



ULPGC COURSE

Presentado por: Mario Rene Peñate Fariñas
Titulación: Grado en Ingeniería Informática
Tutor: Rodríguez Rodríguez, Abraham
Las Palmas de Gran Canaria, 13 de Julio de 2018

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D/D^a Mario Rene Peñate Fariñas, autor del Trabajo de Fin de Título ULPGC COURSE correspondiente a la titulación Grado en Ingeniería Informática,

SOLICITA

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

- Se ha iniciado o hay intención de iniciarlo (defensa no pública).
 No está previsto.

Y para que así conste firma la presente.

Las Palmas de Gran Canaria, a 16 de Julio de 2018.

El estudiante



Fdo.: _____

A rellenar y firmar **obligatoriamente** por el/los tutor/es

En relación a la presente solicitud, se informa:

Positivamente

Negativamente
(la justificación en caso de informe negativo deberá incluirse en el TFT05)

Fdo.: _____

Agradecimientos

Doy las gracias en primer lugar a mi familia, por compartir conmigo el viaje que ha representado el desarrollo de este proyecto. Les doy las gracias por haber estado siempre que la he necesitado, por todo el sacrificio que ha hecho por mí, por esforzarse en darme una buena educación y no permitir que me rindiera para poder terminar la carrera. En especial quiero agradecer a mi madre por su amor, sensibilidad, sabiduría y fuerza, me han inspirado a dar lo mejor de mí mismo. Y a mi padre, por todos sus valiosos consejos en los momentos difíciles, por su enorme esfuerzo de cuidar a la familia y mantenerla siempre unida y por haber sido siempre un modelo para mí.

Doy las gracias a mi hermano Eugenio por las incontables horas que ha dedicado en ayudarme durante todo el desarrollo de este proyecto. Sin lugar a duda, es la persona que más me ha ayudado en esta labor. Gracias a él pude mejorar el rendimiento de la aplicación para que fuera mas rápido aún. Me soportó infinidad de aburridas horas de brainstorming cada vez que algo en el software no funcionaba correctamente.

Doy las gracias a mi novia Dilma Alvarado Espinoza, por hacerme feliz, por hacer que pueda confiar en ti, por estar siempre en los buenos y en los malos momentos, por hacer que mis malas rachas sean cortas y bonitas, porque estás a mi lado. Me gustaría darte las gracias por todo lo que has hecho por mí y decirte que te quiero mucho.

Doy las gracias a mi mejor amigo Marc Bakker por la ayuda, brillante creatividad y entusiasmos que puso en el diseño de la portada. Le agradezco por haber sido el amigo que es desde la infancia y por todos los momentos que hemos compartido juntos.

Doy las gracias a David Santana Molina y a Dailos Aday Medina Méndez por todo el apoyo que me han dado durante los últimos años en mis proyectos personales. Agradezco las incalculables horas que han dedicado en enseñarme nuevos paradigmas de programación y técnicas relacionadas con la experiencia de usuario como pueden ser, los principios básicos del diseño centrado en el usuario, la usabilidad, ergonomía, diseño visual, diseño de información y diseño de interacción.

Y también doy las gracias a mi tutor Abraham Rodríguez por la dedicación, consejos que me han dado y resolverme las dudas que he tenido a lo largo del proyecto. También agradecerle a él y a los profesores por ser parte de mi formación como Ingeniero Informático.

Índice general

AGRADECIMIENTOS.....	3
CAPÍTULO 1	7
1.1. CONTEXTO	7
1.2. OBJETIVOS	7
1.3. JUSTIFICACIÓN DE LAS COMPETENCIAS CUBIERTAS	8
1.4. APORTACIONES.....	9
1.5. METODOLOGÍAS UTILIZADAS	10
1.5.1 Kanban.....	10
CAPÍTULO 2	13
2.1. ANÁLISIS.....	13
2.1.1 Aplicaciones Similares.....	13
2.2. PERFILES DE USUARIOS	14
2.3. REQUISITOS	14
2.3.1 Requisitos funcionales	15
2.3.2 Requisitos no funcionales.....	15
2.4. HISTORIAS DE USUARIO	15
2.4.1 Diseñar arquitectura	16
2.4.2 Registro de usuario.....	16
2.4.3. Login de usuario.....	16
2.4.4. Crear un curso	17
2.4.5. Añadir cursos al carrito.....	17
2.5. DIAGRAMA DE CASOS DE USO	18
2.6. MOCKUPS, DISEÑO DE LA INTERFAZ	19
2.7. MODELO DE NEGOCIO.....	19
2.8. NORMATIVA Y LEGISLACIÓN	21
CAPÍTULO 3	22
3.1. DESARROLLO.....	22
3.1.1 Características Implementadas	22
3.2. DIAGRAMA DE ARQUITECTURA DEL SISTEMA.....	22
3.3. ARQUITECTURA DE SOFTWARE.....	23
3.3.1 Redux.....	23
3.4. DISEÑO E IMPLEMENTACIÓN.....	25
3.4.1 Diagrama Entidad Relación.....	25
3.4.2 Diseño de Interfaz de Usuario.....	27
3.4.3 Arquitectura de Laravel.....	27
3.4.4 Arquitectura React JS.....	29
3.5. TECNOLOGÍAS DEL LADO DEL SERVIDOR	30
3.5.1 Servidor Web Apache.....	30
3.5.2 Lenguaje PHP 7.2.6.....	30
3.5.3 Framework Laravel 5.5	31
3.5.4 Base de Datos Postgres SQL.....	31
3.5.5 Composer.....	31
3.5.5 Amazon S3.....	31
3.5.6 Mailtrap.....	32
3.6 TECNOLOGÍAS DEL LADO DEL CLIENTE.....	32
3.6.1 Lenguaje HTML 5.2.....	32
3.6.2 Lenguaje CSS3	32
3.6.3 Sass.....	32
3.6.4 Framework Bootstrap 4	33
3.6.5 Lenguaje JavaScript	33
3.6.6 JQuery	33
3.6.7 React JS	33
3.6.8 Redux	33

3.6.9 Stripe JS.....	33
3.6.10 Font Awesome 5	34
3.6.11 Axios	34
3.6.12 Google Maps	34
3.7. HERRAMIENTAS DE DESARROLLO.....	35
3.7.1 Sistema Operativo Mac OS X.....	35
3.7.2 PhpStorm	35
3.7.3 Github.....	35
3.7.4 Taiga.....	35
3.7.5 Google Chrome.....	35
3.7.6 Postman	35
3.7.7 Adobe XD.....	36
3.7.8 Unsplash	36
3.7.9 Heroku.....	36
CAPÍTULO 4	37
CONCLUSIONES	37
4.1 Trabajos Futuros.....	37
BIBLIOGRAFÍA.....	38
APÉNDICE A.....	39
ANEXO HISTORIAS DE USUARIO	39
A.0.1 Registro de usuario	39
A.0.2 Login de usuario	39
A.0.3 Crear un curso.....	40
A.0.4 Editar un curso	40
A.0.5 Dar de baja a un curso.....	40
A.0.6 Ver cursos.....	41
A.0.7 Añadir curso al carrito.....	41
A.0.8 Eliminar curso del carrito	41
A.0.9 Comprar curso.....	41
A.0.10 Votar un curso.....	42
A.0.11 Buscar curso	42
A.0.12 Filtrar cursos por categoría	42
A.0.13 Filtrar cursos por dificultad	42
A.0.14 Ver mis cursos comprados.....	43
A.0.15 Cambiar contraseña.....	43
A.0.16 Enviar correo al administrador	43
A.0.17 Llamar por teléfono.....	43
A.0.18 Listar usuarios	44
A.0.19 Editar usuario	44
A.0.20 Dar de baja a un usuario.....	44
A.0.21 Buscar usuario.....	44
A.0.22 Cancelar Cuenta	45
A.0.23 Mandar mensajes	45
A.0.24 Ver mi perfil de usuario.....	45
A.0.25 Editar mi perfil de usuario.....	46
A.0.26 Listar profesores	46
A.0.27 Activar suscripción.....	46
A.0.28 Desactivar suscripción	47
A.0.29 Crear una sección dentro de un curso.....	47
A.0.30 Editar una sección dentro de un curso.....	47
A.0.31 Eliminar una sección dentro de un curso	48
A.0.32 Ordenar las secciones dentro de un curso.....	48
A.0.33 Crear un recurso dentro de una sección.....	48
A.0.34 Editar un recurso dentro de una sección.....	49
A.0.35 Eliminar un recurso dentro de una sección.....	49
A.0.36 Ordenar los recursos dentro de una sección.....	49
APÉNDICE B.....	50

B.1 MANUAL DE USUARIO	50
<i>B.1.1.1 Set Up Laravel (API)</i>	50
<i>B.1.1.2 Postman – Test API</i>	51
<i>B.1.2.1 Set Up React JS</i>	55
<i>B.1.3 Enlaces al código fuente</i>	55
<i>B.1.4 Videos sobre el uso de la aplicación</i>	55
APÉNDICE C	56
WIREFRAME.....	56
MOCKUP	64
PROTOTIPO.....	69
ARCHIVO DEL PROTOTIPO (ADOBE XD)	69

Capítulo 1

1.1. Contexto

No es secreto que Internet revolucionó la educación, y la razón es que revolucionó la manera de comunicar el conocimiento. Hoy en día tenemos toda clase de plataformas que aprovechan Internet para depositar y transmitir conocimiento, desde Wikipedia¹ hasta Twitter.

Las universidades mas importantes del mundo cuentan con una plataforma de formación online que permite a los estudiantes realizar cursos en línea sobre diversos temas como pueden ser: Física, Economía, Historia, Derecho, Matemáticas, Fotografía, Tecnología, etc.

Sin embargo, los estudiantes de la ULPGC no cuentan con una plataforma de enseñanza online donde puedan adquirir nuevos conocimientos, por lo que están obligados a irse a plataformas ya conocidas como puede ser edx², udacity³, etc.

1.2. Objetivos

El objetivo de este proyecto consiste en desarrollar una plataforma de enseñanza online asociada a la universidad de las Palmas de Gran Canaria. Esta plataforma ofrece cursos online para aquellas personas que quieran adquirir nuevos conocimientos o mejorar los conocimientos existentes.

Para ello se mostrará algunas de las características que debe tener la aplicación:

- Proporcionar al alumno una gran variedad de cursos.
- Permitir la formación continua.
- Permitir al alumno especializarse en el campo que mas le convenga.
- Permitir al alumno seguir mas de un curso a la vez.
- Dar a la ULPGC una gran base de usuarios que le permita captar profesionales.
- Posibilitar al personal docente de una herramienta extra de enseñanza.
- Proporcionar una fuente de ingresos para la ULPGC y su personal.
- Proporcionar a sus alumnos de un escaparate al mundo laboral.

¹ <https://es.wikipedia.org/wiki/Wikipedia:Portada>

² <https://www.edx.org/es>

³ <https://eu.udacity.com/>

1.3. Justificación de las competencias cubiertas

Con el desarrollo de este Trabajo de Fin de Grado de la carrera Grado en Ingeniería Informática, se deben cubrir las competencias asignadas a este, las cuales son:

- **CII01:** Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- **CII02:** Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social. Previo al desarrollo se realizó una correcta planificación obteniendo las funcionalidades vitales según la valoración de los usuarios en cuanto a utilidad. Por otra parte, se hizo un estudio del impacto económico y social que tendría la aplicación.
- **CII08:** Capacidad para analizar, diseñar y construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados. En este apartado hice un estudio de los diferentes lenguajes de programación con el objetivo de conseguir una aplicación robusta, segura, eficiente y escalable. Eligiendo Javascript (frontend) y PHP (backend) como lenguajes de programación.
- **CII12:** Conocimiento y aplicación de las características, funcionalidades y estructura de la base de datos, que permitan su adecuado uso, y el diseño y análisis e implementación de aplicaciones basadas en ellos. Debido al estudio de las características y funcionalidades necesarios para la aplicación, se desarrolló una estructura, diseño y análisis de la base de datos óptimos para su uso.
- **CII13:** Conocimiento y aplicación de las herramientas necesarios para el almacenamiento, procesamiento. Acceso a los Sistemas de información, incluidos los basados en web. Esto se refleja en el hecho de que la aplicación y todos sus datos residen en un sistema remoto, cuyo acceso se realiza en forma de lecturas y escrituras a través de peticiones de red.
- **TI02:** Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados. La aplicación construida es el resultado de seleccionar distintas tecnologías, teniendo en cuenta, el rendimiento, coste y calidad.
- **TI03:** Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.
- **TI06:** Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.

1.4. Aportaciones

Uno de los problemas más frecuentes con los que un profesor se encuentra a la hora de impartir una asignatura suele ser la falta de tiempo para poder completar el temario que ha sido propuesto en el proyecto docente bien sea por falta de horas lectivas o por tener que enseñar un contenido extenso, esto suele provocar que el docente se vea obligado a dar por encima temas que quizá hubiera preferido explicar en mas profundidad o incluso que decida no extenderse y dejar parte del temario fuera de la lección.

Gracias a esta aplicación ese problema se verá reducido pues si bien no se puede sustituir una enseñanza presencial, un curso online impartido por el profesor encargado de la asignatura siempre será mejor que una lectura teórica del propio alumno y como ya se ha explicado anteriormente, el profesor podría impartir de formas mas completa la asignatura haciendo hincapié en aquellas áreas mas complejas y utilizando los cursos online como refuerzo.

En el apartado económico los usuarios de esta aplicación dispondrán tanto de cursos gratuitos como de pago, en los cursos gratuitos se encontrará una versión más simplificada de la temática del curso en cuestión, mientras que en los cursos de pago el alumnado encontrará explicaciones más extensas y específicas.

Los cursos no estarán exclusivamente ceñidos a las asignaturas de las distintas facultades, si bien el ámbito Universitario será una parte importante de la aplicación no estaría exclusivamente ligada a ella, la temática de los cursos incluirá una amplia variedad de contenidos diseñados para cumplir las necesidades de los usuarios teniendo también presente el ámbito profesional.

1.5. Metodologías Utilizadas

Con el objetivo de que el desarrollo de la aplicación fuese lo más profesional posible, con una alta calidad y la cantidad de funcionalidades que se deseaban implementar, así como el tiempo establecido para el desarrollo. Se hizo necesario darles orden y prioridad a las funcionalidades.

Esto tiene como objetivo crear un prototipo que cumpliera con las necesidades básicas de los usuarios. Con lo que me base en la metodología ágil Kanban para la gestión del proyecto.

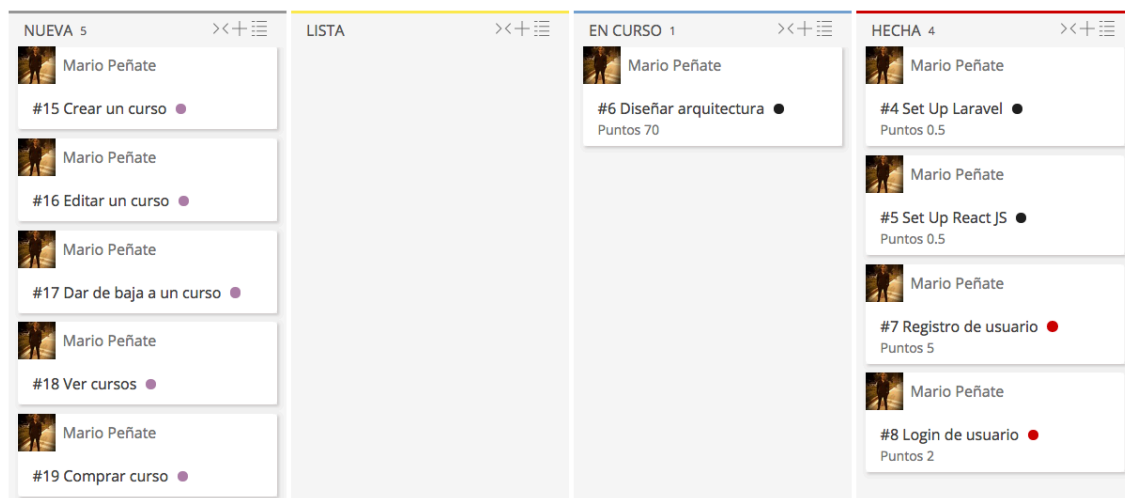


Figura 1.5.1: Ejemplo tablero Kanban

1.5.1 Kanban

Kanban⁴ es una palabra japonesa que significa algo así como “tarjetas visuales” (kan significa visual, y ban tarjeta). Esta técnica se creó en Toyota, y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción. El Kanban está dentro de la estrategia Kaizen, es decir, la mejora continua y continuada.

Las principales ventajas de esta metodología es que es muy fácil de utilizar, actualizar y asumir por parte del equipo. Además, destaca por ser una técnica de gestión de las tareas muy visual, que permite ver a golpe de vista el estado de los proyectos, así como también pautar el desarrollo del trabajo de manera efectiva.

La metodología Kanban se base en una serie de principio que la diferencian del resto de metodologías conocidas como ágiles:

⁴ <http://www.javiergarzas.com/2011/11/kanban.html>

- **Calidad garantizada:** Todo lo que se hace debe salir bien a la primera, no hay margen de error. De aquí a que en Kanban no se premie la rapidez, sino la calidad final de las tareas realizadas. Esto se base en el hecho que muchas veces cuesta más arreglarlo después que hacerlo bien a la primera.
- **Reducción del desperdicio:** Kanban se basa en hacer solamente lo justo y necesario, pero hacerlo bien. Esto supone la reducción de todo aquello que es superficial o secundario (principio YAGNI).
- **Mejora continua:** Kanban no es simplemente un método de gestión, sino también un sistema de mejora en el desarrollo de proyectos, según los objetivos a alcanzar.
- **Flexibilidad:** Lo siguiente a realizar se decide del backlog (o tareas pendientes acumuladas), pudiéndose priorizar aquellas tareas entrantes según las necesidades del momento (capacidad de dar respuesta a tareas imprevistas).

Cinco prácticas centrales del método Kanban:

- **Visualizar:** Visualizar el flujo de trabajo y hacerlo visible es la base para comprender cómo avanza el trabajo. Sin comprender el flujo de trabajo, realizar los cambios adecuados es más difícil. Una forma común de visualizar el flujo de trabajo es el uso de columnas. Las columnas representan los diferentes estados o pasos en el flujo de trabajo.
- **Limitar el trabajo en curso:** Limitar el trabajo en curso implica que un sistema de extracción se aplica en la totalidad o parte del flujo de trabajo. El sistema de extracción actúa como uno de los principales estímulos para los cambios continuos, incrementales y evolutivos en el sistema.
- **Dirigir y gestionar el flujo:** Se debe supervisar, medir y reportar el flujo de trabajo a través de cada estado. Al gestionar activamente el flujo, los cambios continuos, graduales y evolutivos del sistema pueden ser evaluados para tener efectos positivos o negativos.
- **Hacer las Políticas de Proceso Explícitas:** Configure las reglas y directrices de su trabajo. Entienda las necesidades y asegúrese de seguir las reglas. Las políticas definirán cuándo y por que una tarjeta debe pasar de una columna a otra. Escríbalas. Cambie las reglas cuando la realidad cambie.
- **Utilizar modelos para reconocer oportunidades de mejora:** Cuando los equipos tienen un entendimiento común de las teorías sobre el trabajo, el flujo de trabajo, el proceso y el riesgo, es más probable que sea capaz de construir una comprensión compartida de un problema y proponer acciones de mejora que puedan ser aprobadas por consenso. El método Kanban sugiere que un enfoque científico sea utilizado para implementar los cambios continuos, graduales y evolutivos. El método no prescribe un método científico específico para utilizarlo.

Aplicación de la metodología en el proyecto

Para aplicar la metodología en el proyecto, usé una herramienta como plataforma de gestión de proyectos ágiles basado en la metodología Kanban llamado Taiga⁵, en ella definí cuatro fases que fueron:

- **NUEVA:** En esta fase se declaran la tormenta de ideas de historias de usuarios, que pueden llegar a ser implementadas en fases posteriores.
- **LISTA:** En esta fase se escriben y se estiman las historias de usuarios que se decidieron que se van a implementar en la aplicación. En caso de que se decida seguir añadiendo funcionalidades que no estaban previstas en la primera versión de la aplicación, se cogerían de la fase anterior.
- **EN CURSO:** En esta fase se coloca la historia que se va a implementar, poniendo como límite de trabajo una sola historia por persona.
- **HECHA:** En esta última fase se colocan las historias que se han terminado. Una historia está terminada, cuando se realiza la funcionalidad esperada y pasa las pruebas definidas inicialmente.

⁵ <https://taiga.io/>

Capítulo 2

2.1. Análisis

En la actualidad, los cursos online suponen una de las herramientas más útiles tanto para aprender como para enseñar, su uso está extendido en todos los ámbitos, desde el profesional hasta el universitario o el ocio.

Para entender mejor el proyecto que debía realizar, he analizado aplicaciones similares para comprender el tipo de servicio que ofrecen y saber concretamente a que necesidades están orientadas.

2.1.1 Aplicaciones Similares



Figura 2.1.1: Logo edX

Es una plataforma de cursos abiertos masivos en línea, conocidos por sus siglas en inglés como MOOC⁶ (Massive Online Open Course). Fue fundada por el Instituto Tecnológico de Massachusetss y la Universidad de Harvard en mayo de 2012 para hospedar cursos en línea de nivel universitario de amplio rango de disciplinas, para todo el mundo sin costos para propiciar la investigación y el aprendizaje.



Figura 2.1.2: Logo de Udemy

Udemy⁷ es una Marketplace de cursos y formación online creado en 2009 que cuenta actualmente con más de sesenta mil cursos en un amplio rango de temas y quince millones de alumnos. Las características de Udemy son:

- Interfaz y estructura optimizada para facilitar el aprendizaje.

⁶ https://es.wikipedia.org/wiki/Massive_Open_Online_Course

⁷ <https://www.udemy.com/>

- Cursos en inglés y español.
- Incluye cursos gratuitos y de pago.
- Cursos organizados por distintas categorías y disciplinas.
- Puedes crear listas de deseos con cursos que te gustaría hacer.
- Sistema interno entre profesores y alumnos.

2.2. Perfiles de Usuarios

La aplicación está orientada principalmente al perfil de usuario estudiante, pero también otros perfiles como profesor o profesional de una empresa puede hacer uso de la herramienta.

A continuación, se muestra un conjunto de usuarios que podrían hacer uso de la aplicación, con el objetivo de identificar sus comportamientos y poder adaptar el producto a ellos, así como para establecer el diseño que concuerde con sus necesidades.

Para ello se usa una tabla de dummies (usuarios potenciales), describiendo cada uno de ellos.

Ocupación	Edad	Estudios	Motivación
Estudiante	22	Ingeniería Informática	Adquirir nuevos conocimientos
Profesor	50	Ingeniería Informática	Crear un curso
Estudiante	23	Telecomunicaciones	Aprobar asignatura
Profesor	47	Física	Remuneración con cursos
Empresario	50	Empresariales	Contratación de personal cualificado
Trabajador	32	Ingeniería Informática	Desarrollo profesional
Estudiante	25	Arquitectura	Formación

2.3. Requisitos

Como bien se menciona en las asignaturas de Ingeniería del software, la clave de un proyecto exitoso reside en definir correctamente y desde el inicio de este, el conjunto de requisitos y objetivos que tendrá que cumplir. Por ello, es de suma importancia realizar una fase de análisis y diseño previa a la fase de implementación, ya que de otra manera podrían ocurrir imprevistos y modificaciones que ralentizarían su desarrollo y lo encarecerían mucho más. Por tanto, este apartado refleja los resultados de las fases de adquisición de requisitos.

2.3.1 Requisitos funcionales

Se definen los requisitos funcionales como las capacidades o servicios que deberá proporcionar el sistema para alcanzar los objetivos del usuario, cómo deberá reaccionar a las entradas que se produzcan y qué comportamiento deberán tener ante situaciones particulares.

2.3.2 Requisitos no funcionales

Se definen los requisitos no funcionales como las cualidades de ejecución y evolución que debe tener el sistema, restricciones que afectan a los servicios que debe ofrecer y condiciones que ponen límites, tanto al sistema como al proceso de desarrollo.

En base a esta definición, nos encontramos con los siguientes requisitos no funcionales:

- El sistema debe funcionar para más del 90% de los navegadores.
- El software debe ser lo más modular posible con el objetivo de que sea reusable.
- El sistema debe ser eficiente a la hora de realizar consultas y hacer uso de sus recursos con el objetivo de tener un alto rendimiento (el usuario no debe esperar más de 1 segundo).
- El sistema debe desempeñar su función de una manera fluida.
- El usuario enviará los mensajes en formato JSON, muy útil para la transmisión de la información.
- El sistema debe ser estable.

2.4. Historias de Usuario

Una historia de usuario es una representación de un requisito escrita entre una y 4 líneas, utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de validación).

Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir mucho tiempo para los administradores. Las historias de usuario deben ser:

- Independientes unas de otras
- Negociables
- Pequeñas
- Verificables

2.4.1 Diseñar arquitectura

La arquitectura del sistema será una arquitectura Modelo Vista Controlador (MVC). Donde la vista será controlada con la librería de React JS usando un patrón de diseño llamado Redux y los modelos y controladores serán implementados con el framework Laravel.

2.4.2 Registro de usuario

Como usuario público quiero registrarme en la aplicación para poder acceder a los cursos.

Criterios de Aceptación:

- Cuando acceso a la página de registro de usuario.
- Relleno los campos del formulario.
- Y presiono el botón de registro.
- La aplicación debe registrarme.

Criterios de Validación:

- Dos usuarios no pueden registrarse con el mismo correo electrónico.
- El correo electrónico es requerido.
- La contraseña es requerida, debe tener como mínimo seis caracteres y debe estar codificada usando bcrypt.

2.4.3. Login de usuario

Como usuario público quiero iniciar sesión en la aplicación para poder acceder a mis cursos e información personal.

Criterios de Aceptación:

- Cuando accedo a la página de inicia de sesión.
- Relleno los campos del formulario.
- Y presiono el botón de login.
- La aplicación debe loguearme.

Criterios de Validación:

- El correo electrónico es requerido.
- La contraseña es requerida.

2.4.4. Crear un curso

Como profesor quiero crear un nuevo curso.

Criterios de Aceptación:

- Cuando inicie sesión en el sistema.
- Y de click en el botón nuevo curso.
- Debe mostrarse un formulario con los campos: nombre, descripción y precio.
- Cuando rellene el formulario y de click en el botón crear.
- Debe mostrarse un mensaje de si tuvo éxito o no.

Criterios de validación:

- Los campos serán validados y se mostrara un mensaje de error en caso de que sean inválidos.

2.4.5. Añadir cursos al carrito

Como alumno quiero poder añadir curso al carrito para poder comprarlos

Criterios de Aceptación:

- Cuando este en la sección de cursos
- Y de click en el icono del carrito
- Debe mostrarse un mensaje de que se añadió correctamente

2.5. Diagrama de Casos de Uso

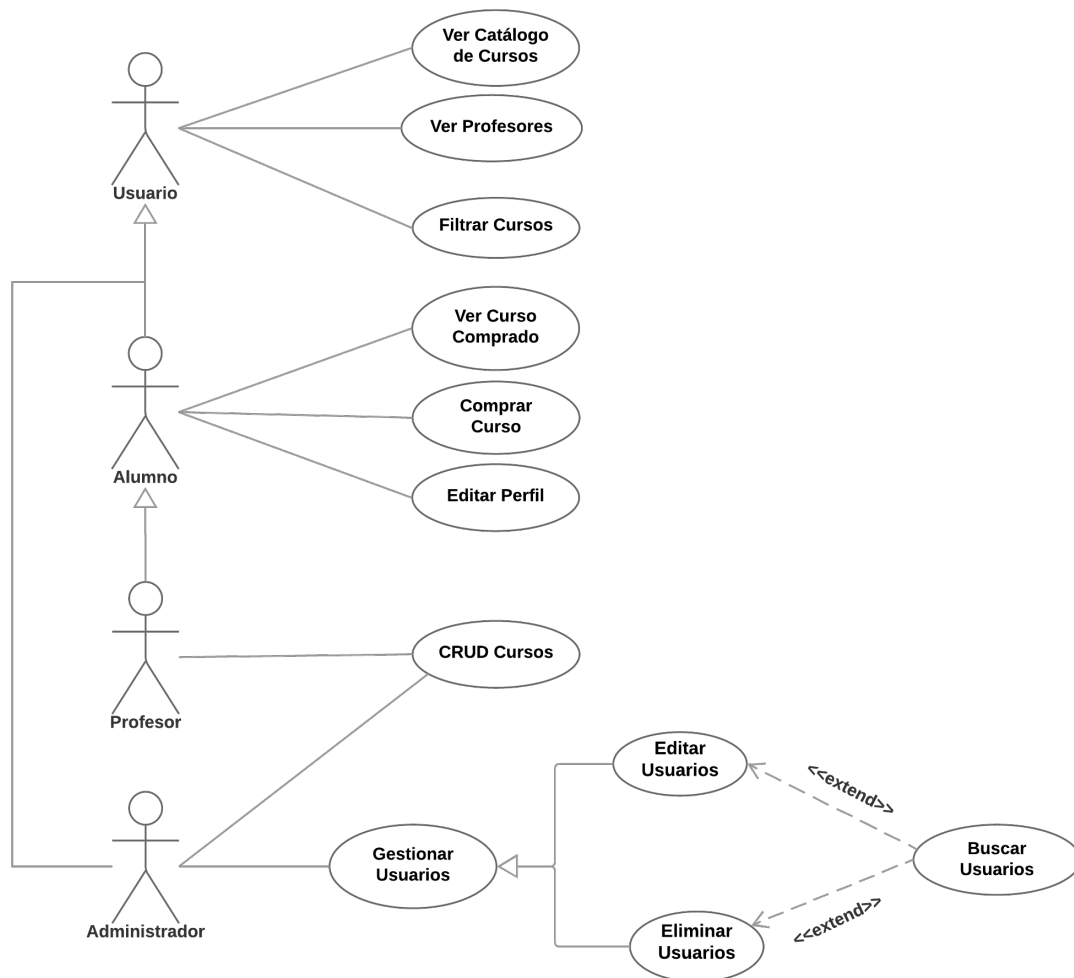


Figura 5.1.1: Diagrama de casos de uso general

En esta sección se describiré los casos de usos de mi aplicación.

- **Ver Catálogo de Cursos:** Permite a cualquier usuario (Alumno, Profesor o Administrador) poder consultar en el catálogo de los cursos disponibles.
- **Ver Profesores:** Permite a cualquier usuario (Alumno, Profesor o Administrador) consultar el listado de profesores disponibles.
- **Filtrar Cursos:** Permite a cualquier usuario (Alumno, Profesor o Administrador) poder filtrar el resultado de búsqueda de cursos.

- **Ver Curso Comprado:** Permite a un usuario autenticado poder visualizar el listado de sus cursos comprados.
- **Comprar Curso:** Permite a un usuario autenticado poder seleccionar un curso y añadirlo al carrito para posteriormente comprarlo y que se quede en su lista de cursos.

- **Editar Perfil:** Permite a un usuario autenticado poder editar su perfil de usuario y guardar dichos cambios, quedando su perfil actualizado en el sistema.
- **CRUD Cursos:** Permite a un usuario autenticado como profesor o administrador gestionar sus cursos permitiendo, añadir un nuevo curso,, ver un curso propio seleccionado, actualizar información del curso quedando en el sistema actualizado y eliminar un curso seleccionado quedando así eliminado del sistema.
- **Editar Usuarios:** Permite a un usuario autenticado como administrador seleccionar de una lista a un usuario (Alumno o Profesor) o buscarlo, para editar su información.
- **Eliminar Usuario:** Permite a un usuario autenticado como administrador eliminar un usuario seleccionado de la lista de usuarios registrados quedando así eliminado del sistema.

2.6. Mockups, diseño de la interfaz

Un mockup, mock-up, o maqueta es un modelo a escala o tamaño real de un diseño o de un dispositivo, utilizando para la demostración, evaluación del diseño, promoción y para otros fines.

Un mockup es un prototipo que proporciona una idea visual de la apariencia y funcionalidad del sistema. Los mockups son utilizados por los diseñadores principalmente para la adquisición de comentarios por parte de los usuarios. Los mockups se encuentran en el Apéndice C.

2.7. Modelo de Negocio

El modelo de negocio⁸ se refiere a la manera en que la empresa, crea, capta y ofrece valor a sus clientes. Dicho de otro modo, es la manera en la que la empresa crea valor para sí misma y para sus clientes transformando las materias primas en producto terminado y vendiendo al cliente ese producto final.

⁸ <https://innokabi.com/mi-modelo-de-negocio-el-de-siempre/>

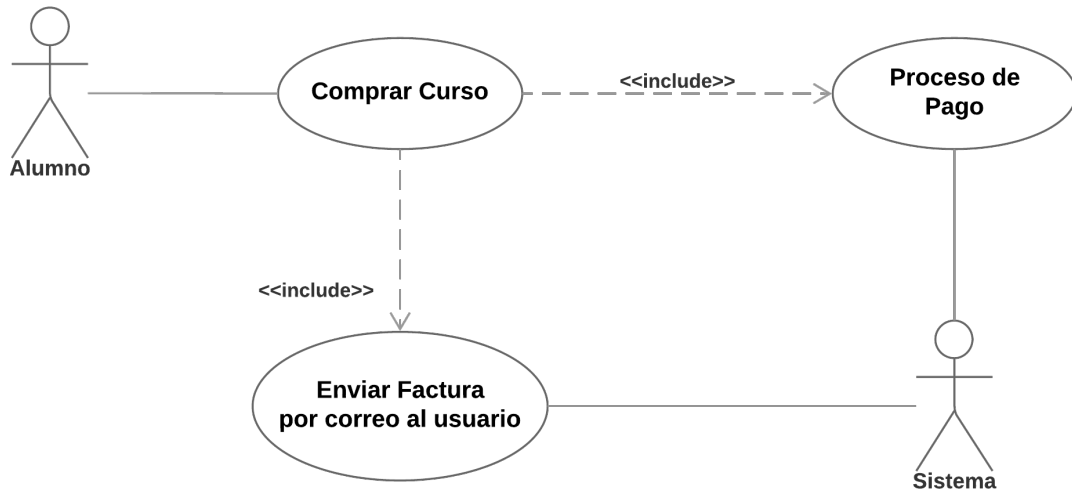


Figura 2.7.1: Modelo de Negocio

El modelo de negocio elegido para la aplicación fue el **Freemium**⁹, donde ofrecemos tanto cursos gratuitos como cursos de pago. La ventaja de este modelo reside en ofrecer un servicio gratuito para atraer a más estudiantes a la plataforma y luego que paguen por cursos más avanzados.

⁹ <https://es.wikipedia.org/wiki/Freemium>

2.8. Normativa y Legislación

El presente trabajo manipula datos personales de nivel básico, como son el correo electrónico, contraseña, nombre y apellido. En el caso de la compra de cursos, la aplicación no maneja ningún dato bancario ya que se redirecciona al usuario a la pasarela de pagos de Stripe, y ésta es la que maneja este tipo de datos.

Las principales normativas que afectan a este trabajo son La Ley Orgánica de Protección de Datos (LOPD) y el Real Decreto 994/199 de Medidas de Seguridad de ficheros automatizados. La primera tiene como objetivo garantizar y proteger, los datos personales, libertades públicas y derechos fundamentales de personas físicas, sobre todo su honor e integridad personal y familiar. Mientras que la segunda es un reglamento desarrollado por la LORTAD que surge con el objetivo de regular las medidas técnicas y organizativas cuando se tratan datos personales de manera automatizada.

Entre las medidas de seguridad de nivel básico adoptadas se encuentran:

- **Control de Acceso:** Los usuarios tendrán acceso solo a aquellos recursos que precisen para el desarrollo de sus funciones.
- **Identificación y autenticación:** Los usuarios se identificarán mediante un correo electrónico y una contraseña que se almacenará usando una herramienta de función hash criptográfico llamada bcrypt¹⁰. Además, se limitó el inicio de sesión a sesenta intentos por minuto, para evitar posibles ataques de fuerza bruta.

¹⁰ <https://en.wikipedia.org/wiki/Bcrypt>

Capítulo 3

3.1. Desarrollo

3.1.1 Características Implementadas

A lo largo del proyecto se fueron escribiendo más de treinta historias de usuario en orden de prioridad. Se implementaron el 80% de las historias escritas que se encuentran entre el apartado de Historias de Usuario del capítulo 2.3 y el Apéndice A.

A continuación, se muestran las historias que no fueron implementadas:

- Mandar mensajes
- Cancelar cuenta
- Ver mi perfil de usuario
- Activar suscripción
- Desactivar suscripción
- Ordenar las secciones dentro de un curso
- Ordenar los recursos dentro de un curso

3.2. Diagrama de Arquitectura del Sistema

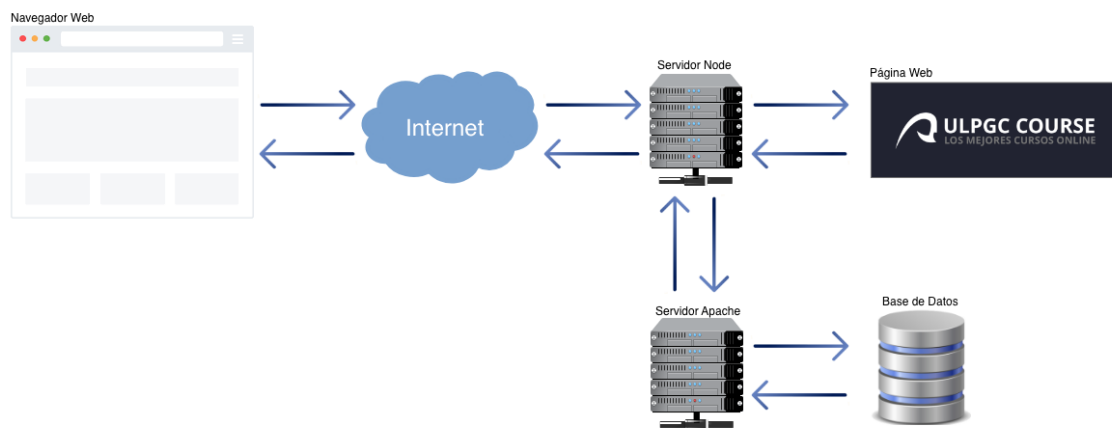


Figura 3.2.1: Diagrama de la Arquitectura del Sistema

El objetivo principal de este diagrama es ofrecer una visión simplificada del sistema, la cual nos muestra las interfaces cliente-servidor y las transacciones que en ellas se realizan. En la figura 3.2.1 se puede observar como está definida la arquitectura del sistema en la aplicación siendo esta una arquitectura Cliente-Servidor, donde se muestran las personas que pueden acceder a la aplicación mediante un cliente, en este caso un navegador web haciendo una petición (Request). Este a su vez accede al servidor a través de Internet, el servidor le envía la petición a la aplicación, esta consulta la base de datos y devuelve una respuesta (Response) de tipo JSON al servidor para que se la entregue al cliente.

3.3. Arquitectura de Software

3.3.1 Redux

Redux¹¹ es un contenedor predecible del estado de aplicaciones en Javascript. Te ayuda a escribir aplicaciones que se comportan de manera consistente, corren en distintos ambientes (cliente, servidor y nativo), y son fáciles de probar. Se encarga de emitir actualizaciones de estado en respuesta a acciones. En lugar de modificar directamente el estado (React JS), la modificación se maneja a través de objetos sencillos llamados acciones. Luego escribes una función pura¹² llamada `reducer` para decidir cómo cada acción transforma el estado de toda la aplicación. Otra característica es que todo se maneja desde un solo lugar llamado `store`.

En Redux existen tres principios que son:

Única fuente de verdad:

El estado de toda la aplicación se almacena en una estructura de árbol dentro de un único `store`.

Esto hace que sea fácil crear aplicaciones universales, ya que el estado en el servidor puede ser serializado e hidratado en el cliente sin ningún esfuerzo adicional de codificación. Un solo árbol de estado también facilita la depuración o la introspección de una aplicación, esto permite que el ciclo de desarrollo sea más rápido.

El estado es read-only

La única manera de cambiar el estado es emitir una acción, un objeto que describe lo sucedido.

Esto garantiza que nada modificará directamente al estado de la aplicación. Por el contrario, la acción expresa una intención de transformar el estado. Debido a que todos los cambios están centralizados y suceden uno por uno en un orden estricto, no hay **race conditions**¹³ que necesitemos monitorear. Dado que las acciones son objetos sencillos, se pueden registrar, serializar, almacenar y reproducir posteriormente para fines de depuración o pruebas.

Los cambios se realizan a través de funciones puras

Para especificar cómo se transforma el estado por acciones, se escriben `reducers`.

Los `reducers` son funciones puras que toman el estado anterior y una acción, y devuelven el siguiente estado. Esta función debe devolver objetos que presentan el nuevo estado, en lugar de cambiar el estado anterior.

¹¹ <https://es.redux.js.org/>

¹² <https://medium.com/entendiendo-javascript/funciones-puras-en-javascript-crea-funciones-libres-c2b9b3785418>

¹³ https://en.wikipedia.org/wiki/Race_condition

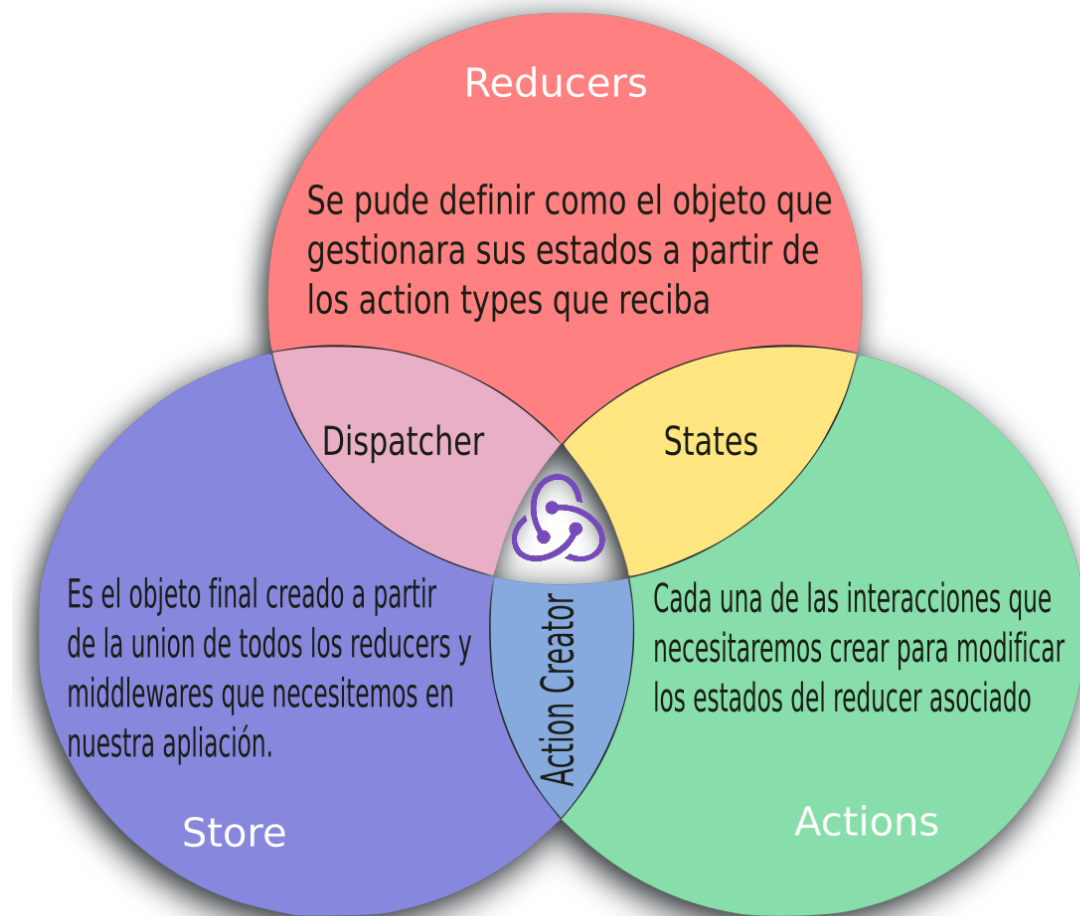
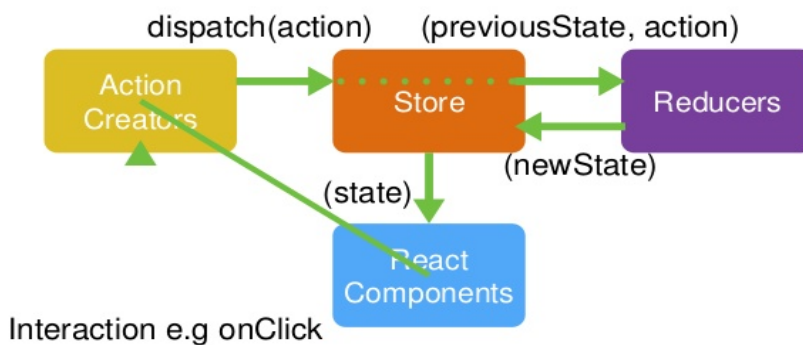


Figura 3.3.1.a: Redux

Redux Flow



React + Redux

@nikgraf

Figura 3.3.1.b: Redux Flow

3.4. Diseño e Implementación

La aplicación se concibió con un diseño separado por capas bien definidas, donde cada capa tiene una responsabilidad específica, de tal forma que sea fácil reutilizar componentes y cambiarlos por otros.

Para la parte del cliente he escogido la librería React JS¹⁴ por ser una librería enfocada a las interfaces de usuarios. Con React podemos crear aplicaciones web tanto del lado del cliente como del servidor, siendo esto una de las principales características de la librería. Con React podemos desarrollar aplicaciones web de una manera más ordenada y con menos código que si usas Javascript puro o librerías como JQuery centradas en la manipulación del DOM.

Sin embargo, la API esta implementada con el framework Laravel¹⁵ por ser muy versátil, por su elegancia y simpleza a la hora de escribir código PHP, pero también por su potencia, ya que trae muchos componentes de ayuda como los validadores, ORM, migraciones, motor de plantillas, etc. Esto hace que solo te centres en escribir tu aplicación, en vez de estar preocupándote en reinventar la rueda. También está hecho siguiendo las mejores prácticas de programación, por lo que te hace escribir código de buena calidad.

3.4.1 Diagrama Entidad Relación

El diagrama entidad relación (DER) se hizo después que se hubiese analizado el modelo de negocio de la aplicación. Para ello se usó la herramienta DbSchema¹⁶ y el diagrama se puede ver en la figura 3.4.1.a, figura 3.4.1.b y figura 3.4.1.c:

¹⁴ <https://reactjs.org/>

¹⁵ <https://laravel.com/>

¹⁶ <https://www.dbschema.com/postgresql-designer-tool.html>

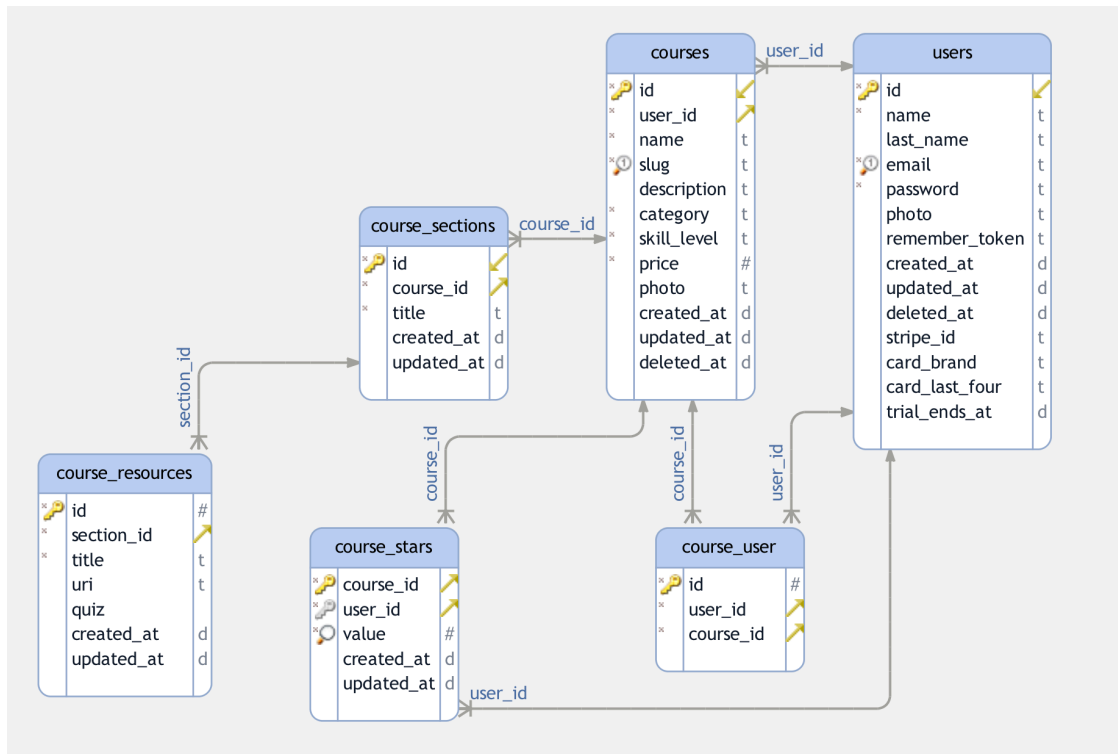


Figura 3.4.1.a: Diagrama Entidad Relación

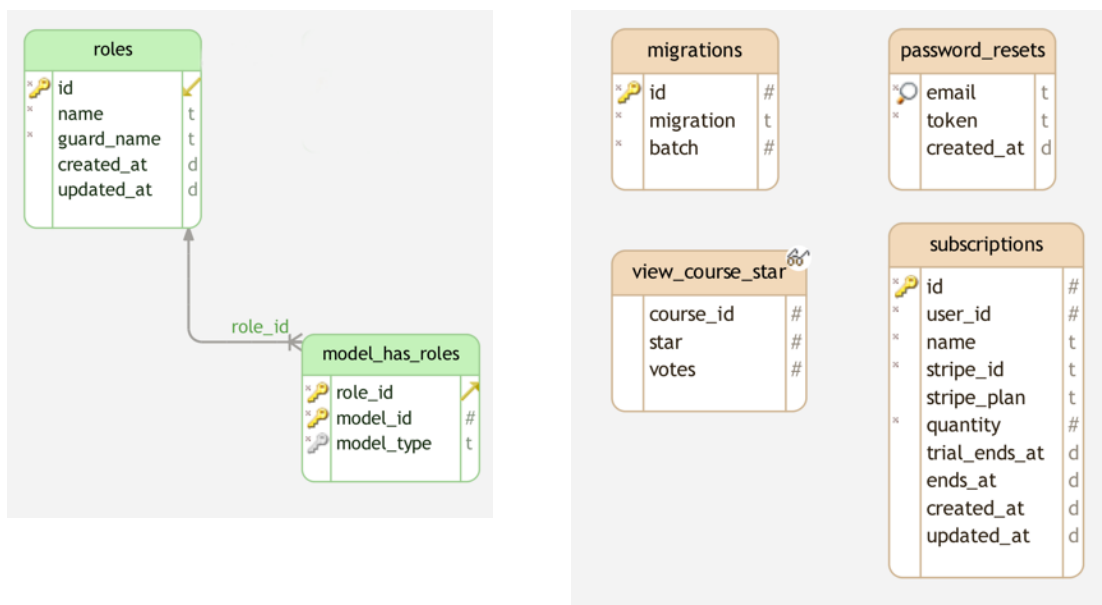


Figura 3.4.1.b: Diagrama Entidad Relación

A continuación, se explican para que se usó cada tabla:

- **users:** Esta tabla guarda los usuarios registrados, así como información referente a los mismos.
- **courses:** Esta tabla guarda los cursos que han sido creados por los profesores. Esta tiene una relación de muchos a uno con la tabla users.
- **course_sections:** Esta tabla guarda las secciones de un curso. Esta tiene una relación muchos a uno con la tabla courses.

- **course_resources:** Esta tabla guarda los recursos que se añaden a las secciones. Esta tiene una relación muchos a uno con la tabla course_sections.
- **course_starts:** Esta tabla guarda las votaciones de los usuarios en los cursos. Esta es una tabla pivote entre usuarios y cursos.
- **course_user:** Esta tabla guarda los cursos que ha comprado un usuario. Esta es una tabla pivote entre usuarios y cursos.
- **roles:** Esta tabla guarda los roles de la aplicación.
- **model_has_roles:** Esta tabla asocia un rol a un usuario. Esta tiene una relación muchos a uno con la tabla roles.
- **migrations:** Esta tabla guarda los nombres de las tablas que han sido creadas.
- **view_course_start:** Esta no es una tabla sino una vista que almacena información sobre el promedio de los votos de un curso, así como cuantos han votado.
- **password_resets:** Esta tabla los usuarios que desean restablecer la contraseña.
- **subscription:** Esta tabla guarda las suscripciones de los usuarios.

3.4.2 Diseño de Interfaz de Usuario

Se usó el framework Bootstrap¹⁷ para maquetar y diseñar la interfaz de usuario, de acuerdo con la idea que se tenía en el análisis de la aplicación.

Bootstrap nos permite simplificar el proceso de maquetación, sirviéndonos de guía para aplicar las buenas prácticas y los diferentes estándares. Como pueden ser:

- Tener una web bien organizada de forma visual.
- Permite utilizar muchos elementos web: desde iconos a despletables.
- Diseño Responsive.

3.4.3 Arquitectura de Laravel

Todos los proyectos de Laravel 5.5 tienen la siguiente estructura de directorios:

- app/
- bootstrap/
- config/
- database/
- public/
- resources/
- routes/
- storage/
- test/
- vendor/
- .env
- .env.example
- gitattributes
- .gitignore
- artisan

¹⁷ <https://getbootstrap.com/>

- composer.json
- composer.lock
- package.json
- phpunit.xml
- readme.md
- server.php
- webpack.mix.js

Directorio app

App es usado para ofrecer un hogar por defecto a todo el código personal del proyecto. Eso incluye clases que pueden ofrecer funcionalidad a la aplicación, archivos de configuración y más. Es considerado el directorio más importante de nuestro proyecto.

El directorio app tiene a su vez otros subdirectorios importantes, pero uno de los más utilizados es el directorio Http en el cuál ubicaremos nuestros **Controllers**, **Middlewares** y **Request** en sus carpetas correspondientes.

Directorio config

La configuración tanto para el framework como para nuestra aplicación. Los archivos más usados del directorio config son:

- **app.php:** En este archivo nos puede interesar configurar el lenguaje de nuestra aplicación, la zona horaria, los providers y alias de las clases más comunes.
- **database.php:** En este archivo podemos configurar principalmente el motor de base de datos al cuál deseamos conectarnos.

Directorio database

Aquí se encontrarán los archivos relacionados con el manejo de la base de datos. Dentro de este directorio se encuentran los subdirectorios:

- **factories:** Aquí escribiremos nuestro model factories.
- **migrations:** Todas las migraciones que creamos se ubican en este subdirectorio.
- **seeds:** Contiene todas las clases de tipo seed.

Directorio public

Dentro de este directorio colocaremos todos los recursos estáticos de nuestra aplicación, es decir, archivos css, js, imágenes y fuentes.

Directorio resources

Dentro de este directorio se encuentran los subdirectorios:

- **assets:** Aquí se ubican todos los archivos sass de nuestra aplicación.
- **lang:** Aquí se encuentran todos los archivos de internacionalización, es decir, los archivos para poder pasar nuestro proyecto de un idioma a otro.
- **views:** Aquí ubicaremos nuestras vistas en formato php.blade.

Directorio routes

Aquí se encuentran todas las rutas o endpoints de nuestra aplicación.

Directorio storage

Cuando Laravel necesita escribir algo en el disco, lo hace en el directorio storage. Por este motivo tu servidor web debe poder escribir en esta ubicación. Aquí podemos encontrar otros directorios entre los cuales el más relevante es el subdirectorio framework, es ahí donde se almacena el cache y las vistas compiladas.

Directorio test

Aquí escribiremos los archivos de pruebas que serán ejecutadas posteriormente por phpunit.

El archivo .env y .env.example

El archivo .env no existe cuando instalamos Laravel, en este archivo se configurará el modo en que se ejecuta la aplicación, por defecto será en modo debug, además podemos configurar la conexión a la base de datos y la conexión con el servidor de correo electrónico. El archivo .env lo creamos copiando el archivo .env.example y renombrado la copia como .env.

Por motivos de seguridad de la base de datos el archivo .env nunca se sube cuando hacemos un push en nuestro repositorio. Es por eso que aparece escrito dentro del archivo .gitignore en la raíz de nuestro proyecto.

3.4.4 Arquitectura React JS

React JS no fuerza a tener ninguna estructura de proyecto en particular. Lo bueno de ello es que te permite tener la estructura que mejor se adapte a lo que necesites. Lo malo es que no es posible tener una estructura ideal que funcione en todos los proyectos. Por tanto, voy a explicar como está esta estructurado el proyecto con React, es una estructura plana como la siguiente:

- public/
- src/
- node_modules/
- .gitignore
- README.md
- gulpfile.js
- package.json
- yarn.lock

Directorio public

La raíz de nuestro servidor donde se podrá encontrar el index.html, el archivo principal y el favicon.ico, el icono de la aplicación.

Directorio src

Aquí es donde trabajaremos nuestro proyecto, donde vamos a colocar los archivos de nuestros componentes de React.

En este directorio existen varios subdirectorios que son:

- **actions:** Aquí guardaremos todas las acciones de redux.
- **assets:** Guardamos todos los archivos sass para compilarlos con gulp y generar un único archivo css.
- **commons:** Aquí guardamos todos los archivos que van hacer compartido por la aplicación como pueden ser constantes, helpers, etc.
- **components:** En esta carpeta guardaremos todos los componentes de la aplicación.
- **containers:** Aquí guardaremos los componentes de orden superior que engloban a otros componentes.
- **reducers:** Aquí se ubican todos los archivos que van a estar relacionado con la manipulación de estado de redux.
- **store:** Contiene un único archivo que se encarga de crear el árbol de estados de redux.

Directorio node module

En este directorio se guardan las dependencias una vez descargadas por el gestor de paquetes de node.

3.5. Tecnologías del lado del servidor

Para el desarrollo de la aplicación se usan varias tecnologías que se ejecutan en el servidor. A continuación, se detallan.

3.5.1 Servidor Web Apache

Apache¹⁸ es un servidor web HTTP de código abierto, especialmente diseñado para transferir datos de hipertexto, es decir, páginas webs con todos sus elementos, utilizando el protocolo HTTP

Es multiplataforma, por lo que corre en sistemas tipo Linux, MacOS y Windows.

3.5.2 Lenguaje PHP 7.2.6

PHP¹⁹ es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenidos dinámicos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la

¹⁸ <https://www.apache.org/>

¹⁹ <https://secure.php.net/manual/es/intro-what-is.php>

página web resultante. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

3.5.3 Framework Laravel 5.5

Laravel²⁰ es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el código espagueti.

Tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

3.5.4 Base de Datos Postgres SQL

PostgreSQL²¹ es un sistema de gestión de bases de datos relación orientado a objetos y libre, publicado bajo la licencia de PostgreSQL, similar a BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, libre o apoyados por organizaciones comerciales.

3.5.5 Composer

Composer²² es una herramienta para gestionar las dependencias de las aplicaciones PHP. Una vez declaradas las librerías de las que depende tu proyecto, Composer es capaz de descargar e instalar automáticamente las versiones correctas de cada una de estas librerías. En el proyecto es usado para gestionar las dependencias de PHP.

3.5.5 Amazon S3

Amazon S3 es un servicio de almacenamiento de objetos creado para almacenar y recupera cualquier volumen de datos desde cualquier ubicación de Internet. Es un servicio de almacenamiento simple que ofrece una infraestructura para almacenar datos con un nivel extremadamente alto de durabilidad, disponibilidad y escalabilidad a un costo muy bajo.

²⁰ <https://laravel.com/>

²¹ <https://www.postgresql.org/>

²² <https://getcomposer.org/>

También proporciona una sencilla interfaz de servicios web que pueden utilizar para almacenar y recupera el volumen de datos que desee, cuando lo necesite y desde cualquier parte de la web. Mediante este servicio web, puede crear de forma sencilla aplicaciones que utilicen almacenamiento de Internet. Como Amazon S3 es un servicio con un alto nivel de escalabilidad y se paga por lo que se usa.

3.5.6 Mailtrap

Es un servicio que podemos utilizar para realizar nuestras pruebas de envío de correos, esto es debido a que es un servidor SMTP falso para que los equipos de desarrollo puedan probar, ver y compartir correos electrónicos enviados desde los entornos de desarrollo y sin clientes reales.

3.6 Tecnologías del lado del cliente

Para el desarrollo de la aplicación se usaron varias tecnologías que se ejecutan en el cliente. A continuación, se detallan.

3.6.1 Lenguaje HTML 5.2

HTML 5.2 (HyperText Markup Language) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML 5.2 especifica dos variantes de sintaxis para HTML: una clásica, HTML (text/html), conocida como HTML 5.2, y una variante XHTML conocida como sintaxis XHTML 5 que deberá servirse como sintaxis XML (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en diciembre de 2017.

3.6.2 Lenguaje CSS3

CSS es un lenguaje de hojas de estilo para definir y crear presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML, el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etc. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

3.6.3 Sass

Sass (acrónimo de Syntactically Awesome StyleSheets) es una extensión de CSS que añade características muy potentes y elegantes a este lenguaje de estilos. Sass permite el uso de variables, reglas CSS anidadas, mixins, importación de hojas de estilos y muchas otras características, al tiempo que se mantiene la compatibilidad con CSS.

También permite organizar mejor las hojas de estilos grandes y permite ser mucho más productivo con las hojas de estilos pequeñas sobre todo gracias a la librería Compass.

3.6.4 Framework Bootstrap 4

Bootstrap es un framework o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript.

3.6.5 Lenguaje JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

3.6.6 Jquery

Jquery es una biblioteca de JavaScript de código abierto, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con llamadas AJAX.

3.6.7 React JS

React JS es una librería JavaScript de Código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página. Es mantenido por Facebook, Instagram y una comunidad de desarrolladores independientes.

3.6.8 Redux

Redux es un contenedor predecible del estado de aplicaciones JavaScript. Te ayuda a escribir aplicaciones que se comportan de manera consistente, corren en distintos ambientes (cliente, servidor y nativo), y son fáciles de probar.

Además de eso, provee una gran experiencia de desarrollo, gracias a la edición en vivo combinado con un depurador sobre una línea de tiempo (Redux DevTools).

3.6.9 Stripe JS

Es un servicio de un proveedor de servicios de aplicación de comercio electrónico, con el que se autorizan pagos a negocios electrónicos (en línea).

3.6.10 Font Awesome 5

Font Awesome es una fuente de iconos mediante la que se pueden insertar algunos iconos sociales y símbolos en lugar de imágenes, con lo que la página es más ligera.

3.6.11 Axios

Axios es una Librería de JavaScript construida con el objetivo de gestionar la programación asíncrona con promesas.

3.6.12 Google Maps

Google Maps es un servidor de aplicaciones de mapas en la web que pertenece a Google. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo.

3.7. Herramientas de Desarrollo

Para poder hacer uso de las tecnologías mencionadas anteriormente, se hizo necesario utilizar herramientas que aumentaran la productividad de trabajo. A continuación, se detallan las herramientas utilizadas.

3.7.1 Sistema Operativo Mac OS X

Es el sistema operativo principal para la familia de computadoras Mac de Apple. Dentro del mercado de computadores de escritorio y portátiles, es el segundo sistema operativo de escritorio más utilizado, después de Microsoft Windows.

3.7.2 PhpStorm

PhpStorm es un IDE de licencia privada de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad. Proporciona un editor para PHP, HTML y JavaScript con análisis de código en la marcha, prevención de errores y refactorizaciones automatizadas para PHP y código JavaScript.

3.7.3 Github

Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

3.7.4 Taiga

Taiga es un software open source de gestión de proyectos que se puede usar tanto online como en un servidor local. Es intuitivo, sencillo y potente. Gracias a él, podrás gestionar proyectos de forma ágil y accesible, asignando tareas, estableciendo objetivos y disponer de un papel de reporte de bugs y eventos. Es una aplicación increíblemente versátil que se puede adaptar a cualquier necesidad de gestión y control del avance del proyecto.

3.7.5 Google Chrome

Google Chrome es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto, como el motor de renderizado Blink. Está disponible gratuitamente bajo condiciones específicas del software privativo o cerrado.

3.7.6 Postman

Es una extensión del navegador Google Chrome, que permite el envío de peticiones HTTP REST sin necesidad de desarrollar un cliente.

3.7.7 Adobe XD

Es una aplicación de diseño vectorial pensada para diseñadores. Tiene algunas herramientas de diseño vectorial bastante buenas, además permite crear prototipos tanto para diseños web como para diseños móviles.

3.7.8 Unsplash

Unsplash es una plataforma que permite a los fotógrafos publicar sus fotos licenciadas con Creative Commons Zero. Es totalmente gratuito y nos permite compartir fotos libres de derechos y en alta resolución.

3.7.9 Heroku

Heroku es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación.

Heroku es propiedad de Salesforce.com y es una de las primeras plataformas de computación en la nube, que fue desarrollada en junio de 2007 por James Lindenbaum, Adam Wiggins y Orion Henry.

Podemos utilizar Heroku para almacenar nuestras aplicaciones, tanto de forma gratuita (limitado dado que los servidores se apagan a la media hora de inactividad) como con planes de pago.

Aparte de esto Heroku tiene unos recursos llamados add-ons, gracias a ellos podemos añadir funcionalidad extra a nuestras aplicaciones creadas.

Para el Trabajo de Fin de Grado he desplegado la aplicación en Heroku, para facilitar el uso de la aplicación sin tener que realizar ningún set-up por parte del tribunal. Pienso que es imprescindible que puedan probar la aplicación y que no ocurra como en otras defensas que no han podido levantar un servidor para poder probarlo.

Para mi es bastante importante porque de este modo puedo obtener un mayor feedback y mejorar ciertos aspectos que probablemente se realicen en futuras versiones.

A continuación, adjunto las direcciones web:

<https://ulpgc-course.herokuapp.com/>

<https://ulpgc-course-api-test.herokuapp.com/>

Capítulo 4

Conclusiones

El desarrollo de este proyecto me ha permitido poder entrar en contacto con distintas tecnologías del desarrollo web, así como conocer de primera mano el proceso que implica el desarrollo de una aplicación de esta magnitud teniendo que analizar sus requisitos, tecnologías similares y necesidades que cubre.

Los cursos online son una de las aplicaciones mas utilizadas en docencia y poco a poco han ido creciendo para abarcar completamente el entorno laboral permitiendo la posibilidad de que personas de distintos ámbitos profesionales puedan compartir sus conocimientos tanto de forma gratuita como remunerada.

Para la realización de este proyecto me han sido indispensable poner en práctica los conocimientos que he ido adquiriendo a lo largo de la carrera, sobretodo de la rama de especialización de Tecnologías de la Información.

A nivel personal estoy agradecido de haber podido desarrollar esta aplicación, la cual me ha permitido aplicar conocimientos de programación web, de tal forma que me permite seguir creciendo como programador.

4.1 Trabajos Futuros

El trabajo futuro dependerá si se pone en marcha esta aplicación que está centrada en los estudiantes. Seria interesante tener una plataforma como esta que ofrezca cursos online para aquellas personas que quieran adquirir nuevos conocimientos o mejorar los conocimientos existentes.

El proyecto tiene historias de usuario que no están desarrolladas, por lo que eso sería un trabajo de futuro, como nuevas implementaciones que vayan surgiendo a partir del feedback de los usuarios.

Bibliografía

Recursos Digitales:

Documentación oficial de heroku. URL <https://www.heroku.com/>

Documentación oficial de laravel. URL <https://laravel.com/docs/5.1>.

Documentación oficial de react. URL <https://reactjs.org/>

Documentación oficial de redux. URL <https://es.redux.js.org/>

Libros

Christopher Pecoraro. (2015). *Mastering Laravel*. Packt Publishin, 1ª edición.

David J. Anderson. (2011). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

David J. Anderson y Andy Carmichael. (2017). *Kanban Escencial Condensado*. Lean - Kanban University.

Gregor Hohpe y Bobby Wool. (2012). *Enterprise Integration Patterns*. Addison-Wesley, 1ª edición.

Josh Lockhart. (2015). *Modern PHP*. O'Reilly, 1ª edición.

Kristopher Wilson. (2016). *The Clean Architecture in PHP*. Leanpub, 1ª edición.

Martin Fowler, (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley, 1ª edición.

Matt Zandstra. (2013). *PHP Objects, Patterns, and Practice*. Apress, 4ª edición.

Robin Nixon. (2014). *Learning PHP, MySQL and JavaScript*. O'Reilly, 4ª edición.

William Sanders. (2013). *Learning PHP Design Patterns*. O'Reilly, 1ª edición.

Apéndice A

Anexo Historias de Usuario

A.0.1 Registro de usuario

Como usuario público quiero registrarme en la aplicación para poder acceder a los cursos.

Criterios de Aceptación:

- Cuando acceso a la página de registro de usuario.
- Relleno los campos del formulario.
- Y presiono el botón de registro.
- La aplicación debe registrarme.

Criterios de Validación:

- Dos usuarios no pueden registrarse con el mismo correo electrónico.
- El correo electrónico es requerido.
- La contraseña es requerida, debe tener como mínimo seis caracteres y debe estar codificada usando bcrypt.

A.0.2 Login de usuario

Como usuario público quiero iniciar sesión en la aplicación para poder acceder a mis cursos e información personal.

Criterios de Aceptación:

- Cuando accedo a la página de inicia de sesión.
- Relleno los campos del formulario.
- Y presiono el botón de login.
- La aplicación debe loguearme.

Criterios de Validación:

- El correo electrónico es requerido.
- La contraseña es requerida.

A.0.3 Crear un curso

Como profesor quiero crear un nuevo curso.

Criterios de Aceptación:

- Cuando inicie sesión en el sistema.
- Y de click en el botón nuevo curso.
- Debe mostrarse un formulario con los campos: nombre, descripción y precio.
- Cuando rellene el formulario y de click en el botón crear.
- Debe mostrarse un mensaje de si tuvo éxito o no.

Criterios de Validación:

- Los campos serán validados y se mostrara un mensaje de error en caso de que sean inválidos.

A.0.4 Editar un curso

Como profesor quiero editar mi curso.

Criterios de Aceptación:

- Una vez creado un curso.
- Y de click en el botón editar curso.
- Debe mostrarse un formulario relleno con la información del curso.
- Cuando modifique algún campo y pulse en el botón actualizar.
- Debe mostrarse un mensaje de si tuvo éxito o no.

Criterios de Validación:

- Los campos serán validados y se mostrara un mensaje de error en caso de que sean inválidos.
- El curso debe existir en la aplicación.

A.0.5 Dar de baja a un curso

Como profesor quiero dar de baja a un curso mío.

Criterios de Aceptación:

- Y de click en el botón eliminar curso.
- Debe mostrarse un dialogo de confirmación para poder dar de baja al curso.

Criterios de Validación:

- El curso debe existir en la aplicación.

A.0.6 Ver cursos

Como usuario registrado quiero poder ver cursos.

Criterios de Aceptación:

- Presionar en la imagen del curso.

Criterios de Validación:

- El curso debe existir en la aplicación.

A.0.7 Añadir curso al carrito

Como usuario quiero poder añadir cursos al carrito.

Criterios de Aceptación:

- Presionar en el icono de añadir al carrito.

Criterios de Validación:

- El curso debe existir en la aplicación.

A.0.8 Eliminar curso del carrito

Como usuario quiero poder eliminar un curso del carrito.

Criterios de Aceptación:

- Presionar en el botón que tiene una X.

Criterios de Validación:

- El curso debe estar en el carrito para poder eliminarlo.

A.0.9 Comprar curso

Como usuario registrado quiero poder pagar los cursos para poder consumirlos.

Criterios de Aceptación:

- Presionar en el botón de realizar pago.
- Rellenar el formulario de la pasarela de pagos de Stripe.

Criterios de Validación:

- La pasarela de pagos validará los campos y en caso de error se le mostrará al usuario.
- El carrito no puede estar vacío.

A.0.10 Votar un curso

Como usuario registrado quiero poder votar un curso para que el profesor pueda obtener un feedback.

Criterios de Aceptación:

- Visitar un curso.
- Y pinchar donde salen las estrellas amarillas.

Criterios de Validación:

- Se debe tener acceso al curso.

A.0.11 Buscar curso

Como usuario quiero poder buscar cursos para ver el catálogo de la web.

Criterios de Aceptación:

- Escribir en el input de tipo texto.
- Presionar enter.

A.0.12 Filtrar cursos por categoría

Como usuario quiero poder filtrar los cursos para poder acceder más rápido.

Criterios de Aceptación:

- Ir a la sección de cursos.
- Pinchar en algún filtro de alguna categoría.
- Presionar enter.

A.0.13 Filtrar cursos por dificultad

Como usuario quiero poder filtrar los cursos para poder acceder más rápido.

Criterios de Aceptación:

- Ir a la sección de cursos.
- Pinchar en algún filtro de dificultad.
- Presionar enter.

A.0.14 Ver mis cursos comprados

Como usuario registrado quiero poder ver mis cursos comprados para verlos de nuevo.

Criterios de Aceptación:

- Presionar en el botón mis cursos.

A.0.15 Cambiar contraseña

Como usuario registrado quiero cambiar la contraseña.

Criterios de Aceptación:

- Cuando inicie sesión en el sistema.
- Y de click en el botón cambiar contraseña.
- Debe mostrarse un formulario que deberemos rellenar.
- Y presionamos el botón actualizar.

Criterios de Validación:

- Los campos serán validados y se mostrara un mensaje de error en caso de que sean inválidos.
- Los campos contraseña nueva y repetir contraseña deben coincidir.

A.0.16 Enviar correo al administrador

Como usuario quiero mandar un correo al administrador para notificar algún tipo de error o simplemente preguntar algún tipo de duda.

Criterios de Aceptación:

- Presionamos en el menú superior donde pone universidad@ulpgccourse.es
- Y se abrirá una ventana donde podemos escribir nuestro correo.

A.0.17 Llamar por teléfono

Como usuario quiero poder llamar por teléfono para notificar algún tipo de error o simplemente preguntar algún tipo de duda.

Criterios de Aceptación:

- Presionamos en el menú superior donde sale el número de teléfono.

A.0.18 Listar usuarios

Como administrador quiero listar todos los usuarios

Criterios de Aceptación:

- Presionar en el botón usuarios.

A.0.19 Editar usuario

Como administrador quiero editar un usuario

Criterios de Aceptación:

- Presionar en el botón usuarios.
- Presionar en el botón editar usuario.
- Actualizar el formulario.
- Y click en actualizar.

Criterios de Validación:

- El usuario debe existir.

A.0.20 Dar de baja a un usuario

Como administrador quiero dar de baja a un usuario

Criterios de Aceptación:

- Presionar en el botón usuarios.
- Presionar en el botón editar usuario.
- Y click en el icono de eliminar usuario.

Criterios de Validación:

- El usuario debe existir.

A.0.21 Buscar usuario

Como administrador quiero poder cambiar buscar a un usuario.

Criterios de Aceptación:

- Escribimos en el input de tipo texto.
- Y presionamos en el botón buscar.

Criterios de Validación:

- El usuario debe existir.

A.0.22 Cancelar Cuenta

Como usuario registrado quiero darme de baja en el sistema porque no quiero seguir usando la aplicación.

Criterios de Aceptación:

- Presiono en el botón mi perfil.
- Y doy click en el botón cancelar cuenta.
- Y confirmo que deseo darme de baja.

Criterios de Validación:

- El usuario debe existir.

A.0.23 Mandar mensajes

Como usuario registrado quiero mandar mensajes a otros usuarios para poder entablar una conversación y preguntar dudas.

Criterios de Aceptación:

- Nos dirigimos a la sección de mensajes.
- Pulsamos en el botón nuevo.
- Rellenamos el formulario.
- Y presionamos en el botón enviar.

Criterios de Validación:

- El usuario debe existir.

A.0.24 Ver mí perfil de usuario

Como usuario registrado quiero poder ver mi perfil de usuario.

Criterios de Aceptación:

- Presionamos en el botón ir a mi perfil.

Criterios de Validación:

- El usuario debe existir.

A.0.25 Editar mí perfil de usuario

Como usuario registrado quiero poder editar mi perfil de usuario.

Criterios de Aceptación:

- Presionamos en el botón ir a mi perfil.
- Pulsamos en el botón editar.
- Rellenamos el formulario.
- Y click en actualizar.

Criterios de Validación:

- El usuario debe existir.

A.0.26 Listar profesores

Como usuario quiero poder ver la lista de profesores.

Criterios de Aceptación:

- Cuando presione en el menú el botón profesores.
- Se deberá mostrar la lista de profesores.

A.0.27 Activar suscripción

Como usuario quiero poder elegir entre pagar un curso completo o pagar una suscripción para poder acceder a todos los cursos.

Criterios de Aceptación:

- Presionamos en el botón ir a mi perfil.
- Damos click en el botón activar suscripción.
- Rellenamos el formulario de la pasarela de pago.

Criterios de Validación:

- Los campos serán validados y se mostrara un mensaje de error en caso de que sean inválidos.

A.0.28 Desactivar suscripción

Como usuario registrado quiero poder cancelar la suscripción.

Criterios de Aceptación:

- Presionamos en el botón ir a mi perfil.
- Damos click en el botón desactivar suscripción.
- Y confirmamos.

Criterios de Validación:

- El usuario debe estar suscrito.

A.0.29 Crear una sección dentro de un curso

Como profesor quiero crear una sección dentro de mi curso para poder organizar bien el contenido.

Criterios de Aceptación:

- Ir al curso.
- Presionar en el botón más.
- Y rellenar el formulario.

Criterios de Validación:

- La sección debe existir.

A.0.30 Editar una sección dentro de un curso

Como profesor quiero editar una sección dentro de mi curso para poder organizar bien el contenido.

Criterios de Aceptación:

- Ir al curso.
- Presionar en el botón editar.
- Y actualizar la información del formulario.

Criterios de Validación:

- La sección debe existir.

A.0.31 Eliminar una sección dentro de un curso

Como profesor quiero eliminar una sección dentro de mí curso.

Criterios de Aceptación:

- Ir al curso.
- Presionar en el botón eliminar.
- Y confirmar.

Criterios de Validación:

- La sección debe existir.
- Para poder eliminar la sección debe estar vacía.

A.0.32 Ordenar las secciones dentro de un curso

Como profesor quiero ordenar las secciones de mí curso.

Criterios de Aceptación:

- Ir al curso.
- Arrastrar la sección donde desee.

Criterios de Validación:

- La sección debe existir.

A.0.33 Crear un recurso dentro de una sección

Como profesor quiero poder crear un recurso mío dentro de una sección.

Criterios de Aceptación:

- Ir al curso.
- Presionar en el botón de añadir recurso.
- Y rellenar el formulario.

Criterios de Validación:

- La sección debe existir.

A.0.34 Editar un recurso dentro de una sección

Como profesor quiero poder editar un recurso mío dentro de una sección.

Criterios de Aceptación:

- Ir al curso.
- Presionar en el botón de editar recurso.
- Y editar los campos del formulario.

Criterios de Validación:

- La sección debe existir.
- El recurso debe existir.

A.0.35 Eliminar un recurso dentro de una sección

Como profesor quiero eliminar un recurso mío dentro de una sección.

Criterios de Aceptación:

- Ir al curso.
- Presionar en el botón editar recurso.
- Presionar el botón de eliminar.
- Y confirmar.

Criterios de Validación:

- La sección debe existir.
- El recurso debe existir.

A.0.36 Ordenar los recursos dentro de una sección

Como profesor quiero ordenar los recursos.

Criterios de Aceptación:

- Ir al curso.
- Arrastrar el recurso donde desee.

Criterios de Validación:

- La sección debe existir.
- El recurso debe existir.

Apéndice B

B.1 Manual de Usuario

La aplicación está desplegada en Heroku para que el tribunal pueda hacer uso de esta sin tener que hacer un Set Up local.

Al tener un servidor gratuito en Heroku, los servicios se apagan tras unos minutos de inactividad, con lo que la primera llamada tardará más de lo normal dado que tiene que levantar dos servidores.

<https://ulpgc-course.herokuapp.com/>

usuarios: admin@tfg.com profesor@tfg.com alumno@tfg.com
contraseña: secret

B.1.1.1 Set Up Laravel (API)

Para la instalación de Laravel es necesario tener instalado los siguientes servicios:

- Servidor web Apache ≥ 2.4 o Nginx > 1.2 .
- Base de Datos MySQL ≥ 5.6 o PostgreSQL ≥ 9 .
- Lenguaje PHP $\geq 7.0.0$.
 - OpenSSL PHP Extension.
 - PDO PHP Extension.
 - Mbstring PHP Extension.
 - Tokenizer PHP Extension.
 - XML PHP Extension.
- Manejador de Dependencias **Composer** $\geq 1.2.0$.

También se puede hacer uso de software ya empaquetado y configurados como pueden ser:

- XAMPP
- WAMP
- MAMP

Además, tenemos que tener una cuenta de AWS S3 para la subida de ficheros en la nube. Después de tener estos servicios instalados, creamos una base de datos desde phpadmin o desde el terminal y ejecutamos lo siguiente:

1. `cd [ubicación_del_proyecto]`
2. `composer install`

Una vez instaladas todas las dependencias de Laravel procedemos a cambiar las variables de entorno ubicadas en el fichero `.env`, las variables que debemos actualizar son:

- DB_CONNECTION=[mysql|pgsql]
- DB_HOST=127.0.0.1
- DB_PORT=XXXX
- DB_DATABASE=[nombre_de_la_base_de_datos]
- DB_USERNAME=[nombre_de_usuario]
- DB_PASSWORD=XXX
- FRONT_END_END_POINT=XXX

Y para finalizar ejecutamos “php artisan migrate –seed” para insertar las tablas en la base de datos y poblarlas de datos.

B.1.1.2 Postman – Test API

Rutas de la API

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	POST	/api/v1/auth/login		App\Http\Controllers\AuthController@login	api
	POST	/api/v1/auth/logout		App\Http\Controllers\AuthController@logout	api, auth:api
	POST	/api/v1/auth/me		App\Http\Controllers\AuthController@me	api, auth:api
	POST	/api/v1/auth/refresh		App\Http\Controllers\AuthController@refresh	api, auth:api
	POST	/api/v1/auth/register		App\Http\Controllers\AuthController@register	api
	POST	/api/v1/cart		App\Http\Controllers\CartController@pay	api, auth:api
	POST	/api/v1/courses		App\Http\Controllers\CourseController@store	api, auth:api, role:admin teacher
	GET HEAD	/api/v1/courses		App\Http\Controllers\CourseController@index	api
	GET HEAD	/api/v1/courses/myCourse		App\Http\Controllers\CourseController@myCourse	api, auth:api
	POST	/api/v1/courses/photo/{course}		App\Http\Controllers\CourseController@photo	api, auth:api, role:admin teacher
	POST	/api/v1/courses/section		App\Http\Controllers\CourseSectionController@store	api, auth:api, role:admin teacher
	POST	/api/v1/courses/section/resource		App\Http\Controllers\CourseResourceController@store	api, auth:api, role:admin teacher
	DELETE	/api/v1/courses/section/resource/{courseResource}		App\Http\Controllers\CourseResourceController@destroy	api, auth:api, role:admin teacher
	PUT	/api/v1/courses/section/resource/{courseResource}		App\Http\Controllers\CourseResourceController@update	api, auth:api, role:admin teacher
	DELETE	/api/v1/courses/section/{courseSection}		App\Http\Controllers\CourseSectionController@destroy	api, auth:api, role:admin teacher
	GET HEAD	/api/v1/courses/{course}		App\Http\Controllers\CourseController@show	api, auth:api
	PUT	/api/v1/courses/{course}		App\Http\Controllers\CourseController@update	api, auth:api, role:admin teacher
	DELETE	/api/v1/courses/section/{course}/vote		App\Http\Controllers\CourseController@destroy	api, auth:api, role:admin teacher
	PUT	/api/v1/courses/section/{courseSection}/vote		App\Http\Controllers\CourseSectionController@update	api, auth:api
	POST	/api/v1/password/email		App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail	api, guest
	POST	/api/v1/password/reset		App\Http\Controllers\Auth\ResetPasswordController@reset	api, guest
	GET HEAD	/api/v1/teachers		App\Http\Controllers\UserController@teachers	api
	GET HEAD	/api/v1/users		App\Http\Controllers\UserController@index	api, auth:api, role:admin
	GET HEAD	/api/v1/users/payment		App\Http\Controllers\UserController@payment	api, auth:api
	DELETE	/api/v1/users/{user}		App\Http\Controllers\UserController@destroy	api, auth:api
	PUT	/api/v1/users/{user}		App\Http\Controllers\UserController@update	api, auth:api
	GET HEAD	/api/v1/users/{user}		App\Http\Controllers\UserController@show	api, auth:api, role:admin

Figura B.1.1.2: Rutas de la API

Registro

POST `{{url}}/api/v1/auth/register` Params Send Save

Authorization Headers (2) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1 - {
2   "name": "Postman",
3   "last_name": "Postman",
4   "email": "postman@gmail.com",
5   "password": "secret"
6 }

```

Body Cookies (2) Headers (10) Test Results Status: 201 Created Time: 2180 ms Size: 476 B

Pretty Raw Preview JSON Save Response

```

1 - {
2   "user": {
3     "name": "Postman",
4     "last_name": "Postman",
5     "email": "postman@gmail.com",
6     "updated_at": "2018-07-13 17:20:24",
7     "created_at": "2018-07-13 17:20:24",
8     "id": 52
9   }
10 }

```

Figura B.1.1.2: Postman Registro

Refresh Token

The screenshot shows a Postman interface for a POST request to `POST {{url}}/api/v1/auth/refresh`. The request headers are set to `Accept: application/json` and `Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...`. The response status is `200 OK` with a time of `156 ms` and a size of `979 B`. The response body is a JSON object:

```
1 {
2   "user": {
3     "id": 1,
4     "name": "Mario",
5     "last_name": "Oberbrunner",
6     "email": "admin@tfg.com",
7     "photo": null,
8     "created_at": "2018-07-03 23:50:26",
9     "updated_at": "2018-07-07 23:01:30",
10    "deleted_at": null,
11    "stripe_id": null,
12    "card_brand": null,
13    "card_last_four": null,
14    "trial_ends_at": null
15  },
16  "rol": "admin",
17  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
    .eyJpc3MiOiJodHRwOlwvXC90ZmcudGVzdFwvYXBPXC92MwVwYXV0aFwvcnVmcnVzaCI6Im1hdCI6MTUzMTUwMzgyNSwiZXhwIjojNTYzMDQxMjY
    0LCJyYmYiOiJlMzE1MDUyNjQsImp0aSI6ImUxejFyRkQwc1FDeHh4dWIiLCJzdWIiOiJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
    zYTE0ZTBiMDQ3NTQzYWEifQ.eo5pmoa_Ati4ouv0Art-BGkMu3m2U_XCf_1rb0bBBKY",
18  "expires_in": 31536000
19 }
```

Figura B.1.1.2: Postman Refresh Token

Logout

The screenshot shows a Postman interface for a POST request to `POST {{url}}/api/v1/auth/logout`. The request headers are set to `Accept: application/json` and `Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...`. The response status is `200 OK` with a time of `97 ms` and a size of `373 B`. The response body is a JSON object:

```
1 {
2   "message": "Successfully logged out"
3 }
```

Figura B.1.1.2: Postman Logout

Me

The screenshot shows a Postman interface for a POST request to `{{url}}/api/v1/auth/me`. The request headers include `Accept: application/json` and `Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ...`. The response body is a JSON object:

```
1 {
2   "user": {
3     "id": 1,
4     "name": "Mario",
5     "last_name": "Oberbrunner",
6     "email": "admin@tfq.com",
7     "photo": null,
8     "created_at": "2018-07-03 23:50:26",
9     "updated_at": "2018-07-07 23:01:30",
10    "deleted_at": null,
11    "stripe_id": null,
12    "card_brand": null,
13    "card_last_four": null,
14    "trial_ends_at": null
15  },
16  "rol": "admin",
```

Figura B.1.1.2: Postman Me

Usuarios

The screenshot shows a Postman interface for a GET request to `{{url}}/api/v1/users`. The request headers include `Accept: application/json` and `Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ...`. The response body is a JSON object:

```
1 {
2   "users": {
3     "current_page": 1,
4     "data": [
5       {
6         "id": 2,
7         "name": "Mrs. Etha Schmeler",
8         "last_name": "Vandervort",
9         "email": "turner10@hotmail.com",
10        "photo": null,
11        "created_at": "2018-07-03 23:50:29",
12        "updated_at": "2018-07-03 23:50:29",
13        "deleted_at": null,
14        "stripe_id": null,
15        "card_brand": null,
16        "card_last_four": null,
17        "trial_ends_at": null,
18        "rol": "teacher"
19      },
```

Figura B.1.1.2: Postman Usuarios

Para no subir una captura de todas las rutas adjunto el fichero de Postman donde están todas las llamadas a la API.

https://drive.google.com/open?id=1FvWXUWrCv_cHdJoI7G1fGUC4YOMXxtxx

B.1.2.1 Set Up React JS

1. Instalar [Node JS](#)
2. Abrir el terminal.
3. cd [ubicación_del_proyecto]
4. Modificamos el valor de REACT_APP_API ubicado en el archivo .env .
5. Ejecutar en la consola “npm i”
6. npm start

B.1.3 Enlaces al código fuente

<https://github.com/ckwanted/tfg-api>

<https://github.com/ckwanted/tfg-web>

B.1.4 Videos sobre el uso de la aplicación

Login

<https://drive.google.com/open?id=1R7emD3PTWcfogT7t1VOXmwL-09f740I1>

Registro

https://drive.google.com/open?id=1XFSyvbrn8jP3NsEDC_s0Qf5LYeHi3kqS

Comprar Curso (Alumno)

<https://drive.google.com/open?id=1LIc4k44DHtlhueFzJGQ43Axv5EitsMMb>

Editar Curso (Profesor o Administrador)

<https://drive.google.com/open?id=1muZVe67b4QMUKmg0Jogp2QDkWIIabEIS>

Editar usuarios (Administrador)

<https://drive.google.com/open?id=1gQe9FPYpzyBVPRWDTnzToLL95tVe4YH>

Apéndice C

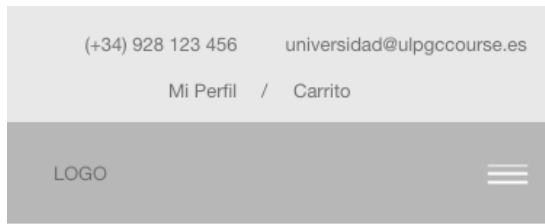
Wireframe



Figura c.1: Wireframe Home XS



Figura c.2: Wireframe Cursos XS



Contenido Del Curso

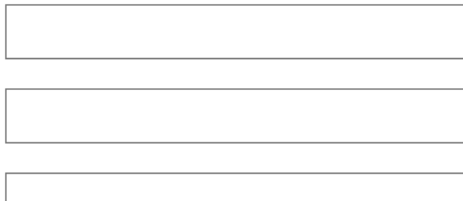


Figura c.3: Wireframe Detalle Del Curso XS

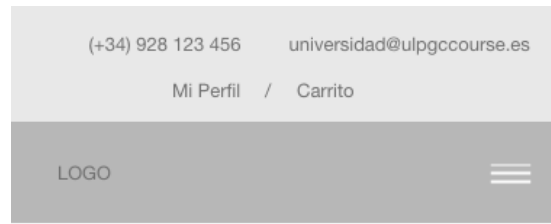
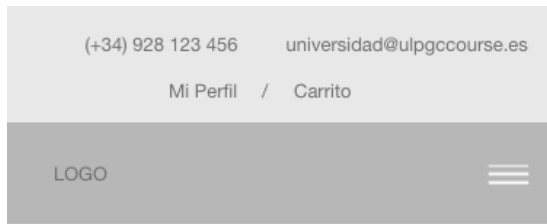


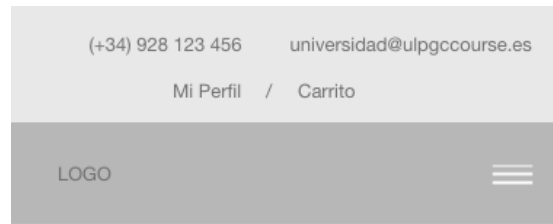
Figura c.4: Wireframe Profesores XS



Texto



Figura c.5: Wireframe Sobre Nosotros XS



Mi Carrito

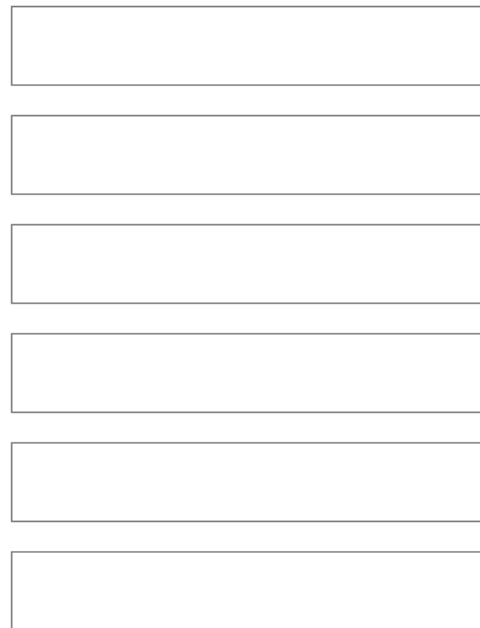
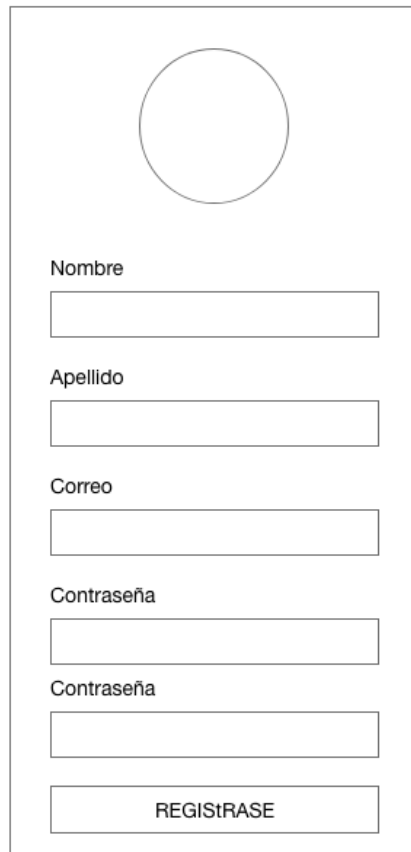


Figura c.6: Wireframe Carrito XS



A wireframe for a login page. At the top center is a large empty circle. Below it, the text "Nombre" is followed by a horizontal input field. Below that, the text "Contraseña" is followed by another horizontal input field. At the bottom, there is a rectangular button with the text "INICIAR SESIÓN" centered inside.

Figura c.7: Wireframe Login XS



A wireframe for a registration page. At the top center is a large empty circle. Below it, the text "Nombre" is followed by a horizontal input field. Below that, the text "Apellido" is followed by another horizontal input field. Below that, the text "Correo" is followed by a horizontal input field. Below that, the text "Contraseña" is followed by a horizontal input field. Below that, the text "Contraseña" is followed by another horizontal input field. At the bottom, there is a rectangular button with the text "REGISTARSE" centered inside.

Figura c.8: Wireframe Registro XS

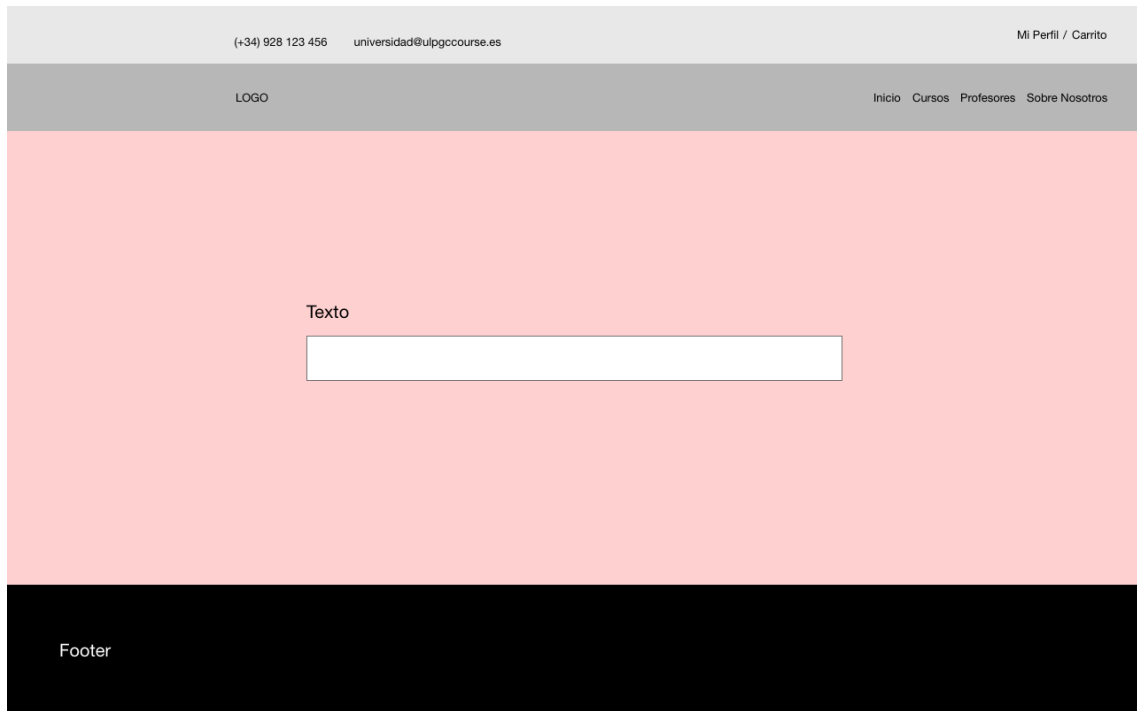


Figura c.9: Wireframe Home

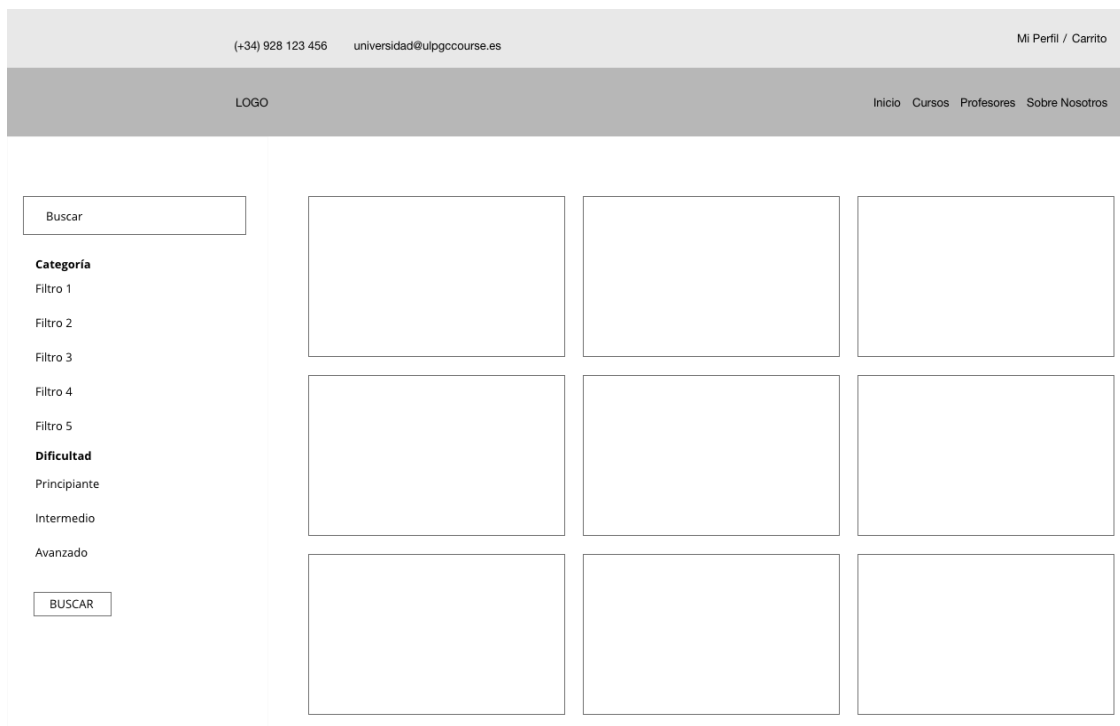


Figura c.10: Wireframe Cursos

Título Del Curso
Descripción Del Curso
Estrellas 2.0 (número de valoraciones)
Creado por Jonh Doe



Contenido Del Curso

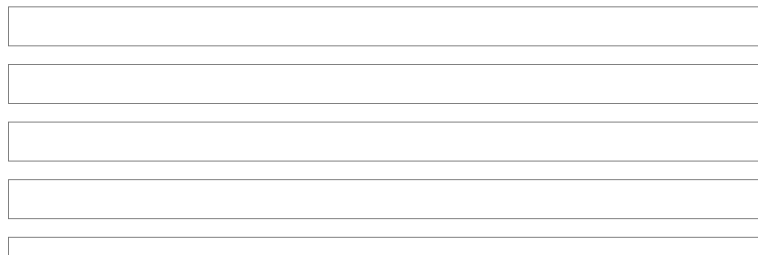


Figura c.11: Wireframe Detalle Del Curso



Figura c.12: Wireframe Profesores



Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno estándar de las industrias desde el año 1500, cuando un impresor (N. del T. persona que se dedica a la

Enlace Cursos

Biblioteca

Enlace Profesores

Cursos Populares



Figura c.13: Wireframe Sobre Nosotros

Nombre

Contraseña

INICIAR SESIÓN

Figura c.14: Wireframe Login

A wireframe of a registration form. At the top center is a circular placeholder for a profile picture. Below it are five input fields, each with a label to its left: 'Nombre', 'Apellido', 'Correo', 'Contraseña', and 'Contraseña'. At the bottom of the form is a rectangular button labeled 'REGISTRASE'.

Figura c.15: Wireframe Registro

Mockup

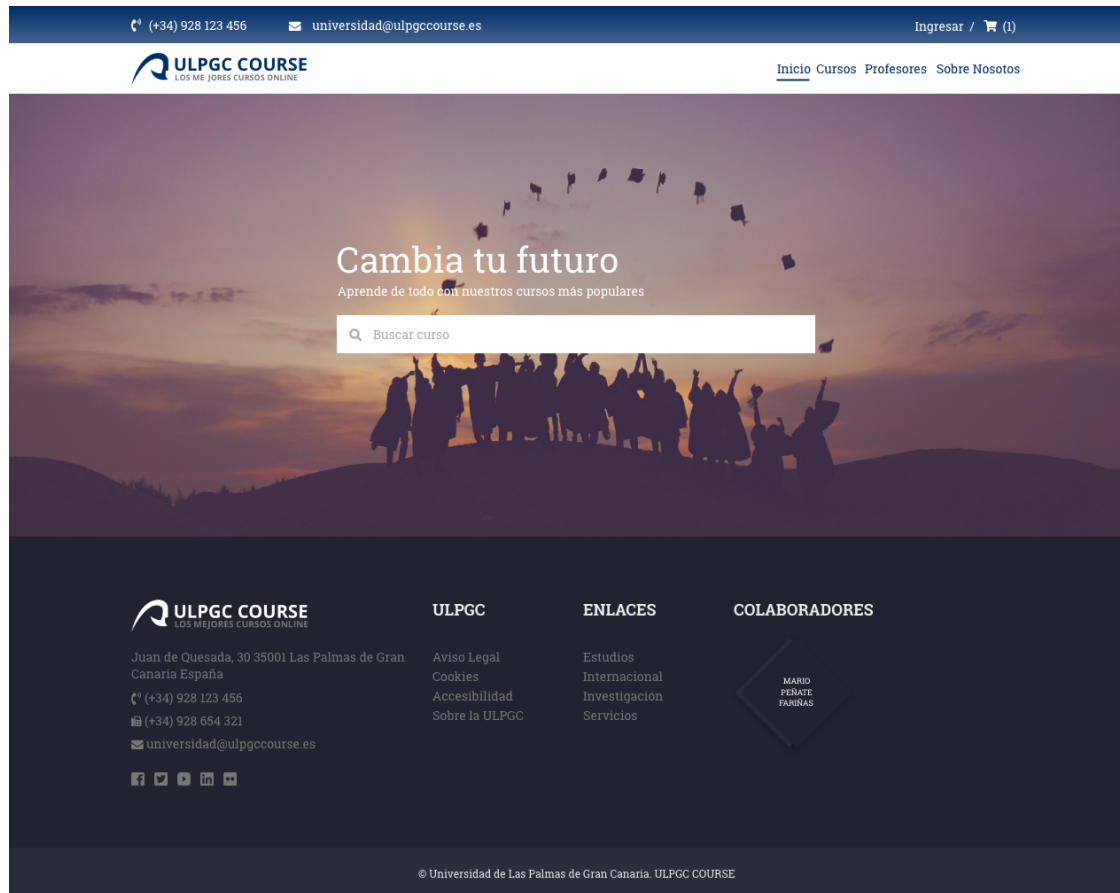


Figura c.16: Mockup Home

Buscar Curso

Categoría

- Front End
- Back End
- Full Stack
- Dev Ops
- Android
- IOS

Categoría

- Principiante
- Intermedio
- Avanzado

BUSCAR



Nombre Del Curso
Nombre Del Profesor
★★★★★
Voluptatem quaerat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Curso
Nombre Del Profesor
★★★★★
Voluptatem quaerat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Curso
Nombre Del Profesor
★★★★★
Voluptatem quaerat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Curso
Nombre Del Profesor
★★★★★
Voluptatem quaerat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.




Nombre Del Curso
Nombre Del Profesor
★★★★★
Voluptatem quaerat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Curso
Nombre Del Profesor
★★★★★
Voluptatem quaerat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.

Figura c.17: Mockup Cursos

(+34) 928 123 456 universidad@ulpgcourse.es
Ingresar / (1)



ULPGC COURSE
LOS MEJORES CURSOS ONLINE


[Inicio](#)
[Cursos](#)
[Profesores](#)
[Sobre Nosotros](#)

Nombre Del Curso

Voluptatem queraat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.


★★★★★ 5.0 (112 valoraciones)

Creado por John Doe



Contenido del curso

Tema 1	▼
Tema 2	▼
Tema 3	▼
Tema 4	▼
Tema 5	▼
Tema 6	▼



Juan de Quesada, 30 35001 Las Palmas de Gran Canaria España

(+34) 928 123 456

(+34) 928 654 321

universidad@ulpgcourse.es

[f](#) [t](#) [y](#) [in](#) [v](#)

ULPGC

Aviso Legal
Cookies
Accesibilidad
Sobre la ULPGC

ENLACES

Estudios
Internacional
Investigación
Servicios

COLABORADORES

MARID
PERANTE
FARIÑAS

© Universidad de Las Palmas de Gran Canaria. ULPGC COURSE

Figura c.18: Mockup Detalle Del Curso



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.



Nombre Del Profesor
Número de cursos
Voluptatem quaeat dolorem ex quis quos eos praesentium consequatur. Quae nulla tempore velit dolor. Voluptas cupiditate molestiae dolore est.

Figura c.19: Mockup Profesores



Sobre ULPGC COURSE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quas voluptatem maiores eaque similique non distinctio voluptates perspiciatis omnis, repellendus ipsa aperiam, laudantium voluptatum nulla?.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quas voluptatem maiores eaque similique non distinctio voluptates perspiciatis omnis, repellendus ipsa aperiam, laudantium voluptatum nulla?.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quas voluptatem maiores eaque similique non distinctio voluptates perspiciatis omnis, repellendus ipsa aperiam, laudantium voluptatum nulla?.



Figura c.20: Mockup Sobre Nosotros

Prototipo

<https://drive.google.com/file/d/1VoyrW0yTvY5aF0vOlinH7NhBM-Lc-wQf/view?usp=sharing>

Archivo del Prototipo (Adobe XD)

https://drive.google.com/open?id=1aGv2OGsWvYE_Ineaj_gsBH9Tc9qsXu84