



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

**Implementación y Diseño del Sistema de
Suscripciones a Proyectos dentro de Edueca**

Autor:
Gerard CURTO LLOPIS

Supervisor:
Héctor SAIZ SÁNCHEZ
Tutor académico:
Vicente CHOLVI JUAN

Fecha de lectura: 17 de julio de 2018
Curso académico 2017/2018

Resumen

En este documento, se describe la estancia en prácticas realizada en *Centre for the Innovation and Development of Education and Technology* (CIDET). Durante esa estancia en la empresa, se ha iniciado el diseño y la implementación del sistema de suscripciones de la plataforma de e-learning "Edueca", donde se han llevado a cabo diversas tareas. Para la implementación de este sistema de suscripciones, se ha utilizado un conjunto de tecnologías entre las que cabe destacar: HTML, PHP, CSS, Javascript y el framework Symfony. Por último, para la realización de este proyecto se ha decidido seguir la metodología de desarrollo ágil (SCRUM).

Palabras clave

Symfony, Edueca, MVC (Modelo-Vista-Controlador), PHP, SCRUM.

Keywords

Symfony, Edueca, MVC (Model-View-Controller), PHP, SCRUM.

Índice general

| | |
|---|-----------|
| 1. Introducción | 11 |
| 1.1. Contexto y motivación del proyecto | 11 |
| 1.1.1. La empresa | 11 |
| 1.1.2. El proyecto | 12 |
| 1.2. Objetivos del proyecto | 12 |
| 2. Descripción del proyecto | 15 |
| 2.1. Situación inicial | 15 |
| 2.2. Tecnologías y herramientas | 15 |
| 2.2.1. Lenguajes | 15 |
| 2.2.2. Frameworks y librerías | 17 |
| 2.2.3. Servidor web | 18 |
| 2.2.4. Bases de datos | 18 |
| 2.2.5. Control de versiones | 18 |
| 2.2.6. Entornos de desarrollo | 19 |
| 2.2.7. Gestión de proyectos | 19 |
| 2.2.8. Diseño | 20 |
| 3. Planificación del proyecto | 21 |

| | |
|---|-----------|
| 3.1. Metodología | 21 |
| 3.2. Planificación | 22 |
| 3.3. Estimación de recursos y costes del proyecto | 22 |
| 3.4. Seguimiento del proyecto | 23 |
| 3.5. Desviaciones en la planificación | 23 |
| 4. Análisis y diseño del sistema | 25 |
| 4.1. Análisis del sistema | 25 |
| 4.1.1. Requisitos funcionales | 25 |
| 4.1.2. Requisitos no funcionales | 30 |
| 4.1.3. Requisitos de datos | 31 |
| 4.2. Diagramas | 33 |
| 4.2.1. Diagrama de casos de uso | 33 |
| 4.2.2. Diagrama de actividades | 35 |
| 4.2.3. Diagrama de clases | 36 |
| 4.3. Diseño de la arquitectura del sistema | 37 |
| 4.4. Diseño de la interfaz | 38 |
| 5. Implementación | 45 |
| 5.1. Sprint 1 | 45 |
| 5.1.1. Planificación del Sprint | 45 |
| 5.1.2. Ejecución del Sprint | 45 |
| 5.2. Sprint 2 | 47 |
| 5.2.1. Planificación del Sprint | 47 |
| 5.2.2. Ejecución del Sprint | 48 |
| 5.3. Sprint 3 | 49 |

| | | |
|-----------|------------------------------------|-----------|
| 5.3.1. | Planificación del Sprint | 49 |
| 5.3.2. | Ejecución del Sprint | 50 |
| 5.4. | Sprint 4 | 51 |
| 5.4.1. | Planificación del Sprint | 51 |
| 5.4.2. | Ejecución del Sprint | 52 |
| 5.5. | Sprint 5 | 53 |
| 5.5.1. | Planificación del Sprint | 53 |
| 5.5.2. | Ejecución del Sprint | 54 |
| 5.6. | Sprint 6 | 55 |
| 5.6.1. | Planificación del Sprint | 55 |
| 5.6.2. | Ejecución del Sprint | 56 |
| 6. | Extensiones del proyecto | 57 |
| 7. | Conclusiones | 59 |
| 7.1. | Ámbito formativo | 59 |
| 7.2. | Ámbito personal | 59 |
| | Bibliografía | 59 |
| A. | Estudio de Symfony | 63 |
| A.1. | Twig | 63 |
| A.2. | Doctrine | 64 |

Índice de tablas

| | |
|---|----|
| 4.1. Requisito funcional «CU01. Gestionar suscripciones» | 26 |
| 4.2. Requisito funcional «CU02. Gestionar pagos» | 26 |
| 4.3. Requisito funcional «CU03. Gestionar datos de facturación» | 27 |
| 4.4. Requisito funcional «CU04. Buscar suscripciones» | 27 |
| 4.5. Requisito funcional «CU05. Buscar pagos» | 28 |
| 4.6. Requisito funcional «CU06. Añadir suscripción» | 28 |
| 4.7. Requisito funcional «CU07. Modificar suscripción» | 29 |
| 4.8. Requisito funcional «CU08. Cancelar suscripción» | 29 |
| 4.9. Requisito no funcional «RNF01. - Interfaz» | 30 |
| 4.10. Requisito no funcional «RNF02. - Interfaz móvil» | 30 |
| 4.11. Requisito no funcional «RNF03. - Usabilidad» | 30 |
| 4.12. Requisito no funcional «RNF04. - Tiempo» | 30 |
| 4.13. Requisito no funcional «RNF05. - Idiomas» | 31 |
| 4.14. Requisito no funcional «RNF06. - Seguridad» | 31 |
| 4.15. Requisito de datos «RD01. - Suscripción» | 31 |
| 4.16. Requisito de datos «RD02. - Tarifa» | 32 |
| 4.17. Requisito de datos «RD03. - Método de pago» | 32 |
| 4.18. Requisito de datos «RD04. - Promoción» | 32 |
| 4.19. Requisito de datos «RD05. - Datos de facturación» | 32 |

| | |
|--|----|
| 4.20. Requisito de datos «RD06. - Requerimiento de pago» | 33 |
| 4.21. Requisito de datos «RD07. - Pago» | 33 |
| 5.1. Pila del producto del Sprint 2 | 47 |
| 5.2. Pila del producto del Sprint 3 | 49 |
| 5.3. Pila del producto del Sprint 4 | 51 |
| 5.4. Pila del producto del Sprint 5 | 53 |
| 5.5. Pila del producto del Sprint 6 | 55 |

Índice de figuras

| | |
|--|----|
| 3.1. Diagrama de Gantt | 24 |
| 4.1. Diagrama de casos de uso | 34 |
| 4.2. Diagrama de actividades | 35 |
| 4.3. Diagrama de clases | 36 |
| 4.4. Arquitectura del sistema | 37 |
| 4.5. Página principal de suscripciones | 39 |
| 4.6. Página principal de suscripciones con pequeña resolución de pantalla | 40 |
| 4.7. Modificación de datos de facturación | 41 |
| 4.8. Modificación de datos de facturación con pequeña resolución de pantalla | 42 |
| 4.9. Selección de una suscripción de Edueca | 43 |
| 4.10. Selección de una suscripción distinta a Edueca | 44 |
| 5.1. Planificación del proyecto | 46 |

Capítulo 1

Introducción

En este primer capítulo se exponen tanto el contexto y motivación del proyecto como los objetivos que se esperan conseguir mediante el desarrollo del proyecto.

1.1. Contexto y motivación del proyecto

En esta parte, se presenta por un lado la empresa en la que se ha realizado el proyecto y por otro lado el proyecto.

1.1.1. La empresa

El *Centre for the Innovation and Development of Education and Technology* (CIDET) es una empresa que pertenece al ecosistema del Espaitec de la Universitat Jaume I desde 2015. La empresa se divide principalmente en tres sectores: cursos, servicios y eLearning. Además, algunas de las actividades de la empresa son la formación, consultoría, creación de materiales y recursos tecnológicos.

El gran objetivo de la empresa es lograr que la educación y las tecnologías sean aprovechadas de una forma efectiva para beneficio personal y de la sociedad. Se diseñan y ofrecen entornos de aprendizaje, materiales educativos y actividades formativas para que las personas puedan aumentar el conocimiento, habilidades y capacidades en competencias y habilidades, pero también en valores y actitudes [1].

Hace poco, la empresa lanzó la plataforma de eLearning Edueca, donde centros de formación, equipos, profesores y cualquier persona interesada en ofrecer cursos puede gestionar el proceso de enseñanza-aprendizaje de modo virtual.

1.1.2. El proyecto

El proyecto cuya propuesta se presenta en este documento, va a tratar de una ampliación de una de las partes de Edueca. Actualmente, cuenta con un pequeño sistema de pagos que solo permite a los profesores realizar pagos para la creación de las cuentas estándar y pro. Al disponer de este tipo de cuentas, un profesor tiene derecho a crearse sus propios cursos, pero se desea ampliar este sistema de pagos para poder añadirle más funcionalidad.

Entonces, este proyecto consiste en desarrollar un sistema de suscripciones dentro de Edueca, donde usuarios registrados deben pagar para poder inscribirse en unos determinados cursos si éstos lo requieren por parte del profesor.

El sistema debe permitir a los usuarios registrados poder elegir el tipo de pago y realizarlo a través de la plataforma. Además, cada profesor tiene la opción de elegir si los usuarios deben de pagar para inscribirse a su curso.

El sistema ha de gestionar las tarifas que se añadan para cada curso. Otra cosa a tener en cuenta, es que un profesor puede realizar descuentos en alguna tarifa durante un periodo de tiempo para el usuario que haya recibido la oferta.

Un sistema como éste mejoraría el sistema de pagos actual, otorgándole nueva funcionalidad que facilitará el pago de los alumnos hacia los profesores.

En cuanto al alcance, este proyecto abarca toda la parte de creación de las nuevas interfaces, gestión de las clases relacionadas con el sistema de suscripciones y el almacenamiento de los datos que introduzcan los usuarios a la hora de realizar un pago. Además, también incluye la funcionalidad que adquirirá este nuevo sistema de suscripciones. Sin embargo, el alcance del proyecto no incluye la creación de cuentas de usuarios y cursos, ya que fueron implementados con anterioridad y este proyecto se trata de la ampliación de la funcionalidad de Edueca.

1.2. Objetivos del proyecto

El objetivo principal de este proyecto es conseguir un sistema de suscripciones que facilite la gestión y pagos de las suscripciones que tienen los usuarios registrados en Edueca.

El objetivo principal se puede desglosar en los siguientes subobjetivos:

- Diseño e implementación de un panel que le permita a un alumno la visualización de sus suscripciones y pagos.
- Modificación de la vista de los proyectos de pago para que incluyan una nueva sección donde se podrán ver las tarifas disponibles.
- Diseño e implementación de un panel que le permita a un profesor la administración de las tarifas que desea en sus proyectos.

- Diseño e implementación de un formulario que le permita a un alumno gestionar sus datos de facturación.
- Diseño e implementación de un panel que le permita a un profesor la administración de las promociones que desea en sus tarifas.
- Implementación de los diferentes métodos de pago.

Capítulo 2

Descripción del proyecto

En este capítulo, se presenta la situación inicial a partir de la cual se va a desarrollar el proyecto. Además, se van a comentar todas las tecnologías y herramientas necesarias para llevar a cabo el proyecto.

2.1. Situación inicial

En la actualidad, nos encontramos con un sistema que permite a usuarios registrados suscribirse a cursos ofrecidos por profesores. Sin embargo, los profesores aún no pueden aplicar tarifas de pago a sus cursos, solo están disponibles de momento las tarifas de pago que ofrece Edueca.

Por ahora no existe una sección para que los usuarios vean y gestionen todas sus suscripciones y pagos. Esto resultaría un problema grave en el momento que los usuarios empezasen a realizar suscripciones de pagos, ya que no tendrían una zona donde poder consultar todas sus suscripciones y estado de los pagos.

2.2. Tecnologías y herramientas

En esta parte, se van a comentar todas aquellas tecnologías y herramientas que han intervenido en el desarrollo del proyecto.

2.2.1. Lenguajes

JavaScript

JavaScript (abreviado comúnmente JS) 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 suele utilizar en el lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas [10].

En el proyecto, JavaScript ha sido utilizado para otorgarles funcionalidad a los diferentes botones que aparecen en las páginas. Además, ha servido para modificar el formato de algunos campos.

HTML5

HTML, siglas en inglés para HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos.

Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web. Además, es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado [9].

En el proyecto, HTML5 ha sido utilizado para definir la estructura de todos los elementos que deben aparecer en cada una de las interfaces.

CSS

Hojas de estilo en cascada (o CSS, siglas en inglés de Cascading Stylesheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML.

Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web, y GUIs para muchas aplicaciones móviles.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes [6].

En el proyecto, CSS ha sido utilizado para modificar el diseño de algunos elementos puestos con HTML. Donde más tiempo se ha invertido ha sido en el diseño de los botones del formulario de datos de facturación del usuario, ya que van a servir como estándar para los próximos formularios que se creen en Edueca.

2.2.2. Frameworks y librerías

Symfony 3

Symfony es un framework que optimiza el desarrollo de aplicaciones web basado en el patrón Modelo Vista Controlado. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación [14].

En el proyecto, Symfony ha sido utilizado para la implementación de la plataforma Edueca con el objetivo de facilitar el desarrollo y hacer el sistema mantenible y adaptable a nuevos cambios.

Bootstrap

Bootstrap es un framework web 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 opcionales adicionales [5].

En el proyecto, Bootstrap ha sido utilizado para la creación de las tablas que representan las suscripciones y los requerimientos de pago que tiene un usuario registrado.

Twig

Twig es un motor de plantillas para el lenguaje de programación PHP. Mediante estas plantillas se puede llamar a funciones, extenderlas, incluirlas dentro de su código, aplicar filtros, etc. Además, Twig está integrado dentro de Symfony como su motor de plantilla por defecto [15].

En el proyecto, Twig ha resultado útil a la hora de interactuar la vista con datos obtenidos a través del controlador.

Doctrine

Doctrine es una biblioteca integrada en Symfony cuyo objetivo es facilitar el acceso a la base de datos. Doctrine automatiza la conexión a la base de datos, las peticiones a la misma y las inserciones a través de objetos [7].

En el proyecto, Doctrine ha permitido realizar algunas acciones, como por ejemplo actualizar un dato, sin tener que escribir ninguna sentencia SQL.

2.2.3. Servidor web

Apache

Apache es un servidor web HTTP abierto y gratuito que está desarrollado y mantenido por la Apache Software Foundation. Este servidor web puede ser utilizado en cualquier plataforma y es uno de los servidores web más utilizados en la actualidad [3].

Apache es el encargado de recibir las peticiones web provenientes de los usuarios y del envío del archivo HTML oportuno. Esta funcionalidad hace posible el acceso y navegación a la plataforma web.

2.2.4. Bases de datos

MySQL

MySQL es un sistema de gestión de bases de datos relacional. Además, este sistema de gestión de bases de datos es uno de los más populares actualmente para entornos de desarrollo web. Es una base de datos con una alta velocidad de lectura y esto la hace ideal para el desarrollo web puesto que la carga de páginas web ha de ser tan rápida como sea posible [12].

En el proyecto, MySQL ha sido utilizado para la creación de las tablas necesarias que almacenan todos los datos referentes al nuevo sistema de suscripciones.

2.2.5. Control de versiones

Git

Git es un software de control de versiones que consiste en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos [8].

Bitbucket

Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Mercurial y Git [4].

En el proyecto, Bitbucket ha permitido que cada miembro de la empresa pueda trabajar en una rama diferente para el proyecto de Edueca. Además, tenemos varias opciones como integrar

en nuestra rama cambios de otras, o subir y comentar la parte nueva de nuestro código para que los demás miembros lo puedan ver.

2.2.6. Entornos de desarrollo

PhpStorm

PhpStorm es el entorno de desarrollo integrado (IDE) que se ha elegido para el desarrollo del proyecto. PhpStorm es un IDE comercial multiplataforma para PHP construido en la plataforma IntelliJ IDEA de JetBrains.

PhpStorm proporciona un editor para PHP, HTML y JavaScript con análisis de código sobre la marcha, prevención de errores y refactorizaciones automatizadas para códigos PHP y JavaScript [13].

En el proyecto, PhpStorm ha facilitado las tareas ya que ofrece compatibilidad con un gran número de frameworks. Además, otra de las ventajas es la gran cantidad de documentación que tiene disponible.

2.2.7. Gestión de proyectos

JIRA

JIRA es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias, y para la gestión operativa de proyectos. Esta herramienta contiene mecanismos que facilitan el desarrollo colaborativo del proyecto así como el seguimiento del mismo [11].

Uno de los mecanismos más interesantes es su pizarra de Kanban. Gracias a ella, el equipo de trabajo puede ver de forma rápida y sencilla el estado de las tareas, así como modificarlas o añadir comentarios. Además, esta pizarra es totalmente personalizable. Tiene por defecto tres columnas, "TO DO", "DOING", "DONE" pero permite añadir nuevas columnas, lo que aporta una mayor flexibilidad.

Para la elaboración del proyecto, también hemos utilizado dos columnas más llamadas "STAND BY", y "TESTING". Estas ayudan a la hora de saber cuales son las tareas en las que se necesita un repaso de otra persona para poder seguir con la tarea, y realizar una comprobación del correcto funcionamiento que requería la tarea.

2.2.8. Diseño

Draw.io

Draw.io es una increíble herramienta que nos permite elaborar diagramas en línea sin necesidad de instalar absolutamente nada en nuestro PC. Su interfaz es bastante sencilla y fácil de utilizar. Dispone de una gran variedad de formas y diseños predeterminadas que luego podemos moldear a nuestro gusto [2].

En el proyecto, Draw.io ha sido utilizado para realizar el diagrama de casos de uso, el diagrama de actividades y el diagrama de clases.

Capítulo 3

Planificación del proyecto

En este capítulo, se explica la metodología a seguir durante el proyecto. Una vez explicada la metodología usada en CIDET, se explicará cómo se ha planificado el proyecto y que tareas se han llevado a cabo. Además, se realiza una estimación de recursos y costes.

3.1. Metodología

En CIDET se utiliza la metodología de desarrollo ágil SCRUM. Esta metodología escogida presenta el sprint, que tiene una duración de quince días aproximadamente. En cada sprint se recogen las distintas tareas que se han de desarrollar y se les asigna uno o varios miembros del equipo. Muchas veces estas tareas nacen a causa de las necesidades de los clientes.

Al finalizar cada sprint, el cliente puede echar un vistazo al progreso del proyecto y los objetivos que se han logrado hasta la fecha. Además, esto significa una gran ayuda para el progreso del proyecto, ya que la aportación de su *feedback* también puede servir para mejorar la funcionalidad del proyecto.

Los tres roles principales que propone la metodología SCRUM son los siguientes: *Product Owner*, *Scrum Master* y *Scrum Team*. También cuenta con otro rol auxiliar conocido como *Stakeholders*, que son las personas que hacen posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Sólo participan directamente durante las revisiones del sprint.

El *Product Owner* se encarga de que el *Scrum Team* trabaje de forma adecuada desde la perspectiva del negocio. Además, ayuda al usuario a escribir las historias de usuario y les otorga una prioridad. El *Scrum Master* elimina los obstáculos que impiden que el equipo alcance el objetivo del sprint. De esta forma se asegura de que el proceso Scrum se utiliza como es debido. El *Scrum Team* es el equipo de desarrollo que implementa el proyecto. Al tratarse de una empresa pequeña, para realizar este proyecto el *Product Owner* y el *Scrum Master* está formado por la misma persona, el supervisor. El *Scrum Team* estará formado por el alumno.

Para gestionar este método de trabajo, CIDET utiliza JIRA. Como se ha comentado anteriormente, JIRA es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos. También posee una pizarra de Kanban mediante la cual todo el equipo de trabajo puede ver de forma rápida y sencilla el estado de las tareas, así como modificarlas e incluirles comentarios.

3.2. Planificación

La planificación del proyecto, se ha dividido en tres etapas: comunicación, planificación y desarrollo.

En la etapa de comunicación, primero necesitaremos hacer un estudio del proyecto en el que vamos a trabajar. Una vez que hayamos hecho esto, podremos definir los objetivos y alcance, y organizar el grupo de trabajo y las reglas. Además, será necesario investigar el funcionamiento de aquellas tecnologías que vamos a utilizar, ya que algunas no se han utilizado con anterioridad.

En la etapa de planificación, una vez que se realiza el estudio del caso, es necesario dividir el proyecto en las tareas correspondientes de tal forma que se puedan indicar los predecesores o cuánto tiempo requiere cada tarea. Con esto se lleva a cabo una planificación del proyecto, donde se indica la funcionalidad que se espera obtener y se preparan todas las tareas a realizar.

Finalmente, en la etapa de desarrollo se realiza el diseño y la implementación del proyecto. Como estamos trabajando con una metodología ágil la duración de cada tarea no es precisa, ya que es posible que se vayan realizando modificaciones durante el proyecto. Una vez finalizadas las tareas, se procederá con una fase de pruebas para comprobar el correcto funcionamiento de todas las partes.

3.3. Estimación de recursos y costes del proyecto

En esta sección, se estima los recursos de los que dispone la empresa. Se estima que serán más que suficientes y por lo tanto no será necesario adquirir más recursos para este proyecto.

En CIDET, se estima el precio en aproximadamente 15€/hora de desarrollo.

Este proyecto se realiza de forma individual y tiene una duración de 300h de trabajo en la empresa. Por lo tanto, el coste de desarrollo del proyecto es:

$$\frac{15e}{h} \times 300h = 4500e$$

3.4. Seguimiento del proyecto

Como se sigue una metodología ágil para la realización del proyecto, cada dos semanas se lleva a cabo un sprint. En él se puede observar el estado de cada tarea, ya que es bastante importante ver su progreso y comentar cualquier duda cuando se hacen pequeñas reuniones con los miembros del equipo de trabajo.

Más adelante, en el capítulo 5, se explica detalladamente las tareas que se produjeron cada semana.

3.5. Desviaciones en la planificación

El proyecto no ha sufrido ningún tipo de desviación en las dos primeras etapas. Sin embargo, la etapa de desarrollo ha sufrido algunos cambios, ya que la creación de los controladores y las interfaces ha durado más tiempo del debido al sufrir variaciones a lo largo del proyecto.

A pesar de todo, el proyecto se ha podido llevar a cabo con éxito y se han cumplido los objetivos principales del proyecto.

Para poder observar mejor el tiempo invertido en cada etapa del proyecto, se ha realizado un diagrama de Gantt que podemos ver en la Figura 3.1.

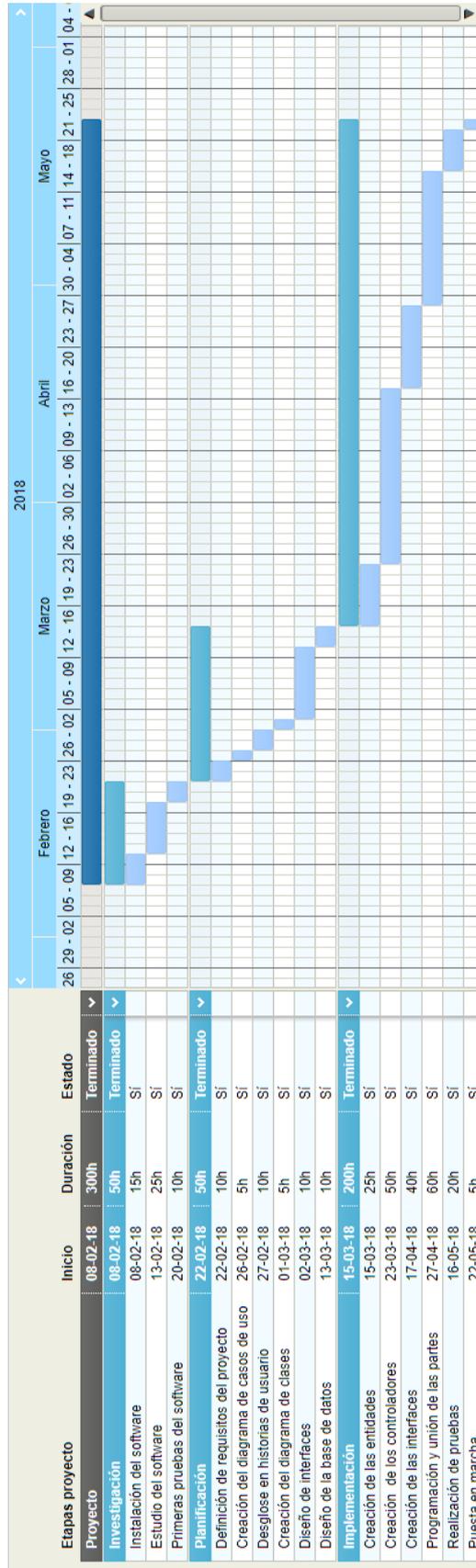


Figura 3.1: Diagrama de Gantt

Capítulo 4

Análisis y diseño del sistema

En este capítulo, se muestra el análisis realizado para el proyecto y el diseño elaborado para cada una de las interfaces que se presentan en el sistema de suscripciones.

Además, se explican los diferentes diagramas obtenidos y el diseño de la arquitectura del sistema.

4.1. Análisis del sistema

En la siguiente sección, se analizan los requisitos que intervienen a lo largo del proyecto.

En este caso, hemos separado los requisitos en tres tipos: funcionales, no funcionales y de datos.

4.1.1. Requisitos funcionales

Los requisitos funcionales representan toda la funcionalidad que el sistema tiene que proporcionar a los usuarios, y se corresponden con los casos de uso.

Cada caso de uso presenta una breve descripción y puede estar relacionado con otros casos de uso y requisitos.

A continuación, se muestra una lista de las acciones que debe seguir un usuario para llevar a cabo el requisito funcional.

Finalmente, se comentan las precondiciones y postcondiciones que se deben de cumplir. También hay que indicar el nivel de importancia que tiene el requisito.

CU01 - Gestionar suscripciones

| | |
|---------------------------|--|
| Descripción | El sistema debe permitir a un usuario acceder al listado de sus suscripciones. |
| Relaciones | CU04, CU06, CU07, CU08, RD01, RD02, RD03, RD04. |
| Secuencia de pasos | Acciones. 1 Pulsar el enlace de suscripciones. 2 El usuario en esa sección puede ver la lista de todas sus suscripciones. También dispone de un botón 'Información' para ver las condiciones legales de la suscripción. Además, tiene un enlace que le permite ver los pagos específicos para esa suscripción. |
| Excepción | 2.1 Si el usuario no tiene suscripciones aún, se le mostrará una página con las tarifas que ofrece Edueca. |
| Precondición | El usuario debe estar registrado en el sistema. |
| Postcondición | El usuario gestiona sus suscripciones. |
| Importancia | Alta. |
| Comentarios | Ninguno. |

Tabla 4.1: Requisito funcional «CU01. Gestionar suscripciones»

CU02 - Gestionar pagos

| | |
|---------------------------|--|
| Descripción | El sistema debe permitir a un usuario acceder al listado de sus pagos. |
| Relaciones | CU05, RD03, RD06, RD07. |
| Secuencia de pasos | Acciones. 1 Pulsar el enlace de suscripciones. 2 El usuario en esa sección puede ver la lista de todos sus pagos. Los pagos presentan 3 estados: Pagado, Pendiente y No pagado. Además, los usuarios tienen la oportunidad de descargar la factura si el pago ya se ha efectuado y está creada la factura. |
| Excepción | 2.1 Si el usuario no tiene pagos aún, se le mostrará una tabla vacía. |
| Precondición | El usuario debe estar registrado en el sistema. |
| Postcondición | El usuario gestiona sus pagos. |
| Importancia | Alta. |
| Comentarios | Aunque una suscripción expire o sea cancelada, se seguirán mostrando sus pagos en el panel. |

Tabla 4.2: Requisito funcional «CU02. Gestionar pagos»

CU03 - Gestionar datos de facturación

| | |
|---------------------------|---|
| Descripción | El sistema debe permitir a un usuario modificar sus datos de facturación. |
| Relaciones | RD05. |
| Secuencia de pasos | Acciones. <ol style="list-style-type: none">1 Pulsar el enlace de modificar datos de facturación.2 Rellenar el formulario con los datos correspondientes.3 Pulsar el botón de guardar. También tiene a su disposición el botón guardar y volver, y el botón cancelar y volver. |
| Excepción | <ol style="list-style-type: none">3.1 Si el usuario ha escrito algún dato incorrecto en el formulario, éste le advertirá del error. |
| Precondición | El usuario debe estar registrado en el sistema. |
| Postcondición | El usuario gestiona sus datos de facturación. |
| Importancia | Alta. |
| Comentarios | El usuario tiene en todo momento un botón 'Ayuda' por si se atasca en alguna parte del formulario. |

Tabla 4.3: Requisito funcional «CU03. Gestionar datos de facturación»

CU04 - Buscar suscripciones

| | |
|---------------------------|---|
| Descripción | El sistema debe permitir a un usuario buscar suscripciones concretas. |
| Relaciones | CU01, RD01. |
| Secuencia de pasos | Acciones. <ol style="list-style-type: none">1 Pulsar el enlace de suscripciones.2 Elegir el estado de sus suscripciones: activas, inactivas o todas.3 Especificar en el buscador, palabras clave que aparezcan en la suscripción que desea buscar. |
| Excepción | <ol style="list-style-type: none">3.1 Si no se encuentra ninguna suscripción con esas características, el panel indicará con un mensaje que no obtuvo ningún resultado. |
| Precondición | El usuario debe estar registrado en el sistema. El usuario debe tener al menos una suscripción. |
| Postcondición | El panel muestra todas las suscripciones que cumplan las características que el usuario introdujo. |
| Importancia | Media. |
| Comentarios | Ninguno. |

Tabla 4.4: Requisito funcional «CU04. Buscar suscripciones»

CU05 - Buscar pagos

| | |
|---------------------------|--|
| Descripción | El sistema debe permitir a un usuario buscar pagos concretos. |
| Relaciones | CU02, RD06, RD07. |
| Secuencia de pasos | Acciones. |
| | <ol style="list-style-type: none"> 1 Pulsar el enlace de suscripciones. 2 Elegir el estado de sus pagos: pagados, pendientes o todos. 3 Especificar en el buscador, palabras clave que aparezcan en el pago que desea buscar. |
| Excepción | |
| | <ol style="list-style-type: none"> 3.1 Si no se encuentra ningún pago con esas características, el panel indicará con un mensaje que no obtuvo ningún resultado. |
| Precondición | El usuario debe estar registrado en el sistema. El usuario debe tener al menos un pago. |
| Postcondición | El panel muestra todos los pagos que cumplan las características que el usuario introdujo. |
| Importancia | Media. |
| Comentarios | Ninguno. |

Tabla 4.5: Requisito funcional «CU05. Buscar pagos»

CU06 - Añadir suscripción

| | |
|---------------------------|---|
| Descripción | El sistema debe permitir a un usuario añadir una suscripción. |
| Relaciones | CU01, RD01, RD02, RD03, RD04, RD05. |
| Secuencia de pasos | Acciones. |
| | <ol style="list-style-type: none"> 1 Acceder al proyecto en el que se desea suscribir. 2 Elegir la tarifa que el usuario prefiera. 3 Pulsar el botón 'Aceptar' de la tarifa que se haya elegido. 4 Si se trata de una tarifa que requiere realizar un pago, el usuario deberá introducir los datos de facturación correspondientes antes de aplicarse la nueva suscripción. |
| Excepción | |
| | <ol style="list-style-type: none"> 4.1 Si el usuario ha escrito algún dato incorrecto en los datos de facturación, el sistema le advertirá del error. |
| Precondición | El usuario debe estar registrado en el sistema. |
| Postcondición | El usuario se suscribe al proyecto. |
| Importancia | Alta. |
| Comentarios | Por defecto, quedas suscrito al proyecto de Edueca gratuito cuando te creas una cuenta en el sistema. Más tarde, puedes modificarla si deseas adquirir la estándar o la pro. |

Tabla 4.6: Requisito funcional «CU06. Añadir suscripción»

CU07 - Modificar suscripción

| | |
|---------------------------|---|
| Descripción | El sistema debe permitir a un usuario modificar una suscripción. |
| Relaciones | CU01, RD01, RD02, RD03, RD04, RD05. |
| Secuencia de pasos | Acciones. <ol style="list-style-type: none">1 Pulsar el enlace de suscripciones.2 Elegir la suscripción que se quiere modificar.3 Pulsar el botón 'Modicar suscripción'.4 Elegir la tarifa del proyecto que el usuario prefiera.5 Pulsar el botón 'Aceptar' de la tarifa que se haya elegido.6 Si se trata de una tarifa que requiere realizar un pago, el usuario deberá introducir los datos de facturación correspondientes antes de aplicarse la nueva suscripción. |
| Excepción | |
| 6.1 | Si el usuario ha escrito algún dato incorrecto en los datos de facturación, el sistema le advertirá del error. |
| Precondición | El usuario debe estar registrado en el sistema. El usuario debe de estar suscrito al menos en una suscripción. |
| Postcondición | La antigua suscripción del usuario es modificada por la nueva. |
| Importancia | Alta. |
| Comentarios | Ninguno. |

Tabla 4.7: Requisito funcional «CU07. Modificar suscripción»

CU08 - Cancelar suscripción

| | |
|---------------------------|--|
| Descripción | El sistema debe permitir a un usuario cancelar una suscripción. |
| Relaciones | CU1. |
| Secuencia de pasos | Acciones. <ol style="list-style-type: none">1 Pulsar el enlace de suscripciones.2 Elegir la suscripción que se quiere modificar.3 Pulsar el botón 'Modicar suscripción'.4 En vez de elegir una nueva tarifa, pulsar el botón 'Cancelar suscripción' y confirmarlo. |
| Precondición | El usuario debe estar registrado en el sistema. El usuario debe de estar suscrito al menos en una suscripción. |
| Postcondición | La suscripción queda cancelada. |
| Importancia | Media. |
| Comentarios | El usuario también tiene la opción de ponerse en contacto con la empresa para que le anulen la suscripción. |

Tabla 4.8: Requisito funcional «CU08. Cancelar suscripción»

4.1.2. Requisitos no funcionales

Los requisitos no funcionales no describen información a guardar, ni funciones a realizar, sino características de funcionamiento. Por tanto, son las restricciones o condiciones que impone el cliente al programa que necesita.

RNF01 - Interfaz

| | |
|--------------------|--|
| Descripción | El sistema debe permitir que los usuarios puedan acceder al contenido de esta plataforma a través de los navegadores más usados, como por ejemplo: Chrome, Firefox, Explorer y Safari. |
| Comentarios | Ninguno. |

Tabla 4.9: Requisito no funcional «RNF01. - Interfaz»

RNF02 - Interfaz móvil

| | |
|--------------------|---|
| Descripción | El sistema debe permitir adaptar el diseño de las interfaces según la resolución de pantalla y tipo de dispositivo que los usuarios están utilizando. |
| Comentarios | Ninguno. |

Tabla 4.10: Requisito no funcional «RNF02. - Interfaz móvil»

RNF03 - Usabilidad

| | |
|--------------------|--|
| Descripción | El sistema debe facilitar la interacción de los usuarios con la plataforma, ya que muchos de ellos son personas mayores que carecen de conocimientos informáticos. |
| Comentarios | Se han incorporado botones de ayuda en algunas interfaces. |

Tabla 4.11: Requisito no funcional «RNF03. - Usabilidad»

RNF04 - Tiempo

| | |
|--------------------|---|
| Descripción | El sistema debe realizar todas las acciones disponibles para los usuarios lo más rápido posible e intentar cargar las páginas en menos de 5 segundos. |
| Comentarios | Ninguno. |

Tabla 4.12: Requisito no funcional «RNF04. - Tiempo»

RNF05 - Idiomas

| | |
|--------------------|---|
| Descripción | El sistema debe permitir a un usuario que tenga la opción de elegir el idioma. Los principales idiomas de la plataforma son el inglés y el español. |
| Comentarios | Ninguno. |

Tabla 4.13: Requisito no funcional «RNF05. - Idiomas»

RNF06 - Seguridad

| | |
|--------------------|---|
| Descripción | El sistema debe tomar medidas de protección y evitar posibles ataques que puedan ver afectados los datos de los usuarios o de la propia plataforma. |
| Comentarios | Ninguno. |

Tabla 4.14: Requisito no funcional «RNF06. - Seguridad»

4.1.3. Requisitos de datos

Los requisitos de datos describen los datos que el sistema debe manejar y almacenar. En cada uno de ellos se pueden observar sus atributos.

RD01 - Suscripción

| | |
|--------------------|---|
| Atributos | id, userDataId, pricelistId, promotionId, created, expires, deleted, paymentMethodId, active |
| Relaciones | RD02, RD03, RD04, RD05, RD06. |
| Comentarios | El valor del atributo active indica mediante un dato boolean si la suscripción sigue en pie o fue expirada/cancelada. |

Tabla 4.15: Requisito de datos «RD01. - Suscripción»

RD02 - Tarifa

| | |
|--------------------|---|
| Atributos | id, creatorUserId, modifierUserId, projectId, name, firstPayment, price, periodicity, startDate, endDate, created, modified, deleted, shortDescription, longDescription |
| Relaciones | RD01, RD03, RD04. |
| Comentarios | El valor de firstPayment es opcional. El valor de periodicity puede ser: Mensual, Trimestral, Semestral, Anual. Debe tener una descripción corta, pero es opcional poner descripción larga. |

Tabla 4.16: Requisito de datos «RD02. - Tarifa»

RD03 - Método de pago

| | |
|--------------------|---|
| Atributos | id, pricelistId, name, type, description, deleted, legalConditions |
| Relaciones | RD01, RD02. |
| Comentarios | La única forma de pago de momento es por TPV, pero más adelante contará con más formas, como por ejemplo: transferencia, domiciliación... |

Tabla 4.17: Requisito de datos «RD03. - Método de pago»

RD04 - Promoción

| | |
|--------------------|--|
| Atributos | id, pricelistId, beneficiaryUserId, name, description, code, discountType, value, startDate, endDate, created, modified, deleted |
| Relaciones | RD01, RD02. |
| Comentarios | Ninguno. |

Tabla 4.18: Requisito de datos «RD04. - Promoción»

RD05 - Datos de facturación

| | |
|--------------------|---|
| Atributos | id, userId, vatId, legalName, address, postalCode, region, country, city |
| Relaciones | RD01. |
| Comentarios | Son los datos que se necesitan del usuario si desea realizar una suscripción de pago. |

Tabla 4.19: Requisito de datos «RD05. - Datos de facturación»

RD06 - Requerimiento de pago

| | |
|--------------------|--|
| Atributos | id, subscriptionId, paymentMethodId, created, expires, price, isPaid, paymentHash, description, fileId |
| Relaciones | RD01, RD07. |
| Comentarios | Ninguno. |

Tabla 4.20: Requisito de datos «RD06. - Requerimiento de pago»

RD07 - Pago

| | |
|--------------------|---|
| Atributos | id, paymentRequirementId, numorder, dsMerchantIdentifier, dsExpiryDate, dsResponse, dsAuthorisationcode |
| Relaciones | RD06. |
| Comentarios | Ninguno. |

Tabla 4.21: Requisito de datos «RD07. - Pago»

4.2. Diagramas

En esta sección, vamos a analizar el diagrama de casos de uso, el diagrama de actividades y el diagrama de clases.

4.2.1. Diagrama de casos de uso

El diagrama de casos de uso (figura 4.1) cuenta con dos actores: Usuario registrado y Profesor. En este proyecto nos hemos enfocado más en la funcionalidad que se le otorga al usuario registrado. En una extensión del proyecto, se llevarían a cabo las acciones que puede realizar un profesor.

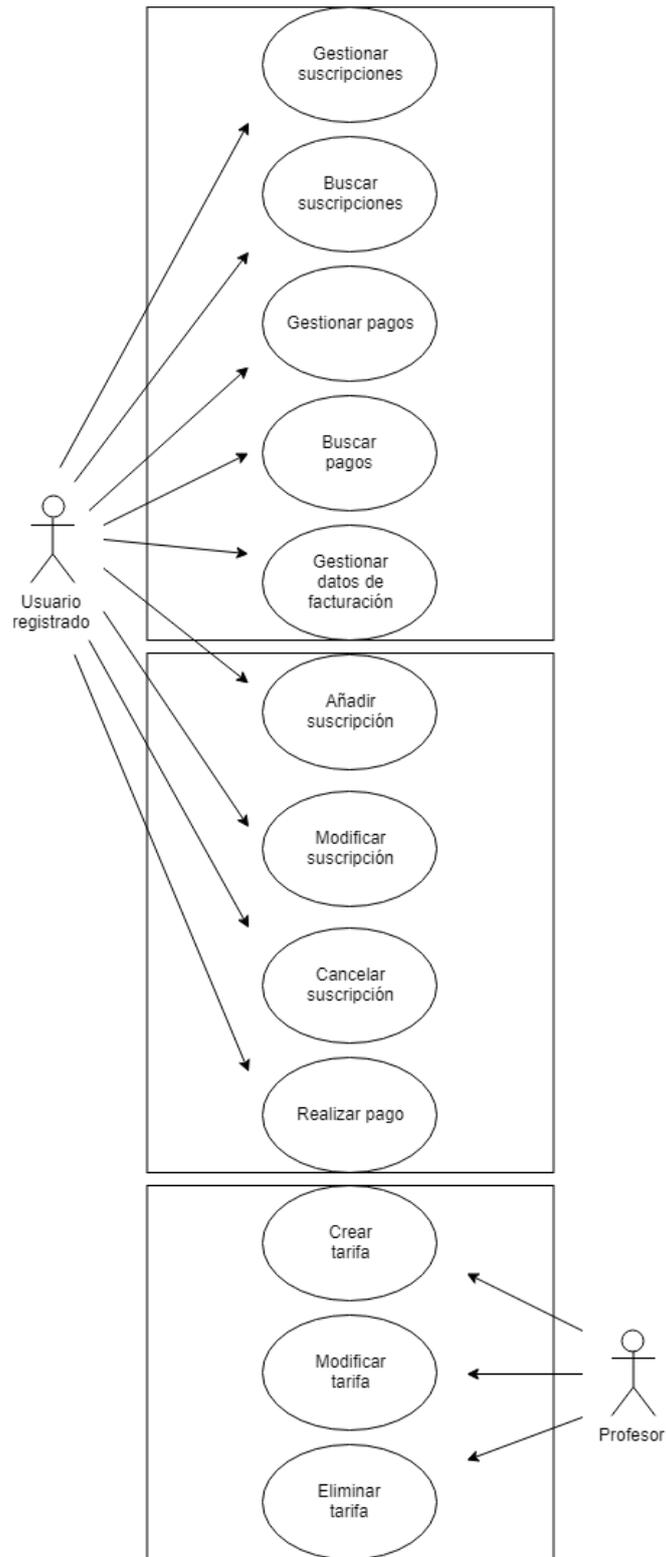


Figura 4.1: Diagrama de casos de uso

4.2.2. Diagrama de actividades

El diagrama de actividades (figura 4.2) indica todas aquellas acciones que un usuario tiene permitido realizar. Es esencial que el usuario esté registrado en el sistema, ya que es un requisito que se debe cumplir si se quiere acceder a la gestión del sistema de suscripciones.

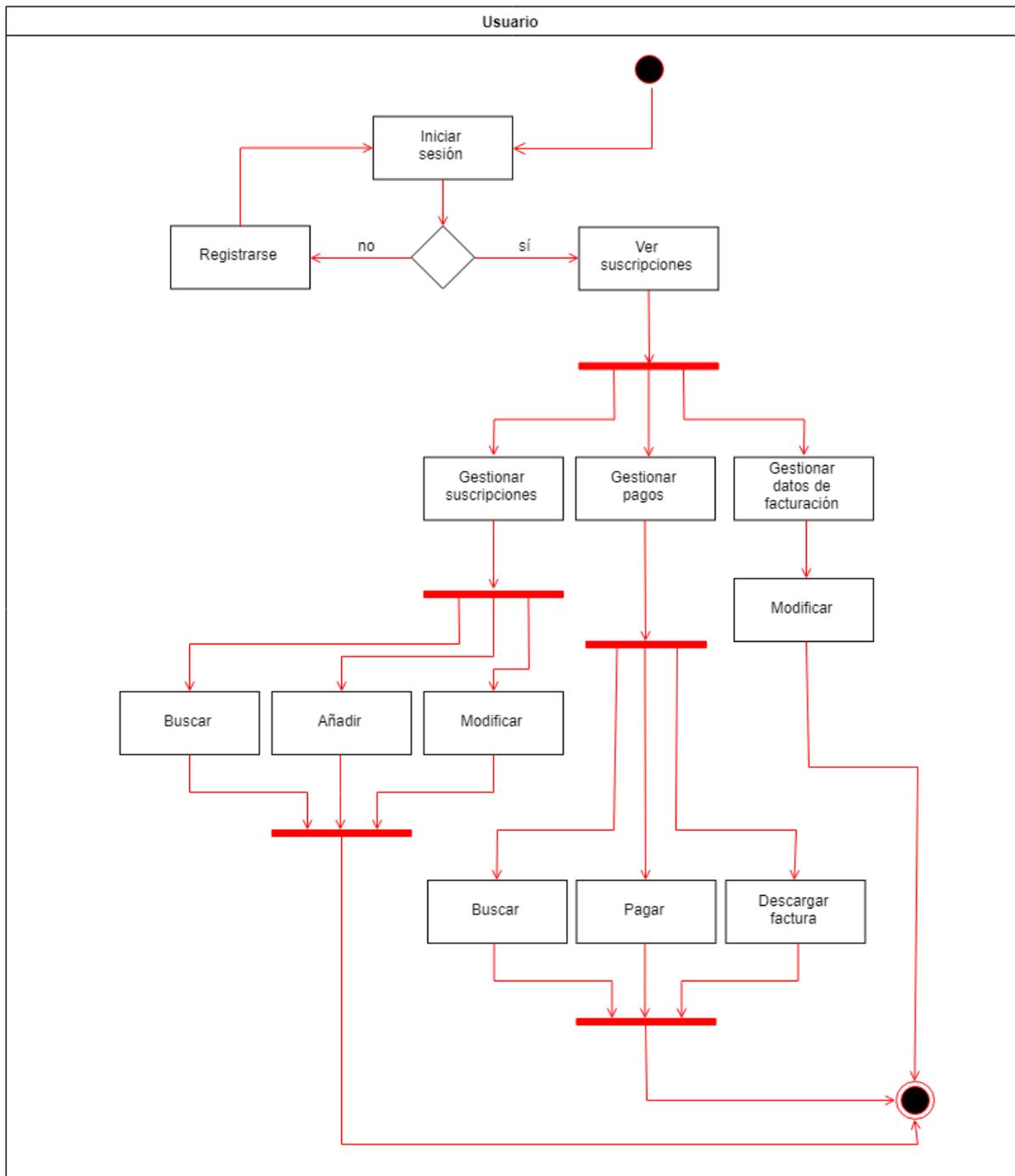


Figura 4.2: Diagrama de actividades

4.2.3. Diagrama de clases

El diagrama de clases (figura 4.3) muestra todas las clases con sus atributos que intervienen en el proyecto. La clases 'Usuario' y 'Proyecto' también están relacionados con el sistema de suscripciones, pero ya fueron implementados con anterioridad.

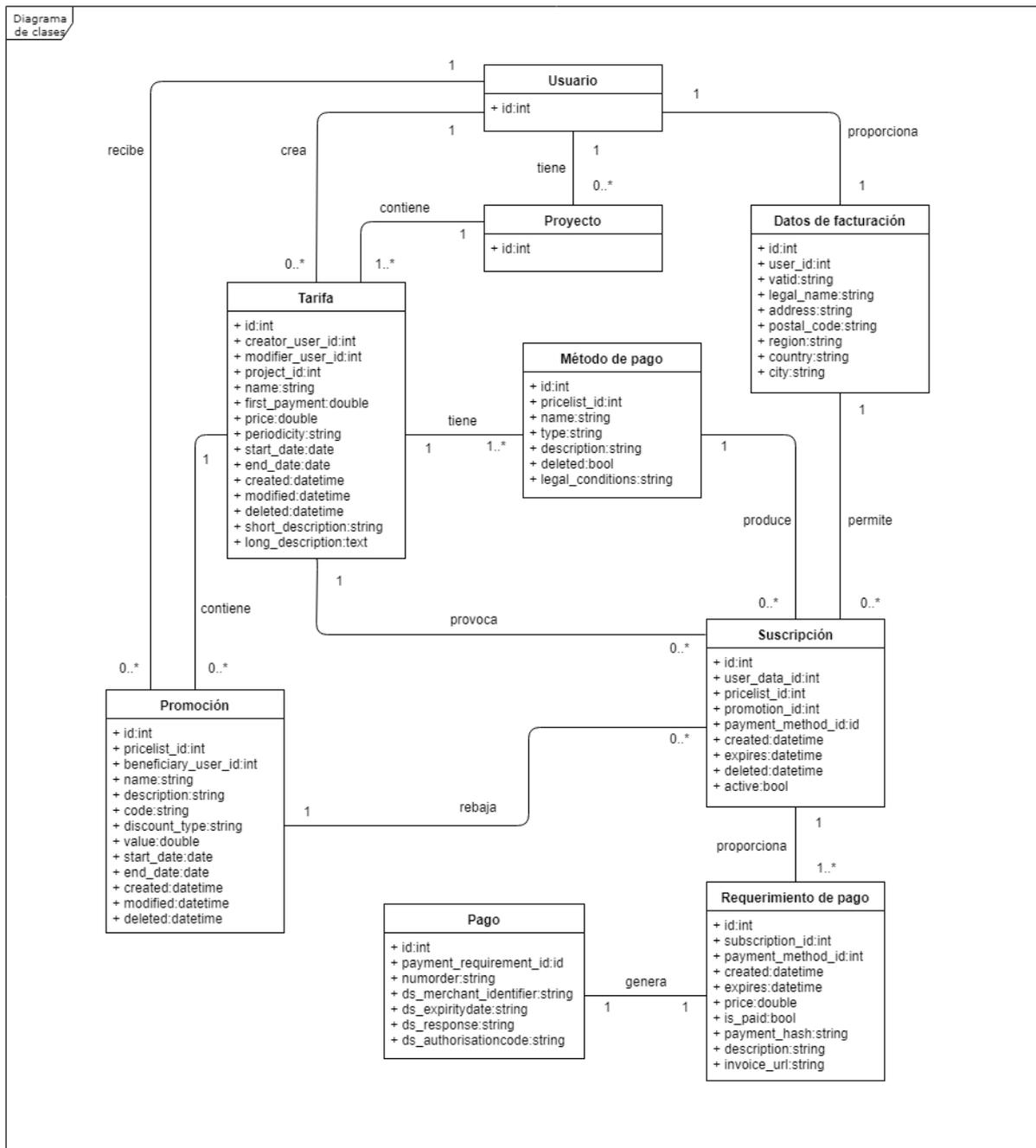


Figura 4.3: Diagrama de clases

4.3. Diseño de la arquitectura del sistema

Los usuarios pueden acceder a la plataforma web a través de múltiples dispositivos, ya que el CSS se encarga del diseño de la página.

Symfony, que se basa en el patrón Modelo Vista Controlador, ha sido el framework con el cual se ha desarrollado el proyecto.

La Figura 4.4 muestra todo el proceso desde que el usuario realiza una acción hasta que obtiene una respuesta.

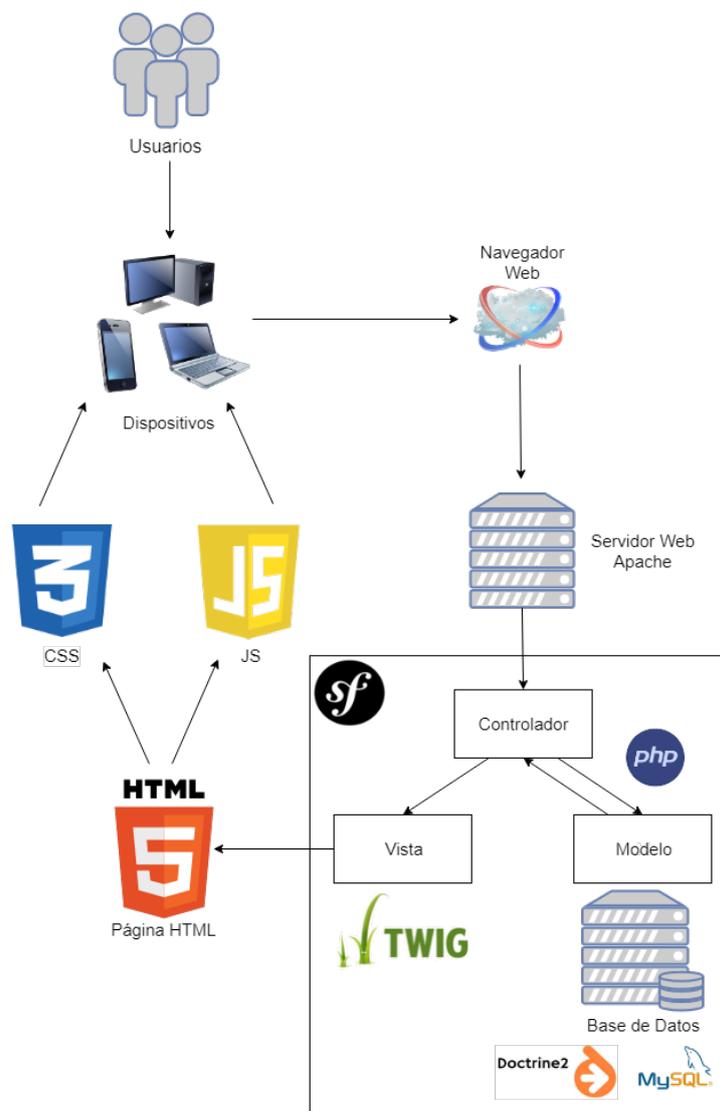


Figura 4.4: Arquitectura del sistema

4.4. Diseño de la interfaz

El diseño de la página principal del sistema de pagos se muestra en la Figura 4.5.

En la parte superior hay una sección que le indica al usuario sus datos de facturación. Si el usuario quiere realizar algún cambio, estos pueden ser modificados en cualquier momento.

Justo debajo de los datos de facturación, se encuentran los paneles de suscripciones y de pagos.

A la izquierda aparece el panel de suscripciones. En este panel un usuario puede gestionar sus suscripciones. Tiene a su disposición algunas opciones de búsqueda para el caso en el que solo quiera ver la suscripción que desea. Por un lado, tiene tres botones llamados: Todas, Activas e Inactivas. Por otro lado, tiene un buscador que encuentra suscripciones según la palabra clave que introduzca. Cada una de las suscripciones tienen un apartado donde se explica toda la información disponible de esa tarifa. Además, si se trata de una suscripción de pago, tendrá un enlace que le mostrará solo los pagos correspondientes a esa suscripción.

A la derecha aparece el panel de pagos. En este panel un usuario puede gestionar sus pagos. También presenta las mismas opciones de búsqueda que el panel de suscripciones, pero éste puede separar los proyectos según si están pagados o no. Se puede encontrar en tres estados: Pagado, Pendiente y No pagado. Una vez el pago se realice, se le generará una factura al cabo de un tiempo, que podrá ser descargada por el usuario.

En el caso de que un usuario tenga algún problema sobre el funcionamiento de la página, hay disponible un botón 'Ayuda' que le indica al usuario las acciones que se pueden realizar.

Para que pueda ser visible en la mayor cantidad de dispositivos, se decidió adaptar su interfaz cuando se accede desde un móvil. En la Figura 4.6 se puede observar que para ese caso todos los paneles se agrugarían en la misma columna.

Como hemos comentado antes, un usuario puede modificar sus datos de facturación. La Figura 4.7 muestra el formulario que debe rellenar el usuario si quiere modificar algún dato. Si el usuario se atasca en alguna parte, tiene a su disposición un botón de 'Ayuda' arriba del formulario.

Junto al formulario, hay implementados tres botones: Guardar, Guardar y volver, y Cancelar y volver. Además, este diseño de los botones servirá como estándar para los formularios que se creen más adelante.

Para esa misma interfaz, también se ha decidido adaptar su interfaz para dispositivos móviles. En la Figura 4.8 se puede observar que la principal diferencia es que los botones pasan a estar justo debajo del formulario. Además, los campos del formulario se adaptan correctamente para diferentes tamaños de pantalla.


GERARD - IDIOMA -

INICIO
PROYECTOS
BIBLIOTECA
RED
AYUDA

Datos de facturación Ayuda

| | | |
|-----------------------|-----------------------|---------------------------------|
| Nombre | Gerard Curto Llopis | Modificar datos |
| CIF/NIF/Código fiscal | 12345 | |
| Dirección | Calle 123 | |
| Ciudad | Castellón de la Plana | |
| Código postal | 12005 | |
| Región | Castellon | |
| País | Spain | |

Mis suscripciones

Todas **Activas** Inactivas

↻

| | | |
|------------|---------------------|---------------------------------------|
| Proyecto | English lessons | Modificar suscripción |
| Tipo | Inglés Demo | |
| Precio | 0.00 € | |
| Pagado por | Gratuito | |
| Iniciado | 16/05/2018 10:22:32 | Información |

| | | |
|--|---------------------|---------------------------------------|
| Proyecto | Edueca | Modificar suscripción |
| Tipo | Standard | |
| Precio | 27.00 € / Mensual | |
| Pagado por | Transferencia | |
| Iniciado | 05/04/2018 00:00:00 | Información |
| Click para ver los pagos | | |

| | | |
|--|---------------------------------|---------------------------------------|
| Proyecto | Curso online auxiliar admvo AGE | Modificar suscripción |
| Tipo | Subscripción mensual | |
| Precio | 75.00 € / Mensual | |
| Pagado por | Transferencia | |
| Iniciado | 01/03/2018 00:00:00 | Información |
| Click para ver los pagos | | |

Mis pagos

Todas **Pendiente** Pagado

↻

| | |
|-------------|---|
| Proyecto | Edueca |
| Estado | Pendiente |
| Precio | 27.00 € |
| Descripción | Pago disponible hasta el 8 de diciembre |

| | |
|-------------|-------------------------------------|
| Proyecto | Curso online auxiliar admvo AGE |
| Estado | No pagado |
| Precio | 75.00 € |
| Descripción | Pago disponible hasta el 8 de abril |

| | |
|----------|---------|
| Proyecto | Edueca |
| Estado | Pagado |
| Precio | 27.00 € |

| | |
|----------|---------------------------------|
| Proyecto | Curso online auxiliar admvo AGE |
| Estado | Descargar factura |
| Precio | 75.00 € |

1 - 3 de 3 registros. registros por página.

1 - 4 de 4 registros. registros por página.



Copyright © 2016 Centre for the Innovation and Development of Education and Technology
[privacy policy](#) | [about us](#) | [Contacta con nosotros en: edueca@edueca.com](#)



Figura 4.5: Página principal de suscripciones

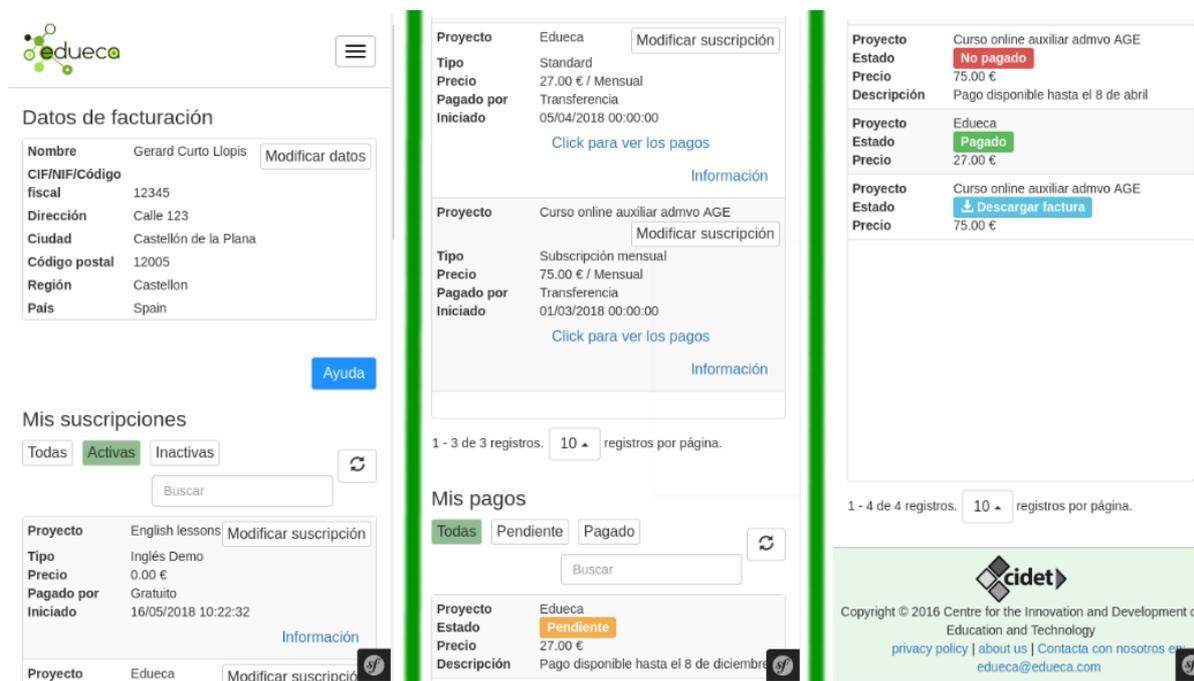


Figura 4.6: Página principal de suscripciones con pequeña resolución de pantalla

En el panel de suscripciones de la página principal, cada suscripción también tiene un botón llamado 'Modificar suscripción'. Una vez pulsado, el usuario es enviado a una página con el diseño de la Figura 4.9 si se trata de una suscripción de Edueca.

En la parte central, se puede ver las tres tarifas que ofrece Edueca con sus características y precio. Para no causar confusión, se añadió un símbolo que le notificaba al usuario la suscripción que tiene en ese momento, antes de realizar cualquier modificación. El usuario tiene la opción de continuar con la misma suscripción que tenía o cambiarla.

Si se trata de una suscripción de pago, se mostraría otra página en la que el usuario debería rellenar los datos necesarios para efectuar el pago. Por ahora, el único método de pago es por TPV.

Hay que destacar, que si un usuario desea cancelar su cuenta, debería ponerse en contacto con la empresa para realizar esa acción.

Si se trata de la modificación de una suscripción distinta de Edueca se mostraría la interfaz de la Figura 4.10. No es la versión definitiva de la página, ya que aún es necesario aplicar algunas modificaciones en su diseño.

La funcionalidad es la misma que tiene la interfaz comentada en la figura anterior.

Cada suscripción presenta el precio, tipo de tarifa, una descripción corta y una descripción larga, que es opcional. La descripción larga permanece oculta en el enlace 'Ver más...', que está debajo de la descripción corta.

Datos de facturación

Ayuda

| | |
|-------------------------------|--|
| Nombre: | <input type="text" value="Gerard Curto Llopis"/> |
| CIF/NIF/Código fiscal: | <input type="text" value="12345"/> |
| Dirección: | <input type="text" value="Calle 123"/> |
| Ciudad: | <input type="text" value="Castellón de la Plana"/> |
| Código postal: | <input type="text" value="12005"/> |
| Región: | <input type="text" value="Castellon"/> |
| País: | <input type="text" value="Spain"/> |

Guardar



Figura 4.7: Modificación de datos de facturación

Datos de facturación

 Ayuda

Nombre:

Gerard Curto Llopis

CIF/NIF/Código fiscal:

12345

Dirección:

Calle 123

Ciudad:

Castellón de la Plana

Código postal:

12005

Región:

Castellon

País:

Spain

 Guardar

 Guardar y volver

 Cancelar y volver



Copyright © 2016 Centre for the Innovation and Development of
Education and Technology

[privacy policy](#) | [about us](#) | [Contacta con nosotros en:
edueca@edueca.com](#)

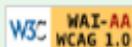


Figura 4.8: Modificación de datos de facturación con pequeña resolución de pantalla

Selecciona un tipo de suscripción

Desde esta página puedes cambiar tu suscripción a Edueca, por favor explora nuestras opciones de suscripción y disfruta utilizando Edueca.

| Plan | Precio | Características |
|----------|---------|---|
| Demo | 0€/mes | <ul style="list-style-type: none">Proyectos de 1 añoCreación de 1 proyectoSolo proyectos de curso20 estudiantes o miembrosDiseña y sube contenidosCapacidad máxima de almacenamiento: 50Mb |
| Estándar | 27€/mes | <ul style="list-style-type: none">Duración ilimitadaCreación de 10 proyectosProyectos de curso, centro y equipo200 estudiantes o miembrosDiseña y sube contenidosCapacidad máxima de almacenamiento: 2GB8 horas de clases virtuales o webinars |
| Pro | 95€/mes | <ul style="list-style-type: none">Duración ilimitadaCreación de cursos ilimitadaProyectos de curso, centro y equipo1000 estudiantes o miembrosDiseña y sube contenidosCapacidad máxima de almacenamiento ilimitadaClases virtuales o webinarsSoporte prioritarioSoluciones personalizadas... |

Figura 4.9: Selección de una suscripción de Edueca

Capítulo 5

Implementación

En este capítulo, se muestra todo el desarrollo del proyecto durante la estancia en la empresa. A continuación, se explica cada uno de los sprint con los que se desarrolló el proyecto.

5.1. Sprint 1

5.1.1. Planificación del Sprint

Al tratarse del primer sprint, se realizó una pequeña reunión para conocer el funcionamiento de los sprint y como se organizaban las tareas que tenían asignadas cada miembro del equipo. Una vez explicado, se creyó conveniente realizar una primera versión del diagrama de casos de uso, el diagrama de actividades y el diagrama de clases. De esta forma, tendremos claro desde el principio todo lo que interviene en el proyecto y que tareas hay que desarrollar para que se cumplan los objetivos. La Figura 5.1 muestra la planificación que se pretende seguir a lo largo del proyecto.

5.1.2. Ejecución del Sprint

Para empezar, fue fundamental la instalación de todo el software necesario para el proyecto que se debía desarrollar.

Además, la primera semana fue centrada en el estudio de las herramientas que no había usado con anterioridad, al igual que los nuevos lenguajes de programación.

La siguiente semana se dedicó a la realización de pequeñas pruebas, donde apliqué los conocimientos adquiridos del software nuevo.

También, se hizo una reunión en la que establecimos el proyecto que debería realizar durante mi estancia en la empresa.

| Id | Nombre de tarea | Tiempo (h.) | Dependencias |
|-----------|---------------------------------------|--------------------|---------------------|
| 1 | Investigación | 50 h | |
| 1.1 | Instalación del software | 15 h | |
| 1.2 | Estudio del software | 25 h | 1.1 |
| 1.3 | Primeras pruebas del software | 10 h | 1.2 |
| 2 | Planificación | 50 h | 1.3 |
| 2.1 | Definición de requisitos del proyecto | 10 h | 1.3 |
| 2.2 | Creación de diagrama de casos de uso | 5 h | 2.1 |
| 2.3 | Desglose en historias de usuario | 10 h | 2.2 |
| 2.4 | Creación de diagrama de clases | 5 h | 2.3 |
| 2.5 | Diseño de interfaces | 10 h | 2.4 |
| 2.6 | Diseño de la base de datos | 10 h | 2.5 |
| 3 | Implementación | 200 h | 2.6 |
| 3.1 | Creación de las entidades | 25h | 2.6 |
| 3.2 | Creación de los controladores | 50 h | 3.1 |
| 3.3 | Creación de las interfaces | 40 h | 3.2 |
| 3.4 | Programación y unión de las partes | 60 h | 3.3 |
| 3.5 | Realización de pruebas | 20 h | 3.4 |
| 3.6 | Puesta en marcha | 5 h | 3.5 |

Figura 5.1: Planificación del proyecto

5.2. Sprint 2

5.2.1. Planificación del Sprint

En el segundo sprint, he realizado dentro del proyecto Edueca las primeras implementaciones de código necesarias para poder almacenar los datos del sistema de suscripciones.

Durante este sprint, se ha empezado a trabajar en el caso de uso CU01.

| CU | Nombre | Prioridad | Estado |
|----|--------------------------------|-----------|---|
| 01 | Gestionar suscripciones | Alta |  |
| 02 | Gestionar pagos | Alta |  |
| 03 | Gestionar datos de facturación | Alta |  |
| 04 | Buscar suscripciones | Media |  |
| 05 | Buscar pagos | Media |  |
| 06 | Añadir suscripción | Alta |  |
| 07 | Modificar suscripción | Alta |  |
| 08 | Cancelar suscripción | Media |  |

Tabla 5.1: Pila del producto del Sprint 2

5.2.2. Ejecución del Sprint

Primera parte

La primera semana fue dedicada a la elaboración de todas las entidades necesarias que necesitaba el sistema de suscripciones para poder manejar todos los datos. Las nuevas entidades elegidas fueron: Suscripción, Tarifa, Promoción, Método de pago, Datos de facturación, Requisito de pago y Pago. Se estuvo pensando con los atributos que debía contener cada entidad. Una vez decidido, se empezó con la creación de todas las tablas en la base de datos. Únicamente hubo algunos problemas a la hora de decidir que relaciones había que crear entre las entidades, ya que también había que tener en cuenta las entidades que ya estaban creadas: Usuario y Proyecto. Además, se introdujeron algunos datos de prueba para comprobar su funcionamiento y poder insertar los primeros datos reales más adelante.

Gracias a un esbozo que se hizo en la pizarra sobre la organización del proyecto y las necesidades que tenía que abarcar, se entendía mejor la estructura que debían seguir las entidades y los datos almacenados en la base de datos.

Para tener todos los ficheros relacionados con el sistema de suscripciones en el mismo sitio se decidió crear un nuevo bundle llamado 'PayBundle'.

Segunda parte

La segunda semana fue dedicada los dos primeros días a la instalación de Linux Mint, ya que resultaba más cómodo para trabajar y Windows a veces daba problemas con el software que solía utilizar. En cuanto al aprendizaje del software que necesitaba Linux no hubo problema, ya que la mayoría de programas que se necesitaban para poder trabajar con el proyecto funcionaban de forma similar a Windows.

Una vez instalado y configurado, empecé con la creación de un controlador y la vista que muestra las suscripciones de un usuario a un curso. Al principio fue un poco difícil decidir el diseño que queríamos para esta página de suscripciones. Por un lado, intentábamos que fuese lo más comprensible posible para los usuarios que accediesen a ella. Por otro lado, buscábamos que cada suscripción detallase la información que podía resultar más útil para el usuario.

Además, hubo que realizar un pequeño estudio de los bootstrap para la creación de las tablas, ya que ofrecían una gran cantidad de diseños para algunas de las tablas que había que crear durante el proyecto y facilitaría a los clientes en su actividad en Edueca.

5.3. Sprint 3

5.3.1. Planificación del Sprint

En el tercer sprint, he realizado dentro del proyecto Edueca el diseño de la vista que le muestra a un usuario registrado sus suscripciones, requerimientos de pago y datos de facturación. Además, se empezó a realizar un formulario que le permite a un usuario modificar sus datos de facturación.

Durante este sprint, se ha seguido trabajando en el caso de uso CU01. Además, se ha empezado a trabajar con los casos de uso CU02 y CU03.

| CU | Nombre | Prioridad | Estado |
|----|--------------------------------|-----------|---|
| 01 | Gestionar suscripciones | Alta |  |
| 02 | Gestionar pagos | Alta |  |
| 03 | Gestionar datos de facturación | Alta |  |
| 04 | Buscar suscripciones | Media |  |
| 05 | Buscar pagos | Media |  |
| 06 | Añadir suscripción | Alta |  |
| 07 | Modificar suscripción | Alta |  |
| 08 | Cancelar suscripción | Media |  |

Tabla 5.2: Pila del producto del Sprint 3

5.3.2. Ejecución del Sprint

Primera parte

La primera semana fue dedicada a la inclusión de los campos que se debían mostrar en el panel de las suscripciones de un usuario. Se decidió, que junto al nombre de la suscripción, apareciese el tipo de tarifa, precio, método de pago y fecha de inicio. También se le proporcionó un nuevo estilo a la representación de las suscripciones, para que en vez de mostrar todos los datos de la suscripción en una fila, se mostrasen en bloques en la misma columna. De esta manera, las suscripciones resultaban ser más visibles para los usuarios.

Además, se comenzó con la creación del panel de requerimientos de pagos de un usuario, que está justo al lado del panel de suscripciones creado anteriormente. En este panel, también se decidió aplicar el mismo diseño.

Segunda parte

La semana siguiente fue dedicada a la realización de modificaciones en las tablas de suscripciones y requerimientos de pagos para que el usuario entienda mejor su funcionalidad.

A continuación se empezó con la creación, en la misma vista, de una sección que mostrase los datos de facturación de un cliente y que éste tenga la opción de modificarlos. En este caso, los datos de facturación que un usuario puede gestionar son: nombre, CIF/NIF, dirección, ciudad, código postal, región y país. En cuanto a la posición de esta sección, se decidió poner justo encima del panel de suscripciones.

Además, también se empezó a crear un formulario para que un usuario registrado pueda modificar sus datos de facturación. Al mismo tiempo, fue añadida la funcionalidad de ese formulario al controlador encargado de los datos de facturación de un usuario.

5.4. Sprint 4

5.4.1. Planificación del Sprint

En el cuarto sprint, se han llevado a cabo pequeñas modificaciones en el diseño de la tabla de suscripciones y la tabla de requerimientos de pago. Además, se les ha otorgado funcionalidad a algunos botones que contienen estas tablas.

Durante este sprint, se ha seguido trabajando en los casos de uso CU01, CU02 y CU03. Al no ocupar mucho tiempo, también se han completado los casos de uso CU04 y CU05. Además, se ha empezado a trabajar con los casos de uso CU06, CU07 y CU08.

| CU | Nombre | Prioridad | Estado |
|----|--------------------------------|-----------|---|
| 01 | Gestionar suscripciones | Alta |  |
| 02 | Gestionar pagos | Alta |  |
| 03 | Gestionar datos de facturación | Alta |  |
| 04 | Buscar suscripciones | Media |  |
| 05 | Buscar pagos | Media |  |
| 06 | Añadir suscripción | Alta |  |
| 07 | Modificar suscripción | Alta |  |
| 08 | Cancelar suscripción | Media |  |

Tabla 5.3: Pila del producto del Sprint 4

5.4.2. Ejecución del Sprint

Primera parte

La primera semana fue dedicada a la modificación de unos atributos de algunas entidades para que almacenen la fecha de cuando se elimina una suscripción, ya que no se desea borrar de la base de datos.

También se añadió un botón que muestra las condiciones legales que tiene una suscripción, y otro que permite realizar la modificación de una suscripción, ya sea para modificar la tarifa o cancelarla.

Durante esta semana se realizaron modificaciones del diseño del formulario creado la semana pasada de los datos de facturación de un usuario.

Además, en el panel de requerimientos de pago, se ha enlazado el botón de descargar factura con la url de esa factura (si dispone de una).

Segunda parte

La segunda semana fue dedicada a la implementación de la funcionalidad de los botones que permiten cancelar o modificar una suscripción.

Además, se añadieron tres botones: Todas, Activas e Inactivas. Estos tres botones refrescarán el panel de suscripciones y mostrarán aquellas que cumplan los requisitos. Del mismo modo, se crearon otros tres botones de búsqueda para el panel de pagos: Todos, Pagados y Pendientes.

Se hizo también una mejora del diseño de la información que aparece al dar clic en las condiciones legales de una suscripción.

Finalmente, se han creado varios modelos de botones para la interfaz de modificación de datos de facturación, ya que aún no está claro su diseño.

5.5. Sprint 5

5.5.1. Planificación del Sprint

En el quinto sprint, se han llevado a cabo algunas correcciones de errores que se producían al interactuar con las suscripciones. Además, se han realizado modificaciones en el diseño y funcionalidad de las suscripciones y datos de facturación.

Durante este sprint, se ha seguido trabajando en los casos de uso CU03, CU06 y CU07. Además, se completaron los casos de uso CU1, CU02 y CU8.

| CU | Nombre | Prioridad | Estado |
|----|--------------------------------|-----------|--------|
| 01 | Gestionar suscripciones | Alta | ✓ |
| 02 | Gestionar pagos | Alta | ✓ |
| 03 | Gestionar datos de facturación | Alta | ↻ |
| 04 | Buscar suscripciones | Media | ✓ |
| 05 | Buscar pagos | Media | ✓ |
| 06 | Añadir suscripción | Alta | ↻ |
| 07 | Modificar suscripción | Alta | ↻ |
| 08 | Cancelar suscripción | Media | ✓ |

Tabla 5.4: Pila del producto del Sprint 5

5.5.2. Ejecución del Sprint

Primera parte

La primera semana fue dedicada a la corrección del error que se producía cuando un usuario se suscribe a un curso y no tiene datos de facturación. Además, hubo que invertir un tiempo para decidir que pantalla mostrar al usuario cuando aún no tiene suscripciones. Para este caso, se decidió redirigirlo a la página que permite seleccionar una suscripción de Edueca.

También se decidió el diseño y posición de los botones de la interfaz de datos de facturación: en una pantalla grande aparecerán al lado derecho del formulario, y en una pantalla pequeña se pondrán justo debajo del formulario.

Una vez implementados los botones y la funcionalidad de modificación de una suscripción, se pasó a diseñar la interfaz con la que se encontraría un usuario.

Durante esta semana se crearon los ficheros css para cada una de las interfaces del proyecto.

Segunda parte

La semana siguiente fue dedicada a tomar medidas de seguridad para prevenir ataques de “Inyección SQL”.

En la base de datos, a parte del campo que indicaba la fecha de expiración de una suscripción, también se añadió un campo que indicaba si una suscripción seguía activa o no.

Además, se ha añadido a la página principal un botón de ayuda y se ha mejorado el diseño de las ventanas modales que se crearon la semana pasada.

Se hizo también unas modificaciones para que saliesen en el panel de suscripciones tanto las que tienen método de pago como las que no, pero finalmente se decidió que todas las suscripciones tienen métodos de pago, incluso las gratuitas, para que no sean un caso exclusivo.

5.6. Sprint 6

5.6.1. Planificación del Sprint

En el sexto sprint, se han llevado a cabo algunas medidas de seguridad al pasar parámetros por la url. Además, se ha creado una interfaz nueva que muestra las tarifas que ofrece un proyecto distinto a Edueca.

Durante este sprint, se han marcado como finalizados los casos de uso CU01, CU03, CU06, CU07 y CU08.

| CU | Nombre | Prioridad | Estado |
|----|--------------------------------|-----------|--------|
| 01 | Gestionar suscripciones | Alta | ✓ |
| 02 | Gestionar pagos | Alta | ✓ |
| 03 | Gestionar datos de facturación | Alta | ✓ |
| 04 | Buscar suscripciones | Media | ✓ |
| 05 | Buscar pagos | Media | ✓ |
| 06 | Añadir suscripción | Alta | ✓ |
| 07 | Modificar suscripción | Alta | ✓ |
| 08 | Cancelar suscripción | Media | ✓ |

Tabla 5.5: Pila del producto del Sprint 6

5.6.2. Ejecución del Sprint

Primera parte

La primera semana fue dedicada a corregir un error que podía resultar grave si no se tomaban medidas de seguridad. Anteriormente, un usuario podía modificar una suscripción que no fuese suya a partir de algunos parámetros de la url. Ahora, el usuario ya no podrá realizar esta acción y será enviado a una página de error explicándole el motivo.

También se ha creado la pantalla que muestra las tarifas que ofrece un proyecto distinto a Edueca. Junto a esta nueva pantalla, se ha añadido al controlador de suscripciones una nueva sección para gestionar las acciones de los usuarios con las tarifas.

La tarifa que crea el profesor tendrá una descripción corta (obligatoria) y/o una descripción larga (opcional). La descripción larga, si hay, permanecerá oculta y solo se le mostrará al usuario una vez pulse el botón "Ver más...", que está justo debajo de la descripción corta. La descripción larga funciona como un modal.

En la página de modificación de los datos de facturación de un usuario se han extraído fuera del formulario los botones que realizan las acciones: Guardar, Guardar y volver, y Cancelar y volver. Para llevar a cabo esta tarea, al pulsar los botones que están fuera del formulario, éstos llamarán a los botones que están dentro del formulario, que están ocultos. De esta manera, se obtiene más flexibilidad al no tener que depender del formato del formulario y puede servir como estándar para otros formularios que se creen en Edueca.

Segunda parte

La segunda semana fue dedicada a refactorizar código generado durante el proyecto, para facilitar la comprensión del código.

Además, durante esta semana se redactó parte de la memoria, donde se incluyeron los diagramas de casos de uso, actividades y de clases, que se realizaron a lo largo de la estancia en la empresa.

Capítulo 6

Extensiones del proyecto

La principal extensión que se debería llevar a cabo es la incorporación de una interfaz que le permita a un profesor crear las tarifas que necesite para su curso, siempre que tenga los permisos necesarios. Del mismo modo, un profesor siempre podría modificar y cancelar sus tarifas, pero sus alumnos deberían ser notificados de los cambios con bastante tiempo de antelación. El alumno debería ponerse en contacto con la empresa si tuviese algún problema con el profesor, para poder encontrar la mejor solución posible.

Otra extensión del proyecto es permitir que los usuarios puedan disponer de varias formas para realizar el pago de una suscripción que lo requiera. De esta manera, cuando un profesor crease una tarifa para su curso, también podría incluir las formas de pago que le puede ofrecer a sus alumnos.

Por último, pero no menos importante, tenemos guardados en la base de datos los datos que debe almacenar una promoción. Sin embargo, no ha sido implementado en el sistema, ya que primero se deseaba conseguir el buen funcionamiento de la gestión de las suscripciones. Por esta razón, los profesores aún no pueden ofrecerles promociones a los usuarios, pero se espera que puedan realizar esta acción en el futuro.

Capítulo 7

Conclusiones

En este capítulo, se exponen las conclusiones a nivel formativo y personal tras la realización del proyecto.

7.1. **Ámbito formativo**

A nivel formativo, he logrado adquirir bastante experiencia en el uso de lenguajes que había utilizado en otras asignaturas, como por ejemplo PHP y JavaScript. Además, he aprendido a manejar herramientas como Symfony, Twig y Doctrine, que nunca antes tuve ocasión de probar.

En cuanto a la metodología empleada, creo que ha sido un acierto utilizar SCRUM para la realización del proyecto. De esta manera, nos ha permitido seguir más de cerca los resultados que se obtenían cada semana, y las reuniones con otros compañeros de trabajo ha servido para resolver dudas y exponer propuestas.

7.2. **Ámbito personal**

A nivel personal, durante la estancia en la empresa me he sentido uno más del equipo, ya que desde el primer momento todos mis compañeros de trabajo me han estado explicando y ayudando mucho. Esto me ha dado seguridad a la hora de trabajar en el proyecto, y he obtenido mayor confianza en mí mismo con el paso de las semanas.

En definitiva, me he sentido muy a gusto realizando el proyecto gracias al agradable entorno de trabajo. Además, he aprendido a utilizar herramientas y lenguajes que creo que me resultarán de bastante utilidad en futuros trabajos y proyectos.

Bibliografía

- [1] CIDET. Web de cidet. <http://www.cidet.es/>. [Consulta: 12 de Abril de 2018].
- [2] Draw.io. Documentación draw.io. <https://about.draw.io/support/>. [Consulta: 6 de Junio de 2018].
- [3] Wikipedia. Apache. https://es.wikipedia.org/wiki/Servidor_HTTP_Apache. [Consulta: 31 de Mayo de 2018].
- [4] Wikipedia. Bitbucket. <https://es.wikipedia.org/wiki/Bitbucket>. [Consulta: 1 de Junio de 2018].
- [5] Wikipedia. Bootstrap. [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework)). [Consulta: 29 de Mayo de 2018].
- [6] Wikipedia. Css. https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada. [Consulta: 28 de Mayo de 2018].
- [7] Wikipedia. Doctrine. [https://es.wikipedia.org/wiki/Doctrine_\(PHP\)](https://es.wikipedia.org/wiki/Doctrine_(PHP)). [Consulta: 30 de Mayo de 2018].
- [8] Wikipedia. Git. <https://es.wikipedia.org/wiki/Git>. [Consulta: 1 de Junio de 2018].
- [9] Wikipedia. Html. <https://es.wikipedia.org/wiki/HTML>. [Consulta: 28 de Mayo de 2018].
- [10] Wikipedia. Javascript. <https://es.wikipedia.org/wiki/JavaScript>. [Consulta: 28 de Mayo de 2018].
- [11] Wikipedia. Jira. <https://es.wikipedia.org/wiki/JIRA>. [Consulta: 4 de Junio de 2018].
- [12] Wikipedia. Mysql. <https://es.wikipedia.org/wiki/MySQL>. [Consulta: 31 de Mayo de 2018].
- [13] Wikipedia. Phpstorm. <https://en.wikipedia.org/wiki/PhpStorm>. [Consulta: 2 de Junio de 2018].
- [14] Wikipedia. Symfony. <https://es.wikipedia.org/wiki/Symfony>. [Consulta: 29 de Mayo de 2018].
- [15] Wikipedia. Twig. [https://es.wikipedia.org/wiki/Twig_\(motor_de_plantillas\)](https://es.wikipedia.org/wiki/Twig_(motor_de_plantillas)). [Consulta: 30 de Mayo de 2018].

Anexo A

Estudio de Symfony

A.1. Twig

Twig es un motor de plantillas desarrollado para el lenguaje de programación PHP y su objetivo principal es ayudar a los desarrolladores de aplicaciones web que utilizan la arquitectura MVC.

A continuación, se muestra un ejemplo que muestra algunas características básicas de Twig que se han utilizado en bastantes ocasiones a lo largo del proyecto.

```
{% extends "base.html" %}
{% block proyectos %}
  <ul id="proyectos">
    {% for proy in proyectos %}
      <li>
        {# Muestra el nombre en mayusculas #}
        {{ proy.nombre|upper }}
        {% if proy.description == true %}
          {# Muestra la descripcion #}
          {{ proy.descripcion }}
        {% endif %}
      </li>
    {% endfor %}
  </ul>
{% endblock proyectos %}
```

Twig presenta tres clases de delimitadores:

- `{% ... %}`, se utiliza para ejecutar declaraciones, como pueden ser los bucles `for`.
- `{{ ... }}`, se utiliza para imprimir el contenido de variables o el resultado de evaluar una expresión.
- `{# ... #}`, se utiliza para añadir comentarios en las plantillas. Estos comentarios no son incluidos en la página renderizada.

A.2. Doctrine

Doctrine es una herramienta que permite asociar objetos a bases de datos relacionales.

Para poder trabajar con estos objetos se generan entidades. Estas entidades son clases de PHP que Doctrine relaciona con una tabla de la base de datos.

A continuación, se muestra un ejemplo de la creación de una clase.

```
// src/PayBundle/Entity/Subscription.php

namespace PayBundle\Entity;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity
 * @ORM\Table(name="subscription")
 */
class Subscription
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id ;

    /**
     * @ORM\Column(type="string", length=100)
     */
    private $name ;
}
```

Además, la clase creada tiene que tener los getters y setters de cada uno de sus atributos.

Una vez creada la entidad, será necesario actualizar la base de datos a través de la consola con el siguiente comando.

```
$ php bin/console doctrine:schema:update --force
```

Se pueden realizar varias acciones sobre los objetos de una entidad, como crear, obtener, actualizar y eliminar un objeto.

En el siguiente fragmento de código se muestra como se actualiza el objeto de una entidad.

```
$em = $this->getDoctrine()->getManager();
$subscription = $em->getRepository('Subscription')->find($id);

if (!$subscription) {
    throw $this->createNotFoundException(
        'No subscription found for id '.$id
    );
}

$subscription->setName('Ejemplo');
$em->flush();
```

La actualización de un objeto únicamente consta de tres pasos:

- Recuperar el objeto desde Doctrine.
- Modificar el objeto.
- Invocar a flush() en el gestor de la entidad.