

DESARROLLO DE UNA APLICACIÓN WEB DE EVENTOS TURÍSTICOS



TRABAJO FIN DE GRADO
CURSO 2023-2024

ÁLVARO LAGUNA ALONSO
SARA CANO GARCÍA

DIRECTOR
ANTONIO SARASA CABEZUELO

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

DEVELOPMENT OF A WEB APPLICATION OF TOURIST EVENTS

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

ÁLVARO LAGUNA ALONSO
SARA CANO GARCÍA

DIRECTOR
ANTONIO SARASA CABEZUELO

CONVOCATORIA: JUNIO 2024

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

JUNIO DE 2024

DEDICATORIA

"Y ojalá parar la vida un rato." - %, Hens

A nosotros.

AGRADECIMIENTOS

A mi familia. A mis amigos: al de siempre, a los que he conocido en el proceso, a los que siguen conmigo y a los que no.

A todos los que han ayudado a que esta etapa sea más llevadera. A mis padres, hermano, amigos y sobre todo a Álvaro por acompañarme.

RESUMEN

Desarrollo de una aplicación web de eventos turísticos

El objetivo de este trabajo es el desarrollo de una página web donde se pueda promocionar alojamientos y a su vez se puedan realizar reservas. En concreto en la página se puede dar de alta tanto como cliente como arrendatario.

El cliente podrá realizar búsquedas en la página para encontrar el alojamiento que más se acomode a sus necesidades y reservarlo para un determinado periodo. Además podrá realizar comentarios y puntuar aquellos alojamientos en los que se ha alojado.

Por otra parte, el arrendatario puede hacer uso de la web para promocionar sus alojamientos de una forma rápida y fácil pudiendo modificar los datos de sus alojamientos o eliminarlos cuando este considere.

Esta aplicación está formada por una página web y una base de datos cuya información irá aumentando cada vez que se dé de alta un nuevo usuario y se vayan publicando nuevos alojamientos.

Palabras clave

Aplicación web, Base de datos, Experiencia de usuario (UX), Alojamiento, Cliente, Vendedor, Reservas

ABSTRACT

Development of a web application for tourist events

The purpose of this work is the development of a web page where accommodations can be promoted and, at the same time, bookings can be made. Specifically, in this page you can register both as a customer and as a renter.

The client will be able to make searches on the page to find the accommodation that best suits their needs and book it for a certain period. They can also make comments and rate the accommodations they have stayed in.

On the other hand, the tenant can make use of the web to promote his accommodations in a quick and easy way, being able to modify the data of his accommodations or delete them whenever he wishes.

This application consists of a web page and a database whose information will increase every time a new user registers and new accommodations are published.

Keywords

Web application, Database, User experience, Accommodations, Clients, Seller, Booking.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción	1
1.1 Estructura de la memoria	1
1.2 Motivación	2
1.3 Objetivos	2
1.4 Plan de trabajo	3
Chapter 1 – Introduction	7
1.1 Structure of the report	7
1.2 Motivation	8
1.3 Goals	8
1.4 Work plan	9
Capítulo 2 - Estado de la cuestión	13
2.1 Booking	13
2.2 Trivago	13
2.3 Airbnb	14
Capítulo 3 - Tecnología usada	15
3.1 Xampp	15
3.2 Visual Studio Code	15
3.3 Github	15
3.4 Xdebug	16
3.5 Bootstrap y Bootstrap Studio	16
3.6 PHP	16
3.7 HTML	16
3.8 CSS	17
3.9 JS y jQuery	17
3.10 Leaflet, OpenStreetMap y Nominatim	17
3.11 Suite de Google	18
Capítulo 4 - Casos de uso	19
4.1 Actores	19
4.2 Diagramas de caso de uso	19
4.2.1 Diagrama de cliente	19

4.2.2 Diagrama de vendedor	20
4.2.3 Diagrama de usuario no registrado	21
4.3 Especificación de requisitos	21
4.3.1 Módulo cuentas de usuario	21
4.3.2 Módulo alquiler	27
4.3.3 Módulo publicar	29
4.3.4 Módulo comentarios	39
4.3.5 Módulo búsqueda	40
Capítulo 5 - Arquitectura de la aplicación y modelo de Datos	42
5.1 Arquitectura de la aplicación	42
5.2 Modelo de Datos	42
5.2.1 Tabla users	44
5.2.2 Tabla sellers	45
5.2.3 Tabla customers	45
5.2.4 Tabla articulos	46
5.2.5 Tabla alojamientos	46
5.2.6 Tabla habitaciones	46
5.2.7 Tabla camas	48
5.2.8 Tabla alquileres	48
5.2.9 Tabla precios	49
5.2.10 Tabla ofertas	49
5.2.11 Tabla características	50
5.2.12 Tabla característicasarticulo	50
5.2.13 Tabla comentarios	51
Capítulo 6 - Implementación y diseño de la aplicación	52
6.1 Diseño de la aplicación	52
6.2 Implementación de la aplicación web	53
6.2.1 Clases	53
6.2.2 Módulo cuentas de usuario	55
6.2.3 Módulo alquiler	69
6.2.4 Módulo Publicar	74
6.2.5 Módulo Comentarios	95
6.2.6 Módulo Búsqueda	97

Capítulo 7 - Conclusiones y trabajo futuro	102
7.1 Conclusiones	102
7.2 Trabajo futuro	103
Capítulo 8 - Conclusions and future work	104
8.1 Conclusions	104
8.2 Future work	105
Capítulo 9 - Contribuciones Personales	106
9.1 Sara Cano García	106
9.2 Álvaro Laguna Alonso	108
Bibliografía	111
Apéndice A - Guía de uso	113
A.1 Vista de inicio	113
A.2 Vista de registro	115
A.3 Vista de iniciar sesión	116
A.4 Vista de perfil	117
A.5 Vista de editar perfil	119
A.6 Vista de alquilar	120
A.7 Vista de publicar artículo	120
A.8 Vista de editar artículo	123
A.9 Vista de modificar precios artículo	124
A.10 Vista de publicar oferta	125
A.11 Vista de modificar oferta	125
A.12 Vista de ver reservas del artículo	126
A.13 Vista de búsqueda	127
A.14 Vista de artículo	128
A.15 Vista de comentarios	131
Apéndice B - Repositorio de código fuente	137

ÍNDICE DE FIGURAS

Figura 4.1: Diagrama de cliente.....	20
Figura 4.2: Diagrama de vendedor.....	20
Figura 4.3: Diagrama de usuario no registrado.....	21
Figura 5.0: Arquitectura de la aplicación.....	45
Figura 5.1: Estructura de la base de datos.....	43
Figura 5.2: Modelo relacional de la base de datos.....	44
Figura 5.3: Tabla users.....	45
Figura 5.4: Tabla sellers.....	45
Figura 5.5: Tabla customers.....	45
Figura 5.6: Tabla articulos.....	46
Figura 5.7: Tabla alojamientos.....	46
Figura 5.8: Tabla habitaciones.....	47
Figura 5.9: Tabla camas.....	47
Figura 5.10: Tabla alquileres.....	48
Figura 5.11: Tabla precios.....	48
Figura 5.12: Tabla ofertas.....	49
Figura 5.13: Tabla características.....	49
Figura 5.14: Tabla característicasarticulos.....	49
Figura 5.15: Tabla comentarios.....	50
Figura 6.1 Estructura de clases.....	53
Figura 6.2 Registro como cliente.....	55
Figura 6.3 Registro como vendedor.....	56
Figura 6.4 Código registrar user.....	57
Figura 6.5 Código crear user customer.....	58
Figura 6.6 Código crear user seller.....	58
Figura 6.7 Formulario login.....	59
Figura 6.8 Código formulario login.....	59
Figura 6.9 Código login.....	60

Figura 6.10 Código logout.....	61
Figura 6.11 Perfil usuario cliente.....	61
Figura 6.12 Código formulario perfil cliente.....	62
Figura 6.13 Variables código formulario perfil cliente.....	63
Figura 6.14 Perfil usuario comerciante.....	63
Figura 6.15 Código formulario perfil comerciante.....	64
Figura 6.16 Variables código formulario perfil comerciante.....	65
Figura 6.17 Código editar perfil.....	66
Figura 6.18 Código editar perfil seller y customer.....	67
Figura 6.19 Código editar user.....	68
Figura 6.20 Código eliminar user.....	69
Figura 6.21 Vista alquilar.....	70
Figura 6.22 Código formulario alquiler.....	71
Figura 6.23 Código comprobar disponibilidad.....	72
Figura 6.24 Código comprobar disponibilidad 2.....	73
Figura 6.25 Código alquilar habitación.....	74
Figura 6.26 Vista publicar artículo.....	74
Figura 6.27 Código estado formulario publicar.....	75
Figura 6.28 Código publicar artículo.....	75
Figura 6.29 Vista publicar alta habitaciones.....	76
Figura 6.30 Código alta habitaciones.....	77
Figura 6.31 Vista formulario precios.....	78
Figura 6.32 Código formulario precios.....	78
Figura 6.33 Código llamada a eliminar artículo.....	79
Figura 6.34 Código eliminar artículo.....	80
Figura 6.35 Vista editar artículo.....	81
Figura 6.36 Código formulario editar artículo.....	82
Figura 6.37 Código editar artículo.....	83
Figura 6.38 Código editar habitaciones artículo.....	83
Figura 6.39 Código formulario editar habitación.....	84
Figura 6.40 Código editar habitación.....	84
Figura 6.41 Formulario modificar camas.....	85
Figura 6.42 Código formulario editar camas.....	85

Figura 6.43 Código editar cama.....	86
Figura 6.44 Código formulario precios.....	87
Figura 6.45 Código publicar precios.....	87
Figura 6.46 Vista publicar oferta.....	88
Figura 6.47 Código publicar ofertas.....	89
Figura 6.48 Vista modificar oferta.....	90
Figura 6.49 Código modificar oferta.....	90
Figura 6.50 Vista eliminar oferta.....	91
Figura 6.51 Código eliminar oferta.....	92
Figura 6.52 Vista ver reservas.....	93
Figura 6.53 Código ver reservas.....	93
Figura 6.54 Código ver reservas 2.....	94
Figura 6.55 Código getReservas().....	94
Figura 6.56 Vista publicar comentario.....	95
Figura 6.57 Código comprobar comentarios.....	95
Figura 6.58 Vista comentario.....	96
Figura 6.59 Código ver comentario.....	97
Figura 6.60 Barra de búsqueda.....	98
Figura 6.61 Búsqueda.....	98
Figura 6.62 Código de búsqueda.....	98
Figura 6.63 Vista paginación búsqueda.....	99
Figura 6.64 Código paginación.....	99
Figura 6.65 Filtros búsqueda.....	100
Figura 6.66 Código filtrar por características.....	101
Figura A.1 Vista inicio.....	113
Figura A.2 Vista inicio usuario registrado como comerciante.....	114
Figura A.3 Vista inicio usuario registrado como cliente.....	114
Figura A.4 Vista registro.....	115
Figura A.5 Vista registro como cliente.....	115
Figura A.6 Vista registro como vendedor.....	116
Figura A.7 Vista iniciar sesión.....	116
Figura A.8 Vista perfil cliente.....	117

Figura A.9 Vista perfil vendedor.....	118
Figura A.10 Vista editar perfil usuario vendedor.....	119
Figura A.11 Vista editar perfil usuario cliente.....	119
Figura A.12 Vista alquilar.....	120
Figura A.13 Vista publicar artículo.....	121
Figura A.14 Vista publicar artículo alta habitaciones.....	121
Figura A.15 Vista publicar artículo precio único.....	122
Figura A.16 Vista publicar artículo precio por día.....	122
Figura A.17 Vista editar artículo.....	123
Figura A.18 Vista editar precio único.....	124
Figura A.19 Vista editar precio por día.....	124
Figura A.20 Vista publicar oferta.....	125
Figura A.21 Vista modificar oferta.....	126
Figura A.22 Vista ver reservas del artículo.....	126
Figura A.23 Vista filtros búsqueda.....	127
Figura A.24 Vista búsqueda.....	128
Figura A.25 Vista artículo.....	130
Figura A.26 Vista comentario.....	131
Figura A.27 Vista comentarios publicados.....	131

ÍNDICE DE TABLAS

Tabla 1.1 Etapas y duración en el desarrollo del proyecto.....	4
Table 1.1 Stages and duration of project development.....	10
Tabla 4.1: Especificación registro de usuario.....	23
Tabla 4.2: Especificación Login.....	24
Tabla 4.3: Especificación Log out.....	24
Tabla 4.4: Especificación Eliminar cuenta de usuario.....	25
Tabla 4.5: Especificación modificar datos de usuario.....	27
Tabla 4.6: Especificación alquilar un alojamiento.....	28
Tabla 4.7: Especificación mostrar historial de reservas.....	29
Tabla 4.8: Especificación publicar un artículo.....	30
Tabla 4.9: Especificación eliminar un artículo.....	31
Tabla 4.10: Especificación modificar datos de un artículo.....	33
Tabla 4.11: Especificación modificar precios de un artículo.....	34
Tabla 4.12: Especificación publicar oferta.....	35
Tabla 4.13: Especificación eliminar ofertas.....	36
Tabla 4.14: Especificación modificar ofertas.....	37
Tabla 4.15: Especificación mostrar historial de artículos publicados.....	38
Tabla 4.16: Especificación ver reservas del artículo.....	39
Tabla 4.17: Especificación Comentar un alojamiento.....	40
Tabla 4.18: Especificación Buscar alojamientos.....	41

Capítulo 1 - Introducción

Este trabajo de Fin de Grado consiste en el desarrollo de una página web en el que los usuarios registrados como clientes podrán realizar reservas vacacionales y los usuarios registrados como arrendatarios podrán promocionar sus alojamientos.

1.1 Estructura de la memoria

Para describir el proyecto y facilitar su comprensión, la memoria se ha dividido en varios apartados. En primer lugar aparece una dedicatoria, agradecimientos y un pequeño resumen (tanto en inglés como en español) sobre la página a desarrollar. Después de estos apartados aparecen los índices de contenidos, de figuras y de tablas para poder moverse por el documento fácilmente.

A continuación aparecen varios capítulos. En el capítulo 1 se realiza una pequeña introducción donde se especifica la motivación, objetivos y plan de trabajo empleado. En el capítulo 2 se citan varias aplicaciones similares a la implementada que han servido de inspiración. Todo lo relacionado con la tecnología y aplicaciones usadas se especifica en el capítulo 3 y en el capítulo 4 se citan los casos de uso. En el capítulo 5 se explica la arquitectura de la aplicación y de la base de datos. A continuación se detalla la implementación y diseño de la aplicación en el capítulo 6 y por último, en el capítulo 7 se comenta el trabajo futuro que se podría realizar sobre la aplicación para añadir más funcionalidad y las conclusiones.

Tras estos capítulos se especifica el trabajo realizado de cada uno de los miembros del equipo, la bibliografía y los apéndices donde aparece una guía de uso de la aplicación.

1.2 Motivación

Se ha implementado una página web donde se pueden realizar reservas de alojamientos turísticos para el periodo que el cliente considere. Además se puede dar de alta en la página como arrendatario para promocionar alojamientos y que los clientes puedan realizar dichas reservas.

Cuando se acercan los periodos vacacionales muchos optan por viajar y darse el lujo de descubrir nuevos sitios y culturas ya sea con amigos, familia o por su cuenta. Para ello es importante disponer de un buen alojamiento para hacer esta experiencia más amena y cómoda. Esta ha sido la principal motivación para diseñar esta página web. Queríamos programar una página exclusivamente para el alquiler de alojamientos vacacionales que fuera sencilla de usar.

Por otra parte muchos acuden a inmobiliarias para promocionar sus alojamientos y que alguien les acompañe en este proceso que a veces puede ser un poco tedioso. Para simplificar este proceso se le da la opción a aquellos que lo deseen de promocionar de una forma fácil y rápida los alojamientos turísticos que quiera desde casa.

1.3 Objetivos

El principal objetivo de este TFG es ofrecer una experiencia única y fácil de alquilar o promocionar alojamientos turísticos. En concreto los objetivos a alcanzar son:

- Alcance global: Una página web permite llegar a un público global, aumentando significativamente la visibilidad de los alojamientos y atrayendo a potenciales clientes de diferentes partes del mundo. Además la página web estará disponible en todo momento, lo que permite a los usuarios buscar y

reservar alojamientos en cualquier momento del día, lo que aumenta las oportunidades de negocio.

- Mayor control y flexibilidad: Tener una plataforma propia brinda la libertad de controlar la presentación de los alojamientos, la disponibilidad y los precios en tiempo real, adaptándose según la demanda y las temporadas.
- Diseño de la página web: Es importante ofrecer un diseño atractivo, claro y sencillo de usar para no generar dudas en los usuarios. Por ello se ha prestado especial atención en el diseño de la web para que todas las opciones que se ofrecen estén claras.
- Interacción directa con los clientes: A través de la página web, se puede establecer una comunicación directa con los clientes, ya que estos pueden comentar su experiencia con el alojamiento reservado pudiendo así mejorar los servicios que el arrendatario ofrece y generar una experiencia más satisfactoria.

1.4 Plan de trabajo

Aquí se describe el plan de trabajo a seguir para la consecución de los objetivos descritos en el apartado anterior.

A continuación se muestra una tabla donde se especifican todas las etapas acordadas al inicio del curso para llevar a cabo el TFG y una estimación de su duración.

Etapa	Duración
Especificación de requisitos	15 Septiembre - 2 Octubre
Cuentas de usuario	2 Octubre - 15 Noviembre
Publicación/Modificación de artículos	15 Noviembre - 15 Diciembre
Búsqueda	15 Diciembre - 1 Febrero

Alquiler de alojamientos	1 Febrero - 15 Marzo
Comentarios	15 Marzo - 20 Abril
Desarrollo de la memoria	20 Abril - 15 Mayo

Tabla 1.1 Etapas y duración en el desarrollo del proyecto

Etapa 1: Especificación de requisitos

En esta etapa se discutió y definió los casos de usos que se iban a implementar en la página web. Se realizó una reunión con el tutor del TFG para establecer los requisitos.

Etapa 2: Cuentas de usuario

El primer hito consistió en la implementación del código para gestionar las cuentas de usuario tanto de clientes como de vendedores. Las funcionalidades que se implementaron fueron:

- Registro: Si el usuario no tiene cuenta en la web puede indicar si quiere crearse una cuenta como cliente o como arrendatario.
- Login: El usuario tiene que tener creada una cuenta para hacer login, para ello se necesita introducir el correo y la contraseña del usuario.
 - Usuario cliente: Si el usuario no tiene cuenta en la web como cliente tiene que registrarse. Para crear la cuenta se le pide sus datos personales.
 - Usuario arrendatario: Si el usuario no tiene cuenta en la web como arrendatario, se le pide sus datos personales para crear la cuenta.
- Logout: El usuario debe haber hecho login para poder cerrar sesión.
- Baja usuario: El usuario podrá eliminar su cuenta. Para llevar a cabo esta acción, se necesita tener una cuenta creada y haber hecho login.

- Modificar datos: Para ello hay que tener una cuenta creada y haber iniciado sesión. El usuario puede modificar los datos personales introducidos durante el registro en la página del perfil.

Etapa 3: Publicación/Modificación de artículos

Esta etapa se centra en las acciones que puede realizar un usuario registrado como arrendatario, las cuales son:

- Publicar un artículo: Un usuario registrado como arrendatario que haya iniciado sesión puede ofrecer un producto para su alquiler.
- Modificar un artículo: Un arrendatario que ha publicado un artículo, puede modificar la información de dicho artículo.
- Eliminar un artículo: Un usuario arrendatario que haya iniciado sesión, puede eliminar un artículo que haya publicado.

Etapa 4: Búsqueda

Cualquier usuario puede buscar los alojamientos que desee en la web (no es necesario estar registrado). Se pueden buscar palabras claves para encontrar el alojamiento que más se ajuste a las necesidades del cliente. Además se han añadido filtros para que la búsqueda sea más rápida y encontrar el alojamiento que mejor se adecue a las necesidades del cliente. Se puede filtrar tanto por precios, puntuación, tipo de alojamiento como por características particulares, por ejemplo si el alojamiento cuenta con WiFi.

Etapa 5: Alquiler de alojamientos

En esta etapa se implementó la funcionalidad de alquilar un alojamiento además del historial:

- Alquilar: Un usuario registrado como cliente, habiendo iniciado sesión, puede alquilar el alojamiento que desee. Una vez alquilado, este aparecerá en su historial de alquileres.
- Historial: Se va a dividir en tres tipos de historial dependiendo del tipo de usuario (comprador o comerciante)
 - Historial de compras: El usuario cliente en su perfil puede ver los alojamientos que ha alquilado. Para poder ver el historial tiene que haber iniciado sesión.
 - Historial de alojamientos alquilados: El usuario registrado como arrendatario y habiendo iniciado sesión, en su perfil puede ver un historial de sus artículos que han sido alquilados.
 - Historial de alojamientos: El usuario registrado como arrendatario habiendo iniciado sesión, en su perfil puede ver todos los artículos que ha publicado.

Etapas 6: Comentarios

La última funcionalidad implementada son los comentarios. Un cliente podrá comentar en aquellos alojamientos en los que haya estado además de dar una puntuación. De esta manera el propietario podrá conocer la opinión de sus clientes y podrá mejorar su alojamiento para hacer que la estancia sea más satisfactoria.

Etapas 7: Desarrollo de la memoria

Por último se ha recopilado toda la información de la página web para facilitar su comprensión y conocer todos sus detalles.

Chapter 1 – Introduction

This Final Bachelor's Degree Project (TFG for its acronym in Spanish) consists in the development of a web page in which the users registered as clients can make vacation reservations and the users registered as renters can promote their lodgings.

1.1 Structure of the report

In order to describe the project and make its understanding easier, the report has been divided into several sections. First of all, there is a dedication, acknowledgements and a short summary (both in English and Spanish version) about the site to be developed. These sections are followed by indexes, figures and tables of contents to make it easier to move through the document.

After that, there are several chapters. Chapter 1 contains a short introduction where the motivation, goals and work plan used are specified. Chapter 2 cites several applications similar to the one implemented that have served as inspiration. Everything related to the technology and applications used are specified in Chapter 3, and in Chapter 4 the use cases are mentioned. Chapter 5 explains the architecture of the application and the database. Then, the implementation and design of the application is explained in Chapter 6, and finally, in Chapter 7, the potential work that could be done on the application in the future in order to add more functionality, and conclusions are discussed.

After these chapters, the work done by each of the team members, the bibliography and the appendices, which include a guide for the use of the application.

1.2 Motivation

We have implemented a web page where it is possible to make reservations of tourist accommodations for the period that the client considers. In addition, it is possible to register on the page as a lessee to promote accommodations so that customers can make these reservations.

When holiday periods are approaching, many people choose to travel and indulge in discovering new places and cultures either with friends, family or by themselves. For this reason, it is important to get a good hosting to make this experience more enjoyable and comfortable. This has been the main motivation to design this website. We wanted to design a site exclusively for vacation rentals that was easy to use.

On the other hand, many people go to real estate agencies to promote their accommodations and have someone to accompany them in this process that can sometimes be a bit tedious. To simplify this process, we give the option to those who wish to promote in an easy and fast way the tourist accommodations they want from home.

1.3 Goals

The main objective of this TFG is to offer a unique and easy experience to rent or promote tourist accommodations. Specifically the objectives to be achieved are:

- Global reach: A website enables reaching a global audience, significantly increasing the visibility of the accommodations and attracting potential customers from different parts of the world. In addition, the website will be available around the clock, allowing users to search and book accommodations at any time of the day, which increases business opportunities.

- Greater control and flexibility: Having your own platform provides the freedom to control the presentation of accommodations, availability and prices in real time, adapting them according to demand and seasons.
- Web page design: It is important to offer an attractive, clear and user-friendly design so as not to generate doubts in users. Therefore, special attention has been paid to the design of the website so that all the options offered are clear.
- Direct interaction with customers: By means of the website, direct communication can be established with customers, as they can comment on their experience with the accommodation booked, thus being able to improve the services offered by the tenant and generate a more satisfactory experience.

1.4 Work plan

This section describes the work plan to be followed in order to achieve the objectives described in the previous section.

Next, it is shown a chart specifying all the stages agreed at the beginning of the course to carry out the TFG and an estimate of its duration.

Stage	Duration
Specification of requirements	September 15 - October 2
User accounts	October 2 - November 15
Publication/Modification of articles	15 November - 15 December
Search	December 15 - February 1
Rental accommodations	February 1 - March 15
Comments	March 15 - April 20
Memory development	April 20 - May 15

Table 1.1 Stages and duration of project development

Stage 1: Requirements specification

In this stage, the use cases to be implemented in the web page were discussed and defined. A meeting was held with the TFG tutor to establish the requirements.

Stage 2: User accounts

The first milestone consisted in implementing the code to manage user accounts for both clients and tenants. The functionalities implemented were:

- Registration: If the user does not have an account on the web, he/she can indicate if he/she wants to create an account as a client or as a tenant.
- Login: The user is required to have created an account to login, for this it is necessary to enter the user's email and password.
 - Client user: If the user does not have an account on the web as a client, he/she has to register. To create the account you will be asked for your personal data.
 - Tenant user: If the user does not have an account on the web as a tenant, he/she will be asked for his/her personal data to create the account.
- Logout: The user must have logged in to be able to close the session.
- User cancellation: The user will be able to delete his account. To carry out this action, it is necessary to have an account created and to have logged in.
- Modify data: To do this, the user must have an account created and be logged in. The user can modify the personal data entered during registration on the profile page.

Stage 3: Publication/Modification of articles

This stage focuses on the actions that can be performed by a user registered as a tenant, which are:

- Publish: A user registered as a renter who is logged in, can offer a product for rent.
- Modify: A renter who has published an item, can modify the information on that item.
- Delete: A logged-in tenant user can delete an item he/she has published.

Stage 4: Search

Any user can search for the accommodations they want on the site (no registration is required). You can search for keywords to find the accommodation that best suits the client's needs.

Stage 5: Rental of accommodations

In this stage, the functionality of renting an accommodation was implemented in addition to the history:

- Rent: A user registered as a client, having logged in, can rent the desired accommodation. Once rented, it will appear in their rental history.
- History: It will be divided into three types of history depending on the type of user (buyer or merchant)
 - Purchase history: The client user can see the accommodations he/she has rented in his/her profile. To be able to see the history, the user must be logged in.
 - History of rented accommodations: The user registered as a renter and logged in, in his/her profile can see a history of the items that have been rented.

- Accommodation history: The user registered as a renter having logged in, in his/her profile can see all the published items.

Stage 6: Comments

The last functionality implemented are the comments. A client will be able to comment on those accommodations in which he/she has stayed and give a score. In this way, the owner will be able to know the opinion of his clients and will be able to enhance his/her accommodation to make the stay more satisfactory.

Stage 7: Development of the report

Finally, all the information of the web page has been compiled to facilitate its understanding and to know all its details.

Capítulo 2 - Estado de la cuestión

En este apartado se citan varias aplicaciones similares a la implementada y que han servido como inspiración.

2.1 Booking

Booking.com [\[1\]](#) es una plataforma líder a nivel mundial en reservas de alojamiento en línea. Ofrece una amplia gama de opciones, que van desde hoteles y apartamentos hasta villas y hostales, en destinos de todo el mundo. Los usuarios pueden buscar y comparar diferentes opciones de alojamiento según sus preferencias y necesidades, como ubicación, precio, comodidades y opiniones de otros usuarios. Además de alojamiento, Booking.com también proporciona información sobre actividades locales, transporte y servicios adicionales para ayudar a los viajeros a planificar y organizar su viaje de manera completa.

Permite que tanto marcas ya establecidas como emprendedores puedan promocionar sus alojamientos ya que facilita que alojamientos de cualquier lugar puedan llegar a una audiencia global e impulsar su negocio. Además permite conocer la opinión de sus clientes a través de los comentarios pudiendo así mejorar sus servicios.

2.2 Trivago

Trivago.es [\[2\]](#) es un motor de búsqueda de hoteles que permite a los usuarios comparar precios de alojamiento en diferentes sitios web de reserva. Los usuarios pueden buscar hoteles en una variedad de destinos, filtrar los resultados según sus preferencias y comparar precios de diferentes proveedores en una sola plataforma. Trivago muestra información detallada sobre cada hotel, incluyendo imágenes,

descripciones y reseñas de usuarios, para ayudar a los viajeros a tomar decisiones informadas. Además de hoteles, Trivago también ofrece opciones de búsqueda para otros tipos de alojamiento, como apartamentos y hostales. Su objetivo principal es facilitar a los usuarios la búsqueda y reserva del alojamiento perfecto para sus viajes.

2.3 Airbnb

Airbnb.es [\[3\]](#) es una plataforma en línea que facilita la reserva de alojamientos únicos y experiencias auténticas en todo el mundo. A través de Airbnb, los anfitriones pueden listar sus propiedades, que van desde apartamentos y casas hasta villas y cabañas, y los viajeros pueden buscar y reservar alojamientos según sus preferencias y presupuesto. La plataforma permite a los usuarios explorar una amplia variedad de opciones de alojamiento, desde espacios compartidos hasta propiedades enteras, y filtrar los resultados según criterios específicos como ubicación, precio y comodidades.

Capítulo 3 - Tecnología usada

A continuación se describe la tecnología que se ha utilizado para realizar el proyecto.

3.1 Xampp

XAMPP [\[4\]](#) es un paquete de software gratuito que facilita la creación de entornos de desarrollo web local. Incluye Apache, un servidor web de código abierto que procesa solicitudes HTTP, y MySQL, un sistema de gestión de bases de datos relacionales para almacenar y administrar datos en sitios web dinámicos. Es una solución todo en uno para desarrollar y probar aplicaciones web en entornos locales.

3.2 Visual Studio Code

Visual Studio Code [\[5\]](#) es un editor de código abierto desarrollado por Microsoft, popular entre los programadores. Ofrece una interfaz intuitiva, soporte para varios lenguajes (siendo este uno de los principales motivos por el que se decidió usar esta aplicación), extensiones personalizables, control de versiones integradas, así como integración con servicios en la nube.

3.3 Github

GitHub [\[6\]](#) es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Permite a los desarrolladores trabajar juntos en proyectos de software, gestionar el código fuente, realizar seguimiento de cambios y colaborar en equipo. Además, ofrece herramientas para revisión de código, seguimiento de problemas y despliegue automatizado. Se ha usado esta herramienta para compartir el código fuente.

3.4 Xdebug

Xdebug [\[7\]](#) es una herramienta de depuración de código para PHP que permite a los desarrolladores detectar y solucionar errores más fácilmente. Proporciona funcionalidades como seguimiento de ejecución, inspección de variables y perfiles de rendimiento en tiempo real. Para conocer el comportamiento del código y depurar más rápidamente se ha utilizado esta herramienta.

3.5 Bootstrap y Bootstrap Studio

Bootstrap [\[8\]](#) es un marco de trabajo de diseño web de código abierto que simplifica la creación de sitios interactivos y visualmente atractivos. Bootstrap Studio [\[9\]](#) es una herramienta de diseño visual que aprovecha las capacidades de Bootstrap, permitiendo a los usuarios crear interfaces web de manera intuitiva y rápida. Juntos, ofrecen una solución eficiente para el diseño y desarrollo de sitios web modernos.

3.6 PHP

PHP [\[10\]](#) es un lenguaje de programación interpretado del lado del servidor y de uso general que se adapta especialmente al desarrollo web. Permite la creación de sitios web dinámicos e interactivos al integrar código con HTML.

3.7 HTML

HTML (HyperText Markup Language) [\[11\]](#) es un lenguaje de marcado utilizado para crear y estructurar el contenido de páginas web. Define la estructura y el significado de los elementos dentro de un documento web mediante etiquetas. Estas etiquetas permiten la incorporación de texto, imágenes, enlaces y otros elementos multimedia, facilitando la creación de páginas web interactivas y accesibles.

3.8 CSS

CSS (Cascading Style Sheets) [\[12\]](#) es un lenguaje utilizado para definir el estilo y la presentación de documentos HTML. Permite controlar el diseño, el color, la tipografía y otros aspectos visuales de un sitio web. CSS se utiliza junto con HTML para crear páginas web visualmente atractivas y con un diseño coherente.

3.9 JS y jQuery

JavaScript (JS) [\[13\]](#) es un lenguaje de programación utilizado principalmente para agregar interactividad y dinamismo a las páginas web, permitiendo manipular elementos HTML, gestionar eventos y crear animaciones. jQuery [\[14\]](#) es una biblioteca de JavaScript que simplifica la escritura de código, proporcionando una serie de funciones predefinidas para realizar tareas comunes como selección de elementos y manejo de eventos, lo que agiliza el desarrollo web. Juntos, JavaScript y jQuery ofrecen herramientas poderosas para crear experiencias interactivas en la web.

3.10 Leaflet, OpenStreetMap y Nominatim

Leaflet [\[15\]](#) es una biblioteca de JavaScript de código abierto diseñada para crear mapas interactivos en páginas web de manera fácil y eficiente. Permite la integración de mapas con diversas funciones, como marcadores, polígonos, líneas y controles de zoom, además de ofrecer soporte para capas base y superposiciones personalizadas. El mapa utilizado es OpenStreetMap (OSM) [\[16\]](#) que es un proyecto colaborativo en línea que crea y proporciona datos geoespaciales gratuitos y editables. Es una alternativa de código abierto a los mapas propietarios, como Google Maps. Nominatim [\[17\]](#) es un motor de búsqueda de código abierto y basado en datos de OpenStreetMap (OSM) utilizada para geocodificar, encontrar coordenadas geográficas a partir de direcciones.

3.11 Suite de Google

La suite de Google [\[18\]](#), también conocida como Google Workspace, ofrece una amplia gama de herramientas colaborativas en la nube, incluyendo Gmail, Google Drive, Docs, Sheets y Slides, que permiten la comunicación, la creación y el almacenamiento de documentos de manera eficiente.

Capítulo 4 - Casos de uso

4.1 Actores

Dentro de la página web se han definido tres tipos de usuarios:

- Usuario registrado como cliente: Este actor puede alquilar alojamientos ofrecidos en la web. Puede modificar los datos de su perfil así como valorar los artículos o realizar búsquedas en la web. Además puede ver el historial de sus artículos alquilados y realizar comentarios sobre estos. Puede dar de baja su cuenta en cualquier momento.
- Usuario registrado como comerciante: Este actor puede publicar y modificar los artículos que quiera ofrecer. Una vez publicado el artículo puede eliminarlo cuando quiera. Puede responder a los comentarios que los usuarios hagan sobre los artículos que ofrece. Puede modificar los datos de su perfil así como dar de baja su cuenta cuando lo desee.
- Usuario no registrado: Puede ver los alojamientos disponibles para alquilar así como navegar por la web. Para poder realizar el resto de acciones tiene que estar dado de alta en la web.

4.2 Diagramas de caso de uso

En concreto se han definido tres diagramas, uno por cada actor que hay en la web. En los siguientes apartados se describen cada uno de ellos y se especifican los casos de uso relacionados con cada uno de ellos.

4.2.1 Diagrama de cliente

En este diagrama se especifican todas las acciones que puede realizar un usuario registrado en la web como cliente.

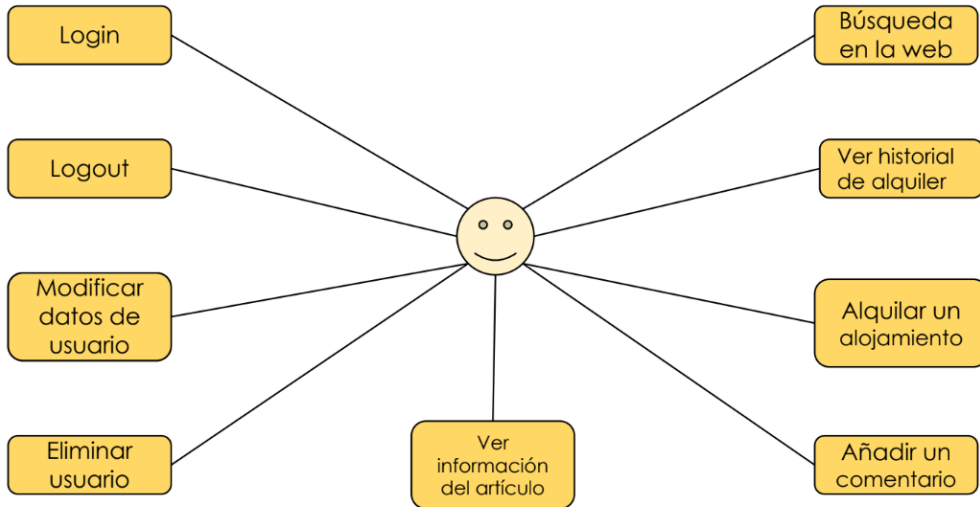


Figura 4.1: Diagrama de cliente

4.2.2 Diagrama de vendedor

En este diagrama se especifican todas las acciones que puede realizar un usuario registrado en la web como vendedor o arrendatario.

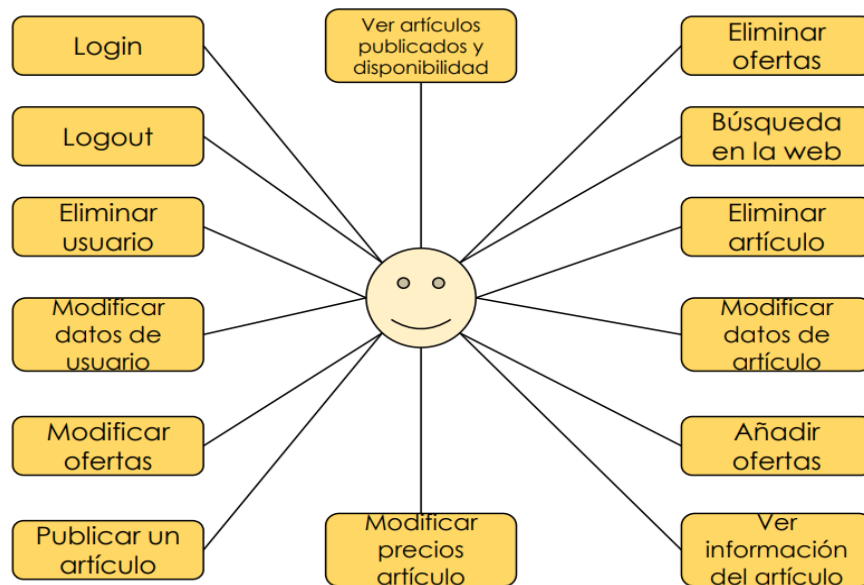


Figura 4.2: Diagrama de vendedor

4.2.3 Diagrama de usuario no registrado

En este diagrama se especifican todas las acciones que pueden realizar los usuarios que no están registrados en la web.

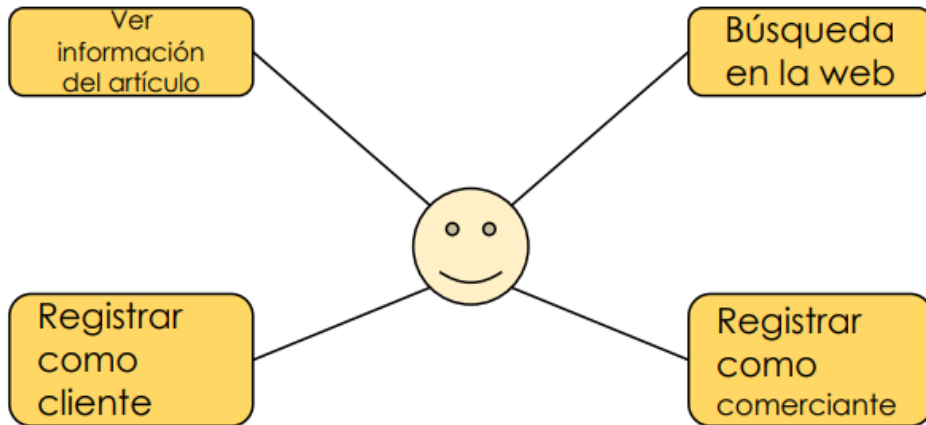


Figura 4.3: Diagrama de usuario no registrado

4.3 Especificación de requisitos

En este apartado se van a especificar los requisitos de los casos de uso definidos para los actores. Los hemos dividido en varios módulos los cuales son:

- Módulo cuentas de usuario
- Módulo alquiler
- Módulo publicar
- Módulo comentarios
- Módulo búsqueda

4.3.1 Módulo cuentas de usuario

Requisito	Registro de usuario
Identificador	1.1

Prioridad	Alta	
Precondición	NA	
Descripción	Los usuarios se crean una cuenta en la web, pudiendo darse de alta como cliente o como vendedor/comerciante.	
Entrada	<ul style="list-style-type: none"> • Registro como cliente: Nombre, Apellidos, Teléfono, Correo electrónico, Fecha de nacimiento, Contraseña. • Registro como vendedor: Nombre de la empresa, Correo electrónico, Teléfono, NIF, Contraseña. 	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario pulsa el botón "Registrar".
	2	El sistema muestra las opciones de registro.
	3	El usuario decide si quiere darse de alta como cliente o como vendedor.
	4	El sistema muestra un formulario con los datos a rellenar.
	5	El usuario rellena los campos del formulario y pulsa el botón.
	6	El sistema valida la información introducida por el usuario.
	7	El sistema muestra la pantalla de inicio.
Postcondición	El usuario se ha creado una cuenta y queda registrado en la web.	
Excepciones	Paso	Acción
	6	Si no se rellena algún campo el sistema muestra un mensaje de error.
	6	Si se introduce un email inválido el sistema muestra un mensaje de error.
	6	Si la contraseña no cumple con los requisitos el sistema muestra un mensaje de error.

	6	Si se introduce un teléfono no válido, el sistema muestra un mensaje de error.
	6	Si el usuario es menor de edad el sistema muestra un mensaje de error.
Comentarios	La contraseña se introduce dos veces y se comprueba que ambas coinciden. Además tiene que tener 8 caracteres mínimo.	
Actores	Usuarios no registrados.	

Tabla 4.1: Especificación registro de usuario

Requisito	Login	
Identificador	1.2	
Prioridad	Alta	
Precondición	El usuario tiene que tener una cuenta creada en la web.	
Descripción	Los usuarios registrados previamente, pueden acceder a su cuenta.	
Entrada	Correo electrónico y Contraseña	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario dentro de la web pulsa el botón de "Login".
	2	El sistema muestra una pantalla con la información requerida.
	3	El usuario introduce los datos pedidos y pulsa el botón de login.
	4	El sistema valida la información.
5	Se inicia sesión y se muestra el perfil del usuario.	
Postcondición	El usuario ha iniciado sesión en la página	

Excepciones	Paso	Acción
	4	Si los datos introducidos no están almacenados en la base de datos el sistema muestra un mensaje de error.
Comentarios	NA	
Actores	Usuario registrado como comerciante o como cliente.	

Tabla 4.2: Especificación Login

Requisito	Log out	
Identificador	1.3	
Prioridad	Alta	
Precondición	El usuario tiene que tener una cuenta creada y haber iniciado sesión en la web.	
Descripción	Los usuarios que han iniciado sesión en la web pueden cerrarla.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario pulsa el botón de "Logout" situado en la cabecera de la web o en el perfil.
	2	El sistema cierra sesión y se muestra la pantalla principal.
Postcondición	Se ha cerrado la sesión del usuario.	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante, usuario cliente.	

Tabla 4.3: Especificación Log out

Requisito	Eliminar cuenta de usuario	
Identificador	1.4	
Prioridad	Media	
Precondición	El usuario tiene que tener una cuenta creada y haber iniciado sesión en la web.	
Descripción	Los usuarios que han iniciado su sesión en la web pueden eliminar su cuenta.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El usuario pulsa el botón de "Eliminar perfil".
	3	El sistema elimina la cuenta del usuario.
	4	El sistema elimina la cuenta y se muestra la pantalla principal.
Postcondición	Se ha eliminado la cuenta del usuario, pasando este a ser un usuario no registrado.	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante, usuario cliente.	

Tabla 4.4: Especificación Eliminar cuenta de usuario

Requisito	Modificar datos de usuario	
Identificador	1.5	
Prioridad	Baja	

Precondición	El usuario tiene que tener una cuenta creada y haber iniciado sesión en la web.	
Descripción	El usuario puede modificar los datos que desee.	
Entrada	Datos que quiera modificar el usuario.	
Salida	Se muestran en su perfil los datos modificados.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra por pantalla el perfil del usuario.
	3	El usuario pulsa el botón de "Editar perfil".
	4	El sistema muestra por pantalla un formulario para modificar los datos deseados.
	5	El usuario modifica los datos que quiera y pulsa el botón "editar".
	6	El sistema valida la información.
7	Se modifican los datos del usuario y se muestra su perfil de usuario.	
Postcondición	Se han modificado los datos deseados.	
Excepciones	Paso	Acción
	6	Si el usuario modifica el correo electrónico por uno que ya está en la base de datos el sistema muestra un mensaje de error.
	6	Si se modifica la contraseña y no cumple con los requisitos el sistema muestra un mensaje de error.
	6	Si se modifica el correo por uno no válido, el sistema muestra un mensaje de error.
6	Si se modifica el teléfono por uno no válido, el sistema	

	muestra un mensaje de error.
Comentarios	La contraseña se introduce dos veces y se comprueba que ambas coinciden. Además esta tiene que tener una longitud de 8 caracteres mínimo.
Actores	Usuario comerciante, usuario cliente.

Tabla 4.5: Especificación modificar datos de usuario

4.3.2 Módulo alquiler

Requisito	Alquilar un alojamiento	
Identificador	2.1	
Prioridad	Alta	
Precondición	El usuario registrado como cliente ha iniciado sesión en la web.	
Descripción	Un usuario registrado como cliente puede realizar la reserva de un alojamiento publicado en la web.	
Entrada	Fecha de entrada, Fecha de salida.	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario pulsa sobre el artículo que está interesado.
	2	El sistema muestra la información detallada del artículo
	3	El usuario introduce la fecha de inicio y fin y pulsa el botón de "Alquilar".
	4	El sistema valida la información.
	5	Se realiza la reserva y se muestra el perfil del usuario.
Postcondición	El usuario ha realizado una reserva.	
Excepciones	Paso	Acción

	4	Si se deja algún campo en blanco el sistema lo notifica.
	4	El sistema comprueba las fechas introducidas.
Comentarios	La fecha de inicio tiene que ser anterior a la de fin. No se puede realizar una reserva para una fecha ya pasada.	
Actores	Usuario cliente.	

Tabla 4.6: Especificación alquilar un alojamiento

Requisito	Mostrar historial de reservas	
Identificador	2.2	
Prioridad	Media	
Precondición	El usuario registrado como cliente ha iniciado sesión y ha realizado alguna reserva.	
Descripción	El cliente puede ver en su perfil un historial de las reservas que ha realizado.	
Entrada	NA	
Salida	Historial de reservas realizadas.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos alquilados.
	3	El usuario pulsa sobre algún artículo reservado.
	4	El sistema muestra la información detallada del artículo.
Postcondición	NA	
Excepciones	Paso	Acción
Comentarios	NA	

Actores	Usuario cliente.
----------------	------------------

Tabla 4.7: Especificación mostrar historial de reservas

4.3.3 Módulo publicar

Requisito	Publicar un artículo	
Identificador	3.1	
Prioridad	Alta	
Precondición	El usuario registrado como comerciante, ha iniciado sesión.	
Descripción	Un usuario comerciante puede publicar un artículo para que los clientes lo alquilen.	
Entrada	Nombre alojamiento, Dirección, Ciudad, País, Descripción, Hora check in, Hora check out, Tipo Alojamiento, Nombre de la habitación, Número máximo de personas, Número de habitaciones, Tipo de cama, Número de camas.	
Salida	Nuevo artículo publicado en la web.	
Secuencia normal	Paso	Acción
	1	El usuario pulsa el botón "Publicar artículo" que aparece en la cabecera de la web o en el perfil del usuario.
	2	El sistema muestra el formulario para publicar un artículo.
	3	El usuario rellena el primer formulario con la información del artículo y pulsa continuar.
	4	El sistema valida la información y si es correcta se muestra un formulario para dar de alta habitaciones.
	5	El usuario rellena el segundo formulario con la información de las habitaciones y pulsa continuar.
6	El sistema valida la información y si es correcta se muestra un formulario para indicar el precio del alquiler.	

	7	El usuario rellena el tercer formulario con la información de los precios y pulsa continuar.
	8	El sistema valida la información.
	9	Se publica el artículo en la web y el sistema muestra la vista con la información de dicho artículo.
Postcondición	Se ha publicado un nuevo artículo en la web.	
Excepciones	Paso	Acción
	4	Si no se rellena algún campo el sistema muestra un mensaje de error.
	4	Si se introduce una dirección que no existe el sistema muestra un mensaje de error.
	6	Si no se rellena algún campo el sistema muestra un mensaje de error.
	6	Si en algún campo numérico se introduce un valor menor o igual que 0 el sistema muestra un mensaje de error.
	8	Si se introduce un valor menor o igual que 0 el sistema muestra un mensaje de error.
	8	Si no se rellena algún campo el sistema muestra un mensaje de error.
Comentarios	La dirección introducida se comprueba con un mapa para comprobar que no es incorrecta.	
Actores	Usuario comerciante.	

Tabla 4.8: Especificación publicar un artículo

Requisito	Eliminar un artículo
Identificador	3.2
Prioridad	Media

Precondición	El usuario registrado como comerciante, ha iniciado sesión y ha publicado algún artículo.	
Descripción	El usuario comerciante puede eliminar un artículo que ha publicado.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
	3	El usuario pulsa el botón "Eliminar".
4	El sistema borra el artículo seleccionado, muestra un mensaje indicando que la acción ha tenido éxito y muestra el perfil del usuario sin el artículo eliminado.	
Postcondición	Se elimina el alojamiento del usuario de la base de datos.	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante.	

Tabla 4.9: Especificación eliminar un artículo

Requisito	Modificar datos de un artículo
Identificador	3.3
Prioridad	Media
Precondición	El usuario registrado como comerciante, ha iniciado sesión y ha publicado algún artículo.
Descripción	El usuario comerciante puede modificar la información de sus artículos publicados.

Entrada	Nombre alojamiento, Dirección, Ciudad, País, Descripción, Hora check in, Hora check out, Tipo Alojamiento, Nombre de la habitación, Número máximo de personas, Número de habitaciones, Tipo de cama, Número de camas.	
Salida	Se muestra el artículo con los datos modificados.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
	3	El usuario pulsa el botón "Editar información".
	4	El sistema muestra por pantalla un formulario para modificar los datos deseados.
	5	El usuario modifica los datos que quiera y pulsa el botón "Guardar cambios".
	5.1	Si el usuario pulsa el botón "cancelar", no se realizan los cambios y se vuelve a la pantalla del perfil.
	6	El sistema valida la información.
7	Se modifican los datos del artículo y se muestra el artículo con la información modificada.	
Postcondición	Se han modificado los datos deseados del artículo.	
Excepciones	Paso	Acción
	6	Si en algún campo numérico se introduce un valor menor o igual que 0 el sistema muestra un mensaje de error.
	6	Si se introduce una dirección que no existe el sistema muestra un mensaje de error.
Comentarios	La dirección introducida se comprueba con un mapa para comprobar que no es incorrecta.	

Actores	Usuario comerciante.
----------------	----------------------

Tabla 4.10: Especificación modificar datos de un artículo

Requisito	Modificar precios de un artículo	
Identificador	3.4	
Prioridad	Media	
Precondición	El usuario registrado como comerciante, ha iniciado sesión y ha publicado algún artículo.	
Descripción	El usuario comerciante puede modificar los precios de sus artículos publicados.	
Entrada	Precio	
Salida	Se muestra el artículo con el precio modificado.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
	3	El usuario pulsa el botón "Editar precios y ofertas".
	4	El sistema muestra por pantalla un formulario para modificar los datos deseados.
	5	El usuario modifica los datos que quiera y pulsa el botón "Continuar".
	6	El sistema valida la información.
7	Se modifica el precio del artículo y se muestra el artículo con la información modificada.	
Postcondición	Se han modificado los datos deseados del artículo.	
Excepciones	Paso	Acción

	6	Si se introduce un valor menor o igual que 0 el sistema muestra un mensaje de error.
Comentarios	NA	
Actores	Usuario comerciante.	

Tabla 4.11: Especificación modificar precios de un artículo

Requisito	Publicar oferta	
Identificador	3.5	
Prioridad	Baja	
Precondición	El usuario registrado como comerciante, ha iniciado sesión y ha publicado algún artículo.	
Descripción	El usuario comerciante puede modificar los precios de sus artículos publicados.	
Entrada	Precio por noche, Fecha de inicio, Fecha de fin.	
Salida	Se muestra el artículo con una nueva oferta	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
	3	El usuario pulsa el botón "Editar precios y ofertas".
	4	El sistema muestra por pantalla la información de los precios.
	5	El usuario pulsa el botón "Publicar oferta"
	6	El sistema muestra por pantalla el formulario de ofertas.
	7	El usuario rellena el formulario y pulsa el botón "Insertar oferta".

	8	El sistema valida la información introducida.
	9	Se añade la oferta y se muestra el artículo.
Postcondición	Se ha añadido una nueva oferta al artículo.	
Excepciones	Paso	Acción
	8	Si se introduce un valor menor o igual que 0 el sistema muestra un mensaje de error.
	8	Si se deja algún campo en blanco el sistema lo notifica.
	8	El sistema comprueba las fechas introducidas.
Comentarios	La fecha de inicio tiene que ser anterior a la de fin. No se puede publicar una oferta para una fecha ya pasada.	
Actores	Usuario comerciante.	

Tabla 4.12: Especificación publicar oferta

Requisito	Eliminar ofertas	
Identificador	3.6	
Prioridad	Media	
Precondición	El usuario registrado como comerciante, ha iniciado sesión, ha publicado algún artículo y tiene alguna oferta para dicho artículo.	
Descripción	El usuario comerciante puede eliminar una oferta publicada para alguno de sus artículos.	
Entrada	NA	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial

		de artículos publicados.
	3	El usuario pulsa el botón "Editar precios y ofertas".
	4	El sistema muestra por pantalla el formulario necesario.
	5	El usuario pulsa el botón "Publicar oferta"
	6	El sistema muestra una tabla con las ofertas publicadas para dicho artículo.
	7	El usuario pulsa el botón "Eliminar"
	8	El sistema borra la oferta y muestra un mensaje indicando que la acción ha tenido éxito.
Postcondición	Se elimina la oferta de dicho artículo.	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante.	

Tabla 4.13: Especificación eliminar ofertas

Requisito	Modificar ofertas
Identificador	3.7
Prioridad	Media
Precondición	El usuario registrado como comerciante, ha iniciado sesión, ha publicado algún artículo y tiene alguna oferta para dicho artículo.
Descripción	El usuario comerciante puede modificar una oferta publicada para alguno de sus artículos.
Entrada	Precio por noche, Fecha de inicio, Fecha de fin.
Salida	Se muestra en la tabla de ofertas, la oferta con los datos modificados.

Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
	3	El usuario pulsa el botón "Editar precios y ofertas".
	4	El sistema muestra por pantalla la información de los precios.
	5	El usuario pulsa el botón "Publicar oferta"
	6	El sistema muestra por pantalla las ofertas publicadas.
	7	El usuario pulsa el botón "Modificar" de la oferta que desee.
	8	El sistema muestra el formulario para modificar la oferta.
	9	El sistema valida la información introducida.
10	Se modifica la oferta y aparece la nueva información en la tabla de ofertas.	
Postcondición	Se ha modificado la oferta de un artículo.	
Excepciones	Paso	Acción
	9	Si se introduce un valor menor o igual que 0 el sistema muestra un mensaje de error.
	9	Si se deja algún campo en blanco el sistema lo notifica.
9	El sistema comprueba las fechas introducidas.	
Comentarios	La fecha de inicio tiene que ser anterior a la de fin. No se puede publicar una oferta para una fecha ya pasada.	
Actores	Usuario comerciante.	

Tabla 4.14: Especificación modificar ofertas

Requisito	Mostrar historial de artículos publicados
------------------	--

Identificador	3.8	
Prioridad	Media	
Precondición	El usuario registrado como comerciante, ha iniciado sesión y ha publicado algún artículo.	
Descripción	El usuario comerciante puede ver en su perfil un historial de artículos publicados.	
Entrada	NA	
Salida	Historial de artículos publicados.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
Postcondición	NA	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante.	

Tabla 4.15: Especificación mostrar historial de artículos publicados

Requisito	Ver reservas del artículo	
Identificador	3.9	
Prioridad	Media	
Precondición	El usuario registrado como comerciante, ha iniciado sesión y ha publicado algún artículo.	
Descripción	El usuario comerciante puede comprobar la disponibilidad del artículo que desee.	

Entrada	NA	
Salida	Información de disponibilidad del artículo.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos publicados.
	3	El usuario pulsa el botón "Ver reservas".
	4	El sistema muestra una pantalla con la disponibilidad de dicho artículo.
Postcondición	NA	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante.	

Tabla 4.16: Especificación ver reservas del artículo

4.3.4 Módulo comentarios

Requisito	Comentar un alojamiento
Identificador	4.1
Prioridad	Baja
Precondición	Estar registrado como cliente, haber reservado un alojamiento y que la fecha de la reserva haya pasado.
Descripción	Un cliente podrá comentar y puntuar los alojamientos en los que se haya hospedado una vez la fecha haya pasado.
Entrada	NA
Salida	NA

Secuencia normal	Paso	Acción
	1	El usuario accede a la pestaña de su perfil.
	2	El sistema muestra la información del usuario con un historial de artículos alquilados.
	3	El usuario pulsa sobre algún artículo reservado.
	4	El sistema muestra la información detallada del artículo.
	5	El usuario realiza un comentario y pulsa el botón comentar.
	6	El sistema publica el comentario y muestra la información del alojamiento con el nuevo comentario.
Postcondición	Se ha añadido una nueva valoración a un alojamiento.	
Excepciones	Paso	Acción
Comentarios	Se puede puntuar los alojamientos en los que se haya hospedado una vez la fecha haya pasado.	
Actores	Usuario cliente.	

Tabla 4.17: Especificación Comentar un alojamiento

4.3.5 Módulo búsqueda

Requisito	Buscar alojamientos
Identificador	5.1
Prioridad	Media
Precondición	NA
Descripción	El usuario puede realizar búsquedas en la web para encontrar el alojamiento que más se adecue a sus necesidades.
Entrada	Información a buscar
Salida	Alojamientos que tengan algo en común con la búsqueda.

Secuencia normal	Paso	Acción
	1	El usuario escribe en la barra de búsqueda lo que quiere encontrar.
	2	El usuario pulsa el botón de "search" de la barra de búsqueda.
	3	El sistema busca en la base de datos alojamientos relacionados con la información introducida en la barra de búsqueda.
	3.1	Si se han encontrado alojamientos relacionados con la búsqueda, se muestra una lista con la información reducida de los alojamientos. Si quiere hacer una búsqueda más específica puede usar los filtros.
	3.2	Si no se ha encontrado ningún alojamiento relacionado con la búsqueda el sistema muestra un mensaje indicándolo.
	4	Si se han encontrado resultados y el usuario pulsa sobre uno de ellos el sistema muestra la información detallada del alojamiento.
Postcondición	NA	
Excepciones	Paso	Acción
Comentarios	NA	
Actores	Usuario comerciante, usuario cliente, usuario no registrado.	

Tabla 4.18: Especificación Buscar alojamientos

Capítulo 5 - Arquitectura de la aplicación y modelo de Datos

5.1 Arquitectura de la aplicación

Para el desarrollo de la página web se ha utilizado la arquitectura cliente-servidor que es un modelo de comunicación donde un cliente solicita servicios a un servidor a través de una red. El servidor responde a estas solicitudes proporcionando los recursos o servicios solicitados. Esta estructura facilita la distribución de tareas y la escalabilidad en sistemas informáticos.

En este caso, los clientes son los usuarios (usuario registrado como cliente, usuario registrado como comerciante y usuario no registrado). Estos solicitan a través del navegador información sobre los alojamientos y los servicios que ofrece la página. Estas peticiones se envían al servidor web que en este caso es Apache. El servidor interactúa con la base de datos que es MySQL para recuperar la información deseada. Una vez procesada la solicitud el servidor envía dicha respuesta al cliente mostrando la información y servicios solicitados mediante la interfaz web. Esto permite gestionar múltiples peticiones simultáneamente garantizando un servicio rápido y eficiente. En la Figura 5.0 se puede observar la arquitectura de la aplicación utilizada durante el desarrollo.

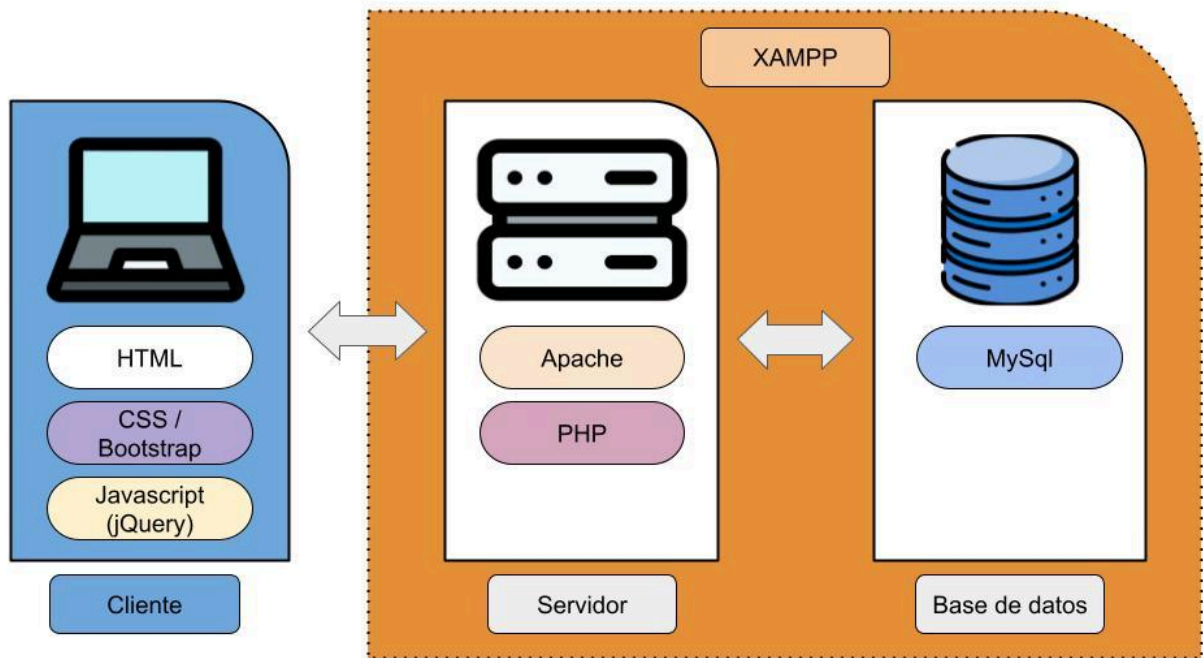


Figura 5.0: Arquitectura de la aplicación

5.2 Modelo de Datos

En la base de datos se almacena información de los usuarios registrados, alojamientos, reservas de alojamientos o comentarios que los usuarios hacen. Para recopilar esta información, se ha dividido la base de datos en varias tablas, en concreto 13.

<input type="checkbox"/>	alojamientos	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	alquileres	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	articulos	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	camas	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	caracteristicas	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	caracteristicasarticulo	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	comentarios	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	customers	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	habitaciones	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	ofertas	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	precios	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	sellers	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar
<input type="checkbox"/>	users	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar

Figura 5.1: Estructura de la base de datos

Como se detalló en el Capítulo 3 (Tecnología usada), se ha utilizado XAMPP [4] que incluye Apache, un servidor web de código abierto que procesa solicitudes HTTP, y MySQL, un sistema de gestión de bases de datos relacionales para almacenar y administrar datos en sitios web dinámicos.

Para definir la base de datos lo primero que se realizó fue identificar las entidades básicas de la aplicación. En nuestro caso son los usuarios, artículos y los alquileres. Una vez definidas las entidades, se especificaron los campos que cada tabla tiene que almacenar y su tipo (String, int, etc). Además para identificar entidades se definen claves primarias las cuales son únicas en esa tabla (por ejemplo el id de un artículo). También se pueden relacionar las tablas entre ellas, para ello es necesario crear una clave foránea. A continuación se explica en detalle cada una de las tablas y sus campos, también se muestra un gráfico donde se muestra el modelo Relacional de la base de datos.

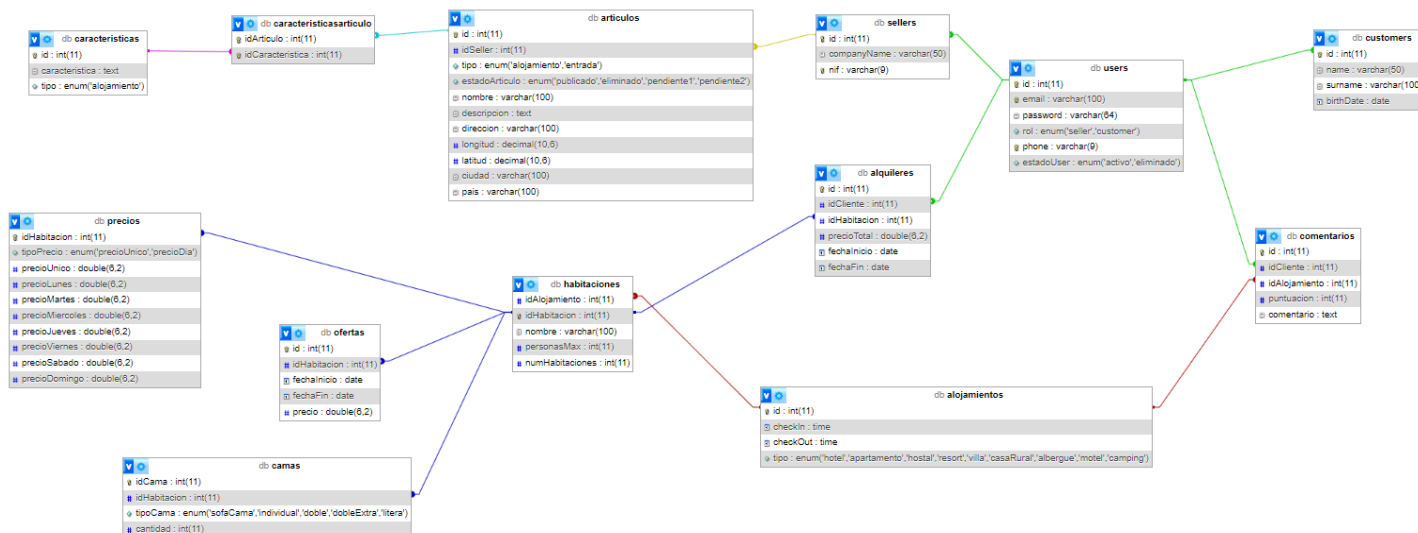


Figura 5.2: Modelo relacional de la base de datos

5.2.1 Tabla users

En esta tabla se guarda la información general de los usuarios que se registran en la web. Al haber dos tipos de usuarios registrados la información específica de cada uno se guarda en otra tabla, en el campo rol se indica si se trata de un usuario registrado como cliente o como vendedor. A cada usuario se le asigna un identificador único, siendo esta la clave primaria de la tabla. Además se guarda el email, necesario para hacer login y la contraseña almacenada en hash para garantizar la seguridad y privacidad de los usuarios. Además para saber si la cuenta ha sido eliminada, se ha definido el campo “estadoUser”.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	int(11)	No		
email	varchar(100)	No		
password	varchar(64)	No		
rol	enum('seller', 'customer')	No		
phone	varchar(9)	No		
estadoUser	enum('activo', 'eliminado')	No		

Figura 5.3: Tabla users

5.2.2 Tabla sellers

En esta tabla se guarda la información específica de los usuarios registrados como comerciantes como por ejemplo el nombre de la empresa o el NIF. Se relaciona con la tabla users mediante la clave primaria id.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
companyName	varchar(50)	No		
nif	varchar(9)	No		

Figura 5.4: Tabla sellers

5.2.3 Tabla customers

Se guarda la información específica de los usuarios registrados como clientes como puede ser el nombre, apellido y su fecha de nacimiento para asegurarnos de que quien realiza la reserva es mayor de edad. Se relaciona con la tabla users mediante la clave primaria id.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
name	varchar(50)	No		
surname	varchar(100)	No		
birthDate	date	No		

Figura 5.5: Tabla customers

5.2.4 Tabla articulos

En esta tabla se almacena la información general de los artículos publicados en la web como es la descripción, dirección, nombre, ciudad o país. La longitud y latitud se almacena directamente al indicar la dirección. Cada artículo tiene un id único siendo este la clave primaria y además, para saber quién ha publicado ese artículo se guarda el id del comerciante. Como trabajo futuro, se ha pensado implementar

nuevos tipos de artículos, para saber de qué artículo se trata, se ha definido un enumerado “tipo” para especificarlo.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
idSeller	int(11)	No		
tipo	enum('alojamiento', 'entrada')	No		
estadoArticulo	enum('publicado', 'eliminado', 'pendiente1', 'pendiente2')	No		
nombre	varchar(100)	No		
descripcion	text	No		
direccion	varchar(100)	No		
longitud	decimal(10,6)	No		
latitud	decimal(10,6)	No		
ciudad	varchar(100)	No		
pais	varchar(100)	No		

Figura 5.6: Tabla articulos

5.2.5 Tabla alojamientos

En esta tabla se guarda la información específica del alojamiento como es la hora de check in o check out además del tipo de alojamiento del que se trata. Se relaciona con la tabla “articulos” mediante la clave primaria id.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
checkIn	time	No		
checkOut	time	No		
tipo	enum('hotel', 'apartamento', 'hostal', 'resort', 'villa', 'casaRural', 'albergue', 'motel', 'camping')	No		

Figura 5.7: Tabla alojamientos

5.2.6 Tabla habitaciones

Un alojamiento puede tener más de una habitación, por ejemplo si se trata de un hotel no todas las habitaciones van a tener la misma capacidad. Por tanto, se ha definido una tabla para definir cada habitación. Se relaciona con la tabla alojamientos mediante el “idAlojamiento”.

Columna	Tipo	Nulo	Predeterminado	Comentarios
idAlojamiento	int(11)	No		
idHabitacion (<i>Primaria</i>)	int(11)	No		
nombre	varchar(100)	No		
personasMax	int(11)	No		
numHabitaciones	int(11)	No		

Figura 5.8: Tabla habitaciones

5.2.7 Tabla camas

Cada cama va a tener información específica como puede ser el tipo de cama del que se trata o el número de personas que pueden dormir en ella (campo "cantidad"). Para saber la habitación a la que pertenece dicha cama se ha añadido el campo "idHabitacion" relacionándose así con la tabla de habitaciones.

Columna	Tipo	Nulo	Predeterminado	Comentarios
idCama (<i>Primaria</i>)	int(11)	No		
idHabitacion	int(11)	No		
tipoCama	enum('sofaCama', 'individual', 'doble', 'dobleExtra', 'litera')	No		
cantidad	int(11)	No		

Figura 5.9: Tabla camas

5.2.8 Tabla alquileres

En esta tabla se almacena toda la información relacionada con las reservas que los clientes hacen en la web. Cada reserva tiene un identificador único, siendo esta la clave primaria. Además para saber quién ha reservado la habitación se almacena el id del cliente ("idCliente") y para saber qué habitación se ha reservado se indica el id de la habitación ("idHabitacion"). También se guarda el precio total de la reserva y la fecha de inicio y fin de la reserva.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
idCliente	int(11)	No		
idHabitacion	int(11)	No		
precioTotal	double(6,2)	No		
fechaInicio	date	No		
fechaFin	date	No		

Figura 5.10: Tabla alquileres

5.2.9 Tabla precios

En esta tabla se especifica el precio de cada habitación. Se puede establecer tanto un precio único como un precio distinto dependiendo del día de la semana del que se trate. La clave primaria es "idHabitacion" relacionándose esta tabla con la tabla "habitaciones".

Columna	Tipo	Nulo	Predeterminado	Comentarios
idHabitacion (<i>Primaria</i>)	int(11)	No		
tipoPrecio	enum('precioUnico', 'precioDia')	No		
precioUnico	double(6,2)	No		
precioLunes	double(6,2)	No		
precioMartes	double(6,2)	No		
precioMiercoles	double(6,2)	No		
precioJueves	double(6,2)	No		
precioViernes	double(6,2)	No		
precioSabado	double(6,2)	No		
precioDomingo	double(6,2)	No		

Figura 5.11: Tabla precios

5.2.10 Tabla ofertas

Para algunas fechas del año puede que el usuario arrendatario quiera ofrecer alguna oferta sobre una habitación en concreto. Por tanto, en esta tabla se especifica para cierta habitación identificada mediante el campo "idHabitacion" el precio que se establece para el rango de días especificados mediante los campos "fechaInicio" y "fechaFin". Cada oferta tiene un id único siendo este la clave primaria de la tabla.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
idHabitacion	int(11)	No		
fechaInicio	date	No		
fechaFin	date	No		
precio	double(6,2)	No		

Figura 5.12: Tabla ofertas

5.2.11 Tabla características

Se ha definido una tabla para las distintas características que puede tener un artículo. Cada característica está definida por un id que a su vez es la clave primaria y un campo de tipo text donde se indica de qué características se trata.

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
caracteristica	text	No		
tipo	enum('alojamiento', 'entrada')	No		

Figura 5.13: Tabla características

5.2.12 Tabla característicasarticulo

Se ha diseñado una tabla para relacionar cada artículo con sus características. La clave primaria es "idArticulo" que identifica al artículo en concreto y para saber de qué característica se trata, se almacena "idCaracteristica" que se relaciona con la tabla características.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios
<input type="checkbox"/> 1	idArticulo 	int(11)			No	Ninguna	
<input type="checkbox"/> 2	idCaracteristica 	int(11)			No	Ninguna	

Figura 5.14: Tabla característicasarticulos

5.2.13 Tabla comentarios

Se ha definido esta tabla para almacenar los comentarios que los usuarios publican en la web. Pueden puntuar cada alojamiento y añadir un comentario. Para saber de qué artículo se trata se ha definido el campo "idAlojamiento" que a su vez se relaciona con la tabla alojamientos.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios
<input type="checkbox"/>	1 id 	int(11)			No	<i>Ninguna</i>	
<input type="checkbox"/>	2 idCliente	int(11)			No	<i>Ninguna</i>	
<input type="checkbox"/>	3 idAlojamiento	int(11)			No	<i>Ninguna</i>	
<input type="checkbox"/>	4 puntuacion	int(11)			No	<i>Ninguna</i>	
<input type="checkbox"/>	5 comentario	text	utf8mb4_general_ci		No	<i>Ninguna</i>	

Figura 5.15: Tabla comentarios

Capítulo 6 - Implementación y diseño de la aplicación

En este apartado se explican brevemente los detalles del diseño de la aplicación web además de su implementación. Para hacer más fácil el seguimiento de la explicación, hemos dividido este capítulo en varios módulos, en concreto 5:

- Módulos cuentas de usuario: recoge todo lo relacionado con el usuario, es decir, registro en la web, login, logout, modificación de sus datos.
- Módulo alquiler: se detalla la funcionalidad de alquilar alojamientos en la web.
- Módulo publicar: se indica la implementación de la funcionalidad de publicar ya sean alojamientos, ofertas o modificar los datos del alojamiento.
- Módulo comentarios: se muestra la implementación para que los clientes realicen comentarios y puntúen los alojamientos de la web.
- Módulo búsqueda: se indica la implementación de los filtros de la web para realizar búsquedas en la misma.

6.1 Diseño de la aplicación

Para el diseño de la aplicación se ha utilizado una interfaz intuitiva y atractiva que permite a los usuarios buscar, publicar y reservar alojamientos de manera fácil y rápida.

Para que la navegación por la página web sea cómoda se ha prestado especial atención a las disposición de los distintos botones y opciones. Además se han usado colores cálidos en su mayoría que da una apariencia moderna y hace que la página sea más atractiva.

Para resaltar las opciones y botones que ofrece la página se ha utilizado un color mostaza para dar luminosidad y hacer que estos destaquen. Se ha decidido utilizar este tono ya que evoca una sensación de calidez, vitalidad y energía que son características de la temporada de verano que es cuando la gente más aprovecha para ir de vacaciones. Además para los botones de acciones sin vuelta atrás, como por ejemplo el botón de eliminar perfil o de eliminar un artículo, se ha usado el color rojo para destacar y resaltar más esta opción. Por último para botones secundarios, como el botón de "logout", se ha utilizado el color gris para dar un contraste sutil pero efectivo, ayudando a que este se destaque sin ser demasiado llamativo.

Para garantizar a los usuarios que su experiencia en la página sea satisfactoria y positiva se ha hecho uso de una estructura clara, limpia y con opciones destacadas y reconocibles fácilmente. Además para evitar que la página quedase sobrecargada de información se ha prestado especial atención a los distintos formularios para hacer que estos sean sencillos.

6.2 Implementación de la aplicación web

6.2.1 Clases

En el marco del desarrollo de aplicaciones, la utilización de clases juega un papel fundamental en la estructura y organización del código. Las clases proporcionan un mecanismo eficiente para encapsular datos y funcionalidades relacionadas, promoviendo la modularidad, reutilización y mantenibilidad del código. El siguiente diagrama muestra las clases, sus atributos y métodos y la relación entre ellas.

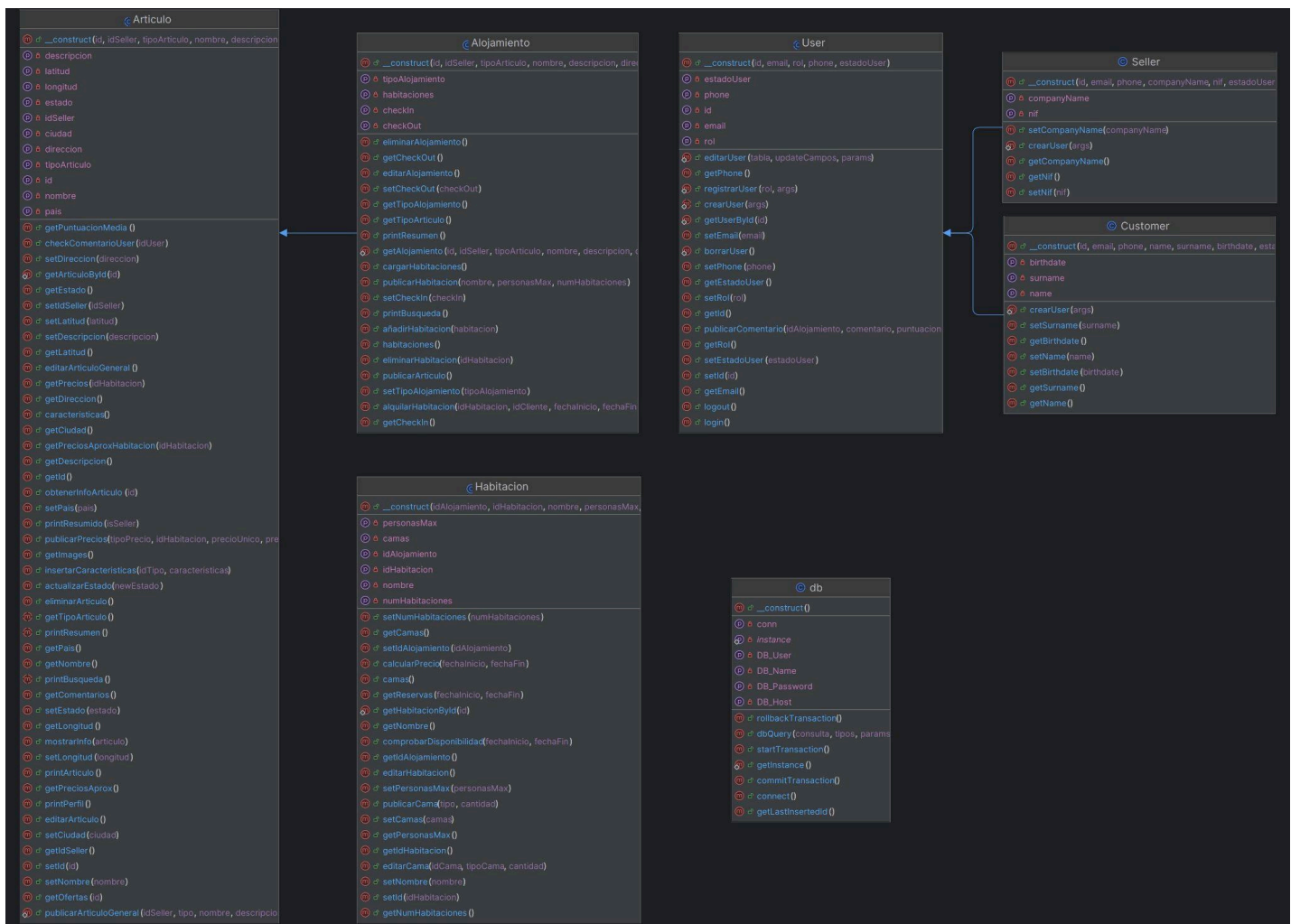


Figura 6.1 Estructura de clases

Clase Artículo

Clase abstracta de la que hereda Alojamiento (y en el caso de tener otros tipos de artículos, también). Tiene como atributos los campos de la tabla Artículos de la base de datos. Los métodos que tiene, además de los *getters* y *setters* de los atributos, son los relacionados con la gestión de dichos artículos: crear, eliminar, modificar la información de los mismos, sus precios y comentarios.

Clase Alojamiento

Clase heredada de Artículo. Sus métodos se utilizan para crear, eliminar, modificar la información de los mismos, además de la gestión de las habitaciones que posean.

Clase Habitación

Clase encargada de gestionar la información de las habitaciones. Entre sus métodos se encuentran los encargados de crear y modificar camas, calcular la disponibilidad de la habitación o calcular su precio.

Clase User

Clase abstracta que tiene como atributos la información básica de los usuarios. Sus métodos se encargan de crear, modificar y eliminar usuarios.

Clases Customer y Seller

Clases heredadas de User. Son las encargadas de representar al cliente y al vendedor respectivamente. Se encargan de crear el usuario de cada tipo así como almacenar la información específica de cada uno

Clase db

Es la clase encargada de gestionar la base de datos, las conexiones y las consultas. Se utiliza el patrón de diseño Singleton para garantizar que la base de datos solo tenga una instancia y una conexión y proporcionar un punto de acceso global a ella.

6.2.2 Módulo cuentas de usuario

En esta sección se detalla la implementación de las funcionalidades que son comunes para todos los usuarios de la web. Estas funcionalidades son: Registro, Login, Logout, Modificar los datos de su perfil y Eliminar su cuenta.

Registro

Un usuario se puede dar de alta como cliente o como vendedor. El formulario a rellenar será distinto dependiendo de la opción elegida. Si el usuario quiere registrarse como cliente el formulario que tiene que rellenar es el siguiente:



El formulario de registro como cliente tiene un botón "Volver" en la parte superior izquierda. El título principal es "Crea una cuenta". Los campos de entrada están organizados en una cuadrícula:

- Nombre:
- Apellidos:
- Teléfono:
- Correo electrónico:
- Fecha de nacimiento:
- Contraseña:
- Repita la contraseña:

En la parte inferior hay un botón amarillo que dice "Registrar como cliente".

Figura 6.2 Registro como cliente

Por el contrario, si se quiere registrar como vendedor el formulario que el sistema muestra es el siguiente:

Volver

Crea una cuenta

Nombre de la empresa	Correo electrónico
<input type="text" value="Nombre de la empresa"/>	<input type="text" value="Correo electrónico"/>
Teléfono	NIF
<input type="text" value="Teléfono"/>	<input type="text" value="NIF"/>
Contraseña	Repita la contraseña
<input type="text" value="Contraseña"/>	<input type="text" value="Repetir contraseña"/>

Registrar como vendedor

Figura 6.3 Registro como vendedor

Como se indica en el apartado 5.2 tenemos una tabla general para almacenar la información que comparten los distintos tipos de usuarios, en este caso es el teléfono, correo electrónico y la contraseña. El código implementado para insertar datos en esta tabla se encuentra en la clase "User.php" y es el siguiente:

```

public static function registrarUser($rol, ...$args){

    require_once $_SERVER['DOCUMENT_ROOT']."/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    $db->startTransaction();

    $query = "INSERT INTO users (rol, password, email, phone, estadoUser) VALUES (?, ?, ?, ?,?)";
    $resultado = $db->dbQuery($query, "sssss", $rol, hash("sha256", $args[0]), $args[1], $args[2], 'activo');

    if($resultado === false){
        $db->rollbackTransaction();
        return false;
    }

    $id = $db->getLastInsertedId();

    if($rol == "customer"){
        //id, name, surname, birthDate
        $resultado = Customer::crearUser($id, $args[3], $args[4], $args[5]);
    }
    else if($rol == "seller")
        //id, companyname, nif
        $resultado = Seller::crearUser($id, $args[3], $args[4]);
    else {
        $db->rollbackTransaction();
        error("errorInterno");
    }

    if($resultado === false){
        $db->rollbackTransaction();
        return false;
    }

    $db->commitTransaction();
    return $id;
}

```

Figura 6.4 Código registrar user

Para almacenar la información particular de cada uno se han diseñado dos tablas. Para insertar el resto de datos del cliente la función implementada es la siguiente (clase "Customer.php"):

```

public static function crearUser(...$args){
    require_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    $query = "INSERT INTO customers (id, name, surname, birthDate) VALUES (?, ?, ?, ?)";
    $resultado = $db->dbQuery($query, "iss", $args[0], $args[1], $args[2], $args[3]);

    if($resultado === false){
        $db->rollbackTransaction();
        return false;
    }

    return true;
}

```

Figura 6.5 Código crear user customer

En el caso del comerciante la implementación se encuentra en la clase "Seller.php" y es la siguiente:

```

public static function crearUser(...$args){
    require_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    $query = "INSERT INTO sellers (id, companyName, nif) VALUES (?, ?, ?)";
    $resultado = $db->dbQuery($query, "iss", $args[0], $args[1], $args[2]);

    if($resultado === false){
        $db->rollbackTransaction();
        return false;
    }

    return true;
}

```

Figura 6.6 Código crear user seller

Login

Para hacer login el usuario tiene que introducir su correo y contraseña. La vista es la siguiente:

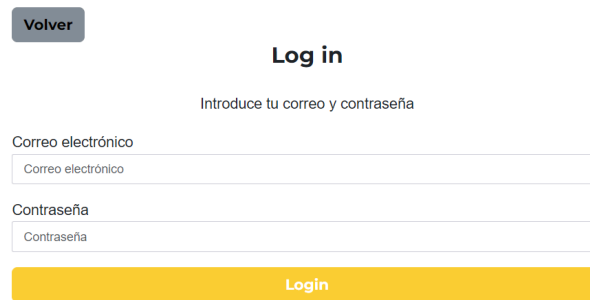


Figura 6.7 Formulario login

El código implementado para este formulario está en la clase "login.php" y es el siguiente:

```
<section id="perfil" style="padding-top: 0%;">
  <div class="container" >
    <body>
      <div class="container">
        <div class="row d-flex d-xl-flex justify-content-center justify-content-xl-center">
          <div class="col-sm-12 col-lg-10 col-xl-9 col-xxl-7 bg-white shadow-lg" style="border-radius: 5px">
            <div class="p-5">
              <a href='index.php' class='btn btn-secondary' style='margin-top:0px; margin-left:0%'>Volver</a>

              <div class='text-center'>
                <h4 class='text-dark mb-4'>Log in</h4>
                <p class="w-lg-50">Introduce tu correo y contraseña</p>
              </div>

              <form action="includes/fLogin.php" method="post">
                <label for="form-control form-control-user">Correo electrónico</label>
                <div class="mb-3"><input class="form-control" type="email" name="email" placeholder="Correo electrónico"></div>
                <label for="form-control form-control-user">Contraseña</label>
                <div class="mb-3"><input class="form-control" type="password" name="password" placeholder="Contraseña"></div>
                <div class="mb-3"><button class="btn btn-primary d-block w-100" type="submit" name="submit">Login</button></div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </body>
  </div>
</section>
```

Figura 6.8 Código formulario login

El sistema comprueba que el correo electrónico y la contraseña están almacenados en la base de datos. En la siguiente imagen se muestra el código donde

se comprueba que la información introducida es correcta. Este código está en la clase "fLogin.php".

```
if(isset($_POST["submit"])){

    if(empty($_POST["email"]) | empty($_POST["password"])){
        error("camposVacios");
    }

    $email = $_POST["email"];
    $passwordHash = hash("sha256", $_POST["password"]);

    require_once $_SERVER["DOCUMENT_ROOT"] . "/includes/funcionesAuxiliares.php";
    require_once $_SERVER["DOCUMENT_ROOT"] . "/classes/User.php";

    $db = getDbInstance();

    $consulta = "SELECT * FROM users WHERE email = ? AND password = ? AND estadoUser != 'eliminado'";
    $resultado = $db->dbQuery($consulta, "ss", $email, $passwordHash);

    if($resultado == false){
        error("errorInterno");
    }

    if ($fila = mysqli_fetch_assoc($resultado)) {
        $id = $fila["id"];
        $user = User::getUserById($id);
    }
    else {
        error("loginError");
    }

    $user->login();

    header("Location: ../perfil.php");
    exit();
}
else {
    error("errorInterno");
}
```

Figura 6.9 Código login

Logout

Para cerrar sesión, el usuario tiene que pulsar el botón de "Logout" que aparece en la cabecera de la web además de en el perfil del usuario. Después se muestra la

pantalla principal de la aplicación. El código implementado se encuentra en el archivo "fLogout.php" y es el siguiente:

```
<?php
session_start();
session_unset();
session_destroy();

header("Location: /index.php");
exit();
```

Figura 6.10 Código logout

Ver y editar datos de usuario

El usuario puede ver los datos de su cuenta en la pestaña "Perfil". Dependiendo del tipo de usuario, la información a mostrar es distinta. Un usuario registrado como cliente además de ver su información personal, puede ver un historial de reservas que haya realizado. La vista será la siguiente:

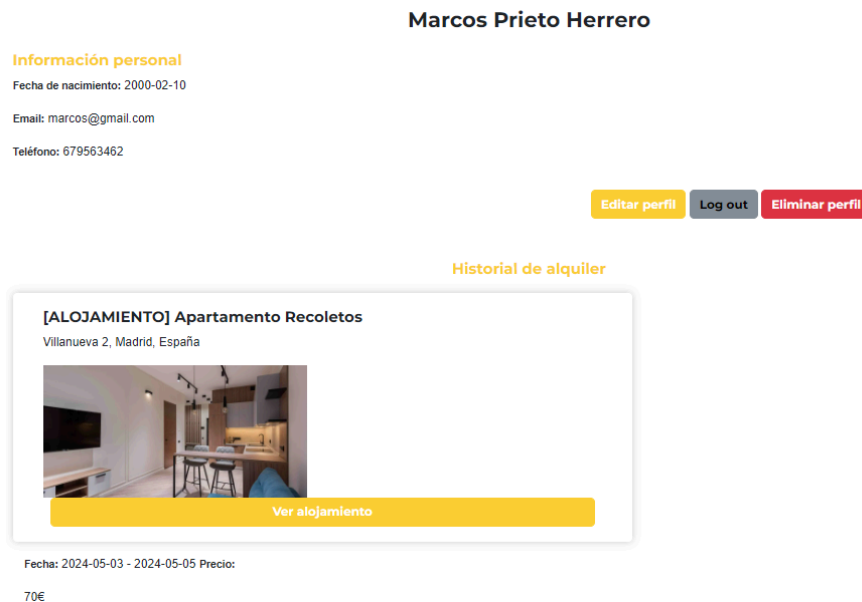


Figura 6.11 Perfil usuario cliente

El código para generar la pantalla anterior se encuentra en la clase "perfilCustomer.php" (Figura 6.12).

```
//Mostrar la información del usuario
echo "
<section id='perfil'>
  <body>
    <div class='row d-flex d-xl-flex justify-content-center justify-content-xl-center'>
      <div class='col-sm-12 col-lg-10 col-xl-9 col-xxl-7 bg-white shadow-lg' style='border-radius: 5px; margin-top:-100px'>
        <div class='p-5'>
          <div class='text-center'>
            <h3 class='user-name'>{$nombre $apellido}</h3>
          </div>
          <div style='margin-top:25px'>
            <h5 class='mt-3 mb-2 text-primary'>Información personal</h5>
            <div class='col-sm-4'>
              <label style='font-size: 14px; display: inline-block; font-weight: bold;'>Fecha de nacimiento: </label>
              <p style='display: inline-block;'>{$cumple}</p>
            </div>
            <div class='col-sm-4'>
              <label style='font-size: 14px; display: inline-block; font-weight: bold;'>Email: </label>
              <p style='display: inline-block;'>{$email}</p>
            </div>
            <div class='col-sm-4'>
              <label style='font-size: 14px; display: inline-block; font-weight: bold;'>Teléfono: </label>
              <p style='display: inline-block;'>{$phone}</p>
            </div>
          </div>
          <div style='margin-top:20px; margin-left:56%'>
            <a type='button' id='submit' name='submit' class='btn btn-primary' href='editarPerfil.php'>Editar perfil</a>
            <a type='button' id='submit' name='submit' class='btn btn-secondary' href='logout.php'>Log out</a>
            <a type='button' id='submit' name='submit' class='btn btn-danger' href='includes/fEliminarPerfil.php'>Eliminar perfil</a>
          </div>
        </div>
      </div>
    </div>
  </body>
</section>";
```

Figura 6.12 Código formulario perfil cliente

Las variables del formulario se han definido antes (Figura 6.13).

```
$idPerfil = $_SESSION["id"];
$user = User::getUserById($idPerfil);

if($user === false) {
    error("errorInterno");
}

$nombre = $user->getName();
$apellido = $user->getSurname();
$cumple = $user->getBirthdate();
$email = $user->getEmail();
$phone = $user->getPhone();
```

Figura 6.13 Variables código formulario perfil cliente

Un usuario registrado como vendedor podrá ver su información personal además de un historial de artículos publicados.


Alojamientos Express

Información personal
NIF: A29268166
Email: alojamiento@express.com
Teléfono: 654321267

[Editar perfil](#) [Publicar artículo](#) [Log out](#) [Eliminar perfil](#)

Mis artículos

Pasarela
Avenida de la Borbolla 11, Sevilla, España



[Mostrar disponibilidad](#) [Editar precios](#) [Editar información](#) [Eliminar](#)

Figura 6.14 Perfil usuario comerciante

El código para generar la pantalla anterior se encuentra en el archivo “perfilSeller.php” (Figura 6.15)

```

id='perfil'>
ly>
<div class='row d-flex d-xl-flex justify-content-center justify-content-xl-center'>
  <div class='col-sm-12 col-lg-11 col-xl-10 col-xxl-8 bg-white shadow-lg' style='border-radius: 5px; margin-top:-150px'>
    <div class='p-5'>
      <div class='text-center'>
        <h3 class='user-name'>${nombre}</h3>
      </div><div style='margin-top:25px'>
        <h6 class='mt-3 mb-2 text-primary'>Información personal</h6>
        <div class='col-sm-4'>
          <label style='font-size: 14px; display: inline-block; font-weight: bold;'>NIF: </label>
          <p style='display: inline-block;'>${nif}</p>
        </div>
        <div class='col-sm-4'>
          <label style='font-size: 14px; display: inline-block; font-weight: bold;'>Email: </label>
          <p style='display: inline-block;'>${email}</p>
        </div>
        <div class='col-sm-4'>
          <label style='font-size: 14px; display: inline-block; font-weight: bold;'>Teléfono: </label>
          <p style='display: inline-block;'>${phone}</p>
        </div>
      </div><div style='margin-top:15px'>
        <a type='button' id='submit' name='submit' class='btn btn-primary btn-user w-45' href='editarPerfil.php'>Editar perfil</a>
        <a type='button' id='submit' name='submit' class='btn btn-primary btn-user w-45' href='publicarArticulo.php'>Publicar articulo</a>
        <a type='button' id='submit' name='submit' class='btn btn-secondary' href='logout.php'>Log out</a>
        <a type='button' id='submit' name='submit' class='btn btn-danger' href='includes/fEliminarPerfil.php'>Eliminar perfil</a>
      </div>;

      echo"<div style='margin-top:33px' >
      <h6 class='mt-3 mb-2 text-primary'>Mis artículos</h6>";

      //Mostrar los articulos publicados por el usuario

      $articulos = array();
      $consulta3 = "SELECT id FROM articulos WHERE idSeller = ?";
      $resultado3 = $db->dbQuery($consulta3, "i", $idPerfil);
      if(!$resultado3){
        error("errorInterno");
      }
      $numArticulos = mysqli_num_rows($resultado3);
      echo"<div class='row-mb-7'>";

```

Figura 6.15 Código perfil comerciante

Las variables del formulario están definidas antes, la información que se muestra se obtiene de la base de datos.

```

$db = getDbInstance();
$consulta = "SELECT * FROM sellers WHERE id = ?";
$resultado = $db->dbQuery($consulta, "i", $idPerfil);

if(!$resultado){
    error("errorInterno");
}

if ($fila = mysqli_fetch_assoc($resultado)) {
    $nombre = $fila["companyName"];
    $nif = $fila["nif"];
}
else {
    error("loginError");
}

$consulta2 = "SELECT * FROM users WHERE id = ?";
$resultado2 = $db->dbQuery($consulta2, "i", $idPerfil);

if(!$resultado2){
    error("errorInterno");
}

if($fila = mysqli_fetch_assoc($resultado2)) {
    $email = $fila["email"];
    $phone = $fila["phone"];
}
else {
    error("loginError");
}

```

Figura 6.16 Variables código formulario perfil comerciante

El usuario puede modificar sus datos pulsando el botón "Editar perfil". Una vez modificada la información que el usuario desee, el formulario llama al archivo "fEditarPerfil.php" que recoge los datos modificados.

```

$id = $_SESSION["id"];

$user = User::getUserById($id);

$correo = isset($_POST["email"]) ? $_POST["email"] : "";
$password1 = isset($_POST["password"]) ? $_POST["password"] : "";
$password2 = isset($_POST["passwordconfirm"]) ? $_POST["passwordconfirm"] : "";
$telefono = isset($_POST["phone"]) ? $_POST["phone"] : "";

if($password1 != "" && $password1 !== $password2){
    error("coincidenciaPassword");
}

$updateCampos = array();
$params = array();

if($user->getEmail() != $correo && $correo !== ""){
    if(!checkEmail($correo)){
        error("emailInvalido");
    }

    if(correoExiste($correo) {
        error("emailExiste");
    }

    $updateCampos[] = "email=?";
    $params[] = $correo;
}

if($password1 !== "") {
    $passwordHash = hash("sha256", $password1);

    $updateCampos[] = "password=?";
    $params[] = $passwordHash;
}

if($user->getPhone() != $telefono && $telefono !== "") {

    if(telefonoExiste($telefono) {
        error("telefonoExiste");
    }

    if(strlen($telefono) !== 9) {
        error("telefonoInvalido");
    }

    $updateCampos[] = "phone=?";
    $params[] = $telefono;
}

if(count($updateCampos) != 0) {
    if(!$user->editarUser("users", $updateCampos, $params)){
        error("errorInterno");
    }
}

```

Figura 6.17 Código editar perfil

```

$updateCampos = array();
$params = array();
$tabla = "";

if($_SESSION["rol"] == "seller"){
    $tabla = "sellers";
    $nombreEmpresa = isset($_POST["companyName"]) ? $_POST["companyName"] : "";
    $nif = isset($_POST["nif"]) ? $_POST["nif"] : "";

    if($user->getCompanyName() != $nombreEmpresa && $nombreEmpresa != "") {
        $updateCampos[] = "companyName=?";
        $params[] = $nombreEmpresa;
    }

    if($user->getNif() != $nif && $nif != "") {
        $updateCampos[] = "nif=?";
        $params[] = $nif;

        if(nifExiste($nif)) {
            error("nifExiste");
        }
    }
}
else {
    $tabla = "customers";
    $nombre = isset($_POST["name"]) ? $_POST["name"] : null;
    $apellido = isset($_POST["surname"]) ? $_POST["surname"] : null;
    $cumple = isset($_POST["birthDate"]) ? $_POST["birthDate"] : null;

    if($user->getName() != $nombre && $nombre != "") {
        $updateCampos[] = "name=?";
        $params[] = $nombre;
    }

    if($user->getSurname() != $apellido && $apellido != ""){
        $updateCampos[] = "surname=?";
        $params[] = $apellido;
    }

    if($user->getBirthDate() != $cumple && $cumple != ""){
        $updateCampos[] = "birthDate=?";
        $params[] = $cumple;
    }
}

if(count($updateCampos) != 0) {
    if(!$user->editarUser($tabla, $updateCampos, $params)){
        error("errorInterno");
    }
}
}

```

Figura 6.18 Código editar perfil seller y customer

A su vez esta clase llama a la función "editarUser" de la clase "User.php" que hace lo siguiente:

```
public static function editarUser($tabla, $updateCampos, $params){
    if(!empty($updateCampos)){
        require_once $_SERVER['DOCUMENT_ROOT']."/includes/funcionesAuxiliares.php";

        $db = getDbInstance();
        $campos = implode(', ', $updateCampos);

        $consulta = "UPDATE $tabla SET $campos WHERE id = ?";

        if (!empty($params)) {
            $tipos = str_repeat('s', count($params));
        }

        $tipos = $tipos."i";
        $params[] = $_SESSION["id"];

        $resultado = $db->dbQuery($consulta, $tipos, ...$params);

        if($resultado === false){
            return false;
        }

        return true;
    }
    else{
        return false;
    }
}
```

Figura 6.19 Código editar user

Eliminar cuenta de usuario

Un usuario puede eliminar su cuenta desde su perfil haciendo click en el botón "Eliminar perfil". Después se muestra la pantalla principal de la página. El código que implementa esta funcionalidad está en la clase "User.php" y es el siguiente:

```

public static function borrarUser(){
    require_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    $db->startTransaction();

    $query = "UPDATE users SET estadoUser = 'eliminado' WHERE id = ?";
    $resultado = $db->dbQuery($query, "i", $_SESSION["id"]);

    if(!$resultado){
        $db->rollbackTransaction();
        return false;
    }

    $db->commitTransaction();
    return true;
}

```

Figura 6.20 Código eliminar user

6.2.3 Módulo alquiler

Alquilar

Un usuario logueado, tras haber seleccionado las fechas deseadas de entrada y salida debe pulsar el botón 'Alquilar' en la opción de habitación que desee.

Alquila este alojamiento

Selecciona las fechas de tu estancia

Fecha de entrada: 09/05/2024

Fecha de salida: 18/05/2024

Opciones de habitación

-Suite

Camas

- Cama Doble extra: 1

Alquilar

Desde
150€

Figura 6.21 Vista alquilar

Una vez pulsado el botón se llama a la función *fAlquilar.php*. Primero se comprueba que los campos existan en la petición POST, que el usuario esté loggeado y que la habitación y alojamiento existan.

```

if($alojamiento->getEstado() != "publicado"){
    error("noExiste");
}

if($alojamiento->getIdSeller() == $idCliente){
    error("noSeller");
}

$fechaInicio = $_POST['fechaInicio'];
$fechaFin = $_POST['fechaFin'];

if(empty($fechaInicio) || empty($fechaFin)){
    error("camposVacios");
}

if($fechaInicio >= $fechaFin){
    error("fechaIncorrecta");
}

//comprobar que la fecha de inicio no sea anterior a la fecha actual
if($fechaInicio < date("Y-m-d")){
    error("fechaPasada");
}

$disponibilidad = $habitacion->comprobarDisponibilidad($fechaInicio, $fechaFin);

if(!$disponibilidad){
    error("noDisponible");
}

$precioTotal = $habitacion->calcularPrecio($fechaInicio, $fechaFin);

if($precioTotal == 0.0){
    error("errorInterno");
}

if(!$alojamiento->alquilarHabitacion($idHabitacion, $idCliente, $fechaInicio, $fechaFin, $precioTotal)){
    error("errorInterno");
}

success("successAlquiler");

header("Location: /perfil.php");
exit();

```

Figura 6.22 Código formulario alquiler

Después, se llama al método `comprobarDisponibilidad()` de la clase `Habitación` y se pasa a calcular el precio en función de las fechas y las ofertas en el caso de que las hubiera.

```

public function comprobarDisponibilidad($fechaInicio, $fechaFin){
    include_once $_SERVER['DOCUMENT_ROOT']."/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    $query = "SELECT COUNT(*) AS cantidad FROM alquileres WHERE idHabitacion = ? AND ((fechaInicio BETWEEN ? AND ?)
    | OR (fechaFin BETWEEN ? AND ?)) GROUP BY idHabitacion;";
    $result = $db->dbQuery($query, "issss", $this->getIdHabitacion(), $fechaInicio, $fechaFin, $fechaInicio, $fechaFin);

    if($result === false){
        //ERROR
        return false;
    }

    //No hay ningun alquiler de la habitacion en esas fechas
    if($result->num_rows == 0){
        return true;
    }

    $cantidad = $result->fetch_assoc()["cantidad"];

    if($cantidad <= 0){
        return false;
    }

    //Comprobar el máximo de habitaciones
    $query = "SELECT numHabitaciones FROM habitaciones WHERE idHabitacion = ?;";
    $result = $db->dbQuery($query, "i", $this->getIdHabitacion());

    if($result === false){
        //ERROR
        return false;
    }

    if($result->num_rows <= 0){
        return false;
    }

    $numHabitaciones = $result->fetch_assoc()["numHabitaciones"];

    if($cantidad >= $numHabitaciones){
        return false;
    }

    return true;
}

```

Figura 6.23 Código comprobar disponibilidad

```

$precioTotal = 0.0;

$fila = $result->fetch_assoc();

$tipoPrecio = $fila["tipoPrecio"];

$fechaActual = $fechaInicio;

while($fechaActual <= $fechaFin){

    //Comprobar si hay una oferta
    $query = "SELECT * FROM ofertas WHERE idHabitacion = ? AND fechaInicio <= ? AND fechaFin >= ?";
    $result = $db->dbQuery($query, "iss", $this->getIdHabitacion(), $fechaActual, $fechaActual);

    if($result === false){
        return false;
    }

    if($result->num_rows > 0){
        $fila = $result->fetch_assoc();
        $precioTotal += $fila["precio"];
        continue;
    }
    else if($tipoPrecio == "precioUnico"){
        $precioTotal += $fila["precioUnico"];
    }
    else{

        //Devuelve un número de 1 a 7, siendo 1 lunes y 7 domingo
        $diaSemana = date("N", $fechaActual);
        switch($diaSemana){
            case 1:
                $precioTotal += $fila["precioLunes"];
                break;
            case 2:
                $precioTotal += $fila["precioMartes"];
                break;
            case 3:
                $precioTotal += $fila["precioMiercoles"];
                break;
            case 4:
                $precioTotal += $fila["precioJueves"];
                break;
            case 5:
                $precioTotal += $fila["precioViernes"];
                break;
            case 6:
                $precioTotal += $fila["precioSabado"];
                break;
            case 7:
                $precioTotal += $fila["precioDomingo"];
                break;
        }
    }

    $fechaActual = date("Y-m-d", strtotime("+1 day", $fechaActual));
}

```

Figura 6.24 Código comprobar disponibilidad 2

Por último se llama al método de Alojamiento `alquilarHabitación()` que hace la consulta para insertar el alquiler en la base de datos.

```

public function alquilarHabitacion($idHabitacion, $idCliente, $fechaInicio, $fechaFin, $precioTotal){
    $db = getDbInstance();

    $query = "INSERT INTO alquileres (idCliente, idHabitacion, fechaInicio, fechaFin, precioTotal) VALUES (?, ?, ?, ?, ?)";

    $result = $db->dbQuery($query, "iissd", $idCliente, $idHabitacion, $fechaInicio, $fechaFin, $precioTotal);

    if($result == false){
        return false;
    }

    return true;
}

```

Figura 6.25 Código alquilar habitación

6.2.4 Módulo Publicar

Publicar un artículo

Un usuario registrado como comerciante, a través del menú principal, de su perfil o de la página de inicio puede publicar un artículo. El proceso consta de 3 pasos: publicar artículo (Figura 6.26), donde se rellenan los datos básicos; publicar habitaciones, donde se dan de alta las opciones de habitación para el alojamiento; y definir los precios.

Publicar artículo

Nombre

Descripción

Dirección

País **Ciudad**

Tipo Alojamiento

Hora Check In **Hora Check Out**

Características del artículo
 Añadir

Subir fotos
 Ningún archivo seleccionado

Continuar **Cancelar**

Figura 6.26 Vista publicar artículo

Además de la información básica se deben seleccionar las características de las que dispone y subir las fotos que se deseen mostrar. Las fotos se almacenan en el servidor dentro de la carpeta "uploads" y en una carpeta con el id del artículo al que pertenecen.

En el caso de que el comerciante tenga un alojamiento a medio publicar, se le redirige directamente al alojamiento que se quedó a medias.

```
$db = getDbInstance();
$query = "SELECT * FROM articulos WHERE idSeller = ? AND (estadoArticulo = 'pendiente1' OR estadoArticulo = 'pendiente2');";
$resultado = $db->dbQuery($query, "i", $_SESSION["id"]);

$articulo = $resultado->fetch_assoc();

if($articulo){
    if($articulo["estadoArticulo"] == "pendiente1"){
        header("location: /publicarAlojamiento.php?id=".$articulo["id"]);
        exit();
    }
    else if($articulo["estadoArticulo"] == "pendiente2"){
        header("location: /precios.php?id=".$articulo["id"]);
        exit();
    }
}
```

Figura 6.27 Código estado formulario publicar

La función encargada de procesar la información para publicar el artículo es fPublicarArticulo. Esta se encarga de comprobar que todos los campos sean correctos, no estén vacíos y los sanitiza. Posteriormente, crea primero el Artículo general y luego el Alojamiento. Todo ello se realiza dentro de una transacción para que en el caso de que haya algún fallo en mitad del proceso se realice un rollback.

```
//Inicializamos la transaccion
$db->startTransaction();

$id = Articulo::publicarArticuloGeneral($idSeller, $tipoArticulo, $nombre, $descripcion, $direccion, $coordenadas["longitud"], $coordenadas["latitud"]);

if(!$id){
    $db->rollbackTransaction();
    error("errorPublicar");
}
```

Figura 6.28 Código publicar artículo

Si todo es correcto y se ha insertado correctamente, pasa al siguiente paso: publicar las habitaciones

Alta de Habitaciones

Habitación 1

Nombre

Número máximo de personas

Número total de habitaciones

Tipo de cama

Número total de camas de ese tipo

Tipo de cama

Número total de camas de ese tipo

Añadir cama

Habitación 2

Nombre

Habitación 2

Nombre

Número máximo de personas

Número total de habitaciones

Tipo de cama

Número total de camas de ese tipo

Añadir cama

Añadir otra habitación

Continuar

Volver

Figura 6.29 Vista publicar alta habitaciones

El usuario puede elegir añadir más habitaciones, y una vez rellenada toda la información, el formulario se envía a fProcesarAlojamiento. Esta función comprueba

que los campos mínimos existan y sean correctos, además de que el alojamiento exista y esté pendiente de ser publicado. Después procede a iterar por la variable POST para ir creando las habitaciones con las camas correspondientes y publicándolas. Todo ello se realiza en una transacción por si ocurriera un error en mitad del proceso.

```
//Inicializamos la transaccion
$db->startTransaction();

try{
    $numHabitacion = 1;
    while(isset($_POST['nombre'].$numHabitacion)){

        $nombre = $_POST['nombre'].$numHabitacion;
        $personasMax = $_POST['personasMax'].$numHabitacion;
        $numHabitaciones = $_POST['numHabitaciones'].$numHabitacion;

        if(empty($nombre) || empty($personasMax) || empty($numHabitaciones)){
            $db->rollbackTransaction();
            error("camposVacios");
        }

        if($personasMax <= 0 || $numHabitaciones <= 0){
            $db->rollbackTransaction();
            error("camposVacios");
        }

        //Publicar habitacion
        $newHabitacion = $alojamiento->publicarHabitacion($nombre, $personasMax, $numHabitaciones);

        if(!$newHabitacion){
            $db->rollbackTransaction();
            error("errorPublicar");
        }

        //Publicar camas
        $numCamas = 1;
        //El identificador de las camas es "tipo"/"cantidad" + idHabitacion + idCama (empezando por 1)
        while(isset($_POST['tipo'].$numHabitacion.$numCamas]) {
```

Figura 6.30 Código alta habitaciones

Si se ha publicado correctamente todas las habitaciones, se pasa a la siguiente fase de definición de precios.

Suite

Selecciona el tipo de precio

Establecer un precio único
 Establecer un precio por día

Precio único

150,99

Figura 6.31 Vista formulario precios

En el formulario, el comerciante puede elegir si poner un precio único o establecer un precio por día de la semana (una vez publicado podrá definir ofertas en periodos específicos). Este formulario es procesado por "fPrecios" que tras comprobar que todos los datos son correctos llama al método "publicarPrecios" de Artículo, que es el encargado de añadir los precios a la base de datos.

```

public function publicarPrecios($tipoPrecio, $idHabitacion, $precioUnico, $precioLunes, $precioMartes, $precioMiercoles, $precioJueves, $
include_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php";
$db = getDbInstance();

$precios = $this->getPrecios($idHabitacion);

if($precios){
    $consulta = "UPDATE precios SET tipoPrecio = ?, precioUnico = ?, precioLunes = ?, precioMartes = ?, precioMiercoles = ?, precioJueves = ?, precioViernes = ?, precioSabado = ?, precioDomingo = ?";
    $resultado = $db->dbQuery($consulta, "siiiiiiii", $tipoPrecio, $precioUnico, $precioLunes, $precioMartes, $precioMiercoles, $precioJueves, $precioViernes, $precioSabado, $precioDomingo);

    if(!$resultado){
        return false;
    }
    return true;
}
else{
    $consulta = "INSERT INTO precios (idHabitacion, tipoPrecio, precioUnico, precioLunes, precioMartes,
        precioMiercoles, precioJueves, precioViernes, precioSabado, precioDomingo) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
    $resultado = $db->dbQuery($consulta, "isiiiiiiii", $idHabitacion, $tipoPrecio, $precioUnico, $precioLunes, $precioMartes, $precioMiercoles, $precioJueves, $precioViernes, $precioSabado, $precioDomingo);

    if($resultado === false){
        return false;
    }
    return true;
}
}
  
```

Figura 6.32 Código formulario precios

Una vez finalizado todo el proceso, el usuario es redirigido a la página del alojamiento que acaba de publicar.

Eliminar un artículo

Un usuario registrado como comerciante que haya publicado algún artículo podrá eliminarlo desde su perfil pulsando sobre el botón “Eliminar” de dicho artículo. Este formulario llama a la función “fEliminarArticulo.php” que gestiona todos los errores y llama a su vez a la función “eliminarArticulo” de la clase “Articulo.php”.

```
if(isset($_SESSION["id"])) {
    $idArticulo = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);
    $idPublicador = $_SESSION["id"];

    $db = getDbInstance();

    $articulo = Articulo::getArticuloById($idArticulo);

    if(!$articulo) {
        error("errorInterno");
    }

    if(!$articulo->eliminarArticulo()){
        error("errorInterno");
    }

    success("articuloEliminado");

    header("location: ../perfil.php?id=".$idPublicador);
}
else{
    error("noLogin");
}
```

Figura 6.33 Código llamada a eliminar artículo

La función “eliminarArticulo” se ha implementado de la siguiente manera:

```

public function eliminarArticulo() {
    include_once $_SERVER['DOCUMENT_ROOT']."/includes/funcionesAuxiliares.php";

    $db = getDbInstance();

    $query = 'UPDATE articulos SET estadoArticulo = "eliminado" WHERE id = ?';
    $result = $db->dbQuery($query, "i", $this->getId());
    if(!$result) {
        return false;
    }

    return true;
}

```

Figura 6.34 Código eliminar artículo

Modificar datos artículo

Un usuario puede modificar los datos de los artículos que haya publicado. En esta página aparecen tres formularios. El primero es para editar los datos del alojamiento. El formulario que el sistema muestra es el siguiente:

Modificar datos

Nombre alojamiento	Dirección
<input type="text" value="Pasarela"/>	<input type="text" value="Avenida de la Borbolla 11, Sevilla, España"/>
Ciudad	País
<input type="text" value="Sevilla"/>	<input type="text" value="España"/>
Descripción	
<input type="text" value="Este hotel sevillano tiene vistas al parque de Maria Luisa. Esta a solo unos minutos a pie de los impresionantes monumentos cercanos a la Catedral."/>	
Hora check in	Hora check out
<input type="text" value="12:00"/>	<input type="text" value="13:00"/>
Tipo Alojamiento	
<input type="text" value="Hotel"/>	

Figura 6.35 Vista editar artículo

La implementación de este formulario está en la clase "editarArticulo.php" que llama a la función "fEditarArticulo.php" donde se recoge toda la información introducida por el usuario.

```

$db->startTransaction();

//Actualiza los datos generales del articulo
$nombre = isset($_POST["nombre"]) ? $_POST["nombre"] : null;
$descripcion = isset($_POST["descripcion"]) ? $_POST["descripcion"] : null;
$direccion = isset($_POST["direccion"]) ? $_POST["direccion"] : null;
$ciudad = isset($_POST["ciudad"]) ? $_POST["ciudad"] : null;
$pais = isset($_POST["pais"]) ? $_POST["pais"] : null;

//En caso de que algún campo quede en blanco no se modifica y se deja el valor anterior
if($nombre !== null) {
    $alojamiento->setNombre(sanitize($nombre));
}

if($descripcion !== null) {
    $alojamiento->setDescripcion(sanitize($descripcion));
}

if($direccion !== null) {
    if($alojamiento->getDireccion() !== $direccion){
        $alojamiento->setDireccion(sanitize($direccion));

        if($ciudad !== null) {
            $alojamiento->setCiudad(sanitize($ciudad));
        }

        if($pais !== null) {
            $alojamiento->setPais(sanitize($pais));
        }

        if(strpos($direccion, $ciudad) === false) {
            $alojamiento->setDireccion($direccion . ", " . $ciudad);
        }

        if(strpos($direccion, $pais) === false) {
            $alojamiento->setDireccion($direccion . ", " . $pais);
        }

        $coordenadas = getCoordenadasDireccion($direccion);

        if($coordenadas) {
            $alojamiento->setLatitud($coordenadas["lat"]);
            $alojamiento->setLongitud($coordenadas["lon"]);
        }
        else {
            $db->rollbackTransaction();
            error("errorDireccion");
        }
    }
}

$result = $alojamiento->editarArticulo();

```

Figura 6.36 Código formulario editar artículo

Desde esta clase se llama a la función "editarArticulo" de la clase "Articulo.php" que actualiza la tabla "articulos".

```
public function editarArticulo() {
    include_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php';

    $db = getDbInstance();

    $query = 'UPDATE articulos SET nombre = ?, descripcion = ?, direccion = ?, latitud = ?, longitud = ?, ciudad = ?, pais = ? WHERE id = ?';
    $result = $db->dbQuery($query, "sssssssi", $this->getNombre(), $this->getDescripcion(), $this->getDireccion(), $this->getLatitud(), $this->getLongitud(), $this->getCiudad(), $this->getPais(), $this->getId());
    if(!$result) {
        return false;
    }
    return true;
}
```

Figura 6.37 Código editar artículo

El segundo formulario se usa para modificar las habitaciones.

Modificar Habitaciones

Habitacion principal

Nombre de la habitación

Número máximo de personas

Número de habitaciones

Figura 6.38 Código editar habitaciones artículo

La información se recoge desde la función "fEditarArticulo.php" como en el caso anterior.

```

$habitaciones = $alojamiento->habitaciones();

if(!$habitaciones) {
    error("errorInterno");
}

foreach($habitaciones as $habitacion) {
    $idHabitacion = $habitacion->getIdHabitacion();
    $nombreHabitacion = isset($_POST["nombreHabitacion".$idHabitacion]) ? $_POST["nombreHabitacion".$idHabitacion] : null;
    $personasMax = isset($_POST["personasMax".$idHabitacion]) ? $_POST["personasMax".$idHabitacion] : null;
    $numHabitaciones = isset($_POST["numHabitaciones".$idHabitacion]) ? $_POST["numHabitaciones".$idHabitacion] : null;

    if($nombreHabitacion !== null) {
        $habitacion->setNombre(sanitize($nombreHabitacion));
    }

    if($personasMax !== null) {
        $habitacion->setPersonasMax(sanitize($personasMax));
    }

    if($numHabitaciones !== null) {
        $habitacion->setNumHabitaciones(sanitize($numHabitaciones));
    }

    $result = $habitacion->editarHabitacion();
}

```

Figura 6.39 Código formulario editar habitación

Se llama a la función “editarHabitacion” de la clase “Habitacion.php” que actualiza la información de la base de datos.

```

public function editarHabitacion(){
    include_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php';
    $db = getDbInstance();
    $query = "UPDATE habitaciones SET nombre = ?, personasMax = ?, numHabitaciones = ? WHERE idHabitacion = ?";
    $result = $db->dbQuery($query, "siii", $this->getNombre(), $this->getPersonasMax(), $this->getNumHabitaciones(), $this->getIdHabitacion());
    return $result;
}

```

Figura 6.40 Código editar habitación

Por último, desde el tercer formulario se puede modificar la información relacionada con las camas.

Modificar camas

Tipo de cama	Número de camas
<input type="text" value="Sofá cama"/>	<input type="text" value="1"/>
Tipo de cama	Número de camas
<input type="text" value="Sofá cama"/>	<input type="text" value="1"/>

Figura 6.41 Formulario modificar camas

Como en los otros dos casos la información se recoge desde la función “fEditarArticulo.php”.

```
foreach($habitacion->camas() as $cama) {
    $idCama = $cama["idCama"];

    $numCamas = isset($_POST["numCamas".$idCama]) ? $_POST["numCamas".$idCama] : null;

    $tipoCama = isset($_POST["tipoCama".$idCama]) ? $_POST["tipoCama".$idCama] : null;

    if(!checkTipoCama($tipoCama)) {
        $db->rollbackTransaction();
        error("errorTipoCama");
    }

    if($numCamas !== null) {
        $result = $habitacion->editarCama(sanitize($idCama), sanitize($tipoCama), sanitize($numCamas));

        if(!$result) {
            $db->rollbackTransaction();
            error("errorInterno");
        }
    }
}
```

Figura 6.42 Código formulario editar camas

Como se muestra en el código, se llama a la función "editarCama" de la clase "Habitacion.php" para actualizar la base de datos.

```
public function editarCama($idCama,$tipoCama, $cantidad){
    include_once $_SERVER['DOCUMENT_ROOT']."/includes/funcionesAuxiliares.php";
    $db = getDbInstance();
    $query = "UPDATE camas SET tipoCama = ?, cantidad = ? WHERE idCama = ?";
    $result = $db->dbQuery($query, "sii", $tipoCama, $cantidad, $idCama);
    return $result;
}
```

Figura 6.43 Código editar cama

Modificar precios

Un usuario registrado como vendedor puede modificar el precio de un artículo que haya publicado. Simplemente tendrá que pulsar sobre el botón "Editar precios" que aparece en cualquiera de los artículos. Se mostrará un formulario donde se podrá establecer un precio único para el alojamiento o uno distinto por día. Este artículo llama a la función "fPrecios.php" (Figura 6.44) la cual recoge los datos introducidos por el usuario.

```

//Recorrer todas las habitaciones y en funcion del tipo de precio, publicar los precios
while(isset($_POST['tipoPrecio'.$i])){

    if($_POST['tipoPrecio'.$i] == "precioUnico"){
        $tipoPrecio = "precioUnico";

        if(!isset($_POST['precioUnico'.$i])){
            error("camposVacios");
        }

        $precioUnico = $_POST['precioUnico'.$i];

        if($precioUnico <= 0){
            error("precioNegativo");
        }

        $precioLunes = 0;
        $precioMartes = 0;
        $precioMiercoles = 0;
        $precioJueves = 0;
        $precioViernes = 0;
        $precioSabado = 0;
        $precioDomingo = 0;
    }
    else if($_POST['tipoPrecio'.$i] == "precioDia"){
        $tipoPrecio = "precioDia";
    }
}

```

Figura 6.44 Código formulario precios

Como se ve esta función llama a la función "publicarPrecios" (Figura 6.44) de la clase "Articulo.php" la cual modifica en la base de datos los datos.

```

public function publicarPrecios($tipoPrecio, $idHabitacion, $precioUnico, $precioLunes, $precioMartes, $precioMiercoles, $precioJueves,
include_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php";
$db = getDbInstance();

$precios = $this->getPrecios($idHabitacion);

if($precios){
    $consulta = "UPDATE precios SET tipoPrecio = ?, precioUnico = ?, precioLunes = ?, precioMartes = ?,
precioMiercoles = ?, precioJueves = ?, precioViernes = ?, precioSabado = ?,
precioDomingo = ? WHERE idHabitacion = ?";
$resultado = $db->dbQuery($consulta, "siiiiiiii", $tipoPrecio, $precioUnico, $precioLunes, $precioMartes, $precioMiercoles, $precioJueves, $precioViernes, $precioSabado, $precioDomingo);

    if(!$resultado){
        return false;
    }
    return true;
}
else{
    $consulta = "INSERT INTO precios (idHabitacion, tipoPrecio, precioUnico, precioLunes, precioMartes,
precioMiercoles, precioJueves, precioViernes, precioSabado, precioDomingo) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

    $resultado = $db->dbQuery($consulta, "isiiiiiiii", $idHabitacion, $tipoPrecio, $precioUnico, $precioLunes, $precioMartes, $precioMiercoles, $precioJueves, $precioViernes, $precioSabado, $precioDomingo);

    if($resultado === false){
        return false;
    }
    return true;
}
}

```

Figura 6.45 Código publicar precios

Publicar ofertas

Un usuario comerciante puede publicar una oferta para uno de sus alojamientos. Desde el formulario de editar precios aparece un botón de “Publicar oferta” el cual lleva al formulario en concreto (Figura 6.46).

Ofertas actuales de Pasarela

Precio por noche	Fecha de inicio	Fecha de fin	Acciones
No hay ofertas publicadas			
Publicar nueva oferta			

Publicar oferta

Precio por noche:

Fecha de inicio:

Fecha de fin:

Publicar oferta

Figura 6.46 Vista publicar oferta

Una vez introducida la información correspondiente, se llama a la función "fOfertas.php" que inserta en la base de datos la oferta en cuestión.

```

if($fechaInicio > $fechaInicioExistente && $fechaFin < $fechaFinExistente){
$consulta1 = "INSERT INTO ofertas (idHabitacion, precio, fechaInicio, fechaFin) VALUES (?, ?, ?, ?)";
$fechaInicioDate = new DateTime($fechaInicio);
$fechaInicioDate->modify('-1 day');
$fechaInicioDate = $fechaInicioDate->format('Y-m-d');
$resultado1 = $db->dbQuery($consulta1, "idss", $idHabitacion, $row["precio"], $fechaInicioExistente, $fechaInicioDate);

if($resultado1 === false){
$db->rollbackTransaction();
error("errorInterno");
}

$consulta2 = "INSERT INTO ofertas (idHabitacion, precio, fechaInicio, fechaFin) VALUES (?, ?, ?, ?)";
$fechaFinDate = new DateTime($fechaFin);
$fechaFinDate->modify('+1 day');
$fechaFinDate = $fechaFinDate->format('Y-m-d');
$resultado2 = $db->dbQuery($consulta2, "idss", $idHabitacion, $row["precio"], $fechaFinDate, $fechaFinExistente);

if($resultado2 === false){
$db->rollbackTransaction();
error("errorInterno");
}

$consulta3 = "DELETE FROM ofertas WHERE id = ?";
$resultado3 = $db->dbQuery($consulta3, "i", $row['id']);

if($resultado3 === false){
$db->rollbackTransaction();
error("errorInterno");
}

```

Figura 6.47 Código publicar ofertas

Modificar ofertas

Un usuario comerciante que haya publicado una oferta para un artículo puede modificarla pulsando sobre el botón "Modificar" de la oferta en cuestión. La vista es la siguiente (Figura 6.48)

Ofertas actuales de Pasarela

Precio por noche	Fecha de inicio	Fecha de fin	Acciones	
23€	11-05-2024	14-05-2024	Modificar	Eliminar
Publicar nueva oferta				

Publicar oferta

Precio por noche:

Fecha de inicio:

Fecha de fin:

Figura 6.48 Vista modificar oferta

La funcionalidad de esta acción se encuentra en el archivo "fOfertas.php". La implementación se encuentra en la Figura 6.49

```
if($fechaInicio <= $fechaInicioExistente && $fechaFin < $fechaFinExistente){
    $consulta = "UPDATE ofertas SET fechaInicio = ?, fechaFin = ? WHERE id = ?";
    $fechaInicioDate = new DateTime($fechaFin);
    $fechaInicioDate->modify('+1 day');
    $fechaInicioDate = $fechaInicioDate->format('Y-m-d');
    $resultado = $db->dbQuery($consulta, "ssi", $fechaInicioDate, $fechaFinExistente, $row['id']);

    if($resultado === false){
        $db->rollbackTransaction();
        error("errorInterno");
    }
}
```

Figura 6.49 Código modificar oferta

Eliminar ofertas

Un usuario comerciante que haya publicado una oferta para un artículo puede eliminarla pulsando sobre el botón "Eliminar" de la oferta en cuestión. La vista es la siguiente:

Ofertas actuales de Pasarela

Precio por noche	Fecha de inicio	Fecha de fin	Acciones	
23€	11-05-2024	14-05-2024	Modificar	Eliminar

[Publicar nueva oferta](#)

Publicar oferta

Precio por noche:

Fecha de inicio:

Fecha de fin:

Publicar oferta

Figura 6.50 Vista eliminar oferta

Su funcionalidad está implementada en la clase "fEliminarOferta.php" que se muestra a continuación.

```

if(!isset($_GET['idOferta'])){
    error("camposVacios");
}

if(!isset($_GET['id'])){
    error("camposVacios");
}

if(!isset($_SESSION['id'])){
    error("noLogin");
}

$idOferta = filter_input(INPUT_GET, 'idOferta', FILTER_SANITIZE_NUMBER_INT);
$idHabitacion = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);

$db = getDbInstance();

#Delete oferta
$consulta = "DELETE FROM ofertas WHERE id = ?;";
$resultado = $db->dbQuery($consulta, "i", $idOferta);

if($resultado === false){
    error("errorInterno");
}

success("successOferta");

header("Location: ../ofertas.php?id=$idHabitacion");
exit();

```

Figura 6.51 Código eliminar oferta

Ver reservas

Un usuario comerciante que haya publicado un artículo podrá ver los días reservados para dicho alojamiento pulsando sobre el botón "Ver reservas". La vista es la siguiente:

[Volver](#)

Reservas de Apartamento Recoletos

Id Reserva	Nombre habitación	Id cliente	Precio	Fecha inicio	Fecha fin	Pasada
6	5	70	Apartamento Recoletos	20-05-2024	26-05-2024	No
5	5	70	Apartamento Recoletos	03-05-2024	05-05-2024	Sí

Figura 6.52 Vista ver reservas

Esta pantalla se ha implementado en el archivo "disponibilidad.php" (Figura 6.53 y Figura 6.54).

```

foreach($habitaciones as $habitacion){
    $reservas = $habitacion->getReservas();
    foreach($reservas as $reserva){
        $idHabitacion = $reserva["idHabitacion"];
        $habitacion = Habitacion::getHabitacionById($idHabitacion);
        if($habitacion == false){
            continue;
        }

        //Si la reserva es para una fecha pasada, se colorea la fila de gris
        if($reserva["fechaFin"] < date("Y-m-d")){
            echo '<td style="background-color: #ffe7e7a;">'. $reserva["id"]. '</td>';
            echo '<td style="background-color: #ffe7e7a;">'. $reserva["idCliente"]. '</td>';
            echo '<td style="background-color: #ffe7e7a;">'. $reserva["precioTotal"]. '</td>';
            echo '<td style="background-color: #ffe7e7a;">'. $habitacion->getNombre(). '</td>';
            echo '<td style="background-color: #ffe7e7a;">'. date("d-m-Y", strtotime($reserva["fechaInicio"])). '</td>';
            echo '<td style="background-color: #ffe7e7a;">'. date("d-m-Y", strtotime($reserva["fechaFin"])). '</td>';
            echo '<td style="background-color: #ffe7e7a;">Sí</td>';
        } else {
            echo '<tr>';
        }
        echo '<td>'. $reserva["id"]. '</td>';
        echo '<td>'. $reserva["idCliente"]. '</td>';
        echo '<td>'. $reserva["precioTotal"]. '</td>';
        echo '<td>'. $habitacion->getNombre(). '</td>';
        echo '<td>'. date("d-m-Y", strtotime($reserva["fechaInicio"])). '</td>';
        echo '<td>'. date("d-m-Y", strtotime($reserva["fechaFin"])). '</td>';
        echo '<td>No</td>';

        echo '</tr>';
    }
}

```

Figura 6.53 Código ver reservas

```

foreach ($reservas as $reserva) {
    $fechaInicio = new DateTime($reserva['fechaInicio']);
    $fechaFin = new DateTime($reserva['fechaFin']);
    $fechaFin->modify('+1 day');

    //Select the days between the start and end date in the calendar modifying Class('selected') to the divs
    $interval = new DateInterval('P1D');
    $daterange = new DatePeriod($fechaInicio, $interval, $fechaFin);

    foreach ($daterange as $date) {
        $day = $date->format('j');
        $month = $date->format('n');

        //Modificar la clase de los días que estén en la reserva
        echo "<script>
            if ($('.day[data-day=$day][data-month=$month]').length > 0) {
                $('.day[data-day=$day][data-month=$month]').addClass('selected');
            } else {
            }</script>";
    }
}

```

Figura 6.54 Código ver reservas 2

Para marcar los días reservados, se llama a la función "getReservas()" de la clase "Habitacion.php" la cual recorre la tabla "alquileres" y devuelve un array con los días reservados para esa habitación. Se puede ver en la siguiente figura Figura 6.55.

```

public function getReservas($fechaInicio = "", $fechaFin = ""){
    include_once $_SERVER['DOCUMENT_ROOT']."/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    //Si las fechas están vacías, se obtienen todas las reservas
    if(empty($fechaInicio) || empty($fechaFin)){
        $query = "SELECT * FROM alquileres WHERE idHabitacion = ? ORDER BY fechaInicio DESC;";
        $result = $db->dbQuery($query, "i", $this->getIdHabitacion());
        $reservas = array();
        while ($row = $result->fetch_assoc()) {
            $reservas[] = $row;
        }
        return $reservas;
    }

    $query = "SELECT * FROM alquileres WHERE idHabitacion = ? AND ((fechaInicio BETWEEN ? AND ?) OR (fechaFin BETWEEN ? AND ?));";
    $result = $db->dbQuery($query, "issss", $this->getIdHabitacion(), $fechaInicio, $fechaFin, $fechaInicio, $fechaFin);
    $reservas = array();
    while ($row = $result->fetch_assoc()) {
        $reservas[] = $row;
    }
    return $reservas;
}

```

Figura 6.55 Código getReservas()

6.2.5 Módulo Comentarios

Publicar comentario

Si un usuario logueado ha alquilado un alojamiento y el alquiler ya ha pasado, en la página del artículo se le habilita un apartado (Figura 6.56) para que publique un comentario con su opinión.

¡Comparte tu experiencia!

Comentario

Fui con mis amigos de vacaciones y nos encanto el sitio. ¡Repetiremos seguro!

Puntuación

10

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Enviar

Figura 6.56 Vista publicar comentario

Una vez el usuario escribe su comentario y su puntuación, el formulario se envía por POST a "fComentar.php". Esta función primero comprueba que no haya campos vacíos, que se hayan enviado los datos correctos y sanitiza las entradas. Por último llama al método "publicarComentario" de User que es el encargado de hacer la consulta para insertar el comentario en la base de datos. Para comprobar si un usuario puede comentar se utiliza el método "checkComentarioUser()" de la Figura 6.57.

```

echo '<th>Puntuación</th>';
echo '<th>Comentario</th>';
echo '</tr>';
echo '</thead>';
echo '<tbody>';
$comentarios = $alojamiento->getComentarios();
if($comentarios === false){
    $comentarios = array();

    //no hay comentarios
    echo '<tr>';
    echo '<td colspan="3" style="text-align:center">No hay comentarios todavía</td>';

} else{
    foreach($comentarios as $comentario){
        echo '<tr>';
        echo '<td style="text-align:center">'.$comentario["puntuacion"].'</td>';
        echo '<td>'.$comentario["comentario"].'</td>';
        echo '</tr>';
    }
}
}

```

Figura 6.57 Código comprobar comentarios

Ver comentarios

Cualquier usuario que entre a ver un alojamiento podrá ver un apartado con la puntuación media del mismo y los comentarios de otros usuarios acerca de su experiencia como se puede apreciar en la Figura 6.58

Puntuación media

6.50 / 10



Comentarios

Puntuación	Comentario
7	Me encantó. Departamento reformado. Moderno. Cómodo y muy bien ubicado.
6	Todo lo que ofrecen cuando uno reserva se cumple. Parking. Ascensores, Comodidad en el apto. Muy práctico y a muy buen precio para la ubicación.

Figura 6.58 Vista comentario

Para conseguir los comentarios, se utiliza el método de "Artículo" "getComentarios()" de la Figura 6.59, el cual realiza la consulta correspondiente a la base de datos y devuelve un array con los comentarios y las puntuaciones de los usuarios. Estos comentarios se muestran en una tabla por orden de fecha. Además la puntuación se muestra de forma gráfica con estrellas además de con números.

```

public function getComentarios(){
    include_once $_SERVER['DOCUMENT_ROOT'].'/includes/funcionesAuxiliares.php";
    $db = getDbInstance();

    $query = "SELECT * FROM comentarios WHERE idAlojamiento = ?";
    $resultado = $db->dbQuery($query, "i", $this->getId());

    if($resultado === false){
        return false;
    }

    if($resultado->num_rows == 0){
        return false;
    }

    $row = $resultado->fetch_all(MYSQLI_ASSOC);

    $comentarios = array();

    foreach($row as $comentario){
        $comentarios[] = $comentario;
    }

    return $comentarios;
}

```

Figura 6.59 Código ver comentario

6.2.6 Módulo Búsqueda


Búsqueda

Cualquier usuario puede introducir un término en la barra de búsqueda del menú superior y darle a buscar. La página (Figura 6.61) muestra los resultados que coinciden con el término introducido en la ciudad, el país o el nombre. Además de mostrar la información básica de cada alojamiento (nombre, dirección y precio), muestra un mapa interactivo con la ubicación de los resultados de la búsqueda.

Figura 6.60 Barra de búsqueda

Resultados de la búsqueda: España

Hotel Rilton
Calle Alcalá 223, Madrid, España



Desde
249,99€

[Ver alojamiento](#)

Neostars Santander

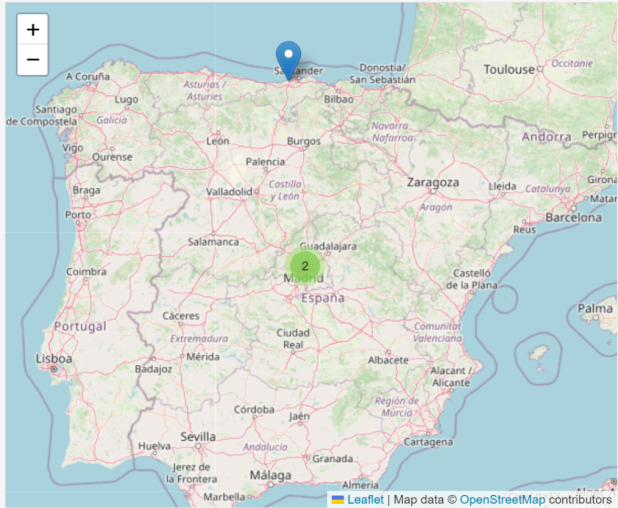


Figura 6.61 Búsqueda

```
//Consulta para obtener los articulos que contengan la palabra clave en el nombre, ciudad o país  
$consulta = "SELECT id FROM articulos WHERE ciudad LIKE ? OR pais LIKE ? OR nombre LIKE ?";  
$resultado = $db->dbQuery($consulta, "sss", "%".$search."%", "%".$search."%", "%".$search."%");
```

Figura 6.62 Código de búsqueda

Antes de mostrar los resultados, se les aplican los filtros introducidos en el caso de que los haya (en el siguiente punto entraremos más en detalle) y se pagan para que haya 10 alojamientos por página (Figura 6.64).

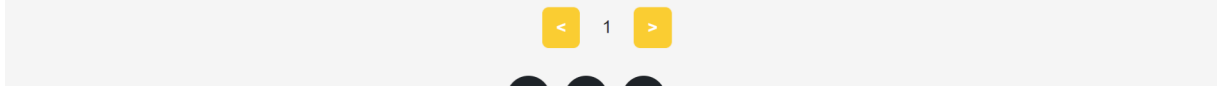


Figura 6.63 Vista paginación búsqueda

```
//Paginación

$perPage = 10;

$offset = ($pagina - 1) * $perPage;
$totalItems = count($articulos);
$totalPages = ceil($totalItems / $perPage);

if($pagina > $totalPages){
    $pagina = $totalPages;
}

$offset = ($pagina - 1) * $perPage;

$articulosSlice = array_slice($articulos, $offset, $perPage);

foreach($articulosSlice as $articulo){
    $articulo->printResumido();
}
```

Figura 6.64 Código paginación

Filtros de búsqueda

El usuario tras hacer la búsqueda puede pulsar en el botón “Mostrar filtros” que despliega un menú con los filtros a aplicar (Figura 6.65). Los filtros permiten seleccionar artículos por precio, puntuación, por tipo de alojamiento y por características.

Filtros de búsqueda

Rango de precios:

1€ - 1000€

Puntuacion media:

0 - 10

Tipo Alojamiento:

- Hotel
- Apartamento
- Hostal
- Resort
- Villa
- Casa rural
- Albergue
- Camping

Características del artículo

Añadir

Figura 6.65 Filtros búsqueda

Los filtros se envían por GET a la propia página de búsqueda que se encargará de filtrar los alojamientos devueltos por la consulta a la base de datos original utilizando la función `array_filter` (función de PHP que filtra los elementos de un array utilizando otra

función de callback para decidir qué elementos mantener. Devuelve un nuevo array con los elementos que pasan la función de callback) que permite filtrar de manera sencilla.

Por cada filtro, se comprueba si se ha aplicado o no, y si se ha aplicado, se filtra el array con todos los artículos para quedarse con los que cumplen los criterios. En la siguiente Figura 6.66 se puede ver el código de filtro por características.

```
//Filtro características
if(isset($_GET["tags"])){
    $tags = filter_input(INPUT_GET, 'tags', FILTER_DEFAULT, FILTER_REQUIRE_ARRAY);
    if(count($tags) > 0){
        $tags = array_map('sanitize', $tags);

        $articulos = array_filter($articulos, function($articulo) use ($tags){
            $caracteristicas = $articulo->caracteristicas();
            if($caracteristicas === false){
                return false;
            }
            foreach($tags as $tag){
                if(!in_array($tag, $caracteristicas)){
                    return false;
                }
            }
            return true;
        });
        echo "<script>document.getElementById('filters').style.display = 'block';document
    }
}
```

Figura 6.66 Código filtrar por características

La ventaja de utilizar GET para filtrar es que la búsqueda puede ser compartida por el usuario, puede ser guardada en los marcadores de su navegador y queda registrada en su historial para poder volver a ella fácilmente.

Capítulo 7 - Conclusiones y trabajo futuro

En este capítulo se explican las conclusiones del proyecto y el trabajo futuro que se realizaría para mejorar la aplicación.

7.1 Conclusiones

El resultado final de este proyecto es una página web que permite a los usuarios tanto publicar como alquilar alojamientos turísticos. Además permite realizar búsquedas a todos aquellos que accedan a la página web para encontrar el alojamiento que más se adecue a sus necesidades. Por otra parte, para que tanto el cliente como el usuario arrendatario pueda conocer la opinión de otros usuarios sobre el alojamiento se ha implementado un módulo de comentarios.

Por otro lado, se ha añadido un historial de reservas para que el cliente pueda comprobar y revisar los detalles del alojamiento que ha alquilado. Además para que el cliente registrado como comerciante pueda conocer el estado de sus alojamientos también dispone de un historial de reservas además de tener en su perfil todos los alojamientos que ha publicado.

En cualquier momento los usuarios registrados pueden modificar los datos de sus cuentas, eliminarlas y el usuario registrado como comerciante puede modificar la información y los precios de los alojamientos que ha publicado.

Por último se ha prestado especial atención al diseño de la interfaz para hacer que esta sea atractiva, intuitiva y fácil de usar y garantizar que la experiencia del usuario sea lo más satisfactoria posible.

7.2 Trabajo futuro

La aplicación cumple gran parte de los objetivos que se detallaron al inicio del curso. Sin embargo, para mejorar y hacer más atractiva la aplicación se podrían añadir distintas funcionalidades como puede ser:

- Añadir nuevos tipos de artículos como podrían ser coches de alquiler o la compra de entradas para distintas actividades, por ejemplo museos o conciertos.
- Para facilitar la comunicación entre el cliente y el comerciante se podría implementar un chat. Sería una mensajería asíncrona en la que el usuario que ha alquilado un artículo, puede ponerse en contacto con el usuario comerciante de dicho artículo. Así mismo el comerciante podrá responder al cliente.
- Se podría añadir un usuario administrador que sería el encargado de aprobar, rechazar o quitar los artículos disponibles en la web. También podrá dar de baja cuentas o eliminar comentarios de algunos artículos. Esto haría que la información que circula en la web sea mucho más segura.
- Además se podría implementar una lista de favoritos para que los usuarios registrados puedan marcar aquellos artículos de la web que le gusten y no perderlos de vista.
- La página web está diseñada en español, para internacionalizar y hacer que cualquiera pueda usar la aplicación independientemente del país o idioma se implementará una opción para traducirla.

Capítulo 8 - Conclusions and future work

This chapter explains the conclusions of the project and the future work that would be done to improve the application.

8.1 Conclusions

The final result of this project is a web page that allows users to both publish and rent tourist accommodations. In addition, it allows searches to all those who access the website to find the accommodation that best suits their needs. On the other hand, so that both the client and the tenant user can know the opinion of other users about the accommodation, a comments section has been implemented.

On the other hand, a booking history has been added so that the client can check and review the details of the accommodation he/she has rented. In addition, so that the customer registered as a trader can know the status of their lodgings, they also have a booking history as well as all the lodgings they have published in their profile.

At any time, registered users can modify their account data, delete them and the user registered as a trader can modify the information and the price of the lodgings he has published.

Finally, special attention has been paid to the design of the interface to make it attractive, intuitive and easy to use and ensure that the user experience is as satisfactory as possible.

8.2 Future work

The application meets most of the objectives detailed at the beginning of the course. However, in order to improve and make the application more attractive, different functionalities can be added, such as:

- Adding new items, as cars or buying tickets for different activities, for instance museums or concerts.
- To facilitate communication between the customer and the trader, a chat could be implemented. It would be an asynchronous messaging in which the user who has rented an item, can contact the user who owns that item. The trader can also respond to the customer.
- Adding an administrator user who would be in charge of approving, rejecting or removing the items available on the website. He/she could also delete accounts or remove comments on some items. This would make the information circulating on the web much safer.
- In addition, a list of favorites will be implemented so that registered users can bookmark those items on the website that they like and not lose sight of them.
- The website is designed in Spanish. In order to internationalize and make the application available to anyone, regardless of the country or language, an option to translate it will be implemented.

Capítulo 9 - Contribuciones Personales

A pesar de haber dividido el trabajo en las diferentes etapas de realización del proyecto, todas las decisiones han sido consensuadas entre ambos miembros del grupo. Ambos miembros hemos hecho un seguimiento del trabajo del otro y nos hemos apoyado si en algún momento uno tenía dificultades. Las pruebas realizadas y la corrección de *bugs* se han realizado entre ambos miembros.

9.1 Sara Cano García

Toma de requisitos:

Durante la toma de requisitos en las primeras semanas, Sara aportó ideas sobre funcionalidades que debería tener el producto final e ideas del diseño como los colores que debería tener. Además sugirió webs como Trivago o AirBnB que sirvieron de inspiración para el estado de arte. Sugirió como tecnología a utilizar Bootstrap para crear el diseño de la parte gráfica de la aplicación web.

En la fase de redacción de los casos de uso se encargó de la creación y redacción de los casos de uso relacionados con el usuario no registrado y el usuario registrado como cliente: Registro, búsqueda, ver artículos, login/logout, alquiler, etc.

Diseño de la interfaz:

Sara encontró la plantilla de estilos en Bootstrap Studio que se ha utilizado para el proyecto. Durante esta fase de diseño también realizó los bocetos de las interfaces correspondientes a los casos de uso redactados en el apartado anterior del proyecto.

A la hora de programar la interfaz, desarrolló el *html* base que se utilizó posteriormente para elaborar todos los formularios. Además diseñó el menú superior que se muestra en todas las páginas.

También ayudó a Álvaro a corregir errores visuales.

Implementación:

Durante la fase de implementación, Sara ha implementado toda la funcionalidad relacionada con la gestión de usuarios: login, logout, registro, modificación y eliminación. Para ello implementó el sistema de clases (User, Seller y Customer) correspondientes para cumplir con el paradigma de programación orientada a objetos. Además implementó el sistema de características para los artículos utilizando Javascript y jQuery para su selección.

También se encargó de la implementación de los precios y su posterior procesamiento, así como la creación y modificación de ofertas y el historial de alquiler.

Gestión de la base de datos:

Sara, en colaboración con Álvaro, se encargó del diseño de las tablas correspondientes a los usuarios, precios y alquileres.

Además como la base de datos no era compartida entre los miembros del equipo, se encargó del mantenimiento de su propia base de datos, actualizando si había algún cambio e introduciendo los datos necesarios para las pruebas.

Redacción de la memoria:

Previo a la redacción de la memoria, ambos miembros del equipo realizaron una lista de apartados y sus contenidos compartida para facilitar su posterior redacción.

Sara ha redactado los siguientes apartados de la memoria: los casos de uso relacionados con el usuario registrado no registrado y el usuario registrado como cliente; la tecnología usada; los índices de contenidos, figuras y tablas; el estado de la cuestión; la implementación de todos los módulos y los apéndices correspondientes a las vistas realizadas por ella.

También se encargó de revisar la traducción de los apartados de la memoria que deben ir en inglés y la revisión ortotipográfica de la memoria.

9.2 Álvaro Laguna Alonso

Toma de requisitos:

Durante la primera parte de toma de requisitos, Álvaro además de aportar ideas sobre funcionalidades que debería tener el producto final, sugirió webs como Booking para tenerlas en cuenta como estado de arte para el proyecto. También investigó sobre qué tecnologías podrían usar. Sugirió tecnologías como PHP o JavaScript para usarlas en la parte de implementación. Respecto a los casos de uso, se encargó de la creación y redacción de los casos de uso relacionados con el usuario registrado como comerciante: Publicar artículos, modificar artículos, eliminar artículos, gestión de los precios de los artículos y la disponibilidad entre otros.

Diseño de la interfaz:

Álvaro se encargó de bocetar las pantallas necesarias para los casos de uso que redactó en el apartado anterior de toma de requisitos. También programó los formularios dinámicos con JavaScript, como por ejemplo el que da de alta las habitaciones de un alojamiento. Además de la parte visual de la función de búsqueda y sus filtros.

También ayudó a Sara en la corrección de errores visuales

Implementación:

Álvaro implementó toda la lógica detrás de los artículos. Creó las clases correspondientes (Artículo, Alojamiento y Habitación), así como las funciones que procesan los formularios que los crean, editan y eliminan.

También se encargó de integrar las APIs de Leaflet y Nominatim para mostrar el mapa con los alojamientos y conseguir las coordenadas de los mismos respectivamente.

Además creó la clase correspondiente a la base de datos (db) utilizando el patrón de diseño Singleton. Esta clase es la encargada de realizar las consultas, gestionar la conexión con la base de datos y gestionar las transacciones.

Por último realizó el sistema de comentarios para los artículos.

Gestión de la base de datos:

Álvaro, en consenso con Sara, se encargó de diseñar las tablas relacionadas con los artículos, alojamientos y características de los mismos.

Además como la base de datos no era compartida entre los miembros del equipo, se encargó del mantenimiento de su propia base de datos, actualizando si había algún cambio e introduciendo los datos necesarios para las pruebas.

Redacción de la memoria:

Previo a la redacción de la memoria, ambos miembros del equipo realizaron una lista de apartados y sus contenidos compartida para facilitar su posterior redacción.

Los apartados redactados por Álvaro son: los casos de uso relacionados con el usuario registrado como comerciante; la descripción de los actores, la arquitectura de la aplicación y modelo de datos; la implementación de las clases; las conclusiones y trabajo futuro; las contribuciones personales; y los apéndices de vistas realizados por él.

También se encargó de redactar la introducción y traducirla al inglés.

BIBLIOGRAFÍA

- [1] Booking.com: <https://www.booking.com>

- [2] Trivago.es: <https://www.trivago.es>

- [3] Airbnb.es: <https://www.airbnb.es>

- [4] XAMPP: <https://www.apachefriends.org/es/index.html>

- [5] Visual Studio Code: <https://code.visualstudio.com>

- [6] GitHub: <https://github.com>

- [7] Xdebug: <https://xdebug.org>

- [8] Bootstrap: <https://getbootstrap.com>

- [9] Bootstrap Studio: <https://bootstrapstudio.io>

- [10] PHP: <https://www.php.net>

- [11] HTML: <https://developer.mozilla.org/es/docs/Web/HTML>

- [12] CSS: <https://developer.mozilla.org/es/docs/Web/CSS>

- [13] JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript>

- [14] jQuery: <https://jquery.com>

- [15] Leaflet: <https://leafletjs.com>

- [16] OpenStreetMap (OSM): <https://www.openstreetmap.org>

[17] Nominatim: <https://nominatim.org>

[18] Suite de Google: <https://workspace.google.com/intl/es>

APÉNDICES

Apéndice A - Guía de uso

En este apéndice se pretende mostrar el funcionamiento de la aplicación.

A.1 Vista de inicio

Al acceder a la aplicación se muestra la pantalla principal accesible para cualquier tipo de usuario. En la cabecera de la pantalla se muestra una barra de búsqueda fija para toda la aplicación además de un menú de opciones las cuales varían dependiendo del tipo de usuario y de si se ha iniciado sesión o no. De primeras las opciones disponibles son “Log in” para los usuarios que tengan creada una cuenta y la opción de “Registrar” para los que no. Además si se pulsa sobre el nombre de la web desde cualquier página se podrá acceder a la vista inicial. La vista es la siguiente:

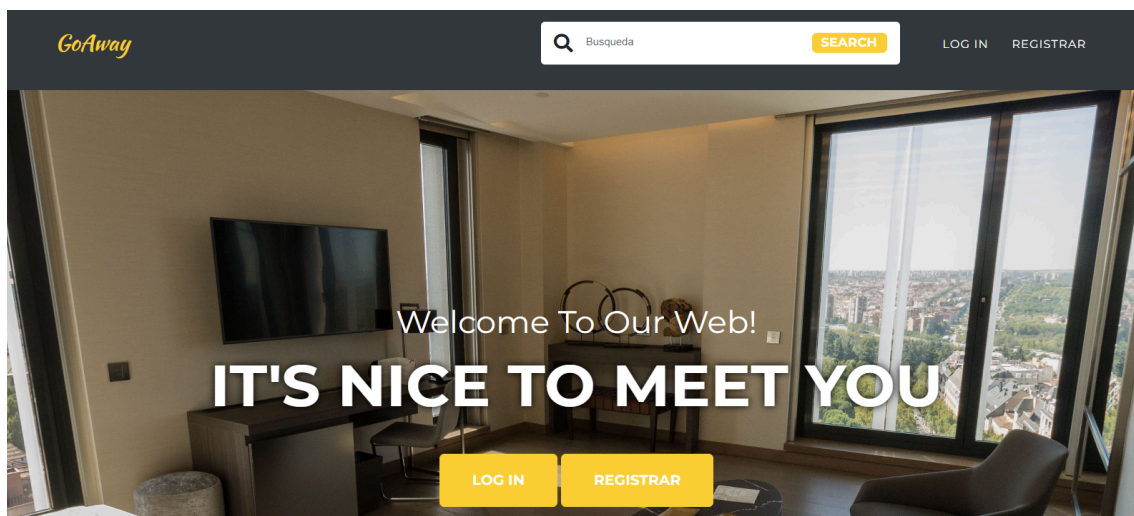


Figura A.1 Vista inicio

Si un usuario comerciante inicia sesión, en la cabecera se muestra una opción para acceder a su perfil, publicar un artículo y cerrar sesión (Figura A.2).

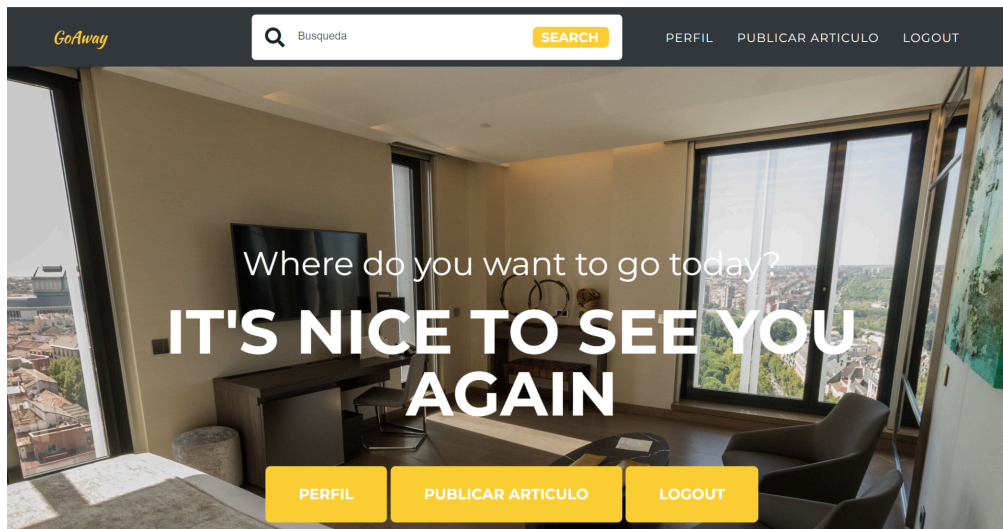


Figura A.2 Vista inicio usuario registrado como comerciante

Las opciones de la cabecera serán diferentes si se trata de un usuario registrado como cliente. Simplemente tendrá la opción de acceder a su perfil y de cerrar sesión.

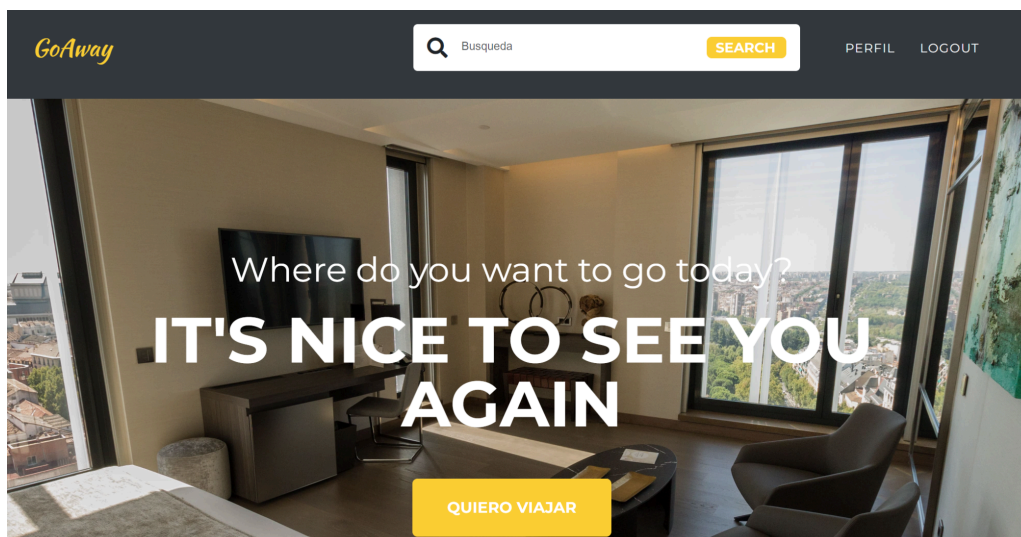


Figura A.3 Vista inicio usuario registrado como cliente

A.2 Vista de registro

Esta vista permite al usuario crear una nueva cuenta en la página web. Para realizar el registro simplemente hay que pulsar la opción "Registrar" que aparece en la parte superior como se especificó en el punto A.1. Para ello se le preguntará por el tipo de cuenta que se quiere crear pudiendo elegir entre usuario vendedor o cliente (Figura A.4). A continuación aparece un formulario a rellenar y por último tiene que pulsar el botón correspondiente. La vista es la siguiente:

¿Cómo se quiere registrar?

Registro como cliente

Registro como vendedor

Figura A.4 Vista registro

Si se pulsa "Registro como cliente" se muestra el formulario de la Figura A.5.

Volver

Crea una cuenta

Nombre	Apellidos
<input type="text" value="Nombre"/>	<input type="text" value="Apellidos"/>
Teléfono	Correo electrónico
<input type="text" value="Teléfono"/>	<input type="text" value="Correo electrónico"/>
Fecha de nacimiento	
<input type="text" value="Fecha de nacimiento"/>	
Contraseña	Repita la contraseña
<input type="text" value="Password"/>	<input type="text" value="Repeat Password"/>

Registrar como cliente

[¿Ya tienes cuenta? Login](#)

Figura A.5 Vista registro como cliente

Si se pulsa “Registro como vendedor” el formulario que aparece es el siguiente:

The screenshot shows the 'Crea una cuenta' (Create an account) registration form for a seller. The form is centered on a white background with a dark header. The header contains the 'GoAway' logo, a search bar with 'Busqueda' and a 'SEARCH' button, and 'LOG IN' and 'REGISTRAR' links. The form itself has a 'Volver' button in the top left. The title 'Crea una cuenta' is centered. Below the title are six input fields arranged in two columns: 'Nombre de la empresa', 'Teléfono', 'Contraseña', 'Correo electrónico', 'NIF', and 'Repita la contraseña'. A large yellow 'Registrar como vendedor' button is at the bottom of the form. Below the button is a link: '¿Ya tienes cuenta? Login'.

Figura A.6 Vista registro como vendedor

A.3 Vista de iniciar sesión

Para acceder a esta vista hay que pulsar el botón de “Login” que aparece en la cabecera de la página. Para iniciar sesión el usuario debe introducir su correo y contraseña. Dicho formulario se muestra en la Figura A.7:

The screenshot shows the 'Log in' (Login) form. The header is identical to the registration form, with the 'GoAway' logo, search bar, and 'LOG IN' and 'REGISTRAR' links. The form is centered on a white background. It has a 'Volver' button in the top left. The title 'Log in' is centered, followed by the instruction 'Introduce tu correo y contraseña'. Below this are two input fields: 'Correo electrónico' and 'Contraseña'. A large yellow 'Login' button is at the bottom of the form.

Figura A.7 Vista iniciar sesión

A.4 Vista de perfil

Una vez iniciada la sesión se puede acceder al perfil desde la opción del menú que aparece en la cabecera de la página web como se muestra en el punto A.1. Dependiendo del tipo de usuario la información a mostrar será diferente. En caso de ser un usuario registrado como cliente se mostrará su información personal y un historial de las reservas que haya realizado. Como se muestra en la Figura A.8.

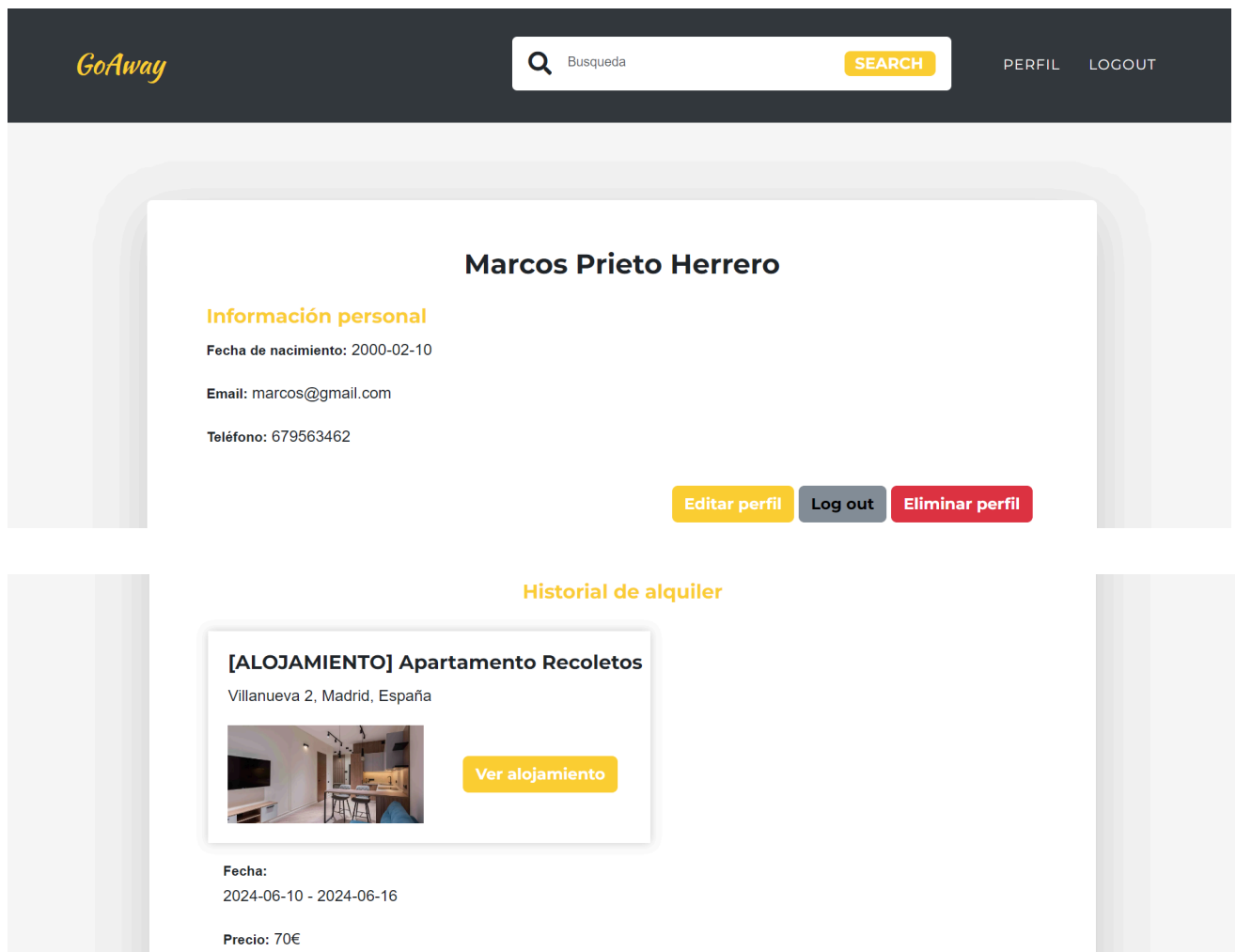


Figura A.8 Vista perfil cliente

Por otro lado, si se trata de un usuario comerciante, también se mostrará su información personal y un historial de artículos publicados como se ve en la Figura A.9.

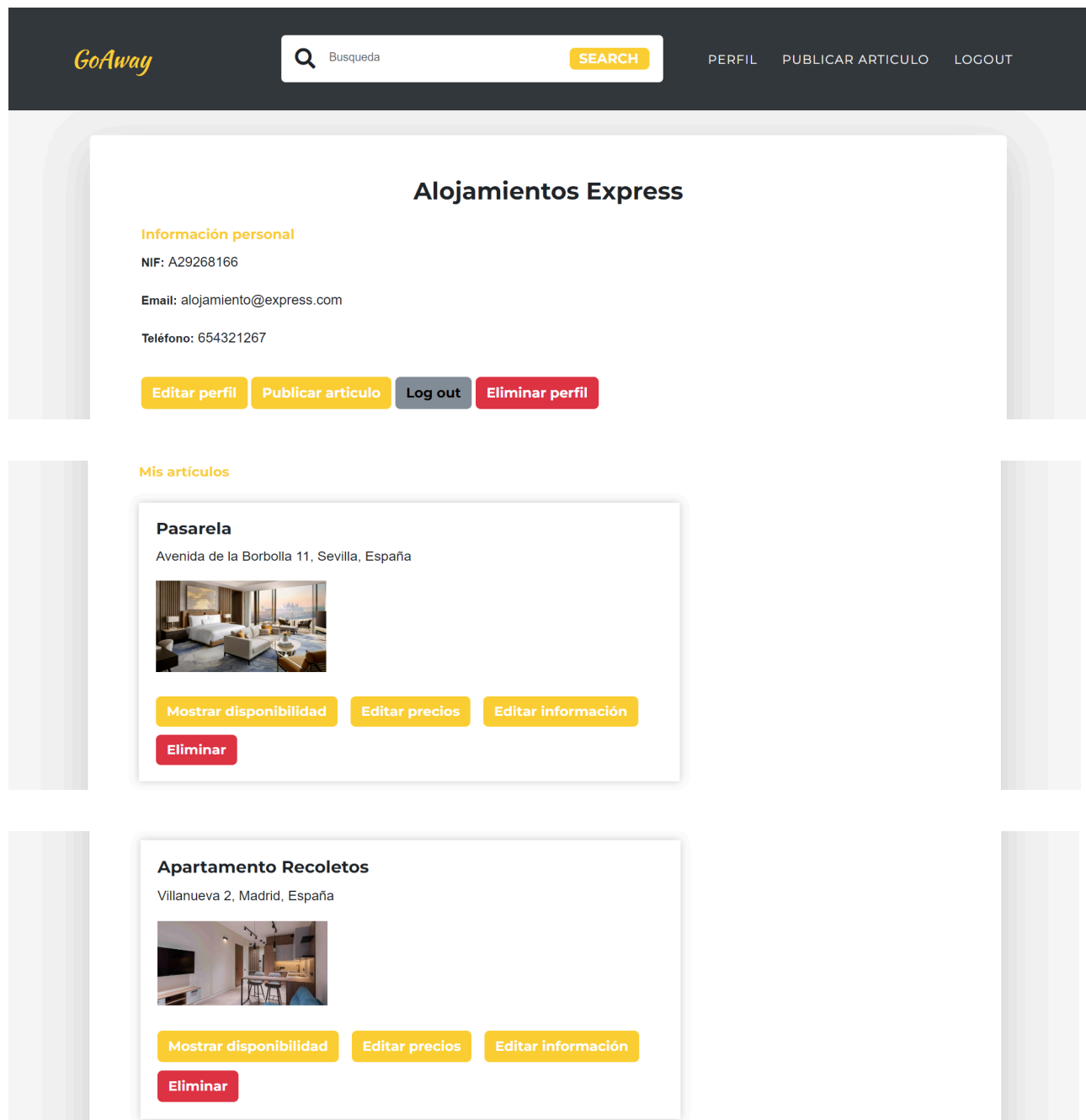


Figura A.9 Vista perfil vendedor

A.5 Vista de editar perfil

Como se indica en la especificación, Capítulo 4.3, para que un usuario modifique sus datos, tiene que iniciar sesión y acceder a su perfil donde aparece una opción para editar sus datos. Dependiendo del tipo de usuario la información a modificar será distinta. Para un usuario vendedor el formulario se muestra en la Figura A.10.

Modificar datos

Nombre empresa <input type="text" value="Alojamientos Express"/>	NIF <input type="text" value="A29268166"/>
Telefono <input type="text" value="654321267"/>	Email <input type="text" value="alojamiento@express.com"/>
Contraseña <input type="text" value="Contraseña"/>	Repite tu contraseña <input type="text" value="Repite tu contraseña"/>

Figura A.10 Vista editar perfil usuario vendedor

Para un usuario registrado como cliente la vista es la siguiente:

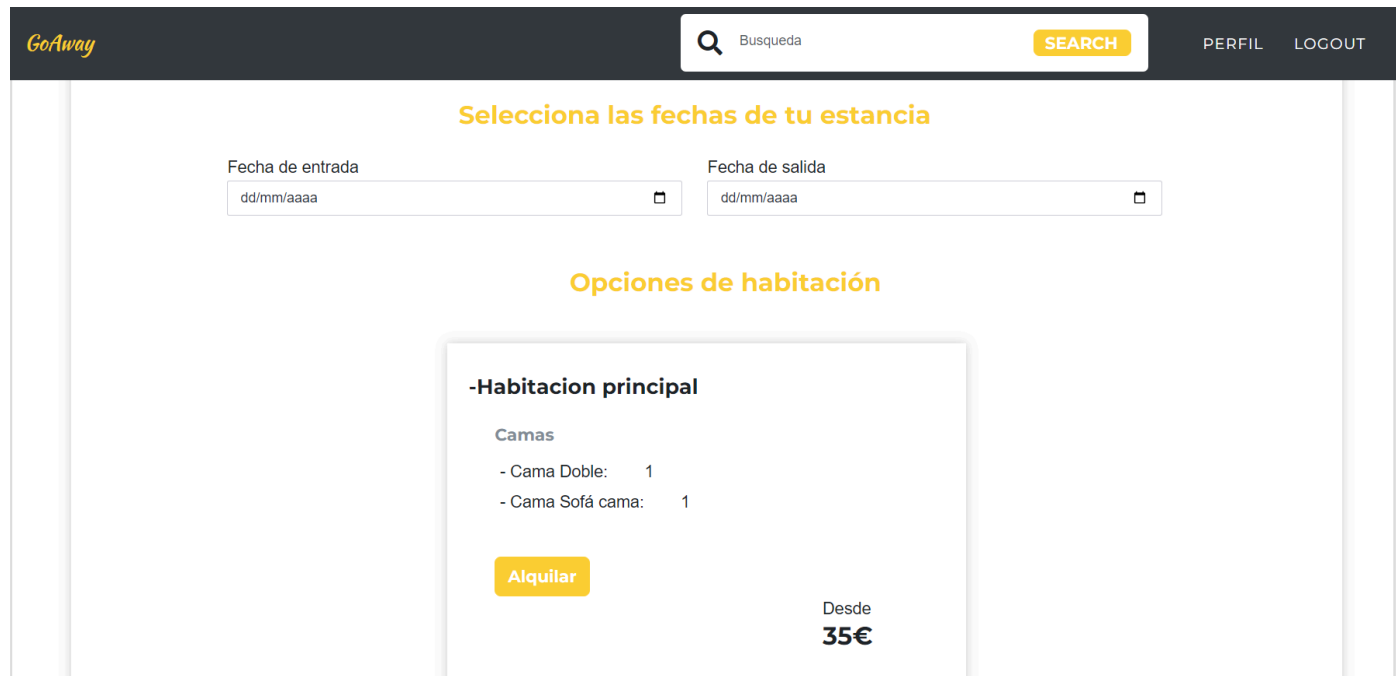
Modificar datos

Nombre <input type="text" value="Marcos"/>	Apellidos <input type="text" value="Prieto Herrero"/>
Telefono <input type="text" value="679563462"/>	Fecha de nacimiento <input type="text" value="2000-02-10"/>
Correo electrónico <input type="text" value="marcos@gmail.com"/>	
Contraseña <input type="text" value="Contraseña"/>	Repite tu contraseña <input type="text" value="Repite tu contraseña"/>

Figura A.11 Vista editar perfil usuario cliente

A.6 Vista de alquilar

Un usuario registrado como cliente puede alquilar cualquier alojamiento de la página. Para ello, tras haber seleccionado el alojamiento que desea, al final de la pantalla tendrá que indicar el periodo durante el que quiere alquilarlo y pulsar el botón “Alquilar”. Como se muestra en la Figura A.12.



The screenshot displays the 'GoAway' website interface for selecting rental dates and room options. At the top, there is a navigation bar with the 'GoAway' logo, a search bar labeled 'Busqueda' with a magnifying glass icon and a yellow 'SEARCH' button, and links for 'PERFIL' and 'LOGOUT'. Below the navigation bar, the main heading is 'Selecciona las fechas de tu estancia'. Underneath, there are two date selection fields: 'Fecha de entrada' and 'Fecha de salida', both with a placeholder 'dd/mm/aaaa' and a calendar icon. Below the date fields, the heading is 'Opciones de habitación'. A central card titled '-Habitacion principal' lists the room details: 'Camas' (Beds) with a sub-list: '- Cama Doble: 1' and '- Cama Sofá cama: 1'. Below the list is a yellow 'Alquilar' button. To the right of the card, the price is shown as 'Desde 35€'.

Figura A.12 Vista alquilar

A.7 Vista de publicar artículo

Un usuario registrado como vendedor puede publicar un artículo en la web pulsando la opción “Publicar artículo” de la cabecera de la página. Aparecerá un formulario con la información a rellenar. En el primer formulario que aparece (Figura A.13) se rellena la información básica del artículo, su vista es la siguiente:

Publicar artículo

Nombre

Descripción

Dirección

País Ciudad

Tipo Alojamiento

Hora Check In Hora Check Out

Características del artículo
 Añadir

Subir fotos
 Ningún archivo seleccionado

Continuar
Cancelar

Figura A.13 Vista publicar artículo

En el siguiente formulario (Figura A.14) se rellenará la información relacionada con las habitaciones y camas pudiéndose añadir más habitaciones o camas.

GoAway

SEARCH

[PERFIL](#)
[PUBLICAR ARTICULO](#)
[LOGOUT](#)

Alta de Habitaciones

Habitación 1

Nombre

<p>Número máximo de personas <input type="text" value="Número máximo de personas"/></p>	<p>Número total de habitaciones <input type="text" value="Número total de habitaciones"/></p>
<p>Tipo de cama <input type="text" value="Sofá cama"/></p>	<p>Número total de camas de ese tipo <input type="text" value="Número total de camas de ese tipo"/></p>

Añadir cama

Añadir otra habitación
Continuar
Volver

Figura A.14 Vista publicar artículo alta habitaciones

Por último, aparece un formulario donde se puede especificar el precio por noche. Para cada habitación que se haya añadido en el formulario anterior. Se puede establecer un precio único (Figura A.15) o indicar un precio distinto (Figura A.16) dependiendo del día de la semana. La vista para estos formularios son las siguientes:

The screenshot shows the GoAway website interface. At the top, there is a dark navigation bar with the GoAway logo on the left, a search bar with a magnifying glass icon and the text 'Busqueda' and a yellow 'SEARCH' button, and links for 'PERFIL', 'PUBLICAR ARTICULO', and 'LOGOUT' on the right. Below the navigation bar, the main content area is titled 'Habitacion principal'. Inside this area, there is a white form titled 'Selecciona el tipo de precio'. It contains two radio button options: 'Establecer un precio único' (which is selected) and 'Establecer un precio por día'. A yellow 'Publicar oferta' button is located to the right of these options. Below the form, the section is titled 'Precio único' and features a single text input field labeled 'Precio único'.

Figura A.15 Vista publicar artículo precio único

The screenshot shows the GoAway website interface, similar to the previous one. The navigation bar is identical. The main content area is titled 'Habitacion principal'. The white form titled 'Selecciona el tipo de precio' now has 'Establecer un precio por día' selected. The yellow 'Publicar oferta' button remains. Below the form, the section is titled 'Precio por día'. It contains seven text input fields, each labeled with a day of the week: 'Lunes' (Precio lunes), 'Martes' (Precio martes), 'Miércoles' (Precio miércoles), 'Jueves' (Precio jueves), 'Viernes' (Precio viernes), 'Sábado' (Precio sábado), and 'Domingo' (Precio domingo).

Figura A.16 Vista publicar artículo precio por día

A.8 Vista de editar artículo

Un usuario comerciante puede modificar la información de los artículos que haya publicado desde su perfil pulsando el botón "Editar información" del alojamiento que quiera. A continuación se mostrará un formulario (Figura A.17):

The screenshot displays the GoAway website interface for editing an accommodation listing. The top navigation bar includes the GoAway logo, a search bar with the text 'Busqueda' and a yellow 'SEARCH' button, and links for 'PERFIL', 'PUBLICAR ARTICULO', and 'LOGOUT'. The main content area is divided into three sections:

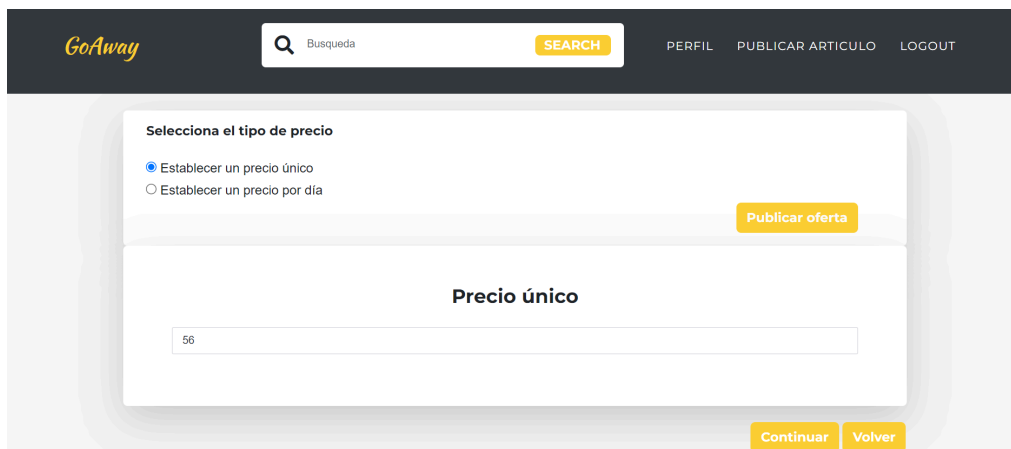
- Modificar datos:** This section contains several input fields: 'Nombre alojamiento' (Pasarela), 'Dirección' (Avenida de la Borbolla 11, España), 'Ciudad' (Sevilla), 'País' (España), 'Descripción' (Este hotel sevillano tiene vistas al parque de Maria Luisa y a la plaza de Espana. Esta; a solo unos minutos a pie de los impresionantes monumentos cercanos a la Catedral.), 'Hora check in' (12:00), 'Hora check out' (13:00), and 'Tipo Alojamiento' (Hotel).
- Modificar Habitaciones:** This section is titled 'Habitacion Doble' and includes fields for 'Nombre de la habitación' (Habitacion Doble), 'Número máximo de personas' (2), and 'Número de habitaciones' (1).
- Modificar camas:** This section includes fields for 'Tipo de cama' (Sofá cama) and 'Número de camas' (1).

At the bottom of the form, there are three buttons: a red 'Eliminar alojamiento' button, a yellow 'Guardar cambios' button, and a yellow 'Cancelar' button.

Figura A.17 Vista editar artículo

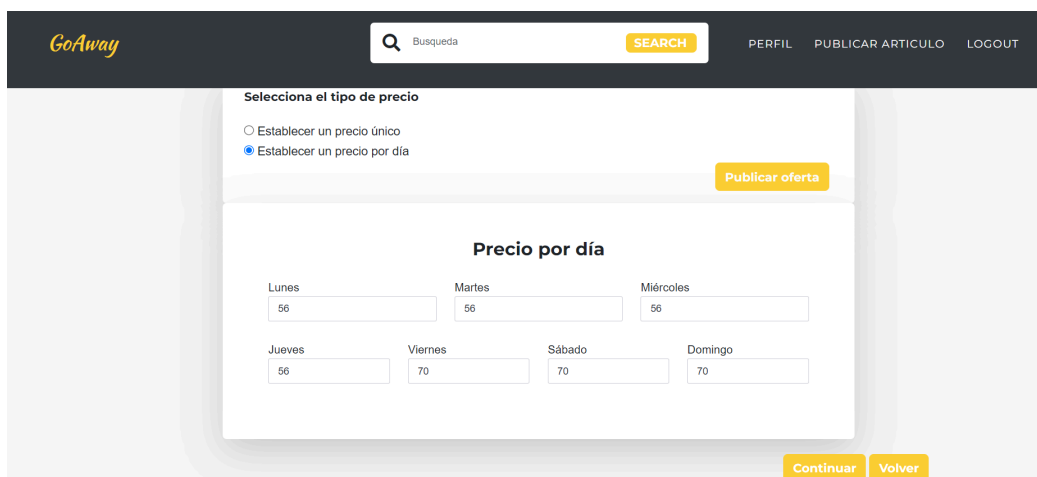
A.9 Vista de modificar precios artículo

Un usuario comerciante puede modificar el precio de la reserva de su alojamiento. Para ello, desde su perfil, tiene que pulsar sobre el botón “Editar precios y ofertas” del alojamiento que quiera modificar. Puede tanto establecer un precio único como establecer uno distinto dependiendo del día de la semana del que se trate. La vista para ambos casos es la misma que en las Figuras A.15 y A.16 respectivamente.



The screenshot shows the GoAway website interface for editing a single price. At the top, there is a search bar with the text 'Busqueda' and a 'SEARCH' button. To the right are links for 'PERFIL', 'PUBLICAR ARTICULO', and 'LOGOUT'. The main content area is titled 'Selecciona el tipo de precio' (Select the type of price) and has two radio buttons: 'Establecer un precio único' (selected) and 'Establecer un precio por día'. A yellow 'Publicar oferta' button is on the right. Below this is a section titled 'Precio único' with a single input field containing the value '56'. At the bottom right are 'Continuar' and 'Volver' buttons.

Figura A.18 Vista editar precio único



The screenshot shows the GoAway website interface for editing a price per day. At the top, there is a search bar with the text 'Busqueda' and a 'SEARCH' button. To the right are links for 'PERFIL', 'PUBLICAR ARTICULO', and 'LOGOUT'. The main content area is titled 'Selecciona el tipo de precio' (Select the type of price) and has two radio buttons: 'Establecer un precio por día' (selected) and 'Establecer un precio único'. A yellow 'Publicar oferta' button is on the right. Below this is a section titled 'Precio por día' with seven input fields for the days of the week: Lunes (56), Martes (56), Miércoles (56), Jueves (56), Viernes (70), Sábado (70), and Domingo (70). At the bottom right are 'Continuar' and 'Volver' buttons.

Figura A.19 Vista editar precio por día

A.10 Vista de publicar oferta

Un usuario comerciante puede publicar una oferta sobre cualquiera de los alojamientos que haya publicado. Para acceder a este formulario desde su perfil, tiene que pulsar sobre el botón "Editar precios y ofertas" del alojamiento que quiera. Una vez dentro tiene que pulsar el botón "Publicar oferta". Se mostrará una tabla (Figura A.20) con todas las ofertas publicadas para ese alojamiento además del formulario a rellenar para publicar una nueva.



The screenshot shows the GoAway website interface. At the top, there is a search bar with the text "Busqueda" and a "SEARCH" button. To the right of the search bar are links for "PERFIL", "PUBLICAR ARTICULO", and "LOGOUT". The main content area is titled "Ofertas actuales de Pasarela". Below this title is a table with the following columns: "Precio por noche", "Fecha de inicio", "Fecha de fin", and "Acciones". The table contains one row with the following data: "43€", "09-05-2024", "24-05-2024", and "Modificar" and "Eliminar" links. Below the table is a link "Publicar nueva oferta". Below the table is a form titled "Publicar oferta". The form has three input fields: "Precio por noche:", "Fecha de inicio:", and "Fecha de fin:". The "Fecha de inicio:" and "Fecha de fin:" fields have a date format "dd/mm/aaaa" and a calendar icon. Below the form is a "Publicar oferta" button.

Figura A.20 Vista publicar oferta

También puede publicar una oferta en el proceso de creación del artículo en el formulario de precios como se muestra en la figura A.15 y A.16.

A.11 Vista de modificar oferta

Un usuario comerciante puede modificar una oferta publicada sobre cualquiera de sus alojamientos. Para ello en la figura A.20 mostrada en el apartado anterior hay que pulsar la opción "Modificar" de la oferta que se desee. La vista es la siguiente:

Figura A.21 Vista modificar oferta

A.12 Vista de ver reservas del artículo

El usuario propietario de un artículo puede ver la disponibilidad de dicho alojamiento (Figura A.22). Para ello tiene que acceder a su perfil y pulsar el botón “Ver reservas” de uno de sus artículos. Las reservas de días pasados aparecen en rojo.

Id Reserva	Nombre habitación	Id cliente	Precio	Fecha inicio	Fecha fin	Pasada
6	5	70	Apartamento Recoletos	20-05-2024	26-05-2024	No
5	5	70	Apartamento Recoletos	03-05-2024	05-05-2024	Sí

Figura A.22 Vista ver reservas del artículo

A.13 Vista de búsqueda

Para facilitar la búsqueda de alojamientos, se han añadido filtros para buscar en función del precio deseado, puntuación y de aquellos que tienen habitaciones libres u

ofertas (Figura A.23). Para acceder a esta vista simplemente hay que realizar una búsqueda desde la barra de búsqueda de la cabecera de la página. La vista es la siguiente:

Ocultar filtros

Filtros de búsqueda

Rango de precios:
9€ - 64€

Puntuacion media:
0 - 10

Tipo Alojamiento:

- Hotel
- Apartamento
- Hostal
- Resort
- Villa
- Casa rural
- Albergue
- Camping

Características del artículo

Introduce una característica Añadir

Aplicar filtros Limpiar filtros

Figura A.23 Vista filtros búsqueda

La vista para los alojamientos encontrados es la siguiente:

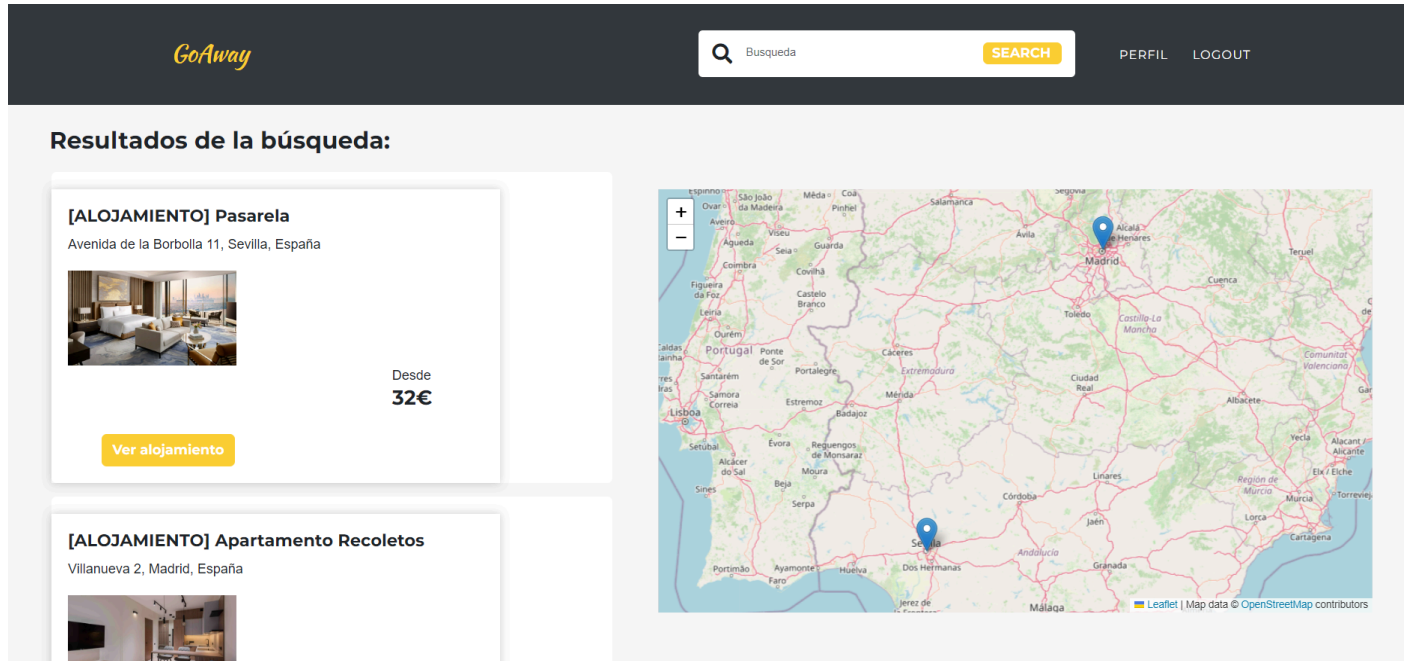


Figura A.24 Vista búsqueda

A.14 Vista de artículo

Pulsando un artículo de la página, se puede ver la información en detalle. Además el usuario puede realizar comentarios y puntuar aquellos alojamientos en los que se haya alojado o realizar una reserva. La vista es la siguiente:

Hotel Peninsular

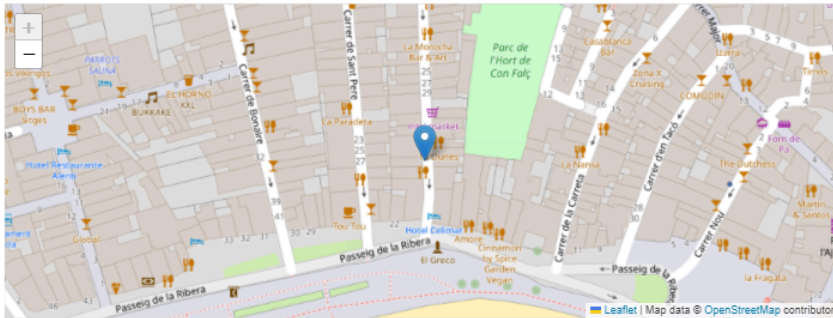


Descripción

Este hotel encantador está situado a pocos minutos a pie de las Ramblas.

Dirección

Carrer de Sant Pau, 34, Barcelona, España



Detalles

Hora check in: 16:00:00

Hora check out: 16:30:00

Tipo: Hotel

Características

Wifi Piscina Calefaccion TV Nevera Ocio

Puntuación media
0 / 10

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Comentarios

Puntuación	Comentario
No hay comentarios todavía	

Alquila este alojamiento

Selecciona las fechas de tu estancia

Fecha de entrada

Opciones de habitación

-Habitacion matrimonial

Camas

- Cama Doble extra: 1


Alquilar

Desde **64€**

Figura A.25 Vista artículo

A.15 Vista de comentarios

Un usuario puede comentar su experiencia y puntuar alojamientos en los que se haya hospedado, es decir la fecha de la reserva tiene que haber pasado. Si el usuario quiere dejar un comentario simplemente tendrá que acceder al alojamiento en concreto y rellenar el cuadro de diálogo. Como se muestra en la Figura A.26.



¡Comparte tu experiencia!

Comentario

Puntuación

0

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Enviar

Puntuación media

0 / 10

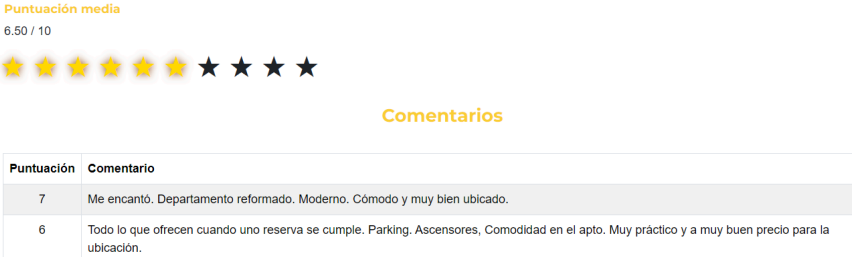
★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Comentarios

Puntuación	Comentario
No hay comentarios todavía	

Figura A.26 Vista comentario

Si el usuario no puede comentar en dicho alojamiento, el recuadro amarillo de la Figura A.26 no aparece. Una vez publicado el comentario la vista es la siguiente:



Puntuación media

6.50 / 10

★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Comentarios

Puntuación	Comentario
7	Me encantó. Departamento reformado. Moderno. Cómodo y muy bien ubicado.
6	Todo lo que ofrecen cuando uno reserva se cumple. Parking. Ascensores, Comodidad en el apto. Muy práctico y a muy buen precio para la ubicación.

Figura A.27 Vista comentarios publicados

Apéndice B - Repositorio de código fuente

En el siguiente enlace a la plataforma Github se puede encontrar el [código fuente](#) al completo. Además se incluyen los archivos SQL necesarios para la creación de las tablas.

<https://github.com/ALLaguna/GoAway>