



**Universidad de Valladolid**

# **Escuela de Ingeniería Informática**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática  
Mención en INGENIERÍA DE SOFTWARE

**Desarrollo de software educativo de apoyo a la  
docencia en la Teoría de Conjuntos.**

Autor:  
**Julio Gracia Gutiérrez**

Tutor:  
**María Felisa Pérez Martínez**



---

*A los amigos, que siempre están ahí cuando los necesito.*



# Agradecimientos

*“Si quieres ir rápido, camina solo; pero si quieres llegar lejos, camina acompañado”* (proverbio africano)

En primer lugar, querría dar las gracias a la Escuela y a sus profesores, por guiarme de la mejor forma que han podido a lo largo de los años que he estado aquí. También dar las gracias a mi tutora Marisa por ser un apoyo constante a lo largo del TFG.

Después, me gustaría dar las gracias a los miembros del GUI por ser amigos y constantes mentores, ayudándome y enseñándome cuando ha sido necesario.

Por último, agradecer a Andrés, Guille, Jose, Isma, Lobo, Victor y Provecho por estar siempre ahí y conseguir que estos años sean inolvidables. No dudo de que si a día de hoy estoy escribiendo estas palabras se debe a su apoyo, comprensión y ayuda.



# Resumen

En este Trabajo Fin de Grado se va a desarrollar un software de ayuda para la docencia de la asignatura “Matemática Discreta” del Grado en Ingeniería Informática de la Escuela de Ingeniería Informática de la Universidad de Valladolid.

El objetivo del presente trabajo es desarrollar un prototipo funcional de una aplicación web orientada a mejorar el aprendizaje y afianzar el conocimiento de los contenidos relativos a la Teoría de Conjuntos impartidos en dicha asignatura.

En este contexto será necesario plantear la arquitectura de la aplicación que se va a implementar. La solución escogida consiste en utilizar Angular programado en TypeScript para el front-end, Node.js programado en JavaScript y una base de datos relacional MariaDB para el back-end. Angular es una plataforma de desarrollo para aplicaciones web de TypeScript de código abierto. Node.js es un entorno en tiempo de ejecución multiplataforma asíncrono, de código abierto, para la capa del servidor. MariaDB es un sistema de gestión de bases de datos derivado de MySQL.



# Abstract

In the present thesis, a help software will be developed for the subject “Discrete Mathematics” in the Degree of Computer Science of the School of Computer Science of the University of Valladolid.

The aim of this project is to develop a functional prototype of an online application, oriented to the improvement in both, the processes of learning and of reinforcement of the contents related to the set theory given in the subject mentioned above.

At this stage, the contemplation of the application setting that is going to be implemented will be necessary. The chosen solution consists of using Angular, programmed in TypeScript for the front-end, Node.js programmed in JavaScript, and a relational MariaDB database for the back-end. Angular is a development platform for TypeScript online open-source model applications. Node.js is an open-source, cross-platform, run-time environment for the server-side. MariaDB is a database management system derived from MySQL.



# Índice general

<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VII</b>
<b>Lista de figuras</b>	<b>XIII</b>
<b>Lista de tablas</b>	<b>XVII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	1
1.2. Objetivo . . . . .	2
1.3. Contexto . . . . .	2
1.4. Solución adoptada . . . . .	2
1.5. Estructura de la memoria . . . . .	3
<b>2. Tecnologías utilizadas</b>	<b>5</b>
2.1. Definiciones previas necesarias . . . . .	5
2.2. Node.js . . . . .	7
2.2.1. Instalación . . . . .	7
2.3. NPM . . . . .	8

2.3.1.	Instalación: . . . . .	8
2.3.2.	Principales comandos utilizados: . . . . .	8
2.4.	Express . . . . .	9
2.4.1.	Instalación . . . . .	9
2.4.2.	Principales utilidades . . . . .	9
2.5.	Angular . . . . .	10
2.5.1.	¿Qué es un controlador? . . . . .	10
2.6.	Angular CLI . . . . .	12
2.6.1.	Instalacion: . . . . .	12
2.6.2.	Comandos relevantes utilizados: . . . . .	12
2.7.	Bootstrap 4 . . . . .	13
2.7.1.	Instalación . . . . .	13
2.8.	MariaDB . . . . .	14
2.8.1.	Instalación . . . . .	14
2.8.2.	Ventajas y deventajas de MariaDB frente MySQL . . . . .	14
<b>3.</b>	<b>Plan del proyecto</b>	<b>17</b>
3.1.	Vision general . . . . .	17
3.1.1.	Propósito, alcance y objetivos . . . . .	17
3.1.2.	Metodología utilizada . . . . .	17
3.1.3.	Evolución del plan . . . . .	18
3.2.	Gestión del proceso . . . . .	18
3.2.1.	Estimación . . . . .	18
3.2.2.	Plan de trabajo . . . . .	18
3.2.3.	Plan de Gestión de Riesgos . . . . .	22
3.2.4.	Presupuesto . . . . .	23

<b>4. Desarrollo</b>	<b>25</b>
4.1. Análisis . . . . .	25
4.1.1. Requisitos . . . . .	25
4.1.2. Casos de uso . . . . .	28
4.1.3. Modelo de dominio . . . . .	45
4.2. Diseño . . . . .	46
4.2.1. Diseño de la base de datos . . . . .	46
4.2.2. Despliegue . . . . .	47
4.2.3. Descomposición modular . . . . .	48
4.2.4. Diagrama de clases . . . . .	60
<b>5. Conclusiones</b>	<b>63</b>
5.1. Trabajo futuro . . . . .	63
<b>A. Manual de despliegue</b>	<b>65</b>
<b>B. Manual de usuario</b>	<b>69</b>
B.1. Manual del alumno: . . . . .	70
B.2. Manual del profesor: . . . . .	74
<b>C. Contenido del CD-ROM</b>	<b>81</b>
C.1. Directorio Documentation . . . . .	81
C.2. Directorio backend . . . . .	81
C.3. Directorio frontend . . . . .	81
<b>Bibliografía</b>	<b>81</b>



# Índice de figuras

2.1. Logo de Node.js . . . . .	7
2.2. Logo de NPM . . . . .	8
2.3. Logo de Express . . . . .	9
2.4. Logo de angular . . . . .	10
2.5. Google sobre TypeScript . . . . .	10
2.6. Logo de angular-cli . . . . .	12
2.7. Logo de Bootstrap 4 . . . . .	13
2.8. Logo de MariaDB . . . . .	14
4.1. Diagrama de casos de uso . . . . .	28
4.2. Modelo de dominio . . . . .	45
4.3. Diagrama relacional . . . . .	46
4.4. Diagrama de despliegue . . . . .	47
4.5. Diagrama de descomposición modular frontend . . . . .	48
4.6. Diagrama de descomposición modular frontend-core . . . . .	49
4.7. Diagrama de ejemplo de un componente . . . . .	50
4.8. Diagrama de ejemplo de un servicio . . . . .	51
4.9. Diagrama de descomposición modular frontend-login . . . . .	52
4.10. Diagrama de descomposición modular frontend-theory . . . . .	53

4.11. Diagrama de descomposición modular frontend-questions . . . . .	54
4.12. Diagrama de descomposición modular frontend-test . . . . .	55
4.13. Diagrama de descomposición modular frontend-statistics . . . . .	56
4.14. Diagrama de descomposición modular backend . . . . .	57
4.15. Diagrama de descomposición modular backend-src . . . . .	58
4.16. Diagrama de descomposición modular backend-model . . . . .	59
4.17. Diagrama de clases . . . . .	60
4.18. Diagrama de clases . . . . .	61
B.1. Pantalla de inicio de sesión . . . . .	69
B.2. Pantalla inicial de teoría . . . . .	70
B.3. Pantalla tras pulsar sobre un concepto . . . . .	71
B.4. Pantalla ver dudas . . . . .	71
B.5. Pantalla inicial de test . . . . .	72
B.6. Pantalla de cuestionario . . . . .	72
B.7. Pantalla de resolución . . . . .	73
B.8. Pantalla inicial de teoría del profesor . . . . .	74
B.9. Pantalla de dudas . . . . .	75
B.10. Pantalla inicial de test del profesor . . . . .	75
B.11. Pantalla de test del profesor . . . . .	76
B.12. Pantalla de test del profesor con duda desplegada . . . . .	76
B.13. Pantalla de añadir pregunta . . . . .	77
B.14. Pantalla de opción verdadera . . . . .	77
B.15. Pantalla de opción falsa . . . . .	77
B.16. Pantalla de principal de estadísticas . . . . .	78
B.17. Pantalla de estadísticas tras buscar . . . . .	78
B.18. Pantalla de detalle de los conceptos visitados . . . . .	78

B.19. Pantalla de detalle de las dudas planteadas . . . . .	79
B.20. Pantalla de detalle de los cuestionarios realizados . . . . .	79



# Índice de tablas

4.1. Descripción del caso de uso Ver elementos de teoría . . . . .	29
4.2. Descripción del caso de uso Ver elementos de teoría asociados . . . . .	30
4.3. Descripción del caso de uso Ver las dudas asociadas . . . . .	31
4.4. Descripción del caso de uso Añadir dudas . . . . .	32
4.5. Descripción del caso de uso Realizar un cuestionario . . . . .	33
4.6. Descripción del caso de uso Añadir elementos de teoría . . . . .	34
4.7. Descripción del caso de uso Ver estadísticas . . . . .	35
4.8. Descripción del caso de uso Gestionar dudas no resueltas . . . . .	36
4.9. Descripción del caso de uso Ignorar dudas . . . . .	37
4.10. Descripción del caso de uso Reportar dudas . . . . .	38
4.11. Descripción del caso de uso Responder dudas . . . . .	39
4.12. Descripción del caso de uso Ver las preguntas de cuestionario registradas en el sistema	40
4.13. Descripción del caso de uso Crear preguntas de cuestionario . . . . .	41
4.14. Descripción del caso de uso Borrar elementos de teoría . . . . .	42
4.15. Descripción del caso de uso Editar elementos de teoría . . . . .	43
4.16. Descripción del caso de uso Borrar dudas resueltas . . . . .	44



# Capítulo 1

## Introducción

### 1.1. Descripción del problema

Los alumnos que entran en el primer curso de una ingeniería experimentan grandes cambios a nivel académico. Esto es debido al uso de diferentes metodologías en la docencia y al aumento del volumen de contenidos tanto teóricos como prácticos en comparación con etapas anteriores. Algunas asignaturas parten de los conocimientos adquiridos durante los cursos de bachillerato pero otras tienen contenidos totalmente nuevos. La procedencia de los estudiantes es heterogénea (PAU y Módulos Superiores) y, por tanto, habrán tenido diferente grado de contacto con los contenidos de la titulación. En particular, la asignatura «Matemática Discreta», objeto de la aplicación desarrollada en este trabajo, incluye conceptos y resultados matemáticos que no han sido explorados con anterioridad.

Para tener una buena base en informática es necesario conseguir una buena base matemática. En el primer año del grado, los estudiantes pueden experimentar una falta de motivación en asignaturas de este tipo debido a la complejidad de las mismas y a que no consiguen encontrar una aplicación práctica real, ni son conscientes de su importancia tanto para asignaturas futuras como para su vida profesional.

La asignatura «Matemática Discreta» forma parte de las asignaturas de carácter básico del plan de estudios del Grado en Ingeniería Informática. Con ella se pretende ofrecer una formación básica y sólida al futuro ingeniero informático. Básica, en el sentido que los diferentes aspectos serán tratados a un nivel introductorio, y sólida, en el sentido de que los conocimientos adquiridos deben sentar las bases para desenvolverse en el resto de su formación académica y desarrollo profesional. Se trata de habilitar a los estudiantes para que adquieran las destrezas necesarias para seguir aprendiendo los aspectos de la Informática relacionados con la Lógica, Conjuntos, Relaciones y Combinatoria, que podrán aplicar en disciplinas como la Inteligencia Artificial, las Bases de Datos o la Estadística.

## 1.2. Objetivo

El objetivo final de este trabajo es proporcionar al alumno un complemento educativo en el estudio de la asignatura «Matemática Discreta» para el afianzamiento de algunos de los conocimientos que en ella se desarrollan. Para ello, se busca crear una plataforma que mejore la comunicación alumno-profesor y dote al profesor de herramientas que faciliten la distribución de los contenidos de la asignatura. La aplicación ha de facilitar al alumno el repaso de los conceptos teóricos más importantes de los temas relacionados con la Teoría de Conjuntos, la evaluación de su grado de aprendizaje a través de la realización de cuestionarios tipo test, así como el planteamiento de dudas al profesor. Además la aplicación recogerá estadísticas acerca de los conceptos consultados, dudas planteadas y test realizados por parte de los alumnos matriculados en la asignatura, que ayuden al profesor a detectar aquellos conceptos que representan mayor dificultad para el estudiante.

## 1.3. Contexto

La metodología docente actual de la asignatura [9] se basa en la combinación de sesiones teórico-prácticas en aula, sesiones prácticas en grupos reducidos y trabajo tanto grupal como individual. Concretamente se destinan 28 horas a las clases teórico-prácticas, 30 horas a las prácticas, 10 horas al estudio y trabajo grupal y 80 horas al estudio y trabajo individual. El presente trabajo se centra en este último apartado, ya que considero que es en el que el alumno puede sufrir mayores dificultades al no contar con la dirección del profesor ni con la ayuda de un grupo.

El desarrollo de la asignatura consiste en clases magistrales participativas, en las que se imparten conceptos y técnicas que deben ser asimilados con las horas de estudio individual. Gracias al afianzamiento de estos contenidos, el alumno será capaz de participar de forma activa en las sesiones prácticas y teóricas, así como superar satisfactoriamente las diversas pruebas escritas que se realizan a lo largo del curso. Todo ello influye positivamente en la calificación.

El material básico del que dispone el alumno es un documento pdf en el que se recogen los conceptos y resultados teóricos, ejemplos y ejercicios tipo que han de adquirirse para superar la asignatura.

## 1.4. Solución adoptada

Debido al tamaño del temario y a su complejidad, el estudio de esta asignatura puede resultar difícil. Tampoco ayuda el formato del material básico, ya que es un documento PDF que no cuenta con ningún marcador. Esto dificulta enormemente la búsqueda de un concepto concreto, así como la relación entre conceptos, ya que las únicas herramientas con la que los alumnos cuentan para ello son las palabras claves que están resaltadas en negrita y la función de búsqueda de palabras del editor de PDF utilizado.

Por otro lado, el procedimiento para plantear dudas requiere que el alumno se ponga en contacto con el profesor, ya sea presencialmente o mediante el correo electrónico, y el profesor resuelva

la duda de manera individual. La duda, por tanto, será respondida únicamente a nivel personal, por lo que es complicado que el resto de alumnos se enteren de las dudas que han planteado sus compañeros y de las soluciones que el profesor ha dado a dichas dudas. A su vez, esto puede conllevar que el profesor resuelva varias veces la misma duda.

Finalmente, es necesario algún sistema que permita al alumno saber de forma aproximada su nivel de preparación para ayudarlo a afrontar las etapas finales de la asignatura.

Lo que buscamos es una herramienta que nos permita localizar conceptos de forma rápida, entender las relaciones entre conceptos, reducir el tiempo malgastado tanto por parte del profesor como del alumno, evitando el planteamiento repetido de dudas ya resueltas con anterioridad y facilitando al alumno la resolución concreta de dudas y, por último, dotar al alumno de un mecanismo de comprobación de conocimientos adquiridos. Esta herramienta debe ser accesible por cualquier miembro de la comunidad educativa de la Universidad de Valladolid.

## 1.5. Estructura de la memoria

En base al esquema seguido para la realización del presente trabajo, se ha estructurado la memoria de la misma manera:

**INTRODUCCIÓN:** Se describe el porqué de este proyecto y los objetivos del mismo

**TECNOLOGÍAS UTILIZADAS:** Contiene información sobre las tecnologías escogidas y explicación de ciertos conceptos clave para entenderlas.

**PLAN DE PROYECTO:** En este capítulo se detalla la planificación del proyecto.

**DESARROLLO DEL PROYECTO:** La parte fundamental del proyecto es el desarrollo de un prototipo de plataforma web. En este capítulo se profundizará en el proceso de desarrollo, que incluye las etapas de análisis, desarrollo y despliegue.

**CONCLUSIONES Y RESULTADOS:** Tras realizar una serie de pruebas al prototipo, se presentarán las conclusiones y se analizarán los resultados así como posibles mejoras a tener en cuenta.



## Capítulo 2

# Tecnologías utilizadas

### 2.1. Definiciones previas necesarias

- **JavaScript:** JavaScript (abreviado como JS) es un lenguaje multi-paradigma, ligero e interpretado. [10]
- **TypeScript:** TypeScript es un superconjunto de JavaScript que permite el uso de clases, interfaces y tipado estático. El uso de variables tipadas de TypeScript aporta mayor robustez al código. [13, 16]
- **ECMAScript:** ECMAScript (abreviado como ES) es un lenguaje de scripting que forma la base de JavaScript. Está recogido en los estándares ECMA-262 y ECMA-402 de ECMA International. Al referirnos a ES5 nos referimos a la quinta versión del estándar ECMA-262, mientras que al hablar de ES2015 o ES6 nos referimos al estándar ECMA-262 en su sexta versión. Actualmente los navegadores soportan ES5, por lo cualquier lenguaje derivado de ECMAScript (como son JavaScript y TypeScript) deben de ser traspilado a ES5. [11]
- **Transpilar:** Transpilar es un término relativamente nuevo, proviene del inglés *transpiler*, fruto de la unión de las palabras *translate* y *compiler*. Es la operación de traducción de un lenguaje a otro, siendo ambos del mismo nivel de abstracción aproximadamente. [8, 2]
- **Gestor de paquetes:** Un gestor de paquetes mantiene un registro del software que está instalado en su ordenador, y le permite instalar software nuevo, actualizarlo a versiones más recientes, o eliminar software de una manera sencilla. Como su propio nombre sugiere, los gestores de paquetes gestionan paquetes: conjuntos de ficheros que se agrupan y que puede instalar y eliminar como conjunto. La labor de un gestor de paquetes es la de presentar una interfaz que asista al usuario en la tarea de administrar el conjunto de paquetes que están instalados en su sistema. [1]
- **Plataforma:** Una plataforma es un sistema que engloba los componentes, interfaces y librerías necesarios para permitir a los desarrolladores compilar, ejecutar y depurar sus aplicaciones.

- **Framework:** Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.
- **Open source:** La terminología open source incluye a aquellos softwares que cumplen los siguientes requisitos [14] :
  - **Distribución libre:** La licencia no restringirá a ninguna de las partes vender o regalar el software como un componente de un conjunto de software.
  - **Código fuente:** El programa debe incluir el código fuente sin ofuscar o dotar de un mecanismo para conseguirlo, preferentemente de forma gratuita mediante una descarga online.
  - **Trabajos derivados:** La licencia debe permitir modificaciones y trabajos derivados, y permitir que se distribuyan de forma libre.
  - **Integridad del código fuente del autor:** La licencia podría permitir no distribuir el código fuente del programa modificado si permite la distribución de parches. La licencia debe permitir explícitamente la distribución del software construido a partir de las fuentes modificadas.
  - **Sin discriminación:** La licencia no debe discriminar a ninguna persona ni colectivo.
  - **Para todos los ámbitos:** La licencia no debe restringir el uso en función del ámbito para el que se vaya a utilizar el software.
  - **Distribución de licencia:** Los derechos asociados al software deben aplicarse a todos los programas redistribuidos.
  - **La licencia no debe estar asociada a un producto:** Los derechos del software no deben depender del paquete de software en el que se distribuya
  - **La licencia no debe restringir a otros programas que se distribuyan junto a ella.**
  - **La licencia debe ser independiente de la tecnología utilizada.**
- **Back-end:** Término técnico para la capa de acceso a datos.
- **Front-end:** Término técnico para la capa de presentación de una aplicación. Conciene los componentes externos del sitio o aplicación web.
- **Propiedad:** [12] Una propiedad de un objeto puede ser explicada como una variable que se adjunta al objeto. Las propiedades de un objeto definen las características de un objeto. Un valor de propiedad puede ser una función, la cual es conocida entonces como un método del objeto.
- **Licencia GPL:** [3, 4] La licencia GPL o GNU GPL es una licencia copyleft. Esto es, un método general que requiere que todas las versiones modificadas y extendidas sean también libres.
- **API:** API significa interfaz de programación de aplicaciones (Application Programming Interface). Las APIs ofrecen una forma de estándar de dotar de funcionalidad a una aplicación, definiendo que funciones y métodos son accesibles.

## 2.2. Node.js



Figura 2.1: Logo de Node.js

Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome

### 2.2.1. Instalación

Para instalar node vamos a usar nvm. En este proyecto utilizaremos la versión **6.9.5**

```
curl -sL "
https://raw.githubusercontent.com/creationix/nvm/v0.32.0/install.sh" "
-o installnvm.sh
bash installnvm.sh
export NVMDIR="/root/.nvm"
[ -s "$NVMDIR/nvm.sh" ] && . "$NVMDIR/nvm.sh"
```

En este proyecto usaremos la versión **6.9.5**

```
nvm install v6.9.5
```

## 2.3. NPM



Figura 2.2: Logo de NPM

Npm es un gestor de paquetes que permite a los desarrolladores de JavaScript compartir y reutilizar código. Gracias a este gestor podemos conseguir de forma sencilla y actualizada las dependencias que va a tener nuestra aplicación. Este gestor de paquetes resulta esencial, ya que de él dependen Angular, Angular CLI y Node.js.

### 2.3.1. Instalación:

NPM es instalado junto con Node.js en el apartado anterior. Vamos a actualizarlo.

```
npm install -g npm
```

### 2.3.2. Principales comandos utilizados:

- `npm install`

Este comando instala todas las dependencias que hayamos declarado en nuestro fichero de configuración. En caso de especificar un nombre al final del comando instalaremos únicamente la dependencia nombrada. Este comando instala los paquetes elegidos, así como todos los paquetes de los que estos dependa. Usualmente se usará la opción `-save` que nos guarda la dependencia en nuestro fichero de configuración, de forma que podamos instalarla posteriormente.

- `npm uninstall [nombre]`

Este comando elimina la dependencia nombrada de nuestro sistema. Usualmente, se usará la opción `-save` para eliminarla también del fichero de configuración.

## 2.4. Express



Figura 2.3: Logo de Express

Express es un framework de desarrollo de aplicaciones web y APIs para Node.js

### 2.4.1. Instalación

```
npm install express
```

### 2.4.2. Principales utilidades

Nos facilita el manejo de:

- **Estáticos:** Ruta pública donde generalmente se alojan assets (CSS, imágenes, JS).

```
app.use(express.static(path.join(dirname, 'public')));
```

- **Controladores:** Encargados de controlar las peticiones http.

```
app.post('/login', function(req, res) -);
```

- **Sesiones y cookies:**

```
app.use(express.cookieParser('your secret here'));  
app.use(express.session());
```

## 2.5. Angular



Figura 2.4: Logo de angular

Angular es una plataforma open source de desarrollo de front-end desarrollado por Google. Está basado en componentes, que es una combinación de una plantilla HTML con un controlador.

Si bien está desarrollado con Javascript y permite el desarrollo con ES5 o superior, la comunidad de desarrolladores(incluyendo los responsables del proyecto de Google) prefiere utilizar TypeScript. [7]

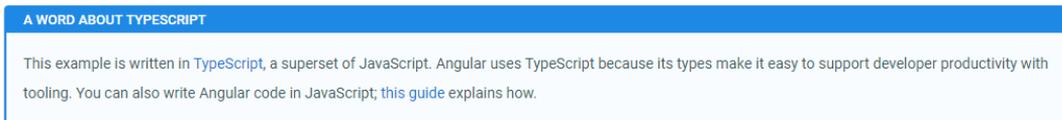


Figura 2.5: Google sobre TypeScript

### 2.5.1. ¿Qué es un controlador?

```
@Component(-
  selector: 'my-app',
  template: `<h1>Hello --name</h1>`
)
```

Todos los componentes inician con el decorador **Component** que describe cómo se comporta el componente. Entre otras opciones, define como incluir el componente en una página HTML, mediante la propiedad selector; y como está estructurado visualmente mediante la propiedad template. En el ejemplo dado, si quisiéramos utilizar el componente en una página HTML se usaría el elemento `<my-app>`, y el componente se ve como la frase «Hello `{{name}}`» dentro de un elemento de título 1.

En un desarrollo de mayor tamaño en lugar de definir la plantilla HTML en el propio componente, se usará la propiedad `templateUrl` para definir dónde buscar la plantilla HTML. De igual forma, se separarían los códigos CSS mediante la propiedad `styleUrls`. Ejemplo:

```
@Component(-
  selector: 'my-app',
  templateUrl: 'nombredelarchivo.html',
  styleUrls: ['nombredelarchivo1.css', 'nombredelarchivo2.css']
)
```

Siguiendo la trayectoria de AngularJs, precursor del actual Angular, conseguimos acceder a las propiedades del controlador mediante el uso de `{{ nombre_variable }}`. En este caso accederíamos a la propiedad `name`.

```
export class AppComponent - name = 'Angular';
```

Define la clase del controlador. En el caso de ejemplo, el controlador solo tiene la propiedad `name`, que contiene la cadena de texto «Angular»

## 2.6. Angular CLI



Figura 2.6: Logo de angular-cli

Angular CLI es una herramienta utilizada para inicializar aplicaciones Angular, desarrollar componentes y para tareas de mantenimiento asociadas a ello. [6]

### 2.6.1. Instalacion:

```
npm install -g @angular/cli
```

### 2.6.2. Comandos relevantes utilizados:

- `ng new [nombre]`  
Inicializa una nueva aplicación Angular con el nombre elegido.
- `ng serve`  
Transpila la aplicación y monta un servidor web.
- `ng generate class [nombre]`  
Genera una clase con el nombre escogido. Una clase es una construcción que permite crear tipos personalizados mediante la agrupación de variables de otras clases y comportamientos comunes.
- `ng generate component [nombre]`  
Genera un componente con el nombre escogido.
- `ng generate service [nombre]`  
Genera un servicio con el nombre escogido. Un servicio es una función, con sus propiedades y métodos que puede ser incluida, mediante inyección de dependencias, en los componentes. Gracias a esto, se pueden desarrollar funciones para tareas específicas, como es la comunicación con el servidor. Permite reutilizar las funciones de forma rápida entre componentes, así como acceder a variables compartidas entre ellos.

## 2.7. Bootstrap 4



Figura 2.7: Logo de Bootstrap 4

Bootstrap es un framework de HTML, CSS y JavaScript para el desarrollo de front-end. En su versión 4 incluye componentes de angular que utilizaremos en el presente proyecto.

### 2.7.1. Instalación

```
npm install bootstrap@4.0.0-alpha.6
```

## 2.8. MariaDB



Figura 2.8: Logo de MariaDB

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Está desarrollado por Michael Widenius (fundador de MySQL) y la comunidad de desarrolladores de software libre. Surgió a partir de la compra de Sun Microsystems por parte de Oracle para asegurar la existencia de una versión de MySQL con licencia GPL.

### 2.8.1. Instalación

```
sudo apt-get install software-properties-common
sudo apt-key adv --recv-keys --keyserver "
keyserver.ubuntu.com 0xF1656F24C74CD1D8
sudo add-apt-repository 'deb [arch=amd64] "
http://tedeco.fi.upm.es/mirror/mariadb/repo/10.2/debian"
stretch main'
sudo apt-get update
sudo apt-get install mariadb-server
```

### 2.8.2. Ventajas y desventajas de MariaDB frente MySQL

- **Ventajas:** [17, 5, 15]

- **Nuevos motores de almacenamiento más eficientes:**

Aria y XtraDB vienen a reemplazar a MyISAM e InnoDB respectivamente. Cabe destacar el mayor rendimiento de Aria, cuando recibe consultas complejas y tiene que realizar tablas temporales, éstas se cachean en memoria en vez de escribirlas en disco.

- **Estadísticas para índices y tablas:**

Esto puede ayudar para la optimización de la base de datos. Se añaden nuevas tablas de sistema para recoger esta información.

- **Mejoras en el rendimiento y la eficiencia con respecto a MySQL:**

Un ejemplo de esto es la eliminación o mejora de algunas conversiones no necesarias respecto a los juegos de caracteres.

- **Software libre:**

MariaDB está respaldada por la comunidad de software libre.

- **Desventajas:**

- **Coste migratorio:**

En líneas generales, MySQL está más extendido, por lo que utilizar MariaDB suele acarrear un coste migratorio de los datos. Sin embargo, MariaDB asegura tener total compatibilidad. En este proyecto no nos afectará en absoluto.



## Capítulo 3

# Plan del proyecto

El plan de desarrollo software es el documento que dirige la gestión de un proyecto software. Define las funciones técnicas y de gestión de proyectos, actividades y tareas necesarias para satisfacer los requisitos del proyecto. La finalidad de esta sección es conocer los puntos básicos de los que consta el proyecto, proporcionar los fundamentos en los que se basa y transmitir los aspectos básicos tal y como han sido entendidos y formulados. A continuación se describirá la visión general del proyecto donde se detalla su propósito, alcance y objetivos y la gestión del proceso donde se explica el coste estimado y la planificación de las fases principales e hitos del proyecto.

### 3.1. Vision general

#### 3.1.1. Propósito, alcance y objetivos

El objetivo del presente proyecto es desarrollar un prototipo de aplicación que sirva de apoyo para la asignatura de Matemática Discreta. La aplicación será un complemento educativo que permitirá al alumno repasar los conceptos teóricos más importantes, evaluar sus conocimientos mediante unos cuestionarios de tipo test y realizar consultas al profesor. Por parte del profesor permitirá introducir contenidos teóricos, resolver dudas, plantear preguntas para formar los cuestionarios y monitorizar el desempeño de los alumnos en la misma.

Toda esta información aparecerá de manera detallada en el apartado Análisis de Requisitos del siguiente capítulo

#### 3.1.2. Metodología utilizada

Se ha utilizado la metodología Kanban. Esta técnica se creó en Toyota, y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción. Los principios que rigen la

metodología son los siguientes:

- **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 basa 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
- **Flexibilidad:** Lo siguiente a realizar se decide del backlog (o tareas pendientes acumuladas), pudiéndose priorizar aquellas tareas entrantes según las necesidades del momento

Para su implementación se ha utilizado la herramienta provista por Github usando las siguientes columnas:

- **To Do:** El backlog de tareas
- **In Progress:** Las tareas que se estan haciendo actualmente
- **Done:** Las tareas ya realizadas

#### 3.1.3. Evolución del plan

El presente documento se revisará a lo largo del Trabajo de Fin de Grado y se irá actualizando conforme a los cambios que surjan.

## 3.2. Gestión del proceso

### 3.2.1. Estimación

Se considerará que la dedicación media al proyecto será de un total de 4 horas diarias con un solo recurso sin días de descanso. La fecha de inicio del proyecto es el 10/03/2017. Tras el estudio del plan de trabajo se estima que se requeriran aproximadamente 500 horas, por lo que se marca el hito de entrega del proyecto el 12/7/2017.

### 3.2.2. Plan de trabajo

- **Reunión con el cliente y documentación**
  - Recopilar información sobre el objetivo que persigue el cliente mediante entrevistas personalizadas, así como estudio de la documentación que este nos facilite.

- **Duración:** 7 días
- **Elicitación de requisitos**
  - Asegurar los requisitos extraídos de las entrevistas previas mediante la confirmación con el cliente. Este proceso se repetirá hasta conseguir extraer todos los requisitos que el cliente busca.
  - **Duración:** 7 días
- **Realización del diagrama de casos de uso y elaboración de un plan de trabajo provisional**
  - Realización de un plan de trabajo provisional para tener un marco de referencia a la hora de distribuir el tiempo. Para ello, se decidirán cuáles son los casos de uso y se hará el diagrama correspondiente, sin profundizar.
  - **Duración:** 2 días
- **Diseño de las vistas de la aplicación y elicitación con el cliente**
  - Diseño en papel de las vistas de la aplicación.
  - **Duración estimada:** 7 días
  - **Duración real:** 7 días
- **Estudio de las tecnologías actuales que nos permitan desarrollar el proyecto**
  - Estudio y comparación de las tecnologías actuales, su grado de adecuación al proyecto y su complejidad técnica, teniendo en cuenta los conocimientos previos de los que partimos.
  - **Duración estimada:** 2 días
  - **Duración real:** 2 días
- **Aprendizaje y práctica de las tecnologías escogidas**
  - Realización de ejemplos sencillos para comprobar de forma práctica los datos recopilados en el punto anterior.
  - **Duración estimada:** 2 días
  - **Duración real:** 2 días
- **Realización del plan de trabajo y estudio de los riesgos**
  - Basándonos en los conocimientos y destrezas adquiridos en el apartado anterior y en el plan provisional realizado con anterioridad, estudiar de los posibles riesgos y realizar planes de actuación para el caso de que se realicen. A mayores, se realiza un plan de trabajo de carácter definitivo.
  - **Duración estimada:** 2 días
  - **Duración real:** 2 días
- **Realización de la descripción en detalle de los casos de uso**
  - Basándonos en el diagrama realizado en los pasos anteriores, profundizaremos en cada uno de los casos de uso, definiendo su descripción en detalle.
  - **Duración estimada:** 4 días
  - **Duración real:** 4 días
- **Modelo de dominio y análisis de la base de datos**

- Se definirá el modelo de dominio de la aplicación y se diseñará la base de datos basándonos en el.
- **Duración estimada:** 2 días
- **Duración real:** 2 días
- **Realización básica de backend**
  - Programación de los elementos básicos y comunes a las aplicaciones Nodejs
  - **Duración estimada:** 4 días
  - **Duración real:** 4 días
- **Realización de las funcionalidades relacionadas con las sesiones**
  - Realización en profundidad del diseño relacionado con las funcionalidades y programación del frontend y el backend de las funcionalidades relacionadas con las sesiones, esto es, permitir al usuario acceder a la aplicación, que se muestre de forma distinta para alumnos y profesores y permitir al usuario salir de la aplicación
  - **Duración estimada:** 12 días
  - **Duración real:** 10 días
- **Realización de las funcionalidades relacionada con teoría**
  - Realización en profundidad del diseño de las funcionalidades y programación del frontend y el backend de las funcionalidades relacionadas con la teoría, esto es, mostrar la teoría organizada por temas, permitir la búsqueda de teoría por concepto y permitir al profesor añadir y editar conceptos.
  - **Duración estimada:** 15 días
  - **Duración real:** 20 días
- **Realización de las funcionalidades relacionadas con dudas**
  - Realización en profundidad del diseño de las funcionalidades y programación del frontend y el backend de las funcionalidades relacionadas con dudas, esto es, permitir al usuario ver las dudas ya preguntadas con anterioridad organizadas por conceptos, permitir generar nuevas dudas y permitir al profesor ver las dudas no resueltas y gestionarlas, ya sea reportándolas, ignorándolas o respondiéndolas.
  - **Duración estimada:** 15 días
  - **Duración real:** 19 días
- **Realización de las funcionalidades relacionadas con cuestiones**
  - Realización en profundidad del diseño de las funcionalidades y programación del frontend y el backend de las funcionalidades relacionadas con cuestiones, esto es, permitir al alumno generar cuestionarios según una serie de parámetros, resolverlos y conocer su resultado y permitir al profesor añadir y editar preguntas.
  - **Duración estimada:** 15 días
  - **Duración real:** 18 días
- **Realización de las funcionalidades relacionadas con estadísticas**
  - Realización en profundidad del diseño de las funcionalidades y programación del frontend y el backend de las funcionalidades relacionadas con cuestiones, esto es, permitir al alumno generar cuestionarios según una serie de parámetros, resolverlos y conocer su resultado y permitir al profesor añadir y editar preguntas.

- **Duración estimada:** 3 días
- **Duración real:** 3 días
- **Realización de las funcionalidades relacionadas con herramientas**
  - Realización en profundidad del diseño de las funcionalidades y programación del frontend y el backend de las funcionalidades relacionadas con herramientas que permitan poner en práctica los conocimientos de la asignatura.
  - **Duración estimada:** 10 días
  - **Duración real:** No realizado
  - **Motivo:** Las funcionalidades relacionadas con herramientas no fueron implementadas ya que, debido a una mala estimación de las funcionalidades anteriores, se descartó para cumplir el plazo de entrega.
- **Mejora de la interfaz**
  - Realización de una interfaz mas allá de la interfaz útil.
  - **Duración estimada:** 4 días
  - **Duración real:** 3 días
- **Realización de documentación del TFG**
  - Realización del presente documento, esto incluye los capítulos Introducción y Conclusiones y los anexos Manual de usuario y Manual de instalación, así como la maquetación del resto de capítulos.
  - **Duración estimada:** 7 días
  - **Duración real:** 9 días
- **Mejora de la interfaz**
  - Mejorar el aspecto de la aplicación
  - **Duración estimada:** 9 días
  - **Duración real:** No realizado
  - **Motivo:** La mejora de la interfaz no fuer implementadas ya que, debido a una mala estimación de las funcionalidades anteriores, se descartó para cumplir el plazo de entrega.

### 3.2.3. Plan de Gestión de Riesgos

La lista de riesgos expuesta a continuación tiene las siguientes características:

- **Impacto:** Los riesgos serán catalogados del 1 al 5, siendo 1 el riesgo menos peligroso y 5 el riesgo más peligroso.
- **Probabilidad:** Los riesgos serán catalogados del 1 al 5, siendo 1 un riesgo muy poco probable y 5 un riesgo muy frecuente.
- **Plan de protección:** Plan para evitar o minimizar la probabilidad.
- **Plan de contingencia:** Plan de solución para minimizar el impacto.

#### Riesgos

- **R-01 - Borrado de datos**
  - **Impacto:** 5
  - **Probabilidad:** 2
  - **Plan de protección:** Utilizar un sistema de control de versiones
  - **Plan de contingencia:** Restaurar la última versión
- **R-02 - Máquina personal averiada**
  - **Impacto:** El impacto se determinará en función del nivel de avería
  - **Probabilidad:** 1
  - **Plan de protección:** No hay
  - **Plan de contingencia:** En función del nivel de avería se detallan 3 posibles planes:
    - **Nivel bajo (Impacto 2):** Intentar arreglar la máquina
    - **Nivel medio (Impacto 3):** Intentar rescatar los datos
    - **Nivel alto (Impacto 5):** Restaurar la última versión en otra máquina y continuar el proyecto desde esta.
- **R-03 - Enfermedad no banal**
  - **Impacto:** 5
  - **Probabilidad:** 1
  - **Plan de protección:** No hay
  - **Plan de contingencia:** Replanificar teniendo en cuenta el tiempo disponible
- **R-04 - Fallo de software de terceros**
  - **Impacto:** 4
  - **Probabilidad:** 3
  - **Plan de protección:** Ninguno
  - **Plan de contingencia:** Se seguirán los siguientes pasos:
    1. Intentar solucionarlo
    2. Intentar esquivarlo

3. Intentar sustituirlo
4. Posponer la funcionalidad concreta hasta que este arreglado(esto puede acarrear que la funcionalidad no sea incluida)

■ **R-05 - Fallo en las etapas de análisis**

- **Impacto:** 5
- **Probabilidad:** 2
- **Plan de protección:** Prestar especial atención a la etapa de análisis y sobre todo a la elicitación de requisitos.
- **Plan de contingencia:** Replanificar teniendo en cuenta el tiempo disponible y la posibilidad de reutilización del proyecto generado hasta el momento.

■ **R-06 - Fallo en las etapas de diseño**

- **Impacto:** Variable. Se considera así puesto que un fallo en las etapas iniciales o en una funcionalidad aislada tendría un impacto bajo (2) pero aumenta según aumenta el número de funcionalidades afectadas.
- **Probabilidad:** 3
- **Plan de protección:** Inicialmente, realizar un boceto general del sistema. Posteriormente realizar siempre un estudio en profundidad del diseño relativo a la funcionalidad antes de implementarla.
- **Plan de contingencia:** Replanificar teniendo en cuenta el tiempo disponible y la posibilidad de reutilización del proyecto generado hasta el momento.

### 3.2.4. Presupuesto

Teniendo en cuenta que un analista programador cobra 10 euros por hora y la aplicación se ha planificado para 492 horas, se fija el coste de la mano de obra en 4920 euros.

Teniendo en cuenta que el coste de mi ordenador fue de 500 euros, los ordenadores portátiles tienen un período medio de vida útil de 4 años y el proyecto dura 4 meses, se fija el coste de hardware en 41,66 euros

**Presupuesto total:** 4961,66 euros



## Capítulo 4

# Desarrollo

### 4.1. Análisis

#### 4.1.1. Requisitos

##### Requisitos funcionales

- **RF-01: Mostrar contenidos teóricos**

La aplicación mostrará contenidos teóricos al alumno

- **RF-02: Buscar contenidos teóricos**

La aplicación permitirá buscar contenidos teóricos que contienen una palabra clave.

- **RF-03: Ver conceptos relacionados**

La aplicación permitirá ver que conceptos teóricos están relacionados con otros conceptos.

- **RF-04: Ver dudas relacionadas**

La aplicación permitirá ver que dudas han sido planteadas sobre un concepto

- **RF-05: Añadir dudas**

La aplicación permitirá añadir dudas sobre un concepto teórico

- **RF-06: Realizar cuestionarios**

La aplicación permitirá al alumno generar cuestionarios relacionados con un concepto, relacionados con un tema o generales, rellenarlo y recibir una corrección.

- **RF-07: Ver dudas no resueltas**

La aplicación permitirá al profesor ver las dudas no resueltas

- **RF-08: Resolver dudas**

La aplicación permitirá al profesor resolver dudas no resueltas, ya sea respondiéndolas o desechándolas

- **RF-09: Introducir contenidos teóricos**

La aplicación permitirá al profesor introducir conceptos teóricos

- **RF-10: Borrar contenidos teóricos**

La aplicación permitirá al profesor borrar contenidos teóricos

- **RF-11: Editar contenidos teóricos**

La aplicación permitirá al profesor editar contenidos teóricos

- **RF-12: Borrar dudas resueltas**

La aplicación permitirá al profesor borrar dudas ya resueltas

- **RF-13: Introducir preguntas de cuestionario**

La aplicación permitirá al profesor introducir preguntas de cuestionario

- **RF-14: Borrar preguntas de cuestionario**

La aplicación permitirá al profesor borrar preguntas de cuestionario

- **RF-15: Editar preguntas de cuestionario**

La aplicación permitirá al profesor editar preguntas de cuestionario

- **RF-16: Ver estadísticas de uso**

La aplicación permitirá al profesor ver las estadísticas de uso de la aplicación

#### **Requisitos no funcionales**

- **RNF-01: Adaptada a los navegadores más populares** La aplicación web estará adaptada a los navegadores más populares (Firefox, Safari, Internet Explorer 11 o Edge y Google Chrome).



4.1.2. Casos de uso

Diagrama de casos de uso

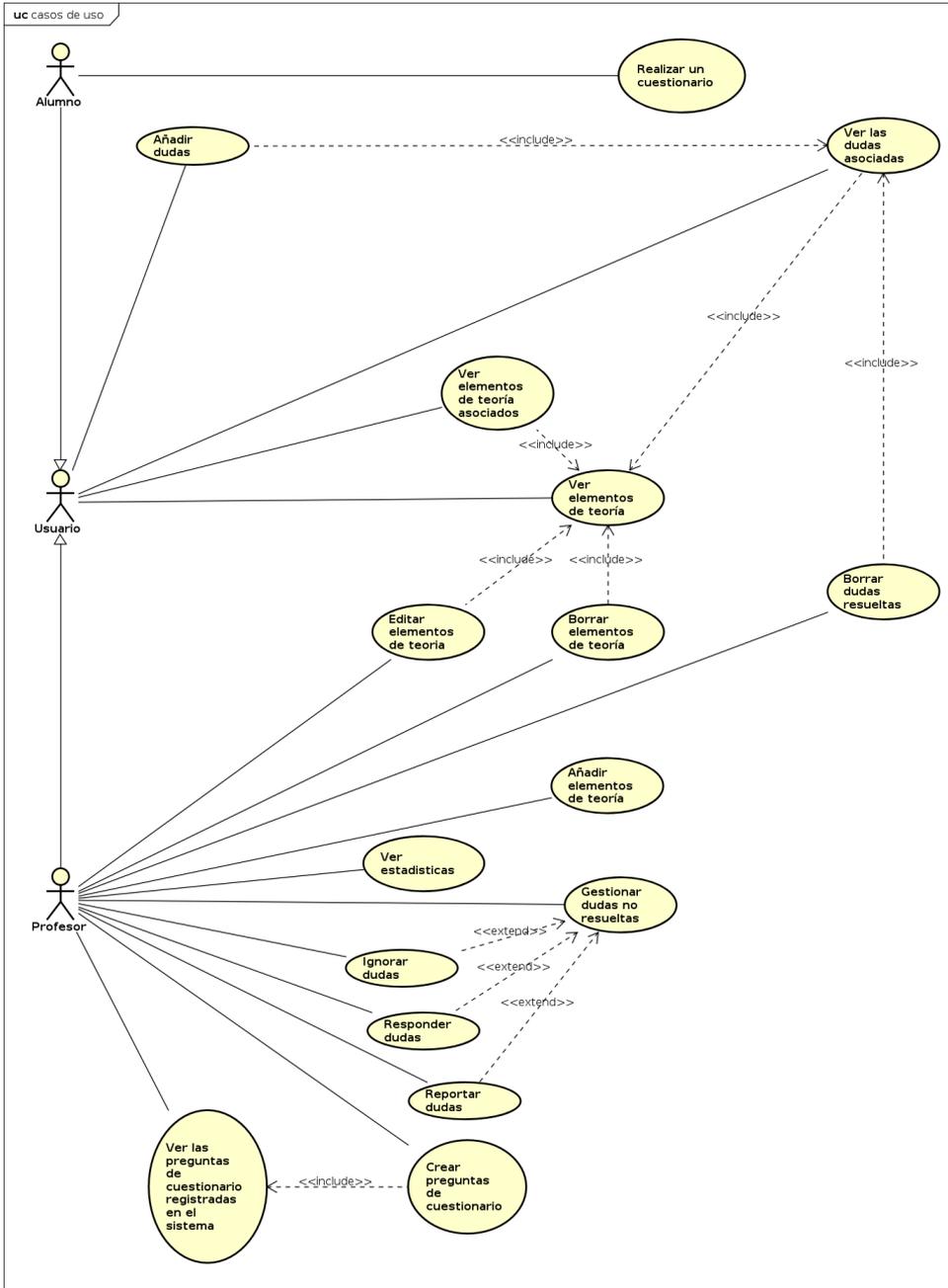


Figura 4.1: Diagrama de casos de uso

**Descripción de los casos de uso**

ITEM	VALUE
UseCase	Ver elementos de teoría
Actor	Usuario
Precondition	El usuario está identificado en el sistema
Postcondition	
Base Sequence	1 - El usuario elige la opción "Ver teoría". 2 - El sistema muestra el tema. 3 - El usuario elige el tema. 4 - El sistema muestra los títulos de los apartados. 5 - El usuario elige un apartado. 6 - El sistema muestra el contenido y las palabras clave relacionadas.
Branch Sequence	5a - El usuario rellena los filtros y el sistema muestra las titulos que coinciden con los filtros. El caso de uso continua por el paso 5.
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al usuario y el caso de uso queda sin efecto. 3a,5b - Si el usuario cancela, el caso de uso queda sin efecto.
Sub UseCase	

Tabla 4.1: Descripción del caso de uso Ver elementos de teoría

#### 4.1. ANÁLISIS

---

ITEM	VALUE
UseCase	Ver elementos de teoría asociados
Actor	Usuario
Precondition	El usuario esta identificado en el sistema. Se ha realizado el caso de uso "Ver elementos de teoría"
Postcondition	
Base Sequence	1 - El usuario elige una palabra clave relacionada. 2 - El sistema muestra los elementos de teoría relacionados.
Branch Sequence	
Exception Sequence	2a - Si el sistema no está en funcionamiento o genera algún error, avis a al usuario y el caso de uso queda sin efecto.
Sub UseCase	Ver elementos de teoría

Tabla 4.2: Descripción del caso de uso Ver elementos de teoría asociados

ITEM	VALUE
UseCase	Ver las dudas asociadas
Actor	Usuario
Precondition	El usuario esta identificado en el sistema. Se ha realizado el caso de uso "Ver elementos de teoria".
Postcondition	
Base Sequence	1 - El usuario escoge la opción "Ver dudas" 2 - El sistema muestra las dudas asociadas.
Branch Sequence	
Exception Sequence	2a - Si el sistema no está en funcionamiento o genera algún error, avisa al usuario y el caso de uso queda sin efecto.
Sub UseCase	Ver elementos de teoría

Tabla 4.3: Descripción del caso de uso Ver las dudas asociadas

ITEM	VALUE
UseCase	Añadir dudas
Actor	Usuario
Precondition	El usuario esta identificado en el sistema. El usuario ha realizado el caso de uso "Ver las dudas asociadas"
Postcondition	Se ha añadido una duda.
Base Sequence	1 - El usuario elige la opción "Añadir duda". 2 - El sistema pide los datos necesarios. 3 - El usuario rellena los datos necesarios. 4 - El sistema pide confirmación. 5 - El usuario confirma. 6 - El sistema registra la duda.
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al usuario y el caso de uso queda sin efecto. 3a,5a - Si el usuario cancela, el caso de uso queda sin efecto.
Sub UseCase	Ver las dudas asociadas

Tabla 4.4: Descripción del caso de uso Añadir dudas

ITEM	VALUE
UseCase	Realizar un cuestionario
Actor	Alumno
Precondition	El alumno está identificado en el sistema
Postcondition	El examen ha sido registrado
Base Sequence	1- El alumno entra en el apartado "Test". 2- El sistema muestra las opciones del apartado "Test". 3- El alumno elige el tipo de exámen que quiere realizar. 4- El sistema muestra el examen generado. 5- El alumno rellena el examen. 6- El sistema corrige el examen, registra los resultados y se los muestra al usuario.
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al alumno y el caso de uso queda sin efecto. 3a,5a - Si el alumno cancela, el caso de uso queda sin efecto.
Sub UseCase	

Tabla 4.5: Descripción del caso de uso Realizar un cuestionario

#### 4.1. ANÁLISIS

---

ITEM	VALUE
UseCase	Añadir elementos de teoría
Actor	Profesor
Precondition	El profesor esta identificado en el sistema.
Postcondition	Se ha añadido un elemento a la teoría
Base Sequence	1 - El profesor elige la opción "Añadir elemento de teoría". 2 - El sistema pide el tema, el título, el contenido y las palabras relacionadas. 3 - El profesor rellena los datos requeridos. 4 - El sistema pide confirmación de los datos. 5 - El profesor confirma los datos. 6 - El sistema guarda los datos.
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a,5a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	

Tabla 4.6: Descripción del caso de uso Añadir elementos de teoría

ITEM	VALUE
UseCase	Ver estadísticas
Actor	Profesor
Precondition	El profesor está identificado en el sistema.
Postcondition	
Base Sequence	1 - El profesor elige la opción "Ver estadísticas". 2 - El sistema pide el nombre de usuario sobre el que ver estadísticas. 3 - El profesor introduce el nombre de usuario sobre el que ver estadísticas. 4 - El sistema muestra las estadísticas requeridas
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al usuario y el caso de uso queda sin efecto. 3a - Si el usuario cancela, el caso de uso queda sin efecto.
Sub UseCase	

Tabla 4.7: Descripción del caso de uso Ver estadísticas

#### 4.1. ANÁLISIS

---

ITEM	VALUE
UseCase	Gestionar dudas no resueltas
Actor	Profesor
Precondition	El profesor esta identificado en el sistema
Postcondition	
Base Sequence	1 - El profesor escoge la opción "Ver dudas no resueltas". 2 - El sistema muestra las dudas no resueltas. 3 - El profesor elige una duda. 4 - El sistema muestra el contenido de la duda y las opciones. 5 - El profesor responde la duda. [Extension - Responder dudas]
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a,5a - Si el profesor cancela, el caso de uso queda sin efecto. 5b - Si la duda contiene contenido inapropiado, el profesor escoge la opción reportar. [Extension - Reportar duda] 5c - Si la duda está repetida, el profesor escoge la opción ignorar [Extension - Ignorar dudas]
Sub UseCase	

Tabla 4.8: Descripción del caso de uso Gestionar dudas no resueltas

ITEM	VALUE
UseCase	Ignorar dudas
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. Se ha realizado el caso de uso "Ver dudas no resueltas"
Postcondition	La duda esta marcada como ignorada.
Base Sequence	1 - El Profesor elige una duda y escoge la opción ignorar. 2 - El sistema pide confirmación. 3 - El profesor confirma. 4 - El sistema marca la duda como ignorada.
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	Gestionar dudas no resueltas

Tabla 4.9: Descripción del caso de uso Ignorar dudas

ITEM	VALUE
UseCase	Reportar dudas
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. Se ha realizado el caso de uso "Ver dudas no resueltas"
Postcondition	La duda esta marcada como reportada.
Base Sequence	1 - El Profesor elige una duda y escoge la opción reportar. 2 - El sistema pide confirmación. 3 - El profesor confirma. 4 - El sistema marca la duda como reportada.
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	Gestionar dudas no resueltas

Tabla 4.10: Descripción del caso de uso Reportar dudas

ITEM	VALUE
UseCase	Responder dudas
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. Se ha realizado el caso de uso "Ver dudas no resueltas"
Postcondition	La duda tiene una respuesta asociada
Base Sequence	1 - El Profesor elige una duda y escoge la opción responder. 2 - El sistema pide la respuesta. 3 - El profesor rellena la respuesta. 4 - El sistema pide confirmación. 5 - El profesor confirma. 6 - El sistema guarda la respuesta.
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a,5a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	Gestionar dudas no resueltas

Tabla 4.11: Descripción del caso de uso Responder dudas

ITEM	VALUE
UseCase	Ver las preguntas de cuestionario registradas en el sistema
Actor	Profesor
Precondition	El profesor está identificado en el sistema.
Postcondition	
Base Sequence	1 - El profesor escoge la opción Test. 2 - El sistema pide que se introduzca el tema. 3 - El profesor introduce el tema. 4 - El sistema muestra las preguntas asociadas a ese tema.
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	

Tabla 4.12: Descripción del caso de uso Ver las preguntas de cuestionario registradas en el sistema

ITEM	VALUE
UseCase	Crear preguntas de cuestionario
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. Se ha realizado el caso de uso "Ver las preguntas de cuestionario registradas en el sistema".
Postcondition	Se ha creado una nueva pregunta de cuestionario.
Base Sequence	1 - El profesor escoge la opción "Añadir pregunta". 2 - El sistema pide los datos necesarios. 3 - El profesor escribe los datos necesarios. 4 - El sistema pide confirmación de los datos. 5 - El profesor confirma. 6 - El sistema guarda la pregunta.
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a,5a- Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	Ver las preguntas de cuestionario registradas en el sistema

Tabla 4.13: Descripción del caso de uso Crear preguntas de cuestionario

#### 4.1. ANÁLISIS

---

ITEM	VALUE
UseCase	Borrar elementos de teoría
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. Se ha realizado el caso de uso "Ver elementos de teoria".
Postcondition	Se ha borrado un elemento de teoria
Base Sequence	1 - El profesor escoge la opción "Borrar" 2 - El sistema pide confirmación 3 - El profesor confirma 4 - El sistema borra el elemento de teoria
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	Ver elementos de teoría

Tabla 4.14: Descripción del caso de uso Borrar elementos de teoría

ITEM	VALUE
UseCase	Editar elementos de teoria
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. Se ha realizado el caso de uso "Ver elementos de teoria".
Postcondition	Se ha editado un elemento de teoría
Base Sequence	1 - El profesor escoge la opción "Borrar" 2 - El sistema pide los datos necesarios 3 - El profesor introduce los datos necesarios 4 - El sistema pide confirmación de los datos 5 - El profesor confirma 6 - El sistema modifica el elemento seleccionado
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a,5a - Si el profesor cancela, el caso de uso queda sin efecto.
Sub UseCase	Ver elementos de teoría

Tabla 4.15: Descripción del caso de uso Editar elementos de teoria

ITEM	VALUE
UseCase	Borrar dudas resueltas
Actor	Profesor
Precondition	El profesor esta identificado en el sistema. El profesor ha realizado el caso de uso "Ver las dudas asociadas"
Postcondition	Se ha borrado una duda resuelta
Base Sequence	1 - El profesor elige una duda y pulsa la opción borrar 2 - El sistema pide confirmación. 3 - El profesor confirma. 4 - El sistema borra la duda.
Branch Sequence	
Exception Sequence	2a,4a - Si el sistema no está en funcionamiento o genera algún error, avisa al profesor y el caso de uso queda sin efecto. 3a - Si el usuario cancela, el caso de uso queda sin efecto.
Sub UseCase	Ver las dudas asociadas

Tabla 4.16: Descripción del caso de uso Borrar dudas resueltas

4.1.3. Modelo de dominio

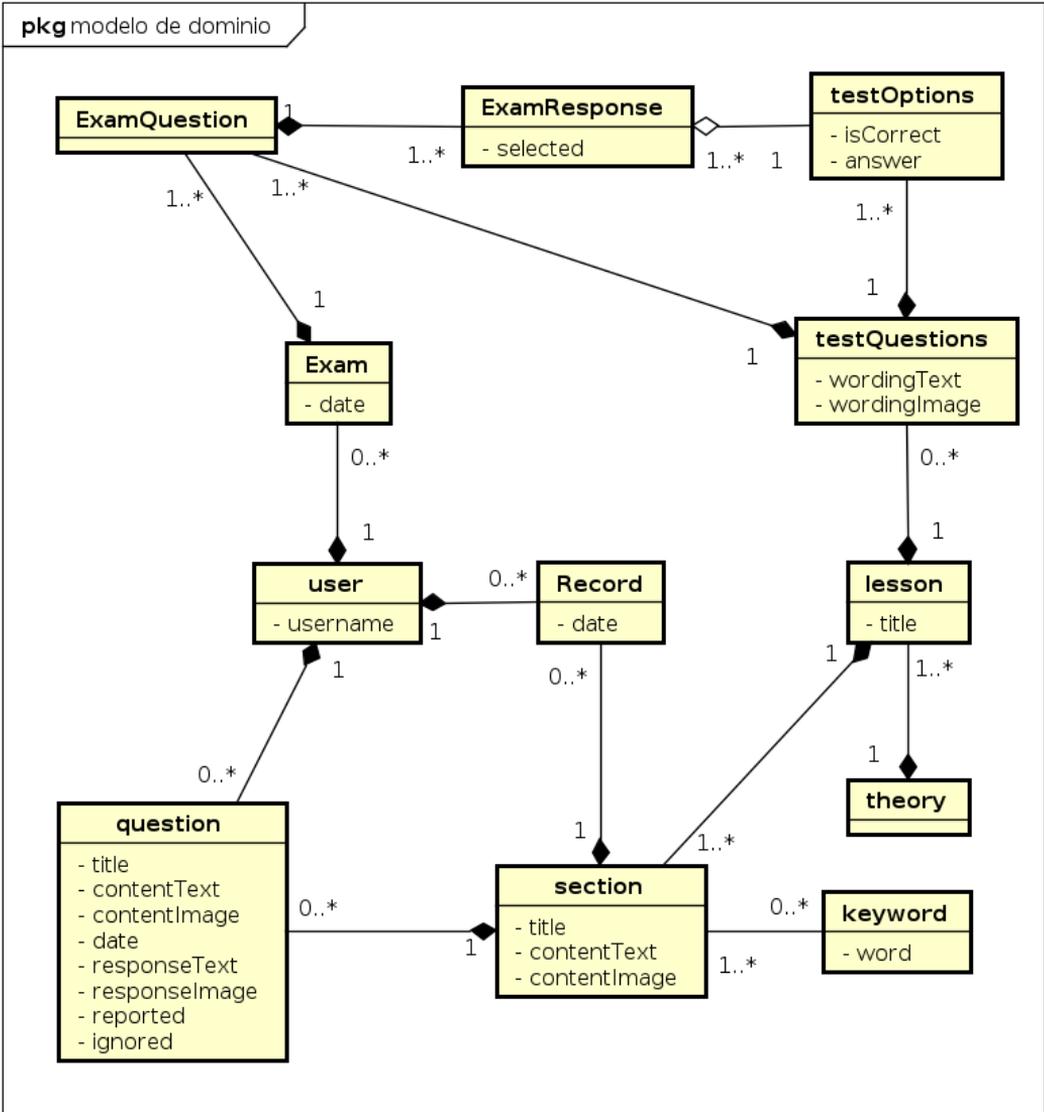


Figura 4.2: Modelo de dominio



### 4.2.2. Despliegue

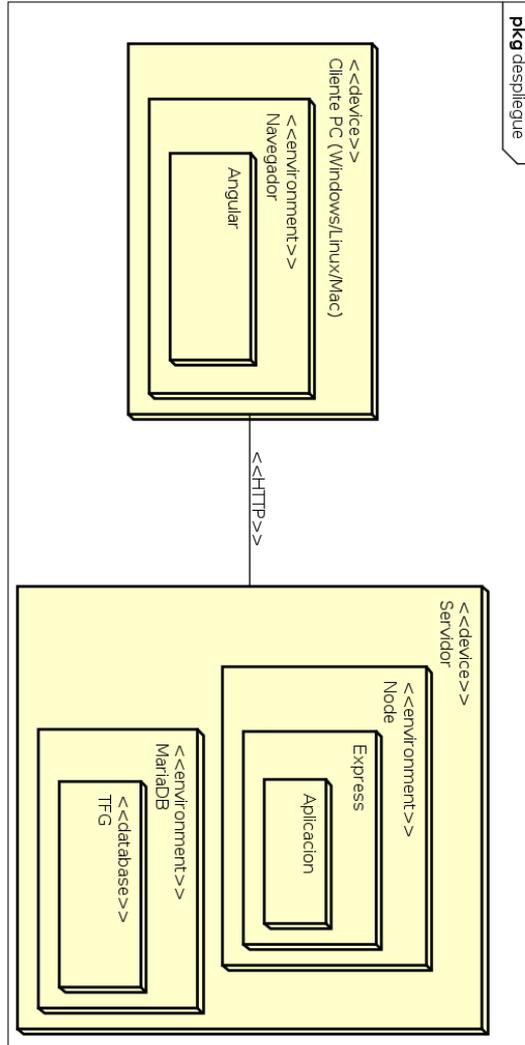


Figura 4.4: Diagrama de despliegue

### 4.2.3. Descomposición modular

#### Front-end

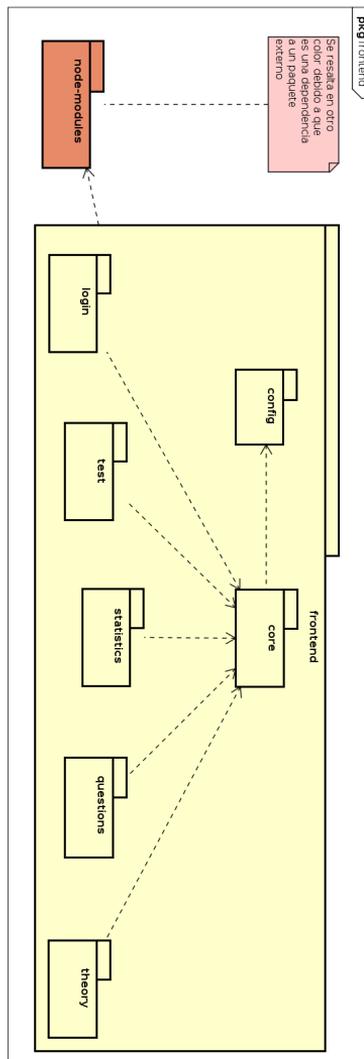


Figura 4.5: Diagrama de descomposición modular frontend

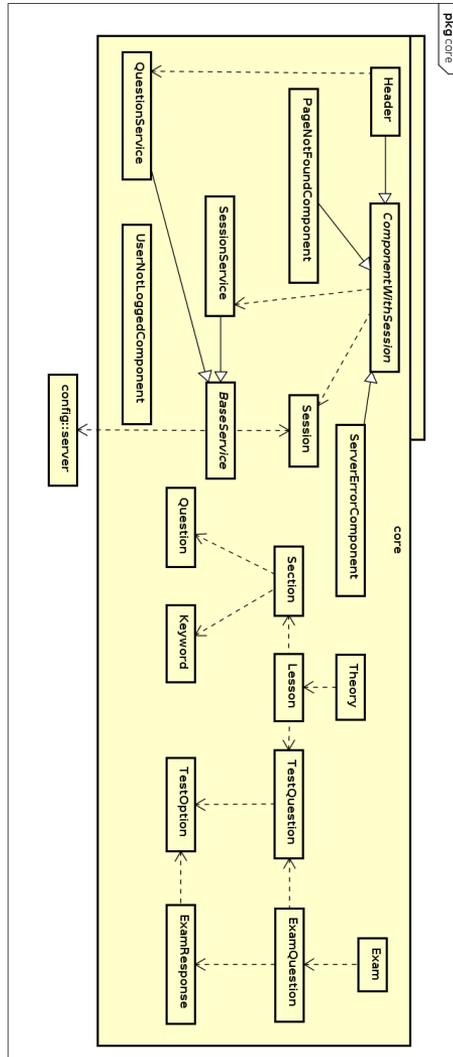


Figura 4.6: Diagrama de descomposición modular frontend-core

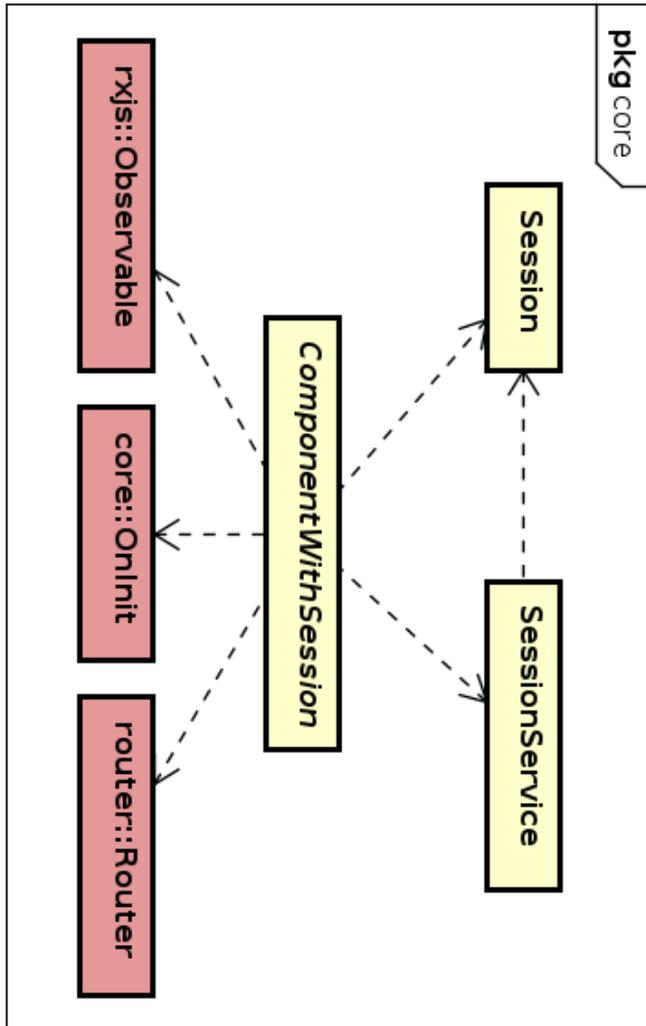


Figura 4.7: Diagrama de ejemplo de un componente

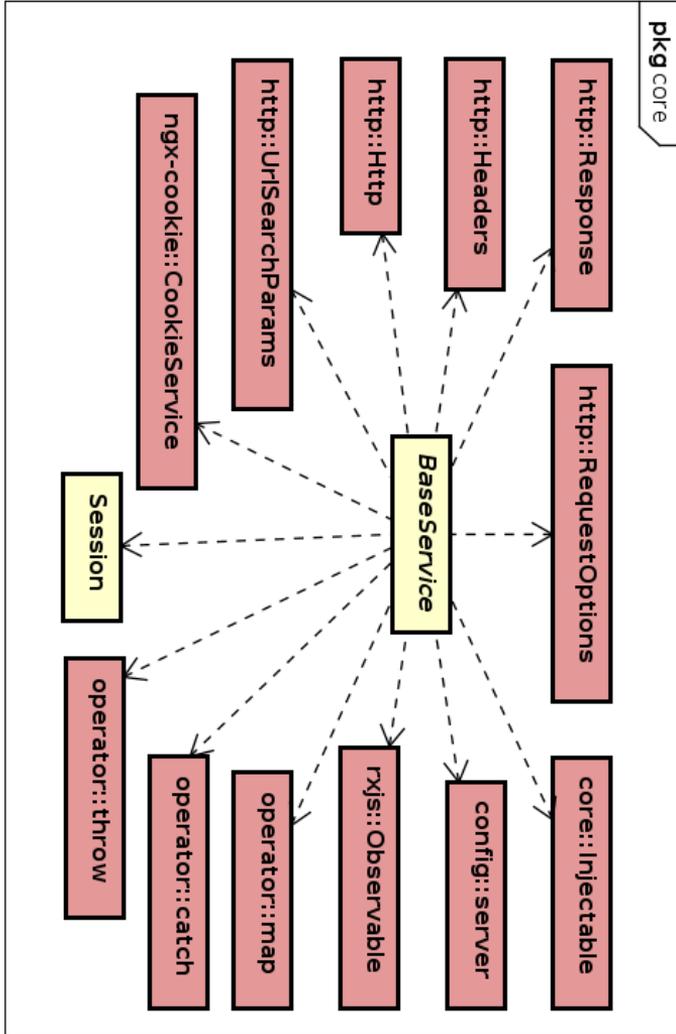


Figura 4.8: Diagrama de ejemplo de un servicio

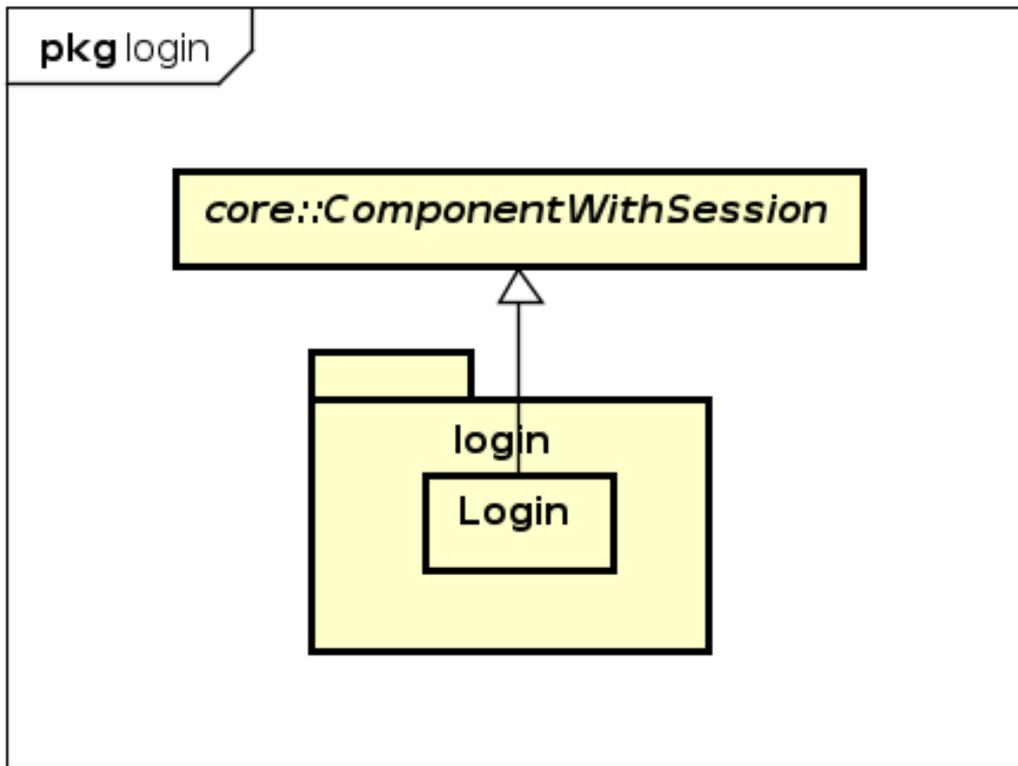


Figura 4.9: Diagrama de descomposición modular frontend-login

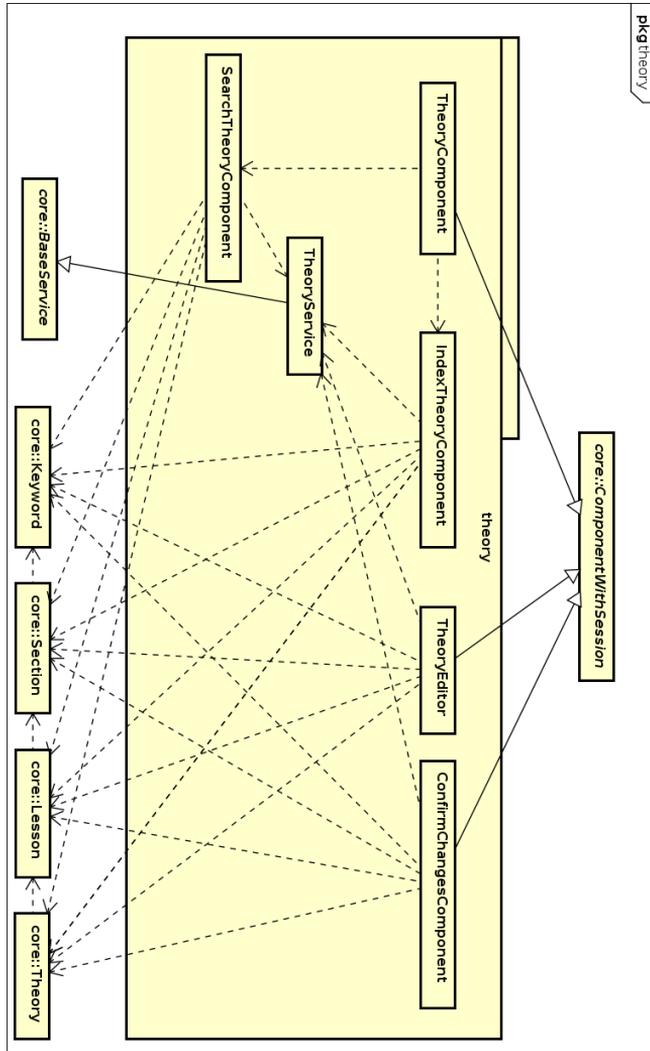


Figura 4.10: Diagrama de descomposición modular frontend-theory

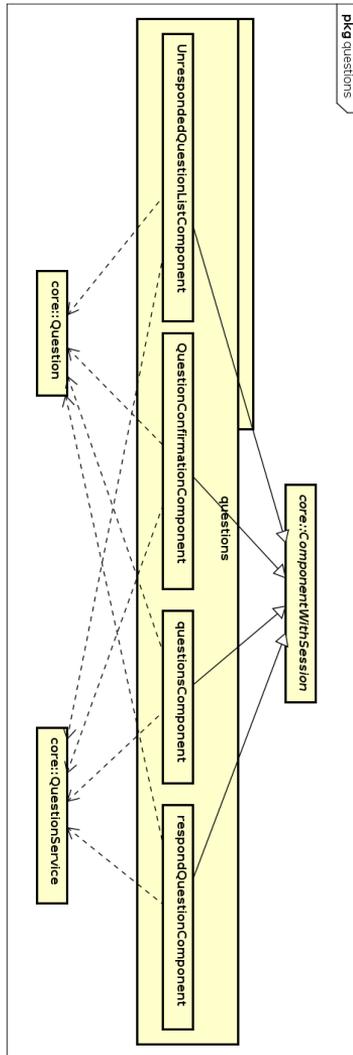


Figura 4.11: Diagrama de descomposición modular frontend-questions



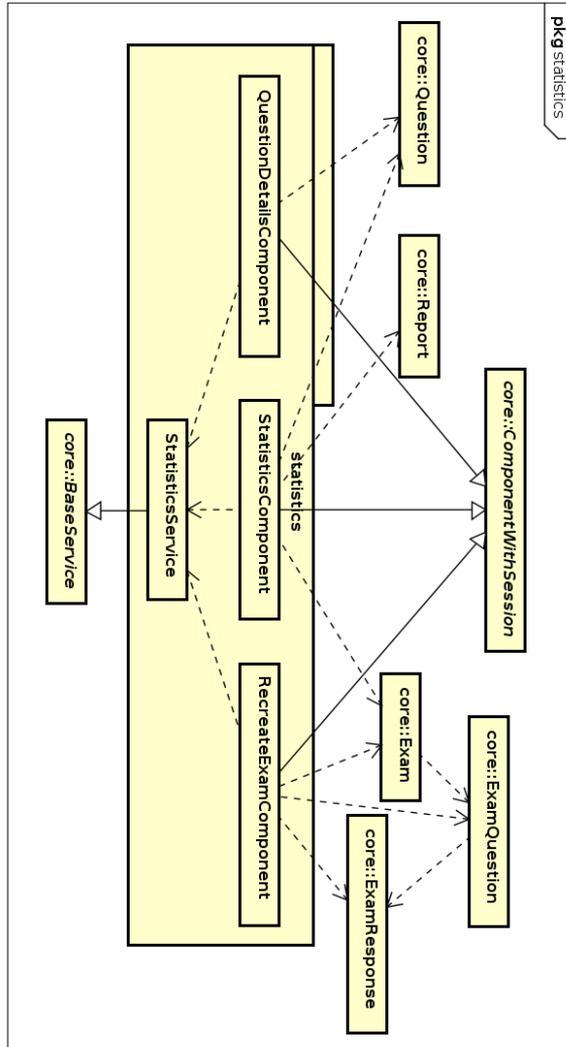


Figura 4.13: Diagrama de descomposición modular frontend-statistics

## Back-end

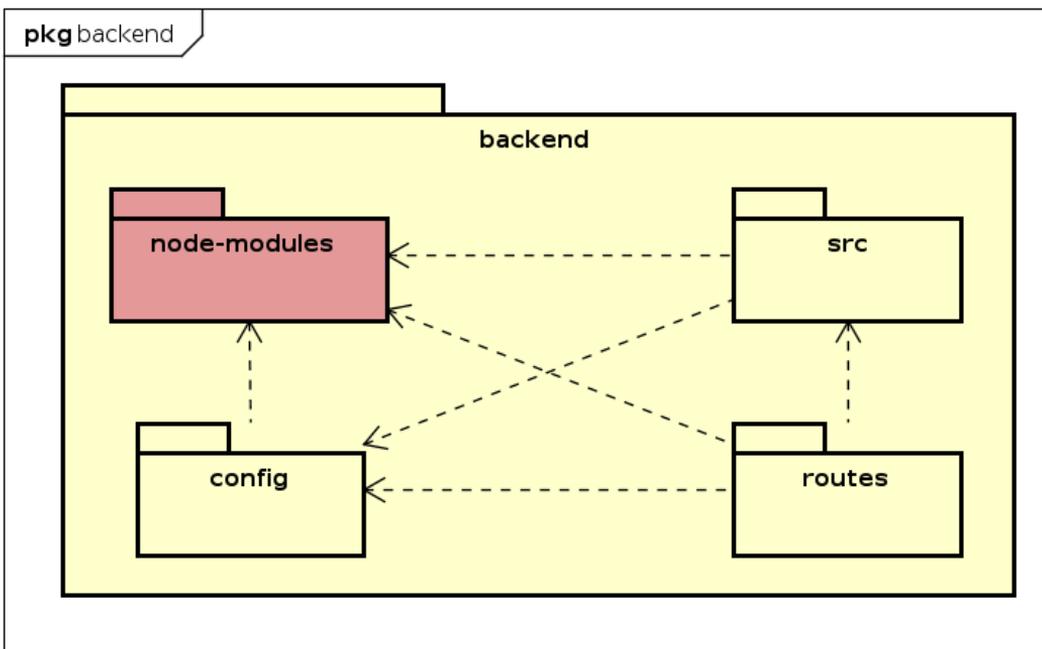


Figura 4.14: Diagrama de descomposición modular backend

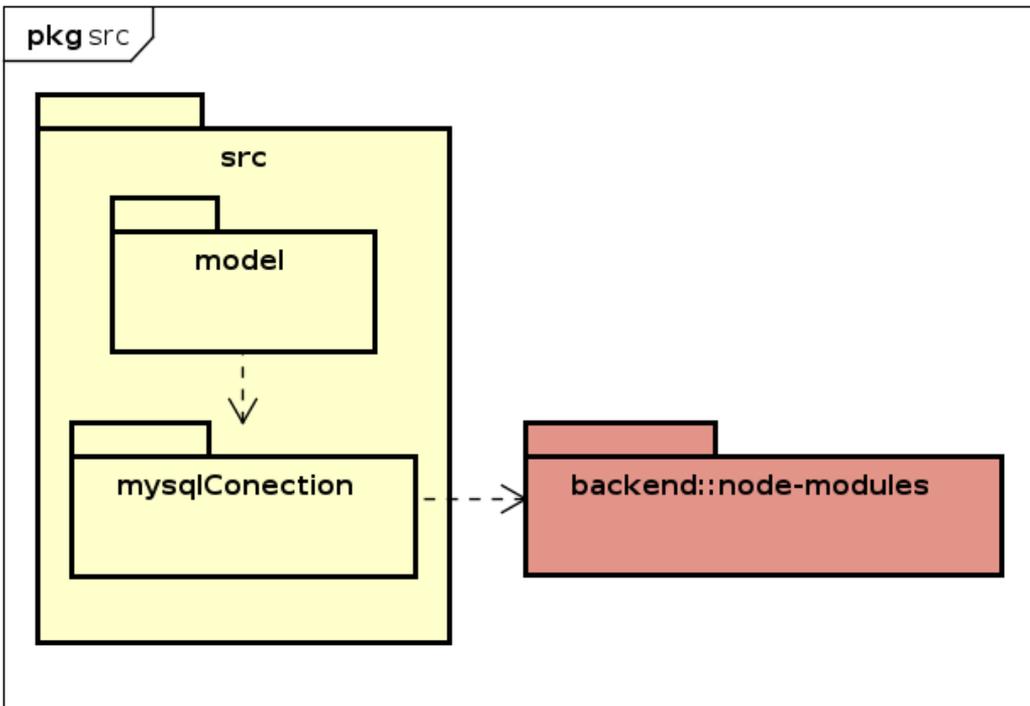


Figura 4.15: Diagrama de descomposición modular backend-src

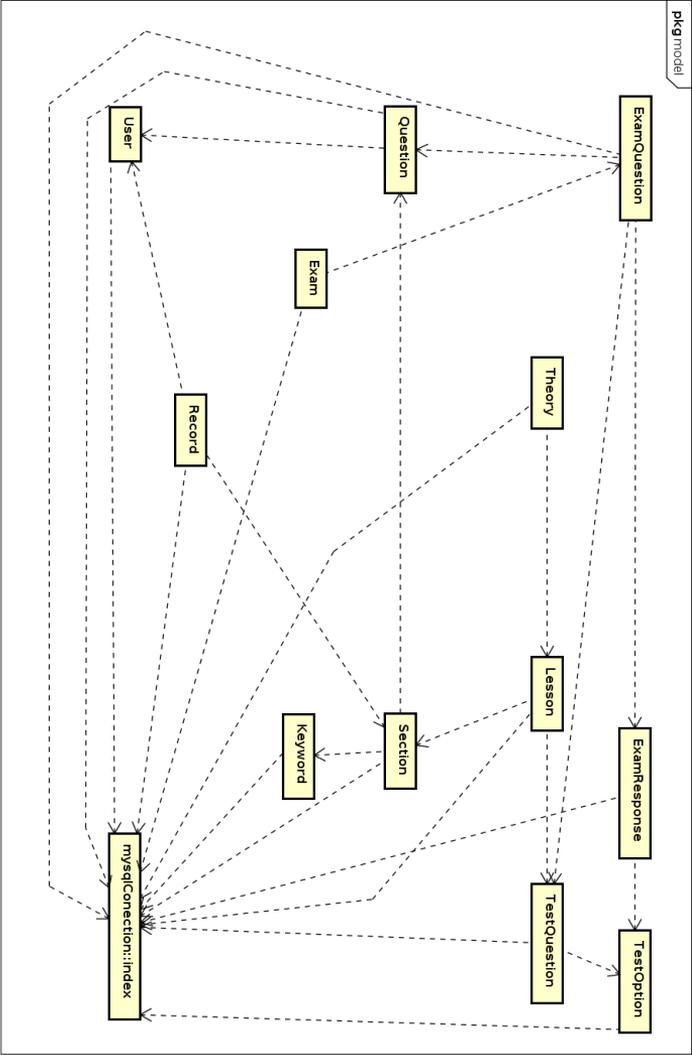


Figura 4.16: Diagrama de descomposición modular backend-model

### 4.2.4. Diagrama de clases

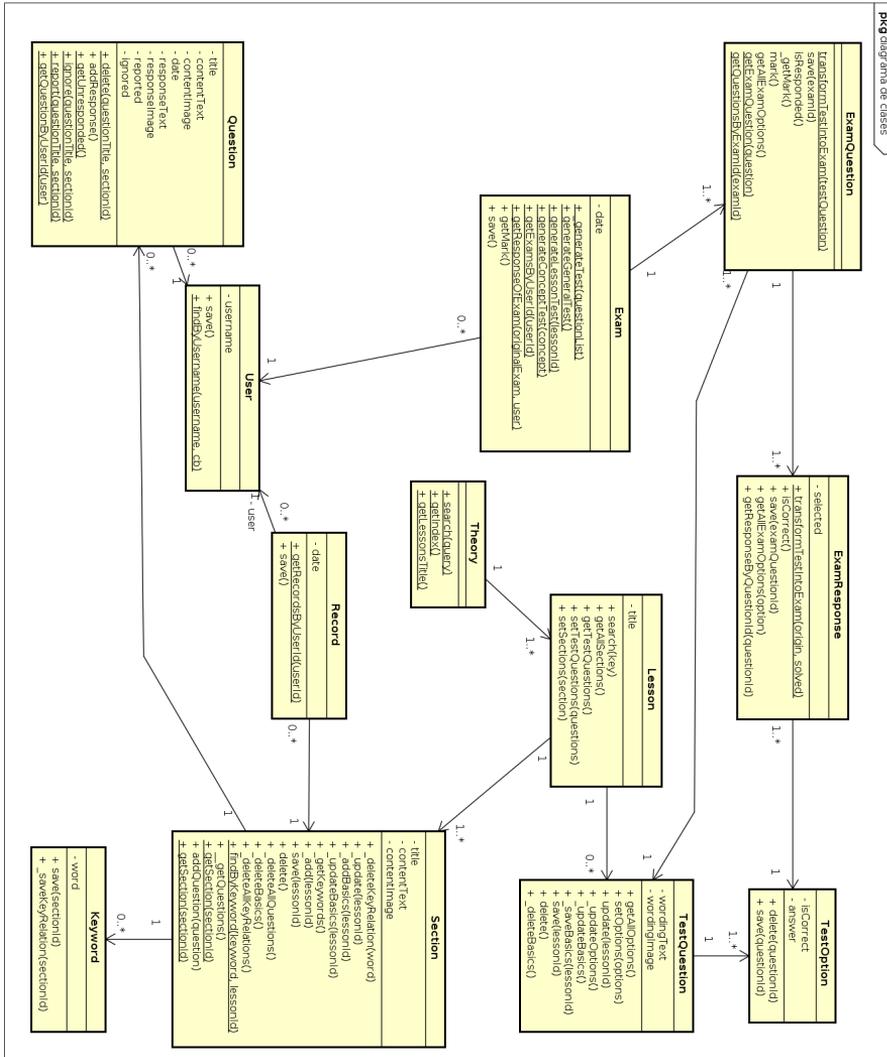


Figura 4.17: Diagrama de clases

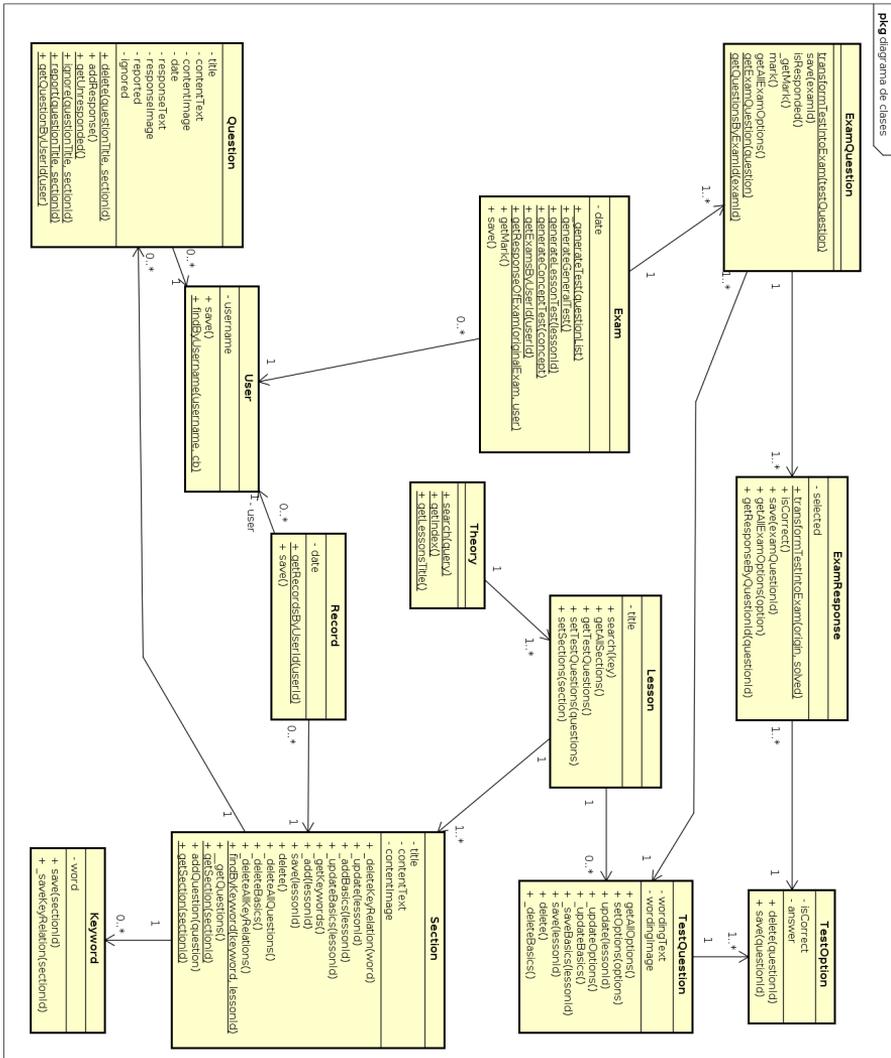


Figura 4.18: Diagrama de clases



## Capítulo 5

# Conclusiones

Se han cumplido la mayoría de objetivos iniciales, desarrollando todas las funcionalidades previstas salvo las relacionadas con la herramienta práctica. Tal y como se explica en el capítulo 3, debido a una mala planificación inicial, se descartó realizar las funcionalidades relacionadas con una herramienta práctica que permitiría al alumno poner en práctica los conocimientos teóricos estudiados.

Se ha desarrollado una interfaz sencilla y clara, orientada a ordenadores de sobremesa.

Durante la realización del proyecto se han adquirido y mejorado nuevos conocimientos del desarrollo web y lenguajes de programación. Se ha aprendido en profundidad el funcionamiento de Angular, Node, Bootstrap 4 y Express.

### 5.1. Trabajo futuro

En el futuro se le podría añadir mayor funcionalidad a la aplicación, aumentando su potencial. Algunas de las que han ido surgiendo durante la relación de este trabajo son:

- Mejora de la interfaz, para hacerla más atractiva para el usuario
- Adecuar el diseño a los dispositivos móviles.
- Añadir la funcionalidad «herramienta práctica» que fue eliminada durante el desarrollo del proyecto.
- Añadir nuevas funcionalidades, como la de permitir al profesor pregenerar un cuestionario específico con un número determinado de preguntas generadas aleatoriamente o bien seleccionadas por el profesor de entre las introducidas previamente en la aplicación. Estos cuestionarios podrían ser utilizados para las pruebas de evaluación continua.
- Aumentar las estadísticas recogidas, así como generar gráficas que muestren los datos.



## Apéndice A

# Manual de despliegue

Para desplegar esta aplicación necesitamos:

- Un ordenador con un sistema operativo basado en UNIX con node y npm instalados
- Conexión a Internet(para descargar dependencias)
- El código fuente de la aplicación.

Aunque se entrega en el CD, esta disponible en <https://github.com/Raikuro/TFG>

Pasos a realizar para desplegar:

### 1. Modificar los ficheros de configuración:

Para entenderlos ficheros de configuración se explica la máquina en la que está actualmente desplegada la aplicación.

La máquina es accesible mediante la dirección «<http://virtual.lab.inf.uva.es>» en el puerto 20052. Debido a la configuración interna de la máquina, redirige el puerto interno 80 al puerto externo 20052 y el puerto 3000 al 20053.

Los ficheros que es necesario modificar son los siguientes:

- backend/config/client.js:

```
const IP = 'http://virtual.lab.inf.uva.es'  
const PORT = 20052  
exports.ADDRESS = IP + ':' + PORT
```

---

Es necesario cambiar las constantes IP y PORT por la IP y el puerto desde el cual será accesible nuestro frontend.

- frontend/.angular-cli.json:

```
...
"defaults": -
  "styleExt": "css",
"component": -,
"serve": -
  "host": "10.0.20.5",
  "port": 80
...
```

Es necesario cambiar los parámetros host y port por la IP y el puerto interno en el que desplegaremos nuestro frontend

- frontend/src/app/config/server.ts

```
const IP = "http://virtual.lab.inf.uva.es";
const PORT = 20053;
export const ADDRESS = IP + ":" + PORT
```

Es necesario cambiar las constantes IP y PORT por la IP y el puerto desde el cual será accesible nuestro backend

## 2. Instalar las dependencias:

Se ejecutarán los siguientes comandos desde la raíz del CD

```
cd backend
npm install
cd frontend
npm install
```

## 3. Desplegar:

Se ejecutarán los siguientes comandos desde la raíz del CD para desplegar el frontend

```
cd frontend
npm start
```

Se ejecutarán los siguientes comandos desde la raíz del CD para desplegar el backend

```
cd backend  
npm start
```



## Apéndice B

# Manual de usuario

A continuación, se explicará el funcionamiento de la aplicación web.

Como primer pantalla, está la página de inicio de sesión. El usuario debe introducir su nombre de usuario de la universidad y su contraseña.



Usuario:

Solo hace falta ser un alumno o profesor de la escuela

Password

Figura B.1: Pantalla de inicio de sesión

Distinguiremos dos casos, según sea un alumno o un profesor el que ingrese en la aplicación.

## B.1. Manual del alumno:

Según se ingresa en la aplicación se accede a la pantalla de teoría. Se puede ver en la parte superior la barra de navegación, y en ella dos apartados, «Teoría» y «Test» y la opción «Salir». Esta última opción, permite cerrar sesión.

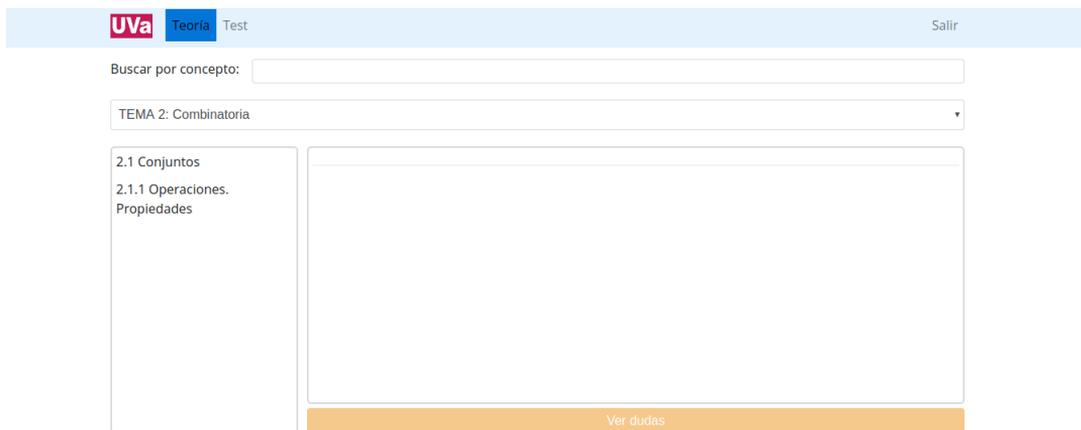


Figura B.2: Pantalla inicial de teoría

Dentro de la pantalla de teoría se puede escoger un tema en el selector, y una vez seleccionado, se puede escoger un concepto. Tras clicar, se ve el contenido del concepto y las palabras relacionadas. A mayores, se activará el boton «Ver dudas». También se puede buscar un concepto basándonos en sus palabras destacadas introduciendo la palabra(o parte de ella) en la barra de búsqueda de la parte superior.

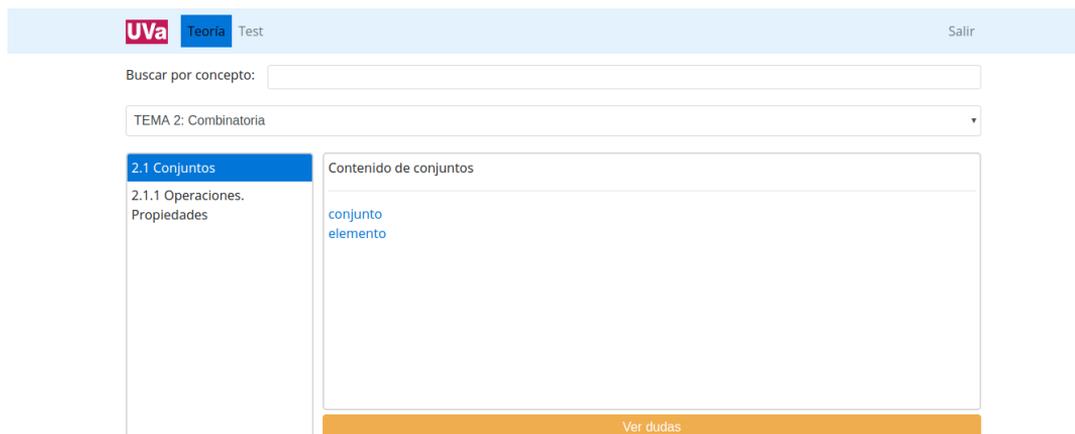


Figura B.3: Pantalla tras pulsar sobre un concepto

Pulsando en el botón «Ver dudas», se accederá a la pantalla de dudas.



Figura B.4: Pantalla ver dudas

Al pulsar en una dudas se verá el contenido de la duda. Al pulsar sobre el boton «Añadir duda» se nos despliega un formulario que permite añadir los datos para insertar una nueva duda. Tras pulsar sobre el botón «Enviar» la aplicación redirige a una pantalla de confirmación. Finalmente, tras confirmar, la duda sera insertada. En esta pantalla también se puede ver una barra de búsqueda, que permite la búsqueda concreta de dudas en función de su contenido.

Pulsando en el botón «Test» de la barra de navegación redirige a la pantalla principal de test.

Figura B.5: Pantalla inicial de test

En este punto, se muestran 3 opciones, buscar test por concepto, buscar test por tema y buscar test general. Cada una de ellas generará un cuestionario basado en la opción escogida y redirigirá a una pantalla que permite rellenar el cuestionario.

Figura B.6: Pantalla de cuestionario

Tras rellenarlo, se pulsa el botón «Enviar» y la aplicación redirigirá a una pantalla de confirmación. Tras confirmar, redirigirá a una pantalla similar a la de cuestionario pero que incluye la resolución del mismo, así como la nota conseguida.

UVa Teoría Test Salir

Tu nota: 1.67

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

Salir

1 - enunciado 2

respuesta1T	✓
respuesta2T	✓
respuesta3F	✗
respuesta4F	✗

2 - enunciado 12

respuesta3T	✓
respuesta4F	✗

3 - enunciado 13

respuesta3T	✓
-------------	---

Figura B.7: Pantalla de resolución

## B.2. Manual del profesor:

Según se entra en la aplicación se accede a la pantalla de teoría. Se puede ver en la barra superior la barra de navegación, y en ella cuatro apartados, «Teoría», «Dudas», «Estadísticas» y «Test» y la opción «Salir». Esta última opción, permite cerrar sesión. El apartado dudas, puede ir acompañado de un número, que indica el número de dudas sin resolver.

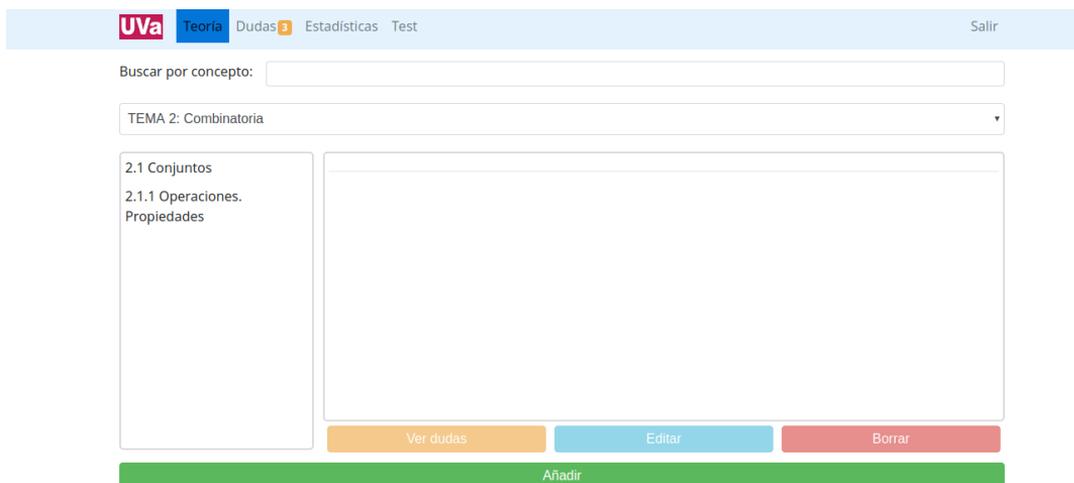


Figura B.8: Pantalla inicial de teoría del profesor

En esta pantalla se mantiene toda la funcionalidad explicada en el apartado «Manual del alumno». Desde esta pantalla se permite añadir un nuevo elemento de teoría, pulsando en el botón «Añadir» y rellenando el formulario correspondiente. En el último paso, la aplicación pedirá confirmación para asegurar que los datos se han introducido correctamente.

Volviendo a la pantalla inicial de teoría, tras pulsar en un concepto se permite borrarlo o editarlo pulsando los botones «Borrar» y «Editar» respectivamente. De forma análoga al paso de añadir tras finalizar el proceso, pedirá confirmación al usuario.

La opción «Ver dudas», funcionará de forma similar a la descrita en el apartado «Manual de alumno», con el añadido de que, una vez seleccionadas, permite al profesor borrar dudas.

Pulsando en el apartado «Dudas» de la barra de navegación la aplicación dirigirá a una nueva pantalla donde se muestran las nuevas dudas sin resolver. Tras pulsar sobre una de ellas, se muestra el contenido, el concepto relacionado y tres opciones, «Reportar», «Ignorar por repetido», «Responder». «Reportar» significa que la duda es inapropiada u ofensiva. «Ignorar por repetido» significa que la dudas está repetida y que no va a ser respondida. Por último, la opción «Responder» permite al profesor responder la duda. El sistema pedirá confirmación antes de hacer cualquiera de estas acciones, ya sea mediante un aviso en el caso de «Reportar» e «Ignorar» o mediante una pantalla de confirmación en el caso de «Responder».

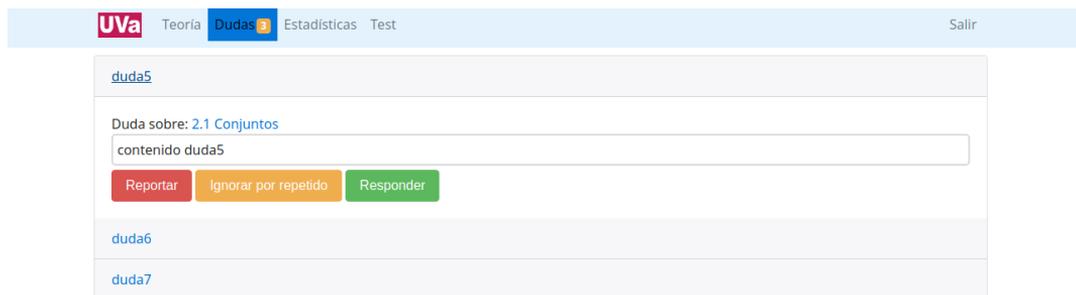


Figura B.9: Pantalla de dudas

La opción «Test» de la barra de navegación redirige a la pantalla de test.



Figura B.10: Pantalla inicial de test del profesor

En ella, el profesor escoge un tema y pulsa el botón «Revisar». Tras ello, se muestran los enunciados de las cuestiones y el botón «Añadir».



Figura B.11: Pantalla de test del profesor

Pulsando sobre cualquiera de ellos, se mostrarán las posibles respuestas, marcando con un tick las correctas y los botones «Editar» y «Borrar». Si se escoge la opción borrar, la aplicación pedirá al profesor confirmación antes de borrarla. Pulsando sobre «Confirmar» la cuestión será eliminada. Si se escoge la opción «Editar», la aplicación nos redirigirá a una pantalla en la cual podremos modificar el contenido de la cuestión. El formulario de editar funciona de manera análoga a la opción añadir, que será explicada a continuación. Finalmente, tras confirmar, la cuestión será editada.

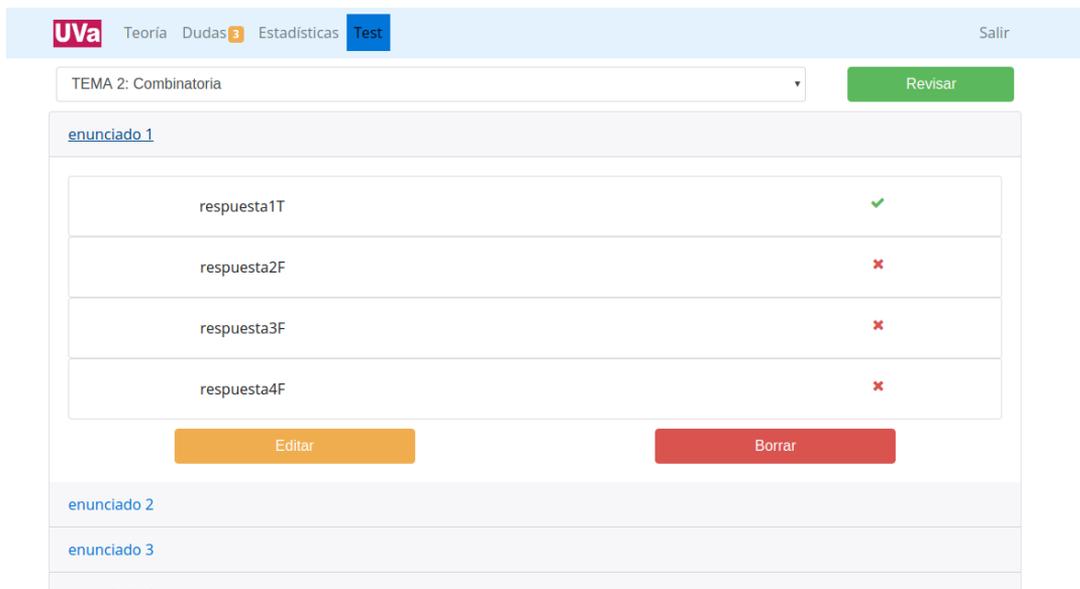


Figura B.12: Pantalla de test del profesor con duda desplegada

La opción «Añadir» permite al profesor añadir una nueva cuestión. Al pulsarla, nos redirigirá a la siguiente pantalla.

Figura B.13: Pantalla de añadir pregunta

Esta pantalla muestra un formulario que formará la pregunta. Para añadir nuevas opciones es necesario que todas las opciones incluidas sean válidas, es decir, no estén vacías ni sean repetidas. En la foto anterior se puede observar que, al estar vacía no permite añadir nuevas opciones. Tras rellenar el campo opción, se puede elegir si la respuesta es correcta o incorrecta cambiando el símbolo a la derecha clicando.

Figura B.14: Pantalla de opción verdadera

Figura B.15: Pantalla de opción falsa

Tras tener una serie de respuestas que cumplan con las condiciones necesarias, esto es, que no tengan opciones repetidas ni vacías y que al menos una de ellas sea verdadera y una falsa; pulsando en el botón «Añadir», la aplicación pedirá confirmación, y tras confirmar, la cuestión será guardada.

Por último, el apartado «Estadísticas» de la barra superior nos redigirá a la página de estadísticas.

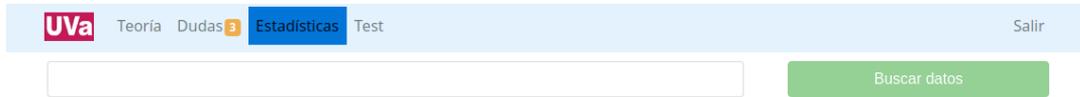


Figura B.16: Pantalla de principal de estadísticas

En ella, el profesor introduce un nombre de alumno y la aplicación muestra el número de conceptos visitados, el número de cuestionarios realizados y el número de dudas preguntadas.

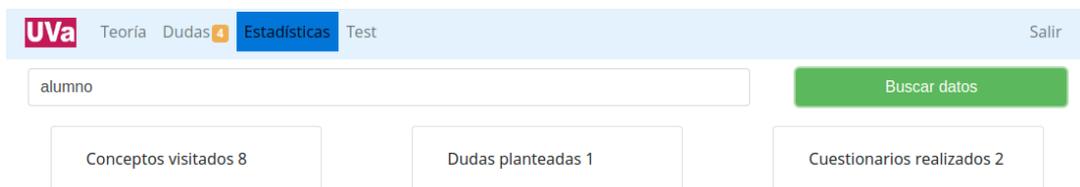


Figura B.17: Pantalla de estadísticas tras buscar

Al pulsar sobre cada uno de estos recuadros, se obtendrá información en detalle.

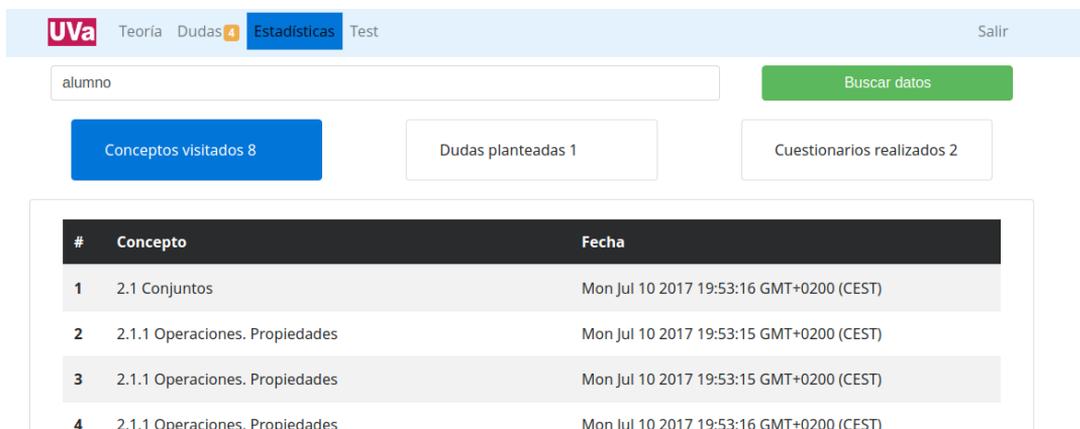


Figura B.18: Pantalla de detalle de los conceptos visitados

Destacar del detalle de las dudas planteadas que marca en rojo las previamente reportadas por el profesor y en naranja las proviamente ignoradas.

#	Duda
1	Tengo una duda con palabras inapropiadas
2	Tengo una duda repetida
3	Tengo una dudad nueva y correcta

Figura B.19: Pantalla de detalle de las dudas planteadas

Destacar del detalle de los cuestionario realizados que marca en rojo los cuestionarios suspensos y en verde los cuestionarios aprobados. Al pulsar el botón «Ver cuestionario», redigirá a una página que mostrará el cuestionario tal y como lo vio el alumno en su corrección.

#	Nota	Cuestionario	Fecha
1	1.33	Ver cuestionario	Mon Jul 10 2017 20:13:01 GMT+0200 (CEST)
2	7	Ver cuestionario	Mon Jul 10 2017 20:13:18 GMT+0200 (CEST)

Figura B.20: Pantalla de detalle de los cuestionarios realizados



## Apéndice C

# Contenido del CD-ROM

El CD incluye los ficheros generados durante la elaboración del trabajo. Éste se divide en tres directorios que describiremos a continuación:

### C.1. Directorio Documentation

Este es el directorio que contiene los ficheros fuente de la elaboración de esta memoria. El contenido se encuentra en formato markdown dividido en secciones en la carpeta secciones. En la carpeta latex podemos ver algunos ficheros necesarios para la creación del documento final. La carpeta img almacena categorizados en subcarpetas las imágenes que después se referencian desde el documento. También están en esa carpeta algunos diagramas creados. En la raíz del directorio está el fichero bibliografia.bib, con la información bibliográfica en formato BibTeX y el fichero main.mdpp que define la estructura del documento y algunos metadatos como el título, agradecimiento y autor.

### C.2. Directorio backend

En este directorio encontramos los ficheros relacionados con el código fuente del backend de la aplicación, así como los ficheros de configuración.

### C.3. Directorio frontend

En este directorio encontramos los ficheros relacionados con el código fuente del frontend de la aplicación, así como los ficheros de configuración.



# Bibliografía

- [1] Debian. *¿Qué es un gestor de paquetes?* Mayo de 2017. URL: <https://www.debian.org/doc/manuals/aptitude/pr01s02.es.html>.
- [2] Built by Edgar. *Compilando ES6 con Babel 6*. Mayo de 2017. URL: <http://blog.builtbyedgar.com/compilando-ecmascript-6-con-babel-6/>.
- [3] Free Software Foundation. *GNU General Public License*. Mayo de 2017. URL: <https://www.gnu.org/licenses/gpl.html>.
- [4] Free Software Foundation. *¿Qué es el copyleft?* Mayo de 2017. URL: <https://www.gnu.org/licenses/copyleft.es.html>.
- [5] Ander González. *MariaDB vs MySQL*. Mayo de 2017. URL: <http://www.tuprogramacion.com/bases-de-datos/mariadb-vs-mysql/>.
- [6] Google. *angular CLI*. Mayo de 2017. URL: <https://cli.angular.io/>.
- [7] Google. *What is Angular?* Mayo de 2017. URL: <https://angular.io/docs>.
- [8] Enrique Fernandez Guerra. *Introducción a TypeScript*. Mayo de 2017. URL: <https://desarrolloweb.com/articulos/introduccion-a-typescript.html>.
- [9] *Guía docente de la asignatura Matemática Discreta – Curso 2016/17*. Jun. de 2017. URL: <https://www.inf.uva.es/wp-content/uploads/2016/06/G46902.pdf>.
- [10] Mozilla Developer Network. *Javascript*. Mayo de 2017. URL: <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- [11] Mozilla Developer Network. *JavaScript language resources*. Mayo de 2017. URL: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language\\_Resources](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Resources).
- [12] Mozilla Developer Network. *Trabajando con objetos*. Mayo de 2017. URL: [https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Trabajando\\_con\\_objetos](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Trabajando_con_objetos).
- [13] Enrique Oriol. *Angular2: ¿Aprendo ES6 o TypeScript?* Mayo de 2017. URL: <http://blog.enriqueoriol.com/2016/06/angular2-aprendo-es6-o-typescript.html>.
- [14] *The Open Source Definition*. Mayo de 2017. URL: <https://opensource.org/osd>.
- [15] Alida Vergara. *Maria DB: una buena opción*. Mayo de 2017. URL: <https://www.facilcloud.com/noticias/maria-db-una-buena-opcion/>.
- [16] *What is TypeScript and why would I use it in place of JavaScript?* Mayo de 2017. URL: <https://stackoverflow.com/questions/12694530/what-is-typescript-and-why-would-i-use-it-in-place-of-javascript>.
- [17] Zeokat. *Que es MariaDB y ventajas frente a MySQL*. Mayo de 2017. URL: <http://www.vozidea.com/que-es-mariadb-y-ventajas-frente-mysql>.

