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

Dedicatoria ejemplo

Agradecimientos

Agradecimiento ejemplo

Resumen

En el presente trabajo se va a desarrollar un software de ayuda para la asignatura “Matemática Discreta”. El objetivo del presente TFG es desarrollar un prototipo funcional de una aplicación web orientada a mejorar el aprendizaje y el afianzamiento de los conceptos relativos a la asignatura. En este punto 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.

Palabras clave: Angular, Node.js, MariaDB, código abierto, TypeScript, JavaScript.

Abstract

Traducción

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Descripción del problema

Los alumnos que entran en el primer curso de una ingeniería experimentan unos grandes cambios a nivel académico. Esto es debido a que cambia la metodología en la docencia y al aumento del volumen de contenidos tanto teóricos como prácticos en comparación con etapas anteriores. Algunas asignaturas se fundamentan en 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, Matemática Discreta incluye conceptos y resultados matemáticos que no han sido explorados con anterioridad. Esta asignatura forma parte de las asignaturas de carácter básico.

Para tener una buena base en informática es necesario conseguir un buena base matemática teórica. 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.

Con la asiganatura Matemática Discreta 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 a lo largo de la vida los aspectos de la Informática relacionados con la lógica, teoría de conjuntos y combinatoria, como son la inteligencia artificial, el uso de bases de datos o la estadística.

El presente trabajo se va a centrar en mejorar el aprendizaje y la resolución de dudas respecto a la asignatura.

1.2. Objetivo

El objetivo final de este trabajo es mejorar el estudio de la asignatura, facilitando el afianzamiento de los conocimientos. Para ello, se busca crear una plataforma que facilite la comunicación alumno-profesor, y dote al profesor de las herramientas para facilitar la distribución de los contenidos de la asignatura, así como permitiendo al alumno comprobar sus conocimientos mediante la implementación de tests. En definitiva, se busca mejorar la preparación de los alumno para afrontar la asignatura satisfactoriamente, ya que esta sienta unas bases muy importantes para otras de cursos futuros.

1.3. Contexto [9]

La metodología docente actual de la asignatura 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 conocimientos que deben ser asimilados con las horas de estudio individual. Gracias al afianzamiento de los conocimientos, el alumno será capaz de participar de forma activa en las sesiones prácticas y teóricas, así como superar satisfactoriamente las diversas pruebas escritas a lo largo del curso. Todo ello influye positivamente en la calificación.

El material básico del que dispone el alumno consiste en un documento pdf con los contenidos teóricos y enunciados de problemas, que forman los conocimientos mínimos 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 unicas 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 ayudarle a afrontar las etapas finales de la asignatura.

Lo que buscamos es una herramienta que nos permita buscar 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. 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:

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

TECNOLOGÍAS UTILIZADAS: En este capítulo se detalla infromación sobre las tecnologías escogidas, así como se explican ciertos conceptos clave para entenderlas.

DESARROLLO DEL PROYECTO: La parte fundamental del proyecto es el desarrollo de un prototipo de plataforma web. En este apartado 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:** [10] JavaScript (abreviado como JS) es un lenguaje multi-paradigma, ligero e interpretado.
- **TypeScript:** [13, 16] 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.
- **ECMAScript:** [11] 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.
- **Transpilar:** [8, 2] 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 aproxiamadamente.
- **Gestor de paquetes:** [1] 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.
- **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:** [14] La terminología open source incluye a aquellos softwares que cumplen los siguientes requisitos:

  - **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. Concierne 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 [7]

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.

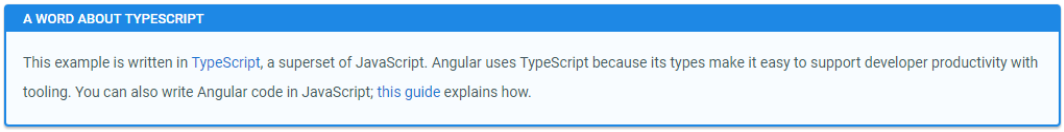


Figura 2.5: Google sobre TypeScript

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 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 [6]

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.

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. Surgío 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 mai
  sudo apt-get update
  sudo apt-get install mariadb-server
```

2.8.2. Ventajas de MariaDB frente MySQL [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.

2.8.3. Desventajas de MariaDB frente MySQL

- **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 soft- ware. 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, que proporciona una descrip- ción de propósito, alcance y objetivos; la gestión de los procesos que explica el coste estimado; y la planificación 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

3.1.2. Metodología utilizada

Explicacion –> Kanban?

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 6 horas diarias con un solo recurso, con un día semanal de descanso.

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:** 1 semana
- **Elicitación de requisitos**
  - Asegurar los requisitos extraidos 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:** 1 semana
- **Realización del diagrama de casos de uso y elaboración de un plan de trabajo provisional**
  - Realización de un plan de trabajo provisional para tener una marco de referencia a la hora de distribuir el tiempo. Para ello, se decidiran cuales son los casos de uso y se hara el diagrama correspondiente, sin profundizar.
  - **Duración:** 1 día
- **Diseño de las vistas de la aplicación y elicitación con el cliente**
  - Diseño en papel de las vistas de la aplicación.
  - **Duración estimada:** 1 semana
  - **Duración real:** 1 semana
- **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:** 1 semana

    - **Duración real:** 1 semana

- **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:** 1 semana
    - **Duración real:** 1 semana
- **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:** 1 semana
    - **Duración real:** 1 semanas 3 días
- **Realizacion de la descripcion en detalle de los casos de uso**
    - Basándonos en el diagrama realizado en el pasos anteriores, profundizamos en cada uno de los casos de uso, definiendo su descripción en detalle.
    - **Duración estimada:** 1 semana
    - **Duración real:** 3 dias
- **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:** 1 semana
    - **Duración real:** 2 semanas
- **Realización básica de backend:**
    - Programación de los elementos básicos y comunes a las aplicaciones Nodejs
    - **Duración estimada:** 1 semana
    - **Duración real:** 2 semanas
- **Realización de un boceto de diseño:**
    - Realización de un diagrama de secuencia que ejemplifique el comportamiento de la aplicación, así como bocetos del resto de diagramas de diseño, que serán completados
    - **Duración estimada:** 1 día
    - **Duración real:** 1 día
- **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
    - **Duracion estimada:** 1 semana
    - **Duracion real:** 1 semana
- **Comprobación de las funcionalidades implementadas:**

  - Comprobación del estado de la aplicación y solución de errores inesperados
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana

- **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 teoria, esto es, mostrar la teoria organizada por temas, permitir la busqueda de teoría por concepto y permitir al profesor añadir y editar conceptos.
  - **Duracion estimada:** 1 semana
  - **Duracion real:** 1 semana
- **Comprobación de las funcionalidades implementadas:**
  - Comprobación del estado de la aplicación y solución de errores inesperados
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana
- **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.
  - **Duracion estimada:** 1 semana
  - **Duracion real:** 1 semana
- **Comprobación de las funcionalidades implementadas:**
  - Comprobación del estado de la aplicación y solución de errores inesperados
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana
- **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 segun una serie de parámetros, resolverlos y conocer su resultado y permitir al profesor añadir y editar preguntas.
  - **Duracion estimada:** 1 semana
  - **Duracion real:** 1 semana
- **Comprobación de las funcionalidades implementadas:**
  - Comprobación del estado de la aplicación y solución de errores inesperados
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana
- **Realización de las funcionalidades relacionadas con estadisticas:**
  - 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 segun una serie de parámetros, resolverlos y conocer su resultado y permitir al profesor añadir y editar preguntas.

  - **Duracion estimada:** 1 semana
  - **Duracion real:** 1 semana

- **Comprobación de las funcionalidades implementadas:**

  - Comprobación del estado de la aplicación y solución de errores inesperados
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana

- **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.
  - **Duracion estimada:** 1 semana
  - **Duracion real:** No realizado

- **Comprobación de las funcionalidades implementadas:**

  - Comprobación del estado de la aplicación y solución de errores inesperados
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** No realizado

- **Mejora de la interfaz:**

  - Realización de una interfaz mas allá de la interfaz útil.
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana

- **Realización de documentación del TFG:**

  - Realización del presente documento, esto incluye los capítulos Introduccion y Conclusiones y los anexos Manual de usuario y Manual de instalación, así como la maquetación del resto de capítulos.
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** 1 semana

- **Mejora de la interfaz:**

  - Mejorar el aspecto de la aplicación
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** No realizado

- **Limpieza de código:**

  - Borrado de funciones en desuso, borrado de comentarios innecesarios.
  - **Duracion estimada:** 0.5 semanas
  - **Duracion real:** No realizado

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 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.

- **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 - Maquina personal averiada**
  - **Impacto:** El impacto se determinará en función del nivel de avería
  - **Probabilidad:** 1
  - **Plan de potección:** 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 numero 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.

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

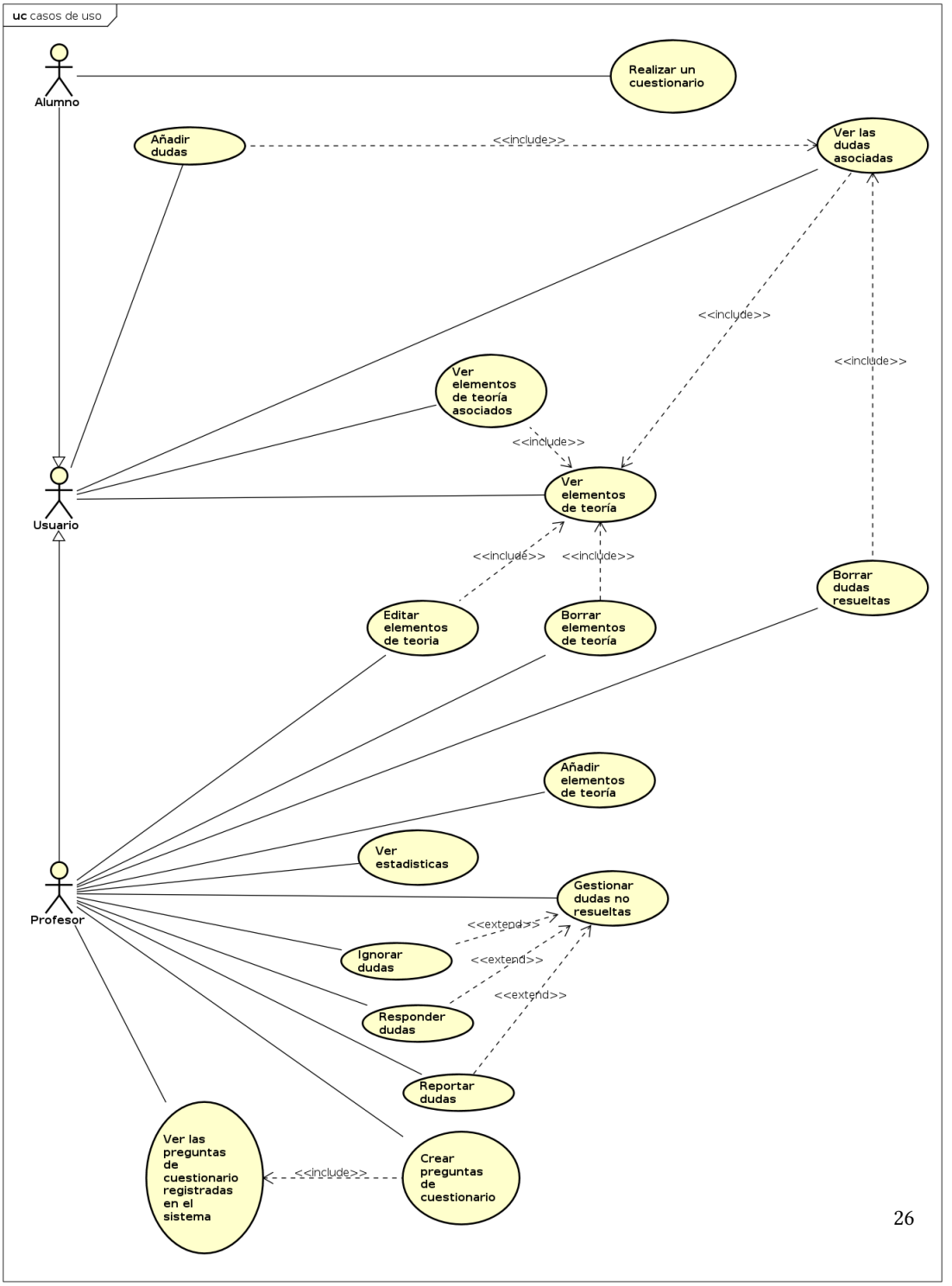
- **RF-10: Borrar contenidos teóricos** La aplicación permitirá al profesor borrar contenidos teóricos
- **RF-11: Editar contenidos teóricos** La aplicación permitirá al profesor editar contenidos teóricos
- **RF-12: Borrar dudas resueltas** La aplicación permitirá al profesor borrar dudas ya resueltas
- **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



Descripcion 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 coi nciden 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 err or, 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

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, avis a 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 err or, 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 muestr a al usuario.
Branch Sequence	
Exception Sequence	2a,4a,6a - Si el sistema no está en funcionamiento o genera algún err or, 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

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 estadisticas
Actor	Profesor
Precondition	El profesor está identificado en el sistema.
Postcondition	
Base Sequence	1 - El profesor elige la opción "Ver estadisticas". 2 - El sistema pide el nombre de usuario sobre el que ver estadisticas. 3 - El profesor introduce el nombre de usuario sobre el que ver estadisticas. 4 - El sistema muestra las estadisticas 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 estadisticas

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 err or, 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

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 err or, 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

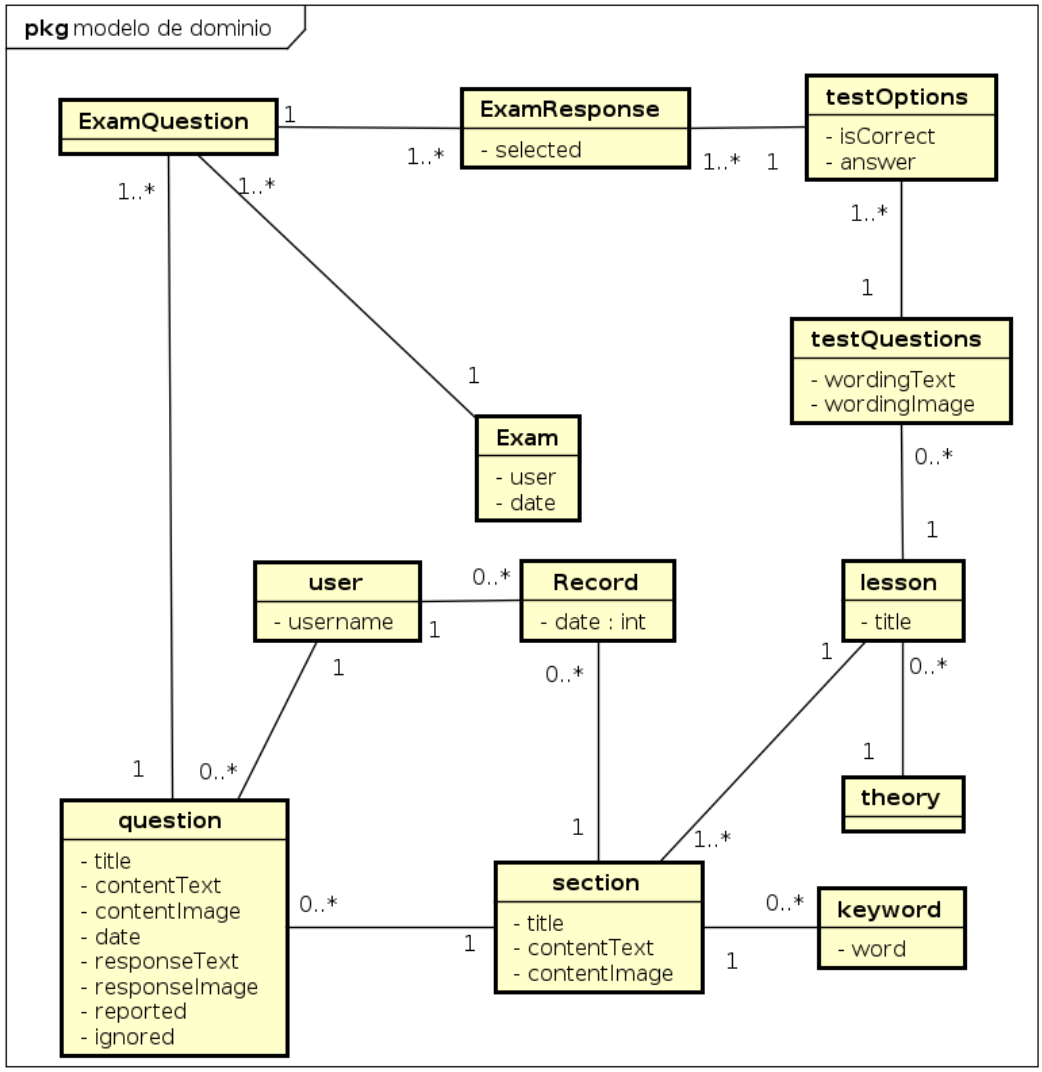


Figura 4.2: Modelo de dominio

4.2. Diseño

4.2.1. Diseño de la base de datos

4.2.2. Despliegue

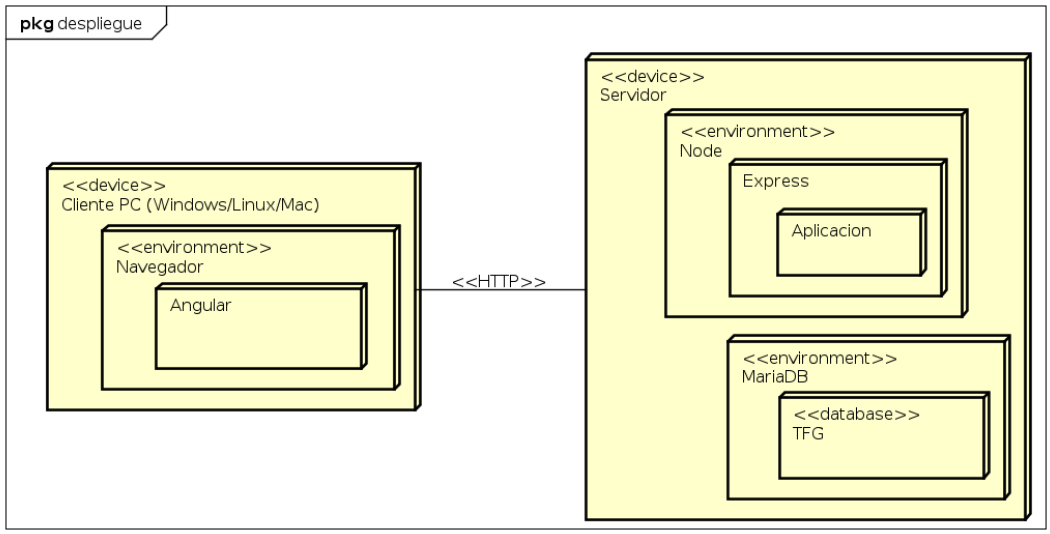


Figura 4.3: Diagrama de despliegue

4.2.3. Arquitectura del sistema

Front-end

Back-end

4.2.4. Diagrama de clases

4.2.5. Diagramas de secuencia

Capítulo 5

Conclusiones

Conclusiones

5.1. Trabajo futuro

Como mejorar la aplicacion

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 raiz del CD

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

3. **Desplegar:**

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

```
cd frontend
npm start
```

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

```
cd backend
npm start
```

Apéndice B

Manual de usuario

Apéndice C

Contenido del CD-ROM

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://desarrollowe
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.

[12] Mozilla Developer Network. *Trabajando con objetos.* Mayo de 2017. URL: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Trabajando_con_objectos.

[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.