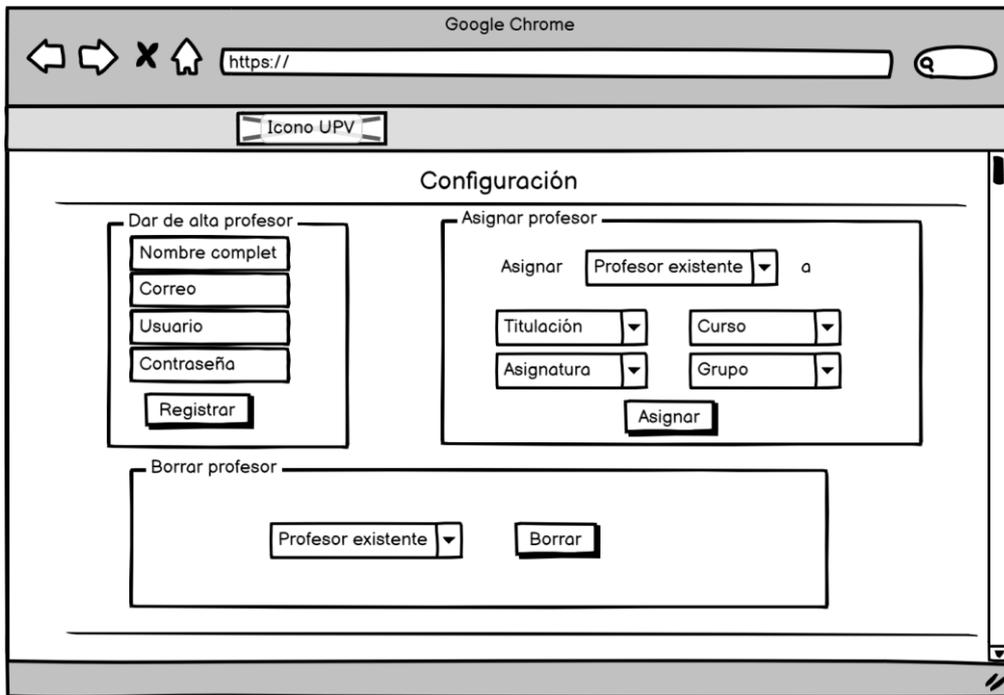


Ilustración 16. Prototipo página principal del administrador – configuración datos de profesores.

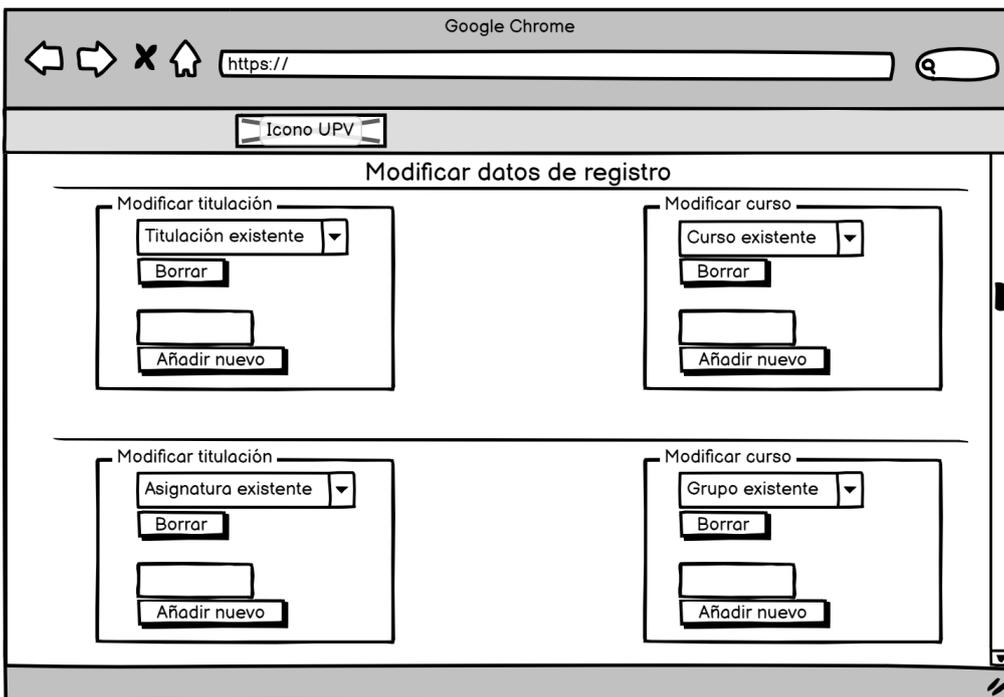


Ilustración 17. Prototipo página principal del administrador – configuración datos de registro.

Profesor: Las funcionalidades del profesor se pueden visualizar en el boceto de la Ilustración 18.

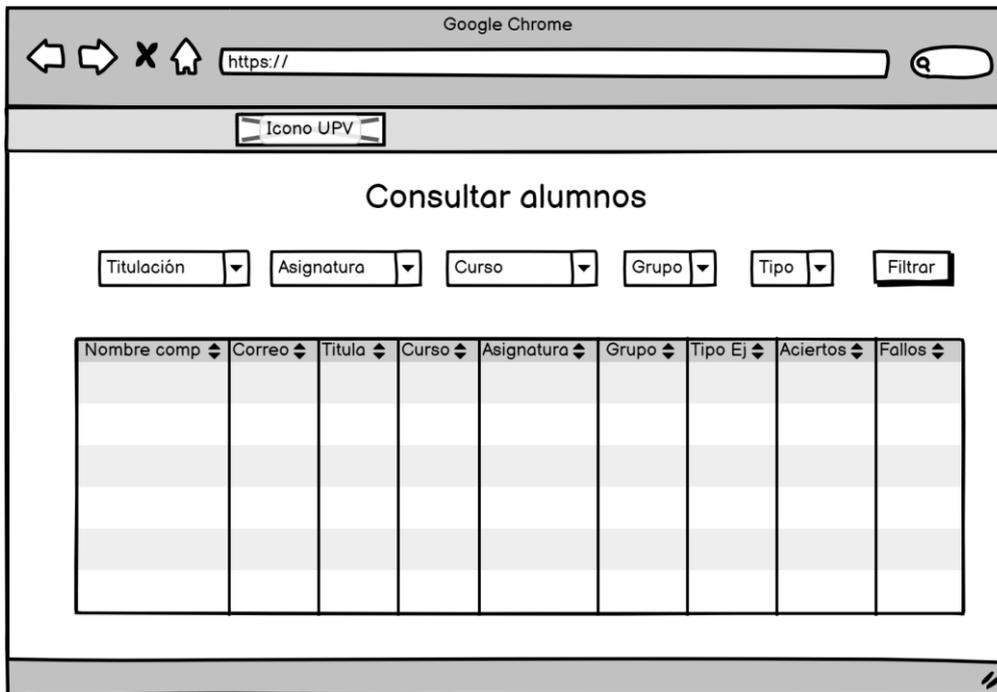


Ilustración 18. Prototipo página principal del profesor.

Alumno: En la Ilustración 19 se muestra la página de inicio del alumno con acceso a los diferentes recursos y actividades disponibles.

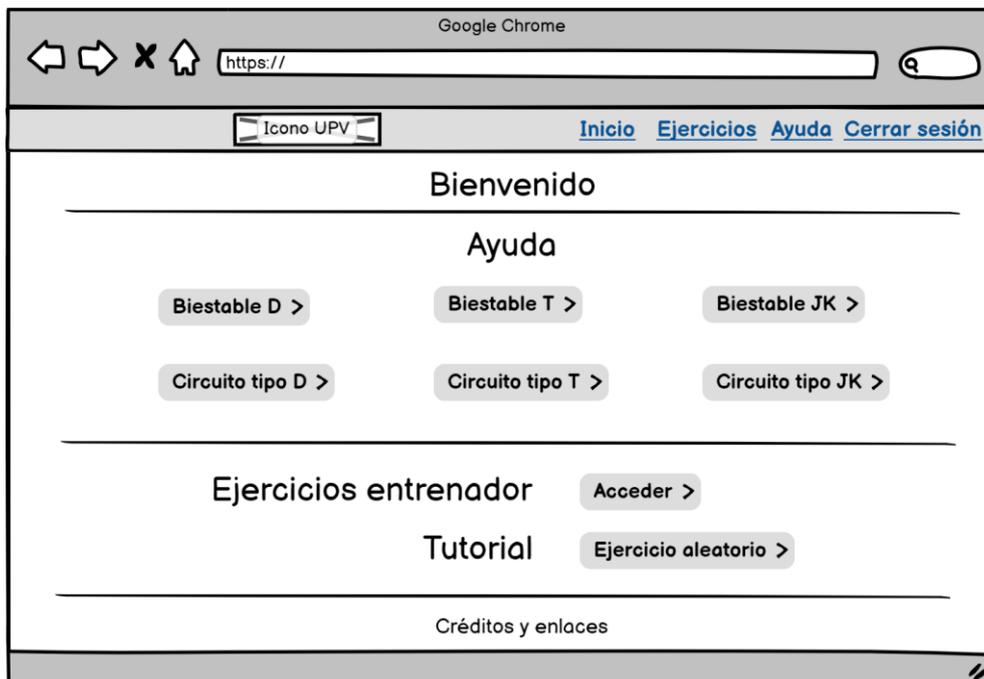


Ilustración 19. Prototipo página principal del alumno.

5.2.3. Página de ejercicios

Esta sección es accesible a través de la página principal del alumno. Por un lado, se muestra una lista de los ejercicios implementados, como se puede ver en la maqueta de la Ilustración 20 y cada uno de estos ejercicios tienen destinada su página web de forma exclusiva como se puede observar en los bocetos ideados en la Ilustración 21 e Ilustración 22.

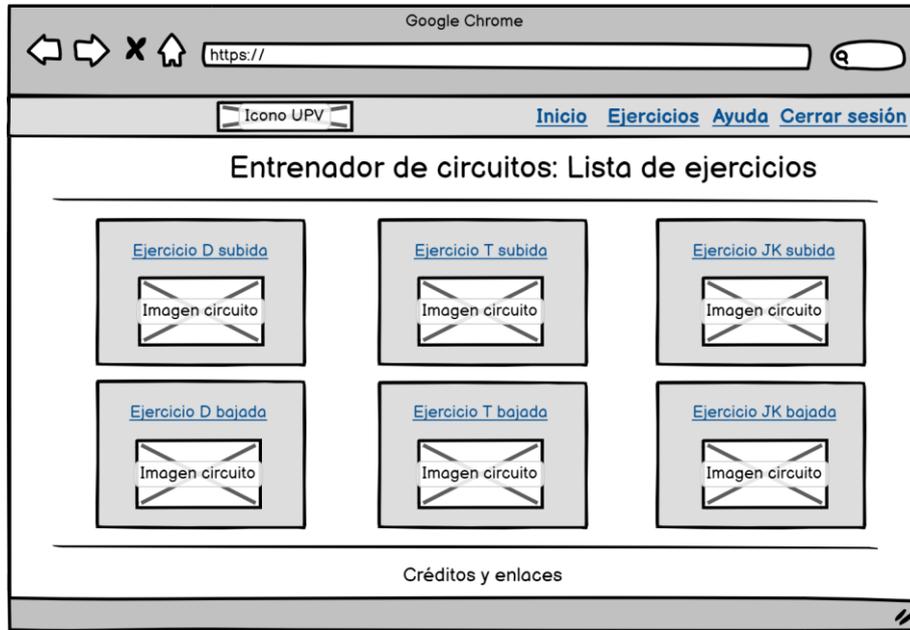


Ilustración 20. Prototipo página de ejercicios.

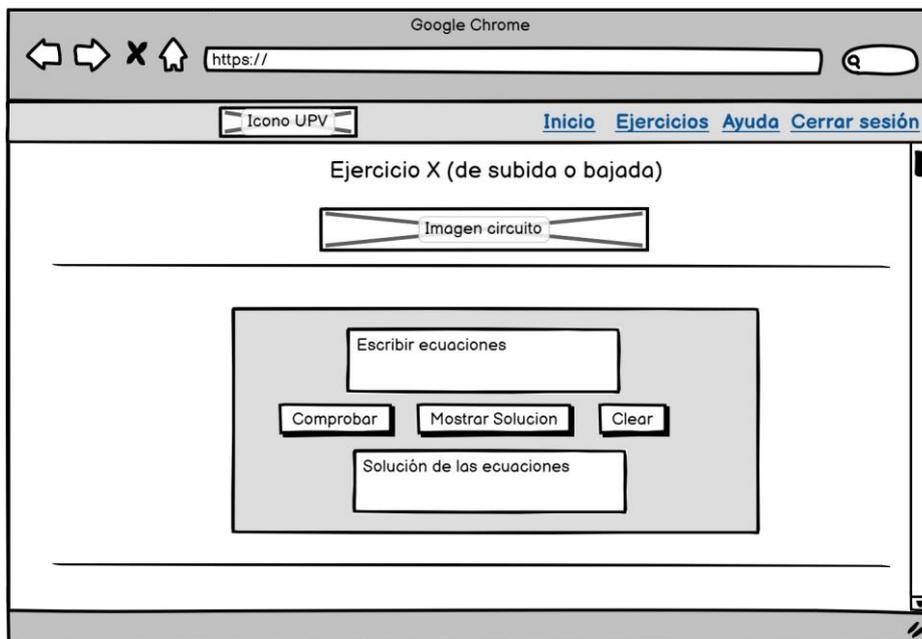


Ilustración 21. Prototipo página de un ejercicio X – Circuito y ecuaciones.

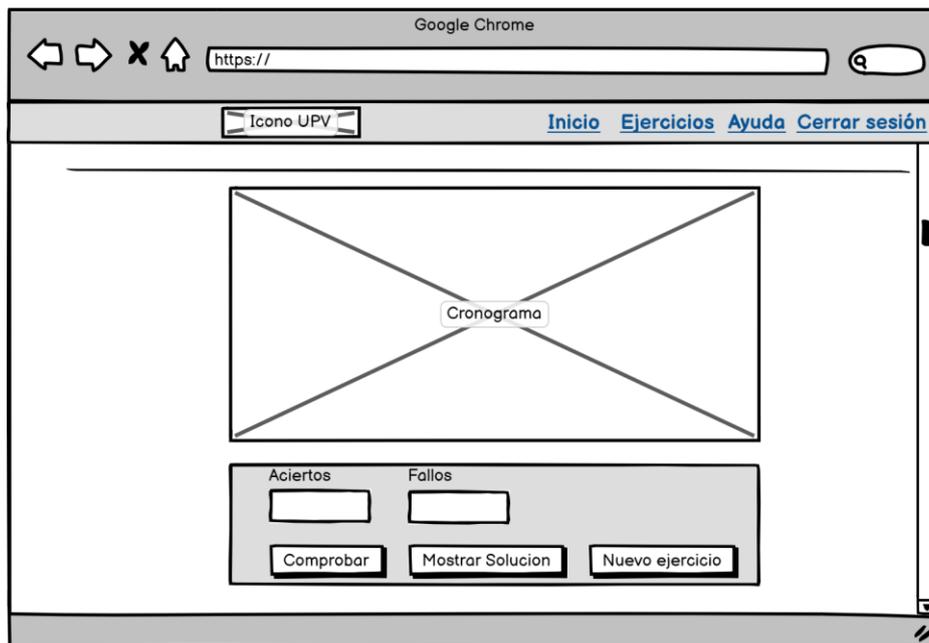


Ilustración 22. Prototipo página de un ejercicio X – Cronograma.

5.2.4. Página de ayuda

En la maqueta de la Ilustración 23 se puede observar la sección que sirve de ayuda para resolver las dudas del alumno.

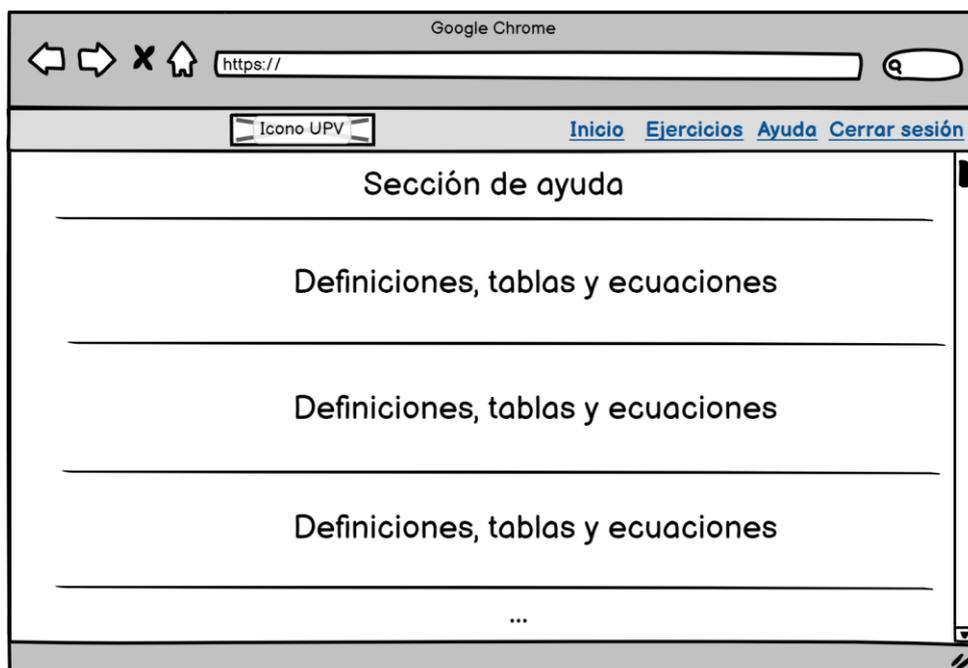


Ilustración 23. Prototipo página de ayuda.

6. Tecnologías y herramientas

A continuación, se describen las diferentes tecnologías y herramientas utilizadas para llevar a cabo la aplicación del entrenador.

6.1. JavaScript

Es un lenguaje de programación o de secuencia de comandos comúnmente empleado para el desarrollo de funciones complejas en páginas web, que permiten crear y animar contenido o de forma general, controlar o actualizar la información. Es uno de los lenguajes más habituales para desarrollar páginas web dinámicas con ejecución en el lado del cliente. La Ilustración 24 muestra el logotipo de este lenguaje.

De este modo, se han implementado varios scripts para la gestión de usuarios, la generación del cronograma con sus señales de entrada y salida, así como la lógica y resolución detrás de cada ejercicio propuesto por el entrenador.

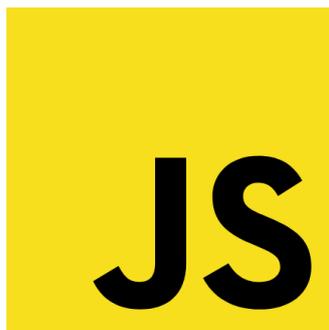


Ilustración 24. Logotipo de JavaScript.

6.2. PHP

Es un lenguaje de programación de código abierto que permite integración con código HTML, aunque se ejecuta en el lado del servidor, enviando al cliente el código HTML resultante de su ejecución. Se utiliza fundamentalmente para el desarrollo web y las aplicaciones web dinámicas (Ilustración 25).

En este caso, se ha empleado para las operaciones SQL de consultas, inserciones y borrado de la base de datos. Al mismo tiempo se ha aprovechado para aumentar la consistencia y seguridad de la aplicación web, ayudando a registrar y mantener las sesiones de usuarios activas.



Ilustración 25. Logotipo de PHP.

6.3. HTML

Se trata de un lenguaje de marcado cuyo propósito es utilizar enlaces e hipertextos para estructurar y desplegar los contenidos de una página web en los servidores web.

Se ha empleado la versión HTML5, para organizar y mostrar el contenido de los formularios, imágenes, botones, el canvas y las diferentes secciones de la aplicación. En la Ilustración 26 se puede ver su logotipo.

Además, las funcionalidades especificadas en etapas anteriores se han podido conectar e implementar correctamente con las demás tecnologías gracias al uso de etiquetas como son `<script>`, `<?php?>`, `<link>`.



Ilustración 26. Logotipo de HTML.

6.4. CSS

Es un lenguaje de diseño gráfico, que ayuda definir y estilizar la presentación y los elementos de un documento web escrito en un lenguaje de marcado como HTML.

Para la aplicación del entrenador se ha utilizado CSS3 (Ilustración 27), con el fin de proporcionar un modelo de diseño a la página web. Se han indicado diferentes fuentes

de colores y efectos para los campos de formularios, los textos y botones, así como en la separación y distinción entre diferentes secciones del documento. Cabe destacar además, la combinación junto a código JavaScript para realizar la página de acceso al entrenador donde se muestra de forma dinámica el formulario de registro o de inicio de sesión.



Ilustración 27. Logotipo de CSS.

6.5. Visual Studio Code

Se trata de una herramienta para editar código fuente desarrollada por Microsoft (Ilustración 28). Es compatible con varios lenguajes de programación y da soporte para operaciones tales como la depuración, fragmentación o refactorización de código, control integrado con herramientas de gestión de versiones (por ejemplo, Git), entre otros [16].

Se ha empleado como entorno de desarrollo para visualizar las carpetas y editar el código de los diferentes archivos que componen el proyecto.



Ilustración 28. Logotipo de Visual Studio Code.

6.6. WAMP

Es una herramienta que viene como una solución integrada para el desarrollo y despliegue de aplicaciones web a nivel local [17]. Al igual que sus siglas indican, es un entorno de desarrollo para el sistema operativo **Windows**, constituido por un servidor web **Apache**, un gestor de base de datos **MySQL** y un lenguaje de programación, generalmente **PHP**.

Se ha empleado para el alojamiento y las pruebas iniciales de la aplicación, así como para la construcción y administración de la base de datos. Su logotipo se muestra en la Ilustración 29.



Ilustración 29. Logotipo de WAMP.

6.7. Herramientas de diseño

Canva [18]: Es una herramienta web para el diseño gráfico y composición de imágenes. Se ha empleado para el desarrollo de figuras de la metodología, arquitectura y algunos de los logos indicados en este documento. La Ilustración 30 muestra su logotipo.



Ilustración 30. Logotipo de Canva.

GenMyModel [19]: Es una plataforma online para el modelado de clases que ha servido para diseñar el diagrama de la base de datos (Ilustración 31).



Ilustración 31. Logotipo de GenMyModel.

Creately [20]: es una herramienta para la elaboración de diagramas galardonada por su facilidad de uso, junto a sus diversas características y posibilidades de colaboración en equipo. Se ha empleado para el diseño de los diagramas de casos de uso, de contexto, de bloques y el modelo de dominio. Se puede observar su logotipo en la Ilustración 32.



Ilustración 32. Logotipo de Creately.

Balsamiq [21]: Es una herramienta de maquetado que permite crear prototipos y bocetos para diferentes tipos de sistemas. Se ha utilizado esta herramienta para diseñar los mockups de la aplicación del entrenador. Se muestra su logotipo en la Ilustración 33.



Ilustración 33. Logotipo de Balsamiq.

7. Implementación

En esta fase de desarrollo se describe el resultado de la solución llevada a cabo para este TFG, empleando el diagrama de bloques y mostrando la estructura del proyecto usada para la implementación.

7.1. Diagrama de bloques

Representa un diagrama de flujo del funcionamiento interno de la aplicación web del entrenador, donde se distinguen mediante bloques y relaciones la organización de las interfaces del sistema.

Como se puede observar en la Ilustración 34, las flechas representan la navegabilidad entre los diferentes bloques. Según el sentido de la flecha, se puede indicar la accesibilidad a las diferentes interfaces desde dos perspectivas:

- **Flecha de único sentido:** cómo se puede observar en el caso de los bloques de *ejercicios*, estos garantizan disponibilidad al bloque de la *página de ayuda*, sin embargo, no de una forma bidireccional. En consecuencia, si se desea acceder a algún ejercicio de modo exclusivo, se tendría que acceder a través del bloque *Página lista de ejercicios*.
- **Flecha de doble sentido:** esta relación indica accesibilidad por parte de ambos bloques conectados. De este modo, al iniciar sesión empleando una cuenta de alumno, a través del bloque con la *Página de acceso*, llevará a la *Página principal del alumno* y al cerrar la sesión de dicho usuario, se asegura el retorno al bloque anterior marcado por la flecha.

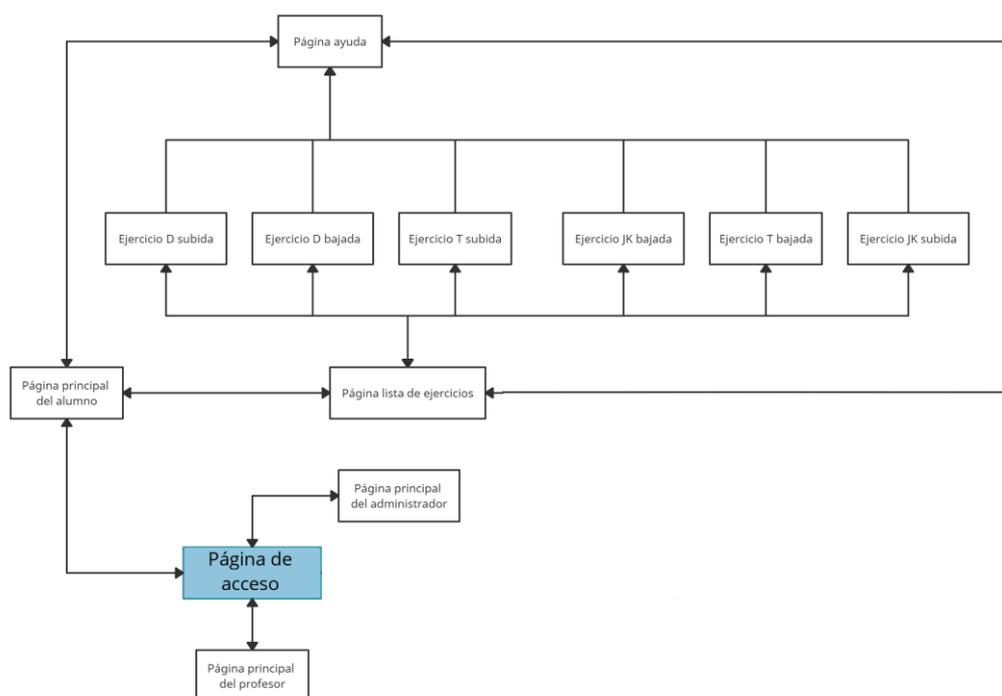


Ilustración 34. Diagrama de bloques.

7.2. Estructura del proyecto

En este subapartado se describe la estructura del proyecto y los diferentes componentes. Como se puede distinguir en la Ilustración 35, se ha optado por clasificar en diferentes carpetas los documentos según un propósito concreto.

Los siguientes documentos PHP no adjuntados a una categoría se han empleado para desarrollar interfaces gráficas de la siguiente forma:

- **index.php**: este archivo se ha implementado para la página web de acceso, donde se encuentran los formularios de inicio de sesión y registro de usuarios.
- **ayuda.php**: esta es la página de orientación y soporte para el usuario alumno donde se describe acerca de los conceptos básicos y la resolución de ejercicios.
- **alumnoPage.php**, **administradorPage.php** y **profesorPage.php**: estos documentos representan las páginas principales de cada perfil de usuario donde se muestran sus respectivas funcionalidades o acceso a ellas.

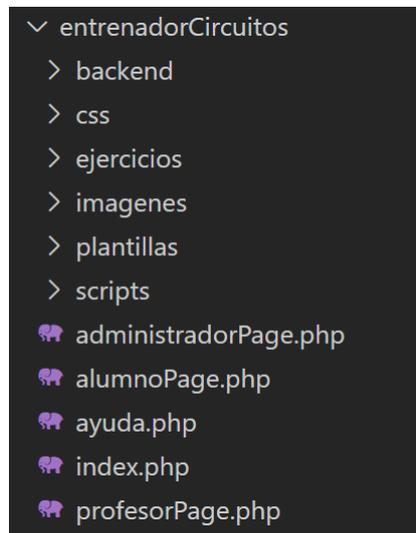


Ilustración 35. Estructura global del proyecto.

A continuación, se desglosa las diferentes carpetas justificando la distinción de su contenido.

7.2.1. Backend

Los archivos de esta carpeta se han enfocado en recoger la información ejecutada por el usuario y transmitirla a la base de datos mediante operaciones SQL. En la Ilustración 36 se pueden ver los archivos implementados.

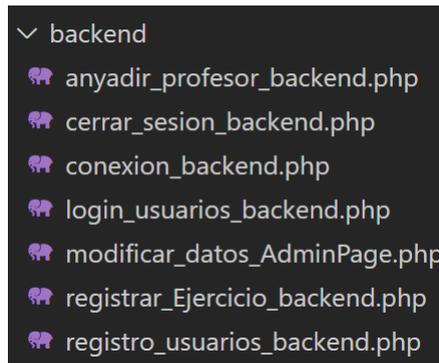


Ilustración 36. Contenido de la carpeta Backend.

- **registro_usuarios_backend.php**: se describe la operación de inserción y comprobaciones del formulario de registro de usuarios para un perfil de alumno.
- **login_usuario_backend.php** y **cerrar_sesion_backend.php**: estos archivos se encargan de comprobar y gestionar la sesión de un usuario autenticado.
- **registrar_Ejercicio_backend.php**: aquí se realizan las inserciones y actualizaciones en la base de datos de los aciertos y fallos en los diversos tipos de ejercicios resueltos por un alumno.
- **conexión_backend.php**: este archivo se ha implementado con un propósito de uso general para establecer la conexión con la base de datos en los diferentes puntos de la aplicación donde se desea realizar alguna operación, ya sea de consulta, inserción, actualización o borrado.
- **anyadir_profesor_backend.php** y **modificar_datos_AdminPage.php**: se ha optado por diferenciar las funcionalidades del administrador entre la creación de nuevos usuarios con el perfil profesor y la modificación de los diferentes datos como son las asignaturas, cursos, titulaciones o grupos y el vaciado de la base de datos.

7.2.2. CSS

Como se puede observar en la Ilustración 37, se ha empleado un solo archivo donde se ha especificado el estilo y los formatos empleados en la solución del entrenador, empleando el lenguaje de diseño CSS. Esta hoja de estilo se encuentra implementada en todos los archivos que representan las interfaces gráficas a la que el usuario tiene acceso.

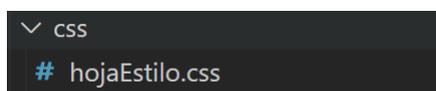


Ilustración 37. Contenido de la carpeta css.

7.2.3. Ejercicios

Esta carpeta contiene las interfaces gráficas de la lista de ejercicios que pueden ser accedidas por un usuario, además de un archivo diferente por cada tipo de ejercicio del entrenador indicando, el biestable y el flanco (Ilustración 38).

Estas páginas muestran al alumno el cronograma y las ecuaciones a resolver del ejercicio. El archivo **ejercicioEjemploTSubida.php** se ha utilizado como un

ejercicio básico en forma de tutorial para que el alumno se acostumbre con los controles y el formato de las ecuaciones.

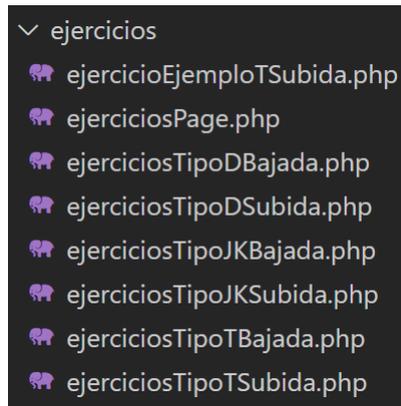


Ilustración 38. Contenido de la carpeta ejercicios.

7.2.4. Scripts

Esta carpeta abarca los archivos desarrollados en JavaScript (Ilustración 39).

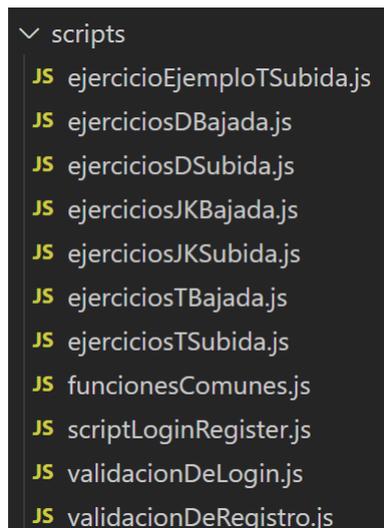


Ilustración 39. Contenido de la carpeta scripts.

Aquí se pueden distinguir archivos enfocados en diferentes operaciones acerca de:

- La lógica de los ejercicios del entrenador, donde cada archivo tiene funciones parametrizadas que se pueden ajustar individualmente para cada circuito: **ejercicioEjemploTSubida.js**, **ejerciciosDBajada.js**, **ejerciciosDSubida.js**, **ejerciciosTBajada.js**, **ejerciciosTSubida.js**, **ejerciciosJBajada.js**, **ejerciciosJSubida.js**.
- El archivo **funcionesComunes.js** que se emplea para métodos generales aplicados a todos los ejercicios, como es el caso de la generación del cronograma, las señales de entrada y salida, la comprobación de ecuaciones, entre otros.
- Se han creado dos archivos para la validación de los diferentes campos de los formularios, tales como restringir la entrada de valores numéricos en el campo de

nombre o indicar una mínima longitud en los campos de texto de nombre de usuario y contraseña.

- Para los formularios de login y registro de usuarios se ha implementado un uso dinámico, como se puede ver en la Ilustración 40 e Ilustración 41. Al pulsar sobre el botón “Registrarse” o “Iniciar sesión”, se muestra el formulario correspondiente a la funcionalidad.

Iniciar sesion

Nombre usuario o correo electronico

Contraseña

Entrar

¿Eres alumno y aun no tienes una cuenta?

Regístrate para tener acceso al contenido

Regístrate

Ilustración 40. Interfaz gráfica de la página de acceso – formulario login.

Registrarse

Nombre completo

Correo electronico

Titulacion

Grupo

Asignatura

Curso

Nombre usuario

Contraseña

Regístrate

¿Ya tienes una cuenta?

Inicia sesión y accede al contenido

Iniciar sesion

Ilustración 41. Interfaz gráfica de la página de acceso – formulario registro.

7.2.5. Imágenes

Esta carpeta almacena las imágenes empleadas en el entrenador (Ilustración 42). La mayoría son esquemas de circuitos de los diferentes ejercicios, pero también se cuenta con la presencia de los logos de la universidad (UPV) y la escuela técnica superior de informática (ETSINF).

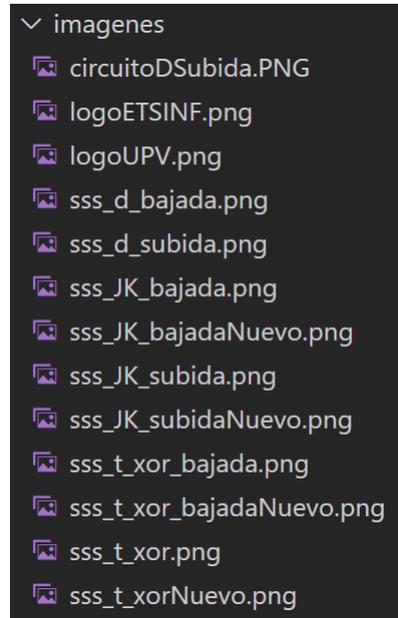


Ilustración 42. Contenido de la carpeta imágenes.

7.2.6. Plantillas

Esta carpeta (Ilustración 43) contiene los archivos necesarios para añadir nuevos ejercicios al entrenador de una forma sencilla siguiendo una serie de pasos que se describen a continuación:

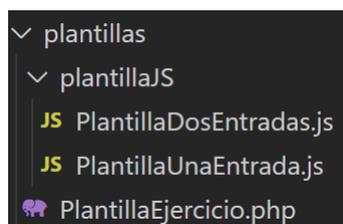


Ilustración 43. Contenido de la carpeta plantillas.

PASO 1: Crear una copia del archivo *PlantillaEjercicio.php* en la carpeta *ejercicios* y asignarle un nombre reconocible sobre el tipo de ejercicio que se desea desarrollar.

PASO 2: En la carpeta imágenes añadir el dibujo del circuito que se desee usar.

PASO 3: Modificar los cuadros enmarcados en rojo de la Ilustración 44 con la descripción conveniente y añadir el nombre de la imagen subida en el paso 2.

```

<section class="seccionTextoEjercicios">
  <div class="contenedorTexto">
    <h2>Circuito secuencial con biestable  ">
    </div>
  </div>
</section>

```

Ilustración 44. Archivo *PlantillaEjercicio.php* – descripción del ejercicio.

PASO 4: En función del número de señales de entrada del circuito previsto, se propone una plantilla en JavaScript u otra. Se utilizará el archivo *PlantillaUnaEntrada.js* en caso de querer implementar un ejercicio con solo una señal de entrada o *PlantillaDosEntradas.js*, si se requieren dos señales de entrada.

Una vez decidida la plantilla, se crea una copia en la carpeta de *scripts* y se renombra para el ejercicio esperado.

PASO 5: En la plantilla PHP del ejercicio se indica en el espacio encuadrado en rojo en la Ilustración 45 el nombre del archivo creado en el paso 4.

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Entrenador secuencial</title>
  <link rel="stylesheet" href="../css/HojaEstilo.css">
  <script type="text/javascript" src="../scripts/funcionesComunes.js"></script>
  <script type="text/javascript" src="../scripts/  "></script>
</head>

```

Ilustración 45. Archivo *PlantillaEjercicio.php* – inserción de la plantilla en JavaScript.

PASO 6: En la plantilla del archivo JavaScript, se añade la ecuación del circuito elegido con la notación oportuna (Ilustración 46). En caso de querer cambiar la nomenclatura de las ecuaciones, se podrá cambiar la descripción que indica al usuario la terminología que tiene que utilizar del archivo php del paso 1 (Ilustración 47).

```
/** Ecuacion del circuito que se desea implementar*/
var solucionEcuacion = "  ";
```

Ilustración 46. Archivo JavaScript – inserción de ecuación del circuito.

```
<div class="contenedorEcuacionesCircuito">
  <h2>Usa la siguiente notación: </h2>
  <p>&nbsp;</p>
  <p>/(Negacion) * (AND) + (OR) ^ (XOR). </p>
  <p>Separar ecuaciones usando ;</p>
```

Ilustración 47. Archivo PlantillaEjercicio.php – descripción de la nomenclatura de las ecuaciones.

PASO 7: Siguiendo con la plantilla JavaScript, en el apartado de la función `solucionar()` se tiene que implementar la lógica del circuito. Se pueden observar los cuadros rojos en la Ilustración 48 que indican cada uno el flanco de subida y bajada de la señal de reloj. En base a eso, se emplean los vectores solución declarados en la plantilla para construir las ecuaciones del circuito.

```
/** Metodo donde se indica la solucion del problema en base a la ecuacion planteada.
 * Modificar según el caso.
 * ---Si es por flanco de subida dejar comentado caso i=0
 * ---Si es por flanco descomentar caso i=0
 */
function solucionar(iteraciones) {
  for (var i = 0; i <= iteraciones; i++) {
    /* Primera posicion siempre es 0 - Condicion impuesta por vectorEntrada
    if (i == 0) {
      vectorSolucion1[i] = 0;
      vectorSolucion2[i] = 0;
      vectorSolucion3[i] = 0;
    } else
    */
    if (vectorReloj[i] == 0) { //Flanco de bajada
      
    } else if (vectorReloj[i] == 1) { //Flanco de subida
      
    }
  }
}
```

Ilustración 48. Archivo JavaScript – función solución del cronograma.

A continuación, se demuestra mediante dos ejemplos como se realiza la implementación. Los elementos de los vectores VectorSolucionX (X es el número de biestable, de izquierda a derecha en el circuito, empezando en 1) y VectorRelej representan el estado lógico (alto o bajo) de los semiperiodos de las señales. El estado inicial de los biestables es siempre cero.

- En la Ilustración 49 se muestra la implementación de un circuito con biestables D por flanco de bajada, con una sola entrada, tal como indican las siguientes ecuaciones:

$$\begin{aligned} D0 &= E; \\ D1 &= Q0; \\ D2 &= Q0 * Q1 \end{aligned}$$

```
function solucionar(iteraciones) {
    for (var i = 0; i <= iteraciones; i++) {
        if (i == 0) { //Primera posicion siempre es 0 - Condicion impuesta por vectorEntrada
            vectorSolucion1[i] = 0;
            vectorSolucion2[i] = 0;
            vectorSolucion3[i] = 0;
        } else if (vectorRelej[i] == 0) { //Flanco de bajada
            vectorSolucion1[i] = vectorEntrada[i-1];
            vectorSolucion2[i] = vectorSolucion1[i-1];
            vectorSolucion3[i] = (vectorSolucion2[i-1] * vectorSolucion1[i-1]);
        } else if (vectorRelej[i] == 1) { //Flanco de subida
            vectorSolucion1[i] = vectorSolucion1[i-1];
            vectorSolucion2[i] = vectorSolucion2[i-1];
            vectorSolucion3[i] = vectorSolucion3[i-1];
        }
    }
}
```

Ilustración 49. Ejemplo implementación de la solución de un circuito con biestables tipo D activo por flanco de bajada.

- La Ilustración 50 presenta un circuito con biestables de tipo JK por flanco de subida, donde emplean dos entradas y cuyas ecuaciones se muestran a continuación:

$$\begin{aligned} J0 &= E; \quad K0 = E; \\ J1 &= Q0; \quad K1 = /Q0; \\ J2 &= Q0 + Q1; \quad K2 = /Q0 + /Q1 \end{aligned}$$

```

function solucionar(iteraciones) {
    for (var i = 0; i <= iteraciones; i++) {
        /* Primera posición siempre es 0 - Condición impuesta por vectorEntrada
        if (i == 0) {
            vectorSolucion1[i] = 0;
            vectorSolucion2[i] = 0;
            vectorSolucion3[i] = 0;
        } else
        */
        /*
        if (vectorReloj[i] == 0) { //Flanco de bajada

            vectorSolucion1[i] = vectorSolucion1[i - 1];
            vectorSolucion2[i] = vectorSolucion2[i - 1];
            vectorSolucion3[i] = vectorSolucion3[i - 1];

        } else if (vectorReloj[i] == 1) { //Flanco de subida
            vectorSolucion1[i] = operadorJK(vectorEntradaJ[i], vectorEntradaK[i], vectorSolucion1[i - 1]);
            vectorSolucion2[i] = operadorJK(vectorSolucion1[i], negar(vectorSolucion1[i]), vectorSolucion2[i - 1]);
            var sumaJ2 = vectorSolucion1[i - 1] | vectorSolucion2[i - 1];
            var sumaK2 = (vectorSolucion1[i - 1] ^ 1) | (vectorSolucion2[i - 1] ^ 1);
            vectorSolucion3[i] = operadorJK(sumaJ2, sumaK2, vectorSolucion3[i - 1]);
        }
    }
}

```

Ilustración 50. Ejemplo implementación de la solución de un circuito con biestables tipo JK activo por flanco de subida.

PASO 8: Por último, se añade la referencia al nuevo ejercicio en la lista de ejercicios del archivo *ejerciciosPage.php* (ver Ilustración 38), tal como se muestra en la Ilustración 51.

Se enlazarán los archivos del paso 1 y del paso 2 para mostrar visualmente al alumno el tipo de ejercicio que va a resolver.

```

<div class="ejercicio">
  <a onclick="window.location.href='PlantillaEjercicio.php'"> Nombre del ejercicio </a>
  <p>(flanco de )</p>
  
</div>

```

Ilustración 51. Archivo *ejerciciosPage.php* – enlace a la plantilla php de un ejercicio.

Inicialmente la solución de este entrenador se plantea para ejercicios con 3 señales de salida y máximo dos entradas. Para añadir más salidas y entradas se recomienda revisar el tamaño del canvas y las coordenadas de las funciones de dibujo y de la generación del cronograma.

7.3. Refactorización

La refactorización es un proceso de transformación de un software que conserva su comportamiento original, modificando solamente la estructura interna. Esta etapa se ha llevado a cabo tras finalizar la implementación de la lógica del entrenador.

Para mejorar la legibilidad y facilitar el mantenimiento de la aplicación web, se han aplicado algunas técnicas de *refactoring* [22].

El propósito principal ha sido identificar y eliminar los posibles *codes smells* (o defectos de código) que podrían obstaculizar la integración de nuevos componentes, tales como estructuras de código largas y complejas o redundancia en la implementación del código.

A continuación, se comenta más en detalle las acciones realizadas, en particular, sobre el código desarrollado en JavaScript:

- Segmentación de métodos largos y complejos, tal como es la función de dibujar en el cronograma del canvas.
- Creación de métodos con propósito específico, por ejemplo, en la identificación de las salidas que son pulsadas en el cronograma por el usuario, la comprobación de estadísticas o en la generación del cronograma.
- Declaración de variables, nombres de clases y métodos descriptivos, de modo que pueda entenderse fácilmente la finalidad de cada una. De forma adicional, se han añadido comentarios en diferentes secciones de las clases para que agilice el proceso de comprensión del código.
- Combinación de métodos, donde se ha extraído código repetitivo y se ha unificado para crear una sola función. Aquí se encuentra por ejemplo el caso de la comprobación de ecuaciones de un circuito.
- Sustitución de algoritmo. En este apartado se han modificado y reestructurado los métodos de comprobar y solucionar el cronograma de un ejercicio. Así, se ha podido introducir variables parametrizadas y añadir más claridad sobre la lectura, además de eliminar elementos que consumían memoria.
- Organización de datos y clases. Se han agrupado los diferentes componentes y se han separado por clases con finalidades similares, como ha sido el caso de los métodos compartidos de los tipos de ejercicios en el archivo *funcionesComunes.js*, los ficheros de validación para el login o registro de usuarios, entre otros.



8. Pruebas

En este apartado se especifican las diferentes pruebas llevadas a cabo para comprobar el correcto funcionamiento de la aplicación del entrenador.

8.1. Usabilidad en la web

Para el desarrollo de la aplicación se han tenido en cuenta las *heurísticas de Nielsen y Molich* [23]. Estas son una serie de reglas para evaluar el diseño de la interfaz de usuario en una página web. Se busca mejorar la calidad en la interacción con el sistema y por ello se comentan a continuación las pautas que se han seguido para conseguirlo con la aplicación del entrenador:

- **Visibilidad del estado del sistema.** El sistema le indicará al usuario el estado en el que se encuentra al realizar cualquier acción, por ejemplo, en la creación de un perfil de usuario, se notifica si se ha podido registrar correctamente (Ilustración 52) o en el caso de comprobar la solución de una gráfica dibujada por el usuario, se muestra por pantalla si ha sido un acierto o fallo (Ilustración 53).
- **Utilizar el mismo lenguaje que el usuario.** Se trata de emplear términos y palabras conocidas por el usuario, como es el caso de los nombres de botones, en los campos de texto o menús desplegados. Además, cabe destacar que el idioma de la aplicación está completamente en español. No se utilizan palabras como password, sino contraseña.
- **Control y libertad para el usuario.** Esta cualidad se centra más en la resolución de los ejercicios. En el caso de pulsar sobre el botón para mostrar la solución de un cronograma, se le anima al usuario a seguir intentando resolver el ejercicio antes de mostrarle la respuesta (Ilustración 54).
- **Consistencia y estándares.** La aplicación indica a los usuarios de forma clara las acciones y situaciones que se puedan obtener en base al perfil de usuario autenticado (Ilustración 55 e Ilustración 56).
- **Ayuda y documentación.** En este caso, la página de ayuda es donde se explica el contenido de la asignatura de forma breve y concisa que sirva de apoyo para la resolución de ejercicios. Además, lo más relevante es que un usuario alumno puede acceder fácilmente a ella, gracias al enlace del *toolbar* (Ilustración 57) que se encuentra disponible en todas las interfaces de usuario o desde la página de inicio (Ilustración 55) al temario en concreto sobre el cual se tiene dudas.

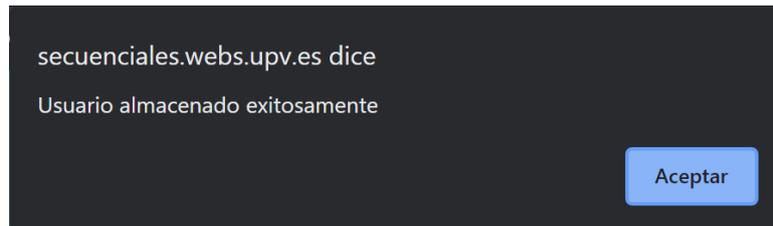


Ilustración 52. Mensaje indicando el correcto registro de un usuario.

Aciertos:
1
Fallos:
7

Ilustración 53. Campos estadísticos de aciertos y fallos.

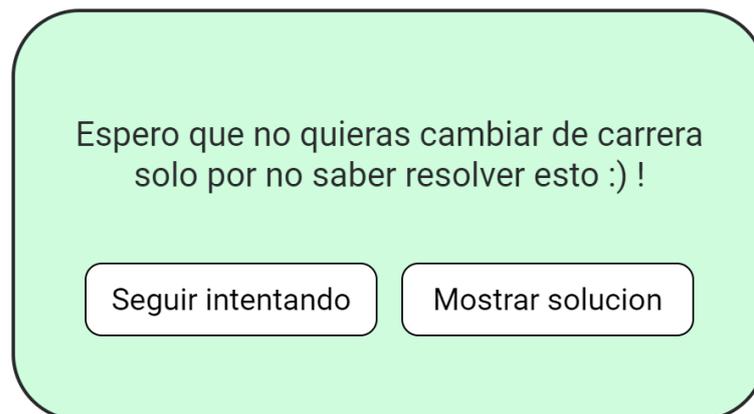


Ilustración 54. Mensaje aleatorio al intentar mostrar solución.

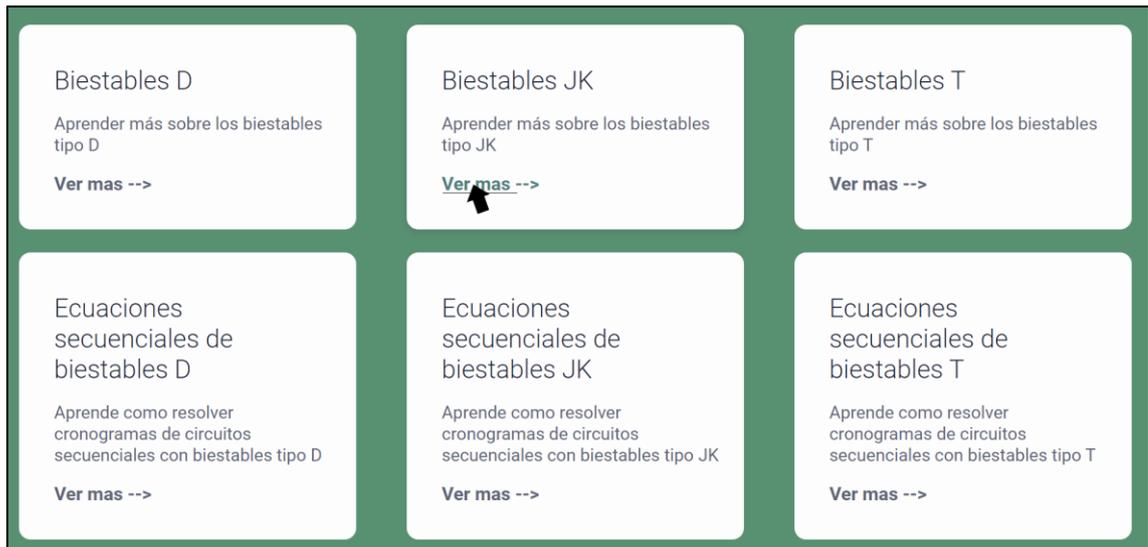


Ilustración 55. Accesos directos a las diferentes secciones de la página de ayuda.

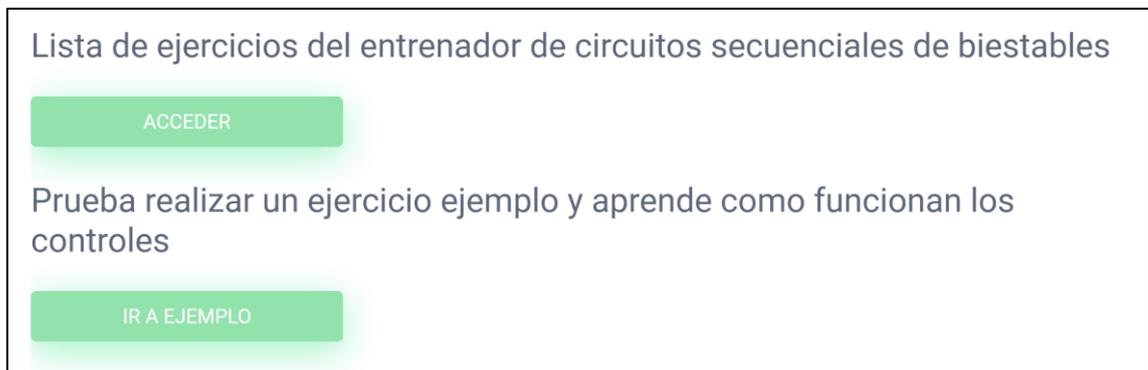


Ilustración 56. Accesos directos a la página de ejercicios y el ejemplo tutorial.



Ilustración 57. Barra de opciones de la aplicación del entrenador.

8.2. Despliegue

Finalizadas ya las etapas anteriores, ahora se puede poner en marcha la aplicación web y sus componentes en un servidor web de hosting. Para ello se ha empleado uno de los servicios de publicación web del ASIC [24], accesible por el personal y los alumnos de la UPV.

El ASIC pone a disposición de cualquier usuario de la UPV un espacio en la red para almacenar o publicar contenido. En este caso, dada la tecnología utilizada en este trabajo se ha requerido un Servicio de Publicación Avanzado [25]. Este servicio es compatible para sitios web dinámicos, junto a lenguajes de programación como PHP, JavaScript, tecnología ASP y bases de datos, entre otros.

De este modo, se ha procedido a rellenar el formulario de solicitud, como se puede observar en la Ilustración 58, donde se han indicado los siguientes criterios:

- **Tipo de solicitud:** Lanzar o suspender un servicio. En este caso se ha procedido a dar de “Alta” una web avanzada.
- **Tipo de web:** Puesto que se trata de una página dedicada a mostrar contenido relacionado con una asignatura, se ha solicitado el tipo “Institucional”.
- **Nombre web:** Este campo registra el nombre de dominio de la aplicación web. En este caso, se ha decidido por “secuenciales”.
- **Descripción web:** Se ha especificado una breve descripción del propósito del sitio web a ser dado de alta.

The screenshot shows a web browser window with the URL https://intranet.upv.es/pls/soalu/sic_piolinweb.SolicitarPlesk?P_IDIOMA=v&p_vista=intranet. The page title is 'Solicitud de alta de una web avanzada'. The form includes the following fields and options:

- ¿Ha leído y acepta la normativa?**: Radio buttons for 'Sí' (selected) and 'No'.
- Tipo de solicitud:** Dropdown menu set to 'Alta'.
- Teléfono:** Text input field.
- Tipo de web:** Dropdown menu set to 'Institucional'.
- Nombre web:** Text input field containing 'secuenciales'.
- Descripción web:** Text input field containing 'Inicio de sistemas secuenciales síncronos'.

An orange button labeled 'Enviar solicitud' is located at the bottom of the form.

Ilustración 58. Solicitud de alta de un servicio de publicación avanzado.

Una vez completado el formulario y enviada la solicitud, se ha accedido a través de la URL “<https://secuenciales.webs.upv.es:8443>” al panel de control web de la aplicación (Ilustración 59). La universidad utiliza los servicios del software Plesk [26] y a través de su panel de control, se puede gestionar el sitio web, el contenido publicado, administrar la base de datos o modificar diferentes configuraciones, entre otras tareas.

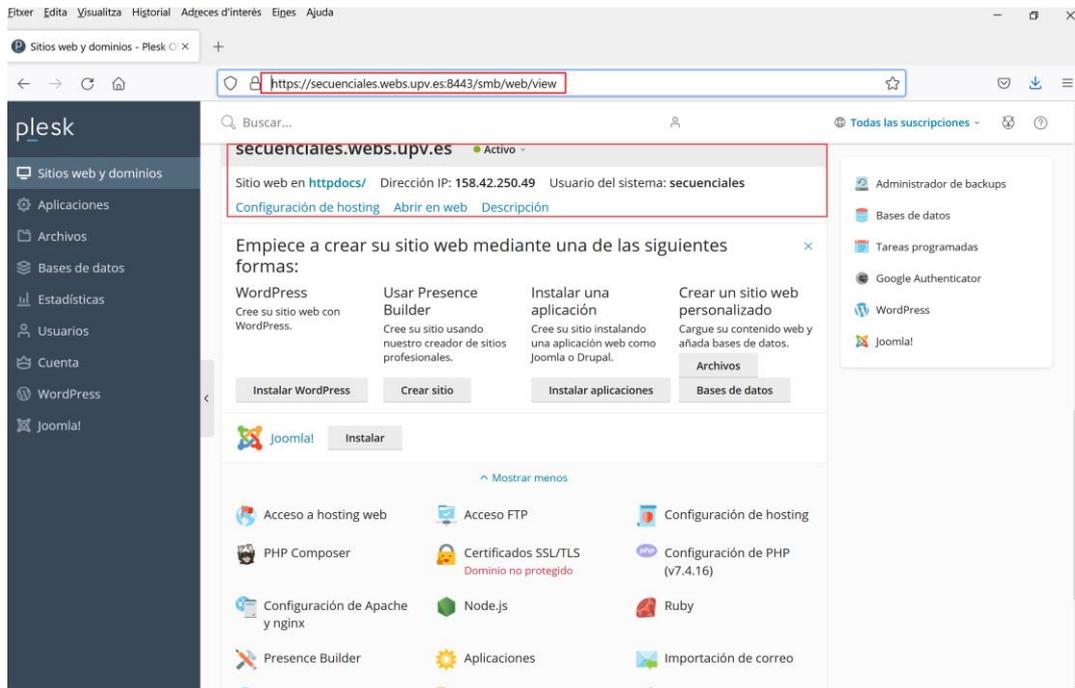
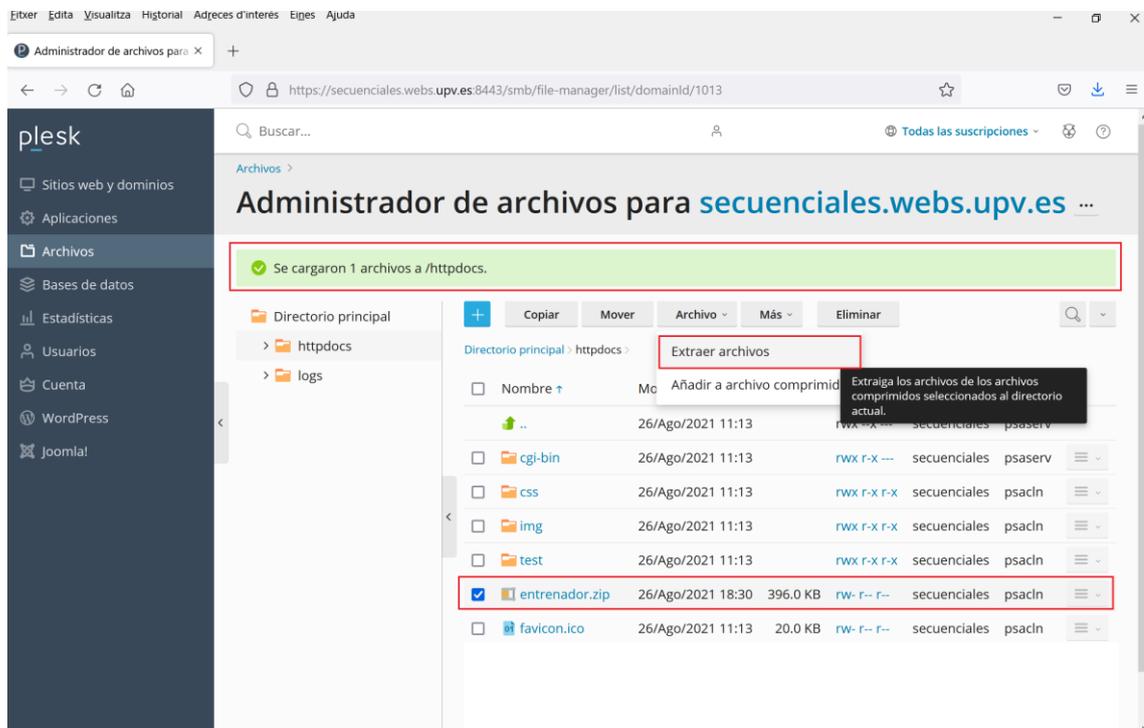


Ilustración 59. Panel de control de la web de secuenciales.

Puesto que el entrenador se ha programado a nivel local, se han comprimido todos los documentos para así subirlos a la carpeta `/httpdocs` disponible a través de la opción Archivos del menú lateral como se puede ver en la Ilustración 60.



A continuación se ha procedido a crear una base de datos [Ilustración 61] indicando un nombre, el sitio web al que queremos que se asocie, además de un usuario y contraseña con el cual se podrá controlar los permisos y acceso a esta.

Ilustración 61. Creación de una base de datos.

Una vez creada, se dispondrá de un panel de control para la base de datos [Ilustración 62], donde se podrá realizar diferentes acciones, entre las cuales se encuentran señaladas:

- **phpMyAdmin** [27]: es un acceso directo a la herramienta para administrar las tablas, relaciones e información de la base de datos.
- **Importar volcado**: esta opción permite volcar el contenido de una base de datos existente en la base de datos creada. En este caso, se ha exportado un backup de la base de datos trabajada en local y se ha subido a la recién añadida, como se puede observar en la Ilustración 63.
- **Información de la conexión**: a través de este botón se puede consultar la información relacionada con la base de datos (el nombre, servidor que la aloja, el usuario y la contraseña) y que se ha utilizado en el fichero de conexión descrito en la fase de implementación. Este fichero se modificará directamente en el directorio donde se aloja en Plesk.

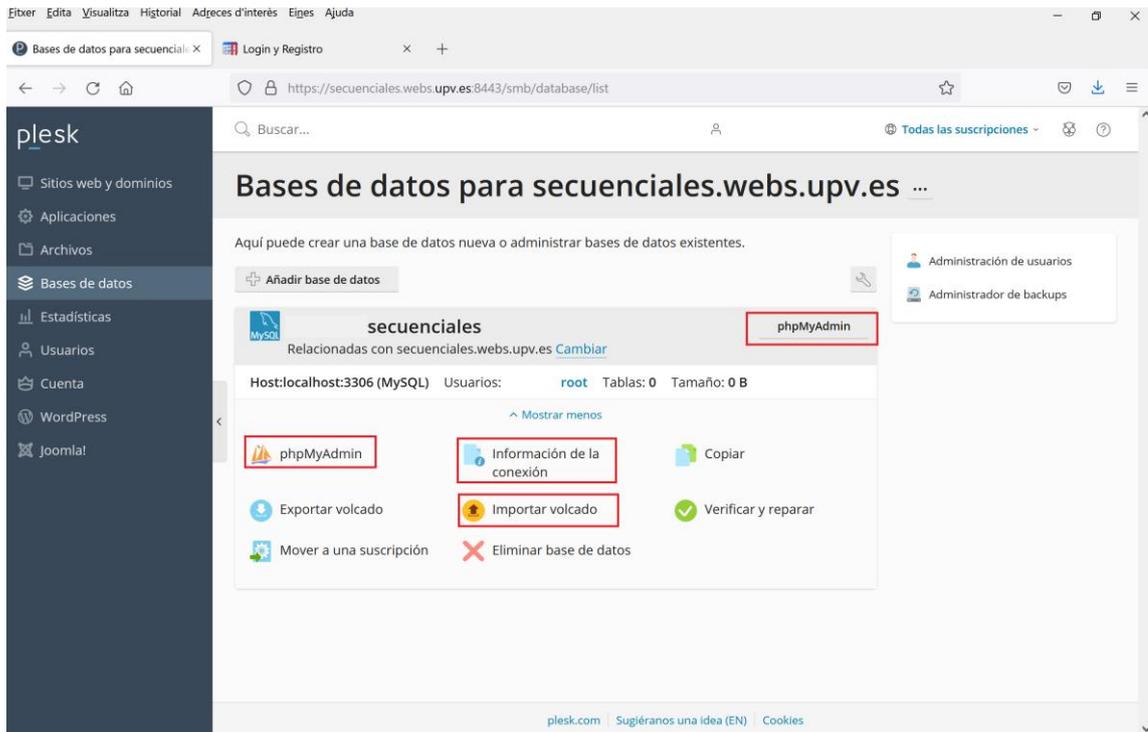


Ilustración 62. Panel de control de la base de datos.

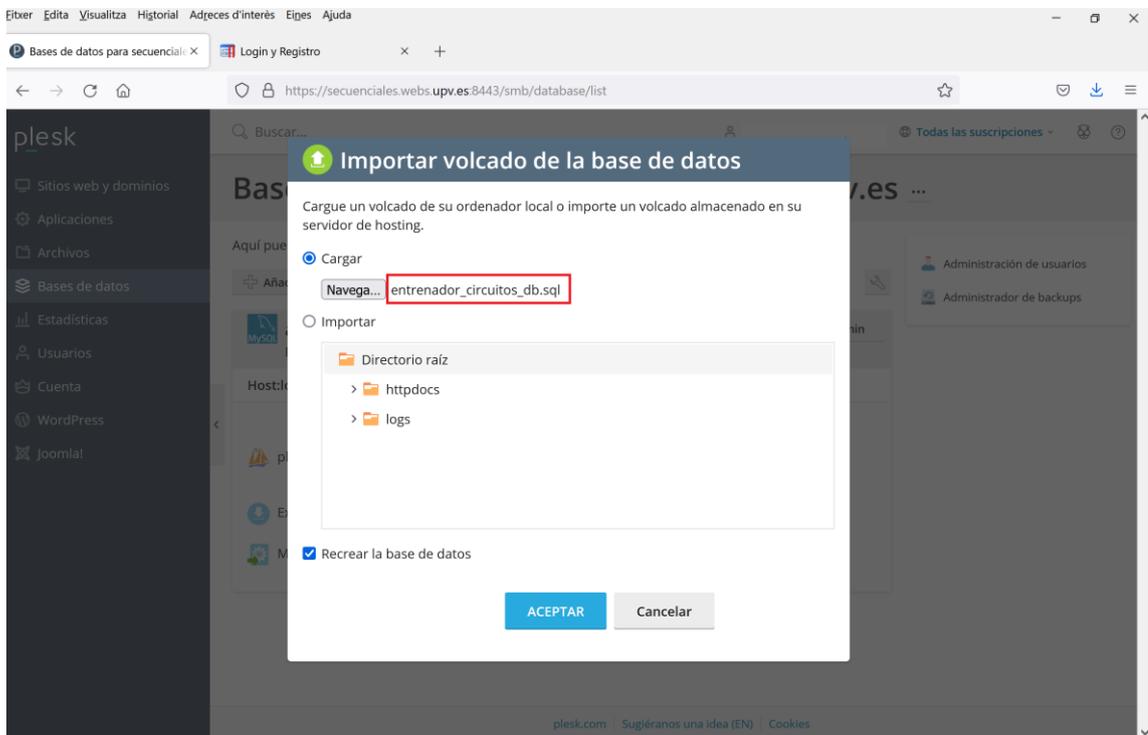


Ilustración 63. Importación de un archivo a la base de datos.

Por último, se comprueba a través del enlace *secuenciales.webs.upv.es* que se ha podido desplegar la aplicación con éxito, como muestra la Ilustración 64.

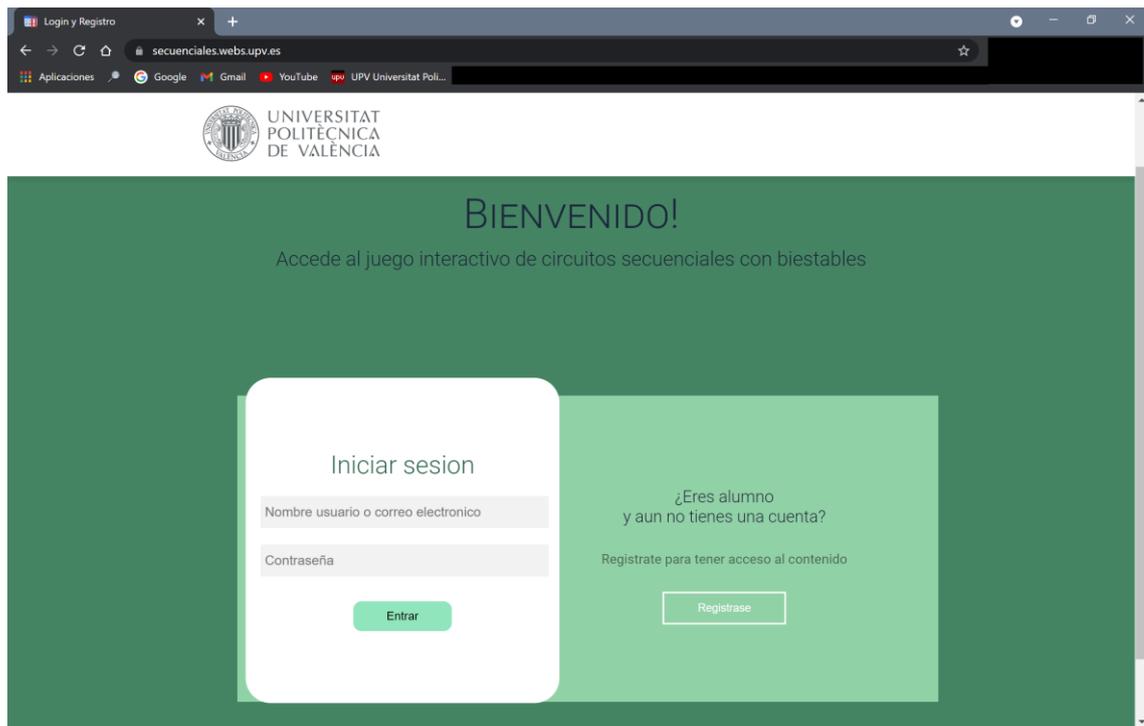


Ilustración 64. Acceso a la página de la aplicación del entrenador.

8.2.1. Pruebas de funcionalidad y compatibilidad

Se ha validado el correcto funcionamiento del entrenador, registro y login de los usuarios y las funcionalidades de los diferentes perfiles de usuario en los navegadores web Google Chrome, Microsoft Edge y Mozilla Firefox.

8.2.2. Pruebas de seguridad

Se ha comprobado que ingresar a las páginas principales de los diferentes perfiles de usuario no es posible al menos que el usuario haya iniciado sesión o se haya registrado en la aplicación del entrenador.

De este modo, si por ejemplo, se intenta acceder desde una URL a la página del administrador, esta mostrará un cuadro de alerta indicando que para obtener acceso es necesario estar logado con un perfil de administrador. En la Ilustración 65 se puede observar el mensaje indicado.

secuenciales.webs.upv.es dice

Para acceder a la pagina, tienes que iniciar sesión con una cuenta de administrador

Aceptar

Ilustración 65. Mensaje de restricción de acceso a la página del administrador.

9. Conclusiones

En vista de los puntos anteriores se puede afirmar que se han alcanzado todos los objetivos propuestos para el entrenador en los apartados de objetivos, requisitos y casos de uso. Se ha desarrollado una aplicación web para los alumnos de primer año de la asignatura de “Fundamentos de computadores” orientada a la resolución de cronogramas de circuitos secuenciales síncronos con biestables, que además cuenta con un sistema de gestión de usuarios para poder almacenar y consultar información relativa a los ejercicios realizados por los alumnos.

9.1. Valoración personal

Desde un punto de vista personal, este trabajo ha supuesto un verdadero desafío para mí. Considerando la rama que he cursado en ingeniería de software, la mayoría de los trabajos que he emprendido han sido en grupos donde se repartían las tareas y se realizaban intercambios de ideas. A pesar de ello, he intentado cubrir la costumbre de apoyarme en un equipo añadiendo creatividad, dedicación y esfuerzo.

En un ámbito más profesional considero que será de mucha utilidad el conocimiento adquirido durante la orquestación de este proyecto, ya sea acerca de las tecnologías empleadas o el ámbito de las aplicaciones web.

Más aún me ha alegrado poder aplicar conocimientos adquiridos a lo largo de mi carrera universitaria en el grado de ingeniería informática. Sobre todo asignaturas como GPR (Gestión de Proyectos), PSW (Proceso del Software), PIN (Proyecto de Ingeniería de software) e ISW (Ingeniería del software) que me han servido como principal respaldo para el desarrollo y la estructuración de las diferentes fases de este proyecto.

Cabe destacar el apoyo de asignaturas como IPC (Interfaz Persona Computador) y CSO (Calidad del Software) con el diseño, la calidad y aspectos de usabilidad que he tenido en cuenta en el análisis, diseño y en la etapa de implementación de este trabajo, la ayuda de la asignatura de BDA (Bases de Datos) en términos de conceptos, esquemas y operaciones a realizar sobre bases de datos relaciones, además de todas las asignaturas de programación vistas hasta ahora que me han proporcionado una base inicial y fundamental para mis futuras etapas de aprendizaje.

9.2. Trabajos futuros

Durante el transcurso de este proyecto software, han surgido diferentes ideas que podrían complementar o mejorar la aplicación. Si bien estas características no se han podido añadir durante el desarrollo de este TFG, se podría tener en cuenta en futuras versiones. A continuación, se listan algunos de las propuestas surgidas:

- **Recuperación de contraseña:** Habilitar una opción en la página de acceso para cualquier usuario.

- **Certificado digital SSL:** Sería interesante obtener un certificado que garantice una conexión segura entre el servidor web y el navegador para proteger el envío de las contraseñas.
- **Diseño automatizado de la gráfica del circuito:** Una posible mejora podría implicar crear un algoritmo que automáticamente diseñe el circuito de un ejercicio en base a las ecuaciones de este.
- **Un perfil de los alumnos:** Permitiendo que los alumnos puedan modificar su información personal o consultar algunos datos estadísticos.

10. Bibliografía

- [1] Aplicación web para el aprendizaje y resolución de mapas de Karnaugh. [En línea] [Fecha de consulta 1 de septiembre de 2021] Disponible en <https://riunet.upv.es/handle/10251/8996>
- [2] Creación de un sitio web para la realización de ejercicios de representación de números enteros. [En línea] [Fecha de consulta 1 de septiembre de 2021] Disponible en <https://riunet.upv.es/handle/10251/15248>
- [3] Entrenador de operaciones bitwise. [En línea] [Fecha de consulta 1 de septiembre de 2021] Disponible en <https://riunet.upv.es/handle/10251/68605>
- [4] Construcción de un sitio web para la realización de ejercicios IEEE 754. [En línea] [Fecha de consulta 1 de septiembre de 2021] Disponible en <https://riunet.upv.es/handle/10251/11581>
- [5] Entrenador gráfico de biestables. [En línea] [Fecha de consulta 1 de septiembre de 2021] Disponible en <https://riunet.upv.es/handle/10251/69230>
- [6] Estudio Nacional sobre la Situación Laboral de los Profesionales del Sector TI . Consejo General de Colegios Oficiales de Ingeniería en Informática (CCII). 2015
- [7] Sistemas digitales. Lloris Ruiz, Antonio | Prieto Espinosa, Alberto | Parrilla Roure, Luis | Aravaca, Madrid : McGraw-Hill/Interamericana de España | D.L. 2003
- [8] Proceso del desarrollo del software. [En línea] [Fecha de consulta 3 de agosto de 2021] Disponible en https://es.wikipedia.org/wiki/Proceso_del_desarrollo_del_software#:~:text=En%20ingenier%C3%ADa%20del%20software%2C%20un,vida%20del%20desarrollo%20de%20software.
- [9] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998.
- [10] Funciones hash en php. [En línea] [Fecha de consulta 2 de julio de 2021] Disponible en <https://www.php.net/manual/es/function.hash.php>
- [11] HTML. [En línea] [Fecha de consulta 15 de abril de 2021] Disponible en <https://www.w3schools.com/html/>
- [12] PHP. [En línea] [Fecha consulta 29 de abril de 2021] Disponible en <https://www.php.net/>
- [13] JavaScript. [En línea] [Fecha consulta 18 de mayo de 2021] <https://www.w3schools.com/js/>
- [14] CSS. [En línea] [Fecha consulta 19 de mayo de 2021] <https://www.w3schools.com/css/>

- [15] Garrido Tejero, Antonio. Arquitectura Del Software Multicapa. Universitat Politècnica de València, 2016. [En línea] [Fecha de consulta 20 de julio de 2021] Disponible en <https://riunet.upv.es/handle/10251/63615>
- [16] Visual Studio Code. [En línea] [Fecha de consulta 5 de febrero de 2021]. Disponible en <https://visualstudio.microsoft.com/>
- [17] WampServer. [En línea] [Fecha de consulta 2 de agosto de 2021]. Disponible en <https://www.wampserver.com/en/>
- [18] Herramienta Canva. [En línea] [Fecha de consulta 17 de agosto de 2021]. Disponible en <https://www.canva.com/>
- [19] Herramienta GenMyModel. [En línea] [Fecha de consulta 11 de agosto de 2021]. Disponible en <https://www.genmymodel.com/>
- [20] Herramienta Creately. [En línea] [Fecha de consulta 20 de agosto de 2021]. Disponible en <https://creately.com/es/home/>
- [21] Herramienta Balsamiq. [En línea] [Fecha de consulta 20 de agosto de 2021]. Disponible en <https://balsamiq.cloud/>
- [22] Sobre refactorización. [En línea] [Fecha de consulta 25 de agosto de 2021]. Disponible en <https://refactoring.guru/es>
- [23] 10 heurísticas de usabilidad de Nielsen y Molich. [En línea] [Fecha de consulta 29 de agosto de 2021] <https://markos-esther.com/10-heuristicas-de-usabilidad-de-nielsen-y-molich/>
- [24] Área de Sistemas de la Información y las Comunicaciones: UPV. [En línea] [Fecha de consulta 27 de agosto de 2021] Disponible en <https://www.upv.es/entidades/ASIC/catalogo/527545normalc.html>
- [25] Servicio de webs avanzadas. [En línea] [Fecha de consulta 27 de agosto de 2021] Disponible en <https://wiki.upv.es/confluence/display/MANUALES/Servicio+de+webs+avanzadas>
- [26] Plesk. [En línea] [Fecha de consulta 27 de agosto de 2021] Disponible en <https://www.plesk.com/>
- [27] phpMyAdmin. [En línea] [Fecha de consulta 29 de agosto de 2021]. Disponible en <https://www.phpmyadmin.net/>