

TEEM-ACTAS

APLICACION WEB DE ACTAS COLABORATIVAS PARA
REUNIONES Y ASAMBLEAS

MARMOL PEREZ, JUAN MANUEL
MARTIN RUIZ, LUIS FERNANDO

Grado en Ingeniería Informática
Facultad de Informática



UNIVERSIDAD
COMPLUTENSE
MADRID

Trabajo de Fin de Grado
Madrid, Febrero 2018

Directores:
Samer Hassan Collado y Antonio Tenorio Fornés

*Dedicado a
nuestras familias*

Autorización de difusión y uso

Autorizo a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos y mencionando expresamente a su autor y tanto la propia memoria como el código, la documentación y el software desarrollado.

Juan Manuel Mármol Perez
Luis Fernando Martín Ruiz

Madrid, Febrero 2018



Copyright by Juan Manuel Mármol Perez and Luis Fernando Martín Ruiz, released under the license Creative Commons Attribution Share-Alike International 4.0 available at:

<https://creativecommons.org/licenses/by-sa/4.0/>

Agradecimientos

En primer lugar queremos dar gracias a los directores de nuestro proyecto Samer Hassan y Antonio Tenorio por darnos la oportunidad de realizar este proyecto, confiar en nosotros y enseñarnos que con esfuerzo y dedicación se puede conseguir lo que uno se propone.

Gracias a los que estuvieron pero se fueron ya que hicieron que valorásemos lo que teníamos antes de empezar, nuestra amistad y compañerismo el uno con el otro. Gracias por enseñarnos que nosotros solos nos valemos para hacer lo que nos proponíamos.

Gracias al equipo de Teem y SwellRT que han apoyado el proyecto dándonos alternativas para complementar éste proyecto y ayudarnos en lo que hemos necesitado.

Gracias a Julia Viejo por dar siempre todo lo que tiene sin esperar recibir nada a cambio. Faltan personas como tú en el planeta.

Yo, Juan Manuel Marmól, quería dar gracias a mi compañero Luis por todo lo que hemos pasado juntos, por el esfuerzo incommensurable y por apoyarme y dar siempre un poquito más cuando se ha necesitado. Sin ti no habría sido posible.

Yo, Luis Fernando Martín, agradezco a mi compañero Juan Manuel Marmól por el impulso y el ánimo que me ha dado para realizar este trabajo, gracias a tu empeño y sacrificio hemos podido llegar a completar nuestros objetivos.

Índice general

Resumen	XI
Abstract	XV
1. Introducción	3
1.1. Motivación y objetivos	3
1.2. Estructura del documento	4
2. Estado del Arte	7
2.1. Blogger y WordPress	9
2.2. Mindmeister	9
2.3. MeetingBurner, WebEx	10
2.4. Fuze Meeting	10
2.5. Twiddla	11
2.6. Google Docs	11
3. Metodologías y Tecnologías	13
3.1. Metodologías	13
3.1.1. E-learning	13
3.1.2. Diseño guiado por objetivos	14
3.1.3. Scrum	15
3.1.4. Pair Programming	16
3.2. Tecnologías	18
3.2.1. HTML 5	18
3.2.2. JavaScript	18
3.2.3. SASS	18
3.2.4. AngularJS	19
3.2.5. Docker	20
3.2.6. Cordova	21
3.2.7. Travis	21
3.2.8. Git	21
3.2.9. NodeJs	21
3.2.10. Gestores de paquetes	22

3.2.11. Material Design	23
3.2.12. Gulp	23
3.2.13. Teem	23
3.2.14. L ^A T _E X	23
3.3. Herramientas	23
3.3.1. VMWare	23
3.3.2. Skype	24
3.3.3. WhatsApp	24
3.3.4. ShareLatex	24
3.3.5. Google Drive	24
3.3.6. Google Docs	24
3.3.7. Hangouts	24
3.3.8. Atom	25
3.3.9. GNU/Linux	25
3.3.10. GitHub	25
3.3.11. Pencil	25
3.3.12. SwellRT	25
3.3.13. Teem	28
4. Diseño de la aplicación	29
4.1. Introducción	29
4.2. Diseño Guiado por Objetivos	29
4.2.1. Investigación	30
4.2.2. Hipótesis de personas	31
4.2.3. Entrevistas	33
4.2.4. Modelado y Definición de Personas	36
4.2.5. Definición de Escenarios y Requisitos	37
4.2.6. Framework de diseño	39
4.2.7. Verificación. Principios de diseño	48
5. Desarrollo e implementación	53
5.1. Introducción	53
5.2. Desarrollo de Teem-Actas	54
5.2.1. Puntos del día	55
5.2.2. Asignación de Tareas	58
5.2.3. Votaciones	61
5.2.4. Turnos de palabra	65
6. Producto final:	71
6.1. Prototipo funcional	71
6.1.1. Puntos del día	71
6.1.2. Asignación de Tareas	74
6.1.3. Turnos de palabra	78

<i>ÍNDICE GENERAL</i>	IX
6.1.4. Añadir una votación	83
6.2. Evaluación con usuarios	85
6.2.1. Objetivos de la evaluación	86
6.2.2. Preguntas de la evaluación	86
6.2.3. Requisitos para los participantes de la evaluación	86
6.2.4. Requisitos para las pruebas de la evaluación	87
6.2.5. Lista de tareas a realizar	87
6.3. Pruebas con usuarios	88
6.3.1. Prueba 1	88
6.3.2. Prueba 2	90
6.3.3. Prueba 3	93
6.3.4. Prueba 4	95
7. Resultados y Trabajo Futuro	97
7.1. Introducción	97
7.2. Reparto de Trabajo entre los integrantes del grupo	97
7.3. Trabajo Futuro	98
Conclusiones	101
Conclusions	105

Resumen

En la actualidad, **Internet** [7] es la mayor y más versátil herramienta de la que se dispone a todos los niveles. Miles de millones de personas acceden a Internet para visitar páginas, buscar información sobre cualquier tema y para realizar tareas como trabajar a través de aplicaciones web, gestionar y compartir información a través de redes sociales y todo lo que se nos pueda ocurrir.

A medida que Internet se ha ido convirtiendo en la herramienta más potente de nuestra época, toda la sociedad se ha ido adaptando a Internet en sus diversas facetas. Es raro encontrar a gente que no utilice un móvil conectado a Internet y con aplicaciones descargadas, una persona que no tenga un ordenador en casa o un trabajador que se sirva de la tecnología y no utilice a diario Internet para satisfacer alguna de las necesidades que se puede encontrar en su día a día.

A este respecto, el desarrollo de las tecnologías relacionadas con el formato web y en particular las **TIC** (Tecnologías de la Información y la Comunicación) han visto cómo se acrecentaba su uso y demanda de una forma casi exponencial. A día de hoy, lo normal en la sociedad es estar informado a cada minuto, comunicarse con personas en el otro extremo del planeta o trabajar entre varias personas que están en países o lugares distintos al mismo tiempo.

Gracias a ésta necesidad de dar solución a lo que la sociedad y las personas requieren nacieron proyectos que a día de hoy son casi imprescindibles para muchas personas, como por ejemplo, redes sociales como **Facebook** o **Twitter** las cuales mantienen una actividad incesante de millones y millones de personas compartiendo información a tiempo real en todas las partes del mundo. Otras, menos demandadas aunque no menos utilizadas ni menos útiles, como **Google Docs** (y, en general, todo el paquete de ofimática que ofrece Google), el cual nos permite trabajar a tiempo real entre varias personas en un mismo documento o compartir archivos entre grupos

de personas que tienen algo en común.

Por otro lado, hay tareas que no se ven tan afectadas con éste cambio en la sociedad y se siguen llevando a cabo como se hacía anteriormente o simplemente no se han desarrollado lo suficiente como para crear una cantidad de herramientas tan amplia como se ha hecho para otras. No hay que dejar de tener en cuenta que los cambios se adaptan a las necesidades de grandes colectivos de personas que buscan o solicitan nuevas formas de llevar a cabo sus quehaceres. Nuestro proyecto se centra en una de esas tareas la cual creemos no se ha terminado de desarrollar en toda su plenitud en el aspecto tecnológico y cuyo uso sigue siendo altamente utilizado por millones de personas.

Tras tantos avances tecnológicos se necesitó redefinir o redimensionar algunos términos con el fin de abarcar todas las facetas de los recursos de los que disponemos hoy en día. Uno de los términos que nos concierne directamente es el concepto del **procomún** (o bien comunal), cuya definición en **Wikipedia** a día de hoy dice así:

“Por los comunes o “el procomún” o bien comunal se entiende aquellos bienes, factores productivos o recursos, procesos o cosas (ya sean materiales o de carácter intangible) cuyo beneficio, posesión o derechos de explotación pertenecen a un grupo o a una comunidad determinada. El grupo en cuestión puede ser extenso, por ejemplo, todos los individuos, o los habitantes de algún país, región, ciudad o pueblo, etc.- o restringido, como por ejemplo, una familia o algunos miembros de alguna familia, o grupo de personas establecido para un propósito específico (por ejemplo, una cooperativa o sociedad anónima). Los bienes, recursos, procesos o cosas que en la actualidad pueden ser considerados como parte del procomún comprenden desde bienes públicos generales (libres) y físicos (tales como el mar o el aire) a “bienes abstractos” (tales como la defensa o seguridad nacional) o el conocimiento en general o instancias específicas de tal: datos u otros elementos de información.”

“Tendremos sociedad de la información. Eso seguro. Nuestra única decisión ahora es si esa sociedad de la información será libre o feudal. La tendencia apunta hacia una sociedad feudal. (...) No hay forma de hablar de la sociedad de la Información a menos que uno hable también del grado de información y cultura que sería libre.” (Lessig, 2004)

Con éste concepto en la cabeza podemos encontrar proyectos que apoyan firmemente algunas de sus ideas como es **P2Pvalue** [10]. Proyecto impulsado por personas como Michel Bauwens, Vasilis Niaros o nuestros dos directores del proyecto, Samer Hassan y Anto-

nio Tenorio Fornés entre muchos otros. P2Pvalue adopta un enfoque multidisciplinario: desarrollo de software de código abierto (UCM), análisis y simulación de redes sociales (UniS, UCM), sociología (UniS, UMIL), ciencias políticas y gobierno en línea (UAB), diseño de leyes e interfaces (CNRS) , innovación económica (UMIL, P2PF, UAB) y prácticas de CBPP (P2PF, UAB).

Éste proyecto desemboca en una gran cantidad de áreas de investigación y diversas ramas del proyecto inicial. Una de ellas es **GRASIA** [5] (Grupo de investigación en Aplicaciones Sociales e Interdisciplinarias basadas en Agentes) cuyo principal objetivo es ayudar a las comunidades a realizar su potencial a través de la ingeniería de sistemas socio-técnicos.

A partir de P2Pvalue, GRASIA desarrolló los dos proyectos que más relación tienen con el nuestro. Éstos son **Teem** [62] y **SwellRT** [14]. Ambos son proyectos de carácter descentralizado y de código abierto enfocados a proyectos colaborativos. SwellRT interactúa con Teem para proveerle de un editor de textos en tiempo real para un grupo de colaboradores. Ambos están pensados para ser utilizados por todo tipo de usuarios y entidades de intereses colaborativos.

Nuestro proyecto, basado en estas dos tecnologías, se plantea como una solución tecnológica, usable y de código abierto cuyo objetivo principal es el de facilitar la toma de actas de cualquier entidad colaborativa. Al estar basado en Teem conseguimos un entorno descentralizado en el que los usuarios pueden abrir proyectos de forma independiente para realizar una toma de actas en torno a una aplicación la cual permite ser visualizada desde fuera sin necesidad de crear un usuario registrándose. De ésta forma se pretende incluir a usuarios menos familiarizados con éste tipo de tecnologías y además, proveer de un registro abierto y de fácil acceso de las actividades y tareas debatidos en cualquier reunión de cualquier entidad. De ésta manera impulsamos a entidades como ONG's o comunidades que pretenden darse a conocer a que los posibles futuros socios puedan visualizar y seguir los distintos temas e intereses que se tratan a diario en las reuniones de las mismas. Otra ventaja de estar basado en Teem radica en la facilidad de incluir nuevos miembros a las comunidades y a los proyectos. Esto permite la creación rápida de un nuevo grupo de usuarios que pretenden tomar actas en reuniones en las que tratan temas en común, además de la inclusión de nuevos miembros a dichos grupos de una manera rápida y con la posibilidad de visualizar anteriores actas generadas.

Gracias a utilizar Teem y, por consiguiente, SwellRT tenemos un entorno en el cual todos los miembros pueden seguir el acta y ser partícipes de él a tiempo real mientras está teniendo lugar la reunión, independientemente de si la persona que lo está siguiendo está presente en el lugar de la reunión o no.

Motivaciones

Motivados por las pocas soluciones tecnológicas de las que actualmente disponemos para una tarea tan básica y recurrente en todos los estratos de la sociedad, tanto empresas multinacionales como start-ups que apenas llevan unos meses en funcionamiento; ONG's mundialmente conocidas o pequeñas comunidades de vecinos que no encuentran un medio viable para que todos sus residentes vean y validen el acta en la que se aprueba pintar la fachada del edificio. Con todo esto y con las grandes posibilidades que las tecnologías anteriormente mencionadas nos proporcionaban nos decidimos a realizar éste proyecto. Ésta aplicación como no podía ser de otra manera intenta poner remedio a lo ya descrito de una manera descentralizada, de software libre y con tendencias para impulsar el procomún.

Objetivos

Los objetivos de éste proyecto son:

- Aprender y experimentar con tecnologías actuales y distintas a las que estamos acostumbrados como **HTML5** [24] o **AngularJS** [18] en un lenguaje el cual nos es relativamente desconocido a nivel de implementación como **JavaScript**.
- Promover las ideas del procomún, el software libre y la descentralización a partir de una aplicación sencilla y usable para ayudar a comunidades colaborativas a superar las barreras que siempre nos encontramos a la hora de tomar actas y difundirlos de una manera eficaz.
- Facilitar y ayudar a la toma de actas de una manera efectiva y satisfactoria para todos los usuarios posibles.
- Sentar las bases para una gran diversidad de trabajos futuros, ya que es una tecnología innovadora y muy puntera, con gran cantidad de posibles aplicaciones.

Palabras clave: Comunidades colaborativas, tiempo real, toma de actas, Teem , SwellRT, Código abierto, Reuniones, AngularJS ,Aplicación Web.

Abstract

Nowadays, the **internet** [7] is the most versatile tool we have at all levels. Thousands of millions of people access the internet to visit websites, look up any kind of data, work with web applications, share information through the social media and anything we can imagine. As the internet has become the most powerful tool of our time, the whole society has been consequently adapting to the internet in all of its aspects. It is unusual to find someone who doesn't use an on-line connected phone with applications, or someone who doesn't have a computer at home, or someone who doesn't use the internet to satisfy his or her daily needs.

On this matter, the development of web-related technologies and especially the use and demand of **ICT** (Information and Communication Technologies) has grown exponentially. Today it's normal to be informed every minute and to communicate with people on the other side of the globe, or even to work with people living in different countries and places.

Thanks to the need to satisfy everything that society and people demand, a lot of social media projects were created, such as **Facebook** or **Twitter**. Today these projects are essential for many people and keep an endless exchange of information between millions of people that share information in real time from every corner of the world. There are others less ambitious but not less useful, such as **Google Docs** (and the whole office software powered by Google), which allows different users to work in real time in the same document or to share files between groups of people with something in common.

On the other hand, there are tasks that haven't been affected by this change on the society and still are being carried out like before, or they haven't been developed enough to create an amount of tools as large as for other tasks. We can't forget that changes adapt to the need of big groups of people who demand new ways to carry out their jobs. Our project is focused in one of those tasks that, in our view, hasn't been fully developed in a technological aspect, and

which is used by millions of people.

After so many technological advances, there was a need to redefine some terms in order to include all the aspects of the resources we have today. One of the terms that concerns directly is the concept of “**commons**”, defined by **Wikipedia**:

“The commons is the cultural and natural resources accessible to all members of a society, including natural materials such as air, water, and a habitable earth. These resources are held in common, not owned privately. Commons can also be understood as natural resources that groups of people (communities, user groups) manage for individual and collective benefit. Characteristically, this involves a variety of informal norms and values (social practice) employed for a governance mechanism.”

“There will be an information society. That’s for sure. Now we just have to decide if that society will be free or feudal. (...) There’s no other way to talk about information society, unless you also talk about the level of freedom on information and culture” (Lessig, 2004)

Thinking about this concept we can find projects based on some of those ideas, such as **P2Pvalue** [10]. This project is powered by people like Michel Bauwens, Vasilis Niaros or our project directors, Samer Hassan and Antonio Tenorio Fornés, among others. P2Pvalue has a multidisciplinary approach: development of open source software (UCM), analysis and simulation of social media (UniS, UCM), sociology (UniS, UMIL), political science and on-line government (UAB), law and interface design (CNRS), economic innovation (UMIL, P2PF, UAB) and practice of CBPP (P2PF, UAB).

This project results in a large amount of investigation areas and derivations of the initial project. One of them is **GRASIA** [5] (in Spanish, Grupo de investigación en Aplicaciones Sociales e Interdisciplinarias basadas en Agentes), whose main objective is to help communities to develop their potential through socio-technical systems engineering.

From P2Pvalue, there have been two projects that have been developed from GRASIA, and which are highly related to our project. They are **Teem** [62] and **SwellRT** [14]. Both are decentralised and open source projects focused on collaborative work. SwellRT interacts with Teem in order to provide it with a text editor in real time for a group of collaborators. Both are designed to be used by any kind of users and entities with collaborative interests.

Our project, based on these two technologies, is designed to be a useful and open source technological solution whose main objective is to take minutes of any meeting of a collaborative entity. Since

it is based on Teem, we get a decentralised environment in which users can open independent projects in order to take minutes on an application which can be viewed by anyone without signing up as a user. This is how we try to include users who are not familiar with this kind of technology and, besides, we provide any entity with an application to easily access to the tasks of any meeting. That is how we encourage entities like NGOs and communities that want to make known, for their potential future partners can view and follow different issues and topics that are discussed in daily meetings. Another advantage of the fact that it is based on Teem is the ease to include new members to communities and projects. This allows a fast creation of new groups of users who want to take minutes in meetings about common topics, apart from the inclusion of new members to those groups easily and the chance to view previous records.

Thanks to Teem and therefore, SwellRT, we have an environment in which every member can follow the record and participate in real time as the meeting goes by, regardless of the presence of the person on the place of the meeting.

Motivations

We have been motivated by the few technological solutions that we have at our disposal to such a simple task, which is so recurring in every social class, from multinational companies to start-ups that have just been created, from world-known NGOs to small neighbour associations that still haven't found a proper way to show and validate their records by every member of the group, for example, to approve of the painting of a façade.

With all of this and the great opportunities that the previously mentioned technologies brought us in the past, we decided to launch this project. This application, as it was bound to be, tries to bring help in the mentioned situations in a decentralised way, thanks to open source software and our intention to promote the commons.

Objectives

The objectives of this project are:

- To learn and experiment with current technologies, different from the ones we are used to, such as **HTML5** or **AngularJS** in a language that is rather unknown for us in terms of implementation, as **JavaScript**.

- To promote the idea of commons, open source software and decentralisation with a simple and usable tool, to help collaborative communities to overcome any limitation they can find when taking minutes of a meeting and share them efficiently.
- To ease and help as many users as possible with meeting records.
- To set the bases of a large diversity of future jobs, since it is an innovative and leading technology with a great amount of potential applications.

Keywords: Collaborative communities, Real-Time, take minutes, Teem, SweelRt, Open Source, Meetings, AngularJS, Web Application.

Capítulo 1

Introducción

1.1. Motivación y objetivos

Actualmente vivimos en una sociedad que cada vez está más digitalizada y globalizada.

El término **procomún** [45] poco a poco se ha hecho eco, donde las **comunidades colaborativas** que forman el procomún sobre un determinado tema y se dan a conocer utilizando herramientas que nos ofrece Internet ya sea, bien a través de redes sociales como Twitter o Facebook, o bien a través de medios de información como WordPress o Blogger.

La idea principal del procomún es hacer visible estas comunidades colaborativas y para ello se necesitan herramientas que puedan ayudarles a darles la visibilidad para poder **comunicarse, participar y colaborar**. Dentro de este marco, cuando los miembros de estas comunidades se reúnen para debatir sobre un tema, ya sea online o presencialmente, necesitan dejar constancia de los temas que van a tratar en esas reuniones para que aquellas personas que no han asistido a la reunión o para futuros miembros de la comunidad puedan ver los temas tratados y acuerdos a los que se haya llegado en estas reuniones.

Hay muchas herramientas en Internet para que estas comunidades puedan reunirse y compartir cosas (Skype, Google Docs, etc) pero no hay muchas aplicaciones para facilitar la **toma de actas** y, además, que sean totalmente gratuitas. Por ello optamos por hacer una aplicación para que estas comunidades colaborativas puedan dejar constancia a través de un **acta** donde quede reflejado los temas que se han tratado, las tareas a realizar y los acuerdos a los que se han llegado en una reunión de alguna de estas comunidades colaborativas.

A éste propósito se ha creado una herramienta que facilita la

organización de entidades colaborativas ayudando al registro y difusión de sus actividades a través de internet basada en la plataforma **Teem** [62] focalizando especialmente la toma de actas en reuniones. En concreto hemos modificado Teem añadiéndole la funcionalidad necesaria (como la preparación de los puntos del día antes de una reunión, un sistema de votaciones, el registro de turnos de palabra y la posibilidad de asignar una tarea a una persona en concreto entre las más destacadas) para que se convierta en una herramienta de toma de actas en tiempo real así como para mantener un registro público y duradero de cada entrada.

1.2. Estructura del documento

En este documento del Trabajo de Fin de Grado se ha querido mostrar los aspectos más característicos y destacados en el proceso de desarrollo del mismo. Para ello, se tuvo en cuenta que investigar y reutilizar recursos que nos han sido muy útiles para el desarrollo de la aplicación que se describe al inicio de este documento, **Teem-actas**, por lo que en esta documentación se busca reflejar claramente las partes que se han reutilizado de **Teem** [62] y qué nuevas funcionalidades se han añadido.

A lo largo de este documento el lector se encontrará la siguiente estructura:

- **Capítulo 2 - Estado del arte** En esta sección se hace un análisis del problema que queremos solucionar además de las características de las posibles soluciones software que hacen uso de tecnologías que permiten la colaboración en tiempo real y específicamente para la toma de actas.
- **Capítulo 3 - Metodologías y Tecnologías de la aplicación** Se repasan brevemente las metodologías empleadas y las tecnologías que se han utilizado durante el desarrollo del Trabajo de Fin de Grado.
- **Capítulo 4 - Diseño de la aplicación** Se exponen los aspectos de la investigación previa y el diseño de la aplicación **Teem-actas** utilizando el **Diseño Guiado por Objetivos (DGO)** [49].
- **Capítulo 5 - Desarrollo e implementación** Se detallan qué aspectos que se han reutilizado de **Teem** para el proyecto, así como las nuevas funcionalidades que se han añadido.

- **Capítulo 6 - Producto final** En ésta parte se expone cuál fue el resultado final de la implementación así como las respectivas evaluaciones con usuarios.
- **Capítulo 7 - Resultados y Trabajo Futuro** Se repasan los resultados obtenidos tras la realización del proyecto así como los siguientes pasos a realizar con el objetivo de mejorar o cambiar Teem-actas. Cabe señalar que en esta sección se expondrá el **reparto de trabajo** realizado por los integrantes de este proyecto.
- **Capítulo 8 - Conclusiones** En éste ultimo apartado se sacan una serie de conclusiones una vez terminado el Trabajo de Fin de Grado y analizados los resultados de las pruebas con usuarios.

Capítulo 2

Estado del Arte



Figura 2.1: Imagen representativa del procomún

Actualmente poco a poco se ha ido haciendo eco de la palabra **procomún** o **procomún colaborativo**, un término que aún muchas personas no tienen claro su significado e incluso el impacto social y económico que puede llegar a tener. Esto nos lleva a hacer una primera pregunta.

¿Qué es el procomún?

Como se menciona en la introducción de este proyecto (1), el **procomún** es un modelo de gobernanza para el bien común, es decir, es aquella manera de producir y gestionar en comunidad bienes y recursos, tangibles e intangibles, que nos pertenecen a todos los miembros de una comunidad o dicho de otra forma, que no pertenecen a nadie. Este concepto ha vuelto tener un repercusión pública gracias al **software libre** [34] y al movimiento **open source** [17], en 2009 se hizo más eco gracias a **Elinor Ostrom** con su premio Nobel de Economía [48] y por

sus aportaciones al gobierno de los bienes comunes.

El procomún lo forman aquellas cosas que heredamos y creamos conjuntamente para legar a las generaciones futuras. Hay una gran diversidad de bienes naturales, culturales o sociales, que son claros ejemplos de algo que puede ser procomún: la biodiversidad, las semillas, Internet, el agua potable, un espacio público, etc. Bienes que muchas veces sólo percibimos cuando están amenazados o en peligro de desaparición o privatización, pero a los que todos tenemos el derecho de poder acceder.

El procomún es creado y recreado, conectado y reconectado. Nace de la interacción entre los miembros de una comunidad, comunidades distribuidas que se **reúnen** alrededor de un tema o de un problema. Por esta razón no existe procomún sin comunidad, y viceversa. Por tanto, **una de las consecuencias principales del procomún es hacer visibles esas comunidades emergentes** de personas afectadas (darles el tiempo, la experiencia, la tecnología, los medios, la palabra...) con la voluntad de construir entre todos un mundo más justo para un mundo común.

Ante este nuevo paradigma, estas comunidades buscan nuevas formas de hacerse eco y darse a conocer al mundo para que más personas puedan unirse a estas comunidades. En un mundo más globalizado y sujeto a los cambios que las nuevas tecnologías propician, dan la oportunidad a que estas comunidades puedan tener una mayor audiencia y difusión, y así nacen nuevas herramientas para poder realizar estos trabajos colaborativos a través de una de las mayores herramientas que existen actualmente: **Internet**.

Hay infinidad de herramientas para poder trabajar de forma colaborativa online. Estas herramientas nos permiten desde compartir archivos y poder editarlos en tiempo real, hasta convocar y asistir a **reuniones** online a través de videoconferencias. En los siguientes apartados de este capítulo se describirán algunas herramientas que **las comunidades del procomún** pueden utilizar para **comunicarse, debatir y colaborar**.

Como se ha señalado a lo largo de ésta sección, uno de los requisitos para que el procomún exista es que los miembros de una comunidad se **reúnan e interactúen**. **Pero si en estas reuniones no queda constancia de los temas que se han debatido o de los acuerdos a los que se ha llegado** además del motivo del mismo, futuros miembros de la comunidad o aquellos miembros que

no hayan podido asistir a alguna de las reuniones no podrán saber por qué razón se han llegado a estos acuerdos y los temas que se han debatido en esas reuniones, y es precisamente aquí donde entra en juego el trabajo que estamos exponiendo: **Teem-actas**.

A continuación se expondrán herramientas y aplicaciones que pueden utilizarse como alternativa para la toma de actas que a su vez se han tomado como referencia y alternativa a Teem-Actas, analizando sus beneficios y deficiencias para realizar una aplicación que incorpore las ventajas de éstas y que pretende solucionar sus inconvenientes.

2.1. Blogger y WordPress

Blogger y **WordPress** son herramientas para la creación de blogs de Google de forma sencilla y eficaz. Al ser gratuito las comunidades del procomún pueden compartir de forma abierta las actas que redacten con los temas o problemas que intenta solventar para así poder llegar a un mayor público con el objetivo de concienciar e invitar a participar a mas gente.



Figura 2.2: Pantalla de Blogger

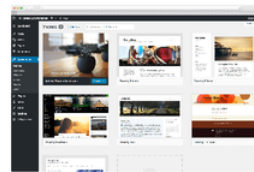


Figura 2.3: Pantalla de WordPress

2.2. Mindmeister

Mindmeister es una aplicación para elaborar mapas mentales en línea y de forma colaborativa, útiles para hacer lluvias de ideas o estructurar los ejes del trabajo. Permite tomar notas, planear proyectos y realizar muchas otras tareas creativas. Además permite insertar multimedia, gestionar y asignar tareas y convertirlos en una presentación o en un documento imprimible.



Figura 2.4: Mapa de Mindmeister

2.3. MeetingBurner, WebEx

MeetingBurner y **WebEx** son plataformas que ofrecen la posibilidad de llevar a cabo reuniones remotamente a un grupo de personas considerable. Estos servicios tienen versiones de prueba sin coste alguno pero con funcionalidad limitada. Son sus servicios de pago los que dan una mayor cobertura a las necesidades que precise cada grupo de trabajo.



Figura 2.5: Pantalla de MeetingBurner



Figura 2.6: Pantalla de WebEx

2.4. Fuze Meeting

Fuze Meeting es una herramienta que sirve para realizar sesiones de colaboración online con videoconferencia, compartiendo contenidos en alta definición y desde cualquier PC o dispositivo móvil en la nube. Las principales ventajas de usar esta herramienta son la movilidad, pues permite convocar o asistir a reuniones online de forma ilimitada, desde cualquier PC, smartphone o tablet, además de poder compartir contenidos durante la sesión, como archivos de audio, vídeo e imágenes, keynotes, archivos PDF y Powerpoint, e incluso compartir el escritorio e interactuar a través de una pizarra, y todo ello en tiempo real.



Figura 2.7: Captura de pantalla de Fuze Meeting

2.5. Twiddla

Twiddla es una herramienta web de colaboración que permite a los usuarios subir o anotar casi cualquier cosa online así como trabajar en grupo sobre un fondo blanco, un espacio similar a una pizarra. Incorpora también herramientas de comunicación para que los colaboradores puedan chatear e incluso hablar por teléfono.



Figura 2.8: Pizarra de Twiddla

2.6. Google Docs

Google Docs es un conjunto de aplicaciones que ofrece un pack de ofimática integradas en la web que permiten la edición de cada archivo en tiempo real así como su difusión entre usuarios de Google a través de invitaciones o externos mediante links especiales. Este conjunto de herramientas incluye toda la potencia que la red social de Google ofrece al estar integradas en su servicio de almacenamiento Google Drive.

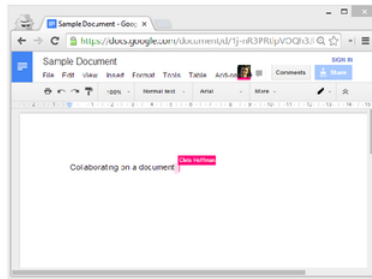


Figura 2.9: Captura de pantalla de Google Docs

Capítulo 3

Metodologías y Tecnologías

3.1. Metodologías

3.1.1. E-learning

E-learning (electronic learning) [46] es un proceso de aprendizaje o enseñanza a distancia por internet por el cual el/la alumno/a se encarga de autogestionar sus recursos para el aprendizaje además de obtener apoyo de compañeros/as y profesores/as. Esta modalidad de enseñanza/aprendizaje ha permitido acceder al conocimiento a toda persona que tenga conexión a internet e interés a desarrollar sus conocimientos en alguna materia. Sus principales características son:

- No hay barreras espaciales ni temporales: al ser a través de internet no es necesaria la asistencia presencial a ningún centro ni estar disponible a una hora concreta ya que el sistema lo permite.
- Reducción de costos: al no ser necesario ningún desplazamiento, material didáctico físico o centro de estudios los costes se reducen al tiempo empleado en el aprendizaje/enseñanza y al material para acceder a los contenidos a través de la red.
- Acceso instantáneo: cualquier alumno/a puede acceder al material didáctico cuando y donde desee siempre y cuando disponga de una conexión a la red.
- Permite la comunicación asíncrona entre el personal docente y los alumnos/as o entre ellos mismos.
- Cambia el rol del personal docente al pasar de ser comunicadores de conocimiento a orientadores adquiriendo más el rol de tutores.

- El alumno/a tiene más responsabilidad sobre su auto-aprendizaje al tener que auto-organizarse para adquirir los conocimientos requeridos.
- Proporciona apoyo a otras modalidades de formación (el campus virtual por ejemplo).
- Mayor razonamiento crítico y menor miedo al error. La falta de presencialidad, la posibilidad de usar medios asíncronos y una cierta sensación inicial de anonimato facilitan la capacidad reflexiva y crítica.
- Facilidad de inclusión de contenidos audiovisuales.

Durante el desarrollo de este proyecto hemos utilizado varias de las características del e-learning para poder satisfacer nuestras necesidades comunicativas y organizativas. Gracias a todas las plataformas donde hemos compartido nuestro contenido y nos hemos comunicado/organizado sin necesidad de estar presencialmente cada vez que hemos trabajado en el proyecto. También hemos recibido el apoyo y orientación necesarios por parte de nuestros tutores de manera síncrona o asíncrona según lo necesitáramos (aunque hemos mantenido reuniones presenciales regularmente para el seguimiento del proyecto).

3.1.2. Diseño guiado por objetivos

Diseño Guiado por Objetivos (DGO) [53] es una de las metodologías implementadas en el diseño centrado en el usuario y que tiene en cuenta el proceso de ingeniería de usabilidad. Este método se centra en el diseño de interacciones acorde a los objetivos que el usuario/a busca al utilizar un sistema. Esta metodología responde a algunas de las preguntas más importantes que surgen durante la definición y el diseño de un sistema interactivo:

- ¿Quiénes son mis usuarios?
- ¿Qué tratan de conseguir mis usuarios?
- ¿Qué experiencias son gratificantes para mis usuarios?
- ¿Cómo debería comportarse mi sistema?
- ¿Qué aspecto debería tomar mi sistema?
- ¿Cómo interactuarán los usuarios con mi sistema?
- ¿Cómo he de organizar de manera efectiva las funcionalidades que ofrece mi sistema?

- ¿Cómo se presenta mi sistema ante un usuario por primera vez?
- ¿Cómo se comporta mi sistema ante los problemas que el usuario puede encontrar?
- ¿Cómo ayuda mi sistema a los usuarios novatos a satisfacer sus necesidades?
- ¿Es mi producto suficientemente interesante para usuarios expertos?

Esta metodología se estructura en varias fases:

- **Investigación:** fase en la que se realiza un estudio sobre los usuarios potenciales y cuáles son sus necesidades, para identificar patrones de comportamiento que sugerirán los objetivos y motivaciones de los usuarios.
- **Modelado:** en esta fase se crearán *personas ficticias* que representarán al grupo de usuarios al que irá dirigido nuestra aplicación. Más adelante este tipo de personas serán las que nos proporcionen algún tipo de *feedback*.
- **Definición de requisitos:** en esta fase se utilizarán los datos obtenidos en la fase anterior para identificar los escenarios de contexto e identificar los requisitos o necesidades de los usuarios.
- **Definición del framework de diseño:** en esta última fase, se definirá el comportamiento general del sistema además de la parte visual, ya sea en papel o como es en este caso, sobre una herramienta de prototipado.

3.1.3. Scrum

Durante todo el desarrollo del proyecto seguimos una metodología Scrum [12]. Scrum es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el “Scrum Master”, que procura facilitar la aplicación de scrum y gestionar cambios, el Product Owner, que representa a los stakeholders (interesados externos o internos), y el Team (equipo) que ejecuta el desarrollo y demás elementos relacionados con él. Durante cada sprint, un período entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar (PBI, Product Backlog Item). El Scrum está caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

Durante los primeros meses del proyecto hicimos sprints de un mes aproximadamente y a medida que avanzamos en el desarrollo y en el proyecto fuimos acortando los sprints hasta llegar a sprints de 2 semanas. Entre el equipo se tenían reuniones cada dos o tres días al principio del proyecto y cada día o cada dos días hacia las fases finales del mismo. En las fases finales del desarrollo el equipo decidió hacer “Pair Programming” para agilizar y trabajar la implementación de una manera más versátil y más coordinada.

3.1.4. Pair Programming

La Programación en Pareja (o Pair Programming) [39] requiere que dos programadores participen en un esfuerzo combinado de desarrollo en un sitio de trabajo. Cada miembro realiza una acción que el otro no está haciendo actualmente: Mientras que uno codifica las pruebas de unidades el otro piensa en la clase que satisfará la prueba, por ejemplo. La persona que está haciendo la codificación se le da el nombre de controlador mientras que a la persona que está dirigiendo se le llama el navegador. Se sugiere a menudo para que a los dos socios cambien de papeles por lo menos cada media hora

o después de que se haga una prueba de unidad. La programación en pareja se enfoca en las siguientes ventajas, ordenadas de mayor a menor.

- Más Disciplina. Emparejando correctamente es más probable que hagan “lo que se debe hacer” en lugar de tomar largos descansos.
- Mejor código. Emparejando similares es menos probable producir malos diseños ya que su inmersión tiende a diseñar con mayor calidad.
- Flujo de trabajo constante. El emparejamiento produce un flujo de trabajo distinto al trabajar solo. En pareja el flujo de trabajo se recupera más rápidamente: un programador pregunta al otro “¿por dónde quedamos?”. Las parejas son más resistentes a las interrupciones ya que un desarrollador se ocupa de la interrupción mientras el otro continúa trabajando.
- Múltiples desarrolladores contribuyen al diseño. Si las parejas rotan con frecuencia en el proyecto significa que más personas están involucradas con una característica en particular. Esto ayuda a crear mejores soluciones, especialmente cuando una pareja no puede resolver un problema difícil.
- Moral mejorada. La programación en parejas es más agradable para algunos programadores, que programar solos.
- Propiedad colectiva del código. Cuando el proyecto se hace en parejas, y las parejas se rotan con frecuencia, todos tienen un conocimiento del código base.
- Enseñanza. Todos, hasta los novatos, poseen conocimientos que los otros no. La programación en pareja es una forma amena de compartir conocimientos.
- Cohesión de equipo. La gente se familiariza más rápidamente cuando programa en pareja. La programación en pareja puede animar el sentimiento de equipo.
- Pocas interrupciones. La gente es más renuente a interrumpir a una pareja que a una persona que trabaja sola.
- Menos estaciones de trabajo. Ya que dos personas van a trabajar en una estación de trabajo, se requieren menos estaciones de trabajo, y las estaciones extras pueden ser ocupadas para otros propósitos.

3.2. Tecnologías

En este apartado vamos a comentar las tecnologías empleadas durante el desarrollo del proyecto.

3.2.1. HTML 5

HTML 5 (HyperText Markup Language, versión 5) [24] es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) [27] o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

3.2.2. JavaScript

JavaScript es una implementación del estándar **ECMA-262** [19] también conocido como ECMAScript desarrollada por la empresa Netscape. ECMAScript es un lenguaje de programación orientado a objetos para llevar a cabo procesos de computación y manipular objetos complejos en un entorno local. Originalmente fue concebido como un lenguaje de scripting que se ejecutaba en el navegador para contrarrestar la baja velocidad de internet al poder ejecutar tareas complejas a través del navegador en un equipo local manteniendo la arquitectura cliente-servidor. Con el tiempo se convirtió en un lenguaje de propósito general y se puede utilizar en cualquier tipo de arquitectura.

3.2.3. SASS

SASS [15] es un lenguaje para crear estilos. Sass es un metalenguaje de Hojas de Estilo en Cascada (CSS) [2]. Presenta grandes ventajas frente al CSS convencional:

- La creación de variables, que permite por ejemplo realizar cambios de color o fuentes de texto de una manera rápida, cómoda y consistente.
- SASS permite asignar a un conjunto de reglas una etiqueta lo que facilita duplicar estilos para varios elementos sin necesidad de duplicar código
- Soporta anidamiento lógico lo que facilita tanto visualmente la lectura del código, como la asignación de estilos de subelementos.

3.2.4. AngularJS

AngularJS [18] es un framework de desarrollo para JavaScript creado por Google. La finalidad de AngularJS es facilitarnos el desarrollo de aplicaciones web SPA [20] y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima. Otro propósito que tiene Angular es la separación completa entre capa del front-end y el back-end en una aplicación web. A continuación explicaremos una visión conceptual de como funciona AngularJS. Primero se explicarán algunos conceptos que se utilizan en Angular para posteriormente ver cómo interaccionan estos componentes:

- **Scope**, es un objeto que hace referencia al modelo de la aplicación. Es un contexto de ejecución para las expresiones. El scope está organizado en una estructura jerárquica similar a la estructura DOM [3] de una aplicación. El scope puede **ver expresiones y propagar eventos**.
- **Templates**, representan la parte visual de la aplicación. Están escritos en HTML, contienen elementos y atributos específicos de AngularJS. Por otro lado, AngularJS combina el template con la información del **model** y el **controller** para representar la vista dinámica que ve el usuario en el navegador.
- **Directives**, sirven para definir nuevos apartados de código html, como botones o crear un **footer** completo. También sirve para crear cosas más complejas que incluyan funciones o variables dinámicas definidas directamente en el DOM.
- **Controller**, son los objetos que permiten desarrollar la lógica de la aplicación, enlazando el ámbito (`$scope`) con la vista (template) y permite tener un control total de los datos, es decir, es el encargado de gestionar los eventos.

- **Services**, son objetos **singleton** [13], inyectables por **Dependency Injection**. Es donde definimos la lógica de negocio de la aplicación, con el objetivo de que sea reutilizable e independiente de las vistas.

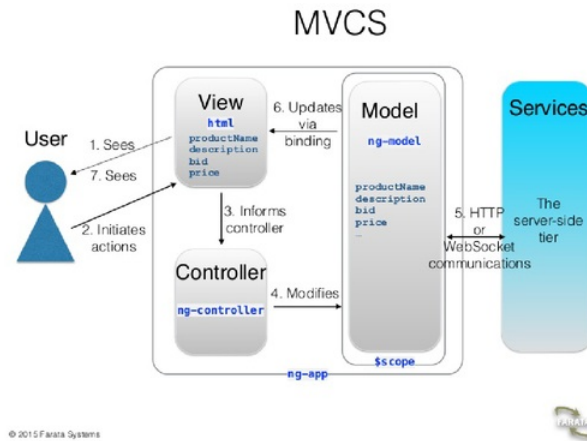


Figura 3.1: Representa la interacción entre los template-controller-service

Las principales ventajas de utilizar AngularJS como framework de desarrollo son:

- **Reusabilidad**. Permite crear componentes (directivas) fácilmente reutilizables.
- **Testeo**. Al tener componentes aislados, podemos testear su comportamiento de manera independiente.
- **Inyección de dependencias**. Si se necesita hacer uso de un servicio, se inyecta en el controlador que va hacer uso de ese servicio.

3.2.5. Docker

Docker [40] es un software de contenedores los cuales se definen entornos multiplataforma que permiten ejecutar software sin necesidad de instalar ningún componente adicional en el equipo ni configuración alguna para que el desarrollador/a se pueda centrar en la programación. Es crucial en este proyecto debido a su complejidad y tamaño ahorrándonos horas de trabajo instalando dependencias y configuraciones. En concreto utilizamos la utilidad **Docker**

Compose [29] que permite crear aplicaciones que utilicen varios contenedores a la vez.

3.2.6. Cordova

Cordova [51] es un conjunto de API's de diferentes dispositivos móviles que permiten encapsular una aplicación web en una aplicación para un dispositivo móvil brindando acceso a los recursos de los dispositivos móviles desde la aplicación web.

3.2.7. Travis

Travis [23] es un servicio de integración continua. Hemos utilizado este servicio para poder subir a un servidor nuestro desarrollo y que los usuarios finales puedan probarlo.

3.2.8. Git

Git [22] es un sistema de control de versiones. Se caracteriza por:

- Facilitar el trabajo colaborativo entre distintos desarrolladores, los cuales pueden editar el mismo archivo, o generar versiones distintas del mismo archivo, y todos los cambios serán reflejados en el documento final.
- Se pueden trabajar en varios flujos de trabajo (branches), en los que los desarrolladores pueden realizar varias tareas al mismo tiempo y luego juntar las distintas “branches” para unificarlas en una sola rama (haciendo un merge), a la rama “Master”, que es la que se contendrá el código que se ejecutará en el servidor.
- También permite regresar a versiones anteriores de forma sencilla y rápida.

3.2.9. NodeJs

NodeJs [33] es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos. Las ventajas que nos ofrece utilizar esta librería son:

- NodeJs resuelve el problema de el cuello de botella en toda la arquitectura de aplicación Web (incluyendo el rendimiento del tráfico, la velocidad de procesador y la velocidad de memoria) que es el número de conexiones concurrente que puede manejar un servidor. Node resuelve este problema cambiando la forma en que se realiza una conexión con el servidor. En lugar de

generar un nuevo hilo de OS para cada conexión (y de asignarle la memoria acompañante), cada conexión dispara una ejecución de evento dentro del proceso del motor de Node. Node también afirma que nunca se quedará en punto muerto, porque no se permiten bloqueos y porque no se bloquea directamente para llamados E/S.

- Es especialmente bueno para aplicaciones de realtime, que necesitan mantener una conexión persistente entre el browser y el servidor.
- Está basado en eventos, así que toda la filosofía asíncrona que ya utilizamos con AJAX [60] en el cliente la podemos pasar al servidor.
- Permite utilizar el mismo lenguaje (javascript) tanto en el cliente como en el servidor.
- Muy buena gestión de paquetes gracias a NPM, muchas cosas que necesitábamos hacer, ya existía una librería/paquete en npm.
- Detrás de Node hay una Comunidad enorme documentando, haciendo tutoriales y creando nuevos módulos.
- Nos permite hacer en el servidor todo lo que necesitamos (acceso a ficheros, a bases de datos, conexiones de clientes..).

3.2.10. Gestores de paquetes

3.2.10.1. NPM

NPM [16] es el gestor de paquetes(del back-end) javascript de NODE.JS [33] por excelencia. Nos ayuda a descargar las dependencias que necesitamos en nuestro proyecto, estas dependencias resuelven problemas de cosas que se buscaban hacer (como abrir un fichero). Estos módulos los gestionamos a través del archivo package.json, que se encargará de descargar las librerías necesarias (al hacer ejecutar el comando 'npm install') para el correcto funcionamiento de la aplicación.

3.2.10.2. Bower

Bower [44] al igual que npm, es un gestor de paquetes javascript de NODE.JS. Este gestor se encarga de la parte visual principalmente, el "front-end".

3.2.11. Material Design

Material Design [57] es una normativa de diseño enfocado en la visualización del sistema operativo Android , además en la web y en cualquier plataforma. En nuestro proyecto lo iconos que nos ofrece Material Design Icons.

3.2.12. Gulp

Gulp [55] es una herramienta en forma de script en NodeJs, que nos ayuda a automatizar tareas comunes en el desarrollo de una aplicación, como puede ser: mover archivos de una carpeta a otra, eliminarlos, minificar código, sincronizar el navegador cuando modificamos código y muchas otras tareas.

3.2.13. Teem

Teem [62] es una herramienta web y una aplicación móvil para la colaboración en comunidades, colectivos y movimientos sociales, lo que facilita la integración de los recién llegados. Está construido dentro del proyecto P2Pvalue [10] con un enfoque en comunidades de producción entre iguales basadas en el uso común (CBPP) [?]. Es la herramienta sobre la que se ha desarrollado nuestra plataforma. Es un editor de texto en tiempo real que utiliza SwellRT [14] para gestionar la concurrencia de la edición del texto. Hemos utilizado esta herramienta porque es la que mejor se adaptaba a nuestras necesidades y nos ofrece muchas opciones para afrontar los problemas que buscamos resolver en la toma de actas. Se ha utilizado esta herramienta a lo largo del desarrollo, por ejemplo para tomar notas e hitos en las reuniones que hemos tenido a lo largo del curso.

3.2.14. L^AT_EX

LaTeX [54] es un lenguaje de marcas que permite la creación y edición de documentos y artículos científicos. Latex es software libre por lo que cualquiera tiene acceso a sus características. Toda la documentación de la memoria está escrita a partir de ésta tecnología.

3.3. Herramientas

3.3.1. VMWare

VMWare [50] es un software de virtualización que permite ejecutar instancias de sistemas operativos sobre una máquina host con un sistema operativo ya instalado. Durante el periodo de desarrollo

de este proyecto ha sido crucial ya que nos ha permitido tener una máquina virtual portable e igual para todo el equipo donde hemos trabajado durante el desarrollo.

3.3.2. Skype

Skype [47] es una herramienta comprada por Microsoft después de ser desarrollada en Estonia que permite realizar llamadas, interactuar mediante el chat y el envío de archivos a través de internet y en tiempo real. Esta aplicación nos ha permitido realizar reuniones a distancia entre los miembros del equipo cuando no hemos podido reunirnos presencialmente.

3.3.3. WhatsApp

WhatsApp [43] es una aplicación de mensajería instantánea que nos ha permitido organizarnos y comunicarnos a distancia.

3.3.4. ShareLatex

ShareLatex [56] es una plataforma que permite compartir, editar y compilar en tiempo real documentos latex. Hemos utilizado esta herramienta para el desarrollo de esta memoria.

3.3.5. Google Drive

Google Drive [38] es una plataforma de almacenamiento masivo en la nube desarrollada por Google que permite guardar y compartir contenido. Hemos utilizado esta herramienta para almacenar y compartir toda la información que hemos generado durante el transcurso del proyecto.

3.3.6. Google Docs

Google Docs [37] es un procesador de texto desarrollado por Google que permite la edición de texto rico en tiempo real así como su compartición. Hemos utilizado esta herramienta para redactar documentos varios del proyecto.

3.3.7. Hangouts

Hangouts [59] es una herramienta desarrollada por Google que permite realizar llamadas y videollamadas a través de internet así como un chat. Esta herramienta la hemos utilizado para llevar a cabo las reuniones con los tutores de este proyecto.

3.3.8. Atom

Atom [26] es un IDE de desarrollo que se ha utilizado en este proyecto para editar el código de la aplicación web.

3.3.9. GNU/Linux

GNU/Linux [35], también conocido como Linux, es un sistema operativo libre tipo Unix; multiplataforma, multiusuario y multitarea. El sistema es la combinación de varios proyectos, entre los cuales destacan GNU (encabezado por Richard Stallman y la Free Software Foundation) y el núcleo Linux (encabezado por Linus Torvalds). Su desarrollo es uno de los ejemplos más prominentes de software libre: todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera, bajo los términos de la GPL (Licencia Pública General de GNU) y otra serie de licencias libres. Todo el desarrollo y su posterior implementación ha sido soportado por sistemas Linux para hacer más fácil el desarrollo y poder desplegar el servidor con nuestra aplicación.

3.3.10. GitHub

GitHub [25] es la plataforma de repositorio remoto que se ha utilizado para gestionar las versiones de nuestro proyecto durante la fase de programación. Esta plataforma está basada en tecnología Git.

3.3.11. Pencil

Pencil [32] es una herramienta case [30] para la creación de diagramas y prototipos que utilizamos para crear nuestro primer prototipo de interacción que empleamos en las pruebas con usuarios.

3.3.12. SwellRT



Tiene su origen en el trabajo realizado dentro del equipo de investigación de **GRASIA** [5] en la Universidad Complutense de Madrid, dentro del proyecto de **P2Pvalue** [10] financiado por la Unión Europea. Actualmente la base del código está acogido por Apache Software Foundation, fundación destacada por su apoyo a proyectos de

código abierto. SwellRT además de ser el único software de colaboración en tiempo real para texto rico y objetos complejos distribuido y libre, proporciona un modelo de programación basado en objetos colaborativos, que pueden almacenar propiedades de tipos simple (String, Integer, etc.) así como textos enriquecido, objetos complejos y referencias a archivos. Estos objetos son parecidos a un **JSON** (JavaScript Object Notation) [8] que pueden ser compartidos por usuarios o grupos de usuarios para realizar cambios en tiempo real. Estos cambios se propagan y notifican en tiempo real a cualquier usuario conectado a dicho objeto. Para nuestro caso, SwellRT guarda estos objetos JSON en una base de datos no relacional (NoSQL) que es **MongoDB**. Este enfoque es el adecuado para utilizarlos en aplicaciones colaborativas basadas en documentos, como editores de texto, que en este caso utiliza tanto **Teem** como nuestro proyecto de **Toma de Actas**.

SwellRT se creó a partir de una bifurcación de Apache Wave [1], por lo que hereda parte de su arquitectura y tecnología. Se ha convertido en una plataforma de tipo back-end-as-a-service (BaaS). Sus principales características son:

- Es un **software de código abierto**.
- Está **desarrollado en Java**.
- Utiliza **Google Web Toolkit** (GWT) [41] con JSInterop [42] para generar **JavaScript API** reutilizando el mismo código fuente.
- Proporciona un **componente de editor de texto extensible y conectable** para Web que es compatible con anotaciones y widgets personalizados.
- El **almacenamiento de los datos en tiempo real** está basado en el Modelo de Transformaciones Operativas de Wave, por lo que es consistente.
- Está **diseñado para maximizar la interoperabilidad**, por lo que sigue el enfoque de federación similar al de Apache Wave, usando el protocolo de comunicación XMPP o Matrix.org [4]. El objetivo principal es apoyar la creación de aplicaciones que están federadas, es decir, que dependen de múltiples servidores interoperables y objetos compartidos en todos los servidores. Esto permite a las organizaciones controlar sus propios datos.



En resumen las principales características y **ventajas de utilizar SwellRT** son:

Algunos ejemplos de otras aplicaciones (aparte de Teem, ya mencionado anteriormente) que utilizan SwellRT son:

- Jetpad [52] es un editor de textos web colaborativo en tiempo real creado con la tecnología SwellRT y el framework Angular 2.
- Incubator-Wave [6] Aplicación principal de Apache Wave con colaboración en tiempo real que utiliza la API de JavaScript y la infraestructura de un servidor federado. Es un “fork” del proyecto de SwellRT.

3.3.13. Teem



Al igual que SwellRT (3.3.12), Teem nace dentro del proyecto de **P2Pvalue** [10]. Teem es una herramienta web y una aplicación móvil para colaborar en comunidades, colectivos y movimientos sociales y facilitar la integración de nuevos miembros en dichas comunidades. Teem utiliza SwellRT como editor de texto para que dichas comunidades compartan sus proyectos en tiempo real. Está desarrollado en AngularJS, además de utilizar mas tecnologías como (NodeJs, SASS, GULP, etc.) que mencionamos en la sección de “Tecnologías” (3.2). éstos son los principales servicios que nos ofrece Teem:

- Crear comunidades.
- Ofrece una visión global de las comunidades y la facilidad de poder participar en dichas comunidades.
- Poder invitar a participar a nuevos miembros en una comunidad.
- Un editor de texto para exponer los temas, tareas e información a todos los miembros de una comunidad.
- Poder añadir dentro del editor de texto propiedades como títulos, subrayados, enlaces a páginas, imágenes, etc.
- Añadir tareas a realizar por la comunidad.
- Al utilizar SwellRT, todo lo que comparten estas comunidades es en tiempo real.

Capítulo 4

Diseño de la aplicación

En este capítulo se expondrá el proceso que se llevó a cabo para detectar los requisitos y detectar la prioridad de cada uno de ellos para su posterior desarrollo en la fase de implementación.

4.1. Introducción

Con el fin de plasmar los conocimientos adquiridos a lo largo de nuestro aprendizaje en la carrera y con la intención de poder crear una aplicación usable y funcional, decidimos que para esta fase de diseño de la aplicación utilizaríamos una de las metodologías que aprendimos en la asignatura de Desarrollo de sistemas interactivos (DSI) : **Diseño Guiado por Objetivos (DGO)** [31]. Para la aplicación de esta metodología se llevaron a cabo las siguientes actividades:

- Investigación
- Modelo y Definición de Personas
- Definición de Escenarios
- Framework de diseño

4.2. Diseño Guiado por Objetivos

En las siguientes secciones se expondrá el proceso de de investigación previo que se realizó para determinar la funcionalidad y diseño gráfico de la aplicación **Teem-actas**.

En este capítulo se desarrollan las fases de la metodología del Diseño Guiado por Objetivos. En cada sección se especificarán los pasos a seguir durante su avance y el resultado final de cada una de ellas.

4.2.1. Investigación

En lo que se refiere al ámbito de la **toma de actas** en las reuniones de una asociación o de una asamblea, el encargado/a de la toma del acta suele realizar esta tarea apuntando todo lo que se debate y los acuerdos a los que se llega en las reuniones sobre un papel o bien utilizando un ordedanor en el que utiliza alguna herramienta (editor de texto, un documento word, Google Docs, etc) que le permita redactar el acta. Una vez que se ha terminado la reunión, el encargado del acta debe repasar todo o que ha apuntado a lo largo de la reunión y verificar no hay erratas para finalmente, o bien enviar el acta por correo electrónico, imprimiendo el documento o compartir el documento para que todos los miembros de la asociación o asistentes a la reunión puedan consultarlo.

Para dar cobertura a ésta situación, decidimos desarrollar una aplicación que facilite la **toma de un acta** para apuntar todo lo que se debate en una reunión, los acuerdos a los que se llega y poder compartir esta información con todos los miembros asistentes a la reunión así como a los que no pueden asistir a dichas reuniones. Además detectamos la necesidad de que en estas reuniones sería útil apuntar los **Puntos del día** que se van a debatir en una reunión, para poder verlos de forma clara pudiendo marcarlos de alguna forma como ya hablados o debatidos. También concluimos que sería útil poder registrar poder registrar en el documento de la toma de actas las votaciones con los resultados y los acuerdos a los que se han llegado en una reunión.

En conclusión, se buscó crear una aplicación que pudiera facilitar la **toma de acta** para que los miembros de una asociación pudieran compartir sus actas con miembros que no hayan podido asistir a la reunión o para futuras personas que quieran unirse a una asociación y quisieran ver los temas que se debaten en estas asociaciones.

SwellRT y Teem

Por otro lado, investigando posibles aplicaciones que pudieran resolver este problema, decidimos reutilizar la aplicación de **Teem** [62] que sirve, entre otras muchas cosas, para compartir proyectos entre comunidades colaborativas, para añadirle las funcionalidades que aportarán valor y utilidad a la hora de realizar la tarea de una toma de acta. Como **Teem** es una aplicación que permite compartir proyectos en tiempo real y de forma colaborativa, observamos que podríamos reutilizar esta aplicación ya que nos ofrecía: poder crear proyectos, dar de alta usuarios (seguridad), un editor de texto basado en SwellRT, un widget para poder dar de alta tareas a realizar

en proyectos, etc.

Por otro lado también se detectó la necesidad de poder asignar y moderar los turnos de palabra dentro de estas reuniones y que quedará constancia de las personas que han intervenido en las reuniones. Otra funcionalidad que se pensó en desarrollar es la de poder realizar un resumen de un acta, con los principales puntos del día, las votaciones con los acuerdos a los que se ha llegado y las tareas a realizar.

Con los principales objetivos de la aplicación y una vez analizado las principales funcionalidades de nuestra aplicación, procedimos a realizar los bocetos de los prototipo (Sección 4.2.6.4) con la herramienta Pencil [32].

4.2.2. Hipótesis de personas

En esta sección nos centramos en identificar a qué tipos de usuarios iría enfocada nuestra aplicación. En este caso, al ser un proyecto orientado a comunidades colaborativas, y dado que éstas comunidades están formadas por cualquier tipo de personas sin ninguna restricción, todos los perfiles tienen un nexo que es común a todos: los usuarios deben participar en proyectos colaborativos.

Lo primero que hicimos fue analizar los factores a tener en cuenta al crear un perfil, descartar aquellos factores que no influían en nuestros perfiles y hacer incapié en aquellos que sí.

Elementos que no influyen en nuestros perfiles:

- Sexo.
- Localización geográfica.
- Nivel económico.
- Nivel académico.

Elementos que influyen en nuestros perfiles:

- La edad es un factor muy importante debido a la brecha generacional y tecnológica actual. Por lo tanto las personas de menos edad tendrán más facilidades y las de mayor edad más dificultades de adaptación a la herramienta.
- Objetivos por los que usa el sistema: conseguir una solución para que la toma de actas sea más eficiente en un entorno colaborativo y descentralizado.

- Actividades que realizan: reuniones frecuentes para organizar distintas actividades y proyectos.
- Uso habitual de dispositivos electrónicos (tablets, ordenadores, móviles, etc.) en las actividades diarias tanto profesionales como personales.
- Se presupone un conocimiento nulo ya que asumimos que cualquier persona puede incorporarse a una actividad/proyecto y no haber utilizado antes una herramienta similar.

Teniendo en cuenta los factores anteriores hicimos una puesta en común para ver qué tipos de perfiles utilizaríamos a la hora de hacer las entrevistas. Éstas son algunas de las características deseadas:

- Edad: Entre 18 y 65 años.
- Sexo: Indiferente.
- Profesión: Indiferente.
- Aptitudes deseables: Trabajo colaborativo.
- Habilidades técnicas: Habitado a tomar acta en reuniones además de tener experiencia media en el uso de aplicaciones web.

Teniendo en cuenta todos estos factores, nos dimos cuenta que solo teníamos un tipo de perfil, que son aquellas personas que participan en proyectos colaborativos y que suelen asistir con bastante frecuencia a las reuniones y suelen **tomar acta** de los temas que se han tratado y de los acuerdos a los que se han llegado, y lo único que los diferencia es la edad debido a que personas de mayor edad están menos acostumbrados a utilizar herramientas tecnológicas. El siguiente paso a realizar fue el de buscar posibles candidatos a los que poder hacerles las entrevistas. Para ello pensamos en buscar a estas personas en distintas asociaciones y asambleas, lo cual nos llevó a entrevistar a:

- **Tesorero de AJE (Asociación Jóvenes Empresarios Madrid).**
- **Presidente de ASCII (Asociación Socio-Cultural de Ingenierías en Informática).**
- **Secretaría de una asociación de Filatelia.**
- **Miembro de una asamblea de barrio.**
- **Presidente del Club Rotary (Club de Jóvenes empresarios)**

4.2.3. Entrevistas

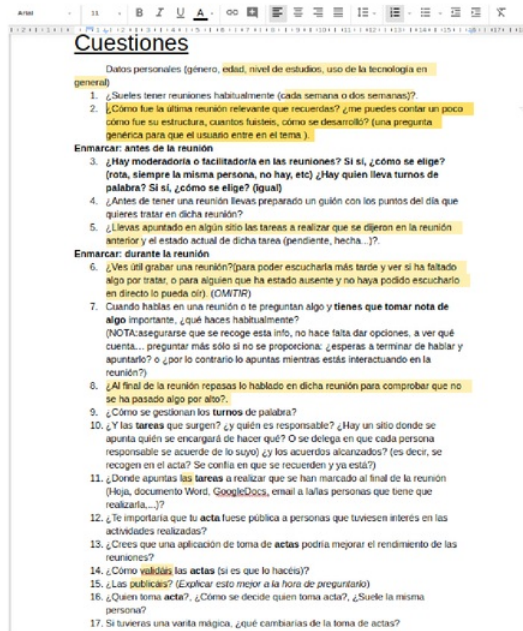


Figura 4.1: Preguntas de las entrevistas

En este apartado se recopilarán algunas de las entrevistas que se realizaron en la fase de investigación del diseño de la aplicación. Hicimos un total de cinco entrevistas: al tesorero de A.JE (Asociación Jóvenes Empresarios Madrid), al presidente de ASCII, una secretaria de una asociación de Filatelia, un miembro de una samblea de barrio (Asociación Socio-Cultural de Ingenierías en Informática) y por último al presidente del Club Rotary (Club de Jóvenes empresarios). Antes de realizar las entrevistas, preparamos un guión con una serie de preguntas que realizaríamos durante la sesión. Pensamos éstas preguntas para que los entrevistados nos contaran sus experiencias a la hora de asistir a reuniones y a la **toma de actas** con el fin de obtener la máximo de información posible para poder sacar una lista con los principales problemas y las necesidades que tenían a la hora de **tomar acta**. Éstas entrevistas tuvieron una duración aproximada de 5-10 minutos. A continuación se expondrá un resumen de cada una de las entrevistas que se realizaron en ésta fase.

Entrevista al Presidente de ASCII

Tuvimos la oportunidad de entrevistarnos con el presidente de ASCII, quién casualmente el día anterior a la entrevista tuvo una reunión con tres personas más para tratar varias cuestiones (no mencionó cuales). Nos contó que al ser tan solo tres personas la reunión fue muy ágil. Además llevaban preparados (apuntados en papel) **los puntos del día** que iban a ser tratados en la reunión y nos comentó que suele haber un encargado de **apuntar los puntos tratados**, para verificar al final de la reunión o en la reunión siguiente qué puntos del día hanbían sido tratados. Sobre los **turnos de palabra** nos informó que era él quien gestionaba los turnos de palabra y tomaba nota de quién levantaba la mano para pedir su turno. Respecto a si tienen un responsable de apuntar las tareas a realizar para la siguiente reunión, depende del tipo de reunión que tengan, cuando es una **reunión de junta** hay un responsable encargado de ello. Si la reunión es de **socios** se puede encargar cualquier persona y se establece un seguimiento de la persona encargada. Éstas tareas se apuntan mediante un borrador o en un libro de actas.

Por último nos comentó que las actas son **públicas** y que hay una persona encargada de elaborar un borrador del acta, para posteriormente enseñarsela al presidente, el cual validará el acta. A continuación se redacta el acta formal y se busca posibles fallos para que finalmente todos los miembros que han asistido a la reunión la validen.

Tesorero de AJE

En esta caso entrevistamos al tesorero de AJE, una persona mayor de unos 40 años, quien suele tener reuniones una vez al mes. Nos contó cómo fue su última reunión, en la que aprobaron el acta anterior y los miembros de la junta hablaron cada uno de sus responsabilidades. También nos comento que en esa reunión llevaron una lista con los **puntos del día**, los cuales se debatieron y que al final de la reunión suelen hacer una ronda de preguntas y respuestas. Por otro lado hay una persona encargada de realizar las funciones de **moderador**. También nos comentó que bastante útil poder **grabar** las reuniones para poder contrastar que los si los datos tomados en la reunión son correctos. Cabe descatcar que en su asociación los **turnos de palabra estaban predefinidos** previamente, aunque se pueden interrumpir, y al final de éstos turnos hay una ronda de turnos libres. Respecto a la asignación de tareas nos comentó que los apuntan en un **Google Docs** para poder hacer un seguimiento

de a qué personas se les ha asignado una tarea, aunque puntualizó que nadie los mira. En este caso las actas no son públicas, aunque cualquier miembro de la asociación puede pedir el acta, la cuál se envía por correo.

Miembro de una asamblea de barrio

En este caso hicimos la entrevista a un participante joven, de unos 25 años, en una asamblea de un barrio. Nos contó que solía asistir a las reuniones semanalmente. Este tipo de reuniones no eran tan organizadas, ya que nos informó de que los **puntos del día se publican el mismo día** y que se reparten fotocopias entre los asistentes, los cuales suelen ser una media de 20 personas. Las **personas encargadas de moderar y tomar actas son voluntarias**. Otro punto importante es que no suelen repasar los puntos del día porque el tiempo es limitado. Por otro lado, las tareas quedan reflejadas en el acta y se apunta el nombre del encargado de realizarla. Por último nos comento que todas sus actas son públicas.

Secretaría de una asociación de Filatelia

Esta entrevista fue realizada a una señora de unos 50 años, quien era la encargada de realizar la toma de actas en las reuniones de la asociación. Sus reuniones pueden variar dependiendo de los temas a tratar, pueden ser anuales, mensuales e incluso semanales. En este caso nos contó su experiencia en su última reunión, que fue una reunión de tipo anual, general para los socios antes de Navidad. El presidente hizo previamente una convocatoria formal y escrita donde exponía los temas a tratar. En este caso no hay un rol de moderador ya que se organizan bien, pues los **turnos de palabra los concede el presidente o la secretaria**. Si hay muchas personas que desean intervenir **la secretaria toma nota en papel y las concede según el orden de llegada**. Al igual que nos informaron en las otras entrevistas, también suelen llevar previamente los temas a tratar además de alguna otra información necesaria para cada reunión. Algo que cabe destacar en esta entrevista es que **revisan semanalmente las tareas a realizar**. Por último cabe señalar que les resulta muy difícil revisar al final de la reunión si se han hablado todos los puntos de día, porque disponen de un tiempo muy limitado ya que todos los miembros de la asociación quieren participar.

Presidente del Club Rotary

Por último hicimos la entrevista al presidente del Club Rotary y nos contó su experiencia a la hora de la toma de actas. Lo primero que hizo fue informarnos de su última reunión, en la cual al inicio se presentó a los invitados de la reunión, **se repasaron los puntos del día** de la reunión anterior, los responsables de los proyectos en curso realizaron un informe de avances y por último realizaron coloquios de temas variados. Hay un **secretario que es el encargado de realizar la toma del acta**. Algo significativo en ésta asociación es que en las reuniones se tiene un **moderador** que se elige anualmente, quien se encarga de varias tareas, entre ellas la de conceder los turnos de palabra en las reuniones y apuntar las tareas a realizar. Además dijo que los **puntos del día a tratar se definen en el mismo día. No suelen publicar las actas** porque en ocasiones se suelen tratar temas personales, por lo que solo **se envía por correo** a los miembros de la asociación.

4.2.4. Modelado y Definición de Personas

El objetivo de esta sección es modelar a **personas** que representan a un grupo de usuarios a los que va destinada ésta aplicación. Esto nos ayudará al diseño de la aplicación para que se adapte a las necesidades de éstas personas. Éstas personas son una representación ficticia pero desarrollada con gran detalle, que vendrá definida por los datos recogidos en la investigación previa de la fase anterior. Por ello se definió un tipo de persona que interactuará con nuestra aplicación. Esta persona será **secretario en la toma de actas** con una edad de 20 años en adelante:

- **Actividad:**
 - Estudia o trabaja
 - Pertenece a alguna asociación o forma parte de algún proyecto colaborativo.
 - Utiliza las redes sociales para comunicarse con otros colectivos, acudir a asambleas además de promover ideas relacionadas con el procomún o proyectos colaborativos.
 - Suele ofrecerse voluntario para la toma de actas.
 - Hace la tarea de moderador en las reuniones o asambleas.

Nombre: Francisco Manzano

Edad: 28
 Grado de conocimientos informáticos: Medio
 Sexo: Hombre
 Número de reuniones semanales: 1
 Profesión/situación laboral: Secretario en la toma de actas.
 Información personal: Francisco es un hombre comprometido con las entidades colaborativas. Actualmente está cursando su tercer Máster y como estudiante acude como a las reuniones de las Asociaciones Estudiantiles de las diversas facultades de la UCM. Este año acudirá a la reunión de la Semana de la Ciencia Indignada que se celebra en Mayo.



Objetivo final:

- Su principal objetivo es reducir el tiempo que emplea en la toma de actas y la validación de la misma.
- Facilitar el seguimiento de las tareas asignadas y reflejar los acuerdos llegados en una reunión.

Objetivos y motivaciones:

- Automatizar la toma de actas.
- Facilitar la moderación de los turnos de palabra.
- Facilitar la asignación y seguimiento de tareas.
 - Medir el interés y de participación de los miembros de la organización sobre las tareas asignadas.
- Tener un control de los puntos del día para poder repasarlos todos.
 - Identificación de los puntos del día antes de la reunión.
- Tener un espacio de consulta para todos los miembros de la organización.
 - Agilizar el envío del acta a los miembros de la reunión o asociación.
 - Publicación del acta en un entorno accesible para todos.
- Facilitar la validación del acta.
- Poder grabar la reunión para que los posibles ausentes a la reunión puedan verlo.
- La generación de un formato imprimible físico del acta (PDF, word...) imprimible.
- Recordatorio de realización de las tareas pendientes. Nombre:

Figura 4.2: Perfil de usuario

4.2.5. Definición de Escenarios y Requisitos

En este apartado se definen los posibles escenarios que puedan surgir en la aplicación. La idea es situarnos en un escenario real que pueda ocurrir en cualquier momento a lo largo del día, para detallar la solución al problema.

En cada uno de los escenarios se especificarán los requisitos necesarios para solventar el problema. Lo primero que tenemos que hacer es diferenciar los siguientes términos:

- **Acción:** es la actividad inmediata que requiere la solución.
- **Objeto:** sujeto principal del escenario.
- **Contexto:** refleja el objetivo final del requisito.

Escenario 1

El presidente de la asamblea, Carlos, ha convocado una reunión con dos días de antelación y le ha pedido a su secretario Antonio que debe avisar a todos los miembros de la asamblea de la citación a la reunión con una lista de los puntos del día que se van tratar.

Antonio accede con su portátil a la aplicación de **Teem-actas** y crea un nuevo proyecto, en el que va apuntando los puntos del día

que se van a tratar en la reunión y por último invita a los miembros citados a la asamblea a participar.

Requisitos

- Avisar (acción) a los miembros de la asamblea (objeto) para que puedan asistir a la reunión (contexto).
- Mostrar (acción) a los miembros invitados (objeto) los puntos del día que se van a tratar (contexto).

Escenario 2

Marta es la encargada de redactar el acta y Jaime, que es un miembro asistente a la reunión, le pregunta cuáles de los puntos del día han sido tratados hasta ese momento y cuales están aún pendientes. Además aprovecha para preguntarle qué tareas de la anterior reunión aún no han sido realizadas y que le diga a qué personas estaban asignadas.

Es entonces el momento en el que Marta mira en su portátil de un solo vistazo aquellos puntos del día que han sido hablados y los que no. Por otro lado busca el acta de la semana anterior y busca aquellas tareas que no han sido resueltas y además mira quien las tenía asignadas.

Requisitos

- Visualizar (acción) los puntos del día (objeto) que han sido tratados y los que no (contexto).
- Mostrar (acción) tareas sin realizar (objeto) de la semana anterior(contexto).
- Mostrar (acción) el nombre de las personas (objeto) que no han realizado su tarea de la semana anterior (contexto).

Escenario 3

Luis está presente en una reunión de una asamblea del barrio, a la cual han asistido más de 30 personas. Tiene como papel ser el moderador a la hora de gestionar los turnos de palabra y han levantado la mano 7 personas a la vez.

Accediendo a la aplicación se va a la sección de turnos de palabra y va apuntado los nombres de las personas según el orden que ha visto levantarse las manos.

Requisitos

- Anotar (acción) el turno de palabra (objeto) de la persona que acaba de levantar la mano (contexto).
- Conceder (acción) el turno de palabra (objeto) a una persona según el orden en el que ha pedido el turno (contexto).

4.2.6. Framework de diseño

En esta etapa describiremos todos los aspectos relacionados con el desarrollo del aspecto visual y la interacción con la aplicación. Para ello se utilizarán los escenarios y requisitos definidos en la fase anterior para crear los bocetos y los prototipos interactivos. Para definir el Framework de interacción realizamos un proceso de seis etapas:

- Definir el factor de forma, la postura y los métodos de entrada.
- Definir los elementos de datos y funcionales.
- Determinar los grupos funcionales y las jerarquías.
- Hacer un boceto del framework de interacción.
- Construir los diseños key-path.
- Validar los diseños con los escenarios de validación.

4.2.6.1. Definir el factor de forma, la postura y los métodos de entrada

En el factor de forma de nuestra aplicación se deberá tener en cuenta que su utilización se realizará principalmente desde portátiles, aunque también se podrá ejecutar en tablets y móviles. La postura de nuestra aplicación es temporal, ya que no requiere la total concentración del usuario como hemos expuesto anteriormente. El usuario podrá utilizar la aplicación mientras realiza otras actividades y solo interactuará cuando él quiera, no cuando la aplicación se lo “ordene”. Por último, el método de entrada para Teem-actas será táctil en pantallas de móviles y tablets, y para los ordenadores se requerirá de un teclado y un ratón.

4.2.6.2. Definir los elementos de datos y funcionales.

Lo primero de todo es definir los elementos de datos de nuestra aplicación, sus atributos y sus relaciones.

Elementos de datos

- Puntos del día
 - Atributos: Asunto, realizar, visualizar.
 - Relaciones: Asunto-realizar, Asunto-visualizar, realizar-visualizar.
- Tarea
 - Atributos: asunto, realizar, asignar, visualizar, usuario.
 - Relaciones: asunto-asignar, asunto-realizar, asunto-visualizar, realizar-visualizar, usuario-asignar, usuario-realizar.
- Turnos de palabra
 - Atributos: orden, usuario, asignar, visualizar.
 - Relaciones: orden-asignar, usuario-orden, usuario-visualizar.
- Votación
 - Atributos: asunto, positivo, negativo, abstención, validar, usuario.
 - Relaciones: asunto-positivo, asunto-negativo, asunto-abstención, validar-positivo, validar-negativo, validar-abstención, usuario-validar.
- Acta
 - Atributos: contenido, validar, enviar, usuario.
 - Relaciones: contenido-validar, usuario-enviar.

Elementos funcionales

- Visualizar los puntos del día de un acta.
- Marcar como revisado un punto del día.
- Visualizar tareas a realizar para el siguiente acta.
- Visualizar tareas pendientes de realizar de un acta anterior.
- Asignar una tarea a un usuario.
- Asignar turnos de palabra.
- Visualizar una lista de los turnos de palabra pendientes.
- Conceder turno de palabra al siguiente en la cola de espera.
- Registrar una votación.
- Validar una votación para que no se pueda modificar.

4.2.6.3. Determinar los grupos funcionales y las jerarquías

Para la pantalla principal de la aplicación. Grupos y elementos funcionales que contiene:

- Visualizar y editar Puntos del día:
 - Visualizar todos los puntos del día.
 - Marcar como revisado los puntos del día.
 - Editar texto del punto del día.
- Añadir y conceder turnos de palabras:
 - Poder añadir un turno de palabra.
 - Pasar al siguiente turno de palabra
 - Ver el historial de las personas que han intervenido.
 - Ver las personas que están a la espera de su turno de palabra.
- Añadir y registrar votaciones:
 - Poder añadir una votación realizada durante la reunión.
 - Visualizar votos a favor, en contra y las abstenciones.
 - Validar votación para que no pueda ser modificada.
- Añadir y asignar tareas:
 - Añadir una nueva tarea a realizar.
 - Asignar usuario que debe realizar la tarea.
 - Marcar tarea como realizada.
- Validar y enviar el acta:
 - Poder validar el acta al final de la reunión.
 - Enviar el acta tanto a los miembros que no han asistido a la reunión como a los que sí lo han hecho.

4.2.6.4. Hacer un boceto del framework de interacción

Individualmente, cada integrante del grupo realizó unos bocetos para las pantallas más relevantes que más se utilizarían en la aplicación. Para realizar estos bocetos, en lugar de hacer los bocetos en papel, se utilizó la herramienta Pencil [32], que es una herramienta para el prototipado. A continuación se mostrarán algunos de los bocetos que cada uno de los miembros realizó.

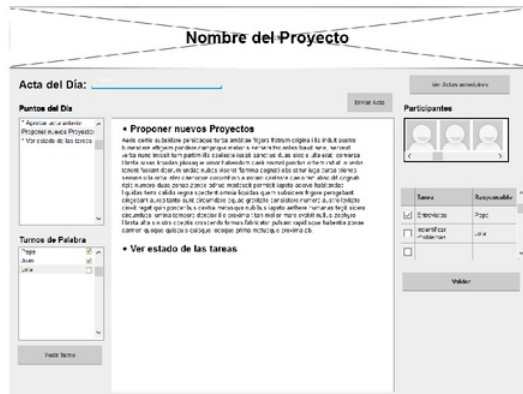


Figura 4.3: Prototipo Boceto 1



Figura 4.4: Prototipo Boceto 1

Escenario 1 - Escenario de cómo señalar los puntos del día que han sido hablados

Desde que ha empezado la reunión han pasado ya 20 minutos y el presidente de la asamblea le pregunta a Margarita, que es la encargada de redactar el acta de la reunión, cuantos puntos del día han sido tratados hasta ese momento y cuales quedan aún pendientes de ser debatidos. Margarita, que está utilizando una nueva plataforma que está pensada para la toma de actas llamada **Teem-actas**, ve en la parte izquierda de la pantalla de su ordenador los Puntos del Día que están tachados, lo cual significa que ya han sido debatidos.

Escenario 2 - Escenario de cómo moderar los turnos de palabra durante la reunión

Pepe es el encargado de redactar el acta y de la concesión los turnos de palabra en la reunión del barrio, a la que han acudido más de 30 personas. Todos los asistentes son muy activos y participativos, por lo que todos quieren intervenir en algún momento durante la reunión. Pepe que está utilizando la aplicación para la toma de acta llamada **Teem-actas**, interviene un momento en la reunión para decirles a los asistentes que pueden acceder a esta aplicación a través de su móvil o de su portátil y pedir el turno de palabra, donde el podrá ver en orden las peticiones para así poder conceder el turno de palabra.

Escenario 3 - Escenario de cómo visualizar y asignar tareas a realizar a personas

Juan es el encargado de redactar el acta. En la reunión actual se han anotado una serie de tareas a realizar que han no han sido asignadas a ningún usuario. Entonces algunos de asistentes se ofrecen voluntarios para realizar las tareas, por lo que Juan, que está redactando el acta en la aplicación web de **Teem-actas**, va añadiendo el nombre de la persona junto con la tarea que va a realizar.

Escenario 4 - Escenario de cómo validar una votación en un acta

Durante la reunión de la asociación de ASCII sale a la luz la propuesta de utilizar la herramienta de Google Docs en lugar de la de **Teem-actas** que es la herramienta que estan utilizando en dicha reunión. La propuesta es rechazada por unanimidad por lo que María, que es la encargada de tomar el acta, valida en la aplicación de Teem-actas que la propuesta a sido rechazada con 12 votos a favor, 1 en contra y ninguna abstención.

4.2.6.6. Validar los diseños con los escenarios de validación

En esta fase se pondrán a prueba los escenarios redactados anteriormente, validando ciertas situaciones que nos ayudarán a encontrar fallos o posibles mejoras en el diseño.

Utilizaremos preguntas para definir la respuesta del sistema para problemas o dudas que puedan surgir.

¿Qué pasaría si un usuario quisiera marcar un Punto del día como ya debatido?



Figura 4.7: Punto del día



Figura 4.8: Tachar Punto del día

El usuario iría al proyecto del cual se está redactando el acta y vería en la parte izquierda de la pantalla los puntos del día que se van a debatir en la reunión y marcaría como hablado aquel que ya

haya sido debatido con un simple click en el checkbox y el punto del día se tacharía automáticamente.

¿Como podría un usuario pedir un turno de palabra sin necesidad del moderador?

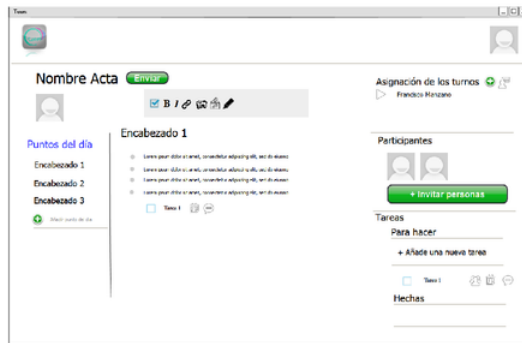


Figura 4.9: Pedir turno de palabra



Figura 4.10: Persona que tiene el turno de palabra

El usuario, que estará en la pantalla principal, vería en la parte superior derecha de la pantalla un título que pone “**Asignación de los turnos**”. Junto a este título habrá un botón con el símbolo + para que se apunte a la lista de espera de los turnos de palabra. El usuario podrá ver que su nombre aparece en la lista de espera de turnos de palabra.

¿Como podría un usuario asignar un turno de palabra a alguna persona?



Figura 4.11: Inicio Pantalla principal

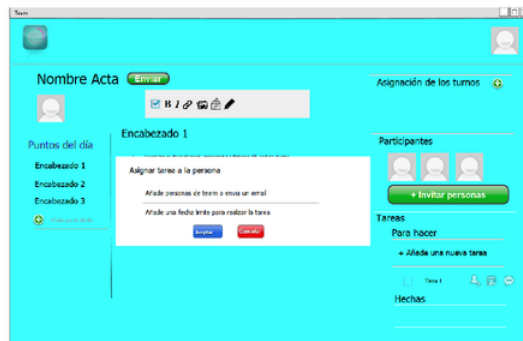


Figura 4.12: Modal para asignar una tarea

La reunión ya habría empezado y se han empezado a enumerar las tareas a realizar. Entonces hay asistentes a la reunión que comienzan a autoasignarse tareas, por lo que el encargado de redactar el acta debe añadir el nombre de la persona encargada en cada tarea. Para ello el encargado vería en la pantalla principal, concretamente en la parte inferior derecha, una sección con la lista de tareas pendientes que pone “Tareas para hacer”; haría click en el primer icono (una persona con el símbolo +) el cual abriría un modal donde se puede poner el nombre de la persona que va a realizar la tarea.

¿Como podría un usuario encargado de redactar el acta añadir una votación?

Por último nos queda resolver el problema de que un usuario que se encargue de tomar el acta tenga que validar una votación sobre un tema acordado. Para ello el usuario iría al icono de votaciones el

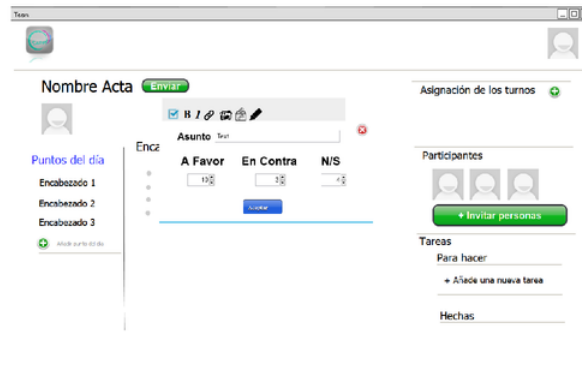


Figura 4.13: Inicio Pantalla principal



Figura 4.14: Modal para realizar una votación

cual abrirá un modal en el que solicitará le pedirá el asunto de la votación, el número de votos a favor, el número de votos en contra y el número de abstenciones.

4.2.7. Verificación. Principios de diseño

Para terminar el diseño de la aplicación y valorar la calidad del sistema, se utilizarán los 10 principios de diseños propuestos por Jakob Nielsen [21].

1. **Visibilidad del estado del sistema:** este principio de usabilidad web nos indica que siempre tenemos que tener informado al usuario de lo que está pasando en nuestra web y ofrecerle una

respuesta en el menor tiempo posible. En nuestro prototipo, a la hora de realizar la tarea de enviar el acta a usuarios de la aplicación, se muestra un pequeño diálogo indicando al número de usuarios al que se ha enviado el acta por correo electrónico.

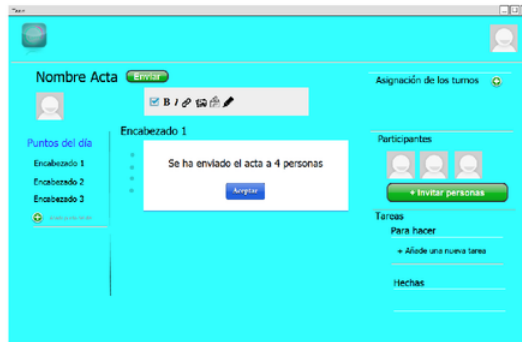


Figura 4.15: Visibilidad del estado del sistema

2. **Relación entre el sistema y el mundo real:** el sistema tiene que “hablar” el lenguaje del usuario con palabras o frases que a éste le sean familiares y que pueda reconocer con facilidad. En nuestra aplicación, añadir para pedir un turno de palabra se muestra de forma clara y concisa en la parte superior derecha el título **Turnos de palabra** en el que hay un icono con el símbolo + para añadir un usuario.

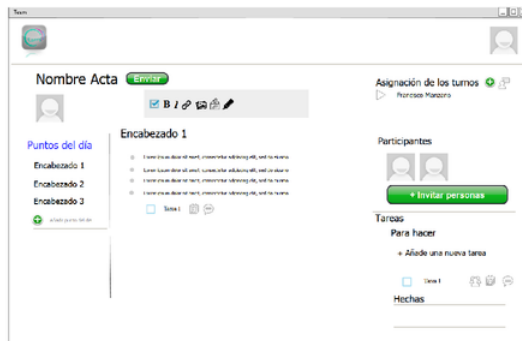


Figura 4.16: Relación entre el sistema y el mundo real

3. **Control y libertad del usuario:** a veces, un usuario se equivoca, es normal, está dentro de la naturaleza humana el equivo-

carce. Tenemos que darle al usuario la posibilidad de subsanar el error y no sentirse frustrado por no poder realizar algo. En este caso en nuestra aplicación cuando un usuario ha marcado por error un Punto del día como hablado, puede volver a desmarcarlo haciendo click en el checkbox que tiene al lado.

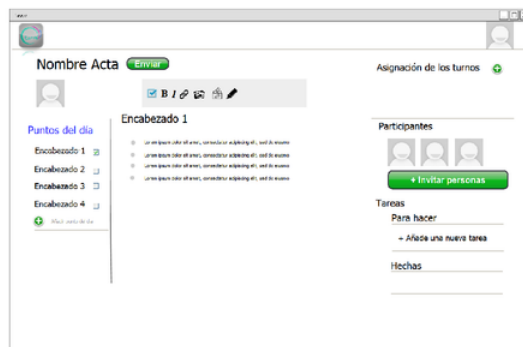


Figura 4.17: Control y libertad del usuario - Parte 1



Figura 4.18: Control y libertad del usuario - Parte 2

4. **Consistencia y estándares:** otro punto que tenemos que tener en cuenta es seguir los convenios establecidos para ciertos iconos. Con el auge de los dispositivos móviles han aparecido nuevos gestos e iconos que ya hemos asumido como normales. La aplicación sigue las pautas de diseño propuestas por las pautas de **Material Design** [9]. En este caso nuestra aplicación también lo cumple porque utilizamos iconos claros como por ejemplo el botón **añadir** que es parecido a un **+**, el cual es intuitivo y significa el hecho de tener que añadir algo.

5. **Prevención de errores:** este principio nos indica que tenemos, en todo lo posible, que prevenir cualquier error que pueda cometer el usuario. Y dado el caso de que este cometa uno, tenemos que poner a su alcance todas las opciones posibles para poder corregirlo. La opción de autocompletar de Google es un buen ejemplo de este principio de usabilidad web. Otro caso puede que pueda ayudar a cumplir este principio es el la autocomprobación de campos de formularios en tiempo real. En este caso, nuestra aplicación comprobaba en tiempo real que los datos introducidos dentro de una votación solo son de tipo numérico.
6. **Reconocer antes que recordar:** este principio nos indica que siempre es mejor reconocer que obligar al usuario a memorizar acciones u objetos para que pueda cumplir su objetivo. En este caso, para **añadir nuevo Punto del Día**, el usuario reconocerá el icono para añadir el punto del día y automáticamente aparecerá el widget en el editor de textos para que ponga el título del Punto del Día.
7. **Flexibilidad y eficiencia de uso:** tenemos que tener un sitio web preparado para todo tipo de usuario, desde los más novatos hasta los más experimentados. También cumple este principio porque en la parte superior del editor de texto, hay un **toolbar** con muchas opciones para que un usuario pueda añadir propiedades al texto que está escribiendo en el acta.
8. **Diseño estético y minimalista:** La mejor forma de recordar este principio básico de usabilidad web es con el acrónimo KISS, Keep It Simple Stupid [28]. En nuestra aplicación tanto los diálogos que muestra como la información visible en la mayoría de las pantalla es simple y consisa. Así evitamos dar información innecesaria al usuario que pueda distraer su atención, lo cual le llevaría a confusión.
9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** tenemos que intentar que todos los errores que puedan ocurrir en la web estén expresados en un lenguaje entendible por todos, no por códigos. Cuando un widget ha sido borrado, pero no se ha producido un error interno, se muestra un mensaje informando al usuario diciendo **Este punto del día ha sido borrado**
10. **Ayuda y documentación:** Cuando el usuario genera contenido, el sistema va indicando al usuario los pasos que tiene

que seguir para completar su tarea de forma satisfactoria. Esto ocurre en nuestra aplicación cuando un usuario construye una votación, donde el sistema le indica los campos que tiene que rellenar y la información que debe ir en cada campo.

Capítulo 5

Desarrollo e implementación

En esta sección se hablará de cómo se han implementado las nuevas funcionalidades que se han descrito en la fase del prototipado, así como las distintas tecnologías que se han utilizado a lo largo del desarrollo.

5.1. Introducción

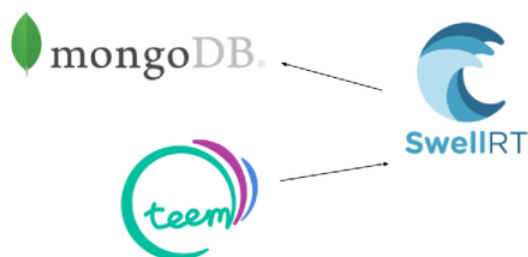


Figura 5.1: Teem-actas

La aplicación que se ha desarrollado utiliza las siguientes tecnologías:

- Teem [62], es una herramienta web y móvil para la colaboración en comunidades, colectivos y movimientos sociales. Además está construido dentro del proyecto P2Pvalue [10]. Hemos utilizado esta herramienta ya que nos ofrecía funcionalidades que podíamos reutilizar, pero hemos tenido que añadir algunas otras para poder realizar las funcionalidades con el mínimo de

requisitos que detectamos cuando hicimos la fase de investigación y prototipado.

- SwellRT [14] es un backend-as-a-service [36] de código abierto. Proporciona funciones para el almacenamiento en tiempo real, gestión de usuarios, autenticación e integración basada en eventos. El objetivo de SwellRT es facilitar el desarrollo de aplicaciones colaborativas en tiempo real, especialmente aquellas que requieren una edición colaborativa de texto.
- MongoDB es una base de datos NoSQL [58] en la que persistirá y se gestionarán los datos de los usuarios, proyectos y el texto de la toma de actas. Es lo que la aplicación utiliza para guardar toda la información.

5.2. Desarrollo de Teem-Actas

El proyecto de Toma de Actas parte del proyecto de **Teem** [61] cuyo código es Open Source y está disponible en GitHub. Lo primero que hicimos fue hacer un fork del proyecto y renombrarlo **teem-actas**. El proyecto está desarrollado en AngularJS combinado con Saas, l10, npm, bower y algunas tecnologías más descritas anteriormente. Las principales partes del proyecto en las que hemos tenido que modificar o añadir código para implementar las nuevas funcionalidades son:

- **src/templates**. En esta sección hemos añadido los archivos html que se corresponden con la parte visual de la aplicación.
- **src/saas**. Aquí se definen los iconos y las clases CSS para formatos y los estilos de la aplicación.
- **src/l10n**. Se definen los labels o partes de la aplicación que dependen del idioma en el que se despliegue la aplicación. En este caso están definidos los labels tanto en castellano como en inglés.
- **src/images**. En esta parte del proyecto se definen los imágenes e iconos utilizados en la aplicación.
- **src/js**. En esta sección hemos modificado tres partes:
 - (1) **src/jc/controllers**. Los controladores en AngularJS son objetos que permiten desarrollar la lógica de la aplicación, enlaza el ámbito, \$scope, con la vista y permite tener un control total de los datos, es decir, es el encargado de gestionar los eventos.

- (II) **src/jc/directives**. Las directivas son la forma de extender los html y nos permiten agregar desde pequeños trozos de código hasta una funcionalidad completa.
- (III) **src/jc/services**. Los servicios son los objetos “singleton”, donde definimos la lógica de negocio de la aplicación, con el objetivo de que sea reutilizable e independiente de las vistas.

En los siguientes apartados expondremos las funcionalidades que hemos añadido al proyecto de Teem [62]:

- **Puntos del día.**
- **Asignación de Tareas.**
- **Votaciones.**
- **Turnos de palabra.**

5.2.1. Puntos del día

Una de las principales necesidades que tiene un usuario a la hora de tomar un acta, es redactar los puntos del día que se van a debatir en una reunión. Además una vez iniciada la reunión y a la vez que va apuntando lo que van hablando los miembros asistentes a la reunión, el usuario debe ir marcando de alguna forma qué puntos del día han sido abordados y los que están pendientes de ser tratados en la reunión. Para solucionar este problema, en un principio se estudió la posibilidad de utilizar los **Headers** o encabezados que se utilizan en Teem, pero nos dimos cuenta que presentaba un inconveniente: los **Headers** solo añaden la propiedad `<h3>` al editor de texto de SwellRT, pero no nos aportaba una propiedad para poder saber si ese texto, o Punto del día, ya había sido hablado a lo largo de una reunión.

En la figura 5.2 representa el diagrama de clases que se ha utilizado para añadir un nuevo **Punto del día**. La principal clase en este diagrama es la del widget **OderDay** que cuenta con los siguientes atributos:

- **id**: debe ser único y representa el ID del Punto del día que se va a añadir.
- **text**: es el título del Punto del día.
- **completed**: indicará si un Punto del día ha sido tratado o no.

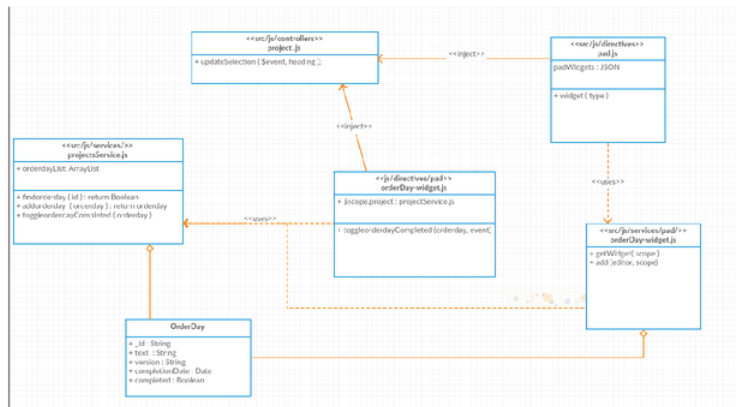


Figura 5.2: Diagrama de clases para añadir un nuevo punto del día

- **completionDate**: fecha en la que se ha marcado un Punto del día como hablado.

La clase **ProjectService** contiene una lista de los **Puntos del día** que contiene el acta. También contiene los siguientes métodos:

- **findorderday (id)**: devuelve un booleano indicando si un Punto del día se encuentra en el acta que se esta redactando o no.
- **addorderday (orderday)**: añade un nuevo Punto del día al acta que se esta redactando.
- **toggleorderdayCompleted (orderday)**: método que marca un Punto del día como tratado.

En la figura 5.3 se muestra el diagrama de secuencia que se produce cuando un usuario quiere añadir un nuevo punto del día. Cuando un usuario añade un nuevo punto del día al editor del texto, lo primero que se debe hacer es crear un nuevo widget, en este caso el widget **orderday** con el campo **text** vacío y el campo **completed** a falso, y añadirlo en la posición que el usuario ha indicado en el editor de texto. Por otro lado para poder marca un punto del día como hablado, el usuario tiene que pulsar el botón de tipo checkbox que contendrá el widget. El controlador se encargará de escuchar los eventos que ocurren en este widget-oderday, que en este caso escuchará los eventos del checkbox del widget, y se marcará el punto del día como **completado(hablado o tratado)**. La figura 5.4 representa el diagrama de secuencia que se produce cuando un usuario

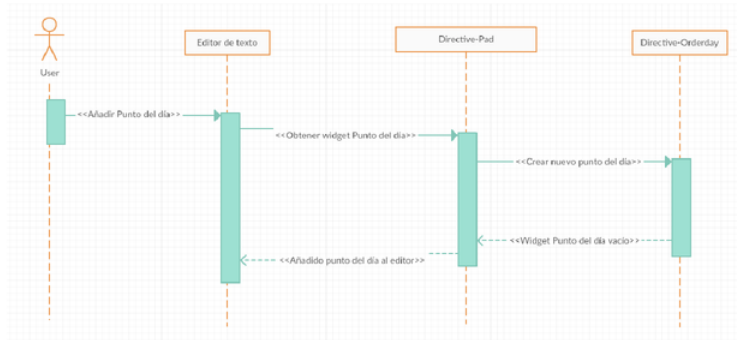


Figura 5.3: Diagrama de secuencia para añadir un nuevo punto del día

realiza la acción de marcar un Punto del día como completado.

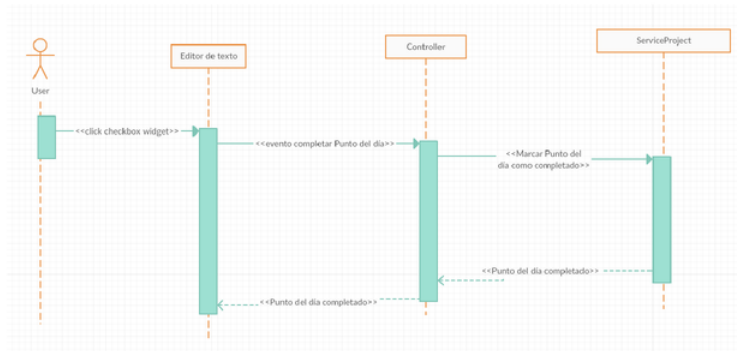


Figura 5.4: Diagrama de secuencia para marcar un Punto del día como completado

A continuación se expondrá la lista de ficheros que se han modificado o añadido dentro del proyecto. Para ver más en detalle del desarrollo de esta funcionalidad, esta disponible en el repositorio de Teem-actas en GitHub [11]:

- `/src/templates/pad/orderday-widget.html`. El template para representar la parte visual del widget dentro del editor de texto la aplicación.
- `/src/templates/projects/outline.html`. El template para representar el listado de los Puntos del Día a la izquierda del editor de texto.

- `/src/controllers/project.js`. El encargado de escuchar los eventos que ocurren en los templates.
- `/src/directives/pad/orderday-widget.js`. Donde añadimos la función `toggleorderdayCompleted(orderday, event)` para poder marcar si un Punto del Día ha sido hablado o no.
- `/src/services/pad/orderday-widget.js`. Es el encargado de crear el objeto singleton que define la lógica del widget dentro del editor de texto (SwellRT).
- `/src/directives/pad.js`. Aquí es donde se llama al objeto singleton que se ha definido en el servicio y se añade a SwellRT.
- `/src/l10n/es.json`. Aquí hemos añadido los labels que se muestran en la aplicación en español.
- `/src/l10n/en.json`. Aquí hemos añadido los labels que se muestran en la aplicación en inglés.
- `/src/saas/orderday.sass`. En esta parte hemos añadido las clases css para los estilos.
- `/src/saas/icon.saas`. En esta sección hemos añadido el icono que hemos utilizado para el widget que hemos sacado de Material Design.
- `/src/services/projectsService.js`. Implementa la lógica que tiene el acta. En este caso se define la lógica de la lista de los Puntos del Día dentro del objeto singleton del Acta.

5.2.2. Asignación de Tareas

Otra de las necesidades que detectamos en la fase de Investigación, es que a la hora de tomar un acta, se definen tareas a realizar para la siguiente reunión y además se asignan dichas tareas a algún miembro asistente a la reunión. Teem [62] ya soluciona uno de estos problemas con el “widget needs”, que representan las tareas a realizar. Con la funcionalidad de las tareas ya implementada, tuvimos que añadir otra funcionalidad para poder asignar estas “needs” o tareas a usuarios.

Las tareas se pueden asignar tanto a usuarios que tengan una cuenta en **Teem-actass** como a persona que no dispongan de dicha cuenta, simplemente poniendo el nombre de la persona a la que se le ha asignado la tarea. Se determinó esto porque puede darse la situación de que algunos de los miembros de la comunidad que asisten a las reuniones no tengan una cuenta en **Teem-actas**.

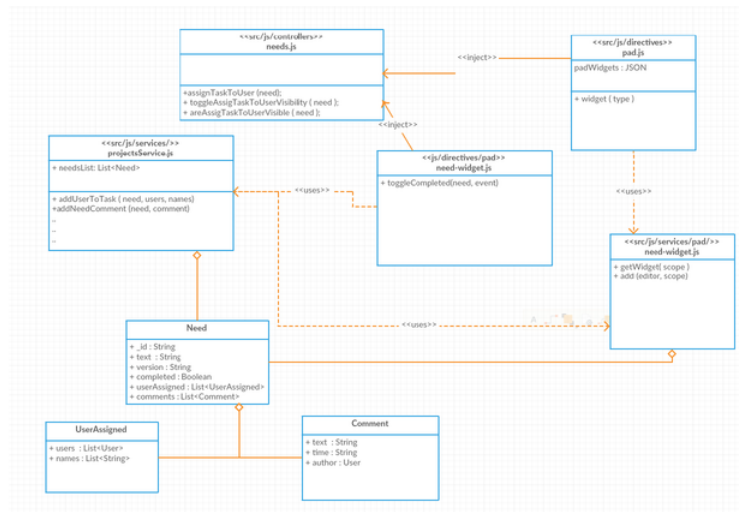


Figura 5.5: Diagrama de clases para asignar una tarea a un usuario

En la figura 5.5 representa el diagrama de clases que se ha reutilizado para añadir la nueva funcionalidad de asignar una **need**(tarea) a un usuario. Dentro de la clase **Need** añadimos un nuevo atributo **userAssigned** que representa una lista de objetos con las personas asignadas a realizar una tarea:

- **users**: representan los usuarios registrados dentro de la aplicación Teem-actas.
- **names**: representan los nombres de las personas asistentes a la reunión.

En la figura anterior 5.6 se muestra el proceso de asignar una tarea a una persona:

1. El encargado de la toma de acta añade el nombre de la persona que va a realizar la tarea.
2. El widget de que se encuentra en el editor de texto, lanza un evento para indicar que se va asignar una persona a una tarea.
3. El controlador escucha este evento y llama al servicio para que añada a la persona a la lista de persona a realizar dicha tarea.
4. El servicio añade a la persona a la lista de personas a realizar una tarea.

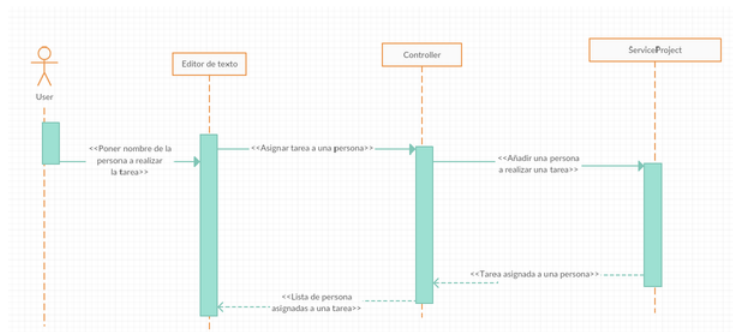


Figura 5.6: Diagrama de secuencia para asignar una tarea a una persona

A continuación se expondrán la lista de ficheros que se han modificado o añadido dentro del proyecto. Para ver más en detalle del desarrollo de esta funcionalidad, esta disponible en el repositorio de Teem-actas en GitHub [11]:

- `/src/template/needs/need.html`: aquí añadimos la parte visual para poder asignar una tarea a un usuario.
- `/src/controllers/needs.js`: es el controlador que está escuchando los eventos que se producen en el template de las `need.html`. Añadimos el evento para añadir a asignar a una tarea a un usuario.
- `/src/services/projectsService.js`: implementa la lógica que tiene el Acta. En este caso se define la lógica de la lista de las Needs o tareas dentro del objeto singleton del Acta. Este caso, para cada Need se ha añadido una propiedad que consta de dos variables :
 1. `need.userAssigned.users`: que representa una lista con los usuarios de Teem-actas a los que se les ha asignado una tarea.
 2. `need.userAssigned.names`: representa la lista de nombres de personas que tiene que realizar una tarea.
- `/src/i10n/es.json`. Aquí hemos añadido los labels que se muestran en la aplicación en español.
- `/src/i10n/en.json`. Aquí hemos añadido los labels que se muestran en la aplicación en inglés.

- `/src/saas/needs.sass`. En esta parte hemos añadido las clases css para los estilos.
- `/src/saas/icon.saas`. En esta sección hemos añadido el icono que hemos utilizado para el widget que hemos sacado de Material Design.

En resumen hemos reutilizado el widget **Need** que ya existía en **Teem** y se ha añadido una funcionalidad más para poder asignar estas “needs” a usuarios de Teem-actas o a nombres de personas.

5.2.3. Votaciones

Por otro lado, cuando una comunidad se reúne para tratar un tema, suelen tomarse decisiones que suelen ser votadas y aprobadas por los asistentes a dicha reunión. Esto nos llevó a implementar esta funcionalidad que no existía en Teem. Para ello, tuvimos que crear otro widget en SwellRT para poder registrar una votación a la hora de tomar un acta. Este widget, a diferencia del widget de los Puntos de Día que solo constaba del título y un parámetro para saber si el Punto del Día había sido hablado o no, consta de cuatro parámetros: un texto que representa el asunto de la votación, los votos a favor, los votos en contra y las abstenciones.

En la figura 5.7 representa el diagrama de clases que se ha utilizado

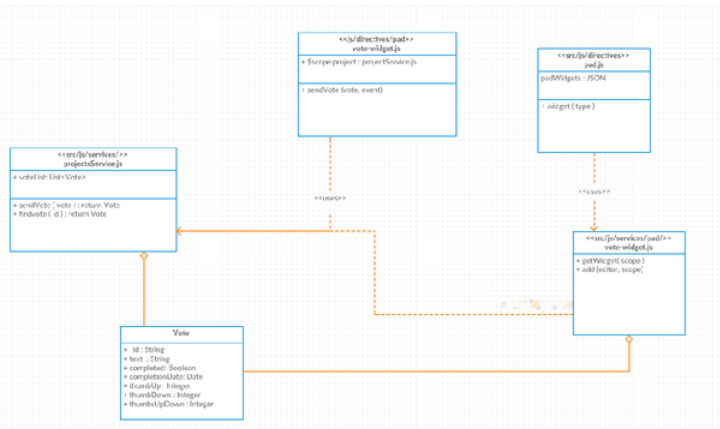


Figura 5.7: Diagrama de clases para añadir una votación

para añadir una nueva **votación**. La principal clase en este diagrama es la del widget **Vote** que cuenta con los siguientes atributos:

- **id**: debe ser único y representa el ID de la votación que se va a añadir.
- **text**: asunto de la votación.
- **completed**: boolean que indica si una votación ha sido validada o no. Si una votación ha sido validada no se podrá modificar los resultados de las votaciones.
- **completionDate**: fecha en la que se ha validado la votación.
- **thumbUp**: votos a favor.
- **thumbDown**: votos en contra.
- **thumbsUpDown**: abstenciones.

La clase **ProjectService** contiene una lista de las **votaciones** que contiene el acta. También contiene los siguientes métodos:

- **findvote (id)**: método para buscar un voto dentro de la lista de votaciones del acta.
- **sendVote (vote)**: método para validar una votación, poniendo el atributo **completed** a true. Si una votación ha sido validada ya no se podrá modificar.

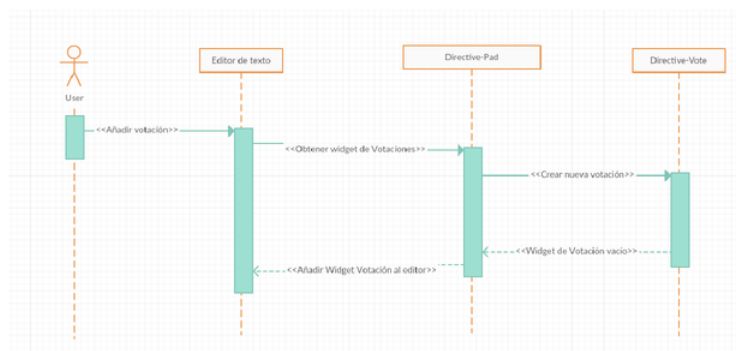


Figura 5.8: Diagrama de secuencia para añadir una votación

En la figura anterior 5.8 se muestra el diagrama de secuencia para añadir una nueva votación en la toma de un acta:

1. El encargado de la toma de acta añade una votación.

2. El editor de texto solicita a la directiva un nuevo widget de votación vacío, es decir, con el campo **text (asunto) vacío** y con atributos de los votos a favor, en contra y las abstenciones con el contador a cero.
3. La directiva devuelve el widget de votaciones.
4. El editor de texto añade el widget de votaciones en la posición en la que ha marcado el usuario.

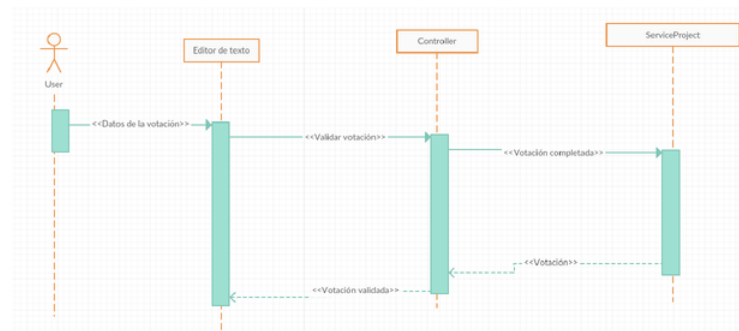


Figura 5.9: Diagrama de secuencia para validar una votación

Por otro lado, el widget de votaciones tiene otra funcionalidad para poder **validar** una votación en un acta. Una votación que esta validada, significa que ya no se puede modificar el valor del resultado de la votación, es decir, no se puede cambiar el número de votos a favor, en contra y las abstenciones.

En la figura anterior 5.8 se muestra el diagrama de secuencia para añadir una nueva votación en la toma de un acta:

1. El encargado de la toma de acta hace click en el botón para validar una votación.
2. El widget lanza un evento para validar la votación.
3. El controlador escucha este evento y llama al servicio para que marque la votación como validada
4. El servicio pone el atributo **completed** a **true** para que la votación quede como validada.

A continuación se expondrán la lista de ficheros que se han modificado o añadido dentro del proyecto. Para ver más en detalle del desarrollo de esta funcionalidad, esta disponible en el repositorio de Teem-actas en GitHub [11]:

- `/src/templates/pad/vote-widget.html`. El template para representar la parte visual del widget dentro del editor de texto la aplicación.
- `/src/controllers/project.js`. El encargado de escuchar los eventos que ocurren en los templates.
- `/src/directives/pad/vote-widget.js`. Aquí añadimos la una función `sendVote(orderday, event)` para poder validar una Votación. En una votación que ya ha sido validada ya no se puede modificar el número de votos.
- `/src/services/pad/vote-widget.js`. Es el encargado de crear el objeto singleton que define la lógica del widget dentro del editor de texto (SwellRT).
- `/src/directives/pad.js`. Aquí es donde se llama al objeto singleton que se ha definido en el servicio y se añade a SwellRT.
- `/src/l10n/es.json`. Aquí hemos añadido los labels que se muestran en la aplicación en español.
- `/src/l10n/en.json`. Aquí hemos añadido los labels que se muestran en la aplicación en inglés.
- `/src/saas/vote.sass`. En esta parte hemos añadido las clases css para los estilos.
- `/src/saas/icon.sass`. En esta sección hemos añadido el icono que hemos utilizado para el widget que hemos sacado de Material Design.
- `/src/services/projectsService.js`. Implementa la lógica que tiene el Acta. En este caso se define la lógica de la lista de las Votaciones dentro del objeto singleton del Acta, como la validación de una votación.

En resumen, se ha añadido un nuevo **widget (vote)** consta de los siguientes parámetros:

1. El campo *text* que representa el asunto de la votación.
2. El campo *completed* para saber si una votación ya ha sido validada o no. En una votación que ya ha sido validada ya no se puede modificar el número de votos.
3. El campo *author* nos indica el usuario de Teem-actas que ha dado de alta la votación y la ha registrado.

4. El campo *time* que nos indica la fecha en la que se ha dado de alta la votación.
5. El campo *thumbDown* indica el número de votos en contra.
6. El campo *thumbUp* indica el número de votos a favor.
7. El campo *thumbsUpDown* indica el número de abstenciones.

5.2.4. Turnos de palabra

Por último, otra de las necesidades que detectamos a la hora de realizar la tarea de la toma de actas, es a la hora de conceder los turnos de palabra en una reunión. Para resolver este problema, se ha implementado la funcionalidad para que, tanto los asistentes a la reunión que disponen de una cuenta en Teem-actas como los que no lo tiene, puedan pedir el turno de palabra y quede constancia de las personas que han intervenido en la reunión. De esta forma se respetará el orden de peticiones en los turnos de palabra y quedará constancia de que quién ha participado en la reunión.

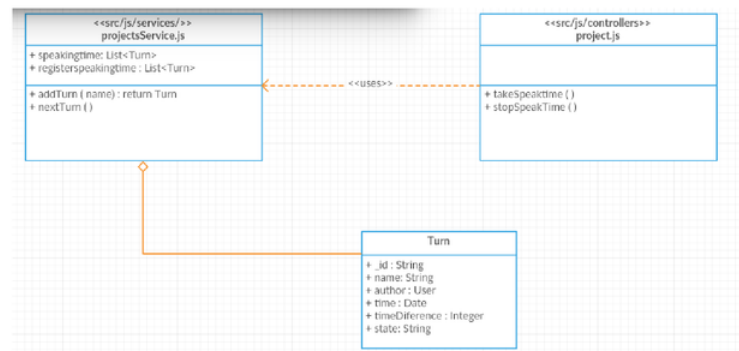


Figura 5.10: Diagrama de clases para solicitar un turno de palabra

En la figura 5.10 representa el diagrama de clases que se ha utilizado para poder asignar **turnos de palabra** en una toma de actas. La clase **Turn** que cuenta con los siguientes atributos:

- **id**: debe ser único y representa el ID de la votación que se va a añadir.
- **name**: nombre de la persona que ha solicitado el turno de palabra.
- **author**: usuario de **Teem-actas** que ha apuntado el nombre de la persona que ha solicitado el turno de palabra.

- **time**: representa la fecha en la que se ha solicitado el turno de palabra.
- **timeDiference**: tiempo que ha hablado la persona que ha solicitado el turno de palabra.
- **state::** estado actual del turno de palabra.
 - **ready**: la persona tiene el turno de palabra.
 - **wait**: la persona esta esperando su turno de palabra.
 - **stopped**: la persona ha terminado su turno de palabra.

La clase **ProjectService** contiene una lista de los **turnos de palabra** que se han solicitado:

- **speakingtime**: lista de las persona que han solicitado el turno de palabra.
- **registerspeakingtime**: lista de personas que representa el historial de los turnos de palabra que se han asignado en una reunión.
- **addTurn (name)**: método para añadir un nuevo turno de palabra a la lista de espera.
- **nextTurn ()**: método para pasar al siguiente turno de palabra. La persona que en ese momento tenga asignado el turno de palabra, pasará al historial de turnos de palabra y se le asignará el turno de palabra a la siguiente persona que este en la lista de espera.

En la figura anterior [5.11](#) se muestra el diagrama de secuencia para pedir el turno de palabra:

1. Un usuario introduce el nombre de la persona que ha solicitado el turno de palabra.
2. El controlador escucha el evento de solicitud un turno de palabra y llama al **servicio** para que añada el nombre de la persona que ha solicitado el turno de palabra
3. El servicio añade a la lista de espera el nuevo turno de palabra.

Para pasar al siguiente turno de palabra, se ha implementado el siguiente diagrama de secuencia [5.12](#) que muestra el flujo que se produce a la hora de solicitar la finalización del turno de palabra actual y pasar al siguiente turno de la lista de espera:

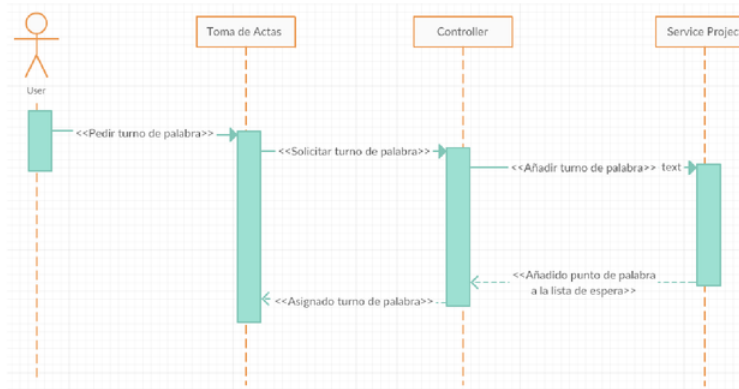


Figura 5.11: Diagrama de secuencia para pedir un turno de palabra

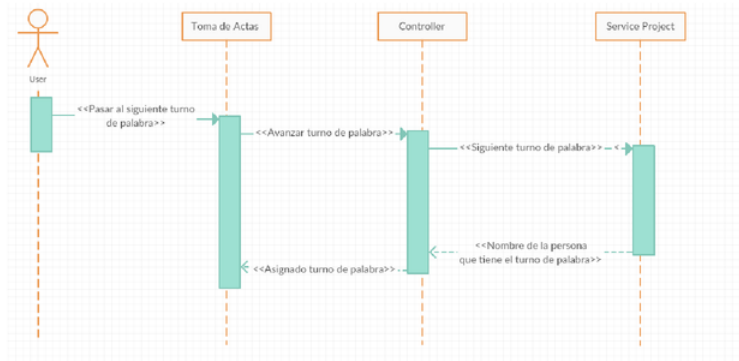


Figura 5.12: Diagrama de secuencia para pasar al siguiente turno de palabra

1. El encargado de la toma de acta asigna el turno de palabra a la siguiente persona que esta pendiente en la lista de espera.
2. El controlador escucha el evento para pasar al siguiente turno de palabra.
3. El servicio registrar dentro del **historial de turnos de palabra** la persona que acaba de terminar su turno de palabra y pasa a asignarle el turno de palabra a la siguiente persona que estaba esperando su turno en la lista de espera.

Por último, otra de las funcionalidades que se ha implementado es la de mostrar un historial con la lista de las persona que han hablado

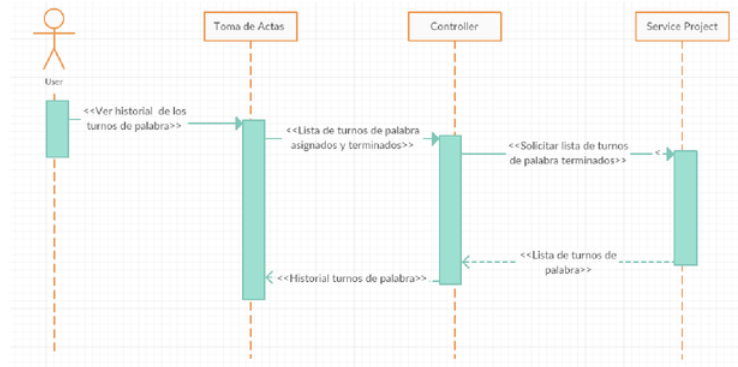


Figura 5.13: Diagrama de secuencia para mostrar el historial de turnos de palabra

a lo largo de la reunión de una toma de actas. En la figura 5.13 se muestra el diagrama de secuencia que se produce cuando un usuario solicita ver el historial de turnos de palabra:

1. El usuario solicita ver la lista de turnos de palabra que ya se han asignado y terminado.
2. El controlador escucha el evento para ver historial de turnos de palabras.
3. El servicio devuelve la lista (**registerspeakingtime**) que contiene el historial de personas que han hablado a lo largo de la reunión.

A continuación se expondrán la lista de ficheros que se han modificado o añadido dentro del proyecto. Para ver más en detalle del desarrollo de esta funcionalidad, esta disponible en el repositorio de Teem-actas en GitHub [11]:

- **/src/template/projects/speaking-time.html**. Aquí añadimos la parte visual para poder pedir un turno de palabra, ver que persona está hablando en ese momento y poder ver un historial de las personas que han hablado en la reunión.
- **/src/controllers/projects.js**. Es el controlador que está escuchando los eventos que se producen en el template de las `/speaking-time.html`. Añadimos el evento para pedir el turno de palabra y unirse a la cola de esperado.

- `/src/services/projectsService.js`. Implementa la lógica que tiene el Acta. En este caso se define la lógica de la lista de los turnos de palabra dentro del objeto singleton del acta, como añadir un turno de palabra, o como pasar al siguiente turno de palabra.
- `/src/l10n/es.json`. Aquí hemos añadido los labels que se muestran en la aplicación en español.
- `/src/l10n/en.json`. Aquí hemos añadido los labels que se muestran en la aplicación en inglés.
- `/src/saas/projects.sass`. En esta parte hemos añadido las clases css para los estilos.
- `/src/saas/icon.saas`. En esta sección hemos añadido el icono que hemos utilizado para el widget que hemos sacado de Material Design.

Capítulo 6

Producto final:

6.1. Prototipo funcional

En este apartado se mostrarán algunas pantallas de la aplicación desarrollada en base al lo que hemos descrito en la sección 5.2. El objetivo es mostrar el resultado final de la implementación para su posterior prueba con usuarios reales.

6.1.1. Puntos del día

Una de las necesidades que tiene un usuario que tome el acta en una reunión, es el tener que añadir los **Puntos del día** que se van a debatir en una reunión. Otra de las necesidades que tienen estos usuarios es el hecho de poder marcar si un **Puntos del día** ya ha concluido. Para solventar estos problemas hemos añadido un widget en SwellRT [14] que tiene dos aspectos importantes: el texto del Punto del Día y un booleano para saber si este punto del día ha sido tratado o no.

6.1.1.1. Añadir un nuevo Punto del día



Figura 6.1: Diagrama de Flujo para dar de alta un Punto del día

En la toma de un acta, lo primero que se hace es redactar una lista con los **Puntos del día** que se van a debatir en una reunión. En la figura 6.1 se muestra el diagrama de flujo que se produce cuando un usuario quiere dar de alta un nuevo punto del día:

1. El usuario tiene que hacer click en el icono del widget “Puntos del día”.
2. Introducir el **título** del Punto del día.

En las siguientes imagenes se muestra un posible caso de uso donde un usuario añade un punto del día con el título “Primer punto del día” en la aplicación de Teem-actas.

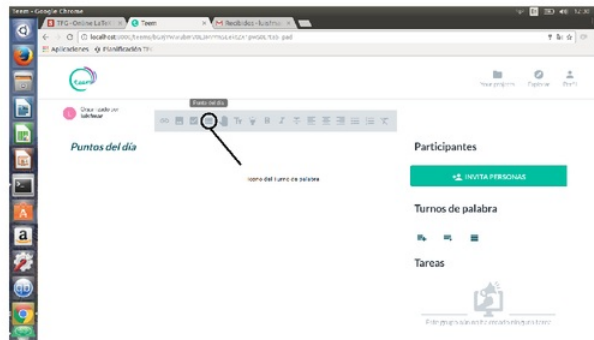


Figura 6.2: Icono del widget para un Punto del día



Figura 6.3: Añadir el título a un Punto del día

6.1.1.2. Marcar un Punto del día como concluido

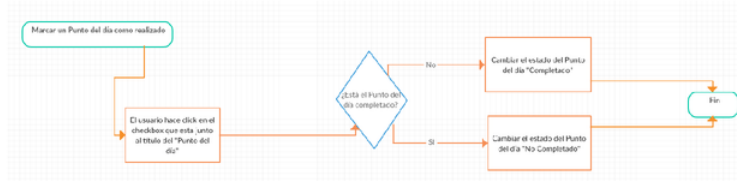


Figura 6.4: Diagrama de flujo para marcar un Punto del día como concluido

Otro caso de uso dentro de los Puntos del día es poder marcar de alguna forma un **Punto del día** como “hablado o tratado”. El siguiente diagrama de flujo 6.5 representa el proceso para **marcar un Punto del día como tratado o hablado**:

1. Hacer click en el checkbox situado a la izquierda del título del turno de palabra.
2. ¿Esta el Punto del día marcado como “hablado o completado”?
 - En caso de que el Punto del día este en el estado “completado” pasa al estado “no completado”
 - En caso de que el Punto del día este en el estado “no completado” pasa al estado “completado”

Para poder realizar esta acción dentro de la aplicación de **Teem-actas**, solo hace falta **marcar o desmarcar el checkbox** que aparece junto al título del Punto del día y automáticamente la aplicación de Teem-actas tachará el título del Punto del día para representar que dicho Punto del día ha sido hablado. La siguiente imagen muestra el resultado final de realizar esta acción.

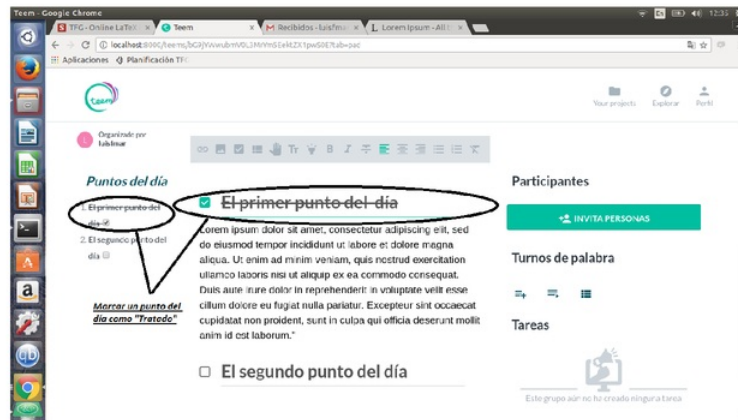


Figura 6.5: Marcar un Punto del día como concluido

6.1.2. Asignación de Tareas

A la hora de una toma de actas en una reunión, se necesita añadir una serie de tareas y anotar el responsable o responsables encargados de realizar dichas tareas. Teem [62] proporciona una solución para una de estas necesidades, que es el widget con nombre **tareas**, que se ha reutilizado en la aplicación de Teem-actasa. Este widget tiene como propiedades:

- **Texto:** un campo de tipo texto para poner el nombre o la tarea a realizar.
- **Checkbox:** un boton de tipo checkbox para poder marcar una tarea como realizada o no realizada.
- **Comentarios:** este widget nos permite añadir comentarios a una tarea.

Con estas propiedades que nos proporciona el widget, tuvimos que añadirle una funcionalidad más para poder asignar estas tareas a usuarios o personas asistentes a la reunión de la toma de actas.



Figura 6.6: Diagrama de flujo para asignar una tarea

La imagen 6.6 es el diagrama de flujo que representa el proceso que produce cuando se da de alta una nueva tarea y se asigna esta tarea a una persona:

1. Hacer click en el icono del widget **tareas**.
2. Poner un título a la tarea que se va a realizar.
3. Añadir nombre de la persona a la que se va asignar una tarea.

Un posible caso de uso es el siguiente: En una reunión en la que, Marcos que es el encargado de la toma de actas, ha creado un nuevo proyecto dentro de la aplicación de Teem-actas y previamente ha dado de alta los Puntos del día, añade una nueva tarea con título **Tarea a realizar** y además asigna esta tarea a una persona asistente a la reunión cuyo nombre es **María Martin**. En las siguientes imágenes se muestra el caso de uso descrito anteriormente.

- Crear una nueva tarea :
 1. Hacer click en el icono de **tareas**.

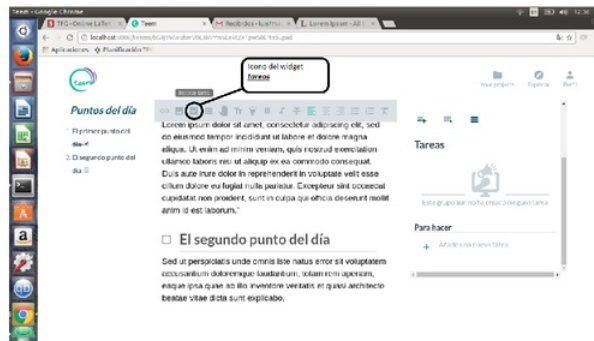


Figura 6.7: Icono del widget de tareas

2. Poner el título del punto del día.

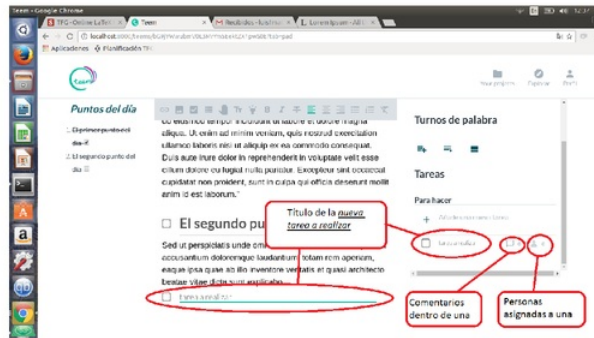


Figura 6.8: Añadir el asunto de la tarea

- Asignar una tarea a María Martín.

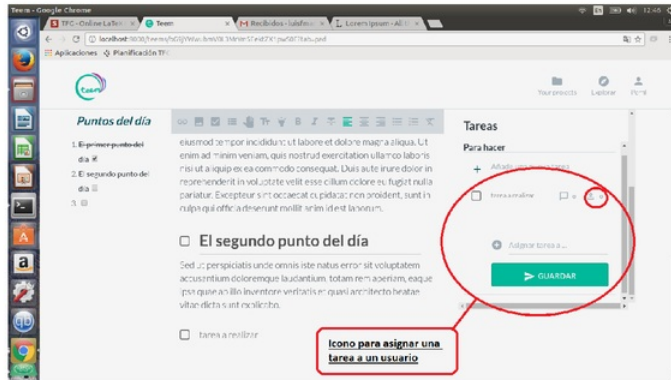


Figura 6.9: Icono para asignar una tarea

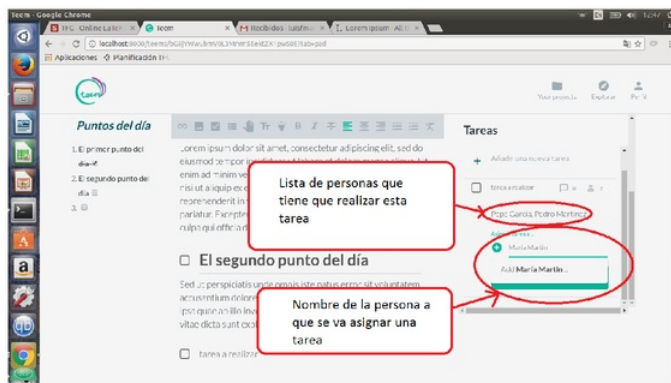


Figura 6.10: Nombre de la persona a la que se va asignar una tarea

6.1.3. Turnos de palabra

Por otro lado, en una toma de actas, otro de los problemas que se encuentran los usuarios que toman el acta en una reunión, es la forma de gestionar las consecuciones de los turnos de palabra. Para solucionar este problema, Teem-actas tiene una funcionalidad que permite gestionar la moderación de los turnos de palabra en una reunión. Esta funcionalidad permite tener una lista de espera que ordena las peticiones de turnos de palabra que los asistentes a la reunión solicitan. Además permite ver un historial completo de las personas que han intervernido dentro de una reunión.

6.1.3.1. Solicitar turno de palabra

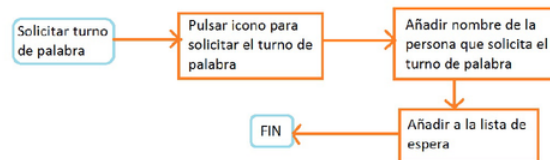


Figura 6.11: Diagrama de flujo para solicitar un turno de palabra

La imagen 6.11 es el diagrama de flujo que representa el proceso que produce cuando una persona solicita un turno de palabra:

1. Pulsar el icono para solicitar el turno de palabra.
2. Añadir el nombre de la persona que ha solicitado el turno de palabra.
3. Añadir el nombre de esta persona al final de la lista de espera.

Caso de uso: Carlos es el encargado de realizar la toma de acta en la reunión de la semana que se realiza en la Asamblea de su barrio. Además debe gestionar y conceder los turnos de palabra. Mientras Pepe está exponiendo un idea para solucionar el problema que se ha planteado en la reunión, María levanta la mano y solicita a Carlos el turno de palabra para intervenir. En las siguientes imágenes se muestra el proceso a seguir para solicitar el turno de palabra.

- Hacer click en el icono para solicitar el turno de palabra.

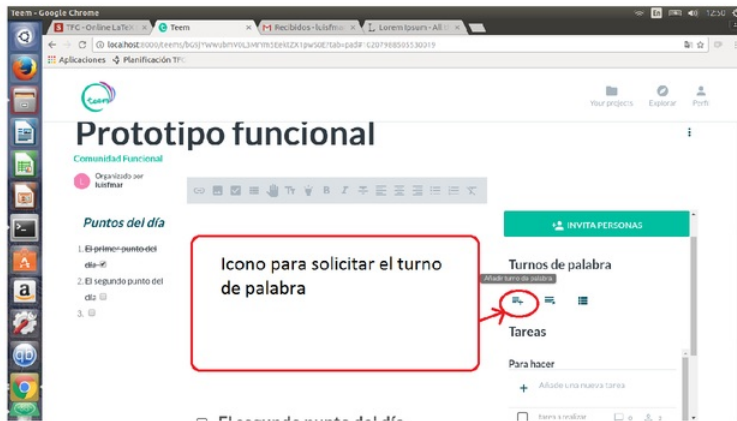


Figura 6.12: Icono del turno de palabra

- Introducir el nombre de la persona.

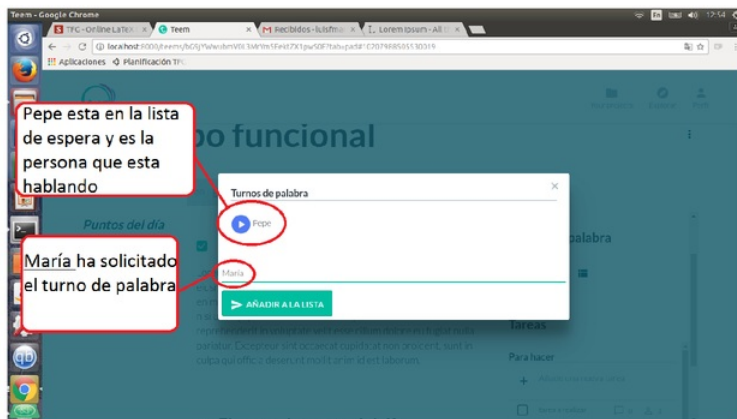


Figura 6.13: Añadir nombre de la persona para el turno de palabra

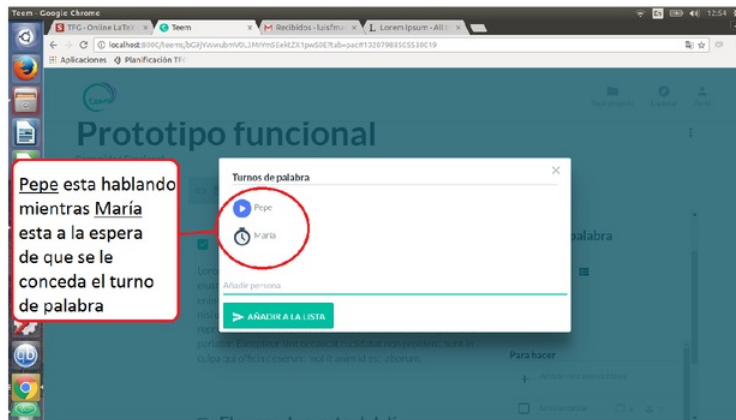


Figura 6.14: Lista de espera del turno de palabra

6.1.3.2. Avanzar al siguiente turno de palabra

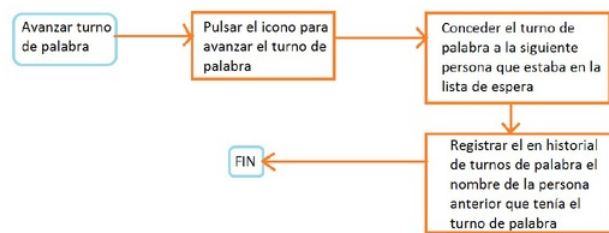


Figura 6.15: Diagrama de flujo para conceder el turno de palabra

La imagen 6.15 es el diagrama de flujo que representa el proceso que produce cuando se concede el turno de palabra a la siguiente persona que está en la lista de espera:

1. Pulsar el icono para conceder el turno de palabra.
2. Conceder el turno de palabra a la siguiente persona según el orden de la llegada que está en la lista de espera.
3. Registrar en el historial de turnos de palabra el nombre de la última persona a la que se le concedió el turno de palabra.

Caso de uso: utilizando el mismo caso de uso de la sección 6.1.3.1, Carlos concede a María el turno de palabra que era la siguiente persona que estaba en la lista de espera de los turnos de palabra. En las siguientes imágenes se muestra el proceso a seguir para conceder el turno de palabra:

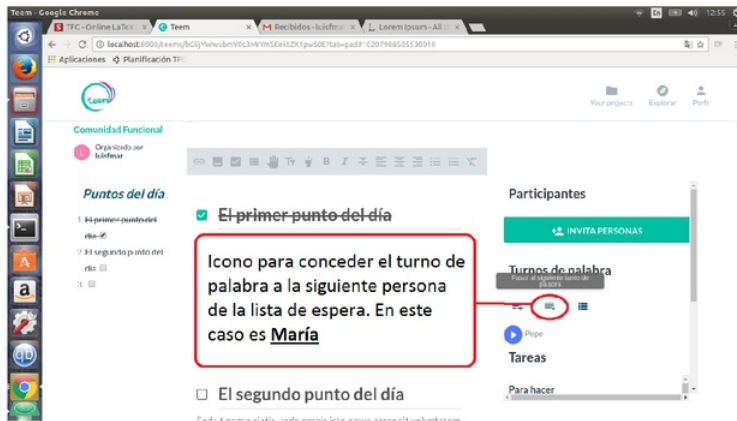


Figura 6.16: Icono para pasar al siguiente turno de palabra de Pepe a María

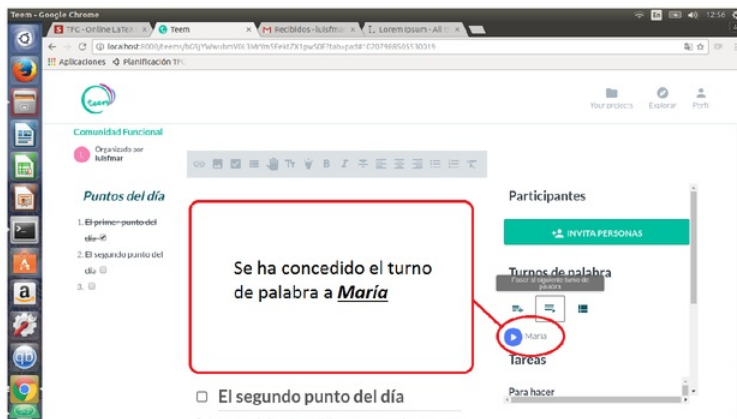


Figura 6.17: Turno de palabra de María

6.1.3.3. Historial de turnos de palabra

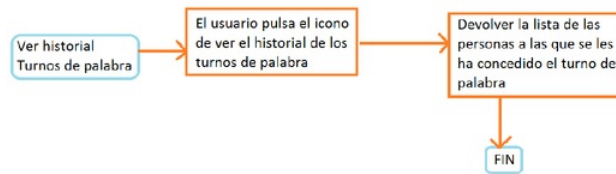


Figura 6.18: Diagrama de flujo para ver el historial de turnos de palabra

La imagen 6.18 es el diagrama de flujo que representa el proceso que produce cuando se va a ver el historial de los turnos de palabra:

1. Pulsar el icono para ver el historial de los turnos de palabra.
2. Devolver la lista de las personas a las que se les ha concedido el turno de palabra.

Caso de uso: utilizando el mismo caso de uso de la sección 6.1.3.1, Carlos quiere ver el historial de los turnos de palabra para poder ver que personas han intervenido más veces en la reunión. En las siguientes imágenes se muestra el proceso a seguir para ver el historial de los turnos de palabra:

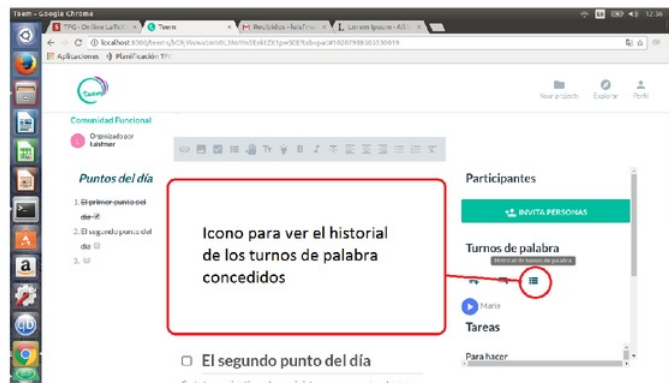


Figura 6.19: Icono de ver historial de los turnos de palabra

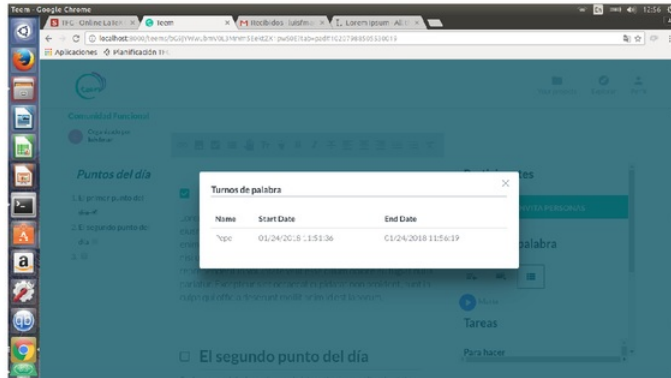


Figura 6.20: Historial de turnos de palabra

6.1.4. Añadir una votación

Cuando los asistentes a una reunión llegan a un acuerdo en un determinado tema, deben registrar en el **acta** la votación del acuerdo al que han llegado todos los asistentes a la reunión. Para solventar el problema de registrar una votación sobre algún tema que se haya debatido en la reunión y con el objetivo de que quede constancia en el acta, hemos **añadido otro widget a la aplicación de Teem-actas** para poder añadir al editor de texto estas votaciones.

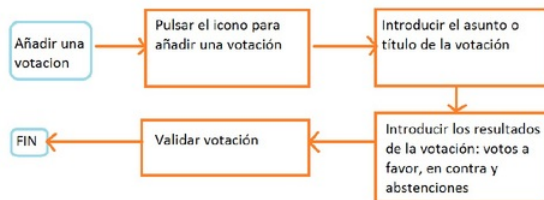


Figura 6.21: Diagrama de flujo para validar una votación

La imagen 6.21 es el diagrama de flujo que representa el proceso que produce cuando se registra en el acta una votación sobre un determinado tema. Con el objetivo de que una votación no puede ser modificada y se puedan alterar los resultados de la votación, dicha votación debe ser **validada**:

1. Hacer click en el icono del widget de una **votación**.
2. Intoducir un título o asunto de la votación.
3. Introducir los resultados de la votación: votos a favor, en contra y las abstenciones.
4. Validar la votación. Si una votación está validada, está no podrá ser modificada.

Caso de uso: Roberto es el encargado de la toma de actas de una reunión y esta utilizando la aplicación de **Teem-actas** para ello. Mientras Roberto esta tomando el acta, los asistentes a la reunión están debatiendo el uso de la aplicación **Google Docs**. para la toma de las futuras actas en sus reuniones. Los miembros de la reunion deciden votar para ver si utilizan o no dicha aplicación y el resultado de la votación es el siguiente: 1 voto a favor, 4 en contra y 0 abstenciones. Roberto debe validar esta votación en la aplicación de **Teem-actas**.

En las siguientes imagenes se muestra el proceso para dar de alta y validar una votación.

- Añadir votación:

1. Hacer click en el icono de añadir **votación**.

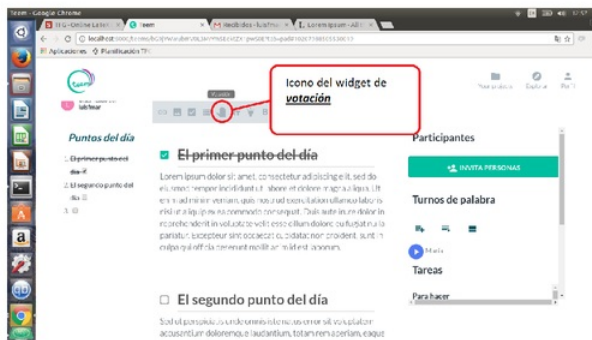


Figura 6.22: Icono del turno de votaciones

2. Poner el asunto de la votación y los resultados de la misma.

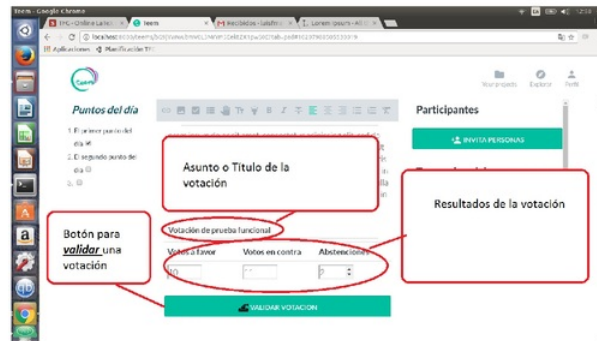


Figura 6.23: Añadir asunto de la votación y resultados

3. Validar votación.

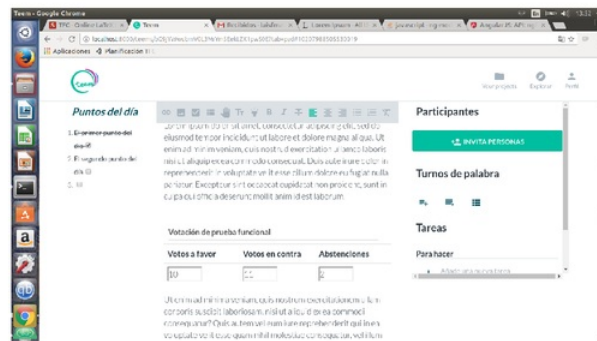


Figura 6.24: Validar votación

6.2. Evaluación con usuarios

En esta fase del proyecto se han realizado pruebas con usuarios probar las tareas que hemos definido en el BackLog y poder analizar en qué partes de la aplicación los usuarios se han encontrado problemas. Con esto se verán los objetivos cumplidos, las posibles mejoras y modificaciones futuras

6.2.1. Objetivos de la evaluación

En esta fase se pretende ver el comportamiento de los usuarios con la aplicación y ver qué funcionalidades de las que se han desarrollado en este proyecto les causan mayores problemas o facilidades. Una vez analizadas veremos qué objetivos se han cumplido y cuáles no, para en un futuro mejorar y modificar aquellas funcionalidades que no hayan aportado una mayor facilidad para la realización de la tarea de una Toma de actas, que es el objetivo principal de esta aplicación. Para ello determinamos si se cumplen los siguientes objetivos:

- El usuario siente el control total sobre la aplicación.
- El usuario es capaz de comprender las actividades que realiza.
- Se cumplen los objetivos iniciales del proyecto (Sección 1.1)

6.2.2. Preguntas de la evaluación

En esta sección se expondrá una lista de preguntas que se quieren responder con los resultados de la evaluación.

- ¿Se pueden visualizar todos los puntos tratados en el acta de un solo vistazo?
- ¿Es fácil de ceder y ver quién tiene el turno de palabra?
- ¿Se pueden ver tareas las tareas a realizar para la siguiente reunión?
- ¿Queda constancia del nombre de la persona que tiene que realizar dicha tarea?
- ¿Se visualizan de forma clara y concisa todos los puntos, temas, votaciones y tareas más relevantes tratados a lo largo de la reunión?

6.2.3. Requisitos para los participantes de la evaluación

La evaluación será realizada para los tipos de perfiles que se corresponden con los definidos en la fase de modelado de personas:

- Usuario ideal
 1. Usuario con mucha experiencia a la hora de la toma de actas. Como mínimo deberá tomar el acta 2 veces al mes.
 2. Familiarizado con el uso de ordenadores o teléfonos móviles.

- Otro perfil
 1. Usuario con experiencia en la toma de actas, pero que suele hacerlo con menos frecuencia.
 2. Poco familiarizado con el uso de ordenadores o teléfonos móviles

6.2.4. Requisitos para las pruebas de la evaluación

Las sesiones de pruebas tuvieron una duración aproximada de 30 minutos. Se realizaron con 2 usuarios, de los cuales todos pertenecen a asociaciones o asambleas y tienen experiencia a la hora de redactar la toma de un acta. Además habrá un moderador para ayudar al usuario y un observador para apuntar las reacciones del evaluado. Para realizar las pruebas se han seguido los siguientes pasos:

- Explicar al usuario en qué consiste la prueba al igual que el motivo y el objetivo que se persigue con ella.
- Darle al usuario una lista con las tareas a realizar durante la sesión.
- Para que la evaluación sea efectiva y no haya ninguna distracción, la sesión se llevará a cabo en un lugar cerrado. Se observarán y anotarán las reacciones del usuario a la hora de realizar las tareas así como las posibles dudas que le pueda surgir a la hora de realizar alguna de las tareas.
- Si el usuario no pudiera resolver alguna tarea, o quedase estancado a la hora de realizar alguna de ellas, el moderador le indicará que pase a la siguiente tarea o le ayudará a terminar con la que estaba. El observador apuntará con detalle por qué razón el usuario no ha podido completar la tarea.
- Por último, el moderador anotará las sugerencias del usuario evaluado, así como las conclusiones del mismo para posteriormente ponerlas en común y analizarlas.

6.2.5. Lista de tareas a realizar

Para realizar las pruebas con usuario se han redactado una tareas a realizar:

1. Crear un proyecto para la realización del Acta.
2. Invitar al proyecto.
3. Dar de alta la lista de Puntos del día del Acta.

4. Añadir texto de lo comentado en un Punto del día.
5. Añadir una tarea a realizar.
6. Asignar una tarea.
7. Añadir un turno de palabra.
8. Añadir un turno de palabra.
9. Pasar al siguiente turno de palabra.
10. Crear una votación
11. Preguntas a responder:
 - (I) ¿Qué parte de la aplicación te ha facilitado más la toma del Acta?
 - (II) ¿Qué tareas te han resultado más complicadas de realizar?
 - (III) ¿Era complicado visualizar algún elemento que sirviera para realizar alguna de las tareas?
 - (IV) ¿Qué es lo que más te ha gustado de la aplicación?
 - (V) ¿Y lo que menos?

6.3. Pruebas con usuarios

6.3.1. Prueba 1

- Perfil de usuario: “Adulto sin conocimiento de herramientas similares”
- Nombre: Manuel Jiménez
- Edad: 49
- Grado de conocimientos informáticos: Medio-Bajo
- Sexo: Hombre
- Número de reuniones mensuales: 2
- Actividad relacionada con la toma de actas: Miembro de una asamblea de vecinos en la que se turnan la toma del acta

Antes de empezar la reunión se pidió que al menos varias personas seguidas pidiesen el turno de palabra durante algún momento de la reunión para interactuar con la parte de pasar al siguiente turno de palabra aunque no fuesen a decir nada muy relevante para la reunión

y así poder hacer el flujo de la interacción completo. Asimismo se pidió que al menos se asignase una tarea a lo largo de la reunión y se realizase una votación sobre cualquier tema.

Un minuto antes de empezar la reunión el usuario crea un nuevo proyecto para añadir los puntos del día e invitar a los hipotéticos usuarios que van a participar de ella.

Antes de añadir un punto del día ha añadido un “Header” o “Encabezado. Viendo que no salía en el apartado de “Puntos del día” ha borrado el Header y ha pulsado en el botón correspondiente a “Punto del día”. Añade varios, todos seguidos sin dejar ningún espacio.

Empieza la reunión. Todas las notas que está tomando las pone después de los puntos del día añadidos anteriormente, no como está pensado para poner, entre punto del día y punto del día para así visualizar de qué se está hablando en cada momento. Cuando uno de los asistentes pide el turno de palabra tarda un poco en darse cuenta de cuál de los botones relacionados con el widget es el correcto para añadir un turno de palabra.

Cuando termina de hablar, finaliza correctamente el turno. Cuando dos usuarios piden el turno de palabra uno después de otro, para la reunión unos segundos para poder añadir a las dos personas en el orden en el que pidieron la palabra. El paso de un turno a otro lo hace rápidamente y sin ningún problema.

Cuando se hace una votación el usuario se dispone a escribir la votación en el editor de textos en vez de utilizar el widget de votaciones, el moderador le indica que hay un botón que le permite realizar la votación sin describirla entera. El usuario no encuentra el botón y al final el moderador le indica cuál es el correcto para ello.

Al final de la reunión se pide un voluntario para realizar la tarea de imprimir el acta y hacerlo circular entre los vecinos de la comunidad. El usuario tarda un poco en añadir la tarea y asignársela a una persona (en este caso, a él mismo ya que se presentó voluntario para hacerlo) pero no necesita ayuda del moderador.

6.3.1.1. Preguntas sobre la aplicación después de la toma del acta

—¿Qué parte de la aplicación te ha facilitado la toma del acta?

—El no tener que escribirlo en papel en el momento o después de la reunión creo que facilita la redacción del acta y hace que sea un poco más concisa cuando un vecino que no ha asistido quiera leerla —aunque tampoco lo hacen muchos si no acuden a la reunión—. Eso está muy bien si estás acostumbrado a escribir con el ordenador

aunque realmente se podría anotar en cualquier sitio que te permita escribir como el “Word”. Creo que la parte que más lo facilita es el poder asignar tareas a una persona o a varias y el tener los puntos del día bien indicados en el lateral para poder leerlos y tacharlos en cualquier momento por si en algún momento la reunión se va por las ramas y no sabéis muy bien por dónde ibais.

-¿Qué tareas te han resultado más complicadas de realizar?

-Lo de los turnos de palabra quizás me ha sido lo más complicado o tedioso más bien ya que a veces no se pide el turno de palabra sino que se habla sin más y andar haciendo click en varios sitios para poder anotar quién habla entorpece un poco el registrarlos de alguna manera.

-¿Era complicado visualizar algún elemento que sirviera para realizar alguna de las tareas?

-Lo de la votación no me parecía que ese botón sirviese para eso, creo que es poco descriptivo, por lo demás los demás iconos y dibujos creo que son bastante intuitivos como para saber lo que hacen aunque lo de poner la letra grande —se refiere a los “Headers”—puede parecer también que quieres añadir un punto del día.

-¿Que es lo que mas te ha gustado de la aplicación?

-La sencillez, que todo esté muy a mano y el apartado de los puntos del día, que los puedas tachar con un simple click para marcarlos que ya se han tratado. Eso y que puedas añadir personas o nombres a las tareas que se van a hacer para la siguiente reunión.

-Además, se pueden escribir comentarios en las tareas, para hacer seguimiento o comentarlas después si alguien no está de acuerdo.

-Pues eso también está bien aunque creo que nosotros no le daríamos mucho uso.

-¿Y lo que menos?

-Que para realizar alguna de las tareas (como los turnos de palabra) quizás es un poco lento y tardas más en hacerlo con lo que hay en la aplicación que apuntándolo simplemente escribiendo.

-Muchas gracias por colaborar, con esto hemos terminado.

6.3.2. Prueba 2

- Perfil de usuario: “Joven con conocimiento de herramientas similares”
- Nombre: Javier García

- Edad: 30
- Grado de conocimientos informáticos: Medio-Alto
- Sexo: Hombre
- Número de reuniones semanales: 1
- Actividad relacionada con la toma de actas: Secretario de una asociación en la universidad

Minutos antes de empezar la reunión se habló con Javier para pedir que al menos un par de personas pidiesen el turno de palabra seguidos y se realizase algún tipo de votación además de asignarse alguna tarea a alguien aunque fuese de forma simbólica para realizar toda la interacción posible con la aplicación.

Nos aseguró que no había problema ya que siempre solían pasar esas cosas sin pedir las, en el caso de no hacer todo lo que se pedía, lo diría para que pudiese registrarlo en la aplicación.

Una vez se le da el control de la aplicación, mientras entra la gente y se preparan para la reunión, crea un nuevo proyecto y hace la interacción de invitar a algunas personas al proyecto.

Empieza a pasar el ratón por todos los botones de la parte superior del editor de textos visualizando los “tooltips” para saber qué hace cada uno. Además, pulsa los distintos botones del widget del turno de palabra para saber qué hace cada uno antes de empezar.

En este punto, pregunta que para qué sirve el botón de “Registro de los turnos” y el moderador le explica que aparecerán registros cuando se añadan turnos de palabra con la duración que han tenido.

Añade rápidamente los puntos del día que se van a tratar en la reunión sin problemas mientras que el director de la asociación comenta con los asistentes lo que se va a tratar y hace varios saltos de línea entre ellos para poder apuntar entre uno y otro lo que se habla cuando se trate ese punto. Además, prueba a clickar varias veces en el “checkbox” del widget de “Puntos del día” para ver que se puede marcar y desmarcar en cualquier momento.

Comienza la reunión y durante el transcurso, varias personas levantan la mano para pedir el turno de palabra, lo registra sin problemas pero tarda un poco, aunque lo hace sin ayuda del moderador, cuando quiere pasar al siguiente turno de palabra. Una vez entiende dónde tiene que pulsar, las siguientes veces que se pasa a otro turno lo hace sin problemas y rápidamente.

Se asignan varias tareas a algunos de los asistentes, lo cual registra en el acta sin problemas, tanto el añadir una tarea como vincular las personas que la van a realizar a dicha tarea. Además, incluye un

comentario en una de ellas para recordar que si la persona a la que está asignada esa tarea no puede realizarla para la semana siguiente, habría que cambiar la persona que la realiza a otra.

Anota una votación sin ningún problema mediante el widget.

Cuando termina la reunión, comprueba que se han registrado los turnos de palabra y parece satisfecho ya que asiente con la cabeza.

6.3.2.1. Preguntas sobre la aplicación después de la toma del acta

-¿Qué parte de la aplicación te ha facilitado la toma del acta?

-Creo que es una herramienta muy útil que reúne la mayoría de las cosas que se necesitan para tomar el acta de una manera muy sencilla. La idea de las tareas y asignárselas a alguien me parece que está muy “guay”. Además, según me habéis comentado, esto lo puede ver todo el mundo aunque no sea de la asociación o no esté invitada al proyecto, ¿no?

-Si, mediante el link se puede compartir y visualizar aunque no se forme parte siquiera de la aplicación, no hace falta estar registrado para verlo.

-Pues eso está muy bien, por lo menos para nosotros que intentamos acercar la asociación y las cosas que organizamos al resto de la gente de la facultad o incluso de otras facultades que estén interesados en nuestras actividades. Podríamos colgar los actas en la página web que tenemos para que la gente los vea o los siga. Además si hay alguien que no haya podido venir, lo de que se pueda ver en tiempo real facilita el seguimiento de la reunión y podría añadir cosas aunque no estuviese presente.

-¿Qué tareas te han resultado más complicadas de realizar?

-Ninguna realmente, aunque creo que lo de los turnos de palabra se podría mejorar sin tener que sacar un popup para añadirlos.

-¿Era complicado visualizar algún elemento que sirviera para realizar alguna de las tareas?

-Bueno, el botón de las votaciones quizás yo lo cambiaría, así de primeras no parece que ese sea el botón para realizar una votación aunque si lees lo que pone cuando pasas por encima con el ratón te queda claro.

-¿Que es lo que mas te ha gustado de la aplicación?

-Me ha gustado mucho la aplicación en general, me parece muy clara y sencilla. Además puedes apuntar muchas cosas y que no todo esté escrito en el acta sino que esté en varios apartados como los puntos del día o las tareas que hay que realizar.

–¿Y lo que menos?

–Nada en particular, tantos pasos al crear el proyecto por decir algo...y lo que ya he comentado de los turnos de palabra.

–¿Usarías la aplicación con regularidad?

–Con lo que he visto hoy si, a lo mejor si encontrase muchos bugs o me diese mucha guerra lo de los turnos de palabra no utilizaría esa parte pero me parece genial. Enhorabuena por vuestro trabajo.

–**Se agradece mucho el comentario. Con esto hemos terminado, gracias por colaborar.**

6.3.3. Prueba 3

- Perfil de usuario: “Joven con conocimiento de herramientas similares”
- Nombre: María García Redondo
- Edad: 23
- Grado de conocimientos informáticos: Medio
- Sexo: Mujer
- Número de reuniones mensuales: 3
- Actividad relacionada con la toma de actas: Miembro de una asociación de juegos de rol

Antes de empezar la reunión la usuaria que va a tomar acta crea un nuevo proyecto para añadir los puntos del día y simular la invitación de los usuarios que van a participar en la reunión.

Mientras entra la gente añade los puntos del día sin problemas probando además varios de los botones que están en la parte del editor de texto para familiarizarse con su funcionamiento y su utilidad. Comenta que se parecen un poco a los botones que tiene Google Docs y que ella lo utiliza a diario para trabajos en su facultad. Añade los puntos del día sin problema.

Se inicia la reunión y antes del primer punto del día anota su nombre y la fecha de hoy. Toma nota de todo lo que se comenta en la reunión, se puede observar que tiene manejo con el uso del ordenador a nivel de mecanografía. Cuando los presentes empiezan a pedir el turno de palabra primero anota los nombres en el editor de textos y cuando se le comenta que hay una herramienta en la aplicación especialmente orientada a éste propósito busca el widget y una vez lo encuentra anota los nombres en ese apartado también, además de lo ya anotado en el texto.

Cuando se comenta la necesidad de ampliar por demanda las existencias de cierto juego se hace una votación en la reunión. La usuaria pasa varias veces por los iconos de la aplicación buscando el widget específico y una vez lo encuentra (transcurridos unos 20 segundos).

A propósito de la votación, que sale con una mayoría a favor, crea una nueva tarea para ello pero la asignación de la tarea la hace escribiéndola en el texto en vez de añadiendo la persona en la parte reservada para ello en el widget de tareas, para no entorpecer la reunión y distraer a la usuaria, el moderador decide comentárselo después de la toma del acta.

Crea otra tarea para proponer el día de la siguiente reunión y vuelve a escribir el encargado de realizarla en el texto.

La reunión finaliza sin mayores contratiempos ni cosas que destacar.

6.3.3.1. Preguntas sobre la aplicación después de la toma del acta

-¿Qué parte de la aplicación te ha facilitado la toma del acta?

-Bueno, la parte de los puntos del día y las votaciones está bien porque no hace falta escribirlo todo y para algunos de nuestros socios que a veces toman acta y que no están tan acostumbrados a escribir a ordenador puede ser bastante útil, además de que cualquiera puede ver lo que se ha escrito si no acuden a la reunión, cosa que pasa mucho y luego es un lío porque hay veces que no están del todo de acuerdo con las decisiones que se han tomado o simplemente no se la leen porque tienen que venir aquí a verlo y leerlo.

-¿Qué tareas te han resultado más complicadas de realizar?

-Bueno, complicado nada, supongo que es cuestión de acostumbrarse a qué hace cada cosa.

-¿Era complicado visualizar algún elemento que sirviera para realizar alguna de las tareas?

-Lo de los turnos de palabra creo que podría ser de otra manera más sencilla de anotar, cuando son reuniones con mucha gente a veces muchos quieren hablar y tener que añadirlos todos en el pop up pues puede resultar un poco pesado.

-¿Que es lo que mas te ha gustado de la aplicación?

-No sabría decir, lo de los puntos del día quizás, está chulo eso de poder tacharlos y que aparezcan todos juntos, se ve bastante claro y se puede repasar fácilmente.

-¿Y lo que menos?

–Lo de los puntos del día, por decir algo. Ah, y que en las votaciones sólo haya esas opciones, hay veces que se realizan votaciones sobre qué cosas podríamos añadir a nuestro inventario de juegos y la gente vota por unos o por otros. Eso no lo podríamos hacer con la aplicación, a no ser que no lo haya visto.

–**No, de momento no se puede aunque en un futuro se podría implementar**

–Pues estaría bien, yo creo que sería una buena mejora.

–**Gracias por prestarnos su tiempo.**

–No hay de qué, a vosotros.

6.3.4. Prueba 4

- Perfil de usuario: “Adulto sin conocimiento de herramientas similares”
- Nombre: Óscar Garrido
- Edad: 53
- Grado de conocimientos informáticos: Bajo
- Sexo: Hombre
- Número de reuniones mensuales: 2
- Actividad relacionada con la toma de actas: Portavoz de una comunidad de vecinos

Unos minutos antes de empezar la reunión se pide al usuario que proponga alguna votación y que se asigne alguna tarea aunque sea simbólica con el propósito de explorar todas las posibilidades de la aplicación y así utilizar el máximo número de componentes de la aplicación.

Mientras que se espera a que lleguen el resto de vecinos el usuario crea un nuevo proyecto para la toma del acta y se le pide que añada los puntos del día que se van a tratar. El usuario nos pregunta que cómo hace eso. El moderador le invita a que pruebe usando los botones del editor de textos, los cuales va pulsando uno a uno hasta que da con el widget de puntos del día. Añade varios, unos detrás de otros al inicio del acta.

Empieza la reunión y el usuario empieza a tomar acta después de todos los puntos del día. Transcurridos unos minutos el usuario pregunta al moderador que cómo se hace una votación y el moderador le indica el botón correspondiente para hacerlo. El usuario corta un momento la reunión antes de empezar el siguiente punto del día, los

cuales no está marcando como tomados, para proponer una votación ficticia ya que, según nos comentan, las votaciones se realizan en situaciones en las que hay que hacer una derrama o cualquier otro tema muy puntual. Una vez el moderador le ha indicado el botón, nos comenta que el botón no le parece adecuado para indicar una votación pero lo hace sin problemas.

Se asigna una tarea a uno de los presentes de la reunión, la cual no asigna a la persona. El moderador le indica que puede hacerlo pulsando sobre el icono correspondiente y el usuario lo hace, aunque comenta que no lo ve claramente.

La reunión finaliza sin haberse realizado ninguna intervención de los presentes y le pedimos al usuario que simule la petición de turnos de palabra. El usuario busca por la aplicación el widget de votaciones pero al no encontrarlo el moderador le tiene que indicar cómo se apunta un nuevo turno de palabra. El usuario comenta que es muy difícil hacer eso en medio de la reunión y darse cuenta en el botón que tiene que pulsar. No lo ve útil ya que ellos no suelen apuntar las intervenciones de la gente porque cuando las hay, hablan muchos a la vez y no tienen demasiado orden.

6.3.4.1. Preguntas sobre la aplicación después de la toma del acta

-¿Qué parte de la aplicación te ha facilitado la toma del acta?

-Lo de los puntos del día es lo más útil y fácil yo creo.

-¿Qué tareas te han resultado más complicadas de realizar?

-Bueno, yo no uso mucho el ordenador y a mi me parece que es muy complicado si no estás acostumbrado, hay muchos botones e iconos y al principio creo que no se entiende nada. Yo prefiero apuntar en papel.

-¿Era complicado visualizar algún elemento que sirviera para realizar alguna de las tareas?

-Lo de los turnos de palabra y las votaciones no lo hubiese descubierto yo creo si no me lo hubieses dicho.

-¿Que es lo que mas te ha gustado de la aplicación?

-Sinceramente no me ha gustado mucho en general, ya digo que no uso mucho el ordenador y me parece muy difícil tomar acta con uno. Y menos haciendo uso de todas las cosas que tiene.

-¿Y lo que menos?

-Lo que menos...Lo de los turnos de palabra por decir algo, pero ya digo que no me acaba de convencer en general.

-Gracias por todo, nos ha sido de gran ayuda.

Capítulo 7

Resultados y Trabajo Futuro

7.1. Introducción

En este ultimo apartado se analizarán y discutirán los resultados en conjunto de todo el Trabajo de Fin de Grado, además de algunas de las tareas que se podrían hacer en un futuro para mejorar la aplicación.

Antes de nada, nos gustaría señalar que el **código del desarrollo está disponible en GitHub**, el código es Open Source y esta bajo la **licencia de GNU Affero General Public License**. El repositorio esta en Teem-Actas [11].

7.2. Reparto de Trabajo entre los integrantes del grupo

Antes de empezar a exponer como ha sido el reparto del trabajo, **queremos dejar constancia** de que este proyecto en un principio estaba programado para realizarse entre 4 personas. En un principio este trabajo los integrantes del grupo eran: Daniel David Abellán Serrano, Julia Gracia Sancha, Juan Manuel Mármol Pérez y Luis Fernando Martín Ruiz. Por circunstancias ajenas y problemas internos en el grupo, como falta de comunicación y de interés por parte de Julia y Daniel, ellos decidieron no finalizar el Trabajo de Fin de Grado, pero nosotros, Luis Fernando y Juan Manuel, decidimos seguir adelante con el proyecto. También cabe recalcar que nuestros directores del TFG **Samer** y **Antonio** nos animaron a seguir en este proyecto y terminarlo, lo cual les agradecemos ya que sin ellos no habría sido posible conseguirlo. Para la realización de este Trabajo

de Fin de Grado, el reparto de tareas fue el siguiente:

- **Diseño e investigación:** cabe recalcar que ésta fue la única tarea que se realizó equitativamente entre los 4 integrantes iniciales del proyecto. Todos asistíamos a reuniones periódicas iniciales, para planificar la investigación y diseñar los prototipos. Durante las entrevistas, nos dividimos en parejas para realizarlas. Posteriormente cada miembro realizó los prototipos de la aplicación y se hizo una puesta en común para generar el prototipo final con la herramienta Pencil. En resumen toda la parte del Diseño Guiado por Objetivos la realizamos en conjunto.
- **Implementación de la aplicación:** todo el desarrollo fue llevado a cabo por parte de Juan Manuel y Luis Fernando. En ésta parte, uno de nosotros (Luis Fernando) ya tenía algo de experiencia a la hora de programar con Angular y JavaScript, así que se decidió hacer “Pair Programming” para que la persona que no tenía tanta experiencia con éstas tecnologías pudiera aprender de la otra.
- **Evaluación con usuarios:** planificamos juntos vía Skype la dinámica de las evaluaciones y como las realizaríamos, haciendo un total de cuatro evaluaciones y como resultado final discutir los resultados y sacar las conclusiones de cada prueba.
- **Elaboración de la memoria:** aquí debemos señalar que al inicio del proyecto nuestro compañero **Daniel David Abeillán Serrano** generó una primera versión del esqueleto de la memoria, y participó en algunos puntos de la misma, como en el estado del arte y en mucha menor medida en las tecnologías utilizadas. El resto de la memoria ha sido realizada equitativamente por parte de Luis Fernando y Juan Manuel, repartiéndose las secciones de la memoria, y siempre revisando lo que escribía el compañero, modificando o añadiendo cosas nuevas para completar cada sección.

7.3. Trabajo Futuro

Hay implementaciones y trabajo que nos hubiese gustado acabar y que están en una fase avanzada aunque no totalmente desarrollada.

El principal objetivo que se podría finalizar totalmente es el de desplegar con Travis en un servidor para que la aplicación pueda ser visible para comunidades colaborativas sin tener que desplegar en “local”.

Otro de los objetivos que están muy cerca de ser totalmente satisfechos a nivel de implementación es el de obtener un archivo con las partes principales del acta como los puntos del día, votaciones y tareas junto con texto añadido manualmente por medio de un botón que remarca las partes que van a ir al archivo. La implementación que tenemos está preparada para devolver mediante una serie de variables y rangos la estructura del documento organizada según está en el texto. Quedaría por hacer la implementación de un botón que generase ese archivo para ser mandado por correo electrónico o imprimirlo, por ejemplo.

La funcionalidad más relevante y que más interesaba a todos los usuarios de la fase de investigación aunque también la más insatisfacible era la de “**Automatizar la toma de actas**”. Nos hubiese gustado poder desarrollar esta funcionalidad pero creemos firmemente que está actualmente fuera de nuestro alcance.

Hay otras funcionalidades que tras la fase de investigación y diseño de la aplicación (Sección 4) pudimos observar que podían ser susceptible de ser desarrollado para mejorar y satisfacer necesidades más específicas o concretas de algunos de los posibles usuarios las cuales no eran tan ambiciosas como la automatización de la toma de actas. Algunas de ellas, con las cuales creemos que se podría realizar un trabajo interesante partiendo de nuestra aplicación son:

- La posibilidad de grabar el sonido de la reunión, teniendo así un registro inequívoco de la completitud de la misma y adjuntarlo al acta.
- La posibilidad de grabar en vídeo (además del sonido) la reunión adjuntando el vídeo al acta.

Como reflexión posterior al trabajo realizado pensamos en la posibilidad de realizar la tarea de automatizar la toma de actas mediante la combinación de la funcionalidad comentada acerca de la grabación del audio de la reunión y un sistema de inteligencia artificial capaz de transcribir el audio recibido a un editor de textos como el que tenemos en la aplicación. Ciertamente es que, de ésta forma sería difícil detectar los distintos turnos de palabra que se han producido a lo largo de la reunión además de la imposibilidad de asignar los distintos turnos en un orden sin la intervención de una persona encargada de ello u otro sistema.

La única posibilidad que se nos ocurre de integrar casi todas las funcionalidades de las que dispone actualmente nuestra aplicación y combinarlas en una toma de actas automatizada sería implementar un sistema de inteligencia artificial bastante complejo, al menos en

nuestra opinión, el cual, además de registrar lo hablado en la reunión de la forma que hemos comentado anteriormente, pudiese diferenciar las distintas tonalidades de voz de los presentes para poder interpretar que una persona distinta del interlocutor principal o portavoz ha tomado el turno de palabra. Asimismo, éste sistema, combinado con la funcionalidad de la grabación en vídeo de la reunión, podría asignar el orden de los turnos de palabra mediante sensores de movimiento o una interpretación especial cuando alguno de los presentes levanta la mano en lo que se está grabando para pedir el turno de palabra. Para funcionalidades como las votaciones o la asignación de tareas a personas se podría realizar un sistema de reconocimiento de ciertas palabras o “keywords” que el sistema reconociese como el inicio y el cierre de una votación o tarea.

Aún con ésto, creemos que habría usuarios que no estarían totalmente satisfechos con el resultado ya que quedan tareas que, bajo nuestro punto de vista, son extremadamente difíciles de automatizar con una IA como podría ser la funcionalidad de señalar (para posteriormente incluir en un resumen) las partes más importantes de la reunión que son susceptibles de ser enviadas como resumen de la reunión a los distintos socios o miembros de la entidad colaborativa.

Conclusiones

Teem-actas es una aplicación web con el objetivo de facilitar la toma de actas en reuniones y asambleas de proyectos o comunidades colaborativas. Para ello empezamos analizando posibles herramientas que pudieran competir con la nuestra, buscando aquellas cosas que nos pudieran distinguir del resto de aplicaciones. Esto nos llevo a pensar que la mejor forma de conseguir eso sería sacandoles mayor información a nuestros usuarios mas potenciales, sin que ellos se dieran cuenta concretamente de ello, para que una vez desarrollada la herramienta, la pudieran probar y nos dieran su “feedback” sin saber qué funcionalidades les daría nuestra aplicación.

La idea principal de éste proyecto era **analizar, investigar y mejorar** en la medida de lo posible las principales deficiencias tecnológicas existentes que se ofrecen actualmente para comunidades colaborativas, concretamente para los **problemas más comunes** que tenían estas comunidades a la hora de tener reuniones y a la tomas de actas.

Además de ésto, todas las tecnologías que pudiésemos aprender para fortalecer nuestra formación académica era un punto a favor por muy inexpertos que fuésemos en un principio.

Investigando y entrevistando a distintas comunidades colaborativas nos dimos cuenta de que los problemas principales de las comunidades eran muy parecidos o prácticamente iguales entre unas y otras.

Puestos a mejorar una plataforma ya existente como es Teem nos dimos cuenta de que había muy pocas aplicaciones que satisficiesen las principales tareas de una cosa tan básica como la toma del acta de cualquier reunión.

A través de nuestra aplicación queremos dar mayor viabilidad, visibilidad y trazabilidad a todos los proyectos de comunidades colaborativas así como mejorar su experiencia y calidad en las reuniones.

Creemos que en éstos aspectos y gracias a las potentes herramientas de Teem, éste tipo de comunidades podrían dar un salto

cualitativo gracias a nuestra aplicación ya que nos hemos centrado en conseguir dar solución a los principales problemas comunes en cualquier toma de actas de cualquier comunidad.

El hecho de partir de una aplicación consistente y probada como es **Teem**, fue fundamental para poder llevar a cabo el desarrollo de este Trabajo de Fin de Grado. Por otro lado, la **investigación y diseño de prototipos inicial** que realizamos nos ayudó mucho a la hora de identificar las necesidades que tenían los usuarios a los que está destinada esta aplicación. En concreto nos facilitó el hecho de identificar claramente aquellas necesidades que no necesitábamos desarrollar, porque Teem ya nos las ofrecía, y las que necesitaban ser desarrolladas de cero o ser modificadas para adaptarlas a el contexto de este trabajo. Esto nos permitió priorizar la lista de tareas a realizar, no en el orden que nosotros como desarrolladores nos hubiese gustado, sobre todo porque nosotros creíamos que algunas funcionalidades le darían más valor a la aplicación, pero nos dimos cuenta haciendo las entrevistas, que esas funcionalidades que para nosotros parecían tener más valor, realmente al usuario final no le aportaban lo que nosotros pensábamos. Así pues una de las conclusiones que hemos sacado, es que **la fase de investigación y prototipado es fundamental a la hora de desarrollar cualquier aplicación**, ya que al final, los usuarios los que determinarán la calidad de una aplicación.

Gracias a los distintos widgets implementados y la base que nos proporciona Teem y SwellRT hemos conseguido crear una aplicación sostenible, sencilla y utilizable por una gran variedad de usuarios.

Debido a los problemas técnicos y a los retrasos que tuvimos durante todo el desarrollo del proyecto todavía no hemos podido crear un entorno estable donde desplegar nuestra aplicación totalmente operativa.

Metiéndonos un poco en profundidad a nivel del desarrollo y la parte visual de la aplicación, consideramos, dadas las opiniones recibidas, que hemos optado en algunos casos por una solución orientada aspectos más técnicos y que pueden generar una sensación de menor control de la aplicación como en el widget de "*Turnos de palabra*", el cual pensamos que se podría depurar la idea para darle un toque menos complejo con la misma funcionalidad. Algo parecido pasa con el icono de las votaciones, tras las pruebas de usuario coincidimos en que no quedaba suficientemente relacionado el icono con la funcionalidad.

Por otro lado, a este mismo nivel, pensamos que hemos acertado de lleno con la forma de plantear los "*Puntos del día*" del acta así como con la inclusión en el widget de las "*Needs*" o "*Tareas*" de la

posibilidad de asignarle una tarea a una o más personas, sean o no usuarios de la aplicación. Éste pensamiento viene del sentimiento general de los usuarios hacia ésta parte de la aplicación y de el “feedback” que recibimos a éste respecto.

Las pruebas finales con usuarios, nos sirvieron para identificar aquellas funcionalidades desarrolladas que no cumplieron con las expectativas del usuario final, lo cual nos llevo a replantear en que punto del desarrollo nos habíamos equivocado. Gracias a esto nos hemos dado cuenta de que un buen número de pruebas con usuarios finales en un entorno fiable es necesario y muy importante a la hora de analizar la aplicación desarrollada y en qué puntos se puede mejorar. Lo que los desarrolladores vemos que puede ser más intuitivo o fácil de utilizar muchas veces dista bastante de la realidad de un usuario final. Sobre todo hay que tener en cuenta la gran diversidad de usuarios a los que puede y se busca llegar con la aplicación, los cuales pueden ser incompatibles unos con otros ya que la amplitud de perfiles de usuarios a la que se pretende llegar puede producir estos efectos.

Lo que hemos podido concluir de éste razonamiento es que a veces es realmente complicado llegar a todo el espectro de usuarios que buscas satisfacer con tu aplicación y se tiene que buscar o bien un denominador común que satisfaga, aunque no completamente, a una gran parte de los perfiles o bien centrarse en una mayoría de perfiles a los cuales puedes orientar más directamente tu aplicación llegando a cumplir con las expectativas de éstos aunque no terminando de satisfacer a la minoría restante.

Algunas cosas que se podrían mejorar, como se pueden observar tras las pruebas con usuarios descritas anteriormente (Sección 6.3) podrían ser:

- Darle un nuevo icono más descriptivo o más fácil de entender para todos los usuarios al widget de votaciones para que sea mucho más reconocible.
- Enfocar de una manera más sencilla y directa la asignación de turnos de palabras de forma que no saque un “popup” cada vez que quieras añadir un nuevo turno.

Otra de las conclusiones que hemos podido obtener a partir del desarrollo realizado es que una mejora que pretende ser sencilla y fácilmente accesible puede convertirse en una tarea realmente difícil de realizar e implementar de una manera concisa cuando trabajas con varias tecnologías integradas en una plataforma. El grado de dificultad de implementar dichas mejoras puede verse incrementado

por la necesidad de interactuar con tecnologías desconocidas y con una curva de aprendizaje no demasiado suave cuando no se está acostumbrado a utilizar dicha tecnología. Integrar nuevas mejoras en sistemas con una estructura ya montada y jerarquizada puede llevar mucho trabajo si no comprendes bien las tecnologías que se utilizan.

Para terminar, la **aplicación cumple con un mínimo de requisitos para solventar los mayores problemas a la hora de tomar un acta**, y al ser totalmente “Open Source”, cualquier persona puede ayudarnos a seguir mejorandola, ya que pensamos que está aplicación tiene muchísimo potencial para ayudar a comunidades colaborativas a mejorar la difusión de sus reuniones, con el fin, de que nuevos usuarios se unan a ellas para crear un mundo mejor.

Conclusions

Teem-actas is a web application with the objective of facilitating the taking of minutes in meetings and assemblies of collaborative projects or communities. For this we started by analyzing possible tools that could compete with ours, looking for those things that could distinguish us from other applications. This led us to think the best way to achieve this would be to get more information to our most potential users, without them realizing this, so that once the application was developed, they could try it and give us their “feedback” without knowing what functionalities our application would give them.

The main idea of this project was to **analyze, investigate** and **improve** as much as possible the main existing technological deficiencies that are currently offered for collaborative communities, specifically for **most common problems** that those communities had when they have meetings.

In addition to this, all the technologies that we could learn to strengthen our academic training was a point in favor no matter how inexperienced we were in the beginning.

Investigating and interviewing different collaborative communities, we realized that the main problems of the communities were very similar or practically the same between each other.

Thinking on improve an existing platform such as Teem we realized that there were just a few applications that met the main tasks of a thing as basic as taking the minutes of any meeting.

Through our application we want to give greater viability, visibility and traceability to all projects of collaborative communities as well as improve their experience and quality in meetings.

We believe that in these aspects and thanks to the powerful tools of Teem, this type of communities could make a qualitative leap thanks to our application since we have focused on getting to solve the main common problems in any record taking of any community.

The fact of starting from a consistent and tested application like **Teem** was fundamental to be able to carry out the development

of this Final Degree Project. On the other hand, the **initial prototype research and design** that we did helped us a lot when it came to identifying the needs that the users had for this application. Specifically, it made it easier for us to clearly identify those needs that we didn't need to develop, because Teem already offered them to us, and those that needed to be developed from scratch or modified to adapt them to the context of this project. This allowed us to prioritize the list of tasks to be performed, not in the order that we as developers would have liked, especially because we believed that some features would give more value to the application, but we realized doing the interviews, that those functionalities that for us seemed to have more value, the end user didn't really provide what we thought. So one of the conclusions we have obtained, is that **the research phase and prototyping is fundamental when developing any application**, because in the end, users will determine the quality of any application.

Thanks to the different widgets implemented and the base provided by Teem and SwellRT, we have managed to create a sustainable, simple and usable application for a wide variety of users.

Due to the technical problems and the delays we had during the whole development of the project, we have not yet been able to create a stable environment where we can deploy our fully operational application.

Going a little deeper into the development and the visual part of the application, we consider, given the opinions received, that we have opted in some cases for a solution oriented to technical aspects and that can generate a feeling of less control of the application as in the "*Speaking time*" widget, which we thought we could debug the idea to give it a less complex touch with the same functionality. Something similar happens with the voting icon, after the user tests we agreed that the icon was not sufficiently related to the functionality.

On the other hand, at this same level, we think that we have been fully successful with the way we present the "*Items on the agenda*" of the session as well as with the inclusion in the widget of "*Needs*" or "*Tasks*" of the possibility of assigning a task to one or more people, whether or not they are users of the application. This thought comes from the general feeling of the users towards this part of the application and the "feedback" that we had received to this.

The final tests with users helped us to identify those functionalities developed that did not meet the expectations of the end user, which led us to rethink at what point of development we had made a mistake. Thanks to this we have realized that a good number of

tests with end users in a reliable environment is necessary and very important when analyzing the developed application and at what points it can be improved. What developers see that can be more intuitive or easy to use is often far from the reality of an end user. Above all, it is necessary to take into account the great diversity of users that can and will be reached with the application, which may be incompatible with each other since the breadth of user profiles that is intended to reach can produce these effects.

Some things that could be improved, as can be seen after the tests with users described above (Section 6.3)) could be:

- Give a new icon more descriptive or easier to understand for all users to the voting widget to make it much more recognizable.
- Focus in a simpler and more direct way the assignment of speaking times so you don't get a "popup" every time you want to add a new turn.

Another of the conclusions that we have been able to obtain from the development carried out is that an improvement that pretends to be simple and easily accessible can become a really difficult task to carry out and implement in a concise way when you work with several technologies already integrated in a platform. The degree of difficulty of implementing these improvements can be increased by the need to interact with unknown technologies and with a learning curve that is not too smooth when you are not used to using such technology. Integrating new improvements in systems with an already assembled and hierarchical structure can take a lot of work if you do not understand well the technologies that are used.

To finish, the **application meets a minimum of requirements to solve the biggest problems when taking a record**, and being totally "Open Source", anyone can help us to continue improving it, since we think that this application has a lot of potential to help collaborative communities to improve the diffusion of their meetings, in order that new users join them to create a better world.

Bibliografía

- [1] *Apache wave (incubating)*, <https://incubator.apache.org/wave/>.
- [2] *Css*, https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada.
- [3] *Dom*, https://es.wikipedia.org/wiki/Document_Object_Model.
- [4] *Extensible messaging and presence protocol*, <https://matrix.org/>.
- [5] *Grasia (group of agent-based, social and interdisciplinary applications)*, <https://grasia.fdi.ucm.es>.
- [6] *Incubator wave, mirror of apache wave*, <https://github.com/haphut/incubator-wave>.
- [7] *Internet*, <https://es.wikipedia.org/wiki/Internet>.
- [8] *Introducing json.*, <http://json.org/>.
- [9] *Material design*, <https://material.io/guidelines/>.
- [10] *The p2p value project*, <https://p2pvalue.eu/>.
- [11] *Repositorio de la aplicación teem-actas*, <https://github.com/Raistm/teem-actas/>.
- [12] *Scrum methodology*, <http://scrummethodology.com/>.
- [13] *Singleton*, <https://es.wikipedia.org/wiki/Singleton>.
- [14] *Swellrt*, <http://swellrt.org/>.
- [15] *Syntactically awesome style sheets*, <http://sass-lang.com/>.
- [16] *What is npm?*, <https://www.npmjs.com/>.
- [17] *What is open source?*, <https://opensource.com/resources/what-open-source>.
- [18] *Why angularjs?*, <https://angularjs.org/>.

- [19] *World wide web consortium*, <https://www.ecma-international.org/publications/standards/Ecma-262.html>.
- [20] M. A. ALVAREZ, *Single page application*, <https://desarrolloweb.com/articulos/que-es-una-spa.html>.
- [21] D. ARENZANA, *Los principios de usabilidad web de jakob nielsen*, <https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/>.
- [22] B. ATLISSIAN, *Why git for your organization*, <https://www.atlassian.com/git/tutorials/why-git>.
- [23] S. BASU, *Travis-ci: What, why, how*, <https://code.tutsplus.com/tutorials/travis-ci-what-why-how-net-34771>.
- [24] V. BEAL, *Html5: what is it?*, <https://www.webopedia.com/TERM/H/HTML5.html>.
- [25] K. BROWN, *What is github, and what is it used for?*, <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>.
- [26] D. CHANDH, *What is ide?*, <https://www.quora.com/What-is-IDE-1>.
- [27] W. W. W. CONSORTIUM, *About w3c*, <https://www.w3.org/>.
- [28] R. J. R. DÁRTHENAY, *El principio kiss*, <https://rizoroy.com/2016/08/09/el-principio-kiss-del-ingles-keep-it-simple-stupid-hazlo-simple-estupido/>.
- [29] D. DOCS., *Overview*, <https://docs.docker.com/compose/overview/>.
- [30] ECURED, *Herramienta case*, https://www.ecured.cu/Herramienta_CASE.
- [31] E. EVANS, *Domain-driven design*.
- [32] EVOLUS, *An open-source gui prototyping tool that's available for all platforms.*, <http://pencil.evolus.vn/>.
- [33] N. FOUNDATION, *Why the hell would you use node.js*, <https://medium.com/the-node-js-collection/why-the-hell-would-you-use-node-js-4b053b94ab8e>.
- [34] I. FREE SOFTWARE FOUNDATION, *What is free software?*, <https://www.gnu.org/philosophy/free-sw.es.html>.
- [35] —, *What is gnu?*, <https://www.gnu.org/home.en.html>.

- [36] G. GUZMÁN, *Introducción y usos del mobile backend as a service*, <https://platzi.com/@galileoguzman/>.
- [37] C. HOPE, *Google docs*, <https://www.computerhope.com/jargon/g/google-docs.htm>.
- [38] L. HORN, *What is google drive?*, <https://gizmodo.com/5904653/what-is-google-drive>.
- [39] V. P. IN AGILE PROJECT MANAGEMENT DEVOPS, *Pair programming*, <https://www.versionone.com/agile-101/agile-software-programming-best-practices/pair-programming/>.
- [40] D. INC., *Overview*, <https://www.docker.com/what-docker>.
- [41] G. INC, *Google web toolkit*, <http://www.gwtproject.org/>.
- [42] —, *Jsinterop*, <http://www.gwtproject.org/doc/latest/DevGuide.html>.
- [43] W. INC., *Whatsapp. simple. seguro. mensajería fiable.*, <https://www.whatsapp.com/?l=es>.
- [44] M. P. JOHANSSON, *Why use bower when there is npm?*, <https://www.quora.com/Why-use-Bower-when-there-is-npm>.
- [45] A. LAFUENTE, *¿ qué es el procomún?*, http://medialab-prado.es/article/video_que_es_el_procomun, (2007).
- [46] N. LEGAULT, *What is e-learning*, <https://community.articulate.com/series/getting-started/articles/what-is-e-learning>.
- [47] MICROSOFT, *¿ qué es skype?*, <https://support.skype.com/es/faq/FA6/que-es-skype>.
- [48] C. V. MIR, *Elinor ostrom y oliver williamson premio nobel de economía 2009*, <http://www.abc.es/20091012/economia-economia/elinor-ostrom-oliver-williamson-200910121304.html>.
- [49] M. NIETO, *Diseño guiado por metas*, https://www.alzado.org/articulo.php?id_art=660.
- [50] NUBEDIGITAL, *¿ qué es vmware?*, <http://nubedigital.co/clientes/knowledgebase/58/Que-es-VMware.html>.
- [51] V. P, *What is cordova and how does it work ?*, <https://blogs.sap.com/2014/07/27/what-is-cordova-and-how-does-it-work/>.

- [52] D. L. PABLO OJANGUREN, ALEJANDRO GARRIDO, *Jetpad, online open source text editor*, <https://jetpad.net/>.
- [53] A. POWELL-MORSE, *Domain-driven design - what is it and how do you use it?*, <https://airbrake.io/blog/software-design/domain-driven-design>.
- [54] L. PROJECT TEAM, *An introduction to latex*, <https://www.latex-project.org/about/>.
- [55] A. PURDILA, *The web designer's guide to gulp*, <https://webdesign.tutsplus.com/courses/the-web-designers-guide-to-gulp/lessons/what-is-gulp>.
- [56] J. D. QUIÑÓNEZ, *Sharelatex, editor de latex colaborativo, online y con decenas de plantillas*, <https://www.whatsnew.com/2013/09/08/sharelatex-editor-latex-online-colaborativo/>.
- [57] J. ROCHELEAU, *What is google's material design?*, <https://envato.com/blog/introduction-material-design/>.
- [58] M. ROUSE, *Nosql (no solo sql)*, <http://searchdatacenter.techtarget.com/es/definicion/NoSQL-No-Solo-SQL>.
- [59] F. RUBIRA, *¿qué es google hangouts y para qué sirve?*, https://www.elconfidencialdigital.com/opinion/tribuna_libre/Google-Hangouts-sirve_0_2089591050.html.
- [60] S. TECHNOLOGIES, *What is ajax and where is it used in technology?*, <https://www.seguetech.com/ajax-technology/>.
- [61] TEEM, *Teem repository*, <https://github.com/Grasia/teem>.
- [62] A. M. UTRATEL, *Teem - the app for collaborative communities*, <https://p2pvalue.eu/teem-the-app-for-collaborative-communities/>.

Índice de figuras

2.1. Imagen representativa del procomún	7
2.2. Pantalla de Blogger	9
2.3. Pantalla de WordPress	9
2.4. Mapa de Mindmeister	10
2.5. Pantalla de MeetingBurner	10
2.6. Pantalla de WebEx	10
2.7. Captura de pantalla de Fuze Meeting	11
2.8. Pizarra de Twiddla	11
2.9. Captura de pantalla de Google Docs	12
3.1. Representa la interacción entre los template-controller- service	20
4.1. Preguntas de las entrevistas	33
4.2. Perfil de usuario	37
4.3. Prototipo Boceto 1	42
4.4. Prototipo Boceto 1	42
4.5. Prototipo Boceto 2	43
4.6. Prototipo Boceto 2	43
4.7. Punto del día	45
4.8. Tachar Punto del día	45
4.9. Pedir turno de palabra	46
4.10. Persona que tiene el turno de palabra	46
4.11. Inicio Pantalla principal	47
4.12. Modal para asignar una tarea	47
4.13. Inicio Pantalla principal	48
4.14. Modal para realizar una votación	48
4.15. Visibilidad del estado del sistema	49
4.16. Relación entre el sistema y el mundo real	49
4.17. Control y libertad del usuario - Parte 1	50
4.18. Control y libertad del usuario - Parte 2	50
5.1. Teem-actas	53

5.2. Diagrama de clases para añadir un nuevo punto del día	56
5.3. Diagrama de secuencia para añadir un nuevo punto del día	57
5.4. Diagrama de secuencia para marcar un Punto del día como completado	57
5.5. Diagrama de clases para asignar una tarea a un usuario	59
5.6. Diagrama de secuencia para asignar una tarea a una persona	60
5.7. Diagrama de clases para añadir una votación	61
5.8. Diagrama de secuencia para añadir una votación	62
5.9. Diagrama de secuencia para validar una votación	63
5.10. Diagrama de clases para solicitar un turno de palabra	65
5.11. Diagrama de secuencia para perder un turno de palabra	67
5.12. Diagrama de secuencia para pasar al siguiente turno de palabra	67
5.13. Diagrama de secuencia para mostrar el historial de turnos de palabra	68
6.1. Diagrama de Flujo para dar de alta un Punto del día	71
6.2. Icono del widget para un Punto del día	72
6.3. Añadir el título a un Punto del día	72
6.4. Diagrama de flujo para marcar un Punto del día como concluido	73
6.5. Marcar un Punto del día como concluido	74
6.6. Diagrama de flujo para asignar una tarea	75
6.7. Icono del widget de tareas	76
6.8. Añadir el asunto de la tarea	76
6.9. Icono para asignar una tarea	77
6.10. Nombre de la persona a la que se va asignar una tarea	77
6.11. Diagrama de flujo para solicitar un turno de palabra	78
6.12. Icono del turno de palabra	79
6.13. Añadir nombre de la persona para el turno de palabra	79
6.14. Lista de espera del turno de palabra	80
6.15. Diagrama de flujo para conceder el turno de palabra	80
6.16. Icono para pasar al siguiente turno de palabra de Pepe a María	81
6.17. Turno de palabra de María	81
6.18. Diagrama de flujo para ver el historial de turnos de palabra	82
6.19. Icono de ver historial de los turnos de palabra	82
6.20. Historial de turnos de palabra	83
6.21. Diagrama de flujo para validar una votación	83
6.22. Icono del turno de votaciones	84

<i>ÍNDICE DE FIGURAS</i>	115
6.23. Añadir asunto de la votación y resultados	85
6.24. Validar votación	85