



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE SEGOVIA**

**Grado en Ingeniería Informática de  
Servicios y Aplicaciones**

---

**Plataforma Innovadora De Comercio Social (App web)  
e-Change**

---

**Alumno: Sergio Fagúndez Manso**

**Tutor/a/es: María del Pilar Grande González**





*"La programación es el lenguaje  
en el que se escribe el futuro."*

*Steve Jobs*

# Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas las personas que me han brindado su apoyo desde el principio.

A mi querida familia, amigos y seres queridos, les doy las gracias por ayudarme a convertirme en una mejor persona. Su amor, apoyo y confianza han sido fundamentales en mi crecimiento y éxito.

En especial, quiero agradecer a la Universidad, el lugar que me ha acompañado durante este trayecto, y a los profesores. También quiero expresar mi gratitud a mi tutora, María del Pilar Grande González, ya que, sin su guía y dedicación, nada de esto habría sido posible.

# Resumen

En la actualidad, vivimos en un mundo impulsado por la tecnología y el cambio constante. A medida que avanzamos, surgen nuevas ideas y desafíos tanto en nuestra vida diaria como a nivel social, afectándonos a todos.

El objetivo de este Trabajo de Fin de Grado es desarrollar una aplicación web que combine el popular concepto de e-commerce con la ayuda social, permitiendo a los usuarios comprar y vender productos para generar un cambio positivo. La clave del cambio radica en ti. Para lograr esto, se establecen objetivos con una meta de dinero a recaudar mediante la venta de diversos productos de moda. A medida que los usuarios realizan compras, una barra de progreso muestra la cantidad recaudada, permitiendo al creador y a los demás usuarios seguir el avance hacia el objetivo.

**Palabras clave:** Tecnología, cambio, e-commerce, ayuda social, compra, venta, objetivo, progreso, recaudación, productos de moda.

# Abstract

Nowadays, we live in a world driven by technology and constant change. As we progress, new ideas and challenges emerge in our daily lives and at the social level, affecting us all.

The aim of this Final Degree Project is to develop a web application that combines the popular concept of e-commerce with social aid, allowing users to buy and sell products to generate positive change. The key to change lies in you. To achieve this, goals are set with a target amount of money to be raised through the sale of various fashion products. As users make purchases, a progress bar shows the amount raised, allowing the creator and other users to track the progress towards the goal.

**Keywords:** Technology, change, e-commerce, social aid, buying, selling, goal, progress, fundraising, fashion products.

# Índice general

<b>Resumen .....</b>	<b>6</b>
<b>Abstract.....</b>	<b>7</b>
<b>Índice general .....</b>	<b>8</b>
<b>Índice de figuras .....</b>	<b>11</b>
<b>Índice de tablas.....</b>	<b>13</b>
<b>Descripción del proyecto.....</b>	<b>17</b>
1.1.- Introducción.....	17
1.2.- Motivación .....	18
1.3.- Objetivos del Proyecto.....	19
1.4.- Entorno de la aplicación .....	20
1.5.- Tecnologías .....	21
1.5.1.- MongoDB .....	21
1.5.2.- Express .....	21
1.5.3.- React .....	22
1.5.4.- Node.js .....	22
1.5.5.- Stripe.....	22
1.5.6.- TypeScript .....	23
1.5.7.- Visual Studio .....	23
1.5.8.- Microsoft Word.....	23
1.5.9.- Draw.io.....	24
1.5.10.- GitHub.....	24
1.5.11.- Npm .....	25
1.5.12.- Cloudinary.....	25
1.6.- Estructura de la documentación.....	26
<b>Estado del arte.....</b>	<b>28</b>
2.1.- Etsy.....	28
2.2.- Vinted.....	29
2.3.- Change.org.....	31
2.4.- Comparativa.....	32
<b>Planificación, estimación y presupuesto .....</b>	<b>34</b>
3.1.- Metodología de trabajo .....	34
3.2.- Planificación temporal .....	35

3.2.1.- Estimación del Esfuerzo .....	35
3.2.2.- Presupuesto económico .....	40
3.2.3.- Balance final.....	43
<b>Análisis del sistema.....</b>	<b>46</b>
4.1.- Actores del sistema.....	46
4.2.- Requisitos de usuario.....	48
4.3.- Requisitos funcionales .....	49
4.4.- Requisitos no funcionales .....	51
4.4.1.- Usabilidad .....	51
4.4.2.- Eficiencia .....	51
4.4.3.- Mantenibilidad.....	51
4.4.4.- Seguridad .....	52
4.4.5.- Disponibilidad .....	52
4.4.6.- Portabilidad.....	52
4.4.7.- Implementación.....	52
4.5.- Requisitos de información .....	53
4.6.- Diagrama de casos de uso .....	53
4.7.- Especificación de casos de uso .....	55
<b>Diseño .....</b>	<b>66</b>
5.1.1.- Arquitectura lógica no detallado .....	66
5.1.2.- Arquitectura lógica detallado .....	67
5.2.- Arquitectura física.....	68
5.3.- Modelo E-R .....	68
5.4.- Diccionario de datos .....	69
5.5.- Modelo lógico de datos .....	71
5.6.- Diagrama de secuencia .....	72
5.7.- Diseño de interfaz.....	73
6.4.1.- Paleta de Colores .....	81
<b>Implementación.....</b>	<b>86</b>
6.1.- Backend.....	86
6.2.- Frontend .....	91
6.3.- Modelo seguido .....	97
6.4.- Dificultades .....	98
6.5.- Despliegue en la nube.....	99
<b>Pruebas .....</b>	<b>102</b>
8.1.- Caja negra .....	102
8.2.- Caja blanca.....	107
<b>Manual de usuario .....</b>	<b>110</b>

8.1.- Manual de instalación.....	110
8.2.- Manual de usuario .....	112
8.2.1.- Usuario anónimo: .....	113
8.2.1.- Administrador .....	127
9.1.- Conclusiones .....	129
9.2.- Trabajo futuro .....	130
9.2.1.- Sistema de notificaciones .....	130
9.2.2.- Validación por admin .....	130
9.2.3.- Gestión de envíos.....	130
9.2.4.- Remitir dinero .....	130
9.2.5.- Productos .....	131
9.2.6.- Redes sociales .....	131
9.2.7.- Aplicación móvil.....	131
<b>Webgrafía.....</b>	<b>133</b>

# Índice de figuras

Figura 1.1: Logo e-Change.....	18
Figura 1.2: Árbol de Características .....	20
Figura 1.3: Logo MongoDB.....	21
Figura 1.4: Logo Express.....	21
Figura 1.5: Logo React.....	22
Figura 1.6: Logo Node.js.....	22
Figura 1.7: Logo Stripe .....	22
Figura 1.8: Logo TypeScript.....	23
Figura 1.9: Logo Visual Studio .....	23
Figura 1.10: Logo Microsoft Word .....	23
Figura 1.11: Logo Draw.io .....	24
Figura 1.12: Logo GitHub.....	24
Figura 1.13: Logo Npm .....	25
Figura 1.14: Logo Cloudinary.....	25
Figura 2.15: Logo Etsy .....	29
Figura 2.16: Logo Vinted .....	30
Figura 2.17: Logo Change.org.....	31
Figura 3.18: Ciclo de vida en cascada.....	34
Figura 3.19: Diagrama de Gantt Estimación .....	42
Figura 3.20: Calendario de trabajo.....	43
Figura 3.21: Fases Proyecto Real.....	44
Figura 3.22: Diagrama de Gantt Real .....	44
Figura 4.23: Jerarquía de actores.....	47
Figura 4.24: Diagrama de casos de uso de Usuario Anónimo.....	53
Figura 4.25: Diagrama de casos de uso de Usuario Autenticado.....	54
Figura 4.26: Diagrama de casos de uso de Admin .....	54
Figura 5.27: Diseño Lógico No detallado.....	66
Figura 5.28: Diseño Lógico Detallado.....	67
Figura 5.29: Diseño Arquitectura Física .....	68
Figura 5.30: Modelo E-R.....	69
Figura 5.31: Modelo Lógico de Datos.....	71
Figura 5.32: Diagrama de secuencia CU_06: Iniciar sesión.....	72
Figura 5.33: Diagrama de secuencia CU_13: Crear Objetivos.....	73
Figura 5.34: Paleta de colores .....	81
Figura 5.35: Color Botones de acción .....	82
Figura 5.36: Color Botones de eliminación .....	82
Figura 5.37: Página Objetivo Responsive .....	82
Figura 5.38: Página Inicio Bento-Box .....	83
Figura 5.39: Página Producto .....	83
Figura 5.40: Logo e-Change.....	84
Figura 6.41: Estructura Carpetas Backend .....	86
Figura 6.42: Modelo Objetivo .....	87
Figura 6.43: Módulos y funciones .....	88
Figura 6.44: API user .....	89
Figura 6.45: Función VerifyToken .....	89
Figura 6.46: index.ts.....	90

Figura 6.47: Estructura de carpetas Frontend .....	92
Figura 6.48: Hook useGetObjetivos .....	93
Figura 6.49: App.tsx .....	96
Figura 6.50: Roles .....	96
Figura 6.51: MVC.....	98
Figura 6.52: Logo Vercel.....	99
Figura 8.53: Página Inicio Bento-Box .....	113
Figura 8.54: Página Objetivos.....	114
Figura 8.55: Página Detalles Objetivo .....	115
Figura 8.56: Página Producto .....	116
Figura 8.57: Toast Añadir a Favoritos .....	116
Figura 8.58: Toast Añadir al Carrito .....	116
Figura 8.59: Barra de Navegación con contenido .....	116
Figura 8.60: Pestaña Favoritos con contenido.....	117
Figura 8.61: Toast de eliminación de producto de Favoritos.....	117
Figura 8.62: Pestaña Favoritos sin contenido .....	117
Figura 8.63: Toast indicando que el producto se ha añadido al Carrito .....	117
Figura 8.64: Pestaña Carrito con Prodcutos.....	118
Figura 8.65: Toast eliminación del producto del Carrito.....	118
Figura 8.66: Pestaña Carrito sin Prodcutos .....	118
Figura 8.67: Página Sign up .....	119
Figura 8.68: Página Iniciar Sesión.....	120
Figura 8.69: Página Objetivos Usuario Autenticado .....	121
Figura 8.70: Página Creación de Objetivos .....	121
Figura 8.71: Página Objetivos Usuario Autenticado .....	122
Figura 8.72: Modificación de Objetivos .....	122
Figura 8.73: Añadir Producto .....	123
Figura 8.74: Eliminar Producto.....	123
Figura 8.75: Compra de Producto .....	124
Figura 8.76: Validación de Compra .....	124
Figura 8.77: Compra exitosa .....	125
Figura 8.78: Compra Fallida .....	125
Figura 8.79: Pedidos.....	125
Figura 8.80: Perfil .....	126
Figura 8.81: Editar Perfil.....	126
Figura 8.82: Panel de Administrador .....	127

# Índice de tablas

Tabla 2.1: Comparativa entre aplicaciones.....	32
Tabla 3.2: Tabla de Grado de Complejidad para ALI y AIE .....	36
Tabla 3.3: Tabla de Grado de Complejidad para EE y CE .....	37
Tabla 3.4: Tabla de Grado de Complejidad para SE .....	37
Tabla 3.5: Tabla de Entradas de Usuario (EE) .....	37
Tabla 3.6: Tabla de Salidas Externas (SE) .....	37
Tabla 3.7: Tabla de Consultas Externas (CE) .....	38
Tabla 3.8: Tabla de Archivos Lógicos Internos (ALI) .....	38
Tabla 3.9: Tabla de Archivos de Interfaz Externos (AIE) .....	38
Tabla 3.10: Tabla Conversión por Complejidad .....	38
Tabla 3.11: Tabla Puntos de Función Sin Ajustar (PFSA).....	39
Tabla 3.12: Tabla Factor de Ajuste (FA) .....	39
Tabla 3.13: Tabla Costes Hardware.....	40
Tabla 3.14: Tabla Costes Software .....	41
Tabla 3.15: Tabla Otros Costes.....	41
Tabla 3.16: Tabla Coste Recursos Humanos .....	42
Tabla 3.17: Tabla Coste Total Estimado .....	42
Tabla 3.18: Estimación Fases Proyecto .....	42
Tabla 4.19: Actores del sistema .....	46
Tabla 4.20: Requisitos de usuario .....	48
Tabla 4.21: Requisitos Funcionales .....	50
Tabla 4.22: Requisitos No Funcionales- Usabilidad .....	51
Tabla 4.23: Requisitos No Funcionales- Eficiencia .....	51
Tabla 4.24: Requisitos No Funcionales- Mantenibilidad.....	51
Tabla 4.25: Requisitos No Funcionales- Seguridad .....	52
Tabla 4.26: Requisitos No Funcionales- Disponibilidad .....	52
Tabla 4.27: Requisitos No Funcionales- Portabilidad.....	52
Tabla 4.28: Requisitos No Funcionales- Implementación .....	52
Tabla 4.29: Requisitos de Información.....	53
Tabla 4.30: Especificación de CU_01.....	55
Tabla 4.31: Especificación de CU_02.....	55
Tabla 4.32: Especificación de CU_03.....	56
Tabla 4.33: Especificación de CU_04.....	56
Tabla 4.34: Especificación de CU_05.....	57
Tabla 4.35: Especificación de CU_06.....	57
Tabla 4.36: Especificación de CU_07.....	58
Tabla 4.37: Especificación de CU_08.....	58
Tabla 4.38: Especificación de CU_09.....	59
Tabla 4.39: Especificación de CU_10.....	59
Tabla 4.40: Especificación de CU_11.....	60
Tabla 4.41: Especificación de CU_12.....	60
Tabla 4.42: Especificación de CU_13.....	61
Tabla 4.43: Especificación de CU_14.....	61
Tabla 4.44: Especificación de CU_15.....	62
Tabla 4.45: Especificación de CU_16.....	62
Tabla 4.46: Especificación de CU_17.....	63

Tabla 4.47: Especificación de CU_18.....	63
Tabla 4.48: Especificación de CU_19.....	64
Tabla 4.49: Especificación de CU_20.....	64
Tabla 5.50: Diccionario de datos Usuario .....	69
Tabla 5.51: Diccionario de datos Objetivo .....	70
Tabla 5.52: Diccionario de datos Producto .....	70
Tabla 5.53: Diccionario de datos Producto .....	70
Tabla 5.54: Diseño Interfaz Vista Principal.....	74
Tabla 5.55: Diseño Interfaz Vista Registro .....	74
Tabla 5.56: Diseño Interfaz Vista Iniciar Sesión .....	75
Tabla 5.57: Diseño Interfaz Vista Buscar Objetivos .....	75
Tabla 5.58: Diseño Interfaz Vista Crear Objetivo .....	76
Tabla 5.59: Diseño Interfaz Vista Añadir Producto .....	76
Tabla 5.60: Diseño Interfaz Vista Navbar .....	77
Tabla 5.61: Diseño Interfaz Vista Mis Objetivos .....	77
Tabla 5.62: Diseño Interfaz Vista Objetivo.....	78
Tabla 5.63: Diseño Interfaz Vista Producto.....	78
Tabla 5.64: Diseño Interfaz Vista Carrito .....	79
Tabla 5.65: Diseño Interfaz Vista Favoritos.....	79
Tabla 5.66: Diseño Interfaz Vista Modificar Perfil.....	80
Tabla 5.67: Diseño Interfaz Vista Realizar Pago .....	80
Tabla 5.68: Diseño Interfaz Vista Eliminar Objetivos Admin.....	81
Tabla 7.69: PCN_01 Registro de Usuario .....	102
Tabla 7.70: PCN_02 Autenticación de Usuario .....	102
Tabla 7.71: PCN_03 Creación de Objetivo .....	103
Tabla 7.72: PCN_04 Modificar Objetivo.....	103
Tabla 7.73: PCN_05 Eliminar Objetivo .....	104
Tabla 7.74: PCN_06 Añadir Producto.....	104
Tabla 7.75: PCN_07 Añadir Producto al Carrito .....	104
Tabla 7.76: PCN_08 Añadir Producto a Favoritos .....	105
Tabla 7.77: PCN_09 Eliminar Producto .....	105
Tabla 7.78: PCN_10 Eliminar Producto de Favoritos .....	105
Tabla 7.79: PCN_11 Eliminar Producto del Carrito .....	105
Tabla 7.80: PCN_12 Editar Perfil de Usuario .....	106
Tabla 7.81: PCN_13 Cerrar Sesión.....	106
Tabla 7.82: PCN_14 Buscar Objetivo.....	106
Tabla 7.83: PCN_15 Filtrar Productos .....	106
Tabla 7.84: PCN_16 Realizar Pago.....	107





# Capítulo 1

## Descripción del proyecto

### 1.1.- Introducción

Hoy en día, la tecnología y la innovación están transformando todos los aspectos de nuestra vida, desde lo personal hasta lo social. Esta evolución constante ha permitido la integración de conceptos como el e-commerce en nuestra rutina diaria, revolucionando nuestra forma de comprar y vender productos.

La idea es permitir a los usuarios comprar y vender productos, generando un cambio beneficioso para la sociedad. La clave del cambio radica en la participación activa de cada usuario.

Para lograr esto, se emplea el uso de objetivos que explican la acción social y el origen de este cambio. Además, se establece una meta de dinero a recaudar que se conseguirá mediante la venta de diversos productos de moda. Cada producto vendido contribuye a un fondo específico, y a medida que los usuarios realizan compras, una barra de progreso muestra la cantidad recaudada. Esto permite al creador y a los demás usuarios seguir el avance hacia el objetivo, fomentando la transparencia y la motivación colectiva.

Según el informe “[1] E-commerce y responsabilidad social” de 2020, un 70% de los consumidores españoles muestran una mayor inclinación a comprar productos de empresas que demuestran un compromiso social. Este dato resalta la importancia de iniciativas que no solo buscan el beneficio económico, sino también el bienestar social.

Este proyecto propone una plataforma donde los usuarios pueden fijar metas específicas, como financiar proyectos comunitarios, apoyar causas benéficas o contribuir a la sostenibilidad ambiental. A través de la venta de productos de moda, los usuarios no solo satisfacen sus necesidades de consumo, sino que también participan activamente en la creación de un mundo mejor.

Además, la aplicación ofrecerá una experiencia de usuario optimizada, permitiendo a los compradores ver el impacto de sus compras en tiempo real. Esta transparencia y la posibilidad de ver el progreso hacia los objetivos establecidos son factores clave que diferencian esta plataforma de otras aplicaciones de e-commerce tradicionales.

## 1.2.- Motivación

La idea de desarrollar esta plataforma surge debido al auge del comercio electrónico en los últimos años, que ha llevado a muchas personas a realizar más compras en línea que de manera presencial. Inspirado por esta tendencia y mi propio interés en la moda, pensé en crear algo innovador dentro del sector de la ropa, que ya es bastante común en el comercio electrónico. Sin embargo, quería darle un enfoque único y diferenciador.

Después de utilizar la aplicación [13] Vinted, una plataforma de compra y venta de ropa de segunda mano, se me ocurrió desarrollar una aplicación similar, pero con una característica especial: permitir a los usuarios crear y apoyar causas sociales o proyectos personales. Recordé los viajes escolares en los que vendíamos boletos de lotería o pulseras para financiar nuestras excursiones, y pensé en cómo esta idea podría adaptarse a un contexto digital.

Así nació e-Change, una plataforma que no solo facilita la compra y venta de ropa, sino que también permite a los usuarios apoyar causas sociales y proyectos personales a través de sus transacciones. Además, la oportunidad de aprender y utilizar la tecnología [12] MERN (MongoDB, Express.js, React y Node.js) me motivó a combinar esta idea con la implementación de esta moderna stack de desarrollo, creando una solución tecnológica robusta y a la vanguardia.

Con e-Change, no solo se busca satisfacer las demandas actuales del mercado de moda en línea, sino también promover un consumo más consciente y solidario. Esta plataforma ofrece a los usuarios la posibilidad de comprar y vender ropa mientras apoyan proyectos significativos, fusionando el comercio electrónico con el apoyo a causas sociales de una manera innovadora y relevante.



Figura 1.1: Logo e-Change

## 1.3.- Objetivos del Proyecto

El proyecto se centra en desarrollar una aplicación web que fusiona compras con un propósito social, brindando a los usuarios la oportunidad de adquirir productos mientras respaldan diversas causas benéficas. Al mismo tiempo, ofrece a los usuarios que deseen vender, una plataforma para comercializar sus productos con un propósito definido. Este objetivo principal, establece el marco para una serie de factores adicionales que mejorarán la funcionalidad y la experiencia de usuario. A continuación, se detallan los objetivos específicos que impulsarán el logro de esta meta:

<b>Objetivo 1</b>	Desarrollar una plataforma web innovadora centrada en productos de moda, enfocada en la responsabilidad social y el financiamiento colectivo.
<b>Objetivo 1.1</b>	Implementar medidas de seguridad para proteger la información sensible y mantener la integridad de los datos de usuarios registrados en la plataforma.
<b>Objetivo 1.2</b>	Permitir la gestión de ventas de productos de moda vinculados a causas sociales o proyectos significativos a través de la plataforma, proporcionando una experiencia fluida y segura para los usuarios.
<b>Objetivo 1.3</b>	Facilitar el proceso de recaudación de fondos mediante la visualización en tiempo real del progreso hacia el objetivo financiero establecido, utilizando una barra de progreso dinámica que se actualiza con cada compra relacionada con la causa.
<b>Objetivo 2</b>	Ofrecer una interfaz intuitiva que permita a los usuarios explorar y apoyar diversas iniciativas sociales relacionadas con la moda de manera conveniente y fácil de entender.
<b>Objetivo 3</b>	Brindar opciones de personalización para usuarios, permitiendo subir imágenes, añadir descripciones al crear y gestionar objetivos y editar datos del perfil.
<b>Objetivo 4</b>	Desarrollar un sistema de seguimiento de recaudación para garantizar la transparencia y el cumplimiento de los objetivos de cada iniciativa.
<b>Objetivo 5</b>	Usar un MERN stack que incluya: MongoDB como Sistema de Bases de Datos no relacionales, Express.js como framework para el backend, React para el frontend y Node.js como entorno de tiempo de ejecución de JavaScript para el servidor.

*Tabla 1.1: Objetivos*



Se denomina MERN al estar compuesto por MongoDB, Express.js, React y Node.js, una combinación poderosa para el desarrollo web. Utilizando JavaScript, tanto en el cliente como en el servidor, permite un desarrollo más eficiente y coherente. MongoDB [8] ofrece una base de datos escalable y flexible, Express.js facilita la creación de APIs robustas, React proporciona interfaces de usuario dinámicas y reutilizables, y Node.js asegura un rendimiento elevado y manejo eficiente de conexiones. Esta tecnología garantiza una plataforma rápida, escalable y de alta calidad, ideal para integrar nuestro e-commerce de apoyo a causas sociales en una aplicación.

## 1.5.- Tecnologías

Aquí, se describirán las tecnologías empleadas para el desarrollo software de este proyecto.

### 1.5.1.- MongoDB

MongoDB [8] es una base de datos no relacional (NoSQL) que almacena datos en formato de documentos JSON. Su facilidad de uso y flexibilidad nos permite gestionar todos los datos de nuestra aplicación de manera eficiente. MongoDB es ideal para aplicaciones que requieren un manejo dinámico y escalable de grandes volúmenes de datos. En nuestro proyecto, hemos utilizado MongoDB para almacenar y consultar datos de forma rápida y eficaz, lo que ha contribuido a mantener un rendimiento óptimo y a simplificar el desarrollo del backend.



*Figura 1.3: Logo MongoDB [8]*

### 1.5.2.- Express

Express [19] es un framework para Node.js que facilita la creación de APIs. Se utiliza para crear el backend de aplicaciones. En nuestro proyecto ha sido implementado para el manejo de rutas y solicitudes del servidor de manera eficiente, permitiendo una comunicación fluida entre el frontend y la base de datos.



*Figura 1.4: Logo Express [19]*

### 1.5.3.- React

React [16] es una biblioteca de JavaScript diseñada para construir interfaces de usuario. Es especialmente útil para aplicaciones web, ya que permite crear componentes reutilizables y mantener una arquitectura de código modular y eficiente. En nuestro proyecto, hemos utilizado React para el desarrollo del frontend de nuestra aplicación, lo que nos ha permitido construir una interfaz de usuario dinámica y reactiva.



Figura 1.5: Logo React [16]

### 1.5.4.- Node.js

Node.js [2] es un entorno de ejecución para JavaScript. Permite construir aplicaciones web rápidas y escalables. Se usa para desarrollar el backend manejando solicitudes del servidor de manera eficiente facilitando la comunicación entre el frontend y la base de datos. Gracias a su arquitectura basada en eventos y su modelo de E/S no bloqueante, Node.js es ideal para aplicaciones en tiempo real y de alta demanda.



Figura 1.6: Logo Node.js [2]

### 1.5.5.- Stripe

Stripe [4] es una plataforma en línea diseñada para procesar pagos de forma segura. Es fácil de integrar en sitios web y aplicaciones, lo que la convierte en una opción conveniente para gestionar transacciones. Gracias a Stripe, he podido manejar los pagos en mi web de manera eficiente, facilitando las compras de los usuarios y garantizando una experiencia de pago segura y fluida.



Figura 1.7: Logo Stripe [4]

## 1.5.6.- TypeScript

TypeScript es un lenguaje de programación basado en JavaScript, pero con tipos estáticos. Permite especificar los tipos de variables y parámetros de funciones en tiempo de compilación. Otra ventaja es que ayuda a encontrar errores más fácilmente y hace que el código sea más claro.



*Figura 1.8: Logo TypeScript*

## 1.5.7.- Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE) para programar. Se ha tomado la decisión de usar este entorno para la realización del código del proyecto ya que es muy completo y te permite implementar varios lenguajes de programación. Además, te permite añadir de una forma muy sencilla el control de versiones de GitHub facilitando la creación de la aplicación software.



*Figura 1.9: Logo Visual Studio*

## 1.5.8.- Microsoft Word

Microsoft Word es un programa utilizado para crear y editar documentos de texto. Su uso ha sido fundamental para la elaboración de esta documentación. Con sus diversas herramientas de edición y formato, Microsoft Word nos ha permitido organizar y presentar la información de manera clara y profesional.



*Figura 1.10: Logo Microsoft Word*

## 1.5.9.- Draw.io

Draw.io es una herramienta gratuita en línea que permite crear diagramas y gráficos de manera sencilla y eficiente. Su interfaz intuitiva y fácil de usar facilita la creación de diagramas, lo que la hace ideal para usuarios de todos los niveles de habilidad. En nuestro proyecto, hemos utilizado Draw.io para la realización de todos los diagramas necesarios, incluyendo diagramas de arquitectura, de secuencia y más. Esta herramienta nos ha permitido visualizar y planificar mejor la estructura y el funcionamiento del sistema.

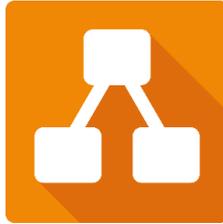


Figura 1.11: Logo Draw.io

## 1.5.10.- GitHub

GitHub [\[17\]](#) es una plataforma en la nube que emplea un sistema de control de versiones, permitiendo a los desarrolladores compartir y colaborar en su código de manera eficiente. Utiliza Git para hacer un seguimiento detallado de los cambios realizados en el código. Esta plataforma proporciona herramientas para la revisión de código, gestión de proyectos y automatización de flujos de trabajo, haciendo que el desarrollo sea más organizado y productivo.

En nuestro proyecto, utilizamos GitHub para llevar un control exhaustivo de las versiones del proyecto a medida que evolucionaba. Cada cambio realizado en el código se registraba con un commit, permitiendo un historial claro y detallado de las modificaciones.



Figura 1.12: Logo GitHub [\[17\]](#)

## 1.5.11.- Npm

Npm [18] es un gestor de paquetes para JavaScript. Permite instalar librerías y herramientas de una manera muy sencilla. Se utiliza principalmente para la instalación de dependencias de Node.js, lo que ayuda a mantener y actualizar las librerías necesarias para el desarrollo y ejecución de aplicaciones JavaScript en el entorno de servidor. Además, npm ofrece funcionalidades para la gestión de versiones y la automatización de tareas, haciendo el desarrollo más eficiente y organizado.



Figura 1.13: Logo Npm [18]

## 1.5.12.- Cloudinary

Cloudinary [3] es un servicio de gestión de medios en la nube que permite almacenar, manipular, optimizar y entregar imágenes y videos de manera sencilla. Se utiliza para la gestión de activos multimedia en aplicaciones web y móviles. Además, Cloudinary optimiza automáticamente los archivos multimedia para garantizar una carga rápida y un rendimiento óptimo en todos los dispositivos y navegadores.



Figura 1.14: Logo Cloudinary [3]

## 1.6.- Estructura de la documentación

En este apartado se explica cómo está estructurada la documentación en diferentes capítulos. Cada capítulo, a su vez, se divide en múltiples secciones. A continuación, se proporciona una descripción general del contenido tratado en cada una de ellas.

- 1. Capítulo 1: Descripción del proyecto:** En esta sección se definen los aspectos generales del proyecto a modo de introducción, proporcionando el contexto de la plataforma, una motivación detallada sobre el origen del proyecto, los objetivos, el entorno tecnológico y las tecnologías.
- 2. Capítulo 2: Estado del arte:** En este capítulo se realiza un análisis de las diferentes plataformas que comparten características similares a e-Change, permitiendo así un estudio comparativo de sus funcionalidades para determinar su efectividad potencial.
- 3. Capítulo 3: Planificación, estimación y presupuesto:** En este apartado se explica la metodología empleada durante el desarrollo del proyecto, además de una planificación temporal y una estimación de costes según el método Albrecht. Por último, se presenta un desglose de los diferentes costes involucrados en la realización del proyecto.
- 4. Capítulo 4: Análisis del sistema:** Este capítulo incluye los requisitos propuestos por el usuario para cada funcionalidad del sistema. También se describe a los diferentes actores que interactúan con él y las diversas acciones que el usuario puede realizar mediante los casos de uso y sus especificaciones.
- 5. Capítulo 5: Diseño:** En esta sección se detallan las diferentes arquitecturas del sistema, incluyendo la arquitectura lógica, que define el correcto funcionamiento de la plataforma, y la arquitectura física, con sus diagramas correspondientes. Además, se describe el modelo E-R con su diccionario de datos, el modelado de datos, los diagramas de secuencia y, por último, el diseño de la interfaz.
- 6. Capítulo 6: Implementación:** En este apartado se detalla la organización del código, incluyendo la estructura de carpetas y archivos, así como los modelos utilizados.
- 7. Capítulo 7: Pruebas:** En este capítulo se explican las diferentes pruebas realizadas para verificar que la plataforma cumple con todos los requisitos y funciona correctamente.
- 8. Capítulo 8: Manuales:** Esta sección incluye las guías y manuales de usuario que detallan el proceso de instalación, así como las instrucciones para aprender y utilizar todos los aspectos esenciales de la aplicación desarrollada.
- 9. Capítulo 9: Conclusiones:** En este apartado se presentan las diversas conclusiones obtenidas tras la finalización de la plataforma, así como una serie de posibles mejoras futuras que podrían implementarse más adelante.
- 10. Capítulo 10: Webgrafía:** Este capítulo recopila todas las fuentes de información consultadas para la elaboración de la documentación.



# Capítulo 2

## Estado del arte

En este capítulo, se realiza un estudio de plataformas web con características similares a e-Change. Para ello, llevaremos a cabo un análisis detallado de las funcionalidades y servicios que proporciona cada una de estas plataformas.

A continuación, se presentan algunas aplicaciones y plataformas que combinan las funcionalidades de **e-commerce** y **causas sociales**:

### 2.1.- Etsy

Etsy [14] es una plataforma en línea que conecta a vendedores independientes con compradores interesados en productos únicos y creativos. Etsy es un mercado líder para artículos hechos a mano, vintage y materiales para artesanía. La plataforma fomenta la sostenibilidad y el apoyo a pequeños negocios independientes.

Sus funcionalidades son las siguientes:

1. Venta de Productos Hechos a Mano y Vintage:
  - a. Productos Hechos a Mano: Vendedores pueden vender productos que han creado ellos mismos.
  - b. Productos Vintage: Venta de artículos que tienen antigüedad
2. Personalización de Productos:
  - a. Ofrece la opción de personalizar los productos, permitiendo a los compradores solicitar modificaciones para crear artículos personalizados.
3. Tienda:
  - a. Los vendedores pueden crear y personalizar sus propias tiendas dentro de la plataforma, incluyendo logotipos, descripciones y políticas de la tienda. Esto ayuda a construir una marca y atraer a los clientes.
4. Sistema de Valoraciones y Reseñas:
  - a. Los compradores pueden dejar valoraciones y reseñas sobre sus compras, lo que proporciona transparencia y ayuda a otros compradores a tomar decisiones.

5. Pagos:
  - a. Solución de pago integrada que permite a los vendedores aceptar pagos de diversas formas, incluyendo tarjetas de crédito, PayPal y otros métodos de pago.
6. Envío:
  - a. Los vendedores pueden comprar e imprimir etiquetas de envío directamente desde Etsy, facilitando el proceso de envío y rastreo de pedidos.
7. Redes Sociales:
  - a. Los vendedores pueden integrar sus tiendas Etsy con plataformas de redes sociales para aumentar la visibilidad de sus productos y atraer a una audiencia más amplia.



*Figura 2.15: Logo Etsy*

## 2.2.- Vinted

Vinted [13] es una plataforma de mercado en línea que permite a sus miembros intercambiar, vender y comprar artículos, principalmente ropa y accesorios de segunda mano. Permite publicar productos categorizándolo por marca, tipo de prenda, color, talla entre otras. Además tiene un chat que permite al comprador comunicarse con el vendedor para posibles dudas, información sobre la prenda e incluso adjuntar más fotos sobre ella. Se puede usar tanto como aplicación web como en dispositivos móviles ya que carece de aplicación para estos últimos.

Sus funcionalidades son las siguientes:

1. Compra y Venta de Artículos de Segunda Mano:
  - a. Los usuarios pueden categorizar sus artículos para la venta además de, especificar detalles como precio y fotos del producto.
  - b. Los compradores pueden buscar y comprar artículos directamente a través de la plataforma.

2. Comunicación:
  - a. Mensajes privados para que los usuarios se comuniquen entre sí, hagan preguntas sobre los productos y negocien detalles de las transacciones.
3. Pagos:
  - a. Ofrece servicios de pago con protección del comprador y ofrece soporte en caso de problemas.
4. Aplicación Móvil:
  - a. Vinted tiene una aplicación móvil gratuita que permite a los usuarios gestionar sus compras y ventas de manera conveniente desde sus dispositivos móviles.
5. Perfil de Usuario:
  - a. Los usuarios pueden crear y personalizar sus perfiles, donde pueden mostrar sus artículos listados y recibir comentarios y calificaciones de otros usuarios, lo que ayuda a construir confianza dentro de la comunidad.

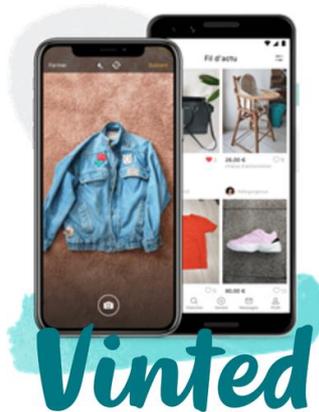


Figura 2.16: Logo Vinted

## 2.3.- Change.org

Change.org [15] es una plataforma abierta que permite a las personas crear y firmar peticiones para promover el cambio social, político y ambiental. Permite que las personas tomen acción sobre los temas que les importan, utilizando el poder de la comunidad para generar cambios significativos en la sociedad.

Sus funcionalidades son las siguientes:

1. Creación de Peticiones:
  - a. Permite a los usuarios crear peticiones para la posterior publicación.
2. Firma de Peticiones:
  - a. Cualquier persona puede firmar peticiones que coincida con sus valores e intereses. Las firmas pueden ser acompañadas de comentarios personales que refuercen el mensaje de la petición.
3. Redes Sociales:
  - a. Las peticiones pueden ser compartidas en redes sociales para ampliar su alcance y atraer más apoyos.
4. Comunicación:
  - a. Los creadores de peticiones pueden enviar mensajes a sus seguidores para mantenerlos informados sobre el progreso de la campaña. Esto ayuda a mantener el compromiso y la participación de los firmantes.
5. Recopilación de Fondos:
  - a. Ofrece la opción de recaudar fondos para apoyar campañas específicas. Estos fondos pueden ser utilizados para promover la petición, o apoyar otras actividades relacionadas con la causa.
6. Análisis y Reportes:
  - a. Change.org ofrece análisis y reportes sobre el rendimiento de las peticiones.



Figura 2.17: Logo Change.org

## 2.4.- Comparativa

Una vez realizadas las revisiones de las demás plataformas web similares, sus características y funcionalidades a las de e-Change, se realiza un análisis comparativo entre todas las demás y la nuestra, de tal forma que conseguiremos ver lo competente que puede ser nuestra web frente a las otras.

	<b>e-Change</b>	<b>Etsy</b>	<b>Vinted</b>	<b>Change.org</b>
<b>Compra y venta de productos</b>	No	Sí	Sí	No
<b>Soporte a causas sociales</b>	Sí	Sí	No	Sí
<b>Historial compras</b>	Sí	Sí	Sí	No
<b>Recopilación de firmas</b>	No	No	No	Sí
<b>Valoraciones y reseñas</b>	No	Sí	Sí	No
<b>Gestor de productos</b>	Sí	Sí	Sí	No
<b>Perfil usuario</b>	Sí	Sí	Sí	No
<b>Carrito de compras</b>	Sí	Sí	Sí	No
<b>Favoritos</b>	Sí	Sí	Sí	No
<b>Búsqueda por filtros</b>	Sí	Sí	Sí	No
<b>Registro de lo recaudado</b>	Sí	No	No	No

*Tabla 2.1: Comparativa entre aplicaciones*



# Capítulo 3

## Planificación, estimación y presupuesto

### 3.1.- Metodología de trabajo

En este apartado se explica de forma detallada, la metodología que se ha llevado cabo para el desarrollo de este proyecto de fin de grado.

En este caso, se ha optado por elegir una metodología en cascada debido a sus características y adecuación a los requisitos del proyecto.

Esta metodología se basa en la división del desarrollo del proyecto en distintas etapas, cada una de las cuales se lleva a cabo solo cuando la anterior ha finalizado completamente. La elección de esta metodología se debe a que los requisitos del proyecto están muy claros desde el principio, y no se espera que sean susceptibles a cambios significativos.

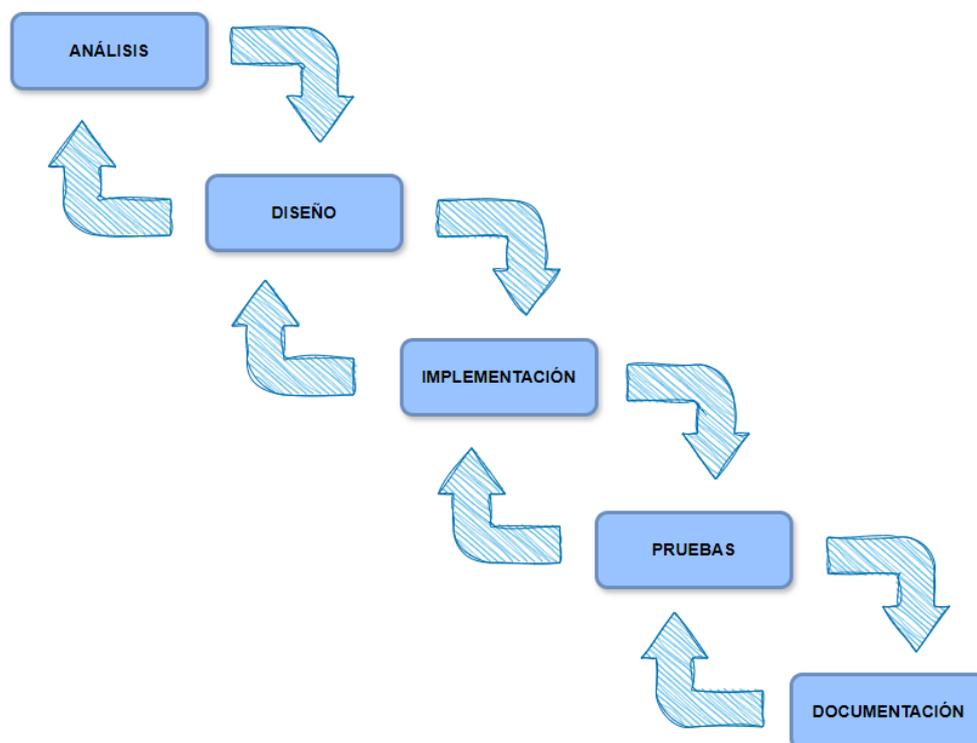


Figura 3.18: Ciclo de vida en cascada

### **Fases del Desarrollo en Cascada:**

El desarrollo del proyecto siguiendo la metodología en cascada se divide en las siguientes fases:

**1.-Análisis de Requisitos:** En esta fase se recopilan y documentan todos los requisitos del proyecto. Se asegura que todas las especificaciones estén claras y completas para evitar cambios futuros. Para ello, se realiza un diagrama de Gantt con la planificación y un presupuesto.

**2.-Diseño del Sistema:** Basado en los requisitos recopilados, se diseña la arquitectura del sistema. Esto incluye la estructura de la base de datos, la arquitectura lógica y física del sistema, y la elaboración de diagramas.

**3.-Implementación:** En esta fase se lleva a cabo la codificación del proyecto conforme al diseño definido en la fase anterior.

**4.-Pruebas:** Una vez que la implementación está completa, se realizan pruebas exhaustivas para asegurar que el sistema cumple con los requisitos especificados y funciona correctamente.

**5.-Documentación:** Finalmente, se realiza una documentación de todo el proceso de desarrollo del sistema.

### **Ventajas e Inconvenientes**

La metodología en cascada presenta varias ventajas, como la claridad en la planificación y la reducción de riesgos debido a la definición temprana de requisitos. Sin embargo, también tiene inconvenientes, como la dificultad para adaptarse a cambios una vez que el proyecto ha avanzado a fases posteriores.

En este proyecto, la elección de la metodología en cascada es adecuada debido a la claridad y estabilidad de los requisitos desde el inicio, lo que permite un análisis completo y un desarrollo estructurado y organizado.

## **3.2.- Planificación temporal**

En esta sección de la documentación se llevará a cabo una medición del esfuerzo necesario para la realización del proyecto. Con los resultados obtenidos se realizará un Diagrama de Gantt para ver gráficamente la duración del proyecto.

### **3.2.1.- Estimación del Esfuerzo**

Para la estimación del esfuerzo se utilizará el método de puntos de función, conocido como el Método de Albrecht. Este método mide la funcionalidad proporcionada por el software desde la perspectiva del usuario, evaluando el tamaño del software en términos de puntos de función. Considera entradas, salidas, consultas, archivos internos y archivos externos para proporcionar una estimación precisa del esfuerzo requerido. El proceso se dividirá en tres fases:

1. **Identificación:** Reconocer y listar todos los elementos funcionales del sistema.
2. **Cálculo de los Puntos de Función No Ajustados:** Asignar valores a cada elemento funcional identificado.
3. **Ajuste de los Puntos de Función:** Aplicar factores de ajuste para reflejar las características específicas del proyecto y obtener una estimación final precisa.

Para la fase de identificación, necesitaremos listar todos los elementos funcionales del sistema en base a estas directrices:

1. **Entradas externas (EE):** Datos o controladores introducidos al sistema por el usuario.
2. **Salidas externas (SE):** Información generada por el sistema y enviada al usuario.
3. **Consultas externas (CE):** Entradas que requieren una respuesta del sistema.
4. **Archivos lógicos internos (ALI):** Conjuntos de datos mantenidos y utilizados internamente por el sistema.
5. **Archivos de interfaz externos (AIE):** Conjuntos de datos utilizados por el sistema, pero mantenidos por otro sistema externo.

Para comenzar con el método, primero se deben clasificar los elementos de cada categoría según su complejidad. Se basa en el número de Tipos de Datos Elementales (TED) y el número de Tipos de Registros (TR).

1. **Tipos de Datos Elementales (TED):** Se refiere a un campo único, no repetitivo y comprensible para el usuario en un Archivo Lógico Interno (ALI) o en un Archivo de Interfaz Externo (AIE).
2. **Tipos de Registros (TR):** Es un subgrupo de datos reconocibles para el usuario dentro de un ALI o AIE.

Hay varias tablas que nos ayudan a determinar el grado de complejidad de cada elemento, dependiendo de la cantidad de TED y TR que contengan. En este caso, se eligen:

ALI o AIE	1 a 19 TED	20 a 50 TED	>51 TED
1 TR	Baja	Baja	Media
2 a 5 TR	Baja	Media	Alta
6 o más TR	Media	Alta	Alta

Tabla 3.2: Tabla de Grado de Complejidad para ALI y AIE

EE y CE	1 a 4 TED	5 a 15 TED	>15 TED
0-1 TR accedidos	Baja	Baja	Media
2 TR accedidos	Baja	Media	Alta
> 2 TR accedidos	Media	Alta	Alta

Tabla 3.3: Tabla de Grado de Complejidad para EE y CE

SE	1 a 5 TED	6 a 19 TED	>19 TED
0-1 FR accedidos	Baja	Baja	Media
2-3 FR accedidos	Baja	Media	Alta
> 3 FR accedidos	Media	Alta	Alta

Tabla 3.4: Tabla de Grado de Complejidad para SE

El siguiente paso será analizar cada uno de los elementos del proyecto:

<b>Entradas de Usuario (EE)</b>	
<b>Descripción</b>	<b>Grado de Complejidad</b>
Formulario de Registro	Baja
Formulario de Inicio de Sesión	Baja
Datos de Creación de Objetivo	Media
Datos de Añadir Producto	Baja
Datos de Pago	Baja
Formulario modificar Perfil	Baja
Datos modificar Objetivo	Baja

Tabla 3.5: Tabla de Entradas de Usuario (EE)

<b>Salidas Externas (SE)</b>	
<b>Descripción</b>	<b>Grado de Complejidad</b>
Página de Confirmación de Registro	Baja
Página de Confirmación de Inicio de Sesión	Baja
Vista de Objetivo Creado	Media
Vista de Producto Añadido	Baja
Recibo Pago	Baja

Tabla 3.6: Tabla de Salidas Externas (SE)

<b>Consultas Externas (CE)</b>	
<b>Descripción</b>	<b>Grado de Complejidad</b>
Búsqueda de Productos	Baja
Búsqueda de Objetivos	Baja
Ver Historial de Pagos	Bajo
Consulta de Detalles del Producto	Baja
Consulta de Detalles del Objetivo	Baja
Consulta Carrito	Baja
Consulta Favoritos	Baja

Tabla 3.7: Tabla de Consultas Externas (CE)

<b>Archivos Lógicos Internos (ALI)</b>	
<b>Descripción</b>	<b>Grado de Complejidad</b>
Archivo de Usuarios	Baja
Archivo de Objetivos	Baja
Archivo de Productos	Baja

Tabla 3.8: Tabla de Archivos Lógicos Internos (ALI)

<b>Archivos de Interfaz Externos (AIE)</b>	
<b>Descripción</b>	<b>Grado de Complejidad</b>
API de Pasarela de Pago	Media

Tabla 3.9: Tabla de Archivos de Interfaz Externos (AIE)

Una vez recopiladas las diferentes complejidades de los elementos de nuestro sistema, los resultados deberán ser trasladados a la tabla de conversión, asignando un peso a cada tipo de componente según su complejidad.

<b>Descripción/Complejidad</b>	<b>Baja</b>	<b>Media</b>	<b>Alta</b>
<b>Entradas</b>	x 3	x 4	x 6
<b>Salidas</b>	x 4	x 5	x 7
<b>Consultas externas</b>	x 3	x 4	x 6
<b>Ficheros internos</b>	x 7	x 10	x 15
<b>Ficheros externos</b>	x 5	x 7	x 10

Tabla 3.10: Tabla Conversión por Complejidad

<b>Puntos de Función Sin Ajustar (PFSA)</b>			
<b>Elemento</b>	<b>Complejidad</b>	<b>Número</b>	<b>Total por elemento</b>
<b>Entradas</b>	Baja (x3)	6	18
	Media(x4)	1	4
	Alta(x6)	0	0
<b>Salidas</b>	Baja (x4)	4	16
	Media (x5)	1	5
	Alta (x7)	0	0
<b>Consultas externas</b>	Baja (x3)	7	21
	Media (x4)	0	0
	Alta (x6)	0	0
<b>Ficheros internos</b>	Baja (x7)	3	21
	Media (x10)	0	0
	Alta (x15)	0	0
<b>Ficheros externos</b>	Baja (x5)	0	0
	Media (x7)	1	7
	Alta (x10)	0	0
<b>Total</b>			<b>92</b>

Tabla 3.11: Tabla Puntos de Función Sin Ajustar (PFSA)

Una vez calculados los PFSA totales, será necesario realizar un ajuste. Para ello, se debe calcular mediante el Factor de Ajuste (FA). Los cálculos de la FA se basan en 14 características generales de los sistemas que miden la funcionalidad y la complejidad/influencia de la aplicación. Todas las características se clasifican en una escala de cero a cinco. Un cero indica que el factor no tiene influencia ni complejidad, y cinco sería la mayor influencia o complejidad alcanzable.

<b>Factor de Ajuste (FA)</b>	
<b>Factor</b>	<b>Complejidad</b>
<b>1. Comunicación de datos</b>	3
<b>2. Funciones distribuidas</b>	2
<b>3. Prestaciones</b>	3
<b>4. Gran uso de la configuración</b>	4
<b>5. Velocidad de las transacciones</b>	3
<b>6. Entrada de datos en línea</b>	4
<b>7. Diseño para la eficiencia del usuario final</b>	4
<b>8. Actualización de datos en línea</b>	4
<b>9. Complejidad del proceso lógico interno de la aplicación</b>	3
<b>10. Reusabilidad del código por otras aplicaciones</b>	2
<b>11. Facilidad de instalación</b>	3
<b>12. Facilidad de operación</b>	4
<b>13. Localizaciones múltiples</b>	2
<b>14. Facilidad de cambios</b>	3
<b>Total</b>	<b>44</b>

Tabla 3.12: Tabla Factor de Ajuste (FA)

Una vez que se han asignado los valores a cada uno de los factores, se procede a calcular el Factor de Ajuste (FA). Para ello, utilizamos la siguiente ecuación:

$$FA = 0,65 + 0,01 \cdot \sum_{i=1}^{14} complejidad_i = 0,65 + 0,01 \cdot 44 = 1,09 FA$$

A continuación, se calculan los Puntos de Función Ajustados (PFA) multiplicando los Puntos de Función Sin Ajustar por el Factor de Ajuste.

$$PFA = PFSA \cdot FA = 92 \cdot 1,09 = 100,28$$

Para finalizar, se calcula el tiempo total que se estima que va a durar la realización del proyecto, en mi caso he asignado que punto de función ajustado equivale a 4 horas de trabajo.

$$Tiempo\ estimado = PFA \cdot N^{\circ}Horas / PFA = 100,28 \cdot 4\ horas = 401,12\ horas$$

El tiempo estimado que tendrá la realización del proyecto es de 401,12 horas.

Para calcular cuánto sería en meses tomamos con que la jornada laboral es, de lunes a viernes y se trabaja en torno a 8 horas diarias durante 21 días laborables.

$$21\ días/mes \cdot 8\ horas/día = 168\ horas/mes$$

$$Tiempo\ en\ meses = \frac{401,12\ horas}{168\ horas/mes} = 2.38\ meses$$

### 3.2.2.- Presupuesto económico

Antes de iniciar el proyecto, es fundamental estimar los costos de desarrollo. Para realizar esta estimación, se considerarán las herramientas de hardware y software utilizadas, así como los factores de impacto relacionados con la duración del proyecto y los gastos derivados del personal. Esta evaluación se basará en la estimación de horas planificadas y en los roles específicos asignados a cada tarea.

El costo total del proyecto se dividirá en cuatro categorías principales: hardware, licencias de software, otros costes y costos de personal.

<b>Tabla Coste Hardware</b>			
<b>Componentes</b>	<b>Coste (€)</b>	<b>Uso(%)</b>	<b>Total (€)</b>
Ordenador portátil	600,00€	5%	30,00€
Ratón	30,00€	2%	0,60€
Monitor	129,00€	5%	6,45€
Teclado	20,00€	2%	0,40€
<b>Total</b>			<b>37,45€</b>

Tabla 3.13: Tabla Costes Hardware

<b>Tabla Coste Software</b>			
<b>Licencia</b>	<b>Coste (€)</b>	<b>Uso(%)</b>	<b>Total (€)</b>
MongoDB	0,00€		0,00€
Visual Studio	0,00€		0,00€
Stripe	0,00€		0,00€
Draw.io	0,00€		0,00€
GitHub	0,00€		0,00€
Google Chrome	0,00€		0,00€
Windows 10	35,00€	8%	2,80€
Word	10,00€	6%	0,60€
<b>Total</b>			<b>3,40€</b>

Tabla 3.14: Tabla Costes Software

En este apartado, se calcularán otros costes que necesitamos contemplar en nuestro proyecto.

<b>Tabla Otros Costes</b>			
<b>Servicio</b>	<b>Coste (€/mes)</b>	<b>Tiempo (mes)</b>	<b>Total (€)</b>
Conexión a internet	40,00€	3	120,00€
Costes eléctricos	35,00€	3	105,00€
<b>Total</b>			<b>225,00€</b>

Tabla 3.15: Tabla Otros Costes

Para estimar los costos humanos del proyecto, es fundamental analizar los salarios anuales de los roles involucrados. En este caso, se han designado dos categorías: Desarrollador Junior y Analista. El salario anual promedio para un Desarrollador Junior según el Colegio Oficial de Ingenieros en Informática de España y estudios de mercado salariales en el sector tecnológico es de aproximadamente 22.450 euros, mientras que, para un Analista, el salario anual oscila entre los 35.000 euros. Con estos datos, se puede realizar una estimación precisa del costo del proyecto basado en el tiempo y los recursos humanos necesarios.

De las 401.12 horas estimadas para el proyecto, procederemos a dividir las horas asignadas según cada rol para determinar la carga de trabajo correspondiente a cada persona.

1. **Desarrollador Junior:** 60% de 401,12 horas = 240,672 horas
2. **Analista:** 40% de 401,12 horas = 160,448 horas

Para el Cálculo del Costo por Hora:

$$\text{Desarrollador Junior} = \frac{22.450\text{€}}{12 \text{ meses} \cdot 173,33 \text{ horas/mes}} = \frac{22.450\text{€}}{2080 \text{ horas/año}} = 10,79\text{€/hora}$$

$$\text{Analista} = \frac{35.000\text{€}}{12 \text{ meses} \cdot 173,33 \text{ horas/mes}} = \frac{35.000\text{€}}{2080 \text{ horas/año}} = 16,83\text{€/hora}$$

Tabla Coste Recursos Humanos			
Rol	Horas (h)	Coste/Hora(€)	Total (€)
Desarrollador	240,672	10,79€	2.596,00€
Analista	160,448	16,83€	2.700,00€
<b>Total</b>			<b>5.297,18€</b>

Tabla 3.16: Tabla Coste Recursos Humanos

Tras haber calculado todos los costes por separado, realizamos una tabla que recoja el total de todos los costes.

Tabla Coste Total	
Hardware	37,45€
Software	3,40€
Otros Costes	225,00€
Recursos Humanos	5.297,18€
<b>Total</b>	<b>5.563,03€</b>

Tabla 3.17: Tabla Coste Total Estimado

Nombre de tarea	Duración	Comienzo	Fin
Etapa de Análisis	11 días	jue 11/04/24	vie 19/04/24
Planificación	14 días	vie 19/04/24	jue 02/05/24
Diseño del Sistema	11 días	jue 02/05/24	lun 13/05/24
Implementación	21 días	lun 13/05/24	jue 30/05/24
Pruebas	8 días	jue 30/05/24	jue 06/06/24
Documentación	11 días	jue 06/06/24	lun 17/06/24

Tabla 3.18: Estimación Fases Proyecto

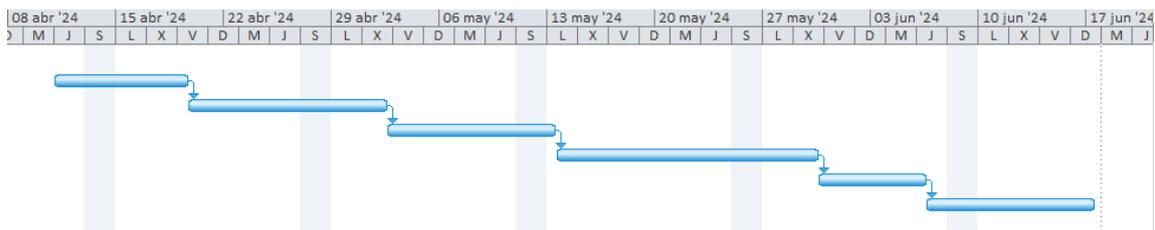


Figura 3.19: Diagrama de Gantt Estimación

### 3.2.3.- Balance final

La planificación inicial del proyecto no se ha cumplido por varias razones, lo que ha resultado en un aumento de la duración del mismo.

El proyecto estaba programado para comenzar el 11 de abril de 2024, con una jornada de trabajo de 8 horas diarias de lunes a viernes. Sin embargo, debido al período de prácticas en empresa y a la necesidad de atender las asignaturas del curso hasta el 15 de mayo, se ha trabajado una jornada de 5 horas diarias. A partir de esa fecha, se ha realizado una jornada de 8 horas diarias de lunes a viernes.

Tras las modificaciones se consigue terminar el proyecto el día **12 de Julio de 2024**.

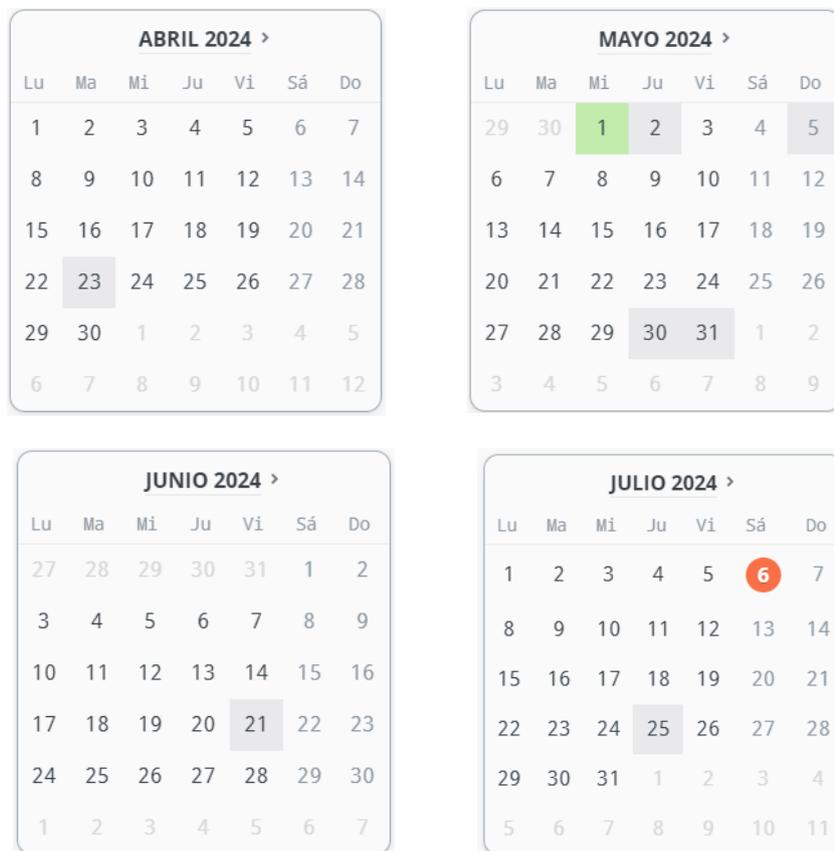


Figura 3.20: Calendario de trabajo

Respecto a lo estimado hay una diferencia de 26 días, período durante el cual se ha tenido que retrasar el proyecto por las causas mencionadas. Debido a esto se ha tenido que hacer un reajuste en el presupuesto.

Siguiendo los horarios anteriormente mencionados, se suman 130 horas adicionales al proyecto. Este incremento representa aproximadamente el 32,40% del tiempo planificado. Para calcular el presupuesto ajustado, multiplicamos nuestro presupuesto estimado de 5.563,03€ por 1,3240, lo que nos da 7.365,45€ como presupuesto real.

Nombre de tarea	Duración	Comienzo	Fin
<b>TFG project</b>	<b>106 días</b>	<b>jue 11/04/24</b>	<b>vie 12/07/24</b>
<b>Etapa de Análisis</b>	<b>15 días</b>	<b>jue 11/04/24</b>	<b>mié 24/04/24</b>
<b>Planificación</b>	<b>17 días</b>	<b>mié 24/04/24</b>	<b>mié 08/05/24</b>
<b>Diseño del Sistema</b>	<b>16 días</b>	<b>jue 09/05/24</b>	<b>mié 22/05/24</b>
<b>Implementación</b>	<b>30 días</b>	<b>jue 23/05/24</b>	<b>mar 18/06/24</b>
<b>Pruebas</b>	<b>11 días</b>	<b>mar 18/06/24</b>	<b>jue 27/06/24</b>
<b>Documentación</b>	<b>17 días</b>	<b>jue 27/06/24</b>	<b>vie 12/07/24</b>

Figura 3.21: Fases Proyecto Real

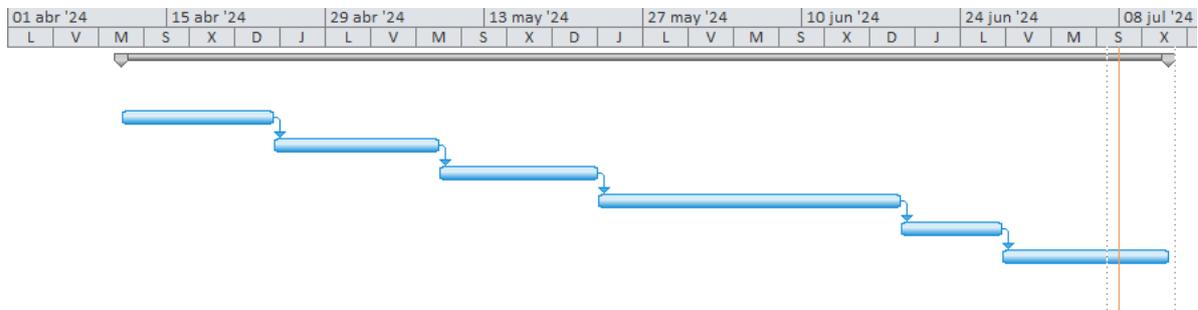


Figura 3.22: Diagrama de Gantt Real



# Capítulo 4

## Análisis del sistema

Aquí se desglosarán los distintos actores considerados en el sistema, incluyendo los requisitos que el sistema debe cumplir hasta las especificaciones y diagramas que los representan. Esta fase es prescindible para asegurarse de que el software se diseñe y construya correctamente.

### 4.1.- Actores del sistema

Los actores, ya sean individuos físicos o sistemas de software externos, representan los diversos roles que interactúan con el sistema. En este proyecto, se han identificado los siguientes:

ID	Nombre	Descripción
AC_01	Usuario Anónimo	Este actor representa a los visitantes del sistema que no han iniciado sesión. Pueden navegar y ver productos, pero tienen funcionalidades limitadas como, por ejemplo, no poder realizar compras ni modificar perfiles.
AC_02	Usuario Autenticado	Este actor representa a los usuarios que han iniciado sesión en el sistema. Pueden realizar compras, añadir productos al carrito, ver su historial de compras y modificar su perfil.
AC_03	Admin	Este actor representa a los administradores del sistema. Tienen permisos avanzados para gestionar causas sociales y recibir notificaciones de sus creaciones y modificaciones gestionando la plataforma para conseguir un uso correcto.
AC_04	Plataforma de pago (Stripe) [4]	Este actor representa a la pasarela de pago utilizada para procesar transacciones. Stripe maneja la verificación y autorización de pagos, garantizando transacciones seguras.

Tabla 4.19: Actores del sistema

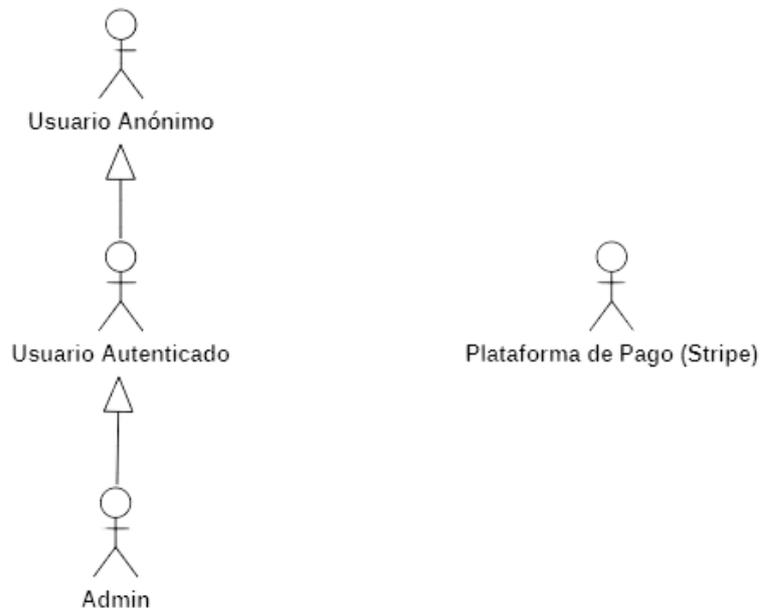


Figura 4.23: Jerarquía de actores

## 4.2.- Requisitos de usuario

Son todas aquellas funcionalidades y características que deberían satisfacer las necesidades del usuario final. La forma en que vamos a dividir los requisitos será por el actor que va a interactuar con la aplicación, lo dividiremos de la siguiente manera:

ID	Nombre	Descripción
CU_01	Ver carrito	Permite al usuario anónimo, autenticado y admin visualizar los productos que ha añadido a su carrito de compras.
CU_02	Buscar objetivos	Permite al usuario anónimo, autenticado y admin buscar objetivos/causas sociales en la plataforma.
CU_03	Crear cuenta	Permite al usuario anónimo registrarse y crear una nueva cuenta en la plataforma.
CU_04	Visualizar objetivo	Permite al usuario anónimo, autenticado y admin ver la información de un objetivo/causa social en la plataforma.
CU_05	Ver favoritos	Permite al usuario anónimo, autenticado y admin ver los productos que ha marcado como favoritos.
CU_06	Iniciar sesión	Permite al usuario anónimo autenticarse e iniciar sesión en su cuenta.
CU_07	Visualizar productos	Permite al usuario anónimo, autenticado y admin ver y explorar los productos disponibles en la plataforma.
CU_08	Visualizar detalles producto	Permite al usuario anónimo, autenticado y admin ver la información de un producto en la plataforma.
CU_09	Añadir al carrito	Permite al usuario anónimo, autenticado y admin añadir el producto a su carrito de compras.
CU_10	Añadir a favoritos	Permite al usuario anónimo, autenticado y admin marcar el producto como favoritos.
CU_11	Cerrar sesión	Permite al usuario autenticado y admin cerrar sesión y salir de su cuenta.
CU_12	Modificar perfil	Permite al usuario autenticado y admin editar y actualizar la información de su perfil.
CU_13	Crear objetivos	Permite al usuario autenticado y admin crear nuevos objetivos /causas sociales.
CU_14	Modificar objetivos	Permite al usuario autenticado y admin editar y actualizar los objetivos y causas sociales que ha creado.
CU_15	Eliminar objetivos	Permite al usuario autenticado y admin eliminar los objetivos/causas sociales que ha creado.
CU_16	Visualizar pedidos	Permite al usuario autenticado y admin ver el historial de sus pedidos y compras realizadas.
CU_17	Añadir productos	Permite al usuario autenticado y admin añadir nuevos productos a los objetivos/ causas sociales de la plataforma para la venta.
CU_18	Realizar pago	Permite al usuario autenticado y admin completar el proceso de pago para los productos en su carrito.
CU_19	Eliminar productos	Permite al usuario autenticado y admin eliminar productos que ha añadido a la plataforma.
CU_20	Eliminar objetivos	Permite al admin eliminar cualquiera de los objetivos/causas sociales que haya creado un usuario.

Tabla 4.20: Requisitos de usuario

### 4.3.- Requisitos funcionales

Aquí se recogerán el conjunto de requisitos funcionales de la aplicación, siendo estos todas aquellas funcionalidades y características que debe tener el sistema para satisfacer las necesidades del usuario final y cumplir con sus objetivos.

ID	Descripción de los requisitos funcionales
RF_01	El sistema debe permitir a los usuarios registrarse y crear una cuenta.
RF_02	El sistema debe permitir a los usuarios iniciar sesión con sus credenciales.
RF_03	El sistema debe permitir a los usuarios visualizar y buscar objetivos/causas sociales.
RF_04	El sistema debe permitir a los usuarios visualizar y buscar productos disponibles.
RF_05	El sistema debe permitir a los usuarios añadir productos al carrito de compras.
RF_06	El sistema debe permitir a los usuarios añadir productos a la sección de favoritos.
RF_07	El sistema debe permitir a los usuarios realizar pagos a través de una pasarela de pago.
RF_08	El sistema debe permitir a los usuarios crear y gestionar objetivos/causas sociales.
RF_09	El sistema debe permitir a los usuarios ver el progreso de las causas sociales.
RF_10	El sistema debe permitir a los usuarios visualizar su historial de compras.
RF_11	El sistema debe permitir a los usuarios editar y actualizar su perfil.
RF_12	El sistema debe permitir a los usuarios cerrar sesión de forma segura.
RF_13	El sistema debe proporcionar un sistema de filtrado y búsqueda avanzada para productos y objetivos.
RF_14	El sistema mostrará en el navbar una opción para navegar a los objetivos.
RF_15	El sistema mostrará en el navbar una opción para navegar a los pedidos.
RF_16	El sistema mostrará en el navbar una opción para navegar al perfil.
RF_17	El sistema mostrará en el navbar una opción para navegar al carrito.
RF_18	El sistema mostrará en el navbar una opción para navegar a la sección de favoritos.
RF_19	El sistema mostrará en el navbar una opción para salir de la cuenta.
RF_20	El sistema mostrará en el navbar una opción para iniciar sesión.
RF_21	El sistema mostrará en el navbar una opción para crear una cuenta.
RF_22	El sistema mostrará un formulario de registro en el que al introducir los datos correctos se creará una cuenta de usuario en la base de datos.
RF_23	El sistema no permitirá que se registre el nuevo usuario y mostrará los mensajes de error correspondientes en caso de introducir mal los datos o se intente registrar un usuario que ya existe en la base de datos.

RF_24	El sistema debe permitir a los usuarios eliminar productos del carrito.
RF_25	El sistema debe permitir a los usuarios eliminar productos de la sección de favoritos.
RF_26	El sistema debe tener una página de detalles de productos que muestre información detallada del producto seleccionado.
RF_27	El sistema validará cualquier formulario para la creación de objetivos/causas sociales.
RF_28	El sistema validará cualquier formulario para la creación de productos.
RF_29	El sistema debe permitir a los usuarios crear y gestionar productos.
RF_30	El sistema mostrará errores cuando una solicitud no se pueda realizar.
RF_31	El sistema mostrará un bento con los objetivos/causas sociales que no hayan llegado al 100% con la cantidad recaudada más elevada.
RF_32	El sistema mostrará una sección de los objetivos/causas sociales que haya creado un usuario en particular .
RF_33	El sistema debe permitir a los administradores recibir notificaciones sobre creaciones y modificaciones de objetivos.
RF_34	El sistema debe permitir a los administradores gestionar objetivos.

*Tabla 4.21: Requisitos Funcionales*

## 4.4.- Requisitos no funcionales

Aquí se recogerán el conjunto de requisitos no funcionales de la aplicación, siendo estos todas aquellas características y condiciones que debe tener el sistema para asegurar su calidad, rendimiento y usabilidad.

### 4.4.1.- Usabilidad

ID	Descripción
RNF_01	El sistema debe tener una interfaz de usuario intuitiva y fácil de usar, siguiendo las mejores prácticas de diseño UX/UI.
RNF_02	El sistema debe tener una interfaz de usuario responsive que se adapte automáticamente a diferentes tamaños de pantalla y resoluciones.
RNF_03	En la documentación se incluirán manuales de usuario que expliquen el funcionamiento de la aplicación.
RNF_04	El sistema debe proporcionar mensajes de error que sean fáciles de entender para el usuario.

Tabla 4.22: Requisitos No Funcionales- Usabilidad

### 4.4.2.- Eficiencia

ID	Descripción
RNF_05	El tiempo de respuesta del sistema debe ser inferior a 10 segundos para cualquier acción del usuario.
RNF_06	El sistema debe utilizar un mecanismo de caché para mejorar el rendimiento y reducir la carga en el servidor.
RNF_07	El sistema debe asegurar que los tiempos de carga inicial de la página no superen los 3 segundos en condiciones normales de red.

Tabla 4.23: Requisitos No Funcionales- Eficiencia

### 4.4.3.- Mantenibilidad

ID	Descripción
RNF_08	El sistema debe garantizar la consistencia de los datos en caso de fallos del sistema.
RNF_09	La aplicación será capaz de restaurarse en la corrección de errores.

Tabla 4.24: Requisitos No Funcionales- Mantenibilidad

#### 4.4.4.- Seguridad

ID	Descripción
RNF_10	El sistema debe encriptar las contraseñas de los usuarios utilizando algoritmos de encriptación seguros (bcrypt).
RNF_11	El sistema debe permitir el control de acceso granular, permitiendo definir roles y permisos detallados para diferentes tipos de usuarios.
RNF_12	El sistema debe garantizar que todas las transacciones de pago se realicen a través de canales seguros y encriptados.

Tabla 4.25: Requisitos No Funcionales- Seguridad

#### 4.4.5.- Disponibilidad

ID	Descripción
RNF_13	La disponibilidad de la aplicación será de un 99% cada vez que el usuario intente acceder.

Tabla 4.26: Requisitos No Funcionales- Disponibilidad

#### 4.4.6.- Portabilidad

ID	Descripción
RNF_14	El sistema debe ser accesible desde los principales navegadores web (Chrome, Firefox, Safari, Edge).
RNF_15	El sistema debe ser compatible con dispositivos móviles, además de ordenadores de escritorio.

Tabla 4.27: Requisitos No Funcionales- Portabilidad

#### 4.4.7.- Implementación

ID	Descripción
RNF_16	La aplicación se desarrollará utilizando el IDE de Visual Studio

Tabla 4.28: Requisitos No Funcionales- Implementación

## 4.5.- Requisitos de información

Aquí se recogerán el conjunto de requisitos de información de la aplicación, siendo estos todas aquellas necesidades de datos y estructuras de información que el sistema debe manejar para cumplir con sus objetivos.

ID	Descripción
RI_01	El sistema debe almacenar la información de los usuarios, incluyendo usuario, correo electrónico, y contraseña encriptada.
RI_02	El sistema debe registrar y mantener un historial de compras de cada usuario.
RI_03	El sistema debe almacenar información detallada de los productos, incluyendo nombre, descripción, imagen, precio, categoría y talla.
RI_04	El sistema debe registrar y gestionar los objetivos/causas sociales, incluyendo nombre de la causa, descripciones, imagen, cantidad objetivo, cantidad recaudada.
RI_05	El sistema asignará un id único a cada usuario

Tabla 4.29: Requisitos de Información

## 4.6.- Diagrama de casos de uso

En este punto, se presentan los diagramas de casos de uso que muestran las interacciones entre los diferentes actores y el sistema, destacando las principales acciones que cada tipo de usuario puede realizar.

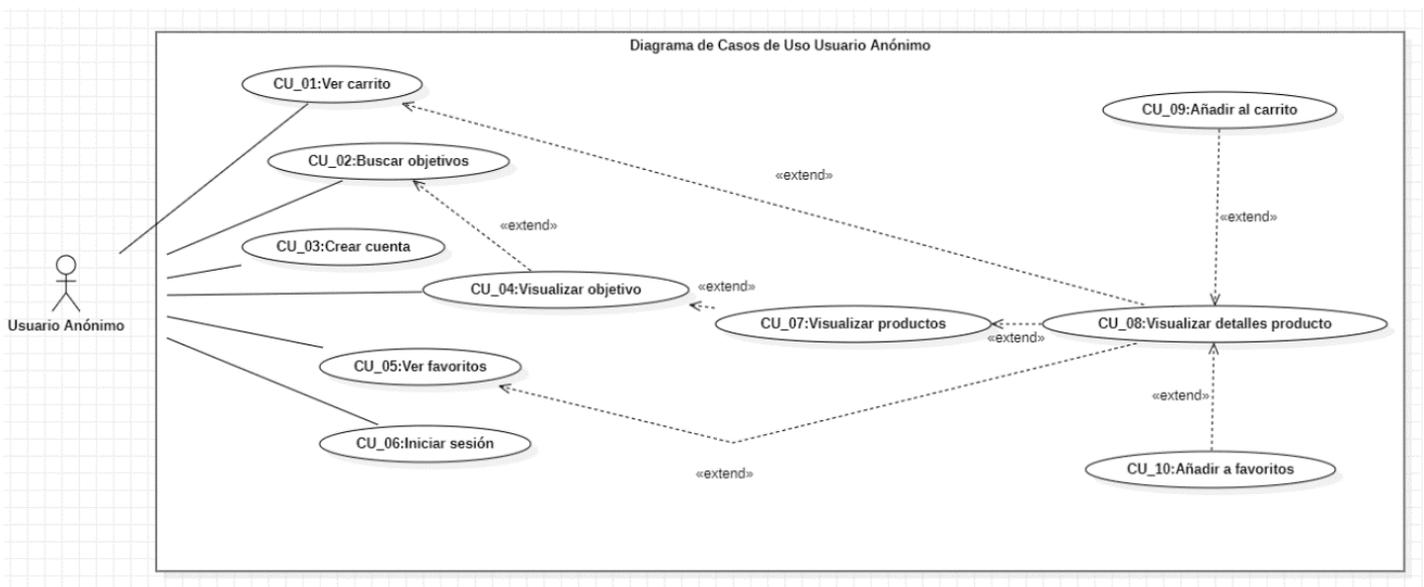


Figura 4.24: Diagrama de casos de uso de Usuario Anónimo

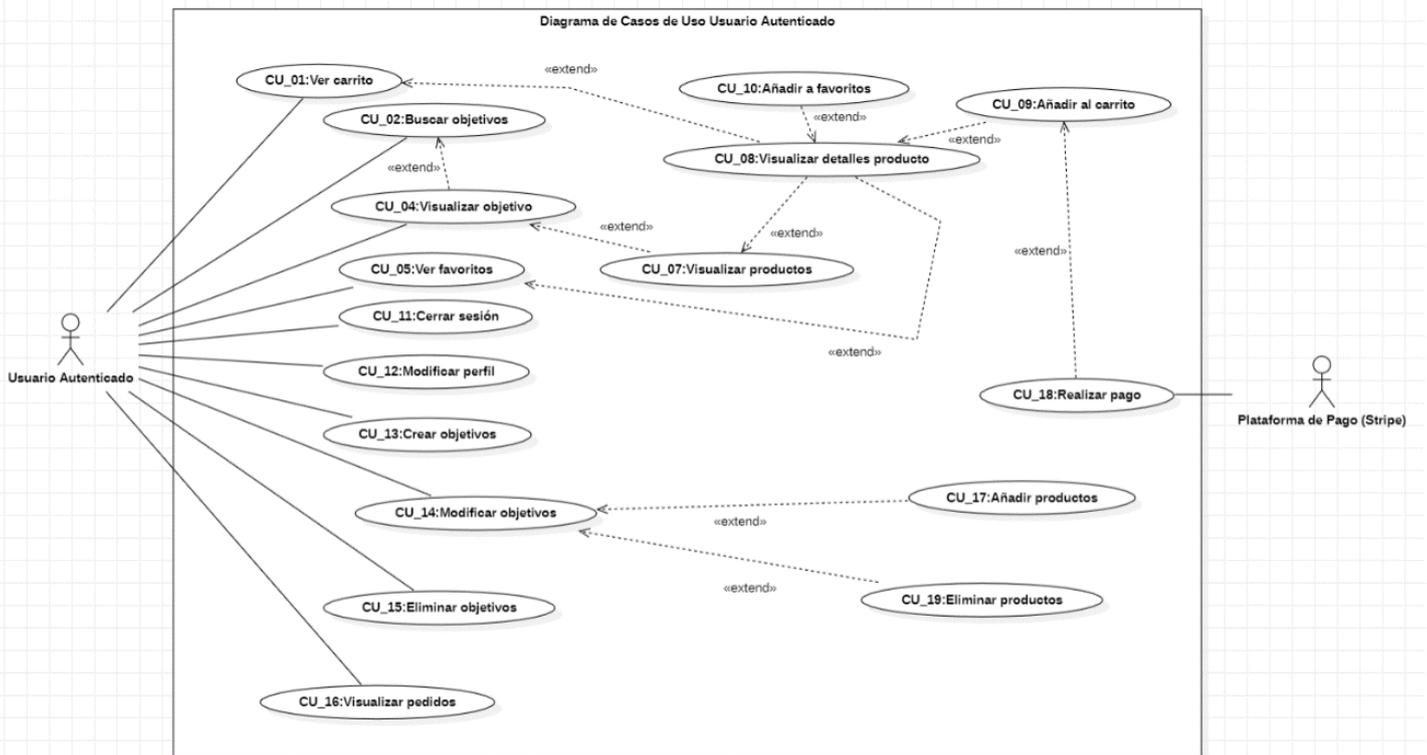


Figura 4.25: Diagrama de casos de uso de Usuario Autenticado

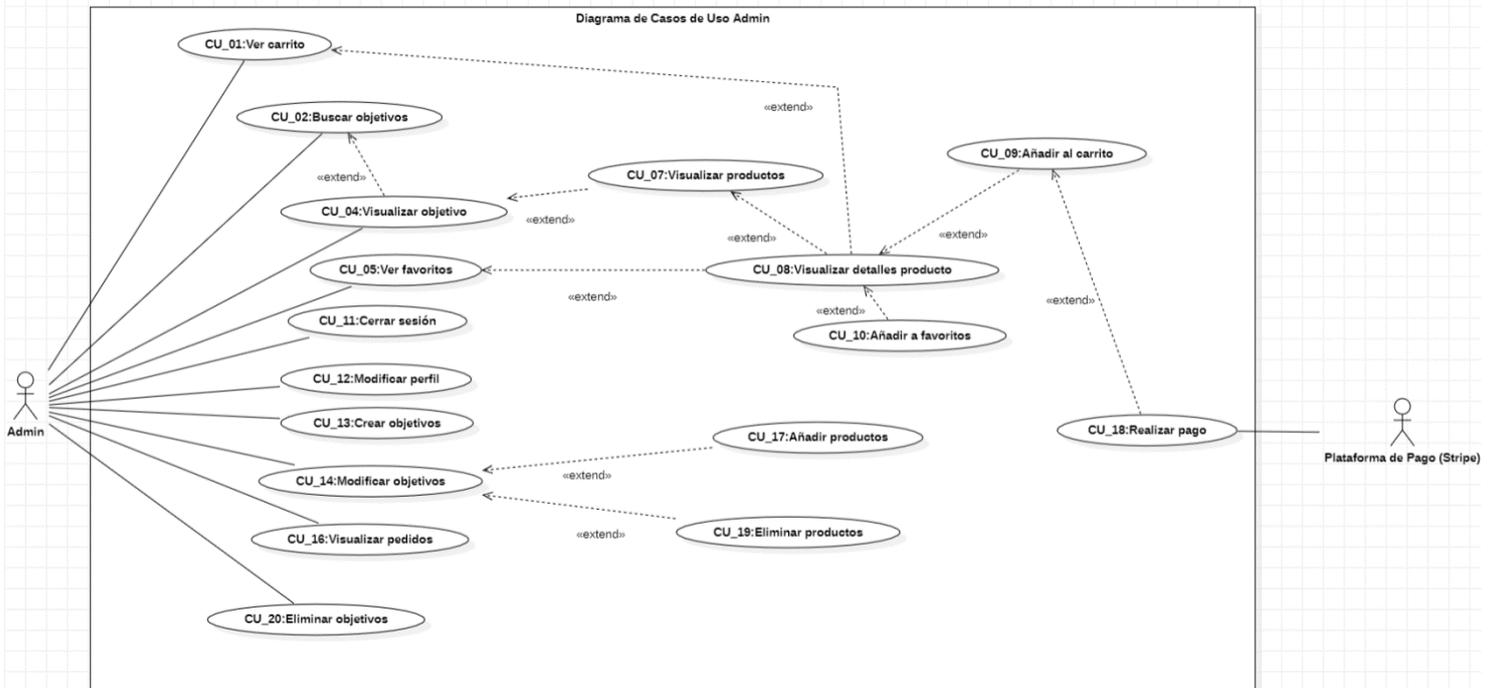


Figura 4.26: Diagrama de casos de uso de Admin

## 4.7.- Especificación de casos de uso

<b>CU_01</b>	<b>Ver carrito</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario visualizar los productos que ha añadido a su carrito de compras.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	N/A
<b>Flujo normal</b>	1. El usuario selecciona "Icono carrito". 2. El sistema muestra los productos en el carrito.
<b>Postcondiciones</b>	Los productos en el carrito se visualizan correctamente.
<b>Excepciones</b>	2a. El carrito está vacío. 1. El sistema muestra un mensaje que indica que el carrito está vacío.
<b>Importancia</b>	Alta

Tabla 4.30: Especificación de CU\_01

<b>CU_02</b>	<b>Buscar objetivos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario buscar objetivos y causas sociales en la plataforma.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El sistema debe contener objetivos
<b>Flujo normal</b>	1. El usuario selecciona objetivos 2. El usuario ingresa criterios de búsqueda. 2. El sistema muestra los resultados correspondientes.
<b>Postcondiciones</b>	Los objetivos resultantes de la búsqueda se muestran correctamente.
<b>Excepciones</b>	2a. No hay objetivos que coincidan con los criterios de la búsqueda. 1. El sistema muestra un mensaje que indica que no hay objetivos para la búsqueda.
<b>Importancia</b>	Media

Tabla 4.31: Especificación de CU\_02

<b>CU_03</b>	<b>Crear cuenta</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario registrarse y crear una nueva cuenta en la plataforma.
<b>Actores</b>	Usuario Anónimo (AC_01)
<b>Precondiciones</b>	El usuario no debe tener una cuenta existente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona Sign up.</li> <li>2. El sistema muestra un formulario.</li> <li>3. El usuario ingresa un email, un nombre de usuario, una contraseña y su confirmación de esta.</li> <li>4. El usuario pulsa botón "Registrar"</li> <li>5. El sistema valida y crea la cuenta mostrando un toast indicando que se ha registrado correctamente.</li> </ol>
<b>Postcondiciones</b>	La cuenta del usuario se crea correctamente.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3a. El correo electrónico ya está en uso. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje de error que indica que el correo electrónico introducido ya está en uso</li> <li>2. Se retoma el paso 1 del flujo normal.</li> </ol> </li> <li>3b. Los datos proporcionados son inválidos. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje de error que indica que los campos introducidos son incorrectos.</li> <li>2. Se retoma el paso 1 del flujo normal.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.32: Especificación de CU\_03

<b>CU_04</b>	<b>Visualizar objetivo</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario ver la información de un objetivo/causa social en la plataforma.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	Debe haber objetivos o causas sociales en el sistema.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción objetivos.</li> <li>2. El sistema muestra los objetivos disponibles.</li> <li>3. El usuario selecciona un objetivo.</li> <li>4. El sistema muestra la información correspondiente al objetivo.</li> </ol>
<b>Postcondiciones</b>	La información del objetivo se muestra correctamente al usuario.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2a. No hay objetivos disponibles. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje que indica que no hay objetivos disponibles.</li> </ol> </li> <li>4a. Url de objetivo no existente. <ol style="list-style-type: none"> <li>1. El sistema muestra un error que indica que no se han podido recuperar los detalles acerca del objetivo con una imagen correspondiente.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.33: Especificación de CU\_04

<b>CU_05</b>	<b>Ver favoritos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario ver los productos que ha marcado como favoritos.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	N/A
<b>Flujo normal</b>	1. El usuario pulsa "Icono corazón". 2. El sistema muestra los productos favoritos del usuario.
<b>Postcondiciones</b>	Los productos favoritos se visualizan correctamente.
<b>Excepciones</b>	2a. La sección de favoritos está vacía. 1. El sistema muestra un mensaje que indica que no hay productos en la sección de favoritos.
<b>Importancia</b>	Media

Tabla 4.34: Especificación de CU\_05

<b>CU_06</b>	<b>Iniciar sesión</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario autenticarse e iniciar sesión en su cuenta.
<b>Actores</b>	Usuario Anónimo (AC_01)
<b>Precondiciones</b>	El usuario debe tener una cuenta registrada.
<b>Flujo normal</b>	1. El usuario selecciona el icono de la persona. 2. El usuario ingresa su correo y contraseña. 3. El usuario pulsa botón "Iniciar sesión". 4. El sistema valida y permite el acceso mostrando un toast de inicio de sesión correcto.
<b>Postcondiciones</b>	El usuario inicia sesión y accede a su cuenta.
<b>Excepciones</b>	2a. Credenciales incorrectas. 1. El sistema muestra un mensaje de error 2. Se retoma el paso 2 del flujo normal. 2b. Cuenta no verificada. 1. El sistema muestra un mensaje que indica que el Usuario no existe 2. Se retoma el paso 2 del flujo normal.
<b>Importancia</b>	Alta

Tabla 4.35: Especificación de CU\_06

<b>CU_07</b>	<b>Visualizar productos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario ver y explorar los productos disponibles en la plataforma.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	Los productos deben estar disponibles en la plataforma.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Objetivos".</li> <li>2. El usuario selecciona un objetivo.</li> <li>3. El sistema muestra los productos disponibles.</li> </ol>
<b>Postcondiciones</b>	Los productos se muestran correctamente al usuario.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2a. No hay productos disponibles. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje que indica que no hay productos disponibles.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.36: Especificación de CU\_07

<b>CU_08</b>	<b>Visualizar detalles producto</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario ver la información de un producto en la plataforma.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El producto debe existir.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Objetivos".</li> <li>2. El usuario selecciona un objetivo.</li> <li>3. El sistema muestra los productos disponibles.</li> <li>4. El usuario selecciona un producto.</li> <li>5. El sistema muestra la información del producto.</li> </ol>
<b>Postcondiciones</b>	Se muestra la información del producto correctamente al usuario.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>3a. No hay productos disponibles. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje que indica que no hay productos disponibles.</li> </ol> </li> <li>4a. Url de producto no existente. <ol style="list-style-type: none"> <li>2. El sistema muestra un error que indica que no se han podido recuperar los detalles acerca del producto con una imagen correspondiente.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.37: Especificación de CU\_08

<b>CU_09</b>	<b>Añadir al carrito</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario añadir el producto a su carrito de compras.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	Debe haber productos disponibles.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Objetivos".</li> <li>2. El usuario selecciona un objetivo.</li> <li>3. El sistema muestra los productos disponibles.</li> <li>4. El usuario selecciona un producto.</li> <li>5. El sistema muestra la información del producto.</li> <li>6. El usuario pulsa el botón de añadir al carrito.</li> <li>7. El sistema añade el producto al carrito mostrando un toast indicando que ha sido añadido.</li> </ol>
<b>Postcondiciones</b>	Los productos se añaden al carrito correctamente.
<b>Excepciones</b>	6a. El producto ya existe en el carrito. <ol style="list-style-type: none"> <li>1. El sistema muestra un toast que indica que el producto ya existe en el carrito</li> <li>2. Se retoma el paso 5 del flujo normal.</li> </ol>
<b>Importancia</b>	Alta

Tabla 4.38: Especificación de CU\_09

<b>CU_10</b>	<b>Añadir a favoritos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario marcar el producto como favoritos.
<b>Actores</b>	Usuario Anónimo (AC_01), Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	Debe haber productos disponibles.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Objetivos".</li> <li>2. El usuario selecciona un objetivo.</li> <li>3. El sistema muestra los productos disponibles.</li> <li>4. El usuario selecciona un producto.</li> <li>5. El sistema muestra la información del producto.</li> <li>6. El usuario pulsa el botón de añadir al carrito.</li> <li>7. El sistema añade el producto al carrito mostrando un toast indicando que ha sido añadido.</li> </ol>
<b>Postcondiciones</b>	Los productos se añaden a la sección favoritos correctamente.
<b>Excepciones</b>	6a. El producto ya existe en la sección favoritos. <ol style="list-style-type: none"> <li>1. El sistema muestra un toast que indica que el producto ya existe en la sección favoritos.</li> <li>2. Se retoma el paso 5 del flujo normal.</li> </ol>
<b>Importancia</b>	Media

Tabla 4.39: Especificación de CU\_10

<b>CU_11</b>	<b>Cerrar sesión</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario cerrar sesión y salir de su cuenta.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión.
<b>Flujo normal</b>	1. El usuario pulsa el botón "Icono de la puerta". 2. El sistema cierra la sesión del usuario.
<b>Postcondiciones</b>	El usuario se desloguea correctamente.
<b>Excepciones</b>	N/A
<b>Importancia</b>	Baja

Tabla 4.40: Especificación de CU\_11

<b>CU_12</b>	<b>Modificar perfil</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario editar y actualizar la información de su perfil.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión.
<b>Flujo normal</b>	1. El usuario selecciona "Icono persona". 2. El usuario selecciona editar perfil 3. El usuario inserta los nuevos datos en el formulario. 4. El usuario pulsa el botón "Guardar cambios" 5. El sistema modifica los datos en la base de datos y muestra un toast que indica que el perfil se ha modificado.
<b>Postcondiciones</b>	El perfil del usuario se actualiza correctamente.
<b>Excepciones</b>	3a. Datos inválidos o incompletos. <ul style="list-style-type: none"> <li>1. El sistema muestra un mensaje que indica que los datos introducidos son inválidos o vacíos.</li> <li>2. Se retoma paso 3 de flujo normal.</li> </ul>
<b>Importancia</b>	Alta

Tabla 4.41: Especificación de CU\_12

<b>CU_13</b>	<b>Crear objetivos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario crear nuevos objetivos/causas sociales.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Objetivos".</li> <li>2. El usuario pulsa botón "Crear objetivo".</li> <li>3. El sistema muestra formulario de registro de objetivos.</li> <li>4. El usuario introduce los detalles del objetivo.</li> <li>5. El usuario pulsa botón "Crear objetivo".</li> <li>6. El sistema crea el objetivo y muestra un toast indicando que el objetivo se ha creado correctamente.</li> </ol>
<b>Postcondiciones</b>	El objetivo se crea correctamente y se muestra en la plataforma.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>4a. Datos inválidos o incompletos. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje de error que indica que los datos son inválidos o vacíos.</li> <li>2. Se retoma paso 4 de flujo normal.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.42: Especificación de CU\_13

<b>CU_14</b>	<b>Modificar objetivos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario editar y actualizar un objetivo/causa social que el usuario haya creado.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y haber creado objetivos previamente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selección "Objetivos".</li> <li>2. El usuario pulsa botón "Mis objetivos".</li> <li>3. El usuario selecciona un objetivo existente y pulsa botón "Modificar".</li> <li>4. El usuario modifica los detalles.</li> <li>5. El usuario pulsa botón "Guardar cambios".</li> <li>6. El sistema guarda los cambios del objetivo y muestra un toast indicando que se ha modificado el objetivo.</li> </ol>
<b>Postcondiciones</b>	El objetivo se actualiza correctamente en la plataforma.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>4a. Datos inválidos o incompletos. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje de error que indica que los datos son inválidos o vacíos.</li> <li>2. Se retoma paso 4 de flujo normal.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.43: Especificación de CU\_14

<b>CU_15</b>	<b>Eliminar objetivos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario eliminar un objetivo/causa social que el usuario haya creado.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y haber creado objetivos previamente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selección "Objetivos".</li> <li>2. El usuario pulsa botón "Mis objetivos".</li> <li>3. El usuario selecciona un objetivo existente y pulsa botón "Eliminar".</li> <li>4. El sistema elimina el objetivo y muestra un toast indicando que se ha eliminado el objetivo.</li> </ol>
<b>Postcondiciones</b>	El objetivo se elimina correctamente de la plataforma.
<b>Excepciones</b>	N/A
<b>Importancia</b>	Alta

Tabla 4.44: Especificación de CU\_15

<b>CU_16</b>	<b>Visualizar pedidos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario ver el historial de sus pedidos y compras realizadas.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y realizado compras previamente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Pedidos".</li> <li>2. El sistema muestra el historial de pedidos.</li> </ol>
<b>Postcondiciones</b>	El historial de pedidos se muestra correctamente.
<b>Excepciones</b>	<p>2a. No hay pedidos en el historial.</p> <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje indicando que no hay pedidos realizados.</li> </ol>
<b>Importancia</b>	Alta

Tabla 4.45: Especificación de CU\_16

<b>CU_17</b>	<b>Añadir productos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario añadir nuevos productos a los objetivos/ causas sociales de la plataforma para la venta.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y debe existir un producto.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selección "Objetivos".</li> <li>2. El usuario pulsa botón "Mis objetivos".</li> <li>3. El usuario selecciona un objetivo existente y pulsa botón "Modificar".</li> <li>4. El usuario pulsa botón "Añadir producto".</li> <li>5. El sistema muestra un formulario de registro de un producto.</li> <li>6. El usuario introduce los datos del producto.</li> <li>7. El usuario pulsa botón "Añadir producto".</li> <li>8. El sistema añade el producto al objetivo y muestra un toast indicando que se ha añadido el producto.</li> </ol>
<b>Postcondiciones</b>	El producto se añade correctamente al objetivo en la plataforma.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>6a. Datos inválidos o incompletos. <ol style="list-style-type: none"> <li>1. El sistema muestra un mensaje de error que indica que los datos son inválidos o vacíos.</li> <li>2. Se retoma paso 6 de flujo normal.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.46: Especificación de CU\_17

<b>CU_18</b>	<b>Realizar pago</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario completar el proceso de pago para los productos en su carrito.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y tener productos en el carrito.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Icono carrito".</li> <li>2. El sistema muestra los productos en el carrito.</li> <li>3. El usuario pulsa botón "Comprar".</li> <li>4. El sistema redirige a la pasarela de pago.</li> </ol>
<b>Postcondiciones</b>	El pago se realiza correctamente y se confirma la compra.
<b>Excepciones</b>	<ol style="list-style-type: none"> <li>2a. El carrito está vacío. <ol style="list-style-type: none"> <li>1. La redirección del botón Comprar estará deshabilitada.</li> </ol> </li> </ol>
<b>Importancia</b>	Alta

Tabla 4.47: Especificación de CU\_18

<b>CU_19</b>	<b>Eliminar productos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al usuario eliminar productos de un objetivo/ causa social de la plataforma.
<b>Actores</b>	Usuario Autenticado (AC_02), Admin (AC_03)
<b>Precondiciones</b>	El usuario debe haber iniciado sesión y haber añadido productos a un objetivo previamente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El usuario selección "Objetivos".</li> <li>2. El usuario pulsa botón "Mis objetivos".</li> <li>3. El usuario selecciona un objetivo existente y pulsa botón "Modificar".</li> <li>4. El usuario pulsa botón "Eliminar producto".</li> <li>5. El sistema muestra una lista de productos que contiene el objetivo.</li> <li>6. El usuario pulsa botón "Eliminar".</li> <li>7. El sistema elimina el producto del objetivo y muestra un toast indicando que se ha eliminado el producto.</li> </ol>
<b>Postcondiciones</b>	El producto se elimina correctamente del objetivo de la plataforma.
<b>Excepciones</b>	N/A
<b>Importancia</b>	Alta

Tabla 4.48: Especificación de CU\_19

<b>CU_20</b>	<b>Eliminar objetivos</b>
<b>Versión</b>	1.0
<b>Autor</b>	Sergio Fagúndez Manso
<b>Descripción</b>	Permite al admin eliminar un objetivo/causa social que un usuario haya creado.
<b>Actores</b>	Admin (AC_03)
<b>Precondiciones</b>	El admin debe haber iniciado sesión y un usuario debe haber creado objetivos previamente.
<b>Flujo normal</b>	<ol style="list-style-type: none"> <li>1. El admin selecciona "Admin".</li> <li>2. El admin selecciona un objetivo existente y pulsa botón "Eliminar".</li> <li>3. El sistema elimina el objetivo y muestra un toast indicando que se ha eliminado el objetivo.</li> </ol>
<b>Postcondiciones</b>	El objetivo se elimina correctamente de la plataforma.
<b>Excepciones</b>	N/A
<b>Importancia</b>	Alta

Tabla 4.49: Especificación de CU\_20



# Capítulo 5

## Diseño

En esta sección del documento se presentarán los aspectos más importantes del diseño de la aplicación, incluyendo las diferentes arquitecturas utilizadas, el modelo E-R con el diccionario de datos y el modelo lógico de datos documentado.

### 5.1.1.- Arquitectura lógica no detallado

La arquitectura lógica representa la estructura conceptual del sistema, incluyendo la organización de los componentes de software y su interacción. Esta arquitectura define cómo se distribuyen las funciones y responsabilidades dentro de la aplicación, como la separación entre la capa de presentación, la capa de negocio y la capa de datos. A través de diagramas y modelos, se ilustra cómo los módulos y servicios se comunican entre sí, facilitando la comprensión del flujo de información y las dependencias entre componentes.

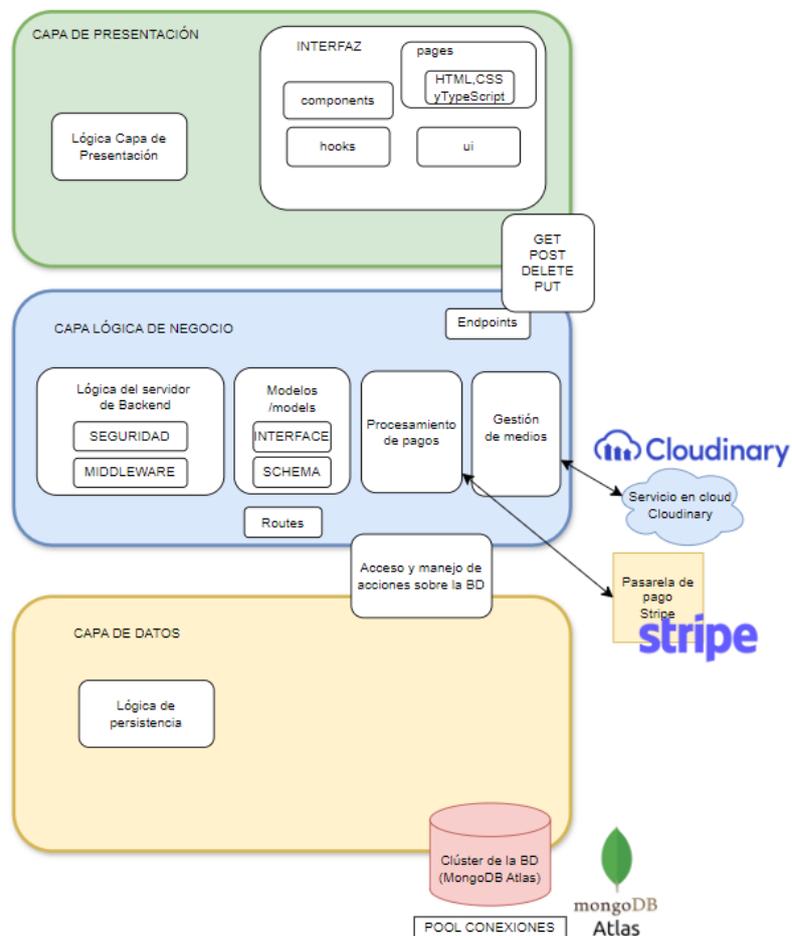


Figura 5.27: Diseño Lógico No detallado

## 5.1.2.- Arquitectura lógica detallado

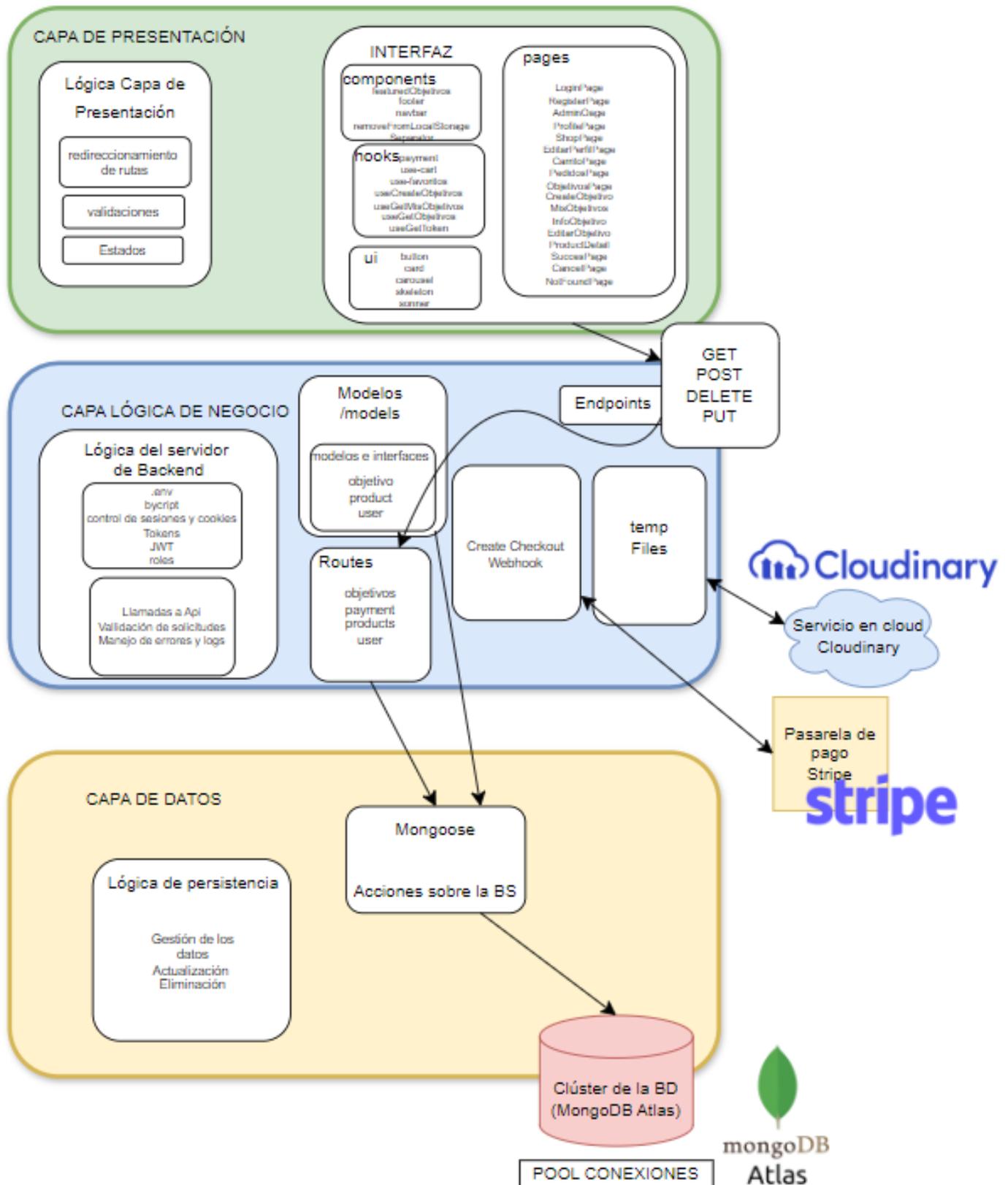


Figura 5.28: Diseño Lógico Detallado

## 5.2.- Arquitectura física

La arquitectura física representa la disposición concreta del hardware y la red que soportan la aplicación. Incluye detalles sobre los servidores, dispositivos de almacenamiento, y otros componentes de infraestructura que se utilizan para ejecutar y mantener el sistema. Esta arquitectura define cómo los distintos elementos se conectan entre sí, así como las configuraciones específicas para asegurar un rendimiento óptimo y una alta disponibilidad.

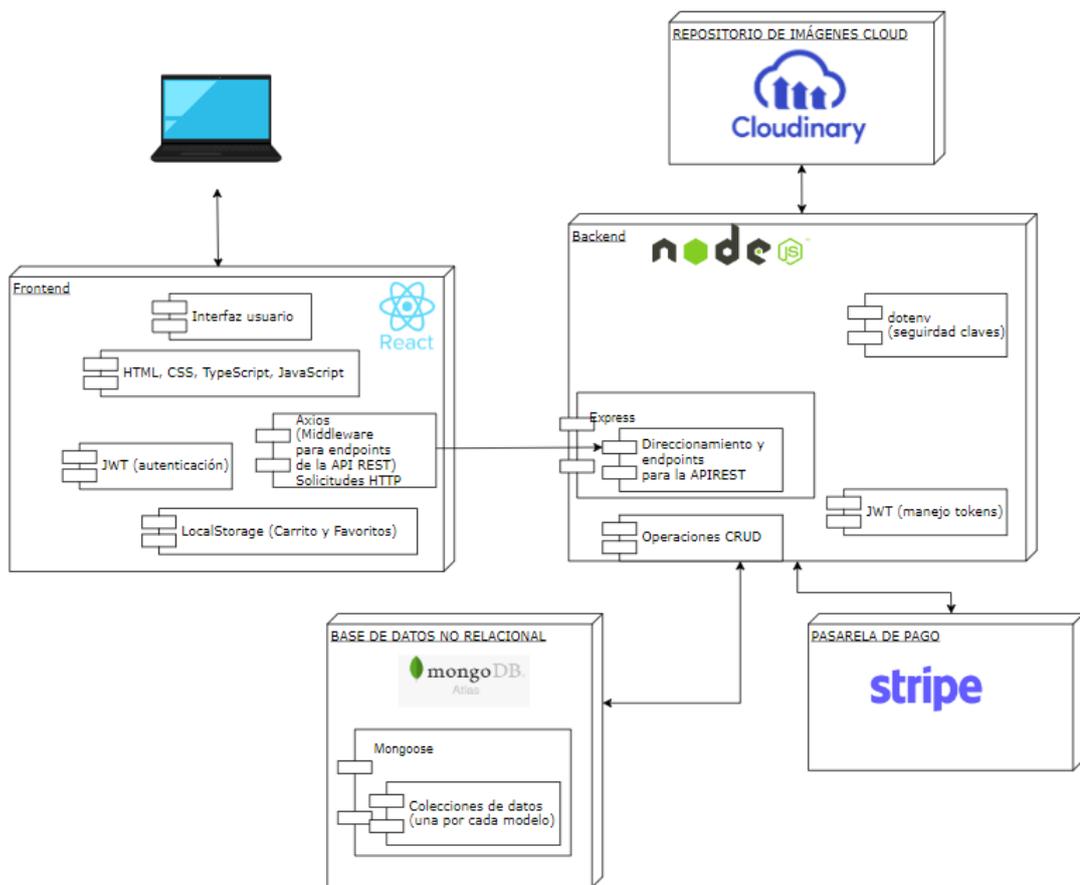


Figura 5.29: Diseño Arquitectura Física

## 5.3.- Modelo E-R

El modelo entidad-relación (E-R) es una herramienta lógica que permite representar los datos de manera sencilla y comprensible. Su principal función es representar de forma gráfica los datos diseñando un modelo conceptual, lo que facilita visualizar cómo interactúan y se relacionan los diferentes elementos entre sí.

Aquí, se presenta el modelo Entidad-Relación que abarcará el proyecto.

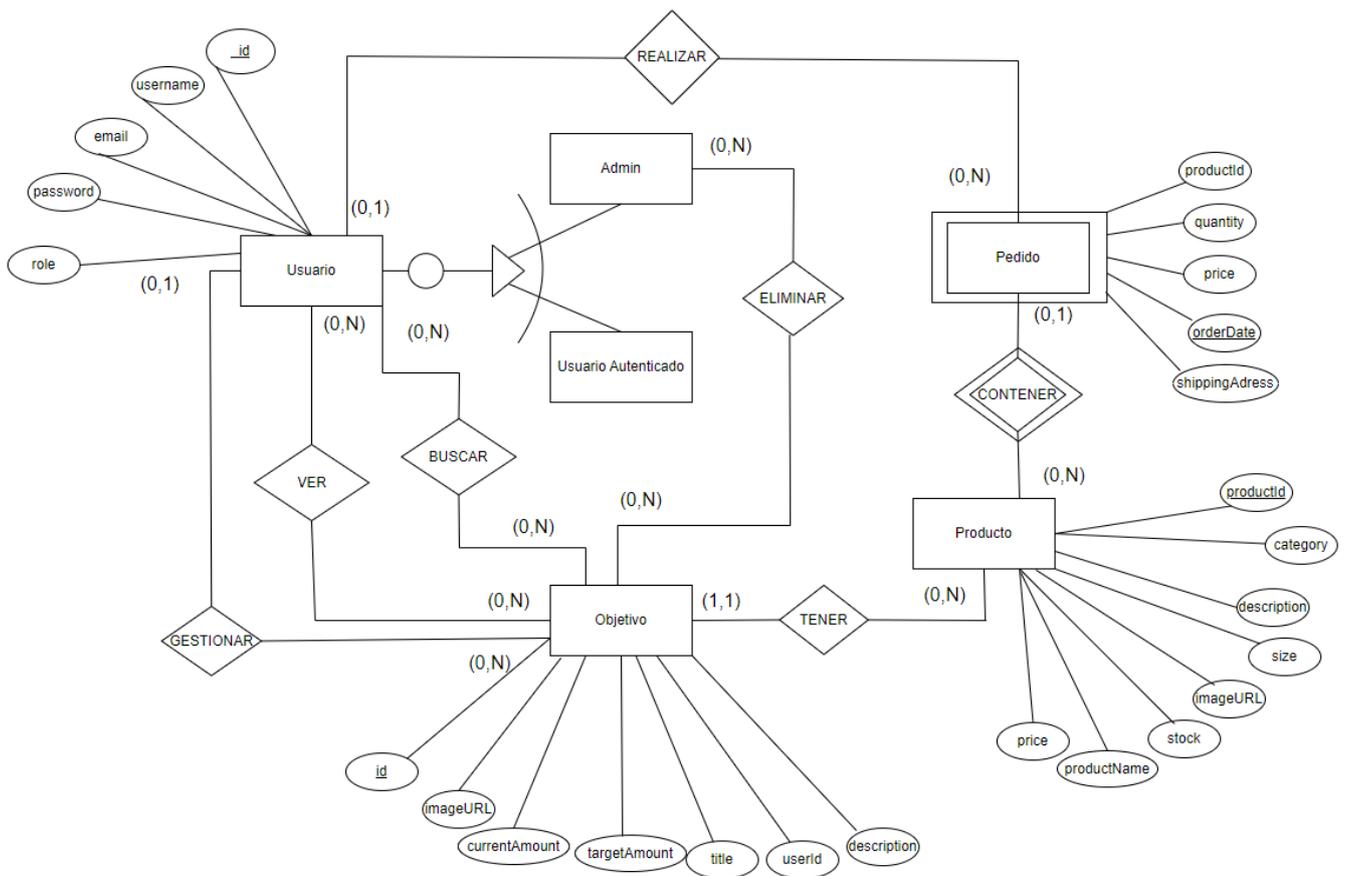


Figura 5.30: Modelo E-R

## 5.4.- Diccionario de datos

El diccionario de datos representa una recopilación detallada de todos los elementos de datos utilizados en la aplicación, describiendo cada uno de ellos y especificando sus atributos, tipos de datos. Incluye restricciones, campos, proporcionando una referencia para cualquier usuario que quiera entender el modelo E-R de nuestro sistema.

Entidad Usuario			
Atributos	Tipo	Único (unique)	Required
<i>_id</i>	String	Sí	Sí
username	String	Sí	Sí
email	String	Sí	Sí
password	String	No	Sí
role	String	No	Sí

Tabla 5.50: Diccionario de datos Usuario

Entidad Objetivo			
Atributos	Tipo	Único (unique)	Required
id	String	Sí	Sí
userid	String	Sí	Sí
title	String	No	Sí
description	String	No	Sí
imageURL	String	No	Sí
targetAmount	Number	No	Sí
currentAmount	Number	No	Sí

Tabla 5.51: Diccionario de datos Objetivo

Entidad Producto			
Atributos	Tipo	Único (unique)	Required
productId	String	Sí	Sí
productName	String	No	Sí
price	Number	No	Sí
description	String	No	Sí
imageURL	String	No	Sí
stockQuantity	Number	No	Sí
category	Enum	No	Sí
size	String	No	Sí

Tabla 5.52: Diccionario de datos Producto

Entidad Pedido			
Atributos	Tipo	Único (unique)	Required
orderDate	Date	Sí	Sí
shippingAdress	String	No	Sí
price	Number	No	Sí
productId	String	Sí	Sí
quantity	Number	No	Sí

Tabla 5.53: Diccionario de datos Producto

## 5.5.- Modelo lógico de datos

El modelo lógico de datos representa una abstracción del diseño de la base de datos, mostrando cómo se organizan y relacionan los datos sin entrar en detalles físicos. El modelo lógico de datos es crucial para asegurar que todos los requisitos de datos se capturen correctamente y que la base de datos sea coherente y eficiente.

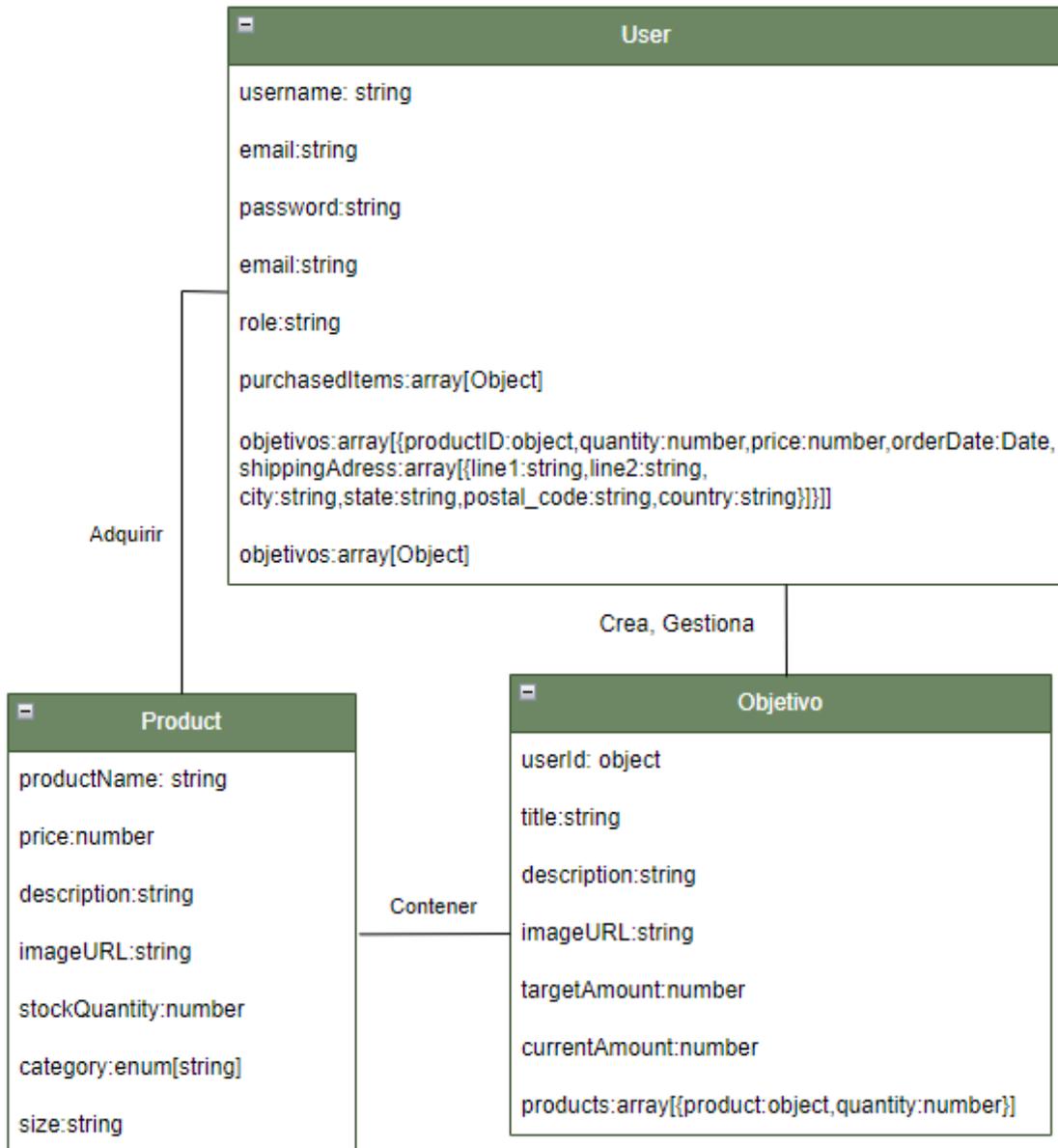


Figura 5.31: Modelo Lógico de Datos

## 5.6.- Diagrama de secuencia

El diagrama de secuencia representa una visión detallada de cómo interactúan los diferentes componentes del sistema a lo largo del tiempo para llevar a cabo un proceso específico. Este diagrama muestra los objetos involucrados, los mensajes que se envían entre ellos y el orden en que ocurren estas interacciones. Es una herramienta crucial para entender el flujo dinámico de control y datos en el sistema, permitiendo identificar posibles puntos de fallo y optimizar la eficiencia del proceso.

A continuación, se muestra el diagrama de secuencia para que un usuario anónimo pueda iniciar sesión en la aplicación.

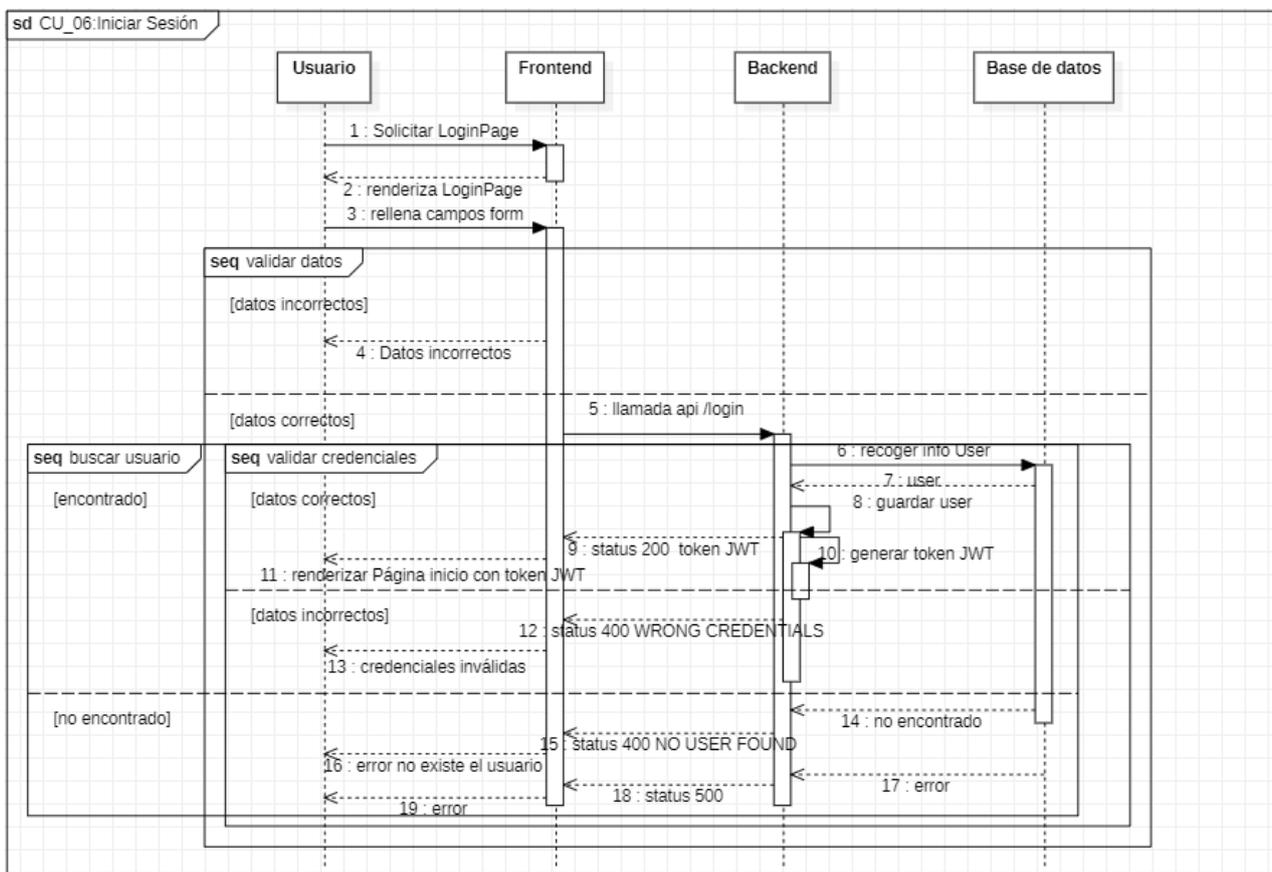


Figura 5.32: Diagrama de secuencia CU\_06: Iniciar sesión

A continuación, se muestra el diagrama de secuencia para que un usuario autenticado o admin cree un objetivo.

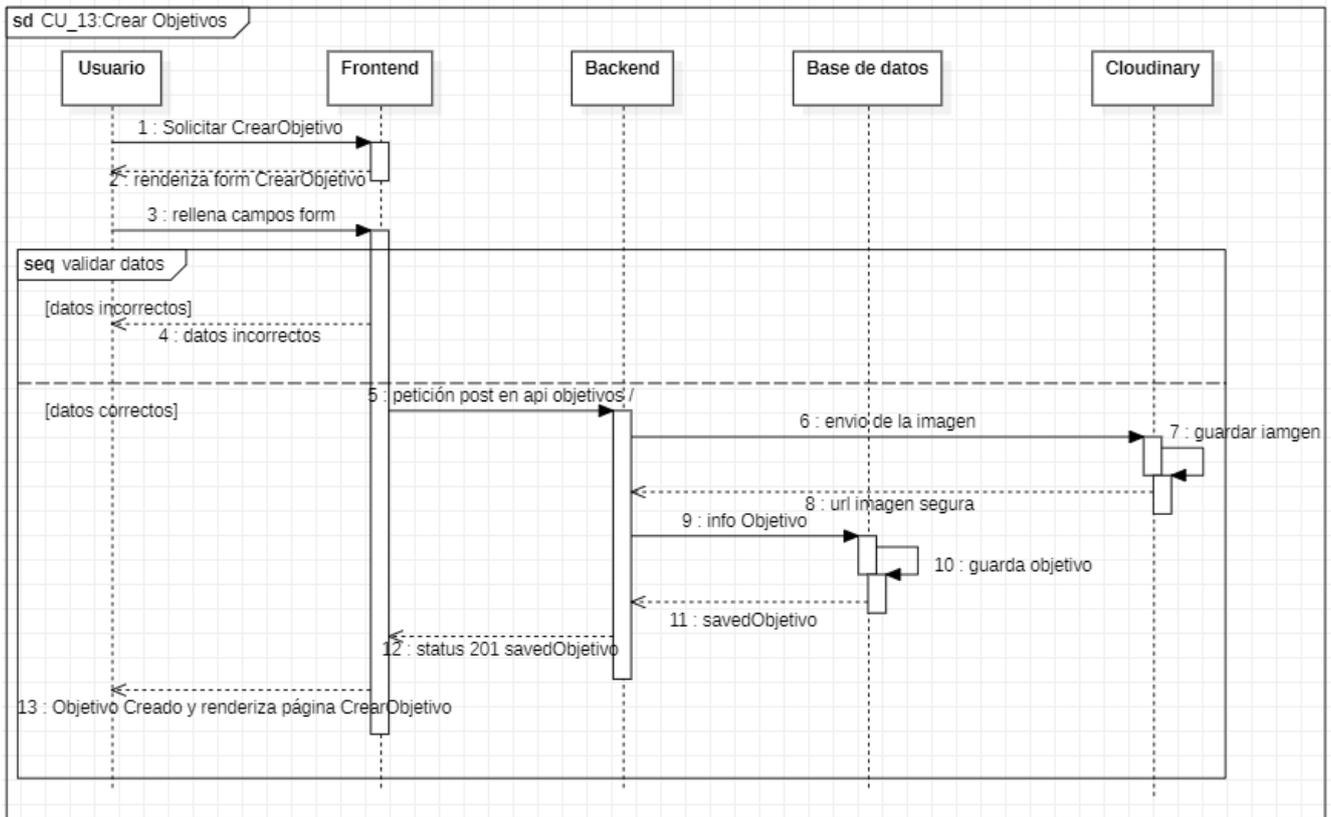


Figura 5.33: Diagrama de secuencia CU\_13: Crear Objetivos

## 5.7.- Diseño de interfaz

En este apartado se describirán las vistas que se mostrarán al usuario de la aplicación. El diseño de interfaz abarca tanto la disposición visual de los elementos como la usabilidad y la experiencia del usuario. Se detallarán los componentes de cada pantalla, como botones, formularios, menús y otros elementos interactivos, asegurando que la navegación sea intuitiva y eficiente.

<b>Nombre</b>	<b>Vista Principal</b>
<b>Descripción</b>	Ventana principal que muestra los objetivos destacados con sus gráficos
<b>Activación</b>	Al iniciar la aplicación
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos el objetivo o el gráfico nos lleva al objetivo y nos muestra toda la información correspondiente.</li> </ol>

Tabla 5.54: Diseño Interfaz Vista Principal

<b>Nombre</b>	<b>Vista Registro</b>
<b>Descripción</b>	Ventana para que el usuario se pueda registrar
<b>Activación</b>	Al pulsar el botón de Sign up
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos el botón una vez introducida la información, nos crea la cuenta.</li> </ol>

Tabla 5.55: Diseño Interfaz Vista Registro

<b>Nombre</b>	<b>Vista Iniciar Sesión</b>
<b>Descripción</b>	Ventana para que el usuario se pueda autenticar
<b>Activación</b>	Al pulsar el Icono de la persona
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos el botón una vez introducida la información, nos valida los datos y nos autentifica.</li> </ol>

Tabla 5.56: Diseño Interfaz Vista Iniciar Sesión

<b>Nombre</b>	<b>Vista Buscar Objetivos</b>
<b>Descripción</b>	Ventana para que el usuario pueda buscar objetivos
<b>Activación</b>	Al pulsar el botón Objetivos
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si escribimos en la barra de búsqueda un objetivo nos muestra los resultados.</li> <li>2. Si seleccionamos una categoría nos muestra los objetivos que contengan esa categoría.</li> </ol>

Tabla 5.57: Diseño Interfaz Vista Buscar Objetivos

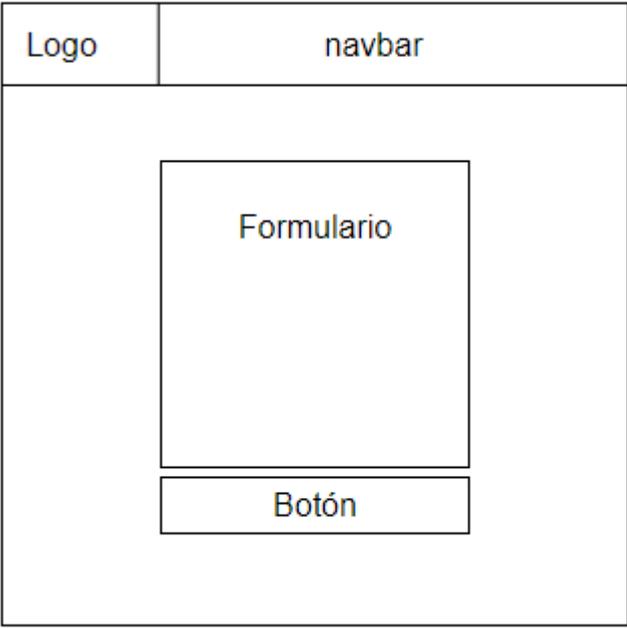
<b>Nombre</b>	<b>Vista Crear Objetivo</b>
<b>Descripción</b>	Ventana para que el usuario autenticado pueda crear un objetivo
<b>Activación</b>	Al pulsar el botón Crear Objetivo
<b>Boceto</b>	
<b>Eventos</b>	1. Si pulsamos el botón una vez introducida la información, nos valida los datos y nos crea el objetivo.

Tabla 5.58: Diseño Interfaz Vista Crear Objetivo

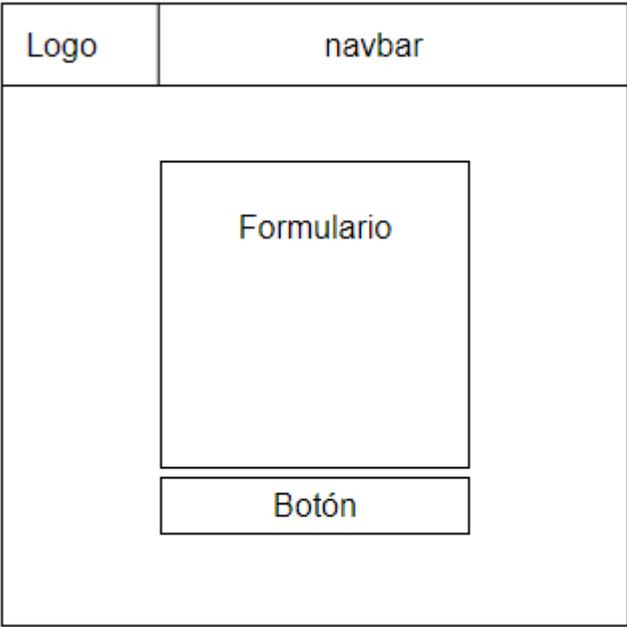
<b>Nombre</b>	<b>Vista Añadir Producto</b>
<b>Descripción</b>	Ventana para que el usuario autenticado pueda añadir un producto a un objetivo
<b>Activación</b>	Al pulsar el botón Añadir Producto
<b>Boceto</b>	
<b>Eventos</b>	1. Si pulsamos el botón una vez introducida la información, nos valida los datos y nos añade el producto al objetivo.

Tabla 5.59: Diseño Interfaz Vista Añadir Producto

<b>Nombre</b>	<b>Vista Navbar</b>
<b>Descripción</b>	Barra de navegación que se mostrará en la cabecera de la aplicación
<b>Activación</b>	
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos el botón del logo nos lleva a la página principal</li> <li>2. Si pulsamos botón Pedidos nos lleva a la página de historial de pedidos</li> <li>3. Si pulsamos botón Objetivos nos lleva a página objetivos</li> <li>4. Si pulsamos icono carrito nos lleva a la página del carrito de la compra donde están los productos añadidos al carrito</li> <li>5. Si pulsamos botón de Icono Favoritos nos lleva a la sección de los productos favoritos</li> <li>6. Si pulsamos icono persona nos lleva al perfil o al iniciar sesión dependiendo de si estamos autenticados</li> <li>7. Si pulsamos icono de Sign Up nos lleva a la vista de Crear cuenta</li> <li>8. Si pulsamos Icono Puerta nos cierra la sesión</li> <li>9. Si pulsamos icono Luna nos cambia el tema del Navbar a modo oscuro o claro</li> </ol>

Tabla 5.60: Diseño Interfaz Vista Navbar

<b>Nombre</b>	<b>Vista Mis Objetivos</b>
<b>Descripción</b>	Ventana que mostrará al usuario los objetivos que ha creado
<b>Activación</b>	Pulsar botón Mis Objetivos
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos sobre un objetivo nos lleva a la página del objetivo</li> <li>2. Si pulsamos en modificar, modificaremos el objetivo</li> <li>3. Si pulsamos en eliminar, eliminaremos el objetivo</li> </ol>

Tabla 5.61: Diseño Interfaz Vista Mis Objetivos

<b>Nombre</b>	<b>Vista Objetivo</b>
<b>Descripción</b>	Ventana que mostrará al usuario la información de un objetivo específico
<b>Activación</b>	Pulsar sobre un objetivo
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos sobre un objetivo nos lleva a la página del objetivo</li> <li>2. Si pulsamos en modificar, modificaremos el objetivo</li> <li>3. Si pulsamos en eliminar, eliminaremos el objetivo</li> </ol>

Tabla 5.62: Diseño Interfaz Vista Objetivo

<b>Nombre</b>	<b>Vista Producto</b>
<b>Descripción</b>	Ventana que mostrará al usuario la información sobre un producto
<b>Activación</b>	Pulsar sobre un producto
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos añadir al carrito, nos lo añade al carrito</li> <li>2. Si pulsamos añadir a favoritos, nos lo añade a favoritos</li> </ol>

Tabla 5.63: Diseño Interfaz Vista Producto

<b>Nombre</b>	<b>Vista Carrito</b>
<b>Descripción</b>	Ventana que mostrará al usuario los productos que estén añadidos al carrito
<b>Activación</b>	Pulsar icono del carrito
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos a comprar, nos deriva a la pasarela de pago</li> <li>2. Si pulsamos botón “X” nos elimina el producto del carrito</li> <li>3. <u>Si pulsamos la imagen del producto nos redirige a él</u></li> </ol>

Tabla 5.64: Diseño Interfaz Vista Carrito

<b>Nombre</b>	<b>Vista Favoritos</b>
<b>Descripción</b>	Ventana que mostrará al usuario los productos que estén añadidos a favoritos
<b>Activación</b>	Pulsar icono de favoritos
<b>Boceto</b>	
<b>Eventos</b>	<ol style="list-style-type: none"> <li>1. Si pulsamos botón “X” nos elimina el producto del carrito</li> <li>2. <u>Si pulsamos la imagen del producto nos redirige a él</u></li> </ol>

Tabla 5.65: Diseño Interfaz Vista Favoritos

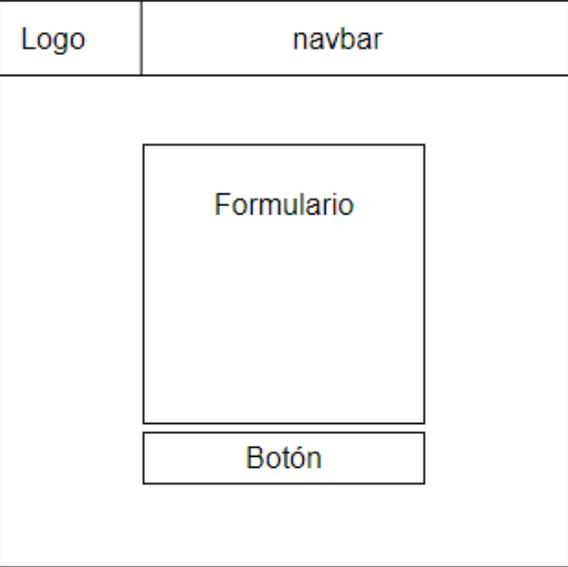
<b>Nombre</b>	<b>Vista Modificar Perfil</b>
<b>Descripción</b>	Ventana que muestra un formulario para que el usuario modifique los datos de su perfil
<b>Activación</b>	Al pulsar el icono de la persona y posteriormente editar perfil
<b>Boceto</b>	
<b>Eventos</b>	1. Si pulsamos el botón una vez introducida la información, nos valida los datos y nos modifica los datos del perfil

Tabla 5.66: Diseño Interfaz Vista Modificar Perfil

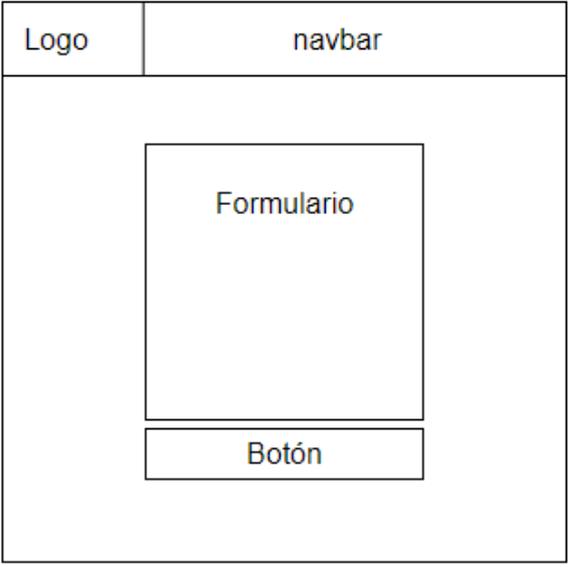
<b>Nombre</b>	<b>Vista Realizar Pago</b>
<b>Descripción</b>	Ventana que muestra un formulario para que el usuario introduzca los datos de compra
<b>Activación</b>	Al pulsar el icono del carrito y posteriormente el botón “Comprar”
<b>Boceto</b>	
<b>Eventos</b>	1. Si pulsamos el botón una vez introducida la información, nos valida los datos y realiza el pago.

Tabla 5.67: Diseño Interfaz Vista Realizar Pago

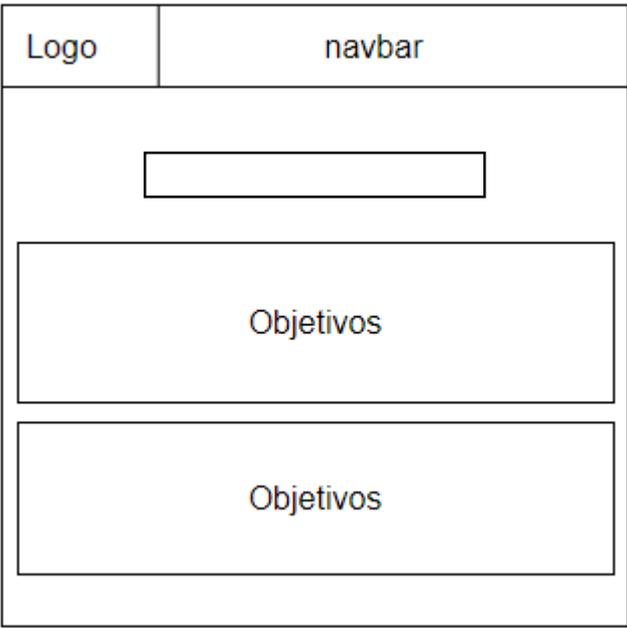
<b>Nombre</b>	<b>Vista Eliminar Objetivos Admin</b>
<b>Descripción</b>	Ventana que muestra todos los objetivos registrados en la aplicación que permite al admin eliminarlos
<b>Activación</b>	Al pulsar el icono de admin siendo usuario administrador
<b>Boceto</b>	
<b>Eventos</b>	1. Si pulsamos el botón eliminar, nos elimina el objetivo de la base de datos.

Tabla 5.68: Diseño Interfaz Vista Eliminar Objetivos Admin

### 6.4.1.- Paleta de Colores

Para el estilo de esta aplicación, se ha seguido una plantilla minimalista y moderna, simulando el diseño de las webs actuales. La paleta de colores principal consiste en tonos de blanco y negro, lo que aporta una apariencia limpia y profesional. Todos los botones de acción, como los de navegación y confirmación, son de color negro, creando un contraste nítido y atractivo con el fondo blanco.

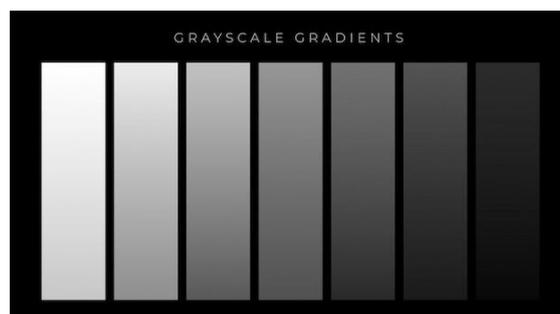


Figura 5.34: Paleta de colores



Figura 5.35: Color Botones de acción

Además, se ha implementado un esquema de color específico para los botones de eliminación, los cuales son de color rojo. Esto no solo proporciona una clara diferenciación visual, sino que también ayuda a prevenir acciones accidentales, ya que el rojo es intuitivamente reconocido como un color de advertencia.



Figura 536: Color Botones de eliminación

La interfaz está diseñada para ser responsiva, adaptándose automáticamente a diferentes tamaños de pantalla y resoluciones, asegurando una experiencia de usuario consistente tanto en dispositivos de escritorio como móviles. Los elementos de navegación están claramente delineados y accesibles, permitiendo a los usuarios moverse por la aplicación con facilidad.

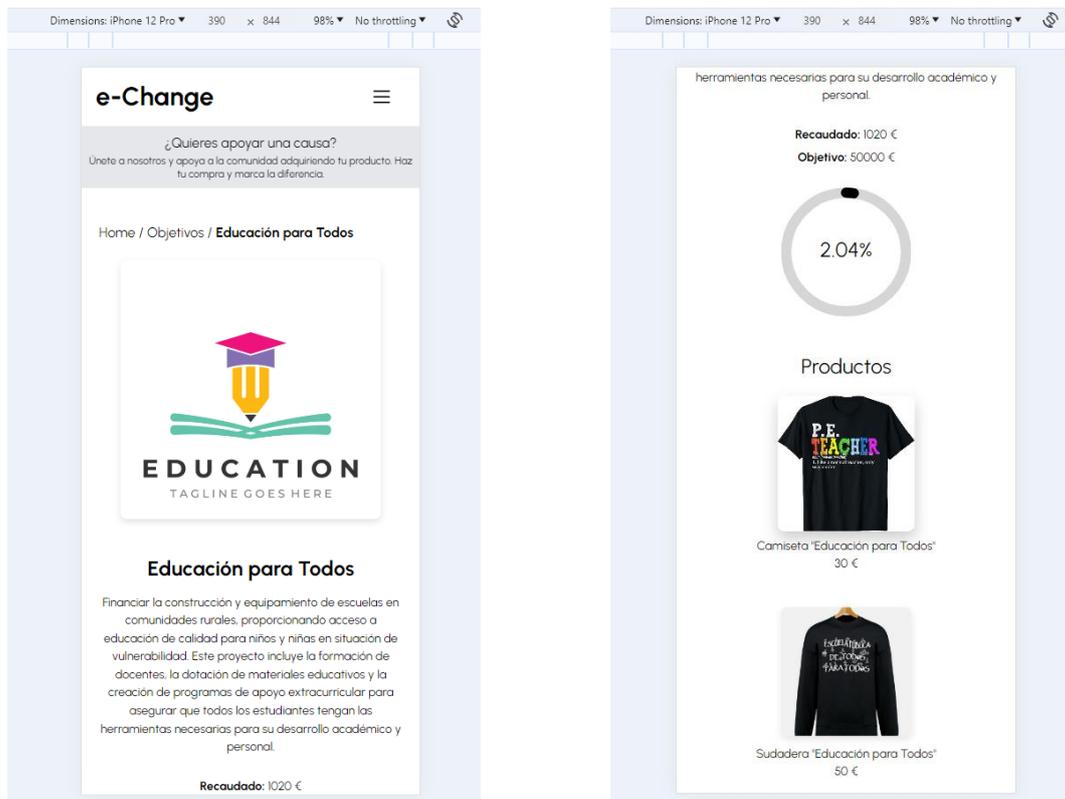


Figura 5.37: Página Objetivo Responsiva

Se ha diseñado una interfaz que capte la atención del usuario, invitándolo a explorar la plataforma hasta completar la compra de un producto, sin sobrecargarlo con estímulos excesivos. Se ha puesto especial atención en seguir tendencias modernas, como el diseño de bento-box, para crear una experiencia de usuario más inmersiva y agradable.

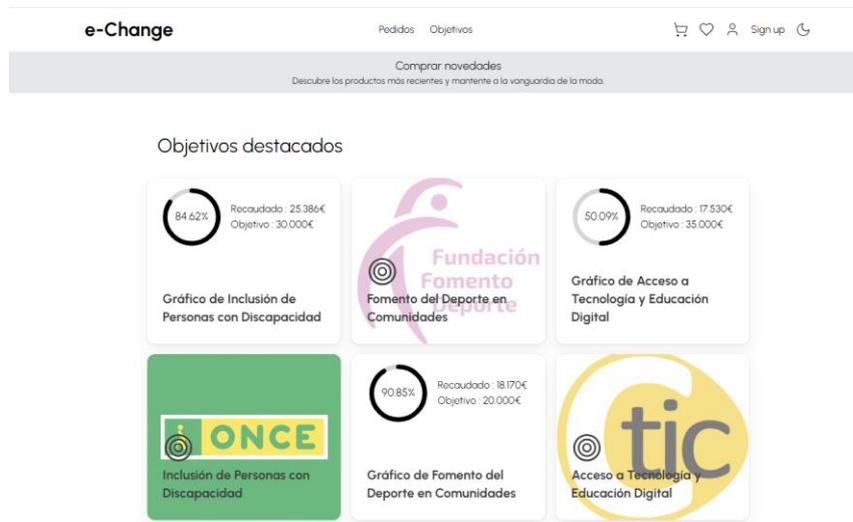


Figura 5.38: Página Inicio Bento-Box

Cada página de la aplicación ha sido cuidadosamente diseñada para ofrecer una navegación intuitiva y fluida, asegurando que todos los elementos visuales y funcionales estén alineados con los objetivos del usuario. Se ha cuidado cada detalle en la presentación de la información, manteniendo un equilibrio entre estética y funcionalidad para que la interfaz no solo sea atractiva, sino también fácil de usar.

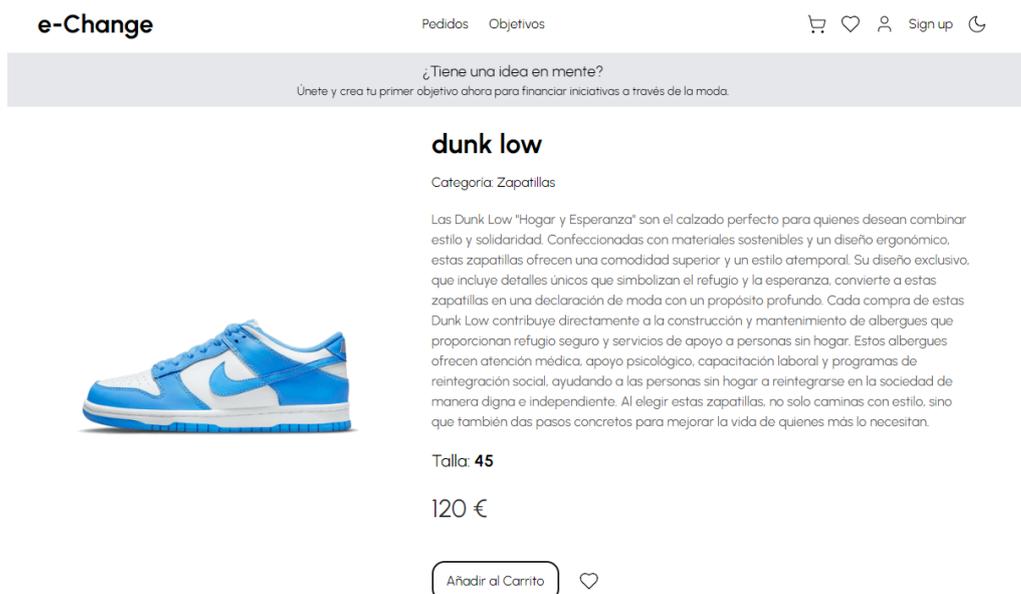


Figura 5.39: Página Producto

El logo de e-Change presenta un diseño simple y significativo que refleja los valores y la funcionalidad de la plataforma. La imagen muestra una mano sosteniendo un corazón, simbolizando la unión entre el comercio y el apoyo a causas sociales. Este símbolo destaca la intención de la plataforma de no solo facilitar la compra y venta de ropa, sino también de permitir a los usuarios contribuir a proyectos personales y sociales.

El texto "e-change" debajo del símbolo refuerza esta idea, combinando la "e" de "electronic" con "change", sugiriendo un cambio positivo impulsado por el comercio electrónico. La línea debajo del texto añade un toque de profesionalidad y equilibrio al diseño, completando el logo de manera armoniosa.



*Figura 5.40: Logo e-Change*



# Capítulo 6

## Implementación

En este capítulo se abordarán todos los aspectos relacionados con la construcción y desarrollo de la aplicación. Se explicarán en detalle los elementos técnicos.

Dividiremos este capítulo en cuatro secciones:

### 6.1.- Backend

La implementación del backend se divide en varias carpetas diferentes que abarcan varias áreas del servidor del proyecto basado en Node.js [2] y TypeScript. A continuación, se detalla cada parte de esta estructura:

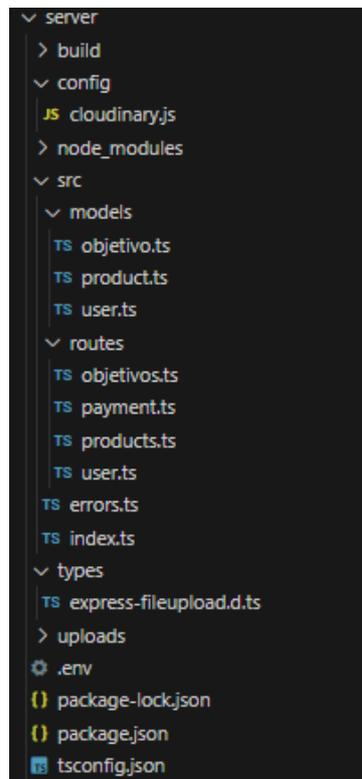


Figura 6.41: Estructura Carpetas Backend

#### 1.- config:

**cloudinary.js:** Este archivo específico contiene configuraciones para integrar Cloudinary [10] con el servidor, incluyendo la exportación de las claves API necesarias para su funcionamiento.

## 2.- node\_modules:

Este directorio contiene todas las dependencias y paquetes instalados a través de npm, necesarios para el funcionamiento del proyecto.

## 3.- src:

Este es el directorio principal de código fuente donde se desarrollan todas las funcionalidades del backend.

Dentro de este directorio podemos encontrar dos carpetas:

### models:

Los modelos de datos definen la estructura de las entidades utilizadas en la aplicación. Estos archivos están escritos en TypeScript (.ts) y se encuentran en la carpeta models. En ella se detallan tres modelos diferentes: un modelo para los objetivos, otro para los productos y otro para los usuarios.

Cada uno de estos modelos incluye una interfaz que define la estructura de los datos y se utiliza mediante el Schema de Mongoose para interactuar con MongoDB [8]. Los modelos detallan cómo serán los datos que se guardarán en la base de datos, garantizando la integridad y validación de los mismos.

```
import { Schema, model, Document, Types } from "mongoose";
import { IProduct } from "../product";

export interface IObjetivo extends Document {
  userId: Types.ObjectId;
  title: string;
  description: string;
  imageURL: string;
  targetAmount: number;
  currentAmount: number;
  products: {
    product: Types.ObjectId | IProduct;
    quantity: number;
  }[];
}

const ObjetivoSchema = new Schema<IObjetivo>({
  userId: { type: Schema.Types.ObjectId, ref: "User", required: true },
  title: { type: String, required: true },
  description: { type: String, required: true },
  imageURL: { type: String, required: true },
  targetAmount: { type: Number, required: true, min: [0, 'El valor objetivo debe ser mayor o igual a 0'] },
  currentAmount: { type: Number, required: true, min: [0, 'El valor actual debe ser mayor o igual a 0'] },
  products: [{
    product: { type: Schema.Types.ObjectId, ref: "Product", required: true },
    quantity: { type: Number, required: true, min: [1, 'La cantidad debe ser al menos 1'] }
  }]
});

export const ObjetivoModel = model<IObjetivo>("Objetivo", ObjetivoSchema);
```

Figura 6.42: Modelo Objetivo

Se define una interfaz que extiende Document de Mongoose. Esta interfaz especifica la estructura de los datos, incluyendo los tipos de cada campo y cualquier relación con otros modelos.

Se crea un Schema de Mongoose utilizando la interfaz definida. El Schema especifica las propiedades de cada campo, incluyendo su tipo, referencias a otros modelos, y validaciones como required y valores mínimos o máximos.

Finalmente, el modelo se exporta utilizando el método `model` de Mongoose. Esto permite que el modelo sea utilizado en otras partes de la aplicación para interactuar con la base de datos.

**routes:**

Contiene las rutas del servidor, que manejan las solicitudes HTTP y definen los endpoints de la API.

**objetivos.ts:**

Maneja las rutas relacionadas con los objetivos. Define endpoints para crear, editar, eliminar y obtener objetivos, así como agregar y eliminar productos de los objetivos.

**payment.ts:**

Maneja las rutas relacionadas con los pagos. Define endpoints para procesar pagos y manejar eventos webhook de Stripe [4].

**products.ts:**

Maneja las rutas relacionadas con los productos. Define endpoints para crear, eliminar y obtener productos.

**user.ts:**

Maneja las rutas relacionadas con los usuarios. Define endpoints para el registro, inicio de sesión y gestión de perfiles de usuario.

Este fragmento de código importa los módulos y funciones necesarios para manejar el registro de usuarios y la verificación de tokens JWT en una aplicación Express. Se utilizan los módulos `express`, `bcrypt`, `jsonwebtoken`, y el modelo de usuario definido en `UserModel`.

```
import { Router, Request, Response, NextFunction } from 'express';
import bcrypt from 'bcrypt';
import jwt from 'jsonwebtoken';
import { IUser, UserModel } from '../models/user';
import { UserErrors } from '../errors';

const router = Router();
```

Figura 6.43: Módulos y funciones

```

router.post("/register", async (req: Request, res: Response) => {
  const { username, email, password } = req.body;

  try {
    const userByUsername = await UserModel.findOne({ username });
    const userByEmail = await UserModel.findOne({ email });

    if (userByUsername) {
      return res.status(400).json({ type: UserErrors.USERNAME_ALREADY_EXISTS });
    }

    if (userByEmail) {
      return res.status(400).json({ type: UserErrors.EMAIL_ALREADY_EXISTS });
    }

    const hashedPassword = await bcrypt.hash(password, 10);

    const newUser = new UserModel({ username, email, password: hashedPassword, role: 'usuario' });
    await newUser.save();
    res.json({ message: "Usuario Registrado Correctamente" });
  } catch (err) {
    res.status(500).json({ type: err });
  }
});

```

Figura 6.44: API user

## Uso de bcrypt para Manejo de Contraseñas

Para garantizar la seguridad de las cuentas de los usuarios, se utiliza el módulo bcrypt. Este módulo permite encriptar las contraseñas antes de almacenarlas en la base de datos. De esta manera, incluso si la base de datos es comprometida, las contraseñas estarán protegidas ya que estarán encriptadas y no en texto plano.

## Implementación de Tokens JWT

Para garantizar que la persona que inicia sesión con una cuenta de usuario es quien dice ser, se implementa un sistema de tokens JWT (JSON Web Tokens). Al iniciar sesión, se genera un token JWT que se envía al cliente. Este token debe ser enviado con cada solicitud posterior que requiera autenticación. El servidor verifica este token para asegurarse de que la solicitud proviene de un usuario autenticado.

```

export const verifyToken = (req: Request, res: Response, next: NextFunction) => {
  const token = req.headers['authorization']?.split(' ')[1];

  if (!token) {
    return res.status(403).json({ message: 'No token provided' });
  }

  try {
    const decoded = jwt.verify(token, "secreto") as { id: string };
    req.userId = decoded.id;
    next();
  } catch (error) {
    return res.status(401).json({ message: 'Unauthorized' });
  }
};

```

Figura 6.45: Función VerifyToken

### errors.ts:

Maneja los errores y sus respuestas. Define enumeraciones para los errores específicos de usuarios y productos.

### index.ts:

Define las rutas principales y agrupa todas las rutas del servidor. Además, se encarga de conectar la base de datos utilizando Mongoose y de definir el puerto en el que se ejecutará el backend. También establece la ruta del frontend.

```

import express from 'express';
import cors from 'cors';
import mongoose from 'mongoose';
import fileUpload from 'express-fileupload';
import { userRouter } from './routes/user';
import { productRouter } from './routes/products';
import { objetivoRouter } from './routes/objetivos';
import { paymentRouter } from './routes/payment';
import cloudinary from './config/cloudinary';
import dotenv from 'dotenv';

dotenv.config();

const app = express();

app.post('/webhook', express.raw({ type: 'application/json' })), (req, res, next) => {
  paymentRouter(req, res, next);
});

app.use(express.json());
app.use(fileUpload({
  useTempFiles: true,
  tempFileDir: './uploads'
}));
app.use(cors({
  origin: 'http://localhost:3000',
  methods: 'GET,HEAD,PUT,PATCH,POST,DELETE',
  credentials: true
}));

app.use('/', userRouter);
app.use('/productos', productRouter);
app.use('/objetivos', objetivoRouter);
app.use('/pedidos', paymentRouter);

mongoose.connect(
  process.env.MONGO_URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true
  }, (err) => {
    if (err) console.log(err);
  });

app.use((req, res, next) => {
  res.status(404).send('Página no encontrada');
});

app.listen(3001, () => console.log("SERVER STARTED"));

```

Figura 6.46: index.ts

Este archivo maneja la ruta sobre la que la plataforma de pago enviará eventos, permitiendo que el servidor backend escuche y ejecute diferentes acciones según la respuesta recibida sobre el pago y los datos asociados.

Al establecer el puerto en el que el servidor backend estará escuchando, permite a la aplicación que responda a solicitudes HTTP en ese puerto específico.

#### 4.- types:

Dentro de types podemos encontrar configuraciones de tipos de TypeScript para poder manejar la subida de archivos mediante Express y que funcione correctamente.

#### express-fileupload.d.ts:

Define los tipos para el manejo de la subida de archivos en Express.

#### 5.- .env:

Archivo de variables de entorno que almacena configuraciones sensibles como claves API, URL de la base de datos, y otros datos de configuración que no deben estar en el código fuente.

#### 6.- package-lock.json:

Archivo generado automáticamente que describe la estructura exacta de las dependencias instaladas en node\_modules. Asegura que las instalaciones futuras sean consistentes con las versiones especificadas.

#### 7.- package.json:

Archivo principal de configuración del proyecto Node.js [2] que incluye metadatos del proyecto, scripts de npm, y la lista de dependencias y sus versiones.

## **8.- tsconfig.json**

Archivo de configuración de TypeScript que define las opciones del compilador y las configuraciones del proyecto TypeScript.

## **6.2.- Frontend**

La principal función de implementar el frontend es la de crear un diseño de la interfaz que atienda a varias necesidades cruciales tanto para los usuarios como para la eficacia general de la aplicación. Un frontend bien diseñado no solo mejora la apariencia visual de la aplicación, sino que también contribuye significativamente a la usabilidad, accesibilidad, y experiencia general del usuario. Se divide en varios directorios y archivos del frontend basados en React [16] y TypeScript

### **1.- app:**

Contiene archivos de configuración de estado global de la aplicación, en este caso tiene un archivo de estilos css que define algunos estilos que se siguen en la aplicación.

### **2.- node\_modules:**

Contiene todas las dependencias y paquetes instalados a través de npm, necesarios para el funcionamiento del proyecto.

### **3.- public:**

Contiene archivos públicos que se sirven directamente al cliente, como index.html, imágenes, y otros recursos estáticos.

### **4.- src:**

Este es el directorio principal de código fuente donde se desarrollan todas las funcionalidades del frontend.

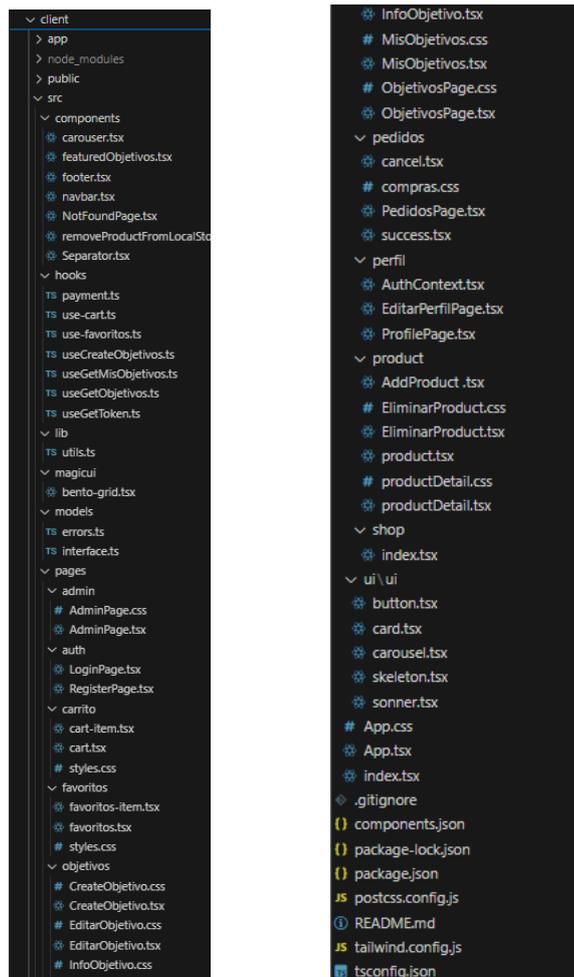


Figura 6.47: Estructura de carpetas Frontend

A su vez, dentro de cada directorio se encuentran archivos o subdirectorios que son importantes para definir el frontend. A continuación, se explica en mayor profundidad la funcionalidad de la implementación de cada uno de ellos:

Dentro del directorio **src**, se puede encontrar el subdirectorio **components** que contiene los distintos componentes reutilizables de la interfaz de usuario, que pueden ser usados en páginas.

**carousel.tsx:** Componente de carrusel para mostrar imágenes o contenido deslizable.

**featuredObjetivos.tsx:** Componente que muestra los objetivos destacados.

**footer.tsx:** Componente de pie de página.

**navbar.tsx:** Componente de barra de navegación.

**NotFoundPage.tsx:** Página que se muestra cuando una ruta no existe.

**removeProductFromLocalStorage.ts:** Función para eliminar un producto del almacenamiento local.

**Separator.tsx:** Componente de separador visual.

Se puede encontrar el subdirectorio **hooks** que contiene hooks personalizados para gestionar el estado y lógica de la aplicación.

**use-cart.ts:** Hook para manejar el carrito de compras.

**use-favoritos.ts:** Hook para manejar los favoritos.

**useCreateObjetivos.ts:** Hook para crear objetivos.

**useGetMisObjetivos.ts:** Hook para obtener los objetivos del usuario.

**useGetObjetivos.ts:** Hook para obtener objetivos.

Este hook permite realizar una petición al backend a la API en la ruta base /objetivos utilizando axios para ejecutar una función asíncrona que devuelve todos los objetivos almacenados en la base de datos de la aplicación. Durante esta llamada, se establece un estado de carga (loading) que se utilizará en diversas páginas para mostrar un skeleton mientras se espera la respuesta.

```
import { useEffect, useState, useCallback, useRef } from "react";
import axios from "axios";
import { useGetToken } from "../useGetToken";
import { IObjetivo } from "../models/interface";

export const useGetObjetivos = () => {
  const [objetivos, setObjetivos] = useState<IObjetivo[]>([]);
  const [error, setError] = useState<string | null>(null);
  const [loading, setLoading] = useState<boolean>(true);
  const { headers } = useGetToken();
  const hasFetched = useRef(false);

  const fetchObjetivos = useCallback(async () => {
    if (hasFetched.current) return;
    hasFetched.current = true;

    setLoading(true);
    try {
      const response = await axios.get('http://localhost:3001/objetivos', { headers });
      setObjetivos(response.data);
    } catch (err) {
      setError("ERROR: Algo ha ido mal al obtener los objetivos");
    } finally {
      setLoading(false);
    }
  }, [headers]);

  useEffect(() => {
    fetchObjetivos();
  }, [fetchObjetivos]);

  return { objetivos, fetchObjetivos, error, loading };
};
```

Figura 6.48: Hook useGetObjetivos

**useGetToken.ts:** Hook para obtener el token de autenticación.

También se encuentra el subdirectorio **lib** que contiene librerías y utilidades que se utilizan en diferentes partes de la aplicación.

El subdirectorio **magicui** [6] que contiene:

**bento-grid.tsx:** Componente para crear una cuadrícula estilo Bento, utilizado para la disposición visual.

El subdirectorio **models** que contiene modelos de datos utilizados en la aplicación.

**errors.ts:** Definiciones de errores específicos de la aplicación.

**interface.ts:** Define un modelo para los productos y los objetivos parecido al del backend.

El subdirectorio **pages** contiene las diferentes páginas de la aplicación, organizadas por funcionalidad.

#### **admin**

**AdminPage.css:** Estilos para la página de administración.

**AdminPage.tsx:** Página de administración.

#### **auth**

**LoginPage.tsx:** Página de inicio de sesión.

**RegisterPage.tsx:** Página de registro.

#### **carrito**

**cart-item.tsx:** Componente de ítem del carrito.

**cart.tsx:** Página del carrito.

**styles.css:** Estilos para el carrito.

#### **favoritos**

**favoritos-item.tsx:** Componente de ítem de favoritos.

**favoritos.tsx:** Página de favoritos.

**styles.css:** Estilos para los favoritos.

#### **objetivos**

**CreateObjetivo.css:** Estilos para la creación de objetivos.

**CreateObjetivo.tsx:** Página para crear objetivos.

**EditarObjetivo.css:** Estilos para editar objetivos.

**EditarObjetivo.tsx:** Página para editar objetivos.

**InfoObjetivo.css:** Estilos para información de los objetivos.

**InfoObjetivo.tsx:** Página de información de objetivos.

**MisObjetivos.css:** Estilos para la página de mis objetivos.

**MisObjetivos.tsx:** Página de mis objetivos.

**ObjetivosPage.css:** Estilos para la página de objetivos.

**ObjetivosPage.tsx:** Página principal de objetivos.

## **pedidos**

**cancel.tsx:** Página de cancelación de pedidos.

**compras.css:** Página de estilo para lascompras.

**PedidosPage.tsx:** Página de pedidos.

**success.tsx:** Página de éxito en pedidos.

## **perfil**

**AuthContext.tsx:** Contexto de autenticación.

**EditarPerfilPage.tsx:** Página para editar el perfil.

**ProfilePage.tsx:** Página del perfil del usuario.

## **product**

**AddProduct.tsx:** Página para agregar productos.

**EliminarProduct.tsx:** Página para eliminar productos.

**EliminarProduct.css:** Estilos para eliminar productos.

**Product.tsx:** Página de producto.

**ProductDetails.tsx:** Página de detalles del producto.

**ProductDetails.css:** Estilos para la página de detalles del producto.

## **shop**

**Index.tsx:** Página principal de la tienda.

El subdirectorio **ui** que contiene componentes de interfaz de usuario reutilizables, esenciales para la creación de una experiencia de usuario cohesiva.

**button.tsx:** Componente de botón.

**card.tsx:** Componente de tarjeta.

**carousel.tsx:** Componente de carrusel.

**skeleton.tsx:** Componente de esqueleto de carga.

**spinner.tsx:** Componente de spinner de carga.

Una vez ya fuera del directorio src, se encuentran los diferentes archivos.

## **App.css:**

Estilos globales de la aplicación.

## App.tsx:

Componente principal de la aplicación que agrupa y gestiona otros componentes. Configura la aplicación React [16], estableciendo las rutas y componentes principales. Utiliza react-router-dom para gestionar la navegación y sonner para mostrar notificaciones.

```
function App() {
  return (
    <AuthProvider>
      <Router>
        <div id="root">
          <div className="App">
            <Navbar />
            <Toaster position="top-right" />
            <div className="content">
              <CarouselText />
              <Routes>
                <Route path="/" element={<ShopPage />} />
                <Route path="/login" element={<PublicRoute><LoginPage /></PublicRoute>} />
                <Route path="/register" element={<PublicRoute><RegisterPage /></PublicRoute>} />
                <Route path="/profile" element={<PrivateRoute><ProfilePage /></PrivateRoute>} />
                <Route path="/edit-profile" element={<PrivateRoute><EditarPerfilPage /></PrivateRoute>} />
                <Route path="/carrito" element={<CarritoPage />} />
                <Route path="/pedidos" element={<PrivateRoute><PedidosPage /></PrivateRoute>} />
                <Route path="/objetivos" element={<ObjetivosPage />} />
                <Route path="/favoritos" element={<Favoritos />} />
                <Route path="/crear-objetivo" element={<PrivateRoute><CreateObjetivo /></PrivateRoute>} />
                <Route path="/mis-objetivos" element={<PrivateRoute><MisObjetivos /></PrivateRoute>} />
                <Route path="/objetivos/:id" element={<InfoObjetivo />} />
                <Route path="/editar-objetivo/:id" element={<PrivateRoute><EditarObjetivo /></PrivateRoute>} />
                <Route path="/productos/:productId" element={<ProductDetail />} />
                <Route path="/success" element={<SucessPage />} />
                <Route path="/cancel" element={<CancelPage />} />
                <Route path="/admin" element={<AdminRoute><AdminPage /></AdminRoute>} />
                <Route path="*" element={<NotFoundPage />} />
              </Routes>
            </div>
          </div>
          <Footer />
        </div>
      </Router>
    </AuthProvider>
  );
}
```

Figura 6.49: App.tsx

```
function PrivateRoute({ children }) {
  const { isAuthenticated } = useAuth();
  return isAuthenticated ? children : <Navigate to="/login" />;
}

function PublicRoute({ children }) {
  const { isAuthenticated } = useAuth();
  return !isAuthenticated ? children : <Navigate to="/" />;
}

function AdminRoute({ children }) {
  const { isAuthenticated, userRole } = useAuth();
  return isAuthenticated && userRole === 'admin' ? children : <Navigate to="/" />;
}
```

Figura 6.50: Roles

**PrivateRoute:** Componente que asegura que solo los usuarios autenticados pueden acceder a ciertas rutas. Redirige a la página de inicio de sesión si no están autenticados.

**PublicRoute:** Componente que asegura que solo los usuarios no autenticados pueden acceder a ciertas rutas. Redirige a la página principal si ya están autenticados.

**AdminRoute:** Componente que asegura que solo los administradores autenticados pueden acceder a ciertas rutas. Redirige a la página principal si no tienen el rol de administrador.

**index.tsx:**

Punto de entrada principal de la aplicación, responsable de renderizar el componente principal (App) y montar la aplicación en el DOM.

**Archivos de Configuración:****components.json:**

Configuración específica de componentes.

**package-lock.json:**

Archivo generado automáticamente que describe la estructura exacta de las dependencias instaladas en node\_modules.

**package.json:**

Archivo principal de configuración del proyecto Node.js que incluye metadatos del proyecto, scripts de npm, y la lista de dependencias y sus versiones.

**tailwind.config.js:**

Configuración de Tailwind CSS para el uso de utilidades de estilos en el proyecto.

**tsconfig.json:**

Archivo de configuración de TypeScript que define las opciones del compilador y las configuraciones del proyecto TypeScript.

## 6.3.- Modelo seguido

Como modelo, se ha seguido el patrón Modelo-Vista-Controlador (MVC) [5]. MVC es un patrón de diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Este patrón enfatiza la separación entre la lógica de negocios y su visualización, lo que proporciona una mejor división del trabajo y facilita el mantenimiento de la aplicación.

El patrón MVC se compone de tres componentes principales:

Modelo:

Maneja los datos y la lógica de negocios de la aplicación. Es responsable de recuperar y almacenar los datos, así como de aplicarle las reglas de negocio. En la aplicación web, el modelo puede interactuar con una base de datos y realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

Vista:

Se encarga del diseño y la presentación de la información al usuario. La vista recibe los datos del modelo y los muestra de una manera específica. En la aplicación web, la vista estaría compuesta por React, TypeScript, HTML y CSS para renderizar la interfaz de usuario.

Controlador:

Actúa como intermediario entre el modelo y la vista. El controlador recibe las entradas del usuario (por ejemplo, a través de formularios o clics en botones), procesa estas entradas (por ejemplo, validando datos o ejecutando lógica de negocio) y actualiza el modelo o la vista en consecuencia.

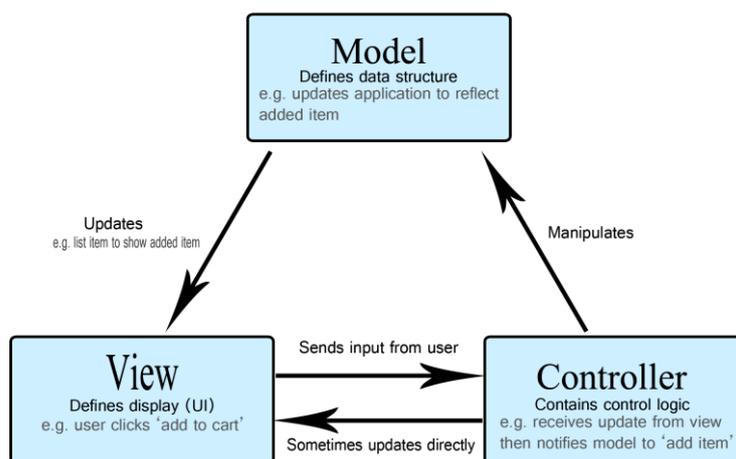


Figura 6.51: MVC

## 6.4.- Dificultades

A lo largo del proyecto he encontrado varias dificultades que me han hecho pelearme con el código en numerosas ocasiones. Incluso ha habido momentos en los que lograba arreglar una cosa o un componente para una página, pero rompía otra cosa en el proceso. Uno de los mayores problemas al desarrollar la aplicación fue implementar la pasarela de pago y hacer que funcionara correctamente. Cuando finalmente conseguí conectar la aplicación para que atendiera las llamadas de la API de la pasarela de pago, tuve que investigar bastante sobre el Webhook para entender cómo podía gestionar los eventos en localhost. Esto era crucial para que el servidor backend pudiera escuchar y recibir los eventos tipo POST lanzados por el Webhook cada vez que se realizaba un pago a través de la pasarela de pago. La instalación del CLI y su configuración para el entorno me llevó varios días, y cada vez que pensaba que lo había conseguido, surgían nuevos problemas que me impedían redirigir correctamente a la pasarela de pago.

## 6.5.- Despliegue en la nube

Aunque la aplicación esté actualmente en localhost, en caso de desplegarla en la nube, el proceso implicaría varios pasos clave para asegurar que todos los componentes de la aplicación funcionen correctamente en un entorno de producción. A continuación, se describen los pasos necesarios para desplegar una aplicación MERN (MongoDB Atlas [8], Express, React, Node.js) con Cloudinary [3] y la pasarela de pago Stripe [4].

Habría que configurar las variables de entorno necesarias para la producción. Esto incluye las claves de API para MongoDB Atlas, Cloudinary y Stripe. Estas se pueden gestionar utilizando un archivo .env que será cargado por un paquete como dotenv.

Crear las diferentes cuentas de las plataformas de MongoDB Atlas, Cloudinary y Stripe, todas en modo producción y no en modo test, lo cual podría implicar cambiar el plan de la plataforma dependiendo de la escalabilidad requerida.

### **Elección del Proveedor de Servicios en la Nube:**

Elegir un proveedor de servicios en la nube como AWS, Heroku, Vercel [11], o DigitalOcean para desplegar tu aplicación.



Figura 6.52: Logo Vercel [11]

### **Configuración del Entorno de Despliegue:**

Configura el entorno de despliegue para que pueda manejar las variables de entorno y las dependencias necesarias.

### **Despliegue del Backend:**

Subir el código del backend a tu proveedor de servicios en la nube. Esto incluye configurar el entorno de ejecución en este caso Node.js, establecer las variables de entorno y asegurar que tu aplicación puede conectarse a MongoDB Atlas, Cloudinary y Stripe.

### **Despliegue del Frontend:**

Subir el código del frontend. En caso de utilizar un servicio como Vercel o Netlify, estos pueden manejar fácilmente aplicaciones React. Configurar las variables de entorno necesarias para el frontend.

### **Configuración de Certificados SSL:**

Asegurarse de que la aplicación esté configurada para usar HTTPS, lo que puede implicar la configuración de certificados SSL.

Por último pero no menos importante realizar pruebas exhaustivas para asegurar que todas las funcionalidades (autenticación, operaciones CRUD, carga de archivos, pagos) funcionen correctamente en el entorno de producción.



# Capítulo 7

## Pruebas

En este capítulo se llevan a cabo las pruebas necesarias para asegurar el correcto funcionamiento de la aplicación web. El contenido se organiza en dos secciones principales: pruebas de caja negra y pruebas de caja blanca.

### 8.1.- Caja negra

Las pruebas de caja negra, también conocidas como pruebas funcionales, se centran en verificar el comportamiento externo del software sin necesidad de conocer su estructura interna. El objetivo es asegurarse de que las funcionalidades de la aplicación cumplen con los requisitos especificados y que los resultados producidos son correctos. Estas pruebas se basan en los inputs y outputs del sistema, evaluando si el software responde adecuadamente a diversas entradas y situaciones.

PCN_01: Registro de Usuario	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios registrarse correctamente.
<b>Precondiciones</b>	El usuario no debe estar registrado en el sistema.
<b>Datos de entrada</b>	Nombre de usuario: Juanele Correo electrónico:juan.lazaro@gmail.com Contraseña:123456Aa
<b>Acción esperada</b>	Al pulsar el botón "Registrar" se mostrará un mensaje de registro exitoso.
<b>Resultado</b>	Positivo

Tabla 7.69: PCN\_01 Registro de Usuario

PCN_02: Autenticación de Usuario	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios autenticarse correctamente.
<b>Precondiciones</b>	El usuario no debe estar registrado en el sistema.
<b>Datos de entrada</b>	Correo electrónico:juan.lazaro@gmail.com Contraseña:123456Aa
<b>Acción esperada</b>	Al pulsar el botón "Iniciar Sesión" se mostrará un mensaje de inicio de sesión exitoso y me lleva a la página principal.
<b>Resultado</b>	Positivo

Tabla 7.70: PCN\_02 Autenticación de Usuario

<b>PCN_03: Creación de Objetivo</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios crear nuevos objetivos.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	Título del objetivo:Cáritas  Descripción:" Cáritas es una organización internacional dedicada a combatir la pobreza y la exclusión social, ofreciendo ayuda humanitaria y promoviendo el desarrollo integral de las personas. Con tu apoyo, podemos seguir brindando alimentos, ropa, y refugio a quienes más lo necesitan, así como implementar programas de educación y empleo que transforman vidas. Únete a nuestra causa y juntos construiremos un mundo más justo y solidario.  Imagen:LogoCáritas.png  Cantidad objetivo:250€
<b>Acción esperada</b>	Al pulsar el botón "Crear Objetivo" se mostrará un mensaje indicando que el objetivo de ha creado.
<b>Resultado</b>	Positivo

Tabla 7.71: PCN\_03 Creación de Objetivo

<b>PCN_04: Modificar Objetivo</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios modificar sus objetivos.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	Título del objetivo:CáritasModificado  Descripción:" Cáritas es una organización internacional dedicada a combatir la pobreza y la exclusión social, ofreciendo ayuda humanitaria y promoviendo el desarrollo integral de las personas. Con tu apoyo, podemos seguir brindando alimentos, ropa, y refugio a quienes más lo necesitan, así como implementar programas de educación y empleo que transforman vidas. Únete a nuestra causa y juntos construiremos un mundo más justo y solidario.  Cantidad objetivo:500€
<b>Acción esperada</b>	Al pulsar el botón "Guardar Cambios" se mostrará un mensaje indicando que el objetivo de ha modificado.
<b>Resultado</b>	Positivo

Tabla 7.72: PCN\_04 Modificar Objetivo

<b>PCN_05: Eliminar Objetivo</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios eliminar sus objetivos.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el botón "Eliminar" se mostrará un mensaje indicando que el objetivo de ha eliminado.
<b>Resultado</b>	Positivo

Tabla 7.73: PCN\_05 Eliminar Objetivo

<b>PCN_06: Añadir Producto</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios añadir productos a sus objetivos.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	Nombre del Producto: Camiseta Inter  Precio:20€  Descripción:" Camiseta del Inter de Milán, cómoda para hacer deporte"  Imagen: Camiseta.png  Cantidad en stock: 1  Categoría: Camisetas  Talla: L
<b>Acción esperada</b>	Al pulsar el botón "Agregar Producto" se mostrará un mensaje indicando que el producto se ha creado en el objetivo.
<b>Resultado</b>	Positivo

Tabla 7.74: PCN\_06 Añadir Producto

<b>PCN_07: Añadir Producto al Carrito</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios añadir productos al carrito.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el icono "Carrito" se mostrará un mensaje indicando que el producto se ha añadido al carrito.
<b>Resultado</b>	Positivo

Tabla 7.75: PCN\_07 Añadir Producto al Carrito

<b>PCN_08: Añadir Producto a Favoritos</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios añadir productos a favoritos.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el icono "Favoritos" se mostrará un mensaje indicando que el producto se ha añadido a favoritos.
<b>Resultado</b>	Positivo

Tabla 7.76: PCN\_08 Añadir Producto a Favoritos

<b>PCN_09: Eliminar Producto</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios eliminar productos de un objetivo.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el botón "Eliminar" se mostrará un mensaje indicando que el producto se ha eliminado.
<b>Resultado</b>	Positivo

Tabla 7.77: PCN\_09 Eliminar Producto

<b>PCN_10: Eliminar Producto de Favoritos</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios eliminar productos de favoritos.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el icono "X" se mostrará un mensaje indicando que el producto se ha eliminado y se eliminará el producto de favoritos.
<b>Resultado</b>	Positivo

Tabla 7.78: PCN\_10 Eliminar Producto de Favoritos

<b>PCN_11: Eliminar Producto de Carrito</b>	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios eliminar productos del carrito.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el icono "X" se mostrará un mensaje indicando que el producto se ha eliminado y se eliminará el producto del carrito.
<b>Resultado</b>	Positivo

Tabla 7.79: PCN\_11 Eliminar Producto del Carrito

PCN_12: Editar Perfil de Usuario	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios modificar su perfil correctamente.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	Nombre de usuario: Juanele2 Contraseña: 123456Aa
<b>Acción esperada</b>	Al pulsar el botón "Guardar Cambios" se mostrará un mensaje de perfil actualizado con éxito.
<b>Resultado</b>	Positivo

Tabla 7.80: PCN\_12 Editar Perfil de Usuario

PCN_13: Cerrar Sesión	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios cerrar su sesión.
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Datos de entrada</b>	N/A
<b>Acción esperada</b>	Al pulsar el botón "Cerrar sesión" o icono "Puerta" se mostrará el login o la página de inicio.
<b>Resultado</b>	Positivo

Tabla 7.81: PCN\_13 Cerrar Sesión

PCN_14: Buscar Objetivo	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios buscar objetivos según sus parámetros de búsqueda.
<b>Precondiciones</b>	Debe existir un objetivo
<b>Datos de entrada</b>	Objetivo: Cáritas
<b>Acción esperada</b>	Al escribir en el cuadro de texto se muestra el objetivo solicitado.
<b>Resultado</b>	Positivo

Tabla 7.82: PCN\_14 Buscar Objetivo

PCN_15: Filtrar Productos	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios buscar objetivos/productos según sus categoría.
<b>Precondiciones</b>	Debe existir un objetivo y un producto
<b>Datos de entrada</b>	Categoría: Camisetas
<b>Acción esperada</b>	Al seleccionar la categoría se muestra el Objetivo que contiene esa categoría.
<b>Resultado</b>	Positivo

Tabla 7.83: PCN\_15 Filtrar Productos

PCN_16:Realizar Pago	
<b>Objetivo</b>	Verificar que el sistema permite a los usuarios realizar el pago del contenido del carrito.
<b>Precondiciones</b>	Deben existir productos en el carrito
<b>Datos de entrada</b>	Titular: Juan Lázaro  Tarjeta: 4242 4242 4242  Validez: 08/24  CVV:424
<b>Acción esperada</b>	Al dar al botón de "Pagar" realiza el pago y me redirige a una página que indica que la compra ha sido exitosa.
<b>Resultado</b>	Positivo

Tabla 7.84: PCN\_16 Realizar Pago

## 8.2.- Caja blanca

Las pruebas de caja blanca, se centran en el análisis detallado del código fuente de la aplicación. Estas pruebas son esenciales para evaluar y validar la funcionalidad interna de cada módulo, asegurando que todos los caminos lógicos del programa se comporten según lo esperado.

Durante estas pruebas, se inspecciona el comportamiento interno del software para detectar posibles fallos en el código. Este proceso es crucial para garantizar la integridad y la corrección del software. En el contexto del proyecto e-Change, se han llevado a cabo diversas pruebas de caja blanca para asegurar la calidad y la robustez de la aplicación.

1.- **Conexión con la Base de Datos:** Verificar que la aplicación puede conectarse correctamente a la base de datos y que las conexiones se establecen y cierran adecuadamente.

2.- **Entrada y Salida de Datos con el Servidor de la Base de Datos:** Asegurar que los datos se envían y recuperan correctamente desde la base de datos sin pérdidas ni errores.

3.- **Manejo de Errores en la Conexión:** Comprobar que la aplicación maneja adecuadamente los errores de conexión con la base de datos, proporcionando mensajes de error claros y precisos al usuario.

4.- **Autenticación y Creación de Cuentas de Usuario:** Verificar que el proceso de registro y autenticación de usuarios funciona correctamente, incluyendo la encriptación de contraseñas y la gestión de sesiones.

5.- **Rutas Introducidas por el Usuario:** Asegurar que las rutas de navegación dentro de la aplicación son gestionadas correctamente, llevando al usuario a la página esperada sin errores.

6.- **Manejo de Sesiones:** Evaluar que las sesiones de usuario se gestionan de manera segura, incluyendo el inicio y cierre de sesión, y la persistencia de la sesión durante la navegación.

7.- **Análisis de roles:** Garantizando el funcionamiento adecuado tanto para usuarios no autenticados como para usuarios autenticados.

8.- **Adaptabilidad de Interfaz:** La capacidad de adaptar la interfaz en función del rol del usuario.



# Capítulo 8

## Manual de usuario

En este capítulo se proporciona la documentación necesaria en forma de manual para que cualquier usuario pueda entender el funcionamiento de la aplicación. Los manuales se dividen en dos categorías principales:

### 8.1.- Manual de instalación

Este manual proporciona una guía detallada paso a paso para la instalación de la aplicación. Incluye todos los requisitos previos necesarios, como el software y hardware requeridos, así como instrucciones específicas para diferentes sistemas operativos. Además, se ofrecen soluciones a problemas comunes que pueden surgir durante el proceso de instalación, asegurando que los usuarios puedan completar la instalación sin contratiempos.

Antes de comenzar, asegúrate de tener instalados los siguientes componentes:

#### 1.- Node.js

#### 2.- npm (Node Package Manager): Se instala automáticamente con Node.js.

Necesitarás clonar un repositorio de la aplicación desde GitHub para poder tener un control de versiones con el que trabajar e ir subiendo tus modificaciones.

```
git clone https://github.com/tu-usuario/tu-repositorio.git  
cd tu-repositorio
```

El siguiente paso será configurar MongoDB Atlas. Para ello, primero necesitarás crear una cuenta a través de su sitio web. Una vez creada, configura un nuevo clúster y crea un usuario de base de datos con los permisos adecuados.

Después de realizar estos pasos, necesitarás obtener la cadena de conexión, que será una dirección de MongoDB con un usuario y una contraseña que hará referencia al clúster que has creado.

```
mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>
```

A continuación, será necesario instalar varias dependencias del servidor, para ello será necesario moverse al directorio del servidor e instalar las dependencias necesarias.

```
cd server  
npm install
```

Luego, de manera opcional, puedes configurar las variables de entorno en un archivo .env, donde se guardarán todas las claves y configuraciones necesarias.

Una vez hecho eso, podremos iniciar el servidor mediante

*npm run dev*

Tendremos la parte del servidor completamente configurada y ejecutándose en el puerto que hayamos designado. En nuestro caso es en el 3001.

Para la parte del cliente, tendremos que instalar las dependencias necesarias, por lo que tendremos que cambiar de directorio. Es recomendable manejar dos terminales diferentes: una para el backend y otra para el frontend. El IDE Visual Studio Code nos permite manejar varias terminales a la vez de manera fácil.

*cd client*

*npm install*

Al igual que en el backend, se puede crear un archivo `.env` para guardar las variables de entorno del cliente. Una vez realizados los pasos anteriores, podremos iniciar nuestra aplicación de React. En nuestro caso estará alojada en el 3000

*npm start*

Una vez realizados estos pasos, tendremos nuestra aplicación MERN stack funcionando perfectamente. Sin embargo, en el caso de e-Change, han sido necesarias dos herramientas adicionales: una plataforma de pago, que es Stripe [4], y una herramienta de almacenamiento de archivos multimedia en la nube.

Para utilizar la plataforma de pago, será necesario crear una cuenta y descargar el CLI (Command Line Interface) que ofrecen en su sitio web. Esto es necesario para que nuestro servidor y la plataforma de pago puedan estar conectados y permitir que, cuando se realice un pago, nuestro servidor backend esté escuchando en el mismo puerto que la plataforma de pago. De este modo, la plataforma podrá enviar un evento de pago a nuestro servidor, ya sea que el pago haya sido validado o no.

Una vez registrados, tendremos que dirigirnos a la sección de "Developers" en el Dashboard y acceder a "API keys". Ahí encontraremos las claves de API públicas y secretas necesarias para integrar la plataforma de Stripe en nuestra aplicación.

Con el servidor detenido, instalaremos el paquete de Stripe en nuestro proyecto Node.js mediante npm.

*npm install stripe*

Como se ha hecho anteriormente, se puede crear un archivo `.env` para guardar las claves API de Stripe [4]. Será necesario configurar el servidor para integrar Stripe, utilizando Express de Node.js para ello.

A continuación, será necesario ejecutar el CLI siguiendo los pasos proporcionados en la página de Stripe. Después de esto, deberás vincular el CLI con tu cuenta de Stripe y configurarlo para el localhost en nuestro caso.

En la terminal, introduce los siguientes comandos:

*stripe login*

El siguiente paso será configurar un endpoint local para escuchar eventos denominados webhooks de Stripe. Para ello, tendremos que configurarlo para que escuche en el mismo puerto que nuestro servidor, en este caso, el puerto 3001.

```
stripe listen --forward-to localhost:3001/webhook
```

Por último, tendremos que crear una cuenta en Cloudfare para poder usar su servicio de nube. Una vez hecho esto, será necesario configurarlo.

Al igual que en los demás servicios, será necesario navegar al Dashboard y recoger el Cloud name, el API Key y el API Secret, credenciales necesarias para la integración en la aplicación.

También será necesario instalar el paquete a través de npm, para ello importante parar el servidor.

```
npm install cloudfare
```

Como se ha hecho anteriormente, se puede crear un archivo .env para guardar las claves API de Cloudfare.

Será necesario configurarlo en el servidor, se empleará Express para su configuración en el backend.

Una vez seguidos todos y cada uno de los pasos descritos en este manual de instalación podremos ejecutar correctamente la aplicación.

## 8.2.- Manual de usuario

Este manual está organizado para atender las necesidades de distintos tipos de usuarios. Ofrece instrucciones claras y detalladas sobre cómo utilizar la aplicación, abarcando todas sus funcionalidades principales. Se incluyen secciones específicas para diferentes roles de usuario anónimo, usuario autenticado, y administradores, proporcionando guías personalizadas que facilitan la comprensión y el uso efectivo de la plataforma. Además, se incorporan ejemplos prácticos para ayudar a los usuarios a resolver cualquier duda que puedan tener durante su interacción con la aplicación.

## 8.2.1.- Usuario anónimo:

Este manual está diseñado para usuarios anónimos de la plataforma e-Change. Como usuario anónimo, puedes consultar objetivos y productos, buscarlos, añadir productos al carrito y a favoritos, así como crear una cuenta o iniciar sesión para acceder a más funcionalidades.

Al ingresar a la aplicación, se te mostrará una ventana con los objetivos destacados. Aquí podrás encontrar aquellos objetivos que tienen mayor índice de dinero recaudado. Estas son probablemente causas que la mayoría de los usuarios están apoyando comprando productos. Te recomiendo que empieces echando un vistazo por esta sección. La interfaz está diseñada en forma de Bento-Box, lo que permite una navegación fácil y ofrece una experiencia inmersiva.

Además, al pasar el ratón sobre alguno de estos objetivos, aparecerá un icono de flecha que indica que puedes hacer clic para navegar hacia él.

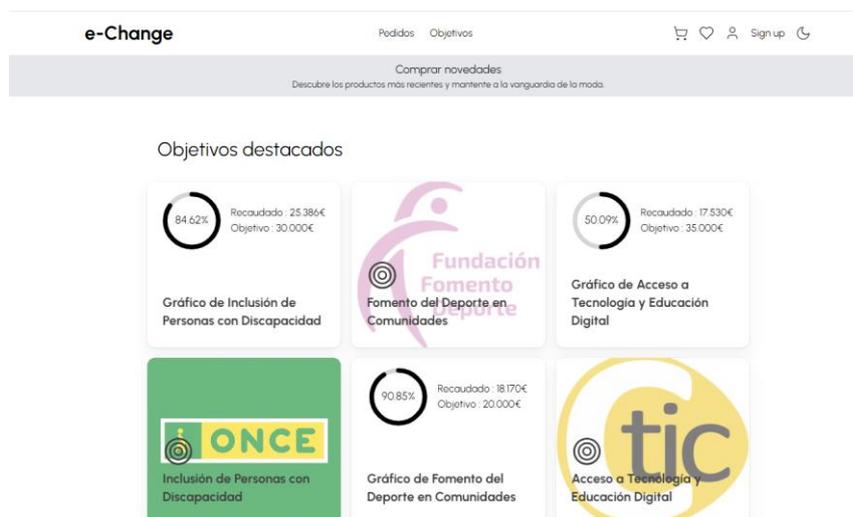


Figura 8.53: Página Inicio Bento-Box

Al seleccionar la opción "Objetivos" en la barra de navegación (Navbar), serás llevado a una página donde se encuentran todos los objetivos y causas sociales disponibles en la plataforma. Puedes realizar una búsqueda utilizando la barra de búsqueda, ingresando cualquier título o descripción que te parezca relevante.

Si deseas apoyar causas recién creadas, puedes filtrar los objetivos a través del menú desplegable, seleccionando aquellos con el menor índice de dinero recaudado.

Además, puedes filtrar por la categoría de producto que ofrecen, lo que te permitirá encontrar el objetivo perfecto que mejor se acomode a tus preferencias y valores.

## Objetivos

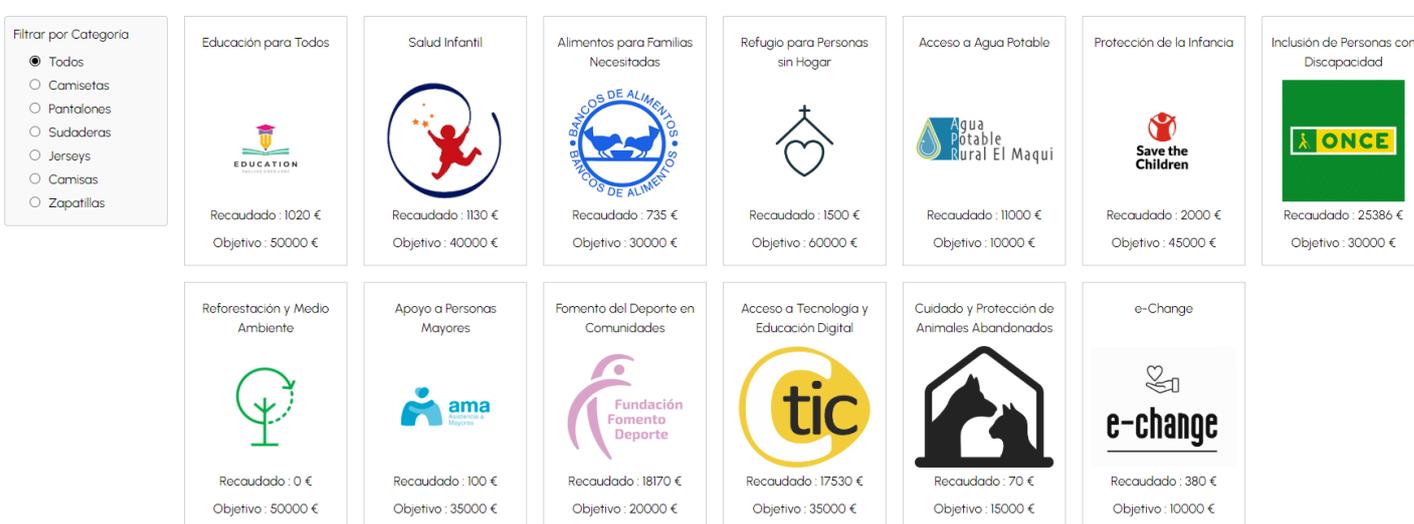


Figura 8.54: Página Objetivos

Si estás interesado en saber más sobre una causa específica, puedes hacer clic en la que más te interese para consultar su información.

Cada vista de objetivo muestra un título sobre la campaña que están realizando y una descripción detallada de la causa, explicando el propósito detrás de la venta de los productos y su objetivo final. Además, se presentan dos atributos importantes: la cantidad de dinero recaudado hasta el momento y la meta de recaudación establecida.

Para una visualización más clara, se incluye un gráfico circular que muestra el progreso de la recaudación, acompañado de un porcentaje que indica cuánto se ha avanzado hacia la meta.

Finalmente, puedes consultar los diferentes productos que cada causa social u objetivo ofrece.



## Educación para Todos

Financiar la construcción y equipamiento de escuelas en comunidades rurales, proporcionando acceso a educación de calidad para niños y niñas en situación de vulnerabilidad. Este proyecto incluye la formación de docentes, la dotación de materiales educativos y la creación de programas de apoyo extracurricular para asegurar que todos los estudiantes tengan las herramientas necesarias para su desarrollo académico y personal.

Recaudado: 1020 €

Objetivo: 50000 €



### Productos



Camiseta "Educación para Todos"  
30 €



Sudadera "Educación para Todos"  
50 €



Camiseta "Aprender para Crecer"  
20 €

Figura 8.55: Página Detalles Objetivo

Al consultar cualquier producto, solo tendrás que hacer clic sobre el que más te interese o necesites. Una vez que hayas hecho clic, se mostrará una ventana con la información detallada del producto, incluyendo:

- 1.- Título:** El nombre del producto.
- 2.- Imagen del producto:** Una foto clara del artículo.
- 3.- Categoría:** La categoría a la que pertenece el producto.
- 4.- Descripción:** Información detallada sobre el producto.

Debajo de esto, encontrarás la talla del producto, seguida del precio.

¿Tiene una idea en mente?

Únete y crea tu primer objetivo ahora para financiar iniciativas a través de la moda.

## dunk low

Categoría: Zapatillas

Las Dunk Low "Hogar y Esperanza" son el calzado perfecto para quienes desean combinar estilo y solidaridad. Confeccionadas con materiales sostenibles y un diseño ergonómico, estas zapatillas ofrecen una comodidad superior y un estilo atemporal. Su diseño exclusivo, que incluye detalles únicos que simbolizan el refugio y la esperanza, convierte a estas zapatillas en una declaración de moda con un propósito profundo. Cada compra de estas Dunk Low contribuye directamente a la construcción y mantenimiento de albergues que proporcionan refugio seguro y servicios de apoyo a personas sin hogar. Estos albergues ofrecen atención médica, apoyo psicológico, capacitación laboral y programas de reintegración social, ayudando a las personas sin hogar a reintegrarse en la sociedad de manera digna e independiente. Al elegir estas zapatillas, no solo caminas con estilo, sino que también das pasos concretos para mejorar la vida de quienes más lo necesitan.



Talla: 45

120 €

Añadir al Carrito



Figura 8.56: Página Producto

Si, después de consultar la información del producto, te parece relevante, puedes hacer clic en el ícono del corazón para añadirlo a la sección de favoritos. Si deseas comprarlo directamente, puedes añadirlo al carrito.

 Producto añadido a la lista 

Figura 8.57: Toast Añadir a Favoritos

 Producto añadido al carrito 

Figura 8.58: Toast Añadir al Carrito

Si nos fijamos, podemos apreciar que la barra de navegación (Navbar) ha cambiado. Esto significa que la sección de favoritos ahora tiene contenido. Además, se muestra el número de productos que has añadido al carrito.



Figura 8.59: Barra de Navegación con contenido

Si consultamos la sección de favoritos, veremos el producto que se había añadido anteriormente, junto con varios botones e información adicional. Aparecerán detalles importantes del producto, como el nombre, la categoría, la imagen, la talla y el precio. Además, se mostrarán dos botones para realizar acciones adicionales.

### Productos que te gustan



**dunk low**

Zapatillas

Talla: 45

120 €

Añadir al Carrito



Figura 8.60: Pestaña Favoritos con contenido

El botón en forma de X nos permite eliminar el producto de la sección de favoritos.

! Producto eliminado de la lista

Figura 8.61: Toast de eliminación de producto de Favoritos

e-Change

Pedidos Objetivos

🛒 ❤️ 👤 Sign up 🔄

Devoluciones y entregas gratuitas

Disfruta de 30 días para realizar devoluciones desde la fecha de compra. ¡Compra con confianza y comodidad!

Productos que te gustan

No hay productos en la sección de Favoritos

Figura 8.62: Pestaña Favoritos sin contenido

Por otro lado, el botón "Añadir al carrito" nos permite añadir el producto al carrito de la compra.

! Producto eliminado de la lista

✔️ Producto añadido al carrito 🛒

Figura 8.63: Toast indicando que el producto se ha añadido al Carrito

Una vez añadidos los productos al carrito, podemos consultar su contenido haciendo clic en el ícono del carrito de compras con el número de artículos. Esto nos mostrará una lista de los productos que hemos añadido al carrito, incluyendo información relevante como el nombre, imagen, categoría, talla y precio, acompañada de un botón para realizar acciones adicionales.

Además, se muestra un subtotal indicando el precio total del contenido del carrito.

## Carrito

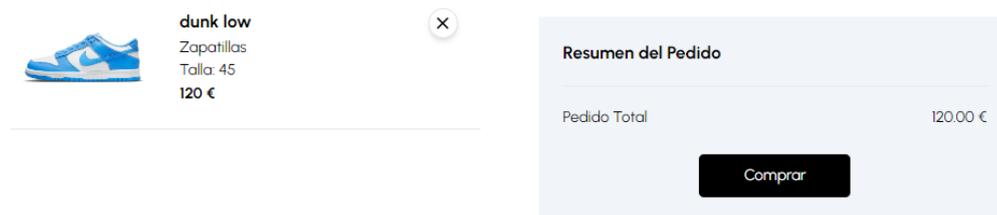


Figura 8.64: Pestaña Carrito con Productos

El botón en forma de X nos permite eliminar el producto del contenido del carrito.

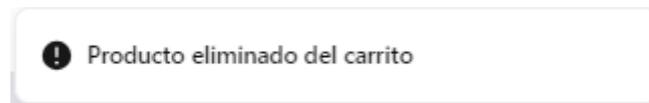


Figura 8.65: Toast eliminación del producto del Carrito

## Carrito

No hay productos en el carrito

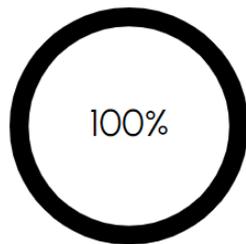


Figura 8.66: Pestaña Carrito sin Productos

Por último, para poder realizar una compra de un producto, será necesario crear una cuenta en la plataforma. Para ello, selecciona el ícono que dice "Sign Up". Al hacer clic, serás redirigido a la página de registro. Completa el formulario de registro con los datos solicitados de forma válida. Mientras introduces tu contraseña, puedes hacer clic en el ícono del ojo para ver la contraseña ingresada y asegurarte de que es correcta. Una vez completados todos los campos, haz clic en el botón "Registrar". Si no existe ninguna cuenta con la misma dirección de correo electrónico, el sistema creará tu cuenta. Ahora estarás listo para realizar compras en la plataforma.

También se te ofrece la posibilidad de acceder mediante el botón iniciar que se encuentra debajo en caso de ya tener una cuenta en la plataforma registrada.

e-Change.



€1.468,00

### Registrarse

Usuario:

Correo Electrónico:

Contraseña:

Confirmar Contraseña:

Registrar

[¿Ya tienes cuenta? Inicia sesión aquí](#)

Figura 8.67: Página Sign up

Una vez creada la cuenta, podrás acceder fácilmente haciendo clic en el ícono de la persona. Esto te redirigirá a la página de inicio de sesión en la plataforma. Solo necesitarás introducir los datos de forma correcta y hacer clic en el botón "Iniciar Sesión". Mientras introduces tu contraseña, puedes hacer clic en el ícono del ojo para ver la contraseña ingresada y asegurarte de que es correcta. Una vez hecho esto, habrás iniciado sesión en la plataforma y serás un usuario autenticado.

buy.

100%

€1.468.00

Iniciar Sesión

Correo Electrónico:

Contraseña:

Iniciar Sesión

¿No tienes cuenta? [Regístrate aquí](#)

Figura 8.68: Página Iniciar Sesión

## 8.2.2.- Usuario autenticado:

Este manual está diseñado para usuarios autenticados de la plataforma e-Change. Como usuario autenticado, puedes consultar objetivos y productos, buscarlos, añadir productos al carrito y a favoritos, al igual que un usuario anónimo. Sin embargo, como usuario autenticado, tienes la posibilidad adicional de crear, editar y eliminar objetivos. Además, puedes añadir y eliminar productos en los objetivos tantas veces como desees. Por último, puedes realizar la compra de productos para apoyar causas sociales y seguir un historial de tus pedidos a través del apartado "Pedidos".

Desde el momento en que inicias sesión, eres un usuario autenticado en la plataforma, y puedes comenzar a realizar los primeros pasos para crear tus objetivos y apoyar a causas. Para hacerlo, simplemente navega a la pestaña de "Objetivos" a través de su nombre en la barra de navegación (Navbar).

Al navegar a esta ventana, verás que aparecen dos botones que antes no estaban disponibles. El primero te dará la opción de crear nuevos objetivos, y el segundo te redirigirá al apartado "Mis objetivos", donde podrás consultar los objetivos que ya has creado y realizar diferentes acciones con ellos.

Crear Objetivo
Mis Objetivos

## Objetivos

Sin orden ▼

Filtrar por Categoría

- Todos
- Camisetas
- Pantalones
- Sudaderas
- Jerseys
- Camisas
- Zapatillas

Educación para Todos



Recaudado : 1020 €

Objetivo : 50000 €

Salud Infantil



Recaudado : 1130 €

Objetivo : 40000 €

Alimentos para Familias Necesitadas



Recaudado : 735 €

Objetivo : 30000 €

*Figura 8.69: Página Objetivos Usuario Autenticado*

Si hacemos clic en el primer botón, seremos redirigidos a la página de creación de un objetivo, donde podrás rellenar un formulario para crear un nuevo objetivo. En este formulario, puedes introducir un título que indique la causa por la cual estás creando el objetivo, acompañado de una descripción detallada. También tienes la opción de subir una imagen para complementar y hacer más llamativo lo que persigues o defiendes, como, por ejemplo, a través de un logo. Por último, puedes introducir la cantidad que deseas recaudar como objetivo. Una vez completado el formulario al pulsar el botón Crear Objetivo, el sistema te guardará un registro del objetivo que has creado.

### Crear Nuevo Objetivo

**Título**

**Descripción**

Descripción

0/500

**Imagen**

Seleccionar archivo Ningún archivo seleccionado

**Cantidad objetivo a recaudar (€)**

Crear Objetivo

*Figura 8.70: Página Creación de Objetivos*

Por otro lado, la opción "Mis Objetivos" te permite navegar a una ventana donde se muestran todos tus objetivos. Desde esta ventana, puedes realizar varias acciones, como editar un objetivo o eliminarlo.

### Mis Objetivos

The screenshot displays three goal cards under the heading "Mis Objetivos". Each card contains a title, a logo, current and target amounts, and "Editar" and "Eliminar" buttons.

Objetivo	Recaudado	Objetivo
Acceso a Tecnología y Educación Digital (tic)	17530 €	35000 €
Cuidado y Protección de Animales Abandonados	70 €	15000 €
e-Change	500 €	10000 €

Figura 8.71: Página Objetivos Usuario Autenticado

Si deseas editar un objetivo, simplemente haz clic en "Editar" sobre el objetivo que quieras modificar. Se mostrará un formulario similar al de creación, donde podrás editar algunos datos del objetivo que habías creado, como el título, la descripción y la cantidad objetivo. Del mismo modo, tendrás la opción de añadir productos y eliminar productos asociados al objetivo.

The screenshot shows the "Editar Objetivo" form for the "e-Change" goal. It includes buttons for "Agregar Producto" and "Eliminar Productos", input fields for "Titulo" and "Monto Objetivo", and a text area for "Descripción".

**Editar Objetivo** [Agregar Producto] [Eliminar Productos]

**Titulo:**  
e-Change

**Descripción:**  
Desarrollar una plataforma digital que recopile y promueva diversas causas sociales, permitiendo a los usuarios descubrir y apoyar proyectos de impacto positivo en sus comunidades y alrededor del mundo. La plataforma facilitará la donación de fondos, el voluntariado y la colaboración entre organizaciones, ofreciendo herramientas de seguimiento y transparencia para asegurar que las contribuciones lleguen a su destino y se utilicen de manera efectiva. Lo recaudado se destinará para mejorar la web.

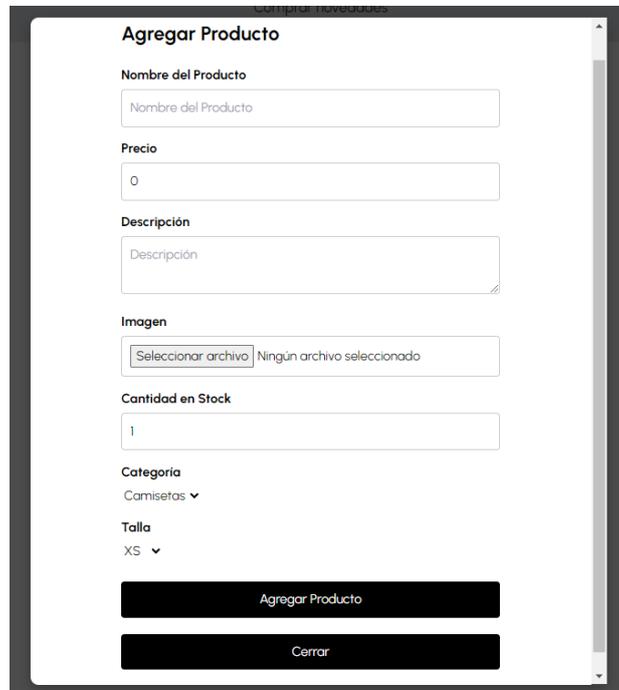
500 / 500

**Monto Objetivo:**  
10000

[Guardar Cambios]

Figura 8.72: Modificación de Objetivos

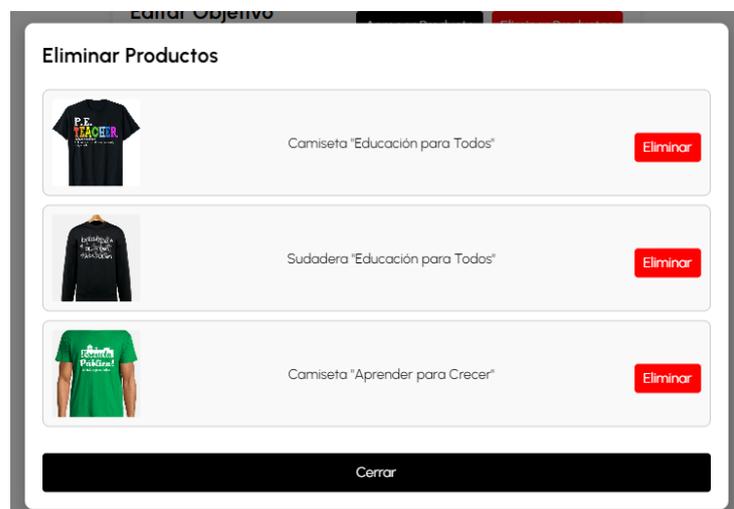
Para añadir un producto, simplemente haz clic en el botón "Agregar Producto". Se abrirá un modal que te permitirá introducir varios datos sobre el producto, incluyendo el nombre, el precio, una descripción detallada sobre por qué estás vendiendo este producto, una imagen, la cantidad en stock disponible y la categoría. También podrás seleccionar la talla, que será una lista desplegable dependiente de la categoría. Una vez completados todos los campos, puedes agregar el producto haciendo clic en "Agregar" o cerrar el modal si decides no añadirlo.



The image shows a modal window titled "Agregar Producto". It contains several input fields: "Nombre del Producto" (text input), "Precio" (text input with "0" entered), "Descripción" (text area), "Imagen" (file selection button showing "Seleccionar archivo | Ningún archivo seleccionado"), "Cantidad en Stock" (text input with "1" entered), "Categoría" (dropdown menu showing "Camisetas"), and "Talla" (dropdown menu showing "XS"). At the bottom, there are two buttons: "Agregar Producto" and "Cerrar".

Figura 8.73: Añadir Producto

En cambio, si lo que deseas es eliminar un producto, simplemente haz clic en el botón "Eliminar Producto". Aparecerá un modal con la lista de productos que contiene el objetivo, mostrando la imagen y el nombre de cada producto, junto con un botón para eliminar el producto en cuestión.



The image shows a modal window titled "Eliminar Productos". It displays a list of three items, each with a small image, a name, and an "Eliminar" button. The items are: "Camiseta 'Educación para Todos'" (black t-shirt), "Sudadera 'Educación para Todos'" (black hoodie), and "Camiseta 'Aprender para Crecer'" (green t-shirt). At the bottom, there is a "Cerrar" button.

Figura 8.74: Eliminar Producto

Como usuario autenticado, además de tener la posibilidad de crear y gestionar tus propios objetivos, también puedes apoyar las causas de otros usuarios comprando sus productos. Para hacerlo, simplemente añade al carrito el producto que te interese, ya sea por sus características o por la causa social a la que apoya. Una vez en el carrito, haz clic en "Comprar", lo cual te llevará a la pasarela de pago. Aquí podrás introducir los datos necesarios tanto de envío como de pago, permitiéndote realizar la compra de forma segura.

The screenshot shows a checkout page for 'Pantalones Two Jeys' priced at 70,00 €. The product image shows light blue jeans with two black stars on the back pocket. The payment form is titled 'Pago con tarjeta' and includes sections for 'Información de envío' and 'Datos del pago'. The 'Información de envío' section has fields for 'Correo electrónico', 'Dirección de envío' (Nombre completo, España, Introduce dirección), and a link to 'Introduce la dirección de forma manual'. The 'Datos del pago' section has fields for 'Información de la tarjeta' (1234 1234 1234 1234, MM / AA, CVC) and a checkbox for 'La información de facturación es la misma que la de envío.' A blue 'Pagar' button is at the bottom.

Figura 8.75: Compra de Producto

Una vez introducidos los datos de envío y de pago, se procesará la compra dependiendo de la validación de estos datos. Si la validación es exitosa, serás redirigido a una ventana donde se te agradecerá la compra y se te informará que se procederá a preparar el envío.

The screenshot shows the same checkout page as Figure 8.75, but with the form fields filled in. The 'Correo electrónico' field contains 'sergio.fagundez.m@gmail.com'. The 'Dirección de envío' fields contain 'Sergio', 'España', 'Plaza Azoguejo 1', 'Línea 2 de dirección', '40001', 'Segovia', and 'Segovia'. The 'Datos del pago' fields contain '4242 4242 4242 4242', '07 / 25', and '123'. The 'Pagar' button is now green with a white checkmark icon.

Figura 8.76: Validación de Compra



Figura 8.77: Compra exitosa

De lo contrario se mostrará un error indicando que la compra no se ha podido realizar.

## ¡Error al realizar tu compra!



Figura 8.78: Compra Fallida

Para finalizar, puedes consultar tu pedido a través de la opción "Pedidos" en la barra de navegación (Navbar). Se mostrará un historial de todos los pedidos de productos que has realizado. Podrás acceder al detalle del producto a través de su imagen. También podrás ver varios datos sobre el pedido, como el nombre del producto, la cantidad, el precio, la talla, la fecha en la que se realizó la compra y la dirección de envío.

### Historial de Pedidos

<p><b>Pantalones Two Jeys</b> Cantidad: 1 Precio: 70 € Talla: L Fecha: 12/7/2024, 23:49:21 Dirección de envío: Plaza Azoguejo 1 Segovia, SG 40001 ES</p>	
<p><b>Zapatillas "Camino a la Salud"</b> Cantidad: 1 Precio: 80 € Talla: 36 Fecha: 10/7/2024, 20:19:29 Dirección de envío: Puente de Segovia S/N Madrid, M 28005 ES</p>	
<p><b>Zapatillas de Fútbol</b> Cantidad: 1 Precio: 120 € Talla: 39 Fecha: 10/7/2024, 20:18:08 Dirección de envío: Avenida de la Hispanidad s/n Madrid, M 28042 ES</p>	

Figura 8.79: Pedidos

Como cualquier cuenta en la actualidad, se permite al usuario gestionar cierta información de su perfil, permitiéndole modificar algunos datos como la contraseña y su nombre de usuario. No es necesario realizar ambas modificaciones; el usuario puede optar por cambiar solo uno de estos datos según lo necesite.



Figura 8.80: Perfil

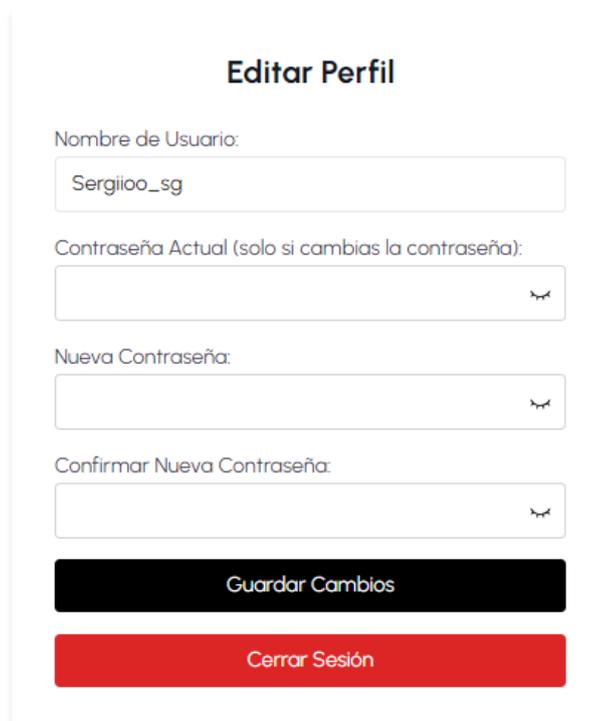
A screenshot of the "Editar Perfil" form. The title "Editar Perfil" is centered at the top. Below the title are four input fields: "Nombre de Usuario:" with the value "Sergioo\_sg"; "Contraseña Actual (solo si cambias la contraseña):"; "Nueva Contraseña:"; and "Confirmar Nueva Contraseña:". Each password field has a small eye icon on the right side. At the bottom of the form are two buttons: a black button labeled "Guardar Cambios" and a red button labeled "Cerrar Sesión".

Figura 8.81: Editar Perfil

## 8.2.1.- Administrador

Este manual está diseñado para administradores de la plataforma e-Change. Como administrador, puedes consultar objetivos y productos, buscarlos, añadir productos al carrito y a favoritos, al igual que un usuario anónimo. Además, al igual que un usuario autenticado, tienes la posibilidad de crear, editar y eliminar objetivos, así como añadir y eliminar productos en los objetivos tantas veces como desees. También puedes realizar la compra de productos para apoyar causas sociales y seguir un historial de tus pedidos a través del apartado "Pedidos".

Como diferencia principal, el administrador tiene acceso a un panel de control en el que puede consultar todos los objetivos disponibles en la plataforma, permitiendo así un control completo sobre ellos y asegurando un manejo más profesional.

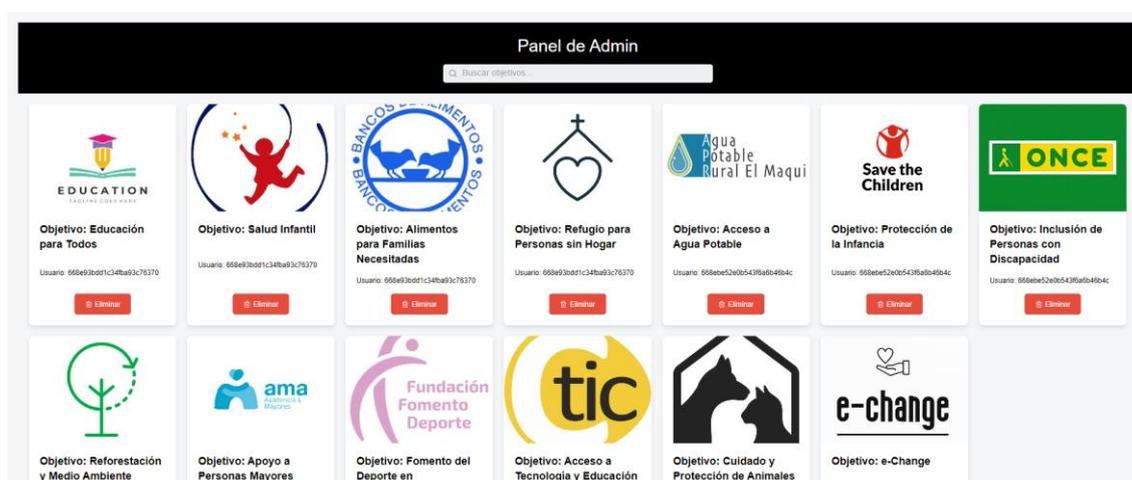


Figura 8.82: Panel de Administrador

Este panel de administrador te permite buscar objetivos específicos a través de una barra de búsqueda. Además, puedes ver el ID del usuario creador del objetivo para identificar quién lo ha creado. Por último, si pulsas el botón de eliminar, el objetivo será eliminado de la plataforma.



# Capítulo 9

## Conclusiones y trabajo futuro

En este apartado, se presentan las conclusiones obtenidas tras el desarrollo de este trabajo de fin de grado, así como una serie de posibles mejoras y ampliaciones que podrían incorporarse en el futuro a la plataforma.

### 9.1.- Conclusiones

El desarrollo de esta plataforma ha permitido alcanzar varios objetivos importantes. Aunque ha habido situaciones más difíciles que otras, siento que, una vez finalizado el proyecto, se puede comprobar que se han cumplido todos y cada uno de los objetivos planteados desde un inicio.

Estoy especialmente orgulloso del proyecto, ya que, a pesar de haber utilizado tecnologías que no había usado anteriormente, he logrado desenvolverme y descubrir que, con dedicación y esfuerzo, es posible alcanzar metas que inicialmente parecían imposibles. Este proceso me ha enseñado que, con tiempo y dedicación, se pueden superar los desafíos más complejos.

También estoy muy satisfecho con el resultado de la interfaz de la aplicación. La encuentro moderna y atractiva, comparable con muchas de las páginas web de gran popularidad que existen actualmente. La usabilidad y el diseño visual de la plataforma son aspectos que considero destacados y bien logrados.

Este proyecto no solo me ha permitido desarrollarme profesionalmente, sino también personal y competentemente. La idea de poner a disposición de los demás una herramienta que puede tener un impacto positivo en la sociedad es muy gratificante. Creo firmemente que esta aplicación, con más tiempo y mejoras, podría llegar a ser una herramienta útil en la vida diaria de muchas personas.

La iniciativa de apoyar causas sociales mediante la compra de productos es innovadora y tiene un gran potencial. Esta plataforma ofrece una forma interesante y significativa de contribuir a proyectos y causas importantes, haciendo que las compras tengan un impacto más allá del consumo personal.

En resumen, este proyecto ha sido una experiencia enriquecedora que me ha permitido crecer tanto técnica como personalmente. Estoy convencido de que, con mejoras continuas, la plataforma podría desempeñar un papel importante en nuestra sociedad, promoviendo un consumo más consciente y solidario.

## 9.2.- Trabajo futuro

Aunque se han logrado muchos objetivos, hay varias áreas en las que se podrían realizar mejoras y ampliaciones para incrementar el valor y la funcionalidad de la plataforma.

### 9.2.1.- Sistema de notificaciones

Hoy en día, con el continuo avance de internet, se descubren nuevas posibilidades que esta herramienta puede ofrecer. Cada vez más, el uso de internet permite a las personas crear y emplearlo en su día a día, ayudándolas en muchos aspectos. Sin embargo, también conlleva un lado oscuro: aquellos que lo usan para hacer el mal. Por ello, implementar un sistema de notificaciones que avise al administrador cuando se cree o modifique un objetivo nos permitiría tener un mayor control y seguridad sobre el contenido que se sube a la plataforma.

### 9.2.2.- Validación por admin

Al igual que la mejora anterior, podría incorporarse una opción para que el administrador pueda validar los objetivos antes de ser publicados. Esto implicaría la incorporación de varios administradores que estarían revisando continuamente cada uno de estos objetivos.

### 9.2.3.- Gestión de envíos

Al ser un comercio electrónico, se ofrece la posibilidad de comprar productos. Por lo tanto, sería ideal implementar un sistema encargado de realizar los envíos de los productos para que lleguen al usuario. Aunque esta funcionalidad no se contempla dentro del TFG debido a su duración, sería una mejora valiosa para desarrollar en el futuro. Esta implementación solo requeriría gestionar la logística y enviar los productos a la dirección de entrega proporcionada en la plataforma al realizar el pago.

### 9.2.4.- Remitir dinero

Otra mejora futura sería la implementación de un sistema encargado de remitir el dinero recaudado a cada organización. Actualmente, soy yo quien se encarga de transferir el dinero a través de la pasarela de pago a cada organización, fuera de la plataforma.

Para automatizar este proceso, se podría implementar un método en el cual cada organización pueda introducir su número de cuenta, y el dinero recaudado se transfiera mensualmente a dicha cuenta. También existiría la posibilidad de transferir el dinero directamente en caso de que el objetivo haya sido completado. Si la recaudación continúa después de alcanzar el objetivo, volveríamos al método de transferencias mensuales.

## 9.2.5.- Productos

Aunque mi aplicación permite gestionar los objetivos y productos, permitiendo crear, editar e incluso eliminar objetivos, los productos solo se pueden crear o eliminar. Por lo tanto, sería beneficioso implementar un método que permita a los usuarios editar los productos.

Además, sería útil implementar un sistema de tallas que permita comprar diferentes tallas de un producto. Actualmente, los productos se contemplan como prendas únicas, por lo que solo están disponibles en una talla y el stock indicado por el usuario.

En cuanto al stock, se podría implementar un sistema que, cuando un producto se agote, permita a los usuarios introducir su correo electrónico o suscribirse para recibir notificaciones cuando se reponga el stock del producto en la talla deseada. Esto aseguraría que los usuarios se mantengan informados y puedan adquirir los productos que necesitan en el momento en que estén disponibles nuevamente.

## 9.2.6.- Redes sociales

Las redes sociales se utilizan continuamente en nuestro día a día; nuestra sociedad vive en un mundo centrado en ellas. Por lo tanto, introducir esta funcionalidad en la plataforma permitiría una mayor visibilidad y ofrecería una relación más cercana con el usuario que compra los productos. Facilitar el acceso a las redes sociales a través de publicaciones, enlaces y perfiles de los creadores de los objetivos no solo mejoraría la experiencia del usuario, sino que también podría aumentar el compromiso y la interacción con la plataforma.

Además, integrar redes sociales permitiría a los usuarios compartir sus objetivos y productos favoritos con sus propias redes, lo que podría atraer a más usuarios a la plataforma y aumentar el alcance de las causas sociales promovidas. También sería beneficioso implementar funciones como comentarios y reseñas a través de redes sociales, lo que proporcionaría feedback valioso a los creadores y fomentaría una comunidad más activa y participativa.

## 9.2.7.- Aplicación móvil

Aunque la plataforma sea responsive, disponer de una aplicación móvil lograría expandir aún más su uso. Uno de los objetivos principales es llegar a la sociedad, por lo que cuantas más personas puedan utilizarla, mejor será. De este modo, se permitiría realizar funcionalidades adicionales, como subir fotos directamente desde la cámara del dispositivo y un sistema de notificaciones para seguir más de cerca cada uno de los objetivos. Una aplicación móvil podría proporcionar una experiencia de usuario más fluida e intuitiva, optimizada específicamente para dispositivos móviles. Esto incluiría la capacidad de recibir notificaciones en tiempo real sobre nuevas oportunidades de compra, actualizaciones de objetivos y alertas de reposición de stock. Además, una aplicación móvil podría aprovechar características nativas del dispositivo, como la geolocalización, para ofrecer recomendaciones personalizadas y mejorar la interacción del usuario con la plataforma.



# Capítulo 10

## Webgrafía

- [1] E-Commerce:  
<https://ticnegocios.camaravalencia.com/servicios/tendencias/el-ano-del-coronavirus-en-el-que-el-ecommerce-cambio-radicalmente-vii-estudio-anual-de-ecommerce-en-espana-2020/>  
Último acceso: Junio 2024
- [2] Node.js:  
<https://nodejs.org/en>  
Último acceso: Junio 2024
- [3] Cloudinary:  
<https://cloudinary.com/>  
Último acceso: Junio 2024
- [4] Stripe:  
<https://docs.stripe.com/development>  
Último acceso: Julio 2024
- [5] MVC:  
<https://developer.mozilla.org/es/docs/Glossary/MVC>  
Último acceso: Julio 2024
- [6] MagicUI:  
<https://magicui.design/docs/>  
Último acceso: Mayo 2024
- [7] UI:  
<https://ui.shadcn.com/docs/>  
Último acceso: Mayo 2024
- [8] MongoDB:  
<https://www.mongodb.com/products/platform/atlas-database>  
Último acceso: Julio 2024

- [9] MERN:  
<https://www.youtube.com/watch?v=DqpL5UtJHus&list=LL&index=29&t=2726s>  
Último acceso: Julio 2024
- [10] Cloudinary upload image:  
<https://www.youtube.com/watch?v=AGMnqfDeDIY&list=LL&index=2&t=4998s>  
Último acceso: Julio 2024
- [11] Vercel:  
<https://vercel.com/new>  
Último acceso: Julio 2024
- [12] MERN stack:  
<https://openwebinars.net/blog/mern-stack-que-es-y-que-ventajas-ofrece/>  
Último acceso: Abril 2024
- [13] Vinted:  
<https://www.vinted.es/>  
Último acceso: Julio 2024
- [14] Etsy:  
<https://www.etsy.com/es/?ref=lgo>  
Último acceso: Julio 2024
- [15] Change.org:  
<https://www.change.org/>  
Último acceso: Julio 2024
- [16] React:  
<https://es.react.dev/learn>  
Último acceso: Mayo 2024
- [17] Github:  
<https://github.com/>  
Último acceso: Julio 2024
- [18] Npm:  
<https://www.npmjs.com/>  
Último acceso: Julio 2024
- [19] Express:  
<https://expressjs.com/es/>  
Último acceso: Junio 2024