

Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
Mención en Ingeniería del Software

Gestor y localizador de oposiciones de la
Administración General del Estado

Autor:
D. Javier Moro García

Tutores:
D. Joaquín Nicolas Adiego Rodríguez
Dña. Natalia Martín Cruz

Agradecimientos

Quiero agradecer a mi familia por haber confiado en mis capacidades y haberme apoyado durante los años de universidad. Por el esfuerzo y empeño en que reciba la mejor formación para la vida adulta y darme unos valores morales para afrontarla como una buena persona.

A mi hermano pequeño, porque sin él no sería la persona que soy.

A mis amigos, que han hecho todo este proceso más ameno, por estar tanto en los momentos tristes como los alegres, por apoyarme y porque siempre habéis estado ahí donde os necesitaba.

A mis dos tutores, Joaquín y Natalia, porque sin su ayuda no habría sido posible este proyecto.

A todos aquellos que me han ayudado a obtener los conocimientos necesarios para formarme como profesional.

Resumen

Las oposiciones de la Administración General del Estado son emitidas diariamente en el Boletín Oficial del Estado. El acceso a estas oposiciones puede llegar a ser una acción bastante compleja y tediosa. Por este motivo, se ha visto la oportunidad de desarrollar una aplicación que permita al usuario localizar y gestionar las oposiciones desde una forma más accesible.

El objetivo de este Trabajo de Fin de Grado es desarrollar una aplicación multiplataforma que contenga las oposiciones emitidas en el BOE, que permita al usuario filtrar las oposiciones según distintos criterios y que se pueda utilizar tanto en un dispositivo móvil como en un ordenador.

Para este objetivo, se ha desarrollado una aplicación web PWA (Aplicación Web Progresiva) que pueda ser usada tanto para un dispositivo de escritorio como para un dispositivo móvil. Se ha contado con un Script que recoge los datos que utiliza la aplicación desde la página oficial, utilizando los datos abiertos que ofrece el BOE. Para el acceso a los datos, se ha contado con una API REST que se comunica con la base de datos.

El trabajo ha sido desarrollado usando distintas herramientas para cada parte de la aplicación. En la parte Backend, se ha utilizado el lenguaje Java en el Script de recogida de datos y en la API REST. En el Frontend, se ha utilizado TypeScript como lenguaje principal y los frameworks de Ionic y Angular para el desarrollo de la PWA. Dentro del marco de trabajo, se ha utilizado una versión adaptada del proceso de gestión SCRUM para el desarrollo ágil del proyecto.

Abstract

Civil service examinations are published daily in the Official State Gazette (Boletín Oficial del Estado). The access to these oppositions can become a quite complex and tedious action. For this reason, we have seen the opportunity to develop an application that allows the user to locate and manage the oppositions in an accessible manner.

The objective of this Final Degree Project is to develop a multiplatform application that contains the competitive examinations issued in the BOE, which allows the user to filter the competitions according to different criteria and that can be used both on a mobile device and on a computer.

For this objective, a PWA (Progressive Web Application) web application has been developed that can be used both for a desktop device and for a mobile device. A Script has been used to collect the data used by the application from the official website, using the open data provided by the BOE. To access the data, a REST API has been used to communicate with the database.

The work has been developed using different tools for each part of the application. In the Backend part, Java language has been used in the data collection Script and in the REST API. In the Frontend, TypeScript has been used as the main language and the Ionic and Angular frameworks for the development of the PWA. Within the framework, an adapted version of the SCRUM management process has been used for the agile development of the project.

Tabla de Contenido

Agradecimientos	3
Resumen	5
Abstract	7
1. Introducción	17
1.1. Contexto	17
1.2. Motivación	19
1.3. Introducción a las PWA	19
1.4. Introducción a los Datos Abiertos	20
1.5. Introducción al Desarrollo Ágil (SCRUM)	21
1.5.1. Eventos	22
1.5.2. Roles	22
1.5.3. Artefactos	23
1.6. API REST	23
1.6.1. Características de los Servicios REST	24
1.7. Objetivos	25
1.8. Estructura de la memoria	25
2. Tecnologías utilizadas	27
2.1. IDE's	27

TABLA DE CONTENIDO

2.1.1. NetBeans	27
2.1.2. Visual Studio Code	27
2.2. Diseño	28
2.2.1. Astah profesional	28
2.3. Pencil Project	28
2.4. Frameworks	29
2.4.1. Ionic	29
2.4.2. AngularJS	29
2.5. Lanzamiento	30
2.5.1. Glassfish	30
2.5.2. Apache Derby	30
2.6. Testing	30
2.6.1. Postman	31
2.6.2. Mozilla Firefox	31
3. Planificación y requisitos	33
3.1. Desarrollo Ágil (SCRUM) adaptado al proyecto	33
3.2. Público objetivo	34
3.3. Riesgos	35
4. Análisis	39
4.1. Análisis del BOE	39
4.2. Oposiciones y concursos	41
4.3. Oposición	41
4.4. Referencias	42
4.5. Casos de Uso	43
4.5.1. Listas	43

4.5.2.	Búsqueda de oposiciones según distintos criterios	44
4.5.3.	Búsqueda por directorio	45
5.	Diseño	47
5.1.	Diagrama de Casos de Uso	47
5.2.	Diagrama de clases	50
5.3.	Recursos REST	51
5.4.	Despliegue	52
5.5.	Diseño de la Página	52
5.5.1.	Ventana de Inicio	53
5.5.2.	Lista de Oposiciones	55
5.5.3.	Lista de Departamentos/Epígrafes	56
5.5.4.	Detalle de la Oposición	56
6.	Implementación y Pruebas	65
6.1.	Backend	65
6.1.1.	Script de Recopilación de Datos	65
6.1.2.	Servicio REST	66
6.2.	Frontend	67
6.3.	Pruebas	67
6.3.1.	Pruebas Automatizadas Servicio REST	67
6.3.2.	Pruebas E2E	70
7.	Seguimiento	73
7.1.	Sprint 1 (31/01/2021 - 7/02/2021)	73
7.2.	Sprint 2 (08/02/2021 - 15/02/2021)	74
7.3.	Sprint 3 (15/02/2021 - 23/03/2021)	75
7.4.	Sprint 4 (23/02/2021 - 02/03/2021)	76

TABLA DE CONTENIDO

7.5. Sprint 5 (04/03/2021 - 14/03/2021)	76
7.6. Época de prácticas (14/03/2021 - 10/04/2021)	77
7.7. Sprint 6 (12/04/2021 - 25/04/2021)	78
7.8. Sprint 7 (26/04/2021 - 02/05/2021)	79
7.9. Sprint 8 (04/05/2021 - 12/05/2021)	79
7.10. Época de prácticas (17/05/2021 - 30/05/2021)	80
7.11. Sprint 9 (31/05/2021 - 07/06/2021)	81
7.12. Sprint 10 (08/06/2021 - 15/06/2021)	82
7.13. Spint Final (18/06/2021 - Entrega del proyecto)	83
7.14. Recapitulación	84
7.14.1. Calendario final	84
7.14.2. Trabajo Total	84
8. Conclusiones	87
8.1. Trabajo Futuro	88
A. Manual	91
A.1. Instalación de las Herramientas	91
A.1.1. Versión de las herramientas	91
A.1.2. Instalación del Backend	91
A.1.3. Instalación del Frontend	92
A.2. Configuración de la Aplicación	92
A.2.1. Ejecutar parte Backend	93
A.2.2. Ejecutar parte Frontend	93
A.3. Manual de uso de la aplicación	93
Bibliografía	101

Lista de Figuras

1.1. Opciones de búsqueda del sistema de búsqueda principal [11]	18
1.2. Sistema de Búsqueda del BOE [15]	18
1.3. SCRUM	21
1.4. REST API - Author: Seobility - License: CC BY-SA 4.0	24
4.1. Ejemplo real de dos oposiciones con un mismo par Departamento-Epígrafe [13]	41
4.2. Ejemplo real de una oposición en formato XML [12]	42
4.3. Ejemplo real de una oposición que contiene una referencia posterior	43
5.1. Diagrama de Casos de Uso	47
5.2. Diagrama de Clases	50
5.3. Diagrama de Recursos REST	51
5.4. Diagrama de Despliegue	52
5.5. Boceto de la ventana de inicio del dispositivo móvil	54
5.6. Boceto de la ventana de inicio del escritorio	55
5.7. Boceto de la ventana de inicio con el menú activo del dispositivo móvil	58
5.8. Boceto de la ventana con la lista de oposiciones del dispositivo móvil	59
5.9. Boceto de la ventana con la lista de oposiciones del escritorio	60
5.10. Boceto de la ventana con la lista de departamentos del dispositivo móvil	61
5.11. Boceto de la ventana con la lista de departamentos del escritorio	62

5.12. Boceto de la ventana con el detalle de la oposición del dispositivo móvil . . .	63
5.13. Boceto de la ventana con el detalle de la oposición del escritorio	64
6.1. Ejemplo de una prueba en la aplicación Postman	69
6.2. Captura de la aplicación Postman con las pruebas automatizadas	69
6.3. Ejemplo de búsqueda con fechas, estado y especialidad rellena	70
A.1. Ventana Inicial de la Aplicación	94
A.2. Herramienta de desarrollo de Firefox	95
A.3. Vista móvil de la aplicación	95
A.4. Vista del listado de las oposiciones	96
A.5. Vista del detalle de un Departamento	96
A.6. Vista en detalle de una oposición	97

Lista de Tablas

3.1. Riesgo de modificación del formato del BOE	35
3.2. Riesgo de modificación del formato de las Oposiciones	35
3.3. Riesgo de falta de tiempo	36
3.4. Riesgo de pandemia	36
3.5. Riesgo de interfaz mal elegida	37
3.6. Riesgo de errores en la disposición de los datos	37
3.7. Riesgo de modificación de los requisitos	37
5.1. Descripción del Caso de Uso Explorar Oposiciones	48
5.2. Descripción del Caso de Uso Explorar Departamentos	48
5.3. Descripción del Caso de Uso Explorar Epígrafes	48
5.4. Descripción del Caso de Uso Buscar mediante búsqueda avanzada	49
5.5. Descripción del Caso de Uso Buscar por directorio	49
5.6. Descripción del Caso de Uso Ver Oposición	50
6.1. Tabla con las pruebas de caja negra del servicio REST	68
7.1. Sprint 1	73
7.2. Sprint 2	74
7.3. Sprint 3	75
7.4. Sprint 4	76

LISTA DE TABLAS

7.5. Sprint 5	77
7.6. Época de prácticas	77
7.7. Sprint 6	78
7.8. Sprint 7	79
7.9. Sprint 8	80
7.10. Segunda Época de Prácticas y Exámenes	80
7.11. Sprint 9	81
7.12. Sprint 10	83
7.13. Sprint Final	83
7.14. Calendario Final	84
7.15. Calendario Final	85

Capítulo 1

Introducción

1.1. Contexto

El Boletín Oficial del Estado (BOE) es el diario nacional del Reino de España donde se publica diariamente leyes, disposiciones y actos de inserción obligatoria. Aquí es donde se publican las oposiciones y concursos que las distintas Administraciones Públicas emiten diariamente.

Al año se publican miles de oposiciones en el BOE. Estas quedan registradas en la web y el acceso a ellas es bastante complejo. Si un usuario desea ver las oposiciones que se generan al día, deberá acceder al BOE emitido en esa fecha y allí localizar la sección que las contiene. No es un proceso difícil, pero si luego quiere ver más oposiciones, deberá reproducir esta serie de acciones para cada día.

Adicionalmente, si el usuario solo desea ver las oposiciones que ha generado un departamento específico, deberá repetir este proceso hasta que encuentre lo que desee. Tampoco es que la web oficial no tenga medios para que este usuario hipotético pueda encontrar lo que busca, la misma página oficial dispone de varios métodos de búsqueda, pero esta tiene bastantes problemas que la convierte en un servicio ineficiente. Figura 1.1

La página oficial posee un método de búsqueda principal. Esta se denomina búsqueda rápida y ofrece la posibilidad de búsqueda por palabra entre 4 opciones: Legislación, Todo el BOE, Notificaciones y Edictos. Estos cuatro términos son bastante generales y no ofrecen una clara visión de lo que representan, además son términos bastante burocráticos y los posibles usuarios puede que no sepan lo que significan.

Asimismo, la página de búsqueda dispone de varios enlaces que nos llevan a otro tipo de búsqueda más avanzada, dependiendo qué acceso hayamos seleccionado podemos encontrar diferencias entre páginas, pero normalmente comparten bastantes campos. Cabe recalcar, que no hay ningún sitio donde diga al usuario que estos enlaces llevan a sistemas de búsqueda más eficientes. Figura 1.2

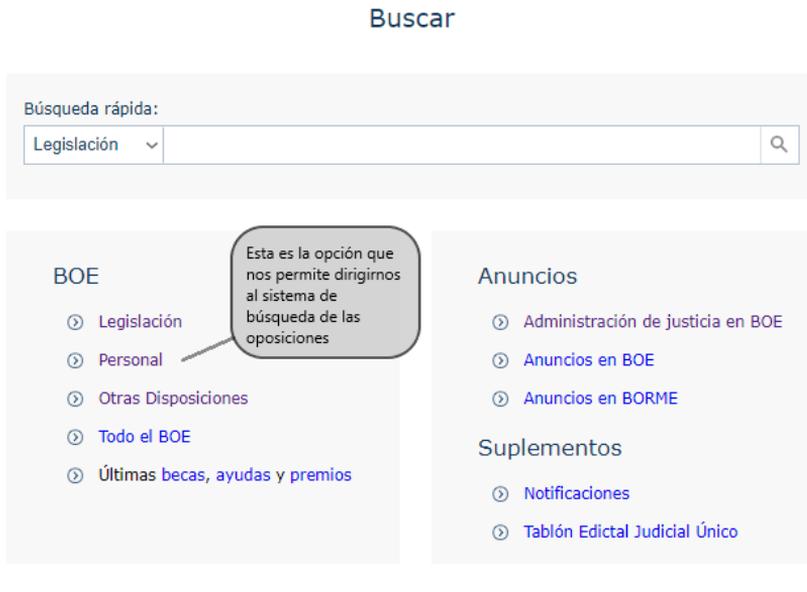


Figura 1.1: Opciones de búsqueda del sistema de búsqueda principal [11]

Personal: Oposiciones, nombramientos...

Disposiciones publicadas en la Sección II "Autoridades y personal" desde 1960.

[Ayuda y contenido](#)

Título	<input type="text"/>	Y	▼
Rango	<input type="text"/>	Y	▼
Departamento	<input type="text"/>	Y	▼
Texto	<input type="text"/>	Y	▼
Núm. de boletín	<input type="text" value="n/aaaa"/>	Y	▼
Núm. oficial	<input type="text" value="n/aaaa o DPTO/n/aaaa"/>	Y	▼
Texto	<input type="text"/>		
Fecha de publicación de	<input type="text" value="dd/mm/aaaa"/>	a	<input type="text" value="dd/mm/aaaa"/>
Fecha de disposición de	<input type="text" value="dd/mm/aaaa"/>	a	<input type="text" value="dd/mm/aaaa"/>
Documentos por página	50	Ordenados por	fecha pub. ▼ descendente ▼

Figura 1.2: Sistema de Búsqueda del BOE [15]

Este tipo de búsqueda puede ofrecer al usuario una experiencia de filtrado bastante completa. El problema es que tantos campos de búsqueda pueden abrumar al usuario, además de que varios de ellos necesitan de información previa sobre lo que significan. Por ejemplo, en

el campo rango, el usuario deberá conocer la diferencia entre real decreto, orden, resolución, etc.

La misma página parece saber de este contratiempo, y dispone al usuario de un botón de ayuda donde se explica el significado de los distintos campos. Este hecho ya implica un escalón más de dificultad por parte de la página.

Resumiendo, si un usuario desea ver unas oposiciones en concreto tiene dos posibilidades: ir de boletín en boletín o usar el buscador. Si opta por utilizar el buscador deberá conocer en qué apartado se encuentran las oposiciones (Personal) y después allí ingresar los datos en los apartados y rezar para que encuentre lo que busque.

1.2. Motivación

Como ya se ha comentado, utilizar el buscador oficial es bastante engorroso y puede llegar a abrumar al usuario, ya sea por el número de campos que aparece o por los términos tan técnicos que aparecen.

El número de usuarios que están interesados en las oposiciones dispuestas por la Administración Pública es bastante notorio. Solo hace falta ver la cantidad de oposiciones que salen a diario y la cantidad de gente que solicita acceder a ellas. Toda esta gente tendrá que acceder a un sistema de búsqueda complejo, teniendo que ir de página en página intentando ver las oposiciones que más les interesan.

Al ver esto, hemos visto la posibilidad de ofrecer al usuario que esté interesado en las oposiciones de la Administración Pública una opción para poder navegar entre ellas sin estos inconvenientes. Una opción multiplataforma que ofrezca al usuario una buena experiencia desde el ordenador y desde su dispositivo móvil.

1.3. Introducción a las PWA

Las aplicaciones web progresivas (Progressive Web Application) son un tipo de software de aplicación entregado a través de la web, construido utilizando tecnologías web comunes que incluyen HTML, CSS y JavaScript . Está diseñado para funcionar en cualquier plataforma que utilice un navegador compatible con los estándares, incluidos dispositivos móviles y de escritorio. Se podría resumir como una página web que se aprovecha de las tecnologías webs a las cuales tiene acceso para proponer una experiencia móvil similar a una aplicación nativa.[29]

La ventaja que tiene frente a una aplicación nativa es que no es necesario realizar ningún tipo de instalación en el dispositivo, ya que se aloja en el servidor web. Las tecnologías que utilizan las aplicaciones progresivas hacen que el usuario móvil tenga una experiencia completa, ya que estas se adaptan para al dispositivo que se esté usando.

En una aplicación PWA nos podemos encontrar las siguientes características:

- **Adaptabilidad:** Estas aplicaciones se adaptan automáticamente a cualquier formato, navegador o dispositivo, ya sea móvil u ordenador.
- **Multiplataforma:** Las aplicaciones web progresivas contemplan la ejecución en diversos dispositivos, sistemas operativos y navegadores. Esto, además de ser clave a la hora de ofrecer una experiencia de usuario satisfactoria, supone facilidades para los desarrolladores y permite abaratar costes, puesto que no se requieren programaciones diferenciadas. Esta es una de las características clave que se han valorado para su elección en este proyecto.
- **Apariencia nativa:** La interfaz de usuario y, en general, la apariencia de una PWA es muy similar a la de las Apps nativas, tanto en estética como en la manera de interactuar y navegar por ella.
- **Rapidez:** Las aplicaciones PWA tiene, por lo general, una velocidad de carga y de navegación bastante optimizada. Esto permite que los contenidos se muestren al usuario prácticamente al instante, ya que se apoyan en el almacenamiento en la caché, permitiendo al usuario tener una experiencia grata, al menos en ese aspecto.
- **Indexable y enlazable:** Al estar basadas en una aplicación web, el contenido de una PWA es rastreable e indexable, de forma que pueda aparecer como resultado en un buscador. Además, esta se puede compartir mediante una URL, con la posibilidad de que la otra persona la utilice sin necesidad de instalarlo. Esta es otra de las ventajas de uso respecto a una aplicación nativa 100 %.
- **Funcionalidades propias de una App nativa:** Las Progressive Web App pueden, por ejemplo, acceder a la geolocalización del dispositivo, al Bluetooth, sincronizarse en segundo plano o enviar notificaciones push. Estas notificaciones son una potente herramienta de comunicación que permite informar al usuario.

Más información en: [22]

1.4. Introducción a los Datos Abiertos

Según la Junta de Castilla y León, «Datos abiertos es una filosofía y práctica que persigue que determinados datos estén disponibles de forma libre a todo el mundo, sin restricciones de copyright , patentes u otros mecanismos de control. Los datos deben publicarse en bruto (sin procesar), bien estructurados y en formatos conocidos que faciliten la reutilización.»[10]

La filosofía detrás de los datos abiertos es el método científico que se basa en la investigación existente para desarrollar avances cuyo fin último es ayudar a las personas y al planeta que compartimos [32].

Todos los Datos Abiertos deben cumplir estos principios básicos:

- Los Datos Abiertos deben ser completos. Los datos públicos son datos que no están sujetos a limitaciones de privacidad, seguridad o privilegios válidos.
- Los Datos Abiertos deben tener gran nivel de detalle, deben presentarse exactamente igual que como surgieron de la fuente primitiva y se debe poder comprobarse su origen y las referencias que contienen, de forma que cualquiera pueda verificar su validez.
- Los Datos Abiertos se deberán poner a disposición tan pronto como sea necesario para preservar el valor de los mismos.
- Los Datos Abiertos deben de estar disponibles para toda la población, sin necesidad de registro alguno.
- Los Datos Abiertos se deberán estructurar para permitir procesamiento automatizado.
- Los Datos Abiertos deberán estar disponibles en un formato sobre el cual ninguna entidad tiene el control exclusivo.

Información más en detalle acerca de estos principios: [4]

En este proyecto, los datos abiertos que se van a utilizar son los dispuestos en el Boletín Oficial del Estado, y más concretamente los datos referidos a las oposiciones.

Más información acerca de los datos abiertos en la Comunidad de Castilla y León: [9]

1.5. Introducción al Desarrollo Ágil (SCRUM)

Debido a la importancia del proyecto, es necesario seguir un marco de trabajo en el que apoyarnos para que la aplicación se lleve a cabo. Para esto, el concepto de Desarrollo Ágil siguiendo la metodología SCRUM se adapta bastante bien a lo que se quiere. Una

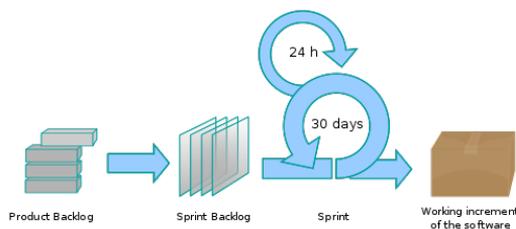


Figura 1.3: SCRUM

característica fundamental de SCRUM es el poder dividir el trabajo en distintas partes, denominados sprints. Estos sprints dan la capacidad de poder reaccionar rápidamente a los distintos cambios que puedan aparecer a lo largo del proyecto.

SCRUM también permite la adopción de una estrategia incremental, en lugar de otra más centrada en la planificación y ejecución completa del producto. [37]

1.5.1. Eventos

Para llevar un proyecto SCRUM correctamente es necesario conocer los distintos eventos que van a aparecer. Hay eventos cuyo objetivo es favorecer la comunicación y relación entre los distintos equipos de desarrollo que pueden haber en un proyecto. Este proyecto, solo existe un equipo de desarrollo formado por una sola persona, la cual es el alumno. Aún así, es interesante el introducir estos eventos:

- **Sprint** El Sprint es la base del Scrum. Es un periodo de tiempo de 1 mes, 3 o 2 semanas, durante el cual se crea un incremento de producto, utilizable y potencialmente liberable. Lo ideal es que siempre tengan la misma duración.
- **Sprint Planning** El trabajo que se realizará en el Sprint está planificado en Sprint Planning, mediante la colaboración de todo el equipo de Scrum. Esta reunión se planificará al inicio del proyecto.
- **Daily Scrum** Es una reunión diaria que sirve para inspeccionar el progreso hacia el objetivo del sprint y cómo avanza el progreso. Para ello, se inspecciona el trabajo realizado en el último día y se comenta el trabajo que se va a realizar ese día.
- **Sprint Review** La reunión se lleva a cabo al final del Sprint con el objetivo de inspeccionar el incremento y adaptar el Product Backlog (lista con todos los requerimientos iniciales del producto que se va a desarrollar) si es necesario.
- **Sprint Retrospective** Esta reunión se realiza en medio del sprint, antes del Sprint Planning siguiente y después del Sprint Review actual. Sirve para que el Scrum Team se inspeccione a sí mismo y cree un plan para mejorar en aquellas áreas que lo necesiten.

Información más detallada acerca de estos eventos en: [30] [31]

En posteriores capítulos, se hablará de cómo estos eventos serán adaptados al proyecto actual.

1.5.2. Roles

Antes de hablar de los documentos que hay dentro de la estrategia SCRUM, es imprescindible introducir los distintos miembros que forman el proyecto.

- **Product Owner** El objetivo de este rol es garantizar la calidad del producto. Es el encargado de comunicar las ideas del cliente, siendo una especie de representante de él. Tiene la capacidad de realizar cambios y tomar decisiones sobre el producto final.

- **SCRUM Master** Es el encargado de gestionar el proceso de trabajo dentro del proyecto. Entre sus obligaciones está la gestión de los sprints (incrementos) y maximizar la productividad del equipo de desarrollo.
- **Equipo de desarrollo** El número de integrantes dentro de este equipo es variable, pero siempre dentro de unos límites. Son los encargados del mismo desarrollo del proyecto.

1.5.3. Artefactos

Los artefactos son todos los elementos que garantizan la transparencia y registro de la información fundamental del proceso SCRUM. [33]

- **Product Backlog** Como anteriormente se ha comentado, el Product Backlog es una lista que contiene todos los requerimientos iniciales del producto que se va a desarrollar. El responsable de este documento es el Product Owner.
- **Sprint Backlog** Es un subconjunto del Product Owner destinado a recopilar las tareas y requisitos dentro de un sprint dado.
- **Incremento** Es el resultado del sprint. El sprint debe estar terminado y ser funcional para poderse crear el Incremento asociado. Este puede ser entregado y puede cumplir la función de prototipo para que los clientes puedan usarlo.

1.6. API REST

Una de las partes fundamentales de la aplicación es cómo el usuario puede acceder a los datos. Para este proyecto se ha decidido que el mecanismo de acceso a los datos sea a través de una API REST. Para entender el concepto de los Servicios REST, primero habrá que presentar lo que es un Servicio WEB.

A nivel conceptual, un servicio web es un componente software proporcionado a través de un endpoint (ordenador, móvil, etc) accesible a través de la red. Los servicios productores y consumidores utilizan mensajes para intercambiar información de invocaciones de petición y respuesta en forma de documentos auto-contenidos que hacen muy pocas asunciones sobre las capacidades tecnológicas de cada uno de los receptores [17].

Dentro de estos servicios, se encuentran los Servicios Web RESTful. Estos permiten intercambiar mensajes escritos en diferentes formatos, y no requieren el publicar una descripción de las operaciones que proporcionan, por lo que requieren una menor "infraestructura" para su implementación.

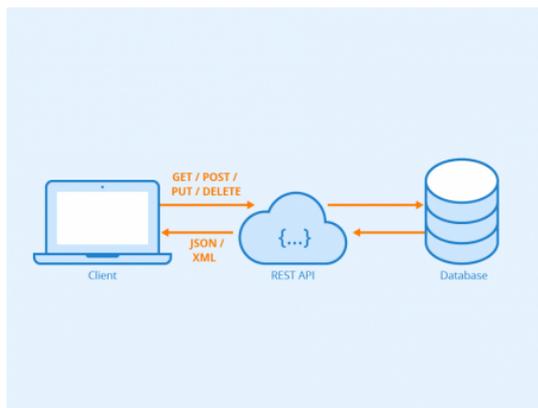


Figura 1.4: REST API - Author: Seobility - License: CC BY-SA 4.0

1.6.1. Características de los Servicios REST

El término REST proviene de la tesis doctoral de Roy Fielding y significa **RE**presentational **S**tate **T**ransfer. Este servicio tiene una serie de características [23]:

- Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite no tener que mantener información acerca del estado de ninguno de los procesos.
- Existen 4 tipos de operación relacionadas con los datos que son:
 - **POST**: Permite crear datos en la aplicación
 - **PUT**: Permite modificar datos de la aplicación
 - **GET**: Permite consultar y leer datos de la aplicación
 - **DELETE**: Permite eliminar datos de la aplicación.

La existencia de estos tipos de operación definidos, permite mantener en el servicio una interfaz uniforme, ya permite sistematizar el proceso con la información.

- Los objetos en REST se manipulan a partir de la URI: La URI sirve de identificador único para cada recurso. Esto permite acceder a la información para su modificación o consulta.

Este servicio ofrece la capacidad de separar el cliente y servidor, al separar la interfaz de usuario y el almacenamiento de datos. Esto implica la ventaja de poder tener un servicio escalable sin excesivos tipos de problema.

Una ventaja considerable de este servicio, y de la que se ha hecho utilidad en el proyecto, es la independencia del servicio del lenguaje que se usa. Esto es lo que permite al proyecto usar TypeScript en la capa de presentación, mientras que el servicio REST está basado en el lenguaje Java.

1.7. Objetivos

El objetivo de este proyecto es la creación de un servicio de búsqueda de oposiciones que permita al usuario ver el histórico de las oposiciones existentes, así como poder filtrar oposiciones mediante diferentes criterios de búsqueda. El servicio constará de una página web a la que puedan acceder los usuarios y que esta tenga soporte para móviles como si fuera una aplicación nativa.

La idea de la aplicación reside en su sencillez, que no abrume al usuario con opciones que no sabe qué significan, que ofrezca la posibilidad de acceder a las oposiciones desde una sencilla búsqueda por directorio, y también que ofrezca una búsqueda avanzada.

1.8. Estructura de la memoria

La memoria del proyecto está estructurada de la siguiente manera:

- **Capítulo 2: Tecnologías utilizadas** Enumeración de las herramientas utilizadas en el desarrollo de la aplicación y de la gestión del mismo.
- **Capítulo 3: Planificación y Requisitos** Descripción de la adaptación al Desarrollo Ágil y el público objetivo del que se espera que use la aplicación, así como la serie de riesgos que se han descubierto en el proyecto.
- **Capítulo 4: Análisis** Explicación de la etapa de análisis que ha tenido el proyecto, así como una descripción del Boletín Oficial del Estado y de sus datos.
- **Capítulo 5: Diseño** Etapa de diseño software que contiene los diagramas y diseños basados en en la etapa de análisis.
- **Capítulo 6: Implementación y Pruebas** Descripción del proceso de implementación y de las pruebas realizadas a la aplicación.
- **Capítulo 7: Seguimiento** Describe el desarrollo del proyecto dividido en sprints siguiendo un desarrollo ágil basado en una adaptación de SCRUM.
- **Capítulo 8: Conclusiones**
- **Anexo A: Manual** Contiene una breve explicación de la instalación de las herramientas, instalación del proyecto y una breve introducción a las funcionalidades de la aplicación.

Dentro del contenido de la memoria, se utilizarán indistintamente los términos de "epígrafe" y "especialidad, ya que estos se refieren al mismo concepto y, dependiendo el contexto, es necesario utilizar un término u otro. El término epígrafe aparece dentro de los datos que ofrece el boletín y se sustituye dentro de la aplicación por el término especialidad, ya que este ofrece una visión más clara de lo que representa. Se usará epígrafe para referirse a la parte técnica de la aplicación y el término especialidad cuando este involucre de alguna forma al usuario final.

Capítulo 2

Tecnologías utilizadas

2.1. IDE's

Los entornos de desarrollo integrado aumentan la productividad del programador al combinar actividades comunes de escribir software en una sola aplicación: editar código fuente, compilar ejecutables y depurar.

2.1.1. NetBeans

Para la parte Backend se ha decidido usar un IDE que ofrezca funcionalidades de alojamiento de servicios REST, así como alojamiento de bases de datos. NetBeans ofrece ambas posibilidades, así como otras herramientas y funcionalidades que se han valorado positivamente a la hora de elegir este entorno de desarrollo integrado como parte del proyecto.

Es un entorno de desarrollo integrado libre, orientado principalmente al desarrollo de aplicaciones Java. Contiene muchas funcionalidades, para distintos tipos de aplicaciones y para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan. NetBeans proporciona integración con algunas herramientas y servicios externos que van a ser utilizados en este proyecto. El uso de este entorno se focaliza en la parte de Backend del proyecto, alojando los servicios REST de la aplicación, así como la base de datos [8].

Para acceder a su página oficial: [26]

2.1.2. Visual Studio Code

Así como tenemos un Entorno de Desarrollo Integrado para la parte Backend de la aplicación, en este proyecto se usará otro IDE con distintas funcionalidades que permitan desa-

rollar la parte Frontend de la aplicación.

Visual Studio Code es un editor de código fuente desarrollado por Microsoft y está presente en muchas plataformas (Windows, Linux y macOS). Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código [38].

Este entorno de desarrollo soporta numerosos Frameworks, entre estos están Angular e Ionic que van a ser una pieza fundamental del proyecto. El uso de este entorno corresponde a la parte Frontend del proyecto.

Para acceder a su página oficial: [24]

2.2. Diseño

Una vez que se analizan y especifican los requisitos, el diseño es la siguiente actividad técnica a realizar. Es la etapa en la que se fomenta la calidad. Sin diseño se corre el riesgo de construir una aplicación inestable, difícil de trabajar con ella y complicada de testear.[5]

2.2.1. Astah profesional

Para esta etapa, se necesitará una aplicación que permita crear diagramas UML con claridad y que sea fácil de usar. También, se valorará las distintas funcionalidades que tenga la herramienta, esta propiedad elimina herramientas sencillas de usar como "draw.io" que permite crear diagramas UML, pero que su funcionalidad se limita en gran manera a la hora de crear distintos tipos de diagrama y que no la diferencia mucho de un lápiz y un papel.

Astah Profesional es una herramienta de diseño de sistemas que soporta UML creada por la compañía japonesa Change Vision. Esta aplicación es fundamental para el modelado y diseño software.[34]

La razón de su uso se encuentra en que históricamente, durante el transcurso de la carrera, se ha utilizado de manera frecuente. Se podría haber utilizado en su lugar la aplicación Visual Paradigm, que también se ha utilizado en varias asignaturas, pero la experiencia de uso, históricamente, ha sido mejor en la aplicación Astah.

Para acceder a la página oficial: [2]

2.3. Pencil Project

El diseño de la aplicación es una parte fundamental. El éxito de una aplicación depende en gran manera del diseño. Por estos motivos, es clave tener una visión clara de cómo se

quiere que sea la aplicación, y para ello la necesidad de tener bocetos, los cuales servirán de guía en la implementación.

Pencil Project es una herramienta de creación de prototipos de GUI de código abierto que ayuda a desarrolladores y empresas a crear wireframes y maquetas del sitio usando varias herramientas integradas. [6]

La aplicación ofrece soporte para la creación de interfaces, tanto en dispositivos móviles, como en dispositivos de escritorio. Su facilidad de uso, sus funcionalidades, el ser código abierto, entre otros motivos, hace que esta aplicación sea la elegida para este apartado.

Para acceder a la página oficial: [16]

2.4. Frameworks

Un Framework es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado.

2.4.1. Ionic

Como ya se ha dicho, la aplicación deberá tener soporte para aplicaciones móviles. Ya se ha presentado la idea de un aplicación PWA (Aplicaciones Web Progresivas). Para eso se necesitará una herramienta que pueda convertir la aplicación en una que pueda crear experiencias similares a las aplicaciones nativas en dispositivos móviles y de escritorio.

Ionic es un SDK de código abierto para desarrollo de aplicaciones móviles híbridas. Proporciona herramientas y servicios para el desarrollo de aplicaciones móviles híbridas, aplicaciones de escritorio y PWA, utilizando tecnologías tales como CSS, HTML5 y SAS.[20] Este SDK está construido sobre el framework AngularJs, y aunque en versiones posteriores den soporte a otros frameworks, en este proyecto se va a seguir utilizando la versión AngularJS.

Para acceder a la página oficial: [19]

2.4.2. AngularJS

Habiendo presentado el framework que nos permitirá tener una aplicación PWA, ahora nos toca hablar sobre el framework en el que se basa Ionic.

AngularJS es un framework basado en JavaScript y creado por la compañía de Google que permite crear páginas web dinámicas. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.[35]

Para acceder a la página oficial: [18]

2.5. Lanzamiento

En esta sección agrupamos las herramientas que servirán para alojar los datos de la aplicación en una base de datos, así como alojar la propia aplicación en un servidor.

2.5.1. Glassfish

Un servicio online siempre va a necesitar un servidor en el que pueda estar alojada la aplicación y que esta pueda ser accedida por distintos usuarios. En este proyecto se usará el servidor Glassfish para alojar el servicio REST de la aplicación.

Glassfish es un servidor de aplicaciones software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.[3]

Para acceder a la página oficial: [27]

2.5.2. Apache Derby

Ya tenemos dónde alojar los servicios REST de la aplicación, ahora hace falta ver la base de datos donde almacenaremos los datos de la aplicación (oposiciones, departamentos, epígrafes, etc.)

Apache Derby es una base de datos relacional de código abierto programada completamente en Java y mantenida por la Fundación Apache. Se basa en los estándares SQL y JDBC y mantiene un tamaño liviano de tan solo unos pocos Mb (2.6 Mb).[7]

Para acceder a la página oficial: [1]

2.6. Testing

El testeo de las aplicaciones de un proyecto es una parte fundamental del mismo. La fase de testeo ayuda a localizar errores de la aplicación, así como implementar una metodología de trabajo.

Esta parte se deberá realizar en paralelo al proceso de desarrollo del software. Esta práctica ayuda a tener un proyecto libre de errores y también a evitar contratiempos innecesarios por detectar un fallo fuera de tiempo.

2.6.1. Postman

Una parte esencial de la aplicación son los servicios REST. Estos dispondrán los datos que van a llegar al usuario final. Debido a esto, es imprescindible tener un mecanismo que nos asegure que los datos van a ser los correctos. Para esto, usaremos la herramienta Postman:

Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas.[21]

Gracias a esta herramienta, además de testear, consumir y depurar API REST, podremos monitorizarlas, escribir pruebas automatizadas para ellas, documentarlas, mockearlas, simularlas, etc.

Para acceder a la página oficial: [28]

2.6.2. Mozilla Firefox

Para comprobar el funcionamiento de una aplicación web, es fundamental tener un navegador en el que se pueda acceder a la funcionalidad de la página. Firefox es un navegador web que ofrece al usuario una experiencia de testeo bastante fructífera, ya que permite cambiar desde una perspectiva de escritorio a una móvil simplemente con un botón.

También ofrece un terminal en el que poder comprobar los mensajes de la consola. Esto es bastante útil a la hora de poder comprobar el historial de procesos que realiza la aplicación.

Mozilla Firefox (o simplemente Firefox) es un navegador web libre y de código abierto desarrollado para distintas plataformas, está coordinado por la Corporación Mozilla y la Fundación Mozilla. Usa el motor Gecko para renderizar páginas web, el cual implementa actuales y futuros estándares web. [36]

Para acceder a la página oficial: [25]

Capítulo 3

Planificación y requisitos

3.1. Desarrollo Ágil (SCRUM) adaptado al proyecto

Habiendo explicado en capítulos anteriores el concepto de Desarrollo Ágil, y en particular la estrategia SCRUM, podemos empezar a explicar cómo se va a adaptar para el proyecto. El proyecto contiene diferentes particularidades que le hace tener un tratamiento especial acerca de la estrategia SCRUM:

- El primero, es el número de miembros del Equipo de Desarrollo. Obviamente, este está formado por una sola persona, la cual es el alumno. Si nos ceñimos a los límites dispuestos por la estrategia SCRUM acerca del número de miembros (de 3 a 9 miembros por equipo de desarrollo), este requisito no se cumple.

Esta particularidad hace que simplifiquemos en gran medida esta estrategia. Las reuniones destinadas a monitorizar el trabajo de los distintos miembros del equipo se eliminarán, ya que todo el peso del proyecto irá a una misma persona. El resto de reuniones se reducirá y la comunicación entre los roles del proyecto se hará por correo electrónico.

- El proyecto está destinado a hacerse durante el curso. Esto dificulta el poder planificar reuniones y sprints con mucha antelación, debido a la existencia de otras variables, como lo son las prácticas curriculares, las otras asignaturas, junto con sus exámenes y prácticas.

Debido a esto, la fecha de los sprints podrá variar de la estipulada en la planificación. Así mismo, la posibilidad de hacer reuniones periódicas tampoco es una opción posible por la misma razón.

Habiendo aclarado estos detalles, comencemos con la asignación de roles en el proyecto.

- Los tutores del proyecto representarán dos roles: El de Product Owner al ser los responsables de garantizar la calidad del producto, así como los encargados de realizar cambios y modificar los requisitos en el producto final.

También, son responsables de gestionar todo el proceso de trabajo, gestionar los incrementos y las reuniones. Estos deberes están asociados al rol de SCRUM Master.

- El alumno será el único miembro del Equipo de Desarrollo. Será el encargado de cumplir los objetivos del proyecto y de acudir a las reuniones organizadas por los tutores para ver el desarrollo del mismo.

Al principio del proyecto, se realizó una reunión explicativa que actuaría como Sprint Planning, donde se habló de los objetivos del proyecto y los requisitos que tendría este.

Posteriormente, se realizarían comunicaciones escritas donde ambas partes comentarían el progreso del proyecto. Estas comunicaciones solían tener documentación visual de la aplicación que funcionaba como demostración de lo que se había hecho durante el sprint. Estas comunicaciones actuarían como Sprint Planning, Sprint Review y Sprint Retrospective.

Tanto las historias de usuario como el Product Backlog Final se confeccionaron a medida que el proyecto fue desarrollándose.

3.2. Público objetivo

Una de las peculiaridades de este proyecto es la amplitud del público objetivo, debido al gran número de variedad que hay en las oposiciones. Hay oposiciones que exigen un número de requisitos y hay oposiciones que no exigen prácticamente nada. La característica común que tienen los miembros del público objetivo es que están interesados en una oposición en particular o en un grupo de ellas. Por esto, la aplicación deberá tener unos requisitos mínimos que sepan satisfacer las necesidades de cada tipo de usuario.

Según los datos recabados por la aplicación, al año se producen de media 18000 documentos referentes a oposiciones, entre ellos están las mismas oposiciones y correcciones. Para cada oposición, hay un público objetivo que está interesado en estas. Estas oposiciones son muy variadas, tanto en requisitos como en los puestos que ofrecen.

Las personas que estén interesadas en las oposiciones no tienen por qué saber de informática. Por este motivo, la aplicación deberá mantener un diseño sencillo, con funcionalidades sencillas que ayuden a este usuario a llegar a su objetivo.

También se ha tenido en cuenta a los usuarios más experimentados dentro del ámbito de internet. La aplicación cuenta con un sistema de búsqueda por directorio, el cual está centrado en una experiencia más técnica por parte de los usuarios.

Adicionalmente, la aplicación cuenta con un histórico de cambios dentro de las oposiciones, que permite a los usuarios interesados en una única oposición ver los cambios que ha tenido esta.

3.3. Riesgos

Los riesgos son un evento que, si sucede, tiene un efecto negativo en uno o varios de los objetivos dispuestos del proyecto.

A continuación, se dispondrán varios de estos riesgos que pueden suceder a lo largo del proyecto:

Riesgo 1: Modificación del formato del BOE	
Tipo	Técnico
Probabilidad	Muy baja
Impacto	Medio
Descripción	La base de datos depende mucho de la forma en la que se disponen los datos en el Boletín Oficial del Estado, y en detalle la forma en la que dispone la información referente a las oposiciones.
Mitigación	Depender lo mínimo de estos datos. Guardar solo lo indispensable en la base de datos.
Contingencia	Evitar mostrar al usuario datos erróneos correspondientes a este fallo, y mostrar un mensaje de error en su lugar para evitar confusiones.

Tabla 3.1: Riesgo de modificación del formato del BOE

Riesgo 2: Modificación del formato de las Oposiciones	
Tipo	Técnico
Probabilidad	Muy baja
Impacto	Medio
Descripción	La base de datos depende mucho de la forma en la que se disponen los datos de las oposiciones, así como en la forma en la que se accede a ellos (el formato de la dirección).
Mitigación	Depender lo mínimo de estos datos. Guardar solo lo indispensable en la base de datos.
Contingencia	Evitar mostrar al usuario datos erróneos correspondientes a este fallo, y mostrar un mensaje de error en su lugar para evitar confusiones y para que los técnicos conozcan este fallo.

Tabla 3.2: Riesgo de modificación del formato de las Oposiciones

Riesgo 3: Falta de tiempo	
Tipo	Personal
Probabilidad	Alta
Impacto	Medio
Descripción	Este proyecto está destinado a hacerse en el segundo cuatrimestre. El alumno está desarrollando este Trabajo de Fin de Grado junto a 2 asignaturas y las prácticas de empresa. Debido a esto, habrá ocasiones en las que la carga de trabajo no permita desarrollar como se quisiera este proyecto.
Mitigación	Realizar un plan de trabajo teniendo en cuenta los espacios de tiempo que tenga poco margen de trabajo. Por ejemplo: viendo en la planificación de las asignaturas los posibles picos de trabajo (entregas, exámenes, etc.)
Contingencia	Replanificar la carga de trabajo destinada al TFG en las zonas de más descanso.

Tabla 3.3: Riesgo de falta de tiempo

Riesgo 4: Pandemia	
Tipo	Global
Probabilidad	Alta
Impacto	Medio
Descripción	Debido a la situación en la que nos encontramos, no extraño pensar en la aparición de una nueva variante o del incremento del número de casos de COVID-19 que conlleven medidas excepcionales del gobierno que alteren la forma de trabajo.
Mitigación	Mantener las medidas de seguridad que el gobierno comunique a lo largo del proyecto. Utilizar entornos de trabajo online.
Contingencia	Realizar reuniones de forma telemática. Reorganizar la carga de trabajo si se ha impedido trabajar en el proyecto por causas relacionadas con la pandemia.

Tabla 3.4: Riesgo de pandemia

Riesgo 5: Interfaz mal elegida	
Tipo	Técnico
Probabilidad	Media
Impacto	Alto
Descripción	Problemas relacionados con la disposición de los componentes de la interfaz que provoquen una mala experiencia por parte del usuario.
Mitigación	Seguir estándares y guías en la construcción de la interfaz. Tener pruebas de usabilidad con usuarios reales y tener en cuenta sus opiniones.
Contingencia	Seguir las recomendaciones de los usuarios y comprobar si se sigue la metodología recomendada por parte de guías y estándares actuales. Realizar pruebas de usabilidad antes de volver a reanudar la aplicación

Tabla 3.5: Riesgo de interfaz mal elegida

Riesgo 6: Errores en la disposición de los datos	
Tipo	Técnico
Probabilidad	Media
Impacto	Alto
Descripción	Problemas relacionados con los datos que reciba el usuario en la propia página.
Mitigación	Pruebas de caja negra en la parte Backend.
Contingencia	Poner en mantenimiento la aplicación e intentar localizar la raíz del problema y actualizar las pruebas para que den los resultados correctos.

Tabla 3.6: Riesgo de errores en la disposición de los datos

Riesgo 7: Modificación de los requisitos	
Tipo	Contractual
Probabilidad	Media
Impacto	Medio
Descripción	Durante el desarrollo de la aplicación es posible que haya cambios en el concepto del proyecto y que estos impliquen la modificación de los requisitos del trabajo. Por ejemplo: acotar el público objetivo a uno con más nivel informático y ampliar los métodos de búsqueda, así como la propia interfaz
Mitigación	Flexibilizar las partes de la aplicación con mayor posibilidad de cambio. Postergar las partes con mayor posibilidad de cambio en la planificación para que estos cambios no penalicen lo ya hecho en la aplicación.
Contingencia	Replanificar la carga de trabajo

Tabla 3.7: Riesgo de modificación de los requisitos

Capítulo 4

Análisis

4.1. Análisis del BOE

De acuerdo con el Real Decreto 181/2008, de 8 de febrero el "Boletín Oficial del Estado", diario oficial del Estado español, es el medio de publicación de las leyes, disposiciones y actos de inserción obligatoria.

En el "Boletín Oficial del Estado" se publican:

1. Las disposiciones generales de los órganos del Estado y los tratados o convenios internacionales.
2. Las disposiciones generales de las Comunidades Autónomas, de acuerdo con lo establecido en los Estatutos de Autonomía y en las normas con rango de ley dictadas para el desarrollo de los mismos.
3. Las resoluciones y actos de los órganos constitucionales del Estado, de acuerdo con lo establecido en sus respectivas leyes orgánicas.
4. Las disposiciones que no sean de carácter general, las resoluciones y actos de los departamentos ministeriales y de otros órganos del Estado y Administraciones públicas, cuando una ley o un real decreto así lo establezcan.
5. Las convocatorias, citaciones, requisitorias y anuncios cuando una ley o un real decreto así lo establezcan.

La publicación del boletín oficial del estado se realiza en la página web oficial todos los días, menos los domingos. La página ofrece dos formatos de lectura: PDF y XML. El acceso a estos formatos está estandarizado, y se puede acceder a distintas boletines siguiendo un

https://boe.es/diario_boe/xml.php?id=BOE-S-YYYYMMDD

simple formato. Por ejemplo, el tipo de archivo de lectura utilizado en el proyecto sigue este formato:

Siendo YYYY, MM y DD el año mes y día en formato ISO.

Gracias a este formato, podemos automatizar la recogida de boletines diariamente, simplemente añadiendo el día deseado.

Dentro del boletín nos lo encontramos divididos en secciones:

- I. **Disposiciones generales:** se publican las leyes orgánicas, leyes, reales decretos legislativos y reales decretos-leyes; los tratados y convenios Internacionales; las normas con rango de ley de las Comunidades Autónomas; los reglamentos y demás disposiciones de carácter general estatales; los autos y providencias del Tribunal Constitucional en procedimientos relativos a disposiciones de carácter general así como las decisiones del Tribunal Supremo que anulan o afectan a disposiciones estatales de carácter general.
- II. **Autoridades y personal:** Dividida por dos subsecciones
 - a) **Nombramientos, situaciones e incidencias:** se publican los nombramientos y ceses de altos cargos y del personal de la Administración.
 - b) **Oposiciones y concursos:** se publican las convocatorias de oposiciones y concursos para ingreso en la Administración y para la provisión de puestos de trabajo.
- III. **Otras disposiciones:** se publican las disposiciones que no tengan carácter general ni correspondan a las demás secciones: bases reguladoras de la concesión de becas, premios y otras ayudas y subvenciones; cartas de servicios; convenios colectivos de ámbito general; planes de estudio, declaraciones de impacto ambiental, convenios entre Administraciones, etc.
- IV. **Administración de Justicia:** se publican los edictos, notificaciones, requisitorias y anuncios de los Juzgados y Tribunales.
- V. **Anuncios:** Dividida por tres subsecciones:
 - a) **Contratación del Sector Público:** Se publican los anuncios de contratación.
 - b) **Otros anuncios oficiales:** Se publican los extractos de convocatorias de becas, premios y otras ayudas y subvenciones, los trámites de información pública, las concesiones administrativas, etc
 - c) **Anuncios particulares.**

Más información de las secciones del boletín en: [14]

Dentro de estas secciones nos interesa la sección II.b donde se publican las oposiciones y concursos que se van a utilizar como datos en la aplicación.

4.2. Oposiciones y concursos

En esta sección del boletín se enumeran las oposiciones y concursos que los distintos organismos emiten. En ambos formatos (PDF y XML) aparece el nombre de los departamentos que emiten las disposiciones, así como los "epígrafes" (especialidad) en los que entra cada oposición. A continuación, aparecen las oposiciones que esta pareja Departamento-Epígrafe ha emitido. Aparece el título de la oposición, así como el identificador de esta. El identificador de la oposición está formado por una serie de caracteres compuestos por el año de la disposición de la oposición y del número de oposición total de ese año.

Entre ambos formatos de archivo existen varias diferencias que hace al formato XML un formato más completo. En el formato XML aparece el identificador de los departamentos que nos servirá de ayuda para guardarlos. También, otro de los puntos a favor de este tipo de archivo, es que aparecen los enlaces a las páginas donde reside la información más detallada de cada oposición (un enlace para su formato XML y otro para PDF). Figura 4.1

```

▼ <departamento nombre="MINISTERIO DE DEFENSA" eta="6110">
  ▼ <epigrafe nombre="Funcionarios de la Administración del Estado">
    ▼ <item id="BOE-A-2020-1541" control="2020/1090">
      <titulo>Resolución 430/38015/2020, de 22 de enero, de la Subsecretaría, por la que se convoca la provisión de puesto de trabajo por el sistema de libre designación.</titulo>
      <urIPdf s2Bytes="221722" s2KBytes="217" numPag="3"/>/boe/dias/2020/02/01/pdfs/BOE-A-2020-1541.pdf</urIPdf>
      <urIHtm>/diario_boe/txt.php?id=BOE-A-2020-1541</urIHtm>
      <urIXml>/diario_boe/xml.php?id=BOE-A-2020-1541</urIXml>
    </item>
    ▼ <item id="BOE-A-2020-1542" control="2020/1303">
      <titulo>Resolución 430/38016/2020, de 24 de enero, de la Subsecretaría, por la que se convoca la provisión de puesto de trabajo por el sistema de libre designación.</titulo>
      <urIPdf s2Bytes="222824" s2KBytes="218" numPag="3"/>/boe/dias/2020/02/01/pdfs/BOE-A-2020-1542.pdf</urIPdf>
      <urIHtm>/diario_boe/txt.php?id=BOE-A-2020-1542</urIHtm>
      <urIXml>/diario_boe/xml.php?id=BOE-A-2020-1542</urIXml>
    </item>
  </epigrafe>
</departamento>

```

Figura 4.1: Ejemplo real de dos oposiciones con un mismo par Departamento-Epígrafe [13]

4.3. Oposición

En esta sección se hablará solo acerca del formato XML, porque como ya se ha comentado, ofrece una gran variedad de información que el formato PDF no posee. Además, este formato será el que va a ser utilizado posteriormente en el desarrollo de la aplicación, debido a las ventajas anteriormente comentadas y al propio formato del archivo que es uno de los estándares de lenguajes para intercambio de datos estructurados.

Dentro del fichero XML podemos encontrar que la información está dividida dentro de 3 subsecciones:

- **Metadatos:** Aquí hay información muy diversa. Dentro de este apartado nos podemos encontrar el identificador de la oposición, el título de la oposición, el departamento junto a su identificador y la página que nos dirigirá al formato PDF.
- **Análisis:** Dentro de esta subsección podemos encontrar diversos datos, entre estos el más destacado son las referencias. Para esta parte hay que explicar un detalle. Hay disposiciones que pueden referirse a correcciones o continuaciones de alguna oposición anterior. Para este propósito están las referencias. Estas referencias sirven para conocer

4.4. REFERENCIAS

```
▼<documento fecha_actualizacion="20200706105601">
  ▼<metadatos>
    <identificador>BOE-A-2015-12582</identificador>
    <título>Resolución de 16 de noviembre de 2015, de la Dirección de la Agencia Estatal Boletín Oficial del Estado, por la que se convoca concurso específico para la provisión de puestos de trabajo.</título>
    <diario codigo="BOE">Boletín Oficial del Estado</diario>
    <diario_numero>279</diario_numero>
    <seccion>2</seccion>
    <subseccion>B</subseccion>
    <departamento codigo="7710">Ministerio de la Presidencia</departamento>
    <rango codigo="1370">Resolución</rango>
    <numero_oficial/>
    <fecha_disposicion>20151116</fecha_disposicion>
    <fecha_publicacion>20151121</fecha_publicacion>
    <fecha_vigencia/>
    <fecha_derogacion/>
    <letra_imagen>A</letra_imagen>
    <pagina_inicial>109849</pagina_inicial>
    <pagina_final>109862</pagina_final>
    <suplemento_letra_imagen/>
    <suplemento_pagina_inicial/>
    <suplemento_pagina_final/>
    <estatus_legislativo/>
    <origen_legislativo codigo="1">Estatal</origen_legislativo>
    <estado_consolidacion codigo=""/>
    <judicialmente_anulada>N</judicialmente_anulada>
    <vigencia_agotada>N</vigencia_agotada>
    <estatus_derogacion>N</estatus_derogacion>
    <url_epub/>
    <url_pdf>/boe/dias/2015/11/21/pdfs/BOE-A-2015-12582.pdf</url_pdf>
    <url_pdf_catalan/>
    <url_pdf_euskera/>
    <url_pdf_gallego/>
    <url_pdf_valenciano/>
  </metadatos>
  ▶< analisis>
    ...
  </ analisis>
  ▼< texto>
    <p class="parrafo">Existiendo puestos de trabajo vacantes en la Agencia Estatal Boletín Oficial del Estado, dotados presupuestariamente, y cuya cobertura se estima conveniente en atención a las necesidades del servicio, la Agencia Estatal Boletín Oficial del Estado, previa autorización de las bases de la presente convocatoria por la Secretaría de Estado de Administraciones Públicas, y de acuerdo con lo previsto en el artículo 20.1.a) y c) de la Ley 30/1984, de 2 de agosto, de Medidas para la Reforma de la Función Pública, modificado por la Ley 13/1996, de 30 de diciembre, y preceptos concordantes del Reglamento General de Ingreso del Personal al Servicio de la Administración General del Estado y de Provisión de Puestos de Trabajo y Promoción Profesional de los Funcionarios Civiles de la Administración General del Estado, y con lo señalado en el artículo 20.1 de la Ley 28/2006, de 18 de julio, de Agencias Estatales para la Mejora de los Servicios Públicos, ha dispuesto convocar concurso específico de méritos.</p>
    <p class="parrafo">El presente concurso, de acuerdo con el artículo 14 de la Constitución Española y la Directiva 2006/54/CE de 5 de julio de 2006, tiene en cuenta el principio de igualdad de oportunidades e igualdad de trato entre hombres y mujeres</p>
  </ texto>
</ documento>
```

Figura 4.2: Ejemplo real de una oposición en formato XML [12]

si esta oposición ha tenido alguna corrección en el futuro y poder acceder a ella, o saber a qué disposición se está refiriendo la oposición actual.

- Texto Aquí se incluye el contenido de la oposición en formato html. Es donde se encuentra la información más útil y es la que menos estandarizada está. Dependiendo la institución, la información que aparece aquí puede estar dispuesta en muchos sitios diferentes, lo que hace muy difícil extraer datos automáticamente.

4.4. Referencias

Este capítulo tiene cierta semejanza con las clases dispuestas en la aplicación, y por eso en esta sección se hablará de las referencias que pueden tener las oposiciones. Estas permiten a la aplicación saber el historial de cambios de las oposiciones, así como el estado de estas. El problema de este método es que su disposición depende del Departamento que las emite. Esto implica que hay oposiciones desactualizadas, ya que el Departamento no ha emitido los cambios y hay oposiciones en las que es muy difícil

saber su estado real. Estas referencias están solo en el formato XML.

Se puede distinguir dos tipos de referencias:

- Posteriores: Estas se encuentran en las oposiciones "normales", las disposiciones que convocan una oposición. Estos documentos a los que se hace referencia, son los que realizan alguna modificación a las oposiciones que hacen referencia.
- Anteriores: Si referencias posteriores son las que hacen referencia a los documentos que realizan cambios, las referencias anteriores son las que referencian a las que afectan estos cambios.

Ambos tipos de referencias contienen la misma información, y esta es la siguiente:

```

▼<referencias>
  <anteriores/>
  ▼<posteriores>
    ▼<posterior referencia="BOE-A-2015-12743" orden="">
      <palabra codigo="201">CORRECCIÓN de errores</palabra>
      <texto>en BOE núm. 282 de 25 de noviembre de 2015</texto>
    </posterior>
  </posteriores>
</referencias>

```

Figura 4.3: Ejemplo real de una oposición que contiene una referencia posterior

- Identificación Posterior: Estas se encuentran en las oposiciones "normales", las disposiciones que convocan una oposición. Estas oposiciones a las que se hace referencia, son las que realizan alguna modificación a la oposición del documento.
- Identificación Anterior: Si las identificaciones posteriores son las que hacen referencia a las oposiciones que realizan cambios, las identificaciones anteriores son las que referencian a las que afectan estos cambios.
- Palabra: El término de este atributo es bastante poco preciso, pero es el dispuesto en el documento oficial. Se podría usar mejor el concepto de tipo de cambio, ya que aquí es donde reside el cambio que hace

4.5. Casos de Uso

Los Casos de Uso contienen una descripción textual de la forma en la que los usuarios podrán interactuar con la aplicación. Estos casos de uso se han tenido en cuenta para la implementación de la funcionalidad de la aplicación.

A continuación, se dispondrá la serie de Casos de Uso asociados al proyecto:

4.5.1. Listas

Las listas son el mejor método para enseñar a los usuarios distintos tipos de datos. Dentro del proyecto se pueden distinguir tres tipos de listas:

- Ver listado de oposiciones: Es la principal funcionalidad que deberá tener la aplicación. Un gestor de oposiciones que no te permita tener un listado de estas es un concepto poco útil. El listado estará ordenado según la fecha de la oposición, poniendo las más recientes primero e indicando el estado de estas.
- Ver listado de los departamentos: Normalmente, el usuario no conoce los distintos tipos de departamentos que tienen la posibilidad de emitir una oposición en el BOE. Dentro de los departamentos están los ministerios, comunidades autónomas, etc. Por eso, es interesante que los usuarios tengan la posibilidad de ver qué departamentos hay. Adicionalmente, se podrá buscar los departamentos por un método de búsqueda por palabra.
- Ver listado de los epígrafes: La misma razón por la cual es interesante dejar al usuario ver los tipos de departamento que hay, es aplicable a los epígrafes. Los epígrafes contienen la información que puede ser entendida como las especialidades que buscan las oposiciones (Cuerpo de la Guardia Civil, Notarios, Personal Funcionario, etc). También permitirá al usuario buscar estos epígrafes por palabra.

4.5.2. Búsqueda de oposiciones según distintos criterios

Una de las funcionalidades que deberá tener la aplicación es la posibilidad de permitir al usuario buscar un conjunto de oposiciones según varios criterios. El sistema de búsqueda deberá permitir al usuario combinar los distintos tipos de criterio, en orden de dejar al usuario la mayor cantidad de posibilidades de búsqueda. Estos criterios son los siguientes:

- Fechas: Este criterio abarca dos tipos de funcionalidad: búsqueda por una fecha específica y por un rango de fechas. Ambas peticiones devolverán un conjunto de oposiciones que cumplan con la consulta.
- Estado: Existen cuatro tipos de estado en cada oposición: aprobada, cancelada, en vigor y corrección. Cada estado tiene su significado y se dejará al usuario consultar cada oposición según el tipo de esta.
- Departamento: Cada oposición tiene un Departamento asociado a él. Podría ser interesante para el usuario filtrar las oposiciones según este criterio, ya que dentro de los departamentos están las comunidades autónomas que emiten las oposiciones, las cuales pueden servir para localizar el lugar donde se va a realizar la oposición.
- Epígrafe: Como con los departamentos, cada oposición está asociado a un Epígrafe (especialidad). Esto permitirá al usuario buscar un conjunto de oposiciones que tengan la misma especialidad.
- Título: Cada oposición tiene un título. La información contenida en los títulos no es muy destacable, debido a que la información que hay hace referencia a criterios ya estandarizados (contiene la fecha del documento, el tipo de este y el departamento que emite). Aun así, es interesante dejar este criterio al usuario por si prefiere dejar en un mismo campo toda la información.

4.5.3. Búsqueda por directorio

La idea detrás de este caso de uso reside en la existencia de usuarios que puedan querer buscar las oposiciones utilizando un sistema de directorios. Estos serían los epígrafes y los departamentos.

El concepto se puede explicar con un sencillo ejemplo: Un usuario que resida en Castilla y León y que no tenga interés en ninguna especialidad y que quiera ver las oposiciones que hay en su comunidad. Este usuario accederá al departamento de la Comunidad de Castilla y León y este le llevará a un listado de los epígrafes asociados al departamento. Aquí podrá ver el "historial" de epígrafes que ha emitido el departamento, si selecciona no de estos podrá ver el conjunto de oposiciones que tiene el mismo par Departamento-Epígrafe.

Este ejemplo es igualmente válido para el caso contrario: Un usuario que esté interesado en oposiciones de la Guardia Civil y que esté interesado en ver qué departamentos han emitido oposiciones acerca de esa especialidad, y a su vez ver las oposiciones que han emitido.

Capítulo 5

Diseño

5.1. Diagrama de Casos de Uso

Con este diagrama buscamos representar la serie de actividades que puede realizar un usuario con la aplicación:

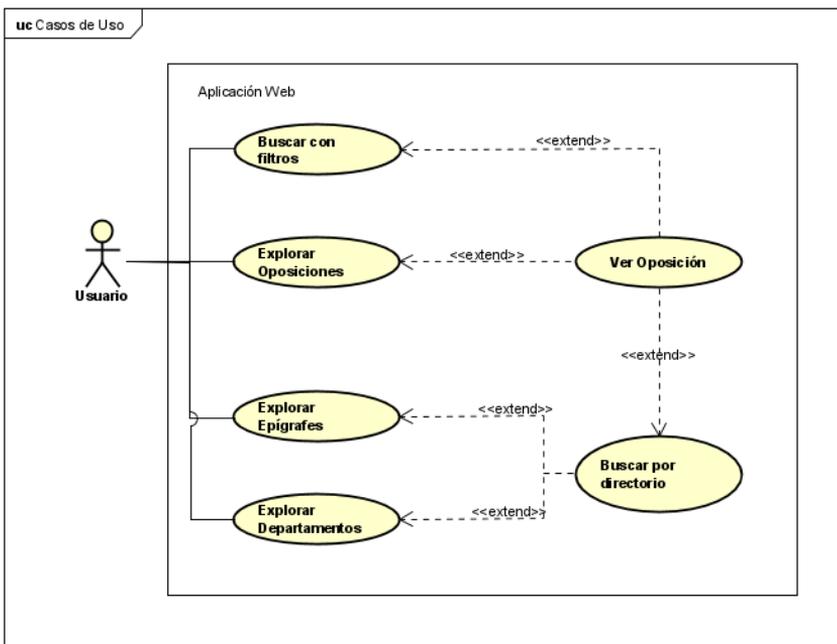


Figura 5.1: Diagrama de Casos de Uso

5.1. DIAGRAMA DE CASOS DE USO

A continuación, se realizará una descripción detallada de estos casos de uso:

Caso de Uso	Explorar Oposiciones
Descripción	Permite al usuario navegar por las oposiciones de la aplicación
Autor	Usuario
Precondiciones	Ninguna
Flujo de Eventos	El usuario selecciona dentro del menú la opción Oposiciones Se le muestra al usuario todas las oposiciones ordenadas de más recientes a más antiguas.
Post-condiciones	Ninguna

Tabla 5.1: Descripción del Caso de Uso Explorar Oposiciones

Caso de Uso	Explorar Departamentos
Descripción	Permite al usuario navegar por los departamentos de la aplicación
Autor	Usuario
Precondiciones	Ninguna
Flujo de Eventos	El usuario selecciona dentro del menú la opción Departamentos Se le muestra al usuario todos los departamentos ordenados alfabéticamente.
Post-condiciones	Ninguna

Tabla 5.2: Descripción del Caso de Uso Explorar Departamentos

Caso de Uso	Explorar Epígrafes
Descripción	Permite al usuario navegar por epígrafes/especialidades de la aplicación
Autor	Usuario
Precondiciones	Ninguna
Flujo de Eventos	El usuario selecciona dentro del menú la opción Especialidades Se le muestra al usuario todos los epígrafes ordenados alfabéticamente.
Post-condiciones	Ninguna

Tabla 5.3: Descripción del Caso de Uso Explorar Epígrafes

Caso de Uso	Buscar mediante búsqueda avanzada
Descripción	Permite al usuario filtrar las oposiciones por una búsqueda avanzada
Autor	Usuario
Precondiciones	Ninguna
Flujo de Eventos	El usuario selecciona dentro del menú la opción Oposiciones
	El usuario rellena las opciones que desee filtrar dentro de las oposiciones.
	Se le muestra al usuario una lista con las oposiciones que cumplan los requisitos del usuario de más recientes a más antiguas.
Post-condiciones	Las oposiciones mostradas cumplen los filtros dispuestos por el usuario

Tabla 5.4: Descripción del Caso de Uso Buscar mediante búsqueda avanzada

Caso de Uso	Buscar por directorio
Descripción	Permite al usuario buscar las oposiciones por un sistema de directorios
Autor	Usuario
Precondiciones	El usuario debe estar en la vista de Departamentos o Epígrafes
Flujo de Eventos	El usuario selecciona un Departamento o un Epígrafe (dependiendo de la vista en la que esté)
	Se le muestra al usuario todos los elementos relacionados al dato seleccionado.
	El usuario selecciona un elemento relacionado.
	Se le muestra al usuario una lista con las oposiciones que cumplan el par Departamento-Epígrafe.
Post-condiciones	Las oposiciones mostradas cumplen el par Departamento-Epígrafe

Tabla 5.5: Descripción del Caso de Uso Buscar por directorio

Caso de Uso	Ver Oposición
Descripción	Permite al usuario acceder al documento original de la oposición
Autor	Usuario
Precondiciones	El usuario tiene que estar en una vista con una serie de oposiciones
Flujo de Eventos	El usuario selecciona una oposición
	Se le muestra en detalle la oposición seleccionada.
	El usuario selecciona el botón PDF.
	El sistema lleva al usuario a la página oficial de la oposición.
Post-condiciones	Ninguna

Tabla 5.6: Descripción del Caso de Uso Ver Oposición

5.2. Diagrama de clases

Este diagrama permite tener una estructura de las clases que hay en la aplicación. Están

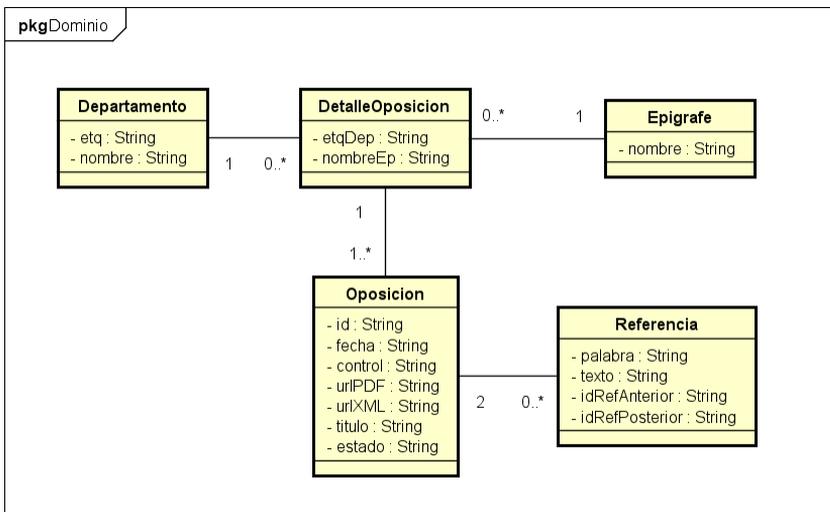


Figura 5.2: Diagrama de Clases

las 3 clases principales de la aplicación: Oposición, Epígrafe y Departamento. Hay dos clases soporte: DetalleOposición que permite a la aplicación navegar entre los departamentos y los epígrafes para acceder a las oposiciones, y Referencia cuya finalidad es tener un historial de los cambios que tienen las oposiciones.

5.3. Recursos REST

Este diagrama ilustra los servicios REST que tiene la aplicación. La información de las clases hace referencia al modelo de dominio que se presentaba en la anterior sección.

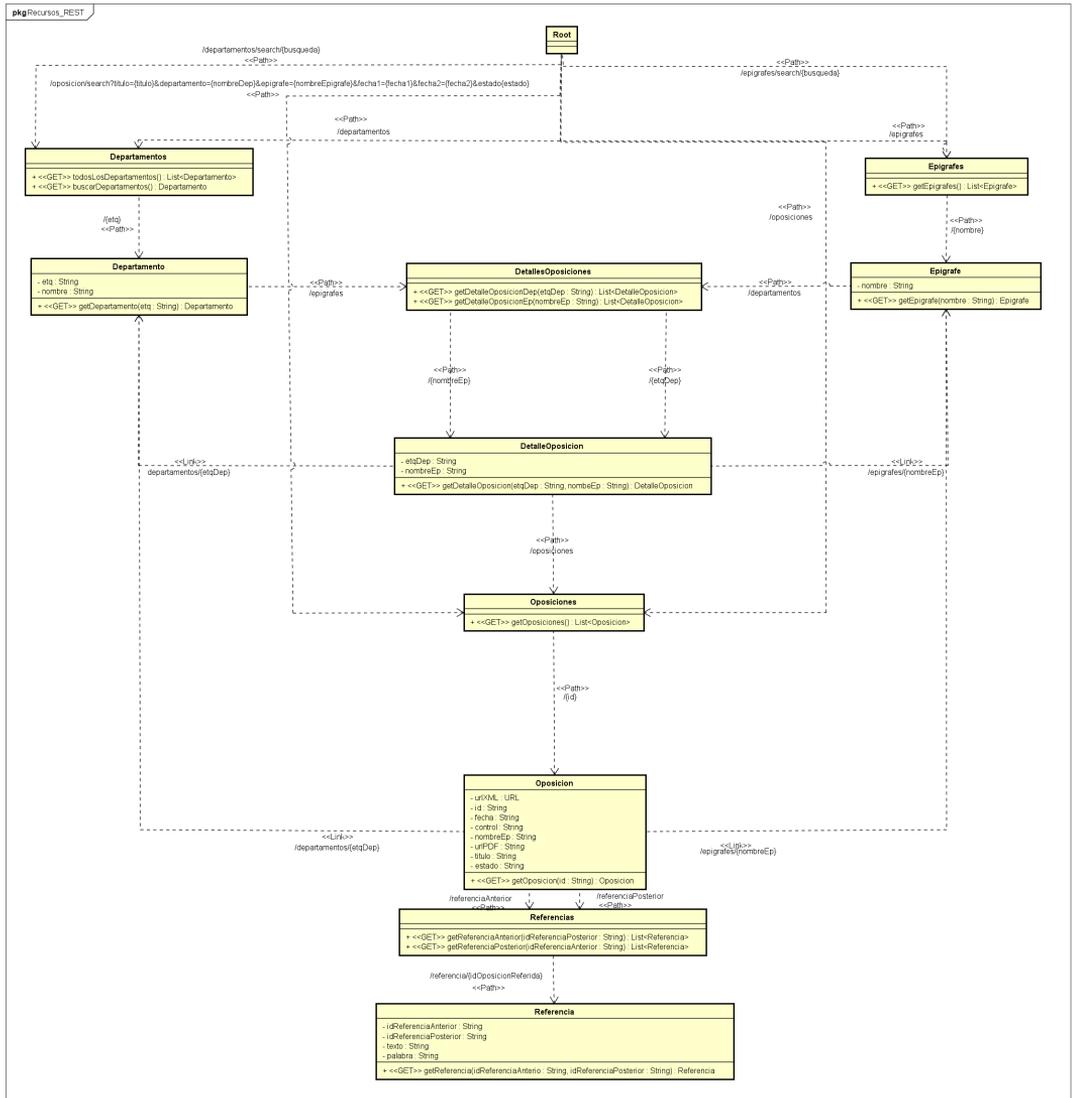


Figura 5.3: Diagrama de Recursos REST

5.4. Despliegue

Para tener una idea visual de cómo está hecho el despliegue del servicio REST, se ha creado este diagrama. Aquí reside la información de qué herramienta se va a utilizar para desplegar la aplicación, así como los entornos donde residen los servicios REST. Las herramientas utilizadas se han explicado en capítulos anteriores.

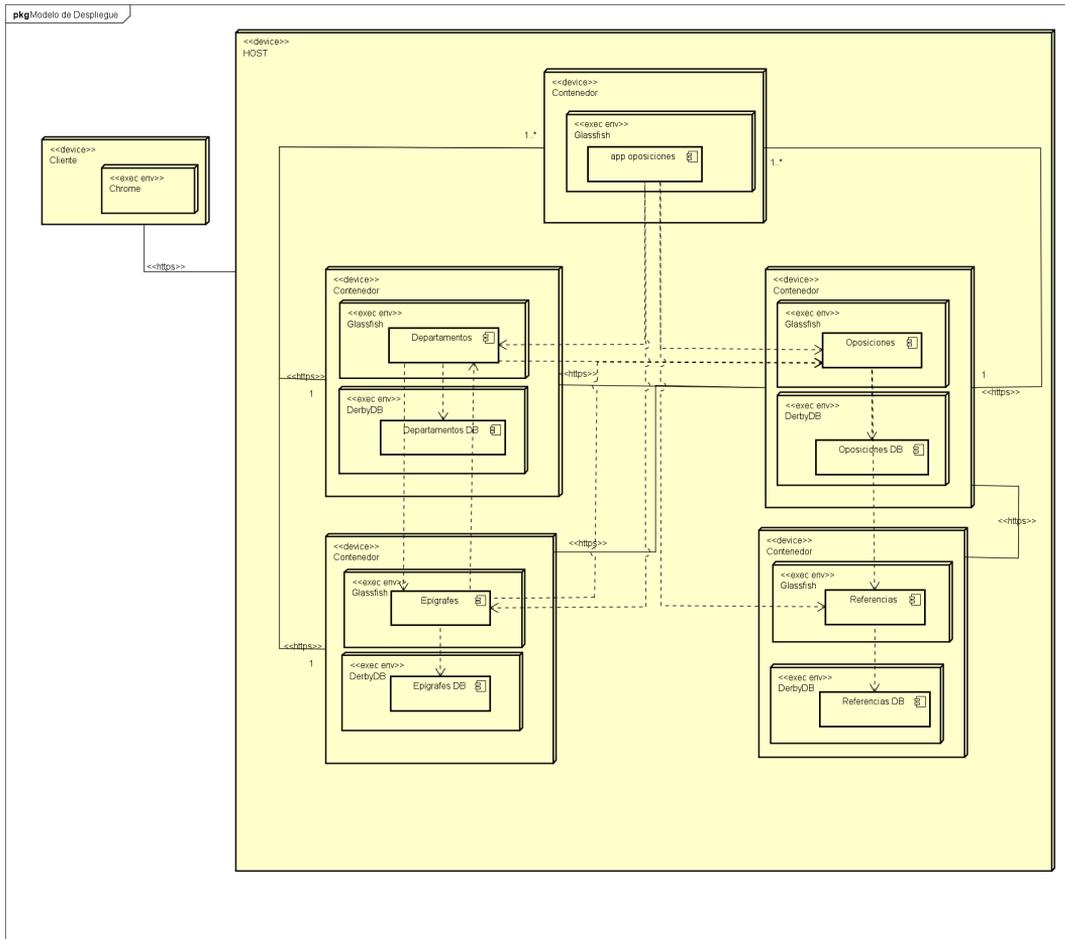


Figura 5.4: Diagrama de Despliegue

5.5. Diseño de la Página

El diseño de la página es una parte fundamental de la aplicación. Una mala interfaz puede echar todo el proyecto por tierra. Hay muchos ejemplos de proyectos que tienen buena

funcionalidad, pero tienen una mala interfaz que provoca que los usuarios no usen esa aplicación. También se da el caso contrario, un proyecto con una funcionalidad mala con una buena interfaz que "encandila" a los usuarios y estos se decantan por esta aplicación.

Un aspecto importante dentro del diseño de la página es la gama cromática. La idea de la página es que tenga una paleta de colores definida, evitando sobrecargar la aplicación con colores que no combinen. Finalmente, el color elegido es un verde pastel que recuerde al color de las típicas mesas y sillas que hay en la mayoría de los institutos de España, y que son usados con normalidad en los exámenes presentes en las oposiciones.

Recapitulando de capítulos anteriores, se ha dicho que la aplicación necesita:

- Una ventana de inicio, de donde puedan partir los usuarios y que contenga una lista con las oposiciones de la semana, así como una explicación de los tipos de datos que se van a presentar.
- Una vista en la que se puedan disponer la lista de las oposiciones junto a un sistema de búsqueda complejo. Este sistema de búsqueda contendrá un tipo de entrada para cada campo, permitiendo así tener un tipo de entrada personalizado si el campo se trata de una fecha o de seleccionar un tipo de estado.x z
- Si se había dicho que se necesitaba una lista con las oposiciones, también se necesitará una vista que disponga una lista con todos los departamentos y otra vista con los epígrafes. Ambas vistas tendrán en común un simple sistema de búsqueda que permita filtrarlas por el nombre.
- Y una última ventana donde se presente en detalle la oposición, junto al hilo de cambios que ha tenido la misma, y un botón que permita al usuario ver el documento desde la página oficial. El contenido de la información de cada documento será distinto dependiendo de si es una oposición o si es un documento de corrección.

5.5.1. Ventana de Inicio

Esta es la primera vista que tendrán los usuarios al entrar a la página. Esta vista tiene bastantes cosas en común con las demás.

Lo primero es la cabecera. Esta aparecerá en todas las demás vistas y contendrá los cuatro enlaces que permitan al usuario moverse por la aplicación, simplemente seleccionando la vista a la que quiera acceder. Esta funcionalidad es indispensable en la aplicación, ya que el usuario querrá moverse por la página sin tener que introducir una nueva dirección.

Otro aspecto importante es la presentación de las vistas que aparece justo al principio. Esta explicación contendrá una imagen que sea fácil de asociar al dato que esté asociado, ya que para una persona es más fácil asociar un término a una imagen que a una simple palabra. Una imagen vale más que mil palabras

5.5. DISEÑO DE LA PÁGINA

Adicionalmente, habrá una breve explicación de la vista a la que esté asociado el dato. Si la presentación está basada en la vista de las oposiciones, se explicará lo que es una oposición, lo que tendrá la página y cómo puede interactuar el usuario con esta página.

Si solo apareciese la presentación, la vista quedaría muy vacía. Por eso, es interesante añadir la posibilidad al usuario de ver las oposiciones más novedosas de la semana sin tener que indagar más en la página, simplemente interactuando con la página principal.

Para ver el boceto del dispositivo móvil vea la figura 5.5. Para ver el boceto del escritorio vea la figura 5.6. Para ver el menú vea la figura 5.7.



Figura 5.5: Boceto de la ventana de inicio del dispositivo móvil

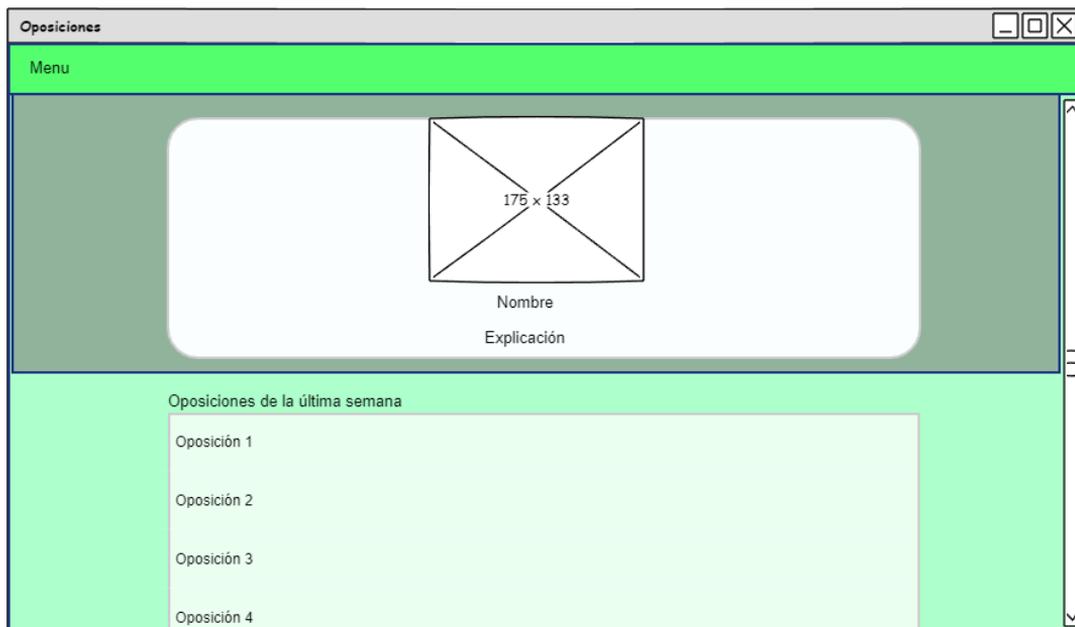


Figura 5.6: Boceto de la ventana de inicio del escritorio

5.5.2. Lista de Oposiciones

Esta es la vista con la principal funcionalidad de la aplicación. El principal objetivo del proyecto es permitir al usuario visualizar las oposiciones que desee, según varios criterios. La elección de estos criterios ya se ha explicado en capítulos anteriores, y ahora queda desarrollar la forma en la que el usuario pueda introducir estos criterios a la hora de realizar una búsqueda.

La primera entrada corresponde al campo del título de la oposición. Se le distingue entre los otros campos, debido a que la información que hay en el título abarca contiene información de los epígrafes, departamentos y de la misma fecha de la oposición. Esto permite al usuario filtrar con gran detalle introduciendo solo información en un solo campo.

Los dos campos siguientes corresponden al Departamento y a la Especialidad de la oposición. Se ha cambiado el término del Epígrafe por Especialidad en la presentación del usuario, ya que se puede encontrar al Epígrafe como un término demasiado poco explicativo para el usuario, aunque la propia fuente oficial se refiera a este campo por Epígrafe.

Los campos "Desde" y "Hasta" permiten al usuario introducir un rango de fechas para poder filtrar oposiciones que estén comprendidas en ese rango. Si el usuario solo introduce el primer campo "Desde", se entenderá que solo quiere ver las oposiciones con la misma fecha que ese campo. Y si el usuario solo introduce el segundo campo "Hasta", la aplicación entenderá que quiere ver las oposiciones se hayan dispuesto, como máximo, hasta esa fecha.

Para poder permitir al usuario filtrar las oposiciones según el estado en el que esté, se ha decidido por desplegable que de la posibilidad al usuario de seleccionar el tipo de estado en el que se puede encontrar la oposición, sin que este tenga que saberlo de memoria.

Finalmente, aparece un botón que servirá para realizar la búsqueda, redirigiendo al usuario a la nueva ventana con las oposiciones que desee.

Las oposiciones aparecerán ordenadas según la fecha de disposición, apareciendo como primera la más nueva. La información que aparecerá en esta ventana, será par departamento-epígrafe, así como la fecha de disposición y el estado de esta.

La idea de la lista de las oposiciones es que se vaya cargando a medida que el usuario navegue por la vista. Así, evitamos que la página espere para recibir todas las oposiciones, las cuales son muy numerosas, lo cual sería un proceso extremadamente lento.

Para ver el boceto del dispositivo móvil 5.8. Para ver el boceto del escritorio vea la figura 5.9.

5.5.3. Lista de Departamentos/Epígrafes

Al haber una gran cantidad de departamentos y de epígrafes (especialidades), es imperativo ofrecer al usuario una forma de ver qué tipos de departamentos y de epígrafes hay. Para ayudar al usuario, aparecerá un sistema de búsqueda, bastante simple, con el que pueda filtrar y seleccionar el dato que quiera. Las vistas de los departamentos y los epígrafes son similares, lo único que se diferencian es el título que aparece debajo del buscador.

También, esta vista se asemeja a la disposición de los datos que referencian los departamentos y epígrafes (los departamentos que están relacionados con un cierto epígrafe, o los epígrafes que se relacionan con un cierto departamento). Igualmente, para esta vista se cambiará de título, para mostrar al usuario que está en una especie de detalle del departamento/epígrafe seleccionado (Departamento/Nombre del departamento/Epígrafes o Epígrafes/Nombre del epígrafe/Departamento).

Para ver el boceto del dispositivo móvil 5.10. Para ver el boceto del escritorio vea la figura 5.11.

5.5.4. Detalle de la Oposición

Para facilitar al usuario, se ha elegido que esta sea una ventana emergente. Esta decisión de diseño facilita al usuario la posibilidad de ir de oposición en oposición, sin tener que moverse de la ventana.

Dentro del diseño, la ventana tendrá información de la oposición, así como un hilo de los cambios que ha tenido, o que esta haciendo ese documento. El diseño de esta información pretende asemejarse al formato que Twitter que con la creación de hilos, permitiendo ver el

histórico de cambios, con tan solo moverse por la pantalla, sin tener que navegar entre otras vistas.

Sobre la información del documento seleccionado, aparecerá el título, la fecha, el par Departamento-Epígrafe y el identificador. Dentro de las referencias, aparecerá también el par Depatamento-Epígrafe y el "título" de la referencia (en realidad es el conjunto del atributo palabra y texto que aparece en el documento oficial).

Debido a la decisión de diseño de tener una base de datos ligera, con poca información. Es indispensable ofrecer al usuario la posibilidad de acceder al documento con la información completa. En la ventana, aparecerá un botón con un Icono que se asocie fácilmente a un documento, y este te llevará a la dirección del documento oficial en formato PDF.

Para ver el boceto del dispositivo móvil 5.12. Para ver el boceto del escritorio vea la figura 5.13.

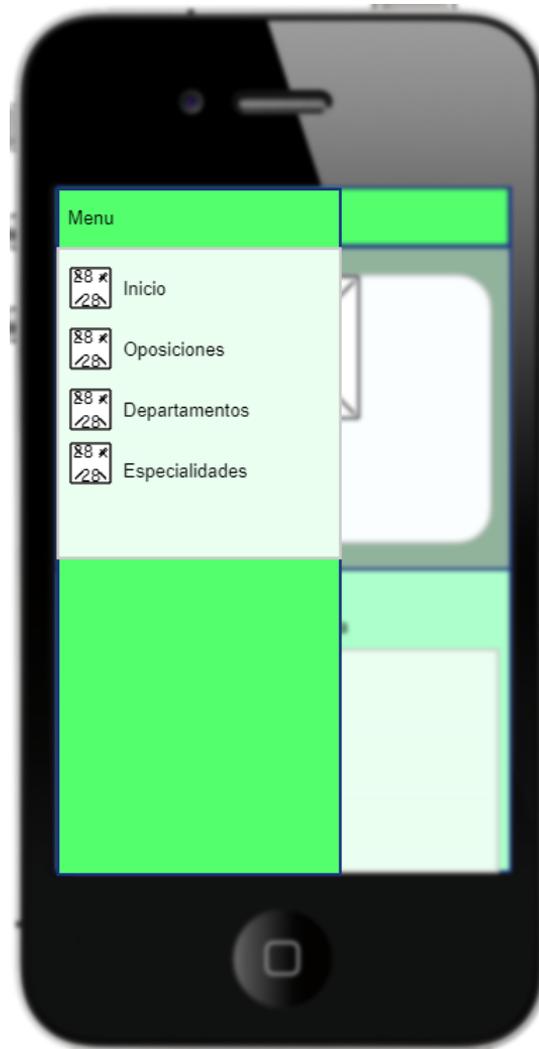


Figura 5.7: Boceto de la ventana de inicio con el menú activo del dispositivo móvil

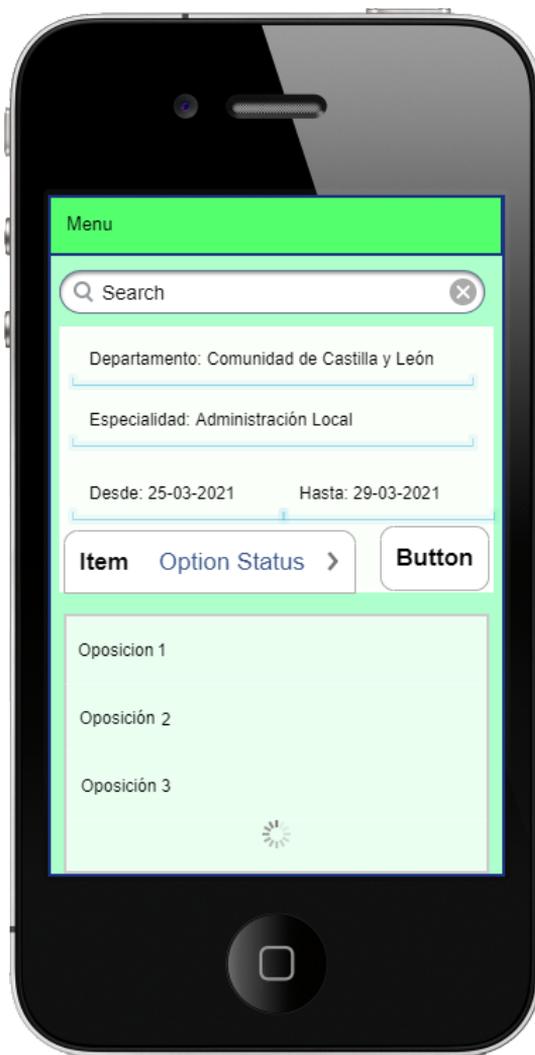


Figura 5.8: Boceto de la ventana con la lista de oposiciones del dispositivo móvil

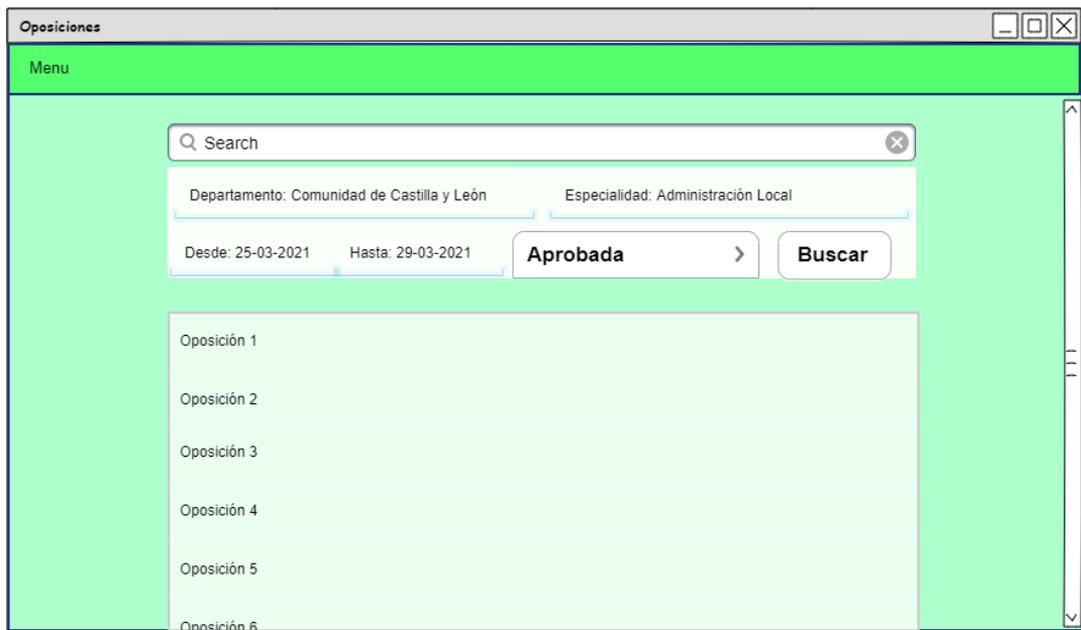


Figura 5.9: Boceto de la ventana con la lista de oposiciones del escritorio

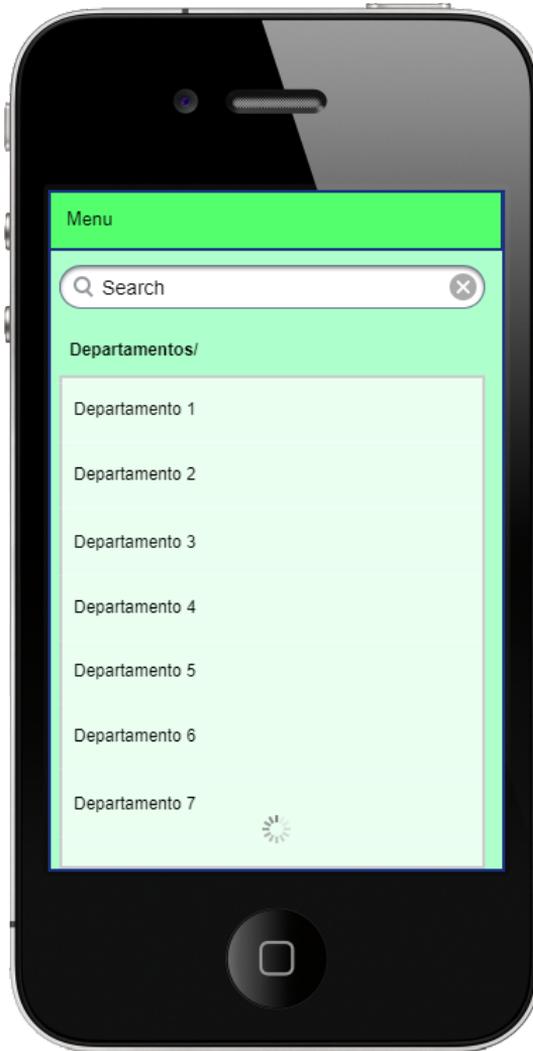


Figura 5.10: Boceto de la ventana con la lista de departamentos del dispositivo móvil

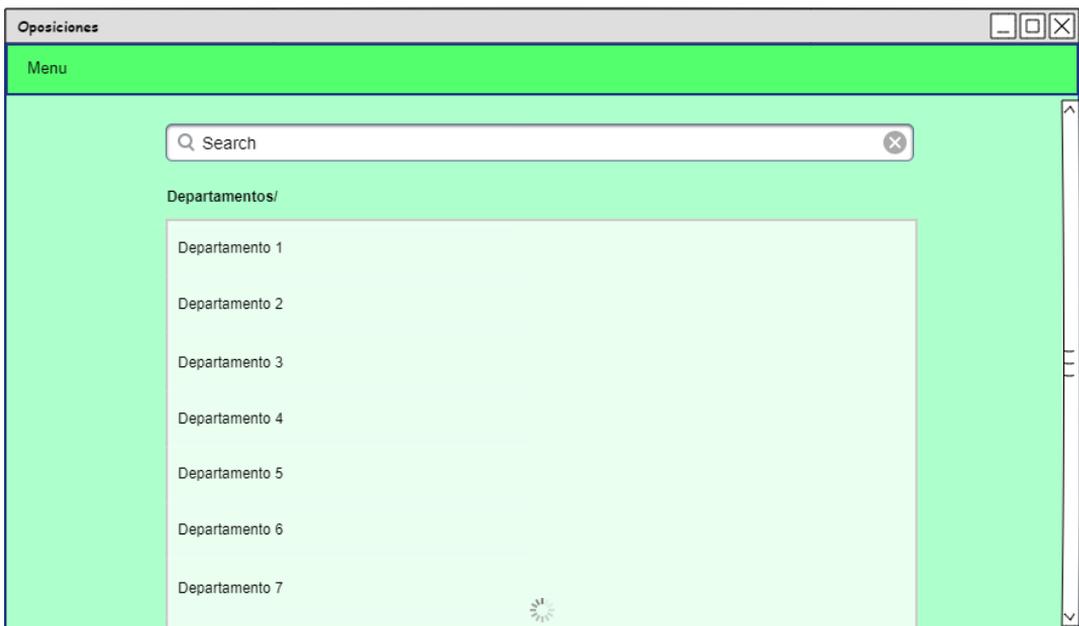


Figura 5.11: Boceto de la ventana con la lista de departamentos del escritorio



Figura 5.12: Boceto de la ventana con el detalle de la oposición del dispositivo móvil

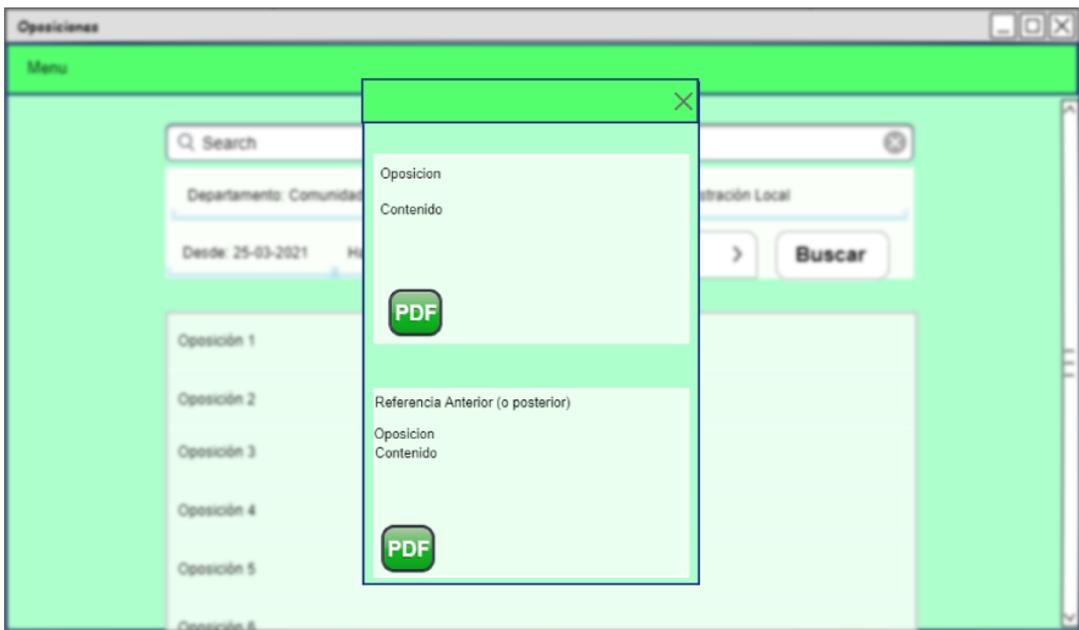


Figura 5.13: Boceto de la ventana con el detalle de la oposición del escritorio

Capítulo 6

Implementación y Pruebas

6.1. Backend

La aplicación se divide entre la Capa de Presentación (Frontend) y Capa de Acceso a Datos (Backend). Cada capa está implementada de forma distinta. La parte Backend conforma la parte de la aplicación donde residen los datos y donde accede el Frontend para acceder a ellos.

Dentro de esta parte, podemos distinguir dos procesos necesarios dentro del Backend. El primero es el Script de recopilación de datos y el segundo es el Servicio REST.

6.1.1. Script de Recopilación de Datos

Este Script permite a la aplicación tener una base de datos actualizada. Este accede al Boletín Oficial del Estado diariamente en busca de oposiciones o documentos que realicen cambios.

El Script se ha hecho utilizando el Entorno de Desarrollo Integrado (IDE) de NetBeans. El lenguaje en el que está basado el programa es Java. Este lenguaje tiene diversas bibliotecas que nos permite comunicarnos con la base de datos y acceder a documentos de internet.

El funcionamiento es simple: El Script en estado normal está esperando hasta que se publique el BOE de ese día. Normalmente, la fecha de publicación ronda entre las 9 a.m. y las 10 a.m.. Otro detalle referente a la publicación del BOE es que no publica los domingos. Con todos estos datos, el Script se lanza las 11 a.m. de lunes a sábado.

Cuando el Script accede al boletín, este busca los documentos referentes a las oposiciones en la sección 2.B Oposiciones y Concursos. En este documento recopila los datos referentes a los Departamentos y Epígrafes (posteriormente este término pasará a llamarse Especialidades). Por cada documento referente a una oposición, hay un acceso al documento detallado

de este. Aquí, el Script recopila el resto de datos importantes que tiene la oposición y comprueba si el documento es una corrección o una oposición normal. Si el documento es una corrección, se comprobará si la corrección tiene un carácter modificador del estado de la oposición a la que se refiere, es decir, si el documento aprueba o cancela una oposición. Si es así, la oposición a la que se refiere el documento cambiará de estado dependiendo del carácter del documento modificador.

Si se ejecuta el Script sin argumentos, se recopilarán las oposiciones del día actual. Esto está hecho así para que un programador de tareas (como el de Windows) pueda ejecutar la aplicación diariamente. Se puede añadir como argumento una fecha, del formato DD MM YYYY, haciendo que se recopilen las oposiciones desde ese día hasta el actual. Este argumento es bastante útil si la base de datos está vacía y se busca rellenarla de datos.

Ya habiendo rellenado la base de datos con los Departamentos, Especialidades, Oposiciones y demás clases soporte, todavía nos queda ver cómo se puede acceder a estos datos.

6.1.2. Servicio REST

La API REST se ha construido con el lenguaje Java utilizando NetBeans como entorno de desarrollo. Como ya se ha comentado, este servicio implementa un acceso definido a los datos.

La aplicación NetBeans cuenta con un sistema automatizado de creación de modelo usando la base de datos de referencia. Esto, permite tener una representación exacta del modelo dentro de la API.

Adicionalmente, con NetBeans se puede generar automáticamente un sistema de Persistencia JPA (Java Persistence API) que nos permite tener una serie de clases que implementan la funcionalidad de acceso a la base de datos.

En el capítulo referente al diseño, se explica en más detalle cómo está organizado este servicio.

La funcionalidad se reparte en 3 recursos: Recurso Oposiciones, Recurso Departamentos y Recurso Epígrafe/Especialidad. Las 3 permite acceder a un único recurso del mismo tipo y al conjunto de ellos.

En el Recurso Oposiciones está la funcionalidad referente a las oposiciones y a las referencias de estas. Mediante el identificador, se puede acceder a las referencias anteriores y a las referencias posteriores, permitiendo al usuario ver el historial de cambios que ha habido en un mismo documento.

La funcionalidad para ver los departamentos que tiene una especialidad y viceversa, la tienen el Recurso Departamento y el Recurso Especialidad/Epígrafe. Estos recursos tienen la funcionalidad básica de acceso a los datos mediante el identificador y el acceso al conjunto de datos, y a mayores tienen esta funcionalidad.

6.2. Frontend

Dentro del Frontend, se encuentra la capa de presentación, es decir, donde el usuario va a interactuar con la aplicación. El proceso de creación del proyecto fue automático, ya que la misma aplicación Visual Studio Code implementa un sistema de generación automática de proyectos, incluyendo soporte para el Framework de Ionic.

El servidor donde se despliega la aplicación web lo implementa la misma aplicación de Visual Studio Code. La generación de vistas y de componentes también la hace automáticamente la aplicación usando los comandos correspondientes.

En total, la aplicación cuenta con cuatro páginas: Página Inicial, Lista de Oposiciones, Lista de Departamentos o Especialidades, Detalle de los Departamentos y Especialidades, y Detalle de la Oposición. Todas estas páginas contienen la cabecera de la página que contiene el menú que permite al usuario navegar entre todas estas vistas.

El componente Detalle de la Oposición es una ventana emergente y está disponible dentro de la página principal y en el listado de las oposiciones.

Las páginas se han basado en los bocetos que se hicieron en la etapa de Diseño. Durante la implementación, hubieron algunos cambios, como la decisión de poner un fondo de pantalla en vez de el color plano que había antes.

6.3. Pruebas

6.3.1. Pruebas Automatizadas Servicio REST

El proceso de prueba del servicio REST se realizó utilizando la herramienta Postman. Esta herramienta ya ha sido introducida en capítulos anteriores. Con Postman se puede crear una serie de peticiones automatizadas con las que probar los servicios REST.

Para realizar las pruebas es necesario tener una serie de datos en la aplicación. Debido a que los datos dependen del boletín y que estos pueden verse modificados a lo largo del tiempo, es necesario crear una serie de datos que imiten el proceso real.

Una vez con los datos dispuestos, se creó una serie de casos de uso en los que se tendría que probar la aplicación.

6.3. PRUEBAS

Nombre de la prueba	Descripción	Estado	Salida
Get Oposición	Petición de una oposición existente en la aplicación	Correcto	Objeto de la oposición pedida
Get Oposición Inexistente	Petición de una oposición inexistente en la aplicación	Fallo	Resultado vacío
Get Epígrafe	Petición de un epígrafe existente en la aplicación	Correcto	Objeto del epígrafe pedido
Get Epígrafe Vacío	Petición de un epígrafe inexistente en la aplicación	Fallo	Resultado vacío
Get Departamento	Petición de un departamento existente en la aplicación	Aceptado	Objeto del departamento pedido
Get Departamento Inexistente	Petición de un departamento inexistente en la aplicación	Fallo	Resultado vacío
Get Epígrafes Departamento	Petición para ver los epígrafes que tiene asociado un departamento	Aceptado	Lista de epígrafes
Get Departamentos Epígrafes	Petición para ver los departamentos que tiene asociado un departamento	Aceptado	Lista de departamentos
Get Oposiciones Departamento Epígrafe	Petición para ver las oposiciones con el mismo par Departamento-Epígrafe	Aceptado	Lista de oposiciones
Get Oposición Búsqueda	Petición de búsqueda completa en la que se rellenan todas las opciones	Aceptado	Lista de oposiciones
Get Oposición Búsqueda Rango Fechas	Petición de búsqueda en la que se rellenan los campos de fecha	Aceptado	Lista de oposiciones

Tabla 6.1: Tabla con las pruebas de caja negra del servicio REST

A continuación, se muestra una imagen de los resultados dentro de la aplicación Postman:

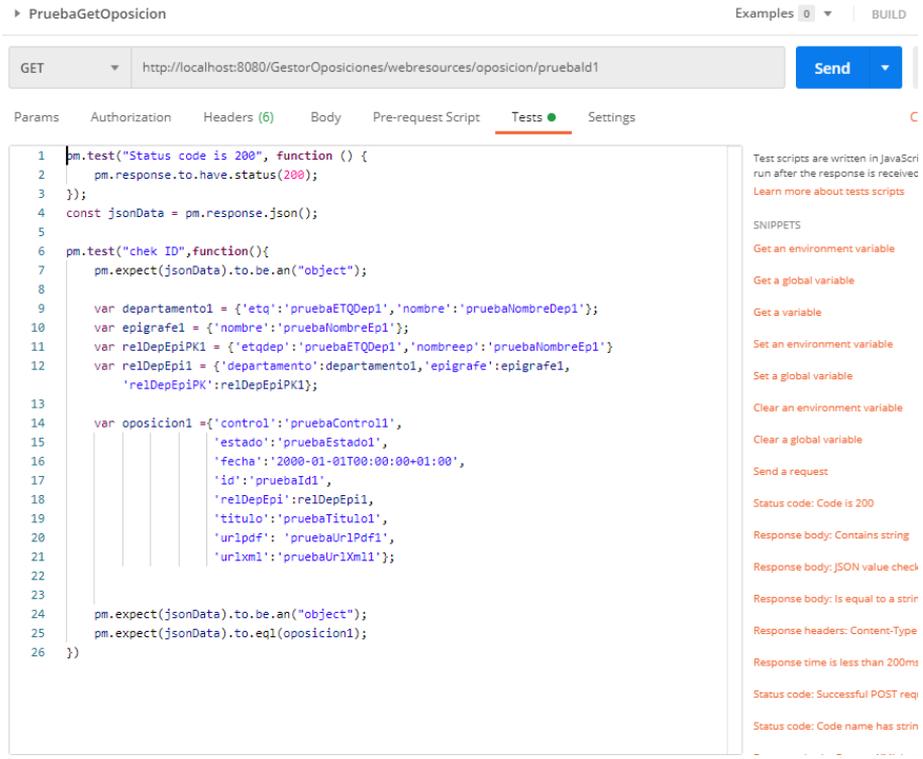


Figura 6.1: Ejemplo de una prueba en la aplicación Postman

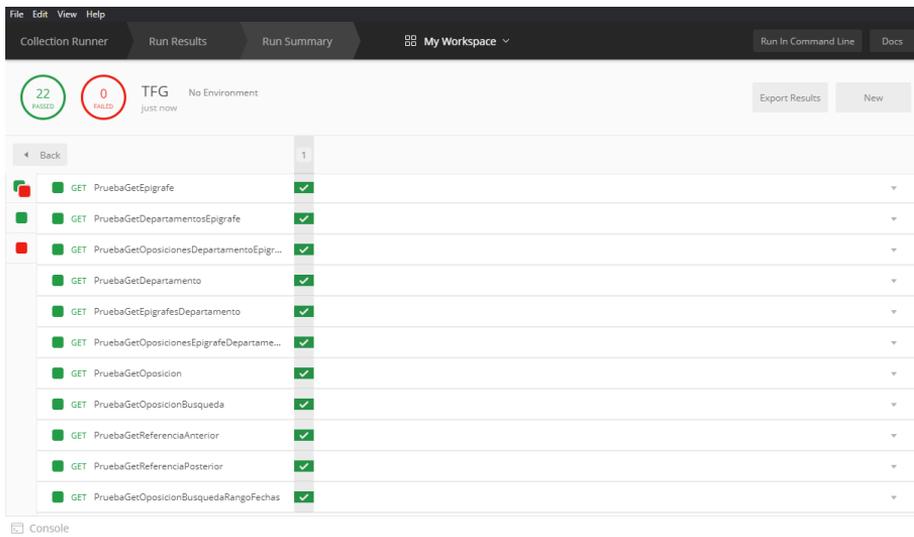


Figura 6.2: Captura de la aplicación Postman con las pruebas automatizadas

6.3.2. Pruebas E2E

Habiendo probado el funcionamiento de los servicios REST, ya tenemos asegurado el acceso a los datos. Debido a diversos imprevistos, se ha visto necesario sustituir los métodos de testeo automáticos por un modo manual. A continuación, se ha descrito el procedimiento que se ha seguido para probar la aplicación.

En la vista de la presentación, solo hay dos elementos que se puedan testear: uno es las redirecciones del menú y de la presentación, y el otro es el listado de oposiciones de la semana. Se probó que los elementos de la interfaz que sirvan para permitir al usuario moverse por las vistas funcione y redirija correctamente a la vista deseada. Para las oposiciones de la semana, se comprobó que el listado no sale de los límites de la fecha, y que al seleccionar una oposición, se abría la ventana emergente con su información detallada.

En la vista del listado de oposiciones, se comprobó que la disposición de las oposiciones aparecen primero las más recientes. Dentro de estas oposiciones se comprobó que al acceder a una, se abría la ventana emergente con su información detallada. Esta vista tiene el sistema de búsqueda avanzado, para esto se realizó una serie de peticiones comprobando que cada entrada de búsqueda funciona correctamente y en combinación con las demás. Con esto, comprobamos correctamente el sistema de búsqueda avanzado, ya que podía haber algún requisito en algún campo que no se hubiese detectado. Para comprobar que al acceder al



The image shows a search interface for 'Oposiciones'. At the top, the word 'Oposiciones' is displayed in a large, brown font. Below this is a search bar with a back arrow on the left and the text 'Buscar por título'. Underneath the search bar are two input fields: 'Departamento' and 'Especialidad personal de'. Below these fields are two date fields: 'Desde: 2021-06-21' and 'Hasta: 2021-05-21'. To the right of the date fields is a dropdown menu labeled 'Corrección' with a downward arrow. At the bottom right of the search area is a green button with the text 'BUSCAR' in white capital letters.

Figura 6.3: Ejemplo de búsqueda con fechas, estado y especialidad rellena

detalle de la oposición se recogen las referencias anteriores y posteriores, se hizo una búsqueda para obtener las oposiciones que han sufrido algún cambio, es decir, las oposiciones con el estado Aprobado o Cancelado. Aquí, se comprobó que al acceder a su detalle, aparecían los documentos que realizaban dichos cambios. Para comprobar las referencias anteriores, se siguió el mismo método pero al revés, buscando las oposiciones que tengan el estado de "Corrección", ya que estas al hacer cambios en oposiciones anteriores, se asegura tener mínimo una referencia anterior.

Dentro del listado de Especialidades y Departamentos, se comprobó se listarán correctamente por orden alfabético que el sistema de búsqueda funcionaba correctamente. Para comprobar el funcionamiento de la búsqueda por directorio, se seleccionó el mismo par

Departamento-Especialidad, ya que las oposiciones resultantes serían las mismas.

Capítulo 7

Seguimiento

Durante finales del mes de enero y de febrero comenzaron las primeras reuniones explicativas del proyecto. Se realizó una reunión que sirvió como Sprint Planning, donde se introdujeron los objetivos del proyecto, así como los requisitos que se querían abarcar. Esta reunión dio comienzo al proyecto en sí.

7.1. Sprint 1 (31/01/2021 - 7/02/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Estudiar la disposición de los datos abiertos en el BOE	Completado	12 horas	10 horas
Estudiar la forma en la que se va a abordar el proyecto	Completado	10 horas	8 horas
Hacer documentación para la memoria	En proceso	4 horas	7 horas
Total	3/2 tareas	25 horas	26 horas

Tabla 7.1: Sprint 1

Este sprint tenía como objetivo estudiar la forma en la que se presentan los datos en el BOE. Esta parte es fundamental, ya que todo el proyecto depende de ello. Si no existe una forma en la que se puedan acceder a los datos, no tiene sentido hacer un gestor de oposiciones.

Se encontró la forma de acceder a estos datos en la página oficial del BOE. Este se presenta a diario en formato XML, y en la sección 2B se encuentran alojados las oposiciones.

Dentro de las oposiciones, se vio un gran problema: los datos que nos interesaban no estaban puestos en ningún formato, ya que la mayoría de datos importantes aparecen dentro

del apartado texto. Este apartado no había signos de estar regularizado, ya que cada oposición presentaba los datos de una manera totalmente distinta

Se buscó algún indicio de que hubiese algún patrón. Al principio, se buscó dentro de los boletines oficiales de cada Comunidad Autónoma. Alguna Comunidad presentaba parecidos entre el formato de las oposiciones, aunque no tenían mucha consistencia.

Debido a estos contratiempos, se intentó usar los datos que sí cumplían un formato y tenían un patrón conjunto. Estos eran los Departamentos y los Epígrafes (posteriormente se pasó a llamar Especialidad), así como el identificador de la oposición y su fecha de disposición.

Otro objetivo era ver la manera en la que iba a construirse la aplicación. La primera idea fue realizar una aplicación web, así como una aplicación móvil. Esta idea tenía el contratiempo del tiempo, ya que para un proyecto así de largo el tiempo que había se quedaba corto.

La siguiente idea fue el desarrollo de una aplicación web PWA. Este concepto ha sido introducido en anteriores capítulos, pero para resumir es una aplicación con funcionalidades de escritorio y de dispositivo móvil. Esta idea daba soporte al objetivo de tener una aplicación multiplataforma, sin el problema del tiempo.

La siguiente idea era cómo iban a disponerse los datos y con qué herramientas. Esta parte fue fácil, ya que se impuso la primera idea que surgió. Esta fue la de dividir la aplicación en Frontend y Backend, donde en el Backend se implementaría un servicio REST y donde estaría la base de datos, y la parte Frontend sería donde los usuarios pudiesen interactuar.

En la elección de las herramientas, se eligieron las que más había usado el alumno durante la universidad, ya que evitaríamos tiempos de aprendizaje innecesarios. Para la parte Backend se eligió a NetBeans como IDE, Glassfish como servidor y Derby como base de datos relacional. En la parte Frontend se eligió a Visual Studio Code como IDE y a Angular e Ionic como Frameworks de apoyo.

7.2. Sprint 2 (08/02/2021 - 15/02/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Configurar aplicación NetBeans para el proyecto	Completado	4 horas	4 horas
Recopilación de datos de forma automática	Completado	16 horas	14 horas
Estudiar patrones	Completado	8 horas	8 horas
Hacer documentación para la memoria	En proceso	2 horas	3 horas
Total	Completado	28 horas	28 horas

Tabla 7.2: Sprint 2

El objetivo del sprint era probar la posibilidad de la recopilación de datos de forma automática. Para ello lo primero que se hizo fue configurar la aplicación destinada al Backend para poder implementar el código allí.

Se descargaron las versiones correspondientes del IDE, así como las bibliotecas y pluggins destinadas a la base de datos.

Con las herramientas dipuestas, empezó la imlementación del Script que recopile las oposiciones diariamente.

La primera versión del Script recopilaba las oposiciones entre un rango de fechas determinado. Esta decisión fue hecha para probar la capacidad de la base de datos.

La idea del Script era que accediera al BOE del día y que allí inspeccionara en busca de oposiciones. Cuando encontraba una oposición, accedía a la página de esta y recopilaba los datos requeridos.

La primera prueba hacía este programa fue la de buscar todas las oposiciones entre la fecha del 05-02-1999 hasta 01-01-2019. El número de oposiciones que se guardaron fueron de un total de 67111 oposiciones, junto con 99 departamentos distintos y 502 epígrafes, y el proceso duró unos 20 minutos.

Gracias a estos datos se vio la capacidad que tenían los epígrafes de filtrar las oposiciones, al asemejar su significado a las especialidades que los usuarios estarían buscando. También se vio que el programa era rápido, teniendo en cuenta que por cada oposición se accedía a 2 direcciones distintas.

7.3. Sprint 3 (15/02/2021 - 23/03/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Realizar el diseño de la aplicación	Completado	18 horas	20 horas
Adaptar cambios al Script	Completado	10 horas	10 horas
Hacer documentación para la memoria	Completado	2 horas	2 horas
Total	Completado	30 horas	32 horas

Tabla 7.3: Sprint 3

El objetivo del sprint era ver qué datos iban a recogerse finalmente del boletín y pasarlos a formato UML donde podían ser estudiados.

Durante el estudio de estos datos, se vio la posibilidad de ofrecer al usuario los epígrafes que ha creado históricamente un departamento y viceversa. Esta idea viene junto a la necesidad del posible usuario de ver las comunidades que pedían un epígrafe (especialidad)

determinado. También, el caso contrario en el que el usuario estaba interesado en las especialidades que se podían pedir en su comunidad autónoma. Al estar las comunidades dentro de los departamentos que disponían diariamente las oposiciones, esta idea al final se incluyó.

Para esto se modificaron los diagramas y se estudió la mejor manera para hacerlo posible. Esta manera también hizo que se modificara el Script de recogida de datos. Este Script al principio estaba concebido como una forma de estudio, pero a medida que se iban añadiendo cosas se incluyó en el formato final.

7.4. Sprint 4 (23/02/2021 - 02/03/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Hacer primera versión del Servicio REST	Completado	22 horas	20 horas
Realizar tests del Servicio REST	Completado	8 horas	7 horas
Hacer documentación para la memoria	En proceso	4 horas	3 horas
Total	Completado	34 horas	30 horas

Tabla 7.4: Sprint 4

Este sprint se caracterizó por el desarrollo del Servicio REST. El objetivo de este sprint era conocer la factibilidad del servicio usando las herramientas disponibles, así como hacer los servicios más generales y que se iban a usar sí o sí.

Con estos servicios podías acceder a todas las oposiciones, epígrafes y departamentos que había en la base de datos. También podías acceder a un dato en concreto, solo introduciendo su identificador. Y sobretodo, podías acceder a los epígrafes de cada departamento y viceversa, implementando la idea que se barajó en el anterior split.

Durante el proceso de testeo, se creó una serie de datos falsos que servían de apoyo en el testeo. Todavía no se implementaron los mecanismos automáticos de testeo, pero la creación de los datos fue útil para los posteriores splits, ya que era fácil saber si la ejecución había salido bien.

7.5. Sprint 5 (04/03/2021 - 14/03/2021)

Como en el caso de la parte Backend, ha sido necesario hacer una instalación limpia de las herramientas. Esta parte fue sencilla ya que todas el software era compatible entre sí.

Al haber tenido el primer contacto con el Framework de Ionic en este proyecto, ha sido necesario reservar un tiempo de aprendizaje con esta herramienta.

Tarea	Estado	Tiempo estimado	Tiempo invertido
Aprendizaje del Framework de Ionic	Completado	14 horas	10 horas
Instalación de Visual Studio Code, así como de los demás Frameworks	Completado	4 horas	4 horas
Hacer documentación para la memoria	Completado	4 horas	4 horas
Total	Completado	22 horas	22 horas

Tabla 7.5: Sprint 5

Para ellos se siguieron los tutoriales existentes dentro de la página oficial. Estos abarcan la parte de diseño de las interfaces de dispositivos móviles. Durante estos tutoriales se intentó implementar también soporte para vistas de escritorio, ya que estas iban a ser las funcionalidades que luego iban a estar implementadas en la aplicación final.

7.6. Época de prácticas (14/03/2021 - 10/04/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Seguir con el aprendizaje de Ionic	Completado	8	9
Seguir estudiando los datos de las oposiciones	Completado	4	3
Hacer documentación para la memoria	Completado	6 horas	8 horas
Total	Completado	10 horas	12 horas

Tabla 7.6: Época de prácticas

Durante este periodo tuvo lugar una serie de prácticas de otras asignaturas, así como las prácticas curriculares que hicieron no tener tiempo para el proyecto y se tuvieron que retrasar los sprints.

Aún así, se mantuvo cierto contacto con la aplicación para no perder los conocimientos y tener que volver a empezar desde 0 el aprendizaje.

Se aprovechó también a terminar la documentación acerca de las herramientas usadas hasta el momento, así como una explicación de la disposición de los datos dentro del BOE.

7.7. Sprint 6 (12/04/2021 - 25/04/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Implementar servicio REST en el Frontend	Completado	5 horas	5 horas
Creación de la primera versión de las vistas	Completado	10 horas	10 horas
Implementar funcionalidad de listar oposiciones	Completado	12 horas	10 horas
Hacer documentación para la memoria	Completado	5 horas	5 horas
Total	Completado	32 horas	30 horas

Tabla 7.7: Sprint 6

El objetivo del sprint era aplicar los conocimientos adquiridos sobre el Framework de Ionic y aplicarlos en la funcionalidad básica de la aplicación. Lo primero fue establecer las comunicaciones con el servidor rest, implementando el modelo dentro de la aplicación y las rutas hacia los servicios.

Se crearon las vistas y se empezó a llenar estas con lo básico. Lo primero fue la creación de la cabecera que iba a ser conjunta en todas las vistas. Esto permitía al usuario navegar entre todas las vistas sin tener que introducir ninguna dirección.

Lo siguiente fue ver si la aplicación era capaz de acceder a los datos del servicio REST. Al implementar esto, se vio la necesidad de cambiar un detalle del servicio REST para dar soporte a peticiones que vayan accediendo a los datos poco a poco.

La manera que tenía el servicio REST de proporcionar las oposiciones era o una solo o todas a la vez. Esto era inviable ya que en la base de datos podían existir decenas de miles de oposiciones, y esto perjudicaba con notoriedad a la velocidad de la aplicación. Para resolver esto se tuvo que implementar un servicio nuevo que permitiera añadir un campo el cual indicaba el número de página que quería acceder.

Se hicieron las pruebas pertinentes a este servicio y se continuó con el resto de la implementación.

Ahora teníamos una aplicación que mostraba las oposiciones que tenía la base de datos de poco a poco. Cuando ibas al final de la página, la aplicación solicitaba al servicio REST los datos siguientes, imitando el modo en el que las distintas páginas de internet mostraban los datos.

Tarea	Estado	Tiempo estimado	Tiempo invertido
Añadir la funcionalidad de listar los departamentos y especialidades para el usuario	Completado	10 horas	10 horas
Mejorar estética de la aplicación siguiendo los bocetos	Completado	6 horas	5 horas
Hacer documentación para la memoria	Completado	5 horas	5 horas
Total	Completado	21 horas	20 horas

Tabla 7.8: Sprint 7

7.8. Sprint 7 (26/04/2021 - 02/05/2021)

Este sprint se caracterizó por permitir al usuario ver los departamentos y especialidades existentes, y por establecer una primera versión de la estética de la página.

Para implementar la funcionalidad de los departamentos y especialidades, se basó en gran medida en lo hecho en el sprint pasado sobre las oposiciones. Teniendo este punto de partida se modificaron los servicios REST correspondientes y se adaptó el código de las oposiciones a los departamentos y a las especialidades.

La aplicación todavía no tenía ningún rasgo estético. Los datos estaban dispuestos en un fondo blanco y sin ningún embellecedor. Para esta parte, se tuvieron en cuenta los bocetos que se habían realizado al principio del proyecto. Sirviendo estos bocetos de guía, y gracias a los ficheros de configuración que ofrecía el Framework de Ionic, fue notoriamente sencillo cambiar la estética de la aplicación. Aunque, todavía quedaba mucho que hacer en este sentido. En este sprint no se añadieron a las vistas los métodos de búsqueda, ya que todavía no se habían creado los servicios REST correspondientes.

7.9. Sprint 8 (04/05/2021 - 12/05/2021)

El objetivo de este sprint era implementar el sistema de búsqueda avanzado en la aplicación de inicio a fin, así como el sistema de búsqueda por directorio.

Lo primero fue hacer el sistema de búsqueda por directorio, simplemente por facilidad, ya que el servicio REST ya estaba hecho. Para esto se creó otra vista que servía para los departamentos y para las especialidades.

Antes de realizar la implementación de la búsqueda avanzada, se estudió qué datos eran relevantes para filtrar las oposiciones. Esto implicó ciertas modificaciones en los bocetos y algún cambio en el modelo.

7.10. ÉPOCA DE PRÁCTICAS (17/05/2021 - 30/05/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Estudio de los datos de las oposiciones	Completado	4 horas	4 horas
Modificación de los bocetos y del modelo para implementar el sistema de búsqueda	Completado	8 horas	6 horas
Implementación del sistema de búsqueda en el Frontend	Completado	12 horas	15 horas
Hacer documentación para la memoria	Completado	4 horas	5 horas
Total	Completado	28 horas	30 horas

Tabla 7.9: Sprint 8

Después, empezó la implementación del sistema de búsqueda avanzado de las oposiciones. Al principio se creó un servicio REST por cada petición, pero al haber tantas combinaciones en la aplicación se cambió por un único servicio REST que tenía como parámetros todos los términos de búsqueda, imitando la forma en la que páginas implementan estas funcionalidades.

Como siempre, se testeó el servicio REST, y después de haber comprobado su validez se empezó a implementar su contraparte en el Frontend.

Con el servicio REST validado y funcional, se empezó a diseñar la funcionalidad dentro de la aplicación. Para las oposiciones se añadió el sistema de búsqueda complejo, y para las especialidades y departamentos se añadió un sistema simple de búsqueda por palabra que permitía al usuario encontrar el resultado deseado.

7.10. Época de prácticas (17/05/2021 - 30/05/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Mejorar estética de la aplicación	Completado	4 horas	4 horas
Seguir estudiando los datos de las oposiciones	Completado	2 horas	6 horas
Hacer documentación para la memoria	En proceso	4 horas	2 horas
Total	3/2 tareas	10 horas	14 horas

Tabla 7.10: Segunda Época de Prácticas y Exámenes

Este fue el segundo periodo en el que se juntaron prácticas de otras asignaturas y demás

exámenes, los cuales impidieron realizar con normalidad el desarrollo.

Esto no impidió que se dedicara cierto tiempo para mejorar el diseño de la aplicación. Fue en esta etapa en la que se cambió el fondo básico verde por un fondo de pantalla que seguía el mismo tono cromático. El fondo seguía un patrón geométrico que ayuda a transmitir la sensación de estar en una clase, además se buscó una imagen libre de derechos y de marcas de agua que empeoraran la estética de la aplicación.

En este periodo también se avanzó en la memoria de la aplicación, debido a que este proceso no implicaba dedicar largas sesiones de tiempo.

Por el mismo motivo, se incentivó el estudio de las oposiciones en busca de datos que pudieran ser relevantes para el usuario. Se creó una versión del Script para realizar pruebas y se encontró que había ciertos documentos que realizaban cambios para ciertas oposiciones.

El Script ayudó para ver la importancia de estos datos, ya que se pudo ver la cantidad de documentos que no eran oposiciones, sino actualizaciones de las mismas.

7.11. Sprint 9 (31/05/2021 - 07/06/2021)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Creación del boceto para la página principal	Completado	4 horas	4 horas
Modificación del boceto del listado de oposiciones para implementar el nuevo sistema de búsqueda	Completado	2 horas	2 horas
Modificación del servicio rest para implementar el nuevo sistema de búsqueda	completado	6 horas	6 horas
Implementación de las nuevas características en el Frontend	completado	6 horas	5 horas
Hacer documentación para la memoria	Completado	6 horas	6 horas
Total	Completado	24 horas	23 horas

Tabla 7.11: Sprint 9

Para este tiempo se realizó una reunión en la que se pidieron añadir más campos de búsqueda en los que el usuario pudiera introducir los datos. También, se vio la necesidad de añadir una ventana de inicio en la que los usuarios pudieran entrar por primera vez y conocer las funcionalidades que tenía la página.

En esta vista podías encontrar una presentación de los datos que había en la página, así como una breve explicación de las funcionalidades que prestaba al usuario.

Adicionalmente, gracias al proceso de investigación de las oposiciones, se encontró que ciertos documentos que incluíamos junto a las oposiciones eran documentos de corrección. Estos documentos referenciaban a las oposiciones y añadían cambios, ofrecían una lista con los aprobados o cancelaban la oposición. Al ver esto, se vio la necesidad de meter estos datos en la aplicación y de presentarlos en forma de un hilo, parecido al concepto de hilo de twitter, en el cual los usuarios pudieran ver el historial de cambios de la oposición.

Este sprint se focalizó en las funcionalidades de la página de inicio y del nuevo sistema de búsqueda.

Para esto, se modificaron los bocetos existentes para añadir soporte a estas nuevas funcionalidades. Ahora se podían filtrar las oposiciones según las especialidades, el departamento y el estado de la oposición junto a la búsqueda por rango de fechas y por el título de la oposición.

Para la ventana de presentación también se tuvieron que crear los bocetos correspondientes. La página debía contener una parte destinada a la explicación de los datos, y como después se vio que la página estaría vacía, se decidió añadir una lista con las oposiciones más recientes del BOE.

Se modificó el modelo de la aplicación, así como los servicios REST existentes para dar soporte a los cambios, y se añadieron los nuevos servicios REST correspondientes al nuevo sistema de búsqueda.

Se trasladaron los conceptos de las interfaces desde los bocetos a la aplicación y se añadió la funcionalidad requerida.

7.12. Sprint 10 (08/06/2021 - 15/06/2021)

Este sprint tenía como objetivo seguir con la implementación de los nuevos requisitos faltantes, así como el análisis de los mismos .

Se modificó el boceto del Detalle de la Oposición para que hubiese espacio para las referencias que tenga el documento seleccionado. El objetivo era que estas referencias se parecieran al concepto de hilo que aparece en la red social Twitter, el cual sirve para mostrar un histórico de cambios en las oposiciones. Estas modificaciones tendrán un botón que permita al usuario ver el documento oficial en la página del Boletín Oficial del Estado.

Dentro de la implementación, se tuvo que modificar el Script de recolección de datos para implementar la extracción de estas nuevas referencias, y también modificar las oposiciones a las que se referencien los nuevos cambios. Esto implica añadir más accesos a la base de datos, lo que implica que el proceso de recolección tarde ahora más tiempo. Esto no es problema ya que el tiempo de recolección de datos no tiene como objetivo ser el más rápido, sino ofrecer un servicio completo y fiable.

Tarea	Estado	Tiempo estimado	Tiempo invertido
Modificación del boceto del detalle de la oposición	Completado	2 horas	2 horas
Modificación del Script para implementar la nueva funcionalidad	Completado	5 horas	4 horas
Implementación del servicio REST de la nueva funcionalidad	Completado	4 horas	2 horas
Rediseñar la vista de detalle de la oposición	Completado	1 horas	1 horas
Implementar nueva funcionalidad en el Frontend	Completado	8 horas	4 horas
Desarrollo de tests automatizados del servicio REST	Completado	10 horas	10 horas
Hacer documentación para la memoria	Completado	6 horas	5 horas
Total	Completado	36 horas	28 horas

Tabla 7.12: Sprint 10

Para el servicio REST se añadió esta nueva funcionalidad dentro del servicio de Oposiciones. Este permitía al usuario consultar si un documento se ha referenciado en el pasado y ver estas referencias, así como también permitir ver las oposiciones a las que referencia el documento.

Usando de guía los bocetos, se editó la vista de detalles de la oposición para añadir la funcionalidad. Dentro de cada documento, se podía consultar si este había sido referenciado por otro documento, o si este documento estaba referenciando a otros.

7.13. Spint Final (18/06/2021 - Entrega del proyecto)

Tarea	Estado	Tiempo estimado	Tiempo invertido
Rematar documentación la memoria del TFG	Completado	20 horas	18 horas
Limpieza del código	Completado	4 horas	2 horas
Total	Completado	20 horas	18 horas

Tabla 7.13: Sprint Final

7.14. RECAPITULACIÓN

El objetivo del sprint era acabar la memoria y depurar el código en busca de imperfecciones.

La memoria a estas alturas estaba inacabada y faltaban temas que tratar. Había que añadir la documentación pertinente referente a las pruebas y la implementación del código. También, había que hacer un resumen del documento y traducirlo al inglés.

Adicionalmente, había que añadir las referencias que se habían usado durante el desarrollo de la memoria en la sección de Bibliografía. También, se buscó mejorar la estética de la memoria añadiendo imágenes en las herramientas, así como capturas del código.

En al depuración del código, se eliminaron los comentarios que se iban haciendo durante el desarrollo sobre las partes que faltaban, etc. También, se hicieron unas últimas pruebas en las que se buscaba eliminar el código sobrante que no se utilizaba.

7.14. Recapitulación

7.14.1. Calendario final

Durante el desarrollo del proyecto, y debido al horario caótico que tenía el estudiante con las asignaturas y exámenes, el calendario final quedaría de esta forma:

Nombre del Sprint	Duración del sprint
Sprint 1	31/01/2021 - 7/02/2021
Sprint 2	08/02/2021 - 15/02/2021
Sprint 3	15/02/2021 - 23/02/2021
Sprint 4	23/02/2021 - 02/03/2021
Sprint 5	04/03/2021 - 14/04/2021
Época de prácticas	14/03/2021 - 10/04/2021
Sprint 6	12/04/2021 - 25/04/2021
Sprint 7	26/04/2021 - 02/05/2021
Sprint 8	04/05/2021 - 12/05/2021
Época de prácticas	17/05/2021 - 30/05/2021
Sprint 9	31/05/2021 - 07/06/2021
Sprint 10	08/06/2021 - 15/06/2021
Sprint Final	15/06/2021 - Entrega Proyecto

Tabla 7.14: Calendario Final

7.14.2. Trabajo Total

A continuación, se presentará una tabla con el trabajo total acumulado durante todos los sprints, junto al trabajo estimado que en un principio se pensaba que iba a durar.

Nombre del Sprint	Trabajo Estimado	Trabajo Real
Sprint 1	25 horas	26 horas
Sprint 2	28 horas	28 horas
Sprint 3	30 horas	32 horas
Sprint 4	34 horas	30 horas
Sprint 5	22 horas	22 horas
Época de prácticas	10 horas	12 horas
Sprint 6	32 horas	30 horas
Sprint 7	21 horas	20 horas
Sprint 8	28 horas	30 horas
Época de prácticas	10 horas	14 horas
Sprint 9	24 horas	23 horas
Sprint 10	36 horas	28 horas
Sprint Final	25 horas	26 horas
Total	300 horas	295 horas

Tabla 7.15: Calendario Final

Capítulo 8

Conclusiones

Acabada el proyecto, se han visto cumplidos gran parte de los objetivos planteados. Gracias al Framework de Ionic, hemos podido convertir la aplicación en una PWA (Aplicación Web Progresiva), lo que permite tener soporte para ordenadores y para dispositivos móviles. Contamos con un Script que permite tener acceder al Boletín Oficial del Estado y crear un registro de las oposiciones que se emiten diariamente. Dentro de la funcionalidad del usuario, contamos con un sistema de búsqueda que permite filtrar las oposiciones según distintos criterios. También, permitimos al usuario poder navegar con facilidad a los documentos oficiales que emite el BOE, presentando una vista con un historial de cambios que ha ido teniendo la oposición.

El proyecto comenzó a finales de Enero durante el inicio del segundo cuatrimestre y realizándose en paralelo junto con 2 asignaturas y las prácticas curriculares. Esto repercutió en el plan de trabajo, ya que había ocasiones en las que el desarrollo se tenía que ver detenido debido a la aparición de exámenes y prácticas. El plan de trabajo se dividió en sprints en los que se establecieron objetivos y tareas a realizar. Durante el proyecto, hubieron dos grandes parones en los que el alumno tuvo que dedicar tiempo a las demás asignaturas, aunque en todo momento hubo un mínimo contacto con el desarrollo.

Durante la primera etapa del proyecto se focalizó en la manera de recoger los datos del Boletín Oficial del Estado. Las oposiciones las emiten los departamentos dividiendo el propósito de estas en epígrafes (el término puede asociarse con la especialidad de la oferta de la oposición). Con esta información, se desarrollo una serie de diagramas de diseño y un Script para recoger los datos de las oposiciones. El objetivo del Script era doble: tener un mecanismo por el cual poder estudiar las oposiciones y la disposición de los datos en el BOE, y también la de rellenar la base de datos que después se utilizaría en la aplicación.

En la segunda etapa, se desarrolló una API REST para implementar un mecanismo de acceso a los datos para la capa de presentación. Los servicios se construyeron en base a los diagramas que se crearon en la anterior etapa. La fase de prueba de estos servicios se hizo con el programa Postman, donde se creó una serie de pruebas automatizadas. Esta etapa fue dividida en dos, ya que se produjo el primer parón del proyecto. Aun así, se realizó la etapa

con éxito.

La última etapa se caracterizó por la construcción de la capa de presentación. Para esta etapa se creó una serie de bocetos para tener una guía visual a la hora de implementar las páginas. Se definió un estilo visual y con ello ya empezó la implementación. Durante esta etapa, se dedicó un tiempo al aprendizaje de Ionic, ya que este no había sido utilizado nunca por el alumno. Aún con este contratiempo y el parón que se produjo durante el desarrollo de esta etapa, se consiguió desarrollar una página web en la que los usuarios pudieran consultar las oposiciones del Boletín Oficial del Estado.

El problema más grande que se ha tenido a lo largo del proyecto, ha sido la falta de datos con un formato definido en las oposiciones. Los datos más importantes estaban en el apartado "texto". Esto no sería problema si, aun así, hubiera un formato mínimo, ya que con un simple proceso de data mining se podrían haber recogido los datos. El problema es que no había un formato definido, ya que dependiendo el formato dependía del departamento en el que se emitiera la oposición. Como se comentó a lo largo de la memoria, se estudiaron los boletines de otras comunidades para ver si estas cumplían un formato, y lo que se sacó en claro es que había oposiciones que sí cumplían un formato definido y otras que no. Debido a este problema, y a la escasa formación en data mining del estudiante, no se pudieron cumplir algunas funcionalidades deseadas.

8.1. Trabajo Futuro

El proyecto ofrece un servicio completo en el que el usuario pueda consultar las oposiciones usando diversas funcionalidades, pero aun con todo esto, el proyecto no es perfecto y hay detalles que se pueden añadir:

- **Mejora del aspecto visual.** Debido a la aparición de imprevistos que implicaron tener parado el proyecto durante varias semanas, este aspecto de la aplicación tuvo que verse mermado. El aspecto visual no es del todo malo, pero puede mejorarse mucho.
- **Implementación de data mining en la recogida de los datos.** El mayor problema que se ha tenido a la hora de recoger los datos de las oposiciones, es la falta de un formato claro dentro del BOE. La información más importante (numero de plazas que ofrece la oposición, la fecha de recogida de solicitud, etc) no siguen un formato claro. Esto puede arreglarse mediante un complejo sistema de data mining. Esta solución no ha podido hacerse en este proyecto, ya que el alumno no tenía conocimientos acerca de estos métodos, por lo cual, el proyecto se ha tenido que hacer con la poca información formateada.
- **Guardar los datos del PDF en la base de datos.** Actualmente, la base de datos es bastante ligera en términos de memoria. Esto se puede cambiar, guardando la información de las oposiciones en la base de datos, para poder acceder a su PDF, independientemente de si su versión dentro del boletín está disponible. Con esto, podemos aumentar la independencia de la página oficial.

- **Cambiar el sistema de búsqueda con los nuevos datos.** Este aspecto va de la mano con el proceso anterior. Como se ha dicho antes, es muy interesante por parte del usuario poder conocer el número de plazas y la fecha de solicitud de una oposición dada, entre otros datos. Para esto, se podría ofrecer al usuario la posibilidad de conocer las oposiciones que tengan abierto el plazo de solicitud, pudiendo eliminar otros métodos de búsqueda que sean menos usados.
- **Mejorar captura de errores.** El programa ha sido testeado en busca de fallos, pero no se han podido crear métodos automáticos de testeo. Esta parte puede ser de gran utilidad a la hora de añadir funcionalidad, ya que se podría probar que no se cometiera ningún fallo en la funcionalidad actual.

Apéndice A

Manual

A.1. Instalación de las Herramientas

En capítulos anteriores se han descrito las herramientas que se han utilizado durante el desarrollo de la aplicación.

A.1.1. Versión de las herramientas

- NetBeans: Debe tener el complemento para programas en JEE. Versión usada para el proyecto: 8.0.1.
- Java EE (Enterprise Edition), versión 7 del SDK.
- JDK. Versión 8, JDK1.8.0_144.
- Oracle GlassFish, versión para este servidor de aplicaciones es la 4.1
- Apache Derby, por simplicidad se va a usar la versión de Oracle que viene con el JDK.
- Visual Studio Code
- NodeJS
- Angular
- Ionic

A.1.2. Instalación del Backend

Después de descargar las versiones comentadas, se procederá siguiendo lo siguiente:

- Instalar el JDK
- Instalar NetBeans, evitando la opción de instalar el servidor Glassfish ya que esta versión da problemas. El resto de las opciones dan igual.
- Desempaquetar el archivo zip que contiene el servidor Glassfish y guardar la carpeta en una dirección conocida, por ejemplo en la carpeta Documentos.
- Añadir el servidor Glassfish seleccionando la opción Añadir Servidor dentro de la ventana Servicios/Servidores. Aquí seleccionar el tipo de servidor Glassfish y rellenando la ruta del servidor con la dirección de la carpeta donde habíamos extraído la versión de Glassfish.
- Añadir la base de datos seleccionando la opción Nueva Conexión dentro de la ventana Servicios/Base de Datos. Aquí, seleccionar a Derby como tipo de base de datos y rellenando la dirección de la carpeta con la versión de Apache Derby.

Esta versión soporta los servicios

A.1.3. Instalación del Frontend

Para esta sección, solo hay que instalar la aplicación de Visual Studio Code y node.js, ya que los otros servicios se descargarán desde el terminal de la aplicación.

- Instalar Visual Studio Code
- Abriremos la consola desde el buscador de aplicaciones o simplemente escribiendo cmd dentro del explorador de archivos

Después, escribiremos:

```
npm install -g ionic
```

Con todo esto ya tendríamos todas las herramientas dispuestas

A.2. Configuración de la Aplicación

Una vez tengamos los ficheros de la aplicación descargados, descomprimos el fichero (si está comprimido) y moveríamos la carpeta en un directorio conocido, por ejemplo la carpeta Documentos.

El contenido de la carpeta estará dividido entre la parte Frontend y la Backend.

A.2.1. Ejecutar parte Backend

Dentro del Backend están el Script y el proyecto con la API REST.

Para el Script, necesitaremos localizar la nueva carpeta que contiene los drivers de la base de datos Apache Derby. Una vez, actualicemos las referencias, tendremos que crear la base de datos en la que queramos mantener los datos y modificar las referencias dentro del Script para esta nueva base de datos. Con esto, ya estaría funcional el Script.

La ejecución del Script puede verse acompañada por un gestor de tareas (como el de Windows) que ejecute el .JAR diariamente sin argumentos. Es recomendable, realizar una ejecución pasándole una fecha como argumento del formato DD MM YYYY para llenar la base de datos de oposiciones, ya que hay datos que referencian a cambios de oposiciones y estas al no estar en la base de datos no se guardan.

Para la API REST necesitaremos modificar el fichero de configuración para actualizar la nueva base de datos de la que queramos sacar los datos. Y simplemente con eso ya estaría funcional. Para ejecutar el servicio, habría que seleccionar la opción de deploy y esto nos devolvería una dirección en la que se podrían empezar a realizar las peticiones.

A.2.2. Ejecutar parte Frontend

Para tener funcional la parte Frontend, se necesita tener funcional la capa de acceso a los datos. Aun así, se podría ejecutar sin problemas, pero la aplicación devolvería errores ya que no habría datos que recoger.

Para abrir la aplicación, lo recomendable es navegar por el explorador de archivos y entrar dentro del proyecto Ionic. Ahí, escribiremos dentro del buscador el término cmd, esto se podría sustituir por navegar desde el terminal hasta la carpeta del proyecto.

Una vez dentro del terminal, abriremos el proyecto escribiendo "code.". No habrá que configurar nada, lo único que se tendría que configurar sería la dirección del Servicio REST en caso de que en la sección anterior se haya modificado este campo.

Para ejecutar la aplicación web, el proceso sería abrir el terminal de la aplicación y ejecutar el comando "ionic serve". Cuando termine se abrirá la página de inicio dentro del navegador predeterminado.

A.3. Manual de uso de la aplicación

La aplicación, al ser una aplicación web, necesita un navegador web en la que poder ver las vistas. Durante el desarrollo de la aplicación, el navegador que se ha ido utilizando es el Mozilla Firefox. Este navegador permite cambiar el aspecto a un simulador móvil. Esta característica permite poder comprobar la usabilidad de la aplicación en un entorno móvil.

A continuación, la explicación se realizará utilizando Mozilla Firefox. Se podría probar la aplicación con otro navegador, pero algunos "comandos" serán distintos. En la ventana

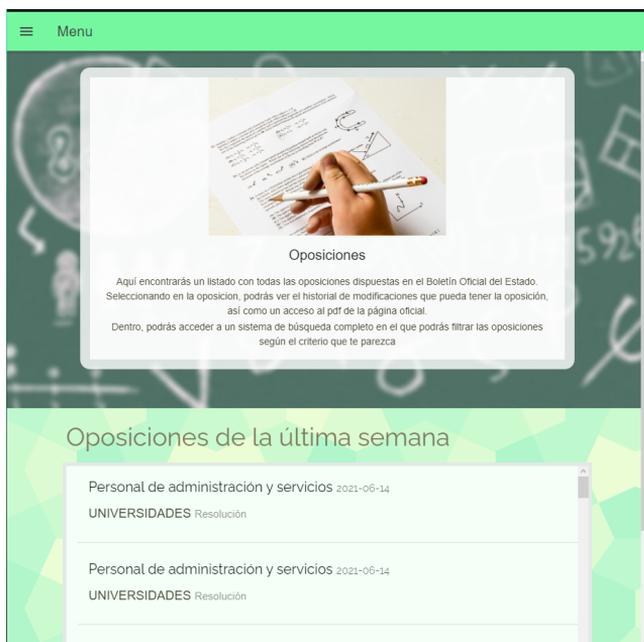


Figura A.1: Ventana Inicial de la Aplicación

inicial, encontraremos una explicación de los datos, así como una lista con las oposiciones "novedad" que se ha dispuesto a lo largo de la semana. Para movernos en la presentación, bastará con deslizar a los lados para ir cambiando de explicación. Para acceder a las demás vistas, se podrá utilizar el menú superior o seleccionar la explicación.

Aquí, se procederá a explicar la forma de cambiar la vista a dispositivo móvil. En Firefox y en Google Chrome, para abrir las herramientas de desarrollo hay que presionar el F12, en otros navegadores como Opera, la combinación de botones es la siguiente: ctrl+shift+j.

Una vez abierto la herramienta de desarrollo, veremos el icono de un móvil A.6, que si lo presionamos podemos cambiar de vista a la pestaña. Es recomendable refrescar la página apretando F5, porque la aplicación no está preparada para estos cambios rápidos de dispositivo.



Figura A.2: Herramienta de desarrollo de Firefox



Figura A.3: Vista móvil de la aplicación

El funcionamiento de las vistas correspondientes al listado de algún elemento de la aplicación son bastante similares. Tienen un sistema de búsqueda y debajo presentan el listado de los elementos. En el caso de las oposiciones, el sistema de búsqueda tiene más entradas con las cuales el usuario puede interactuar. Una vez están ingresados todas las opciones deseadas, el usuario deberá apretar el botón buscar para que la aplicación realice la búsqueda. En las vistas de Especialidades y Departamentos, el funcionamiento es el mismo, puedes buscar el nombre del elemento y una vez seleccionado, la aplicación te llevará a una página que tiene como título el tipo de elemento que se ha seleccionado (Departamentos o Especialidades)

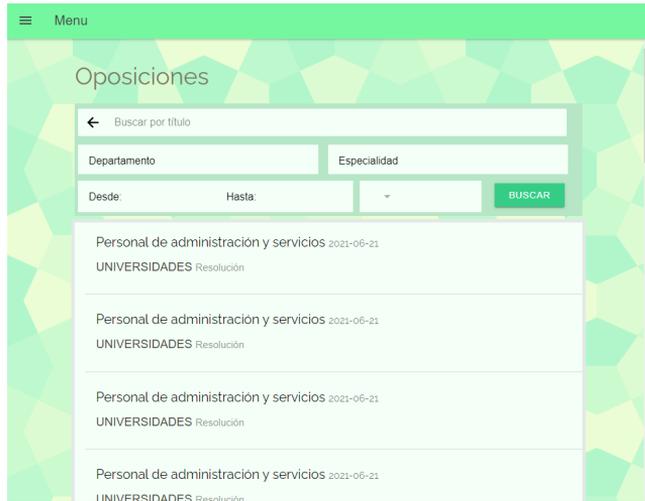


Figura A.4: Vista del listado de las oposiciones

junto al nombre del elemento y el tipo de elemento al que se está refiriendo. Si se selecciona

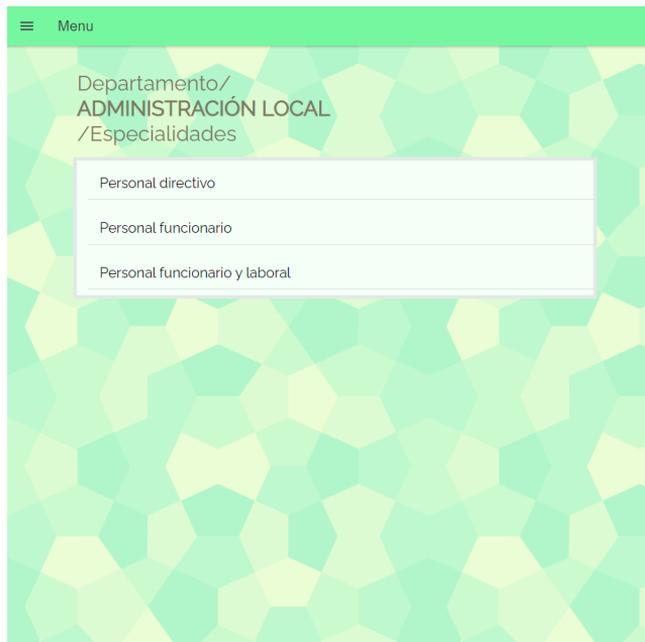


Figura A.5: Vista del detalle de un Departamento

uno de estos elementos, volveremos a la vista de las oposiciones con el par Departamento-Especialidad definido. Seleccionando cualquiera de estas oposiciones, aparecerá una ventana emergente con el detalle de la oposición y documentos a los que está relacionado.

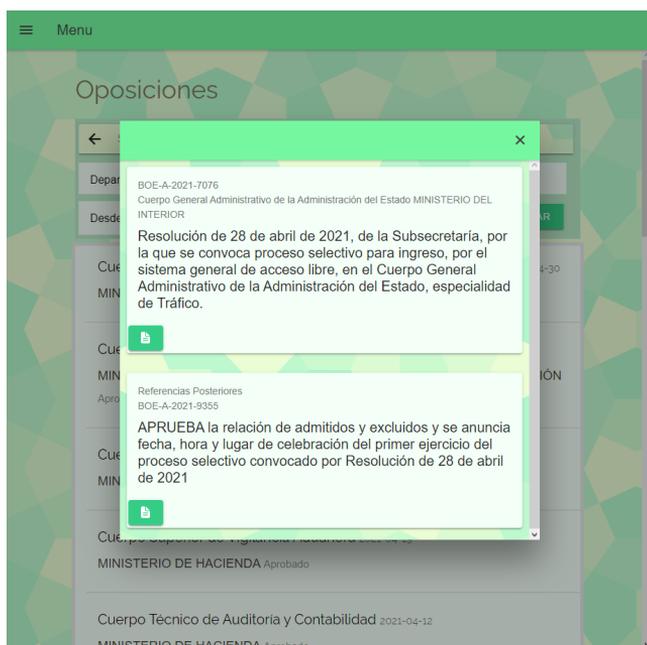


Figura A.6: Vista en detalle de una oposición

Bibliografía

- [1] Apache. Apache derby. <https://db.apache.org/derby/>, 2021. Último acceso: 08-05-2021.
- [2] Astah. Astah professional. <https://astah.net/products/astah-professional/>, 2021. Último acceso: 16-04-2021.
- [3] Autograndad. Glassfish. <https://amp.es.autograndad.com/1014314/1/glassfish.html>, 2020. Último acceso: 08-05-2021.
- [4] Azahara Benito. Los 8 principios básicos de los datos abiertos. <https://www.ogooov.com/es/blog/los-8-principios-basicos-de-los-datos-abiertos/>, 2019. Último acceso: 26-04-2021.
- [5] brunocascio. Resumen ingeniería de software 2 (diseño, pruebas y mantenimiento). <https://gist.github.com/brunocascio/5e89fafa7fd86bdd1a715d2f6f0432d1>, 2016. Último acceso: 01-05-2021.
- [6] Capterra. Pencil project. <https://www.capterra.es/software/176481/pencil-project>, 2018. Último acceso: 12-04-2021.
- [7] Víctor Cuervo. ¿qué es apache derby? <https://www.oracle.com/middleware/technologies/glassfish-server.html>, 2016. Último acceso: 08-05-2021.
- [8] Junta de Andalucía. Netbeans. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/888>, 2019. Último acceso: 03-05-2021.
- [9] Junta de Castilla y León. Datos abiertos de castilla y león. <https://datosabiertos.jcyl.es/web/es/datos-abiertos-castilla-leon.html>, 2021. Último acceso: 26-04-2021.
- [10] Junta de Castilla y León. ¿quÉ son los datos abiertos? <https://datosabiertos.jcyl.es/web/es/iniciativa-datos-abiertos/datos-abiertos.html>, 2021. Último acceso: 26-04-2021.
- [11] Agencia Estatal Boletín Oficial del Estado. Buscar. <https://www.boe.es/buscar/>, 2021. Último acceso: 22-04-2021.

- [12] Agencia Estatal Boletín Oficial del Estado. Documento xml de la oposición boe-a-2020-9967. https://boe.es/diario_boe/xml.php?id=BOE-A-2020-9967, 2020. Último acceso: 23-04-2021.
- [13] Agencia Estatal Boletín Oficial del Estado. Documento xml del documento boe-s-20200201. https://boe.es/diario_boe/xml.php?id=BOE-S-20200201, 2020. Último acceso: 23-04-2021.
- [14] Agencia Estatal Boletín Oficial del Estado. Personal: ayuda y contenido. https://www.boe.es/buscar/ayudas/personal_ayuda.php, 2021. Último acceso: 22-04-2021.
- [15] Agencia Estatal Boletín Oficial del Estado. Personal: Oposiciones, nombramientos... <https://www.boe.es/buscar/personal.php>, 2021. Último acceso: 22-04-2021.
- [16] Evolus. Pencil project. <https://pencil.evolus.vn>, 2021. Último acceso: 07-05-2021.
- [17] María Isabel Alfonso Galipienso. Servicios rest. <http://expertojava.ua.es/experto/restringido/2014-15/rest/rest.html>, 2014. Último acceso: 23-05-2021.
- [18] Google. Angularjs. <https://angularjs.org>, 2021. Último acceso: 07-05-2021.
- [19] Ionic. Ionic. <https://ionicframework.com>, 2021. Último acceso: 07-05-2021.
- [20] Krama. ¿qué es ionic? <https://www.krama.es/blog-20-04-29-que-es-ionic.html>, 2020. Último acceso: 07-05-2021.
- [21] Alejandro López. ¿qué es postman? <https://openwebinars.net/blog/que-es-postman/>, 2019. Último acceso: 13-04-2021.
- [22] Sara López. Aplicaciones web progresivas: qué son, cómo funcionan y qué tienes que saber. [https://www.digital55.com/desarrollo-tecnologia/que-es-pwa-ventajas-desventajas/#:~:text=Funcionalidades%20propias%20de%20una%20App%20nativa&text=Las%20Progressive%20Web%20App%20pueden,no%20está%20abierta%20la%20PWA\).](https://www.digital55.com/desarrollo-tecnologia/que-es-pwa-ventajas-desventajas/#:~:text=Funcionalidades%20propias%20de%20una%20App%20nativa&text=Las%20Progressive%20Web%20App%20pueden,no%20está%20abierta%20la%20PWA).), 2020. Último acceso: 01-06-2021.
- [23] BBVA API Market. Api rest: qué es y cuáles son sus ventajas en el desarrollo de proyectos. , 2016. Último acceso: 23-05-2021.
- [24] Microsoft. Visual studio code. <https://code.visualstudio.com>, 2021. Último acceso: 03-05-2021.
- [25] Mozilla. Mozilla firefox. <https://www.mozilla.org/es-ES/firefox/new/>, 2021. Último acceso: 13-04-2021.
- [26] Apache NetBeans. Netbeans. <https://netbeans.apache.org>, 2021. Último acceso: 03-05-2021.
- [27] Oracle. Glassfish server. <https://www.oracle.com/middleware/technologies/glassfish-server.html>, 2021. Último acceso: 08-05-2021.
- [28] Postman. Postman. <https://www.postman.com>, 2021. Último acceso: 02-06-2021.

- [29] Juan Ranchal. Aplicaciones web progresivas: qué son, cómo funcionan y qué tienes que saber. <https://www.muycomputerpro.com/2019/09/26/aplicaciones-web-progresivas-que-son-como-funcionan-y-que-tienes-que-saber>, 2019. Último acceso: 01-06-2021.
- [30] Sara Serrano. Eventos en scrum i. <https://www.saraclip.com/eventos-en-scrum/>, 2017. Último acceso: 21-05-2021.
- [31] Sara Serrano. Eventos en scrum ii. <https://www.saraclip.com/eventos-en-scrum-ii/>, 2017. Último acceso: 21-05-2021.
- [32] Team. ¿qué son los datos abiertos? <https://www.opendatasoft.com/es/blog/2017/05/09/que-son-los-datos-abiertos>, 2017. Último acceso: 26-04-2021.
- [33] ViewNext. Artefactos scrum ¿qué son y para qué sirven? <https://www.viewnext.com/artefactos-scrum/>, 2019. Último acceso: 22-05-2021.
- [34] Change Vision. astah* professional. <https://software.com.ar/p/astah-professional#product-description>, 2019. Último acceso: 16-04-2021.
- [35] Wikipedia. Angular (framework). [https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework)), 2021. Último acceso: 07-05-2021.
- [36] Wikipedia. Mozilla firefox. https://es.wikipedia.org/wiki/Mozilla_Firefox, 2021. Último acceso: 13-04-2021.
- [37] Wikipedia. Scrum (desarrollo de software). [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)), 2021. Último acceso: 21-05-2021.
- [38] Wikipedia. Visual studio code. https://es.wikipedia.org/wiki/Visual_Studio_Code, 2021. Último acceso: 03-05-2021.