



Escuela  
Politécnica  
Superior

# Gestión de asociaciones



Grado en Ingeniería Informática

## Trabajo Fin de Grado

Autor:

Francisco García Mora

Tutor/es:

Jaume Aragonés Ferrero

Junio 2021



Universitat d'Alacant  
Universidad de Alicante



## Resumen

El presente documento recoge, de forma técnica, el proceso de desarrollo de un proyecto tecnológico, que trata de dar solución a un problema concreto planteado a continuación. En él, se detalla el proceso en su totalidad, desde la evaluación previa del problema con el planteamiento de la solución propuesta, hasta la implementación de esta, pasando por el análisis de requisitos, el diseño de las funcionalidades propuestas junto a su desarrollo y posterior implementación.

El desarrollo y los avances tecnológicos producidos, cada vez de una forma más exponencial, en los últimos años han permitido desarrollar grandes herramientas tecnológicas, pero también crear una gran brecha digital entre quienes han sabido, o han podido, adaptarse a esta nueva era y quienes no han tenido esa posibilidad o no han sabido ejecutarla de forma correcta.

Además, la aparición de la pandemia de la covid19<sup>1</sup> ha dejado ver las carencias que esta brecha ha causado en la población. Desde grandes empresas hasta pequeños comercios, todos han tenido que dar un paso al frente y renovar su forma de funcionar.

Con ello nos enfocamos a cómo una fiesta tradicional, que en la actualidad no tiene esta posibilidad, puede renovarse e informatizarse para llevar la totalidad de sus gestiones de una forma óptima.

En este proyecto se ha desarrollado una herramienta informática que permite a un conjunto de asociaciones gestionar toda su documentación y procesos orgánicos de forma totalmente digital.

---

<sup>1</sup> <https://cnecovid.isciii.es/covid19/>



## Motivación, justificación y objetivo general.

Meses antes de comenzar el período de selección de propuesta para mi Trabajo de Fin de Grado (TFG<sup>2</sup>) tenía en mente varias ideas para desarrollar, sin embargo, como he citado anteriormente, la situación creada por la aparición de la covid19 hizo que dejase a un lado los otros dos proyectos que tenía planteados y enfocarme plenamente en este.

Toda mi vida he pertenecido a una *foguera* de Alicante<sup>3</sup>, y sé cómo funcionan a nivel interno y qué gestiones se desarrollan en ellas, ya que, además, he tenido que desempeñar la labor de secretario durante dos ejercicios. Es por lo que, el confinamiento me hizo dar prioridad a esta herramienta y aportar una aplicación informática que, de este soporte informático a las comisiones de hogueras y barracas, del que actualmente carecen.

Este proyecto me ha permitido poner en práctica casi todas las asignaturas cursadas durante el grado de ingeniería informática. Desde la asignatura más básica cursada de programación hasta la más avanzada de gestión de calidad de software. En este proceso se ha hecho de 0 a 100 un aplicativo web, trabajando el diseño e implementación de una base de datos, un aplicativo que ofrece servicios *REST*<sup>4</sup> y un cliente web.

Además, durante el desarrollo de este TFG he aprendido herramientas como Docker<sup>5</sup>, que me han permitido virtualizar todo el aplicativo y poder desplegarlo con algo tan simple como es un comando. Lo que permite en cierto modo, que, cuando este proyecto se ponga en marcha, su despliegue sea más fácil y rápido.

---

<sup>2</sup> <https://eps.ua.es/es/ingenieria-informatica/trabajos-fin-de-grado.html>

<sup>3</sup> [https://es.wikipedia.org/wiki/Hogueras\\_de\\_Alicante](https://es.wikipedia.org/wiki/Hogueras_de_Alicante)

<sup>4</sup> <https://rockcontent.com/es/blog/api-rest/>

<sup>5</sup> <https://www.docker.com/>

En los últimos meses del grado conocí a un profesor que, sin ser titular de Universidad, me supo transmitir su pasión por el desarrollo de software y vi en él todo cuanto necesitaba para ser guiado en este proyecto. Por tanto, una vez abierto el plazo, decidí presentar esta propuesta como TFG y elegir al profesor Jaume Aragonés Ferrero.

## Agradecimientos

En un momento delicado, que atraviesa la fiesta de *les Fogueres de Sant Joan*, provocado en cierto modo por la pandemia a causa de la covid19, me gustaría agradecer a todos y cada uno de los *foguerers* y *barraquers* que han querido aportar su granito de arena en este proyecto. Mi eterno agradecimiento a ellos, no sólo por su implicación en este proyecto, sino por mantener viva la llama de nuestra fiesta más querida.

A mis compañeros y compañeras del grado, agradecerles el apoyo que nos hemos brindado durante estos largos años juntos.

A mis compañeros más cercanos, que en muchas ocasiones hemos formado grupo de trabajo, agradecerles el compromiso que hemos tenido, así como el diálogo siempre empleado en los momentos más complicados.

A mi familia, por el apoyo fundamental que siempre me han ofrecido.

A mi tutor, Jaume Aragonés, por todo lo que me ha enseñado durante estos meses y el buen trato que me ha dado. Ha sido un auténtico placer trabajar contigo, y sin duda, todo un acierto al elegirte como tutor.

Y por último, pero no por ello menos importante, a todas y cada una de las personas, que de forma directa o indirecta, me han hecho ser la persona que soy hoy en día.

Gracias.





## Citas

“Si después de esta pandemia no cambia nada,  
es que nada habremos aprendido.”

Popular

“¿Qué es la vida? Un frenesí.  
¿Qué es la vida? Una ilusión,  
una sombra, una ficción;  
y el mayor bien es pequeño,  
que toda la vida es sueño,  
y los sueños, sueños son.”

D. Pedro Calderón de la Barca



# Índice de contenidos

1.	Introducción .....	1
2.	Estudio de viabilidad .....	5
3.	Planificación .....	11
4.	Estado del arte .....	19
5.	Objetivos .....	33
6.	Metodología .....	35
7.	Análisis y especificación .....	39
8.	Diseño .....	63
9.	Implementación .....	91
10.	Resultados .....	95
11.	Pruebas.....	97
12.	Conclusiones y trabajo futuro .....	99

# Índice de figuras

Figura 1. Matriz DAFO. ....	6
Figura 2. Web Federació de Les Fogueres de Sant Joan. ....	21
Figura 3. Intranet Federació de Les Fogueres de Sant Joan. ....	21
Figura 4. Intranet Federació de Les Fogueres de Sant Joan. Perfil usuario. ....	22
Figura 5. Panel de inicio de cucunver. ....	24
Figura 6. Gestión económica en Cucunver. ....	25
Figura 7. Gestión de actividades en Cucunver. ....	25
Figura 8. Gestión de informes (documentos) en GONG. ....	26
Figura 9. Gestión económica en GONG. ....	26
Figura 10. Gestión económica en Central Station CRM. ....	27
Figura 11. Control de tareas en Central Station CRM. ....	28
Figura 12. Gráfico del flujo de proceso en cascada. ....	35
Figura 13. Planificación inicial de Trello para la versión v1.1 (iteración 1). ....	36
Figura 14. Detalle de la información de una tarea en Trello. ....	37
Figura 15. Revisión sprint v1.1 ....	38
Figura 16. Diagrama EER capa artística. ....	64
Figura 17. Diagrama EER capa económica. ....	65
Figura 18. Diagrama EER capa socios. ....	66
Figura 19. Gráfico estructura contenedores Docker con mysql y configuración NGINX y Let's Encrypt en el servidor. ....	72
Figura 20. Gráfico estructura contenedores Docker con mysql y configuración NGINX y Let's Encrypt en el servidor. ....	72
Figura 21. Documentación API Gefoc. ....	73
Figura 22. Endpoints tag /artistas API Gefoc. ....	74
Figura 23. Request endpoint POST /artistas API Gefoc. ....	74
Figura 24. Response endpoint POST /artistas API Gefoc. ....	75
Figura 26. Schema Acto API Gefoc. ....	75
Figura 27. Mockups – BackOffice Master – Inicio de sesión. ....	80
Figura 28. Mockups – BackOffice Master – Página de inicio. ....	80
Figura 29. Mockups – BackOffice Admin – Home. ....	81
Figura 30. Mockups – BackOffice Admin – Ejemplos de modales diversos. ....	81
Figura 31. Mockups – BackOffice Admin – CMS. ....	82
Figura 32. Mockups – BackOffice Admin – CMS. ....	82
Figura 33. Mockups – Intranet Socio – Página de inicio. ....	83
Figura 34. Mockups – Intranet Socio – Modales. ....	83
Figura 35. Mockups – Intranet Socio – Socios. ....	84
Figura 36. Mockups – Intranet Socio – Eventos. ....	84
Figura 37. Mockups – Intranet secretario – Convocar reunión. ....	85
Figura 38. Mockups – Intranet secretario – Generar documento genérico. ....	85
Figura 39. Mockups – Intranet secretario – Generar acta. ....	86
Figura 40. Mockups – Intranet Socio – Pagos. ....	86
Figura 41. Mockups – Intranet Tesorero – Cobros. ....	87
Figura 42. Mockups – Intranet Tesorero – Movimientos. ....	87

Figura 43. Mockups – Intranet Tesorero – Cuotas.....	88
Figura 44. Mockups – WEB – Página inicial. ....	89
Figura 45. Tablero trello v1.1 .....	91

# Índice de tablas

Tabla 1. Distribución horas durante el proceso de desarrollo. ....	11
Tabla 2. Distribución de horas de trabajo durante el proceso de desarrollo con los meses ponderados. ....	12
Tabla 3. Distribución de las horas de trabajo en función de la fase del proyecto. ....	13
Tabla 4. Planificación temporal del proyecto. Distribución de horas de trabajo con las tareas correspondientes. ....	14
Tabla 5. Planificación de los sprints de la fase de diseño e implementación. ....	15
Tabla 6. Tabla comparativa de funcionalidades principales de cada herramienta. ....	29
Tabla 7. Tabla Asociaciones_Ejercicios de la BD. ....	67
Tabla 8 Tabla Ejercicios de la BD. ....	67
Tabla 9 Tabla Asociaciones de la BD. ....	68
Tabla 10 Tabla Asociaciones_Socios de la BD. ....	69
Tabla 11 Tabla Socios de la BD. ....	69
Tabla 12. Tabla Familias de la BD. ....	70
Tabla 13. Tabla Asociaciones_Admin de la BD. ....	70
Tabla 14. Tabla Templates de la BD. ....	70
Tabla 15. Tabla Templates_Activas de la BD. ....	71
Tabla 16. Tabla Configuraciones de la BD. ....	71
Tabla 17. Tabla Configuraciones_Asociaciones de la BD. ....	71
Tabla 18. Endpoints tag “actos” ....	76
Tabla 19. Endpoints tag “documentos” ....	76
Tabla 20. Endpoints tag “paginas” ....	77
Tabla 21. Endpoints tag “reuniones” ....	77
Tabla 22. Endpoints tag “socios” ....	78
Tabla 23. Endpoints tag “templates” ....	78

# Índice de abreviaturas

API<sup>6</sup>: *Application Programming Interface*

SSL<sup>7</sup>: *Secure Sockets Layer*

---

<sup>6</sup> <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

<sup>7</sup> [https://www.verisign.com/es\\_LA/website-presence/online/ssl-certificates/index.xhtmll](https://www.verisign.com/es_LA/website-presence/online/ssl-certificates/index.xhtmll)







## 1. Introducción

En la ciudad de Alicante se celebra, de forma anual durante el mes de junio, una fiesta popular llamada *Les Fogueres de Sant Joan*<sup>8</sup>. Esta fiesta ha sido declarada de Interés Turístico Internacional<sup>9</sup> y Bien de Interés Cultural Inmaterial. Además<sup>10</sup>, son las fiestas oficiales de la ciudad de Alicante.

*Les Fogueres de Sant Joan*, en adelante *les Fogueres*, representan la esencia del carácter mediterráneo de la ciudad alicantina. Tienen su culmen con el ritual de la *Nit del Foc*<sup>11</sup>, celebrada en la noche del 24 de junio. Fecha en la que se realiza la tradicional *cremà* de los monumentos *fogueroles*.

Aunque la mayor parte del peso de esta fiesta se desarrolle durante el mes de junio, a lo largo del año, los comisionados de *les Fogueres*, conocidos popularmente como *fogueroles*, desarrollan otro tipo de actividades con fechas fijas, a las que se añaden actividades eventuales como son almuerzos populares, barbacoas nocturnas o fiestas temáticas entre otros.

En Valencia, existe, una celebración similar a *Les Fogueres de Sant Joan*, las *Fallas*<sup>12</sup> de Valencia. Esta celebración hermana fue declarada Patrimonio Inmaterial de la Humanidad por la UNESCO en noviembre de 2016<sup>13</sup>. *Les fogueres*, en cambio, fueron declaradas Bien de Interés Cultural Inmaterial por la Generalitat Valenciana en el mes de julio de 2014. Siguiendo los pasos de las Fallas de Valencia, *les Fogueres* solicitarán en breve la declaración que posee

---

<sup>8</sup> <https://www.hogueras.es/historia/>

<sup>9</sup> <https://www.hogueras.es/historia/les-fogueres-se-convierten-en-fiestas-de-interes-turistico-internacional/>

<sup>10</sup> <https://sercomunicacion.es/las-hogueras-de-san-juan-bic-inmaterial/>

<sup>11</sup> <https://www.informacion.es/alicante/2019/06/24/hemos-contado-crema-hogueras-san-5386190.html>

<sup>12</sup> [https://es.wikipedia.org/wiki/Fallas\\_de\\_Valencia](https://es.wikipedia.org/wiki/Fallas_de_Valencia)

<sup>13</sup> <https://www.levante-emv.com/valencia/2016/11/30/fallas-son-patrimonio-humanidad-unesco-12308726.html>

actualmente su fiesta hermana. Por tanto, en vista a un futuro, es previsible que también reciban dicho nombramiento y ello suponga un aumento en la atracción turística de esta fiesta.

Para el correcto desarrollo de la fiesta, los presidentes de esta han de realizar gestiones durante todo el ejercicio festero. En la actualidad, una gran parte de estas gestiones se realizan de forma manual. Entre estas gestiones se incluyen la convocatoria de reuniones, redacción de actas y documentos, autorizaciones para actividades con menores, pagos de cuotas, gestión de altas y bajas...etc. En definitiva, una gran cantidad de acciones que podrían llevarse acabo de una forma más eficiente a través de una informatización del proceso.

Además, estas comisiones, que pueden ser de foguera o de barraca, no disponen de un espacio web corporativo, por lo que tampoco pueden comercializar su marca y llegar a un público más amplio utilizando otra herramienta que no sean las redes sociales más conocidas.

Existe un órgano rector llamado *Federació de Les Fogueres de Sant Joan*<sup>14</sup>, compuesta y elegida por los *foguerers* y *barraquers*. Esta *Federació* sí que dispone de una web propia<sup>15</sup> y una intranet<sup>16</sup> para gestión, sin embargo, estas herramientas no están disponibles para cada asociación, lo que produce que actualmente las asociaciones, que quieran disponer de una web o de algún sistema informático, tengan que contratar profesionales, creando cada una su sistema propio.

---

<sup>14</sup> <https://intranet.hogueras.es/federacion/>

<sup>15</sup> <https://www.hogueras.es/>

<sup>16</sup> <https://intranet.hogueras.es/>

En la actualidad, la mayoría de estas comisiones intentan realizar páginas web propias y crear cuentas en diferentes redes sociales para poder tener una mayor visibilidad y un punto donde introducir una comunicación más fluida en plena era digital. No obstante, la mayor parte de los gestores o dirigentes de estas asociaciones son personas de avanzada edad o con un conocimiento prácticamente nulo de las nuevas tecnologías.

Además, con la aparición de la covid19, se han tenido que suprimir muchos eventos (prácticamente todos) incluyendo las gestiones presenciales. Esto ha supuesto un grave impacto para la mayoría de las comisiones de *fogueres*, ya que no sólo las gestiones administrativas se realizaban de forma presencial y en papel, sino que cerca del 90% de las gestiones económicas también se realizan de esta forma.

Todo ello lleva a plantear como propuesta la creación de un sistema informático que permita a todas las comisiones, de foguera o de barraca, gestionar todos sus procesos de forma informática. Este sistema debe ser un sistema que sea efectivo para todas las asociaciones que sean dadas de alta en él, sin permitir un cruce de datos privados entre asociaciones.

Además, la herramienta desarrollada ha de servir como web pública para habilitar a las asociaciones un espacio de publicidad donde puedan darse a conocer sin depender de otras aplicaciones como pueden ser las redes sociales.



## 2. Estudio de viabilidad

Realizar un estudio de viabilidad es realmente importante a la hora de realizar un proyecto.

Para poder planificar y desarrollar un aplicativo, es necesario realizar un estudio previo de viabilidad y analizar los posibles riesgos a los que se pueda enfrentar el proyecto. Desarrollando además una serie de medidas que puedan reducir los posibles impactos de estos riesgos en caso de darse.

Para poder realizar correctamente el estudio de viabilidad, primero realizaremos un análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades)<sup>17</sup> que nos permita determinar qué estrategia debemos afrontar gracias a la matriz en la que se indican estas propiedades.

Además, también realizaremos un análisis de riesgos para poder establecer planes de contingencia que ayuden a minimizar el impacto de estos riesgos.

### 2.1. Análisis DAFO

Siguiendo las directrices ya citadas, este método de análisis permite diseñar la estrategia en la que se basará el proyecto. Para ello se elaborará un mapa, o matriz, en el que se indican las propiedades DAFO, que pueden ser positivas o negativas, de origen interno o externo.

Las debilidades y fortalezas serán consideradas de origen interno mientras que las amenazas y oportunidades serán consideradas de origen externo.

---

<sup>17</sup> [https://es.wikipedia.org/wiki/An%C3%A1lisis\\_FODA](https://es.wikipedia.org/wiki/An%C3%A1lisis_FODA)

La siguiente matriz, es por tanto el resultado de este análisis DAFO:

	Positivos	Negativos
Origen interno	<u>Fortalezas</u> <ul style="list-style-type: none"> <li>Habilidades adquiridas como Ingeniero en Informática.</li> <li>Disposición de recursos y herramientas necesarias para trabajar.</li> <li>Amplio conocimiento del dominio del aplicativo</li> </ul>	<u>Debilidades</u> <ul style="list-style-type: none"> <li>Desconocimiento de despliegue de aplicativos webs</li> <li>Limitación de recursos humanos para un proyecto que podría ser más grande de lo esperado.</li> </ul>
Origen externo	<u>Oportunidades</u> <ul style="list-style-type: none"> <li>Baja competencia</li> <li>Influye en un gran número de usuarios</li> <li>Amplia base de usuarios para testear el proyecto.</li> </ul>	<u>Amenazas</u> <ul style="list-style-type: none"> <li>Aumento de la competencia</li> <li>Bajo nivel de destreza con herramientas tecnológicas por parte del usuario.</li> </ul>

Figura 1. Matriz DAFO.  
(Fuente: Elaboración propia)

- **Propiedades de origen interno.**

Dentro de las propiedades de origen interno, en el lado positivo están las fortalezas mientras que en el lado negativo están las debilidades.

En cuanto a fortalezas considero que principalmente están las habilidades adquiridas como Ingeniero en Informática, entre las que destacaría el análisis del sistema, la toma de requisitos, el diseño de la solución al problema, la implementación de estos diseños...etc. En definitiva, todo el conocimiento adquirido es una buena base que sirve como fortaleza para el proyecto.

Otra fortaleza encontrada es que actualmente, gracias al grado universitario, dispongo de las herramientas y recursos suficientes y necesarios para llevar a cabo el proyecto.

Por último, y quizá la más importante, el amplio conocimiento del dominio del aplicativo, y es que un ingeniero cuanto más conozca el dominio sobre el que trabaja, mayor facilidad tendrá para realizar el desarrollo del proyecto.

En cuanto a las debilidades del proyecto, la más grande que podemos observar es el desconocimiento sobre el despliegue de aplicativos en la web, ya que a lo largo del grado no se han facilitado este tipo de conocimientos. Esta debilidad podría provocar un retraso de la puesta en producción del aplicativo, es por lo que estimo que es la debilidad más grande, pero a su vez, la que se puede solucionar de una forma muy simple.

La última de las debilidades estimadas es la posible limitación de recursos humanos, ya que este proyecto al ser una idea propia podría crecer de forma casi ilimitada, por tanto, la única limitación que se va a poner sobre este proyecto es la propia capacidad de trabajo.

- **Propiedades de origen externo**

En cuanto a las propiedades de origen externo, se encuentran las oportunidades y amenazas.

En cuanto a las oportunidades, la principal que he podido observar es que la escasa oferta de alternativas se traduce en una baja competencia, a esto habría que añadir, que el dominio del aplicativo no es muy conocido, y por tanto en la actualidad no existe un desarrollo activo de una herramienta similar.

Otra de las oportunidades que pueden existir es que dentro de *les Fogueres* hay una amplia base de usuarios, por lo que este aplicativo sería usado por un público objetivo grande.



Por último, esta misma amplia base de usuarios, que sería el objetivo de la aplicación, servirían como usuarios de prueba, por lo que el aplicativo podría ser ampliamente testado antes de ser lanzado al mercado.

En cuanto a las amenazas, el principal problema lo podríamos encontrar en la baja destreza con herramientas tecnológicas por parte de los usuarios del aplicativo. Además, habría que sumar un posible aumento de la competencia en caso de que este aplicativo tenga éxito.

## 2.2. Análisis de riesgos

Además del análisis DAFO, vamos a analizar los riesgos que pueden alterar la planificación del proyecto y por tanto hacer peligrar el éxito del aplicativo.

Cualquier tipo de proyecto está expuesto a riesgos que pueden afectar al desarrollo de este, por lo que es necesario identificarlos antes de que se puedan producir contratiempos. Además, esto garantiza que podamos plantear posibles planes de contingencia que se puedan aplicar en caso de producirse algún riesgo.

Los posibles riesgos identificados son:

- Aumento de la carga de trabajo de otras asignaturas: el TFG está enmarcado dentro del ámbito académico, por lo que una mala planificación temporal, podría derivar en una acumulación tan grande de trabajo que provoque una saturación del ritmo de trabajo. Para poder solventar este problema, se dispondrá de un calendario de forma anticipada con todas las entregas de proyectos académicos actuales.
- Falta de conocimiento de las tecnologías empleadas: las tecnologías empleadas en este proyecto son conocidas por parte del desarrollador, por lo que en un principio no debería suponer un riesgo para tener en cuenta. Sin embargo, como he comentado

anteriormente, el despliegue en un servidor es algo desconocido por el momento, por lo que podría suponer un retraso a la hora de la puesta en producción. Para poder solventar este problema, se cuenta con el apoyo del tutor del TFG, así como de cursos online para el despliegue de proyectos.

- Fallo del equipo: este factor es un factor de riesgo que puede ocurrir en cualquier proyecto y en cualquier momento. Es quizá el riesgo más grave ya que podría conllevar una pérdida de información y por tanto del trabajo realizado. Sin embargo, existen muchas herramientas para poder solventar este problema, por ejemplo, el empleo de un sistema de control de versiones de forma descentralizada como GIT<sup>18</sup>, que permite trabajar en remoto, nos proporcionaría una copia de seguridad continua del trabajo realizado.
- Enfermedades: este es otro factor que también puede ocurrir en cualquier momento y en cualquier proyecto. El tiempo destinado a este proyecto es de 300 horas, por lo que una enfermedad inesperada podría restar horas necesarias al proyecto. Este año, en especial, a causa de la pandemia provocada por la covid19 puede suponer un riesgo elevado para el proyecto, en mi caso la contraje, por lo que podría suponer una pequeña ventaja contra esta. Para poder solventar este problema, se podría proponer una modificación del trabajo planificado, dosificando las horas afectadas y aumentando las horas de otro día.

Una vez realizado el análisis DAFO y el análisis de riesgos, podemos valorar con objetividad la viabilidad del proyecto.

En cuanto a los riesgos analizados ninguno resulta ser una amenaza grave, ya que son problemas comunes a cualquier proyecto y además se ha propuesto un posible plan de contingencia.

---

<sup>18</sup> <https://es.wikipedia.org/wiki/Git>

Por otro lado, en el análisis DAFO, podríamos ver que las características positivas tienen un mayor peso que las negativas. Es, por tanto, posible realizar una planificación aprovechando estas oportunidades y fortalezas para contrarrestar las propiedades negativas.

Finalmente, podemos concluir que el proyecto es viable y se puede llevar a cabo con garantías de éxito.

### 3. Planificación

Si realizar un análisis de riesgos es importante, no menos lo es la planificación del proyecto. Una buena planificación junto a una, también, buena estimación de las tareas asegura que el proyecto se pueda realizar en el tiempo deseado. Esto nos ayuda a establecer una fecha límite de entrega, pudiendo definir varias fases de análisis, planificación, diseño, implementación y despliegue del proyecto.

En este apartado no vamos a entrar en la metodología empleada, sino en la planificación temporal en cuanto a qué tiempo se puede dedicar al proyecto.

El arranque del proyecto estaba previsto para el mes de octubre, teniendo clara la idea a presentar, sin embargo, se tuvo que retrasar debido a problemas de salud. Finalmente, el proyecto arrancó en el mes de febrero.

Teniendo en cuenta que al proyecto se le han de dedicar aproximadamente 300 horas, la planificación se realizará repartiendo estas 300 horas entre los meses de febrero, marzo, abril y mayo.

*Tabla 1. Distribución horas durante el proceso de desarrollo.  
(Fuente: Elaboración propia)*

Mes	Febrero	Marzo	Abril	Mayo
<b>Horas</b>	75	75	75	75

Sin embargo, teniendo en cuenta los factores que pueden retrasar el proyecto, como son la jornada laboral y los proyectos de asignaturas de la universidad, realizamos una reasignación de las horas. Para esto, es necesario conocer que los meses de abril y mayo habrá más horas disponibles dado que, en abril hay vacaciones de Semana Santa y a finales

de abril finalizo mi contrato laboral. Atendiendo a esto último, la estimación inicial quedaría de la siguiente forma:

*Tabla 2. Distribución de horas de trabajo durante el proceso de desarrollo con los meses ponderados.  
(Fuente: Elaboración propia)*

Mes	Febrero	Marzo	Abril	Mayo
<b>Horas</b>	40	46	110	105
<b>Horas diarias</b>	2	2	5	5
<b>Días de trabajo</b>	20	23	22	21

En total quedan repartidas 301 horas de trabajo en cuanto a planificación temporal, pero es necesario realizar una estimación en cuanto a las fases de desarrollo, para ello, primero vamos a realizar una estimación de la carga de trabajo que conllevará cada fase de estas.

Las fases de desarrollo, como hemos indicado antes, son: Análisis, planificación, diseño, implementación y despliegue. Brevemente, las tareas a desarrollar en cada fase son las siguientes:

- **Análisis:** En esta primera fase se ha de realizar una evaluación de cuál es el alcance del proyecto, qué requerimientos tiene el proyecto y qué funcionalidades se proponen para dar solución al proyecto.
- **Planificación:** en un segundo paso, debemos realizar la planificación de las tareas surgidas tras el análisis de las funcionalidades que se van a implementar en el proyecto.

- **Diseño:** posteriormente, se ha de confeccionar el diseño de los *mockups* y las pantallas que va a tener el aplicativo. Además, en esta fase incluiremos el diseño de la arquitectura, la aplicación de servicios *Rest* y la base de datos para el proyecto.
- **Implementación:** es la fase en la que se realizará la programación como tal de todo el proyecto, dónde se dará forma al aplicativo, que finalmente presentaremos.
- **Despliegue:** como fase final realizaremos el despliegue del aplicativo en un servidor para que el proyecto sea público y este visible para todo el mundo.

Así pues, las horas estimadas para cada fase son las siguientes:

*Tabla 3. Distribución de las horas de trabajo en función de la fase del proyecto.  
(Fuente: Elaboración propia)*

Fase	Análisis	Planificación	Diseño e implementación	Despliegue
<b>Horas</b>	90	15	165	30

Trasladando las horas estimadas a porcentajes, destinaremos un 30% del total a la fase de análisis, ya que un buen análisis hará que el proyecto tenga una buena base, y por consiguiente una mayor fiabilidad. Un 5% será destinado a la planificación, ya que, a pesar de su importancia, no es una fase demasiado grande. En cuanto al diseño e implementación se realizará en una única fase, distinguiendo el nivel de diseño y de implementación. Al nivel de diseño, se le asignará un 15% del tiempo. En cuanto a la implementación, dispondrá del 40% del tiempo total del proyecto, siendo la fase más importante en cuanto a carga de trabajo. Por último, para el despliegue se dedicará un 10%, aunque en caso de que se produzca algún adelanto en el resto de las fases, se podrá disponer de más tiempo para esta última fase.

En consecuencia, la planificación temporal del proyecto será la siguiente:

Tabla 4. Planificación temporal del proyecto. Distribución de horas de trabajo con las tareas correspondientes.  
(Fuente: Elaboración propia)

Fase y tareas	Horas	Fecha inicio	Fecha límite
<b>Análisis</b> <ul style="list-style-type: none"> <li>• Estudiar aplicativos existentes</li> <li>• Toma de requerimientos</li> <li>• Analizar diferencias con otras aplicaciones</li> <li>• Definir servicios o funcionalidades</li> </ul>	90	1 de febrero	1 de abril
<b>Planificación</b> <ul style="list-style-type: none"> <li>• Definir historias de usuario</li> <li>• Realizar estimación de las historias de usuario</li> <li>• Definir límite del proyecto</li> </ul>	15	2 de abril	6 de abril
<b>Diseño e implementación</b> <ul style="list-style-type: none"> <li>• Realizar <i>mockups</i> para el proyecto</li> <li>• Diseñar estructura de la <i>API</i></li> <li>• Diseñar la base de datos</li> <li>• Instalar <i>software</i> necesario</li> <li>• Implementar diseño de la base de datos</li> <li>• Crear base de datos</li> <li>• Crear estructura <i>API</i> en <i>Swagger</i><sup>19</sup></li> <li>• Crear proyecto <i>API</i> de servicios <i>Rest</i> en <i>nodejs</i><sup>20</sup> (<i>express</i><sup>21</sup>)</li> <li>• Crear proyecto “intranet” en <i>Angular</i><sup>22</sup></li> <li>• Crear proyecto “web” en <i>Angular</i></li> <li>• Crear proyecto “BackOffice” en <i>Angular</i></li> </ul>	165	7 de abril	21 de mayo
<b>Despliegue</b> <ul style="list-style-type: none"> <li>• Crear archivos <i>Dockerfile</i><sup>23</sup> para desplegar cada proyecto</li> <li>• Crear imágenes <i>Docker</i> para cada proyecto</li> </ul>	30	24 de mayo	31 de mayo

<sup>19</sup> <https://swagger.io/>

<sup>20</sup> <https://nodejs.org/es/>

<sup>21</sup> <https://expressjs.com/es/>

<sup>22</sup> <https://angular.io/>

<sup>23</sup> <https://docs.docker.com/engine/reference/builder/>

<ul style="list-style-type: none"> <li>• Crear <i>docker-compose</i><sup>24</sup> para poder desplegar todos los proyectos</li> <li>• Crear servidor</li> <li>• Gestionar DNS<sup>25</sup> y rutas NGINX<sup>26</sup></li> <li>• Desplegar con <i>Docker</i></li> </ul>			
---	--	--	--

### 3.1. Planificación de las iteraciones.

Para el desarrollo de la fase de diseño e implementación realizaremos una serie de iteraciones, conocidas por su nombre en inglés como *sprints*<sup>27</sup>, concretamente hemos planteado 4 *sprints*.

Al final de cada iteración tendremos desarrollada una nueva versión lista para la entrega, por lo que cada iteración aportará valor al proyecto, permitiendo realizar en momentos puntuales despliegues del aplicativo con las funcionalidades plenamente activas.

Estas cuatro iteraciones serán las siguientes:

Tabla 5. Planificación de los sprints de la fase de diseño e implementación.  
(Fuente: Elaboración propia)

Iteración	Horas	Fecha inicio	Fecha límite
<b>Iteración 1 (V1.1)</b> <ul style="list-style-type: none"> <li>• Instalar software necesario para el proyecto.</li> <li>• Funcionalidades de intranet para el perfil de socio.</li> </ul>	80	7 de abril	28 de abril

<sup>24</sup> <https://docs.docker.com/compose/>

<sup>25</sup> [https://es.wikipedia.org/wiki/Sistema\\_de\\_nombres\\_de\\_dominio](https://es.wikipedia.org/wiki/Sistema_de_nombres_de_dominio)

<sup>26</sup> <https://es.wikipedia.org/wiki/Nginx>

<sup>27</sup> <https://openwebinars.net/blog/que-es-un-sprint-scrum/>



<ul style="list-style-type: none"> <li>• <b>Mockups<sup>28</sup></b> para las pantallas de las funcionalidades.</li> <li>• Diseñar e implementar tablas base de datos relativas a la funcionalidad.</li> <li>• Diseño <b>endpoints<sup>29</sup> API</b>.</li> <li>• Crear estructura de la <b>API</b> en <b>Swagger</b>.</li> <li>• Implementar servicios <b>API</b> en el proyecto <b>API</b>.</li> <li>• Implementar servicios <b>API</b> en el proyecto de Angular – Intranet.</li> <li>• Implementar funcionalidades <b>frontend</b> en el proyecto de Angular – Intranet.</li> </ul>			
<b>Iteración 2 (V1.2)</b> <ul style="list-style-type: none"> <li>• Funcionalidades de intranet para el perfil de secretario.</li> <li>• <b>Mockups</b> para las pantallas de las funcionalidades.</li> <li>• Diseñar e implementar tablas base de datos relativas a la funcionalidad.</li> <li>• Diseño <b>endpoints API</b>.</li> <li>• Añadir a la estructura de la <b>API</b> en <b>Swagger</b> los <b>endpoints</b> del <b>sprint</b>.</li> <li>• Añadir implementación de los servicios <b>API</b> en el proyecto <b>API</b>.</li> <li>• Añadir implementación de los servicios <b>API</b> en el proyecto de Angular – Intranet.</li> <li>• Añadir implementación de las funcionalidades <b>frontend</b> en el proyecto de Angular – Intranet.</li> </ul>	40	29 de abril	8 de mayo
<b>Iteración 3 (V1.3)</b> <ul style="list-style-type: none"> <li>• Funcionalidades del proyecto web.</li> <li>• <b>Mockups</b> para las pantallas de las funcionalidades.</li> <li>• Diseño <b>endpoints API</b>.</li> <li>• Añadir a la estructura de la <b>API</b> en <b>Swagger</b> los <b>endpoints</b> del <b>sprint</b>.</li> <li>• Añadir implementación de los servicios <b>API</b> en el proyecto <b>API</b>.</li> </ul>	20	9 de mayo	14 de mayo

<sup>28</sup> <https://estudioka.es/que-es-un-mock-up/>

<sup>29</sup> <https://asiermarques.com/2013/conceptos-sobre-apis-rest/>

<ul style="list-style-type: none"> <li>• Añadir implementación de los servicios <i>API</i> en el proyecto de Angular – WEB.</li> <li>• Añadir implementación de las funcionalidades <i>frontend</i> en el proyecto de Angular – WEB.</li> </ul>			
<p>Iteración 4 (V1.4)</p> <ul style="list-style-type: none"> <li>• Funcionalidades de <i>BackOffice</i> para el perfil de administrador.</li> <li>• <i>Mockups</i> para las pantallas de las funcionalidades.</li> <li>• Diseñar e implementar tablas base de datos relativas a la funcionalidad.</li> <li>• Diseño <i>endpoints API</i>.</li> <li>• Añadir a la estructura de la <i>API</i> en <i>Swagger</i> los <i>endpoints</i> del <i>sprint</i>.</li> <li>• Añadir implementación de los servicios <i>API</i> en el proyecto <i>API</i>.</li> <li>• Añadir implementación de los servicios <i>API</i> en el proyecto de Angular – <i>BackOffice</i>.</li> <li>• Añadir implementación de las funcionalidades <i>frontend</i> en el proyecto de Angular – <i>BackOffice</i>.</li> </ul>	25	15 de mayo	21 de mayo



## 4. Estado del arte

En este apartado se tratará, con más detalle, un análisis de aquellas aplicaciones o proyectos que tienen una aproximación a lo que queremos desarrollar, ya que, por muy original o impensable que sea una propuesta de proyecto, hoy en día lo más probable es que exista algo similar. Por ello es más que recomendable, analizar con detalle todas aquellas plataformas, aplicaciones, webs o blogs que tengan una similitud con el proyecto propuesto.

### 4.1. Antecedentes

El principal objetivo de este proyecto, como hemos explicado antes, es dotar, a todas las asociaciones pertenecientes al ámbito festero de *les Fogueres de Sant Joan*, de una herramienta tecnológica que les permita disponer una página web y una página de gestión interna (intranet) para poder adaptarse a la tecnología actual, sin tener que depender de conocimientos técnicos o de empresas externas que les gestionen estas herramientas.

Tras la realización de una encuesta en la que han participado cerca de 60 asociaciones, lo que supone un 40% del total existente, podemos concluir que el 90% de las asociaciones de *les Fogueres* no disponen de páginas web. Además, el 98% siguen realizando todos los trámites de forma tradicional, es decir, no disponen de medios tecnológicos para realizarlo.

En esta misma encuesta el 95% afirman emplear las redes sociales más conocidas para darse a conocer y comunicar eventos, actividades, publicar fotos...etc. Así mismo, el 100% de estas asociaciones utilizan la red social Facebook, dado que proporciona lo más similar a una página web sin tener que programar o contratar un profesional que la realice. Por otro lado, de las asociaciones que tienen página web, ninguna ha solicitado servicios profesionales para realizarla, siendo los propios asociados quienes se han hecho cargo de ello.

## 4.2. Herramientas existentes

A continuación, vamos a analizar una por una las herramientas, con cierta similitud al proyecto propuesto, que existen en la actualidad.

- **FEDERACIÓ DE LES FOGUERES DE SANT JOAN:**  
(<https://www.hogueras.es/>)

Actualmente los recursos de las diferentes asociaciones para modernizarse e incluirse de lleno en la era digital son escasos. Estos recursos son cuentas y perfiles en las diferentes redes sociales, páginas web basadas en *frameworks*<sup>30</sup> fáciles de usar o sencillos blogs donde publicar fotos y vídeos de sus actividades.

Además de estos recursos, se cuenta con una página web creada por la *Federació de Les Fogueres de Sant Joan* (máximo gestor de esta fiesta) que cuenta además con una intranet que permite a las asociaciones gestionar sus trámites con la *Federació*. Ya que esta es la encargada de realizar ciertos trámites con la administración pública. Esto nos permitirá realizar un estudio más exhaustivo del posible producto final que se asemejará de una forma mucho más fiel a la herramienta deseada por estas asociaciones festeras.

La imagen mostrada a continuación es la página web pública de la *Federació de Les Fogueres de Sant Joan*, puede servir a modo de idea, cuál es el camino que seguir en el desarrollo de este proyecto. Ya que el objetivo principal, es dotar a cada hoguera (asociación) de esta herramienta, que le permita tener una mayor visibilidad sin depender de conocimientos informáticos avanzados, ni de una consultora que les realice de forma externa este tipo de aplicaciones.

---

<sup>30</sup> <https://jordisan.net/blog/2006/que-es-un-framework/>



Figura 2. Web Federació de Les Fogueres de Sant Joan.  
(Fuente: <https://www.hogueras.es/>)

En la parte interna, intranet o zona de gestión, nos encontramos con todos los documentos e información pública relevante para las distintas asociaciones.

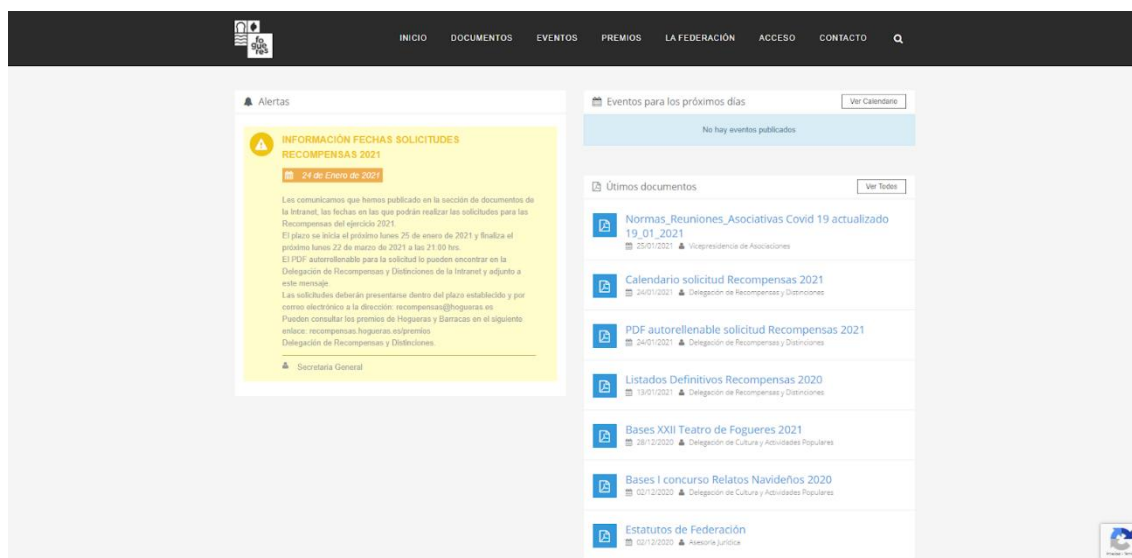


Figura 3. Intranet Federació de Les Fogueres de Sant Joan.  
(Fuente: <https://intranet.hogueras.es/>)

Además, si se accede con el usuario y contraseña de cada asociación, podemos ver un escritorio como el de la siguiente imagen, donde cada hoguera, puede gestionar sus altas, bajas, modificaciones, inscripciones en actividades realizadas por la *Federació de Les Fogueres de Sant Joan*, en definitiva, todo tipo de gestiones relacionadas con la *Federació*.

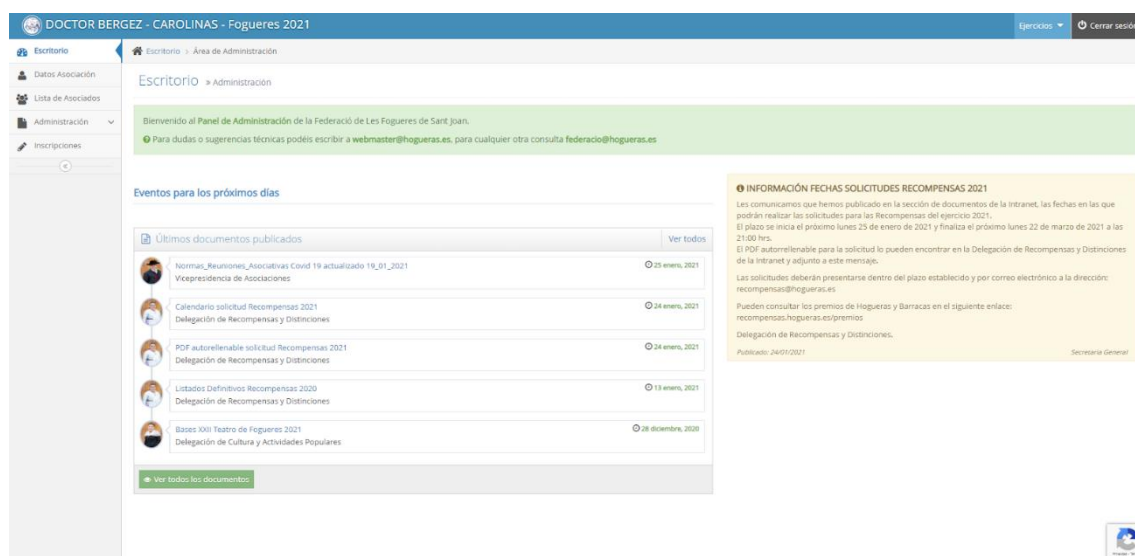


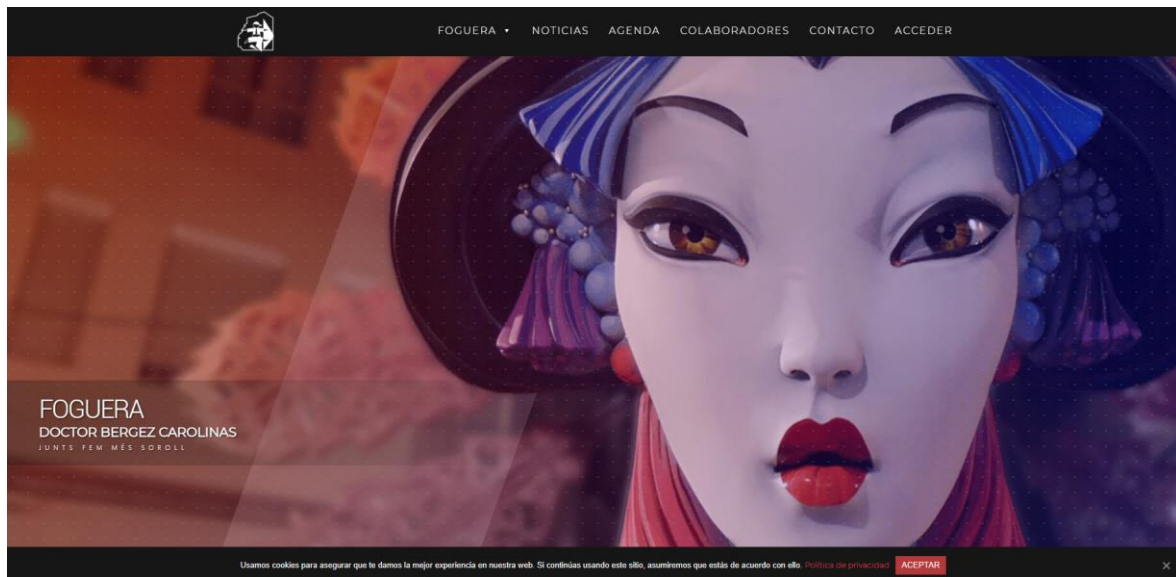
Figura 4. Intranet Federació de Les Fogueres de Sant Joan. Perfil usuario.  
(Fuente: <https://intranet.hogueras.es/escritorio/>)

En la búsqueda de una herramienta para poder gestionar una asociación festiva ha sido imposible encontrar ningún aplicativo relacionado, sin embargo, sí que existe un amplio abanico de aplicaciones enfocadas a la gestión de asociaciones sin ánimo de lucro, dado que es lo más próximo a la finalidad de este proyecto, emplearemos estas aplicaciones para realizar un estudio que nos permita mejorar el enfoque de nuestra aplicación.

Contamos, además de la página web de la *Federació de Les Fogueres de Sant Joan*, con la web de varias asociaciones y con las siguientes aplicaciones:

- **WEB FOGUERA DR. BERGEZ CAROLINAS:**  
(<https://www.hogueradoctorbergez.es/>)

La asociación (foguera) Doctor Bergez Carolinas, cuenta con casi 40 años de historia, está en proceso de publicar su página web, con la esperanza de dar un nuevo salto a las tecnologías actuales. Esta es un web escaparate, aunque disponen, también, de una intranet propia con la que realizar algunas gestiones que hasta ahora realizaban de forma manual. Esta última parte aún no está operativa.





- **CUCUNVER:**  
(<https://cucunver.com/>)

Cucunver es un programa de gestión para asociaciones sin ánimo de lucro que facilita la gestión de las tareas diarias llevadas a cabo por los directivos, o gestores, de una entidad. Está destinado al control y gestión de una asociación.

En esta imagen se puede ver el panel principal de la aplicación, que cuenta con una vista general de los aspectos más importantes de la asociación, en el ejemplo las comunicaciones de esa asociación.

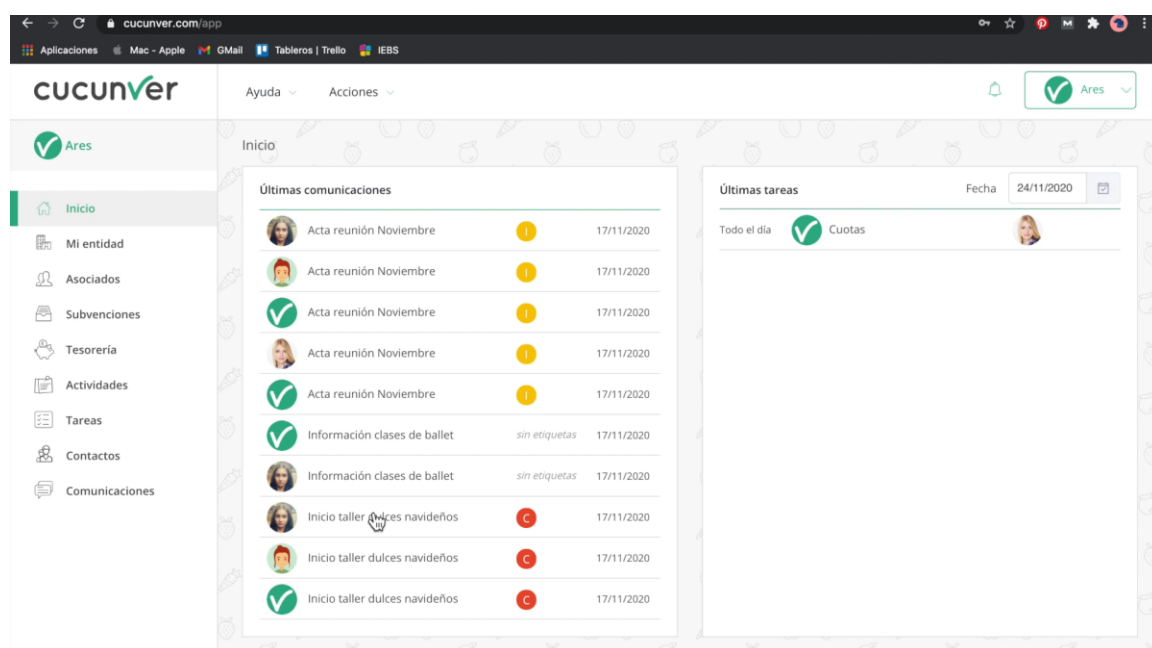


Figura 5. Panel de inicio de cucunver.  
(Fuente: <https://www.cucunver.com/>)

Además, en las dos siguientes imágenes podemos ver la forma en la que este aplicativo permite al usuario realizar la gestión económica de su asociación o las actividades que proponen y realizan.

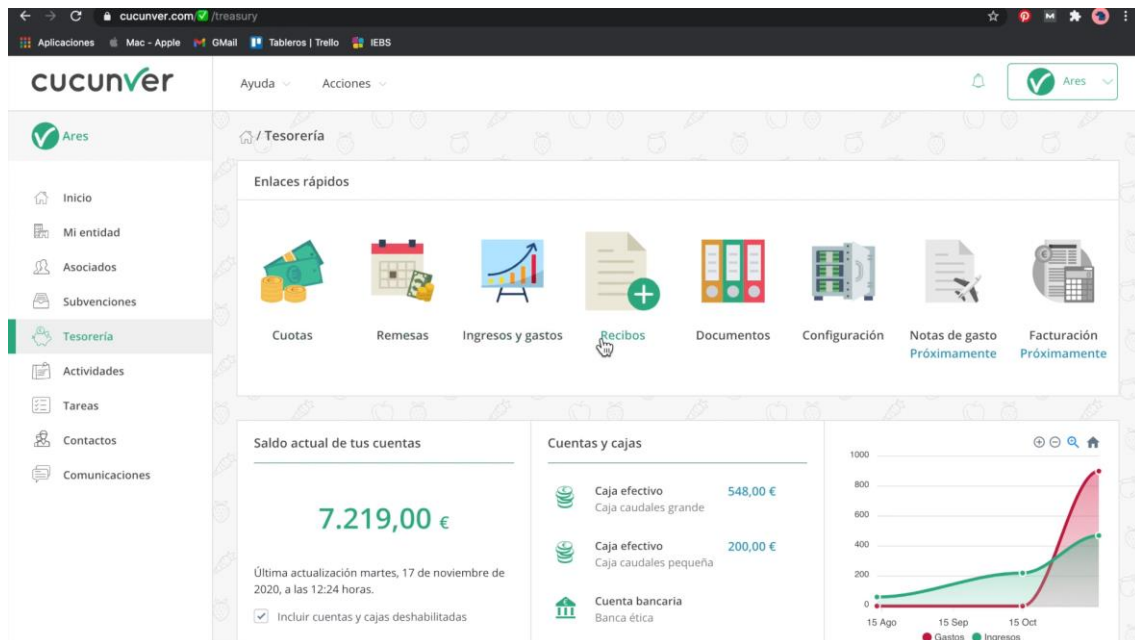


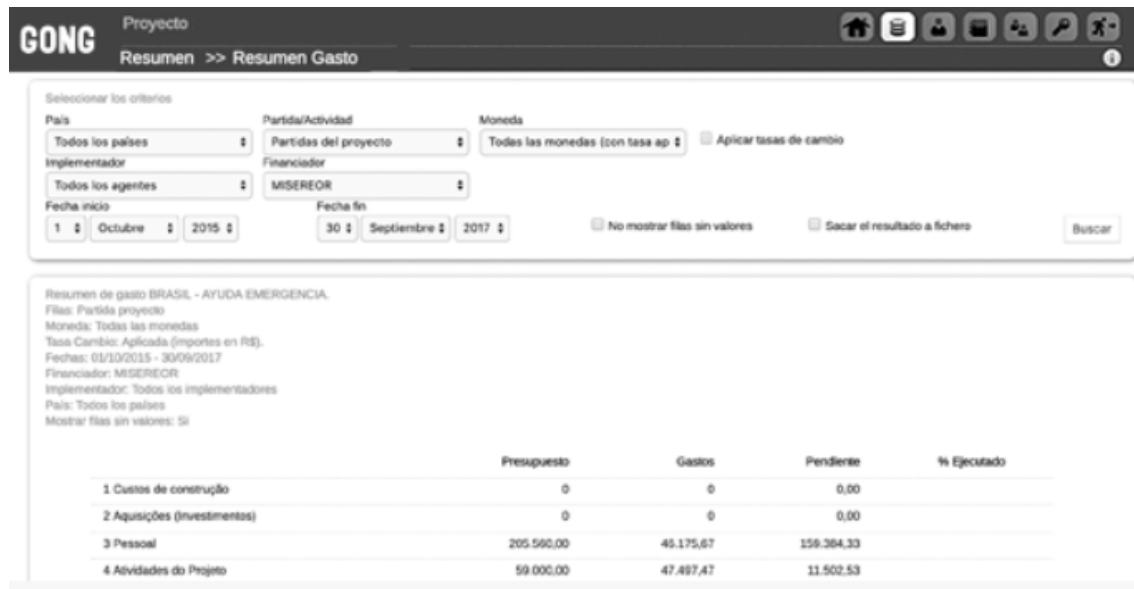
Figura 6. Gestión económica en Cucunver.  
(Fuente: <https://www.cucunver.com/>)

The screenshot shows the Cucunver Activities Management interface. The left sidebar contains a menu with options: Inicio, Mi entidad, Asociados, Subvenciones, Tesorería, Actividades (selected), Tareas, Contactos, and Comunicaciones. The main content area is titled 'Taller dulces navideños online'. It includes a 'Tipo' dropdown set to 'Taller', 'Etiquetas' (Publicada), 'Imagen' (Seleccionar archivo), 'Dirección' (online), 'Teléfono de contacto' (908 797 878), 'Fecha de inicio\*' (17/11/2020), 'Fecha fin\*' (17/12/2020), and 'Horarios' (Nuevo horario). The 'Horarios' section shows 'Sábado' with a time slot of 10:00-12:30.

Figura 7. Gestión de actividades en Cucunver.  
(Fuente: <https://www.cucunver.com/>)

- **GONG:**  
(<https://gong.es/>)

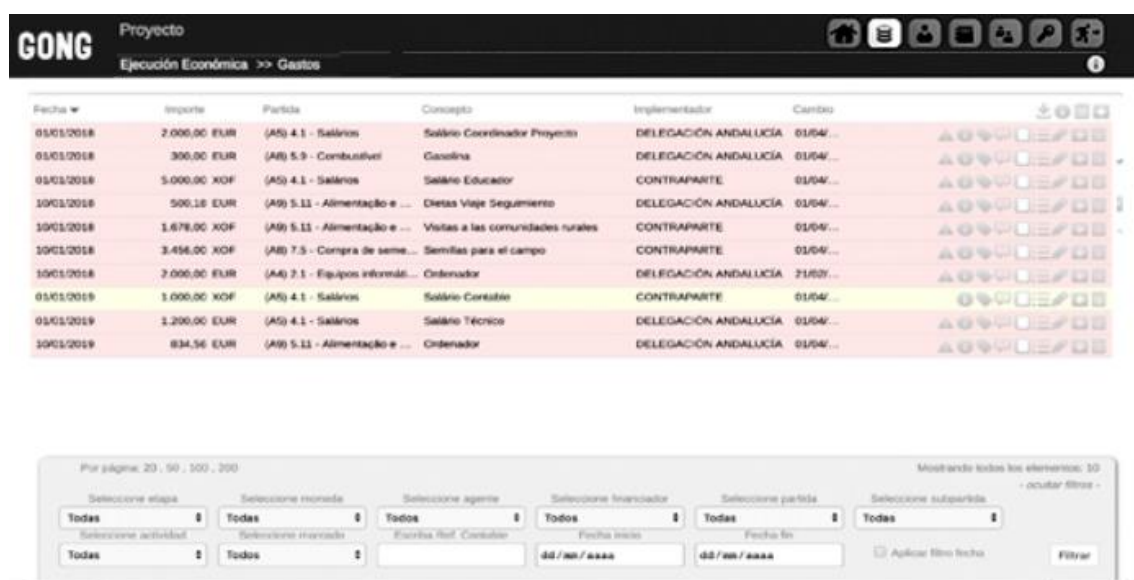
GONG es un software libre de gestión para organizaciones de desarrollo y cooperación. Dispone de cuatro grandes herramientas para la gestión eficiente de proyectos: informes, gestión técnica, gestión económica y elaboración.



**Resumen de gasto BRASIL - AYUDA EMERGENCIA.**  
 Fila: Partida proyecto  
 Moneda: Todas las monedas  
 Tasa Cambio: Aplicada (Importes en R\$).  
 Fechas: 01/10/2015 - 30/09/2017  
 Financiador: MISEREOR  
 Implementador: Todos los implementadores  
 País: Todos los países  
 Mostrar filas sin valores: Si

	Presupuesto	Gastos	Pendiente	% Ejecutado
1 Custos de construção	0	0	0,00	
2 Aquisições (investimentos)	0	0	0,00	
3 Pessoal	205.560,00	45.175,67	159.384,33	
4 Atividades do Projeto	59.000,00	47.497,47	11.502,53	

Figura 8. Gestión de informes (documentos) en GONG.  
(Fuente: <https://www.gong.es/>)



Fecha	Importe	Partida	Concepto	Implementador	Cambio
05/01/2018	2.000,00 EUR	(A5) 4.1 - Salarios	Salario Coordinador Proyecto	DELEGACIÓN ANDALUCÍA	01/04/...
01/01/2018	300,00 EUR	(A8) 5.9 - Combustivos	Gasolina	DELEGACIÓN ANDALUCÍA	01/04/...
05/01/2018	5.000,00 XOF	(A5) 4.1 - Salarios	Salario Educador	CONTRAPARTE	01/04/...
10/01/2018	500,18 EUR	(A9) 5.11 - Alimentação e ...	Dieta Viaje Seguimiento	DELEGACIÓN ANDALUCÍA	01/04/...
10/01/2018	1.678,00 XOF	(A9) 5.11 - Alimentação e ...	Viates a las comunidades rurales	CONTRAPARTE	01/04/...
10/01/2018	3.456,00 XOF	(A8) 7.5 - Compra de seme...	Semillas para el campo	CONTRAPARTE	01/04/...
10/01/2018	2.000,00 EUR	(A4) 7.1 - Equipos informát...	Ordenador	DELEGACIÓN ANDALUCÍA	21/02/...
01/01/2019	1.000,00 XOF	(A5) 4.1 - Salarios	Salario Contable	CONTRAPARTE	01/04/...
05/01/2019	1.200,00 EUR	(A5) 4.1 - Salarios	Salario Técnico	DELEGACIÓN ANDALUCÍA	01/04/...
10/01/2019	834,56 EUR	(A9) 5.11 - Alimentação e ...	Ordenador	DELEGACIÓN ANDALUCÍA	01/04/...

Por página: 20, 50, 100, 200

Mostrando todos los elementos: 10 - ocultar filtros -

Seleccione etapa: Todas  
 Seleccione moneda: Todas  
 Seleccione agente: Todas  
 Seleccione financiador: Todas  
 Seleccione partida: Todas  
 Seleccione subpartida: Todas

Seleccione actividad: Todas  
 Seleccione moneda: Todas  
 Escriba Rol/Contador: Fecha inicio: dd/mm/aaaa Fecha fin: dd/mm/aaaa

Aplicar filtro fecha Filtrar

Figura 9. Gestión económica en GONG.  
(Fuente: <https://www.gong.es/>)

- **Central Station CRM:**  
(<https://centralstationcrm.es/tour/software-crm-para-asociaciones>)

Central Station permite gestionar los contactos de una asociación, permitiendo realizar búsquedas de cualquier tipo y por cualquier tipo de clave. Esto agiliza la gestión a la hora de realizar búsquedas, pero, además, está implementado en web, por lo que los usuarios únicamente necesitan un navegador web para poder acceder en tiempo real a todos los datos de la aplicación.



Figura 10. Gestión económica en Central Station CRM.  
(Fuente: <https://centralstationcrm.es/tour/software-crm-para-asociaciones>)

Todas las tareas				Filtrar tareas
<b>Vencido</b>				
<input type="checkbox"/>	Devolver llamada a Orion	Llama...	Mi, 15. feb 11:00	EM
<b>Hoy 16.02</b>				
<input type="checkbox"/>	Llamar responsable para reunión	Llama...	17. feb Hora	EM
<b>Mañana 17.02</b>				
<input type="checkbox"/>	Club Wholesale Concretar contenido formación Estrella Moreira y Azarías Dávila		10:00 - 12:00 horas	EM
<input type="checkbox"/>	Previsión próxima semana según resultado llamada		17. feb 14:00	EM
<b>La semana que viene Martes 21.02</b>				
<input type="checkbox"/>	Reunión equipo para contenido	Prepa...	21. feb 11:00	EM
<input type="checkbox"/>	Azarías Dávila, Club Whol... Presentar el contenido a los responsables	Prepa...	21. feb Hora	EM
<b>La semana que viene Jueves 23.02</b>				
<input type="checkbox"/>	Presentación plan empresa	Impor...	23. feb 11:00	EM

Figura 11. Control de tareas en Central Station CRM.  
(Fuente: <https://centralstationcrm.es/tour/software-crm-para-asociaciones>)

Una vez resumidos los tres aplicativos con los que comparar el proyecto propuesto, paso a detallar en forma de tabla cuáles son las funcionalidades de cada aplicativo existente comparándolos con la futura aplicación desarrollada.

Tabla 6. Tabla comparativa de funcionalidades principales de cada herramienta.  
(Fuente: Elaboración propia)

Utilidad de la herramienta	GEFOC	FED	ASOCIACION	CUCUNVER	GONG	CS
Gestión de socios	X	X	X	X	X	
Control de inventario				X	X	
Subvenciones				X	X	
Control de ingresos y gastos	X			X	X	X
Organización de actividades	X	X	X	X	X	X
Planificación de tareas	X		X	X	X	X
Zona privada de usuarios	X	X	X	X	X	X
Comunicaciones	X	X	X	X	X	X
Cuotas	X			X	X	
Remesas				X	X	
Documentación	X	X	X		X	
Personalización Contenido	X				X	
Contacto	X	X	X			
Planificación de reuniones	X					
Editor de documentos	X					
Control de proveedores	X					
Control de inversores	X					

\*FED: Federació de Les Fogueres de Sant Joan, ASOCIACION: Asociaciones de hogueras y barracas, CUCUNVER: Cucunver, GONG: GONG, CS: Central Station CRM.

### **Utilidades de la herramienta.**

- Gestión de socios: permite al usuario obtener un listado de socios y en caso de tener permisos para ello, realizar tareas de mantenimiento.
- Control de inventario: permite al usuario tener un control de mantenimiento sobre diferentes ítems de la asociación.
- Subvenciones: permite al secretario de la asociación, o en su defecto, el usuario autorizado, gestionar subvenciones con organismos públicos.
- Control de ingresos y gastos: permite al tesorero de la asociación, o en su defecto, al usuario autorizado, gestionar los movimientos económicos de la asociación.
- Organización de actividades: crear actividades o eventos relativos a la asociación y gestionar las invitaciones.

- Planificación de tareas: permite a la persona autorizada crear tareas organizadas por la asociación y planificar su correcto desarrollo.
- Zona privada de usuarios: funciona a modo de intranet, es una web privada a la que sólo tienen acceso usuarios con autorización. En ella se pueden realizar una serie de acciones que requieren autenticación.
- Comunicaciones: los usuarios autenticados disponen de una sección en la que recibirán las notificaciones correspondientes.
- Cuotas: permite a los usuarios autenticados realizar gestiones relativas a la cuota que pagan en la asociación. Además, el tesorero de la asociación, o en su defecto, la persona autorizada, podrá realizar estas gestiones para todos los asociados.
- Remesas: gestión de los recibos expedidos por la asociación.
- Documentación: gestión de todos los documentos relativos a cuestiones burocráticas de la asociación. Como, por ejemplo, autorizaciones para excursiones con menores.
- Personalización de contenido: permite al usuario autorizado gestionar los contenidos que se mostrarán en la página web.
- Contacto: permite a cualquier visitante contactar con la dirección de la asociación, o en su defecto, la persona designada para ello.
- Planificación de reuniones: los usuarios pueden visualizar las reuniones que tendrán en un futuro y gestionar su asistencia a ellas.
- Editor de documentos: provee al usuario de un editor que permite elaborar documentos en la plataforma online sin necesidad de tener que subirlos desde un dispositivo.
- Control de proveedores: permite al tesorero, o en su defecto, al usuario autorizado, gestionar los proveedores de la asociación.
- Control de inversores: permite al tesorero, o en su defecto, al usuario autorizado, gestionar los anunciantes o inversores de la asociación.

Como conclusión, podemos observar que algunas funcionalidades no son pertinentes o apropiadas para nuestro objetivo, sin embargo, hemos recogido una gran parte de ellas para que sean desarrolladas en este proyecto.

Finalmente, nuestro aplicativo, tendrá más funcionalidades de las que a priori tienen sus principales competidores.





## 5. Objetivos

El objetivo principal de este proyecto es que cada asociación de las que componen la fiesta de *les Fogueres de Sant Joan*, dispongan de una página web pública donde se puedan dar a conocer y una web privada en la que puedan realizar las gestiones internas, es decir, las gestiones propias de la asociación, ya sea entre sus socios, o en relación con otra asociación.

Para lograr este objetivo, será necesario que el proyecto cumpla con 3 expectativas principales:

- Garantista: todas las asociaciones deben disponer de un espacio web accesible por todos los visitantes de internet que permitan una mayor visibilidad de esta fiesta, y que, por tanto, pueda generarles más ingresos.
- Seguro: todos los datos de las asociaciones mostrados, o existentes, dentro del proyecto solo deben ser accesibles por los socios adecuados en caso de que estos datos no deban ser públicos.
- Accesible: todos los socios deben disponer de un espacio web privado en el que puedan realizar las gestiones propias de la asociación.

Estas 3 ideas son las que forman el objetivo global del proyecto, siendo por tanto un objetivo específico, medible, ya que podemos conocer en qué grado hemos alcanzado este objetivo, y alcanzable, puesto que es un objetivo realista, siendo ambicioso pero posible de realizar. Por tanto, se cumpliría el principio SMART<sup>31</sup>, ya que además de lo descrito, tenemos un horizonte temporal fijado en las 300 horas de desarrollo para el proyecto.

---

<sup>31</sup> <https://anatrenza.com/metodo-smart-que-es-para-que-sirve/>

Los objetivos más específicos serán los siguientes:

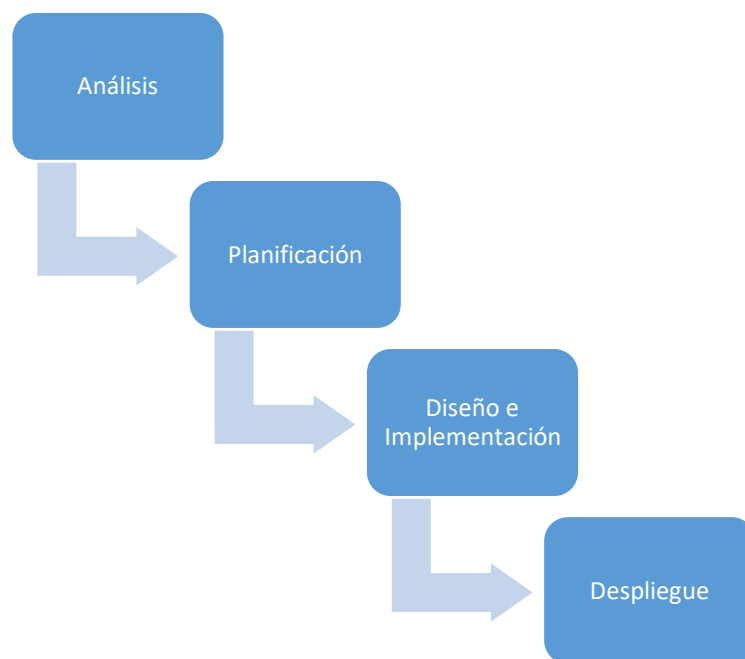
- Las asociaciones dispondrán de una página web personalizable.
- Las asociaciones dispondrán de una intranet privada.
- Las asociaciones podrán modificar y realizar mantenimientos en las diferentes secciones de la intranet.
- Los usuarios podrán realizar gestiones con su asociación gracias al aplicativo.
- Los usuarios podrán informarse de eventos y actividades relacionadas con la asociación.
- Los usuarios podrán permanecer informados de reuniones relativas a la asociación.

## 6. Metodología

Una vez hemos realizado el análisis de riesgos del proyecto, hemos definido la planificación de este, estudiado a fondo cuáles son las herramientas existentes y los antecedentes del proyecto, y, sobre todo, una vez tenemos claro, el objetivo del proyecto es hora de hablar de cómo vamos a realizarlo, de qué forma se va a trabajar.

Este paso es realmente importante ya que teniendo clara la metodología a emplear, será mucho más fácil y rápido realizar los flujos de trabajo.

En cuanto al proceso de desarrollo se ha seguido una metodología en cascada<sup>32</sup> en la que se han realizado las fases descritas anteriormente. Este sería un ejemplo de cómo se ha seguido esta metodología.



*Figura 12. Gráfico del flujo de proceso en cascada.  
(Fuente: Elaboración propia)*

<sup>32</sup> [https://es.wikipedia.org/wiki/Desarrollo\\_en\\_cascada](https://es.wikipedia.org/wiki/Desarrollo_en_cascada)

En cuanto a la fase de diseño e implementación se ha introducido la metodología lean Kanban<sup>33</sup>, para centrarnos en la entrega de las funcionalidades y gestionar la gran carga de trabajo que supone esta fase, recordemos que es un 40% del total del proyecto.

Para gestionar todo el flujo del trabajo se ha utilizado la herramienta *Azure Projects*<sup>34</sup> y en cuanto a la fase de implementación, un tablero de Trello<sup>35</sup>. Esta es una muestra del proceso de la fase de implementación.

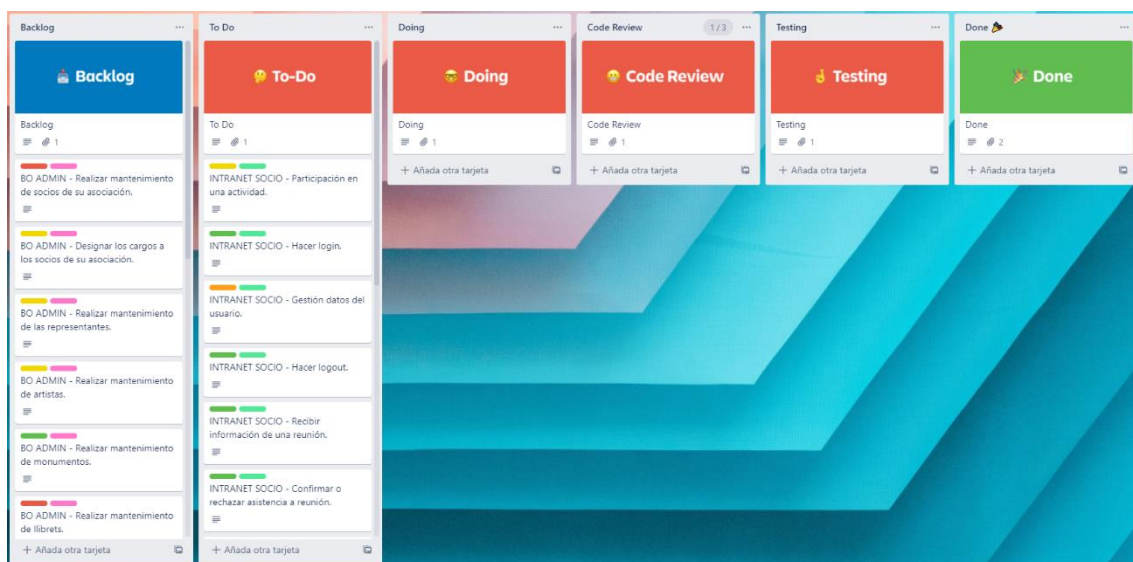


Figura 13. Planificación inicial de Trello para la versión v1.1 (iteración 1).  
(Fuente: <https://trello.com>)

<sup>33</sup> <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>

<sup>34</sup> <https://azure.microsoft.com/es-es/features/devops-projects/>

<sup>35</sup> <https://trello.com/es>

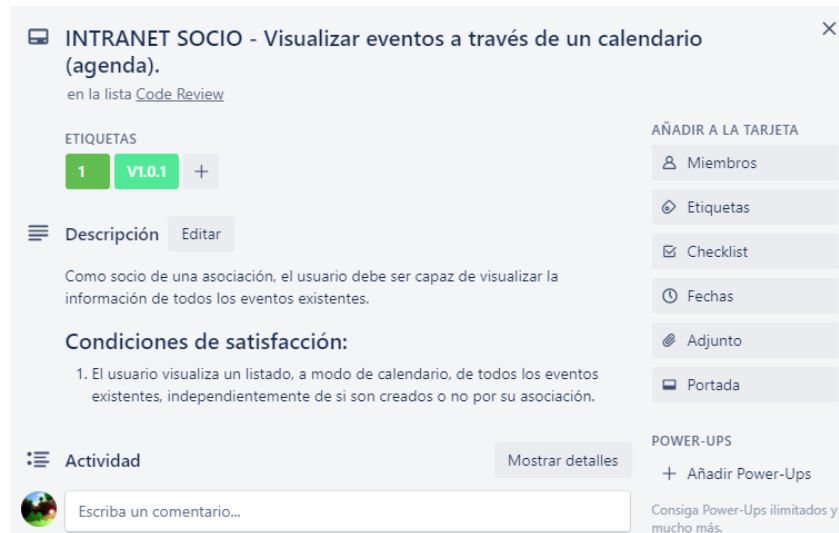


Figura 14. Detalle de la información de una tarea en Trello.  
(Fuente: <https://trello.com>)

Antes de finalizar el sprint, se realiza una retrospectiva<sup>36</sup> donde se analizan las correcciones necesarias y se elabora una lista de tareas. Se añade a Trello. Una vez finalizadas las tareas de la lista, se puede dar por concluido el sprint.

<sup>36</sup> <https://proyectosagiles.org/retrospectiva-sprint-retrospective/>

Figura 15. Revisión sprint v1.1  
(Fuente: <https://trello.com>)

## 7. Análisis y especificación

Para comenzar el planteamiento del análisis y especificación del proyecto, vamos a tener en cuenta que el proyecto será dividido en cuatro subproyectos. Además, existirán cuatro perfiles distintos, que tendrán un funcionamiento propio en cada proyecto.

Los cuatro perfiles existentes serán los siguientes:

- **Webmaster<sup>37</sup> aplicación:** Será la persona encargada de gestionar la aplicación al completo. Tendrá permisos para realizar cualquier tipo de acción.
- **Administrador asociación:** Será la persona encargada de gestionar los contenidos y el diseño de la aplicación para la asociación a la que pertenece. También podrá gestionar los permisos de esta asociación. Habrá como mínimo un administrador por asociación. Los permisos otorgados a este administrador pueden ser revocados exclusivamente por el *webmaster* de la aplicación.
- **Asociado:** Serán los asociados al menos a una asociación. Estos podrán ver datos internos de la asociación a la que pertenecen, pudiendo participar en eventos y actividades, añadir *items* a la agenda de la asociación y realizar cualquier gestión interna de la propia asociación. Este perfil será validado por el administrador de la asociación. Además, este perfil podrá tener varias categorías, socio, tesorero y secretario, en función de las que podrá realizar una determinada serie de acciones.
- **Visitante:** Serán todos aquellos navegantes que no tengan perfil de asociado o aquellos navegantes que teniendo perfil de asociado estén intentando acceder a la web de una asociación distinta a la que pertenecen.

---

<sup>37</sup> <https://es.wikipedia.org/wiki/Webmaster>



Los cuatro subproyectos serán:

- **Proyecto API:** desarrollado en node.js (express). Consiste en una *API* de servicios *Rest* que sirva todas las peticiones realizadas por el resto de los proyectos. Se encargará de conectar la base de datos con los clientes. Este proyecto no cuenta con perfiles dado que es un proyecto puramente del lado de servidor. Sin embargo, sí que cuenta con permisos y comprobaciones mediante el empleo de *tokens JWT*<sup>38</sup> para evitar acciones no deseadas.
- **Proyecto Intranet:** desarrollado con Angular 11. Será la web a la que tengan acceso los asociados que pertenezcan a esta, es decir, aunque un usuario tenga el rango de asociado, no puede acceder a la intranet que contenga la información de la asociación '3' si éste pertenece a la asociación '2'.

En esta web cada asociado podrá realizar consultas sobre la asociación a la que pertenece, realizar gestiones y colaborar con eventos realizados por la asociación.

Usuarios autorizados: *Webmaster*, administrador, asociado.

- **Proyecto BackOffice**<sup>39</sup>: desarrollado con Angular 11. Será la web de gestión en la que se podrán realizar tareas de mantenimiento, gestión y administración tanto de contenidos como de usuarios.

Usuarios autorizados: *Webmaster*, administrador.

- **Proyecto WEB:** desarrollado con Angular 11. Será la web pública de cada asociación en la que se mostrará toda la información que la asociación considere de interés para el público en general.

Usuarios autorizados: *Webmaster*, administrador, asociado, visitante.

---

<sup>38</sup> <https://jwt.io/>

<sup>39</sup> <https://www.occamagenciadigital.com/blog/backoffice-sirve-las-empresas>

## 7.1. Toma de requisitos

Una vez que tenemos claros los proyectos y los perfiles que se van a desarrollar, vamos a explicar qué requisitos debe cumplir el proyecto para poder alcanzar el objetivo propuesto.

En cuanto al proyecto de API, el requisito es que cumpla con todas las solicitudes que se le hagan, controlando la seguridad y privilegios de acceso por tipo de usuario.

En el proyecto de web, tomaremos tres requisitos:

1. Todo usuario debe poder contactar con la asociación.
2. Todo usuario debe poder estar informado sobre la actividad de la asociación.
3. Todo usuario debe poder conocer los elementos informativos de cada asociación

En el proyecto de intranet, tomaremos cuatro requisitos:

1. Todo asociado debe poder ver información relevante de la asociación, eventos, noticias, actividades, reuniones y documentos.
2. Todo asociado debe poder gestionar su asistencia a actividades, eventos o reuniones.
3. Todo asociado debe ser capaz de visionar sus datos personales y gestionarlos.
4. Todo asociado, con perfil de secretario, debe poder gestionar documentos y reuniones, realizando el mantenimiento de estos.

En el proyecto de *BackOffice*, cada entidad deberá cumplir un único requisito:

1. Todo administrador debe poder realizar el mantenimiento de las siguientes entidades:
  - Socios.
  - Cargos y representantes.
  - Artistas.
  - Monumentos.
  - Eventos y actos.
  - Contenido web de su asociación.

## 7.2. Análisis de funcionalidades

### Proyecto WEB

- Perfil: Visitante

- Funcionalidades de contacto.

Como visitante de la web, el usuario debe ser capaz de visualizar un formulario de contacto con la asociación y ver los datos de contacto de esta.

**Condiciones de satisfacción:**

1. Visualizar formulario de contacto.
2. Realizar contacto a través del formulario.
1. Visualizar datos de contacto.

- Funcionalidades de Noticias y Agenda.

Como visitante de la web, el usuario debe ser capaz de visualizar un listado con las noticias y otro con la agenda de la asociación.

**Condiciones de satisfacción:**

1. Visualizar calendario de agenda.
2. Visualizar noticias.

- Funcionalidades de contenido.

Como visitante de la web, el usuario debe ser capaz de visualizar los distintos contenidos de la web.

**Condiciones de satisfacción:**

1. Visualizar sección "Historia de la foguera".
2. Visualizar sección "Cargos".
3. Visualizar sección "Monumentos".
4. Visualizar sección "Llibrets".
5. Visualizar sección "Premios".
6. Visualizar sección "Comisión".
7. Visualizar sección "Anunciantes".
8. Visualizar sección "Galería".

## Proyecto Intranet

- Perfil: Socio

- Participación en una actividad.

Como socio de una asociación, el usuario debe ser capaz de apuntarse o desapuntarse a/de una actividad existente.

**Condiciones de satisfacción:**

2. El usuario puede apuntarse a la actividad.
3. El usuario puede desapuntarse de la actividad.

- Obtener información de todas las actividades existentes.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de todas las actividades existentes.

**Condiciones de satisfacción:**

1. El usuario visualiza un listado de actividades de su asociación.

- Obtener información de una actividad concreta.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de una actividad concreta.

**Condiciones de satisfacción:**

1. El usuario, al pinchar en una actividad, visualiza la información detallada de la actividad.

- Visualizar eventos a través de un calendario (agenda).

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de todos los eventos existentes.

**Condiciones de satisfacción:**

1. El usuario visualiza un listado, a modo de calendario, de todos los eventos existentes, independientemente de si son creados o no por su asociación.

- Obtener información detallada de un evento.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de un evento concreto.

**Condiciones de satisfacción:**

1. El usuario, al pinchar en un evento, visualiza la información detallada del evento.

- Establecer un recordatorio para un evento.

Como socio de una asociación, el usuario debe ser capaz establecer un recordatorio para ser notificado antes del inicio de un cierto evento.

**Condiciones de satisfacción:**

1. El usuario, al establecer un recordatorio para un evento, este es notificado antes de su comienzo.

- Visualizar todos los socios existentes.

Como socio de una asociación, el usuario debe ser capaz de visualizar todos los socios existentes.

**Condiciones de satisfacción:**

1. El usuario visualiza un listado de los socios de su asociación.

- Leer y descargar documentos.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de todos los documentos existentes.

**Condiciones de satisfacción:**

1. El usuario visualiza un listado de documentos de su asociación.
2. El usuario visualiza un listado de las actas de su asociación.
3. El usuario puede descargar cualquier documento.

- Confirmar o rechazar asistencia a reunión.

Como socio de una asociación, el usuario debe ser capaz de confirmar la asistencia a una reunión.

**Condiciones de satisfacción:**

1. El usuario puede confirmar o rechazar la asistencia a una reunión.

- Recibir información de una reunión.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de todas las reuniones existentes.

**Condiciones de satisfacción:**

1. El usuario visualiza la información de una reunión seleccionada.

- Iniciar sesión.

Como socio de una asociación, el usuario debe ser capaz de identificarse para poder acceder a la web.

**Condiciones de satisfacción:**

1. El usuario puede iniciar sesión.

- Cerrar sesión.

Como socio de una asociación, el usuario debe ser capaz de cerrar sesión.

**Condiciones de satisfacción:**

1. El usuario puede cerrar la sesión.

- Gestión datos del usuario.

Como socio de una asociación, el usuario debe ser capaz de visualizar todos sus datos, actualizarlos o darse de baja.

**Condiciones de satisfacción:**

1. El usuario visualiza sus datos correctamente.
2. El usuario puede actualizar sus datos.
3. El usuario puede darse de baja del sistema.

- Visualizar pagos.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de los pagos pendientes y realizados.

**Condiciones de satisfacción:**

1. El usuario visualiza los pagos realizados.
2. El usuario visualiza los pagos pendientes.

- Gestionar cuota del socio.

Como socio de una asociación, el usuario debe ser capaz de visualizar la información de la cuota que tiene asignada y solicitar un cambio.

**Condiciones de satisfacción:**

1. El usuario visualiza la información de su cuota asignada.
2. El usuario puede solicitar un cambio de cuota.

- Realizar pago especial.

Como socio de una asociación, el usuario debe ser capaz de realizar un pago especial.

**Condiciones de satisfacción:**

1. El usuario puede realizar un pago.

- Obtener certificado de pago.

Como socio de una asociación, el usuario debe ser capaz de obtener un certificado o recibo de su pago.

**Condiciones de satisfacción:**

1. El usuario obtiene un certificado o recibo de su pago.



- Perfil: Secretario

- Mantenimiento de reuniones

Como secretario de una asociación, el usuario debe ser capaz crear reuniones. Añadir contenido a éstas y gestionar la asistencia a las mismas.

**Condiciones de satisfacción:**

1. Al crear una reunión, todos los socios pueden verla.
2. Al crear una reunión, se pueden añadir puntos del orden del día.
3. Al crear una reunión se puede adjuntar el acta anterior.

- Convocar una reunión.

Como secretario de una asociación, el usuario debe ser capaz convocar una reunión creada. Cuando se genera una reunión, no tiene por qué ser convocada en el instante, sino que posteriormente será el secretario quien convoque a los socios.

**Condiciones de satisfacción:**

1. Al convocar una reunión, el secretario elige los socios que son convocados (invitados).

- Mantenimiento de documentos genéricos.

Como secretario de una asociación, el usuario debe ser capaz de realizar el mantenimiento de los documentos existentes.

**Condiciones de satisfacción:**

1. Crear un documento.
2. Modificar un documento.
3. Cambiar visibilidad de un documento.
4. Borrar un documento.

- Mantenimiento de actas de reuniones.

Como secretario de una asociación, el usuario debe ser capaz de realizar el mantenimiento de las actas existentes.

**Condiciones de satisfacción:**

1. Crear un acta.
2. Modificar un acta.
3. Cambiar visibilidad de un acta.
4. Borrar un acta.

- Perfil: Tesorero

- Mantenimiento de cuotas.

Como tesorero de una asociación, el usuario debe ser capaz de gestionar las cuotas existentes. Habrá unas cuotas genéricas y el tesorero podrá seleccionar cuales son visibles para su asociación.

**Condiciones de satisfacción:**

1. Al crear una nueva cuota, esta es elegible por todos los asociados de esta asociación.
2. Al realizar el borrado, se permitirá un borrado físico de una cuota en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.

- Asignar/cambiar cuota de familia.

Como tesorero de una asociación, el usuario debe ser capaz de gestionar las cuotas asignadas a cada familia.

**Condiciones de satisfacción:**

1. El tesorero puede asignar las cuotas que tiene cada familia.
2. El tesorero puede quitar la cuota asignada (o cambiarla) una familia concreta.

- Mantenimiento de proveedores.

Como tesorero de una asociación, el usuario debe ser capaz de gestionar los proveedores de una asociación. Los proveedores serán comunes a todas las asociaciones, por lo que se podrá dar de alta un proveedor, pero no borrarlo, sino que solicitará al *webmaster* su borrado.

**Condiciones de satisfacción:**

1. Al crear un proveedor, podrá ser utilizado por todas las asociaciones.
2. Un proveedor no podrá ser borrado.

- Añadir movimiento manualmente.

Como tesorero de una asociación, el usuario debe ser capaz de añadir un movimiento de forma manual. Este movimiento puede ser un cobro o un pago.

**Condiciones de satisfacción:**

1. Al añadir un movimiento se debe actualizar a todos los implicados.

- Aprobar cambio de cuota de un socio.

Como tesorero de una asociación, el usuario debe ser capaz de aprobar un cambio de cuota solicitado por un socio.

**Condiciones de satisfacción:**

1. Al aprobar un cambio de cuota se cambiará la cuota asignada a esa familia.

## Proyecto *BackOffice*

- Perfil: *Webmaster*

- Realizar mantenimiento de ejercicios (años)

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder realizar el mantenimiento de los ejercicios que se usarán en la aplicación. Un ejercicio hace referencia al período de tiempo que transcurre entre la fecha de inicio y la fecha de final de un año festero, en este caso desde septiembre del año 1 hasta agosto del año 2.

**Condiciones de satisfacción:**

1. Al crear un ejercicio, es visible para todos los usuarios.
2. Al crear un ejercicio se puede operar con él.
3. Al crear un ejercicio, se crean todos los códigos de asociación-ejercicio necesarios.
4. Al realizar el borrado, se permitirá un borrado físico de un ejercicio en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.
5. Al actualizar un ejercicio se actualizan todos los códigos de asociación-ejercicio relacionados con él.

- Realizar mantenimiento de asociaciones.

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder realizar el mantenimiento de las asociaciones que se usarán en la aplicación. Una asociación se entiende como un conjunto de socios gestionados por un administrador.

**Condiciones de satisfacción:**

1. Al crear una asociación, es visible para todos los usuarios.
2. Al crear una asociación se puede operar con ella.
3. Al crear una asociación, ésta debe tener al menos un administrador.
4. Al realizar el borrado, se permitirá un borrado físico de una asociación en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.

- Designar administradores de las distintas asociaciones.

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder designar el administrador de cada asociación.

**Condiciones de satisfacción:**

1. Al asignar un administrador, este administrador obtiene todos los permisos.
2. Al quitar un administrador, éste pierde todos los permisos.
3. No se permitirá quitar un administrador, si al quitarlo no queda ninguno en la asociación.

- Realizar mantenimiento de categorías de monumentos.

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder realizar el mantenimiento de las categorías que se usarán para clasificar a los monumentos. Existen ocho categorías para los monumentos adultos (Especial-1<sup>a</sup>-2<sup>a</sup>-3<sup>a</sup>-4<sup>a</sup>-5<sup>a</sup>-6<sup>a</sup>A-6<sup>a</sup>B) y siete para los infantiles (Especial-1<sup>a</sup>-2<sup>a</sup>-3<sup>a</sup>-4<sup>a</sup>-5<sup>a</sup>-6<sup>a</sup>).

**Condiciones de satisfacción:**

1. Al crear una categoría, ésta es asignable a los monumentos.
2. Al realizar el borrado, se permitirá un borrado físico de una categoría en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.

- Mantenimiento de los cargos globales.

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder realizar el mantenimiento de los cargos globales. Entiéndase por cargo global a los cargos comunes de todas las asociaciones, éstos son: Presidencia, secretaría, tesorería, vicepresidencia 1ª-4ª, belleza, dama, banderín y vocal.

**Condiciones de satisfacción:**

1. Al crear un cargo, éste es asignable a todos los usuarios.
2. Los cargos deben tener una propiedad que indique si se pueden repetir por asociación o no, por ejemplo, el cargo de presidente y el de belleza no es repetible, por el contrario, el de dama sí lo es.
3. Al realizar el borrado, se permitirá un borrado físico de un cargo en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.
4. Al aceptar un cargo, éste se guardará en la base de datos y podrá ser usado por todas las asociaciones.
5. Al rechazar un cargo, éste será borrado de la base de datos y no podrá ser usado.

- Realizar mantenimiento de los premios.

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder realizar el mantenimiento de los premios. Los premios que se otorgan cada ejercicio no varían, son siempre los mismos, por ello al asignar un premio a una asociación se hará sin modificar los datos del premio.

**Condiciones de satisfacción:**

1. Al crear un premio, éste puede ser asignado a cualquier par de ejercicio-asociación.
2. Al realizar el borrado, se permitirá un borrado físico de un premio en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.

- Aprobar solicitudes de artistas nuevos.

Como *webmaster* de la aplicación, el usuario debe ser capaz de poder aceptar la solicitud de un artista nuevo. Cuando un administrador crea un artista, éste no será aceptado en la base de datos hasta que el *webmaster* de la aplicación no lo revise.

**Condiciones de satisfacción:**

1. Al aceptar una solicitud, el artista se guarda en la base de datos y se elimina de la solicitud.
2. Al rechazar una solicitud, el artista no se guarda en la base de datos, se elimina la solicitud.



- Perfil: Administrador

- Realizar mantenimiento de socios de su asociación

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de sus socios.

**Condiciones de satisfacción:**

1. Al dar de alta a un socio, éste se crea en la base de datos, se le asigna una familia con una cuota.
2. Al realizar el borrado, se permitirá un borrado físico de un socio en caso de que no haya una relación con otras tuplas, en otro caso, se realizará un borrado lógico.
3. Si al dar de baja a un socio, éste es el último que tiene asignada una familia, ésta se borrará.
4. No se podrá dar de baja a un socio si es el último asignado a una familia y ésta tiene pagos pendientes.

- Designar los cargos a los socios de su asociación.

Como administrador de una asociación, el usuario debe ser capaz de asignar los cargos existentes a sus socios.

**Condiciones de satisfacción:**

1. Al asignar un cargo, el socio actualiza su cargo.
2. Al asignar un cargo, si éste está repetido, mostrará un error a no ser que el cargo permita ser utilizado por más de una misma persona en su asociación.

- Realizar mantenimiento de cargos existentes.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de sus cargos. Además de los globales (comunes a todas) el administrador puede crear otros cargos que se empleen en su asociación y éstos podrán ser usados por todas las asociaciones.

**Condiciones de satisfacción:**

1. Al crear un cargo, se guardará en la base de datos y estará
2. Un administrador NUNCA podrá borrar un cargo.
3. Al modificar un cargo se generará una solicitud de aprobación para el *webmaster*, un administrador nunca podrá modificar un cargo.

- Realizar mantenimiento de las representantes.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de las representantes de cada año.

**Condiciones de satisfacción:**

1. Al asignar un cargo de representante a un socio, primero se debe comprobar que el cargo cumpla las condiciones de repetición.
2. Al quitar un cargo, se elimina de la base de datos.

- Realizar mantenimiento de artistas.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de sus artistas. En caso de no existir un artista, se generará una solicitud de artista nuevo.

**Condiciones de satisfacción:**

1. Al crear un artista, se guardará en la base de datos y estará pendiente de aprobación por el *webmaster*.
2. Un administrador NUNCA podrá borrar un artista.

- Realizar mantenimiento de monumentos.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de los monumentos de cada año.

**Condiciones de satisfacción:**

1. Al crear un monumento, se asignará una categoría y un artista.

- Realizar mantenimiento de anunciantes.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de los anunciantes de cada año.

**Condiciones de satisfacción:**

1. Al crear un anunciante, éste se guardará de forma global para que pueda ser utilizado por otras asociaciones.
2. No se podrá borrar un anunciante de la base de datos, tan sólo se podrá modificar la relación entre la asociación y el anunciante.

- Realizar mantenimiento de *llibrets*.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento del *llibret* anual. El *llibret* podrá ser añadido digitalizando su contenido y realizando una subida de datos al servidor.

**Condiciones de satisfacción:**

1. Se podrá crear un *llibret*.
2. Se podrá actualizar los datos de un *llibret*.
3. Se podrá acceder a los datos de un *llibret*.
4. Se podrá borrar un *llibret*.
5. Se podrá subir un *llibret* en formato PDF.

- Realizar mantenimiento de actos.

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de actos. Los actos podrán ser eventos (organizados por una asociación) o actividades de la asociación (organizados por un socio de la asociación para el disfrute de la asociación). El administrador debe poder crear estos dos tipos de actos e invitar a los socios de su asociación.

**Condiciones de satisfacción:**

1. Al crear un evento, éste será visible para todas las asociaciones.
2. Al crear una actividad, ésta sólo será visible para los socios de la asociación organizadora.
3. Se podrá invitar a todos los socios de la asociación del administrador.
4. Se podrá borrar una actividad si es de la asociación del administrador.
5. Se podrá borrar un evento si es de la asociación del administrador.

- Realizar mantenimiento de contenidos web (CMS<sup>40</sup>).

Como administrador de una asociación, el usuario debe ser capaz de realizar el mantenimiento de los contenidos web.

**Condiciones de satisfacción:**

1. Se podrá elegir una *template* de las existentes.
2. Al seleccionar las páginas a mostrar, éstas estarán visibles en la web pública.
3. Al desactivar una página, ésta dejará de ser visible para cualquier visitante.

---

<sup>4040</sup> <https://desarrolloweb.com/home/cms>

### 7.3. Funcionalidades propuestas

Teniendo la descripción al detalle de cada una de las funcionalidades en el apartado anterior, se propone realizar un subconjunto de funcionalidades para una primera versión del proyecto. Este subconjunto se ha seleccionado teniendo en cuenta el volumen de trabajo proyectado y el tiempo estimado para realizar el TFG, así como la intención de realizar una entrega de un producto mínimo viable, pudiendo añadir más entregas con posterioridad.

#### Proyecto WEB

- Perfil: Visitante
  - Funcionalidades de contacto.
  - Funcionalidades de Noticias y Agenda.
  - Funcionalidades de contenido.

#### Proyecto Intranet

- Perfil: Socio
  - Participación en una actividad.
  - Obtener información de todas las actividades existentes.
  - Obtener información de una actividad concreta.
  - Visualizar eventos a través de un calendario (agenda).
  - Obtener información detallada de un evento.
  - Establecer un recordatorio para un evento.
  - Visualizar todos los socios existentes.
  - Leer y descargar documentos.
  - Confirmar o rechazar asistencia a reunión.
  - Recibir información de una reunión.
  - Iniciar sesión.
  - Cerrar sesión.
  - Gestión datos del usuario.

- Perfil: Secretario
  - Mantenimiento de reuniones
  - Convocar una reunión.
  - Mantenimiento de documentos genéricos.
  - Mantenimiento de actas de reuniones.

### Proyecto *BackOffice*

- Perfil: Administrador
  - Realizar mantenimiento de socios de su asociación
  - Designar los cargos a los socios de su asociación.
  - Realizar mantenimiento de las representantes.
  - Realizar mantenimiento de artistas.
  - Realizar mantenimiento de monumentos.
  - Realizar mantenimiento de llibrets.
  - Realizar mantenimiento de actos.
  - Realizar mantenimiento de contenidos web.



## 8. Diseño

En este último apartado, antes de pasar a la implementación, trataremos todo el diseño del proyecto, desde la base de datos hasta las pantallas que tendrá el aplicativo, pasando por aspectos fundamentales como la arquitectura o la *API* de servicios *Rest*.

### 8.1. Diseño de la persistencia.

Para este proyecto se creará la base de datos empleando MySQL 8.0<sup>41</sup> usando para su gestión la herramienta MySQL Workbench<sup>42</sup>. Durante el análisis y diseño de la estructura de la BD he utilizado la herramienta draw.io<sup>43</sup> que permite confeccionar los diagramas necesarios.

Para hacer frente al desafío de seguridad, que plantea el limitar la visibilidad de los datos de las asociaciones, a los socios de cada asociación, se plantea la creación de una relación entre la tabla “Ejercicios” y “Asociaciones”, lo que resulta en la piedra angular de esta base de datos, que es la tabla “Asociaciones\_Ejercicios”.

Además de esta tabla, tenemos la tabla “Representantes” que es una relación ternaria entre “Socios”, “Cargos” y “Asociaciones\_Ejercicios”.

Por último, otra relación especial, es la de “Familias” que es una relación reflexiva de la tabla “Socios”.

---

<sup>41</sup> <https://dev.mysql.com/doc/relnotes/mysql/8.0/en/>

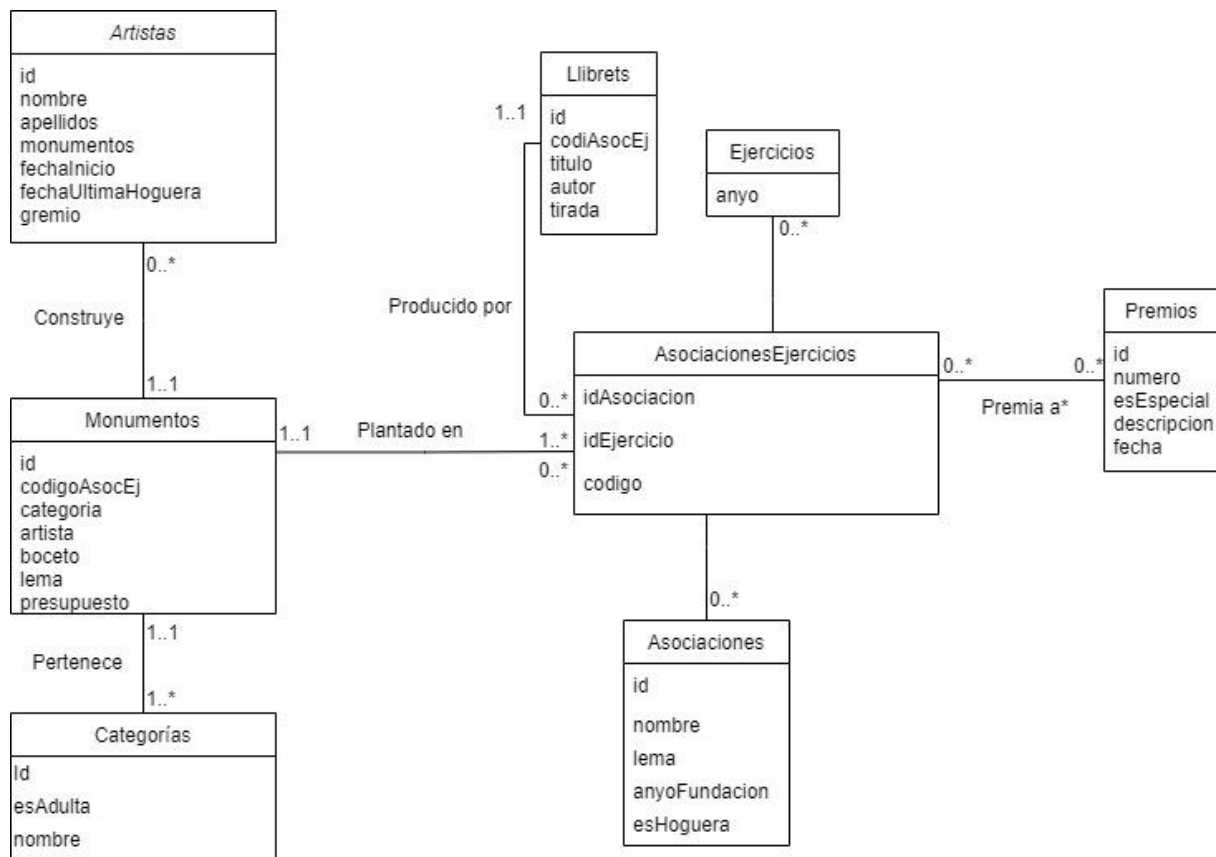
<sup>42</sup> <https://www.mysql.com/products/workbench/>

<sup>43</sup> <https://app.diagrams.net/>



Ya que la base de datos es amplia, vamos a desglosarla en tres partes (Las siguientes figuras están incluidas como anexos para un mejor visionado, son los anexos 1-4).

- Capa artística: esta primera parte tiene las tablas relacionadas con la parte artística de la asociación, categorías, artistas, monumentos...etc.



*Figura 16. Diagrama EER capa artística.  
(Fuente: Elaboración propia)*





A continuación, se detalla el diccionario de datos (DD<sup>44</sup>) empleado, no se incluyen todas las tablas dado que muchas comparten un esquema parecido. Se muestran las tablas principales, incluyendo algunos ejemplos de relaciones importantes como una relación reflexiva o una relación ternaria.

Estas tres primeras tablas conforman la piedra angular de la base de datos. Una relación muchos a muchos entre las tablas “Ejercicios” y “Asociaciones”.

*Tabla 7. Tabla Asociaciones\_Ejercicios de la BD.  
(Fuente: Elaboración propia)*

ASOCIACIONES_EJERCICIOS		
Nombre	Tipo	Descripción
idAsociacion	int	PK, FK(Asociaciones)
idEjercicio	int	PK, FK(Ejercicios)
codigo	int	UQ

*Tabla 8 Tabla Ejercicios de la BD.  
(Fuente: Elaboración propia)*

EJERCICIOS		
Nombre	Tipo	Descripción
anyo	int	PK

<sup>44</sup> [https://dapayox.wordpress.com/2011/06/24/diccionario\\_base\\_datos/](https://dapayox.wordpress.com/2011/06/24/diccionario_base_datos/)

*Tabla 9 Tabla Asociaciones de la BD.  
(Fuente: Elaboración propia)*

ASOCIACIONES		
Nombre	Tipo	Descripción
id	int	PK
nombre	varchar	UQ
lema	varchar	
anyoFundacion	int	
esHoguera	bool	

## Relación entre “Socios” y “Asociaciones”

Tabla 10 Tabla Asociaciones\_Socios de la BD.  
(Fuente: Elaboración propia)

ASOCIACIONES_SOCIOS		
Nombre	Tipo	Descripción
asociación	int	PK, FK(Asociaciones)
socio	varchar	PK, FK(Socios)
cargo	Int	FK, NN

Tabla 11 Tabla Socios de la BD.  
(Fuente: Elaboración propia)

SOCIOS		
Nombre	Tipo	Descripción
nif	varchar	PK
nombre	varchar	
apellidos	varchar	
direccion	varchar	
fechaNacimiento	date	
alergias	varchar	Array
cargo	varchar	
telefono	varchar	
email	varchar	UQ
password	varchar	
familia	int	FK(Familias)
foto	varchar	

## Relación reflexiva de la tabla “Socios”

Tabla 12. Tabla Familias de la BD.  
(Fuente: Elaboración propia)

FAMILIAS		
Nombre	Tipo	Descripción
id	int	PK
codPagador	varchar	
nombre	varchar	

Con esta tabla guardamos los administradores de cada asociación.

Tabla 13. Tabla Asociaciones\_Admin de la BD.  
(Fuente: Elaboración propia)

ASOCIACIONES_ADMIN		
Nombre	Tipo	Descripción
asociación	int	PK, FK(Asociaciones)
admin	varchar	PK, FK(Socios)

Estas dos tablas sirven para detectar qué páginas son visibles para la web de cada asociación.

Tabla 14. Tabla Templates de la BD.  
(Fuente: Elaboración propia)

TEMPLATES		
Nombre	Tipo	Descripción
Id	int	PK
descripcion	varchar	

Tabla 15. Tabla Templates\_Activas de la BD.  
(Fuente: Elaboración propia)

TEMPLATES_ACTIVAS		
Nombre	Tipo	Descripción
asociación	int	PK, FK(Asociaciones)
template	Int	PK, FK(Templates)

A la hora de guardar documentos, fotos o archivos de cualquier tipo, se almacenarán en el servidor, nunca en la base de datos para evitar daños en la base de datos como puede ser la entrada de archivos con contenido malicioso. Con estas tablas tendremos las referencias de qué directorios se utilizan para cada asociación.

Tabla 16. Tabla Configuraciones de la BD.  
(Fuente: Elaboración propia)

CONFIGURACIONES		
Nombre	Tipo	Descripción
id	int	PK
documentos	varchar	
galeria	varchar	
<u>socios</u>	varchar	

Tabla 17. Tabla Configuraciones\_Asociaciones de la BD.  
(Fuente: Elaboración propia)

CONFIGURACIONES_ASOCIACIONES		
Nombre	Tipo	Descripción
asociacion	int	PK, FK(Asociaciones)
configuracion	Int	PK, FK(Configuraciones)



## 8.2. Diseño arquitectura tecnológica.

Como hemos explicado, el proyecto se divide en cuatro subproyectos, tres de Angular y uno de node.js (express). Para hacer un enrutamiento más simple y que cada parte del aplicativo tenga un mantenimiento más sencillo, se propone esta división en cuatro subproyectos y que cada uno sea incluido en una imagen Docker.

Además, para poder tener la persistencia del apartado anterior, se incluye un quinto contenedor con una imagen de mysql 8.0.

Llegados a este punto tendríamos 5 contenedores *Docker*, para poder hacer el enrutamiento correcto necesitaríamos agregar un servicio de reverse proxy como NGINX.

Por último, sería conveniente agregar una certificación SSL al proyecto, de esta forma la navegación sería segura. Así pues, añadimos también *Let's encrypt*<sup>45</sup>.

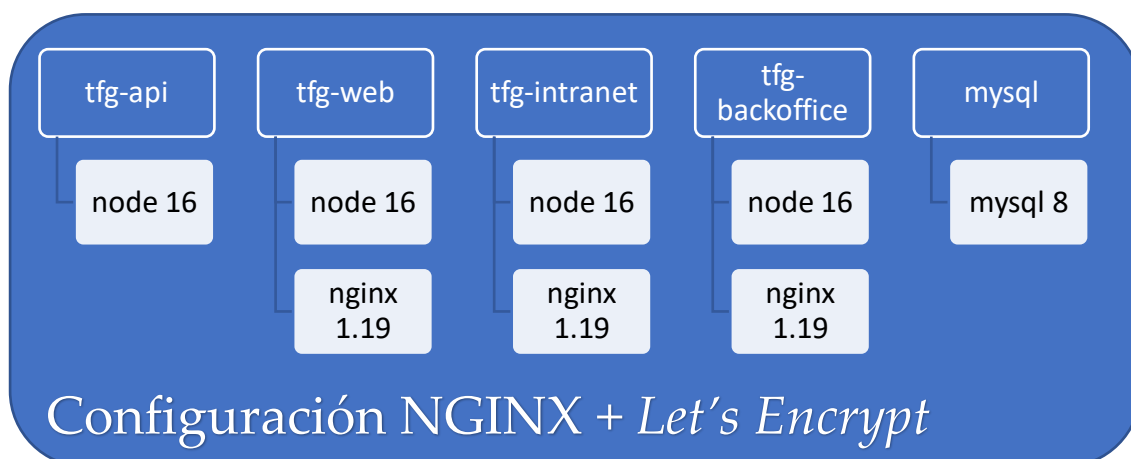


Figura 20. Gráfico estructura contenedores Docker con mysql y configuración NGINX y Let's Encrypt en el servidor.  
(Fuente: Elaboración propia)

En el apartado de despliegue veremos que, finalmente, se queda todo configurado en un archivo *docker-compose* para facilitar la portabilidad y mantenibilidad del proyecto.

<sup>45</sup> <https://letsencrypt.org/es/>

### 8.3. Diseño *API Rest*

Para realizar el diseño de la API he decidido usar la herramienta [swagger.io](https://swagger.io/)<sup>46</sup> que permite generar la documentación de la API y, además, tiene una utilidad llamada *swagger-codegen*<sup>47</sup> que crea la estructura básica para la API y genera los servicios para que el cliente consuma la API.

Tiene un editor de texto con el que puedes generar en la web el archivo en formato “.yaml”, además, si paras de escribir compila el archivo y resalta los posibles errores, por lo que se pueden corregir casi al instante en caso de que haya.

Se puede consultar la documentación de la API<sup>48</sup>, ya que la dispone en un formato muy práctico, con ejemplos de llamadas y respuestas.

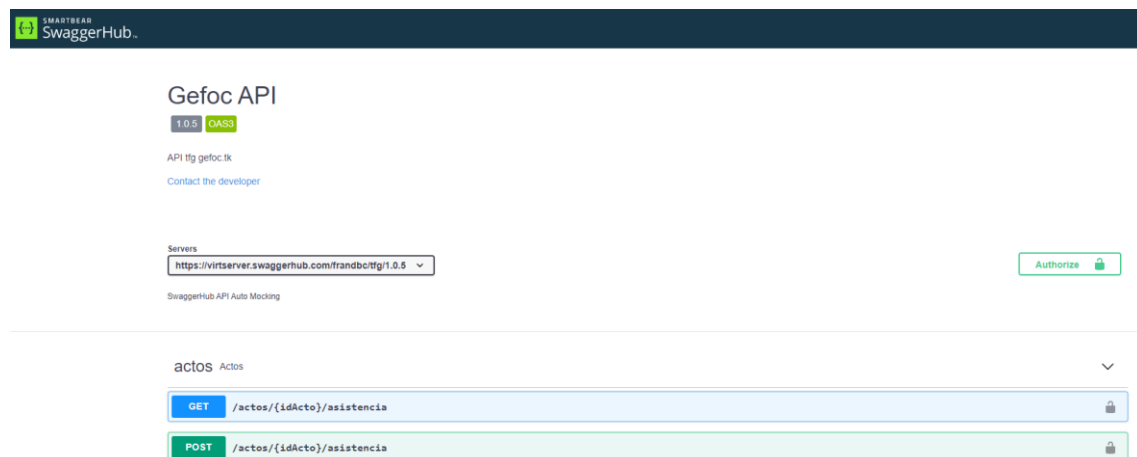


Figura 21. Documentación API Gefoc.  
(Fuente: <https://app.swaggerhub.com/apis-docs/frandbc/tfg/1.0.5>)

<sup>46</sup> <https://swagger.io/>

<sup>47</sup> <https://swagger.io/tools/swagger-codegen/>

<sup>48</sup> <https://app.swaggerhub.com/apis-docs/frandbc/tfg/1.0.5>

Permite agrupar los *endpoints* con tags, así pues, en este ejemplo se muestran todos los *endpoints* creados para el tag “artistas”.

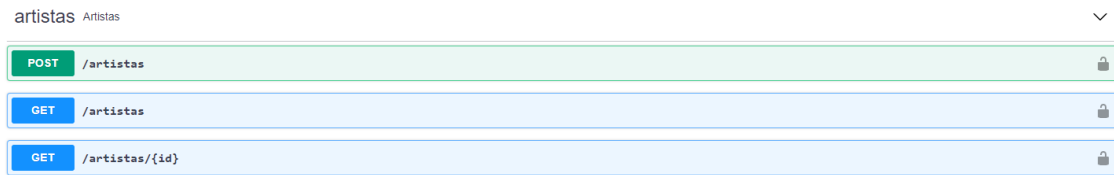


Figura 22. Endpoints tag /artistas API Gefoc.  
(Fuente: <https://app.swaggerhub.com/apis-docs/frandbc/tfg/1.0.5#/artistas>)

Al clicar en un *endpoint* podemos ver toda la información sobre el mismo, los parámetros de entrada (pueden ser *request params* o *request body*<sup>49</sup>) y las response que da el *endpoint*. En este ejemplo, podemos ver la información del *endpoint* “POST /actos”.

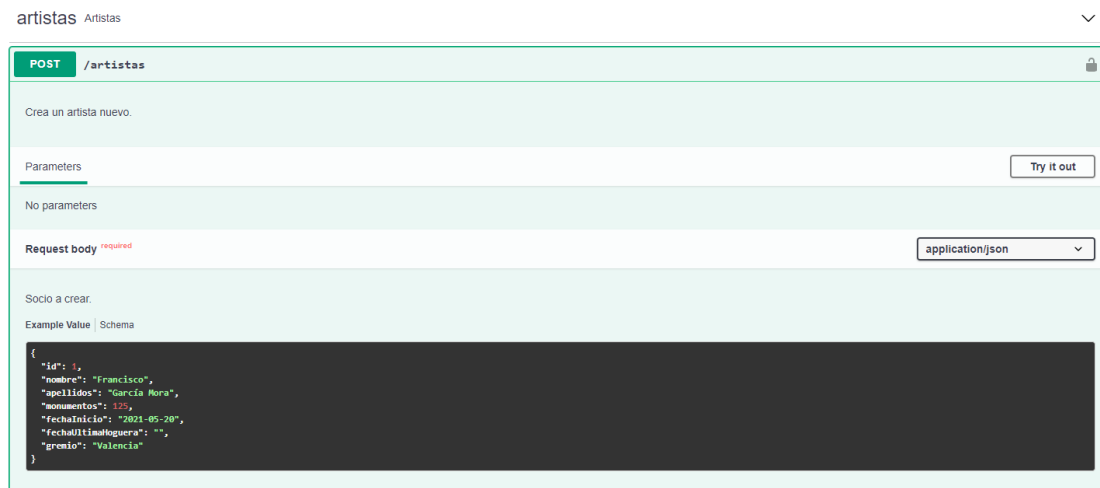


Figura 23. Request endpoint POST /artistas API Gefoc.  
(Fuente: <https://app.swaggerhub.com/apis-docs/frandbc/tfg/1.0.5#/artistas/createArtista>)

<sup>49</sup> [https://www.tutorialspoint.com/http/http\\_parameters.htm](https://www.tutorialspoint.com/http/http_parameters.htm)

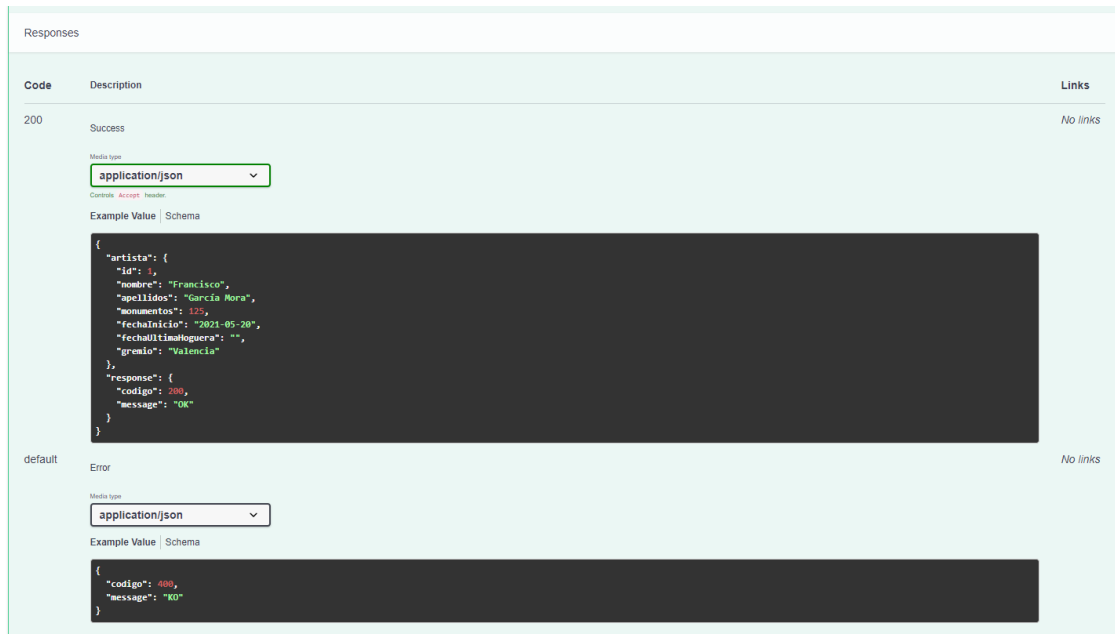


Figura 24 Figura 25. Response endpoint POST /artistas API Gefoc.  
(Fuente: <https://app.swaggerhub.com/apis-docs/frandbc/tfg/1.0.5#/artistas/createArtista>)

También permite crear estructuras para ser devueltas en las respuestas o también para ser empleadas en las peticiones. Esto facilita mucho crear la API ya que permite reutilizar una gran cantidad de líneas de código. En este ejemplo, podemos ver una estructura (*schema*) de “Acto”.

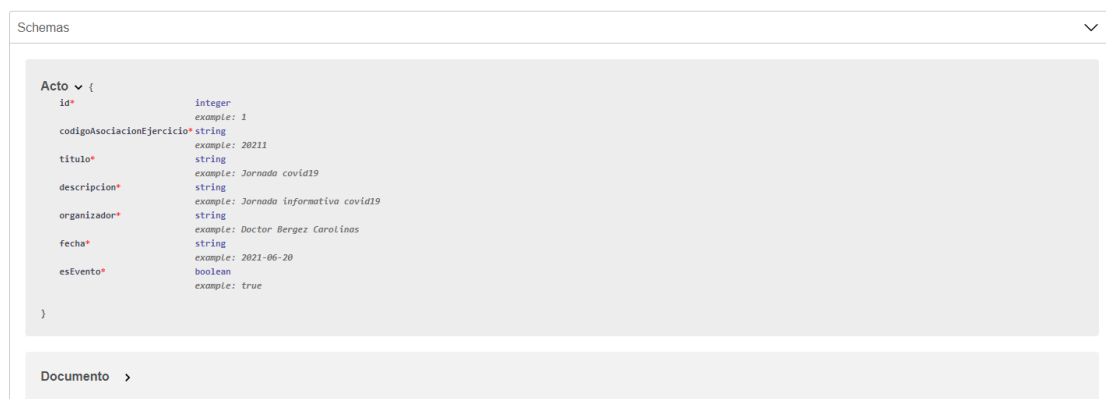


Figura 26. Schema Acto API Gefoc.  
(Fuente: <https://app.swaggerhub.com/apis-docs/frandbc/tfg/1.0.5#/Acto>)

**Endpoints API.**

- *Endpoints* para actos.

Tabla 18. Endpoints tag "actos"  
(Fuente: Elaboración propia)

Endpoint	Método	Observaciones
/actos/{idActo}/asistencia	GET/POST	
/actos	GET/GET{id}/POST/PUT/DELETE	
/actos/eventos	GET	
/actos/actividades	GET	
/recordatoriosActos	GET/POST	

- *Endpoints* para documentos.

Tabla 19. Endpoints tag "documentos"  
(Fuente: Elaboración propia)

Endpoint	Método	Observaciones
/documentos/{id}	GET/GET{id}/PUT/DELETE	
/archivos/upload	POST	Sube archivos al servidor
/archivos/download	POST	Descarga archivos del servidor
/documentos	POST	
/documentos/actas	GET	
/documentos/documentos	GET	

- *Endpoints* para páginas.

Tabla 20. Endpoints tag "paginas"  
(Fuente: Elaboración propia)

Endpoint	Método	Observaciones
/paginas	GET	
/asociaciones/{id}/paginas	GET/DELETE/PUT	
/asociaciones/{id}/paginas/{pagina}	GET/DELETE	Información de la página solicitada

- *Endpoints* para reuniones.

Tabla 21. Endpoints tag "reuniones"  
(Fuente: Elaboración propia)

Endpoint	Método	Observaciones
/reuniones	GET/POST	
/reuniones/{id}	GET/DELETE/PUT	
/reuniones/{id}/asistencia	GET/POST	
/reuniones/{id}/adjuntarActa	PUT	Edita la reunión seleccionada añadiendo el acta
/reuniones/{id}/convocar	POST	Convoca una reunión ya creada

- *Endpoints* para socios.

Tabla 22. Endpoints tag "socios"  
(Fuente: Elaboración propia)

Endpoint	Método	Observaciones
/socios	POST	
/socios/{id}	GET/DELETE/PUT	
/socios/{id}/cargo	GET	
/socios/perfil/asociacionActiva	GET/POST	
/socio/perfil	GET	
/login	POST	
/asociación/{id}/socios	GET	Obtiene todos los socios de una asociación

- *Endpoints* para *templates*.

Tabla 23. Endpoints tag "templates"  
(Fuente: Elaboración propia)

Endpoint	Método	Observaciones
/templates	GET	
/asociaciones/{id}/template	GET/DELETE/PUT/POST	
/asociaciones/{id}/paginas	POST	

## 8.4. Diseño de interfaces

En cuanto al diseño de las interfaces y pantallas he realizado un diseño básico con la herramienta *Figma*<sup>50</sup>. Posteriormente con *Illustrator*<sup>51</sup> se ha realizado un diseño más profesional. Además, aprovechando las funcionalidades de *Figma*, se realizó un prototipo con esta herramienta para hacer más intuitivo el flujo del aplicativo durante el proceso de implementación.

A continuación, se muestran los mockups de los tres proyectos, no están todas las pantallas dado que son una gran cantidad, por ello sólo se mostrarán las más importantes. Todos los datos que aparecen son ficticios.

---

<sup>50</sup><sup>50</sup> <https://www.figma.com/>

<sup>51</sup> <https://www.adobe.com/es/products/illustrator.html>



- **Mockups BackOffice – webmaster**

En primer lugar, tenemos dos mockups del proyecto *BackOffice* para el perfil de *webmaster*. En este caso sólo se muestra el inicio de sesión y la página de inicio, dado que todas las páginas serán idénticas.

Figura 27. Mockups – BackOffice Master – Inicio de sesión.  
(Fuente: Elaboración propia)

Ejercicio 2021 ▾					
Dato 1	Dato 1	Dato 1	Dato 1	X	↺
Dato 1	Dato 1	Dato 1	Dato 1	X	↺
Dato 1	Dato 1	Dato 1	Dato 1	X	↺
Dato 1	Dato 1	Dato 1	Dato 1	X	↺
Dato 1	Dato 1	Dato 1	Dato 1	X	↺

Nueva asociación

Figura 28. Mockups – BackOffice Master – Página de inicio.  
(Fuente: Elaboración propia)

## • **Mockups BackOffice – Administrador**

Continuando con el BackOffice, tenemos dos mockups más, el primero sería la vista inicial para el perfil de administrador, siendo en este caso, un CRUD con los diálogos modales o *popups* que se generan y una vista especial que es el CMS donde el administrador puede gestionar el contenido de la web pública.



Figura 29. Mockups – BackOffice Admin – Home.  
(Fuente: Elaboración propia)



Figura 30. Mockups – BackOffice Admin – Ejemplos de modales diversos.  
(Fuente: Elaboración propia)

**Fran García Mora**

Socios  
Cargos  
Representantes  
Premios  
Artistas  
Monumentos  
Anunciantes  
Llibrets  
Actividades  
Eventos  
**CMS**

Páginas **Contenidos**

Página 1 - Título

Contenido

Templates

Figura 31. Mockups – BackOffice Admin – CMS.  
(Fuente: Elaboración propia)

**Fran García Mora**

Socios  
Cargos  
Representantes  
Premios  
Artistas  
Monumentos  
Anunciantes  
Llibrets  
Actividades  
Eventos  
**CMS**

Páginas **Contenidos**

Página 1	<input checked="" type="checkbox"/>
Página 2	<input type="checkbox"/>
Página 3	<input checked="" type="checkbox"/>
Página 4	<input checked="" type="checkbox"/>
Página 5	<input type="checkbox"/>

Figura 32. Mockups – BackOffice Admin – CMS.  
(Fuente: Elaboración propia)

## • **Mockups Intranet**

El proyecto más importante en cuanto a lógica y programación, pero más sencillo de diseñar, es el de la Intranet. Para hacer una muestra del diseño sin extenderme demasiado vamos a mostrar seis mockups. El primero será la pantalla de inicio.

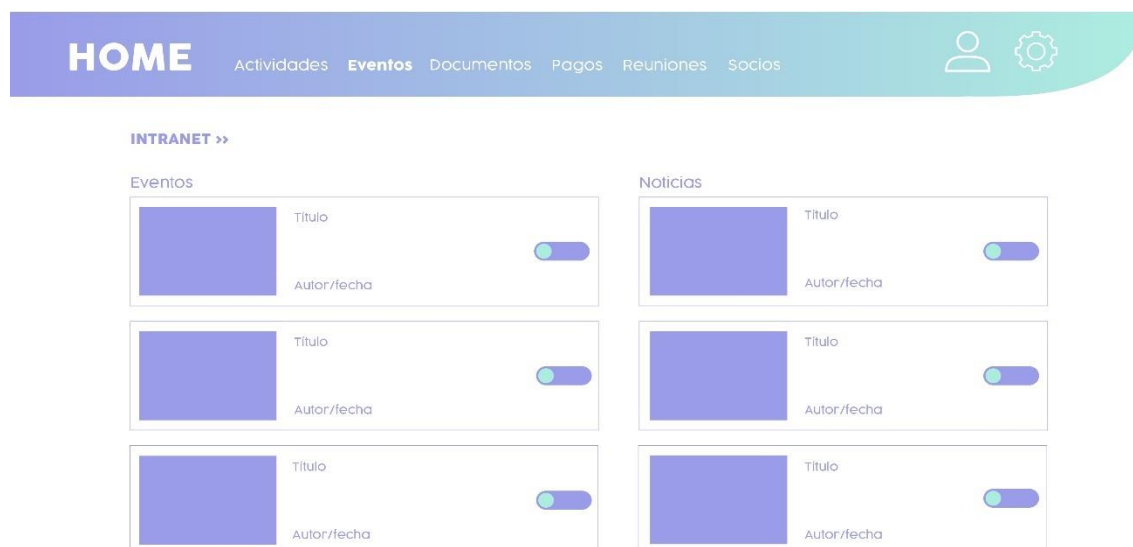


Figura 33. Mockups – Intranet Socio – Página de inicio.  
(Fuente: Elaboración propia)

Además, todos los usuarios podrán visualizar datos con diálogos modales como estos que se muestran a continuación.



Figura 34. Mockups – Intranet Socio – Modales.  
(Fuente: Elaboración propia)

Todos los usuarios podrán acceder a su perfil y un calendario que incluya eventos y actividades. Incluidos en los siguientes dos mockups.

**HOME**   Actividades   Eventos   Documentos   Pagos   Reuniones   Socios

**INTRANET >> PERFIL**

**ASOCIACIONES**  
Cementerio norte  
Carolinas del medio  
Plazos sanitarios

Nombre  
Apellidos  
DNI  
Dirección  
Fecha de nacimiento  
Alergias  
Cargo  
Teléfono  
Email  
Contraseña

Actualizar datos

Figura 35. Mockups – Intranet Socio – Socios.  
(Fuente: Elaboración propia)

**HOME**   Actividades   **Eventos**   Documentos   Pagos   Reuniones   Socios

**INTRANET >> EVENTOS**

JUNIO 2021						
<						>
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
x	x	x	x	x	x	x
x	Reunión 20:30 - 21:30	x	x	x	x	x
x	x	x	Actividad 17:00 - 20:00	x	x	x
x	x	x	x	x	x	x
x	x	x	x	x	x	x

Figura 36. Mockups – Intranet Socio – Eventos.  
(Fuente: Elaboración propia)

Ahora, centrándonos en el perfil del usuario con el cargo de secretario, podemos ver las pantallas correspondientes a la funcionalidad de convocar una reunión o las de crear documentos genéricos y actas.

**INTRANET >> REUNIONES >> CONVOCAR**

Convocatoria:

Lugar:

**Acta reunión anterior**

**Puntos de la reunión**

1	Aprobación del acta anterior		21:34	21:35
2	Informe del presidente		21:35	21:43

**Asistentes**

Asociado 1	Asociado 2	Asociado 3	Asociado 4	Asociado 5	Asociado 6
Asociado 7	Asociado 8	Asociado 9	Asociado 10	Asociado 11	Asociado 12

*Figura 37. Mockups – Intranet secretario – Convocar reunión.  
(Fuente: Elaboración propia)*

**INTRANET >> DOCUMENTOS**

**Actas Genéricos**

Título

Autor

Fecha

**Previsualizar**

*Figura 38. Mockups – Intranet secretario – Generar documento genérico.  
(Fuente: Elaboración propia)*

HOME

Actividades

Eventos

Documentos

Pagos

Reuniones

Socios

INTRANET >> DOCUMENTOS

Actas

Genéricos

Convocatoria:

Reunión inicio

Hora inicio:

17:30

Hora fin:

17:30

Lugar:

xxxxxx

Fecha:

01/06/21

Puntos de la reunión

1	Aprobación del acta anterior		21:34	21:35	
2	Informe del presidente		21:35	21:43	

Votaciones

1	Propuesta 1		15	2	
2	Propuesta 2		17	0	

Asistentes

Asociado 1		Asociado 2		Asociado 3		Asociado 4		Asociado 5		Asociado 6		
Asociado 7		Asociado 8		Asociado 9		Asociado 10		Asociado 11		Asociado 12		

Figura 39. Mockups – Intranet secretario – Generar acta.  
(Fuente: Elaboración propia)

Por último, quedan diseñadas las pantallas de las futuras funcionalidades que se podrán ver en próximas versiones. Una muestra de ello es la gestión económica, desde la vista del perfil de usuario y la vista del perfil de tesorero.

HOME

Actividades

Eventos

Documentos

Pagos

Reuniones

Socios

INTRANET >> PAGOS

Pagos pendientes

Actualizar datos

ID	Descripción del pago	Cantidad	Fecha de vencimiento
1	Cuota mes de Junio	70	16/06/2021

Pagos realizados

ID	Descripción del pago	Cantidad	Fecha de vencimiento
2	Cuota mes de Mayo	70	16/06/2021

Cuota actual - Pack familiar

Cambiar cuota

Concepto del pago	Cantidad del pago	Deuda acumulada	Periodicidad del pago
Lotería semanal	3.60€	8€	Semanal
Lotería navidad	360€	162€	Anual
Anuncios libret	360€	50€	Anual
Cuota hoguera	50€	30€	Mensual
Total	1200€	250€	Anual

Figura 40. Mockups – Intranet Socio – Pagos.  
(Fuente: Elaboración propia)

HOME

Actividades

Eventos

Documentos

Pagos

Reuniones

Socios

INTRANET >> PAGOS >> TESORERÍA

Proveedores

Movimientos

Cobros

Pagos

Cuotas

Cobros pendientes

Añadir cobro pendiente

ID	Descripción del cobro	Cantidad del cobro	Fecha	Cobrador	ID Transacción
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245

Cobros pendientes

Añadir cobro realizado

ID	Descripción del cobro	Cantidad del cobro	Fecha	Cobrador	ID Transacción
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245

Figura 41. Mockups – Intranet Tesorero – Cobros.  
(Fuente: Elaboración propia)

HOME

Actividades

Eventos

Documentos

Pagos

Reuniones

Socios

INTRANET >> PAGOS >> TESORERÍA

Proveedores

Movimientos

Cobros

Pagos

Cuotas

Movimientos realizados

ID	Descripción del cobro	Cantidad del cobro	Fecha	Cobrador	ID Transacción
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245
0001	Provisión coca cola	700€	20/06/21	Amstel	0000000001245

Saldo cuenta bancaria:

7250 €

Cobros pendientes:

1232 €

Saldo a final de mes:

5350 €

Pagos pendientes:

3140 €

Figura 42. Mockups – Intranet Tesorero – Movimientos.  
(Fuente: Elaboración propia)



HOME

Actividades

Eventos

Documentos

Pagos

Reuniones

Socios

INTRANET >> PAGOS >> TESORERÍA

Proveedores

Movimientos

Cobros

Pagos

Cuotas

cuota familiar

Crear nueva cuota

Concepto del pago	Cantidad del pago	Periodicidad del pago
Lotería semanal	3.60€	Semanal
Lotería navidad	360€	Anual
Anuncios libret	360€	Anual
Cuota hoguera	50€	Mensual
Total	1200€	Anual

☐
☒
☐
☐

cuentas de asociados

Asociado	Familia	Cuenta asociada
María	1	Cuenta familiar
xxxxx	2	Cuenta individual
xxxxx	3	xxxxx

Figura 43. Mockups – Intranet Tesorero – Cuotas.  
(Fuente: Elaboración propia)

- **Mockups WEB**

Finalizando la presentación de los mockups, queda la página escaparate, dado que el diseño de este proyecto puede variar entre cada asociación, dejamos una muestra de una plantilla de ejemplo, con la página inicial.



Figura 44. Mockups – WEB – Página inicial.  
(Fuente: Elaboración propia)



## 9. Implementación

Para explicar la implementación del proyecto vamos a centrarnos en un solo subproyecto, escogemos el de la intranet dado que es el que tiene las funcionalidades con más valor del proyecto.

### 9.1. Control de versiones

El control del versionado se ha realizado con GIT, a través de la herramienta de GIT que aporta *VsCode*<sup>52</sup>. Se propone realizar cuatro iteraciones, en las que al final de cada iteración se realizará una *release*. Para la gestión de las tareas de cada iteración he utilizado un tablero Kanban.

Las *releases* tendrán como objetivo publicar cuatro versiones finales, v1.1, v1.2, v1.3 y v1.4 que será la versión marcada como objetivo para el proyecto.

A continuación, se muestran las tareas que se desarrollarán para la versión v1.1

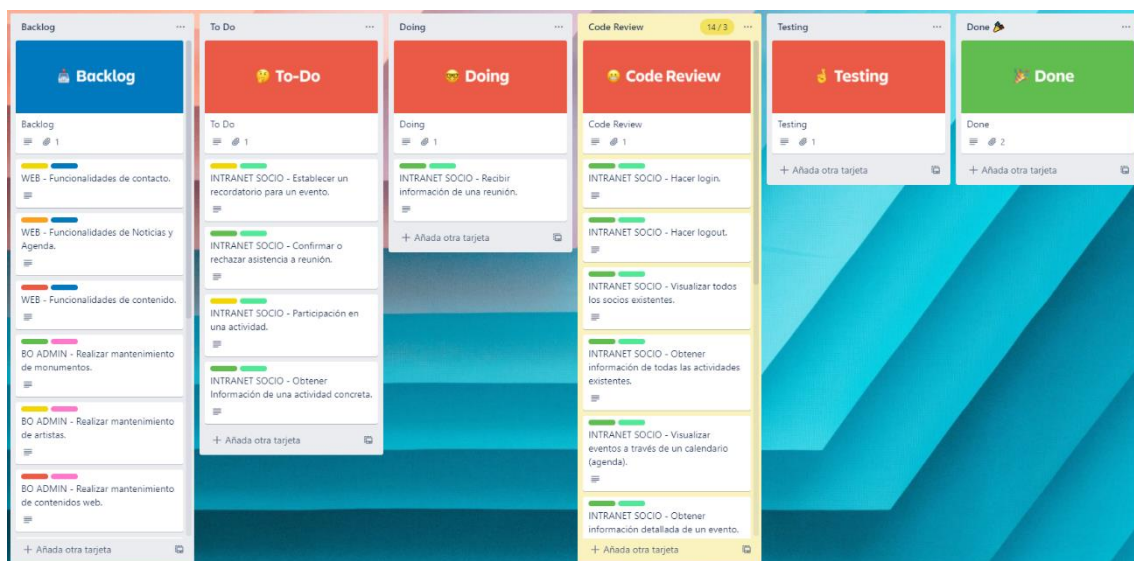


Figura 45. Tablero trello v1.1  
(Fuente: <https://trello.com>)

<sup>52</sup> <https://code.visualstudio.com/>

Antes de comenzar a programar, descargamos los cambios, si los hubiera del repositorio y se crea una nueva rama para la *release*. Esta rama tendrá que partir de la rama *develop*.

Primero se desarrolla la funcionalidad en la API, y una vez completada, en el proyecto de intranet se desarrolla la funcionalidad deseada. Al acabar de desarrollarla se comprueba con `'ng build --prod'` que no haya ningún error en el proyecto y se ejecuta.

Cuando comprobemos que la funcionalidad cumple las condiciones de satisfacción procedemos a crear una imagen de Docker con el proyecto. Para hacerlo se debe ejecutar el siguiente comando `'docker build -t tfg-intranet .'`

Una vez tengamos la imagen de *Docker* creada, le asignamos un tag y la subimos a *DockerHUB*<sup>53</sup> para posteriormente poder desplegarla en el servidor. Sólo tendríamos que ejecutar los comandos:

- `'docker tag tfg-intranet nomobreUsuarioDocker/tfg-intranet:versión'`
- `'docker push nomobreUsuarioDocker/tfg-intranet:versión'`

Después de este paso ya estaría la funcionalidad hecha y lista para desplegar, por lo que pasaríamos a la siguiente funcionalidad.

Cuando estén hechas todas las funcionalidades, sólo quedaría fusionar la rama de la versión con *develop* y realizar la subida al repositorio remoto, manteniendo así nuestro control de versiones.

---

<sup>53</sup> <https://hub.docker.com/>

## 9.2. Despliegue de aplicativo

Para realizar el despliegue del proyecto he seguido los siguientes pasos:

1. Crear imágenes *Docker* para cada proyecto.
2. Publicar estas imágenes en *DockerHUB*.
3. Desplegar las imágenes desde *DockerHUB* en el servidor.
4. Configurar el servicio NGINX para poder acceder a cada proyecto.

Una vez finalizados estos pasos ya está desplegado correctamente el aplicativo en el servidor, sin embargo, he decidido avanzar un paso más y virtualizar por completo el proyecto creando un único archivo *docker-compose* que permita desplegar toda la aplicación con un solo comando. Para más detalles sobre el despliegue, en el “*Anexo 5 - Configuración y despliegue del aplicativo*”, se describe todo el proceso.

Cuando está todo correctamente configurado sólo se necesita ejecutar el comando ``docker-compose up -d`` y en unos segundos estarán todos los servicios en marcha. Como recordatorio final, los subdominios empleados tienen que estar registrados como DNS, sino los dispositivos, que intenten acceder a la dirección del aplicativo, no reconocerán esta dirección web. También es aconsejable revisar las reglas de firewall del servidor, ya que es una práctica habitual que todos los puertos de los servidores estén cerrados al tráfico, si no se crea una regla de excepción para los puertos 3306 (BD) y 80 (NGINX) no se podrá acceder a la BD y al servicio NGINX respectivamente.



## 10. Resultados

Los objetivos propuestos para esta primera versión se han cumplido con creces. Se ha conseguido desarrollar un aplicativo que cumple y satisface los objetivos planteados. Además, durante el proceso de desarrollo se ha contactado con varias asociaciones de hogueras, que están expectantes de ver el resultado final, ya que algunas han podido participar en el proceso de pruebas.

Como resultado final, podemos concluir que el proyecto se ha podido realizar en el tiempo estimado, que ha cumplido los objetivos y, además, ha servido para que, como estudiante, refuerce todo el conocimiento adquirido durante el grado de ingeniería en informática.





## 11. Pruebas

Para verificar que todo el proyecto esté bien desarrollado debemos probar todas las funcionalidades de la forma más exhaustiva posible.

### 11.1. Pruebas de funcionalidades

Todas las funcionalidades del proyecto han sido probadas ampliamente al finalizar el desarrollo de cada una de ellas. Para realizar estas pruebas se realizaba un despliegue del aplicativo y se seguía una a una las condiciones de satisfacción de forma manual, en este caso, el usuario era un compañero de la carrera, al que le pedía que intentase completar las condiciones de satisfacción de cada funcionalidad.

Una vez mi compañero comprobaba que todo funcionase podíamos asegurar que esta funcionalidad ha pasado la primera prueba.

### 11.2. Pruebas de usuario.

Se solicitó a todas las asociaciones de hogueras y barracas una participación en una encuesta previa, algunas de ellas decidieron participar de forma activa en el proceso de comprobación final.

Durante este proceso, los usuarios seleccionados recibían un usuario y contraseña en sus correos electrónicos y comprobaban, en el aplicativo ya desplegado en producción, que las funcionalidades cumplían las condiciones de satisfacción.

Gracias a esta prueba, también se pudieron cambiar o reorganizar algunos elementos del proyecto, haciéndolo más intuitivo para el usuario y centrando el diseño en la experiencia del usuario.

### 11.3. Pruebas de código

Si bien es cierto que el *frontend* no se ha podido probar con pruebas automatizadas, en el *backend* sí que se ha podido realizar una serie de pruebas automáticas gracias al *framework* de pruebas de JavaScript Mocha<sup>54</sup>.

También se realizaron algunas pruebas manuales con Postman, aunque estas no se pueden integrar en el código y dependen de la herramienta externa.

---

<sup>54</sup> <https://mochajs.org/>

## 12. Conclusiones y trabajo futuro

Como estudiante, estoy satisfecho con el trabajo realizado ya que he podido desarrollar de inicio a fin una idea propuesta por mí mismo surgida de una necesidad de un ámbito en el que desarrollo parte de mi vida privada.

Se trata de un proyecto de informática de gestión en un entorno web donde la innovación ha consistido en usar tecnologías recientes y novedosas. Además, dada la importancia de disponer de una herramienta como esta, he preferido dar prioridad a cubrir esta necesidad tecnológica de un colectivo que hasta el momento no la ha podido resolver satisfactoriamente.

Debido a la escasez de tiempo no se ha podido desarrollar todas las funcionalidades. Pero en un futuro próximo, se seguirá trabajando en una nueva versión que incluya algunas de las funcionalidades que quedan pendientes por realizar, como, por ejemplo, el perfil del tesorero para la intranet. Que aportará un gran valor al aplicativo.

No obstante, se propone a continuación algunas mejoras para implementar de cara al futuro y con ello dotar al aplicativo de un mayor valor tecnológico.

- Permitir al usuario realizar pagos a la asociación.
- Permitir al tesorero de la asociación efectuar pagos a proveedores.
- Permitir al tesorero tener una proyección de ingresos futuros con machine learning<sup>55</sup>.
- Introducir búsquedas inteligentes para la documentación de una asociación a partir de un repositorio o índice generado automáticamente.
- Permitir al usuario cambiar su método de pago y la frecuencia de estos.
- Crear un foro para los socios.

---

<sup>55</sup> <https://cleverdata.io/que-es-machine-learning-big-data/>

- Permitir al secretario realizar una votación durante una reunión.
- Permitir al secretario establecer tiempos de palabra.

Finalmente, como trabajo a corto plazo se habilitará el aplicativo para un número determinado de asociados de diferentes asociaciones y así realizar una prueba a mayor escala.