



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una aplicación web para la gestión de
donaciones entre empresas y ONGs

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Pérez Martín, Ángel

Tutor/a: Herrero Cucó, Carlos

CURSO ACADÉMICO: 2022/2023

Agradecimientos

En primer lugar, me gustaría agradecer a Carlos Herrero Cucó, por haberme dado la confianza y la oportunidad de realizar este proyecto.

Por otro lado, me gustaría agradecer a mis padres, que me han estado apoyado y animado en cada decisión, los cuales espero que estén orgullosos de lo que he logrado, y también a mi familia.

Por último, agradecer a mis amigos de toda la vida y a compañeros de clase, que se han acabado convirtiendo mis amigos, que me han acompañado en estos 4 años de carrera, haciéndome los más llevaderos.

Resumen

Actualmente, el proceso de donación de bienes o consumibles a fundaciones, organizaciones de caridad, cooperación y desarrollo y no gubernamentales en general habitualmente ad-hoc. Esto se hace a base de llamamientos por emergencias que salen en medios de comunicación o de contactos directos por relaciones sociales o por redes. Las organizaciones tienen dificultades a la hora de contactar con empresas para ver si disponen de recursos para donar, y estas a su vez suelen realizar siempre las donaciones a las mismas organizaciones conocidas ya que no tienen constancia de las necesidades de otras, siendo este es el principal motivo del desarrollo de este proyecto. Por lo tanto, este proyecto tiene como objetivo crear una aplicación web en la que se facilite la comunicación entre ambas, permitiendo que puedan colaborar de una manera más eficiente, facilitando así las donaciones. Cualquier empresa que disponga de algún producto o recurso podrá subirlo a la aplicación, indicando las cantidades, precios estimados (por si es necesario para el cálculo de desgravaciones), etc., haciendo que la ONG interesada pueda contactar con la empresa para solicitar dicha donación. Asimismo, las ONGs podrán publicar las necesidades que tengan para que las empresas vean que es lo que se necesita en cada momento. La aplicación también contendrá información relevante para ambas partes, como datos sobre la distribución de las donaciones, la localización de las ONGs receptoras, buscando siempre garantizar una colaboración efectiva y segura. Para llevar a cabo el desarrollo del proyecto se ha aplicado una metodología de trabajo ágil, con la que se ha ido desarrollando el software de manera incremental, permitiendo asegurar la calidad del producto, buscando siempre la mejora continua y evitando posibles errores. Las tecnologías utilizadas han sido Microsoft .NET para el servidor y Angular para la interfaz web del usuario.

Palabras clave: aplicación web, ONGs, donaciones, empresas, cooperación, emprendimiento social, código abierto.



Resum

Actualment, el procés de donació de béns o consumibles a fundacions, organitzacions de caritat, cooperació i desenvolupament i no governamentals en general habitualment ad-hoc. Això es fa a base de crides per emergències que surten a mitjans de comunicació o de contactes directes per relacions socials o per xarxes. Les organitzacions tenen dificultats a l'hora de contactar amb empreses per veure si disposen de recursos per a donar, i aquestes al seu torn solen realitzar sempre les donacions a les mateixes organitzacions conegudes ja que no tenen constància de les necessitats d'altres, sent aquest és el motiu principal del desenvolupament d'aquest projecte. Per tant, aquest projecte té com a objectiu crear una aplicació web on es facilite la comunicació entre totes dues, permetent que puguin col·laborar d'una manera més eficient, facilitant així les donacions. Qualsevol empresa que dispose d'algun producte o recurs podrà pujar-lo a l'aplicació, indicant les quantitats, preus estimats (per si es necessari per al càlcul de desgravacions), etc., fent que l'ONG interessada pugui contactar amb l'empresa per sol·licitar aquesta donació. Així mateix, les ONG podran publicar les necessitats que tinguen perquè les empreses veguen què és el que es necessita a cada moment. L'aplicació també contindrà informació rellevant per a totes dues parts, com ara dades sobre la distribució de les donacions, la localització de les ONG receptores, buscant sempre garantir una col·laboració efectiva i segura. Per dur a terme el desenvolupament d'aquest projecte s'ha seguit una metodologia de treball àgil, amb la qual es podia anar desenvolupant el programari de manera incremental, permetent assegurar la qualitat del producte, buscant sempre la millora contínua i evitant possibles errors. On les tecnologies utilitzades han sigut Microsoft .NET per al servidor i Angular per a la interfície web de l'usuari.

Paraules clau: aplicació web, ONG, donacions, empreses, cooperació, empenedoria social, codi obert.

Abstract

Currently, the process of donating goods or consumables to foundations, charity, cooperation and development organizations and non-governmental organizations in general is usually ad-hoc. It is made based on calls for emergencies that appear in the media or through direct contacts on social relations or networks. Organizations have difficulties when it comes to contacting companies to see if they have resources to donate and these tend to always make donations to the same known organizations since they are not aware of the needs of others, this being the main reason for the development of this project. Therefore, this project aims to create a web application that facilitates communication between both, allowing them to collaborate more efficiently and thus facilitating donations. Any company that has a product or resource can upload it to the application, indicating the quantities, estimated prices (in case it is necessary for the calculation of tax relief), etc., allowing the interested NGO to contact the company to request the donation. Likewise, NGOs will be able to publish their needs so that companies can see what is needed at all times. The application will also contain relevant information for both parties, such as data about the distribution of donations, the location of the receiving NGOs, in order to guarantee effective and safe collaboration. To carry out this project an agile work methodology has been followed, in which the software could be developed incrementally and allowing to ensure the quality of the product, to always seek continuous improvement, avoiding possible errors. The technologies that have been used are Microsoft .NET for the server and Angular for the web user interface.

Keywords: web application, NGOs, donations, companies, cooperation, social entrepreneurship, open source.



Índice de contenidos

1.	Introducción	12
1.1.	Motivación	12
1.2.	Objetivos	12
1.3.	Metodología	13
2.	Estado del arte	16
2.1.	Análisis de la competencia	16
2.1.1.	Idealist	16
2.1.2.	HacesFalta	17
2.1.3.	GlobalGiving	18
2.2.	Conclusión.....	19
3.	Análisis del problema.....	21
3.1.	Diagrama de contexto.....	21
3.2.	Especificación de requisitos	22
3.2.1.	Requisitos funcionales.....	22
3.2.2.	Requisitos no funcionales.....	23
3.3.	Casos de uso	23
3.4.	Prototipos	37
4.	Diseño de la solución	41
4.1.	Arquitectura del sistema.....	41
4.2.	Diseño detallado.....	42
5.	Implementación.....	45
5.1.	Tecnologías utilizadas	45
5.1.1.	Visual Studio Code.....	45
5.1.2.	Visual Studio Enterprise.....	45
5.1.3.	Angular.....	46
5.1.4.	.Net	46
5.1.5.	Bootstrap	46
5.1.6.	Entity Framework.....	46
5.1.7.	GitHub.....	47
5.1.8.	Postman	47
5.2.	Desarrollo del Back-End	47

5.2.1.	Controladores	48
5.2.2.	Repositorios.....	49
5.2.3.	Entidades	52
5.2.4.	Base de datos.....	53
5.3.	Desarrollo del front-end	54
5.3.1.	Enrutamiento	54
5.3.2.	Componentes.....	55
5.3.3.	Servicios.....	56
6.	Despliegue.....	59
6.1.	Problemas del despliegue	61
7.	Pruebas	64
7.1.	Pruebas de compatibilidad	64
7.2.	Pruebas de rendimiento	65
7.3.	Pruebas funcionales.....	68
8.	Conclusiones y trabajos futuros	70
8.1.	Conclusiones	70
8.2.	Trabajos futuros.....	70
8.3.	Relación con los estudios	71
9.	Referencias	74
Anexo A	77
Anexo B	79
Anexo C	81



Índice de figuras

Figura 1.1. Tablero Kanban en Trello	14
Figura 2.1. Web de Idealist (Fuente: https://www.idealist.org/es).....	16
Figura 2.2. Web de HacesFalta (Fuente: https://www.hacesfalta.org/).....	17
Figura 2.3. Web de GlobalGiving (Fuente: https://www.globalgiving.org/)	18
Figura 3.1. Diagrama de contexto	21
Figura 3.2. Diagrama de caso de uso acceso a la web.....	24
Figura 3.3. Diagrama de caso de uso gestión del perfil del usuario.....	26
Figura 3.4. Diagrama de caso de uso gestión de los recursos	28
Figura 3.5. Diagrama de caso de uso gestión de las necesidades.....	31
Figura 3.6. Diagrama de caso de uso acceso a listados	34
Figura 3.7. Mock-up pantalla principal.....	37
Figura 3.8. Mock-up mi perfil.....	38
Figura 3.9. Mock-up crear recurso	38
Figura 3.10. Mock-up ver recursos	39
Figura 3.11. Mock-up ver empresas.....	39
Figura 4.1. Arquitectura del sistema (Fuente: www.bitacorasoftware.com).....	41
Figura 4.2. Modelo de datos.....	42
Figura 5.1. Controladores Back-End.....	48
Figura 5.2. Código RecursoController	48
Figura 5.3. Creación Token JWT	49
Figura 5.4. Interfaz Repositorio Recurso	50
Figura 5.5. Método Repositorio Recurso	50
Figura 5.6. Método envío de correo para empresa registrada	51
Figura 5.7. Entidades Proyecto	52
Figura 5.8. Objetos DTO.....	52
Figura 5.9. Mapeo Recurso	53
Figura 5.10. Enrutamiento de componentes	54
Figura 5.11. Estructura interfaz en app.component.....	55
Figura 5.12. Componentes front-end.....	55
Figura 5.13. EventEmitter método emit	56
Figura 5.14. EventEmitter método subscribe	56
Figura 5.15. Service Recurso	57
Figura 5.16. Constructor componente Pantalla Inicio.....	57
Figura 6.1. Web proveedor Azure	59

Figura 6.2. Servicios utilizados de Azure	60
Figura 6.3. Web EasyDonation	61
Figura 7.1. EasyDonation en Google Chrome	64
Figura 7.2. Web en iPhone 12 y Realme 7 Pro	65
Figura 7.3. Rendimiento Móvil y Ordenador PageSpeed	66
Figura 7.4. Rendimiento Gtmetrix	67
Figura B.1. Script para Empresas	79
Figura B.2. Script para ONGs	80
Figura C.1. Encabezado licencia GNU GPL	81



Índice de tablas

Tabla 2.1. Tabla análisis de la competencia.....	19
Tabla 3.1. Caso de uso Crear Cuenta	24
Tabla 3.2. Caso de uso iniciar sesión	25
Tabla 3.3. Caso de uso cerrar sesión	25
Tabla 3.4. Caso de uso recuperar contraseña	25
Tabla 3.5. Caso de uso ver mi perfil	26
Tabla 3.6. Caso de uso usuario registrado.....	27
Tabla 3.7. Caso de uso ver historial	27
Tabla 3.8. Caso de uso ver mis seguidos.....	27
Tabla 3.9. Caso de uso ver mis publicaciones.....	28
Tabla 3.10. Caso de uso ver recurso.....	29
Tabla 3.11. Caso de uso crear recurso.....	29
Tabla 3.12. Caso de uso modificar recurso	29
Tabla 3.13. Caso de uso eliminar recurso	30
Tabla 3.14. Caso de uso ver certificado	30
Tabla 3.15. Caso de uso solicitar donación.....	30
Tabla 3.16. Caso de uso aceptar donación	31
Tabla 3.17. Caso de uso ver necesidad.....	32
Tabla 3.18. Caso de uso crear necesidad.....	32
Tabla 3.19. Caso de uso modificar necesidad	32
Tabla 3.20. Caso de uso eliminar necesidad	33
Tabla 3.21. Caso de uso aceptar solicitud	33
Tabla 3.22. Caso de uso solicitar donación.....	33
Tabla 3.23. Caso de uso ver lista de donantes.....	34
Tabla 3.24. Caso de uso ver perfil donante	35
Tabla 3.25. Caso de uso ver lista de necesidades.....	35
Tabla 3.26. Caso de uso ver lista recursos	35
Tabla 3.27. Caso de uso ver lista beneficiarios	36
Tabla 3.28. Caso de uso ver perfil beneficiario.....	36
Tabla A.1. Relación ODS.....	77

1. Introducción

En la era actual, cada vez es más importante la colaboración entre las empresas y las organizaciones no gubernamentales (ONGs), con el fin de solucionar muchos de los problemas de nuestra sociedad. En este contexto, es imprescindible contar con la disponibilidad de herramientas para permitir la colaboración entre dichas entidades. Por lo que, en esta situación, este Trabajo de Fin de Grado se centra en el desarrollo de una aplicación web y responsive para móvil, cuyo propósito principal es facilitar la conectividad entre las entidades, proporcionando una plataforma donde las empresas y ONGs puedan colaborar de una manera sencilla y directa.

1.1. Motivación

Una de las principales motivaciones que me llevó a decantarme por este proyecto ha sido la idea de desarrollar una plataforma web destinada a brindar ayuda a la ONGs para que puedan contar con una gran cantidad de recursos para poder continuar desarrollando las labores sociales que hacen, ayudando en la mayoría de las situaciones a aquellos que más lo necesitan, ya que sentí que podía contribuir a mejorar un poco la sociedad.

Además, otra de las razones para adentrarme en este proyecto fue el hecho de no haber realizado nunca ningún proyecto sin fines lucrativos, con el único objetivo de ayudar. Motivándome así a investigar sobre este mundo, como pueden ser las tareas que realizan las ONGs, cuál es su funcionamiento, en que ámbitos actúan... Asimismo, otra de las razones que me impulsaron fue el reto de realizar un proyecto de principio a fin de forma independiente, en el que yo asumiese toda la responsabilidad del proyecto, mejorando así mi toma de decisiones.

El hecho de que no existiese ninguna plataforma previa para dicha colaboración aumentaba mi motivación para poder desarrollar dicho proyecto e intentar hacerlo de la mejor forma posible, por lo que podría ir aprendiendo nuevas tecnologías, herramientas, frameworks... ampliando así mi conocimiento en el mundo tecnológico y del desarrollo web, siendo esta otra de las principales razones.

1.2. Objetivos

El principal objetivo de desarrollo de este proyecto es crear una plataforma directa y sin intermediarios donde las empresas sean conscientes de lo que se necesita en las ONGs, y además ver e informarse sobre lo que hacen las mismas. Asimismo, se

permite que las empresas puedan publicar los recursos de los que disponen y que no saben qué utilidad darles, ya que, en muchos casos estos recursos acaban siendo desechados por no poder darles otra oportunidad, evitando así su desperdicio. Sumado a eso, desde el lado de las ONGs es muy importante la posibilidad que se les brinda con esta plataforma, sobre todo para las de pequeña envergadura ya que tienen grandes dificultades en encontrar donantes. Estas podrán ver los recursos que ofrecen las empresas, solicitarlos y ponerse en contacto con las empresas.

Otro de los objetivos clave es hacer una aplicación sencilla e intuitiva, de modo que cualquier persona encargada de manejar tanto el perfil de empresa u ONG sepa como solicitar las donaciones e interactuar con la web. Por lo tanto, con esta idea, es indispensable hacer un diseño amigable para el usuario y responsive, permitiendo así acceder a la plataforma tanto desde un ordenador como desde dispositivos móviles.

Además, se fomentará el uso de la licencia GNU GPL¹ (General Public License), la cual permite que cualquier persona que esté interesada en seguir ampliando o manteniendo el proyecto por su cuenta, puede hacerlo libremente siempre que siga cumpliendo las condiciones de la licencia GPL. Con este objetivo se busca garantizar que el software desarrollado se pueda seguir utilizando, siendo accesible para cualquier persona.

1.3. Metodología

A lo largo del desarrollo de este proyecto se ha implementado una metodología ágil, la cual permite una mejor toma de decisiones, incluyendo una mayor flexibilidad y adaptabilidad ante posibles cambios, además de una mayor rapidez y control sobre el proyecto. Por tanto, una de las características más aprovechadas de la metodología es el Producto Mínimo Viable (MVP), ya que permite ir viendo como avanza el proyecto e ir realizando evaluaciones del mismo, haciendo así un uso del desarrollo incremental.

Para llevar a cabo dicha metodología, se utilizará un tablero Kanban [2] mediante el cual se puede obtener una perspectiva global del estado del proyecto. Dicho tablero está compuesto por varias columnas: “To Do”, “In Progress” y “Done”, en las cuales se colocan las tareas indicando en qué estado están y avanzando según se progresa en el flujo de trabajo. Así se comienza con “To Do”, donde se colocan las tarjetas que están pendientes de realizar, “In Progress” las tareas que se están en curso en estos momentos y las que ya se han completado, que se marcan como “Done”.

Este tablero de organización de proyectos también proporciona un marco para establecer y gestionar el orden de prioridades sobre qué es lo que se quiere desarrollar antes, puesto que habrá tareas que tengan una prioridad mayor que el resto.

¹ Consultar Anexo C para más información

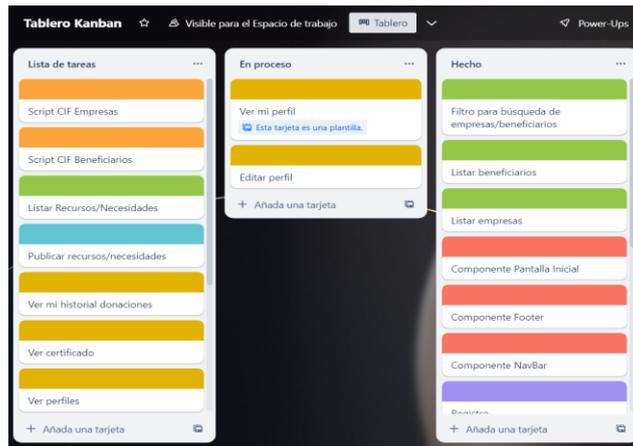


Figura 1.1. Tablero Kanban en Trello

Para poder llevar el tablero Kanban se ha utilizado la herramienta gratuita de gestión y administración de software Trello², siendo esto apreciado en la figura 1.1. En este escenario se pueden ver las tres columnas anteriormente mencionadas, asimismo, en el caso de que fuese necesario añadir alguna columna extra se podría crear sin problemas.

² Trello es una herramienta que permite la gestión y organización de proyectos de una forma visual y muy intuitiva, utilizando el tablero Kanban como eje principal. <https://trello.com/es>

2. Estado del arte

Para esta sección se ha llevado a cabo un estudio de mercado en el que se han buscado plataformas que tengan un objeto o idea similar al de este proyecto. El nombre de la plataforma desarrollada en este Trabajo de Fin de Grado es EasyDonation y con ella se realizará el proceso de comparación descrito en esta sección, cabe destacar la inexistencia de ninguna plataforma cuyos objetivos o funcionalidades coincidan con las de este proyecto por lo que se buscaron los proyectos más similares.

2.1. Análisis de la competencia

2.1.1. Idealist

En primer lugar, se encuentra Idealist³, una plataforma que nació en Nueva York en 1995 pero que actualmente se utiliza en todo el mundo. Idealist tiene unos objetivos e ideas principales muy claros, la necesidad de conectar a las personas con las ONGs, ya que su idea principal es la creación de un entorno inclusivo para todo el mundo independientemente de su identidad racial, sexo, nacionalidad... Además, es una aplicación muy sencilla de usar, muy directa y que no requiere de conocimientos previos para manejarla.



Figura 2.1. Web de Idealist (Fuente: <https://www.idealist.org/es>)

Una de las características más relevantes de la web son las oportunidades de empleo, siendo las organizaciones quienes publican avisos para que todas las personas sin tener que depender de sus características puedan acceder a estos puestos. Adicionalmente, también se pueden realizar publicaciones de voluntariado, permitiendo así que las organizaciones tengan un mayor rango de visibilidad y puedan mantenerse.

³ Idealist <https://www.idealist.org/es>

Otra de sus características es la posibilidad de interactuar y establecer conexiones directamente entre los usuarios y las organizaciones.

2.1.2. HacesFalta

Otra de las plataformas vistas en el mercado es HacesFalta⁴. Esta organización, cuenta con el apoyo del Gobierno de España, lo que le permite disponer de una mayor cantidad herramientas y recursos que el resto de proyectos. La idea principal de HacesFalta es la colaboración entre las ONGs y los usuarios.

Su principal característica es la sección de voluntarios, donde cualquier individuo que esté interesado en el tema en cuestión o en una organización puede solicitar dicho voluntariado, incluyendo sus capacidades, conocimientos, etc.

Otra de las características principales de esta plataforma está relacionada con el ámbito empresarial, ya que la página permite a las empresas publicar las vacantes para sus puestos de trabajo, sin embargo, esta les solicita un pequeño coste para poder mantenerse. Con esta característica, este proyecto ha llegado a superar la cifra de 9 millones de oportunidades de empleos entre empresas y ONGs.



The image shows a screenshot of the HacesFalta website. At the top left, there is the logo for 'hacesfalta.org' with the tagline 'Fundación hazlo posible' and a circular badge indicating '100% transparente'. To the right, there are navigation links: 'VOLUNTARIADO', 'EMPLEO', 'DONACIONES', and 'ACCESO / REGISTRO'. Below these, a green button labeled 'DONACIONES' is visible. The main content area is titled 'MATERIAL ACAMPADA' and is promoted by 'Asociación La Frontera'. It features a 'Formulario de donación' with instructions: 'Vas a colaborar con este proyecto. Por favor, introduce la cantidad que quieres donar y luego pulsa el botón "DONAR CON TARJETA".' The form includes a dropdown menu for 'INTRODUCE LA CANTIDAD A DONAR' with the text 'Selecciona una cantidad', a text input field for 'OTRA CANTIDAD', and a prominent pink button labeled 'DONAR CON TARJETA'. Below the button are logos for Mastercard, American Express, and Visa. A 'Privacidad' link is located at the bottom right of the form area.

Figura 2.2. Web de HacesFalta (Fuente: <https://www.hacesfalta.org/>)

Por último, la funcionalidad menos utilizada corresponde al apartado de las donaciones, en el que las ONGs publican el evento o la necesidad por la que solicitan la donación y los usuarios pueden donar a las organizaciones que les sean de su interés.

⁴ HacesFalta <https://www.hacesfalta.org/>

2.1.3. GlobalGiving

En último lugar, se encuentra la plataforma de GlobalGiving⁵, siendo fundada en el 2002, actúa como una red encargada de unir a las empresas, donantes y ONGs de todo el mundo, con el fin de proporcionar herramientas, recursos y apoyo para que dichas entidades puedan desarrollar al máximo las capacidades de su trabajo.

Una de las funciones más interesantes de esta plataforma son las herramientas internas de las que dispone, con estas se pueden proporcionar a las organizaciones registradas. Permitiendo generar y obtener fondos para estas. Además, se da pie a un asesoramiento exclusivo que permite conectar a las organizaciones con el resto de las organizaciones o empresas que necesiten.

Además, GlobalGiving permite a las ONGs publicitar sus proyectos o necesidades, permitiendo así tener un mayor rango de visibilidad tanto para los usuarios como para las empresas que deseen realizar una donación, como se puede ver en la figura 2.3.

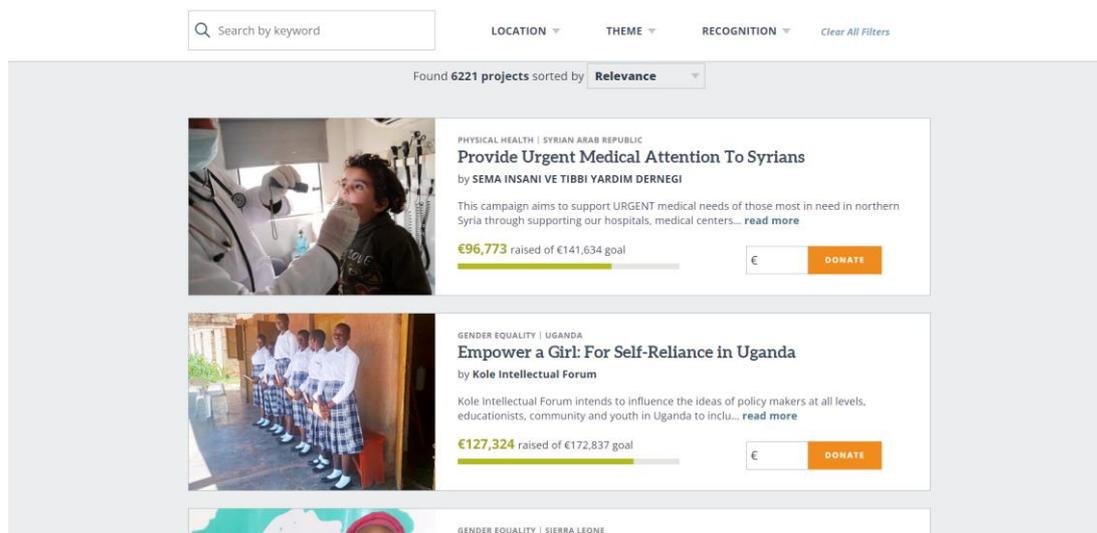


Figura 2.3. Web de GlobalGiving (Fuente: <https://www.globalgiving.org/>)

Otra de las características más diferenciales de esta aplicación es la posibilidad de hacer tarjetas regalo, a las cuales se les añade la cantidad deseada de dinero y el mensaje que el usuario desee, pudiendo donarlas a las organizaciones que están en la plataforma o enviarlas a la casa del usuario para que este pueda darlas a una organización en mano.

⁵ GlobalGiving <https://www.globalgiving.org/>

2.2. Conclusión

En definitiva, se puede observar que no hay ninguna plataforma web cuya función principal sea la conexión directa y sin intermediarios entre las empresas y las ONGs. La gran mayoría de plataformas web se centran en la búsqueda de voluntarios para las ONGs y la publicación de las necesidades para que los usuarios realicen donaciones. Cabe destacar que la única que podría asemejarse al proyecto desarrollado es GlobalGiving, pero sigue sin ser un competidor real, ya que, no se centra como tal en esa conexión directa, sino que esta más centrada en las donaciones y en los voluntariados, a pesar de tener ese asesoramiento para las ONGs. Por lo que, como resultado de este análisis, se puede determinar que EasyDonation no tiene competidores en el mercado actualmente.

A continuación, se muestra una matriz comparativa de los competidores con relación a nuestra aplicación.

Tabla 2.1. Tabla análisis de la competencia

Características	Idealist	HacesFalta	GlobalGiving	EasyDonation
Destinada a	Empresas, ONGs y usuarios	Empresas, ONGs y usuarios	Empresas, ONGs y usuarios	Empresas y ONGs
Interfaz y usabilidad	Muy agradable	Poco agradable	Agradable	Muy agradable
Coste de uso	Gratis	Gratis, menos empresas	Gratis	Gratis
Conecta empresas y ONGs	No	No	Sí	Sí
Poder seguir a empresas u ONGs	Sí	No	No	Sí
Contactar con empresas u ONGs	Sí	Sí	No	Sí
Ver recursos de empresas	No	No	No	Sí
Certificado donación	No	No	No	Sí
Contacto con la plataforma	Sí	No	Sí	Sí
Nivel geográfico	Global	España	Global	España

3. Análisis del problema

A lo largo de este capítulo se presenta el diagrama de contexto con sus actores, los requisitos funcionales y no funcionales, y los diagramas de casos de uso, dando así una imagen global de nuestra aplicación para una mayor comprensión de su funcionalidad y objetivos.

3.1. Diagrama de contexto

Con respecto a los diagramas de contexto, gracias a estos se permite visualizar la relación del sistema con sus usuarios, sistemas o dispositivos. Con él, se pueden conocer los límites del sistema y así definir tanto los requisitos funcionales, como los no funcionales, los cuales se comentarán en el siguiente apartado. Es importante resaltar que el diagrama de contexto no muestra la funcionalidad ni como son sus interacciones a nivel interno.

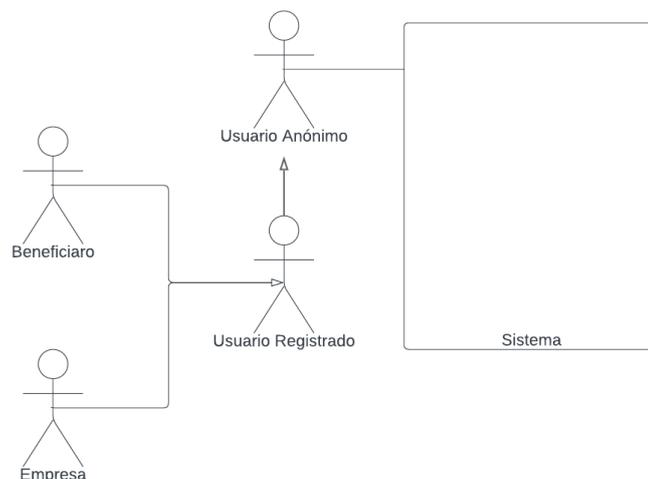


Figura 3.1. Diagrama de contexto

En el caso del diagrama de contexto de EasyDonation, figura 3.1, se puede observar la interacción directa de tres tipos de usuarios con el sistema. En primer lugar, se encuentra el usuario del que hereda el resto, el anónimo, este es quien puede acceder a la web simplemente para informarse o simplemente sea un posible futuro usuario que aún no está registrado. Por otro lado, se encuentran a los usuarios registrados, estos se dividen en dos, beneficiarios, es decir, las ONGs, y las empresas.

3.2. Especificación de requisitos

En este apartado de la memoria se especifican y se describen tanto los requisitos funcionales como no funcionales. Los primeros se basan en las acciones y las características que el sistema debe poder realizar, ya sea, crear un usuario, ver el perfil del usuario o eliminarlo. Sin embargo, los requisitos no funcionales son conocidos como las restricciones del sistema, cuyo eje se centra en los aspectos de calidad del software, como puede ser la escalabilidad, rendimiento, seguridad...

Algo a destacar de los requisitos de este proyecto es que se han acordado y se validaron con la *Fundación Juntos por la Vida*⁶. Para llevar a cabo esta tarea se mantuvieron varias sesiones de trabajo en la que se realizó un proceso de recopilación de necesidades y características para la plataforma con una persona de una ONG, que sería usuario de la plataforma.

3.2.1. Requisitos funcionales

Respecto a los requisitos funcionales, en la sesión de trabajo, se pudieron obtener las siguientes características que reflejaban la necesidad del representante:

- Un usuario sin registrar podrá acceder a la lista de empresas y beneficiarios asociados a la plataforma.
- Un usuario sin registrar podrá acceder a los recursos y necesidades publicados por las empresas y beneficiarios respectivamente.
- Un usuario con sus credenciales podrá iniciar sesión.
- Un usuario podrá registrar su empresa o ONG con sus correspondientes datos.
- Un usuario registrado podrá recuperar su contraseña en caso de pérdida.
- Un usuario registrado podrá ver su perfil.
- Un usuario registrado podrá modificar su perfil.
- Un usuario registrado podrá ver su historial de donaciones.
- Un usuario registrado podrá ver su lista de seguidos.
- Una empresa podrá crear, editar o borrar un recurso.
- Una empresa podrá ver sus recursos.
- Un beneficiario podrá crear, editar o borrar una necesidad.

⁶ Fundación Juntos por la Vida <http://www.juntosporlavida.org/>

- Un beneficiario podrá ver sus necesidades.
- Un beneficiario podrá seguir, dejar de seguir a las empresas.
- Una empresa podrá seguir, dejar de seguir a los beneficiarios.
- Una empresa podrá ver su certificado de donación.
- Un usuario registrado podrá solicitar o aceptar la donación de dicho recurso o necesidad.
- Los usuarios registrados recibirán notificaciones cuando se les solicite su recurso o necesidad.

3.2.2. Requisitos no funcionales

En cuanto a los requisitos no funcionales, se pueden destacar los más importantes en esta lista:

- Usabilidad: la interfaz de la aplicación debe ser intuitiva y sencilla para que los representantes de las ONGs y empresas sepan cómo utilizarla.
- Seguridad: este es uno de los atributos más importantes, debido a que la plataforma dispone de los datos sensibles de las donaciones y de los usuarios registrados, además de tener que gestionar los permisos y el acceso a las rutas.
- Rendimiento: la aplicación debe ser capaz de mantener un buen nivel, asegurando así una experiencia al usuario fluida, a pesar de un número elevado de usuarios.
- Disponibilidad: la aplicación debe estar accesible el mayor tiempo posible, evitando así cualquier problema de inactividad o de tiempo.
- Accesibilidad: la interfaz de la aplicación debe estar disponible en el móvil para facilitar así su uso.
- Eficiencia: la plataforma debe devolver los datos solicitados en el menor tiempo posible, para evitar esperas innecesarias.

3.3. Casos de uso

A continuación, se representan los casos de uso y sus respectivos diagramas para aportar una visión más amplia y global de la interacción del usuario con el sistema. Cada uno de estos casos de uso describe los pasos a seguir por el usuario para poder llevar la tarea a cabo, es decir, describe que es lo que hace el sistema, pero no el cómo.

Además, los casos de uso son utilizados para facilitar la comunicación con los clientes, usuarios... ya que, su representación es visual y permite una mejor comprensión. Asimismo, es muy útil para la faceta de realización de pruebas, puesto que en los casos de uso se establece un orden de interacción y un resultado esperado, pudiendo llegar a ser utilizado para las pruebas de aceptación.

Acceso a la web

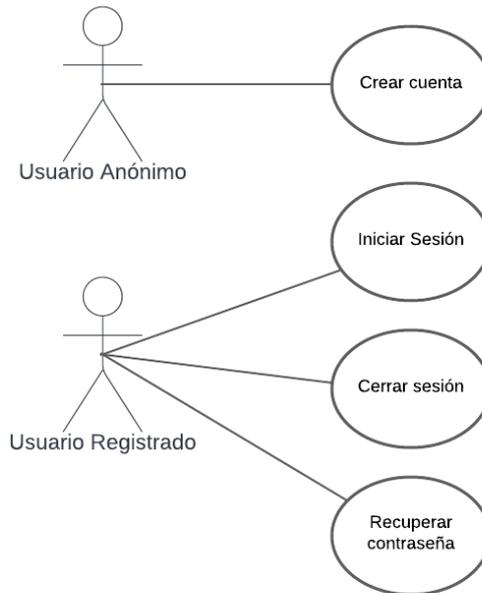


Figura 3.2. Diagrama de caso de uso acceso a la web

Tabla 3.1. Caso de uso Crear Cuenta

Caso de uso	Crear cuenta.	CU-01
Actor	Usuario anónimo.	
Descripción	El usuario introduce los datos solicitados para poder crear una cuenta en la aplicación.	
Precondición	No.	
Secuencia Principal	<ol style="list-style-type: none"> 1. El usuario introduce los datos. 2. El sistema comprueba los datos introducidos. 3. Se crea una cuenta. 	
Postcondición	El usuario dispone de una cuenta en la web.	

Tabla 3.2. Caso de uso iniciar sesión

Caso de uso	Iniciar sesión.	CU-02
Actor	Usuario registrado.	
Descripción	El usuario accede a la web con sus credenciales.	
Precondición	El usuario disponga de sus propias credenciales.	
Secuencia Principal	<ol style="list-style-type: none"> 1. El usuario introduce los datos. 2. El sistema valida los datos introducidos. 3. El sistema le da acceso a la web como usuario. 	
Postcondición	El usuario accede a la web con su cuenta.	

Tabla 3.3. Caso de uso cerrar sesión

Caso de uso	Cerrar sesión.	CU-03
Actor	Usuario registrado.	
Descripción	El usuario abandona la sesión de la web.	
Precondición	El usuario debe haber iniciado sesión previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de cerrar sesión. 2. El sistema elimina los datos de la sesión y la finaliza. 3. Se devuelve al usuario a la pantalla principal. 	
Postcondición	Se finaliza la sesión al usuario.	

Tabla 3.4. Caso de uso recuperar contraseña

Caso de uso	Recuperar contraseña.	CU-04
Actor	Usuario registrado.	
Descripción	El usuario olvida la contraseña y desea recuperarla.	
Precondición	El usuario debe disponer de credenciales.	

Secuencia principal	<ol style="list-style-type: none"> 1. El usuario solicita al sistema la recuperación de su contraseña introduciendo su correo. 2. El sistema le notificará al usuario con sus credenciales.
Postcondición	El usuario obtiene la contraseña.

Gestión del perfil del usuario

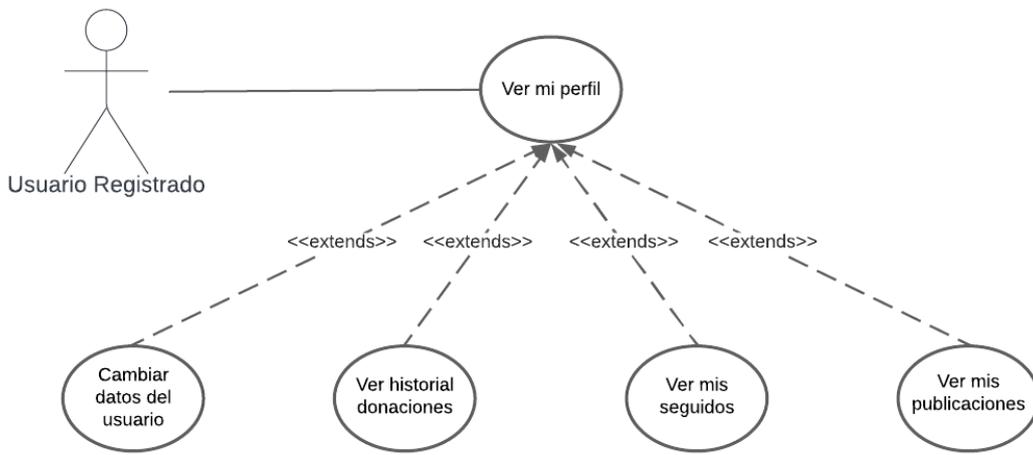


Figura 3.3. Diagrama de caso de uso gestión del perfil del usuario

Tabla 3.5. Caso de uso ver mi perfil

Caso de uso	Ver mi perfil.	CU-05
Actor	Usuario registrado.	
Descripción	El usuario desea acceder a su perfil.	
Precondición	El usuario debe haber iniciado sesión previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver mi perfil. 2. El sistema le redirige a su perfil. 	
Postcondición	El usuario accede al perfil con toda su información.	

Tabla 3.6. Caso de uso usuario registrado

Caso de uso	Cambiar datos del usuario.	CU-06
Actor	Usuario registrado.	
Descripción	El usuario desea modificar el nombre de su cuenta.	
Precondición	El usuario debe acceder al perfil previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de modificar perfil. 2. El usuario modifica los datos que desee. 	
Postcondición	Los datos del usuario son modificados correctamente.	

Tabla 3.7. Caso de uso ver historial

Caso de uso	Ver historial.	CU-07
Actor	Usuario registrado.	
Descripción	El usuario desea acceder a su historial de donaciones.	
Precondición	El usuario debe haber accedido al perfil previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver el historial de donaciones. 2. El sistema le muestra la lista con todas sus donaciones recibidas/enviadas. 	
Postcondición	El usuario obtiene la lista de donaciones.	

Tabla 3.8. Caso de uso ver mis seguidos

Caso de uso	Ver mis seguidos.	CU-08
Actor	Usuario registrado.	
Descripción	El usuario desea acceder a su lista de seguidos.	
Precondición	El usuario debe acceder al perfil previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver la lista de seguidos. 2. El sistema le muestra la lista con todas las entidades seguidas. 	

Postcondición	El usuario obtiene la lista de seguidos.
----------------------	--

Tabla 3.9. Caso de uso ver mis publicaciones

Caso de uso	Ver mis publicaciones.	CU-09
Actor	Usuario registrado.	
Descripción	El usuario desea acceder a sus publicaciones.	
Precondición	El usuario debe acceder al perfil previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ver sus publicaciones. 2. El sistema le muestra la lista con todas sus publicaciones. 	
Postcondición	El usuario obtiene la lista de publicaciones.	

Gestión de los recursos

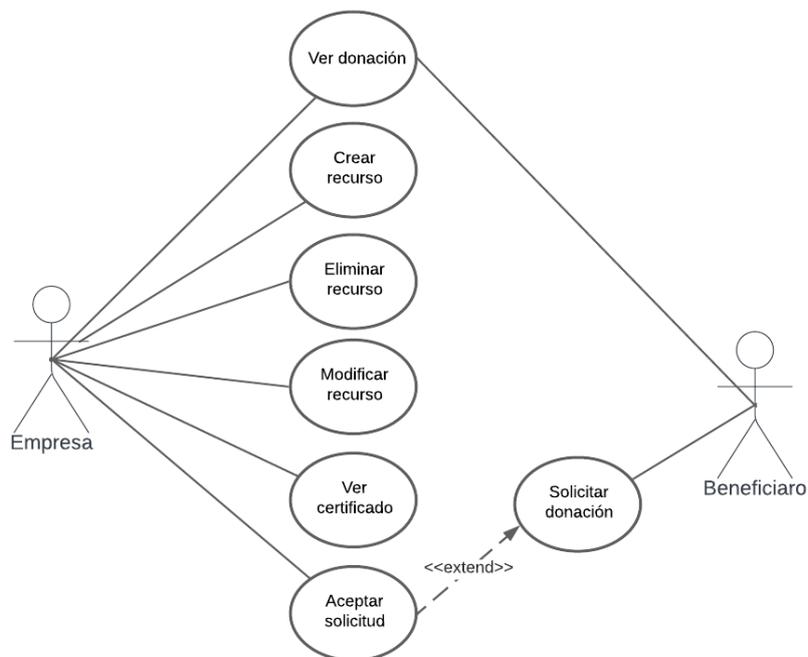


Figura 3.4. Diagrama de caso de uso gestión de los recursos

Tabla 3.10. Caso de uso ver recurso

Caso de uso	Ver recurso.	CU-10
Actor	Empresa/Beneficiario.	
Descripción	El actor desea ver la información del recurso.	
Precondición	El recurso debe estar creado previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El actor selecciona un recurso. 2. El sistema le muestra el recurso y su información correspondiente. 	
Postcondición	El usuario visualiza la información del recurso.	

Tabla 3.11. Caso de uso crear recurso

Caso de uso	Crear recurso.	CU-11
Actor	Empresa.	
Descripción	La empresa desea crear un recurso.	
Precondición	La empresa debe estar dada de alta en el sistema.	
Secuencia principal	<ol style="list-style-type: none"> 1. La empresa selecciona la opción de crear un recurso. 2. La empresa rellena la información correspondiente al recurso. 3. El sistema publica el recurso. 	
Postcondición	El recurso se crea y publica en la aplicación.	

Tabla 3.12. Caso de uso modificar recurso

Caso de uso	Modificar recurso.	CU-12
Actor	Empresa.	
Descripción	La empresa desea modificar un recurso.	
Precondición	La empresa debe haber creado ese recurso.	
Secuencia principal	<ol style="list-style-type: none"> 1. La empresa selecciona la opción de modificar el recurso. 	

	<ol style="list-style-type: none"> 2. La empresa modifica la información correspondiente al recurso. 3. El sistema publica el recurso modificado.
Postcondición	EL recurso se modifica y actualiza en la aplicación.

Tabla 3.13. Caso de uso eliminar recurso

Caso de uso	Eliminar recurso.	CU-13
Actor	Empresa.	
Descripción	La empresa desea eliminar un recurso.	
Precondición	La empresa debe haber creado ese recurso.	
Secuencia principal	<ol style="list-style-type: none"> 1. La empresa selecciona la opción de eliminar el recurso. 2. El sistema pregunta si está seguro de eliminarlo y si es así lo elimina. 	
Postcondición	El recurso se elimina y desaparece de la aplicación.	

Tabla 3.14. Caso de uso ver certificado

Caso de uso	Ver certificado.	CU-14
Actor	Empresa.	
Descripción	Se desea acceder al certificado de donación, el cual acredita que se ha realizado la donación a un beneficiario.	
Precondición	Se debe haber realizado una donación a un beneficiario y esta debería aceptar su obtención.	
Secuencia principal	<ol style="list-style-type: none"> 1. La empresa accede a la donación para ver su información. 2. Selecciona el apartado de ver certificado. 	
Postcondición	La empresa puede acceder al certificado y descargárselo.	

Tabla 3.15. Caso de uso solicitar donación

Caso de uso	Solicitar donación.	CU-15
Actor	Beneficiario.	

Descripción	El beneficiario ve una donación que ofrece una empresa y envía una solicitud para poder recibirla.
Precondición	Una empresa debe haber creado previamente una donación.
Secuencia principal	<ol style="list-style-type: none"> 1. El beneficiario accede a la lista de donaciones. 2. Selecciona la donación que le interesa y la solicita.
Postcondición	Se ha realizado la solicitud correctamente.

Tabla 3.16. Caso de uso aceptar donación

Caso de uso	Aceptar donación.	CU-16
Actor	Empresa.	
Descripción	La empresa recibe la solicitud de donación y decide aceptarla.	
Precondición	Un beneficiario debe haber solicitado la donación.	
Secuencia principal	<ol style="list-style-type: none"> 1. La empresa ve que tiene una solicitud de donación y accede a esta. 2. La empresa acepta dicha solicitud o deniega si ve que no es correcto. 	
Postcondición	La donación es aceptada y se le notifica al beneficiario.	

Gestión de las necesidades

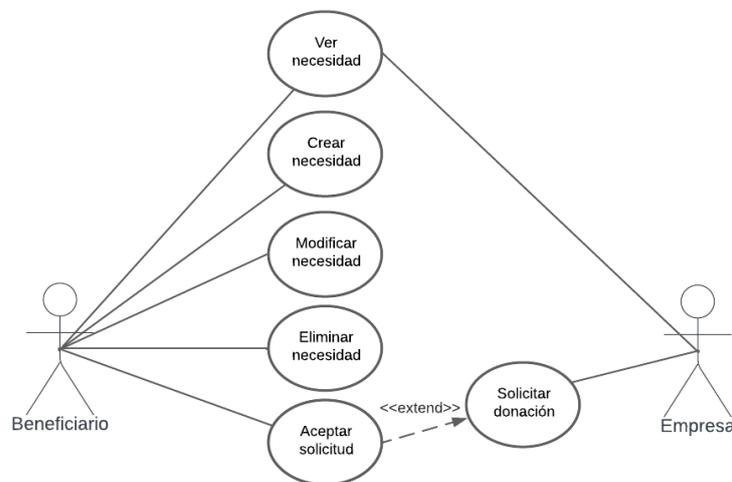


Figura 3.5. Diagrama de caso de uso gestión de las necesidades

Tabla 3.17. Caso de uso ver necesidad

Caso de uso	Ver necesidad.	CU-17
Actor	Beneficiario/Empresa.	
Descripción	El actor desea ver la información de la necesidad.	
Precondición	La necesidad debe estar creada previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El actor selecciona una necesidad. 2. El sistema le muestra la necesidad y su información correspondiente. 	
Postcondición	El usuario visualiza la información de la necesidad.	

Tabla 3.18. Caso de uso crear necesidad

Caso de uso	Crear necesidad.	CU-18
Actor	Beneficiario.	
Descripción	El beneficiario desea crear una necesidad.	
Precondición	El beneficiario debe estar dada de alta en el sistema.	
Secuencia principal	<ol style="list-style-type: none"> 1. El beneficiario selecciona la opción de crear una necesidad. 2. El beneficiario rellena la información correspondiente a la necesidad. 3. El sistema publica la necesidad. 	
Postcondición	La necesidad se crea y publica en la aplicación.	

Tabla 3.19. Caso de uso modificar necesidad

Caso de uso	Modificar necesidad.	CU-19
Actor	Beneficiario.	
Descripción	El beneficiario desea modificar una necesidad.	
Precondición	El beneficiario debe haber creado esa necesidad.	
Secuencia principal	<ol style="list-style-type: none"> 1. El beneficiario selecciona la opción de modificar la necesidad. 	

	<ol style="list-style-type: none"> 2. El beneficiario modifica la información correspondiente a la necesidad. 3. El sistema publica la necesidad modificada.
Postcondición	La necesidad se modifica y actualiza en la aplicación.

Tabla 3.20. Caso de uso eliminar necesidad

Caso de uso	Eliminar necesidad.	CU-20
Actor	Beneficiario.	
Descripción	El beneficiario desea eliminar una necesidad.	
Precondición	El beneficiario debe haber creado esa necesidad	
Secuencia principal	<ol style="list-style-type: none"> 1. El beneficiario selecciona la opción de eliminar la necesidad. 2. El sistema pregunta si está seguro de eliminarla y si es así la elimina. 	
Postcondición	La necesidad se elimina y desaparece de la aplicación.	

Tabla 3.21. Caso de uso aceptar solicitud

Caso de uso	Aceptar solicitud.	CU-21
Actor	Beneficiario.	
Descripción	El beneficiario recibe la solicitud de donación y decide aceptarla.	
Precondición	Una empresa debe haber solicitado la donación.	
Secuencia principal	<ol style="list-style-type: none"> 1. El beneficiario ve que tiene una solicitud de donación y accede a esta. 2. El beneficiario acepta dicha solicitud o deniega si ve que no es correcto. 	
Postcondición	La donación es aceptada y se le notifica a la empresa.	

Tabla 3.22. Caso de uso solicitar donación

Caso de uso	Solicitar donación.	CU-22
Actor	Empresa.	



Descripción	La empresa ve una necesidad que solicita un beneficiario y envía una solicitud para poder enviar los recursos.
Precondición	Un beneficiario debe haber creado previamente una necesidad.
Secuencia principal	<ol style="list-style-type: none"> 1. La empresa accede a la lista de necesidades. 2. Selecciona la necesidad que le interesa y la solicita.
Postcondición	Se ha realizado la solicitud correctamente.

Acceso a listados

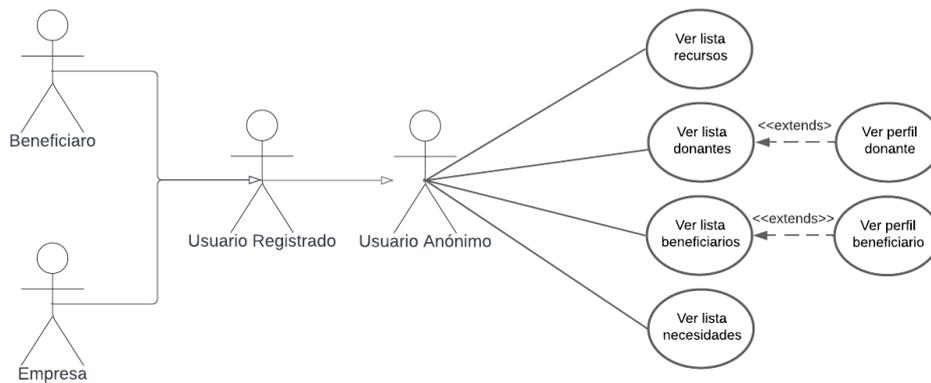


Figura 3.6. Diagrama de caso de uso acceso a listados

Tabla 3.23. Caso de uso ver lista de donantes

Caso de uso	Ver lista de donantes.	CU-23
Actor	Usuario anónimo.	
Descripción	Un usuario accede a la web y desea acceder a la lista de donantes con su correspondiente información.	
Precondición	No.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. El usuario selecciona el apartado de ver lista de donantes. 3. El sistema le muestra la lista. 	
Postcondición	El usuario puede visualizar la lista de donantes.	

Tabla 3.24. Caso de uso ver perfil donante

Caso de uso	Ver perfil donante.	CU-24
Actor	Usuario anónimo.	
Descripción	Si un usuario está interesado en obtener más información del donante, puede acceder al perfil y saber más de este.	
Precondición	Se deben haber listado los donantes previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona una empresa donante. 2. El sistema le muestra el perfil. 	
Postcondición	El usuario tiene acceso al perfil del donante.	

Tabla 3.25. Caso de uso ver lista de necesidades

Caso de uso	Ver lista necesidades.	CU-25
Actor	Usuario anónimo.	
Descripción	Un usuario accede a la lista de necesidades a ver lo que los beneficiarios necesitan.	
Precondición	No.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede al apartado de la web de necesidades. 2. El sistema le muestra las necesidades de los beneficiarios. 	
Postcondición	El usuario puede visualizar las necesidades.	

Tabla 3.26. Caso de uso ver lista recursos

Caso de uso	Ver lista recursos.	CU-26
Actor	Usuario anónimo.	
Descripción	Un usuario accede a la lista de recursos a ver lo que las empresas ofrecen.	
Precondición	No.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede al apartado de la web de los recursos. 	

	2. El sistema le muestra los recursos de las empresas.
Postcondición	El usuario puede visualizar los recursos.

Tabla 3.27. Caso de uso ver lista beneficiarios

Caso de uso	Ver lista beneficiarios.	CU-27
Actor	Usuario anónimo.	
Descripción	Un usuario accede a la web y desea acceder a la lista de beneficiarios con su correspondiente información.	
Precondición	No.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario accede a la web. 2. El usuario selecciona el apartado de ver lista de beneficiarios. 3. El sistema le muestra la lista. 	
Postcondición	El usuario puede visualizar la lista de beneficiarios.	

Tabla 3.28. Caso de uso ver perfil beneficiario

Caso de uso	Ver perfil beneficiario.	CU-28
Actor	Usuario anónimo.	
Descripción	Si un usuario está interesado en obtener más información del beneficiario, puede acceder al perfil y saber más de este.	
Precondición	Se deben haber listado los beneficiarios previamente.	
Secuencia principal	<ol style="list-style-type: none"> 1. El usuario selecciona un beneficiario. 2. El sistema le muestra el perfil. 	
Postcondición	El usuario tiene acceso al perfil del beneficiario.	

3.4. Prototipos

Una vez ya se ha explicado el diagrama de contexto, los requisitos funcionales y no funcionales y los casos de uso, se continua con los prototipos de la aplicación. El prototipado es una faceta sumamente relevante en el desarrollo del software, gracias a este se permiten validar los requisitos funcionales antes de su posterior implementación, así como el diseño y su usabilidad respecto a los usuarios que la utilicen. Por tanto, se consigue una gran reducción de los riesgos y una mejora del producto final, en este caso se han escogido los mock-ups como tipo de prototipado.

Los mock-ups son dibujos que representan los posibles diseños que va a tener la interfaz de la aplicación. Suelen ser representaciones de bajo nivel de complejidad y se pueden utilizar herramientas gráficas o de dibujo simples, como puede ser Pencil⁷ [22], que es con la que se ha trabajado, o simplemente esbozando dibujos a mano.

A continuación, se expondrán seis mock-ups representativos de la interfaz del usuario, es decir, los más relevantes, ya que en algunos casos la diferencia que puede haber entre un boceto y otro es mínima. Por tanto, se ha decidido mostrar uno de cada tipo, evitando así la duplicidad y que sean muy pesados de visualizar.

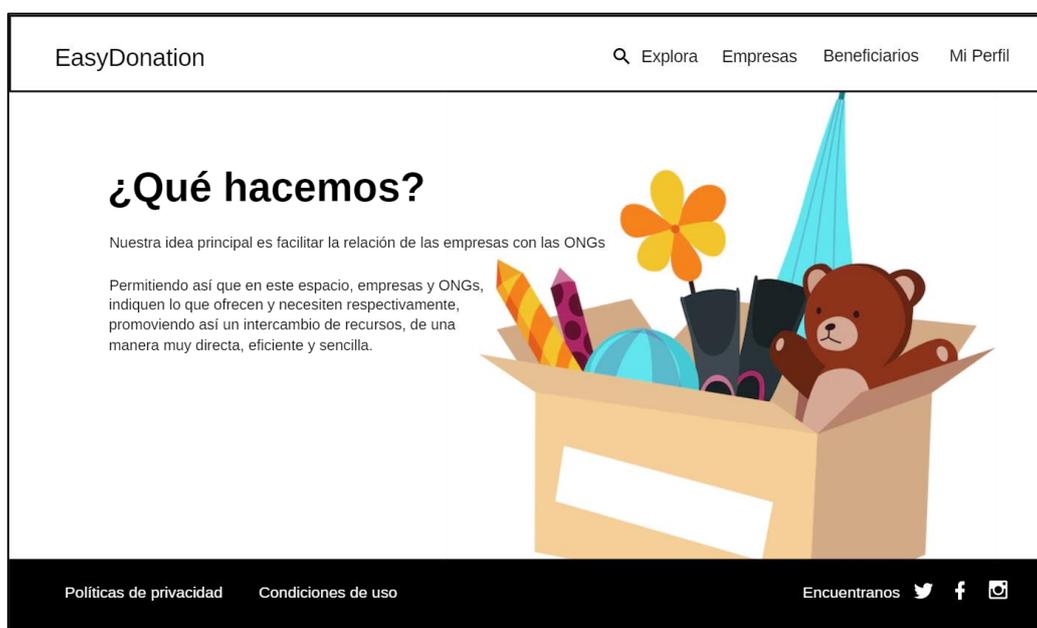


Figura 3.7. Mock-up pantalla principal

En la figura 3.7 se puede visualizar la pantalla inicial de la plataforma, desde la que se pueden realizar varios casos de uso como ver mi perfil, listar empresas o beneficiarios y explorar los recursos o necesidades. Todo esto se puede realizar en la *navbar*, es decir, la barra de navegación de la aplicación, que se encuentra en la parte

⁷ Pencil es una herramienta de software libre utilizada para crear prototipos de interfaces de manera sencilla y rápida, con la que se obtienen mock-ups con un estilo de dibujo. <https://pencil.evolus.vn/>

superior de la ventana y estará visible desde cualquier lugar de la plataforma, sin embargo, esta tendrá una serie de accesos dependiendo de si el usuario esta registrado o no.

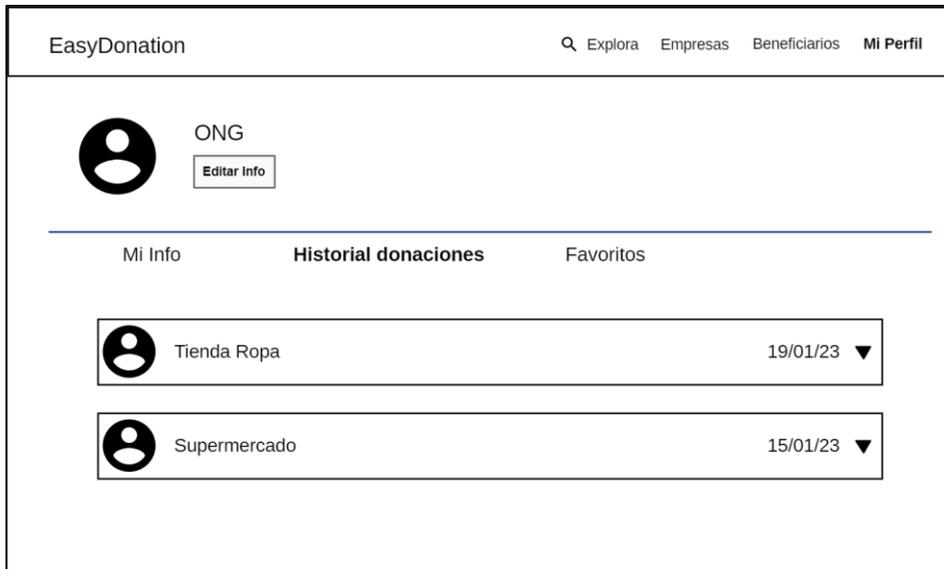


Figura 3.8. Mock-up mi perfil

Respecto a la figura 3.8, en esta se pueden observar los casos de uso de listar el historial de las donaciones, ver las empresas favoritas de la ONG, en el que se mostraría un listado con las empresas seguidas y, por último, la posibilidad de editar el perfil del usuario.



Figura 3.9. Mock-up crear recurso

La figura 3.9 representa la interfaz de crear recurso de una empresa, con los datos que se van a solicitar sus respectivos botones, la interfaz de crear necesidad se ha decidido obviar por su gran parecido a esta.



Figura 3.10. Mock-up ver recursos

Respecto a la figura 3.10, en esta se puede visualizar el caso de uso de listar recursos publicados, en los que aparecen los recursos de todas las empresas, cabe destacar también que este boceto es muy similar al de listar necesidades.



Figura 3.11. Mock-up ver empresas

Y por último, se puede observar la figura 3.11, en la que se ve representado el caso de uso de listar empresas registradas en la plataforma, donde también se puede encontrar el caso de uso de ver el perfil de la empresa.

4. Diseño de la solución

Una vez se ha explicado la fase de análisis, se aborda la fase de diseño de la solución, en la cual se detalla la arquitectura del sistema, permitiendo así tener una visión más técnica y desarrollada de las partes internas de nuestra aplicación.

4.1. Arquitectura del sistema

Respecto a la arquitectura del sistema [3], para esta aplicación se ha utilizado la arquitectura de tres niveles, esta está dividida en los niveles de presentación, negocio y persistencia, como se puede observar en la figura 4.1. Cada una de estas tres capas lógicas son independientes entre sí y cada una de ellas tiene su propia funcionalidad.



Figura 4.1. Arquitectura del sistema (Fuente: www.bitacorasoftware.com)

En primer lugar, se puede observar la capa de presentación, esta es la encargada de interactuar con el usuario final de la aplicación, desde la que se pueden realizar solicitudes de datos a la capa de negocio y estar a la espera de la devolución de los mismos para mostrarlos por pantalla al usuario. En esta capa se debe evitar la inclusión de métodos relacionados con la lógica de negocio y conexiones directas a la base de datos.

En segundo lugar, está la capa de negocio, en esta se encuentra toda la lógica del proyecto desarrollado, desde donde se manipulan los datos solicitados de la capa de presentación y los que la capa de negocio solicita a la capa de persistencia para poder realizar las acciones necesarias.

Y en último lugar se encuentra la capa de persistencia, esta es la responsable del manejo de los datos del sistema, tanto para proporcionarlos a la capa de negocio y usuario como para almacenarlos. Esta capa es la encargada de realizar las operaciones de lectura, escritura o borrado a la base de datos.

En este proyecto, la capa de presentación se encuentra en la parte de front-end. Y las capas de lógica de negocio y persistencia se sitúan en el back-end. Ambas partes se comentarán y desarrollarán en más profundidad en la sección Implementación.

4.2. Diseño detallado

Para poder llevar a cabo el diseño de la base de datos se ha utilizado el siguiente modelo de datos como referencia, como se puede visualizar en la figura 4.2. En este se pueden observar las propiedades y las relaciones entre las entidades de la base de datos.

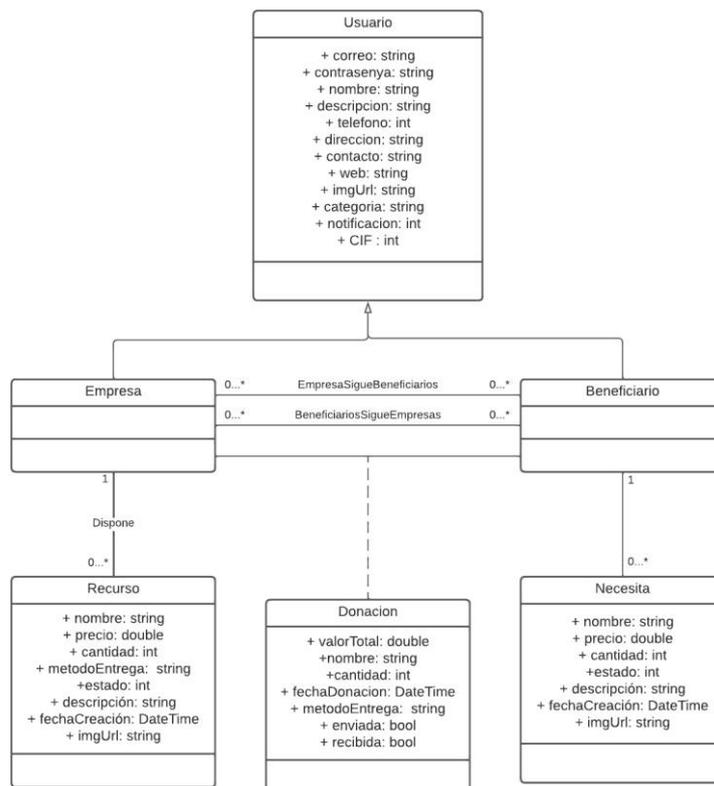


Figura 4.2. Modelo de datos

Este modelo de datos no tiene un grado de dificultad muy elevado ya que este está formado por seis tablas, cada una con sus propias relaciones que las conectan con otras tablas y con sus propios atributos. Si se visualiza la figura 4.2 de arriba abajo, la primera tabla que aparece es la de Usuario, esta tabla es el padre de las tablas Empresa y Beneficiario, por lo que ambas heredarán todos sus atributos.

Por otro lado, las tablas de Empresa y Beneficiario son las representantes de los usuarios de la aplicación, encargadas de almacenar todos los datos de los usuarios y sus relaciones. Cada una de estas tablas tiene una tabla asociada: Recurso en el caso de la tabla Empresa y Necesita en el caso de la tabla Beneficiario. Estas son la representación de lo que publican las Empresas y Beneficiarios en la aplicación, con sus propios atributos para poder guardar la información necesaria.

Además, las tablas de Empresa y Beneficiario tienen dos relaciones muchos a muchos entre sí para poder saber a qué empresas y/o beneficiarios siguen cada una de ellas y así permitir almacenar la información de los seguidos. Asimismo, también existe la tabla Donación conectada entre estas a partir de una relación muchos a muchos, donde se almacenará la información cuando se confirme la solicitud de aceptación de un recurso o una necesidad entre una Empresa y un Beneficiario.

5. Implementación

Una vez se ha explicado detalladamente el diseño de la solución, se presenta el desarrollo de la implementación de EasyDonation. En primer lugar, se encuentran las tecnologías utilizadas para su desarrollo, continuando con la explicación del desarrollo del back-end y del front-end.

5.1. Tecnologías utilizadas

A continuación, se explican tanto las tecnologías como los frameworks⁸ utilizados para llevar a cabo la aplicación. Destacando antes que para poder utilizar las siguientes tecnologías y frameworks se han utilizado los dos siguientes entornos de desarrollo.

5.1.1. Visual Studio Code

Visual Studio Code [12] es un entorno de desarrollo altamente utilizado para el desarrollo de aplicaciones web, debido a la cantidad de extensiones que permite incorporar y su gran versatilidad, además de su gran facilidad de uso y una interfaz muy agradable a la hora de programar. Asimismo, también admite una gran cantidad de lenguajes de programación. Con el conjunto de estas características se decidió que este entorno era el idóneo para el desarrollo del front-end.

5.1.2. Visual Studio Enterprise

Visual Studio Enterprise [13] es otro entorno de desarrollo, este es una versión más completa de Microsoft Visual Studio que la versión gratuita. Con esta versión se añaden una gran cantidad de características, herramientas y opciones adicionales que lo hace un gran entorno de trabajo, tanto para las aplicaciones web, como aplicaciones de escritorio, móviles... Asimismo, permite una serie de funcionalidades bastante interesantes como son la depuración y las pruebas, ambas muy importantes para detectar errores en el código.

⁸ Un framework es una estructura organizada con un conjunto de reglas para el desarrollo software, cuyo objetivo es simplificar el trabajo.

5.1.3. Angular

Angular [11] se trata de uno de los frameworks más populares en el sector del desarrollo de aplicaciones web desarrollada por Google, se centra sobre todo en la parte front-end de las aplicaciones, y se utiliza en Visual Studio Code. Este framework usa JavaScript y TypeScript para construir las aplicaciones, ya que TypeScript se basa principalmente en JavaScript añadiendo ciertas características más avanzadas que le permiten obtener un poco más de seguridad en el código. Por otro lado, respecto a sus características, se pueden destacar la facilidad a la hora de generar código con las plantillas que permiten generar componentes, servicios... además, tiene una arquitectura basada en componentes, lo que favorece la reutilización de código. También se puede destacar la facilidad que proporciona para el enrutamiento.

5.1.4. .Net

.Net [1] es un framework centrado en el desarrollo software, creado por Microsoft. Este proporciona una amplia cantidad de funcionalidades, como es el acceso a bases de datos o la seguridad. Además, este permite que el código sea ejecutado independientemente del sistema en el que se ejecute. Asimismo, proporciona una biblioteca muy amplia para incorporar a las aplicaciones web. Dicho framework acepta una gran cantidad de lenguajes de programación, pero en este caso se decidió utilizar C#.

5.1.5. Bootstrap

Bootstrap [15] es uno de los frameworks más populares en el mundo de las aplicaciones web, ya que proporciona una gran cantidad de estilos y diseños CSS para la aplicación de EasyDonation, haciéndola más vistosa y agradable a la vista para el usuario. Por otro lado, además consigue que la interfaz sea responsive, permitiendo así que se pueda utilizar tanto en ordenador como en móvil.

5.1.6. Entity Framework

Entity Framework [14] es un framework y ORM, es decir un Mapeo de Objeto-Relacional, creado por Microsoft. Con este framework permite interactuar con la base de datos de una forma muy sencilla y rápida, permitiendo así a los usuarios poco experimentados ser capaces de utilizarlo. Gracias a este se puede utilizar código para interactuar con la base de datos en vez de utilizar sentencias SQL.

5.1.7. GitHub

Para poder llevar un control de versiones de la parte front-end y de la parte back-end se ha utilizado la herramienta GitHub [16], esta permite tanto alojar tu código como revisarlo desde la misma web. Gracias a GitHub se puede realizar un seguimiento de las versiones del código de una manera muy sencilla. Esta herramienta es altamente utilizada en proyectos de desarrollo por equipos, facilitando así la colaboración entre estos.

5.1.8. Postman

En cuanto a Postman [5], esta es una herramienta que permite realizar pruebas sobre las APIs que se puedan haber creado en el proyecto, además permite documentarlas. Con Postman se envían peticiones HTTP a la API para validar si los resultados son válidos y ver su funcionamiento.

5.2. Desarrollo del Back-End

Para la implementación del desarrollo del back-end de la aplicación, se ha empleado Visual Studio Enterprise, .NET, EntityFramework y Postman, todas mencionadas en la sección anterior. En el back-end es donde se ubican las capas de lógica de negocio, ya que es donde se realizan las operaciones de manejo de datos, de forma que el usuario no sea consciente de cuál es el tratamiento que se hace con estos, y la capa de persistencia encargada de almacenarlos y/o extraerlos de la base de datos. Además, se han utilizado dos patrones de diseño, el patrón repositorio [21] y el patrón DTO [10].

Para empezar con la explicación de como se ha realizado el desarrollo, se introduce la estructura utilizada en el proyecto. En el caso de EasyDonation, se han utilizado los controladores, estos son los encargados de recibir las peticiones del usuario y es donde se llama a los repositorios para poder manejar los datos solicitados. Asimismo, para poder realizar dicho manejo de datos se crearon unas entidades DTO que serán las que se devuelvan al usuario, haciendo así que nunca se manejen los datos directamente de la base de datos.



5.2.1. Controladores

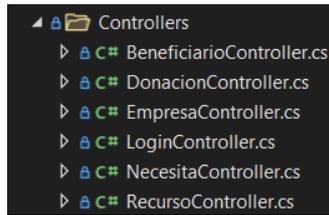


Figura 5.1. Controladores Back-End

En primer lugar, se encuentran los controladores [8], en la figura 5.1 se puede apreciar su estructura en el proyecto, ubicados en la misma carpeta. Antes de empezar a desarrollar para que se utilizan los controladores, se debe comentar que el back-end se conecta con el front-end actuando como un endpoint, es decir, el back actúa como una API que está a la espera de solicitudes del front, en la que el usuario solicita la información que necesite y envía solicitudes HTTP, que son leídas e interpretadas por parte del back-end. Y es en este punto donde aparecen los controladores ya que estos se encargan de recibir las llamadas desde el front y realizar las acciones necesarias llamando a los métodos de las interfaces de los repositorios que utilice.

En los controladores, para permitir que no se accedan a todos los endpoints de la API, se han aplicado unas capas de seguridad a partir de las cuales se limita el acceso a los usuarios dependiendo de su rol [25] y si tienen la capacidad de poder realizar la acción. Así se evitan los posibles problemas con el tratado de datos y su acceso.

```
[Authorize(Roles = "Empresa")]
[HttpDelete("{id}")]
0 referencias | angelperez21, Hace 21 días | 1 autor, 3 cambios
public async Task<IActionResult> Delete(int id)
{
    try
    {
        var recurso = await _recursoRepository.GetRecurso(id);

        var userId = HttpContext.User.FindFirst(ClaimTypes.Name)?.Value;

        if (recurso.IdEmpresa.ToString() != userId)
        {
            return Unauthorized();
        }

        if (recurso == null)
        {
            return NotFound();
        }

        await _recursoRepository.DeleteRecurso(recurso);

        return NoContent();
    }
    catch (Exception ex)
    {
        return BadRequest(ex.Message);
    }
}
```

Figura 5.2. Código RecursoController

Como se puede observar en la figura 5.2, este es un método que actúa como un endpoint de la clase RecursoController, a la espera de una solicitud HttpDelete desde la capa de presentación. Para poder acceder al end-point, junto al cuerpo de la solicitud se debe incluir un header en el que se pasa un token JWT [18] (Json Web Token) cifrado,

donde se esconde la Id del usuario que ha iniciado sesión, su rol y la fecha de validez de dicho token. Una vez se envía toda esta información en el header de la solicitud, se comprobará si dicho usuario tiene el rol que se necesita, en este caso Beneficiario y si el usuario solicitante corresponde con el usuario que se está solicitando de baja, evitando así que cualquier usuario pueda dar de baja a otro.

Gracias al token JWT se puede ampliar un nivel más el proyecto, permitiendo asegurar que las llamadas a la API solo son accesibles para los usuarios que disponen de los permisos oportunos. A continuación, en la figura 5.3 se puede apreciar como es el proceso de creación desde el repositorio de Login, en esta figura se ve como se guarda en el token tanto el Id del usuario, como su email y su rol, además de una fecha de validez de siete días. Y una vez se guarda esa información, se crea el Token y se envía al usuario donde se guardará y posteriormente será utilizado para comprobar los permisos de acceso a los end-points y a otras facetas de la aplicación como las rutas.

```
var tokenHandler = new JwtSecurityTokenHandler();
var key = Encoding.ASCII.GetBytes(secretKey);

var tokenDescriptor = new SecurityTokenDescriptor
{
    Subject = new ClaimsIdentity(new Claim[]
    {
        new Claim(ClaimTypes.Name, user.Id.ToString()),
        new Claim(ClaimTypes.Email, user.Email.ToString()),
        new Claim(ClaimTypes.Role, user.GetType().Name)
    }),
    Expires = DateTime.UtcNow.AddDays(7),
    SigningCredentials = new(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha256Signature)
};

var token = tokenHandler.CreateToken(tokenDescriptor);
```

Figura 5.3. Creación Token JWT

5.2.2. Repositorios

Para poder manipular los datos que suministra la base de datos, se utilizan los repositorios, estos proporcionan una interfaz a la que los controladores se encargan de llamar y acceder a sus métodos para obtener los datos que se han solicitado. En primer lugar, para crear un repositorio se define una interfaz, donde se ubican los métodos que van a ser accesibles desde los controladores y los que se encargarán de interactuar con los datos. En gran parte de los casos, en estas interfaces se definen las operaciones básicas CRUD (Crear, leer, actualizar y eliminar), pero, además, se pueden seguir definiendo más operaciones dependiendo del tratado que se desee aplicar a los datos, como se puede observar en la figura 5.4.

```

4 referencias | angelperez21, Hace 11 días | 1 autor, 6 cambios
public interface IRecursoRepository
{
    3 referencias | angelperez21, Hace 33 días | 1 autor, 1 cambio
    Task<List<Recurso>> GetListRecursosPublicados();
    2 referencias | angelperez21, Hace 33 días | 1 autor, 1 cambio
    Task<List<Recurso>> GetListRecursosEmpresa(int id);
    5 referencias | angelperez21, Hace 36 días | 1 autor, 1 cambio
    Task<Recurso> GetRecurso(int id);
    2 referencias | angelperez21, Hace 36 días | 1 autor, 1 cambio
    Task DeleteRecurso(Recurso recurso);
    2 referencias | angelperez21, Hace 11 días | 1 autor, 1 cambio
    Task UpdateRecurso(RecursoDto recursoDto, int id);
    2 referencias | angelperez21, Hace 36 días | 1 autor, 1 cambio
    Task<Recurso> AddRecurso(Recurso recurso);
    2 referencias | angelperez21, Hace 27 días | 1 autor, 1 cambio
    Task<Recurso> SolicitarRecurso(int idRecurso, int idBeneficiario);
    2 referencias | angelperez21, Hace 24 días | 1 autor, 1 cambio
    Task<Recurso> AceptarRecurso(int idRecurso, int idBeneficiario);
    2 referencias | angelperez21, Hace 24 días | 1 autor, 1 cambio
    Task<List<SolicitanteDto>> GetNotificaciones(RecursoDto recurso);
    2 referencias | angelperez21, Hace 21 días | 1 autor, 1 cambio
    Task<List<Recurso>> GetSolicitudesRecursos(int id);
}
    
```

Figura 5.4. Interfaz Repositorio Recurso

Una vez ya se han definido los métodos de la interfaz, se procede a implementar dichos métodos, donde se ubica la lógica real de la aplicación, responsable de realizar las acciones con la base de datos y su tratamiento, en esta situación para acceder e interactuar con la base de datos se realizan consultas LINQ⁹.

```

3 referencias | angelperez21, Hace 52 días | 1 autor, 2 cambios
public async Task<List<Recurso>> GetListRecursosPublicados()
{
    var recursos = await _context.Recursos.ToListAsync();
    List<Recurso> listRecursos = new List<Recurso>();

    foreach (var rec in recursos)
    {
        if (rec.Estado == 1)
        {
            listRecursos.Add(rec);
        }
    }

    return listRecursos;
}
    
```

Figura 5.5. Método Repositorio Recurso

Como se puede visualizar en la figura 5.5, en esta se realizan las medidas requeridas para la obtención de los datos. Asimismo, en la primera línea se puede visualizar una consulta LINQ a la base de datos de la que se extraen una lista de recursos. Una vez se obtienen los datos necesarios, de estos se extraen los recursos que estén en el estado de publicado y así devolver la lista al controlador que ha realizado la llamada, que a su vez devolverá la lista a la capa de presentación desde donde se ha realizado la llamada a la API.

⁹ LINQ es un conjunto de métodos que se encargan de facilitar la redacción consultas a la base de datos.

Una de las características más importantes de los repositorios es que se encargan de ocultar la lógica de implementación del método y del tratado de datos, permitiendo que los controladores o servicios puedan acceder a los repositorios y así estos se encarguen de manejar los datos, evitando que se hagan llamadas a la base de datos desde los controladores. Estos ofrecen una serie de beneficios que prácticamente hacen indispensable su uso, mejorando la modularidad del código y facilitando el mantenimiento.

Por otro lado, en alguno de los repositorios también se encuentran una serie de métodos que utilizan la librería Mail¹⁰ de .Net, con la que se permite enviar mensajes de correo electrónico utilizando el protocolo SMTP (Simple Mail Transfer Protocol). Gracias a esta se permite enviar correos de una forma muy sencilla desde el back-end de la aplicación, haciendo así que el usuario no tenga ninguna responsabilidad en este aspecto. Esta funcionalidad dota a la aplicación de una mejor comunicación e interacción de los usuarios y el sistema, permitiendo informar al usuario en el caso de que se haya solicitado un recurso que ha publicado, para indicar que su solicitud de donación ha sido aceptada o en el caso de que un usuario desee recuperar su contraseña. En la figura 5.6 se puede ver un ejemplo en el que se utiliza dicha librería, en la que se crea el mensaje, asignándole quien es el que envía dicho correo, cuál es su asunto, a quien va dirigido y su el cuerpo de mensaje. A su vez, también se incluirá el correo de la cuenta y su contraseña para que la librería sea capaz de conectarse con el sistema de mensajería y permita enviar el correo.

```
string fromMail = "easyDonationORG@gmail.com";
string fromPassword = ████████████████████;

MailMessage message = new MailMessage();
message.From = new MailAddress(fromMail);
message.Subject = "Bienvenido a EasyDonation";
message.To.Add(new MailAddress(empresa.Email));
message.Body = "<html><body><h1>Bienvenido a nuestro sitio web</h1><p>Hola "
+ empresa.Nombre +
",</p><p>Gracias por registrarte en nuestro sitio web. Esperamos que disfrutes " +
"de nuestros servicios y te sientas como en casa.</p>" +
"<p>Tus credenciales de inicio de sesión son:</p><ul><li><strong>Email:</strong> "
+ empresa.Email +
"</li><li><strong>Contraseña:</strong> " + empresa.Contrasenya +
"</li></ul><p>Gracias,</p><p>El equipo de EasyDonation</p></body></html>";

message.IsBodyHtml = true;

var smtpClient = new SmtpClient("smtp.gmail.com")
{
    Port = 587,
    Credentials = new NetworkCredential(fromMail, fromPassword),
    EnableSsl = true,
};

smtpClient.Send(message);
```

Figura 5.6. Método envío de correo para empresa registrada

¹⁰ La librería Mail es una funcionalidad que incorpora .Net para facilitar el envío de correos electrónicos al usuario.

5.2.3. Entidades

Para poder manejar los objetos de la base de datos se han creado una serie de entidades que son las encargadas de representarlas, como se puede ver en la figura 5.7, cada una de estas contendrá sus propias propiedades y relaciones con el resto de las entidades. Todas estas ya han sido definidas previamente en el modelo de datos en el apartado de diseño de la solución.

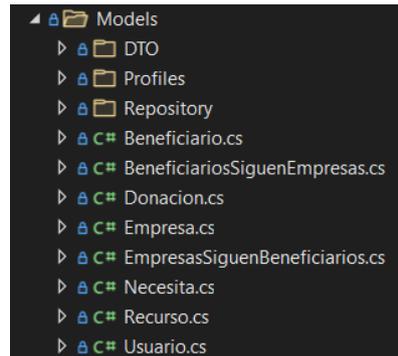


Figura 5.7. Entidades Proyecto

Para que las diferentes capas sean capaces de comunicarse se aprovecha la existencia del patrón DTO, con el que se permite representar la estructura de los datos y poder utilizar la misma de la forma que más beneficie al desarrollador. Con el patrón DTO las entidades contienen las propiedades de la entidad que representa, sin embargo, es importante resaltar que no tienen por qué tener todas las propiedades, sino que también puede tener solo las que sean necesarias en ese momento y, además, podrá disponer de algún método si fuese necesario también. En este proyecto se han utilizado objetos DTO visibles en la figura 5.8.

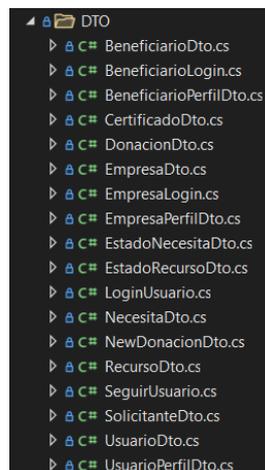


Figura 5.8. Objetos DTO

Por otro lado, para convertir la entidad que se extrae de la base de datos y se quiere devolver a la capa de presentación mediante un objeto DTO se utilizan los Profiles [7], estos se encargan de mapear automáticamente las entidades a los DTO o viceversa, pero para poder hacer esto es necesario crear unos perfiles previamente que se encarguen de enlazar las entidades, como se aprecia en la figura 5.9. En este se aprecia que se ha creado un mapeo bidireccional, tanto para mapear una entidad Recurso a una entidad RecursoDto, como en el sentido opuesto.

```
0 referencias | angelperez21, Hace 36 días | 1 autor, 1 cambio
public RecursoProfile()
{
    CreateMap<Recurso, RecursoDto>();
    CreateMap<RecursoDto, Recurso>();
}
```

Figura 5.9. Mapeo Recurso

Gracias al uso del patrón DTO, y en consecuencia de los Profiles, se puede mejorar el rendimiento y dotar de una mayor flexibilidad a nuestra aplicación, debido a que solo pasamos los objetos con los atributos que sean necesarios en cada momento, reduciendo así la cantidad de datos transferidos entre los componentes y las llamadas.

5.2.4. Base de datos

Por último, se encuentra la base de datos, lugar al que accede la capa de persistencia para extraer los datos para que se puedan utilizar en la capa de negocio. Para su implementación se ha aplicado la técnica conocida como Code First [4], en la que primero se definen todas las entidades del proyecto, con sus propiedades y relaciones. Una vez ya están todas definidas, se procede a realizar la migración [19] de las entidades a la base de datos, donde automáticamente se crean todas las tablas y las relaciones. Las migraciones son una funcionalidad que proporciona Entity Framework para poder exportar las entidades a la base de datos de una forma muy sencilla y prácticamente automática.

En primera instancia, para poder probar el funcionamiento correcto de la base de datos se utiliza en local, para ello se procede a la instalación de MySQL en local y tras esto se inicia la instalación del MySQL Server, en el que estará activa la base de datos con su propio connection string. Este connection string deberá estar en el archivo Program.cs del proyecto para que acceda a la base de datos.

5.3. Desarrollo del front-end

En el otro lado del desarrollo se encuentra el front-end, en este se utiliza el framework de Angular, el cual permite mantener una arquitectura interna basada en componentes, con los que se facilita el trabajo gracias a la reutilización de los mismos y con la gran rapidez que lleva la creación de estos.

Al igual que se ha hecho con el desarrollo del back-end, antes de empezar con la explicación del front-end, se realiza una pequeña introducción sobre la estructura que sigue esta parte. Esta está formada principalmente por los componentes, como ya se ha mencionado en el párrafo anterior, el uso de services que se encargan de realizar las peticiones a la API y el enrutamiento de los componentes.

5.3.1. Enrutamiento

Una de las partes a destacar en este desarrollo, es la importancia que tiene el enrutamiento [20] de los componentes, con esto se permite que se navegue entre los diferentes componentes de la aplicación sin la necesidad de tener que estar cargando toda la aplicación continuamente. Así, simplemente cuando se desee un componente en específico, el sistema de enrutamiento hace que solo se cargue el componente deseado, lo que permite tener una mejor estructura en el proyecto. Cuando se utiliza el enrutamiento, cada componente que quiera ser visto deberá tener asignada una URL, con la que dicho componente será cargado. En la figura 5.10, se puede visualizar cómo cada componente de la aplicación tiene una URL personalizada, y además alguno de ellos contienen una propiedad llamada `canActivate`, encargada de controlar el acceso a los componentes.

```
const routes: Routes = [
  { path: '', redirectTo: 'Inicio', pathMatch: 'full' },
  { path: 'Inicio', component: PantallaInicialComponent },
  { path: 'InicioSesion', component: LoginComponent, canActivate: [VigilanteNoInicio] },
  { path: 'Registro', component: RegistroComponent, canActivate: [VigilanteNoInicio] },
  { path: 'ListaBeneficiarios', component: ListarBeneficiariosComponent },
  { path: 'ListaEmpresas', component: ListarEmpresasComponent },
  { path: 'ListaDonaciones', component: ListarDonacionesComponent, canActivate: [VigilanteGuard] },
  { path: 'ListaNecesidades', component: ListarNecesitaComponent },
  { path: 'ListaRecursos', component: ListarRecursosComponent },
  { path: 'Explorar', component: ExplorarComponent },
  { path: 'Beneficiario/:id', component: VerBeneficiarioComponent },
  { path: 'Empresa/:id', component: VerEmpresaComponent },
  { path: 'Recurso', component: CrearEditarRecursoComponent, canActivate: [VigilanteEmpresaGuard] },
  { path: 'Recurso/:id', component: CrearEditarRecursoComponent, canActivate: [VigilanteEmpresaGuard] },
  { path: 'Necesita', component: CrearEditarNecesitaComponent, canActivate: [VigilanteBeneficiarioGuard] },
  { path: 'Necesita/:id', component: CrearEditarNecesitaComponent, canActivate: [VigilanteBeneficiarioGuard] },
  { path: 'MiPerfil/:id', component: VerPerfilComponent, canActivate: [VigilanteGuard] },
  { path: 'PolíticasDePrivacidad', component: PrivacidadComponent },
  { path: '**', redirectTo: 'Inicio', pathMatch: 'full' }
];
```

Figura 5.10. Enrutamiento de componentes

Respecto a la propiedad de `canActivate`, esta se hace responsable de validar qué usuarios pueden acceder a los componentes a través de su URL, por ejemplo, en

el caso de la creación de un Recurso, solo podría acceder una Empresa a esa URL ya que es la encargada de crearlo y publicarlo, por lo que, si un Beneficiario o un usuario anónimo intentase acceder a la URL, este sería devuelto a la página de Inicio de la web por falta de permisos. Por lo que gracias a estos Guards [17] se genera una capa más de seguridad a parte de las ya mencionadas en el back-end del proyecto.

Todas esas rutas serán cargadas desde el componente `app.component`, que se llama en el archivo `index.html`. En este archivo, se encuentra cómo está organizada la interfaz, y cuáles son los componentes que se van a ir cargando y los que se mantendrán estáticos. En el caso de los últimos se tienen dos el `footer`, es decir, el pie de página de la web, y la `nav-bar`, ya que ambos estarán en todas las páginas de la aplicación, por lo que no tendrán que disponer de una URL para el acceso al componente. Sin embargo, el resto de los componentes serán accedidos desde el router-outlet como se ve en la figura 29, cada vez que se cambie la URL se recargará el router-outlet y accederá a uno de los componentes ya vistos en la figura 5.11.

```
<app-nav-bar></app-nav-bar>
  <router-outlet></router-outlet>
<app-footer></app-footer>
```

Figura 5.11. Estructura interfaz en `app.component`

5.3.2. Componentes

Como ya se ha comentado, Angular basa todo en los componentes, estos son los encargados de construir la interfaz de la aplicación web. La plantilla del componente está formada por varios archivos, una plantilla HTML, estilos CSS y un archivo TypeScript, donde se encontrará la lógica de la interfaz, como se puede apreciar en la figura 5.12, en esta se ve el componente `footer` desplegado con sus archivos y el resto de los componentes de la aplicación sin desplegar, los cuales pueden estar formados por más subcomponentes.

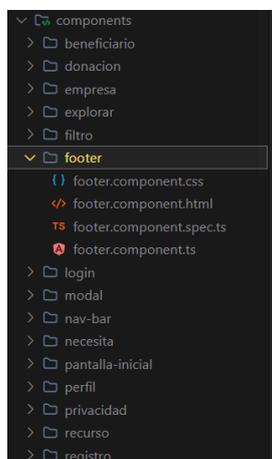


Figura 5.12. Componentes front-end

Los componentes permiten al desarrollador reutilizar código, ya que al ser independientes entre sí se pueden utilizar en distintas partes de la aplicación sin ningún tipo de problema, aumentando la flexibilidad del código, escalabilidad y su mantenibilidad. Además, una de sus características principales es que pueden comunicarse e interactuar entre sí, llegando a intercambiar mensajes, eventos o datos, lo que hace que le sea más sencillo al desarrollador la coordinación y comunicación entre los distintos componentes, logrando que la aplicación sea más dinámica.

Para que los componentes sean capaces de comunicarse entre sí, una de las formas más utilizadas son los EventEmitters [6], estos son esencialmente una clase que proporciona el propio framework de Angular, lo que permite el intercambio de información.

```
if(this.route.includes("ListaBeneficiarios")){  
  
  this._beneficiario.disparador.emit({nombre,ubi, followed, categoriaDisp,orden})
```

Figura 5.13. EventEmitter método emit

```
this._empresaService.disparador.subscribe(data=>{  
  this.aplicarFiltros(data.nombre,data.ubi,data.followed,data.categoriaDisp,data.orden);  
})
```

Figura 5.14. EventEmitter método subscribe

Si se quiere llevar a cabo la comunicación se necesita definir una instancia de la clase EventEmitter, en este proyecto se definen en los servicios, que serán explicados más adelante, para ser instanciados y poder usar el método emit(), que contendrá los datos asociados, como se puede visualizar en la figura 5.13. En esta se muestra cómo se utiliza el método emit de la instancia con el servicio Beneficiario, consiguiendo que se envíe el nombre, la ubicación, un bool indicando si los sigue o si no, la categoría y el orden. Y con todo esto se transmite al otro componente donde tendrá también instanciada la clase con un método subscribe, que estará a la espera de la información, figura 5.14.

5.3.3. Servicios

Y para terminar la sección del desarrollo, pasamos al apartado de los servicios. Principalmente, los servicios son clases que se encargan de mantener la lógica con la que se interactúa con la API, es decir, son las clases que se encargan de realizar las solicitudes HTTP y recibir su respuesta.

```

private myAppUrl = "https://easydonation.azurewebsites.net/";
private myApiUrl = 'api/Recurso'

constructor(private http: HttpClient) { }

getRecurso(id: number, httpOptions: any): Observable<any>{
  return this.http.get(this.myAppUrl + this.myApiUrl + '/' + `${id}`, httpOptions);
}

postRecurso(recurso: any, httpOptions: any): Observable<any>{
  return this.http.post(this.myAppUrl + this.myApiUrl, recurso, httpOptions);
}

deleteRecurso(id: number, httpOptions: any): Observable<any>{
  return this.http.delete(this.myAppUrl + this.myApiUrl + '/' + `${id}`, httpOptions);
}

updateRecurso(id: number, recurso: any, httpOptions: any): Observable<any>{
  return this.http.put(this.myAppUrl + this.myApiUrl + '/' + `${id}`, recurso, httpOptions);
}

```

Figura 5.15. Service Recurso

Los métodos que se encuentran definidos en los service para realizar llamadas a la API utilizan el módulo HttpClient, con el que se les permite enviar las peticiones de tipo Get, Post, Put y Delete, es decir los llamados Verbos de HTTP. Como se observa en la figura 5.15, con cada llamada a la API se añade en todos los casos el parámetro httpOptions, esto se debe a que ese parámetro es el que contiene el elemento header, donde está ubicado el token que posteriormente en el end-point se comprobará si cumple los permisos necesarios. Asimismo, en los casos que se modifique un objeto o se publique uno nuevo, este se pasará como parámetro.

Los servicios se pueden inyectar como dependencias en los componentes que los necesiten, facilitando así la reutilización de código y la gestión estas. Al incorporar el servicio en un componente, se puede acceder a todos los métodos que contenga ese service, desde donde se realizan las llamadas a la API y a las instancias de los EventEmitter en el caso de que tenga. Respecto a la inyección de dependencias, esta se realiza desde el constructor del componente en el que se desea utilizar el servicio, como se ve en la figura 5.16, donde se pasa como parámetro seis servicios y una vez hecho esto ya se puede acceder a cualquier método del service llamando a dicha variable.

```

constructor(private _donacion: DonacionService,
  private _empresa: EmpresaService,
  private _beneficiario: BeneficiarioService,
  private _necesita: NecesitaService,
  private _recurso: RecursoService,
  private toastr: ToastrService) {
}

```

Figura 5.16. Constructor componente Pantalla Inicio

6. Despliegue

Finalizado el capítulo de la implementación de la aplicación web de este proyecto, se pasa a una parte fundamental para poder verla en funcionamiento, el despliegue. Antes de iniciar con la explicación del despliegue de la aplicación se realizó un análisis de las opciones que había en el mercado para hacer esto mismo. Principalmente la duda se centraba en Amazon Web Service¹¹ y Azure¹², ya que, estos dos son los mayores proveedores de servicios del mercado. Cabe destacar que ambos proveedores ofrecen una gran cantidad de servicios por lo que era posible lanzar la parte front-end como el back-end de la aplicación web en ambos proveedores.

Finalmente se escogió la opción del proveedor Azure por una serie de motivos. Empezando por la facilidad de uso de su interfaz y su sencillez a la hora de comprender los menús y como interactuar con estos, como se puede observar en la figura 6.1, haciéndola mucho más intuitiva que la interfaz de la competencia. Todo esto teniendo en cuenta que el encargado de utilizar el proveedor es un usuario inexperto en este ámbito y en ambos proveedores. Además, al elegir la opción de Azure, como esta es propiedad de Microsoft, ofrece una mejor integración con las tecnologías que se han utilizado en el proyecto, ya que Visual Studio Code y Visual Studio Enterprise también son propiedades de Microsoft, haciendo que se cree un mejor ecosistema para ser desplegado. Gracias a esto, desplegar el proyecto en Azure se hace mucho más cómodo, con una pequeña configuración de los servicios de 15 minutos, tanto para el front-end, como el back-end se tiene desplegado el proyecto.

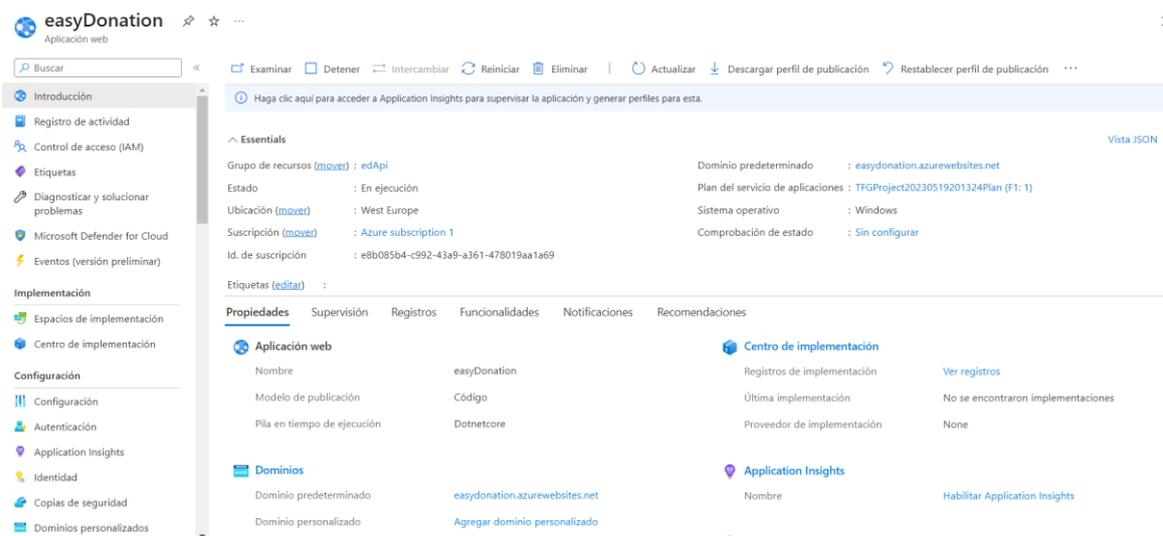


Figura 6.1. Web proveedor Azure

¹¹ Amazon Web Service <https://aws.amazon.com/es>

¹² Azure <https://azure.microsoft.com/es-es/>

Para desplegar tanto la parte front-end, como la parte back-end de la aplicación se han utilizado varios tipos de servicios, estos pueden ser visualizados en la figura 6.2.



Figura 6.2. Servicios utilizados de Azure

En primer lugar, en la figura 6.2, se pueden apreciar dos recursos que proporcionan un mismo tipo de servicio llamado App Service, `easyDonation` y `easyDonationFE`, representando al back-end [24] y al front-end [23] respectivamente, estos son los encargados de mantener activa la web y que sea accesible para todo el mundo. Con una simple configuración desde el Visual Studio Code y el Visual Studio Enterprise, en el que se crea el servicio y se asigna la tarifa gratuita, ya estarían creados los servicios y la aplicación prácticamente desplegada.

Poniendo énfasis en el servicio de `easyDonationFE`, en este se encuentra el código correspondiente a la interfaz y su lógica con las correspondientes llamadas a la API, encargadas de proporcionar la información necesaria al usuario. Y por el otro lado, en `easyDonation` se mantiene activo el servidor que está a la espera de recibir las peticiones que realice el usuario y devolver la información solicitada.

De la misma forma se encuentran los servicios necesarios con el fin de hacer que la base de datos esté en funcionamiento y accesible por el back-end para el tratado de datos. Para ello sí que fue necesario algo más de tiempo. Para realizar el despliegue de la base de datos se necesita un servidor SQL en el que se mantenga activa la base de datos para ser accedida y un plan para establecer el presupuesto, ya que Azure es un proveedor de servicios que tiene servicio basado en pagar únicamente por lo que se utiliza. En este caso se escogió el plan gratuito, con el que se puede llegar a obtener un peor rendimiento que un plan de pago, pero no se dispone de un presupuesto para poder adquirirlo. Y, por último, lo más importante, la base de datos SQL donde se ubican las tablas representativas de las entidades de nuestra aplicación.

Desde el servicio de la base de datos SQL [9] se permite realizar múltiples funciones con ella, ya sea administrar la base de datos mediante copias de seguridad, restauraciones o monitorizar el rendimiento de esta, lo que permite tener una gran fuente de información acerca del estado actual de la base de datos. Asimismo, también permite realizar análisis sobre los datos almacenados y extraer información de estos.

Una vez se ha explicado cómo se ha realizado el despliegue de nuestra aplicación web solo falta acceder a ella y poder darle la funcionalidad que estaba establecida en los objetivos de este proyecto. En la figura 6.3 se puede apreciar la web ya desplegada y accesible desde internet.

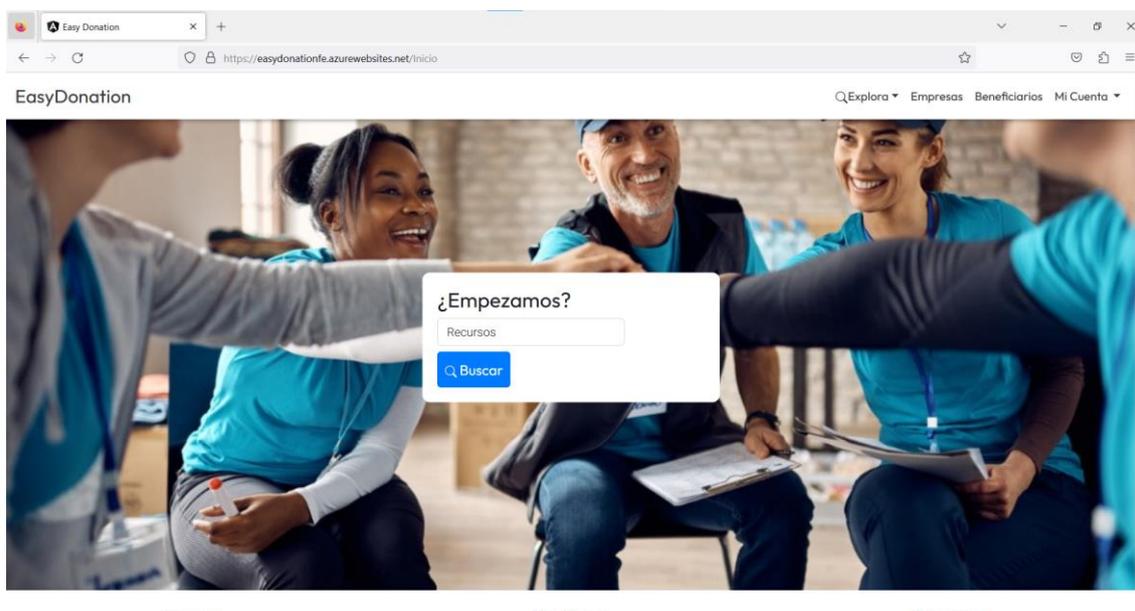


Figura 6.3. Web EasyDonation

6.1. Problemas del despliegue

El despliegue de la aplicación tiene un inconveniente y es que, para poder validar que tanto las Empresas como las ONGs sean reales, a nivel local se utilizaban dos scripts, ambos incorporados en el Anexo B, que permitían verificar la existencia de dicha entidad. Dichos scripts eran diferentes, pero al final se reducían a lo mismo, comprobar que el CIF que se introducía perteneciese a una entidad registrada en algún sistema que se encargue de almacenar información sobre Empresas o ONGs. Para ello en el caso de las Empresas se utilizaba la plataforma de Registro mercantil¹³, y en el caso de

¹³ El Registro mercantil es una plataforma en la que un usuario puede acceder y consultar los datos de la empresa buscada, ya sea el estado en el que esta o quiénes son sus representantes. <https://sede.registradores.org/site/home>

las ONGs se utilizaba Dateas¹⁴ ya que en el Registro Mercantil solo se podrían encontrar Empresas.

Una vez explicada la existencia del script se pasa al problema real, el cual surge cuando se hace el despliegue del back-end en Azure, ya que estos scripts funcionan con la herramienta Selenium¹⁵, herramienta que permitía su ejecución con el navegador que se encontrase en el dispositivo local. Por lo que al usar dicha herramienta mientras la aplicación está desplegada en el servidor de Azure, cuando se intenta ejecutar el script no se llega a desplegar por no disponer de ningún navegador en el servidor, impidiendo así que se pueda comprobar la validez real del CIF, dejando la validación final en una simple comprobación numérica.

Por otro lado, en este momento a la aplicación no se le puede asignar un dominio personalizado que le permita ser más fácil de identificar a los usuarios, puesto que requiere un plan compartido con un coste que no se puede asumir ya que no se tiene ningún presupuesto para poder mantener la aplicación. De la misma forma, con los certificados sucede algo similar, en este caso se intentó incorporar un certificado SSL a la web para establecer unas conexiones más seguras entre el usuario y el servidor, pero para ello se requiere también modificar el plan de Azure teniendo un coste aproximado de ocho céntimos la hora.

¹⁴ Dateas es una plataforma que proporciona información de cualquier entidad registrada, en este caso solo se utilizaba para las ONGs. <https://www.dateas.com/es>

¹⁵ Selenium es una herramienta que permite la automatización de pruebas, pero además permite obtener información de páginas web scrappeando estas.

7. Pruebas

Una vez ya se ha terminado la aplicación y está disponible para su uso se pasa a las pruebas, encargadas de validar que cumple el funcionamiento deseado. Las pruebas son una parte muy importante, ya que gracias a estas se permite evaluar y validar la aplicación, permitiendo identificar posibles errores. Asimismo, gracias a las pruebas también se permite mejorar la calidad del software, llegando a validar los requisitos y reducir los riesgos.

Para realizar la validación de la aplicación se realizan una serie de pruebas que van a ir determinando el estado de la aplicación en la sección en la que se realice la prueba, como puede ser a nivel de interfaz, funcionalidad o de rendimiento.

7.1. Pruebas de compatibilidad

En este tipo de pruebas se testea si la aplicación web es válida tanto en distintos navegadores como en distintos dispositivos móviles. En primer lugar, se prueba con el navegador de Mozilla Firefox, como se observa en la figura 6.3 ubicada en la sección de despliegue, donde se muestra la interfaz correctamente, al igual que sucede en la figura 7.1 donde se utiliza el navegador Google Chrome.

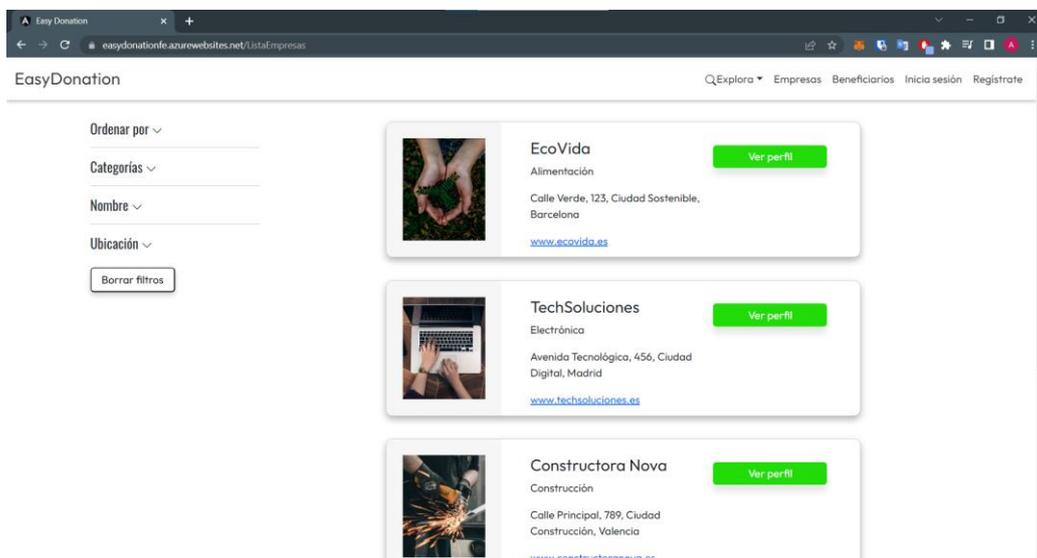


Figura 7.1. EasyDonation en Google Chrome

Por otro lado, se encuentran las pruebas en los dispositivos móviles, este caso se utilizan dos dispositivos móviles diferentes. La figura 7.2 está formada por cuatro imágenes, las dos primeras pertenecientes a un iPhone 12 y las dos últimas a un Realme 7 pro, con estas muestras se puede observar como en los dispositivos iPhone

se va a obtener una correcta visualización de la interfaz, sin embargo, para los dispositivos Android, en las dos últimas imágenes, parece que se ven un poco distorsionadas y se sobrepone el texto en el nombre del perfil. A pesar de estos fallos el usuario puede seguir dándole una correcta funcionalidad a la aplicación web.

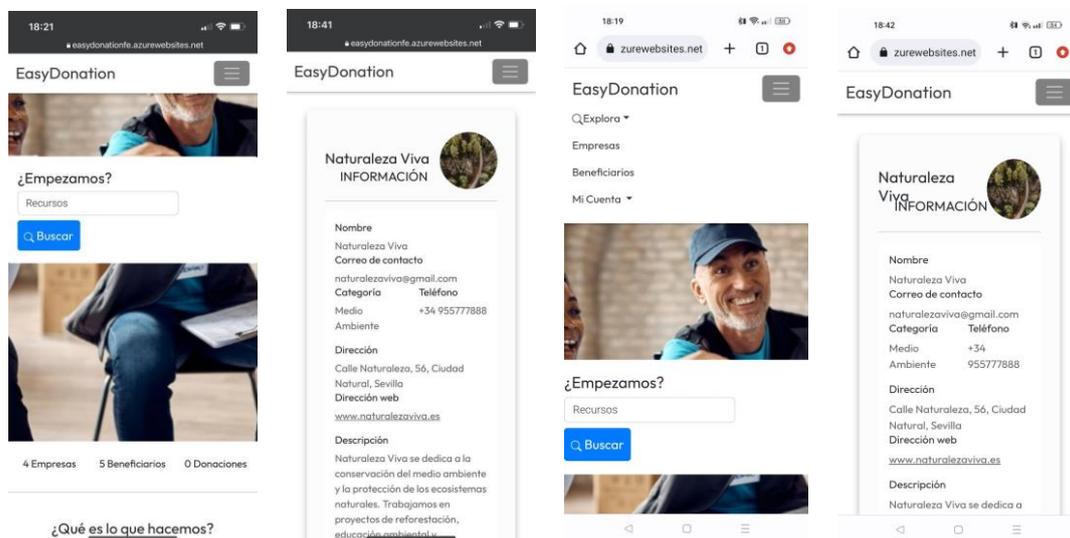


Figura 7.2. Web en iPhone 12 y Realme 7 Pro

7.2. Pruebas de rendimiento

Para poder realizar las pruebas de rendimiento se han utilizado dos páginas web que permiten realizar una serie de pruebas para determinar el estado de la web. En este caso las webs escogidas han sido, PageSpeed¹⁶ con la que se permite ver el rendimiento tanto en un dispositivo móvil como en un ordenador y Gtmetrix¹⁷ con la que se obtienen unos valores que hacen referencia a las métricas de calidad del software de esta página, a partir de los que se obtiene una valoración de la web. Se han escogido dos páginas web para tener más referencias, así como más datos a comparar, para no solo aceptar lo que dice una de estas.

En primer lugar, se encuentra la herramienta de análisis del rendimiento de aplicaciones web, PageSpeed, creada por Google. Con esta se pretende informar a los desarrolladores de las aplicaciones sobre el estado de su aplicación proporcionando una serie de puntuaciones sobre el rendimiento, incluyendo, además, posibles mejoras para optimizar el código en los sitios que se tenga una peor valoración.

Si se realiza un enfoque en la primera imagen de la figura 7.3, se puede visualizar el resultado de la ejecución del testeo. En este se obtienen cuatro métricas, rendimiento,

¹⁶ PageSpeed es una herramienta online que permite dar unos resultados sobre el rendimiento de las webs y opciones de mejora para una optimización del rendimiento. <https://pagespeed.web.dev>

¹⁷ Gtmetrix es otra herramienta online que muestra los resultados obtenidos después de realizar ciertas pruebas sobre las webs. <https://gtmetrix.com/>

accesibilidad, prácticas recomendadas y SEO. La primera imagen pertenece al rendimiento de un ordenador, como resultados se obtiene un rendimiento de un 86 sobre 100, además, en la parte inferior se pueden ver también más métricas específicas donde cinco de las seis métricas tienen un valor óptimo. Sin embargo, la métrica que se basa en el renderizado de los elementos indica que se tarda más de lo que se debería, incluyendo, además, información sobre cómo se podrían guardar las imágenes en un formato específico para las webs, evitando así que los archivos sean muy pesados y reducir la puntuación negativa de esta métrica. Por lo que se refiera a la accesibilidad y prácticas adecuadas se han obtenido unos valores extraordinarios.

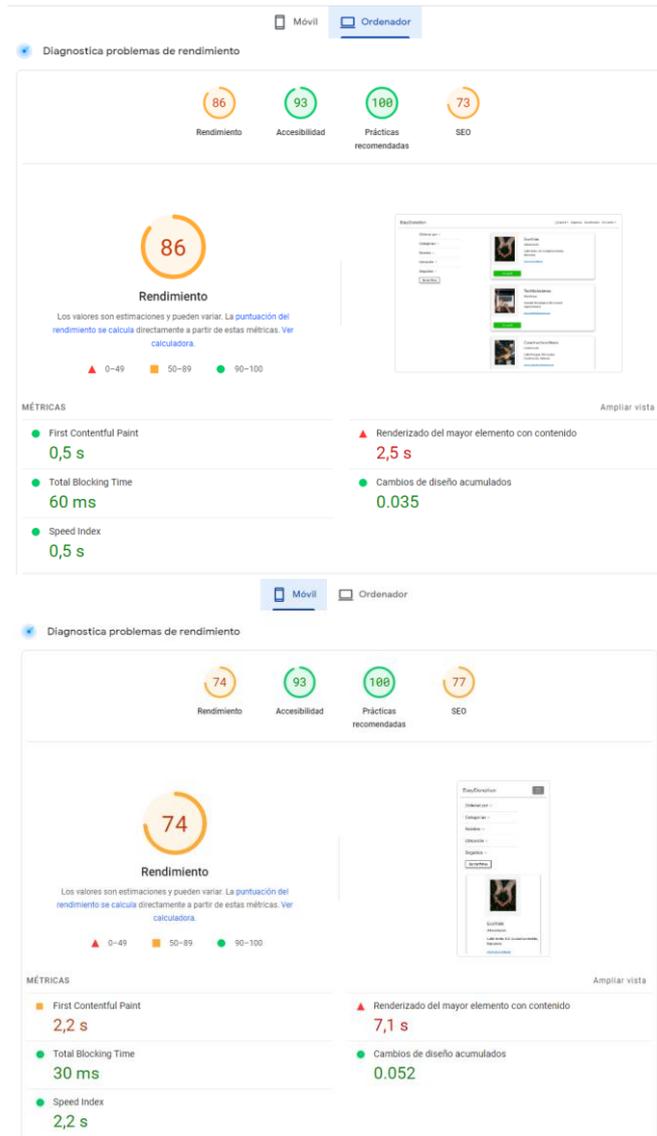


Figura 7.3. Rendimiento Móvil y Ordenador PageSpeed

Por otra parte, en la segunda imagen de la figura 7.3 se aprecia el resultado de los tests para los dispositivos móviles. En cuyo caso la aplicación sale con una valoración negativa, más de lo que se desearía, siendo de un 74 sobre 100. Las métricas que aparecen en la parte inferior de la imagen hacen indicar los motivos por los que se ha obtenido este resultado, siendo dos de las seis negativas.

En segundo lugar, se ha utilizado la herramienta Gtmetrix, esta plataforma proporciona unas métricas bastante similares a PageSpeed para así poder tener dos referencias sobre el rendimiento de la aplicación. Una vez la herramienta Gtmetrix realiza las pruebas necesarias con la aplicación devuelve los resultados mostrados en la figura 7.4.

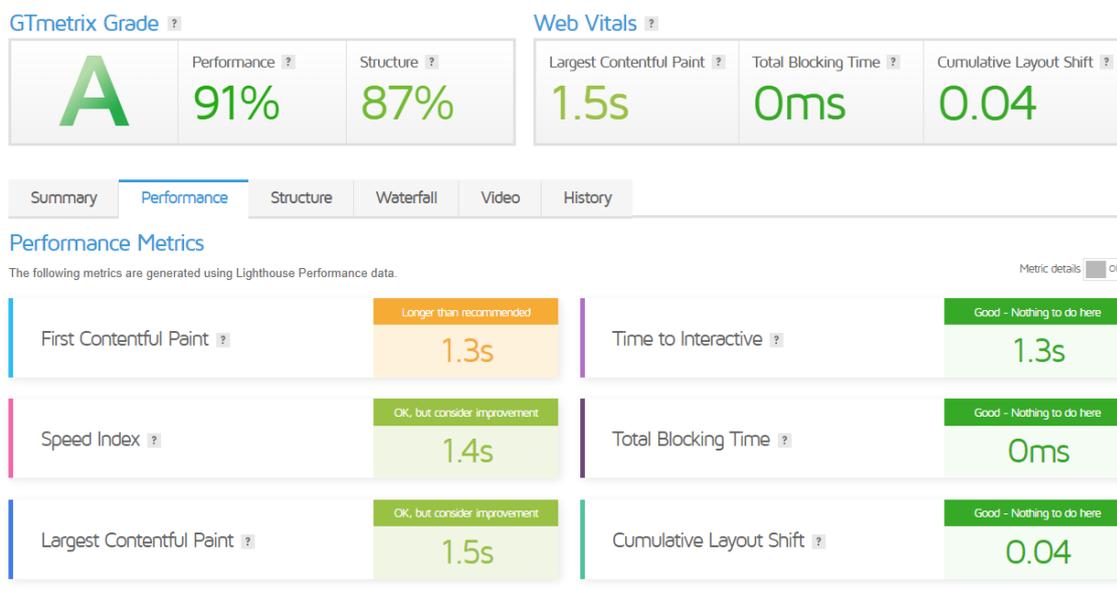


Figura 7.4. Rendimiento Gtmetrix

Con los resultados que proporciona esta herramienta se puede determinar que el rendimiento de EasyDonation es bueno en líneas generales, obteniendo una A como puntuación general, y un 91% y 87% en el rendimiento y estructura de la aplicación respectivamente. Asimismo, las métricas que proporciona del rendimiento también son muy positivas, sin embargo, cabe destacar el tiempo que tarda en cargar los primeros elementos ya que es algo a tener en cuenta para mejorar.

Una vez ya se han comentado las dos herramientas utilizadas para las pruebas de rendimiento y sus respectivos resultados, queda claro que en ambas plataformas las puntuaciones que se obtienen son bastante positivas y no difieren mucho entre las dos herramientas, por lo que logramos obtener un resultado más preciso, claro y conciso. Sin embargo, sí que hay que destacar los valores obtenidos para mostrar los elementos por primera vez, puesto que tanto para la métrica del elemento más pesado y del dispositivo móvil no han sido resultados tan positivos como se desearía, se podría asignar esto como unas posibles mejoras de cara al futuro trabajo que se desarrolle una vez se finalice este trabajo.

Cabe destacar que los resultados de estas herramientas se obtienen mientras la aplicación web se mantiene en unos servidores gratuitos, los cuales no disponen de tantos recursos como podrían disponer si se estuviese pagando por el uso de un servidor.



7.3. Pruebas funcionales

Otra de las pruebas utilizadas son las pruebas funcionales, estas pruebas son una de las técnicas más utilizadas en la ingeniería del software. Con estas se evalúa cómo funciona y comporta el sistema cuando el usuario interactúa con este, con el objetivo de cumplir siempre los requisitos que se han especificado.

Para llevar a cabo estas pruebas se siguen los pasos redactados en los casos de uso, donde especifica que es lo que hace el usuario para llevar adelante cada requisito. A lo largo de este proyecto cada vez que se finalizaba una tarea que involucraba a los casos de uso se realizaba la prueba, en el caso de resultar errónea, se solucionaba y una vez se arreglaba se volvía a realizar la prueba para comprobar el si ya tenía un funcionamiento correcto. Evitando así encontrar los errores en un futuro, lo que conllevaría un aumento del coste y tiempo puesto que cuantos más avances se realizan en el proyecto es más complejo realizar cambios en este.

Además de realizar esas pruebas durante la fase de desarrollo, una vez se terminó la implementación y el despliegue se volvieron a realizar todas las pruebas siguiendo los pasos de los casos de uso para comprobar el correcto funcionamiento de la aplicación ya terminada.

8. Conclusiones y trabajos futuros

En este capítulo se realiza un repaso de lo que ha sido el proyecto, desde sus objetivos hasta la finalización de este, obteniendo unas conclusiones sobre el trabajo realizado, las previsiones sobre el posible trabajo futuro que le depararía a este proyecto y para finalizar la relación del mismo con los estudios cursados.

8.1. Conclusiones

Una vez ya se ha terminado el proyecto en su totalidad, es momento de realizar una conclusión sobre lo que ha sido y lo que me ha aportado como experiencia personal. Recapitulando los objetivos planeados, uno de los principales era permitir que las Empresas y las ONGs pudiesen conectarse y colaborar entre ellas para facilitarles las tareas a las ONGs pequeñas que son las que más problemas tienen en este aspecto, además, como consecuencia de estos se evitarían así también que en muchos casos las Empresas se deshiciesen de productos sin poder darles una utilidad. Otro de los objetivos se centraba en la creación de una aplicación fácil de usar e intuitiva para que cualquier persona que este encargada de manejar la cuenta la aplicación sepa que es lo que debe hacer, facilitándole las tareas en lo máximo de lo posible.

Estos objetivos se han cumplido totalmente, pero para ello se ha requerido de un trabajo y una constancia muy alta por la falta de tiempo para poder terminarlo, ya que no había trabajado nunca en ningún proyecto de este estilo. Esto implica que además del trabajo realizado también se ha tenido que aprender a utilizar algunas de las tecnologías mencionadas en la implementación del proyecto, puesto que eran completamente nuevas para mí.

En resumen, gracias a esto, he podido descubrir nuevas herramientas y tecnologías para poder realizar trabajos de una forma más rápida y sencilla de lo que sabía antes de empezar este proyecto. Este ha sido un aprendizaje constante desde el principio hasta el final, gestionando las tareas, su avance y su implicación en el proyecto, permitiéndome mejorar en la gestión del tiempo y ser lo más eficiente posible.

8.2. Trabajos futuros

Mientras se iba desarrollando el proyecto se han ido identificando posibles mejoras para un futuro posterior a la realización de este trabajo académico. Además de estos aspectos o funcionalidades para un futuro también se tienen en cuenta una serie

de requisitos que no se incorporaron a la aplicación ya que debido a la falta de tiempo era imposible llevarlos a cabo. Entre estos trabajos a futuro se destacan:

- Sistema de pago de terceros para mantener la web.
- Desarrollo de una aplicación móvil.
- Implementación de los scripts para verificar el CIF de los usuarios.
- Funciones para publicar las necesidades y/o recursos publicados por las redes sociales.
- Sistema de mensajería para una mejor colaboración entre usuarios.
- Sistema de almacén virtual en el que las empresas mantengan sus recursos sin necesidad de donarlo a una ONG directamente.
- Sistema de soporte para usuarios por si hubiese problemas.
- Agregar una sección de foros para publicitar eventos de ONGs y poder realizar donaciones más masivas.
- Sección de estadísticas sobre los recursos donados para Empresas.
- Mejora del rendimiento modificando el tamaño de las imágenes.
- Mejora del rendimiento para la aplicación web móvil y su interfaz responsive.
- Implementar un nuevo tipo de usuario donante, que pueda realizar donaciones más pequeñas a las ONGs sin necesidad de ser una Empresa.
- Incorporar las donaciones monetarias.

8.3. Relación con los estudios

A lo largo del desarrollo de este trabajo se han aplicado una serie de habilidades y los conocimientos adquiridos a lo largo de los años de estudio en el grado de Ingeniería Informática, con una especialización en Ingeniería del Software donde se encuentran las asignaturas que tienen un mayor peso en este proyecto. Acto seguido, se citan las asignaturas de las que se han extraído dichos conocimientos:

- Proceso del software (PSW) y Proyecto de Ingeniería de Software (PIN): Estas dos asignaturas son progresivas empezando con PSW y finalizando con PIN, gracias a estas se han aprovechado los

conocimientos sobre las metodologías de desarrollo ágil y como desarrollar un proyecto desde su inicio a su fin.

- Ingeniería del software (ISW): Gracias a la ingeniería del software se han podido desarrollar los diagramas UML ya que es en esta asignatura donde se introducen.
- Desarrollo Dirigido por Modelos (DSM) y Análisis y Especificación de Requisitos (AER): AER es la continuación de lo visto en la asignatura de DSM de las que se ha podido aplicar los conocimientos sobre la especificación de requisitos y su representación.
- Diseño del software (DDS): Respecto al diseño del software se han aprovechado algunas técnicas como los patrones de diseño para mejorar el código y las buenas prácticas de programación.
- Calidad del Software (CSO): De esta asignatura se utilizaron los conocimientos para poder realizar pruebas sobre la calidad del software desarrollado y su rendimiento.
- Mantenimiento del Software (MES): Se han aplicado los conocimientos adquiridos sobre el uso de Git para el control del proyecto.
- Gestión de proyectos (GPR): En cuanto a la asignatura de gestión de proyectos se ha aprovechado lo aprendido con el programa Trello para poder organizar las tareas a lo largo del proyecto.

Además de las asignaturas también se han demostrado una serie de competencias transversales, aprendidas a lo largo del grado de Ingeniería Informática:

- Pensamiento crítico.
- Planificación y gestión del tiempo.
- Diseño y proyecto.
- Innovación, creatividad y emprendimiento.
- Análisis y resolución de problemas.

9. Referencias

- [1] «.NET documentation,» [En línea]. Available: <<https://learn.microsoft.com/en-us/dotnet/>>. [Último acceso: 03 04 2023].
- [2] «¿Qué es un tablero Kanban y cómo utilizarlo? Explicación básica,» [En línea]. Available: <<https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban>>. [Último acceso: 10 04 2023].
- [3] «Arquitectura en Capas,» [En línea]. Available: <<https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/capas>>. [Último acceso: 01 04 2023].
- [4] «Code First Entity Framework,» [En línea]. Available: <<https://learn.microsoft.com/es-es/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application>>. [Último acceso: 07 04 2023].
- [5] WESTERVELD, Dave, 2021. *API Testing and Development with Postman: A practical guide to creating, testing, and managing APIs for automated software testing*. Packt Publishing Ltd. ISBN 9781800565739. Google-Books-ID: 93QqEAAAQBAJ [Último acceso: 22 04 2023].
- [6] «Component Events in Angular with EventEmitter,» [En línea]. Available: <<https://www.topcoder.com/thrive/articles/component-events-in-angular-with-eventemitter-and-output>>. [Último acceso: 05 07 2023].
- [7] «Configuration — AutoMapper documentation,» [En línea]. Available: <<https://docs.automapper.org/en/stable/Configuration.html>>. [Último acceso: 15 04 2023].
- [8] «Controllers ASP .NET,» [En línea]. Available: <<https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/controllers-and-routing/aspnet-mvc-controllers-overview-cs>>. [Último acceso: 10 04 2023].
- [9] «Create a single database - Azure SQL Database,» [En línea]. Available: <<https://learn.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart>>. [Último acceso: 19 05 2023].
- [10] «Data transfer object(Dto),» [En línea]. Available: <<https://reactiveprogramming.io/blog/es/patrones-arquitectonicos/dto>>. [Último acceso: 15 04 2023].
- [11] PUCIARELLI, Luciano, 2020. *Angular: TypeScript – Arquitectura – Instalación – Directivas y Bindings – Forms – Ruteo y más*. RedUsers. ISBN 9789874958273. Google-Books-ID: 6aHTDwAAQBAJ

[Último acceso: 03 04 2023].

- [12] «Documentación Visual Studio Code,» [En línea]. Available: <<https://code.visualstudio.com/>>. [Último acceso: 03 04 2023].
- [13] «Documentación Visual Studio Enterprise,» [En línea]. Available: <<https://visualstudio.microsoft.com/es/vs/>>. [Último acceso: 03 04 2023].
- [14] «Entity Framework documentation hub.,» [En línea]. Available: <<https://learn.microsoft.com/en-us/ef/>>. [Último acceso: 05 04 2023].
- [15] «Getting started · Bootstrap,» [En línea]. Available: <<https://getbootstrap.com/docs/3.4/getting-started/>>. [Último acceso: 03 04 2023].
- [16] «Github,» [En línea]. Available: <<https://github.com/>>. [Último acceso: 10 06 2023].
- [17] «Guards en Angular, ¿Cómo funcionan?,» [En línea]. Available: <<https://binarycoffee.dev/post/guards-en-angular-como-funcionan>>. [Último acceso: 24 05 2023].
- [18] «JWT Token,» [En línea]. Available: <<https://jwt.io/introduction>>. [Último acceso: 29 04 2023].
- [19] «Migraciones Entity Framework,» [En línea]. Available: <<https://learn.microsoft.com/es-es/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/migrations-and-deployment-with-the-entity-framework-in-an-asp-net-mvc-application>>. [Último acceso: 09 04 2023].
- [20] «Páginas y rutas Angular SPA. Academia Binaria.,» [En línea]. Available: <<https://academia-binaria.com/paginas-y-rutas-angular-spa/>>. [Último acceso: 24 04 2023].
- [21] «Patrón repositorio (Repository pattern) y unidad de trabajo (Unit of work) en asp. Net,» [En línea]. Available: <<https://dev.to/ebarrioscode/patron-repositorio-repository-pattern-y-unidad-de-trabajo-unit-of-work-en-asp-net-core-webapi-3-0-5goj>>. [Último acceso: 15 04 2023].
- [22] «Pencil,» [En línea]. Available: <<https://pencil.evolus.vn/>>. [Último acceso: 01 04 2023].
- [23] «Quickstart: Create a Node.js web app - Azure App Service.,» [En línea]. Available: <<https://learn.microsoft.com/en-us/azure/app-service/quickstart-nodejs>>. [Último acceso: 18 05 2023].
- [24] «Quickstart: Deploy an ASP.NET web app - Azure App Service,» [En línea]. Available: <<https://learn.microsoft.com/en-us/azure/app-service/quickstart-dotnetcore>>. [Último acceso: 19 05 2023].



- [25] «Role-based authorization in asp. Net core.,» [En línea]. Available: <
<https://learn.microsoft.com/en-us/aspnet/core/security/authorization/roles?view=aspnetcore-7.0>>. [Último acceso: 20 04 23].

Anexo A

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Tabla A.1. Relación ODS

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.	X			

En EasyDonation al estar centrada en facilitar las conexiones, para establecer una comunicación y colaboración más directa, la mayoría de ODS establecidos en la tabla se relacionan con el proyecto en mayor o menor medida, en este caso este grado de relación se ha determinado dependiendo de si ese ODS era propio del proyecto o dependiendo de las ONGs que se creen que van a ser las que más utilicen la plataforma.

Por un lado, se encuentran los ODS que tienen un mayor grado de relación con el proyecto, como es el caso de Alianzas para lograr objetivos y producción y consumo responsables, debido a que nuestro proyecto se basa en conectar a ONGs y Empresas para que las ONGs puedan llegar a cumplir sus objetivos con los recursos necesarios, ya que, sin estos, les es muy difícil, permitiendo así la colaboración y construcción de alianzas entre estos. Y en cuanto a la producción y consumo responsable, se puede relacionar también con el ODS anterior ya que la mayoría de los recursos que las Empresas publican en la plataforma suelen ser aquellos que no saben qué hacer con ellos o como deshacerse de estos, así que, si son capaces de realizar donaciones con dichos recursos, se fomenta la producción y consumo responsable, evitando así que se tiren y se cree un malgasto innecesario, permitiendo su uso a las ONGs.

Por otro lado, se encuentran los ODS de grado medio de relación con el proyecto, en este nivel se encuentran los ciudades y comunidades sostenibles y reducción de las desigualdades. En primer lugar, respecto al primer ODS puede ser un poco la consecuencia de producción y consumo sostenibles ya que esto conlleva que en la sociedad en la que se vive pueda ser más sostenible al dar una segunda oportunidad a los recursos que ofrecen las empresas. Y respecto a la reducción de las desigualdades, este destaca debido a que con el proyecto se busca que las ONGs más pequeñas puedan tener más visibilidad, permitiéndoles así que sea más fácil que las empresas les puedan ayudar y no solo se done a las grandes ONGs.

Anexo B

```
EdgeOptions options = new EdgeOptions();
options.AddArgument("--headless");
options.AddArgument("--disable-dev-shm-usage");
options.AddArgument("--start-maximized");
options.AddArgument("--no-sandbox");

var driverPath = Path.Combine(Directory.GetCurrentDirectory(), "ejecutable/", "msedge.exe");

EdgeDriver driver = new EdgeDriver(driverPath, options);
driver.Navigate().GoToUrl("https://sede.registradores.org/site/mercantil");

IWebElement cookies = driver.FindElement(By.Id("closeCookiesInfo"));
cookies.Click();

IWebElement botonBuscar = driver.FindElement(By.LinkText("Buscar por sociedad"));
botonBuscar.Click();

IWebElement divEliminar = driver.FindElement(By.Id("captcha-0"));
driver.ExecuteScript("arguments[0].remove();", divEliminar);
divEliminar = driver.FindElement(By.Id("ccode"));
driver.ExecuteScript("arguments[0].remove();", divEliminar);

IWebElement radioButton = driver.FindElement(By.Id("tipoIdMerci"));
IJavaScriptExecutor jsExecutor = (IJavaScriptExecutor)driver;
jsExecutor.ExecuteScript("arguments[0].checked = true;", radioButton);

IWebElement boton = driver.FindElement(By.CssSelector("button[data-qa='buscarMercSubmit']"));
IWebElement element = driver.FindElement(By.Id("terminoBusquedaMerc"));
IJavaScriptExecutor js = (IJavaScriptExecutor)driver;

js.ExecuteScript("arguments[0].dataset.validationLength='4-15';", element);
js.ExecuteScript("arguments[0].classList.remove('disable');", element);
js.ExecuteScript("arguments[0].className = arguments[0].className.replace('error', 'valid');", element);
js.ExecuteScript("arguments[0].maxLength= '15';", element);
js.ExecuteScript("arguments[0].dataset.validationHasKeyupEvent = true;", element);
js.ExecuteScript("arguments[0].dataset.validationErrorLength = 'NIF debe tener entre 4 y 15 caracteres'", element);
js.ExecuteScript("arguments[0].dataset.validationHasKeyupEvent= true;", element);
js.ExecuteScript("arguments[0].value = '" + CIF + "';", element);
js.ExecuteScript("arguments[0].textContent = 'Valencia Club De Futbol';", element);

((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].removeAttribute('disabled')", boton);
((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].classList.remove('btn--disabled')", boton);

System.Threading.Thread.Sleep(5000);

boton.Click();

IWebElement tdElement = driver.FindElement(By.CssSelector("td[data-qa='resultNif1']"));

string spanText = tdElement.FindElement(By.TagName("span")).Text;

driver.Close();

if (spanText != CIF) return false;

return true;
```

Figura B.1. Script para Empresas

```

EdgeOptions options = new EdgeOptions();
options.AddArgument("--headless");
options.AddArgument("--start-maximized");

EdgeDriver driver = new EdgeDriver(options);

driver.Navigate().GoToUrl("https://www.dateas.com/es/explore/search?entity_name=asociaciones-organizaciones-espana");

IWebElement campoEntrada = driver.FindElement(By.Id("mainfield"));
IJavaScriptExecutor js = (IJavaScriptExecutor)driver;

js.ExecuteScript("arguments[0].value = 'Caritas';", campoEntrada);
IWebElement boton = driver.FindElement(By.CssSelector("button-red-large.button-centered"));
boton.Click();

IWebElement tabla = driver.FindElement(By.CssSelector("tbody"));
IWebElement primeraFila = tabla.FindElement(By.CssSelector("tr"));
IWebElement enlaceOrganizacion = primeraFila.FindElement(By.CssSelector("td[data-label='Organización'] a"));

driver.Navigate().GoToUrl(enlaceOrganizacion.GetAttribute("href"));

IWebElement filaCif = driver.FindElement(By.XPath("//tr[th='Cif']"));

string cif = filaCif.FindElement(By.XPath("td")).Text;

driver.Close();

if (cif != beneficiario.CIF) return false;

return true;

```

Figura B.2. Script para ONGs

Anexo C

La licencia GNU es un conjunto de licencias software creadas por Free Software Foundation (FSF), este conjunto tiene como objetivo promover el software libre y para llevarlo a cabo se establece una serie de condiciones sobre las que se puede utilizar, modificar o mantener. Con esta licencia se busca fomentar la continuidad del proyecto en caso de que el principal desarrollador no pueda continuar ampliando o manteniendo el producto.

En este proyecto se utiliza la licencia GPL (General Public License), esta es una de las licencias más utilizadas en el mundo del desarrollo de software libre. Con esta licencia para llevar a cabo las condiciones mencionadas anteriormente se debe actuar de la misma forma que los usuarios anteriores, es decir, que todos los derechos y responsabilidades de la licencia deben ser respetados por todos los usuarios que utilicen el software. Esto implica que cualquier tipo de modificación o distribución del código fuente se deban cumplir los términos de la licencia GNU GPL.

Para aplicar la licencia al proyecto se ha incluido un archivo de texto LICENSE.txt tanto en el front-end como en el back-end en el que se incluye el texto completo de la licencia GNU GPL¹⁸.

Además, se debe incluir un encabezado de la licencia en cada archivo del código del proyecto en el que se especifique claramente cuál es la licencia con la que se va a distribuir el código con una referencia al archivo LICENSE.txt donde se encuentra toda la información de la licencia, como se observa en la figura C.1.

```
// <Aplicación destinada a facilitar la colaboración entre Empresas y ONGs>
// Copyright (C) <2023> <Ángel Pérez Martín>

// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.

// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.

// You should have received a copy of the GNU General Public License
// in the LICENSE.txt file. If not, see <https://www.gnu.org/licenses/>.
```

Figura C.1. Encabezado licencia GNU GPL

¹⁸ Texto completo de la licencia GNU GPL <https://www.gnu.org/licenses/gpl-3.0.html>