



GRADO EN INGENIERÍA INFORMÁTICA

Escuela Técnica Superior de Ingeniería Informática

Curso académico 2021-2022

Trabajo fin de grado

ForeverLearn:

Un espacio virtual para la educación centrado en cMOOC

Tutor: Oriol Borrás Gené

Autor: José Justo Tena Agudo



Este trabajo se distribuye bajo los términos de la licencia internacional CC BY-NC-SA International License (Creative Commons AttributionNonCommercial-ShareAlike 4.0). Usted es libre de *(a) compartir*: copiar y redistribuir el material en cualquier medio o formato; y *(b) adaptar*: remezclar, transformar y crear a partir del material. El licenciador no puede revocar estas libertades mientras cumpla con los términos de la licencia:

- *Atribución.* Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatante.
- *No comercial.* Usted no puede hacer uso del material con propósitos comerciales.
- *Compartir igual.* Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la la misma licencia del original.

Documento de José Justo Tena Agudo.

Agradecimientos

Gracias a mi madre por ayudarme a ser el hombre que soy, a mi abuela Jacinta por dedicar su vida a criarme, a mi hermana por escucharme siempre, a mi padre por sus esfuerzos para que pueda vivir una buena vida y a mi prima Cintia por estar siempre atenta a mí.

Agradezco también al resto de mi familia por su apoyo incondicional y a mis amigos de Almendralejo, Corte de Peleas y Madrid, especialmente a Adrián, vuestras risas siempre me llenan de fuerza para seguir adelante.

Por último gracias a mi tutor Oriol, sin tus ideas y tu atención este proyecto no hubiera sido posible.

Aunque todavía quede trabajo por hacer, una parte de ForeverLearn 1 ya es un realidad y es gracias a todos vuestros ánimos.



Figura 1: Logo de ForeverLearn

Resumen

ForeverLearn es un proyecto de innovación educativa que persigue crear una plataforma tanto para la impartición como el aprendizaje basado en MOOCs, ofreciendo herramientas para la creación y el aprendizaje de este tipo de cursos.

El proyecto se enmarca en la iniciativa de los REA, eliminando cualquier característica que impida el acceso a los cursos, por ello estos serán gratuitos y se podrá acceder a todo su contenido sin la necesidad de registrarse. La plataforma se ha centrado en los cMOOC, una variante de los MOOC que pone su énfasis en la comunidad que se puede crear alrededor del curso debido a su gran número de alumnos.

ForeverLearn ofrece la libertad a sus usuarios de ser tanto profesores como alumnos. Como profesores tendrán a su disposición múltiples herramientas de creación y gestión del contenido de sus cursos, como la posibilidad de insertar contenido embebido de otros sitios web. Como alumnos podrán valorar, matricularse y gestionar su lista de cursos deseados.

Este primer trabajo sirve por tanto para establecer una plataforma de cursos en abierto conforma a la modalidad MOOC y los principios del movimiento REA, que permita a sus usuarios ser profesores creando cursos, alumnos pudiendo recibir la formación de estos y cuente con múltiples opciones de administración.

Esta labor se realizará teniendo en cuenta que debe servir como una base que será expandida en un segundo trabajo de fin de grado que vendrá a continuación. El segundo trabajo se enfocará principalmente en la innovación educativa, implementando un sistema de evaluación de los cursos basado en la participación de los usuarios, motivada mediante técnicas de Gamificación, en pos de crear una red de conocimiento que envuelva al curso. Con todo lo planteado el objetivo es crear un sistema que permita la educación a través de cMOOC, que integre las últimas corrientes educativas y que suponga una alternativa viable al tipo de evaluación clásica.

Acrónimos

MOOC *Massive Open Online Course*

xMOOC *eXtended Massive Open Online Course*

cMOOC *connectivist Massive Open Online Course*

REA *Recursos Educativos Abiertos*

OCW *OpenCourseWare*

TFG *Trabajo de Fin de Grado*

IPO *Interacción Persona Ordenador*

DCU *Diseño Centrado en el Usuario*

Índice general

1. Introducción	1
1.1. Motivación	2
1.2. Estructura del documento	2
2. Objetivos	5
2.1. Descripción del problema	5
2.2. Objetivos de la aplicación	6
2.3. Estado del arte	7
2.4. Estudio de alternativas	8
2.5. Tecnologías utilizadas	11
2.6. Metodología	14
3. Descripción informática	18
3.1. Perfil de usuario	18
3.2. Actores, roles y organización	18
3.3. Lista de tareas	19
3.4. Descripción de requisitos	20
3.5. Casos de uso	25
3.6. Arquitectura de alto nivel	26
3.7. Diseño de la base de datos	27
3.8. Diseño de la interfaz y experiencia de usuario	31
3.9. Implementación	31
4. Experimentos y validación	37
4.1. Pruebas unitarias	37
5. Conclusiones	41
5.1. Conclusiones generales	41
5.2. Visión de futuro a corto plazo	43
5.3. Visión de futuro a largo plazo	44

6. Bibliografía	46
A. Páginas del sistema	50
B. Casos de uso	64

Índice de figuras

1.	Logo de ForeverLearn	III
2.1.	Diagrama de Gantt de la duración de las fases del proyecto.	16
3.1.	Diagrama de casos de uso	26
3.2.	Diagrama de arquitectura de alto nivel	27
3.3.	Diagrama de la base de datos	30
3.4.	Diagrama de navegación entre pantallas	32
3.5.	Diagrama de arquitectura del backend	34
3.6.	Diagrama de arquitectura del backend con clases	35

Índice de cuadros

2.1. Diferencias entre alternativas.	10
B.1. Registrarse.	64
B.2. Iniciar sesión.	65
B.3. Cerrar sesión.	65
B.4. Configurar cuenta.	66
B.5. Editar información básica.	66
B.6. Editar información de perfil.	67
B.7. Cambiar contraseña.	67
B.8. Borrar cuenta.	68
B.9. Ver listado de cursos.	68
B.10.Ver cursos por categoría.	69
B.11.Ver cursos por título.	69
B.12.Ver curso.	70
B.13.Ver perfil de usuario.	70
B.14.Acceder al área de profesor.	71
B.15.Crear curso.	71
B.16.Editar curso.	72
B.17.Borrar curso.	72
B.18.Crear tema.	73
B.19.Editar tema.	74
B.20.Borrar tema.	75
B.21.Crear lección.	76
B.22.Editar lección.	77
B.23.Borrar lección.	78
B.24.Crear requisito.	79
B.25.Editar requisito.	80
B.26.Borrar requisito.	81
B.27.Crear objetivo.	82

B.28.Editar objetivo.	83
B.29.Borrar objetivo.	84
B.30.Acceder al área de alumno.	84
B.31.Matricularse.	85
B.32.Desmatricularse.	86
B.33.Ver listado de cursos matriculados.	86
B.34.Añadir a deseados.	87
B.35.Eliminar de deseados.	88
B.36.Ver listado de cursos deseados.	88
B.37.Ver listado de cursos completados.	89
B.38.Dar me gusta.	89
B.39.Quitar me gusta.	90
B.40.Dar no me gusta.	91
B.41.Quitar no me gusta.	92
B.42.Acceder al área de administrador.	92
B.43.Administrar usuarios.	93
B.44.Administrar usuarios por nombre.	93
B.45.Banear usuario.	94
B.46.Desbanear usuario.	94
B.47.Eliminar usuario.	95
B.48.Administrar cursos.	95
B.49.Administrar cursos por nombre.	96
B.50.Banear curso.	97
B.51.Desbanear curso.	97
B.52.Eliminar curso.	98
B.53.Administrar categorías.	98
B.54.Administrar categorías por nombre.	99
B.55.Crear categoría.	99
B.56.Editar categoría.	100
B.57.Eliminar categoría.	101

Capítulo 1

Introducción

Los MOOC [1] son cursos masivos, abiertos y en línea que buscan llevar la educación a un número ilimitado de alumnos, con este fin se sirven de internet para reunir el conocimiento a impartir y dar acceso a todo aquel que desee aprenderlo.

Existen principalmente dos divisiones de MOOCs [16]: los xMOOC que establecen una interacción limitada entre profesor y alumno y se evalúan mediante pruebas cerradas, y los cMOOC que se enfocan en la creación de una red que va más allá de lo que el profesor enseña y que conecta personas, ideas y recursos de los alumnos u otros autores.

Este proyecto tiene como objetivo crear un espacio o aula virtual que permita una correcta enseñanza a través de MOOC, concretamente a través de su variante cMOOC.

El aula virtual desarrollada cuenta además con el valor añadido de una fuerte apuesta por la apertura de la educación, como se ve en estas dos decisiones:

- *Libertad de acción.* En este espacio cualquier usuario puede ser profesor, creando y administrando cursos, o ser alumno, accediendo y participando en los mismos. Sin poner límites de ningún tipo al rol que nuestros usuarios deseen desempeñar en el sistema.
- *Contenido abierto.* Los cursos estarán totalmente abiertos, permitiendo que cualquier usuario, incluso sin registrarse ni inscribirse, pueda acceder a la totalidad de su contenido.

Con este último punto la plataforma pretende continuar la línea de los REA, recursos educativos abiertos, una iniciativa que busca popularizar la publicación de documentos con fines relacionados con la educación o investigación, de acceso libre y licencia abierta.

Para solventar los principales obstáculos de los cMOOC: la creación de una fuerte comunidad entre alumnos y la evaluación del conocimiento adquirido por estos, en un

siguiente trabajo de fin de grado se implementará una solución común a ambos factores que sigue la política de abrir al mundo tanto conocimiento como podamos.

La solución se encuentra en un sistema de foros que acompañe a los cursos y que permita a los alumnos traer nuevas conversaciones a la mesa de debate, proponer y resolver dudas, así como aportar nuevas ideas e información en un espacio común para todos ellos. Los foros se podrán leer sin la necesidad de registrarse y nos sirven para crear la deseada comunidad, además será en base a la participación de los alumnos en estos como calificaremos su desempeño en el curso.

1.1. Motivación

Los cMOOC aprovechan el gran número de alumnos de los MOOC para crear una red de conocimiento que de otro modo sería imposible, mientras que los xMOOC se limitan a implementar en línea los métodos educativos clásicos. Pero debido a las dificultades técnicas de implementar la propuesta de los cMOOC, apenas llegaron a ser una opción viable a tener en cuenta, por ello existe un gran aliciente en realizar el esfuerzo de desarrollar una plataforma que sea capaz de dar su lugar a este tipo de cursos.

Otro importante factor que incentivó este trabajo fue investigar como funcionan las actuales herramientas para la enseñanza a través de MOOC y comprobar que la mayoría de las soluciones no perseguía potenciar el carácter abierto de este tipo de cursos, dejando clara la necesidad e importancia de la existencia de un portal que de verdad apueste por abrir su contenido al mundo, contemplando todas las posibilidades en el ámbito educativo que una decisión así podía suponer.

Con la ilusión de crear un sistema abierto que aproveche las posibilidades de los cMOOC, se decidió crear una plataforma que permitiera la impartición de manera real y efectiva de este tipo de cursos y que aprovechara los nuevos recursos digitales en pos de crear una red de conocimiento alrededor de los cursos.

1.2. Estructura del documento

Este primer punto de la memoria busca servir como introducción a aquellos que deseen conocer más acerca del proyecto y contextualizar del trabajo que se está realizando. Será en el resto de secciones donde se encontrará una mayor información sobre el proyecto.

Tras esta toma de contacto, se detallarán en el segundo punto una serie de aspectos que rodean al proyecto, comenzando por una puesta en contexto más detallada del problema, las soluciones alternativas existentes y una exposición del proceso que seguiremos para llevar a cabo el trabajo.

Una vez visto el proyecto desde fuera y su organización, se tratará en el tercer punto el grueso de la memoria. En él se verá el desarrollo desde una perspectiva técnica, comentando cuales son los usuarios, requisitos de la aplicación, casos de uso del sistema, la arquitectura que se ha seguido, el diseño de la interfaz y la implementación del mismo.

La explicación técnica viene seguida de un cuarto punto en el que se habla a cerca de las pruebas que se han realizado para asegurar la calidad del diseño e implementación del sistema.

Finalmente tendremos un quinto punto en el que se desarrollan las conclusiones extraídas del trabajo realizado y se pondrá la vista hacia delante para reflexionar acerca de que le depara al proyecto en el futuro, tanto a corto como a largo plazo.

Capítulo 2

Objetivos

Si disfrutaste aprendiendo, no hay razón por la que debas detenerte a una cierta edad. [17]

Isaac Asimov.

2.1. Descripción del problema

La impartición de cMOOC puede llegar a ser un completo desafío [9], pues desde su propio concepto trae consigo una serie de dificultades y limitaciones intrínsecas que han obstaculizado su expansión y desarrollo más allá de un planteamiento teórico.

Se debe poner una especial atención al acceso abierto a la educación [10], queremos derrumbar todas aquellas barreras que limiten el acceso a la enseñanza en nuestra plataforma y poner un esfuerzo en abrir nuestros contenidos de manera coherente al mayor público posible.

Una de las barreras que dificultan el acceso es la falta de competencias digitales en los usuarios, el sistema debe ser diseño atendiendo a la accesibilidad y usabilidad, permitiendo que los usuarios puedan aprender y utilizar las distintas herramientas a su disposición sin encontrar dificultades.

El principal problema reside en la necesidad de crear un espacio en el que puedan impartirse los cMOOC, se necesita por tanto una plataforma base que permita por un lado a los profesores poder crear cursos y añadir el contenido que crean necesario, y por otro lado que los alumnos puedan encontrar cursos de su interés con facilidad, inscribirse y acceder al contenido de estos.

Además el proyecto pretende enfrentar los problemas encontrados en las distintas alternativas actuales: cursos cerrados mediante pagos, limitaciones en las funcionalidades

de profesores y alumnos y fuerte separación de roles.

Los problemas referentes a la creación y mantenimiento de la comunidad formada por los alumnos serán tratados y resueltos en un próximo TFG que expandirá los planteamientos expuestos en este y completará de manera definitiva el proyecto global.

2.2. Objetivos de la aplicación

ForeverLearn propone una serie de objetivos imprescindibles para dar por satisfechas las inquietudes y desafíos que nos plantea el problema expuesto. Para ello planteamos un objetivo principal, que se corresponde directamente con el corazón de nuestra propuesta: promover una filosofía de aprendizaje abierto, accesible por y para todos, a través de la cual nuestros usuarios puedan crecer profesional y personalmente.

Tenemos además una serie de objetivos secundarios que sirven como apoyo en la consecución de nuestra principal meta y que nos ayudan a definir la forma del sistema que deseamos crear.

- Conceder acceso al contenido de los cursos de manera gratuita y sin necesidad de registrarse, esto se hace con el fin de alcanzar una educación genuinamente abierta para todos los interesados en adquirir conocimiento.
- Garantizar que cualquier usuario registrado pueda realizar todas las funciones que corresponderían tanto al rol de profesor como al de alumno, dándole la libertad de interactuar con el sistema en base a sus intereses y necesidades.
- Permitir que el profesor pueda diseñar sus cursos como desee, para ello dispondrá de herramientas para poder dividir los cursos en los módulos o temas que considere y así poder estructurar el contenido del mismo.
- Facilitar la búsqueda de cursos, para ello existirá una sección a modo de tienda que permita ver todos los cursos disponibles, así como filtrar por categorías o buscar en base a su nombre.
- Aportar las herramientas necesarias para que los alumnos puedan inscribirse en cursos, valorarlos y gestionar su lista de cursos deseados.

Además de los objetivos propios del problema a resolver, se plantean otros motivos que resultan un fuerte aliciente para el desarrollo de la aplicación.

- Demostrar todo lo aprendido a lo largo del grado a través de un proyecto de gran envergadura que me desafíe a mí mismo a utilizar todos los recursos adquiridos en materia de gestión de proyectos, desarrollo web, algoritmia, arquitectura de software, diseño de interfaces, gestión de bases de datos y muchas otras áreas de conocimiento impartidas en la carrera.
- Enfrentar nuevos obstáculos y desafíos que me enriquezcan como persona, me animen a adquirir nuevos conocimientos y me motiven a seguir formándome y mejorando como profesional del desarrollo de software.

2.3. Estado del arte

Los MOOC [1] son cursos masivos, abiertos y en línea que buscan llevar la educación a un número ilimitado de alumnos, con este fin se sirven de internet para reunir el conocimiento a impartir y dar acceso a todo aquel que desee aprenderlo. Los MOOC se escindieron en dos variantes [2] según pusieran su foco:

- *xMOOC*. El principal interés de este tipo de cursos se trata de su contenido, desarrollado únicamente por el profesor, busca que los alumnos adquieran un conocimiento técnico o científico de alto nivel.

Dada su condición, frecuentemente se examina a través de pruebas de tipo test, que permitan asegurar de manera automática que el usuario ha realizado un correcto aprendizaje, puesto que si tuvieran que evaluarse manualmente requerirían de un gran esfuerzo por parte del profesor debido al volumen de alumnos.

La popularidad de este tipo de MOOC ha explotado principalmente por su facilidad de elaboración y evaluación, provocando que se conviertan en la variante hegemónica de MOOCs que han adoptado las principales plataformas de cursos.

- *cMOOC*. Los primeros MOOC pertenecen a este tipo, se centran en aprovechar el gran volumen de alumnos que tienen para crear una comunidad basada en el conocimiento y enriquecer la experiencia de los participantes gracias a las relaciones nacidas de su interacción [3].

Sin embargo debido a la naturaleza de estos, resultan muy difíciles de evaluar basándose en métodos tradicionales, sean exámenes o prácticas. Se requiere de

nuevas formas de evaluación que todavía no se han instaurado [4].

Una de las características de los cMOOCs es que su contenido está repartido en distintos nodos de una amplia red [5], la información se encuentra distribuida en foros, páginas webs o redes sociales de los alumnos matriculados en el curso. La plataforma que pretenda alojar cMOOCs debe poner especial atención en permitir que sea sencillo seguir todo este entramado de contenido tan disperso.

Todos estos obstáculo desalentaron el crecimiento de esta variante [6], desplazándola hasta casi hasta ser más un concepto que una alternativa real a los xMOOCs.

Los REA [7] son una iniciativa que busca popularizar la publicación de documentos con fines relacionados con la educación o investigación, de acceso libre y licencia abierta.

Una parte de los REA son los OCW [8], que consisten en publicaciones digitales gratuitas y abiertas organizadas como cursos y pueden incluir material para su planificación, evaluación u otro contenido temático. Como podemos ver este planteamiento se acerca más a nuestra idea de MOOC cuyo contenido se encuentra abierto para cualquier usuario del sistema.

2.4. Estudio de alternativas

En esta sección valoraremos la viabilidad real de las opciones disponibles en el mercado [11] de cara a resolver el problema que nos ocupa y cómo encajan con nuestros objetivos.

Estrictamente no existe ninguna plataforma real dedicada a la impartición de cMOOC debido a las dificultades de evaluación ya presentadas, sin embargo existen múltiples sitios dedicados a la impartición de xMOOC. Debido a que en este trabajo solo se va a desarrollar el aula virtual para los cursos, sin entrar en materia de evaluación de los mismos, podemos comparar los servicios que ofrecen las demás opciones en este aspecto. Para el desarrollo de este apartado se ha realizado un estudio de las siguientes plataformas virtuales para la impartición de MOOC: Coursera¹, edX², MiriadaX³, Udemy⁴ y Canvas Network⁵.

¹<https://www.coursera.org/>

²<https://www.edx.org/>

³<https://formacion.miriadax.net/>

⁴<https://www.udemy.com/>

⁵<https://www.canvas.net/>

Entre las distintas opciones contempladas no encontramos fuertes particularidades, dado que todas tienen la misma finalidad ofrecen servicios muy parecidos, pues saben que características son necesarias en su plataforma y cualquiera que no lo sea es descartada. Entre las características más comunes se encuentran:

- Expositor de cursos que incluye un buscador por palabras y dividido por temáticas, dificultad del nivel educativo y otras categorías como precio, entidad emisora, etc. Dependiendo del tamaño de la plataforma los expositores tiene un mayor peso a la hora de guiar al alumno en sus intereses o de dar visibilidad a los cursos.
- Perfil en el que puedes acceder y modificar tus datos personales, un listado de los cursos en los que estás inscrito y otro listado en el que figuran aquellos completados.
- Cesta de la compra donde se depositarán los cursos comprados y una lista de deseos en la que el usuario pueda incluir aquellos que sean de su interés.
- Apartado para empresas con los planes de pago para compras de un gran volumen de cursos y un gestor de seguimiento que permite obtener información sobre el avance de los empleados en los cursos comprados.

Aunque no existan grandes diferencias de conceptos entre los portales, sí podemos observar una serie de factores que ayudan a clasificarlos:

- *Precio de los cursos.* Es cierto que un gran porcentaje de los cursos de las plataformas comerciales son gratuitos, pero dependiendo de la plataforma concreta puede haber un mayor o menor porcentaje de cursos de pago.

Existen plataformas como Coursera y edX, respaldadas por las universidades de mayor prestigio y las empresas más exitosas, cuyos cursos de niveles educativos superiores pueden superar con mucha facilidad los 1000 euros.

Otras de menor tamaño como MiriadaX y Udemy, tienen entre su profesorado excelentes profesionales pero no alcanzan el nivel de las anteriores, en este caso incluso los cursos más avanzados suelen oscilar entre los 100 y los 300.

A pesar de estos datos, siguen existiendo plataformas como Canvas Network en las que todos sus cursos son gratuitos, a pesar de que esto suele traducirse en un detrimento del número y variedad de los cursos.

- *Profesorado de los cursos.* En este sentido se repite una división parecida al punto anterior, con tres claros niveles de libertad.

Encontramos un escaso nivel de libertad en las grandes plataformas como Coursera o edX que solo admiten a las mejores universidades y empresas para dar sus cursos, manteniendo así su prestigio.

De aquí pasamos a un nivel medio como MiriadaX y Canvas Network en la que solo pueden impartir cursos instituciones, universidades y escuelas de negocio que previamente se hayan puesto en contacto y hayan sido aprobadas por ellos.

Finalmente tenemos el caso de Udemy que nos ofrece una gran libertad, permitiendo a cualquier usuario ser profesor e impartir su propio curso.

Una vez desarrolladas las características y diferencias de las plataformas profesionales estudiadas se expone en la Tabla 2.1 un resumen de esta información y se da paso a un análisis de la viabilidad de estas como alternativas al sistema que se propone en este trabajo.

Plataforma	Precio	Profesorado
Coursera	Los más altos superan los 1000€	Solo reputadas universidades y empresas
edX	Los más altos superan los 1000€	Solo reputadas universidades y empresas
Miriada	Los más altos oscilan entre 100€ y 300€	Solo instituciones aprobadas
Udemy	Los más altos oscilan entre 100€ y 300€	Cualquier usuario
Canvas Network	Todos gratuitos	Solo instituciones aprobadas

Cuadro 2.1: Diferencias entre alternativas.

En primer lugar, todas aquellas plataformas que cobren dinero a sus usuarios para acceder a cualquiera de sus cursos dejan de ser una alternativa viable, el carácter abierto de los MOOCs persigue llevar la educación al mayor número posible de usuarios y la idea de cobrar por los cursos puede impedir a un gran porcentaje del posible alumnado acceder a ellos. Si queremos que nuestro principal valor sea la comunidad de alumnos que surja entorno al curso, carece de sentido limitar la puerta de entrada a estos poniendo un precio

que muchos usuarios no puedan pagar.

En segundo lugar, referente al carácter abierto de los MOOCs, en nuestro proyecto no queremos que este carácter se limite solo a una educación para todos, también deseamos que sea posible una educación por todos, en la que todo aquel que tenga algo que enseñar al mundo pueda encontrar las herramientas necesarias en nuestra plataforma para ello. Siendo consecuentes con esta decisión deberíamos descartar la mayoría de plataformas actuales, pues por lo general poseen un férreo control sobre la posibilidad de impartir cursos.

Además de todo lo anteriormente mencionado, estas páginas tienen un factor común al que nos oponemos: solo los usuarios registrados pueden acceder a la totalidad del contenido de los cursos, si buscamos abrir la educación sería contraproducente esconderla detrás de un registro. Esta decisión sigue los principios de la iniciativa Encorsetare que incentiva la publicación de materiales docentes como “contenidos abiertos”.

Como se puede observar, ninguna de las alternativas estudiadas cumple con los requisitos de educación abierta y masiva hasta el punto en el que nosotros deseamos, y por consiguiente no existe actualmente una alternativa real que solucione el problema planteado.

2.5. Tecnologías utilizadas

En este punto se presentan todas las tecnologías que han sido necesarias en el proceso de desarrollo de este proyecto, desde las más relacionadas con la codificación del aula, a otras auxiliares al mismo que han servido como ayuda al programador o para completar otros objetivos.

- *HTML*⁶. Es un lenguaje de marcado para la creación de documentos de hipertexto, se utilizará para definir la estructura y elementos de las páginas web que conformarán el frontend de la aplicación.
- *CSS*⁷. Es un lenguaje de reglas de estilo, se utilizará para definir el diseño que se aplica a los elementos definidos en HTML.

⁶<https://developer.mozilla.org/es/docs/Web/HTML>

⁷<https://developer.mozilla.org/es/docs/Web/CSS>

- *JavaScript*⁸. Es un lenguaje de programación que nos permite implementar funciones y lógica en las páginas web, es decir, aporta dinamismo e interacción al frontend.
- *Java*⁹. Es un lenguaje de programación orientado a objetos, se ha decidido utilizar este lenguaje para programar el backend debido a su robustez, su capacidad de realizar cualquier tipo de proyecto gracias a sus múltiples librerías y el constante crecimiento y evolución de su entorno tecnológico.
- *IntelliJ*¹⁰. El entorno de desarrollo elegido para el proyecto ha sido IntelliJ, la decisión ha sido tomada en base a múltiples factores como su soporte para las últimas versiones de Java, su completado de código inteligente de gran ayuda, admite múltiples frameworks, en concreto ofrece una gran compatibilidad con Spring Boot que es el que utilizaremos y el acceso a un alto número de plugins de utilidad.
- *SQL*¹¹. Es un lenguaje de consulta estructurado que permite manipular y acceder a información de una base de datos, se ha elegido este lenguaje debido a la importancia de las relaciones entre los datos de nuestro sistema, sean profesores, cursos o alumnos, es casi imposible que una entidad no esté relacionada con otra en una aplicación como la que nos ocupa.
- *MySQL*¹². Es un sistema de gestión de bases de datos relacionales, se ha decidido utilizar este gestor para la persistencia de los datos debido a que proporciona soporte completo para todas las necesidades del desarrollo de aplicaciones, alto rendimiento, escalabilidad y flexibilidad.
- *Maven*¹³. Es una herramienta que estandariza la configuración de un proyecto en todo su ciclo de vida, empaquetaremos el proyecto con él y se encargara de gestionar y descargar todas las dependencias del mismo cuando deba ser ejecutado en una nueva máquina.
- *Git*¹⁴. Es una herramienta de control de versiones de código de forma distribuida, es el instrumento estándar para ello y, pese a que al ser un trabajo individual no le sacaremos el máximo partido, nos servirá para tener un mejor control del avance del proyecto.

⁸<https://developer.mozilla.org/es/docs/Web/JavaScript>

⁹<https://www.java.com/es/>

¹⁰<https://www.jetbrains.com/es-es/idea/>

¹¹<https://developer.mozilla.org/es/docs/Glossary/SQL>

¹²<https://www.mysql.com/>

¹³<https://maven.apache.org/>

¹⁴<https://git-scm.com/>

- *GitHub*¹⁵. Es un portal que permite alojar a través de repositorios código que utiliza el sistema Git, nos permitirá tener una copia de seguridad del proyecto y apreciar con mayor claridad el desarrollo del mismo gracias a sus múltiples vistas y métricas.
- *Spring Boot*¹⁶. Es una herramienta que facilita la inicialización y creación de proyectos Spring, el cual es el framework por excelencia para Java debido a sus multitud de características útiles en el desarrollo de aplicaciones como la inyección de dependencias y módulos propios.
- *Moustache*¹⁷. Es un motor de plantillas web que nos permite insertar y visualizar en el frontend información proveniente del backend de la aplicación.
- *Lombok*¹⁸. Es una librería de Java que permite eliminar código repetitivo y generar métodos como gets, sets, toString, etc, a través de anotaciones, se integra de manera correcta con Maven y pese a no ser imprescindible mejora la limpieza del código.
- *JUnit*¹⁹. Es la librería de testing por excelencia para Java, permite establecer métodos que prueban funcionalidades del sistema comparando la salida esperada con la que se obtiene realmente.
- *Mockito*²⁰. Es un framework de testing que nos permite establecer que respuestas nos devuelven métodos del sistema, por tanto se utiliza para simular fragmentos de código que no deseamos ejecutar de verdad.
- *H2*²¹. Es una base de datos integrada escrita en Java que se ejecuta en memoria, no puede usarse para implementar la persistencia de datos, pero ofrece las mismas funcionalidades que una base de datos persistente, por lo que resulta una buena herramienta en las fases de desarrollo y testing.
- *Hibernate*²². Es una herramienta de mapeo objeto-relacional para Java que nos permite realizar conversiones entre clases de Java y esquemas de una base de datos relacional tradicional.

¹⁵<https://github.com/>

¹⁶<https://spring.io/projects/spring-boot>

¹⁷<https://mustache.github.io/>

¹⁸<https://projectlombok.org/>

¹⁹<https://junit.org/junit5/>

²⁰<https://site.mockito.org/>

²¹<https://www.h2database.com/>

²²<https://hibernate.org/>

2.6. Metodología

Inicialmente se pensó que el desarrollo del proyecto siguiera un proceso de Diseño Centrado en el Usuario (DCU), dado que nuestro objetivo es llevar la educación al mayor número posible de gente sin poner ningún límite, cabe esperar que el sistema sea visitado por una gran cantidad de usuarios de todo tipo y por tanto es imprescindible que el diseño sea accesible para todos ellos y que sepa dar respuesta a sus necesidades.

Sin embargo, en una primera fase de documentación se advirtió que este tipo de sistemas siguen unas pautas muy claras, diferenciándose en pequeños detalles que definen la estrategia de cada sitio. En el caso que nos ocupa, los pequeños detalles pasarán por reforzar nuestra apuesta por el carácter abierto de la educación y el verdadero valor diferencial estará en la evaluación a través del sistema de foros que extenderá este trabajo en un futuro TFG.

A donde quiero apuntar es a que, dado que este trabajo es más bien una base sobre la que se cimentará la verdadera innovación, podemos aprovechar el trabajo realizado por nuestras alternativas en materia de diseño y requisitos de sus usuarios, sin perder tiempo en la realización de una serie de prototipos y encuestas que, con total seguridad, apuntarían en la misma dirección que los estudios realizados por las alternativas.

En conclusión, se ha decidido descartar un Diseño Centrado en el Usuario, aprovecharemos el conocimiento sobre las alternativas y modificaremos todos aquellos puntos que entren en conflicto con nuestros objetivos.

Una vez aclarado el punto anterior, trataremos cual será el proceso de desarrollo del proyecto, consistente en un desarrollo por fases, concretamente se llevarán a cabo cinco fases que se explican a continuación y cuyo avance se puede seguir en el repositorio de GitHub accesible a través del siguiente enlace: <https://github.com/jj-tena/ForeverLearn>, para un mejor seguimiento del proyecto se creará un tag de cada fase de la 1 a la 4.

1. *Documentación.* Esta fase tiene como objetivo realizar una extensa documentación de cara a entender el contexto y la naturaleza del proyecto. Para ello se ha investigado la historia de la disciplina y su situación actual, la viabilidad del proyecto y el estado de las alternativas. Sobre todo se ha centrado en conocer las corrientes educativas que se desean integrar: cMOOC, Recursos Educativos en Abierto y Gamificación.

2. *Definición.* En esta primera fase entraremos en materia con el proyecto, para ello definiremos los objetivos que se pretenden alcanzar, los tipos de usuarios del sistema, así como los requisitos que debe cumplir para dar por satisfechas sus necesidades y comenzaremos a plantear los aspectos más técnicos del proyecto: lenguajes, entornos, entidades de la base de datos y herramientas. Una vez completadas estas dos primeras fases se creará un repositorio de código libre que alojará el avance actual de la memoria, y en las siguientes fases comenzará el desarrollo del código de la aplicación que usará este mismo repositorio para subir los avances del proyecto.
3. *Frontend.* En esta fase se construirá el frontend de nuestra aplicación, por tanto se realizará la maquetación de páginas web y el esquema de navegación entre pantallas. Para ello se diseñará y desarrollará la interfaz de la aplicación teniendo en cuenta los principios de usabilidad y accesibilidad, así como todo lo aprendido de las alternativas que dominan el panorama actual.
4. *Backend.* En esta fase se implementará el backend de nuestra aplicación, configuraremos la base datos y le daremos una funcionalidad completa a toda la lógica detrás de la aplicación.
5. *Testing.* Esta será la fase final del proyecto, con toda la funcionalidad implementada, solo quedará trabajar en la documentación del sistema a través de la memoria y realizar las pruebas de calidad y testing que aseguren que el proyecto opera como es debido y no presenta errores.

Finalmente en la Figura 2.1 podemos apreciar como se han ido sucediendo las fases y la evolución temporal que ha tenido el proyecto a lo largo de las semanas, cabe destacar la diferencia de duración entre las fases.

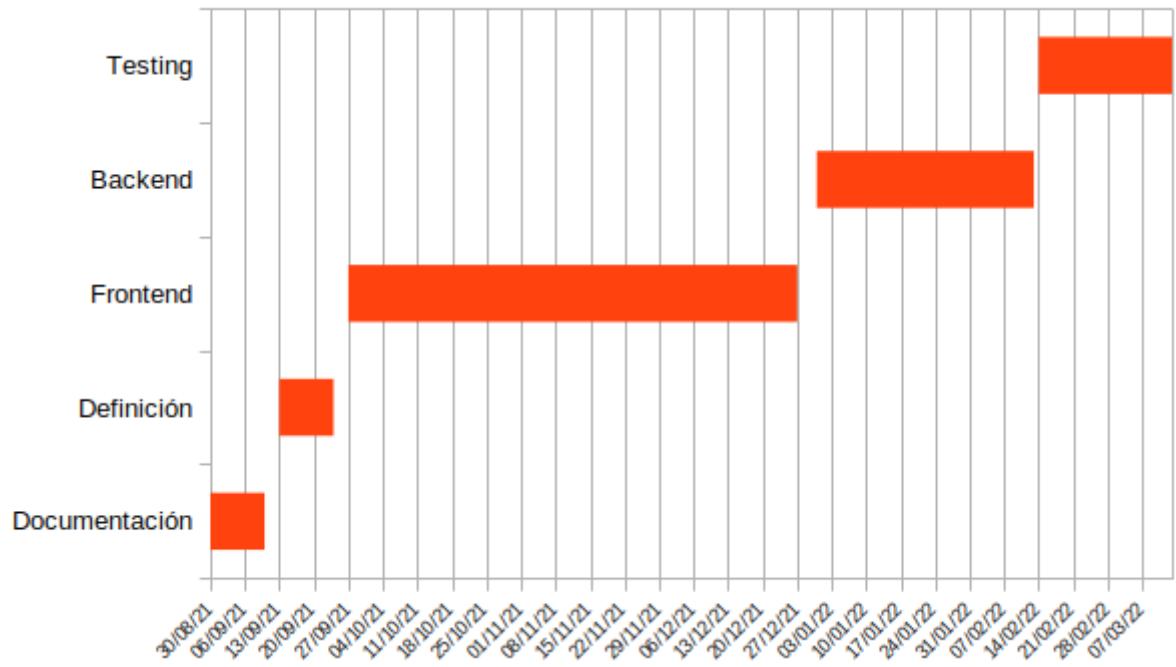


Figura 2.1: Diagrama de Gantt de la duración de las fases del proyecto.

Capítulo 3

Descripción informática

3.1. Perfil de usuario

El usuario del sistema dispondrá de una serie de datos básicos que le caracterizan: nombre, apellidos, correo electrónico, contraseña, foto de perfil y descripción.

El sistema permite a sus usuarios ser tanto profesores como alumnos, por lo tanto todos los usuarios contarán con información para ambos roles: listado de cursos propios, listado de cursos en los que está matriculado, lista de cursos deseados, lista de cursos completados, lista de cursos que ha valorado positivamente y lista de cursos que ha valorado negativamente.

En puntos anteriores se indicó que uno de los problemas de los cMOOCs era la distribución de su contenido, por ello los usuarios podrán introducir datos que faciliten su seguimiento: información de contacto, nombre de usuario de Facebook, nombre de usuario de Twitter y canal de YouTube.

3.2. Actores, roles y organización

La plataforma debe poder responder a las necesidades de sus usuarios y para ello necesita comprender que no todos interactuarán con el sistema del mismo modo, traduciéndose en un estudio de la división que se da entre estos y las particularidades de cada tipo. El sistema contempla en primer lugar tres posibles roles para sus usuarios:

- *Usuario no registrado.* Se trata del usuario que entra en el sistema por primera vez o que todavía no ve incentivos suficientes en el mismo como para decidir registrarse, debido a la carencia de una cuenta a la que asociar sus acciones es el rol con un menor número de funcionalidades a su disposición.

- *Usuario registrado.* Se trata del usuario que ha decidido dar el paso de crearse una cuenta en nuestro sistema, gracias a haber tomado esta decisión ahora disfruta de todas las acciones que un profesor o alumno pueda requerir para desempeñar su labor.
- *Usuario administrador.* Se trata del usuario con mayores privilegios del sistema, pudiendo afectar con sus decisiones a otras entidades externas a él dentro de la aplicación como usuarios, cursos y categorías.

Entre estos roles se pueden apreciar una relación de “piedra-papel-tijeras”, el usuario no registrado se convierte ser usuario registrado en el momento en que cree su cuenta obteniendo así un mayor poder de acción, la libertad de acción del usuario registrado dentro del sistema está limitada por los permisos del usuario administrador, el usuario administrador no posee poder alguno sobre el usuario no registrado dado que no tiene una cuenta que le identifique.

Debemos prestar especial atención al rol de Usuario registrado pues debe permitir que operen dos tipos de actores dentro de él:

- *Profesor.* Se trata del actor que busca compartir su conocimiento con los demás usuarios, este impulso le lleva a la decisión de crear cursos a través de los cuales expresar aquél saber que considere digno de ser infundido.
- *Alumno.* Se trata del actor que busca ampliar sus conocimientos y saciar su curiosidad a través de los recursos impartidos por el anterior actor.

Entre estos dos actores se da una relación de retroalimentación pues sin uno no tendría sentido el otro, es decir, el profesor necesita un público que escuche sus enseñanzas, mientras que el alumno requiere de un maestro que le conduzca en su aprendizaje. El corazón del sistema reside en esta simbiosis entre alumno y profesor.

3.3. Lista de tareas

En este apartado se detallan todas las funciones que nuestra plataforma permite realizar a sus usuarios, incidiendo además en el acceso a estas en base a su rol. Es importante el orden en el que se tratan los tipos de usuario, pues están dispuestos de tal modo que cada rol puede acceder a sus funciones propias y a todas las de los que estén descritos antes de ellos.

- *Usuario no registrado.* Tiene acceso al listado completo de cursos del sistema, permitiéndole buscarlos mediante distintos métodos y acceder a todo su contenido, sin embargo no podrá inscribirse y por consiguiente nunca podrá licenciarse en ellos, además podrá acceder al sistema de foros que los acompaña, pero solo como lector sin poder interferir con ellos. Esta decisión se toma con el objetivo de incentivarle a usar la única otra funcionalidad a su disposición: registrarse en el sistema, pero desde ese momento abandonará su rol.
- *Usuario registrado.* Tiene control sobre toda la información referente a su cuenta, pero lo verdaderamente importante es que, por un lado tiene a su disposición todas las herramientas necesarias para crear cursos y organizarlos según le convenga, y por otro lado puede inscribirse en cursos y acceder a todo el contenido de los mismos, además de valorarlos y tener un buen seguimiento todos aquellos en los que se haya inscrito.
- *Usuario administrador.* Tiene permisos para prohibir el acceso al sistema a todos los usuarios cuyo comportamiento no cumpla los estándares cívicos y de sentido común, y bloquear el acceso a cursos que sean irrelevantes o dañinos.

3.4. Descripción de requisitos

El sistema se fundamenta en una serie de requisitos cuyo cumplimiento nos lleva a la resolución, en última instancia, del problema que enfrentamos.

Concretamente el listado se encuentra dividido entre requisitos funcionales, aquellos que describen qué debe hacer un sistema en términos de las funcionalidades que ofrece a sus usuarios, y requisitos no funcionales, aquellos que imponen restricciones a cerca de cómo el sistema debe llevar a cabo su trabajo.

3.4.1. Requisitos funcionales

- *RF1.* El sistema permite la navegar entre pantallas.
- *RF2.* El sistema implementará paginación a la hora de mostrar su información, para mayor claridad y organización.
- *RF3.* El sistema mostrará un mensaje distinto en caso de que no haya información que mostrar.

- *RF4.* El sistema permite ver el listado de todos los cursos que existen en el sistema.
- *RF5.* El sistema permite ver el listado de todos los cursos por categorías.
- *RF6.* El sistema permite buscar cursos en base al título de los mismos.
- *RF7.* El sistema permite ver todo el contenido de un curso.
- *RF8.* El sistema permite ver el listado de cursos creados por un usuario.
- *RF9.* El sistema permite ver los datos de perfil de los usuarios.
- *RF10.* El sistema permite a sus usuarios almacenar y manipular datos de la base de datos.
- *RF11.* El sistema permite la creación de cuentas de usuario mediante el registro.
- *RF12.* El sistema permite a sus usuarios iniciar sesión.
- *RF13.* El sistema permite a sus usuarios editar su información de perfil.
- *RF14.* El sistema permite a sus usuarios cambiar su contraseña.
- *RF15.* El sistema permite a sus usuarios borrar su cuenta.
- *RF16.* El sistema permite a sus usuarios crear cursos mediante una serie de datos iniciales.
- *RF17.* El sistema permite a sus usuarios ver el listado de cursos propios.
- *RF18.* El sistema permite a su autor borrar sus cursos.
- *RF19.* El sistema permite a su autor editar los datos iniciales de un curso.
- *RF20.* El sistema permite a su autor crear temas para sus cursos.
- *RF21.* El sistema permite a su autor editar los temas de sus cursos.
- *RF22.* El sistema permite a su autor borrar los temas de sus cursos.
- *RF23.* El sistema permite a su autor ver el listado de temas de sus cursos.
- *RF24.* El sistema permite a su autor crear lecciones dentro de los tema de sus curso.
- *RF25.* El sistema permite a su autor editar las lecciones de los temas de sus cursos.
- *RF26.* El sistema permite a su autor borrar las lecciones de los temas de sus cursos.

- *RF27.* El sistema permite a su autor ver el listado de lecciones de los temas de sus cursos.
- *RF28.* El sistema permite a su autor crear requisitos para sus cursos.
- *RF29.* El sistema permite a su autor editar los requisitos de sus cursos.
- *RF30.* El sistema permite a su autor borrar los requisitos de sus cursos.
- *RF31.* El sistema permite a su autor ver el listado de requisitos de sus cursos.
- *RF32.* El sistema permite a su autor crear objetivos para sus cursos.
- *RF33.* El sistema permite a su autor editar los objetivos de sus cursos.
- *RF34.* El sistema permite a su autor borrar los objetivos de sus cursos.
- *RF35.* El sistema permite a su autor ver el listado de objetivos de sus cursos.
- *RF36.* El sistema permite a sus usuarios matricularse en cursos que no sean propios.
- *RF37.* El sistema permite a sus usuarios desmatricularse de cursos en los que estaban matriculados.
- *RF38.* El sistema permite a sus usuarios ver el listado de cursos en los que están matriculados.
- *RF39.* El sistema permite a sus usuarios añadir a su lista de cursos deseados aquellos que no sean propios.
- *RF40.* El sistema permite a sus usuarios quitar de su lista de cursos deseados aquellos que había añadido.
- *RF41.* El sistema permite a sus usuarios ver su lista de cursos deseados.
- *RF42.* El sistema permite al usuario ver el listado de cursos que ha completado.
- *RF43.* El sistema permite a sus usuarios dar me gusta a un curso.
- *RF44.* El sistema permite a sus usuarios quitar el me gusta a un curso al que había dado me gusta.
- *RF45.* El sistema permite a sus usuarios dar no me gusta a un curso.
- *RF46.* El sistema permite a sus usuarios quitar el no me gusta a un curso al que había dado no me gusta.

- *RF47.* Si un usuario le da me gusta a un curso al que había dado no me gusta se quitará este último.
- *RF48.* Si un usuario le da no me gusta a un curso al que había dado me gusta se quitará este último.
- *RF49.* El sistema debe mostrar para cada curso la puntuación que tiene mediante un sistema de 5 estrellas basándose en su proporción de me gusta frente a no me gusta.
- *RF50.* El sistema permite al administrador ver el listado de usuarios del sistema.
- *RF51.* El sistema permite al administrador buscar usuarios en base a su nombre.
- *RF52.* El sistema permite al administrador banear usuarios prohibiéndoles el acceso al mismo.
- *RF53.* El sistema permite al administrador desbanear usuarios concediéndoles de nuevo acceso al mismo.
- *RF54.* El sistema permite al administrador eliminar usuarios del sistema.
- *RF55.* El sistema permite al administrador ver el listado de cursos del mismo.
- *RF56.* El sistema permite al administrador buscar cursos en base a su nombre.
- *RF57.* El sistema permite al administrador banear cursos impidiendo el acceso a ellos.
- *RF58.* El sistema permite al administrador desbanear cursos volviéndolos accesibles de nuevo.
- *RF59.* El sistema permite al administrador eliminar cursos del sistema.
- *RF60.* El sistema permite al administrador ver el listado de las categorías existentes.
- *RF61.* El sistema permite al administrador buscar categorías en base a su nombre.
- *RF62.* El sistema permite al administrador crear nuevas categorías para clasificar los cursos.
- *RF63.* El sistema permite al administrador editar la información de una categoría.
- *RF64.* El sistema permite al administrador eliminar categorías.

3.4.2. Requisitos no funcionales

- *RNF1.* La base de datos utilizada será MySQL.
- *RNF2.* El backend se implementará en Java bajo el framework Spring Boot.
- *RNF3.* La interfaz seguirá los principios de usabilidad y accesibilidad aprendidos en la asignatura de IPO.
- *RNF4.* El sistema debe seguir un diseño responsive capaz de adaptarse a los distintos tamaños de pantalla.
- *RNF5.* El sistema contará con una página de ayuda que responda las dudas que puedan surgir acerca del sistema y su uso.
- *RNF6.* El sistema contará con una página en la que se expliquen los términos de servicio de la aplicación.
- *RNF7.* El sistema contará con una página en la que se explique la política de privacidad de la aplicación.
- *RNF8.* El sistema debe tener una página de error propia para cuando falle alguna operación de la aplicación.
- *RNF9.* El sistema debe responder de manera apropiada a las acciones del usuario.
- *RNF10.* El sistema no permite crear más de una cuenta de usuario que comparten el mismo correo electrónico.
- *RNF11.* El inicio de sesión debe realizarse mediante el correo electrónico y contraseña del usuario.
- *RNF12.* No se limitará el número de cursos que un usuario puede crear.
- *RNF13.* Los cursos creados no caducan, solo desaparecen si son borrados.
- *RNF14.* No se limitará el número de cursos en los que un usuario puede matricularse.
- *RNF15.* No se limitará el número de cursos que un usuario puede añadir a la lista de deseados.
- *RNF16.* Cuando el administrador borre una categoría todos los cursos de dicha categoría irán a la categoría de “Otros”.
- *RNF17.* El administrador no podrá borrar la categoría de “Otros”.

3.5. Casos de uso

Los casos de uso son un artefacto que define una secuencia de acciones que da lugar a un resultado observable, los utilizaremos para representar la manera en la que un usuario interactúa con el sistema persiguiendo un objetivo determinado.

3.5.1. Descripción de casos de uso

Los casos de uso representan de manera gráfica los requisitos funcionales, pues estos definen las funcionalidades que ofrece el sistema y que el usuario empleará para alcanzar sus objetivos.

Dada la restricción de extensión de la memoria que establece la normativa es imposible abarcar en esta sección todos los casos de uso del sistema, por ello los casos de uso se han ubicado en el Anexo 2: Casos de uso. Es necesario advertir aquí de su ubicación pues en el siguiente punto nos referiremos a todos ellos a la hora de construir el diagrama de casos de uso.

3.5.2. Diagrama de casos de uso

Los casos de uso siempre persiguen algún objetivo concreto, sin embargo en muchas ocasiones debemos conseguir pequeños objetivos para alcanzar uno mayor, ahí es donde entran en juego las relaciones que ligan casos de uso que forman parte del mismo flujo de eventos. Existen dos relaciones fundamentales:

- *Inclusión.* Se representa mediante «include», expresa que un caso de uso necesita la información y servicios de otro para llevar a cabo su función de manera imprescindible. A incluye a B si cada vez que queremos completar A tenemos que completar B.
- *Extensión.* Se representa mediante «extend», expresa que un caso de uso puede utilizar la información y servicios de otro si así lo requiere. A extiende de B si para completar A hemos completado previamente B.

La Figura 3.1 expone un diagrama que muestra los posibles flujos que pueden darse en el sistema a partir de los casos de uso vistos en la sección anterior unidos mediante las relaciones vistas:

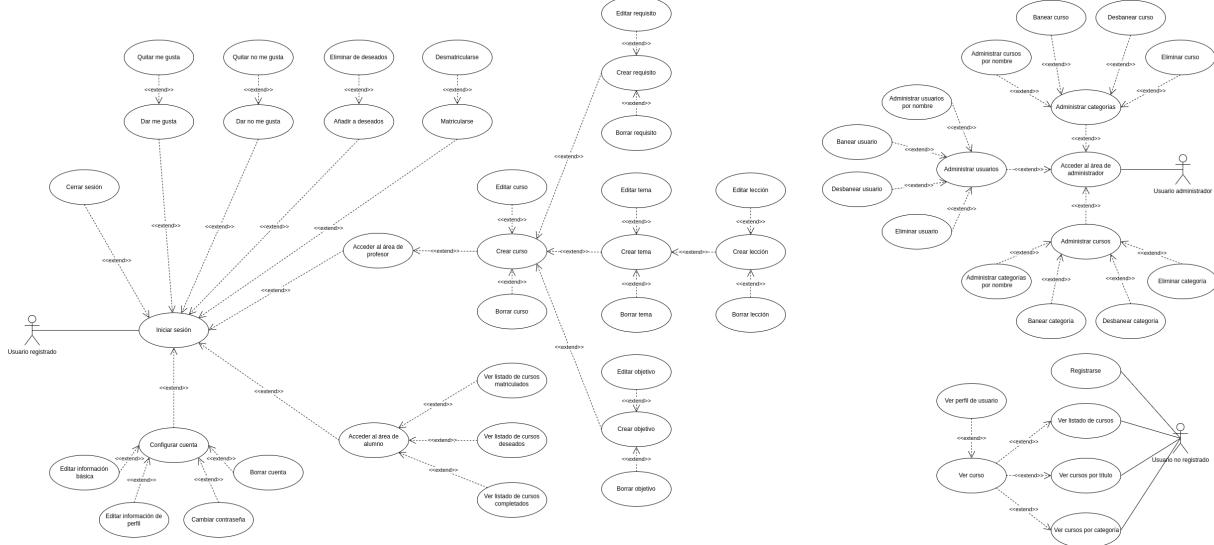


Figura 3.1: Diagrama de casos de uso

3.6. Arquitectura de alto nivel

La aplicación se ha construido siguiendo el patrón de arquitectura Modelo-Vista-Controlador, MVC por sus siglas, cuya principal idea es separar el código en distintas capas acotadas por las responsabilidades de las que se encargarán. Concretamente encontramos tres capas como su nombre indica: modelo, vista y controlador.

- *Modelo.* Es la capa que se encarga de los datos de la aplicación, estos se encuentran normalmente en bases de datos, por lo que esta capa será la encargada de proporcionar métodos para crear, modificar, consultar y eliminar los datos guardados. El modelo debe contener los datos que serán enviados a la capa de la vista para ser mostrados al usuario.
- *Vista.* Es la capa que contiene la interfaz del sistema y por tanto el código de las pantallas que conforman la aplicación. Además se encarga de renderizar los datos enviados en el modelo para mostrar correctamente los distintos tipos de datos e información que este contiene.
- *Controlador.* Es la capa encargada de responder a las peticiones que el usuario de la aplicación solicitará mediante su interacción con la vista. Esta capa es la que dota de funcionalidad al sistema, para ello sirve de puente entre la vista y el modelo, recibiendo solicitudes de la vista que pide al modelo e implementando entre medias la lógica y algoritmos que sustentan la aplicación.

Finalmente podemos apreciar de un modo más simple el funcionamiento de esta arquitectura en la Figura 3.2:

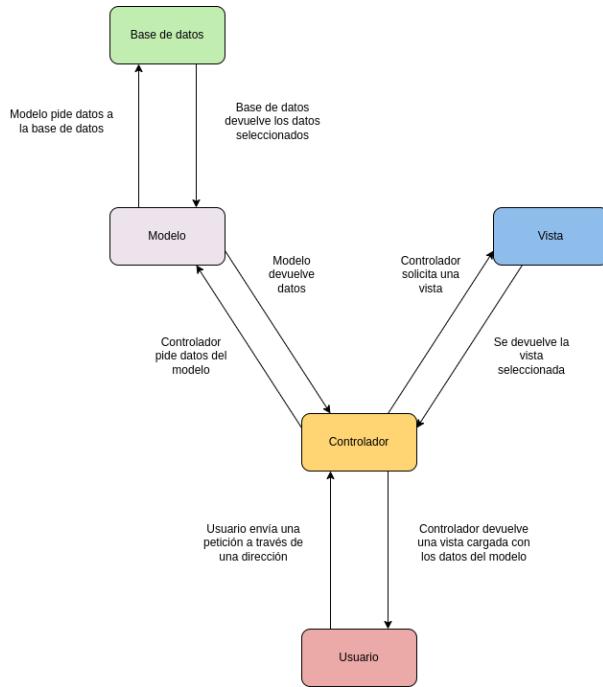


Figura 3.2: Diagrama de arquitectura de alto nivel

3.7. Diseño de la base de datos

Las bases de datos son los almacenes en los que guardamos los datos para permitir su persistencia, principalmente existen dos tipos de bases de datos:

- *Bases de datos relacionales*. Cumplen con el modelo relacional y para ello almacenan, representan y permiten acceder a datos relacionados entre sí.
- *Bases de datos no relacionales*. Los datos que almacenan no tienen identificadores y por tanto no permiten enlazar unos datos con otros en base a estos. Son más adecuadas para almacenar información organizada mediante documentos, objetos de programación orientada a objetos o casos en los que no se tiene un esquema exacto de lo que se va a almacenar

Para la persistencia de los datos se decidió utilizar una base de datos de tipo relacional, debido a la importancia y abundancia de las relaciones que se dan entre las entidades que existen en el sistema como veremos más adelante. La necesidad de almacenar todos estos enlaces entre los datos fue el principal factor que inclinó la decisión, pero además las bases de datos relacionales poseen otras ventajas como:

- Mantienen la uniformidad de los datos en todas las instancias de la base de datos y las aplicaciones que la usen, lo cual puede favorecer la difusión de los cursos del sistema en caso de que otras aplicaciones decidan acceder a ellos, pues siempre estarán accediendo a la última versión de los datos.
- Debido a que las entidades tienen identificadores pueden asegurar que no se produzca duplicidad de registros y favorece la normalización de los datos.
- Evitan conflictos cuando varios usuarios o aplicaciones intentan acceder a los datos al mismo tiempo y pueden bloquear el acceso mientras los datos se están actualizando.
- Implementan concurrencia que gestiona consultas de varios usuarios o aplicaciones al mismo tiempo a la base de datos.

Una vez detallados los motivos de la selección del tipo de base de datos elegido, se pasará a explicar qué son las entidades y las relaciones en el ámbito de las bases de datos, para finalmente hablar de cuales son las que componen el sistema.

Las entidades son las unidades de la base de datos que contienen información para representar objetos del mundo real, para ello estructuran su información en base a una serie de atributos. Las instancias de una entidad se guardan en tablas dentro de la base de datos, de modo que las columnas son los atributos y a cada fila le corresponde una instancia.

Las relaciones son el mecanismo de unión entre distintas entidades o tablas, para ello al menos una entidad de la relación debe tener un atributo (clave foránea) que apunte al identificador (clave primaria) de la otra. Podemos encontrar relaciones de una entidad con otra, una entidad con muchas otras entidades y muchas entidades con otras muchas entidades.

En el proyecto encontramos las siguientes tablas:

- *Course*. Entidad que representa un curso en el sistema.
- *Category*. Entidad que representa las posibles categorías con las que se pueden clasificar los cursos.
- *Requirement*. Entidad que represente un requisito que el alumnado de un curso debería cumplir o al menos tener en cuenta a la hora de matricularse e intentar superar el mismo.

- *Course-requirements*. Esta tabla relaciona un requisito con un curso, cada requisito debe estar asociado con un único curso, por su parte múltiples requisitos pueden estar asociados al mismo curso.
- *Objective*. Entidad que representa uno de los objetivos que el curso intenta que sus alumnos alcanzan a través de la superación del mismo.
- *Course-objectives*. Tabla que relaciona un objetivo con un curso, cada objetivo debe estar asociado con un único curso, por su parte múltiples requisitos pueden estar asociados al mismo curso.
- *Lesson*. Entidad que representa una lección, las lecciones son el contenido de los cursos y donde realmente se encuentra el material educativo que el profesor intenta impartir a sus alumnos.
- *Theme-lessons*. Tabla que relaciona una lección con un tema, cada lección debe estar asociado con un único tema, por su parte múltiples lecciones pueden estar asociados al mismo tema.
- *Theme*. Entidad que representa un tema que compone el temario del curso y por tanto el grueso de su contenido.
- *Course-themes*. Tabla que relaciona un tema con un curso, cada tema debe estar asociado con un único curso, por su parte múltiples requisitos pueden estar asociados al mismo curso.
- *User*. Entidad que representa la información asociada a un usuario de la aplicación.
- *User-user-courses*. Tabla que relaciona a un usuario con los cursos que el mismo ha creado, por tanto se trata de una relación en la que un único usuario puede estar asociado a multitud de cursos distintos.
- *User-enrolled-courses*. Tabla que relaciona a un usuario con los cursos que se ha matriculado, por tanto se trata de una relación en la que un único usuario puede estar asociado a multitud de cursos distintos.
- *User-completed-courses*. Tabla que relaciona a un usuario con los cursos que ha superado, por tanto se trata de una relación en la que un único usuario puede estar asociado a multitud de cursos distintos.

- *User-wished-courses*. Tabla que relaciona a un usuario con los cursos que ha añadido a su lista de deseados, por tanto se trata de una relación en la que un único usuario puede estar asociado a multitud de cursos distintos.
- *User-liked-courses*. Tabla que relaciona a un usuario con los cursos a los que ha dado me gusta, por tanto se trata de una relación en la que un único usuario puede estar asociado a multitud de cursos distintos.
- *User-disliked-courses*. Tabla que relaciona a un usuario con los cursos que a los que ha dado no me gusta, por tanto se trata de una relación en la que un único usuario puede estar asociado a multitud de cursos distintos.

Por último se puede encontrar en la Figura 3.3 el modelo entidad relación que representa todo lo que hemos desarrollado previamente:

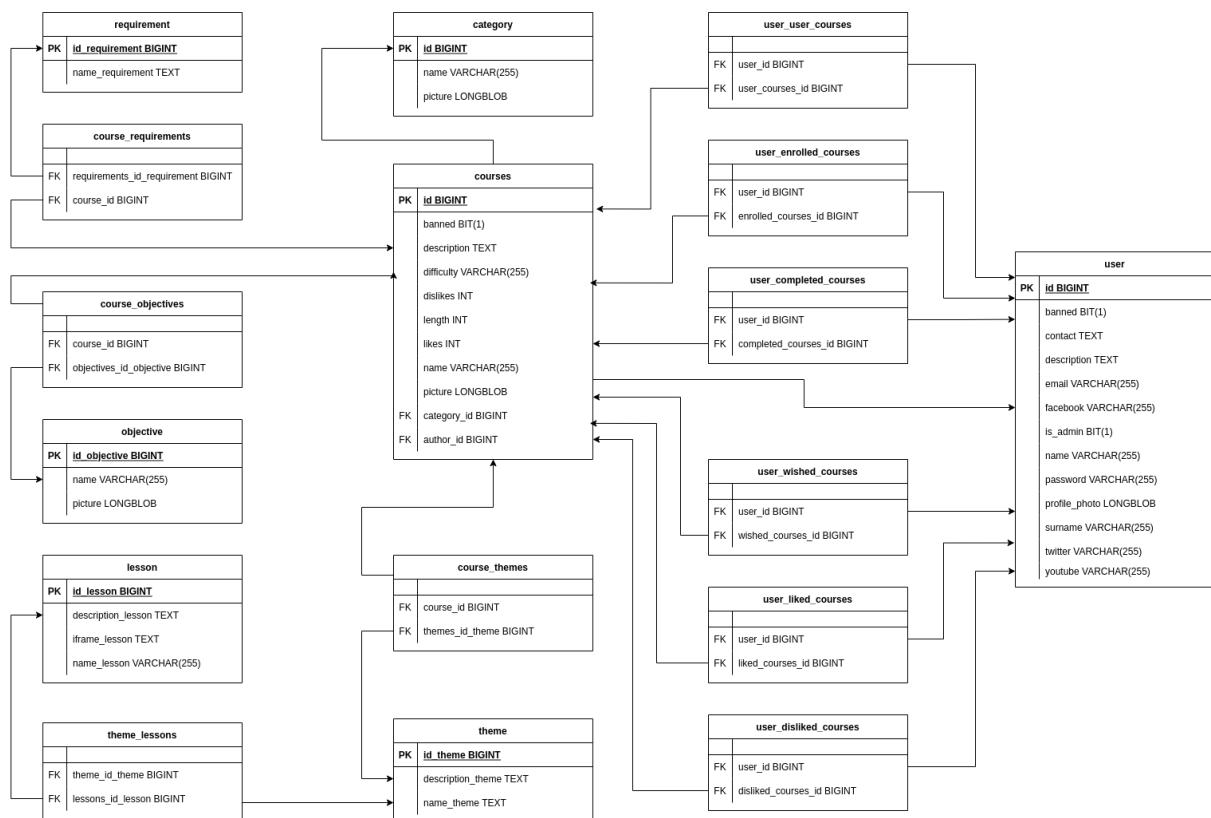


Figura 3.3: Diagrama de la base de datos

3.8. Diseño de la interfaz y experiencia de usuario

El diseño de la interfaz (UI) y experiencia (UX) del usuario ha seguido los principios de usabilidad y accesibilidad aprendidos en la asignatura de IPO [12].

Pese a la importancia de la aplicación de estos principios, se ha decidido no seguir un enfoque DCU en este primer TFG, actualmente el sistema todavía no presenta las características que lo diferenciarán del resto de sus competidores, por ahora nos apoyaremos en el estudio que han realizado estos para el diseño de sus sistemas y tomaremos referencias de sus diseños para los puntos comunes. Será en el siguiente TFG cuando, a la hora de diseñar las particularidades de nuestra apuesta, nos basemos en un estudio propio realizado bajo la metodología de DCU.

Debido a nuestra apuesta por la apertura de la educación, el desarrollo técnico de la interfaz ha seguido un diseño responsive para que tenga un correcto visionado independientemente de las características del dispositivo en el que se muestre y por tanto no existan limitaciones técnicas que dificulten el acceso al sistema. El sistema es totalmente funcional y accesible desde un ordenador, tablet o dispositivo móvil.

En el Anexo 3 se encuentra el listado de pantallas del sistema ordenadas alfabéticamente y acompañadas por una pequeña descripción, a partir de ellas se ha realizado a modo de guía el diagrama que se observa en la Figura 3.4.

3.9. Implementación

Para hablar sobre la implementación del proyecto me gustaría dividir el estudio en dos partes: frontend y backend, respecto a lo visto acerca de la arquitectura de la aplicación, las capas de modelo y controlador caerían del lado del backend mientras que la vista se encuentra del lado del frontend.

- *Frontend.* Es la parte de la aplicación con la que interactúa el usuario, es decir, las páginas o vistas que conforman el sistema. En el anterior punto se habló del frontend desde una perspectiva de diseño, mientras que en este apartado trataré de abordarlo desde una perspectiva técnica.

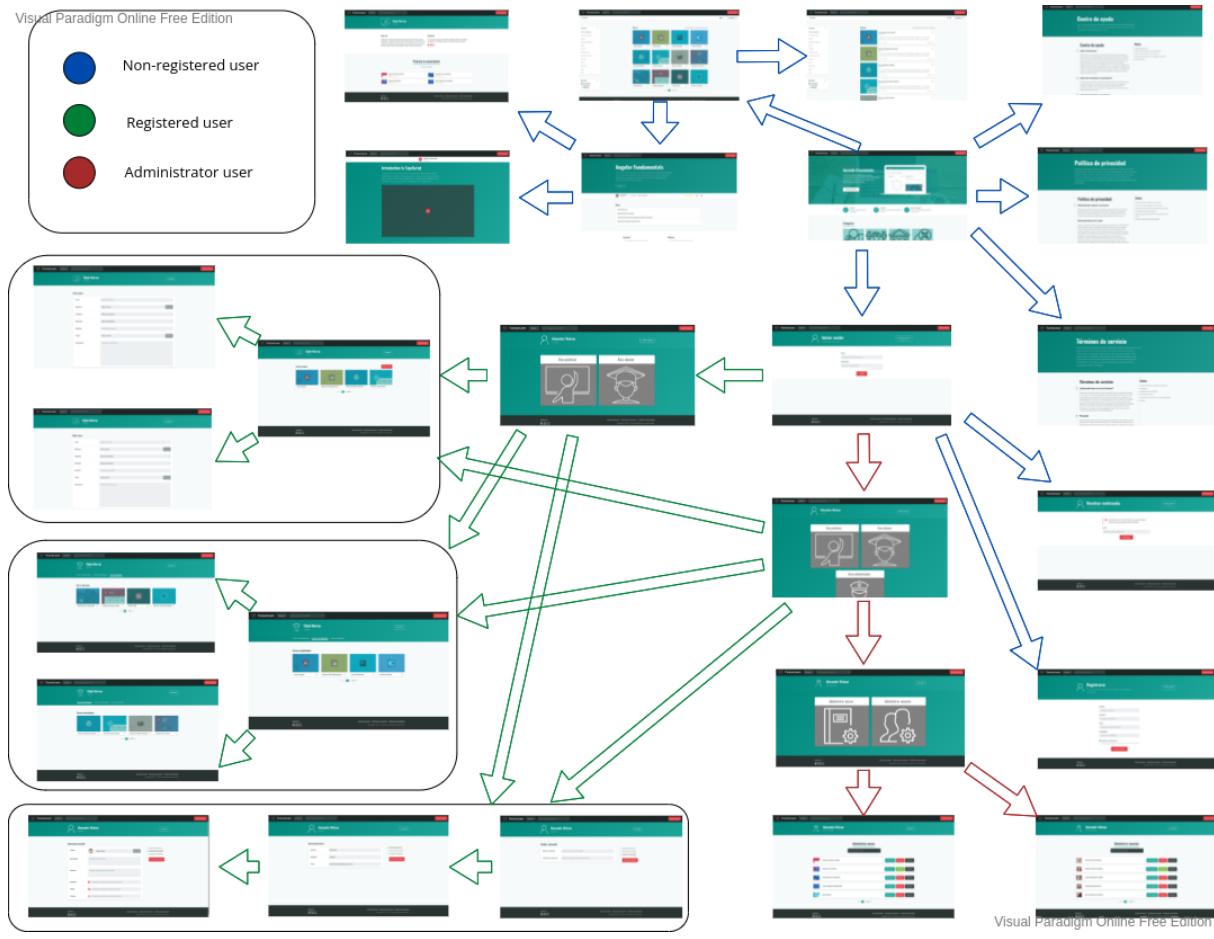


Figura 3.4: Diagrama de navegación entre pantallas

La interfaz del sistema se basa en la plantilla hecha por FrontendMatter: <http://tutorio-bootstrap.frontendmatter.com/index.html>, construida mediante las tecnologías básicas de frontend: HTML para el maquetado, CSS para el diseño y JavaScript para el comportamiento dinámico. Se valoró la posibilidad de pasar la interfaz a algún framework moderno como Angular, pero se descartó la idea debido a que aumentaría los tiempos de desarrollo y no aportaría nada que no se pudiese hacer sin él.

- *Backend.* Es la parte de la aplicación encargada de ejecutar toda su lógica, implementar toda su funcionalidad, responder a las peticiones del usuario generadas a través de su interacción con el frontend y proporcionar una capa de operación sobre la información de la base de datos.

Antes de desarrollar la estructura del backend, conviene aclarar dos términos que resultan confusos y pueden servir para aclarar dicha estructura, hablamos de los dos

tipos de lógica presentes a la hora de implementar la funcionalidad de un sistema informático:

- *Lógica de negocio.* Define las operaciones que se pueden realizar para que el sistema cumpla con las condiciones acordadas cuando se diseñó la aplicación, se llama así porque teóricamente le corresponde a expertos empresariales diseñar estas condiciones. En resumen, la lógica de negocio define que puede y debe hacer la aplicación.
- *Lógica de aplicación.* Es la lógica que no tiene que ver con los negocios o decisiones empresariales, pero que debe implementarse para que todo funcione como es debido. Idealmente la lógica de aplicación correspondería al código encargado de recibir peticiones, seleccionar qué lógica de negocio aplicar y finalmente devolver una vista. Como podemos ver es imprescindible para que el sistema se mantenga en pie, pero determina cómo funcionan las operaciones a las que llama.

El backend del sistema sigue un diseño por capas: capa del controlador, capa de servicio, capa de repositorio y capa de modelo, cada una de ellas se encarga de una responsabilidad, pero en términos generales podríamos decir que la lógica de la aplicación está integrada en la capa del controlador, mientras que la lógica de negocio se encuentra presente en las capas de servicio, modelo y repositorio.

- *Capa del controlador.* Esta capa se encarga de atender a las peticiones de los usuarios, recibiendo los datos que sean oportunos, llamando a los métodos de la capa de servicio que realicen la funcionalidad que el usuario está solicitando y finalmente devolviéndole una vista apropiada a su demanda.
- *Capa de servicio.* Esta capa implementa las funcionalidades de la aplicación, cualquier acción que el usuario quiera realizar implicará una llamada de la capa de controlador a alguno de los métodos aquí presentes.
- *Capa de repositorio.* El sistema utiliza un modelo de programación llamado mapeo objeto-relacional (ORM), que consiste en la transformación de las tablas de la base de datos en entidades que abstraen al desarrollador del lenguaje de la base de datos utilizada y le permiten realizar operaciones CRUD (crear, leer, modificar y eliminar) con mayor facilidad. En esta capa se definen todas las posibles operaciones que CRUD que deseemos realizar sobre los datos del modelo en la base de datos.

- *Capa de modelo.* Esta capa define las distintas entidades que guardaremos en la base de datos y las relaciones entre estas, serán por tanto los objetos con los que operemos en el sistema. Realmente todas las capas deben tener acceso a esta: el controlador necesita saber que objetos recibirá e integrará en la vista que devolverá en las peticiones del usuario, el servicio debe conocer al modelo para poder operar sobre él y realizar las funcionalidades de la aplicación y el repositorio necesita conocer al modelo para saber como se estructura la información de la base de datos a la que intenta acceder.

Existen muchos posibles casos de uso, pero si tuviéramos que definir de manera generalizada cómo se relacionan estas capas podríamos decir que a la capa del controlador le llegará una petición de un usuario que desea realizar una acción, esta primera capa le pedirá a la capa de servicio que realice dicha acción, para ello tal vez tenga que pedir al repositorio que acceda a la base de datos, tras esto el repositorio devuelve su información al servicio, este opera con él y finalmente se lo retorna al controlador para que le envíe al usuario los datos integrados en una vista.

La Figura 3.5 expone el reflejo de dicho comportamiento y la interrelación entre las distintas capas del sistema:

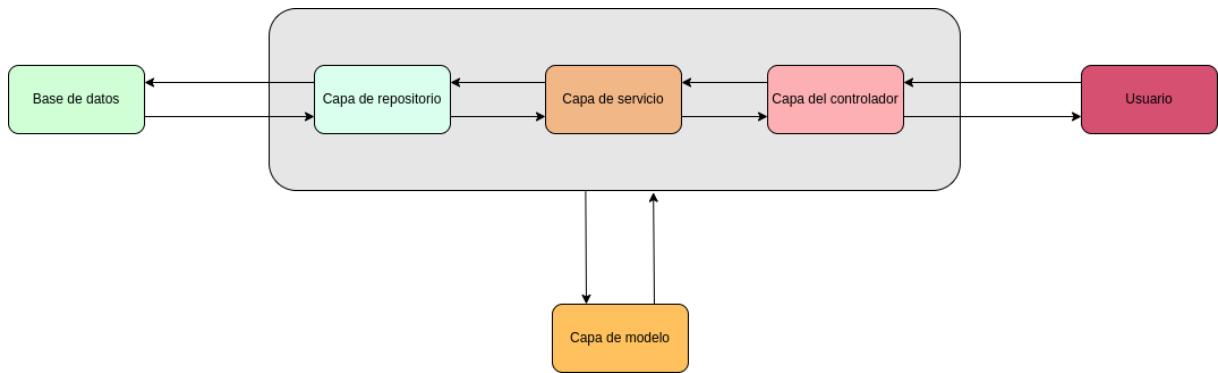


Figura 3.5: Diagrama de arquitectura del backend

Por último la Figura 3.6 muestra una evolución del anterior diagrama, ahora puede observarse como las clases que conforman el código siguen el diseño por capas:

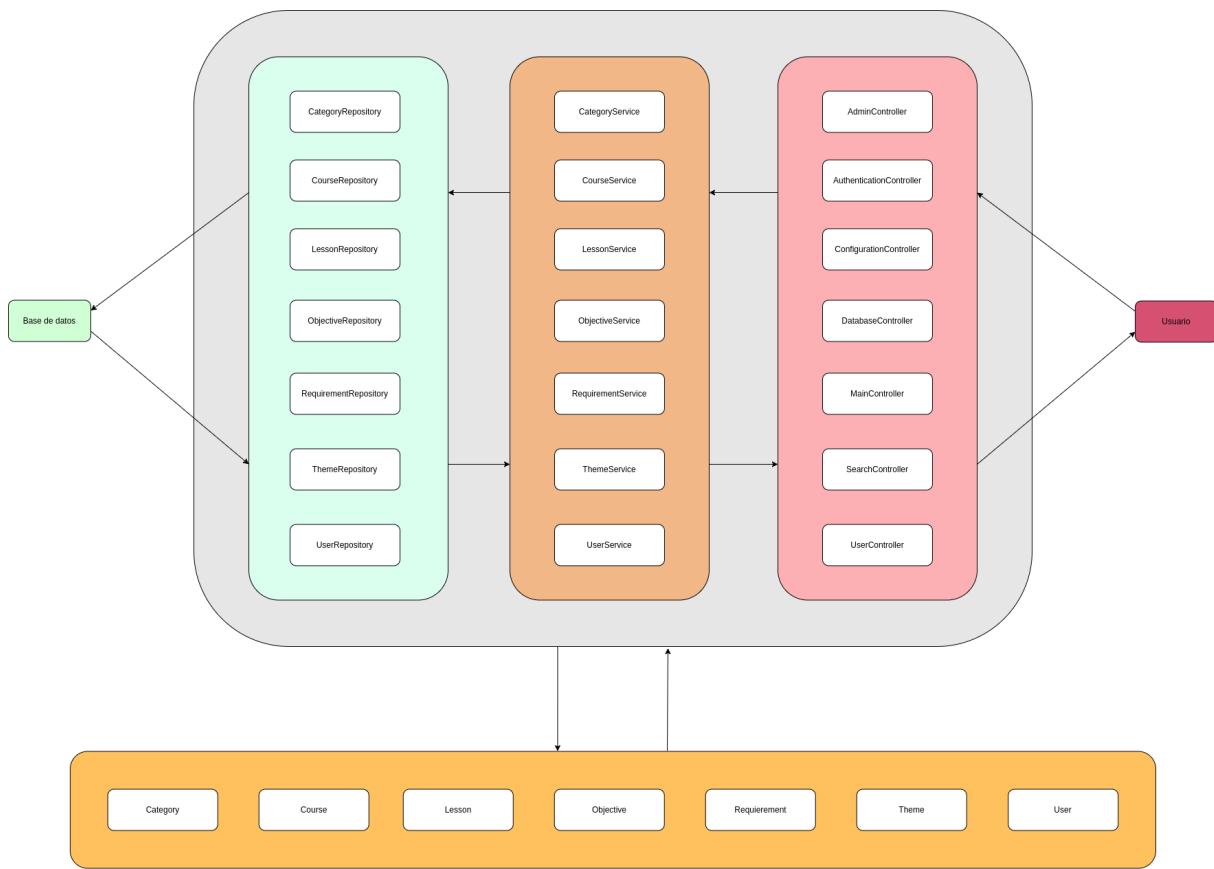


Figura 3.6: Diagrama de arquitectura del backend con clases

Capítulo 4

Experimentos y validación

La intención inicial era la de entregar un trabajo verdaderamente sólido y pulido, que fuese robusto frente los posibles errores que los usuarios puedan encontrar y provocar. En esta sección veremos cuales han sido los esfuerzos dedicados en materia de testing para asegurar la calidad de la plataforma.

4.1. Pruebas unitarias

Las pruebas unitarias son uno de los tipos de pruebas más extendidos en el testing, sirven para comprobar el correcto funcionamiento de fragmentos de código, normalmente se refiere a métodos de la aplicación. Se llaman así debido a que prueban código aislado, es decir, exclusivamente esa unidad de código.

Para ejecutar este tipo de pruebas en proyectos de Spring Boot se suelen utilizar mayoritariamente dos tecnologías: JUnit y Mockito.

En el punto 3.9 se estableció como la estructura del backend se fundamentaba en una división en tres capas: capa del controlador, capa de servicio y capa de repositorio, que se comunicaban según ese orden. Cada capa tiene distintas responsabilidades y por tanto lo que buscamos demostrar mediante las pruebas unitarias será distinto para cada una.

La primera capa que se debe tratar es la capa base: la capa de repositorio, esta se encarga de operar con la base de datos y para ello nos ofrece una serie de métodos por defecto como pueden ser eliminar o guardar entidades.

Sin embargo a la hora de realizar operaciones más complejas como búsquedas en base a combinaciones de parámetros específicos de la aplicación, será responsabilidad del programado definir dichas operaciones.

No es necesario que realicemos pruebas unitarias a las operaciones por defecto,

pues estas ya han sido probadas por el fabricante de la tecnología. El testing en esta capa se reduce por tanto a las operaciones definidas por el desarrollador, dado que es responsabilidad suya que funcionen correctamente.

Probar el correcto funcionamiento de estas operaciones requiere de una base de datos con la que puedan operar, el problema está en que no queremos que las pruebas afecten al sistema real y que la base de datos de la aplicación almacene información proveniente del testing.

Para solucionar este problema se decide utilizar una base de datos independiente, que tiene como margen de operación la ejecución de los testa. Con el objetivo de no aumentar el número de dependencias ni añadir complejidad innecesaria al proyecto, se optó por utilizar una base de datos H2, debido a que se trata de una base de datos en memoria que no requiere instalar software adicional.

De este modo se pueden probar las operaciones que se han definido en la capa de repositorio interactuando con la base de datos H2 y demostrar que cumplen con el funcionamiento deseado.

La siguiente capa a tratar es la capa de servicio, esta capa implementa la lógica de negocio del sistema y se apoya sobre la capa vista anteriormente para operar con la base de datos. Esta dependencia de la capa de repositorio choca con la filosofía de las pruebas unitarias de probar métodos que sean independientes por sí mismos sin la necesidad de otros módulos.

La solución a este problema viene del framework Mockito, se utilizará con dos propósitos: definiremos que queremos que devuelvan los métodos de la capa de repositorio que se invocan y configuraremos capturadores de argumentos que atrapan los valores que se les pasa por parámetros a los métodos del repositorio.

Con todo esto se pretende comprobar que los filtros y las operaciones implementadas por el servicio se aplican correctamente y simular todos aquellos métodos de la capa de repositorio que sean necesarios para probar correctamente la capa de servicio.

Por último resta la capa del controlador, esta recibe direcciones url, decide que operaciones de la capa de servicios se ejecutan, trae información de esta con la que carga

el modelo y lo integra en una vista que devuelve al usuario. De nuevo la dependencia de la capa de servicio se soluciona mediante Mockito, dado que esta capa será simulada los datos con los que carguemos el modelo también estarán simulados.

Podemos considerar que el verdadero interés de testear esta capa de manera unitaria se encuentra en comprobar que para cada dirección disponible en el sistema, se devuelve la vista que le corresponde.

Finalmente cabría preguntarse que hay de la integración entre las capas.

Los tests unitarios de cada capa respetan esta filosofía gracias a que hemos simulado el comportamiento de las capas de las que dependen mediante Mockito, esta simulación se ha realizado exactamente acorde a lo que nos devolvería dicha capa y dado que se ha demostrado que cada una funciona correctamente, podemos afirmar que, si no se tratase de una simulación, el flujo de la información y la compatibilidad entre las capas se daría sin la aparición de ningún tipo de problemas.

Capítulo 5

Conclusiones

5.1. Conclusiones generales

En este trabajo de fin de grado se ha desarrollado un repositorio de cursos público y abierto sin restricciones de ningún tipo, que permite a sus usuarios ponerse en el lugar tanto del profesor, dándoles todo tipo de herramientas necesarias para que puedan modelar sus cursos como deseas, como del alumno, permitiéndole matricularse, valorar cursos y gestionar su lista de cursos deseados.

A pesar de que la verdadera innovación educativa será desarrollada en el siguiente trabajo, aquí hemos añadido pequeñas características que no son comunes en página de esta índole como la incorporación de contenido embebido en los cursos para permitir el acceso a recursos que de otro modo se encontrarían más dispersos, y otras de gran utilidad como la gestión de las categorías como contenido dinámico o el trabajo puesto en las herramientas del administrador que permiten una gestión sencilla de la plataforma.

Cabe destacar que, pese a estar implementado sobre una plantilla, el diseño de la interfaz del sistema ha partido desde cero, sin la necesidad de copiar ni apoyarse en el de ninguna de las alternativas actuales.

La fase de implementación de la interfaz ha sido la más extensa de todas las fases debido a que todas las decisiones de diseño, flujo entre pantallas, creación de elementos y organización de contenidos y recursos, son de elaboración propia.

No solo se realizó este trabajo para la versión de ordenador, debido a que ForeverLearn sigue un diseño responsive que se adapta a los distintos marcos de pantalla, tuvo que realizarse un esfuerzo extra en adaptar toda la interfaz a la pantalla de un dispositivo móvil, obligando a redefinir elementos y a implementar nuevas estructuras para hacer usable y accesible la plataforma desde una pantalla tan limitada.

En este punto es necesario dedicar una líneas ha considerar el grado de éxito en la persecución de los distintos objetivos que se planteaban en el punto 2.2.

El principal objetivo era el de promover la filosofía de aprendizaje abierto, para ello ForeverLearn permite que cualquier usuario que navegue por la página pueda acceder a todo el contenido de los cursos, sin necesidad de pagar ni registrarse.

Entre los objetivos planteados estaba que el usuario registrado pudiera ser tanto profesor como alumno, y que ambos roles tengan herramientas para llevar a cabo sus intereses.

Cuando un usuario se registra en ForeverLearn se le da acceso a la pantalla de usuario en la que se encuentran dos secciones: el área de profesor donde se ubican los cursos creados por el usuario y las herramientas de creación, edición y eliminación de cursos, y el área de alumno desde la que puede acceder al listado de cursos matriculados, cursos completados y cursos deseados.

Una de las metas era la de facilitar la búsqueda de cursos, ForeverLearn ofrece herramientas para que buscar cursos y navegar entre ellos resulte sencillo y que el usuario pueda encontrar lo que busca sin dificultades.

Además de los objetivos técnicos, se plantearon propósitos de mejora personal y profesional, que si bien puede resultar difícil que uno mismo valore algo así, considero que ForeverLearn ha planteado los suficientes problemas para obligarme a aprender mucho más de lo que sabía cuando comenzó el proyecto.

Pero sobre todo a nivel personal ha supuesto un reto mantener la disciplina necesaria para encarar un proyecto así de principio a final y realizar un esfuerzo constante para que este primer trabajo de fin de grado sea una realidad.

Por último, de manera personal, quiero añadir que me siento realmente satisfecho con el estado final de este primer proyecto, a pesar de necesitar las mejoras y los avances que serán desarrollados en el siguiente Trabajo de Fin de Grado para estar completo, se ha construido una buena base de la que poder partir.

5.2. Visión de futuro a corto plazo

A corto plazo se encuentra el desarrollo del siguiente TFG, en el que implementaremos la evaluación de los cursos desde un enfoque conectivista, es decir, se diseñará un sistema que ofrezca herramientas de interacción a los alumnos del curso con el objetivo de crear una red de difusión del conocimiento para cada curso.

Estableceremos como incentivar esta interacción mediante la gamificación, una de las técnicas educativas más recientes y de mayor auge. Creando recompensas para los esfuerzos de los alumnos que les permitan aumentar su alcance e influencia en la red del curso.

El siguiente paso será decidir cuales son los requisitos para considerar un curso como superado e implementar un módulo de creación de certificados dinámico para el usuario, el curso y los méritos que este realizó para superarlo, de modo que sea una prueba irrefutable del esfuerzo y los logros del alumno.

Además de lo referente a la innovación educativa, existen una serie de añadidos no se pudieron incluir en este primer trabajo por falta de tiempo y debido a que no son cruciales para la idea del proyecto. Por todo ello se decidió desplazarlos al segundo TFG en caso de que haya tiempo suficiente para implementarlos.

La primera de estas mejoras se ubica en la dirección de los Recursos Educativos Abiertos, ForeverLearn no solo pretendo que los cursos sean accesibles por sus navegantes, sino por cualquier otro sistema que deseé acceder a ellos. Para este propósito se utilizará uno de los mecanismos de comunicación entre sistemas más extendidos: se construirá una API REST que ofrezca la información de los cursos a través de distintos sistemas de filtrado.

Por otra parte se desea realizar un fuerte trabajo en reforzar toda la seguridad informática del sistema, concretamente el principal interés se encuentra en sustituir el protocolo de conexión HTTP por el protocolo HTTPS, una versión mejorada de HTTP que cifra la información intercambiada entre el navegador y el servidor.

Además se pretenden integrar algunas funcionalidades que ofrece Spring Security: la parte de Spring dedicada a la seguridad informática, como pueden ser la separación de

los roles de usuario y la limitación del acceso a URLs dependiendo del rol.

Otras cuestiones de seguridad que se postulan para ser implementadas son el cifrado de contraseñas en la base de datos y la protección CSRF (Cross Site Request Forgery) que impide que una página web realice peticiones a otra en la que el usuario ha iniciado sesión.

Cabe la posibilidad de que durante el transcurso del desarrollo del próximo TFG algunas de estas ideas se modifiquen, se descarten o incluso se añadan nuevas, pero actualmente estas son las intenciones de desarrollo para ForeverLearn a corto plazo.

5.3. Visión de futuro a largo plazo

ForeverLearn tiene potencial para ser algo más que un Trabajo de Fin de Grado, el sistema que estamos creando así como los conceptos de: cMOOC y Recursos Educativos Abiertos que integra pueden llegar a consolidar una alternativa seria a los sistemas de evaluación clásicos.

En los últimos años se ha realizado un gran trabajo en innovación educativa explorando las posibilidades que ofrece la informática y a largo plazo mi intención es que ForeverLearn pueda aportar su granito de arena en ese camino.

En el futuro me gustaría seguir trabajando en explorar nuevas formas de integración de las tecnologías de la información en la educación, sea a través de lo que propone ForeverLearn o desde otras perspectivas, trabajando por mi cuenta o desde la línea de investigación de un doctorado.

En definitiva, trabajar en este proyecto me ha hecho darme cuenta del potencial que tiene este campo, de lo que puede aportar a la sociedad y de la importancia de que la educación siga evolucionando siempre hacia delante.

Capítulo 6

Bibliografía

- [1] Conole G. Los MOOC como tecnologías disruptivas: estrategias para mejorar la experiencia de aprendizaje y la calidad de los MOOC. Red [Internet]. 21 de octubre de 2016 [citado 25 de febrero de 2022];(50). Disponible en: <https://revistas.um.es/red/article/view/270791>
- [2] Cabero AlmenaraJ, Llorente CejudoM del C, Vázquez MartínezAI. Las tipologías de MOOC: Su diseño e implicaciones educativas. profesorado [Internet]. 1 de abril de 2014 [citado 25 de febrero de 2022];18(1):13-6. Disponible en: <https://revistaseug.ugr.es/index.php/profesorado/article/view/19116>
- [3] AlDahdouh A, Osorio A, Caires S. Understanding knowledge network, learning and Connectivism [Internet]. 2015 [citado el 25 de febrero de 2022]. Disponible en: <https://papers.ssrn.com/abstract=3063495>
- [4] Mackness J, Waite M, Roberts G, Lovegrove E. Learning in a small, task-oriented, connectivist MOOC: Pedagogical issues and implications for higher education. Int Rev Res Open Distrib Learn [Internet]. 2013 [citado el 25 de febrero de 2022];14(4):140–59. Disponible en: <https://www.erudit.org/en/journals/irrodl/1900-v1-n1-irrodl05094/1066904ar/abstract/>
- [5] Yeager C, Hurley-Dasgupta B, Bliss CA. CMOOCs and global learning: An authentic alternative. Journal of Asynchronous Learning Networks [Internet]. 2013 [citado el 25 de febrero de 2022];17(2):133–47. Disponible en: <https://eric.ed.gov/?id=EJ1018269>
- [6] Combéfis S, Roy V, Eriksson T;., Adawi T, Stöhr C, Lackner E, et al. MOOCs in Scandinavia 2016-extended abstracts balyasin, Mikhail IT'S TIME TO MOVE ON: ANALYSIS OF ARGUMENTS AROUND MOOCS [Internet]. Chalmers.se. [citado el 25 de febrero de 2022]. Disponible en:

https://www.chalmers.se/en/conference/moocs2016/Documents/extended_abstracts.pdf

[7] Burgos Aguilar JV, Burgos V, Título S. DISTRIBUCIÓN DE CONOCIMIENTO Y ACCESO LIBRE A LA INFORMACIÓN CON RECURSOS EDUCATIVOS ABIERTOS (REA) [Internet]. Com.ar. [citado el 25 de febrero de 2022]. Disponible en: https://produccion-animal.com.ar/temas_varios/temas_varios/96-reavladimirburgos.pdf

[8] Rhoads RA, Berdan J, Toven-Lindsey B. The open courseware movement in higher education: Unmasking power and raising questions about the movement's democratic potential: Open courseware movement in higher education. Educ Theory [Internet]. 2013;63(1):87–110. Disponible en: <http://dx.doi.org/10.1111/edth.12011>

[9] Sánchez-Vera M-M, León-Urrutia M, Davis H. Challenges in the creation, development and implementation of MOOCs: Web Science course at the University of Southampton. Comunicar [Internet]. 2015 [citado el 25 de febrero de 2022];22(44):37–44. Disponible en: <https://www.revistacomunicar.com/indice-en/articulo.php?numero=44-2015-04>

[10] Sicília M-Á. Más allá de los contenidos: compartiendo el diseño de los recursos educativos abiertos. RUSC Univ Knowl Soc J [Internet]. 2007 [citado el 25 de febrero de 2022]; Disponible en: <https://raco.cat/index.php/Rusc/article/view/58132> [11] Yuan L, Powell SJ. MOOCs and open education: Implications for higher education. 2013 [citado el 25 de febrero de 2022]; Disponible en: <https://e-space.mmu.ac.uk/619735/>

[12] Título: Diseño de Interfaces de Usuario Autor: B. Shneiderman, C. Plaisant Editorial: Pearson-Addison Wesley, 2006, 4^a ed.

[13] Glusac D, Karuovic D, Milanov D. Massive open online courses - pedagogical overview. En: Proceedings of the 2015 16th International Carpathian Control Conference(ICCC). IEEE; 2015.p.142–6. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=7145063>

[14] Martinez S. OCW (OpenCourseWare) and MOOC (Open Course Where?) [Internet]. Free.fr. [citado el 13 de marzo de 2022]. Disponible en: http://cdlh7.free.fr/OCWC2014/Final_papers/Paper16.pdf

[15] Li C-S, Irby B. An overview of online education: Attractiveness, benefits, challenges, concerns and recommendations. Coll Stud J [Internet]. 2008/06//Jun2008 Part A;42(2):449–58. Disponible en:
<https://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=32544879&site=ehost-live&scope=site>

[16] Kesim M, Altinpulluk H. A theoretical analysis of moocs types from a perspective of learning theories. Procedia Soc Behav Sci [Internet]. 2015 [citado el 20 de marzo de 2022];186:15–9. Disponible en:
<https://www.sciencedirect.com/science/article/pii/S1877042815023162>

[17] Asimov I. Entrevistado por Moyers B. “El Mundo de las Ideas”. 1988. Disponible en: <https://cuevadelobo.com/asimov-entrevistado-por-bill-moyers/>

Apéndice A

Páginas del sistema

En esta sección se encuentra el listado de pantallas del sistema ordenadas alfabéticamente, acompañadas por una pequeña descripción.

- *Administrar cursos.* Pantalla para que el administrador gestione como desee los cursos, pudiendo verlos, buscarlos, banearlos, desbanearlos y borrarlos.

The screenshot shows the 'Administrar cursos' (Manage Courses) page. At the top, there is a header bar with the 'ForeverLearn' logo, a 'Cursos' dropdown, a search bar containing '¿Qué te gustaría aprender?', and a red 'Iniciar sesión' button. Below the header, a teal sidebar on the left displays a user profile picture of Alexander Watson, his name, and his role as 'Administrador', with a 'Ver perfil' button. The main content area has a title 'Administrar cursos' and a search bar with placeholder text 'Buscas algún curso?'. Below this, there is a table listing five courses:

Icono	Título del curso	Acciones
	Angular Routing In-Depth	Ver curso Banear Eliminar
	Angular Unit Testing	Ver curso Desbanear Eliminar
	Introduction to TypeScript	Ver curso Banear Eliminar
	Learn Angular Fundamentals	Ver curso Banear Eliminar
	React Native	Ver curso Banear Eliminar

At the bottom of the page, there are navigation links for 'PREV < 1 2 > NEXT' and a footer bar with links for 'Síguenos' (Facebook, Twitter, LinkedIn), 'Centro de ayuda', 'Términos de servicio', 'Política de privacidad', and 'Copyright 2021 © Todos los derechos reservados'.

- *Administrar usuarios.* Pantalla para que el administrador gestione como desee los usuarios, pudiendo verlos, buscarlos, banearlos, desbanearlos y borrarlos.

Administrador

Administrador

Adrián Sancho Vázquez

Beatriz Cuesta Rodríguez

Carlos Menéndez Vilyuela

Diana Bustamante Ruiz

Elena Zambrano Gutiérrez

Ver perfil

Iniciar sesión

Administrador

Adrián Sancho Vázquez

Beatriz Cuesta Rodríguez

Carlos Menéndez Vilyuela

Diana Bustamante Ruiz

Elena Zambrano Gutiérrez

Ver usuario Banear Eliminar

Ver usuario Desbanear Eliminar

Ver usuario Banear Eliminar

Ver usuario Banear Eliminar

Ver usuario Banear Eliminar

Siguientes

Centro de ayuda Términos de servicio Política de privacidad

Copyright 2021 © Todos los derechos reservados.

- *Catálogo de cursos.* Página para ver y filtrar los cursos de un modo simple.

Filtros

Categoría

Todas las categorías

Crecimiento personal

Deportes

Disciplinas académicas

Diseño

Finanzas

Fotografía y video

Informática y software

Marketing

Música

Profesiones

Salud

Otros

Dificultad

Principiante

Intermedio

Avanzado

Cursos

Learn Angular

Build an iOS

Learn Photoshop

Craft by inVision

Build a WordPress

Become a React

Become a Redis

Angular Unit Testing

Introduction to TS

Angular Routing In...

InVision App

Become a DevOps

12 resultados en Todas las categorías

Siguientes

Centro de ayuda Términos de servicio Política de privacidad

- *Centro de ayuda.* Página en la se responden dudas sobre el funcionamiento y concepto del sistema.

Centro de ayuda

En el Centro de ayuda encontrarás información acerca del concepto detrás de ForeverLearn así como respuestas a las principales dudas que puedan surgir entorno a su uso.

Índice

- [¿Qué es ForeverLearn?](#)
- [¿Qué ofrece ForeverLearn a los profesores?](#)
- [¿Qué ofrece ForeverLearn a los alumnos?](#)
- [¿Cómo utiliza ForeverLearn el concepto de CMOOOC?](#)
- [Reconocimientos](#)

- *Crear cursos.* Página para que un usuario pueda crear un curso.

Crear curso

Título	Introduce el título
Miniatura	Elegir archivo <input type="button" value="Browse"/>
Categoría	Elige una categoría
Dificultad	Elige una dificultad
Duración	Introduce la duración
Trailer	Elegir archivo <input type="button" value="Browse"/>
Descripción	Introduce la descripción

- *Cursos completados.* Página en la que el usuario puede ver su listado de cursos completados.

Cursos completados

- Learn Angular
- Build an iOS Application in Swift
- Learn Photoshop
- Craft by inVision

< PREV 1 2 NEXT >

- *Cursos deseados.* Página en la que el usuario puede ver el listado de cursos a los que ha dado me gusta.

Cursos deseados

- Introduction to Typescript
- Angular Routing In-Depth
- inVision App
- Become a DevOps Engineer

< PREV 1 2 NEXT >

- *Cursos matriculados.* Página en la que el usuario puede ver el listado de cursos en los que se ha matriculado.

- *Cursos propios.* Página en la que el usuario puede ver el listado de curso que él mismo ha creado.

- *Detalles de la lección.* Página en la que el usuario puede ver el contenido de una lección de un tema de un curso.

Angular Fundamentals
Eddie Murray

Introduction to TypeScript

JavaScript is now used to power backends, create hybrid mobile applications, architect cloud solutions, design neural networks and even control robots. Enter TypeScript: a superset of JavaScript for scalable, secure, performant and feature-rich applications.

- *Detalles del curso.* Página en la que el usuario puede ver el contenido de un curso.

Eddie Bryan
Profesor

6 horas Principiante

★★★★★ 1 like 4 reviews

Angular Fundamentals

This course will teach you the fundamentals of working with Angular 2. You will learn everything you need to know to create complete applications including: components, services, directives, pipes, routing, HTTP, and even testing.

[Ver trailer](#)

Índice

- Course Overview
- Getting Started with Angular
- Creating and Communicating Between Angular Components
- Exploring the Angular Template Syntax

Requisitos
✓ Fundamentals of working with Angular

Objetivos
✓ Fundamentals of working with Angular

- *Editar contraseña.* Página para que el usuario pueda configurar su contraseña.

Cambiar contraseña

Nueva contraseña Introduce tu nueva contraseña

Confirmar contraseña Vuelve a introducir tu nueva contraseña

Información básica
Información de perfil
Cambiar contraseña

Guardar cambios

Siguemos [Centro de ayuda](#) [Términos de servicio](#) [Política de privacidad](#)
Copyright 2021 © Todos los derechos reservados.

- *Editar curso.* Página para que el usuario pueda editar uno los cursos que ha creado.

Editar curso

Título	<input type="text"/> Introduce el título
Miniatrura	<input type="file"/> Elegir archivo Browse
Categoría	<input type="text"/> Elige una categoría
Dificultad	<input type="text"/> Elige una dificultad
Duración	<input type="text"/> Introduce la duración
Trailer	<input type="file"/> Elegir archivo Browse
Descripción	<input type="text"/> Introduce la descripción

- *Editar información básica.* Página para que el usuario pueda configurar su información base.

Alexander Watson
Usuario

Información básica

Nombre	Alexander
Apellidos	Watson
Email	alexander.watson@fake-mail.com

● Información básica
● Información de perfil
● Cambiar contraseña

Guardar cambios

Sigueros [Centro de ayuda](#) [Términos de servicio](#) [Política de privacidad](#)
Copyright 2021 © Todos los derechos reservados.

- *Editar perfil.* Página para que el usuario pueda configurar su información de perfil.

Alexander Watson
Usuario

Información de perfil

Tu foto		Elegir archivo	Browse
Descripción	Introduce tu descripción		
Contacto	Introduce tu información de contacto		
Facebook		Introduce tu nombre de usuario de facebook	
Twitter		Introduce tu nombre de usuario de Twitter	
Youtube		Introduce el nombre de tu canal de YouTube	

● Información básica
● Información de perfil (selected)
● Cambiar contraseña

Guardar cambios

Sigueros [Centro de ayuda](#) [Términos de servicio](#) [Política de privacidad](#)
Copyright 2021 © Todos los derechos reservados.

- *Iniciar de sesión.* Página para que el usuario pueda iniciar sesión en el sistema mediante sus credenciales.

The screenshot shows the 'Iniciar sesión' (Login) page of the ForeverLearn website. At the top, there's a dark header with the 'ForeverLearn' logo, a 'Cursos' dropdown, a search bar, and a red 'Iniciar sesión' button. Below the header, the main title 'Iniciar sesión' is displayed next to a user icon, with the subtitle 'Configuración de cuenta'. To the right, there's a link '¿No tienes cuenta? ¡Regístrate hoy!'. The central part of the page contains two input fields: 'Email:' with placeholder 'Introduce tu correo electrónico' and 'Contraseña:' with placeholder 'Introduce tu contraseña'. Below the password field is a link '¿Has olvidado tu contraseña?'. A red 'Login' button is positioned at the bottom right of the form area. At the very bottom of the page is a dark footer with links for 'Síguenos' (Facebook, Twitter, YouTube), 'Centro de ayuda', 'Términos de servicio', 'Política de privacidad', and copyright information 'Copyright 2021 © Todos los derechos reservados.'

- *Menú de administrador.* Página con las acciones que puede realizar el usuario administrador.

The screenshot shows the administrator menu page of the ForeverLearn website. The top navigation bar is identical to the login page. The main content area features a profile section for 'Alexander Watson' (Administrator) with a 'Ver perfil' button. Below this, there are two large cards: 'Administrar cursos' (Manage Courses) showing a gear and a document icon, and 'Administrar usuarios' (Manage Users) showing two user profile icons and a gear icon. The bottom of the page has a dark footer with social media links, help links, and copyright information.

- *Página principal.* Página inicial que recibe el usuario cuando entra en el sistema.

The screenshot shows the ForeverLearn homepage. At the top, there's a navigation bar with the logo, a 'Cursos' dropdown, a search bar, and a 'Iniciar sesión' button. The main area has a teal background with a central white tablet-like interface displaying various metrics and charts. Below the tablet, there are three circular icons with text: 'Alumno' (Student), 'Profesor' (Teacher), and 'Acceso ilimitado' (Unlimited access). A large section titled 'Categorías' follows, featuring four icons representing different fields: a person at a desk, dumbbells, a building, and crossed pens.

- *Perfil de administrador.* Página de opciones del usuario administrador.

The screenshot shows the administrator profile page. It features a header with the user's name, 'Alexander Watson', and role, 'Usuario'. There's a 'Editar cuenta' button. Below the header, there are three main sections: 'Área profesor' (Professor area) showing a teacher icon, 'Área alumno' (Student area) showing a graduation cap icon, and 'Área administrador' (Administrator area) showing a police officer's hat icon.

- *Perfil de profesor.* Página en la que se puede ver la información de perfil y cursos creados por un usuario.

The screenshot shows a user profile for 'Elijah Murray' on the ForeverLearn website. At the top, there is a navigation bar with the logo, a search bar, and a 'Iniciar sesión' button. Below the header, Elijah's profile picture is displayed next to his name and title 'Profesor'. The main content area features two sections: 'Sobre mí' (About me) and 'Contacto' (Contact). Under 'Sobre mí', there is a bio describing Elijah's passion for cross-cultural advertising and academic foundations in psychology and sociology. Below this, there are four course cards: 'Angular Routing In-Depth', 'Introduction to TypeScript', 'Angular Unit Testing', and 'Learn Angular Fundamentals'. A footer bar at the bottom includes social media links, a copyright notice, and links to 'Centro de ayuda', 'Términos de servicio', and 'Política de privacidad'.

- *Perfil de usuario.* Página de opciones del usuario registrado.

The screenshot shows a user profile for 'Alexander Watson' on the ForeverLearn website. The layout is similar to the professor's profile, with a header, search bar, and 'Iniciar sesión' button. Alexander's profile picture and name are at the top. Below, there are two sections: 'Área profesor' (Professor Area) featuring a teacher icon, and 'Área alumno' (Student Area) featuring a student icon wearing a graduation cap. A footer bar at the bottom includes social media links, a copyright notice, and links to 'Centro de ayuda', 'Términos de servicio', and 'Política de privacidad'.

- *Política de privacidad.* Página en la que se puede ver la política de privacidad del sistema.

Política de privacidad

Creemos que debe conocer en todo momento qué datos de usted recopilamos y cómo los utilizamos, y qué debe poder tener un control efectivo sobre ambas cuestiones. Queremos darle poder para que tome las decisiones más adecuadas sobre la información que comparte con nosotros. Esta es la finalidad de la presente Política de privacidad.

Política de privacidad

1 Información que comparte con nosotros

Requerimos cierta información para prestarle nuestros servicios. Por ejemplo, usted debe tener una cuenta para crear un curso u obtener un certificado en ForeverLearn. Cuando usted opta por compartir la siguiente información con nosotros, la recopilamos y utilizamos para administrar nuestros servicios.

Información básica de la cuenta

Usted no necesita crear una cuenta para utilizar algunas de las características de nuestro servicio, como la búsqueda y visualización de los cursos. Si opta por crear una cuenta, le proporciona información adicional para que podamos prestarle nuestros servicios. En ForeverLearn esto incluye su nombre, sus apellidos, una contraseña y una dirección de correo electrónico. Su nombre y apellidos son ejemplo públicos, pero puede utilizar su nombre real o un seudónimo. También podrá facilitarnos información adicional, incluyendo su descripción, datos de contacto y otras redes sociales; todo lo anterior será público siempre. También puede crear y gestionar múltiples cuentas, por ejemplo,

Índice

- Información que comparte con nosotros
- Información que compartimos y revelamos
- Gestionar nuestra información personal sobre usted
- Los menores y nuestros servicios
- Nuestras operaciones globales y Transferencias de datos
- Cambios a esta Política de Privacidad

- *Registro.* Página para que el usuario pueda registrarse en el sistema.

Registrarse

Forma parte de una inmensa red de difusión y aprendizaje de conocimiento.

Nombre:
Introduce tu nombre

Apellidos:
Introduce tus apellidos

Email:
Introduce tu correo electrónico

Contraseña:
Introduce tu contraseña

Términos y condiciones
Al seleccionar esta casilla estoy aceptando los Términos de servicio y Política de privacidad de ForeverLearn

Crear cuenta

Síguenos

Centro de ayuda **Términos de servicio** **Política de privacidad**
Copyright 2021 © Todos los derechos reservados.

- *Términos de servicio.* Página en la que se pueden ver los términos de uso del sistema.

The screenshot shows the ForeverLearn website with a dark header bar. On the left is the logo, followed by the text "ForeverLearn". To the right are buttons for "Cursos" (Courses), a search bar with placeholder text "¿Qué te gustaría aprender?", and a red "Iniciar sesión" (Log in) button.

The main content area has a teal header with the title "Términos de servicio". Below it is a detailed text block about the terms of service. To the right is an "Índice" (Index) sidebar with links to other sections: "¿Quién puede hacer uso de los Servicios?", "Privacidad", "Contenido de los Servicios", "Uso de los Servicios", "Exclusiones y limitaciones de responsabilidad", and "General".

Términos de servicio

Estos Términos de servicio (los "Términos") rigen su acceso y uso de nuestros recursos (denominados de forma colectiva como los "Servicios"), y cualquier información, texto, enlace, gráfico, foto, audio, video o material o disposición de material cargados o que aparezcan en los Servicios (denominados de forma colectiva el "Contenido"). Al hacer uso de estos Servicios, usted accede a vincularse a estos Términos.

1 ¿Quién puede hacer uso de los Servicios?

Puede hacer uso de los Servicios solo si accede a firmar un contrato vinculante con Twitter y no es usted una persona vetada para hacer uso de los servicios de conformidad con la legislación de su jurisdicción aplicable. En cualquier caso, usted deberá tener al menos 13 años, para hacer uso de los Servicios. Si acepta estos Términos y usa los Servicios en nombre de una empresa, organización, gobierno u otra entidad jurídica, afirma y garantiza que está autorizado a hacerlo y cuenta con los poderes necesarios para obligarla al cumplimiento de estos Términos, en cuyo caso el uso de las palabras "usted", "su" y "sus" en estos Términos hará referencia a dicha entidad jurídica.

2 Privacidad

Nuestra Política de privacidad describe cómo manejamos la información que nos proporciona al hacer uso de nuestros Servicios. Comprende que mediante el uso que usted hace de nuestros Servicios, usted consiente a la recopilación y uso (tal

Apéndice B

Casos de uso

En esta sección se encuentra el listado de casos de uso del sistema que han sido considerados en la creación del diagrama de casos de uso.

- *Registrarse.*

Objetivo: Un usuario no registrado desea registrarse en la aplicación.

Actor: Usuario no registrado.

Precondición: El usuario debe acceder al sistema.

Postcondición: El usuario queda registrado en la base de datos.

Acción del actor	Respuesta del sistema
1. El usuario no registrado entra en la aplicación y pulsa el botón de “Iniciar sesión”.	2. El sistema devuelve al usuario la pantalla de inicio de sesión.
3. El usuario pulsa el botón de: “¿No tienes cuenta? ¡Regístrate hoy!”	4. El sistema devuelve al usuario la pantalla de registro.
5. El usuario no registrado rellena con su nombre, apellidos, correo electrónico y contraseña el formulario de registro, marca que acepta los términos y condiciones y pulsa el botón de “Crear cuenta”.	6. El sistema comprueba que el correo no exista en la base de datos. Si todo es correcto le registrará y le llavará a la página de acciones de usuario.

Cuadro B.1: Registrarse.

- *Iniciar sesión.*

Objetivo: Un usuario desea iniciar sesión en la aplicación.

Actor: Usuario registrado.

Precondición: El usuario debe estar previamente registrado.

Postcondición: El usuario ahora tiene acceso a todas las funciones de los usuarios registrados.

Acción del actor	Respuesta del sistema
1. El usuario registrado entra en la aplicación y pulsa el botón de “Iniciar sesión”.	2. El sistema devuelve al usuario la pantalla de inicio de sesión.
3. El usuario introduce el email y contraseña de su cuenta en el formulario que se le presenta y pulsa el botón de “Login”	4. El sistema comprueba que en la base de datos exista algún usuario cuyo correo y contraseña se correspondan con los que ha recibido del formulario. Si todo es correcto le llavará a la página de acciones de usuario.

Cuadro B.2: Iniciar sesión.

- *Cerrar sesión.*

Objetivo: Un usuario desea cerrar sesión en la aplicación.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe quedar sin muestras de que haya ningún usuario con su sesión activa.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Cerrar sesión” situado en el header.	2. El sistema devuelve al usuario la pantalla principal y desaparece todo rastro de la sesión del usuario.

Cuadro B.3: Cerrar sesión.

- *Configurar cuenta.*

Objetivo: Un usuario desea realizar cambios en su cuenta.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe reflejar los cambios que el usuario haya realizado sobre su cuenta.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Editar cuenta”.	4. El sistema devuelve la pantalla de edición de información básica.

5. El usuario edita la información que se desee de los campos que se presentan y pulsa el botón de “Guardar cambios”

6. El sistema recoge la información introducida por el usuario, actualiza dicha información en la base de datos y devuelve la pantalla de acciones de usuario.

Cuadro B.4: Configurar cuenta.

- *Editar información básica.*

Objetivo: Un usuario desea realizar cambios sobre los datos de registro de su cuenta.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe reflejar los cambios que el usuario haya realizado sobre sus datos de registro.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Editar cuenta”.	4. El sistema devuelve la pantalla de edición de información básica.

5. El usuario edita la información que se desee de los campos que se presentan y pulsa el botón de “Guardar cambios”

6. El sistema recoge la información introducida por el usuario, actualiza dicha información en la base de datos y devuelve la pantalla de acciones de usuario.

Cuadro B.5: Editar información básica.

- *Editar información de perfil.*

Objetivo: Un usuario desea realizar cambios sobre los datos de perfil de su cuenta.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe reflejar los cambios que el usuario haya realizado sobre su cuenta.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Editar cuenta”.	4. El sistema devuelve la pantalla de edición de información básica.
5. El usuario selecciona en el panel vertical de opciones “Información de perfil”.	6. El sistema devuelve la pantalla de edición de información de perfil.
7. El usuario edita la información que se desee de los campos que se presentan y pulsa el botón de “Guardar cambios”.	8. El sistema recoge la información introducida por el usuario, actualiza dicha información en la base de datos y devuelve la pantalla de acciones de usuario.

Cuadro B.6: Editar información de perfil.

- *Cambiar contraseña.*

Objetivo: Un usuario desea cambiar la contraseña de su cuenta.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: Cuando el usuario vuelva a iniciar sesión solo le dejará hacerlo con la nueva contraseña.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Editar cuenta”.	4. El sistema devuelve la pantalla de edición de información básica.
5. El usuario selecciona en el panel vertical de opciones “Cambiar contraseña”.	6. El sistema devuelve la pantalla de edición de contraseña.
7. El usuario introduce la nueva contraseña en los dos campos y pulsa el botón de “Guardar cambios”.	8. El sistema recoge la información introducida por el usuario, comprueba la introducida por el usuario, comprueba que las dos contraseñas introducidas sean la misma, la actualiza en el usuario y devuelve la pantalla de acciones de usuario.

Cuadro B.7: Cambiar contraseña.

- *Borrar cuenta.*

Objetivo: Un usuario desea borrar su cuenta en el sistema.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: En el sistema no debe quedar rastro del usuario ni de sus cursos, cuando intente iniciar sesión con sus antiguas credenciales no se lo permitirá.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Editar cuenta”.	4. El sistema devuelve la pantalla de edición de información básica.
5. El usuario selecciona en el panel vertical de opciones “Eliminar cuenta”.	6. El sistema devuelve la pantalla de borrar cuenta.
7. El usuario pulsa el botón de “Borrar cuenta”	8. El sistema cierra su sesión, borra sus cursos, lo elimina de la base de datos y le devuelve la pantalla principal.

Cuadro B.8: Borrar cuenta.

- *Ver listado de cursos.*

Objetivo: Un usuario desea ver todos los cursos existentes en el sistema.

Actor: Usuario no registrado.

Precondición: El usuario debe acceder al sistema.

Postcondición: El sistema debe mostrar todos los cursos hayan sido creados.

Acción del actor	Respuesta del sistema
1. El usuario no registrado pulsa el botón de “Buscar cursos” situado en la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos paginados.

Cuadro B.9: Ver listado de cursos.

- *Ver cursos por categoría.*

Objetivo: Un usuario desea ver todos los cursos que pertenecen a una categoría.

Actor: Usuario no registrado.

Precondición: El usuario debe acceder al sistema.

Postcondición: El sistema debe mostrar todos los cursos de la categoría seleccionada.

Acción del actor	Respuesta del sistema
1. El usuario no registrado pulsa el botón desplegable de “Cursos” situado en el header y selecciona la categoría de la que deseé buscar los cursos.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos de la categoría seleccionada paginados.

Cuadro B.10: Ver cursos por categoría.

- *Ver cursos por título.*

Objetivo: Un usuario desea ver todos los cursos en base a un título.

Actor: Usuario no registrado.

Precondición: El usuario debe acceder al sistema.

Postcondición: El sistema debe mostrar todos los cursos que comparten el título buscado.

Acción del actor	Respuesta del sistema
1. El usuario no registrado introduce en el formulario “¿Qué te gustaría aprender?” del header el título en base al que desea buscar cursos y pulsa sobre el ícono de la lupa.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos que comparten el título introducido.

Cuadro B.11: Ver cursos por título.

- *Ver curso.*

Objetivo: Un usuario desea ver el contenido de un curso

Actor: Usuario no registrado.

Precondición: El usuario debe acceder al sistema.

Postcondición: El sistema debe mostrar toda la información, temario, lecciones, requisitos y objetivos del curso.

Acción del actor	Respuesta del sistema
<p>1. El usuario no registrado pulsa el botón de “Buscar cursos” situado en la pantalla principal.</p> <p>3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.</p>	<p>2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos paginados.</p> <p>4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.</p>

Cuadro B.12: Ver curso.

- *Ver perfil de usuario.*

Objetivo: Un usuario desea ver el perfil de otro usuario del sistema.

Actor: Usuario no registrado.

Precondición: El usuario debe acceder al sistema.

Postcondición: El sistema debe mostrar toda la información de un usuario, su descripción, datos de contacto, redes sociales y cursos que ha creado.

Acción del actor	Respuesta del sistema
<p>1. El usuario no registrado pulsa el botón de “Buscar cursos” situado en la pantalla principal.</p>	<p>2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos paginados.</p>
<p>3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.</p>	<p>4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.</p>
<p>5. El usuario pulsará en nombre del profesor del curso situado en la banda blanca horizontal debajo de la descripción del curso.</p>	<p>6. El sistema devolverá la página de perfil del usuario seleccionado con toda su información correspondiente y cursos creados por él.</p>

Cuadro B.13: Ver perfil de usuario.

- *Acceder al área de profesor.*

Objetivo: Un usuario desea acceder a su área de profesor.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe devolverle la pantalla con las opciones de profesor.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.

Cuadro B.14: Acceder al área de profesor.

■ *Crear curso.*

Objetivo: Un usuario desea crear un nuevo curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe mostrar el nuevo curso en el listado de cursos y en la lista de propios de su autor.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario pulsa el botón de “Crear curso”.	6. El sistema devuelve la pantalla de creación de curso.
7. El usuario rellena los campos del formulario con la información del curso que desea crear.	8. El sistema crea un curso con los datos introducidos y asocia como autor al usuario.

Cuadro B.15: Crear curso.

■ *Editar curso.*

Objetivo: Un usuario desea editar alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y haber creado el curso que desea modificar.

Postcondición: El sistema debe mostrar el curso que ya estaba creado con su información actualizada.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. El usuario baja hasta la sección editar del curso donde modificará aquellos datos que desee.	8. El sistema edita aquellos campos del curso que el usuario ha introducido y guarda los cambios.

Cuadro B.16: Editar curso.

■ *Borrar curso.*

Objetivo: Un usuario desea borrar alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y haber creado el curso que desea borrar.

Postcondición: El sistema debe eliminar el curso de todos los listados en los que aparezca y borrarlo de su base de datos.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Borrar curso”.	6. El sistema elimina el curso de todos los listados en los que aparece y finalmente lo borra de la base de datos.

Cuadro B.17: Borrar curso.

■ *Crear tema.*

Objetivo: Un usuario desea crear un tema para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y haber creado el

curso para el que quiere crear temas.

Postcondición: El sistema debe mostrar el nuevo tema cuando se acceda al curso.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. El usuario pulsa el botón “Crear tema” en la sección de temario, a continuación se le mostrará una ventana modal con un formulario que debe llenar con los datos del tema.	8. El sistema recibe dichos datos con los que crea el tema, lo guarda en la base de datos, a continuación lo asigna al curso y finalmente guarda este en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.18: Crear tema.

■ *Editar tema.*

Objetivo: Un usuario desea editar un tema para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y haber creado para este el tema que desea editar.

Postcondición: El sistema debe mostrar la información del tema tal y como el usuario ha modificado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de temario, verá el listado de temas, una vez encuentre el tema que desea editar pulsará el icono de lápiz que se encuentra a su derecha, a continuación se le mostrará una ventana modal con un formulario con los datos del tema, editará los que deseé y pulsará “Editar tema”.	8. El sistema recibe información sobre el tema y los campos que se desean editar, los actualiza y guarda los cambios en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.19: Editar tema.

- *Borrar tema.*

Objetivo: Un usuario desea borrar un tema para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y haber creado para este el tema que desea borrar.

Postcondición: El sistema debe mostrar el curso sin el tema que se ha borrado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de temario, verá el listado de temas, una vez encuentre el tema que desea editar pulsará el icono de papelera que se encuentra a su derecha, a continuación se le mostrará una ventana modal que le preguntará si está seguro de que desea borrar el tema, si este es el caso pulsará “Borrar tema”.	8. El sistema recibe que tema se desea borrar, lo quita del temario del curso y lo elimina de la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.20: Borrar tema.

■ *Crear lección.*

Objetivo: Un usuario desea crear una lección para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y dentro de él el tema para el que quiere crear la lección.

Postcondición: El sistema debe mostrar la nueva lección dentro de su correspondiente tema cuando se acceda al curso.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. El usuario seleccionará en la sección de temario el tema para el que quiere crear la lección, a la derecha de este aparecerá un botón “Crear lección” que debe pulsar, se le abrirá una ventana modal con campos que debe llenar, una vez hecho pulsará el botón “Crear lección” que aparece en esta.	8. El sistema recibe dichos datos con los que crea la lección, la guarda en la base de datos, a continuación la asocia al tema y finalmente guarda este en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.21: Crear lección.

■ *Editar lección.*

Objetivo: Un usuario desea editar una lección para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso, dentro de este el tema y dentro del tema la lección que desea editar.

Postcondición: El sistema debe mostrar la información de la lección tal y como el usuario ha modificado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de temario buscará el tema al que pertenezca la lección que quiere editar, una vez encuentre el tema abrirá su sección desplegable donde podrá ver la lección a modificar, deberá pulsar el icono de lápiz que se encuentra a su derecha, a continuación se le mostrará una ventana modal con un formulario con los datos de la lección, editará los que deseé y pulsará “Editar lección”.	8. El sistema recibe información sobre la lección y los campos que se desean editar, los actualiza y guarda los cambios en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.22: Editar lección.

■ *Borrar lección.*

Objetivo: Un usuario desea borrar una lección para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso, dentro de este el tema y dentro del tema la lección que desea borrar.

Postcondición: El sistema debe mostrar el curso sin la lección que se ha borrado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de temario buscará el tema al que pertenezca la lección que quiere borrar, una vez encuentre el tema abrirá su sección desplegable donde podrá ver la lección a borrar, deberá pulsar el icono de papelera que se encuentra a su derecha, a continuación se le mostrará una ventana modal que le preguntará si desea borrarla y pulsará “Borrar lección”.	8. El sistema recibe la lección que se desea borrar, la elimina del tema y la elimina de la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.23: Borrar lección.

- *Crear requisito.*

Objetivo: Un usuario desea crear un requisito para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y haber creado el curso para el que quiere crear requisitos.

Postcondición: El sistema debe mostrar el nuevo requisito cuando se acceda al curso.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. El usuario pulsa el botón “Crear requisito” en la sección de temario, a continuación se le mostrará una ventana modal con un formulario que debe llenar con los datos del requisito.	8. El sistema recibe dichos datos con los que crea el requisito, lo guarda en la base de datos, a continuación lo asigna al curso y finalmente guarda este en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.24: Crear requisito.

■ *Editar requisito.*

Objetivo: Un usuario desea editar un requisito para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y haber creado para este el requisito que desea editar.

Postcondición: El sistema debe mostrar la información del requisito tal y como el usuario ha modificado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de requisitos, verá el listado de requisitos, una vez encuentre el requisito que desea editar pulsará el icono de lápiz que se encuentra a su derecha, a continuación se le mostrará una ventana modal con un formulario con los datos del requisito, editará los que desee y pulsará “Editar requisito”.	8. El sistema recibe información sobre el requisito y los campos que se desean editar, los actualiza y guarda los cambios en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.25: Editar requisito.

- *Borrar requisito.*

Objetivo: Un usuario desea borrar un requisito para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y haber creado para este el requisito que desea borrar.

Postcondición: El sistema debe mostrar el curso sin el requisito que se ha borrado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de requisitos, verá el listado de requisitos, una vez encuentre el requisito que desea editar pulsará el icono de papelera que se encuentra a su derecha, a continuación se le mostrará una ventana modal que le preguntará si está seguro de que desea borrar el requisito, si este es el caso pulsará “Borrar requisito”.	8. El sistema recibe que requisito se desea borrar, lo quita del listado de requisitos del curso y lo elimina de la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.26: Borrar requisito.

■ *Crear objetivo.*

Objetivo: Un usuario desea crear un objetivo para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y haber creado el curso para el que quiere crear objetivos.

Postcondición: El sistema debe mostrar el nuevo objetivo cuando se acceda al curso.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. El usuario pulsa el botón “Crear objetivo” en la sección de objetivos, a continuación se le mostrará una ventana modal con un formulario que debe llenar con los datos del objetivo.	8. El sistema recibe dichos datos con los que crea el objetivo, lo guarda en la base de datos, a continuación lo asigna al curso y finalmente guarda este en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.27: Crear objetivo.

- *Editar objetivo.*

Objetivo: Un usuario desea editar un objetivo para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y haber creado para este el objetivo que desea editar.

Postcondición: El sistema debe mostrar la información del objetivo tal y como el usuario ha modificado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de objetivos, verá el listado de objetivos, una vez encuentre el requisito que desea editar pulsará el icono de lápiz que se encuentra a su derecha, a continuación se le mostrará una ventana modal con un formulario con los datos del objetivo, editará los que deseé y pulsará “Editar objetivo”.	8. El sistema recibe información sobre el objetivo y los campos que se desean editar, los actualiza y guarda los cambios en la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.28: Editar objetivo.

- *Borrar objetivo.*

Objetivo: Un usuario desea borrar un objetivo para alguno de sus cursos.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión, haber creado el curso y haber creado para este el objetivo que desea borrar.

Postcondición: El sistema debe mostrar el curso sin el objetivo que se ha borrado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área profesor”.	4. El sistema devuelve la pantalla de profesor.
5. El usuario selecciona de entre sus cursos aquel que desee modificar, pulsa sobre él y aparecerá un desplegable, en él debe pulsar el botón “Editar curso”.	6. El sistema devuelve la pantalla de edición de curso.
7. En la sección de objetivos, verá el listado de objetivos, una vez encuentre el objetivo que desea editar pulsará el icono de papelera que se encuentra a su derecha, a continuación se le mostrará una ventana modal que le preguntará si está seguro de que desea borrar el objetivo, si este es el caso pulsará “Borrar objetivo”.	8. El sistema recibe que objetivo se desea borrar, lo quita del listado de objetivos del curso y lo elimina de la base de datos, tras ello le devuelve la pantalla de edición del curso.

Cuadro B.29: Borrar objetivo.

■ *Acceder al área de alumno.*

Objetivo: Un usuario desea acceder a su área de alumno.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El sistema debe devolverle la pantalla con las opciones de alumno.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área alumno”.	4. El sistema devuelve la pantalla de cursos matriculados.

Cuadro B.30: Acceder al área de alumno.

■ *Matricularse.*

Objetivo: Un usuario desea matricularse en un curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso en el que desea matricularse debe existir, no puede ser autor del mismo y no debe haberse

matriculado en él previamente.

Postcondición: El curso debe aparecer en la lista de cursos matriculados del usuario.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. En la sección inicial debe pulsar el botón que indica “Matricularse”.	6. El sistema añadirá el curso a la lista de cursos en los que está matriculado el usuario y le devolverá la página de cursos matriculados donde ahora aparecerá dicho curso.

Cuadro B.31: Matricularse.

■ *Desmatricularse.*

Objetivo: Un usuario desea desmatricularse en un curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso del que desea desmatricularse debe existir y debe haberse matriculado en él previamente.

Postcondición: El curso no debe aparecer en la lista de cursos matriculados del usuario.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal. 3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema. 4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. En la sección inicial debe pulsar el botón que indica “Desmatricularse”.	6. El sistema quitará el curso de la lista de cursos en los que está matriculado el usuario y le devolverá la página de cursos matriculados donde ya no se encontrará dicho curso.

Cuadro B.32: Desmatricularse.

- *Ver listado de cursos matriculados.*

Objetivo: Un usuario desea acceder a la lista de cursos en los que está matriculado.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El usuario debe poder ver en una pantalla todos los cursos en los que se ha matriculado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área alumno”.	4. El sistema devuelve la pantalla de cursos matriculados.

Cuadro B.33: Ver listado de cursos matriculados.

- *Añadir a deseados.*

Objetivo: Un usuario desea añadir un curso a su lista de cursos deseados.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso que desea añadir a deseados debe existir, no puede ser autor del mismo y no debe haberlo añadido previamente.

Postcondición: El curso debe aparecer en la lista de cursos deseados del usuario.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. En la sección inicial debe pulsar el botón que indica “Añadir a deseados”.	6. El sistema añadirá el curso a la lista de cursos deseados del usuario y le devolverá la página de cursos deseados donde ahora aparecerá dicho curso.

Cuadro B.34: Añadir a deseados.

■ *Eliminar de deseados.*

Objetivo: Un usuario desea quitar un curso de su lista de cursos deseados.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso que desea eliminar de la lista de deseados debe existir, no puede ser autor del mismo y debe haberlo añadido a la lista previamente.

Postcondición: El curso no debe aparecer en la lista de cursos deseados del usuario.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. En la sección inicial debe pulsar el botón que indica ”Quitar de deseados”.	6. El sistema quitará el curso de la lista de cursos deseados del usuario y le devolverá la página de cursos deseados donde ya no se encontrará dicho curso.

Cuadro B.35: Eliminar de deseados.

- *Ver listado de cursos deseados.*

Objetivo: Un usuario desea acceder a su lista de cursos deseados.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El usuario debe poder ver en una pantalla todos los cursos que ha añadido a su lista de deseados.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área alumno”.	4. El sistema devuelve la pantalla de cursos matriculados.
5. El usuario debe pulsar en la barra horizontal de opciones el botón “Cursos deseados”.	4. El sistema devuelve la pantalla de cursos deseados.

Cuadro B.36: Ver listado de cursos deseados.

- *Ver listado de cursos completados.*

Objetivo: Un usuario desea la lista de cursos que ha superado.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión.

Postcondición: El usuario debe poder ver en una pantalla todos los cursos que ha

completado.

Acción del actor	Respuesta del sistema
1. El usuario registrado pulsa el botón de “Ver perfil” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de usuario.
3. El usuario pulsa el botón de “Área alumno”.	4. El sistema devuelve la pantalla de cursos matriculados.
5. El usuario debe pulsar en la barra horizontal de opciones el botón “Cursos completados”.	4. El sistema devuelve la pantalla de cursos completados.

Cuadro B.37: Ver listado de cursos completados.

■ *Dar me gusta.*

Objetivo: Un usuario desea dar me gusta a un curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso al que desea dar me gusta debe existir.

Postcondición: El curso al que ha dado me gusta debe reflejarlo mostrando el ícono de pulgar arriba de un color destacado verde.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. El usuario pulsará en el ícono de pulgar arriba situado en la banda blanca horizontal debajo de la descripción del curso.	6. El sistema añadirá el me gusta a la lista de votos positivos del curso, añade el curso a la lista de cursos a los que el usuario ha dado me gusta y devuelve la página del curso, ahora resaltando el ícono de pulgar arriba y mostrando su puntuación actualizada.

Cuadro B.38: Dar me gusta.

■ *Quitar me gusta.*

Objetivo: Un usuario desea quitar su me gusta a un curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso al que desea dar me gusta debe existir y debe haberle dado me gusta previamente.

Postcondición: El curso del que ha quitado su me gusta debe reflejarlo mostrando el ícono de pulgar arriba de un color gris.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. El usuario pulsará en el ícono de pulgar arriba que estará previamente resaltado situado en la banda blanca horizontal debajo de la descripción del curso.	6. El sistema quitará el me gusta a la lista de votos positivos del curso, eliminará el curso de la lista de cursos a los que el usuario ha dado me gusta y devuelve la página del curso, ahora mostrando el ícono de pulgar arriba de color gris y mostrando la puntuación del curso actualizada.

Cuadro B.39: Quitar me gusta.

■ *Dar no me gusta.*

Objetivo: Un usuario desea dar no me gusta a un curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso al que desea dar no me gusta debe existir.

Postcondición: El curso al que ha dado no me gusta debe reflejarlo mostrando el ícono de pulgar abajo de un color destacado verde.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. El usuario pulsará en el ícono de pulgar abajo situado en la banda blanca horizontal debajo de la descripción del curso.	6. El sistema añadirá el no me gusta a la lista de votos negativos del curso, inserta el curso en la lista de cursos a los que el usuario ha dado no me gusta y devuelve la página del curso, ahora resaltando el ícono de pulgar abajo y mostrando su puntuación actualizada.

Cuadro B.40: Dar no me gusta.

- *Quitar no me gusta.*

Objetivo: Un usuario desea quitar su no me gusta a un curso.

Actor: Usuario registrado.

Precondición: El usuario debe haber iniciado previamente sesión y el curso al que desea dar no me gusta debe existir y debe haberle dado no me gusta previamente.

Postcondición: El curso del que ha quitado su no me gusta debe reflejarlo mostrando el ícono de pulgar abajo de un color gris.

Acción del actor	Respuesta del sistema
1. El usuario registrado busca el curso pulsando el botón “Buscar cursos” de la pantalla principal.	2. El sistema devuelve al usuario la pantalla librería que muestra todos los cursos del sistema.
3. El usuario busca en la librería el curso que desea y una vez encontrado pulsa sobre él para desplegar la ventana modal con información adicional, en esta encontrará un botón “Ver curso” que debe pulsar.	4. El sistema debe devolver la página de curso con la información correspondiente al curso que desea ver.
5. El usuario pulsará en el ícono de pulgar abajo que estará previamente resaltado situado en la banda blanca horizontal debajo de la descripción del curso.	6. El sistema quitará el no me gusta a la lista de votos negativos del curso, eliminará el curso de la lista de cursos a los que el usuario ha dado no me gusta y devuelve la página del curso, ahora mostrando el ícono de pulgar abajo de color gris y mostrando la puntuación del curso actualizada.

Cuadro B.41: Quitar no me gusta.

- *Acceder al área de administrador.*

Objetivo: Un usuario desea acceder a su área de administrador.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El sistema debe devolverle la pantalla con las opciones de administrador.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.

Cuadro B.42: Acceder al área de administrador.

- *Administrar usuarios.*

Objetivo: Un usuario desea ver el listado de usuarios de la aplicación para administrar.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario debe poder ver el listado de todos los usuarios del sistema

para administrar.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar usuarios”.	4. El sistema devuelve la pantalla de administrar usuarios.

Cuadro B.43: Administrar usuarios.

- *Administrar usuarios por nombre.*

Objetivo: Un usuario desea ver buscar por nombre usuarios de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario debe poder ver el listado de todos los usuarios que existen en el sistema que comparten el nombre buscado.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar usuarios”.	4. El sistema devuelve la pantalla de administrar usuarios.
5. El usuario introduce el nombre que desea buscar en el campo “¿Buscas algún usuario?” y pulsa el icono de la lupa.	6. El sistema devuelve la pantalla de administrar usuarios en la que muestra el listado de todos los usuarios que comparten el nombre introducido.

Cuadro B.44: Administrar usuarios por nombre.

- *Banear usuario.*

Objetivo: Un usuario desea banear a otro usuario de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y el usuario que desea banear debe existir y no estar previamente baneado.

Postcondición: El usuario administrador debe poder ver en el listado de usuarios alguna indicación de que el usuario ha sido baneado, por su parte el usuario que ha

sido baneado no podrá iniciar sesión en el sistema hasta que sea desbaneado.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar usuarios”.	4. El sistema devuelve la pantalla de administrar usuarios.
5. Encuentra el usuario que busca y pulsa el botón “Banear” a la derecha de este.	6. El sistema marca que el usuario seleccionado está baneado, cuando intente iniciar sesión en el sistema no se lo permitirá debido a esta marca y finalmente devuelve la pantalla de administrar usuarios, pero para el botón “Desbanear”.

Cuadro B.45: Banear usuario.

■ *Desbanear usuario.*

Objetivo: Un usuario desea desbanear a otro usuario de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y el usuario que desea desbanear debe existir y estar previamente baneado.

Postcondición: El usuario administrador debe poder ver en el listado de usuarios alguna indicación de que el usuario ha sido desbaneado, por su parte el usuario que ha sido desbaneado ahora podrá iniciar sesión en el sistema sin problemas.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar usuarios”.	4. El sistema devuelve la pantalla de administrar usuarios.
5. Encuentra el usuario que busca y pulsa el botón “Desbanear” a la derecha de este.	6. El sistema quita la marca de que el usuario seleccionado estaba baneado, ahora podrá iniciar sesión en el sistema y finalmente devuelve la pantalla de administrar usuarios, pero para el usuario desbaneado ahora aparecerá un botón “Banear”.

Cuadro B.46: Desbanear usuario.

- *Eliminar usuario.*

Objetivo: Un usuario desea eliminar a otro usuario de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y el usuario que desea eliminar debe existir.

Postcondición: El usuario eliminado ya no aparecerá en ningún listado, se borraran sus curso del sistema y cuando intente iniciar sesión no podrá.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header. 3. El usuario pulsa el botón de “Administrar usuarios”.	2. El sistema devuelve al usuario la pantalla de acciones de administrador. 4. El sistema devuelve la pantalla de administrar usuarios.
5. Encuentra el usuario que busca y pulsa el botón “Eliminar” a la derecha de este.	6. El sistema primero elimina todos los cursos que ha creado el usuario seleccionado, a continuación lo elimina del sistema y finalmente devuelve la pantalla de administrar usuarios sin que aparezca en el listado.

Cuadro B.47: Eliminar usuario.

- *Administrar cursos.*

Objetivo: Un usuario desea ver el listado de cursos de la aplicación para administrar.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario debe poder ver el listado de todos los cursos del sistema para administrar.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar cursos”.	4. El sistema devuelve la pantalla de administrar cursos.

Cuadro B.48: Administrar cursos.

- *Administrar cursos por nombre.*

Objetivo: Un usuario desea ver buscar por nombre cursos de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario debe poder ver el listado de todos los cursos que existen en el sistema que comparten el nombre buscado.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar cursos”.	4. El sistema devuelve la pantalla de administrar cursos.
5. El usuario introduce el nombre que desea buscar en el campo “¿Buscas algún curso?” y pulsa el icono de la lupa.	6. El sistema devuelve la pantalla de administrar cursos en la que muestra el listado de todos los cursos que comparten el nombre introducido.

Cuadro B.49: Administrar cursos por nombre.

- *Banear curso.*

Objetivo: Un usuario desea banear un curso de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y el curso que desea banear debe existir y no estar previamente baneado.

Postcondición: El usuario administrador debe poder ver en el listado de cursos alguna indicación de que el curso ha sido baneado, el curso baneado no se mostrará en el sistema hasta que sea desbaneado.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar cursos”.	4. El sistema devuelve la pantalla de administrar cursos.
5. Encuentra el curso que busca y pulsa el botón “Banear” a la derecha de este.	6. El sistema marca que el curso seleccionado está baneado, no permitirá que se acceda a él y finalmente devuelve la pantalla de administrar cursos, pero para el curso baneado ahora aparecerá un botón “Desbanear”

Cuadro B.50: Banear curso.

- *Desbanear curso.*

Objetivo: Un usuario desea desbanear un curso de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y el curso que desea desbanear debe existir y estar previamente baneado.

Postcondición: El usuario administrador debe poder ver en el listado de cursos alguna indicación de que el curso ha sido desbaneado, ahora el curso será accesible en el sistema.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar cursos”.	4. El sistema devuelve la pantalla de administrar cursos.
5. Encuentra el curso que busca y pulsa el botón “Desbanear” a la derecha de este.	6. El sistema quita la marca de que el curso seleccionado estaba baneado, ahora aparecerá de nuevo en el sistema, devuelve la pantalla de administrar cursos, pero para el curso desbaneado ahora aparecerá un botón “Banear”.

Cuadro B.51: Desbanear curso.

- *Eliminar curso.*

Objetivo: Un usuario desea eliminar un curso de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y el curso que desea eliminar debe existir.

Postcondición: El curso eliminado ya no aparecerá en ningún listado y se borrará del sistema.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar cursos”.	4. El sistema devuelve la pantalla de administrar cursos.
5. Encuentra el curso que busca y pulsa el botón “Eliminar” a la derecha de este.	6. El sistema primero elimina el curso de la base de datos y finalmente devuelve la pantalla de administrar cursos sin que aparezca en el listado.

Cuadro B.52: Eliminar curso.

- *Administrar categorías.*

Objetivo: Un usuario desea ver el listado de categorías de la aplicación para administrar.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario debe poder ver el listado de todas las categorías del sistema para administrar.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar categorías”.	4. El sistema devuelve la pantalla de administrar categorías.

Cuadro B.53: Administrar categorías.

- *Administrar categorías por nombre.*

Objetivo: Un usuario desea ver buscar por nombre categorías de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario debe poder ver el listado de todas las categorías que existen en el sistema que comparten el nombre buscado.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar categorías”.	4. El sistema devuelve la pantalla de administrar categorías.
5. El usuario introduce el nombre que desea buscar en el campo “¿Buscas alguna categorías?” y pulsa el icono de la lupa.	6. El sistema devuelve la pantalla de administrar categorías en la que muestra el listado de todas las categorías que comparten el nombre introducido.

Cuadro B.54: Administrar categorías por nombre.

■ *Crear categoría.*

Objetivo: Un usuario desea crear una categoría para los cursos de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión y ser administrador.

Postcondición: El usuario administrador debe poder ver en el listado de categorías que aparece la que ha creado.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar categorías”.	4. El sistema devuelve la pantalla de administrar categorías.
5. El usuario pulsa el botón “Crear categoría” que desplegará una ventana modal que le pide los datos de la categoría a crear, una vez introducidos pulsa el botón “Crear categoría” que aparece en la ventana.	6. El sistema recoge la información, crea una categoría a partir de ella y devuelva la página de administrar categorías en la que ahora aparecerá la que se ha creado.

Cuadro B.55: Crear categoría.

- *Editar categoría.*

Objetivo: Un usuario desea editar una de las categorías de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y la categoría que desea editar debe existir.

Postcondición: El usuario administrador debe poder ver en el listado de categorías la que ha modificado con los cambios aplicados.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar categorías”.	4. El sistema devuelve la pantalla de administrar categorías.
5. Encuentra en el listado la categoría que desea editar y pulsa el botón de “Editar categoría” a su derecha, se despliega una ventana modal con la información que desea editar, introduce los cambios y pulsa el botón “Editar categoría” presente en la ventana.	6. El sistema modifica la categoría con los datos introducidos, guarda los cambios en la base de datos y devuelve la pantalla de administrar categorías.

Cuadro B.56: Editar categoría.

- *Eliminar categoría.*

Objetivo: Un usuario desea eliminar una categoría de la aplicación.

Actor: Usuario administrador.

Precondición: El usuario debe haber iniciado previamente sesión, ser administrador y la categoría que desea eliminar debe existir.

Postcondición: La categoría eliminada ya no aparecerá en ningún listado y todos los cursos que estaban clasificados con esta categoría ahora pasarán a estar bajo la categoría de “Otros”.

Acción del actor	Respuesta del sistema
1. El usuario administrador pulsa el botón de “Administrar” situado en el header.	2. El sistema devuelve al usuario la pantalla de acciones de administrador.
3. El usuario pulsa el botón de “Administrar categorías”.	4. El sistema devuelve la pantalla de administrar categorías.
5. Encuentra la categoría que busca y pulsa el botón “Eliminar” a la derecha de esta.	6. El sistema primero cambia la categoría de todos los cursos que tenían esta categoría a la de “Otros”, a continuación la elimina del sistema y finalmente devuelve la pantalla de administrar categorías sin que aparezca en el listado.

Cuadro B.57: Eliminar categoría.

