



Universidad de  
Oviedo



Universidad de Oviedo

## **ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

### **GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN**

**ÁREA DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA  
ARTIFICIAL**

**TRABAJO FIN DE GRADO N.º 17010059**

**MEJORA DEL RENDIMIENTO Y AMPLIACIÓN DE UNA WEB PARA  
LA GESTIÓN INTEGRAL DE CONGRESOS CIENTÍFICOS**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



# Índice de contenidos

DOCUMENTO 1: MEMORIA .....	14
1.1 Resumen .....	16
1.2 Introducción.....	17
1.3 Objetivos y alcance.....	18
1.3.1 Objetivos de la aplicación .....	18
1.3.2 Objetivos académicos y de aprendizaje.....	19
1.4 Estudios y análisis previos.....	20
1.4.1 Estudios de carácter teórico.....	20
1.4.1.1 Aconf.....	21
1.4.1.2 EasyChair .....	22
1.4.1.3 Primoris .....	23
1.4.1.4 Sistedes .....	24
1.4.2 Estudios de carácter técnico .....	26
1.4.2.1 Patrones de diseño .....	26
1.4.2.1.1 Modelo .....	27
1.4.2.1.2 Vista .....	27
1.4.2.1.3 Controlador .....	27
1.4.2.2 Lenguajes de Programación.....	28
1.4.2.2.1 PHP .....	29
1.4.2.2.2 Java.....	29
1.4.2.2.3 Ruby .....	29
1.4.2.2.4 Python .....	29
1.4.2.2.5 C# .....	29
1.4.2.2.6 Node.js .....	29
1.4.2.3 Sistema de gestión de bases de datos.....	30
1.5 Organización del documento .....	31
1.5.1 Documento 1 – Memoria .....	31
1.5.2 Documento 2 – Presupuesto y planificación .....	31
1.5.3 Documento 3 – Requisitos de usuario y análisis de alternativas .....	31
1.5.4 Documento 4 – Análisis de requisitos del sistema .....	32

1.5.5 Documento 5 – Diseño del sistema .....	32
1.5.6 Documento 6 – Pruebas .....	32
1.5.8 Documento 7 – Manuales.....	32
1.6 Conclusiones y ampliaciones.....	32
1.6.1 Conclusiones .....	33
1.6.2 Ampliaciones .....	33
1.6.2.1 Mantenimiento continuo de actualizaciones de <i>software</i> .....	33
1.6.2.2 Creación de aplicación móvil nativa.....	34
1.6.2.3 Internacionalización.....	34
1.6.2.4 Creación de un Chatbot en el sistema.....	34
1.7 Bibliografía.....	35
1.7.1 Libros y artículos.....	35
1.7.2 Referencias en internet .....	35
<b>DOCUMENTO 2: PLANIFICACIÓN Y PRESUPUESTO.....</b>	<b>38</b>
2.1 Introducción.....	40
2.2 Planificación temporal .....	40
2.2.1 Fase de análisis.....	41
2.2.2 Fase de diseño .....	41
2.2.3 Fase de implementación .....	42
2.2.4 Fase de pruebas .....	42
2.3 Presupuesto.....	43
2.3.1 Coste del material software/hardware .....	43
2.3.2 Coste del personal .....	45
2.3.3 Coste total .....	46
<b>DOCUMENTO 3: REQUISITOS DE USUARIO Y ANÁLISIS DE ALTERNATIVAS .....</b>	<b>47</b>
3.1 Introducción.....	49
3.2 Identificación de usuarios participantes en el sistema .....	49
3.2.1 Rol de participante en el congreso .....	49
3.2.2 Rol de usuario administrador .....	49
3.3 Requisitos de usuario.....	50
3.3.1 Requisitos de usuario comunes .....	50
3.3.2 Requisitos de usuario – rol de participante en el congreso.....	50

3.3.3 Requisitos de usuario – rol de usuario administrador.....	52
3.4 Análisis de alternativas.....	53
3.4.1 Tecnologías <i>backend</i> .....	53
3.4.1.1 PHP.....	54
3.4.1.2 Ruby .....	54
3.4.1.3 Python.....	55
3.4.1.4 ASP.NET .....	55
3.4.1.5 Java EE .....	56
3.4.1.6 Node.js.....	56
3.4.2 Tecnologías <i>frontend</i> .....	57
3.4.2.1 HTML5.....	57
3.4.2.2 XML .....	58
3.4.2.3 JavaScript .....	58
3.4.2.4 jQuery .....	58
3.4.2.5 TypeScript .....	59
3.4.2.6 Angular .....	59
3.4.2.7 CSS3 .....	60
3.4.2.8 Sass.....	60
3.4.2.9 Bootstrap.....	61
3.4.3 Base de datos.....	61
3.4.3.1 MySQL.....	62
3.4.3.2 PostgreSQL.....	62
3.4.3.3 MongoDB .....	62
3.4.3.4 Firebase Cloud Firestore.....	63
3.5 Descripción de la alternativa seleccionada .....	63
3.5.1 Tecnologías <i>frontend</i> .....	64
3.5.2 Tecnologías <i>backend</i> .....	65
3.5.3 Base de datos.....	66
3.5.4 Sistema operativo .....	68
DOCUMENTO 4: ANÁLISIS DE REQUISITOS DEL SISTEMA .....	69
4.1 Introducción.....	71
4.2 Subsistemas de análisis y casos de uso.....	71

4.2.1 Acceso y registro de usuarios .....	71
4.2.1.1 Registro de usuario .....	72
4.2.1.2 <i>Login</i> .....	73
4.2.1.3 Recuperación de contraseña .....	74
4.2.1.4 Navegación a página informativa del congreso .....	75
4.2.1.5 Cerrar sesión .....	75
4.2.2 Gestión de usuarios asistentes al congreso.....	76
4.2.2.1 Modificación de la información de usuario .....	77
4.2.2.2 Modificación de la contraseña de usuario.....	78
4.2.2.3 Subir justificante de pago .....	79
4.2.2.4 Eliminar justificante de pago .....	79
4.2.2.5 Visualización de justificante de pago .....	80
4.2.2.6 Añadir datos de factura.....	80
4.2.2.7 Subir artículo .....	81
4.2.2.8 Visualizar artículo.....	82
4.2.2.9 Modificación de artículo.....	82
4.2.2.10 Eliminar artículo .....	83
4.2.2.11 Descarga de factura .....	84
4.2.2.12 Descarga de certificado de asistencia .....	85
4.2.2.13 Descarga de certificado de ponencia .....	85
4.2.3 Gestión de usuario administrador.....	86
4.2.3.1 Visualización tabla de usuarios .....	86
4.2.3.2 Modificación usuario .....	87
4.2.3.3 Eliminar usuario .....	88
4.2.3.4 Visualización tabla de pagos .....	88
4.2.3.5 Visualización justificante de pago de usuario.....	89
4.2.3.6 Modificación cuota de pago de usuario .....	89
4.2.3.7 Modificación validación de pago de usuario .....	90
4.2.3.8 Descarga de factura de usuario .....	91
4.2.3.9 Visualización tabla de autores .....	91
4.2.3.10 Visualización tabla de artículos .....	92
4.2.3.11 Visualización artículo de usuario.....	92

4.2.3.12 Modificación estado de artículo de usuario .....	92
4.2.3.13 Modificación artículo de usuario .....	93
4.2.3.14 Eliminar artículo de usuario.....	94
4.2.3.15 Visualización tabla de conferencias.....	95
4.2.3.16 Añadir nueva conferencia .....	95
4.2.3.17 Modificación conferencia .....	96
4.2.3.18 Eliminar conferencia.....	96
4.2.3.19 Descarga de certificados de asistencia.....	97
4.2.3.20 Descarga de certificados de ponencia .....	98
4.2.3.21 Descarga de facturas .....	98
4.2.3.22 Descarga de lista de usuarios .....	99
4.2.3.23 Descarga de lista de artículos .....	99
4.2.3.24 Descarga de lista de autores.....	99
4.2.3.25 Descarga de etiquetas .....	100
4.2.3.26 Modificación de la configuración .....	101
4.2.3.27 Resetear base de datos .....	101
4.3 Requisitos no funcionales .....	103
4.4 Modelo de datos .....	104
4.4.1 Descripción de los modelos.....	104
4.4.1.1 Colección Users .....	104
4.4.1.2 Colección Config .....	105
4.4.1.3 Colección Conferences .....	106
4.4.1.4 Colección Bill .....	106
4.4.1.5 Colección Papers .....	106
4.4.1.6 Colección Payment_file .....	107
4.4.1.7 Colección Authors .....	107
DOCUMENTO 5: DISEÑO DEL SISTEMA .....	109
5.1 Introducción.....	111
5.2 Diagrama de paquetes.....	111
5.2.1 Componentes.....	112
5.2.2 Vista .....	113
5.2.3 Controlador .....	113

5.2.4 Estilos.....	113
5.2.5 Tests .....	113
5.2.6 Configuración.....	114
5.2.7 Assets .....	114
5.2.8 Servicios.....	114
5.2.9 Mail .....	114
5.2.10 Base de datos.....	115
5.3 Diagrama de diseño de componentes .....	115
5.4 Diagrama de interacción .....	117
5.5 Diagrama de diseño de servicios .....	118
5.6 Diseño de la interfaz .....	119
DOCUMENTO 6: PRUEBAS.....	121
6.1 Introducción.....	123
6.2 Pruebas unitarias.....	123
6.3 Pruebas de integración y de sistema .....	125
6.4 Pruebas de usabilidad y accesibilidad.....	130
6.4.1 Criterios generales.....	130
6.4.2 Criterios de identidad e información .....	131
6.4.3 Criterios de lenguaje y redacción .....	132
6.4.4 Criterios de rotulado.....	132
6.4.5 Criterios de navegación .....	133
6.4.6 <i>Layout</i> de la página .....	133
6.4.7 Criterios de búsqueda.....	134
6.4.8 Accesibilidad.....	135
6.4.9 Criterios de control y retroalimentación.....	135
6.5 Pruebas de rendimiento .....	136
6.5.1 Análisis PageSpeed Insights.....	137
6.5.2 Análisis Lighthouse.....	138
DOCUMENTO 7: MANUALES .....	141
7.1 Introducción.....	143
7.2 Manual de instalación y ejecución en modo local .....	143
7.3 Manual de sustitución de la base de datos .....	144

7.4 Manuales de despliegue.....	147
7.4.1 Manual de despliegue en Firebase.....	148
7.4.2 Manual de despliegue en servidor web .....	149
7.5 Manual de usuario .....	149
7.5.1 Pantalla de inicio .....	149
7.5.2 Pantalla de registro .....	151
7.5.3 Pantalla de inicio de sesión .....	152
7.5.4 Pantalla de recuperación de contraseña.....	154
7.5.5 Pantalla de perfil de usuario .....	155
7.5.6 Pantalla de pagos y facturas .....	156
7.5.7 Pantalla de subida de artículos .....	158
7.5.8 Pantalla de documentos .....	160
7.5.9 Pantalla de usuarios – rol administrador .....	162
7.5.10 Pantalla de pagos – rol administrador .....	163
7.5.11 Pantalla de autores – rol administrador .....	164
7.5.12 Pantalla de artículos – rol administrador .....	165
7.5.13 Pantalla de conferencias – rol administrador.....	166
7.5.14 Pantalla de documentos – rol administrador .....	167
7.5.15 Pantalla de configuración – rol administrador.....	168

## Índice de figuras

Figura 1: Página inicial de Aconf .....	22
Figura 2: Página inicial de EasyChair .....	23
Figura 3: Página inicial de Primoris .....	24
Figura 4: Página inicial de Sistedes.....	25
Figura 5: Esquema patrón MVC.....	27
Figura 6: Diagrama Gantt – Visión global .....	40
Figura 7: Diagrama Gantt - Fase de análisis.....	41
Figura 8: Diagrama Gantt - Fase de diseño .....	41
Figura 9: Diagrama Gantt - Fase de implementación.....	42
Figura 10: Diagrama Gantt - Fase de pruebas .....	42
Figura 11: Logo PHP.....	54
Figura 12: Logo Ruby .....	54
Figura 13: Logo Python.....	55
Figura 14: Logo ASP.NET .....	55
Figura 15: Logo Java EE .....	56
Figura 16: Logo Node.js.....	57
Figura 17: Logo HTML5.....	57
Figura 18: Logo XML .....	58
Figura 19: Logo JavaScript .....	58
Figura 20: Logo jQuery .....	59
Figura 21: Logo TypeScript .....	59
Figura 22: Logo Angular .....	60
Figura 23: Logo CSS3 .....	60
Figura 24: Logo Sass .....	61
Figura 25: Logo Bootstrap.....	61
Figura 26: Logo MySQL.....	62
Figura 27: Logo PostgreSQL.....	62
Figura 28: Logo MongoDB .....	63
Figura 29: Logo Firebase.....	63

Figura 30: Subsistema Acceso y registro de usuarios .....	72
Figura 31: Subsistema Gestión de usuarios asistentes al congreso .....	77
Figura 32: Gestión de usuario administrador .....	86
Figura 33: Diagrama estructural de la base de datos .....	108
Figura 34: Diagrama de paquetes del sistema .....	112
Figura 35: Diagrama de componentes .....	116
Figura 36: Diagrama de interacción .....	117
Figura 37: Diagrama de diseño de servicios.....	118
Figura 38: Diseño general de la interfaz.....	119
Figura 39: Prueba unitaria .....	124
Figura 40: PageSpeed Insights análisis móvil .....	137
Figura 41: PageSpeed Insights análisis escritorio .....	138
Figura 42: Lighthouse análisis.....	139
Figura 43: Panel del proyecto de Firebase.....	145
Figura 44: Configuración del proyecto de Firebase .....	145
Figura 45: Panel de gestión de Storage .....	146
Figura 46: Panel de gestión de Database – Colección Conferences .....	146
Figura 47: Panel de gestión de Database – Colección Config.....	147
Figura 48: Panel de gestión de Hosting .....	148
Figura 49: Pantalla de inicio.....	150
Figura 50: Pantalla de registro.....	151
Figura 51: Pantalla de inicio de sesión .....	153
Figura 52: Pantalla de recuperación de contraseña .....	154
Figura 53: Pantalla de perfil de usuario.....	155
Figura 54: Pantalla de pagos y facturas .....	157
Figura 55: Pantalla de subida de artículos .....	159
Figura 56: Pantalla de documentos.....	161
Figura 57: Pantalla de usuarios – rol administrador .....	162
Figura 58: Pantalla de pagos – rol administrador .....	163
Figura 59: Pantalla de autores – rol administrador.....	164
Figura 60: Pantalla de artículos – rol administrador.....	165

Figura 61: Pantalla de conferencias – rol administrador .....	166
Figura 62: Pantalla de documentos – rol administrador .....	167
Figura 63: Pantalla de configuración – rol administrador .....	169

## Índice de tablas

Tabla 1: Coste del material software/hardware .....	44
Tabla 2: Coste del personal .....	45
Tabla 3: Coste total.....	46
Tabla 4: Colección Users.....	105
Tabla 5: Colección Config.....	105
Tabla 6: Colección Conferences.....	106
Tabla 7: Colección Bill.....	106
Tabla 8: Colección Papers.....	107
Tabla 9: Colección Payment_file .....	107
Tabla 10: Colección Authors.....	107
Tabla 11: Pruebas de integración y sistema.....	129
Tabla 12: Pruebas de usabilidad y accesibilidad - Criterios generales .....	131
Tabla 13: Pruebas de usabilidad y accesibilidad - Criterios de identificación e información .....	131
Tabla 14: Pruebas de usabilidad y accesibilidad - Criterios de lenguaje y redacción .....	132
Tabla 15: Pruebas de usabilidad y accesibilidad - Criterios de rotulado .....	132
Tabla 16: Pruebas de usabilidad y accesibilidad - Criterios de navegación .....	133
Tabla 17: Pruebas de usabilidad y accesibilidad - Criterios de layout de la página .....	134
Tabla 18: Pruebas de usabilidad y accesibilidad - Criterios de búsqueda .....	134
Tabla 19: Pruebas de usabilidad y accesibilidad - Criterios de accesibilidad .....	135
Tabla 20: Pruebas de usabilidad y accesibilidad - Criterios de control y retroalimentación.....	136

## **DOCUMENTO 1: MEMORIA**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 1.1 Resumen

El presente trabajo fin de grado consiste en la mejora del rendimiento y ampliación de una web ya existente y en producción, dedicada a la gestión integral de congresos científicos. En particular, la implementación realizada tiene como objeto la gestión del congreso *International Conference on Computational and Mathematical Methods in Science and Engineering* (de aquí en adelante CMMSE).

Parte de la aplicación web ha sido desarrollada por un antiguo alumno de la escuela politécnica de ingeniería de Gijón. Dicha aplicación lleva en uso desde 2014 y ha sustituido a otra aplicación previa, de objetivo similar. Cientos de profesionales científicos han usado la aplicación hasta día de hoy.

En primer lugar, se han actualizado las tecnologías existentes en el proyecto para optimizar la seguridad y el rendimiento tanto del servidor como de la aplicación. En segundo lugar, se ha mejorado la interfaz de usuario y la usabilidad de la aplicación, además de añadir nuevas funcionalidades a petición de la directiva del CMMSE.

La aplicación web funcionará principalmente como una herramienta de gestión integral de congresos científicos, celebrados una vez al año. Los usuarios serán capaces de registrarse para acudir al congreso anual, además de poder subir a la plataforma archivos con contenido científico para presentarlos en dicho congreso. También podrán seguir el proceso de aceptación de artículos a presentar, así como el alta y modificación de sus datos y archivos. Otra de las partes importantes de la aplicación es la del usuario administrador, ya que será el único usuario capaz de ver, modificar o borrar los datos relativos a los usuarios registrados, relacionados al congreso y de configuración de la aplicación.

La herramienta principal que se ha escogido para el desarrollo de esta aplicación web ha sido el framework web *open source* Angular de Google. Más en concreto su versión 5, sacada al mercado el pasado 1 de noviembre de 2017, sustituyendo a su versión anterior Angular 4. Lenguajes como HTML5, Sass, TypeScript, JavaScript o jQuery se han utilizado para satisfacer las necesidades del *front-end* del proyecto, garantizando la mejor experiencia de usuario posible. Por otro lado, la base de datos utilizada para almacenar y gestionar la información de la aplicación web es Firebase Cloud Firestore, también de Google. Por parte del servidor, el lenguaje de programación usado es Node.js por lo que la aplicación ha sido desarrollada a medida para ejecutarse sobre un servidor web o para ser subida a un hosting con unos

requisitos mínimos, teniendo en cuenta las particularidades del congreso a gestionar, el presupuesto y del entorno *hardware* y *software* disponible. Otro punto importante a destacar es el uso de GitHub, repositorio de control de versiones usado a lo largo del desarrollo del proyecto.

## 1.2 Introducción

Este trabajo tiene como objetivo la ampliación de desarrollo y mejora de una aplicación web para la gestión de una conferencia científico-técnica sobre la aplicación de métodos matemáticos en problemas de ciencia e ingeniería que se celebra desde el año 2001 ininterrumpidamente. La aplicación desarrollada tiene como finalidad la gestión completa del congreso y su administración.

El proyecto surge con el objetivo de mejorar una aplicación web existente que la organización venía utilizando en ediciones anteriores del congreso. Para ello se ha realizado un estudio de las funcionalidades cubiertas por la aplicación existente en la actualidad que se encuentra en producción, además de considerar e implementar numerosas mejoras relativas a la usabilidad, a los tiempos de respuesta de la aplicación y al uso de componentes *software* actuales, utilizando librerías de código abierto (*open source*) ampliamente testadas por diferentes comunidades de desarrolladores de aplicaciones.

Inicialmente se plantea la continuación de la aplicación existente ya que la aplicación funciona correctamente a excepción de algunos errores encontrados y mejoras mínimas. Por ello, se realiza durante los primeros meses de desarrollo un mantenimiento y solución de *bugs* para que la edición de 2017 pudiese llevarse a cabo. Gracias a esto, durante las fechas de edición del congreso de ese mismo año, la aplicación estuvo operativa satisfaciendo las necesidades de los asistentes y la directiva.

Posteriormente, se llegó a un punto de no retorno por lo que la aplicación fue reconstruida por completo. Las nuevas funcionalidades requeridas y la actualización de las tecnologías supusieron la necesidad de rehacer la aplicación desde cero. Por supuesto, la nueva aplicación mantendrá los requisitos existentes.

La nueva aplicación desarrollada cubre todos los aspectos principales de la gestión de la conferencia, desde el registro de los asistentes y el envío de los artículos a presentar, hasta la generación

de facturas, justificantes de pago, asistencia y presentación de ponencias, además del envío de notificaciones por correo electrónico tanto a los asistentes como a los gestores de la misma. También cubre otros aspectos administrativos, como la gestión de documentos oficiales del congreso, la generación de listados de usuarios y artículos, la verificación de pagos o la gestión de aspectos logísticos, como la preparación de etiquetas identificativas de distintos tipos para los asistentes y el personal.

En este primer capítulo correspondiente a la memoria del proyecto se plantean en primer lugar los objetivos que se pretenden conseguir con su realización y su alcance. A continuación, se expondrán los estudios y análisis realizados, así como la organización general de documentos que sigue este trabajo fin de grado. Acto seguido se detallarán las conclusiones y ampliaciones tras haber realizado el trabajo de desarrollo del sistema y finalmente se expondrá la bibliografía consultada.

## **1.3 Objetivos y alcance**

A continuación, se expondrán los principales objetivos que se pretenden alcanzar con el desarrollo de este trabajo fin de grado, así como el alcance de éste.

Se dividirá este apartado en dos, mostrando en primer lugar los objetivos principales de la aplicación que se va a desarrollar, los cuales detallarán el servicio que se pretende proporcionar a aquellos usuarios que utilicen la aplicación. En segundo lugar, se mostrarán aquellos objetivos de ámbito académico que se pretenden conseguir durante el desarrollo de este trabajo.

### **1.3.1 Objetivos de la aplicación**

El principal objetivo de este trabajo fin de grado es desarrollar una ampliación de una plataforma web dirigida a las personas que asistan al CMMSE y a todas las personas que la gestionan.

De esta forma los usuarios finales podrán usar la plataforma online con todos los requisitos de seguridad necesarios y con las últimas actualizaciones de todas las tecnologías que se usan en la actualidad.

Su interfaz será mucho más intuitiva permitiendo una navegación más cómoda para todos los usuarios. El aspecto contará con una apariencia moderna respetando las peticiones del CMMSE.

La usabilidad de la plataforma será sencilla, así como práctica, ofreciendo al conferenciante la posibilidad de inscribirse al congreso y administrar todos sus datos y archivos de manera online, sin necesidad de realizar ninguna gestión presencial.

Por otro lado, la dirección del CMMSE utilizará la aplicación para gestionar todo el congreso de forma online. Desde ella podrá consultar rápidamente cualquier dato de los usuarios, de los artículos que se van a presentar o de las facturas generadas, entre otras muchas cosas.

La aplicación web será accesible a través de Internet, tanto para conferenciantes como para el personal del CMMSE, accediendo por medio de un navegador o bien descargando la aplicación desde el repositorio de GitHub en el que se encuentra alojada.

### **1.3.2 Objetivos académicos y de aprendizaje**

Otro de los objetivos principales de este trabajo fin de grado es desarrollar las nuevas funcionalidades de la aplicación con los lenguajes de programación actuales destinados al desarrollo web.

Debido al crecimiento de este tipo de aplicaciones en los últimos años, el conocimiento y uso de este tipo de lenguajes se ha convertido en una de las competencias profesionales más demandadas e importantes en el sector de la programación web hoy en día y por tanto de la informática.

Además, se pretende cumplir con unos objetivos mínimos de seguridad en el servidor de producción y en las tecnologías usadas para el desarrollo de la aplicación.

Algunos de los objetivos académicos y de aprendizaje que se pretenden conseguir con el desarrollo de este trabajo fin de grado son:

- Desarrollar una ampliación de funcionalidades de una aplicación web ya existente, con los lenguajes y tecnologías más modernos del mercado.

- Utilizar el patrón de diseño arquitectónico modelo-vista-controlador (MVC) durante el desarrollo de las nuevas funcionalidades de la aplicación.
- Utilizar un sistema de gestión de bases de datos moderno y escalable con acceso en tiempo real.
- Mejorar la usabilidad y accesibilidad de la aplicación actual, utilizando herramientas de desarrollo web de gran importancia en la actualidad, así como lenguajes de estilos para la interfaz de la aplicación.
- Utilizar una plataforma de control de versiones durante el desarrollo.
- Aplicar las técnicas necesarias para blindar el servidor de producción y la aplicación ante cualquier posible ataque o uso inapropiado del sistema.

## 1.4 Estudios y análisis previos

A continuación, se presentarán aquellas plataformas, tecnologías o herramientas más modernas, potentes y utilizadas hoy en día en el mercado de las aplicaciones web. Se dividirá este apartado en dos; estudios de carácter teórico y estudios de carácter técnico.

### 1.4.1 Estudios de carácter teórico

En los últimos años el desarrollo de aplicaciones web se ha incrementado notablemente, por lo que la mayoría de las aplicaciones que se pueden encontrar en la web han sido mejoradas y actualizadas. Entre ellas, también se incluyen aquellas plataformas web dedicadas a la gestión de congresos científico-tecnológicos. Cada vez son más las personas que utilizan estas plataformas con el objetivo de asistir a congresos internacionales y así gestionar la asistencia a dichos congresos desde su lugar de residencia.

Plataformas como Aconf, EasyChair, o Primoris ofrecen diversos servicios para gestionar eventos o conferencias como la CMMSE. Entre sus servicios está el alojamiento de la aplicación, la gestión de la base de datos, su desarrollo y despliegue y su personalización.

En este tipo de aplicaciones web las organizaciones se registran y contratan un servicio que en algunos casos puede ser gratuito. Se trata de un contrato de software para gestionar conferencias, con el objetivo de ahorrarse el desarrollo de la aplicación por parte de la organización del congreso. Sin embargo, esto tiene la desventaja de no poder ofrecer a los usuarios de la aplicación todas las funcionalidades deseadas o totalmente adaptadas a lo que se necesita. La organización contratante deberá ajustarse al servicio contratado.

Por otro lado, estas plataformas se ofertan en algunos casos gratuitamente con unos servicios mínimos, con la idea de atraer clientes que más tarde necesitarán alguno de los servicios de pago. Es el caso de Aconf y EasyChair.

Otra de las posibilidades existentes en el mercado es el desarrollo de la aplicación de gestión del congreso por parte de la propia organización que lo imparte. Un ejemplo es Sistedes, congreso científico técnico de Ingeniería y Tecnologías del Software.

A continuación, se hará una breve descripción de las plataformas anteriormente mencionadas, así como las herramientas con las que se han desarrollado.

#### **1.4.1.1 Aconf**

Creada oficialmente en el año 2013, esta plataforma es gratuita para congresos académicos. Se trata de un *service as a software (SaaS)*, diseñado para mejorar la eficiencia de la gestión de congresos y conferencias antes, durante y después de su ejecución.

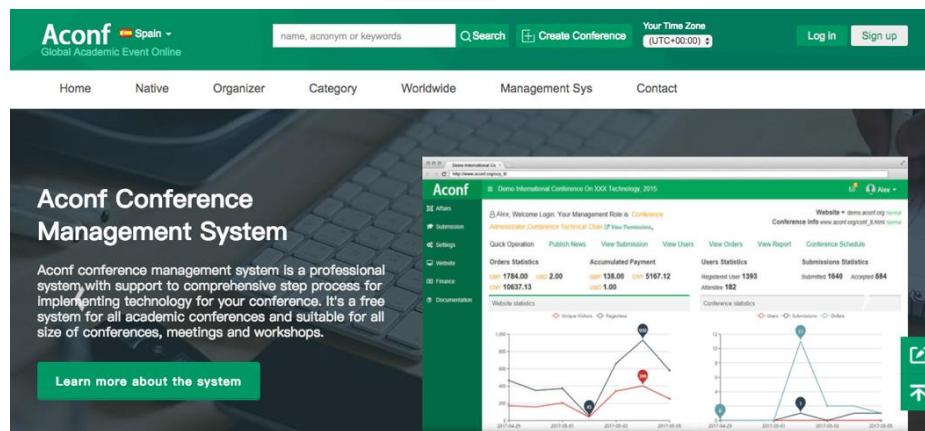


Figura 1: Página inicial de Aconf

Respecto a las tecnologías utilizadas, está basado en una herramienta WYSIWYG (*What You See Is What You Get*). Se trata de un sistema de gestión de contenidos (CMS) que permite escribir un documento viendo directamente el resultado final, es decir, se genera una vista preliminar del código HTML que luego el programa se encarga de generar con su código fuente HTML. En cuanto al resto de tecnologías, las mantienen en secreto.

Entre las ventajas de esta aplicación está la creación de plataformas de gestión de congresos académicos totalmente gratuitos. Su principal objetivo es promover la comunicación académica y construir comunidades académicas.

Una de las desventajas obvias es que no permiten la creación de plataformas no dedicadas a propósitos académicos.

#### 1.4.1.2 EasyChair

Junto a Aconf, EasyChair es otro de los portales web más famosos para gestionar congresos. Se encuentra en el mercado desde el 2002, con más de dos millones de usuarios y más de 60.000 conferencias en sus servidores. Al igual que Aconf, oferta un servicio gratuito para conferencias académicas. En cambio, en su web ofertan un servicio mejor para aquellos que necesiten más prestaciones.

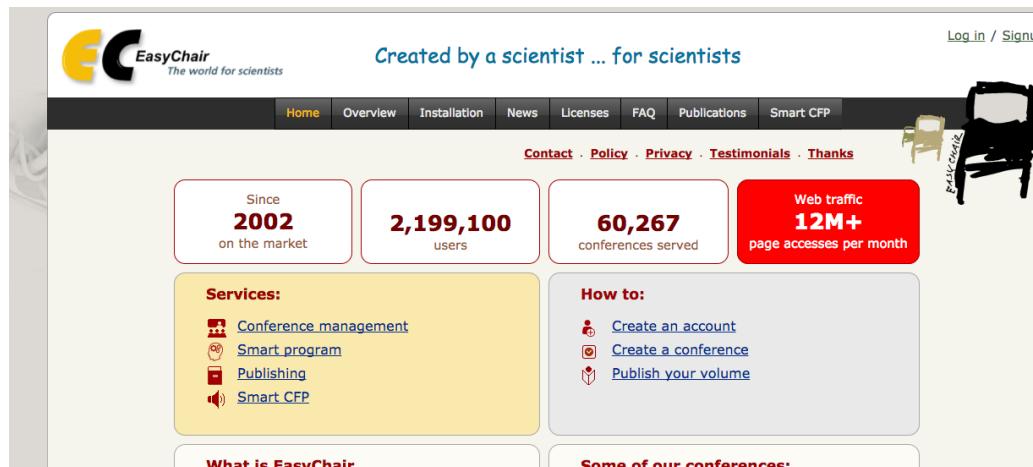


Figura 2: Página inicial de EasyChair

El lenguaje para el desarrollo de esta plataforma es PHP. Al igual que Aconf, el resto de tecnologías usadas se mantienen en secreto.

En relación con Aconf, el propósito de sus servicios es similar, con la diferencia de que EasyChair ofrece un servicio de creación de aplicación para aquellas organizaciones no académicas que deseen organizar un congreso.

#### 1.4.1.3 Primoris

Esta plataforma que se comercializa con el slogan “*La primera de la nueva generación de sistemas de gestión de eventos*”, soporta todo el proceso de gestión de congresos como Aconf y EasyChair. Con una interfaz moderna, gestiona todas las operaciones necesarias para el *front-end* y *back-end* de la aplicación. Organizada en módulos, tiene diferentes alternativas de configuración y personalización.



Figura 3: Página inicial de Primoris

La plataforma ha sido desarrollada mediante el entorno de aplicaciones web ASP.NET usando el *framework* .NET de Microsoft.

La diferencia con las otras plataformas ya comentadas es que Primoris no ofrece públicamente sus tarifas. Es por ello por lo que para contratar uno de sus servicios hay que entrar en contacto con el departamento de eventos. Dos de los congresos que utilizan esta plataforma son:

- <http://www.icsoft.org/>
- <http://www.enase.org/>

#### 1.4.1.4 Sistedes

Esta plataforma perteneciente a la Universidad de La Laguna de Tenerife fue desarrollada con el objetivo de organizar las jornadas de la Sociedad de Ingeniería del Software y Tecnologías de Desarrollo de Software (SISTEDES). Este evento científico-técnico de carácter nacional engloba diferentes conferencias nacionales sobre ingeniería del software, bases de datos, ciencia e ingeniería de servicios y programación y lenguajes.

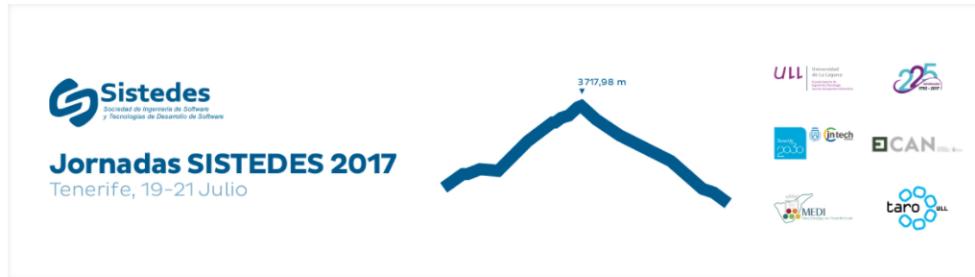


Figura 4: Página inicial de Sistedes

La principal diferencia con el resto de las plataformas es que está gestionada únicamente para la misma organización, por lo que las tecnologías seleccionadas para su desarrollo son las óptimas para la gestión de las conferencias. En cambio, su desarrollo es más largo.

En cuanto a las tecnologías, Sistedes fue desarrollada con WordPress, un sistema de gestión de contenidos (CMS) muy utilizado a día de hoy. WordPress está basado en PHP y MySQL ejecutado en un servidor Apache. En el lado del cliente, WordPress proporciona un amplio abanico de plantillas creadas por diversos desarrolladores. Estas plantillas suelen estar hechas con código JavaScript, jQuery, HTML5 y CSS3. Algunas de estas plantillas son gratuitas pero la mayoría son de pago.

Sin duda, Sistedes es una de las plataformas web más parecidas al tipo de aplicación que se desea desarrollar en este trabajo fin de grado, ya que su finalidad es ajustar las funcionalidades deseadas por el CMMSE al desarrollo, de modo que no haya restricción a la hora de personalizar la interfaz y la lógica de la aplicación.

Tras analizar varias de las plataformas web más conocidas y usadas actualmente para la gestión de congresos, se puede llegar a la conclusión de que una de las mejoras respecto a otras aplicaciones similares será la usabilidad de la aplicación. Para ello se mejorará la interfaz existente y se implementará un estilo definido para toda la aplicación que ayudará a mejorar la experiencia de usuario.

Una vez expuestas las plataformas anteriormente mencionadas, se hará una recopilación de las mejores y más populares herramientas usadas en la actualidad para desarrollar una aplicación web, ya sean lenguajes de programación web, *frameworks* o lenguajes de bases de datos.

## 1.4.2 Estudios de carácter técnico

En primer lugar, se diferenciará entre *framework* y lenguaje de programación. En general, un *framework* es aquel entorno o estructura software que cuenta con aquellos componentes necesarios para realizar una aplicación, un *framework* podrá estar desarrollado por uno o más lenguajes de programación distintos y normalmente están estructurados con patrones de diseño. Estos componentes del *framework* se encargan de facilitar la tarea al desarrollador ya que agilizan el desarrollo de una aplicación.

Por otra parte, los lenguajes de programación son utilizados por los desarrolladores para realizar aquellas acciones que se desean ejecutar, es decir, es el lenguaje que interactúa directamente con la máquina.

Por último, se hará hincapié en los gestores de bases de datos, parte fundamental de cualquier sistema informático, encargados del almacenamiento de la información y de su tratamiento.

A continuación, se expondrán más detalladamente estos conceptos.

### 1.4.2.1 Patrones de diseño

En *frameworks* de ámbito web, es imprescindible utilizar uno de los patrones de arquitectura de software más conocidos, el patrón MVC (modelo-vista-controlador), patrón de diseño por el cual se rigen la mayoría de *frameworks* de desarrollo web.

Como se puede ver en la, la principal característica del patrón MVC es que está basado en la división de capas y/o conceptos, organizando la aplicación en tres modelos separados: modelo, vista y controlador.

#### 1.4.2.1.1 Modelo

Esta capa se encarga de comunicarse con la base de datos mediante las correspondientes peticiones que hace un usuario, pudiendo acceder a los datos, así como también modificarlos, eliminarlos o crearlos si es preciso.

#### 1.4.2.1.2 Vista

Capa encargada de tratar el código de la interfaz de usuario para presentar los datos que el modelo proporciona. Es lo que el usuario verá de la aplicación. Para aplicaciones web actuales, los lenguajes de programación más usados en el mercado y que están estrechamente ligados a la vista son HTML y CSS.

#### 1.4.2.1.3 Controlador

Esta es una de las capas más importantes, ya que sirve de intermediario entre la vista y el modelo. A diferencia del modelo, el controlador no accede a los datos ni los manipula, si no que envía las correspondientes peticiones originadas por los usuarios al modelo cuando estos solicitan información y también se encarga de enviar los respectivos comandos a las vistas asociadas a estos cuando se solicita un cambio en la forma que estas se presentan.

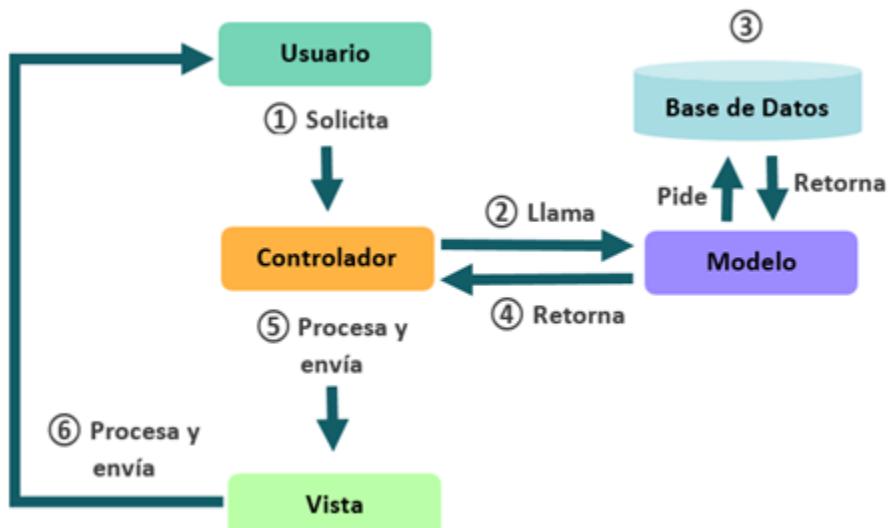


Figura 5: Esquema patrón MVC

Entre las ventajas de utilizar *frameworks* de desarrollo web que utilicen este patrón de diseño se encuentran las siguientes:

- La rapidez con que se desarrolla la aplicación, dado que esta se divide en módulos diferentes e independientes, los cuales se pueden desarrollar en cualquier momento con independencia de los demás.
- Se integra fácilmente con lenguajes de programación web como PHP, JavaScript, HTML, jQuery, etc. y se puede compatibilizar con técnicas de comunicación asíncrona (como AJAX).
- La facilidad para crear múltiples vistas para un modelo. Cada una de ellas será independiente del resto de vistas de la aplicación.
- La organización de código entre las diferentes capas ayuda a una sencilla implementación de nuevas funcionalidades dentro de la aplicación desarrollada.

#### 1.4.2.2 Lenguajes de Programación

Los lenguajes de programación en el ámbito del desarrollo web son aquellos lenguajes que un navegador web podrá entender y, por lo tanto, para los que está preparado para interactuar permitiendo que el usuario visualice desde el navegador un sistema o aplicación que se encuentra alojado en una máquina dentro de la red. Normalmente en un proyecto de desarrollo web se clasifican los lenguajes en dos tipos, lenguajes en el lado del servidor (*back-end*) y lenguajes en el lado del cliente (*front-end*).

Los lenguajes en el lado del servidor (*back-end*) son aquellos lenguajes que se ejecutan en el servidor y se encargan de interactuar con la base de datos, así como también crear las sesiones de usuario o proporcionar los datos necesarios para las vistas de la aplicación. Lenguajes como PHP, Java, Ruby, Python, C# o Node.js son algunos de los lenguajes más utilizados en la actualidad en el lado del servidor.

Por otra parte, los lenguajes en el lado del cliente (*front-end*) son aquellos lenguajes o tecnologías que se ejecutan en el propio navegador web. Los ficheros escritos en estos lenguajes son servidos directamente al navegador para que los interprete. Los navegadores web tienen la capacidad de maquetar y estilizar la vista con la que el usuario está interactuando, con el objetivo de mejorar la experiencia que el usuario tiene con la aplicación. Lenguajes como HTML5, CSS3, JavaScript, TypeScript o librerías como

jQuery y/o *frameworks* como Angular o React son los principales lenguajes del lado del cliente usados en la mayoría de aplicaciones web.

Estos dos tipos de lenguajes no tienen mucho éxito el uno sin el otro, ya que el *back-end* de una aplicación necesitará de los lenguajes del lado del cliente para que el usuario pueda interactuar con la información de la base de datos y el *front-end* necesitará aquella información suministrada por el *back-end* para poder mostrar los resultados de sus interacciones.

A continuación, se listan algunos de los diferentes lenguajes *back-end* con sus correspondientes *frameworks*, los cuales siguen el patrón de diseño MVC y que son muy utilizados en el mercado actualmente:

#### **1.4.2.2.1 PHP**

Para este lenguaje los *frameworks* más utilizados son Laravel, CakePHP, Symfony o CodeIgniter.

#### **1.4.2.2.2 Java**

Struts, Spring o JavaServerFaces son algunos de los *frameworks* más usados.

#### **1.4.2.2.3 Ruby**

Para Ruby el *framework* más usado es Ruby on Rails.

#### **1.4.2.2.4 Python**

Para este lenguaje de programación el *framework* más usado en la actualidad es Django.

#### **1.4.2.2.5 C#**

El *framework* usado por este lenguaje es .NET, propiedad de Microsoft.

#### **1.4.2.2.6 Node.js**

Basado en JavaScript, los *frameworks* más potentes y utilizados son Angular y React.

### 1.4.2.3 Sistema de gestión de bases de datos

Para finalizar este apartado, una de las partes más importantes en una aplicación web es el sistema de gestión de bases de datos, que almacenará aquella información esencial de la aplicación web de la forma más eficiente y segura posible y que deberá estar disponible en el momento que se necesite.

Normalmente en el desarrollo de aplicaciones web las bases de datos suelen ser relacionales (generalmente SQL ya que es el lenguaje más extendido). Sin embargo, en la actualidad muchos desarrolladores optan por una nueva tendencia, las bases de datos no relacionales. A continuación, se exponen algunas de las características más importantes de estos dos tipos:

- Las bases de datos relacionales permiten combinar de una forma eficiente diferentes tablas con el objetivo de extraer información relacionada, mientras que las no relacionales no lo permiten o lo hacen con un peor rendimiento que las relacionales. Ejemplos: PostgreSQL, MySQL, Oracle, etc.
- Las bases de datos no relacionales permiten distribuir grandes cantidades de información de forma más rápida, por lo que este tipo de bases de datos están pensadas para el uso en redes sociales o destinadas al Big Data, es decir, cuando el volumen de datos es de gran tamaño o las estructuras de datos son variables. También cuentan con un escalado horizontal, es decir, al agregar más nodos su rendimiento mejora considerablemente. Ejemplos: MongoDB, Firebase Cloud Firestore, Cassandra, CouchDB, Redis, etc.

Esta breve introducción al estudio de carácter técnico de este trabajo fin de grado no tiene como objetivo debatir cuál de las tecnologías anteriormente indicadas es mejor. Sin embargo, es importante exponer cuáles son las ventajas e inconvenientes de cada una de ellas. El objetivo final es escoger la tecnología adecuada para la realización de la aplicación web.

En el apartado “3.1.4. Análisis de alternativas” del documento número 3 “Requisitos de usuario y análisis de alternativas” se hará una descripción más detallada de algunos de los lenguajes y herramientas anteriormente mencionados, aportando datos y resultados reales, para posteriormente escoger las tecnologías con las que se desarrollará la aplicación web.

## 1.5 Organización del documento

En este apartado se mostrará una lista detallada de los diferentes capítulos que componen este trabajo fin de grado, además de una breve descripción de cada uno de ellos.

### 1.5.1 Documento 1 – Memoria

Se trata del presente capítulo, en él se incluye en primer lugar un resumen del proyecto y una introducción. A continuación, los objetivos y el alcance de este trabajo fin de grado. Después se detallan los estudios y análisis previamente realizados. Por último, se comentarán las conclusiones a las que se ha llegado al finalizar el presente trabajo y se mostrará la organización de los diferentes capítulos que componen este trabajo fin de grado, además de la bibliografía consultada.

### 1.5.2 Documento 2 – Presupuesto y planificación

En este capítulo se detallará la planificación temporal de las diferentes fases del proyecto, desde el inicio hasta su conclusión. A continuación, se describirá el cálculo del coste total del proyecto, así como el coste de los recursos necesarios para el desarrollo de este.

### 1.5.3 Documento 3 – Requisitos de usuario y análisis de alternativas

En este documento se incluirá la especificación de requisitos de usuario y análisis de alternativas de este trabajo fin de grado. Para acabar el documento se expondrá la alternativa seleccionada.

## **1.5.4 Documento 4 – Análisis de requisitos del sistema**

En este capítulo se recogerán los casos de uso de toda la aplicación, así como también se detallarán los requisitos no funcionales y el modelo de datos.

## **1.5.5 Documento 5 – Diseño del sistema**

En este capítulo se expondrán los diagramas de paquetes, de clases, de iteración, así como también el diseño de la base de datos y el diseño de la interfaz.

## **1.5.6 Documento 6 – Pruebas**

En este documento se detallarán las pruebas que se han hecho durante el desarrollo de este: pruebas unitarias, de integración y sistema, de usabilidad y accesibilidad y de rendimiento.

## **1.5.8 Documento 7 – Manuales**

En este último documento se detalla una descripción detallada de los pasos que hay que seguir para instalar la aplicación desde cero, además de una recopilación de las librerías más importantes utilizadas en el desarrollo de la aplicación web.

## **1.6 Conclusiones y ampliaciones**

En el presente trabajo fin de grado se han definido claramente los objetivos y el alcance del mismo, por lo que en este apartado se hablará de las conclusiones a las que se ha llegado, además de una pequeña lista de las posibles ampliaciones que pueden llevarse a cabo en esta aplicación.

## 1.6.1 Conclusiones

Finalizado todo el proceso de implementación y desarrollo asociado a este trabajo fin de grado se puede llegar a la conclusión de que se ha cumplido con los objetivos propuestos inicialmente.

Se han desarrollado una serie de funcionalidades nuevas en la aplicación web y ha sido mejorada la experiencia de usuario para los dos posibles roles definidos (asistentes del congreso y personal de la organización), utilizando tecnologías actualizadas y modernas en el campo de las tecnologías web.

Por otro lado, se han actualizado las tecnologías que se usan en la aplicación web, aumentando la seguridad de la aplicación y su consistencia. Además, la aplicación es escalable en cuanto a cantidad de datos posibles a almacenar.

En definitiva, este trabajo fin de grado ha sido de gran utilidad para la organización ya que se ha conseguido mejorar considerablemente la aplicación y su seguridad con un presupuesto limitado.

## 1.6.2 Ampliaciones

Con el desarrollo de la aplicación actual finalizado, no cabe duda de que aún existen algunas mejoras que se podrían llevar a cabo. Cualquier aplicación o sistema puede ser mejorado y/o ampliado continuamente, por lo que a continuación se detallan algunas de las posibles ampliaciones y mejoras. Entre ellas se encuentran mejoras de las tecnologías, nuevas funcionalidades, mantenimiento continuo del sistema y mejoras de seguridad.

### 1.6.2.1 Mantenimiento continuo de actualizaciones de *software*

El software implementado en esta aplicación proviene de librerías, lenguajes y sistemas operativos de código abierto. Esta comunidad *open source* es muy amplia por lo que continuamente se renueva el software utilizado y todas sus versiones. Lo ideal para un sistema de estas características es la actualización continua de todas las versiones estables que salen al mercado. De esta forma se mejora el rendimiento y la seguridad, ya que estas actualizaciones mejoran ambos aspectos.

### **1.6.2.2 Creación de aplicación móvil nativa**

Una de las posibles mejoras que más fundamento tiene es la creación de una aplicación móvil nativa para Android e iOS del sistema. Una de las grandes ventajas de usar Angular es que se adapta con mucha facilidad a la responsividad de las pantallas móviles. Además, existen algunos *frameworks* como Ionic, que permiten adaptar la mayoría del código web de una aplicación en Angular para convertirla con unos simples comandos de consola a una aplicación móvil nativa.

Por otro lado, otra ventaja muy importante es el uso de Firebase. Los datos de la base de datos se encuentran alojados en la nube y gracias a su arquitectura, los datos pueden ser servidos en tiempo real tanto a aplicaciones web como a aplicaciones nativas móvil Android e iOS.

### **1.6.2.3 Internacionalización**

La aplicación cuenta con una internacionalización mínima, puede seleccionarse inglés o español como idioma dentro de la aplicación. Sin embargo, muchos de los usuarios de la aplicación tienen otra lengua nativa.

Esta mejora se basa en la inclusión de otros idiomas en la internacionalización, además de modificaciones para idiomas como el árabe, que se leen en modo espejo respecto a los idiomas occidentales como el español y por lo que habría que realizar cambios en la interfaz para que los textos y campos del sistema se pudieran leer de derecha a izquierda en lugar de izquierda a derecha.

### **1.6.2.4 Creación de un Chatbot en el sistema**

Una de las tendencias actuales en el mercado web es el uso de un Chatbot dentro de las aplicaciones web. Se trata de un componente que ayude al usuario a realizar acciones dentro del sistema con un agente de IA que responda a las preguntas o cuestiones que el usuario se plantea. Dialogflow de Google es una de las plataformas que más valor tienen a día de hoy. Con esta herramienta se pueden crear agentes que interactúen con el usuario si este lo desea.

## 1.7 Bibliografía

A continuación, se incluye la bibliografía consultada en la realización de este trabajo fin de grado. En primer lugar, se hará referencia a los libros y/o artículos. En el siguiente punto, se detallarán los sitios web consultados.

### 1.7.1 Libros y artículos

Durante el desarrollo del proyecto se han consultado los siguientes libros, ya sea en versión papel o en versión digital:

- **Guía de PHP:** Diego Lázaro – Leanpub, 2015.
- **Ng-book: The Complete Guide to Angular 4:** Nathan Murray – FULLSTACK.io, 2017.
- **Angular: From Theory To Practice:** Asim Hussain – Code Craft , 2016.
- **JavaScript and JQuery: Interactive Front-End Web Development:** Jon Duckett – Wiley, 2014.
- **Firebase Cookbook:** Houssem Yahiaoui – Packt, 2017.

### 1.7.2 Referencias en internet

En este apartado se detallarán los sitios webs consultados para el desarrollo del proyecto, así como para la documentación y la instalación de las herramientas utilizadas:

- **Angular:** <https://angular.io/>
- **Angular Material:** <https://material.angular.io/>
- **Node.js:** <https://nodejs.org/es/>
- **Firebase:** <https://firebase.google.com/>
- **Visual Studio Code:** <https://code.visualstudio.com/>
- **Sass:** <https://sass-lang.com/>

- **Medium:** <https://medium.com/>
- **Express Server:** <http://expressjs.com/es/>
- **NPM Package manager:** <https://www.npmjs.com/>
- **Udemy – Cursos online:** <https://www.udemy.com/>
- **Librería jsPDF:** <https://parall.ax/products/jspdf>
- **Plantillas HTML y Bootstrap:** <https://startbootstrap.com/>
- **Programa virtualización de máquinas VirtualBox:** <https://www.virtualbox.org/>
- **Sistema operativo CentOS:** <https://www.centos.org/>
- **Servidor Apache:** <https://httpd.apache.org/>
- **Lenguaje PHP:** <http://php.net/>
- **Doctrine ORM:** <http://www.doctrine-project.org/>
- **Tutoriales web W3Schools:** <https://www.w3schools.com/>
- **Lenguaje de bases de datos MySQL:** <https://www.mysql.com/>
- **Librería Bootstrap:** <http://getbootstrap.com>
- **Librería jQuery:** <https://jquery.com/>
- **Lenguaje JavaScript:** <https://www.javascript.com/>
- **Librería Datatables:** <https://datatables.net/>
- **GitHub – Control de versiones:** <https://github.com/>
- **Comunidad de desarrolladores:** <https://es.stackoverflow.com/>
- **Información, definiciones y significados:** <https://www.wikipedia.org/>
- **Artículos y blogging sobre infraestructura web:** <https://geekflare.com/>
- **Congreso Sistedes:** <https://fg.ull.es/sistedes2017/>
- **Congreso ICSOFT:** <http://www.icsoft.org/>
- **Congreso ENASE:** <http://www.enase.org/>
- **Sistema de gestión de congresos (Aconf):** <https://www.aconf.org/>
- **Sistema de gestión de congresos (EasyChair):** <http://easychair.org/licenses.cgi>
- **Sistema de gestión de congresos (Primoris):** <https://www.institicc.org/Primoris/Default.aspx/>
- **Comunidad *open source* de seguridad del software:** <https://www.owasp.org/>
- **Plataforma de *cloud computing* con comunidad de soporte:** <https://www.digitalocean.com>

- **Artículo de usabilidad web:** <https://www.crazyegg.com/blog/principles-website-usability/>
- **Artículo de métodos de usabilidad:** <http://www.usabilityfirst.com/usability-methods/>
- **Herramienta de medición de usabilidad:** <http://www.tawdis.net>
- **Guía heurística de usabilidad y accesibilidad:**  
<http://www.nosolousabilidad.com/articulos/heuristica.htm>
- **Herramienta Lighthouse:** <https://developers.google.com/web/tools/lighthouse/>
- **Herramienta PageSpeed:** <https://developers.google.com/speed/>

## **DOCUMENTO 2: PLANIFICACIÓN Y PRESUPUESTO**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 2.1 Introducción

El presente documento se dividirá en dos partes, siendo la primera la planificación temporal del desarrollo de este trabajo fin de grado y la segunda parte una estimación detallada del coste total de la realización de este.

## 2.2 Planificación temporal

El inicio de este proyecto comenzó el 16 de noviembre de 2016, con fecha prevista de finalización para el 15 de mayo de 2018, sumando una duración total de 18 meses aproximadamente. El desarrollo se ha producido con una hora al día de media, sin incluir festivos ni fines de semana o periodos vacacionales.

El desarrollo principal se ha dividido en cuatro fases principales: fase de análisis, fase de diseño, fase de implementación y fase de pruebas. Además, se ha planificado también la parte correspondiente a la documentación. A continuación, se detallan distintas imágenes pertenecientes al diagrama Gantt creado durante el desarrollo del proyecto, con el objetivo de establecer una planificación detallada y concisa del trabajo.

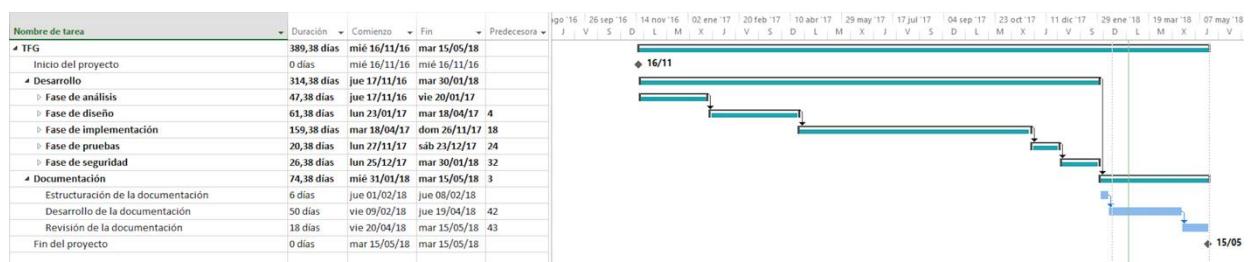


Figura 6: Diagrama Gantt – Visión global

## 2.2.1 Fase de análisis

Se trata de la fase de análisis del proyecto. En primer lugar, se detalla la planificación de los estudios y análisis previos. Más tarde los requisitos de usuario y el análisis de alternativas. Por último, se planifica el análisis de requisitos del sistema.

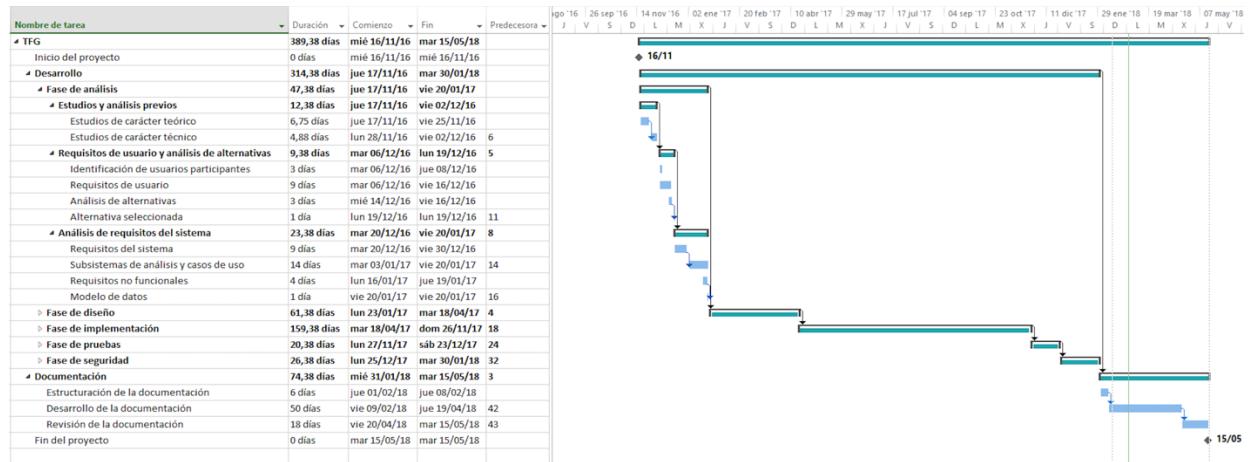


Figura 7: Diagrama Gantt - Fase de análisis

## 2.2.2 Fase de diseño

En esta fase se planifican los diseños específicos de ingeniería del software, diseño de la base de datos y de la interfaz de usuario.

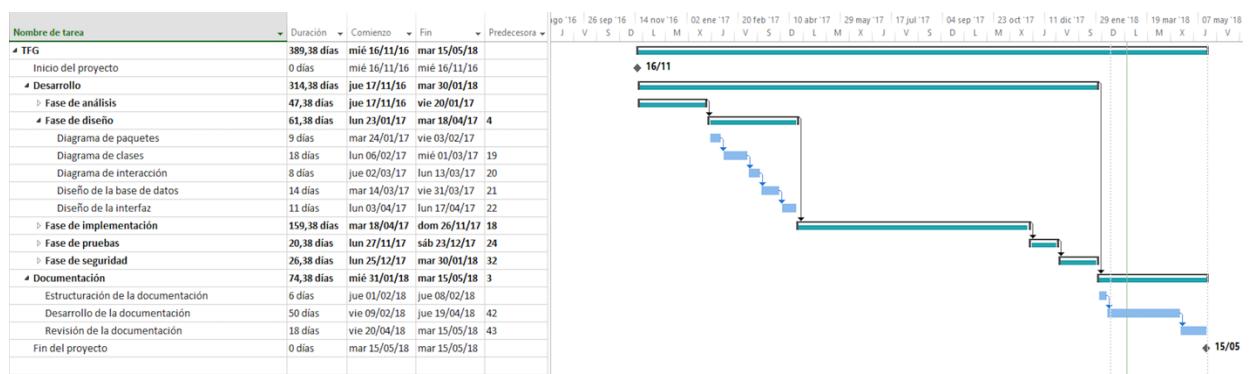


Figura 8: Diagrama Gantt - Fase de diseño

## 2.2.3 Fase de implementación

Es la fase de creación de la aplicación y de su estructura, y de programación de todas las funcionalidades necesarias en la mejora de la web.

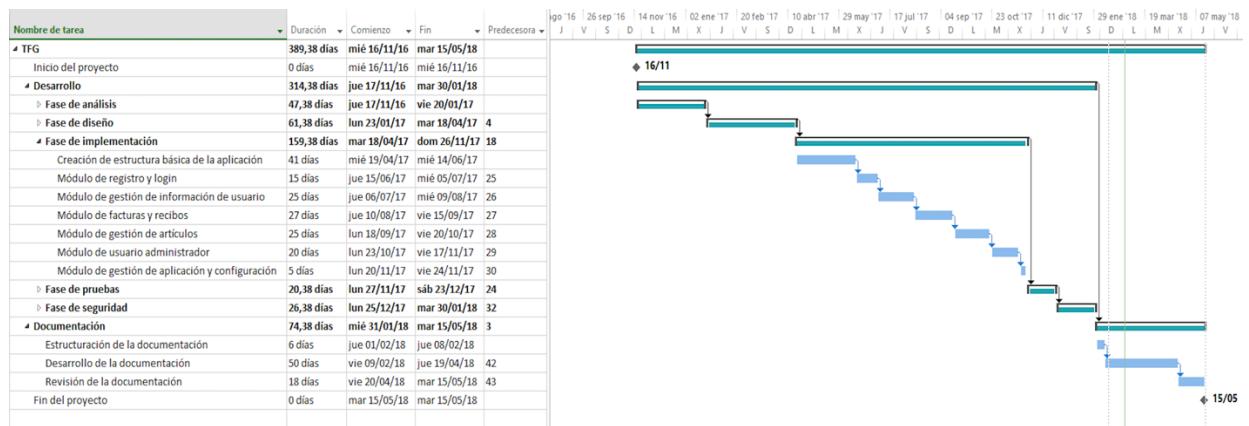


Figura 9: Diagrama Gantt - Fase de implementación

## 2.2.4 Fase de pruebas

Se trata de la fase de pruebas de la aplicación. Pruebas funcionales y no funcionales.

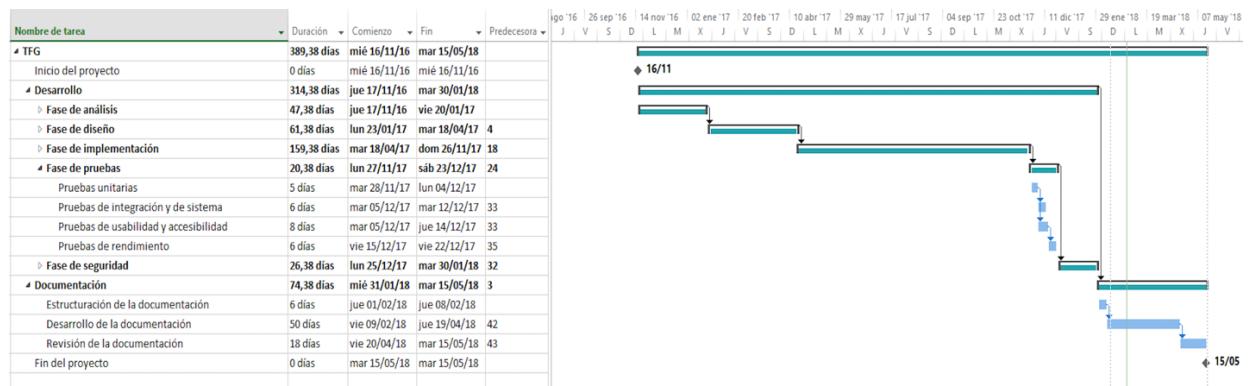


Figura 10: Diagrama Gantt - Fase de pruebas

## 2.3 Presupuesto

El presupuesto del presente trabajo fin de grado se divide en dos partes. La primera de ellas corresponde al coste de los elementos hardware y software del sistema. La segunda parte corresponde al coste del personal involucrado en el desarrollo del mismo. En los siguientes apartados se presentará una estimación detallada del coste de las dos partes anteriormente citadas y del coste total que conlleva el desarrollo de este trabajo fin de grado.

### 2.3.1 Coste del material software/hardware

El coste del material software y hardware utilizado en el desarrollo de la aplicación web se realizará en este primer apartado.

El desarrollo de la aplicación web ha sido íntegramente desarrollado con tecnologías *open source* por lo que la parte software no tiene coste alguno. Por otro lado, el IDE (entorno integrado de desarrollo) usado en el desarrollo de todo el proyecto ha sido el Visual Studio Code, un editor de código gratuito que tampoco supone ningún coste al proyecto.

Un aspecto importante a destacar es que el equipo con el que se desarrolla la aplicación web y con el que se realiza la documentación solo se utilizará durante este periodo de tiempo. Por tanto, el coste que supone utilizar el equipo durante el tiempo de desarrollo y documentación del proyecto se debe calcular a partir de su coste de amortización. Según la Agencia Tributaria y como se especifica en la tabla de coeficientes de amortización lineal, aquellos equipos electrónicos e informáticos catalogados como equipos para procesos de información tienen un gasto de amortización del 25% anual (con un máximo de 8 años). Por lo tanto, el equipo MacBook Pro 15'' del año 2016 con un coste total de compra de 2.255,59€ (IVA incluido), tiene un gasto de amortización asociado de:

$$2.255,59\text{€} \times 25\% = 563,89\text{€} \text{ anuales.}$$

Como se ha utilizado 1,67 años, se deduce un gasto final de:

563,89€ x 1,67 años = **939,82€** deducibles de gastos de amortización.

Además del equipo, se tiene que añadir el gasto de la conexión a internet, por lo que se ha incluido en el presupuesto una tarifa de red necesaria para el desarrollo de la aplicación (descarga de librerías, actualizaciones, consulta de documentación, etc.).

Cabe destacar que, si se decide alojar la aplicación web en un hosting diferente o si se supera la capacidad gratuita del hosting actual, se debería contratar un servicio de hosting adecuado y un dominio específico para alojar la aplicación, lo cual supondría un coste adicional. Los siguientes enlaces muestran ejemplos de tarifas competitivas actuales:

- <https://raiolanetworks.es/hosting/>
- <https://sered.net/hosting-espa%C3%B1a-ssd>
- <https://www.adjenet.net/hosting>
- <https://firebase.google.com/pricing/>

#	Descripción	Unidad	Cantidad	Precio/Ud. (€)	Total
1	MacBook Pro 15'' 2016 Intel i7 - 16GB	Ud.	1	939,82 €	939,82 €
2	Tarifa de red	Mes	20	25 €	500 €
3	Sistema operativo (macOS High Sierra)	Ud.	1	0 €	0 €
4	Framework (Angular)	Ud.	1	0 €	0 €
5	IDE (Visual Studio Code)	Ud.	1	0 €	0 €
				<b>SUBTOTAL</b>	<b>1.439,82 €</b>

Tabla 1: Coste del material software/hardware

## 2.3.2 Coste del personal

Para calcular el coste del personal involucrado en el desarrollo del proyecto se hace una distinción de dos perfiles distintos. Por un lado, el perfil de analista informático será el encargado de desarrollar la fase de análisis y el diseño del proyecto, así como su inclusión en la documentación. Por otro lado, el perfil de desarrollador informático será el encargado de la fase de implementación y pruebas, así como también su inclusión en la documentación. Además, el desarrollador será el encargado de realizar los manuales de usuario e instalación.

Para calcular el precio final de este apartado se ha estimado una jornada de trabajo de 2 horas diarias durante toda la duración del proyecto (excluyendo días no laborales y festivos). El 70% del tiempo se ha destinado a la realización de las tareas correspondientes de cada fase y el 30% restante a la documentación. El precio para el personal es de 15€/hora.

#	Descripción	Cantidad (días)	Precio/Ud.	Total
1	Analista informático	-	-	-
1.1	Fase de análisis	45	30 €	1.350 €
1.2	Fase de diseño	53	30 €	1.590 €
1.3	Documentación	50	30 €	1.500 €
2	Programador informático	-	-	-
2.1	Fase de implementación	160	30 €	4.800 €
2.2	Fase de pruebas	30	30 €	900 €
2.3	Documentación	55	30 €	1.650 €
			<b>SUBTOTAL</b>	<b>11.790 €</b>

Tabla 2: Coste del personal

### 2.3.3 Coste total

Por último, en este apartado se calculará el coste total del proyecto, siendo este la suma del coste de material software/hardware y el coste del personal involucrado. Nótese que además se aplica el impuesto sobre el valor añadido (I.V.A.) al subtotal calculado.

#	Descripción	Precio	Total
1	Coste del material (software/hardware)	1.439,82 €	1.439,82 €
2	Coste del personal	11.790 €	11.790 €
		<b>SUBTOTAL</b>	<b>13.229,82 €</b>
		I.V.A (21%)	2.778,26 €
		<b>TOTAL</b>	<b>16.008,08 €</b>

Tabla 3: Coste total

## **DOCUMENTO 3: REQUISITOS DE USUARIO Y ANÁLISIS DE ALTERNATIVAS**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 3.1 Introducción

Este tercer documento desglosa con detalle la parte de análisis del proyecto, siendo una de las partes más importantes de cualquier proyecto de desarrollo software. Se definirán los diferentes roles que puede tener un usuario de la aplicación, así como los requisitos de usuario, un estudio más detallado de las tecnologías y herramientas a utilizar y finalmente la alternativa seleccionada.

## 3.2 Identificación de usuarios participantes en el sistema

La aplicación contará con dos roles de usuario bien definidos. Ambos podrán usar la misma aplicación, pero los roles son independientes. No compartirán ninguna funcionalidad, excepto la página de inicio de la aplicación y el *login*.

### 3.2.1 Rol de participante en el congreso

La aplicación web permitirá crear un perfil de usuario para todos los participantes del congreso anual a través de un registro en el que deberán introducir unos datos mínimos de contacto. En el perfil creado los usuarios podrán modificar sus datos personales, modificar la contraseña, adjuntar justificante de pago, añadir datos para la factura, subir artículos y descargar documentos asociados al congreso.

### 3.2.2 Rol de usuario administrador

El usuario administrador podrá acceder a la aplicación a través del *login*. Las credenciales (usuario y contraseña) están predefinidas, pero podrán ser modificadas dentro de la aplicación únicamente por el usuario administrador. Además, en este rol se podrá ver y modificar todos los usuarios registrados, los artículos subidos, los pagos y su aceptación, las conferencias disponibles y los autores de los artículos. Por último, el usuario administrador podrá descargar todos los documentos asociados a la conferencia y cambiar la configuración global de toda la aplicación y de la conferencia.

### 3.3 Requisitos de usuario

Los requisitos de usuario tienen una importancia muy alta en el desarrollo de cualquier producto software. Son declaraciones en lenguaje natural de los servicios que un usuario espera y que el sistema debe proveer. Además, especifican con detalle todas las restricciones de la aplicación.

A continuación, se detallarán los requisitos de usuario divididos en tres apartados: requisitos de usuario comunes a los dos roles de usuario existentes, requisitos de usuarios propios del rol de participante en el congreso y por último requisitos de usuario propios del rol de usuario administrador.

#### 3.3.1 Requisitos de usuario comunes

- *Login* en la aplicación con el nombre de usuario y la contraseña. El usuario accederá a las funciones únicas de su rol y tendrá acceso solo a sus datos.
- *Logout* de la sesión de usuario. El usuario saldrá a la pantalla de *login* y cerrará su sesión, por lo que tendrá que volver a iniciar sesión si desea volver a la pantalla anterior.
- Navegación a la web organizadora del congreso. Desde la pantalla inicial el usuario podrá navegar a dicha web.
- Cambiar idioma de la aplicación. Por defecto la aplicación se encontrará en inglés, pero el usuario podrá cambiar el idioma desde cualquier pantalla y en cualquier momento.

#### 3.3.2 Requisitos de usuario – rol de participante en el congreso

- Registro en la aplicación. Se deberán añadir los datos personales del usuario, así como una dirección email válida y una contraseña con unos requisitos de seguridad mínimos. Algunos de los campos del registro seránopcionales, pero para validar el registro se deberán aceptar las condiciones del congreso. El registro no se efectuará a menos que se rellenen los campos obligatorios del usuario. El usuario recibirá un correo informando de su registro en la aplicación.

- Modificar los datos personales del usuario, excepto la dirección email. No se podrá dejar ningún campo obligatorio del usuario vacío durante la modificación.
- Modificar la contraseña del usuario. El usuario deberá introducir su contraseña actual y su nueva contraseña con una verificación de seguridad.
- Solicitar nueva contraseña al sistema. Si el usuario no recuerda la contraseña, deberá introducir su correo electrónico y recibirá una nueva contraseña aleatoria vía email.
- Subir justificante de pago a la aplicación. El usuario podrá subir un justificante de pago almacenado de forma local y podrá modificarlo siempre que lo desee. El sistema recibirá un email indicando que el usuario ha subido un justificante de pago.
- Añadir datos de factura, todos sus campos serán obligatorios para poder almacenarlos en la aplicación. La factura asociada al usuario tendrá estos datos.
- Subir artículo a la aplicación añadiendo los datos relativos al artículo y a los autores del artículo. Esta acción enviará un correo a la dirección del congreso indicando la subida de un nuevo artículo.
- Modificación de los datos y el archivo de los artículos subidos por el usuario. Dependiendo del estado del artículo, es posible que la modificación no esté habilitada. Todos los campos obligatorios deben estar cubiertos para la modificación de cualquier artículo. Además, dependiendo del estado del artículo dichos cambios podrán implicar el envío de un email al sistema indicando los cambios realizados por el usuario. Dependiendo del estado del artículo, este podrá ser eliminado por el usuario.
- Descarga de los documentos asociados al congreso para el usuario. Los documentos descargables son: factura, certificado de ponencia y certificado de asistencia. Estos documentos serán descargables cuando se haya validado el pago por parte del administrador. Además, solo se habilitará el certificado de ponencia si el usuario ha subido algún artículo y este ha sido aceptado por el congreso.

### 3.3.3 Requisitos de usuario – rol de usuario administrador

- Visión global y modificación de todos los usuarios registrados en el sistema. Se podrán modificar todos los datos de los usuarios excepto el email y la contraseña (no accesible). También se podrá eliminar el usuario.
- Visión global y modificación de pagos de los usuarios registrados. Se podrá descargar el justificante de pago de cada usuario, se podrá cambiar el precio a pagar por la asistencia al congreso para cada usuario, se podrá verificar la validez del justificante de pago y se podrá descargar la factura individual de cada usuario.
- Visión global de los autores principales de los artículos subidos. Se indicará el número de artículos en los que participa cada autor y el número de posters. No se podrán hacer modificaciones en esta parte.
- Visión global y modificación de los datos de todos los artículos. Se podrá descargar el documento subido por cada usuario. También se podrá cambiar el estado de un artículo, lo que enviaría un email tanto al usuario como al sistema indicando el cambio de estado del artículo. Además, se podrá modificar todos los datos de cada artículo en cualquier momento. Por último, se podrá eliminar el artículo, por lo que el usuario no verá dicho artículo en la aplicación.
- Visión global y modificación de conferencias. Se mostrarán todas las conferencias existentes para el congreso actual, el número de artículos pertenecientes a cada conferencia y el número de posters. Se podrá modificar el título de la conferencia, se podrá añadir una nueva conferencia y se podrá eliminar una conferencia ya existente.
- Descarga de documentos relativos a la conferencia. Por un lado, se podrá descargar todos los certificados de asistencia, todos los certificados de ponencia y todas las facturas de todos los usuarios registrados, todos ellos en formato PDF. Por otro lado, se podrá descargar en formato Excel la lista de todos los usuarios registrados, la lista de todos los artículos subidos y la lista de todos los autores participantes. Por último, se podrá generar un PDF con etiquetas identificativas para todos los usuarios asistentes al congreso. Estas etiquetas podrán ser modificadas en color y

tamaño a elección del usuario administrador y podrán llevar un texto en lugar del nombre de los usuarios participantes.

- Visión general y modificación de la configuración de la aplicación. Se podrá consultar y modificar todos los parámetros de configuración como el usuario y la contraseña de administrador, el año de la conferencia, las fechas de inicio y clausura, el lugar donde será celebrado el congreso, la dirección web oficial del congreso y otros datos. Además, será posible modificar la configuración de correo electrónico y/o cerrar o abrir el servicio de correos. Por último, se podrá reiniciar la base de datos de la aplicación y cerrar o abrir tanto el acceso general como el periodo de subida de artículos a la aplicación.

## 3.4 Análisis de alternativas

En este apartado se hará un estudio de las diferentes herramientas y lenguajes más usados hasta hoy en la programación web. En primer lugar, se detallará una breve descripción de los lenguajes de programación y sus correspondientes *frameworks* que forman parte del lado del servidor (*backend*). En segundo lugar, se hará lo correspondiente con los lenguajes de programación y los *frameworks* asociados en la parte del cliente (*frontend*). Por último, se mostrará un resumen y descripción de las bases de datos más usadas.

### 3.4.1 Tecnologías *backend*

La parte del lado del servidor o tecnología *backend* es la capa intermedia entre la base de datos y el código que el navegador web va a ejecutar. Estos lenguajes de programación son ejecutados desde el servidor donde está alojada la aplicación. Se encargan de tratar con los datos de la base de datos y de enviarlos al *front* para que estos sean mostrados en el navegador.

### 3.4.1.1 PHP

A día de hoy es el lenguaje más usado para el desarrollo web. Se trata de un lenguaje de código abierto con más de 20 años de antigüedad, basado en los lenguajes C# y PERL. Cuenta con una gran comunidad de desarrolladores y usuarios desde sus inicios, además de numerosas aplicaciones web de éxito en las que se ha utilizado este lenguaje (Yahoo, Wikipedia, Facebook, Wordpress o Flickr).

Entre las ventajas de este lenguaje está la continua demanda por parte de las empresas y su compatibilidad con los sistemas operativos y plataformas del mercado. Además, cuenta con varios *frameworks* de desarrollo como Laravel, Symfony2, Yii, etc.



Figura 11: Logo PHP

### 3.4.1.2 Ruby

Lanzado al mercado en 1995, Ruby es un lenguaje de programación orientado a objetos. Ruby on Rails es el *framework* de desarrollo que se utiliza para construir aplicaciones web bajo el lenguaje Ruby. Este *framework* es de código abierto y, como PHP, cuenta con muchos usuarios a día de hoy por lo que la comunidad también es extensa. Twitter, Github, LinkedIn o SlideShare son algunas de las aplicaciones desarrolladas en Ruby on Rails. Entre las ventajas de utilizar este *framework* para el desarrollo web está la rapidez y facilidad que aporta el entorno de trabajo para construir la aplicación, así como también la sencillez para incluir paquetes de librerías.



Figura 12: Logo Ruby

### 3.4.1.3 Python

Se trata de un lenguaje interpretado y multiparadigma creado a finales de los 80. Python se caracteriza por ser un lenguaje con tipificación dinámica y multiplataforma. La comunidad de desarrolladores también es muy alta. Es de código abierto y su filosofía es favorecer el desarrollo de código legible.

No es el lenguaje más utilizado en la programación web, pero Python cuenta un potente *framework* web de alto nivel con una amplia documentación llamado Django. Instagram, Pinterest o The New York Times son algunas de las aplicaciones web que se han desarrollado con este lenguaje.



Figura 13: Logo Python

### 3.4.1.4 ASP.NET

ASP.NET es un entorno para aplicaciones web propiedad de Microsoft. Creado en 2002, es el sucesor de ASP. En este entorno se permite usar cualquier lenguaje admitido en el .NET *framework*, siendo el más común C# o C Sharp. Al ser propiedad de Microsoft, no es de código abierto por lo que se requiere de una licencia para su uso. Entre las ventajas de su uso está la compatibilidad con módulos de Microsoft como SQL Server, o el lenguaje Visual Basic. Además, este *framework* también ofrece tecnologías *frontend*.



Figura 14: Logo ASP.NET

### 3.4.1.5 Java EE

Java Enterprise Edition, Java EE o J2EE es una plataforma de programación destinada a desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. En ella se permiten utilizar arquitecturas de N capas distribuidas y se apoya en componentes modulares ejecutados sobre un servidor de aplicaciones. Los beneficios de usar esta tecnología son: código abierto en la gran mayoría de herramientas que la usan, miles de librerías para extender la plataforma y simplificar el desarrollo.

Algunas de los *frameworks* más usados y destinados al desarrollo web que se integran con los estándares de la plataforma Java EE son: JavaServer Faces framework, Spring framework o Struts framework.



Figura 15: Logo Java EE

### 3.4.1.6 Node.js

Node.js es un entorno en tiempo de ejecución multiplataforma y de código abierto creado en 2009 orientado en el lado del servidor y que utiliza ECMAScript (JavaScript) como lenguaje. Asíncrono y con una arquitectura orientada a eventos, está basado en el motor V8 de Google. Creado específicamente para ser útil en la creación de programas de red altamente escalables como los servidores web. Una de las ventajas de Node.js es que puede ser combinado con una base de datos y mediante elementos JSON permite desarrollar un entorno JavaScript unificado. Adaptando los patrones para desarrollo del lado del servidor facilitando la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor. Se trata del más moderno comentado hasta ahora y el que más futuro tiene de cara a las tecnologías web. A pesar de su juventud, la comunidad de desarrolladores es muy activa.



Figura 16: Logo Node.js

### 3.4.2 Tecnologías *frontend*

Las tecnologías *frontend* son la capa más alta de una aplicación web. Son las encargadas de recibir los datos finales desde las capas más bajas y construir mediante diversos lenguajes una interfaz que el navegador mostrará al usuario. Es importante destacar que las tecnologías de *frontend* son muy importantes ya que el usuario final interactúa directamente con esta capa, por lo que la interfaz debe ser moderna, usable y accesible. A continuación, se hará un resumen detallado de las tecnologías que más se utilizan hoy en día.

#### 3.4.2.1 HTML5

HTML (HyperText Markup Language) es un estándar que define una estructura básica y un código por el que se define el contenido de una página web. El principal objetivo de este lenguaje es dar formato a los diversos contenidos de una página web para que un navegador los interprete. HTML5 es la última versión de este lenguaje y desde el 2014 es pública.



Figura 17: Logo HTML5

### 3.4.2.2 XML

XML (eXtensive Markup Language) es un metalenguaje que permite definir lenguajes de marcas. Fue desarrollado por el W3C (World Wide Web Consortium) y sirve para organizar contenidos y describir la información de una página web. Aunque sus características puedan ser similares a las del HTML, en general el lenguaje XML suele funcionar como un complemento del lenguaje HTML.



Figura 18: Logo XML

### 3.4.2.3 JavaScript

JavaScript o JS, es un lenguaje de programación interpretado. Principalmente se utiliza en el lado del cliente implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque también puede ser usado en el lado del servidor, como es el caso de Node.js. Mayormente influenciado por Java, también tiene influencias del C, Python o Scheme. Se ha convertido en uno de los lenguajes más populares de la programación web y con él se pretende mejorar la experiencia del usuario.



Figura 19: Logo JavaScript

### 3.4.2.4 jQuery

jQuery es una librería multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM (Doment Object Model), manejar

eventos, desarrollar animaciones y en general contenido dinámico a la aplicación web. Desarrollada en 2006 y de código abierto, se trata de la librería JavaScript más utilizada.



Figura 20: Logo jQuery

### 3.4.2.5 TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado y mantenido por Microsoft sacado al mercado en 2012. Se trata de un superconjunto de JavaScript, que básicamente añade tipificación estática y objetos basados en clases, lo que le hace un lenguaje orientado a objetos. Cualquier código JavaScript funciona en TypeScript. A través de un compilador, el código de TypeScript es traducido a JavaScript para que los navegadores puedan ejecutar dicho código.



Figura 21: Logo TypeScript

### 3.4.2.6 Angular

Angular es un *framework* que ha evolucionado mucho en los últimos años. Es propiedad de Google y es de código abierto. Su primera versión llamada AngularJS fue lanzada en 2010. El lenguaje en el que se basa su primera versión es JavaScript. En el año 2016 se lanzó su segunda versión llamada Angular 2. Actualmente se ha lanzado la versión Angular 6 aunque comúnmente se le denomina Angular ya que desde su versión dos las actualizaciones no cambiaron la arquitectura del *framework* por lo que, comúnmente, se llama Angular a cualquier versión posterior. Cada cierto tiempo, los desarrolladores lanzan una nueva versión del *framework* con nuevas características y mejoras. Desde la segunda versión el lenguaje en el que se basa es TypeScript. Aún así, Angular sirve al navegador el código TypeScript de la aplicación traducido a JavaScript para que los navegadores puedan ejecutar dicho código. Angular sigue el patrón arquitectónico MVC ya que se basa en la creación de componentes y servicios. Cada componente

tiene un archivo HTML, un archivo Typescript y un archivo CSS o SCSS. Esto permite organizar el código y es Angular el que se encarga de juntar todos los ficheros en la compilación.



Figura 22: Logo Angular

### 3.4.2.7 CSS3

CSS (Cascading StyleSheets) es un lenguaje utilizado para definir los estilos de una aplicación web. Este lenguaje se suele combinar con HTML para definir la interfaz del usuario. Actualmente se utiliza su última versión CSS3. Se basa en una modularización de ficheros que contienen nombres de clases incluidos en el DOM (Definition Object Model). Las clases son aplicadas siguiendo un sistema de puntuación. La clase con mayor valor es la que hace efecto en el DOM, invalidando las clases que afectan al mismo elemento en el DOM con menor valor.



Figura 23: Logo CSS3

### 3.4.2.8 Sass

Sass es un metalenguaje de CSS. Es un lenguaje de script que es traducido a CSS. La sintaxis más reciente, SCSS, usa el formato de bloques como CSS. Se trata de un lenguaje precompilado de hojas de estilo muy útil en la actualidad para la maquetación de aplicaciones web. A diferencia del lenguaje CSS, Sass permite la definición de variables en sus ficheros, la creación de *mixins*, anidamiento de clases y

herencia de selectores entre otras muchas cosas. Estos ficheros son precompilados a CSS para servirlos posteriormente al navegador como clases CSS comunes.



Figura 24: Logo Sass

### 3.4.2.9 Bootstrap

Bootstrap es un framework de código abierto lanzado en el año 2011. Se trata de una de las mejores herramientas de maquetado y diseño para aplicaciones web que existen en la actualidad. Bootstrap proporciona un conjunto de plantillas, *grids* y otros elementos basados en HTML y CSS, así como de extensiones JavaScript adicionales.



Figura 25: Logo Bootstrap

### 3.4.3 Base de datos

La base de datos es la capa más baja de una aplicación web. En ella se almacenan todos los datos relevantes al sistema. En el caso de la aplicación planteada en el presente trabajo fin de grado, se guardarán todos los datos de todos los usuarios registrados, incluyendo los justificantes de pago y los artículos de cada usuario. Además, se almacenarán las conferencias y la configuración de la aplicación.

### 3.4.3.1 MySQL

Es un sistema de gestión de bases de datos relacional y está considerada como la base de datos de código abierto más popular del mundo. Es soportada por los lenguajes de programación más importantes y la comunidad de desarrolladores es muy amplia. Wikipedia, Google, Facebook, Twitter, Flickr o YouTube son algunos ejemplos de sitios web que la utilizan.



Figura 26: Logo MySQL

### 3.4.3.2 PostgreSQL

Es un sistema de gestión de bases de datos relacional, orientado a objetos y de código libre. Cuenta con alta concurrencia, amplia variedad de tipos nativos, amplia seguridad, etc. Al igual que con MySQL, PostgreSQL da soporte a la mayoría de los lenguajes de programación.



Figura 27: Logo PostgreSQL

### 3.4.3.3 MongoDB

Es un sistema de base de datos NoSQL orientado a documentos y de código abierto. Lanzado en 2009, los datos son guardados en estructuras de datos de documentos similares a JSON en lugar de tablas como en los sistemas relacionales tradicionales. MongoDB utiliza una especificación llamada BSON, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. Al igual que las relacionales comentadas anteriormente, MongoDB tiene *drivers* oficiales para la mayoría de los lenguajes de programación más extendidos a día de hoy.



Figura 28: Logo MongoDB

#### 3.4.3.4 Firebase Cloud Firestore

Es un servicio derivado de Google Cloud Platform, adaptado a la plataforma Firebase adquirida por Google en 2014. Se trata de una base de datos NoSQL que organiza sus datos en forma de documentos agrupados en colecciones. En ellos se pueden incluir campos de diversos tipos (cadenas de texto, números, puntos geográficos, referencias a la propia base de datos, booleanos, etc.) y otras subcolecciones. Junto con el resto de las herramientas de Firebase y de Google Cloud Platform, las posibilidades de esta base de datos en una aplicación web son inmensas.



Figura 29: Logo Firebase

### 3.5 Descripción de la alternativa seleccionada

Después de analizar todas las alternativas para cada una de las capas de la aplicación, se escogerán aquellas tecnologías que más se ajustan al desarrollo de este proyecto. Cabe destacar que el presente trabajo fin de grado se titula “Mejora del Rendimiento y Ampliación de una Web para la gestión integral de Congresos Científicos”, por lo que una de las premisas es la de modernizar toda la tecnología ya existente. Además, la aplicación tiene un alcance exclusivo para la web, por lo que deberán ser seleccionadas aquellas tecnologías que mejor se adapten a una aplicación web.

Las tecnologías web avanzan con una rapidez mucho mayor que cualquier otra tecnología, por eso es recomendable apostar por tecnologías modernas que tienen dentro de su *roadmap* una mejora continua y un soporte técnico de un gran número de profesionales.

A continuación, se detallarán las razones de las tecnologías seleccionadas para cada una de las capas: *backend*, *frontend* y base de datos.

### 3.5.1 Tecnologías *frontend*

Tras analizar todas las posibilidades tratadas en el apartado anterior se ha decidido usar el moderno *framework* Angular, en su versión 5. Las razones se detallan a continuación:

- Este *framework* sigue el patrón MVC proporcionando a la aplicación web una estructura organizada. El comienzo del proyecto es sencillo ya que Angular crea un esqueleto mínimo que puede ser ejecutado con un simple comando ‘*ng serve -o*’.
- Angular puede usar diferentes gestores de paquetes. Destacan npm y yarn, ambos con miles de paquetes disponibles para cualquier aplicación desarrollada con Angular. Se basa en software libre por lo que cualquier persona puede subir su propia librería y hacerla pública para el resto de los usuarios. El uso de estos gestores simplifica el desarrollo de una aplicación ya que muchas de las librerías disponibles fueron desarrolladas para añadirlas a aplicaciones web de la misma índole.
- Angular se basa principalmente en la creación y reutilización de componentes. Cada componente está compuesto por cuatro ficheros: la vista, la lógica, los estilos y los test. La vista es un fichero HTML. La lógica es un fichero TypeScript. Los estilos es un fichero CSS o SCSS. Los test es un fichero TypeScript con una sintaxis de validación. Todos estos ficheros forman parte del componente y éste puede ser usado en la aplicación añadiendo el selector que se encuentra en la lógica a cualquier otro fichero de la vista de cualquier otro componente. Esto quiere decir que los componentes pueden usarse unos dentro de otros, lo que simplifica la programación y la hace más limpia, evitando duplicidad de código.
- Otra de las razones es que Angular es mantenido por Google por lo que tiene un soporte continuo de los mejores profesionales del sector, siendo actualizado día a día con nuevas características y versiones del *framework*. Aún así, es de código abierto y por ello la comunidad actual de Angular

es muy grande y tiende a ser cada vez mayor, los programadores web son atraídos por la agilidad de mejora, el rendimiento y la sencillez para crear aplicaciones web potentes y modernas.

- Angular a partir de su versión cinco incluye su propio repositorio de componentes ya definidos que son totalmente personalizables: Angular Material. Estos componentes están optimizados para que ofrezcan el mejor rendimiento a las aplicaciones web.
- Angular Universal, propiedad de Angular permite ejecutar la primera vista de la aplicación en el lado del servidor para una adaptación casi instantáneo y para la mejora del SEO en la aplicación.
- Angular CLI, herramienta de línea de comandos que permite realizar tareas tediosas con simples comandos, ayudan significativamente el desarrollo de cualquier aplicación.
- Angular posee decenas de *plugins* para los IDEs de desarrollo web más comunes. Estos *plugins* facilitan enormemente el desarrollo. El IDE, Visual Studio Code, tiene numerosas ventajas como la sugerencia de código inteligente y detección de errores además de múltiples *plugins*.
- Angular permite fácilmente la integración con otras tecnologías o *frameworks* web como jQuery, Bootstrap, HTML5, CSS3, JavaScript o Sass que también serán utilizadas en el lado del cliente de la aplicación.
- Angular e Ionic, *framework* para la creación de aplicaciones móviles, son como uña y carne para crear aplicaciones web a versión móvil. Con tan solo unos comandos de instalación y de ejecución, cualquier aplicación escrita en Angular puede pasar a ser una aplicación móvil de Ionic.

### 3.5.2 Tecnologías *backend*

En el lado del servidor todas las tecnologías comentadas podrían ser usadas ya que todas ellas cumplirían con las expectativas previstas. Sin embargo, se ha escogido Node.js ya que es la que mejor se adapta con Angular. Las razones por las que se ha elegido son:

- Tanto Node.js como Angular comparten indirectamente el mismo lenguaje de programación. Aunque en Angular se use TypeScript, el código TypeScript es similar al código JavaScript de Node.js y cualquier fragmento de código JavaScript funcionaría también en TypeScript. Esto es útil si se desea replicar código del lado del servidor en el lado del cliente.
- Tanto Node.js como Angular poseen la misma estructura de intercambio de datos: JSON. Cualquier comunicación entre el cliente y el servidor o viceversa es directa a través de ficheros JSON.
- Al igual que Angular, Node.js puede usar npm o yarn como gestor de paquetes.
- Node.js posee una gran comunidad *open source* a sus espaldas que evoluciona ágilmente. A pesar de su corta vida, el futuro de Node.js es muy favorable gracias a ello.
- El navegador más usado a día de hoy, Google Chrome, usa el mismo entorno de ejecución escrito en JavaScript que Node.js. Se trata de V8. Esto hace que las aplicaciones de Node.js estén optimizadas cuando se usa el navegador más usado en el mundo.
- Node.js funciona en un único hilo de ejecución, con entradas y salidas asíncronas que pueden ejecutarse concurrentemente hasta en cientos de miles sin acarrear costos asociados al cambio de contexto. Por ello escala increíblemente bien cuando una aplicación supera inesperadamente el número de visitas o el número de peticiones.

### 3.5.3 Base de datos

Una vez escogidas las tecnologías *backend* y *frontend* de la aplicación, hay que escoger el sistema de almacenamiento que mejor se adapte a las tecnologías elegidas. En este caso, y ya que se ha apostado por usar las tecnologías más modernas del mercado, una de las tendencias actuales para las bases de datos es Firebase, plataforma integrada en Google Cloud Platform que cuenta con muchas herramientas útiles para el desarrollo web, entre las que se incluyen las bases de datos. Arriesgando un poco, se ha elegido

Cloud Firestore como base de datos NoSQL o no relacional en su versión BETA. Las razones son las siguientes:

- Cloud Firestore es propiedad de Google. Al formar parte de Google, el soporte y la documentación es muy extensa y cuenta con numerosos tutoriales.
- Como parte de Firebase, la integración con otras herramientas de Firebase como DebugView, Audiences, Functions, AdMob, Predictions o muchas otras, enriquece enormemente las posibilidades de la aplicación.
- Storage, también de Firebase, al igual que Cloud Firestore almacena en la nube todos los archivos de la aplicación por carpetas.
- En cualquier momento, en cualquier lugar y con cualquier dispositivo con una conexión a internet es posible acceder, modificar o eliminar cualquier dato o archivo de la aplicación.
- Cloud Firestore almacena los datos en colecciones de documentos, al estilo JSON. Las colecciones pueden ser almacenadas en otras colecciones, por lo que la escalabilidad es muy buena.
- Firebase ofrece unos parámetros de configuración para aplicaciones web, aplicaciones Android y aplicaciones iOS a la misma base de datos. Si la aplicación se desarrolla en un futuro para móvil, esto agiliza enormemente el tiempo de integración con la base de datos, ya que la implementación es inmediata con estos parámetros de configuración.
- Cloud Firestore usa la sincronización de datos para actualizar los datos de cualquier dispositivo conectado, usando consultas de recuperación únicas de manera eficiente.
- Cloud Firestore cuenta con asistencia sin conexión. Los datos se almacenan en la caché de la aplicación de forma activa, aunque no haya conexión. Cuando se recupera la conexión, Cloud Firestore sincroniza todos los cambios cacheados.

### **3.5.4 Sistema operativo**

Las tecnologías escogidas de todas las capas son multiplataforma, por lo que cualquier sistema operativo puede ejecutar el código de la aplicación. El usuario del sistema podrá escoger el sistema operativo que más le guste para la ejecución de la aplicación o el desarrollo en local.

## **DOCUMENTO 4: ANÁLISIS DE REQUISITOS DEL SISTEMA**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 4.1 Introducción

En este documento se hará una identificación de los subsistemas de análisis que descompondrán el sistema con el objetivo de facilitar el análisis global de éste. Una vez identificados, se procederá a la obtención de requisitos de sistema mediante el desarrollo de casos de uso.

Los casos de uso describen las funciones que la aplicación debe ejecutar cuando el usuario interactúa con el sistema. Normalmente se desglosa en una secuencia de interacciones entre el sistema y el usuario.

## 4.2 Subsistemas de análisis y casos de uso

Los subsistemas de la aplicación web que conformarán nuestro sistema principal son los siguientes:

1. Acceso y registro de usuarios.
2. Gestión de usuarios asistentes al congreso.
3. Gestión de usuario administrador.

A continuación, se describirán los casos de uso de cada subsistema. Para cada subsistema se incluirá un diagrama para dar una visión global del subsistema.

### 4.2.1 Acceso y registro de usuarios

En este apartado se desarrollarán los casos de uso para el subsistema “Acceso y registro de usuarios”, que engloba aquellas funcionalidades relacionadas con el acceso a la aplicación web (registro y *login*). En la siguiente figura se detallan los principales casos de uso de este subsistema:

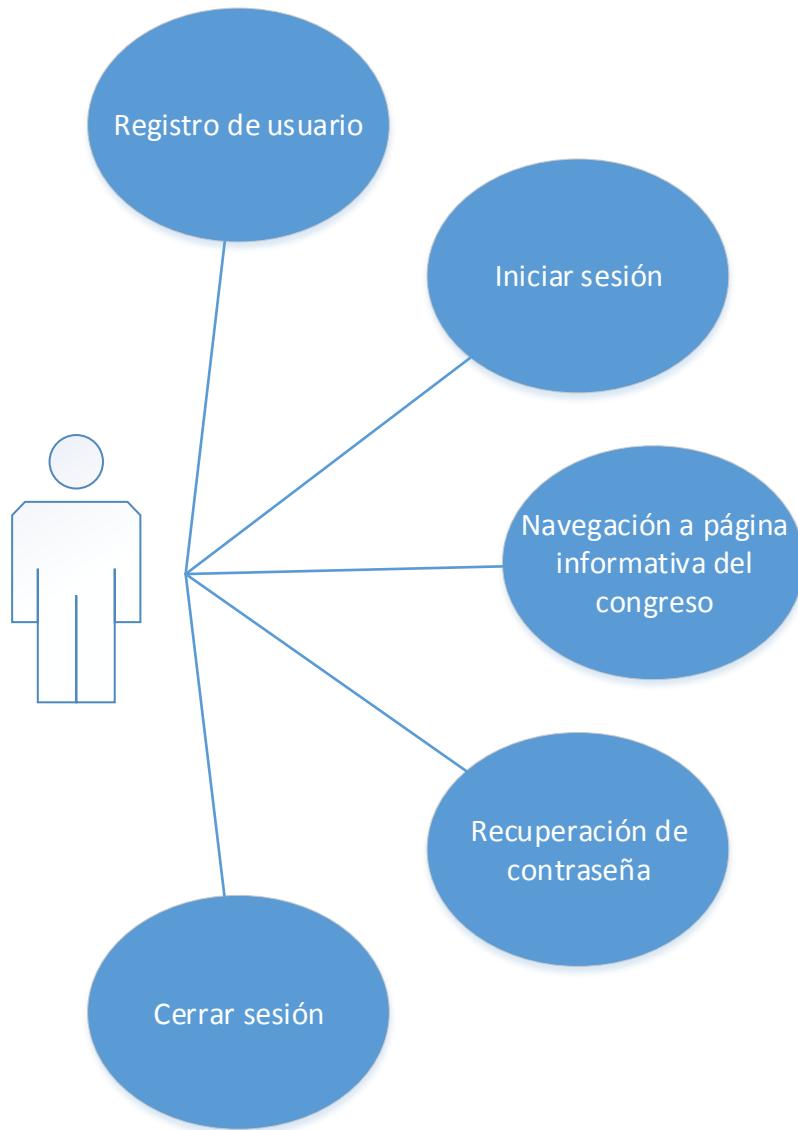


Figura 30: Subsistema Acceso y registro de usuarios

#### 4.2.1.1 Registro de usuario

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario no puede estar registrado en la aplicación.
- Descripción: el usuario se podrá registrar en la aplicación creando su propio perfil de usuario.

- Flujo normal del caso de uso:

1. El usuario accede a la pantalla principal de la aplicación web.
2. El usuario accede a la página de registro.
3. El usuario rellena el formulario de registro.
4. El usuario envía el formulario llenado.
5. El sistema valida la información y registra al usuario en la aplicación.
6. El sistema notifica al usuario vía mail de que se ha registrado correctamente. El sistema también notificará vía mail al usuario administrador.

- Excepciones:

4. El usuario navega a otro apartado o página de la aplicación antes de clicar en el botón de envío del formulario. El registro no se producirá y el usuario no se registrará en el sistema.
5. El sistema valida la aplicación y detecta que el correo electrónico pertenece a otro usuario ya registrado en la aplicación. El sistema notifica al usuario que el usuario ya existe.

#### **4.2.1.2 Login**

- Actores: usuarios asistentes al congreso o usuario administrador.
- Precondiciones: el usuario debe estar registrado en la plataforma.
- Descripción: el usuario iniciará sesión en la aplicación web.
- Flujo normal del caso de uso:

1. El usuario accede a la pantalla principal de la aplicación web.
2. El usuario accede a la página de *login*.

3. El usuario rellena el nombre de usuario (email) y contraseña.
  4. El usuario clica en el botón de inicio de sesión o pulsa la tecla *enter*.
  5. El sistema valida los datos e inicia la sesión del usuario.
- Excepciones:
4. El usuario navega a otro apartado o página de la aplicación antes de clicar en el botón de inicio de sesión o pulsar la tecla *enter*. Los datos introducidos no se validarán y el usuario no iniciará sesión.
  5. El sistema valida los datos de inicio de sesión y éstos son incorrectos, el sistema notificará al usuario que los datos no son correctos.

#### 4.2.1.3 Recuperación de contraseña

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma.
- Descripción: el usuario establecerá una nueva contraseña.
- Flujo normal del caso de uso:
1. El usuario accede a la pantalla principal de la aplicación web.
  2. El usuario accede a la página de *login*.
  3. El usuario accede a la pantalla de recuperación de contraseña.
  4. El usuario introduce su dirección de correo electrónico.
  5. El usuario clica en el botón de recuperar.
  6. El sistema valida la dirección y envía un email al correo electrónico previamente introducido por el usuario con una nueva contraseña generada aleatoriamente por el sistema. El sistema también actualizará la base de datos con esta nueva contraseña.

- Excepciones:
  6. El sistema valida el correo electrónico introducido por el usuario y éste no es válido. El sistema indica al usuario que el usuario no existe.

#### **4.2.1.4 Navegación a página informativa del congreso**

- Actores: usuarios asistentes al congreso o usuario administrador.
- Precondiciones: ninguna.
- Descripción: el usuario navegará a la página informativa del congreso, propiedad de la Universidad de Salamanca.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla principal de la aplicación web.
  2. El usuario hace clic en el botón “Volver a Conferencias”.
- Flujo alternativo del caso de uso:
  2. El usuario hace clic en el botón “Sobre CMMSE” de la cabecera del menú.
- Flujo alternativo del caso de uso:
  2. El usuario navega a la pantalla de *login*, registro o recuperación de contraseña.
  3. El usuario hace clic en el botón “Sobre CMMSE” de la cabecera del menú.

#### **4.2.1.5 Cerrar sesión**

- Actores: usuarios asistentes al congreso o usuario administrador.
- Precondiciones: el usuario debe haber iniciado sesión.
- Descripción: el usuario cerrará su sesión y volverá a la página de *login*.

- Flujo normal del caso de uso:

1. El usuario hace clic en el botón “Cerrar Sesión” de la cabecera del menú.
2. El sistema navegará a la pantalla de *login* y eliminará la información almacenada del usuario en el navegador.

#### **4.2.2 Gestión de usuarios asistentes al congreso**

En este apartado se analizarán los casos de uso del subsistema “Gestión de usuarios asistentes al congreso”, el cual engloba todas las funcionalidades relativas a los usuarios registrados en el sistema. En la siguiente figura se pueden ver los casos de uso principales de este subsistema:

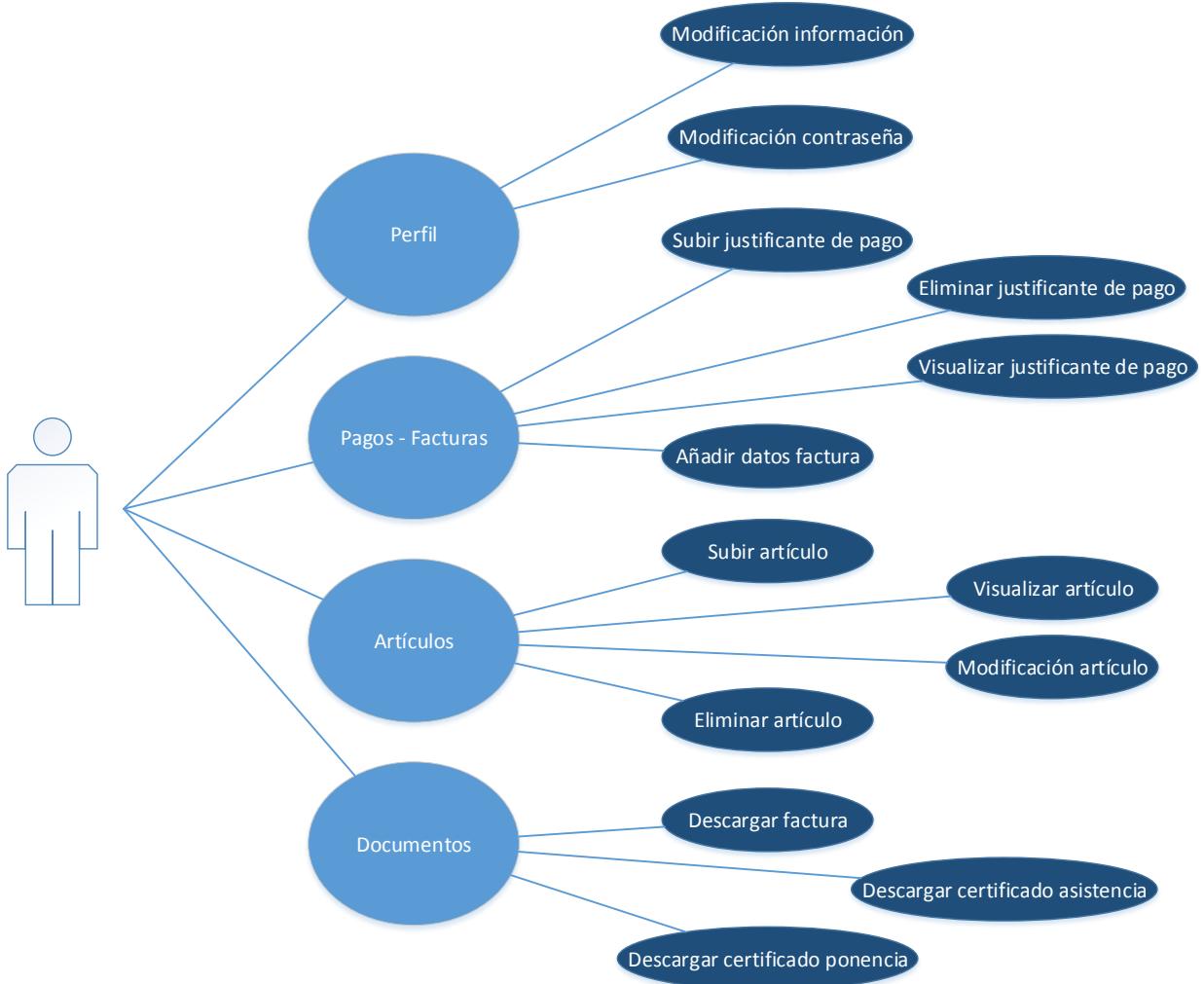


Figura 31: Subsistema Gestión de usuarios asistentes al congreso

#### 4.2.2.1 Modificación de la información de usuario

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en el sistema y haber iniciado sesión.
- Descripción: el usuario podrá cambiar toda la información de su cuenta a excepción del email.
- Flujo normal del caso de uso:

1. El usuario accede a la pantalla perfil.
  2. El usuario clica en el botón “Cambiar datos”.
  3. El usuario modifica aquellos campos de su cuenta que desee cambiar.
  4. El usuario guarda los cambios clicando el botón “Guardar cambios”.
  5. El sistema valida la información y actualiza los datos.
  6. El sistema notifica al usuario del éxito de la operación.
- Excepciones:
4. El usuario accede a otro apartado o página de la aplicación antes de clicar el botón “Guardar cambios”. Los nuevos datos introducidos no se guardarán.

#### **4.2.2.2 Modificación de la contraseña de usuario**

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.
- Descripción: el usuario podrá modificar su contraseña.
- Flujo normal del caso de uso:
1. El usuario accede a la pantalla perfil.
  2. El usuario clica en el botón “Cambiar contraseña”.
  3. El sistema abre una ventana modal con tres campos: contraseña actual, contraseña nueva y confirmar contraseña nueva.
  4. El usuario introduce los datos en la ventana modal.
  5. El usuario clica en el botón “Guardar cambios”.

6. El sistema guarda la nueva contraseña definida por el usuario y notifica a éste del éxito de la operación.

- Excepciones:

5. El usuario accede a otro apartado o página de la aplicación antes de clicar el botón “Guardar cambios”. La contraseña no se cambiará y se perderán los datos introducidos.

#### **4.2.2.3 Subir justificante de pago**

- Actores: usuarios asistentes al congreso.

- Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.

- Descripción: el usuario podrá subir un justificante de pago.

- Flujo normal del caso de uso:

1. El usuario accede a la pantalla de pagos y facturas.

2. El usuario clica en el botón “Seleccionar archivo” y escoge un archivo almacenado en su dispositivo.

3. El sistema almacena el archivo seleccionado y notifica al usuario el éxito de la operación.

4. El sistema envía un correo electrónico al usuario administrador indicando que el usuario ha subido un justificante de pago.

#### **4.2.2.4 Eliminar justificante de pago**

- Actores: usuarios asistentes al congreso.

- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión y haber subido previamente un justificante de pago.

- Descripción: el usuario podrá eliminar el justificante de pago subido previamente.

- Flujo normal del caso de uso:

1. El usuario accede a la pantalla de pagos y facturas.
2. El usuario clica en el botón “x” de la sección “Pagos”.
3. El sistema elimina el justificante de pago y notifica al usuario el éxito de la operación.

#### **4.2.2.5 Visualización de justificante de pago**

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión y haber subido previamente un justificante de pago.
- Descripción: el usuario podrá visualizar el justificante de pago subido previamente.
- Flujo normal del caso de uso:

1. El usuario accede a la pantalla de pagos y facturas.
2. El usuario clica en el enlace del nombre del archivo subido.
3. El sistema abre una pestaña nueva en el navegador con el justificante de pago subido.

#### **4.2.2.6 Añadir datos de factura**

- Actores: usuarios asistentes al congreso.
  - Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.
  - Descripción: el usuario podrá añadir datos para la generación de su factura.
  - Flujo normal del caso de uso:
1. El usuario accede a la pantalla de pagos y facturas.
  2. El usuario rellena los datos del formulario de la sección “Facturas”.
  3. El usuario clica en el botón “Enviar”.

4. El sistema guarda los datos introducidos y notifica al usuario el éxito de la operación.

- Excepciones:

3. El usuario accede a otro apartado o página de la aplicación antes de clicar el botón “Enviar”. Los datos no se almacenarán y se perderán los datos introducidos.

#### 4.2.2.7 Subir artículo

- Actores: usuarios asistentes al congreso.

- Precondiciones: el usuario debe estar registrado en la plataforma y haber iniciado sesión.

- Descripción: el usuario podrá subir un artículo a la plataforma.

- Flujo normal del caso de uso:

1. El usuario accede a la pantalla de artículos.

2. El usuario rellena los datos del formulario de la sección “Nuevo artículo”. Incluyendo el archivo a subir y los autores del artículo.

3. El usuario clica en el botón “Enviar”.

4. El sistema guarda los datos introducidos, el archivo subido y notifica al usuario el éxito de la operación.

5. El sistema muestra en una tabla dentro de la sección “Mis artículos” el artículo subido.

6. El sistema envía un correo electrónico al usuario administrador indicando que el usuario ha subido un nuevo artículo.

- Excepciones:

3. El usuario accede a otro apartado o página de la aplicación antes de clicar el botón “Enviar”. Los datos no se almacenarán y se perderán los datos introducidos.

-

#### 4.2.2.8 Visualizar artículo

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión y haber subido un artículo previamente.
- Descripción: el usuario podrá visualizar un artículo subido previamente.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla de artículos.
  2. En la sección “Mis artículos” se encuentra una tabla con el artículo a visualizar. El usuario clica en el ícono de la columna “Archivo” de la tabla.
  3. El sistema abre en una pestaña nueva del navegador el artículo seleccionado.

#### 4.2.2.9 Modificación de artículo

Correo al sistema si el estado es Major /Minor

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión y haber subido un artículo previamente. Además, el estado del artículo debe ser “subido” o “mayor/menor”.
- Descripción: el usuario podrá modificar un artículo subido previamente.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla de artículos.
  2. En la sección “Mis artículos” se encuentra una tabla con el artículo a modificar. El usuario clica en el ícono izquierdo (editar) de la columna “Acciones” de la tabla. El artículo está en estado “subido”.
  3. El sistema abre una ventana modal con los datos del artículo. Todos los datos son modificables.

4. El usuario modifica los datos que desea del artículo.
5. El usuario clica en el botón “Guardar cambios”.
6. El sistema actualiza los datos modificados en la base de datos, cierra la ventana modal y notifica al usuario el éxito de la operación.

- Flujo alternativo:

2. En la sección “Mis artículos” se encuentra una tabla con el artículo a modificar. El usuario clica en el ícono izquierdo (editar) de la columna “Acciones” de la tabla. El artículo está en estado “mayor/menor”.
3. El sistema abre una ventana modal con todos los datos del artículo modificables a excepción del “Autor principal”.
4. El usuario modifica los datos que desea del artículo.
5. El usuario clica en el botón “Guardar cambios”.
6. El sistema actualiza los datos modificados en la base de datos, cierra la ventana modal y notifica al usuario el éxito de la operación.
7. El sistema envía un correo electrónico al usuario administrador indicando que el usuario ha modificado los datos del artículo en estado “mayor/menor”.

- Excepciones:

5. El usuario cierra la ventana modal antes de clicar el botón “Guardar cambios”. Los datos no se almacenarán y se perderán los datos introducidos.

#### **4.2.2.10 Eliminar artículo**

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión, haber subido un artículo previamente y que el estado de éste sea “subido”.

- Descripción: el usuario podrá subir un artículo a la plataforma.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla de artículos.
  2. En la sección “Mis artículos” se encuentra una tabla con el artículo a eliminar. El usuario clica en el ícono derecho (borrar) de la columna “Acciones” de la tabla.
  3. El sistema abre una ventana modal con dos botones de acción.
  4. El usuario clica en el botón “Sí”.
  5. El sistema cierra el modal, elimina el artículo de la base de datos y actualiza la tabla de la sección “Mis artículos”. Además, notifica al usuario del éxito de la operación.
- Excepciones:
  4. El usuario clica en el botón “Cancelar” o cierra la ventana modal. El artículo no se eliminará.

#### **4.2.2.11 Descarga de factura**

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión, haber subido un justificante de pago y que el usuario administrador haya validado el pago.
- Descripción: el usuario podrá descargar la factura que acredita el pago realizado previamente. Para evitar fraudes fiscales, la descarga solo podrá realizarse una única vez.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla de documentos.
  2. El usuario clica en el botón “Descargar” de la sección “Factura”.
  3. El sistema inicia la descarga del archivo en formato PDF.

#### 4.2.2.12 Descarga de certificado de asistencia

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión, haber subido un justificante de pago y que el usuario administrador haya validado el pago.
- Descripción: el usuario podrá descargar el certificado de asistencia al congreso. Para evitar fraudes fiscales, la descarga solo podrá realizarse una única vez.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla de documentos.
  2. El usuario clica en el botón “Descargar” de la sección “Certificado de asistencia”.
  3. El sistema inicia la descarga del archivo en formato PDF.

#### 4.2.2.13 Descarga de certificado de ponencia

- Actores: usuarios asistentes al congreso.
- Precondiciones: el usuario debe estar registrado en la plataforma, haber iniciado sesión, haber subido un justificante de pago, que el usuario administrador haya validado el pago, haber subido al menos un artículo que se encuentre en estado “aceptado”.
- Descripción: el usuario podrá descargar el certificado de ponencia al congreso. Para evitar fraudes fiscales, la descarga solo podrá realizarse una única vez.
- Flujo normal del caso de uso:
  1. El usuario accede a la pantalla de documentos.
  2. El usuario clica en el botón “Descargar” de la sección “Certificado de ponencia”.
  3. El sistema inicia la descarga del archivo en formato PDF. Se añadirá una hoja al archivo por cada artículo en estado “aceptado”.

### 4.2.3 Gestión de usuario administrador

En este apartado se detallarán los casos de uso para el subsistema “Gestión de usuario administrador”, que engloba todas las funcionalidades relacionadas con la gestión del congreso y que solo es accesible por el usuario administrador. En la siguiente figura se pueden ver los diferentes casos de uso de este apartado:

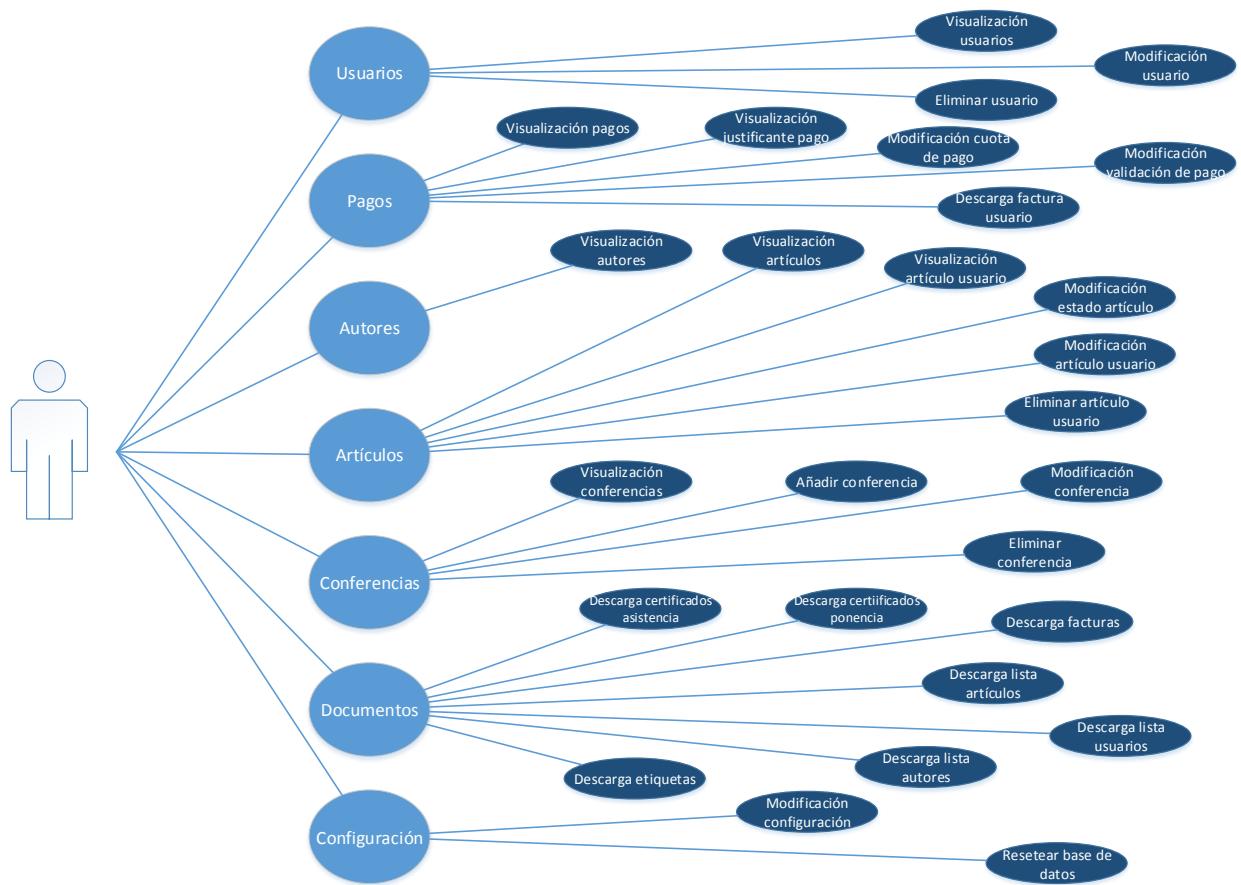


Figura 32: Gestión de usuario administrador

#### 4.2.3.1 Visualización tabla de usuarios

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.

- Descripción: el usuario administrador podrá visualizar en forma de tabla todos los usuarios registrados.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de usuarios.
  2. El sistema muestra una tabla resumen con el listado de usuarios.

#### 4.2.3.2 Modificación usuario

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá modificar los datos de cualquier usuario registrado.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de usuarios.
  2. El usuario administrador clica en el ícono izquierdo de la columna “acciones” situado en la fila del usuario que se va a modificar.
  3. El sistema abre una ventana modal con los datos del usuario seleccionado deshabilitados.
  4. El usuario administrador clica en el botón “Editar datos”.
  5. El sistema habilita los campos con los datos del usuario.
  6. El usuario administrador modifica los datos del usuario y clica en el botón “Guardar cambios”.
  7. El sistema actualiza los datos modificados en la base de datos, cierra el modal y notifica al usuario administrador el éxito de la operación.

- Excepciones:
  6. El usuario administrador cierra la ventana modal. Los datos no se cambiarán y se perderán los cambios realizados.

#### **4.2.3.3 Eliminar usuario**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá eliminar a cualquier usuario registrado de la aplicación.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de usuarios.
  2. El usuario administrador clica en el ícono derecho de la columna “acciones” situado en la fila del usuario que se va a eliminar.
  3. El sistema abre una ventana modal con dos acciones.
  4. El usuario administrador clica en el botón “Sí”.
  5. El sistema elimina al usuario de la base de datos, actualiza la tabla y notifica al usuario administrador el éxito de la operación.
- Excepciones:
  4. El usuario administrador cierra la ventana modal. El usuario no se elimina.

#### **4.2.3.4 Visualización tabla de pagos**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá visualizar el estado de pago de cualquier usuario registrado.

- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de pagos.
2. El sistema muestra una tabla resumen de pagos con el listado de usuarios.

#### **4.2.3.5 Visualización justificante de pago de usuario**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión y el usuario seleccionado debe tener justificante de pago subido.
- Descripción: el usuario administrador podrá visualizar el justificante de pago de cualquier usuario registrado de la aplicación que haya subido el archivo.
- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de pagos.
2. El usuario administrador clica en el ícono de la columna “archivo de pago” situado en la fila del usuario elegido.
3. El sistema abre una pestaña nueva con el archivo.

#### **4.2.3.6 Modificación cuota de pago de usuario**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá modificar la cuota de pago de cualquier usuario registrado de la aplicación.
- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de pagos.

2. El usuario administrador modifica el campo de la tabla de la columna “a pagar” situado en la fila del usuario elegido.
3. El sistema actualiza la cuota del usuario en la base de datos y notifica al usuario administrador el éxito de la operación.

#### **4.2.3.7 Modificación validación de pago de usuario**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión y el usuario seleccionado debe tener justificante de pago subido.
- Descripción: el usuario administrador podrá validar o no validar el justificante de pago de cualquier usuario registrado de la aplicación.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de pagos.
  2. El usuario administrador modifica el campo de la tabla de la columna “pagado” situado en la fila del usuario elegido.
  3. El sistema abre una ventana modal con las acciones posibles.
  4. El usuario clica en el botón “Sí”.
  5. El sistema actualiza la validación de pago del usuario y notifica al usuario administrador el éxito de la operación.
    5. a) Si se valida por primera vez, el sistema asigna número de factura al usuario y habilita los documentos en el perfil del usuario.
    5. b) Si se valida y no es la primera vez, el sistema habilita los documentos en el perfil de usuario.

5. c) Si se quita la validación, el sistema deshabilita los documentos en el perfil de usuario, pero mantiene el número de factura para el usuario.

- Excepciones:

4. El usuario administrador cierra la ventana modal. El pago no cambia de estado.

#### **4.2.3.8 Descarga de factura de usuario**

- Actores: usuario administrador.

- Precondiciones: el usuario administrador debe haber iniciado sesión.

- Descripción: el usuario administrador podrá visualizar la factura de cualquier usuario registrado de la aplicación.

- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de pagos.

2. El usuario administrador clica en el ícono de la columna “factura” situado en la fila del usuario elegido.

3. El sistema abre una pestaña nueva con el archivo.

#### **4.2.3.9 Visualización tabla de autores**

- Actores: usuario administrador.

- Precondiciones: el usuario administrador debe haber iniciado sesión.

- Descripción: el usuario administrador podrá visualizar la tabla con un resumen de los autores.

- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de autores.

2. El sistema muestra una tabla resumen con el listado de autores.

#### **4.2.3.10 Visualización tabla de artículos**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá visualizar la tabla con un resumen de todos los artículos subidos.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de artículos.
  2. El sistema muestra una tabla resumen con el listado de todos los artículos subidos a la aplicación.

#### **4.2.3.11 Visualización artículo de usuario**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión y al menos un usuario debe haber subido un artículo.
- Descripción: el usuario administrador podrá visualizar el artículo de cualquier usuario registrado de la aplicación que haya subido el archivo.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de artículos.
  2. El usuario administrador clica en el ícono de la columna “artículo” situado en la fila del artículo elegido.
  3. El sistema abre una pestaña nueva en el navegador con el archivo.

#### **4.2.3.12 Modificación estado de artículo de usuario**

- Actores: usuario administrador.

- Precondiciones: el usuario administrador debe haber iniciado sesión y al menos un usuario debe haber subido un artículo.
- Descripción: el usuario administrador podrá modificar el estado de cualquier artículo subido a la aplicación.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de artículos.
  2. El usuario administrador modifica el campo de la columna “estado” situado en la fila del artículo elegido.
  3. El sistema guarda el cambio de estado del artículo en la base de datos y notifica al usuario administrador el éxito en la operación.
  4. El sistema envía un correo electrónico al usuario que subió el artículo y al usuario administrador indicando el cambio de estado del artículo.

#### 4.2.3.13 Modificación artículo de usuario

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión y al menos un usuario debe haber subido un artículo.
- Descripción: el usuario administrador podrá modificar los datos de cualquier artículo subido a la aplicación.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de artículos.
  2. El usuario administrador clica en el ícono izquierdo de la columna “acciones” del artículo a modificar.

3. El sistema abre una ventana modal con los datos del artículo.
  4. El usuario administrador modifica los datos del artículo que deseé cambiar.
  5. El usuario clica en el botón “Guardar cambios”.
  6. El sistema actualiza los datos del artículo en la base de datos, cierra la ventana modal y notifica al usuario el éxito de la operación.
- Excepciones:
    5. El usuario administrador cierra la ventana modal. Los datos modificados no se guardan.

#### 4.2.3.14 Eliminar artículo de usuario

- Actores: usuario administrador.
  - Precondiciones: el usuario administrador debe haber iniciado sesión y al menos un usuario debe haber subido un artículo.
  - Descripción: el usuario administrador podrá eliminar cualquier artículo subido a la aplicación.
  - Flujo normal del caso de uso:
    1. El usuario administrador accede a la pantalla de artículos.
    2. El usuario administrador clica en el ícono derecho de la columna “acciones” del artículo a modificar.
    3. El sistema abre una ventana modal con las acciones posibles.
    4. El usuario administrador clica en el botón “Sí”.
    5. El sistema elimina el artículo de la base de datos, cierra la ventana modal y notifica al usuario el éxito de la operación.
- Excepciones:
    5. El usuario administrador cierra la ventana modal. El artículo no se elimina.

#### 4.2.3.15 Visualización tabla de conferencias

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá visualizar la tabla con todas las conferencias guardadas en el sistema.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de conferencias.
  2. El sistema muestra una tabla con el listado de todas las conferencias disponibles en la aplicación.

#### 4.2.3.16 Añadir nueva conferencia

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá añadir una nueva conferencia a la aplicación.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de conferencias.
  2. En la sección “Nueva conferencia”, el usuario añade un título para la nueva conferencia.
  3. El usuario administrador clica en el botón “Añadir conferencia”.
  4. El sistema añade a la base de datos la nueva conferencia, actualiza la tabla en la pantalla conferencias y notifica al usuario administrador el éxito en la operación.
- Excepciones:
  3. El usuario administrador cambia de pantalla o sale de la aplicación. La conferencia no se añade.

#### 4.2.3.17 Modificación conferencia

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo una conferencia en la aplicación.
- Descripción: el usuario administrador podrá modificar una conferencia ya existente en la aplicación.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de conferencias.
  2. El usuario clica en el ícono izquierdo (editar) de la columna “acciones” de la conferencia a la que va a modificar.
  3. El sistema abre una ventana modal con el título de la conferencia.
  4. El usuario modifica el título.
  5. El usuario clica en el botón “Guardar cambios”.
  4. El sistema modifica en la base de datos el título de la conferencia, actualiza la tabla en la pantalla conferencias y notifica al usuario administrador el éxito en la operación.
- Excepciones:
  5. El usuario administrador cierra la ventana modal. La conferencia no se modifica.

#### 4.2.3.18 Eliminar conferencia

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo una conferencia en la aplicación.
- Descripción: el usuario administrador podrá eliminar una conferencia ya existente en la aplicación.

- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de conferencias.
2. El usuario clica en el ícono derecho (borrar) de la columna “acciones” de la conferencia a la que va a eliminar.
3. El sistema abre una ventana modal con las opciones posibles.
4. El usuario clica en el botón “Sí”.
5. El sistema elimina de la base de datos la conferencia, actualiza la tabla en la pantalla conferencias y notifica al usuario administrador el éxito en la operación.

- Excepciones:

4. El usuario administrador cierra la ventana modal. La conferencia no se elimina.

#### **4.2.3.19 Descarga de certificados de asistencia**

- Actores: usuario administrador.

- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un usuario registrado en la aplicación.

- Descripción: el usuario administrador podrá descargar en formato PDF todos los certificados de asistencia de los usuarios.

- Flujo normal del caso de uso:

1. El usuario administrador accede a la pantalla de documentos.
2. El usuario clica en el botón “Certificados de asistencia” de la sección “Documentos PDF”.
3. El sistema descarga un archivo PDF con todos los certificados de asistencia.

#### 4.2.3.20 Descarga de certificados de ponencia

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un artículo subido a la aplicación con estado “aceptado”.
- Descripción: el usuario administrador podrá descargar en formato PDF todos los certificados de ponencia de todos los artículos con estado “aceptado”.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de documentos.
  2. El usuario clica en el botón “Certificados de ponencia” de la sección “Documentos PDF”.
  3. El sistema descarga un archivo PDF con todos los certificados de ponencia de todos los artículos en estado “aceptado”.

#### 4.2.3.21 Descarga de facturas

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un usuario registrado.
- Descripción: el usuario administrador podrá descargar todas las facturas de los usuarios registrados en formato PDF.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de documentos.
  2. El usuario clica en el botón “Facturas” de la sección “Documentos PDF”.
  3. El sistema descarga un archivo PDF con todas las facturas de los usuarios registrados.

#### 4.2.3.22 Descarga de lista de usuarios

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un usuario registrado.
- Descripción: el usuario administrador podrá descargar la lista de usuarios en formato Excel.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de documentos.
  2. El usuario clica en el botón “Usuarios” de la sección “Documentos Excel”.
  3. El sistema descarga un archivo Excel con la lista de los usuarios registrados.

#### 4.2.3.23 Descarga de lista de artículos

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un artículo subido.
- Descripción: el usuario administrador podrá descargar la lista de artículos en formato Excel.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de documentos.
  2. El usuario clica en el botón “Artículos” de la sección “Documentos Excel”.
  3. El sistema descarga un archivo Excel con la lista de los artículos subidos.

#### 4.2.3.24 Descarga de lista de autores

- Actores: usuario administrador.

- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un artículo subido.
- Descripción: el usuario administrador podrá descargar la lista de autores en formato Excel.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de documentos.
  2. El usuario clica en el botón “Autores” de la sección “Documentos Excel”.
  3. El sistema descarga un archivo Excel con la lista de los autores de la aplicación.

#### **4.2.3.25 Descarga de etiquetas**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión. Debe haber como mínimo un usuario registrado.
- Descripción: el usuario administrador podrá descargar la lista de etiquetas para todos los usuarios en formato PDF.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de documentos.
  2. El usuario rellena los campos de la sección “Etiquetas” a excepción del último campo (no obligatorio).
  3. El usuario clica en el botón “Etiquetas”.
  3. El sistema descarga un archivo PDF con las etiquetas identificativas de todos los usuarios.
- Flujo alternativo:
  2. El usuario rellena todos los campos de la sección “Etiquetas”.

3. El usuario clica en el botón “Etiquetas”.
4. El sistema descarga un archivo PDF con etiquetas que incluyen el texto introducido en el último campo de la sección “Etiquetas”.

#### **4.2.3.26 Modificación de la configuración**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.
- Descripción: el usuario administrador podrá modificar la configuración de la conferencia.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de configuración.
  2. El usuario clica en el botón “Cambiar datos”.
  3. El sistema habilita los campos para que puedan ser modificados.
  4. El usuario modifica los campos correspondientes.
  5. El usuario administrador clica en el botón “Guardar cambios”.
  6. El sistema actualiza la base de datos con la configuración modificada, notifica al usuario administrador el éxito de la operación y vuelve a deshabilitar los campos de la pantalla.
- Excepciones:
  5. El usuario administrador cambia de pantalla o sale de la aplicación. Los datos de la configuración no son modificados.

#### **4.2.3.27 Resetear base de datos**

- Actores: usuario administrador.
- Precondiciones: el usuario administrador debe haber iniciado sesión.

- Descripción: el usuario administrador podrá eliminar los datos de la base de datos.
- Flujo normal del caso de uso:
  1. El usuario administrador accede a la pantalla de configuración.
  2. El usuario clica en el *toggle* “Habilitar botones”.
  3. El sistema habilita los botones de reseteo de la base de datos.
  4. El usuario clica en los botones “Resetear usuarios” y “Resetear conferencias”.
  5. El sistema elimina de la base de datos todos los usuarios, todas las conferencias y todos los artículos. Además, notifica al usuario administrador el éxito de la operación.
- Flujo alternativo:
  4. El usuario administrador clica en el botón “Resetear usuarios”.
  5. El sistema elimina de la base de datos todos los usuarios y todos los artículos. Además, notifica al usuario administrador el éxito de la operación.
- Flujo alternativo:
  4. El usuario administrador clica en el botón “Resetear conferencias”.
  5. El sistema elimina de la base de datos todas las conferencias. Además, notifica al usuario administrador el éxito de la operación.
- Excepciones:
  4. El usuario administrador cambia de pantalla o sale de la aplicación. La base de datos no será reseteada.

## 4.3 Requisitos no funcionales

Los requisitos no funcionales se centran en aspectos como la eficiencia, seguridad, rendimiento, usabilidad o accesibilidad. La aplicación debe garantizar estos requisitos para obtener la mayor calidad posible en el uso del sistema. A continuación, se detallan estos requisitos:

1. La aplicación podrá ser usada en cualquier navegador web o móvil.
2. La aplicación tendrá que ser responsive, es decir, deberá adaptarse visualmente a cualquier dispositivo, ya sea web o móvil.
3. El sistema deberá ser internacionalizado. Los usuarios de habla inglesa y castellana podrán disfrutar de una versión en su idioma nativo.
4. La aplicación debe mantener una interfaz similar en todas las pantallas, con los mismos colores, formas y tipos de letras.
5. El sistema tendrá que almacenar las contraseñas en la base de datos con un método de encriptación simétrica.
6. La aplicación tendrá que sacar de sesión a los usuarios a los 30 minutos de inactividad.
7. Los tiempos de espera para la carga y ejecución de la aplicación deben ser inferiores a 10 segundos.
8. La aplicación web contará con un certificado SSL que garantice que es un sitio seguro.
9. La aplicación no podrá tener enlaces rotos. El usuario podrá navegar por la aplicación sin encontrar pantallas que no le permitan seguir navegando.
10. Cualquier persona, sin importar su edad, sexo, condición social o cultura podrá usar la aplicación.
11. La aplicación tendrá que ser amigable con el usuario, fácil de usar y fácil de aprender.
12. La aplicación deberá indicar al usuario cualquier fallo o éxito en las operaciones que realice.
13. El sistema debe ser autónomo, los propios usuarios deben tener el control sobre la aplicación.
14. La aplicación tendrá reversibilidad, es decir, permitirá deshacer cualquier acción realizada por el usuario, ya sea un error o un cambio futuro.

## 4.4 Modelo de datos

En este apartado se expondrá el modelo de datos de la aplicación. El objetivo del diseño del modelo de datos es representar la información del sistema de forma ordenada y óptima. A continuación, se detallará la descripción de los modelos de la aplicación.

### 4.4.1 Descripción de los modelos

Para representar correctamente la estructura del modelo de datos de la aplicación, se hará una descomposición de las colecciones de documentos en las que está formada la base de datos no relacional del sistema. Algunas de las colecciones forman parte de otras colecciones. En primer lugar, se detallarán las colecciones padre, que no forman parte de otra colección. Después se mostrarán las colecciones hijas o subcolecciones.

#### 4.4.1.1 Colección Users

Se trata de la colección que almacena toda la información relativa a los usuarios registrados en la aplicación. Cada vez que se registra un usuario, se añadirá un elemento a esta colección. La siguiente tabla muestra en detalle la colección de usuarios del sistema:

Atributos de la colección	Tipo	Descripción
address	string	Dirección del usuario
attendance_downloaded	boolean	Estado del certificado de asistencia descargado
bill	collection	Información relativa a la factura
check_payment	boolean	Estado del justificador de pago
city	string	Ciudad del usuario
country	string	País del usuario
email	string	Email del usuario
first_name	string	Nombre del usuario
id	string	ID del usuario
invoice_downloaded	boolean	Estado de la factura descargada
last_name	string	Apellidos del usuario
papers	collection	Colección que recoge los artículos del usuario
password	string	Contraseña cifrada del usuario

payment_file	collection	Información relativa al justificante de pago
phone	number	Número de teléfono del usuario
postal_code	string	Código postal del usuario
presentation_downloaded	boolean	Estado de certificado de presentación descargado
state	string	Estado o comunidad del usuario
tax	number	Tarifa a pagar por el usuario
title	string	Título del usuario
university_company	string	Universidad, compañía o colegio

Tabla 4: Colección Users

#### 4.4.1.2 Colección Config

Es la colección que almacena toda la información relativa a la configuración de la aplicación. Esta información solo será accesible para el usuario administrador. La siguiente tabla muestra en detalle la colección de configuración del sistema:

Atributos de la colección	Tipo	Descripción
CIF	string	CIF del CMMSE
ISBN	string	Número ISBN del CMMSE
ISSN	string	Número de ISSN del CMMSE
bill_number	number	Número de factura a asignar
certificate_signature	date	Fecha de firma de certificados
cmmse_opened	boolean	Estado del congreso
conference_end_day	date	Fecha de fin del congreso
conference_initial_day	date	Fecha de comienzo del congreso
conference_place	string	Lugar del congreso
conference_url	string	Enlace a la web del congreso
conference_year	number	Año del congreso
email_opened	boolean	Estado de gestión de envío de correos
email_password	string	Contraseña del correo
email_sender	string	Dirección de envío de correo
email	array	Conjunto de direcciones en copia de correos
fee_to_pay	number	Tarifa estándar para facturas
id	string	ID de la configuración
invoice_signature	date	Fecha de firma de facturas
root_password	string	Contraseña cifrada del usuario administrador
root_user	string	Nombre de usuario del usuario administrador
send_term_opened	boolean	Estado de subida de artículos al sistema

Tabla 5: Colección Config

#### 4.4.1.3 Colección Conferences

La colección *Conferences* es la que almacena todas las conferencias del sistema. El usuario administrador podrá añadir una nueva conferencia a la colección para que los usuarios puedan indicar al subir un artículo que ese artículo pertenece a esa conferencia. La siguiente tabla muestra en detalle la colección de conferencias del sistema:

Atributos de la colección	Tipo	Descripción
<code>id</code>	string	ID de la conferencia
<code>value</code>	string	Título de la conferencia

Tabla 6: Colección Conferences

#### 4.4.1.4 Colección Bill

La colección *Bill* es una subcolección de *Users*. En ella se almacenan los datos relativos a la factura del usuario. La siguiente tabla muestra en detalle la colección de factura de los usuarios:

Atributos de la colección	Tipo	Descripción
<code>CIF</code>	string	CIF para la factura
<code>address</code>	string	Dirección de facturación
<code>first_name</code>	string	Nombre para la factura
<code>invoice_number</code>	number	Número de factura
<code>last_name</code>	string	Apellido para la factura
<code>title</code>	string	Título para la factura
<code>university_company</code>	string	Universidad, compañía o colegio para la factura

Tabla 7: Colección Bill

#### 4.4.1.5 Colección Papers

La colección *Papers* es una subcolección de *Users*. En ella se almacenan los datos relativos a los artículos subidos por el usuario. La siguiente tabla muestra en detalle la colección de artículos de los usuarios:

Atributos de la colección	Tipo	Descripción
<code>authors</code>	collection	Información relativa a los autores
<code>id_file</code>	string	ID del archivo del artículo
<code>date</code>	date	Fecha de subida del artículo
<code>minisymposium</code>	string	Título de la conferencia a la que pertenece

name_file	string	Nombre del archivo del artículo
poster	boolean	Valor que indica si es el artículo es un poster
state	string	Estado del artículo
title	string	Título del artículo
url_file	string	Dirección del archivo en Firebase

Tabla 8: Colección Papers

#### 4.4.1.6 Colección Payment\_file

La colección *Payment\_file* es una subcolección de *Users*. En ella se almacenan los datos relativos al justificante de pago subido por el usuario. La siguiente tabla muestra en detalle esta subcolección:

Atributos de la colección	Tipo	Descripción
id_file	string	ID del archivo del justificante de pago
name_file	string	Nombre del archivo del justificante de pago
url_file	string	Dirección del archivo en Firebase

Tabla 9: Colección Payment\_file

#### 4.4.1.7 Colección Authors

La colección *Authors* es una subcolección de *Papers*. En ella se almacenan los datos relativos a los autores de cada artículo subido por el usuario. La siguiente tabla muestra en detalle esta subcolección:

Atributos de la colección	Tipo	Descripción
author	number	Número de autor del artículo
email	string	Dirección email del autor
first_name	string	Nombre del autor
last_name	string	Apellidos del autor

Tabla 10: Colección Authors

Una vez vistas todas las colecciones existentes, en la siguiente figura se detalla un diagrama completo de la base de datos (Cloud Firestore). Dentro de Firebase, se pueden crear proyectos independientes entre sí. El proyecto creado en Firebase para alojar la aplicación se llama CMMSE. Del proyecto, en la base de datos se alojan las colecciones tal como se indica en el siguiente diagrama.

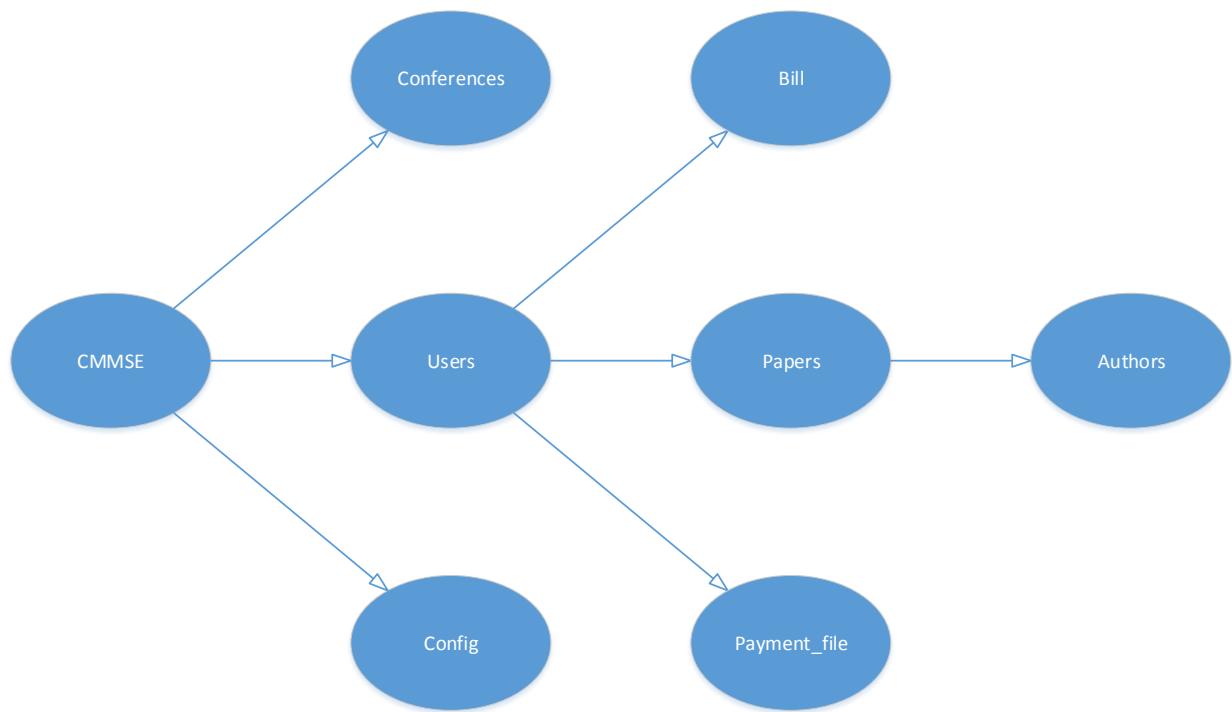


Figura 33: Diagrama estructural de la base de datos

## **DOCUMENTO 5: DISEÑO DEL SISTEMA**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 5.1 Introducción

Este documento tiene como objetivo el desarrollo del diseño de la aplicación web. Para ello se describirán varios diagramas que ayuden a entender el diseño del sistema. En los siguientes apartados se entrará en detalle en los siguientes diagramas: de paquetes, de diseño de componentes, de interacción y de diseño de servicios. Además, se mostrará el diseño de la interfaz de usuario, común a todas las pantallas de la aplicación.

## 5.2 Diagrama de paquetes

Un diagrama de paquetes representa las dependencias entre los paquetes que componen un modelo, es decir, muestra cómo un sistema está dividido en directorios, suministrando una descomposición jerárquica del sistema. Esta jerarquía es fundamental en una aplicación web que use el patrón de diseño MVC (Modelo Vista Controlador) como la que se presenta en este proyecto. En el siguiente diagrama se ven los paquetes y como están relacionados entre sí:

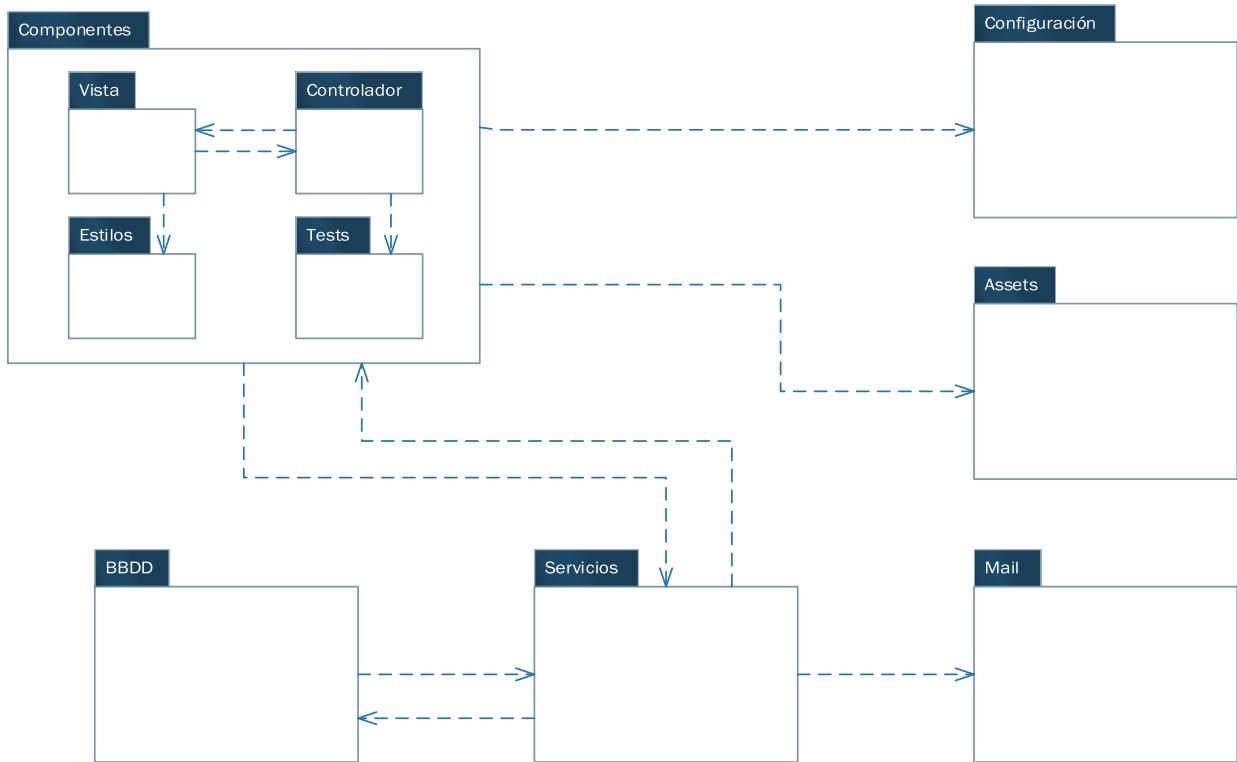


Figura 34: Diagrama de paquetes del sistema

Una vez visto el diagrama de paquetes, se hará una breve descripción de cada uno de los paquetes de la aplicación.

### 5.2.1 Componentes

Angular es un *framework* que se estructura principalmente en componentes. Estos componentes se integran mediante módulos y pueden ser usados en la aplicación tantas veces como se desee. Cada componente necesita al menos una vista y un controlador para funcionar. Además, se pueden añadir hojas de estilo y test.

## 5.2.2 Vista

La vista de un componente es un fichero HTML. Angular permite modificar dinámicamente el contenido HTML en el DOM de la aplicación con una sintaxis única de Angular. El código HTML de la vista de un componente puede estar embebido dentro del controlador, aunque no es recomendable.

## 5.2.3 Controlador

Se trata de un fichero en código TypeScript que maneja todo el componente. En él se pueden declarar clases, atributos y funciones que pueden ser usadas en la interacción con la vista. Además, se pueden declarar los ciclos de vida de cada componente, muy útiles para la interacción entre los diferentes componentes y servicios de la aplicación.

## 5.2.4 Estilos

Los estilos de los componentes de la aplicación se definen en archivos Sass. Este metalenguaje se trata de una variación de CSS que permite extender la funcionalidad de este lenguaje, facilitando el desarrollo de las hojas de estilos. Los estilos de un componente afectan únicamente a ese componente. Durante la compilación todos los archivos Sass son traducidos a CSS.

## 5.2.5 Tests

Los tests garantizan la fiabilidad necesaria para que un componente no contenga errores y no falle. Estos ficheros usan una sintaxis especial para definir los datos. Angular usa estos ficheros para ejecutar una prueba concreta para cada componente y así comprobar si contienen errores.

## 5.2.6 Configuración

La configuración es un fichero que contiene una constante llamada ‘AppConfig’. En esta constante se definen aquellos valores que son usados en diferentes sitios de la aplicación. Los componentes pueden acceder a este fichero para así no replicar el mismo código en todos los componentes.

## 5.2.7 Assets

Se trata de un directorio que contiene los ficheros de internacionalización, las imágenes, y el archivo Sass con los estilos generales de la aplicación.

## 5.2.8 Servicios

Los servicios forman parte de la estructura del patrón de diseño MVC. Es la capa que recoge los datos de la base de datos y los envía a la vista. También es la capa que recoge los eventos o llamadas realizadas en la vista para interactuar con la base de datos. Además, existe un servicio que encripta y desencripta los datos delicados; y un servicio encargado de enviar correos electrónicos cuando los usuarios realizan ciertas acciones.

## 5.2.9 Mail

Es un conjunto de archivos que se ejecutan en el lado del servidor. Estos archivos esperan la llamada desde el lado del cliente para enviar un correo electrónico diferente dependiendo de la acción realizada por el usuario. Estos archivos necesitan un servidor web Express para su ejecución en Node.js.

## 5.2.10 Base de datos

La base de datos es la capa donde se almacenan los datos de la aplicación. Alojada en Cloud Firestore de Firebase, el sistema puede acceder a ella por medio de un servicio que solicita crear, actualizar o borrar datos.

## 5.3 Diagrama de diseño de componentes

Otra parte fundamental en el diseño de la aplicación de este trabajo fin de grado es el diagrama de componentes. Angular es un *framework* basado en componentes, por lo que es importante diseñar una estructura robusta de componentes.

Para seguir una línea de trabajo común, se ha decidido crear un componente único por pantalla. Cada componente por lo tanto tiene una interfaz personalizada que se adecúa a las necesidades de cada pantalla.

Un punto importante que destacar es el uso de Angular Material dentro de los componentes de la aplicación. Angular Material es una librería desarrollada por Google enfocada a una normativa de diseño lanzada en 2014. La versión 5 de Angular permitía incluir esta librería a cualquier proyecto de Angular. Gracias a esta librería todos los componentes de la aplicación pueden usar los componentes creados para Angular Material aportando fiabilidad, usabilidad y un diseño moderno e intuitivo para los usuarios.

La siguiente figura muestra el diagrama de componentes de la aplicación y sus relaciones directas de navegabilidad.

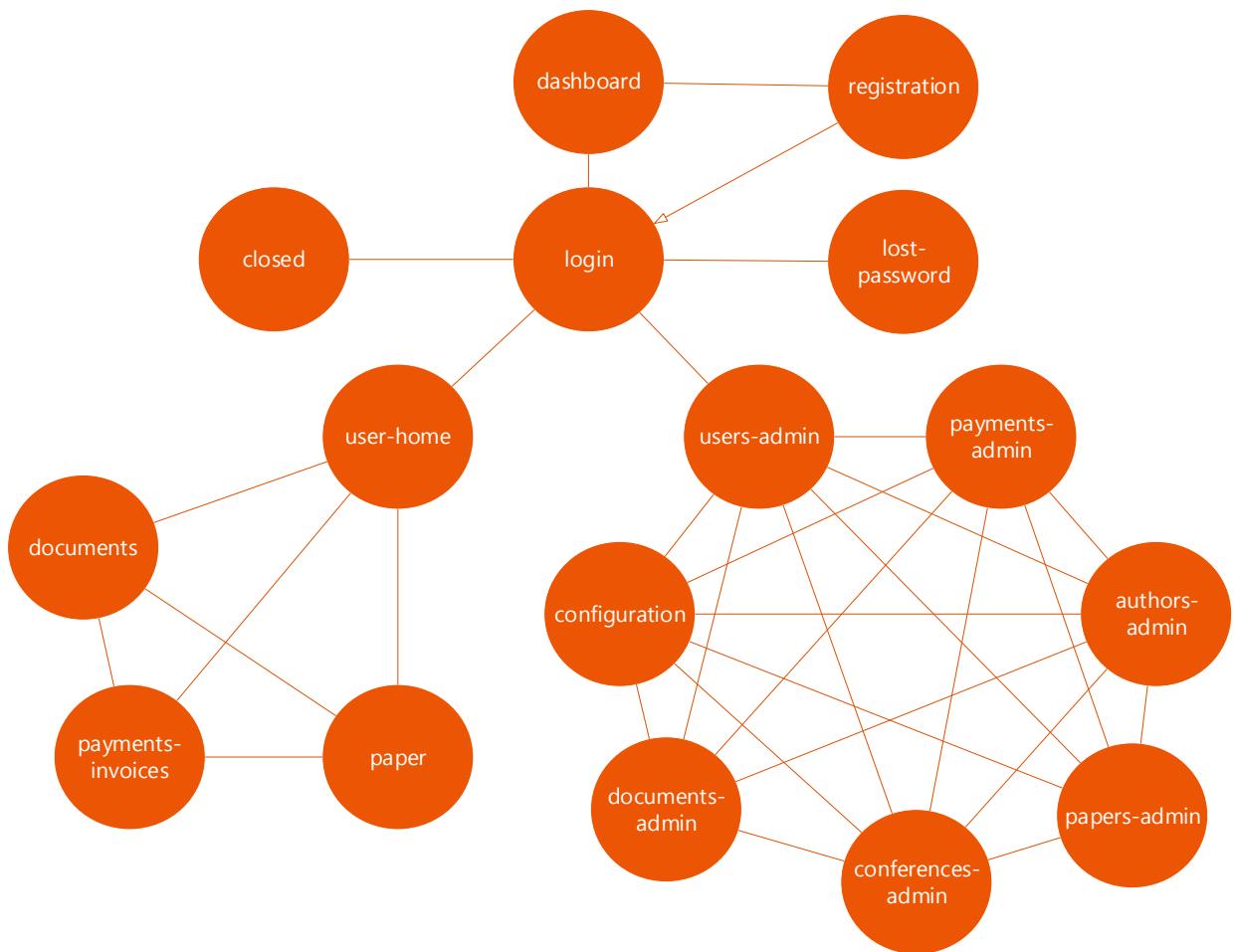


Figura 35: Diagrama de componentes

Como se puede ver, el diagrama muestra tres partes bien definidas de la aplicación: acceso y registro de usuarios, gestión de usuarios asistentes al congreso y gestión de usuario administrador. El nexo entre estas partes es el *login*.

## 5.4 Diagrama de interacción

El objetivo de este apartado es presentar el diagrama de interacción del usuario con el sistema. Una vez más, puede verse en el siguiente diagrama que el patrón MVC está estrechamente ligado al diseño del sistema ya que el usuario interactúa con el sistema siguiendo este patrón.

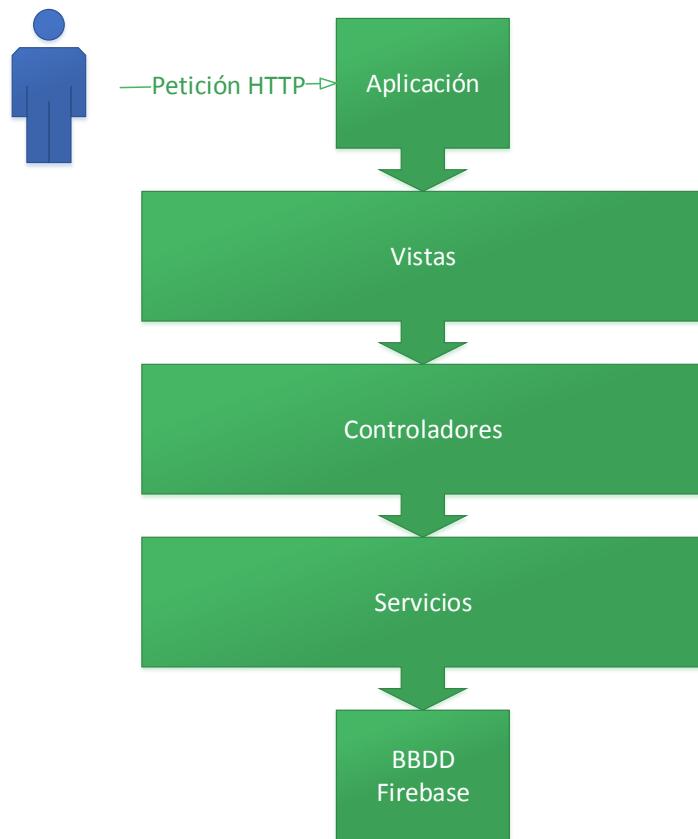


Figura 36: Diagrama de interacción

En el momento que un usuario realice una petición HTTP a la aplicación web a través de un navegador, la vista lanzará un evento tras la acción del usuario. El evento lo capturará el controlador del componente y será entonces cuando el control pase al servicio encargado de interactuar con la base de datos. La base de datos será modificada o consultada.

Cabe destacar que el usuario no tiene acceso directo al controlador, al servicio o a la base de datos. A su vez la vista no tendrá acceso a los servicios o a la base de datos. Al revés ocurre lo mismo.

Este proceso termina cuando la consulta, creación, modificación o eliminación de datos en la base de datos llega de vuelta al usuario siguiendo el flujo contrario. De la base de datos al servicio, del servicio al controlador, del controlador a la vista y de la vista a la aplicación final totalmente legible por el usuario.

## 5.5 Diagrama de diseño de servicios

En este apartado se explicará el diseño de los servicios de la aplicación. Todos los servicios de la aplicación están en un directorio común. El diagrama de diseño de los servicios es el siguiente:

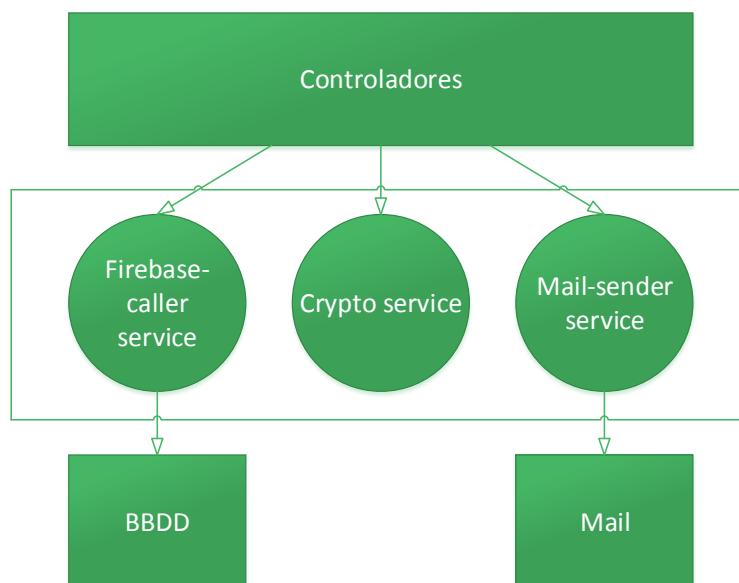


Figura 37: Diagrama de diseño de servicios

Los controladores de los componentes podrán utilizar tres servicios diferentes dentro de la aplicación. Tal como se ve en la figura anterior, el servicio *firebase-caller* es el encargado de hacer las llamadas a la base de datos. Por otro lado, el servicio *crypto* se encarga de encriptar y desencriptar los datos delicados para que puedan ser leídos o almacenados en la base de datos de forma segura. Por último,

el servicio *mail-sender* se encarga de llamar al servidor para enviar correos electrónicos a los usuarios o al sistema.

## 5.6 Diseño de la interfaz

El diseño de la interfaz de usuario es una de las partes más importantes de la aplicación. Los usuarios estarán satisfechos con el uso de la aplicación si la interfaz es amigable, intuitiva, simple, accesible y usable. Además, es fundamental que la interfaz de todas las pantallas sea similar.

La siguiente figura muestra cómo es el diseño general de la aplicación. Este diseño es usado para todas las pantallas:

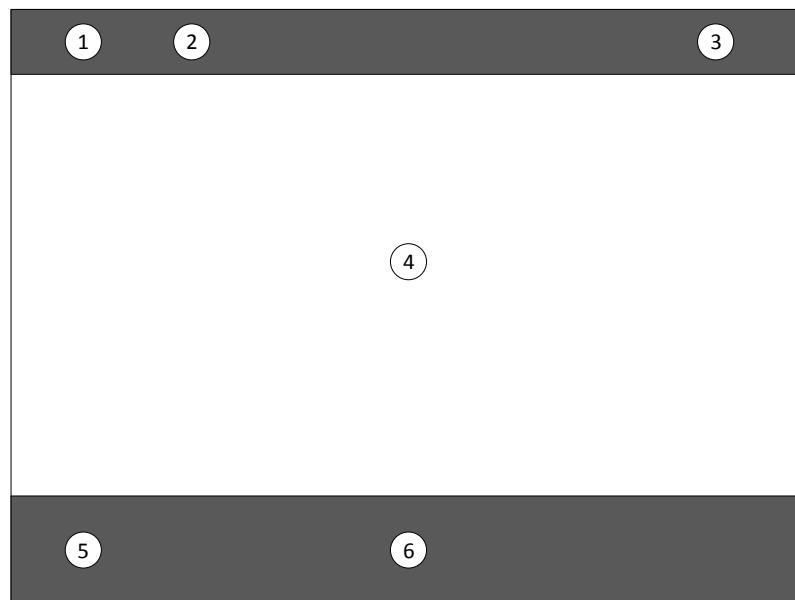


Figura 38: Diseño general de la interfaz

La figura anterior muestra el esqueleto del diseño de una pantalla de la aplicación. A continuación, se detallan los números de la imagen con una breve descripción:

- 1) Se trata del logotipo, el título y el año del congreso.

- 2) Cuando el usuario entra en sesión, se muestra el menú navegable. Este menú varía dependiendo si el usuario es el administrador o un asistente al congreso. Si el usuario no ha entrado en sesión este menú no se mostrará.
- 3) Se trata de un submenú que muestra el nombre de usuario y el botón de cerrar sesión si éste ha iniciado sesión. El submenú será distinto si el usuario no ha iniciado sesión ya que mostrará la navegabilidad para ir a la página de información del congreso, para volver a la página inicial de la aplicación o para iniciar sesión.
- 4) Se trata del contenido principal de cada pantalla. A pesar de variar el contenido entre las pantallas, se mantendrá un aspecto similar en todas ellas, respetando tipo y tamaño de letras, orientación, títulos e iconos.
- 5) El usuario podrá cambiar de idioma en todas las pantallas en este punto.
- 6) Información relativa a la aplicación web.

Los puntos 1, 2 y 3 forman parte de la cabecera de la pantalla. Los puntos 5 y 6 forman parte del pie de página. Por último, el punto 4 forma parte del cuerpo de la pantalla y contará con varias secciones bien definidas por títulos que las identifiquen.

## **DOCUMENTO 6: PRUEBAS**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 6.1 Introducción

En este documento se entrará en detalle en las pruebas realizadas durante el desarrollo de este proyecto. Todas estas pruebas han servido para garantizar que el sistema sea estable, respondiendo como se espera ante las diferentes situaciones posibles durante la ejecución y el uso del sistema.

Se ha dividido el documento en cuatro apartados: pruebas unitarias, pruebas de integración y sistema, pruebas de rendimiento y por último pruebas de usabilidad y accesibilidad. En cada apartado se explicarán los resultados obtenidos.

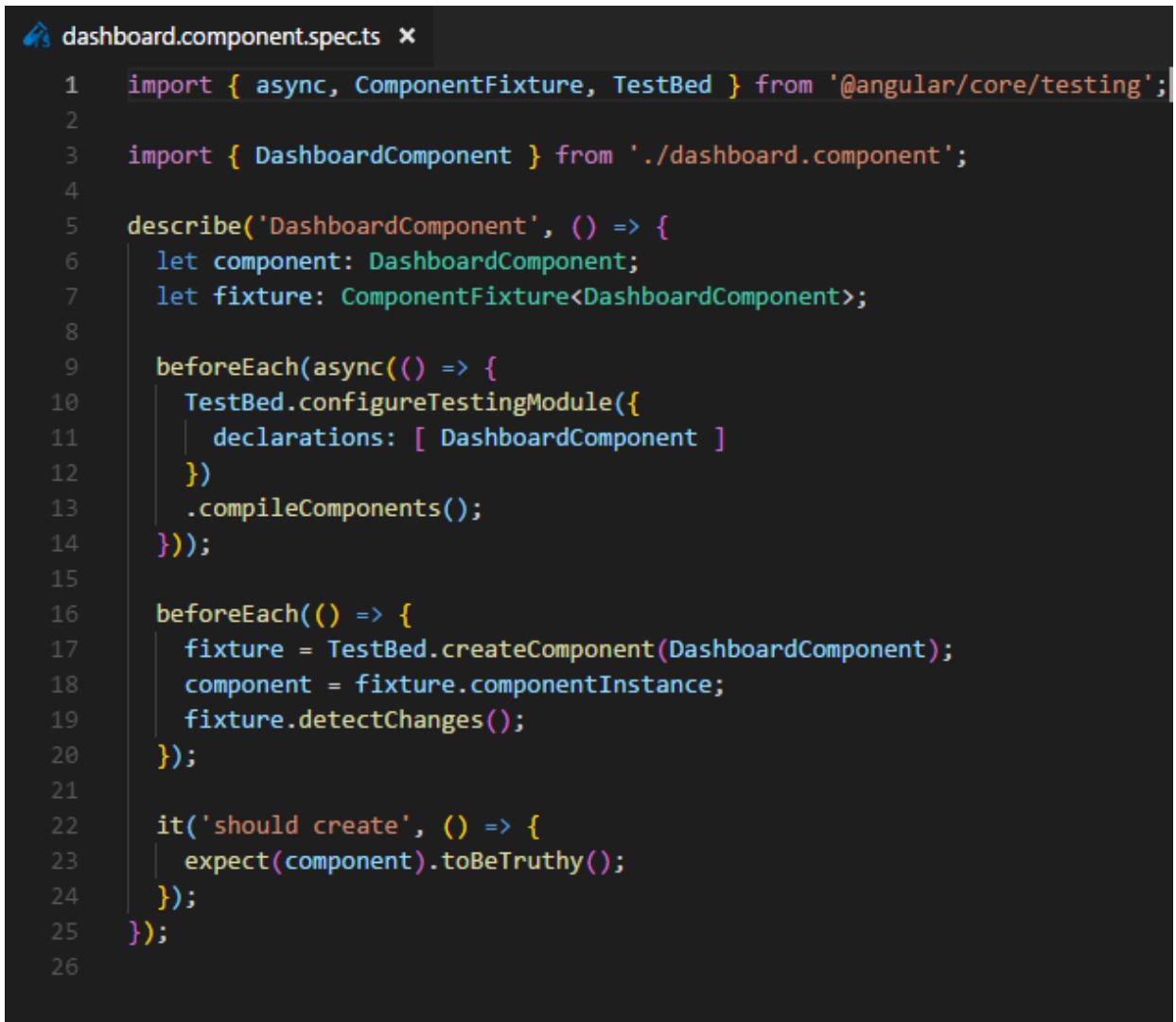
## 6.2 Pruebas unitarias

Las pruebas unitarias son fundamentales para comprobar que el código funciona correctamente. Sirven para asegurar que cada unidad de código sea eficiente, que hace lo que tiene que hacer, que el nombre, los tipos de parámetros, el tipo de lo que se devuelve y sus estados sean válidos.

En Angular las pruebas unitarias se pueden realizar para cada fichero TypeScript del proyecto. Para ello se pueden crear ficheros con la extensión “.spec.ts”. Estos ficheros con una sintaxis especial ejecutan el fichero a probar de forma ficticia, con datos no reales.

Para ejecutar las pruebas Angular CLI provee del comando “ng test”, por lo que no es necesario instalar ninguna herramienta para realizar o ejecutar las pruebas unitarias en un proyecto de Angular. Este comando ejecuta todos los ficheros de pruebas unitarias y devuelve el resultado en el navegador.

En la siguiente figura se muestra la sintaxis básica de una prueba unitaria de la aplicación:



```

1 import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { DashboardComponent } from './dashboard.component';
4
5 describe('DashboardComponent', () => {
6   let component: DashboardComponent;
7   let fixture: ComponentFixture<DashboardComponent>;
8
9   beforeEach(async(() => {
10     TestBed.configureTestingModule({
11       declarations: [ DashboardComponent ]
12     })
13     .compileComponents();
14   }));
15
16   beforeEach(() => {
17     fixture = TestBed.createComponent(DashboardComponent);
18     component = fixture.componentInstance;
19     fixture.detectChanges();
20   });
21
22   it('should create', () => {
23     expect(component).toBeTruthy();
24   });
25 });
26

```

Figura 39: Prueba unitaria

En la figura se puede comprobar que se trata de una prueba unitaria del componente *dashboard* de la aplicación. En este test se comprueba que el componente se crea, de no ser así saltaría la excepción indicando el error de la prueba.

En la aplicación todos los componentes y servicios tienen una prueba unitaria específica creada para probar las diferentes posibilidades de cada componente.

## 6.3 Pruebas de integración y de sistema

Las pruebas de integración y de sistema servirán para probar la funcionalidad total de la aplicación. Las ventajas principales de este tipo de pruebas son la verificación de las especificaciones de diseño; navegación entre componentes y pantallas satisfactoria; y comprobación de ingreso, modificación y borrado de datos correctamente.

En la siguiente tabla se detalla el desarrollo de estas pruebas sobre los casos de uso más relevantes de la aplicación:

<b>Caso de uso – Registro de usuario</b>	
<b>Prueba</b>	<b>Resultado Esperado</b>
Registro de usuario no existente.	El sistema registra al nuevo usuario correctamente.
	<b>Resultado obtenido</b>
	El sistema registra al nuevo usuario correctamente.
<b>Prueba</b>	<b>Resultado esperado</b>
Registro de usuario ya existente (con el mismo mail).	El sistema no registra al usuario y lo notifica.
	<b>Resultado obtenido</b>
	El sistema no registra al usuario y se recibe una notificación.
<b>Prueba</b>	<b>Resultado esperado</b>
Registro de usuario con campos obligatorios sin llenar.	El sistema no registra al usuario y notifica los campos que faltan por llenar.
	<b>Resultado obtenido</b>
	Tal y como se esperaba el sistema no añade al usuario y le notifica los fallos.
<b>Prueba</b>	<b>Resultado esperado</b>
Registro de usuario con contraseña no válida.	El sistema no registra al usuario y notifica que la contraseña no cumple con los requisitos de seguridad.
	<b>Resultado obtenido</b>

	Tal y como se esperaba el sistema no registra al usuario y le notifica los errores de validación.
<b>Caso de uso – Login de usuario</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Intento de <i>login</i> con un usuario existente en el sistema.	El usuario inicia sesión correctamente en el sistema.
	<b>Resultado obtenido</b>
	El usuario inicia sesión en la aplicación correctamente.
<b>Prueba</b>	<b>Resultado esperado</b>
Intento de acceso a una sección no autorizada por el usuario.	El usuario es redirigido a la pantalla de inicio.
	<b>Resultado obtenido</b>
	Efectivamente no se le permite el acceso al usuario y se le notifica que no tiene permisos.
<b>Prueba</b>	<b>Resultado esperado</b>
Intento de <i>login</i> con un mail no existente en la base de datos.	El sistema notifica al usuario que el usuario no es correcto.
	<b>Resultado obtenido</b>
	El sistema notifica al usuario que el usuario no es correcto.
<b>Caso de uso – Recuperar contraseña</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Recuperación de contraseña usuario registrado.	El sistema envía un correo al usuario con la nueva contraseña.
	<b>Resultado obtenido</b>
	El sistema envía un correo al usuario con la nueva contraseña.
<b>Prueba</b>	<b>Resultado esperado</b>
Recuperación de contraseña usuario no registrado.	El sistema notifica al usuario que el usuario no se encuentra en la base de datos.
	<b>Resultado obtenido</b>

	El sistema notifica al usuario que el usuario no se encuentra en la base de datos.
<b>Caso de uso – Modificación información usuario asistente al congreso</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Edición de datos personales de un usuario.	El sistema actualiza los datos del usuario.
	<b>Resultado obtenido</b>
	Efectivamente se actualiza la información del usuario.
<b>Caso de uso – Subir justificante de pago del usuario asistente al congreso</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Subida de justificante de pago de un usuario.	El sistema guarda el archivo subido y lo notifica al usuario.
	<b>Resultado obtenido</b>
	El sistema guarda el archivo subido y lo notifica al usuario.
<b>Prueba</b>	<b>Resultado esperado</b>
Subida de otro justificante de pago de un usuario.	El sistema borra el archivo anterior, guardará el archivo nuevo y lo notifica al usuario.
	<b>Resultado obtenido</b>
	El sistema borra el archivo anterior, guardará el archivo nuevo y lo notifica al usuario.
<b>Caso de uso – Subir artículo de usuario asistente al congreso</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Subida de artículo de usuario.	El sistema guarda el archivo subido y lo notifica al usuario.
	<b>Resultado obtenido</b>
	El sistema guarda el archivo subido y lo notifica al usuario.
<b>Prueba</b>	<b>Resultado esperado</b>
Subida de nuevo artículo.	El sistema guarda el archivo subido y lo notifica al usuario.
	<b>Resultado obtenido</b>
	El sistema guarda el archivo subido y lo notifica al usuario.
<b>Prueba</b>	<b>Resultado esperado</b>

Edición de artículo subido previamente.	El sistema actualiza los datos del artículo editado y lo notifica al usuario.
	<b>Resultado obtenido</b>
	El sistema actualiza los datos del artículo editado y lo notifica al usuario.
<b>Caso de uso – Modificación usuario desde el rol de usuario administrador</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Modificación de datos de usuario.	El sistema modifica los datos en la base de datos y lo notifica al usuario administrador.
	<b>Resultado obtenido</b>
	Efectivamente el sistema modifica los datos y lo notifica.
<b>Prueba</b>	<b>Resultado esperado</b>
Borrado de usuario.	El sistema elimina de la base de datos al usuario seleccionado y lo notifica.
	<b>Resultado obtenido</b>
	El sistema elimina de la base de datos al usuario seleccionado y lo notifica.
<b>Caso de uso – Modificación artículo desde el rol de usuario administrador</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Modificación de los datos de un artículo.	El sistema modifica los datos en la base de datos y lo notifica al usuario administrador.
	<b>Resultado obtenido</b>
	El sistema modifica los datos en la base de datos y lo notifica al usuario administrador.
<b>Prueba</b>	<b>Resultado esperado</b>
Borrado de artículo.	El sistema elimina de la base de datos el artículo seleccionado y lo notifica.
	<b>Resultado obtenido</b>
	El sistema elimina de la base de datos el artículo seleccionado y lo notifica.

<b>Prueba</b>	<b>Resultado esperado</b>
Modificación estado de un artículo.	El sistema almacena en la base de datos el nuevo estado del artículo modificado y envía un correo notificando la acción.
	<b>Resultado obtenido</b>
	El sistema almacena en la base de datos el nuevo estado del artículo modificado y envía un correo notificando la acción.
<b>Caso de uso – Descarga de documentos</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Descarga de facturas / factura de usuario.	El sistema descarga un archivo con los datos de el/los usuario/s en forma de factura.
	<b>Resultado obtenido</b>
	El sistema descarga un archivo con los datos de el/los usuario/s en forma de factura.
<b>Prueba</b>	<b>Resultado esperado</b>
Descarga de etiquetas identificativas.	El sistema descarga un archivo con las etiquetas identificativas de cada usuario.
	<b>Resultado obtenido</b>
	El sistema descarga un archivo con las etiquetas identificativas de cada usuario.
<b>Caso de uso – Modificación de la configuración desde el rol de administrador</b>	
<b>Prueba</b>	<b>Resultado esperado</b>
Modificación de datos de la configuración.	El sistema almacena en la base de datos la nueva configuración con los datos modificados y lo notifica al usuario.
	<b>Resultado obtenido</b>
	Efectivamente el sistema almacena los nuevos datos y lo notifica al usuario.

*Tabla 11: Pruebas de integración y sistema*

## 6.4 Pruebas de usabilidad y accesibilidad

Para las pruebas de usabilidad y accesibilidad se ha decidido seguir una guía heurística completa que evalúa este tipo de pruebas. Esta guía está basada en criterios tales como navegación, lenguaje, control o retroalimentación, entre otros.

### 6.4.1 Criterios generales

Criterios	¿Se ha cumplido?
<b>¿Cuáles son los objetivos de la aplicación web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?</b> La aplicación se ha diseñado para ser usada en producción, con unos requisitos existentes previamente y nuevas funcionalidades.	Sí
<b>¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?</b> La aplicación consta de una URL acorde al tema del proyecto, corta y sencilla. La navegación interna también. Todas ellas son claras y permanentes.	Sí
<b>¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web?</b> La pantalla principal contiene una información breve y concisa de los objetivos principales de la aplicación para los usuarios que la utilizarán.	Sí
<b>¿La estructura general del sitio web está orientada al usuario?</b> La estructura está orientada para los dos roles disponibles en la aplicación.	Sí
<b>¿El look &amp; feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web?</b> Al tratarse de un sistema en producción y para un congreso en concreto, se ha tenido en cuenta el diseño del logotipo y los colores para que se corresponda al contenido.	Sí
<b>¿Es coherente el diseño general del sitio web?</b> Todas las pantallas de la aplicación siguen una línea de diseño similar por lo que es coherente para el usuario.	Sí
<b>¿Es reconocible el diseño general del sitio web?</b>	Sí

El diseño de la aplicación es único y contiene estilos únicos personalizados de los requisitos del sistema.	
<b>¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza?</b> Al ser un proyecto en producción, la aplicación seguirá evolucionando a medida que la dirección del CMMSE quiera introducir cambios o mejoras.	Sí

Tabla 12: Pruebas de usabilidad y accesibilidad - Criterios generales

#### 6.4.2 Criterios de identidad e información

Criterios	¿Se ha cumplido?
<b>¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?</b> El logotipo del CMMSE está presente en todas las pantallas de la aplicación por lo que se muestra claramente la identidad de la aplicación.	Sí
<b>¿El logotipo es significativo, identifiable y suficientemente visible?</b> El logotipo forma parte de la firma del CMMSE ya que se lleva mostrando en la aplicación previa desde su creación. Además, está visible en todas las pantallas y contrasta bien en color con el resto de la aplicación.	Sí
<b>¿El eslogan o tagline expresa realmente qué es la empresa y qué servicios ofrece?</b> El título de la aplicación define claramente el alcance del CMMSE y a qué usuarios va dirigida.	Sí
<b>¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster', etc.?</b> Uno de los botones de la pantalla principal y un botón de la cabecera navegan al sitio web que ofrece información detallada del CMMSE.	Sí

Tabla 13: Pruebas de usabilidad y accesibilidad - Criterios de identificación e información

### 6.4.3 Criterios de lenguaje y redacción

Criterios	¿Se ha cumplido?
<b>¿El sitio web habla el mismo lenguaje que sus usuarios?</b> La aplicación puede verse tanto en español como en inglés. Dado que se trata de un congreso internacional, todos los asistentes dominan alguno de estos idiomas.	Sí
<b>¿Emplea un lenguaje claro y conciso?</b> Todas las secciones de la aplicación muestran un diseño sencillo y se ayudan de iconos que facilitan que el usuario entienda el contenido de cada sección.	Sí
<b>¿Es amigable, familiar y cercano?</b> El lenguaje y la redacción es amigable, familiar, cercano y acorde a las necesidades del CMMSE.	Sí

Tabla 14: Pruebas de usabilidad y accesibilidad - Criterios de lenguaje y redacción

### 6.4.4 Criterios de rotulado

Criterios	¿Se ha cumplido?
<b>¿Los rótulos son significativos?</b> Todas las secciones, botones y mensajes tienen palabras clave que identifican claramente las acciones que el usuario puede realizar.	Sí
<b>¿Usa rótulos estándar?</b> Los rótulos son idénticos para todas las pantallas.	Sí
<b>¿El título de las páginas es correcto? ¿Ha sido planificado?</b> Cada página tiene un título que describe perfectamente el alcance de lo que el usuario puede hacer. Ha sido planificado.	Sí

Tabla 15: Pruebas de usabilidad y accesibilidad - Criterios de rotulado

## 6.4.5 Criterios de navegación

Criterios	¿Se ha cumplido?
<b>¿Los enlaces son fácilmente reconocibles como tales? ¿su caracterización indica su estado?</b> Los enlaces de la aplicación son iconos o botones reconocibles que indican con color azul que es un enlace.	Sí
<b>¿En menús de navegación se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística?</b> El menú de navegación tiene pocos elementos, por lo que es fácil recordar dónde se encuentra cada parte de la aplicación.	Sí
<b>¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace?</b> Cada ícono o botón tiene una descripción visual o de texto sobre la acción que se va a realizar al hacer clic en el enlace.	Sí
<b>¿Se ha controlado que no haya enlaces que no llevan a ningún sitio?</b> Todos los enlaces han sido probados en todos los casos posibles por lo que no hay ningún enlace roto.	Sí
<b>¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación?</b> Todas las pantallas de la aplicación tienen un enlace para volver a la pantalla anterior.	Sí
<b>¿Se ha evitado la redundancia de enlaces?</b> En las pantallas no hay redundancia de enlaces.	Sí
<b>¿Se ha controlado que no haya páginas "huérfanas"?</b> Las pantallas incluidas en la aplicación se han revisado para asegurar que no haya páginas huérfanas accesibles por los usuarios.	Sí

Tabla 16: Pruebas de usabilidad y accesibilidad - Criterios de navegación

## 6.4.6 Layout de la página

Criterios	¿Se ha cumplido?
<b>¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia?</b>	Sí

El contenido principal de cada pantalla se ha estructurado de tal forma que siempre quede en la zona central.	
<b>¿Se ha evitado la sobrecarga informativa?</b> La información relativa a cada pantalla solo se muestra en esa pantalla. Los contenidos de la aplicación web se han minimizado para que sea breve y concisa.	Sí
<b>¿Es una interfaz limpia, sin ruido visual?</b> Todas las pantallas cuentan con un estilo común, sin incluir objetos que no sean necesarios para la funcionalidad de la aplicación.	Sí
<b>¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?</b> Entre cada sección de cada pantalla hay una separación mínima que ayuda al usuario a dividir cada apartado y así no agobiar al usuario con demasiada información en corto espacio.	Sí
<b>¿Se ha controlado la longitud de página?</b> Toda la información de cada pantalla tiene una longitud pequeña. En ninguna pantalla hay más de tres secciones.	Sí

Tabla 17: Pruebas de usabilidad y accesibilidad - Criterios de layout de la página

#### 6.4.7 Criterios de búsqueda

Criterios	¿Se ha cumplido?
<b>¿La búsqueda se encuentra fácilmente accesible?</b> Todas las tablas de la aplicación cuentan con un campo de búsqueda de datos accesible para los usuarios.	Sí
<b>¿Es fácilmente reconocible como tal?</b> Sí. Los filtros de búsqueda indican al usuario en todo momento que se pueden usar.	Sí
<b>¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?</b> Sí. La búsqueda de los datos es instantánea.	Sí
<b>¿La caja de texto es lo suficientemente ancha?</b> La caja de texto de búsqueda ocupa todo el ancho de la pantalla.	Sí
<b>¿Asiste al usuario en caso de no poder ofrecer resultados para una consulta dada?</b> El sistema mostrará un resultado vacío cuando se haga una búsqueda que no obtenga resultados.	Sí

Tabla 18: Pruebas de usabilidad y accesibilidad - Criterios de búsqueda

## 6.4.8 Accesibilidad

Criterios	¿Se ha cumplido?
<b>¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?</b> La fuente y el tamaño son suficientemente grandes para su perfecta legibilidad.	Sí
<b>¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleados facilitan la lectura?</b> Todas las pantallas cuentan con el mismo tipo de fuente, alineación, espaciado y estilo de letra.	Sí
<b>¿Existe un alto contraste entre el color de fuente y el fondo?</b> Todas las fuentes en el cuerpo de las pantallas tienen color negro y el fondo de pantalla es de color blanco. Para la cabecera y el pie de página las fuentes tienen color blanco y el fondo es negro.	Sí
<b>¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla?</b> La aplicación funciona en todos los navegadores actuales y es responsive por lo que se visualiza bien tanto en móviles, tabletas y ordenadores.	Sí
<b>¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar plugins adicionales?</b> Esta aplicación funciona sin necesidad de la instalación de plugins adicionales.	Sí
<b>¿Se ha controlado el peso de la página?</b> El peso de la aplicación es mínimo ya que no consta de imágenes pesadas o vídeos.	Sí
<b>¿Se puede imprimir la página sin problemas?</b> Sí, se ha probado a imprimir todas las pantallas sin problema alguno.	Sí

Tabla 19: Pruebas de usabilidad y accesibilidad - Criterios de accesibilidad

## 6.4.9 Criterios de control y retroalimentación

Criterios	¿Se ha cumplido?
<b>¿Tiene el usuario todo el control sobre el interfaz?</b> El usuario puede controlar todas las acciones posibles en la aplicación si tiene los permisos adecuados.	Sí
<b>¿Se informa al usuario de lo que ha pasado?</b> Cada vez que el usuario interactúa con el sistema, este lo notifica con un mensaje.	Sí

<b>Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema?</b> Cuando algo no va bien, se muestra un aviso de error, así como su solución de una forma no alarmista y sencilla.	Sí
<b>¿Posee el usuario libertad para actuar?</b> En cualquier momento el usuario puede realizar cualquier acción disponible por lo que tiene total libertad.	Sí
<b>¿Se ha controlado el tiempo de respuesta?</b> Los tiempos de respuesta en las llamadas a la base de datos y los tiempos de ejecución y de carga de la aplicación son mínimos, en milisegundos.	Sí

Tabla 20: Pruebas de usabilidad y accesibilidad - Criterios de control y retroalimentación

## 6.5 Pruebas de rendimiento

En este apartado se realizará un estudio de los resultados obtenidos por algunas herramientas que miden el rendimiento, la optimización y la velocidad de una aplicación web. Se usarán dos herramientas actuales y potentes que miden estos factores. En primer lugar, se usará “PageSpeed Insights” y después se hará un análisis de “Lighthouse”, ambas herramientas de Google.

Durante el desarrollo del proyecto se han realizado una serie de buenas prácticas para obtener un buen rendimiento en la aplicación. Entre ellas están las siguientes:

- No duplicar código TypeScript (duplicidad de funciones).
- No incluir elementos pesados que ralenticen la aplicación (imágenes, vídeos, etc.)
- Incluir archivos minimizados para los estilos de la aplicación.
- Incluir archivos minimizados para el código JavaScript (TypeScript traducido) de la aplicación.
- Eliminar suscripciones a todos los observables creados en el código, evitando que se sobrecargue la aplicación.

## 6.5.1 Análisis PageSpeed Insights

PageSpeed Insights es una herramienta que permite medir la velocidad y la optimización de un sitio web. El análisis puede realizarse mediante la siguiente dirección:

- <https://developers.google.com/speed/pagespeed/insights/>

Desde la dirección anterior se puede introducir la URL de la página web a analizar. Tras unos minutos, la herramienta devuelve el análisis completo para la versión móvil y la versión de escritorio. La siguiente figura muestra el análisis para la versión móvil:

The screenshot shows the PageSpeed Insights interface for a mobile device. At the top, the URL 'https://cmmse-app.firebaseioapp.com/' is entered into the address bar, and a blue 'ANALYZE' button is visible. Below the address bar, there are two tabs: 'Mobile' (selected) and 'Desktop'. The main content area displays the analysis results. On the left, under 'Page Speed', it says 'Unavailable'. On the right, under 'Optimization', it says 'Medium' with a score of '64 / 100'. A note below states: 'PSI is currently showing a single-page report. Chrome User Experience Report [does not have sufficient real-world speed data](#) for this page, but PSI was still able to analyze this page to identify potential optimizations that may improve the speed of this page. Please investigate the recommendations below. [Learn more](#)'. To the right of this note is a preview of the mobile registration system's landing page, which includes sections for 'Step 1: Personal Information' and 'Step 2: Conference Information'.

Figura 40: PageSpeed Insights análisis móvil

Se puede comprobar que para la versión móvil se ha obtenido una puntuación de 64 puntos de 100 posibles en optimización.

Por otro lado, la siguiente figura muestra el análisis para la versión de escritorio:

### PageSpeed Insights

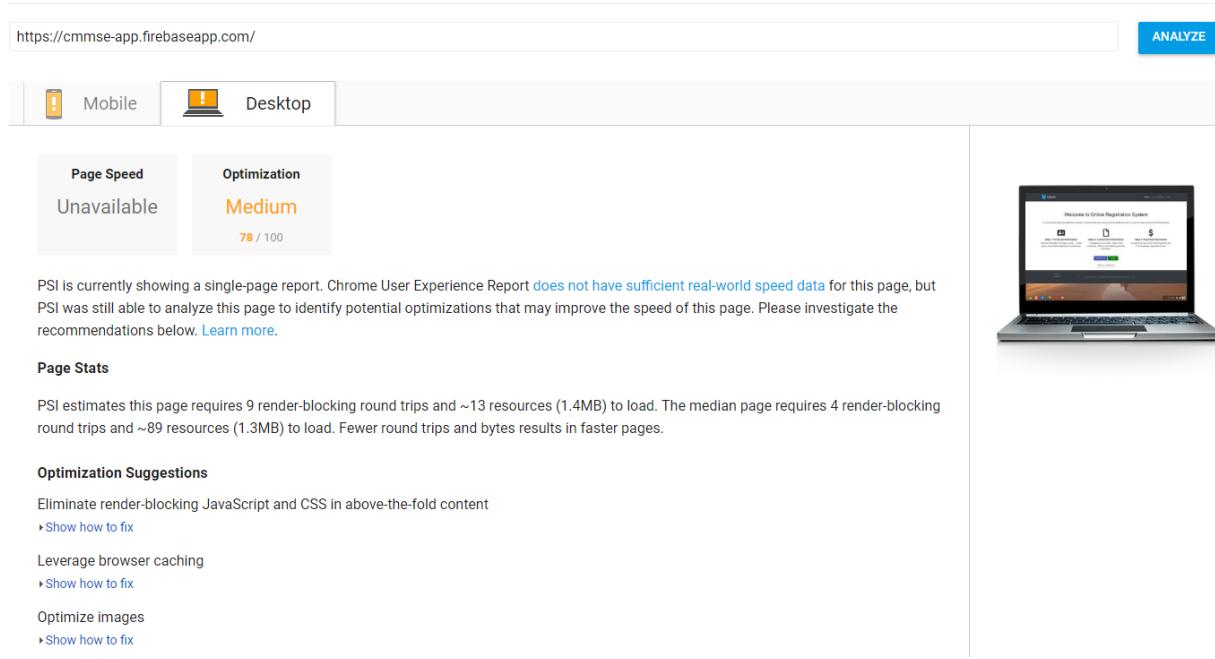


Figura 41: PageSpeed Insights análisis escritorio

Para la versión de escritorio la puntuación obtenida es de 78 puntos de 100 posibles en optimización. Como es una aplicación orientada a la versión de escritorio, se ha hecho más hincapié en optimizar el sistema para esta versión.

## 6.5.2 Análisis Lighthouse

Lighthouse es una herramienta que permite medir el rendimiento, la accesibilidad, el SEO, si es una aplicación web progresiva y si utiliza buenas prácticas:

- <https://developers.google.com/web/tools/lighthouse/>

Para realizar el análisis hay que instalar en el navegador Google Chrome la extensión de Lighthouse. Una vez instalada, desde la aplicación a analizar se puede generar el reporte. Tras unos

minutos, la herramienta devuelve el análisis completo. La siguiente figura muestra el análisis con todos los valores calculados del 0 al 100:

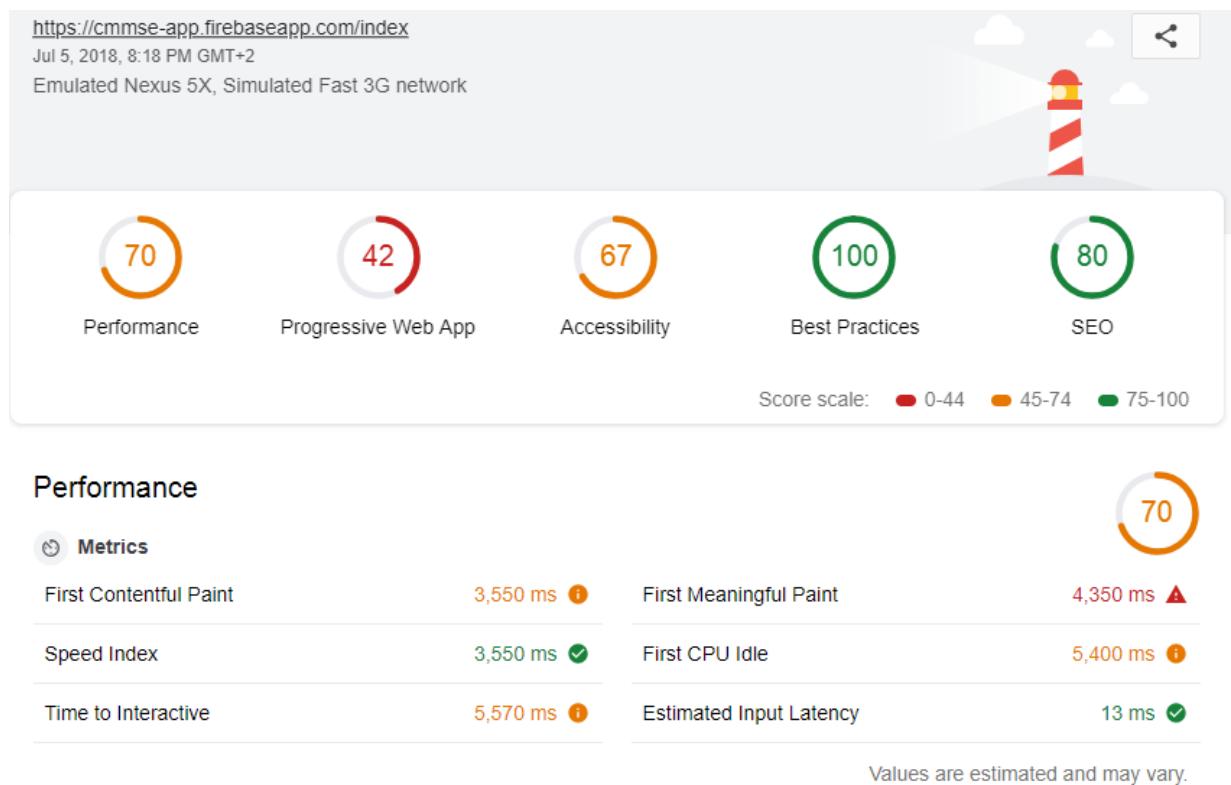


Figura 42: Lighthouse análisis

Como se puede apreciar, las cifras son buenas en general. En rendimiento la puntuación es de 70 puntos, por lo que puede todavía mejorarse.

La puntuación de aplicación web progresiva (*Progressive Web App*) es la más baja. Este término creado por Google para medir las aplicaciones web valida muchas características que se salen del objetivo de este trabajo fin de grado. Sin embargo, la lista de características analizadas en este apartado se puede consultar en el siguiente enlace: <https://developers.google.com/web/progressive-web-apps/checklist>.

La puntuación de accesibilidad es de 67 puntos. Esta puntuación puede mejorarse y para ello el análisis te ofrece algunos consejos que se pueden seguir.

La puntuación de buenas prácticas es de 100. Esto es debido a que no se muestran errores en la consola durante la ejecución de la aplicación, se usa el protocolo HTTPS, no se permite el pegado en campos de contraseña, se evitan APIs deprecadas, las imágenes se muestran correctamente, etc.

Por último, la puntuación de SEO es de 80 puntos. Aunque todavía puede ser mejorada, el análisis revela que se han pasado audiciones como una buena indexación y *crawling*, versión móvil amigable, etc.

## **DOCUMENTO 7: MANUALES**

**D. VIGIL RODRÍGUEZ, Guillermo  
TUTOR: D. RANILLA PASTOR, José  
COTUTOR: D. REDONDO LÓPEZ, José Manuel**

**FECHA: Julio 2018**



## 7.1 Introducción

El presente documento recogerá los siguientes manuales: manual de instalación y ejecución en modo local, manual de sustitución de la base de datos, manuales de despliegue de la aplicación y manual de usuario. Todos ellos indicarán los pasos necesarios para conseguir hacer funcionar el sistema usando cualquier plataforma de desarrollo, con cualquier dispositivo que tenga conexión a internet.

## 7.2 Manual de instalación y ejecución en modo local

Este manual permitirá instalar la aplicación de forma local. Para ello se deberán seguir los siguientes pasos:

1. Instalar **Node.js** en su versión LTS desde su página oficial: <https://nodejs.org/es/> . Node.js es el lenguaje del lado del servidor que se necesita para ejecutar cualquier aplicación Angular, aunque sea de forma local. Al descargar Node.js, también se descargará **npm**, gestor de paquetes que se usará en el paso número 6.
2. Instalar **git** desde su página oficial: <https://git-scm.com/> . Git es un sistema de control de versiones necesario para descargar el proyecto.
3. Crear directorio donde almacenar la aplicación.
4. Abrir terminal en el directorio y ejecutar:

**git clone <https://github.com/guillermotti/webcmmse.git>**

Este comando de Git descargará el código alojado en el repositorio GitHub del enlace.

5. Acceder al directorio **webcmmse-app**. Una vez descargado el código, es necesario acceder a la carpeta que contiene toda la aplicación.
6. Ejecutar desde el terminal: **npm install**  
Este comando instalará todas las dependencias (librerías y tecnologías) de la aplicación.
7. Despues ejecutar: **ng serve -o**  
Este comando de Angular CLI, compilará y abrirá la aplicación.

Todos los pasos anteriores requieren ser ejecutados de forma secuencial. Cada paso necesita que el anterior haya terminado para poder ser efectuado.

## **7.3 Manual de sustitución de la base de datos**

Este proyecto funciona con una base de datos ya definida. No necesita ser instalada una nueva base de datos para el uso de la aplicación ya que todos los datos se encuentran alojados en Google Cloud Platform de manera distribuida y estos datos son accesibles desde cualquier dispositivo con conexión a internet. La aplicación en producción tiene una base de datos diferente, y no puede ser usada por otro organismo que no sea el CMMSE.

Para sustituir la base de datos que tiene por defecto la aplicación cuando se descarga, es necesario seguir los siguientes pasos:

1. Tener una cuenta de Google.
2. Acceder a la siguiente dirección: <https://console.firebaseio.google.com/>
3. En el panel hacer clic en “Añadir proyecto”. Rellenar los datos necesarios.
4. Acceder al proyecto creado.
5. Hacer clic en el botón que indica la siguiente figura:

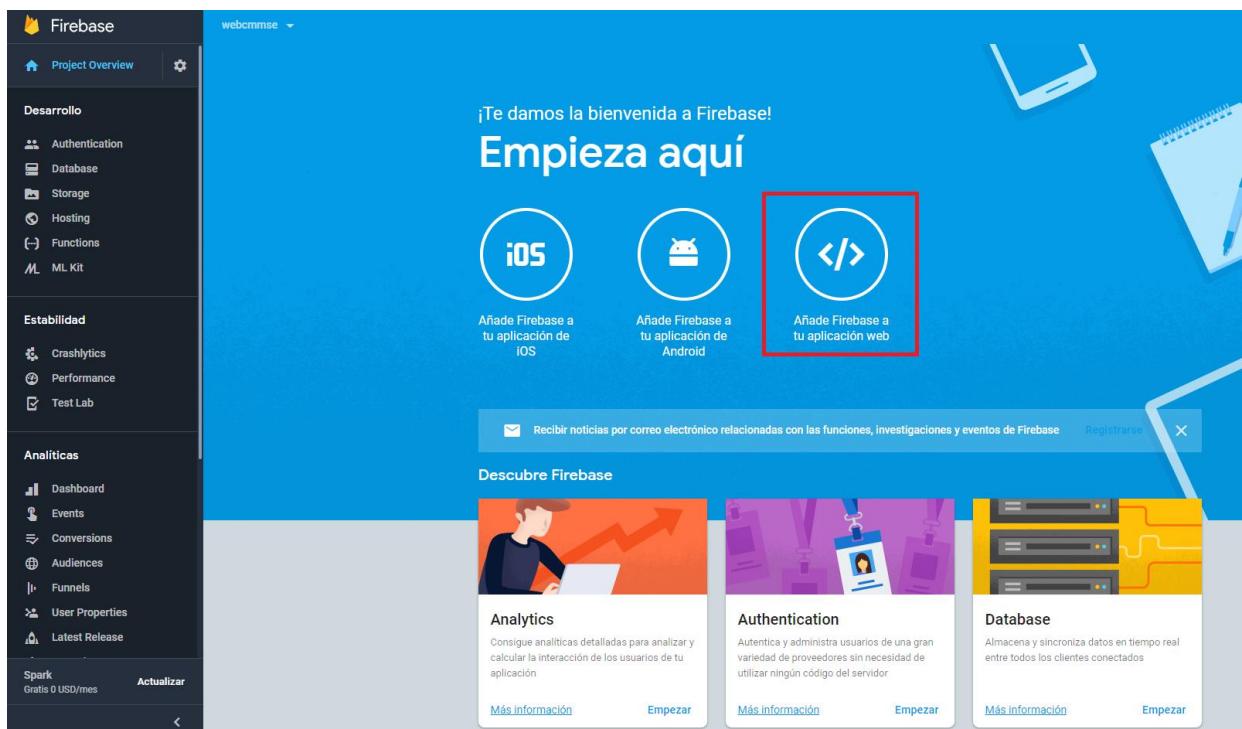


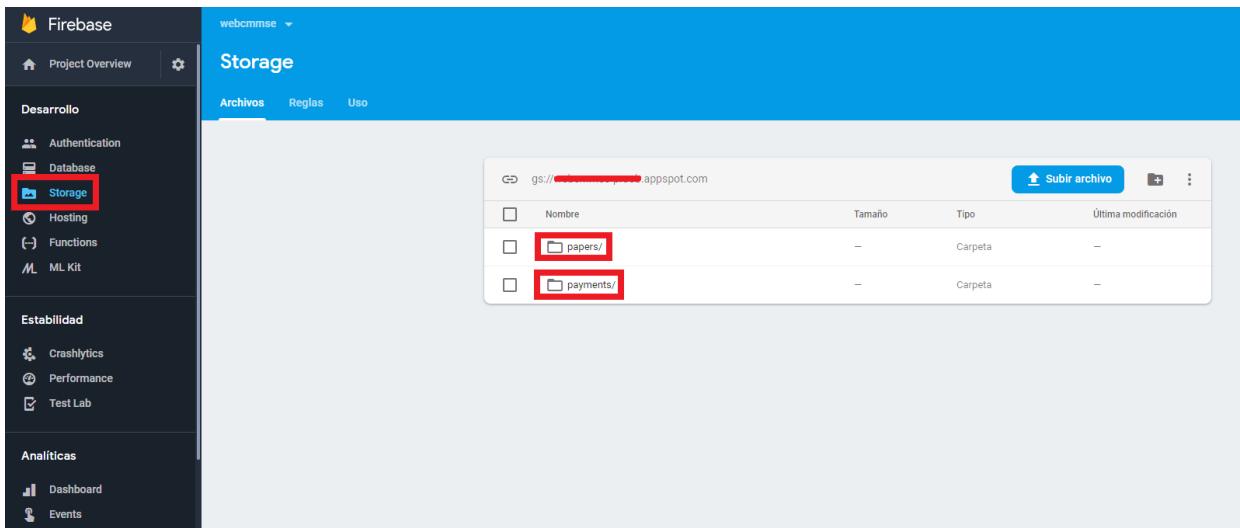
Figura 43: Panel del proyecto de Firebase

6. Añadir a los archivos **environment.ts** y **environment.prod.ts** de la aplicación, la configuración que aparece en la ventana modal del botón anterior de la siguiente forma:

```
TS environment.ts X TS environment.prod.ts
1 // The file contents for the current environment will overwrite these during build.
2 // The build system defaults to the dev environment which uses `environment.ts`, but if you do
3 // `ng build --env=prod` then `environment.prod.ts` will be used instead.
4 // The list of which env maps to which file can be found in `.angular-cli.json`.
5
6 export const environment = {
7   production: false,
8   firebase: {
9     apiKey: '████████████████████████████████',
10    authDomain: '████████.firebaseapp.com',
11    databaseURL: 'https://████████.firebaseio.com',
12    projectId: '████████',
13    storageBucket: '████████.appspot.com',
14    messagingSenderId: '████████'
15  }
16};
```

Figura 44: Configuración del proyecto de Firebase

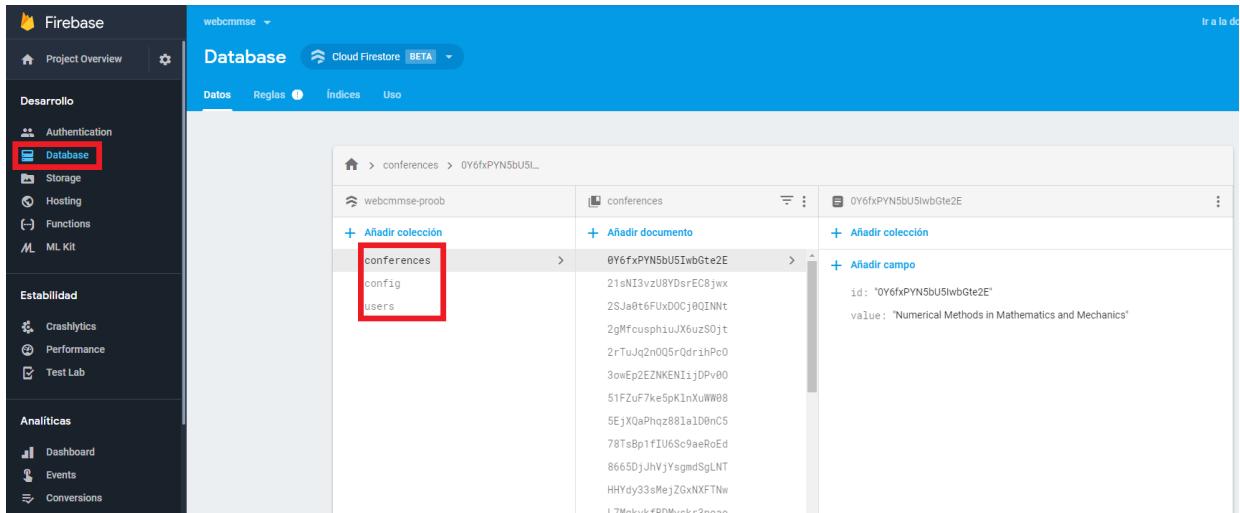
7. En el apartado de “Storage” de Firebase añadir las carpetas como indica la siguiente figura:



The screenshot shows the Firebase Storage interface. On the left, the navigation sidebar is visible with the 'Storage' option selected. The main area is titled 'Storage' and contains a table with two entries. The first entry is a folder named 'papers/' and the second is a folder named 'payments/'. Both entries have a status of 'Carpeta' (Folder) and a timestamp of '—'. A red box highlights the 'papers/' folder.

*Figura 45: Panel de gestión de Storage*

8. En el apartado de “Database” de Firebase añadir las colecciones como indica la siguiente figura:



The screenshot shows the Firebase Database interface. On the left, the navigation sidebar is visible with the 'Database' option selected. The main area is titled 'Database' and shows a tree structure under the 'conferences' node. The 'conferences' node has two children: 'config' and 'users'. A red box highlights the 'config' child node. To the right, there is a list of document IDs under the 'config' node, such as '0Y6fxPYN5bU5IwbGte2E', each with a preview of its value.

*Figura 46: Panel de gestión de Database – Colección Conferences*

9. Añadir en la colección “config” los campos que se ven en la siguiente figura:

The screenshot shows the Firebase Database interface for a project named 'webcmmse'. The left sidebar includes sections for Desarrollo (Authentication, Database, Storage, Hosting, Functions, ML Kit), Estabilidad (Crashlytics, Performance, Test Lab), and Analíticas (Dashboard, Events, Conversions, Audiences). The main area displays the 'Database' section with 'Cloud Firestore' as the selected provider. Under 'Database', tabs for Datos, Reglas, Índices, and Uso are available. The 'Datos' tab is selected, showing a tree structure for collections: 'conferences' and 'users'. A specific document under 'config' is selected, with its ID 'SsifDVbgsG60IpM05Uo9' highlighted by a red box. The document's content is displayed in a code-like format:

```

cmmse_opened: true
conference_url: "http://cmmse.usal.es/"
conference_year: "2018"
email_opened: false
id: "SsifDVbgsG60IpM05Uo9"
root_password: "U2FsdGVkX19D0E19p5LFZlL03yyTispg8F0BxE2on9ls"
root_user: "root"
send_term_opened: true

```

Figura 47: Panel de gestión de Database – Colección Config

Cabe destacar que el id del campo debe ser igual al id del documento. Además, el campo ***root\_password*** está encriptado. Para acceder a la aplicación como usuario administrador se deberá usar como usuario: ***root***, y como contraseña: ***cmmse***.

10. Las colecciones “conferences” y “users” no necesitan ningún documento para que la aplicación funcione, pero la colección “config” sí que necesita esos campos en el documento para que el sistema pueda usarse.

## 7.4 Manuales de despliegue

En este apartado se explicarán paso a paso dos formas para desplegar la aplicación en un entorno o servidor accesible por la red. En primer lugar, se darán los detalles necesarios para desplegar la aplicación en un Hosting de Firebase. Después, se explicará cómo desplegar la aplicación en un servidor web usando un Hosting de pago, con un dominio personalizado.

## 7.4.1 Manual de despliegue en Firebase

Para desplegar la aplicación en un Hosting de Firebase es necesario seguir los siguientes pasos en orden, esperando a que termine cada uno de ellos antes de empezar el siguiente paso:

1. Instalar las herramientas de Firebase con el comando: ***npm install -g firebase-tools***
2. Iniciar sesión en Google con el comando: ***firebase login***
3. Iniciar el proyecto con el comando: ***firebase init***
4. Indicar las opciones por defecto del comando anterior.
5. Ejecutar en consola el siguiente comando: ***ng build --prod --aot=false***
6. Copiar los archivos generados en el directorio **dist** al directorio **public** (creado al hacer el paso 3).
7. Ejecutar el comando: ***firebase deploy***

Al acabar el último paso se verá en la consola un enlace que dirige al sitio donde estará alojada la aplicación. En la siguiente figura se puede ver cómo es el panel de Hosting de Firebase:

Estado	Hora	Implementación	Archivos
★ Actual	4 jul. 2018 19:23	webcmmse@gmail.com Zla4D7	36
↑ Implementado	27 jun. 2018 0:56	webcmmse@gmail.com ox99_E	36
↑ Implementado	27 jun. 2018 0:51	webcmmse@gmail.com n5FgSV	36
↑ Implementado	27 jun. 2018 0:42	webcmmse@gmail.com aiHwSVu	36
↑ Implementado	23 jun. 2018	webcmmse@gmail.com	48

Figura 48: Panel de gestión de Hosting

En el panel de gestión de Hosting de Firebase se puede conectar un dominio personalizado. Además, se puede ver un historial de despliegues.

## 7.4.2 Manual de despliegue en servidor web

Para desplegar la aplicación en servidor web se deberá contratar una tarifa de Hosting y un dominio. Los pasos a seguir para poder desplegar la aplicación son los siguientes:

1. Ejecutar desde el directorio de la aplicación el comando: `ng build --prod --aot=false`
2. Subir los archivos generados en la carpeta dist a la carpeta **public\_html** del servidor web.
3. La aplicación será accesible en el dominio contratado.

## 7.5 Manual de usuario

En este apartado se detallará el manual de usuario. Se trata de un manual que indicará el funcionamiento de la aplicación mediante figuras y acciones. El enlace a la aplicación desplegada es:

- <https://cmmse-app.firebaseio.com>

### 7.5.1 Pantalla de inicio

La pantalla de inicio se muestra en la siguiente figura:



## Welcome to Online Registration System

You are about to begin the registration process. It involves three steps and you can do at different times. You will be asked to provide the following data:



### Step 1: Personal Information

Personal information (Full Name, email, ...) of the person who would be attending the conference.



### Step 2: Conference Information

If registering as an Author, Name of the conference, Title(s) of the Paper(s) and other information.



### Step 3: Payment Information

To upload the copy of the Proof of payment and, if it is necessary, require the Invoice.

[Register now](#)

[Log In](#)

[Return to Conferences](#)

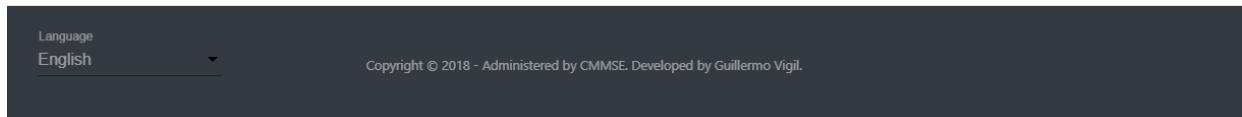


Figura 49: Pantalla de inicio

Esta pantalla muestra la información general del congreso. El pie de página es el mismo para todas las demás pantallas de la aplicación. En él se encuentra el selector de idioma. Haciendo clic en el selector se puede cambiar el idioma en toda la aplicación.

En la parte izquierda de la cabecera se muestra el logo y el título con el año del CMMSE. Esta parte también se muestra en todas las pantallas de la aplicación. En la parte derecha de la aplicación se encuentran tres enlaces: el primero de ellos dirige a esta misma parte, el segundo dirige a la página independiente de esta aplicación que contiene toda la información relativa al CMMSE y el tercero dirige a la pantalla de inicio de sesión.

En el cuerpo de la pantalla hay tres botones diferentes. El botón azul dirige a la pantalla de registro de usuario que se verá más adelante. El botón verde dirige a la pantalla de inicio de sesión. El botón blanco dirige a la pantalla del CMMSE como el enlace de la cabecera.

## 7.5.2 Pantalla de registro

La pantalla de registro se muestra en la siguiente figura:

The screenshot shows the 'Registration' page of the CMMSE 2018 website. At the top, there is a header with the CMMSE 2018 logo and links for Home, About CMMSE, and Log in. The main section is titled 'Registration' and contains two tabs: 'Personal information' and 'Login information'. Under 'Personal information', there are fields for Title, First name, Last name, Country, State, City, Address, Postal code, University, college or company, and Telephone. Under 'Login information', there are fields for Email, Confirm email, Password, and Confirm password. A checkbox for 'Accordance terms' is present, with a 'See details' link below it. Below the login fields, error messages are displayed: 'Required fields are empty' and 'Use at least eight characters with a combination of letters, numbers and symbols'. At the bottom, there are 'Submit' and 'Back' buttons. The footer includes language selection (English) and copyright information: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

*Figura 50: Pantalla de registro*

Esta pantalla permite a cualquier usuario ser registrado en la aplicación. Consiste en un conjunto de campos, algunos de ellos obligatorios, que el usuario debe llenar para poder registrarse en la aplicación. Una vez que se llenan todos los campos obligatorios, se podrá hacer clic en el botón “Submit” que realizará el registro. Mientras el botón esté deshabilitado, se mostrará en rojo aquella información que impide realizar el registro satisfactoriamente.

El botón “Back” permite volver a la pantalla de inicio. Haciendo clic en el enlace “See details” se podrá ver en una ventana modal la información de los términos de acuerdo y aceptación en la operación de registro.

Si el registro es satisfactorio, la aplicación redirigirá al usuario a la pantalla de inicio de sesión y el usuario y el sistema recibirán un correo notificando el nuevo usuario registrado.

### **7.5.3 Pantalla de inicio de sesión**

La pantalla de inicio de sesión se muestra en la siguiente figura:

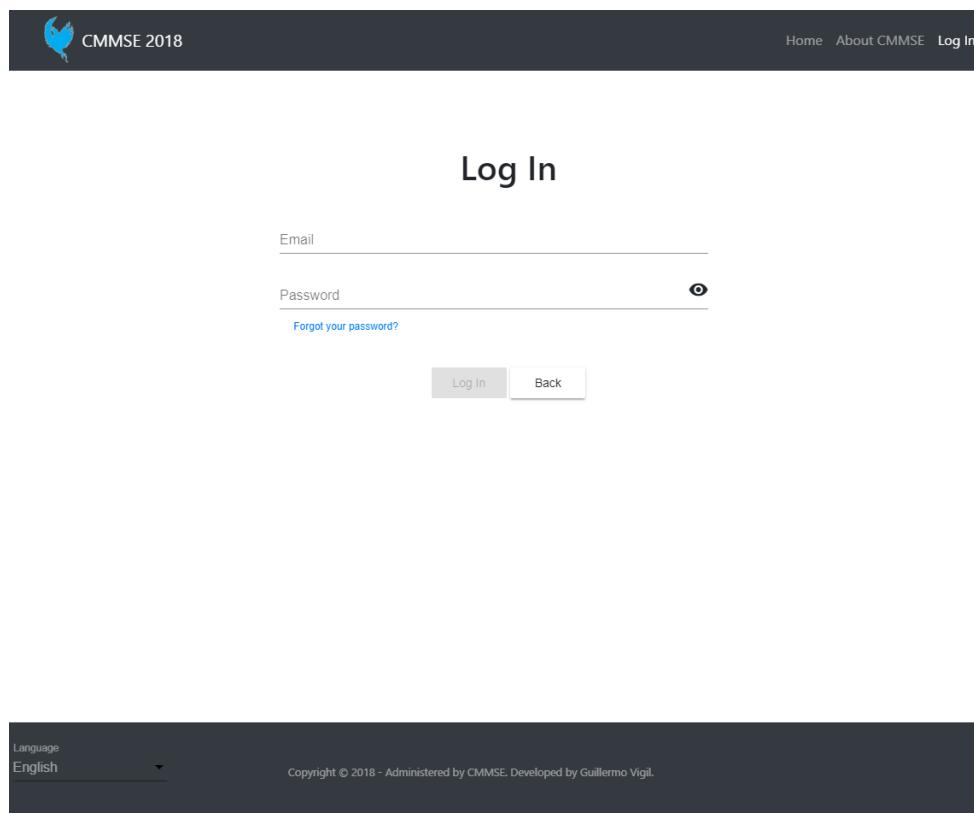


Figura 51: Pantalla de inicio de sesión

En esta pantalla el usuario podrá iniciar sesión introduciendo sus datos de usuario. Cuando estos datos se hayan introducido, el botón de “Log In” se habilitará y el usuario podrá hacer clic en él para iniciar sesión.

Si el usuario no existe, aparecerá un mensaje rojo en la pantalla que notificará lo ocurrido. Pasará lo mismo si la contraseña es incorrecta. Si el usuario introduce correctamente sus datos de usuario será redirigido a la pantalla de perfil de usuario.

El botón “Back” dirigirá al usuario a la pantalla de inicio. El enlace “Forgot your password?” dirigirá al usuario a la pantalla de recuperación de contraseña.

## 7.5.4 Pantalla de recuperación de contraseña

La pantalla de recuperación de contraseña se muestra en la siguiente figura:

The screenshot shows a dark-themed web page for 'CMMSE 2018'. At the top left is a blue bird logo and the text 'CMMSE 2018'. At the top right are links for 'Home', 'About CMMSE', and 'Log In'. The main title 'Recover password' is centered above a text input field labeled 'Email \*'. Below the input field are two buttons: 'Recover' (disabled) and 'Back'. At the bottom left is a language selection dropdown set to 'English'. At the bottom center is a copyright notice: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

*Figura 52:* Pantalla de recuperación de contraseña

Esta pantalla permite al usuario recuperar su contraseña. Para ello debe introducir su correo en el campo que aparece en pantalla. Será entonces cuando el usuario pueda hacer clic en el botón “Recover”. Este botón enviará un correo al usuario con su nueva contraseña.

El botón “Recover” no estará habilitado hasta que se rellene el campo que hay en pantalla. Si el correo introducido es incorrecto, aparecerá un mensaje en rojo indicándolo después de hacer clic en el botón.

## 7.5.5 Pantalla de perfil de usuario

La pantalla de perfil de usuario se muestra en la siguiente figura:

The screenshot shows the 'Personal information' section of a user profile. At the top, there is a navigation bar with the logo 'CMMSE 2018', links for 'Profile', 'Payments - Invoices', 'Paper', 'Documents', and the user's email 'guillermovigilr@gmail.com'. A 'Log Out' button is also present. Below the navigation bar, the 'Personal information' section is titled 'Personal information'. It contains fields for 'Title' (Dr.), 'First name' (Guillermo), 'Last name' (Vigil), 'Country' (Spain), 'State' (dropdown menu), 'City' (dropdown menu), 'Address' (dropdown menu), 'Postal code' (dropdown menu), 'University, college or company' (Uniovi), and 'Telephone' (534623625). At the bottom of this section are two blue buttons: 'Change data' and 'Change password'. At the very bottom of the page, there is a footer with a language selection dropdown set to 'English' and a copyright notice: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

Figura 53: Pantalla de perfil de usuario

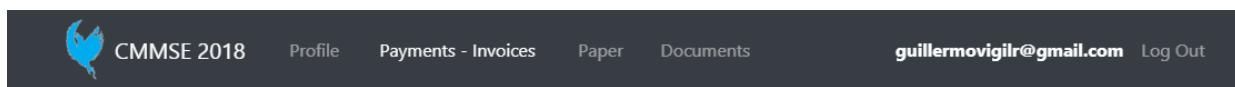
Esta pantalla muestra los datos de perfil del usuario que ha iniciado sesión. La cabecera permite navegar entre las pantallas del usuario. Además, permite cerrar la sesión en el enlace “Log Out”.

Los datos de perfil no son editables, pero si se hace clic en el botón “Change data”, los campos se habilitan, pueden ser modificados y el botón pasa a llamarse “Save changes”. Después de modificar los datos, el usuario podrá guardar las modificaciones haciendo clic en este botón de nuevo.

El botón “Change password” abrirá una ventana modal desde la que el usuario podrá cambiar su contraseña introduciendo previamente la actual, para luego introducir la nueva con doble factor de seguridad.

## **7.5.6 Pantalla de pagos y facturas**

La pantalla de pagos y facturas se muestra en la siguiente figura:



## Payments

You can modify your proof of payment (uploading a new file) until payment has been confirmed. Upload a proof of payment:

Select file

## Invoices

**IMPORTANT.** If you want invoice, fill below fields:

Title \* Dr.

First name \* Guillermo  Last name \* Vigil

University, college or company \* Rodriguez  CIF/VAT Number/TAX ID \* 46234623

Address \* FHskjfhs

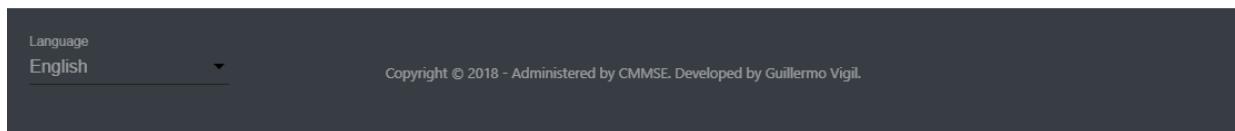


Figura 54: Pantalla de pagos y facturas

En esta pantalla el usuario podrá subir un justificante de pago en la sección “Payments”. Haciendo clic en el botón “Select file” el usuario podrá seleccionar un archivo de forma local para subirlo a la base de datos. Este archivo podrá ser eliminado haciendo clic en la “x” a la derecha del título del archivo. Además, el archivo podrá ser visualizado en el enlace con el nombre del archivo.

Se permite subir una nueva versión del archivo haciendo clic de nuevo en el botón “Select file”. Esto sobreescibirá el archivo anterior.

En la sección “Invoices” el usuario podrá añadir los datos que quiera para la factura. Una vez que se hayan rellenado todos los campos, el botón “Submit” se habilitará. El usuario podrá hacer clic en el botón para guardar los datos relevantes a la factura.

### **7.5.7 Pantalla de subida de artículos**

La pantalla de subida de artículos se muestra en la siguiente figura:

The screenshot shows a user interface for managing conference papers. At the top, there's a navigation bar with the logo 'CMMSE 2018', links for 'Profile', 'Payments - Invoices', 'Paper', 'Documents', and a user email 'guillermovigil@gmail.com' along with a 'Log Out' button.

The main area is titled 'My papers'. It includes a 'Filter...' input field and a table listing six papers:

Special Session & Minisymposiums	Title	File	Status	Actions
Numerical Methods in Mathematics and Mechanics	Numerical		UPLOADED	
Numerical Methods in Mathematics and Mechanics	Methods		UNDER REVIEW	
Numerical Methods in Mathematics and Mechanics	Mates		MAJOR/ MINOR	
Numerical Methods in Mathematics and Mechanics	Prueba Error 2		ACCEPTED	
Numerical Methods in Mathematics and Mechanics	Prueba Error 3		UPLOADED	

Below the table are pagination controls: 'Items per page: 5' (with a dropdown arrow), '1 - 6 of 11', and navigation arrows. There's also a 'New paper' button.

The 'New paper' section has a dropdown menu 'Special Session & Minisymposiums' set to 'Special Session & Minisymposiums'. It includes fields for 'Title \*' (with placeholder 'Poster'), file upload buttons ('Paper', 'Select file', 'No selected file'), author information ('First name \*', 'Last name \*', 'Email \*', 'Add author', 'Remove author'), and a 'Submit' button.

At the bottom, there's a footer with language selection ('Language English'), copyright notice ('Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'), and a 'Submit' button.

Figura 55: Pantalla de subida de artículos

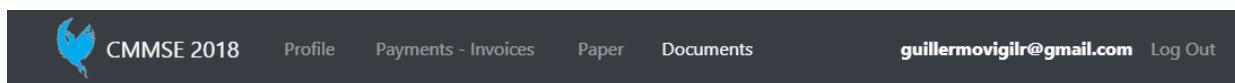
Esta pantalla permite al usuario subir un artículo para presentarlo en el congreso. La primera sección “My papers” permanece oculta si el usuario no ha subido aún ningún artículo. Cuando el usuario haya subido un artículo, este aparecerá en una fila de la tabla.

Se podrá editar la información del artículo dependiendo del estado en el que se encuentre, para ello se abrirá una ventana modal con todos los datos. También se podrá eliminar el artículo. Además, se podrá visualizar el archivo subido en la columna “File” de la tabla.

En la sección “New paper” el usuario deberá llenar todos los campos obligatorios para que el botón “Submit” se habilite. Cuando esté habilitado se podrá subir el archivo y todos sus datos se guardarán en la base de datos. El botón “Select file” sirve para escoger el archivo de forma local y los botones “Add author” y “Remove author” permitirán al usuario añadir autores del artículo. Como mínimo habrá un autor y como máximo diez autores.

### **7.5.8 Pantalla de documentos**

La pantalla de inicio se muestra en la siguiente figura:



## Documents

**IMPORTANT.** To prevent fraud, you just could download each file ONCE. After first download, you could not download the file again.

### Invoice file

You already download the file, you can not download it again.

[Download](#)

### Attendance certificates

You already download the file, you can not download it again.

[Download](#)

### Presentation certificates

You already download the file, you can not download it again.

[Download](#)

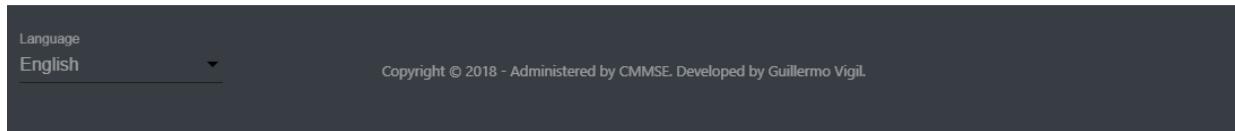


Figura 56: Pantalla de documentos

Esta pantalla muestra los documentos que el usuario podrá descargar. Los botones permitirán realizar la descarga de cada documento de forma independiente. Estos documentos solo podrán ser descargados una única vez, para evitar fraudes fiscales.

Para que estos documentos sean visibles por el usuario, el usuario administrador debe haber validado el justificante de pago, subido por el usuario previamente.

Los documentos que pueden ser descargados son: factura, certificado de asistencia al congreso y certificado de ponencia de todos los artículos que estén en estado aceptado.

## 7.5.9 Pantalla de usuarios – rol administrador

La pantalla de usuarios del usuario administrador se muestra en la siguiente figura:

The screenshot shows the 'Users' section of the CMMSE 2018 web application. At the top, there is a navigation bar with the logo 'CMMSE 2018', links for 'Users', 'Payments', 'Authors', 'Papers', 'Conferences', 'Documents', and 'Configuration', and a log out link for 'root'. Below the navigation bar, the title 'Users' is displayed next to a user icon. A search bar labeled 'Filter...' is present. The main content area is a table listing eight users:

First name	Last name	Email	University, college or company	Country	Actions
Guillermo	Vigil	gvigilrodrig@dxc.com	Uni Pak	Pakistan	
Inés	García	igarciapalacio@gmail.com	Uniovi	Spain	
Guille	Vigil	guillevr_92@hotmail.com	Morocco Uni	Morocco	
Juan	Fernández	guillermottt@icloud.com	Camb Uni	Cambodia	
Guillermo	Vigil	guillermovigilr@gmail.com	Uniovi	Spain	
Jose Manuel	Redondo	redondojose@uniovi.es	University of Oviedo	Spain	
jose	ranilla	ranilla@uniovi.es	oviedo	Spain	

At the bottom of the table, there are pagination controls: 'Items per page: 25' with a dropdown arrow, '1 - 7 of 7', and navigation arrows.

At the very bottom of the page, there is a footer with a language selection dropdown set to 'English' and the copyright notice 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

*Figura 57: Pantalla de usuarios – rol administrador*

Esta pantalla reúne en una tabla todos los usuarios registrados en el sistema. Desde la propia tabla es posible editar el usuario, donde se abrirá una ventana modal para cambiar los datos del usuario. También es posible eliminar al usuario.

Encima de la tabla, el usuario podrá filtrar una búsqueda de los datos de la tabla. Esto se aplica a todas las tablas de la aplicación.

### 7.5.10 Pantalla de pagos – rol administrador

La pantalla de pagos del usuario administrador se muestra en la siguiente figura:

The screenshot shows a web application interface for managing payments. At the top, there is a dark header bar with the logo 'CMMSE 2018', navigation links for 'Users', 'Payments' (which is highlighted in blue), 'Authors', 'Papers', 'Conferences', 'Documents', and 'Configuration', and user information 'root Log Out'. Below the header is a search bar labeled 'Filter...'. The main content area has a title '€ Payments' with a subtitle 'Payments'. A table lists payment records for several users. The columns are: 'Full name', 'Email', 'Country', 'Payment file', 'To paid', 'Paid', and 'Invoice file'. The 'Paid' column contains checkboxes, some of which are checked (indicated by a red checkmark). The 'Invoice file' column contains download icons. At the bottom of the table, there are pagination controls for 'Items per page' (set to 25), '1 - 7 of 7', and navigation arrows. At the very bottom of the page, there is a footer with a language selection dropdown set to 'English' and a copyright notice: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

Full name	Email	Country	Payment file	To paid	Paid	Invoice file
Guillermo Vigil	gvigilrodrig@dxc.com	Pakistan		450 €	<input type="checkbox"/>	<a href="#">Download</a>
Inés García	igarciapalacio@gmail.com	Spain		450 €	<input type="checkbox"/>	<a href="#">Download</a>
Guille Vigil	guillevr_92@hotmail.com	Morocco		450 €	<input type="checkbox"/>	<a href="#">Download</a>
Juan Fernández	guillermotti@icloud.com	Cambodia		450 €	<input type="checkbox"/>	<a href="#">Download</a>
Guillermo Vigil	guillermovigil@gmail.com	Spain	<a href="#">Edit</a>	450 €	<input checked="" type="checkbox"/>	<a href="#">Download</a>
Jose Manuel Redondo	redondojose@uniovi.es	Spain		200 €	<input type="checkbox"/>	<a href="#">Download</a>
jose ranilla	ranilla@uniovi.es	Spain	<a href="#">Edit</a>	450 €	<input checked="" type="checkbox"/>	<a href="#">Download</a>

*Figura 58: Pantalla de pagos – rol administrador*

Esta pantalla reúne en una tabla todos los usuarios registrados en el sistema. Desde la propia tabla es posible abrir el justificante de pago subido por el usuario, cambiar la tarifa, validar el pago y descargar la factura de cada usuario. Para validar el pago, aparecerá una ventana modal que confirme la acción, ya que esta validación permitirá al usuario descargar sus documentos desde su sesión.

### 7.5.11 Pantalla de autores – rol administrador

La pantalla de autores del usuario administrador se muestra en la siguiente figura:

First name	Last name	University, college or company	Country	Papers	Posters
Guille	Vigil	Morocco Uni	Morocco	1	1
Guillermo	Vigil	Uniovi	Spain	11	3
Jose Manuel	Redondo	University of Oviedo	Spain	1	0
jose	ranilla	oviedo	Spain	1	1

Figura 59: Pantalla de autores – rol administrador

En esta pantalla se muestra una tabla con los usuarios que han subido algún artículo. En ella no se podrá realizar ninguna acción; se trata de una pantalla informativa, por lo que solo se podrá filtrar una búsqueda entre los autores de la tabla.

## 7.5.12 Pantalla de artículos – rol administrador

La pantalla de artículos del usuario administrador se muestra en la siguiente figura:

The screenshot shows a web application interface for managing conference papers. At the top, there is a dark header bar with the logo 'CMMSE 2018' and navigation links: Users, Payments, Authors, Papers, Conferences, Documents, Configuration. On the right side of the header, it shows the user 'root' and a 'Log Out' link. Below the header, the page title is 'Papers' with a document icon. A search bar labeled 'Filter...' is present. The main content is a table listing eight papers. The columns are: Title, Conference title, Corresponding author, Paper, Status, and Actions. Each row contains a small preview thumbnail of the paper, the title, the conference title, the email of the corresponding author, the status (e.g., UPLOADED, UNDER REVIEW, MAJOR/ MINOR, ACCEPTED), and two small icons for edit and delete. At the bottom of the table, there is a language selection dropdown set to 'English' and a copyright notice: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

Title	Conference title	Corresponding author	Paper	Status	Actions
It works fine	Numerical Methods in Mathematics and Mechanics	juan@gmail.com		UPLOADED	
Numerical	Numerical Methods in Mathematics and Mechanics	guille@gmail.com		UPLOADED	
Methods	Numerical Methods in Mathematics and Mechanics	guille@gmail.com		UNDER REVIEW	
Mates	Numerical Methods in Mathematics and Mechanics	gui@gmail.com		MAJOR/ MINOR	
Prueba Error 2	Numerical Methods in Mathematics and Mechanics	juan@gmail.com		ACCEPTED	
Prueba Error 3	Numerical Methods in Mathematics and Mechanics	juan@gmail.com		UPLOADED	
Prueba Error 4	Numerical Methods in Mathematics and Mechanics	pica@gmail.com		UPLOADED	
Prueba Error 5	Numerical Methods in Mathematics and Mechanics	ana@gmail.com		UPLOADED	

Figura 60: Pantalla de artículos – rol administrador

En esta pantalla se muestran todos los artículos subidos al sistema. En la tabla se puede visualizar el artículo, cambiar el estado del artículo, editar los datos relativos al artículo y eliminar el artículo si es necesario.

Si se cambia el estado de un artículo el usuario y el sistema recibirán un correo indicando el cambio de estado del artículo seleccionado.

### **7.5.13 Pantalla de conferencias – rol administrador**

La pantalla de conferencias del usuario administrador se muestra en la siguiente figura:

The screenshot shows the 'Conferences' section of the CMMSE 2018 web application. At the top, there is a navigation bar with links for Users, Payments, Authors, Papers, Conferences, Documents, Configuration, and a user session indicator (root Log Out). Below the navigation bar, the title 'Conferences' is displayed next to a speech bubble icon. A search bar labeled 'Filter...' is present. The main content area contains a table with conference data:

Title	Papers	Posters	Actions
Numerical Methods in Mathematics and Mechanics	14	5	

Below the table, there are pagination controls: 'Items per page: 25' with a dropdown arrow, '1 - 1 of 1', and navigation arrows. At the bottom of the page, there is a 'New conference' button, a 'Title' input field, and an 'Add conference' button. A footer bar at the very bottom includes a language selection dropdown set to 'English', a copyright notice 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.', and a dark background.

*Figura 61: Pantalla de conferencias – rol administrador*

Desde esta pantalla se pueden visualizar todas las conferencias disponibles para el congreso. Se pueden editar las existentes y eliminar cualquiera de ellas.

Además, se podrá añadir una nueva conferencia rellenando el campo “Title” y haciendo clic en el botón “Add conference”. Estas conferencias son elegidas a la hora de subir un artículo y el usuario deberá seleccionar una de ellas para poder subir un artículo nuevo a la aplicación.

### 7.5.14 Pantalla de documentos – rol administrador

La pantalla de documentos del usuario administrador se muestra en la siguiente figura:

The screenshot shows the 'Documents' section of the CMMSE 2018 web application. At the top, there is a navigation bar with links for 'Users', 'Payments', 'Authors', 'Papers', 'Conferences', 'Documents', 'Configuration', 'root', and 'Log Out'. Below the navigation bar, the title 'Documents' is displayed next to a document icon. There are two main sections: 'PDF Documents' and 'Excel Documents'. Under 'PDF Documents', there are three buttons: 'Attendance certificates' (red), 'Presentation certificates' (red), and 'Invoices' (white). Under 'Excel Documents', there are three buttons: 'Users' (green), 'Papers' (green), and 'Authors' (green). Below these sections is a form for creating an 'Id badge'. It includes fields for 'Size \*' (dropdown menu showing 'Red color'), 'Red color' (text input '255'), 'Green color' (text input '255'), 'Blue color' (text input '255'), and 'Text for the badge' (text input 'Text for the badge'). A button labeled 'Id badge' is located below the form. At the bottom of the page, there is a footer with a 'Language' dropdown set to 'English' and a copyright notice: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

Figura 62: Pantalla de documentos – rol administrador

Esta pantalla permite al usuario administrador descargar una serie de documentos relacionados con el congreso:

- El botón “Attendance certificates” descargará en formato PDF todos los certificados de asistencia para todos los usuarios.
- El botón “Presentation certificates” descargará en formato PDF todos los certificados de ponencia para todos los artículos que se encuentren en estado aceptado.
- El botón “Invoices” descargará en formato PDF todas las facturas de todos los usuarios registrados en el sistema.
- El botón “Users” descargará en formato Excel todos los usuarios del sistema.
- El botón “Papers” descargará en formato Excel todos los artículos subidos a la aplicación.
- El botón “Authors” descargará en formato Excel todos los autores de la aplicación.
- El botón “Id badge” descargará en formato PDF las tarjetas identificativas para todos los usuarios. Para ello hay que seleccionar un tamaño en el campo “Size”. Si se rellena el campo “Text for the badge”, se descargará una única hoja con el texto puesto en el campo dentro de las etiquetas. Además, es posible cambiar el color de las tarjetas usando los campos de color rojo, verde y azul. En una escala RGB de 0 a 255.

### **7.5.15 Pantalla de configuración – rol administrador**

La pantalla de configuración del usuario administrador se muestra en la siguiente figura:

The screenshot shows the 'Configuration' section of the CMMSE 2018 web application. At the top, there is a navigation bar with links for 'Users', 'Payments', 'Authors', 'Papers', 'Conferences', 'Documents', 'Configuration', 'root', and 'Log Out'. The main content area is titled 'Configuration' with a wrench icon.

**CMMSE information**

The public access to the page is OPEN (checked). The delivery period is OPEN (checked). Root user is 'root'. Root password is masked. Conference year is 2018. Conference initial day is 7/9/2018. Conference last day is 7/14/2018. Conference place is 'Rota, Cádiz - Spain'. Conference URL is 'http://cmmse.usal.es/'. Conference CIF is 'G-53827147'. ISBN is '978-84-897-7881-6'. Fee to pay is 450 €. Certificate signature date is 7/9/2018. Spain invoice number is 202. Invoice number for other places is 199.

**Email configuration**

The email service is OPEN (checked). Email sender is 'webcmmse@gmail.com'. Email sender password is masked. There is a note about emails on carbon copy (BCC).

**Danger zone**

Enable buttons (checked). Buttons include 'Reset users' and 'Reset conferences'. A 'Change data' button is also present.

Language is set to English. Copyright notice at the bottom states: 'Copyright © 2018 - Administered by CMMSE. Developed by Guillermo Vigil.'

Figura 63: Pantalla de configuración – rol administrador

Desde esta pantalla se puede gestionar toda la configuración de la aplicación. Para modificar los datos relativos a la configuración es necesario hacer clic en el botón “Change data”, entonces se habilitarán los campos y el usuario administrador podrá modificar lo que desee. Haciendo clic en el botón “Save changes” se guardará la configuración y se volverán a deshabilitar los campos de la pantalla.

En la sección “Danger zone” se pueden habilitar dos botones que permiten limpiar la base de datos. Para habilitar estos botones es necesario hacer clic en el toggle “Enable buttons”. Después de habilitar los botones, el usuario administrador podrá hacer clic en el botón “Reset users” que borrará todos los usuarios registrados en el sistema. El botón “Reset conferences” hará lo mismo con las conferencias guardadas en el sistema.