



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Sitio Web para la gestión y publicitación de proyectos de cooperación

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Shaheer Shahzad

Tutor: Antonio Martí Campoy

Co-Tutora: Maria Teresa Gassò Matoses

2019-2020

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis padres y mis hermanos por todos sus sacrificios y por apoyarme siempre. Gracias a ellos, he sido capaz de llegar hasta aquí porque siempre han sido un apoyo muy importante para mí.

A mis tutores Antonio Martí Campoy y María Teresa Gassò Matoses por guiarme con sus consejos en esta etapa tan importante, por ayudarme a llevar a cabo este TFG, por compartir conmigo su conocimiento y experiencia y animarme siempre a seguir.

A todos los profesores con los que he compartido esta etapa académica por enseñarnos tanto, por compartir sus conocimientos, por su dedicación y por prepararnos para afrontar los retos.

A mis amigos/compañeros de la universidad por hacer que estos cuatro años hayan sido una experiencia inolvidable y por estar ahí en los momentos complicados.

A mi empresa Stylewest, a mis superiores y mis compañeros de trabajo por mostrarme siempre su apoyo y confianza y por ayudarme en todo lo posible para poder finalizar mis estudios sin problemas.

Resumen

En el presente trabajo se diseña y se desarrolla una aplicación accesible mediante navegador web cuya función principal es gestionar y publicar proyectos de una ONG y también permitir la realización de donaciones para ellos a través de distintas formas de pago.

Como resultado se obtiene una aplicación web que está compuesta por un *backend* y un *frontend*; dentro del *frontend* está el interfaz gráfico de usuario (GUI de sus siglas en inglés) en forma de página web. Este interfaz se divide en dos partes. La parte pública en la que se registran los usuarios, se publican los proyectos, se avisa de los eventos organizados por la ONG. Por otra parte el *backoffice* a través del cual los trabajadores de la ONG pueden gestionar los proyectos, los usuarios registrados, las donaciones recibidas, la información general y el contenido de la propia página web y otros aspectos de la aplicación. El *backend* se encarga de ejecutar un servidor Express utilizando distintos módulos de NodeJS, las rutas de las APIs, controladores y modelos de las entidades de la aplicación.

En este documento se dan detalles sobre el análisis, el diseño, las tecnologías y la implementación del proyecto. Además, se habla de los problemas afrontados y la investigación que se ha llevado a cabo para cumplir con los requisitos y alcanzar los objetivos planteados.

Palabras clave: aplicación web, ONG, proyectos, donaciones, usuarios

Abstract

In this project has been designed and developed an application accessible by web browsers whose main functions are publication and management of projects, users and donations of a non-governmental organization (NGO).

As a result a web application is obtained composed by a backend and a frontend; in the frontend there is the graphic user interace (GUI) in form of website. The GUI is divided in two parts. On the public side users can register on the website, the projects can be published and the events organized by the NGO can be notified. On the other hand there is a backoffice where the NGO's workers are able to manage the projects, the



registered users, received donations, the general information and the content of the website including other aspects of the application. The backend deploys an Express server using different NodeJS modules, API routes, controllers and models of the application entities

In this document is presented the analysis, design, technology stack and the development of the project. In addition, the problems that have been faced and all the study that has been done to accomplish the requirements and achieve the planned goals will also be discussed.

Palabras clave: web application, NGO, projects, donations, users

Índice de contenidos

1.	Introducción	21
1.1.	Motivación.....	21
1.2.	Objetivos.....	22
1.3.	Plan de trabajo	22
1.4.	Estructura de la memoria.....	23
2.	Metodología	25
3.	Especificación de requisitos.....	27
3.1.	Necesidad de especificar los requisitos	27
3.2.	Resultado de la fase	27
4.	Análisis	31
4.1.	UML.....	31
4.1.1.	Diagrama de contexto.....	31
4.1.2.	Modelo de dominio	32
4.1.3.	Diagrama de clases	33
4.1.4.	Diagrama de casos de uso	36
4.1.4.1.	Frontoffice	36
4.1.4.2.	Backoffice.....	42
4.1.4.3.	Modelado por actores	67
4.1.4.3.1.	Actor Anónimo	67
4.1.4.3.2.	Actor Registrado.....	68
4.1.4.3.3.	Actor Donante	69
4.1.4.3.4.	Actor Voluntario.....	70
4.1.4.3.5.	Actor Administrador	71
4.1.4.3.6.	Actor Sistema	73
5.	Diseño.....	75
5.1.	Diseño arquitectónico MVC.....	75
5.1.1.	Elección de MVC	76
5.2.	Interfaz gráfica de la aplicación web	76
5.2.1.	<i>Frontend</i>	77
5.2.2.	<i>Backoffice</i>	86
6.	Implementación.....	119



6.1.	Tecnologías	119
6.1.1.	Elección del <i>stack MEAN</i>	120
6.1.2.	<i>MEAN</i>	122
6.1.2.1.	MongoDB.....	122
6.1.2.2.	Express.....	123
6.1.2.3.	Angular	123
6.1.2.4.	NodeJS.....	123
6.1.3.	Funcionamiento de <i>MEAN stack</i>	124
6.1.4.	Módulos NodeJS	125
6.1.5.	Módulos Angular.....	126
6.1.6.	Entorno de desarrollo.....	128
6.1.7.	Herramientas y utilidades	128
6.2.	Desarrollo de la solución	131
6.2.1.	Servidor (Express)	131
6.2.1.1	Rutas (<i>Routes</i>).....	132
6.2.1.2	Modelos (<i>Models</i>)	132
6.2.1.3	Controladores (<i>Controllers</i>).....	133
6.2.2	Cliente (Angular).....	133
6.2.2.1	Componentes (<i>Components</i>)	135
6.2.2.2	Modelos (<i>Models</i>)	135
6.2.2.3	Páginas (<i>Pages</i>).....	136
6.2.2.4	Servicios (<i>Services</i>).....	137
6.2.3	Seguridad.....	139
7.	Pruebas.....	141
7.1	Pruebas funcionales.....	141
7.2	<i>HTML W3C Validator</i>	159
8.	Despliegue y publicación.....	161
8.1.	Heroku.....	162
8.1.1.	Despliegue con Heroku	162
8.1.1.1.	Línea de comandos	163
8.1.1.2.	Panel de control.....	165
8.1.1.3.	GitHub	167
8.1.1.4.	Aplicar cambios en el dominio	167
8.2.	MongoDB Atlas.....	168
8.2.1.	Creación de la base de datos en la nube	168
8.2.2.	Conectar la base de datos con la aplicación	170

8.3.	Prestaciones de los servicios	171
9.	Presupuesto.....	175
10.	Conclusiones	177
10.1.	Trabajo futuro	177
10.2.	Valoración.....	178
11.	Anexos	181
11.1.	Configuración de Stripe.....	181
11.1.1.	Activación de la cuenta	181
11.1.2.	Desactivar los datos de prueba	186
11.1.3.	Conectar Stripe con la aplicación.....	187
12.	Referencias bibliográficas	189



Índice de imágenes

Imagen 1. Plan de trabajo	23
Imagen 2. Proceso de desarrollo de <i>software</i>	25
Imagen 3. Logotipo de UML.....	31
Imagen 4. Diagrama de contexto	32
Imagen 5. Modelo de dominio	33
Imagen 6. Diagrama de clases.....	35
Imagen 7. Diagrama de casos de uso para el actor Anonimo.....	67
Imagen 8. Diagrama de casos de uso para el actor Registrado.....	68
Imagen 9. Diagrama de casos de uso para el actor Donante.....	69
Imagen 10. Diagrama de casos de uso para el actor Voluntario	70
Imagen 11. Diagrama de casos de uso para el actor Administrador (1).....	71
Imagen 12. Diagrama de casos de uso para el actor Administrador (2).....	72
Imagen 13. Diagrama de casos de uso para el actor Sistema	73
Imagen 14. Patrón de diseño MVC.....	76
Imagen 15. Logotipo de <i>Balsamiq Mockups 3</i>	77
Imagen 16. Página de inicio.....	79
Imagen 17. Página para registrarse.....	79
Imagen 18. Página para iniciar sesión	80
Imagen 19. Página de perfil	81
Imagen 20. Página de detalles del proyecto	82
Imagen 21. Página para ver los eventos.....	83
Imagen 22. Página para ver las noticias	84
Imagen 23. Página de donación	85
Imagen 24. Página de contacto	86
Imagen 25. Página principal del backoffice.....	87
Imagen 26. Página para listar los usuarios	88
Imagen 27. Formulario para añadir usuario	89
Imagen 28. Página para ver los detalles del usuario	90
Imagen 29. Formulario para editar un usuario	91
Imagen 30. Confirmación para borrar un usuario	92
Imagen 31. Página para listar los proyectos	93
Imagen 32. Formulario para añadir un proyecto.....	94
Imagen 33. Página para ver los detalles de un proyecto	95
Imagen 34. Formulario para editar un proyecto.....	96
Imagen 35. Confirmación para borrar un proyecto.....	97
Imagen 36. Página para añadir/quitar voluntarios	98
Imagen 37. Página para listar las donaciones	99
Imagen 38. Formulario para añadir una donación	100
Imagen 39. Página para ver los detalles de una donación	101
Imagen 40. Formulario para editar una donación	102
Imagen 41. Confirmación para borrar una donación	103
Imagen 42. Página para listar los eventos	104
Imagen 43. Formulario para añadir un evento	105



Imagen 44. Página para ver los detalles de un evento.....	106
Imagen 45. Formulario para editar un evento	107
Imagen 46. Confirmación para borrar un evento	108
Imagen 47. Página para listar las noticias	109
Imagen 48. Formulario para añadir una noticia.....	110
Imagen 49. Página para ver los detalles de una noticia	111
Imagen 50. Formulario para editar una noticia.....	112
Imagen 51. Confirmación para borrar una noticia.....	113
Imagen 52. Página para actualizar el SEO.....	114
Imagen 53. Página para listar los suscriptores	115
Imagen 54. Confirmación para borrar un suscriptor	116
Imagen 55. Página para listar los ficheros.....	117
Imagen 56. Confirmación para borrar un fichero.....	118
Imagen 57. MEAN stack	119
Imagen 58. LAMP stack.....	120
Imagen 59. MERN stack.....	120
Imagen 60. Logotipo de MongoDB.....	122
Imagen 61. Logotipo de Express	123
Imagen 62. Logotipo de Angular.....	123
Imagen 63. Logotipo de NodeJS	124
Imagen 64. Arquitectura MEAN	125
Imagen 65. Importación módulos Angular (1).....	127
Imagen 66. Importación módulos Angular (2)	127
Imagen 67. Cambio de tema en materialize-css.....	128
Imagen 68. Logotipo de Visual Studio Code	128
Imagen 69. Logotipo de Postman	129
Imagen 70. Logotipo de UMLet	129
Imagen 71. Logotipo de Git.....	129
Imagen 72. Logotipo de Leaflet.....	129
Imagen 73. Logotipo de Open Street Map.....	130
Imagen 74. Logotipo de Stripe	130
Imagen 75. Logotipo de Gmail	130
Imagen 76. Logotipo de Robo 3T.....	130
Imagen 77. Logotipo de MaterializeCSS	131
Imagen 78. Estructura global del servidor	132
Imagen 79. Contenido de la carpeta routes	132
Imagen 80. Contenido de la carpeta models	133
Imagen 81. Contenido de la carpeta controllers	133
Imagen 82. Estructura global del frontend	134
Imagen 83. Contenido de la carpeta components	135
Imagen 84. Contenido de la carpeta models	136
Imagen 85. Comando de Angular para generar una clase	136
Imagen 86. Contenido de la carpeta pages.....	137
Imagen 87. Comando de Angular para generar un componente.....	137
Imagen 88. Contenido de la carpeta services.....	138
Imagen 89. Comando de Angular para generar un servicio.....	138
Imagen 90. Validación de campos vacíos en "Registrarse"	141
Imagen 91. Correo existente en "Registrarse"	142

Imagen 92. Cambio menú principal al iniciar sesión	142
Imagen 93. Correo de bienvenida	142
Imagen 94. Correo de suscripción al boletín de noticias	142
Imagen 95. Detección de datos incorrectos en "Iniciar sesión"	143
Imagen 96. Mensaje al fallar la validación de credenciales	143
Imagen 97. Opción "Salir" al iniciar sesión	144
Imagen 98. Estado del menú principal al cerrar sesión	144
Imagen 99. Página del perfil.....	144
Imagen 100. Ventana para solicitar correo de recuperación	145
Imagen 101. Correo de recuperación de contraseña.....	145
Imagen 102. Formulario para actualizar la contraseña.....	145
Imagen 103. Detección del formulario del boletín.....	146
Imagen 104. Mensaje al dar de alta en el boletín.....	146
Imagen 105. Lista de proyectos en la página principal.....	147
Imagen 106. Página de detalles del proyecto	147
Imagen 107. Lista de los eventos.....	148
Imagen 108. Lista de las noticias	148
Imagen 109. Validación del formulario de contacto.....	148
Imagen 110. Requisito de aceptar las políticas de privacidad.....	149
Imagen 111. Mensaje del formulario de contacto.....	149
Imagen 112. Lista de donaciones del usuario	150
Imagen 113. Opciones de menú principal para trabajadores	150
Imagen 114. Página del <i>backoffice</i>	151
Imagen 115. Validación del formulario para crear proyecto	151
Imagen 116. Mensaje de éxito al crear proyecto.....	152
Imagen 117. Ventana con los detalles del proyecto.....	152
Imagen 118. Listado de ficheros del proyecto	153
Imagen 119. Formulario para editar proyecto.....	153
Imagen 120. Validación del formulario al editar proyecto.....	154
Imagen 121. Opciones de los ficheros al editar proyecto.....	154
Imagen 122. Subida de fichero al editar proyecto.....	155
Imagen 123. Fichero subido a la carpeta del servidor	155
Imagen 124. Lista actualizada al subir fichero	155
Imagen 125. Ventana de confirmación para borrar proyecto	156
Imagen 126. Fichero borrado en la carpeta del servidor	156
Imagen 127. Proyecto borrado de la lista de proyectos	156
Imagen 128. Ventana para añadir/quitar voluntarios.....	157
Imagen 129. Mensaje y cambio de icono al añadir voluntario.....	157
Imagen 130. Mensaje y cambio de icono al quitar voluntario	157
Imagen 131. Formulario de datos web antes de pulsar el botón de actualizar.....	158
Imagen 132. Mensaje al actualizar los datos web	158
Imagen 133. Título actualizado al pulsar el botón de actualización	158
Imagen 134. Ventana de confirmación al borrar suscriptor del boletín.....	159
Imagen 135. Mensaje al borrar el suscriptor del boletín	159
Imagen 136. Análisis <i>HTML w3C Validator</i>	160
Imagen 137. Logotipo de Heroku.....	162
Imagen 138. Comando para iniciar sesión en Heroku	163
Imagen 139. Comandos para inicializar repositorio Git.....	164



Imagen 140. Comando para crea una aplicación en Heroku	164
Imagen 141. Comando para asignar repositorio remoto	164
Imagen 142. Comando para lanzar el despliegue.....	165
Imagen 143. Comando para desplegar una rama específica	165
Imagen 144. Opciones para crear una aplicación nueva	165
Imagen 145. Formulario para la nueva aplicación.....	166
Imagen 146. Comandos para desplegar la aplicación.....	166
Imagen 147. Botón para acceder a la aplicación	167
Imagen 148. Despliegue con GitHub	167
Imagen 149. Comandos para desplegar cambios en Heroku	168
Imagen 150. Crear una base de datos nueva	168
Imagen 151. Elegir el servicio de la nube para la base de datos	169
Imagen 152. Nombre de la base de datos	169
Imagen 153. Opciones de la base de datos	170
Imagen 154. Opciones para conectar la base de datos	171
Imagen 155. URL para conectar la base de datos con la aplicación.....	171
Imagen 156. Presupuesto estimado del proyecto.....	176
Imagen 157. Menú principal de Stripe.....	182
Imagen 158. Formulario para añadir la dirección de la entidad.....	182
Imagen 159. Formulario para los datos legales de la entidad	183
Imagen 160. Formulario para los datos del representante legal	184
Imagen 161. Opción de añadir director de la entidad.....	184
Imagen 162. Información para los extractos bancarios.....	185
Imagen 163. Formulario para añadir información bancaria.....	185
Imagen 164. Activación de autenticación en dos pasos.....	186
Imagen 165. Opción para activar/desactivar datos de prueba.....	186
Imagen 166. Páginas de las claves de API	187
Imagen 167. Uso de la clave secreta en el proyecto	187

Índice de tablas

Tabla 1. Caso de uso: Registrarse	36
Tabla 2. Caso de uso: Iniciar sesión	37
Tabla 3. Caso de uso: Cerrar sesión	37
Tabla 4. Caso de uso: Ver perfil.....	38
Tabla 5. Caso de uso: Resetear contraseña.....	38
Tabla 6. Caso de uso: Ver perfil.....	39
Tabla 7. Caso de uso: Ver eventos	39
Tabla 8. Caso de uso: Ver noticias	39
Tabla 9. Caso de uso: Donar	40
Tabla 10. Caso de uso: Contactar	41
Tabla 11. Caso de uso: Suscribirse al boletín de noticias	41
Tabla 12. Caso de uso: Ver donaciones realizadas	42
Tabla 13. Caso de uso: Ver documentos del proyecto.....	42
Tabla 14. Caso de uso: Listar usuarios.....	43
Tabla 15. Caso de uso: Ver usuario	43
Tabla 16. Caso de uso: Crear usuario	44
Tabla 17. Caso de uso: Editar usuario	45
Tabla 18. Caso de uso: Borrar usuario	45
Tabla 19. Caso de uso: Listar proyectos.....	46
Tabla 20. Caso de uso: Ver proyecto.....	46
Tabla 21. Caso de uso: Crear proyecto	47
Tabla 22. Caso de uso: Editar proyecto	48
Tabla 23. Caso de uso: Borrar proyecto.....	48
Tabla 24. Caso de uso: Añadir voluntario	49
Tabla 25. Caso de uso: Quitar voluntario	50
Tabla 26. Caso de uso: Subir fichero.....	50
Tabla 27. Caso de uso: Descargar fichero	51
Tabla 28. Caso de uso: Borrar fichero	52
Tabla 29. Caso de uso: Borrar fichero del servidor.....	52
Tabla 30. Caso de uso: Listar donaciones	53
Tabla 31. Caso de uso: Ver donación.....	53
Tabla 32. Caso de uso: Crear donación.....	54
Tabla 33. Caso de uso: Editar donación	55
Tabla 34. Caso de uso: Borrar donación.....	55
Tabla 35. Caso de uso: Listar eventos	56
Tabla 36. Caso de uso: Ver evento	56
Tabla 37. Caso de uso: Crear evento	57
Tabla 38. Caso de uso: Editar evento.....	58
Tabla 39. Caso de uso: Borrar evento	58
Tabla 40. Caso de uso: Listar noticias	59
Tabla 41. Caso de uso: Ver noticia.....	59
Tabla 42. Caso de uso: Crear noticia.....	60
Tabla 43. Caso de uso: Editar noticia	60



Tabla 44. Caso de uso: Borrar noticia	61
Tabla 45. Caso de uso: Cambiar título	62
Tabla 46. Caso de uso: Cambiar meta-título	62
Tabla 47. Caso de uso: Cambiar meta-descripción	63
Tabla 48. Caso de uso: Listar suscriptores	64
Tabla 49. Caso de uso: Borrar suscriptor	64
Tabla 50. Caso de uso: Listar ficheros	65
Tabla 51. Caso de uso: Descargar fichero	65
Tabla 52. Caso de uso: Borrar fichero.....	66
Tabla 53. Caso de uso: Borrar fichero del servidor	66

Abreviaturas y acrónimos

- **Backend:** la parte encargada de lanzar el servidor de la aplicación y conectar con la base de datos.
- **Backoffice:** Es la parte privada y que sirve para administrar la aplicación.
- **Commit:** es la acción para confirmar un conjunto de cambios provisionales de forma permanente en un repositorio de control de versiones.
- **CSS:** *Cascading Style Sheets* de sus siglas en inglés, es un lenguaje de diseño gráfico para aplicar estilos a las páginas web.
- **DB:** *Database*
- **Framework:** estructura conceptual y tecnológica de soporte definido que puede servir de base para la organización y desarrollo de software.
- **Frontend (GUI):** la interfaz gráfica de la aplicación disponible para los usuarios.
- **Frontoffice:** Es la parte pública de la aplicación.
- **GB:** *Gigabyte*
- **Git:** es un software de control de versiones.
- **GUI:** *Graphic User Interface*
- **Hardware:** componentes informáticos tangibles para ejecutar distintos programas o algoritmos.
- **Hosting:** es un servicio en línea que te permite publicar un sitio o aplicación web en Internet.
- **HTML:** *HyperText Markup Language* de sus siglas en inglés, es un lenguaje de marcado que se utiliza para construir páginas web.
- **HTTP:** *Hypertext Transfer Protocol* de sus siglas en inglés, es el protocolo de comunicación que permite las transferencias de información en la *World Wide Web*.
- **JavaScript:** es un lenguaje de programación.
- **JS:** *JavaScript*
- **JSON:** *JavaScript Object Notation* de sus siglas en inglés, es un formato para representar los datos en pares clave-valor.
- **LAMP:** Linux OS, Apache web server, MySQL y PHP
- **MB:** *Megabyte*
- **MEAN:** MongoDB, Express, Angular y NodeJS



- **MVC:** *Model View Controller* de sus siglas en inglés, es una propuesta de arquitectura del *software* utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta.
- **Newsletter:** es el término en inglés que se utiliza para referirse al boletín de anuncios.
- **NodeJS:** es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación *JavaScript*.
- **NPM:** *Node Package Manager*
- **ONG:** Organización No Gubernamental
- **Online:** algo que está conectado o a alguien que está haciendo uso de una red (generalmente, Internet).
- **OSM:** *Open Street Map* de sus siglas en inglés, es una librería de código abierto para la integración de mapas.
- **PaaS:** *Platform as a Service*
- **Parsear:** procesar las entradas de una petición para que sean entendibles por el resto de componentes integrados.
- **PDF:** *Portable Document Format*
- **RAM:** *Random Access Memory*
- **REST:** *Representational State Transfer* de sus siglas en inglés, es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en formatos como XML o JSON.
- **SEO:** *Search Engine Optimization*
- **SMS:** *Short Message Service*
- **Software:** conjunto de componentes informáticos intangibles que cooperan entre sí para ofrecer un servicio a los usuarios.
- **SSL:** *Secure Sockets Layer*
- **Stack:** conjunto o lista de algo.
- **TFG:** Trabajo Final de Grado
- **Token:** cadena encriptada para ciertas verificaciones.
- **TS:** *TypeScript*
- **TypeScript:** es un lenguaje de programación y superconjunto de *JavaScript*.
- **UML:** *Unified Modeling Language* de sus siglas en inglés, es un lenguaje estandarizado de modelado.
- **URI:** *Uniform Resource Identifier*

- **URL:** *Uniform Resource Locator*
- **VPS:** *Virtual Private Server*
- **W3C:** *World Wide Web Consortium*
- **XML:** *Extensible Markup Language* de sus siglas en inglés, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos.

1. Introducción

En este trabajo de fin de grado de ingeniería de informática se desarrolla y se despliega un sitio web para publicar y gestionar proyectos de cooperación de una ONG y permitir la realización y gestión de la recepción de donaciones económicas.

Para desarrollar esta aplicación web se ha utilizado la *suite* llamada *stack MEAN*. *MEAN* es un conjunto de diferentes tecnologías de desarrollo web basadas en programación con el lenguaje *JavaScript* que son: MongoDB (M)¹, Express (E)², Angular (A)³ y NodeJS (N)⁴.

En el presente TFG se exponen los diseños y esquemas conceptuales para el desarrollo de la aplicación web, las diferentes entidades del proyecto y sus relaciones, las investigaciones realizadas para escoger el *stack MEAN* comparándolo con tecnologías alternativas utilizadas en el mundo laboral, se muestra la configuración y puesta en marcha tanto de la parte del servidor como de la parte pública de la página web, el diseño gráfico de la aplicación, las pruebas realizadas para validar todas las funcionalidades y las conclusiones.

1.1. Motivación

Desde que el sacerdote Vicente Berenguer llegó a Mozambique en el año 1967, ha estado realizando diferentes trabajos solidarios para ayudar a las personas más desfavorecidas. Al ver que la mayoría de los niños no tenían acceso a la educación, Vicente Berenguer empezó a trabajar para cambiar esta situación y creó colegios, guarderías, institutos, residencias, etc. para apoyar a todos esos niños.

Ahora Vicente Berenguer ya se ha retirado y actualmente vive en España y no puede gestionar ni supervisar presencialmente los proyectos y fondos necesarios para llevar a cabo las acciones que planifica su ONG. Por tanto, hace falta encontrar una solución a este problema y para ello se va a realizar este TFG para intentar ayudar a la

¹ <https://www.mongodb.com/es>

² <https://expressjs.com/>

³ <https://angular.io/>

⁴ <https://nodejs.org/en/>

organización a seguir con su trabajo gestionando los proyectos y recaudando donaciones económicas a través de una aplicación web.

1.2. Objetivos

El objetivo principal de la aplicación es permitir la gestión y publicitación de distintos proyectos de cooperación que lanza la ONG o en las que ya está trabajando y recaudación de donaciones a través de distintos métodos de pago.

Entre otros objetivos, se tiene:

- Sistema de autenticación y registro en la aplicación para todos los usuarios y trabajadores.
- Suscripción al boletín de noticias para que los usuarios estén al tanto de las novedades de la ONG.
- Acceso exclusivo para los donantes al área privada de los proyectos para así obtener más información respecto a los gastos y gestiones.
- Permitir un mantenimiento eficiente de la aplicación creando, editando, eliminando y consultando diferentes recursos (usuarios, proyectos, donaciones, avisos, etc.) a medida que se necesiten o dejen de utilizarse.
- Despliegue de la aplicación en un *hosting*.
- Comunicar diferentes eventos a los usuarios mediante la página web.
- Uso del SEO en la página web.
- Creación y publicación de eventos y noticias.
- Promover y facilitar las aportaciones económicas.

Con esta aplicación se pretende facilitar el proceso de recaudación de fondos y ampliar el rango de visibilidad de la ONG y mejorar la comunicación con todo tipo de usuarios.

1.3. Plan de trabajo

A continuación se presenta el plan de trabajo previsto para este TFG. El plan se elaboró considerando que el alumno trabaja a jornada parcial y la dedicación al proyecto sería de 4 horas diarias. El total de horas dedicadas es **320 horas**. La Imagen 1 muestra la planificación temporal del trabajo:

Tareas	Duración	Semanas															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Requisitos	10 días	■	■	■													
Análisis	7 días			■	■	■											
Diseño	14 días						■	■									
Implementación	40 días							■	■	■	■	■	■	■	■		
Pruebas	7 días									■				■	■		
Despliegue	10 días														■	■	■
Memoria	15 días										■	■	■	■	■	■	■

Imagen 1. Plan de trabajo

1.4. Estructura de la memoria

El presente documento está dividido en diferentes capítulos para exponer los detalles de las distintas fases del TFG desarrollado. La memoria se estructura de la siguiente forma:

- **Capítulo 1:** se introduce el trabajo desarrollado, se detallan los problemas que se quieren solucionar con el proyecto, se listan los objetivos y se muestra el plan de trabajo.
- **Capítulo 2:** se explica la metodología utilizada para llevar a cabo el desarrollo de la aplicación y se explican las fases por las que está compuesta dicha metodología.
- **Capítulo 3:** se profundiza en la fase de especificación de requisitos, se explica la importancia de trabajar esta fase y se muestra el resultado de aplicar dicha fase al presente proyecto.
- **Capítulo 4:** se muestran los distintos diagramas elaborados para el proyecto (de contexto, de clase y de casos de uso) y se listan las funcionalidades con todos los detalles.
- **Capítulo 5:** se enseña el diseño arquitectónico a seguir para la aplicación, se comentan sus ventajas y se muestran las maquetas de la interfaz gráfica de la aplicación web.
- **Capítulo 6:** se presentan las distintas tecnologías de desarrollo web, se detallan las tecnologías elegidas para la elaboración del proyecto, se explican sus ventajas y desventajas, se comenta el funcionamiento de dichas tecnologías, se listan las librerías y otras herramientas utilizadas para la implementación de la aplicación y se muestra el resultado del desarrollo.

- **Capítulo 7:** se explican las pruebas realizadas en la aplicación y se muestran los resultados de las mismas.
- **Capítulo 8:** se comentan las distintas opciones para desplegar la aplicación web en un dominio público, se presenta la opción elegida, se justifica su elección, se muestra el proceso de despliegue con total detalle (el despliegue de la página web y la conexión con la base de datos) junto al proceso para aplicar futuros cambios al dominio.
- **Capítulo 9:** se muestra el presupuesto estimado del proyecto detallando los distintos factores que influyen en el cálculo.
- **Capítulo 10:** se presentan las conclusiones obtenidas tras el desarrollo de la aplicación, se comentan algunas mejoras o funcionalidades nuevas que se podrían incorporar a la aplicación y se valora la experiencia obtenida.

2. Metodología

Para el desarrollo de esta aplicación se ha seguido el proceso de *software* [1] llamado desarrollo en cascada que consiste en las siguientes fases: especificación de requisitos, el análisis, el diseño, la implementación, pruebas y despliegue. Seguir un proceso definido es muy importante para la realización de cualquier tipo de proyecto porque proporciona cierto tipo de orden y pasos a seguir para el desarrollo del trabajo. En la Imagen 2 se muestra el proceso seguido en este TFG:

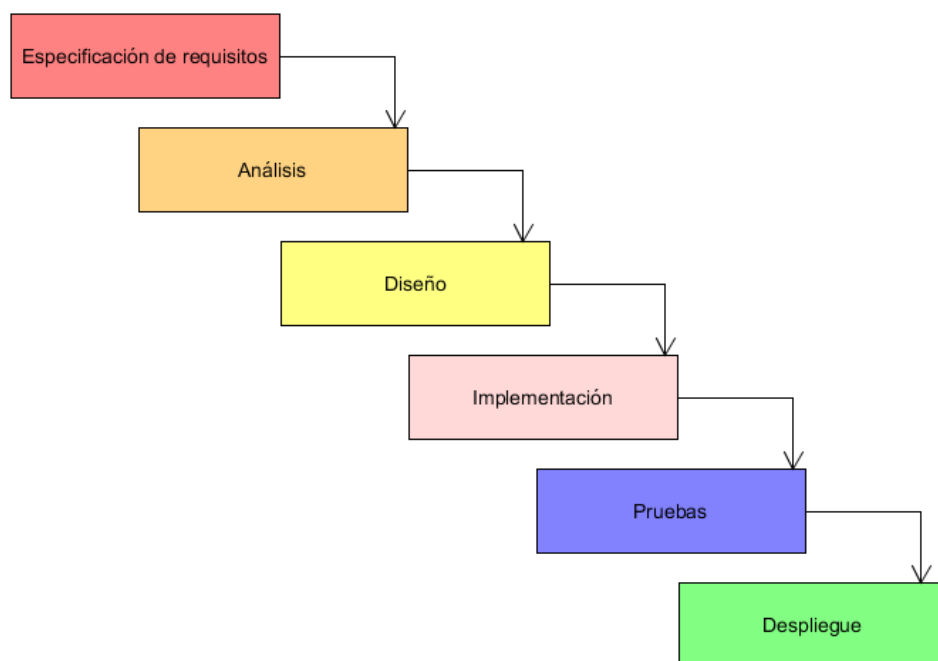


Imagen 2. Proceso de desarrollo de *software*

A continuación se describen los pasos que conforman el proceso de desarrollo:

- **Especificación de requisitos:** en esta fase se recogen los requisitos necesarios para el desarrollo del *software* mediante reuniones con el cliente para documentar lo que se espera del proyecto. En este trabajo se organizó una reunión con el tutor y la cotutora y se hizo un documento con los objetivos esperados y las necesidades de la aplicación; dicho documento se refinó en reuniones posteriores, hasta alcanzar el detalle mostrado en el capítulo 5 de este documento.

- **Análisis:** en esta fase se estructuran las ideas documentadas en la fase de requisitos y se empieza a trabajar la arquitectura del proyecto. Se realizan diferentes diagramas para planificar la estructura y el desarrollo de las funcionalidades.
- **Diseño:** una vez realizado el análisis del proyecto, se procede a realizar el diseño arquitectónico, el diseño organizacional y el diseño gráfico de la aplicación. Este paso ayuda a especificar el funcionamiento interno del *software*, la separación de los componentes, la estructura de la base de datos y el diseño de la interfaz gráfica de acuerdo a las funcionalidades de la aplicación.
- **Implementación:** esta fase consiste en programar las funcionalidades documentadas y planificadas con la utilización de diferentes tecnologías y herramientas.
- **Pruebas:** una vez finalizada la aplicación, en esta fase se empiezan a probar las funcionalidades desarrolladas para asegurar el correcto funcionamiento del *software*.
- **Despliegue:** en esta fase se procede a publicar o desplegar el proyecto para que todos los usuarios puedan acceder a la aplicación.

Este proceso se puede seguir en cascada o de forma iterativa. El desarrollo en cascada o también llamado secuencial consiste en terminar cada fase del proceso para avanzar al siguiente. Al final de cada etapa se realiza una revisión final que determina si el proyecto está preparado para avanzar a la siguiente fase. Por otro lado, está el desarrollo iterativo o incremental el cual consiste en aplicar todo el proceso de *software* sobre un conjunto reducido de tareas en etapas repetitivas (iteraciones) e ir incrementando dicho conjunto conforme va avanzando el proyecto.

Dependiendo del proyecto y de cada equipo se sigue una opción u otra. Como se ha mencionado anteriormente, en este trabajo se ha usado el proceso en cascada.

3. Especificación de requisitos

La especificación de requisitos es una de las fases más importantes en el proceso de desarrollo de *software*. Es el primer paso para empezar a desarrollar un producto *software* y cuanto más se trabaja esta fase, mejor es el resultado. Los fallos o errores que se encuentren en esta fase serán los más fáciles de corregir y en cuanto más se avance de fase, más difícil es corregir un fallo y el coste también es mayor.

3.1. Necesidad de especificar los requisitos

Como bien se ha comentado, antes de empezar a desarrollar un proyecto es muy importante que se hable con los usuarios que lo van a utilizar para recoger los máximos detalles para el *software*, saber a qué tipo de problemas se enfrentan, cuáles son las tareas que más se utilizan, cuáles son las tareas que menos se utilizan, en qué tipo de funcionalidades se sienten más incómodos, en qué tipo de entorno se va trabajar con el *software*, cuál es el nivel de manejo de los usuarios objetivos del producto, etc. Otro motivo por el cual es de gran importancia la especificación de requisitos es que ayuda a entender de una muy buena forma la necesidad del cliente o de los usuarios porque es la fase donde más se interacciona con los usuarios.

El objetivo es que al finalizar esta fase se obtenga un documento donde se defina todo lo que se quiere conseguir con el *software* teniendo en cuenta las necesidades y las exigencias del cliente. Es muy importante saber lo que tiene que hacer el *software*, cómo lo tiene que hacer y también lo que no tiene que hacer. Si el análisis de esta fase se realiza correctamente, el producto resultante también será de mejor calidad ya que ayuda a detectar con antelación muchos errores que en caso de no haberlos detectado en las fases iniciales del proceso de *software*, costarían bastante más debido a que se requerirá de un tiempo mayor para la corrección y el *software* será de peor calidad.

3.2. Resultado de la fase

En esta sección se va a analizar el resultado que se obtuvo al aplicar esta fase a este proyecto. Debido a la situación extraordinaria del COVID-19 la especificación de

requisitos y todo el proceso de desarrollo en general se vio afectado ya que en lugar de tener reuniones físicas para resolver dudas o consultar diferentes aspectos de la aplicación se realizaron videoconferencias lo cual ha sido una nueva experiencia de aprendizaje a lo largo de todo el proceso de realización del presente TFG.

Como resultado de esta fase, se obtuvieron los siguientes requisitos:

- La aplicación se debe dividir en *frontoffice* y *backoffice* para separar la parte pública de la aplicación que será accesible por todos los usuarios de la red de la parte privada para administrar la página web.
- La aplicación web debe dotar de un sistema de roles para controlar el acceso de los usuarios a ciertas funcionalidades. Se especificaron cinco roles diferentes:
 1. **Administrador:** control total de la aplicación. Puede gestionar y administrar todos los aspectos de la aplicación como pueden ser los usuarios, los proyectos, las donaciones, los eventos, las noticias, el SEO, el boletín de noticias y los ficheros.
 2. **Voluntario:** acceso a la administración pero con ciertas restricciones. Permiso para ver solamente la información del proyecto en el que colabore y subir los ficheros al mismo si fuera preciso.
 3. **Donante:** sin acceso al panel de administración. Podrá acceder a su perfil en la parte pública de la aplicación y también podrá ver información detallada del proyecto o de los proyectos en los que haya colaborado mediante alguna donación como pueden ser las facturas, las imágenes o el equipo que trabaja en el proyecto.
 4. **Registrado:** un usuario que se ha dado de alta en la página web pero no ha realizado ninguna donación hasta el momento. Este usuario podrá acceder a su perfil, podrá ver toda la información disponible en la parte pública de la aplicación pero sin llegar a ver los detalles privados de los proyectos.
 5. **Anónimo:** este rol podrá acceder a la página web pública y tendrá la opción de registrarse si así lo desea.
- Los proyectos de la ONG se pueden crear de forma estática o dinámica. Es decir, si se optara por crear proyectos estáticos, el usuario administrador seleccionará los tipos predefinidos para los proyectos y se creará un proyecto nuevo con un título, descripción y más información relacionada al proyecto que se haya especificado en el código de desarrollo sin la posibilidad de ser modificada sin la intervención del desarrollador. En cambio, si se opta por proyectos dinámicos opción implementada en la aplicación web, el administrador podrá añadir de

forma personalizable toda la información del proyecto, la cual será editable a posteriori. Los proyectos dinámicos, deberán estar compuestos por las siguientes secciones:

1. Un organigrama para ver el equipo involucrado en el proyecto.
 2. Un mapa para seleccionar la ubicación física donde se realiza el proyecto.
 3. Un campo para añadir una descripción al proyecto.
 4. Un área privada donde estarán disponibles los documentos relacionados con el proyecto.
 5. Una sección de contabilidad privada donde estarán disponibles los justificantes de los gastos del proyecto.
 6. El área privada de un proyecto debe tener un formulario de subida de ficheros y un visor de documentos, con funciones de ordenación y filtrado de los mismos.
- Seleccionar un *hosting* para desplegar la página web.
 - Creación, publicitación y gestión de eventos y noticias.
 - Gestión del SEO de la página web; de forma estática o dinámica.
 - Sistema de avisos a los usuarios por correo electrónico o por número de teléfono, comúnmente llamado *newsletter*.
 - Se deben permitir realizar pagos a través de la página web. Preferiblemente con los siguientes métodos de pago:
 1. Domiciliación bancaria
 2. Tarjeta de crédito/débito
 - El diseño de la aplicación debe ser adaptable a todo tipo de dispositivos, conocido también como diseño *responsive*.
 - La gama de colores para la aplicación debe seguir un patrón uniforme para que no produzca estrés visual a los usuarios.



4. Análisis

En este apartado se estudia con detalle la estructura que tiene la aplicación. Se analiza el diseño estructural del proyecto, las entidades por las que estará compuesta la aplicación web, las distintas funcionalidades que se requieren, los usuarios, las acciones que pueden realizar, el diseño de la base de datos y descripciones de las características del proyecto resultante mediante la elaboración de distintos diagramas.

4.1. UML

Para la construcción de los diagramas estructurales y funcionales del proyecto se ha utilizado el lenguaje Unificado de Modelado o UML (“*Unified Modeling Language*”)⁵. UML es un lenguaje estandarizado de modelado. Está especialmente desarrollado para ayudar a todos los equipos que colaboran en el desarrollo y modelado de un sistema o un producto *software* a describir, diseñar, especificar, visualizar, construir y documentar todos los artefactos que lo componen. La Imagen 3 muestra el logotipo de UML.



Imagen 3. Logotipo de UML

Los diagramas realizados para la representación de este trabajo son los siguientes:

4.1.1. Diagrama de contexto

El diagrama de contexto representa la visión general de la aplicación. No requiere mucho detalle y lo que se quiere conseguir es que se pueda entender de una forma muy

⁵ <https://www.uml.org/>

genérica la estructura del *software* y los componentes o servicios externos con los que se conecta para llevar a cabo las diferentes funciones.

La aplicación web desarrollada trabaja internamente la gestión de usuarios, la gestión de proyectos, la gestión de eventos, la gestión de noticias, la información SEO de la página web, la gestión de los avisos del boletín de anuncios y la gestión de las donaciones realizadas pero los servicios de pago con tarjeta y ejecución del envío de avisos por correo electrónico se realizan mediante servicios externos como son Stripe⁶ y Gmail⁷. La Imagen 4 muestra el diagrama de contexto de la aplicación web desarrollada.

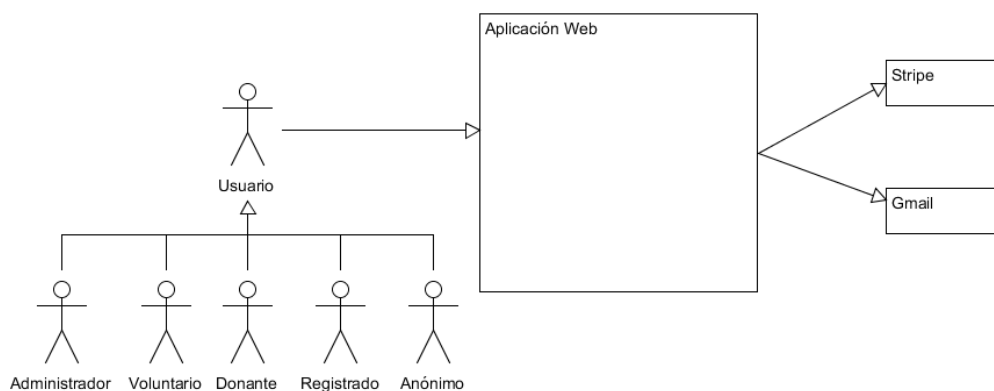


Imagen 4. Diagrama de contexto

En el diagrama de contexto se observa la entidad externa de Stripe, la cual se ha utilizado en el proyecto para permitir pagos con tarjetas pero ese servicio puede ser reemplazado por otro servicio bancario que se adapte mejor a las necesidades de la ONG. La mayoría de las entidades bancarias ofrecen plataformas de pagos on-line con variadas tarifas, pero en el momento del desarrollo de este TFG se desconocía la entidad bancaria con la que se trabajará por lo que se ha optado por una plataforma capaz de trabajar con cualquier entidad bancaria.

4.1.2. Modelo de dominio

El diagrama de modelo de dominio permite establecer un lenguaje común entre clientes y desarrolladores. Puede verse como un glosario gráfico para conceptos del dominio del problema. Ayuda a identificar y nombrar conceptos importantes en el

⁶ <https://stripe.com/es>

⁷ <https://gmail.com/>

contexto del sistema a desarrollar: entidades que se deben manipular o sobre los que se registra información (ej. usuario, proyecto, donación, etc.).

Se utiliza para identificar y especificar relaciones simples o múltiples entre las distintas entidades del proyecto. Son clases sin servicios que permiten identificar y nombrar las relaciones entre estos conceptos (opcionalmente atributos relevantes) y las relaciones son solamente de asociación y cardinalidades (opcionalmente herencia). Este tipo de diagramas se centran en el dominio y no en los aspectos de diseño. La Imagen 5 muestra el modelo de dominio de la aplicación.

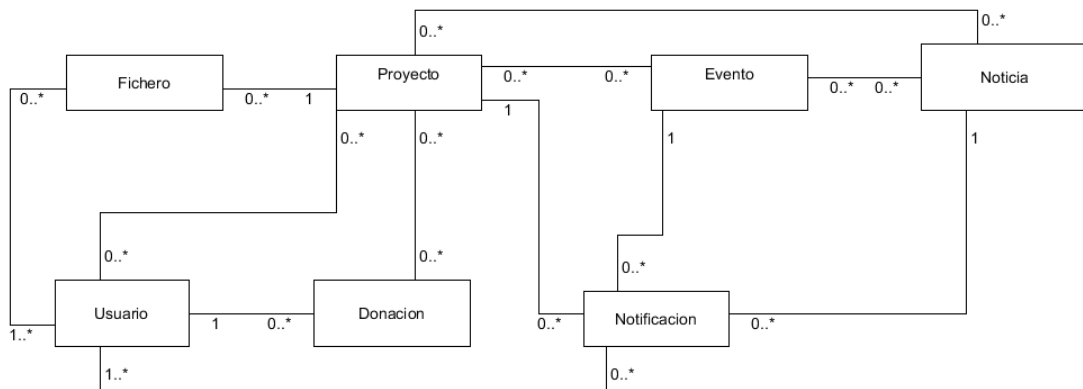


Imagen 5. Modelo de dominio

4.1.3. Diagrama de clases

El diagrama de clases muestra la estructura del sistema, subsistema o componente utilizando clases con sus características, restricciones y relaciones: asociaciones, generalizaciones, dependencias, etc.

Las clases y las relaciones que se tendrán en cuenta para el desarrollo de este proyecto son y el diagrama se muestra en la Imagen 6:

- **Usuario:** compuesto por los atributos `_id`, `name`, `email`, `password`, `role`, `birthday`, `newsletter` y `timestamps`.
- **Proyecto:** compuesto por los atributos `_id`, `name`, `description`, `coordinates`, `location` y `timestamps`.
- **Donación:** compuesto por los atributos `_id`, `donerId`, `amount`, `projectId`, `paymentMethod`, `paymentDate` y `timestamps`.



- **Evento:** compuesto por los atributos `_id`, `name`, `description`, `location`, `date`, `time` y `timestamps`.
- **Noticia:** compuesto por los atributos `_id`, `name`, `description`, `link` y `timestamps`.
- **Notificación:** compuesto por los atributos `_id`, `email` y `timestamps`.
- **Fichero:** compuesto por los atributos `_id`, `uploadedName`, `path`, `size`, `type`, `projectId`, `uploaderId` y `timestamps`.

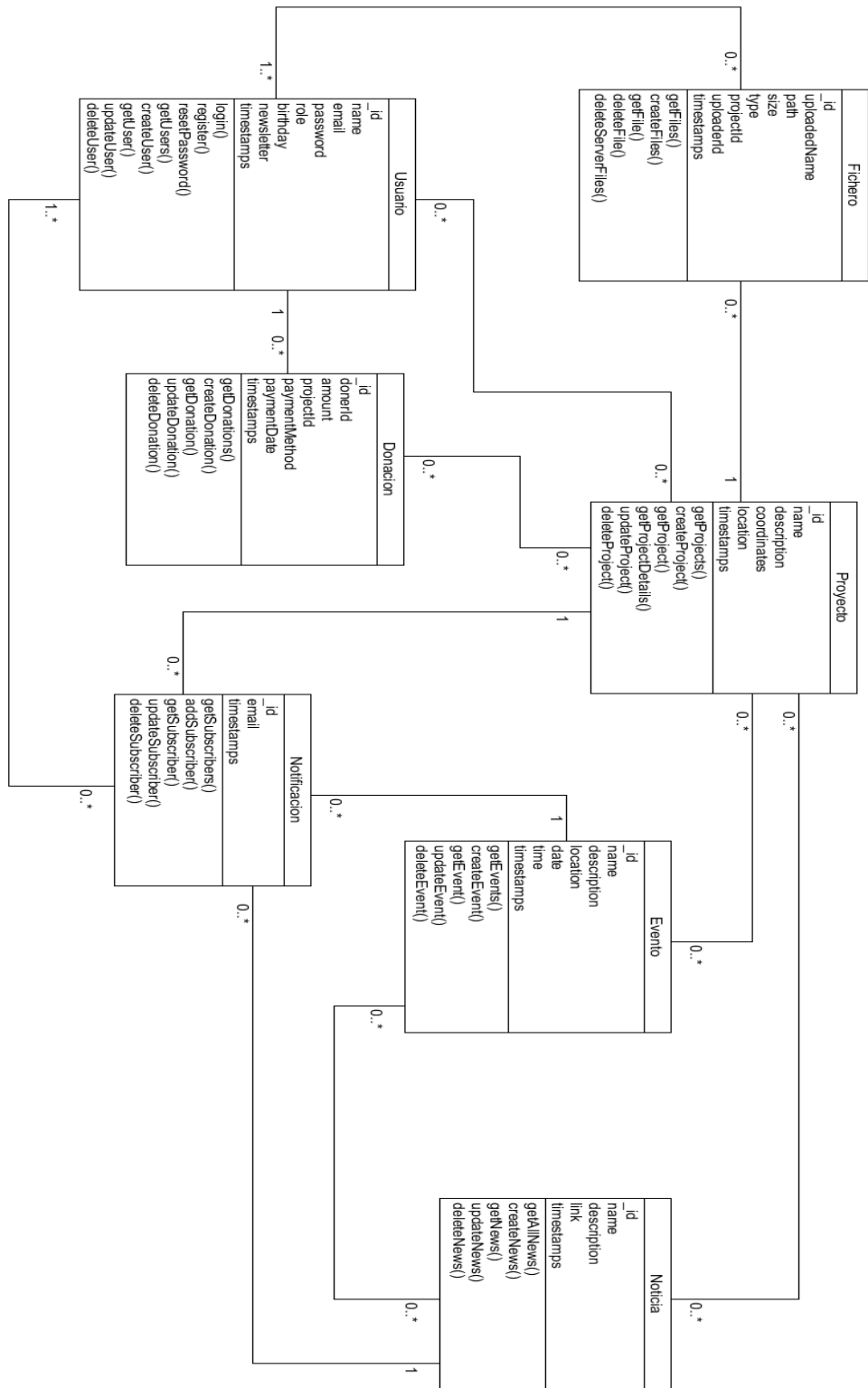


Imagen 6. Diagrama de clases



4.1.4. Diagrama de casos de uso

El diagrama de casos de uso se utiliza para especificar el comportamiento deseado de un sistema o subsistema. Su especificación describe el conjunto de secuencias de acciones que se lleva a cabo en el sistema para producir un resultado para un actor. Capturan el comportamiento deseado del sistema, sin especificar cómo se lleva a cabo dicho comportamiento. Ayudan a validar y verificar el sistema durante el desarrollo.

La aplicación está dividida en dos partes; *frontoffice* y *backoffice*. Por tanto, los casos de uso están separados para cada parte. Las características junto a sus casos de uso para este proyecto se detallan a continuación:

4.1.4.1. Frontoffice

Las funcionalidades que se mencionan a continuación están disponibles en la parte pública o el *frontoffice* de la aplicación web y el acceso a ellas depende del rol de cada usuario.

- **Registrarse:** todos los usuarios anónimos tienen disponible esta opción y una vez registrados, cambian de rol y se convierten en registrados, voluntario o administrador. En caso de realizar una donación, pasan al rol de donante. La Tabla 1 muestra la descripción detallada del caso de uso “Registrarse”:

Caso de uso 01: Registrarse	
Descripción	Crear una cuenta en la aplicación.
Actores	Anónimo
Precondición	1. El usuario no tiene cuenta en la aplicación.
Secuencia	1. El usuario accede a la página del formulario de registro. 2. Rellena todos los campos. 3. Pulsa el botón “Crear Cuenta”. 4. Se da de alta al usuario en la base de datos. 5. Se le lleva al usuario a la página principal. 6. Se envía un correo de bienvenida al usuario.
Postcondición	1. El usuario tiene una cuenta creada en la aplicación. 2. El usuario recibe el correo electrónico de bienvenida en su buzón.

Tabla 1. Caso de uso: Registrarse

- **Iniciar sesión:** esta opción está disponible en la página web para todos los usuarios. La Tabla 2 muestra la descripción detallada del caso de uso “Iniciar sesión”:

Caso de uso 02: Iniciar sesión	
Descripción	Acceder a la aplicación.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	1. El usuario tiene cuenta en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página del formulario de iniciar sesión. 2. Rellena el campo del correo electrónico y contraseña. 3. Pulsa el botón “Iniciar sesión”. 4. Se comprueban las credenciales y se concede el acceso a la aplicación según el rol del usuario. 5. Se le lleva al usuario a la página principal.
Postcondición	1. El usuario puede usar la aplicación según sus permisos.

Tabla 2. Caso de uso: Iniciar sesión

- **Cerrar sesión:** esta opción también está disponible para todos los usuarios que tenga cuenta en la página web. La Tabla 3 muestra la descripción detallada del caso de uso “Cerrar sesión”:

Caso de uso 03: Cerrar sesión	
Descripción	Salir de la aplicación.
Actores	Registrado, Donante, Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene cuenta en la aplicación. 2. EL usuario tiene la sesión iniciada en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede al menú principal de la aplicación. 2. Pulsa el botón “Salir”. 3. Se borran los datos de la caché del navegador. 4. Se le lleva al usuario a la página principal.
Postcondición	1. El usuario vuelve a la página de inicio sin ningún permiso especial.

Tabla 3. Caso de uso: Cerrar sesión

- **Ver perfil:** esta opción estará disponible cuando un usuario inicie sesión en la aplicación web. La Tabla 4 muestra la descripción detallada del caso de uso “Ver perfil”:

Caso de uso 04: Ver perfil	
Descripción	Ver la información personal.

Actores	Registrado, Donante, Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene cuenta en la aplicación. 2. EL usuario tiene la sesión iniciada en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede al menú principal de la aplicación. 2. Pulsa el botón “Mi perfil”. 3. Se le lleva al usuario a la página del perfil.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede consultar la información personal de su cuenta.

Tabla 4. Caso de uso: Ver perfil

- **Resetear contraseña:** esta función permite resetear la contraseña a un usuario. La Tabla 5 muestra la descripción detallada del caso de uso “Resetear contraseña”:

Caso de uso 05: Resetear contraseña	
Descripción	Cambiar la contraseña de acceso a la aplicación.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene cuenta en la aplicación.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede al formulario de iniciar sesión. 2. Pulsa el enlace “¿Has olvidado la contraseña?”. 3. Se muestra una ventana donde se solicita el correo electrónico de recuperación. 4. El usuario rellena el campo y envía el dato. 5. Se le envía un correo al usuario a la dirección indicada que contiene un enlace para resetear la contraseña.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede cambiar su contraseña de acceso.

Tabla 5. Caso de uso: Resetear contraseña

- **Ver proyectos:** esta funcionalidad es pública y cualquier usuario de la página puede acceder a ella para consultar los proyectos en los que está trabajando la ONG. La Tabla 6 muestra la descripción detallada del caso de uso “Ver proyectos”:

Caso de uso 06: Ver proyectos	
Descripción	Consultar la información de los proyectos de la ONG.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	Ninguna
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación. 2. Pulsa sobre cualquier proyecto de los visibles en la página web.

	3. Accede a la página donde puede consultar el nombre, la descripción y el lugar del proyecto.
Postcondición	1. El usuario puede consultar la información pública del proyecto.

Tabla 6. Caso de uso: Ver perfil

- **Ver eventos:** permite a todos los usuarios consultar los eventos organizados por la ONG. La Tabla 7 muestra la descripción detallada del caso de uso “Ver eventos”:

Caso de uso 07: Ver eventos	
Descripción	Consultar la información de los eventos de la ONG.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	Ninguna
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación. 2. Pulsa sobre la sección de eventos en la página web. 3. Accede a la página donde se listan todos los eventos públicos junto con sus detalles como es el nombre, resumen, lugar fecha y hora.
Postcondición	1. El usuario puede consultar la información pública de los eventos.

Tabla 7. Caso de uso: Ver eventos

- **Ver noticias:** permite a todos los usuarios consultar las noticias relacionadas con la ONG. La Tabla 8 muestra la descripción detallada del caso de uso “Ver noticias”:

Caso de uso 08: Ver noticias	
Descripción	Consultar la información de las noticias de la ONG.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	Ninguna
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación. 2. Pulsa sobre la sección de noticias en la página web. 3. Accede a la página donde se listan todas las noticias de la ONG junto con sus detalles como es el titular, detalles, enlace y fecha.
Postcondición	1. El usuario puede consultar la información pública de las noticias.

Tabla 8. Caso de uso: Ver noticias

- **Donar:** esta función está disponible para cualquier usuario, esté registrado o no y permite realizar una donación económica a la ONG a través de la página web. La Tabla 9 muestra la descripción detallada del caso de uso “Donar”:

Caso de uso 09: Donar	
Descripción	Realizar una donación económica a la ONG.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	Ninguna
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación. 2. Pulsa sobre la opción de “Donar” en el menú principal de la página web. 3. Accede a la página donde tiene disponible el formulario a rellenar para realizar el pago. 4. El usuario rellena los campos y envía el formulario. 5. Se comprueban los datos mediante los sistemas externos de pagos o se muestra un PDF con los datos bancarios de la ONG en caso de solicitar domiciliación bancaria.
Postcondición	<ol style="list-style-type: none"> 1. El usuario realizar una donación a la ONG.

Tabla 9. Caso de uso: Donar

- **Contactar:** es un formulario de contacto disponible en la página web y es accesible para todos los usuarios. La Tabla 10 muestra la descripción detallada del caso de uso “Contactar”:

Caso de uso 10: Contactar	
Descripción	Enviar un mensaje a la ONG.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	Ninguna
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación. 2. Pulsa sobre la opción de “Contacto” en el menú principal o el pie de página de la web. 3. Accede a la página donde tiene disponible el formulario a rellenar para enviar el mensaje. 4. El usuario rellena los campos y envía el formulario. 5. Se comprueban los datos. 6. Se envía el mensaje y los datos del usuario al correo de la ONG.
Postcondición	<ol style="list-style-type: none"> 1. El usuario ve una notificación de mensaje enviado.

	2. La ONG recibe el mensaje del usuario en su buzón de correo electrónico.
--	--

Tabla 10. Caso de uso: Contactar

- **Suscribirse al boletín de noticias:** esta función es pública y cualquier persona puede darse de alta en el boletín de noticias. La Tabla 11 muestra la descripción detallada del caso de uso “Suscribirse al boletín de noticias”:

Caso de uso 11: Suscribirse al boletín de noticias	
Descripción	Darse de alta en la lista de avisos.
Actores	Anónimo, Registrado, Donante, Voluntario y Administrador
Precondición	Ninguna
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación. 2. Introduce su correo electrónico en el campo del boletín y pulsa el botón “Suscribirse”. 3. Se valida el formulario. 4. Se da de alta el correo electrónico en la lista de avisos. 5. Se le envía un correo electrónico al usuario comunicando la suscripción.
Postcondición	<ol style="list-style-type: none"> 1. El usuario ve una notificación de mensaje enviado. 2. El usuario recibe un correo electrónico de suscripción.

Tabla 11. Caso de uso: Suscribirse al boletín de noticias

- **Ver donaciones realizadas:** esta funcionalidad está disponible para los usuarios donantes (solo las donaciones que haya realizado). La Tabla 12 muestra la descripción detallada del caso de uso “Ver donaciones realizadas”:

Caso de uso 12: Ver donaciones realizadas	
Descripción	Ver las donaciones que ha realizado un usuario.
Actores	Donante
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario ha realizado al menos una donación a la ONG.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación. 2. El usuario inicia sesión con su cuenta. 3. Pulsa la opción “Ver perfil” en el menú principal de la página web. 4. Accede a la página de su perfil y se listan las donaciones que ha realizado con una comprobación previa de su

	rol.
Postcondición	1. El usuario donante puede consultar todas sus donaciones.

Tabla 12. Caso de uso: Ver donaciones realizadas

- **Ver documentos del proyecto:** esta funcionalidad está disponible para los usuarios donantes (solos de los proyectos a los que han donado). La Tabla 13 muestra la descripción detallada del caso de uso “Ver documentos del proyecto”:

Caso de uso 13: Ver documentos del proyecto	
Descripción	Ver los documentos visibles del proyecto.
Actores	Donante
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario ha realizado al menos una donación para un proyecto a la ONG. 4. El proyecto al que se ha donado tiene documentos subidos y son visibles.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario inicia sesión con su cuenta. 3. El usuario accede a ver la información de cualquier proyecto al que ha donado. 4. Se listan los documentos que pertenecen al proyecto.
Postcondición	1. El usuario donante puede consultar todos los documentos de los proyectos a los que han donado.

Tabla 13. Caso de uso: Ver documentos del proyecto

4.1.4.2.Backoffice

En esta sección las características y casos de uso que se describen están disponibles en la parte privada o el *backoffice* de la aplicación web y el acceso a ellas solamente se permite a los trabajadores de la ONG (administradores y voluntarios).

Gestión de usuarios: el acceso a esta característica solo está permitido a los administradores.

- **Listar usuarios:** listar todos los usuarios registrados en la página web. La Tabla 14 muestra la descripción detallada del caso de uso “Listar usuarios”:

Caso de uso 14: Listar usuarios	
Descripción	Ver los usuarios dados de alta en la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar usuarios”. 5. Se muestra el listado de todos los usuarios dados de alta en la aplicación.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede ver el listado de los usuarios de la aplicación.

Tabla 14. Caso de uso: Listar usuarios

- **Ver usuario:** ver la información de un usuario. La Tabla 15 muestra la descripción detallada del caso de uso “Ver usuario”:

Caso de uso 15: Ver usuario	
Descripción	Ver la información de un usuario.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar usuarios”. 5. El usuario pulsa el icono del ojo. 6. Se muestra una ventana con los datos del usuario.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede consultar los datos de un usuario.

Tabla 15. Caso de uso: Ver usuario

- **Crear usuario:** añadir un usuario en la base de datos. La Tabla 16 muestra la descripción detallada del caso de uso “Crear usuario”:

Caso de uso 16: Crear usuario

Descripción	Dar de alta a un usuario en la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar usuarios”. 5. El usuario pulsa el botón “Añadir usuario”. 6. Se muestra una ventana con un formulario para añadir un usuario nuevo. 7. El usuario rellena los campos y envía el formulario. 8. Se valida el formulario. 9. Se da de alta al usuario en la aplicación. 10. Se muestra una notificación de “Usuario añadido”. 11. Se envía un correo de bienvenida al usuario.
Postcondición	<ol style="list-style-type: none"> 1. El nuevo usuario se da de alta en la aplicación. 2. El usuario dado de alta recibe un correo de bienvenida.

Tabla 16. Caso de uso: Crear usuario

- **Editar usuario:** actualizar los datos de un usuario en la base de datos. La Tabla 17 muestra la descripción detallada del caso de uso “Editar usuario”:

Caso de uso 17: Editar usuario	
Descripción	Editar los datos de un usuario de la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar usuarios”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con un formulario con los

	campos rellenos con la información del usuario. 7. El usuario actualiza los campos y envía el formulario. 8. Se valida el formulario. 9. Se actualizan los datos en la base de datos. 10. Se muestra una notificación de “Usuario actualizado”.
Postcondición	1. Los datos se actualizan correctamente en la base de datos.

Tabla 17. Caso de uso: Editar usuario

- **Borrar usuario:** borrar un usuario en la base de datos. La Tabla 18 muestra la descripción detallada del caso de uso “Borrar usuario”:

Caso de uso 18: Borrar usuario	
Descripción	Borrar un usuario de la aplicación.
Actores	Administrador
Precondición	1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i> . 4. El usuario accede al apartado de “Gestionar usuarios”. 5. El usuario pulsa el icono de la papelera. 6. Se muestra una ventana de confirmación. 7. El usuario pulsa el botón “Borrar”. 8. Se borra el usuario de la base de datos.
Postcondición	1. Los datos del usuario se eliminan correctamente de la base de datos.

Tabla 18. Caso de uso: Borrar usuario

Gestión de proyectos: los administradores tienen acceso total a esta característica y los voluntarios pueden acceder a “Ver proyectos”, “Borrar fichero” y “Subir fichero” pero únicamente pueden visualizar los datos de los proyectos a los que están vinculados.

- **Listar proyectos:** listar todos los proyectos de la ONG. La Tabla 19 muestra la descripción detallada del caso de uso “Listar proyectos”:

Caso de uso 19: Listar proyectos	
Descripción	Ver los proyectos dados de alta en la aplicación.

Actores	Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario debe tener al menos un proyecto asignado.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. Se muestra el listado de todos los proyectos de la ONG.
Postcondición	<ol style="list-style-type: none"> 1. El usuario administrador puede ver la lista de todos los proyectos de la ONG. 2. El usuario voluntario puede ver la lista de todos los proyectos que tiene asignado.

Tabla 19. Caso de uso: Listar proyectos

- **Ver proyecto:** ver la información de un proyecto. La Tabla 20 muestra la descripción detallada del caso de uso “Ver proyecto”:

Caso de uso 20: Ver proyecto	
Descripción	Ver la información de un proyecto.
Actores	Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario debe tener el proyecto asignado.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono del ojo. 6. Se muestra una ventana con los datos del proyecto.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede consultar los datos de un proyecto.

Tabla 20. Caso de uso: Ver proyecto

- **Crear proyecto:** añadir un proyecto nuevo en la base de datos. La Tabla 21 muestra la descripción detallada del caso de uso “Crear proyecto”:

Caso de uso 21: Crear proyecto	
Descripción	Añadir un proyecto en la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el botón “Añadir proyecto”. 6. Se muestra una ventana con un formulario para añadir un proyecto nuevo. 7. El usuario rellena los campos y envía el formulario. 8. Se valida el formulario. 9. Se crea un proyecto nuevo en la aplicación. 10. Se muestra una notificación de “Proyecto añadido”.
Postcondición	<ol style="list-style-type: none"> 1. El proyecto se añade correctamente en la base de datos.

Tabla 21. Caso de uso: Crear proyecto

- **Editar proyecto:** actualizar los datos de un proyecto en la base de datos. La Tabla 22 muestra la descripción detallada del caso de uso “Editar proyecto”:

Caso de uso 22: Editar proyecto	
Descripción	Editar la información de un proyecto de la ONG.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con un formulario con los campos rellenos con la información del proyecto.

	<ol style="list-style-type: none"> 7. El usuario actualiza los campos y envía el formulario. 8. Se valida el formulario. 9. Se actualizan los datos en la base de datos. 10. Se muestra una notificación de “Proyecto actualizado”.
Postcondición	<ol style="list-style-type: none"> 1. Los datos se actualizan correctamente en la base de datos.

Tabla 22. Caso de uso: Editar proyecto

- **Borrar proyecto:** borrar un proyecto en la base de datos. La Tabla 23 muestra la descripción detallada del caso de uso “Borrar proyecto”:

Caso de uso 23: Borrar proyecto	
Descripción	Borrar un proyecto de la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono de la papelera. 6. Se muestra una ventana de confirmación. 7. El usuario pulsa el botón “Borrar”. 8. Se borra el proyecto de la base de datos. 9. Se borran todos los ficheros relacionados al proyecto si el usuario lo ha indicado así en la ventana de confirmación.
Postcondición	<ol style="list-style-type: none"> 1. Los datos del proyecto se eliminan correctamente de la base de datos. 2. Los datos de los ficheros del proyecto se borran correctamente de la base de datos si así lo ha indicado el usuario. 3. Los ficheros se borran del servidor si así lo ha indicado el usuario.

Tabla 23. Caso de uso: Borrar proyecto

- **Añadir voluntario:** añadir voluntarios a un proyecto. La Tabla 24 muestra la descripción detallada del caso de uso “Añadir voluntario”:

Caso de uso 24: Añadir voluntario	
Descripción	Añadir voluntarios al proyecto.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono de persona. 6. Se muestra una ventana con el listado de los trabajadores de la ONG junto con los botones para añadir o quitar. 7. El usuario pulsa el botón de “Añadir usuario”.
Postcondición	<ol style="list-style-type: none"> 1. El trabajador se añade en la lista de los voluntarios del proyecto.

Tabla 24. Caso de uso: Añadir voluntario

- **Quitar voluntario:** quitar voluntarios de un proyecto. La Tabla 25 muestra la descripción detallada del caso de uso “Quitar voluntario”:

Caso de uso 25: Quitar voluntario	
Descripción	Quitar voluntarios del proyecto.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono de persona. 6. Se muestra una ventana con el listado de los trabajadores de la ONG junto con los botones para añadir o quitar.

	7. El usuario pulsa el botón de “Quitar usuario”.
Postcondición	1. El trabajador se quita de la lista de los voluntarios del proyecto.

Tabla 25. Caso de uso: Quitar voluntario

- **Subir fichero:** subir un fichero para un proyecto en concreto. La Tabla 26 muestra la descripción detallada del caso de uso “Subir fichero”:

Caso de uso 26: Subir fichero	
Descripción	Subir un fichero para un proyecto de la aplicación.
Actores	Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario tiene asignado el proyecto al que se va subir el fichero.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con un formulario con los campos rellenos con la información del proyecto. 7. El usuario va al campo de subir ficheros y selecciona los ficheros que quiere subir. 8. Se envía el formulario. 9. Se añaden los datos de los ficheros en la base de datos. 10. Se guardan los ficheros en el servidor. 11. Se muestra una notificación de “Proyecto actualizado”.
Postcondición	<ol style="list-style-type: none"> 1. Los datos de los ficheros se añaden correctamente en la base de datos. 2. Los ficheros se suben correctamente al servidor. 3. Los ficheros subidos están visibles en los detalles del proyecto.

Tabla 26. Caso de uso: Subir fichero

- **Descargar fichero:** descargar o visualizar cualquier fichero de un proyecto. La Tabla 27 muestra la descripción detallada del caso de uso “Descargar fichero”:

Caso de uso 27: Descargar fichero	
Descripción	Descargar un fichero del proyecto.
Actores	Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario tiene asignado el proyecto del que se quiere descargar el fichero.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono del ojo. 6. Se muestra una ventana con la información del proyecto que incluye los ficheros. 7. El usuario pulsa el icono de descarga. 8. Se abre el fichero si es compatible con el navegador y en caso contrario, se descarga.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede descargar o visualizar los ficheros de los proyectos.

Tabla 27. Caso de uso: Descargar fichero

- **Borrar fichero:** borrar la información del fichero de la base de datos. La Tabla 28 muestra la descripción detallada del caso de uso “Borrar fichero”:

Caso de uso 28: Borrar fichero	
Descripción	Borrar un fichero del proyecto.
Actores	Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario tiene asignado el proyecto del que se quiere borrar el fichero.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>.



	<ol style="list-style-type: none"> 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con la información del proyecto que incluye los ficheros. 7. El usuario pulsa el icono de la papelera. 8. La información del fichero se borra de la base de datos.
Postcondición	<ol style="list-style-type: none"> 1. La información del fichero se borra correctamente de la base de datos.

Tabla 28. Caso de uso: Borrar fichero

- **Borrar fichero del servidor:** borrar el fichero subido al servidor. La Tabla 29 muestra la descripción detallada del caso de uso “Borrar fichero del servidor”:

Caso de uso 29: Borrar fichero del servidor	
Descripción	Borrar un fichero del servidor.
Actores	Voluntario y Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario tiene asignado el proyecto del que se quiere borrar el fichero.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar proyectos”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con la información del proyecto que incluye los ficheros. 7. El usuario pulsa el icono de la papelera. 8. El fichero se borra del servidor.
Postcondición	<ol style="list-style-type: none"> 1. El fichero ya no está disponible en el servidor.

Tabla 29. Caso de uso: Borrar fichero del servidor

Gestión de donaciones: el acceso a esta característica solo está permitido a los administradores.

- **Listar donaciones:** listar todas las donaciones a la ONG. La Tabla 30 muestra la descripción detallada del caso de uso “Listar donaciones”:

Caso de uso 30: Listar donaciones	
Descripción	Ver las donaciones realizadas a la ONG.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar donaciones”. 5. Se muestra el listado de todas las donaciones realizadas a la ONG.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede ver la lista de todas las donaciones.

Tabla 30. Caso de uso: Listar donaciones

- **Ver donación:** ver la información de una donación. La Tabla 31 muestra la descripción detallada del caso de uso “Ver donación”:

Caso de uso 31: Ver donación	
Descripción	Ver la información de una donación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar donaciones”. 5. El usuario pulsa el icono del ojo. 6. Se muestra una ventana con los detalles de la donación.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede consultar los detalles de una donación.

Tabla 31. Caso de uso: Ver donación

- **Crear donación:** añadir una nueva donación en la base de datos. La Tabla 32 muestra la descripción detallada del caso de uso “Crear donación”:

Caso de uso 32: Crear donación	
Descripción	Añadir una donación en la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar donaciones”. 5. El usuario pulsa el botón “Añadir donación”. 6. Se muestra una ventana con un formulario para añadir una nueva donación. 7. El usuario rellena los campos y envía el formulario. 8. Se valida el formulario. 9. Se añade la donación en la aplicación. 10. Se muestra una notificación de “Donación añadida”.
Postcondición	<ol style="list-style-type: none"> 1. La nueva donación se añade correctamente en la base de datos de la aplicación.

Tabla 32. Caso de uso: Crear donación

- **Editar donación:** actualizar los datos de una donación en la base de datos. La Tabla 33 muestra la descripción detallada del caso de uso “Editar donación”:

Caso de uso 33: Editar donación	
Descripción	Editar la información de una donación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar donaciones”.

	<ol style="list-style-type: none"> 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con un formulario con los campos rellenos con la información de la donación. 7. El usuario actualiza los campos y envía el formulario. 8. Se valida el formulario. 9. Se actualizan los datos en la base de datos. 10. Se muestra una notificación de “Donación actualizada”.
Postcondición	<ol style="list-style-type: none"> 1. Los datos se actualizan correctamente en la base de datos.

Tabla 33. Caso de uso: Editar donación

- **Borrar donación:** borrar la información de una donación en la base de datos.

La Tabla 34 muestra la descripción detallada del caso de uso “Borrar donación”:

Caso de uso 34: Borrar donación	
Descripción	Borrar una donación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar donaciones”. 5. El usuario pulsa el icono de la papelera. 6. Se muestra una ventana de confirmación. 7. El usuario pulsa el botón “Borrar”. 8. Se borra la donación de la base de datos.
Postcondición	<ol style="list-style-type: none"> 1. Los datos de la donación se eliminan correctamente de la base de datos.

Tabla 34. Caso de uso: Borrar donación

Gestión de eventos: el acceso a esta característica solo está permitido a los administradores.

- **Listar eventos:** listar todos los eventos de la ONG. La Tabla 35 muestra la descripción detallada del caso de uso “Listar eventos”:

Caso de uso 35: Listar eventos



Descripción	Ver los eventos de la ONG.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar eventos”. 5. Se muestra el listado de todos los eventos de la ONG.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede ver la lista de todos los eventos.

Tabla 35. Caso de uso: Listar eventos

- **Ver evento:** ver la información de un evento. La Tabla 36 muestra la descripción detallada del caso de uso “Ver evento”:

Caso de uso 36: Ver evento	
Descripción	Ver la información de un evento.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar eventos”. 5. El usuario pulsa el icono del ojo. 6. Se muestra una ventana con los detalles del evento.
Postcondición	<ol style="list-style-type: none"> 1. El usuario puede consultar los detalles de un evento.

Tabla 36. Caso de uso: Ver evento

- **Crear evento:** añadir un evento nuevo en la base de datos. La Tabla 37 muestra la descripción detallada del caso de uso “Crear evento”:

Caso de uso 37: Crear evento	
Descripción	Añadir un evento en la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación.

	<ol style="list-style-type: none"> 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar eventos”. 5. El usuario pulsa el botón “Añadir evento”. 6. Se muestra una ventana con un formulario para añadir un nuevo evento. 7. El usuario rellena los campos y envía el formulario. 8. Se valida el formulario. 9. Se añade el evento en la aplicación. 10. Se muestra una notificación de “Evento añadido”.
Postcondición	<ol style="list-style-type: none"> 1. El nuevo evento se añade correctamente en la base de datos de la aplicación.

Tabla 37. Caso de uso: Crear evento

- **Editar evento:** actualizar los datos de un evento en la base de datos. La Tabla 38 muestra la descripción detallada del caso de uso “Editar evento”:

Caso de uso 38: Editar evento	
Descripción	Editar la información de un evento.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar eventos”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con un formulario con los campos rellenos con la información del evento. 7. El usuario actualiza los campos y envía el formulario. 8. Se valida el formulario. 9. Se actualizan los datos en la base de datos.

	10. Se muestra una notificación de “Evento actualizado”.
Postcondición	1. Los datos del evento se actualizan correctamente en la base de datos.

Tabla 38. Caso de uso: Editar evento

- **Borrar evento:** borrar un evento en la base de datos. La Tabla 39 muestra la descripción detallada del caso de uso “Borrar evento”:

Caso de uso 39: Borrar evento	
Descripción	Borrar un evento.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar eventos”. 5. El usuario pulsa el icono de la papelera. 6. Se muestra una ventana de confirmación. 7. El usuario pulsa el botón “Borrar”. 8. Se borra el evento de la base de datos.
Postcondición	1. Los datos del evento se eliminan correctamente de la base de datos.

Tabla 39. Caso de uso: Borrar evento

Gestión de noticias: el acceso a esta característica solo está permitido a los administradores.

- **Listar noticias:** listar todas las noticias de la ONG. La Tabla 40 muestra la descripción detallada del caso de uso “Listar noticias”:

Caso de uso 40: Listar noticias	
Descripción	Ver las noticias de la ONG.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú

	<p>principal.</p> <ol style="list-style-type: none"> El usuario accede a la página del <i>backoffice</i>. El usuario accede al apartado de “Gestionar noticias”. Se muestra el listado de todas las noticias de la ONG.
Postcondición	<ol style="list-style-type: none"> El usuario puede ver la lista de todas las noticias.

Tabla 40. Caso de uso: Listar noticias

- **Ver noticia:** ver la información de una noticia. La Tabla 41 muestra la descripción detallada del caso de uso “Ver noticia”:

Caso de uso 41: Ver noticias	
Descripción	Ver los detalles de una noticia.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> El usuario tiene una cuenta en la aplicación. El usuario ha iniciado sesión con su cuenta. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> El usuario accede a la página de inicio de la aplicación. El usuario pulsa la opción “Backoffice” del menú principal. El usuario accede a la página del <i>backoffice</i>. El usuario accede al apartado de “Gestionar noticias”. El usuario pulsa el icono del ojo. Se muestra una ventana con los detalles de la noticia.
Postcondición	<ol style="list-style-type: none"> El usuario puede consultar los detalles de una noticia.

Tabla 41. Caso de uso: Ver noticia

- **Crear noticia:** añadir una noticia nueva en la base de datos. La Tabla 42 muestra la descripción detallada del caso de uso “Crear noticia”:

Caso de uso 42: Crear noticia	
Descripción	Añadir una nueva noticia.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> El usuario tiene una cuenta en la aplicación. El usuario ha iniciado sesión con su cuenta. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> El usuario accede a la página de inicio de la aplicación. El usuario pulsa la opción “Backoffice” del menú principal. El usuario accede a la página del <i>backoffice</i>. El usuario accede al apartado de “Gestionar noticias”.

	<ol style="list-style-type: none"> 5. El usuario pulsa el botón “Añadir noticia”. 6. Se muestra una ventana con un formulario para añadir una nueva noticia. 7. El usuario rellena los campos y envía el formulario. 8. Se valida el formulario. 9. Se añade la noticia en la aplicación. 10. Se muestra una notificación de “Noticia añadida”.
Postcondición	<ol style="list-style-type: none"> 1. La nueva noticia se añade correctamente en la base de datos de la aplicación.

Tabla 42. Caso de uso: Crear noticia

- **Editar noticia:** actualizar los datos de una noticia en la base de datos. La Tabla 43 muestra la descripción detallada del caso de uso “Editar noticia”:

Caso de uso 43: Editar noticia	
Descripción	Editar los detalles de una noticia.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar noticias”. 5. El usuario pulsa el icono del lápiz. 6. Se muestra una ventana con un formulario con los campos rellenos con la información de la noticia. 7. El usuario actualiza los campos y envía el formulario. 8. Se valida el formulario. 9. Se actualizan los datos en la base de datos. 10. Se muestra una notificación de “Noticia actualizada”.
Postcondición	<ol style="list-style-type: none"> 1. La noticia se actualiza correctamente en la base de datos.

Tabla 43. Caso de uso: Editar noticia

- **Borrar noticia:** borrar una noticia de la base de datos. La Tabla 44 muestra la descripción detallada del caso de uso “Borrar noticia”:

Caso de uso 44: Borrar noticia

Descripción	Borrar una noticia.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar noticias”. 5. El usuario pulsa el icono de la papelera. 6. Se muestra una ventana de confirmación. 7. El usuario pulsa el botón “Borrar”. 8. Se borra la noticia de la base de datos.
Postcondición	<ol style="list-style-type: none"> 1. Los datos de la noticia se eliminan correctamente de la base de datos.

Tabla 44. Caso de uso: Borrar noticia

Gestión SEO: el acceso a esta característica solo está permitido a los administradores.

- **Cambiar título:** cambiar el título público de la página web. La Tabla 45 muestra la descripción detallada del caso de uso “Cambiar título”:

Caso de uso 45: Cambiar título	
Descripción	Cambiar el título de la página web.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar web”. 5. Se muestra el formulario para actualizar el título de la página web. 6. El usuario rellena el campo y envía el formulario. 7. Se valida el formulario. 8. Se actualizan los datos en la base de datos.



	9. Se actualiza la página para aplicar los cambios.
Postcondición	1. El título se actualiza correctamente en la base de datos. 2. El título se actualiza correctamente en la página web.

Tabla 45. Caso de uso: Cambiar título

- **Cambiar meta-título:** cambiar el meta-título de la página web. El meta-título es el título que leen todos los robots rastreadores de los buscadores. La Tabla 46 muestra la descripción detallada del caso de uso “Cambiar meta-título”:

Caso de uso 46: Cambiar meta-título	
Descripción	Cambiar el meta-título de la página web.
Actores	Administrador
Precondición	1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i> . 4. El usuario accede al apartado de “Gestionar web”. 5. Se muestra el formulario para actualizar el meta-título de la página web. 6. El usuario rellena el campo y envía el formulario. 7. Se valida el formulario. 8. Se actualizan los datos en la base de datos. 9. Se actualiza la página para aplicar los cambios.
Postcondición	1. El meta-título se actualiza correctamente en la base de datos. 2. El meta-título se actualiza correctamente en la página web.

Tabla 46. Caso de uso: Cambiar meta-título

- **Cambiar meta-descripción:** cambiar la meta-descripción de la página web. La meta-descripción es la descripción de la página que leen todos los robots rastreadores de los buscadores. La Tabla 47 muestra la descripción detallada del caso de uso “Cambiar meta-descripción”:

Caso de uso 47: Cambiar meta-descripción	
Descripción	Cambiar la meta-descripción de la página web.
Actores	Administrador

Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar web”. 5. Se muestra el formulario para actualizar la meta-descripción de la página web. 6. El usuario rellena el campo y envía el formulario. 7. Se valida el formulario. 8. Se actualizan los datos en la base de datos. 9. Se actualiza la página para aplicar los cambios.
Postcondición	<ol style="list-style-type: none"> 1. La meta-descripción se actualiza correctamente en la base de datos. 2. La meta-descripción se actualiza correctamente en la página web.

Tabla 47. Caso de uso: Cambiar meta-descripción

Gestión de newsletter: El acceso a esta característica solo está permitido a los administradores.

- **Listar suscriptores:** listar todos los suscriptores del boletín de noticias. La Tabla 48 muestra la descripción detallada del caso de uso “Listar suscriptores”:

Caso de uso 48: Listar suscriptores	
Descripción	Listar todos los suscriptores del boletín de noticias.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar newsletter”. 5. Se muestra el listado de todos los suscriptores del

	boletín de noticias.
Postcondición	1. El usuario puede ver la lista de todos los suscriptores.

Tabla 48. Caso de uso: Listar suscriptores

- **Borrar suscriptor:** borrar suscriptor de la base de datos para darlo de baja. La Tabla 49 muestra la descripción detallada del caso de uso “Borrar suscriptor”:

Caso de uso 49: Borrar suscriptor	
Descripción	Borrar un suscriptor del boletín de noticias.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar newsletter”. 5. El usuario pulsa el icono de la papelera. 6. Se muestra una ventana de confirmación. 7. El usuario pulsa el botón “Borrar”. 8. Se borra el suscriptor de la base de datos.
Postcondición	<ol style="list-style-type: none"> 1. Los datos del suscriptor se borran correctamente de la base de datos. 2. El suscriptor deja de recibir avisos por correo electrónico.

Tabla 49. Caso de uso: Borrar suscriptor

Gestión de ficheros: el acceso a esta característica solo está permitido a los administradores.

- **Listar ficheros:** listar todos los ficheros subidos a la aplicación web. La Tabla 50 muestra la descripción detallada del caso de uso “Listar ficheros”:

Caso de uso 50: Listar ficheros	
Descripción	Listar todos los ficheros subidos a la aplicación.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta.

	3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar ficheros”. 5. Se muestra el listado de todos los ficheros de la aplicación.
Postcondición	1. El usuario puede ver la lista de todos los ficheros subidos.

Tabla 50. Caso de uso: Listar ficheros

- **Descargar fichero:** visualizar o descargar un fichero. La Tabla 51 muestra la descripción detallada del caso de uso “Descargar fichero”:

Caso de uso 51: Descargar fichero	
Descripción	Descargar un fichero del proyecto.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar ficheros”. 5. El usuario pulsa el icono de descarga. 6. Se abre el fichero si es compatible con el navegador y en caso contrario, se descarga.
Postcondición	1. El usuario puede descargar o visualizar los ficheros de la aplicación.

Tabla 51. Caso de uso: Descargar fichero

- **Borrar fichero:** borrar la información de un fichero de la base de datos. La Tabla 52 muestra la descripción detallada del caso de uso “Borrar fichero”:

Caso de uso 52: Borrar fichero	
Descripción	Borrar un fichero de la aplicación.
Actores	Administrador
Precondición	1. El usuario tiene una cuenta en la aplicación.

	<ol style="list-style-type: none"> 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de administrador.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar ficheros”. 5. El usuario pulsa el icono de la papelera. 6. La información del fichero se borra de la base de datos.
Postcondición	<ol style="list-style-type: none"> 1. La información del fichero se borra correctamente de la base de datos.

Tabla 52. Caso de uso: Borrar fichero

- **Borrar fichero del servidor:** borrar el fichero subido al servidor. La Tabla 53 muestra la descripción detallada del caso de uso “Borrar fichero del servidor”:

Caso de uso 53: Borrar fichero del servidor	
Descripción	Borrar un fichero del servidor.
Actores	Administrador
Precondición	<ol style="list-style-type: none"> 1. El usuario tiene una cuenta en la aplicación. 2. El usuario ha iniciado sesión con su cuenta. 3. El usuario tiene el rol de voluntario o administrador. 4. El usuario voluntario tiene asignado el proyecto del que se quiere borrar el fichero.
Secuencia	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario pulsa la opción “Backoffice” del menú principal. 3. El usuario accede a la página del <i>backoffice</i>. 4. El usuario accede al apartado de “Gestionar ficheros”. 5. El usuario pulsa el icono de la papelera. 6. El fichero se borra del servidor.
Postcondición	<ol style="list-style-type: none"> 1. El fichero ya no está disponible en el servidor.

Tabla 53. Caso de uso: Borrar fichero del servidor

4.1.4.3. Modelado por actores

Como se ha mencionado en los apartados anteriores, en la aplicación hay seis actores que son: anónimo, registrado, donante, voluntario, administrador y el propio sistema.

4.1.4.3.1. Actor Anónimo

La Imagen 7 muestra el diagrama de casos de uso para usuario “**Anónimo**”:

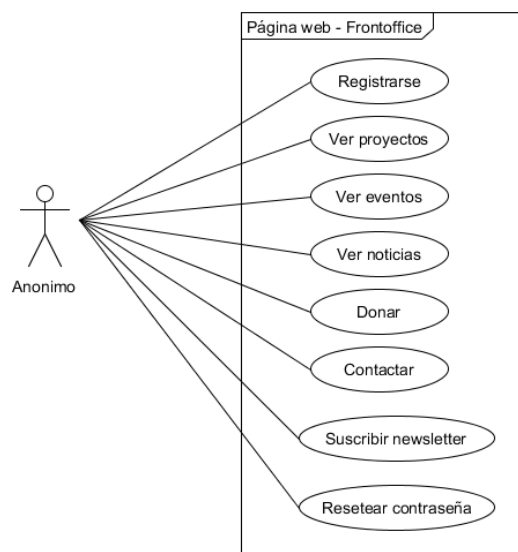


Imagen 7. Diagrama de casos de uso para el actor Anonimo

4.1.4.3.2. Actor Registrado

La Imagen 8 muestra el diagrama de casos de uso para el usuario “**Registrado**”:

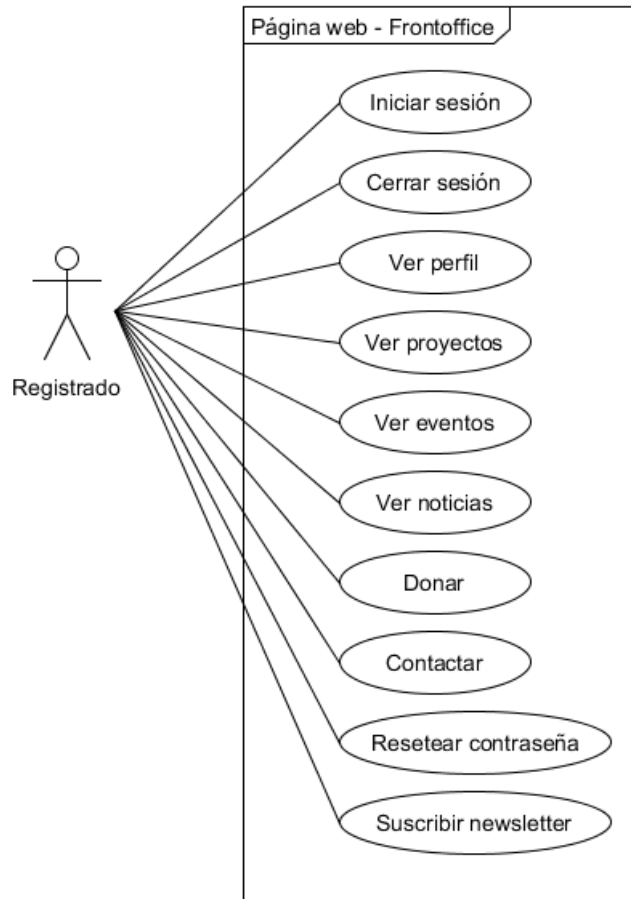


Imagen 8. Diagrama de casos de uso para el actor Registrado

4.1.4.3.3. Actor Donante

La Imagen 9 muestra el diagrama de casos de uso para el usuario “Donante”:

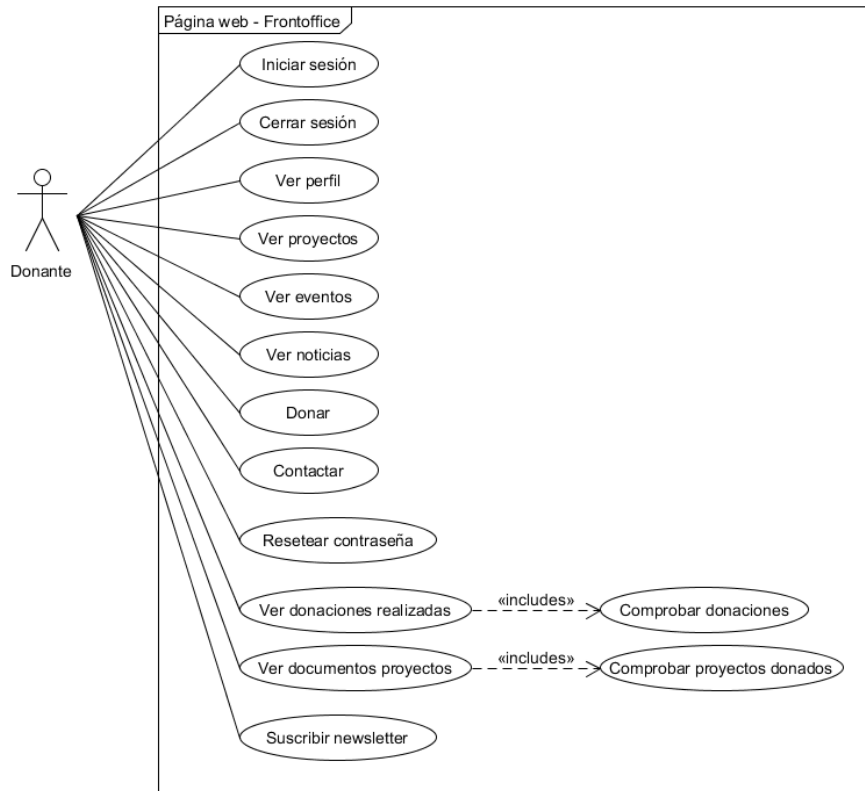


Imagen 9. Diagrama de casos de uso para el actor Donante

4.1.4.3.4. Actor Voluntario

La Imagen 10 muestra el diagrama de casos de uso para el usuario “**Voluntario**”:

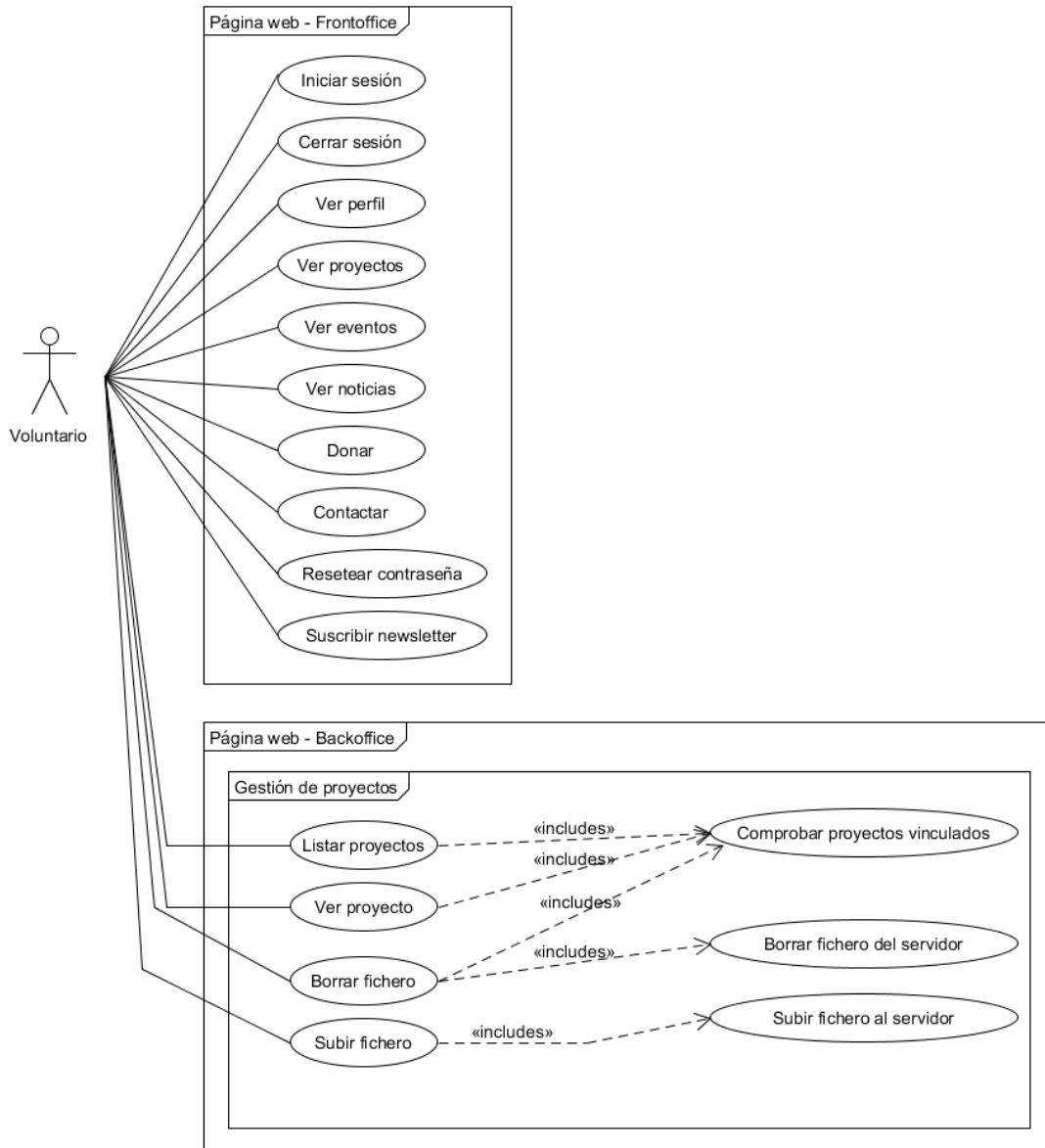


Imagen 10. Diagrama de casos de uso para el actor Voluntario

4.1.4.3.5. Actor Administrador

La Imagen 11 muestra el diagrama de casos de uso para el usuario “**Administrador**” donde se resume la parte del backoffice debido a la falta de legibilidad del diagrama:

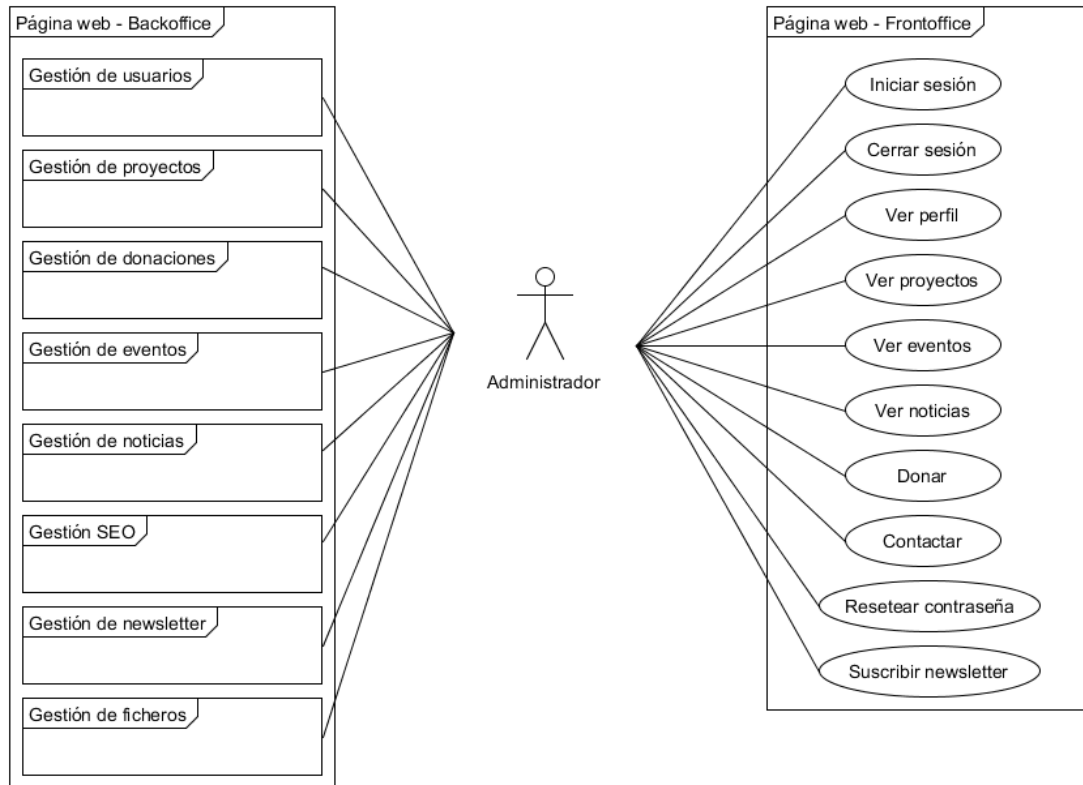


Imagen 11. Diagrama de casos de uso para el actor Administrador (1)

La Imagen 12 muestra el diagrama de casos de uso para el usuario “**Administrador**” con la parte del backoffice detallada:

Sitio Web para la gestión y publicitación de proyectos de cooperación

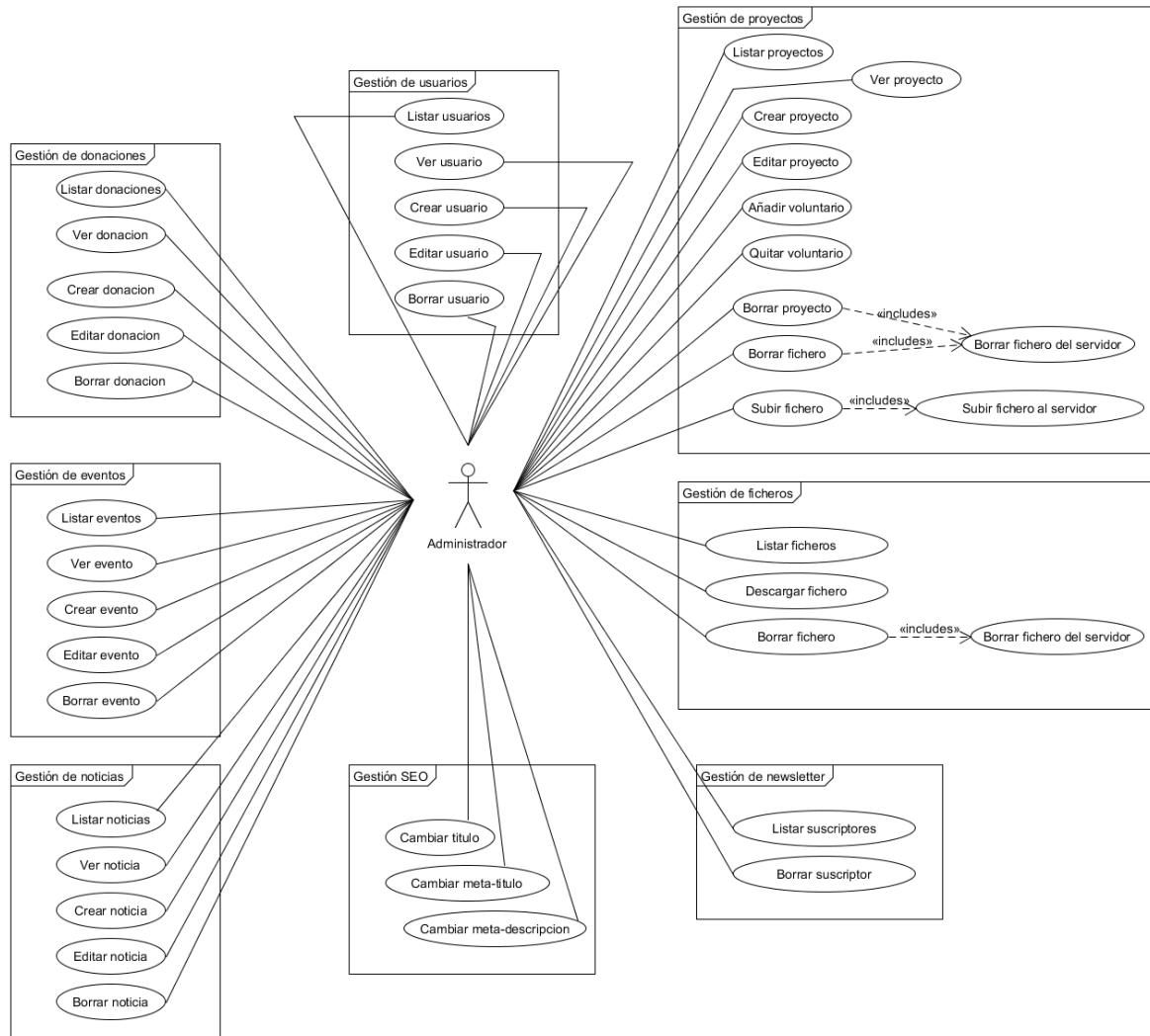


Imagen 12. Diagrama de casos de uso para el actor Administrador (2)

4.1.4.3.6. Actor Sistema

La Imagen 13 muestra el diagrama de casos de uso para el usuario “**Sistema**”:

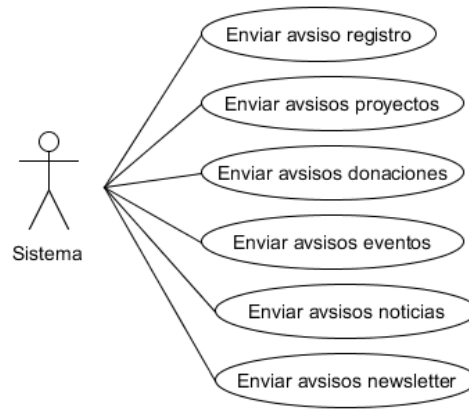


Imagen 13. Diagrama de casos de uso para el actor Sistema

5. Diseño

En esta fase de desarrollo de *software* se estudia el establecimiento de las estructuras de datos, la arquitectura general de *software* y la interfaz gráfica de la aplicación. El proceso de diseño traduce los requisitos de los usuarios en una representación *software*.

5.1. Diseño arquitectónico MVC

El diseño arquitectónico que sigue la aplicación web es el MVC (Modelo-Vista-Controlador o *Model-View-Controller* de sus siglas en inglés) [2]. MVC es un patrón de diseño arquitectónico que separa la aplicación en tres capas principales: el modelo, la vista y el controlador. Cada una de estas capas está construida con el fin de tratar partes específicas del desarrollo de una aplicación. MVC es uno de los *frameworks* estandarizados más utilizados del sector para el desarrollo de proyectos escalables y extensibles.

Este patrón está compuesto por las siguientes capas:

- **Modelo:** se encarga de manejar la lógica de los datos que se trabajan en el proyecto y para ello, tendrá mecanismos para acceder a la información y también para actualizar su estado.
- **Vista:** maneja la parte visual de la aplicación. Se encarga de mostrar al usuario las salidas o resultados de las peticiones realizadas.
- **Controlador:** esta capa se encarga de tratar las acciones solicitadas a través de la aplicación. Sirve de enlace entre la capa de vista y modelo para procesar toda la lógica de negocio.

La Imagen 14 muestra el flujo de interacciones del patrón MVC y el usuario:

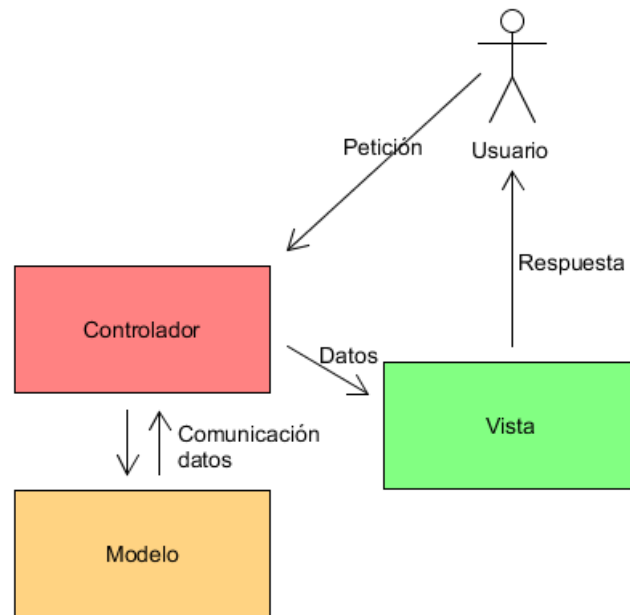


Imagen 14. Patrón de diseño MVC

5.1.1. Elección de MVC

Los motivos por los cuales se ha elegido este patrón de diseño son [3]:

- Al tener la estructura separada y las responsabilidades bien definidas, permite desarrollar *software* más limpio, simple, fácil de mantener y robusto.
- Es una muy buena opción para proyectos grandes. Por tanto, si en un futuro se decide extender la aplicación, será un proceso relativamente más sencillo.
- Permite desarrollar múltiples vistas a partir del mismo modelo aprovechando así el desarrollo y asegurando la consistencia entre las vistas.
- Proporciona facilidad para realizar pruebas unitarias.

5.2. Interfaz gráfica de la aplicación web

En esta sección se hacen los bocetos de la interfaz gráfica que será visible para el usuario final. Para trabajar este apartado, se tienen en cuenta los requisitos especificados en la primera fase del desarrollo debido a que es ahí de donde se obtienen las funcionalidades que necesita el usuario y se consigue información respecto al uso de las interfaces para los usuarios finales.

Los bocetos o también llamados *mockups*, se pueden crear de forma sencilla utilizando lápiz y papel o se puede decantar por opciones más complejas llegando a presentar prototipos muy similares al *software* final a obtener. En este caso, se ha utilizado la herramienta “**Balsamiq Mockups 3**”⁸ que permite diseñar las interfaces gráficas con mucha facilidad con un nivel de detalle muy decente sin llegar a ser ni muy complejo ni muy sencillo. La Imagen 15 muestra el logotipo de la herramienta Balsamiq:



Imagen 15. Logotipo de Balsamiq Mockups 3

Como se ha comentado anteriormente, la aplicación se divide en dos grandes bloques: por una parte está la página web que es accesible por cualquier usuario de la red, el *frontoffice*, y por otro lado, para gestionar la página web está el *backoffice* que tiene acceso limitado por roles. Desde la página web se pueden consultar los proyectos, los eventos, las noticias, consultar información de la ONG, enviar formulario de contacto y se pueden hacer donaciones. Por la parte del *backoffice* se pueden gestionar los usuarios, los proyectos, las donaciones, los eventos, las noticias, el SEO, los suscriptores del boletín de noticias y los ficheros subidos por los usuarios.

A continuación se presentan los bocetos que se han diseñado para este TFG:

5.2.1. Frontend

La parte visible de la aplicación y accesible por todo el público.

Página de inicio: la Imagen 16 muestra la maqueta para la página de inicio de la aplicación web:

⁸ <https://balsamiq.com/>

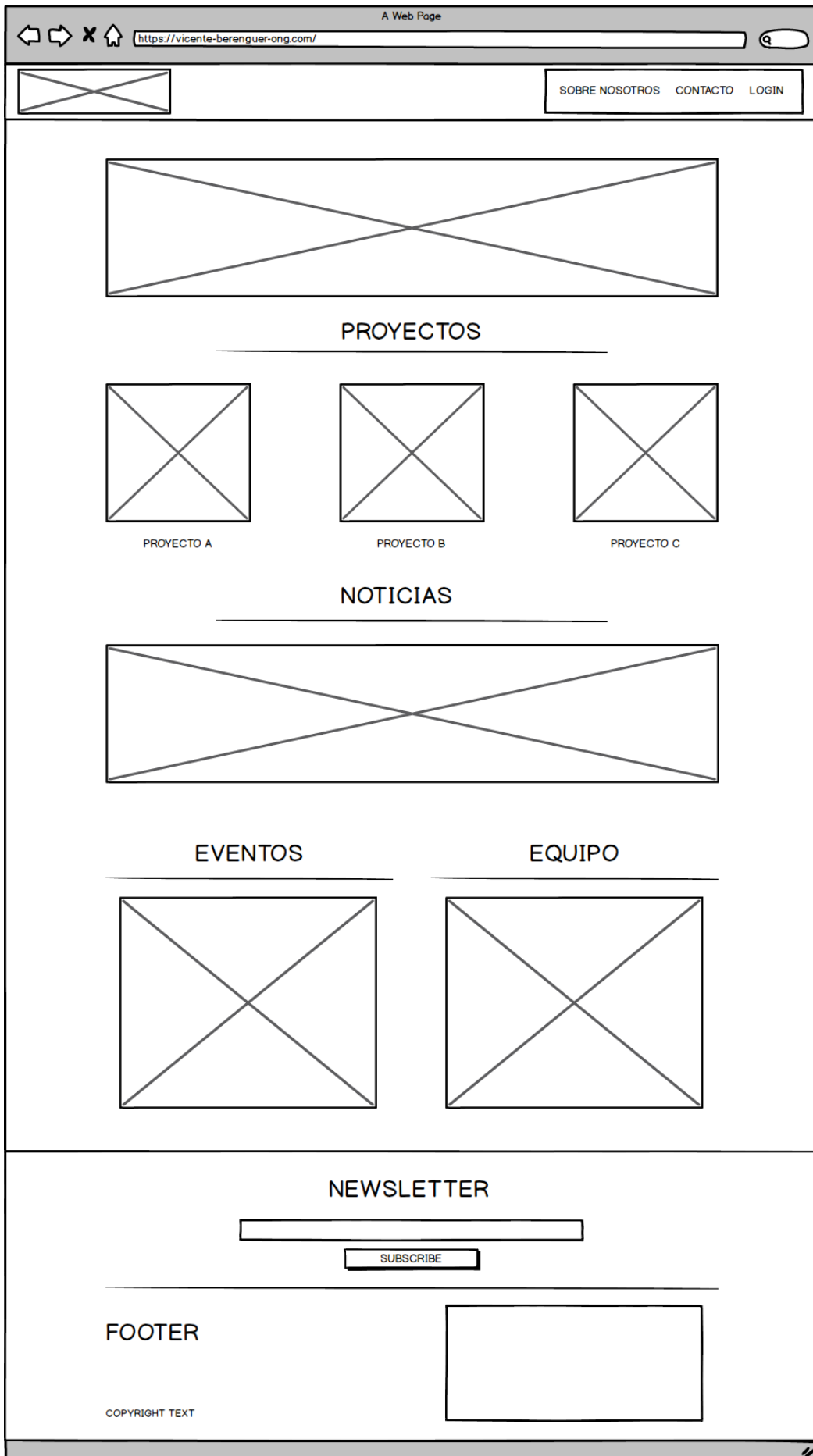


Imagen 16. Página de inicio

Página para registrarse: la Imagen 17 muestra la maqueta para la página de registro de la aplicación web:

The image shows a wireframe of a web browser window. The address bar contains the URL `https://vicente-berenguer-ong.com/`. The browser title is "A Web Page". The page layout includes a navigation menu with links for "SOBRE NOSOTROS", "CONTACTO", and "LOGIN". The main content area is titled "CREAR CUENTA" and contains a registration form with the following fields: "Nombre y apellidos", "Correo electrónico", "Contraseña", and "Fecha de nacimiento" (with a calendar icon). Below the form is a checkbox labeled "¿Quiere recibir novedades?" with a green toggle switch. A "Registrarse" button is positioned below the checkbox, and a link "Ya tengo una cuenta" is located underneath. Below the registration form is a "NEWSLETTER" section with a text input field and a "SUBSCRIBE" button. The footer area is labeled "FOOTER" and contains the text "COPYRIGHT TEXT" and a large empty rectangular box.

Imagen 17. Página para registrarse

Página para iniciar sesión: la Imagen 18 muestra la maqueta para la página de inicio de sesión de la aplicación web:

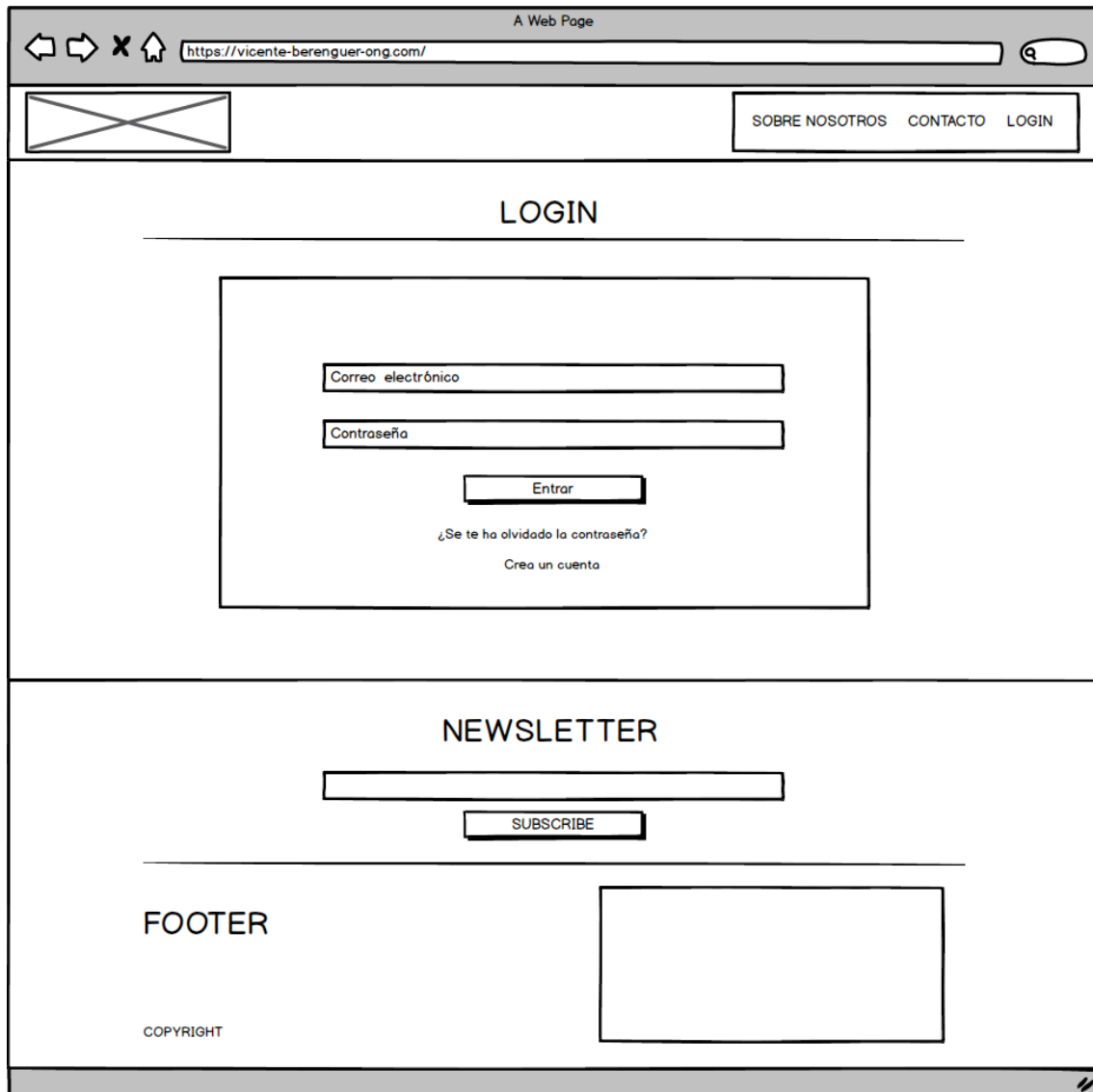


Imagen 18. Página para iniciar sesión

Página para ver los datos personales: la Imagen 19 muestra la maqueta para la página del perfil de la aplicación web:

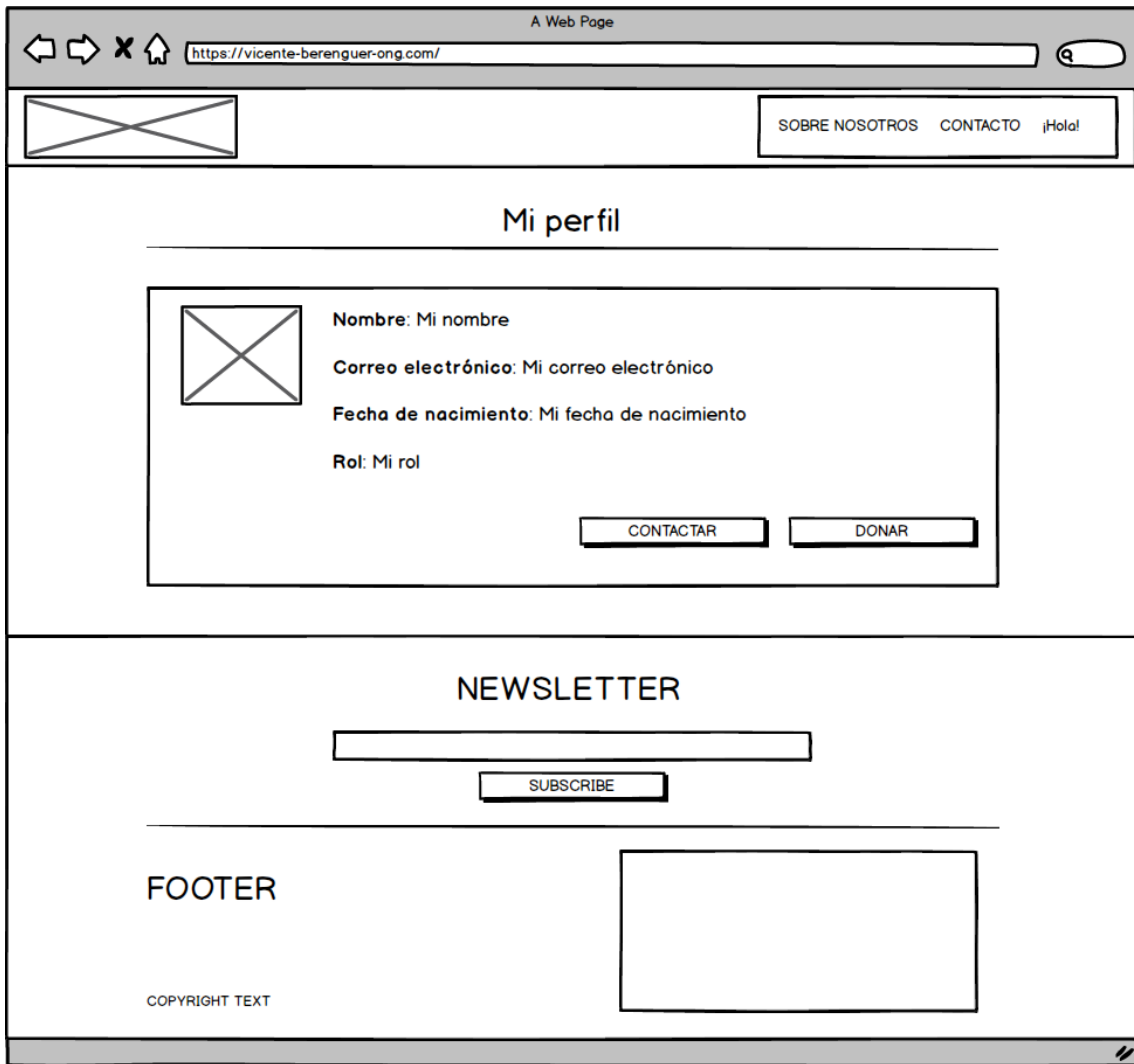


Imagen 19. Página de perfil

Página para ver los detalles del proyecto: la Imagen 20 muestra la maqueta para la página de detalles de los proyectos de la aplicación web:

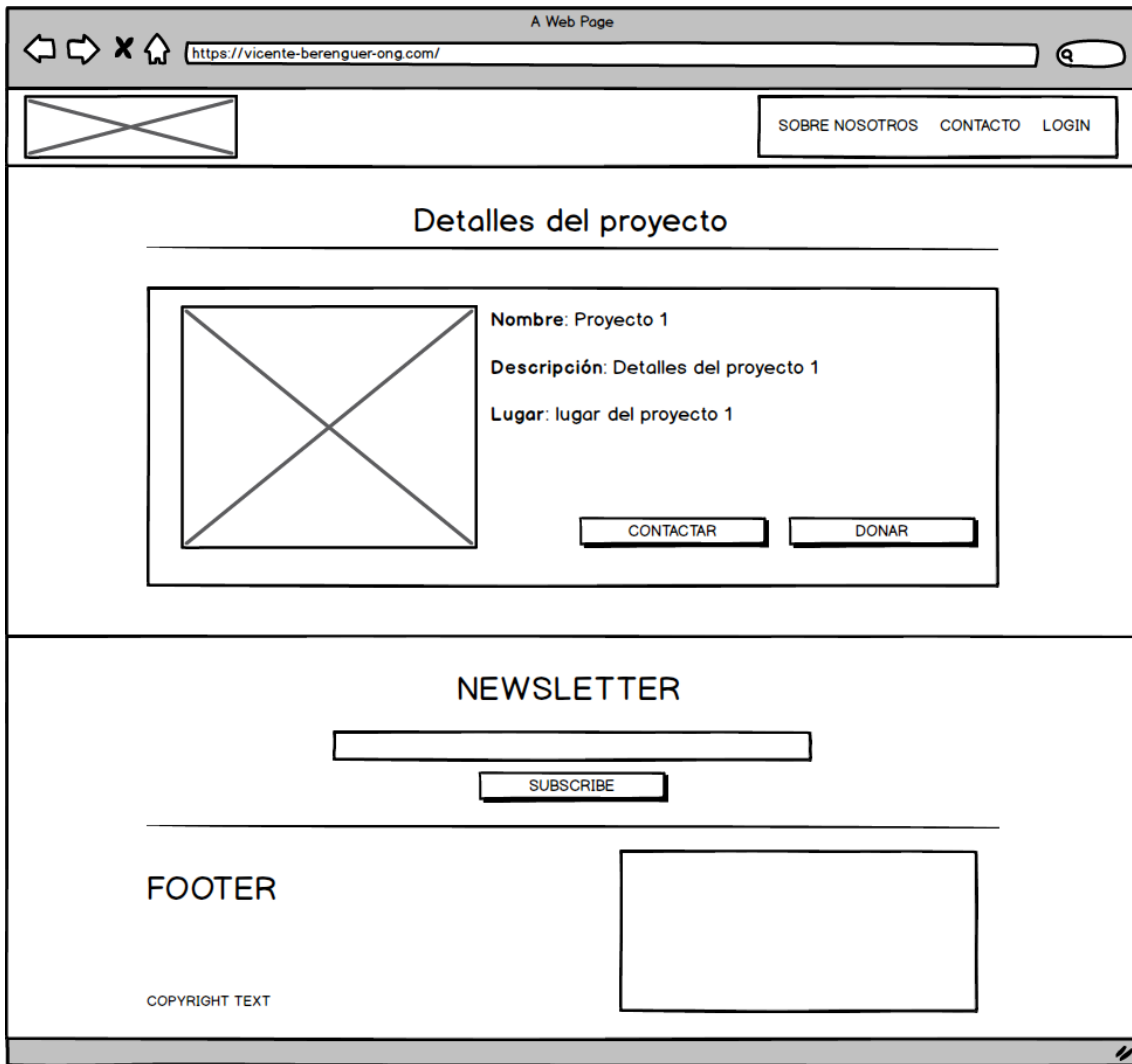


Imagen 20. Página de detalles del proyecto

Página para ver los eventos organizados: la Imagen 21 muestra la maqueta para la página de los eventos de la aplicación web:

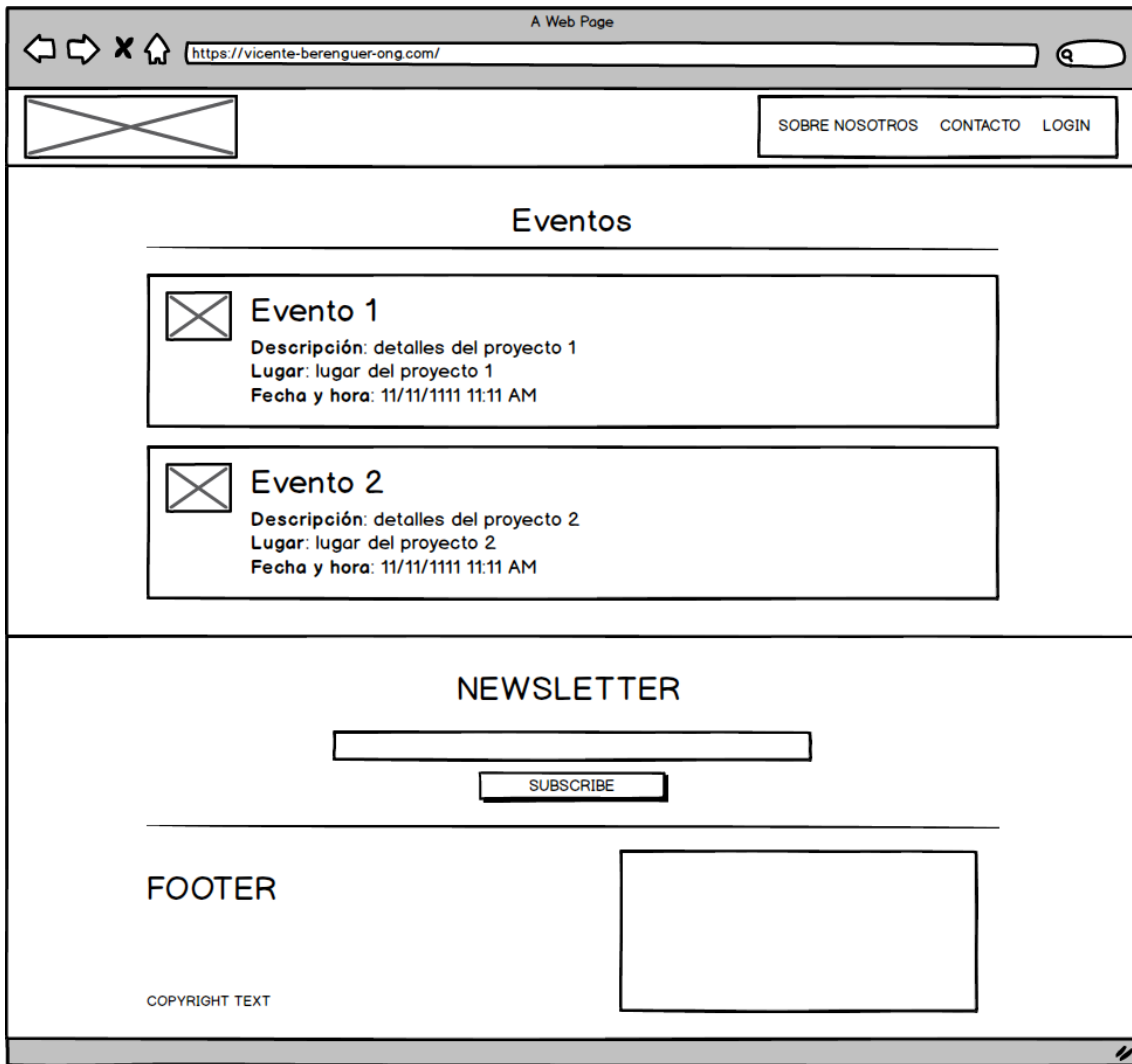


Imagen 21. Página para ver los eventos

Página para ver las noticias de la ONG: la Imagen 22 muestra la página para consultar el listado de las noticias de la ONG.

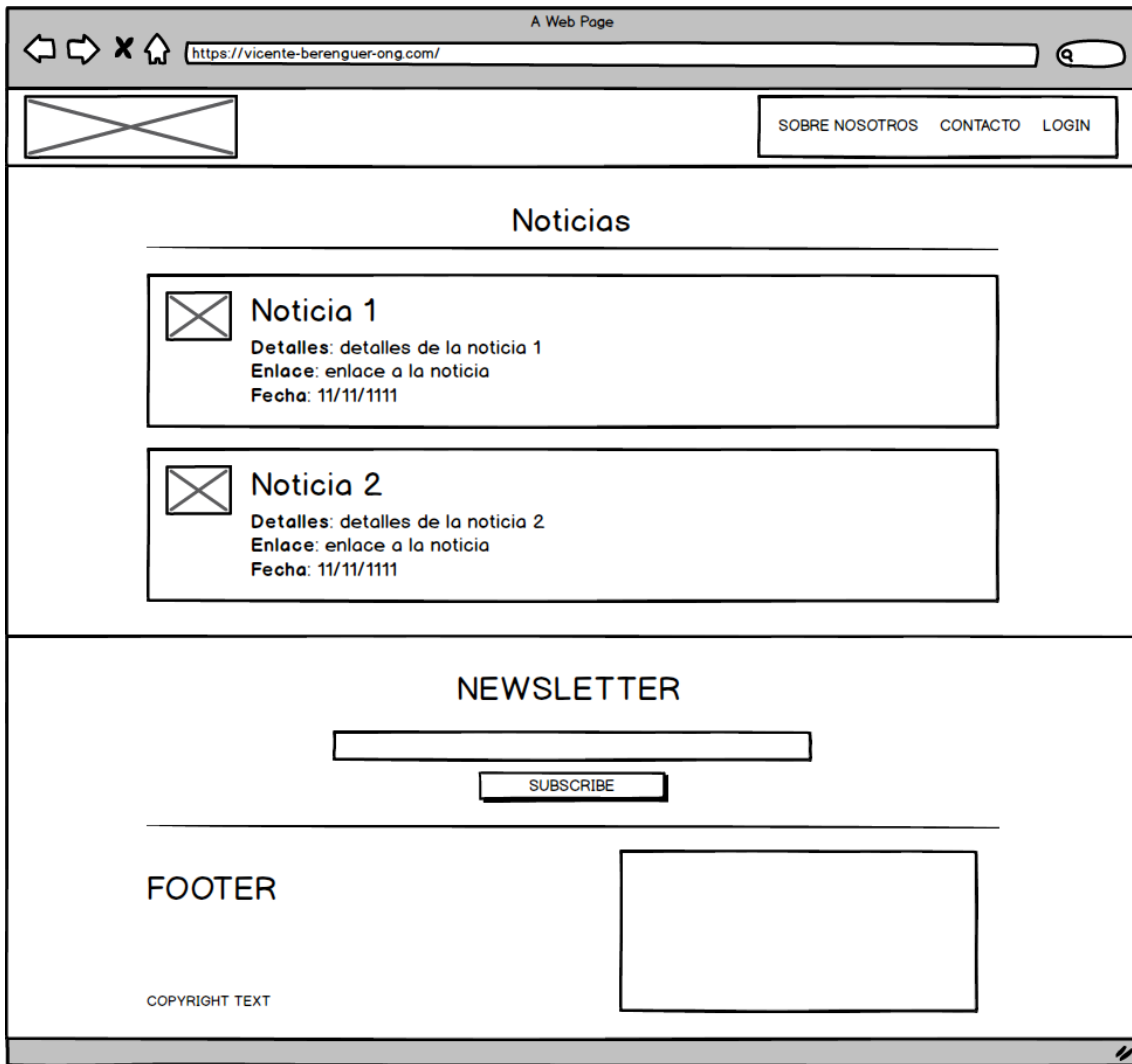


Imagen 22. Página para ver las noticias

Página para hacer donaciones: la Imagen 23 muestra la página para realizar donaciones en la página web:

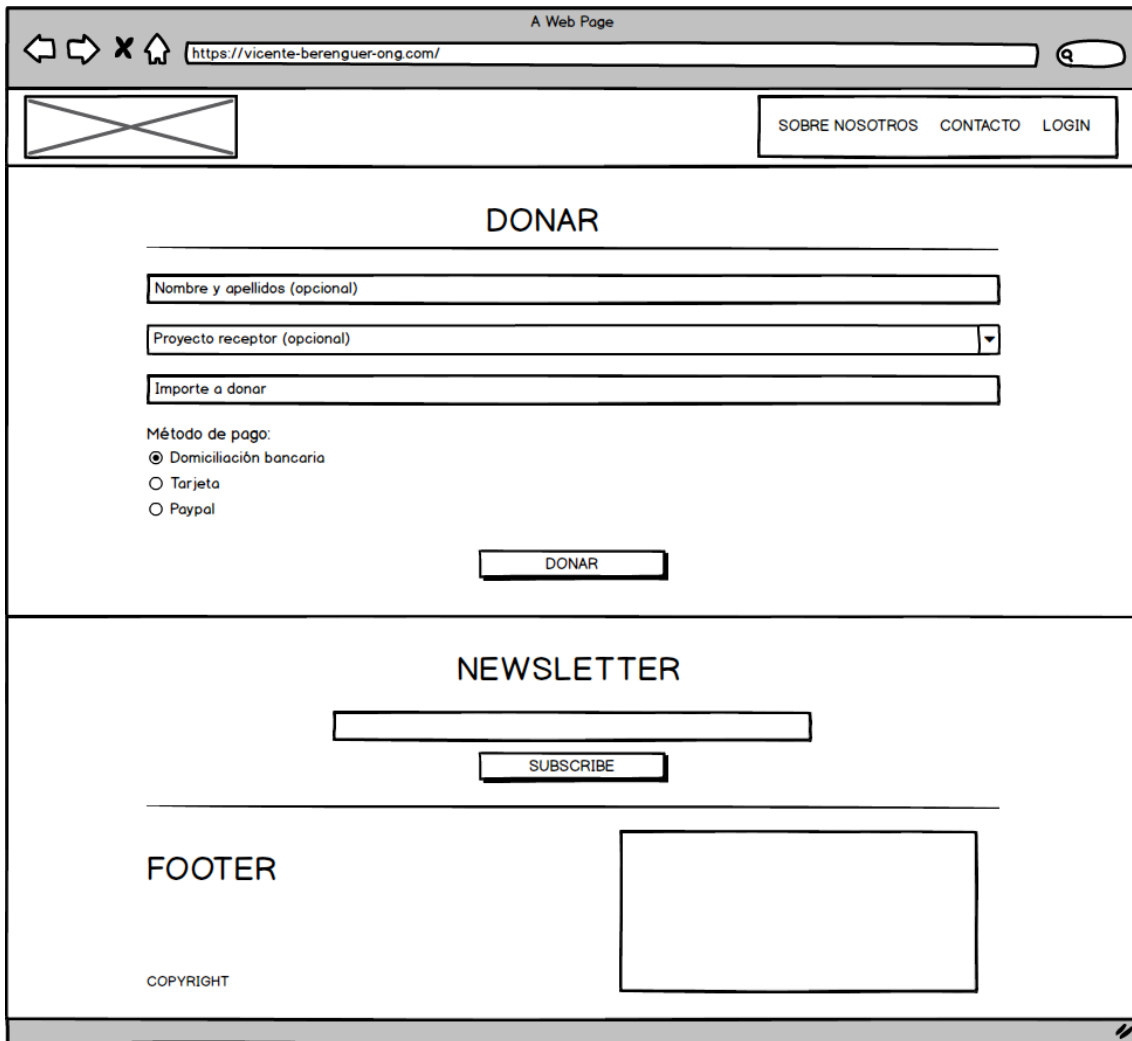


Imagen 23. Página de donación

Página de contacto: la Imagen 24 muestra la página del formulario de contacto:

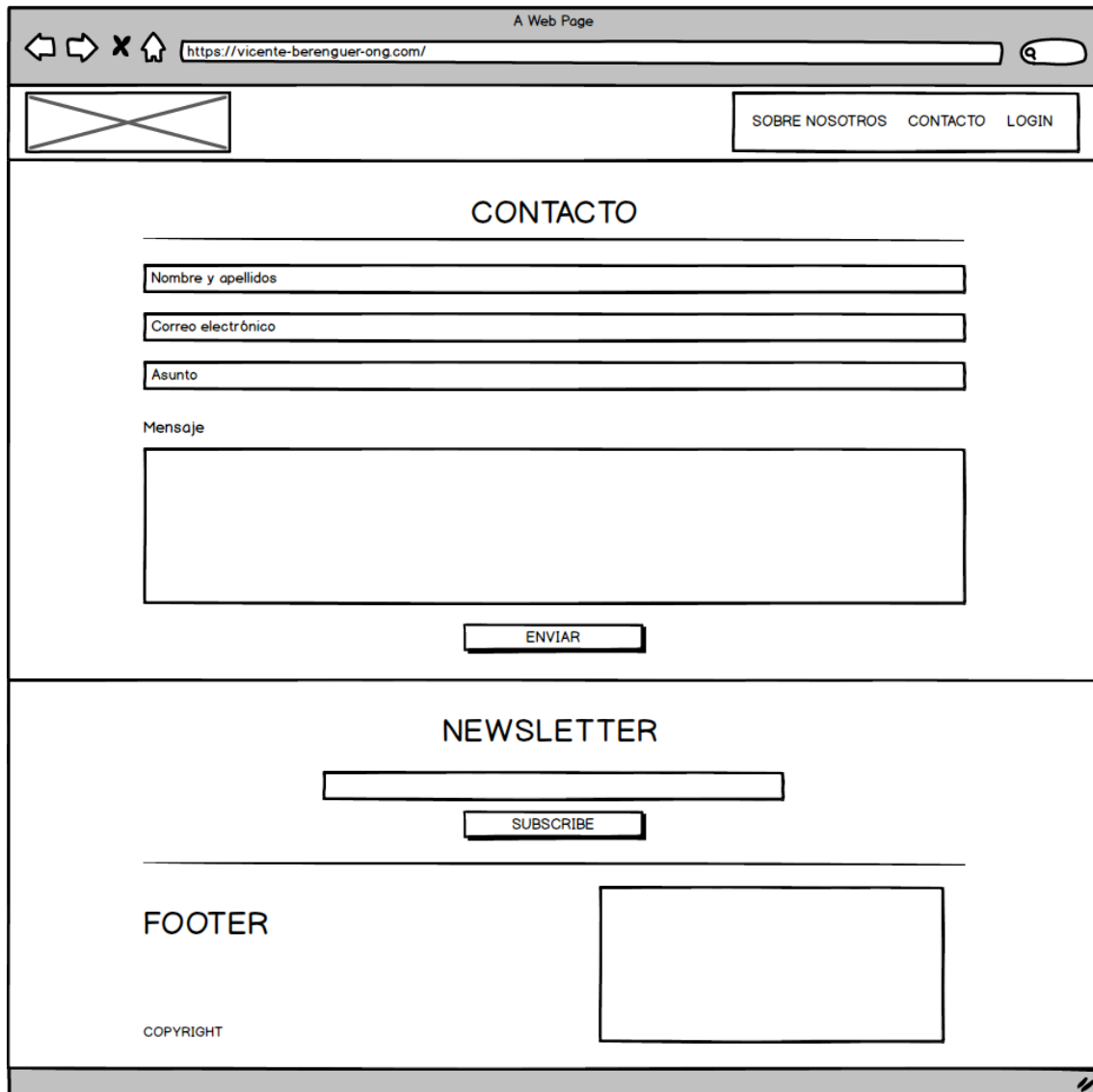


Imagen 24. Página de contacto

5.2.2. *Backoffice*

La parte de la aplicación web para la administración accesible exclusivamente por los empleados de la ONG.

Página principal del *backoffice*: la Imagen 25 muestra la página principal de acceso a las funcionalidades del *backoffice*.

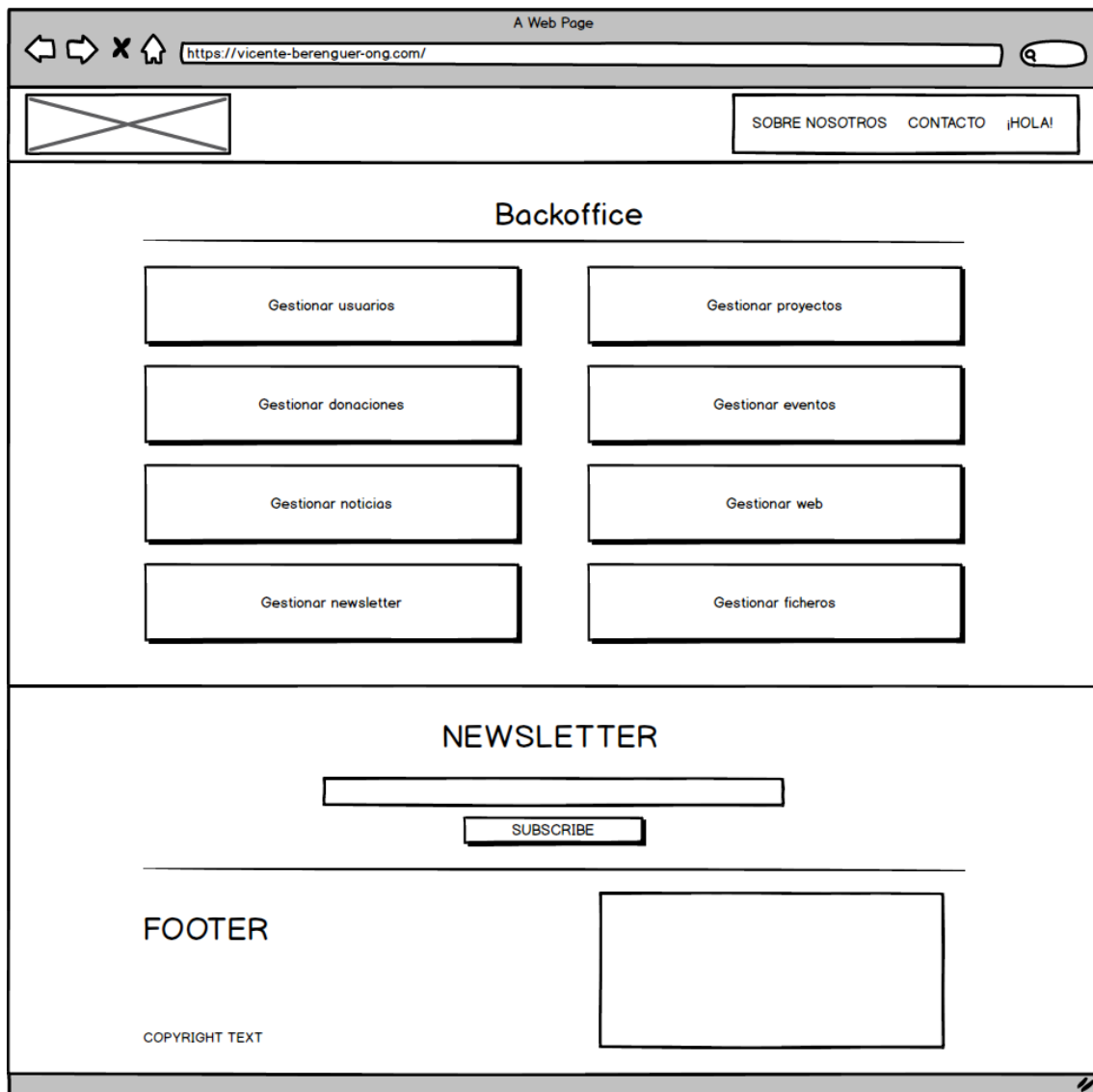


Imagen 25. Página principal del backoffice

Página de gestión de usuarios: la Imagen 26 muestra la página donde se listan y se gestionan los usuarios de la aplicación:



Imagen 26. Página para listar los usuarios

Añadir usuario: la Imagen 27 muestra el formulario para añadir un usuario nuevo en aplicación. Este formulario es accesible desde la página de gestión de usuarios:

The image shows a web browser window with the address bar containing "https://vicente-berenguer-ong.com/". The page has a header with navigation links: "SOBRE NOSOTROS", "CONTACTO", and "¡HOLA!". The main content area is titled "Añadir usuario" and contains a registration form with the following fields: "Nombre y apellidos", "Correo electrónico", "Contraseña", "Tipo de usuario" (a dropdown menu), and "Fecha de nacimiento" (with a calendar icon). Below the form is a toggle switch for "¿Quiere recibir novedades?". At the bottom of the form are two buttons: "Cancelar" and "Añadir". Below the registration form is a "NEWSLETTER" section with a text input field and a "SUBSCRIBE" button. The footer area contains the text "FOOTER" and "COPYRIGHT TEXT" on the left, and a large empty rectangular box on the right.

Imagen 27. Formulario para añadir usuario

Ver detalles del usuario: la Imagen 28 muestra la página para ver la información un usuario específico. Accesible desde la página de gestión de usuarios:

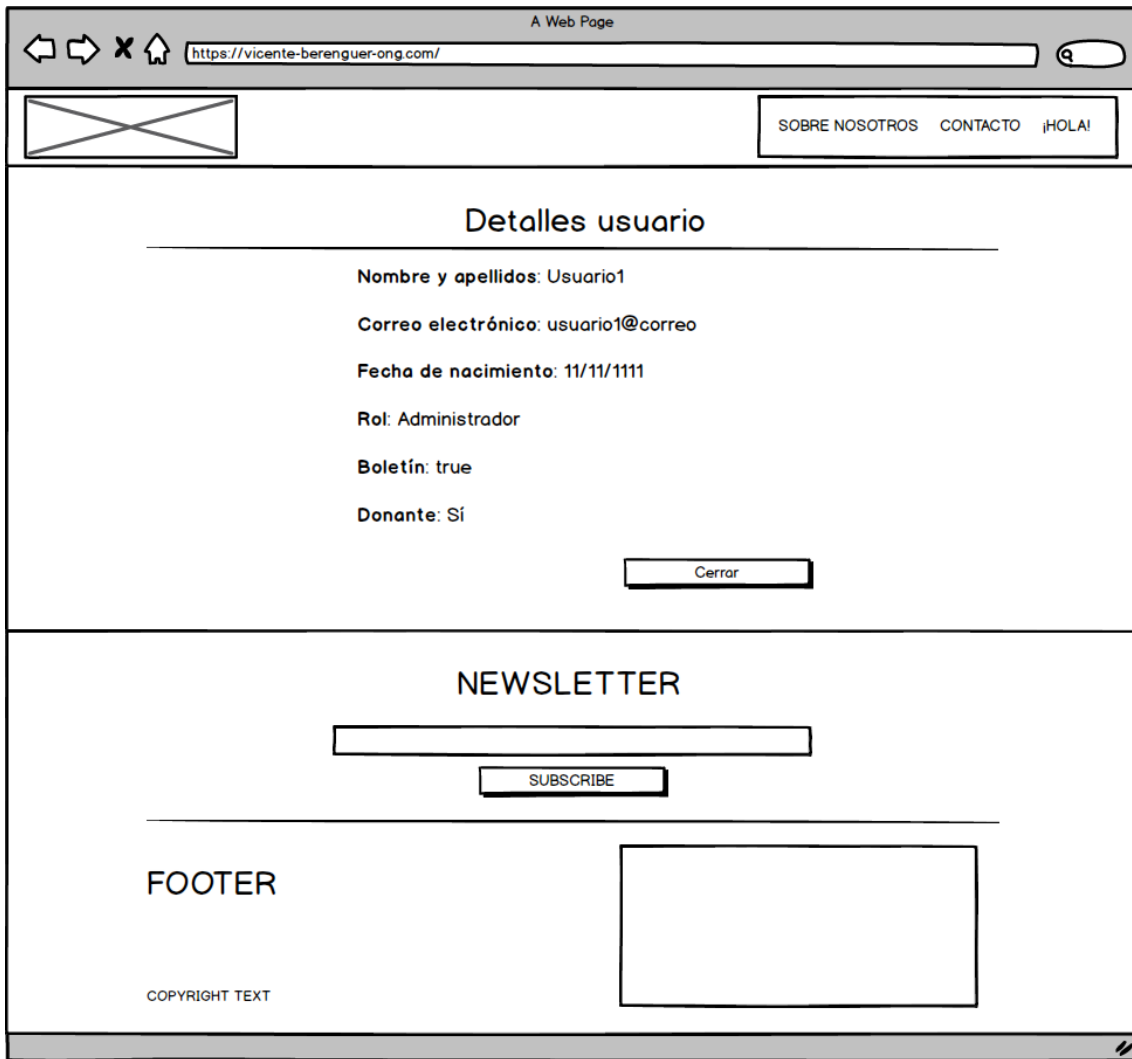


Imagen 28. Página para ver los detalles del usuario

Editar usuario: la Imagen 29 muestra el formulario para editar la información de un usuario específico. Accesible desde la página de gestión de usuarios:

The image shows a web browser window with the address bar containing "https://vicente-berenguer-ong.com/". The page has a header with navigation links: "SOBRE NOSOTROS", "CONTACTO", and "¡HOLA!". The main content area is divided into three sections:

- Actualizar usuario:** A form with the following fields:
 - Text input: "Usuario1"
 - Text input: "usuario1@correo"
 - Text input: "*****"
 - Dropdown menu: "Administrador"
 - Text input: "11/11/1111" with a calendar icon
 - Toggle switch: "¿Quiere recibir novedades?" (checked)
 - Buttons: "Cancelar" and "Actualizar"
- NEWSLETTER:** A text input field and a "SUBSCRIBE" button.
- FOOTER:** The text "FOOTER" and "COPYRIGHT TEXT" on the left, and a large empty rectangular box on the right.

Imagen 29. Formulario para editar un usuario

Borrar usuario: la Imagen 30 muestra la ventana de confirmación al borrar un usuario específico:



Imagen 30. Confirmación para borrar un usuario

Página de gestión de proyectos: la Imagen 31 muestra la página donde se listan y se gestionan todos los proyectos de la ONG:



Imagen 31. Página para listar los proyectos

Añadir proyecto: la Imagen 32 muestra el formulario para añadir un proyecto nuevo. Accesible desde la página de gestión de proyectos:

The image shows a web browser window with the URL <https://vicente-berenguer-ong.com/>. The page has a navigation menu with links for "SOBRE NOSOTROS", "CONTACTO", and "¡HOLA!". The main content area is titled "Añadir proyecto" and contains a form with the following elements:

- Input field for "Nombre"
- Input field for "Descripción"
- Section "Ubicación:" with a map showing a location pin and a highlighted area.
- Button "Subir ficheros"
- Buttons "Cancelar" and "Añadir"

Below the form is a "NEWSLETTER" section with an input field and a "SUBSCRIBE" button. The footer area contains the text "FOOTER" and "COPYRIGHT TEXT" next to a large empty rectangular box.

Imagen 32. Formulario para añadir un proyecto

Ver detalles del proyecto: la Imagen 33 muestra la página donde se puede consultar la información detallada de un proyecto específico. Accesible desde la página de gestión de proyectos:

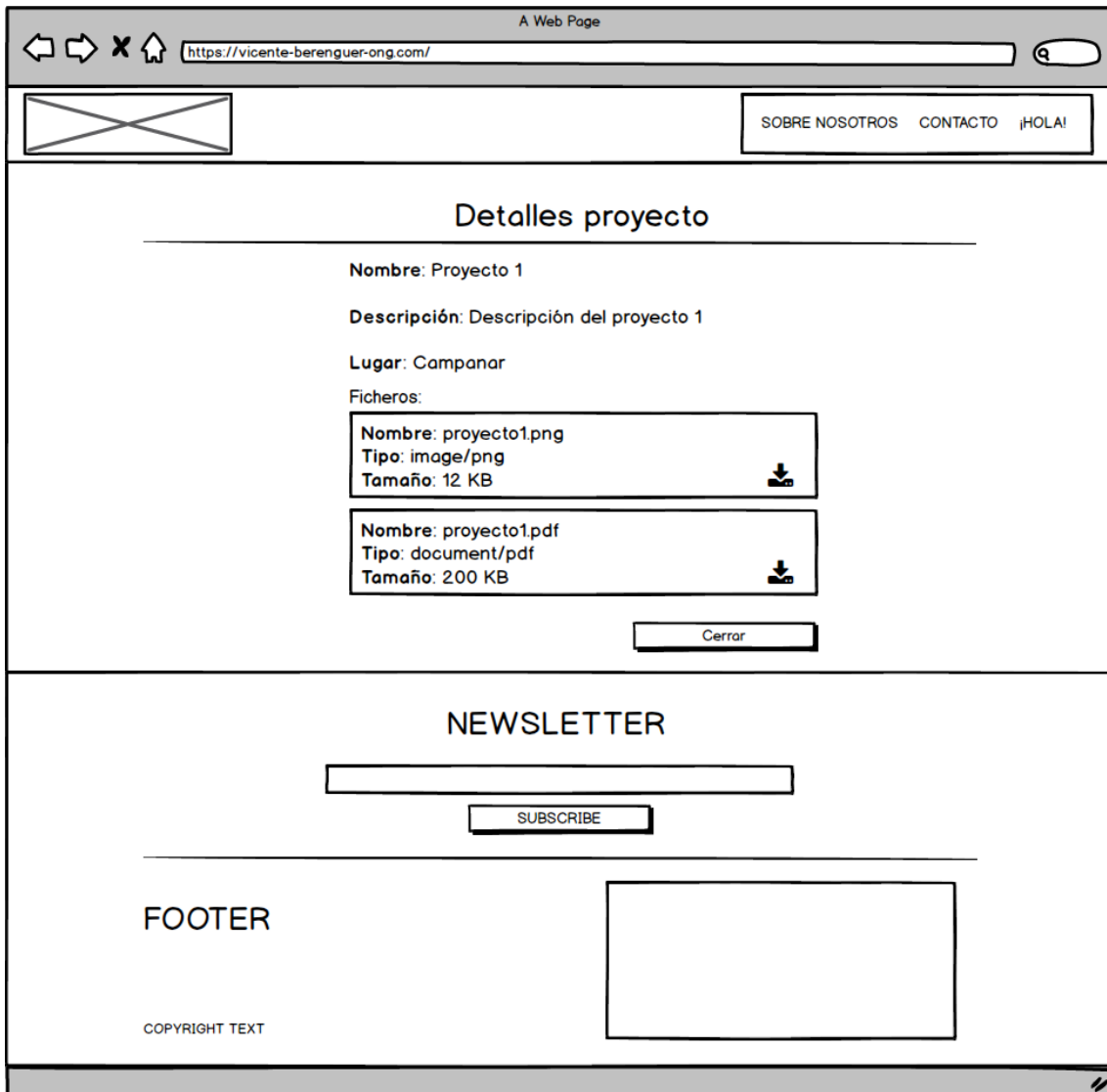


Imagen 33. Página para ver los detalles de un proyecto

Editar proyecto: la Imagen 34 muestra el formulario para editar la información de un proyecto ya existente. Accesible desde la página de gestión de proyectos:

A Web Page

https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Actualizar proyecto

Proyecto 1

Descripción del proyecto 1

Campanar

Ficheros:

Nombre: proyecto1.png
Tipo: image/png
Tamaño: 12 KB

Nombre: proyecto1.pdf
Tipo: document/pdf
Tamaño: 200 KB

Cancelar Actualizar

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 34. Formulario para editar un proyecto

Borrar proyecto: la Imagen 35 muestra la ventana de confirmación para borrar un proyecto:



Imagen 35. Confirmación para borrar un proyecto

Añadir/quitar voluntario: la Imagen 36 muestra la página para añadir o quitar voluntarios del proyecto:

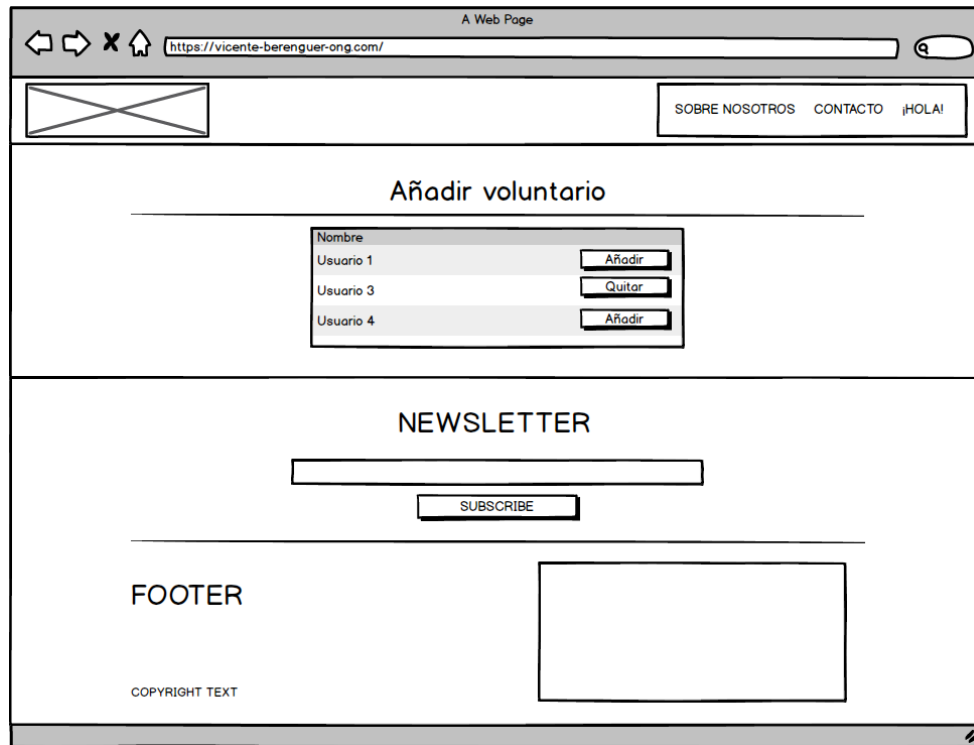


Imagen 36. Página para añadir/quitar voluntarios

Página de gestión de donaciones: la Imagen 37 muestra la página donde se listan y se gestionan las donaciones realizadas a la ONG:

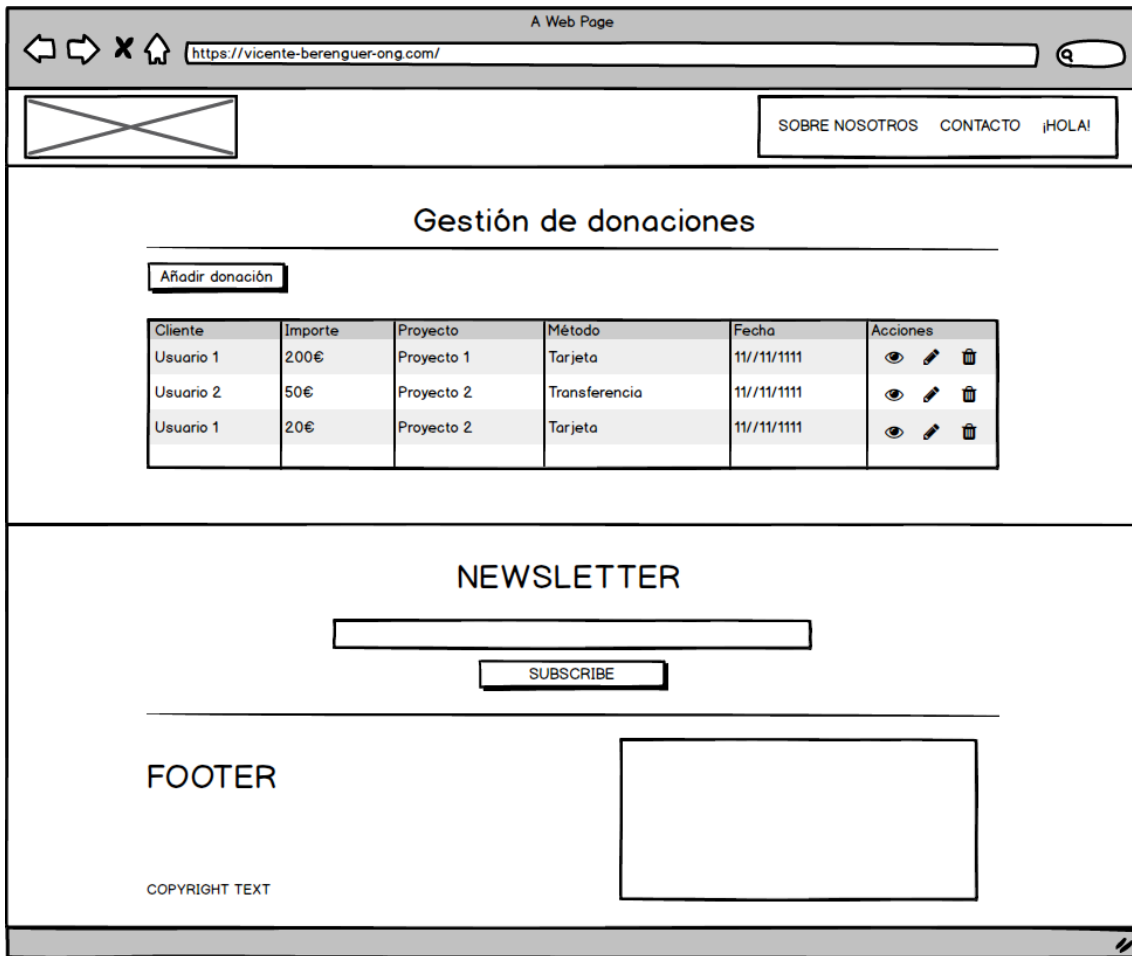


Imagen 37. Página para listar las donaciones

Añadir donación: la Imagen 38 muestra el formulario para añadir una donación nueva. Accesible desde la página de gestión de donaciones:

A Web Page
https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Añadir donación

Nombre cliente

Importe

Proyecto..

Método de pago

Fecha del pago

Cancelar Añadir

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 38. Formulario para añadir una donación

Ver detalles de la donación: la Imagen 39 muestra la página para ver la información detallada de una donación específica. Accesible desde la página de gestión de donaciones:

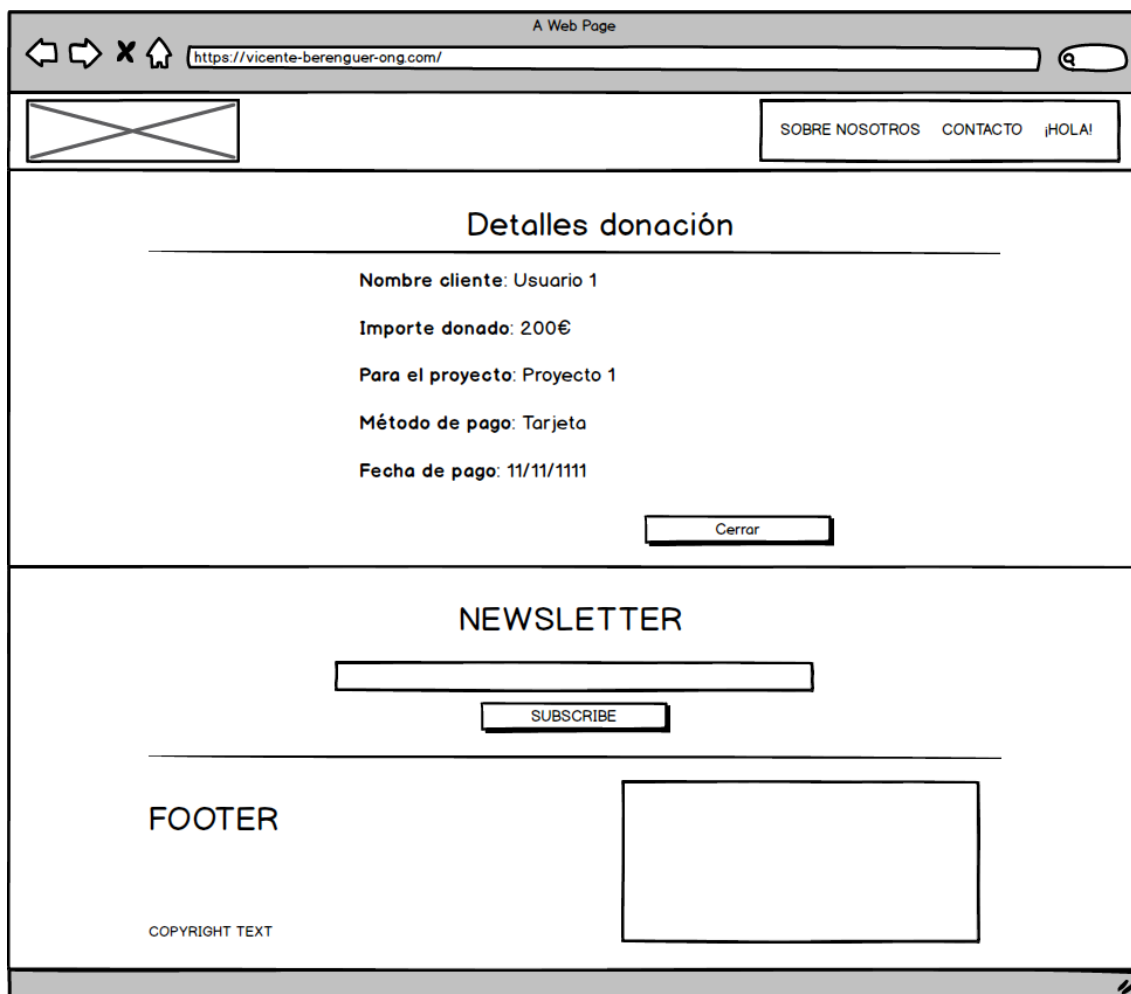


Imagen 39. Página para ver los detalles de una donación

Editar donación: la Imagen 40 muestra el formulario para editar la información de una donación ya existente. Accesible desde la página de gestión de donaciones:

A Web Page

https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Actualizar donación

Usuario 1

200€

Proyecto 1

Tarjeta

11/11/1111

Cancelar Actualizar

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 40. Formulario para editar una donación

Borrar donación: la Imagen 41 muestra la ventana de confirmación para borrar una donación:



Imagen 41. Confirmación para borrar una donación

Página de gestión de eventos: la Imagen 42 muestra la página donde se listan y se gestionan los eventos de la ONG:

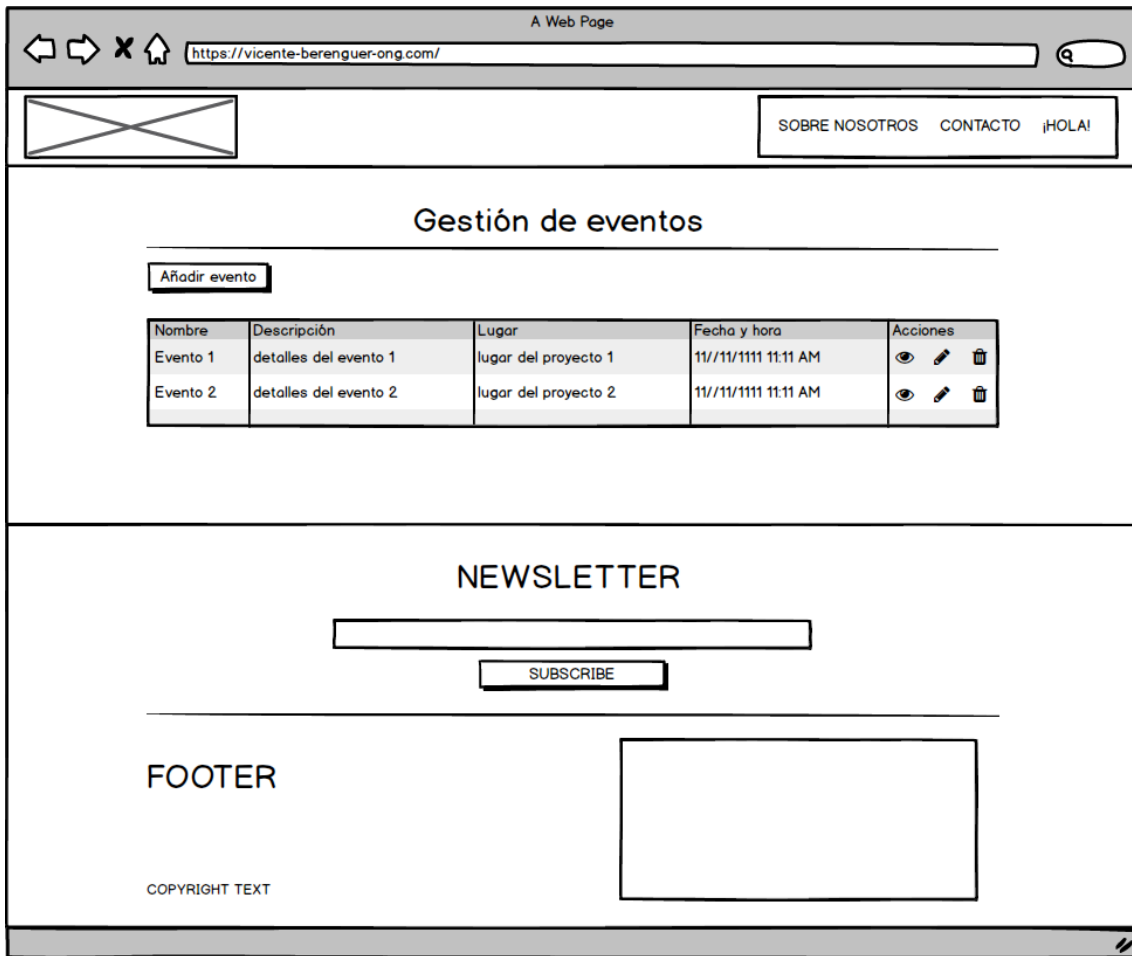


Imagen 42. Página para listar los eventos

Añadir evento: la Imagen 43 muestra el formulario para añadir un evento nuevo en la ONG. Accesible desde la página de gestión de eventos:

A Web Page
https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Añadir evento

Nombre del evento

Descripción del evento

Lugar del evento

Fecha del evento

Horas del evento:

6:00
AM
PM

CANCEL OK

Cancelar Añadir

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 43. Formulario para añadir un evento

Ver detalles del evento: la Imagen 44 muestra la página donde se pueden consultar los datos de un evento en concreto. Accesible desde la página de gestión de eventos:

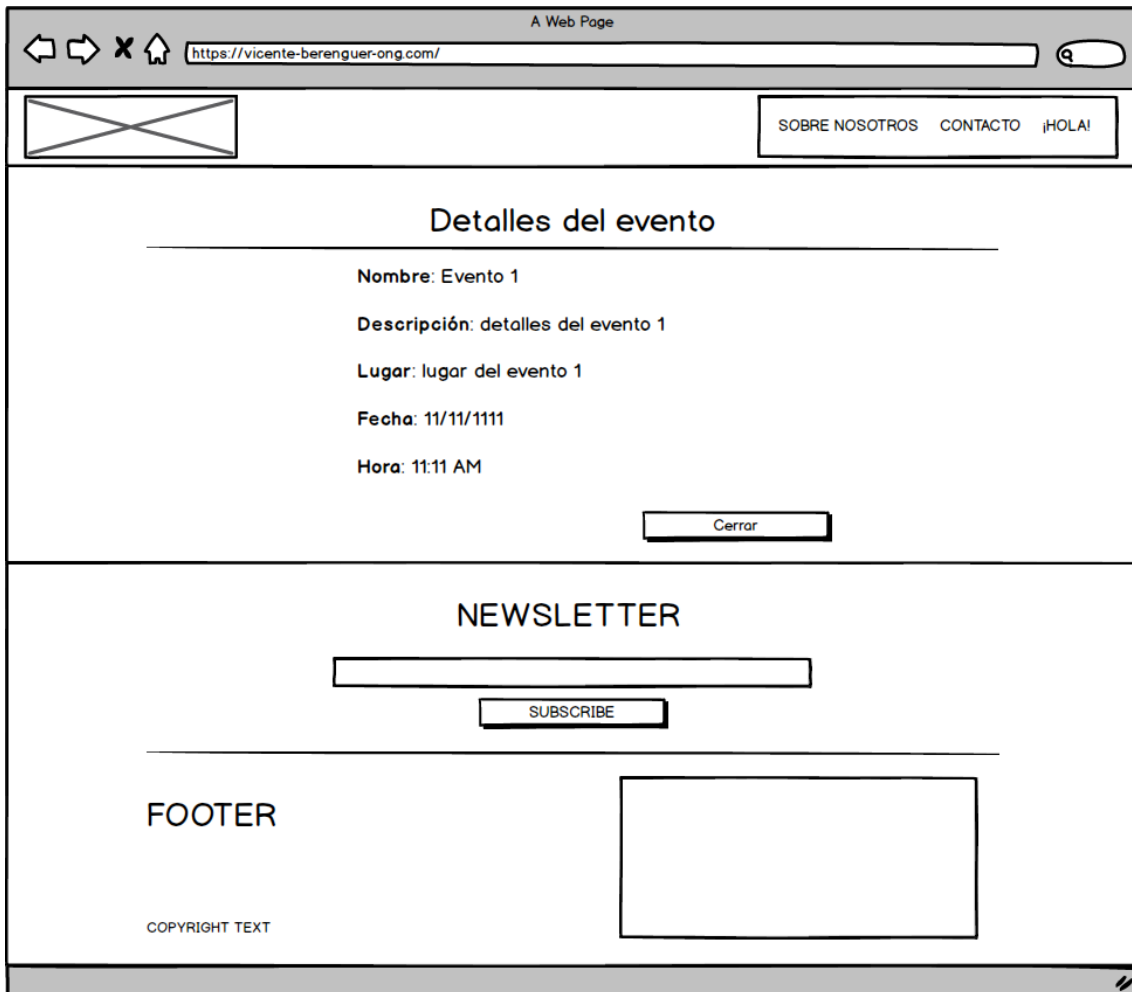


Imagen 44. Página para ver los detalles de un evento

Editar evento: la Imagen 45 muestra el formulario para editar la información de un evento existente. Accesible desde la página de gestión de eventos:

A Web Page

https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Actualizar evento

Evento 1

detalles del evento 1

lugar del evento 1

11/11/1111

Horas del evento:

6:00 AM PM

CANCEL OK

Cancelar Actualizar

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 45. Formulario para editar un evento

Borrar evento: la Imagen 46 muestra la ventana de confirmación para borrar un evento:



Imagen 46. Confirmación para borrar un evento

Página de gestión de noticias: la Imagen 47 muestra la página donde se listan y se gestionan todas las noticias de la ONG:



Imagen 47. Página para listar las noticias

Añadir noticia: la Imagen 48 muestra el formulario para añadir una noticia nueva. Accesible desde la página de gestión de noticias:

A Web Page
https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Añadir noticia

Nombre de la noticia

Descripción de la noticia

Enlace a la noticia

Cancelar Añadir

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 48. Formulario para añadir una noticia

Ver detalles de la noticia: la Imagen 49 muestra la página para ver la información detallada de una noticia específica. Accesible desde la página de gestión de noticias:

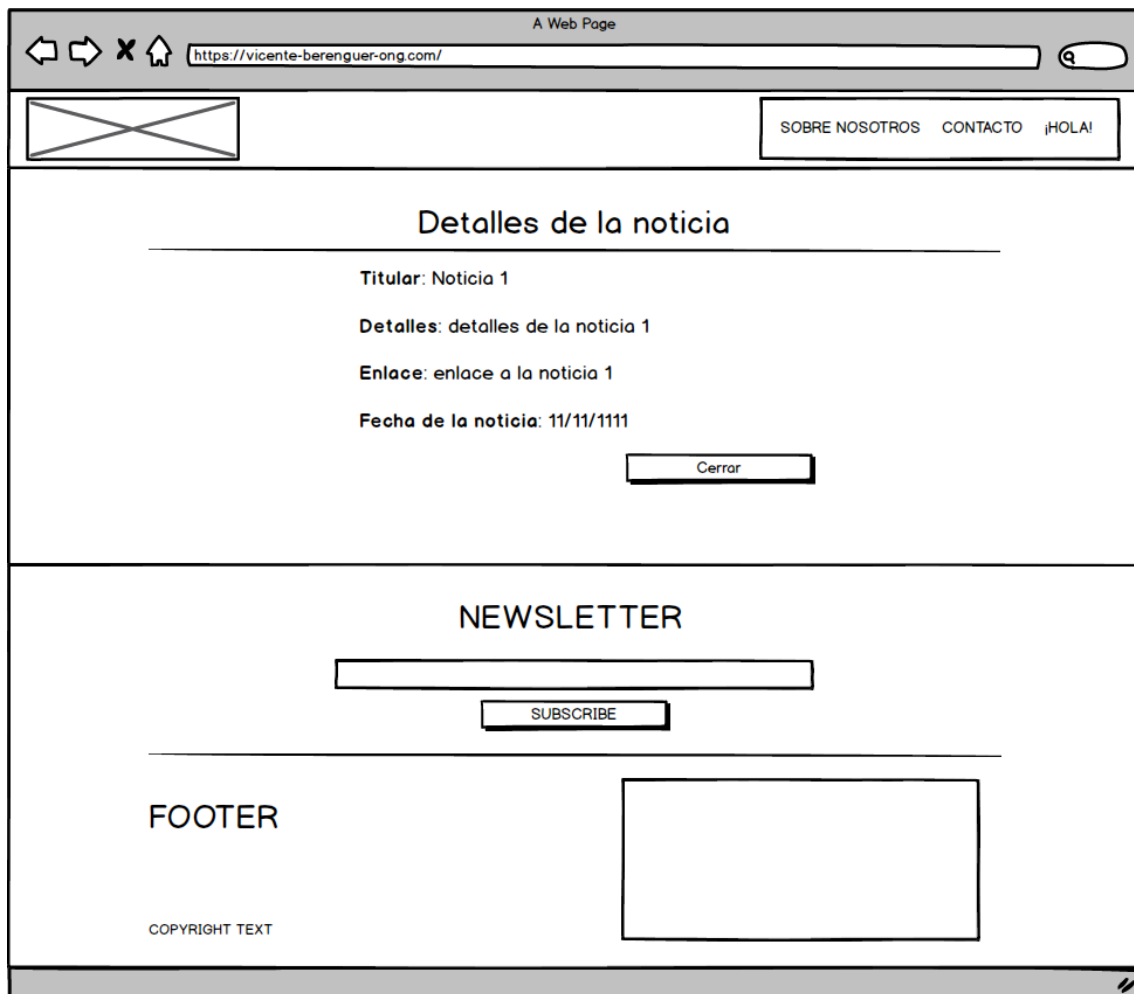


Imagen 49. Página para ver los detalles de una noticia

Editar noticia: la Imagen 50 muestra el formulario para actualizar la información de una noticia existente. Accesible desde la página de gestión de noticias:

A Web Page

https://vicente-berenguer-ong.com/

SOBRE NOSOTROS CONTACTO ¡HOLA!

Actualizar noticia

Noticia 1

detalles de la noticia 1

enlace a la noticia 1

Cancelar Actualizar

NEWSLETTER

SUBSCRIBE

FOOTER

COPYRIGHT TEXT

Imagen 50. Formulario para editar una noticia

Borrar noticia: la Imagen 51 muestra la ventana para confirmar el borrado de una noticia:



Imagen 51. Confirmación para borrar una noticia

Página para gestionar información SEO: la Imagen 52 muestra el formulario para actualizar la información SEO de la aplicación web. Accesible desde la página principal del *backoffice*:

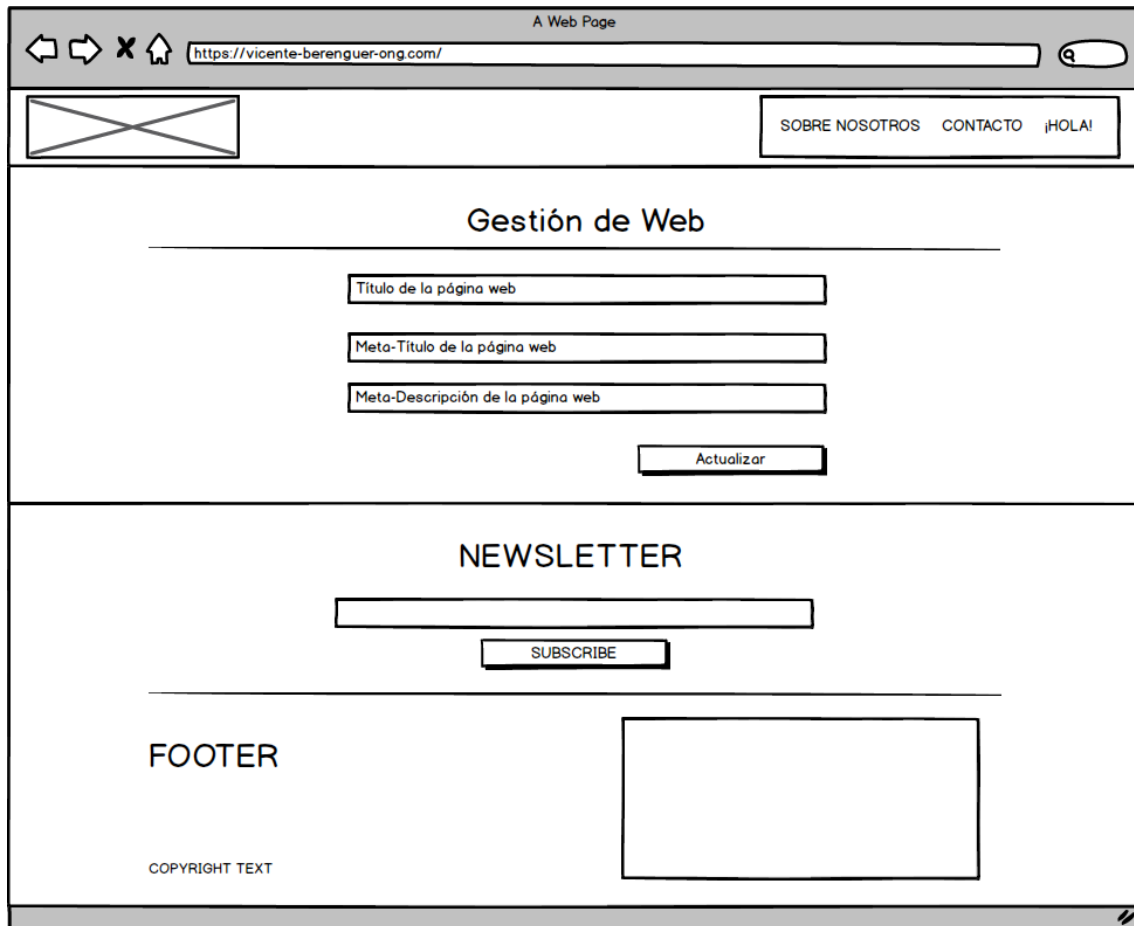


Imagen 52. Página para actualizar el SEO

Página para listar suscriptores del boletín: la Imagen 53 muestra la página donde se listan los suscriptores del boletín de noticias. Accesible desde la página principal del *backoffice*:

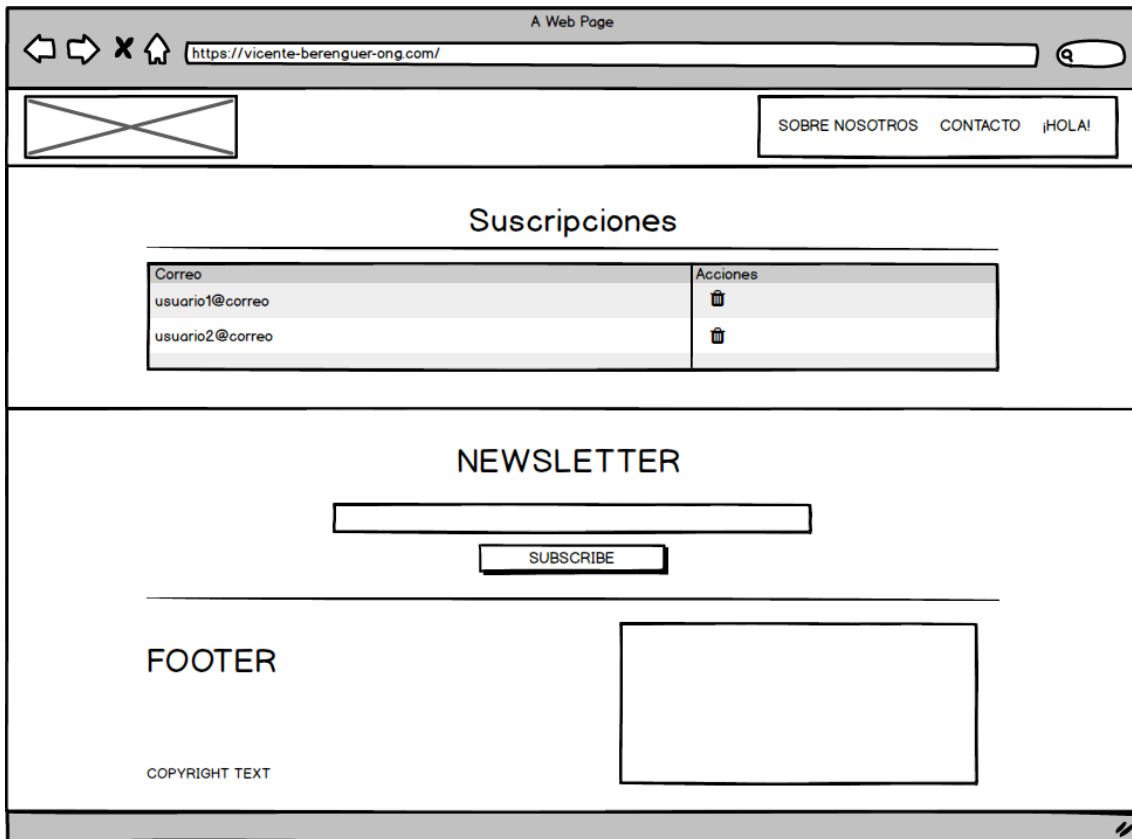


Imagen 53. Página para listar los suscriptores

Borrar suscriptor: la Imagen 54 muestra la ventana de confirmación para borrar un suscriptor de la lista de avisos:

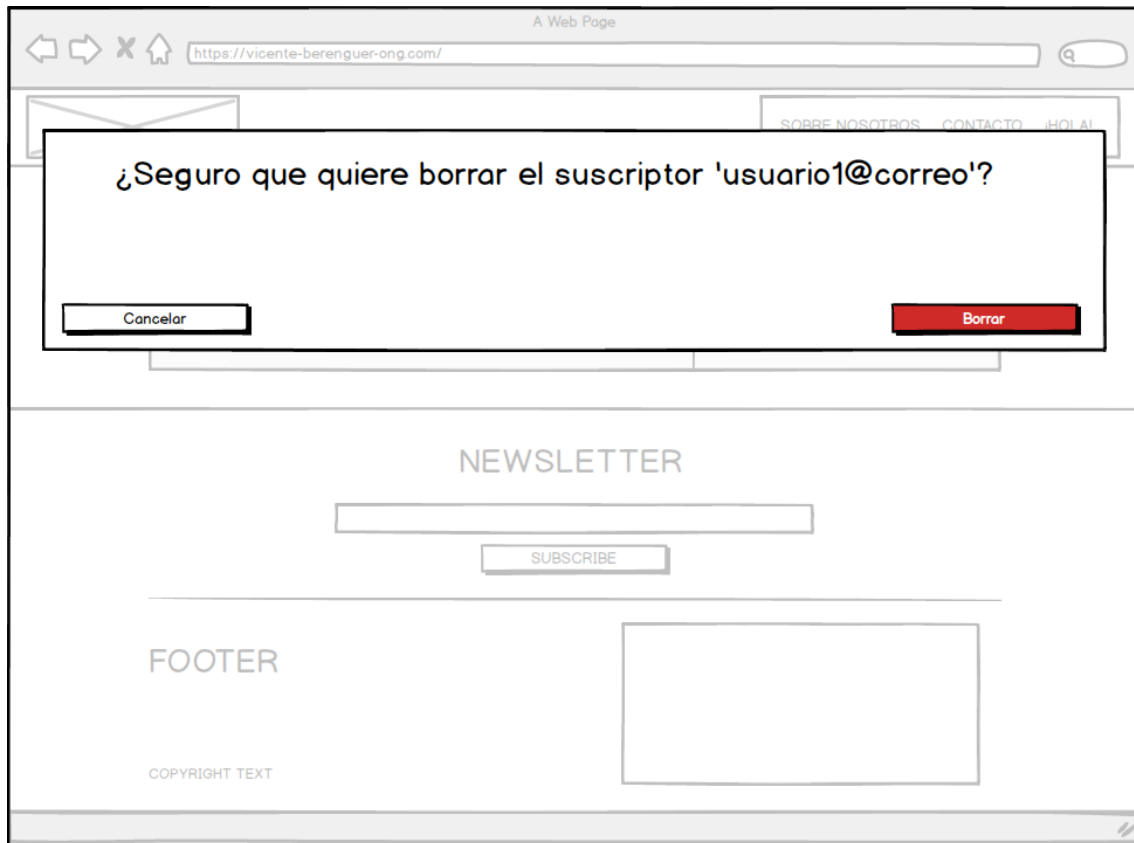


Imagen 54. Confirmación para borrar un suscriptor

Página para ver los ficheros subidos: la Imagen 55 muestra la página donde se listan todos los ficheros subidos a la aplicación web. Accesible desde la página principal del *backoffice*:

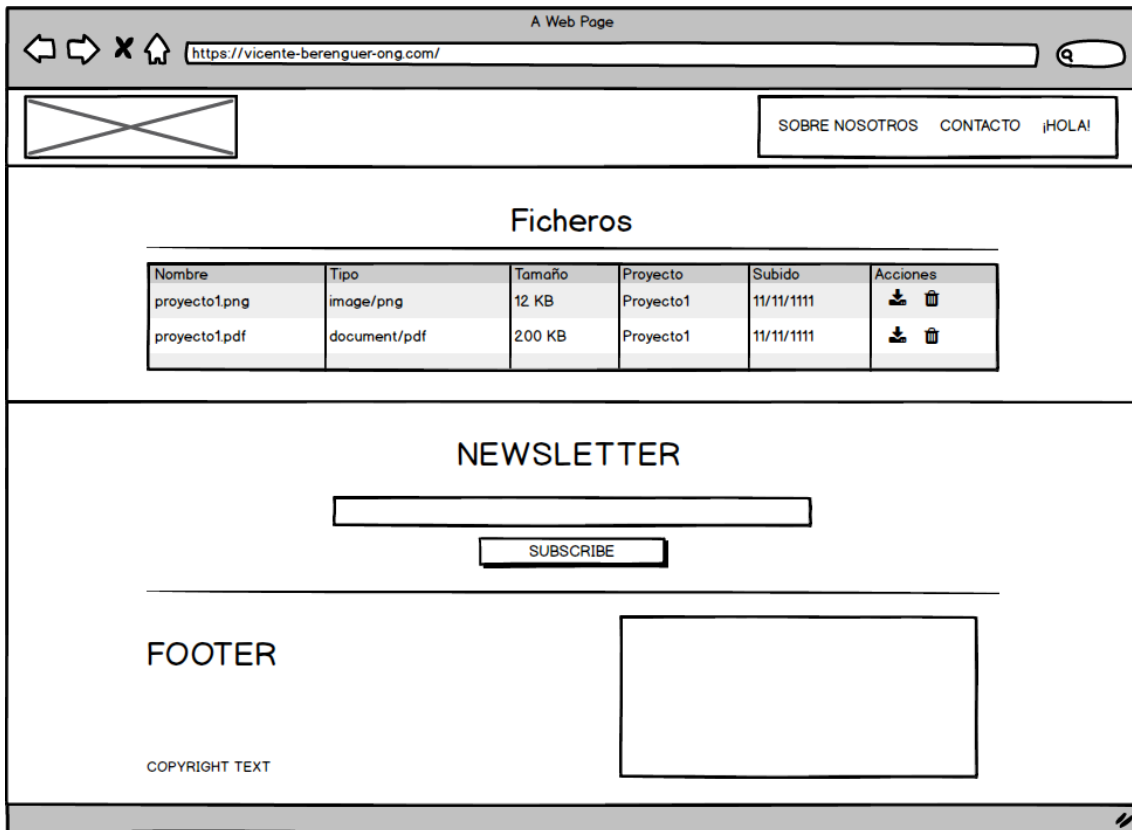


Imagen 55. Página para listar los ficheros

Borrar fichero: la Imagen 56 muestra la ventana de confirmación para borrar un fichero:



Imagen 56. Confirmación para borrar un fichero

6. Implementación

Al llegar a esta fase está bastante claro lo que se quiere desarrollar (especificación de requisitos y análisis), y cómo va a estar organizado el *software*, tanto conceptualmente como la forma en que se mostrará la información al usuario final (diseño). Ahora se procederá a explicar la implementación de la solución propuesta. Para el desarrollo de cualquier *software* es necesario hacer uso de diferentes tecnologías de tal forma que se integren correctamente y se pueda obtener un producto de calidad.

6.1. Tecnologías

La elección de las tecnologías de desarrollo depende del proyecto a implementar y del equipo y/o empresa que lo va a realizar ya que cada tecnología ofrece unas ventajas e inconvenientes y dependiendo de las necesidades del proyecto se elige una herramienta u otra. Para el desarrollo de este trabajo se consideraron tres conjuntos diferentes de tecnologías que son:

- **MEAN [4]:** este conjunto de tecnologías de código abierto está basado en *JavaScript* [5]. Está compuesto por MongoDB, Express, Angular y NodeJS. Estas cuatro tecnologías diferentes cubren todos los aspectos importantes de una aplicación como son la persistencia de datos, el servidor, la interfaz de usuario y un entorno para ejecutar todos los componentes de la misma. La Imagen 57 muestra los logotipos de las tecnologías que componen el *stack* **MEAN**:



Imagen 57. MEAN stack

- **LAMP [6]:** un conjunto de tecnologías de código abierto y muy utilizado ya que está compuesto por Linux OS, Apache web server, base de datos MySQL y PHP.

La Imagen 58 muestra los logotipos de las tecnologías que componen el *stack LAMP*:



Imagen 58. LAMP stack

- **MERN [7]**: es igual que el *stack MEAN* con la única diferencia de que en este conjunto de tecnologías para trabajar en el lado del cliente se utiliza ReactJS en vez de Angular. La Imagen 59 muestra los logotipos de las tecnologías que componen el *stack MERN*:

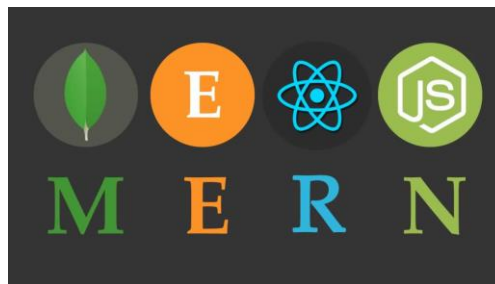


Imagen 59. MERN stack

En este trabajo de fin de grado se ha utilizado el *stack MEAN* para el desarrollo de la aplicación web. A continuación, se explica por qué se ha elegido el *stack MEAN* para este proyecto mostrando las ventajas y desventajas, se explica con detalle lo que es *MEAN*, las tecnologías que lo componen y cómo funcionan dichas tecnologías.

6.1.1. Elección del *stack MEAN*

Se ha elegido el *stack MEAN* para el desarrollo de este trabajo atendiendo a las siguientes razones [8]:

- Las tecnologías por las que está compuesto el *stack MEAN* son modernas y se han adaptado a las necesidades actuales del desarrollo web.

- Todas las tecnologías están basadas en *JavaScript* lo que proporciona ventajas como: menor tiempo de aprendizaje y menos cambios de contexto para trabajar con diferentes partes de la aplicación.
- NodeJS es un entorno de ejecución basado en *JavaScript*, lo cual permite que la ejecución y la interacción de los componentes de la aplicación sea más eficiente, simplifica la implementación de la capa del servidor y reduce el tiempo de puesta en marcha del entorno servidor.
- Agiliza el proceso de creación de aplicaciones web permitiendo crearlas de forma rápida y eficaz.
- Permite crear aplicaciones modulares, robustas y fáciles de mantener de forma más sencilla.
- MongoDB es una base de datos eficiente, ligera y flexible que ayuda a trabajar con una cantidad elevada de datos y está pensada para integrarse en la nube.
- El uso del formato JSON es otra de las ventajas ya que se pueden intercambiar datos entre el lado servidor y cliente sin la necesidad de utilizar librerías externas y también facilita trabajar con APIs externas.
- Es una buena opción para aplicaciones que muestran la información o los cambios en tiempo real.
- Todas las tecnologías son multiplataforma y por consiguiente, se puede trabajar desde cualquier lugar sin limitaciones de *software* o *hardware*.
- El uso de la memoria interna del *hardware* es bajo.
- Ayuda a ahorrar una cantidad considerable de programación básica gracias a los *frameworks* y permite centrarse en la implementación de las funcionalidades de la aplicación.
- *MEAN stack* hace que el código sea isomorfo, es decir, el código generado es utilizable tanto en el lado del cliente como el lado servidor. Este aspecto ayuda a repartir la carga del servidor con el navegador web permitiendo ejecutar ciertas funcionalidades del servidor en el lado del cliente lo cual permite reducir la cantidad de peticiones tratadas por el servidor.
- Se dispone de una buena documentación *online* para cada tecnología.

Las desventajas a las que se afronta al utilizar este *stack* son:

- MongoDB es una opción viable cuando se quiere manejar una gran cantidad de datos pero es posible que en algunos casos no sea la opción ideal ya que se pierde algo de control sobre los datos si no se trabaja correctamente.
- No hay pautas generales o específicas para programar en *JavaScript*.



- La separación entre el servidor y la capa lógica no está claramente definida.

6.1.2. *MEAN*

MEAN stack es un conjunto de tecnologías *JavaScript* de código abierto que se utilizan para desarrollar aplicaciones web. Esto quiere decir que desde la parte del cliente al servidor y desde el servidor hasta la base de datos, todo el proceso está basado en *JavaScript*. *MEAN* es un *kit* completo para implementar aplicaciones web rápidas de ejecutar y robustas. Es también una solución muy buena para construir páginas y aplicaciones web dinámicas de forma rápida.

Tal y como indican las siglas, *MEAN* está compuesto por cuatro tecnologías diferentes [9]:

- **MongoDB**: sistema de base de datos NoSQL.
- **Express**: es un *framework* utilizado para desarrollar aplicaciones web en NodeJS.
- **Angular**: es una *framework* basado en *JavaScript* desarrollado por Google.
- **NodeJS**: es un entorno de ejecución *JavaScript* del lado del servidor.

6.1.2.1. MongoDB

MongoDB es una base de datos multiplataforma de código abierto desarrollada en C++. Esta base de datos guarda la información en pares clave-valor utilizando el formato JSON y tiene un sistema NoSQL basado en documentos. Este sistema de base de datos es una opción ideal cuando se quiere manejar una gran cantidad de información ya que insertar datos en los documentos de MongoDB no requiere actualizar toda la tabla/colección. La Imagen 60 muestra el logotipo de MongoDB:



Imagen 60. Logotipo de MongoDB

6.1.2.2. Express

Express es un *framework* ligero y flexible para el lado servidor. Está diseñado para construir aplicaciones web dinámicas, de una sola página o de múltiples páginas. La Imagen 61 muestra el logotipo de Express:



Imagen 61. Logotipo de Express

6.1.2.3. Angular

Angular es un *framework* de código abierto basado en *JavaScript* y soportado por Google. El objetivo de esta herramienta es introducir la arquitectura MVC (*Model View Controller*) en las aplicaciones de navegadores para hacer más sencillo el proceso de desarrollo y testeo. Este *framework* nos permite crear aplicaciones dinámicas y personalizables.

Angular también permite el uso de HTML con lenguaje de plantillas. De este modo se puede extender la sintaxis del HTML para incluir componentes de nuestra aplicación en el marcado de las páginas web. Las características de Angular como los servicios inyectables o el enlazado de datos permiten ahorrar una cantidad considerable de código. En este proyecto se ha utilizado Angular 9. La Imagen 62 muestra el logotipo de Angular:



Imagen 62. Logotipo de Angular

6.1.2.4. NodeJS

NodeJS es un entorno de ejecución multiplataforma del lado servidor que permite a los desarrolladores lanzar servidores web y ejecutar aplicaciones web desarrolladas en el lenguaje *JavaScript*. Esta tecnología utiliza un modelo de entrada/salida no bloqueante y dirigido por eventos. Eso hace que sea una herramienta ligera, eficiente y permita servir a las aplicaciones en tiempo real. La Imagen 63 muestra el logotipo de NodeJS:



Imagen 63. Logotipo de NodeJS

6.1.3. Funcionamiento de *MEAN stack*

Las distintas tecnologías de las que está compuesto el *stack MEAN* funcionan e interactúan de la siguiente forma [4]:

- **Angular:** acepta peticiones de los usuarios y les muestra el resultado de las mismas.
- **NodeJS:** maneja las peticiones tanto por parte de cliente como del servidor.
- **Express:** se comunica con NodeJS y al recibir una petición, se conecta con la base de datos y devuelve una respuesta.
- **MongoDB:** este sistema de base de datos almacena y extrae información.

La Imagen 64 muestra el funcionamiento de los componentes del *stack MEAN*:

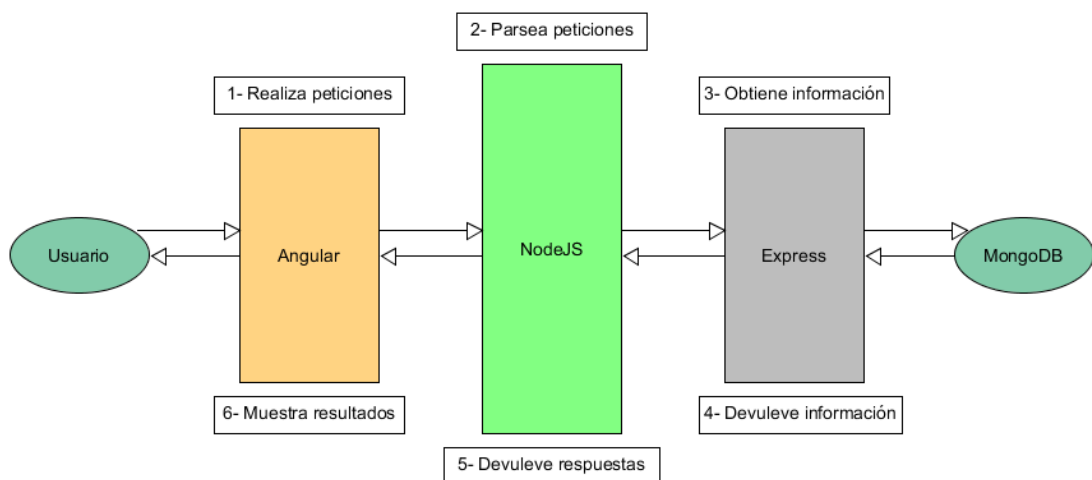


Imagen 64. Arquitectura MEAN

1. El cliente realiza una petición que es procesada por Angular.
2. Luego la petición se trasladada a NoseJS quien se encarga de *parsear* (procesar las entradas para que sean entendibles por los componentes integrados) la petición del usuario.
3. Express recibe la petición *parseada* de NodeJS y se comunica con la base de datos para obtener o insertar datos.
4. MongoDB realiza las acciones que solicita Express y devuelve la respuesta.
5. NodeJS recibe la respuesta de Express y se la envía al usuario.
6. En el lado del usuario, Angular recibe la respuesta de NodeJS y la muestra.

6.1.4. Módulos NodeJS

Al llegar a este punto, ya se conoce lo que es el *stack MEAN*, qué ventajas ofrece y por qué tecnologías está compuesto, pero para desarrollar una aplicación se requiere de más herramientas para construir todas las funcionalidades. En esta sección se detallan los diferentes módulos de NodeJS que se han utilizado en este trabajo de fin de grado. Los módulos son librerías de código abierto que ofrecen distintas funcionalidades y que se integran en el proyecto para su posterior utilización.

Para cargar un módulo en NodeJS se utiliza “**require(‘{nombre_del_módulo}’)**” y de esta forma se puede acceder a las funcionalidades ofrecidas por dicho módulo.

Los módulos utilizados en este proyecto son:

- **bcryptjs**: este módulo permite encriptar la información. Se ha utilizado para guardar las contraseñas de los usuarios y para hacer las comprobaciones correspondientes.
- **body-parser**: este módulo permite extraer la información del cuerpo de una petición HTTP entrante para su posterior tratamiento. En las versiones recientes de Express este módulo ya viene integrado pero en las versiones antiguas había que instalarlo por separado.
- **connect-multiparty**: este módulo permite subir los ficheros desde la aplicación web a una dirección específica.
- **cors**: con el uso de este módulo se permite realizar peticiones desde otros dominios. Se puede especificar el dominio concreto del que se quieren recibir peticiones o se puede permitir que cualquier dominio pueda enviar sus

peticiones. Este módulo es necesario para que el servidor pueda trabajar sin restricciones con el *frontend*.

- **dotenv**: este módulo permite añadir variables estáticas en el entorno de ejecución (NodeJS en esta aplicación) y permite el acceso a ellas desde cualquier punto del servidor.
- **express**: es el *framework* servidor que recibe las peticiones del usuario y se conecta con la base de datos.
- **jsonwebtoken**: este módulo permite generar un *token* de inicio de sesión para almacenarlo en la caché junto con el tiempo de expiración.
- **mongoose**: este módulo permite trabajar con la base de datos MongoDB desde el servidor Express.
- **morgan**: este módulo también es muy útil cuando se está desarrollando una aplicación debido a que permite mostrar por la consola del entorno de desarrollo las peticiones que realiza el *frontend* al *backend* y de esta forma facilitar el proceso de depuración.
- **nodemailer**: este módulo también desarrollado en *JavaScript* permite conectar desde la aplicación web a un servidor externo de correo electrónico para poder enviar notificaciones.
- **stripe**: este módulo es del servicio de pagos online. Permite integrar los pagos en la aplicación web.
- **nodemon**: otro módulo de gran ayuda a la hora de desarrollar un proyecto ya que reinicia tanto el *backend* como el *frontend* en caso de realizar y guardar cambios en los ficheros. Con esta herramienta se ahorra la tediosa tarea de parar el servidor y lanzarlo de nuevo para visualizar los cambios.

6.1.5. Módulos Angular

En este apartado se detallan los módulos que se han utilizado en el *frontend*. El proceso de instalación es igual que los módulos del servidor pero la forma de cargarlos para su posterior utilización es distinta en Angular. La carga de un módulo instalado en Angular se hace de la siguiente forma:

1. Importar en el fichero “**app.module.ts**” el módulo que se quiere utilizar tal y como se muestra en la Imagen 65:

```
import { BrowserModule, Title } from '@angular/platform-browser';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { NgModule } from '@angular/core';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
```

Imagen 65. Importación módulos Angular (1)

2. Después, en el mismo fichero, se añade el módulo importado en la lista “**imports**” como se puede observar en la Imagen 66:

```
imports: [
  BrowserModule,
  AppRoutingModule,
  FormsModule,
  HttpClientModule,
  BrowserAnimationsModule,
  ReactiveFormsModule
],
```

Imagen 66. Importación módulos Angular (2)

3. Ahora el módulo se puede importar en cualquier fichero de Angular donde se quiera hacer uso de sus funcionalidades. La importación se hace igual que en la Imagen 65.

Los módulos utilizados en el *frontend* del proyecto son:

- **bson**: este módulo permite generar identificadores para guardar un documento nuevo en la base de datos MongoDB.
- **jspdf**: este módulo permite generar y descargar ficheros PDF con HTML o texto básico.
- **materialize-css**: es la librería para el diseño de la interfaz gráfica de la aplicación web.
- **@types/materialize-css [10]**: este módulo también pertenece a la librería de la interfaz gráfica, pero se han importado los tipos especiales de colores de materialize-css para modificar los colores del tema por defecto de la librería. Se necesitan estos tipos debido a que para cambiar los colores del tema, se hace uso de colores ya definidos en materialize-css y sin la importación correspondiente, Angular no entiende los colores definidos y lanza un error. Los colores se cambian modificando el fichero “**_variables.scss**” y a modo de ejemplo, se muestran los cambios en el color primario, secundario, de éxito, de error y de enlaces en la Imagen 67:

```
$primary-color: color("materialize-red", "lighten-2") !default;
$primary-color-light: lighten($primary-color, 15%) !default;
$primary-color-dark: darken($primary-color, 15%) !default;

$secondary-color: color("teal", "lighten-1") !default;
$success-color: color("green", "base") !default;
$error-color: color("red", "base") !default;
$link-color: color("light-blue", "darken-1") !default;
```

Imagen 67. Cambio de tema en materialize-css

Como se puede observar en la imagen, se utilizan los colores en formato de texto y Angular no lo entiende hasta que no se haga la importación de tipos ya que por defecto, Angular solamente compila los colores en hexadecimal.

Todos los módulos utilizados en NodeJS están disponibles en el repositorio oficial de NPM⁹.

6.1.6. Entorno de desarrollo

Como entorno de desarrollo se ha utilizado el editor de código llamado “**Visual Studio Code**”¹⁰. Algunas de las ventajas que ofrece este editor es que es más ligero que otros editores o entornos de desarrollo para la cantidad de funcionalidades que proporciona, tiene soporte integrado de depuración, autocompletación de código, resaltado de sintaxis y control integrado de Git¹¹ entre otras características. La Imagen 68 muestra el logotipo del editor de código Visual Studio Code:



Imagen 68. Logotipo de Visual Studio Code

6.1.7. Herramientas y utilidades

⁹ <https://www.npmjs.com/>

¹⁰ <https://code.visualstudio.com/>

¹¹ <https://git-scm.com/>

Ya se conocen cuáles son las principales tecnologías con las que se ha desarrollado la aplicación web pero aparte hay más herramientas o utilidades que ayudan a terminar las funcionalidades del proyecto o comprobar su funcionamiento. A continuación, se detallan otras tecnologías usadas en el desarrollo:

- **Postman**¹²: esta herramienta permite realizar peticiones HTTP REST a un servidor sin la necesidad de crear un cliente. La Imagen 69 muestra el logotipo de Postman:



Imagen 69. Logotipo de Postman

- **UMLet**¹³: este programa permite elaborar diagramas UML de diferentes tipos. La Imagen 70 muestra el logotipo de UMLet:



Imagen 70. Logotipo de UMLet

- **Git**: es un software de control de versiones. Sirve para guardar versiones del código en un repositorio en la nube y así poder usarlo desde cualquier lugar del mundo. La Imagen 71 muestra el logotipo de Git:



Imagen 71. Logotipo de Git

- **Leaflet**¹⁴: es una biblioteca *JavaScript* de código abierto que permite integrar mapas en las aplicaciones. La Imagen 72 muestra el logotipo de Leaflet:



Imagen 72. Logotipo de Leaflet

¹² <https://learning.postman.com/docs/publishing-your-api/documenting-your-api/>

¹³ <https://www.umlet.com/changes.htm>

¹⁴ <https://leafletjs.com/>

- **OpenStreetMap (OSM)**¹⁵: es una librería de código abierto para construir mapas libres y editables. Ofrece una API muy extensa para obtener la información de las ubicaciones. La Imagen 73 muestra el logotipo de OpenStreetMap:



Imagen 73. Logotipo de Open Street Map

- **Stripe**: es una plataforma para integrar pagos online en las aplicaciones web. La Imagen 74 muestra el logotipo de Stripe:



Imagen 74. Logotipo de Stripe

- **Gmail**: es uno de los gestores de correo electrónico más conocidos. Se ha utilizado para enviar avisos del boletín de noticias de la aplicación web. La Imagen 75 muestra el logotipo de Gmail:



Imagen 75. Logotipo de Gmail

- **Robo 3T**¹⁶: es una herramienta moderna, robusta y multiplataforma para llevar la gestión de la base de datos MongoDB de forma gráfica. La Imagen 76 muestra el logotipo de Robo 3T:



Imagen 76. Logotipo de Robo 3T

- **MaterializeCSS**¹⁷: es un *framework* CSS que permite aplicar diseños a las páginas web. La Imagen 77 muestra el logotipo de MaterializeCSS:

¹⁵ <https://www.openstreetmap.org/>

¹⁶ <https://robomongo.org/>

¹⁷ <https://materializecss.com/>



Imagen 77. Logotipo de MaterializeCSS

6.2. Desarrollo de la solución

Ahora se va a presentar el resultado del desarrollo llevado a cabo para obtener la aplicación web solicitada para este TFG. Se muestra la estructura del proyecto junto con los componentes y recursos que se han creado para la implementación. En el entorno de desarrollo el proyecto está separado en dos partes principales: uno es el servidor (Express) [11] y el otro es el cliente (Angular) [12].

6.2.1. Servidor (Express)

En la parte del servidor existe un fichero principal llamado “**server.js**” que se encarga de obtener las rutas de la API, importar los módulos necesarios para trabajar con el intercambio de datos, especificar el contenido estático de la aplicación web y conectarse con la base de datos MongoDB.

Además, hay varias carpetas que separan las rutas de las entidades, los modelos de datos, los controladores para manejar las peticiones, la ubicación de los ficheros que se suban y la configuración de conexión con la base de datos. La Imagen 78 muestra la estructura de la carpeta “**server**”:



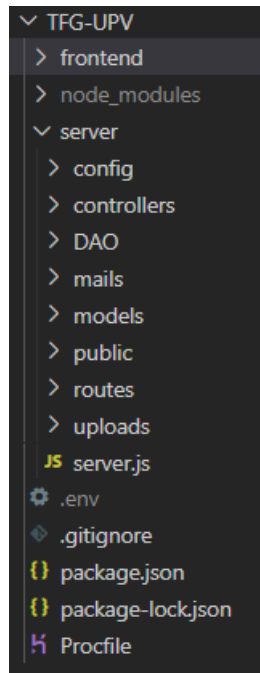


Imagen 78. Estructura global del servidor

6.2.1.1 Rutas (*Routes*)

La carpeta “**routes**” contiene todos los ficheros relacionados con las rutas de la API [13]. La Imagen 79 muestra el contenido de la carpeta:

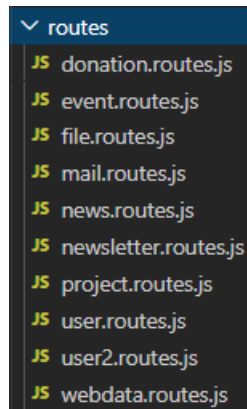


Imagen 79. Contenido de la carpeta routes

6.2.1.2 Modelos (*Models*)

La carpeta “**models**” contiene todos los ficheros relacionados con los modelos de datos de las entidades a utilizar en MongoDB. La Imagen 80 muestra el contenido de la carpeta:

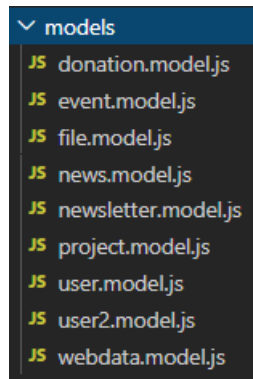


Imagen 80. Contenido de la carpeta models

6.2.1.3 Controladores (*Controllers*)

La carpeta “**controllers**” contiene todos los ficheros que tienen el código necesario para tratar las peticiones de los usuarios que se reciban por parte del lado del cliente. La Imagen 81 muestra el contenido de la carpeta:

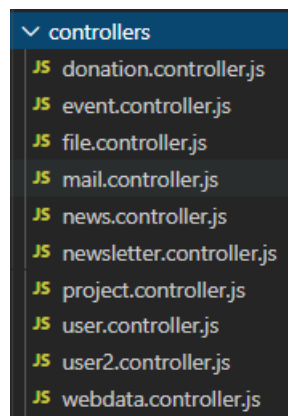


Imagen 81. Contenido de la carpeta controllers

6.2.2 Cliente (Angular)

En la parte del cliente se ha utilizado Angular y cuando se instala dicho *framework*, éste, genera un conjunto de carpetas y ficheros necesarios para su correcto funcionamiento. A partir de ese momento es cuando se empieza a trabajar con Angular y se pueden ir creando los elementos que se necesitan. La estructura global del lado del cliente es la que se muestra en la Imagen 82:

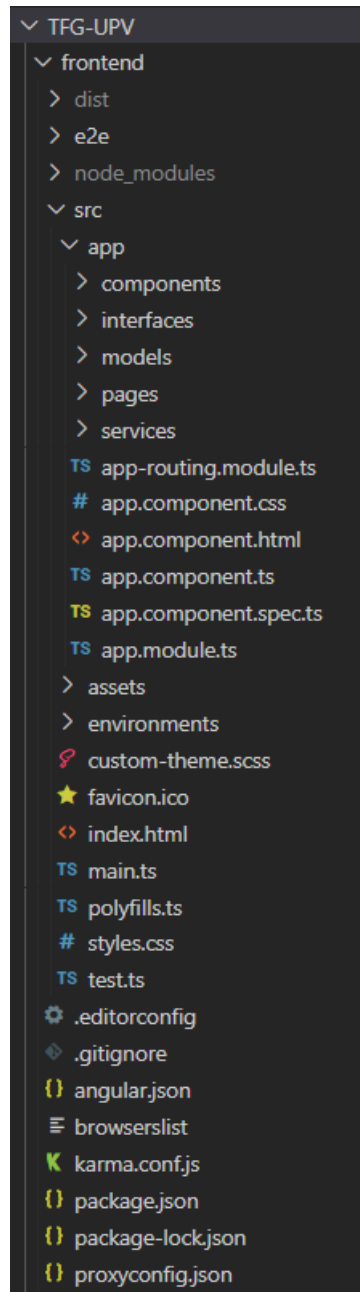


Imagen 82. Estructura global del frontend

Como se puede observar, la carpeta “**app**” es la que contiene los componentes que se trabajan principalmente en la aplicación. Lo que está fuera de esa carpeta es código generado por el propio Angular durante su instalación. Cabe destacar que aunque no se trabaje con mucha frecuencia con el contenido exterior de la carpeta “**app**”, sí que se utiliza para aplicar diseño CSS global, la carpeta “**assets**” se utiliza para guardar los recursos de la aplicación o los componentes “**app.module.ts**”, “**app-routing.module.ts**” y “**app.component.html**” son una parte esencial para la ejecución de la aplicación web:

- **app.modules.ts**: en este fichero se importan todos los componentes, servicios y librerías que se pretenden utilizar en Angular.
- **app-routing.module.ts**: este fichero contiene todas las rutas de la aplicación junto con el control de acceso a las mismas.
- **app.component.html**: este es el HTML principal de la aplicación; el contenido de este fichero es el que se mostrará al usuario cuando acceda a la página web.

6.2.2.1 Componentes (*Components*)

En esta carpeta se han creado los componentes que son compartidos para así reutilizarlos cuando haga falta como pueden ser la barra del menú, el pie de página, la plantilla para mostrar la información de los proyectos/eventos/noticias o el mapa para elegir ubicaciones. El contenido de la carpeta “**componentes**” se muestra en la Imagen 83:

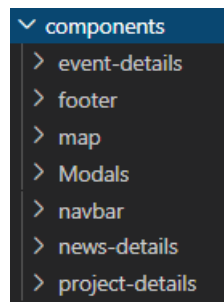


Imagen 83. Contenido de la carpeta components

6.2.2.2 Modelos (*Models*)

Para trabajar con las entidades que se guardan en la base de datos, es necesario crear las clases correspondientes en la parte del cliente para enviar dichos objetos al servidor y que éstos se guarden con comprobaciones previas para que no haya inconsistencia de datos o de estructuras. El contenido de la carpeta “**models**” se muestra en la Imagen 84:

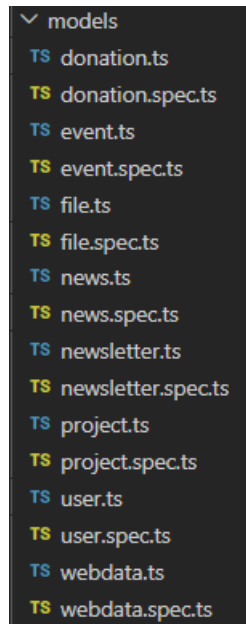


Imagen 84. Contenido de la carpeta models

Como se ve en la imagen, cada clase tiene también el fichero correspondiente a las pruebas. Estos ficheros se generan automáticamente cuando se crean clases con los comandos de Angular. El comando para generar una clase en Angular se muestra en la Imagen 85:

```
ng generate class nombreClase
```

Imagen 85. Comando de Angular para generar una clase

6.2.2.3 Páginas (*Pages*)

Para separar las páginas de la aplicación web del resto del código, se ha creado esta carpeta. Contiene todas las páginas que están disponibles para los usuarios. En la raíz de la carpeta están las páginas públicas o de la parte accesible por todos los usuarios y luego hay una subcarpeta llamada “**Backoffice**” que contiene todas las páginas que pertenecen a la gestión del *backoffice*. El contenido de la carpeta “**pages**” se muestra en la Imagen 86:

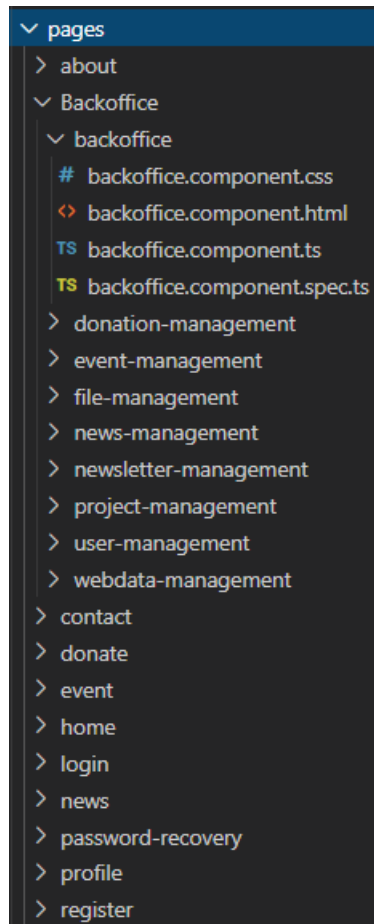


Imagen 86. Contenido de la carpeta pages

Tal y como se observa en la imagen, cada una de las páginas tiene el fichero correspondiente a los estilos CSS, el HTML, *JavaScript/TypeScript* y pruebas. Estos ficheros se generan automáticamente cuando se crean componentes con el comando de Angular. El comando para generar un componente en Angular se muestra en Imagen 87:

```
ng generate component nombreComponente
```

Imagen 87. Comando de Angular para generar un componente

6.2.2.4 Servicios (*Services*)

La carpeta de servicios contiene los ficheros que realizan peticiones HTTP a la API desde el lado del cliente y reciben las respuestas correspondientes. El contenido de la carpeta “**services**” se muestra en la Imagen 88:

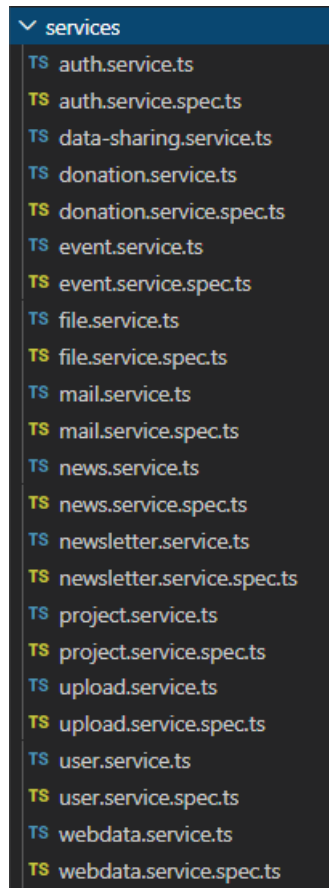


Imagen 88. Contenido de la carpeta services

En este caso también se encuentran los ficheros de pruebas junto a cada fichero de servicio y la generación de estos ficheros también es debida a la ejecución del comando de Angular. El comando para generar un servicio en Angular se muestra en la Imagen 89:

```
ng generate service nombreServicio
```

Imagen 89. Comando de Angular para generar un servicio

Como se ha podido observar en los apartados anteriores, se ha construido la aplicación con una estructura lo más modular posible para que el mantenimiento de la página web y los futuros cambios se puedan hacer de una forma más fácil. La modularidad permite que se puedan añadir o quitar componentes de la aplicación sin alterar el funcionamiento del resto de componentes y haciendo mínimas modificaciones en el código lo cual evita fallos en cascada o aparición de errores en otros componentes de la aplicación. Las tecnologías utilizadas son un factor muy importante para conseguirlo debido a que permiten separar las partes de la página web de una forma sencilla ofreciendo mecanismos de comunicación entre los componentes integrados.

6.2.3 Seguridad

La seguridad es un aspecto de suma importancia en cualquier proyecto para proteger los datos personales de los usuarios, la información de la propia aplicación o para evitar accesos no autorizados a partes del sistema. Se han implementado algunos mecanismos de seguridad en la aplicación desarrollada en el presente TFG que se detallan en este apartado.

- **Guardado de la contraseña del usuario:** se ha utilizado la encriptación *hash* para guardar los datos en la base de datos con el uso del módulo “**bcrypt**”. La función criptográfica *hash* es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. La ventaja que ofrece usar este algoritmo es que una vez aplicado el algoritmo, no se puede desencriptar y por tanto, se evita el riesgo de deducir la contraseña tanto por los administradores del sistema como por accesos ilícitos.
- **Comparación de la contraseña:** a la hora de validar los datos para iniciar sesión en la aplicación web, se aplica el algoritmo *hash* al valor que ha introducido el usuario como contraseña y se compara con el valor guardado en la base de datos con el algoritmo ya aplicado. Esto es debido a que no existe una forma para desencriptar una contraseña con el algoritmo *hash* y la única forma de saber si la contraseña que ha introducido el usuario es correcta es comparando ambas cadenas y si son iguales, significa que la contraseña introducida por el usuario coincide con el dato almacenado.
- **Mensaje de fallo de autenticación:** en caso de fallar la autenticación, se muestra un mensaje de aviso al usuario pero el mensaje es general y no indica si el fallo ha sido al comprobar el correo electrónico o la contraseña para así evitar cualquier tipo de deducción.
- **Resetear la contraseña:** al no existir una forma de desencriptar una contraseña al que se le haya aplicado el algoritmo *hash* (para aumentar la seguridad), no se puede recuperar el valor original de la contraseña y enviársela al usuario directamente en caso de solicitar la recuperación. Para hacer más seguro el proceso de resetear la contraseña se hace lo siguiente: se le solicita al usuario un correo de recuperación, se comprueba el correo introducido y si existe alguna cuenta en la base de datos con ese correo, se genera una URL especial compuesta por el identificador único de ese usuario y se envía dicho enlace al correo electrónico proporcionado para que el usuario pueda entrar a la



aplicación y actualizar su contraseña. En caso de intentar entrar a la página de recuperación sin el identificador único, la página se redirige automáticamente a la página principal.

- **Rutas con autenticación:** hay ciertas partes de la aplicación que solamente están accesibles por los trabajadores de la ONG y por tanto ningún otro usuario debería ser capaz de acceder a ellas. Para ello, la aplicación dota de un mecanismo de autenticación para cada URL que necesite controlar el acceso. Al acceder a dichas URLs, se comprueba el rol del usuario para mostrar el contenido de la página; en caso de detectar un rol no autorizado, se redirige al usuario a la página principal.
- **Donación:** la seguridad a la hora de realizar una donación es esencial ya que el usuario introduce sus datos bancarios y las claves correspondientes y se deben proteger esos datos. La plataforma que se ha usado en la aplicación para los pagos con tarjeta es Stripe, el cual permite externalizar todo el proceso de pago. Es decir, cuando el usuario selecciona el pago con tarjeta, se abre un formulario donde se solicitan los datos bancarios pero todas las validaciones y comprobaciones y los cobros se hacen en el servidor de Stripe (que sigue todos los protocolos de seguridad exigidos y cumple con las normativas necesarias) a través de su API.

7. Pruebas

Después de desarrollar la aplicación es muy importante que se hagan las pruebas y validaciones necesarias para asegurar el correcto desempeño de las funcionalidades desarrolladas antes de pasarlas a producción o desplegarlas en un entorno de uso real. En esta fase se comprueba que la aplicación hace lo que se pide y lo hace de forma correcta sin interrumpir el funcionamiento de la página web.

7.1 Pruebas funcionales

A continuación, se muestran las pruebas funcionales realizadas en esta aplicación y los resultados correspondientes.

- **Registrarse:** permite al usuario crearse una cuenta en la aplicación.
 - Se realizan las validaciones en el formulario para comprobar si se han introducido todos los datos obligatorios y en caso de faltar algún campo, se muestra un mensaje especificando exactamente el campo que se debe corregir tal y como se puede observar en la Imagen 90:

Crear una cuenta

Nombre y apellidos

Correo electrónico

shaheer@gmail.com

Contraseña

.....

Mínimo 6 caracteres

Fecha de nacimiento

¿Quieres recibir novedades?

No Sí

CREAR CUENTA

¿Ya tienes una cuenta?

Nombre incorrecto

Imagen 90. Validación de campos vacíos en "Registrarse"

- Si el usuario intenta darse de alta con un correo electrónico ya existente, se muestra el mensaje de la Imagen 91:

Crear una cuenta

Nombre y apellidos
Shaheer Shahzad

Correo electrónico
shaheer@gmail.com

Contraseña
.....

Mínimo 6 caracteres

Fecha de nacimiento
17/08/2020

¿Quiere recibir novedades?

No Sí

CREAR CUENTA

¿Ya tienes una cuenta?

El correo ya existe

Imagen 91. Correo existente en "Registrarse"

- Si se introducen todos los datos correctamente, se crea la cuenta, se inicia la sesión, se redirige al usuario a la página principal con el menú principal actualizado (Imagen 92) y el usuario recibe un correo de bienvenida (Imagen 93):

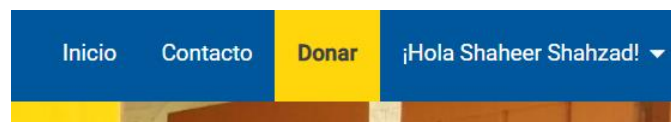


Imagen 92. Cambio menú principal al iniciar sesión

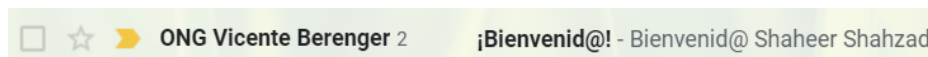


Imagen 93. Correo de bienvenida

- Si el usuario ha activado la opción de darse de alta en el boletín de noticias, recibirá un correo de aviso de la suscripción como se ve la Imagen 94:

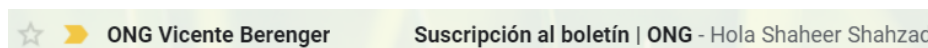
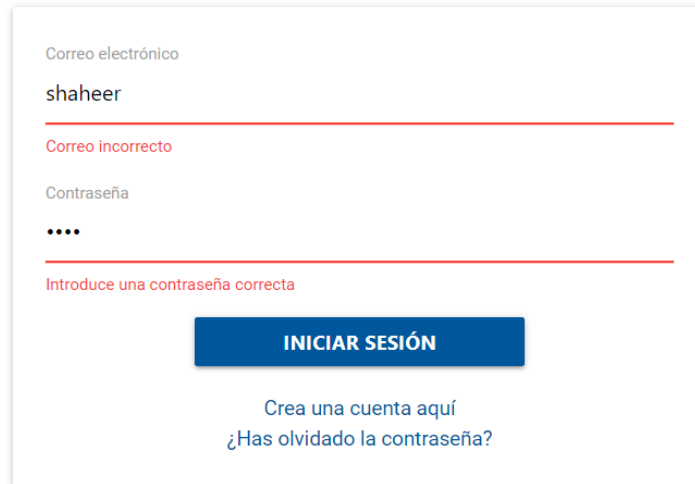


Imagen 94. Correo de suscripción al boletín de noticias

- **Iniciar sesión:** permite al usuario acceder a la aplicación web si este ya tiene creada una cuenta.

- Conforme el usuario va introduciendo los datos, se van realizando las comprobaciones de los campos para validar los datos y resaltar los campos en caso de detectar algún dato incorrecto como se puede observar en la Imagen 95:

Iniciar sesión



Correo electrónico
shaheer

Correo incorrecto

Contraseña
••••

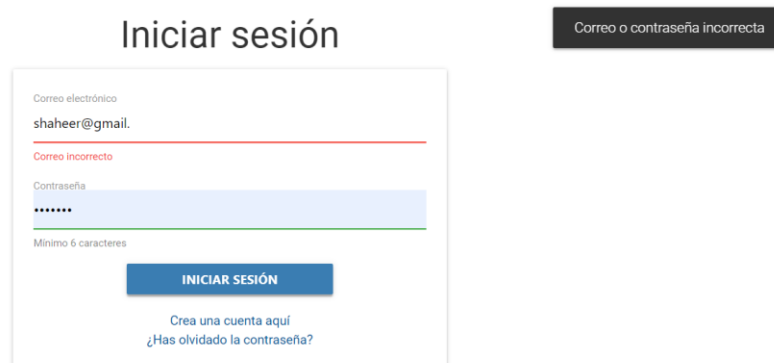
Introduce una contraseña correcta

INICIAR SESIÓN

[Crea una cuenta aquí](#)
[¿Has olvidado la contraseña?](#)

Imagen 95. Detección de datos incorrectos en "Iniciar sesión"

- Cuando el usuario intenta enviar el formulario con datos incorrectos, se le muestra un mensaje flotante avisando del error. En el mensaje de error para iniciar sesión, por seguridad, no se indica si falla el correo o la contraseña y se muestra el mensaje de la Imagen 96:



Iniciar sesión

Correo electrónico
shaheer@gmail.

Correo incorrecto

Contraseña
••••••

Minimo 6 caracteres

INICIAR SESIÓN

[Crea una cuenta aquí](#)
[¿Has olvidado la contraseña?](#)

Correo o contraseña incorrecta

Imagen 96. Mensaje al fallar la validación de credenciales

- Si el usuario introduce las credenciales correctamente, se inicia la sesión y se le redirige a la página principal de la aplicación con el menú superior actualizado como se muestra en la Imagen 92.
- **Cerrar sesión:** permite al usuario salir de la aplicación.

- Al acceder con las credenciales, se habilita la opción “**Salir**” en el menú principal como se observa en la Imagen 97:

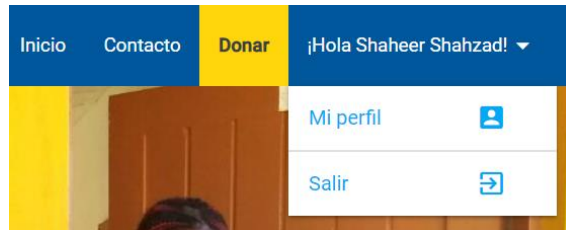


Imagen 97. Opción “Salir” al iniciar sesión

- Si el usuario pulsa la opción “**Salir**” del menú principal, se borran los datos de la caché del navegador y se actualiza la página para reiniciar las opciones del menú tal y como se muestra en la Imagen 98:

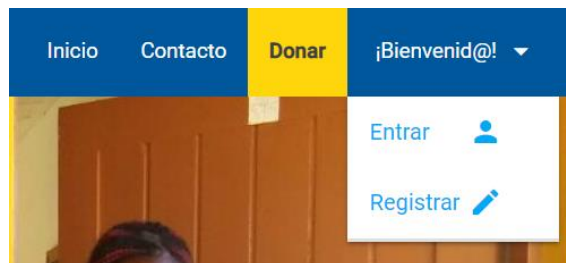


Imagen 98. Estado del menú principal al cerrar sesión

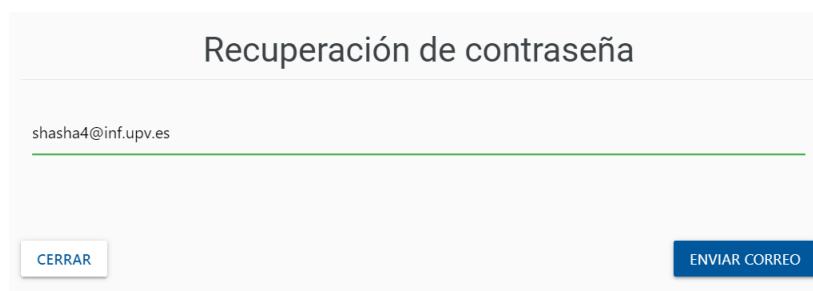
- **Ver perfil:** permite al usuario consultar su información personal.
 - Al acceder a la aplicación como usuario identificado, se habilita la opción “**Mi perfil**” en el menú principal como se muestra en la Imagen 97.
 - Al pulsar dicha opción, se lleva al usuario a la página donde se le muestra la información personal del mismo como se muestra en la Imagen 99:



Imagen 99. Página del perfil

- **Resetear contraseña:** esta funcionalidad le permite a un usuario actualizar su contraseña.
 - En la página para iniciar sesión hay una opción “**¿Has olvidado la contraseña?**” tal y como se muestra en la Imagen 95.

- Cuando el usuario pulsa dicha opción, se le pide un correo electrónico para enviar un correo de actualización como se muestra en la Imagen 100:



Recuperación de contraseña

shasha4@inf.upv.es

CERRAR ENVIAR CORREO

Imagen 100. Ventana para solicitar correo de recuperación

- Al pulsar el botón “**Enviar correo**” el usuario recibe un correo con el enlace para actualizar la contraseña como se observa en la Imagen 101:

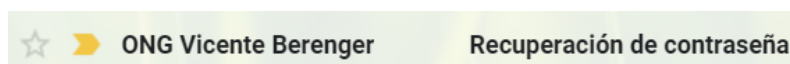
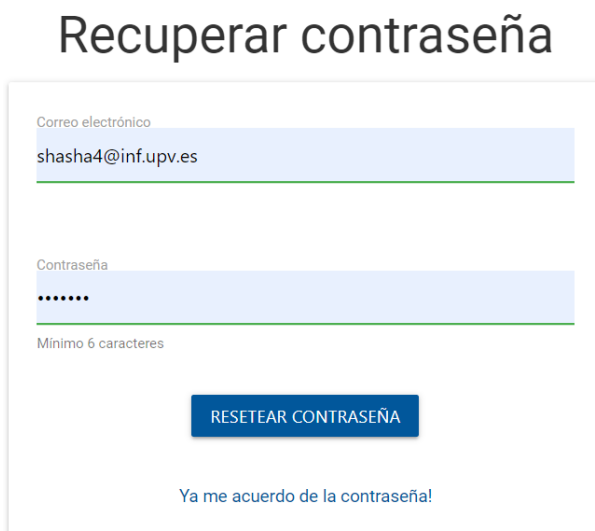


Imagen 101. Correo de recuperación de contraseña

- El correo enviado contiene un enlace para acceder al formulario para resetear la contraseña tal y como se muestra en la Imagen 102:



Recuperar contraseña

Correo electrónico
shasha4@inf.upv.es

Contraseña
.....
Mínimo 6 caracteres

RESETEAR CONTRASEÑA

Ya me acuerdo de la contraseña!

Imagen 102. Formulario para actualizar la contraseña

En este formulario se solicitan de nuevo el correo y la contraseña para confirmar que el usuario que ha accedido al formulario conoce el correo al que se envió el enlace de actualización. Cuando el usuario introduce los datos, se actualiza la contraseña en la base de datos y se redirige a la página de iniciar sesión de la aplicación web.

- **Suscripción al boletín:** permite añadir un correo electrónico en la lista de los suscriptores del *newsletter*.

Sitio Web para la gestión y publicitación de proyectos de cooperación

- Se comprueba el campo del correo electrónico para detectar si se ha dejado el campo vacío o se ha introducido un correo erróneo y mostrar el mensaje correspondiente tal y como se ve en la Imagen 103:

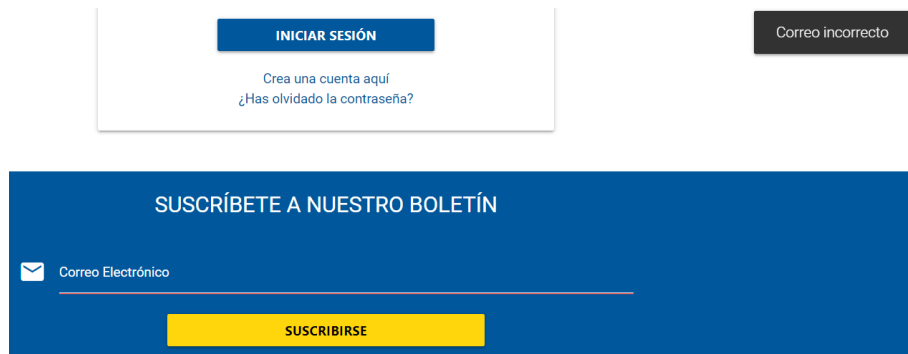


Imagen 103. Detección del formulario del boletín

- En caso de introducir un correo electrónico correcto, se añade dicho correo en la lista de suscriptores y se muestra un mensaje al usuario como se puede observar en la Imagen 104:

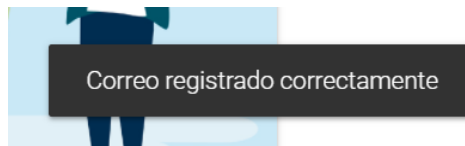


Imagen 104. Mensaje al dar de alta en el boletín

- **Ver proyectos:** permite a los usuarios ver los proyectos que se han creado en la ONG.
 - Los proyectos se listan en la página principal de la aplicación como se muestra en la Imagen 105:

Nuestros Proyectos

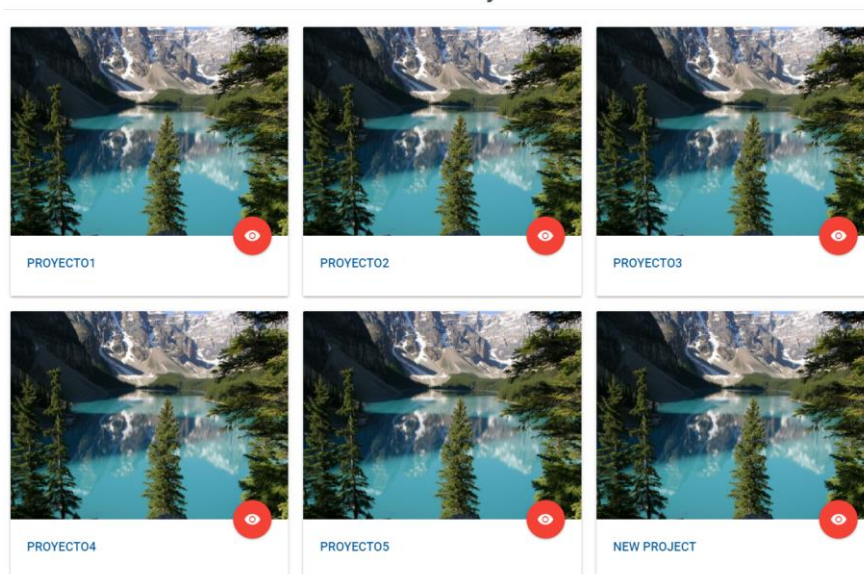


Imagen 105. Lista de proyectos en la página principal

- Al pulsar sobre un proyecto se puede consultar la información del mismo tal y como se observa en la Imagen 106:

Detalles del proyecto



Imagen 106. Página de detalles del proyecto

- **Ver eventos:** permite al usuario ver los eventos programados de la ONG.
 - Al acceder al apartado “**Eventos**” que está disponible en la página principal de la aplicación, se muestra el listado de los eventos que ha programado la ONG tal y como se muestra en la Imagen 107:

Eventos

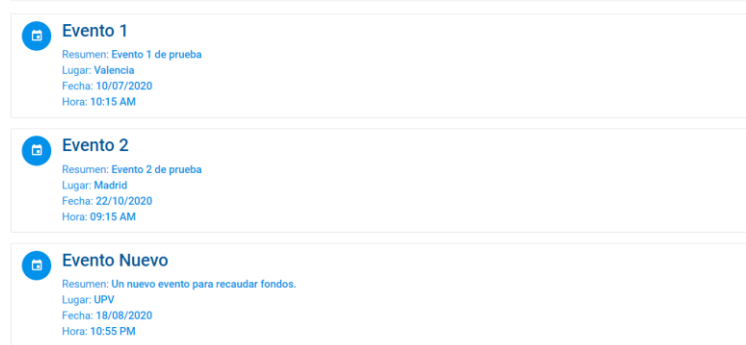


Imagen 107. Lista de los eventos

- **Ver noticias:** permite al usuario ver las noticias de la ONG.
 - Al acceder al apartado “**Noticias**” que está disponible en la página principal de la aplicación, se muestra el listado de las noticias tal y como se muestra en la Imagen 108:

Noticias

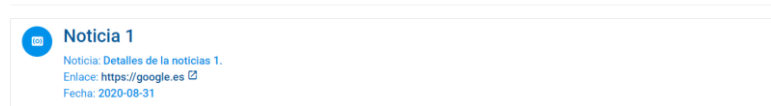


Imagen 108. Lista de las noticias

- **Contactar:** permite enviar un mensaje a la ONG accediendo al apartado “**Contacto**” de la página web.
 - Se comprueban los datos del formulario y en caso de faltar algún campo, se muestra un mensaje indicando el error tal y como se observa en la Imagen 109:

The screenshot shows the 'Contacto' form with a validation error message 'Nombre incorrecto' in a dark box at the top right. The form fields are: 'Nombre y apellidos' (empty), 'Correo electrónico' (filled with 'shasha4@inf.upv.es'), 'Asunto' (filled with 'Contacto'), and 'Mensaje' (filled with 'Este es un mensaje de contacto'). There is a checkbox for 'Acepto la política de privacidad' which is checked. At the bottom, there is a blue button labeled 'ENVIAR' with a right-pointing arrow.

Imagen 109. Validación del formulario de contacto

- Si se introducen los datos correctamente pero no se acepta la política de privacidad, no se puede enviar el formulario. En la Imagen 110 se puede ver

que el botón de “**Enviar**” está desactivado porque no se han aceptado las políticas:

Contacto

Nombre y apellidos
Shaheer Shahzad

Correo electrónico
shasha4@inf.upv.es

Asunto
Contacto

Mensaje
Este es un mensaje de contacto

Acepto la política de privacidad

ENVIAR ➤

Imagen 110. Requisito de aceptar las políticas de privacidad

- Si se rellena correctamente el formulario y se aceptan las políticas, el usuario puede enviar un mensaje y se le avisa al usuario de que el mensaje ha sido enviado correctamente.
- El mensaje se recibe en el buzón del correo de la ONG como se muestra en la Imagen 111:

Formulario contacto: Contacto - Este es un mensaje de contacto

Imagen 111. Mensaje del formulario de contacto

- **Ver donaciones realizadas:** permite a un usuario ver sus donaciones en el apartado “**Mi perfil**”.
 - En caso de que el usuario haya hecho alguna donación, se muestra el apartado “**Mis donaciones**” en la página del perfil donde se listan las donaciones realizadas tal y como se muestra en la Imagen 112:

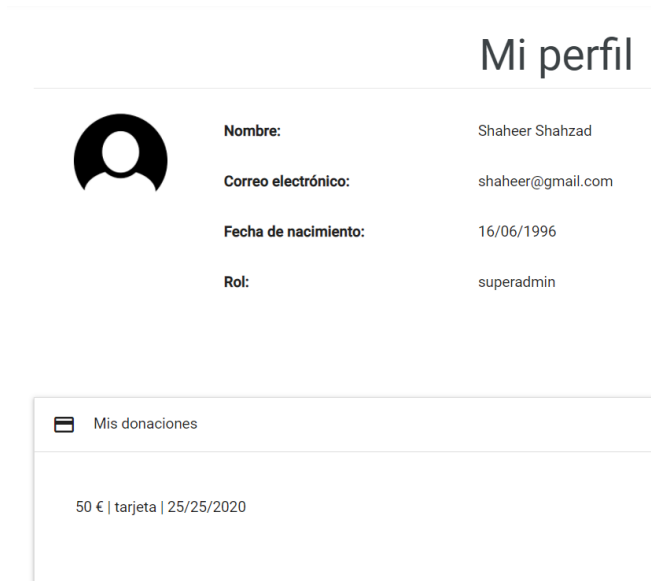


Imagen 112. Lista de donaciones del usuario

- En caso de no tener ninguna donación, el apartado “**Mis donaciones**” no se muestra.
- **Acceso por roles:** mostrar las opciones de acceso dependiendo del rol del usuario.
 - Al acceder a la aplicación con un rol de “**anónimo**”, “**registrado**” o “**donante**”, las opciones del menú principal son las que se muestran en la Imagen 97.
 - En caso de acceder como “**voluntario**” o “**administrador**”, se muestra la opción “**Backoffice**” que lleva al usuario al panel de administración de la aplicación web tal y como se muestra en la Imagen 113:

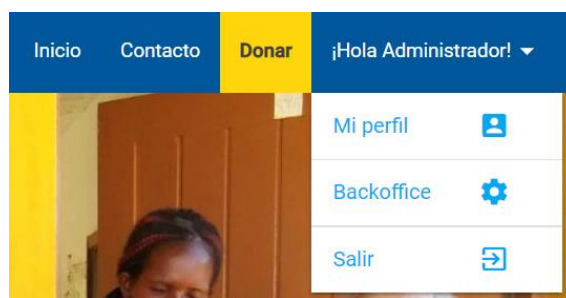


Imagen 113. Opciones de menú principal para trabajadores

- Al pulsar la opción “**Backoffice**” se accede al panel de control que se observa en la Imagen 114:

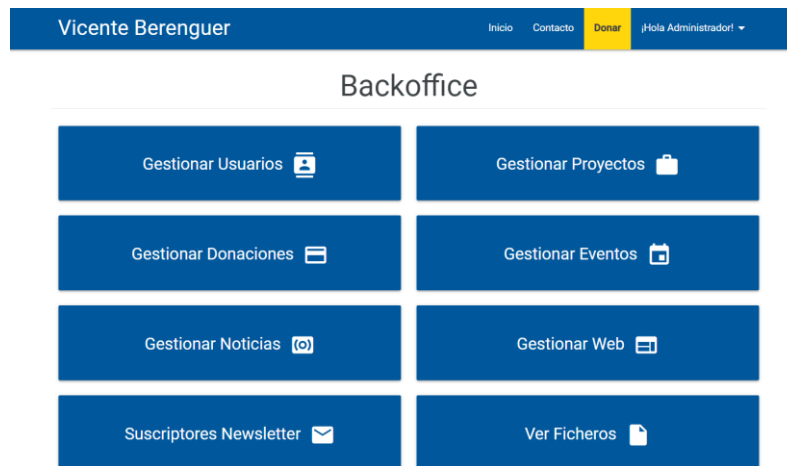


Imagen 114. Página del *backoffice*

- Al intentar acceder a la URL del *backoffice* con un rol distinto al “**voluntario**” o “**administrador**”, se le redirige al usuario a la página principal de la aplicación.
- **Crear proyecto:** permite añadir un proyecto nuevo en la aplicación.
 - Al intentar crear un proyecto sin rellenar el formulario, se muestra un mensaje al usuario especificando el error tal y como se observa en la Imagen 115:

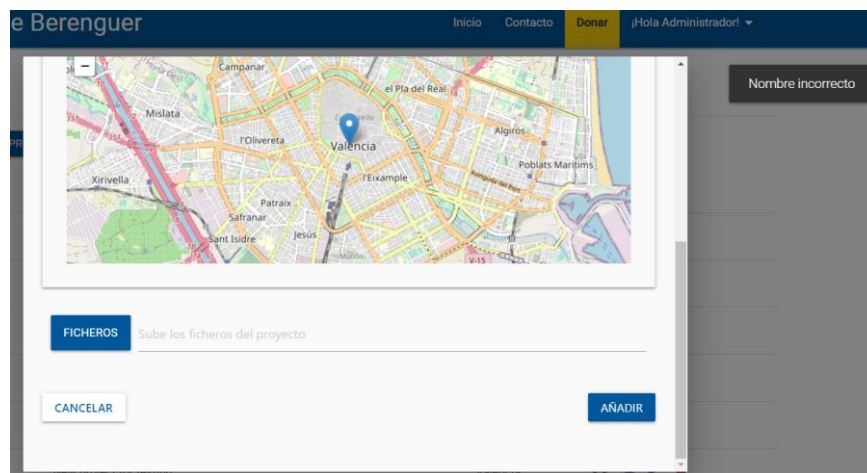


Imagen 115. Validación del formulario para crear proyecto

- Al rellenar el formulario correctamente, se añade el proyecto en la base de datos, se actualiza la lista y se muestra un mensaje para avisar al usuario tal y como se muestra en la Imagen 116:

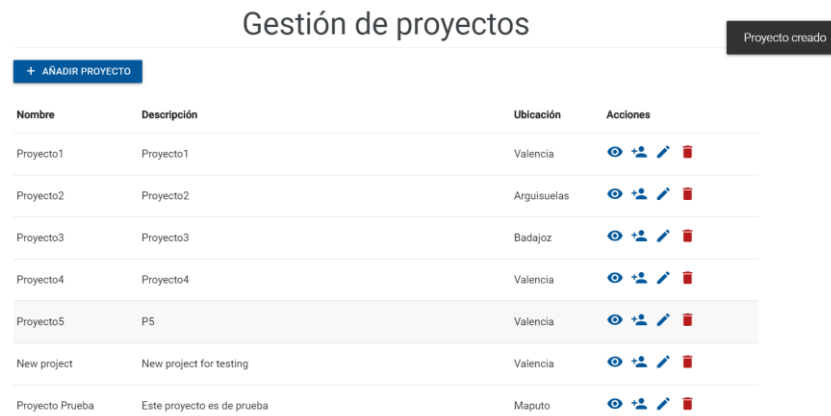


Imagen 116. Mensaje de éxito al crear proyecto

- **Ver proyecto:** permite ver los detalles del proyecto.
 - Al pulsar el icono del ojo, se muestra una tabla con los datos del proyecto que se quiere consultar y el listado de documentos si los tiene, tal y como se puede observar en la Imagen 117:

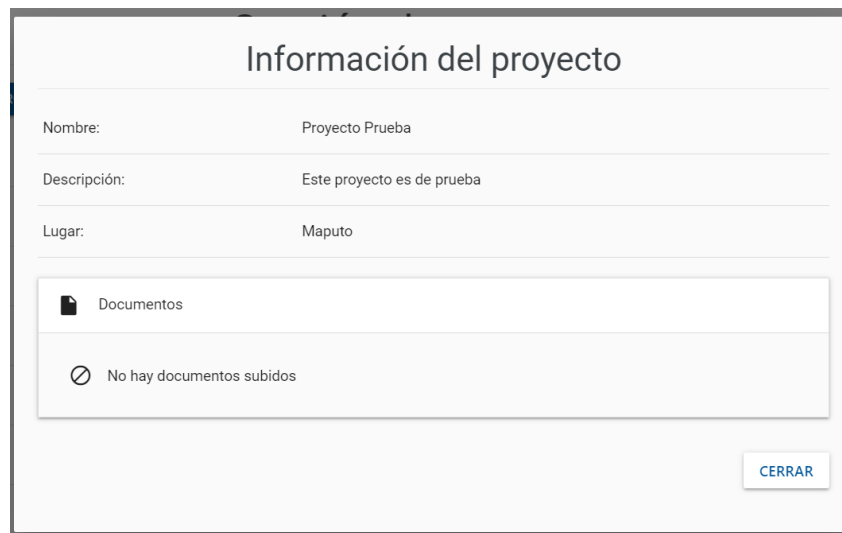


Imagen 117. Ventana con los detalles del proyecto

- Si el proyecto tiene algún fichero subido, este se lista en el apartado “**Documentos**” con la opción de descarga como se ve en la Imagen 118:

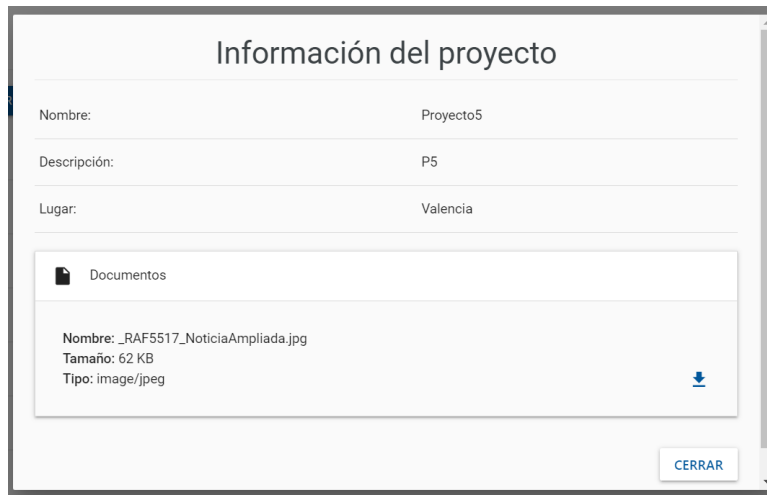


Imagen 118. Listado de ficheros del proyecto

- **Editar proyecto:** permite actualizar los datos del proyecto o añadir/quitar los ficheros del mismo si los tiene.
 - Al pulsar el icono del lápiz, se muestra un formulario con los datos del proyecto a editar ya introducidos para su actualización. La Imagen 119 muestra el formulario de edición:

Imagen 119. Formulario para editar proyecto

- Al enviar el formulario de edición, se comprueban los campos para detectar datos incorrectos o campos vacíos. En la Imagen 120 se puede observar el mensaje que se le muestra al usuario en caso de dejar un campo vacío:

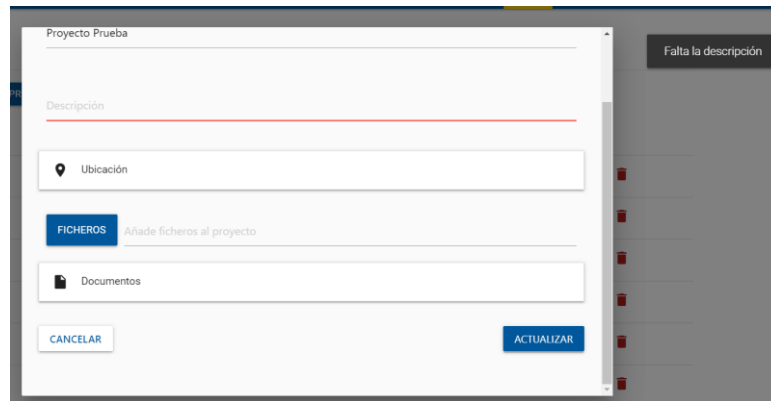


Imagen 120. Validación del formulario al editar proyecto

- En caso de que el proyecto a editar tenga algún fichero subido, se muestra la opción de descarga o borrado para el mismo tal y como se muestra en la Imagen 121:

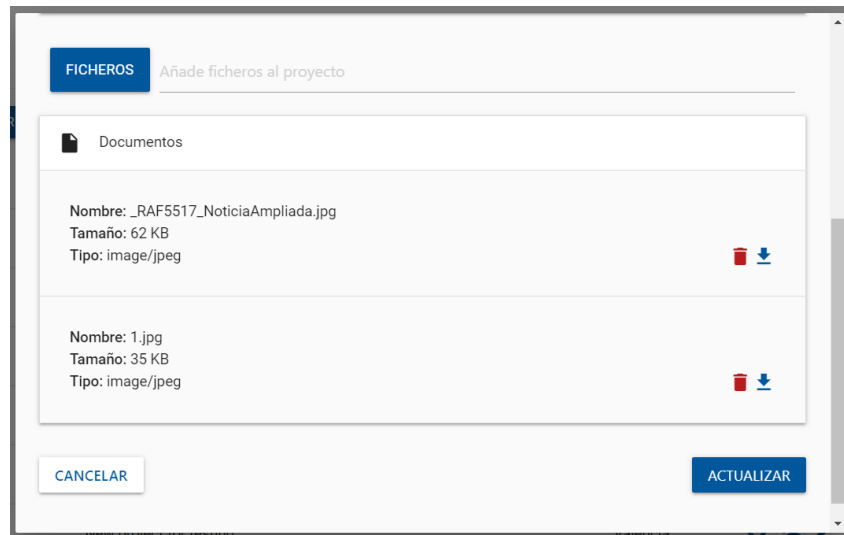


Imagen 121. Opciones de los ficheros al editar proyecto

- Al añadir un fichero en un proyecto (Imagen 122), se añaden los datos en la base de datos, se sube el fichero al servidor (Imagen 123) y se añade al listado de ficheros del proyecto (Imagen 124/Imagen 126):

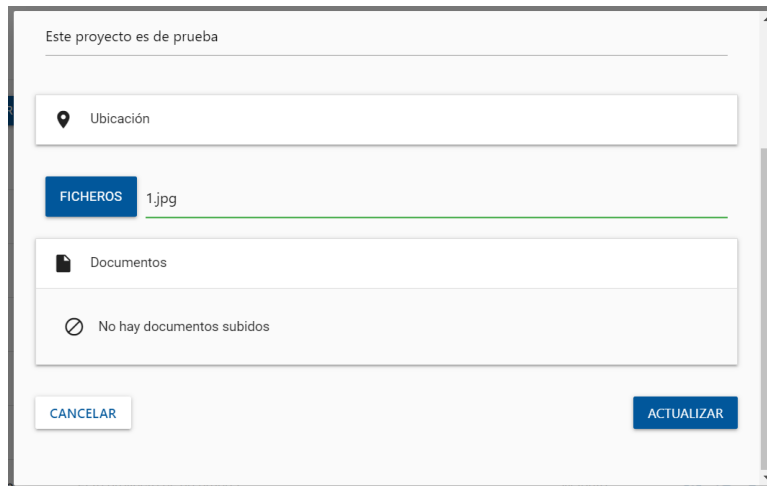


Imagen 122. Subida de fichero al editar proyecto

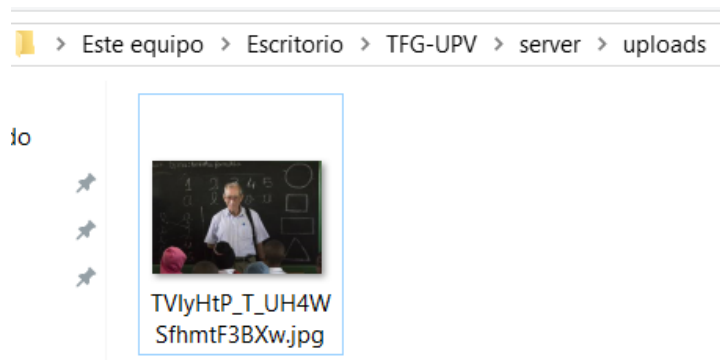


Imagen 123. Fichero subido a la carpeta del servidor

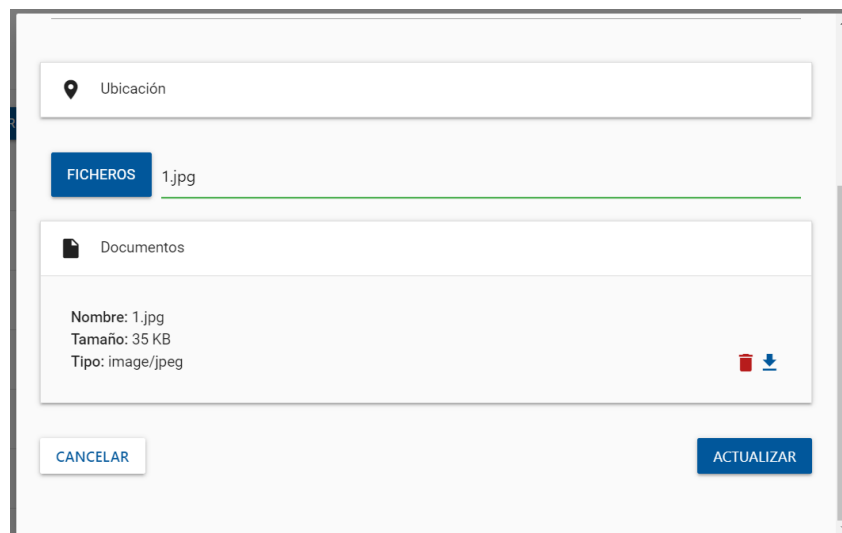


Imagen 124. Lista actualizada al subir fichero

- **Borrar proyecto:** permite borrar los datos y ficheros de un proyecto.
 - Al pulsar sobre el icono de la papelera, se abre una ventana para solicitar la confirmación tal y como se muestra en la Imagen 127:

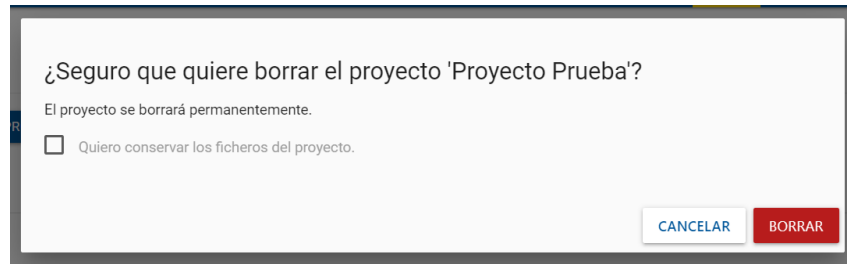


Imagen 125. Ventana de confirmación para borrar proyecto

- Se puede borrar un proyecto conservando sus ficheros si así lo desea el usuario. Se borra el proyecto con las opciones de la Imagen 125 y se comprueba que el fichero del proyecto ha sido borrado del servidor (Imagen 126) y el proyecto ya no aparece en la lista de los proyectos (Imagen 127):



Imagen 126. Fichero borrado en la carpeta del servidor

Gestión de proyectos

[+ AÑADIR PROYECTO](#)

Nombre	Descripción	Ubicación	Acciones
Proyecto1	Proyecto1	Valencia	   
Proyecto2	Proyecto2	Arguisueles	   
Proyecto3	Proyecto3	Badajoz	   
Proyecto4	Proyecto4	Valencia	   
New project	New project for testing	Valencia	   

Imagen 127. Proyecto borrado de la lista de proyectos

- **Añadir/quitar voluntarios:** permite añadir o quitar voluntarios a un proyecto.
 - Al pulsar el icono de la persona, se abre una ventana con el listado de los trabajadores de la ONG tal y como se muestra en la Imagen 128:



Imagen 128. Ventana para añadir/quitar voluntarios

- Al añadir un voluntario al proyecto, se cambia el icono y se muestra un mensaje al usuario tal y como se observa en la Imagen 129:

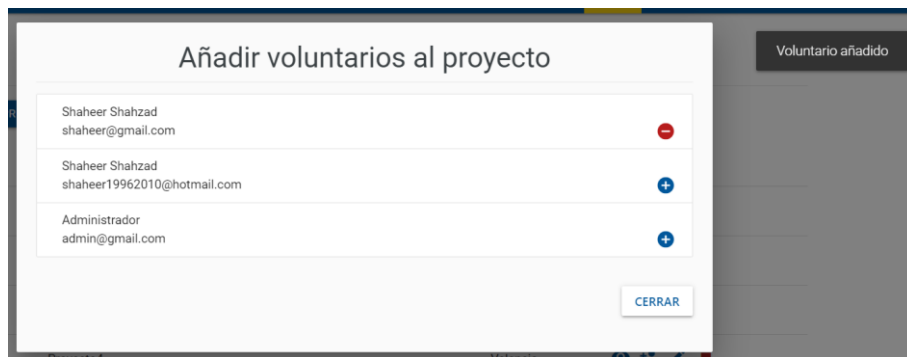


Imagen 129. Mensaje y cambio de icono al añadir voluntario

- Al quitar un voluntario, se cambia el icono y se muestra un mensaje al usuario tal y como se observa en la Imagen 130:

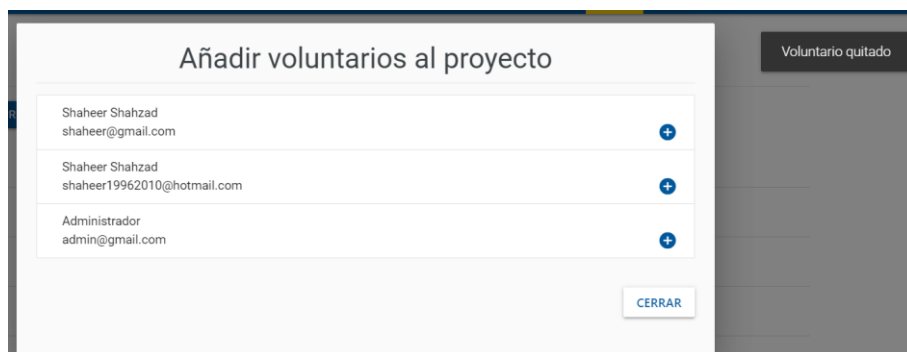


Imagen 130. Mensaje y cambio de icono al quitar voluntario

- **Información SEO:** permite cambiar el título, el meta-título y la meta-descripción de la página web.
 - En la Imagen 131 se puede observar la página web sin pulsar el botón de “Actualizar datos”:

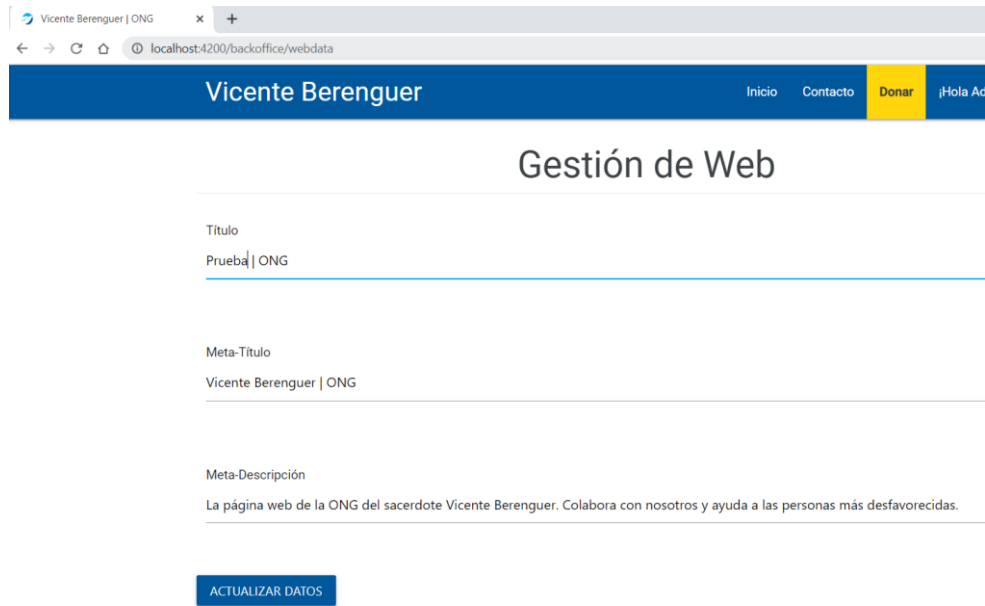


Imagen 131. Formulario de datos web antes de pulsar el botón de actualizar

- Al pulsar el botón de “**Actualizar datos**”, se muestra un mensaje al usuario (Imagen 132):



Imagen 132. Mensaje al actualizar los datos web

- Después de mostrar el mensaje, se refresca la página web y la información del formulario se traslada al mismo. En la Imagen 133 se puede ver la página web después de ser actualizada:

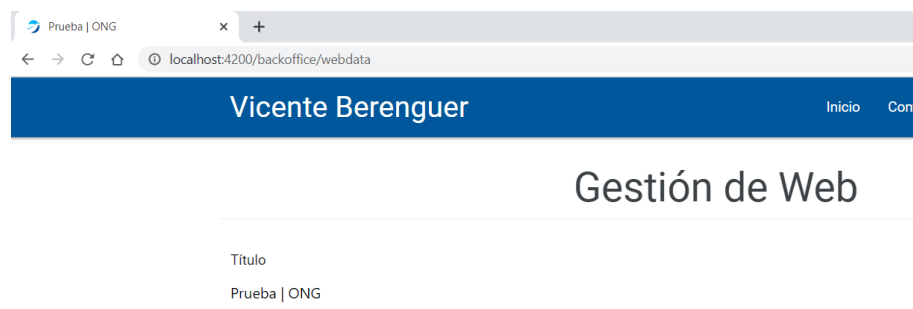


Imagen 133. Título actualizado al pulsar el botón de actualización

- **Quitar suscriptores del boletín:** permite al usuario acceder a la lista de los suscriptores del boletín de noticias y quitarlos.
 - Al pulsar el botón de la papelera, se abre una ventana para solicitar la confirmación del borrado tal y como se muestra en la Imagen 134:

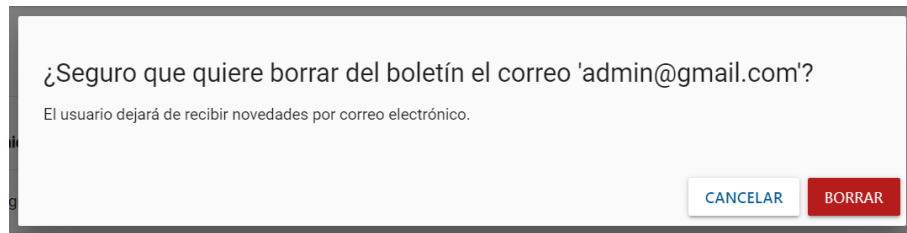


Imagen 134. Ventana de confirmación al borrar suscriptor del boletín

- Al pulsar el botón “**Borrar**”, se borra el dato en la base de datos, se muestra un mensaje al usuario y se actualiza la lista de los suscriptores en el *backoffice* como se puede ver en la Imagen 135:

Correo electrónico	Acciones
shaheer1996@gmail.com	
doner@gmail.com	
shaheer19962010@hotmail.com	
shaheer19962012@gmail.com	
shasha4@inf.upv.es	

Imagen 135. Mensaje al borrar el suscriptor del boletín

La gestión de usuarios, donaciones, eventos y noticias tienen las mismas funcionalidades que la de gestión de proyectos (crear, ver, editar y borrar). Debido a que son funcionalidades similares y lo único que cambia son los campos de las entidades, no se han mostrado mediante capturas, aunque sí se han realizado todas las pruebas de funcionamiento.

7.2 ***HTML W3C Validator***

El Consorcio *World Wide Web* (W3C)¹⁸ es una comunidad internacional que desarrolla y establece los estándares para el desarrollo web. *HTML W3C Validator* es una herramienta online disponible en la página web oficial de la organización para validar el código HTML de una página web para detectar posibles errores o avisos. La Imagen 136 muestra el resultado del análisis del dominio <https://vicente-berenguer-ong.herokuapp.com/>:

¹⁸ <https://www.w3c.es/Consortio/>

The screenshot shows a web browser window with the address bar containing the URL: `validator.w3.org/nu/?doc=https%3A%2F%2Fvicente-berenguer-ong.herokuapp.com%2F`. The page title is "Nu Html Checker". Below the title, there is a blue banner with the text "Nu Html Checker". A paragraph states: "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". Below this, it says "Showing results for `https://vicente-berenguer-ong.herokuapp.com/`". The "Checker Input" section includes a "Show" menu with options for "source", "outline", and "image report", along with an "Options..." button. The "Check by" dropdown is set to "address". The input field contains the URL `https://vicente-berenguer-ong.herokuapp.com/` and a "Check" button. A green banner displays the message: "Document checking completed. No errors or warnings to show." Below this, it notes: "Used the HTML parser. Externally specified character encoding was UTF-8. Total execution time 365 milliseconds." At the bottom, there are links for "About this checker", "Report an issue", and the version number "20.9.1".

Imagen 136. Análisis *HTML w3C Validator*

8. Despliegue y publicación

Esta es la última fase de todo el proceso de desarrollo de software lo cual significa que ya se ha pasado por la especificación de requisitos, el análisis de la solución, el diseño, la implementación y las pruebas. Ahora está todo listo para lanzar la aplicación para que sea accesible por los usuarios.

El despliegue se puede hacer en un *hosting* o en un servidor de computación. Existen varias alternativas a la hora de desplegar una aplicación entre las cuales se encuentran:

- **OpenShift**¹⁹: es la plataforma de desarrollo, con características de *Cloud Computing* de la capa PaaS (Plataforma como Servicio), que ofrece la empresa Red Hat.
- **Digital Ocean**²⁰: es un servicio de *hosting* y alojamiento en la nube, que incluye VPS (Servidor Privado Virtual) especialmente para desarrolladores.
- **AWS (Amazon Web Services)**²¹: ofrece servicio de computación en la nube.
- **GitHub pages**²²: es un servicio de alojamiento de sitio estático que toma archivos HTML, CSS y JavaScript directamente desde un repositorio en GitHub²³, opcionalmente ejecuta los archivos a través de un proceso de compilación y publica un sitio web.
- **SOME**²⁴: es un servicio de *hosting* con servidores Windows y Linux. Ofrece servidores virtuales.
- **Loading**²⁵: servicio de alojamiento web o *hosting*, VPS o servidores dedicados.
- **Heroku**: es una plataforma de servicio de computación en la nube que soporta distintos lenguajes de programación.
- **Microsoft Azure**²⁶: es un servicio de computación en la nube creado por Microsoft para construir, probar, desplegar y administrar aplicaciones y servicios mediante el uso de sus centros de datos.

El servicio que se ha utilizado para este proyecto es Heroku.

¹⁹ <https://www.openshift.com/>

²⁰ <https://www.digitalocean.com/>

²¹ <https://aws.amazon.com/es/what-is-aws/>

²² <https://pages.github.com/>

²³ <https://github.com/>

²⁴ <https://somee.com/>

²⁵ <https://www.loading.es/>

²⁶ <https://azure.microsoft.com/es-es/>

8.1. Heroku

Heroku es una empresa especializada en ofrecer servicios de plataforma administrada, por sus siglas en inglés PasS, en otras palabras ofrece servicios de servidores y redes administrados donde se pueden alojar aplicaciones de diferentes lenguajes de programación como Node, Python, Java, PHP, Ruby, Scala, Go, etc [14]. La Imagen 137 muestra el logotipo de Heroku:



Imagen 137. Logotipo de Heroku

Los motivos por los que se ha elegido Heroku como servicio de alojamiento para este proyecto son [15]:

- Servicio gratuito para empezar.
- Dominio seguro con el certificado SSL.
- La aplicación web es muy intuitiva y también tiene un cliente de comandos para desplegar proyectos.
- Es un servicio multiplataforma.
- Seguridad. Permite gestionar el acceso al panel de control.
- Ofrece una interfaz muy amigable.
- Es fácil de configurar.
- Proporciona buena escalabilidad.
- Buena documentación.
- Trabaja con Git pero el repositorio no es accesible.
- Integración con GitHub.
- Soporta varios entornos de ejecución y lenguajes.
- Es uno de los servicios gratuitos de despliegue más utilizados en la actualidad.

8.1.1. Despliegue con Heroku

El proceso para desplegar un proyecto en Heroku es algo fácil (una de las razones por la cual se ha decantado por este servicio) y se puede hacer desde el panel de control de Heroku, desde su línea de comandos o conectándose con GitHub [16] [17] [18].

Los pre-requisitos para desplegar la aplicación con Heroku es que se debe iniciar un repositorio local de Git en la carpeta del proyecto y hacer un *commit* al mismo y también se debe registrar en la plataforma (registro gratuito). Una vez que se han realizado los pasos anteriormente comentados, se procede al despliegue de la aplicación en Heroku:

8.1.1.1. Línea de comandos

- Lo primero que se debe hacer para desplegar el proyecto con la línea de comandos de Heroku es descargar e instalar el programa correspondiente.
- Después de instalar el programa hay que iniciar sesión en Heroku mediante el comando que se muestra en la Imagen 138:

```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit
> Warning: If browser does not open, visit
> https://cli-auth.heroku.com/auth/browser/**
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

Imagen 138. Comando para iniciar sesión en Heroku

Cuando se pulse cualquier tecla, se abre una pestaña del navegador para introducir las credenciales y volver a la línea de comandos para seguir con el proceso.

- Si aún no se ha inicializado un repositorio Git en la carpeta del proyecto, se puede hacer con los comandos que se muestran en la Imagen 139:

```
$ cd myapp
$ git init
Initialized empty Git repository in .git/
$ git add .
$ git commit -m "My first commit"
Created initial commit 5df2d09: My first commit
 44 files changed, 8393 insertions(+), 0 deletions(-)
 create mode 100644 README
 create mode 100644 Procfile
 create mode 100644 app/controllers/source_file
...
```

Imagen 139. Comandos para inicializar repositorio Git

- “**myapp**” se sustituye por el nombre de la carpeta principal que contiene el proyecto.
- “**My first commit**” se puede sustituir por otro mensaje pero siempre debe ir entre comillas.
- Ahora está todo listo para subir el código al repositorio de Heroku y desplegar la aplicación. Para ello, se debe crear una aplicación en la plataforma Heroku como se muestra en la Imagen 140:

```
$ heroku create
Creating app... done, ● thawing-inlet-61413
https://thawing-inlet-61413.herokuapp.com/ | https://git.heroku.com/thawing-inlet-61413.git
```

Imagen 140. Comando para crea una aplicación en Heroku

Cuando se ejecuta el comando “**heroku create**”, se crea una aplicación nueva en la plataforma con un nombre arbitrario que se muestra como resultado de la ejecución del comando y se asigna el repositorio recientemente creado como el repositorio remoto del proyecto local.

- Si ya se dispone de una aplicación en la plataforma, se le puede asignar un repositorio remoto con el comando que se muestra en la Imagen 141 donde se sustituye “**thawing-inlet-61413**” por el nombre de la aplicación a desplegar:

```
$ heroku git:remote -a thawing-inlet-61413
set git remote heroku to https://git.heroku.com/thawing-inlet-61413.git
```

Imagen 141. Comando para asignar repositorio remoto

- Ya está todo preparado y se puede desplegar la aplicación ejecutando el comando que se muestra en la Imagen 142:

```
$ git push heroku master
Initializing repository, done.
updating 'refs/heads/master'
...
```

Imagen 142. Comando para lanzar el despliegue

Con este comando se sube todo el repositorio local a la rama *master* del repositorio remoto y al finalizar la ejecución, la línea de comandos de Heroku proporciona la URL para acceder y visualizar la página web o también se puede abrir con el comando “**heroku open**”.

- En caso de querer desplegar una rama específica del proyecto, se haría de la forma que se muestra en la Imagen 143 sustituyendo “**testbranch**” por el nombre de la rama:

```
$ git push heroku testbranch:master
```

Imagen 143. Comando para desplegar una rama específica

8.1.1.2. Panel de control

Otra posibilidad de despliegue es utilizar la plataforma *online* de Heroku que ofrece un panel de control muy intuitivo y permite reducir la utilización de línea de comandos.

- Lo primero que se tiene que hacer es iniciar sesión en la plataforma. Al entrar en el panel de control, se listan todos los despliegues realizados. Se puede trabajar con uno ya existente o crear una nueva aplicación en Heroku.
- Para crear una nueva aplicación, se pulsa el botón “**New**” y se elige la opción “**Create new app**” tal y como se muestra en la Imagen 144:

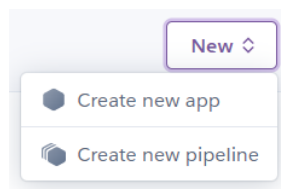


Imagen 144. Opciones para crear una aplicación nueva

- Ahora se solicita el nombre de la aplicación y la región de despliegue. Se rellenan los datos y se pulsa el botón “**Create app**” como en la Imagen 145:

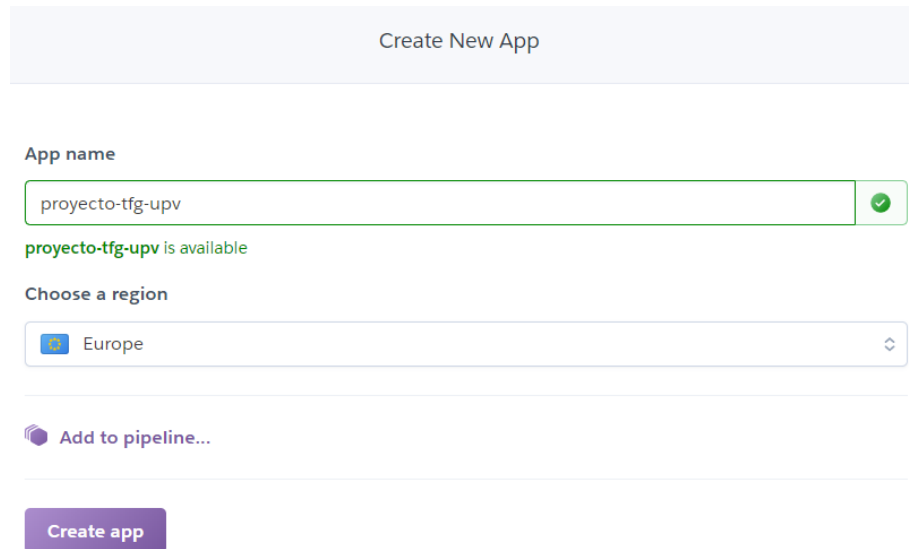


Imagen 145. Formulario para la nueva aplicación

- Una vez creada la aplicación, en Heroku se muestran los comandos necesarios a ejecutar para finalizar el despliegue de la aplicación. Simplemente se tienen que copiar los comandos y ejecutarlos en la línea de comandos de Heroku. Los comandos a ejecutar son los que se muestran en la Imagen 146:

Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a proyecto-tfg-upv
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```



You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

Existing Git repository

For existing repositories, simply add the `heroku` remote

```
$ heroku git:remote -a proyecto-tfg-upv
```

Imagen 146. Comandos para desplegar la aplicación

- Cuando se ejecuta el comando de despliegue, se proporciona la URL de la aplicación que se puede abrir ejecutando el comando **“heroku open”** o alternativamente se puede acceder a la página pulsando el botón **“Open app”** que está disponible en el panel de control como se muestra en la Imagen 147:

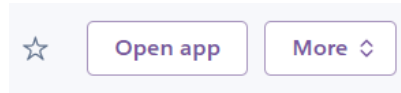


Imagen 147. Botón para acceder a la aplicación

8.1.1.3. GitHub

Otra opción de despliegue es conectar con Heroku un repositorio GitHub ya existente y se hace de la siguiente forma:

- Se accede al panel de control de Heroku.
- Se crea una aplicación nueva o se accede a una ya existente.
- Se accede a la pestaña **“Deploy”**.
- Se elige la opción de GitHub, se accede al repositorio de GitHub con las credenciales correspondientes, se busca el repositorio a desplegar y se pulsa el botón **“Connect”** tal y como se muestra en la Imagen 148:

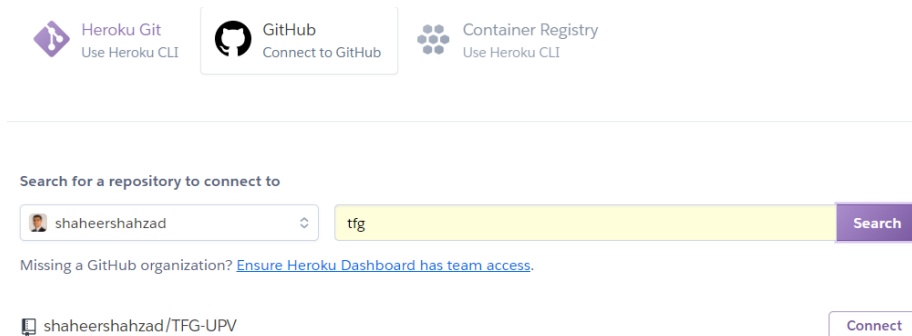


Imagen 148. Despliegue con GitHub

- Se lanza el proceso de integración y cuando finalice dicho proceso, el proyecto del repositorio estará disponible en un dominio.

8.1.1.4. Aplicar cambios en el dominio

Conforme se van haciendo cambios en el proyecto, hay que subirlos al repositorio de Heroku para que se despliegue de nuevo el código y los cambios se hagan visibles en la

página web. Esto se consigue con los siguientes tres comandos que se muestran en la Imagen 149:

Deploy your changes

Make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```

Imagen 149. Comandos para desplegar cambios en Heroku

1. El primer comando añade los ficheros con cambios a la lista de ficheros a confirmar.
2. El segundo comando confirma los cambios a subir con un mensaje.
3. El tercer comando sube los cambios al repositorio remoto y ejecuta el despliegue de nuevo.

8.2. MongoDB Atlas

Seguidos los procesos anteriores, la aplicación está disponible en un dominio pero aún no está operativa debido a que no está conectada con una base de datos. Para este proyecto se ha utilizado MongoDB Atlas como proveedor de base de datos en la red.

8.2.1. Creación de la base de datos en la nube

- Para crear una base de datos en MongoDB Atlas, lo primero que hay que hacer es registrarse de forma gratuita en la plataforma. Una vez registrados, se accede al panel de control y se pulsa el botón “**Create a New Cluster**” que se muestra en la Imagen 150:




Imagen 150. Crear una base de datos nueva

- Ahora se accede a la página de configuración de nuestra base de datos. Lo primero que se tiene que hacer es elegir el proveedor en la nube. En este trabajo

se ha elegido AWS (Amazon Web Services). Las opciones para seleccionar el proveedor son las que se muestran en la Imagen 151:

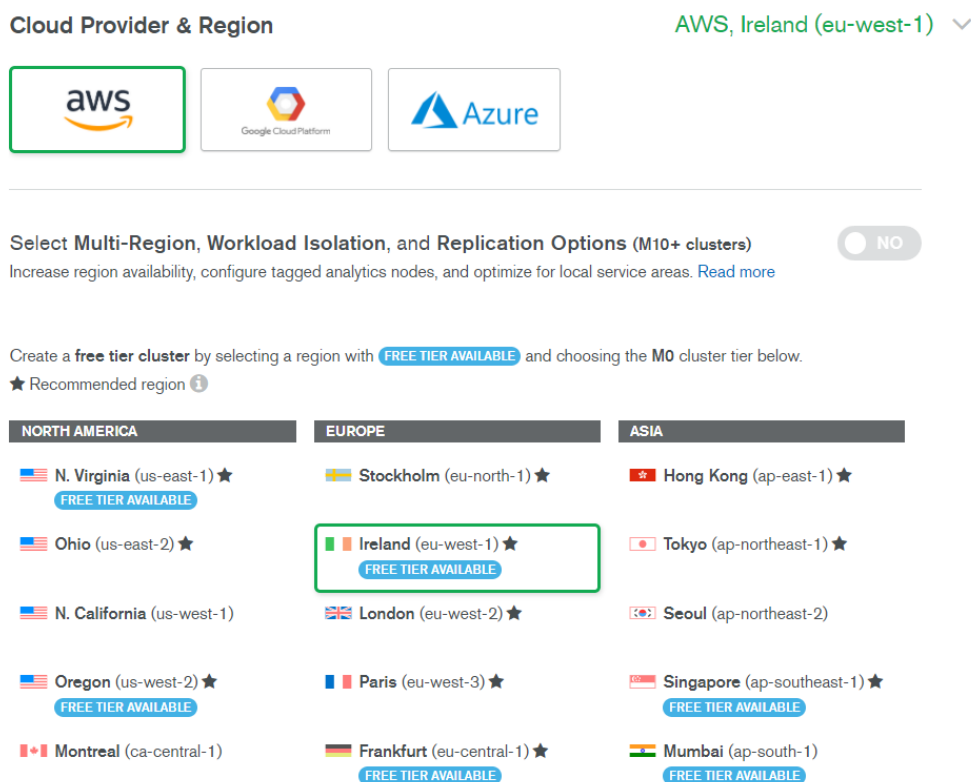


Imagen 151. Elegir el servicio de la nube para la base de datos

Para crear una base de datos gratuita se debe elegir una zona donde haya servidores gratuitos disponibles.

- Ahora se selecciona el paquete de prestaciones de la base de datos. Para el servicio gratuito se debe elegir el paquete “**Mo Sandbox**”.
- El apartado de configuración adicional se deja con las opciones por defecto.
- Se especifica un nombre para la base de datos tal y como se muestra en la Imagen 152:



Imagen 152. Nombre de la base de datos

- Para terminar, se pulsa el botón “**Create Cluster**”. El proceso de la creación de base de datos tomará unos minutos. Al finalizar, la base de datos estará disponible en la nube.

8.2.2. Conectar la base de datos con la aplicación

Llegados a este punto, la aplicación está ejecutándose en un dominio de Heroku y hay una base de datos disponible para guardar datos pero falta conectar ambas partes para que la página web esté operativa.

- Para conectar la base de datos en MongoDB Atlas con la aplicación que se ejecuta en Heroku, se tiene que acceder al panel de control de MongoDB Atlas y pulsar el botón “**CONNECT**” que está disponible entre las opciones de la base de datos tal y como se muestra en la Imagen 153:

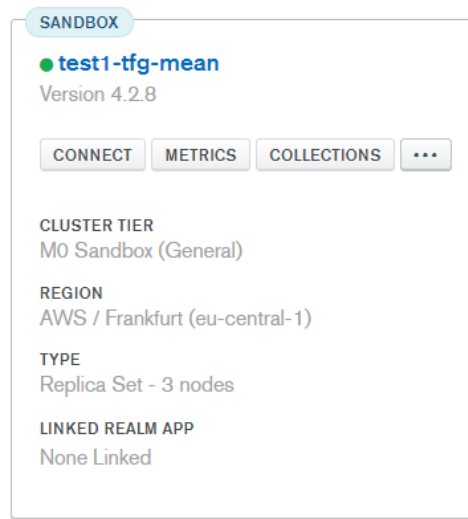


Imagen 153. Opciones de la base de datos

- Al pulsar ese botón, se muestran las distintas formas de conexiones. Se elige la segunda opción “**Connect your application to your cluster of MongoDB’s native drivers**” como en la Imagen 154:

✓ Setup connection security > Choose a connection method > Connect

Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.




-  **Connect with the mongo shell**
Interact with your cluster using MongoDB's interactive Javascript interface >
-  **Connect your application**
Connect your application to your cluster using MongoDB's native drivers >
-  **Connect using MongoDB Compass**
Explore, modify, and visualize your data with MongoDB's GUI >

Imagen 154. Opciones para conectar la base de datos

- En este paso se elige la tecnología que se está usando, su versión y se copia la URL que se proporciona tal y como se muestra en la Imagen 155:

✓ Setup connection security > ✓ Choose a connection method > Connect

1 Select your driver and version

DRIVER	VERSION
Node.js	3.6 or later

2 Add your connection string into your application code

Include full driver code example

```
mongodb+srv://dbVicenteBerenguer:<password>@test1-tfg-mean.bcxwb.mongodb.net/?retryWrites=true&appName=test1-tfg-mean
```

Copy

Replace **<password>** with the password for the **dbVicenteBerenguer** user. Replace **<dbname>** with the name of the database that connections will use by default. Ensure any option params are **URL encoded**.

Having trouble connecting? [View our troubleshooting documentation](#)

Imagen 155. URL para conectar la base de datos con la aplicación

- La URL que se copia de la ventana de MongoDB Atlas, se pega en la variable donde se especifica la URI de la base de datos local dentro del servidor del proyecto.

8.3. Prestaciones de los servicios

A la hora de trabajar con tecnologías y servicios gratuitos, un punto que puede generar dudas es si las prestaciones ofrecidas por dichas tecnologías y/o servicios son

suficientes. La respuesta depende de la escala del proyecto desarrollado ya que para proyectos pequeños o medianos las limitaciones pueden ser suficientes pero si la aplicación es para un público grande o se trabaja con recursos pesados, es posible que se tenga que optar por otras alternativas. Otros aspectos que se deben tener en cuenta son las especificaciones del servidor en el que se ejecuta la aplicación, la base de datos con la que se trabaja para tratar los datos y el tipo de servicio que se quiere ofrecer en la plataforma. Las tecnologías y servicios utilizados en este proyecto ofrecen las siguientes prestaciones:

NodeJS [19]:

- Necesita 256 MB de memoria RAM para ejecutarse.
- Soporta hasta 250 usuarios simultáneos con la configuración por defecto y permite a cada usuario solicitar cuatro recursos estáticos cada uno.

Se considera que las características ofrecidas por NodeJS son suficientes ya que el límite de accesos simultáneos suele ser inferior a 250 usuarios en aplicaciones pequeñas o medianas y permitir obtener hasta 1000 recursos estáticos es suficiente para empezar.

MongoDB Atlas [20]: se ha utilizado la versión gratuita “**Mo Sandbox**”.

- Ofrece una capacidad de almacenamiento de 512 MB.
- Permite hasta 500 conexiones concurrentes.
- Permite realizar 100 operaciones por segundo en la base de datos.
- El servidor es compartido lo cual ralentiza el rendimiento.
- El límite máximo de datos que se pueden tratar en siete días es de 10 GB.
- No permite realizar copias de seguridad.
- El historial de accesos a la base de datos no está disponible.
- Permite crear un máximo de 100 bases de datos y un total de 500 colecciones de datos.

Las prestaciones de MongoDB Atlas se consideran suficientes para empezar a registrar datos. Se debe tener en cuenta que la aplicación trabaja principalmente con datos en formato de texto lo cual permite almacenar una gran cantidad de información con 512 MB de capacidad de almacenamiento y el número de conexiones concurrentes es suficiente.

Heroku [21]: se ha elegido el plan gratuito.

- Ofrece una memoria RAM de 512 MB para cada aplicación ejecutada.
- Permite realizar un máximo de 4500 peticiones por hora a la API.
- Soporta 20 conexiones concurrentes.
- El tiempo que puede estar caído el servidor es de cuatro horas en un mes.
- Desactiva la máquina virtual donde se ejecuta la aplicación después de 30 minutos de inactividad y la reactiva automáticamente al detectar un acceso.
- La primera petición HTTP tarda 30 segundos en procesarse y mostrar la aplicación web.

Ofrece un buen rendimiento para empezar con un proyecto y considerando que al principio el tráfico no será mayor. La memoria RAM es suficiente para ejecutar la instancia de NodeJS y sobra la mitad para dedicarla al resto de procesos.

Cabe destacar que tanto MongoDB Atlas como Heroku son servicios que ofrecen ampliar sus planes de forma fácil sin la necesidad de realizar modificaciones en el código y como las tecnologías que se han utilizado para desarrollar la aplicación son escalables, se adaptan a las nuevas prestaciones con facilidad.



9. Presupuesto

Al desarrollar un proyecto, algo muy importante es saber cuál será el coste del mismo. En este apartado se va a exponer el estudio realizado para estimar el coste total de la aplicación web desarrollada en el presente TFG considerando el trabajo realizado por el alumno, el tiempo dedicado por las personas que han colaborado en el desarrollo y el coste del material utilizado que incluye el equipo *hardware* y los gastos realizados en el uso de las tecnologías y/o servicios contratados para la puesta en marcha de la aplicación.

Según los estudios, el salario de un ingeniero informático recién titulado ronda los 18.000 € y 22.000 € [22] y el de los profesores universitarios se encuentra entre los 30.000 € y 36.000 € brutos al año [23]. Por tanto, se podría considerar que el coste por hora aproximado de un ingeniero es de 12 € y el de un profesor universitario es de 24 €.

En el coste del material entran el consumo del equipo en el que se ha desarrollado el proyecto, la base de datos contratada en la nube y el servicio de computación para el despliegue de la aplicación web. En el caso de este proyecto se ha utilizado un equipo de gama media-alta y el consumo medio de un equipo con dichas características es de 0,03 € por hora [24]. La base de datos utilizada ha sido MongoDB Atlas²⁷ que ofrece una tarifa mínima de 7,60 € al mes [25] y el servicio de despliegue utilizado ha sido Heroku²⁸ que ofrece un plan mensual de 21,11 € al mes [26].

Considerando que el alumno ha dedicado 320 horas al proyecto, el tutor 30 horas y que la aplicación web ha estado activa durante un mes utilizando el servicio de base de datos y de computación, se estima que el coste del proyecto es de **4.598,31 €** tal y como se muestra en la Imagen 156:

²⁷ <https://www.mongodb.com/cloud/atlas>

²⁸ <https://www.heroku.com/>

Recurso	Horas	Coste	
Alumno	320	12,00 €	3.840,00 €
Profesor	30	24,00 €	720,00 €
Equipo	320	0,03 €	9,60 €
Base de datos	Coste mensual	7,60 €	7,60 €
Despliegue	Coste mensual	21,11 €	21,11 €
Coste total			4.598,31 €

Imagen 156. Presupuesto estimado del proyecto

10. Conclusiones

Todos los objetivos planteados para este TFG han sido cumplidos. Se ha desarrollado y desplegado una aplicación web para la ONG del padre Vicente Berenguer que permite publicar, anunciar, gestionar y mantener los proyectos, los eventos y las noticias, gestionar la información SEO de la página web, y también permite recibir donaciones económicas a través de diferentes tipos de tarjetas o proporcionando los datos bancarios a la ONG para domiciliar los pagos si así lo desea el donante. La parte pública de la página web se ha desplegado en un dominio público lo que permite el acceso a ella a todos los usuarios de la red y la parte privada está dotada de un sistema de registro y autenticación para el acceso. Todos los visitantes de la página pueden darse de alta en un boletín de noticias que se encarga de avisar de las novedades de la ONG mediante el correo electrónico.

10.1. Trabajo futuro

A medida que se ha ido desarrollando el proyecto, han surgido varias ideas que se pueden implementar para mejorar la aplicación web debido a que se han observado ciertas posibilidades de mejora o funcionalidades que ayudarían a los usuarios y también a la propia ONG. La envergadura de este proyecto puede crecer bastante ya que una ONG suele tener muchos otros aspectos a tratar que también se pueden conectar con la aplicación. A continuación, se mencionan algunas características y/o funcionalidades que se pueden trabajar en un futuro y potenciar la página web:

- Más métodos de pagos como Paypal, Apple Pay, Google Pay, Amazon Pay y Bizum.
- Avisos a través de SMS.
- Mostrar en la página web las colaboraciones con otras organizaciones.
- Subida de vídeos para los proyectos.
- Filtros para organizar los diferentes listados.
- Buscadores en las páginas de listados.
- Registros y accesos a la aplicación web mediante las redes sociales como Google, Facebook, Twitter o Instagram.
- Recordatorios de eventos.

- Creación de páginas dinámicas junto la administración correspondiente en el *backoffice*.
- Mejorar la gestión del SEO. Ofrecer la posibilidad de modificar más información SEO.
- Integrar publicaciones de redes sociales en la página web (Twitter, Facebook o Instagram).
- Atención por WhatsApp y un acceso desde la página web.
- Asociación de etiquetas a los ficheros para optimizar las búsquedas.
- Funcionalidad para editar el perfil personal.
- Posibilidad de subir una foto de perfil para los usuarios.
- Posibilidad de descargar justificantes de donaciones.

10.2. Valoración

Desde el punto de vista profesional, este trabajo me ayudará mucho debido a que se ha trabajado con numerosas tecnologías que actualmente se utilizan con mucha frecuencia en el mundo laboral y haber trabajado y familiarizado con dichas tecnologías me facilitará mi inserción laboral y mejorará mi desempeño profesional. Para comprender el funcionamiento de las herramientas utilizadas he necesitado consultar una gran cantidad de documentación lo cual me ha permitido entender las estructuras generales de las documentaciones para el acceso más rápido a las partes más importantes de ellas y obtener la información necesaria de una forma óptima. Se ha intentado seguir una planificación lo cual me ha permitido cumplir con las entregas con fechas límites y acostumbrarme al entorno real de trabajo. Seguir un proceso definido de desarrollo de *software* también ha sido un aspecto importante de este proyecto ya que las empresas también utilizan procesos parecidos y conocer alguno de ellos es una ventaja. Estar en contacto con el tutor para las correcciones o mejoras en el trabajo también me ha ayudado a comunicarme mejor, ver los problemas desde otra perspectiva y aplicar los cambios y/o sugerencias del tutor entendiendo el motivo de las mismas.

En el aspecto personal también he crecido mucho. He aprendido a organizarme mejor para cumplir con la planificación y las entregas de los borradores. La utilización de herramientas nuevas me ha permitido expandir mis conocimientos y me siento satisfecho con el aprendizaje. Al desarrollar un proyecto desde cero, he explorado muchas formas diferentes de superar los problemas afrontados y he encontrado maneras alternativas de resolver situaciones complejas de desarrollo. He aprendido a

gestionar el estrés o la presión de cumplir con los objetivos marcados de una forma correcta. Lograr finalizar el trabajo de forma satisfactoria me ha dado mucha confianza.



11. Anexos

11.1. Configuración de Stripe

En este apartado se van a mostrar las distintas configuraciones de la plataforma Stripe [27]. Se verá cómo se puede conectar la herramienta con una cuenta real de la ONG y realizar los cambios necesarios en el proyecto desarrollado, la activación de la cuenta nueva, desactivar los pagos de prueba y la forma en la que se pueden transferir los fondos desde Stripe a la cuenta bancaria que se desea.

11.1.1. Activación de la cuenta

Stripe es una herramienta que permite integrar pagos *online* en una aplicación web. La aplicación desarrollada en el presente TFG hace uso de esta herramienta pero los pagos que se realizan en modo de prueba, es decir, las donaciones se hacen utilizando las tarjetas de pruebas que proporciona el propio Stripe a los desarrolladores. La conexión con el servidor de Stripe para autenticar los datos, la confirmación del cobro y la respuesta del mismo se realizan igual que en caso de una tarjeta real. Por el lado del panel de control de Stripe, los cobros se ingresan correctamente en la cuenta pero también son datos de pruebas que se deben desactivar para empezar a trabajar con ingresos reales.

A continuación, se muestra cómo se debe configurar Stripe para recibir los ingresos a una cuenta bancaria real y desactivar el entorno de pruebas. Para empezar, se deben completar los datos solicitados por Stripe para activar la cuenta. Este paso es necesario ya que la plataforma necesita comprobar los datos de la entidad y cumplir con la normativa de pagos *online*. Para completa los datos de la activación, se debe acceder a la opción “**Activa tu cuenta**” del menú principal del panel de control de Stripe tal y como se muestra en la Imagen 157:

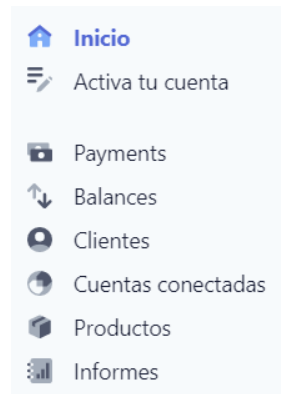


Imagen 157. Menú principal de Stripe

Los datos que necesita Stripe para permitir ingresos reales a la cuenta son:

1. La dirección de empresa u organización. La Imagen 158 muestra el formulario para dar de alta la dirección:

Un formulario web con un fondo gris claro. El título es 'Dirección comercial registrada'. Hay un campo de selección desplegable con 'España' y un icono de flechas. A continuación hay tres campos de texto: 'Línea de dirección 1', 'Línea de dirección 2' y 'Código postal'. Luego un campo de texto 'Ciudad' y un campo de selección desplegable 'Provincia' con un icono de flechas. Debajo de esto, el título es 'Tipo de empresa' y hay un campo de selección desplegable con 'Empresario individual (autónomo)' y un icono de flechas. En la parte inferior hay un botón azul con el texto 'Siguiete →'.

Imagen 158. Formulario para añadir la dirección de la entidad

2. Los datos de la empresa u organización. La Imagen 159 muestra el formulario para añadir los datos legales de la empresa u organización:

Datos de la empresa

Cuéntanos algunas cosas básicas de tu empresa.

Nombre legal de la empresa

Empresa

El nombre que proporciones debe coincidir exactamente con el nombre asociado a tu ID fiscal.

Número de ID fiscal (NIF)

A12345678

Número de IVA (NIF) Opcional

Con el nombre comercial Opcional

Dirección comercial registrada

Línea de dirección 1

Línea de dirección 2

Código postal

Ciudad

Valencia

Teléfono comercial

ES +34 810 12 34 56

Industria

Organizaciones benéficas y de servicios sociales

Sitio web de la empresa

empresa.com

Descripción del producto

Imagen 159. Formulario para los datos legales de la entidad

3. Datos del representante legal de la empresa u organización. La Imagen 160 muestra el formulario para añadir dicha información:



Nombre legal de la persona

Nombre de pila

Apellidos

Dirección de correo electrónico

ejecutivo@empresa.com

Nombre del cargo

CEO, gerente, socio

Fecha de nacimiento

DD / MM / AAAA

Domicilio particular

España

Línea de dirección 1

Línea de dirección 2

Código postal

Ciudad

Provincia

Número de teléfono

ES +34 810 12 34 56

Imagen 160. Formulario para los datos del representante legal

4. Información sobre los directores (opcional). La Imagen 161 muestra el formulario para añadir información de los directores:



Directores de la empresa

En virtud de la normativa, debemos recopilar información sobre los directores de la empresa. [Más información.](#)

Enumera a todas las personas que sean miembros del consejo de administración de la empresa.

+ Añadir un director

Continuar sin directores →

Imagen 161. Opción de añadir director de la entidad

5. Completar la información que se mostrará a los donantes en los cargos a sus respectivas cuentas. El formulario para completar esta información es el que se muestra en la Imagen 162:

Confirmar la información del extracto bancario del cliente

Descripción del cargo en el extracto bancario

Descripción abreviada

Número de teléfono de soporte al cliente

Siguiente →



La descripción del cargo en el extracto bancario es el nombre comercial que verá el cliente en los extractos de su tarjeta. Elige un nombre que sea fácil de reconocer para evitar posibles contracargos.

Imagen 162. Información para los extractos bancarios

6. Los datos bancarios para transferir los ingresos de Stripe al banco de la empresa u organización. La Imagen 163 muestra el formulario para completar este paso:

¿Adónde debemos enviar tus transferencias?

Divisa

País de la cuenta bancaria

IBAN

Confirmar IBAN

Yo, el titular de la cuenta, soy la única persona que debe autorizar los adeudos. Al presentar una cuenta bancaria, autorizo a Stripe a enviar fondos desde y hacia esta cuenta bancaria a través del sistema de adeudo de la Zona Única de Pagos en Euros (SEPA) y confirmo que he leído y acepto el [Contrato de servicios](#), incluido el Mandato de Adeudo Directo de SEPA.

Siguiente →

Imagen 163. Formulario para añadir información bancaria

7. Un aspecto muy importante es la seguridad en los pagos y en los cobros. Stripe necesita que se active la autenticación en dos pasos para activar la cuenta. En la Imagen 164 se puede observar lo que es la autenticación en dos pasos y el formulario para activarla:



Imagen 164. Activación de autenticación en dos pasos

8. Se muestra el resumen de los datos añadidos y se finaliza la activación de la cuenta.

11.1.2. Desactivar los datos de prueba

Una vez activada la cuenta, se procede a desactivar los datos de prueba para que lleguen ingresos reales a la plataforma de Stripe y posteriormente puedan ser transferidos a la cuenta bancaria añadida. Para desactivar los datos de prueba se debe deshabilitar la opción “**Viendo datos de prueba**” en el menú principal del panel de control de Stripe que se muestra en la Imagen 165:

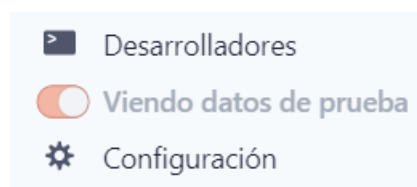


Imagen 165. Opción para activar/desactivar datos de prueba

Recordar que Stripe no permite desactivar los datos de prueba hasta que la cuenta no esté activada.

11.1.3. Conectar Stripe con la aplicación

Para conectar la aplicación web desarrollada en este TFG con Stripe se deben seguir los siguientes pasos:

1. Acceder a la opción “**Desarrolladores**” del menú principal del panel de control de Stripe y dentro de ella pulsar la opción “**Claves de API**” para ir a la página donde están disponibles todas las claves tanto públicas como secretas de la API tal y como se muestra en la Imagen 166:



Imagen 166. Páginas de las claves de API

En la imagen se observa la clave pública con el prefijo “**pk_test_**” debido a que la cuenta utilizada para este anexo utiliza los datos de prueba. Las claves en un entorno real tienen el prefijo “**pk_live_**” si es la clave pública y “**sk_live_**” si es una clave secreta.

2. Copiar la clave secreta de la API y sustituirla en el fichero de configuración de los pagos donde se importa el módulo de Stripe. En el proyecto, el fichero de configuración de los pagos con tarjeta se llama “**payment.controller.js**” y se debe buscar la línea que se muestra en la Imagen 167:

```
// Clave secreta de la API stripe va aquí.  
const stripe = require("stripe")(process.env.stripeKey);
```

Imagen 167. Uso de la clave secreta en el proyecto

Sitio Web para la gestión y publicitación de proyectos de cooperación

Se debe sustituir “**process.env.stripeKey**” por la clave secreta copiada desde el panel de control de Stripe y debe ir entre comillas.

Ya está la aplicación conectada con Stripe y funciona con pagos reales.

12. Referencias bibliográficas

- [1] *Proceso del desarrollo del software*. (2020). en *Wikipedia*, <[https://es.wikipedia.org/wiki/Proceso del desarrollo del software#:~:text=En%20ingenier%C3%ADa%20del%20software%2C%20un,vida%20del%20desarrollo%20de%20software.](https://es.wikipedia.org/wiki/Proceso_del_desarrollo_del_software#:~:text=En%20ingenier%C3%ADa%20del%20software%2C%20un,vida%20del%20desarrollo%20de%20software.)> [Consulta: 8 de agosto de 2020]
- [2] Angel Alvarez, M. (2020). “Qué es MVC” en *Desarrolloweb*, 28 de julio. <<https://desarrolloweb.com/articulos/que-es-mvc.html>> [Consulta: 13 de agosto de 2020]
- [3] María Aguilar, J. (2019). “¿Qué es el patrón MVC en programación y por qué es útil?” en *CampusMVP*, 15 de octubre. <<https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>> [Consulta: 15 de agosto de 2020]
- [4] Rungta, K. (2020). “What is Mean Stack Developer? Skills, Salary, Growth” en *Guru99*, 01 de enero. <<https://www.guru99.com/mean-stack-developer.html>> [Consulta: 22 de junio de 2020]
- [5] Haverbeke, M. (2014). *Eloquent JavaScript: A Modern Introduction to Programming*. San Francisco. No Starch Press. <https://polibuscador.upv.es/primo-explore/fulldisplay?docid=TN_cdi_skillsoft_books24x7_bks00077614&context=PC&vid=bibupv&lang=ca_ES&search_scope=ALL&adaptor=primo_central_multiple_fe&tab=default_tab&query=any,contains,javascript>
- [6] Kadivar, N. (2019). “LAMP Stack or MEAN Stack—Which One to Choose For Your Next Web Application?” en *Hackernoon*, 05 de julio. <<https://hackernoon.com/lamp-stack-or-mean-stack-which-one-to-choose-for-your-next-web-application-75dcbb2b69fi>> [Consulta: 13 de mayo de 2020]
- [7] Ganguly, S. (2019). “Most Popular Technology Stack To Choose From: Full Stack Vs. MEAN Stack Vs. MERN Stack In 2019” en *Hackernoon*, 27 de febrero. <<https://hackernoon.com/most-popular-technology-stack-to-choose-from-full-stack-vs-mean-stack-vs-mern-stack-in-2019-d12c0a17439a>> [Consulta: 13 de mayo de 2020]
- [8] Patel, R. (2019). “Why Choose Mean Stack For Your Web & Mobile App Development Projects?” en *Medium*, 09 de agosto. <<https://medium.com/@ronak8036/why-mean-stack-ec42aa82818>> [Consulta: 16 de mayo de 2020]



- [9] Manuel Alarcón, J. (2015). “Qué es el stack MEAN y cómo escoger el mejor para ti” en *CampusMVP*, 19 de enero. <<https://www.campusmvp.es/recursos/post/Que-es-el-stack-MEAN-y-como-escoger-el-mejor-para-ti.aspx>> [Consulta: 15 de mayo de 2020]
- [10] Java Blogs. (2016). “Customizing Materialize css – changing default theme with Sass” en *Ghariaonline*, 14 de marzo. <<https://ghariaonline.wordpress.com/2016/03/14/materialize-css-changing-default-theme-with-sass/>> [Consulta: 30 de junio de 2020]
- [11] YOUTUBE, “MEAN Curso - Mongodb, Express, Angular 6 y Nodejs, Parte 1 - Backend con Nodejs, Express y Mongodb” en *Youtube* <<https://www.youtube.com/watch?v=khCIunNAEHI>> [20 de junio de 2020]
- [12] YOUTUBE, “MEAN Curso - Mongodb, Express, Angular 6 y Nodejs, Parte 2 - Frontend con Angular 6” en *Youtube* <<https://www.youtube.com/watch?v=ccBtSAMFjto>> [Consulta: 20 de junio de 2020]
- [13] Watmore, J. (2019). “Angular 7 Tutorial Part 3 - Add Routing & Multiple Pages” en *Jasonwatmore*, 29 de abril. <<https://jasonwatmore.com/post/2019/04/29/angular-7-tutorial-part-3-add-routing-multiple-pages>> [Consulta: 20 de junio de 2020]
- [14] *Heroku*. <<https://www.heroku.com/platform>> [Consulta: 15 de julio de 2020]
- [15] Urrutia, V. (2018). “QUE ES HEROKU, PARA QUÉ SIRVE, VENTAJAS Y DESVENTAJAS” en *Videlcloud*, 22 de diciembre. <<https://videlcloud.wordpress.com/2018/12/22/que-es-heroku-para-que-sirve-ventajas-y-desventajas/>> [Consulta: 16 de junio de 2020]
- [16] *Heroku Help*. <<https://help.heroku.com/?t=true>> [Consulta: 20 de julio de 2020]
- [17] YOUTUBE, “MEAN Stack Front To Back [Part 10] - App Deployment to Heroku” en *Youtube* <<https://www.youtube.com/watch?v=cBfcbbo7Tqk>> [Consulta: 20 de julio de 2020]
- [18] Klement, O. (2020). “How to Deploy Your Angular 9 App to Heroku in Minutes” en *Medium*, 17 de febrero. <<https://medium.com/better-programming/how-to-deploy-your-angular-9-app-to-heroku-in-minutes-51d171c2f0d>> [Consulta: 25 de julio de 2020]
- [19] K6. (2013). “Know your node.js” en *K6*, 19 de marzo. <<https://k6.io/blog/know-your-node-js>> [Consulta: 29 de agosto de 2020]

[20] *Atlas Mo (Free Tier), M2, and M5 Limitations.* <<https://docs.atlas.mongodb.com/reference/free-shared-limitations/>> [Consulta: 29 de agosto de 2020]

[21] *Build apps for free on Heroku.* <<https://www.heroku.com/free>> [Consulta: 29 de agosto de 2020]

[22] Universidad Europea. (2019). “Cuánto gana un ingeniero informático” en *Universidad Europea.* <<https://universidadeuropea.es/blog/cuanto-gana-un-ingeniero-informatico>> [Consulta: 27 de agosto de 2020]

[23] *Universidad Politécnica de Valencia. RETRIBUCIONES PROFESORADO FUNCIONARIO EN BASE A LA LEY DE PRESUPUESTOS PARA EL EJERCICIO 2019.* <<http://www.upv.es/entidades/SRH/retribuciones/U0807596.pdf>> [Consulta: 27 de agosto de 2020]

[24] Gana Energía. (2018). “¿Qué consumo nos supone utilizar el ordenador?” en *Gana Energía*, 11 de noviembre. <<https://ganaenergia.com/blog/que-consumo-nos-supone-utilizar-el-ordenador/#:~:text=de%20nuestro%20aparato.-.Potencia%20ordenador%20kw,%2C%20unos%200%2C30%E2%82%AC.>> [Consulta: 27 de agosto de 2020]

[25] *Heroku Pricing.* <<https://www.heroku.com/pricing>> [Consulta: 27 de agosto de 2020]

[26] *Atlas Pricing Details.* <<https://www.mongodb.com/pricing>> [Consulta: 27 de agosto de 2020]

[27] *Stripe API.* <<https://stripe.com/docs/api>> [Consulta: 10 de agosto de 2020]

