



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

APLICACIÓN WEB PARA EL SEGUIMIENTO DEL APRENDIZAJE ACTIVO EN ASIGNATURAS DE ESTUDIOS UNIVERSITARIOS

Alumno: Francisco José Calzado Moya

Tutor: D. Antonio Jesús Rivera Rivas y
D. Francisco Charte Ojeda

Dpto: Departamento de Informática

Junio, 2019



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Antonio Jesús Rivera Rivas y Don Francisco Charte Ojeda , tutores del Proyecto Fin de Carrera titulado: Aplicación web para el seguimiento del aprendizaje activo en asignaturas de estudios universitarios, que presenta Francisco José Calzado Moya, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2019

El alumno:

Los tutores:

Francisco José Calzado Moya

Antonio Jesús Rivera Rivas

Francisco Charte Ojeda

Expresar mi enorme agradecimiento a todas las personas que han contribuido, en mayor o menor medida, al desarrollo de este trabajo final de grado, especialmente a:

Mi familia por confiar siempre en mí y apoyarme en cada momento de devilidad que he podido tener a lo largo de, no solo el desarrollo de este proyecto, sino a lo largo de todo el grado. Sin ellos no me cabe duda de que no me encontraría ni remotamente cerca de alcanzar los objetivos que en su día me propuse.

Mis tutores Paco y Antonio, por vuestra completa disposición e involucración en el proyecto, guíarme de forma excepcional y resolver toda duda que os presentara.

Por último a todos mis amigos que han sugerido mejoras en el desarrollo del proyecto y que han estado en cada momento de devilidad y sobre todo de festejo.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	4
ÍNDICE DE TABLAS	7
CAPÍTULO 1.- INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS DEL PROYECTO.....	8
1.1 Introducción al proyecto.....	8
1.2 Motivación.....	8
1.3 Objetivos.....	8
1.4 Estructura de la memoria	9
CAPÍTULO 2.- LENGUAJES Y HERRAMIENTAS UTILIZADAS.	11
2.1 Frontend	11
2.2 Backend.....	13
2.3 Herramientas útiles en el desarrollo	14
CAPÍTULO 3.- INGENIERÍA DEL SOFTWARE.....	18
3.1 Introducción a la ingeniería del software.	18
3.2 Etapas	18
3.3 Definición de Requisitos.....	20
3.3.1 Requisitos funcionales.....	21
3.3.2 Requisitos no funcionales.....	22
3.3.3 Requisitos de Calidad:.....	22
3.3.4 Restricciones.....	22
3.4 Modelo de desarrollo software	22
3.5 Elección de desarrollo software	24
CAPÍTULO 4.- ANÁLISIS	26
4.1 Especificación de requisitos	26
4.2 Definición de Actores	29
4.3 Modelos de casos de uso	31
4.4 Narrativas de los casos de uso	33
CAPÍTULO 5.- PLANIFICACIÓN	40
5.1 Estimación de requisitos	40
5.2 Diagrama de Gantt.....	46
5.3 Estimación de costes	47
CAPÍTULO 6.- DISEÑO.....	51
6.1 Diseño Arquitectónico	51
6.2 Diseño de Base de Datos	54

6.3 Diseño de página web	55
6.4 Modelo de Dominio	68
6.5 Diagramas de secuencia.....	68
CAPÍTULO 7.- IMPLEMENTACIÓN Y PRUEBAS.	75
7.1 Uso de plantillas	75
7.2 Organización interna del proyecto.....	75
7.3 Creación de tablas en base de datos	77
7.4 Clases orientadas a objetos.....	77
7.5 Validadores.....	78
7.6 Uso de AJAX en PHP	79
7.7 Enrutamiento y redireccionamientos	81
7.7.1 Enrutamiento	81
7.7.2 Redirección	81
7.8 Uso de métodos PHP	82
CAPÍTULO 8.- CONCLUSIONES.....	83
8.1 Resumen final.....	83
8.2 Conclusión Final	83
8.3 Desarrollo Futuro	84
ANEXO I. Código fuente.....	86
ANEXO II. Manual de usuario.....	87
ANEXO III. Instalación y Configuración XAMPP	104
Bibliografía	111

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Logo JavaScript.....	11
Ilustración 2. Logo HTML	12
Ilustración 3. Logo CSS.....	12
Ilustración 4. Logo PHP	13
Ilustración 5. Logo MySQL	13
Ilustración 6. Logo Visual Paradigm	14
Ilustración 7. Logo Netbeans	15
Ilustración 8. Logo XAMPP	15
Ilustración 9. Logo Docker	16
Ilustración 10. Fases de la ingeniería del software [43]	20
Ilustración 11. Modelo incremental	22
Ilustración 12. Logo Admin	25
Ilustración 13. Logo Profesor.....	25
Ilustración 14. Logo Estudiante	25

Ilustración 15. Logo estudiante no aceptado	25
Ilustración 16. Equipo de torre.....	25
Ilustración 17. Equipo portátil	25
Ilustración 18. Logo Admin	26
Ilustración 19. Logo Profesor.....	26
Ilustración 20. Logo Estudiante	26
Ilustración 21. Logo estudiante no aceptado	26
Ilustración 22. Logo Admin	29
Ilustración 23. Logo Profesor.....	30
Ilustración 24. Logo Estudiante	30
Ilustración 25. Logo estudiante no aceptado	30
Ilustración 26. Casos de uso: Gestión de asignaturas	31
Ilustración 27. Casos de uso: Gestión de usuarios	32
Ilustración 28. Casos de uso: Gestión de estadísticas.....	33
Ilustración 29. Diagrama de Gantt simplificado.....	46
Ilustración 30. Diagrama de Gantt desarrollado.....	46
Ilustración 31. Equipo de torre.....	48
Ilustración 32. Equipo portátil	48
Ilustración 33. Resumen visual del MVC [19]	52
Ilustración 34. Modelo Entidad Relación.....	54
Ilustración 35. Mockup: Registro	56
Ilustración 36. Mockup: <i>Login</i>	56
Ilustración 37. Mockup: Asignaturas no matriculadas	57
Ilustración 38. Mockup: Asignaturas matriculadas	57
Ilustración 39. Mockup: Temas de la asignatura.....	58
Ilustración 40. Mockup: Recursos Visuales	59
Ilustración 41. Mockup: <i>Flashcards</i> [Profesores]	60
Ilustración 42. Mockup: <i>Flashcards</i> [Estudiantes] I	61
Ilustración 43. Mockup: <i>Flashcards</i> [Estudiante] II. Fallo	62
Ilustración 44. Mockup: <i>Flashcards</i> [Estudiante] II. Acierto.....	63
Ilustración 45. Mockup: <i>Flashcards</i> [Estudiante] III. No superado.....	64
Ilustración 46. Mockup: <i>Flashcards</i> [Estudiante] III. Superado.....	65
Ilustración 47. Mockup: Estadísticas I.....	66
Ilustración 48. Mockup: Estadísticas II.....	66
Ilustración 49. Mockup: Gestión de usuarios	67
Ilustración 50. Diagrama del modelo de dominio	68
Ilustración 51. Diagrama de Secuencia: Registro	69
Ilustración 52. Diagrama de Secuencia: Inicio de sesión	70
Ilustración 53. Diagrama de Secuencia: Modificar usuario	70
Ilustración 54. Diagrama de Secuencia: Añadir Asignatura	71
Ilustración 55. Diagrama de Secuencia: Acceder a los recursos visuales.....	71
Ilustración 56. Diagrama de Secuencia: Añadir <i>flashcard</i>	72
Ilustración 57. Diagrama de Secuencia: Editar <i>flashcard</i>	72
Ilustración 58. Diagrama de Secuencia: Realizar <i>flashcard</i> (Estudiante)	73
Ilustración 59. Diagrama de Secuencia: Consultar estadísticas.....	74
Ilustración 60. Diagrama de Secuencia: Descargar estadísticas con formato CSV.....	74
Ilustración 61. Organización interna del proyecto: app	75
Ilustración 62. Organización interna del proyecto: Control.....	76

Ilustración 63. Organización interna del proyecto: Plantillas	76
Ilustración 64. Clase orientada a objetos: Asignatura	77
Ilustración 65. Constructor del validador del registro.	78
Ilustración 66. Función validar_nombre() asociado al validador de registro.	78
Ilustración 67. Función validar_confirm_password() asociada al validador de registro.	79
Ilustración 68. Script con AJAX	80
Ilustración 69. Consulta.php con AJAX.....	80
Ilustración 70. Redirección	81
Ilustración 71. Redireccionamiento a ruta de los temas de una asignatura.....	82
Ilustración 72. Manual de usuario. Registro	87
Ilustración 73. Manual de usuario. Registro correcto	88
Ilustración 74. Manual de usuario. Registro incorrecto	88
Ilustración 75. Manual de usuario. <i>Login</i>	89
Ilustración 76. Manual de usuario. <i>Login</i> incorrecto.....	89
Ilustración 77. Manual de usuario. <i>Login</i> correcto.....	90
Ilustración 78. Manual de usuario. <i>Index</i> usuario.....	90
Ilustración 79. Manual de usuario. <i>Index</i> usuario sin asignaturas	90
Ilustración 80. Manual de usuario. Asignaturas no matriculadas	91
Ilustración 81. Manual de usuario. Temario de una asignatura.....	92
Ilustración 82. Manual de usuario. Recursos visuales	92
Ilustración 83. Manual de usuario. <i>Flashcards</i> para estudiante I	93
Ilustración 84. Manual de usuario. <i>Flashcards</i> correcta	93
Ilustración 85. Manual de usuario. <i>Flashcard</i> incorrecta	94
Ilustración 86. Manual de usuario. Tema superado	94
Ilustración 87. Manual de usuario. Tema no superado	95
Ilustración 88. Manual de usuario. Vista de recursos visuales de profesor	96
Ilustración 89. Manual de usuario. Añadir nuevo recurso visual	97
Ilustración 90. Manual de usuario. Gestión de <i>flashcards</i>	98
Ilustración 91. Manual de usuario. Selección de estadísticas	99
Ilustración 92. Manual de usuario. Estadísticas agregadas	100
Ilustración 93. Manual de usuario. Gestión de usuarios.....	101
Ilustración 94. Manual de usuario. Añadir asignatura	102
Ilustración 95. Manual de usuario. Formulario de nueva asignatura	102
Ilustración 96. Manual de usuario. Añadir nuevo tema	103
Ilustración 97. Instalación y configuración XAMPP. Acceso a la web oficial	104
Ilustración 98. Instalación y configuración XAMPP. Instalación I	104
Ilustración 99. Instalación y configuración XAMPP. Instalación II	105
Ilustración 100. Instalación y configuración XAMPP. Instalación IV	105
Ilustración 101. Instalación y configuración XAMPP. Instalación V	106
Ilustración 102. Instalación y configuración XAMPP. Instalación VI	106
Ilustración 103. Instalación y configuración XAMPP. Selección lenguaje	107
Ilustración 104. Instalación y configuración XAMPP. Pantalla principal	107
Ilustración 105. Instalación y configuración XAMPP. Añadir base de datos.....	108
Ilustración 106. Instalación y configuración XAMPP. Inserción de la base de datos	108
Ilustración 107. Instalación y configuración XAMPP. Panel de administración	109
Ilustración 108. Instalación y configuración XAMPP. Ubicación del proyecto	109
Ilustración 109. Instalación y configuración XAMPP. Acceso al proyecto	110

ÍNDICE DE TABLAS

Tabla 1. Requisitos, prioridad y estimación de los mismos.....	43
Tabla 2. División de Incrementos	44
Tabla 3. Planificación temporal de los incrementos	45
Tabla 4. Estimación de costes.....	50

CAPÍTULO 1.- INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS DEL PROYECTO.

1.1 Introducción al proyecto.

La sociedad en la que vivimos se encuentra cada vez más inmersa en el mundo tecnológico. Lo podemos apreciar en todos y cada uno de los aspectos de nuestra vida. Vivimos en un mundo global en el que el uso de las nuevas tecnologías juega un papel clave, nos facilita el poder comunicarnos de una manera que hace años era impensable, tenemos acceso a infinidad de información y recursos a los que, sin estas nuevas tecnologías, sería imposible acceder; Hoy en día es posible estar conectados desde la más remota aldea de África con las grandes urbes del planeta utilizando simplemente un teléfono. De esta forma aparecen nuevos instrumentos y tecnologías que facilitan la comunicación entre las personas y el acceso al conocimiento. El ámbito educativo no deja de ser un trasunto del resto de la sociedad y la aplicación de estas nuevas tecnologías ha modificado y mejorado la relación entre el profesor y el estudiante, de manera que a las lecciones magistrales que se impartían en la universidad les viene a complementar el uso de estas nuevas tecnologías y partiendo de esa idea hemos desarrollado una aplicación informática que ayude a profesor y alumnos a establecer una relación dinámica y eficaz en el desarrollo de la asignatura.

1.2 Motivación

La motivación de esta aplicación web es ofrecer una plataforma online que facilite una serie de herramientas que permitan la implementación de una metodología de aprendizaje activo para una asignatura.

1.3 Objetivos

El objetivo de este proyecto es el desarrollo de una aplicación que ofrezca a los profesores una forma de ejercer un seguimiento activo sobre el alumnado (que participe activamente en la plataforma) de una asignatura.

Para llevar a cabo el objetivo principal del proyecto, se presentan diferentes subobjetivos:

- Estudio del lenguaje de marcado HTML y las hojas de estilos CSS, así como del lenguaje ECMAScript. Estos serán los elementos esenciales para el desarrollo de la interfaz de usuario de la aplicación.
- Selección de una tecnología de servidor web, incluyendo lenguaje (PHP, Node.js, Ruby, etc.) y base de datos, para el desarrollo de la parte servidor de la solución.
- Diseño de la base de datos que alojará los recursos didácticos, tales como *flashcards* y enlaces a vídeos, así como las estadísticas de uso que de ellos hagan los estudiantes registrados.
- Implementación e integración de las partes cliente (interfaz) y servidor de la solución, definiendo la API apropiada.
- Redacción de los manuales de instalación y uso de la solución desarrollada, así como de la memoria correspondiente del TFG.

1.4 Estructura de la memoria

Los capítulos de la memoria serán los siguientes:

- **Lenguajes, herramientas y tecnologías utilizadas.**
 - Se exponen los principales lenguajes, herramientas y tecnologías utilizadas en el desarrollo del proyecto, así como una breve descripción de la opción elegida y por qué se ha elegido dicha opción.
- **Ingeniería del software.**
 - Se define la ingeniería del software, sus fases, distintos tipos de requisitos, así como el modelo utilizado en el desarrollo del proyecto junto a la explicación del por qué es la opción más atractiva.
- **Análisis**
 - Se realiza la especificación de requisitos, incluye los distintos modelos de casos de uso junto a narrativas de los propios casos de uso.

- **Planificación.**
 - Se realiza la definición de actores, estimación de requisitos en dificultad y prioridad, diagrama de Gantt tanto simplificado como desarrollado y estimación de todos los costes que podría suponer el desarrollo del proyecto.
- **Diseño.**
 - Se explica el diseño arquitectónico utilizado (Modelo-Vista-Controlador) y cómo surgió, además del diseño de base de datos junto al diseño de la página web llevado a cabo con el uso de mockups. Finalmente se incluye el modelo de dominio diseñado para el proyecto.
- **Implementación y pruebas.**
 - Se lleva a cabo una explicación resumida de las herramientas y técnicas de programación utilizadas.
- **Conclusiones.**
 - Incluye la conclusión final del proyecto tanto a nivel personal como a nivel de desarrollo junto al desarrollo que sufrirá el proyecto en los próximos incrementos.
- **Anexos varios.**
 - Cuenta con varios anexos donde se adjunta el código fuente, manual de usuario e instalación y configuración de la herramienta XAMPP

CAPÍTULO 2.- LENGUAJES Y HERRAMIENTAS UTILIZADAS.

A continuación, se va a hacer referencia a los lenguajes y herramientas utilizadas en el desarrollo del proyecto.

2.1 Frontend

JavaScript



Ilustración 1. Logo JavaScript

Es un lenguaje de programación interpretado usado para la creación de aplicaciones web, tanto en *frontend* como en el *backend*. Es el lenguaje principal de este proyecto. Últimamente está en auge, ya que su eficiencia y su fácil curvatura de aprendizaje, hace que sea muy usado. Actualmente se encuentra en el puesto número 6 del ranking de TIOBE.

¿Por qué utilizar JavaScript en mi aplicación?

JavaScript es el lenguaje que todo programador debería conocer, debido a su relativamente baja curva de aprendizaje, versatilidad y altísima presencia en prácticamente cualquier proyecto que se realice hoy día [30].

Podemos encontrar JavaScript en aplicaciones móviles, web, bases de datos, videojuegos, servidores...El alcance que presenta este lenguaje es enorme y considero importante el uso del mismo.

HTML (HyperText Markup Language)



Ilustración 2. Logo HTML

Es un lenguaje de etiquetas usado para desarrollar páginas webs, en él se puede incrustar tanto textos, imágenes, vídeos y juegos entre otros. Es un *standard* de *World Wide Web Consortium* (W3C).

¿Por qué usar HTML en mi aplicación?

HTML dota de estructura y contenido a la página web, además de complementarse de manera sensacional con CSS y con la combinación de estos dos se puede obtener una página web de gran calidad tanto a nivel visual como funcional [31].

CSS (Cascading Stylesheets)



Ilustración 3. Logo CSS

Es un lenguaje de estilos para la definición de documentos HTML. En él se pueden definir los colores, los márgenes y otros aspectos, para mejorar el aspecto de las páginas webs.

¿Por qué usar CSS en mi aplicación?

La filosofía de CSS trata de separar estructura de estilo. De la estructura el encargado será HTML y del estilo CSS, de esta forma es fácilmente

diferenciable y escalable el uso de estilos respecto a una página HTML básica [32].

En mi caso me ha sido muy útil para dotar a toda la aplicación de un estilo común en todas las secciones de la aplicación así de facilitar enormemente la labor de realizar cambios en el estilo visual de la misma.

2.2 Backend

PHP



Ilustración 4. Logo PHP

PHP, acrónimo recursivo en inglés de PHP: *Hypertext Preprocessor*, es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

¿Por qué usar PHP?

PHP es un lenguaje totalmente libre y abierto. Tiene una curva de aprendizaje muy baja, su sintaxis es simple y cumple estándares básicos de la programación orientada a objetos. No son necesarios complejos entornos de desarrollo, que incluso necesitan su propio periodo de aprendizaje. Puedes programar en PHP sin más ayuda que el bloc de notas, todos los IDEs disponibles son gratuitos y los entornos de desarrollo son de rápida y fácil configuración.

MySQL



Ilustración 5. Logo MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por *Oracle Corporation* y está considerada como la base de datos de código abierto más popular del mundo

¿Por qué usar MySQL?

MySQL es una base de datos gratuita, fácil de usar, segura, muy rápida estable y eficiente, además de ser compatible con *Linux* y *Windows*, lo cual la hace la base de datos más popular en todo el mundo

2.3 Herramientas útiles en el desarrollo

Visual Paradigm



Visual Paradigm

Ilustración 6. Logo Visual Paradigm

Es una herramienta UML CASE compatible con UML 2, SysML y *Business Process Modeling Notation* del *Object Management Group*. Además de la compatibilidad con el modelado, proporciona capacidades de generación de informes e ingeniería de código [33].

¿Por qué usar Visual Paradigm?

Tras estudiar otras herramientas de modelado UML y modelado de diagramas como puede ser *Visio*, de *Microsoft*, he terminado decantándome por la herramienta con la que más familiarizado estoy, ya que ha sido utilizada en diversas asignaturas a lo largo del grado y gracias a los convenios realizados con la universidad de Jaén con la herramienta tengo acceso a la aplicación completa a través de la Licencia Académica.

Netbeans



Netbeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. *NetBeans IDE* es un producto libre y gratuito sin restricciones de uso.

Ilustración 7. Logo Netbeans

¿Por qué utilizar Netbeans?

NetBeans para mi es el entorno de desarrollo más completo e intuitivo que he encontrado tras probar otros entornos como Eclipse o Visual Code, debido a que es el entorno con el que he trabajado desde un principio debido a la facilidad de uso que permite, grado de personalización y destacable autocompletado concretamente, en aplicaciones PHP [34].

XAMPP



XAMPP es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener que acceder a internet, integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, un servidor de FTP FileZilla, el popular administrador de base de datos escrito en PHP, MySQL, entre otros módulos.

Ilustración 8. Logo XAMPP

¿Cuáles son las ventajas y desventajas de utilizar esta herramienta?

Xampp es una herramienta muy práctica que nos permite instalar el entorno MySQL, Apache y PHP, suficiente para empezar proyectos web o revisar alguna aplicación localmente. Además, trae otros servicios como servidor de correos y servidor FTP.

La mayor ventaja de Xampp es que es muy fácil de instalar y las configuraciones son mínimas o inexistentes, lo cual nos ahorra bastante tiempo. Sin embargo, hay ocasiones en que es mejor dejar atrás la comodidad por las siguientes razones [35]:

- No soporta MySQL desde la consola.
- Xampp trae PhpMyAdmin para administrar las bases de datos de MySQL, sin embargo, para tareas más específicas es mejor utilizar la consola (línea de comandos) y Xampp no la soporta.
- Xampp trae las últimas versiones de las aplicaciones que instala, sin embargo, cuando pasa el tiempo y salen nuevas versiones de las mismas, no queda otra salida que reinstalar todo Xampp.
- Xampp además presenta muchas dificultades a la hora de configurar aplicaciones de terceros.

Docker



Ilustración 9. Logo Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del *kernel* Linux, tales como cgroups y espacios de nombres (*namespaces*) para permitir que "contenedores" independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

¿Qué es Docker?

La idea detrás de Docker es crear contenedores ligeros y portables para las aplicaciones software que puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina tenga por debajo, facilitando así también los despliegues [1].

¿Y qué diferencia a Docker respecto a una máquina virtual?

Digamos que el contenedor de Docker toma los recursos más básicos, que no cambian de un ordenador a otro del sistema operativo de la máquina en la que se ejecuta y gracias a esto, es mucho más ligero que una máquina virtual. Y los aspectos más específicos del sistema que pueden dar más problemas a la hora de llevar el software de un lado a otro, se meten en el interior del contenedor.

CAPÍTULO 3.- INGENIERÍA DEL SOFTWARE.

En este capítulo se va a describir en qué consiste la ingeniería del software, qué etapas tiene, definición de los diversos tipos de requisitos y el modelo que se ha elegido para realizar el desarrollo del proyecto, así como sus etapas y los motivos que me han llevado a tomar esa elección.

3.1 Introducción a la ingeniería del software.

La **Ingeniería del Software** [2] es una disciplina o área de la Informática que ofrece métodos y técnicas para desarrollar y mantener **software** de calidad que resuelven problemas de todo tipo. Existen varias definiciones sobre esta ciencia de la computación que permiten describir este proceso, la más ajustada bajo mi criterio es la siguiente: “Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software.”

3.2 Etapas

El proceso de ingeniería del software [3][4] consta de 7 etapas, las 5 primeras son fundamentales, mientras que la 6 está (o debe estar) presente en cada una de las etapas anteriores.

Estas etapas son:

Análisis de requerimientos: Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios. Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible.

Especificación: Es la tarea de describir detalladamente el software a ser escrito, de una forma rigurosa. Se describe el comportamiento esperado del software y su interacción con los usuarios y/o otros sistemas.

Diseño y arquitectura: Determinar cómo funcionará de forma general sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la segunda etapa también conocidas como las *entidades de negocio*. Generalmente se realiza en base a diagramas que permitan describir las interacciones entre las entidades y su secuenciado.

Programación: Se traduce el diseño a código. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados «tangibles». No necesariamente es la etapa más larga ni la más compleja, aunque una especificación o diseño incompletos/ambiguos pueden exigir que, tareas propias de las etapas anteriores se tengan que realizarse en esta.

Prueba: Consiste en comprobar que el software responda/realice correctamente las tareas indicadas en la especificación. Es una buena praxis realizar pruebas a distintos niveles (por ejemplo, primero a nivel unitario y después de forma integrada de cada componente) y por equipos diferenciados del de desarrollo (pruebas cruzadas entre los programadores o realizadas por un área de test independiente).

Documentación: Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema. Las tareas de esta etapa se inician ya en la primera fase, pero sólo finalizan una vez terminadas las pruebas.

Mantenimiento: En esta etapa se realizan un mantenimiento correctivo (resolver errores) y un mantenimiento evolutivo (mejorar la funcionalidad y/o dar respuesta a nuevos requisitos).



Ilustración 10. Fases de la ingeniería del software [43]

3.3 Definición de Requisitos

La ingeniería de requisitos es el conjunto de actividades y tareas del proceso de desarrollo de *sistemas software* que tiene como **objetivos**:

1. Definir, con la mejor calidad posible, las características de un sistema software que satisfaga las *necesidades de negocio* de clientes y usuarios y que se integre con éxito en el entorno en el que se explote. La definición de dicho sistema se realiza mediante lo que se conoce como una *especificación de requisitos*.
2. Gestionar las *líneas base* y las *peticiones de cambios* que se vayan produciendo en la especificación de requisitos, manteniendo la *trazabilidad* entre los requisitos y otros productos del desarrollo.

La **especificación de requisitos de software (ERS)** es una descripción completa del comportamiento del sistema que se va a desarrollar.

Está dirigida tanto al cliente como al equipo de desarrollo. El lenguaje utilizado para su redacción debe ser informal, de forma que sea fácilmente comprensible para todas las partes involucradas en el desarrollo.

Una buena especificación de requisitos de software se basa en estos principios [10]:

- *Completa*. Todos los requerimientos deben estar reflejados en ella y todas las referencias deben estar definidas.
- *Consistente*. Debe ser coherente con los propios requerimientos y también con otros documentos de especificación.
- *Inequívoca*. La redacción debe ser clara de modo que no se pueda mal interpretar.
- *Correcta*. El software debe cumplir con los requisitos de la especificación.
- *Trazable*. Se refiere a la posibilidad de verificar la historia, ubicación o aplicación de un ítem a través de su identificación almacenada y documentada.
- *Priorizable*. Los requisitos deben poder organizarse jerárquicamente según su relevancia para el negocio y clasificándolos en esenciales, condicionales y opcionales.
- *Modificable*. Aunque todo requerimiento es modificable, se refiere a que debe ser fácilmente modificable.
- *Verificable*. Debe existir un método finito sin costo para poder probarlo.

3.3.1 Requisitos funcionales

Son aquellos que definen una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Los requisitos de comportamiento para cada requisito funcional se muestran en los casos de uso [11].

3.3.2 Requisitos no funcionales

Son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares a aplicar en el mismo. Son propiedades o cualidades que el producto debe tener [12].

3.3.3 Requisitos de Calidad:

Un modelo de calidad define una serie de características de calidad que debe cumplir un determinado producto [13].

3.3.4 Restricciones

Limitaciones para el producto, las cuales pueden restringir el sistema en algún ámbito concreto tanto del entorno como de la implementación del mismo.

A diferencia de los requisitos funcionales o de calidad es que las restricciones no se implementan, simplemente acotan la posible solución bajo unos márgenes preestablecidos

3.4 Modelo de desarrollo software

Modelo Incremental

El modelo incremental de gestión de proyectos tiene como objetivo un crecimiento progresivo de la funcionalidad. Es decir, el producto va evolucionando con cada una de las entregas previstas (incremento o versión) hasta que se amolda a lo requerido por el cliente o destinatario.

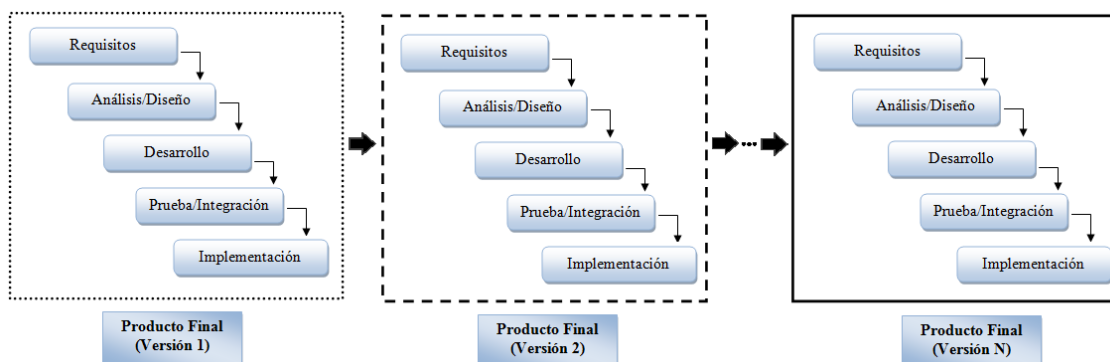


Ilustración 11. Modelo incremental

El modelo incremental utiliza el modelo en cascada dentro de cada incremento, cada fase de este modelo se encarga de [5]:

Análisis de requisitos del software

En esta fase se analizan las necesidades de los usuarios finales del *software* para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se debe **consensuar** todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del *software* de una manera.

Diseño del sistema

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Descripción del diseño del software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

Diseño del programa

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario, así como también los análisis necesarios para saber qué herramientas usar en la etapa de Codificación.

Codificación

Es la fase en donde se implementa el código fuente, haciendo uso de prototipos, así como de pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

Pruebas

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente. Se buscan sistemáticamente y se corrigen todos los errores antes de ser entregado al usuario final.

Implementación o Verificación del programa

Es la fase en donde el usuario final o el cliente ejecuta el sistema, y se asegura que cubra sus necesidades.

3.5 Elección de desarrollo software

Tras estudiar varios modelos de desarrollo software he decidido que el modelo que mejor se adapta al proyecto que se me presenta es el uso del **modelo incremental** por las siguientes razones:

- Si hubiera algún cambio respecto a alguno de los requisitos recogidos o se incorporaran nuevos, este modelo es ideal ya que es propicio a sufrir cambios.

- Contar con incrementos operativos desde el inicio del desarrollo hace que esta opción sea más atractiva que un modelado en cascada o lineal a la hora de reflejar los cambios (si los hubiere) de requisitos.
- El modelo es adaptable a las necesidades que vayan surgiendo en el proyecto.

CAPÍTULO 4.- ANÁLISIS

4.1 Especificación de requisitos

En este apartado se recogen todos los requisitos asociados al proyecto.

Requisitos Funcionales

1.- Gestión de usuarios:

- **Registro de usuario.**
 - El usuario puede registrarse en la aplicación.
- **Alta de usuario.**
 - El administrador podrá aceptar en el sistema a un usuario.
- **Modificar usuario.**
 - El administrador podrá modificar los datos asociados a un usuario.
- **Eliminar usuario.**
 - El administrador podrá eliminar los datos asociados a un usuario.
- **Consultar datos de usuario.**
 - El administrador podrá consultar los datos asociados a un usuario.

2.- Gestión de asignaturas:

- **Añadir nueva asignatura.**
 - El administrador puede añadir nuevas asignaturas al sistema.
- **Eliminar asignatura.**
 - El administrador puede eliminar asignaturas del sistema.
- **Matriculación en asignatura.**
 - El sistema debe permitir a un usuario matricularse en una asignatura.
- **Darse de baja en asignatura.**
 - El sistema debe permitir a un usuario darse de baja de una asignatura.
- **Añadir nuevos temas.**
 - El administrador puede añadir nuevos temas a las asignaturas.

- **Eliminar temas.**
 - El administrador puede eliminar temas de las asignaturas.
- **Acceso al contenido.**
 - Cualquier usuario matriculado en una asignatura podrá acceder al contenido de la misma

3.- Gestión de *flashcards*

- **Añadir *flashcard*.**
 - El profesor podrá añadir una *flashcard* y registrar sus datos.
- **Eliminar *flashcard*.**
 - El profesor podrá eliminar una *flashcard*
- **Consultar datos de una *flashcard*.**
 - El profesor podrá consultar los datos de una *flashcard*
- **Modificar datos de una *flashcard*.**
 - El profesor podrá modificar los datos de una *flashcard*.
- **Realizar *flashcard*.**
 - El estudiante podrá realizar un taco de *flashcards* asociado a un tema tantas veces como el estudiante precise.
- **Obtener resultados**
 - El estudiante obtendrá al finalizar el taco de *flashcards* asociado a un tema información acerca de los resultados obtenidos, como puede ser la tasa de acierto o el score global conseguido.

4.- Gestión de recursos audio visuales

- **Añadir recurso.**
 - El profesor podrá añadir recursos visuales a un tema de una asignatura.
- **Modificar recurso.**
 - El profesor podrá modificar recursos visuales.
- **Eliminar recurso.**
 - El profesor podrá eliminar recursos visuales.
- **Consultar recursos.**

- Cualquier usuario (matriculado en la asignatura) tendrá acceso a la visualización de los recursos visuales.

5.- Gestión de Estadísticas

Se podrán consultar distintas agrupaciones estadísticas, en forma de tabla y en forma gráfica, asociada a una asignatura tanto a nivel de agregación global como a nivel específico por tema, además de poder descargar la información en formato CSV.

- **Consultar Estadísticas.**
 - El profesor podrá consultar las estadísticas de una asignatura y de un tema concreto tanto a nivel tabla estadística como a nivel gráfico.
- **Descarga en formato CSV.**
 - El profesor podrá descargar un archivo formato CSV con la información presentada en la tabla estadística.

Requisitos no Funcionales

- Concurrencia: el profesorado podrá trabajar simultáneamente para introducir información al sistema, así como los alumnos podrán acceder a los recursos visuales de sus asignaturas y realizar las *flashcards* adjuntas.
- Autenticación: se controlará el acceso a través de autenticación a la plataforma.
- El sistema deberá soportar un volumen de datos mínimo de 60 usuarios por asignatura registrada para ofrecer una experiencia correcta a los usuarios registrados del uso de la aplicación.
- El usuario no debe de dedicar más de 15 minutos al aprendizaje de uso de la plataforma.
- El sistema deberá ofrecer recursos visuales y *flashcards* adjuntos a dichos recursos, los cuales irán asociados a una única asignatura.
- La plataforma debe tener interfaces intuitivas, de fácil uso y atractivas visualmente.

- El sistema debe facilitar mensajes de información y error al usuario siempre que sea necesario
- El sistema debe de asegurar protección a los datos, de los cuales el acceso sea restringido.
- El sistema deberá funcionar correctamente en todos los navegadores.
- Realizar una aplicación *responsive design*.

Requisitos de Calidad

- La aplicación debe estar disponible el 100% del tiempo.
- El usuario podrá visualizar información respecto a la aplicación web, así como contacta con nosotros, políticas de privacidad o direcciones de correo electrónico del administrador del sistema.
- El usuario debe poder acceder a la aplicación independientemente del dispositivo que se utilice para el acceso a la misma.
- La aplicación no funcionará offline.
- El sistema debe contar con unos datos de prueba para poder testear la correcta funcionalidad completa del sistema implementado.

Restricciones

- El tiempo máximo de desarrollo por incremento será de 30 días.
- Todo el software usado en el desarrollo del proyecto debe ser software libre o con acuerdos de licencias académicas con la Universidad de Jaén.

4.2 Definición de Actores

Administrador



Ilustración 22. Logo Admin

Usuario que administra el sistema, con acceso a las funcionalidades tanto de estudiantes como de profesores, con la capacidad de dotar de un rol a los usuarios (estudiante o profesor) así como la capacidad de gestionar toda la aplicación.

Profesor



Actor que se ha registrado previamente en la aplicación e interactúa con las funcionalidades de su propio rol. Capacidad de añadir recursos visuales y *flashcards* a una asignatura a la cual esté asignado, así como consultar las estadísticas asociadas a la asignatura que imparte.

Ilustración 23. Logo Profesor

Estudiante



Actor que se ha registrado previamente en la aplicación y que el administrador ha admitido en la plataforma, e interactúa con las funcionalidades de su propio rol. Capacidad de reproducir recursos visuales y realizar *flashcards* de una asignatura a la cual esté registrado o invitado.

Ilustración 24. Logo Estudiante

Estudiante no aceptado



Actor que se ha registrado previamente en la aplicación y que el administrador no ha admitido en la plataforma. No tiene capacidad de interactuar con la plataforma más allá de Iniciar y Cerrar Sesión.

Ilustración 25. Logo estudiante no aceptado

4.3 Modelos de casos de uso

Un Diagrama de casos de uso es un modelo basado en el escenario representado mediante notación UML donde se muestra la relación entre los actores y los casos de uso en un sistema

Gestión de asignaturas

En este caso de uso de la ilustración 26 se muestra la relación de los distintos actores (usuario aceptado en la aplicación, profesor y administrador) con los casos de uso asociados a la gestión de las asignaturas.

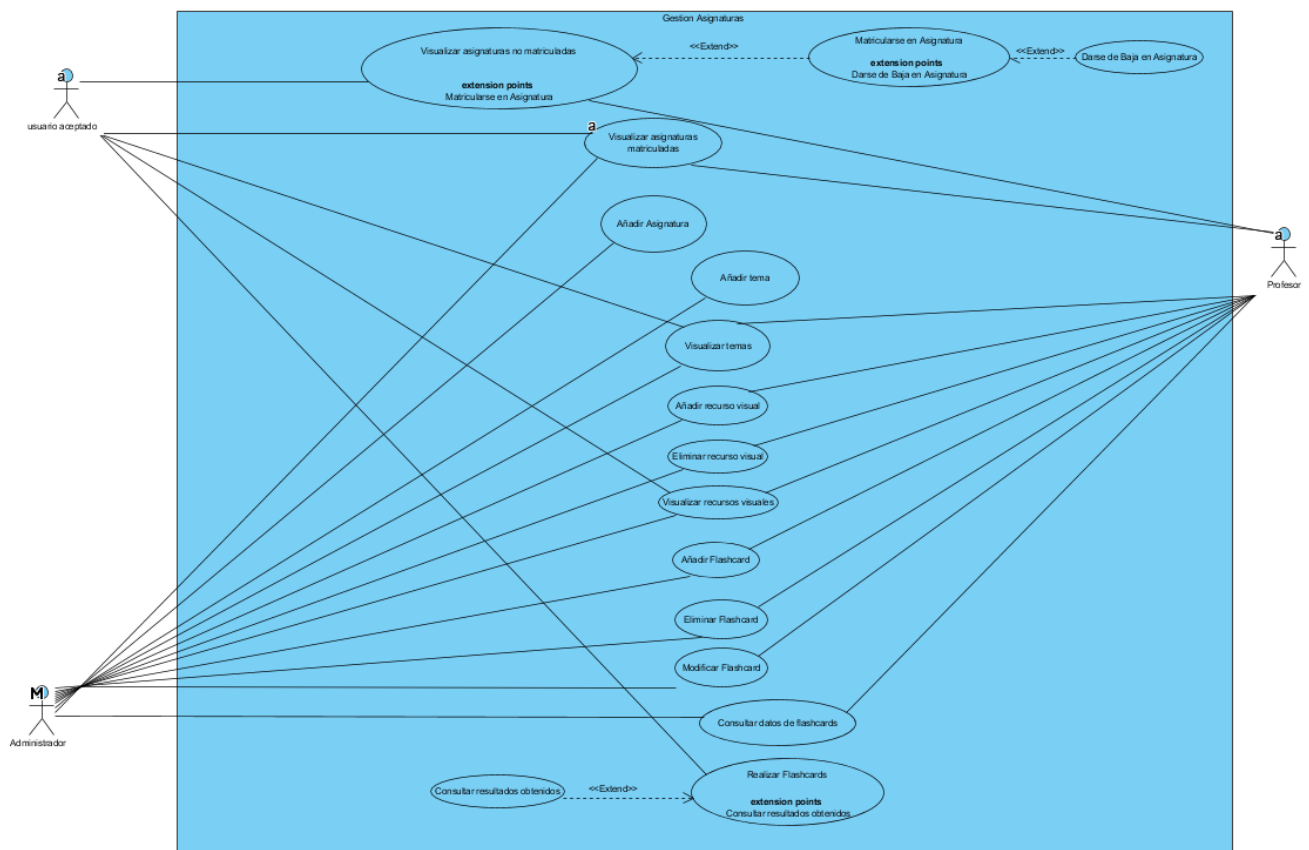


Ilustración 26. Casos de uso: Gestión de asignaturas

Gestión de usuarios

En el caso de uso de la ilustración 27 se muestra la relación de los distintos actores (usuario no registrado, usuario aceptado en la aplicación y administrador) con los casos de uso asociados a la gestión de usuarios

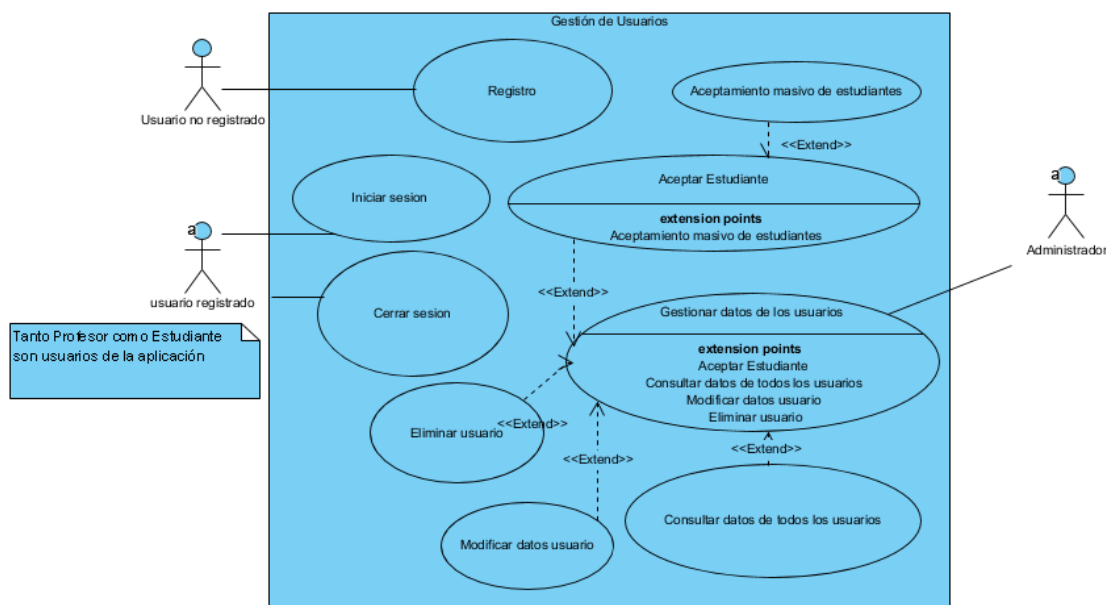


Ilustración 27. Casos de uso: Gestión de usuarios

Gestión de Estadísticas

En el caso de uso de la ilustración 28 se muestra la relación del Profesor con los casos de uso asociados a la gestión de estadísticas

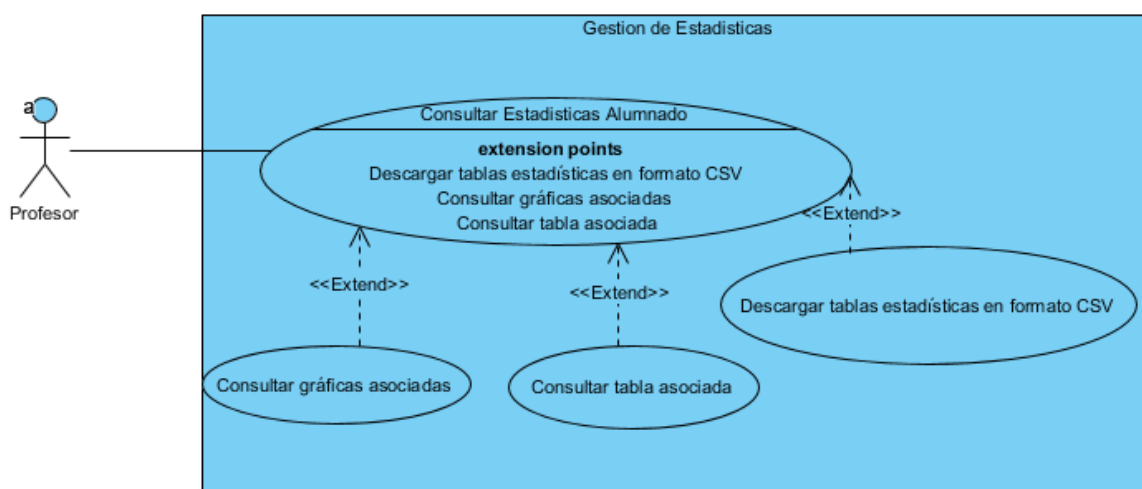


Ilustración 28. Casos de uso: Gestión de estadísticas

4.4 Narrativas de los casos de uso

GESTIÓN DE USUARIOS

CASO DE USO	Registro
Resumen	Registro de un usuario en el sistema
Actores	Todos
Prerrequisito	Que el correo electrónico no esté previamente registrado.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario usa la opción 'Registro'. 2. El sistema muestra el formulario de registro. 2. El usuario rellena los datos requeridos 3. El usuario usa la opción 'Registrar' 4.El sistema comprueba si el formulario es correcto y completo. 5. El sistema registra al usuario. 6. El sistema muestra la confirmación del registro exitoso.
Propósito	Registrar una cuenta en el sistema
Alternativa 4.A El registro es incorrecto o incompleto	<ol style="list-style-type: none"> 4.A.1. El sistema muestra el formulario de registro junto al error que se detecta. 4.A.2 Volver al paso 2.
Excepciones	

CASO DE USO	Identificación
Resumen	Acceso a la aplicación
Actores	Todos
Prerrequisito	Estar registrado en el sistema

Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario usa la opción 'Iniciar Sesión' 2. El sistema muestra el formulario de inicio de sesión 3. El usuario rellena los datos requeridos 4. El usuario usa la opción 'Iniciar Sesión' 5. El sistema comprueba si el formulario es correcto y completo. 6. El sistema inicia la sesión del usuario. 7. El sistema muestra el index del usuario.
Propósito	Iniciar sesión en la aplicación
Alternativa 4.A El inicio de sesión es incorrecto o incompleto	<ol style="list-style-type: none"> 4.A.1. El sistema muestra el formulario de inicio de sesión junto al error que se detecta. 4.A.2 Volver al paso 2.
Excepciones	

CASO DE USO	Cerrar sesión
Resumen	Cerrar sesión en la aplicación
Actores	Todos
Prerrequisito	Tener la sesión iniciada
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario usa la opción 'Cerrar Sesión' 2. El sistema cierra la sesión del usuario. 3. El sistema muestra el <i>index</i> de bienvenida.
Propósito	Cerrar sesión en la aplicación.
Alternativas	
Excepciones	

GESTIÓN DE ASIGNATURAS

CASO DE USO	Añadir asignatura
Resumen	Añadir una nueva asignatura al sistema.
Actores	Administrador
Prerrequisito	
Operaciones Básicas	<ol style="list-style-type: none"> 1. El administrador usa la opción ‘Mis Asignaturas’ en la barra de navegación 2. El sistema muestra las Asignaturas asociadas al usuario. 3. El administrador usa la opción ‘Nueva Asignatura’ 4. El sistema muestra el formulario para añadir una nueva asignatura. 5. El administrador rellena el formulario. 6. El administrador usa la opción ‘Añadir’ 7. El sistema añade la asignatura. 8. El sistema muestra la nueva asignatura añadida.
Propósito	Añadir una nueva asignatura a la lista de asignaturas ofertadas.
Alternativas	
Excepciones	

CASO DE USO	Darse de alta en asignatura
Resumen	Registrarse en una asignatura
Actores	Todos
Prerrequisito	Tener una sesión iniciada con rol de, al menos, estudiante activo.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario usa la opción ‘Asignaturas no matriculadas’. 2. El sistema muestra las asignaturas no matriculadas para el usuario. 3. El usuario usa la opción la opción ‘Matricular’ asociada a una asignatura. 4. El sistema matricula al usuario en la asignatura seleccionada.

	5. El sistema muestra las asignaturas matriculadas para el usuario.
Propósito	Tener acceso a recursos y flashcards de una asignatura concreta.
Alternativas	
Excepción 2.A El usuario está matriculado en todas las asignaturas que se ofertan.	2.A.1 El sistema muestra un mensaje informativo de que no hay más asignaturas por matricular.

CASO DE USO	Darse de baja en asignatura
Resumen	Dejar de pertenecer a una asignatura
Actores	Todos
Prerrequisito	Tener una sesión iniciada con permisos, al menos, de estudiante activo.
Operaciones Básicas	1. El usuario usa la opción 'Asignaturas matriculadas'. 2. El sistema muestra las asignaturas matriculadas del usuario 3.El usuario selecciona la opción 'Dar de baja' asociada a una asignatura. 4. El sistema da de baja al usuario de la asignatura. 5. El sistema muestra la tabla de asignaturas matriculas actualizada.
Propósito	Eliminar la asignatura de tus asignaturas matriculadas.
Alternativas	
Excepción 2.A El usuario no está matriculado en ninguna asignatura.	2.A.1 El sistema muestra un mensaje informativo de que no hay asignaturas matriculadas.

GESTIÓN DE RECURSOS VISUALES

CASO DE USO	Añadir Recurso
Resumen	Añadir recurso al sistema
Actores	Profesor, Administrador
Prerrequisito	Estar matriculado en la asignatura a la cual queremos añadir un recurso
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario usa la opción ‘Añadir Recurso Visual’ 2. El sistema muestra formulario. 3. El usuario rellena todos los campos del formulario. 4. El usuario usa la opción Añadir. 5. El sistema registra el nuevo recurso en el sistema 6. El sistema muestra la tabla de recursos con el nuevo recurso añadido.
Propósito	Añadir nuevos recursos a un tema de una asignatura concreta.
Alternativas	

CASO DE USO	Eliminar Recurso
Resumen	Borrar recurso del sistema
Actores	Profesor, Administrador
Prerrequisito	Estar matriculado en la asignatura a la que pertenezca el recurso a eliminar.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario usa la opción ‘Recursos Visuales’ 2. El sistema muestra la tabla de recursos visuales asociados al tema. 3. El usuario selecciona la opción ‘Eliminar’ asociada al recurso que se desea eliminar 4. El sistema elimina el recurso del sistema. 5. El sistema muestra la tabla de recursos visuales.
Propósito	Eliminar un recurso previamente registrado en el sistema

Alternativas 4.A	4.A.1 El sistema muestra un mensaje informativo de que no hay recursos visuales asociados al tema.
Tras borrar el recurso visual, no queda ninguno que mostrar.	
Excepciones	

GESTIÓN DE *FLASHCARDS*

CASO DE USO	Añadir Flashcard
Resumen	Añadir Flashcard al sistema
Actores	Profesor, Administrador
Prerrequisito	Estar matriculado en la asignatura a la cual queremos añadir una Flashcard
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario accede a la gestión de Flashcards del tema al cual se quiera añadir. 2. El usuario rellena todos los campos del formulario ofrecido en la interfaz. 3.El usuario usa la opción Añadir. 4.El sistema registra la nueva Flashcard en el sistema 5. El sistema muestra la tabla de flashcards con la nueva Flashcard añadida
Propósito	Añadir nuevas flashcards a un tema de una asignatura concreta.
Alternativas	
Excepciones	

CASO DE USO	Modificar Flashcard
Resumen	Modificar Flashcard del sistema
Actores	Profesor, Administrador
Prerrequisito	Estar matriculado en la asignatura a la que pertenezca la Flashcard a modificar
Operaciones Básicas	<ol style="list-style-type: none"> 1. El usuario accede a la gestión de Flashcards del tema al que pertenezca la Flashcard a modificar. 2. El usuario selecciona la Flashcard a editar de la tabla mostrada.

	<p>3. El usuario rellena o modifica los campos deseados.</p> <p>4. El usuario usa la opción Modificar.</p> <p>5. El sistema registra la Flashcard modificada en el sistema.</p> <p>6. El sistema muestra la tabla de flashcards con la nueva Flashcard modificada</p>
Propósito	Modificar una Flashcard previamente añadida al sistema.
Alternativa 4.A. El usuario selecciona una pregunta y en vez de modificarla, añade una nueva Flashcard con las modificaciones realizadas, pero dejando la Flashcard original en el sistema	<p>4.A.1 El usuario usa la opción Añadir.</p> <p>4.A.2 El sistema registra la nueva Flashcard en el sistema.</p> <p>4.A.3 El sistema muestra la tabla de flashcards con la nueva Flashcard añadida.</p>
Excepciones	

CASO DE USO	Eliminar Flashcard
Resumen	Borrar Flashcard del sistema
Actores	Profesor, Administrador
Prerrequisito	Estar matriculado en la asignatura a la que pertenezca la Flashcard a eliminar
Operaciones Básicas	<p>1. El actor selecciona la Flashcard a eliminar de la tabla.</p> <p>2. El sistema carga los datos de la Flashcard seleccionada</p> <p>3. El actor pulsa en 'Eliminar'.</p> <p>4. El sistema realiza un borrado en cascada de la flashcard e información asociada a ella.</p> <p>5. El sistema refresca la tabla de Flashcard registradas en el sistema</p>
Propósito	Eliminar una Flashcard previamente registrada en el sistema
Alternativas	
Excepciones	

CAPÍTULO 5.- PLANIFICACIÓN

Se expondrá la planificación llevada a cabo a lo largo del proyecto junto a los actores que harán uso de la aplicación.

Se desarrollará el proyecto con metodología ágil, con una jornada de trabajo de 5 horas desde el 1 de noviembre de 2018 hasta el 1 de Julio de 2019 (242 días).

5.1 Estimación de requisitos

A continuación, se presentan los métodos de estimación del esfuerzo (Tallas) y de priorización (MOSCOW) de requisitos elegidos tras el estudio de varias opciones a la hora de realizar esta tarea.

Método MOSCOW como metodología de priorizar requisitos

La elección e implementación de un nuevo software puede ser todo un trabajo. Las empresas que quieren tomar una buena decisión, deben saber exactamente lo que pueden esperar del software. **Establecer las prioridades** es primordial, pero este proceso no siempre es fácil. Por ello, Daig Clegg en 1994 ideó el método MoSCoW. Las consonantes de este acrónimo representan los **diferentes tipos de deseos y requisitos** que una empresa debería distinguir: *Must-have* (debe-tener); *Should-have* (debería-tener), *Could-have* (podría-tendría) y *Won't-have/Would-have* (no-tendría) [14].

***Must-have* (Debe-tener)**

Los *must-have* son los **requisitos mínimos** que un software tiene que cumplir. Sin el cumplimiento de estos requisitos, se considera que el proyecto no ha tenido éxito. Por ejemplo, si una empresa se interesa primero por un sistema ERP [36] porque su programación manual de la producción provoca errores semanalmente y, por lo tanto, conlleva pérdidas de tiempo, el módulo de planificación de la producción [37] será un *must-have*.

Should-have (Debería-tener)

Los *should-have* en un proceso de orientación, son aquellas **funcionalidades que resultan altamente deseables**. Sin embargo, el software también puede demostrar su valía sin estas funcionalidades. La adición de los *should-have* generalmente se encargan de conseguir un mayor retorno de la inversión ROI (Return On Investment) [38]. También puede tratarse de funcionalidades que no tienen una prioridad tan alta en el momento, pero que será necesaria en un periodo de tiempo relativamente corto. Una empresa que ahora puede convertir los pedidos por correo electrónico en órdenes de producción de forma relativamente rápida puede seguir considerando la integración del correo electrónico como una *should-have* de proyección de crecimiento.

Could-have (podría-tendría)

A los *could-have* también se les conoce como *nice-to-have*. Estas son cualidades que gustaría añadir si entran **dentro del rango de tiempo y dinero**. Sin embargo, cuando suponen un esfuerzo extra, no tiene sentido contemplarlas. Por ejemplo, en el caso de una empresa de producción con una clientela baja de B2B(*business-to-business*), un *could-have* podría ser la integración de EDI (Electronic Data Interchange) [39]. El intercambio directo de datos puede acelerar el proceso de pedido y facturación, pero trae consigo un crecimiento realmente significativo de los pedidos.

Won't have/ Would-have (no-tendría)

Los *won't have* son más bien **deseos utópicos**, o deseos que en el **momento de la orientación no merecen aún la pena invertir en ellos**. Por ejemplo, esto puede ser involucrar la recogida de pedidos [40] a través de realidad aumentada. Es una tecnología muy interesante, pero para la mayoría de las empresas este sólo supone un coste enorme. A los *won't have* también se les conoce como *would-have*, porque una empresa podría implementarlos bajo ciertas condiciones en el futuro.

Estimación basada en Tallas

Debido a su carácter relativo, la estimación basada en tallas permite disminuir el error asociado a otras técnicas de estimación basadas en el cálculo del esfuerzo en unidades temporales reales (días, horas, ...). Como ejemplo basta pensar en lo complicado que es conseguir que dos desarrolladores (con habilidades variadas y un grado de experiencia diferente) den una misma estimación para una misma H.U. Por ejemplo, uno de los desarrolladores podría estimar la H.U. (Historias de Usuario) en 2 días, y el otro no darle más de 2 horas. Volviendo sobre el carácter relativo de la estimación basada en tallas, la potencia de este tipo de estimaciones se basa en que al comparar dos H.U. y determinar cuál es mayor o menor que la otra, incluso en el caso de que se proporcione una estimación de cuanto más grande o más pequeña resulta la una respecto a la otra, raramente existirá desacuerdo entre los desarrolladores. Y si existe desacuerdo, entonces podremos estar seguros de que la H.U. no está suficientemente clara para poder ser estimada. Para aplicar esta técnica de estimación relativa se suelen utilizar las habituales tallas de las camisetas: muy pequeña (XS), pequeña (S), mediana (M), grande (L) y muy grande (XL). Hay que tener en cuenta que el uso de otras medidas intermedias, en vez de contribuir de manera positiva en la estimación, podría provocar que el método resultase excesivamente complejo [15].

Organización de los requisitos previamente obtenidos en una tabla junto a la prioridad siguiendo el método **MOSCOW** y la duración del requisito con el método de estimación de tiempos '**Talla de camiseta**'.

La conversión de Tallas a días de trabajo es la siguiente:

'S' 5 días (25 horas).

'M' 10 días (50 horas).

'L' 15 días (75 horas).

'XL' 20 días (100 horas).

Requisito alto nivel	Sub-requisitos	Prioridad MOSCOW	Estimación tiempos TALLAS
GESTIÓN DE USUARIOS	Registro de usuario	M	M
	Aceptar usuario	M	S
	Modificar usuario (usuario)	W	M
	Eliminar usuario	S	L
	Consultar datos de usuarios (admin)	M	M
	Modificar datos de usuario (admin)	M	S
Prioridad y estimación de tiempos media		M	M
GESTIÓN DE ASIGNATURAS	Añadir nuevas asignaturas	M	S
	Eliminar asignatura	W	XL
	Matricularse en asignatura	M	S
	Darse de baja de asignatura	M	S
	Añadir tema a asignatura	M	S
	Eliminar tema de asignatura	W	L
	Consultar contenido de asignatura	M	L
Prioridad y estimación de tiempos media		M	L
GESTIÓN DE FLASHCARDS	Añadir Flashcard	M	S
	Eliminar Flashcard	S	S
	Consultar datos de Flashcard	M	XL
	Modificar datos de Flashcard	M	S
	Realizar taco de flashcards	M	M
	Obtener estadísticas propias	S	S
Prioridad y estimación de tiempos media		M	M
GESTIÓN DE RECURSOS	Añadir Recurso	M	S
	Modificar Recurso	W	M
	Eliminar Recurso	S	S
	Consultar Recursos	M	S
Prioridad y estimación de tiempos media		S	S
GESTIÓN DE ESTADÍSTICAS	Crear distintos tipos de agrupaciones estadísticas	M	M
	Consultar datos estadísticos (tabla)	M	S
	Consultar datos estadísticos (gráfico)	S	M
	Descarga de estadísticas formato CSV	C	S
Prioridad y estimación de tiempos media		S	M

Tabla 1. Requisitos, prioridad y estimación de los mismos.

A continuación, se muestra la tabla de incrementos con sus respectivos requisitos a completar.

Incremento	Requisitos	Prioridad MOSCOW	Estimación tiempos TALLAS	Total
Incremento 1	Registro de usuario	M	M	30
	Aceptar usuario	M	S	
	Consultar datos de usuarios (admin)	M	M	
	Modificar datos de usuario (admin)	M	S	
Incremento 2	Eliminar Usuario	S	L	30
	Añadir nuevas asignaturas	M	S	
	Añadir tema a asignatura	M	S	
	Matricularse en asignatura	M	S	
Incremento 3	Darse de baja en asignatura	M	S	25
	Consultar contenido de asignatura	M	L	
	Añadir Flashcard	M	S	
Incremento 4	Consultar datos de Flashcard	M	XL	30
	Modificar datos de Flashcard	M	S	
	Eliminar Flashcard	S	S	
Incremento 5	Realizar taco de flashcards	M	M	30
	Obtener estadísticas propias	S	S	
	Añadir Recurso	M	S	
	Consultar Recursos	M	S	
	Eliminar Recursos	S	S	
Incremento 6	Crear distintos tipos de agrupaciones estadísticas	M	M	30
	Consultar datos estadísticos (tabla)	M	S	
	Consultar datos estadísticos (gráfico)	S	M	
	Descarga de estadísticas formato CSV	C	S	

Tabla 2. División de Incrementos

La planificación final del proyecto es la siguiente:

Tarea	Duración	F. Inicio	F. Fin
Preparación de entornos de desarrollo y herramientas	22	01/11/18	22/11/18
Estudio de herramientas y tecnologías y elección de ellas.	7	01/11/18	07/11/18
Estudio de lenguajes de programación y elección de ellas.	12	08/11/18	19/11/18
Preparación e instalación de herramientas y lenguajes elegidos.	3	20/11/18	22/11/18
Búsqueda bibliográfica	7	23/11/18	29/11/18
Ingeniería Software	182	29/11/18	30/05/19
Adquisición de requisitos	2	29/11/18	30/11/18
Incremento 1	30	01/12/18	30/12/18
Análisis	8	01/12/18	08/12/18
Diseño	8	09/12/18	16/12/18
Implementación	12	17/12/18	28/12/18
Pruebas	2	29/12/18	30/12/18
Incremento 2	30	01/01/19	31/01/19
Análisis	8	01/01/19	08/01/19
Diseño	6	09/01/19	14/01/19
Implementación	10	15/01/19	24/01/19
Pruebas	6	25/01/19	30/01/19
Incremento 3	25	01/02/19	25/02/19
Análisis	4	01/02/19	04/02/19
Diseño	4	05/02/19	08/02/19
Implementación	11	09/02/19	19/02/19
Pruebas	6	20/02/19	25/02/19
Incremento 4	30	01/03/19	31/03/19
Análisis	3	01/03/19	03/03/19
Diseño	7	04/03/19	10/03/19
Implementación	12	11/03/19	22/03/19
Pruebas	8	23/03/19	30/03/19
Incremento 5	30	01/04/19	30/04/19
Análisis	4	01/04/19	04/04/19
Diseño	5	05/04/19	09/04/19
Implementación	13	10/04/19	22/04/19
Pruebas	8	23/04/19	30/04/19
Incremento 6	30	01/05/19	31/05/19
Análisis	4	01/05/19	04/05/19
Diseño	6	05/05/19	10/05/19
Implementación	10	11/05/19	20/05/19
Pruebas	10	21/05/19	30/05/19
Documentación	231	01/11/18	19/06/19
Memoria	231	01/11/18	19/06/19
Manuales	18	01/06/19	18/06/19
Duración total	231	01/11/18	20/06/19

Tabla 3. Planificación temporal de los incrementos

5.2 Diagrama de Gantt

El **diagrama de Gantt** [16] es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Diagrama de Gantt Simplificado

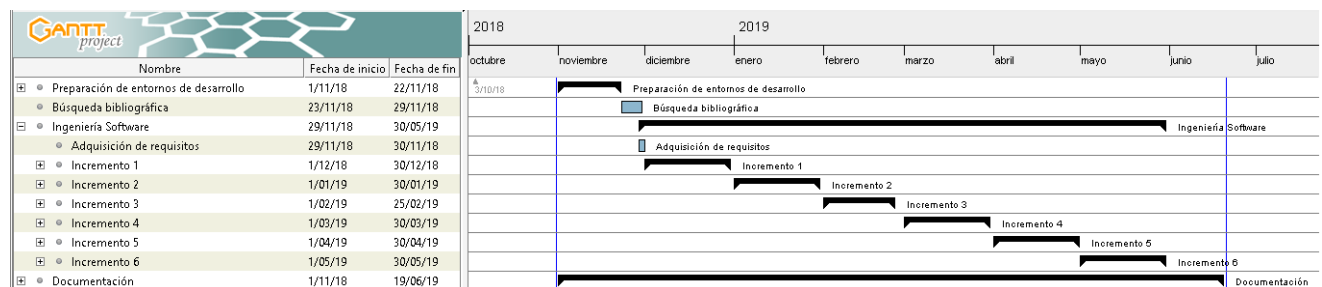


Ilustración 29. Diagrama de Gantt simplificado

Diagrama de Gantt Desarrollado

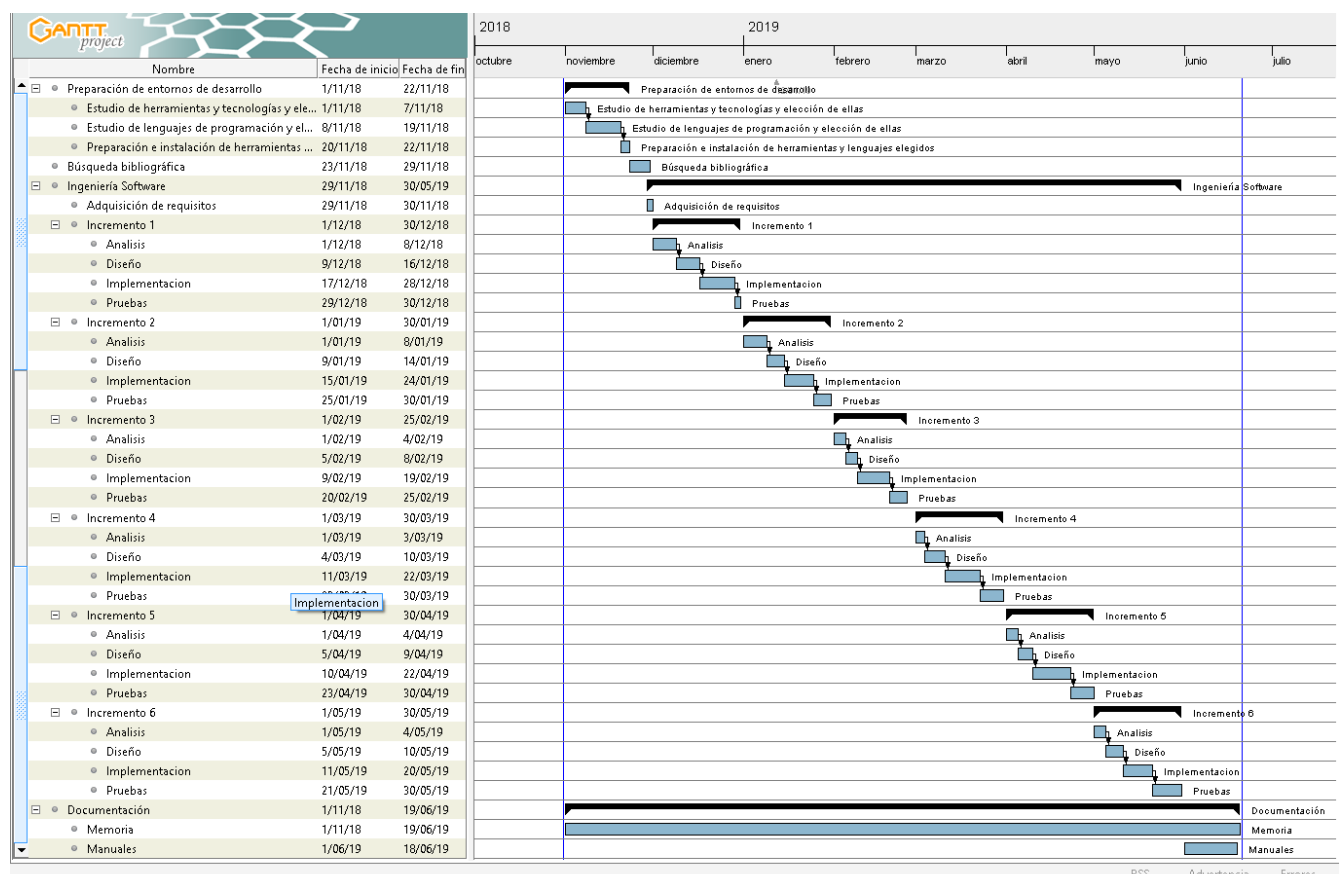


Ilustración 30. Diagrama de Gantt desarrollado

5.3 Estimación de costes

Recursos

En esta parte se describe los recursos utilizados, tanto hardware como software y coste de estos.

Recursos hardware

Para el desarrollo de este sistema se ha utilizado:

- Ordenador personal, Lenovo 80XR, con las siguientes características:
 - Intel Celeron 1.1 GHz
 - 4GB RAM
 - Disco duro de 500 GB
- Ordenador personal, IdeaCentre AIO 520-24IKU con las siguientes características:
 - Procesador: i3-6006U
 - 8 GB RAM
 - Disco duro de 1 TB

Recursos Software

- Sistema Operativo: Windows 10
- IDE: Netbeans 8.2 & Visual Studio Code
- Lenguajes de programación:
 - JavaScript
 - HTML
 - CSS
 - PHP

Costes

Licencias

Todas las licencias utilizadas en el desarrollo del proyecto han sido libres o a través de claves académicas, por lo tanto, el coste de las licencias ha sido de 0 €

Recursos hardware

Se utilizarán 2 equipos para llevar a cabo el proyecto



Ilustración 31. Equipo de torre

IdeaCentre AIO 520-24IKU (sobremesa)

Adquirido por 800 €



Ilustración 32. Equipo portátil

Lenovo 80XR (portátil)

Adquirido con un valor de 425 €.

Costes ordenadores

Estos equipos los usaremos durante el desarrollo del TFG (noviembre-junio) lo cual hace un total de 242 días.

Cada equipo tiene una vida útil de 5 años, o lo que es lo mismo, 1825 días, por lo tanto, cada equipo tendrá un coste de amortización.

Coste total = SUM (Coste de Amortización por equipo)

Coste Amortización = (Precio de adquisición x días de uso) / vida útil

Coste Amortización equipo sobre mesa:

$(800\text{euros} \times 242\text{días}) / 1825\text{días} = 106.08 \text{ €}$

Coste Amortización equipo portátil:

$(425\text{euros} \times 242\text{días}) / 1825\text{días} = 56.35\text{€}$

Coste total equipos amortizados = $106,08 + 56,35 = 162.43\text{€}$

Recursos humanos

Para este proyecto se miró el perfil de desarrollador *Full-Stack*, encargado tanto del Backend como del Frontend:

- Desarrollador *Full-Stack*:
 - Coste: 34000 € / Año, 2833 € / Mes.

El tiempo del desarrollo del proyecto ha durado un total de 8 meses, por lo tanto, el gasto en salarios sería de un total de 22664 €.

Alquiler de oficina

Para este proyecto se estima necesario el alquiler de una oficina donde físicamente llevar a cabo el trabajo. El coste de una oficina con un despacho en Jaén capital se estima que asciende a 400 € mensuales.

Consumo de luz y agua

El consumo de luz y agua de la oficina se estima en un coste de 125 € mensuales.

Conexión a internet

Será necesaria la contratación de una línea de fibra óptica que supone un gasto mensual de 17 € mensuales

Diversos consumibles

En el desarrollo funcional del trabajo es necesario auxiliarse de diverso material de oficina fungible (papel, bolígrafos, tóner, etc.). El coste total de los diversos consumibles utilizados asciende a 100 €.

Coste total

El coste total se calculará teniendo en cuenta tanto el hardware y el software, además del salario del desarrollador *full-stack*:

Descripción	Detalles	Coste
Licencias software	Software Libre	0 €
Hardware	Ordenadores	162.43 €
Desarrollador	2833€/mes x 8 meses	22664€
Alquiler de oficina	400€/mes x 8 meses	2400 €
Consumo de luz y agua	125€/mes x 8 meses	1000 €
Conexión a internet	17 €/mes x 8 meses	136 €
Diversos consumibles	Papel, tóner, bolígrafos...	100 €
TOTAL		26462.43 €

Tabla 4. Estimación de costes

CAPÍTULO 6.- DISEÑO.

El **diseño** [18] es la primera de las tres actividades técnicas que implica un proceso de **ingeniería de software**; estas etapas son **diseño**, codificación (Desarrollo e Implementación) y pruebas. El diseño del software es un proceso que cuenta con numerosos pasos.

En general esta fase de basa en establecer estructuras de datos, arquitectura general software, representación de interfaces e interacción entre las distintas partes del sistema a través de, por ejemplo, algoritmos. Realiza la función de conversión requisito – representación de software.

6.1 Diseño Arquitectónico

El diseño arquitectónico elegido para el desarrollo de la Aplicación Web es el Modelo – Vista – Controlador (MVC), dicho diseño utiliza 3 componentes (Vistas, *Models* y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación.

¿Cómo surge el patrón MVC?

El primer documento en que se habla de la separación de los elementos de un sistema software en un **modelo** de ordenador, un software **controlador** y una **vista** manipulable por el usuario surge en los **Xerox PARC** (*Palo Alto Research Center*) en 1978 en *The original MVC reports*, mencionándose un componen adicional, llamado **Editor**, que forma parte de la vista.

Durante el desarrollo de una aplicación (por parte de un científico visitante que pasaba una temporada en el PARC) se hizo patente la necesidad de buscar **una solución general** al problema que planteaba el control de un conjunto de datos grande y complejo por parte de un usuario. El objetivo era **adaptar el modelo de ordenador** de esos datos **al modelo mental** de la persona que debía usarlos, interponiendo los componentes software apropiados para ello.

La aplicación a desarrollar era un sistema de planificación y en él se empleaban cuatro metáforas: **thing**, **model**, **view** y **editor**, extrapolación del **objeto** de interés para el usuario, la **abstracción** de ese objeto en forma de datos en un ordenador, una o más posibles **representaciones** de dicho objeto y la **interfaz** entre el usuario y esas vistas, respectivamente.

Tras un proceso de refinamiento se definió el **modelo** como una representación del conocimiento con el que debe trabajar el sistema, la **vista** como una representación visual de ese modelo y se introduce el **controlador** como el enlace entre el usuario y el sistema. Estas ideas se plasman en un documento publicado el 10 de diciembre de 1979 con el título *Model-Views-Controllers*, naciendo el acrónimo MVC.

La primera implementación de MVC como patrón arquitectónico, más allá de la aplicación puntual que le dio origen, se efectuó en el propio PARC para la librería Smalltalk-80. A partir de ese momento comenzó su popularización y se extendió su uso, con independencia de plataformas, lenguajes de programación e incluso arquitectura del sistema a desarrollar [41].

Este diseño arquitectónico ayuda a facilitar la tarea del desarrollo, así como el posterior mantenimiento del sistema desarrollado a través de la reutilización de código y de la separación conceptual de la aplicación.

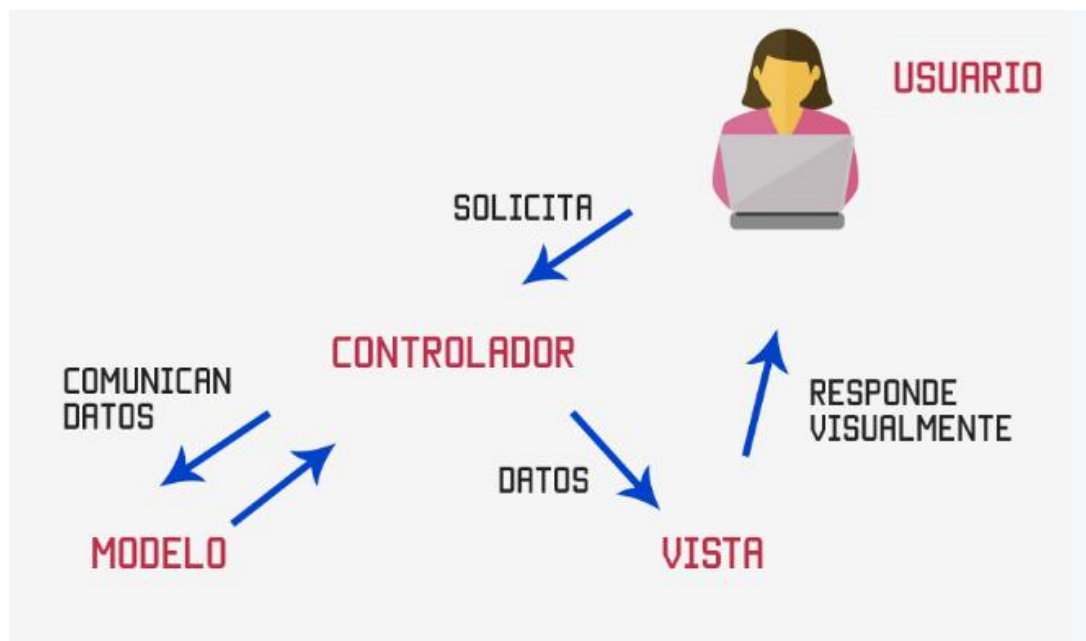


Ilustración 33. Resumen visual del MVC [19]

De manera genérica, los componentes de MVC se podrían definir como sigue [19]:

MODELO

Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va aquí, en el modelo.

CONTROLADOR

Se encarga de... controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista.

VISTAS

Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

6.2 Diseño de Base de Datos

Modelo Entidad-Relación (ER):

El **modelo entidad-relación ER** [20] es un **modelo** de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados **entidades** y **relaciones**, incorporando una representación visual conocida como diagrama **entidad-relación**

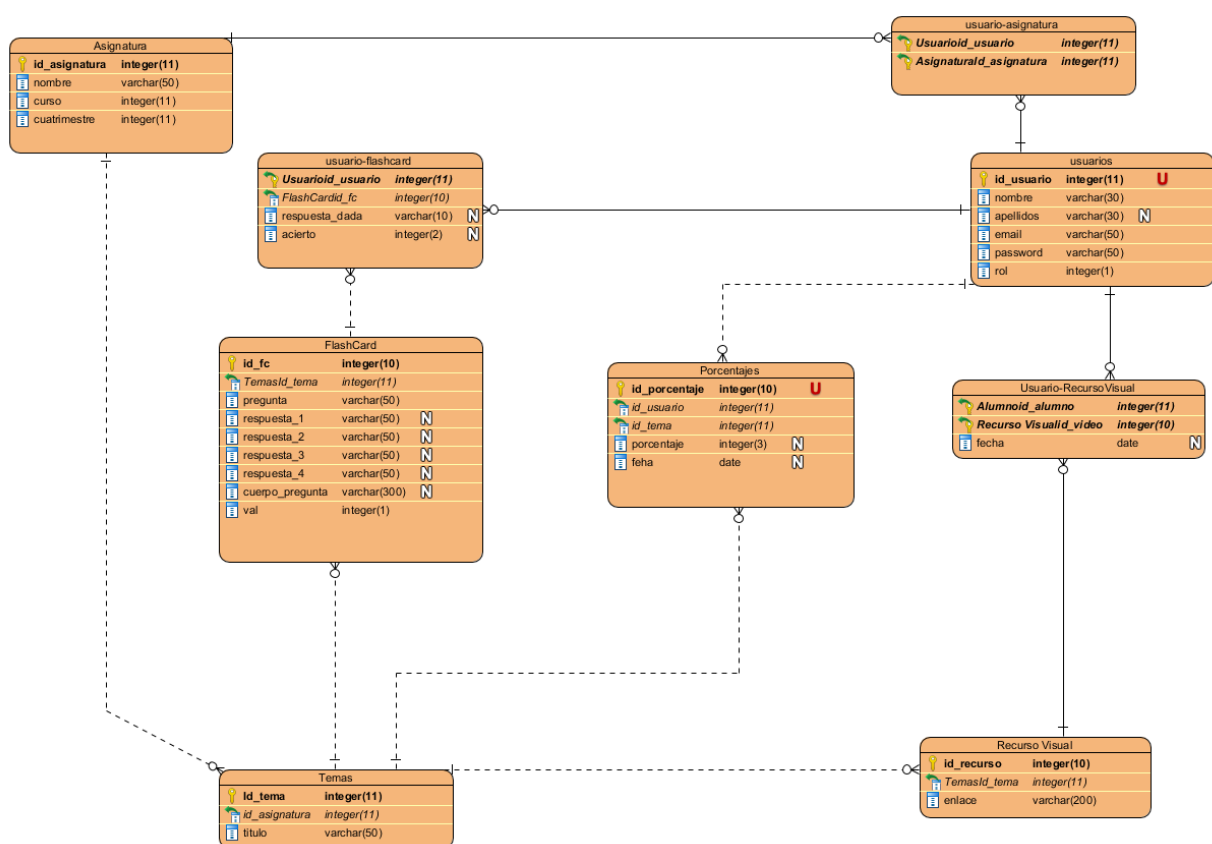


Ilustración 34. Modelo Entidad Relación

6.3 Diseño de página web

Se diseñaron **mockups** con la aplicación de Google **drawIO** para facilitar la fase de maquetación de la web.

¿Qué es un *mockup*?

En la manufactura y diseño, un ***mockup***, ***mock-up***, o **maqueta** es un modelo a escala o tamaño real de un diseño o un dispositivo, utilizado para la demostración, evaluación del diseño, promoción, y para otros fines. Un mockup es un prototipo si proporciona al menos una parte de la funcionalidad de un sistema y permite pruebas del diseño. Los *mockups* son utilizados por los diseñadores principalmente para la adquisición de comentarios por parte de los usuarios. Los *mock-ups* abordan la idea capturada en la ingeniería popular: «Usted puede arreglarlo ahora en el dibujo con una goma de borrar o más tarde en la obra con un martillo»

¿Qué herramientas puedo utilizar para construir un *mockup*?

Contamos con distintas herramientas interesantes a la hora de construir *mockups* pero tras comparar varias herramientas, entre las cuales está “<https://moqups.com/>”, “<https://mockflow.com/>”, “<https://balsamiq.com/products/mockups/>” y **drawIO**, la más atractiva es esta última al ser una herramienta de Google te permite de una manera fácil guardar tus proyectos en Google Drive, además de ser totalmente gratuita (las otras herramientas citadas son de pago o están limitadas a un servicio gratuito no muy extenso), es muy cómoda a la hora de utilizarla ya que te permite utilizar diseños Bootstrap que en concreto para mi aplicación permitía un diseño mucho más fiel y ajustado al producto final deseado [42].

Ciertos diseños puedes haberse vistos alterados mínimamente en el transcurso de desarrollo de la aplicación.

Registro

En esta página se muestra el formulario asociado al registro, así como la funcionalidad de ¿Ya tienes cuenta? Que redirigirá al *login*.

The mockup shows a registration form titled "Registro" on a grid background. The form is a rounded rectangle with a dark header bar at the top. The header bar contains the text "AppLearn", "Información", and a right arrow icon on the left, and "Usuario Registrados: X", "Iniciar Sesión", and "Registro" on the right. The form itself has five input fields: "Nombre", "Apellidos", "@ujaen.es", "Contraseña", and "Repetir Contraseña". Below these fields is a blue button labeled "Registrar". At the bottom of the form is a link that says "¿Ya tienes cuenta?".

Ilustración 35. Mockup: Registro

Login

En esta página se muestra el formulario asociado al *Login*

The mockup shows a login form titled "Login" on a grid background. The form is a rounded rectangle with a dark header bar at the top. The header bar contains the text "AppLearn", "Información", and a right arrow icon on the left, and "Usuario Registrados: X", "Iniciar Sesión", and "Registro" on the right. The form itself has two input fields: "@ujaen.es" and "Contraseña". Below these fields is a blue button labeled "Entrar".

Ilustración 36. Mockup: Login

Asignaturas no matriculadas

En esta página se muestran las asignaturas registradas en el sistema en la que no estamos matriculados y la opción de matricularnos en ellas.

AppLearn

Mis Asignaturas

Asignaturas no matriculadas

Estadísticas

Gestión de Usuarios

Rol

Correo

Cerrar Sesión

Asignatura	Curso	Acciones
Asigantura n	1º	Matricular
Asigantura n	2º	Matricular
Asigantura n	2º	Matricular
Asigantura n	3º	Matricular

Ilustración 37. Mockup: Asignaturas no matriculadas

Mis Asignaturas

En esta página se muestran las asignaturas en la que estamos matriculados y la opción de acceder a ellas o darnos de baja.

Además, el administrador del sistema contará con una opción que permita añadir nuevas asignaturas al sistema.

AppLearn

Mis Asignaturas

Asignaturas no matriculadas

Estadísticas

Gestión de Usuarios

Rol

Correo

Cerrar Sesión

Asignatura	Acción	Dar de baja
Asignatura n	Acceder	Dar de baja
Asignatura n	Acceder	Dar de baja
Asignatura n	Acceder	Dar de baja
Asignatura n	Acceder	Dar de baja

Ilustración 38. Mockup: Asignaturas matriculadas

Temas de asignatura

En esta página se muestran los temas asociados a una asignatura, junto a los botones de acceso a recursos visuales y *flashcards*, además, el profesor contará con la opción que permite añadir nuevos recursos visuales y el administrador además con la opción de añadir nuevos temas a la asignatura.

AppLearn

Mis Asignaturas

Asignaturas no matriculadas

Estadísticas

Gestión de Usuarios

Rol

Correo

Cerrar Sesión

Tema	Recursos Visuales		Flashcards
Value 1	Añadir nuevo	Acceder	Acceder
Value 4	Añadir nuevo	Acceder	Acceder
Value 7	Añadir nuevo	Acceder	Acceder
Value 10	Añadir nuevo	Acceder	Acceder

Ilustración 39. Mockup: Temas de la asignatura

Recursos Visuales

En esta página los usuarios tienen acceso a los recursos visuales que hayan sido subidos a la plataforma por el profesorado, además el profesor contará con la opción de borrar recurso

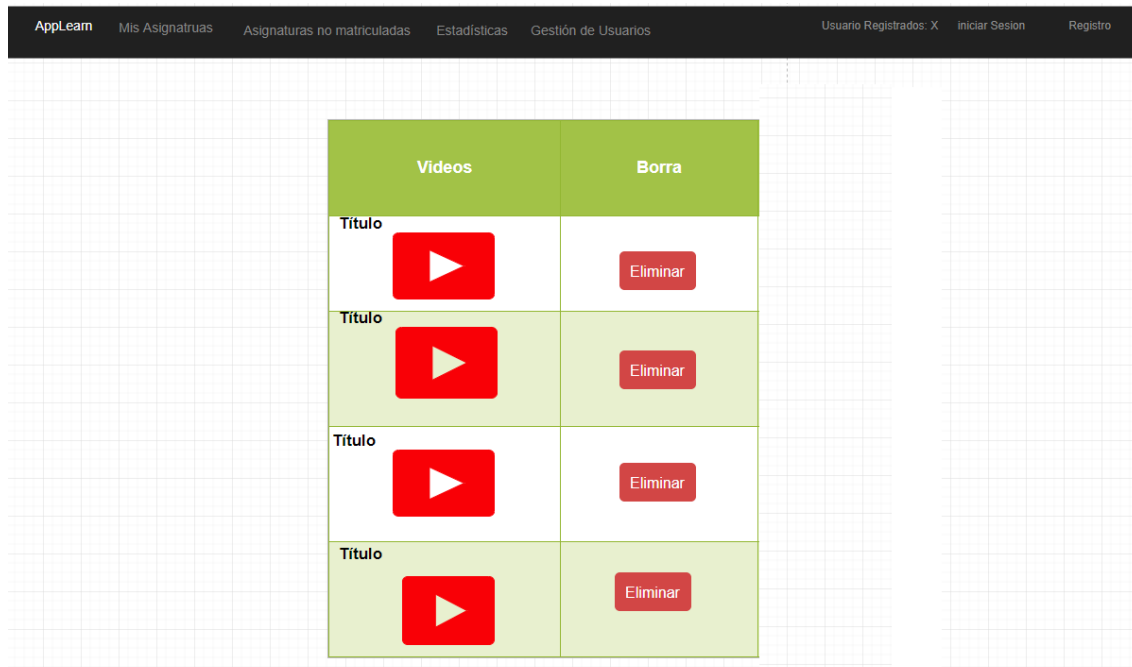


Ilustración 40. Mockup: Recursos Visuales

Flashcards (PROFESORADO)

En esta página el profesorado tiene acceso a datos de todas las *flashcards* asociadas a un tema concreto, así como las opciones de Añadir nueva *Flashcard*, Editar *Flashcard* previamente registrada en el sistema o borrar la *Flashcard* seleccionada.

Información Flashcard	Correcta	Acciones
Value 1	1	Seleccionar
Value 4	3	Seleccionar
Value 7	2	Seleccionar
Value 10	3	Seleccionar

Datos de Flashcard seleccionada

Pregunta

Cuerpo

R1

R2

R3

R4

Respuesta Correcta

Ilustración 41. Mockup: *Flashcards* [Profesores]

Flashcards (ESTUDIANTE) (I)

Es esta página se muestran *Flashcards* para que el usuario responda a ellas, junto a un tiempo bonus (bonificación si la respuesta correcta es respondida con rapidez) y tiempo restante.

SCORE

Tiempo BONUS

Tema

Título Flashcard

Cuerpo Flashcard

☐ R1

☐ R2

☐ R3

☐ R4

Confirmar

Ilustración 42. Mockup: *Flashcards* [Estudiantes] I

Flashcards (ESTUDIANTE) (II. FALLO)

Es esta página se muestra la *Flashcard* recién contestada por el estudiante, el *score* global y la respuesta correcta, ya que el estudiante ha fallado la *flashcard*.

Se informa al usuario que la respuesta es incorrecta.



Ilustración 43. Mockup: *Flashcards* [Estudiante] II. Fallo

Flashcards (ESTUDIANTE) (II. ACIERTO)

Es esta página se muestra la *Flashcard* recién contestada por el estudiante, y el score global.

Se informa al usuario que la respuesta es correcta.

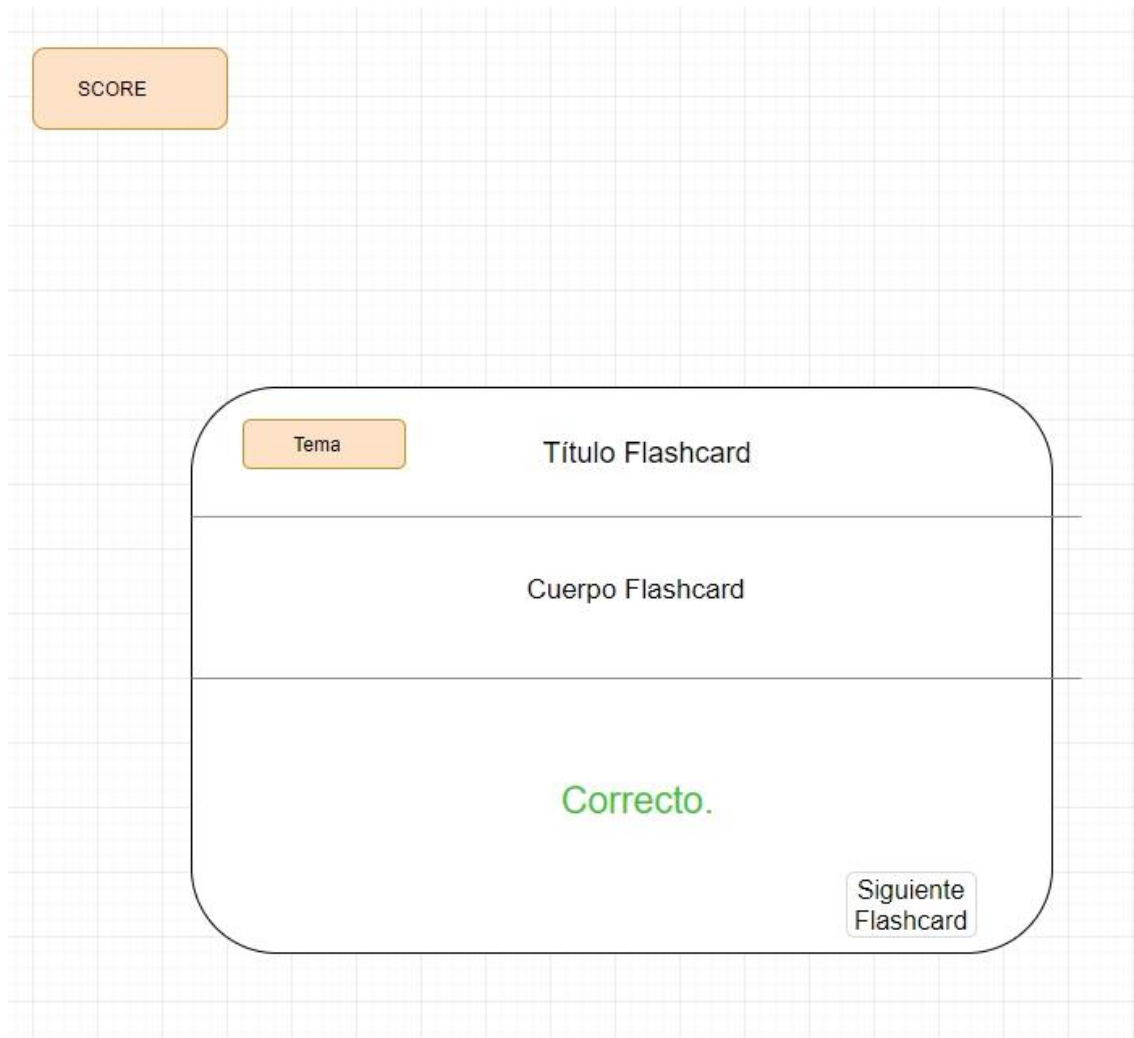


Ilustración 44. Mockup: *Flashcards* [Estudiante] II. Acierto

Flashcards (ESTUDIANTE) (III.NO SUPERADO)

En esta página se muestra al estudiante los resultados obtenidos respecto a score total y porcentaje de preguntas acertadas al finalizar el taco de *flashcards* junto a la opción de ir al menú principal.



Ilustración 45. Mockup: *Flashcards* [Estudiante] III. No superado

Flashcards (ESTUDIANTE) (III.SUPERADO)

En esta página se muestra al estudiante los resultados obtenidos respecto a score total y porcentaje de preguntas acertadas al finalizar el taco de *flashcards* junto a la opción de ir al menú principal,

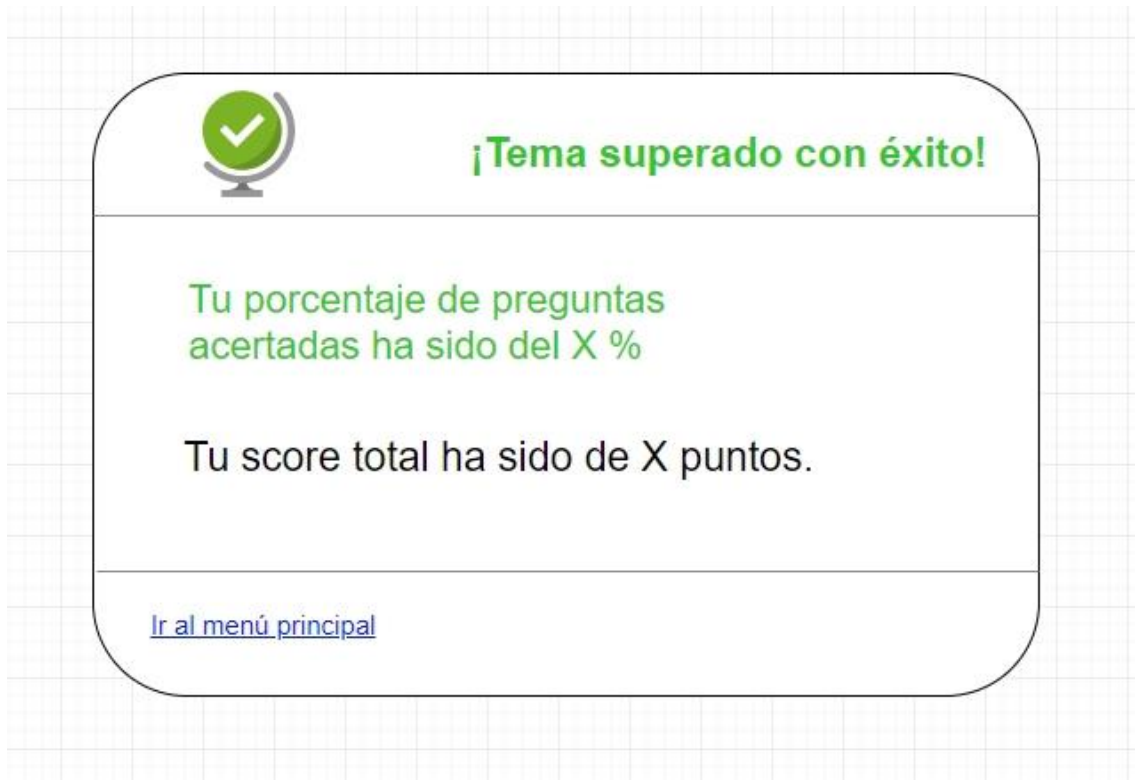


Ilustración 46. Mockup: Flashcards [Estudiante] III. Superado

Estadísticas (I)

Es esta página se ofrece al profesor que seleccione la asignatura, tema de la asignatura y agrupamiento estadístico que se quiera consultar.

Ilustración 47. Mockup: Estadísticas I

Estadísticas (II)

Es esta página se ofrece una tabla asociada al agrupamiento estadístico previamente seleccionado, junto con una representación gráfica (si fuere posible) en relación a dichos datos, además de la opción de Exportar los datos en formato CSV

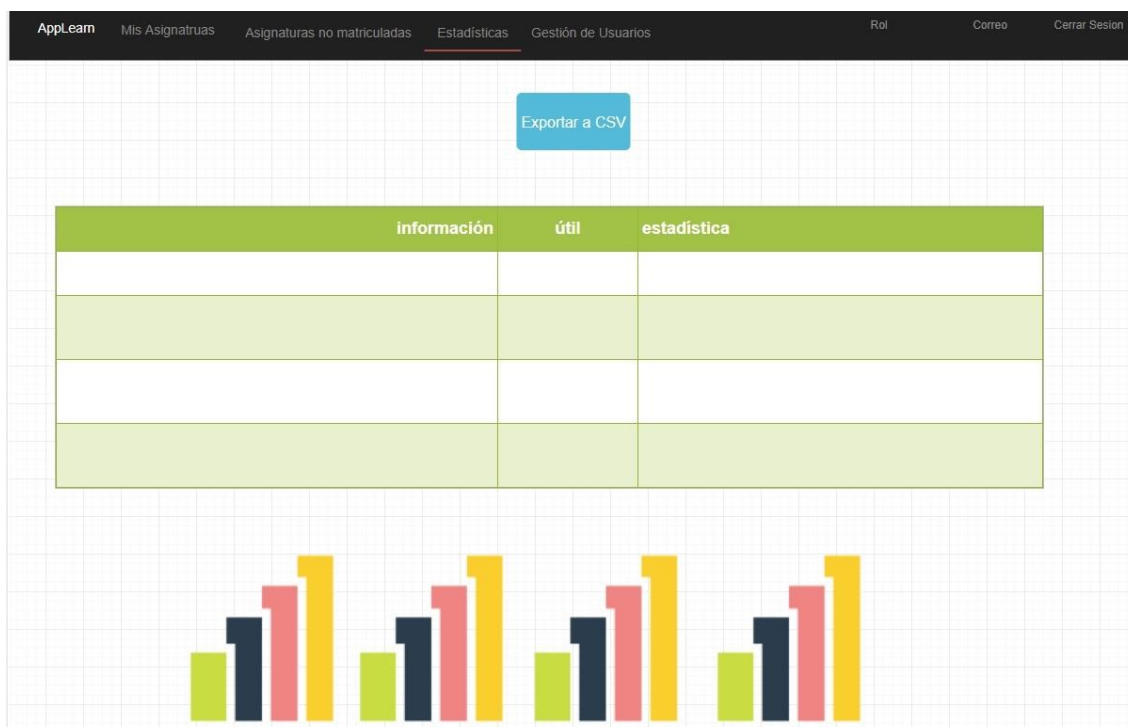


Ilustración 48. Mockup: Estadísticas II

Gestión de Usuarios (ADMIN)

En esta página el administrador tiene acceso a datos de todos los usuarios registrados en el sistema, así como las opciones de, Modificar usuarios previamente registrados en el sistema o Eliminar usuario seleccionado.

Información usuario	Rol	Acciones
Value 1	Rol01	Seleccionar
Value 4	Rol01	Seleccionar
Value 7	Rol02	Seleccionar
Value 10	Rol03	Seleccionar

Datos del usuario

Rol ▼

ModificarEliminar

Ilustración 49. Mockup: Gestión de usuarios

6.4 Modelo de Dominio

El modelo de dominio es un modelo conceptual formado por todos los temas relacionados con la resolución del problema a resolver. En este se describen las distintas entidades, sus atributos y relaciones entre si.

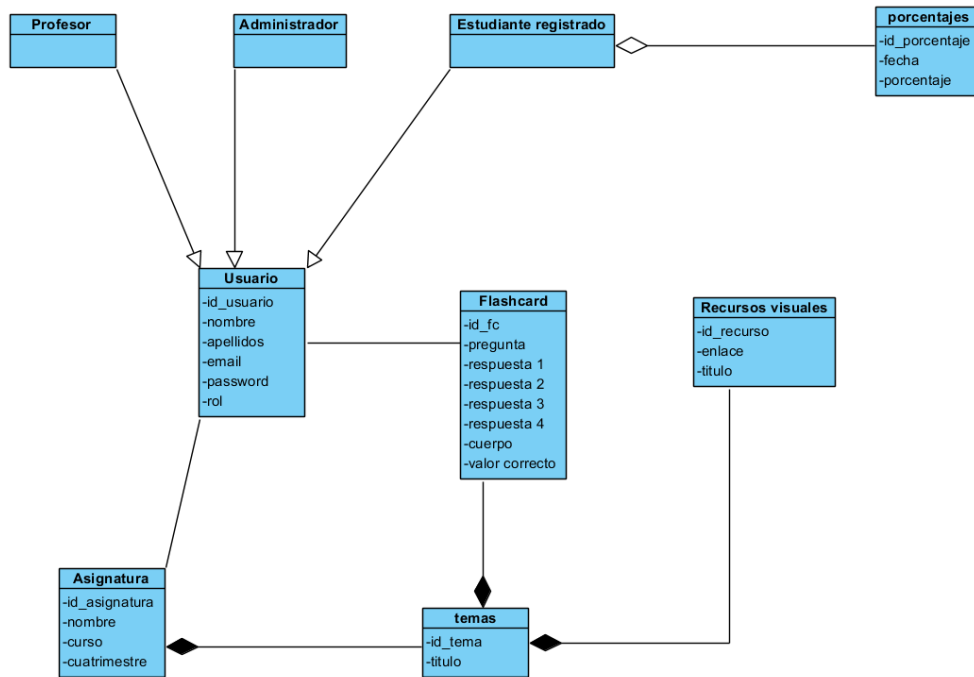


Ilustración 50. Diagrama del modelo de dominio

6.5 Diagramas de secuencia

Los diagramas de secuencia se utilizan para describir la forma en la que el sistema se comunica entre sí y con el actor.

Registro

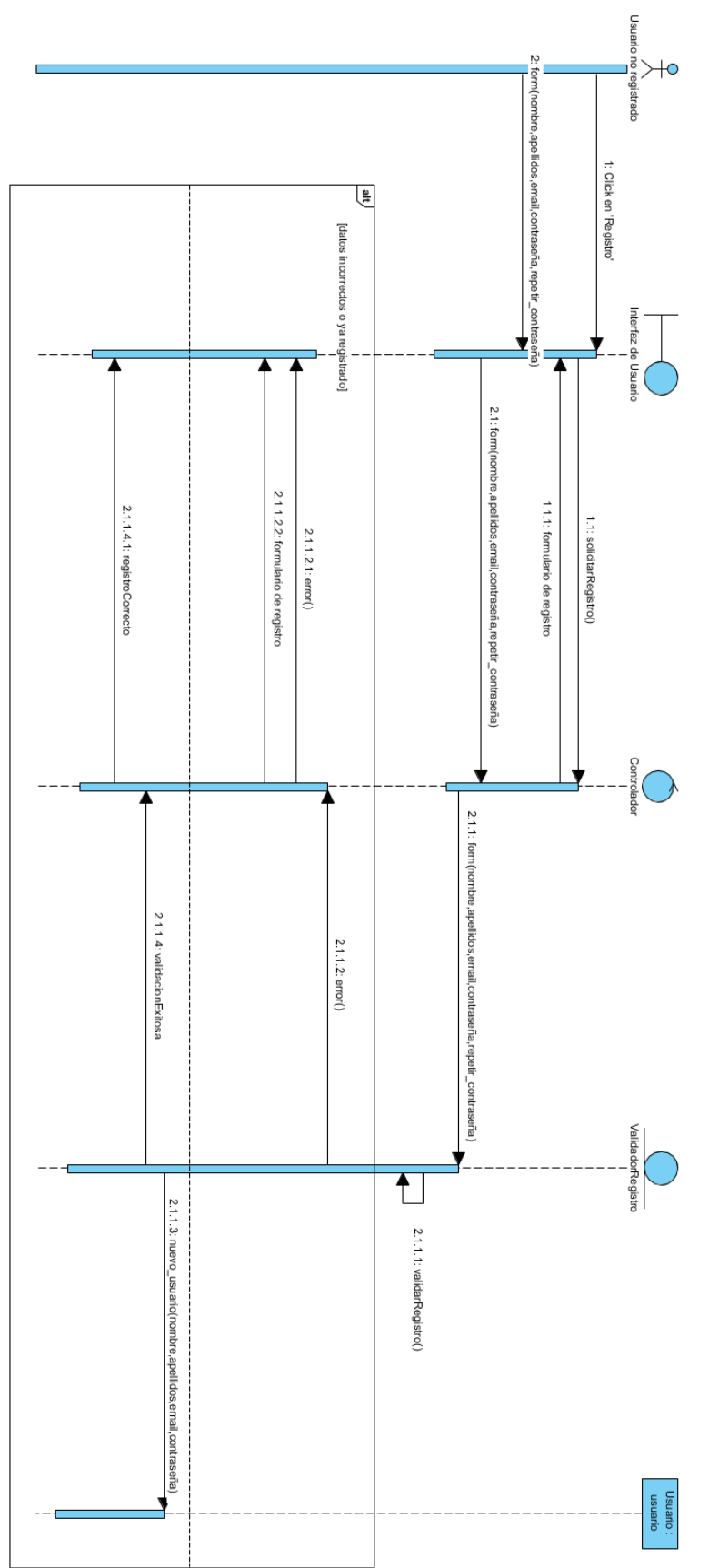


Ilustración 51. Diagrama de Secuencia: Registro

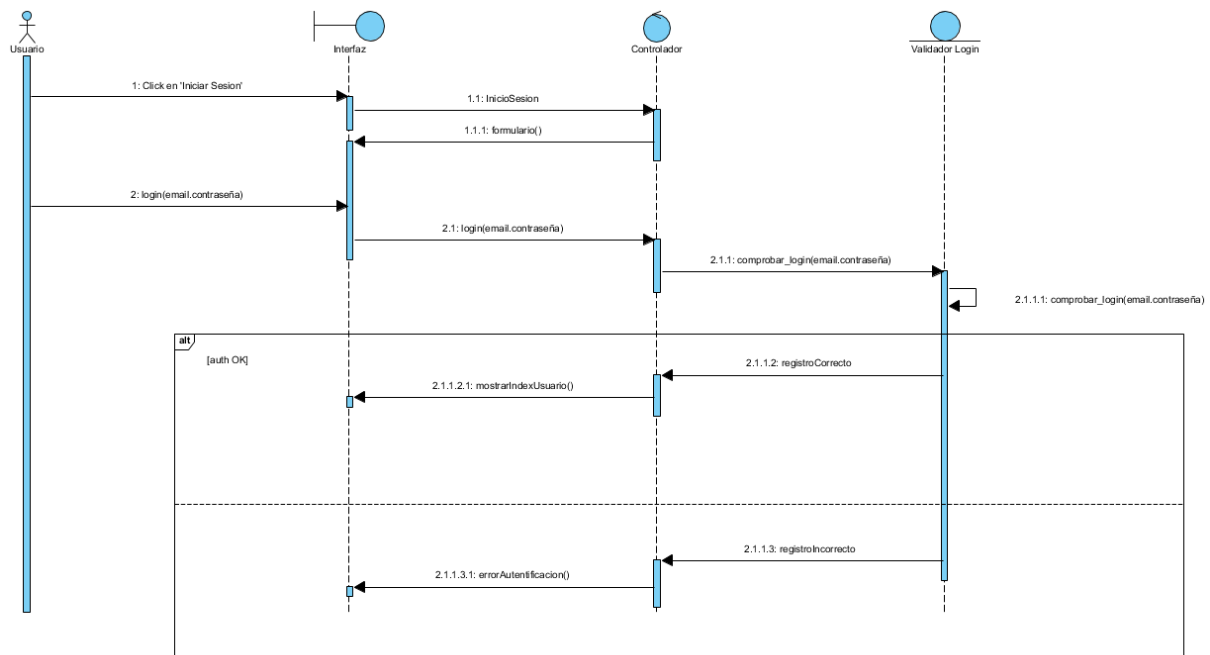
Inicio Sesión

Ilustración 52. Diagrama de Secuencia: Inicio de sesión

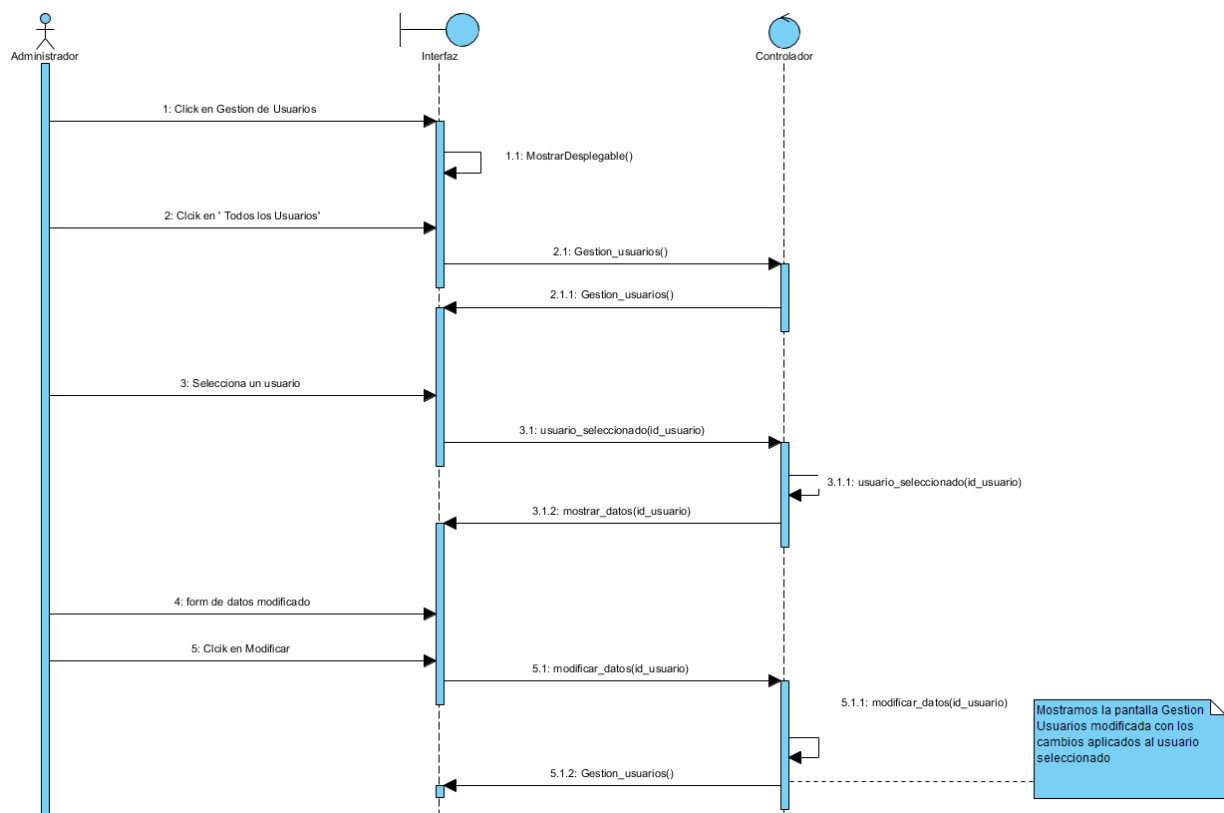
Modificar Usuario

Ilustración 53. Diagrama de Secuencia: Modificar usuario

Añadir Asignatura

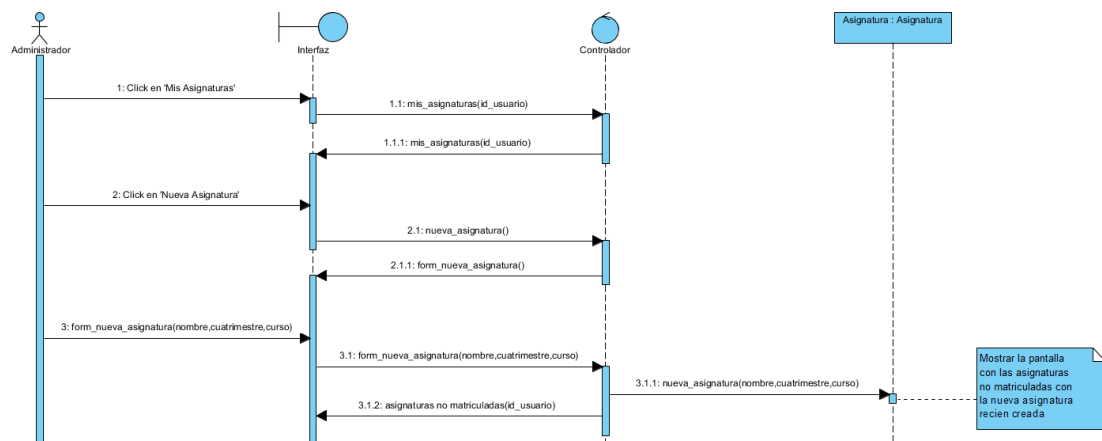


Ilustración 54. Diagrama de Secuencia: Añadir Asignatura

Ver Recursos Visuales

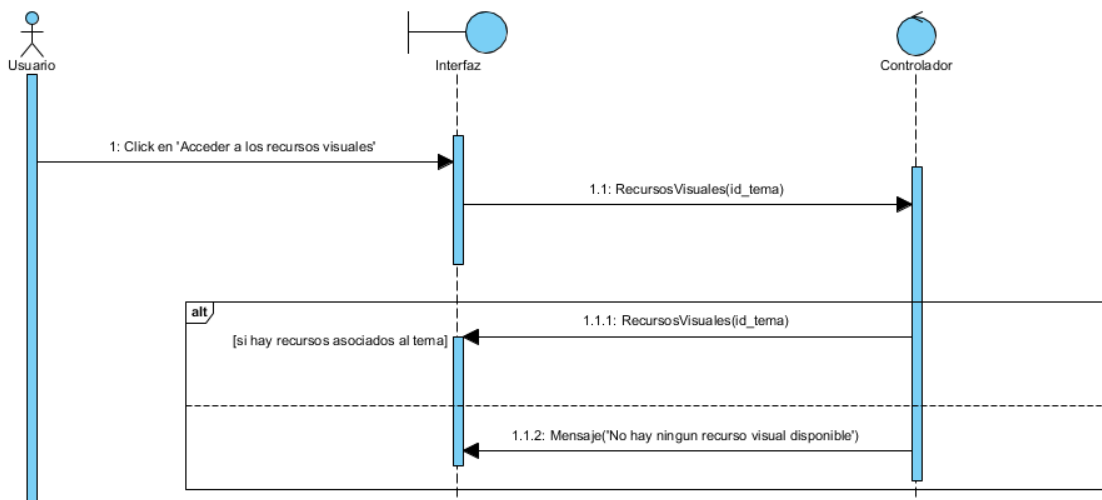


Ilustración 55. Diagrama de Secuencia: Acceder a los recursos visuales

Añadir Flashcard

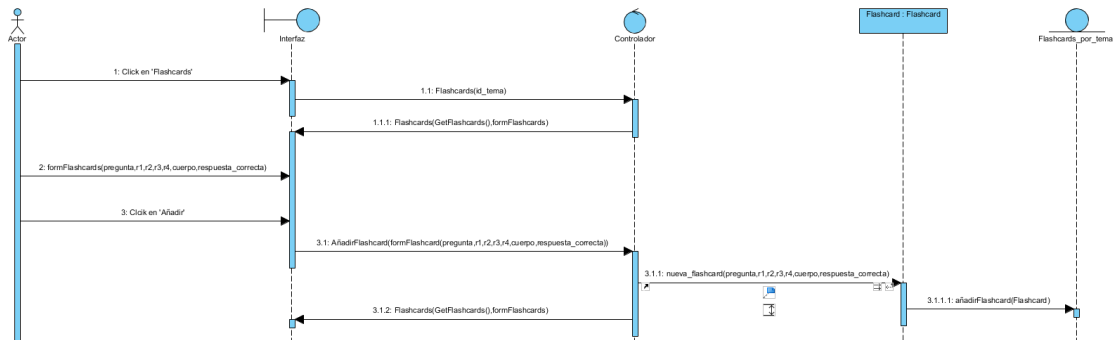


Ilustración 56. Diagrama de Secuencia: Añadir *flashcard*

Editar Flashcard

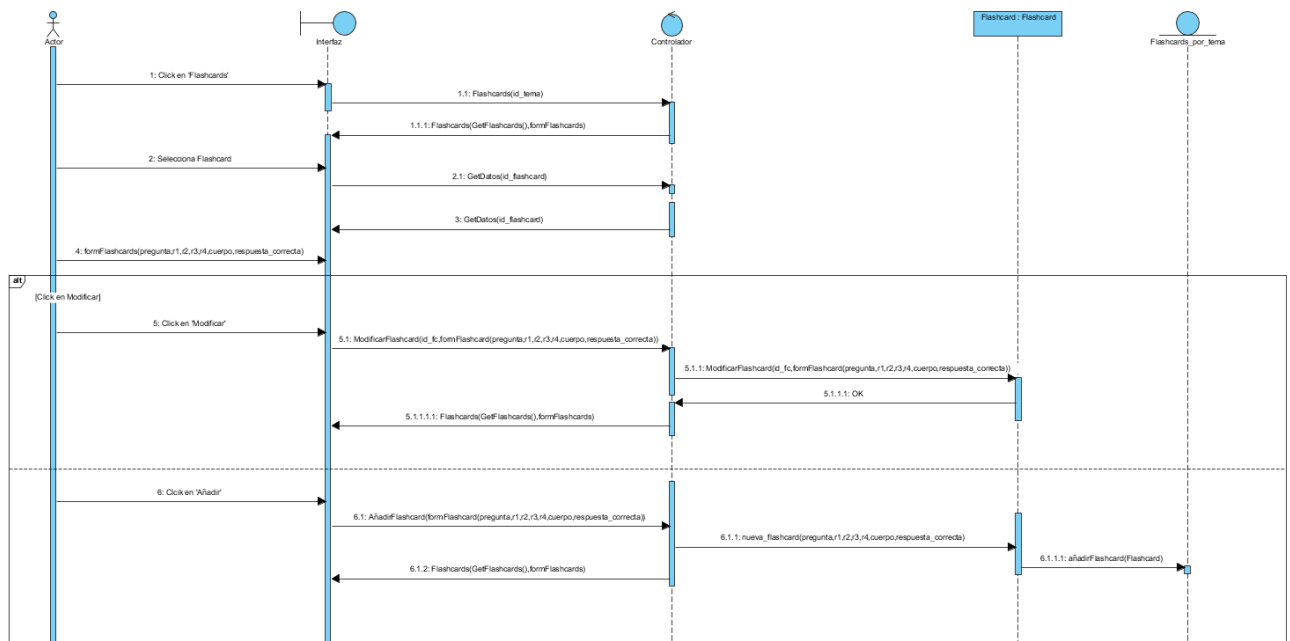
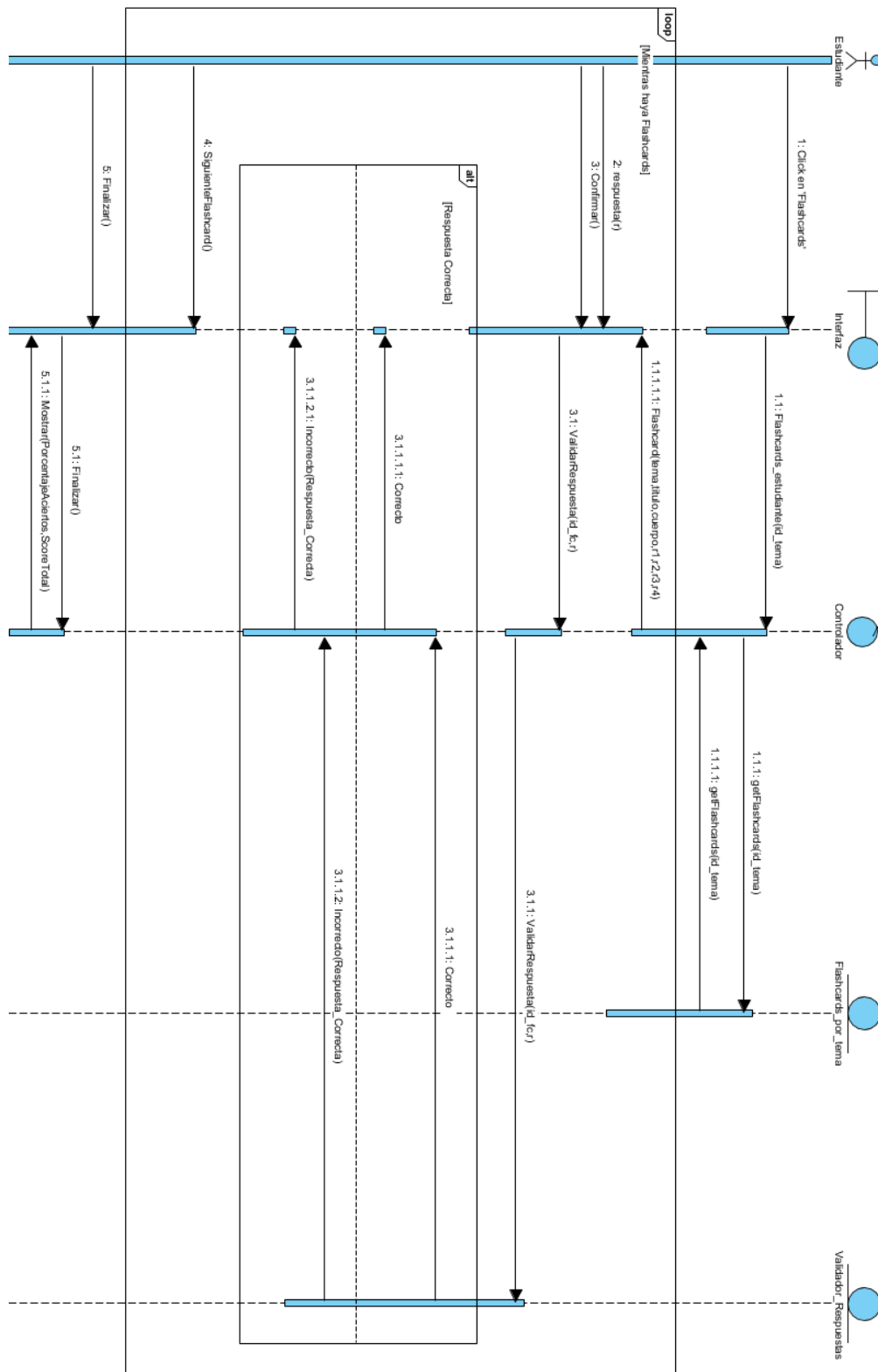


Ilustración 57. Diagrama de Secuencia: Editar *flashcard*

Realizar *Flashcard* (Estudiante)Ilustración 58. Diagrama de Secuencia: Realizar *flashcard* (Estudiante)

Consultar Estadísticas

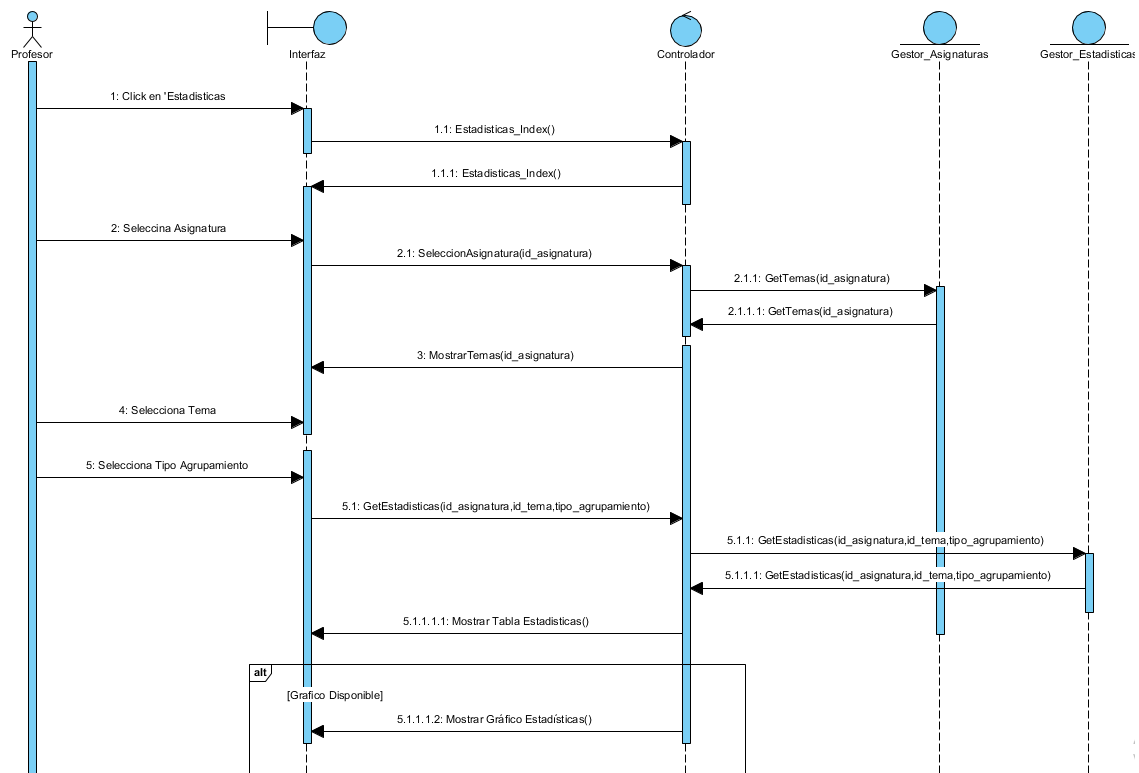


Ilustración 59. Diagrama de Secuencia: Consultar estadísticas

Descargar Estadísticas en formato CSV

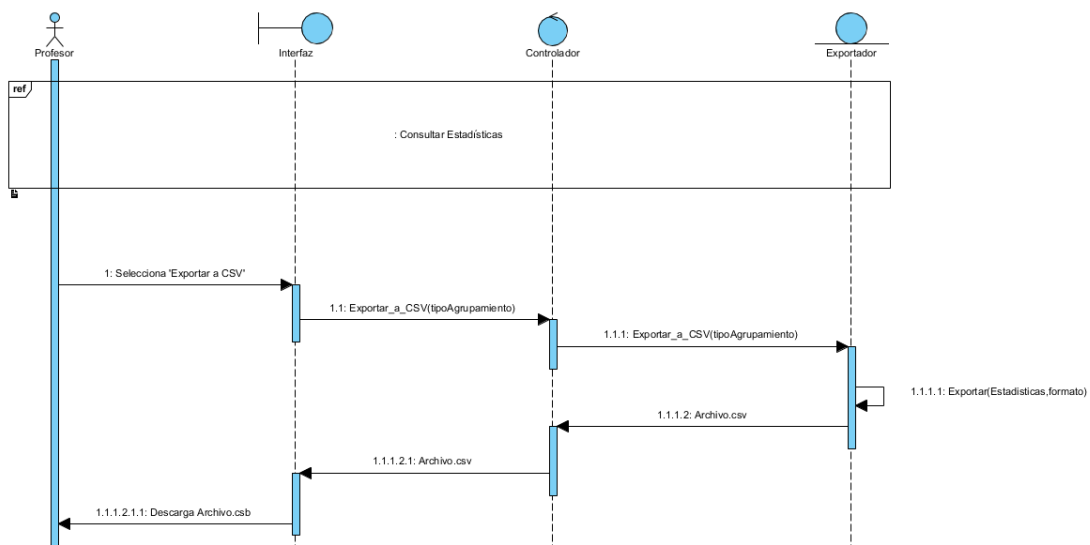


Ilustración 60. Diagrama de Secuencia: Descargar estadísticas con formato CSV

CAPÍTULO 7.- IMPLEMENTACIÓN Y PRUEBAS.

7.1 Uso de plantillas

Se han usado plantillas de declaración y cierre del documento, además de una barra de navegación común para dotar de una estructura homogénea a todo el proyecto.

7.2 Organización interna del proyecto

Se ha subdividido el proyecto en carpetas de la siguiente forma:

- App

Carpeta formada por **Validadores** (encargados de comprobar que la creación de objetos cumple con las restricciones definidas) y **Clases** con orientación a objetos (encargados de crear modificar, eliminar y mostrar objetos (o atributos del mismo) de una clase).

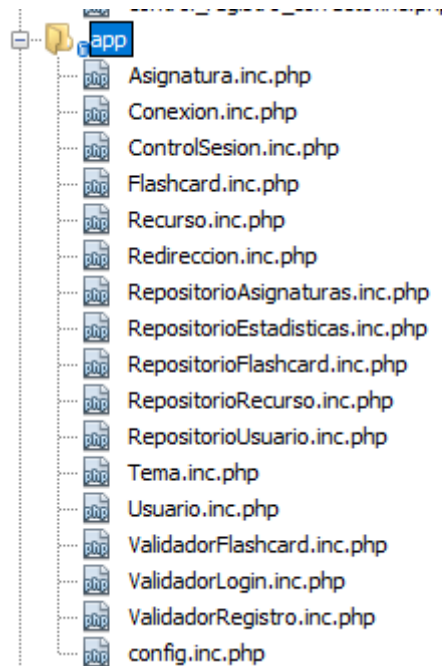


Ilustración 61. Organización interna del proyecto: app

- Control

Carpeta formada por controladores de una página concreta. Se encargan de toda la comunicación con el servidor.

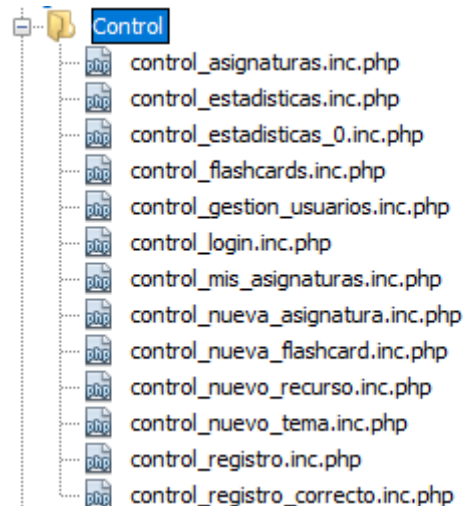


Ilustración 62. Organización interna del proyecto: Control

- Plantillas

Carpeta que incluye las plantillas útiles que son comunes a ciertas partes del sistema.

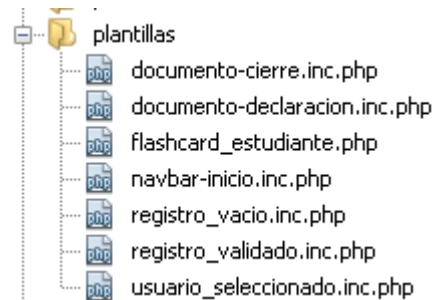


Ilustración 63. Organización interna del proyecto: Plantillas

Fuera de la organización de las carpetas se encuentran las páginas PHP con el HTML (*Frontend*) que se mostrará en la página correspondiente.

7.3 Creación de tablas en base de datos

Módulo SQL que carga en una base de datos las tablas requeridas en el proyecto.

- <https://github.com/francalzado/TFG-final/tree/master/base%20de%20datos>

7.4 Clases orientadas a objetos

Se hace uso de ciertas clases orientadas a objetos como es el caso de la clase Asignatura, para facilitar la gestión de los atributos y funcionalidades de los objetos.

```
<?php
```

```
class Asignatura {

    private $id_asignatura;
    private $nombre;
    private $curso;
    private $cuatrimestre;

    public function __construct($id_asignatura, $nombre, $curso, $cuatrimestre)
    {
        $this->id_asignatura = $id_asignatura;
        $this->nombre = $nombre;
        $this->curso = $curso;
        $this->cuatrimestre = $cuatrimestre;
    }

    public function obtenerId() {
        return $this->id_asignatura;
    }

    public function obtenerNombre() {
        return $this->nombre;
    }

    public function obtenerCurso() {
        return $this->curso;
    }

    public function obtenerCuatrimestre() {
        return $this->cuatrimestre;
    }
}
```

Ilustración 64. Clase orientada a objetos: Asignatura

7.5 Validadores

Un validador es una clase php asociada a un proceso en el que un usuario introduce ciertos datos sensibles, los cuales son objeto de estudio para asegurar la validez de los mismos. Son muy útiles a la hora de poner ciertas restricciones y asegurar un correcto funcionamiento de la aplicación. En la aplicación se hace uso de múltiples validadores para validar el registro, validar el *login* o para validar una *flashcard* a la hora de ser incorporada al sistema.

A continuación, se muestra el validador asociado al registro con algunas de sus funciones validadoras.

Constructor del validador del registro.

```
public function __construct($nombre, $apellidos, $email, $password, $confirm_password, $conexion) {
    $this->aviso_inicio = "<br><div class = 'alert alert-danger' role = 'alert'>";
    $this->aviso_cierre = "</div>";
    $this->nombre = $nombre;
    $this->apellidos = $apellidos;
    $this->email = $email;
    $this->password = $password;
    $this->confirm_password = $confirm_password;
    $this->error_nombre = $this->validar_nombre($nombre);
    $this->error_email = $this->validar_email($email, $conexion);
    $this->error_password = $this->validar_password($password);
    $this->error_confirm_password = $this->validar_confirm_password($password, $confirm_password);

    if ($this->error_password === "" && $this->error_confirm_password === "")
        $this->password = $password;
}
```

Ilustración 65. Constructor del validador del registro.

Función *validar_nombre* del validador de registro.

```
private function validar_nombre($nombre) {
    if (!$this->variable_iniciada($nombre)) {
        return "Debes escribir tu nombre";
    } else {
        $this->nombre = $nombre;
    }
    if (strlen($nombre) < 1 || strlen($nombre) > 25)
        return "Debes introducir un nombre de entre 2 y 25 caracteres";

    return "";
}
```

Ilustración 66. Función *validar_nombre()* asociado al validador de registro.

Función `validar_confirm_password` del Validador de registro.

```
private function validar_confirm_password($password, $confirm_password) {  
    if (!$this->variable_iniciada($password))  
        return "Debes introducir una contraseña";  
  
    if (!$this->variable_iniciada($confirm_password))  
        return "Debes repetir tu password";  
  
    if ($password != $confirm_password)  
        return "Ambas contraseñas deben ser iguales";  
  
    return "";  
}
```

Ilustración 67. Función `validar_confirm_password()` asociada al validador de registro.

7.6 Uso de AJAX en PHP

AJAX es usado en PHP para darle dinamismo a nuestras aplicaciones y hacerlas más interactivas.

Debajo muestro como hago uso de AJAX, en este caso, en la página de 'Estadísticas' para mostrar dinámicamente los temas asociados a una asignatura previamente seleccionada.

Parte de la página web donde se invoca el script

```
<select class="seleccion" name="cosa" onchange="cambia(this.value)">
```


Script JavaScript AJAX

```

<script type="text/javascript">
    function cambia(str) {

        var datos = {
            "id_asignatura": str,
        };

        $.ajax({
            type: 'POST',
            url: 'consulta.php',
            data: datos,
            dataType: 'text',
            success: function (d) {
                $("#txtHint").html(d);
                document.getElementById('txtHint').innerHTML = (d);
            }
        });
        return false;
    }
</script>

```

Ilustración 68. Script con AJAX

Consulta.php es el archivo donde se lleva a cabo la consulta AJAX. Se realiza un SELECT de los temas asociados a una asignatura previamente seleccionada y dichos temas se muestran en las distintas opciones del desplegable de las estadísticas de la siguiente forma.

```

if(isset($_REQUEST['id_asignatura'])) {
    $json_temas = RepositorioEstadisticas::obtener_temas_por_asignatura
        (Conexion :: obtener_conexion(), $_REQUEST['id_asignatura']);
    $response = '
    <select class="seleccion" name="opt" id="opt">
        <option value="-">Todos los temas';
        if (!empty($json_temas)) {
            for ($i = 0; $i < (COUNT($json_temas)); $i++) {
                $response.= ' <option value="'. $json_temas[$i][1]. '">'. $json_temas[$i][2];
            }
        }
    $response.= '</select>';
}

echo $response;

```

Ilustración 69. Consulta.php con AJAX

Una vez realizada la consulta dinámicamente, hay que capturar la respuesta de AJAX en un *DIV* y mostrarlo.

Parte de página web donde se recoge dinámicamente la respuesta de AJAX

```
<div id="txtHint" class="container"></div>
```

7.7 Enrutamiento y redireccionamientos

7.7.1 Enrutamiento

El enrutamiento de las distintas páginas las he previamente configurado en un archivo 'config.inc.php' en el que defino las rutas relativas de las páginas, ya que de esta forma independientemente de donde se cargue la aplicación el enrutamiento será correcto.

En este archivo mencionado anteriormente también está definido el acceso a la base de datos (credenciales) para una sencilla modificación de los mismos.

7.7.2 Redirección

La redirección es llevada a cabo a través de esta función contenida en la clase 'Redirección'.

```
<?php

class Redireccion {

    public static function redirigir($url) {
        header('Location: ' . $url, true, 301);
        exit();
    }

}
```

Ilustración 70. Redirección

En la imagen muestro como se realiza una redirección a la ruta de los temas, pasando por cabecera la asignatura de los cuales se pretende obtener los temas.

```
if ($accion) {  
    try {  
        //REDIRECCIONAMIENTO A LOS TEMAS DE LA ASIGNATURA SELECCIONADA  
        Redireccion :: redirigir(RUTA_TEMAS . '?id_asignatura=' . $txtId_asignatura);  
    } catch (PDOException $ex) {  
        print 'Error' . $ex->getMessage();  
    }  
}
```

Ilustración 71. Redireccionamiento a ruta de los temas de una asignatura.

7.8 Uso de métodos PHP

He utilizado los métodos PHP **POST**, **GET**, **REQUEST** y **SESSION**. Como vía de comunicación y flujo de información relevante entre distintas páginas.

Para la realización de pruebas se ha utilizado la consola de la Herramienta “Inspeccionar Elemento” junto a funciones del estilo `console.log()`, `print_r()` o `var_dump()`

CAPÍTULO 8.- CONCLUSIONES.

8.1 Resumen final

Se ha llevado a cabo el desarrollo de una aplicación web con el objetivo de realizar un seguimiento individualizado del uso de técnicas de aprendizaje activo por parte de los estudiantes que, voluntariamente, deseen participar en los programas de las asignaturas ofertadas.

La aplicación ofrecerá diversos perfiles:

- Administrador
- Profesor
- Estudiante

El administrador podrá gestionar los usuarios, así como las asignaturas y todo lo relativo a las mismas (temas, recursos y *flashcards*).

El profesor tendrá la opción de obtener información tanto detallada como agregada, numérica y visualmente, de los resultados obtenidos de los estudiantes participantes en su asignatura, así como gestionar los recursos y *flashcards* de su asignatura.

El estudiante tendrá la opción de acceder a un conjunto de recursos específicos de las asignaturas en las cuales esté matriculado, de cuyo uso se efectuará el seguimiento.

8.2 Conclusión Final

La ejecución del presente trabajo fin de grado ha hecho que me sienta muy realizado después de haber estudiado en profundidad las tecnologías más punteras en el desarrollo de aplicaciones web. Qué son, para que vale cada una de ellas, qué ventajas y desventajas nos ofrecen, cómo se complementan entre ellas, etc. además de haber estudiado cual es la que mejor se adaptaba a mi proyecto. He desarrollado una Aplicación Web con unas tecnologías muy demandadas en el mundo laboral a día

de hoy, como son PHP, Bootstrap, AJAX o JavaScript. Gracias a esta elección he podido afianzar mis conocimientos respecto a estas tecnologías y lenguajes de programación (algunas de ellas, totalmente desconocidas para mi hasta este momento) lo que ha enriquecido enormemente mi formación.

Considero además una oportunidad excelente la que se me ha brindado en este proyecto para formarme no solo de cara a mi futuro laboral, sino además a la hora de tomar decisiones importantes de diseño en el proyecto, así como a gestionar reuniones con mis tutores de proyecto.

Mi satisfacción es plena en la elección del proyecto ya que bajo mi punto de vista esta aplicación web es una herramienta que puede llegar a ser muy útil tanto para el profesorado a la hora de planificar el desarrollo una asignatura o tema concreto, así como para el estudiante a la hora de trabajar una asignatura de manera constante.

El resultado de esta aplicación es funcionalmente completo y satisface todos los requisitos exigidos; No obstante seguiré aprendiendo y formándome en el desarrollo de aplicaciones web, para poder aplicarlo a este proyecto, ya que considero que, es una herramienta que permite seguir trabajando en ella, para ofrecer más funcionalidades y así mejorar la experiencia de usuario.

8.3 Desarrollo Futuro

Las futuras versiones de la aplicación web añadirán valor a la aplicación a través de:

- Implementación de nuevos agrupamientos de estadísticas que mostrar al profesorado, junto a sus correspondientes graficas que muestren de una forma más visual las propias estadísticas obtenidas, con objeto de facilitar el estudio de las mismas.
- Modificación de datos por parte del propio usuario.
- Cada perfil de usuario contará con su foto de perfil.
- Recuperación de contraseña en caso de haberla olvidado.

- Investigar nuevas tecnologías aplicables a la aplicación para mejorar u ofrecer nuevas opciones tanto a los estudiantes como al profesorado y administradores que vayan a hacer uso de ella.

ANEXO I. Código fuente

El código fuente y documentación se encuentra en el repositorio GitHub.

- <https://github.com/francalzado/TFG-final>

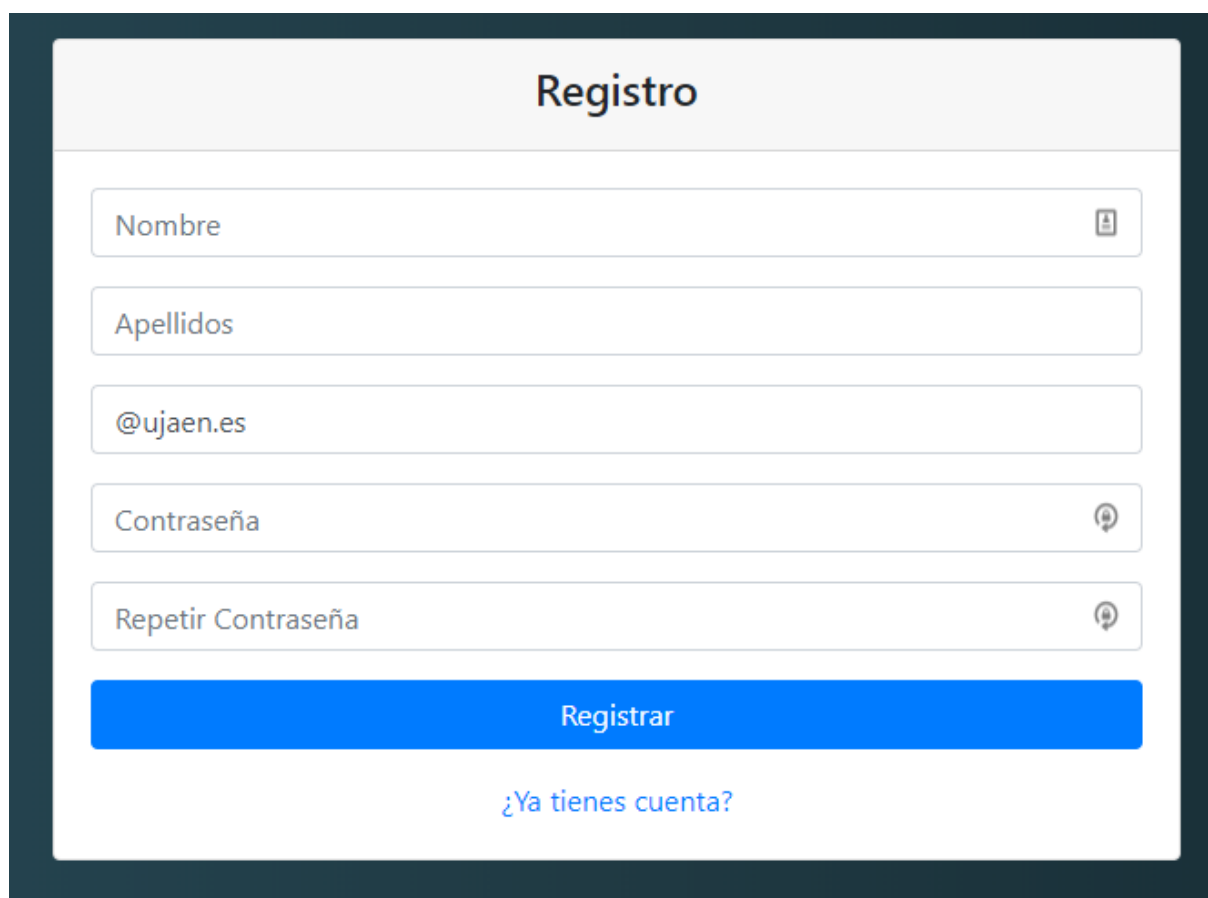
Para ejecutar el proyecto en local se hará uso de XAMPP y de la base de datos asociada, disponible también en el repositorio citado anteriormente.

ANEXO II. Manual de usuario.

A continuación, se muestra el manual de usuario junto a una guía de imágenes que faciliten la comprensión del mismo.

Registro

Para registrarnos en la aplicación debemos seleccionar la opción 'Registro', situada en la parte derecha de la barra de navegación y se mostrará un formulario de registro.



The image shows a registration form titled "Registro". It contains five input fields: "Nombre" (with a person icon), "Apellidos", "@ujaen.es", "Contraseña" (with an eye icon), and "Repetir Contraseña" (with an eye icon). Below the fields is a blue "Registrar" button. At the bottom, there is a link that says "¿Ya tienes cuenta?".

Ilustración 72. Manual de usuario. Registro

Si el registro lo realizamos correctamente la aplicación nos mostrará un mensaje como el mostrado en la siguiente captura de pantalla.

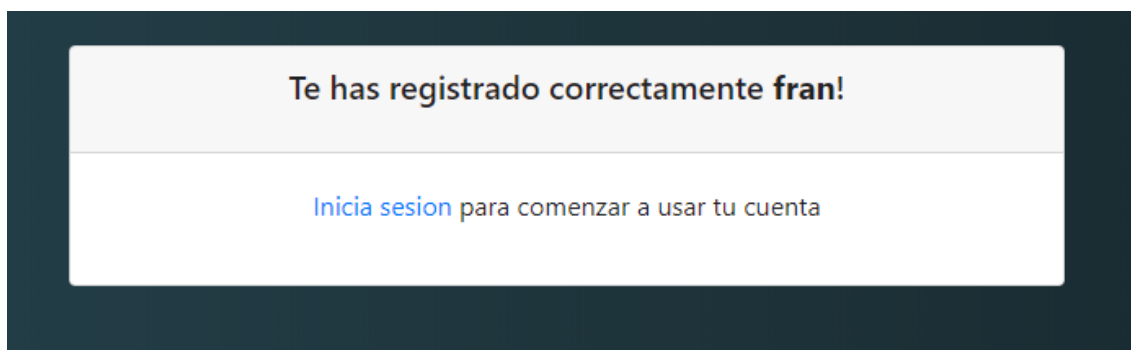


Ilustración 73. Manual de usuario. Registro correcto

Si, por el contrario, no realizamos el registro correctamente, se mostrará el formulario de registro junto al campo que falta por rellenar o es incorrecto.

Una captura de pantalla del formulario de registro de una aplicación web. El formulario tiene un título "Registro" en la parte superior. Contiene varios campos de entrada y mensajes de error. El primer campo es "Nombre", con un mensaje de error rojo que dice "Debes escribir tu nombre". El segundo campo es "Apellidos", con el valor "estudiante@ujaen.es" ingresado. El tercer campo es "Email", con un mensaje de error rojo que dice "Este email ya está en uso. recuperar contraseña". El cuarto campo es "Contraseña", con un mensaje de error rojo que dice "Debes escribir tu password". El quinto campo es "Repetir Contraseña", con un mensaje de error rojo que dice "Debes introducir una contraseña". En la parte inferior del formulario hay un botón azul que dice "Registrar" y un enlace azul que dice "¿Ya tienes cuenta?".

Ilustración 74. Manual de usuario. Registro incorrecto

Iniciar Sesión

Una vez realizado el registro en la aplicación, para iniciar sesión nos dirigiremos a la opción 'Iniciar Sesión' situada en la parte superior derecha, en la barra de navegación y se nos mostrará el siguiente formulario.

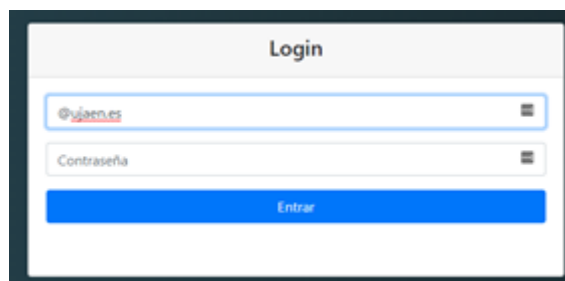
A screenshot of a login form titled 'Login'. It features two input fields: the first is for an email address, containing '@ujaen.es', and the second is for a password, labeled 'Contraseña'. Below these fields is a blue button labeled 'Entrar'.

Ilustración 75. Manual de usuario. *Login*

Si introducimos unas credenciales incorrectas, se mostrará de la siguiente forma:

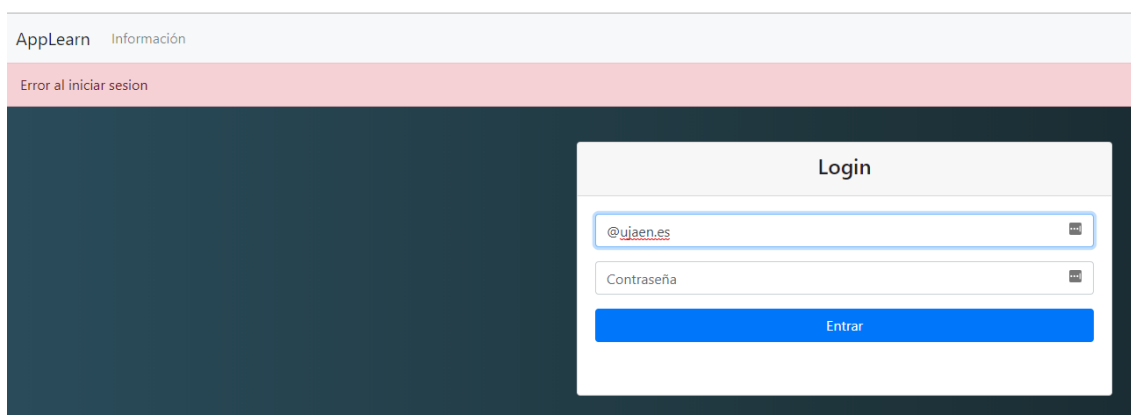
A screenshot of the application's login page. At the top, there is a header with 'AppLearn' and 'Información'. Below the header, a red banner displays the message 'Error al iniciar sesion'. The main content area is dark blue, and on the right side, the 'Login' form is visible. The form has the same fields as in the previous illustration, with the email field containing '@ujaen.es' and the password field labeled 'Contraseña'. A blue 'Entrar' button is at the bottom of the form.

Ilustración 76. Manual de usuario. *Login incorrecto*

Una vez iniciada sesión en la aplicación se nos mostrará la página de 'Mis Asignaturas', a no ser que no estemos aceptados en el sistema por el administrador.

En este caso no se nos mostrará la página citada anteriormente ya que necesitamos ser aceptados en el sistema. Se mostrará la siguiente página.

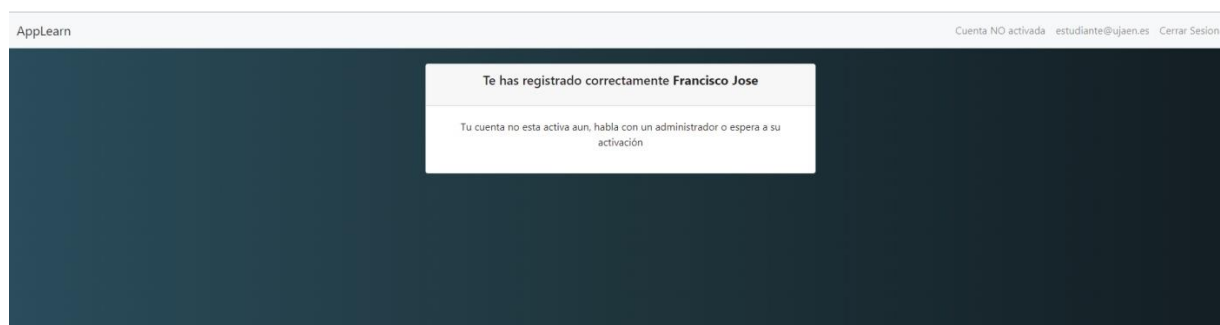


Ilustración 77. Manual de usuario. Login correcto

Para cerrar sesión en la aplicación es tan sencillo como, una vez iniciada la sesión, seleccionar la opción 'Cerrar Sesión', situada en la parte superior derecha de la barra de navegación.

En el caso de sí estar aceptados en el sistema, sí se mostrará la página 'Mis Asignaturas', la cual incluye (si las hubiere) las asignaturas en las cuales estamos matriculados como usuarios del sistema.

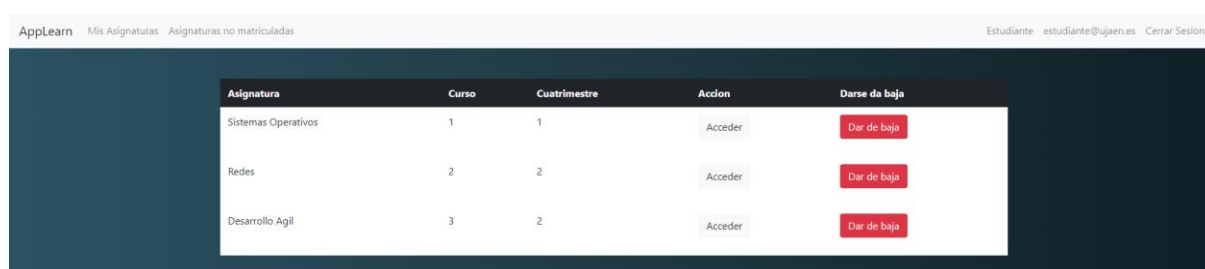


Ilustración 78. Manual de usuario. Index usuario

Si es la primera vez que accedemos al sistema y somos usuarios aceptados por el administrador, no tendremos ninguna asignatura matriculada por ahora y encontraremos la siguiente pantalla.

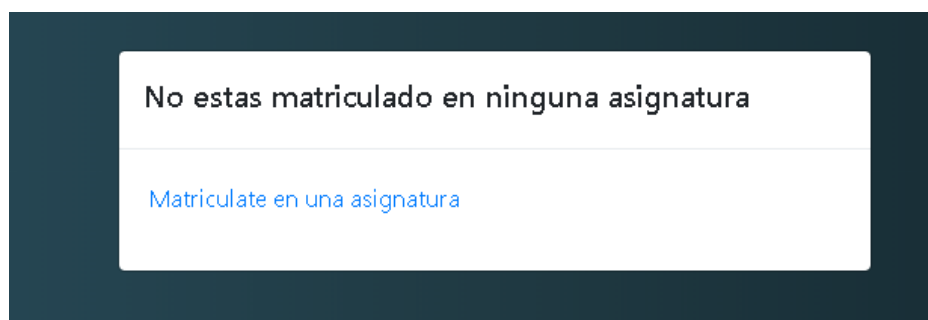


Ilustración 79. Manual de usuario. Index usuario sin asignaturas

Matriculación en asignatura.

Para matricular una asignatura podemos acceder a través del enlace 'Matricúlate en una asignatura' o bien seleccionar la opción 'Asignaturas no matriculadas' en la barra de navegación.

Se nos mostrará la siguiente página, en la cual figuraran las asignaturas ofertadas en las cuales no estamos matriculados.

Si deseamos matricularnos en una asignatura es tan sencillo como seleccionar la opción 'Matricular' de la asignatura, así como si queremos darnos de baja de una asignatura debemos seleccionar la opción 'Dar de baja' de la asignatura en la cual estemos previamente matriculados (pestaña de 'Mis Asignaturas').

AppLearn

Mis Asignaturas

Asignaturas no matriculadas

Estudiante

estudiante@ujaen.es

Cerrar Sesión

Asignatura	Curso	Cuatrimestre	Acciones
Sistemas Operativos	1	1	Matricular
Redes	2	2	Matricular
Estructuras de Datos	4	2	Matricular

Ilustración 80. Manual de usuario. Asignaturas no matriculadas

Acceder al temario de una asignatura.

Si queremos acceder al temario de una asignatura concreta en la cual estemos matriculados seleccionar la opción 'Acceder' y encontraremos una página con el temario de la asignatura, junto a los recursos visuales y *flashcards* vinculadas a cada tema.

En el caso de ser un profesor, en la parte de recursos visuales, tendremos también la opción de añadir nuevos recursos visuales.

Temario	Recursos Visuales	Flashcards
SCRUM	Acceder	Realizar
eXtreme Programming	Acceder	Realizar

Ilustración 81. Manual de usuario. Temario de una asignatura

Si eres estudiante.

Acceso a recursos visuales.

Al acceder a la opción 'Recursos Visuales' encontrarás los distintos videos que te ofrecen para el estudio del tema seleccionado.

Videos

Scrum en 6 minutos

#3. Scrum en 6 minutos - Metodolo...

Product Owner, Scrum Team, Scrum Master, Daily Scrum, Sprint, Sprint Review, Product, Sprint Retrospective, Sprint BackLog, Product BackLog, Sprint Planning Meeting.

Métodos Ágiles. SCRUM

Metodologías ágiles. El proceso SC...

Sprints

Sprint 1, Sprint 2, Sprint 3, Sprint 4

Ilustración 82. Manual de usuario. Recursos visuales

Realizar *Flashcards*.

Si accedemos a la opción de 'Realizar' referente a *flashcards* encontraremos la siguiente página con un score referente al taco de *flashcards* que vamos a realizar, un tiempo bonus que nos dará puntos extras si contestamos con rapidez y una *flashcard* a contestar con diversas opciones a marcar. Podemos marcar 1 opción o dejarla en blanco, y una vez hayamos hecho nuestra elección, seleccionaremos 'Confirmar'.



Ilustración 83. Manual de usuario. *Flashcards* para estudiante I

En el caso de contestar correctamente encontraremos la siguiente pantalla, con un score actualizado y acumulado en función de la rapidez de respuesta y un mensaje confirmando que la respuesta dada es Correcta. Para pasar a la siguiente *flashcard* basta con pulsar el botón 'Siguiente *Flashcard*'.

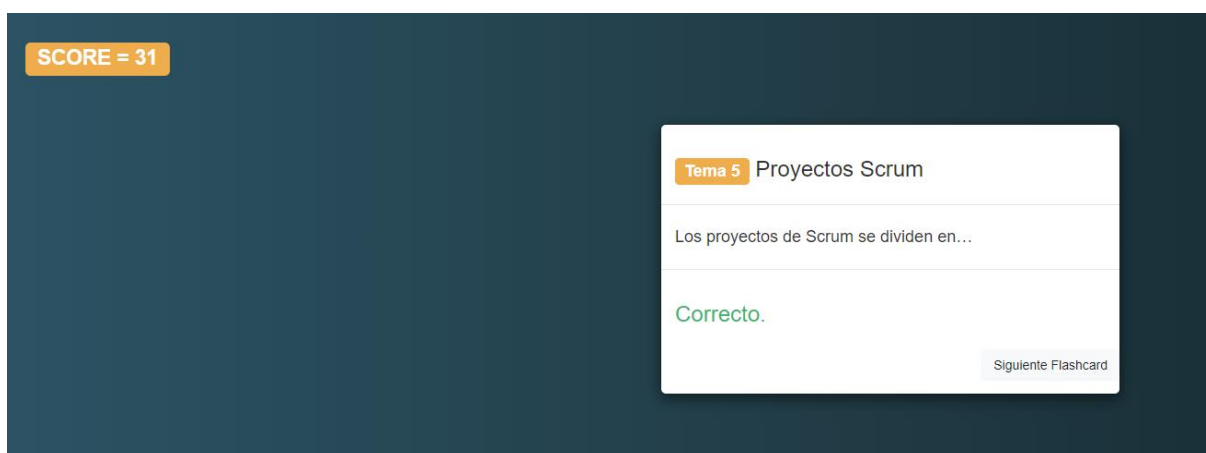


Ilustración 84. Manual de usuario. *Flashcards* correcta

En el caso de contestar erróneamente encontraremos la siguiente pantalla con un mensaje confirmando que la respuesta dada es errónea, junto a la respuesta correcta. Para pasar a la siguiente *flashcard* basta con pulsar el botón 'Siguiente *Flashcard*'.

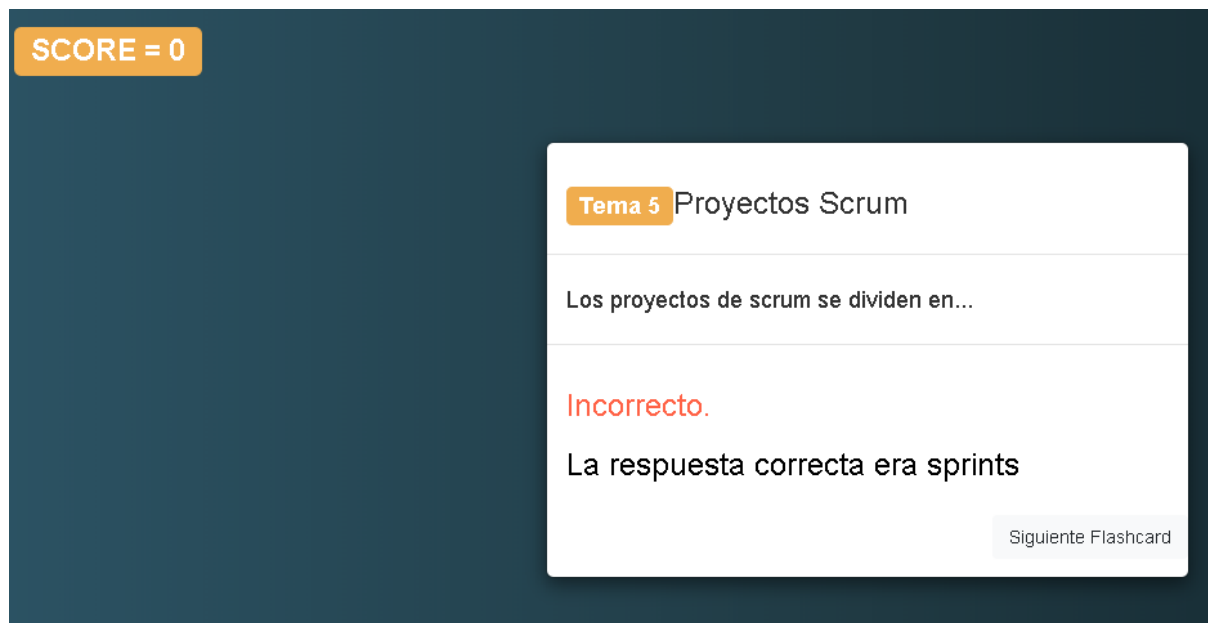


Ilustración 85. Manual de usuario. *Flashcard incorrecta*

Al finalizar el taco de *flashcards* del tema, se mostrará un resumen con el porcentaje de preguntas acertadas, el score total conseguido y si el tema se da por superado o no.

Si se diera por superado se mostrará la siguiente pantalla.

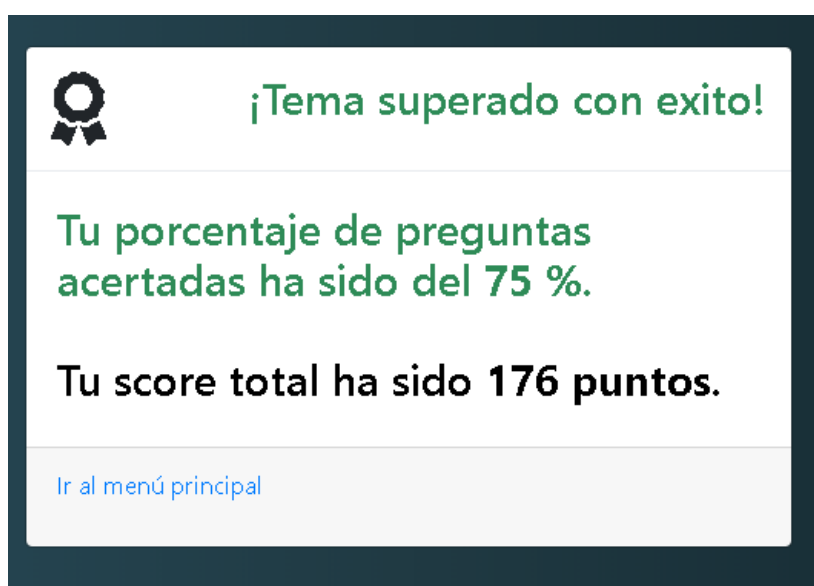


Ilustración 86. Manual de usuario. Tema superado

Si se diera por no superado, se mostrará la siguiente pantalla.

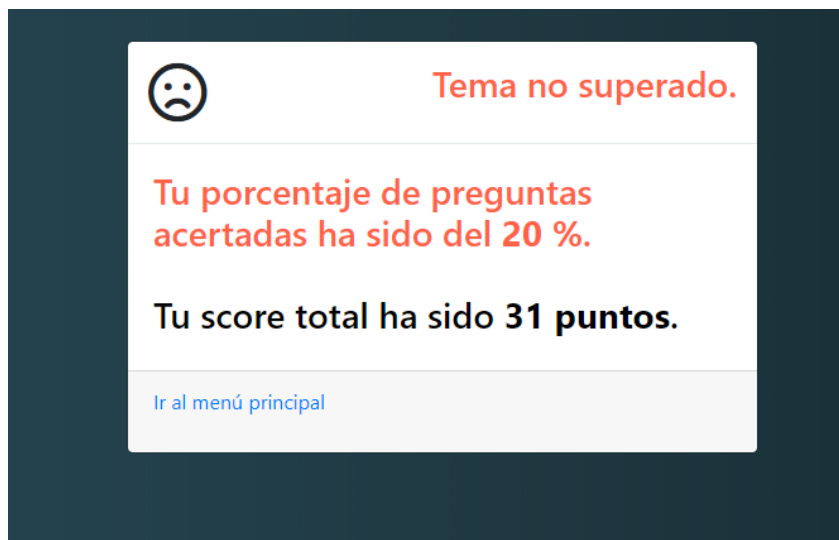


Ilustración 87. Manual de usuario. Tema no superado

En cualquier caso, para volver a los temas de la asignatura habría que seleccionar la opción 'Ir al menú principal'.

Si eres profesor.

Acceso y gestión de recursos visuales.

Al acceder a la sección de recursos visuales, se obtendrá una vista como la que se muestra a continuación. Como se puede observar, si se quiere eliminar un video es posible a través de la opción 'Eliminar'

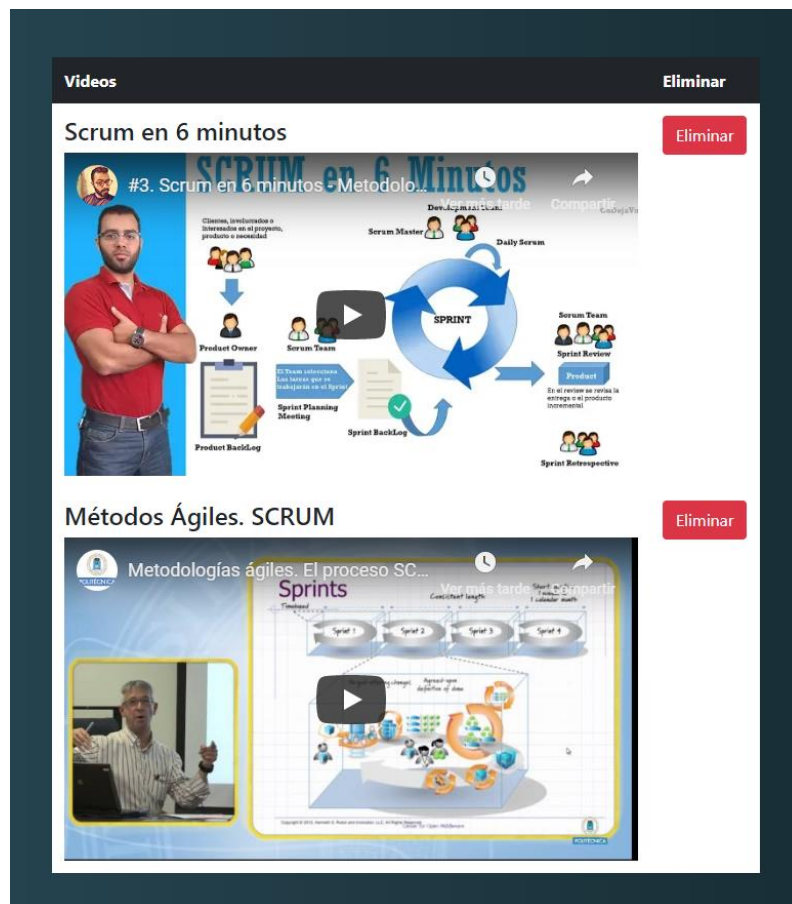


Ilustración 88. Manual de usuario. Vista de recursos visuales de profesor

Si queremos añadir un recurso visual nuevo, basta con acceder a la opción 'Añadir nuevo recurso', donde podremos completar el formulario para introducir un nuevo recurso y asociarlo a un tema concreto.



Ilustración 89. Manual de usuario. Añadir nuevo recurso visual

Como se puede observar, también tenemos la opción de 'Ver recursos visuales' para observar si el video se ha introducido correctamente o simplemente se quiere revisar la disposición de videos global del tema.

Para añadir un nuevo recurso basta con añadir un Título y el enlace de YouTube del video en cuestión.

Para obtener el enlace correctamente no basta con copiar el enlace del video, sino que hay que cambiar parte de la ruta de la siguiente forma:

Si el enlace al video de YouTube es el siguiente:

`https://www.youtube.com/watch?v=abc123`

Habría que modificar del enlace la parte de **/watch?v=** por **/embed/**

Quedando el enlace de la siguiente forma:

`https://www.youtube.com/embed/abc123`

De esta forma conseguimos un enlace con el video insertable.

Acceso y gestión de *Flashcards*.

Para gestionar las *flashcards* de un tema concreto accederemos a través de la opción ‘Acceder’ en el apartado ‘*Flashcards*’ de la tabla y obtendremos una vista como la siguiente.

The screenshot shows the AppLearn interface. At the top, there's a navigation bar with 'AppLearn', 'Mis Asignaturas', 'Asignaturas no matriculadas', 'Estadísticas', and 'Gestión de Usuarios'. On the right, it says 'Administrador admin@ujaen.es' and 'Cerrar Sesión'.

The main content area is divided into two parts. On the left is a table of flashcards. On the right is a form titled 'Datos de Flashcard seleccionada'.

Pregunta	Cuerpo	R1	R2	R3	R4	Correcta	Acciones
Proyectos Scrum	Los proyectos de Scrum se dividen en...	Fases	Sprints	Hitos	Oleadas	2	Seleccionar
Reunion Diaria	¿Cuál de las siguientes afirmaciones sobre la reunión diaria es FALSA?	Todo el mundo permanece en pie	Es lo mismo que una reunión de control	Es mas efectiva cuando todos se escuchan	Es una oportunidad para que todos se impliquen	1	Seleccionar
Mejor Opcion Daily StandUp	Eres el jefe de un equipo que acaba de empezar a trabajar ágilmente. El primer cambio que hacéis es tener una "daily standup meeting", pero algunos miembros del equipo dicen que no les gusta asistir. A pesar de que estás recibiendo información valiosa en cada reunión, te preocupa que esta práctica pueda dañar la cohesión de tu equipo. ¿Qué es lo MEJOR que puedes hacer?	Dejar de tener las reuniones diarias, y buscar otr	Establecer una regla para que cada persona en la r	Reunirte con los miembros del equipo uno a uno des	Trabajar con el equipo para cambiar su actitud	4	Seleccionar
Equipo de trabajo Scrum	Un equipo scrum no debe ser mayor a...	5 personas incluyendo Scrum Master y Product Owner	7 personas incluyendo Scrum Master y Product Owner	9 personas incluyendo Scrum Master y Product Owner	9 personas excluyendo Scrum Master y Product Owner	3	Seleccionar
Reunion Diaria	¿Cuál de las siguientes afirmaciones sobre la reunión diaria es VERDAD?	Todo el mundo permanece en pie	No debe durar mas de 15 minutos	Se habla sobre lo que se va a hacer mañana	Se puede hacer por grupos separados	2	Seleccionar

The right-hand form, 'Datos de Flashcard seleccionada', contains the following fields:

- Pregunta:
- Cuerpo:
- R1:
- R2:
- R3:
- R4:
- Respuesta Correcta:
- Respuesta 1:

At the bottom of the form are three buttons: 'Añadir' (green), 'Modificar' (yellow), and 'Eliminar' (red).

Ilustración 90. Manual de usuario. Gestión de *flashcards*

En esta vista se ven reflejadas las *flashcards* registradas y un formulario.

Si queremos añadir una *flashcard*, habría que rellenar los campos del formulario requeridos y posteriormente pinchar en la opción ‘Añadir’.

Si queremos modificar una *flashcard* previamente registrada, habría que seleccionar la opción ‘Seleccionar’ de la tabla que contiene las *flashcards* y los datos de dicha *flashcard* aparecerán en el formulario situado a la derecha.

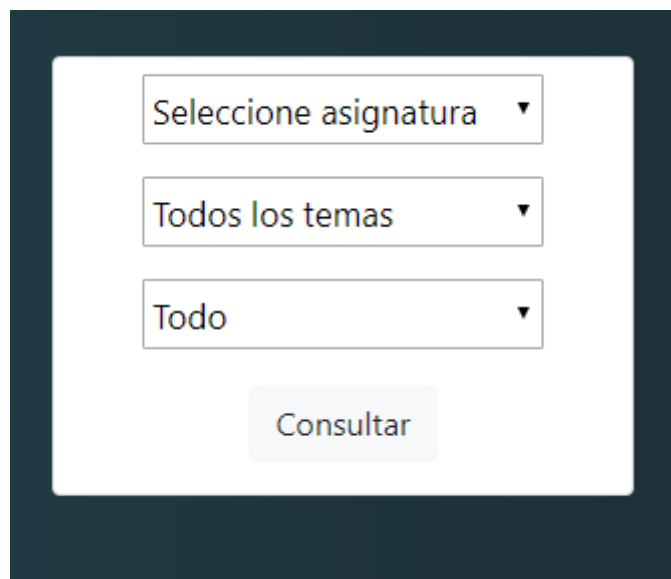
Una vez modificados los campos deseados, pinchar en la opción ‘Modificar’ para reflejar los cambios.

Si queremos eliminar una *flashcard* previamente registrada, habría que seleccionar la opción ‘Seleccionar’ de la tabla que contiene las *flashcards* y los datos de dicha *flashcard* aparecerán en el formulario situado a la derecha.

Bastará con pinchar en la opción ‘Eliminar’ para eliminar la *flashcard* del sistema, así como todos los datos y estadísticas generados que guardan relación con ella.

Consulta de Estadísticas.

En la barra de navegación encontraremos la opción ‘Estadísticas’ donde al acceder a ella encontraremos la siguiente página



Seleccione asignatura ▼

Todos los temas ▼

Todo ▼

Consultar

Ilustración 91. Manual de usuario. Selección de estadísticas

En esta, se nos ofrecen 3 desplegables. En el primero seleccionaremos la asignatura de la cual queremos obtener las estadísticas, en el segundo, el tema (también tenemos la opción de consultar a cerca de todos los temas) respecto al cual realizar la consulta y en el tercero elegiremos el tipo de agrupamiento estadístico que deseamos consultar

Una vez seleccionadas las opciones deseadas y usando el botón 'Consultar' obtendremos una página con el siguiente formato.

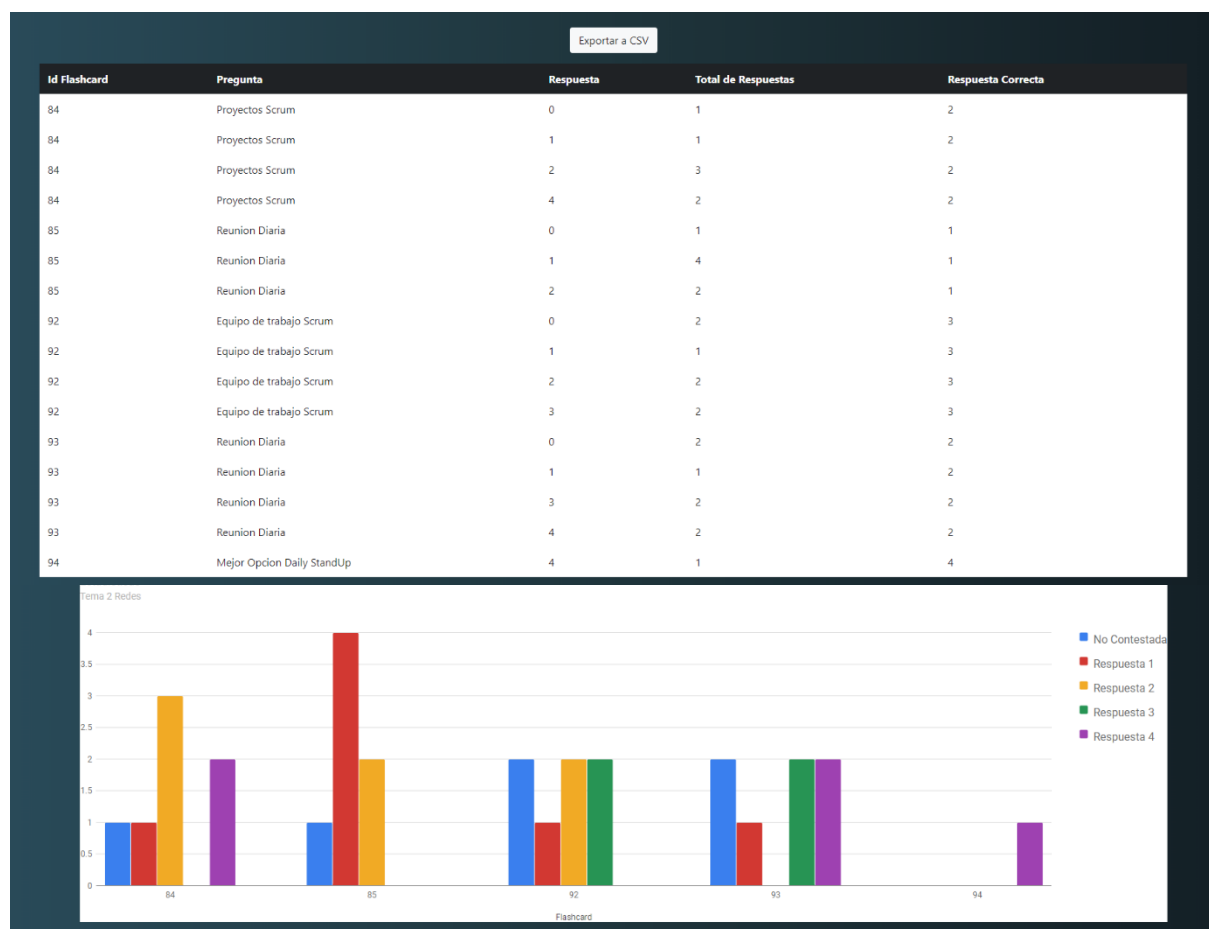


Ilustración 92. Manual de usuario. Estadísticas agregadas

Como se puede observar en la ilustración anterior, contamos con una tabla estadística, un gráfico asociado a la misma y la opción de exportar a tu pc los datos consultados con formato CSV.

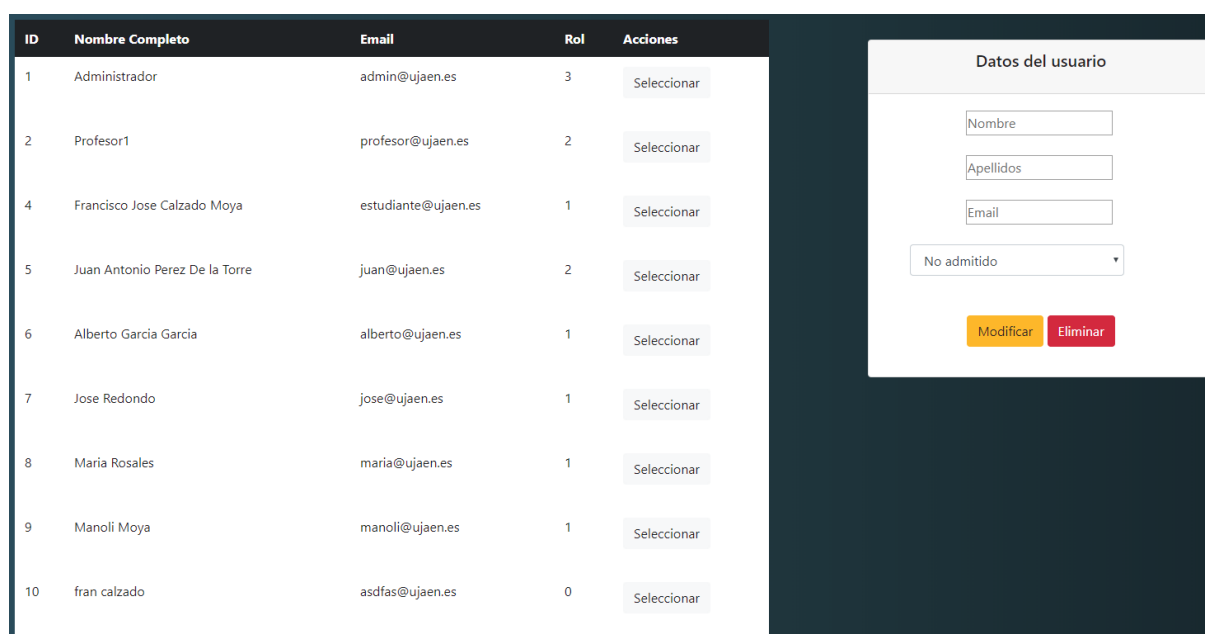
Si eres administrador.

Como administrador del sistema, se tendrá acceso a todas las funcionalidades del perfil profesor citadas anteriormente.

Gestión de usuarios.

En la barra de navegación encontramos la opción desplegable ‘Gestión de Usuarios’, la cual nos ofrece dos opciones, ‘Todos los usuarios’ y ‘Nuevos Usuarios’, siendo esta segunda opción una extensión de la primera, con la misión de facilitar la aceptación en masa de los nuevos usuarios registrados al sistema.

Al hacer uso de esta opción encontraremos la siguiente página.



The screenshot displays a user management interface. On the left, a table lists users with columns for ID, Nombre Completo, Email, Rol, and Acciones. On the right, a 'Datos del usuario' form is shown, which is currently empty, indicating that no user has been selected yet. The table contains 10 rows of user data.

ID	Nombre Completo	Email	Rol	Acciones
1	Administrador	admin@ujaen.es	3	Seleccionar
2	Profesor1	profesor@ujaen.es	2	Seleccionar
4	Francisco Jose Calzado Moya	estudiante@ujaen.es	1	Seleccionar
5	Juan Antonio Perez De la Torre	juan@ujaen.es	2	Seleccionar
6	Alberto Garcia Garcia	alberto@ujaen.es	1	Seleccionar
7	Jose Redondo	jose@ujaen.es	1	Seleccionar
8	Maria Rosales	maria@ujaen.es	1	Seleccionar
9	Manoli Moya	manoli@ujaen.es	1	Seleccionar
10	fran calzado	asdfas@ujaen.es	0	Seleccionar

Datos del usuario

Nombre

Apellidos

Email

No admitido

Ilustración 93. Manual de usuario. Gestión de usuarios

Esta página cuenta con una tabla donde se recoge información de los usuarios registrados en el sistema y en la parte derecha, un formulario que nos permitirá modificar los datos de un usuario.

Para Modificar un usuario haremos uso del botón ‘Seleccionar’ referente a un usuario concreto. La información del usuario se plasmará en el formulario de la parte derecha donde podremos modificar los datos deseados. Para guardar la modificación pulsar el botón ‘Modificar’.

Si se desea eliminar un usuario del sistema, haremos uso del botón 'Seleccionar' referente a un usuario concreto. La información del usuario se plasmará en el formulario tal y como se ha explicado anteriormente y pulsando la opción 'Eliminar' se borrará al usuario del sistema.

Añadir Asignatura

Como administrador del sistema también tendremos la posibilidad de añadir una nueva asignatura al sistema. Para ello se ha de acceder a la opción 'Mis Asignaturas' en la barra de navegación y pulsar la opción 'Nueva Asignatura', situada en la parte inferior de la tabla.

Asignatura	Curso	Cuatrimestre	Acción	Darse da baja
Sistemas Operativos	1	1	Acceder	Dar de baja
Redes	2	2	Acceder	Dar de baja
Nueva Asignatura				

Ilustración 94. Manual de usuario. Añadir asignatura

Se mostrará un formulario a completar con la información de esta nueva asignatura a añadir, una vez completado, pulsar en 'Registrar'

Registrar Nueva Asignatura

Primer Cuatrimestre

▼

Segundo Curso

▼

Registrar

Ilustración 95. Manual de usuario. Formulario de nueva asignatura

Añadir tema.

Si se precisa añadir un nuevo tema a una asignatura, habrá que acceder a los temas de la asignatura como se describe previamente y hacer uso de la opción 'Nuevo Tema'



Ilustración 96. Manual de usuario. Añadir nuevo tema

Encontraremos un formulario básico, donde simplemente tendremos que indicar el título del tema y tras pulsar el botón 'Guardar', se nos redireccionará a los temas ofertados en la asignatura, con el nuevo tema recién registrado disponible para añadirle nuevos recursos visuales y *flashcards* al mismo.

ANEXO III. Instalación y Configuración XAMPP

Para descargar XAMPP vamos a su página oficial y descargamos la versión acorde a nuestro sistema operativo, en este caso, Windows.

Paso 1. Descargar XAMPP para Windows a través de la web oficial



Ilustración 97. Instalación y configuración XAMPP. Acceso a la web oficial

Paso 2. Abrir instalador XAMPP

Pulsamos *Next*

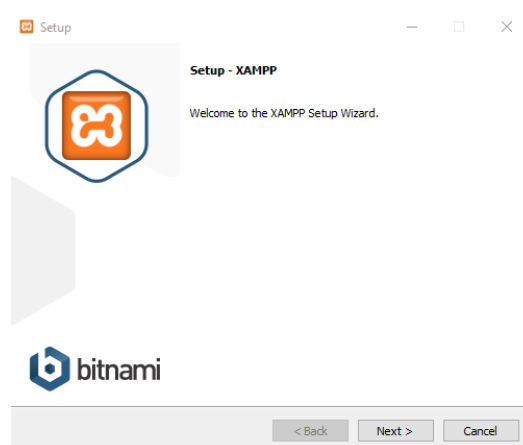


Ilustración 98. Instalación y configuración XAMPP. Instalación I

Paso 3. Marcar como **mínimo** las opciones Apache, MySQL, PHP, PhpMyAdmin.

Pulsar *Next*

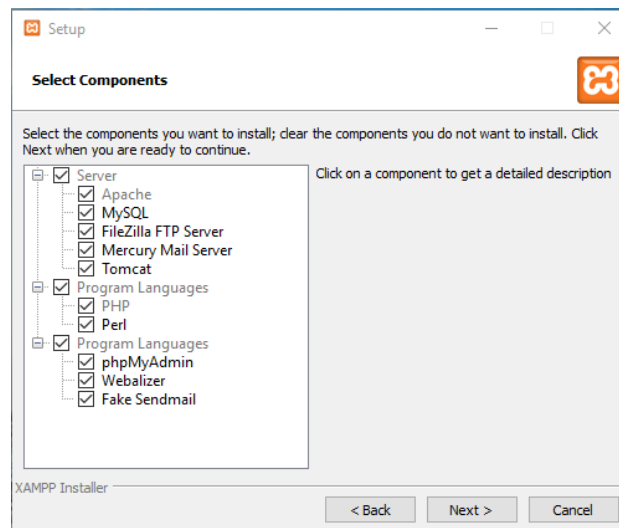


Ilustración 99. Instalación y configuración XAMPP. Instalación II

Paso 4. Elegiremos la ruta por defecto que nos ofrece el instalador y pulsamos *Next*

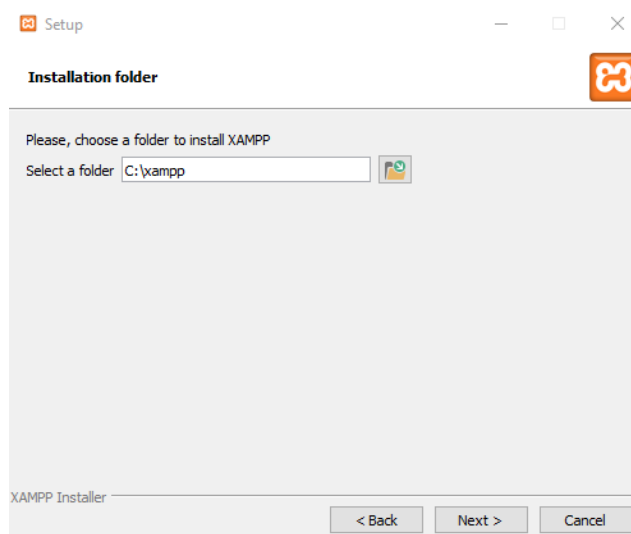


Ilustración 100. Instalación y configuración XAMPP. Instalación IV

Paso 5. Desmarcar la opción '*Learn more...*' y *Next*

Next de nuevo.

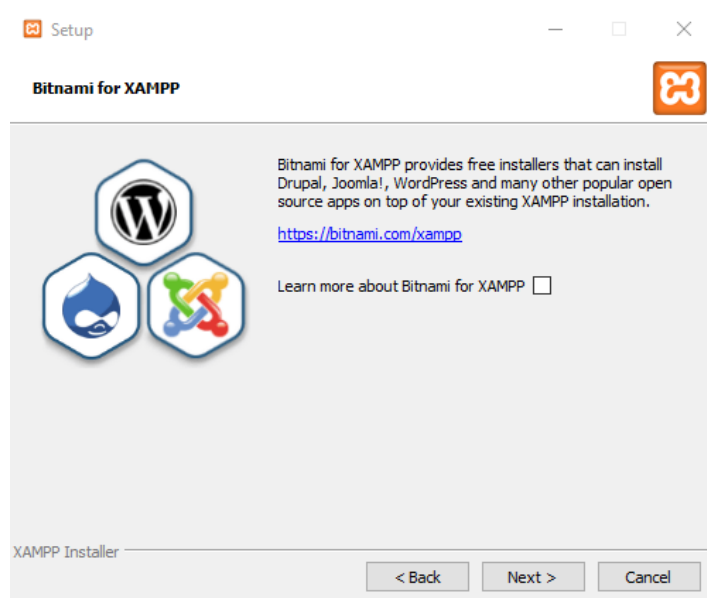


Ilustración 101. Instalación y configuración XAMPP. Instalación V

Paso 6. Esperar a que se complete la instalación y lanzar la aplicación.

Ilustración 102. Instalación y configuración XAMPP. Instalación VI

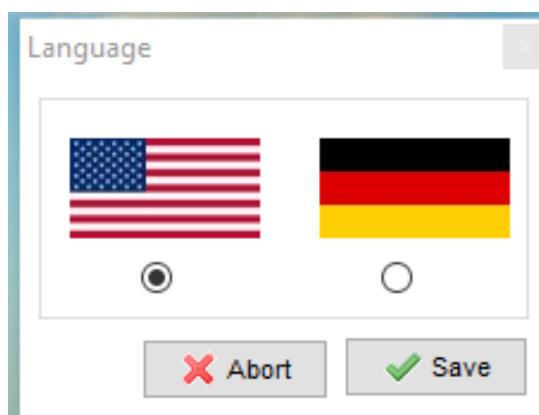
Paso 7. Seleccionar inglés como lenguaje

Ilustración 103. Instalación y configuración XAMPP. Selección lenguaje

Paso 8. Lanzar Módulos Apache y MySQL (configuración de Puertos por defecto)

La acción Admin de MySQL nos lleva a la herramienta de administración de bases de datos PhpMyAdmin, en la cual vamos a introducir nuestra base de datos.

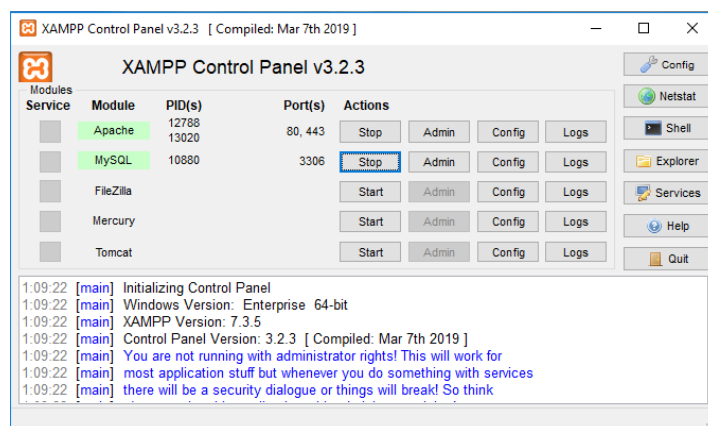


Ilustración 104. Instalación y configuración XAMPP. Pantalla principal

Paso 9. Añadir nueva base de datos, dotarla de un nombre y pulsar Crear.

Ilustración 105. Instalación y configuración XAMPP. Añadir base de datos

Paso 10. En el apartado **SQL** insertar nuestra base de datos y seguidamente, pulsar en continuar.



Ilustración 106. Instalación y configuración XAMPP. Inserción de la base de datos

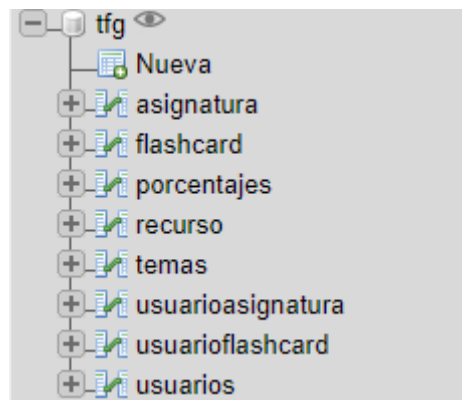
Paso 11. Recargar panel de administración

Ilustración 107. Instalación y configuración XAMPP. Panel de administración

Una vez preparada la base de datos del sistema, es necesario localizar adecuadamente nuestro proyecto dentro de XAMPP para que sea accesible desde **localhost**

Paso 12. Localizar nuestro proyecto en la carpeta 'htdocs' de la carpeta XAMPP.
En este caso, la carpeta de mi proyecto se llama TFG-final

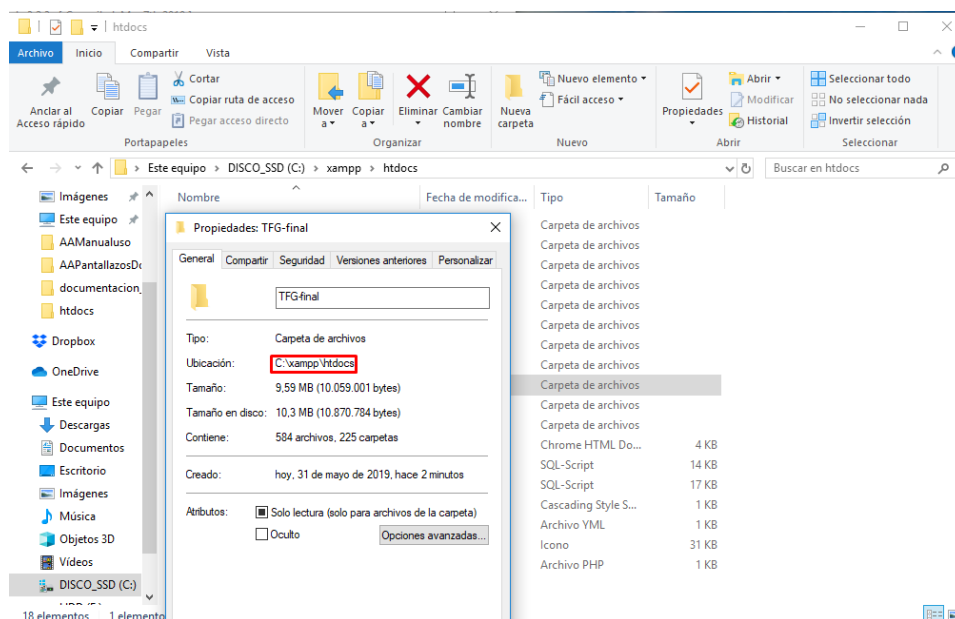


Ilustración 108. Instalación y configuración XAMPP. Ubicación del proyecto

Paso 13. Eliminar *index* por defecto de la carpeta htdocs

Acceder a localhost

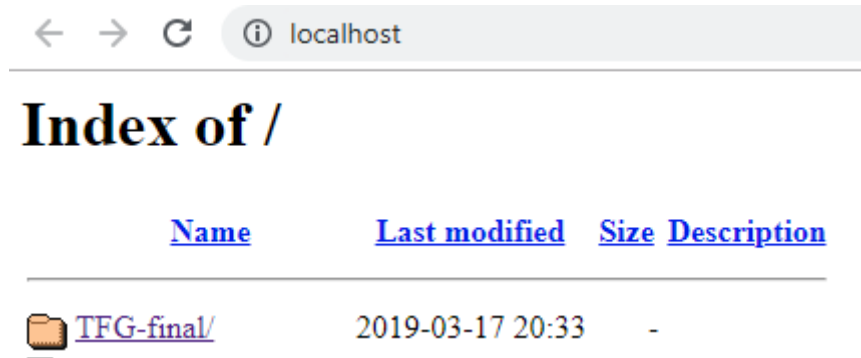


Ilustración 109. Instalación y configuración XAMPP. Acceso al proyecto

Paso 14. Acceder a la carpeta contenedora de tu proyecto para cargar el *index* de tu aplicación.

El motivo de que al acceder a *localhost* se muestre un *Index* con las carpetas con distintos proyectos es para precisamente tener acceso a diferentes proyectos software desde *localhost* con los cuales establecer conexión a través de Apache.

Bibliografía

- [1] "¿Qué es Docker? ¿Para qué se utiliza?" [online]. Disponible: <https://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>
- [2] "Las 5 etapas de la ingeniería del software" [online]. Disponible: <http://proyectosquerrilla.com/blog/2013/02/las-cinco-etapas-en-la-ingenieria-del-software/>
- [3] Pressman, Roger S., Ingeniería del Software. Vol. I. 250 p. 2005.[online]. Disponible: https://alelopij.weebly.com/uploads/9/3/6/4/936494/roger_pressman-ingeniera_del_software-v_ed-cap1.pdf
- [4] "Ingeniería del software" [online]. Disponible: https://www.ecured.cu/Ingenier%C3%ADa_de_software
- [5] "Desarrollo en cascada" [online]. Disponible: https://es.wikipedia.org/wiki/Desarrollo_en_cascada
- [6] "Características y fases del modelo incremental" [online]. Disponible: <https://www.obs-edu.com/es/blog-project-management/metodologias-agiles/caracteristicas-y-fases-del-modelo-incremental>
- [7] "Ingeniería de requisitos" [online]. Disponible: <http://www.juntadeandalucia.es/servicios/madeja/contenido/subsistemas/ingenieria/ingenieria-requisitos>
- [8] "The foundations of specification" [online]. Disponible: https://cswww.essex.ac.uk/staff/turnr/cswww.essex.ac.uk_files/Mypapers/foundations-specification.pdf
- [9] "Especificaciones de los requisitos software" [online]. Disponible: https://www.ctr.unican.es/asignaturas/is1/IEEE830_esp.pdf
- [10] "Especificaciones de requisitos software" [online]. Disponible: https://es.wikipedia.org/wiki/Especificaci%C3%B3n_de_requisitos_de_software
- [11] "Requisitos funcionales" [online]. Disponible: https://es.wikipedia.org/wiki/Requisito_funcional
- [12] "Requisitos no funcionales" [online]. Disponible: https://www.ecured.cu/Requisitos_no_funcionales
- [13] "Modelo de calidad de requisitos" [online]. Disponible: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/413>
- [14] "Método MoSCoW" [online]. Disponible: <https://www.ticportal.es/glosario-tic/metodo-moscow>
- [15] "Estimación basada en tallas" [online]. Disponible: <https://s3.eu-west-1.amazonaws.com/eu.storage.safecreative.org/1/2011/07/07/00000131/0411/ea2e/3a37/70c4a283b8dc/Estimacionbasadaentallas.pdf?response-content-type=application%2Fpdf&X->

[Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20190616T111809Z&X-Amz-SignedHeaders=host&X-Amz-Expires=86400&X-Amz-Credential=1SXTY4DXG6BJ3G4DXHR2%2F20190616%2Feu-west-1%2Fs3%2Faws4_request&X-Amz-Signature=a0b82f5f0714b869aeeaad4d5b0caac82ff4360babb92d0c1f268bcd02d3d41d](#)

[16] "Diagrama de Gantt" [online]. Disponible: https://es.wikipedia.org/wiki/Diagrama_de_Gantt

[17] "The Harmonogram of Karol Adamiecki." [online]. Disponible: <https://journals.aom.org/doi/10.5465/ambpp.1974.17530521>

[18] "Ingeniería del software, análisis y diseño" [online]. Disponible: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_if/capitulo2.pdf

[19] "MVC explicado" [online]. Disponible: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

[20] "Fundamentos y diseños de base de datos" [online]. Disponible: <http://ccdoc-basesdedatos.blogspot.com/2013/02/modelo-entidad-relacion-er.html>

[21] "Apache" [online]. Disponible: <https://www.apachefriends.org/es/index.html>

[22] "Visual Paradigm" [online]. Disponible: <https://www.visual-paradigm.com/>

[23] "HeidiSQL" [online]. Disponible: <https://www.heidisql.com/>

[24] "GitHub" [online]. Disponible: <https://github.com/>

[25] "Gantt Project" [online]. Disponible: <https://www.ganttproject.biz/>

[26] "Docker" [online]. Disponible: <https://www.docker.com/>

[27] "Bootstrap" [online]. Disponible: <https://getbootstrap.com/>

[28] "w3schools tutoriales" [online]. Disponible: <https://www.w3schools.com/>

[29] "Tecnologías usadas en entorno cliente-servidor" [online]. Disponible: <http://mialtoweb.es/tecnologias-usadas-en-entorno-cliente-y-servidor/>

[30] "5 razones por las que todo programador debería aprender JavaScript" [online]. Disponible: <https://www.campusmvp.es/recursos/post/5-razones-por-las-que-todo-programador-deberia-aprender-JavaScript.aspx>

[31] "5 razones de por qué es importante saber HTML" [online]. Disponible: <https://www.nextu.com/blog/por-que-es-importante-aprender-html/>

[32] "Qué es y para qué sirve CSS" [online]. Disponible: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo&catid=46&Itemid=163

- [33] "Visio vs Visual Paradigm" [online]. Disponible: <https://www.capterra.com/flowchart-software/compare/161413-145716/Visio-vs-Visual-Paradigm>
- [34] "Por qué usar Netbeans" [online]. Disponible: <http://www.joseluisestevez.com/index.php/2018/10/09/por-que-uso-netbeans/>
- [35] "Servidores: Ventajas y desventajas de XAMPP" [online]. Disponible: <http://pnfiservidores.blogspot.com/2013/01/ventajas-y-desventajas-de-xampp.html>
- [36] "Qué es un sistema ERP" [online]. Disponible: <https://www.ticportal.es/temas/enterprise-resource-planning/que-es-sistema-erp>
- [37] "Planificación de la producción. ¿Qué hay que tener en cuenta?" [online]. Disponible: <https://www.ticportal.es/glosario-tic/planificacion-produccion>
- [38] "Retorno de la inversión (ROI) ¿Cómo calcular los costes y beneficios?" [online]. Disponible: <https://www.ticportal.es/glosario-tic/retorno-inversion-roi>
- [39] "Intercambio electrónico de datos (EDI) ¿Por qué lo usan las empresas?" [online]. Disponible: <https://www.ticportal.es/glosario-tic/edi-intercambio-electronico-datos>
- [40] "Recogida de pedidos. ¿Cómo optimizar la gestión del almacén?" [online]. Disponible: <https://www.ticportal.es/glosario-tic/recogida-pedido-order-picking>
- [41] "¿Cómo surge el patrón MVC?" [online]. Disponible: https://fcharte.com/tutoriales/00_patronmvc/
- [42] "¿Qué son los mockups y como se hacen?" [online]. Disponible: <http://www.ticarte.com/contenido/que-son-los-mockup-y-como-se-hacen>
- [43] "Imagen palabras clave ingeniería del software" [online] Disponible: https://es.123rf.com/photo_58194257_stock-vector-software-engineering-chart-with-keywords-and-icons-flat-design.html