



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

**Plataforma de Televoto de Ámbito General y con
Carácter Anónimo**

Alumno: Francisco de la Puente Rivera

Tutor: José Vicente Álvarez Bravo

Plataforma de Televoto de Ámbito General y con Carácter Anónimo

Francisco de la Puente Rivera

Índice general

Lista de figuras	III
Lista de tablas	V
Resumen	XI
I Memoria del proyecto	1
1. Descripción del proyecto	3
1.1. Introducción	3
1.2. Objetivos del trabajo	4
1.3. Entorno de aplicación	4
2. Metodología	9
2.1. Proceso de desarrollo	9
2.2. Herramientas utilizadas	12
2.3. Arquitectura	13
2.4. Diseño de interfaz	15
3. Planificación	17
3.1. Estimación del esfuerzo	17
3.2. Planificación temporal	22
3.3. Presupuesto económico	24
3.3.1. Hardware y software	24
3.3.2. Recursos humanos	25
3.3.3. Presupuesto total	25
II Documentación técnica	27
4. Análisis	29
4.1. Requisitos	29
4.1.1. Iniciativa I: Creación de la aplicación de administración	29

4.1.2. Iniciativa II: Creación de la aplicación para usuarios	42
4.2. Atributos de calidad	47
5. Diseño	49
5.1. Diseño de datos	49
5.2. Diagrama de clases	50
5.3. Diagramas de secuencia	51
5.4. Diagramas de estado	54
6. Implementación	55
6.1. Implementación de la aplicación backend	55
6.2. Implementación de la aplicación de administración	59
6.3. Implementación de la aplicación para usuarios	60
7. Pruebas	63
7.1. Pruebas unitarias	63
7.2. Pruebas de integración	65
7.3. Pruebas funcionales	66
III Manuales de las aplicaciones	69
8. Manuales de instalación	71
8.1. Manual de instalación del backend y la aplicación web	71
8.2. Manual de instalación de la aplicación móvil	71
9. Manuales de usuarios	73
9.1. Manual de administración	73
9.2. Manual de usuario	83
IV Conclusiones	89
10. Trabajo futuro y conclusiones	91
10.1. Trabajo futuro	91
10.2. Conclusiones personales	92
Bibliografía	93

Índice de figuras

1.1. Proyecto de los Presupuestos Participativos Valladolid 2022	5
1.2. Resultado de los Presupuestos Participativos Valladolid 2022	6
1.3. Infografía votación Eurovisión Junior 2020	7
2.1. Arquitectura física de la plataforma	14
2.2. Arquitectura lógica de la plataforma	14
2.3. Paleta de colores de la plataforma	16
2.4. Logotipo de la plataforma	16
3.1. Periodos de trabajo de Sprints	23
5.1. Diagrama de clases	50
5.2. Diagrama de secuencia de la creación de una votación	51
5.3. Diagrama de secuencia del uso de una opción de un conjunto de opciones en una votación	52
5.4. Diagrama de secuencia del envío de voto	53
5.5. Diagrama de estados de una votación	54
7.1. Pruebas unitarias	64
7.2. Pruebas de integración, inicio de sesión	65
7.3. Pruebas de integración, entidad del usuario	66
9.1. Inicio de sesión en la aplicación de administración	73
9.2. Pantalla de inicio de la aplicación de administración	74
9.3. Pantalla de entidades de la aplicación de administración	74
9.4. Modal de edición de entidades	75
9.5. Modal de creación de subentidades	75
9.6. Modal de bloqueo de subentidades	76
9.7. Pantalla de conjuntos de opciones	76
9.8. Modal de conjuntos de opciones	77
9.9. Tarjeta de edición de opciones	77
9.10. Selección de imagen de una opción	78
9.11. Pantalla de votaciones de la aplicación de administración	78
9.12. Modal de creación de votaciones (1)	79

9.13. Modal de creación de votaciones (2)	79
9.14. Pantalla de votación de la aplicación de administración	80
9.15. Modal de edición de imagen de la votación	80
9.16. Pantalla de selección de opciones para una votación	81
9.17. Pantalla de claves privadas de una votación	82
9.18. Pantalla de resultados de una votación	82
9.19. Guía de inicio de la aplicación para usuarios	83
9.20. Pantalla principal de la aplicación para usuarios	83
9.21. Modal de votación de la aplicación para usuarios	84
9.22. Estación de votación	84
9.23. Pantalla de entidad de la aplicación para usuarios	85
9.24. Búsqueda en la aplicación para usuarios	85
9.25. Búsqueda de una entidad en la aplicación para usuarios	86
9.26. Búsqueda de una votación en la aplicación para usuarios	86
9.27. Pantalla de información de la aplicación para usuarios	87

Índice de cuadros

3.1. Estimación Sprint 1	18
3.2. Estimación Sprint 2	19
3.3. Estimación Sprint 3	19
3.4. Estimación Sprint 4	19
3.5. Estimación Sprint 5	20
3.6. Estimación Sprint 6	20
3.7. Estimación Sprint 7	20
3.8. Estimación Sprint 8	20
3.9. Estimación Sprint 9	21
3.10. Estimación Sprint 10	21
3.11. Presupuesto de hardware y software	24
3.12. Presupuesto de recursos humanos	25
3.13. Presupuesto total	25
4.1. Épicas de la Iniciativa I	29
4.2. HU-AD-LV1, Listado de votaciones	30
4.3. HU-AD-LV2, Filtros lista votaciones	30
4.4. HU-AD-LV3, Nueva votación	30
4.5. HU-AD-LV4, Ver votación	31
4.6. HU-AD-LV5, Iniciar votación	31
4.7. HU-AD-LV6, Finalizar votación	31
4.8. HU-AD-LV7, Eliminar votación	32
4.9. HU-AD-V1, Información principal de una votación	33
4.10. HU-AD-V2, Nueva opción en votación	33
4.11. HU-AD-V3, Nueva opción desde conjunto de opciones en votación	33
4.12. HU-AD-V4, Editar opción de votación	34
4.13. HU-AD-V5, Eliminar opción de votación	34
4.14. HU-AD-V6, Generar claves privadas	34
4.15. HU-AD-V7, Descarga claves privadas	35
4.16. HU-AD-V8, Visualizar resultados votación	35
4.17. HU-AD-E1, Ver información de mi entidad	36
4.18. HU-AD-E2, Modificar información de mi entidad	36
4.19. HU-AD-E3, Modificar datos inicio de sesión	36

4.20. HU-AD-E4, Crear subentidad	37
4.21. HU-AD-E5, Lista de subentidades	37
4.22. HU-AD-E6, Bloqueo y desbloqueo subentidad	37
4.23. HU-AD-CO1, Crear conjunto de opciones	38
4.24. HU-AD-CO2, Renombrar conjunto de opciones	38
4.25. HU-AD-CO3, Eliminar conjunto de opciones	38
4.26. HU-AD-CO4, Nueva opción en conjunto de opciones	39
4.27. HU-AD-CO5, Editar opción en conjunto de opciones	39
4.28. HU-AD-CO6, Eliminar opción de conjunto de opciones	39
4.29. HU-AD-S1, Iniciar sesión	40
4.30. HU-AD-S2, Cerrar sesión	40
4.31. HU-AD-S3, Acceso a página privada	40
4.32. HU-AD-S4, Sesión ya iniciada	41
4.33. HU-AD-EI1, Menú lateral	41
4.34. HU-AD-EI2, Página principal	41
4.35. Épicas de la Iniciativa II	42
4.36. HU-US-L1, Listado votaciones	42
4.37. HU-US-L2, Listado de votaciones de una entidad	42
4.38. HU-US-L3, Listado de votaciones invisibles de una entidad	43
4.39. HU-US-L4, Búsqueda de votación	43
4.40. HU-US-L5, Búsqueda de entidad	43
4.41. HU-US-V1, Pantalla de votación	44
4.42. HU-US-V2, Introducir clave privada	44
4.43. HU-US-V3, Votación de una opción	44
4.44. HU-US-V4, Votación de varias opciones	45
4.45. HU-US-V5, Votación de ranking	45
4.46. HU-US-V6, Votación de suma de valores	45
4.47. HU-US-EI1, Barra de navegación	46
4.48. HU-US-EI2, Guía de primera vez	46
4.49. HU-US-EI3, Pantalla de información	46
7.1. Prueba funcional, correspondiente a la Historia HU-AD-LV3	67
7.2. Prueba funcional, correspondiente a la Historia HU-US-L1	68
7.3. Prueba funcional, correspondiente a la Historia HU-US-V1	68

La que quieres.

Agradecimientos

Me gustaría dar las gracias en primer lugar a mi familia, porque ellos me han permitido llegar hasta aquí, dejándome elegir siempre mi camino.

También me gustaría agradecer a todos los profesores que han dedicado parte de su tiempo a mi formación y a la del resto de mis compañeros, y en especial agradecer a mi tutor, D. José Vicente Álvarez Bravo, por toda su ayuda a lo largo del Grado y por aceptar tutorizarme en este Trabajo Fin de Grado.

Por último, gracias a mis amigas y amigos por haber estado a mi lado durante todo este tiempo, porque aunque no se lo recuerde lo suficiente, sin su apoyo no habría podido llegar hasta aquí.

Resumen

Cada vez más instituciones, organizaciones o empresas buscan recibir la opinión pública sobre un tema del que son responsables, ya sea por tomar decisiones colaborativas o conseguir atención de su público potencial, pero hay una falta de innovación en este ámbito que dificulta que estas votaciones se puedan realizar de manera anónima a la vez que se mantiene un control sobre los usuarios para que no se salten sus normas.

En este Trabajo Fin de Grado se pretende crear una plataforma de televoto de ámbito general que garantice el anonimato de sus usuarios, explicando su previa planificación y el análisis, diseño e implementación que se han llevado a cabo.

Palabras claves: Televote, votaciones, plataforma full stack, aplicación web, aplicación móvil.

Parte I

Memoria del proyecto

Capítulo 1

Descripción del proyecto

1.1. Introducción

Mediante este Trabajo Fin de Grado se pretende documentar la creación de una plataforma de televoto, desde sus fases iniciales de análisis y planificación, hasta su implementación y final despliegue en un entorno real.

El motivo por el que se elige el tema de este Trabajo es la falta de innovación que ocurre en su ámbito. Son cada vez más las instituciones que buscan recibir la opinión pública sobre un tema concreto del que son responsables, o en casos menos serios, como puede ser un programa de televisión que permite a sus espectadores tomar decisiones que verán reflejadas en sus pantallas para así generar interés y evitar que cambien de canal. Pero en todos los casos surge el mismo problema: es necesario que el usuario cree una cuenta personal para poder controlar que solo vote una vez y no se salte las normas.

En los tiempos que estamos, en los que se busca cada vez más la protección de nuestros datos personales, estamos cediendo nuestros correos, números de teléfono, nombres y apellidos casi a cualquier paso que damos en internet. Y no hace falta entrar en el debate de si cada cuenta creada debe ir asociada una persona o si toda la red debe ser totalmente anónima, pero si de manera personal puedes ir a un lugar donde, con una papeleta y una urna, puedes elegir al compañero de clase que quieres que te represente en el consejo estudiantil, por ejemplo, ¿por qué para hacerlo telemáticamente debes sacarte un *selfie* mostrando un documento que te identifique? No es lo mismo enseñar un DNI a la persona encargada de reunir las papeletas que mandar una imagen con datos sensibles por internet.

Y es que actualmente contamos con las tecnologías necesarias para evitar todo esto. Es más, con todas estas tecnologías se podría abordar el problema de que cada institución, organización, empresa o **entidad** necesite desarrollar una aplicación propia desde la que consultar a quien desee el tema que quieran. Gracias a la abstracción se podrían ver los puntos comunes que tienen todas estas aplicaciones y desarrollar una solución que pueda ser utilizada de manera general por todas esas entidades. Y, aunque no se puedan satisfacer todas las necesidades de una plataforma como esta en un primer momento, por algún sitio hay que empezar.

Teniendo en cuenta todo esto, el Trabajo se divide en 4 grandes bloques:

1. El primer bloque consiste en la **memoria del proyecto**, donde se incluye la motivación y los objetivos, además de una explicación sobre cómo se ha trabajado para llevarlo a cabo y la planificación y estimación de todo lo que se ha querido incluir.
2. El segundo bloque trata la **documentación técnica** del Trabajo, desde su análisis y diseño inicial a la implementación, incluyendo un capítulo dedicado a las pruebas que se han incluido para asegurar su correcto funcionamiento.
3. El tercer bloque incluye los **manuales de las aplicaciones**, tanto para la instalación de las distintas aplicaciones como para su manejo por parte de los usuarios finales.
4. El cuarto y último bloque contiene las **conclusiones** del Trabajo después de ser realizado, desde los puntos de vista del desarrollo y el personal, además de las posibles funciones que se podrían agregar en el futuro.

1.2. Objetivos del trabajo

La principal finalidad de este Trabajo es poder agrupar todos los conocimientos adquiridos a lo largo de los estudios de Grado. Para conseguirlo, se propondrán objetivos más concretos que conduzcan a ese fin:

- Desarrollar la plataforma con una arquitectura realista en el mundo actual, con capas de persistencia, lógica de negocio y visual.
- Utilizar tecnologías novedosas que faciliten la implementación de grandes soluciones de software.
- Antes del uso de esas tecnologías, investigar sus funcionamientos y relacionarlos con los fundamentos aprendidos en los estudios.
- Implementar marcos de trabajo ágiles durante el transcurso del Trabajo que faciliten el proceso de desarrollo de este.
- Realizar un proceso de análisis y diseño exhaustivo previo a la implementación, para así poder conocer el estado final que se desea conseguir.
- Documentar correctamente cada funcionalidad, actividad o prueba presente en la plataforma.

1.3. Entorno de aplicación

En esta sección se estudiarán los casos de dos aplicaciones de muy distinto ámbito que tienen como fin obtener la opinión de un conjunto de personas. Cada una de estas aplicaciones es propia de una organización y son utilizadas en un momento puntual del

año, por lo que el coste de su creación se amortiza unos pocos días únicamente. Aún así, ambas aplicaciones pertenecen a entidades públicas, por lo que se entiende que su intención no era obtener beneficios.

Presupuestos participativos: cada vez más ayuntamientos se unen a esta iniciativa como, por ejemplo, el Ayuntamiento de Valladolid, que tras un año de descanso debido a la pandemia recuperó la convocatoria el año 2021. Esta iniciativa consta de varias fases, las primeras con el objetivo de recibir proyectos y hacer una elección de los que son viables, todo de manera pública. En la última fase, el Ayuntamiento abre una votación en la que cualquier residente en la ciudad puede participar.

Recuperación como patrimonio industrial del depósito de agua de San Pedro

4 44907412G · 30/01/2021 · Zona Esgueva 2 ·



Depósito de agua de San Pedro Regalado

Figura 1.1: Proyecto de los Presupuestos Participativos Valladolid 2022

Cada persona es asignada a la zona de la ciudad donde vive y se le ofrecen 500.000€ que puede repartir entre los diferentes proyectos correspondientes a su zona. Al final, los proyectos más votados por los ciudadanos son incluidos en los presupuestos de la ciudad del año siguiente.

Esta aplicación es muy completa, pero a menudo se ve afectada por fallos en la autorización que impiden la participación de algunos ciudadanos. Tiene implementado un acceso a la aplicación mediante un código único que genera el Ayuntamiento y permite al ciudadano acceder sin registrarse, aunque realmente ofrece ambas posibilidades. Otra función que destaca es el método de votación, ya que permite que



Figura 1.2: Resultado de los Presupuestos Participativos Valladolid 2022

sea el usuario quien reparta el presupuesto total con libertad, pudiendo elegir pocos proyectos de elevado coste o una gran cantidad de pequeños proyectos.

Eurovision Junior: En el mundo de la televisión existen multitud de ejemplos de aplicaciones de votación. La mayoría de ellas son accesibles a través de un registro, mediante un correo o una red social, y ofrece la posibilidad de votar a una única opción. Pero existe un programa, el Festival de Eurovisión, que en su versión infantil experimenta con las nuevas tecnologías para permitir a su principal audiencia, los niños, la posibilidad de votar por sus canciones favoritas.

Hay quienes dicen que buscan conseguir una tecnología fiable para poder sustituir el principal método de votación en la televisión, los SMS. Y es que si a día de hoy se siguen utilizando los SMS es porque permiten controlar muchos detalles que todavía no se han visto igualados en aplicaciones de votación. Las votaciones de Eurovisión Junior tienen defectos, aunque año tras año se intentan solucionar. Estos son algunos de ellos y su situación previa al quinto año de uso del sistema:

- **Caídas en el servicio.** El primer año de implantación resultó imposible votar con la aplicación, ya que no estaba preparada para el nivel de carga que recibió. Esto se mejoró el segundo año, y actualmente este problema se encuentra solucionado.
- **Múltiples votos.** Esta aplicación evita la necesidad de crear una cuenta para votar guardando *cookies* en el navegador del usuario, y así controlar que una misma persona no pueda votar más de una vez. O al menos esa es la intención, porque si borras los datos del navegador puedes volver a emitir un voto. No hay una mejor forma de controlarlo al tratarse de una aplicación web, pero desde la organización observan las IP de los usuarios que votan para, en casos extremos, anular los votos.
- **Votar por tu propio país.** Al tratarse de un festival infantil, esta opción se ofrece como un añadido, pero realmente no lo es. No saber la ubicación desde donde se manda el voto es un problema serio al que se enfrentarían si quisieran exportar este sistema a la versión adulta de Eurovisión, y es que con los SMS es tan sencillo como ver el prefijo del teléfono que manda el voto. Seguramente se encuentren trabajando en las opciones que tienen para controlarlo, como pedir los permisos de ubicación al usuario o comprobar la localización de su IP, pero ambas opciones se pueden falsear muy fácilmente y pedir datos personales para descubrirlo no parece una opción.

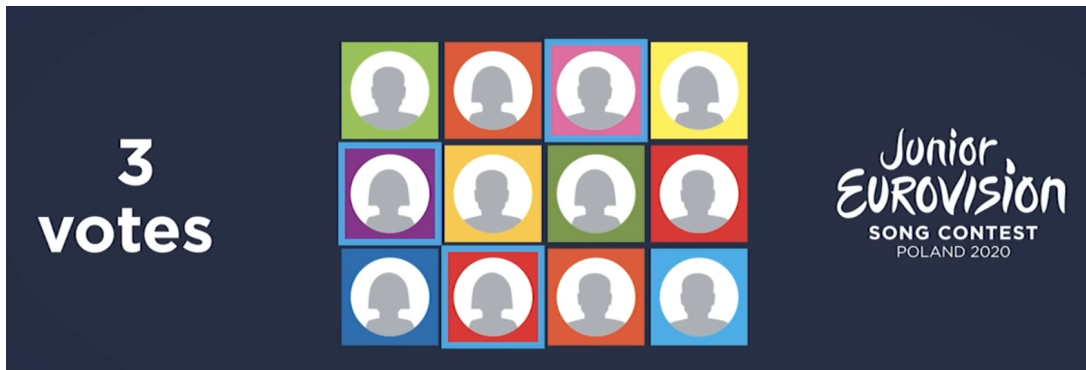


Figura 1.3: Infografía votación Eurovisión Junior 2020

Dejando atrás estos problemas, la aplicación cuenta con un diseño muy actual y adaptable a cualquier tamaño de pantalla. Además, permiten al usuario elegir más de una opción para votar, los primeros años de 3 a 5 opciones y actualmente solo 3.

Capítulo 2

Metodología

Este Trabajo Fin de Grado se realizará dentro de un marco de trabajo ágil, similar a Scrum pero adaptado al trabajo personal que este supone, apoyado de distintas herramientas que se utilizarán para la correcta ejecución del Trabajo.

2.1. Proceso de desarrollo

Scrum es un marco de trabajo compuesto de procesos que se utiliza para gestionar el trabajo de productos complejos. Está compuesto por los Equipos Scrum, sus Roles, Eventos, Artefactos y Reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y esencial para el éxito de Scrum y para su uso.

Al tratarse de un Trabajo individual, habrá que adaptar este marco de trabajo a una única persona que conformará íntegramente el Equipo Scrum y realizará el trabajo de los tres Roles existentes: Scrum Master, Product Owner y Desarrollador, a la vez que se realizan los Eventos que el marco de trabajo ofrece: Sprint, Planificación, Daily Scrum, Revisión y Retrospectiva, y mientras se utilizan los Artefactos de los que dispone Scrum: Pila de Producto, Pila de Sprint e Incremento.

- El rol de Desarrollador es el que se compromete a trabajar en cualquier tarea que sirva al Incremento de cada Sprint. Además, es el encargado de seleccionar las Historias de Usuario que se incluirán en el siguiente Sprint, y tendrá que adaptar a diario su trabajo para cumplir el Objetivo del Sprint, siguiendo los estándares de calidad que se definirán en la Definición de Hecho (DoD).
- Como Product Owner se deberá maximizar el valor del producto resultante del trabajo del equipo a lo largo de los diferentes Sprints. Este será el encargado del Product Backlog, donde se encuentran todas las Historias de Usuario que deberán realizarse a lo largo del desarrollo del producto, y tendrá que asegurarse de que el equipo entienda en todo momento el trabajo restante y el objetivo al que se pretende llegar.

- El Scrum Master tiene como objetivo que se utilice el marco de trabajo Scrum según lo define la Guía Scrum, con el fin de que se obtenga el máximo beneficio posible de esta. También ayudará a que el equipo se autogestione y sea multifuncional, resolviendo posibles impedimentos para que el equipo se centre en sus objetivos del Sprint.

Dentro de lo que cabe, integrar las funciones de los tres roles en una sola persona no es complicado y asegura que el equipo será autogestionado, multifuncional y totalmente transparente, pero a la hora de realizar la mayoría de los eventos se van a tener que producir cambios respecto a la definición original de Scrum.

- Los Sprints son un evento de longitud fija donde se realizará todo el trabajo necesario para alcanzar el Objetivo del Producto, que incluyen el resto de eventos que se explican a continuación. Un nuevo Sprint se inicia inmediatamente después de acabar el Sprint anterior, y se considerarán pequeños proyectos en los que no se puede modificar el Objetivo del Sprint que se ha definido.
 - En el caso de este Trabajo, los Sprints tendrán una duración fija de 10 días de trabajo, ya que al tener que ser compaginado con otras asignaturas, prácticas, etc. no se puede asegurar que se pueda realizar un trabajo constante.
- La Planificación del Sprint es el primer evento que se realiza dentro del Sprint, donde el equipo establecerá el trabajo a realizar durante el nuevo Sprint. Aquí se definirá cuál es el Objetivo del Sprint, las Historias de Usuario del Product Backlog que se desarrollarán y el plan para hacerlo, todo esto conocido como Sprint Backlog.
- El Daily Scrum, o Daily, tiene como objetivo visualizar el progreso realizado hacia el Objetivo del Sprint y adaptar el trabajo del día siguiente según sea necesario. Estos eventos se realizarán de forma diaria y no deben durar más de 15 minutos.
 - Dado que el equipo está formado por una única persona, el Daily se realizará visualizando un tablero Kanban en el que se encontrarán las Historias de Usuario del Sprint Backlog y el progreso de cada una de ellas separado en columnas: Pendientes, En progreso, Hechas. El tablero se actualizará a diario, y servirá para visualizar el progreso del Sprint y la siguiente Historia de Usuario a realizar.
- La Revisión del Sprint tiene como fin revisar el resultado final del Sprint, lo que se ha logrado y pensar en qué se trabajará en el siguiente Sprint, además de hacer posibles ajustes en los elementos del Product Backlog que así se decidan.
 - Durante la Revisión del Sprint se dedicará un tiempo a evaluar los resultados del Sprint Backlog en busca de posibles fallos o modificaciones, dar por completas las Historias de Usuario finalizadas y comprobar el trabajo restante del Product Backlog, todo de forma personal pero desde el punto de vista del Product Owner.

- La Retrospectiva se realiza a continuación de la Revisión del Sprint, y son los dos últimos eventos de este. En ella se buscan posibles cambios para mejorar la calidad y la eficacia del trabajo. Para ello se identifica qué fue bien en el Sprint, qué problemas se encontraron y se ofrecen soluciones a estos. Estas soluciones se aplicarán al siguiente Sprint, pudiendo incluso ser agregadas al Sprint Backlog.

Por último, a lo largo del Trabajo se utilizarán los Artefactos de Scrum, diseñados para dar transparencia y poder medir el progreso mediante el compromiso que contienen.

- El Product Backlog es una lista ordenada que contiene todo lo que se necesita para mejorar el producto. Los elementos de esta pueden ser refinados para descomponerlos y definirlos en otros más pequeños y precisos. Cuando uno de estos elementos está suficientemente definido, se considerará listo para añadirlo al Sprint Backlog en una Planificación.
 - El compromiso del Product Backlog es el Objetivo del Producto, que describe un estado futuro del producto desde el que se planificará el trabajo pendiente. Este objetivo es a largo plazo, y debe ser cumplido antes de asumir uno nuevo.
- El Sprint Backlog está compuesto por el Objetivo del Sprint, los elementos del Product Backlog seleccionados durante la Planificación para el Sprint y un plan de entrega. Es la forma de visibilizar el progreso del Sprint, por lo que tiene que estar constantemente actualizado.
 - El compromiso del Sprint Backlog es el Objetivo del Sprint, que es creado en la Planificación y tenido en cuenta por el equipo a lo largo del Sprint como único objetivo a alcanzar.
- Un Incremento es la entrega de uno o más elementos del Sprint Backlog que, añadido a los incrementos anteriores, funciona y es utilizable. Puede crearse más de un Incremento a lo largo de un Sprint, y estos serán presentados en la Revisión del Sprint. No se puede considerar parte de un Incremento un trabajo que no cumpla la Definición de Hecho.
 - El compromiso de los Incrementos es la Definición de Hecho (DoD), que es el conjunto de medidas de calidad requeridas para considerar un elemento de trabajo pendiente parte de un Incremento. En el caso de este Trabajo, la Definición de Hecho será:
 - El nuevo trabajo se encuentra subido al repositorio del proyecto y, llegado el momento, ha sido desplegado.
 - El nuevo trabajo no produce ningún fallo, tanto visual como de lógica de negocio.
 - Todos los Incrementos anteriores funcionan correctamente al incluir el nuevo trabajo.

2.2. Herramientas utilizadas

Durante el proceso de desarrollo de este Trabajo Fin de Grado se han utilizado multitud de aplicaciones, tecnologías, entornos de desarrollo, lenguajes de programación y frameworks. A continuación están listadas todas, junto a los motivos por los cuales se han elegido frente a otras opciones.

1. Como sistema operativo principal para realizar el Trabajo se ha utilizado Windows 10.
2. Aún así, parte del desarrollo fue realizado sobre Ubuntu 20.04 LTS, instalado a través de Windows Subsystem for Linux (WSL), para encapsular algunas de las herramientas necesarias y tener mejor control sobre ellas. Ubuntu fue elegido frente al resto de posibilidades porque es el sistema operativo basado en Linux más utilizado.
3. Para la planificación y otras partes del proceso de desarrollo se utilizó la aplicación web Miro, que fue elegida por su interfaz amigable y sus posibilidades a la hora de crear y mantener pizarras.
4. En el caso de la generación de documentación, así como para realizar esta Memoria, ha sido utilizado Latex, con el fin de aprender su manejo y poder crear de manera simple documentos complejos.
5. La creación de diagramas ha sido realizada mediante la aplicación web diagrams.net, debido a su sencillez y sus múltiples posibilidades.
6. Cuando ha sido necesaria la edición de imágenes se ha utilizado GIMP.
7. Como gestor de bases de datos se ha usado MongoDB Atlas, un servicio cloud que ofrece una base de datos en línea, siempre activa y escalable en caso de que fuera necesario. Se optó por una tecnología de base de datos NoSQL por su mejor rendimiento a la hora de tratar altos volúmenes de datos no estructurados. A su vez, se decidió utilizar un servicio en la nube para poder centrar más el desarrollo en la aplicación y simplificar el despliegue de esta.
8. Respecto a la base de datos, también se utilizó la herramienta MongoDB Compass, que proporciona una interfaz gráfica para visualizar la base de datos, hacer búsquedas y modificar los datos que contiene.
9. Como editor de código se ha utilizado Visual Studio Code, por haberlo utilizado previamente, por su amplia integración con distintos lenguajes de programación y por disponer de un control de versiones intuitivo.
10. El control de versiones se ha hecho mediante Git, debido a que esta tecnología se integra con Visual Studio Code.

11. Todo el proyecto se ha guardado en distintos repositorios de GitHub, haciendo uso de Git, por ser uno de los servicios de alojamiento de proyectos más utilizados y haber sido utilizado anteriormente.
12. La parte backend de la aplicación ha sido desarrollada con el lenguaje Python y el microframework Flask. Se ha elegido este framework debido a que ofrece las herramientas básicas para la creación de una aplicación web y, con esa base, se puedan añadir los componentes o extensiones que sean necesarios, frente a sus competidores como, por ejemplo, Django¹, que da al usuario un conjunto de herramientas comunes para todos y que pueden no satisfacer las necesidades o requisitos de la aplicación.
13. Las partes frontend de la aplicación se desarrollaron con el framework Flutter y el lenguaje Dart. Ha sido elegido por la capacidad de crear aplicaciones compiladas de forma nativa con un único código de base para distintas plataformas como, por ejemplo, Android, iOS y Web.
14. Se ha usado la herramienta Postman para probar los endpoints de la parte backend durante su desarrollo, previamente a ser incluidos en las aplicaciones frontend.
15. También se ha utilizado la tecnología de Docker y la aplicación Docker Desktop para la creación de contenedores que incluyan la aplicación y simplificar la ejecución de esta.
16. Para que la aplicación sea accesible desde internet, se han utilizado dos servicios cloud. La primera Cloud Run, un servicio que da Google Cloud y permite el despliegue de contenedores, elegido frente a otras opciones por su simple integración con GitHub, que ha sido utilizado para desplegar la parte backend de la aplicación.
17. Por último, se ha utilizado GitHub Pages para el despliegue del frontend web de la aplicación ya que, al tener el repositorio creado en GitHub, este servicio viene incluido y se puede activar fácilmente.

2.3. Arquitectura

Antes de comenzar un desarrollo, es necesario tener clara cuál será la arquitectura física de la aplicación, para saber qué componentes van a ser necesarios y las relaciones que ocurrirán entre ellos. Por ello se crea el diagrama que se muestra a continuación.

¹<https://codeahoy.com/compare/django-vs-flask>

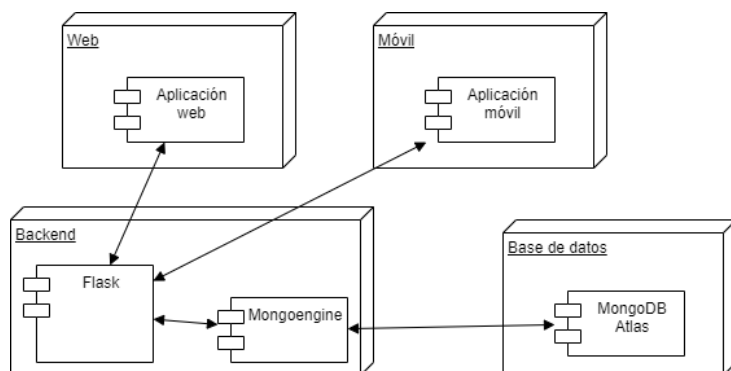


Figura 2.1: Arquitectura física de la plataforma

En el diagrama se muestran los 5 componentes básicos de la aplicación:

- Como base de datos se utilizará el servicio MongoDB Atlas.
- Este será comunicado al backend a través del ODM Mongoengine.
- Flask hará uso de la base de datos a través de Mongoengine.
- Por último, Flask servirá desde el backend la información a ambas aplicaciones, web y móvil.

La definición de la arquitectura física suele venir acompañada por una arquitectura lógica, en la que se representan los componentes lógicos que participan en la aplicación o plataforma.

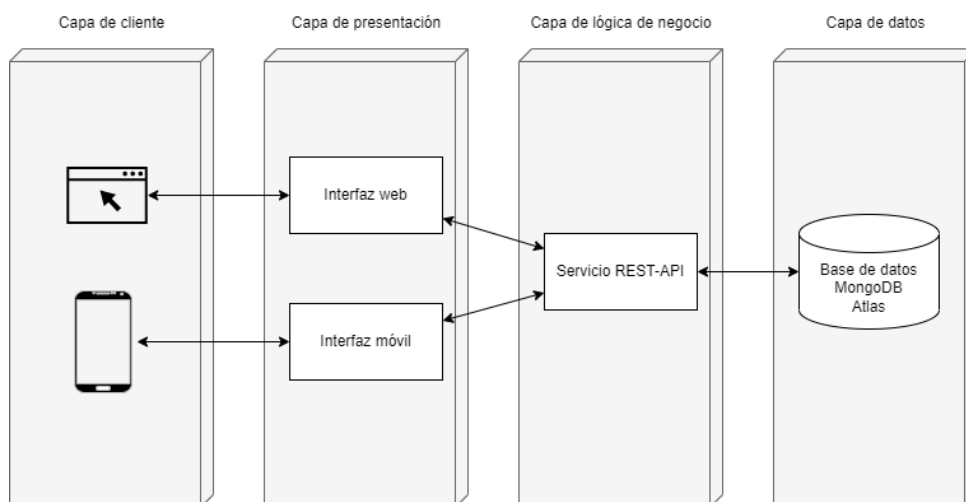


Figura 2.2: Arquitectura lógica de la plataforma

Este diagrama está dividido en las cuatro capas que conforma una plataforma:

- En la capa de datos nos encontramos con la base de datos que se ocupará de la persistencia.
- La capa de lógica de negocio se encarga de trabajar con los datos para aislarlos de las aplicaciones que tendrán los clientes. En este caso, aquí se encontraría el servicio REST-API.
- Con la capa de presentación se presentan los datos tratados por la capa de lógica, y en esta plataforma se encuentra dividida en dos componentes, el que se encarga de la interfaz web para la administración y el que se encarga de la interfaz móvil para los usuarios.
- Por último se encuentra la capa de cliente, que tiene únicamente relación con la capa de presentación para evitar fallos.

2.4. Diseño de interfaz

Durante el desarrollo de la interfaz de usuario de la plataforma se utilizarán distintos componentes incluidos en el kit de desarrollo de Flutter, el framework con el que se implementarán las aplicaciones de administración y usuarios. Estos componentes siguen la línea de Material Design², por lo que serán adaptables a cualquier tipo de pantalla.

Con el objetivo de dar una identidad propia y común a toda la plataforma, se seleccionaron varios colores para crear una paleta de colores que la identifique, de carácter minimalista y basada en un único color primario acompañado de blanco, negro y un tono claro de gris.

²<https://material.io/components?platform=flutter>

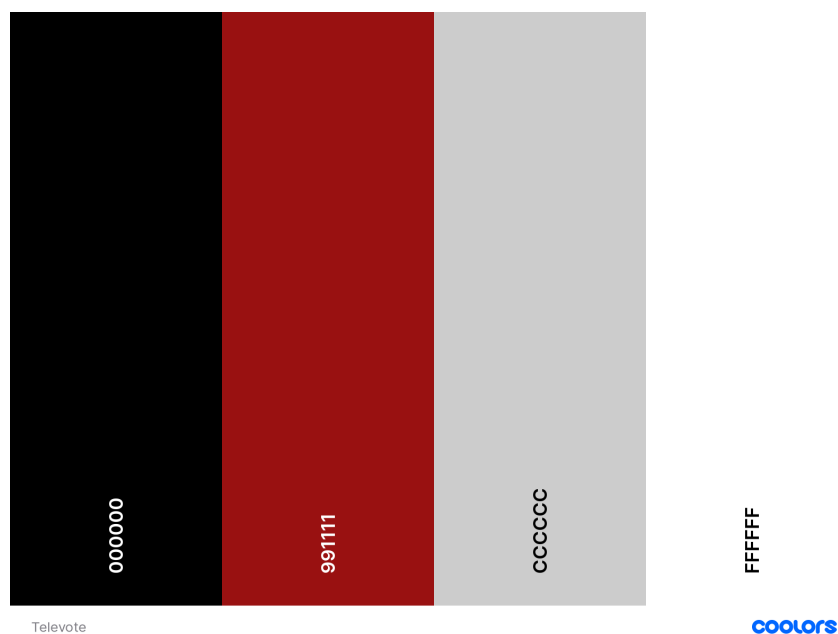


Figura 2.3: Paleta de colores de la plataforma

Siguiendo el color principal de esta paleta, se diseñó un logotipo para que la plataforma pudiera identificarse inequívocamente. Se decidió optar por un diseño minimalista que mezclase dos objetos, una urna y un sobre, que son necesarios para las votaciones más habituales.



Figura 2.4: Logotipo de la plataforma

Capítulo 3

Planificación

En este capítulo se abordarán las cuestiones relativas a la planificación del trabajo, teniendo en cuenta el uso del marco de trabajo Scrum según se ha explicado en la sección 2.1 del Trabajo.

Antes de comenzar ningún Sprint se tuvo que hacer un análisis de lo que se quería realizar en el Trabajo. Este análisis fue plasmado en un enunciado del que, posteriormente, se extrajeron distintas Historias de Usuario, las cuales fueron colocadas en un tablero. Estas Historias de Usuario fueron ordenadas en el Product Backlog según el valor que iban a aportar a la aplicación y adelantando las Historias bloqueantes a las que eran bloqueadas.

Las Historias de Usuario que estaban al principio del Product Backlog se fueron refinando para así poder ser incluidas en el primer Sprint. El resto de Historias se fueron refinando a lo largo de todo el desarrollo, siempre estando preparadas antes de incluirlas en un Sprint.

3.1. Estimación del esfuerzo

Al principio de cada Sprint se debe de hacer una Planificación, en la que se estima el esfuerzo de las Historias de Usuario que aparecen al principio del Product Backlog y se encuentran totalmente refinadas. Esta estimación se puede hacer siguiendo distintas técnicas, pero la utilizada en todos los Sprints ha sido el Planning Poker.

El Planning Poker es una técnica para medir el esfuerzo de una Historia de Usuario o tarea. Lo primero que hay que hacer es leer la Historia o tarea, en voz alta si se hace en grupo, y después cada persona puntúa con un número de la serie de Fibonacci, teniendo en cuenta que cuanto mayor es la puntuación, mayor será el esfuerzo estimado para completar la Historia o tarea. De las posibles puntuaciones se suele retirar el 0 y añadir un 0.5 para las tareas más pequeñas, siendo por tanto las posibilidades: 0.5, 1, 2, 3, 5, 8...

La puntuación obtenida se traducirá en los Puntos de Historia que implica una Historia o tarea. Estos Puntos de Historia nunca se podrán convertir en unidades de tiempo, ya que lo que representan es:

1. La cantidad de trabajo a realizar,
2. La complejidad de la Historia o tarea,
3. Los posibles riesgos o incertidumbres que pueden presentarse.

Una vez estimadas las Historias de Usuario que se encuentran al principio del Product Backlog, se decidirán las que pasarán a formar parte del Sprint Backlog, además de tener que decidir cuál será el Objetivo del Sprint y el plan de entrega. Después de esto, se podrá dar por finalizada la Planificación del Sprint y se podrá iniciar el desarrollo de las diferentes Historias.

El resultado final, después de todo el desarrollo, es un total de 10 Sprints con un número variables de Historias de Usuario que suman 70 Puntos de Historia, 7 en cada Sprint. En el primer Sprint se estimó que se podían realizar 7 Puntos de Historia y se comprobó empíricamente que era la cantidad perfecta para este proyecto y equipo de trabajo. De no haber sido así, se podría haber determinado como problema en una Retrospectiva y haberse solucionado para los siguientes Sprints.

A continuación se detallan las Historias de Usuario que se realizaron en cada Sprint y los Puntos de Historia de cada una de ellas, aunque estas se detallarán más tarde en este documento.

Estimación Sprint 1		
Historia de Usuario		Puntos de Historia
HU-AD-LV1	Listado de votaciones	1
HU-AD-LV3	Nueva votación	3
HU-AD-LV4	Ver votación	1
HU-AD-V1	Información principal de una votación	1
HU-AD-EI2	Página principal	1
Total de Puntos de Historia		7

Cuadro 3.1: Estimación Sprint 1

Estimación Sprint 2		
Historia de Usuario		Puntos de Historia
HU-AD-E1	Ver información de mi entidad	1
HU-AD-E2	Modificar información de mi entidad	2
HU-AD-E4	Crear subentidad	2
HU-AD-E5	Lista de subentidades	1
HU-AD-E6	Bloqueo y desbloqueo subentidad	1
Total de Puntos de Historia		7

Cuadro 3.2: Estimación Sprint 2

Estimación Sprint 3		
Historia de Usuario		Puntos de Historia
HU-AD-CO1	Crear conjunto de opciones	2
HU-AD-CO4	Nueva opción en conjunto de opciones	1
HU-AD-CO5	Editar opción en conjunto de opciones	2
HU-AD-V2	Nueva opción en votación	1
HU-AD-V3	Añadir opción desde conjunto de opciones en votación	1
Total de Puntos de Historia		7

Cuadro 3.3: Estimación Sprint 3

Estimación Sprint 4		
Historia de Usuario		Puntos de Historia
HU-AD-CO2	Renombrar conjunto de opciones	1
HU-AD-CO3	Eliminar conjunto de opciones	1
HU-AD-CO6	Eliminar opción de conjunto de opciones	1
HU-AD-V4	Editar opción de votación	2
HU-AD-V5	Eliminar opción de votación	1
HU-AD-LV5	Iniciar votación	1
Total de Puntos de Historia		7

Cuadro 3.4: Estimación Sprint 4

Estimación Sprint 5		
Historia de Usuario		Puntos de Historia
HU-AD-V6	Generar claves privadas	3
HU-AD-V7	Descarga claves privadas	1
HU-AD-LV6	Finalizar votación	1
HU-AD-E3	Modificar datos inicio de sesión	2
Total de Puntos de Historia		7

Cuadro 3.5: Estimación Sprint 5

Estimación Sprint 6		
Historia de Usuario		Puntos de Historia
HU-AD-V8	Visualizar resultados votación	3
HU-AD-LV7	Eliminar votación	1
HU-AD-S1	Iniciar sesión	2
HU-AD-S2	Cerrar sesión	1
Total de Puntos de Historia		7

Cuadro 3.6: Estimación Sprint 6

Estimación Sprint 7		
Historia de Usuario		Puntos de Historia
HU-AD-S3	Acceso a página privada	0,5
HU-AD-S4	Sesión ya iniciada	0,5
HU-AD-EI1	Menú lateral	1
HU-AD-LV2	Filtros lista votaciones	2
HU-US-L1	Listado votaciones	3
Total de Puntos de Historia		7

Cuadro 3.7: Estimación Sprint 7

Estimación Sprint 8		
Historia de Usuario		Puntos de Historia
HU-US-V1	Pantalla de votación	3
HU-US-V2	Introducir clave privada	2
HU-US-V3	Votación de una opción	2
Total de Puntos de Historia		7

Cuadro 3.8: Estimación Sprint 8

Estimación Sprint 9		
Historia de Usuario		Puntos de Historia
HU-US-EI1	Barra de navegación	1
HU-US-L4	Búsqueda de votación	2
HU-US-L5	Búsqueda de entidad	2
HU-US-V5	Votación de ranking	2
Total de Puntos de Historia		7

Cuadro 3.9: Estimación Sprint 9

Estimación Sprint 10		
Historia de Usuario		Puntos de Historia
HU-US-V6	Votación de suma de valores	2
HU-US-V4	Votación de varias opciones	2
HU-US-L2	Listado de votaciones de una entidad	1
HU-US-L3	Listado de votaciones invisibles de una entidad	0,5
HU-US-EI2	Guía de primera vez	1
HU-US-EI3	Pantalla de información	0,5
Total de Puntos de Historia		7

Cuadro 3.10: Estimación Sprint 10

3.2. Planificación temporal

Como ya se ha comentado anteriormente, los Sprints de este Trabajo tienen una duración fija de 10 días de trabajo. Se ha tomado esta decisión porque, al estar cursando una asignatura y haciendo las prácticas en empresa, no se puede asegurar un trabajo constante a lo largo de un periodo de tiempo cerrado, haciendo posible que un Sprint tenga más o menos días, pero dedicando un número de horas similar a todos.

Pese a no tener límites de días fijados, se fueron apuntando los días de comienzo y fin de cada uno de los Sprints, por lo que se pueden mostrar los periodos de tiempo que se dedicaron a todos ellos. Además, como se comentó al principio de este capítulo, antes de empezar con los Sprints hubo un tiempo de análisis, creación de Historias de Usuario y refinamiento cuyos días también quedaron reflejados.

- Análisis: del 10 de febrero al 18 de febrero.
- Sprint 1: del 19 de febrero al 4 de marzo.
- Sprint 2: del 5 de marzo al 20 de marzo.
- Sprint 3: del 21 de marzo al 4 de abril.
- Sprint 4: del 5 de abril al 20 de abril.
- Sprint 5: del 21 de abril al 3 de mayo.
- Sprint 6: del 4 de mayo al 15 de mayo.
- Sprint 7: del 16 de mayo al 28 de mayo.
- Sprint 8: del 29 de mayo al 7 de junio.
- Sprint 9: del 8 de junio al 17 de junio.
- Sprint 10: del 18 de junio al 27 de junio.



Figura 3.1: Periodos de trabajo de Sprints

3.3. Presupuesto económico

En esta sección se va a tratar de realizar una estimación del presupuesto de una manera similar a como se haría en un proyecto real, aunque con algunos matices, ya que en la realidad este es un Trabajo individual y se han optado por usar herramientas gratuitas y open-source en la mayoría de casos. El presupuesto cubrirá un total de 5 meses, que es el tiempo en el que se desarrolló la plataforma.

3.3.1. Hardware y software

Para realizar el presupuesto de hardware y software se va a suponer que se han utilizado servicios y herramientas de pago con el objetivo de desplegar la plataforma en un entorno real. El software se compondrá de las versiones de pago de las herramientas que se han usado realmente, las cuales se pueden encontrar en la sección 2.2 de esta memoria, y cuyos precios se han obtenido de sus respectivas páginas ^{1 2 3 4 5 6}.

Categoría	Artículo	Precio	Total
Hardware	Equipo de trabajo (incluye SO)	1000€	1000€
Software de gestión	Miro 1 miembro	13,26€/mes 66,30€	66,30€
Servicios cloud	Cluster M10	50,37€/mes	320,62€
	MongoDB Atlas	251,85€	
	GitHub Team	3,54€/mes 17,7€	
	Cloud Run	8,73€/mes	
	Google Cloud	43,65€	
	Simple Email Service AWS	0,10€/mes 0,50€	
	Dominio televote.link	4,42€	
Route 53 (DNS) AWS	0,50€/mes 2,50€		
Total			1386,92€

Cuadro 3.11: Presupuesto de hardware y software

¹<https://miro.com/pricing/>

²<https://www.mongodb.com/pricing>

³<https://github.com/pricing>

⁴<https://cloud.google.com/run/pricing>

⁵<https://aws.amazon.com/es/ses/pricing/>

⁶<https://aws.amazon.com/es/route53/pricing/>

3.3.2. Recursos humanos

El presupuesto estimado en recursos humanos se va a realizar teniendo en cuenta que todo el trabajo lo ha realizado un único desarrollador. Este ha invertido un total de 10 días de trabajo para cada uno de los 10 Sprints, por lo que en total ha trabajado 100 días.

Teniendo en cuenta que el salario medio por día de un desarrollador de software son 65,44€⁷, este es el resultado.

Trabajador	Días	Salario/día	Total
Desarrollador	100 días	65,44€/día	6544€
Total			6544€

Cuadro 3.12: Presupuesto de recursos humanos

3.3.3. Presupuesto total

En la siguiente tabla se muestra la suma de los presupuestos de hardware y software y de recursos humanos calculados anteriormente, resultando en la estimación del presupuesto total.

Presupuesto	Total
Hardware y software	1386,92€
Recursos humanos	6544€
Total	7930,92€

Cuadro 3.13: Presupuesto total

⁷<https://es.indeed.com/career/desarrollador-de-software/salaries>

Parte II

Documentación técnica

Capítulo 4

Análisis

4.1. Requisitos

En esta sección se van a detallar las Historias de Usuario que surgieron del análisis del Trabajo y el continuo refinamiento que tuvieron a lo largo del desarrollo hasta ser implementadas.

Estas Historias fueron agrupadas en distintas Épicas según las similitudes respecto a las clases y elementos que iban a ser desarrollados. A su vez, estas Épicas fueron separadas en dos grandes Iniciativas, cada una con un objetivo concreto pero que servían para alcanzar el objetivo común, la creación de una plataforma de televoto. Estas dos iniciativas son la creación de la aplicación de administración y la creación de la aplicación para usuarios.

4.1.1. Iniciativa I: Creación de la aplicación de administración

Esta primera Iniciativa cuenta con 6 Épicas, las cuales se pueden ver en la siguiente tabla:

Creación de la aplicación de administración		
Épica	Historias de Usuario	Puntos de Historia
1. Lista de votaciones	7	10
2. Votación	8	13
3. Entidad	6	9
4. Conjuntos de opciones	6	8
5. Seguridad	4	4
6. Elementos de la interfaz	2	2
Total	33	46

Cuadro 4.1: Épicas de la Iniciativa I

A continuación se muestran las Historias de Usuario de la Épica 1, Lista de votaciones:

HU-AD-LV1: Listado de votaciones	
Definición	COMO administrador de una entidad QUIERO ver un listado de todas mis votaciones PARA poder visualizarlas y administrarlas
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.2: HU-AD-LV1, Listado de votaciones

HU-AD-LV2: Filtros lista votaciones	
Definición	COMO administrador de una entidad QUIERO filtrar la lista de votaciones PARA encontrar más fácilmente la votación que busco
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes - Se debe poder filtrar por los distintos estados en que se puede encontrar una votación	

Cuadro 4.3: HU-AD-LV2, Filtros lista votaciones

HU-AD-LV3: Nueva votación	
Definición	COMO administrador de una entidad QUIERO crear una nueva votación PARA que los usuarios decidan sobre el tema de esta
Esfuerzo	3 Puntos de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Una entidad bloqueada no puede crear una nueva votación - Puede haber 4 tipos de votación: una opción, varias opciones, ranking y suma de valores - Una votación puede ser pública o privada - Una votación puede ser visible o invisible - Se puede elegir el número de votos por usuario, de 1 a infinitos - El inicio y el fin puede ser manual o fijo en una fecha 	

Cuadro 4.4: HU-AD-LV3, Nueva votación

HU-AD-LV4: Ver votación	
Definición	COMO administrador de una entidad QUIERO ver una votación PARA acceder a una información más detallada de esta
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Se creará una página con formato tabular para añadir más secciones en el futuro - Desde aquí se podrá cambiar la imagen de la votación 	

Cuadro 4.5: HU-AD-LV4, Ver votación

HU-AD-LV5: Iniciar votación	
Definición	COMO administrador de una entidad QUIERO iniciar una votación PARA que los usuarios puedan empezar a votar
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Solo se podrá iniciar una votación si tiene opciones suficientes - Una entidad bloqueada no puede iniciar una votación 	

Cuadro 4.6: HU-AD-LV5, Iniciar votación

HU-AD-LV6: Finalizar votación	
Definición	COMO administrador de una entidad QUIERO finalizar una votación PARA que no se pueda votar más
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Al finalizar una votación, se accederá a una página con su resultado final 	

Cuadro 4.7: HU-AD-LV6, Finalizar votación

HU-AD-LV7: Eliminar votación	
Definición	COMO administrador de una entidad QUIERO eliminar una votación PARA que se borren sus datos cuando ya se hayan obtenido sus datos
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.8: HU-AD-LV7, Eliminar votación

A continuación se muestran las Historias de Usuario de la Épica 2, Votación:

HU-AD-V1: Información principal de una votación	
Definición	COMO administrador de una entidad QUIERO ver la información principal de una votación PARA tomar decisiones sobre esta posteriormente
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.9: HU-AD-V1, Información principal de una votación

HU-AD-V2: Nueva opción en votación	
Definición	COMO administrador de una entidad QUIERO añadir una nueva opción a mi votación PARA completar la votación
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.10: HU-AD-V2, Nueva opción en votación

HU-AD-V3: Nueva opción desde conjunto de opciones en votación	
Definición	COMO administrador de una entidad QUIERO añadir una opción desde un conjunto de opciones a mi votación PARA completar la votación
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.11: HU-AD-V3, Nueva opción desde conjunto de opciones en votación

HU-AD-V4: Editar opción de votación	
Definición	COMO administrador de una entidad QUIERO editar una opción añadida a la votación PARA asegurar que contiene información actualizada
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Para votaciones de suma de valores, se podrá cambiar el valor de la opción 	

Cuadro 4.12: HU-AD-V4, Editar opción de votación

HU-AD-V5: Eliminar opción de votación	
Definición	COMO administrador de una entidad QUIERO eliminar una opción de una votación PARA que deje de estar disponible en esta
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes 	

Cuadro 4.13: HU-AD-V5, Eliminar opción de votación

HU-AD-V6: Generar claves privadas	
Definición	COMO administrador de una entidad QUIERO generar claves privadas PARA controlar los usuarios que pueden votar en mi votación
Esfuerzo	3 Puntos de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes 	

Cuadro 4.14: HU-AD-V6, Generar claves privadas

HU-AD-V7: Descarga claves privadas	
Definición	COMO administrador de una entidad QUIERO descargar en un fichero las claves privadas generadas PARA compartirlas con los usuarios que quiera
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.15: HU-AD-V7, Descarga claves privadas

HU-AD-V8: Visualizar resultados votación	
Definición	COMO administrador de una entidad QUIERO visualizar los resultados de mi votación PARA recibir la opinión de los usuarios
Esfuerzo	3 Puntos de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Si la votación sigue en curso, se mostrarán resultados lo más actualizados posible - Si la votación ha finalizado, se mostrarán resultados consolidados 	

Cuadro 4.16: HU-AD-V8, Visualizar resultados votación

A continuación se muestran las Historias de Usuario de la Épica 3, Entidad:

HU-AD-E1: Ver información de mi entidad	
Definición	COMO administrador de una entidad QUIERO ver la información de mi entidad PARA asegurar que se encuentra actualizada
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.17: HU-AD-E1, Ver información de mi entidad

HU-AD-E2: Modificar información de mi entidad	
Definición	COMO administrador de una entidad QUIERO cambiar la información de mi entidad PARA mantenerla actualizada
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.18: HU-AD-E2, Modificar información de mi entidad

HU-AD-E3: Modificar datos inicio de sesión	
Definición	COMO administrador de una entidad QUIERO modificar los datos de inicio de sesión PARA seguir accediendo a la aplicación si externamente son alterados
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.19: HU-AD-E3, Modificar datos inicio de sesión

HU-AD-E4: Crear subentidad	
Definición	COMO administrador de una entidad QUIERO crear una subentidad PARA delegar la creación de votaciones en un nuevo administrador
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Se mandará un correo al nuevo administrador sobre como entrar en la aplicación 	

Cuadro 4.20: HU-AD-E4, Crear subentidad

HU-AD-E5: Lista de subentidades	
Definición	COMO administrador de una entidad QUIERO ver una lista de mis subentidades PARA controlar los cambios que estas hacen
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes 	

Cuadro 4.21: HU-AD-E5, Lista de subentidades

HU-AD-E6: Bloqueo y desbloqueo subentidad	
Definición	COMO administrador de una entidad QUIERO bloquear y desbloquear una subentidad PARA no permitirle crear o iniciar nuevas votaciones
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes 	

Cuadro 4.22: HU-AD-E6, Bloqueo y desbloqueo subentidad

A continuación se muestran las Historias de Usuario de la Épica 4, Conjuntos de opciones:

HU-AD-CO1: Crear conjunto de opciones	
Definición	COMO administrador de una entidad QUIERO crear un conjunto de opciones PARA guardar opciones relacionadas que sean reutilizables en distintas votaciones
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.23: HU-AD-CO1, Crear conjunto de opciones

HU-AD-CO2: Renombrar conjunto de opciones	
Definición	COMO administrador de una entidad QUIERO renombrar un conjunto de opciones PARA mantenerlo actualizado
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.24: HU-AD-CO2, Renombrar conjunto de opciones

HU-AD-CO3: Eliminar conjunto de opciones	
Definición	COMO administrador de una entidad QUIERO eliminar un conjunto de opciones PARA deshacerme de él cuando no lo vaya a utilizar más
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.25: HU-AD-CO3, Eliminar conjunto de opciones

HU-AD-CO4: Nueva opción en conjunto de opciones	
Definición	COMO administrador de una entidad QUIERO crear una opción en un conjunto de opciones PARA añadir una nueva opción reutilizable
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.26: HU-AD-CO4, Nueva opción en conjunto de opciones

HU-AD-CO5: Editar opción en conjunto de opciones	
Definición	COMO administrador de una entidad QUIERO editar una opción PARA que contenga información actualizada
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.27: HU-AD-CO5, Editar opción en conjunto de opciones

HU-AD-CO6: Eliminar opción de conjunto de opciones	
Definición	COMO administrador de una entidad QUIERO eliminar una opción PARA deshacerme de ella cuando no la vaya a utilizar más
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.28: HU-AD-CO6, Eliminar opción de conjunto de opciones

A continuación se muestran las Historias de Usuario de la Épica 5, Seguridad:

HU-AD-S1: Iniciar sesión	
Definición	COMO usuario sin sesión iniciada QUIERO iniciar sesión PARA entrar a la aplicación como administrador de una entidad
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.29: HU-AD-S1, Iniciar sesión

HU-AD-S2: Cerrar sesión	
Definición	COMO administrador de una entidad QUIERO cerrar sesión de la aplicación PARA finalizar mi trabajo
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.30: HU-AD-S2, Cerrar sesión

HU-AD-S3: Acceso a página privada	
Definición	COMO usuario sin sesión iniciada QUIERO que se me informe cuando intento acceder a una página privada PARA no provocar fallos de seguridad
Esfuerzo	0,5 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	
- Mostrar un botón para redirigir a la página de inicio de sesión	

Cuadro 4.31: HU-AD-S3, Acceso a página privada

HU-AD-S4: Sesión ya iniciada	
Definición	COMO administrador de una entidad QUIERO saber que tengo una sesión iniciada si entro a la página de inicio de sesión PARA no realizar acciones innecesarias
Esfuerzo	0,5 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes - Mostrar un botón para redirigir a la página de inicio	

Cuadro 4.32: HU-AD-S4, Sesión ya iniciada

A continuación se muestran las Historias de Usuario de la Épica 6, Elementos de la interfaz:

HU-AD-EI1: Menú lateral	
Definición	COMO administrador de una entidad QUIERO un menú lateral PARA navegar por las páginas más importantes de la aplicación
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.33: HU-AD-EI1, Menú lateral

HU-AD-EI2: Página principal	
Definición	COMO administrador de una entidad QUIERO una página principal con enlaces a otras páginas y sus funciones PARA acceder a la página que necesite fácilmente
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.34: HU-AD-EI2, Página principal

4.1.2. Iniciativa II: Creación de la aplicación para usuarios

La segunda Iniciativa está compuesta por 3 Épicas, que se pueden visualizar en la tabla que se encuentra a continuación:

Creación de la aplicación para usuarios		
Épica	Historias de Usuario	Puntos de Historia
1. Listados	5	8,5
2. Votación	6	13
3. Elementos de la interfaz	3	2,5
Total	14	24

Cuadro 4.35: Épicas de la Iniciativa II

A continuación se muestran las Historias de Usuario de la Épica 1, Listados:

HU-US-L1: Listado votaciones	
Definición	COMO usuario QUIERO tener un listado de votaciones PARA acceder a la que me interesa
Esfuerzo	3 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes - Al pulsar una votación, se abrirá un modal con información extra - En ese modal aparecerá un botón que lleve a la página de votación. Puede estar deshabilitado - En cada votación debe aparecer la entidad que la ha creado - Si se pulsa sobre la entidad, llevará a la página de la entidad	

Cuadro 4.36: HU-US-L1, Listado votaciones

HU-US-L2: Listado de votaciones de una entidad	
Definición	COMO usuario QUIERO acceder a un listado de votaciones de una entidad PARA navegar entre las votaciones que esta ha creado
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.37: HU-US-L2, Listado de votaciones de una entidad

HU-US-L3: Listado de votaciones invisibles de una entidad	
Definición	COMO usuario QUIERO acceder a un listado de votaciones invisibles de una entidad PARA navegar entre las votaciones que ocultó de las listas principales
Esfuerzo	0,5 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.38: HU-US-L3, Listado de votaciones invisibles de una entidad

HU-US-L4: Búsqueda de votación	
Definición	COMO usuario QUIERO realizar la búsqueda de una votación PARA encontrar fácilmente una votación por su nombre
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.39: HU-US-L4, Búsqueda de votación

HU-US-L5: Búsqueda de entidad	
Definición	COMO usuario QUIERO realizar la búsqueda de una entidad PARA encontrar fácilmente una entidad por su nombre
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.40: HU-US-L5, Búsqueda de entidad

A continuación se muestran las Historias de Usuario de la Épica 2, Votación:

HU-US-V1: Pantalla de votación	
Definición	COMO usuario QUIERO acceder a una pantalla de votación PARA mandar mi voto
Esfuerzo	3 Puntos de Historia
Criterios de aceptación	
<ul style="list-style-type: none"> - Cumplir los criterios de aceptación comunes - Al acceder, comprueba que el usuario no ha votado más veces de las permitidas - La pantalla tiene un tiempo máximo para votar de 5 minutos, si se excede el voto no es válido - Al salir de la pantalla se muestra un modal de voto enviado o inválido 	

Cuadro 4.41: HU-US-V1, Pantalla de votación

HU-US-V2: Introducir clave privada	
Definición	COMO usuario QUIERO introducir mi clave privada PARA acceder a una votación privada
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.42: HU-US-V2, Introducir clave privada

HU-US-V3: Votación de una opción	
Definición	COMO usuario QUIERO votar en una votación de una opción PARA mandar mi voto según el método indicado
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.43: HU-US-V3, Votación de una opción

HU-US-V4: Votación de varias opciones	
Definición	COMO usuario QUIERO votar en una votación de varias opciones PARA mandar mi voto según el método indicado
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.44: HU-US-V4, Votación de varias opciones

HU-US-V5: Votación de ranking	
Definición	COMO usuario QUIERO votar en una votación de ranking PARA mandar mi voto según el método indicado
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.45: HU-US-V5, Votación de ranking

HU-US-V6: Votación de suma de valores	
Definición	COMO usuario QUIERO votar en una votación de suma de valores PARA mandar mi voto según el método indicado
Esfuerzo	2 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.46: HU-US-V6, Votación de suma de valores

A continuación se muestran las Historias de Usuario de la Épica 3, Elementos de la interfaz:

HU-US-EI1: Barra de navegación	
Definición	COMO usuario QUIERO disponer de una barra de navegación PARA navegar por la aplicación fácilmente
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.47: HU-US-EI1, Barra de navegación

HU-US-EI2: Guía de primera vez	
Definición	COMO usuario QUIERO ver una guía la primera vez que abra la aplicación PARA entender su funcionamiento
Esfuerzo	1 Punto de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.48: HU-US-EI2, Guía de primera vez

HU-US-EI3: Pantalla de información	
Definición	COMO usuario QUIERO acceder a una pantalla de información PARA ver información extra de la aplicación
Esfuerzo	0,5 Puntos de Historia
Criterios de aceptación	
- Cumplir los criterios de aceptación comunes	

Cuadro 4.49: HU-US-EI3, Pantalla de información

4.2. Atributos de calidad

Durante el desarrollo se han seguido varios atributos de calidad. En el caso de las Historias de Usuarios, cada una cuenta con distintos criterios de aceptación que se han ido detallando en la sección anterior. Además, todas cuentan con unos criterios de aceptación comunes, que son los siguientes:

1. Todos los elementos de la interfaz han sido creados con una perspectiva responsive, además de pensada para dispositivos móviles.
 - En el caso de la aplicación de administración, el diseño debe funcionar correctamente a partir de los tamaños de pantalla de tabletas inteligentes.
 - En el caso de la aplicación de usuarios, el diseño deberá estar pensado para móvil principalmente.
2. Las funciones que impliquen envío o modificación de datos deben contar con doble validación en cliente y servidor.
3. Los datos sensibles deben ser enviados cifrados, tanto por la parte frontend como por la backend.
4. Todas las llamadas a un endpoint deben tener en cuenta los posibles errores que pueden surgir.

Por otra parte, durante un Sprint se crean uno o más Incrementos que, por definición, deben funcionar al añadirlos a los Incrementos anteriores y cumplir con la Definición de Hecho, que en este Trabajo se compone por las siguientes medidas:

- El nuevo trabajo se encuentra subido al repositorio del proyecto y, llegado el momento, ha sido desplegado.
- El nuevo trabajo no produce ningún fallo, tanto visual como de lógica de negocio.
- Todos los Incrementos anteriores funcionan correctamente al incluir el nuevo trabajo.

Capítulo 5

Diseño

En este capítulo se detallarán los aspectos más importantes del diseño de la plataforma, desde el diseño de datos con el que se ha modelado la base de datos hasta algunos de los diagramas de secuencia de las actividades más importantes que se pueden llevar a cabo.

5.1. Diseño de datos

Como se ha comentado anteriormente, se ha utilizado una base de datos no relacional, MongoDB, para la capa de persistencia de la plataforma. Para la conexión con la base de datos desde el backend se ha hecho uso del ODM ¹ MongoEngine, que permite además la creación de las colecciones que compondrán la aplicación.

Dado que la base de datos está compuesta por estas colecciones de documentos y no por entidades y relaciones como las bases de datos relacionales, para representar el modelo de datos no puede utilizarse un diagrama Entidad-Relación, pero tampoco existe una alternativa consensuada. Lo más habitual es el uso de un diagrama de clases, que se mostrará en la siguiente sección, pero antes se va a hacer un resumen de las distintas colecciones existentes y su significado.

User: En esta colección se guardarán los datos de inicio de sesión de un administrador, así como la entidad que administra.

Entity: Estos documentos incluyen toda la información de una entidad, además de referencias a sus subentidades, conjuntos de opciones y votaciones creadas.

OptionSet: En esta colección se guardará la información de un conjunto de opciones y las opciones que lo completan.

Option: Estos documentos modelan las opciones, y son referenciados tanto por los conjuntos de opciones como por las votaciones.

¹Un ODM, Object Document Mapper, se encarga de traducir entre los documentos de una base de datos no relacional y las clases usadas en el código de una aplicación.

Voting: Esta es la colección que incluirá toda la información de una votación, desde los parámetros que el administrador asigna a la hora de crearla hasta los votos que recibe por parte de los usuarios.

UserVoting: Esta colección sirve para mantener un control para que los usuarios no puedan votar más veces de las debidas en una misma votación, pero no tiene ninguna relación con el voto que el usuario mande.

Vote: En esta colección se modela el voto de un usuario, referenciando la votación a la que pertenece y la opción que el usuario decide. En caso de votaciones a las que se pueden votar múltiples opciones, se crean más de un documento, todos los necesarios.

5.2. Diagrama de clases

Según se ha mencionado en la sección anterior, se ha realizado un diagrama de clases para poder visualizar de forma más sencilla el modelo de la base de datos. Hay que tener en cuenta dos cosas: las clases representan las colecciones y las relaciones entre clases son las referencias que un documento hace a otros.

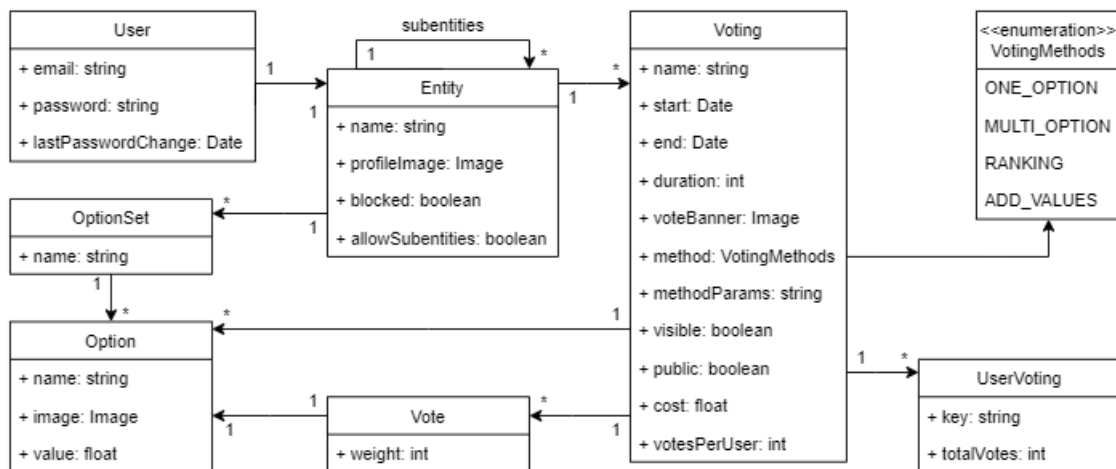


Figura 5.1: Diagrama de clases

5.3. Diagramas de secuencia

Se han considerado 3 actividades como las más importantes a la hora de usar la plataforma, y en esta sección van a ser explicadas mediante diagramas de secuencia. La primera es parte de la aplicación de administración, y consiste en la creación de una nueva votación, el elemento principal de toda la plataforma.

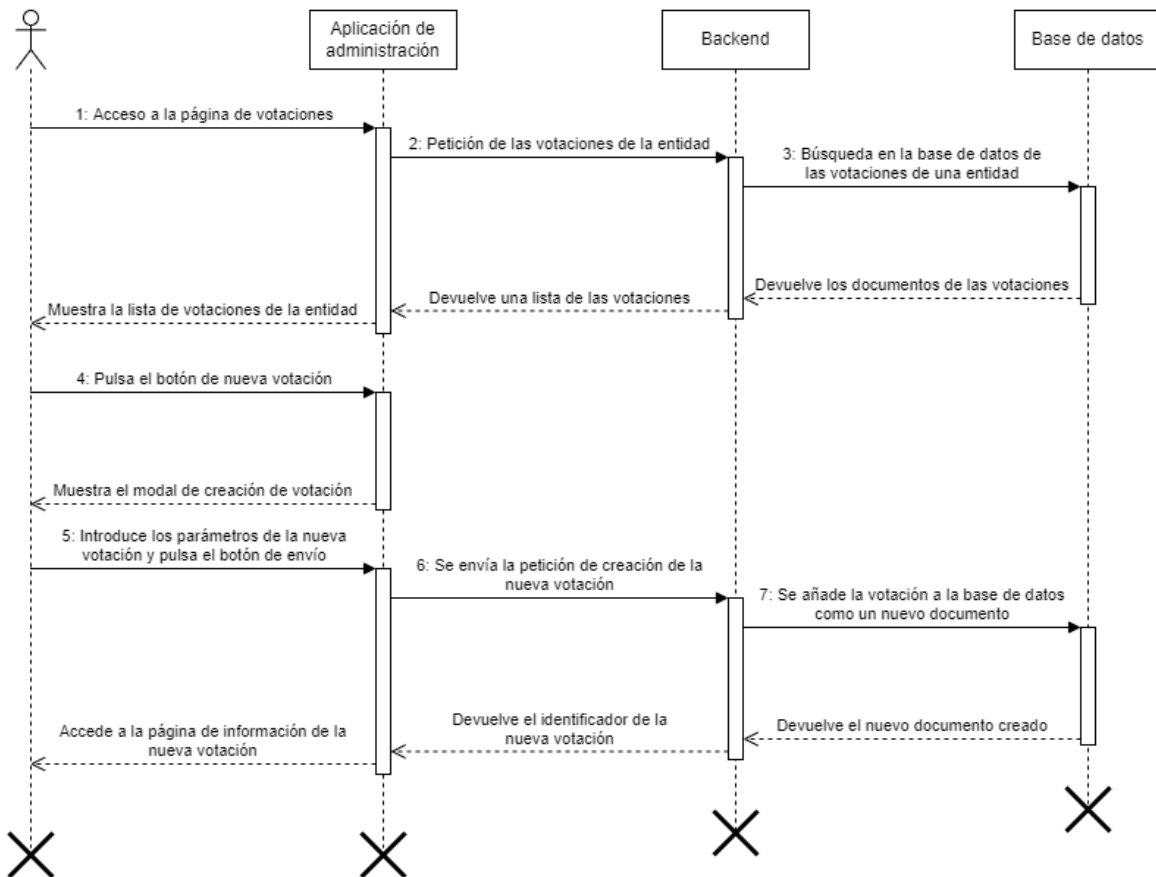


Figura 5.2: Diagrama de secuencia de la creación de una votación

La segunda actividad que se va a mostrar se encuentra también en la aplicación de administración, y es el uso de una opción de un conjunto de opciones en una votación, útil cuando hay varias votaciones que ofrecen las mismas opciones para no tener que repetir su creación en cada votación.

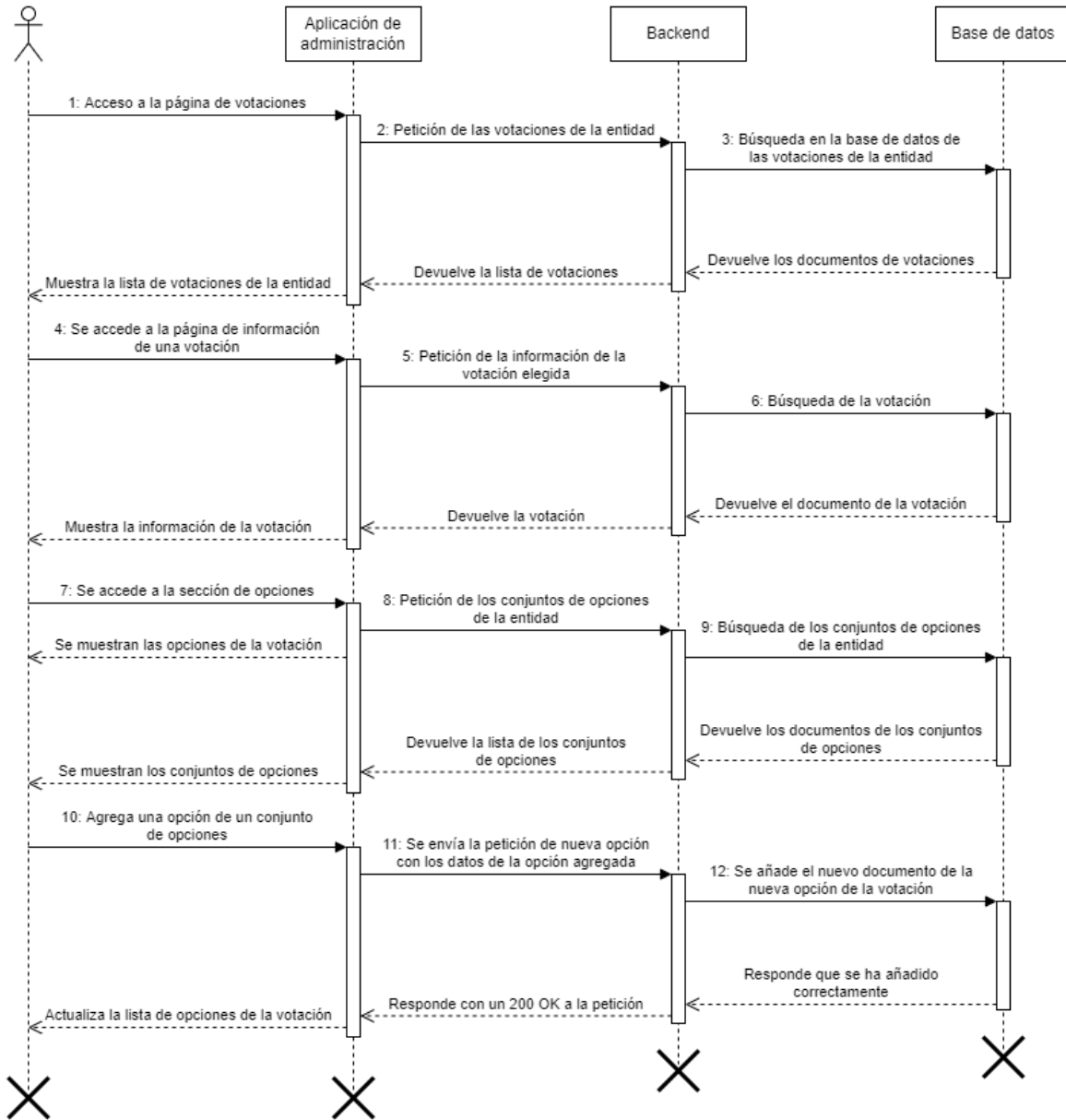


Figura 5.3: Diagrama de secuencia del uso de una opción de un conjunto de opciones en una votación

Por último, se va a mostrar el uso de la aplicación de usuarios para enviar un voto a una de las votaciones creadas por los administradores. Esta es la actividad más importante de la plataforma, y de su correcto funcionamiento depende de la fiabilidad de los datos que después sean recogidos. 6.1

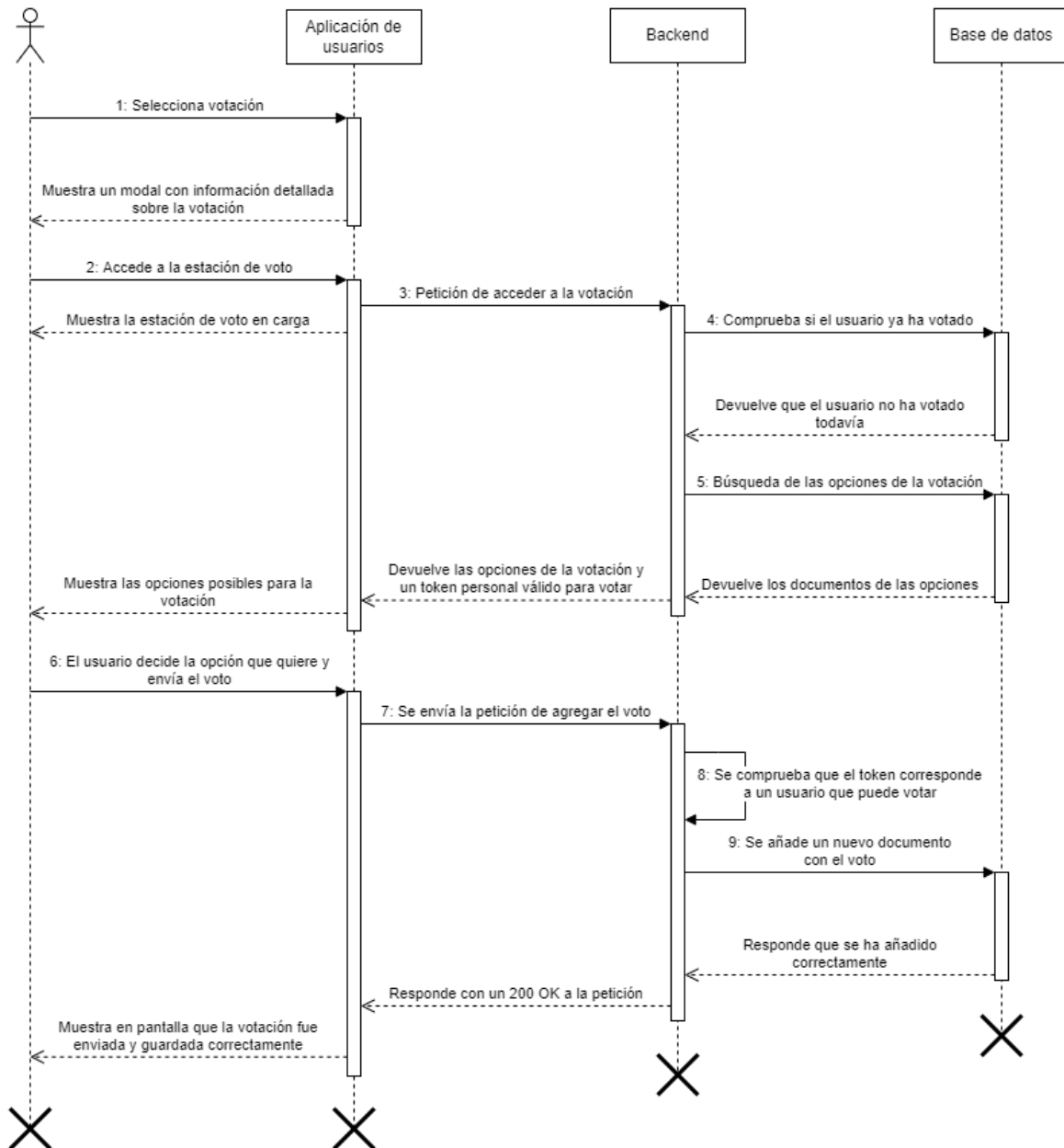


Figura 5.4: Diagrama de secuencia del envío de voto

5.4. Diagramas de estado

Debido al nivel de importancia de las votaciones, se ha realizado un diagrama que muestra los distintos estados en los que se puede encontrar una votación y las acciones que debe hacer un administrador para hacer que la votación pase de uno a otro.

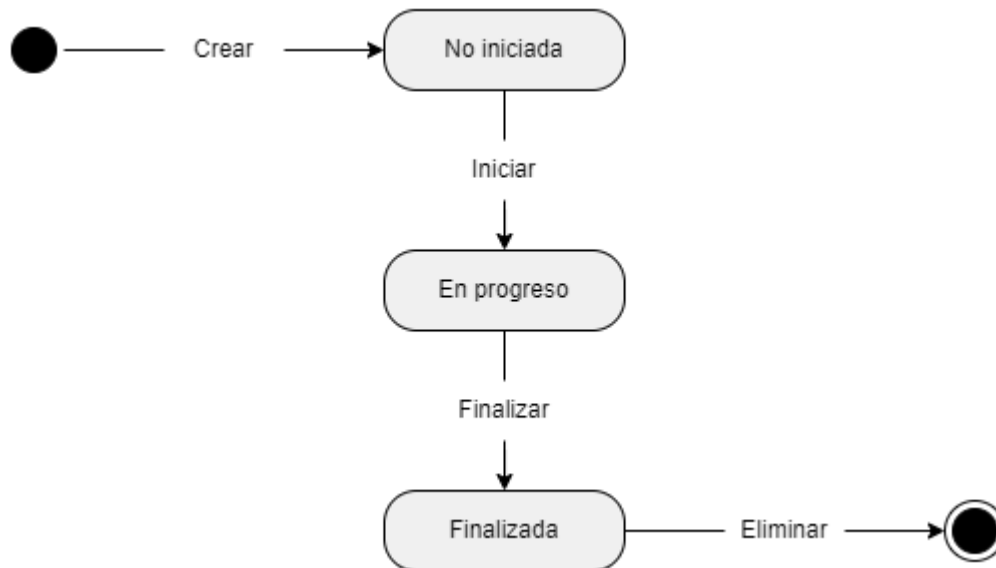


Figura 5.5: Diagrama de estados de una votación

Capítulo 6

Implementación

En este capítulo se va a explicar la estructura del código de la plataforma, además de explicar algunos de los puntos más importantes sobre ella. Hay que tener en cuenta que la plataforma está compuesta por tres aplicaciones distintas:

1. Una aplicación que sirve de backend a las otras dos. Mediante diferentes endpoints, esta aplicación servirá contenido comprobando que el usuario que esté intentando acceder a él tenga los permisos suficientes para ello. Además, tiene la ventaja de desacoplar la base de datos, donde se guarda información sensible, de las aplicaciones que tendrán los usuarios finales.
2. Una aplicación de administración, desde la que se podrán crear las votaciones, entre otras muchas funcionalidades.
3. Una aplicación para los usuarios, con la que podrán mandar sus votos a las votaciones que hayan sido creadas.

Para seguir un orden a la hora de explicar estas aplicaciones, a continuación hay 3 secciones enfocadas en cada una de ellas.

6.1. Implementación de la aplicación backend

La parte backend de la plataforma se ha desarrollado en el lenguaje Python junto al microframework Flask. Este framework ofrece las funcionalidades mínimas para crear una aplicación web, para así poder ampliar después con las funcionalidades específicas que la aplicación necesite, sin ningún añadido de más. Aún así, es común el uso de otros frameworks o librerías que implementen las funcionalidades frecuentes que necesita casi cualquier aplicación.

En el caso de esta aplicación se ha añadido una librería que ayuda a conseguir su objetivo, Flask-RESTful. Esta librería añade las herramientas necesarias para crear de forma más sencilla una API REST. Esta y otras librerías se explicarán según se encuentren.

Estructura de la aplicación

app.py - Este es el fichero que se ejecuta al iniciar la aplicación. Desde aquí se importan la mayoría de librerías que se necesitan en el resto del código y se inicializan las que así lo requieren.

```
1 app = Flask(__name__)      # Inicializacion de la app
2
3 app.config.from_envvar('ENV_FILE_LOCATION')
4 CORS(app, resources={r"/*": {"origins": "*"}})
5
6 api = Api(app)            # Creacion de la API
7 bcrypt = Bcrypt(app)     # Cifrado de datos sensibles
8 jwt = JWTManager(app)    # Gestion de sesiones
9 talisman = Talisman(app) # Conexiones HTTPS
10
11 initialize_db(app)       # Base de datos
12 initialize_mail(app)    # Servicio de correo
13 initialize_routes(api)   # Inicializacion de endpoints
```

/database - En esta carpeta se encuentran los ficheros relacionados con la base de datos.

db.py - Realiza la conexión con la base de datos haciendo uso del Object Document Mapper **flask_mongoengine**.

models.py - Contiene el modelo de datos de la base de datos. Con las clases creadas en este fichero se podrán instanciar objetos con los datos que se encuentran en la base de datos fácilmente. Como ejemplo, a continuación se muestra la creación de la clase **User**, que además contiene un método para cifrar la contraseña del usuario y otro para comprobarla.

```
1 class User(db.Document):
2     email = db.EmailField(required=True, unique=True)
3     password = db.StringField(required=True)
4     lastPasswordChange = db.DateTimeField()
5     entity = db.ReferenceField('Entity')
6
7     def hash_password(self):
8         self.password = generate_password_hash(self.password)
9                             .decode('utf8')
10
11     def check_password(self, password):
12         return check_password_hash(self.password, password)
```

/mail/mail.py - Realiza la conexión con el servicio de correo con la ayuda de la librería **flask_mail**.

/resources - Contiene toda la lógica de la aplicación, dividida según la parte de la aplicación con la que trabajan.

routes.py - Este fichero contiene las rutas de todos los endpoints de la aplicación, y las relaciona con su respectiva lógica de negocio. Por ejemplo, los endpoints de las búsquedas de votaciones y entidades tienen la siguiente forma.

```

1 def initialize_routes(api):
2
3     # ...
4
5     api.add_resource(SearchVotingsApi, '/search/votings')
6     api.add_resource(SearchEntitiesApi, '/search/entities')
7
8     # ...

```

auth.py - Fichero que contiene la funcionalidad de sesiones para la aplicación de administración.

entity.py - Fichero que contiene la funcionalidad de las entidades para la aplicación de administración.

option.py - Fichero que contiene la funcionalidad de los conjuntos de opciones para la aplicación de administración.

vote.py - Fichero que contiene la funcionalidad de las votaciones para la aplicación de administración.

public_voting.py - Fichero que contiene las distintas funcionalidades de la aplicación para usuarios. Entre estas funcionalidades se encuentra la encargada de permitir el voto a los usuarios. Es una parte delicada que además ya fue explicada mediante un diagrama de secuencia de la sección , y ahora se mostrará cómo se realizó en el código.

```

1 class UserVotingOptionsApi(Resource):
2     def post(self, voting_id):
3         voting = Voting.objects.get(id=voting_id)
4         body = request.get_json()
5         key = body['key']
6         if voting.public:
7             try:
8                 userVoting = UserVoting.objects.get(
9                     key=key,
10                    voting=voting)
11            except:
12                userVoting = UserVoting(key=key, voting=voting)
13                userVoting.save()
14                voting.usersVoting.append(userVoting)
15                voting.save()
16        else:
17            try:
18                userVoting = UserVoting.objects.get(
19                    key=key,
20                    voting=voting)
21            except:

```

```

22         return '', 401
23     if(userVoting.totalVotes < voting.votesPerUser or
24        voting.votesPerUser == -1):
25         expires = datetime.timedelta(minutes=5)
26         token = create_access_token(
27             identity=str(key),
28             expires_delta=expires)
29         options = Option.objects(vote=voting).to_json()
30         return {'token': token, 'options': options}, 200
31     else:
32         return '', 418

```

Este primer fragmento de código corresponde a la obtención de las posibles opciones en la votación.

1. En la línea 3 se recupera de la base de datos la votación a la que se está intentando acceder.
2. En la línea 5 se lee del cuerpo de la petición un atributo **key**, que para las votaciones públicas es un identificador único del dispositivo desde el que se vota y para las votaciones privadas es la clave privada que el usuario ha introducido.
3. Si la votación es pública, comprueba si el usuario ha accedido anteriormente en la votación. Si no lo ha hecho, registra un nuevo documento. Este documento no implica que el usuario haya votado, aunque guarda el número de votos que ha emitido.
4. Si la votación es privada, comprueba que la clave privada introducida es válida. Si no lo es, devuelve un error.
5. En las líneas 23 y 24 se comprueba el número de votos que ha emitido el usuario. Si este es superior al permitido, devuelve un error.
6. Si el usuario puede votar, se genera un token con la **key** del usuario, que se devuelve junto con las opciones de la votación.

```

1 class UserVoteApi(Resource):
2     @jwt_required()
3     def post(self, voting_id):
4         voting = Voting.objects.get(id=voting_id)
5         user = get_jwt_identity()
6         userVoting = UserVoting.objects.get(
7             key=user,
8             voting=voting)
9         if(userVoting.totalVotes < voting.votesPerUser or
10            voting.votesPerUser == -1):
11            body = request.get_json()
12            for vote in body:
13                optionId = vote['option']
14                option = Option.objects.get(id=optionId, vote=voting)
15                newVote = Vote(option=option, voting=voting)
16                if vote.get('weight') != '':

```

```
17         newVote.weight = int(vote.get('weight'))
18         newVote.save()
19         voting.votes.append(newVote)
20         voting.save()
21         userVoting.update(totalVotes=userVoting.totalVotes + 1)
22         userVoting.save()
23         return '', 200
24     else:
25         return '', 418
```

Este fragmento corresponde a la recepción del voto del usuario.

1. En la línea 4 se recupera de la base de datos la votación a la que se está intentando votar.
2. En la línea 5 se recupera el token que se mandó junto a las opciones, del cual se puede extraer la **key** del usuario sabiendo que ya ha pasado por la verificación de que puede emitir el voto.
3. Aún así, como se puede ver en las líneas 9 y 10, se vuelve a comprobar que el número de votos emitidos hasta el momento sigue sin ser superior al límite.
4. Después se añade un voto por cada opción que el usuario ha votado. El número de opciones que haya votado el usuario depende del método de votación que la entidad eligió al crear la votación, así como si estas incluyen un valor (`weight`).
5. Finalmente y después de haber agregado todos los votos, se actualizará el número de votos que ha emitido el usuario y se devolverá una respuesta positiva para que el usuario sepa que su voto fue guardado correctamente.

6.2. Implementación de la aplicación de administración

La aplicación de administración está desarrollada con Flutter¹, un framework open source desarrollado por Google, pensado para el desarrollo de aplicaciones multiplataforma con un mismo código de base. En sus inicios estuvo orientado al desarrollo de aplicaciones de dispositivos móviles, pero en sus últimas versiones ofrece también construir aplicaciones web y de escritorio. Esta aplicación está pensada como únicamente web, pero se ha decidido el uso de este framework por ser el mismo con el que se desarrollará la aplicación de usuarios.

¹<https://flutter.dev/>

Estructura de la aplicación

Todo el código desarrollado se encuentra dentro de la carpeta **/lib**, por lo que será desde donde se analizará la estructura.

main.dart - Es el fichero principal de la aplicación, desde donde se lanzará la pantalla de inicio.

/screens - En esta carpeta se encuentran todas las pantallas de la aplicación.

/widgets - Aquí se agrupan los widgets o componentes comunes que han sido creados con el objetivo de reutilizar código y evitar duplicados.

button.dart - Componente botón, parametrizado para poder ser reutilizado en toda la aplicación.

modal.dart - Componente que muestra un modal en pantalla. También cuenta con numerosos parámetros.

/classes - Esta carpeta reúne las clases de la plataforma, comentadas ya en el análisis del modelo de datos, para poder ser instanciadas en la aplicación.

/constants - Aquí se incluyen distintos ficheros con atributos constantes y muy utilizados en la aplicación, como colores o rutas a imágenes.

/utils - En esta carpeta se guardan distintas funcionalidades que son comunes a la aplicación.

session.dart - Permite guardar y obtener el token de sesión, almacenado como una cookie.

theme.dart - Utilizado para crear el tema de la aplicación.

time_left.dart - Devuelve formateada una duración de tiempo.

/api - Esta carpeta incluye los ficheros donde se encuentran las llamadas a la aplicación backend y así poder ser referenciados desde cualquier punto de la aplicación cuando se necesite recibir o enviar información.

6.3. Implementación de la aplicación para usuarios

Al igual que la aplicación de administración, esta también ha sido creada con el framework Flutter, aunque esta vez con el objetivo de ser exportada a dispositivos móviles. Con el mismo código se pueden crear las aplicaciones tanto para Android como para iOS aunque, debido a que el ecosistema de Apple es más restrictivo a la hora de la instalación de aplicaciones, solo se ha tenido en cuenta el desarrollo para Android.

Estructura de la aplicación

Al haber sido desarrollada con el mismo framework que la aplicación de administración, la estructura de la aplicación es la misma. Por ello, solo se van a comentar 2 carpetas.

/widgets - Aquí se agrupan los widgets o componentes comunes que han sido creados con el objetivo de reutilizar código y evitar duplicados.

button.dart - El mismo componente botón que en la otra aplicación. Además de por trabajar sobre el mismo framework, el componente puede ser compartido y utilizado gracias a su parametrización para adaptarlo a las necesidades de cada pantalla.

listed_entity.dart - Componente que muestra una entidad, pensado para usarlo en listas.

listed_voting.dart - Componente que muestra una votación, pensado para usarlo en las distintas listas de votaciones de la aplicación.

detailed_voting.dart - Componente que muestra un modal con la información detallada de una votación. También incluye el botón con el que se puede acceder a la estación de votación.

/utils - En esta carpeta se guardan distintas funcionalidades que son comunes a la aplicación.

theme.dart - Utilizado para crear el tema de la aplicación.

time_left.dart - Devuelve formateada una duración de tiempo.

welcome_items.dart - Contiene la información para crear la guía de inicio de la aplicación.

prettier_double_string.dart - Formatea un número con decimales.

navbarPainter.dart - Crea la forma de la barra de navegación inferior de la aplicación.

Capítulo 7

Pruebas

Una parte muy importante del desarrollo de software son las pruebas. A medida que un proyecto se hace más grande, surgen más puntos posibles de fallo, los cuales hay que mantener controlados para poder solucionar esos errores en el mismo momento que surgen. Por este motivo, cualquier aplicación debería implementar pruebas a distintos niveles para así poder centrarse en el desarrollo sin preocuparse porque una parte deje de funcionar, ya que, en caso de error, este sería alertado rápidamente.

Para este Trabajo se han implementado pruebas en 3 niveles distintos:

1. Pruebas unitarias automatizadas dentro de la aplicación backend.
2. Pruebas de integración automatizadas de la aplicación backend.
3. Pruebas funcionales sobre las aplicaciones de administración y para usuarios.

7.1. Pruebas unitarias

Las pruebas unitarias se realizan a bajo nivel, aislando la funcionalidad que se quiera probar en una función o método que contenga únicamente el código necesario para conseguir su propósito. Con estas pruebas se puede comprobar que el valor que devuelve la función o método probado coincida con el valor esperado.

En el caso de la aplicación backend se ha usado la librería **unittest**, que incluye por defecto Python, para implementar dos pruebas unitarias para dos funcionalidades de la aplicación: la generación de una contraseña segura para una nueva cuenta de administrador y la generación de claves privadas para una votación. Se puede observar como se han implementado ambas pruebas y el resultado de su ejecución en la figura a continuación.

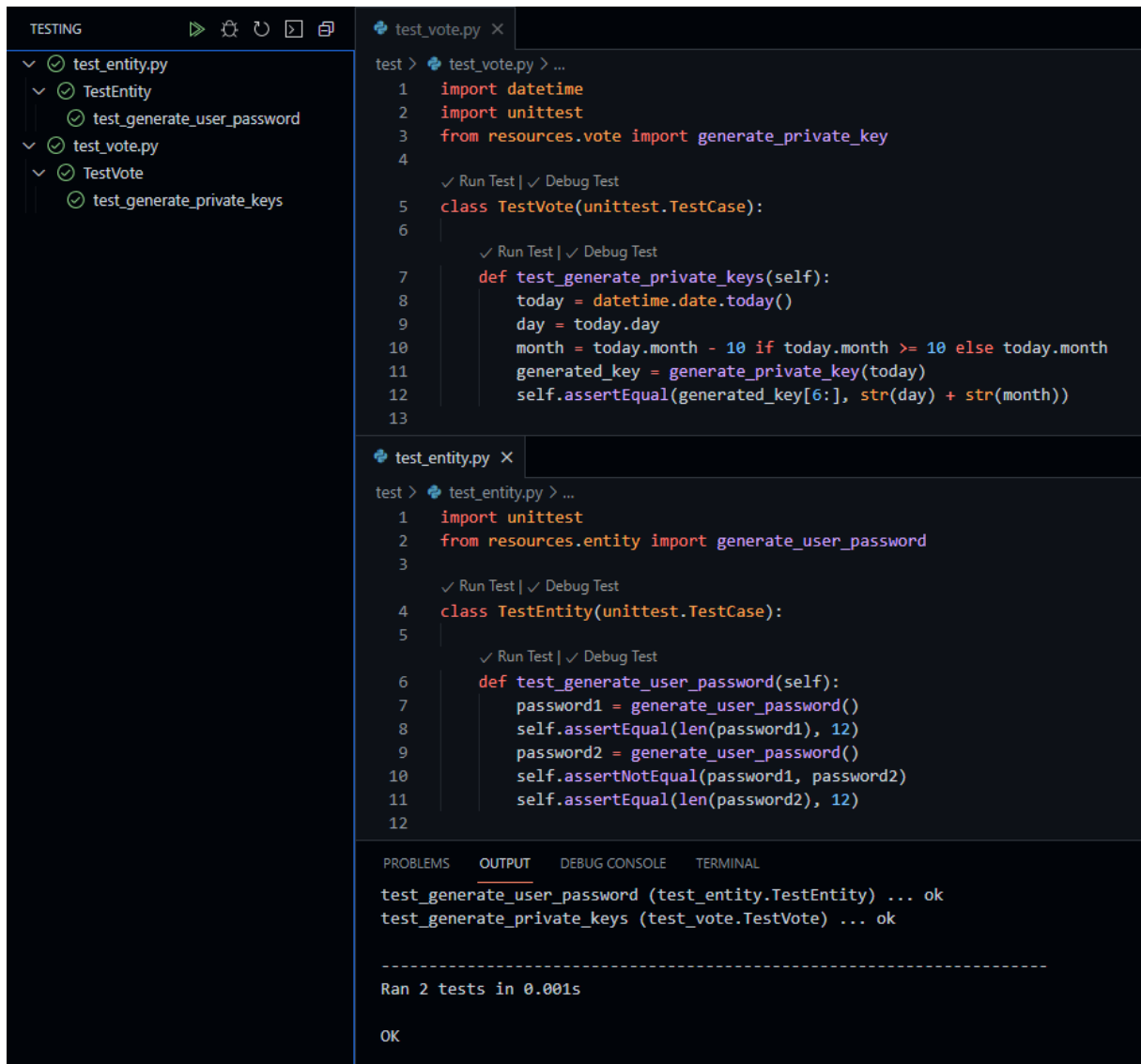


Figura 7.1: Pruebas unitarias

7.2. Pruebas de integración

Las pruebas de integración son realizadas para verificar que, al trabajar con distintos servicios o librerías, el resultado es el deseado.

Para aplicar pruebas de integración en la plataforma se ha hecho uso de la herramienta Postman. Esta es comunmente utilizada para enviar peticiones a una aplicación o API y comprobar que funciona, pero incluye una sección donde se pueden implementar distintas pruebas que comprueben que las respuestas obtenidas de las peticiones enviadas cumplan ciertas reglas. Con estas pruebas se pretende comprobar que todo el funcionamiento del backend es correcto, incluyendo sus múltiples librerías y la conexión con la base de datos.

A continuación se mostrarán dos de estas pruebas. La primera corresponde al inicio de sesión en la aplicación de administración. En la prueba se controla que la respuesta por parte del servidor tenga un estado 200, que significa que todo ha ido bien, y que el cuerpo del mensaje de la respuesta contiene el token de sesión que permitirá al usuario navegar por la aplicación autenticado.



```
1 pm.test("Login: status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test('Login: devuelve el token de sesion', function(){
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.have.property('token');
8 });
```

Body Cookies Headers (5) **Test Results (2/2)**

All Passed Skipped Failed

PASS Login: status code is 200

PASS Login: devuelve el token de sesion

Figura 7.2: Pruebas de integración, inicio de sesión

La siguiente prueba se encarga de verificar que la petición de la entidad del usuario funciona correctamente. Primero comprueba el estado de la respuesta del servidor, que debe ser 200, y posteriormente que el cuerpo del mensaje contenga todos los atributos que tiene un objeto Entidad.

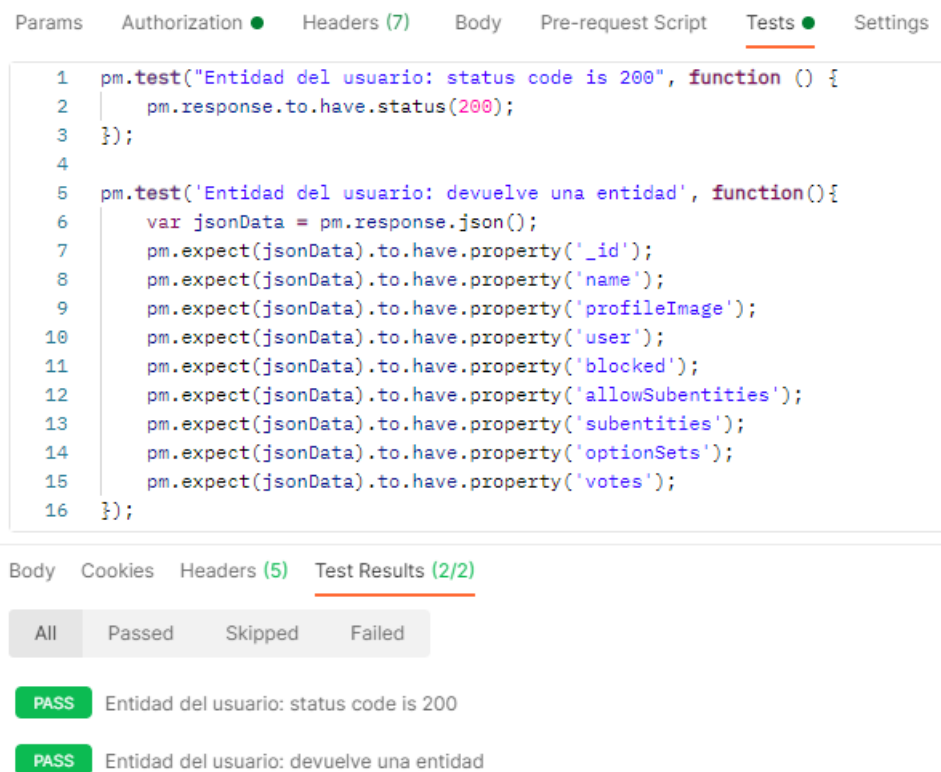


Figura 7.3: Pruebas de integración, entidad del usuario

7.3. Pruebas funcionales

Por último, las pruebas funcionales tienen como objetivo comprobar que una funcionalidad cumple con los criterios de aceptación de la Historia de Usuario que tiene asociada, sin importar el comportamiento interno de la aplicación, solo el que verá el usuario. Estas son las únicas pruebas que se han realizado manualmente, debido al elevado coste de su implementación de manera automatizada.

En esta memoria se van a presentar 3 pruebas funcionales en formato de tabla. La primera de ellas busca crear una nueva votación por parte de una entidad y que todo funcione correctamente.

PF-AD-1: Creación de una nueva votación	
Historia de Usuario asociada	HU-AD-LV3
Objetivo	Se quiere crear una nueva votación, comprobando que los campos incorporan todas las posibilidades indicadas en los criterios de aceptación de la Historia de Usuario
Prerrequisitos	Se tiene que iniciar sesión en la aplicación de administración con el usuario de una entidad que no se encuentre bloqueada
Pasos	<ol style="list-style-type: none"> 1. Acceder al listado de votaciones 2. Pulsar el botón de nueva votación 3. Se introduce el nombre de la votación
Salida esperada	<ul style="list-style-type: none"> - Se puede fijar la fecha de inicio y fin de la votación - Se puede elegir entre los distintos métodos de votación - Se puede elegir entre crear una votación pública o privada - Se puede elegir entre crear una votación visible o invisible - Se puede elegir el número de votos por usuario
Salida obtenida	<ul style="list-style-type: none"> - Se puede fijar la fecha de inicio y fin de la votación - Se puede elegir entre los distintos métodos de votación - Se puede elegir entre crear una votación pública o privada - Se puede elegir entre crear una votación visible o invisible - Se puede elegir el número de votos por usuario

Cuadro 7.1: Prueba funcional, correspondiente a la Historia HU-AD-LV3

La segunda prueba funcional se corresponde al listado de votaciones de la aplicación de usuario.

Por último, la tercera prueba de carácter funcional que se muestra busca comprobar que existe un límite de tiempo en la pantalla de votación de la aplicación para usuarios, y que si se sobrepasa, la votación será correctamente informada como inválida.

PF-US-1: Funcionamiento correcto del listado de votaciones	
Historia de Usuario asociada	HU-US-L1
Objetivo	Se quiere comprobar que el listado de votaciones contiene las funcionalidades que se han fijado como criterios de aceptación de la Historia de Usuario
Prerrequisitos	Tiene que haberse creado mínimo una votación desde la aplicación de administración
Pasos	1. Abrir la aplicación 2. Esperar que se carguen las votaciones creadas
Salida esperada	- En el listado se muestran las votaciones creadas - Se puede acceder a la página de votaciones de una entidad - Al pulsar la votación, se abre un modal con información - En el modal se encuentra un botón que permite votar
Salida obtenida	- En el listado se muestran las votaciones creadas - Se puede acceder a la página de votaciones de una entidad - Al pulsar la votación, se abre un modal con información - En el modal se encuentra un botón que permite votar

Cuadro 7.2: Prueba funcional, correspondiente a la Historia HU-US-L1

PF-US-2: Tiempo límite de votación	
Historia de Usuario asociada	HU-US-V1
Objetivo	Se quiere comprobar que la página de votación dispone de un tiempo límite para votar, y que, si este es sobrepasado, se informe al usuario de que su voto es inválido
Prerrequisitos	Tiene que haberse creado mínimo una votación desde la aplicación de administración
Pasos	1. Abrir la aplicación 2. Esperar que se carguen las votaciones creadas 3. Abrir la información detalla de una votación en progreso 4. Entrar a la estación de votación 5. Esperar más de 5 minutos 6. Mandar el voto
Salida esperada	- Un modal que informe al usuario de que su voto es inválido
Salida obtenida	- Un modal que informe al usuario de que su voto es inválido

Cuadro 7.3: Prueba funcional, correspondiente a la Historia HU-US-V1

Parte III

Manuales de las aplicaciones

Capítulo 8

Manuales de instalación

8.1. Manual de instalación del backend y la aplicación web

En esta sección se explicará cómo arrancar las aplicaciones de backend y web de la plataforma. El único requerimiento es tener instalado Docker¹ en el ordenador.

En el repositorio del proyecto se encuentran las carpetas de cada aplicación y un fichero **docker-compose.yaml**. Si en la carpeta del repositorio descargado se ejecuta el siguiente comando, ambas aplicaciones se iniciarán.

```
1 docker -compose up
```

- *localhost:8080* será la entrada para la aplicación backend, que se puede probar con herramientas como Postman.
- *localhost:80*, o directamente *localhost*, será la entrada para la aplicación web, y se podrá acceder a ella a través de un navegador.

8.2. Manual de instalación de la aplicación móvil

Si se quiere generar la aplicación móvil de forma que conecte a la API desplegada en local, habrá que realizar pasos extra. En este caso será necesario disponer en el ordenador de el SDK de Flutter².

Dentro del proyecto app, habrá que acceder a */lib/constants/api_path.dart* y modificar la primera constante, que se corresponde con la ruta de la API, para que apunte a la IP del ordenador donde esta se esté ejecutando, en el puerto 8080, por ejemplo:

```
1 const String localPath = '192.168.0.16:8080';
```

¹<https://www.docker.com/get-started>

²<https://docs.flutter.dev/get-started/install>

Después habrá que generar la aplicación con el siguiente comando. El archivo APK se podrá encontrar en la ruta *build/app/outputs/apk/release/app-arm64-v8a-release.apk* y podrá ser instalado en cualquier teléfono con sistema operativo Android posterior a la versión 7.

```
1 flutter build apk --split-per-abi
```


Capítulo 9

Manuales de usuarios

9.1. Manual de administración

En esta sección se explicará como usar la plataforma Televote. El primer paso será iniciar sesión. Para poder hacerlo, una entidad que forme parte de la aplicación deberá crear una subentidad con tu correo electrónico. Entonces, se enviará un correo con los datos de inicio de sesión.



Figura 9.1: Inicio de sesión en la aplicación de administración

Al iniciar sesión, se verá una página que resume y enlaza las funcionalidades de la aplicación.

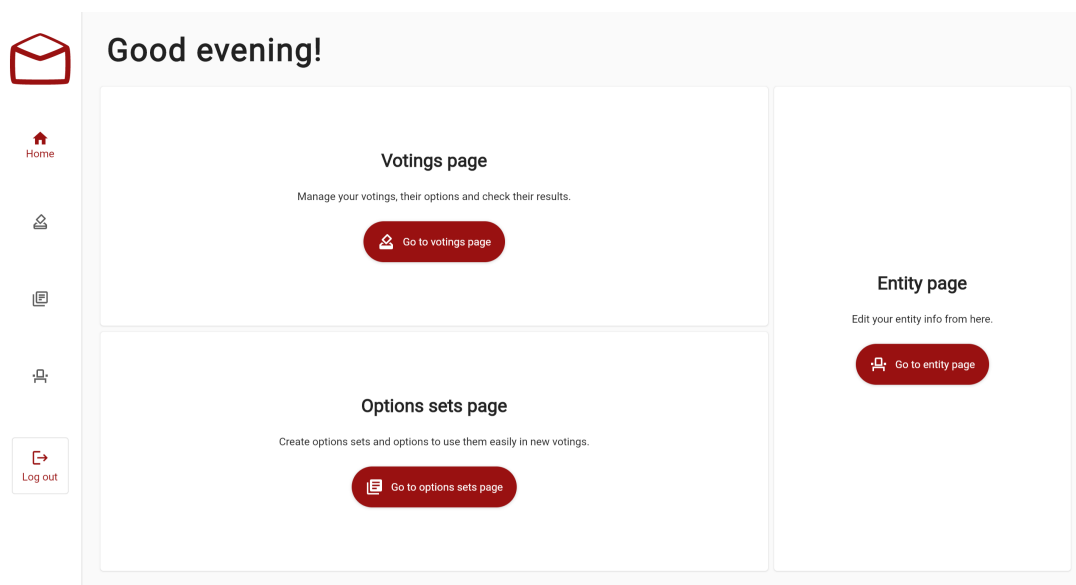


Figura 9.2: Pantalla de inicio de la aplicación de administración

Vamos a comenzar con la gestión de la entidad. En esta pantalla se pueden visualizar los datos de la entidad que manejas, además de las subentidades que hayas creado.

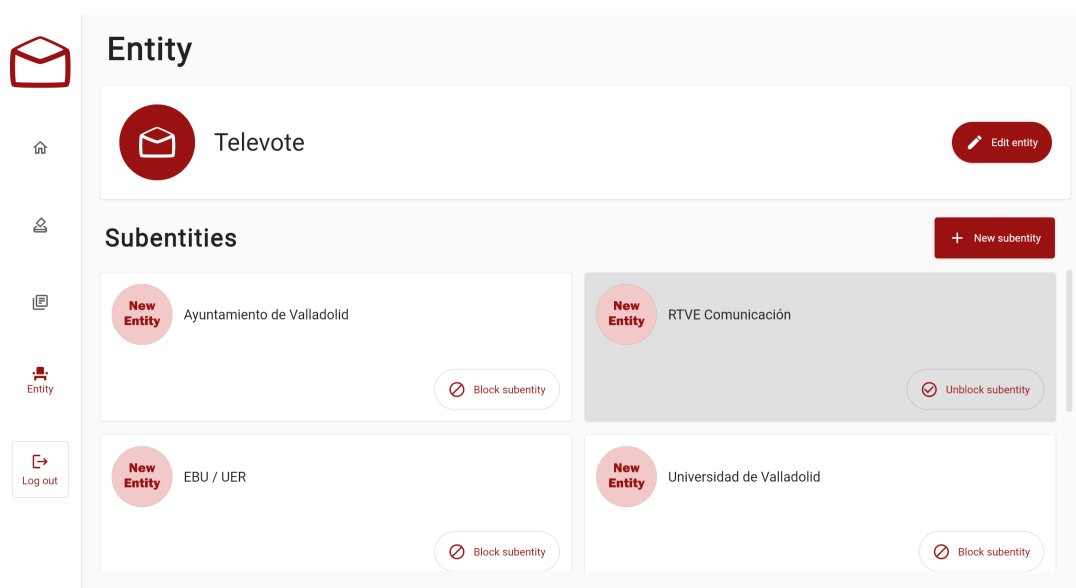
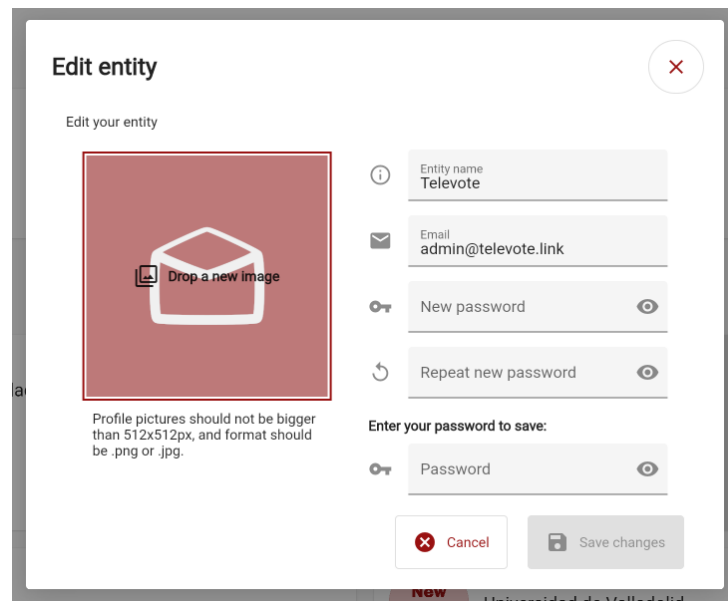


Figura 9.3: Pantalla de entidades de la aplicación de administración

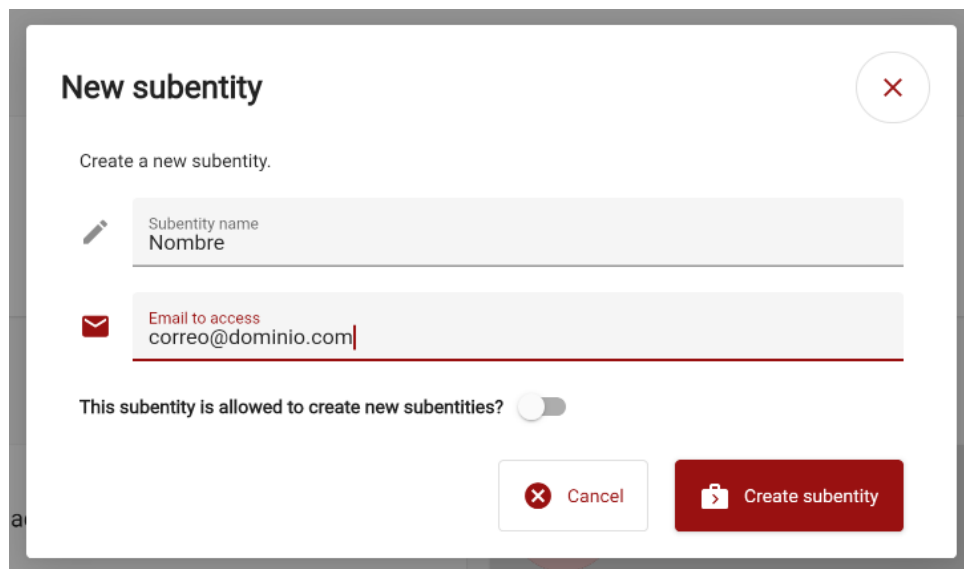
En esta pantalla se puede acceder a la modificación de datos de la entidad, desde donde podrás cambiar la imagen de perfil, nombre y datos de inicio de sesión. Para guardar, siempre hay que introducir la contraseña actual.



The image shows a modal window titled "Edit entity" with a close button (X) in the top right corner. Below the title is the instruction "Edit your entity". On the left side, there is a large red square placeholder for a profile picture with the text "Drop a new image" and a small icon of a document. Below this placeholder, a note states: "Profile pictures should not be bigger than 512x512px, and format should be .png or .jpg." On the right side, there are several input fields: "Entity name" with the value "Televote", "Email" with the value "admin@televote.link", "New password" (with an eye icon to toggle visibility), and "Repeat new password" (also with an eye icon). Below these fields is a section titled "Enter your password to save:" with a "Password" field (with an eye icon). At the bottom of the modal, there are two buttons: "Cancel" (with a red X icon) and "Save changes" (with a floppy disk icon).

Figura 9.4: Modal de edición de entidades

En la pantalla de entidad también se encuentra la función de crear una nueva subentidad. En ella se puede escoger el nombre, el correo del administrador y si esta entidad podrá crear nuevas subentidades.



The image shows a modal window titled "New subentity" with a close button (X) in the top right corner. Below the title is the instruction "Create a new subentity.". There are two input fields: "Subentity name" with the value "Nombre" and "Email to access" with the value "correo@dominio.com". Below these fields is a toggle switch labeled "This subentity is allowed to create new subentities?". At the bottom of the modal, there are two buttons: "Cancel" (with a red X icon) and "Create subentity" (with a floppy disk icon).

Figura 9.5: Modal de creación de subentidades

Por último, cada subentidad listada podrá ser bloqueada o desbloqueada, si es que se decide prescindir de su uso. Para confirmar la acción, se abrirá un nuevo modal.

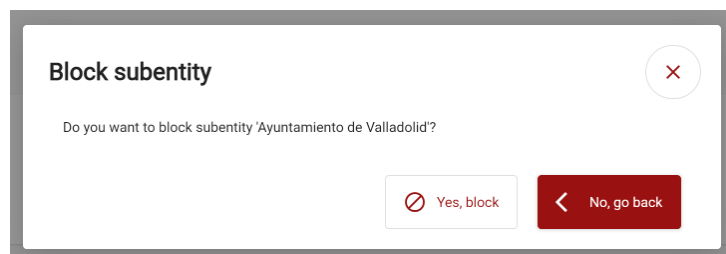


Figura 9.6: Modal de bloqueo de subentidades

Pasamos ahora a la pantalla de conjuntos de opciones. En ella podemos visualizar los conjuntos de opciones reutilizables que hemos creado. Está dividida en dos tarjetas, la izquierda corresponde a la selección de opciones y la derecha se utilizará para su edición.

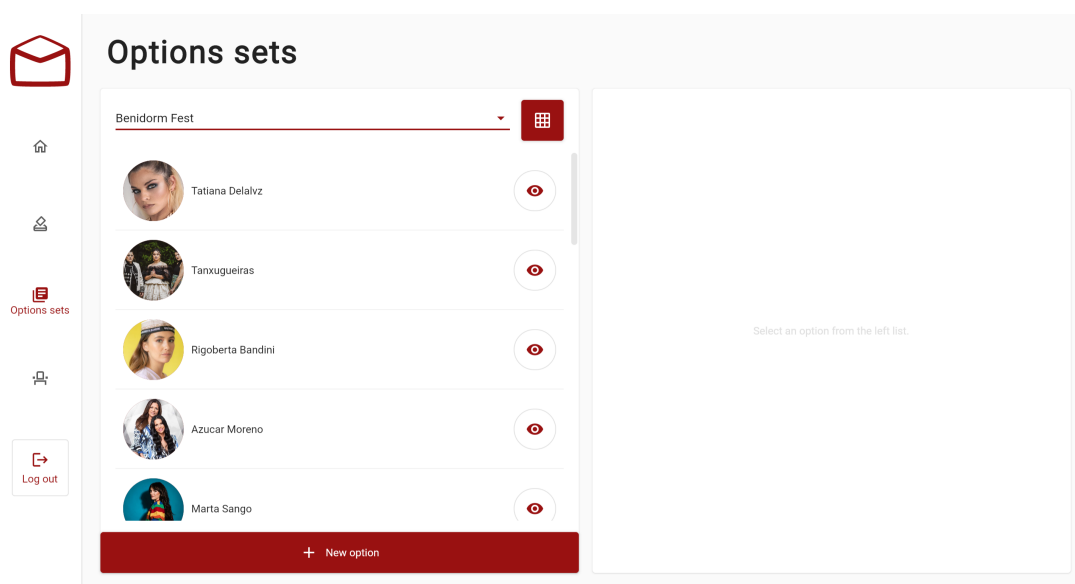


Figura 9.7: Pantalla de conjuntos de opciones

Para poder gestionar los conjuntos de opciones existe un modal, abierto a través del botón de arriba a la derecha de la primera tarjeta. Desde este modal se podrán crear nuevos conjuntos de opciones, además de renombrar o borrar los ya existentes.

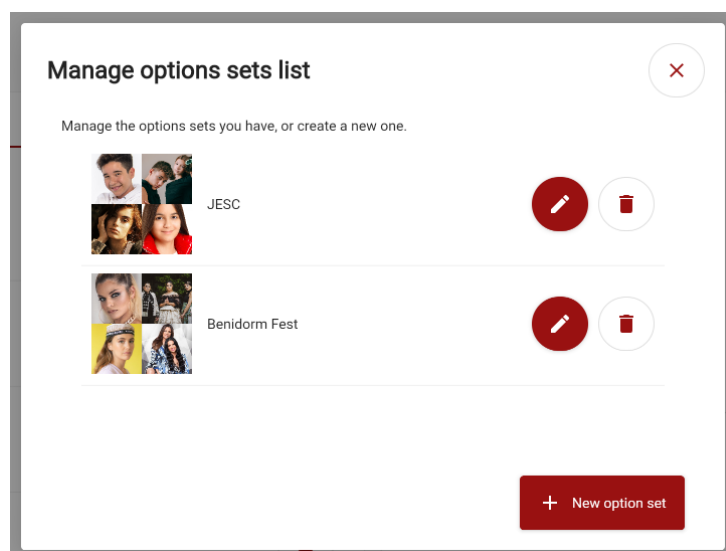


Figura 9.8: Modal de conjuntos de opciones

En la parte inferior de la tarjeta izquierda se encuentra un botón con el que crear nuevas opciones. Al crearlas, se abrirán en la tarjeta de edición.

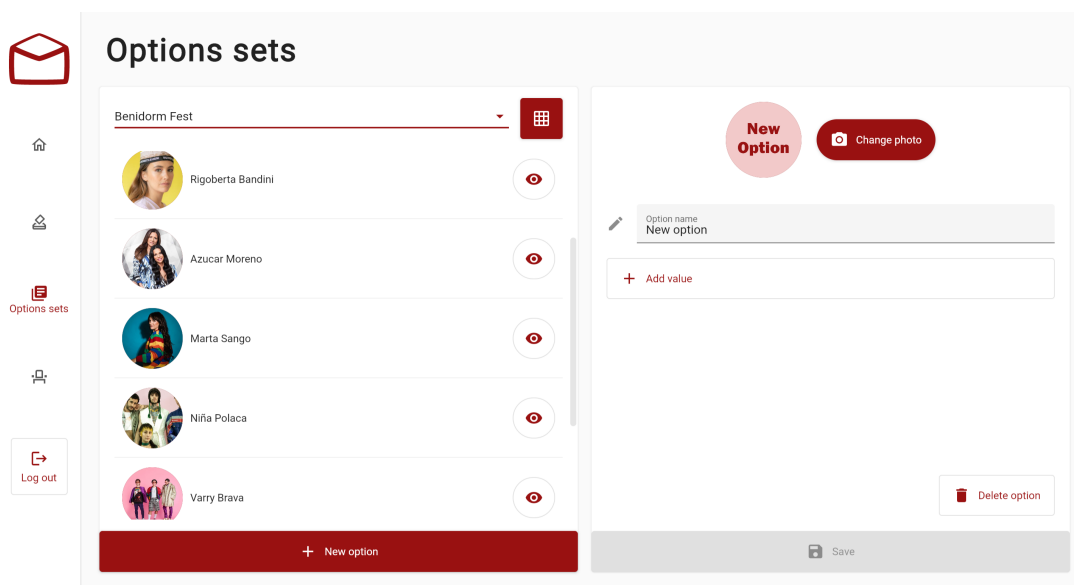


Figura 9.9: Tarjeta de edición de opciones

Para poder cambiar la imagen de una opción existe un modal que, al arrastrar una nueva imagen, la cargará y podrá ser guardada.

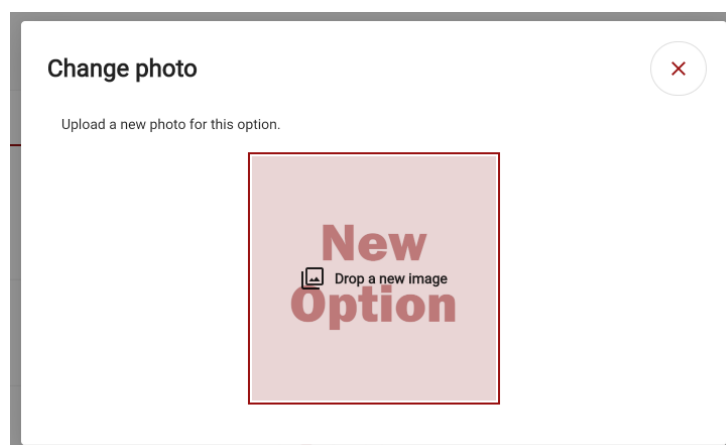


Figura 9.10: Selección de imagen de una opción

Por último, accederemos a las pantallas que tienen que ver con las votaciones. Al abrirla, veremos una lista de todas las votaciones que hemos creado, la cual se puede filtrar entre votaciones controlables, en progreso o finalizadas.

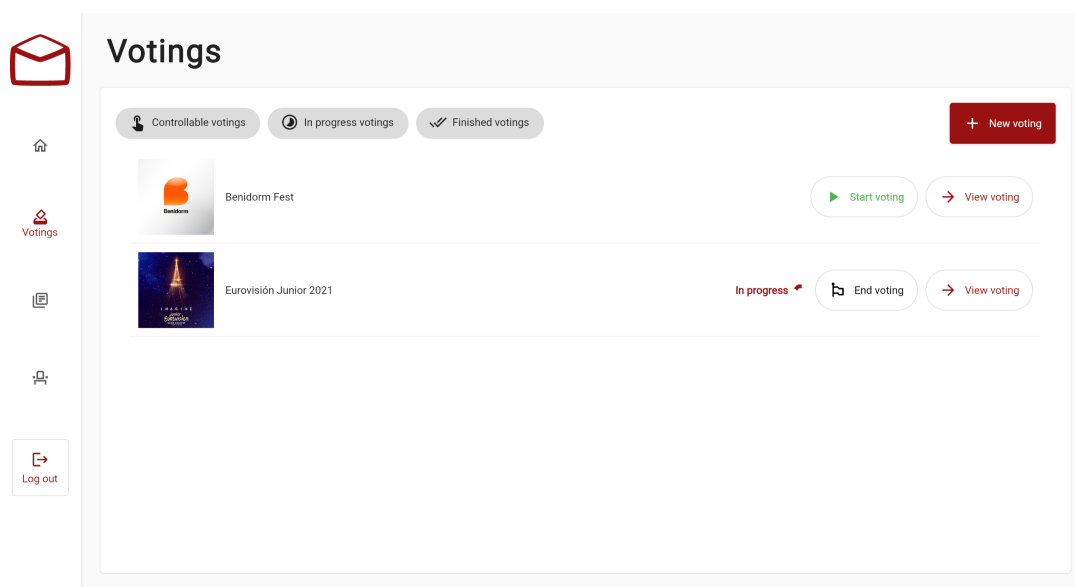


Figura 9.11: Pantalla de votaciones de la aplicación de administración

En esta pantalla se encuentra el modal de creación de nuevas votaciones. En este se puede indicar el nombre, los modos de inicio o fin de la votación, que pueden ser manuales o fijos, y el método de votación entre los 4 disponibles actualmente: una opción, múltiples opciones, ranking y suma de valores.

New voting

Create a new voting.

Voting name
Eurovisión Junior 2021

Start mode:
 Manual Fixed

End mode:
 Manual Duration

Voting method:
Multi-option

Min. options
3

Max. options
3

Cancel Create voting

Figura 9.12: Modal de creación de votaciones (1)

Además, se podrá elegir si una votación es pública y accesible por todo el mundo, o privada y accesible solo por los usuarios a los que se entregue una clave. Por último, se pueden indicar el número de veces que un usuario puede votar, desde 1 hasta infinitas.

New voting

Create a new voting.

Voting method:
Multi-option

Min. options
3

Max. options
3

Can the voting be found in the users application?
 Visible

Can the voting be accessed by any user?
 Public

Votes per user:
 1 Infinite votes per user?

Cancel Create voting

Figura 9.13: Modal de creación de votaciones (2)

Al crear una votación se accederá a una página sobre esta. Tiene varias categorías, según el estado de la votación, aunque siempre se puede encontrar la categoría de información. En ella se muestra el detalle de la votación, al igual que lo verá un usuario antes de votar.



Figura 9.14: Pantalla de votación de la aplicación de administración

En la parte superior izquierda de esta pantalla se encuentra la imagen de la votación, y al pulsar sobre ella se abrirá un modal que permita cambiarla, siempre y cuando la votación todavía no se haya iniciado.

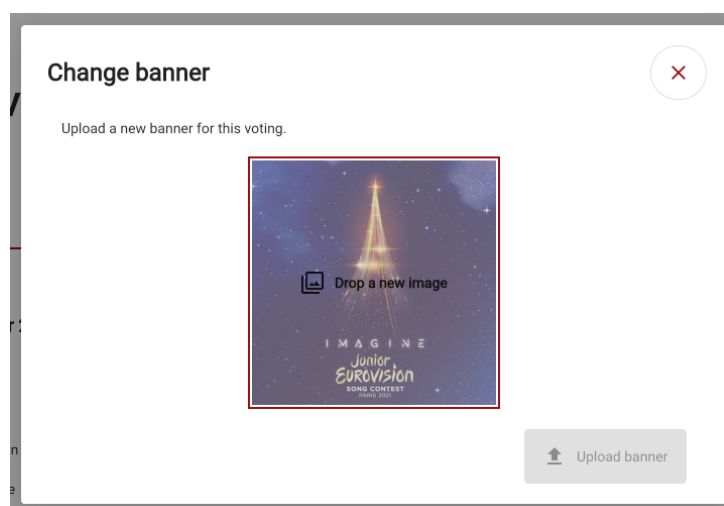


Figura 9.15: Modal de edición de imagen de la votación

La siguiente categoría que aparece en una votación aún sin iniciar es la que permite elegir las opciones que compondrán la votación. Estas pueden ser escogidas de un conjunto de opciones o pueden ser directamente creadas.

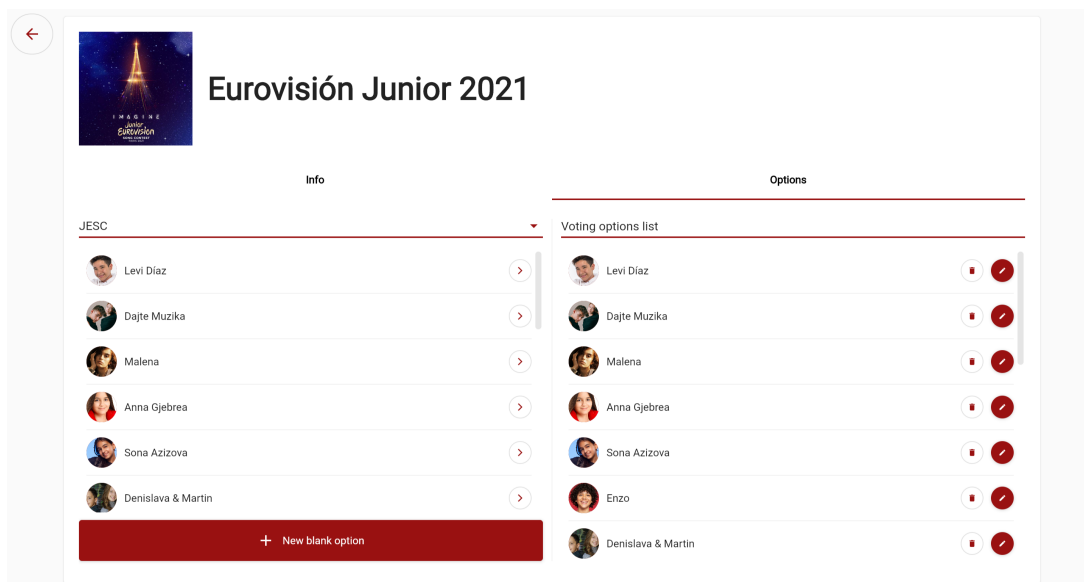


Figura 9.16: Pantalla de selección de opciones para una votación

La última categoría que puede aparecer antes de iniciar una votación es la de generación y descarga de claves privadas. Esta únicamente aparecerá cuando la votación sea privada. Permite generar múltiples claves al mismo tiempo, además de descargar todas las que se han generado en formato CSV.



Figura 9.17: Pantalla de claves privadas de una votación

Por último, nos encontramos con la categoría que muestra los resultados de la votación. Esta categoría aparecerá cuando una votación se haya iniciado, pudiendo hacer un seguimiento de los resultados. Estos resultados serán temporales, y se podrán actualizar según lo desee el administrador. Si la votación ya ha finalizado, los resultados serán finales.

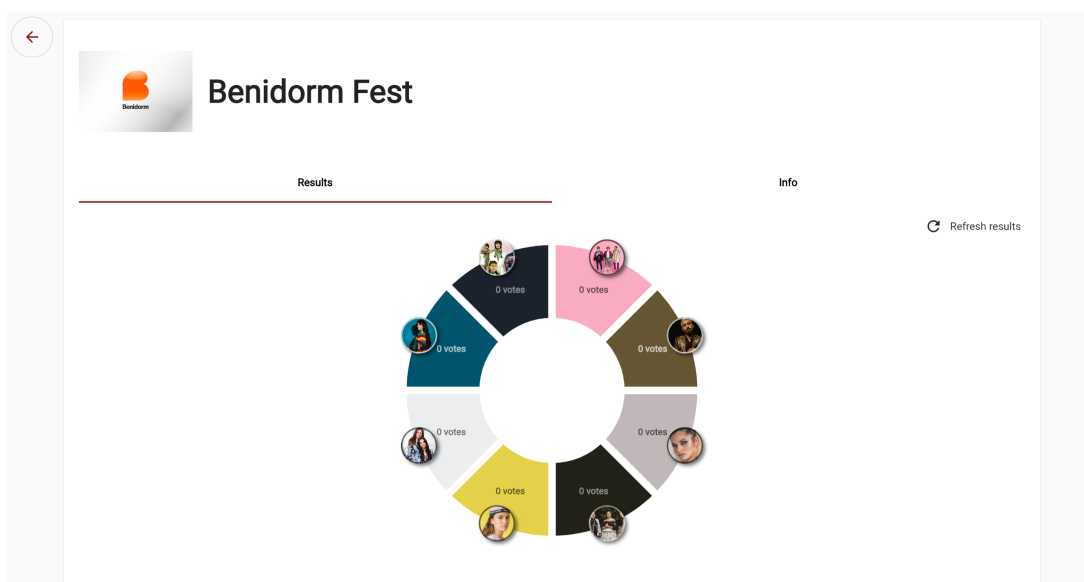


Figura 9.18: Pantalla de resultados de una votación

9.2. Manual de usuario

Al iniciar la aplicación por primera vez, se mostrarán unas pantallas resumen de las funciones que ofrece y como usar la aplicación.

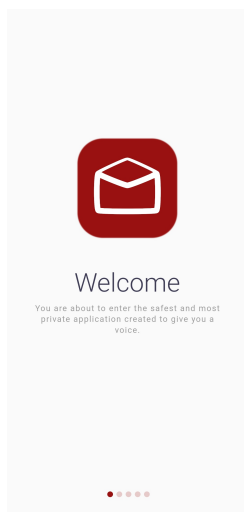


Figura 9.19: Guía de inicio de la aplicación para usuarios

Tras completar la guía de inicio, se abrirá la pantalla principal de la aplicación, que lista tres clases de opciones: las más populares, elegidas por el número de usuarios que han votado en ellas, las que están a punto de acabar y las creadas más recientemente.

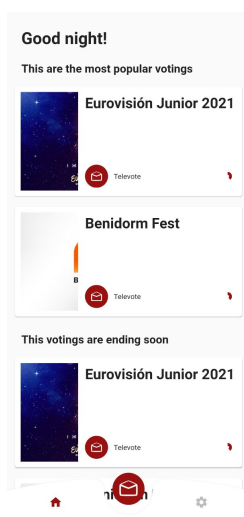


Figura 9.20: Pantalla principal de la aplicación para usuarios

Desde esta lista se podrá acceder a la información detallada de una votación. Esta será mostrada en un modal en la parte inferior de la pantalla, y permitirá al usuario entrar en la estación de votación.

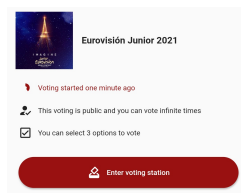


Figura 9.21: Modal de votación de la aplicación para usuarios

En la estación de votación, el usuario podrá emitir su voto siguiendo el método decidido por la entidad que creo la votación.

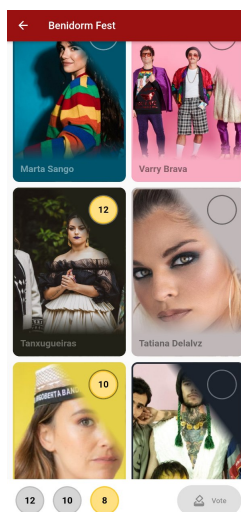


Figura 9.22: Estación de votación

Si lo que se quiere es acceder a la página de una entidad para así visualizar todas las votaciones que ha creado, se podrá pulsar sobre la entidad que creo una votación que esté listada. Desde esta pantalla de entidad también se podrá acceder a las votaciones invisibles de una entidad.

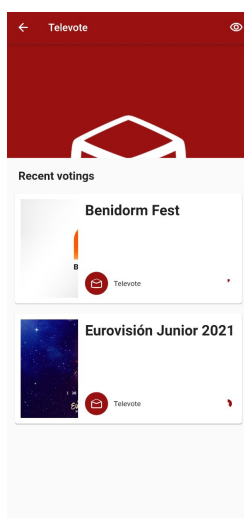


Figura 9.23: Pantalla de entidad de la aplicación para usuarios

Si la entidad a la que se quiere acceder no se encuentra en las listas de la pantalla principal, se podrá acceder a un menú flotante que permite hacer dos tipos de búsqueda.

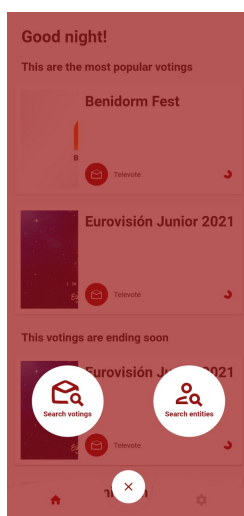


Figura 9.24: Búsqueda en la aplicación para usuarios

Para buscar una entidad, simplemente se tendrá que introducir el nombre y pulsar el botón.

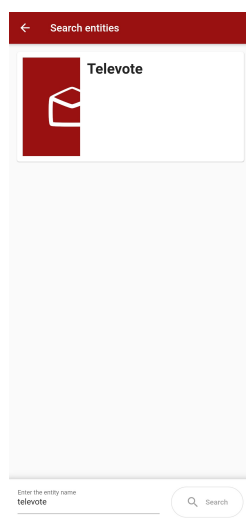


Figura 9.25: Búsqueda de una entidad en la aplicación para usuarios

También se podrán realizar búsquedas de votaciones que no esten listadas en la pantalla principal. Para ello, habrá que introducir el nombre de la votación que se desea buscar y pulsar el botón.

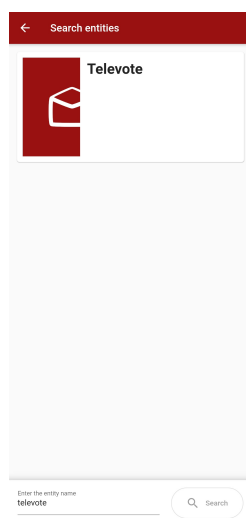


Figura 9.26: Búsqueda de una votación en la aplicación para usuarios

Por último, en la barra de navegación inferior nos encontramos con una pestaña de información. Desde esta se podrá conocer el tratamiento que la aplicación hace con los datos de los usuarios, además de poder volver a ver la guía inicial.

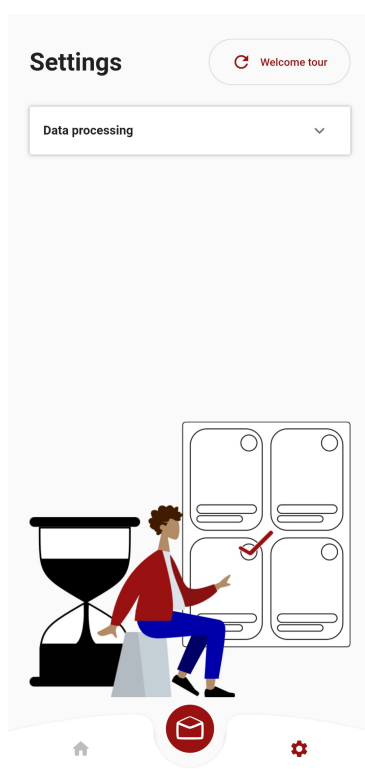


Figura 9.27: Pantalla de información de la aplicación para usuarios

Parte IV
Conclusiones

Capítulo 10

Trabajo futuro y conclusiones

10.1. Trabajo futuro

Gracias a la adaptabilidad del marco de trabajo que se ha usado para el proceso de desarrollo de la plataforma, sería posible y muy sencillo continuar implementando nuevos incrementos. Bastaría con analizar nuevas funcionalidades, escribir Historias de Usuario y refinarlas para así agregarlas al Product Backlog y comenzar un nuevo Sprint. Algunas de las posibles mejoras que se podrían desarrollar en la plataforma son:

- **Votaciones de pago.** Es una práctica común en los servicios de entretenimiento, y se podría implementar de múltiples maneras, siempre y cuando se siga respetando el anonimato de los usuarios.
- **Votaciones geobloqueadas.** Puede que por parte de una entidad exista la necesidad de crear distintas votaciones para distintas áreas geográficas.
- **Más opciones de interfaz** que ayuden al usuario a visualizar los datos de la votación, como, por ejemplo, descripciones de las opciones, un temporizador con el tiempo restante dentro de la propia estación de votación. . .
- **Nuevos métodos de votación** que permitan a nuevas entidades usar la plataforma y a las entidades que ya la utilizasen crear votaciones mejor adaptadas a sus necesidades.
- **Internacionalización.** Permitir visualizar la aplicación en diferentes idiomas es algo necesario en aplicaciones reales.

10.2. Conclusiones personales

Teniendo en cuenta los objetivos que fijé al inicio del Trabajo, me parece que todos ellos han sido logrados, incluido el más importante, agrupar todos los conocimientos adquiridos a lo largo de los estudios de Grado. Y es que este Trabajo ha supuesto un esfuerzo mucho mayor que ningún otro proyecto o práctica.

Creo que he conseguido incluir, si no todos, la mayoría de fundamentos que se enseñan en las distintas asignaturas del Grado, y usarlos como base para entender las nuevas tecnologías con las que he desarrollado toda la plataforma.

Además, este Trabajo ha implicado muchas más cosas que escribir líneas de código. Gracias al marco de trabajo ágil que he seguido, he conseguido analizar todos los detalles que quería incluir en la plataforma y diseñar una colección de diagramas con los que visualizar cada paso a seguir en la posterior implementación, y, más tarde, comprobar que todos los nuevos añadidos encajaban con el trabajo que ya había realizado e iba en buena dirección.

En definitiva, creo que supone un antes y un después en mi forma de trabajar, aprender e investigar, al igual que lo supuso empezar a estudiar en la universidad. Pero de eso trata todo, de mejorar paso a paso hasta conseguir ser la persona que quieres.

Bibliografía

- [1] María Fernanda Aguirre. “Realiza estimaciones ágiles y precisas gracias al Planning Poker”. En: *Appvizer* (18 Noviembre 2020). URL: <https://www.appvizer.es/revista/organizacion-planificacion/gestion-proyectos/planning-poker>.
- [2] *Los diferentes tipos de testing en el desarrollo de software*. Programación y más. URL: <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>.
- [3] R. Pressman. *Ingeniería del Software*. McGraw-Hill, 2014.
- [4] Max Rehkopf. *Stories, epics, and initiatives*. Atlassian Agile Coach. URL: <https://www.atlassian.com/agile/project-management/epics-stories-themes>.
- [5] Ken Schwaber y Jeff Sutherland. *La Guía Scrum*. Noviembre 2020. URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>.
- [6] Juliana Tyler. *What is the difference between ODM and ORM?* Medium. URL: <https://medium.com/@julianam.tyler/what-is-the-difference-between-odm-and-orm-267bbb7778b0>.