



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA  
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

---

GESTOR Y LOCALIZADOR DE  
OPOSICIONES EN CASTILLA Y LEÓN

---

Alumno: Daniel Fernández Jiménez

Tutor/es/as: Joaquín Adiego Rodríguez y Natalia Martín Cruz





*A mi familia.*



# Agradecimientos

A mi familia, siempre me ha ayudado en los momentos difíciles y no me han dejado tirar la toalla.

A mis amigos y amigas, que siempre me han apoyado y ayudado a crecer tanto personalmente como académicamente. Sin muchos de ellos no podría haber conseguido acabar este largo camino.

En especial mención a mi hermano mayor, sin él no podría haber descubierto lo que iba a ser mi futuro laboral. A Javier, amigo que he tenido desde que entré en la carrera, cuyo apoyo y ayuda ha sido siempre incondicional.

**Muchas gracias a todos.**



# Resumen

El objetivo de este proyecto es desarrollar una página web de consulta de las oposiciones vigentes en el territorio de Castilla y León.

Esta página pretende que los usuarios encuentren con facilidad las ofertas y convocatorias de empleo público que les son de interés, sin necesidad de estar pendientes del Boletín Oficial de la Comunidad. Para ello, se han gestionado los documentos de las oposiciones por etapas y desarrollado una serie de filtros que facilitan lo que el usuario quiere encontrar.

El proyecto se ha desarrollado con el framework Vue.js, HTML5, CSS, y Javascript. Se ha elaborado siguiendo la metodología ágil Scrum.





# Abstract

The aim of this project is to develop a website to consult the current civil service examinations in the territory of Castilla y León.

This page aims to make it easy for users to find the public employment offers and announcements that are of interest to them, without the need to keep an eye on the Official Gazette of the Autonomous Community. For this purpose, the documents of the public examinations have been managed by stages and a series of filters have been developed to facilitate what the user wants to find.

This project has been developed using Vue.js framework, HTML5, CSS and Javascript. It has been developed following the agile Scrum's methodology.



# Índice general

<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VII</b>
<b>Lista de figuras</b>	<b>XV</b>
<b>Lista de tablas</b>	<b>XVII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Motivación . . . . .	1
1.3. Objetivos de proyecto . . . . .	2
1.3.1. Objetivos de desarrollo . . . . .	2
1.3.2. Objetivos personales . . . . .	2
1.4. Estructura de la memoria . . . . .	2
<b>2. Estado del arte</b>	<b>5</b>
2.1. Boletín Oficial de Castilla y Leon . . . . .	5
2.1.1. Regulación en BOCyL . . . . .	5
2.2. Oposiciones y concurso . . . . .	6
2.3. Categorización . . . . .	7

<b>3. Tecnologías utilizadas</b>	<b>11</b>
3.1. Vue.js . . . . .	11
3.1.1. Ciclo de vida . . . . .	12
3.2. NPM . . . . .	14
3.2.1. Router . . . . .	14
3.2.2. Vuetify . . . . .	14
3.2.3. Axios . . . . .	14
3.2.4. iso-8859-2 . . . . .	14
3.2.5. xml-js . . . . .	15
3.3. Firebase . . . . .	15
3.4. Google Cloud Platform . . . . .	16
3.5. Regex101 . . . . .	16
3.6. Git . . . . .	16
3.6.1. Github . . . . .	17
3.7. Trello . . . . .	17
3.8. LaTeX . . . . .	17
3.8.1. Overleaf . . . . .	17
3.9. Astah . . . . .	17
<b>4. Planificación y desarrollo del proyecto</b>	<b>19</b>
4.1. Scrum . . . . .	19
4.1.1. Introducción . . . . .	19
4.1.2. Equipo Scrum . . . . .	20
4.1.3. Eventos Scrum . . . . .	20
4.1.4. Artefactos Scrum . . . . .	21
4.2. Planificación. . . . .	22
4.3. Tareas realizadas . . . . .	23

4.3.1. Sprint 0 . . . . .	25
4.3.2. Sprint 1 . . . . .	26
4.3.3. Sprint 2 . . . . .	27
4.3.4. Sprint 3 . . . . .	28
4.3.5. Sprint 4 . . . . .	29
4.3.6. Sprint 5 . . . . .	30
4.3.7. Sprint 6 . . . . .	31
4.3.8. Sprint 7 . . . . .	32
4.3.9. Sprint 8 . . . . .	33
4.3.10. Sprint 9 . . . . .	34
4.4. Modificación de planificación . . . . .	35
4.4.1. Sprint 10 . . . . .	36
4.4.2. Sprint 11 . . . . .	36
4.4.3. Sprint 12 . . . . .	37
4.4.4. Sprint 13 . . . . .	39
4.4.5. Sprint 14 . . . . .	40
<b>5. Plan de riesgos y estimación de costes</b>	<b>43</b>
5.1. Plan de riesgos . . . . .	43
5.1.1. Riesgos encontrados en este proyecto . . . . .	44
5.2. Presupuesto . . . . .	48
<b>6. Análisis del proyecto</b>	<b>49</b>
6.1. Introducción . . . . .	49
6.2. Requisitos funcionales . . . . .	49
6.3. Requisitos no funcionales . . . . .	49
6.4. Casos de uso . . . . .	49
6.5. Diagramas de secuencia . . . . .	50

<b>7. Diseño</b>	<b>55</b>
7.1. BaaS . . . . .	55
7.1.1. Ventajas . . . . .	56
7.1.2. Desventajas . . . . .	56
7.2. SaaS . . . . .	56
7.2.1. Ventajas . . . . .	57
7.2.2. Desventajas . . . . .	57
7.3. Modelo-Vista-Modelo de vista . . . . .	57
7.3.1. Ventajas . . . . .	59
7.4. Diseño guiado por componentes . . . . .	60
7.4.1. Ventajas . . . . .	60
7.4.2. Desventajas . . . . .	60
7.4.3. Componentes creados en este proyecto . . . . .	60
7.5. Front-End . . . . .	61
7.6. Back-End . . . . .	65
7.6.1. Functions . . . . .	65
7.6.2. Schedule Function . . . . .	66
7.6.3. Métodos . . . . .	67
7.6.4. Modelo de despliegue de la web . . . . .	68
<b>8. Implementación</b>	<b>71</b>
8.1. Estructura del código del proyecto . . . . .	71
8.2. Decisiones a lo largo del proyecto . . . . .	73
8.3. Cambios realizados a lo largo del proyecto. . . . .	74
8.4. Licencia . . . . .	74
<b>9. Conclusiones y futuro</b>	<b>75</b>
9.1. Conclusiones . . . . .	75

9.2. Trabajo futuro . . . . .	76
<b>A. Manual de usuario</b>	<b>77</b>
<b>B. Manual de despliegue</b>	<b>91</b>
B.1. Herramientas necesarias . . . . .	91
B.2. Preparación del entorno y despliegue . . . . .	91
<b>Bibliografía</b>	<b>93</b>





# Índice de figuras

2.1. Tipos de oposiciones según estudios . . . . .	8
2.2. Fases y plazos de una oposición. . . . .	9
3.1. Ciclo de vida de Vue . . . . .	13
6.1. Diagrama de casos de uso . . . . .	51
6.2. Secuencia seleccionando fase . . . . .	51
6.3. Secuencia aplicando filtros . . . . .	53
7.1. Arquitectura Baas . . . . .	55
7.2. Arquitectura MVVM . . . . .	58
7.3. MVVM en Vue . . . . .	58
7.4. Diagrama de componentes . . . . .	61
7.5. Captura de página web (1) . . . . .	62
7.6. Captura de página web (2) . . . . .	63
7.7. Captura de página web (3) . . . . .	63
7.8. Captura de página web (4) . . . . .	64
7.9. Vista principal final . . . . .	65
7.10. Vista de información de un documento final . . . . .	65
7.11. Modelo de despliegue de página web . . . . .	69

A.1. Vista principal de la página web . . . . .	77
A.2. Últimos documentos publicados por el BOCyL . . . . .	78
A.3. Todos los documentos que se encuentran en la fase aprobado . . . . .	79
A.4. Todos los documentos que se encuentran en la fase abierto . . . . .	80
A.5. Todos los documentos que se encuentran en la fase provisional . . . . .	80
A.6. Todos los documentos que se encuentran en la fase definitivo . . . . .	81
A.7. Todos los documentos que se encuentran en la fase modificaciones . . . . .	82
A.8. Documentos filtrados por localidad en fase recientes . . . . .	82
A.9. Todos los documentos que se encuentran en la fase aprobado filtrando por fecha de disposición . . . . .	83
A.10. Todos los documentos que se encuentran en la fase abierto filtrando por rango	83
A.11. Todos los documentos que se encuentran en la fase provisional filtrando por organismo . . . . .	84
A.12. Todos los documentos que se encuentran en la fase definitivo filtrando por fecha de publicación . . . . .	85
A.13. Todos los documentos que se encuentran en la fase modificaciones filtrando por número de disposición (orden) . . . . .	86
A.14. Todos los documentos que se encuentran en la fase abierto filtrando por pala- bras clave y localidad . . . . .	86
A.15. Resultado de buscar documento utilizando filtros localidad, fecha de publica- ción y organismo . . . . .	87
A.16. Enlace PDF del resultado de filtrar por localidad, fecha de publicación y or- ganismo . . . . .	88
A.17. Resultado de búsqueda utilizando filtros localidad, fecha de publicación y pa- labra clave . . . . .	89
A.18. Documento encontrado con filtros utilizados pero resultado no deseado . . . .	89
A.19. Documento encontrado con filtros utilizados y resultado satisfactorio . . . . .	90

# Índice de cuadros

4.1. Planificación inicial de sprints . . . . .	23
4.2. Detalle de las historias de usuario . . . . .	24
4.3. Tareas del sprint 0 . . . . .	26
4.4. Tareas del sprint 1 . . . . .	27
4.5. Tareas del sprint 2 . . . . .	28
4.6. Tareas del sprint 3 . . . . .	29
4.7. Tareas del sprint 4 . . . . .	30
4.8. Tareas del sprint 5 . . . . .	31
4.9. Tareas del sprint 6 . . . . .	32
4.10. Tareas del sprint 7 . . . . .	33
4.11. Tareas del sprint 8 . . . . .	34
4.12. Tareas del sprint 9 . . . . .	35
4.13. Planificación final de sprints . . . . .	35
4.14. Tareas del sprint 10 . . . . .	36
4.15. Tareas del sprint 11 . . . . .	37
4.16. Tareas del sprint 12 . . . . .	39
4.17. Tareas del sprint 13 . . . . .	40
4.18. Tareas del sprint 14 . . . . .	41
5.1. Riesgo de falta de formación . . . . .	44

5.2. Riesgo de enfermedad . . . . .	44
5.3. Riesgo de retraso de las tareas . . . . .	45
5.4. Riesgo de ausencia del <i>Scrum Master</i> . . . . .	45
5.5. Riesgo de modificación de funcionalidades . . . . .	45
5.6. Riesgo de ausencia de información . . . . .	46
5.7. Riesgo de cambio en las tecnologías utilizadas . . . . .	46
5.8. Riesgo de uso de Firebase . . . . .	46
5.9. Riesgo de planificación optimista . . . . .	47
5.10. Riesgo de API no operativa . . . . .	47
5.11. Riesgo de fallo en Schedule Function . . . . .	47
6.1. Requisitos funcionales . . . . .	50
6.2. Requisitos no funcionales . . . . .	50

# Capítulo 1

## Introducción

### 1.1. Introducción

Buscar oposiciones es una tarea sencilla, simplemente hay que buscar en el boletín oficial de Castilla y León (BOCyL) y dedicarle un tiempo a la lectura después de haber entendido cómo funciona dicha página web. Sin embargo, a pesar de su sencillez, es una tarea tediosa y podemos estar bastante tiempo buscando una oposición que nos interese y no encontrarla. Muchas webs muestran de una forma u otra las diferentes oposiciones y exponen diferentes filtros para encontrarlas, pero en ocasiones éstos no son útiles o bien no son suficiente. Por otro lado, algunas se centran en una determinada oposición y dejan de lado que el usuario esté interesado en otra oposición que sea compatible con la que busca. Estos dilemas se han intentado resolver a la hora de realizar este proyecto.

### 1.2. Motivación

La principal motivación para llevar a cabo este proyecto ha sido mi interés por aprender más acerca del desarrollo web. Quería conocer cómo funcionan tanto Back-End como Front-End, ya que ambas me llaman la atención, y desearía que mi futuro laboral estuviera dirigido a trabajar en esto.

La idea del proyecto, a pesar de parecer a priori sencilla, tiene una importante parte de investigación que es lo que hace que se diferencie de otros buscadores de oposiciones ya existentes [11]. También hay otras páginas interesantes en este campo, como son el buscador de empleo de la administración del gobierno [2]. Estas alternativas ofrecen datos a nivel nacional, pero sus interfaces no son tan sencillas, y los campos para filtrar tienen carencias o son insuficientes, por ello decidí desarrollar una página web que facilitara a los usuarios el proceso de encontrar las oposiciones que buscan.

## 1.3. Objetivos de proyecto

### 1.3.1. Objetivos de desarrollo

El principal objetivo de desarrollo de este proyecto se basa en crear una herramienta que se actualice diariamente para obtener los últimos datos sobre oposiciones en Castilla y León. Con ella, los usuarios podrán buscar y encontrar con total facilidad la información que les sea necesaria. Para ello, se utilizarán una serie de filtros que ayudarán a gestionar y localizar dichas oposiciones.

Además, con el fin de facilitar la búsqueda a los usuarios lo máximo posible, se han establecido una serie de filtros que clasifican las fases por las que pasan las oposiciones: aprobada, abierta, provisional, definitiva y modificaciones.

De esta forma, alguien que esté a la espera de que se abra el plazo de su oposición podrá dirigirse al apartado "abierto", y una vez realice su solicitud, podrá consultar si ha sido o no admitido en el apartado "provisional", y posteriormente en "definitivo".

### 1.3.2. Objetivos personales

Debido a que este proyecto se desarrolla en un contexto educativo y sirve como Trabajo de Fin de Grado, existen también una serie de objetivos enfocados a la formación académica y personal, los cuales son:

- Conocer en profundidad el framework Vue.js.
- Aumentar mis habilidades en el desarrollo web. *Front-end* utilizando HTML, CSS y JavaScript. *Back-End*.
- Ampliar mis conocimientos de base de datos utilizando Firebase.
- Mejorar mi capacidad y destreza a la hora de diseñar y desarrollar la estructura de una página web.
- Entender y aprender la metodología de Scrum, herramienta útil que cada vez se vuelve más popular debido a su facilidad de entendimiento y flexibilidad al emplear diferentes procesos y técnicas.

## 1.4. Estructura de la memoria

El resto del contenido de la memoria esta estructurado de la siguiente manera:

- **Capítulo 2:** se explica brevemente el funcionamiento, las fases y las divisiones que puede tener una oposición (organismo, titulación, etc).

- **Capítulo 3:** se explica brevemente cada tecnología utilizada a lo largo del proyecto.
- **Capítulo 4:** se explica el desarrollo y planificación seguida en el proyecto. Metodología empleada y tareas realizadas a lo largo del proyecto.
- **Capítulo 5:** se explican los riesgos que pueden surgir y el presupuesto del proyecto.
- **Capítulos 6, 7 y 8:** se explica el análisis, diseño e implementación del proyecto.
- **Capítulo 9:** se finaliza la memoria con unas conclusiones y posibles mejoras que podrían ser implementadas.
- **Apéndices:** se presentan dos apéndices, el manual de usuario y el manual de despliegue.





## Capítulo 2

# Estado del arte

### 2.1. Boletín Oficial de Castilla y León

El Boletín Oficial de Castilla y León (BOCyL) [8] es el diario oficial de la comunidad autónoma de Castilla y León, a través del cual se da publicidad a los documentos que deben ser objeto de publicación oficial, de acuerdo con el ordenamiento jurídico.

Se publica diariamente de lunes a viernes, excepto los días declarados inhábiles en toda la comunidad autónoma de Castilla y León. Es editado por la Consejería de la Presidencia de la Junta de Castilla y León a través del Servicio del Boletín Oficial de la Comunidad, adscrito a la Direcciones General del Secretario de la Junta y de Relaciones con las Cortes.

#### 2.1.1. Regulación en BOCyL

Según el documento [9] publicado el 2 de octubre de 2017, DECRETO 30/2017, por el que se regula el Boletín Oficial de Castilla y León, los documentos publicados en el Boletín tendrán la consideración de oficiales y auténticos de acuerdo con lo previsto en dicho decreto. El acceso será universal y gratuito.

Se establece el contenido, estructura y edición.

En cuanto al contenido, se publican leyes, normas, disposiciones administrativas, actos emanados de instituciones, resoluciones de órganos, etc. En resumen, cualquier documento de carácter público o privado cuya publicación esté prevista por el ordenamiento jurídico. En este proyecto solo interesa el contenido acerca de las oposiciones, que se engloba en el apartado “Oposiciones y concurso”.

Sobre la estructura de esta parte, esta se divide en:

- **Comunidad de Castilla y León**

- Estado y otras Comunidades Autónomas
- Administración Local
- Administración de Justicia
- Otros anuncios.

Como el proyecto se enfoca en las oposiciones de Castilla y León, los datos necesarios pertenecerán únicamente al apartado “Oposiciones y concursos” de la Comunidad de Castilla y León y de su Administración Local. Para conseguir estos datos usando un único filtro, se ha conseguido una API [10]. Una vez recopilados estos, se seguiría con el trabajo de investigación sobre cómo funcionan las oposiciones y el análisis de los objetos obtenidos a través de la API.

En cuanto a la edición, cada documento del BOCyL irá precedido de un resumen de su contenido. Todos los documentos que se publiquen figurarán numerados y estarán identificados con un código de verificación que permitirá comprobar su autenticidad.

Por último, es necesario hablar sobre la publicación en esta entidad (BOCyL). Este proyecto se describe como un proyecto de desarrollo en vertical, es decir, el porcentaje funcionalidad-investigación es de 30-70, ya que la idea desde un principio era que mostrara las oposiciones que estuvieran aprobadas y abiertas.

Por ello, un gran porcentaje de la investigación se empleó en comprobar que las oposiciones que se encontraran en el BOCyL fueran las mismas que existen en cada Boletín Oficial de la Provincia (BOP). El resultado fue el esperado: la mayoría de ellas estaban en el Boletín Oficial de Castilla y León. Se investigó enero, febrero, marzo de 2021 y algunos meses de 2020. El porcentaje de los documentos que no coincidían, es decir, que aparecían en el Boletín Oficial de la Provincia y no en el Boletín Oficial de Castilla y León era extremadamente bajo, por lo que se prefirió tener una sola fuente (el BOCyL), para facilitar las peticiones HTTP.

## 2.2. Oposiciones y concurso

Dentro de las oposiciones publicadas en el Boletín Oficial de Castilla y León (BOCyL) existen diferentes procesos selectivos. Estos son de dos tipos: oposición y concurso-oposición.

- **Oposición:** es un proceso de selección para acceder a los puestos de trabajo de la Administración pública. Este proceso selectivo está basado en los principios de igualdad, mérito y capacidad. Generalmente, este proceso conlleva una o varias pruebas y exámenes que tienen la finalidad de evaluar la competencia de los candidatos para acceder al puesto.

A su vez, dentro del proceso de selección mediante oposición existen dos variantes: oposición libre y de promoción interna.

- *Oposición libre:* conocida como de turno libre, es una forma de acceso a una convocatoria de empleo público que permite a cualquier persona que cumpla los requisitos establecidos en las bases, presentarse a la misma.

- *Oposición promoción interna*: es una forma de acceso a una convocatoria de empleo público en la que son necesarios los mismos requisitos de la oposición libre y uno más: ser funcionario de la Administración con al menos dos años de antigüedad. La finalidad de este proceso selectivo es que los funcionarios de distintos cuerpos y escalas puedan promocionarse en su trabajo, es decir, puedan escalar a otro grupo de titulación inmediatamente superior (promoción interna vertical).
- **Concurso-oposición**: es la forma de acceso a una convocatoria que consiste en dos fases: la presentación y valoración de méritos propios (fase de concurso) y la realización de pruebas exigidas en la fase de oposición. Un vez realizadas, se suman las puntuaciones obtenidas en ambas fases y se realiza un listado con ellas.
- *Concurso*: es el proceso selectivo basado en la valoración de méritos del aspirante, según el baremo de puntos publicado en la convocatoria. Los aspirantes no podrán alcanzar más de 10 puntos, repartiéndose éstos en: experiencia previa, formación académica y otros méritos.

Todos estos procesos selectivos constan de las mismas etapas/fases, que se explican en la sección 2.3.

## 2.3. Categorización

Una de las primeras tareas de este proyecto (y una de las más tediosas, cabe destacar) fue categorizar las oposiciones. Se dedicó mucho tiempo a investigar. Se planteó categorizar en función de la titulación académica exigida, del organismo perteneciente a la oposición, del puesto de trabajo ofrecido, etc. Finalmente se dedujo que la forma más adecuada de acercarnos al objetivo deseado era categorizar dependiendo de la etapa en la que se encontrara la oposición.

Antes de explicar por qué se implemento la solución que se detalla en esta sección, es necesario informar de los diferentes títulos [38] (ver figura 2.1 ) y organismos que se pueden encontrar en el BOCyL, ya que sirven para poder filtrar en la página web. En función de la titulación, las oposiciones se dividen en:

- **Grupo A1**: Título de Licenciatura Universitaria,
- **Grupo A2**: Título de Diplomatura Universitaria
- **Grupo B**: Titulación de Técnico Superior (FP Grado Superior)
- **Grupo C1**: Título de Bachiller o Técnico (también estudios equivalentes de leyes anteriores, hasta 3º de BUP, FPII, bachillerato LOGSE, o COU)
- **Grupo C2**: Titulación de Grado en Educación Secundaria Obligatoria o equivalente.
- **Grupo E**: No exige ninguna titulación.

### 2.3. CATEGORIZACIÓN

La siguiente imagen resume lo recientemente explicado:

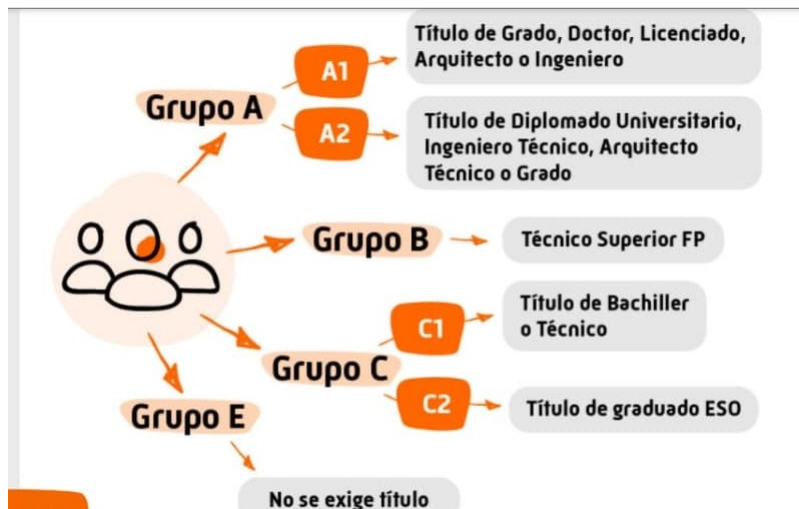


Figura 2.1: Tipos de oposiciones según estudios

Por otro lado, nos encontramos con diferentes organismos:

- **Consejería.** Dentro de éste podemos encontrar las consejerías de sanidad, de la presidencia, de educación, de economía y hacienda, de fomento y medio ambiente, de cultura y turismo, de agricultura, ganadería y desarrollo rural, de empleo e industria, de familia e igualdad de oportunidades, y de cada una de las provincias.
- **Universidad.** Dentro de éste podemos encontrar de cada una de las provincias.
- **Ayuntamiento.** Dentro de éste podemos encontrar de cada una de las provincias y de pueblos de las provincias.
- **Diputación provincial.** Dentro de éste podemos encontrar de cada una de las provincias.
- **Cortes.** Dentro de éste podemos encontrar únicamente de Castilla y León.
- **Consejo.** Dentro de éste podemos encontrar el consejo de cuentas de Castilla y León y el consultivo de Castilla y León.
- **Mancomunidad de municipios.** Dentro de éste podemos encontrar de cada uno de los municipios.

En primera instancia, se quería categorizar o realizar un filtro que abarcara la mayor parte del trabajo tedioso. Su función consistía en gestionar las oposiciones dependiendo de la titulación (licenciatura universitaria, bachiller, educación secundaria obligatoria, etc.). Esta

opción no se pudo realizar debido a este dato solo se detalla cuando se oferta el puesto de trabajo citado (cuando se aprueba) y no cuando se abre convocatoria, es decir, no siempre aparece este dato.

Otra opción que se planteó fue categorizar dependiendo del área del puesto de trabajo ofrecido (educación, justicia, sanidad, etc.). Esta opción era inviable debido a que en la mayoría de los documentos no se hace una referencia clara (es difícil de encontrar a simple vista) o no viene declarado.

Como consecuencia de esto, se decidió gestionar las oposiciones en función a sus etapas, y estas vienen detalladas en la imagen 2.2. Este proyecto como se dijo anteriormente, se basa en su mayor parte en investigación, y la realización de funcionalidades es menor. Por ello, se podría deducir que es un proyecto con mayor desarrollo vertical que horizontal.

FASES Y PLAZOS EN UNA OPOSICIÓN	
La Oferta Pública de Empleo (OPE)	En general se aprueban anualmente por los Ayuntamientos y se publican en el BOP
¿Cuál es la diferencia entre OPE y la convocatoria de la oposición?	<ul style="list-style-type: none"> <li>- <b>Oferta pública de empleo</b> es el anuncio que hace la administración de las futuras plazas que se van a ofertar de empleo público</li> <li>- <b>La convocatoria</b> es el acto de llevarlas a cabo</li> </ul>
PLAZO entre oferta y convocatoria de la oposición.	<b>PLAZO MÁXIMO de 3 años</b> desde que se publica la oferta de empleo público hasta que se constituye la convocatoria de las oposiciones
Bases de convocatoria de la oposición	Tras la publicación de la oferta, el Ayuntamiento aprueba las bases de la convocatoria de la oposición, especificando el número de plazas convocadas, el sistema selectivo escogido, los derechos de examen y el temario, entre muchas otras variables así como que el resto de publicaciones será en el Tablón de Anuncios del Ayuntamiento.
PLAZO presentación de instancias en la oposición	Este plazo comienza tras la publicación de la convocatoria en el BOE y el plazo de presentación de instancias son <b>20 días hábiles o naturales</b> a elección de la administración; estos empiezan a contabilizar un día después de que salga publicada la oposición.
Abono de las tasas de la oposición	Todos los candidatos deben abonar unas tasas por derechos de examen en la oposición, cuyo justificante deberá adjuntar a la instancia de la convocatoria
Lista provisional de admitidos y excluidos en la oposición	Tras los 20 días hábiles o naturales, la administración debe publicar la lista provisional de los aspirantes admitidos y excluidos en la oposición y el plazo máximo es de <b>1 mes</b> desde el día siguiente a la terminación del plazo de admisión de solicitudes
Reclamaciones a la lista provisional de admitidos y excluidos	<b>Plazo para subsanar</b> la solicitud de la oposición; este normalmente, consta de <b>10 días hábiles</b> desde la publicación en el Tablón de Anuncios. La subsanación puede ser por error material al rellenar la solicitud, ausencia del resguardo de haber abonado las tasas, etc.
Lista definitiva de admitidos y excluidos en la oposición	Tras resolver las alegaciones presentadas, se publicara la lista definitiva ( <b>sin plazo</b> ) .
Fecha del primer ejercicio	La lista definitiva de admitidos y excluidos determinara el día y la hora del primer ejercicio y en las bases debe especificar el tiempo máximo que debe transcurrir entre la lista definitiva mencionada y el primer ejercicio. La casuística nos da plazos como <b>5 días hábiles</b> , por ejemplo.
Plazo entre ejercicios	<b>No existe plazo prefijado</b> y en las bases debe especificar el tiempo mínimo y máximo que debe transcurrir entre el primer ejercicio y el segundo. La casuística nos da plazos como un <b>mínimo de 5 días hábiles</b> , y un <b>máximo de 45 días hábiles</b> .
Plazos de alegaciones	<b>Tras cada ejercicio</b> , la administración puede dar un plazo de alegaciones, que no tendrá naturaleza de recurso, que debe estar especificado en las bases y que la casuística nos da de <b>5 días hábiles</b> , por ejemplo.
Publicación de la relación de aspirantes que han superado el proceso selectivo	<b>Sin plazo</b> desde la realización del último ejercicio. La lista se publica en el Tablón de anuncios del Ayuntamiento.
Acreditación de las condiciones exigidas en las bases	Desde la publicación de la lista de aprobados anterior, en <b>20 días hábiles</b>
Lista definitiva de aspirantes que han superado el proceso selectivo	Se publica en el BOP y los aprobados tiene <b>5 días hábiles</b> desde la publicación para tomar posesión

Figura 2.2: Fases y plazos de una oposición.

Como se puede ver el flujo natural de una oposición es:

APROBADO→ABIERTO→PROVISIONAL→DEFINITIVO ——— MODIFICACIONES

### 2.3. CATEGORIZACIÓN

---

La etapa modificaciones es opcional, ya que no todos los procesos las sufren. Se decidió realizar de esta manera, debido a que al obtener los documentos procedentes del BOCyL se observó que no existe un campo identificativo concreto para cada oposición, sino que cada una tiene diferentes documentos relacionados con ella, puede identificarse de diferentes maneras: por la localidad, el organismo o utilizando palabras claves.

De esta forma, el usuario puede ver de manera muy simple si la oposición que busca está aprobada, abierta, si ha sufrido una modificación, etc. Además, podría consultar también los resultados provisionales y definitivos de la misma.

Cabe destacar que la mayoría de las oposiciones son publicadas en el Boletín Oficial de la provincia correspondiente y en el Boletín Oficial de Castilla y León. Sin embargo, después de publicarse la convocatoria que hace referencia a una oposición en concreto, se abre el plazo para su solicitud y sus resultados no siempre se publican en el BOCyL. No obstante, tanto las modificaciones como las etapas de aprobado y abierto, sí suelen aparecer en ambos boletines.

## Capítulo 3

# Tecnologías utilizadas

### 3.1. Vue.js

Vue.js [41] es un framework progresivo Javascript para crear interfaces de usuario. En los últimos años, se ha utilizado como desarrollo en *Front-end* para crear aplicaciones de tipo SPA (Single Page Application) [35]. Las aplicaciones de este tipo se caracterizan por caber en una sola página con el propósito de dar una experiencia más fluida a los usuarios, es decir, todas las vistas de la aplicación estarán recogidas en una misma página. El punto de entrada a ellas es el fichero `index.html`. Por lo que no habrá necesidad de cargar el navegador para cambiar de vistas.

Este framework ofrece un conjunto de características y funcionalidades que permiten crear webs completas. Las características por las que destaca son las siguientes:

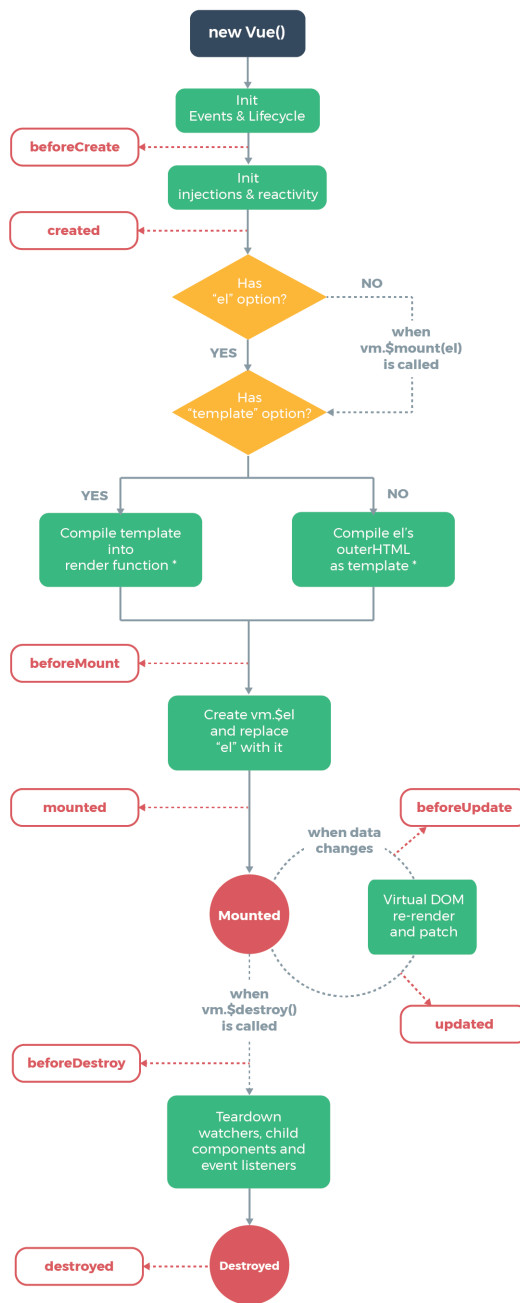
- **Modularidad.** Se tiene más libertad para usar ciertas características, es decir, tendrás que instalar el paquete que prefieras para realizar la funcionalidad que desees implementar.
- **Reactividad.** Vue.js tiene propiedades reactivas, quiere decir que si cambia una variable en una parte de la vista de la página, Vue actualizará su valor sin necesidad de que lo hagas manualmente. Esta es una de las características de más importancia y ayuda a que sea menos complejo realizar cambios en las vistas.
- **Componentes Web.** Vue.js permite componetizar el *Front-end* de la página web. Está formado por numerosos archivos con extensión `.vue`, los cuales constituirán los componentes. Estos archivos están basados en la idea de Componentes de Un Solo Archivo (Single File Componentes). Se dividen en tres fragmentos, encapsulados cada uno de ellos entre etiquetas estilo XML.
  - **Template.** Código HTML que crea los elementos visuales. Se vincula con el DOM virtual, característica que se explicará posteriormente.

- **Script.** Código Javascript o TypeScript (en este caso Javascript) que gestiona toda la lógica de los elementos y datos que mostrarán. Se divide en:
  1. name: Nombre del componente.
  2. components: Declaración de todos los componentes que va a usar el componente. Habrá una referencia a todos los componentes en el template en formato HTML. Se pueden pasar las propiedades del componente como atributos de cada uno de ellos.
  3. data: Función que almacena las variables del estado del componente. En el template se permitirá usar estas variables para mostrar datos.
  4. props: Conjunto de atributos que se pueden dar valor cuando este componente sea utilizado por otro.
  5. methods: Conjunto de métodos del componente. Pueden ser utilizados por eventos de elementos del template o por la propia lógica interna del componente.
  6. computed: Conjunto de variables que requieren de un cálculo previo en base a otras. Está relacionado con la reactividad. Si una de las variables de las que depende cambia, la variable que muestra el resultado cambiará.
  7. watch: Permite realizar acciones en base a los cambios de una propiedad de un elemento de la parte data o de una variable de la parte computed. Permite “observar” constantemente los cambios del estado de un componente y puede implementar reacciones ante dichos cambios.
- **Style.** Clases CSS con sus propiedades que permiten contener el código como tal o también pueden hacer referencia a ficheros externos que contendrán dicho código.
- **Virtual Dom.** Vue genera un DOM virtual haciendo uso de los elementos visuales declarados en el template. Por lo que si se produce un cambio en la vista, bien por cambios en los datos o variables, hará que se actualice el DOM virtual en consonancia. Esto se consigue con una propiedad llamada reactividad.
- **Eventos y transiciones.** Vue ofrece una variedad de maneras para aplicar eventos y efectos de transición cuando los elementos son insertados, actualizados o eliminados del DOM.

#### 3.1.1. Ciclo de vida

Una aplicación Vue está formada por componentes, es decir, instancias [21]. Pasan por una serie de pasos desde que se inicializan hasta que se destruyen, esto se conoce como ciclo de vida, representado en la figura 3.1. En la parte izquierda de la figura se puede ver por los diferentes métodos y etapas por las que pasa. Estos métodos pueden ser utilizados en su componente en la sección script.





\* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

Figura 3.1: Ciclo de vida de Vue

## 3.2. NPM

Node Packager Manager (NPM) [27] es el administrador de paquetes para la plataforma Node Javascript. Con estos paquetes podemos añadir dependencias, librerías y módulos a la aplicación en desarrollo. Se han añadido algunas dependencias más las que se añaden por defecto al utilizar Vue.

### 3.2.1. Router

Es el plugin enrutador oficial de Vue.js [40]. Es lo más común de utilizar en las aplicaciones SPA creadas por Vue.js pues facilita el paso entre rutas y vistas. En un archivo se declaran las rutas posibles y después utilizando comandos sencillos según lógica de la aplicación y del plugin en cuestión, se podrá cambiar fácilmente de vista.

### 3.2.2. Vuetify

Vuetify [42] es un framework de componentes que combina la potencia de Vue.js con la estética de Material Design. Permite añadir elementos visuales para las vistas de la aplicación. Permite acelerar el desarrollo de aplicaciones web complejas, ya que los componentes están "listos para usar". Es una gran herramienta debido a que dispone de una enorme librería, la cuál es la recomendada por Vue.js.

### 3.2.3. Axios

Se trata de una librería Javascript [4] que facilita las peticiones HTTP y es capaz de ejecutarse tanto en navegador como en NodeJS. Se pueden realizar solicitudes contra un servidor, completamente configurables, y recibir la respuesta de manera sencilla de procesar.

Está basado en promesas, por lo que se puede generar código asíncrono ordenado, e incluso se puede usar *Async/Await*. Está pensado para facilitar el consumo en servicios de API REST, es sencillo de utilizar y tiene muy poco peso (13 kilobytes o menos si el servidor lo envía comprimido). Con Axios se usa API de promesas, por lo que cuando se reciba la respuesta del servidor se ejecutará una función callback configurada (*then*) y cuando se produzca un error una función callback definida (*catch*).

### 3.2.4. iso-8859-2

Este paquete [20] es una implementación robusta de Javascript de la codificación de caracteres iso-8859-2 según lo definido por el estándar de codificación. Se pueden utilizar tres funciones:

- **iso88592.labels:** Una matriz de cadenas, cada una de las cuales representa una etiqueta para esta codificación.
- **iso88592.encode(input, options):** Esta función toma una cadena de texto sin formato (*input*) y la codifica de acuerdo con iso-8859-2. El valor de retorno es un retorno independiente *Uint16Array*. *Options* es un objeto opcional y junto a su propiedad *mode* se puede establecer el modo de error.
- **iso88592.decode(input, options):** Función que decodifica *input* según iso-8859-2. El parámetro *input* puede ser un *Uint16Array* o una cadena de bytes. *Options* hace lo mismo que la anterior función.

### 3.2.5. xml-js

El paquete `xml-js` [43] permite la conversión de texto XML en objeto Javascript/texto JSON y viceversa. La principal razón de creación de esta herramienta fue la posibilidad de obtener el objeto JSON compacto y no compacto. A modo de conclusión, compacto quiere decir que ocupa menos y se obtiene desordenado; Por el contrario, no compacto significa que ocupará más, pero vendrá ordenado.

Algunas otras características por las que también destaca es porque es totalmente compatible con XML, reversible, dependencias mínimas, cambiar propiedad propiedad nombre clave, admite recorrido ascendente... etc.

## 3.3. Firebase

Firebase [14] es una plataforma digital cuyo propietario es Google. Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea rápido, pero sin renunciar a la calidad requerida. Dispone de diferentes funcionalidades, que se pueden dividir en los siguientes grupos grupos: Desarrollo, crecimiento y monetización y analítica.

Solo se van a explicar aquellas que se hayan utilizado.

- **Firestore:** base de datos NoSQL en tiempo real de Firebase.
- **Hosting:** servicio que se encarga del alojamiento de ficheros que componen y despliegan una web.
- **Functions:** funciones almacenadas en al nube de Google [] y se ejecuta en un entorno administrado. Después de escribir e implementar la función con algún propósito específico, los servidores de Google comienzan a administrarla de inmediato. En este proyecto ha sido de vital importancia, pues se han utilizado *functions* para insertar cada mes los documentos de las oposiciones a través de la API y una *function schedule* que va a estar constantemente cada día a una determinada hora realizando peticiones

HTTP al Boletín Oficial de Castilla y León. Ésta última es la que nos va a permitir actualizar el proyecto cada día.

## 3.4. Google Cloud Platform

Google Cloud Platform [34] es una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado. Utilizada para disponer de diferentes tipos de soluciones a través de la tecnología almacenada en la nube. Destaca por su rapidez y escalabilidad.

Se han utilizado dos servicios: Google Cloud Scheduler y Google Cloud Functions.

- **Google Cloud Scheduler:** servicio relacionado con las *Functions scheduler* de Firebase que nos permite tener un mayor control sobre las funciones que hemos programado. Permite actualizarlas, editar, pausar, borrar, reanudar, ver registros más ampliamente y volver a ejecutar.
- **Google Cloud Functions:** servicio relacionado con las *Functions* de Firebase que nos permite tener un mayor control sobre estas funciones. Permite editar, borrar, copiar, ver registros, variables, detalles, fuente, invocaciones/segundo, tiempo de ejecución, entre otras.

Esta sección se ha utilizado con el objetivo de modificar el tiempo de espera, memoria asignada y observar más detenidamente los errores que hemos ido obteniendo. No se ha introducido como servicio Firebase, a pesar de que lo es, porque era necesaria una explicación más detallada del uso de dicha base de datos.

## 3.5. Regex101

Regex101 [31] es una herramienta online que sirve como depurador de expresiones regulares multilingüe gratuito con explicación en tiempo real, detección de errores y resaltado. Admite PCRE, JavaScript, GoLang y Java. Actualmente, se ha convertido en uno de los servicios de prueba de expresiones regulares más grandes del mundo.

## 3.6. Git

Git [17] es una herramienta de control de versiones de código abierto gratuita. Control de versiones es análogo a controlar y gestionar los cambios realizados a lo largo en el desarrollo de un proyecto. Git almacena estos cambios y la información relacionada en una estructura de datos o repositorio, cada uno de los cuales está formado por los archivos que lo constituyen.

En este proyecto se han utilizado diferentes herramientas procedentes de Git para mantener, organizar y actualizar el código fuente del proyecto.

### 3.6.1. Github

GitHub [18] se trata de una herramienta que permite gestionar los proyectos creados, tener un mayor control de versiones y ordenar el código en cada una de éstas (versiones). No requiere de pago, por lo que su uso ha sido totalmente gratuito.

## 3.7. Trello

Trello [39] es una herramienta que permite la creación, gestión y compartición de tableros en línea. En ellos se pueden crear listas y pueden contener dentro de ellas tarjetas más específicas. Las ventajas son muy variadas, versión gratuita, aplicación online, simple e intuitiva, sistema de notificaciones, etiquetado de colores, motor de búsquedas bueno y rápido, tableros privados o públicos, entre otras.

Se ha utilizado esta herramienta con el objetivo de controlar las tareas que hay que hacer, las que están en proceso y las que están hechos (En cada columna aparece *To Do*, *Doing*, *Done*). Como consecuencia, tendremos un mayor control sobre el proyecto de forma general

## 3.8. LaTeX

LaTeX [22] es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Actualmente es popularmente utilizado por artículos y libros científicos. Es un software libre y gratuito.

### 3.8.1. Overleaf

Overleaf [28] es una herramienta de publicación y redacción colaborativa en línea que hace que todo el proceso de redacción, edición y publicación de documentos sea mucho más rápido y sencillo. Se ha utilizado esta herramienta para realizar la memoria del proyecto.

## 3.9. Astah

Astah [29] es una herramienta de diseño de sistemas que soporta UML, diagrama de relación de entidades, diagramas de flujo, CRUD, diagrama de flujo de datos, etc. Se ha utilizado para hacer el diagrama de componentes, casos de uso y modelo de despliegue web.



## Capítulo 4

# Planificación y desarrollo del proyecto

### 4.1. Scrum

#### 4.1.1. Introducción

Scrum [33] es un marco de trabajo que promueve la colaboración en los equipos para lograr reducir el desarrollo de productos complejos. Es un proceso de desarrollo ágil que aplica un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto.

Scrum está basado por un proceso empírico el cual se divide en 3 tres pilares fundamentales:

- **Transparencia.** Los aspectos que más destacan en el proyecto deben ser conocidos por todos los miembros del equipo y por los interesados del resultado. Por ello, todo el proceso debe de seguir un estándar común que ayude al entendimiento entre las partes integrantes del proyecto e interesados.
- **Inspección.** Cada cierto tiempo se debe de evaluar la información del avance del producto, es decir, inspeccionar el estado del producto del proyecto para detectar si ha habido o van existir variaciones indeseadas de cara al futuro.
- **Adaptación.** En el caso de que existan variaciones indeseadas que desvíen al proyecto del resultado final, el proceso de desarrollo debe de ser adaptado de la manera más adecuada.

### 4.1.2. Equipo Scrum

El equipo Scrum está compuesto por un grupo de trabajadores que se reparten los roles. Cada uno de ellos tiene diferentes responsabilidades y debe de rendir cuentas de distinta manera, tanto entre ellos como para el resto de la organización. Los roles son:

- **Product Owner.** Es el encargado de optimizar y maximizar el valor del producto. Es el responsable de modificar el *Product Backlog*, artefacto que se explicará en la sección 4.1.4. Su objetivo es actualizar constantemente la lista de tareas del equipo de desarrollo para que el producto esté bien estructurado, detallado y priorizado.
- **Scrum Master.** Tiene dos funciones principales: gestionar el proceso Scrum y eliminar impedimentos que puedan afectar a la entrega del producto. Es la persona encargada de comprobar que el equipo cumple las reglas y principios de la metodología.
- **Equipo de desarrollo.** Son las personas encargadas de desarrollar el producto, auto-organizándose y auto-gestionándose para conseguir entregar cada incremento (ver sección 4.1.4). Se encargarán de crear un incremento terminado a partir de los elementos del *Product Backlog* seleccionados durante el *Sprint Planning* (ver sección 4.1.4)

### 4.1.3. Eventos Scrum

Scrum cuenta con cinco eventos para mantener los mínimos necesarios para facilitar el control empírico de procesos funciona y por tanto no dinamitar ninguno de los pilares fundamentales de Scrum.

**Sprint:** Es un contenedor para el resto de eventos de Scrum. Es un bloque de tiempo fijo (aconsejablemente) con una duración de cuatro semanas aproximadamente, en la cual se entrega un incremento. Cada sprint tiene diferentes requisitos y objetivos a lo largo de su duración los cuales el *Scrum Master* se encarga de supervisar que se sigan cumpliendo. Cuando uno acaba comienza inmediatamente el siguiente.

**Reunión de planificación (*Sprint Planning*):** Antes de comenzar el sprint se debe realizar una reunión de planificación del mismo, por lo que deben de estar todos los miembros del equipo. En ella el cliente presenta *Product Backlog* y se establecen los objetivos y desarrollos del sprint. Para cada tarea se debe estimar una carga de trabajo, lo que se conoce como punto de historia.

**Objetivo del sprint (*Sprint Goal*):** Meta establecida en el *Sprint Planning* que se desglosa en las historias de usuario del backlog (ver sección 4.1.4) que pertenecen al sprint. El equipo de desarrollo se concentra en ese objetivo, ya que es el motivo por el que realizar dicho sprint.

**Reunión diaria (*Daily Scrum*):** Es una reunión diaria de 15 minutos aproximadamente que realiza el equipo de desarrollo para hablar del trabajo realizado anteriormente y fija las siguientes tareas a realizar. La función de esta reunión es controlar y evaluar el progreso del equipo de desarrollo.



**Revision del sprint (*Sprint Review*):** Al final del *sprint* se realiza una reunión con la finalidad de revisar si en el incremento se han conseguido los requisitos impuestos, y si no es el caso se modifica y adapta el *backlog* (ver sección 4.1.4) para actualizar requisitos y funcionalidades requeridas, y tener en mente el siguiente *Sprint*. Participan tanto el *Product Owner*, como el *Equipos de desarrollo*, así como *Scrum Master*. Es en esta reunión se comenta qué ha ocurrido durante el *Sprint*, los impedimentos que se han encontrado y las soluciones tomadas.

**Retrospectiva del sprint (*Sprint Retrospective*):** Reunión previa al siguiente *Sprint* con *Scrum Master* con el objetivo de hacer reflexión sobre el último *Sprint* e identificar posibles mejoras para el próximo. El formato común es analizar que ha ido bien durante el *Sprint*, qué ha fallado y que se puede mejorar. Un formato adecuado de retrospectiva se resumiría en cinco fases: preparar el ambiente, recolectar información, generación de ideas, decidir qué hacer y cierre.

### 4.1.4. Artefactos Scrum

Los artefactos son los elementos que te garantizan la transparencia y el registro de la información fundamental del proceso de Scrum. Son los siguientes:

**Product Backlog:** Es la principal fuente de información sobre el producto en Scrum, es decir, es una lista de requisitos del producto (historias de usuario). Dichos requisitos son ordenados por su prioridad y la persona encargada de hacerlo es el *Product Owner*. Al comenzar el proyecto, no es necesaria una lista completa y exhaustiva de todos los elementos, pues puede ir cambiando o añadiendo elementos conforme los requisitos que se pretenden abordar. En estas variaciones también trabaja el Equipo de Desarrollo, siendo un proceso continuo y variable.

**Sprint Backlog:** Se trata de una lista de elementos en los que se tiene que trabajar durante la etapa del *Sprint*. Son elementos que se componen de tareas más técnicas y permiten conseguir un incremento (siguiente artefacto). Ayuda a visualizar el trabajo a realizar durante cada *Sprint* y está gestionado únicamente por el Equipo de Desarrollo. Es fijo, no se puede modificar una vez haya comenzado el *Sprint* y su objetivo es mantener la transparencia dentro del desarrollo actualizando su estado durante toda la iteración a través de las *Daily Scrum*. Este artefacto es muy importante debido a que permite entender cuál es la evolución del trabajo durante el *Sprint*, así como hacer un análisis de riesgos. Hace visibles las tareas necesarias para alcanzar el *Sprint Goal*.

**Incremento:** Es la suma de los elementos completos del *Product Backlog*, es decir, todas las tareas, casos de uso, historias de usuario y cualquier elemento que se haya desarrollado en el *Sprint*. El resultado será puesto a disposición del usuario final en forma de software aportando un valor de negocio.

### 4.2. Planificación.

En este proyecto se ha adaptado el marco de trabajo de Scrum en todos los contextos, utilizando los roles anteriormente explicados para aplicar adecuadamente el proceso empírico en lo que está basado esta metodología. Desde el principio, el alumno ha adaptado los roles de *Equipo de desarrollo* y *Product Owner*. Por su parte, los tutores han adaptado el rol de *Scrum Master* para llevar una correcta y adecuada aplicación de metodología sobre el *Product Backlog* y así mantener un control sobre el proyecto.

La duración de los sprints será de dos semanas. La finalización de cada uno de estos será el martes, por lo que ese mismo día se realizarán las reuniones de *Sprint Review*, *Sprint Retrospective* y el *Sprint Planning* del siguiente.

Se ha utilizado la herramienta Trello para gestionar y supervisar tanto *Product Backlog* como el *Sprint Backlog*. Su uso se ha debido a la necesidad de tener un control constante de las tareas a realizar en cada *Sprint*.

Al comienzo del proyecto se realizará un *Sprint 0*, el cual comienza el 1 de Marzo de 2021. Se destaca esta etapa debido a que es la primera, pero también porque es de vital importancia, ya que se dedica gran cantidad de horas en preparar la documentación, planificación, arquitectura y estructura del proyecto. En esta etapa también se encargará el alumno de investigar, conocer y poner en marcha todas las herramientas que hayan sido consideradas para el proyecto y dejar preparado el entorno de desarrollo del mismo. Este *Sprint* tiene el objetivo de profundizar en los aspectos más importantes del proyecto para el alumno, redactarlo en dicho documento y controlar que se hace una correcta realización de este. Una vez realizado este *Sprint*, los siguientes estarán basados en el desarrollo, implementación y despliegue del proyecto, sin dejar de lado la presente memoria del Trabajo de Fin de Grado.

Consultando la correspondiente guía del Trabajo de Fin de Grado de Ingeniería Informática de la Universidad de Valladolid [19], se ha podido observar que dicho trabajo son 12 créditos, por lo que su equivalencia serían 300 horas. Dada la disponibilidad del alumno, cada interacción serán 35 horas y la estimación para acabar el proyecto será de 8 *Sprint*, dejando 1 *Sprint* de margen para la corrección de errores o posibles imprevistos que puedan surgir.

Se ha invertido gran parte del tiempo de algunos *Sprints* en problemas relacionados con el CORS, por lo que me ha llevado a postponer muchas tareas y dedicar menos tiempo del esperado. Se tuvo otros problemas en la parte de *Back-End* y también se tuvo que dedicar gran parte del tiempo a su investigación. Estos están relacionados con los conversores, métodos para filtrar documentos, etc.

Por todos estos inconvenientes y algunos más, se ha forzado a extender el proyecto y ha sido necesario aumentar los *Sprint*. Se modifica la planificación inicial como se puede ver en la sección 4.4. La fecha estimada para acabar el proyecto sería a finales de septiembre, principios de octubre. Habrá algunas tareas que no serán informadas, ya que son tareas rutinarias cuyo tiempo empleado es menor a cinco minutos (comprobar que se han metido los documentos en la BD, comprobar que cada documento está en su fase, acabar bibliografía, insertar imágenes, etc.)

A continuación se expone la planificación inicial de los sprints:

<b>Sprint</b>	<b>Fecha inicio</b>	<b>Fecha Fin</b>	<b>Observaciones</b>
Sprint 0	01/03/2020	15/03/2020	
Sprint 1	15/03/2020	29/03/2020	
Sprint 2	29/03/2020	12/04/2020	
Sprint 3	12/04/2020	26/04/2020	
Sprint 4	26/04/2020	10/05/2020	
Sprint 5	10/05/2020	24/05/2020	
Sprint 6	24/05/2020	07/06/2020	
Sprint 7	07/06/2020	21/06/2020	
Sprint 8	21/06/2020	05/07/2020	
Sprint 9	05/07/2020	19/07/2020	Periodo extra. Sprint de refuerzo.

Cuadro 4.1: Planificación inicial de sprints

### 4.3. Tareas realizadas

A continuación se van a mostrar en la tabla 4.2 las historias de usuario de este proyecto.

### 4.3. TAREAS REALIZADAS

Núm.	Título	Descripción
1	Consultar últimos documentos publicados	Como usuario quiero consultar los últimos documentos publicados.
2	Filtrar la localidad	Como usuario quiero filtrar por localidad los documentos de las oposiciones.
3	Filtrar la fecha de disposición	Como usuario quiero filtrar por fecha de disposición los documentos de las oposiciones
4	Filtrar rango	Como usuario quiero filtrar el rango (ORDEN, DECRETO...) de los documentos de las oposiciones.
5	Filtrar organismo	Como usuario quiero filtrar por organismo los documentos de las oposiciones.
6	Filtrar fecha de publicación	Como usuario quiero filtrar los documentos de las oposiciones que estén entre rango de fechas o en una fecha en concreto.
7	Filtrar orden de los documentos	Como usuario quiero filtrar los documentos de las oposiciones que contengan dicha orden específica.
8	Filtrar palabras claves	Como usuario quiero filtrar los documentos que contengan las palabras claves utilizadas.
9	Consultar etapa aprobado	Como usuario quiero consultar las oposiciones que se encuentran aprobadas.
10	Consultar etapa abierto	Como usuario quiero consultar las oposiciones que se encuentran abiertas.
11	Consultar etapa provisional	Como usuario quiero consultar los resultados provisionales de las oposiciones.
12	Consultar etapa definitivo	Como usuario quiero consultar los resultados definitivos de las oposiciones.
13	Consultar etapa modificaciones	Como usuario quiero consultar las modificaciones de los documentos de las oposiciones.
14	Filtrar localidad en etapa de aprobado	Como usuario quiero consultar las oposiciones que se encuentran aprobadas filtrando por una localidad específica.
15	Filtrar rango en etapa abierto	Como usuario quiero consultar las oposiciones que se encuentran abiertas y filtrar por un rango en específico.
16	Filtrar fecha de publicación en etapa provisional	Como usuario quiero consultar los resultados provisionales de las oposiciones y filtrar por una fecha en concreto o un rango.
17	Filtrar orden en etapa definitiva	Como usuario quiero consultar los resultados definitivos de las oposiciones y filtrar por una orden en específico.
18	Filtrar palabra clave en etapa abierta	Como usuario quiero consultar las oposiciones que se encuentran abiertas y filtrar por la palabra clave que se quiere encontrar en dichos documentos de la etapa.
19	Filtrar localidad y palabra clave en etapa aprobado	Como usuario quiero consultar el/los documento/s que existan en la etapa de aprobado filtrando por una localidad en específico y una palabra clave.

Cuadro 4.2: Detalle de las historias de usuario

Se ha creado un identificador para cada tarea el cual está formado por: número, tipo y su descripción. También se van a mostrar las horas empleadas de cada tarea en cada sprint y el estado final (completa o incompleta). Esto servirá de apoyo para ver qué tareas se están realizando, cuáles son las siguientes y, si ha habido alguna incompleta, se realizará en el próximo *Sprint*.

Los tipos de tareas se dividen en:

- **DOC.** Tarea asociada a la realización de la documentación, estudio e investigación de la parte teórica del proyecto.
- **DEV.** Tarea relativa al desarrollo del producto.
- **CHECK.** Tarea asociada a la realización de pruebas con la finalidad de comprobar funcionalidad.
- **UPG.** Tarea relativa a realizar mejoras sobre una funcionalidad. Las mejoras observadas en alguna funcionalidad pueden provenir tanto del tutor como de usuarios.

### 4.3.1. Sprint 0

A pesar de que anteriormente se dijo que este *Sprint* se dedicaría únicamente a la realización de la documentación e investigación (tanto de las oposiciones como de las tecnologías a utilizar) y los siguientes iban a ser desarrollo del producto, el próximo *Sprint* tendrá algunas tareas de investigación con la finalidad de estabilizar la carga de trabajo en las dos primeras interacciones.

En la tabla 4.3 se muestran las tareas planificadas y el total de horas consumidas (36 horas), de las cuales hay algunas que no se pudieron finalizar debido a que el tiempo empleado no fue lo suficiente (T - 006) y, por lo tanto, no se podía redactar el capítulo relacionado con el estado del arte (T - 008).

### 4.3. TAREAS REALIZADAS

Tarea	Tipo	Descripción	Horas	Estado
T - 001	DOC	Definición inicial del backlog	1h	Completada
T - 002	DOC	Redacción de la introducción y los objetivos	3h	Completada
T - 003	DOC	Redacción de la planificación	4'5h	Completada
T - 004	DOC	Documentación de las tecnologías que se utilizarán	6h	Completada
T - 005	DOC	Investigación sobre funcionamiento de oposiciones	2'5h	Incompleta
T - 006	DOC	Investigación de boletines de cada provincia	7h	Incompleta
T - 007	DOC	Investigación de BOCyL y su API	6h	Completa
T - 008	DOC	Redacción del capítulo estado del arte	0h	Incompleta
T - 009	DOC	Formación sobre Vue	6h	Completa
<b>Total</b>			<b>36h</b>	

Cuadro 4.3: Tareas del sprint 0

#### 4.3.2. Sprint 1

Este *Sprint* comienza inmediatamente el mismo día que acaba el anterior. Antes de realizar las tareas del respectivo incremento, se intentarán finalizar las anteriores tareas.

En este *Sprint* se documentará el capítulo del estado del arte, pero no se completará porque todavía se sigue entendiendo las diferentes oposiciones, su funcionamiento y las diferentes fases por las que pasan. Mientras, se buscará una forma de categorizar las oposiciones, la cual se debatió con el tutor en la reunión de cada martes, concretamente en reunión *Sprint Planning*.

También se habló de qué herramienta se va a utilizar para realizar la petición todos los días y mantener el proyecto actualizado con las oposiciones de cada día. Las oposiciones que vienen en cada boletín oficial de cada provincia son, casi en su totalidad, las mismas que están en el BOCyL (Boletín Oficial de Castilla y León). Sin embargo, se seguirá investigando por si existiera alguna oposición que esté en el boletín oficial de la provincia y no en el BOCyL.

Por último, se creará el proyecto en GitLab para preparar el entorno de trabajo y tener todo listo para el siguiente *Sprint*. Se empieza a preparar el Front-End, pero en este *Sprint* se dedica más tiempo al Back-End, ya que este requiere más tiempo para entender la conexión con la API y cómo gestionar los datos obtenidos relativos a las oposiciones.

La tabla 4.4 muestra tareas planificadas. Las horas totales consumidas en este *Sprint* han sido de 34 horas.

Tarea	Tipo	Descripción	Horas	Estado
T - 010	DOC	Investigación de boletines de cada provincia	3h	Completada
T - 011	DOC	Investigación sobre funcionamiento de oposiciones	2'5h	Completada
T - 012	DOC	Categorización de oposiciones	4h	Incompleta
T - 013	DOC	Redacción del capítulo del estado del arte	3h	Incompleta
T - 014	DOC	Diseño del proyecto	6h	Completada
T - 015	DOC	Arquitectura y estructura inicial del proyecto	3'5h	Completada
T - 016	DOC	Investigación de herramientas en Firebase	7h	Incompleta
T - 017	DEV	Creación de GitLab	0'5h	Completa
T - 018	DEV	Creación de base de datos y entorno de trabajo	3'5h	Completa
T - 019	DOC	Redacción de <i>Sprints</i> 0 y 1	1h	Completa
<b>Total</b>			<b>34h</b>	

Cuadro 4.4: Tareas del sprint 1

### 4.3.3. Sprint 2

En la anterior interacción se acabó la investigación de los boletines de cada provincia, y se concluyó que las oposiciones que aparecían en cada provincia aparecían en el BOCyL por lo que se utilizará su API para recoger las oposiciones de Castilla y León.

Al inicio de éste *Sprint* se finalizará la tarea doce, en la cual había diferentes ideas para categorizar las oposiciones. Finalmente se redujo a una cuando se hicieron las primeras obtenciones de las oposiciones a través de la API.

Cabe destacar que en un principio se pensó realizar las categorías dependiendo de su grupo funcional, sectores, tipos de plaza... etc. Sin embargo, no en todos los documentos existía dicha información. Por ello finalmente se decidió categorizar las oposiciones dependiendo de la etapa en la que se encontraran.

Las peticiones encargadas de obtener las oposiciones hicieron que surgiera un problema relacionado con el CORS [13]. Dicho problema me ha llevado a postponer la T-022 a la siguiente interacción debido a que esta se solventó utilizando un proxy inverso, pero era una solución temporal y debe realizarse otra. Al obtener las oposiciones, se necesitaba un parseador (T-023) para convertir el archivo XML en un JSON, para visualizar qué estructura tenía dicho objeto, pero no se pudo completar la tarea debido a que no todos convertían correctamente o no funcionaban.

Se encontró una herramienta de Firebase que se encargará de realizar una petición todos los días. Está programada a una determinada hora y permitirá actualizar las oposiciones. Esta herramienta es conocida como *Schedule Function*.

### 4.3. TAREAS REALIZADAS

Tarea	Tipo	Descripción	Horas	Estado
T - 020	DOC	Categorización de oposiciones	1h	Completada
T - 021	DOC	Redacción del capítulo del estado de arte	3'5h	Completada
T - 022	DEV	Obtención y preparación de oposiciones	12h	Incompleta
T - 023	DEV	Desarrollo conversión XML a JSON	4	Incompleta
T - 024	DOC	Investigación de herramientas en Firebase	7h	Incompleta
T - 025	CHECK	Pruebas con functions schedule	7'5h	Completa
T - 026	DOC	Interfaz inicial mostrando oposiciones	1'5h	Completada
T - 027	DOC	Redacción de <i>Sprint 2</i>	0'5	Completada
<b>Total</b>			<b>37h</b>	

Cuadro 4.5: Tareas del sprint 2

El desglose al completo de las tareas a realizar en esta interacción se puede ver en la tabla 4.5, el cuál he realizado 37 horas en total.

#### 4.3.4. Sprint 3

En esta interacción sigue existiendo el problema relacionado con el CORS. Antes de seguir explicando, es necesario saber que se va a tratar con documentos, es decir, una oposición pasa por una serie de fases, y dichas fases se dividen en documentos.

Al hacer las peticiones para obtener los documentos, se consiguen datos significantes relacionados con la oposición, como el organismo, rango, fecha de publicación, fecha de disposición, etc. Pero es necesario hacer otra petición sobre el XML del documento para conseguir parsear los datos contenidos en dicho XML y obtenerlos en un objeto JSON. Por lo que, antes de parsear, hay que conseguir hacer la petición sobre el XML, pero no se puede debido al CORS.

A pesar de estos problemas, continué realizando otras tareas relacionadas con la inserción en la base de datos, seguí investigando las herramientas de Firebase, creando otra vista al clicar en un documento para mostrar algunos datos relevantes. Se modificó la vista principal donde se muestran las oposiciones: se añadieron pestañas dependiendo de su rango (solo para testear) para comprobar como podría quedar cuando se realice la división de las oposiciones por fases.

En la tabla 4.6 se muestran las tareas planificadas, las cuales hay algunas del anterior *Sprint* que no se han podido solventar. En total se han empleado 34'25 horas para la realización de esta interacción.



Tarea	Tipo	Descripción	Horas	Estado
T - 028	DEV	Obtención y preparación de las oposiciones	14h	Incompleta
T - 029	DEV	Desarrollo conversión XML a JSON <i>Front-End</i>	2'5h	Incompleta
T - 030	CHECK	Pruebas conversor en <i>Front-End</i>	1h	Incompleta
T - 031	DOC	Investigación de herramientas en Firebase	4h	Incompleta
T - 032	DEV	Creación de barra superior	0'75h	Incompleta
T - 033	DEV	Creación de nueva vista (Información documento)	3'5	Completa
T - 034	DEV	Integración y flujo de nueva vista	3'5h	Completa
T - 035	DEV	Modificación configuración Firebase	1'5h	Completa
T - 036	CHECK	Pruebas de inserción en base de datos	3h	Completa
T - 037	DOC	Redacción de <i>Sprint 3</i>	0'5h	Completa
<b>Total</b>			<b>34'25h</b>	

Cuadro 4.6: Tareas del sprint 3

#### 4.3.5. Sprint 4

Al comienzo de este *Sprint* se consiguen resolver parcialmente los problemas con el CORS. La solución temporal aplicada me permitió seguir avanzando en el proyecto y analizar los datos obtenidos, es decir, el objeto JSON. La solución se realizó en *Front-End* y se basaba en cambiar el prefijo “http://bocyl.jcyl.es/” por el prefijo de la máquina del alumno (localhost) y el puerto utilizado. Esto no es una solución permanente, pero servirá para analizar el objeto JSON.

En *Back-End*, por su parte, no sabía qué utilizar para realizar las conversiones (porque no sirvió la misma solución que en *Front-End*) e insertar los documentos en la base de datos.

Como consecuencia, seguiría investigando sobre un parseador que me funcionara en *Back-End* y una herramienta de Firebase que me permitiera insertar mensualmente o un conjunto de meses, ya que así se estabilizaría la carga de trabajo.

En la tabla 4.7 se desglosan las tareas realizadas en esta interacción. Se dedica mucho tiempo a obtener y preparar las peticiones, ya que el problema del CORS está siempre presente. El total de horas empleadas han sido de 29 horas.

### 4.3. TAREAS REALIZADAS

Tarea	Tipo	Descripción	Horas	Estado
T - 038	DEV	Obtención y preparación de las oposiciones	9h	Incompleta
T - 039	DEV	Análisis objeto JSON	3'5h	Completa
T - 040	DEV	Desarrollo conversión XML a JSON <i>Front-End</i>	1'5h	Completa
T - 041	CHECK	Pruebas conversor en <i>Front-End</i>	3h	Completa
T - 042	DEV	Desarrollo conversión XML a JSON <i>Back-End</i>	3'5h	Incompleta
T - 043	CHECK	Pruebas conversor en <i>Back-End</i>	0'5h	Incompleta
T - 044	DOC	Investigación de herramientas en Fire-base	4h	Incompleta
T - 045	DOC	Investigación herramienta para categorizar oposiciones	3h	Incompleta
T - 046	DOC	Redacción de <i>Sprint 4</i>	1h	Completa
<b>Total</b>			<b>29h</b>	

Cuadro 4.7: Tareas del sprint 4

#### 4.3.6. Sprint 5

Este *Sprint* se centra principalmente en la parte de *Back-End* para finalizar algunas tareas. Se solventó el problema del CORS en *Back-End*. Este problema sucedía cuando se lanzaban las peticiones para conseguir los documentos de las oposiciones a través de la API cuya conexión era client-to-server. No obstante, al realizar las peticiones desde *Back-End* la conexión era server-to-server y se solucionó el problema. De esta manera, la solución realizada en la anterior interacción no sirvió porque el cambio de prefijo no sería necesario, ya que las peticiones se lanzan desde el servidor de Firebase.

Antes de resolver dicho problema, se finalizó la tarea 044 del anterior *Sprint* utilizando las functions de firebase para realizar una function por cada mes que queremos insertar en la base de datos.

Se siguió investigando qué conversor utilizar, porque a pesar de que ya obteníamos el XML y, por consecuente, los datos del documento, no se podía observar con claridad qué contenía estos debido a que no era un objeto JSON. En paralelo a esto, se realizaron pruebas no satisfactorias con las functions por falta de formación.

También se realizaron algunas mejoras en cada vista, tanto donde se muestran los documentos de las oposiciones como cuando se clicha en un documento.

En la tabla 4.8 se muestran las tareas planificadas. En total se han empleado 30 horas.

Tarea	Tipo	Descripción	Horas	Estado
T - 047	DEV	Obtención y preparación de las oposiciones	4h	Completa
T - 048	DOC	Investigación de herramientas en Fire-base	2h	Completa
T - 049	DEV	Formación functions	6h	Completa
T - 050	CHECK	Inserción con functions	5'5h	Incompleta
T - 051	DEV	Desarrollo conversión XML a JSON <i>Back-End</i>	4h	Incompleta
T - 052	CHECK	Pruebas conversor en <i>Back-End</i>	2'5h	Incompleta
T - 053	DOC	Investigación herramienta para categorizar oposiciones	3h	Incompleta
T - 054	UPG	Mejora en vista principal (muestra oposiciones)	3h	Completa
T - 055	UPG	Mejora en vista contenido de documento	3h	Completa
T - 056	DOC	Redacción de <i>Sprint</i> 5	1h	Completa
<b>Total</b>			<b>30h</b>	

Cuadro 4.8: Tareas del sprint 5

### 4.3.7. Sprint 6

A lo largo de este sexto *Sprint* se van a abordar tareas inacabadas y se va avanzará notablemente el proyecto.

Se encontró un parseador que funcionase en *Back-End*, un proceso que necesitó de varias herramientas. Primeramente, al hacer la petición a enlace XML, fue necesario guardar el resultado codificado en binario. Después, se pasó a un string y se decodificó de nuevo a XML. Una vez obtenido el XML de la manera más adecuada para el proyecto, se utilizó el parseador de XML a objeto JSON.

Por lo tanto, en este *Sprint* se puso más atención a la forma de utilizar las functions e insertar documentos en la base de datos. Se tuvo que aumentar tanto el tiempo de espera como la memoria utilizada por la function.

Una vez terminada una gran parte del desarrollo del producto, se dedicó gran parte del tiempo en decidir qué herramientas utilizar para categorizar las oposiciones por fases. Para ello, fueron necesarias varias reuniones con el tutor y finalmente se escogieron las expresiones regulares (regex).

Esta interacción me resultó interesante, ya que conocí nuevas herramientas de gran utilidad. Se han empleado un total de 32'25 horas como se puede comprobar en la tabla 4.9

### 4.3. TAREAS REALIZADAS

Tarea	Tipo	Descripción	Horas	Estado
T - 057	DEV	Desarrollo conversión XML a JSON	4h	Completa
T - 058	CHECK	Pruebas conversor en <i>Back-End</i>	1'5h	Completa
T - 059	CHECK	Inserción con functions	3'5h	Completa
T - 060	DEV	Inserción documentos año 2020	6'5h	Completa
T - 061	DOC	Creación de método para petición de documentos	2'5h	Completa
T - 062	DOC	Investigación herramienta para categorizar oposiciones	1'5h	Completa
T - 063	DOC	Redacción de tecnologías utilizadas	1'75h	Completa
T - 064	DOC	Formación en expresiones regulares	8h	Incompleta
T - 065	UPG	Mejora en tabla de vista principal (reactividad)	3'5h	Completa
T - 066	DOC	Redacción de <i>Sprint 6</i>	0'5h	Completa
<b>Total</b>			<b>32'25h</b>	

Cuadro 4.9: Tareas del sprint 6

#### 4.3.8. Sprint 7

En este *Sprint* se van a centrar las tareas en formalizarse en las expresiones regulares y aplicarlas para categorizar los documentos de las oposiciones.

Tanto esta interacción como la siguiente se van a dedicar a analizar los documentos del BOCyL y a realizar las expresiones regulares. Se investigó enero y febrero, para que posteriormente se realizarán sus expresiones regulares. Las tareas (T-071 hasta T-075) de crear las expresiones regulares para las etapas se acabarán, pero se irán refinando continuamente, hasta llegar a un punto que no sea necesario modificar las expresión regular de cada etapa.

Hubo un problema de entendimiento con las etapas de los documentos, pero se solventó refactorizando. En un primer momento se pensó que se podían dividir en: previsto, aprobado, bases, abierto, provisional, resultados y modificaciones. Pero no eran necesarias tantas fases, por lo que después de una reunión con los tutores propusieron la siguiente alternativa: aprobado, abierto, provisional, resultados y modificaciones.

Entendiendo mejor las expresiones regulares, se utilizó una página [31] para la creación temporal de las etapas: aprobado, provisional, resultados y modificaciones. Resultados probablemente se cambie por definitivo. Se dejó abierto debido a que eran muchos documentos los que recogían esa etapa, y tenía una carga de trabajo demasiado grande para realizarla en este *Sprint*.

En esta iteración se han consumido un total de 37 horas como se puede comprobar en la tabla 4.10

Tarea	Tipo	Descripción	Horas	Estado
T - 067	DOC	Formación en expresiones regulares	6h	Incompleta
T - 068	DOC	Refactorización de etapas	4h	Completa
T - 069	DOC	Análisis documentos de enero	3h	Completa
T - 070	DOC	Análisis documentos de febrero	5h	Completa
T - 071	DEV	Creación de regex aprobado	3'5h	Completa
T - 072	DEV	Creación de regex provisional	5'5h	Completa
T - 073	DEV	Creación de regex resultados/definitivo	5h	Completa
T - 074	DOC	Creación de regex modificaciones	3'5h	Completa
T - 075	DOC	Creación de regex abierto	1h	Incompleta
T - 076	DOC	Redacción de <i>Sprint 7</i>	0'5h	Completa
<b>Total</b>			<b>37h</b>	

Cuadro 4.10: Tareas del sprint 7

### 4.3.9. Sprint 8

En esta interacción se completó primeramente algunas de las tareas del anterior *Sprint*. Estas tareas se basaban en la formalización de las regex y la creación de una de sus etapas.

Después se analizó marzo, el cuál contenía muchos documentos (163). Con ello, las mejoras de cada expresión regular perteneciente a cada etapa se realizaron correctamente. No obstante, hubo error al mejorar la expresión regular de la etapa abierto, ya que ésta era la más complicada.

Al analizar el mes de abril, las expresiones regulares se mejoraron correctamente a excepción de la etapa abierto. En dicho mes, las expresiones regulares empezaron a dar sus frutos y cada vez eran menos documentos los que se tenía que analizar.

Por último, a partir de este *Sprint* no se pudo dedicar todo el tiempo que necesitaba el proyecto, ya que el alumno tenía que realizar otras tareas relacionadas con la universidad (prácticas de empresa). Por lo que se propuso al tutor en una reunión la nueva fecha de entrega.

En esta iteración se han consumido 44 horas como se puede comprobar en la tabla 4.11

### 4.3. TAREAS REALIZADAS

---

Tarea	Tipo	Descripción	Horas	Estado
T - 077	DOC	Formación en expresiones regulares	5h	Completa
T - 078	DOC	Creación de regex abierto	8h	Completa
T - 079	DOC	Análisis documentos de marzo	6h	Completa
T - 080	DOC	Análisis documentos de abril	4'5h	Completa
T - 081	UPG	Mejora de regex aprobado	2h	Completa
T - 082	UPG	Mejora de regex provisional	4h	Completa
T - 083	UPG	Mejora de regex resultados/definitivo	4h	Completa
T - 084	UPG	Mejora de regex modificaciones	3h	Completa
T - 085	UPG	Mejora de regex abierto	7h	Completa
T - 086	DOC	Redacción de <i>Sprint 8</i>	0'5h	Completa
<b>Total</b>			<b>44h</b>	

Cuadro 4.11: Tareas del sprint 8

#### 4.3.10. Sprint 9

Se utiliza el *Sprint* de refuerzo y el proyecto sufrirá una ampliación de *Sprints*. Esta modificación se explicará en la siguiente sección.

En esta interacción se hacen las mismas tareas, pero en distintos meses. Al llevar cuatro meses trabajando con las expresiones y analizando los documentos, se obtienen resultados satisfactorios. El motivo de esto es que son menos los documentos que al utilizar las expresiones regulares se quedan sin etapa, es decir, al utilizar las regex por cada documento, las identifica en alguna etapa (aprobado, abierto, privisional, resultados o modificaciones), pero cuando la regex no está mejorada del todo, hay documentos que no identifica con ninguna de éstas y por tanto hay que mejorar la regex.

Se analizan los meses mayo y junio. Se consigue mejorar satisfactoriamente la regex correspondiente a abierto. Se crea un método en *Back-End* para modularizar mucho más el código y que quede más limpio. Este se basa en aplicar las expresiones regulares y aplicar condiciones para que dependiendo de lo que nos devolviera verdadero o falso, ese documento era una etapa u otra.

A modo de ejemplo, para que un documento esté en “modificaciones”, solamente se comprobará si devuelve 'verdadero'. Por otro lado, para que un documento forme parte de la etapa provisional tiene que devolver verdadero al comprobar por la regex de provisional y falso al comprobar por la regex de modificaciones. Esto se aplica tanto en provisional como en definitivo. Esto es debido a que en este tipo de documentos se han encontrado muchas palabras claves en las que devuelve verdadero modificaciones y se trata por tanto de una modificación. Por eso mismo, se hacen estas condiciones.

La tabla 4.12 muestra las tareas realizadas en esta iteración. Se ha consumido un total de 21 horas.

Tarea	Tipo	Descripción	Horas	Estado
T - 087	DOC	Análisis documentos de mayo	3'5h	Completa
T - 088	DOC	Análisis documentos de junio	3h	Completa
T - 089	DEV	Creación de método para categorizar documentos en <i>Back-End</i>	1h	Completa
T - 090	UPG	Mejora de regex aprobado	1'5h	Completa
T - 091	UPG	Mejora de regex provisional	2h	Completa
T - 092	UPG	Mejora de regex resultados/definitivo	2'5h	Completa
T - 093	UPG	Mejora de regex modificaciones	1'75h	Completa
T - 094	UPG	Mejora de regex abierto	3'25h	Incompleta
T - 095	DOC	Redacción de <i>Sprint 9</i>	0'5h	Completa
<b>Total</b>			<b>21h</b>	

Cuadro 4.12: Tareas del sprint 9

## 4.4. Modificación de planificación

Segun la planificación inicial, el proyecto se tenía que haber acabado como muy tarde en julio, pero todavía había que realizar tareas. Estas tareas están relacionadas con la funcionalidad de la página, la eficiencia, el despliegue y tomas de decisiones sobre documentos con los tutores. La nueva fecha de finalización de proyecto se verá afectada y estará próxima a los meses de septiembre y octubre.

Se ampliarán 7 *Sprints* a mayores para completar las tareas restantes. Dicho esto, la nueva planificación quedará reflejado como en la tabla 4.13.

<b>Sprint</b>	<b>Fecha inicio</b>	<b>Fecha Fin</b>	<b>Observaciones</b>
Sprint 0	01/03/2021	15/03/2021	
Sprint 1	15/03/2021	29/03/2021	
Sprint 2	29/03/2021	12/04/2021	
Sprint 3	12/04/2021	26/04/2021	
Sprint 4	26/04/2021	10/05/2021	
Sprint 5	10/05/2021	24/05/2021	
Sprint 6	24/05/2021	07/06/2021	
Sprint 7	07/06/2021	21/06/2021	
Sprint 8	21/06/2021	05/07/2021	
Sprint 9	05/07/2020	19/07/2020	
Sprint 10	19/07/2020	02/08/2020	
Sprint 11	02/08/2020	16/08/2020	
Sprint 12	16/08/2020	30/08/2020	
Sprint 13	30/08/2020	13/09/2020	
Sprint 14	13/09/2020	27/09/2020	Sprint de refuerzo.

Cuadro 4.13: Planificación final de sprints

### 4.4.1. Sprint 10

En este *Sprint* se van a seguir mejorando las expresiones regulares utilizando los documentos de las oposiciones de 2020. Estos documentos se insertaron en bases de datos (T-060) para su posterior análisis. Se han analizado los documentos del año 2020 con el objetivo de mejorar las expresiones regulares, no para mostrarlas posteriormente en el despliegue final. El tiempo empleado es muy inferior debido a que son pocos los documentos que no se identifican en una etapa (a excepción de algunos meses).

En paralelo a la anterior tarea comentada, se realizó otra tarea relacionada con la creación de filtros. En primer lugar, se crearon filtros sencillos y objetivos, que sirvan para que el usuario pueda encontrar el documento de la oposición que está buscando. Alguno de ellos son: localidad, fecha de disposición, rango, organismo y fecha de publicación. Además, se cambió el nombre de la etapa “resultados” por “definitivo”. No se pudo completar algunos filtros debido a falta de tiempo por parte del alumno.

Tarea	Tipo	Descripción	Horas	Estado
T - 096	DOC	Análisis documentos enero 2020	1'5h	Completa
T - 097	DOC	Análisis documentos de febrero 2020	2h	Completa
T - 098	DEV	Creación de filtro localidad	2h	Completa
T - 099	DEV	Creación de filtro fecha de disposición	0.25h	Incompleta
T - 100	DEV	Creación de filtro rango	2'5h	Completa
T - 101	DEV	Creación de filtro organismo	2h	Completa
T - 102	DEV	Creación de filtro fecha de publicación	1'5h	Incompleta
T - 103	UPG	Mejora de regex aprobado	1h	Completa
T - 104	UPG	Mejora de regex definitivo	0'5h	Completa
T - 105	UPG	Mejora de regex modificaciones	2h	Completa
T - 106	UPG	Mejora de regex abierto	3'5h	Completa
T - 107	DOC	Redacción de <i>Sprint</i> 10	0'25h	Completa
<b>Total</b>			<b>19h</b>	

Cuadro 4.14: Tareas del sprint 10

### 4.4.2. Sprint 11

El anterior *Sprint* se dejó incompleta la tarea 102 (relacionada con fecha de publicación) y se completó en esta interacción. Ésta resuelve a la vez la T-099. Se analizarán los meses de marzo, abril, mayo, junio y julio de 2020. Respectivamente tenían 18, 1, 1, 2 y 10 documentos que no se identificaban con ninguna etapa, por lo que se dedicó gran atención en mayo y julio. Además realizar estos análisis y mejorar las expresiones regulares, se comprobó que en 2021 no nos dé ningún falso positivo, es decir, que al modificar alguna expresión regular un documento dé falso verdadero en otra etapa que no le corresponde.

Por otro lado, se realizó un nuevo filtro que ayudará a encontrar más fácilmente el documento al usuario. Este filtro se ha llamado “Palabra clave”. Su finalidad se basa en devolver todos los documentos que contengan la palabra a filtrar por el usuario. Por el momento se



hizo para una palabra o un 'string' (como digo eso), pero no se completó la tarea. Todos los filtros realizados hasta el momento era reactivos, es decir, por cada letra que se introduce te aparecen los documentos coincidentes. Esto con pocos documentos es rápido, pero cuando los documentos sea un número elevado, el cómputo será más grande y puede que haya modificar esta parte de los filtros de la tabla.

Por último, se empezó a investigar como se realizaría el despliegue. Se observó que Firebase tiene una herramienta propia para poder desplegar.

Tarea	Tipo	Descripción	Horas	Estado
T - 108	DOC	Análisis documentos marzo 2020	4h	Completa
T - 109	DOC	Análisis documentos de abril 2020	0'25h	Completa
T - 110	DOC	Análisis documentos de mayo 2020	0'25h	Completa
T - 111	DOC	Análisis documentos de junio 2020	0'25h	Completa
T - 112	DOC	Análisis documentos de julio 2020	2h	Completa
T - 113	DEV	Creación de filtro fecha de disposición	0'25h	Completa
T - 114	DEV	Creación de filtro fecha de publicación	1'5h	Completa
T - 115	DEV	Creación de palabra clave	5h	Incompleta
T - 116	CHECK	Comprobación falsos positivos 2021	6h	Completa
T - 117	UPG	Mejora de regex aprobado	2'5h	Completa
T - 118	UPG	Mejora de regex definitivo	2h	Completa
T - 119	UPG	Mejora de regex modificaciones	3'5h	Completa
T - 120	UPG	Mejora de regex abierto	5h	Completa
T - 121	DOC	Investigación sobre herramienta para despliegue	4h	Completa
T - 122	DOC	Redacción de <i>Sprint</i> 11	0'25h	Completa
<b>Total</b>			<b>36'75h</b>	

Cuadro 4.15: Tareas del sprint 11

### 4.4.3. Sprint 12

En este *Sprint* se acabó de analizar los documentos del año 2020, se mejoraron las expresiones regulares y se volvió a comprobar los falsos verdaderos de 2021. Seguidamente se mejoró el filtro de palabra clave y se propuso otro filtro. Seguía existiendo documentos que no se identificaban en ninguna etapa: agosto (3), septiembre (4), octubre(5), noviembre (7), diciembre (7).

En un primer momento, los filtros utilizados en la tabla de la vista principal, son reactivos. Esta manera no era muy eficiente, ya que cuando se hiciera la base de datos de las oposiciones de 2021 los tiempos de espera no iban a ser muy bajos. Como consecuencia, se empezó a trabajar en botones de "buscar" "limpiar" para que el usuario fuera consciente de que le es necesario clicar al botón tanto para filtrar como para limpiar y así los filtros no sean reactivos. Esta tarea se consiguió satisfactoriamente, pero con algunos inconvenientes.

Todos los filtros funcionaban correctamente a excepción de la palabra clave, esta seguía

#### 4.4. MODIFICACIÓN DE PLANIFICACIÓN

---

teniendo problemas porque el objeto JSON al acceder a un parrafo o a una tabla, cada vector se desglosaba de manera diferente. Se creó un método para poder acceder a las dos formas, pero se invirtió bastante tiempo.

En una reunión con los tutores se propuso un nuevo filtro llamado ".orden". Su finalidad es encontrar todos los documentos donde se encuentre la orden que filtre el usuario. La orden sirve para identificar ciertos documentos (siempre van a ser documentos que están en la etapa abierta y son de rango tipo orden) que han sido creados con un identificador con la siguiente estructura: CA/Número/Año.

**CA:** Son tres letras que hace referencia al código alfabético del departamento.

**Número:** Número.

**Año:** Año.

Este filtro va orientado para los usuarios que estén pendientes de alguna modificación que se haya realizado sobre una oposición. Toda modificación (resolución, decreto... cualquier documento) de una oposición debe hacer referencia al documento que hace pública su convocatoria (dicha orden), es decir, abre plazo de inscripción para presentarse a la oposición. No siempre se abre una convocatoria con el rango de tipo orden, pero sí una gran parte son de tipo orden, por eso mismo se decidió implementar dicho filtro. No obstante, no quiere decir que todas las ordenes van a ser documentos que abran convocatorias, cualquiera que sea orden tiene este identificador y será para identificar un documento que está en la etapa aprobado, abierto, provisional, definitivo o modificaciones.

Tarea	Tipo	Descripción	Horas	Estado
T - 123	DOC	Análisis documentos agosto 2020	0'25h	Completa
T - 124	DOC	Análisis documentos de septiembre 2020	0'25h	Completa
T - 125	DOC	Análisis documentos de octubre 2020	0'5h	Completa
T - 126	DOC	Análisis documentos de noviembre 2020	0'5h	Completa
T - 127	DOC	Análisis documentos de diciembre 2020	1h	Completa
T - 128	DEV	Creación de filtro orden	5h	Completa
T - 129	DEV	Creación de palabra clave	8h	Completa
T - 130	UPG	Mejora de tabla y filtros (sin reactividad)	3'5h	Completa
T - 131	CHECK	Comprobación falsos positivos 2021 (se repite)	4h	Completa
T - 132	UPG	Mejora de regex aprobado	0'5h	Completa
T - 133	UPG	Mejora de regex definitivo	1'5h	Completa
T - 134	UPG	Mejora de regex modificaciones	2'5h	Completa
T - 135	UPG	Mejora de regex abierto	3h	Completa
T - 136	DOC	Redacción de <i>Sprint</i> 12	0'25h	Completa
<b>Total</b>			<b>30'75h</b>	

Cuadro 4.16: Tareas del sprint 12

#### 4.4.4. Sprint 13

En esta iteración primeramente se hace un análisis de los documentos de junio (otra vez), julio y agosto para insertarlo en la base de datos. En consecuencia, se mejoraron los filtros, pero apenas hubo mejoras, solo en algunos. En paralelo se crea la *schedule function* para que empiece a insertar los nuevos documentos cada día.

Se mejora la lógica de los filtros creando *watchers* [30] y limpiando código. Estos *watchers* sirven como un observador personalizado, que cada vez que una variable cambia su valor, emite un evento, método o lo que quiera el desarrollador. En este caso, se ha utilizado para cada vez que se cambia entre diferentes fases y para los filtros fecha de disposición y fecha publicación. Se mejoran los filtros orden y palabra clave (se creó un componente). Se redujo el código y se hizo de manera más eficiente.

Por último, se despliega la página y se comparte con algunas personas. La mayoría de estas personas, tenían conocimientos sobre el BOCyL, por lo que no tuvieron problemas a la hora de entender los filtros. No obstante, hubo personas que no entendieron algunos filtros y por tanto los utilizaron. Estas personas echaron de menos una breve explicación de cada filtro, así como de cuál era la función de la página web. La página al principio tarda unos segundos, alrededor de 15, debido a que tiene que traer todos los documentos y cada vez que se clica en una fase, se hace una petición a la base de datos, consiguiendo solo los de esa fase (el tiempo de espera es muy inferior). Se debatirá con los tutores algunas alternativas para mejorar el tiempo de espera.

#### 4.4. MODIFICACIÓN DE PLANIFICACIÓN

---

Como se puede observar en la tabla 4.17 se ha consumido un total de 22,5 horas en las tareas que se han realizado en este *Sprint*.

Tarea	Tipo	Descripción	Horas	Estado
T - 137	DOC	Análisis documentos junio	0'25h	Completa
T - 138	DOC	Análisis documentos de julio 2020	1h	Completa
T - 139	DOC	Análisis documentos de agosto 2020	0'5h	Completa
T - 140	DEV	Creación de watchers	2h	Completa
T - 141	UPG	Mejora de filtro fecha disposición	0'5h	Completa
T - 142	UPG	Mejora de filtro fecha publicación	0'5h	Completa
T - 143	UPG	Mejora de filtro de orden	2h	Completa
T - 144	UPG	Mejora de filtro palabra clave	4h	Completa
T - 145	UPG	Creación y puesta en marcha de schedule function	5h	Completa
T - 146	DEV	Despliegue de página web	2h	Completa
T - 147	DEV	Creación de formulario para recoger feedback	1h	Completa
T - 148	UPG	Mejora de regex modificaciones	2h	Completa
T - 149	UPG	Mejora de regex abierto	1'5h	Completa
T - 150	DOC	Redacción de <i>Sprint</i> 13	0'25h	Completa
<b>Total</b>			<b>22'5h</b>	

Cuadro 4.17: Tareas del sprint 13

#### 4.4.5. Sprint 14

En esta última iteración se hacen mejoras que se han hablado en la daily con los profesores y aportaciones de las personas que han probado la página web.

Una de las primeras tareas fue arreglar la schedule function, ya que funcionaba algunos días y otros no. Una vez arreglado este error crítico, se hicieron mejoras que se intuían en el anterior *Sprint*: explicar filtros y función de la página. Se mejoró el filtro palabra clave, antes solo podías meter un conjunto de palabras o una palabra, ahora puedes meter una lista de palabras si fuera necesario. Este filtro está orientado para personas que quieren encontrar su puesto de trabajo en el documento, pero también puede que venga su grupo. A modo de ejemplo, una persona que busca ser policía, introducirá esa palabra, después introducirá el subgrupo correspondiente a esa oposición, C1. La página web devolverá los documentos que contengan las palabras policía y c1. Hay que respetar siempre las tildes.

Otro problema, era el tiempo de espera. Se creó un loading en la vista principal, para que el usuario supiera que la página estaba funcionando y tendría que esperar. No obstante, esto no gustaba y se planteó a los tutores que en vez de poner una columna con todos los documentos se trajeran solo los más recientes y cuando el usuario se desplazara a otra fase, se hiciera una petición a la base de datos. Se realizó de esta manera y los tiempos de espera bajaron considerablemente, teniendo solo que esperar como máximo en la fase de abierto (es la que contiene más documentos) 5-7 segundos.

CAPÍTULO 4. PLANIFICACIÓN Y DESARROLLO DEL PROYECTO

---

Tarea	Tipo	Descripción	Horas	Estado
T - 151	DEV	Solucionar error crítico de schedule function	7h	Completa
T - 152	UPG	Mejora de vista principal (Explicación de filtros y página web)	5h	Completa
T - 153	DOC	Mejora de palabra clave	3'5h	Completa
T - 154	UPG	Mejora de tiempo de espera	7h	Completa
T - 155	DOC	Bibliografía	1h	Completa
T - 156	DOC	Redacción de manual de usuario	4h	Completa
T - 157	DOC	Redacción de manual de despliegue	2h	Completa
T - 158	DOC	Redacción de <i>Sprint 14</i>	0'25h	Completa
<b>Total</b>			<b>29'75h</b>	

Cuadro 4.18: Tareas del sprint 14

Todas las tareas realizadas están en la tabla 4.18 y se ha dedicado un total 29'75 de horas. Por último, se dedicó gran parte a completar algunos capítulos de la memoria, se completó la bibliografía y se redactaron los manuales de usuario y de despliegue.



## Capítulo 5

# Plan de riesgos y estimación de costes

### 5.1. Plan de riesgos

El riesgo [7] es un suceso que si ocurre, puede alterar el desarrollo normal y previsto de un proyecto, tarea o actividad, produciendo un daño.

En un proyecto siempre existen factores que pueden poner en peligro el alcance de los objetivos en los plazos y con los costes, recursos y prestaciones previstas. Los factores de riesgo pueden ser de muy distinta naturaleza y dependen del tipo de proyecto del que se trate.

A lo largo del proyecto es conveniente tener una adecuada planificación y gestión de riesgos. Estos se pueden clasificar según las posibilidades de predicción:

- **Predecibles y conocidos:** se pueden identificar analizando el proyecto y su entorno. Se puede estimar probabilidad de ocurrencia.
- **Predecibles:** Se pueden intuir a partir de la experiencia en otros proyectos, pero no se puede estimar la probabilidad de ocurrencia.
- **Impredecibles:** Pueden ocurrir, pero son difíciles de identificar con anticipación.

La procedencia de los riesgos es variada. Puede ser: técnica, administrativa, contractual, financiera, política, derivada de recursos inadecuados o insuficientes, de relaciones laborales, del entorno, de impacto social, por fuerza mayor, de mercado, de dirección y gestión y otros.

Para realizar un plan de riesgos correcto es recomendable seguir una serie de pasos.

1. Identificación de riesgos.
2. Análisis y establecimiento de prioridades.
3. Planificación de riesgos.
4. Monitorización de riesgos.

### 5.1.1. Riesgos encontrados en este proyecto

En este apartado se va a exponer un análisis de los riesgos de mayor importancia que se han encontrado a lo largo del proyecto.

A continuación, se exponen una serie de tablas, una para cada riesgo, en las que se ofrece una breve descripción del riesgo, su probabilidad de que suceda (muy baja, baja, media, alta y muy alta), su impacto (muy baja, baja, media, alta y muy alta), un plan de mitigación que reduzca su probabilidad y un plan de contingencia. Los riesgos expuestos vienen descritos desde la tabla 5.1 hasta la tabla 5.11

<b>Riego</b>	Riesgo-01
<b>Descripción</b>	Un miembro del equipo de desarrollo no presenta la suficiente formación en las tecnologías empleadas
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Medio
<b>Plan de mitigación</b>	Se han utilizado tecnologías con las que se ha trabajado a lo largo de la carrera y se están actualmente utilizando en las prácticas de empresa
<b>Plan de contingencia</b>	Se ofree ayuda al miembro del equipo para obtener conocimientos sobre dichas tecnologías

Cuadro 5.1: Riesgo de falta de formación

<b>Riesgo</b>	Riesgo-02
<b>Descripción</b>	Un miembro del equipo de desarrollo enferma y no se ve capacitado para desarrollar las tareas asignadas
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Alto
<b>Plan de mitigación</b>	Disponer de un tiempo extra al final del proyecto. En este proyecto, se ha previsto con un <i>Sprint</i> de refuerzo al final para infortunios de este tipo
<b>Plan de contingencia</b>	Retrasar algunas tareas para el próximo <i>Sprint</i>

Cuadro 5.2: Riesgo de enfermedad



<b>Riego</b>	Riesgo-03
<b>Descripción</b>	Se planificó un tiempo inadecuado para realizar tareas que requerían de más tiempo
<b>Probabilidad</b>	Media
<b>Impacto</b>	Medio
<b>Plan de mitigación</b>	Disponer de un tiempo extra al final del proyecto. En este proyecto, se ha previsto con un <i>Sprint</i> de refuerzo al final para infortunios de este tipo
<b>Plan de contingencia</b>	Retrasar algunas tareas para el próximo <i>Sprint</i>

Cuadro 5.3: Riesgo de retraso de las tareas

<b>Riego</b>	Riesgo-04
<b>Descripción</b>	Por disponibilidad horaria, el <i>Scrum Master</i> (rol ejercido por el tutor) no se encontrará disponible para acudir a las reuniones importantes a lo largo del desarrollo del proyecto.
<b>Probabilidad</b>	Media
<b>Impacto</b>	Bajo
<b>Plan de mitigación</b>	Se acuerdan reuniones con antelación para los posibles imprevistos. En caso de que no haya disponibilidad de reunirse, se usan emails para contactar y plantear dudas concretas.
<b>Plan de contingencia</b>	El desarrollador se le exige una mayor responsabilidad que tiene que asumir <i>Sprint</i>

Cuadro 5.4: Riesgo de ausencia del *Scrum Master*

<b>Riego</b>	Riesgo-05
<b>Descripción</b>	Debido al aprendizaje que se iba obteniendo acerca de las oposiciones a lo largo del proyecto, fue necesario modificar las funcionalidades a realizar a lo largo del proyecto (tanto añadiendo como restando)
<b>Probabilidad</b>	Media
<b>Impacto</b>	Medio
<b>Plan de mitigación</b>	El proyecto ha sido creado desde un desarrollo ágil, por lo que permite añadir o eliminar funcionalidades según se crea conveniente.
<b>Plan de contingencia</b>	En las reuniones se vuelven a planificar las siguientes tareas del <i>Sprint</i> o de los siguientes.

Cuadro 5.5: Riesgo de modificación de funcionalidades

<b>Riego</b>	Riesgo-06
<b>Descripción</b>	La información necesaria no se encontró, o al menos no con facilidad, por lo que fue más difícil el flujo/entendimiento de las oposiciones
<b>Probabilidad</b>	Media
<b>Impacto</b>	Alto
<b>Plan de mitigación</b>	Se obtiene información de personas que han trabajado para la Administración o tengan los conocimientos suficientes sobre el tema del proyecto
<b>Plan de contingencia</b>	Se comparara información de diferentes fuentes, tanto de internet como procedente de otras personas, con la finalidad de obtener un mejor conocimiento sobre las oposiciones

Cuadro 5.6: Riesgo de ausencia de información

<b>Riego</b>	Riesgo-07
<b>Descripción</b>	Algunas tecnologías empleadas tuvieron que ser sustituidas por otras para mejorar el proyecto
<b>Probabilidad</b>	Media
<b>Impacto</b>	Alto
<b>Plan de mitigación</b>	Formación sobre las nuevas tecnologías a utilizar
<b>Plan de contingencia</b>	Disponer de un tiempo extra para realizar tareas añadidas, modificadas o incompletas.

Cuadro 5.7: Riesgo de cambio en las tecnologías utilizadas

<b>Riego</b>	Riesgo-08
<b>Descripción</b>	Se utilizaron los planes gratuitos que ofrece Firebase, pero estos pueden ser insuficientes y es posible que se requiera realizar esfuerzo económico para que el proyecto siga adelante.
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Alto
<b>Plan de mitigación</b>	Previamente se consultaron los planes gratuitos de Firebase, los cuales proporcionan accesos diarios (20.000 y 50.000), límite de escritura (500 documentos por cada escritura) y el tamaño máximo de un documento es de 1,048,487 bytes
<b>Plan de contingencia</b>	Se asumió el coste, ya que era muy bajo. En cuanto a las escrituras, nunca se hace una inserción al día de 500 documentos. Por último, solo hubo un documento que llegaba a dicho límite de tamaño, pero no fue necesario insertarlo, ya que no era relevante en las oposiciones.

Cuadro 5.8: Riesgo de uso de Firebase

<b>Riego</b>	Riesgo-09
<b>Descripción</b>	Las tareas programadas no cumplen la fecha de entrega estimada y el tiempo de desarrollo del proyecto aumenta respecto al esperado inicialmente.
<b>Probabilidad</b>	Media
<b>Impacto</b>	Medio
<b>Plan de mitigación</b>	Se limitaron las tareas y se centró en realizar las tareas principales que forman la parte fundamental del proyecto.
<b>Plan de contingencia</b>	Omitir las tareas complementarias para centrarse en las principales.

Cuadro 5.9: Riesgo de planificación optimista

<b>Riego</b>	Riesgo-10
<b>Descripción</b>	La API no funciona adecuadamente o deja de dar servicio en algún momento
<b>Probabilidad</b>	Muy baja
<b>Impacto</b>	Alto
<b>Plan de mitigación</b>	Se comprueba cada semana si el número de documentos que están en la API se corresponde con el número de documentos publicados en el BOCyL.
<b>Plan de contingencia</b>	Sería necesario crear una función específica para introducir dichos documentos correspondientes con las fechas que faltan.

Cuadro 5.10: Riesgo de API no operativa

<b>Riego</b>	Riesgo-11
<b>Descripción</b>	Schedule Function no funciona correctamente y no introduce documentos en la base de datos
<b>Probabilidad</b>	Baja
<b>Impacto</b>	Muy alto
<b>Plan de mitigación</b>	Se comprueba cada día a través de Firebase si los documentos son recogidos a través de Schedule Function y si son mostradas en la página web.
<b>Plan de contingencia</b>	Sería necesario crear una función específica para introducir dichos documentos correspondientes con las fechas que faltan.

Cuadro 5.11: Riesgo de fallo en Schedule Function

## 5.2. Presupuesto

Este proyecto cuenta con una única persona como desarrollador. Se ha consultado el sueldo base de un programador junior [32] y el valor es de 18.821 euros por año, es decir, 1.333 euros mensuales. Se han invertido aproximadamente 400 horas en el proyecto, por lo que  $400 * 9,42$  (salario promedio por hora) supone un coste final de 3.768 euros.

Además, hay que añadir el valor del hardware utilizado, así como la amortización a lo largo del proyecto. Se ha utilizado un ordenador valorado en 1200 euros. Consultando la tabla de amortizaciones lineales [37] se puede comprobar que el coeficiente línea máximo tiene un valor de un 20%. Como consecuencia, el coste del hardware durante 7 meses será de  $1200 * 0,20 * (7/12)$ , de 140 euros.

Por último, el coste por herramientas utilizadas es de 0 euros (software, licencias, etc). El trabajo realizado a lo largo del proyecto ha sido con herramientas de libre uso y herramientas proporcionadas gratuitas por parte de la universidad.

## Capítulo 6

# Análisis del proyecto

### 6.1. Introducción

Este capítulo se documentará ciertos requisitos que la página web deberá satisfacer así misma (Requisitos funcionales). También se documentarán los requisitos no funcionales. Por último se expondrán los casos de uso.

### 6.2. Requisitos funcionales

Los requisitos funcionales definen las funciones que debe ser capaz de realizar la página web. Se recogen en la tabla 6.1:

### 6.3. Requisitos no funcionales

Los requisitos no funcionales son las propiedades del sistema: rendimiento, seguridad y disponibilidad. También están relacionadas con las restricciones de la página web. Se pueden observar en la tabla 6.2.

### 6.4. Casos de uso

Esta sección resume los casos de uso que puede realizar el usuario al utilizar la aplicación web. La figura 6.1 muestra los casos de uso: buscar oposiciones y acceder documento de oposiciones.

ID	Descripción
RF-1	El sistema deberá ser capaz de mostrar los documentos más recientes al acceder a la página web.
RF-2	El sistema deberá ser capaz de mostrar los documentos de cada etapa de la oposición.
RF-3	El sistema deberá ser capaz de mostrar los documentos con los filtros que aplique el usuario.
RF-4	El sistema deberá mostrar al usuario la información de un documento seleccionado.
RF-5	El sistema deberá permitir al usuario acceder al documento.

Cuadro 6.1: Requisitos funcionales

ID	Descripción
RNF-1	El sistema deberá estar desarrollado y adaptado para funcionar utilizando el framework Vue.js
RNF-2	El sistema deberá acceder a la base de datos de Firebase donde se almacenan los documentos.
RNF-3	El sistema deberá actualizarse cada día (conectividad).
RNF-4	El sistema deberá permitir el acceso desde cualquier equipo conectado a internet.

Cuadro 6.2: Requisitos no funcionales

Buscar oposiciones será un caso de uso y el usuario podrá aplicar filtros para que encuentre el documento que le interesa. Por otro lado, también puede acceder dicho documento, visualizar información relativa al mismo y acceder al archivo a través del enlace PDF.

## 6.5. Diagramas de secuencia

Son múltiples las maneras de utilizar los filtros en la página, pero se ha decidido realizar únicamente dos diagramas de secuencia que resumen el comportamiento de la página web con el usuario.

Primero, nos encontramos con el diagrama de la figura 6.2 que resume el comportamiento de la página web al seleccionar la fase que quiera el usuario. Cuando se selecciona una fase (o

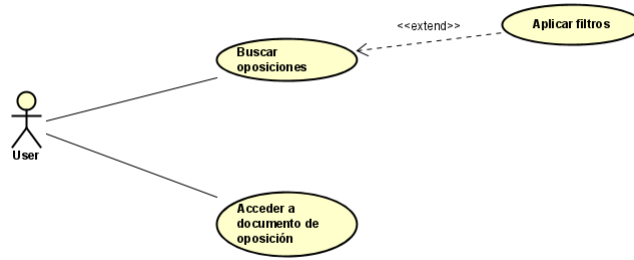


Figura 6.1: Diagrama de casos de uso

nada más acceder a la página) se llama a un método que funciona como refresco y comprueba en qué fase está. Este método hace una petición a la base de datos filtrando por la fase en la que se encuentra el usuario. No obstante, si volvemos a una fase/etapa en la que hemos estado previamente no tendremos que esperar como la primera vez a la petición de la base de datos, ya que se ha cargado ya en caché.

Este comportamiento se repite cada vez que se quiere acceder a una etapa a la que no se ha accedido previamente. Por otro lado, si se ha accedido con anterioridad, se omite la petición a la base de datos.

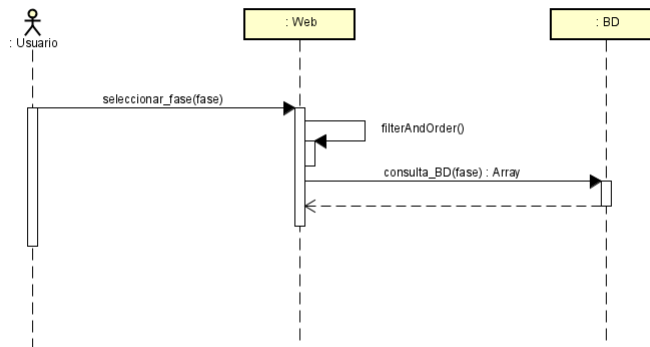


Figura 6.2: Secuencia seleccionando fase

Después de tener los documentos, dependiendo en que fase nos encontremos, se sigue un proceso similar al de la figura 6.3. Se ha supuesto que el usuario utiliza los filtros: localidad y fecha de publicación.

Al utilizar el filtro "localidad", una vez el usuario ha clicado en el botón enter o ha

dado al botón “buscar” se llama inmediatamente al método `filterLocalidad`. Este método filtra los documentos por el lugar donde se oferta la oposición. La localidad escrita por el usuario se guarda en una variable. No se devuelve nada porque este método modifica la lista de documentos (array) que se muestra por pantalla. Una vez se ha realizado el filtro, se muestran los documentos filtrados. Si se hubiera utilizado el filtro “orden” hubiera seguido el mismo proceso.

Al utilizar el filtro “fecha de publicación”, el usuario tiene que escoger dos fechas. Si decide mirar solo un día, tiene que clicar dos veces en el mismo día. Una vez seleccionadas, se activa un watcher que llama al método `filterAndOrder`. Este comprueba, primero, en que fase está el usuario y si es necesario realizar una petición a la base de datos. Después, comprueba qué filtro ha usado el usuario (en este caso, localidad y fecha de publicación). Vuelve a realizar `filterLocalidad` y después realiza `filterFechaPublicacion`. `FilterFechaPublicacion` no devuelve nada, modifica la lista de documentos con la que se está trabajando y muestra los documentos filtrados. Pasaría lo mismo si se utilizará el filtro “fecha de disposición”.

Por último, si se deciden eliminar todos los filtros, se llama inmediatamente al método `resetAllFilters`, que resetea las variables pertenecientes a los filtros y se vuelve a refrescar la página.

Hay que destacar que si se utilizan los filtros “rango”, “organismo”, “orden.” o “palabra clave”, el flujo es el mismo que el filtro “fecha de publicación” (omitiendo el watcher). Una vez se selecciona el rango u organismo, se llama al método `filterAndOrder`. Una vez se ha introducido la orden o palabras claves que quiera el usuario y se da al enter o al botón de “buscar”, se llama al método `filterAndOrder`.

El método `filterAndOrder` es importante ya que es el encargado de detectar en que fase está el usuario y que filtros ha utilizado. A modo de ejemplo, si el usuario ha escogido localidad, rango y fecha de publicación, al usar la fecha de publicación, el método `filterAndOrder` llamará a `filterLocalidad`, `filterRango` y por último `filterFechaPublicación`. Es decir, se repiten los filtros de localidad y rango, pero no importa, porque el tiempo de espera no es muy alto. Si el usuario decide utilizar el botón “limpiar.” en uno de los filtros, se limpiará ese filtro y este llamará a `filterAndOrder` para hacer el proceso ya explicado.



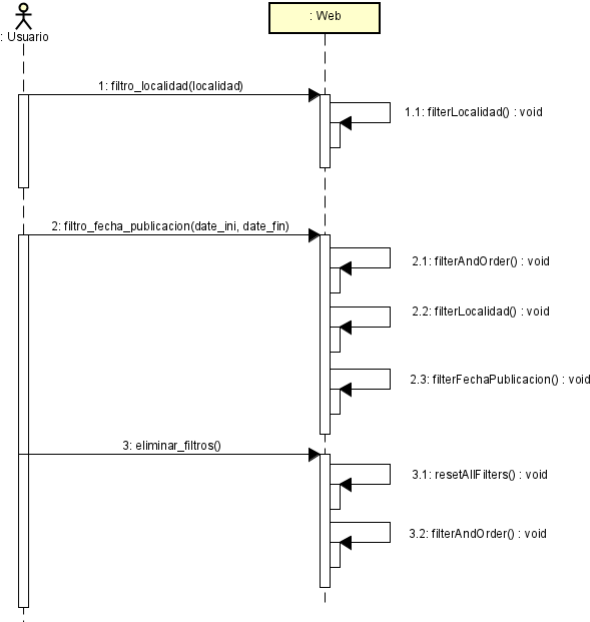


Figura 6.3: Secuencia aplicando filtros



# Capítulo 7

## Diseño

En este capítulo se va a explicar los modelos y patrón utilizados. En lo referente al diseño arquitectónico se ha utilizado el patrón MVVM combinado con el diseño guiado por componentes, ya que se ha utilizado Vue.js. También se explicará brevemente los modelos BaaS y SaaS junto con algunas ventajas y desventajas que nos proporcionan. Por último, se va a hablar tanto de la parte *Front-end* y *Back-End*, siendo ésta última a destacar, debido a la gran parte del tiempo que se le ha dedicado en este proyecto.

### 7.1. BaaS

*Backend-as-a-Service* (BaaS) [5] es una de las arquitecturas cloud más conocidas actualmente. Se basa en ofrecernos una forma de crear fácilmente todo el *Back-End* de nuestras aplicaciones basadas en Cloud Functions (funciones en la nube), las cuales suelen ser, por lo general, funciones escritas en Javascript y que luego BaaS las expone como servicios. De esta forma, en lugar de tener una aplicación con demasiados objetos y procedimientos, tendremos una serie de Cloud Functions que estarán en la nube.

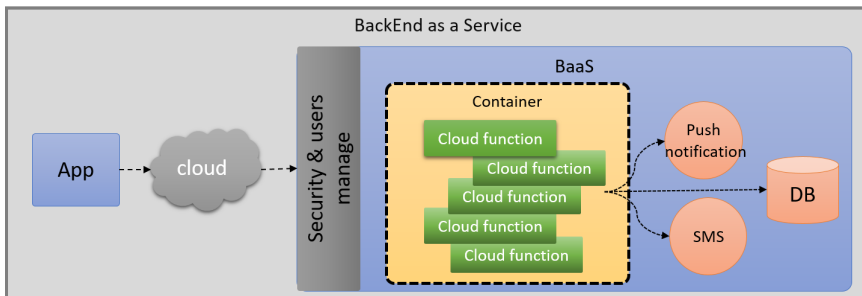


Figura 7.1: Arquitectura Baas

La figura 7.1 resume cómo funciona dicha arquitectura. BaaS ofrece un conjunto de utilerías que ya están implementadas en la nube, por lo que el desarrollador solo tendrá que estar preocupado del desarrollo de las funciones de negocio o servicios. Se basa en la idea de almacenar todo tipo de funciones, estructuras en el lado del servidor en proveedores en la nube y se encarga de ofrecer un correcto mantenimiento, diponibilidad y funcionamiento de los servicios.

De esta manera, el desarrollador podrá quitarse tareas tediosas relacionadas con la seguridad, base de datos sistemas de autenticación, sistemas de archivos, etc. Al utilizar esta arquitectura, hay ventajas y desventajas [6]:

### 7.1.1. Ventajas

- Ofrece funcionalidades tediosas en forma de servicio por lo que ya está implementado en la nube y solo hace falta consumirlo.
- Ofrece almacenamiento en la nube, por lo que se podrá acceder desde cualquier lugar.
- Sus costes de mantenimiento. Es mucho más barato que comprar el equipo, instalarlo y mantenerlo. El proveedor se encarga del servidor cobrando una cuota en función del uso.
- Obtención de *Back-End* completo. Fácil accesibilidad sobre los datos en tiempo reales y ejecución del código en la nube.

### 7.1.2. Desventajas

- Es necesario una conexión a Internet para acceder a los servicios ofrecidos.
- Dependencia del proveedor, tanto por disponibilidad (caída de servidores), pobre rendimiento o funcionamiento inadecuado.
- Menos flexibilidad a la hora de construir *Back-End*.

Como soporte del modelo, se ha utilizado Firebase.

## 7.2. SaaS

*Software-as-a-service* (SaaS) [36] es un modelo de prestación de software basado en la nube, en el cual el proveedor de nube desarrolla y mantiene un software de aplicaciones en la nube. Hace que el software sea accesible para los clientes a través de internet mediante el pago por uso. Es un programa que está ubicado en un servidor externo y se accede a él a través de internet. El proveedor de la nube administra el hardware, middleware, software de las aplicaciones y la seguridad.

Esta tecnología tiene especial importancia en el desarrollo web, ya que permite a los clientes acceder al mismo a través de un navegador. Presenta las siguientes ventajas y desventajas:

### 7.2.1. Ventajas

- Reducir drásticamente costos, así como desplegar, escalar y actualizar las soluciones de negocios más rápido que con mantenimiento de software y sistemas en las instalaciones.
- Acceso desde cualquier punto, siempre y cuando se tenga conexión a internet.
- Facilidad de configuración y personalización.
- Fácil despliegue y puesta en marcha rápida.
- Portabilidad de datos rápida.

### 7.2.2. Desventajas

- El software y datos utilizados se encuentran en el servidor del proveedor por lo que la seguridad y confidencialidad que contengan pueden verse comprometidos
- Requiere de conexión a Internet.

## 7.3. Modelo-Vista-Modelo de vista

En este proyecto se ha utilizado el patrón arquitectónico de software MVVM (Model-View-ViewModel) [16]. Es un método de programación basado en eventos que simplifica la interfaz de usuario. Ayuda a separar la lógica de negocios con la interfaz de usuario, facilitando las pruebas, mantenimiento y escalabilidad de los proyectos. Está formado por 3 partes:

- **Vista (View):** Es la capa de vista, la cual corresponde con la interfaz de usuario. Se compone principalmente de HTML y CSS. Muestra los elementos que contenga al exterior y captura todo tipo de interacciones de los usuarios.
- **Modelo (Model):** Es la capa correspondiente al modelo de datos, generalmente se refiere a varios procesos de lógica de negocio y manipulación de datos realizados en Back-End.
- **Modelo de vista (ViewModel):** Es una capa de vista que es organizada y mantenida por los desarrolladores encargados del Front-End. Se encarga de enlazar la interfaz del usuario con el modelo, conteniendo los métodos necesarios para interactuar con este tras la captura de la interacción por parte de la vista. Cabe destacar que en esta capa se incluye el estado y comportamiento de la vista, mientras que en la capa Modelo solo contiene el estado.

Este patrón es similar al MVC (Modelo-Vista-Controlador) [26], pero tiene una característica principal que los diferencia. El MVVM permite que los datos del modelo se actualicen

automáticamente cuando el usuario produce un cambio al interactuar con la vista sin tener que estar manipulando el DOM, por lo que será más eficiente y menos problemático que el MVC. En la figura 7.2 se puede observar la interacción entre los elementos del patrón MVVM.

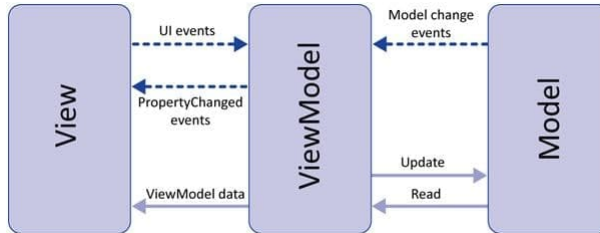


Figura 7.2: Arquitectura MVVM

En este proyecto se ha utilizado Vue.js, aspecto que facilita la utilización del patrón MVVM. Cada instancia actúa como modelo de vista, como se puede apreciar en la figura 7.3. Hay un elemento trascendental en este framework, conocido como binder, el cual actúa entre la vista y el modelo de la vista. Se encarga de actualizar los datos del modelo. Este elemento es bidireccional. Los componentes del patrón aplicados a este framework son:

- **Modelo:** Conjunto de datos que contiene la variables de la sección data y estado de Vuex.
- **Modelo de vista:** conjunto de métodos en la sección computed e instancia Vue.
- **Vista:** sección template.

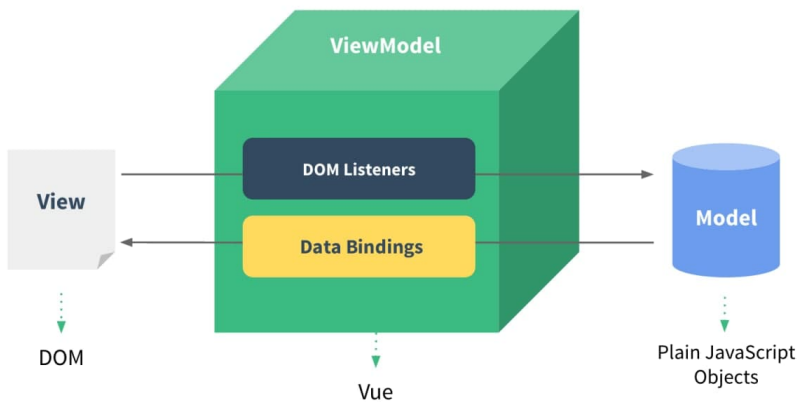


Figura 7.3: MVVM en Vue

La comunicación bidireccional entre los elementos de la vista y los datos del modelo que se realiza a través del binding, está ciertamente relacionado con la reactividad que se utiliza en Vue.js. Se utiliza para conseguirlo:

- **Double mustache binding:** muestra datos en la interfaz de usuario, escribiendo la variable a mostrar entre dobles llaves.
- **Directivas:** atributos especiales [1] que se colocan en las etiquetas HTML y están prefijados por `v-`. Permiten realizar acciones dinámicas potentes que no se pueden realizar en HTML por sí solo.

**Directivas básicas:** Permiten realizar tareas simples. Se ha utilizado *v-model*, permite crear un modelo de datos bidireccional entre un elemento HTML concreto y una variable Vue, es decir, se puede sincronizar el contenido de una variable con el contenido que tenga un elemento HTML en su atributo `value`.

**Directivas condicionales:** permiten realizar acciones según condiciones. Se ha utilizado *v-if*, *v-else*, *v-else-if*, permite generar la interfaz de usuario o no dependiendo del valor que se encuentre en la condición. Si la condición se cumple, se mostrarán los elementos visuales y por tanto la instancia Vue lo tendrá en cuenta. Si no se cumple, la instancia Vue no lo tendrá en cuenta y no se incluirá en el DOM virtual de Vue. También se ha utilizado *v-show*. Es igual que el anterior, solo que si se incluye en el DOM virtual pero no se muestra.

**Directivas de bucles:** Permiten realizar operaciones varias veces. Se ha utilizado *v-for*, permite crear estructuras repetitivas de código HTML de una forma sencilla y sin que el código resulte complejo. Se realiza un bucle en la capa vista del elemento que contenga esta directiva en base a una lista de elementos.

- **Directivas avanzadas:** Permiten realizar tareas más específicas. Se ha utilizado *v-on*, permite gestionar los eventos del DOM desde los templates de vue, haciendo más cómoda y práctica su utilización. En un elemento visual se crea un listener y se especifica un evento y se asocia a un método que controle las acciones que se realicen en dicho elemento.

### 7.3.1. Ventajas

- **Comprobable:** la interfaz siempre ha sido difícil de probar, pero ahora la prueba se puede escribir para el modelo de vista.
- **Reutilizable:** exponer algo en la lógica de vista en un modelo de vista y permitir que muchas vistas reutilicen esta lógica de vista.
- **Independiente:** Los desarrolladores pueden centrarse en la lógica de negocio y desarrollo de datos (modelo de vista), y los diseñadores pueden centrarse en el diseño de páginas (vista).

### 7.4. Diseño guiado por componentes

El diseño guiado por componentes [3] o también conocido como Atomic Web Design, es un patrón de diseño que permite desglosar un sistema software en partes más pequeñas (estas también se pueden desglosar así mismas, y así sucesivamente). Se suele utilizar a la hora de crear una interfaz web. Una de las principales ventajas es la reutilización de los componentes para crear otro más complejos.

#### 7.4.1. Ventajas

- Independientes.
- Reutilizables.
- Fáciles de testear.
- Autocontenidos y encapsulados, cada componente almacena su propio estado interno.
- Aislan errores.

#### 7.4.2. Desventajas

- Aumento de carga para los diseñadores, ya que deben descomponer el sistema y observar las partes que podrán ser reutilizables y convertirlas en componentes
- Aumento de carga en la parte de testing. Con el objetivo de comprobar que todo funcione correctamente, se debe comprobar cada componente individualmente para que a la hora de integrarlo con el resto de componentes tampoco haya problemas.

#### 7.4.3. Componentes creados en este proyecto

En esta sección se van a explicar los componentes creados. No ha sido necesario crear multitud de ellos, ya que la mayor parte del tiempo se ha empleado en investigar las oposiciones y gestionarlas. Esto se ha hecho para que el usuario tenga facilidad a la hora de entender y encontrar el documento que busca. Ha sido realizado acorde con los componentes creados y los elementos que contienen.

Como se puede ver en la figura 7.4 solo han sido necesario crear 5 componentes:

- **App:** Componente que viene ya creado y contiene toda la aplicación web.
- **Principal:** Pantalla que muestra la pantalla principal de la aplicación.
- **InformationOposicion:** Pantalla que muestra información sobre un documento.



- **ListadoOposiciones:** Tabla que permite mostrar los documentos según la fase que se encuentre el usuario. Además, se encuentran una serie de filtros para facilitar la búsqueda del usuario.
- **KeywordTag:** Componente utilizado por ListadoOposiciones que controla la lógica de las etiquetas. Las etiquetas se crean al introducir palabras claves.
- **ContentOposicion:** Componente que contiene información específica de un documento. (Organismo, fecha de publicación, rango y enlaces).

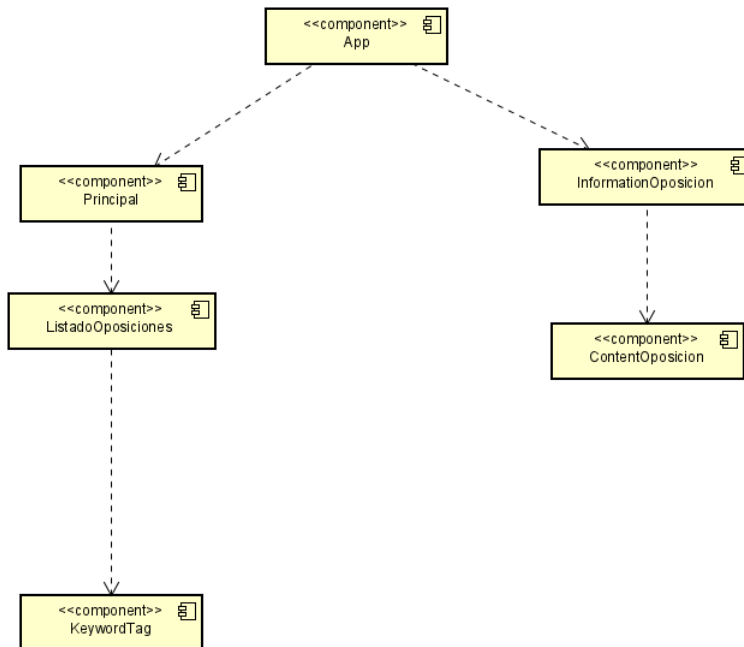


Figura 7.4: Diagrama de componentes

## 7.5. Front-End

En este apartado se explicarán los cambios que se han desarrollado en la interfaz del usuario. Antes de empezar, cabe destacar que se trata de solamente dos vistas, y donde más tiempo se ha consumido es en la vista principal. Desde un principio se ha buscado la sencillez y la simplicidad, con el objetivo que al usuario le sea fácil aprender a usar la página web.

En una primera toma de contacto, el fin de esta captura era mostrar los documentos y algunos campos de interés, como el lugar, puesto de trabajo, organismo, fecha de publicación

## 7.5. FRONT-END

y estado. Como ya se comentó en la sección 4.3.2, a priori no se tenía claro cómo se iban a organizar las oposiciones, ya que la idea principal era enseñar todas y filtrar su puesto o área de trabajo, su organismo y su localidad, pero esa opción no se pudo implementar. Al igual que la columna estado, era muy difícil saber en qué fase se encontraba la oposición en ese momento, ya que no se disponía de la suficiente información para gestionar las oposiciones de manera adecuada.



Lugar	Puesto de trabajo	Organismo	Fecha de publicación	Estado
		AYUNTAMIENTO DE GUILJUELO (SALAMANCA)	2021-03-17	
		AYUNTAMIENTO DE BURGOS	2021-03-17	
		CONSEJERÍA DE CULTURA Y TURISMO	2021-03-17	
		CONSEJERÍA DE SANIDAD	2021-03-17	
		CONSEJERÍA DE SANIDAD	2021-03-17	
		CONSEJERÍA DE SANIDAD	2021-03-17	
		UNIVERSIDAD DE VALLADOLID	2021-03-17	
		CONSEJERÍA DE CULTURA Y TURISMO	2021-03-17	
		CONSEJERÍA DE SANIDAD	2021-03-17	
		AYUNTAMIENTO DE BENAVENTE (ZAMORA)	2021-03-16	

Fin de página 10 de 1-10 de 18 < >

Figura 7.5: Captura de página web (1)

En la siguiente imagen 7.6, se puede ver como se introdujo una serie de tabla en horizontal que permite desplazarnos y obtener todos los documentos que sean de rango: orden, resolución, decreto, acuerdo y otros. Esta tabla se preparó para que cuando se consiguiera realizar un filtro encargado de gestionar las oposiciones por sus fases, se cambiara por el rango. En ese momento, esa tarea de investigación no estaba completada y mucho menos implementada en Back-End.

Rango	Organismo	Fecha de publicación	Estado
ORDEN	CONSEJERÍA DE AGRICULTURA, GANADERÍA Y DESARROLLO RURAL	2021-01-18	
ORDEN	CONSEJERÍA DE AGRICULTURA, GANADERÍA Y DESARROLLO RURAL	2021-01-18	
ORDEN	CONSEJERÍA DE LA PRESIDENCIA	2021-01-26	
ORDEN	CONSEJERÍA DE ECONOMÍA Y HACIENDA	2021-01-06	
ORDEN	CONSEJERÍA DE AGRICULTURA, GANADERÍA Y DESARROLLO RURAL	2021-01-18	
ORDEN	CONSEJERÍA DE FOMENTO Y MEDIO AMBIENTE	2021-01-29	
ORDEN	CONSEJERÍA DE CULTURA Y TURISMO	2021-01-07	
ORDEN	CONSEJERÍA DE CULTURA Y TURISMO	2021-01-26	
ORDEN	CONSEJERÍA DE AGRICULTURA, GANADERÍA Y DESARROLLO RURAL	2021-01-18	
ORDEN	CONSEJERÍA DE FOMENTO Y MEDIO AMBIENTE	2021-01-29	

Figura 7.6: Captura de página web (2)

Se continuó modificando la interfaz de la vista principal y se hicieron avances en Back-End (y, por tanto, también en Front-End), ya que se podría mostrar las oposiciones de diferente manera o empezar con los filtros. En la figura 7.7 se puede ver cómo se modificó la tabla horizontal que permite deslizarse por diferentes etapas y muestra los documentos de las oposiciones que se corresponden con las mismas. Se realizó solo un filtro, una barra buscadora que serviría para comprobar el funcionamiento de la página web.

Rango	Organismo	Fecha de publicación	Estado
OTROS	AYUNTAMIENTO DE SANTA MARÍA DEL PÁRAMO (LEÓN)	2021-01-11	
RESOLUCIÓN	AYUNTAMIENTO DE VALLADOLID	2021-01-14	
ACUERDO	DIPUTACIÓN PROVINCIAL DE PALENCIA	2021-01-12	
OTROS	AYUNTAMIENTO DE SEGOVIA	2021-01-25	
RESOLUCIÓN	AYUNTAMIENTO DE GUJUELO (SALAMANCA)	2021-01-19	
OTROS	AYUNTAMIENTO DE PONFERRADA (LEÓN)	2021-01-28	
RESOLUCIÓN	AYUNTAMIENTO DE CANDELEDA (ÁVILA)	2021-01-18	
ACUERDO	AYUNTAMIENTO DE CIUDAD RODRIGO (SALAMANCA)	2021-01-29	
ACUERDO	AYUNTAMIENTO DE MEDINA DEL CAMPO (VALLADOLID)	2021-01-14	
RESOLUCIÓN	AYUNTAMIENTO DE SALAMANCA	2021-01-26	

Figura 7.7: Captura de página web (3)

Algunos de los cambios más importantes de la vista principal se pueden encontrar en la figura 7.8, comparándola con la 7.7. Uno de los más radicales fue la absorción de algunas etapas (previsto y bases). También se modificó el apartado con nombre “resultados” por

“definitivo”. Se realizó una nueva tabla que mostrara campos que fueran de verdadero interés y permitiera agilizar la búsqueda (localidad, fecha de disposición, rango, organismo, fecha de publicación, orden y palabra clave). Algunos de ellos no requieren de explicación, ya que se sabe como funcionan (localidad, organismo y fecha de publicación). Sin embargo, existen otros que si que lo necesitan:

**Fecha de disposición:** Se pueden consultar los documentos por su fecha de disposición. Ésta hace referencia al acto de reunión donde se fijan los contenidos que integran los documentos que se publican.

**Rango:** Se puede consultar los documentos por el tipo de documento que sea (orden, resolución, decreto, acuerdo y otros). Normalmente, una oposición suele abrir convocatoria cuando es de tipo orden o resolución, pero también existen algunas que abren convocatoria con decreto u acuerdo. En otros podemos encontrar modificaciones, correcciones, etc.

**Orden:** Permite encontrar la orden de una oposición. Dicha orden sirve como identificador, el cual se establece cuando una oposición publica su convocatoria. Este filtro va orientado a conocer los resultados de una convocatoria, tanto resultados provisionales, como definitivos. También se puede utilizar para conocer si se se ha modificado dicha convocatoria, ya que harán referencia a dicha orden.

**Palabra clave:** Permite consultar todos los documentos que contengan esas palabras claves. Este filtro va orientado a lo que se quería conseguir en la figura 7.5 en la columna ” puesto de trabajo”. Al escribir el puesto de trabajo/grupo/palabras deseado se encuentran todos los documentos que lo contengan en cualquier parte del documento.



Localidad	Fecha de disposición	Rango	Organismo	Fecha de publicación	Orden	Palabra clave
Valladolid	2021-04-14	ORDEN	CONSEJERÍA DE SANIDAD	2021-04-30		
Valladolid	2021-01-21	RESOLUCIÓN	CONSEJERÍA DE SANIDAD	2021-01-28		

Figura 7.8: Captura de página web (4)

Por último, la vista principal (figura 7.9) y la vista que muestra la información del documento (figura 7.10) han quedado de la siguiente manera:

Localidad	Fecha de publicación	Rango	Operativa	Fecha de publicación	Orden	Página clave
Villa de Cogol	2021-04-26	RESOLUCIÓN	AYUNTAMIENTO DE VIANA DE CEGA (VALLADOLID)	2021-05-10		
Villabona	2021-04-07	RESOLUCIÓN	AYUNTAMIENTO DE VALLADOLID	2021-04-14		
Brihueca	2021-03-31	ACUERDO	AYUNTAMIENTO DE BRIHUESCA (BURGOS)	2021-04-09		
Alba de Tormes	2021-03-23	RESOLUCIÓN	AYUNTAMIENTO DE ALBA DE TORMES (SALAMANCA)	2021-04-06		
Medina de Rioseco	2021-03-15	DECRETO	AYUNTAMIENTO DE MEDINA DE RIOSECO (VALLADOLID)	2021-03-22		
Bofar	2021-04-20	RESOLUCIÓN	AYUNTAMIENTO DE BOFAR (LEÓN)	2021-04-28		
Peñaranda de Bracamonte	2021-04-29	DECRETO	AYUNTAMIENTO DE PEÑARANDA DE BRACAMONTE (SALAMANCA)	2021-05-06		
Buzillo del Páramo	2021-05-07	RESOLUCIÓN	AYUNTAMIENTO DE BUZILLO DEL PÁRAMO (LEÓN)	2021-05-19		
Salas de los Infantes	2021-08-24	RESOLUCIÓN	AYUNTAMIENTO DE SALAS DE LOS INFANTES (BURGOS)	2021-08-31		
Castrizuelo	2021-01-22	RESOLUCIÓN	AYUNTAMIENTO DE CASTRIZUELO (BURGOS)	2021-02-03		

Figura 7.9: Vista principal final

OPCYL

AYUNTAMIENTO DE SALAS DE LOS INFANTES (BURGOS)

Fecha de publicación: 2021-08-31

Rango: RESOLUCIÓN

Enlace PDF: <https://opcy.es/boletines/2021/08/31/mes/BOCYL-03-1082021-17.pdf>

Enlace HTML: <https://opcy.es/boletines/2021/08/31/mes/BOCYL-03-1082021-17.htm>

Figura 7.10: Vista de información de un documento final

## 7.6. Back-End

En este apartado se va explicar brevemente el contenido Back-End. Como base de datos se tuvo Firestore de Firebase para sus Cloud Functions [12]. Para empezar, se explicarán las functions realizadas para ingresar los documentos cada mes, después la functions schedule, los métodos realizados en esta parte y por último el modelo de despliegue de la web.

### 7.6.1. Functions

Se han utilizado funciones que se activan mediante solicitud HTTPS para ingresar los documentos de cada mes. De esta manera, que si se realiza la petición HTTPS desde cualquier lugar, la function realizará su contenido, en este caso, ingresará los documentos de un documento en la base de datos específica.

Se podía haber realizado dos meses en una función, pero existía una restricción en Firebase: la cantidad máxima de operaciones de escritura que se pueden pasar a una operación *commit* o realizar en una transacción (500). Normalmente, cada mes suele tener aproximadamente unos 100 documentos (la media entre enero y julio es de 132 documentos) pero se decidió proceder de esta manera debido a la gran modularidad que supone, ya que no siempre se ha trabajado en una base de datos en concreto. En primer lugar se creó una base de datos por cada mes para analizar dichos documentos y después de ese análisis (a continuación se explicará en que consistió este), se introducían en una base de datos en común.

Tienen el siguiente formato:

```
exports.mesX = functions.https.onRequest((req, res) => {
  // ...
});
```

### 7.6.2. Schedule Function

Se trata de una función igual que la explicada anteriormente, pero de diferente funcionamiento. Una *schedule function* [schedule function] se utiliza para programar funciones y ejecutarlas en un determinado momento.

Esta función se va a encargar de comprobar si se han introducido nuevos documentos en la API procedente del Boletín Oficial de Castilla y León (BOCyL), realizará el análisis de cada documento que estará en un método, y por último los introducirá a la base de datos. A modo de resumen los pasos que sigue son los siguientes:

1. **Conseguir documentos:** Esta operación se consigue con *axios*, utilizando en la cabecera el token registrado para poder realizar dicha petición a la API. Es posible que no existan documentos, en ese caso se sigue realizando la función. La URL cambia diariamente, ya que cada día hace una petición para conseguir los documentos del día anterior (día en el que se suben los documentos a la API).
2. **Recorrer cada documento:** Esta operación se realiza con el propósito de analizar (se explicará a continuación) el documento con el que se trata. Esto se realiza pasándolo por un método que después acumula la inserción del documento y, finalmente, cuando ya se han recorrido todos los elementos, se hacen todas las transacciones (inserciones) en un lote.

Es importante conocer que por cada acumulación en la inserción de un documento, si este contiene la localidad en un campo, se ingresará con la siguiente estructura:

- id
- organismo
- fecha\_publicacion
- rango

- enlace\_fichero\_xml
- enlace\_fichero\_pdf
- titulo
- localidad
- fecha
- tipo
- JSON

Si no viniera expuesto dicho campo (localidad) se ingresará el documento con la misma estructura a excepción de localidad y fecha.

3. **Inserción de documentos:** Se han utilizado las escrituras por lotes, primero creando batch, después acumulando la transacción de inserción de cada documento con batch.set y finalmente realizando la escritura en un lote con batch.commit

El formato de esta function es de la siguiente manera:

```
exports.scheduledFunction = functions.pubsub.schedule('every 5 minutes').onRun((context) => {
  // ...
});
```

### 7.6.3. Métodos

Se han creado dos métodos además de las functions, con el objetivo de modularizar. Se explicarán a continuación:

**GetJson:** Se pasa por parámetro una URL la cual será sobre la que se tiene que hacer otra petición utilizando axios, ya que servirá para devolver el contenido del documento en un objeto JSON. Este proceso se explicó en la sección 4.3.7 (el parseador).

**Clasificar:** Se pasa por parametro un texto, el cual es el título del documento. Este texto se analizará utilizando expresiones regulares y dependiendo del título se clasificará como aprobado, abierto, provisional, definitivo o modificaciones. A modo de ejemplo, las expresiones regulares utilizadas para modificaciones y aprobado son las siguientes:

```
let modificaciones =/(corrección|modifica|concede|corrige|acuerda.*amplia|aprueba.*modificación|(amplia|suspende).*plazo|rectifica|nuev[o,a].*(inscripciones|puntuaci|relativa.*(concurso|proce)))/;

let aprobado =/(apr[ueo]{1,2}|anuncia|p.blica)(?!.*(modificación|bases|convocatoria|aspirantes|lista|definitiv)).*oferta/;
```

Se han utilizado operadores OR en modificaciones.

- .: el punto significa cualquier carácter.

- \*: el asterisco significa que el carácter precedente puede ser cero o muchos.
- **(contenido)**: todo lo que esté dentro de los paréntesis es un contenido exacto. Como se puede observar, la más compleja es la expresión: nuev[o,a].\*(incipciones—puntuaci). Esta dará verdadero si encuentra las expersión; nuevo/a (cualquier caracteres)incipciones o puntuaci.

En Aprobado, los delimitadores son los siguientes:

- [ueo]{1,2}: de los caracteres u,e,o puede contener uno o dos. Aprue o apro.
- ?!.\*(**palabras prohibidas**): todas las palabras que estén dentro del paréntesis se consideran prohibidas. Por ello, todo lo que esté después de "se aprueba la convocatoria pública" (palabras prohibidas) no dará verdadero. Esto se hace para realizar excepciones.
- .\*: cualquier carácter, puede ser de cero a muchos.

Una vez realizadas las expresiones regulares se comprobará cada una de ellas con el título (que es lo que se ha pasado por parámetro). Si, por ejemplo, se obtiene "verdadero.<sup>en</sup> aprobado, devolverá un string con "APROBADOz por lo tanto sabremos que dicho documento está en la etapa aprobado.

### 7.6.4. Modelo de despliegue de la web

En este apartado se va a explicar brevemente el modelo de despliegue de la web.

Se ha utilizado el servicio de hosting de Firebase [15]. Para realizarlo se siguieron los pasos que se comentan en la sección B.2. Los archivos que son necesarios van a estar en el servidor de hosting de firebase. La base de datos es la que proporciona firebase (firestore) y la estructura del modelo de despliegue de la web se puede ver en la figura 7.11.



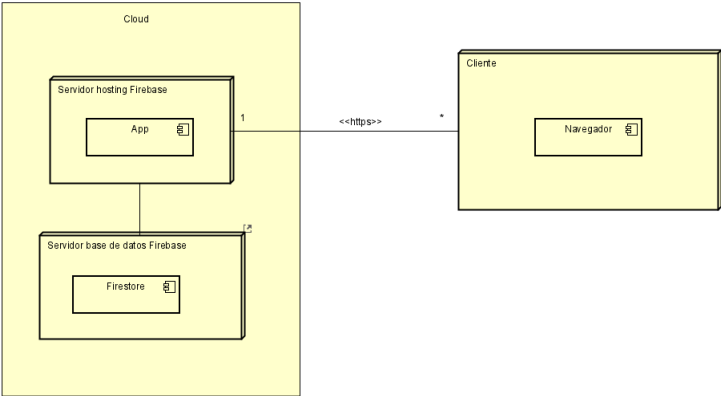


Figura 7.11: Modelo de despliegue de página web



## Capítulo 8

# Implementación

En este capítulo se va a explicar detalladamente la estructura del código y los cambios realizados a lo largo del proyecto.

### 8.1. Estructura del código del proyecto

Los ficheros del proyecto tienen la siguiente estructura:

```
| .browserslistrc
| .eslintrc.js
| .firebaserc
| .gitignore
| babel.config.js
| firebase.json
| jest.config.js
| package.json
| README.md
|
+---public
|   index.html
|   logo-BOCyL.png
|
+---node_modules (se omite su contenido por brevedad)
|
+---functions
|   +---node_modules (se omite su contenido por brevedad)
|   |
|   +---.gitignore
|   |
|   +---index.js
```

```
| |
| | +---package-lock.json
| | |
| | +---package.json
| |
+---src
| |   App.vue
| |   main.js
| |
| | +---assets
| | |
| | +---components
| | |   +---ContentOposicion.vue
| | |   |
| | |   +---KeywordTag.vue
| | |   |
| | |   +---ListOposiciones.vue
| | |
| |
| | +---plugins
| | |   firebase.js
| | |   vuetify.js
| | |
| | +---router
| | |   index.js
| | |
| | +---store
| | |   |   index.js
| | |
| | +---views
| | |   +---InformationOposicion.vue
| | |   +---Principal.vue
| |
| |
```

Los archivos que están colocados al principio (browserlistrc, env, eslintrc, firebase, babel.config.js, package-lock.json y package.json) están relacionados con la configuración de la página web.

El archivo gitignore realiza la tarea de ignorar los archivos que estén contenidos en él a la hora de realizar las subidas del repositorio de Git. Los archivos que están incluidos son node\_modules (dependencias del proyecto y módulos) y dist (archivo que se utiliza para desplegar la página web).

A continuación se van a detallar el contenido de los archivos que forman la estructura del código:

- **public:** Contiene un fichero index que se utiliza como punto de entrada a la aplicación y el logo de la aplicación web.

- **src**: Es el directorio más importante, ya que contiene todo el código fuente de la aplicación web. En la raíz de esta carpeta están los archivos `App.vue` y `main.js`. `App` es el componente principal y `main` es el archivo encargado de lanzar la aplicación y crear la instancia Vue inicial.
  - **assets**: Archivos estáticos (por ejemplo, imágenes) que se utilizan a lo largo de la aplicación web.
  - **components**: Archivos que contienen todos los componentes de la página web.
  - **plugins**: Archivos de configuración de los diferentes plugins utilizados. En este proyecto se han utilizado `firebase` (`firebase.js`) y `Vuetify` (`vuetify.js`)
  - **router**: Archivos de navegación de la aplicación web. Sirven para desplazarse por las diferentes vistas que existen en el proyecto. El fichero `index.js` contiene información sobre estas (ruta, componente...etc).
  - **store**: Archivos encargados de gestionar los estados de la aplicación.
  - **views**: Archivos que contienen los componentes de la aplicación web pero funcionan a modo de vista.
- **functions**: carpeta necesaria para realizar las functions. Dicho servicio es ofrecido por `firebase`. Es necesario que tenga su `node_modules` independiente, por lo que tendrá paquetes y dependencias totalmente diferentes a la aplicación web. También existen archivos de configuración y otro `gitignore`. Por último y de gran relevancia, está el archivo `index.js`, donde están todas las functions que queremos utilizar a lo largo del proyecto.

## 8.2. Decisiones a lo largo del proyecto

A lo largo del proyecto se han tomado decisiones importantes que caben ser resaltadas. Se comentarán algunas de ellas a continuación.

Una de las primeras decisiones fue escoger como framework `Vue.js` en la parte de *Front-End*. Esto se debió a su curva de aprendizaje y por la facilidad a la hora de encontrar problemas que puedan surgir. Existe una gran comunidad para solventar estos. También se decidió utilizar este framework porque se utilizó con anterioridad en una de las asignaturas cursadas durante el curso. Por otro lado, se hizo uso de `firebase` como base de datos en gran parte del proyecto por el mismo motivo que `Vue`. `firebase` aporta flexibilidad y facilidad de uso. Aparte, existen servicios que han sido de gran utilidad (ver sección 7.6).

Otra decisión clave fue utilizar la API perteneciente al `BOCyL`, debido a la comodidad y facilidad que aportaba esta. En la sección 4.3.2 se explica por qué se escoge dicha API como única fuente. No obstante, cabe destacar que esto conlleva riesgos, ya que la aplicación web depende de esta a la hora de actualizar cada día.

A la hora de crear las expresiones regulares, se decidió dejar de lado algunos documentos relacionados con nombramientos de tribunales y comisiones, acuerdos que nombran comisiones de valoración, sustituciones de vocales en los tribunales calificadoros de las oposiciones

y derivados de estas. Se decidió que no se mostrarían porque no son los documentos que nuestro tipo de usuario busca.

## 8.3. Cambios realizados a lo largo del proyecto.

A lo largo de los *Sprints* se fue modificando (en su mayor parte) la vista principal que muestra los documentos. Estos cambios se pueden apreciar en la sección 7.5.

Se produjeron más cambios en la parte de *Back-End*, en la *schedule function*, ya que esta no funcionaba correctamente al realizar la operación de insertar los datos en la base de datos. Es decir, recogía todos los documentos correctamente (los conseguía a través de la petición), pero no insertaba. Se decidió implementar una función que fuera llamada desde la *schedule function* para solventar dicho problema.

## 8.4. Licencia

Se ha utilizado la licencia MIT [25] en este proyecto. Licencia reconocida como The Open Source Initiative (OSI) que cumple con la definición de código abierto. Es compatible con licencias BSD [23] y [24].

## Capítulo 9

# Conclusiones y futuro

### 9.1. Conclusiones

Finalizando la memoria, se exponen unas breves conclusiones y el trabajo que queda para el futuro.

Una de las principales conclusiones obtenidas tras el desarrollo del proyecto, es que un trabajo de este tipo, puede ser difícil de realizar por una sola persona. Se echa en falta el trabajo en equipo, tener un compañero para debatir las decisiones a tomar o para pedir ayuda.

En rasgos generales, he experimentado las dificultades de sacar adelante proyecto. Tienes que asegurarte que toda la funcionalidad se ejecute correctamente, contar con errores imprevistos e imaginarte que pueda pasar cualquier riesgo por poco probable que parezca

También es importante tener una buena planificación para sacar adelante el proyecto. Marcarse objetivos por cada *Sprint*, no solo a nivel de completar la tarea, sino personal. Marcarse plazos y horarios que nos sirvan para no desviarnos con tareas secundarias y hacer las que son críticas.

El resultado de la página web es satisfactorio, principalmente para aquellas personas que encuentren la oposición que buscan desesperadamente y solo situándose en la fase abierto, ya tienen todo tipo de oposiciones a su disposición. No obstante, la fase de modificaciones es también importante, ya que muestra las oposiciones que se han vuelto a abrir.

Ha habido problemas con los tiempos de espera en el primer despliegue, pero se consiguieron bajar considerablemente.

## 9.2. Trabajo futuro

A lo largo del proyecto se han observado varias mejoras que podrían implementarse en un futuro:

- La página, en ocasiones falla, y hay que cerrar la ventana y volver a meterse en ella debido a un problema relacionado con la memoria. La página se pausa antes de un posible colapso de memoria.
- Página web responsive. Actualmente solo funciona con ordenadores, si se accede desde un dispositivo móvil, no funciona correctamente.
- Creación de base de datos con usuario. Se podrían introducir usuarios con su respectivo perfil para que puedan guardar los documentos que les interesen o que salten alertas según sus intereses.
- Disminuir tiempo de espera.
- Modificar el enlace URL para acceder a la página web.



# Apéndice A

## Manual de usuario

En este capítulo se va a explicar cómo utilizar la página web. Se centrará principalmente en la primera vista, ya que es la más importante y donde se gestionan y se filtran los documentos. Antes de comenzar, se puede acceder a la página desde la dirección que se encuentra en la sección B.2, pero es totalmente necesario que se acceda desde un ordenador.

Gestor y localizador de oposiciones en Castilla y León

Esta página es una herramienta que se ha creado con el fin de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son gestionadas por las etapas que pelean con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECIENTES APROBADO ABIERTO PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de convocatoria ?	Rango ?	Organismo ?	Fecha de publicación ?	Orden ?	Palabra clave ?
Valladolid	2021-10-07	ORDEN	CONSEJERÍA DE EMPLEO E INDUSTRIA	2021-10-11		
Valladolid	2021-10-06	ORDEN	CONSEJERÍA DE EMPLEO E INDUSTRIA	2021-10-11		
Valladolid	2021-09-30	ORDEN	CONSEJERÍA DE LA PRESIDENCIA	2021-10-11		
Estájar	2021-10-04	ACUERDO	AYUNTAMIENTO DE ESTÉJAR (BURGOS)	2021-10-11		
Salamanca	2021-10-04	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-10-11		
Valladolid	2021-09-29	ORDEN	CONSEJERÍA DE LA PRESIDENCIA	2021-10-09		
Valladolid	2021-09-23	ORDEN	CONSEJERÍA DE SANIDAD	2021-10-09		
Valladolid	2021-09-23	ORDEN	CONSEJERÍA DE SANIDAD	2021-10-09		
Valladolid	2021-09-23	RESOLUCIÓN	CONSEJERÍA DE SANIDAD	2021-10-09		
Valladolid	2021-09-23	ORDEN	CONSEJERÍA DE SANIDAD	2021-10-09		

Filtros por página: 10 1-10 de 58

ELIMINAR TODOS LOS FILTROS

Figura A.1: Vista principal de la página web

Como se puede apreciar en la figura A.1 hay un título y una breve descripción acerca de qué trata la página web. En la parte superior de la tabla, nos encontramos con una serie de etapas que gestionan los documentos de las oposiciones del Boletín Oficial de Castilla y León.

Al principio, la página tardará muy poco en cargar, ya que en el apartado correspondiente a “recientes” hay pocos documentos. Sin embargo, el tiempo de espera variará en cada etapa dependiendo de la cantidad de documentos que haya en cada una de ellas. El peor caso es la fase “abierto” debido a que hace una petición a la base de datos por cada etapa. No obstante, esto solo ocurrirá una vez, y si posteriormente se vuelve a la misma etapa, el tiempo de espera será considerablemente bajo.

Cada etapa tiene un tooltip que explica el contenido de la misma:

- **Recientes:** Son los cincuenta últimos documentos recogidos en la API del BOCyL. No es una etapa, pero sirve para que el usuario pueda comprobar cuáles son los últimos documentos publicados.

Localidad ?	Fecha de publicación ?	Rango ?	Organismo ?	Fecha de publicación ?	Orden ?	Título clave ?
Vilagor	2021-10-07	ORDEN	CONSEJERÍA DE EMPLEO E INDUSTRIA	2021-10-11		
Vilagor	2021-10-06	ORDEN	CONSEJERÍA DE EMPLEO E INDUSTRIA	2021-10-11		
Vilagor	2021-09-30	ORDEN	CONSEJERÍA DE LA PRESIDENCIA	2021-10-11		
Zalduer	2021-10-04	ACUERDO	AYUNTAMIENTO DE ESTREPAR (BURGOS)	2021-10-11		
Salamanca	2021-10-04	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-10-11		
Vilagor	2021-09-29	ORDEN	CONSEJERÍA DE LA PRESIDENCIA	2021-10-08		
Vilagor	2021-09-23	ORDEN	CONSEJERÍA DE SANIDAD	2021-10-08		
Vilagor	2021-09-23	ORDEN	CONSEJERÍA DE SANIDAD	2021-10-08		
Vilagor	2021-09-23	RESOLUCIÓN	CONSEJERÍA DE SANIDAD	2021-10-08		
Vilagor	2021-09-23	ORDEN	CONSEJERÍA DE SANIDAD	2021-10-08		

Figura A.2: Últimos documentos publicados por el BOCyL

- **Aprobado:** Son todos los documentos relacionados con las ofertas de empleo aprobadas. En un documento nos encontraremos generalmente con la aprobación de una oferta de empleo. Esta oferta puede contener varias plazas de puestos de trabajo.

Esta etapa sirve para conocer los puestos que se ofertarán en próximas convocatorias.

OPCYL

Gestor y localizador de oposiciones en Castilla y León

Esta página tiene los documentos relacionados con las fases de trabajo aprobadas de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son gestionadas por las entes que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECIENTES APROBADO ABIERTO PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de expedición ?	Trámite ?	Organismo ?	Fecha de publicación ?	Otros ?	Palabra clave ?
Valsa de Cega	2021-04-26	RESOLUCIÓN	AYUNTAMIENTO DE VIANA DE CESA (VALLADOLID)	2021-05-10		
Valsestán	2021-04-07	RESOLUCIÓN	AYUNTAMIENTO DE VALLADOLID	2021-04-14		
Bruñaco	2021-03-31	AQUERDO	AYUNTAMIENTO DE BRIVIESCA (BURGOS)	2021-04-09		
Alba de Tormes	2021-03-25	RESOLUCIÓN	AYUNTAMIENTO DE ALBA DE TORMES (SALAMANCA)	2021-04-06		
Medina de Rioseco	2021-03-15	DECRETO	AYUNTAMIENTO DE MEDINA DE RIOSECO (VALLADOLID)	2021-03-22		
Bolar	2021-04-20	RESOLUCIÓN	AYUNTAMIENTO DE BUÑAN (LEÓN)	2021-04-28		
Peñacande Bracamonte	2021-04-29	DECRETO	AYUNTAMIENTO DE PEÑACANDE DE BRACAMONTE (SALAMANCA)	2021-05-05		
Bustillo del Páramo	2021-05-07	RESOLUCIÓN	AYUNTAMIENTO DE BUSTILLO DEL PÁRAMO (LEÓN)	2021-05-19		
Salas de los Infantes	2021-08-24	RESOLUCIÓN	AYUNTAMIENTO DE SALAS DE LOS INFANTES (BURGOS)	2021-08-31		
Castrojeriz	2021-01-22	RESOLUCIÓN	AYUNTAMIENTO DE CASTROJERIZ (BURGOS)	2021-02-03		

Eliminar todos los filtros

Filas por página: 10 110 de 67

Figura A.3: Todos los documentos que se encuentran en la fase aprobado

- Abierto:** Son todos los documentos relacionados con las convocatorias de oposiciones abiertas. Generalmente, suele abrirse una convocatoria para un puesto de trabajo, pero en ocasiones se puede encontrar que en una convocatoria se ofrezcan varios puestos de trabajo.

Esta etapa sirve para avisar al usuario de que dispone, normalmente, de entre 15 y 20 días para realizar la solicitud de inscripción. En esta etapa, los documentos suelen mostrar las bases (en las que viene la nacionalidad, edad, titulación, capacidad, habilitación, etc.), formación necesaria, cursos de formación requeridos y otros requisitos que deberá aportar el candidato (fotocopia del DNI, por ejemplo). Además, en ocasiones suele venir en un anexo la solicitud de admisión a la convocatoria.

**OPCYL**

**Gestor y localizador de oposiciones en Castilla y León**

Esta página es una herramienta que se ha diseñado para facilitar el acceso a los documentos relacionados con las oposiciones en el territorio de Castilla y León. Los documentos son gestionados por las etapas que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECEBIDOS APROBADO **ABIERTO** PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ? Fecha de expedición ? Tipo ? Organismo ? Fecha de publicación ? Orden ? Fecha clave ?

Localidad	Fecha de expedición	Tipo	Organismo	Fecha de publicación	Orden	Fecha clave
Salamanca	2021-02-01	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-02-15		
Vilagorria	2021-02-01	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-02-10		
León	2021-01-19	RESOLUCIÓN	UNIVERSIDAD DE LEÓN	2021-01-28		
Vilagorria	2021-04-20	ORDEN	CONSEJERIA DE EDUCACIÓN	2021-05-04		
Palencia	2021-03-29	RESOLUCIÓN	DEPUTACION PROVINCIAL DE PALENCIA	2021-04-09		
Vilagorria	2020-12-21	ORDEN	CONSEJERIA DE ECONOMÍA Y HACIENDA	2021-01-08		
Vilagorria	2021-06-08	ORDEN	CONSEJERIA DE ECONOMÍA Y HACIENDA	2021-06-21		
Vilagorria	2021-03-01	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-03-10		
Vilagorria	2021-05-06	ORDEN	CONSEJERIA DE CULTURA Y TURISMO	2021-05-18		
Vilagorria	2021-09-10	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-09-24		

Filas por página: 10 1-10 de 455

**ELIMINAR TODOS LOS FILTROS**

Figura A.4: Todos los documentos que se encuentran en la fase abierto

- Provisional:** Permite visualizar todos los documentos relacionados con los resultados provisionales de las oposiciones. En esta etapa nos podemos encontrar con lista de admitivos y excluidos provisionales, así como con resultados que no son definitivos por el momento.

Esta etapa sirve para subsanar las razones que hayan motivado a la exclusión del usuario en la admisión (por ejemplo, falta de documentación requerida). Tiene un plazo de unos 10 o 15 días desde la publicación del documento para ello. En este mismo plazo también se podrán presentar alegaciones contra el órgano emisor del documento.

**OPCYL**

**Gestor y localizador de oposiciones en Castilla y León**

Esta página es una herramienta que se ha diseñado para facilitar el acceso a los documentos relacionados con las oposiciones en el territorio de Castilla y León. Los documentos son gestionados por las etapas que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECEBIDOS APROBADO **PROVISIONAL** DEFINITIVO MODIFICACIONES OTROS

Localidad ? Fecha de expedición ? Tipo ? Organismo ? Fecha de publicación ? Orden ? Fecha clave ?

Localidad	Fecha de expedición	Tipo	Organismo	Fecha de publicación	Orden	Fecha clave
Salamanca	2021-04-06	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-04-13		
León	2021-07-26	RESOLUCIÓN	UNIVERSIDAD DE LEÓN	2021-08-03		
Vilagorria	2021-07-01	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-07-09		
Vilagorria	2021-07-01	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-07-09		
Vilagorria	2021-05-12	ORDEN	CONSEJERIA DE LA PRESIDENCIA	2021-05-19		
Vilagorria	2021-06-14	RESOLUCIÓN	CONSEJERIA DE FAMILIA E IGUALDAD DE OPORTUNIDADES	2021-06-23		
León	2021-06-01	RESOLUCIÓN	UNIVERSIDAD DE LEÓN	2021-06-07		
Vilagorria	2021-02-16	RESOLUCIÓN	CONSEJERIA DE AGRICULTURA, GANADERIA Y DESARROLLO RURAL	2021-02-26		
León	2021-07-26	RESOLUCIÓN	UNIVERSIDAD DE LEÓN	2021-08-03		
Vilagorria	2021-09-23	ORDEN	CONSEJERIA DE SANIDAD	2021-10-08		

Filas por página: 10 1-10 de 126

**ELIMINAR TODOS LOS FILTROS**

Figura A.5: Todos los documentos que se encuentran en la fase provisional

- Definitivo:** Permite visualizar todos los documentos relacionados con los resultados definitivos de las oposiciones. En esta etapa nos podemos encontrar con listas de admitidos y excluidos definitivos, así como resultados que son definitivos. En esta etapa se abre un período de entre 10 días y 1 mes, aproximadamente, después de la publicación del documento, en el que el usuario podrá interponer un recurso de alzada (o cualquier otro, por ejemplo contencioso-administrativo), el cual significa que el órgano administrativo tiene que revisar el acto dictado por otro órgano dependiente jerárquicamente de él.

Localidad	Fecha de publicación	Resolución	Órgano	Fecha de disposición
Palencia	2021-03-04	RESOLUCIÓN	CONSEJERÍA DE SANIDAD	2021-03-15
Valladolid	2021-07-27	RESOLUCIÓN	CONSEJERÍA DE EDUCACIÓN	2021-07-30
Valladolid	2021-01-07	RESOLUCIÓN	CONSEJERÍA DE EDUCACIÓN	2021-01-15
Valladolid	2021-05-17	RESOLUCIÓN	CONSEJERÍA DE EDUCACIÓN	2021-05-28
Valladolid	2021-07-20	RESOLUCIÓN	CONSEJERÍA DE FOMENTO Y MEDIO AMBIENTE	2021-07-28
Valladolid	2021-05-26	RESOLUCIÓN	CONSEJERÍA DE EDUCACIÓN	2021-06-02
Boria	2021-06-02	RESOLUCIÓN	CONSEJERÍA DE SANIDAD	2021-06-09
Valladolid	2021-03-16	RESOLUCIÓN	CONSEJERÍA DE ECONOMÍA Y HACIENDA	2021-03-24
Valladolid	2021-04-05	RESOLUCIÓN	CONSEJERÍA DE EMPLEO E INDUSTRIA	2021-04-15
Sabániga	2021-06-02	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-06-09

Figura A.6: Todos los documentos que se encuentran en la fase definitivo

- Modificaciones:** Muestra todos los documentos que modifican otros documentos. En esta etapa nos podemos encontrar con correcciones, ampliaciones de plazo, modificaciones de listas de admitidos y excluidos, resultados, etc. Esta etapa es de importante interés ya que puede mostrar cambios relevantes en los documentos previamente publicados (por ejemplo, una convocatoria que ya se cerró se vuelve a abrir).

En esta etapa se recomienda no utilizar siempre el filtro de localidad, ya que en la API no se encontró dicho campo en algunos documentos (y por eso aparece vacío). Esto mismo ocurre con la fecha de disposición.

OPCyL

Gestor y localizador de oposiciones en Castilla y León

Esta página es una herramienta que se ha creado con el fin de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son generadas por las entes que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

Todos los documentos relacionados con convocatorias, resoluciones de plaza, modificaciones de bases de acceso y exámenes, etc.

RECIENTES APROBADO ABIERTO PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de disposición ?	Rango ?	Organismo ?	Fecha de publicación ?	Orden ?	Palabra clave ?
VALLADOLID	2021-04-05	ORDEN	CONSEJERIA DE SANIDAD	2021-04-21		
Salamanca	2021-07-12	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-07-20		
VALLADOLID	2021-09-27	RESOLUCIÓN	AYUNTAMIENTO DE VALLADOLID	2021-09-03		
VALLADOLID	2021-04-26	ORDEN	CONSEJERIA DE LA PRESIDENCIA	2021-05-06		
VALLADOLID	2021-04-30	ORDEN	CONSEJERIA DE SANIDAD	2021-05-24		
VALLADOLID	2021-04-05	ORDEN	CONSEJERIA DE SANIDAD	2021-04-21		
VALLADOLID	2021-03-05	ORDEN	CONSEJERIA DE SANIDAD	2021-03-19		
VALLADOLID	2021-01-28	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-02-10		
VALLADOLID	2021-03-24	RESOLUCIÓN	UNIVERSIDAD DE VALLADOLID	2021-03-31		
VALLADOLID	2021-07-15	RESOLUCIÓN	CONSEJERIA DE FAMILIA E IGUALDAD DE OPORTUNIDADES	2021-07-27		

Filas por página: 10 1-15 de 234

ELIMINAR TODOS LOS FILTROS

Figura A.7: Todos los documentos que se encuentran en la fase modificaciones

Por otro lado, tenemos los filtros, los cuales voy a explicar cada uno en una etapa diferente y finalmente aplicaré un conjunto de filtros para mostrar su funcionalidad.

**Localidad:** Permite filtrar los documentos por su localidad. En la figura A.8 se puede ver que se ha utilizado la localidad “valladolid”. Una vez se ha dado al “intro.º bien se clicla en el botón de “buscar”, se bloquea el campo de texto, ya que es necesario limpiar dicho filtro o resetear todos los filtros para que se vuelva a habilitar.

Gestor y localizador de oposiciones en Castilla y León

Esta página es una herramienta que se ha creado con el fin de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son generadas por las entes que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECIENTES APROBADO ABIERTO PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de disposición ?	Rango ?	Organismo ?	Fecha de publicación ?	Orden ?	Palabra clave ?
VALLADOLID	2021-05-06	ORDEN	CONSEJERIA DE EMPLEO E INDUSTRIA	2021-10-11		
VALLADOLID	2021-05-06	ORDEN	CONSEJERIA DE EMPLEO E INDUSTRIA	2021-10-11		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE LA PRESIDENCIA	2021-10-11		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE LA PRESIDENCIA	2021-10-08		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE SANIDAD	2021-10-08		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE SANIDAD	2021-10-08		
VALLADOLID	2021-05-20	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-10-08		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE SANIDAD	2021-10-08		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE LA PRESIDENCIA	2021-10-08		
VALLADOLID	2021-05-20	ORDEN	CONSEJERIA DE SANIDAD	2021-10-08		

Filas por página: 10 1-10 de 41

ELIMINAR TODOS LOS FILTROS

Figura A.8: Documentos filtrados por localidad en fase recientes

**Fecha de disposición:** Permite filtrar los documentos en base a la fecha de disposición

de la resolución, orden, acuerdo, etc. Esta fecha hace referencia a cuándo se dicta la resolución, orden, acuerdo... Puede ser de utilidad para personas que tenga más conocimientos sobre el BOCyL. En la figura A.9 se puede ver como se filtra del 1-31 de agosto y se sitúa en la fase aprobado.

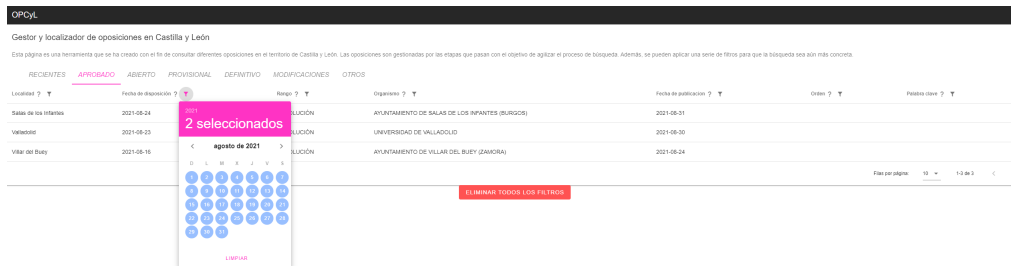


Figura A.9: Todos los documentos que se encuentran en la fase aprobado filtrando por fecha de disposición

**Rango:** Desplegable que permite filtrar los documentos en base a su categoría normativa. Los posibles valores son: orden, resolución, acuerdo, decreto y otros. En la figura A.10 se puede ver que se ha utilizado el filtro con RESOLUCIÓN en la etapa abierto.

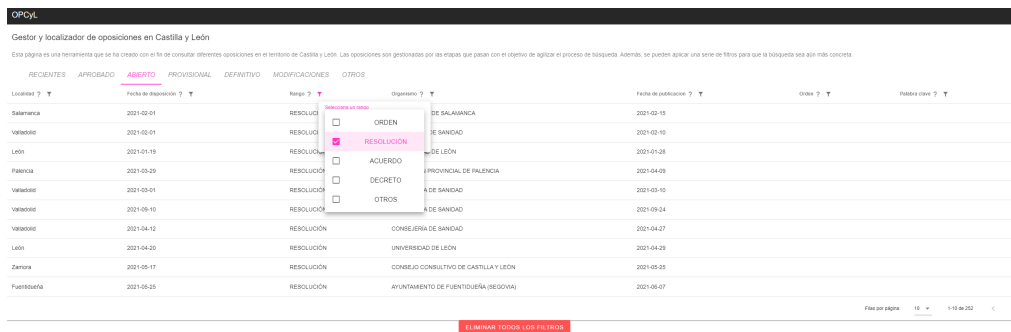


Figura A.10: Todos los documentos que se encuentran en la fase abierto filtrando por rango

**Organismo:** Desplegable que permite filtrar los documentos en base al organismo que emite el documento. Los posibles valores son: consejería, universidad, ayuntamiento, diputación, cortes y consejo. En la figura A.11 se puede visualizar cómo se ha filtrado el organismo siendo UNIVERSIDAD en la etapa provisional.

OPCYL

Gestor y localizador de oposiciones en Castilla y León

Esta página es una herramienta que se ha creado con el fin de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son gestionadas por las etapas que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECIENTES APROBADO ABIERTO **PROVISIONAL** DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de expedición ?	Rango ?	Organismo ?	Fecha de publicación ?	Otros ?	Fecha clave ?
Salamanca	2021-04-06	RESOLUCIÓN	UNIVERSIDAD	2021-04-13		
León	2021-07-26	RESOLUCIÓN		2021-08-03		
León	2021-06-01	RESOLUCIÓN		2021-06-07		
León	2021-07-26	RESOLUCIÓN	UNIVERSIDAD DE LEÓN	2021-08-03		
Salamanca	2021-05-04	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-05-11		
Salamanca	2021-05-03	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-05-11		
Valladolid	2021-04-09	RESOLUCIÓN	UNIVERSIDAD DE VALLADOLID	2021-04-19		
Salamanca	2021-03-08	RESOLUCIÓN	UNIVERSIDAD DE SALAMANCA	2021-03-16		
Valladolid	2021-03-10	RESOLUCIÓN	UNIVERSIDAD DE VALLADOLID	2021-03-17		
Valladolid	2021-04-21	RESOLUCIÓN	UNIVERSIDAD DE VALLADOLID	2021-04-30		

Eliminar todos los filtros

Filas por página: 10 1-12 de 24

Figura A.11: Todos los documentos que se encuentran en la fase provisional filtrando por organismo

**Fecha de publicación:** Permite filtrar documentos en base a su fecha de publicación. En la figura A.12 se puede apreciar que se ha filtrado del 3-18 de febrero en la etapa definitiva.



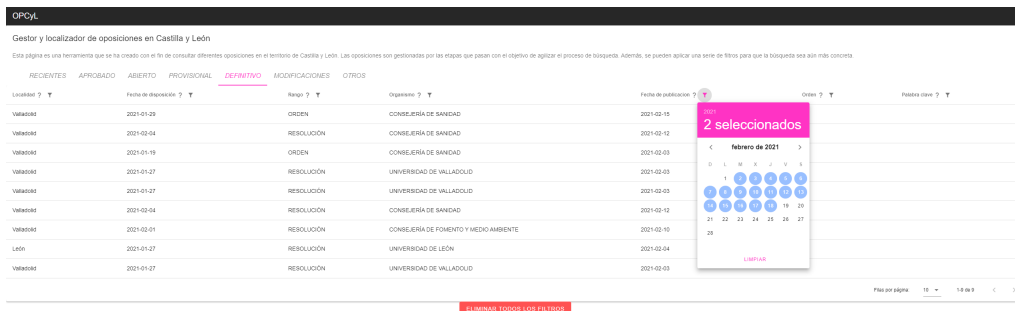


Figura A.12: Todos los documentos que se encuentran en la fase definitivo filtrando por fecha de publicación

**Orden:** Permite filtrar todos los documentos que contengan el número de disposición a buscar. La estructura del número de disposición es: ca/nm/an. Siendo CA el código alfabético del departamento, NM número y AN año. Algunos ejemplos de CA son FYM (FOMENTO Y MEDIO AMBIENTE), EDU (educación), CYT (cultura y turismo), PRE (presidencia), SAN (sanidad), etc. Ocurre lo mismo que en localidad, hasta que no se limpie el filtro dando a “Limpiar“ o al boton de resetear todos los filtros, no se podrá volver a utilizar. En la figura A.13 se puede visualizar que se ha filtrado por documentos que contengan la orden SAN en la etapa modificaciones.

OPCyL

Gestor y localizador de oposiciones en Castilla y León

Esta página es una herramienta que se ha creado con el fin de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son gestionadas por los estados que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECIENTES APROBADO ABIERTO PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de disposición ?	Respi ?	Organismo ?	Fecha de publicación ?	Orden ?	Palabra clave ?
VANEGAS	2021-04-05	ORDEN	CONSEJERIA DE SANIDAD	2021-04-21		
SALAMANCA	2021-07-12	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-07-20		
VANEGAS	2021-04-30	ORDEN	CONSEJERIA DE SANIDAD	2021-05-24		
VANEGAS	2021-04-05	ORDEN	CONSEJERIA DE SANIDAD	2021-04-21		
VANEGAS	2021-03-05	ORDEN	CONSEJERIA DE SANIDAD	2021-03-19		
VANEGAS	2021-01-26	RESOLUCIÓN	CONSEJERIA DE SANIDAD	2021-02-10		
VANEGAS	2021-04-30	ORDEN	CONSEJERIA DE SANIDAD	2021-05-24		
VANEGAS	2021-04-30	ORDEN	CONSEJERIA DE SANIDAD	2021-05-24		
VANEGAS	2021-02-08	ORDEN	CONSEJERIA DE SANIDAD	2021-02-17		
VANEGAS	2021-04-28	ORDEN	CONSEJERIA DE SANIDAD	2021-05-17		

Fila por página: 10 1/18 de 00 < >

ELIMINAR TODOS LOS FILTROS

Figura A.13: Todos los documentos que se encuentran en la fase modificaciones filtrando por número de disposición (orden)

**Palabra clave:** Permite filtrar los documentos que contengan la palabra o palabras introducidas. Este filtro va orientado a personas que sepan cuál es el nombre del puesto de trabajo concreto que buscan o para personas que busquen un trabajo compatible y por tanto busquen el grupo (dicha mención de grupos se hace en la sección 2.3. En la figura A.14 se puede ver como se han utilizado dos filtros, localidad y palabras claves. Localidad se filtró por medina. Palabra clave se filtró con policía y c1.

OPCyL

Gestor y localizador de oposiciones en Castilla y León

Esta página es una herramienta que se ha creado con el fin de consultar diferentes oposiciones en el territorio de Castilla y León. Las oposiciones son gestionadas por los estados que pasan con el objetivo de agilizar el proceso de búsqueda. Además, se pueden aplicar una serie de filtros para que la búsqueda sea aún más concreta.

RECIENTES APROBADO ABIERTO PROVISIONAL DEFINITIVO MODIFICACIONES OTROS

Localidad ?	Fecha de disposición ?	Respi ?	Organismo ?	Fecha de publicación ?	Orden ?	Palabra clave ?
Medina de Rioseco	2021-03-15	DECRETO	AYUNTAMIENTO DE MEDINA DE RIOSECO (VALLADOLID)	2021-03-22		
Medina de Pomar	2021-04-20	RESOLUCIÓN	AYUNTAMIENTO DE MEDINA DE POMAR (BURGOS)	2021-04-29		

Fila por página: 10 1/2 de 00 < >

ELIMINAR TODOS LOS FILTROS

Figura A.14: Todos los documentos que se encuentran en la fase abierto filtrando por palabras clave y localidad

Por último, mostraré dos ejemplos de cómo utilizar correctamente los filtros. La utilización de un filtro u otro depende de la situación de la persona, ya que cada uno se moverá por las diferentes fases y utilizará los filtros que les sean necesarios.

Una persona que busca en Salamanca está esperando que una oposición se apruebe y lleva sin consultar dos meses. La oposición que está interesado está relacionada con arquitectura. Se tiene que situar en la etapa .<sup>a</sup>probado”según su interés y los filtros correctos a utilizar serían los siguientes:

- **Localidad:** Buscará Salamanca.
- **Fecha de publicación:** Vamos a suponer que se encuentra en mayo, por lo que mirará abril y mayo. Desde el 1 de abril hasta el 27 de mayo.
- **Organismo:** Filtrará por ayuntamiento, ya que normalmente este tipo de ofertas las publica éste organismo.

El resultado obtenido (figura A.15) es una resolución procedente de Salamanca que se publicó el 21 de mayo.

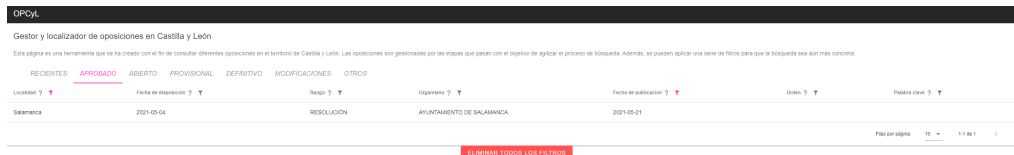


Figura A.15: Resultado de buscar documento utilizando filtros localidad, fecha de publicación y organismo

Si clicamos sobre el enlace pdf, se podrá visualizar la siguiente información (ver figura A.16).

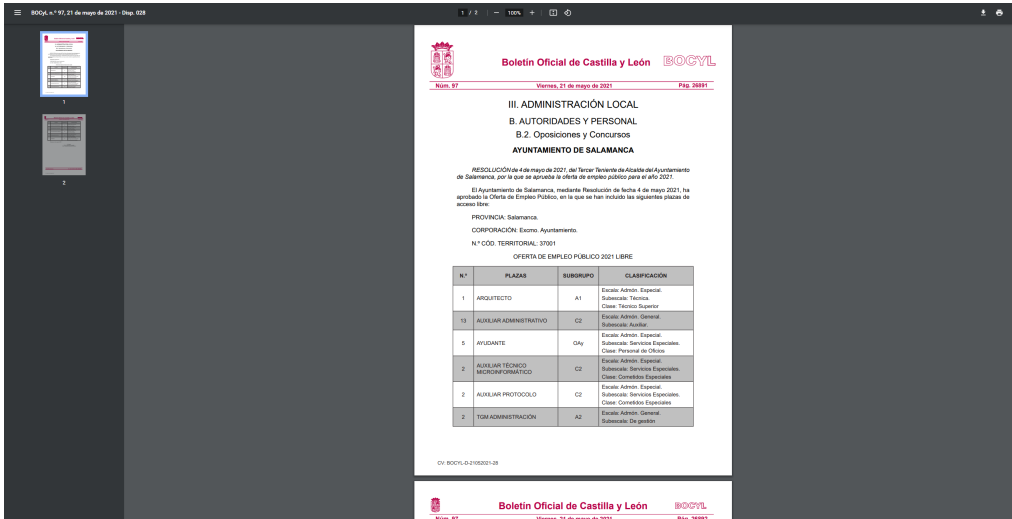


Figura A.16: Enlace PDF del resultado de filtrar por localidad, fecha de publicación y organismo

Una persona está consultando si su oposición se ha abierto, y quiere mirar si se ha publicado en el mes de agosto. La oposición que busca es en Valladolid y el puesto de trabajo en el que está interesado es de técnico auxiliar. Vamos a suponer que el usuario no sabe en que organismo se encuentra su puesto de trabajo. Se tiene que situar en la fase abierto y los filtros correctos que tiene que utilizar son:

- **Localidad:** Buscará Valladolid.
- **Fecha de publicación:** Filtrará dos fechas como anteriormente se hizo, pero solo contendrá un mes, del 1 de agosto hasta el 31 del mismo mes.
- **Palabra clave:** Las palabras clave que busca son o bien técnico o bien técnico auxiliar. Utilizaremos la segunda opción ya que concreta más.

El resultado obtenido (ver figura A.17) son dos documentos, los dos publicados en la misma fecha y por el mismo organismo.

# APÉNDICE A. MANUAL DE USUARIO



Figura A.17: Resultado de búsqueda utilizando filtros localidad, fecha de publicación y palabra clave

Como se puede ver en la figura A.18 el primer documento no nos sirve, ya que no oferta una convocatoria cuyo puesto de trabajo sea técnico auxiliar, sino de mecánico-mantenimiento. No obstante si que se encuentra dichas palabras en el pdf.

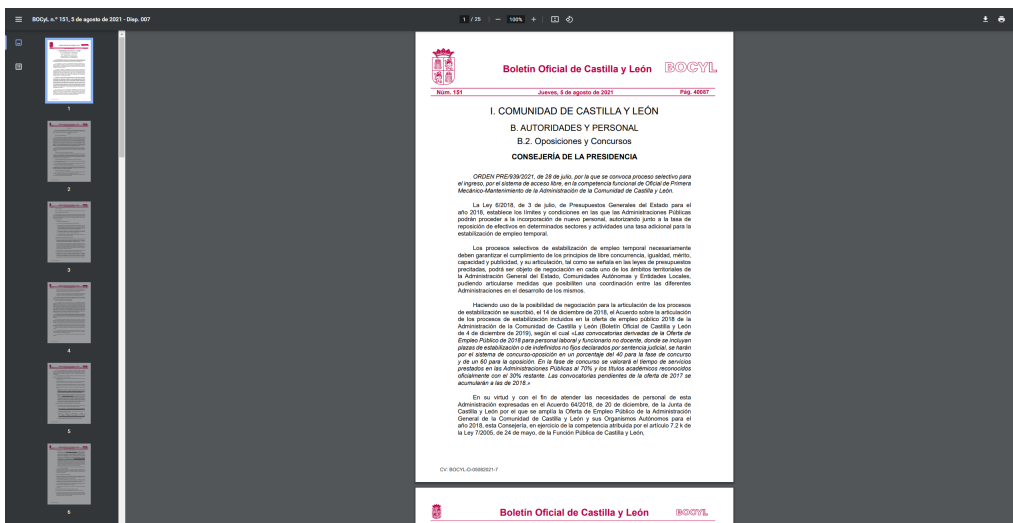


Figura A.18: Documento encontrado con filtros utilizados pero resultado no deseado

En la figura A.19 se puede apreciar que éste documento si que serviría al usuario, ya que oferta el puesto de trabajo en el que el usuario está interesado.

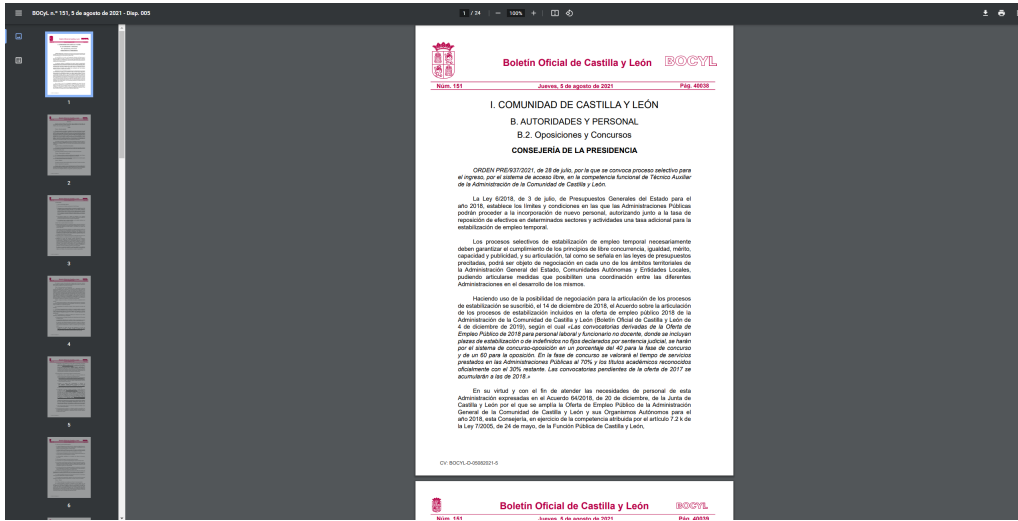


Figura A.19: Documento encontrado con filtros utilizados y resultado satisfactorio

El filtro de palabra clave es de gran valor, ya que nos reduce considerablemente la búsqueda de un puesto de trabajo para una persona que no sepa cuál es su organismo. Puede llegar a conseguir, por ejemplo, pasar de buscar en 10 documentos (utilizando solo los filtros localidad y fecha de publicación, aún así ya es reducida) a buscar en solo 2 documentos.

Por último, si quisiéramos quitar todos los filtros utilizados, solo tendríamos que utilizar el botón que se encuentra en la parte inferior de la tabla.

## Apéndice B

# Manual de despliegue

### B.1. Herramientas necesarias

Las herramientas necesarias para crear el entorno de trabajo y poder desplegar la página web son las siguientes:

- Firebase CLI versión 9.16.5.
- NPM versión 6.14.11.
- NODE JS versión 14.15.5.

### B.2. Preparación del entorno y despliegue

En primer lugar es necesario tener el código, el cuál se localiza en [https://github.com/danfern8/OPCyL\\_TFG](https://github.com/danfern8/OPCyL_TFG)). Una vez obtenido, en la carpeta raíz del proyecto, se ejecutará **npm install**. Éste comando instala las dependencias y paquetes del proyecto, y se introducirán automáticamente en la carpeta `node_modules`.

Si deseamos visualizar la página web de forma local, lo que haremos será ejecutar el comando **npm run serve**. Este comando crea un servidor web local en el puerto 8080.

Para comenzar a realizar el despliegue será necesario ejecutar el comando **npm run build**. Este comando crea una carpeta llamada *dist* en la raíz del proyecto, cuyo propósito será preparar el despliegue en un servidor web, en nuestro caso en Firebase.

Se ha utilizado Firebase, que aporta una herramienta para poder desplegar la página web. Para aprovechar esta herramienta de hosting va a ser necesario realizar los siguientes pasos:

1. Iniciar sesión en Firebase. Esto se consigue ejecutando en una consola de comandos **firebase login**.
2. Nos redirecciona a la web de Firebase para iniciar sesión con una cuenta que tenga permisos suficientes en el proyecto.
3. Desplegar. Esto se consigue ejecutando en la consola de comandos desde la carpeta raíz del proyecto el comando **firebase deploy --only hosting**.

En el último paso, despliega el contenido de *dist*. Es necesario especificar la sección *hosting*, ya que hay otra conocida como *functions*. Esta última no es necesario desplegarla porque ya lo está desde un inicio, y contiene las *functions* que se han ido utilizando a lo largo del proyecto. Actualmente solo contiene una activa (la encargada de actualizar la base de datos y con ello la página web).

Si, por alguna razón se hiciera sin especificar, se desplegarán también las *functions*, pero el proceso tardaría algunos minutos más. Finalmente, la página estaría desplegada en ***https://my-app-563be.web.app***



# Bibliografía

- [1] *¿Qué son las directivas de Vue? - Javascript en español*. URL: <https://lenguajejs.com/vuejs/directivas-vue/que-son-directivas/> (visitado 14-09-2021).
- [2] *Administracion.gob.es - Inicio*. URL: <https://administracion.gob.es/> (visitado 14-10-2021).
- [3] *Atomic Web Design o Diseño Guiado por Componentes*. URL: <https://adrianalonso.es/arquitectura-del-software/atomic-web-design-o-diseno-guiado-por-componentes/> (visitado 14-09-2021).
- [4] *axios - npm*. URL: <https://www.npmjs.com/package/axios> (visitado 05-10-2021).
- [5] *Backend as a service - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Backend\\_as\\_a\\_service](https://es.wikipedia.org/wiki/Backend_as_a_service) (visitado 06-10-2021).
- [6] *Backend como servicio: ¿Qué es un BaaS? — Low-code backend to build modern apps*. 1 de ene. de 2021. URL: <https://blog.back4app.com/es/que-es-un-baas-backend-como-servicio/> (visitado 09-09-2021).
- [7] *Bob Hughes y Mike Cotterell. Software Project Management. 5a edición. Europa, Oriente Medio y Africa: McGraw-Hill Education, 2009*. (Visitado 05-10-2021).
- [8] *BOCYL — BOLETÍN OFICIAL DE CASTILLA Y LEÓN*. URL: <https://bocyl.jcyl.es/> (visitado 04-10-2021).
- [9] *BOCYL — BOLETÍN OFICIAL DE CASTILLA Y LEÓN*. URL: <https://bocyl.jcyl.es/> (visitado 04-10-2021).
- [10] *Boletín Oficial de Castilla y León — Análisis de datos abiertos JCyL*. URL: [https://analisis.datosabiertos.jcyl.es/explore/dataset/bocyl/api/?flg=es&sort=fecha\\_publicacion&q=oposiciones&refine.fecha\\_publicacion=2021](https://analisis.datosabiertos.jcyl.es/explore/dataset/bocyl/api/?flg=es&sort=fecha_publicacion&q=oposiciones&refine.fecha_publicacion=2021) (visitado 04-10-2021).
- [11] *Buscador de Oposiciones 2021*. URL: <https://www.opobusca.com/> (visitado 14-10-2021).
- [12] *Cloud Functions para Firebase*. URL: <https://firebase.google.com/docs/functions> (visitado 27-09-2021).
- [13] *Control de acceso HTTP (CORS) - HTTP — MDN*. URL: <https://developer.mozilla.org/es/docs/Web/HTTP/CORS> (visitado 13-10-2021).
- [14] *Firebase*. URL: <https://firebase.google.com/?hl=es> (visitado 05-10-2021).

- [15] *Firestore Hosting — Hosting web rápido y seguro*. URL: [https://firebase.google.com/products/hosting?gclid=Cj0KCQjw5JSLBhCxARIsAHg02Sd\\_KDsJtqz22MrvRh9JJDhwfD3aYegwCB&gclsrc=aw.ds](https://firebase.google.com/products/hosting?gclid=Cj0KCQjw5JSLBhCxARIsAHg02Sd_KDsJtqz22MrvRh9JJDhwfD3aYegwCB&gclsrc=aw.ds) (visitado 12-10-2021).
- [16] *Getting Started - vue.js*. URL: <https://programmerclick.com/article/8016972475/> (visitado 06-10-2021).
- [17] *Git*. URL: <https://git-scm.com/> (visitado 05-10-2021).
- [18] *GitHub*. URL: <https://github.com/> (visitado 05-10-2021).
- [19] “Guía docente de la asignatura”. En: (). URL: [https://albergueweb1.uva.es/guia\\_docente/uploads/2021/545/46977/1/Documento.pdf](https://albergueweb1.uva.es/guia_docente/uploads/2021/545/46977/1/Documento.pdf) (visitado 04-10-2021).
- [20] *iso-8859-2 - npm*. URL: <https://www.npmjs.com/package/iso-8859-2> (visitado 05-10-2021).
- [21] *La instancia Vue — Vue.js*. URL: <https://es.vuejs.org/v2/guide/instance.html> (visitado 05-10-2021).
- [22] *LaTeX - A document preparation system*. URL: <https://www.latex-project.org/> (visitado 05-10-2021).
- [23] *Licencia BSD - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Licencia\\_BSD](https://es.wikipedia.org/wiki/Licencia_BSD) (visitado 25-10-2021).
- [24] *Licencia BSD - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Licencia\\_BSD](https://es.wikipedia.org/wiki/Licencia_BSD) (visitado 25-10-2021).
- [25] *Licencia MIT - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Licencia\\_MIT](https://es.wikipedia.org/wiki/Licencia_MIT) (visitado 25-10-2021).
- [26] *MVC y MVVM (3-5) - Adictos al trabajo*. URL: <https://www.adictosaltrabajo.com/2012/10/07/zk-mvc-mvvm/> (visitado 14-09-2021).
- [27] *npm*. URL: <https://www.npmjs.com/> (visitado 05-10-2021).
- [28] *Overleaf, Editor de LaTeX online*. URL: <https://es.overleaf.com/> (visitado 14-10-2021).
- [29] *Premier Diagramming, Modeling Software & Tools — Astah*. URL: <https://astah.net/> (visitado 05-10-2021).
- [30] *Propiedades y observadores calculados - Vue.js*. URL: <https://vuejs.org/v2/guide/computed.html#Watchers> (visitado 13-10-2021).
- [31] *regex101: build, test, and debug regex*. URL: <https://regex101.com/> (visitado 05-10-2021).
- [32] *Salario de un Programador/a junior en España*. URL: <https://es.indeed.com/career/programador-junior/salaries> (visitado 14-10-2021).
- [33] *Scrum — Scrum.org*. URL: <https://www.scrum.org/> (visitado 05-10-2021).
- [34] *Servicios de cloud computing — Google Cloud*. URL: <https://cloud.google.com/> (visitado 05-10-2021).
- [35] *Single-page application - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Single-page\\_application](https://es.wikipedia.org/wiki/Single-page_application) (visitado 05-10-2021).
- [36] *Software como servicio - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Software\\_como\\_servicio](https://es.wikipedia.org/wiki/Software_como_servicio) (visitado 06-10-2021).

- [37] *Tabla de coeficientes de amortización lineal*. - Agencia Tributaria. URL: [https://www.agenciatributaria.es/AEAT.internet/Inicio/\\_Segmentos\\_/Empresas\\_y\\_profesionales/Empresas/Impuesto\\_sobre\\_Sociedades/Periodos\\_impositivos\\_a\\_partir\\_de\\_1\\_1\\_2015/Base\\_imponible/Amortizacion/Tabla\\_de\\_coeficientes\\_de\\_amortizacion\\_lineal\\_.shtml](https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml) (visitado 14-10-2021).
- [38] *Tipos de Oposiciones 2021: Grupos según Estudios*. URL: <https://www.masterd.es/blog/informacion-oposiciones> (visitado 25-10-2021).
- [39] *Trello*. URL: <https://trello.com/es> (visitado 05-10-2021).
- [40] *Vue Router*. URL: <https://router.vuejs.org/> (visitado 05-10-2021).
- [41] *Vue.js*. URL: <https://vuejs.org/> (visitado 05-10-2021).
- [42] *Vuetify — A Material Design Framework for Vue.js*. URL: <https://vuetifyjs.com/en/> (visitado 05-10-2021).
- [43] *xml-js - npm*. URL: <https://www.npmjs.com/package/xml-js> (visitado 05-10-2021).