



# Deusto

Facultad de Ingeniería  
Universidad de Deusto

Ingeniaritza Fakultatea  
Deustuko Unibertsitatea

## Grado en Ingeniería Informática Informatikako Ingeniaritzako Gradua

### Proyecto fin de grado Gradu amaierako proiektua

Colmena: diseño e implementación de un widget web para micro donaciones, con página web de soporte, almacenamiento de datos y su posterior visualización

Rubén Sánchez Corcuera

Director: Pablo García Bringas

Bilbao, mayo de 2017



## Resumen

El proyecto consta de varios módulos. La base de todo es un servidor implementado en node.js. En él se aloja la página web, los widgets de las diferentes empresas que han decidido implantarlo en sus páginas web y una restful API que administra el enrutamiento de la web y permite varias funciones sobre la base de datos.

La página web sirve de apoyo al widget, es donde se muestra la información general del widget, los diferentes proyectos a los que apoyar, un wizard con el que poder crear tu propio widget y la plataforma para poder conseguir el certificado de donación.

La base de datos noSQL aloja los datos de las diferentes donaciones para posteriormente poder crear el certificado de donación y eventualmente generar estadísticas sobre las cantidades, fechas y lugares en los que más donaciones se están obteniendo.

Por último, el widget. Está pensado para que las empresas que tengan portales de compra online lo incrusten durante algunas de sus fases compra del cliente con el fin de que este haga una pequeña donación a la causa que las empresas han “apadrinado”. El widget es personalizable por las empresas que lo quieran implantar en su web, así conseguir una mayor integración con ellas. Entre sus opciones de personalización destaca la posibilidad de elegir entre una cantidad fija o variable de donación.

## Descriptores

solidaridad, widget, RESTful, NodeJS, MongoDB



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Presentación del documento . . . . .	1
<b>2. Objetivos del proyecto</b>	<b>3</b>
2.1. Alboan . . . . .	3
2.1.1. Misión y Visión . . . . .	3
2.1.2. Dedicación . . . . .	4
<b>3. Situación de partida</b>	<b>5</b>
3.1. Necesidades . . . . .	5
3.2. Objetivos . . . . .	6
3.3. Requerimientos . . . . .	6
3.4. Estado del arte . . . . .	6
3.4.1. Crowdfunding . . . . .	6
3.4.2. Micro donaciones . . . . .	7
<b>4. Definición del proyecto</b>	<b>9</b>
4.1. Descripción del producto . . . . .	9
4.1.1. Widget . . . . .	9
4.1.2. Pagina web . . . . .	10
4.1.3. Base de datos . . . . .	11
4.1.4. Servidor . . . . .	11
4.1.5. Visualización . . . . .	11
4.2. Descripción de la realización . . . . .	13
4.2.1. SCRUM . . . . .	13
4.2.2. Trello . . . . .	13
4.2.3. Tareas . . . . .	14
4.3. Productos intermedios . . . . .	16
<b>5. Condiciones de ejecución</b>	<b>17</b>
5.1. Instalaciones . . . . .	17
5.1.1. Hardware . . . . .	17
5.1.2. Software . . . . .	17
5.2. Equipo de proyecto . . . . .	18
5.2.1. Interacción con el cliente . . . . .	18
5.2.2. Perfiles profesionales . . . . .	18
5.2.3. Procedimientos de seguimiento y control . . . . .	19
<b>6. Planificación</b>	<b>21</b>
6.1. Presupuesto . . . . .	21
6.2. Planificación del proyecto . . . . .	22
6.3. Justificación financiera . . . . .	22
<b>7. Especificación de requisitos</b>	<b>25</b>
7.1. Visión general . . . . .	25
7.2. Especificación de requisitos del servidor Node.js . . . . .	25
7.3. Especificación de requisitos de la página web . . . . .	25
7.4. Especificación de requisitos del widget . . . . .	26
7.5. Especificación de requisitos de la base de datos . . . . .	26

7.6. Especificación de requisitos del sistema de visualización . . . . .	26
<b>8. Tecnologías utilizadas</b>	<b>27</b>
8.1. Node.js . . . . .	27
8.1.1. Funcionamiento . . . . .	27
8.1.2. Express . . . . .	28
8.1.3. Express-ejs-layouts . . . . .	28
8.1.4. Nodemailer . . . . .	28
8.1.5. Mongojs . . . . .	29
8.1.6. Pdffiller . . . . .	29
8.2. MongoDB . . . . .	30
8.2.1. Razón de uso . . . . .	30
8.3. JSON . . . . .	30
8.3.1. Funcionamiento . . . . .	30
8.4. Sass (CSS3) . . . . .	31
8.4.1. Razón de uso . . . . .	31
8.5. D3.js . . . . .	31
8.5.1. Funcionamiento . . . . .	32
8.5.2. Razón de uso . . . . .	32
<b>9. Especificación del diseño</b>	<b>33</b>
9.1. Visión general . . . . .	33
9.2. Herramientas utilizadas . . . . .	33
9.2.1. Draw.io . . . . .	33
9.2.2. gomockingbird . . . . .	33
9.3. Diseño de la arquitectura . . . . .	33
9.4. Diseño del servidor . . . . .	34
9.5. Diseño de la página web . . . . .	34
9.6. Diseño del widget . . . . .	34
<b>Bibliografía</b>	<b>37</b>

# Índice de figuras

2.1. Logo de Alboan . . . . .	3
3.1. Ejercicio de cuentas . . . . .	5
4.1. Infraestructura de Colmena . . . . .	9
4.2. Widget Colmena . . . . .	10
4.3. Proceso scrum . . . . .	13
4.4. Tablón Trello . . . . .	14
5.1. Gráfico de la interacción en el proyecto . . . . .	18
5.2. Gráfico del equipo de proyecto . . . . .	19
6.1. Lista de tareas con su coste en horas . . . . .	21
6.2. Diagrama Gantt . . . . .	23
8.1. Diagrama de la gramática de un objeto JSON . . . . .	30
8.2. Ejemplo de un JSON . . . . .	31
8.3. Scss y css compilado respectivamente . . . . .	32
9.1. Arquitectura del proyecto . . . . .	34





# Índice de tablas

6.1. Presupuesto del proyecto desglosado . . . . . 21



# 1. INTRODUCCIÓN

---

## 1.1 Presentación del documento

El presente documento recoge proyecto desarrollado por el alumno Rubén Sánchez para la ONG ALBOAN. En el proyecto consta de todo el sistema necesario para crear un sistema de micro donaciones mediante un widget 5 en cualquier tienda online. El sistema consta de: una página web, una base de datos, el widget y un servidor que da soporte a todo. Además, el sistema cuenta con una funcionalidad añadida por la que se puede crear gráficas de visualización.

El proyecto se desarrolla para la ONG Alboan la cual al ver el cambio que se da en la sociedad en materia de solidaridad y que las grandes donaciones han sufrido una gran caída decide que la manera de recoger las donaciones tiene que cambiar. Alboan, al explorar las oportunidades y ver que la única opción disponible cobra a las ONGs un porcentaje de las donaciones decide crear un sistema de micro donaciones gratuito que cualquier empresa pueda añadir a su tienda online y así fomentar las donaciones a pequeña escala, consiguiendo así que el 100 % del dinero vaya al destino final.

Los principales capítulos del documento son los siguientes:

- **Definición del proyecto**

Establecimiento del objetivo fundamental del proyecto, especificando su alcance.

- **Producto final**

Especificación de los elementos que componen el proyecto Colmena.

- **Organización**

Definición del equipo de trabajo que desarrollará el proyecto y los perfiles profesionales que formaran parte de este. También incluye la estructura organizativa y el sistema utilizado para gestionar el proyecto.

- **Condiciones de ejecución**

Definición del entorno de trabajo y del hardware y software a utilizar y de la metodología que se utilizará para hacer las modificaciones o mejoras que alteren el planteamiento inicial en el proyecto.

- **Planificación**

Estimación de la duración de las tareas durante el transcurso del proyecto, así como su planificación en el tiempo.

- **Valoración económica**

Determinación del valor correspondiente a este proyecto, las horas de desarrollo y las herramientas y elementos utilizados.



## 2. OBJETIVOS DEL PROYECTO

---

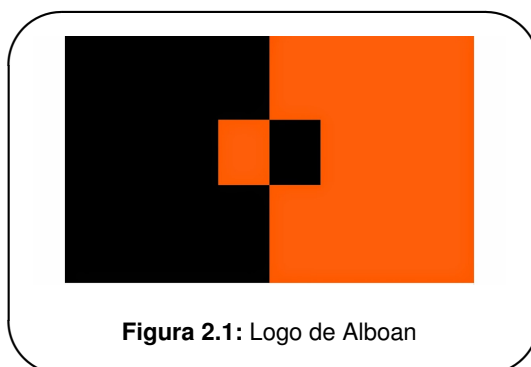
En este capítulo hablaremos sobre el beneficiario del proyecto, en este caso es la ONG ALBOAN.

### 2.1 Alboan

La actividad de Alboan comenzó en 1994, asumiendo las iniciativas de voluntariado internacional que ya estaban en marcha, pero fue en 1996 cuando se configuró jurídicamente bajo la figura de Fundación. Es la ONG promovida por la Compañía de Jesús en la Provincia de Loyola (País Vasco y Navarra). El nombre, en euskera, Alboan, quiere reflejar el arraigo a la cultura de la tierra vasca y el espíritu de la entidad: estar al lado de las personas más excluidas, junto a organizaciones y centros educativos.

Su logo(Figura 2.1) visibiliza el rol de bisagra y puente que quería jugar la institución para poner en relación dos mundos que en realidad son uno solo. Su misión: ser plataforma de encuentro de personas y organizaciones de aquí y allá que quisieran comprometerse en la construcción de un mundo mejor.

Hoy Alboan cuenta con 7000 personas entre contratados, voluntarios y entidades que le permiten apoyar y acompañar a más de 200 proyectos llevados a cabo por 105 organizaciones aliadas que impactan directamente en las vidas de más de 600.000 personas en todo el mundo.



**Figura 2.1:** Logo de Alboan

#### 2.1.1 Misión y Visión

##### ■ Misión

Trabaja por la construcción de una ciudadanía global que denuncie las injusticias que provocan desigualdad en el mundo, construya una cultura que promueva el bien común y transforme las estructuras generadoras de pobreza a nivel local y global. Para lograrlo, se une en red con personas y grupos de todo el mundo.

La colaboración de Alboan se centra en 4 temáticas principales en las que enmarca todos sus proyectos y labores de ayuda: Educación de calidad, Desarrollo económico-productivo sostenible y equitativo, Acción humanitaria en crisis recurrentes y Democracia a favor de las personas excluidas.

Todas las actividades que realiza la ONG incluyen 3 ejes transversales que otorgan a los proyectos el carisma que Alboan quiere dar:

- La espiritualidad como dimensión en el horizonte de desarrollo humano.
- El reconocimiento de las desigualdades entre mujeres y hombres y el compromiso con la equidad de género.
- La participación ciudadana para la incidencia social y política.

### ■ Visión

Lograr una Alboan que contenga las siguientes características:

- **Enraizada** en el nuevo proyecto unificado de la Compañía de Jesús a través de sus plataformas locales y territorial.
- **Querida** por las organizaciones y la base social con las que se alía.
- **Reconocida** por su valor añadido en el acompañamiento a entidades, la formación y la construcción de ciudadanía global.
- **Sostenible** gracias a un equipo comprometido y una financiación estable y diversificada.
- **Ilusionante**, por sus propuestas y su comunicación esperanzadora.
- **Puente** entre nuestro estilo de vida y las situaciones de frontera de deshumanización

### 2.1.2 Dedicación

Alboan tiene unos ámbitos de dedicación marcados en los que

- La **Cooperación al Desarrollo** en África, Asia, Centroamérica y Sudamérica, principalmente con proyectos educativos, de promoción económica y de formación de grupos excluidos para la defensa de sus derechos.
- La **Educación para la solidaridad** en Euskadi y en Navarra, mediante la participación en campañas, la elaboración de materiales educativos, la promoción del comercio justo, impartiendo formaciones y talleres y asesorando a grupos y centros educativos.
- La **Acción Política** participando en redes y elaborando estudios e investigaciones para incidir y mejorar las políticas que afectan al desarrollo tanto locales como internacionales.

### 3. SITUACIÓN DE PARTIDA

En este capítulo se habla sobre la necesidad de la cual surge crear este proyecto, la motivación por la cual se decide desarrollar el producto final y los objetivos que se pretender conseguir con el mismo.

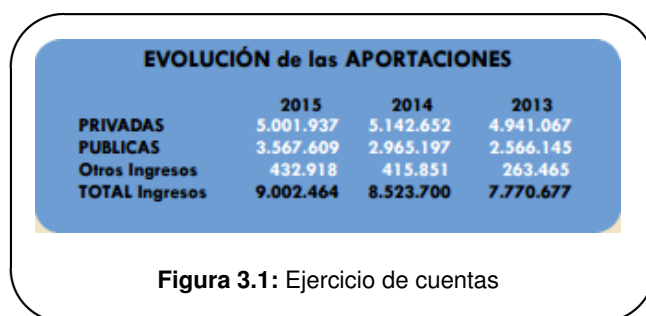
#### 3.1 Necesidades

Alboan ingreso el año 2015 nueve millones de euros. Este dinero proviene de donaciones tanto privadas como públicas. Las donaciones o ayudas públicas dependen del gobierno del año en la que las recibe, por lo que estas no tienen una solución posible ya que no dependen de nadie más que del gobierno. En cambio, las donaciones privadas, provenientes de personas o instituciones no públicas. Gracias a todas estas donaciones Alboan tiene 200 proyectos activos en 18 países diferentes, esto hace que mantener el dinero que la ONG invierte en cada uno de ellos sea crucial año tras año.

En el ejercicio de 2014 la financiación privada crece gracias a los legados solidarios<sup>1</sup> que la gente deja a la organización, también son muy importantes las cuotas de los socios que representan un 28 % de las aportaciones privadas. Estas donaciones son importantes para que Alboan cada vez pueda llegar a más gente y pueda crear más proyectos necesarios.

Años atrás Alboan comienza a detectar una bajada en la forma tradicional de hacer donaciones. Es entonces cuando se plantea buscar nuevos métodos para realizar estas donaciones y que el dinero siga fluyendo. En el ejercicio de 2015 la financiación privada fue ligeramente menor que el año anterior y se rompió la tendencia en alza. A parte de tener menos financiación Alboan se da cuenta de que algunos de sus proyectos no son conocidos y que la ayuda para que estos continúen es crucial, por lo que la preocupación hacia estos proyectos crece.

Es entonces cuando le surgen a Alboan varias necesidades, la de volver a conseguir financiación privada ya que no pueden sustentarse únicamente de los legados solidarios y las cuotas que pagan sus socios y por otra parte la de hacer llegar a más personas sus proyectos para que de esta manera aumente su sensibilidad con el tema. Para lograr cubrir estas necesidades Alboan intenta pensar en una solución en la que los proyectos de la ONG se den a conocer y que la gente que este interesada en ellos pueda hacer una pequeña aportación, en este momento surge la idea de las pequeñas donaciones.



<sup>1</sup>Son las herencias que ciertas personas dejan a entidades solidarias con el fin de construir un mejor mundo.

## 3.2 Objetivos

El principal objetivo de este proyecto es crear un widget solidario que se pueda implantar en cualquier tienda online y crear la página web en la que se va a apoyar y se dará a conocer este widget. La página web también será el portal en el que la gente que ha realizado alguna donación pueda recoger su certificado de donación con el fin de presentarlo en la declaración de la renta.

Otro objetivo sería el de crear el servidor con la base de datos para alojar los datos de las donaciones realizadas mediante el widget. Este servidor sería accesible para las personas de Alboan que quisieran consultar los datos de las donaciones o ver las gráficas relacionadas con esto.

## 3.3 Requerimientos

En este apartado listaremos los diferentes requerimientos del proyecto.

- Formar a las personas que realizarán el mantenimiento del sistema mediante tutoriales.
- Utilizar las últimas tecnologías en diseño web.
- Crear una web responsiva para que todos los usuarios puedan acceder a ella y que evite la brecha digital.
- Garantizar los tiempos de respuesta del widget para que las empresas que lo usen no vean perjudicados sus comercios online.
- Visualización de datos comprensible e intuitiva.

## 3.4 Estado del arte

En este apartado se hace un análisis de las diferentes opciones que hay para cubrir las necesidades citadas anteriormente.

### 3.4.1 Crowdfunding

“Cooperación colectiva, llevada a cabo por personas que realizan una red para conseguir dinero u otros recursos, se suele utilizar Internet para financiar esfuerzos e iniciativas de otras personas u organizaciones”[?]

En el crowdfunding hay diferentes tipos de ayudas y de sistemas creados para financiar los proyectos que la gente sube a las plataformas: basado en donaciones (migranodearena, teaming...), basados en recompensas(Goteo, Verkami...) y para préstamos(ECrowd!) entre otros.

#### ■ Basado en donaciones

El usuario realiza una donación para ayudar a un proyecto que le gusta y no recibe ningún beneficio económico en retorno (solo la satisfacción de hacer algo bueno y de apoyar un proyecto que le emociona). Son proyectos solidarios con un impacto social grande.

##### ■ Migranodearena.org:

La página web migranodearena es un proyecto creado por la Fundación Privada real dreams. Es una plataforma de crowdfunding solidario, pionera en España, abierta para todas las Entidades No Lucrativas, legalmente constituidas y con sede en España.

En migranodearena se puede liderar un reto y recaudar fondos en grupo a favor de una ONG. Una persona o grupo de personas, lanza un reto en migranodearena para apoyar una causa solidaria. Comparte el reto con todos sus familiares, amigos, conocidos, clientes, empleados, proveedores y entre todos suman granitos de arena.

##### ■ Teaming:

Teaming es una herramienta online para recaudar fondos para causas sociales a través de micro donaciones de 1 € al mes. La filosofía de Teaming se basa en la idea de que con 1 €, nosotros solos no podemos hacer mucho pero si nos unimos, podemos conseguir grandes cosas.

El sistema de funcionamiento de Teaming es muy sencillo, puedes publicar un grupo en el que la gente se apunte y done 1 € al mes o puedes apuntarte a un grupo que otra persona haya publicado para donar tu ese euro. Entre las ideas clave mas destacadas de la plataforma están la de solo donar 1 € al mes, ni más ni menos y la de solo aceptar causas sociales.



## ■ Basado en recompensas

Se trata de aportar dinero a un proyecto y se recibe a cambio una recompensa. Los proyectos pueden consistir en crear y diseñar un nuevo producto o financiar un proyecto cultural (película, festival, musical). Es el modelo de crowdfunding más utilizado y conocido. En cada proyecto se puede elegir entre un rango de recompensas en función de la cantidad de dinero que se aporte.

### ■ Goteo:

A simple vista, Goteo es una plataforma de crowdfunding cívico y colaboración en torno a iniciativas ciudadanas, proyectos sociales, culturales, tecnológicos y educativos. Con réplicas y alianzas en varios países, gracias a su código abierto, además de reconocida y premiada internacionalmente desde 2011. Constituye una herramienta de generación de recursos, gota a gota, para una comunidad de comunidades compuesta por más de 65.000 personas, con un porcentaje de éxito de financiación superior al 70 %.

Pero en realidad Goteo es mucho más que eso. Tras la plataforma existe una fundación sin ánimo de lucro y un equipo multidisciplinar desde el que desarrollan herramientas y servicios de co-creación y financiación colectiva. Con una misión común vinculada siempre a principios de transparencia, progreso y mejora de la sociedad.

## ■ Para préstamos

El usuario realiza una inversión en un proyecto o en una empresa de la plataforma y recibe un retorno económico en forma de intereses. La empresa contrata un préstamo con la plataforma de crowdlending, que lo gestiona, y el usuario recupera el dinero invertido junto a los intereses a lo largo del tiempo según las condiciones pactadas en el préstamo.

### ■ ECrowd!:

ECrowdInvest es una plataforma de crowdlending (crowdfunding en forma de préstamo) para proyectos con impacto social y medio ambiental positivo (lo que venimos a llamar impacto positivo). Es una clara versión del clásico "win-win", donde todos ganan, ya que los buenos proyectos consiguen financiarse, los inversores reciben unos intereses mucho más altos de los que se obtienen con los bancos y siempre se genera un impacto positivo sobre el medio ambiente, por ejemplo, en la financiación colectiva de proyectos que impliquen la reducción de emisiones de dióxido de carbono hacia la atmósfera.

## 3.4.2 Micro donaciones

Las micro donaciones, como su nombre indica, son donaciones muy pequeñas realizadas con el fin de aportar una pequeña cantidad de dinero al proyecto que nos interesa. En este ámbito nos encontramos con Worldcoo una herramienta online con bastante recorrido y reconocimiento.

### ■ Worldcoo:

Worldcoo es una empresa nacida en 2012 que pretende crear un nuevo canal de financiación mediante un widget incrustado en tiendas online. Esto permite que las miles de personas que acceden y compran en las tiendas online puedan donar una pequeña cantidad al proyecto que esa página tiene apadrinado.

Cuentan con un extenso equipo de personas y de embajadores en varios países del mundo, lo que les da acceso a muchos comercios online. Las empresas contactan con Worldcoo y acuerdan entre los dos la instalación del widget en su tienda online. Este widget está relacionado con uno de los proyectos de las ONGs a las que Worldcoo ayuda.

Esto es exactamente lo que Alboan quiere, pero tiene algunas carencias. La empresa Worldcoo se queda un 8 % de las donaciones realizadas a cada proyecto por lo que esto empieza a disgustar a Alboan, esta empresa tampoco permite ver las personas que están donando a los proyectos por lo que no permite mantener el contacto con esas personas para hacerles conscientes de que su ayuda tiene un resultado.

Después de hacer un análisis de todas las opciones y ver los resultados Alboan decide crear su propia plataforma de micro donaciones mediante widgets en tiendas online.



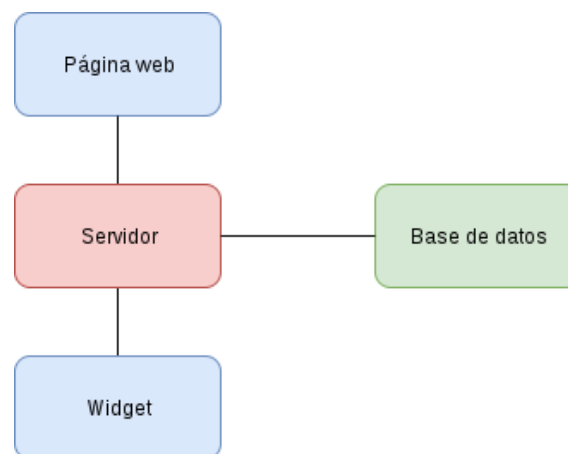
## 4. DEFINICIÓN DEL PROYECTO

En este apartado se procede a definir el proyecto, la metodología utilizada para desarrollarlo y el alcance del mismo. Durante todo el desarrollo del proyecto y las decisiones tomadas en el mismo se ha tenido en mente lo escrito en este apartado.

### 4.1 Descripción del producto

En este apartado se presenta una descripción completa del producto final, el cual consta de 4 partes, que juntas, forman el sistema completo Colmena.

El sistema Colmena consta de las 4 partes que se muestran en la Figura 4.1. El servidor central que une las otras 3 partes es el núcleo del proyecto. A este servidor se conectan los otros 3 componentes de la infraestructura completa a las que el servidor otorga funcionalidad. En la página web se permite al usuario informarse sobre los proyectos y sobre el proyecto Colmena en general. También ofrece la expedición de certificados y la creación de nuevos widgets. En la base de datos se almacenan los datos respectivos a las donaciones y a los donantes, en este caso el servidor extrae e introduce datos en ella para poder gestionar los certificados de donación. Por último, el widget, este es el producto final que la mayoría de los usuarios verán. Este se coloca en los comercios online para que las personas puedan hacer una pequeña aportación a los proyectos que la tienda haya apadrinado.



**Figura 4.1:** Infraestructura de Colmena

#### 4.1.1 Widget

El widget es la parte principal del proyecto. Es el objeto que las empresas colocaran en sus tiendas online y que permitirá a los clientes hacer una donación. Consiste en un pequeño rectángulo(4.2) en el que se le ofrece al a persona que está leyéndolo que aporte una pequeña cantidad de dinero, siempre entre 1 y 5 euros, a un proyecto en concreto.

El widget es 100 % personalizable. En la web existe un asistente que permite a las empresas personalizar el widget de modo que este encaje bien en sus comercios online. Este asistente permite cambiar las siguientes características del widget:

- **Proyecto:** Permite elegir a que proyecto destinar el dinero entre los proyectos de Alboan.
- **Opciones de donación:** Permite decidir si la donación será estática, de 1 €, o si esta podrá ser variable, entre 1 y 5 euros.

- **Fondo:** Permite elegir entre una foto, única para cada proyecto, o un fondo de color estático mediante un selector de color.
- **Color de la fuente:** Permite elegir el color de la fuente

El widget está compuesto de lenguaje HTML5 y CSS3, no tiene ningún plugin externo, lo que permite incrustarlo en cualquier página web sin que esta se vea afectada o haya que hacer alguna modificación en ella. Se ha desarrollado para que sea responsivo.

Por último, la inclusión del widget en la tienda esta simplificada al máximo de manera que la tienda no se vea afectada o tenga que realizar labores de reestructuración del código de su página web. Este proceso se hace mediante un script de código en JavaScript que permite añadir, en el espacio reservado para el mismo, el widget.



Figura 4.2: Widget Colmena

#### 4.1.2 Pagina web

La página web de la Colmena cubre las labores de publicidad y de soporte del widget. Esta incluye funcionalidades como recoger el certificado de donación después de haber realizado una, crear nuevos widgets para las empresas que quieran añadirlos a su página web o contacto con el soporte de Alboan.

La página esta desarrollada en *HTML5*, *CSS3* (mediante el pre procesador *sass*) y *JavaScript*. A parte de estas tecnologías se ha utilizado el framework web *Bootstrap* y una gran cantidad de plugins que permiten una mejor navegabilidad en la página web, entre ellos destaca *jQuery*. Se han seleccionado estas tecnologías tras una investigación sobre las tecnologías más recomendadas para el tipo de proyecto. Dado que no teníamos ningún requerimiento sobre estas, hemos apostado por las tecnologías que más nos apetecía utilizar, innovando en algunas de las situaciones y siendo más conservadores en otras.

La página web permite al usuario consultar cierta información y realizar varias opciones. Estas opciones se explican a continuación:

- **Conocer el proyecto Colmena:** Es el comienzo de la página web, tras la introducción, compuesta por una foto, está localizada este apartado. Aquí se explica el funcionamiento y la filosofía del proyecto Colmena.
- **Conocer proyectos de Alboan:** En este apartado se exponen los diferentes proyectos que dispone la ONG Alboan. Los proyectos están expuestos con el nombre de este y el país en el que actúa. Si hacemos clic en cualquiera de los proyectos una ventana se despliega y ofrece más información sobre este.
- **Crear un widget:** En este apartado de la página web se ofrece la posibilidad de crear un widget personalizado. Al hacer clic en el botón que ofrece esta posibilidad se despliega una ventana con un asistente. En este asistente se ofrecen varias opciones para personalizar el widget final. Este asistente no es el definitivo que las empresas usaran para realizar el widget que quieren que sea colocado en su comercio online.
- **Recibir un certificado:** En este apartado los usuarios pueden utilizar el código de donación que reciben al donar en cualquier tienda que tenga implantada la Colmena. Después de introducir los datos en un formulario, se le enviará el certificado de donación al mail que ha especificado.
- **Contactar con el soporte de Colmena:** Este es el último apartado de la página web. En él se puede enviar un mail al soporte de Colmena. Es el método de interlocución entre las empresas que quieran implantar el widget en sus comercios y el soporte de Colmena.

### 4.1.3 Base de datos

La base de datos de la Colmena está desarrollada en la base de datos NoSQL por excelencia, MongoDB. Se ha utilizado este tipo de base de datos por la versatilidad que proporciona a la hora de acceder a los datos y al conectarse a ella. Es una base de datos muy ágil por lo que permite consultar en un tiempo muy reducido las donaciones. Por último, se ha utilizado esta base de datos por la experimentación con la misma, al no ser una base de datos SQL el equipo tuvimos las ganas de probarla y ver su potencial.

La base de datos se encarga de almacenar todos los datos relativos a las donaciones mediante los siguientes campos:

- **Importe:** El importe de la donación.
- **Usada:** Un booleano que marca si la donación ha sido canjeada por el certificado o no.
- **Fecha:** La fecha dividida por día, mes y año.
  - Día
  - Mes
  - Año
- **idDonacion:** Un id asignado a cada donación.

Una vez almacenada la donación esta está disponible para canjearla por un certificado de donación en la página web. Una vez el certificado de donación es expedido, los datos de la persona donante se guardan en la base de datos con el fin de hacerle llegar información sobre los proyectos, si así lo desea, o hacer diferentes estadísticas con las que la organización pueda mejorar o cambiar sus métodos. Estos son los datos que se añaden al archivo de la donación:

- **DNI/CIF:** Número de identificación fiscal de la persona física o jurídica.
- **Nombre y Apellidos:** El nombre y los apellidos de la persona física o nombre de la empresa.
- **Razón social:** Denominación por la cual se conoce colectivamente a la empresa y en caso de ser una persona, introducirá la palabra "Individuo"
- **Correo electrónico**
- **Dirección**
- **Código Postal**
- **Población**
- **Provincia**

### 4.1.4 Servidor

El servidor del proyecto es el núcleo del mismo. En el convergen todas las funcionalidades de la solución. Este está desarrollado en Node.js. Tras una investigación de las posibles tecnologías y tras ver que no había requerimientos en este aspecto decidimos utilizar esta herramienta ya que es innovadora y emergente.

La funcionalidad que el servidor ofrece a los demás componentes del proyecto se basa en recabar la información necesaria de la base de datos y ofrecérsela a la página web para que esta la utilice en sus funciones. También recaba la información del widget y se lo envía a la base de datos para que esta la almacene. Por último el servidor también se utiliza como sistema de almacenamiento de los widgets para tenerlos centralizados y poder repararlos rápidamente en caso de error.

El servidor se desarrolla de esta manera, como pieza central, por el hecho de unificar todas las funcionalidades que el sistema pueda necesitar en el mismo nodo. Gracias a la flexibilidad de Node.js y a los plugins que se ofrecen en *NPM*, el gestor de paquetes para *JavaScript*. Con estos plugins se ha conseguido unificar todas las funcionalidades que sin ellos habría que haber desarrollado mediante otros métodos, por ejemplo, el envío de mails o el sistema de creación del certificado de donación.

### 4.1.5 Visualización

La visualización de los datos es la parte final del proyecto. Este módulo permite visualizar mediante una serie de gráficos interactivos los datos de las donaciones. Los gráficos, como ya he dicho anteriormente son interactivos para que los usuarios puedan 'jugar' con ellos e ir relacionando los datos. Estos gráficos están divididos en dos:

- **Gráfico de queso:** En este gráfico interactivo los usuarios pueden ir clusterizando la información en diferentes divisiones. La primera división por ejemplo sería la dividir las donaciones por el proyecto al que han donado y posteriormente se podría ir profundizando en más opciones.
- **Gráfico con mapa:** En este gráfico se puede ver un mapa de calor en el que se pueden ver las zonas desde las que más se ha donado.

Posteriormente estos gráficos se pueden analizar para sacar conclusiones de ellos y poder ofrecer esta información de vuelta a los donantes y socios de la ONG. Con esta información los donantes pueden ver lo que la entidad está recibiendo y como es la sociedad que les rodea gracias al mapa de calor. Por último, Alboan podrá analizar esta información y tomar algunas decisiones basándose en ella además de añadirla a los informes que publica.

## 4.2 Descripción de la realización

En este apartado se hablará de la metodología que se utilizará para desarrollar el proyecto. En este caso se utilizará SCRUM.

### 4.2.1 SCRUM

Scrum es la metodología que se va a utilizar para desarrollar el proyecto y se define de esta manera:

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Se va a utilizar esta metodología por las siguientes razones:

- El proyecto se espera que sea bastante dinámico por lo que esta metodología ayudará a introducir ciertas modificaciones con relativa sencillez.
- Las tecnologías que se van a utilizar en este proyecto no están dominadas por el equipo, por lo que en caso de tener que modificar alguna de ellas con esta metodología será mas sencillo.
- Con esta metodología nos será muy fácil crear entregables intermedios.

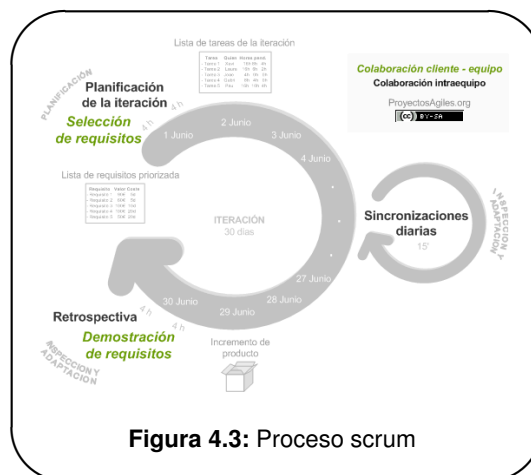


Figura 4.3: Proceso scrum

Las iteraciones o sprints serán de 3 semanas cada uno. Al comienzo de los sprints se hará una reunión para pensar cuáles deben ser las tareas que deben ir dentro de dicho sprint y al final del mismo se realizará la reunión de revisión en la que se revisará el trabajo realizado y las tareas pendientes. Estas reuniones tendrán unos productos intermedios, los cuales serán explicados más adelante.

### 4.2.2 Trello

Trello es una herramienta que sirve para organizar tareas/proyectos en un tablero (boards) al que le podemos asignar el nombre del proyecto que estamos trabajando. Este tablero está compuesto por diferentes listas: 1. "por hacer", 2. "en proceso" y 3. "finalizada". Trello te hace saber en todo momento la parte del proyecto en que te encuentras.

En esta aplicación se irán creando los diferentes paneles para cada sprint. Dentro de cada panel se organizarán las tareas acordadas en la reunión al inicio de cada sprint. En el panel los miembros del equipo se irán asignando las tareas y moviéndolas entre los diferentes estados.

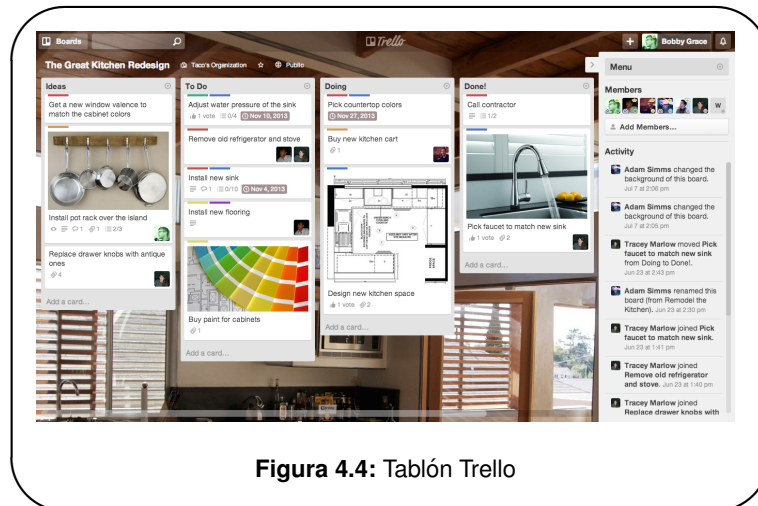


Figura 4.4: Tablón Trello

### 4.2.3 Tareas

A continuación, se procederá a listar las tareas que tendrá cada uno de los sprints. Estas tareas estarán divididas en los diferentes tableros de Trello para que se asignen a los empleados. El primer sprint mezcla tareas de gestión del proyecto con tareas de desarrollo como la creación de la arquitectura del proyecto, por lo que se puede comenzar el sprint desde el momento en el que se acuerde el proyecto. Esta es la lista de tareas:

- T1 - Sprint 1
  - T1.1 - Investigación sobre las tecnologías a utilizar
  - T1.2 - Redactar el documento sobre las tecnologías a utilizar
  - T1.3 - Instalar las herramientas de desarrollo
  - T1.4 - Configurar la herramienta de control de versiones
  - T1.5 - Diseñar la arquitectura del proyecto
- T2 - Sprint 2
  - T2.1 - Crear el primer diseño de la página web
  - T2.2 - Crear el primer diseño del widget
  - T2.3 - Desarrollar la lógica del servidor
  - T2.4 - Pruebas servidor
  - T2.5 - Configurar la base de datos
  - T2.6 - Pruebas base de datos
- T3 - Sprint 3
  - T3.1 - Desarrollar la lógica de la página web
  - T3.2 - Desarrollar la lógica del widget
  - T3.3 - Consolidar el diseño de la página web
  - T3.4 - Pruebas de la página web
  - T3.5 - Consolidar el diseño del widget
  - T3.6 - Pruebas del widget
  - T3.7 - Crear manual de usuario para los técnicos de mantenimiento
- T4 - Sprint 4
  - T4.1 - Investigar los frameworks de visualización de datos
  - T4.2 - Configurar el frameworks de visualización de datos
  - T4.3 - Pruebas sobre el frameworks de visualización de datos
  - T4.4 - Analizar los datos
  - T4.5 - Extraer las estadísticas de los datos



### ■ Investigación sobre las tecnologías a utilizar

En esta tarea se investigarán las principales tecnologías a utilizar en el proyecto principal. Se buscarán las mejores tecnologías para desarrollar un proyecto web integral. Las principales funciones de las tecnologías serán las siguientes, crear un servidor web, crear una página web, crear un widget responsivo que se comunique con el servidor y crear una base de datos que aloje los datos.

### ■ Redactar el documento sobre las tecnologías a utilizar

En esta tarea se redactará lo decidido en la tarea anterior. Se explicarán las diferentes tecnologías que se utilizarán y porque se han elegido estas.

### ■ Instalar las herramientas de desarrollo

En esta tarea se instalarán las herramientas pertinentes para el desarrollo del proyecto y su correspondiente gestión.

### ■ Configurar la herramienta de control de versiones

En esta tarea se instalará la herramienta seleccionada anteriormente para el control de versiones en online. Ya que en el desarrollo estarán implicadas más de una persona, se utilizará una herramienta de este tipo para controlar más el desarrollo.

### ■ Diseñar la arquitectura del proyecto

El objetivo de esta tarea será el de crear una arquitectura completa del proyecto que tenga en cuenta todos los elementos de este. Se crearán los diferentes proyectos en las herramientas de programación, con los métodos más generales. También habrá que crear las conexiones entre las diferentes tecnologías.

### ■ Crear el primer diseño de la página web

En esta tarea se crearán varios mockups para la interfaz de la página web, que luego serán entregados al cliente para que este las revise y decida cual desea.

### ■ Crear el primer diseño del widget

En esta tarea se crearán varios mockups para la interfaz del widget, que luego serán entregados al cliente para que este las revise y decida cual desea.

### ■ Desarrollar la lógica del servidor

En esta tarea se desarrollarán todos los métodos y servicios que el servidor tenga que implementar para dar servicio a todos los componentes del proyecto.

### ■ Pruebas servidor

En esta tarea se deberá probar el servidor ante todo tipo de situaciones para evitar fallos futuros.

### ■ Configurar la base de datos

En esta tarea se deberá configurar la base de datos para que sea capaz de alojar los datos que el servidor, el widget y la página web.

### ■ Pruebas base de datos

En esta tarea se deberá probar la base de datos ante todo tipo de situaciones para evitar fallos futuros.

### ■ Desarrollar la lógica de la página web

En esta tarea se desarrollará el sistema de enrutado y de funcionalidad de la página web.

### ■ Desarrollar la lógica del widget

En esta tarea se desarrollará la funcionalidad y la conexión con el servidor del widget.

### ■ Consolidar el diseño de la página web

En esta tarea se establecerá el diseño definitivo de la página web.

### ■ Pruebas de la página web

En esta tarea se deberá probar la página web ante todo tipo de situaciones para evitar fallos futuros.

### ■ Consolidar el diseño del widget

En esta tarea se establecerá el diseño definitivo de la página web.

### ■ Pruebas del widget

En esta tarea se deberá probar el widget ante todo tipo de situaciones para evitar fallos futuros.

### ■ Crear manual de usuario para los técnicos de mantenimiento

En esta tarea se deberá crear un manual que indique detalladamente como son todos los procesos que hay que realizar para mantener el sistema y crear nuevos widget para las empresas.

### ■ Investigar los frameworks de visualización de datos

En esta tarea se deberán investigar los frameworks que se utilizarán a posteriori para visualizar los datos extraídos de las donaciones. Los gráficos deberán ser interactivos.

### ■ Configurar el framework de visualización de datos

En esta tarea se configurará el framework elegido en la tarea anterior.

### ■ Pruebas sobre el framework de visualización de datos

En esta tarea se deberá probar el framework de visualización de datos ante todo tipo de situaciones para evitar fallos futuros.

### ■ Analizar los datos

En esta tarea se analizarán los datos extraídos de la base de datos y se cargarán en diagramas para su posterior visualización.

### ■ Extraer las estadísticas de los datos

En esta tarea se deberán extraer conclusiones de los datos y análisis realizados en la anterior tarea. Se deberá realizar un documento que recoja las conclusiones extraídas del proceso de análisis de los datos.

## 4.3 Productos intermedios

Gracias a la metodología que se va a utilizar es muy fácil generar productos intermedios. Durante el proceso de desarrollo del proyecto se crearán varios prototipos/mockups del widget y de la página web. Al final de cada sprint se hará una revisión del trabajo que se ha realizado, esto será posible gracias al tablón Trello en el que se podrán ver las tareas realizadas. Para definir los productos intermedios que se van a generar se procede a listarlos a continuación:

- Documento sobre el pedido del proyecto
- Documento sobre las tecnologías a usar
- Documento de seguimiento de las tareas realizadas
- Varios prototipos de la página web
- Varios prototipos del widget
- Manual de usuario para los técnicos que mantendrán el proyecto
- Documento de conclusiones sobre las donaciones realizadas

Gracias a estos productos intermedios se podrá mantener un seguimiento del proyecto y tener una mejor estimación de las tareas y los tiempos necesarios para estas.

## 5. CONDICIONES DE EJECUCIÓN

---

El desarrollo tendrá lugar en Deusto Tech. Aquí es donde tendrá su lugar de trabajo el equipo. El proyecto se desarrollará en el transcurso del año 2016-2017. DeustoTech y Alboan ofrecerán todos los medios necesarios para el buen desarrollo del proyecto.

### 5.1 Instalaciones

En este apartado se listará el software y hardware utilizado para desarrollar el proyecto.

#### 5.1.1 Hardware

Este es el hardware que se utilizará para desarrollar el proyecto:

- Lenovo ThinkPad X220
- Monitor Acer AL1714 (Segundo monitor)

#### 5.1.2 Software

Este es el software que se utilizará para desarrollar el proyecto. Este será explicado más adelante:

- Atom
- MongoDB
- Git
- Brackets

##### ■ Atom

Atom es un editor de código fuente abierto para macOS, Linux, y Windows con soporte para plug-ins escrito en Node.js, Incrustando Git Control, desarrollado por GitHub. Atom es una aplicación de escritorio construida utilizando tecnologías web. La mayor parte de los paquetes tienen licencias de software libre y es construido y mantenido por su comunidad. Atom está basado en Electrón (Anteriormente conocido como Atom Shell), Un framework que permite aplicaciones de escritorio multiplataforma usando Chromium y Node.js. Está escrito en CoffeeScript y Less. También puede ser utilizado como un entorno de desarrollo integrado (IDE). Atom liberó su beta en la versión 1.0, encima junio 25, 2015. Sus desarrolladores lo llaman un "Editor de textos hackable para el siglo XXI".

##### ■ MongoDB

MongoDB es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto. MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En lugar de guardar los datos en tablas como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

##### ■ Git

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

## ■ Brackets

Brackets es un editor de código fuente abierto escrito en HTML, CSS y JavaScript enfocado en el diseño web. Fue creado por Adobe Systems, bajo licencia del MIT y actualmente mantenido por GitHub. Brackets está disponible para Mac, Windows y Linux.

## 5.2 Equipo de proyecto

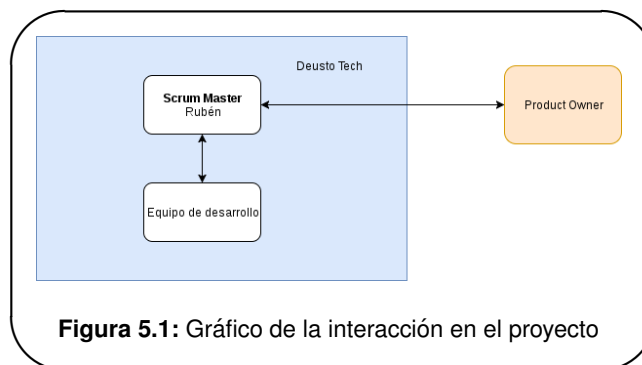
En este apartado se describirá al equipo del proyecto desde dos perspectivas diferentes. En la sección de Interacción con el cliente se describirá como deberá ser el contacto con el cliente poniendo en práctica las buenas prácticas de la metodología Scrum. En el caso de los perfiles profesionales se definirán los perfiles que deberá haber para desarrollar el proyecto y cómo será el organigrama del proyecto.

### 5.2.1 Interacción con el cliente

El esquema organizativo

- **Product Owner:** Este rol será representado por una persona de Alboan. Sus principales funciones son las siguientes:
  - Ser el representante de todas las personas interesadas en los resultados del proyecto
  - Definir los objetivos del producto o proyecto.
  - Colaborar con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.
- **Scrum master:** No se debe confundir con el jefe de proyecto. Sus principales tareas son:
  - Facilitar las reuniones de Scrum
  - Proteger y aislar al equipo de interrupciones externas
  - Quitar los impedimentos que el equipo tiene en su camino
- **Equipo de desarrollo:** Compuesto por los profesionales que desarrollarán el proyecto. Su principal función es desarrollar el proyecto y estas son otras de sus funciones:
  - Equipo auto organizado.
  - Equipo multidisciplinar, capacidad de desarrollar diferentes tareas.
  - Equipo estable durante el proyecto y establecidos en la misma localización física.

En la figura 5.1 se puede ver cómo será la interlocución entre los miembros del equipo. El Scrum master, representado por Rubén Sánchez, será el encargado de interaccionar con el Product Owner, que estará representado por una persona de Alboan. Por último, el equipo de desarrollo incluirá a las personas que desarrollen el proyecto, las cuales podremos ver más adelante.



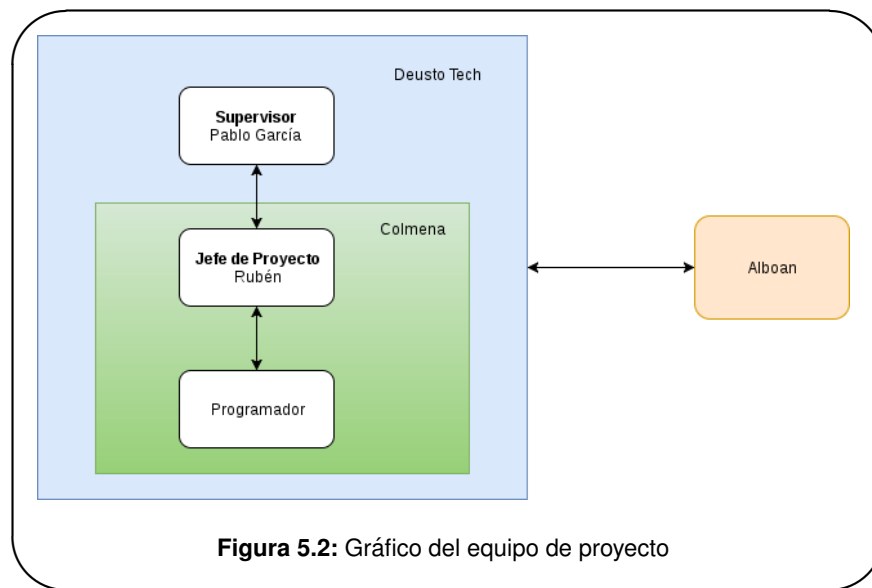
### 5.2.2 Perfiles profesionales

En este apartado se describirá al equipo de proyecto que será el encargado de desarrollar el producto y los diferentes perfiles necesarios para ello.

- **Jefe de proyecto:** Es el encargado de controlar que el proyecto se desarrolle correctamente y de garantizar los tiempos de desarrollo y la calidad del mismo. También se encargará de los aspectos de gestión del proyecto.

- **Diseñador:** Es el encargado de diseñar la interfaz de la página web y del widget.
- **Analista de datos:** Es el encargado de analizar los datos que se extraen del proyecto y de crear informes interactivos y conclusiones de ellos.
- **Arquitecto web:** Es el encargado de diseñar y desarrollar la arquitectura principal del proyecto. También se encarga de configurar las herramientas que vayan a utilizar los demás para asegurar su correcto funcionamiento.
- **Programador:** Es el encargado de desarrollar los métodos que el arquitecto haya establecido.

En la figura 5.2 se ve la organización en el equipo de proyecto. Alboan está como cliente externo a Deusto Tech. Dentro de Deusto Tech tenemos a Pablo García, el cual será el supervisor del proyecto desde la empresa, y al equipo de la Colmena. Dentro del equipo de la Colmena contamos con dos personas, el jefe del proyecto, que hará las veces de Diseñador, Analista de datos y Arquitecto Web y, por otra parte, el programador.



### 5.2.3 Procedimientos de seguimiento y control

El seguimiento del proyecto se hará desde el tablón de Trello. En el tablón se registran los cambios que se generan y se generan notificaciones para las personas involucradas en la tarea que ha sufrido cambios. Finalmente se puede sacar un reporte de los cambios en las tareas.

En cuanto a reuniones de seguimiento del proyecto, tendremos una reunión cada viernes en la que el equipo de proyecto hablará con el supervisor de Deusto Tech. En esa reunión se podrán ver los avances y problemas que hay cada semana y se pensarán diferentes soluciones para ellos. Estas reuniones también servirían por si se necesitará algo de Deusto Tech.



## 6. PLANIFICACIÓN

En este capítulo hablaremos sobre la planificación y presupuestos del proyecto.

### 6.1 Presupuesto

En este apartado se detalla el coste en días de cada tarea. Cada día está compuesto por 8h laborales. Cada tarea será realizada por un perfil profesional, cobrando cada uno su precio en horas.

	📌	Name	Duration	Start	Finish	Predecessors
1		<b>Sprint 1</b>	<b>15 days</b>	<b>4/3/17 8:00 AM</b>	<b>4/21/17 5:00 PM</b>	
2		Investigación sobre las tecnologías a utilizar	4 days	4/3/17 8:00 AM	4/6/17 5:00 PM	
3		Redactar el documento sobre las tecnolo...	1 day	4/7/17 8:00 AM	4/7/17 5:00 PM	2
4		Instalar las herramientas de desarrollo	2 days	4/7/17 8:00 AM	4/10/17 5:00 PM	2
5		Configurar la herramienta de control de v...	1 day	4/7/17 8:00 AM	4/7/17 5:00 PM	2
6		Diseñar la arquitectura del proyecto	9 days	4/11/17 8:00 AM	4/21/17 5:00 PM	3;4;5
7		<b>Sprint 2</b>	<b>14 days</b>	<b>4/24/17 8:00 AM</b>	<b>5/11/17 5:00 PM</b>	<b>1</b>
8		Crear el primer diseño de la página web	4 days	4/24/17 8:00 AM	4/27/17 5:00 PM	
9		Crear el primer diseño del widget	4 days	4/28/17 8:00 AM	5/3/17 5:00 PM	8
10		Desarrollar la lógica del servidor	6 days	4/24/17 8:00 AM	5/1/17 5:00 PM	
11		Pruebas servidor	2 days	5/2/17 8:00 AM	5/3/17 5:00 PM	10
12		Configurar la base de datos	4 days	5/4/17 8:00 AM	5/9/17 5:00 PM	11
13		Pruebas base de datos	2 days	5/10/17 8:00 AM	5/11/17 5:00 PM	12
14		<b>Sprint 3</b>	<b>15 days</b>	<b>5/15/17 8:00 AM</b>	<b>6/2/17 5:00 PM</b>	<b>7</b>
15		Desarrollar la lógica de la página web	5 days	5/15/17 8:00 AM	5/19/17 5:00 PM	
16		Desarrollar la lógica del widget	6 days	5/15/17 8:00 AM	5/22/17 5:00 PM	
17		Consolidar el diseño de la página web	4 days	5/22/17 8:00 AM	5/25/17 5:00 PM	15
18		Pruebas de la página web	2 days	5/26/17 8:00 AM	5/29/17 5:00 PM	17
19		Consolidar el diseño del widget	3 days	5/23/17 8:00 AM	5/25/17 5:00 PM	16
20		Pruebas del widget	3 days	5/26/17 8:00 AM	5/30/17 5:00 PM	19
21		rear manual de usuario para los técnicos ...	3 days	5/31/17 8:00 AM	6/2/17 5:00 PM	18;20
22		<b>Sprint 4</b>	<b>15 days</b>	<b>6/5/17 8:00 AM</b>	<b>6/23/17 5:00 PM</b>	<b>14</b>
23		Investigar los frameworks de visualización ...	2 days	6/5/17 8:00 AM	6/6/17 5:00 PM	
24		Configurar el frameworks de visualización ...	3 days	6/7/17 8:00 AM	6/9/17 5:00 PM	23
25		Pruebas sobre el frameworks de visualizac...	2 days	6/12/17 8:00 AM	6/13/17 5:00 PM	24
26		Cargar y analizar los datos	3 days	6/14/17 8:00 AM	6/16/17 5:00 PM	25
27		Extraer las estadísticas de los datos	5 days	6/19/17 8:00 AM	6/23/17 5:00 PM	26
Colmena - page1						

**Figura 6.1:** Lista de tareas con su coste en horas

Perfil profesional	Horas	Precio (por horas)	Importe
Diseñador	120	20	2400
Arquitecto web	136	20	2720
Programador	240	20	4800
Analista de datos	120	25	3000
Precio total			12920

**Tabla 6.1:** Presupuesto del proyecto desglosado

## 6.2 Planificación del proyecto

En la figura 6.2 se puede ver el diagrama Gantt del proyecto. Este está dividido en los 4 sprints de los que antes hemos hablado. Las barras negras corresponden a los sprints, cada uno de los sprints esta compuesto por unas barras que indican la duración de las tareas dentro de ellos. Las flechas indican la precedencia de las tareas

## 6.3 Justificación financiera

Este proyecto no genera beneficios ya que se trata de una plataforma para obtener donaciones para otros proyectos, por lo que el dinero recaudado por la plataforma está directamente destinado a los proyectos. Por otra parte, la plataforma ofrecerá numerosos beneficios a la ONG ya que hará más visibles sus proyectos y ofrecerá una manera más fácil de hacer donaciones a las personas que estén interesadas. También unificará la manera de recibir estas donaciones y las tendrá todas almacenadas en un sistema completo por lo que hacer estudios sobre estas donaciones será muy sencillo.



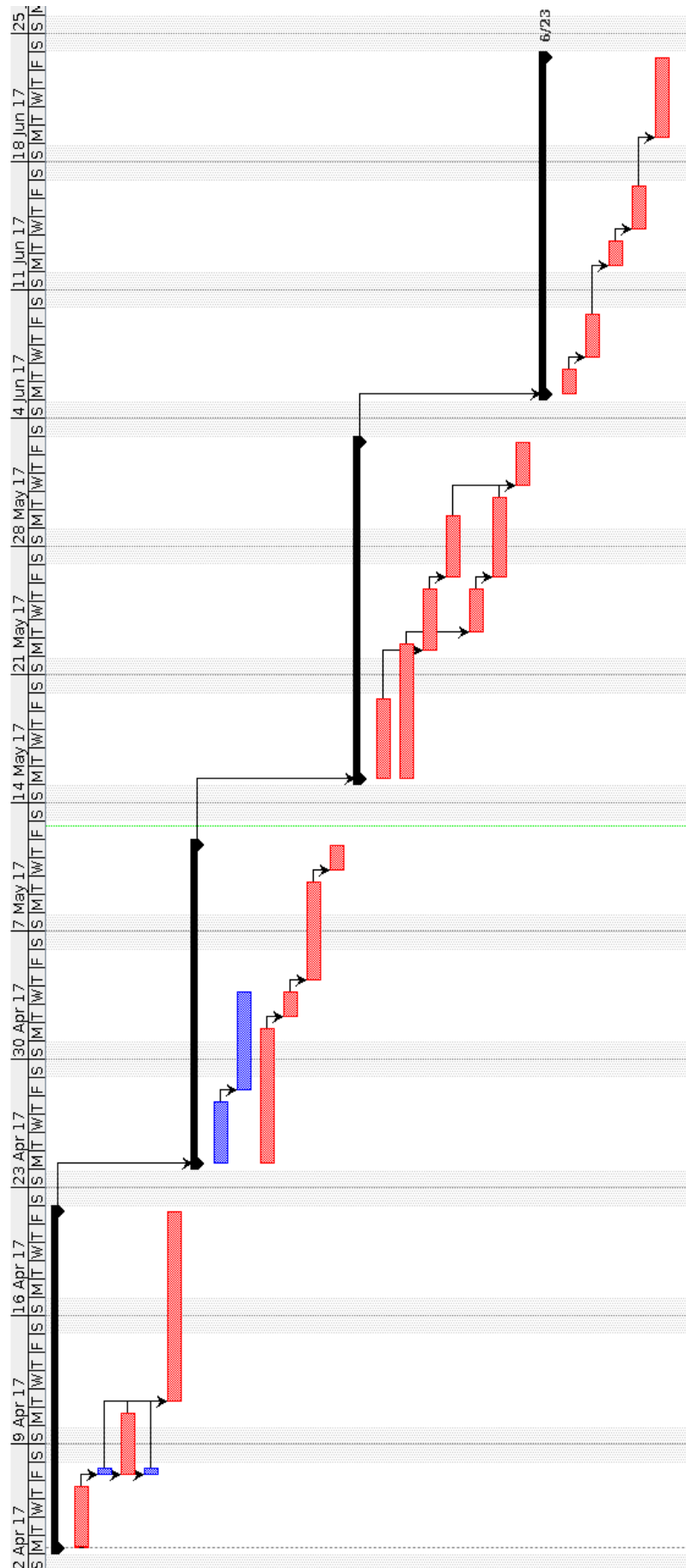


Figura 6.2: Diagrama Gantt



## 7. ESPECIFICACIÓN DE REQUISITOS

---

### 7.1 Visión general

En este capítulo se especifican los requisitos que el proyecto debe satisfacer y que definen el funcionamiento de todo el software que compone este proyecto. Para una mejor comprensión de los mismos, se dividen en los siguientes bloques:

- *Especificación de requisitos del servidor Node.js*: en esta sección se recogen los requisitos que debe de satisfacer el servidor, este es el encargado de soportar todo el sistema.
- *Especificación de requisitos de la página web*: en esta sección se recogen los requisitos que debe satisfacer la página web del proyecto.
- *Especificación de requisitos del widget*: en esta sección se recogen los requisitos que debe satisfacer el widget del proyecto, el elemento que estará incrustado en las tiendas online.
- *Especificación de requisitos de la base de datos*: en esta sección se recogen los requisitos que debe satisfacer la base de datos, esta albergará los datos de las donaciones y donantes.
- *Especificación de requisitos del sistema de visualización*: en esta sección se recogen los requisitos que debe satisfacer el sistema de visualización, el encargado de crear gráficas interactivas para la visualización y estudio de los datos.

### 7.2 Especificación de requisitos del servidor Node.js

Los requisitos funcionales del servidor Node.js son:

- **RF.0.1** El servidor debe ser capaz de albergar la página web.
- **RF.0.2** El servidor debe ser capaz de albergar las rutas de la página web y ofrecer el enrutamiento a cada una de estas.
- **RF.0.3** El servidor debe ser capaz de conectarse con la base de datos.
- **RF.0.4** El servidor debe ser capaz de enviar mails.
- **RF.0.5** El servidor debe ser capaz de alterar un PDF.
- **RF.0.6** El servidor debe ser capaz de crear un script y almacenarlo.
- **RF.0.7** El servidor debe ser capaz de ofrecer un script a páginas de terceros.
- **RF.0.8** El servidor debe ser capaz de ofrecer datos a una tercera aplicación.

Los requisitos no funcionales son:

- **RNF.0.1** Mantenibilidad: el sistema tiene que tener un mantenimiento sencillo ya que tiene conexiones con páginas de terceros lo que obliga a que el mantenimiento sea sencillo y rápido.
- **RNF.0.2** Escalabilidad: el servidor tiene que ser escalable ya que la creación de nuevos widgets o la oferta de datos a terceras aplicaciones puede ser grande.

### 7.3 Especificación de requisitos de la página web

Los requisitos funcionales de la página web son:

- **RF.1.1** La página web debe ofrecer un wizard para la creación de nuevos widgets.
- **RF.1.2** La página web debe ofrecer un formulario para la obtención de un certificado de donación.
- **RF.1.3** La página web debe ofrecer un sistema para comunicarse con el soporte del proyecto.

Los requisitos no funcionales de la página web son:

- **RNF.1.1** La página web debe informar sobre el proyecto.

- **RNF.1.2** La página web debe informar sobre los proyectos de la ONG.
- **RNF.1.3** Accesibilidad: La página web tiene que ser responsiva para ser adaptable en diferentes dispositivos.
- **RNF.1.4** Interfaz: la página web debe tener un diseño simple para facilitar la navegación.
- **RNF.1.5** Escalabilidad: la página tiene que ser escalable ya que se tienen que poder añadir nuevos proyectos a ella.

### 7.4 Especificación de requisitos del widget

Los requisitos funcionales del widget son:

- **RF.2.1** El widget debe permitir la donación fija o variable de una cantidad de dinero.
- **RF.2.2** El widget debe ser de fácil implantación por parte de la tienda online.

Los requisitos no funcionales del widget son:

- **RNF.2.1** El widget debe informar sobre el proyecto al que va destinado.
- **RNF.2.2** Accesibilidad: el widget debe ser responsivo para que pueda añadirse en cualquier tienda.
- **RNF.2.3** Interfaz: el widget debe ser 100 % personalizable.
- **RNF.2.4** Disponibilidad: el widget debe estar disponible en todo momento para su uso por parte de los comercios online.

### 7.5 Especificación de requisitos de la base de datos

Los requisitos funcionales de la base de datos son:

- **RF.3.1** La base de datos debe almacenar los datos de las donaciones.
- **RF.3.2** La base de datos debe almacenar los datos de los donantes.
- **RF.3.3** La base de datos debe proporcionar los datos que se le pidan.

Los requisitos no funcionales de la base de datos son:

- **RNF.3.1** Rendimiento: la base de datos debe almacenar y proporcionar los datos en un tiempo razonable.
- **RNF.3.2** Seguridad: la base de datos debe ser segura para no poner en peligro los datos de los donantes.
- **RNF.3.3** Disponibilidad: La base de datos debe estar disponible para almacenar las donaciones e información de donantes.

### 7.6 Especificación de requisitos del sistema de visualización

Los requisitos funcionales del sistema de visualización son:

- **RF.4.1** El sistema de visualización debe ofrecer gráficos interactivos.
- **RF.4.2** El sistema de visualización debe conectarse con el servidor para adquirir los datos.

Los requisitos no funcionales del sistema de visualización son:

- **RNF.4.1** Escalabilidad: el sistema de visualización de datos debe ser escalable ya que los datos pueden crecer y la manera de mostrarlos cambiar.
- **RNF.4.2** Interfaz: el sistema de visualización de datos debe tener una interfaz intuitiva que permita una buena navegación por los gráficos.

## 8. TECNOLOGÍAS UTILIZADAS

En esta sección se presentan las diferentes tecnologías que se han usado para el desarrollo del proyecto. Las principales tecnologías utilizadas son las siguientes:

### 8.1 Node.js

Esta es la autodefinición que se hace Node.js en su propia página web:

Node.js[5] es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema mas grande de librerías de código abierto en el mundo.

Node.js es una solución con un solo hilo de ejecución que permite que las peticiones a esta no bloqueen peticiones futuras ni exige grandes pools de hilos para tener conexiones concurrentes. Esto permite que los comercios online conecten con el servidor y le hagan una petición para recibir el widget a lo que el servidor responderá con el envío y el cierre de la conexión, evitando así el cuello de botella que se puede generar con conexiones masivas.

Node.js cuenta con un gestor de paquetes llamado npm del que se pueden descargar diferentes modulos para ampliar la funcionalidad de esta tecnología. Gracias a este gestor de paquetes node.js se convierte en una solución integral para la parte servidora habilitándole con todo lo necesario para cumplir las funciones del backend de una solución web.

Entre las ventajas que ofrece Node.js se encuentran las siguientes:

- **Gran documentación:** tiene una gran documentación y 8 años de experiencia por lo que la mayoría de los casos y posibilidades están testadas haciendo su desarrollo mas sencillo.
- **Gran comunidad:** gracias al gestor de paquetes publico y a los años de experiencia Nodejs cuenta con una gran comunidad con la que poder consultar las dudas y usos de los diferentes paquetes.
- **npm:** su gestor de paquetes publico permite reutilizar código y no perder tiempo implementando código que ya ha sido desarrollado y probado anteriormente.
- **Multiplataforma y open-source:** esta desarrollado para ser utilizado en cualquier sistema operativo y cuenta con una licencia MIT, lo cual lo hace gratuito y permite arreglar e incluso mejorar el propio código de la herramienta por los usuarios de esta.

#### 8.1.1 Funcionamiento

En Node.js es muy sencillo crear aplicaciones nuevas que actuen en la parte servidora de una aplicación. Gracias a npm, Node.js es muy polivalente, pero a continuación se mostrará un ejemplo de un servidor HTTP:

```
http.createServer(function (request, response) {
  response.writeHead(200, 'Content-Type': 'text/plain');
  response.end('Hello World');
}).listen(8081);
console.log('Server running at http://127.0.0.1:8081/');
```

En este ejemplo se crea un servidor HTTP en el puerto 8081 de la maquina local. Una vez se entra al puerto 8081 de la maquina local, el servidor creara una respuesta en texto plano y la enviará al navegador, que mostrará por pantalla el mensaje.

Node.js funciona de una manera muy diferente dependiendo de los paquetes utilizados para el desarrollo de la aplicación por lo que a continuación explicare los paquetes utilizados para el desarrollo de este proyecto y como es el funcionamiento de cada uno de estos:

### 8.1.2 Express

Express es un framework web minimalista y flexible para el desarrollo de soluciones web en node.js. Express además aplica una fina capa con las características fundamentales de las aplicaciones web sobre la base de node.js permitiendo así mantener todas las funcionalidades que ofrece Node.js. Finalmente, Express ofrece una amplia y robusta API para hacer uso de todas las funcionalidades y características que ofrece.

#### ■ Funcionamiento

Express no tiene una estructura de proyecto definida por los desarrolladores o comunidad, en cambio, tanto en la documentación como en numerosos sitios ofrecen un sistema de carpetas para tener el proyecto ordenado y las rutas definidas.

```
project/
|-- node_modules/
|-- public/
|   |-- images/
|   |-- css/
|   |-- javascript/
|-- routes/
|-- views/
|-- app.js
|-- package.json
```

- **Node\_modules:** en esta carpeta irán incluidas todas las dependencias del proyecto una vez descargadas automáticamente por npm.
- **Public:** en esta carpeta estarán los elementos de uso publico por parte del proyecto, imágenes, scripts de JavaScript y hojas de estilo.
- **Routes:** aquí se encuentran las rutas a las diferentes entidades en archivos diferentes.
- **Views:** en esta carpeta estarán incluidas las vistas del proyecto, es decir, las diferentes pantallas de la página web.
- **App.js:** es el archivo principal del proyecto.
- **Package.json:** es un archivo de configuración general del proyecto, en el irán las dependencias y los metadatos del proyecto.

#### ■ Razón de uso

Se ha elegido este framework ante otros disponibles para Node.js por varias razones:

- El primer commit de esta herramienta fue 2 meses después de la creación de Node.js y la primera versión 1 año después por lo que tiene un gran recorrido y madurez.
- Su sencillez para manejar las rutas y las vistas en una solución web.
- Gran comunidad de desarrolladores en la que apoyarse.

### 8.1.3 Express-ejs-layouts

### 8.1.4 Nodemailer

Nodemailer es un módulo para Node.js que permite enviar emails. Este módulo esta securizado de manera que los mensajes que envía los hace de manera segura. Permite adjuntar elementos a los mensajes y no depende de ningún otro paquete, lo que le convierte en un paquete muy completo.

#### ■ Funcionamiento

Nodemailer es muy sencillo de usar, solo hay que definir el mensaje con los campos y archivos adjuntos que quieras añadir y también habrá que definir el transporter que sera el encargado de enviar el mensaje. En este caso no esta añadido la configuración del transporter.

```
var message = {
  from: 'sender@server.com',
  to: 'receiver@sender.com',
  subject: 'Message title',
  text: 'Plaintext version of the message',
  html: '<p>HTML version of the message</p>'
};
transporter.sendMail(data);
```

### ■ Razón de uso

Se ha utilizado este paquete para enviar los mails por su sencillez, el gran recorrido que tiene en esta materia, ya que fue creado en 2010 y por su completa documentación.

## 8.1.5 Mongojs

Mongojs es un paquete para Node.js que permite la conexión con bases de datos MongoDB. Este paquete intenta emular completamente la comunicación directa con la base de datos por lo que su API se asemeja mucho a la de MongoDB.

### ■ Funcionamiento

Mongojs es muy sencillo de usar, con la siguiente línea de código, conectaremos con una base de datos MongoDB que este en un servidor remoto y si añadimos la variable mycollection, conectaremos directamente con la colección dentro de la base de datos elegida.

```
var db = mongojs('example.com/mydb', ['mycollection'])
```

A continuación se muestra un ejemplo en el que se busca en la base de datos todos los documentos en los que name=Jhon. Esta función devuelve los documentos que cumplan la condición y posteriormente se pueden tratar dentro de la función.

```
db.mycollection.find({ 'Name': 'Jhon' })function (err, docs) {
  // Insert code here
};
```

### ■ Razón de uso

Se ha utilizado este paquete para conectar con la base de datos por la facilidad con la que conecta con la base de datos y porque emula lo máximo posible la API de mongoDB haciendo su uso muy fácil si ya has utilizado mongoDB con anterioridad.

## 8.1.6 Pdffiller

Pdffiller es un paquete para Node.js que permite rellenar los formularios de los PDF's con datos. Su uso se basa en recibir PDF's con formularios sin rellenar, unos datos y combinarlos de manera que la salida sea un PDF completo.

### ■ Razón de uso

Se ha utilizado este paquete por ser el mas utilizado por los usuarios entre las posibilidades.

## 8.2 MongoDB

MongoDB es la base de datos NoSQL líder y permite a las empresas ser más ágiles y escalables. Organizaciones de todos los tamaños están usando MongoDB para crear nuevos tipos de aplicaciones, mejorar la experiencia del cliente, acelerar el tiempo de comercialización y reducir costes.

Es una base de datos muy ágil por lo que permite cambiar los esquemas al cambiar los requisitos y a la vez proporciona las mismas funcionalidades que se esperan de una base de datos tradicional, manteniendo la velocidad en las búsquedas y siendo consistente.

Gracias a ser una base de datos orientada a documentos no hay que ajustarse a un esquema estándar ni obligar a todos los registros a tener la misma información. Esto nos da mucha flexibilidad a la hora de querer introducir nuevos datos o alterar los que ya tenemos.

La base de datos del proyecto ha sido poblada mediante JSON, un formato de texto para el intercambio de datos del que hablaremos a continuación.

### 8.2.1 Razón de uso

Se ha utilizado esta base de datos para el proyecto por las siguientes razones:

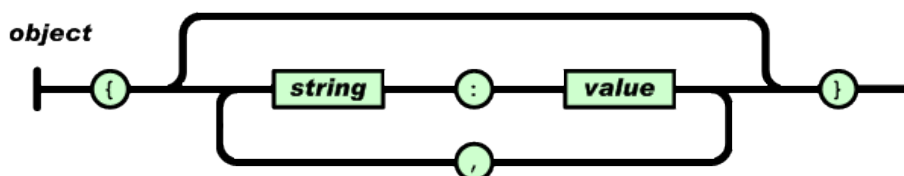
- Gracias a su flexibilidad permite alojar diferentes datos en cada documento, permitiendo así ir modificándolos a medida que se van transformando mientras se mantienen unificados.
- Gran soporte para proyectos realizados con Node.js y sobre todo con express, lo que hace su integración muy sencilla.
- El formato de intercambio de datos que se va a utilizar a lo largo de todo el proyecto será el JSON.
- Los límites que oferta la base de datos en cuanto a los documentos encajan con los requisitos del proyecto.
- Gratuita y open-source.

## 8.3 JSON

Json (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo.

### 8.3.1 Funcionamiento

Un objeto es un conjunto desordenado de pares nombre/valor. Los objetos comienzan con la llave (llave apertura) y terminan con la llave (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).





```

{
  "Nombre" : "Bolígrafo",
  "Tipo" : "Normal Bic",
  "Colores" : [
    "Azul",
    "Rojo",
    "Verde"
  ]
}

```

**Figura 8.2:** Ejemplo de un JSON

## 8.4 Sass (CSS3)

La definición de Sass y CSS3 integrados es la siguiente:

Sass es un metalenguaje de Hojas de Estilo en Cascada (CSS). Es un lenguaje de script que es traducido a CSS. Sass consiste en dos sintaxis. La sintaxis más reciente, SCSS, usa el formato de bloques como CSS. Éste usa llaves para denotar bloques de código y punto y coma (;) para separar las líneas dentro de un bloque.

CSS3 consiste en una serie de selectores y pseudo-selectores que agrupan las reglas que son aplicadas. Sass (en el amplio contexto de ambas sintaxis) extiende CSS proveyendo de varios mecanismos que están presentes en los lenguajes de programación tradicionales, particularmente lenguajes orientados a objetos, pero éste no está disponible para CSS3 como tal. Cuando SassScript se interpreta, éste crea bloques de reglas CSS para varios selectores que están definidos en el fichero SASS. El intérprete de SASS traduce SassScript en CSS. Alternativamente, Sass puede monitorear los ficheros .sass o .scss y convertirlos en un fichero .css de salida cada vez que el fichero .sass o .scss es guardado. Sass es simplemente azúcar sintáctica para escribir CSS.

Las ventajas que ofrece este metalenguaje son las siguientes:

- **Mixins:** Ofrece la posibilidad de crear bloques de estilo o mixins que se apliquen a mas de una etiqueta o clase, así, al compilar el código, el estilo se replica a todas las etiquetas y clases aligerando el trabajo.
- **Argumentos:** permite utilizar argumentos para unificar la definición de valores.
- **Herencia:** permite definir una herencia para que los hijos implementen el estilo de los padres.

### 8.4.1 Razón de uso

Se ha utilizado este metalenguaje en vez de su competidor más claro, less, por las siguientes razones:

- Sass genera un código más optimo en algunos de las características que ambos metalenguajes comparten.
- Sass tiene varios años más que Less por lo que le hace más robusto y con una mayor comunidad de desarrolladores.

Y se usa este metalenguaje en vez de usar CSS directamente por las siguientes razones:

- Permite un desarrollo más ágil de las hojas de estilo del proyecto.
- Innovación: permite aprender otra herramienta que agilice el desarrollo del CSS.

## 8.5 D3.js

D3.js (o simplemente D3 por las siglas de Data-Driven Documents) es una librería de JavaScript para producir, a partir de datos, infogramas dinámicos e interactivos en navegadores web. En contraste con

muchas otras librerías, D3.js permite tener control completo sobre el resultado visual final.

Gracias a esta librería se pueden crear gráficos interactivos en los que se unifiquen datos que anteriormente se mostrarían en varios gráficos diferentes, creando así una experiencia más inmersiva en cuanto a la visualización de los datos.

Las ventajas principales que ofrece esta librería son:

- **No añade carga a la página:** utiliza lenguajes ya existentes y que no suponen una carga mas para la página web en la que se aloja.
- **Amplias funcionalidades:** permite seleccionar diferentes elementos en una página web y alterarlos. También incluye transiciones para generar cambios visuales, muy similar a JQuery.
- **Asociación de datos:** los datos pueden dirigir la creación de los elementos permitiendo así a los datos gobernar la visualización y crear diferentes gráficos dependiendo del dataset introducido.

### 8.5.1 Funcionamiento

codigo scss

```

$blue: #3bbfce
$margin: 16px

.content-navigation
  border-color: $blue
  color: darken($blue, 9%)

.border
  padding: $margin/2
  margin: $margin/2
  border-color: $blue

.content-navigation {
  border-color: #3bbfce;
  color: #2b9eab;
}

.border {
  padding: 8px;
  margin: 8px;
  border-color: #3bbfce;
}

```

**Figura 8.3:** Scss y css compilado respectivamente

### 8.5.2 Razón de uso

Se ha utilizado esta librería de visualización frente a otras por las siguientes razones:

- Innovación: existe un creciente interés de desarrolladores por la herramienta y a su vez falta de programadores con conocimientos sobre ella.
- Gran integración con JSON y las herramientas utilizadas en el proyecto.
- Open-source y gratuita.

## 9. ESPECIFICACIÓN DEL DISEÑO

---

### 9.1 Visión general

Este capítulo tiene como objetivo describir la labor de diseño realizada para desarrollar el proyecto, así como las herramientas que se han utilizado para realizar estos diseños. En el siguiente listado se describen los diferentes diseños que se van a explicar en este capítulo:

- Diseño de la arquitectura: descripción de la arquitectura elegida para el proyecto.
- Diseño del servidor: descripción del diseño del servidor.
- Diseño de la página web: descripción del diseño de la página web.
- Diseño del widget: descripción del diseño del widget.
- Diseño de la base de datos: descripción del diseño de la base de datos

### 9.2 Herramientas utilizadas

Las herramientas que se han utilizado a la hora de diseñar el proyecto y sus elementos son las siguientes:

#### 9.2.1 Draw.io

Draw.io es una herramienta web que se utiliza para el desarrollo de diagramas de cualquier tipo, lo cual la hace una herramienta muy potente. Ofrece una gran cantidad de iconos y personalización de los mismos para realizar los diagramas lo más atractivos posibles. Entre sus características destaca la integración con sistemas de almacenamiento online como Google Drive o Dropbox y Github.

En lo respectivo a este proyecto, solo se ha utilizado para diseñar los diagramas: AÑADIR DIAGRAMAS.

#### 9.2.2 gomockingbird

Aplicación para mockups.

### 9.3 Diseño de la arquitectura

La arquitectura del proyecto (ver figura 9.1) se basa en un modelo en tres capas que gira en torno a el servidor Node.js. Estas tres capas permiten separar la parte visual, la lógica y la arquitectura de datos. Gracias a este modelo se puede centralizar la lógica del proyecto en la capa intermedia y abstraerla de los elementos externos a los que ofrece soporte.

En la capa de presentación se encuentran la página web y el widget. Estos dos elementos dan soporte a la parte visual del proyecto y están conectados con el servidor para que este les proporcione funcionalidad. En esta capa también se encuentra el sistema de visualización de datos del proyecto, este recibe los datos de la base de datos por medio del servidor que se los provee formateados.

En la capa intermedia, la de proceso, se encuentra el servidor que implementa la funcionalidad de toda la solución. Este alberga los widgets que los comercios online van a implementar en sus páginas web y les da la funcionalidad para permitir las donaciones. A la página web le provee del enrutado necesario para implementar sus funcionalidades principales, como la de obtener un certificado de donación o el contacto con el soporte del proyecto. En cuanto al sistema de visualización le otorga los datos, con el formato que necesita, para generar los gráficos deseados. Por último, esta conectada también con la capa de datos, en la que se encuentra la base de datos, con la cual conecta para enviar y pedir datos.

Finalmente, en la tercera capa, la de datos, se encuentra el sistema de base de datos. La base de datos, conectada con el servidor, almacena los datos que le llegan desde este y realiza las búsquedas y peticiones

que el servidor le pide. Todo esto lo hace mediante el lenguaje de intercambio de datos JSON del que hemos hablado anteriormente.

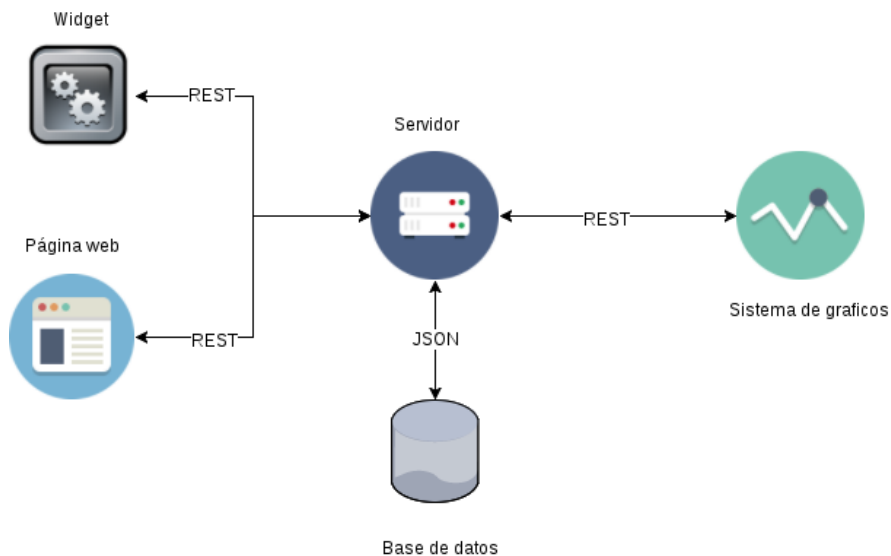


Figura 9.1: Arquitectura del proyecto

## 9.4 Diseño del servidor

El diseño del servidor se ha hecho siguiendo el modelo en el que el servidor Node.js con el apoyo de Express ofrece un API a las diferentes aplicaciones que están conectadas con él. En esta API el servidor ofrece toda la funcionalidad que estas aplicaciones tienen que desarrollar, por demanda de estas, y además puede ofrecer más funcionalidades que el proyecto pueda ofrecer a aplicaciones o elementos que puedan incorporarse en el futuro del proyecto.

Gracias a este modelo es muy sencillo añadir nuevas rutas y funcionalidades que las aplicaciones, nuevas o ya existentes, demanden del servidor. Además esta API mantendrá una conexión con la base de datos lo que permitirá el envío, consulta y almacenamiento de información de una manera muy veloz.

El servidor estará organizado de manera que en la carpeta *public* estarán alojados todos los archivos comunes incluyendo los scripts de JavaScript que sean necesarios para el desarrollo de las aplicaciones y las librerías correspondientes entre otras cosas.

Por último cabe destacar que en el diseño creado para el servidor todo gira alrededor del archivo en el que está desarrollada la API, aunque la creación del servidor en el puerto especificado y la configuración de red se encuentre en el archivo original del servidor.

HACER DIAGRAMA DEL SERVIDOR

## 9.5 Diseño de la página web

El diseño de la página web se ha realizado basándose en las convenciones de Bootstrap para que la web sea totalmente responsiva y visualizable desde la gran mayoría de los navegadores. Gracias al diseño con este framework se podrán añadir los componentes específicos que ofrece e implementar utilidades que sin este framework serían más complejas de desarrollar.

La página web se ha diseñado de manera que toda la información y las funcionalidades disponibles estén a la vista. Por eso se ha planteado una sola vista en la que por medio de algunos botones se desplieguen nuevas ventanas en las que poder añadir alguna funcionalidad o datos que de otra manera alargarían la

página de manera negativa.

La página se ha planteado muy minimalista en cuanto a contenido, siguiendo siempre el estilo y colores de Alboan. Otro de los planteamientos que se hace para la página es el de añadirle

## 9.6 Diseño del widget

El diseño del widget se ha hecho de la manera mas simple posible, utilizando solo tecnologías que todos los navegadores puedan implementar, estas tecnologías son HTML, CSS y JavaScript.

El widget



## BIBLIOGRAFÍA

---

- [1] Atom(Wikipedia) [online]. mayo 2017. URL: [https://es.wikipedia.org/wiki/Atom\\_\(editor\\_de\\_textos\)](https://es.wikipedia.org/wiki/Atom_(editor_de_textos)).
- [2] Equipo (Team) [online]. mayo 2017. URL: <https://proyectosagiles.org/equipo-team/>.
- [3] Facilitador (Scrum Master) [online]. mayo 2017. URL: <https://proyectosagiles.org/facilitador-scrum-master/>.
- [4] Alboan [online]. abril 2017. URL: <https://www.alboan.org/es>.
- [5] Node.js [online]. mayo 2017. URL: <https://nodejs.org/es/>.
- [6] Atom [online]. mayo 2017. URL: <https://atom.io/>.
- [7] MongoDB(Wikipedia) [online]. mayo 2017. URL: <https://es.wikipedia.org/wiki/MongoDB>.
- [8] MongoDB [online]. mayo 2017. URL: <https://www.mongodb.com/es>.
- [9] Git(Wikipedia) [online]. mayo 2017. URL: <https://es.wikipedia.org/wiki/Git>.
- [10] Git [online]. mayo 2017. URL: <https://git-scm.com/>.
- [11] Brackets(Wikipedia) [online]. mayo 2017. URL: [https://en.wikipedia.org/wiki/Brackets\\_\(text\\_editor\)](https://en.wikipedia.org/wiki/Brackets_(text_editor)).
- [12] Brackets [online]. mayo 2017. URL: <http://brackets.io/>.
- [13] Cliente (Product Owner) [online]. mayo 2017. URL: <https://proyectosagiles.org/cliente-product-owner/>.