



universidad
de león

Escuela de Ingenierías



Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

APLICACIÓN WEB DE GESTIÓN
INMOBILIARIA

Autor: Miguel Turrión Gutiérrez

Tutora: Lidia Sánchez González

(JULIO, 2023)

UNIVERSIDAD DE LEÓN
Escuela de Ingenierías
GRADO EN INGENIERÍA INFORMÁTICA
Trabajo de Fin de Grado

ALUMNO: Miguel Turrión Gutiérrez

TUTORA: Lidia Sánchez González

TÍTULO: Aplicación web de gestión inmobiliaria

CONVOCATORIA: Julio, 2023

RESUMEN:

Este trabajo fin de grado diseña y desarrolla una aplicación web que gestiona información inmobiliaria, mostrando alojamientos disponibles y permitiendo a los usuarios arrendarlos durante un periodo determinado de tiempo. Incluye la gestión de roles de usuarios, donde hay perfiles que permiten la incorporación a la base de datos de alojamientos, mientras otros pueden alquilar los mismos, gestionando toda la información del proceso de forma centralizada en la aplicación. Además la aplicación cuenta con otras características como la recomendación, comparación, valoración y búsqueda con filtros simples y otros más avanzados para facilitar el descubrimiento de nuevos lugares que se adecuen a los gustos e intereses de los usuarios. También, se han creado otras herramientas que han facilitado el desarrollo de la aplicación, una para la obtención de datos de alojamientos de forma automática y otra para la comprobación automática del correcto funcionamiento de la misma.

Índice general

Índice de figuras	III
Índice de tablas	V
Glosario de términos	VII
Introducción	1
1. Estudio del problema	5
1.1. El contexto del problema	5
1.2. El estado de la cuestión	5
1.3. La definición del problema	6
2. Gestión de proyecto software	8
2.1. Alcance del proyecto	8
2.1.1. Definición del proyecto	8
2.1.2. Estimación de tareas y recursos	9
2.1.3. Presupuesto	10
2.2. Plan de trabajo	12
2.2.1. Identificación de tareas	12
2.2.2. Asignación de tareas y estimación	15
2.2.3. Planificación de tareas	16
2.3. Gestión de recursos	16
2.3.1. Especificación de recursos	16
2.3.2. Asignación de recursos	16
2.4. Gestión de riesgos	18
2.4.1. Identificación de riesgos	18
2.4.2. Análisis de riesgos	18
2.5. Legislación y normativa	19

3. Solución	29
3.1. Descripción de la solución	29
3.2. El proceso de desarrollo	31
3.2.1. Análisis	31
3.2.2. Diseño	54
3.2.3. Implementación	60
3.2.4. Pruebas	62
3.3. El producto del desarrollo	63
4. Evaluación	72
4.1. Proceso de evaluación	72
4.1.1. Forma de evaluación	72
4.1.2. Casos de prueba	72
4.2. Análisis de resultados	73
Conclusión	76
Lista de referencias	80
A. Control de versiones	83
B. Seguimiento de proyecto	85
B.1. Forma de seguimiento	85
B.2. Planificación inicial	85
B.3. Planificación final	85
C. Manual de usuario	87
D. Cuestionario de evaluación	90

Índice de figuras

2.1. Gráfico Gantt	17
3.1. Popularidad de React vs Angular vs Vue.js	31
3.2. Tecnologías usadas en el TFG	54
3.3. Modelo MVC en el proyecto	56
3.4. Diagrama de tablas de la base de datos	56
3.5. Cambios en la estructura página web	57
3.6. Respuesta del backend en formato JSON	57
3.7. Respuesta de imagen del backend	58
3.8. JWT almacenado en el navegador	59
3.9. Validación del JWT	60
3.10. Lenguajes usados en el proyecto según GitHub	60
3.11. Resultado de tests automáticos	63
3.12. Selenium ejecutando uno de los tests	64
3.13. Página de inicio desde un ordenador	64
3.14. Página de inicio desde un móvil	65
3.15. Diferentes estados del usuario en la aplicación	65
3.16. Vista de un alojamiento	66
3.17. Perfil público de un usuario	67
3.18. Registro de la aplicación	67
3.19. Correo para validar cuenta	68
3.20. Perfil de usuario	68
3.21. Alojamientos creados por un usuario	68
3.22. Formulario para crear un nuevo alojamiento	69
3.23. Reservas realizadas	69
3.24. Ganancias mensuales obtenidas	70
3.25. Valoraciones hechas de alojamientos	70
3.26. Sistemas de mensajes de la aplicación	70
3.27. Vista de administración	71

3.28. Administración de usuarios de la aplicación	71
3.29. Ejecución del <i>frontend</i> y <i>backend</i> de la aplicación	71
4.1. Resultado de test automáticos	73
4.2. Resultado de test automáticos	74
A.1. Repositorio en GitHub	83
A.2. Commits del repositorio	84
B.1. Gráfico Gantt final	86

Índice de tablas

2.1. Presupuesto de personal	10
2.2. Presupuesto de hardware	11
2.3. Costes indirectos	11
2.4. Presupuesto total	12
2.5. Tarea A	15
2.6. Tarea B	15
2.7. Tarea C	21
2.8. Tarea D	21
2.9. Tarea E	22
2.10. Tarea F	23
2.11. Tarea G	24
2.12. Tarea H	24
2.13. Riesgo 1	24
2.14. Riesgo 2	25
2.15. Riesgo 3	25
2.16. Riesgo 4	26
2.17. Riesgo 5	26
2.18. Riesgo 6	27
2.19. Riesgo 7	27
2.20. Riesgo 8	27
2.21. Riesgo 9	28
2.22. Análisis de riesgos	28
3.1. Caso de uso 1	35
3.2. Caso de uso 2	36
3.3. Caso de uso 3	37
3.4. Caso de uso 4	39
3.5. Caso de uso 5	40
3.6. Caso de uso 6	40

3.7. Caso de uso 7	41
3.8. Caso de uso 8	42
3.9. Caso de uso 9	42
3.10. Caso de uso 10	43
3.11. Caso de uso 11	44
3.12. Caso de uso 12	44
3.13. Caso de uso 13	45
3.14. Caso de uso 14	46
3.15. Caso de uso 15	47
3.16. Caso de uso 16	48
3.17. Caso de uso 17	49
3.18. Caso de uso 18	50
3.19. Caso de uso 19	51
3.20. Caso de uso 20	52
3.21. Caso de uso 21	53
3.22. Códigos de error HTML	58

Glosario de términos

Angular : *Framework* utilizado para la creación de aplicaciones web.

API : En inglés *Application Programming Interfaces*, son un conjunto de procedimientos que ofrece una biblioteca para ser utilizada por otro software.

backend : Parte oculta de la aplicación la cual recibe los datos, los procesa y envía una respuesta.

backup : Una copia de seguridad que es utilizada como respaldo en caso de pérdida de datos.

bcrypt : Función concreta de *hashing* de contraseñas.

Bootstrap : *Framework frontend* utilizado para desarrollar aplicaciones web.

commit : En GitHub hacer un commit es confirmar los cambios que se han realizado.

cookies : Pequeña información enviada por un sitio web y almacenada en el navegador del usuario, para que la página pueda consultar la actividad previa del navegador.

CSS : En inglés *Cascading Style Sheets*, es un lenguaje de diseño gráfico para cambiar los estilos de nuestra página web.

driver : Programa informático que permite al sistema operativo interactuar con un elemento.

express : Entorno de trabajo para aplicaciones web más popular de *Nodejs*.

fetch : Mecanismo asíncrono que permite realizar llamadas a una *API* externa sin detener la ejecución de otras instrucciones.

FontAwesome : Conjunto de herramientas de fuentes e iconos basado en *CSS* y *Less*.

- framework** : Herramientas de software ya desarrolladas que nos ayudan en la creación de nuestro proyecto.
- frontend** : Interfaz de la aplicación visible por el usuario y con la que interactúa, permitiéndole la navegación por la aplicación.
- GET** : Método para obtener respuestas a nuestras peticiones.
- hardware** : Elementos físicos o materiales que componen un sistema informático.
- hash** : Función que convierte los elementos de entrada mediante un algoritmo matemático en una nueva serie de caracteres de longitud fija.
- JavaScript** : Lenguaje de programación interpretado.
- JSON** : Formato de texto sencillo para el intercambio de datos.
- JWT** : En inglés JSON WEB TOKEN mecanismo de identidad para un determinado usuario.
- link** : Elemento de un documento electrónico que permite acceder a otro documento o a otra parte del mismo.
- mariaDB** : Sistema de gestión de bases de datos relacionales derivado de *MySQL*.
- Materialize** : *Framework* de *CSS* responsive basado en *Material Design*.
- mongoDB** : Sistema de datos *NoSQL* orientado a documentos.
- Nodejs** : Entorno de *backend* multiplataforma que se ejecuta con *JavaScript*.
- open source** : En español código abierto es un tipo de software cuyo código fuente puede ser visto, modificado y distribuido.
- OpenStreetMap** : Colección de herramientas de código abierto y gratuitas para mapear el mundo.
- php** : Lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web.
- POST** : Método de solicitud que envía los datos a un servidor web que van incluidos en el cuerpo del mensaje de la solicitud.
- React** : Librería de código abierto para diseñar interfaces de usuario en una sola página.
- responsive** : Modelo de diseño web, que permite adaptar la apariencia dependiendo del dispositivo que se esté usando en el momento.

- salt** : Bits aleatorios que se usan como una de las entradas en una función *hash*.
- sandbox** : Entorno de pruebas.
- scraper** : Programa informático que permite extraer rápidamente y de forma automática la información deseada de una página web.
- Secuestro de sesión** : Uso de la cookie de sesión válida y activa del ordenador del atacado para obtener acceso no autorizado a información o servicios de una aplicación.
- Selenium** : Entorno de pruebas de software para aplicaciones web.
- SHA-256** : Conjunto de funciones *hash* diseñadas por la Agencia de Seguridad Nacional.
- sofort** : Sistema de pago por internet que se realiza desde tu cuenta bancaria a la del comercio sin intermediarios.
- software** : Conjunto de programas que permiten a un sistema informático realizar tareas.
- imagen de stock** : Imágenes que el autor mantiene los derechos de autor y cede el uso de la fotografía.
- Tailwind** : *Framework* de *CSS* de código abierto para el diseño de páginas web.
- tipado** : Forma de declarar los tipos de variable.
- Vue.js** : *Framework* de código abierto para la creación de interfaces de usuario y aplicaciones de una sola página.
- XSS** : En inglés *Cross-site scripting*, es un tipo de vulnerabilidad típico en las aplicaciones web, que permite a un atacante inyectar código malicioso del lado del cliente.

Introducción

Este Trabajo Fin de Grado diseña y desarrolla una aplicación web centrada en la gestión inmobiliaria utilizando tecnologías actuales como son *React* y *Nodejs*, entre otras, creando todas las funcionalidades y servicios que una aplicación de este tipo pueda requerir, desarrollando diferentes pruebas para la verificación del correcto funcionamiento y todo lo necesario para realizar un proyecto software profesional.

Planteamiento del problema

Se quiere desarrollar una aplicación web profesional, con tecnologías actuales y de manera que pueda tener algo de impacto en el mercado. Este trabajo ayuda a reforzar los conocimientos aprendidos durante el grado y aprender nuevas tecnologías que se necesiten en el desarrollo del proyecto. Con este planteamiento y viendo el avance de la tecnología y la mejora en las infraestructuras de los países que permiten a los seres humanos recorrer grandes distancias en poco tiempo, la fácil accesibilidad a estos servicios, su bajo coste y la gran variedad que se ofrecen, hay una gran cantidad de desplazamientos. Un claro ejemplo es el AVE en España, que nos permite hacer trayectos de 600 kms en solamente 2 horas y 30 minutos. Por lo que el desarrollo de una aplicación enfocada en este sector se vuelve interesante.

Por ello se ha de investigar sobre las aplicaciones ya existentes en el mercado como pueden ser Airbnb¹ o VRBO², ver las características que tienen y crear una aplicación con las mejores funcionalidades de cada una.

Objetivos

El objetivo general es realizar una aplicación web funcional, que pueda ser utilizada por los usuarios sin encontrarse con errores que perjudiquen su experiencia y que

¹ *Airbnb página web.* <https://www.airbnb.es/>

² *VRBO página web.* <https://www.vrbo.com/es-es>

cubran todas las posibles necesidades que puedan tener. Tener un diseño adaptable, de fácil mantenimiento y modular que permita adicionar nuevas funcionalidades en el futuro de forma sencilla. Nuestro proyecto en el sector inmobiliario va a necesitar tener funcionalidades comunes en este tipo de aplicaciones como pueden ser realizar búsquedas, filtrar, etc. Todo esto respaldado con un conjunto de pruebas que verifiquen el correcto funcionamiento de la aplicación.

Detallando este objetivo general, se enumeran los siguientes objetivos secundarios:

- Comprender cuáles son las necesidades del sector de mi aplicación, qué funciones debe tener y entender cómo utilizan los recursos y servicios este tipo de páginas web.
- Averiguar cuáles son las tecnologías más habituales en el desarrollo de una aplicación web y valorar su utilización.
- Desarrollar un *scraper* para la obtención automática de información sobre alojamientos.
- Diseñar y desarrollar una base de datos que cubra todas las necesidades del trabajo.
- Aprender sobre el funcionamiento de un sistema de mapas como GoogleMaps e integrarlo en la aplicación.
- Investigar y aprender sobre la implementación de un sistema de pagos e integrarlo en la aplicación.
- Desarrollar un sistema que permita el envío de correos electrónicos para que los usuarios se mantengan informados sobre novedades o actualizaciones.
- Crear un diseño web adaptable dependiendo del dispositivo que se esté utilizando y que su interfaz sea sencilla, simple e intuitiva, proporcionando una buena experiencia de usuario.
- Desarrollar pruebas de testeo que validen el correcto funcionamiento de la aplicación.

A nivel personal el desarrollo de este tipo de aplicaciones siempre es enriquecedor, ya que siempre surgen cuestiones que hay que resolver y te permiten obtener experiencia en tecnologías que ya conoces y aprender otras nuevas.

Metodología

En el proyecto se ha utilizado una metodología de desarrollo software tradicional de forma incremental [18] por lo que el producto se va construyendo de manera progresiva. Las etapas han sido divididas por periodos de entre una y tres semanas y en cada etapa se ha intentado agregar al menos una nueva funcionalidad, exceptuando las primeras fases del desarrollo que se han utilizado para investigar y establecer los requisitos que se deben cumplir.

En cada fase del desarrollo de las características del software se encuentran los procesos de planteamiento, análisis, diseño, programación, pruebas y puesta en marcha de la funcionalidad hecha.

El uso de esta metodología permite ver los resultados de forma rápida, por lo que facilita las modificaciones o implementaciones de nuevas características según las necesidades que surjan.

Estructura del trabajo

El documento consta de diferentes apartados y a continuación se explica el contenido de cada uno de ellos.

Sección 1: Estudio del problema. En este primer apartado se expone la situación inicial de la que se parte, qué tecnologías han sido seleccionadas y el planteamiento para el desarrollo de la aplicación.

Sección 2: Gestión de proyecto software. En este punto se define el alcance del proyecto, estimación de tareas y recursos que han sido necesarios. Además se identifican las tareas, su división, su duración y su planificación.

Sección 3: Solución. En este apartado se explica la solución llevada a cabo justificándola y describiéndola.

Sección 4: Evaluación. En esta sección se explica la forma de valorar el funcionamiento del proyecto, cómo se ha evaluado y el análisis de los resultados

Sección 5: Conclusión. En este último capítulo se exponen los problemas encontrados durante la realización del proyecto, las conclusiones obtenidas, valoración de las posibles mejoras futuras y la opinión personal.

Anexo A: Control de versiones. En este anexo se indica cuál ha sido la herramienta para el control de versiones de la aplicación y su evolución.

Anexo B: Seguimiento del proyecto. En este anexo se explica la evolución que ha tenido el proyecto.

Anexo C: Manual de usuario. En este anexo se explica cómo utilizar la aplicación.

Anexo D: Cuestionarios de evaluación. En este anexo se muestran los cuestionarios realizados para evaluar el funcionamiento.

Capítulo 1

Estudio del problema

En ese apartado se describe la situación inicial de la que se parte en el proyecto y las soluciones que se han planteado para llevar a cabo el desarrollo.

1.1. El contexto del problema

El objeto del trabajo de fin de grado era diseñar y realizar una aplicación web plenamente funcional, que permitiera aprender y reforzar las tecnologías utilizadas actualmente en este sector y adquirir experiencia en el desarrollo de aplicaciones de cara al mundo laboral. Entre las posibles aplicaciones, mi tutora me sugirió la realización de una plataforma de gestión de inmuebles, puesto que incluía todos los aspectos de una aplicación profesional.

Por eso, se desarrolla una aplicación web que ofrece características similares a los de un hotel pero con mayores servicios y diversidad de localizaciones. La propia web no ofrece los alojamientos por sí misma sino que son los propios usuarios los que publican los alojamientos y ésta es la encargada de validar la veracidad de la vivienda y de las transacciones.

1.2. El estado de la cuestión

Existen una gran cantidad de aplicaciones y páginas web que ofrecen este tipo de servicios y es normal ya que es un mercado que va a ir creciendo a lo largo de los años.

Una de las aplicaciones más conocidas es *Airbnb* [32]; al ser una empresa ya madura se nota en las funcionalidades de su aplicación, ya que disponen de mayores

herramientas para el desarrollo y para la variedad de productos. Una de las características que se pueden ver a simple vista al entrar en la web es el uso de etiquetas para diferenciar las viviendas que permite al usuario tener más filtros a la hora de buscar alojamientos más fácilmente y mi aplicación carece de ellos, ya que no hay registradas tanta variedad de alojamientos para que el uso de estas etiquetas sean precisas. También dividen a los usuarios en clientes y hospedadores mientras que en mi aplicación un usuario puede ser cliente y hospedador con la misma cuenta sin necesidad de realizar más burocracia.

Dentro de la información de los alojamientos disponen de muchos más detalles de la vivienda, mientras que en mi aplicación se pueden encontrar los básicos y si estás interesado en conocer más información puedes hablar directamente con el hospedador.

Te permiten hacer listas de favoritos y mi aplicación no, ya que lo veo algo innecesario y tedioso a veces, ya que sólo quieres añadirlo rápidamente para verlo más tarde. Dispone de una mayor cantidad de filtros debido a que almacenan mayor información de los alojamientos.

Una característica ausente en *Airbnb* es mostrar qué alojamientos ya has entrado a ver sus detalles y marcártelos como vistos como por ejemplo sí tiene *Idealista* [33] y poder filtrar por las viviendas más populares de la aplicación.

Tampoco disponen de un sistema de recomendación de alojamientos, que puedan hacer a los clientes descubrir nuevos lugares interesantes a los que viajar.

Este tipo de aplicaciones también permiten los anuncios de alojamientos de hoteles y de las transacciones se llevan un porcentaje en forma de comisión.

1.3. La definición del problema

Después de investigar este tipo de aplicaciones, en este proyecto no se va a reinventar la rueda, pero se busca crear una aplicación que sea de fácil uso para los usuarios y disponga de las mejores características de las aplicaciones que hay en el mercado.

Por lo que en mi aplicación va a cumplir las necesidades básicas de este tipo de aplicaciones como pueden ser hacer búsquedas de lugares, ya sea un país, una comunidad o una ciudad, poder filtrar por diferentes parámetros precio, popularidad, capacidad y pudiéndolos combinar para realizar búsquedas más precisas, acceso a los detalles de los alojamientos aportando información más concreta sobre ellos,

añadirlos a tus favoritos, compartirlo con tus amigos o poder comunicarte con los hospedadores mediante un sistema de mensajería de la aplicación para resolver tus dudas, realizar reservas y pagar dentro de la propia aplicación, añadir reseñas ya sea sobre el alojamiento que has visitado o como hospedador sobre tus inquilinos.

Y otras funcionalidades más concretas de mi aplicación van a ser la facilidad de poder publicar alojamientos, simplemente rellenando el formulario sin necesidad de hacer tareas adicionales para ello, marcar los alojamientos como vistos si ya has accedido a sus detalles alguna vez y un sistema de recomendación de nuevos alojamientos a los que visitar basándose en tus experiencias anteriores.

Capítulo 2

Gestión de proyecto software

2.1. Alcance del proyecto

En este apartado se van a explicar detalladamente las tareas, los recursos, presupuesto y riesgos esperados para desarrollar la aplicación.

2.1.1. Definición del proyecto

Se pretende desarrollar una página web que permita a los usuarios tanto buscar y reservar viviendas como poder publicar sus propios alojamientos, también con otras características como añadir a favoritos, recomendaciones de alojamientos entre otras y un sistema de administración para controlar a los usuarios.

La aplicación va a estar dividida en dos partes, *frontend* y *backend*; ambos se irán desarrollando simultáneamente a la vez que se van añadiendo las diferentes vistas de la página. Éstas se crearán de forma modular para que sea más fácil su entendimiento y edición, gracias a las funcionalidades que nos otorgan *React* y *Nodejs* respectivamente, que hace que sea posible. Se desarrolla una herramienta para realizar las pruebas automáticas con otras dos herramientas una para *scrapear* información útil en mi caso de *AirBnB* para tener algunos alojamientos de ejemplo y poder probar la aplicación, y la otra para hacer pruebas de funcionamiento.

La página también cuenta con un sistema de recomendaciones, que se basa en tus reservas y alojamientos que tienes en favoritos para recomendar nuevas experiencias que te puedan interesar.

2.1.2. Estimación de tareas y recursos

Tareas

Cada tarea tiene un tiempo determinado previsto, como mínimo esa duración será de una semana. Se han tenido en cuenta todas las posibles tareas iniciales que pueden darse y al ser un desarrollo de manera incremental la cantidad de funcionalidades que tiene la aplicación serán aumentadas semanalmente. También se realizarán reuniones semanales para comprobar la evolución del sistema y valorar nuevas tareas.

Recursos

Para el desarrollo de la aplicación se necesitarán tanto recursos humanos como físicos para poder realizarla:

- **Humanos:** Se pretende contratar un equipo que esté compuesto por 5 empleados los cuales cada uno tendrá una función específica en el proyecto que serán:
 - **Jefe de proyecto**, tiene el rol más importante ya que es el que va a asignar las tareas y comprobar la evolución del proyecto a lo largo de todo el desarrollo.
 - **Diseñador gráfico**, estará centrado en las fases iniciales del proyecto para plantear tanto el logotipo, como el diseño de las diferentes vistas que pueda tener la web.
 - **Backend developer**, es el que va a desarrollar toda la lógica de la aplicación.
 - **Frontend developer**, trabajará en conjunto con el *backend developer* en la parte visible de la aplicación.
 - **Tester**, es el encargado de ir probando las nuevas funcionalidades que aparecen en la aplicación.
- **Físicos:** Para desarrollar la aplicación se necesitará adquirir hardware, software y un lugar de trabajo.
 - **Hardware**, se indicará la amortización del equipo necesario para cada empleado para que puedan desarrollar su función.
 - **Software**, no hay ningún coste debido a que todas las herramientas utilizadas son de *open source*.

- **Costes indirectos:**

- Oficina, se ha decidido alquilar una oficina *coworking*, ya que ya están equipadas con diferentes servicios como puede ser internet, luz o impresora, su precio es asequible y hay tarifas muy flexibles. Supone el 15% del presupuesto total y se indica como costes indirectos.

2.1.3. Presupuesto

En el presupuesto se va a detallar una estimación tanto de los gastos para personal y para materiales que se han considerado en el proyecto. Este ha sido medido para realizar el proyecto en León ciudad en España, por lo que los gastos en sueldos/oficina puede variar según la región.

Coste de personal

- Jefe de proyecto, el sueldo medio aproximado es de 46.200 €/ año por lo que 26 €/ hora según [29].
- Diseñador gráfico, el sueldo medio aproximado es de 28.700 €/ año por lo que 16 €/ hora según [28].
- *Backend y Frontend developer*, el sueldo es parecido aunque cada uno esté especializado en una tecnología, aproximadamente es de 31.600 €/ año por lo que 18 €/ hora según [27].
- *Tester*, el sueldo medio aproximado es de 25.500 €/ año por lo que 13 €/ hora según [30].

Tabla 2.1: Presupuesto de personal

Tarea	Perfil	Horas	Euros/Hora	Total
Supervisión del Proyecto	Jefe de Proyecto	960	26	24960 €
Diseño aplicación	Diseñador Gráfico	640	16	10240 €
Probar la aplicación	Tester	800	13	10400 €
Desarrollo aplicación	Backend Developer	960	18	17280 €
Desarrollo aplicación	Frontend Developer	960	18	17280 €
Total				80160 €

Amortización del hardware

- Ordenadores, se necesitan 5 ordenadores portátiles [23] para que los empleados puedan trabajar desde casa si en alguna situación es requerida, con los siguientes componentes.
 - Procesador Intel® Pentium® Gold 4425Y
 - Memoria 8 GB de RAM
 - Almacenamiento Unidad de estado sólido (SSD): 128 GB
 - Controlador gráfico Intel® UHD Graphics 615
- Periféricos, se necesitan diferentes periféricos para facilitar el trabajo de los empleados.
- Ratones, se adquirirán 5 ratones de este tipo [25].
- Auriculares, se adquirirán 5 auriculares con micrófono de este tipo [1].

Con un porcentaje del 33% [34] de amortización nos queda como:

Tabla 2.2: Presupuesto de hardware

Concepto	Amortización	Cantidad	Total
Portátil	204,27	5	1021,35 €
Ratón	1,3	5	6,5 €
Auricular	2,6	5	13 €
Total			1040,85 €

Costes indirectos

- Despacho privado, para que los empleados desempeñen su trabajo con un precio de 450 €/ mes con IVA incluido [6].

Tabla 2.3: Costes indirectos

Concepto	Precio Euros	Meses	Total
Despacho privado	450	6	2700 €
Total			2700 €

Coste total

El coste total del proyecto es de 125049,6€, como se muestra en la Tabla 2.4.

Tabla 2.4: Presupuesto total

Concepto	Coste (Euros)
Costes de personal	80160
Amortización de hardware (5 %)	4008
Costes indirectos (15 %)	12024
Beneficio industrial (15 %)	12024
Impuestos (I.V.A. 21 %)	16833,6
Total Proyecto	125049,6 €

2.2. Plan de trabajo

En esta sección se va a explicar cómo se han establecido las diferentes tareas y tiempo aproximado que durará cada una en el desarrollo de la aplicación.

2.2.1. Identificación de tareas

(A) Investigación sobre el sector y tecnologías.

Al tratar del desarrollo de una página web y una vez conocido el sector lo primero que debemos hacer es informarnos sobre el funcionamiento de páginas similares a lo que queremos hacer y redactar una lista de requisitos que queremos que cumpla.

Una vez hecho esto, seleccionamos las tecnologías que sean consideradas más adecuadas para el proyecto

(B) Obtener información de ejemplo.

Para probar nuestra web necesitamos alojamientos de ejemplo. Para ello se han descargado imágenes de stock de Unplash [17] y los detalles de los alojamientos han sido inventados.

(C) Selección nombre y creación logotipo.

Como cualquier empresa debe tener un nombre y diseñar un logotipo acorde al mismo y al sector que corresponde, para que los usuarios puedan identificarlo fácilmente con la empresa.

(D) Diseño de interfaz de la aplicación y estructura.

La interfaz es lo primero que el usuario ve al usar la aplicación web por lo que debe tener una paleta de colores que sea atrayente y una estructura fácil de usar para que los clientes reconozcan con una simple mirada qué es lo que

tienen que hacer para buscar lo que les interesa. También conocer qué vistas será necesario registrar un usuario y cuáles te permitirá una navegación libre.

(E) Diseño de las pruebas automáticas de funcionamiento Se ha diseñado y desarrollado una aplicación que permite comprobar que el funcionamiento es el apropiado de forma automática. Para ello se ha utilizado *Selenium* y el driver de *Mozilla* llamado *Gecko* que nos permite simular el navegador, rellenar las entradas de los diferentes formularios y comprobar la respuesta.

(F) Desarrollo de la aplicación.

1 Diseño de la base de datos y creación de repositorio en *GitHub*.

Hacer lo primero la estructura de la base de datos para conocer las relaciones que van a tener las diferentes tablas, ver qué columnas va a tener cada tabla y tener claro cómo empezar a desarrollar la aplicación. También se creará un repositorio para tener diferentes versiones y un lugar de copia de seguridad del código fuente.

2 Creación *frontend*, *backend* y página de inicio.

Se iniciará el proyecto con las tecnologías elegidas y se ejecutarán para comprobar que todo está bien instalado. También se hará una pequeña modificación para comenzar la comunicación entre ambos servicios y que muestre algunos alojamientos de ejemplo.

3 Inicio de sesión y registro.

Una vista para poder registrar un usuario en la página y poder usar todas las funcionalidades, también para iniciar sesión y poder recuperar la contraseña en caso de olvidarla.

4 Perfil del usuario y edición.

Una página para poder acceder a la información de tu cuenta y poder modificar diferentes datos.

5 Ver información sobre un alojamiento.

Una página para poder tener información más detallada sobre un alojamiento como quién es el dueño, qué valoraciones tiene, etc.

6 Ver tus alojamientos, crear nuevos y ver otros.

Una página para poder controlar los alojamientos que hayas creado para modificar alguna información como el precio, las imágenes u otros y desde aquí también se podrá añadir nuevas viviendas.

7 Implementación *GoogleMaps* para la búsqueda de lugares.

Como una de las características más importantes que tiene que tener la web es las ubicaciones, se ha añadido la *API* de *GoogleMaps* para auto-completar la búsqueda cuando escribamos una localización.

8 Favoritos y recomendaciones.

Posibilidad de añadir alojamientos a una lista de favoritos al tener alojamientos agregados aquí te avisará si su precio ha disminuido y otra página donde te recomendará nuevas experiencias basadas en tus reservas anteriores y alojamientos en favoritos.

9 Valoración de la estancia en un alojamiento.

Una vez terminada tu estancia en un alojamiento consideramos importante dar una opinión de cómo ha ido todo para que otros usuarios sepan qué se pueden encontrar al reservar la vivienda.

10 Perfil público de los usuarios y denunciar usuario.

Acceder al perfil de un usuario para obtener más información sobre ella para ver si se puede confiar, también una opción para denunciarlo en caso de que haya cometido alguna infracción.

11 Sistema de mensajes, emoticonos e imágenes.

La comunicación entre los usuarios es importante para obtener más detalles sobre un alojamiento, así que se ha creado un sistema de chat. Aunque no es en tiempo real, permite el intercambio de ideas para llegar a un acuerdo, con la posibilidad de enviar emoticonos e imágenes si es necesario más información sobre un alojamiento.

12 Reservar alojamientos, ver tus reservas y tus alojamientos.

Una página para poder seleccionar los días y personas que vais a querer arrendar el alojamiento y otra para tener un control sobre tus reservas hechas y las reservas que han hecho de tus alojamientos y ver en el estado que se encuentran.

13 Añadido *PayPal* como método de pago.

Una vez conocida tu reserva puedes confirmarla enviando el pago se ha añadido la *API* de *PayPal* para poder comprobar que la transacción se ha hecho correctamente.

14 Ganancias y generación en *PDF* y *.csv* de factura mensual.

Los hospedadores quizá ganen mucho dinero por lo que en esta página

les permite descargar con exactitud cuales han sido las ganancias de esos meses en formato *PDF* o *.csv* a elección.

15 Página de administración y control de usuarios.

Una página para que los dueños de la aplicación tengan mayor control de la misma y puedan restringir el acceso a los usuarios que infrinjan las normas.

- (G) **Documentación memoria.** Presente durante todo el desarrollo, que aporta una explicación de lo que hace y el contexto de la aplicación web.
- (H) **Corrección de errores.** Errores que se han ido solucionando durante el desarrollo de la aplicación web, que han aparecido debido al avance de las funcionalidades de la misma.
- (I) **Pruebas unitarias.** Pruebas realizadas para comprobar que todo el funcionamiento de la aplicación final va correctamente.

2.2.2. Asignación de tareas y estimación

Aquí se muestra con detalle los participantes de cada tarea y las fechas de la misma.

Tabla 2.5: Tarea A

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Diseñador gráfico - Frontend developer - Backend developer	17/10/2022	31/10/2022	2

Tabla 2.6: Tarea B

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Diseñador gráfico - Frontend developer - Backend developer	31/10/2022	14/11/2022	2

2.2.3. Planificación de tareas

En la figura 2.1 se muestra un diagrama de Gantt para tener una visión global de la evolución del proyecto y en el cual se representan todas las tareas y su duración. Éstas han sido explicadas en los apartados anteriores.

2.3. Gestión de recursos

En este apartado se van a especificar cuales han sido los diferentes recursos utilizados en el desarrollo.

2.3.1. Especificación de recursos

Humanos:

- Jefe de proyecto.
- Equipo de desarrollo: un diseñador gráfico, dos desarrolladores y un *tester*.

Materiales:

- 5 portátiles.
- Periféricos (ratones y auriculares).
- Alquiler de un despacho privado.

Económicos:

- Especificados en la tabla de coste final del apartado de presupuestos (Tabla 2.4).

2.3.2. Asignación de recursos

La asignación de los recursos que se han adquirido para el proyecto ha sido:

- Un portátil, un ratón y un auricular para cada miembro del equipo
- El uso del despacho y de las herramientas que tiene, a necesidad de los empleados.

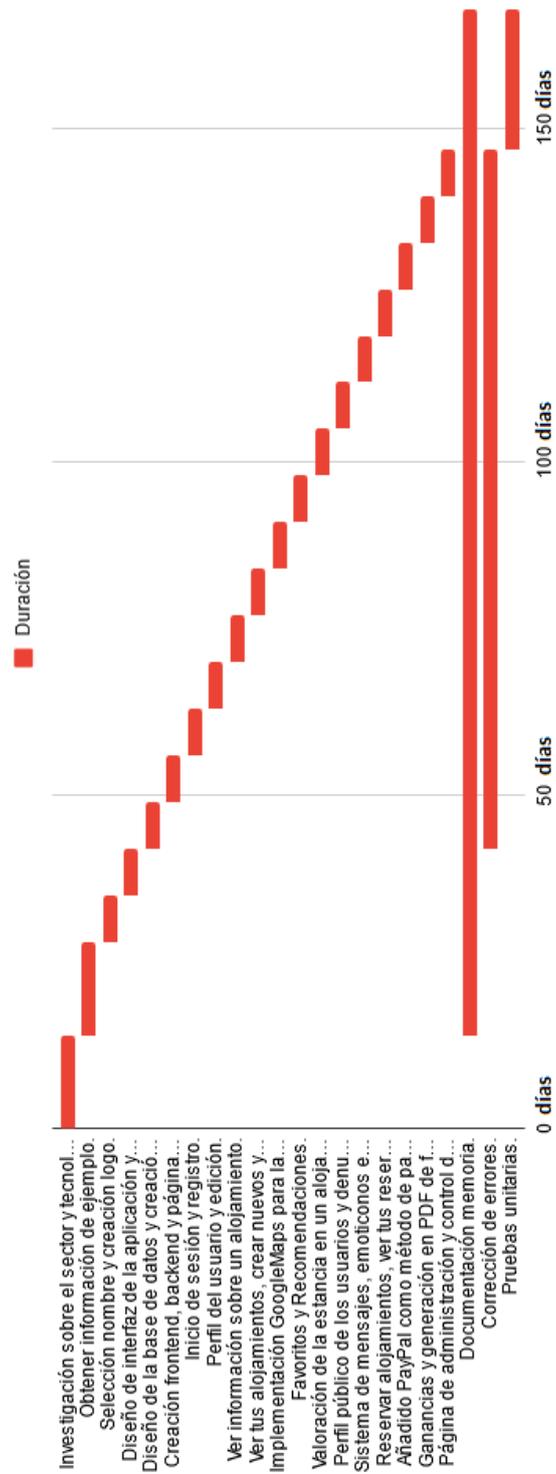


Figura 2.1: Gráfico Gantt

2.4. Gestión de riesgos

En este apartado se van a enumerar los diferentes riesgos que pueden aparecer durante el desarrollo de la aplicación web y cuál serían las posibles medidas a tomar.

2.4.1. Identificación de riesgos

Aquí se van a identificar los diferentes riesgos con sus posibles motivos, efectos y soluciones en caso de que aparecieran. Los datos sobre los mismos se recogen en las siguientes tablas.

- En la tabla 2.13 se muestran los riesgos relacionados con problemas de hardware.
- En la tabla 2.14 se muestran los riesgos relacionados con la pérdida de información.
- En la tabla 2.15 se muestran los riesgos relacionados con la baja productividad del equipo.
- En la tabla 2.16 se muestran los riesgos relacionados con el exceso de trabajo.
- En la tabla 2.17 se muestran los riesgos relacionados con la compenetración de los empleados.
- En la tabla 2.18 se muestran los riesgos relacionados con la dificultad de definir los requisitos del proyecto.
- En la tabla 2.19 se muestran los riesgos relacionados con los fallos en *GitHub*.
- En la tabla 2.20 se muestran los riesgos relacionados con los fallos en la *API* de *GoogleMaps*.
- En la tabla 2.21 se muestran los riesgos relacionados con los fallos en la *API* de *Paypal*.

2.4.2. Análisis de riesgos

Aquí se intentará ver el impacto que tienen todos los problemas anteriores sobre el proyecto, valorando cómo afectará en el coste y en el tiempo para completar el desarrollo y la probabilidad de que ocurran. Se recoge la información sobre este análisis en la tabla 2.22.

2.5. Legislación y normativa

- Ley de acceso electrónico de los ciudadanos a los Servicios Públicos <https://boe.es/buscar/pdf/2007/BOE-A-2007-12352-consolidado.pdf>
- Real Decreto sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-12699-consolidado.pdf>
- Real Decreto sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-12699-consolidado.pdf>
- Esquema nacional de interoperabilidad <https://www.boe.es/buscar/pdf/2010/BOE-A-2010-1331-consolidado.pdf>
- Esquema nacional de seguridad https://www.boe.es/diario_boe/txt.php?id=BOE-A-2022-7191
- Ley Orgánica de protección de datos personales y garantía de los derechos digitales <https://www.boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf>

La aplicación tiene en cuenta esta ley debido a que se almacena información personal sobre los usuarios y los alojamientos que publican, estos datos pueden ser modificados y eliminados desde el perfil y visualizados desde el perfil público. No se permite el registro a personas menores de edad.

- Reglamento General de Protección de Datos <https://www.boe.es/doue/2016/119/L00001-00088.pdf>
- Ley de conservación de datos relativos a las comunicaciones electrónicas y a las redes públicas de comunicaciones <https://www.boe.es/buscar/pdf/2007/BOE-A-2007-18243-consolidado.pdf>

La aplicación tiene en cuenta esta ley debido a que existen comunicaciones entre los usuarios para consultar más información sobre los alojamientos. Estos mensajes se almacena tanto el emisor como el receptor, fecha y hora de envío del mensaje y su contenido.

- Real Decreto por el que se adoptan medidas urgentes por razones de seguridad pública en materia de administración digital, contratación del sector público y

telecomunicaciones <https://www.boe.es/buscar/pdf/2019/BOE-A-2019-15790-consolidado.pdf>

- Real Decreto ley de servicios de pago y otras medidas urgentes en materia financiera <https://www.boe.es/boe/dias/2018/11/24/pdfs/BOE-A-2018-16036.pdf>

La aplicación tiene en cuenta esta ley debido a que funciona como intermediario entre los inquilinos y los hospedadores para brindar seguridad en los pagos.

- Ley de servicios de la sociedad de la información y de comercio electrónico <https://www.boe.es/buscar/pdf/2002/BOE-A-2002-13758-consolidado.pdf>

La aplicación tiene en cuenta esta ley debido a que obtiene beneficios de manera directa en forma de comisión de los alquileres de los alojamientos y el envío de correos electrónicos informativos que puede ser desactivados desde el perfil del usuario.

- Ley General de Telecomunicaciones <https://www.boe.es/buscar/pdf/2014/BOE-A-2014-4950-consolidado.pdf>
- ISO/IEC 29155-4:2016(en) Systems and software engineering — Information technology project performance benchmarking framework — Part 4: Guidance for data collection and maintenance

Tabla 2.7: Tarea C

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Diseñador gráfico - Frontend developer - Backend developer	14/11/2022	21/11/2022	1

Tabla 2.8: Tarea D

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Diseñador gráfico - Frontend developer - Backend developer	21/11/2022	28/11/2022	1

Tabla 2.9: Tarea E

Subtarea	Empleados	Fecha inicio	Fecha fin	Duración (semanas)
1	- Jefe de proyecto - Diseñador gráfico - Frontend developer - Backend developer - Tester	28/11/2022	05/12/2022	1
2	- Jefe de proyecto - Diseñador gráfico - Frontend developer - Backend developer - Tester	05/12/2022	12/12/2022	1
3	- Jefe de proyecto - Frontend developer - Backend developer - Tester	12/12/2022	19/12/2022	1
4	- Jefe de proyecto - Frontend developer - Backend developer - Tester	19/12/2022	26/12/2022	1
5	- Jefe de proyecto - Frontend developer - Backend developer - Tester	26/12/2022	02/01/2023	1
6	- Jefe de proyecto - Frontend developer - Backend developer - Tester	02/01/2023	09/01/2023	1
7	- Jefe de proyecto - Frontend developer - Backend developer - Tester	09/01/2023	16/01/2023	1
8	- Jefe de proyecto - Frontend developer - Backend developer - Tester	16/01/2023	23/01/2023	1
9	- Jefe de proyecto - Frontend developer - Backend developer - Tester	23/01/2023	30/01/2023	1

Subtarea	Empleados	Fecha inicio	Fecha fin	Duración (semanas)
10	- Jefe de proyecto - Frontend developer - Backend developer - Tester	30/01/2023	06/02/2023	1
11	- Jefe de proyecto - Frontend developer - Backend developer - Tester	06/02/2023	13/02/2023	1
12	- Jefe de proyecto - Frontend developer - Backend developer - Tester	13/02/2023	20/02/2023	1
13	- Jefe de proyecto - Frontend developer - Backend developer - Tester	20/02/2023	27/02/2023	1
14	- Jefe de proyecto - Frontend developer - Backend developer - Tester	27/02/2023	06/03/2023	1
15	- Jefe de proyecto - Frontend developer - Backend developer - Tester	06/03/2023	13/03/2023	1

Tabla 2.10: Tarea F

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Frontend developer - Backend developer - Tester	31/10/2022	03/04/2023	1

Tabla 2.11: Tarea G

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Frontend developer - Backend developer - Tester	28/11/2022	13/03/2023	1

Tabla 2.12: Tarea H

Empleados	Fecha inicio	Fecha fin	Duración (semanas)
- Jefe de proyecto - Frontend developer - Backend developer - Tester	13/03/2023	03/04/2023	3

Tabla 2.13: Riesgo 1

Riesgo 1	
Descripción	Problemas en el hardware.
Tipo	Externo
Motivo	El funcionamiento de los equipos no es el correcto, debido algún fallo físico en algún componente.
Efectos	Retraso en el desarrollo de la aplicación.
Solución	Reparar el equipo o adquirir otro dependiendo de la importancia de la reparación.

Tabla 2.14: Riesgo 2

Riesgo 2	
Descripción	Pérdida de información.
Tipo	<ul style="list-style-type: none"> ▪ Interno ▪ Externo
Motivo	<ul style="list-style-type: none"> ▪ Error al guardar el avance por algún empleado. ▪ Fallo en algún componente hardware.
Efectos	Retraso en el desarrollo de la aplicación.
Solución	<ul style="list-style-type: none"> ▪ Rehacer la parte del desarrollo perdido. ▪ Descubrir la razón para poder evitar que ocurra de nuevo. ▪ Hacer <i>backups</i> diariamente.

Tabla 2.15: Riesgo 3

Riesgo 3	
Descripción	Baja productividad en el equipo.
Tipo	Interno
Motivo	Algún empleado o los empleados no se encuentran motivados con el proyecto.
Efectos	<ul style="list-style-type: none"> ▪ Calidad del código deficiente. ▪ Retraso en el desarrollo de la aplicación.
Solución	<ul style="list-style-type: none"> ▪ Buscar motivación para los empleados. ▪ Contratar nuevos empleados.

Tabla 2.16: Riesgo 4

Riesgo 4	
Descripción	Cantidad de trabajo excesivo.
Tipo	Interno
Motivo	<ul style="list-style-type: none"> ▪ Planificación inicial muy optimista. ▪ Añadir nuevas características o muchos cambios no pensados en el planteamiento inicial.
Efectos	Retraso en el desarrollo de la aplicación.
Solución	Reorganizar las tareas y priorizar las funciones más importantes.

Tabla 2.17: Riesgo 5

Riesgo 5	
Descripción	Poca compenetración entre empleados.
Tipo	Interno
Motivo	El trabajo en equipo no es bueno debido a diferentes factores como puede ser la comunicación, escucha activa, etc.
Efectos	<ul style="list-style-type: none"> ▪ Calidad del código deficiente. ▪ Retraso en el desarrollo de la aplicación.
Solución	Fomentar la comunicación entre los empleados y otorgar más poder al Jefe de equipo.

Tabla 2.18: Riesgo 6

Riesgo 6	
Descripción	Dificultad para definir los requisitos del proyecto.
Tipo	Interno
Motivo	<ul style="list-style-type: none"> ▪ Discordancias a la hora de la definición de los requisitos. ▪ No se tiene en cuenta las ideas de los empleados.
Efectos	<ul style="list-style-type: none"> ▪ Descontento y desconfianza de los empleados. ▪ Retraso en el desarrollo de la aplicación.
Solución	Valorar la opinión de los empleados y hacerlos partícipes del proyecto.

Tabla 2.19: Riesgo 7

Riesgo 7	
Descripción	Fallo en <i>GitHub</i> .
Tipo	Externo
Motivo	El sistema está caído.
Efectos	Retraso en el desarrollo de la aplicación.
Solución	<ul style="list-style-type: none"> ▪ Esperar a que el sistema vuelva a reanudarse. ▪ Buscar una alternativa de uso o usar varios sistemas a la vez.

Tabla 2.20: Riesgo 8

Riesgo 8	
Descripción	Fallo con la <i>API</i> de <i>GoogleMaps</i> .
Tipo	Externo
Motivo	El sistema está caído.
Efectos	Retraso en el desarrollo de la aplicación.
Solución	<ul style="list-style-type: none"> ▪ Esperar a que el sistema vuelva a reanudarse. ▪ Reorganizar tareas y desarrollar otras que no dependan de la <i>API</i>.

Tabla 2.21: Riesgo 9

Riesgo 9	
Descripción	Fallo con la <i>API</i> de <i>PayPal</i> .
Tipo	Externo
Motivo	El sistema está caído.
Efectos	Retraso en el desarrollo de la aplicación.
Solución	<ul style="list-style-type: none"> ▪ Esperar a que el sistema vuelva a reanudarse. ▪ Reorganizar tareas y desarrollar otras que no dependan de la <i>API</i>.

Tabla 2.22: Análisis de riesgos

Riesgo	Incremento Coste	Incremento Tiempo	Probabilidad
1	Alto	Medio	Bajo
2	Alto	Alto	Medio
3	Medio	Medio	Medio
4	Medio	Medio	Medio
5	Bajo	Medio	Medio
6	Medio	Medio	Medio
7	Medio	Medio	Bajo
8	Bajo	Medio	Bajo
9	Bajo	Medio	Bajo

Capítulo 3

Solución

En esta sección se va a explicar detalladamente todo el proceso de desarrollo.

3.1. Descripción de la solución

Para la aplicación se han utilizado diferentes tecnologías, las cuales se han considerado las más adecuadas para el desarrollo de la misma, al tratarse de una aplicación web ésta estará dividida en la parte del servidor o *backend* donde se encuentra la lógica de la página y ofrece las respuestas necesarias y la parte del cliente o *frontend* con la que el usuario interactuará.

El proyecto no sólo cuenta con características para los clientes sino que también dispone de un sistema de administración para que los dueños puedan controlar a los usuarios que infrinjan las normas y poder penalizarlos, visualizar los países y ciudades más buscadas para extraer información de esos datos y añadir y fomentar a que haya usuarios que añadan sus alojamientos en esas ubicaciones.

También cuenta con otras muchas más herramientas que automatizan el trabajo de los dueños de la aplicación, como destacar los alojamientos que se han agregado nuevos, sus descuentos, informar por correo electrónico a los usuarios de una disminución del precio de un alojamiento, entre otros.

En el *backend* se lanzará un servidor utilizando *Nodejs* y *express* el cual atenderá a todas las peticiones que se hagan; la elección de esta tecnología sobre otras como puede ser *php*, es la facilidad que da al desarrollador al utilizarlo, ya que con unas pocas líneas de código ya estará escuchando peticiones, además es una tecnología que he usado anteriormente en otros proyectos y con buenos resultados. El *backend*

no sólo utiliza *Nodejs*, sino que utiliza otros módulos que ayudan a la persistencia de los datos, el funcionamiento de la aplicación y a ofrecer los servicios requeridos.

Se usará como almacenamiento de información una base de datos usando *MariaDB*; ésta nos permite hacer tablas relacionales que tienen conexiones entre ellas permitiéndonos una mayor facilidad de entendimiento. Se valoró usar otro sistema de base de datos como puede ser *MongoDB* ya que tiene una mayor rapidez y diversidad, pero la carencia de tener tablas relacionales hizo que finalmente se usara *MariaDB* como sistema de almacenamiento.

Otro módulo usado es *bcrypt* el cual permite encriptar las contraseñas de los usuarios mediante un valor llamado *salt* que se utiliza para generar el *hash* asociado a la contraseña. Éste se usa por encima de SHA-256 con *salt* que a pesar de que su velocidad sea mayor, las contraseñas generadas son menos robustas.

Se ha usado *Paypal* como método de pago, ya que nos ofrece un servicio muy completo de *sandbox* para poder probar el funcionamiento del sistema, una fácil transición para cuando se ponga la aplicación en producción y además permite el pago directamente con tarjeta de débito o crédito, *sofort* o desde el propio *Paypal* al cual puedes vincular tu cuenta bancaria.

En el *frontend* se utiliza *React* que nos facilita el diseño de interfaces, es utilizado por páginas muy conocidas como *Facebook* y además es actualmente un *framework* muy popular por lo que encontrar información y guías resulta bastante sencillo y es muy demandado en el sector laboral por lo que se tomó la decisión de usarlo, frente a otros *frameworks* como puede ser *Vue.js* o *Angular*.

Para *scrapear* *AirBnb* al ser una página dinámica no se puede extraer la información directamente por lo que hay que simularla primero y después sacar lo que nos interese. Para ello se utiliza *Selenium* y el driver de *Mozilla* llamado *Gecko* que nos permite simular en el navegador.

Como apoyo a este *framework* se ha utilizado *Bootstrap* que nos da un estilo agradable sobre todos los elementos básicos de una página como son los botones, las tablas, entre otros y es *responsive* por lo que no tenemos que preocuparnos a la hora de adaptar los componentes. La elección de *Bootstrap* sobre otras formas de estilo como pueden ser *Tailwind CSS*, *Materialize CSS* u otros, es que la selección de colores y formas es la que más se adecuaba al proyecto y a mis gustos.

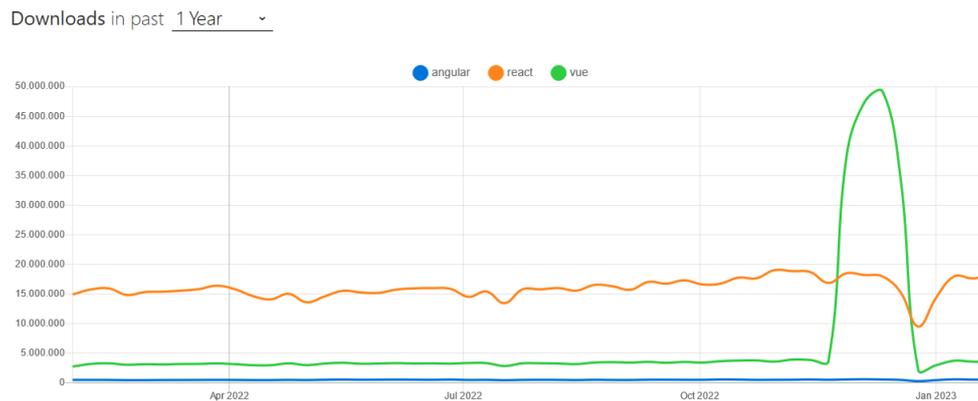


Figura 3.1: Popularidad de React vs Angular vs Vue.js

Fuente: <https://npmrends.com/angular-vs-react-vs-vue>

FontAwesome nos otorga una gran cantidad iconos para usar basados en *CSS* de forma gratuitos, simples y de código abierto y nos da más variedad de los que tiene *Bootstrap* en su librería.

Como tener control de las localizaciones y ubicaciones de los alojamientos y búsquedas es muy importante, se ha utilizado la *API* de *GoogleMaps* para completar automáticamente las ubicaciones que deseemos averiguar para las viviendas disponibles y para marcar el lugar exacto del domicilio. A pesar de que para usar el servicio es necesario vincular una tarjeta de crédito, las utilidades que ofrece son mucho mayores a las que ofrece otras *APIs* como puede ser la de *OpenStreetMap*.

3.2. El proceso de desarrollo

El desarrollo de la página web se ha realizado de forma incremental como todo el proyecto en sí, por lo que en cada tarea se han añadido uno o más requisitos a realizar, dependiendo de la valoración de dificultad de cada uno. En este apartado se determinarán más detalladamente cuáles han sido los requisitos que se han tenido en cuenta para la aplicación.

3.2.1. Análisis

Aquí se especifican cuáles van a ser las funcionalidades que va a tener la página web. Para ello se han especificado diferentes requisitos que se deben cumplir para que el funcionamiento de la aplicación sea el correcto.

Definición de requisitos

▪ Requisitos funcionales:

Requisito funcional 1

La página web tiene que poder ser accesible y utilizada desde cualquier dispositivo.

Requisito funcional 2

Interfaz principal, utilizable por todo el mundo.

Requisito funcional 3

Restricción de interfaces.

Requisito funcional 4

Interfaz de registro, una vista para poder registrar una cuenta en la aplicación.

Requisito funcional 5

Interfaz de inicio de sesión, una vista para poder iniciar sesión en la aplicación y poder utilizar las demás vistas.

Requisito funcional 6

Enviar correos electrónicos al correo de los usuarios registrados.

Requisito funcional 7

Interfaz de validación de cuenta, una vez registrada una cuenta se enviará un código al correo para validar la cuenta.

Requisito funcional 8

Interfaz para ver las condiciones de uso y privacidad de la página.

Requisito funcional 9

Una base de datos para la consistencia de la información.

Requisito funcional 10

Interfaz para recuperar contraseña, en caso de que el usuario se haya olvidado de ella.

Requisito funcional 11

Poder ordenar los alojamientos mediante la fecha de creación, la relevancia y el precio del mismo

Requisito funcional 12

Filtros avanzados para mejorar tu búsqueda de un alojamiento.

Requisito funcional 13

Interfaz de favoritos, una lista de alojamientos que has ido añadiendo a favoritos.

Requisito funcional 14

Interfaz de recomendados, una lista de alojamientos que se basará en tus favoritos y en reservas anteriores.

Requisito funcional 15

Búsqueda por ubicación, poder buscar los alojamientos de un país entero, ciudad, comunidad, etc.

Requisito funcional 16

Interfaz de perfil del usuario, para ver tu información y modificarla.

Requisito funcional 17

Interfaz de tus alojamientos, para ver tus alojamientos, modificarlos y crear nuevos.

Requisito funcional 18

Interfaz de tus reservas, para ver las reservas que has hecho, las que han hecho de tus alojamientos.

Requisito funcional 19

Poder descargar mensualmente las ganancias obtenidas por el alquiler de tus alojamientos.

Requisito funcional 20

Interfaz de valoraciones, donde puedes ver las valoraciones hechas a usuarios y alojamientos y las recibidas sobre tus alojamientos y de usuarios.

Requisito funcional 21

Interfaz de chat, un sistema de mensajes para poder comunicarte con otros usuarios.

Requisito funcional 22

Centro de ayuda, donde se resolverán las dudas más comunes que te puedan aparecer.

Requisito funcional 23

Poder seleccionar un nuevo idioma.

Requisito funcional 24

Interfaz de administración.

Requisito funcional 25

Control de usuarios.

Requisito funcional 26

Interfaz de perfil público de usuario.

Requisito funcional 27

Poder denunciar usuarios.

Requisito funcional 28

Interfaz de vista de alojamiento con toda la información del mismo.

Requisito funcional 29

Poder copiar el *link* del alojamiento para compartirlo.

Requisito funcional 30

Mostrar alojamiento como visto, si ya has accedido a su página.

Requisito funcional 31

Interfaz de reservar un alojamiento.

Requisito funcional 32

Valorar la estancia de un alojamiento.

Requisito funcional 33

Valorar la estancia de un cliente.

■ Requisitos no funcionales:**Requisito no funcional 1**

El sistema debe avisar en caso de que alguna de las herramientas falle.

Requisito no funcional 2

El sistema debe notificar cuál es la razón por la que no pueda iniciarse.

Requisito no funcional 3

El sistema debe tener poco tiempo de respuesta.

Requisito no funcional 4

El sistema debe poder iniciarse y visualizarse en cualquier sistema operativo.

Requisito no funcional 5

El sistema debe ser escalable.

Requisito no funcional 6

El sistema se debe poder mantener fácilmente.

Requisito no funcional 7

El sistema debe utilizar los recursos adecuadamente para no sobrecargarlo.

Requisito no funcional 8

El sistema debe ser seguro.

Requisito no funcional 9

El sistema debe tener consistencia en los datos.

Especificación de requisitos

Se han diseñado 21 casos de uso, recogidos en las tablas 3.1-3.21.

Tabla 3.1: Caso de uso 1

Caso de uso 1	
Descripción	Diseño <i>responsive</i> .
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y accede a la página principal.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial con diferentes dispositivos. 2. Puede ver el diseño e interactuar con los elementos correctamente.
Postcondiciones	El diseño se adapta según la resolución de cada dispositivo.
Excepciones	
Comentarios	

Tabla 3.2: Caso de uso 2

Caso de uso 2	
Descripción	Restricción de interfaces.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none">1. Usuario accede a la página inicial sin haber iniciado sesión.2. Pulsa en el botón de favoritos.
Postcondiciones	Aparece una ventana para iniciar sesión.
Excepciones	El usuario debe iniciar sesión para acceder a la página de favoritos.
Comentarios	Al no haber iniciado sesión el sistema no permite acceder a la página de favoritos.

Tabla 3.3: Caso de uso 3

Caso de uso 3	
Descripción	Interfaz de registro.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial sin haber iniciado sesión. 2. Pulsa en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Regístrate'. 4. Se abre una nueva ventana, donde tiene que rellenar el formulario. 5. Una vez rellenado, pulsar el botón de registrar cuenta.
Postcondiciones	Aparece una alerta de que debes validar tu cuenta en el correo electrónico del registro.
Excepciones	<ol style="list-style-type: none"> 1. Nombre vacío: El sistema informa de que es necesario rellenar este campo. 2. Nombre con números: El sistema informa de que no es posible colocar números en este campo. 3. Apellidos vacío: El sistema informa de que es necesario rellenar este campo. 4. Apellidos con números: El sistema informa de que no es posible colocar números en este campo. 5. Correo electrónico vacío: El sistema informa de que es necesario rellenar este campo. 6. Correo electrónico formato inválido: El sistema informa de que es necesario poner un correo electrónico válido.

Excepciones (cont.)	<ol style="list-style-type: none">7. Fecha nacimiento vacío: El sistema informa de que es necesario rellenar este campo8. Fecha menor a 18 años: El sistema informa de que el usuario debe tener al menos 18 años.9. Contraseña vacío: El sistema informa de que es necesario rellenar este campo.10. Contraseña con menos de cinco caracteres de longitud: El sistema informa de que la longitud debe ser mayor.11. Las contraseñas no coinciden: El sistema informa que las contraseñas deben coincidir.12. Teléfono vacío: El sistema informa de que es necesario rellenar este campo.13. Teléfono con letras: El sistema informa de que el teléfono debe ser un número.14. No aceptar términos y condiciones: El sistema informa de que es necesario aceptarlos15. No aceptar el aviso de privacidad: El sistema informa de que es necesario aceptarlo.
Comentarios	Se debe rellenar tanta información para que se te pueda identificar en la página.

Tabla 3.4: Caso de uso 4

Caso de uso 4	
Descripción	Interfaz de inicio de sesión.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial sin haber iniciado sesión. 2. Pulsa en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Inicia sesión'. 4. Se abre una nueva ventana, donde hay que rellenar el formulario. 5. Una vez rellenado, pulsar el botón de iniciar sesión.
Postcondiciones	Aparece una alerta de que has iniciado sesión correctamente.
Excepciones	<ol style="list-style-type: none"> 1. Correo electrónico vacío: El sistema informa de que es necesario rellenar este campo. 2. Correo electrónico formato inválido: El sistema informa de que es necesario poner un correo electrónico válido. 3. Correo electrónico inválido: El sistema informa de que los datos introducidos son incorrectos. 4. Contraseña vacío: El sistema informa de que es necesario rellenar este campo. 5. Contraseña con menos de cinco caracteres de longitud: El sistema informa de que la longitud debe ser mayor. 6. Contraseña inválida: El sistema informa de que los datos introducidos son incorrectos.
Comentarios	Se debe rellenar tanta información para que se te pueda identificar en la página.

Tabla 3.5: Caso de uso 5

Caso de uso 5	
Descripción	Validar cuenta.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se acaba de registrar la cuenta.
Flujo básico	<ol style="list-style-type: none"> 1. Una vez registrado una cuenta en la página y ver la alerta de entrar al correo. 2. Acceder a tu <i>email</i> y abrir el correo recibido de la aplicación. 3. Pulsar en el <i>link</i> que hay en el mensaje. 4. Te abrirá la página de la aplicación en otra pestaña.
Postcondiciones	Aparece una alerta de que tu cuenta ha sido validada correctamente.
Excepciones	<ol style="list-style-type: none"> 1. La cuenta ya ha sido validada: El <i>link</i> ha caducado ya que la cuenta ya ha sido validada.
Comentarios	Se debe validar la cuenta mediante el correo para que los usuarios se registren con un <i>email</i> válido.

Tabla 3.6: Caso de uso 6

Caso de uso 6	
Descripción	Ver términos y condiciones y privacidad.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar abajo de todo al texto de 'Privacidad' o 'Condiciones'.
Postcondiciones	Te redireccionará a la página correspondiente y te mostrará la información requerida.
Excepciones	
Comentarios	

Tabla 3.7: Caso de uso 7

Caso de uso 7	
Descripción	Recuperar contraseñas.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Inicia sesión'. 4. Pulsar donde pone '¿Has olvidado tu contraseña?'. 5. Te redireccionará a otra página donde se debe escribir el correo electrónico vinculado a la cuenta. 6. Pulsar el botón de enviar.
Postcondiciones	El sistema te mostrará una alerta de que se ha enviado un <i>email</i> a tu correo electrónico y te redireccionará a la página inicial.
Excepciones	<ol style="list-style-type: none"> 1. Correo electrónico vacío: El sistema informa de que es necesario rellenar este campo. 2. Correo electrónico formato inválido: El sistema informa de que es necesario poner un correo electrónico válido. 3. Correo electrónico inexistente: El sistema informa de que ese correo electrónico no se encuentra registrado en la aplicación.
Comentarios	

Tabla 3.8: Caso de uso 8

Caso de uso 8	
Descripción	Ordenar alojamientos.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en alguno de los botones que están al lado de 'Ordenar por'.
Postcondiciones	El orden de los alojamientos se actualizará según la opción que hayas seleccionado.
Excepciones	
Comentarios	

Tabla 3.9: Caso de uso 9

Caso de uso 9	
Descripción	Filtros avanzados.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón que pone 'Otros filtros'. 3. Seleccionar los filtros que se desee. 4. Pulsar en el botón que pone 'Mostrar x alojamientos'.
Postcondiciones	Los alojamientos se actualizarán según los filtros que hayas seleccionado.
Excepciones	
Comentarios	

Tabla 3.10: Caso de uso 10

Caso de uso 10	
Descripción	Interfaz de favoritos.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none">1. Usuario accede a la página inicial.2. Pulsar en el botón de 'Favoritos'.3. Te redireccionará a la página de favoritos.
Postcondiciones	Te mostrará los alojamientos añadidos a favoritos.
Excepciones	<ol style="list-style-type: none">1. No has iniciado sesión: El sistema te abrirá la ventana para iniciar sesión.2. No aparece ningún alojamiento: No has añadido ningún alojamiento todavía.
Comentarios	

Tabla 3.11: Caso de uso 11

Caso de uso 10	
Descripción	Interfaz de recomendados.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón de 'Recomendados'. 3. Te redireccionará a la página de recomendados.
Postcondiciones	Te mostrará los alojamientos recomendados pensados por el sistema.
Excepciones	<ol style="list-style-type: none"> 1. No has iniciado sesión: El sistema te abrirá la ventana para iniciar sesión. 2. No aparece ningún alojamiento: Las recomendaciones están basadas en tus reservas anteriores y alojamientos añadidos a favoritos, así que es necesario tener alguna de esas dos cosas.
Comentarios	

Tabla 3.12: Caso de uso 12

Caso de uso 12	
Descripción	Búsqueda por ubicación.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar la celda donde pone 'Escribe un lugar'. 3. Escribir un lugar y seleccionar uno de la lista para que se auto complete la ubicación.
Postcondiciones	Te mostrará los alojamientos que están ubicados en esa localización.
Excepciones	<ol style="list-style-type: none"> 1. No aparece ningún alojamiento: No hay ningún alojamiento en esa ubicación.
Comentarios	

Tabla 3.13: Caso de uso 13

Caso de uso 13	
Descripción	Modificar perfil.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se ha iniciado sesión.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Perfil'. 4. Seleccionar una opción de la tabla que tenga la flecha hacia la derecha. 5. Se abrirá un formulario para modificar la opción. 6. Si se ha editado aparecerá otra celda para escribir la contraseña. 7. Pulsar el botón de modificar.
Postcondiciones	El sistema te mostrará un aviso de que la opción se ha modificado.
Excepciones	<ol style="list-style-type: none"> 1. Si no cambiamos la opción a editar no aparecerá la celda para añadir la contraseña. 2. Contraseña vacío: El sistema informa de que es necesario rellenar este campo. 3. Contraseña inválida: El sistema informa de que los datos introducidos son incorrectos.
Comentarios	El formulario para editar una opción u otra puede cambiar dependiendo de lo que queramos modificar, pero el funcionamiento general es el mismo.

Tabla 3.14: Caso de uso 14

Caso de uso 14	
Descripción	Modificar alojamiento.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento, se ha iniciado sesión y tienes un alojamiento creado.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Mis alojamientos'. 4. Seleccionar un alojamiento de la tabla. 5. Se abrirá un menú con las diferentes opciones a editar. 6. Seleccionar una opción y se abrirá un formulario para modificarlo. 7. Si se ha editado la opción aparecerá el botón para confirmar la modificación. 8. Pulsar el botón de modificar.
Postcondiciones	El sistema te mostrará un aviso de que la opción se ha modificado.
Excepciones	
Comentarios	El formulario para editar una opción u otra puede cambiar dependiendo de lo que queramos modificar, pero el funcionamiento general es el mismo.

Tabla 3.15: Caso de uso 15

Caso de uso 15	
Descripción	Crear nuevo alojamiento.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se ha iniciado sesión.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Mis alojamientos'. 4. Pulsar el botón de 'Crear nuevo'. 5. Se abrirá un formulario a rellenar. 6. Rellenar las opciones del formulario. 7. Pulsar el botón de 'Crear alojamiento'.
Postcondiciones	El sistema te mostrará un aviso de que el alojamiento se ha creado.
Excepciones	<ol style="list-style-type: none"> 1. Título vacío: El sistema informa de que es necesario rellenar este campo. 2. Título menor a 20 caracteres: El sistema informa de que el número de caracteres debe ser mayor. 3. Descripción vacío: El sistema informa de que es necesario rellenar este campo. 4. Descripción menor a 100 caracteres: El sistema informa de que el número de caracteres debe ser mayor. 5. Ubicación vacío: El sistema informa de que es necesario rellenar este campo. 6. Lugar exacto sin seleccionar: El sistema informa de que es necesario seleccionar el lugar exacto del alojamiento. 7. Imágenes vacío: El sistema informa de que es necesario añadir al menos una imagen.
Comentarios	

Tabla 3.16: Caso de uso 16

Caso de uso 16	
Descripción	Descargar ganancias.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se ha iniciado sesión.
Flujo básico	<ol style="list-style-type: none">1. Usuario accede a la página inicial.2. Pulsar en el botón a la derecha de la pantalla.3. Se abrirá un menú y pulsar en la opción de 'Mis reservas'.4. Pulsar la opción de 'Mis ganancias'.5. Pulsar la opción de 'Descargar .csv' o 'Descargar .pdf'.
Postcondiciones	El sistema te descargará el documento en el formato seleccionado.
Excepciones	<ol style="list-style-type: none">1. No sale ningún botón para descargar: No has tenido ningún alojamiento reservado durante ese mes.
Comentarios	

Tabla 3.17: Caso de uso 17

Caso de uso 17	
Descripción	Enviar mensaje por chat.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se ha iniciado sesión.
Flujo básico	<ol style="list-style-type: none">1. Usuario accede a la página inicial.2. Pulsar en el botón a la derecha de la pantalla.3. Se abrirá un menú y pulsar en la opción de 'Mis mensajes'.4. Seleccionar una conversación con un usuario.5. Se abrirá el chat con los mensajes anteriores.6. Pulsar en la celda donde pone 'Escribe tu mensaje aquí!'.7. Escribir el mensaje y pulsar <i>intro</i>.
Postcondiciones	El nuevo mensaje aparecerá dentro de la caja de mensajes.
Excepciones	<ol style="list-style-type: none">1. No sale ninguna conversación: No has iniciado ninguna conversación con ningún usuario.
Comentarios	Enviar un emoticono es igual que un mensaje pero seleccionándolo desde el icono de una cara.

Tabla 3.18: Caso de uso 18

Caso de uso 18	
Descripción	Enviar imagen por chat.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se ha iniciado sesión.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón a la derecha de la pantalla. 3. Se abre un menú y pulsar en la opción de 'Mis mensajes'. 4. Seleccionar una conversación con un usuario. 5. Se abrirá el chat con los mensajes anteriores. 6. Pulsar el icono con la cámara. 7. Se abrirá un menú para seleccionar una imagen. 8. Seleccionar la imagen y pulsar el botón de 'Abrir'.
Postcondiciones	La imagen aparecerá dentro de la caja de mensajes.
Excepciones	<ol style="list-style-type: none"> 1. No sale ninguna conversación: No has iniciado ninguna conversación con ningún usuario. 2. Formato seleccionado: Si la imagen no tiene un formato correcto el sistema te informará de ello.
Comentarios	

Tabla 3.19: Caso de uso 19

Caso de uso 19	
Descripción	Control de usuarios.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento, se ha iniciado sesión y tiene rol de administrador.
Flujo básico	<ol style="list-style-type: none"> 1. Usuario accede a la página inicial. 2. Pulsar en el botón a la derecha de la pantalla que tiene dos herramientas. 3. Se abrirá un menú de administración y pulsar el cuadro que pone usuarios. 4. Se mostrará una tabla con todos los usuarios registrados en la aplicación. 5. En la columna de opciones a la derecha del todo se puede bloquear, desbloquear, promocionar a administrador, quitar administrador y verificar cuenta. 6. Pulsar una de las opciones
Postcondiciones	El sistema te mostrará una alerta de que el usuario se ha modificado correctamente.
Excepciones	<ol style="list-style-type: none"> 1. Rol insuficiente: El sistema no cambiará nada.
Comentarios	

Tabla 3.20: Caso de uso 20

Caso de uso 20	
Descripción	Denunciar usuario.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento y se ha iniciado sesión.
Flujo básico	<ol style="list-style-type: none">1. Usuario accede a la página inicial.2. Pulsar sobre algún alojamiento.3. Ir al lugar donde está la información del hospedador y pulsar el botón 'Ir al perfil'.4. Ir abajo del todo de la página y pulsar el botón de 'Denunciar este perfil'.5. Rellenar la celda del mensaje.6. Pulsar el botón de 'Enviar'.
Postcondiciones	El sistema te mostrará una alerta de que la denuncia se ha enviado con éxito.
Excepciones	<ol style="list-style-type: none">1. No haber iniciado sesión: Se abrirá la ventana de iniciar sesión.2. Mensaje vacío: El sistema informa de que es necesario rellenar este campo.3. Mensaje menor a 20 caracteres: El sistema informa de que el número de caracteres debe ser mayor.
Comentarios	

Tabla 3.21: Caso de uso 21

Caso de uso 21	
Descripción	Valorar un alojamiento.
Actores	Usuario
Precondiciones	El sistema está en funcionamiento, se ha iniciado sesión y se ha reservado un alojamiento.
Flujo básico	<ol style="list-style-type: none">1. Usuario accede a la página inicial.2. Pulsar en el botón a la derecha de la pantalla.3. Se abre un menú y pulsar en la opción de 'Mis reservas'.4. Pulsar el botón de 'Valorar' a la derecha del todo.5. Rellenar el formulario y pulsar en 'Enviar valoración'.
Postcondiciones	El sistema te mostrará una alerta de que la valoración se ha enviado con éxito.
Excepciones	<ol style="list-style-type: none">1. Mensaje vacío: El sistema informa de que es necesario rellenar este campo.2. Mensaje menor a 25 caracteres: El sistema informa de que el número de caracteres debe ser mayor.
Comentarios	Para valorar un cliente, hay que hacer los mismos pasos pero en el menú de 'Mis alojamientos'.

3.2.2. Diseño

Diseño de sistema

Lo primero que se va a mostrar es cómo interactúan los diferentes componentes desde el usuario, el hardware y software indicando en la siguiente figura todos los elementos que participan y las tecnologías usadas en cada grupo.

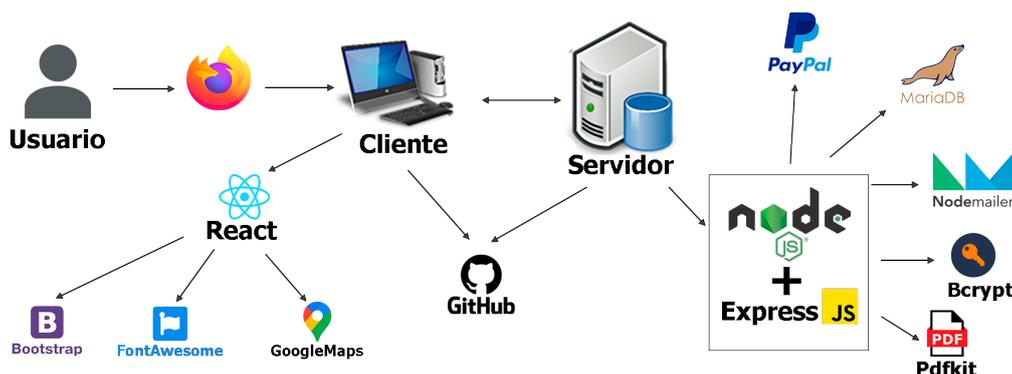


Figura 3.2: Tecnologías usadas en el TFG

El sistema está compuesto por la parte del cliente y la parte del servidor y ambos se comunican para ofrecer los servicios al usuario.

En el cliente se utiliza:

- **React**, es una biblioteca de *Javascript* de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones, [11].
- **Bootstrap**, es un conjunto de herramientas para diseños de sitios y páginas web, nos facilita mucho el trabajo, ya que automatiza mucho del trabajo que realizamos al estilizar y decorar los elementos web, [12].
- **FontAwesome**, es un conjunto de herramientas de fuentes e iconos basados en *CSS* y *Less* con una gran variedad de elementos. Nos permite utilizar iconos para tener un diseño más agradable en nuestra aplicación web, [7].
- **GoogleMaps**, es una herramienta que nos permite insertar en nuestra web diferentes funciones para nuestro uso desde desplegar mapas, auto completado de localizaciones, [13].

En el servidor funciona como una *API* suministrando la información necesaria al cliente y utiliza para su funcionamiento las siguientes herramientas:

- **Nodejs**, es un entorno de trabajo en *JavaScript* diseñado para la capa de servidor y multiplataforma, [9].
- **Express**, es un entorno de trabajo para aplicaciones web para *Nodejs* y que nos permite desarrollar en nuestro caso la *API* para satisfacer las peticiones del cliente [20].
- **MariaDB**, es un servidor que es de los más populares para bases de datos relacionales, [8].
- **Paypal**, es un servicio que te permite pagar, enviar dinero y aceptar pagos sin tener que introducir datos financieros con diferentes posibilidades para realizar los pagos, [10].
- **NodeMailer**, es un módulo para *Nodejs* que nos permite enviar correos electrónicos a través de una cuenta de *email* [21].
- **Bcrypt**, es un módulo para *Nodejs* que nos permite encriptar las contraseñas y compararlas para el inicio de sesión de la página [19].
- **Pdfkit**, es un módulo para *Nodejs* que nos permite crear archivos *.pdf* desde la parte del servidor [22].

Otros elementos que son:

- **Usuario**, es el actor que va a utilizar la página web a través de diferentes herramientas.
- **Mozilla Firefox**, es el navegador que el actor utiliza para acceder a la página web y poder navegar por ella.
- **GitHub**, es un sistema de control de versiones que nos permite almacenar nuestro código en sus servidores en el cual se puede ver toda la evolución del proyecto [26].

Diseño detallado

Para el proyecto se ha utilizado el patrón de diseño MVC conocido como Modelo-Vista-Controlador que contiene interfaces de usuario, datos y lógica de control. Como se puede ver en la siguiente Figura 3.3 el sistema está dividido en la carpeta de *client* y *server*.

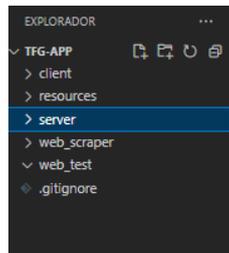


Figura 3.3: Modelo MVC en el proyecto

Para la persistencia de los datos la aplicación utiliza una base de datos que es relacional por lo que las tablas están conectadas entre ellas y nos permite agregar, eliminar, modificar, organizar y ver los datos de diferentes manera, como se puede ver en el diseño de la misma en el siguiente diagrama de tablas. En la figura no están representadas todas las columnas de cada una de las tablas para que no quede sobrecargada la imagen.

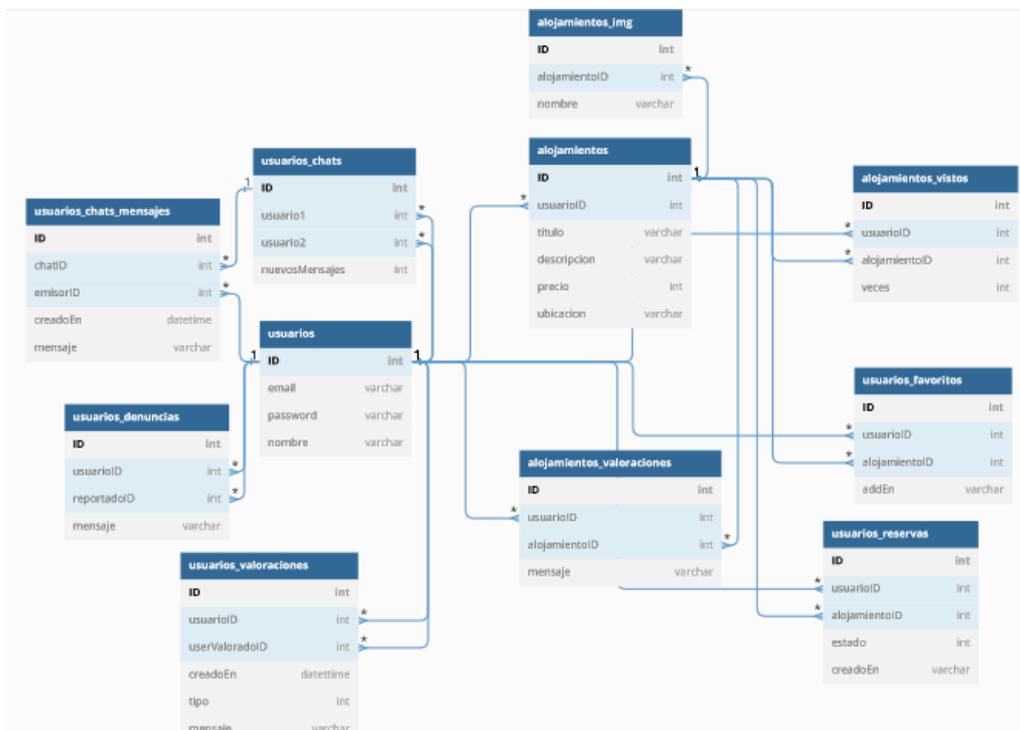


Figura 3.4: Diagrama de tablas de la base de datos

Este tipo de diseño como se ve en la Figura 3.4 nos permite de una manera simple ver cómo está compuesta toda la estructura de la base de datos.

La interfaz y estructura de la página también ha cambiado desde el inicio del proyecto, ya que algunos elementos tuvieron una revisión sobre su colocación y diseño. Esto se ha basado en la organización que tienen otras aplicaciones parecidas y en la estructura lógica de una página web con el encabezado, contenido y pie de páginas que son las partes básicas de cualquier página.

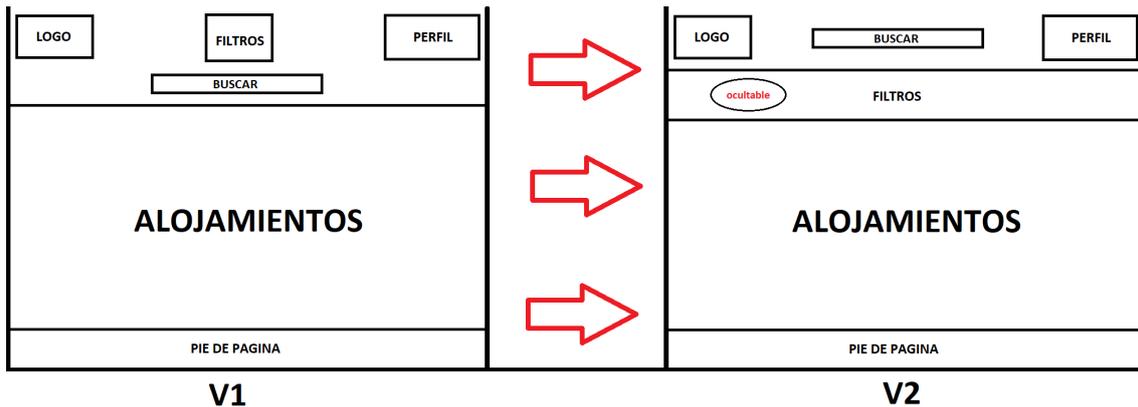


Figura 3.5: Cambios en la estructura página web

Como se puede ver en la figura 3.5 en el diseño inicial los filtros estaban unidos al encabezado con los demás elementos; se tomó la decisión de separarlos en otro apartado para que las vistas que no fueran necesarios se mantengan ocultos y los otros elementos necesarios para la navegación sigan visibles.

El intercambio de información entre *backend* y *frontend* se realiza mediante estructuras de datos en formato *JSON* que nos permiten la comunicación de manera fácil y sencilla. Un ejemplo es el siguiente: el cliente entra desde el navegador por primera vez a la página web; éste hace un *fetch* al *backend* en busca de la información de los alojamientos, si la petición es correcta éste devolverá el código de estado 200 que equivale a que la acción se ha completado satisfactoriamente y, en formato *JSON*, la información de los alojamientos como se puede ver en la figura 3.6

```
res.status(200).json({ respuesta: 'correcto', alojamientos: result });
```

Figura 3.6: Respuesta del backend en formato JSON

Las peticiones que se realizan desde el *frontend* pueden ser de dos tipos *GET* o *POST*, según el tipo de la petición el servidor responderá con una información u otra.

Existen muchos códigos de respuesta [16] para las peticiones. En la tabla 3.22 se recogen los más utilizados en la aplicación.

Tabla 3.22: Códigos de error HTML

Código	Descripción
200	Acción completada con éxito.
201	Éxito después de una petición <i>POST</i> .
401	Datos incorrectos para acceder a un archivo.
404	Recurso no encontrado.
500	Error en el servidor del código o del programa.

El *backend* no sólo envía datos en formato *JSON* también es necesario que mande las imágenes de los alojamientos o la fotos de los perfiles de otros usuarios. Esto se hace primero buscando en los archivos del servidor y luego se llevan al *frontend* para que sean visibles; para ello se indica que lo que se va a enviar como contenido es una *'image/formato'* como se puede ver en la figura 3.7

```
const imagen = result[0].imgPerfil;

fs.readFile('./imagenes/perfil/' + imagen, function (err, file) {

  if (err) {
    res.status(500).json({ respuesta: 'err_file' });
    return;
  }

  res.set({ 'Content-Type': 'image/jpg' });
  res.end(file);
});
```

Figura 3.7: Respuesta de imagen del backend

Como vemos, en la segunda línea de código utilizando el módulo *FileSystem* que viene ya incluido en *Nodejs* lo usamos para buscar en la carpeta *'/imagenes/perfil/'* la fotografía que necesitamos, si la imagen no se encuentra o ha ocurrido un error devolverá el error y si no, enviará la imagen colocando el contenido de que es una imagen explicado anteriormente. El mismo sistema es utilizado para enviar archivos pero el contenido que colocamos varía en el caso de un *PDF*; el *'Content-Type'* sería *'application/csv'*.

Para la persistencia de la sesión de usuario se utiliza *JWT* y *cookies*. El usuario accede al formulario para iniciar sesión en la aplicación, escribe sus datos y envía la

información al *backend* para comprobar si es correcta; en caso de que sea correcta se genera un *JWT* con el número de identificación que tiene el usuario en la base de datos y un secreto que está establecido por la aplicación. De esta forma si el token es modificado o se ha insertado alguna credencial, se puede verificar que la firma no es correcta y se deniega la solicitud.

Este *token* se almacena en las *cookies* del navegador como podemos ver en la siguiente figura y se mantienen en el navegador aunque el usuario cierre la pestaña. Este tipo de *cookies* se llaman *cookies* permanentes y expiran en una fecha específica o tras un periodo de tiempo determinado.



Figura 3.8: JWT almacenado en el navegador

También se especifica el atributo *cookie* de *HttpOnly* para que el valor no pueda ser accedido a través de *JavaScript*; esto ayuda a evitar ataques de secuestro de sesión y *XSS*.

Y por último, el *token* se envía en la cabecera de todas las peticiones que se realicen al *backend* para validar su integridad y es comprobado mediante el método *comprobarToken* que se ve en la figura 3.9

```
const comprobarToken = (req, __, next) => {  
  const token = req.cookies.token;  
  
  if (token) {  
    try {  
      const decoded = jwt.verify(token, cookie_secret);  
      req.userId = decoded.id;  
  
      mysql.query('UPDATE usuarios SET ultimaConexion=NOW() WHERE ID=? LIMIT 1', decoded.id);  
    } catch (err) {  
      console.log(err);  
    }  
  }  
  
  next();  
};
```

Figura 3.9: Validación del JWT

3.2.3. Implementación

Lenguajes:

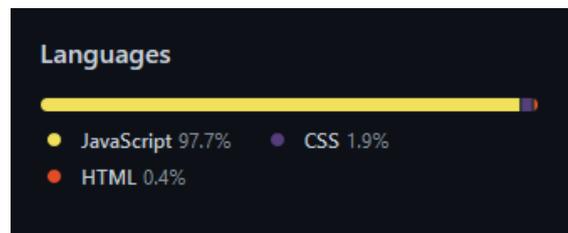


Figura 3.10: Lenguajes usados en el proyecto según GitHub

- **JavaScript**, es un lenguaje de programación interpretado que es utilizado en toda la aplicación como el lenguaje principal tanto en el *frontend* con *React* y en el *backend* con *Nodejs* ya que estos están basados en él. Es uno de los lenguajes de programación más utilizados y es muy versátil ya que es débilmente *tipado* y no es necesario especificar los tipos de las variables, funciones, etc.
- **HTML**, es un lenguaje de marcado para la creación de páginas web. Usado en la web para mostrar al usuario las diferentes vistas y elementos que tiene la aplicación.
- **CSS**, es un lenguaje de diseño gráfico para definir la presentación de un documento en nuestro caso de un documento *HTML*, que mediante éste podemos definir diferentes estilos para que nuestra web sea más llamativa.

- **SQL**, es un lenguaje de dominio específico que en nuestro caso nos permite administrar y recuperar información de nuestro sistema de base de datos relacional.

Herramientas:

- **VisualStudioCode**, es un editor de código fuente, es la herramienta con la que se ha programado toda la aplicación.
- **Xampp**, es un sistema de gestión de bases de datos que nos permite de forma interactiva ver las bases de datos, tablas, columnas, etc de manera muy rápida y sencilla.
- **HeidiSQL**, es un sistema de gestión de bases de datos que permite conectarse a *MySQL*; no dispone de una interfaz gráfica tan agradable como *Xampp*, pero a pesar de esto se ha decidido utilizar *HeidiSQL*, ya su interfaz no va a ser utilizada, porque todas las acciones se va a realizar mediante sentencias *SQL* desde la parte del servidor y la creación de la base de datos importando el archivo *.sql*. *HeidiSQL* dispone de otras ventajas como que es portable, por lo que no necesita instalación y su peso en disco es mucho menor.
- **Adobe Photoshop**, es un editor de imágenes con el que se ha creado el logotipo y las demás imágenes que se han necesitado para la aplicación o la memoria.
- **Google Sheets**, es un editor de hojas de calculo de *Google* en línea que nos ha permitido diseñar el diagrama de Grantt de una manera sencilla.
- **Prefijos numero teléfono**, es una lista de los diferentes prefijos de todos los países del mundo en formato *JSON* [24].
- **BoredHumans**, es una *IA* que nos permite generar caras humanas de gente que no existe. Esta herramienta se ha utilizado para añadir imágenes de perfil a algunos de los usuarios de la aplicación [2].
- **DB Diagram**, es una herramienta en línea que nos permite diseñar la estructura de bases de datos relacionales [5].
- **DatePicker**, es un selector de fechas que nos permite seleccionar todas las fechas entre dos fechas dadas una de inicio y otra de fin, también nos permite restringir días para que no sean seleccionables. Se utiliza a la hora de hacer las reservas y todo esto ha sido posible gracias a utilizar esta herramienta [4].

- **Selenium**, para *Nodejs* es una biblioteca de automatización del navegador, que nos permite generar cualquier tipo de web, tiene muchas utilidades. En la aplicación se ha utilizado para *scrapear* la información base para ésta y para las pruebas automáticas [31].
- **GitHub Desktop**, herramienta que nos permite conectarnos con nuestro repositorio en *GitHub* para subir los cambios realizados a la página.

En la organización del proyecto, como ya se pudo ver en la figura 3.3, el *frontend* y el *backend* se encuentran divididos en sus propias carpetas, y dentro de ellas el código se encuentra organizado en módulos para que sea fácilmente modificable.

3.2.4. Pruebas

Para comprobar el funcionamiento y para encontrar errores se ha utilizado *Selenium* para automatizar las pruebas unitarias de los diferentes componentes de la aplicación. En éstas se han ido validando que las condiciones requeridas sean correctas y validadas para poder realizar la acción correspondiente.

Gracias a *Selenium* podemos simular la navegación del usuario por los diferentes componentes y comprobar que se muestran las alertas correspondientes para informar al usuario cuando está realizando una acción incorrecta y también cuando lo hace correctamente.

Para poder llevar a cabo estos tests es necesario que tanto el cliente como el servidor se estén ejecutando y se probarán todas las características como si un usuario realizara las diferentes acciones.

```
Editar perfil usuario
✓ Modificar trabajo exitoso (3723ms)
✓ Modificar trabajo fallido, contraseña es incorrecta (791ms)
✓ Modificar email exitoso (1078ms)
✓ Modificar email fallido, el campo está vacío (699ms)
✓ Modificar email fallido, el formato del email no es válido (1005ms)
✓ Modificar email fallido, contraseña vacía (862ms)
✓ Modificar foto de perfil exitoso (588ms)
✓ Modificar foto de perfil fallido, formato de archivo inválido (580ms)
✓ Modificar foto de perfil fallido, imagen de demasiado peso (574ms)
✓ Eliminar foto de perfil exitoso (563ms)
✓ Modificar contraseña exitoso (784ms)
✓ Modificar contraseña fallida, hay pocos caracteres (716ms)

Ganancias
✓ No hay datos sobre un mes (3235ms)
✓ Descargar ganancias en .csv exitoso (422ms)
✓ Descargar ganancias en .pdf exitoso (219ms)

Loguear una cuenta
✓ Login exitoso (960ms)
✓ Login fallido, email incorrecto (965ms)
✓ Login fallido, contraseña incorrecta (950ms)
✓ Login fallido, campos vacíos (685ms)
✓ Login fallido, cuenta sin verificar (1407ms)
✓ Login fallido, cuenta bloqueada (1188ms)

Recuperar contraseña
✓ Recuperar contraseña exitoso (367ms)
✓ Recuperar contraseña fallido, email inválido (365ms)
✓ Recuperar contraseña fallido, campo de email vacío (230ms)
✓ Recuperar contraseña fallido, mal formato en el campo del email (365ms)

Registrar una cuenta
✓ Registro exitoso (3704ms)
✓ Registro fallido, el correo ya existe (2020ms)
✓ Registro fallido, el nombre contiene numeros (2099ms)
✓ Registro fallido, los apellidos contiene numeros (2039ms)
✓ Registro fallido, la fecha de nacimiento es menor a 18 años (2021ms)
✓ Registro fallido, mal formato en el campo del email (2038ms)
✓ Registro fallido, el campo del nombre está vacío (719ms)
✓ Registro fallido, no se han aceptado los términos y condiciones (2209ms)
✓ Registro fallido, no se ha aceptado el aviso de privacidad (2435ms)
✓ Registro fallido, contraseña demasiado corta (1753ms)
✓ Registro fallido, las contraseñas no coinciden (1638ms)

Validar una cuenta
✓ Cuenta sin verificar
✓ Cuenta ya verificada

72 passing (3m)
```

Figura 3.11: Resultado de tests automáticos

3.3. El producto del desarrollo

Como se ha explicado anteriormente, la aplicación es una aplicación web *responsive* por lo que debe ser visible correctamente desde cualquier dispositivo -tanto ordenadores de sobremesa (ver Fig. 3.13), como desde un dispositivo móvil (ver Fig. 3.14).

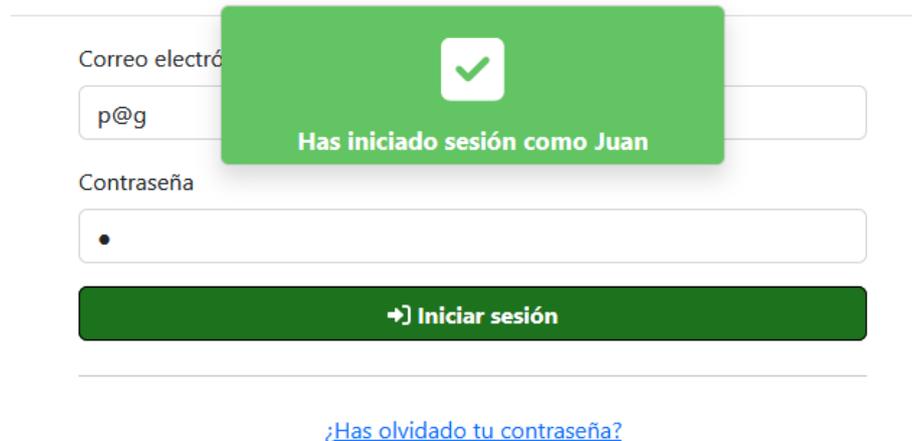


Figura 3.12: Selenium ejecutando uno de los tests

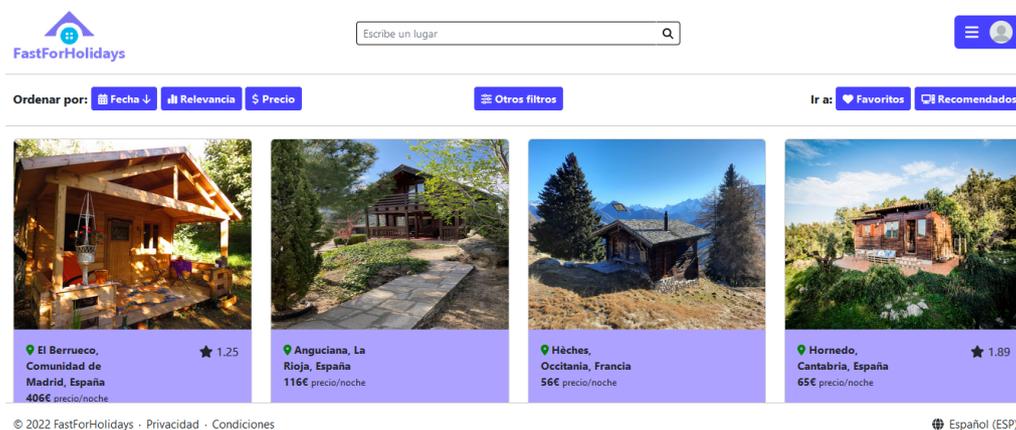


Figura 3.13: Página de inicio desde un ordenador

Se puede ver en las anteriores figuras que ambas tienen las herramientas necesarias para navegar por la página web en diferentes dispositivos, pero para explicar su funcionamiento se utilizará la versión por defecto para dispositivos con pantallas más grandes por comodidad.

En la aplicación el usuario se puede encontrar en tres estados, que son los que determinan a qué características y funcionalidades puede acceder; estos estados son:

- **Sin haber iniciado sesión**, es la primera vez que el usuario entra a la página web o su sesión anterior se ha caducado.
- **Con sesión activa**, el usuario ha registrado, validado y ha iniciado sesión con una cuenta en la aplicación.

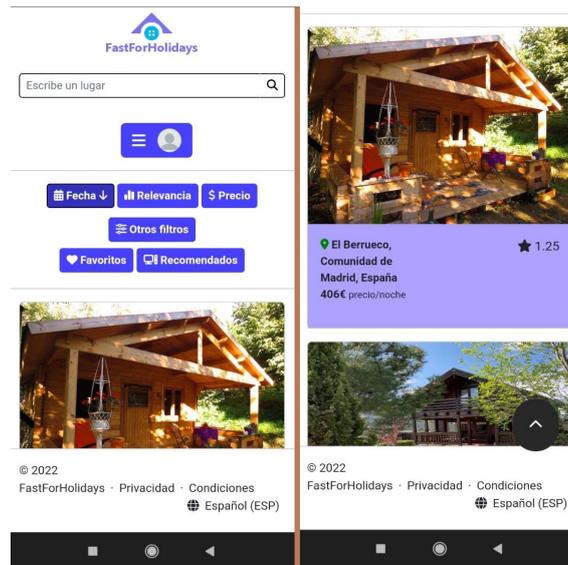


Figura 3.14: Página de inicio desde un móvil

- **Con sesión activa y rol administrado**, el usuario ha hecho los mismos pasos que con la sesión activa pero se le ha concedido el rol de administrador para tener acceso a funcionalidades para organizar y cuidar el correcto uso de la página.

El estado del usuario se ve visualmente en la página en la esquina superior derecha ya que las opciones que aparecen serán diferentes según el estado.

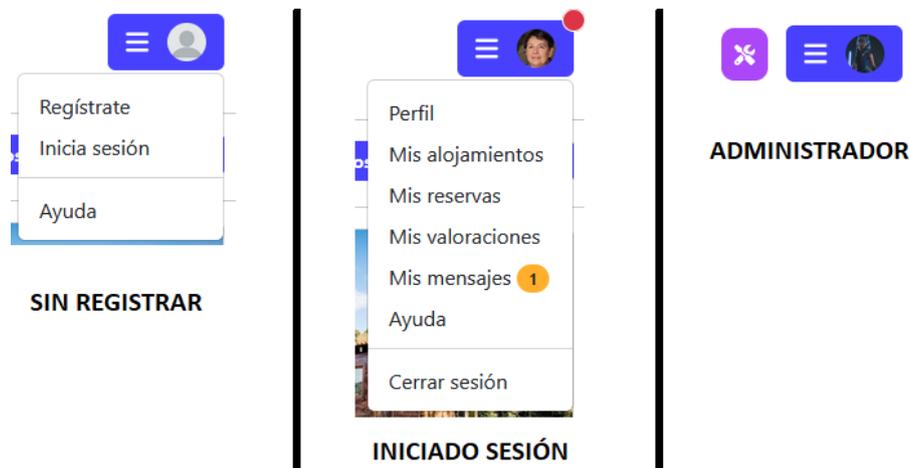


Figura 3.15: Diferentes estados del usuario en la aplicación

Los usuarios que no están registrados pueden acceder a algunas vistas de la aplicación y usar algunas de las características de la misma como pueden ser ver información de un alojamiento, ver el perfil público de un usuario, buscar alojamientos

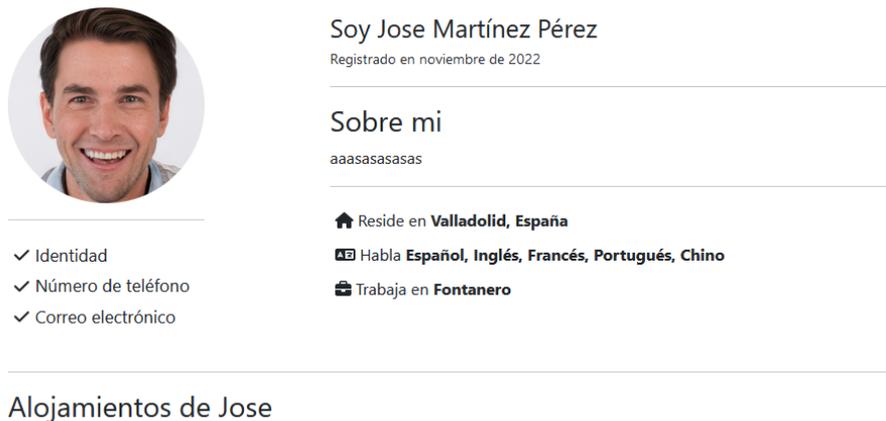
en una ubicación concreta y usar los filtros para cambiar los alojamientos que se muestran.



Figura 3.16: Vista de un alojamiento

Como podemos ver en la figura 3.16, al acceder a la información de un alojamiento nos dará información más detallada del mismo como:

- **Ubicación**, localización aproximada de dónde está situado el alojamiento.
- **Composición**, cantidad de viajeros máximos que pueden ir al alojamiento y habitaciones, camas y baños que hay disponibles.
- **Imágenes**, diferentes fotografías del alojamiento para que los usuarios puedan hacerse una idea de cómo es la vivienda.
- **Descripción**, breve texto donde el hospedador puede explicar de manera más detallada qué hay en el alojamiento o qué se puede hacer por la zona.
- **Servicios**, electrodomésticos o características que están disponibles en el alojamiento.
- **Normas**, pequeñas restricciones que el hospedador especifica al alquilar el alojamiento.
- **Hospedador**, breve descripción de quién es tu hospedador con posibilidad de contactar con él mediante el sistema de mensajería de la aplicación o acceder a su perfil público.
- **Valoraciones**, aquí puedes ver las reseñas que anteriores clientes han dejado sobre el alojamiento.
- **Reservar**, apartado donde te da información del precio por noche del alojamiento y posible selección de días y viajeros que van a reservar la vivienda.



Perfil público de un usuario. El usuario es Jose Martínez Pérez, registrado en noviembre de 2022. Su nombre de usuario es aaasasasas. Reside en Valladolid, España. Habla Español, Inglés, Francés, Portugués, Chino. Trabaja en Fontanero. Se muestran verificaciones de identidad, número de teléfono y correo electrónico.

Soy Jose Martínez Pérez
Registrado en noviembre de 2022

Sobre mi
aaasasasas

Reside en **Valladolid, España**
Habla **Español, Inglés, Francés, Portugués, Chino**
Trabaja en **Fontanero**

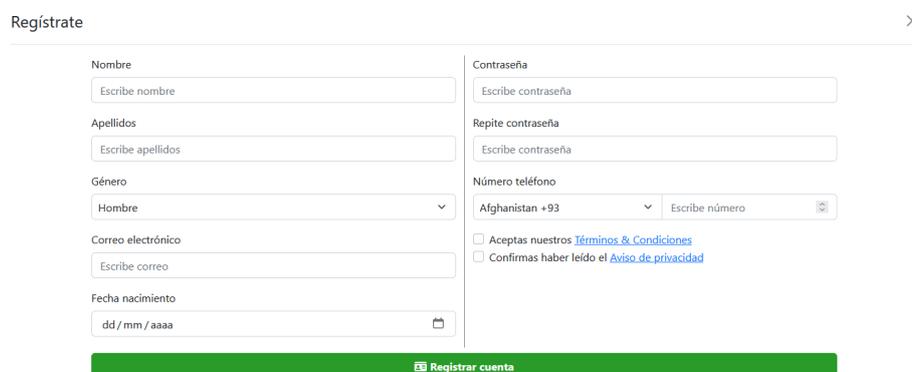
✓ Identidad
✓ Número de teléfono
✓ Correo electrónico

Alojamientos de Jose

Figura 3.17: Perfil público de un usuario

En el perfil público del usuario (ver figura 3.17) se muestra información más precisa del mismo para poder conocerlo mejor; entre esta información se encuentra:

- **Información general**, foto de perfil, nombre y apellidos, fecha de registro, pequeña biografía, lugar de residencia, idiomas que puede comunicarse y trabajo. Algunos de estos datos son opcionales por lo que cada usuario puede mostrar los que considere oportunos.
- **Identificación**, el usuario tiene confirmado su número de teléfono y correo electrónico.
- **Alojamientos**, otros alojamientos que el usuario tiene publicados en la aplicación.
- **Valoraciones**, reseñas que han hecho a sus alojamientos y a él mismo como cliente de un alojamiento.



Regístrate

Nombre
Escribe nombre

Apellidos
Escribe apellidos

Género
Hombre

Correo electrónico
Escribe correo

Fecha nacimiento
dd / mm / aaaa

Contraseña
Escribe contraseña

Repite contraseña
Escribe contraseña

Número teléfono
Afghanistan +93 Escribe número

Aceptas nuestros [Términos & Condiciones](#)
 Confirmas haber leído el [Aviso de privacidad](#)

Registrar cuenta

Figura 3.18: Registro de la aplicación

Para registrar una cuenta debemos rellenar todos los campos que hay en el formulario correctamente, si no la aplicación nos indicará qué campo es incorrecto (ver Figura 3.18); una vez registrada la cuenta, debemos validarla: se enviará un correo electrónico al *email* indicado en el registro con un *link* para poder confirmar la identidad (Figura 3.19).



Figura 3.19: Correo para validar cuenta

Una vez validada nuestra cuenta, ya podemos iniciar sesión en la aplicación. Esto nos dará acceso a más características aparte de las que ya tienen los usuarios que no están registrados, que son lista de favoritos, alojamientos recomendados, perfil, tus alojamientos, tus reservas, tus valoraciones y tus mensajes.

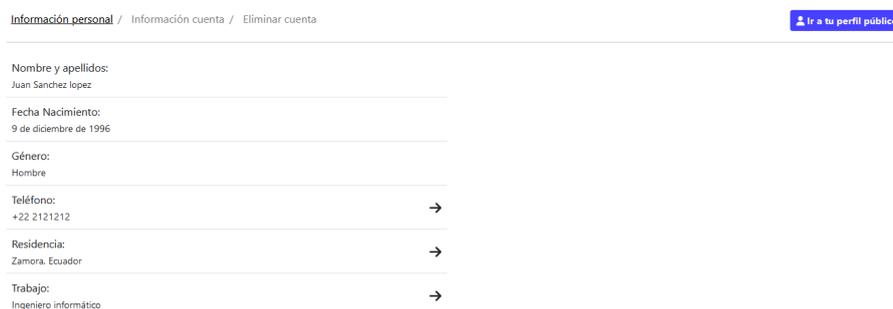


Figura 3.20: Perfil de usuario

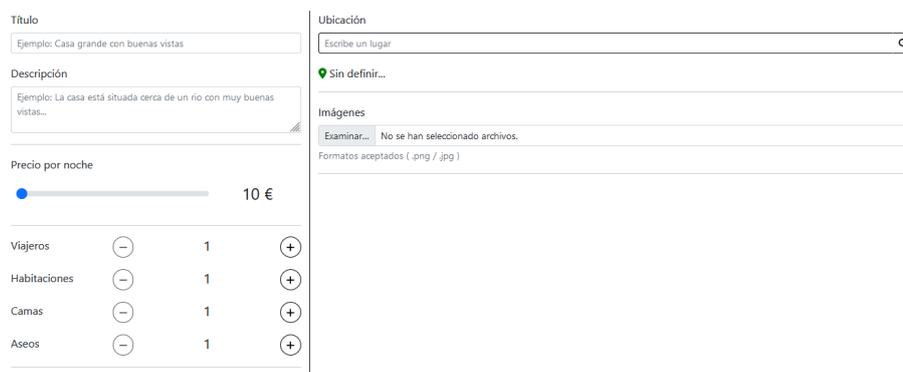
En el perfil puedes modificar la información personal o de tu cuenta, también puedes desactivar o eliminar tu usuario si lo deseas, para confirmar los cambios es necesario escribir la contraseña, como se muestra en la Figura 3.20.



Figura 3.21: Alojamientos creados por un usuario

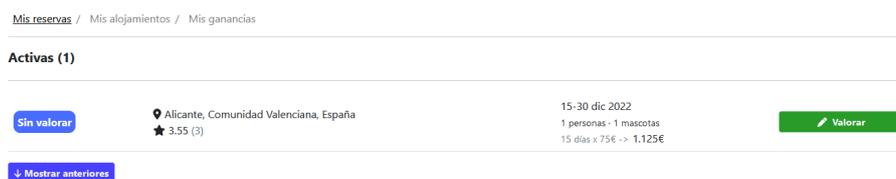
En esta vista te mostrará todos los alojamientos que has publicado en la aplicación (Figura 3.21), si pulsas sobre alguno de ellos te llevará a una vista para modificar ese alojamiento.

En esta vista también podrás crear nuevos alojamientos después de rellenar el formulario que se muestra en la Figura 3.22.



Formulario para crear un nuevo alojamiento. El formulario está dividido en dos columnas. La columna izquierda contiene: un campo de 'Título' con el ejemplo 'Casa grande con buenas vistas'; un campo de 'Descripción' con el ejemplo 'La casa está situada cerca de un río con muy buenas vistas...'; un control deslizante de 'Precio por noche' fijado en 10 €; y una lista de características con botones de '+' y '-': 'Viajeros' (1), 'Habitaciones' (1), 'Camas' (1) y 'Aseos' (1). La columna derecha contiene: un campo de 'Ubicación' con el texto 'Escribe un lugar'; un botón 'Sin definir...'; un campo de 'Imágenes' con el texto 'No se han seleccionado archivos.' y 'Formatos aceptados (.png / .jpg)'; y un botón 'Examinar...'.

Figura 3.22: Formulario para crear un nuevo alojamiento



Reservas realizadas. La interfaz muestra una barra de navegación con 'Mis reservas' seleccionado, 'Mis alojamientos' y 'Mis ganancias'. Debajo, se indica 'Activas (1)'. Se muestra una tarjeta de reserva con: un botón 'Sin valorar'; la ubicación 'Alicante, Comunidad Valenciana, España' con una puntuación de 3.55 (3); la fecha '15-30 dic 2022'; el detalle '1 personas - 1 mascotas'; el precio '15 días x 75€ -> 1.125€'; y un botón 'Valorar'.

Figura 3.23: Reservas realizadas

En esta página verás las reservas que has hecho y el estado de las mismas (Figura 3.23). Una vez pase la fecha de salida de la estancia el sistema te permitirá valorar el alojamiento y sus condiciones.

En la sección de 'Mis alojamientos' se mostrará las reservas que han hecho a alguno de tus alojamientos, el sistema es similar al de las reservas que has hecho pero aquí podrás cancelar o aceptar al usuario que quiere hacer una reserva.

Si tienes alojamientos publicados y algún usuario lo ha reservado aquí te mostrará un resumen de tus ganancias mensual (ver Figura 3.24). También podrás descargar esta información en dos formatos *PDF* o *.csv* a elección.

También hay una sección que te permite visualizar todas las valoraciones que has hecho tanto a alojamientos o inquilinos de tus viviendas y las que has recibido tu como hospedador y como inquilino (Figura 3.25).

Mis reservas / Mis alojamientos / **Mis ganancias**

DICIEMBRE DE 2022
01/12/2022 - 31/12/2022

[Descargar .csv](#) [Descargar .pdf](#)

ID:	4	ID	Info	Coste
Cliente:	Jose Martínez Pérez	1	2 noches x 406€	816€
Reservas:	5	2	13 noches x 406€	5.278€
		3	15 noches x 75€	1.125€
		5	2 noches x 406€	812€
		8	7 noches x 75€	525€
		Total:		8.556€

NOVIEMBRE DE 2022 ENERO DE 2023

Figura 3.24: Ganancias mensuales obtenidas

Hechas-Alojamientos / Hechas-Usuarios / Recibidas-Alojamientos / Recibidas-Usuarios

Valoraciones (3)

<p>Hornedo, Cantabria, España</p> <p>★ 1.89 (1)</p>	<p>Llegada: 4 Veracidad: 2.6 Comunicación: 5 Ubicación: 1.9 Limpieza: 4.6 Calidad: 4.6</p>	<p>20 de diciembre de 2022</p> <p>Muy buen alojamiento, con muy buenos servicios estuvimos muy a gusto allí.</p>
<p>El Berrueco, Comunidad de Madrid, España</p> <p>★ 1.25 (2)</p>	<p>Llegada: 0 Veracidad: 0 Comunicación: 0 Ubicación: 0 Limpieza: 0 Calidad: 0</p>	<p>11 de diciembre de 2022</p> <p>Otra prueba de valoración a ver si se suma bien xd</p>

Figura 3.25: Valoraciones hechas de alojamientos



Figura 3.26: Sistemas de mensajes de la aplicación

El sistema de mensajería te permite comunicarte con otros usuarios, por si quieres conocer más información sobre un alojamiento o acordar algún elemento para hacer la reserva (Figura 3.26).

Y por último el estado de un usuario que tiene el rol de administrador, permite acceder al panel de administración (Figura 3.27).

En esta vista mostrará información general de la aplicación y podrás pulsar en los elementos de alojamientos, usuarios y reportes cada uno de ellos te enseñará datos sobre cada uno de ellos que se encuentran en la aplicación (Figura 3.28).

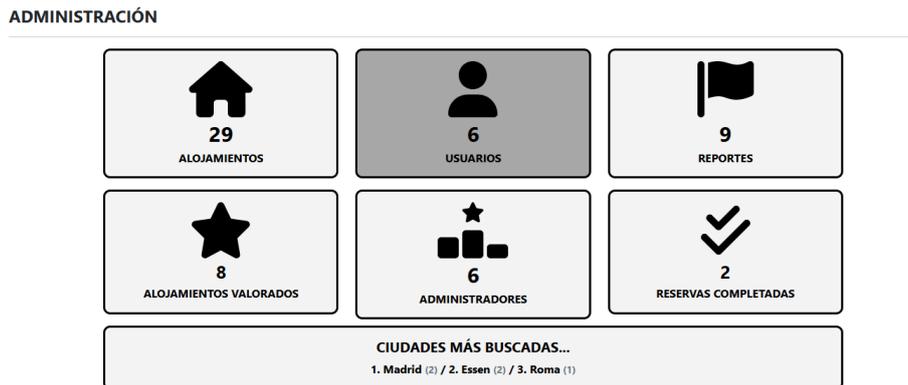


Figura 3.27: Vista de administración

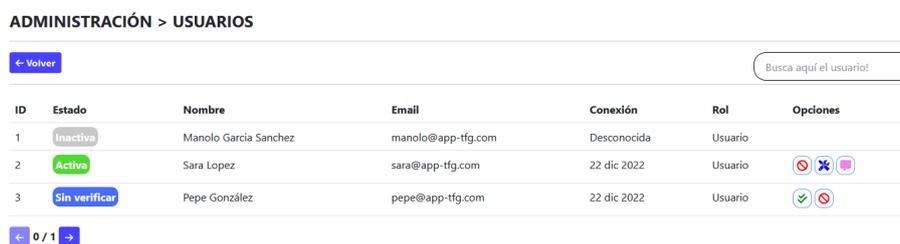


Figura 3.28: Administración de usuarios de la aplicación

Se pueden ver los usuarios que hay registrados en la aplicación y el estado de su cuenta, como administrador podrás modificar su estado para verificar, bloquear, desbloquear, hacer administrador o quitar el rol de la cuenta según lo que quieras hacer, también podrás enviar mensajes al usuario que selecciones. Y puedes filtrar el usuario por el nombre para facilitar su búsqueda.

```

TERMINAL
You can now view client in the browser.
Local: http://localhost:3000
On Your Network: http://192.168.1.83:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully

PS C:\Users\Wigue1\Desktop\TFG-APP\server> node .\server.js
El servidor se está ejecutando en http://localhost:8000/
[OK] MySQL
  
```

Figura 3.29: Ejecución del *frontend* y *backend* de la aplicación

Para la correcta ejecución de la aplicación y poder navegar por ella se explicará en el manual de usuario que también está incluido en este documento en las secciones finales del mismo.

Capítulo 4

Evaluación

4.1. Proceso de evaluación

Para probar el funcionamiento de la aplicación se han realizado test unitarios del sistema directamente sobre la página web utilizando *Selenium* y *mocha*.

4.1.1. Forma de evaluación

Para evaluar los diferentes componentes se utilizan los casos de prueba para conocer cuál es la entrada y la salida correspondiente para cada test y a partir de esa información se construye el test unitario poniendo los datos de entrada y validando que la salida es la correcta para ese dato de entrada concreto.

Todas estas pruebas se realizan de forma automática mediante *Selenium* y *Gecko*, que nos permiten simular un navegador y probar los elementos como si una persona real estuviera insertando los datos en la aplicación.

4.1.2. Casos de prueba

Se han realizado diferentes casos de prueba para los diferentes componentes, haciendo que en algunas ocasiones inserte un valor fallido y en otros uno correcto y comprobando que ese es el resultado esperado para esa acción como se puede ver en la figura 4.1 y en la figura 4.2.

```
Editar perfil usuario
✓ Modificar trabajo exitoso (3723ms)
✓ Modificar trabajo fallido, contraseña es incorrecta (791ms)
✓ Modificar email exitoso (1078ms)
✓ Modificar email fallido, el campo está vacío (699ms)
✓ Modificar email fallido, el formato del email no es válido (1005ms)
✓ Modificar email fallido, contraseña vacía (862ms)
✓ Modificar foto de perfil exitoso (588ms)
✓ Modificar foto de perfil fallido, formato de archivo inválido (580ms)
✓ Modificar foto de perfil fallido, imagen de demasiado peso (574ms)
✓ Eliminar foto de perfil exitoso (563ms)
✓ Modificar contraseña exitoso (784ms)
✓ Modificar contraseña fallida, hay pocos caracteres (716ms)

Ganancias
✓ No hay datos sobre un mes (3235ms)
✓ Descargar ganancias en .csv exitoso (422ms)
✓ Descargar ganancias en .pdf exitoso (219ms)

Loguear una cuenta
✓ Login exitoso (960ms)
✓ Login fallido, email incorrecto (965ms)
✓ Login fallido, contraseña incorrecta (950ms)
✓ Login fallido, campos vacíos (685ms)
✓ Login fallido, cuenta sin verificar (1407ms)
✓ Login fallido, cuenta bloqueada (1188ms)

Recuperar contraseña
✓ Recuperar contraseña exitoso (367ms)
✓ Recuperar contraseña fallido, email inválido (365ms)
✓ Recuperar contraseña fallido, campo de email vacío (230ms)
✓ Recuperar contraseña fallido, mal formato en el campo del email (365ms)

Registrar una cuenta
✓ Registro exitoso (3704ms)
✓ Registro fallido, el correo ya existe (2020ms)
✓ Registro fallido, el nombre contiene numeros (2099ms)
✓ Registro fallido, los apellidos contiene numeros (2039ms)
✓ Registro fallido, la fecha de nacimiento es menor a 18 años (2021ms)
✓ Registro fallido, mal formato en el campo del email (2038ms)
✓ Registro fallido, el campo del nombre está vacío (719ms)
✓ Registro fallido, no se han aceptado los términos y condiciones (2209ms)
✓ Registro fallido, no se ha aceptado el aviso de privacidad (2435ms)
✓ Registro fallido, contraseña demasiado corta (1753ms)
✓ Registro fallido, las contraseñas no coinciden (1638ms)

Validar una cuenta
✓ Cuenta sin verificar
✓ Cuenta ya verificada

72 passing (3m)
```

Figura 4.1: Resultado de test automáticos

4.2. Análisis de resultados

Como se puede ver en la figura 4.1 y en la figura 4.2 todos los tests se cumplen de forma correcta, por lo que de esta forma se comprueba el correcto funcionamiento de la aplicación.

Al estar probando directamente sobre el navegador esto también nos indica que los errores se informan al usuario para que pueda cambiar los datos que ha insertado por unos valores válidos, ya que en las pruebas para comprobar si la respuesta es la correcta se busca el elemento de advertencia que se envía al usuario.

```
Editar perfil usuario
✓ Modificar trabajo exitoso (3723ms)
✓ Modificar trabajo fallido, contraseña es incorrecta (791ms)
✓ Modificar email exitoso (1078ms)
✓ Modificar email fallido, el campo está vacío (699ms)
✓ Modificar email fallido, el formato del email no es válido (1005ms)
✓ Modificar email fallido, contraseña vacía (862ms)
✓ Modificar foto de perfil exitoso (588ms)
✓ Modificar foto de perfil fallido, formato de archivo inválido (580ms)
✓ Modificar foto de perfil fallido, imagen de demasiado peso (574ms)
✓ Eliminar foto de perfil exitoso (563ms)
✓ Modificar contraseña exitoso (784ms)
✓ Modificar contraseña fallida, hay pocos caracteres (716ms)

Ganancias
✓ No hay datos sobre un mes (3235ms)
✓ Descargar ganancias en .csv exitoso (422ms)
✓ Descargar ganancias en .pdf exitoso (219ms)

Loguear una cuenta
✓ Login exitoso (960ms)
✓ Login fallido, email incorrecto (965ms)
✓ Login fallido, contraseña incorrecta (950ms)
✓ Login fallido, campos vacíos (685ms)
✓ Login fallido, cuenta sin verificar (1407ms)
✓ Login fallido, cuenta bloqueada (1188ms)

Recuperar contraseña
✓ Recuperar contraseña exitoso (367ms)
✓ Recuperar contraseña fallido, email inválido (365ms)
✓ Recuperar contraseña fallido, campo de email vacío (230ms)
✓ Recuperar contraseña fallido, mal formato en el campo del email (365ms)

Registrar una cuenta
✓ Registro exitoso (3704ms)
✓ Registro fallido, el correo ya existe (2020ms)
✓ Registro fallido, el nombre contiene numeros (2099ms)
✓ Registro fallido, los apellidos contiene numeros (2039ms)
✓ Registro fallido, la fecha de nacimiento es menor a 18 años (2021ms)
✓ Registro fallido, mal formato en el campo del email (2038ms)
✓ Registro fallido, el campo del nombre está vacío (719ms)
✓ Registro fallido, no se han aceptado los términos y condiciones (2209ms)
✓ Registro fallido, no se ha aceptado el aviso de privacidad (2435ms)
✓ Registro fallido, contraseña demasiado corta (1753ms)
✓ Registro fallido, las contraseñas no coinciden (1638ms)

Validar una cuenta
✓ Cuenta sin verificar
✓ Cuenta ya verificada

72 passing (3m)
```

Figura 4.2: Resultado de test automáticos

También se han realizado unas preguntas a los usuarios después de haber usado el sistema para obtener retroalimentación y ver qué problemas o ideas nuevas se pueden implementar en la aplicación. Se han utilizado personas de un rango de edad amplio y con diferentes conocimientos en informática para poder ver opiniones diferentes e investigar los resultados.

El formulario se compone de 1 pregunta corta a responder que es opcional y 10 afirmaciones que se debe seleccionar el nivel de en desacuerdo o muy a favor que se está de la misma, siendo el valor 1 como muy en desacuerdo y 5 muy a favor. Las preguntas son:

- Añadirías alguna nueva funcionalidad o sientes que falta algo en la aplicación.
- Creo que me gustaría utilizar este sistema con frecuencia.
- Encontré el sistema innecesariamente complejo.
- Pensé que el sistema era fácil de usar.
- Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.
- Creo que las funciones del sistema están bien integradas.
- El sistema es inconsistente.
- Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.
- Encontré el sistema muy complicado de usar.
- Me sentí muy seguro usando el sistema.
- Necesitaba aprender muchas cosas antes de poder usar este sistema.

Las opiniones generales de los usuarios que más coinciden ha sido que no han encontrado el sistema innecesariamente complejo, no necesitarían apoyo técnico, funciones bien integradas, sistema consistente, medianamente fácil de usar, seguridad usando el sistema y no era necesario aprender cosas antes de usarlo.

Conclusión

Aportaciones realizadas

Las aportaciones realizadas con este trabajo incluye la realización de una aplicación web completa, con amplias características y funcionalidades, incluyendo su carácter *responsive*, y con tecnologías actuales. Se ha diseñado dentro del sector inmobiliario así que puede ser utilizado en el futuro para crear una empresa de este tipo. Se han creado test automáticos con *Selenium* para comprobar el correcto funcionamiento de la aplicación.

Trabajos futuros

Posible mejoras que podrían implementarse en la aplicación:

- **Backup automático de la base de datos**, hacer este proceso automático elimina trabajo a los empleados de la aplicación, además de que tener una replica de una información es importante para casos de pérdida por algún problema físico, ataque cibernético o algún empleado al ejecutar erróneamente alguna sentencia *SQL*. Esto se haría en la parte del *backend* y si es necesario se podría enviar la información a otro servidor para evitar perder toda la información en caso de que en el que corre la aplicación esté comprometido.
- **Añadir más tipos de alquiler o compra**, se podrían añadir secciones para buscar un alquiler mensual o compra de alojamientos dentro de la aplicación, ya que el sistema base para mostrar las viviendas sería muy similar para todos estos tipos de comercio y por lo tanto fácilmente modificable.
- **Facilitar registro a hoteles**, parecido al punto anterior añadir hoteles da la posibilidad de tener un mayor rango de lugares de destino dentro de la aplicación lo que conlleva un posible incremento de ingresos económicos. Además, los usuarios podrían ver reseñas de otros usuarios sobre los hoteles.

- **Mapa navegable**, añadir un mapa interactivo en el cual te puedas ir moviendo por los diferentes países y ver qué alojamientos hay en cada lugar de una manera más simple que ir buscando lugar por lugar.
- **Búsqueda por etiquetas**, añadir etiquetas a los alojamientos para si el usuario está buscando algo en concreto le sea más fácil.
- **Idiomas**, el menú seleccionable está hecho pero no la funcionalidad, habría que contratar varios dominios para los diferentes lenguajes que vaya a tener la aplicación.

Problemas encontrados

Algunos de los problemas más importantes que han ocurrido a lo largo del desarrollo de la aplicación han sido:

- **Scrapear AirBnb**, fue el primer problema que hubo al empezar el desarrollo, ya que la página *AirBnb* es una página que se genera dinámicamente, y los sistemas para *scrapear* que estaba utilizando no servían ya que no podía obtener la información que yo deseaba, por lo que indagando más sobre el tema encontré la herramienta *Selenium* que da la posibilidad de simular la página con los driver de un navegador y ya por fin obtener los datos necesarios.
- **Buscar un destino**, está claro que para buscar un destino lo que tienes que hacer es escribir la ubicación, pero el problema estaba en cómo funcionaba por detrás para comparar el destino buscando con las ubicaciones de los alojamientos y mostrar las coincidencias. Al principio se usó la ubicación completa de los alojamientos usando sólo un campo que se comparaba con el texto de la ubicación, pero este método era poco eficaz. Después de probar más a fondo la *API* de *GoogleMaps* se barajó la posibilidad de obtener la latitud y la longitud de la ubicación buscada y compararlas con la latitud y la longitud almacenada, pero tampoco fue muy eficaz ya que si por ejemplo colocas sólo el país la variación de estos va a ser muy alta. Por lo que finalmente se tomó la decisión de partir la ubicación almacenada en localidad, provincia, comunidad, país ya que las búsquedas mejoran cuando buscas alojamientos en un país concreto, una comunidad, etc. Por tanto, al introducir un destino también se dividirá en estos campos y si alguno no se encuentra no es comparado y se pasa al siguiente para hacer el emparejamiento.
- **Persistencia de la sesión**, inicialmente se utilizó el almacenamiento local para conocer si el usuario había iniciado sesión y las credenciales se guardaban

sin ninguna forma de encriptación por lo que en temas de seguridad no era muy bueno y tener que enviar estos datos manualmente al *backend* era tedioso. Después de investigar más acerca del tema se decidió utilizar *JWT* y *cookies* que facilitan mucho el trabajo para la comunicación y almacenamiento de la información de sesión.

- **GoogleMaps tarjeta de crédito**, una vez se implantó *GoogleMaps* en el proyecto no entendía por qué no funcionaba, ya que realizando anteriormente pruebas con algunos tutoriales [15] todo funcionaba correctamente, podía agregar mapas, autocompletar ubicaciones, etc. Sin embargo, usándolo en el proyecto no funcionaba. Buscando más información vi que era necesario crear una cuenta en *Google Cloud* [14] y también añadir una tarjeta de crédito para poder utilizar la *API*, por lo que en un principio se barajó cambiar a otra alternativa como puede ser *OpenMaps* pero las posibilidades que da *GoogleMaps* eran mucho mayores, por lo que se tomó la decisión de mantener este último y añadir una tarjeta para poder utilizarlo.

Opiniones personales

Desde mi punto de vista, siendo una persona que me gustan los retos y poder superarme, el desarrollo de una aplicación desde 0 me ha hecho aprender muchas cosas no sólo en el apartado de conocer nuevas tecnologías y obtener más experiencia en las conocidas, sino también a la hora de resolver los problemas, metodologías de trabajo y ver proporción de lo que puede llegar a ser un proyecto real, ya que durante la carrera en muchas asignaturas te piden desarrollar diferentes aplicaciones cada una con sus dificultades pero no de este manera. Además hay mucha más libertad a la hora de elegir qué funcionalidades va a tener la aplicación, ya que en las asignaturas el proyecto suele estar definido por un guión y unos pasos a seguir, por lo que también se fomenta la creatividad. Por ello voy a mencionar los puntos que para mí han sido más importantes y que me pueden servir en un futuro cuando me enfrente a algún problema similar.

- **Resolución de problemas**, en proyectos así siempre surgen diferentes problemas y se deben buscar soluciones para poder seguir avanzando, por lo que esto siempre te ayuda a mejorar como profesional, ya que te permite buscar soluciones de manera más rápida o si ya te has encontrado este problema es más fácil solucionarlo en un futuro, también buscar mejores herramientas para solucionarlos o mejores formas de buscar la solución.

- **Centrarse en lo importante**, y qué es lo importante, pues en un proyecto como éste lo importante y principal es seguir avanzando en el desarrollo y cumplir todos los objetivos propuestos, pero también se debe tener en cuenta que hay que evitar pensamientos o ideas negativas, que puedan hacer que se cancele o retrase, tener una mentalidad positiva, saber escuchar las opiniones o sugerencias de los demás, son cosas fundamentales que te ayudan para continuar o darte motivación.
- **Sobre pensar las cosas**, a veces pensar demasiado sobre cómo hacer las cosas o cómo mejorarlas hace que no avancemos en las demás, por lo que hay que saber cuándo pasar a una nueva cuando ya hemos completado la actual y así evitar retrasarnos demasiado al centrarnos sólo en un elemento.

Lista de referencias

- [1] Auriculares empleados pccomponentes. <https://www.pccomponentes.com/owlotech-ear-office-next-auriculares-con-microfono-negros>, [Online; accessed 10-Febrero-2023]
- [2] Boredhumans ia. <https://boredhumans.com/faces.php>, [Online; accessed 19-Diciembre-2022]
- [3] Cuestionario google forms. https://docs.google.com/forms/d/e/1FAIpQLScJbkBe0wHEjQmti1rtD0nKjpQ7AVP3RY_CtWnynwESgY6P2g/viewform, [Online; accessed 13-Abril-2023]
- [4] Datepicker. <https://kiarash-z.github.io/react-modern-calendar-datepicker/docs/getting-started>, [Online; accessed 13-Enero-2022]
- [5] Db diagram. <https://dbdiagram.io/d>, [Online; accessed 08-Noviembre-2022]
- [6] Despacho privado coworkingspain. <https://coworkingspain.es/espacios/coworking/leon/centro-de-negocios-leon>, [Online; accessed 10-Febrero-2023]
- [7] Documentación fontawesome. <https://fontawesome.com/docs>, [Online; accessed 09-Enero-2023]
- [8] Documentación mariadb. <https://mariadb.org/documentation/>, [Online; accessed 09-Enero-2023]
- [9] Documentación nodejs. <https://nodejs.org/en/docs>, [Online; accessed 09-Enero-2023]
- [10] Documentación paypal. <https://developer.paypal.com/home>, [Online; accessed 09-Enero-2023]
- [11] Documentación react. <https://es.react.dev/reference/react>, [Online; accessed 09-Enero-2023]

-
- [12] Documentación react-bootstrap. <https://react-bootstrap.netlify.app/>, [Online; accessed 09-Enero-2023]
- [13] Documentación react-googlemaps. <https://react-google-maps-api-docs.netlify.app/#section-introduction>, [Online; accessed 09-Enero-2023]
- [14] Google cloud. <https://console.cloud.google.com/getting-started>, [Online; accessed 09-Enero-2023]
- [15] Googlemaps tutorial. <https://developers.google.com/maps/documentation/javascript/adding-a-google-map?hl=es-419>, [Online; accessed 09-Enero-2023]
- [16] Html error codes. <https://kb.iu.edu/d/bfrc>, [Online; accessed 15-Febrero-2022]
- [17] Imágenes de stock. <https://unsplash.com>, [Online; accessed 09-Noviembre-2022]
- [18] Modelo incremental. <https://blog.comparasoftware.com/que-es-el-modelo-incremental/>, [Online; accessed 09-Enero-2023]
- [19] Nodejs bcrypt. <https://www.npmjs.com/package/bcrypt>, [Online; accessed 12-Diciembre-2022]
- [20] Nodejs express. <https://www.npmjs.com/package/express>, [Online; accessed 05-Diciembre-2022]
- [21] Nodejs nodemailer. <https://www.npmjs.com/package/nodemailer>, [Online; accessed 14-Diciembre-2022]
- [22] Nodejs pdfkit. <https://www.npmjs.com/package/pdfkit>, [Online; accessed 27-Febrero-2023]
- [23] Portátil empleados pccomponentes. <https://www.pccomponentes.com/microsoft-surface-go-2-intel-pentium-gold-4425y-8gb-128gb-105-tactil>, [Online; accessed 10-Febrero-2023]
- [24] Prefijos telefónicos mundiales. <https://gist.github.com/anubhavshrial/75f6183458db8c453306f93521e93d37>, [Online; accessed 12-Diciembre-2022]
- [25] Ratones empleados pccomponentes. <https://www.pccomponentes.com/owlotech-m30-raton-1000-dpi-negro>, [Online; accessed 10-Febrero-2023]

- [26] Repositorio de github. <https://github.com/MiguelTurr/TFG-APP>, [Online; accessed 02-Noviembre-2022]
- [27] Salario Desarrollador Web jobted. <https://www.jobted.es/salario/desarrollador-web>, [Online; accessed 10-Febrero-2023]
- [28] Salario Diseñador Gráfico jobted. <https://www.jobted.es/salario/dise%C3%B1ador-gr%C3%A1fico>, [Online; accessed 10-Febrero-2023]
- [29] Salario Jefe Proyecto jobted. <https://www.jobted.es/salario/project-manager>, [Online; accessed 10-Febrero-2023]
- [30] Salario Tester talent. <https://es.talent.com/salary?job=tester>, [Online; accessed 10-Febrero-2023]
- [31] Selenium. <https://www.npmjs.com/package/selenium-webdriver>, [Online; accessed 31-October-2022]
- [32] Airbnb: Alojamientos vacacionales y particulares en Airbnb. <https://www.airbnb.es/> (2023), [Online; accessed 02-Febrero-2023]
- [33] Idealista: Casas y pisos, alquiler y venta. Anuncios gratis. <https://www.idealista.com/> (2023), [Online; accessed 02-Febrero-2023]
- [34] Quoters: Aprende a calcular la amortización de los equipos de tu empresa desde cero. <https://www.quoters.io/es/aprende-a-calcular-la-amortizacion-de-los-equipos-de-tu-empresa-desde-cero/>, [Online; accessed 25-Marzo-2023]

Anexo A

Control de versiones

Como controlador de versiones se ha utilizado *GitHub*, en donde se ha creado un repositorio en el cual se ha ido subiendo el avance del desarrollo de la aplicación. Este servicio además de tener un *backup* de nuestro código fuente también nos permite volver a versiones anteriores de nuestra aplicación en caso de que hayamos cometido algún error.

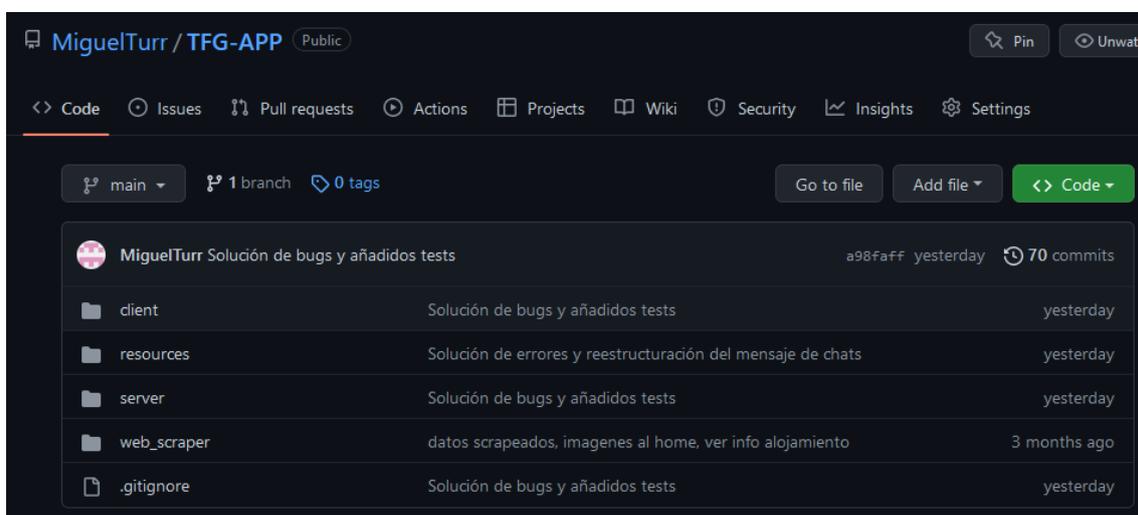


Figura A.1: Repositorio en GitHub

GitHub también nos permite tener información de los cambios que se han realizado en cada *commit*, las fechas de las subidas, etc.

El repositorio se encuentra público por lo que se puede acceder mediante el siguiente enlace [26].



Figura A.2: Commits del repositorio

Anexo B

Seguimiento de proyecto

B.1. Forma de seguimiento

Para el seguimiento que han realizado reuniones semanales en las cuales se ha revisado la evolución del proyecto, también se han hecho revisiones de funcionalidades y pruebas de las características para comprobar que no hay errores.

B.2. Planificación inicial

La planificación que se ha seguido se muestra en el apartado 2.2.3, pero debido a diferentes complicaciones con algunas tareas se ha tenido que reestructurar algunas.

El retraso se debió en la implementación de *GoogleMaps*, ya que requería de añadir una tarjeta de crédito para poder usar la *API* y se valoró cambiar a otra herramienta, pero finalmente se consiguió que funcionase pero hizo que el proyecto se retrasase.

B.3. Planificación final

Debido a estos retrasos el Diagrama Grantt final queda como se muestra en la figura B.1.

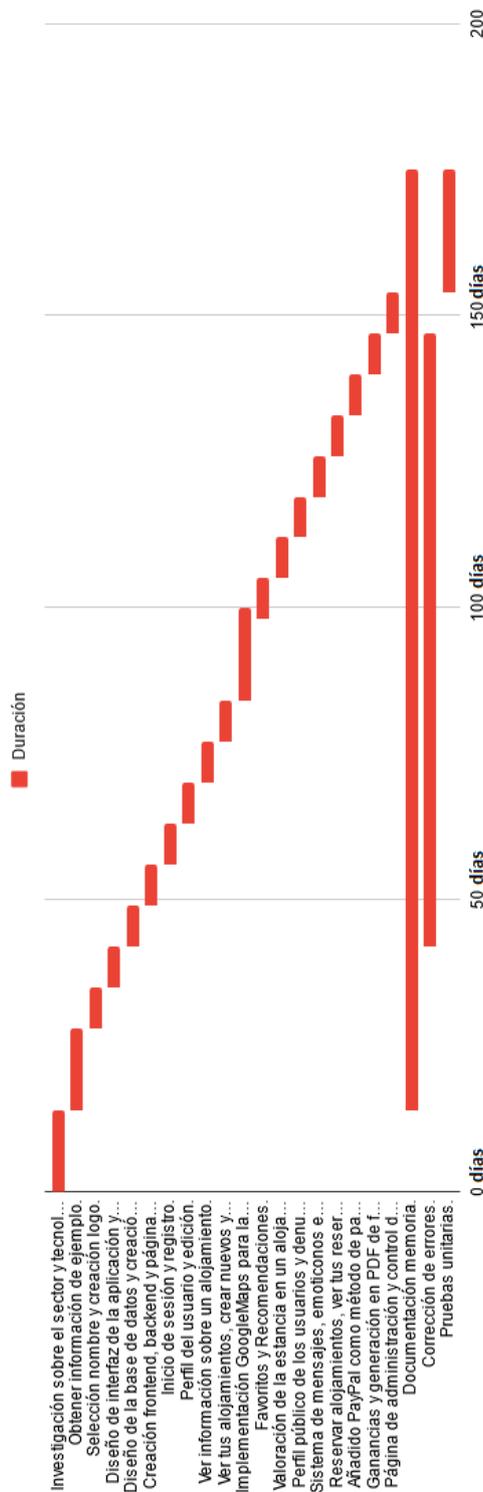


Figura B.1: Gráfico Gantt final

Anexo C

Manual de usuario

Para poder poner en funcionamiento la aplicación debemos tener instalado *npm* que es una herramienta para la terminación que nos permite crear proyectos de *JavaScript*, instalar módulos y ejecutar nuestro proyecto. Debemos tener instalado *MySQL* para el sistema de gestión de bases de datos y debemos tenerlo iniciado o tener un servidor virtual privado donde se aloje el servicio.

También es necesario que el servidor *MySQL* contenga la base de datos de la aplicación.

Todos los demás módulos se descargarán automáticamente una vez ejecutemos nuestro proyecto.

Lo primero que debemos hacer es descargar el proyecto del siguiente repositorio [26].

Se puede usar el siguiente comando de git para clonar el proyecto:

```
1 git clone https://github.com/MiguelTurr/TFG-APP.git nombre_carpeta
```

Este comando clonará el proyecto en nuestro ordenador y podemos cambiar el parámetro de **'nombre_carpeta'** para que se descargue en la carpeta con el nombre que queramos.

Una vez tengamos el proyecto en nuestro ordenador debemos crear el archivo de configuración del servidor para ello nos dirigimos a **/server/services/**.

Aquí dentro crearemos el archivo **'config.js'** que tendrá la siguiente información.

- **server_hostname**, por defecto **'localhost'** es la dirección que usará el servidor para atender las peticiones.

- **server_port**, por defecto 8000 es el puerto que usará el servidor para atender las peticiones.
- **mysql_host**, por defecto 'localhost' es la localización de donde se encuentra la base de datos.
- **mysql_user**, el usuario que se usará para conectarse a la base de datos.
- **mysql_password**, la contraseña que se usará para conectarse a la base de datos.
- **mysql_database**, por defecto 'app' es el nombre de la base de datos que se va a usar en la aplicación.
- **email_user**, correo utilizado para enviar los *emails* de la aplicación, debe ser un correo de *gmail* ya que se ha configurado para usar este servicio.
- **email_password**, clave del correo.
- **bcrypt_salt**, por defecto 10 el salto que se utilizará para encriptar las contraseñas.
- **cookie_secret**, palabra secreta que se usa para generar los *JWT*.
- **google_api_key**, clave para conectarnos con la *API* de *GoogleMaps*.
- **paypal_client**, cliente para utilizar el servicio de *PayPal*.
- **paypal_secret**, clave para utilizar el servicio de *PayPal*.
- **comision_porcentaje**, por defecto 10, es la comisión que se lleva la aplicación por el alquiler de los alojamientos.
- **dev_state**, por defecto *true* es para indicar si nos encontramos en desarrollo.

Este archivo se debe crear ya que contiene información sensible con respecto a la aplicación, que si es conocida por una persona externa puede causar daños tanto en el funcionamiento como en los datos.

En caso de que nuestro servidor MySQL no tenga creada la tabla para la aplicación podemos encontrar el archivo de creación dentro de la carpeta de **/resources/** aquí veremos dos archivos:

- **db-vacia.sql**, que es la creación de la base de datos por defecto sin ninguna información.
- **db-datos.sql**, que creará la base de datos con alguna información por defecto.

Si hemos añadido la base de datos con información ya creada también debemos añadir al servidor las imágenes de los alojamientos que estos se encuentran en la carpeta de `/resources/alojamients-imagenes/` copiamos y pegamos todos los archivos en `/server/imagenes/casas/`

Guardamos el archivo y abrimos dos terminales, una para ejecutar el servidor y otra para ejecutar el cliente. Nos dirigimos en cada caso a:

```
1 cd ./server/  
2 cd ./client/
```

Primero debemos ejecutar el cliente para que no haya ningún error, por lo que en la terminal que tenemos dentro de la carpeta del servidor ejecutaremos el siguiente comando:

```
1 node ./server.js
```

Y en la carpeta del cliente ejecutaremos:

```
1 npm start
```

Al introducir este comando nos redireccionará a la página principal de la aplicación en nuestro navegador en caso de que no podemos abrir el navegador y colocar en el buscador la dirección que nos indique en la consola, que sería de la forma:

```
http://localhost:puerto/
```

Si además queremos ejecutar los test unitarios, creamos otro terminal y nos dirigimos a la carpeta servidor.

```
1 cd ./server/
```

Una vez ahí, podemos utilizar el siguiente comando:

```
1 npm test
```

Este comando empezará a ejecutar todos los test que haya creados en la carpeta `./test/integration/`

Y así es como pondríamos en ejecución nuestro proyecto en modo desarrollo y ya podemos navegar por la página web.

Este procedimiento sirve tanto para un dispositivo con *Windows* o *Linux*.

Anexo D

Cuestionario de evaluación

Se ha realizado un cuestionario [3] a los usuarios después de probar el sistema para conocer la opinión y obtención de nuevas ideas, utilizando *Google Forms*.

Los resultados se han analizado en el apartado 4.2.