

# Aplicación Android para la gestión de proyectos mediante la metodología Scrum



Grado en Ingeniería Informática

## Trabajo Fin de Grado

Autor:

Sergio Julio García Urdiales

Tutor/es:

Antonio Javier Gallego Sánchez



Universitat d'Alacant  
Universidad de Alicante

Junio 2018

*Quiero dedicar este trabajo a José Carbonell por toda la ayuda que me ha prestado durante estos años. También se lo dedico a Virgi por confiar siempre en mí. Pero especialmente, se lo dedico a mi madre, mi padre y mi hermano, por el apoyo incondicional y el esfuerzo durante estos años.*

*Gracias.*

# Índice

1. Justificación y Objetivos.....	10
2. Recursos .....	11
3. Introducción.....	11
3.1. Historia.....	11
3.2. La idea .....	13
3.3. Objetivos .....	14
4. Estado del arte .....	15
4.1. Trello .....	17
4.2. Jira.....	19
4.3. Conclusiones del mercado.....	20
5. Metodología.....	20
5.1. <i>Sprint</i> 1.....	22
5.2. <i>Sprint</i> 2.....	22
5.3. <i>Sprint</i> 3.....	23
5.4. <i>Sprint</i> 4.....	23
5.5. <i>Sprint</i> 5.....	23
5.6. <i>Sprint</i> 6.....	24
5.7. <i>Sprint</i> 7.....	24
5.8. <i>Sprint</i> 8.....	24
6. Diseño.....	24
6.1. Prototipo temprano.....	25
6.2. Diseño final.....	26
7. Tecnologías .....	35
7.1 Gestión de proyectos y documentación.....	35

7.2	API y BBD .....	36
7.3	APP .....	37
8.	Arquitectura.....	38
9.	Implementación.....	39
9.1.	API.....	39
9.1.1.	Proyecto .....	43
9.1.2.	Sprints .....	43
9.1.3.	Estados .....	44
9.1.4.	Tareas.....	44
9.1.5.	Votaciones .....	45
9.2.	APP.....	45
9.2.1.	Interfaces sencillas.....	46
9.2.2.	Interfaces medias.....	48
9.2.3.	Interfaces complejas .....	49
9.2.4.	Otros elementos .....	52
9.2.5.	Diseños adaptativos .....	52
10.	Conclusiones.....	52
11.	Agradecimientos .....	54
12.	Bibliografía.....	55
13.	Anexos.....	56
13.1.	Endpoints API .....	56
13.3.1.	User.....	56
13.3.2.	Permission .....	61
13.3.3.	Role .....	64
13.3.4.	Project.....	68
13.3.5.	Sprint.....	74
13.3.6.	Status .....	79
13.3.7.	Task .....	83
13.3.8.	Change .....	89
13.3.9.	Poll .....	92
13.3.10.	Vote.....	96

13.2.	Modelo de datos.....	102
13.3.	Diagramas de caso de uso.....	103
13.3.1.	Rol.....	103
13.3.2.	Permiso.....	103
13.3.3.	Usuario.....	104
13.3.4.	Proyecto.....	104
13.3.5.	Sprint.....	105
13.3.6.	Estado.....	105
13.3.7.	Tarea.....	106
13.3.8.	Votación.....	106
13.3.9.	Voto.....	107
13.3.10.	Cambio.....	107

## Índice de Tablas

Tabla 1. Mockups de la aplicación móvil .....	26
Tabla 2. Registro Usuario .....	57
Tabla 3. Login Usuario.....	57
Tabla 4. Obtener usuario. ....	58
Tabla 5. Actualizar Usuario.....	59
Tabla 6. Borrar Usuario.....	60
Tabla 7. Buscar usuario.....	61
Tabla 8. Crear Permiso.....	62
Tabla 9. Consultar Permisos.....	63
Tabla 10. Obtener permiso por nombre.....	64
Tabla 11. Borrar permiso .....	64
Tabla 12. Crear Rol.....	66
Tabla 13. Obtener todos los servicios .....	67
Tabla 14. Obtener rol concreto .....	68
Tabla 15. Borrar rol.....	68
Tabla 16. Crear proyecto.....	69
Tabla 17. Obtener proyectos de usuario.....	71
Tabla 18. Consultar proyecto concreto .....	72
Tabla 19. Actualizar proyecto .....	73
Tabla 20. Borrar proyecto .....	74
Tabla 21. Crear sprint.....	75
Tabla 22. Obtener sprints de proyecto.....	76
Tabla 23. Obtener sprint concreto.....	77
Tabla 24. Actualizar sprint .....	78
Tabla 25. Eliminar sprint. ....	79
Tabla 26. Crear estado.....	80
Tabla 27. Obtener estados.....	81
Tabla 28. Obtener un estado concreto .....	82
Tabla 29. Actualizar estado .....	83

Tabla 30. Crear tarea .....	85
Tabla 31. Obtener tareas .....	86
Tabla 32. Obtener tarea concreta.....	87
Tabla 33. Actualizar tarea.....	88
Tabla 34. Borrar tarea.....	89
Tabla 35. Crear cambio .....	90
Tabla 36. Obtener los cambios.....	92
Tabla 37. Crear votación.....	93
Tabla 38. Consultar votación.....	94
Tabla 39. Actualizar votación.....	95
Tabla 40. Borrar votación.....	96
Tabla 41. Crear voto.....	97
Tabla 42. Obtener votos .....	98
Tabla 43. Obtener voto concreto .....	99
Tabla 44. Modificar tarea .....	100
Tabla 45. Borrar voto.....	101

## Índice de Ilustraciones

Ilustración 1. Cuota de mercado Android vs iOS, 2017 (DeviceAtlas, 2017).....	16
Ilustración 2.Captura de pantalla de la aplicación Trello (Desarrollo propio).....	18
Ilustración 3.Tablero Trello con el estado del proyecto. (Desarrollo propio.).....	21
Ilustración 4. Login .....	27
Ilustración 5. Registro .....	27
Ilustración 6. Proyectos.....	28
Ilustración 7. Menú lateral.....	28
Ilustración 8. Perfil de usuario .....	28
Ilustración 9. Selector de avatar .....	28
Ilustración 10. Creación de proyectos.....	29
Ilustración 11. Selector de fechas .....	29
Ilustración 12. Información de proyecto.....	30
Ilustración 13. Gráfico de evolución .....	30
Ilustración 14. Buscador de usuarios .....	30
Ilustración 15. Editor de proyectos .....	31
Ilustración 16. Mensaje de confirmación .....	31
Ilustración 17. Sprints .....	31
Ilustración 18. Creación de sprints.....	31
Ilustración 19. Información sprint.....	32
Ilustración 20. Modificación de sprint.....	32
Ilustración 21. Agregar nuevo estado.....	32
Ilustración 22. Panel de estados .....	33
Ilustración 23. Estados cerrados.....	33
Ilustración 24. Selector de color .....	34
Ilustración 25. Información tarea .....	34
Ilustración 26. Historial de cambios .....	35
Ilustración 27. Poker planning .....	35
Ilustración 28. Arquitectura .....	38
Ilustración 29. Modelo de datos API (desarrollo propio).....	40



Ilustración 30. Tablero .....50

## Índice de diagramas

Diagrama 1. Datos.....	102
Diagrama 2. Caso de uso - Rol.....	103
Diagrama 3. Caso de Uso - Permiso .....	103
Diagrama 4. Caso de Uso - Usuario.....	104
Diagrama 5. Caso de Uso - Proyecto .....	104
Diagrama 6. Caso de Uso - Sprint.....	105
Diagrama 7. Caso de uso - Estado .....	105
Diagrama 8. Caso de Uso - Tarea.....	106
Diagrama 9. Caso de Uso - votación.....	106
Diagrama 10. Caso de Uso - Voto .....	107
Diagrama 11. Caso de Uso - Cambio.....	107

# 1. Justificación y Objetivos

Las metodologías ágiles están en auge. Su desarrollo basado en iteraciones ha cambiado la concepción del desarrollo software, convirtiendo cada una de estas iteraciones en un proceso propio, con objetivos muy concretos y con el propósito de aportar con el software valor de negocio a los clientes. Una de estas metodologías es Scrum.

En esta metodología se plantean una serie de elementos que ayudan a la comunicación del equipo y al conocimiento de este sobre el estado del desarrollo. Algunos de estos elementos son el uso de tableros y el "*Planning poker*".

Este trabajo plantea el desarrollo de una aplicación con toda la infraestructura necesaria para la gestión de proyectos mediante tableros, aproximándose al concepto de la metodología Scrum. La aplicación permitirá la creación de proyectos en los que pueden participar varios usuarios. Para poder dividir el proyecto en iteraciones se plantea la creación de *Sprints*, con la intención de indicar un periodo de tiempo en el que se van a realizar una serie de tareas. Estas tareas necesitan de un estado para conocer en qué fase del desarrollo se encuentran ("En desarrollo", "Terminado", ...), siendo estos estados de creación propia, pudiendo tener *Sprints* con diferentes estados a criterio del equipo de desarrollo.

Las tareas concentran los objetivos principales del trabajo. La posibilidad de cambiar su estado de desarrollo, plasmar información acerca de su objetivo y asignar a usuarios a tareas concretas permiten al equipo conocer cómo está avanzando el proceso de desarrollo. Pero, además, la opción de abrir una votación sobre cada tarea con el objetivo de determinar la dificultad de esta, basándose en la opinión de los desarrolladores que participan en el proyecto, cubre junto a los tableros otra de las premisas de este trabajo: la comunicación con el equipo. Esto permite conocer la opinión de cada persona sobre la dificultad de esa tarea.

Por último, la posibilidad de incluir la dirección del repositorio *GitHub*, así como los datos que se aportan sobre cada uno de los elementos del proyecto, permitirán conocer de manera gráfica el estado del desarrollo con información relevante como gráficos *Burndown* o el número de tareas por estado (columna).

## 2. Recursos

Los archivos fuente de este proyecto se encuentran alojados en la plataforma GitHub de forma pública, de modo que pueden ser consultados por cualquier usuario.

- API: <https://github.com/Sjgu1/tfg-api/>
- APP: <https://github.com/Sjgu1/tfg-app/>

La aplicación se encuentra disponible en Play Store, la tienda de aplicaciones de Google. Se puede comprobar su funcionamiento accediendo a la siguiente dirección e instalando la aplicación:

- <https://play.google.com/store/apps/details?id=sergiojuliogu.m.myapplication>

## 3. Introducción

### 3.1. Historia

En la actualidad, el concepto de desarrollo software es muy diferente al que se tenía en mente hace unos años. Se ha demostrado que el uso de una metodología es casi indispensable para el buen desarrollo software, sobre todo en un entorno laboral en el que hay un equipo de trabajo. También ha quedado patente que las metodologías clásicas funcionan, si bien, con la aparición de nuevas plataformas, como dispositivos móviles, o la mutabilidad de los proyectos durante su proceso de desarrollo, ha dado lugar a la aparición de nuevas interpretaciones del desarrollo software, y con ellas, nuevas metodologías.

Estas nuevas metodologías han recibido el nombre de metodologías ágiles, y todas tienen elementos en común y características propias. Una de estas metodologías que ha estado cogiendo protagonismo en los últimos años es Scrum.

Aunque conocer la metodología no es el fin de este trabajo, es importante tener en cuenta algunos de estos conceptos puesto que están directamente relacionados con la realización de este proyecto.

SCRUM plantea una serie de valores y principios para hacer que el desarrollo software no pertenezca exclusivamente a los desarrolladores, con una estructura cerrada en la que cada uno recibe una tarea y la realiza sin plantearse si es necesaria o si aporta valor al producto.

Esta metodología pretende incorporar a los clientes o interesados en la aplicación (a partir de ahora, "*stakeholders*") durante todo el proceso, y hacer del equipo de desarrollo un equipo de verdad, en el que se consideren las ideas, se resuelvan dudas y se ponga en valor la experiencia de los desarrolladores. Una forma de que el equipo tenga la comunicación necesaria para conocer el estado de la aplicación es realizar reuniones en diferentes ocasiones para discutir cómo evoluciona el desarrollo.

Además de estas reuniones, Scrum está basada en iteraciones, sobre las cuales se tienen unos objetivos concretos con el objetivo de llegar al final de cada una con algo que aporte valor al producto. Por lo que la división de un proyecto software en distintas iteraciones es indispensable. Esto nos plantea una división temporal del proyecto, en porciones de tiempo pequeñas en las que los objetivos pueden ser más concretos.

Los objetivos del desarrollo están consensuados con los *stakeholders* y el equipo de desarrollo, llegando a un acuerdo entre ambas partes. Este consenso no es inmutable, por lo que puede variar en el tiempo, siempre con la aceptación por ambas partes del desarrollo (equipo e interesados). La orquestación de este proceso se lleva a cabo por el rol conocido como *Product owner*. De estas reuniones, entrevistas y diferentes elementos utilizados para la obtención de los objetivos del software, se acaba obteniendo una idea de lo que se pide, y esta idea ha de ser dividida en elementos más pequeños, asumibles para el desarrollo dentro del tiempo marcado como un *sprint*, y esto es lo que se conoce como "Historia de usuario". Estas historias son los elementos que se han de desarrollar en la aplicación y que acaban aportando valor al software.

Este trabajo se centrará en la división de un proyecto en diferentes *sprints*, y sobre esto, en ayudar a decidir qué historias de usuario van a ser realizadas para ese *sprint*. La priorización de historias de usuario viene por parte del *product owner*, pero el decidir cuáles de estas historias de usuario se van a realizar en el *sprint* viene dado por el acuerdo entre el *product owner* y el equipo de desarrollo. Para poder realizar esta criba se ha de conocer el nivel de esfuerzo que requiere cada una de estas historias de usuario, y una de las formas de darle este valor a cada historia es mediante la votación, por parte de los miembros del equipo de desarrollo, de cada una de las historias, asignándoles la dificultad que cada uno considere.

En cada *sprint* solo se pueden realizar una cantidad limitada de historias de usuario, pero no todas requieren el mismo esfuerzo para ser desarrolladas, algunas pueden ser más complicadas que otras, por lo que el *sprint* que tenga estas historias tendrá menor cantidad que si todas las historias acordadas fuesen fáciles de implementar. Por eso, en lugar de asignar un número concreto de historias a un *sprint*, se pacta el valor de dificultad que se asume para desarrollar, adaptando el número de historias elegidas en función de su valor de dificultad asociado. Uno de los objetivos de este trabajo es proporcionar este tipo de herramienta, permitiendo a los usuarios que participan en el proyecto asignarle un valor concreto a cada una de las tareas.

Otro de los valores de Scrum es la comunicación que ha de tener el equipo de desarrollo a lo largo de todo el proceso. Y esto, a parte de las reuniones que se realizan a diario, viene de la mano con el conocer el estado de desarrollo de la aplicación. La manera más cómoda es de forma visual, en un tablero, en el que las historias estén clasificadas por el estado en el que se encuentran, permitiendo a los desarrolladores conocer si hay algún cuello de botella para poder dedicar esfuerzo extra con el fin de no bloquear el flujo de trabajo.

El conocimiento de este estado es importante en cualquier situación, por lo que el tener a mano esta información puede ser crucial, y el poder modificarla para que cualquier miembro del equipo pueda consultarla es muy importante. Este es otro de los elementos principales de este trabajo, un conjunto de herramientas que permitan la gestión completa de un proyecto, desde la creación de los mismos, hasta la gestión individual de cada uno de los estados.

### 3.2. La idea

Desde el comienzo en el grado se nos ha ido proponiendo diferentes herramientas para la gestión de los proyectos realizados en grupo, como Jira® o Trello®. Por lo que ya nos es conocido el uso de tableros que nos permitan controlar el estado del desarrollo.

La idea de realizar uno de estos gestores orientada a la metodología Scrum viene por parte de un equipo de trabajo formado por cuatro alumnos, entre los que me incluyo, para la asignatura "Desarrollo de sistemas software". El equipo, a parte de por mí, estaba formado por Mario Martínez, Laila Pascual y Jesús Perales, y estaba tutorizado por Antonio Javier Gallego Sánchez.

En esta propuesta se realizó una plataforma web para la gestión de proyectos Scrum con tableros. Este sistema planteaba la subdivisión de los proyectos en *sprints* para poder gestionar de manera independiente cada uno de sus tableros, así como las tareas que lo formaban. Además, ofrecía la posibilidad de mostrar gráficas sobre el desarrollo del mismo, todo desde una plataforma web con un sistema montado en local, con estados de las tareas estáticos. Esta fue la primera aproximación al proyecto que se ha desarrollado en este trabajo.

Con la realización de otras asignaturas he conocido en más profundidad el uso de las metodologías ágiles, en concreto Scrum, así como el uso de tableros y la necesidad de poder consultar datos en más ocasiones que cuando se está delante del ordenador. El realizar trabajos en grupo exige que se tengan que realizar modificaciones sobre ciertos elementos en diferentes momentos, por lo que la posibilidad de realizar estos cambios sobre un dispositivo móvil es lo que ha dado lugar a este proyecto.

El poder dividir el sistema en una API que permita gestionar los datos y una aplicación que consuma la API es otro de los motivos de este proyecto, puesto que teniendo diferenciada estas partes da lugar a que se puedan desarrollar otras aplicaciones (escritorio, web, otros dispositivos) para obtener los datos.

Por último, y para completar la aplicación, el conocer la dificultad de una historia de usuario ha sido crucial para ajustarse a los tiempos de desarrollo con los que contamos a la hora de realizar prácticas para diferentes asignaturas, por lo que la posibilidad de realizar esta votación directamente sobre la aplicación, sin necesidad de realizar una reunión exclusiva para esto, es lo que termina de componer el cuerpo de la aplicación y justificar su desarrollo.

### 3.3. Objetivos

Se propone el desarrollo de una aplicación para dispositivos Android para la gestión de proyectos software mediante la metodología Scrum. Para poder trabajar de forma colaborativa sobre un mismo proyecto se utilizará un servidor que centralice toda la información y de acceso a una serie de servicios Web que permitan la gestión y consulta de los proyectos.

Los objetivos que el trabajo debe cumplir son:

- Creación y gestión de proyectos en la aplicación, con posibilidad de invitar a otros usuarios para colaborar.
- Gestión de los usuarios en los proyectos mediante un sistema de roles que limiten las acciones a realizar por cada usuario.
- Creación y gestión de *sprints* dentro de los proyectos, permitiendo introducir datos con información sobre los mismos.
- Creación y gestión de un sistema de tareas dentro de los *sprints*, permitiendo clasificar las tareas en función de los estados de los que disponga el *sprint*.
- Gestión de las tareas pudiendo modificar tanto su información como el estado, dejando un historial de las acciones ocurridas sobre la tarea.
- Un módulo de votación por cada tarea que permita a los usuarios que participan en un proyecto asignar un valor de dificultad a esa tarea.

La gestión de estos elementos significa que pueden ser modificados, consultados o borrados por parte de las personas que tengan los permisos necesarios.

## 4. Estado del arte

Hoy en día el uso de terminales móviles se ha convertido en algo habitual dentro de nuestra vida diaria. La gestión de muchas de las actividades diarias ahora es gestionada por aplicaciones móviles, como puede ser la agenda, el ejercicio o elementos para la comunicación con otras personas.

Pero no solo los elementos de nuestra vida personal han encontrado un marco dentro de los dispositivos móviles, cada vez son más el número de empresas que utilizan distintas aplicaciones para mantenerse en contacto con sus equipos de trabajo, pudiendo estar al tanto de su actividad laboral en cualquier situación. Las aplicaciones están cambiando el modo en el que vivimos y trabajamos (DeviceAtlas, 2017), y esto ha causado el aumento de aplicaciones disponibles para todo tipo de ámbitos.

Todo esto supone un aumento en el número de ventas y de ingresos generados por este mercado. Según un artículo de Merca2.0, "Se espera que para el 2020, el mercado de las aplicaciones genere alrededor de 189 mil millones de dólares en ingresos." (Mesa editorial



Merca 2.0, 2018). Por lo que este mercado sigue teniendo una larga vida por delante, por lo que apostar por él puede ser una buena opción de negocio.

Pero dentro del mundo de las aplicaciones móviles, el mercado está dividido por dos sistemas operativos que aglutinan la mayor cantidad de mercado, pero no de forma equitativa. Como se muestra en la Figura 1, el mercado está dominado por el sistema operativo Android.

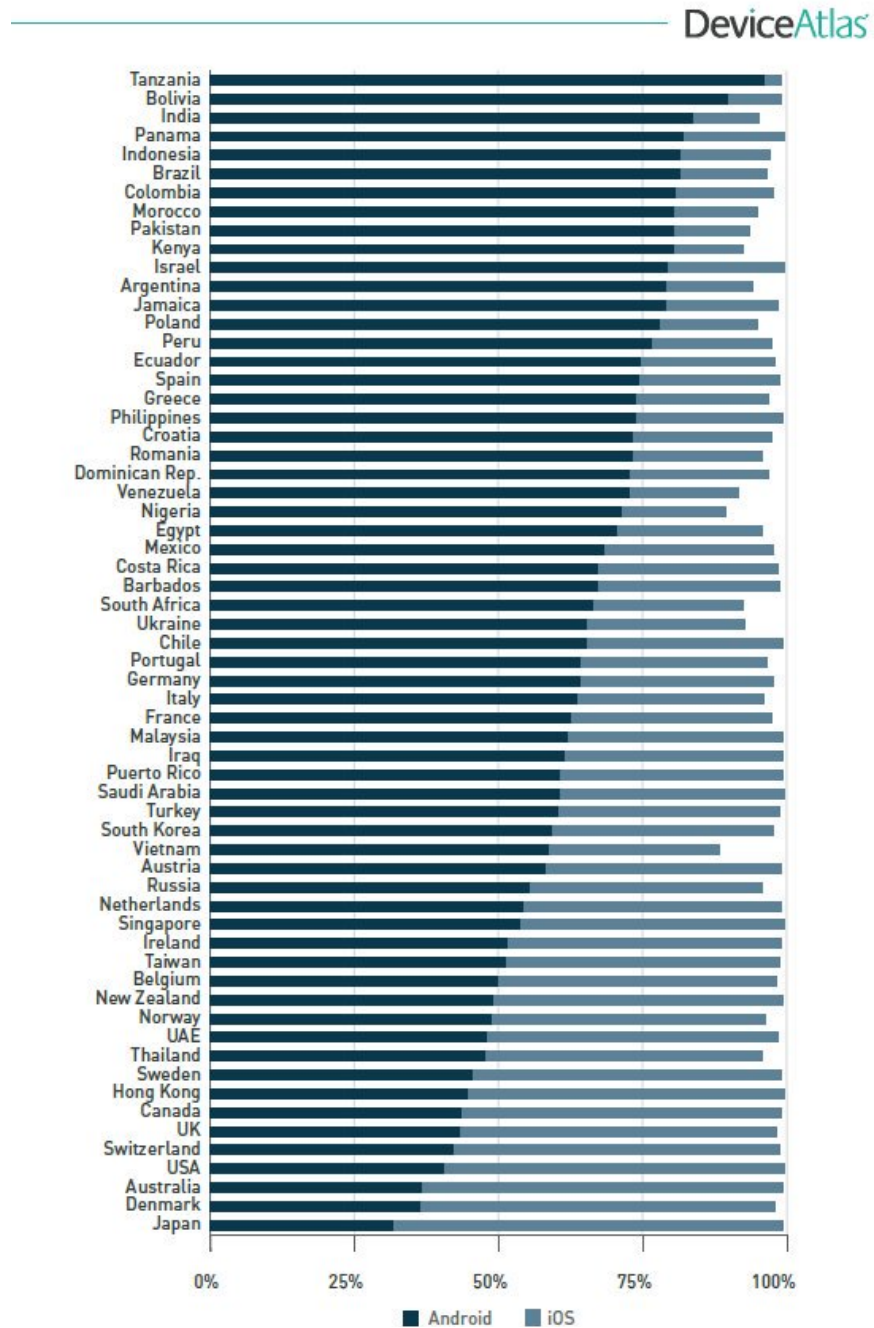


Ilustración 1. Cuota de mercado Android vs iOS, 2017 (DeviceAtlas, 2017)

Dentro del mercado de las aplicaciones móviles, voy a analizar cómo se encuentra el mercado de las aplicaciones para la gestión de proyectos en el desarrollo software.

Como he comentado anteriormente, muchas de las aplicaciones que se utilizan hoy en día para gestionar los proyectos software tienen un soporte web, dando por cierta la premisa de que los informáticos van a estar trabajando sobre su ordenador. Pero esto no siempre es así, puesto que aparte de reuniones, puede haber diferentes grupos de trabajos entre los que no se coincide en el horario de realizar las actividades propuestas del proyecto, por lo que puede ser necesario el uso de dispositivos móviles para realizar estas tareas.

Estas aplicaciones web han visto este problema, por lo que han adaptado sus aplicaciones a dispositivos móviles, compartiendo algunas características y diferenciándose en otras. Vamos a analizar las dos principales aplicaciones para la gestión de proyectos.

#### 4.1. Trello

Trello es un organizador de tareas. Permite la creación de tableros sobre los que crear tarjetas y organizar en distintas columnas personalizables como se muestra en la figura 2. No solo está orientado al ámbito del desarrollo software, sino a cualquier proyecto incluso a las tareas de la vida diaria. Esto hace de Trello una aplicación de uso muy general, permitiendo la gestión de los tableros en el desarrollo software.

El uso de Trello está muy popularizado entre los estudiantes de ingeniería informática, puesto que el aprender a manejarla no requiere de un gran esfuerzo y permite gestionar algunos aspectos de los proyectos, como pueden ser las tareas, de manera visual. Además, la posibilidad de crear equipos de trabajo facilita el poder gestionar tableros colaborativos, perfecto para las prácticas en grupo.

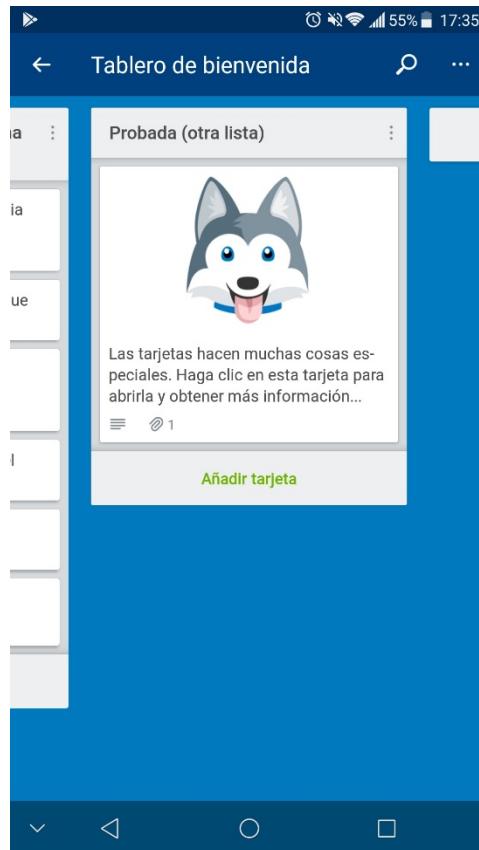


Ilustración 2. Captura de pantalla de la aplicación Trello (Desarrollo propio)

Sus puntos fuertes:

- Fácil de usar.
- Muy personalizable.
- Posibilidad de subir documentos y comentar tarjetas.

Esto hace de Trello una aplicación sencilla para empezar a conocer el tipo de aplicaciones para la gestión de tableros. Muy útil para el aprendizaje de las metodologías ágiles. Pero a su vez tiene unos cuantos defectos cuando se habla del desarrollo software bajo la metodología Scrum. Estos son algunos de sus defectos.

- Imposibilidad de crear proyectos con su información.
- Imposibilidad de crear *sprints* con su información.
- No dispone de datos estadísticos.
- No permite votaciones sobre una tarea.

Aunque la aplicación es atractiva, se queda corta en el manejo de datos orientados a las metodologías ágiles, y sobre todo en la información concreta y resumida sobre un proyecto, no solo sobre sus tareas.

## 4.2. Jira

Jira es una aplicación completamente orientada al desarrollo software, pudiendo crear proyectos, *sprints*, *issues*, documentarlos y muchas posibilidades útiles para los desarrolladores. Algunas de sus posibilidades son:

- Creación de proyectos.
- Creación de *sprints*.
- Creación de *issues*.
- Tableros personalizables.
- Documentación.

Tanto Trello como Jira pertenecen a Atlassian, que al ser una empresa dedicada a ofrecer estos servicios te garantiza unas calidades en su software. Pero Jira tiene algunas carencias y algunas dificultades que hacen que su uso no sea el más adecuado para todos los casos.

El software desarrollado para el móvil no es más que una adaptación de su versión web, teniendo esta como principal, por lo que no permite registrarse a través del móvil, únicamente acceder a una cuenta ya existente para poder consultar algunos de los datos.

Jira es de pago, por lo que puede no ser adecuado para ciertos puntos de aprendizaje en el mundo de las metodologías ágiles.

Por último, pero no menos importante, la curva de aprendizaje de Jira es muy elevada. Cuesta comenzar a utilizarlo, conocerlo y llegar a dominarlo, por lo que puede no ser la opción más adecuada para todos los grupos de desarrolladores.

Jira complementa las carencias de Trello, pero no permite su acceso a cualquier nivel de usuario, dejando un hueco para muchos desarrolladores que buscan una herramienta sencilla con la que mejorar el desarrollo de los proyectos software, que además pueden ayudarse de datos estadísticos sobre los proyectos para comprobar el trabajo pendiente, o de los usuarios que participan.

### 4.3. Conclusiones del mercado

Ha quedado reflejado que el mercado de las aplicaciones móviles sigue siendo una buena oportunidad de negocio, y siempre se pueden aportar nuevas ideas que ofrezcan alternativas a las aplicaciones actuales. Por esto, el desarrollo de una aplicación móvil, para dispositivos con sistema operativo Android, sigue siendo una buena opción.

Pero la pregunta más concreta es si hay cabida para la aplicación propuesta en este proyecto. Y analizando a los mayores competidores actuales, queda claro que hay una cuota de mercado dentro de las aplicaciones móviles orientadas al desarrollo software que no está cubierta.

Actualmente, las aplicaciones o no ofrecen una perspectiva de la gestión del desarrollo software orientada a las metodologías ágiles (como le pasa a Trello siendo de interés general), o los requisitos de conocimiento, aprendizaje y costos son muy elevados como para ser utilizada por usuarios que no pertenezcan a una organización que les pueda ofrecer esta formación o gastos.

Aparte de las razones anteriormente expuestas, este trabajo plantea otorgar nuevas funcionalidades, como son las estadísticas sobre el desarrollo o el módulo de votaciones para asignar a las historias de usuario la dificultad de desarrollo.

Esto hace que la aplicación sea diferente a las opciones que ofrece el mercado de aplicaciones móviles, mejorando algunas de las características ya existentes, añadiendo nuevas y cubriendo una cuota de mercado actualmente desierta.

## 5. Metodología

La API ha comenzado con una pequeña fase de diseño en la que se marcaban algunas de las relaciones que tenían las distintas entidades que la forman. Era un boceto conceptual sin más intención que la de conocer cómo interaccionan estas entidades con otras. Este diagrama acabaría convirtiéndose en el diagrama de datos que luego será explicado.

Para la planificación del trabajo opté por dividir el proyecto en tres partes diferenciadas:

- API.
- APP.

- Documentación.

Esto forman tres proyectos diferentes, pero que están relacionados entre sí. Para la gestión de los proyectos se ha utilizado un tablero de Trello (Figura 3.), en los que se han indicado las tareas que pertenecen a cada uno. Cada una de las tareas tiene asignado un color identificativo de modo que las tareas en rojo identifican aquellas relacionadas con la documentación, las violetas hacen referencia a las relacionadas con la API y las azules a la APP. En función de la fase de desarrollo del trabajo aparecerán unas u otras tarjetas, pudiendo juntarse en la etapa “En proceso” tarjetas de Documentación con uno de los otros dos elementos, puesto que se va documentando a la vez que se va desarrollando el proyecto. Sin embargo, los otros dos proyectos no deberían de juntarse en esta etapa, puesto que la Aplicación móvil depende directamente de la API, así que necesita de su completo funcionamiento para empezar su propio desarrollo.

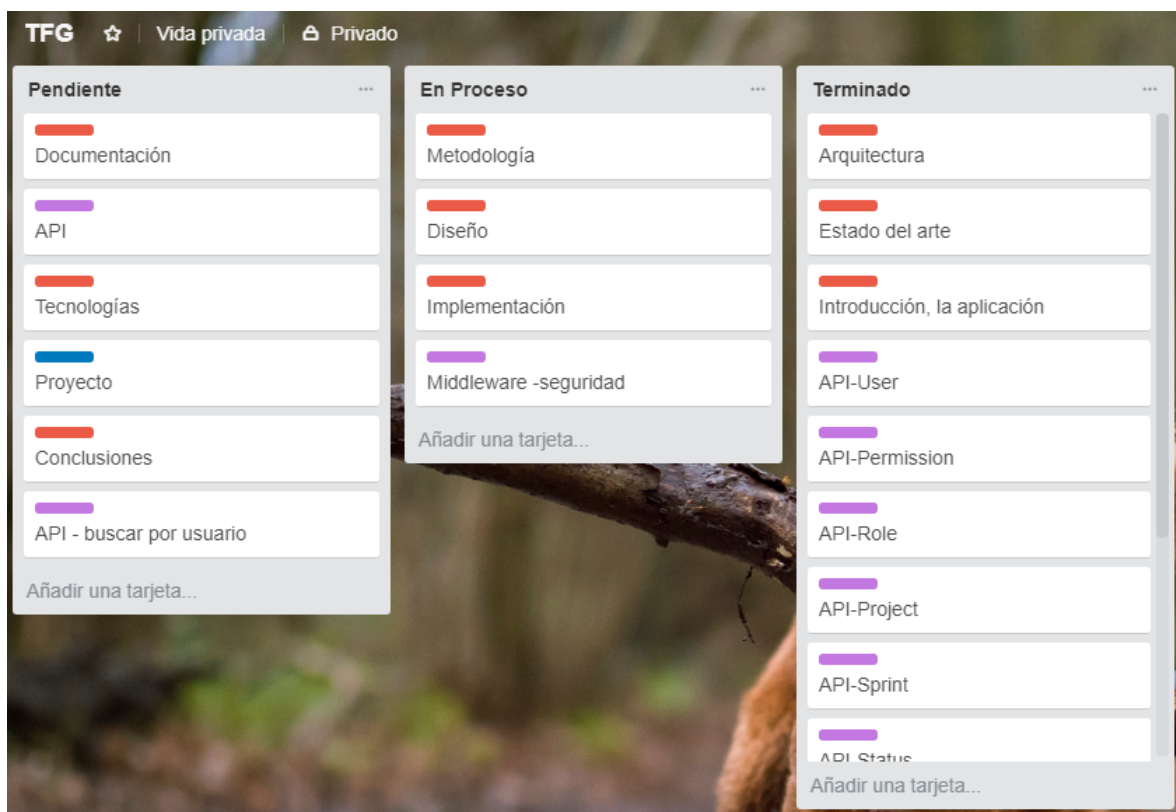


Ilustración 3. Tablero Trello con el estado del proyecto. (Desarrollo propio.)

Esta división en tres proyectos ha continuado dentro de cada uno, desgranando cada uno de estos proyectos individuales en elementos más pequeños y que aportan valor al trabajo una vez que están completados. Esto serían las “Historias de usuario”. En la figura 3, el

proyecto se encuentra en un punto intermedio entre el desarrollo de la API y el desarrollo de la APP, por lo que esta última aún no ha sido dividida en tareas más pequeñas, sin embargo, la API aparece por completo dividida y con casi todas sus historias terminadas. A la vez conviven con varias tareas de documentación que están siendo desarrolladas junto a una historia de usuario de la API. De hecho, algunas de estas tareas de documentación estarán en el estado “En Proceso” durante la mayor parte de desarrollo del trabajo.

Aunque el desarrollo no fue continuo, y no hay un equipo de trabajo como para poder asignarse roles, los principios seguidos son los de la metodología ágil Scrum, basándose en iteraciones y obteniendo requisitos, transformarlos a historias de usuario y estos a tareas más pequeñas. Es decir, he desarrollado una aplicación para la gestión de la metodología Scrum siguiendo la propia metodología Scrum.

Dividí la aplicación en diferentes *sprints* con unos objetivos claros al final de cada uno, ajustándolos a las fechas en las que más tiempo para realizarlos tenía. Cada uno tiene una duración de dos semanas, comenzando a contar en domingo. Durante todos los *sprints* he ido desarrollando la documentación, dedicando mayor tiempo a esta tarea en aquellos *sprints* donde finalizaba la implementación de la API y de la APP. Estos son los *sprints* con el contenido desarrollado en cada uno de ellos.

### 5.1. *Sprint 1*

1/10/2017 – 14/10/2017

Durante estas dos primeras semanas planteé los objetivos de la aplicación, realicé un boceto de las entidades y relaciones entre estas e investigué las tecnologías con las que se iba a desarrollar la API, que serán desarrolladas en el apartado correspondiente.

Este *sprint* finalizó con los requisitos mínimos que debería tener la aplicación, así como los conocimientos sobre cómo desarrollar un API rest en las tecnologías elegidas.

### 5.2. *Sprint 2*

15/10/2017 – 28/10/2017

Este *sprint* tenía planteado la implementación de la mitad de las entidades en la API, así como la documentación de los *endpoints* correspondientes. Además, también había planificado montar la base de datos no relacional en local para guardar la información.

El *sprint* finalizó con la implementación planteada, además pude desplegar la base de datos en remoto, que era el objetivo final de la aplicación, por lo que se adelantó trabajo planteado para el siguiente sprint.

### 5.3. *Sprint 3*

29/10/2017 – 11/11/2017

Este *sprint* tenía planteado la implementación del resto de entidades de la API, así como su documentación. Además, me propuse desplegar la API en un servidor externo, de forma que no tenga que tener mi máquina encendida para probar peticiones, pudiendo realizarlas desde cualquier lugar que me permitiese realizar peticiones HTTP.

El *sprint* finalizó con todo lo planteado, con la API funcionando en remoto con su correspondiente base de datos.

### 5.4. *Sprint 4*

18/03/2018 – 31/03/2018

Este *sprint* tenía planteado el diagrama de datos completo, con todas las entidades relacionadas, mejorando el diseño actual y con el diagrama completo de la API. Además, me planteé mejorar el uso de la base de datos que hasta ahora realizaba la aplicación con las últimas actualizaciones realizadas sobre el sistema de base de datos *MongoDB* y con el uso de las nuevas versiones de *Mongoose* para el uso mejorado de los *arrays* de modelos y de las colecciones.

El *sprint* finalizó con la actualización de la implementación a las nuevas versiones, con métodos más fáciles de implementar y comprender, preparándose para la implementación del resto de elementos que quedaban para completar la API.

### 5.5. *Sprint 5*

01/04/2018 – 14/04/2018

Este *sprint* tenía planteado la finalización de la implementación de la API, así como toda su documentación. Además, planteé el diseño de la aplicación móvil.



El *sprint* finalizó con la implementación de la API, el desarrollo de la documentación correspondiente. Faltó comenzar el diseño de la aplicación móvil, retrasado para la siguiente semana.

### 5.6. *Sprint 6*

15/04/2018 – 28/04/2018

Este *sprint* tenía planteado el desarrollo de los prototipos tempranos y la implementación del sistema de registro, conexión y gestión de los usuarios en la aplicación móvil.

El *sprint* finalizó con el desarrollo de todo lo planteado y se comenzó con la implementación de los apartados relacionados con la gestión de proyectos.

### 5.7. *Sprint 7*

29/04/2018 – 12/05/2018

Este *sprint* tenía planteado la finalización de la gestión de proyectos, y la gestión de *sprints* con sus paneles de estados y tareas.

El *sprint* finalizó con todo lo planteado.

### 5.8. *Sprint 8*

13/05/2018 – 26/05/2018

Este *sprint* tenía planteado finalizar la implementación de la aplicación, con las tareas, historial y Scrum póker.

El *sprint* finalizó con la aplicación terminada. El tiempo restante se utilizó para la creación de datos de prueba y terminar la documentación tanto.

## 6. Diseño

El diseño es una de la partes principales de la aplicación móvil, puesto que es el cliente mediante el cual los usuarios interactúan con el sistema, pudiendo consumir todas las funcionalidades que éste ofrece. Para poder conocer bien las acciones que los usuarios pueden realizar en la aplicación realicé una serie de diagramas de uso disponibles en el anexo 13.3. El diseño se ha dividido en dos apartados (prototipo temprano, diseño final), siendo uno la base para el resultado final de la aplicación.

## 6.1. Prototipo temprano

A la hora de realizar cualquier interfaz gráfica es fundamental tener claro desde un principio el diseño que se quiere tener del sistema. Para la realización de esta tarea he utilizado la herramienta Balsamiq (balsamiq, 2018) que permite la creación de *mockups* con varios elementos por defecto y la posibilidad de agregar nuevos elementos para mejorar los diseños.

La aplicación tiene un diseño claro debido a que se compone de varias jerarquías y pantallas de información. Para poder obtener la información de una tarea hay que localizar el *sprint* al que pertenece esta tarea, y éste se ha de localizar en el proyecto correspondiente. Durante el camino entre interfaces hasta llegar a la que queramos, se puede obtener la información referente al elemento en pantalla.

Los *mockups* no hacen referencia a todas las interfaces de la aplicación, solo son una aproximación para poder realizar la implementación con una idea desde el principio. No son contratos fijos que no puedan ser modificados, por lo que en comparación a la aplicación final existen variaciones. En la tabla 1 aparecen estos diseños. La herramienta utilizada no está orientada al 100% para el desarrollo de prototipos tempranos en dispositivos móviles, por lo que el aspecto puede recordar mucho a aplicaciones web, pero permite entender los datos que se especifican para esa interfaz en concreto.



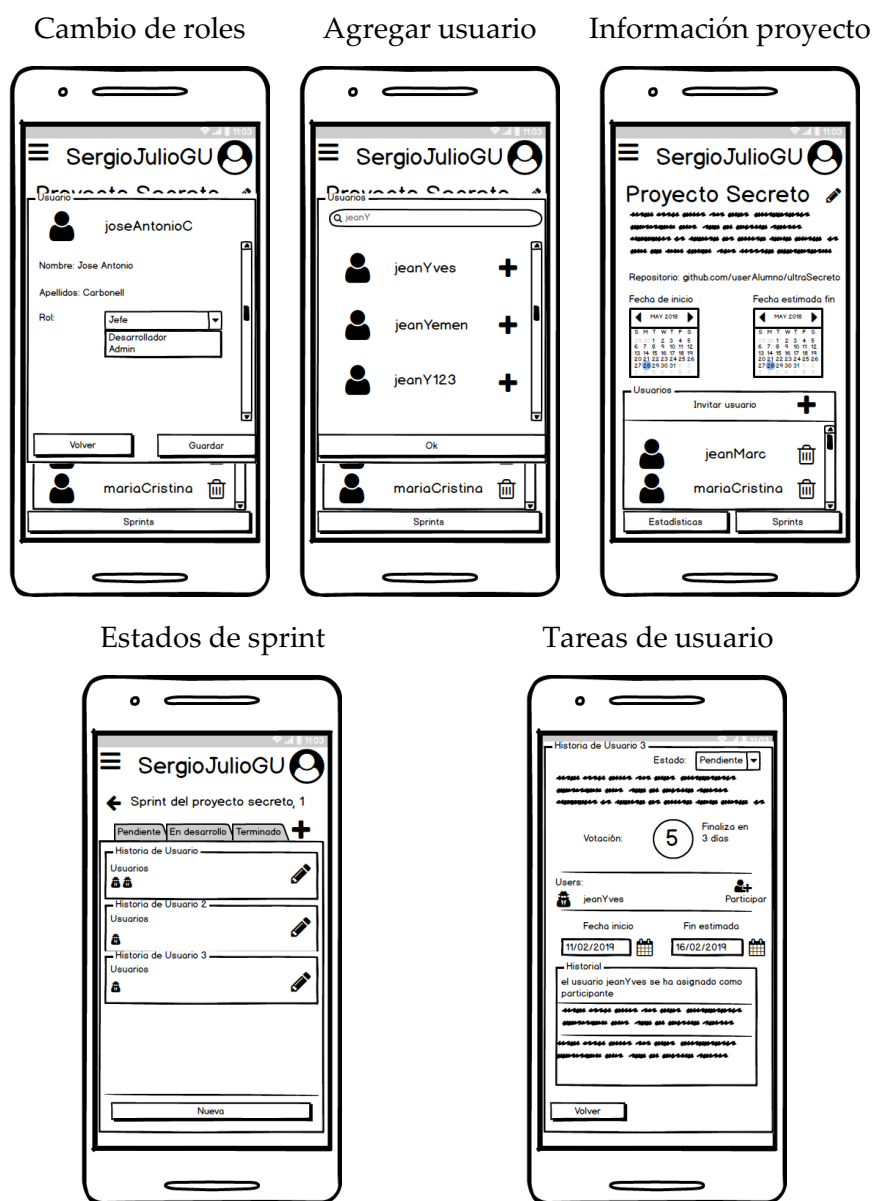


Tabla 1. Mockups de la aplicación móvil

## 6.2. Diseño final

Durante la implementación, y debido a las limitaciones del diseño de los *mockups*, algunas interfaces han sufrido modificaciones con el fin de hacer la experiencia más completa. Además, algunos de los elementos fueron divididos en diferentes interfaces para no saturarla de información no necesaria para ciertos usuarios, como puede ser la edición de proyectos a la que solo tienen acceso usuarios con los privilegios correspondientes en ese proyecto.

Para poder mostrar el diseño de la aplicación y entender las decisiones tomadas en lo referente a este apartado, a continuación se va a explicar las interfaces realizadas a modo de una secuencia real, de forma que este apartado sirva de referencia como guía de interacción con la aplicación.

Las funcionalidades básicas para un sistema de gestión de usuarios son el registro y el *login*, siendo básicas y con los campos mínimos para su ejecución con éxito.

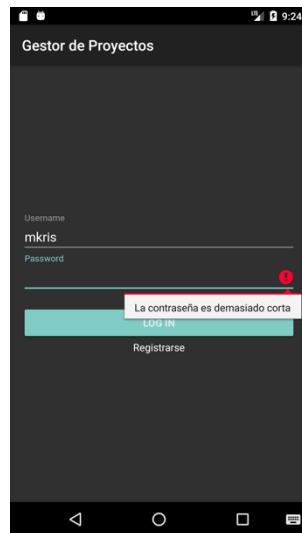


Ilustración 4. Login

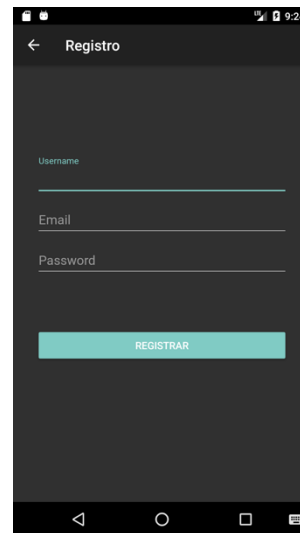


Ilustración 5. Registro

En la Ilustración 4 se muestra un ejemplo de la validación de datos. Siempre que tengamos algún error en los campos de entrada aparecerá el mensaje correspondiente al error, evitando que se realicen acciones con fallos.

Una vez que se ha accedido a la aplicación, en la primera interfaz se muestran los proyectos en los que estamos participando. Además, desde este punto se pueden acceder a dos menús diferentes para realizar otras acciones. El menú situado en la parte superior derecha (ilustración 6) permitirá hacer *Logout* en la aplicación. En el menú lateral izquierdo (ilustración 7) ofrece el acceso al perfil de usuario y a una interfaz de información con unos breves datos sobre la aplicación.

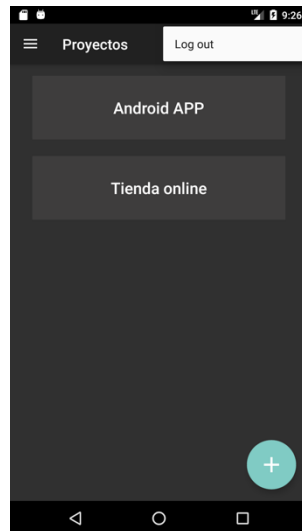


Ilustración 6. Proyectos

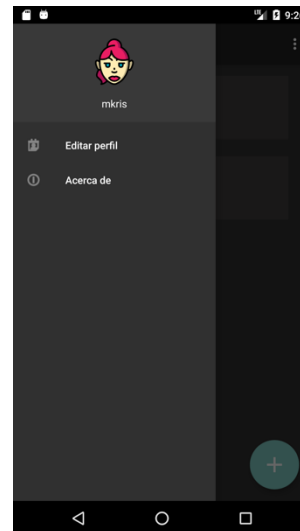


Ilustración 7. Menú lateral

Desde la gestión del perfil (ilustración 8) se pueden modificar los datos personales, así como seleccionar uno de los avatares que están dentro de la aplicación. Para poder acceder a esta opción se ha de pulsar el avatar actualmente asignado. El menú que se abre (ilustración 9) permite el cambio de avatar pulsando sobre la nueva imagen deseada.

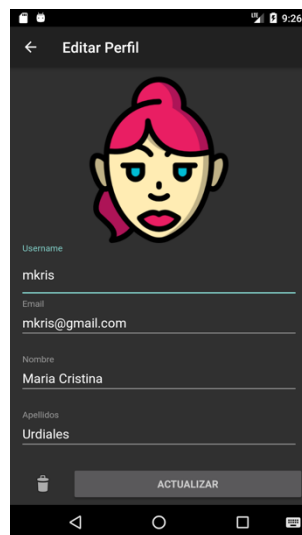


Ilustración 8. Perfil de usuario

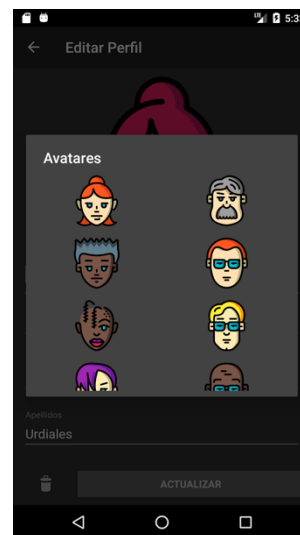


Ilustración 9. Selector de avatar

La interfaz de proyectos permite comenzar a utilizar la aplicación en caso de que aún no estemos asignados a ningún proyecto. Los iconos flotantes (el botón circular con el signo +) hacen referencia a la creación de nuevos elementos en función de donde nos encontremos, de forma que, si nos encontramos en un la interfaz inicial, se pueden crear nuevos

proyectos. En caso de estar dentro de un proyecto en la interfaz de *sprints*, se podrán crear estos elementos. Por último, si estamos en un estado, el botón permitirá agregar nuevas tareas a este estado.

Las interfaces de creación tienen los elementos necesarios que necesita esa entidad. En este caso, la información necesaria para un proyecto como se muestra en la ilustración 10. Los iconos de calendarios permiten abrir este tipo de elemento para facilitar la selección de fechas.



Ilustración 10. Creación de proyectos

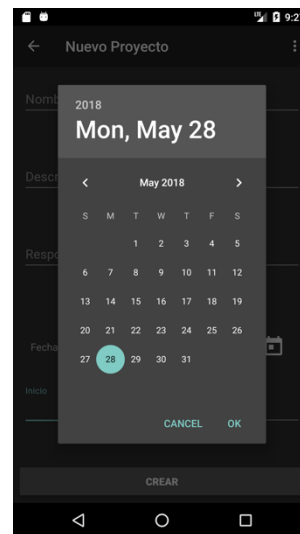


Ilustración 11. Selector de fechas

Si ya disponemos de proyectos, se puede acceder a la información de estos. En la interfaz del proyecto (Ilustración 12) se pueden observar todos los datos sobre éste, incluyendo los participantes del mismo. En esta interfaz se pueden realizar 4 acciones diferentes:

- **Agregar participantes al proyecto:** Permite a los administradores del proyecto invitar nuevos usuarios al proyecto realizando una búsqueda sobre su nombre de usuario e indicando el rol que se le asigna a este usuario, este rol se puede cambiar más adelante.
- **Editar:** Los administradores de los proyectos pueden acceder a esta sección donde podrán modificar los datos del mismo o realizar nuevas acciones.
- **Consultar estadísticas:** Desde este botón se pueden consultar una estadística del avance del proyecto.
- **Acceder a los *sprints*:** Acceso a los *sprints* de los que dispone este proyecto.

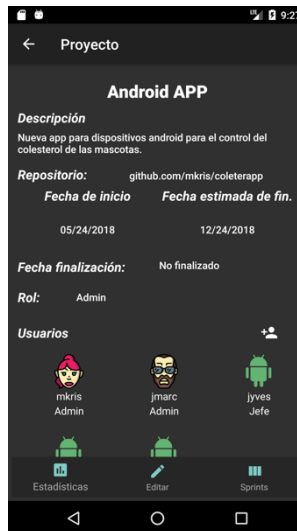


Ilustración 12. Información de proyecto

Las estadísticas (ilustración 13) muestran información si se tienen los datos suficientes sobre el proyecto, en este caso se tienen tres *sprints* finalizados y hay un *sprint* más que está sin finalizar, por eso a cada uno le corresponde un 25% del proyecto. El primer *sprint* se finalizó un poco más tarde de . La interfaz de agregar participantes (ilustración 14) sigue el diseño de los avatares, siendo un diálogo que permite seleccionar al nuevo usuario. Además dispone de la opción de búsqueda y de asignación de roles.

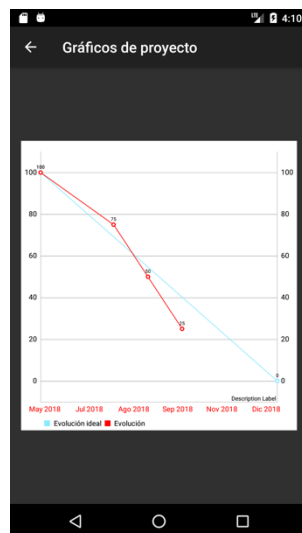


Ilustración 13. Gráfico de evolución

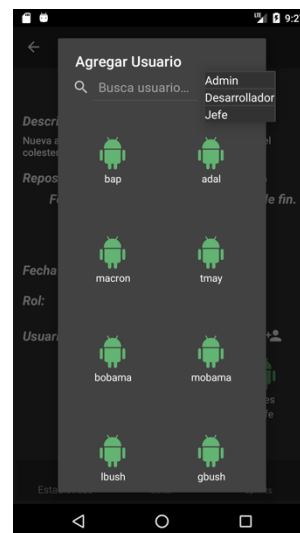


Ilustración 14. Buscador de usuarios

Si se tienen los permisos requeridos, se puede modificar la información de los proyectos (ilustración 15). Además, se pueden expulsar usuarios o modificar sus roles en el proyecto. En caso de querer eliminar el proyecto, la opción está localizada en esta interfaz, y los

elementos de eliminar tienen mensajes de confirmación (ilustración 16) para evitar posibles borrados accidentales.

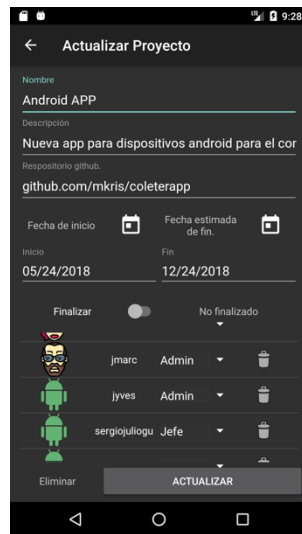


Ilustración 15. Editor de proyectos

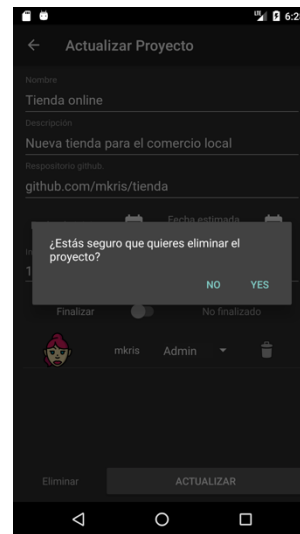


Ilustración 16. Mensaje de confirmación

Desde la información de un proyecto se puede acceder a los *sprints* que tiene asignados, así como crear nuevos con los datos correspondientes. El diseño de estas interfaces son similares a las de proyectos, puesto que su información es similar. Se pueden listar (ilustración 17) y crear los *sprints* (ilustración 18) como se muestra a continuación.

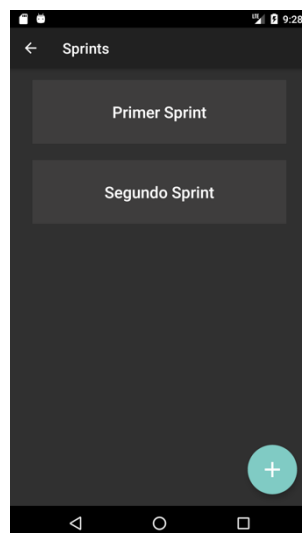


Ilustración 17. Sprints



Ilustración 18. Creación de sprints



En este punto comienza la parte más interactiva. La interfaz de información de un *sprint* (ilustración 19) es similar a la de un proyecto, con los mismo botones de editar y de visualizar gráficas, que en este caso se mostrará una estadística del *sprint*. Además de la información habitual de este tipo de elementos, se muestra una lista en la que se reflejan los estados (columnas) de los que dispone un sprint, con una referencia al número de historias de usuario que tiene asociadas. Desde el menú de edición del *sprint* (ilustración 20) al que tienen acceso solo los usuarios con los permisos mínimos para esta opción, se pueden modificar los datos o eliminar el *sprint*. Otra de las opciones que permiten dar dinamismo a la aplicación es la creación de nuevos estados, que se realizarán pulsando la opción “Agregar estado” disponible en esta interfaz. El diálogo correspondiente (ilustración 21) permite agregar el nuevo estado.



Ilustración 19. Información sprint

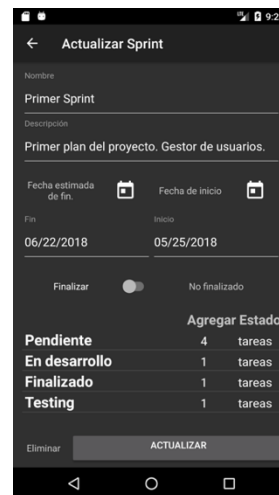


Ilustración 20. Modificación de sprint

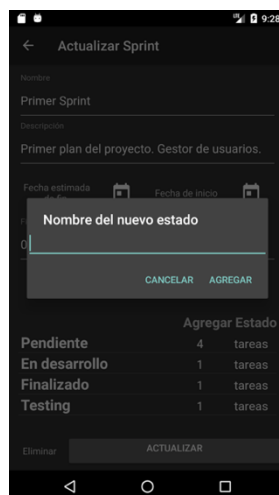


Ilustración 21. Agregar nuevo estado

Pulsando el botón de tableros se accederá a esta interfaz (ilustración 22) en la que se pueden clasificar las historias de usuario, crear nuevas, cambiar de estado las tareas, modificar el nombre del estado o cerrarlo. En estas interfaces solo se muestran los estados abiertos, en caso de que alguno se encuentre cerrado porque no va a ser utilizado, se puede acceder a ellos desde el menú superior derecho de la interfaz. Los estado cerrados se pueden volver a abrir como se muestra en la ilustración 23.

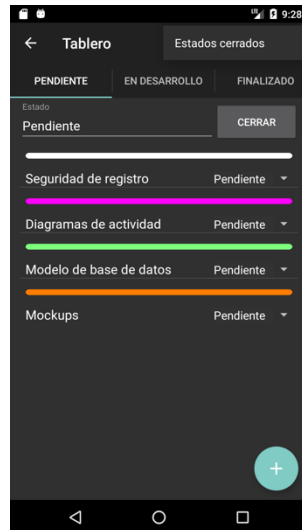


Ilustración 22. Panel de estados

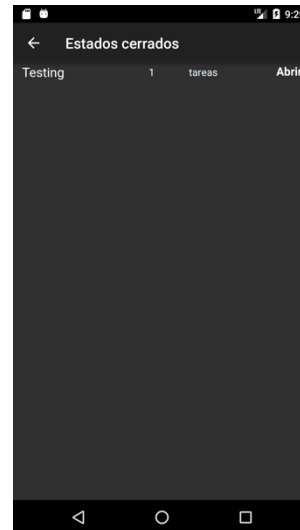


Ilustración 23. Estados cerrados

La creación de las tareas tiene los atributos correspondientes a esta entidad, teniendo como característica especial la posibilidad de asignar un color personalizado para poder diferenciar entre categorías. Este selector (ilustración 24) permite elegir cualquier color, pudiendo diferenciar entre muchas posibilidades las historias de usuario.

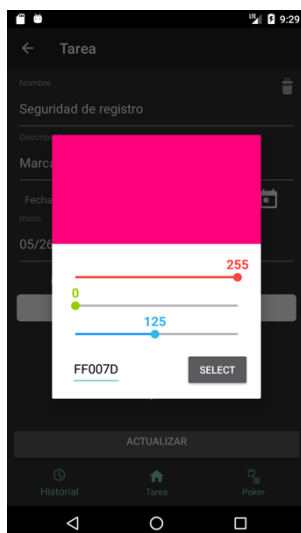


Ilustración 24. Selector de color

Como se puede observar en la ilustración 25, la tarea es editable para todos los usuarios, aunque solo es eliminable por usuarios con los permisos mínimos. Uno de los apartados de la tarea es asignarse a la misma, o en caso de ya estar asignado, borrarse de la tarea. Esta opción puede ser seleccionada por cualquier usuario para marcarse como responsable.

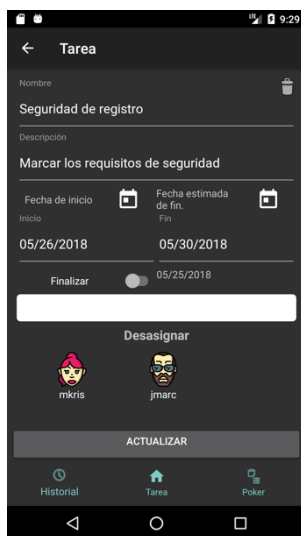


Ilustración 25. Información tarea

Cada tarea tiene un historial para conocer los cambios que ha ido sufriendo y localizar posibles errores cometidos. En la ilustración 26 se muestra un ejemplo de este historial. Pero, además de ver el historial, los usuarios con los permisos suficientes pueden crear una votación (*poker planning*) sobre esta tarea en caso de que no existiese. Si algún usuario ha creado una votación (ilustración 27), los usuarios del proyecto pueden acceder a votar por la dificultad que consideran que tiene esa historia de usuario. La votación requiere que el

usuario deje un comentario, que se mostrará en la información de la votación. Los votos se pueden cambiar dando una nueva justificación y también se puede eliminar la votación por un usuario con los permisos mínimos asociados.

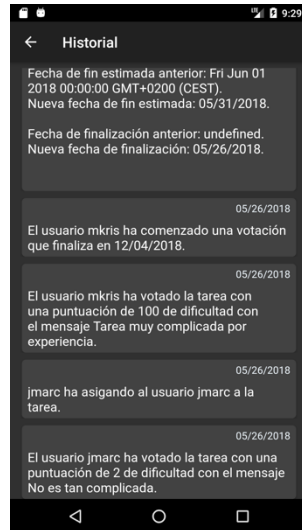


Ilustración 26. Historial de cambios



Ilustración 27. Poker planning

## 7. Tecnologías

Para el desarrollo del trabajo se han utilizado diferentes herramientas con objetivos muy distintos. Para poder explicar las tecnologías que se han utilizado este apartado se divide en tres categorías.

### 7.1 Gestión de proyectos y documentación

Algunas de las herramientas utilizadas son comunes para ambos proyectos, por lo que son nombradas en este apartado.

- **Microsoft Word:** Este procesador de textos permite realizar la memoria de un trabajo de forma muy cómoda y sencilla, disponiendo de herramientas para la generación de referencias, tablas e índices.
- **Trello:** Herramienta para la gestión de proyectos mediante tableros. En la Figura 3 aparece un ejemplo del uso de esta herramienta.
- **Git:** Git es un sistema de control de versiones multiplataforma, muy utilizado para la creación de proyectos informáticos, permitiendo guardar un historial de todos los

cambios que se han ido realizando, pudiendo recuperar cualquier versión del proyecto. Indispensable para escribir código hoy en día.

- **GitHub:** GitHub es una plataforma que permite almacenar los proyectos *git* de forma remota. Al disponer de una licencia por parte de la universidad, se pueden privatizar los proyectos para garantizar que no es accesible por usuarios externos.
- **Draw.io:** Esta herramienta permite la creación de diagrama de distintos tipos, incluyendo los de UML. Se ha utilizado para realizar los diagramas de datos y de casos de uso que se verán en los apartados correspondientes.

## 7.2 API y BBD

- **Visual Studio Code:** es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. (Wikipedia, 2018)
- **JavaScript:** Lenguaje de programación asíncrono utilizado en el lado del servidor para generar la API. Es un lenguaje interpretado, por lo que no necesita ser compilado cada vez que se realiza algún cambio. Fácil de trabajar en el IDE Visual Studio Code gracias a diferentes extensiones que hacen más cómodo implementar con él.
- **NodeJS:** Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente (Node.js Foundation, 2018). Es la base sobre la que se ha desarrollado la API.
- **ExpressJS:** Framework para NodeJS que ofrece la posibilidad de crear rutas sobre las que realizar llamadas HTTP. Además, ofrece la posibilidad de crear middleware que se ejecuten antes de los controladores de las rutas. Es gratuito y está bajo licencia de la fundación NodeJS.
- **MongoDB:** Se trata de un sistema de gestión de bases de datos no relaciones. En lugar de guardar los datos en tablas dentro de una base de datos, lo hace en archivos JSON, almacenando cadenas de texto relacionadas. Permite almacenar documentos con una estructura dinámica que se guardan dentro de colecciones. (Munro, 2017)

- **MongooseJS:** Es un *Object Document Mapper* (ODM). Esto significa que Mongoose permite definir objetos con un esquema fuertemente tipado que se asigna a un documento MongoDB (Munro, 2017). Permite la creación de modelos de datos que facilitan la gestión de la base de datos no relacional, permitiendo acciones como consultas, ediciones o borrados en modo de relaciones entre los modelos.
- **Mlab:** Es un servicio de bases de datos en la nube que permite alojar bases de datos MongoJs. En este servicio es donde se encuentra la base de datos, para no tener que montarla en un servidor propio. Este servicio utiliza otras plataformas como Amazon Web Services o Azure, para dar soporte a sus usuarios. Se considera un *Platform as a Service* (PaaS). Tiene una versión gratuita en la que permite el uso de hasta 500Mb de almacenamiento.
- **Heroku:** Se trata de otro PaaS que soporta diferentes lenguajes de programación. En mi caso es el servidor que aloja y despliega la API rest, de forma que no tenga que ser montada en un servidor propio. Tiene una versión gratuita que permite un número limitado de consultas mensuales.
- **ValidatorJS:** Librería escrita en JavaScript que permite la validación de datos en este lenguaje. Disponible en GitHub e instalado en nuestro servidor NodeJs mediante su gestor de paquetes *npm*. Se usa para comprobar que algunos de los requisitos de las peticiones se cumplen.
- **Postman:** Es un programa multiplataforma que permite realizar todo tipo de peticiones HTTP, con todos sus métodos. Además, ha sido utilizado para documentar los *endpoints*, puesto que guarda tanto la petición, como la respuesta, pudiendo agregar información adicional y publicar para tener esta información disponible en línea.

### 7.3 APP

- **Android studio:** Entorno de desarrollo para aplicaciones Android de la empresa Google basado en el IDE IntelliJ IDEA de JetBrains. Permite implementar aplicaciones para este sistema con interfaces base y facilidades de desarrollo para Android nativo.
- **Balsamiq:** Aplicación para la realización de *mockups* para plasmar la idea principal de las interfaces de la aplicación.

- **MPAndroidChart:** Se trata de una librería para proyectos Android que genera distintos gráficos a partir de ciertos datos mínimos.
- **Play Store:** Plataforma digital oficial para aplicaciones Android propietaria de Google. En esta plataforma se aloja la aplicación para facilitar su descarga por parte de todos los usuarios.
- **Java:** Lenguaje de programación nativo para las aplicaciones desarrolladas en Android.
- **XML:** Meta-lenguaje para la declaración de interfaces gráficas y recursos en Android.
- **Photoshop:** Herramienta de diseño gráfico utilizada para la creación del logo de la aplicación.

## 8. Arquitectura

La arquitectura que componen los elementos del proyecto es una arquitectura cliente-servidor de tres capas.

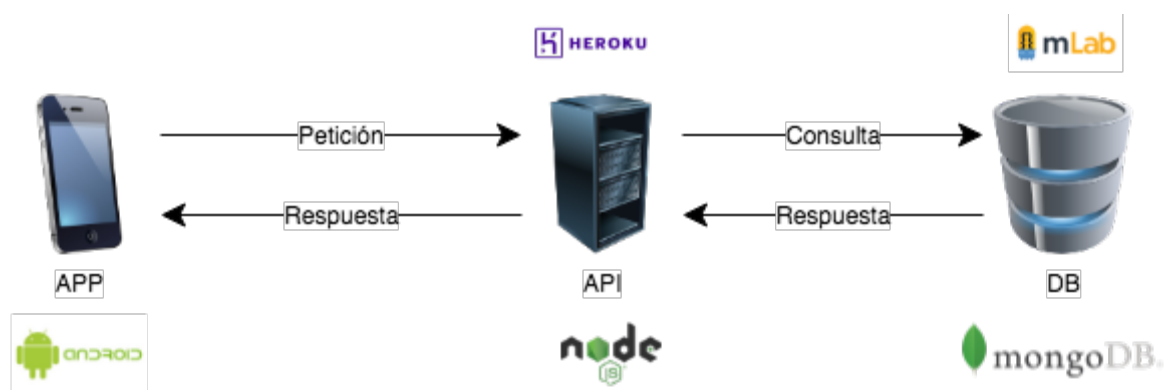


Ilustración 28. Arquitectura

En las arquitecturas tradicionales de dos capas, el servidor accede directamente a los datos y se los devuelve al cliente que realizó la petición.

En la arquitectura de tres capas, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:

- Un cliente, es decir, un equipo que solicita los recursos.
- Un servidor de aplicaciones, cuya tarea es proporcionar los recursos solicitados, pero que requieren de otro servidor para hacerlo.

- Un servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere.

Esto ofrece ciertas ventajas como:

- Los componentes de la aplicación pueden estar esparcidos en múltiples servidores permitiendo una mayor escalabilidad.
- Los problemas de limitación para las conexiones a las bases de datos se minimizan ya que la base de datos solo es vista desde la capa intermedia y no desde todos los clientes. Además de que las conexiones y los drivers de las bases de datos no tienen que estar en los clientes.
- Los clientes pueden ser desarrollados en diferentes lenguajes, pudiendo desarrollarlos para plataformas web, móviles o cualquier dispositivo que permita las conexiones a la capa intermedia.

En lo que se refiere a este proyecto, la APP es el cliente que realiza las peticiones al servidor en el que se localiza la API. Ésta es la encargada de realizar sus tareas y hacer peticiones al servidor de bases de datos que dispone de la información solicitada. La información solicitada es devuelta a la API y ésta responde a la APP para poder mostrarla.

## 9. Implementación

Como ya se ha comentado, el trabajo tiene dos partes importantes que hacen que el conjunto funcione, que son la API y la aplicación móvil. Las dos son importantes porque se complementan, necesitando de un cliente que sea capaz de interpretar y crear los datos de la API. Al ser tan importantes cada una de ellas, existen detalles de la implementación de ambos que se deben de tener en cuenta.

### 9.1. API

La API permite manejar todos los datos que se plantean en este trabajo para la gestión de los proyectos. Pero no solo sirve de conexión entre la base de datos y el cliente, además tiene parte de la lógica de negocio para así liberar de carga a la aplicación, que por lo general tendrá recursos más limitados de los que se pueden disponer en un servidor dedicado para la API.



Está desarrollada sobre *NodeJS*, utilizando las tecnologías anteriormente nombradas, utilizando rutas HTTP con los métodos principales que estas permiten para la gestión de los datos según se especifica en la arquitectura REST.

Para la implementación es necesario conocer el modelo de datos, para de esta forma conocer cuáles son y cómo interaccionan entre si los diferentes elementos que conforman un proyecto. Atendiendo a las necesidades que se planteaban, el modelo de datos resultante es el que se muestra en la Ilustración 4 (ampliado en los anexos).

Del modelo de datos se puede obtener el comportamiento de la aplicación. La base de todo comienza con un Usuario, puesto que es el que guarda relación con todas las acciones que se pueden realizar en el sistema. El Usuario es el encargado de crear los proyectos, pero no solo puede crearlos, puesto que lo importante de los proyectos Scrum es la colaboración, por lo que también pueden ser participantes de los proyectos, siendo invitados por algunos miembros del mismo. Aquí entra uno de los primeros conceptos de seguridad en los proyectos, que es el rol que cada uno de los usuarios participantes en un proyecto tiene en ese proyecto.

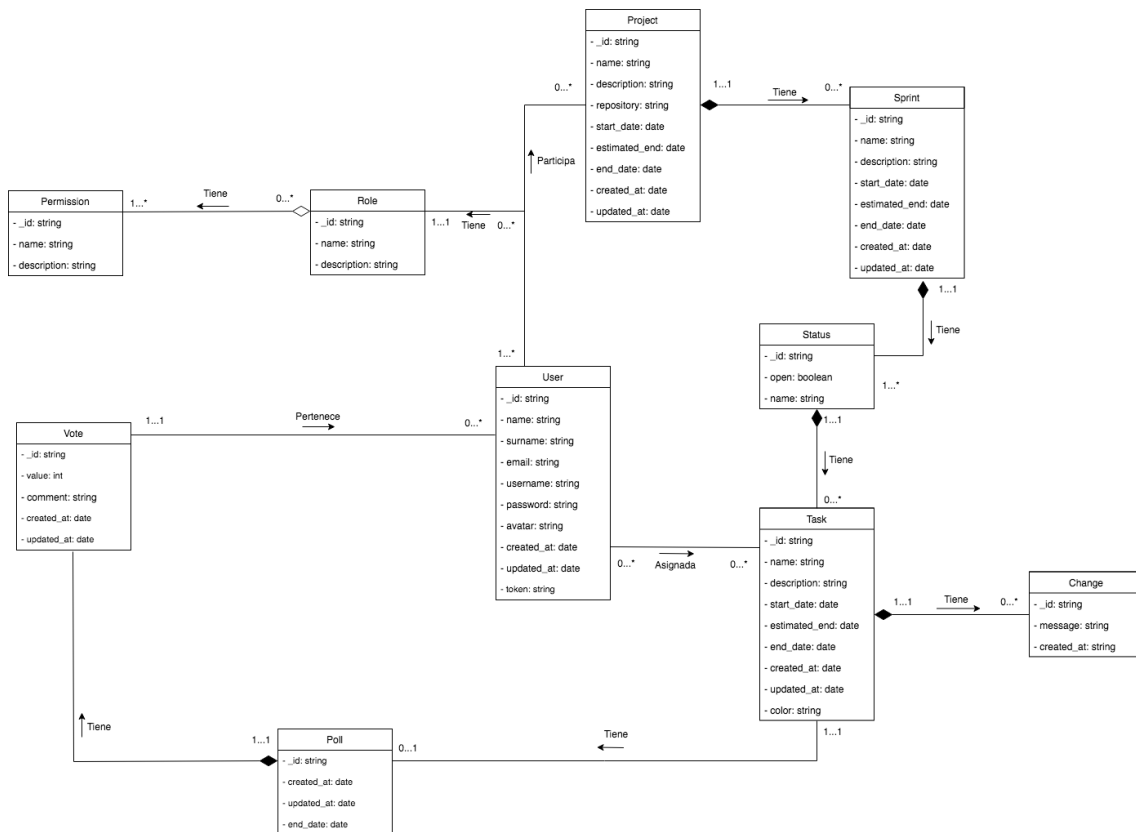


Ilustración 29. Modelo de datos API (desarrollo propio)

Para este trabajo se plantean roles estáticos, es decir, no se permite a los usuarios crear más.

Los roles actuales del sistema son:

- **Admin:** En el momento que un usuario crea un proyecto pasa a ser administrador del mismo, pudiendo realizar sobre este cualquier acción sobre cualquier elemento. Es el encargado de invitar al resto de usuarios, así como de cambiar los roles que estos tienen.
- **Jefe:** Este rol permite la gestión de los *Sprints* así como todo lo que esto implica, por lo que puede crear las historias de usuario que se realizarán en ese *Sprint*, gestionarlas y crear votaciones sobre ellas para asignarle una dificultad.
- **Desarrollador:** Puede gestionar todas las historias de usuario, comentarlas, votarlas, asignarse... todo lo que el sistema permite sobre estas tareas, sin posibilidad de crearlas o eliminarlas, recayendo esto sobre un jefe, para poder llevar un control de lo que ocurre.

Estos roles tienen unos permisos asociados que son los conocidos como CRUD (*Create, Read, Update, Delete*), teniendo cada rol unos permisos en concreto sobre unos elementos en concreto como he comentado. Por ejemplo, un desarrollador no podrá crear un nuevo *Sprint* al no disponer de estos permisos.

Estas dos entidades no pueden ser creadas ni modificadas, solo pueden ser consultadas por los usuarios de la aplicación. Son administradas directamente en el servidor.

Para el control de estas tareas se han desarrollado una serie de *middlewares* que controlan el acceso a las rutas, ejecutándose unos *middlewares* u otros en función del control que se quiera tener sobre la ruta. Los tres *middleware* que gestionan el acceso en función del rol son:

- **Comprobar Administrador:** Su función es controlar el acceso a las rutas de gestión de roles y permiso al administrador del sistema. Solo se utiliza para estas rutas.
- **Comprobar permisos Admin:** Actúa sobre las rutas de creación, modificación y eliminación de proyectos. Es el único que sobre un proyecto puede invitar nuevos usuarios y modificar su rol. No es exclusivo tener un solo *admin* sobre un proyecto, pueden asignarse varios.

- **Comprobar permisos Jefe:** Encargado de la gestión de los *sprints*, actúa sobre la creación y la actualización de estos. Puede crear columnas y las historias de usuario sobre estas (se explicará más adelante), también actúa sobre las rutas de gestión de votaciones para las historias de usuario.

Este sistema de *middlewares* garantiza cierto control sobre los proyectos, pero no la seguridad de los mismos, permitiendo únicamente acceso a las diferentes rutas de un proyecto si el usuario conectado participa o administra un proyecto. Para gestionar esto, sobre todas las rutas que parten de un proyecto se controlan con otro *middleware*:

- **Participa en Project:** Basándose en el usuario conectado, permitirá acceder a las rutas de los proyectos con los que tiene alguna relación, de forma que si consigue la ruta de un proyecto en el que no participa, no le permitirá acceder a ninguno de sus métodos.

Pero esto sigue sin garantizar que se accedan a las rutas usando el nombre de otro usuario, para esto existe un sistema de autenticación basado en el usuario conectado. A la hora de utilizar el sistema, todas las rutas pasan por un *middleware* que obliga a que el usuario esté conectado al sistema. Las únicas dos rutas que no lo hacen son las de registro y la de *login*.

Para el uso de este *middleware* se debe conocer el sistema de autenticación, que aunque es simple, da seguridad para el uso de la aplicación. El primer paso de seguridad comienza con el registro, que obliga a elegir un nombre de usuario y un email que no exista, garantizando que estos dos datos van a ser únicos. Además se pide una contraseña con la que se va a acceder al sistema.

Para garantizar que esa contraseña es conocida únicamente por el usuario que la crea, se almacena en la base de datos mediante *bcrypt*, por lo que no se guarda en claro.

Cuando un usuario se conecta a la aplicación, facilitando su usuario y la contraseña que eligió a la hora de registrarse, la API genera un *token* único, con un tiempo de expiración, de forma que cada vez que el usuario conectado quiera acceder a un recurso, es necesario agregar en la cabecera *Authorization* de la petición, este *token*. El *middleware* es el encargado de comprobar que el *token* es válido, incluyendo el tiempo de expiración. Al conectarse un usuario este *token* se actualiza, así como su tiempo de expiración.

Estos son los principios básicos que garantizan a los usuarios que sus proyectos no sean alterados por personas no autorizadas.

Con el sistema de roles y permisos, así como la seguridad, la interacción con el resto de entidades es la que comporta los objetivos de este trabajo, como la gestión de todos los proyectos. Los roles marcan qué puede realizar cada usuario en cada entidad (anexo 15.3 “Diagramas de casos de uso”), pero el entender cómo se relacionan es importante para saber los pasos que se han de realizar para la gestión de un proyecto.

### 9.1.1. Proyecto

A la hora de crear un proyecto se requieren algunos campos obligatoriamente, como es el nombre del proyecto, aunque éste no tiene por qué ser único, siendo identificado por la id que crea la base de datos. Pero otro de los datos que se solicitan como obligatorios es una url del repositorio. En este caso se comprueba que esta dirección sea válida, pero no que corresponda con un repositorio real o no. Es importante proporcionar la dirección de un repositorio de GitHub para poder ofrecer al usuario ciertas estadísticas sobre la evolución del proyecto en cuanto a código se refiere.

Otros datos son opcionales y pueden ser introducidos en otro momento, como son las fechas de inicio, de previsión de finalización y de finalización real. Estos datos serán usados por la aplicación móvil para desarrollar ciertas gráficas que también proporcionan información sobre el estado del proyecto. Aunque no sean obligatorios son importantes para obtener estas estadísticas, que siempre son una ayuda para los miembros del equipo, pudiendo modificar la organización en función de la situación en la que se encuentre.

No es el la única entidad que dispone de estos tres tipos de fechas. Son útiles para las estadísticas y permiten realizar retrospectivas más claras pudiendo hablar de cuándo se comenzó algo, cuándo estaba previsto terminarlo y cuándo se terminó en realidad.

Los proyectos se componen de *sprints*, por lo que borrando el proyecto, estos también desaparecerán.

### 9.1.2. Sprints

Los proyectos se organizan en periodos de tiempo conocidos como *sprints*. La creación de estos queda supeditada a usuarios en un proyecto que tengan el rol de Jefe como mínimo.

Estos *sprints*, además de nombres o descripciones, también posibilita la asignación de fechas para el desarrollo de sus estadísticas, teniendo los mismos significados que en un proyecto.

Si nos llevamos los *sprints* a algo visual dentro de la aplicación, serían “tableros”, sobre los que se van a poner las historias de usuario planteadas para ese *sprint*, y con esto viene algo importante que se plantea para este trabajo. Es conocido que las historias de usuario se suelen colocar en columnas en función del estado de desarrollo en que se encuentren, siendo las principales: Pendiente, En desarrollo, Finalizado. Pero se propone que estos estados no sean fijos, por lo que los *sprints* a su vez se componen de un número variable de estados (“columnas”), siendo diferente en cada *sprint*, y haciendo que si este desaparece, los estados y todo lo que estos tienen también desaparezcan.

Esta agregación obliga a que para poder crear las historias de usuarios sea obligatorio crear como mínimo un estado.

### 9.1.3. Estados

Forman parte de los *sprints*, y piden un nombre que sea descriptivo para la situación de las historias de usuario que albergará. Además, dispone de un atributo que permite abrir o cerrar el estado, lo que quiere decir que podemos cerrar un estado con todas las historias de usuario que tenga dentro. Es lo que se ve como columnas dentro de la aplicación, y tiene un agregación con las tareas. Esta agregación pretende manejar todas las tareas asociadas a un estado, a la vez, de forma que un estado (columna) no puede ser eliminado, pero sí puede cerrarse, de forma que se encuentre apartado de los estados actuales del *sprint* que tienen importancia. De esta manera, si un estado esta compuesto por muchas tareas que por ejemplo están en estado de pruebas, pero se ha decidido que ese estado no importa en ese momento, se puede cerrar el estado y con él todas las tareas asociadas. De manera visual desaparece toda la columna, dejando en el tablero únicamente los estados que son relevantes o que quieren ser mostrados en ese momento.

### 9.1.4. Tareas

La metodología Scrum plantea la división del trabajo en tareas pequeñas y asequibles, a esto se le llaman historias de usuario. Visualmente, en un tablero, estas historias se escriben sobre tarjetas que se colocan en las columnas. Este es el caso de esta entidad, las tareas son el simil de las historias de usuario, pudiendo pertenecer a cualquiera de los estados que

estén en un sprint concreto. Estas tareas tienen varios apartados para poder tener claro sobre qué trata, como son el nombre y la descripción. Además de estos dos elementos, las tareas tienen las tres fechas anteriormente mencionadas, de nuevo con el propósito de obtener datos estadísticos sobre el proyecto, tareas y evolución del *sprint*.

Uno de los apartados importantes de este trabajo y de la metodología es que las tareas tienen responsables o personas asignadas, de forma que sea sencillo reconocer quién está realizando qué tarea. La API también implementa esta lógica, pudiendo asignar o “desasignar” usuarios que participan en el proyecto en cada tarea.

#### 9.1.5. Votaciones

El punto diferenciador de este trabajo es la posibilidad de ofrecer a los miembros de un proyecto la posibilidad de votar la dificultad de las tareas, de forma que se pueda aproximar el esfuerzo necesario para realizar la tarea basándose en la opinión y experiencia de los miembros del equipo. Estas votaciones pueden ser abiertas sobre las tareas por parte de los jefes o administradores de proyectos, y cada usuario solo puede tener un voto sobre cada tarea. Además se ofrece la posibilidad de dejar un comentario en caso de que se quiera explicar por qué se ha asignado cierto valor en la votación, así como para decir por qué se ha cambiado el valor de un voto. Es importante que los que votan den su opinión, de modo que el resto de miembros puedan conocer y valorar el porqué se le ha estimado esa dificultad. De esta forma el resto de usuarios pueden cambiar de opinión para entre todos intentar aproximar de la forma más adecuada.

Las votaciones disponen de una fecha de finalización en la que se dejarán de permitir votos.

Las tareas disponen de un historial de cambios por los que ha pasado, desde las modificaciones que se le ha hecho a la misma, los cambios de estado y las votaciones que se han realizado sobre ella. Esta información es útil para evitar que se pierda información que puede ser relevante, como una descripción o los usuarios que han realizados las acciones.

## 9.2. APP

La aplicación está desarrollada para sistemas operativos Android, programada de forma nativa en Java y XML. Se trata del cliente mediante el cual los usuarios interactúan con el sistema.

Para poder conocer el desarrollo y las características de la aplicación, así como partes importantes o curiosas del código, se van a agrupar las implementaciones con características similares, de forma que compartan algunos elementos con implementaciones muy parecidas.

Antes de empezar con la implementación como tal, voy a explicar los elementos que puede tener un fichero. En lo referente a las interfaces, los elementos que podemos encontrar son:

- *Layouts* - Define la interfaz gráfica de una pantalla (o de una parte) de la aplicación.
- *Linear\_layouts* o *grupos de vistas* – Define la disposición visual de los elementos en la interfaz.
- Menús – Permiten definir los menús de opciones de la aplicación.
- Recursos – Permiten almacenar los recursos de la aplicaciones como imágenes, estilos, textos, etc.

Los tres primeros elementos no son mas que interfaces gráficas, con distintos elementos para montar la interfaz de usuario, pero cada uno se ha usado en diferentes contextos y con diferentes fines. Se explicará más adelante qué realiza cada uno.

En lo referente a programación Java, existen varios elementos diferentes:

- *Activities* - Define una actividad o pantalla de la aplicación.
- *Async Tasks* – Hilos asíncronos en Android que permiten la comunicación con la interfaz.
- *Fragments* – Definen una parte de una interfaz o pantalla de la aplicación.
- *Adapters* – Clase que gestiona la asignación de datos a elementos visuales repetitivos de la interfaz (como listados).

Algunas interfaces requieren del uso de los cuatro elementos, sin embargo otros, con una actividad es suficiente para mostrar la información necesaria. Durante el desarrollo de esta sección se hará referencia a elementos explicados en el apartado de diseño, donde se pueden apreciar los elementos de los que se habla.

### 9.2.1. Interfaces sencillas

A la hora de crear una nueva interfaz, Android Studio permite seleccionar una plantilla base sobre la que comenzar a programar. La mayoría de las interfaces están creadas sobre

un *template* vacío. Todas las interfaces, menos la de *Login*, tienen un padre, de forma que al retroceder siempre se va a acceder al elemento inmediatamente anterior.

Las interfaces sencillas, como pueden ser registro (ilustración 5), *login* (ilustración 4) o perfil (ilustración 8) únicamente están formadas por *layouts* en la interfaz, teniendo cada uno sus elementos característicos a modo de elementos de entrada, botones, o imágenes.

Estas actividades tienen asociada una clase *Activity* en java. En ella se carga la información necesaria para ser mostrada por pantalla, se activa o desactiva la barra de progreso durante las llamadas a la API y se crean los *listeners* para hacer la interfaz interactiva que así permita realizar las tareas asociadas.

Las actividades tienen asociados una serie de métodos que permiten realizar acciones en diferentes momentos de su ciclo de vida. Uno de los métodos principales sería el método “onCreate”, el cual sería el equivalente al constructor, y desde el cual se pueden inicializar todos los datos de la actividad. Para hacer referencia a los elementos de la interfaz o *layout* asociado se hace uso del método “findViewById”, el cual permite recuperar una vista desde código Java e interactuar con ella.

Con los elementos del interfaz localizado, es hora de pintar la información que estas interfaces han de tener, o de enviar información a la API mediante la gestión de eventos. Para realizar estas tareas se utiliza la clase **AsyncTask** (Android developer, 2018), que nos permite realizar llamadas a la API sin bloquear la interfaz de usuario.

Cada pantalla tiene una serie de tareas asíncronas asociadas, por ejemplo, para registrarse únicamente se necesita una tarea asíncrona que haga una llamada al *endpoint* correspondiente con la información necesaria. Sin embargo, para la gestión del perfil existen varias tareas asíncronas, una para modificar y otra para eliminar el perfil. Además de estas dos clases asíncronas, una tercera clase se ejecuta nada más abrir esta interfaz, y es la encargada de obtener la información referente al usuario y pintarla sobre la interfaz. Esto va a ocurrir en todas las interfaces que muestren algún tipo de información que se encuentre almacenada en la base de datos, de forma que siempre exista una llamada para obtener esta información.



Las llamadas asíncronas no siempre dan los resultados esperados, pueden surgir problemas y en caso de que esto ocurra, en pantalla se mostrará el error que ha ocurrido, de forma que si es un campo obligatorio que no se ha introducido, o la duplicidad de un campo único en la base de datos, el método correspondiente de gestionar el error es el encargado de indicar por pantalla el problema de forma que pueda ser solucionado por el usuario.

### 9.2.2. Interfaces medias

Algunas actividades requieren elementos más complejos para ser funcionales. Casi todas las interfaces tienen una dificultad de programación media debido a que tienen información obtenida de diferentes peticiones a la API, o elementos con muchos más datos.

Muchas interfaces tienen listas de datos, como pueden ser usuarios que participan en un proyecto o estados de los que dispone un *sprint*. Para poder rellenar estas secciones de la interfaz es necesario utilizar *adapters*.

Los *adapters* permiten rellenar las listas con *layouts* personalizados, de forma que se repita el *layout* por cada elemento de la lista. Además, permite que los componentes gráficos de cada elemento tengan sus propios *listeners* y realicen tareas concretas sobre el elemento.

Por ejemplo, cuando queremos invitar a un usuario a un proyecto, al pulsar sobre el usuario se envía una petición a la API con el identificador correspondiente al elemento de la lista seleccionado, sin importar su posición.

Para poder rellenar las listas con los elementos propios he creado los *linear\_layouts*, que es un nombre identificativo de las interfaces que sirven para rellenar las listas, de modo que sea fácil localizar la interfaz principal y los elementos que forman parte de listas dentro de esa interfaz. Algunos de estos elementos son recursos, como los avatares que pueden utilizar los usuarios para personalizar su perfil.

Uno de los retos con estas listas es que su contenido es dinámico, por lo que no sabemos el tamaño de la lista. Esto nos obliga a definir un tamaño por defecto y permitir que se haga *scroll* a lo largo de la lista. Por defecto no se pueden tener varios *scrolls* es una pantalla, y dado que todas tiene uno por defecto (se explicará en la sección de diseños adaptativos), las listas quedarían fijas mostrando únicamente los primeros datos. Para poder permitir

que varios *scroll* estén en la misma interfaz se ha utilizado el siguiente manejador de eventos sobre cada lista.

```
mListView.setOnTouchListener(new View.OnTouchListener() {  
    @Override  
    public boolean onTouch(View v, MotionEvent event) {  
        v.getParent().requestDisallowInterceptTouchEvent(true);  
        return false;  
    }  
});
```

Lo que permite diferenciar entre los diferentes *scrolls* de la interfaz.

Otros elementos que hacen más complicadas de implementar estas interfaces son los menús. Existen diferentes menús que dependen de la pantalla en la que se encuentre el usuario. En la pantalla de inicio por ejemplo existen dos menús, uno que permite acceder a la información del usuario (ilustración 7) y otro que permite al usuario desconectarse de la aplicación.

Estos menús se forman incluyendo en la interfaz otra interfaz del tipo menú y gestionando sus eventos en la actividad correspondiente.

### 9.2.3. Interfaces complejas

Las interfaces con más complejidad son las que se componen de todos los elementos anteriormente mencionados, que es el caso del tablero de cada sprint.

Aunque a simple vista parezca que no tiene muchos elementos, este tipo de interfaces depende de muchos factores. Voy a analizar junto con la interfaz los elementos que se necesitan para poder generar el tablero de un sprint. En la ilustración 30 se muestra un ejemplo de este tipo de interfaz.

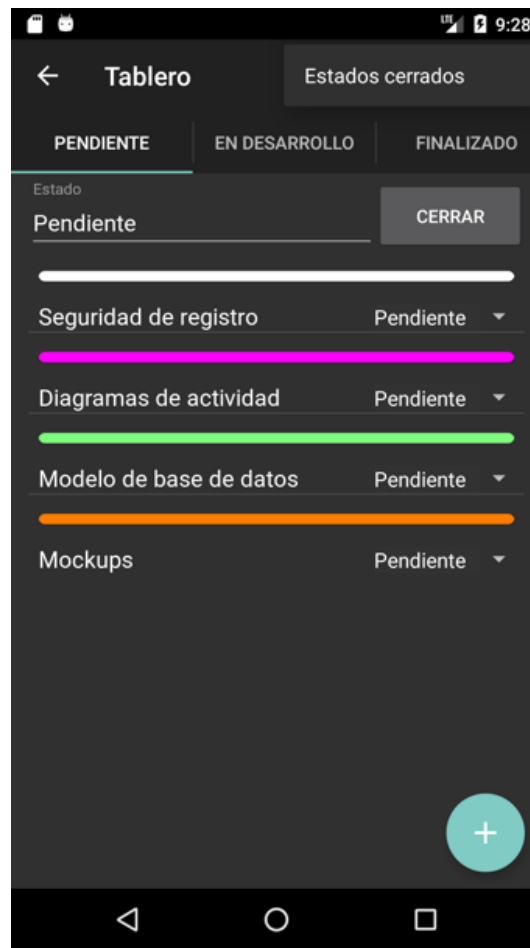


Ilustración 30. Tablero

En este ejemplo se muestra un tablero que tiene tres estados diferentes (Pendiente, En desarrollo, Finalizado). Cada uno de estos estados se formará por un conjunto de tareas independientes de las demás.

Lo primero es generar tantas pantallas como número de estados abiertos tenga el sprint. Para ello, antes de poder visualizar la interfaz, la actividad correspondiente crea tantos fragmentos como estados se tengan. Pero dado que este dato solo se conoce después de haber realizado una consulta a la API, es la tarea asíncrona correspondiente la que pasará esta información que depende del sprint, por lo que no ocurre en tiempo de compilación, sino que se tiene que realizar en tiempo de ejecución.

Cada fragmento, a parte de las historias de usuario que tenga asignadas, tienen elementos que parecen comunes pero que depende del estado seleccionado en el que nos encontremos. El nombre del estado que es editable desde esta pantalla tiene un evento que realiza la petición de cambio de nombre al perder el *focus* sobre la entrada de texto, de forma que, si

el texto cambia, se realizará la llamada correspondiente para modificarlo. Lo mismo ocurre con el botón de cerrar, que permite cerrar los estados para evitar tenerlos en el panel, pudiendo volver a abrirlos en cualquier momento desde su interfaz correspondiente. Y también ocurre con el botón de creación de una nueva tarea, que depende del estado actualmente seleccionado.

Estas tareas normalmente se realizan en la actividad, pero ahora el control lo tienen los fragmentos, complicando la implementación del código.

Esto causa problemas a la hora de actualizar la interfaz con posibles cambios, puesto que eso es tarea de las actividades, para ello he agregado un evento que ocurre en el momento que se selecciona un nuevo fragmento y he hecho uso de una clase estática que indique si existen o no cambios que deban ser reflejados.

```
@Override
public void onTabSelected(android.support.v7.app.ActionBar.Tab
tab, android.support.v4.app.FragmentTransaction ft) {

    mViewPager.setCurrentItem(tab.getPosition());
    if(Session.isCambios()){
        Session.setCambios(false);
        mSprintTask = new
SprintInfoTask(Session.getSprintSelected());
        mSprintTask.execute((Void) null);
    }
}
```

Para poder rellenar cada estado con sus tareas asociadas se utilizan los *adapters*, pero en este caso también tiene algo más de complicación, puesto que cada uno tiene un desplegable con información variable en función del estado, por lo que para ello se necesita enviar el resultado de dos llamadas a la API, una con la información de la tarea y otra con los posibles estados a los que esa tarea se puede cambiar.

Esta interfaz también tiene un menú personalizado que le permite acceder a los estados cerrados, por eso hace que esta interfaz tenga todos los elementos utilizados en la implementación de la aplicación.

#### 9.2.4. Otros elementos

Para poder realizar intercambios de información entre actividades se han utilizado características que ofrece Android, como puede ser Bundle o métodos que suceden al finalizar el ciclo de vida de una actividad.

Además de las características de Android, existe una clase estática con los valores propios de sesión, como puede ser el token de acceso, o los identificadores por los que un usuario va navegando en la aplicación, de forma que muchos datos son accesibles desde cualquier clase sin necesidad de utilizar el paso de datos ni parámetros entre métodos y clases. De esta forma, las llamadas asíncronas tienen acceso a las ID de los elementos para poder crear la url de un *endpoint* concreto.

Los elementos para seleccionar colores han sido implementados utilizando la librería Android material color picker dialog (captswag, 2018), que permiten seleccionar todos los colores en distintos formatos.

#### 9.2.5. Diseños adaptativos

Android permite asignar unos pesos a los elementos, tanto horizontal como verticalmente, de modo que no tienen un tamaño fijo asignado, sino que tienen una proporción de espacio del interfaz asignado. Esto me ha permitido asignar a las interfaces con información sin listas, un peso equitativo, de modo que cada elemento ocupe el mismo porcentaje de espacio. Sin embargo, todas aquellas interfaces con listas tienen asignadas a este elemento el doble o triple de espacio, puesto que se compone de muchos otros elementos y en ocasiones necesitan de desplazamiento vertical para poder ver todos los elementos.

Aparte de los pesos asignados por elemento, todas las interfaces disponen de un *scroll* lateral que permite que, si algún elemento queda fuera de pantalla por el tamaño de esta, se pueda hacer un desplazamiento vertical para poder mostrar toda la información.

## 10. Conclusiones

Para este trabajo de fin de grado se propuso el desarrollo de una aplicación para dispositivos móviles que permitiera realizar la gestión de proyectos mediante la metodología Scrum junto con un módulo de votaciones para realizar un planning póker sobre las historias de usuario.

Estos objetivos se han cumplido, desarrollando un proyecto dividido en tres partes.

Se ha utilizado un sistema de base de datos no relacional que permite guardar mucha información en poco espacio, alojando toda la información en un servidor externo. De esta forma he podido aprender a manejar este tipo de modelo de bases de datos y cumplir con el objetivo de almacenar la información.

La creación de una API REST ha eliminado la mayor parte de lógica de negocio del cliente, evitando la sobrecarga de los dispositivos. En su lugar se ha creado una API que puede ser reutilizada por otras plataformas, como pueden ser futuros proyectos web u otras aplicaciones, de forma que no es una implementación exclusiva de este trabajo, sino que tiene posibilidades de expansión. Además, al alojarse en un servidor como Heroku permite gran escalabilidad, de forma que no se limite a una simple prueba, sino que pueda ser utilizado por muchos usuarios.

La aplicación móvil me ha dado la capacidad de conocer la programación para otros dispositivos diferentes a los trabajados realizados durante el grado. La implementación de este cliente móvil se ha basado en reflejar todas las opciones que la API ofrece, de forma gráfica y usable por los usuarios, con poca carga de trabajo para poder ser utilizada por dispositivos más limitados.

Estos tres elementos han permitido poder completar todos los objetivos del planteamiento inicial. Además se implementó un sistema de gestión de usuarios para que se pueda utilizar la aplicación de forma colaborativa, con todo lo referente a proyectos que siguen la metodología Scrum, como pueden ser *sprints*, historias de usuario, comunicación entre los participantes, análisis de evolución...

Como planteamiento extra se propuso un módulo de votaciones sobre la dificultad de las tareas, lo que es conocido como *planning poker*. Este módulo se ha podido implementar de forma que lleve a más interacción entre usuarios y a posibles debates sobre la dificultad de las historias de usuario.

El proyecto tiene muchas posibles mejoras, como puede ser la obtención de datos estadísticos más completos, o personalización de roles y permisos, puesto que la lógica ya está implementada, solo habría que aplicarlo sobre el cliente.

## 11. Agradecimientos

Quiero agradecer a mi tutor Antonio Javier Gallego Sánchez por la dedicación y tiempo que ha empleado para que este proyecto saliera adelante.

Me gustaría felicitar y agradecer a la Universidad de Alicante y a los profesores del grado en informática por permitirme aprender e interesarme cada vez más por este mundo de las tecnologías y el desarrollo informático.

Por último, me gustaría dar las gracias al personal de la escuela politécnica, desde directivos, coordinadores, equipos de limpieza, secretariado, por hacer que el lugar de estudio tenga todas las comodidades que necesitamos y actividades complementarias para el disfrute de la etapa universitaria.

## 12. Bibliografía

- Android developer. (26 de Mayo de 2018). *AsyncTask*. Obtenido de <https://developer.android.com/reference/android/os/AsyncTask>
- balsamiq. (Abril de 2018). *Balsamiq*. Obtenido de <https://balsamiq.com/>
- captswag. (2018). *Android material color picker dialog*. Obtenido de <https://github.com/Pes8/android-material-color-picker-dialog>
- DeviceAtlas. (2017). *DeviceAtlas Mobile Web Intelligence Report Q2 2017*. DeviceAtlas.
- Fearn, N. (20 de Julio de 2017). The 50 best productivity apps for mobile devices in 2017. *techradar.com*.
- Jeff. (30 de Mayo de 2018). *Redes - Arquitectura Cliente/Servidor en 3 niveles*. Obtenido de <https://es.ccm.net/contents/147-redes-arquitectura-cliente-servidor-en-3-niveles>
- Mesa editorial Merca 2.0. (14 de Enero de 2018). Esto debes saber del mercado de las aplicaciones móviles. *Merca20.com*.
- Munro, J. (09 de Octubre de 2017). *Una introducción a Mongoose para MongoDB y Node.js*. Obtenido de <https://code.tutsplus.com/es/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527>
- Node.js Foundation. (08 de Abril de 2018). *NodeJs*. Obtenido de [nodejs.org](https://nodejs.org)
- Tallerbd. (30 de Mayo de 2018). *ARQUITECTURA CLIENTE-SERVIDOR DE 3 CAPAS*. Obtenido de <https://tallerbd.wikispaces.com/ARQUITECTURA+CLIENTE-SERVIDOR+DE+3+CAPAS>
- Wikipedia. (08 de Abril de 2018). *Visual Studio Code*. Obtenido de [wikipedia.org](https://wikipedia.org)



## 13. Anexos

### 13.1. Endpoints API

#### 13.3.1. User

USER		
Tipo de petición	POST	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/auth/signup	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/auth/signup	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
Cuerpo de la petición	<pre>{   "username": "newUserDoc",   "password": "password",   "email" : "newUserDoc@gmail.com" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
201	<pre>{   "user": {     "projects": [],     "_id": "5acb6155be9c3d000432e2ee",     "username": "newUserDoc",     "password": "\$2a\$10\$0/jIjXkVAxDE7/TwAYnHUusx8bGqKCPX6qmK3O6MChwN2YR26Z2Kq",     "email": "newUserDoc@gmail.com",     "created_at": "2018-04-09T12:49:25.586Z",     "updated_at": "2018-04-09T12:49:25.586Z"   } }</pre>	
400	Los campos username, password y email son obligatorios	
409	Existe un usuario con el mismo nombre de usuario	
409	El email ya existe en la base de datos	

500	Error al crear el usuario por parte del servidor
-----	--

Tabla 2. Registro Usuario

USER		
Tipo de petición	POST	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/auth/login	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/auth/login	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
Cuerpo de la petición	<pre>{   "username": "newUserDoc",   "password": "password" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
201	<pre>{   "token":   "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUuLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94" }</pre>	
401	No existe el usuario en la base de datos	
500	Contraseña errónea	
500	Error al guardar el token	

Tabla 3. Login Usuario

USER		
Tipo de petición	GET	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc	
	Nombre	Valor

Cabeceras de la petición	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
<b>Respuestas</b>		
<b>Status</b>	<b>Cuerpo de la respuesta</b>	
200	<pre>{   "_id": "5acb6155be9c3d000432e2ee",   "projects": [],   "username": "newUserDoc",   "password": "\$2a\$10\$0/jIjXkVAxDE7/TwAYnHUusx8bGqKCPX6qmK3O6MChwN2YR26Z 2Kq",   "email": "newUserDoc@gmail.com",   "created_at": "2018-04-09T12:49:25.586Z",   "updated_at": "2018-04-09T12:57:37.403Z",   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94" }</pre>	
404	No existe el usuario	
500	Error del servidor	

Tabla 4. Obtener usuario.

USER	
Tipo de petición	PUT
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc

		Nombre	Valor
Cabeceras de la petición		Content-type	Application/json
		Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "username": "newUserDocUpdated",   "email": "newUserDoc@gmail.com",   "name": "Documentación",   "surname": "Apellidos documentación" }</pre>		
Respuestas			
Status	Cuerpo de la respuesta		
204	Actualizado		
400	Los campos username y email son obligatorios		
400	Email incorrecto		
404	No existe el usuario		
409	Existe un usuario con el mismo nombre de usuario		
409	El email ya existe en la base de datos		
500	Error al actualizar		

Tabla 5. Actualizar Usuario

USER	
Tipo de petición	DELETE
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc
	Nombre
	Valor

Cabeceras de la petición	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTA5ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
<b>Respuestas</b>		
Status	Cuerpo de la respuesta	
204	Se ha borrado el usuario	
404	No existe el usuario	
500	Error al borrar	

Tabla 6. Borrar Usuario

<b>USER</b>		
Tipo de petición	GET	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/search/:username	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/search/ser	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
<b>Respuestas</b>		
Status	Cuerpo de la respuesta	
200	<pre>[   {     "_id": "5acb35bf0069b90004b71063",     "username": "newUser"   },   {     "_id": "5acb6155be9c3d000432e2ee",     "username": "newUserDoc"   }, ]</pre>	

	<pre> {   "_id": "5acb1ca099c0d7035e848c4d",   "username": "sergio" }, {   "_id": "5acb2deb8f67f005f814ac3a",   "username": "sergiojuliogu" } ] </pre>
404	No existe el usuario
500	Error del servidor

Tabla 7. Buscar usuario.

### 13.3.2. Permission

PERMISSION							
Tipo de petición	POST						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/permissions						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/permissions						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0NTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0NTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0NTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
Cuerpo de la petición	<pre> {   "name": "Consultar2",   "description": "Permisos de consulta" } </pre>						
<b>Respuestas</b>							

Status	Cuerpo de la respuesta
201	<pre>{   "permission": {     "_id": "5acb7be07cddff1bc66d2b9f",     "name": "Consultar2",     "description": "Permisos de consulta"   } }</pre>
400	El campo name es obligatorio
409	Existe un permiso con el mismo nombre, debe ser único
500	Error al crear el permiso

Tabla 8. Crear Permiso

PERMISSION							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/permissions						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/permissions						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
<b>Respuestas</b>							
Status	Cuerpo de la respuesta						
200	<pre>[   {     "_id": "5ac37d0b26c8340e03f66392",     "name": "Crear",</pre>						

	<pre>       "description": "Permisos de creación"     },     {       "_id": "5ac37d3126c8340e03f66393",       "name": "Editar",       "description": "Permisos de edicición"     }   ] </pre>
500	Error al conseguir los permisos

Tabla 9. Consultar Permisos

PERMISSION							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/permissions/:name						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/permissions/Consultar2						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
<b>Respuestas</b>							
Status	Cuerpo de la respuesta						
200	<pre> {   "_id": "5acb7be07cddff1bc66d2b9f",   "name": "Consultar2",   "description": "Permisos de consulta" } </pre>						
404	No existe el permiso						



500	Error al conseguir el permiso
-----	-------------------------------

Tabla 10. Obtener permiso por nombre

PERMISSION							
Tipo de petición	DELETE						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/permissions/:name						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/permissions/Consultar2						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
Respuestas							
Status	Cuerpo de la respuesta						
204	Se ha borrado el permiso						
404	No existe el permiso						
500	Error al conseguir el permiso						

Tabla 11. Borrar permiso

### 13.3.3. Role

ROLE			
Tipo de petición	POST		
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com		
Ruta	/users/:username/roles		
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/roles		
	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> </tbody> </table>	Nombre	Valor
Nombre	Valor		

Cabeceras de la petición	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "name": "Jefe2",   "description": "Tiene permisos para realizar todo lo posible sobre los sprints.",   "permissions": [{     "name": "Editar"   }] }</pre>	
<b>Respuestas</b>		
<b>Status</b>	<b>Cuerpo de la respuesta</b>	
201	<pre>{   "role": {     "_id": "5acb85c51dc7931c1c6520b0",     "name": "Jefe2",     "description": "Tiene permisos para realizar todo lo posible sobre los sprints.",     "permissions": [       {         "_id": "5acb85c51dc7931c1c6520b1",         "permiso": "5ac37d3126c8340e03f66393"       }     ]   } }</pre>	
400	El campo name es obligatorio	
400	Se necesitan permisos asociados al rol	
400	No existe ninguno de los permisos enviados	

409	Existe un rol con el mismo nombre, debe ser único
500	Error al crear el rol

Tabla 12. Crear Rol

Role							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/roles						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/roles						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTA2ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTA2ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTA2ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94						
<b>Respuestas</b>							
Status	Cuerpo de la respuesta						
200	<pre>[   {     "permissions": [       {         "_id": "5acb85c51dc7931c1c6520b1",         "permiso": {           "_id": "5ac37d3126c8340e03f66393",           "name": "Editar",           "description": "Permisos de edicición"         }       }     ]   },   {     "_id": "5acb85c51dc7931c1c6520b0",     "name": "Jefe2",</pre>						

	<pre>       "description": "Tiene permisos para realizar todo lo posible sobre los sprints."     }   ] </pre>
500	Error al conseguir los roles

Tabla 13. Obtener todos los servicios

ROLE							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/roles/:name						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/roles/Jefe2						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
Respuestas							
Status	Cuerpo de la respuesta						
200	<pre> {   "permissions": [     {       "_id": "5acb85c51dc7931c1c6520b1",       "permiso": "5ac37d3126c8340e03f66393"     }   ],   "_id": "5acb85c51dc7931c1c6520b0",   "name": "Jefe2",   "description": "Tiene permisos para realizar todo lo posible sobre los sprints." } </pre>						

	}
404	No existe el rol
500	Error al conseguir el rol

Tabla 14. Obtener rol concreto

ROLE							
Tipo de petición	DELETE						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/roles/:name						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/roles/Jefe2						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
Respuestas							
Status	Cuerpo de la respuesta						
204	Se ha borrado el rol						
404	No existe el rol						
500	Error al borrar el rol						

Tabla 15. Borrar rol.

### 13.3.4. Project

PROJECT	
Tipo de petición	POST
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects

		Nombre	Valor
Cabeceras de la petición		Content-type	Application/json
		Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jdQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "name": "Proyecto Doc",   "repository": "www.github.com/doc",   "description": "Proyecto de prueba." }</pre>		
Respuestas			
Status	Cuerpo de la respuesta		
201	<pre>{   "sprints": [],   "users": [],   "_id": "5acb8bb484aa7600043310c4",   "name": "Proyecto Doc",   "description": "Proyecto de prueba.",   "repository": "www.github.com/doc",   "created_at": "2018-04-09T15:50:12.344Z",   "updated_at": "2018-04-09T15:50:12.344Z" }</pre>		
400	Los campos name y respository son obligatorios		
400	El respositorio tiene que ser una url válida		
404	No existe el usuario		
404	No existe el rol		
500	Error al crear el proyecto		
500	Error al linkear proyecto al usuario		

Tabla 16. Crear proyecto.

PROJECT							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxYWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxYWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxYWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94						
Respuestas							
Status	Cuerpo de la respuesta						
201	<pre>[   {     "sprints": [],     "users": [       Se muestran los usuarios que participan con su rol y permiso.<sup>1</sup>     ],     "_id": "5acb8b1b4152940004f54e25",     "name": "Proyecto Doc",     "description": "Proyecto de prueba.",     "repository": "www.github.com/doc",     "created_at": "2018-04-09T15:47:39.641Z",     "updated_at": "2018-04-09T15:47:39.641Z"   } ]</pre>						
404	No existe el usuario						

<sup>1</sup> Se muestra la información de los usuarios que participantes en el Proyecto, junto con el rol que tienen y sus permisos asociados. Ocultado para facilitar la compresión de la petición.

500	No se han localizado los proyectos
-----	------------------------------------

Tabla 17. Obtener proyectos de usuario.

PROJECT							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects/:idProject						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJE1MjMyNzg2NTcsImV4cCI6MTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJE1MjMyNzg2NTcsImV4cCI6MTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJE1MjMyNzg2NTcsImV4cCI6MTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
<b>Respuestas</b>							
Status	Cuerpo de la respuesta						
200	<pre>{   "sprints": [],   "users": [     Se muestran los usuarios que participan con su rol y permiso.<sup>2</sup>   ],   "_id": "5acb8b1b4152940004f54e25",   "name": "Proyecto Doc",   "description": "Proyecto de prueba.",   "repository": "www.github.com/doc",   "created_at": "2018-04-09T15:47:39.641Z",   "updated_at": "2018-04-09T15:47:39.641Z"</pre>						

<sup>2</sup> Se muestra la información de los usuarios que participantes en el Proyecto, junto con el rol que tienen y sus permisos asociados. Ocultado para facilitar la compresión de la petición.



	}
404	No existe el usuario
404	El proyecto o no existe o no tiene acceso el usuario conectado
500	No se han localizado los proyectos

Tabla 18. Consultar proyecto concreto

PROJECT							
Tipo de petición	PUT						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
Cuerpo de la petición	<pre>{   "description": "Modificando la descripción",   "name": "Nuevo Nombre",   "repository": "www.github.com/new",   "users": {     "user": "sergiojuliogu",     "role": "Desarrollador"   } }</pre>						
Ejemplo 1 (insertar usuario)							

Cuerpo de la petición Ejemplo 2 (cambiar rol)	{ "operation": "UpdateRole", "username": "sergiojuliogu", "role": "Admin" }
Cuerpo de la petición Ejemplo 3 (expulsar usuario)	{ "operation": "DeleteUser", "username": "sergiojuliogu" }
Respuestas	
Status	Cuerpo de la respuesta
204	Se ha actualizado sin problemas
400	El repositorio es obligatorio
400	El repositorio tiene que ser una url válida
400	El usuario a agregar ya participa en el proyecto
404	El proyecto no existe o no tiene acceso el usuario conectado
404	No existe el rol asignado
404	No existe el usuario invitado
404	No existe el usuario
404	El usuario no participa
404	No existe el rol solicitado a modificar
500	Error del servidor

Tabla 19. Actualizar proyecto

PROJECT	
Tipo de petición	DELETE
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects/:idProject
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25



		5DvN708PsVtMFBEP- YzSJwVuvy94
Cuerpo de la petición	<pre>{   "name": "Primer Sprint Doc",   "description": "Principio de la aplicación",   "start_date": "12/04/2018",   "estimated_end": "12/05/2018" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
201	<pre>{   "status": [],   "_id": "5acbb03c75e4ae1f78ee816c",   "name": "Primer Sprint Doc",   "description": "Principio de la aplicación",   "start_date": "2018-12-03T23:00:00.000Z",   "estimated_end": "2018-12-04T23:00:00.000Z",   "project": "5acb8b1b4152940004f54e25",   "created_at": "2018-04-09T18:26:04.472Z",   "updated_at": "2018-04-09T18:26:04.472Z" }</pre>	
400	El campo name es obligatorio	
500	Error al crear el sprint	

Tabla 21. Crear sprint

SPRINT		
Tipo de petición	GET	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI

		1YWNiNjE1NWJlOWMzZDAwMDQ zMmUyZWUiLCJpYXQiOjE1MjM yNzg2NTcsImV4cCI6MTYwMTA zODY1N30.jDQlHfVNxAENf1p 5DvN708PsVtMFBEp- YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
200	<pre>[   {     "status": [],     "_id": "5acbb03c75e4ae1f78ee816c",     "name": "Primer Sprint Doc",     "description": "Principio de la aplicación",     "start_date": "2018-12-03T23:00:00.000Z",     "estimated_end": "2018-12-04T23:00:00.000Z",     "project": "5acb8b1b4152940004f54e25",     "created_at": "2018-04-09T18:26:04.472Z",     "updated_at": "2018-04-09T18:26:04.472Z"   } ]</pre>	
404	No se encuentran los sprints asociados	
500	No se ha localizado el proyecto	

Tabla 22. Obtener sprints de proyecto

SPRINT		
Tipo de petición	GET	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c	
	Nombre	Valor
	Content-type	Application/json



		Nombre	Valor
Cabeceras de la petición		Content-type	Application/json
		Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "name": "Sprint actualizado",   "description": " Actualizada",   "start_date": "11/23/2018",   "estimated_end": "11/23/2019",   "end_date": "10/12/2019" }</pre>		
Respuestas			
Status	Cuerpo de la respuesta		
204	Se ha actualizado		
400	El nombre es obligatorio		
404	No existe el proyecto		
404	El sprint o no existe o no tiene acceso el usuario conectado		
500	Error al actualizar el sprint		

Tabla 24. Actualizar sprint

SPRINT	
Tipo de petición	DELETE
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects/:idProject/sprints/:idSprint
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8

	b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
<b>Respuestas</b>		
Status	Cuerpo de la respuesta	
204	Se ha borrado el sprint	
404	No existe el proyecto	
404	El proyecto o no existe o no tiene acceso el usuario conectado	
500	Error al actualizar los proyectos	
500	Error al conseguir el sprint	

Tabla 25. Eliminar sprint.

### 13.3.6. Status

STATUS					
Tipo de petición	POST				
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com				
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status				
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status				
	<table border="1" style="width: 100%;"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json
Nombre	Valor				
Content-type	Application/json				



Cabeceras de la petición	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	{ "name": "En desarrollo" }	
Respuestas		
Status	Cuerpo de la respuesta	
201	{ "tasks": [], "_id": "5acbc4122708f4204c4d7ae0", "name": "En desarrollo", "open": true }	
400	El campo name es obligatorio	
500	Error al crear el estado	

Tabla 26. Crear estado

STATUS		
Tipo de petición	GET	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acbb1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94

		1YWNiNjE1NWJlOWMzZDAwMDQ zMmUyZWUiLCJpYXQiOjE1MjM yNzg2NTcsImV4cCI6MTYwMTA zODY1N30.jDQlHfVNxAENf1p 5DvN708PsVtMFBEp- YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
200	<pre>[   {     "status": [       {         "tasks": [],         "_id": "5acbc4122708f4204c4d7ae0",         "name": "En desarrollo",         "open": true       }     ],     "_id": "5acbb03c75e4ae1f78ee816c",     "name": "Sprint actualizado",     "description": " Actualizada",     "start_date": "2018-11-23T00:00:00.000Z",     "estimated_end": "2019-11-23T00:00:00.000Z",     "project": "5acb8b1b4152940004f54e25",     "created_at": "2018-04-09T18:26:04.472Z",     "updated_at": "2018-04-09T18:48:36.082Z",     "end_date": "2019-10-12T00:00:00.000Z"   } ]</pre>	
404	No se encuentran los estados asociados	
500	Error al buscar los estados	

Tabla 27. Obtener estados

STATUS	
Tipo de petición	GET
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com

<b>Ruta</b>	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus	
<b>Ejemplo</b>	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0	
<b>Cabeceras de la petición</b>	<b>Nombre</b>	<b>Valor</b>
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0eXNzZG2NTcsImV4cCI6MTYwMTA0ZDQ1N30uamR1LHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
<b>Respuestas</b>		
<b>Status</b>	<b>Cuerpo de la respuesta</b>	
200	<pre>{   "tasks": [],   "_id": "5acbc4122708f4204c4d7ae0",   "name": "En desarrollo",   "open": true }</pre>	
404	No existe el sprint	
404	El estado o no existe o no tiene acceso el usuario conectado	
500	Error al conseguir el estado	

Tabla 28. Obtener un estado concreto

STATUS	
<b>Tipo de petición</b>	PUT
<b>URL Base</b>	https://sergiojuliogu-tfg-2018.herokuapp.com
<b>Ruta</b>	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus

Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTA0ODY1N30.jdQ1HfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "name": "Actualizado",   "open": "false" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
204	Estado actualizado	
400	El nombre y el estado es obligatorio	
404	El estado o no existe o no tiene acceso el usuario conectado	
404	No existe el sprint	
500	Error al actualizar el estado	

Tabla 29. Actualizar estado

### 13.3.7. Task

TASK	
Tipo de petición	POST
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks

Ejemplo	<a href="https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks">https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks</a>	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0YnZg2NTcsImV4cCI6MTYwMTA0ZDY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "name": "Tarea 1",   "description": "Nueva tarea",   "start_date": "10/02/2018",   "estimated_end": "10/09/2018",   "color": "#FFFFFF" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
201	<pre>{   "users": [],   "changes": [],   "_id": "5acbcaa52109bc2090ca8ea9",   "name": "Tarea 1",   "description": "Nueva tarea",   "start_date": "2018-10-01T22:00:00.000Z",   "estimated_end": "2018-10-08T22:00:00.000Z",   "color": "#FFFFFF",   "poll": null,   "created_at": "2018-04-09T20:18:45.984Z",   "updated_at": "2018-04-09T20:18:45.984Z" }</pre>	

400	El campo name es obligatorio
500	Error al crear la tarea

Tabla 30. Crear tarea

TASK							
Tipo de petición	GET						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status/5acbc4122708f4204c4d7ae0/tasks						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
<b>Respuestas</b>							
Status	Cuerpo de la respuesta						
200	<pre>[   {     "tasks": [       {         "users": [],         "changes": [],         "_id": "5acbaa52109bc2090ca8ea9",         "name": "Tarea 1",         "description": "Nueva tarea",         "start_date": "2018-10-01T22:00:00.000Z",</pre>						

	<pre> "estimated_end": "2018-10-08T22:00:00.000Z", "color": "#FFFFFF", "poll": null, "created_at": "2018-04-09T20:18:45.984Z", "updated_at": "2018-04-09T20:18:45.984Z" } ], "_id": "5acbc4122708f4204c4d7ae0", "name": "Actualizado", "open": false } ] </pre>
404	No se encuentran tareas asociadas
500	Error al obtener las tareas

Tabla 31. Obtener tareas

TASK		
Tipo de petición	GET	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idSTask	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbcaa52109bc2090ca8ea9	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM

		yNzg2NTcsImV4cCI6MTYwMTA zODY1N30.jDQlHfVNxAENf1p 5DvN708PsVtMFBEp- YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
200	<pre>{   "users": [],   "changes": [],   "_id": "5acbcaa52109bc2090ca8ea9",   "name": "Tarea 1",   "description": "Nueva tarea",   "start_date": "2018-10-01T22:00:00.000Z",   "estimated_end": "2018-10-08T22:00:00.000Z",   "color": "#FFFFFF",   "poll": null,   "created_at": "2018-04-09T20:18:45.984Z",   "updated_at": "2018-04-09T20:18:45.984Z" }</pre>	
404	No existe el estado	
404	La tarea o no existe o no tiene acceso el usuario conectado	
500	Error al obtener el estado	
500	Error al obtener la tarea	

Tabla 32. Obtener tarea concreta

TASK	
Tipo de petición	PUT
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idSTask
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbcaa52109bc2090ca8ea9



	Nombre	Valor
Cabeceras de la petición	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición Ejemplo 1 (cambiar datos)	<pre>{   "name": "Tarea actualizada",   "description": "Nueva descripción",   "start_date": "12/01/2018",   "end_date": "01/07/2019",   "estimated_end": "01/03/2019",   "color": "#AAAAAA" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
204	Actualizado	
400	El usuario ya está asignado a esta tarea	
400	El usuario no está asignado a esta tarea	
404	No existe el estado	
404	El usuario asignado no existe	
404	La tarea no existe o no tiene acceso el usuario conectado	
404	El usuario asignado no existe	
404	Tarea no encontrada	
500	Error del servidor	

Tabla 33. Actualizar tarea

TASK	
Tipo de petición	DELETE
URL Base	<a href="https://sergiojuliogu-tfg-2018.herokuapp.com">https://sergiojuliogu-tfg-2018.herokuapp.com</a>

<b>Ruta</b>	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idSTask	
<b>Ejemplo</b>	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbaa52109bc2090ca8ea9	
<b>Cabeceras de la petición</b>	<b>Nombre</b>	<b>Valor</b>
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
<b>Respuestas</b>		
<b>Status</b>	<b>Cuerpo de la respuesta</b>	
201	Borrada	
404	No existe el estado	
404	La tarea o no existe o no tiene acceso el usuario conectado	
500	Error al obtener la tarea	

Tabla 34. Borrar tarea

### 13.3.8. Change

CHANGE	
<b>Tipo de petición</b>	POST
<b>URL Base</b>	https://sergiojuliogu-tfg-2018.herokuapp.com
<b>Ruta</b>	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/changes/
<b>Ejemplo</b>	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8

	b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbcaa52109bc2090ca8ea9/changes/	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
Cuerpo de la petición	{ "message": "Sergio ha modificado la tarea" }	
Respuestas		
Status	Cuerpo de la respuesta	
201	{ "_id": "5acccbba97b53b700048b57a8", "message": "Sergio ha modificado la tarea", "created_at": "2018-04-10T14:35:21.686Z" }	
400	El campo message es obligatorio	
500	Error al crear el cambio	

Tabla 35. Crear cambio

CHANGE	
Tipo de petición	GET
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/changes/
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee81

	6c/status/5acbc4122708f4204c4d7ae0/tasks/5acbcaa52109bc2090ca8ea9/changes/	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
201	<pre>[   {     "users": [       "5acb2deb8f67f005f814ac3a"     ],     "changes": [       {         "_id": "5acccba97b53b700048b57a8",         "message": "Sergio ha modificado la tarea",         "created_at": "2018-04-10T14:35:21.686Z"       }     ],     "_id": "5acbcaa52109bc2090ca8ea9",     "name": "Tarea actualizada",     "description": "Nueva descripción",     "start_date": "2018-12-01T00:00:00.000Z",     "estimated_end": "2019-01-03T00:00:00.000Z",     "color": "#AAAAAA",   } ]</pre>	

	<pre> "poll": null, "created_at": "2018-04-09T20:18:45.984Z", "updated_at": "2018-04-09T20:39:24.971Z", "end_date": "2019-01-07T00:00:00.000Z" } ] </pre>
404	No se encuentra la tarea asociada
500	No se han localizado los cambios

Tabla 36. Obtener los cambios

### 13.3.9. Poll

POLL							
Tipo de petición	POST						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbcaa52109bc2090ca8ea9/polls/						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM1YnZg2NTcsImV4cCI6MTYwMTA1ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM1YnZg2NTcsImV4cCI6MTYwMTA1ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM1YnZg2NTcsImV4cCI6MTYwMTA1ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94						
Cuerpo de la petición	<pre> {   "end_date": "12/04/2018" } </pre>						
<b>Respuestas</b>							
Status	Cuerpo de la respuesta						

201	<pre>{   "votes": [],   "_id": "5accd673c4caec0004cfdb7e",   "end_date": "2018-12-04T00:00:00.000Z",   "created_at": "2018-04-10T15:21:23.606Z",   "updated_at": "2018-04-10T15:21:23.606Z" }</pre>
400	El campo end_date es obligatorio
404	No existe la tarea
404	Ya existe la votación
500	Error al crear la votación

Tabla 37. Crear votación

POLL							
Tipo de petición	GET						
URL Base	https://sergijuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll						
Ejemplo	https://sergijuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbccaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM1YzYyNTcsImV4cCI6MTYwMTA5ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM1YzYyNTcsImV4cCI6MTYwMTA5ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
	Nombre	Valor					
Content-type	Application/json						
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM1YzYyNTcsImV4cCI6MTYwMTA5ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94						
Respuestas							
Status	Cuerpo de la respuesta						

200	<pre>{   "votes": [],   "_id": "5accd673c4caec0004cfdb7e",   "end_date": "2018-12-04T00:00:00.000Z",   "created_at": "2018-04-10T15:21:23.606Z",   "updated_at": "2018-04-10T15:21:23.606Z" }</pre>
404	No existe la tarea
404	No se ha encontrado la votación
500	Error al encontrar la tarea
500	Error al conseguir la votación

Tabla 38. Consultar votación

POLL		
Tipo de petición	PUT	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbccaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBep-YzSJwVuvy94

Cuerpo de la petición	{ "end_date": "13/04/2018" }
Respuestas	
Status	Cuerpo de la respuesta
204	Actualizado
400	El campo end_date es obligatorio
404	No existe la tarea
404	Ya existe la votación
500	Error al crear la votación

Tabla 39. Actualizar votación

POLL		
Tipo de petición	DELETE	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbccaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiE1MjMyNzg2NTcsImV4cCI6MTYwMTA0ODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
201	Borrado	



404	No existe la tarea
500	Error del servidor

Tabla 40. Borrar votación

### 13.3.10. Vote

VOTE		
Tipo de petición	POST	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll/votes/	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbccaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e/votes	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxYWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTA2OjY1Nz0udQ01HfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
Cuerpo de la petición	{ "value": "11", "comment": "considero que es considerablemente difícil" }	
Respuestas		
Status	Cuerpo de la respuesta	



		5DvN708PsVtMFBEP- YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
200	<pre>[   {     "votes": [       {         "_id": "5accde37f06f740004ab35a3",         "value": 11,         "user": "5acb6155be9c3d000432e2ee",         "comment": "considero que es considerablemente difícil",         "created_at": "2018-04-10T15:54:31.008Z",         "updated_at": "2018-04-10T15:54:31.008Z"       }     ],     "_id": "5accd673c4caec0004cfdb7e",     "end_date": "2019-01-01T00:00:00.000Z",     "created_at": "2018-04-10T15:21:23.606Z",     "updated_at": "2018-04-10T15:44:40.182Z"   } ]</pre>	
404	No se encuentra la votación	
500	No se ha localizado la votación	

Tabla 42. Obtener votos

VOTE	
Tipo de petición	GET
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll/votes/:idVote

Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4ae1f78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbcaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e/votes/5accde37f06f740004ab35a3	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0eXNzZmVtMTYwMTA0ODY1N30uamR1LHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
Respuestas		
Status	Cuerpo de la respuesta	
200	<pre>{   "_id": "5accde37f06f740004ab35a3",   "value": 11,   "user": { ...<sup>3</sup> },   "comment": "considero que es considerablemente difícil",   "created_at": "2018-04-10T15:54:31.008Z",   "updated_at": "2018-04-10T15:54:31.008Z" }</pre>	
404	No existe la votación	
404	No se ha encontrado el voto	
500	Error del servidor	

Tabla 43. Obtener voto concreto

VOTE
------

<sup>3</sup> Información del usuario resumida.

Tipo de petición	PUT	
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com	
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll/votes/:idVote	
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbccaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e/votes/5accde37f06f740004ab35a3	
Cabeceras de la petición	Nombre	Valor
	Content-type	Application/json
	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM5Nzg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEp-YzSJwVuvy94
Cuerpo de la petición	<pre>{   "value": "11",   "comment": "considero que es considerablemente difícil" }</pre>	
Respuestas		
Status	Cuerpo de la respuesta	
204	Actualizado	
400	El campo value es obligatorio y ha de ser un entero	
404	No se ha encontrado el voto	
404	No existe la votación	
500	Error al crear el voto	

Tabla 44. Modificar tarea

VOTE							
Tipo de petición	DELETE						
URL Base	https://sergiojuliogu-tfg-2018.herokuapp.com						
Ruta	/users/:username/projects/:idProject/sprints/:idSprint/status/:idStatus/tasks/:idTask/polls/:idPoll/votes/:idVote						
Ejemplo	https://sergiojuliogu-tfg-2018.herokuapp.com/users/newUserDoc/projects/5acb8b1b4152940004f54e25/sprints/5acbb03c75e4aelf78ee816c/status/5acbc4122708f4204c4d7ae0/tasks/5acbccaa52109bc2090ca8ea9/polls/5accd673c4caec0004cfdb7e/votes/5accde37f06f740004ab35a3						
Cabeceras de la petición	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Content-type</td> <td>Application/json</td> </tr> <tr> <td>Authorization</td> <td>eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0IiwiaWF0IjoiYnZg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94</td> </tr> </tbody> </table>	Nombre	Valor	Content-type	Application/json	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0IiwiaWF0IjoiYnZg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94
	Nombre	Valor					
	Content-type	Application/json					
Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1YWNiNjE1NWJlOWMzZDAwMDQzMmUyZWUiLCJpYXQiOiJlMjM0IiwiaWF0IjoiYnZg2NTcsImV4cCI6MTYwMTAzODY1N30.jDQlHfVNxAENf1p5DvN708PsVtMFBEP-YzSJwVuvy94						
Respuestas							
Status	Cuerpo de la respuesta						
201	Borrado						
404	No existe la votación						
404	No se ha encontrado el voto						
500	Error del servidor						

Tabla 45. Borrar voto

## 13.2. Modelo de datos

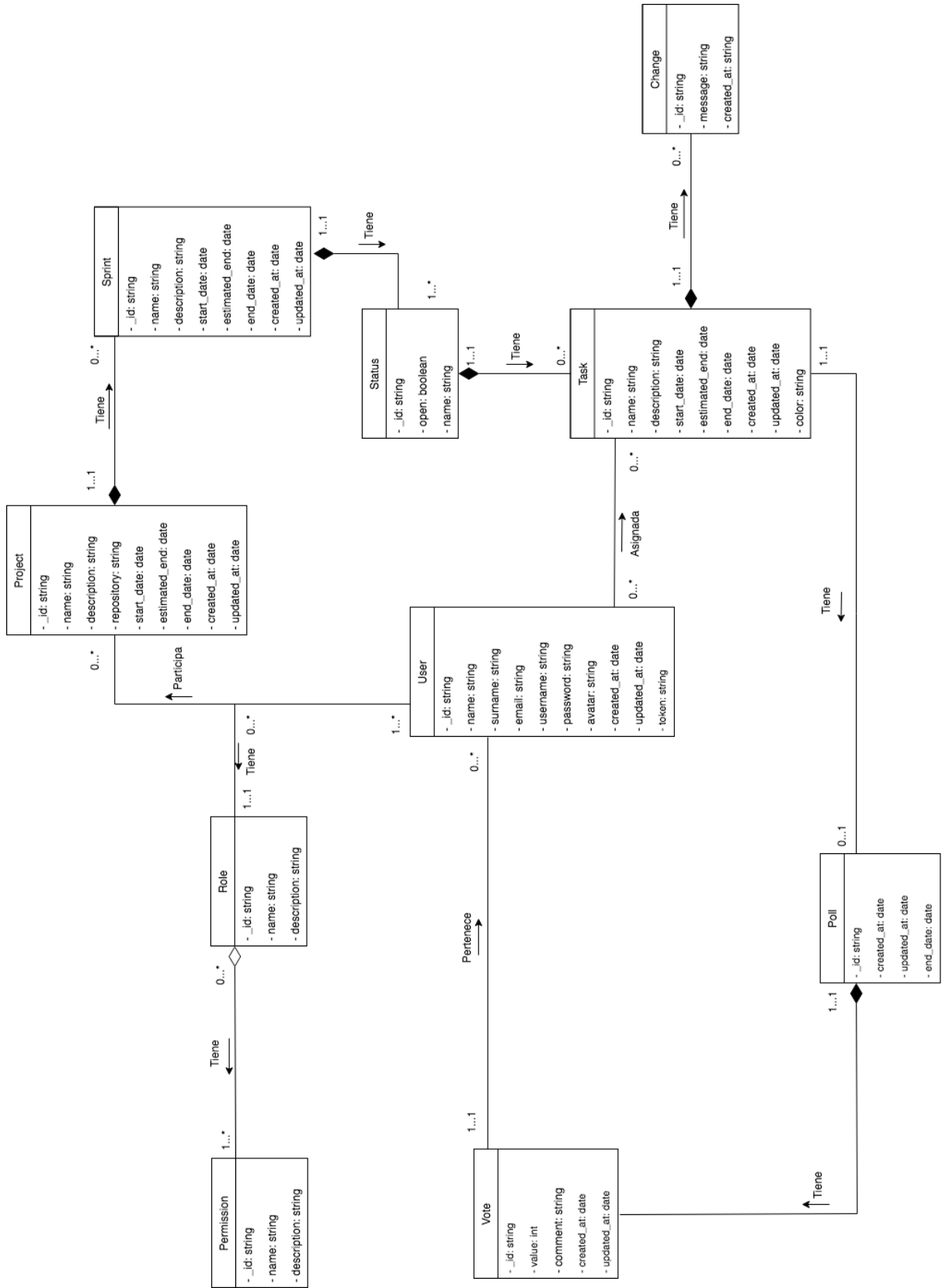


Diagrama 1. Datos.

### 13.3. Diagramas de caso de uso

#### 13.3.1. Rol

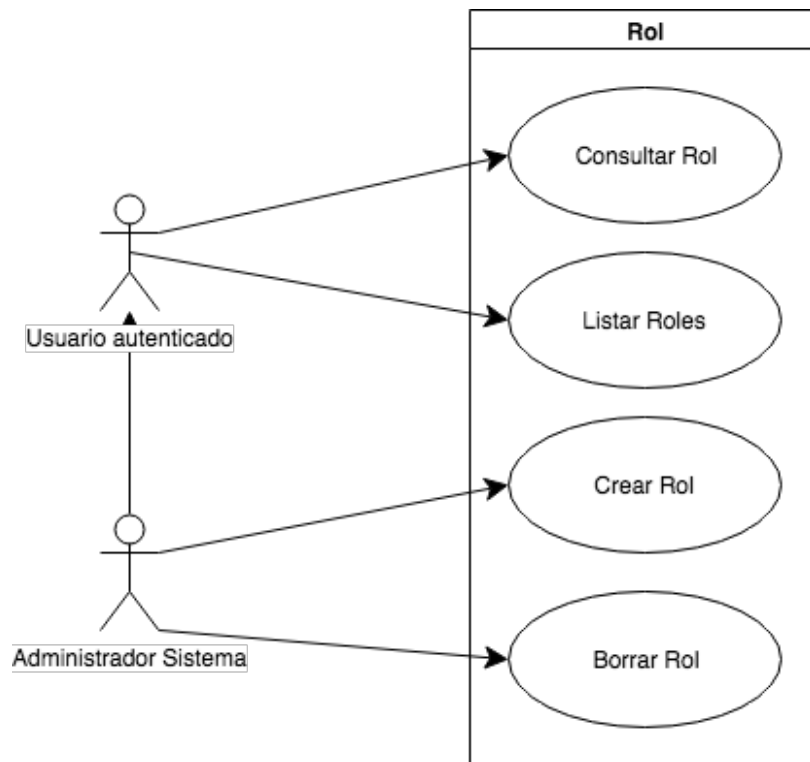


Diagrama 2. Caso de uso - Rol

#### 13.3.2. Permiso

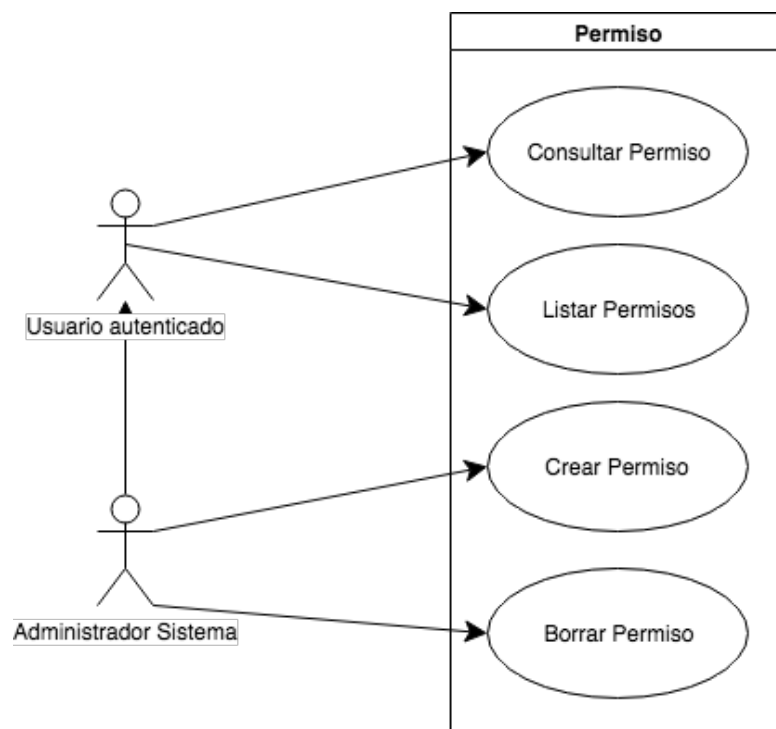


Diagrama 3. Caso de Uso - Permiso



### 13.3.3. Usuario

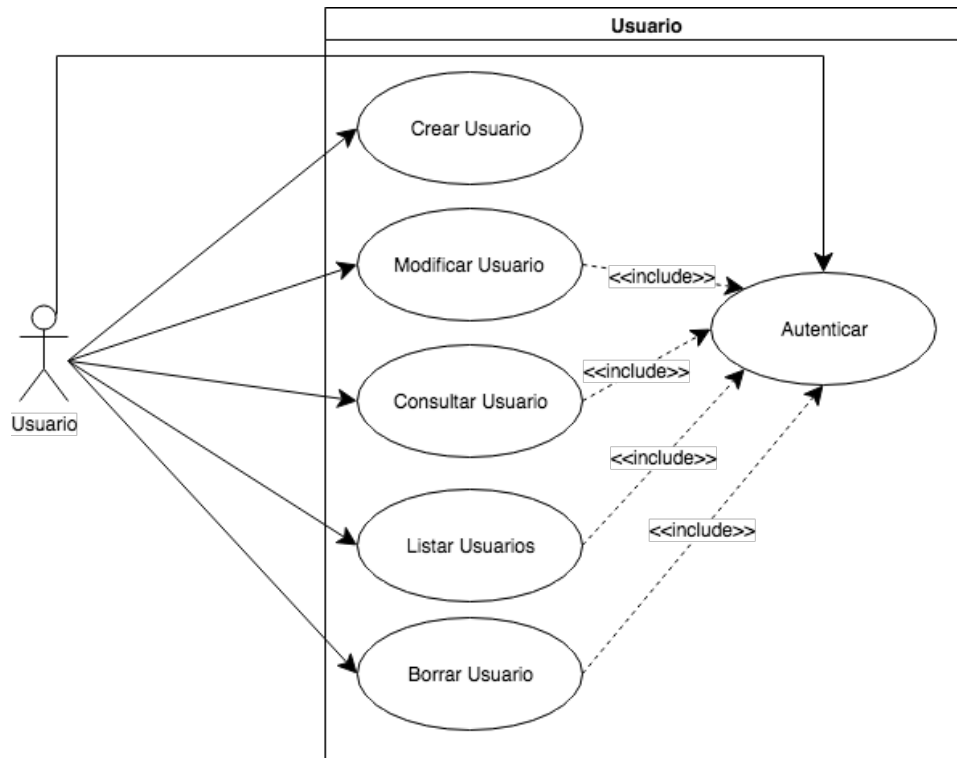


Diagrama 4. Caso de Uso - Usuario

### 13.3.4. Proyecto

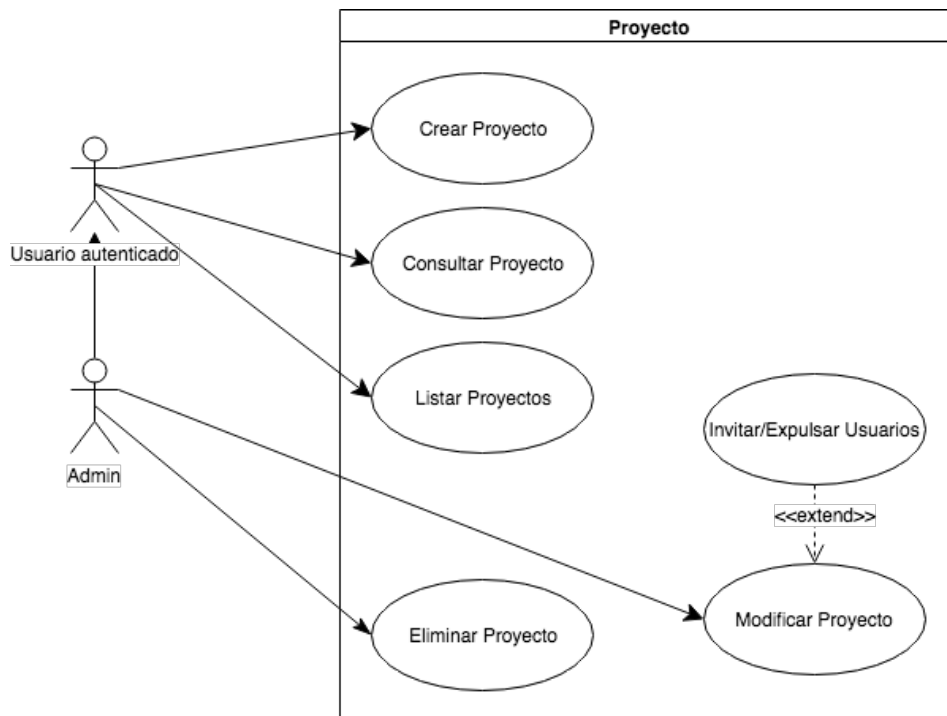


Diagrama 5. Caso de Uso - Proyecto

### 13.3.5. Sprint

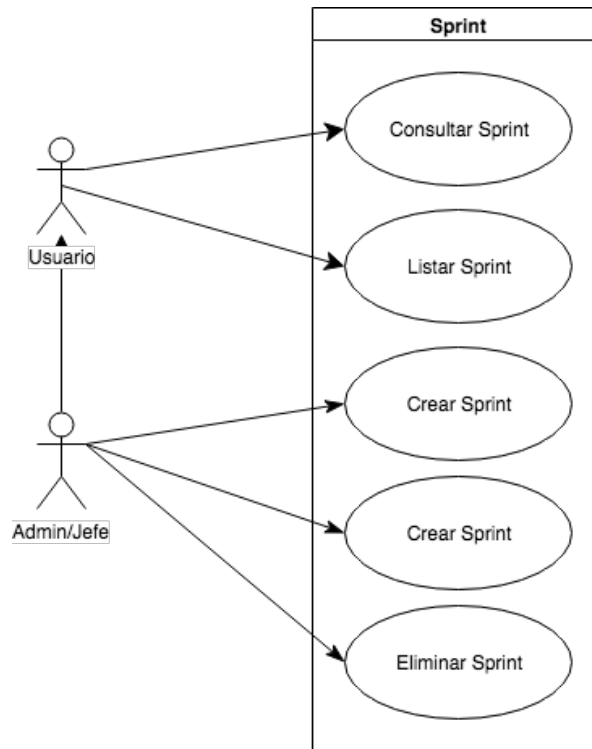


Diagrama 6. Caso de Uso - Sprint

### 13.3.6. Estado

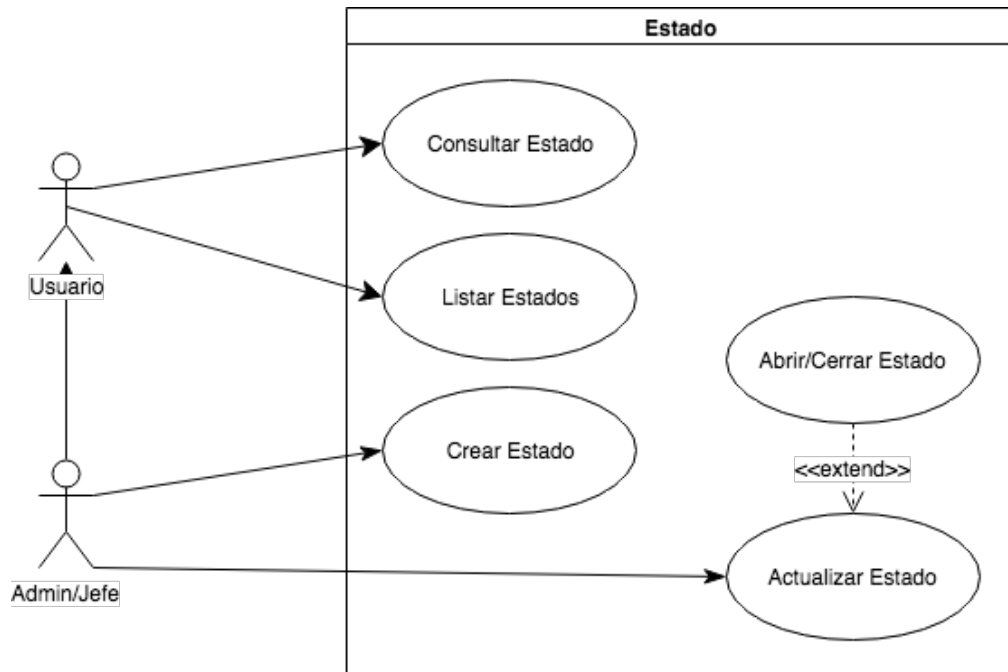


Diagrama 7. Caso de uso - Estado

### 13.3.7. Tarea

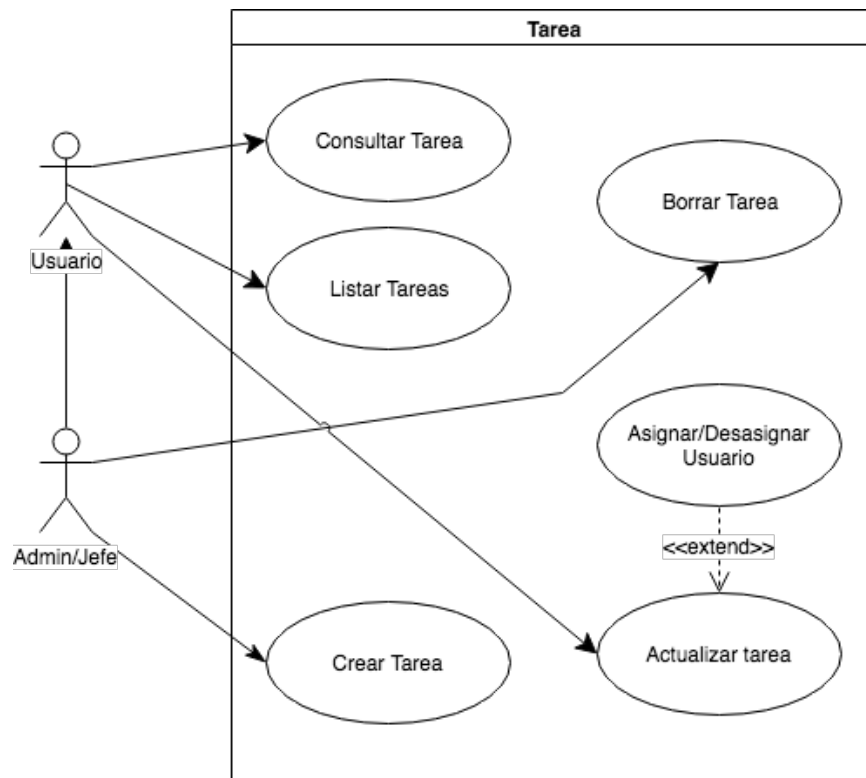


Diagrama 8. Caso de Uso - Tarea

### 13.3.8. Votación

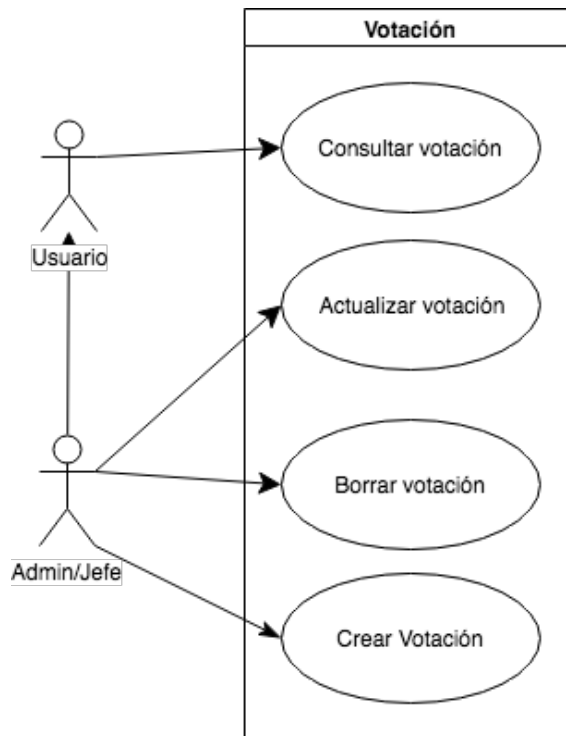


Diagrama 9. Caso de Uso - votación

13.3.9. Voto

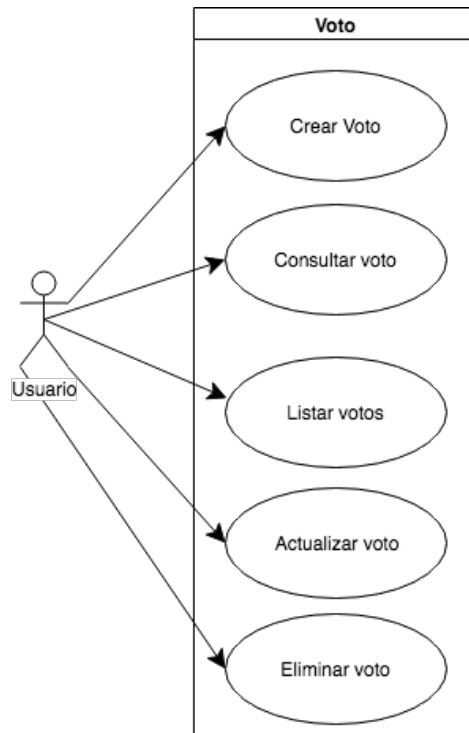


Diagrama 10. Caso de Uso - Voto

13.3.10. Cambio

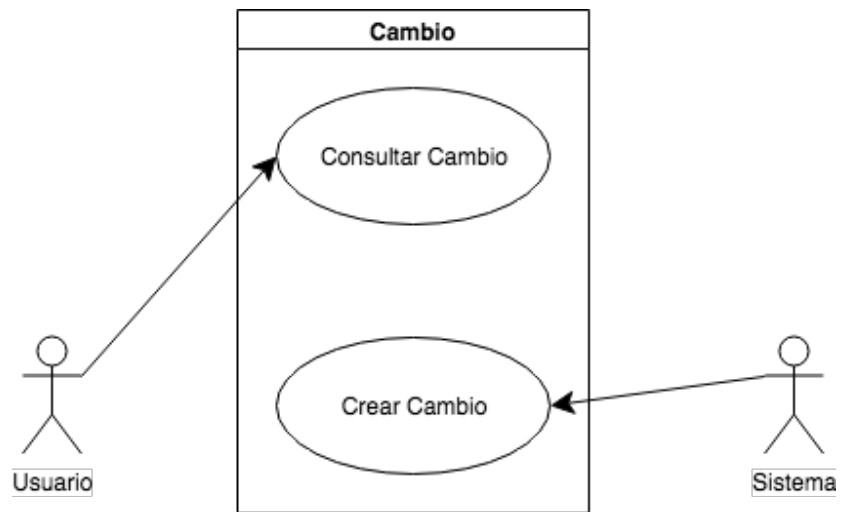


Diagrama 11. Caso de Uso - Cambio