



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

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

Wallaorg. Diseño de una aplicación solidaria.

Alumna: Eva Máiquez Pérez

Tutora: Amelia García Garrosa

Wallaorg. Diseño de una aplicación solidaria.

Eva Máiquez Pérez

Contenido

Lista de figuras	III
Lista de tablas	V
Resumen	IX
Abstract	XI
1 Descripción del proyecto	1
1.1 Introducción	1
1.2 Objetivos del trabajo	2
1.3 Entorno de aplicación	2
1.4 Estructura de la memoria	4
2 Metodología	7
2.1 Proceso de desarrollo	7
2.2 Herramientas utilizadas	9
2.3 Arquitectura	11
3 Planificación	13
3.1 Planificación inicial	13
3.1.1 Planificación temporal	13
3.1.2 Presupuesto económico	15
3.2 Planificación según estimación de esfuerzo	16
3.2.1 Estimación de esfuerzo	16
3.2.2 Planificación temporal	19
3.2.3 Presupuesto económico	20
3.3 Comparativa entre planificaciones y seguimiento real	21
4 Análisis	23
4.1 Actores	23
4.2 Requisitos	23
4.3 Casos de uso	26
4.4 Atributos de calidad	40

5	Diseño	41
5.1	Diseño de datos	41
5.1.1	Modelo entidad-relación	41
5.1.2	Modelo relacional	41
5.1.3	Diccionario de datos	44
5.2	Diagramas de clase y de secuencia	45
5.2.1	Diagrama de clases	45
5.2.2	Diagramas de secuencia	45
6	Implementación	47
7	Pruebas	53
7.1	Caja blanca	53
7.2	Caja negra	53
8	Manuales de instalación y de usuario	57
8.1	Manual de instalación	57
8.2	Manual de usuario	58
8.2.1	Acceder a la web	59
8.2.2	Registrarse	59
8.2.3	Iniciar sesión	60
8.2.4	Ver artículos disponibles	60
8.2.5	Donar artículos	61
8.2.6	Comprar	61
8.2.7	Pujar en subasta directa	62
8.2.8	Ver y modificar perfil	62
8.3	Manual de administrador	63
8.3.1	Publicar artículo	63
8.3.2	Modificar artículos publicados	64
8.3.3	Ver y modificar listado de usuarios	64
8.3.4	Ver pedidos	65
9	Conclusiones y líneas futuras	67
Anexo		69
A.1	Diccionarios de datos	69
A.2	Diagramas de secuencia	77
A.3	Pruebas caja negra	84
Bibliografía		89

Lista de figuras

2.1	Tablero Trello Sprint Usuarios y artículos	7
2.2	Arquitectura del patrón Modelo-Vista-Controlador	8
2.3	Arquitectura lógica	12
2.4	Arquitectura física	12
3.1	Diagrama de Gantt inicial	14
3.2	Diagrama de Gantt según estimación	19
4.1	Diagrama casos de uso	26
5.1	Diagrama Entidad-Relación	42
5.2	diagrama de secuencia: cerrar sesión	46
5.3	diagrama de secuencia: donar	46
6.1	Estructura carpetas raiz	49
6.2	Controladores	49
6.3	Migraciones	50
6.4	Modelos	50
6.5	Rutas	50
6.6	Vistas	51
6.7	Usuarios	51
8.1	XAMPP	58
8.2	Importar base de datos	58
8.3	Open Folder Visual Studio Code	58
8.4	New Terminal Visual Studio Code	59
8.5	npm install Visual Studio Code	59
8.6	npm start Visual Studio Code	59
A.1	diagrama de secuencia: eliminar artículo	77
A.2	diagrama de secuencia: eliminar usuario	77
A.3	diagrama de secuencia: eliminar donación	78
A.4	diagrama de secuencia: iniciar sesión	78
A.5	diagrama de secuencia: modificar artículo	79
A.6	diagrama de secuencia: modificar usuario	80
A.7	diagrama de secuencia: publicar artículo	81

Lista de figuras

A.8 diagrama de secuencia: publicar donación	81
A.9 diagrama de secuencia: pujar artículo	81
A.10 diagrama de secuencia: realizar pedido	82
A.11 diagrama de secuencia: registrarse	82
A.12 diagrama de secuencia: ver donaciones	83

Lista de tablas

1.1	Estado del arte	4
3.1	Planificación temporal inicial	14
3.2	Presupuesto hardware inicial	15
3.3	Presupuesto software	15
3.4	Presupuesto recursos humanos inicial	16
3.5	Presupuesto total inicial	16
3.6	Factor ambiental	18
3.7	Planificación temporal según estimación	19
3.8	Presupuesto hardware según estimación	20
3.9	Presupuesto recursos humanos según estimación de tiempo	20
3.10	Presupuesto total según estimación	20
3.11	Comparativa planificación temporal	21
3.12	Comparativa presupuestos	21
4.1	ACTORES	23
4.2	Requisitos de negocio	24
4.3	Reglas de negocio	24
4.4	Requisitos de usuario	25
4.5	Requisitos funcionales	25
4.6	Requisitos de información	26
4.7	CU-01. Registrarse	28
4.8	CU-02. Iniciar sesión	28
4.9	CU-03. Cerrar sesión	29
4.10	CU-04. Usuario modificar perfil	32
4.11	CU-05. Administrador publicar artículos	33
4.12	CU-06. Administrador modificar o eliminar artículos	34
4.13	CU-07. Donar artículo	35
4.14	CU-08. Ver donaciones	36
4.15	CU-09. Comprar artículo	39
4.16	CU-10. Pujar artículo	39
4.17	Atributos de calidad	40
5.1	Modelo relacional	43
5.2	Diccionario de datos: COMPRAS	44

Lista de tablas

5.3	Diccionario de datos: CONTENER	45
7.1	PCN-01 Registrarse	54
7.2	PCN-02 Iniciar sesión	55
7.3	PCN-03 Cerrar sesión	55
A.1.1	Diccionario de datos: FOTOS	69
A.1.2	Diccionario de datos: ARTICULOS	70
A.1.3	Diccionario de datos: DIRECCIONS	71
A.1.4	Diccionario de datos: DONACIONES	72
A.1.5	Diccionario de datos: USUARIOS	73
A.1.6	Diccionario de datos: CONVERTIRSE	73
A.1.7	Diccionario de datos: MANDARSE	74
A.1.8	Diccionario de datos: PUBLICAR	74
A.1.9	Diccionario de datos: PUJAR	74
A.1.10	Diccionario de datos: QUERER	74
A.1.11	Diccionario de datos: REALIZAR1	75
A.1.12	Diccionario de datos: REALIZAR2	75
A.1.13	Diccionario de datos: TENER1	75
A.1.14	Diccionario de datos: TENER3	75
A.1.15	Diccionario de datos: TENER4	76
A.3.1	PCN-04 Modificar perfil	84
A.3.2	PCN-05 Publicar artículos	85
A.3.3	PCN-06 Modificar o eliminar artículos	85
A.3.4	PCN-07 Donar artículos	86
A.3.5	PCN-08 Publicar donaciones	86
A.3.6	PCN-09 Comprar artículo	87
A.3.7	PCN-10 Pujar artículo	88

Agradecimientos

Quiero expresar mi más profundo agradecimiento a todas aquellas personas que han hecho posible la realización de este Trabajo de Fin de Grado.

En primer lugar, a mi novio, cuya paciencia, amor y apoyo incondicional han sido una fuente constante de motivación y fortaleza. Su presencia ha sido indispensable en cada etapa de este proceso, brindándome el aliento necesario para superar los desafíos y seguir adelante.

A mi familia, por su apoyo incondicional y su fe en mis capacidades. Gracias por creer en mí y por ofrecerme siempre el apoyo emocional y logístico que he necesitado. Vuestra comprensión y ánimo han sido fundamentales para alcanzar este logro.

Finalmente, a mi tutora, cuyo esfuerzo, dedicación y valiosa ayuda han sido clave en la realización de este proyecto. Tu orientación y tus consejos han sido invaluable, permitiéndome mejorar y perfeccionar mi trabajo. Gracias por tu tiempo y por compartir tu conocimiento conmigo.

A todos vosotros, mi más sincero agradecimiento.

Resumen

Este trabajo fin de grado desarrolla una aplicación para financiar organizaciones benéficas mediante un sistema de comercio electrónico. La originalidad del trabajo radica en su orientación hacia este tipo de entidades, que conlleva ciertas particularidades.

En primer lugar, debemos considerar el origen de los productos vendidos. En la mayoría de los casos, procederán de donaciones o serán fabricados por la propia organización, pero generalmente serán objetos de poco valor. Esto conlleva ya dos requisitos específicos, por un lado, la aplicación debe contemplar la funcionalidad de admitir donaciones, tanto económicas como de objetos y, por otro, debe contemplar sistemas de venta específicos que permitan obtener algún rendimiento de esos objetos poco valiosos.

En segundo lugar, debemos tener en cuenta que la actividad principal de la entidad no es el comercio, lo que supone que generalmente no dispondrá de almacenes u otros servicios logísticos y tendrá que dar una salida rápida a todos sus productos y rentabilizarlos rápidamente, aunque no sea al precio óptimo. Este hecho, unido al ya comentado del poco valor de sus productos ha llevado a considerar tres tipos de venta diferentes. La venta directa, la subasta y la subasta inversa.

La venta directa es el sistema habitual en que cada bien tiene un precio fijo marcado. Es aplicable a aquellos productos que tienen un precio de mercado bien definido. Bastará con fijar un precio ligeramente inferior, para conseguir el objetivo de una venta rápida.

La subasta es aplicable a aquellos productos que no disponen de un precio claro de mercado porque la oferta es pequeña. Al aumentar el precio de venta con cada puja, se puede obtener un alto rendimiento económico, pues un potencial comprador podría estar dispuesto a pagar un precio elevado para aprovechar la oportunidad de obtener un producto poco común.

Por último, la subasta inversa se aplica a los productos de poco valor o excesiva oferta. Se parte de un precio relativamente elevado y se va bajando hasta que aparece algún comprador interesado. Se persigue dar una salida rápida, aunque sea a un precio bajo, a los artículos de poco interés comercial.

El proyecto se ha desarrollado mediante una metodología Agile con paneles Kanban. El uso de este modelo iterativo e incremental nos ha permitido ir añadiendo funcionalidades nuevas a la vez que se mejoraban las ya implementadas.

Se ha empleado el patrón Modelo-Vista-Controlador, pues se adapta perfectamente a este tipo de aplicaciones. El modelo gestiona la base de datos con los artículos, los usuarios, las ventas y las donaciones, la vista presenta las interfaces sobre las que actúan los usuarios que acceden a la aplicación y el controlador maneja las interacciones de los usuarios y coordina las acciones entre el modelo y la vista. Los principales lenguajes de programación empleados para

RESUMEN

modelo, vista y controlador han sido respectivamente MySQL, HTML y JavaScript.

Palabras claves: organizaciones benéficas, comercio electrónico, financiación, subasta, subasta inversa.

Abstract

This final degree project develops an application to fund charitable organizations through an e-commerce system. The originality of the project lies in its focus on these types of entities, which involves certain particularities:

Firstly, we must consider the origin of the products sold. In most cases, they will come from donations or be manufactured by the organization itself, but generally, they will be low-value items. This entails two specific requirements: on the one hand, the application must include the functionality to accept donations, both monetary and in-kind, and on the other hand, it must contemplate specific sales systems that allow some return on these low-value items.

Secondly, we must take into account that the primary activity of the entity is not commerce, which means that it generally will not have warehouses or other logistical services and will need to quickly dispose of all its products and monetize them quickly, even if it is not at the optimal price. This fact, along with the previously mentioned low value of their products, has led to considering three different types of sales: direct sale, auction, and reverse auction.

Direct sale is the usual system where each item has a fixed price. It is applicable to those products that have a well-defined market price. It will be enough to set a slightly lower price to achieve the goal of a quick sale.

The auction is applicable to those products that do not have a clear market price because the supply is small. By increasing the sale price with each bid, high economic returns can be obtained, as a potential buyer might be willing to pay a high price to take advantage of the opportunity to obtain a rare product.

Finally, the reverse auction applies to low-value products or those with excessive supply. It starts with a relatively high price and gradually decreases until an interested buyer appears. The aim is to quickly dispose of items of little commercial interest, even at a low price.

The project has been developed using an Agile methodology with Kanban boards. The use of this iterative and incremental model has allowed us to add new functionalities while improving the already implemented ones.

The model-view-controller pattern has been used, as it perfectly fits this type of application. The Model manages the database with the items, users, sales, and donations; the View presents the interfaces that users interact with when accessing the application; and the Controller handles user interactions and coordinates actions between the Model and the View. The main programming languages used for the Model, View, and Controller have been MySQL, HTML, and JavaScript, respectively.

Keywords: charitable organizations, e-commerce, funding, auction, reverse auction.

Capítulo 1

Descripción del proyecto

1.1. Introducción

Este TFG desarrolla una aplicación de comercio electrónico especialmente diseñada para posibilitar la financiación de organizaciones sin ánimo de lucro. Inicialmente se ha pensado en ONG, pero bien podría aplicarse a otro tipo de instituciones con dificultades de financiación, como pueden ser por ejemplo, residencias de ancianos, clubes deportivos u otras muchas.

En general, todas estas entidades se financian, ya sea total o parcialmente, mediante donativos, que pueden ser económicos u objetos de poco valor, o mediante la venta de productos fabricados artesanalmente por ellos mismos. A diferencia de los comercios al uso, estas entidades no ofrecen un servicio de intermediación para la venta de bienes fabricados con fines comerciales, ni disponen de almacenes en los que guardar sus productos indefinidamente, hasta que llegue una oferta de compra adecuada. En su caso, lo que se busca es obtener algún ingreso económico de los objetos donados y no es necesario superar un importe mínimo en la venta, ya que en ningún caso se realizará a pérdidas. Por supuesto, se debe buscar conseguir el precio de venta más alto posible, pero es preferible vender a un precio bajo que mantener el artículo.

Para conseguir este objetivo se plantean tres tipos de venta, en cuanto a la forma en que se fijan los precios:

- Venta directa.
- Subasta.
- Subasta inversa.

La venta directa resulta adecuada para aquellos productos que tienen un valor claro definido en el mercado, por existir una alta oferta y demanda de los mismos. En este caso, la organización debe fijar un precio ligeramente inferior al de mercado para conseguir una venta rápida.

En el caso de la subasta, la entidad marca un precio inicial de salida para el producto, pero la venta no se produce hasta transcurrido un plazo establecido. Durante este plazo, los potenciales compradores pueden pujar, ofreciendo en cada puja un precio de compra superior al anterior.

Con cada puja se actualiza el precio de venta. Una vez finalizado el plazo estipulado, se realiza la venta al mejor postor, al que realizó la última puja.

La subasta es adecuada para productos sobre los que hay una fuerte demanda, pero poca oferta. Al ser pequeña la oferta, no se producen suficientes transacciones como para conocer cuál es exactamente su precio de mercado. Un potencial comprador podría estar dispuesto a pagar un precio más elevado del fijado inicialmente para aprovechar la oportunidad de adquirir un producto del que generalmente no hay oferta.

La subasta inversa, por el contrario, es adecuada para productos de los que hay poca demanda o excesiva oferta y son difíciles de vender. En este caso, el vendedor fija un precio de salida que va bajando periódicamente hasta que encuentra algún comprador interesado. Se puede conseguir así, en la mayoría de los casos, alguna remuneración económica, aunque sea pequeña, por la venta de productos de poco interés comercial.

1.2. Objetivos del trabajo

Con la realización de este Trabajo Fin de Grado se pretenden alcanzar tres tipos de objetivos:

- Académicos.
- Sociales.
- Comerciales.

Desde el punto de vista académico, el TFG permitirá mostrar cómo los conocimientos adquiridos por la alumna durante la realización del grado en ingeniería informática son adecuados para el desarrollo y ejercicio de la profesión. Este TFG desarrolla una aplicación informática completa y plenamente funcional, similar en complejidad a otras a las que el desarrollador podría enfrentarse en el ejercicio profesional.

La realización de este TFG también cumple un objetivo social, ya que define una aplicación especialmente orientada a la financiación de entidades sin ánimo de lucro que ayudan a los grupos menos favorecidos de la sociedad. Los métodos específicos de venta que se establecen no pretenden obtener el máximo beneficio posible de cada producto, sino más bien aprovechar todos los productos donados y los fabricados en la organización, incluso los de poco valor, para obtener algún aprovechamiento económico que sirva de microfinanciación.

Por último, aunque como decíamos la aplicación cubre un objetivo social, también es susceptible de ser rentabilizada mediante su venta a empresas comerciales. Bien directamente o bien adaptándola con las particularizaciones necesarias.

1.3. Entorno de aplicación

Existen diversas plataformas que tienen como objetivo fundamental el financiamiento de organizaciones sin ánimo de lucro, pero la mayoría de ellas solo aceptan donaciones económicas o, en caso de aceptar otro tipo de bienes, solo los aceptan para uso propio, no para monetizarlos. Las más conocidas son:

1. Donorbox:

- Donorbox es una plataforma online que se enfoca en facilitar donaciones para organizaciones sin fines lucrativos. No suele aceptar donaciones no monetarias, ya que está diseñada principalmente para recibir contribuciones financieras a través de métodos de pago en línea, como tarjetas de crédito, débito y PayPal.

2. GoFundMe Charity:

- GoFundMe Charity es una plataforma de recaudación de fondos online que permite a las organizaciones sin ánimo de lucro crear campañas personalizadas para sus causas. Aunque su enfoque principal es la recaudación de fondos, algunas organizaciones pueden optar por incluir la opción de donaciones no dinerarias en sus campañas.

3. GlobalGiving:

- GlobalGiving es una plataforma de crowdfunding que conecta a donantes con organizaciones de este tipo en todo el mundo. Si bien su enfoque principal es la recaudación directa de fondos para proyectos específicos, algunas organizaciones pueden incluir opciones para otro tipo de donaciones no monetarias en sus campañas.

4. Kickstarter:

- Kickstarter es una plataforma de crowdfunding donde las ONG pueden lanzar campañas para financiar proyectos específicos. Su enfoque principal es la recaudación de fondos económicos para proyectos creativos, tecnológicos y empresariales, por lo que no suele aceptar otro tipo de donaciones.

Por su parte, las aplicaciones o plataformas que permiten la compra venta o subasta de objetos, están enfocadas a obtener un beneficio económico para el propietario del artículo y no contemplan la donación. Las más habituales son:

1. eBay:

- Es una de las plataformas de compraventa online más grandes y conocidas a nivel mundial. Permite a usuarios tanto particulares como comerciantes vender una amplia variedad de artículos nuevos y usados a través de subastas o ventas directas.

2. Wallapop:

- Es una aplicación móvil y una plataforma online que permite a los usuarios comprar y vender productos de segunda mano entre particulares. Los usuarios pueden crear listados de productos y conectarse con compradores cercanos para realizar transacciones en persona.

3. Segundamano:

- Es una plataforma online que permite a los usuarios comprar y vender una gran variedad de artículos de segunda mano, desde productos electrónicos hasta muebles y vehículos, por ejemplo. Ofrece anuncios clasificados y la posibilidad de realizar transacciones entre particulares.

4. **Amazon:**

- Aunque Amazon es más conocida como una tienda online que vende productos nuevos, también ofrece la posibilidad de vender productos de segunda mano a través de su plataforma. Además, cuenta con opciones de subastas para ciertos productos a través de su servicio “Amazon Auctions”.

5. **Etsy:**

- Es una plataforma popular para la venta de productos artesanales, únicos y hechos a mano, así como productos vintage. Los vendedores pueden crear sus propias tiendas online y listar sus productos para la venta.

	Donación monetaria	Donación artículos	Venta directa	Subasta directa	Subasta inversa
Donorbox	✓	✗	✗	✗	✗
GoFoundMe Charity	✓	✓	✗	✗	✗
GlobalGiving	✓	✓	✗	✗	✗
Kickstarter	✓	✓	✗	✗	✗
eBay	✗	✗	✓	✓	✗
Wallapop	✗	✗	✓	✗	✗
Segundamano	✗	✗	✓	✗	✗
Amazon	✗	✗	✓	✓	✗
Etsy	✗	✗	✓	✗	✗
Wallaorg	✓	✓	✓	✓	✓

Tabla 1.1: Estado del arte

Como se puede apreciar en la tabla 1.1, a diferencia de todas las anteriores, nuestra aplicación contempla un modelo de financiación que emplea de forma conjunta las donaciones, la compra venta y la subasta, no conociéndose ninguna otra que proporcione la misma funcionalidad.

1.4. Estructura de la memoria

La memoria de este proyecto se estructura en los siguientes capítulos:

1. Descripción del proyecto.
2. Metodología.
3. Planificación.
4. Análisis.
5. Diseño.
6. Implementación.
7. Pruebas.
8. Manuales de instalación y de usuario.
9. Conclusiones y líneas futuras.
10. Bibliografía.

Este primer capítulo, descripción del proyecto, contiene una descripción general del trabajo, explica los objetivos que se persiguen conseguir y el entorno en que se desarrolla la aplicación.

El segundo capítulo, dedicado a la metodología, explica el proceso de desarrollo seguido, en nuestro caso uno iterativo, para hacer la aplicación, mostrando las herramientas empleadas y la arquitectura lógica y física por la que se ha optado.

El tercer capítulo realiza la planificación temporal y la obtención del presupuesto económico. En él, también se estima el esfuerzo necesario para desarrollar la aplicación y, a partir de dicha estimación, calcular otra planificación temporal y obtener otro presupuesto económico.

En el cuarto capítulo se describe el análisis de la aplicación, como son los usuarios, requisitos, casos de uso y atributos de calidad.

El quinto capítulo, diseño, se expone tanto el modelo de datos como los diferentes diagramas de clase y de secuencia.

El sexto capítulo se centra en especificar los pasos seguidos durante la implementación de la aplicación y la estructura de carpetas que sigue.

En el séptimo capítulo se evidencian las pruebas empleadas para garantizar el correcto funcionamiento de la página web.

El octavo capítulo, dedicado a los manuales de instalación y de usuario, se explican los pasos que se han de seguir para iniciar la página y todas las funcionalidades de cada uno de los usuarios.

El noveno capítulo está dedicado a las conclusiones y a las posibles líneas de investigación futuras que pueden desarrollarse a partir de este trabajo.

Por último, en el capítulo de la bibliografía, se detallan los libros consultados para el desarrollo de la aplicación.

Capítulo 2

Metodología

2.1. Proceso de desarrollo

El proyecto se ha desarrollado mediante una metodología Agile con paneles Kanban. Se ha elegido la metodología Agile porque permite gestionar el proyecto de manera iterativa e incremental, lo que se consigue dividiéndolo en pequeñas entregas funcionales que pueden ser desarrolladas en cortos periodos de tiempo, que se conocen como sprints. Aunque esta metodología está indicada para emplearla en equipos de trabajo pluripersonales, se ha optado por utilizarla en este caso porque la autora, que en este caso es única, quería experimentar todos los roles, con el fin de demostrar su capacidad para desempeñarlos en un entorno laboral complejo.

Se ha empleado Kanban para visualizar y gestionar el flujo de trabajo, dividiendo el tablero Kanban en cinco columnas: “To Do“, “In Progress“, “Blocked“, “Testing“ y “Done“, lo que ha permitido un control detallado del estado de todas las tareas de cada sprint. La herramienta empleada para gestionar los paneles ha sido Trello.

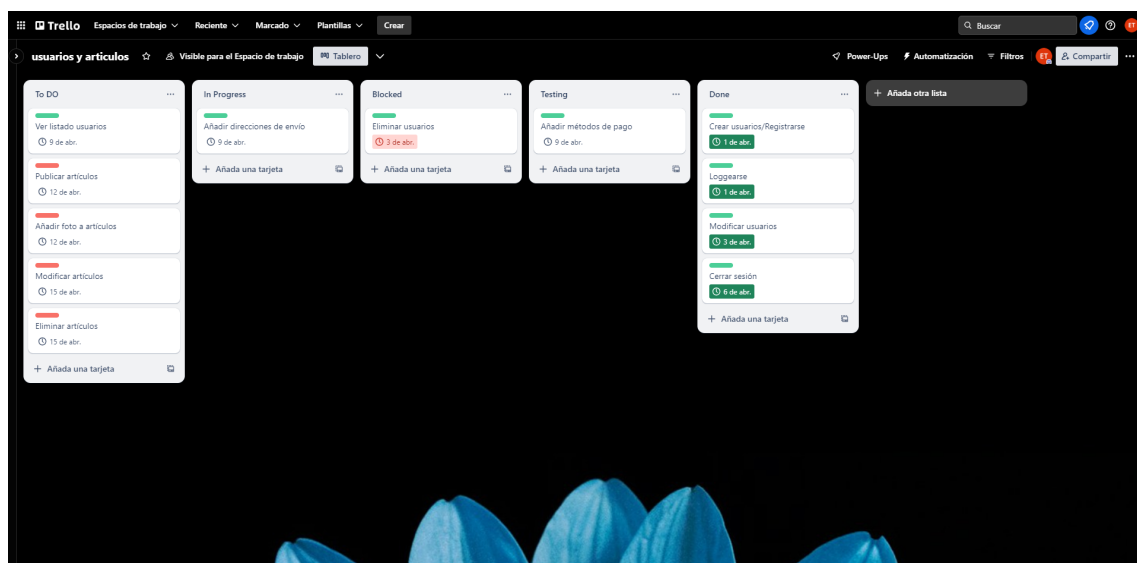


Figura 2.1: Tablero Trello Sprint Usuarios y artículos

En la figura 2.1 se muestra una captura de pantalla de uno de los tableros utilizados en el desarrollo del proyecto.

El desarrollo del proyecto se ha realizado mediante un Modelo Iterativo e Incremental, que divide el desarrollo de la aplicación en iteraciones o sprints, cada una de ellas centrada en la entrega de funcionalidades específicas y completas. Al final de cada iteración, se realiza una revisión con el equipo y los usuarios para evaluar el progreso y realizar los ajustes necesarios, aunque en este caso, el desarrollador ha tenido que asumir todos los roles. En cada iteración se hace entrega de un producto nuevo y del producto anterior revisado. Se consigue así entregar en cada sprint producto de valor.

La gran utilidad de este modelo radica en su capacidad para obtener una retroalimentación temprana y frecuente, lo que facilita la adaptación a los cambios y la mejora continua del producto.

El proyecto se ha dividido en tres sprints de una duración de tres semanas cada uno, que se han denominado: usuarios y artículos, el primero, donaciones y subasta directa, el segundo y venta directa y subasta inversa, el tercero.

Por último se ha utilizado el patrón Modelo-Vista-Controlador (MVC), que separa la lógica de negocio (Modelo), la presentación de la interfaz de usuario (Vista) y la gestión de las interacciones entre ambas (Controlador), lo que permite modularizar y organizar el código de manera eficiente, a la vez que facilita el mantenimiento y la escalabilidad de la aplicación.

El Modelo gestiona la lógica de acceso a la base de datos y el procesamiento de datos, la Vista se encarga de la presentación de la interfaz de usuario y el Controlador maneja las interacciones del usuario y coordina las acciones entre el Modelo y la Vista.

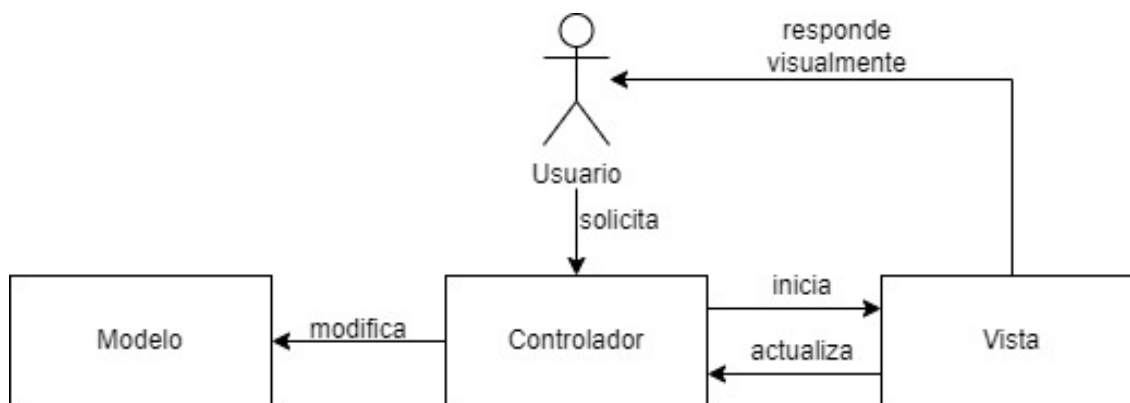


Figura 2.2: Arquitectura del patrón Modelo-Vista-Controlador

La figura 2.2 representa la arquitectura de aplicación que sigue el patrón Modelo-Vista-Controlador.

Se ha optado por una programación estructural en lugar de una orientada a objetos porque se ha considerado más adecuada para trabajar con bases de datos en SQL. Esto es debido a que en la programación estructural resulta más sencillo integrar las consultas a las bases de datos, que generalmente solo implican la manipulación directa de estos, sin ninguna necesidad de mapear objetos.

2.2. Herramientas utilizadas

El editor de código fuente escogido para el desarrollo de la aplicación ha sido Visual Studio Code de Microsoft, por sus excelentes funcionalidades para desarrolladores, su amplia versatilidad y, por supuesto, por ser uno de los editores más extendidos. También ha resultado definitivo para su selección que incluya el control integrado de Git, herramienta que permite una eficiente gestión del control de cambios.

La aplicación se ha desarrollado empleando Node.js, Express y Sequelize. Express que es uno de los marcos web más populares y ampliamente utilizados en el ecosistema de Node.js, mientras que Sequelize se ha utilizado para manejar la interacción con la base de datos. Para el desarrollo del código de la aplicación se ha empleado JavaScript como lenguaje de programación, CSS como lenguaje de estilos, MySQL como sistema gestor de bases de datos y XAMPP para administrar MySQL.

Tanto Node.js como Express están desarrollados en JavaScript, por lo que los tres se combinan de forma muy eficiente.

Para la gestión del pago se ha utilizado Stripe.

- **Visual Studio Code se ha utilizado como editor de código fuente:**

Visual Studio Code es un editor de código fuente de la empresa Microsoft. Permite la depuración de código, tiene control integrado de Git para control de versiones y funciones avanzadas como editor.

- **GitHub se ha utilizado como herramienta de gestión de control de cambios:**

GitHub es una plataforma de desarrollo de software que hace uso del conocido sistema de control de versiones Git. Aunque además permite desarrollo colaborativo del software, no se ha hecho uso de esta funcionalidad, pero aun así ha resultado muy útil para gestionar la evolución del software.

- **Node.js se ha utilizado como entorno de ejecución:**

Node.js es un entorno de ejecución de JavaScript en el lado del servidor. Permite ejecutar código JavaScript fuera del navegador web, lo que brinda la posibilidad de emplear JavaScript para desarrollar aplicaciones de servidor, APIs y otras aplicaciones back-end.

- **Express se ha empleado como marco de aplicación web:**

Express es un marco web minimalista y flexible para Node.js que facilita la creación de aplicaciones web y APIs. Se combina con Node.js para proporcionar una estructura y una serie de características que simplifican el desarrollo de aplicaciones web, como el enrutamiento de URL, el manejo de solicitudes HTTP, middleware y otros.

- **Sequelize se ha empleado como ORM (Object-Relational Mapping):**

Sequelize es una librería JavaScript que permite interactuar con bases de datos relaciones utilizando JavaScript. Entre otras, destaca su característica para realizar consultas a bases de datos con métodos que proporciona como findAll, findOne, create, update, destroy...

- **JavaScript se ha utilizado como lenguaje principal de programación:**

Precisamente, la ventaja de usar Node.js y Express es poder desarrollar todo el código en JavaScript, tanto en el lado del servidor como en el cliente. Con ello, hemos conseguido simplificar todo el desarrollo al proporcionar un lenguaje de programación coherente en toda la aplicación.

- **CSS (Cascading Style Sheets) se ha utilizado como lenguaje principal de estilos:**

CSS es un lenguaje utilizado para definir el aspecto y el diseño visual de una página web. Permite aplicar estilos como colores, fuentes, márgenes, tamaños y disposición a los elementos node.js. CSS separa el contenido estructural de la presentación visual, lo que permite una mayor flexibilidad y mantenimiento en el diseño de una página web.

- **XAMPP se ha utilizado para el entorno de desarrollo local:**

XAMPP es un paquete de software libre que incluye Apache, MySQL, PHP y Perl. Proporciona un entorno de desarrollo local completo para probar y desarrollar aplicaciones web. Permite a los desarrolladores ejecutar un servidor web local en sus propias máquinas y simular un entorno de servidor de producción sin necesidad de acceder a servidores en la nube o en red.

- **MySQL (My Structured Query Language) se ha empleado como sistema gestor de bases de datos:**

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que utiliza el lenguaje SQL (Structured Query Language) para realizar consultas, administrar y manipular datos en bases de datos relacionales.

- **Stripe se ha utilizado para gestionar los pagos en la aplicación:**

Proporciona la infraestructura técnica, de prevención de fraude y bancaria necesaria para operar sistemas de pago en línea. También ofrece herramientas para desarrolladores y tarjetas de pago de prueba, para simular pagos y así poder probar el correcto funcionamiento de la integración.

Además de las herramientas empleadas para el desarrollo, la memoria se ha escrito en \LaTeX , utilizando el editor TexMaker y nuevamente GitHub para el control de cambios. Los diagramas se han realizado con StarUML, draw.io y GanttProject.

Por último, la gestión de tareas se ha realizado mediante un tablero en Trello.

- **\LaTeX :** Se ha utilizado para escribir la memoria.

- **TexMaker:** Editor utilizado para la escritura de documentos en \LaTeX .

- **StarUML:** Herramienta empleada para la creación de diagramas UML.

- **draw.io:** Utilizado para la realización de diagramas de flujo y otros gráficos.

- **GanttProject:** Herramienta para la creación de diagramas de Gantt y planificación de proyectos.
- **Trello:** Utilizado para la gestión de tareas mediante un tablero.

2.3. Arquitectura

En la figura 2.3 podemos ver la arquitectura lógica cliente servidor. En ella se pueden apreciar tres capas:

- **Capa de presentación.**

Esta capa representa la interfaz de usuario de la aplicación. En ella, los usuarios pueden ver y manipular la información presentada por la aplicación. El cliente que los usuarios finales utilizarán para interactuar con el sistema será un navegador web, como puede ser Google Chrome, Firefox, Safari...

- **Capa de aplicación.**

En esta capa se encuentra el servidor de la aplicación, construido por Node.js y Express. También incluye el acceso a datos mediante Sequelize.

- **Capa de datos.**

Esta capa representa el almacenamiento de datos de la aplicación. El origen de los datos se encuentra en MySQL, donde se almacenan y recuperan para ser utilizados por la aplicación.

La figura 2.4 representa la arquitectura física correspondiente a dicho modelo lógico. Se pueden observar los siguientes dos elementos:

- **Servidor.**

En él, se aloja y ejecuta la aplicación. Es el encargado de manejar las solicitudes de los clientes y proporcionarles respuestas.

- **Clientes.**

Son los dispositivos finales. Cualquier dispositivo que solicite y reciba servicios de los servidores. Cada uno de ellos ejecuta un navegador web, que se conecta al servidor a través de internet.

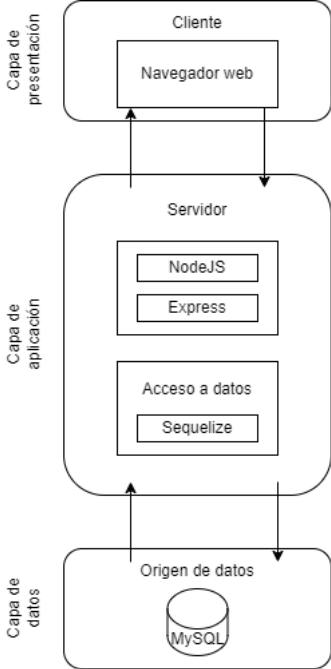


Figura 2.3: Arquitectura lógica

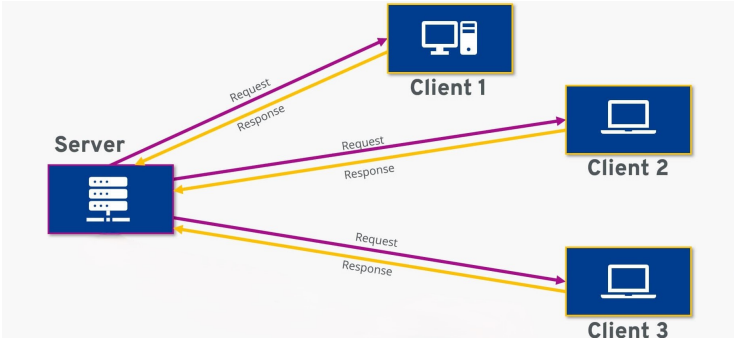


Figura 2.4: Arquitectura física

Capítulo 3

Planificación

3.1. Planificación inicial

3.1.1. Planificación temporal

El desarrollo de una página web puede variar en duración y etapas dependiendo de varios factores, como el tamaño y la complejidad del proyecto, así como la disponibilidad de recursos y el equipo de desarrollo.

Podemos diferenciar cinco etapas:

1. **Planificación y análisis:** en esta etapa, se define el propósito de la página web, se identifican los objetivos y se recopilan los requisitos del proyecto. Se investiga a la competencia y se determinan las funcionalidades necesarias. Suele abarcar entre un 10 % y un 20 % de la duración total del proyecto, en este caso vamos a suponer un 10 % para nuestra aplicación.
2. **Diseño:** Se crea el diseño visual de la página web, incluyendo la estructura de navegación, la disposición de los elementos en la página y el diseño de interfaz de usuario (UI). Suele suponer entre un 20 % y un 30 % de la duración, para nuestro proyecto vamos a suponer un 20 %.
3. **Desarrollo:** en esta etapa, se lleva a cabo la implementación del diseño en código. Se implementa la base de datos, se escriben y se integran los componentes HTML, CSS y JavaScript necesarios para crear la funcionalidad y el aspecto visual de la página web. Supone entre un 30 % y un 50 % de la duración, vamos a suponer un 40 %.
4. **Pruebas y revisión:** Se realizan pruebas exhaustivas para identificar y corregir errores en la funcionalidad, el rendimiento y la compatibilidad con diferentes dispositivos y navegadores. Abarca entre un 10 % y un 20 %, en nuestro caso un 20 %.
5. **Documentación:** creación de documentación detallada que incluya manuales de usuario, guías de administración de contenido, descripción de funcionalidades, diagramas de

Capítulo 3. Planificación

arquitectura, entre otros. Supone entre un 5 % y un 10 % del total, en nuestro caso un 10 %.

Suponiendo una jornada de 5 horas diarias, 5 días a la semana y 4 semanas al mes; 300 horas serían 60 días, 12 semanas o, lo que es lo mismo, 3 meses.

Siguiendo los porcentajes descritos, en la tabla 3.1 se encuentra la planificación temporal inicial para las 300 horas de duración de un TFG de 12 créditos ECTS. Como se puede observar en la figura 3.1, la fecha de inicio es el cuatro de marzo de dos mil veinticuatro y la fecha de finalización estimada es el veintiocho de mayo de dos mil veinticuatro.

	Planificación y análisis	Diseño	Desarrollo	Pruebas y revisión	Documentación
Porcentaje (%)	10	20	40	20	10
Planificación	30	60	120	60	30

Tabla 3.1: Planificación temporal inicial

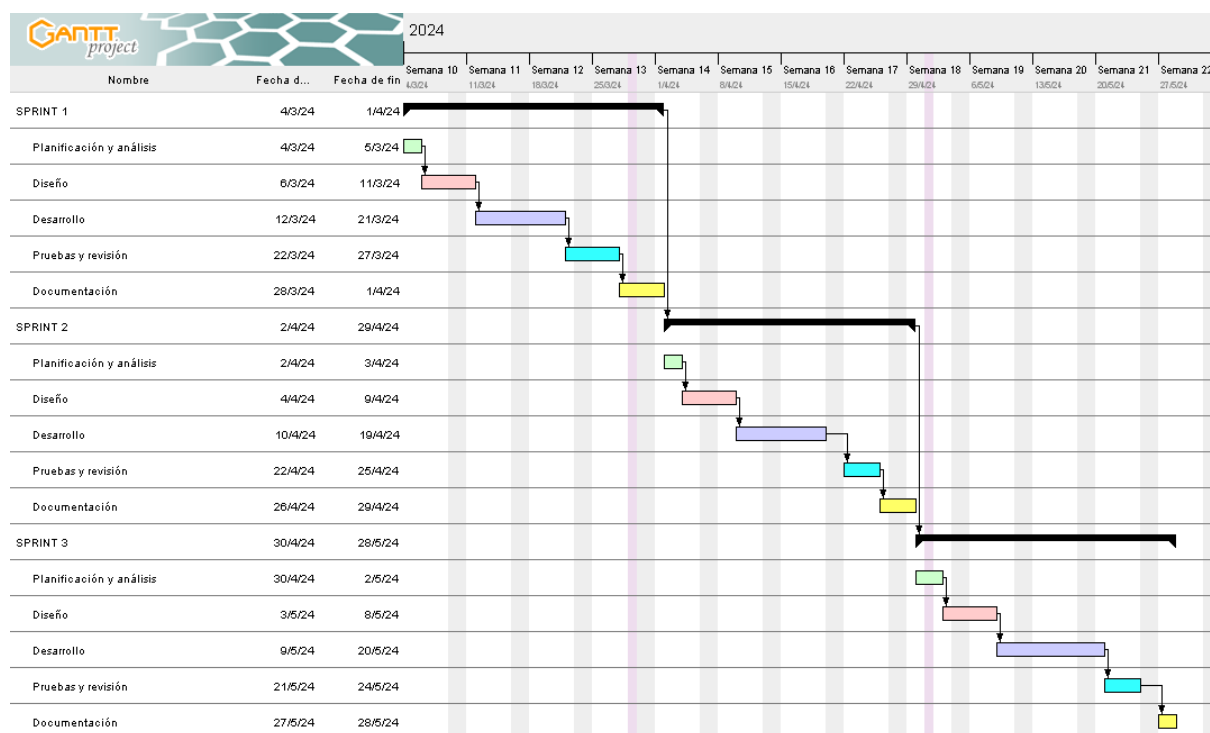


Figura 3.1: Diagrama de Gantt inicial

Un diagrama de Gantt se visualiza mediante barras horizontales en un eje cronológico, donde cada barra denota una tarea particular del proyecto. El eje horizontal generalmente muestra

el tiempo, mientras que el vertical muestra cada una de las tareas. Por ejemplo, en el diagrama de Gantt 3.1, las tareas de desarrollo del primer sprint empezarían el doce de marzo de dos mil veinticuatro y finalizarían el veintiuno de marzo de dos mil veinticuatro.

3.1.2. Presupuesto económico

En la tabla 3.2 se puede ver el presupuesto de hardware utilizado, mientras que en la tabla 3.3 se representa el presupuesto software. Dado que se emplean programas de software libre, el presupuesto para este último es de 0 €.

Hardware	Precio total	Vida útil	Porcentaje uso	Coste en proyecto
Ordenador portátil	1.000 €	6 años	4,17 %	41,7 €
Teclado y ratón	35,09 €	6 años	4,17 %	1,46 €
Monitor	89,99 €	6 años	4,17 %	3,75 €
Router	0 €	-	3 meses	0 €
Fibra	27 €/mes	1 mes	3 meses	81 €
Coste energético	40 €/mes	1 mes	3 meses	120 €
Total	-	-	-	247,91 €

Tabla 3.2: Presupuesto hardware inicial

Software	Precio total	Coste en proyecto
Node.js	0 €	0 €
Visual Studio Code	0 €	0 €
GitHub	0 €	0 €
XAMPP	0 €	0 €
Google Chrome	0 €	0 €
Total	-	0 €

Tabla 3.3: Presupuesto software

Como no se dispone de un perfil de trabajo para elaborar toda la documentación, vamos a repartir el 10 % del tiempo que conlleva de forma proporcional entre el resto de roles. Así, el analista emplea un 1,1 % más de tiempo para realizar la documentación, el diseñador y el tester un 2,2 % más cada uno y el desarrollador un 4,5 % más.

En la tabla 3.4 podemos ver el presupuesto en recursos humanos.

Los salarios medios se han obtenido de <https://es.talent.com/>, y se han adaptado a una jornada laboral de 35 horas semanales en vez de 40.

Recursos humanos	Salario	Tiempo	Coste en proyecto
Jefe de proyecto	21.222,16 €/año 1.768,21 €/mes	100 % 3 meses	5.304,63 €
Analista	18.008,49 €/año 1.500,70 €/mes	11,1 % 0,4 meses	600,28 €
Diseñador	14.141,55 €/año 1.178,46 €/mes	22,2 % 0,6 meses	707,08 €
Desarrollador	18.145 €/año 1.512,08 €/mes	44,5 % 1,3 meses	1.965,70 €
Tester	16.759,06 €/año 1.396,59 €/mes	22,2 % 0,7 meses	977,61 €
Total	-	-	9.555,3 €

Tabla 3.4: Presupuesto recursos humanos inicial

Por lo que, teniendo en cuenta los presupuestos anteriores ascendería a 10.916,67 €, como se muestra en la tabla 3.5.

	Hardware	Software	RRHH	Total
Coste inicial	247,91 €	0 €	9.555,3 €	9.803,21 €

Tabla 3.5: Presupuesto total inicial

3.2. Planificación según estimación de esfuerzo

3.2.1. Estimación de esfuerzo

La selección de una metodología u otra para estimar el esfuerzo de desarrollo de una aplicación, en general depende del contexto específico del proyecto y de las preferencias del equipo de desarrollo. En este caso se ha optado por utilizar el método de puntos de casos de uso porque proporciona una perspectiva centrada en el usuario, facilita la comunicación y la comprensión de los requisitos del sistema, ofrece flexibilidad para adaptarse a posibles cambios en los requisitos y porque se centra en la funcionalidad y el valor para el usuario.

El método de estimación de esfuerzo mediante casos de uso, básicamente consiste en identificar los casos de uso del proyecto, ajustarlo mediante factores técnicos y ambientales y calcular las horas por caso de uso en función de la productividad, para, con todo ello, obtener la productividad final.

1. Puntos de casos de uso sin ajustar (UUCP):

En este paso, calculamos los puntos de caso de uso sin ajustar asignando un factor de peso a cada uno de los casos de uso de la sección 4.3 y de los actores de la sección 4.1, en función de su complejidad.

- **Factor de peso de los actores sin ajustar (UAW):**

Complejidad de los actores	Peso	Número de actores	Total
Simple	1	1 (Stripe)	1
Medio	2	0	0
Alto	3	3 (usuarios)	9
Total (UAW)	-	4	10

UAW=10

- **Factor de peso de los casos de uso sin ajustar (UAW):**

Complejidad de los casos de uso	Peso	Num. casos de uso	Total
Simple ($1 \leq$ Número de Transacciones ≤ 3)	5	4	20
Medio ($4 \leq$ Número de Transacciones ≤ 7)	10	4	40
Alto ($8 \leq$ Número de Transacciones)	15	2	30
Total (UAW)	-	10	90

UAW=90

- **Puntos de caso de uso sin ajustar (UUCP):**

$$UUCP = UAW + UAW$$

$$UUCP = 10 + 90 = 100$$

2. Factor técnico (complejidad tecnológica) (TCF):

En este paso, obtenemos la complejidad tecnológica del proyecto mediante la ponderación de trece criterios, establecida en función de sus características.

- **Factor técnico (TF):**

Factor	Peso	Relevancia	Total
Sistema distribuido	2	0	0
Tiempos de respuesta	2	4	8
Eficiencia del usuario final	1	4	4
Procesamiento interno complejo	1	3	3
Código reutilizable	1	3	3
Facilidad de instalación	0.5	0	0
Facilidad de uso	0.5	5	2,5
Portabilidad	2	0	0
Facilidad de cambio	1	3	3
Concurrencia	1	5	5
Seguridad	1	5	5
Acceso directo a terceras partes	1	1	1
Formación de los usuarios	1	2	2
Total (TF)	-	-	36,5

- **Factor de ajuste técnico (TCF):**

$$TFC = 0,6 + (0,01 * TF)$$

$$TFC = 0,6 + (0,01 * 36,5) = 0,965$$

3. Factor ambiental (ECF) :

Calculamos ahora el factor ambiental, de forma análoga, ponderando ocho criterios.

■ **Factor ambiental (EF):**

Factor	Peso	Presencia	Total
1. Familiaridad con el modelo de proyecto	1.5	4	6
2. Experiencia en la aplicación	0.5	3	1,5
3. Experiencia en programación secuencial	1	4	0
4. Capacidad de analista lider	0.5	3	1,5
5. Motivación	1	4	4
6. Estabilidad de los requisitos	2	4	8
7. Personal a tiempo parcial	-1	0	0
8. Dificultad del lenguaje de programación	-2	3	-6
Total	-	-	15

Tabla 3.6: Factor ambiental

■ **Factor de ajuste ambiental (ECF):**

$$EF = 1,4 + (-0,03 * EF)$$

$$ECF = 1,4 + (-0,03 * 15) = 0,95$$

4. Casos de uso ajustados (UCP):

Obtenemos los casos de uso ajustados multiplicando los no ajustados por los factores de ajuste, técnico y ambiental.

$$UCP = UUCP * TCF * ECF$$

$$UCP = 100 * 0,965 * 0,95 = 91,675$$

5. Horas por caso de uso (CF):

Para calcular el número de horas por caso de uso, se tiene en cuenta el factor de productividad (FP).

Si:

$$FP \leq 2, CF = 20 \text{ horas}$$

$$3 \leq FP \leq 4, CF = 28 \text{ horas}$$

$$5 \leq FP, CF = 36 \text{ horas}$$

■ **Factor de productividad (FP):**

Factores ambientales del 1 al 6 con presencia ≤ 3 y factores ambientales 7 y 8 con presencia >3 .

$$FP = 2 + 0 = 2$$

Como el Factor de productividad es 2,

- **Número de horas por caso de uso = 20.**

Por lo tanto,

- **Número de horas totales:**

$$\text{Número de horas totales} = CF * UCP$$

$$\text{Número de horas totales} = 20 * 91,675 = 1.833,5$$

6. Esfuerzo (E):

Finalmente, calculamos el esfuerzo mediante la siguiente fórmula:

$$E = \text{Número de horas totales} / \text{personas}$$

con lo que resulta:

$$E = 1.833,5/1 = 1.833,5 \text{ horas/persona}$$

3.2.2. Planificación temporal

Teniendo en cuenta las etapas y porcentajes descritos en la subsección 3.1.1, en la tabla 3.7 se encuentra la planificación temporal según la estimación calculada anteriormente.

	Planificación y análisis	Diseño	Desarrollo	Pruebas y revisión	Documentación
Porcentaje (%)	10	20	40	20	10
Estimación (horas)	183,35	366,7	733,4	366,7	183,35

Tabla 3.7: Planificación temporal según estimación

En la figura 3.2 se encuentra el diagrama de Gantt según la estimación. Según esto, la fecha de inicio es el cuatro de marzo de dos mil veinticuatro y la fecha de finalización estimada es el veintidos de agosto de dos mil veintiocho.

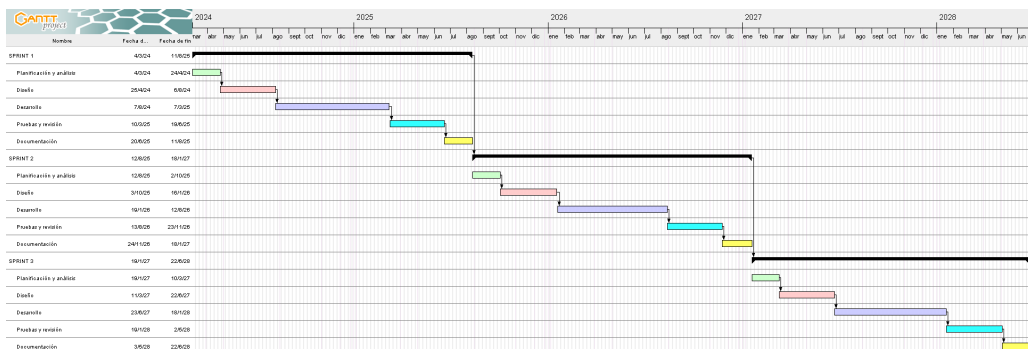


Figura 3.2: Diagrama de Gantt según estimación

3.2.3. Presupuesto económico

Suponiendo la misma jornada laboral, precios y salarios de la sección 3.1.2, los presupuestos económicos tanto de hardware, como de recursos humanos según la estimación, se detallan las tablas 3.8 y 3.9 respectivamente. El presupuesto software, al ser de 0€ en el caso anterior, en este caso se mantiene igual.

Hardware	Precio total	Vida útil	Porcentaje uso	Coste en proyecto
Ordenador portatil	1.000 €	6 años	21,29 %	212,9 €
Teclado y ratón	35,09 €	6 años	21,29 %	7,47 €
Monitor	89,99 €	6 años	21,29 %	19,16 €
Router	0 €	-	15,34 meses	0 €
Fibra	27 €/mes	1 mes	15,34 meses	414,18 €
Coste energético	40 €/mes	1 mes	15,34 meses	616,6 €
Total	-	-	-	1.267,31 €

Tabla 3.8: Presupuesto hardware según estimación

Recursos humanos	Salario	Tiempo	Coste en proyecto
Jefe de proyecto	21.222,16 €/año 1.768,21 €/mes	100 % 15,34 meses	27.129 €
Analista	18.008,49 €/año 1.500,70 €/mes	11,1 % 2,29 meses	3.436,62 €
Diseñador	14.141,55 €/año 1.178,46 €/mes	22,2 % 4,13 meses	4.867,05 €
Desarrollador	18.145 €/año 1.512,08 €/mes	44,5 % 7,79 meses	11.779,13 €
Tester	16.759,06 €/año 1.396,59 €/mes	22,2 % 4,13 meses	5.767,91 €
Total	-	-	52.979,71 €

Tabla 3.9: Presupuesto recursos humanos según estimación de tiempo

Teniendo en cuenta los presupuestos obtenidos, en la figura 3.10 se indican los costes en cada uno de los recursos, así como el total, que ascendería a 54.114,73 €.

	Hardware	Software	RRHH	Total
Estimación	1.267,31 €	0 €	52.979,71 €	54.247,02 €

Tabla 3.10: Presupuesto total según estimación

3.3. Comparativa entre planificaciones y seguimiento real

En las tablas 3.11 y 3.12 se comparan la planificación temporal y el presupuesto de la estimación por casos de uso frente a la planificación y presupuesto real.

	Planificación y análisis	Diseño	Desarrollo	Pruebas y revisión	Documentación	Total
Planificación inicial (horas)	30	60	120	60	30	300
Estimación (horas)	183,35	366,7	733,4	366,7	183,35	1.833,5

Tabla 3.11: Comparativa planificación temporal

	Hardware	Software	RRHH	Total
Coste inicial	247,91 €	0 €	9.555,3 €	9.803,21 €
Estimación	1.267,31 €	0 €	52.979,71 €	54.247,02 €

Tabla 3.12: Comparativa presupuestos

Como se puede apreciar, la diferencia entre ambas es muy significativa. La planificación por casos de uso resulta cinco veces superior en tiempo y coste a la real. Las causas de esta desviación pueden ser debidas a los siguientes factores:

1. En la tabla 3.6 se observa que en la estimación por casos de uso es recomendable contar con personal experimentado en el análisis de proyectos y familiarizado con el entorno de trabajo. Estos profesionales aportan una comprensión de los desafíos y oportunidades específicos del proyecto, lo que resulta en un factor ambiental de ajuste más bajo que el obtenido en nuestra estimación, para la que no disponemos de este tipo de personal.
2. Además, cuando se dispone de datos históricos de proyectos previos, tales como tamaño, costo y duración, pueden utilizarse como una referencia para realizar estimaciones más precisas. Las organizaciones que mantienen un registro detallado de proyectos anteriores pueden usar esta información para ajustar sus estimaciones. En nuestro caso se carece de datos de proyectos anteriores, lo que afecta a la fiabilidad de la estimación.
3. Por último, este método suele sobrestimar el esfuerzo cuando las funcionalidades definidas son reducidas. Nuestra aplicación tiene casos de uso generalmente con funcionalidades muy reducidas, por lo que las 20 horas por cada caso de uso pueden resultar excesivas. Una duración aproximada de 4 horas por cada caso de uso sería más realista y resultaría en una duración de 366,7 horas/persona, mucho más ajustada a la planificación real.

Por el contrario, la planificación inicial de la sección 3.1.1 se ha ajustado mucho mejor al desarrollo real. La duración de cada etapa se ha calculado con los porcentajes habituales que suelen utilizarse para este fin. Estos valores son apropiados para personal con experiencia. En nuestro caso, el reparto final de tiempos no se ha ajustado fielmente a lo planificado, porque la etapa de desarrollo ha requerido más tiempo del esperado, debido a que se ha empleado un lenguaje de programación nuevo y ha sido necesario familiarizarse con el mismo, mientras que por el contrario, la etapa de pruebas y revisión ha resultado más corta de lo planificado inicialmente, ya que ha sido posible agrupar las pruebas de varias funcionalidades en un único test. La etapa de documentación finalmente ha llevado más tiempo del previsto. Esto junto a algunos factores más no esperados, como jornadas laborales de mayor trabajo y otro tipo de inconvenientes, han aplazado la fecha final a las primeras semanas de junio. Estos retrasos no suponen un incremento importante en horas, por lo que no se ha creído necesario revisar el presupuesto económico por no desviarse significativamente de las 300 horas.

Capítulo 4

Análisis

4.1. Actores

Los actores de nuestro sistema son los operarios que pueden interactuar con él. Es decir, todos los usuarios.

En nuestro caso, se consideran tres tipos de actores, según la forma en que pueden interactuar con el sistema. Los denominamos: usuario, usuario registrado y administrador.

La siguiente cuadro explica de forma esquemática las características de cada uno de ellos.

ACTOR-1	Usuario	Cualquier usuario que puede acceder al sistema, no necesita estar registrado. Podrá ver el contenido básico.
ACTOR-2	Usuario registrado	Usuario que ha iniciado sesión en el sistema. Obligatoriamente debe estar registrado. Podrá ver todo el contenido y ofrecer objetos, realizar compras y pujar en las distintas subastas disponibles.
ACTOR-3	Administrador	Usuario registrado en el sistema como administrador. Puede realizar acciones como publicar y gestionar los artículos, responder las cuestiones de los usuarios, ...

Tabla 4.1: ACTORES

4.2. Requisitos

El análisis incluye la definición de todos los requisitos del sistema, que se enumeran en forma de cuadros agrupados en: requisitos de negocio (RN-XX), reglas de negocio (RgN-XX), requisitos de usuario (RU-XX), requisitos funcionales (RF-XX) y requisitos de información (RI-XX).

RN-01	Ofrecer un sistema de venta de objetos mediante venta directa, subasta directa o subasta inversa orientado a lograr una venta rápida de los mismos.
RN-01	Ofrecer la posibilidad de donaciones, tanto económicas, como de objetos para su posterior venta.

Tabla 4.2: Requisitos de negocio

RgN-01	El sistema solo permitirá a los usuarios administradores publicar, modificar o eliminar artículos.
RgN-02	El sistema solo permitirá a los administradores ver un listado de todos los usuarios dados de alta en la aplicación y todas las donaciones recibidas.
RgN-03	El sistema solo permitirá a los usuarios administradores crear nuevos usuarios administradores.

Tabla 4.3: Reglas de negocio

RU-01	Los usuarios podrán registrarse.
RU-02	Los usuarios, tanto registrados como no, podrán ver todos los artículos disponibles en la página.
RU-03	Los usuarios registrados podrán iniciar sesión.
RU-04	Los usuarios con sesión abierta podrán cerrar sesión.
RU-05	Los usuarios administradores podrán dar de alta, baja o modificar a otros usuarios, incluido a usuarios administradores.
RU-06	Los usuarios administradores podrán publicar artículos en venta directa. Podrán fijar el precio de los artículos en venta directa. Podrán modificar los datos siempre que lo deseen.
RU-07	Los usuarios administradores podrán publicar artículos en subasta directa. Podrán fijar el precio inicial de los artículos, también fijan la fecha en la que termina el plazo de dicha subasta. Podrán modificar los datos siempre que lo deseen.
RU-08	Los usuarios administradores podrán publicar artículos en subasta inversa. Podrán fijar el precio inicial de los artículos en venta mediante subasta inversa, el porcentaje de descuento diario y el precio mínimo de venta. Podrán modificar los datos siempre que lo deseen.
RU-09	Los usuarios registrados podrán donar artículos.
RU-10	Los usuarios registrados podrán comprar cualquier artículo que esté a la venta directa en la plataforma.
RU-11	Los usuarios registrados podrán comprar cualquier artículo que esté en subasta inversa.
RU-12	Los usuarios registrados podrán pujar por artículos en subasta directa tantas veces como deseen al mismo artículo, mientras la subasta no esté cerrada.
RU-13	Los usuarios administradores podrán dar de baja artículos no vendidos.
RU-14	Los usuarios administradores podrán ver los pedidos realizados.

RU-15	Los usuarios podrán elegir entre múltiples opciones de pago.
RU-16	Los usuarios podrán revisar y modificar el pedido antes de confirmarlo.
RU-17	Los usuarios deben poder utilizar un motor de búsqueda para poder encontrar artículos.
RU-18	Los usuarios deben poder eliminar artículos del carrito.
RU-19	Los usuarios deben poder ver el historial de compras.
RU-20	Los usuarios podrán añadir artículos al carrito.
RU-21	Los usuarios registrados podrán consultar sus datos.
RU-22	Los usuarios registrados podrán modificar sus datos.
RU-23	Los usuarios administradores podrán ver un listado de todos los usuarios registrados.
RU-24	Los usuarios registrados podrán ver los artículos que tienen en la cesta.
RU-25	Los usuarios administradores podrán un listado de todos los artículos donados.
RU-26	Los usuarios que hayan donado un artículo podrán modificarlo o eliminarlo, siempre que la donación aún no se haya publicado como artículo.
RU-27	Los usuarios administradores podrán publicar los artículos donados, pudiendo modificar los campos que deseen antes de que sea publicado.
RU-28	Los usuarios podrán ver un listado de los artículos que hayan donado.
RU-29	Los usuarios podrán realizar donaciones económicas.
RU-30	Los usuarios que han ganado una puja podrán formalizar la dirección de envío y pago.

Tabla 4.4: Requisitos de usuario

RF-01	El sistema actualizará el precio de cada artículo en subasta directa, cada vez que se produzca una puja.
RF-02	Cada día, el sistema actualizará el precio de los artículos en subasta inversa descontando el porcentaje fijado por el administrador.
RF-03	Si no quedan unidades disponibles de un artículo en venta directa, el sistema no mostrará el artículo.
RF-04	Una vez realizada una compra, el sistema generará un informe de compra, que deberá ser atendido por el usuario administrador.
RF-05	El sistema debe ser capaz de mostrar artículos con su precio e imágenes mediante un campo de búsqueda.
RF-06	El sistema actualizará los campos que sean modificados por el usuario, tanto de los artículos, como de las donaciones y de los perfiles de los usuarios.
RF-07	Si un usuario registrado lleva loggeado cinco minutos sin realizar ninguna acción sobre la página, este pasa a estar deslogueado automáticamente.
RF-08	Cuando un usuario se registre por primera vez, el sistema logea directamente al usuario.
RF-09	El sistema mostrará un aviso a un usuario cuando este haya ganado una puja.
RF-10	El sistema mostrará un aviso al administrador cuando se haya realizado un nuevo pedido.
RF-11	El sistema mostrará una pantalla al abrir la página mostrando información.

Tabla 4.5: Requisitos funcionales

RI-01	El sistema deberá almacenar el tipo de venta de cada producto (venta directa, subasta directa o subasta inversa).
RI-02	El sistema deberá almacenar para cada usuario al menos el nombre, la dirección de correo electrónico y la contraseña.
RI-03	El sistema deberá almacenar para cada producto al menos su nombre, descripción detallada, imágenes, precio y disponibilidad.

Tabla 4.6: Requisitos de información

4.3. Casos de uso

En este apartado se definen de forma esquemática todas las formas en que cada uno de los actores descritos en los apartados anteriores puede interactuar con el sistema. Aunque en rigor, cada una de estas formas de interacción constituye un caso de uso, vamos a contemplar únicamente diez, que aglutinan a todos los del diagrama, de acuerdo con los requisitos de usuario definidos en el punto anterior.

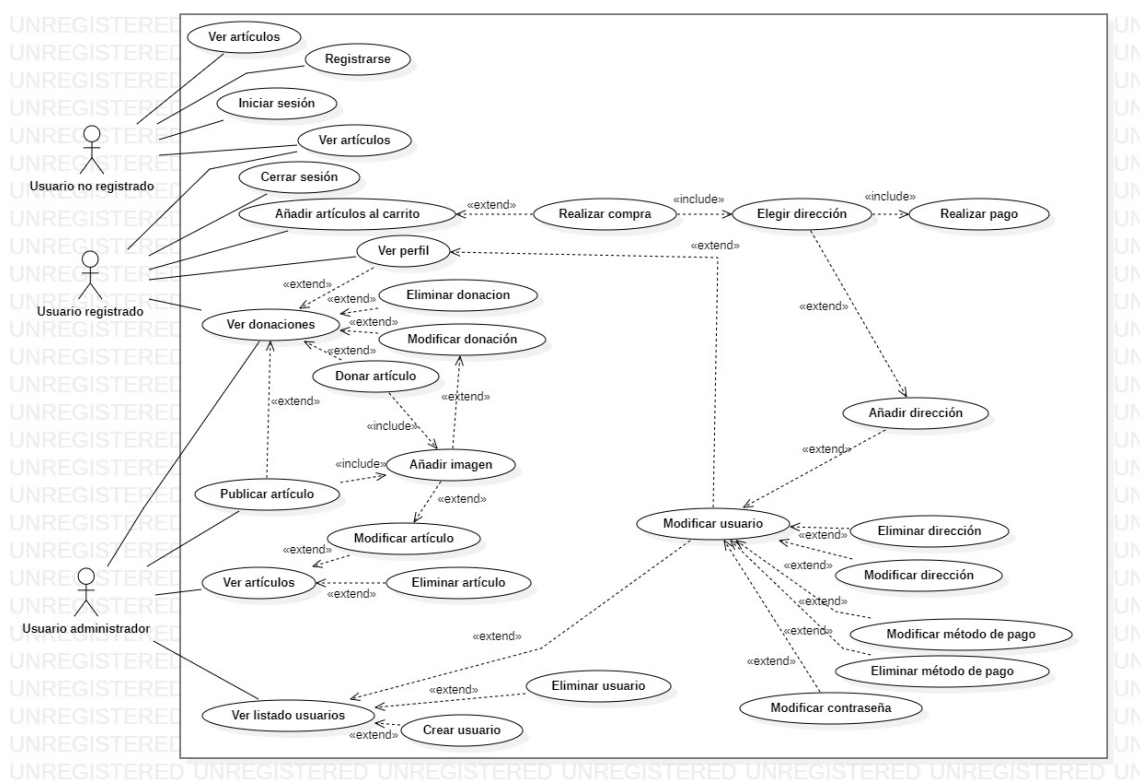


Figura 4.1: Diagrama casos de uso

En la tabla 4.7 se hace referencia al primer caso de uso, Registrarse, que describe el proceso que ha de seguir el usuario para poder registrarse en el sistema y sus posibles excepciones.

CU-01 Registrarse	
Nombre e ID del Requisito	RU-01. Los usuarios podrán registrarse. RU-05. Los usuarios administradores podrán dar de alta, baja o modificar a otros usuarios, incluido a usuarios administradores. RF-08. Cuando un usuario se registre por primera vez, el sistema loggea directamente al usuario. RI-02. El sistema deberá almacenar para cada usuario al menos el nombre, la dirección de correo electrónico y la contraseña.
Actor	ACTOR-1 ACTOR-3
Descripción	Todos los usuarios deben ser capaces de registrarse en la aplicación. El usuario envía sus datos de usuario al sistema, de manera que este creará un nuevo usuario, con el que posteriormente podrá iniciar sesión.
Precondiciones	PRE-1. El usuario no está identificado y loggeado en el sistema.
Postcondiciones	POST-1 Un usuario nuevo con los datos introducidos es almacenado en el sistema.
Flujo normal	FN1 El usuario accede a la página de registro de la tienda, mediante el botón "Registrarse". FN2 El actor completa el formulario de registro. FN3 El sistema comprueba los datos introducidos. FN4 Si los datos son correctos y no existe previamente una cuenta con el correo electrónico introducido, el sistema crea un nuevo usuario en la base de datos e inicia la sesión con ese usuario de manera automática.
Flujo alternativo 1	FA4 Si los datos son incorrectos se informa al usuario del error y no se procede a crear el usuario.
Flujo alternativo 2	FA3 Si existe un usuario con el mismo correo electrónico se informa al usuario y no se crea un usuario nuevo en la base de datos.
Flujo alternativo 3	FA1 El usuario administrador entra en la sección "Usuarios" y selecciona la opción "Crear nuevo usuario".

Excepciones	E1 El usuario ha dejado campos requeridos sin rellenar.
Prioridad	Alta

Tabla 4.7: CU-01. Registrarse

En la tabla 4.8 se hace referencia al segundo caso de uso, Iniciar sesión, que describe el proceso que ha de seguir el usuario para poder iniciar sesión en el sistema y sus posibles excepciones.

CU-02 Iniciar sesión	
Nombre e ID del Requisito	RU-03. Los usuarios registrados podrán iniciar sesión.
Actor	ACTOR-1
Descripción	Todos los usuarios registrados deben ser capaces de iniciar sesión en la web. El usuario envía sus datos de usuario al sistema, de manera que este consulta en la base de datos si existe un usuario asociado a esos datos, y si es así, logea al usuario.
Precondiciones	PRE-1. El usuario se ha registrado previamente en el sistema 4.7.
Postcondiciones	POST-1. El usuario pasa a estar loggeado en el sistema y puede acceder a las funciones de su cuenta.
Flujo normal	FN1 El usuario accede a la página de inicio de sesión, mediante el botón “Iniciar sesión”. FN2 El actor introduce sus datos de usuario e indica al sistema que quiere iniciar sesión. FN3 El sistema comprueba los datos introducidos. FN4 Si existe un usuario con los datos introducidos, el sistema logea al usuario.
Flujo alternativo 1	FA4 Si no existe un usuario con esos datos, el sistema informa al usuario y da la posibilidad de registrarse 4.7.
Excepciones	E1 El usuario ha dejado campos requeridos sin rellenar.
Prioridad	Alta

Tabla 4.8: CU-02. Iniciar sesión

En la tabla 4.9 se hace referencia al tercer caso de uso, Cerrar sesión, que describe el proceso

que ha de seguir el usuario para poder cerrar la sesión en el sistema y así pasar a no estar loggeado.

CU-03 Cerrar sesión	
Nombre e ID del Requisito	RU-04. Los usuarios con sesión abierta podrán cerrar sesión.
Actor	ACTOR-2
Descripción	Todos los usuarios loggeados deben ser capaces de cerrar sesión.
Precondiciones	PRE-1. El usuario se ha loggeado previamente en el sistema 4.7.
Postcondiciones	POST-1. El usuario pasa a no estar loggeado en el sistema y no puede acceder a las funciones de su cuenta.
Flujo normal	<p>FN1 El usuario accede a la opción "Cerrar sesión".</p> <p>FN2 El sistema finaliza la sesión del usuario y redirige al usuario a la página de inicio de sesión.</p>
Prioridad	Alta

Tabla 4.9: CU-03. Cerrar sesión

En la tabla 4.10 se hace referencia al cuarto caso de uso, Modificar perfil, que describe el proceso que ha de seguir el propio usuario, o en su defecto el administrador, para poder modificar los datos del usuario.

CU-04 Modificar perfil	
Nombre e ID del Requisito	<p>RU-05. Los usuarios administradores podrán dar de alta, baja o modificar a otros usuarios, incluido a usuarios administradores.</p> <p>RU-21. Los usuarios registrados podrán consultar sus datos.</p> <p>RU-22. Los usuarios registrados podrán modificar sus datos.</p> <p>RU-23. Los usuarios administradores podrán ver un listado de todos los usuarios registrados.</p> <p>RI-02. El sistema deberá almacenar para cada usuario al menos el nombre, la dirección de correo electrónico y la contraseña.</p>
Actor	ACTOR-2
Descripción	Todos los usuarios loggeados deben ser capaces de modificar sus datos.
Precondiciones	PRE-1. El usuario se ha loggeado previamente en el sistema 4.7.
Postcondiciones	POST-1. El usuario tiene asociados los datos actualizados.

<p>Flujo normal</p>	<p>FN1 El usuario accede a la opción “Mi perfil”.</p> <p>FN2 El sistema muestra la información de todos los datos almacenados, excluyendo la contraseña. Las diferentes direcciones de envío se observan seleccionando el botón “Direcciones de envío”.</p> <p>FN3 El usuario selecciona el botón “Editar”.</p> <p>FN4 El sistema muestra todos los datos de manera que el usuario pueda modificar el que desee. Si el número de teléfono no está introducido, será obligatorio añadirlo.</p> <p>FN5 El sistema comprueba los datos introducidos.</p> <p>FN6 Si los datos son correctos, actualiza la información del usuario.</p>
<p>Flujo alternativo 1</p>	<p>FA3 Si el usuario desea modificar la contraseña, seleccionará la opción “Cambiar contraseña”</p> <p>FA4 El sistema muestra un formulario solicitando la contraseña anterior, la nueva contraseña y repetir la nueva contraseña.</p> <p>FA5 El sistema comprueba que la contraseña anterior es correcta.</p> <p>FA6 Si la contraseña anterior es correcta, el sistema comprueba que la nueva contraseña y la confirmación de nueva contraseña son iguales.</p> <p>FA7 Si son iguales, actualiza la contraseña en la base de datos.</p>

<p>Flujo alternativo 2</p>	<p>FA3 Si se desea modificar o añadir una dirección de envío, el usuario selecciona la opción "Direcciones de envío".</p> <p>FA4 El sistema muestra todas las direcciones de envío asociadas, y da la opción de modificar cada una de las existentes o eliminarlas y de añadir una nueva dirección.</p> <p>FA5 El usuario elige la opción que desea, y tanto si es modificar una de las direcciones existentes o añadir una nueva dirección, el sistema muestra un formulario solicitando los datos.</p> <p>FA6 El sistema comprueba que los datos introducidos son correctos.</p> <p>FA7 Si los datos son correctos, actualiza la dirección en la base de datos o crea una nueva, en función de la opción elegida por el usuario.</p>
<p>Flujo alternativo 3</p>	<p>FA3 Si se desea modificar o añadir una dirección de envío, el usuario selecciona la opción "Direcciones de envío".</p> <p>FA4 El sistema muestra todas las direcciones de envío asociadas, y da la opción de modificar cada una de las existentes o eliminarlas y de añadir una nueva dirección.</p> <p>FA5 Si el usuario selecciona la opción de eliminar una dirección ya existente, el sistema muestra una pestaña de confirmación, y si esta es aceptada, elimina la dirección de la base de datos.</p> <p>FA6 Si el usuario selecciona la opción de modificar, el sistema muestra un formulario en el que el usuario puede modificar el campo que desee, y cuando este de al botón "Confirmar", la dirección se actualiza en la base de datos.</p>

Flujo alternativo 4	<p>FA3 Si se desea modificar o añadir una donación, el usuario selecciona la opción “Mis donaciones”.</p> <p>FA4 El sistema muestra todas las donaciones asociadas, y da la opción de modificar cada una de las existentes o eliminarlas, siempre y cuando aún no se haya publicado la donación como artículo, o de añadir una donación.</p> <p>FA5 Si el usuario selecciona la opción de eliminar una donación ya existente, el sistema muestra una pestaña de confirmación, y si esta es aceptada, elimina la donación de la base de datos.</p> <p>FA6 Si el usuario selecciona la opción de modificar, el sistema muestra un formulario en el que el usuario puede modificar el campo que desee, y cuando este de al botón “Confirmar”, la donación se actualiza en la base de datos.</p>
Flujo alternativo 5	<p>FA1 El usuario administrador entra en la sección “Usuarios” y selecciona el usuario al que desea modificar los datos.</p>
Excepciones	<p>E1 El usuario ha dejado campos requeridos sin rellenar.</p>
Prioridad	Alta

Tabla 4.10: CU-04. Usuario modificar perfil

En la tabla 4.11 se hace referencia al quinto caso de uso, Publicar artículos, que describe el proceso que ha de seguir el usuario administrador para poder poner a la venta productos en la página web.

CU-05 Publicar artículos	
Nombre e ID del Requisito	<p>RU-06. Los usuarios administradores podrán publicar artículos en venta directa. Podrán fijar el precio de los artículos en venta directa. Podrán modificar los datos siempre que lo deseen.</p> <p>RU-07. Los usuarios administradores podrán publicar artículos en subasta directa. Podrán fijar el precio inicial de los artículos, también fijan la fecha en la que termina el plazo de dicha subasta. Podrán modificar los datos siempre que lo deseen.</p> <p>RU-08. Los usuarios administradores podrán publicar artículos en subasta inversa. Podrán fijar el precio inicial de los artículos en venta mediante subasta inversa, el porcentaje de descuento diario y el precio mínimo de venta. Podrán modificar los datos siempre que lo deseen.</p> <p>RU-27. Los usuarios administradores podrán publicar los artículos donados, pudiendo modificar los campos que deseen antes de que sea publicado.</p> <p>RI-01. EL sistema deberá almacenar el tipo de venta de cada producto (venta directa, subasta directa o subasta inversa).</p> <p>RI-03. El sistema deberá almacenar para cada producto al menos su nombre, descripción detallada, imágenes, precio y disponibilidad.</p>
Actor	ACTOR-3
Descripción	Los usuarios administradores podrán publicar artículos siempre que lo deseen.
Precondiciones	PRE-1. El administrador se ha loggeado previamente en el sistema.
Postcondiciones	POST-1. El artículo pasa a estar publicado o modificado.
Flujo normal	<p>FN1 El administrador accede a la sección “Publicar artículo”.</p> <p>FN2 El usuario rellena todos los campos del formulario, incluido el tipo de venta y la fotografía.</p> <p>FN3 El administrador selecciona la opción aceptar.</p>
Flujo alternativo 1	<p>FA1 El administrador accede a la sección “Donaciones” y selecciona la donación que desea publicar.</p>
Excepciones	<p>E1 El usuario ha dejado campos requeridos sin rellenar.</p>
Prioridad	Alta

Tabla 4.11: CU-05. Administrador publicar artículos

En la tabla 4.12 se hace referencia al sexto caso de uso, Modificar o eliminar artículos, que describe el proceso que ha de seguir el usuario administrador para poder modificar o eliminar

los productos publicados en la página web.

CU-06 Modificar o eliminar artículos	
Nombre e ID del Requisito	<p>RU-02. Los usuarios, tanto registrados como no, podrán ver todos los artículos disponibles en la página.</p> <p>RU-06. Los usuarios administradores podrán publicar artículos en venta directa. Podrán fijar el precio de los artículos en venta directa. Podrán modificar los datos siempre que lo deseen.</p> <p>RU-07. Los usuarios administradores podrán publicar artículos en subasta directa. Podrán fijar el precio inicial de los artículos, también fijan la fecha en la que termina el plazo de dicha subasta. Podrán modificar los datos siempre que lo deseen.</p> <p>RU-08. Los usuarios administradores podrán publicar artículos en subasta inversa. Podrán fijar el precio inicial de los artículos en venta mediante subasta inversa, el porcentaje de descuento diario y el precio mínimo de venta. Podrán modificar los datos siempre que lo deseen.</p> <p>RU-13. Los usuarios administradores podrán dar de baja artículos no vendidos.</p> <p>RU-27. Los usuarios administradores podrán publicar los artículos donados, pudiendo modificar los campos que deseen antes de que sea publicado.</p> <p>RI-01. EL sistema deberá almacenar el tipo de venta de cada producto (venta directa, subasta directa o subasta inversa).</p> <p>RI-03. El sistema deberá almacenar para cada producto al menos su nombre, descripción detallada, imágenes, precio y disponibilidad.</p>
Actor	ACTOR-3
Descripción	Los usuarios administradores podrán eliminar artículos siempre que lo deseen.
Precondiciones	PRE-1. El administrador se ha loggeado previamente en el sistema.
Postcondiciones	POST-1. El artículo pasa a estar modificado o eliminado.
Flujo normal	<p>FN1 El administrador accede a la sección “Artículos”.</p> <p>FN2 El administrador selecciona el artículo que desea modificar o eliminar.</p> <p>FN3 El administrador selecciona la opción “Modificar” en caso de querer modificarlo o “Eliminar” en caso de querer eliminarlo.</p> <p>FN4 Si el administrador a elegido modificar el artículo, rellena los campos que quiere que sean modificados y le da al botón “Aceptar”. En caso de querer eliminarlo, selecciona directamente la opción “Aceptar”.</p>
Prioridad	Alta

Tabla 4.12: CU-06. Administrador modificar o eliminar artículos

En la tabla 4.13 se hace referencia al séptimo caso de uso, Donar artículo, que describe el proceso que ha de seguir el usuario para poder donar un artículo.

CU-07 Donar artículo	
Nombre e ID del Requisito	RU-09. Los usuarios registrados podrán donar artículos.
Actor	ACTOR-2
Descripción	Todos los usuarios loggeados deben poder enviar el formulario para donar artículos. El usuario rellena el formulario de donar artículos, de manera que el sistema creará una nueva solicitud que deberá ser gestionada por el administrador.
Precondiciones	PRE-1. El usuario debe estar loggeado.
Flujo normal	FN1 El usuario selecciona la opción “Donaciones” y rellena los campos del formulario. FN2 El sistema comprueba los datos introducidos. FN3 Si los datos son correctos, el sistema crea una nueva solicitud de donación que deberá ser gestionada por los administradores del sistema.
Flujo alternativo 1	FA3 Si los datos son incorrectos se informa al usuario del error y no se procede a crear la solicitud.
Excepciones	E1 El usuario ha dejado campos requeridos sin rellenar.
Prioridad	Alta

Tabla 4.13: CU-07. Donar artículo

En la tabla 4.14 se hace referencia al octavo caso de uso, Ver donaciones, que describe el proceso que ha de seguir el usuario o el administrador para ver todas las donaciones que ha publicado y como modificarlas si aún no han sido publicadas.

CU-08 Ver donaciones	
Nombre e ID del Requisito	RU-25. Los usuarios administradores podrán ver un listado de todos los artículos donados. RU-26. Los usuarios que hayan donado un artículo podrán modificarlo o eliminarlo siempre que la donación aún no se haya publicado como artículo. RU-28. Los usuarios podrán ver un listado de los artículos que haya donado.
Actor	ACTOR-2

Descripción	Los usuarios podrán ver los artículos que hayan donado, y en caso de ser el usuario administrador podrá ver un listado de todos los artículos donados. Los usuarios también podrán modificar o eliminar una donación siempre que esta aún no hay sido publicada.
Precondiciones	PRE-1. El usuario debe estar loggeado.
Flujo normal	<p>FN1 El usuario selecciona la opción “Mis donaciones” en la sección “Mi perfil” o la sección “Donaciones”.</p> <p>FN2 El sistema muestra todas las donaciones asociadas, y da la opción de modificar cada una de las existentes o eliminarlas, siempre y cuando aún no se haya publicado la donación como artículo, o de añadir una donación. En caso de ser el usuario administrador el sistema muestra todas las donaciones existentes.</p> <p>FN3 Si el usuario selecciona la opción de eliminar una donación ya existente, el sistema muestra una pestaña de confirmación, y si esta es aceptada, elimina la donación de la base de datos.</p>
Flujo alternativo 1	<p>FA3 Si el usuario selecciona la opción de modificar, el sistema muestra un formulario en el que el usuario puede modificar el campo que desee, y cuando este de al botón “Confirmar”, la donación se actualiza en la base de datos.</p>
Excepciones	<p>E1 El usuario ha dejado campos requeridos sin rellenar.</p>
Prioridad	Alta

Tabla 4.14: CU-08. Ver donaciones

En la tabla 4.15 se hace referencia al noveno caso de uso, Comprar artículo, que describe el proceso que ha de seguir el usuario para poder comprar un artículo, ya esté en subasta inversa o en venta directa.

CU-09 Comprar artículo	
Nombre e ID del Requisito	<p>RU-02. Los usuarios, tanto registrados como no, podrán ver todos los artículos disponibles en la página.</p> <p>RU-10. Los usuarios registrados podrán comprar cualquier artículo que esté a la venta directa en la plataforma.</p> <p>RU-11. Los usuarios registrados podrán comprar cualquier artículo que esté en subasta inversa.</p> <p>RU-14. Los usuarios administradores podrán ver los pedidos realizados.</p> <p>RU-15. Los usuarios podrán elegir entre múltiples opciones de pago.</p> <p>RU-16. Los usuarios podrán revisar y modificar el pedido antes de confirmarlo.</p> <p>RU-17. Los usuarios deben poder utilizar el motor de búsqueda para poder encontrar artículos.</p> <p>RU-18. Los usuarios deben poder eliminar artículos del carrito.</p> <p>RU-19. Los usuarios deben poder ver el historial de compras.</p> <p>RU-20. Los usuarios podrán añadir artículos al carrito.</p> <p>RU-24. Los usuarios registrados podrán ver los artículos que tienen en la cesta.</p> <p>RU-29. Los usuarios podrán realizar donaciones económicas.</p> <p>RU-30. Los usuarios que han ganado una puja podrán formalizar la dirección de envío y pago.</p> <p>RF-04. Una vez realizada una compra, el sistema generará un informe de compra, que deberá ser atendido por el usuario administrador.</p>
Actor	ACTOR-2
Descripción	Todos los usuarios loggeados deben ser capaces de comprar un artículo que esté a la venta directa o subasta inversa.
Precondiciones	PRE-1. El usuario debe estar identificado en el sistema.
Postcondiciones	<p>POST-1. El número de unidades disponibles debe descender el número de unidades que ha sido comprado.</p> <p>El sistema debe generar un nuevo informe de envío de artículos, que deberá ser atendido por el usuario administrador (ACTOR-03).</p>

<p>Flujo normal</p>	<p>FN1 El usuario busca el artículo deseado o navega por la tienda. En caso de querer realizar una donación económica, elige el artículo que hace referencia a la cantidad de dinero que quiere donar.</p> <p>FN2 El cliente ve los detalles del artículo (imagenes, descripción, precio...) haciendo clic en el artículo deseado.</p> <p>FN3 El usuario añade tantas unidades como desea del artículo al carrito mediante el botón "Agregar al carrito".</p> <p>FN4 El cliente accede al carrito y revisa los artículos seleccionados, pudiendo hacer modificaciones, añadir o eliminar artículos y cantidades o aplicar códigos de descuento.</p> <p>FN5 El sistema comprueba que el número de artículos solicitados está disponible.</p> <p>FN6 El usuario pasará a introducir la dirección de envío, seleccionar el método de pago y revisar el resumen del pedido.</p> <p>FN7 Si todos los campos están correctamente rellenos, el sistema almacenará la información de envío y habilitará al usuario para rellenas los datos de pago.</p> <p>FN8 Si los datos están correctamente rellenos, el sistema almacenará los datos de pago y generará un nuevo informe de envío de artículos, añadirá el pedido a la sección "Mis pedidos" del usuario y a la sección "Pedidos" del administrador.</p>
<p>Flujo alternativo 1</p>	<p>FA5 Si no hay suficientes unidades de alguno de los artículos disponibles, el sistema limitará el número de ejemplares.</p>
<p>Flujo alternativo 2</p>	<p>FA6 Si no están bien rellenos los campos, el sistema notificará al usuario e impedirá que se pase al siguiente paso.</p>
<p>Flujo alternativo 3</p>	<p>FA8 Si no están bien rellenos los campos, el sistema notificará al usuario e impedirá que se pase al siguiente paso.</p>

Flujo alternativo 4	<p>FA1 El usuario accede a la sección "Mi Perfil", "Mis pedidos" y selecciona la opción "Añadir dirección de entrega" del pedido que quiere formalizar.</p> <p>FA2 Selecciona la dirección de entrega que desea y selecciona "Continuar con el pago".</p> <p>FA3 Introduce los datos de la tarjeta y finaliza la compra.</p>
Excepciones	E1 El usuario ha dejado campos requeridos sin rellenar.
Prioridad	Alta

Tabla 4.15: CU-09. Comprar artículo

En la tabla 4.16 se hace referencia al décimo caso de uso, Pujar artículo, que describe el proceso que ha de seguir el usuario para poder pujar por un artículo y el comportamiento del sistema después de que se realice una puja.

CU-10 Pujar artículo	
Nombre e ID del Requisito	RU-12. Los usuarios registrados podrán pujar por artículos en subasta directa tantas veces como deseen al mismo artículo, mientras la subasta no esté cerrada.
Actor	ACTOR-2
Descripción	Todos los usuarios loggeados deben ser capaces de pujar por artículos que estén en subasta.
Precondiciones	PRE-1. El usuario debe estar identificado en el sistema.
Postcondiciones	POST-1. El sistema debe actualizar el precio del artículo al de la última puja.
Flujo normal	<p>FN1 El actor selecciona la opción "Pujar" del artículo que desee.</p> <p>FN2 El sistema comprueba que el usuario está loggeado, y si es así, aumenta el precio de la subasta tanto como el usuario administrador indicó en el porcentaje al publicar el artículo.</p>
Flujo alternativo 1	FA2 Si el usuario no está loggeado, el sistema redirige al usuario a la página de inicio de sesión 4.8.
Prioridad	Alta

Tabla 4.16: CU-10. Pujar artículo

4.4. Atributos de calidad

En este apartado se tienen en cuenta y analizan ciertos requerimientos, también denominados atributos de calidad, que completan aspectos concretos de la funcionalidad de la aplicación impuestos por el cliente, tales como comportamientos, restricciones o condicionantes para su correcto funcionamiento. Estos se organizan según su tipología en diferentes categorías.

RNF-01	Rendimiento	El sistema debe soportar una tasa de 500 usuarios concurrentes.
RNF-02	Rendimiento	El tiempo de respuesta del sistema no puede ser superior a 20 segundos.
RNF-03	Rendimiento	El sistema debe tener disponibilidad 24/7 (24 horas al día 7 días a la semana).
RNF-04	Rendimiento	La página debe utilizar recursos (CPU, memoria) de manera eficiente para ofrecer un rendimiento óptimo.
RNF-07	Robustez	Solo se permitirá la compra de un artículo en venta mediante subasta directa al usuario al que se le haya adjudicado.
RNF-12	Seguridad	Las contraseñas se almacenará cifradas mediante SHA-256.
RNF-13	Seguridad	Los usuarios iniciarán sesión a través del correo electrónico y contraseña.
RNF-14	Seguridad	Los datos de los usuarios deben manejarse de forma segura y cumplir con las regulaciones de privacidad.
RNF-17	Seguridad	Los datos de los usuarios y productos deben mantenerse íntegros y precisos.
RNF-18	Usabilidad	La navegación por la tienda debe ser intuitiva y organizada por categorías y subcategorías.
RNF-19	Usabilidad	El sistema debe disponer de un sistema de búsqueda con el fin de mejorar la experiencia del cliente.
RNF-20	Usabilidad	El sistema debe poseer menús despleglabes con las diferentes categorías y subcategorías de los artículos con el fin de mejorar la experiencia del cliente.
RNF-21	Usabilidad	Las páginas de productos deben mostrar imágenes de alta calidad, descripciones detalladas y precios así como información sobre disponibilidad y opciones de compra.
RNF-22	Usabilidad	Debe mostrarse el precio total del carrito con claridad.
RNF-29	Usabilidad	La interfaz de usuario debe ser intuitiva.
RNF-30	Usabilidad	La tienda debe funcionar en variedad de navegadores web.
RNF-31	Usabilidad	El sistema deberá mostrar una foto en cada artículo que permita diferenciar el tipo de venta de cada uno.

Tabla 4.17: Atributos de calidad

Capítulo 5

Diseño

5.1. Diseño de datos

5.1.1. Modelo entidad-relación

En la figura 5.1 podemos ver el diagrama entidad-relación que modela la estructura y las relaciones entre las entidades dentro del sistema.

Las entidades, mostradas como rectángulos, representan objetos del mundo real.

Los atributos, mostradas como óvalos, representan la información que se almacena de cada una de las anteriores.

Las relaciones, mostradas como rombos, representa como las entidades están conectadas entre sí. Las relaciones tienen participación (P) y cardinalidad (C), representado como (P, C); en la que ambas pueden tomar los valores 0, 1 o N, que indican si no está presente, si solo puede estar presente una vez o si puede estar presente varias veces respectivamente.

5.1.2. Modelo relacional

En la tabla 5.1 se muestra el modelo relacional. En ella, se describen las diferentes tablas que surgen del modelo entidad-relación 5.1.

Para cada una de estas tablas, se muestra su nombre, una pequeña descripción, su identificador y la clave foránea.

Las claves foráneas son atributos nuevos que surgen de las relaciones 1:N, y hacen referencia al identificador de la tabla padre.

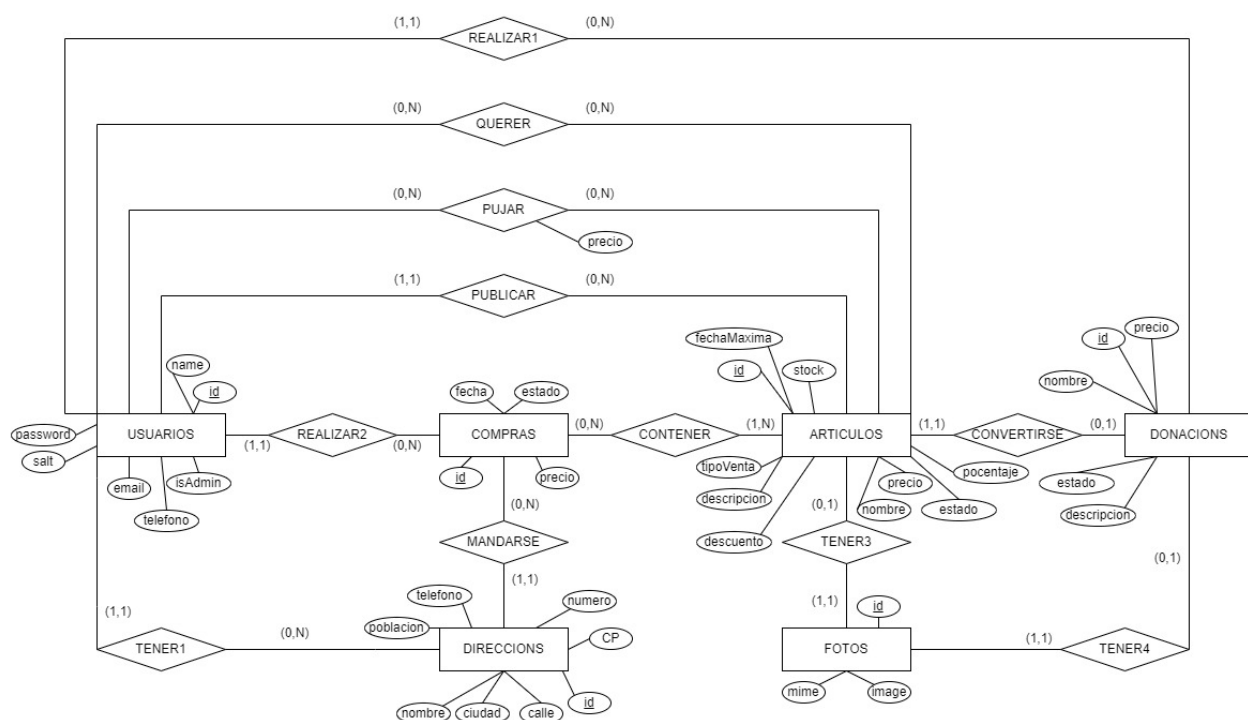


Figura 5.1: Diagrama Entidad-Relación

Nombre de la tabla	Descripción	Clave primaria	Clave/s foránea/s
ARTCOMPRA	Almacena cada artículo vendido, con el precio en ese momento y la compra en la que ha sido vendido	id	compra referencia COMPRAS (id) artículo referencia ARTICULOS(id).
ARTICULOS	Almacena cada uno de los artículos que están disponibles, tanto para su compra como para subasta. Esto se almacena en el campo tipoVenta, en el que: 1: venta directa 2: subasta directa 3: subasta inversa También se almacena el usuario que ha publicado dicho artículo.	id	usuario, referencia USUARIO (id) foto, referencia FOTOS (id)

Nombre de la tabla	Descripción	Clave primaria	Clave/s foránea/s
CARRITOS	Surge como resultado de la unión de las tablas USUARIOS y ARTICULOS. Almacena un artículo que está en venta directa o subasta inversa y el usuario indica que quiere comprar. Los datos de esta tabla asociados a un usuario en concreto se eliminan cuando este finaliza la compra o indica que ya no quiere uno de los artículos contenidos en ella. Almacena un registro por cada uno de los artículos que desea cada uno de los usuarios.	id	usuario, referencia USUARIOS (id) articulo, referencia ARTICULOS (id)
COMPRAS	Almacena cada uno de los pedidos que se han realizado y el usuario que los ha realizado.	id	direccion, referencia DIRECCIONS (id) pago, referencia PAGOS (id) usuario, referencia USUARIOS (id)
DIRECCIONS	Almacena las direcciones de envío registradas por cada usuario y el usuario al que pertenecen.	id	usuario, referencia USUARIOS (id)
DONACIONES	Almacena las donaciones hechas, el usuario que las ha realizado y la foto asociada a cada una de ellas.	id	usuario, referencia USUARIOS (id) articulo, referencia ARTICULOS (id)
FOTOS	Almacena la foto asociada a cada artículo, y el artículo al que pertenecen.	id	articulo, referencia ARTICULOS (id)
PAGOS	Almacena los métodos de pago asociados a cada usuario.	id	NA
PUJARARTICULOS	Surge como resultado de la unión de las tablas USUARIOS y ARTICULOS. Almacena un artículo que está en subasta directa y el usuario indica que quiere comprar. Los datos de esta tabla se eliminan cuando finaliza la subasta. Almacena un registro por cada usuario y cada vez que se puja por cada artículo.	id	articulo, referencia ARTICULOS (id) usuario, referencia USUARIOS (id).
USUARIOS	Almacena la información de los usuarios registrados, incluyendo si son o no administradores.	id	NA

Tabla 5.1: Modelo relacional

5.1.3. Diccionario de datos

En las tablas A.1.2, 5.2, A.1.3, A.1.4, A.1.1, A.1.5, 5.3, A.1.6, A.1.7, A.1.8, A.1.9, A.1.10, A.1.11, A.1.12, A.1.13, A.1.14 y A.1.15 se describe cada uno de los elementos de datos que se utilizan en el sistema, es decir, cada una de las tablas de la base de datos.

Cada una de las entidades tiene los siguientes campos: definición de la tabla, notas, nombre, definición, tipo, unique, null y default de cada atributo. Tipo hace referencia al tipo de dato que se almacena, unique indica si es un campo único o se puede repetir en las distintas entradas de la tabla, null indica si puede ser null o no y default si tuviera algún valor por defecto.

A modo de ejemplo se muestran las tablas 5.2 y 5.3.

COMPRAS						
Definición	Describe cada una de las compras realizadas.					
Notas						
	Nombre	Definición	Tipo	Unique	Null	Default
Atributos	id	Identificador	INTEGER	Si	No	-
	estado	Estado en el que se encuentra el pedido (realizado, preparado, enviado, entregado)	VARCHAR(255)	No	No	-
	precio	precio total del pedido en €	FLOAT	No	No	-
	fecha	fecha en la que se ha realizado el pedido	VARCHAR(255)	No	No	-

Tabla 5.2: Diccionario de datos: COMPRAS

CONTENER			
Definición	Relaciona cada compra con los artículos que contiene.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	ARTICULOS	1	1
	COMPRAS	0	N

Tabla 5.3: Diccionario de datos: CONTENER

5.2. Diagramas de clase y de secuencia

5.2.1. Diagrama de clases

En la programación estructural, los programas se organizan principalmente en torno a funciones o procedimientos que realizan tareas específicas. No hay concepto de clases ni objetos como en la programación orientada a objetos (POO). Por lo tanto, la noción de un diagrama de clases, que es una representación visual de las clases y sus relaciones en un sistema orientado a objetos, no tiene sentido en el contexto de la programación estructurada, ya que no hay clases ni objetos.

5.2.2. Diagramas de secuencia

En las figuras 5.2, 5.3, A.1, A.3, A.2, A.4, A.5, A.6, A.7, A.8, A.9, A.10, A.11 y A.12 se detallan los diagramas de secuencia para cerrar sesión, donar un artículo, eliminar un artículo, eliminar una donación aún no publicada, eliminar un usuario, iniciar sesión, modificar artículo, modificar usuario, publicar artículo, publicar donación, registrarse y ver donaciones respectivamente.

Los diagramas de secuencia son una forma de representar la interacción entre objetos en un sistema a lo largo del tiempo.

En un diagrama de secuencia, los objetos se representan como cajas rectangulares en una línea horizontal, y las interacciones entre ellos se muestran mediante flechas que indican la secuencia de mensajes intercambiados. Cada objeto tiene su propia línea de vida vertical que muestra su existencia y el tiempo durante el cual está activo.

Del mismo modo, se muestran a modo de ejemplo las figuras 5.2 y 5.3.

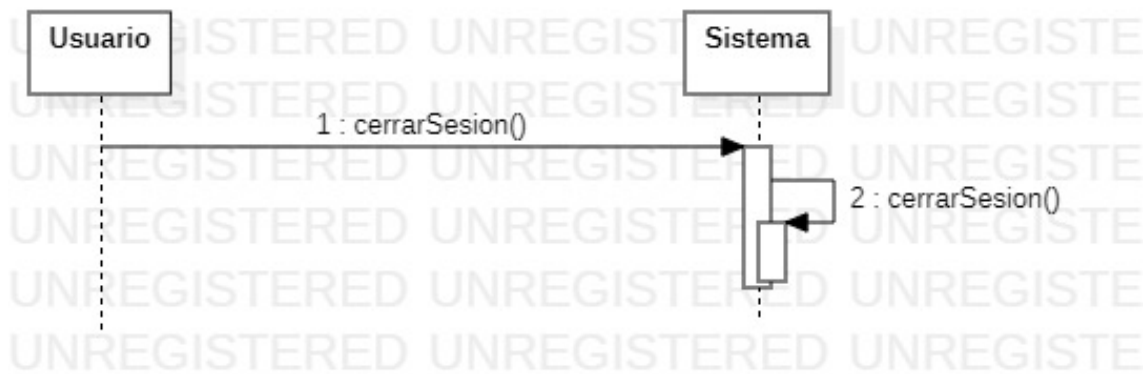


Figura 5.2: diagrama de secuencia: cerrar sesión

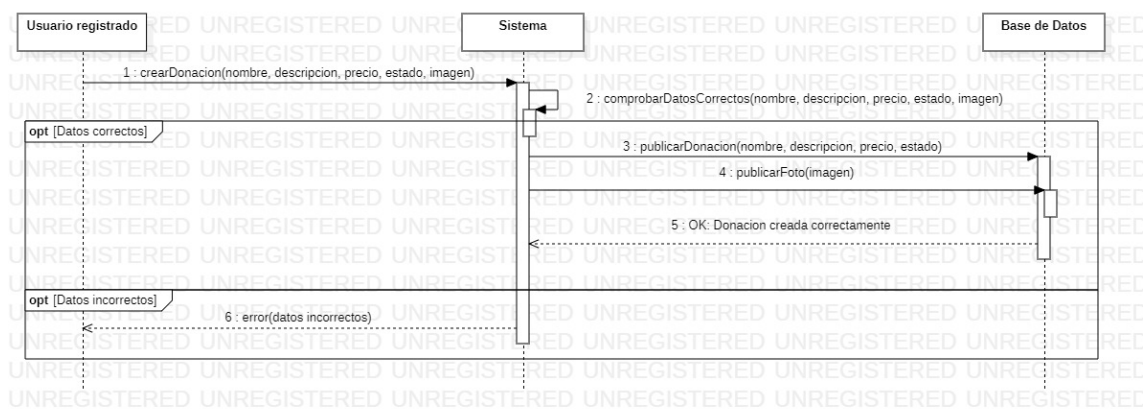


Figura 5.3: diagrama de secuencia: donar

Capítulo 6

Implementación

Para la implementación del proyecto, se ha comenzado por instalar Visual Studio Code, Xampp, Node.js y las extensiones necesarias, Sequelize y Express.

A continuación, se ha creado el proyecto en el editor de código y se ha empezado con el primer sprint, usuarios y artículos.

- El primer paso ha sido implementar la base de datos y posteriormente, las primeras funcionalidades desarrolladas han sido las de registrar y logear usuarios.
- Una vez comprobado que funcionaban correctamente, se han desarrollado las de listar, modificar y eliminar usuarios. Con éstas, se han completado todas las funcionalidades relacionadas con los usuarios.
- Para completar el sprint, se han implementado las funcionalidades de crear, ver, modificar y eliminar artículos, con sus correspondientes imágenes.

Durante el segundo sprint, donaciones y subasta directa, se ha hecho lo siguiente:

- Revisar y corregir los comentarios del usuario sobre el primer sprint.
- Se ha añadido la funcionalidad relativa a las donaciones (crear, listar, modificar, eliminar y publicar).
- Una vez que se ha comprobado que eran correctas, se han desarrollado las funcionalidades relativas a la subasta directa, que incluyen pujar, aumentar el precio del artículo o finalizar la puja.
- Para crear una nueva compra cuando finaliza una subasta se ha creado el evento "crear_compra_final_puja" 6.1 en la base de datos, que se ejecuta cada día. Este evento crea una nueva fila en la tabla "compras" cuando la fecha de fin de subasta del artículo es la fecha actual.

```
1 CREATE EVENT 'crear_compra_final_puja'
2 ON SCHEDULE EVERY 1 DAY
3 STARTS '2024-01-01 00:00:00.000000'
4 ENDS '2026-01-01 00:00:00.000000'
5 ON COMPLETION PRESERVE
6 ENABLE
7 DO
8   INSERT INTO compras('estado', 'precio', 'usuario', 'direccion', '
9     pago', 'createdAt', 'updatedAt')
10  SELECT 'Pendiente', pa.precio, pa.usuario, NULL, NULL,
11    CURRENT_TIMESTAMP, CURRENT_TIMESTAMP
12 FROM articulos a
13 JOIN pujararticulos pa ON pa.articulo = a.id
14 WHERE DATE(a.fechaMaxima) = CURRENT_DATE()
15 ORDER BY pa.id DESC
16 LIMIT 1;
```

Listing 6.1: Evento crear_compra_final_puja

En el tercer sprint, venta directa y subasta inversa, se han:

- Revisado y corregido los comentarios sobre el sprint anterior, análogamente a como se hizo en el segundo con el primero.
- Se han desarrollado el resto de funcionalidades, comunes tanto a la venta directa como a la subasta inversa, como son añadir los artículos al carrito, elegir la dirección de entrega, el método de pago y reducir el número de unidades disponibles.
- Para bajar el precio de los artículos en subasta inversa de manera automática y periódica, en este caso cada día, se ha creado el evento "actualizar_precios_subasta_inversa" 6.2 en la base de datos.

```
1 CREATE EVENT 'actualizar_precio_subasta_inversa'
2 ON SCHEDULE EVERY 1 DAY
3 STARTS '2024-01-01 00:00:00.000000'
4 ENDS '2026-01-01 00:00:00.000000'
5 ON COMPLETION PRESERVE
6 ENABLE
7 DO
8   UPDATE articulos
9   SET precio = precio - (precio * porcentaje / 100)
10  WHERE tipoVenta = 2;
```

Listing 6.2: Evento actualizar_precios_subasta_inversa

Durante toda las fases del proyecto se han ido realizando copias de seguridad periódicas de la base de datos, para asegurar su integridad.

A continuación, se explica la estructura del proyecto que se encuentra en el repositorio (link al repositorio) a nivel de archivos.

La figura 6.1 muestra las carpetas y archivos que se encuentran en la raíz. Podemos destacar las carpetas controllers, migrations, models, routes y views.

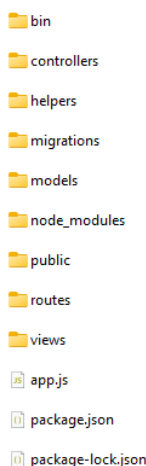


Figura 6.1: Estructura carpetas raíz

En la figura 6.2 vemos los distintos controladores que forman la aplicación. En cada uno de los módulos encontramos funciones como index, load, show, create, update o destroy; que se utilizan para listar todos registros de la tabla, obtener el objeto asociado al identificador pasado como parámetro, mostrar la información de un objeto concreto, crear nuevos registros o eliminar registros ya existentes respectivamente; además de las funciones específicas de cada controlador, como en el caso de los artículos, actualizar el precio del artículo cuando se realiza una puja por el mismo.

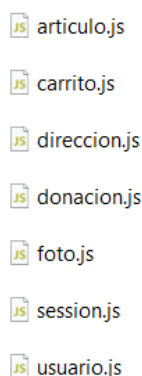


Figura 6.2: Controladores

En la figura 6.3 se encuentran los ficheros de creación de las tablas de la base de datos, y en cada uno se encuentran los distintos atributos que forman la tabla y sus características.

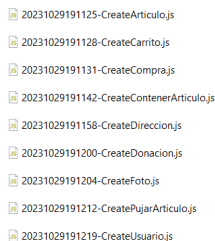


Figura 6.3: Migraciones

En la figura 6.4 podemos ver el contenido de la carpeta models, donde se encuentran los modelos, que son representaciones de las tablas de la base de datos en forma de objetos JavaScript. Sequelize utiliza esta información para generar consultas SQL y realizar operaciones en la base de datos de manera eficiente y segura.

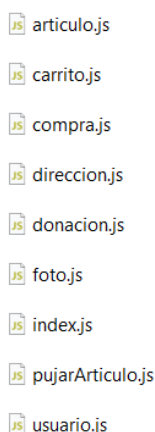


Figura 6.4: Modelos

En la figura 6.5 se ve el fichero index.js, en el que se definen las rutas de la aplicación, es decir, las URLs específicas que el servidor maneja y responde a través de funciones que se encuentran en la carpeta controllers.

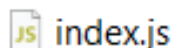


Figura 6.5: Rutas

En la figura 6.6 se ve como están estructuradas las vistas, como el ejemplo de la figura 6.7, donde se encuentran las vistas de los usuarios. En cada una de las carpetas se almacenan las plantillas de las vistas que serán renderizadas con los datos que devuelvan los controladores.

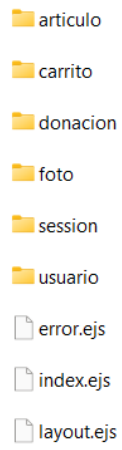


Figura 6.6: Vistas

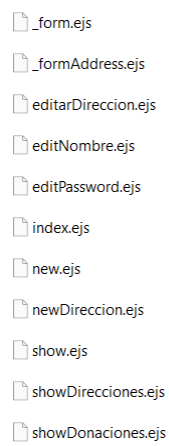


Figura 6.7: Usuarios

Capítulo 7

Pruebas

En este capítulo vamos a hablar sobre las pruebas que se han realizado. Se han hecho pruebas de caja negra y caja blanca.

7.1. Caja blanca

Las pruebas de caja blanca son aquellas en las que el tester sabe cómo está estructurado internamente el código fuente. El objetivo principal es probar la lógica interna del programa y asegurar que todas las ramas del código han sido probadas al menos una vez.

Para ello, se ha hecho lo siguiente:

- Añadir logs que se muestran en la terminal al ejecutar el código.
- Validar la conexión con la base de datos.
- Validar que los datos introducidos en los formularios tienen el formato esperado.
- Probar todos los bucles y condicionales.
- Gestionar errores en tiempo de ejecución mediante el manejo de excepciones.

7.2. Caja negra

En estas pruebas el tester no tiene conocimientos acerca del código fuente de la página. El objetivo principal de estas pruebas es probar el comportamiento del sistema.

Las pruebas de caja negra realizadas se muestran en las tablas 7.1, 7.2, 7.3, A.3.1, A.3.2, A.3.3, A.3.4, A.3.5, A.3.6 y A.3.7.

A modo de ejemplo se muestran las tablas 7.1, 7.2 y 7.3.

PCN-01 Registrarse	
Objetivo	Confirmar que un usuario no se puede registrar si ya existe un usuario con el mismo email, si el email no tiene el formato esperado o si las contraseñas no coinciden. Si ninguna de estas condiciones se cumplen el usuario podrá registrarse.
Precondiciones	N/A
Paso	Resultado esperado
Crear un usuario con un email que aún no esté registrado	Usuario registrado correctamente
Crear un usuario con el email introducido en el paso anterior	Error al registrar usuario
Crear un usuario con email sin formato email	Error al registrar usuario
Crear un usuario con la contraseña distinta al campo confirmar contraseña	Error al registrar usuario
Iniciar sesión como usuario administrador (email: admin@gmail.com contraseña:1234) , acceder al apartado “Usuarios“, seleccionar “Crear nuevo usuario“ y rellenar los datos solicitados	Usuario registrado correctamente
Resultado final	OK

Tabla 7.1: PCN-01 Registrarse

PCN-02 Iniciar sesión	
Objetivo	Un usuario solo podrá iniciar sesión cuando ese email esté registrado y la contraseña sea correcta.
Precondiciones	El usuario debe estar registrado
Paso	Resultado esperado
Iniciar sesión con el usuario <code>eva1@gmail.com</code>	El sistema no permite al usuario loggarse
Iniciar sesión con el usuario <code>eva@gmail.com</code> y contraseña 123	El sistema no permite al usuario loggarse
Iniciar sesión con el usuario <code>eva@gmail.com</code> y contraseña 1234	El usuario ha sido loggeado correctamente
Resultado final	OK

Tabla 7.2: PCN-02 Iniciar sesión

PCN-03 Cerrar sesión	
Objetivo	Cuando haya un usuario loggeado, al presionar el botón “Cerrar sesión“, debe pasar a no estar loggeado.
Precondiciones	El usuario debe estar loggeado (por ejemplo, email: <code>eva@gmail.com</code> contraseña:1234)
Paso	Resultado esperado
Esperar 5 minutos	El usuario se desloggea automáticamente
Volver a iniciar sesión y seleccionar el botón “Cerrar sesión“	El usuario se ha desloggeado automáticamente
Resultado final	OK

Tabla 7.3: PCN-03 Cerrar sesión

Capítulo 8

Manuales de instalación y de usuario

8.1. Manual de instalación

Para desplegar la web en el ordenador, es necesario instalar XAMPP, Node.js y un editor de código fuente como Visual Studio Code para poder lanzar la aplicación.

1. Instalar XAMPP, Node.js y Visual Studio Code

Es necesario instalar XAMPP para poder arrancar el servidor de base de datos de MySQL.

- Para ello accederemos a las páginas oficiales <https://www.apachefriends.org/es/index.html> para descargar XAMPP, <https://nodejs.org/en> para descargar Node.js y <https://code.visualstudio.com/> para Visual Studio Code.
- Descargaremos la versión necesaria para el sistema operativo que dispone el ordenador, la descarga empezará automáticamente.
- Una vez descargado, ejecutamos los archivos y seguimos con el proceso de instalación hasta el final.

2. Despliegue de la aplicación

- Abrimos XAMPP y le damos a los dos botones “Start” que aparecen tanto a la derecha de MySQL como de Apache, como aparece en la figura 8.1.
- Desde el navegador accedemos a <http://localhost/phpmyadmin/>, seleccionamos la opción “Importar”, “Seleccionar archivo”, como aparece en la figura 8.2 y seleccionamos el documento wallaorg.sql descargado del repositorio.
- Descomprimos la carpeta Wallaorg con el código fuente en cualquier parte del ordenador, por ejemplo en el escritorio, y la abrimos con Visual Studio Code (File->Open Folder como en la figura 8.3y seleccionamos la carpeta Wallaorg), ejecutamos en la terminal (Terminal->New Terminal, como en la figura 8.4) los comandos “npm install” (figura 8.5) para asegurar que están todas las dependencias instalados y “npm start” (figura 8.6) para lanzar el servidor.

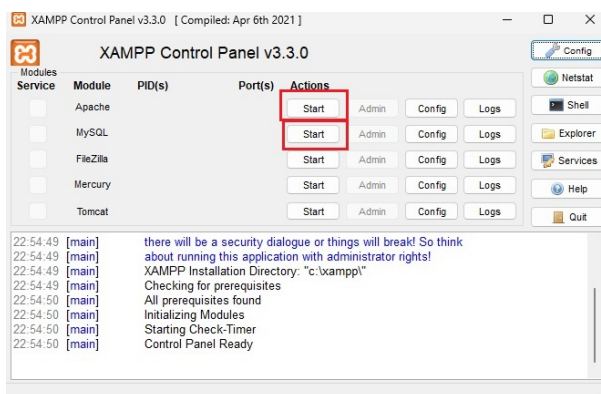


Figura 8.1: XAMPP

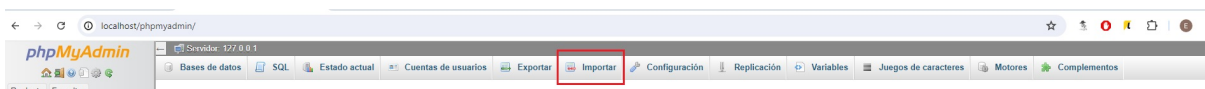


Figura 8.2: Importar base de datos

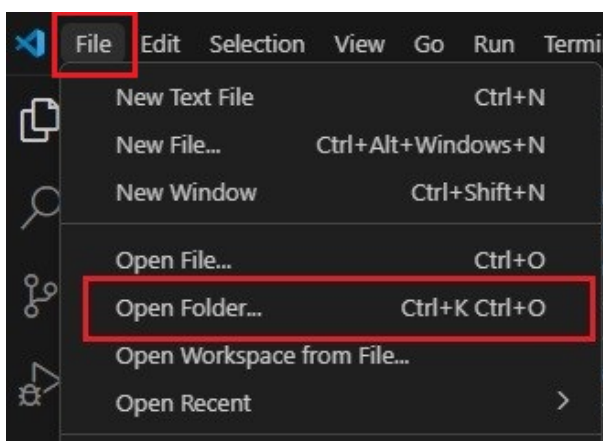


Figura 8.3: Open Folder Visual Studio Code

8.2. Manual de usuario

En este manual, vamos a explicar todas las formas en que un usuario puede interactuar con el sistema. Las acciones que puede realizar son:

- Acceder a la web.
- Registrarse.
- Iniciar sesión.
- Ver artículos disponibles.
- Donar artículos.

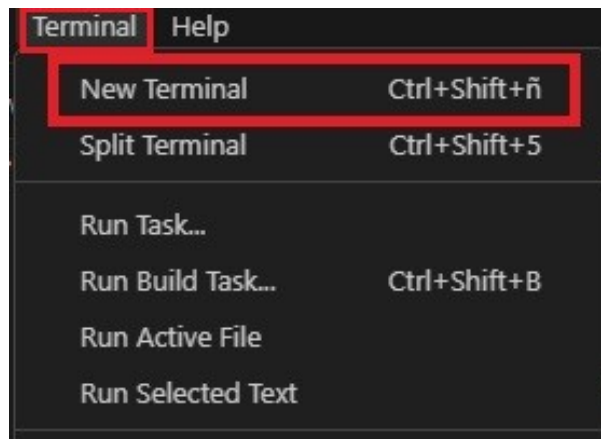


Figura 8.4: New Terminal Visual Studio Code

```
PS C:\Users\Eva\Desktop\repo_git\tfg\Wallaorg> npm install
```

Figura 8.5: npm install Visual Studio Code

```
PS C:\Users\Eva\Desktop\repo_git\tfg\Wallaorg> npm start
```

Figura 8.6: npm start Visual Studio Code

- Comprar.
- Pujar en una subasta directa.
- Ver y modificar perfil.

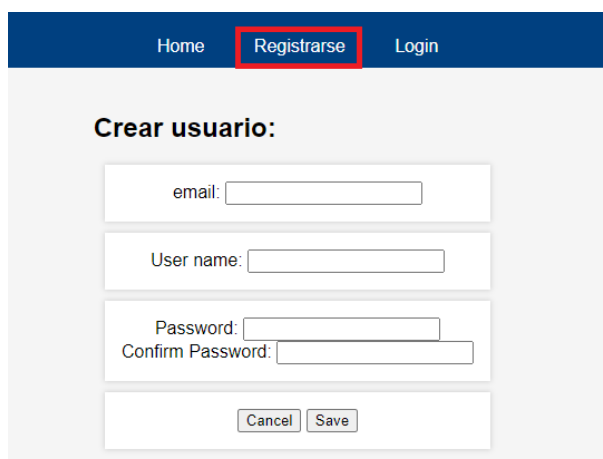
8.2.1. Acceder a la web

El acceso a la web es el primer paso que todo usuario debe realizar para llevar a cabo cualquier acción posterior. En la fase de preproducción, se realiza tecleando en cualquier navegador web la dirección `http://localhost:3000`.

8.2.2. Registrarse

Para registrarse, el usuario debe pulsar en el botón registrarse. Desde aquí se le dirigirá a una pantalla en la que debe completar los campos con sus datos personales (nombre, email y contraseña). Una vez completados, deberá confirmar los datos en el botón “Aceptar“. Desde aquí, se le dirigirá a la pantalla de inicio.

El sistema lanzará una advertencia si el correo ya está registrado.

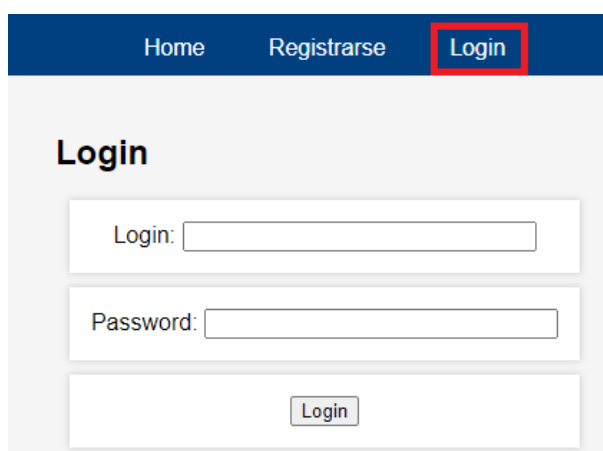


The screenshot shows a navigation bar with three links: "Home", "Registrarse", and "Login". The "Registrarse" link is highlighted with a red box. Below the navigation bar is a form titled "Crear usuario:". The form contains four input fields: "email:", "User name:", "Password:", and "Confirm Password:". At the bottom of the form are two buttons: "Cancel" and "Save".

8.2.3. Iniciar sesión

El inicio de sesión se realiza pulsando en el botón "Iniciar sesión" e introduciendo a continuación el email registrado y la contraseña. Si se introdujese una contraseña errónea, el sistema advertirá el error y solicitará que vuelva a introducirse.

Solamente una vez iniciada la sesión, al usuario se le permitirá donar o comprar artículo o pujar en una subasta directa. Para ver los artículos disponibles, no es necesario el inicio de sesión.

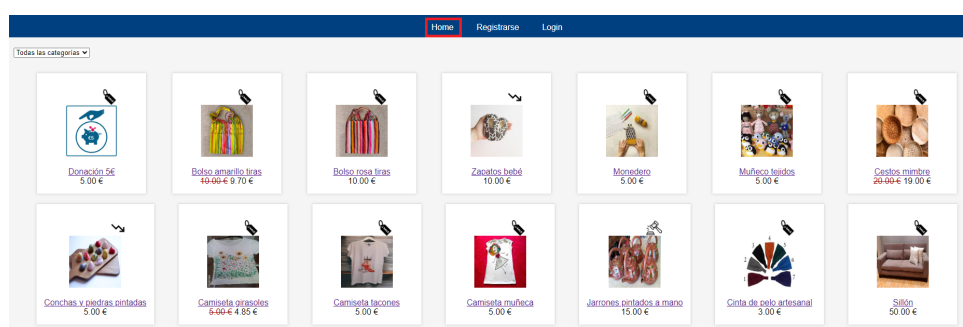


The screenshot shows a navigation bar with three links: "Home", "Registrarse", and "Login". The "Login" link is highlighted with a red box. Below the navigation bar is a form titled "Login". The form contains two input fields: "Login:" and "Password:". At the bottom of the form is a button labeled "Login".

8.2.4. Ver artículos disponibles

En la pantalla principal de la página web se muestran todos los artículos disponibles. A esta página se puede acceder desde la dirección principal o pulsando el botón inicio desde cualquier otra página del dominio.

En esta página se muestra para cada artículo, además de su imagen, nombre y precio, un icono que identifica la forma de adquisición (venta directa, subasta o subasta inversa). Pulsando en cualquier artículo, se accede a una información más detallada del mismo.



8.2.5. Donar artículos

Para donar un artículo, el usuario registrado y con sesión iniciada debe acceder a la opción “Donaciones” del menú superior de la página. Una vez presionado, accederá a un formulario que deberá cumplimentar. Para completar la donación, necesariamente deberá subir una imagen e informar del nombre del artículo, descripción, estado y precio propuesto.

Posteriormente, el administrador decidirá si publicar el artículo, el método de venta y el precio (precio final en el caso de venta directa o precio inicial para las subastas directa e inversa).

Nueva donación:

Nombre:

Descripción:

Precio recomendado: €

Estado:

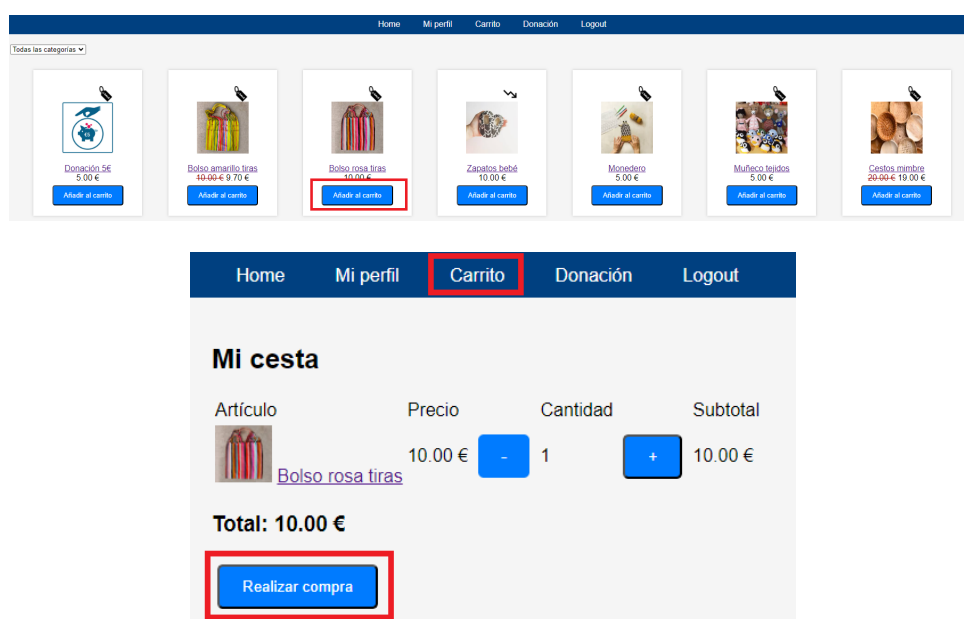
Seleccione las imágenes del producto (Max size 2MB): Ningún archivo seleccionado

[Mis donaciones](#)

8.2.6. Comprar

Para adquirir un artículo desde las opciones de venta directa o subasta inversa, el usuario loggeado debe simplemente pulsar sobre el botón “Añadir al carrito” de los artículos en los que esté interesado.

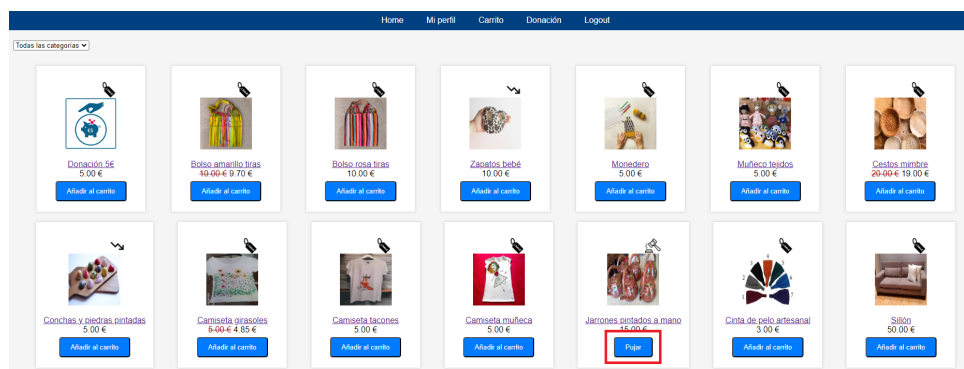
Después debe acceder a su “Carrito” desde el menú superior, revisarlo y presionar en “Realizar compra”, donde deberá elegir la dirección de envío y método de pago. Ambas opciones se guardan en los datos del usuario por si quiere reutilizarlas en futuras compras.



8.2.7. Pujar en subasta directa

Los artículos en subasta directa muestran el precio solicitado para realizar la siguiente puja. Para pujar, es suficiente con que el usuario loggeado presione sobre el botón “Pujar” y después confirmar la puja.

Una vez finalizado el plazo de la subasta, el sistema adjudicará el artículo al último pujador y retirará el artículo de la página.



8.2.8. Ver y modificar perfil

Desde cualquier página del dominio, el usuario loggeado puede acceder a la opción “Mi perfil” del menú superior. Desde ella, puede consultar sus datos de usuario, sus direcciones y el historial de artículos donados y adquiridos. También podrá modificar todos sus datos, excepto el correo electrónico, y la descripción de las donaciones que aún no hayan sido publicadas. Las direcciones pueden eliminarse.



8.3. Manual de administrador

El Manual del administrador describe las acciones específicas que puede realizar el administrador del sistema. Son las siguientes:

- Publicar artículo.
- Modificar artículos publicados.
- Ver y modificar listado de usuarios.

8.3.1. Publicar artículo

El administrador puede publicar artículos donados en la web por otros usuarios registrados o artículos procedentes de otros orígenes, por ejemplo elaborados en los talleres propios.

En el menú superior, el administrador dispone de la opción “Ver donaciones“. Al acceder a ella, se le muestra un listado de todas las donaciones. Las no publicadas, tienen un botón “Publicar“. Al presionar en este botón, se accede a un formulario con la información proporcionada por el donante y, por defecto, la opción de “venta directa“. Todo ello puede ser modificado por el administrador si así lo desea antes de que se produzca la publicación, que se efectúa presionando el botón “Aceptar“.

Si el artículo no ha sido donado el administrador deberá presionar sobre la opción “Publicar artículo“ del menú principal, rellenar todos los campos del formulario que le aparecerá en ese momento y presionar la opción “Aceptar“ para finalizar la publicación.

Home **Publicar artículos** Usuarios Donaciones Pedidos Logout

Añadir artículo:

Nombre:

Descripción:

Stock:

Precio: €

Descuento: %

Estado:

Tipo venta: Venta directa Subasta directa Subasta inversa

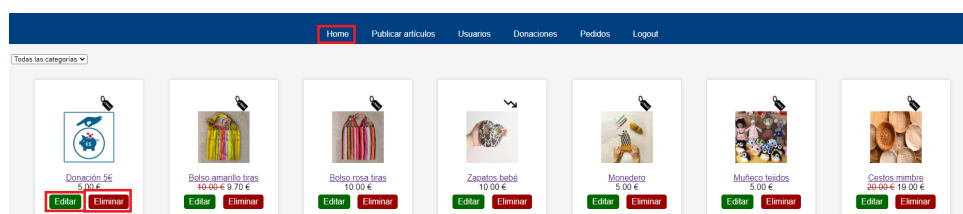
Seleccione las imágenes del producto (Max size 2MB): Ningún archivo seleccionado

8.3.2. Modificar artículos publicados

Desde la página de inicio, donde se muestran todos los artículos, el administrador dispone de dos opciones para cada artículo. Son: “Modificar” y “Eliminar”.

Al presionar la opción “Modificar”, el administrador accede al formulario del artículo publicado y puede modificar todos sus campos.

Con la opción “Eliminar” el administrador puede eliminar un artículo publicado. Se le solicitará confirmación para realizar esta acción.



8.3.3. Ver y modificar listado de usuarios

El administrador tiene acceso a la opción “Clientes” del menú superior. Al pulsar en ella, se le presenta un listado de todos los usuarios registrados y un botón para crear nuevos usuarios.

Al presionar sobre un usuario registrado, puede ver todos sus datos, excepto la contraseña, y modificarlos. También tiene la posibilidad de eliminar a cualquier usuario.

Para crear un nuevo usuario, debe presionar en el botón correspondiente y rellenar todos los datos del formulario. Por defecto, el usuario creado será un usuario no administrador, pero puede asignarle este rol marcando la opción correspondiente.



Home Publicar artículos **Usuarios** Donaciones Pedidos Logout

Usuarios

[Crear nuevo usuario](#)

admin@gmail.com	(admin)	Editar Borrar
eva@gmail.com		Editar Borrar
user1@gmail.com		Editar Borrar

8.3.4. Ver pedidos

El administrador puede consultar todos los pedidos realizados, seleccionando la opción “Ver pedidos” del menú superior.



Home Publicar artículos Usuarios Donaciones **Pedidos** Logout

Pedidos

Todos los pedidos ▼

Fecha: 10 jun 2024, 22:21
Usuario: EVA MÁIQUEZ PEREZ, 696837360
Dirección: Paseo de la Constitución, 32, 40100, Segovia, La Granja

Cantidad	Artículo	Precio
2x	Donación 5€	5.00€
1x	Zapatos bebé	10.00€

Total: 20 €
Estado: Enviado

Fecha: 10 jun 2024, 22:34
Usuario: Sofia Perez Lopez, 696837361
Dirección: Calle de abajo, 2, 28000, Madrid, Madrid

Cantidad	Artículo	Precio
1x	Bolso rosa tiras	10.00€

Total: 10 €
Estado: Enviado

Capítulo 9

Conclusiones y líneas futuras

Este TFG ha logrado cumplir los objetivos académicos que se planteaban y que básicamente consistían en demostrar que la alumna había alcanzado los conocimientos y madurez suficientes para afrontar el ejercicio profesional, pues se ha desarrollado una aplicación web plenamente funcional, de una complejidad similar a la de otras aplicaciones que podría tener que realizar la alumna durante su vida laboral.

Para el desarrollo de la aplicación, ha sido necesario emplear un gran número de herramientas nuevas, como Node.js, Express, Sequelize, etc. lo que ha supuesto un esfuerzo adicional, pero a la vez ha servido para completar la formación. Los diversos conocimientos adquiridos durante los estudios de grado han permitido una curva de aprendizaje muy rápida en estas herramientas.

La elaboración de la memoria también ha requerido un esfuerzo nada desdeñable y ha servido para asentar y poner en práctica muchos de los conocimientos adquiridos durante la carrera y profundizar en otros.

Todo ello, unido a la necesidad de cierto aprendizaje autónomo que siempre conlleva este tipo de trabajos, aporta un valor añadido muy significativo en los estudios universitarios.

Respecto a la aplicación en sí, se ha logrado desarrollar en su totalidad, cumpliendo todos los requisitos planteados. Se contempla desde el registro de usuarios y la donación de artículos y dinero, hasta la venta por los tres métodos previstos.

Si bien, tal y como se plantea en el apartado dedicado al estado del arte, existen muchas aplicaciones de venta electrónica y también otras orientadas a gestionar donaciones, esta aplicación combina ambos aspectos, lo que le confiere un elevado grado de originalidad. Combinar las opciones de venta directa, subasta y subasta inversa también supone una cierta innovación, pues no se conocen otras aplicaciones que lo hagan.

A pesar de que como se ha indicado, la aplicación es plenamente operativa, todavía se podría completar añadiendo otras funcionalidades que han quedado fuera del alcance de este TFG, por la necesaria limitación del mismo y que se refieren como posibles ampliaciones. Algunas de estas funcionalidades extras podrían ser: aplicar códigos de descuento, gestionar las devoluciones, implementar la gestión de los envíos y, por supuesto, desarrollar una aplicación móvil para un uso más cómodo desde este tipo de dispositivos.

Anexo

A.1. Diccionarios de datos

FOTOS						
Definición	Describe cada una de las fotos asociadas a cada artículo.					
Notas						
	Nombre	Definición	Tipo	Unique	Null	Default
Atributos	id	Identificador	INTEGER	Si	No	-
	mime	extensión de la imagen	VARCHAR(255)	No	No	-
	image	numero de casa	LONGBLOB	No	No	-

Tabla A.1.1: Diccionario de datos: FOTOS

ARTICULOS						
Definición	Describe cada uno de los artículos disponibles.					
Notas						
	Nombre	Definición	Tipo	Unique	Null	Default
Atributos	id	Identificador	INTEGER	Si	No	-
	nombre	Nombre que se le asocia artículo	VARCHAR(255)	No	No	-
	stock	Unidades disponibles	INTEGER	No	No	1
	precio	precio del artículo, en €	FLOAT	No	No	-
	descripcion	descripción del artículo	VARCHAR(255)	No	No	-
	descuento	descuento aplicado al artículo	INTEGER	No	Si	-
	estado	estado del artículo	VARCHAR(255)	No	No	-
	tipoVenta	tipo de venta del producto en el que: 1: venta directa 2: subasta directa 3: subasta inversa	INTEGER	No	No	1
	porcentaje	porcentaje de bajada de precio cada día	INTEGER	No	Si	-
	fechaMaxima	En el caso de los artículos en subasta inversa, fecha en la que el artículo es eliminado si no se ha vendido anteriormente	DATETIME	No	Si	-

Tabla A.1.2: Diccionario de datos: ARTICULOS

DIRECCIONS						
Definición	Describe cada una de las direcciones asociadas a cada usuario.					
Notas						
	Nombre	Definición	Tipo	Unique	Null	Default
Atributos	id	Identificador	INTEGER	Si	No	-
	nombre	nombre de la persona que se pre-vee que recogerá el pedido	VARCHAR(255)	No	No	-
	calle	nombre de la calle	VARCHAR(255)	No	No	-
	numero	numero de casa	VARCHAR(255)	No	Si	-
	ciudad	ciudad en la que se encuentra la dirección	INTEGER	No	Si	-
	poblacion	población en la que se encuentra la dirección	VARCHAR(255)	No	Si	-
	CP	código postal en el que se encuentra la dirección	VARCHAR(255)	No	Si	-
	telefono	teléfono de la persona que se pre-vee que recogerá el pedido	VARCHAR(255)	No	Si	-

Tabla A.1.3: Diccionario de datos: DIRECCIONS

Donacions						
Definición	Describe cada una de las donaciones realizadas.					
Notas						
	Nombre	Definición	Tipo	Unique	Null	Default
Atributos	id	Identificador	INTEGER	Si	No	-
	nombre	nombre elegido para la donacion	VARCHAR(255)	No	No	-
	descripcion	descripción del artículo donado	VARCHAR(255)	No	No	-
	precio	precio recomendado de venta del artículo	VARCHAR(255)	No	No	-
	estado	estado en el que se encuentra el artículo. Puede ser: Sin estrenar Nuevo Como nuevo En buen estado En condiciones aceptables Lo ha dado todo	VARCHAR(255)	No	No	Nuevo

Tabla A.1.4: Diccionario de datos: DONACIONES

USUARIOS						
Definición	Describe cada una de las direcciones asociadas a cada usuario.					
Notas						
	Nombre	Definición	Tipo	Unique	Null	Default
Atributos	id	Identificador	INTEGER	Si	No	-
	email	email con el que se ha registrado el usuario	VARCHAR(255)	No	No	-
	password	contraseña encriptada de acceso a la página web	VARCHAR(255)	No	No	-
	name	nombre asociado al usuario	VARCHAR(255)	No	Si	-
	telefono	número de teléfono asociado al usuario	VARCHAR(255)	No	Si	-
	isAdmin	indica si el usuario tiene o no permisos de administrador	TINYINT(1)	No	No	0
	salt	cadena única generada de manera aleatoria que se utiliza para encriptar la contraseña	VARCHAR(255)	No	No	-

Tabla A.1.5: Diccionario de datos: USUARIOS

CONVERTIRSE			
Definición	Relaciona cada usuario con la compra.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	ARTICULOS	1	1
	DONACIONES	0	1

Tabla A.1.6: Diccionario de datos: CONVERTIRSE

MANDARSE			
Definición	Relaciona cada compra con la dirección a la que se envía.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	DIRECCIONS	1	1
	COMPRAS	0	N

Tabla A.1.7: Diccionario de datos: MANDARSE

PUBLICAR			
Definición	Relaciona cada artículo con el usuario que lo publica.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	USUARIOS	1	1
	ARTICULOS	0	N

Tabla A.1.8: Diccionario de datos: PUBLICAR

PUJAR			
Definición	Relaciona cada usuario con el artículo al que puja.		
Notas	Esta relación se establece cuando el artículo está en subasta directa.		
	Nombre	Participación	Cardinalidad
Atributos	USUARIOS	0	N
	ARTICULOS	0	N

Tabla A.1.9: Diccionario de datos: PUJAR

QUERER			
Definición	Relaciona cada usuario con el artículo que desea.		
Notas	Esta relación se establece cuando el artículo está en venta directa o en subasta inversa.		
	Nombre	Participación	Cardinalidad
Atributos	USUARIOS	0	N
	ARTICULOS	0	N

Tabla A.1.10: Diccionario de datos: QUERER

REALIZAR1			
Definición	Relaciona cada usuario con la compra.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	USUARIOS	1	1
	COMPRAS	0	N

Tabla A.1.11: Dicionario de datos: REALIZAR1

REALIZAR2			
Definición	Relaciona cada usuario con la donación.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	USUARIOS	1	1
	DONACIONES	0	N

Tabla A.1.12: Dicionario de datos: REALIZAR2

TENER1			
Definición	Relaciona cada usuario con la dirección.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	USUARIOS	1	1
	DIRECCIONES	0	N

Tabla A.1.13: Dicionario de datos: TENER1

TENER3			
Definición	Relaciona cada usuario con el método de pago.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	ARTICULOS	0	1
	FOTOS	1	1

Tabla A.1.14: Dicionario de datos: TENER3

TENER4			
Definición	Relaciona cada usuario con el método de pago.		
Notas			
	Nombre	Participación	Cardinalidad
Atributos	DONACIONES	0	1
	FOTOS	1	1

Tabla A.1.15: Diccionario de datos: TENER4

A.2. Diagramas de secuencia

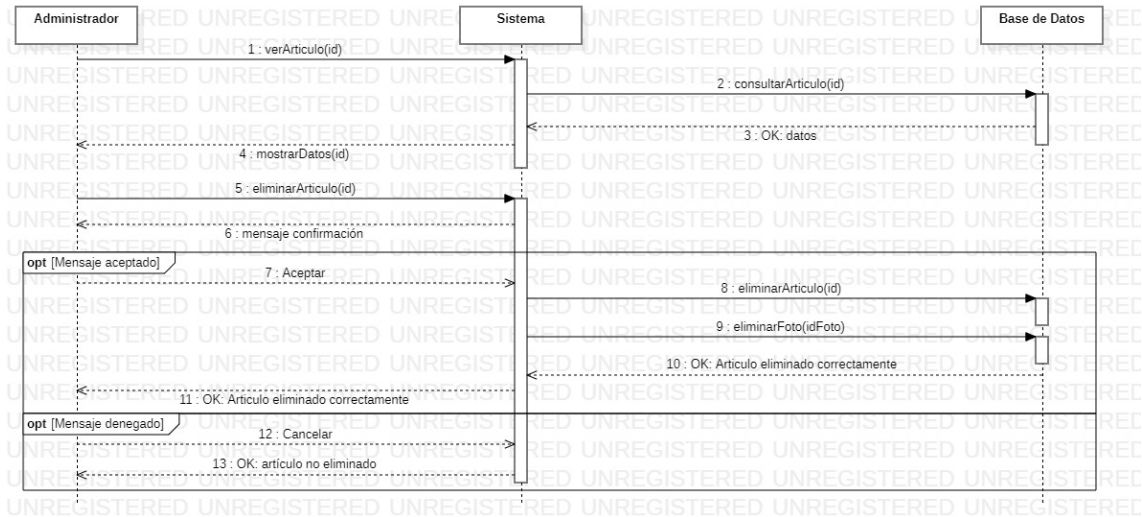


Figura A.1: diagrama de secuencia: eliminar artículo

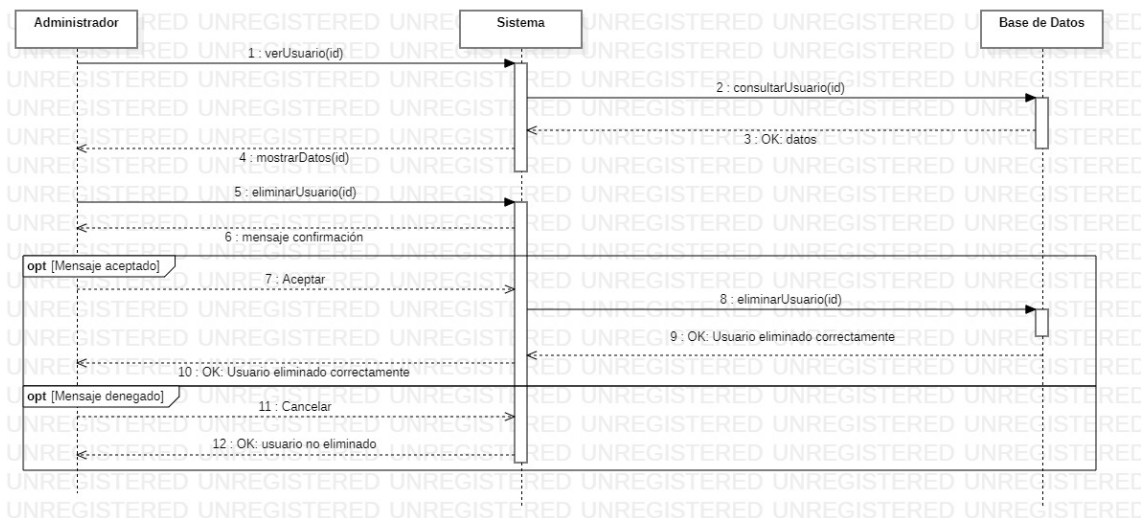


Figura A.2: diagrama de secuencia: eliminar usuario

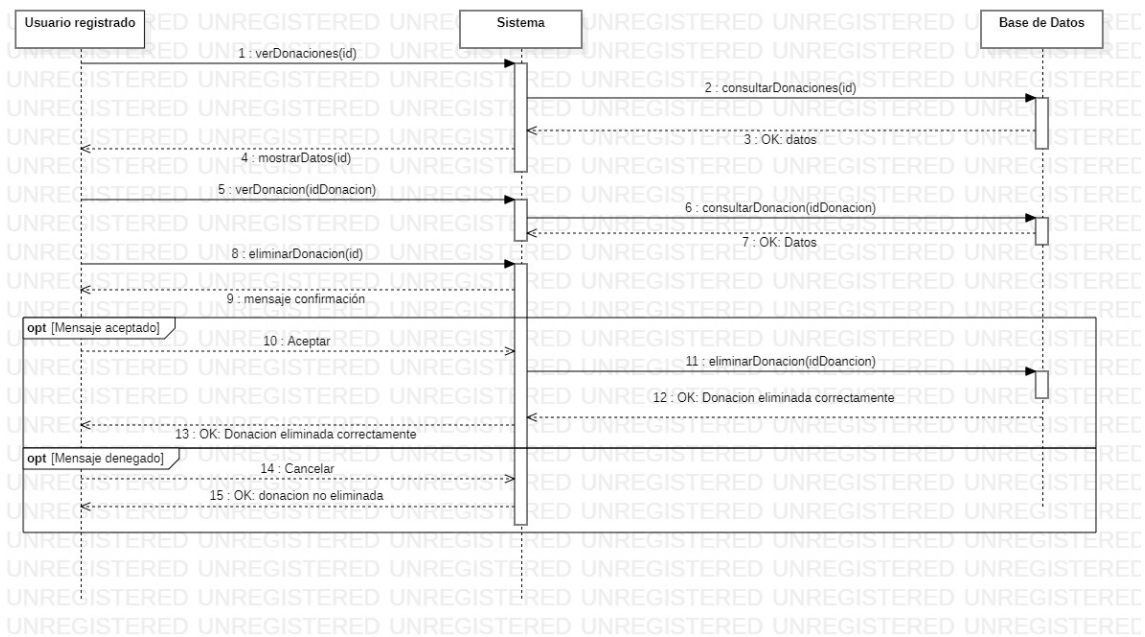


Figura A.3: diagrama de secuencia: eliminar donación

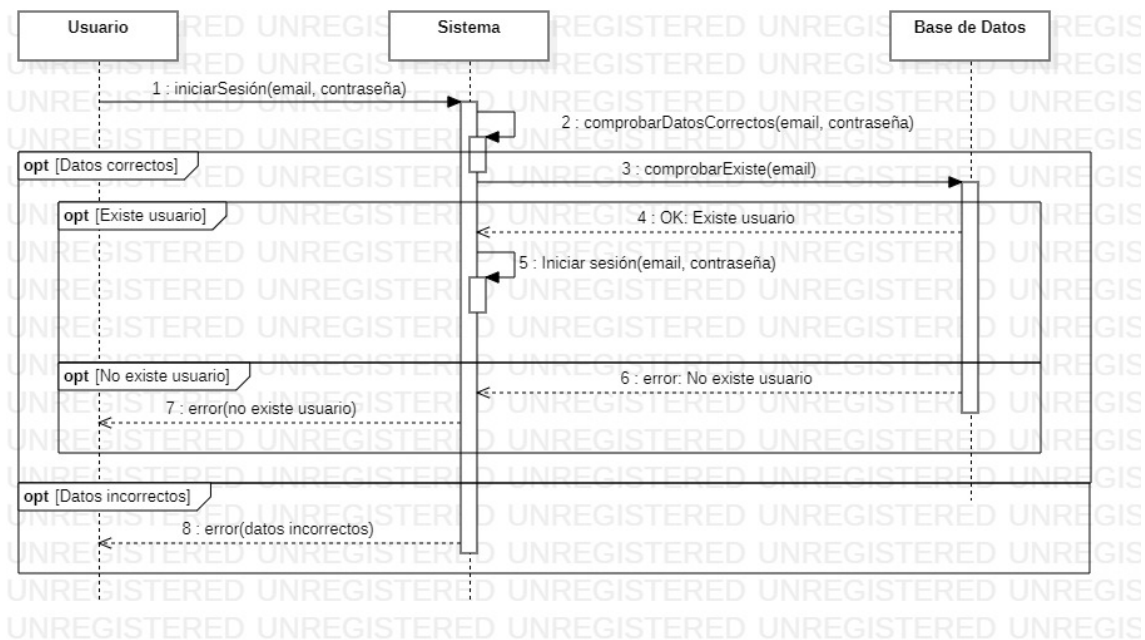


Figura A.4: diagrama de secuencia: iniciar sesión

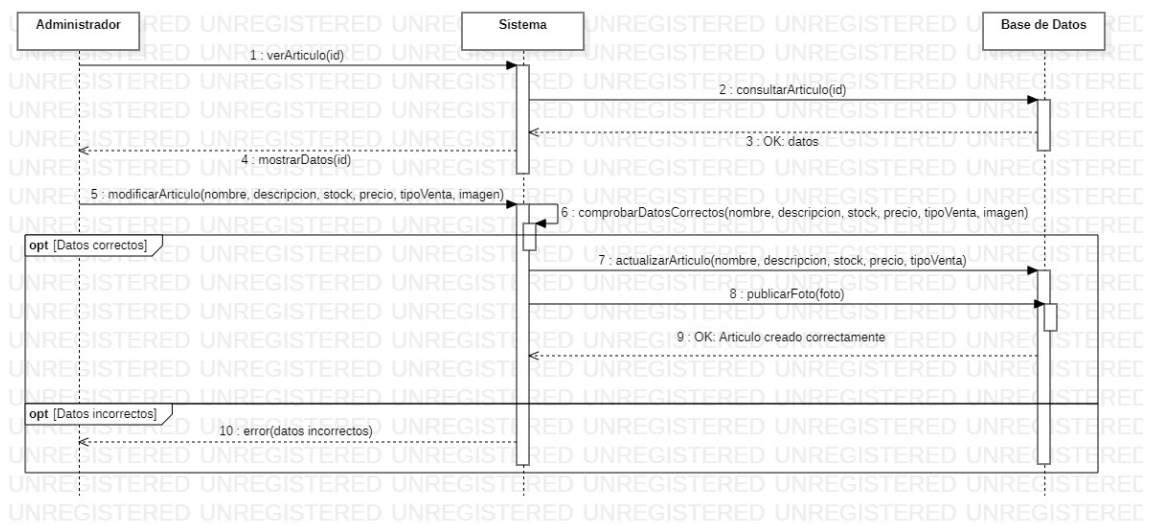


Figura A.5: diagrama de secuencia: modificar artículo

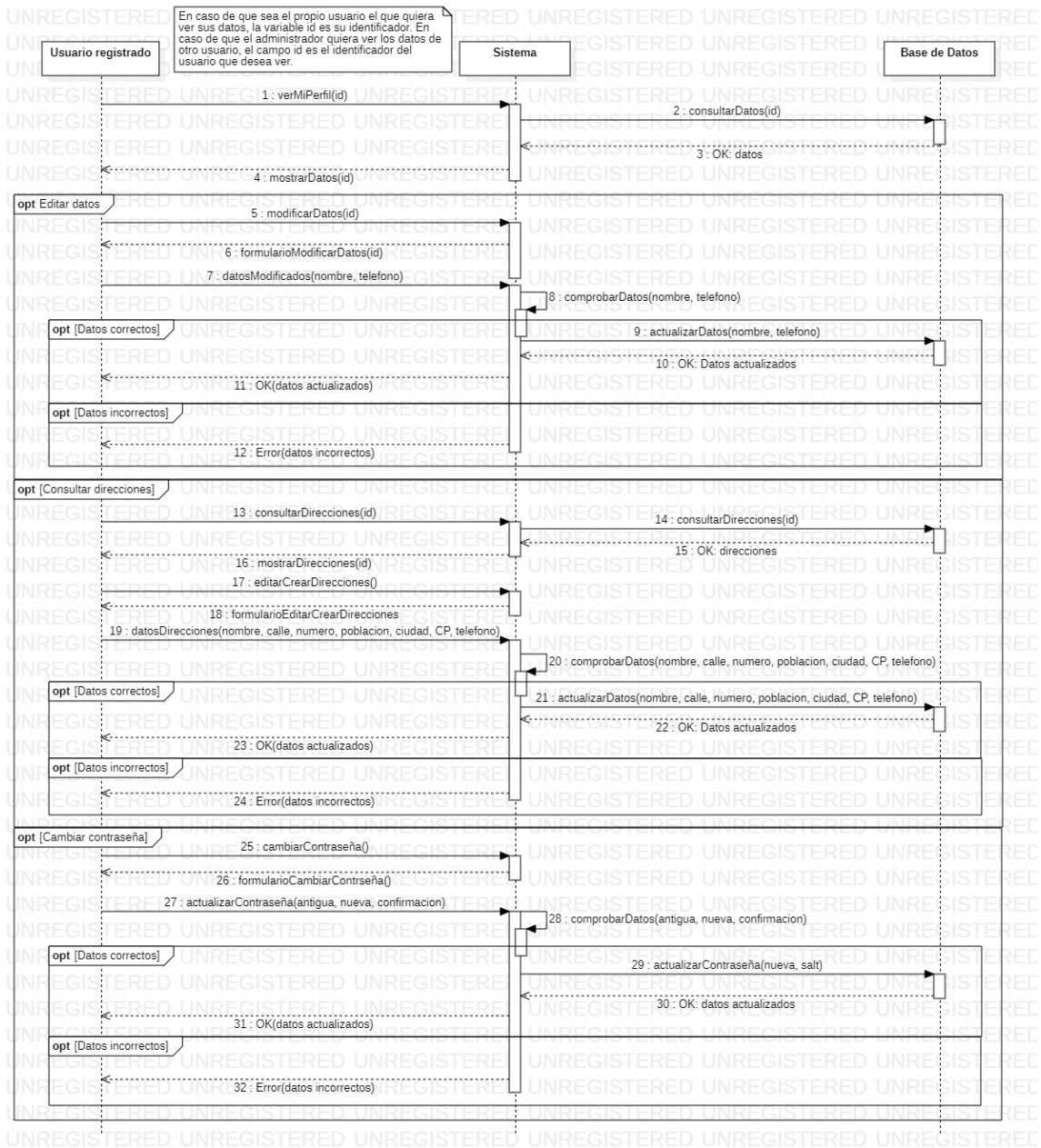


Figura A.6: diagrama de secuencia: modificar usuario

A.2. Diagramas de secuencia

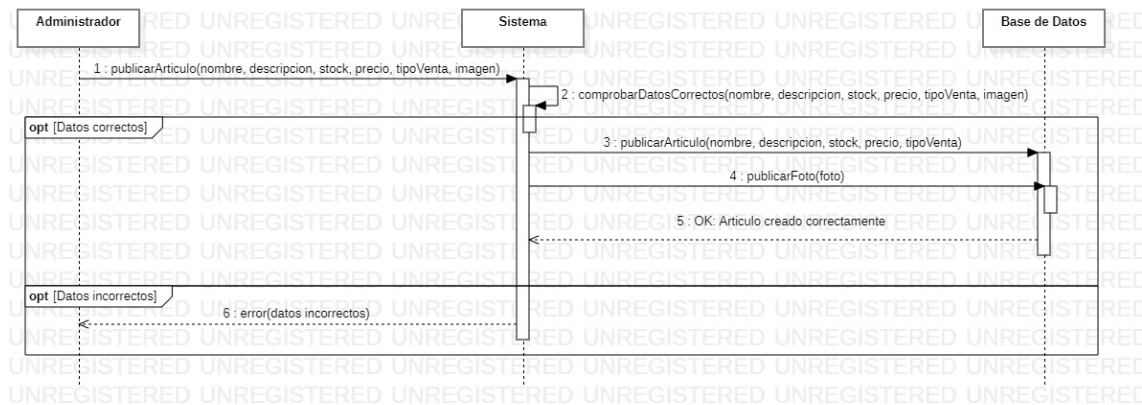


Figura A.7: diagrama de secuencia: publicar artículo

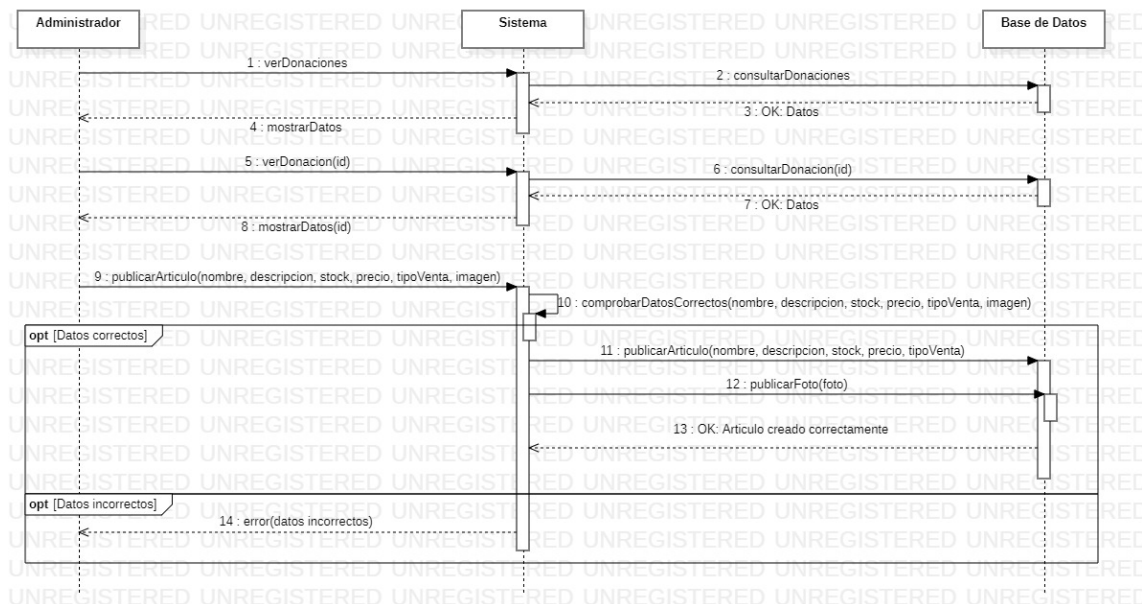


Figura A.8: diagrama de secuencia: publicar donación

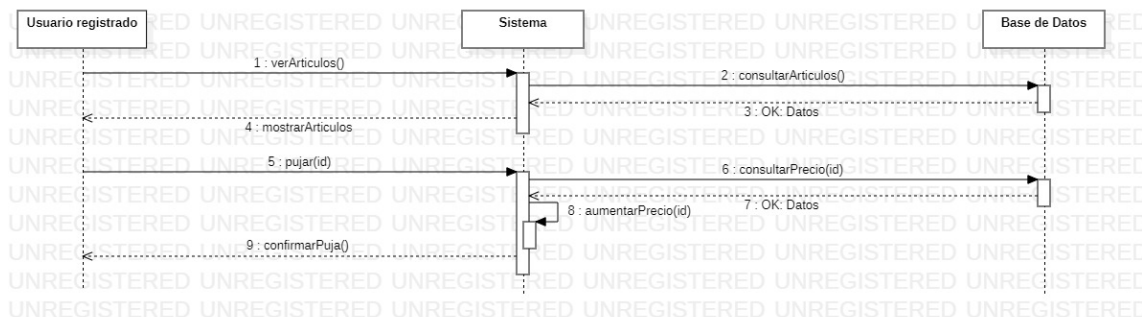


Figura A.9: diagrama de secuencia: pujar artículo

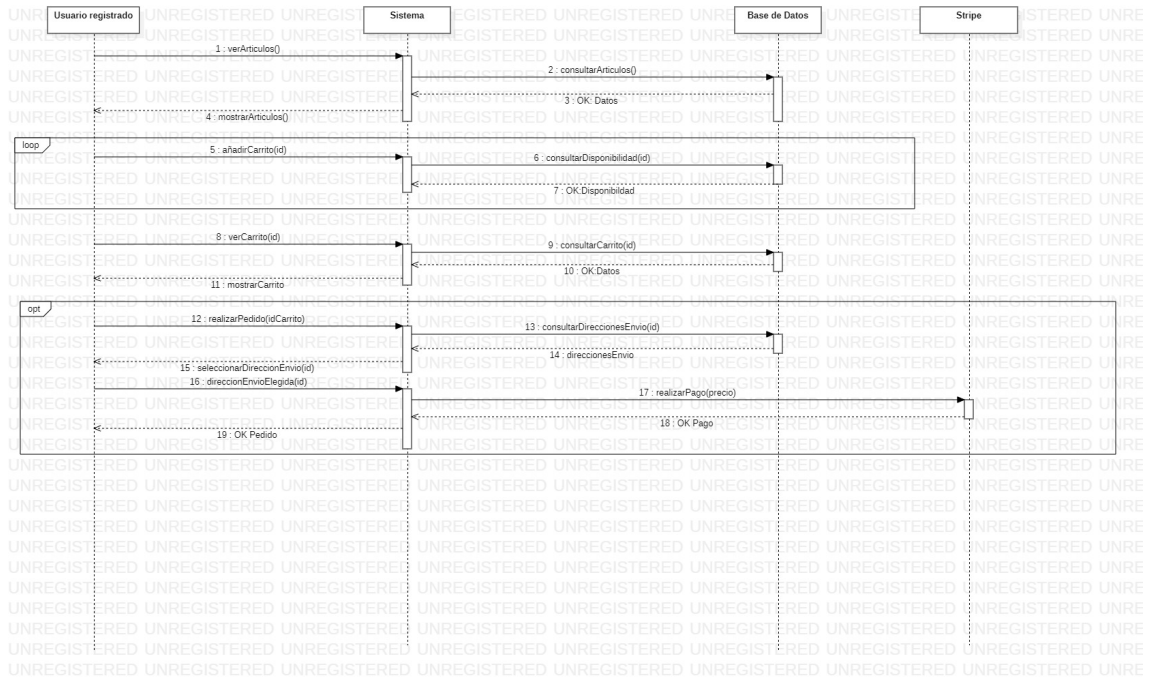


Figura A.10: diagrama de secuencia: realizar pedido

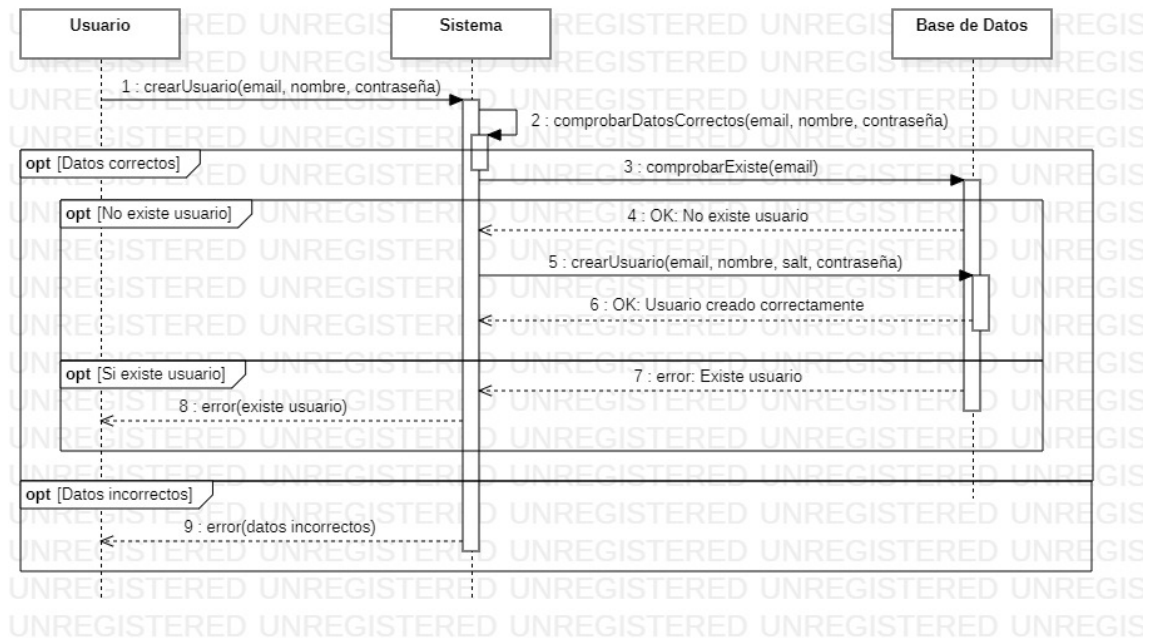


Figura A.11: diagrama de secuencia: registrarse

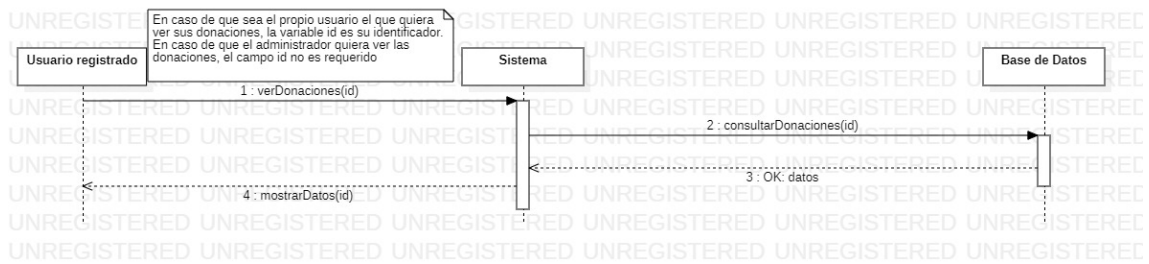


Figura A.12: diagrama de secuencia: ver donaciones

A.3. Pruebas caja negra

PCN-04 Modificar perfil	
Objetivo	Un usuario podrá modificar todos sus datos excepto el correo electrónico. El usuario administrador también podrá modificar los datos de otros usuarios.
Precondiciones	El usuario debe estar loggeado (por ejemplo, email: eva@gmail.com contraseña:1234)
Paso	Resultado esperado
Acceder a “Mi perfil“, “Editar“ y modificar todos los datos posibles.	Los datos se actualizan correctamente
Iniciar sesión como usuario administrador (email: admin@gmail.com contraseña:1234) , acceder al apartado “Usuarios“, seleccionar un cliente y modificar sus datos	Los datos se actualizan correctamente
Resultado final	OK

Tabla A.3.1: PCN-04 Modificar perfil

PCN-05 Publicar artículos	
Objetivo	El usuario administrador podrá publicar artículos
Precondiciones	El usuario debe estar loggeado como administrador (email: admin@gmail.com contraseña: 1234)
Paso	Resultado esperado
Acceder a “Publicar artículo“ y rellenar el formulario dejando algún campo vacío	El sistema solicita rellenar todos los datos
Acceder a “Publicar artículo“ y rellenar todos los campos del formulario, incluida la imagen	Un nuevo artículo es creado y se puede ver en la página principal
Resultado final	OK

Tabla A.3.2: PCN-05 Publicar artículos

PCN-06 Modificar o eliminar artículos	
Objetivo	El usuario administrador podrá modificar y eliminar artículos ya publicados
Precondiciones	El usuario debe estar loggeado como administrador (email: admin@gmail.com contraseña: 1234) Tiene que haber al menos un artículo publicado.
Paso	Resultado esperado
Desde la pantalla principal, seleccionar el botón “Modificar“ de un artículo y modificar los campos	Los campos s actualizan correctamente
Desde la pantalla principal, seleccionar la opción “Eliminar“ de un artículo y confirmar la operación	El artículo se elimina del sistema y no aparece en la página de inicio
Resultado final	OK

Tabla A.3.3: PCN-06 Modificar o eliminar artículos

PCN-07 Donar artículos	
Objetivo	Un usuario loggeado podrá donar artículos
Precondiciones	El usuario debe estar loggeado (por ejemplo, email: eva@gmail.com contraseña: 1234)
Paso	Resultado esperado
El usuario selecciona la opción “Donaciones“ y deja algunos campos del formulario sin rellenar	El sistema pide que se rellenen los campos que faltan
El usuario rellena todos los campos del formulario	El sistema crea una solicitud de donación, que podrá ser vista por el usuario en la sección “Mi perfil“ ->“Mis donaciones“
Resultado final	OK

Tabla A.3.4: PCN-07 Donar artículos

PCN-08 Publicar donaciones	
Objetivo	EL usuario administrador podrá publicar donaciones
Precondiciones	El usuario debe estar loggeado (email: admin@gmail.com contraseña: 1234)
Paso	Resultado esperado
El usuario selecciona la opción “Donaciones“	El sistema muestra un listado con todas las donaciones realizadas
El usuario selecciona la opción “Publicar donación“ que se encuentra al lado de las donaciones no publicadas	El sistema muestra un formulario con todos los campos rellenos que el administrador puede modificar
El usuario modifica algún campo y publica el artículo	Un nuevo artículo con los campos del formulario es publicado
Resultado final	OK

Tabla A.3.5: PCN-08 Publicar donaciones

PCN-09 Comprar artículo	
Objetivo	Un usuario registrado podrá comprar artículos
Precondiciones	El usuario debe estar loggeado (ejemplo email: eva@gmail.com contraseña: 1234)
Paso	Resultado esperado
El usuario añade un artículo en venta directa al carrito y selecciona la opción "Carrito"	El artículo elegido es añadido al carrito
El usuario añade un artículo en subasta inversa al carrito y selecciona la opción "Carrito"	Ambos artículos aparecen en el carrito
El usuario modifica el carrito	Los cambios se ven reflejados
El usuario selecciona "Continuar con la compra"	Se muestra una nueva página para elegir la dirección de envío
El usuario selecciona una dirección de envío y da a "Continuar"	Se muestra una nueva página para elegir el método de pago
El usuario selecciona un método de pago y da a "Finalizar compra"	El proceso de compra finaliza y se puede ver un nuevo pedido en la sección "Mi perfil" -> "Mis pedidos"
Inicia sesión como administrador (email: admin@gmail.com contraseña:1234) y accede a "Pedidos"	El pedido realizado anteriormente aparece en esta sección
Resultado final	OK

Tabla A.3.6: PCN-09 Comprar artículo

PCN-10 Pujar artículo	
Objetivo	Un usuario registrado podrá comprar artículos
Precondiciones	El usuario debe estar loggeado (ejemplo email: eva@gmail.com contraseña: 1234)
Paso	Resultado esperado
El usuario selecciona la opción “Pujar“ de un artículo en subasta directa	Aparece un mensaje de confirmación
El usuario deniega el mensaje de confirmación	El precio de la subasta no ha sido modificado
El usuario selecciona la opción “Pujar“ de un artículo en subasta directa y acepta el mensaje de confirmación	El precio del artículo se incrementa
Inicia sesión como administrador (email: admin@gmail.com contraseña:1234) y selecciona el artículo anterior. Selecciona la opción “Ver pujadores“	Solo una entrada con el email eva@gmail.com aparece
Resultado final	OK

Tabla A.3.7: PCN-10 Pujar artículo

Bibliografía

- [1] Brian Baxley. *Making the Web Work: Designing Effective Web Applications*. Morgan Kaufmann, 2007.
- [2] Ethan Brown. *Web Development with Node and Express: Leveraging the JavaScript Stack*. O'Reilly Media, 2014.
- [3] Sam Hughes. *Node.js & the Right Way: Practical, Server-Side JavaScript That Scales*. Pragmatic Bookshelf, 2019.
- [4] Steve McConnell. *Software Estimation: Demystifying the Black Art*. Microsoft Press, 2006.
- [5] Peter Morville y Jeff Callender. *Ambient Findability: What We Find Changes Who We Become*. O'Reilly Media, 2007.
- [6] Ian Sommerville. *Software Engineering*. Addison-Wesley, 2015.
- [7] Karl Wieggers y Joy Beatty. *Software Requirements*. Microsoft Press, 2013.
- [8] Frank Zammetti. *Mastering the Sequelize ORM: Build full-stack applications with Node.js, Express.js, and PostgreSQL using Sequelize*. Packt Publishing, 2016.