

PROYECTO FIN DE GRADO: PLATAFORMA DE OFERTA DE ALOJAMIENTOS

Miguel Sánchez Ramírez
miguelSchz1997@gmail.com



Centro Oficial de Formación Profesional

Índice:

1. Resumen.....	3
2. Introducción	5
3. Objetivos y características del proyecto	7
4. Finalidad	9
5. Medios materiales usados	11
5.1. Especificaciones hardware	11
5.2. Visual Studio Code	12
5.3. HTML	13
5.4. CSS	15
5.5. SASS.....	17
5.6. JavaScript	18
5.7. Gulp.....	20
5.8. PHP	20
5.9. XAMPP	22
5.10. Workbench	23
5.11. Git	23
5.12. GitHub.....	24
5.13. npm.....	25
5.14. Chocolatey.....	27
5.16. Responsively.....	28

5.17. 000WebHost	29
6. Planificación del proyecto.....	31
6.1. Diseño de la base de datos.....	31
6.2. Diseño de la interfaz de usuario	34
6.2.1. Diagrama de casos de uso.....	34
6.2.2. Diagrama de estado	35
6.3. Estructura del código	36
6.3.1. Creación de la estructura inicial.....	37
6.4. API Google Maps JavaScript	39
6.5. Subida del proyecto a 000webhost	42
7. Fase de pruebas	49
7.1. ¿Por qué es importante realizar el testing?.....	49
7.2. Tipos de testing.....	49
7.2.1. End to End	50
7.2.2. Unitarias	50
7.2.3. Integración.....	50
7.4. Pruebas en la plataforma	51
8. Conclusiones y trabajos futuros o posibles mejoras	69
9. Apéndices.....	71
10. Referencias bibliográficas	79

1. Resumen

El problema que trata de solucionar este proyecto se centra en el alquiler y la compra de alojamientos de estancia corta de manera rápida y sencilla. Consistiría en una plataforma web mediante la cual los anfitriones pueden publicitar y contratar el arriendo de sus propiedades con sus huéspedes; los anfitriones pueden valorar a los huéspedes, como referencia para futuros usuarios. El líder actual en este tipo de negocio en la web es Airbnb, fundada en 2008 en Estados Unidos, aunque también hay muchas alternativas a este, como HomeAway o Wimdu.

Por lo tanto, la idea de proyecto no es una innovación total, sin embargo, mi idea era hacer mucho más sencilla y accesible este tipo de plataformas a usuarios, adaptando las funcionalidades esenciales al proyecto y creando una plataforma simple y rápida.



Principales competidores del mercado.

2. Introducción

El proyecto va a estar estructurado en varios puntos en los que iremos desarrollando el problema tratado y su solución de forma modular, formando en un fin, un proyecto sólido y totalmente funcional.

Los primeros puntos que vamos a ver serán los de Objetivos y finalidad del proyecto, en los que mostraremos lo que queremos conseguir con el desarrollo del proyecto. A continuación le seguirán los puntos con más grueso en el proyecto, que son:

- Medios materiales usados: indicaremos todo tipo de medios (humanos, hardware, software) que se usarán para la realización del proyecto, explicados con detalle.
- Planificación del proyecto: En este punto indicaremos como se va a ir desarrollando el proyecto explicado paso a paso (incluiremos código en caso de que sea necesario, diagramas, estructura de base de datos, ficheros de configuración...).
- Fase de pruebas: Servirá para describir posibles fallos en programación o configuración de la aplicación y se mostrarán las pruebas pertinentes para comprobar el correcto funcionamiento de la misma.

Después de estos puntos, irán los de Conclusiones y trabajos futuros o posibles mejoras, Apéndices (código importante que se ha hecho referencia en la memoria, manual de usuario, manual de instalación) y Referencias bibliográficas.

3. Objetivos y características del proyecto

Este punto se puede dividir en dos partes; Por una parte los objetivos generales del proyecto, y por otra los objetivos específicos (una subdivisión de los objetivos generales):

- **Objetivos generales:** Tener una plataforma de alquiler de alojamientos disponible para todo el público y que el usuario pueda tanto reservar alojamientos como crearlos.
- **Objetivos específicos:** Desarrollar una base de datos que almacene todos los datos necesarios sobre propiedades, y usuarios, crear una interfaz web para el usuario completamente responsive, realizar un CRUD (Create. Read. Update, Delete) de la base de datos sobre la interfaz gráfica, tener un control de entrada y salida de datos, validación de formularios y sesiones, implementar APIs con funcionalidades útiles para nuestro proyecto, alojar nuestro proyecto en un servidor web para hacerlo público para todo el mundo.

4. Finalidad

Como hemos mencionado en el punto anterior, el objetivo general consiste en tener una plataforma de alquiler de alojamientos disponible para todo el público y que el usuario pueda tanto reservar alojamientos como crearlos.

A esto podemos añadir la finalidad de aprendizaje y aplicación de nuevas tecnologías para el desarrollo del proyecto y así conseguir la solución del problema de la necesidad del usuario de buscar alojamientos vacacionales de una manera rápida y sencilla.

5. Medios materiales usados

Este es uno de los puntos más importantes del proyecto. Es donde definiremos todos los medios que vamos a usar en el proyecto, por lo que es importante tener claro desde un inicio y explicar con detalle todo lo que se usará.

5.1. Especificaciones hardware

Partiendo desde lo más básico a lo más específico, para el desarrollo de este proyecto usaré como hardware mi ordenador personal. No requerirá ser un hardware muy potente ni caro, en mi caso estas son las especificaciones de mi pc:

Especificaciones del PC	
Procesador	AMD Ryzen 7 2700X Eight-Core Processor 3.70 GHz
RAM instalada	16,0 GB
Tipo de sistema	Sistema operativo de 64 bits, procesador basado en x64
Edición	Windows 10 Home
Almacenamiento	500GB unidad principal + 1TB unidad secundaria
Tarjeta gráfica	NVIDIA GeForce GTX 1660

Una vez indicado el hardware que vamos a utilizar, pasemos al apartado de software. Lo primero que necesitaremos instalar es un editor de código, en este caso usaremos Visual Studio Code, pero podría valer cualquier otro.

5.2. Visual Studio Code



Logo Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

Una vez se instala, Visual Studio Code tiene multitud de extensiones que podemos usar para ayudarnos durante nuestra etapa de codificación. Nosotros usaremos los siguientes:

- **PHP Debug:** Ofrece una herramienta de depuración de código para tus archivos PHP.

- PHP Intelliphense: Intelephense es un servidor de lenguaje PHP de alto rendimiento. repleto de características esenciales para el desarrollo productivo de PHP.
- PHP IntelliSense: Soporte avanzado de autocompletado y refactorización para PHP.
- Prettier: Herramienta para dar formato a tu código.

Añadidas estas extensiones a nuestro Visual Studio Code tendremos la configuración básica de nuestro editor de código lista.

5.3. HTML



Logo HTML5

HTML, siglas en inglés de HyperText Markup Language (‘lenguaje de marcado de hipertexto’), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera

el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

HTML es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas. Este lenguaje nos ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.) No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita

en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores todavía mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, todavía existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

5.4. CSS



Logo CSS3

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS).

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en voz (mediante un navegador de voz o un lector de pantalla, y dispositivos táctiles basados en el sistema Braille. También se puede mostrar una página web de manera diferente dependiendo del tamaño de la pantalla o tipo de dispositivo. Los lectores pueden especificar una hoja de estilos diferente, como una hoja de estilos CSS guardado en su computadora, para sobrescribir la hoja de estilos del diseñador.

La especificación CSS describe un esquema prioritario para determinar qué reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas son aplicadas con un sistema llamado de cascada, de modo que las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C). El MIME type text/css está registrado para su uso por CSS descrito en el RFC 2318. El W3C proporciona una herramienta de validación de CSS gratuita para los documentos CSS.

5.5. SASS.



Logo SASS

Sass (Syntactically Awesome Stylesheets) es un lenguaje de hoja de estilos en cascada inicialmente diseñado por Hampton Catlin y desarrollado por Natalie Weizenbaum. Después de sus versiones iniciales, Nathan Weizenbaum y Chris Eppstein han continuado extendiendo Sass con SassScript, un lenguaje de script simple, usado en los ficheros Sass.

Sass es un metalenguaje de Hojas de Estilo en Cascada (CSS). Es un lenguaje de script que es traducido a CSS, SassScript es el lenguaje de script en sí mismo. Sass consiste en dos sintaxis. La sintaxis original, llamada indented syntax («sintaxis con sangrado») que usa una sintaxis similar al Haml. Este usa la sangría para separar bloques de código y el carácter nueva línea para separar reglas. La sintaxis más reciente, SCSS, usa el formato de bloques como CSS. Este usa llaves para denotar bloques de código y punto y coma (;) para separar las líneas dentro de un bloque. La sintaxis indentada y los ficheros SCSS tienen las extensiones .sass y .scss respectivamente.

CSS3 consiste en una serie de selectores y pseudo-selectores que agrupan las reglas que son aplicadas. Sass (en el amplio contexto de ambas sintaxis) extiende CSS proveyendo de varios mecanismos que están presentes en los lenguajes de programación tradicionales, particularmente lenguajes orientados a objetos, pero este no está disponible para CSS3 como tal. Cuando SassScript se interpreta, este crea bloques de reglas CSS para varios selectores

que están definidos en el fichero SASS. El intérprete de SASS traduce SassScript en CSS.

Alternativamente, Sass puede monitorear los ficheros .sass o .scss y convertirlos en un fichero .css de salida cada vez que el fichero .sass o .scss es guardado. Sass es simplemente azúcar sintáctica para escribir CSS.

La implementación oficial de Sass es open-source y está escrita en Ruby, sin embargo existen otras implementaciones, incluyendo una en PHP para Drupal.

La sintaxis indentada es un metalenguaje. SCSS es un metalenguaje anidado, lo que es válido en CSS es válido en SCSS con la misma semántica. Sass soporta la integración con la extensión de Firefox Firebug.

SassScript proporciona los mecanismos siguientes: variables, nesting (anidamiento), mixins, y herencia de los selectores.

5.6. JavaScript



Logo JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio de 2015

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

5.7. Gulp



Logo Gulp

Gulp es un kit de herramientas JavaScript de código abierto creado por Eric Schoffstall que se utiliza como sistema de compilación de transmisión en el desarrollo web front-end.

Gulp te ayuda a automatizar tareas comunes en el desarrollo de una aplicación, como pueden ser: mover archivos de una carpeta a otra, eliminarlos, minificar código, sincronizar el navegador cuando modificas tu código, validar sintaxis y un largo etcétera.

5.8. PHP



Logo PHP

PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus

Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. PHP originalmente significaba Personal Home Page (Página personal), pero ahora significa el inicialismo recursivo PHP: Hypertext Preprocessor.

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado — que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta. También se puede interpretar y ejecutar un código PHP cualquiera a través de una interfaz de línea de comandos (CLI).

El intérprete estándar de PHP, impulsado por Motor Zend, es un software libre publicado bajo Licencia PHP. PHP ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi todos los sistemas operativos y plataformas, de forma gratuita.

El lenguaje PHP evolucionó sin una especificación formal escrita o un estándar hasta 2014, con la implementación original actuando como el estándar de facto que otras implementaciones intentaban seguir. Desde 2014, se ha trabajado para crear una especificación formal de PHP.

Para abril de 2020, más de la mitad de los sitios en la web que usan PHP siguen en la versión descontinuada 5, 6 o anterior y con la versión 7.0 y 7.1 más del 68%, que no están oficialmente soportadas por The PHP Development Team, mientras que el soporte de seguridad es proporcionado por terceros, como Debian (hasta junio de 2020 para PHP 5).

Adicionalmente, la versión 7.2, la versión con soporte más popular, dejó de recibir actualizaciones de seguridad el 30 de noviembre de 2020.

5.9. XAMPP



Logo XAMPP

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

5.10. Workbench



Logo MySQL Workbench

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle.

5.11. Git



Logo Git

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

En cuanto a derechos de autor Git es un software libre distribuible bajo los términos de la versión 2 de la Licencia Pública General de GNU.

5.12. GitHub



Logo GitHub

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena típicamente de forma pública.

El 4 de junio de 2018 Microsoft compró GitHub por la cantidad de 7500 millones de dólares, al inicio el cambio de propietario generó preocupaciones y la salida de algunos proyectos de este repositorio, sin embargo no fueron representativos. GitHub continúa siendo la plataforma más importante de colaboración para proyectos Open Source.

5.13. npm



Logo npm

npm es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript, bajo Artistic License 2.0. Los desarrolladores usan npm para compartir y tomar prestados paquetes, y muchas organizaciones también usan npm para administrar el desarrollo privado.

npm consta de tres componentes distintos:

- el sitio web
- la interfaz de línea de comandos (CLI)
- el registro

Utilice el sitio web para descubrir paquetes, configurar perfiles y administrar otros aspectos de su experiencia npm. Por ejemplo, puede configurar organizaciones para administrar el acceso a paquetes públicos o privados.

La CLI se ejecuta desde una terminal y es la forma en que la mayoría de los desarrolladores interactúan con npm.

El registro es una gran base de datos pública de software JavaScript y la metainformación que la rodea.

npm se usa para:

- Adaptar paquetes de código para sus aplicaciones o incorporar paquetes tal como están.
- Descargar herramientas independientes que puede usar de inmediato.
- Ejecutar paquetes sin descargar usando npx.
- Compartir código con cualquier usuario de npm, en cualquier lugar.
- Restringir el código a desarrolladores específicos.
- Crear organizaciones para coordinar el mantenimiento, la codificación y los desarrolladores de paquetes.
- Formar equipos virtuales utilizando organizaciones.
- Administrar múltiples versiones de código y dependencias de código.
- Actualizar las aplicaciones fácilmente cuando se actualice el código subyacente.
- Buscar otros desarrolladores que estén trabajando en problemas y proyectos similares.

5.14. Chocolatey



Logo chocolatey

Chocolatey es un administrador de paquetes para Windows diseñado para permitir a los usuarios descargar e instalar aplicaciones directamente de Internet con un simple comando para PowerShell o CMD. En lugar de tener que guardar en nuestro ordenador todos los programas que bajemos (que a menudo estarán desactualizados), con esta herramienta es posible configurar en la nube.

Podemos ejecutar un simple comando desde la consola de CMD o PowerShell para instalar las últimas versiones de las aplicaciones que utilicemos a menudo. Además, Chocolatey nos permite también crear nuestros propios scripts para que, con un doble clic, podamos dejar nuestro ordenador listo.

5.16. Responsively



Logo Responsively

Esta aplicación te permite obtener una vista previa de todas las pantallas de destino en una sola ventana, una al lado de la otra, lo que reducirá considerablemente tu tiempo de desarrollo:

- Interacciones de usuario duplicadas en todos los dispositivos.
- Diseño de vista previa personalizable para satisfacer todas tus necesidades.
- Un práctico inspector de elementos para todos los dispositivos en vista previa.
- Más de 30 perfiles de dispositivos integrados con la opción de agregar dispositivos personalizados.
- Captura de pantalla con un clic de todos tus dispositivos.
- Hot-Reload compatible con desarrolladores.

5.17. 000WebHost



Logo 000webhost

000webhost es uno de los servicios de hosting gratuitos más populares, tiene mucho tiempo de vida, gracias a su intuitivo panel de control es muy sencillo de utilizar, además de que han agregado opciones para actualizar nuestra cuenta de hosting gratuito a hosting Premium, con lo cual si tenemos un proyecto que crece mucho en tráfico podemos mejorar la cuenta y obtener un mayor rendimiento de nuestro hosting.

Las características que nos ofrece 000webhost en su Free Hosting son

- 300 MB de almacenamiento.
- 10 GB de Ancho de banda.
- Hosting para 2 sitios web.
- Sin publicidad.
- Soporte de PHP y MySQL.
- 2 Bases de datos MySQL, por sitio.
- Auto instalador de WordPress.

6. Planificación del proyecto

Una vez vistos todos los medios que se usarán en el proyecto, habrá que comenzar la planificación del proyecto. Vamos a organizar la planificación en una serie de puntos clave que desarrollaremos paso a paso. Estos son:

- Diseño de la base de datos.
- Diseño de la interfaz de usuario.
- Estructura del código de la plataforma.
- Integración de la interfaz con la base de datos.
- Subida del proyecto al servidor web.

6.1. Diseño de la base de datos

En este apartado vamos a explicar detalladamente la estructura de nuestra base de datos que usaremos en nuestra plataforma de alojamientos.

La base de datos debe estar bien diseñada y se ha de tener clara su estructura desde un comienzo, puesto que cualquier mínimo cambio en ella puede implicar el rediseño de toda la página al completo.

Lo que se muestra en esta imagen es el modelo relacional diseñado en Workbench de nuestra base de datos, llamada dfind:

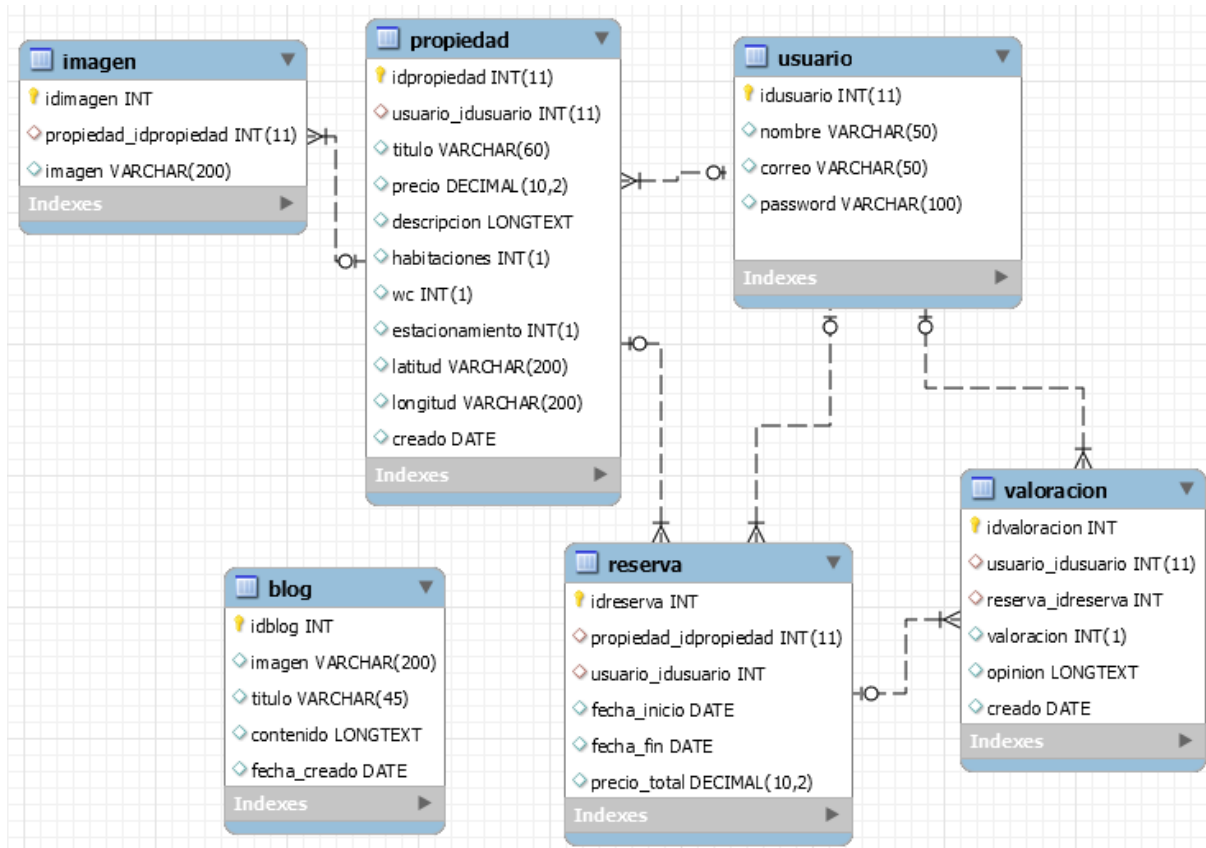


Diagrama Entidad Relación de la Base de Datos dfind

A continuación, se pasa a explicar cada una de las tablas y sus campos:

- propiedad: almacena las propiedades que tienen los usuarios. Está compuesta por:
 - idpropiedad: clave primaria de la tabla.
 - usuario_idusuario: clave foránea de usuario.
 - titulo: título de la propiedad.
 - precio: precio por día de la propiedad.
 - descripción: descripción detallada de la propiedad.
 - habitaciones: número de habitaciones de la propiedad.

- wc: número de habitaciones de la propiedad.
 - estacionamiento: número de habitaciones de la propiedad.
 - latitud: latitud de la ubicación física de la propiedad.
 - longitud: longitud de la ubicación física de la propiedad.
 - creado: fecha de creación.
- imagen: Almacena la ruta de las imágenes de cada una de las propiedades. Está compuesta por:
 - idimagen: clave primaria de la tabla.
 - propiedad_idpropiedad: clave foránea de propiedad.
 - imagen: nombre de la imagen que se almacenará en el servidor.
- blog: Almacena cada una de las entradas del blog de la plataforma. Está compuesta por:
 - idblog: clave primaria de la tabla.
 - imagen: nombre de la imagen que se almacenará en el servidor.
 - titulo: título introductorio del post.
 - contenido: contenido detallado del post.
 - fecha_creado: fecha de creación.
- usuario: Almacena a cada usuario de la plataforma. Está compuesta por:
 - idusuario: clave primaria de la tabla.
 - nombre: nombre del usuario (si no tiene añade ninguno será Anónimo).
 - correo: correo del usuario.
 - password: contraseña del usuario.
- reserva: Almacena las reservas de propiedades de cada usuario. Está compuesta por:
 - idreserva: clave primaria de la tabla.
 - propiedad_idpropiedad: clave foránea de propiedad.

- usuario_idusuario: clave foránea de usuario.
- fecha_inicio: fecha de inicio de la reserva.
- fecha_fin: fecha fin de la reserva.
- precio_total: precio total sumando los días.
- valoración: Almacena las reseñas de cada reserva pasada de los usuarios. Está compuesta por:
 - idvaloracion: clave primaria de la tabla.
 - usuario_idusuario: clave foránea de usuario.
 - reserva_idreserva: clave foránea de reserva.
 - valoración: número de estrellas que el usuario da a la propiedad.
 - opinión: reseña escrita del usuario sobre la propiedad.
 - creado: fecha de creación.

6.2. Diseño de la interfaz de usuario

En cuanto al diseño de la web, he querido realizar un par de diagramas para que quedara clara su estructura; Un diagrama de casos de uso y un diagrama de estado.

6.2.1. Diagrama de casos de uso

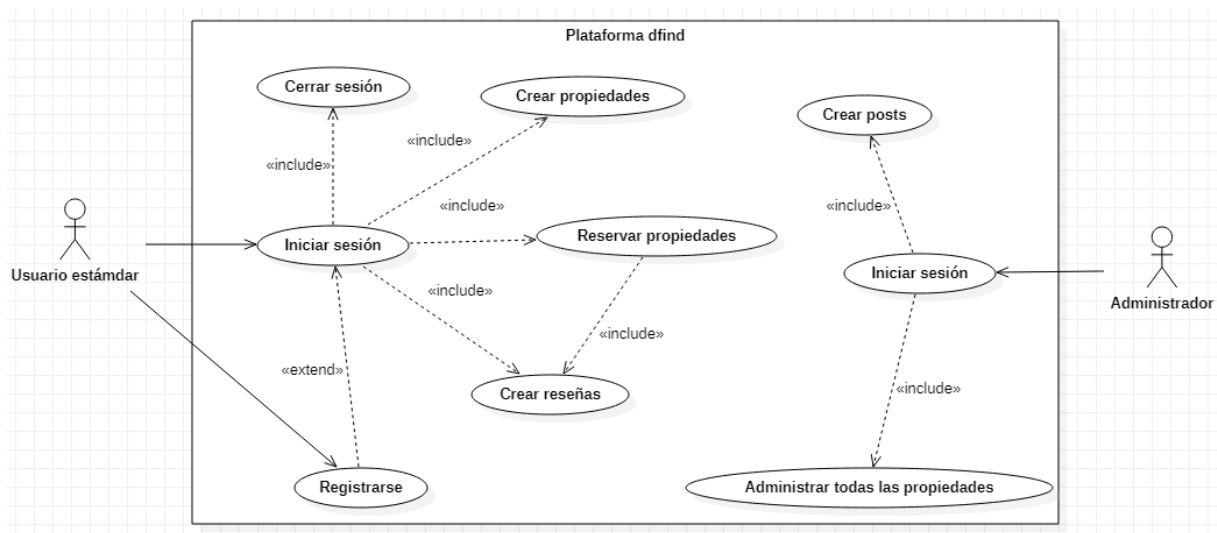
El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

En nuestro diagrama presentamos dos actores, el cliente y el administrador

- Cliente: puede realizar una serie de acciones, pero todas implican iniciar sesión o registrarse previamente excepto visualizar cada una de las propiedades y entradas de

blog disponibles. Cuando iniciamos sesión, podemos crear y eliminar reservas, comentar reservas pasadas, crear, actualizar y eliminar alojamientos propios y cerrar sesión

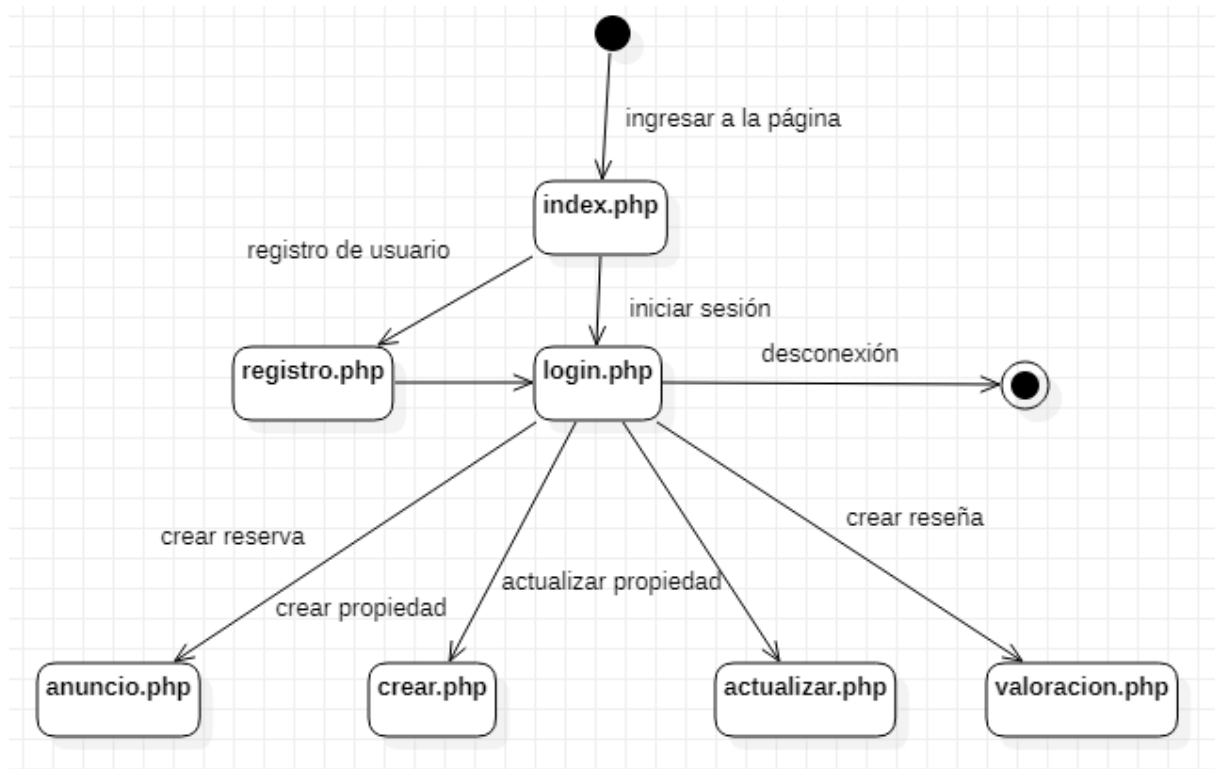
- Administrador: partiendo de la base de que tiene que iniciar sesión para realizar cualquier acción, puede, crear nuevos post del blog y administrar las propiedades de todos los usuarios (actualizar o eliminar).



6.2.2. Diagrama de estado

Al desarrollar un producto o programa informático, los diagramas de estado UML pueden ayudar a visualizar el ciclo de vida de cada objeto de forma clara y comprensible..

En nuestro diagrama hemos elegido que cada estado se corresponda a una página con acciones determinadas más allá de visualización por parte del usuario, mostrando el orden y la jerarquía de las mismas, mostrando desde el estado de inicio que es estar en la página de inicio sin iniciar sesión hasta su fin que hemos interpretado como la acción de cerrar sesión.



6.3. Estructura del código

La arquitectura web la podemos definir como la forma en que las páginas de un sitio web están estructuradas y enlazadas entre sí, de manera lógica y coherente. Una arquitectura web ideal ayuda a los usuarios a encontrar fácilmente lo que están buscando en un sitio web.

En nuestro caso usaremos una estructura simple, sin adentrarnos en una arquitectura MVC (Modelo-Vista-Controlador), ya que no es un proyecto de gran envergadura y no es muy complejo comprender el orden del mismo. La carpeta de nuestro proyecto contiene los siguientes archivos:

Nombre	Fecha de modificación	Tipo	Tamaño
.git	19/05/2021 17:17	Carpeta de archivos	
build	05/05/2021 13:30	Carpeta de archivos	
img_blog	19/05/2021 17:10	Carpeta de archivos	
img_propiedades	18/05/2021 20:08	Carpeta de archivos	
includes	05/05/2021 13:30	Carpeta de archivos	
node_modules	17/05/2021 12:21	Carpeta de archivos	
src	05/05/2021 13:32	Carpeta de archivos	
.gitignore	03/05/2021 14:46	Documento de te...	1 KB
actualizar.php	18/05/2021 15:22	Archivo PHP	9 KB
alquileres.php	19/05/2021 17:15	Archivo PHP	4 KB
anuncio.php	18/05/2021 18:14	Archivo PHP	10 KB
anuncios.php	07/05/2021 18:09	Archivo PHP	1 KB
base.php	08/05/2021 15:59	Archivo PHP	1 KB
blog.php	19/05/2021 17:04	Archivo PHP	1 KB
crear.php	19/05/2021 16:31	Archivo PHP	8 KB
crear-post.php	19/05/2021 17:38	Archivo PHP	4 KB
entrada.php	18/05/2021 16:08	Archivo PHP	1 KB
gulpfile.js	02/05/2021 13:59	Archivo JS	2 KB
index.php	19/05/2021 17:12	Archivo PHP	3 KB
login.php	19/05/2021 16:35	Archivo PHP	3 KB
logout.php	13/05/2021 10:42	Archivo PHP	1 KB
package.json	19/05/2021 12:20	Archivo JSON	1 KB
package-lock.json	19/05/2021 12:20	Archivo JSON	329 KB
registro.php	08/05/2021 12:55	Archivo PHP	3 KB
reservas.php	16/05/2021 19:32	Archivo PHP	4 KB
reservas_old.php	16/05/2021 19:32	Archivo PHP	4 KB
valoracion.php	08/05/2021 20:47	Archivo PHP	3 KB

Estructura de archivos del proyecto

6.3.1. Creación de la estructura inicial

Vamos a indicar paso a paso lo que hay que hacer para crear una estructura base en el proyecto. Como detalle a añadir, por cada cambio considerado importante en nuestro proyecto se han estado realizando los *commits* necesarios para que se guarden los distintos cambios en Git, y se ha volcado el proyecto a GitHub con *push* por cada uno de esos *commits*. Los archivos que debemos de tener en un inicio serán:

- La carpeta *src*: donde se almacenarán las carpetas *img* para las imágenes, *scss* para los archivos de estilo *.scss* y *js* para los archivos de javascript.

- El archivo `gulpfile.js`: que servirá para compilar los archivos de `scss` y `js` y cambiar el formato de las imágenes a `webp` durante el desarrollo del proyecto.

Una vez tenemos estos archivos podemos generar el archivo `package.json` introduciendo en la terminal con ubicación en la ruta de nuestro proyecto la orden *`npm init`*.

Tendremos que introducir datos que se nos irán pidiendo en la consola, como el nombre del proyecto, la versión, la descripción, el autor... etc. Cuando se finaliza se crea automáticamente el archivo `package.json` con la información introducida.

El archivo `package.json` es importante porque en el vamos a instalar las dependencias de desarrollo necesarias para nuestro archivo `gulpfile.js` que configuraremos ahora.

Para introducir las dependencias de `gulp` necesarias para nuestro archivo `gulpfile.js` solo tendremos que introducir *`npm install --save-dev gulp`*. Esto descargará los archivos necesarios para que `gulp` funcione.

El siguiente paso será instalar todas las herramientas que se necesitará `gulp` con las siguientes ordenes:

- *`npm install --save-dev gulp-sass`*
- *`npm install --save-dev autoprefixer`*
- *`npm install --save-dev gulp-postcss`*
- *`npm install --save-dev gulp-sourcemaps`*
- *`npm install --save-dev cssnano`*
- *`npm install --save-dev gulp-concat`*
- *`npm install --save-dev gulp-terser-js`*
- *`npm install --save-dev gulp-rename`*
- *`npm install --save-dev gulp-imagemin`*

- `npm install --save-dev gulp-notify`
- `npm install --save-dev gulp-cache`
- `npm install --save-dev gulp-webp`

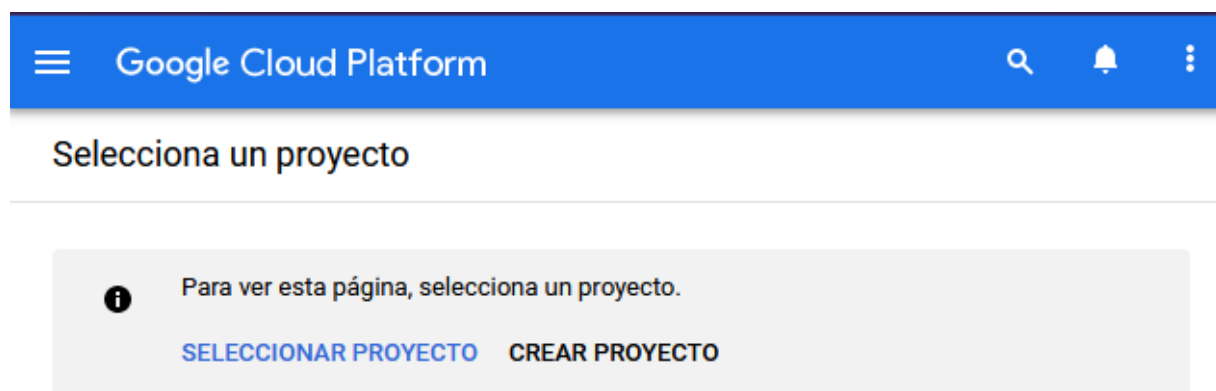
Una vez instaladas las dependencias, solo habrá que configurar el archivo `gulpfile` para que compile los archivos `scss`, las imágenes y los archivos `.js`.

*Ver archivo `gulpfile.js` en el apartado apéndices.

6.4. API Google Maps JavaScript


Para usar mapas en nuestro proyecto deberemos de activar la biblioteca de Google Maps y obtener una clave propia. Esto lo haremos a través de Google Cloud Platforms, desde nuestra cuenta de Google, en el siguiente enlace:

<https://console.cloud.google.com/projectselector2/apis/dashboard?pli=1&supportedpurview=project>. Una vez aquí, haremos clic en *CREAR PROYECTO*:




Nos llevará a un formulario en el que se pedirá el nombre del proyecto. Nosotros lo llamaremos dfind. Hacemos clic en **CREAR**:


Proyecto nuevo

 Tienes 22 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto *
dfind 

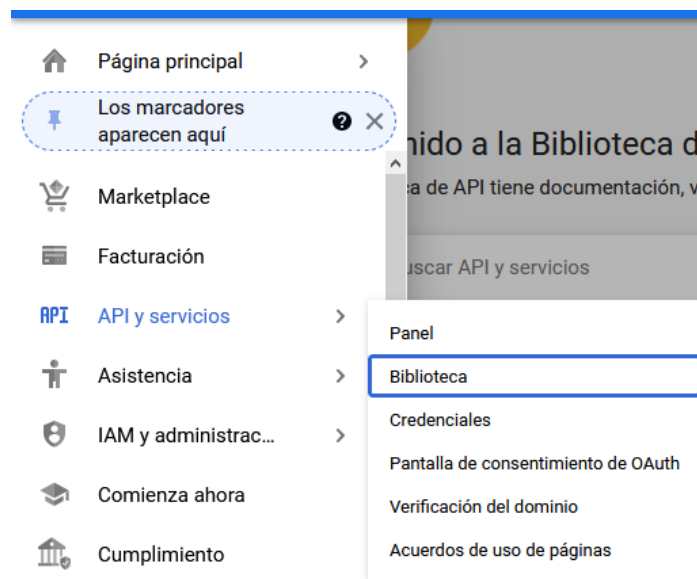
ID de proyecto: dfind-314309.No se podrá cambiar más tarde. [EDITAR](#)

Ubicación *
 Sin organización [EXPLORAR](#)

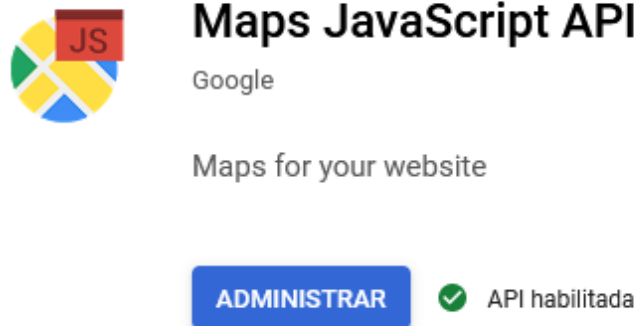
Organización o carpeta superior

CREAR **CANCELAR**

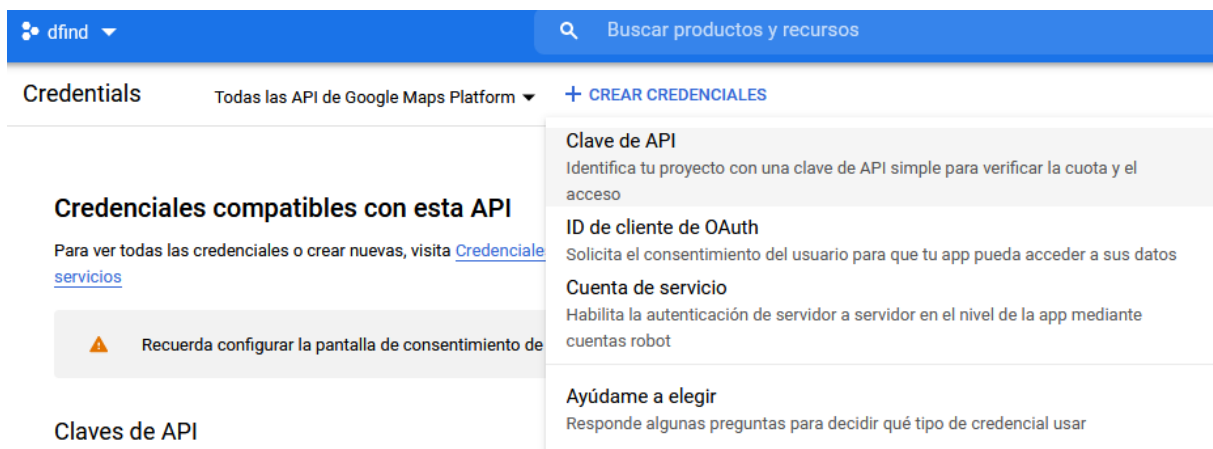
Esto nos creará el proyecto y nos llevará al panel de administración del mismo. Lo que habrá que hacer una vez aquí es ir a la sección *API y Servicios* → *Biblioteca* donde se encuentra un listado de las APIs de Google:



Aquí habrá que seleccionar la *API Maps JavaScript API* y hacer clic en habilitar:






Cuando esté habilitada, tendremos que crear unas credenciales para esa API. Esto lo haremos dentro de la sección *Credenciales* y haciendo clic en *CREAR CREDENCIALES* → *Clave de API*:



Esto nos generará una clave de API automáticamente y será la que estemos usando en nuestro proyecto cuando usemos mapas:

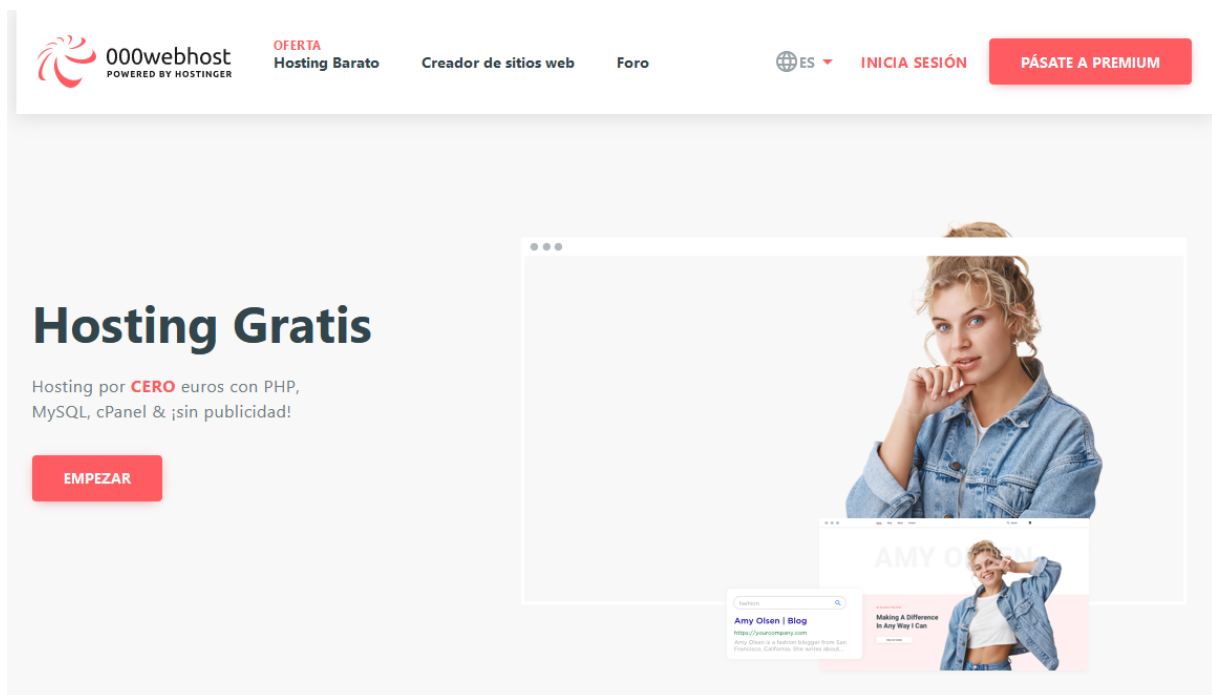
Claves de API

Nombre	Fecha de creación	Restricciones ↓	Clave	
▲ Clave de API GoogleMaps	26 abr. 2021	Ninguna	AIzaSyCass...S8naX0vdXM	  

6.5. Subida del proyecto a 000webhost

En cuanto a la subida del proyecto a un servidor público estaremos usando 000webhost debido a que es gratuito, sencillo y bastante eficiente para el uso que le vamos a dar. Su configuración y subida de archivos es muy sencilla y la estaremos viendo paso a paso en este punto

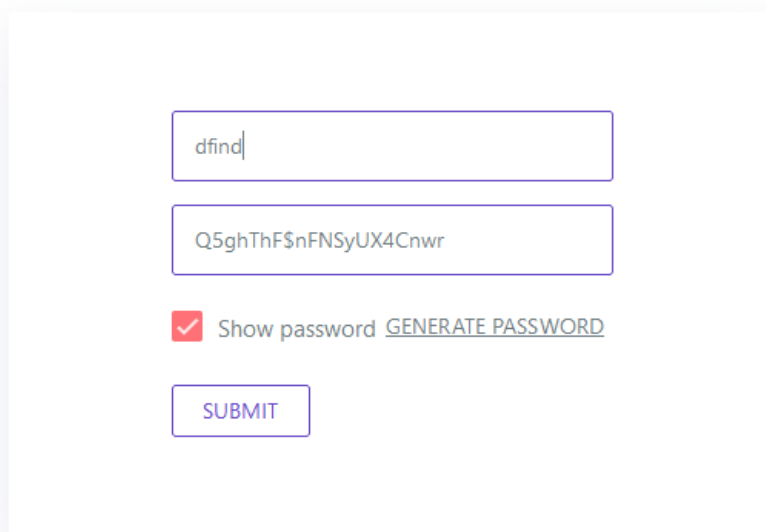
Para empezar, deberemos ingresar en la página oficial de 000webhost <https://es.000webhost.com/> y daremos clic al botón de iniciar sesión. Puedes crear una cuenta en 000webhost, pero en mi caso he accedido con mi correo de Gmail:



Si es nuestra primera vez en 000webhost nos pasará directamente al proceso de crear un proyecto, donde primeramente se nos pedirá un nombre de proyecto y una contraseña:

A great start is half the work

Name Your Project



dfind

Q5ghThF\$nFNSyUX4Cnwr

☒ Show password [GENERATE PASSWORD](#)

SUBMIT

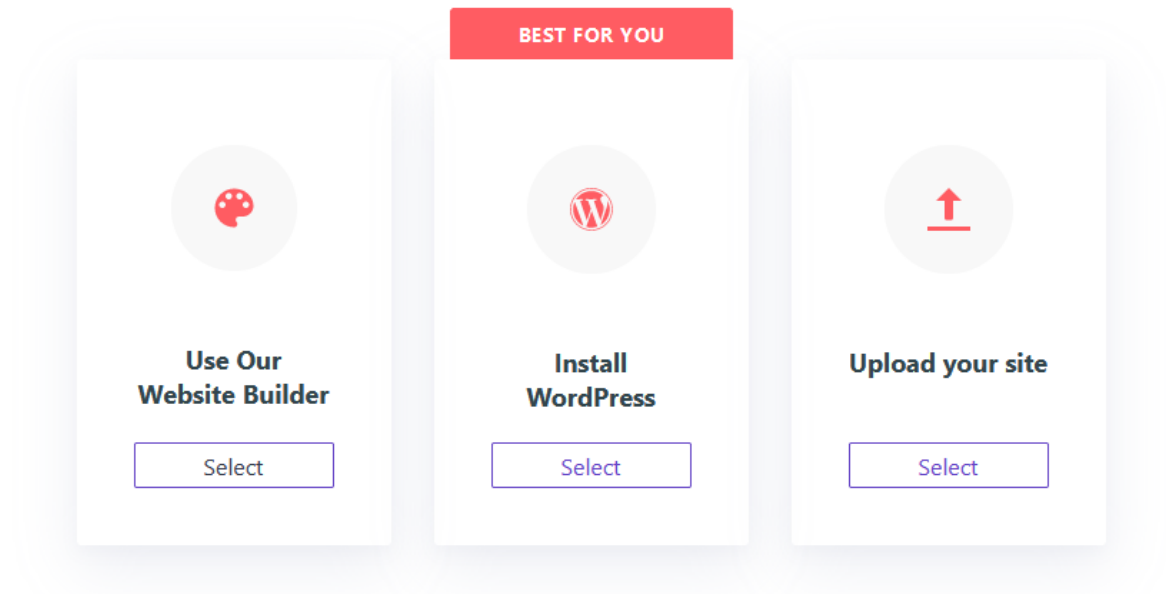
Cuando le demos a submit se creará el proyecto y luego nos dará a escoger que vamos a usar para crear nuestro sitio web:

- crearlo con su constructor de sitios web
- instalando WordPress
- Subiendo un sitio ya creado

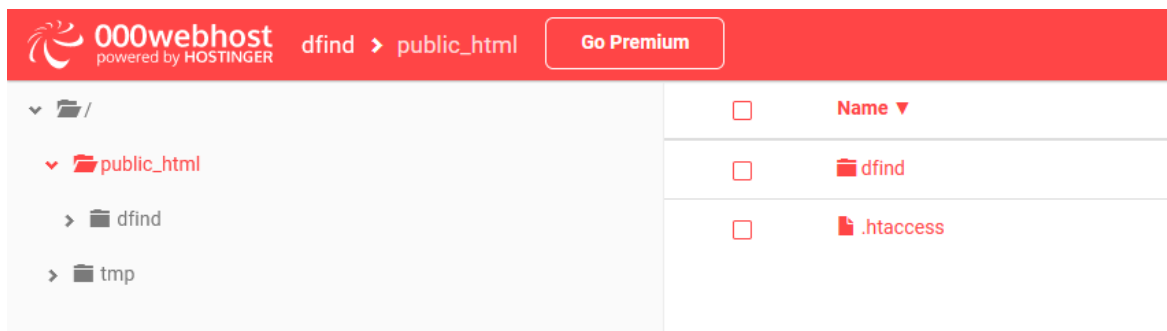
En nuestro caso escogeremos esta última, ya que ya hemos creado nuestro proyecto localmente:

Here is your tool box

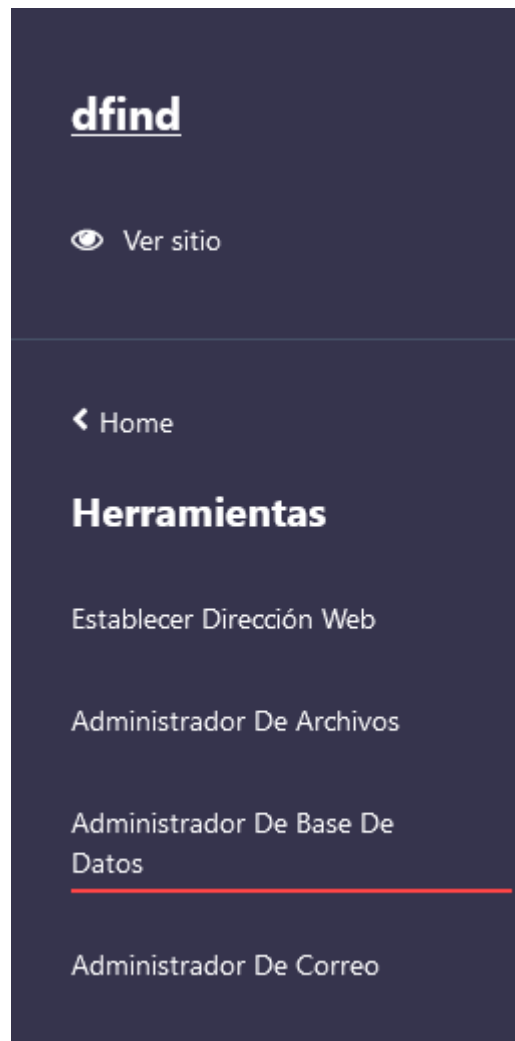
Choose your weapon of mass creation



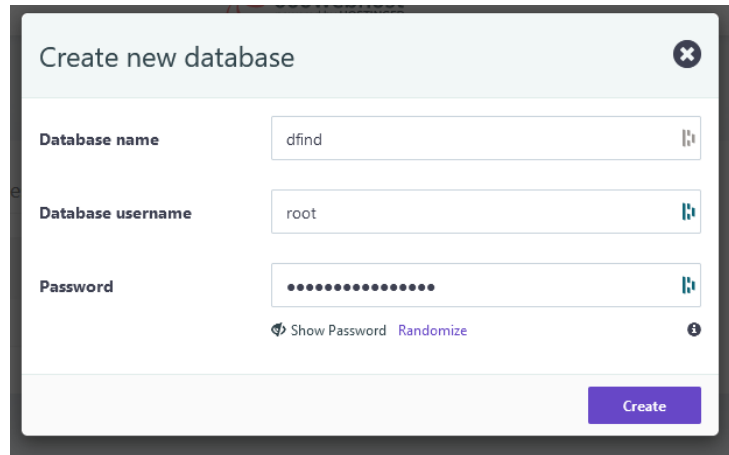
Cuando seleccionamos la opción de subir nuestro sitio, nos llevará al administrador de archivos de nuestro proyecto, donde encontraremos una carpeta llamada *public_html*. Es aquí donde subiremos los archivos de nuestro proyecto. Para ahorrar espacio, no deben subirse los archivos o directorios relacionados con la etapa de desarrollo y que no influyen en el comportamiento de la página, como *node_modules*, *gulpfile.js*, *src*, *.git* *package.json* o *package-lock.json*.



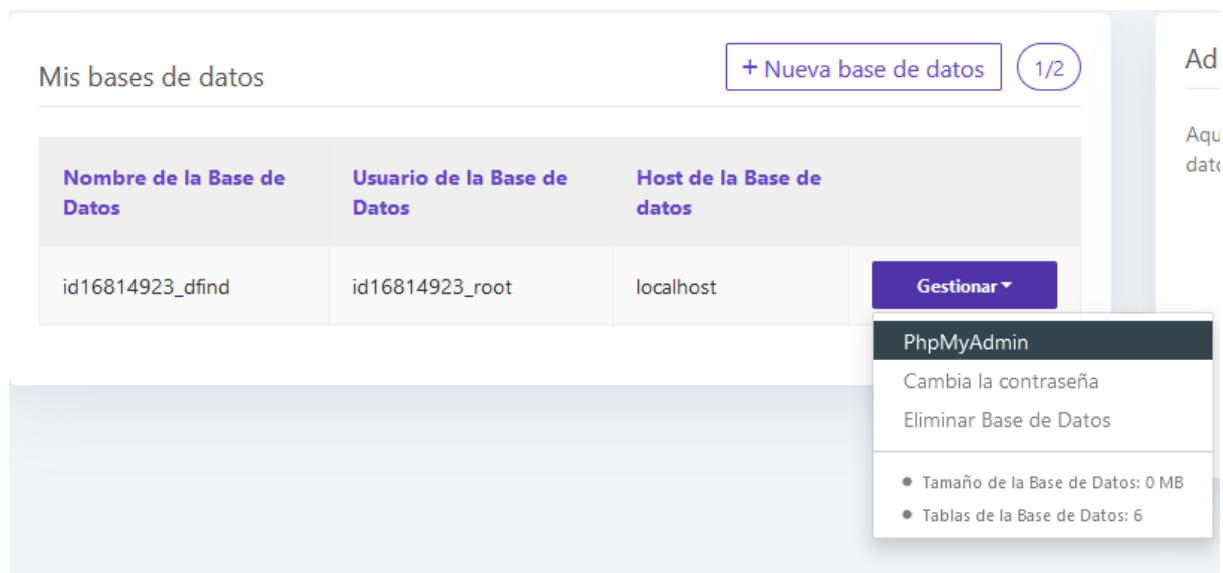
Ahora habrá que crear la base de datos de nuestro proyecto en el servidor. Para ello, dentro del apartado *Mis sitios* daremos clic en *Administrar sitio web*. Una vez aquí iremos a *Herramientas* → *Administrador De Base De Datos*.



Dentro de la sección seleccionaremos la opción de *Nueva base de datos* e introduciremos el nombre de la base de datos, el usuario y la contraseña:



Cuando se crea, se le asigna al usuario y el nombre de la base de datos un prefijo para distinguirlo de otras bases de datos que existen en el servidor. Ahora es momento de importar las tablas de nuestra base de datos local. Para ello daremos clic en *Gestionar* y le daremos a *PhpMyAdmin*.



Nombre de la Base de Datos	Usuario de la Base de Datos	Host de la Base de datos
id16814923_dfind	id16814923_root	localhost

- Tamaño de la Base de Datos: 0 MB
- Tablas de la Base de Datos: 6

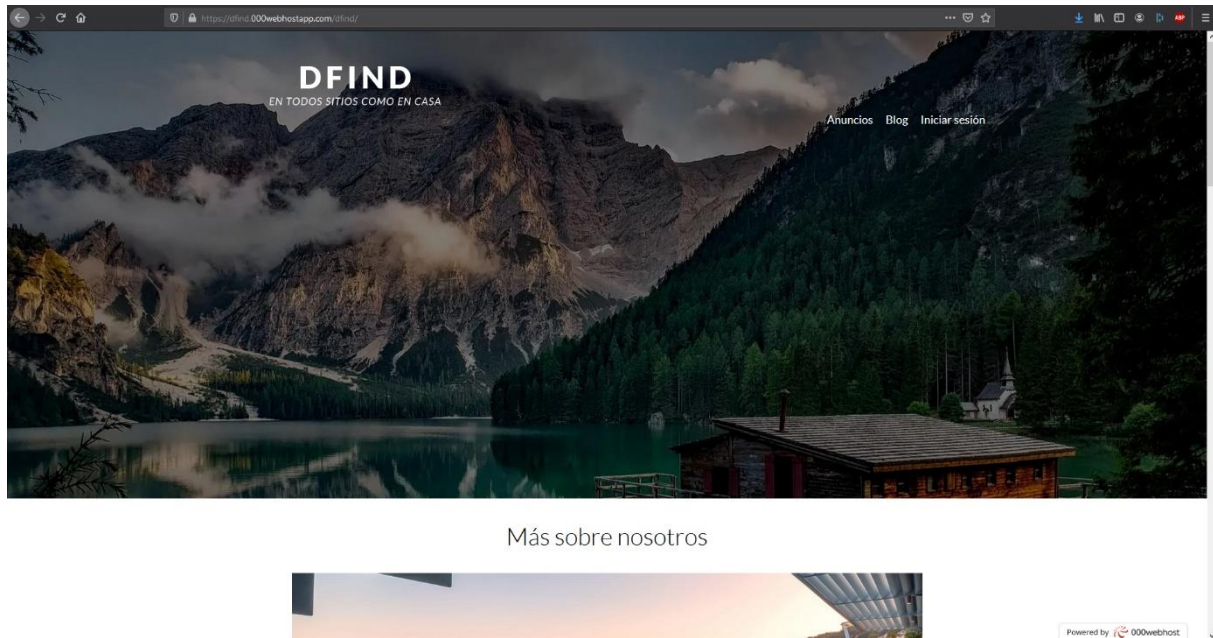
Cuando cargue por completo, seleccionaremos nuestra base de datos e iremos a la sección de Importar, donde seleccionaremos el archivo .sql de que exportamos de nuestra base de datos local:

Por último habrá cambiar el archivo de configuración de la base de datos, puesto que ya no tiene el mismo nombre debido al prefijo antes mencionado:

/public_html/dfind/includes/config/database.php

```
1 <?php
2
3 function conectarDB(): mysqli
4 {
5     $db = mysqli_connect('localhost', 'id16814923_root', 'St(*HBq(8{r?nJVk', 'id16814923_dfind');
6
7     if (!$db) {
8         echo "Error no se pudo conectar";
9         exit;
10    }
11
12    return $db;
13 }
14
```


Si todo ha ido bien, ahora podremos visualizar nuestra plataforma desde cualquier sitio introduciendo la dirección <https://dfind.000webhostapp.com/dfind>



7. Fase de pruebas

Durante la etapa de producción de nuestro proyecto hemos ido comprobando que todo se iba desarrollando correctamente, pero no basta solo con eso.

Tenemos que ver que todo funcione y todo se integre bien, es decir, que las páginas estén bien enlazadas unas con otras y funcionen correctamente,

Debemos revisar que los mensajes de validación sean los esperados, que las imágenes se muestren correctamente.

7.1. ¿Por qué es importante realizar el testing?

El testing te va a permitir errores antes de lanzar el proyecto a producción. La fase de pruebas es muy importante, ya que muchas veces, cuando se desarrolla el proyecto se van creando ciertas secciones, y conforme se van creando se prueba que haga lo que espera, pero no se tiene en cuenta que añadir nuevas funciones puede hacer que fallen otras previamente implementadas.

El problema es que cuando se añaden tantas funciones nuevas, no hay tiempo de volver a probar lo que se hizo al inicio. Probando nuestras apps podemos asegurarnos de que las nuevas funciones no afectan a lo creado anteriormente.

7.2. Tipos de testing

Veamos los diferentes tipos de prueba que existen (hay más, pero éstas son las más importantes).

7.2.1. End to End

Las pruebas de extremo a extremo son una técnica que prueba todo el producto de software de principio a fin para garantizar que el flujo de la aplicación se comporte como se espera. Define las dependencias del sistema del producto y garantiza que todas las piezas integradas funcionen juntas como se espera.

El objetivo principal de las pruebas de extremo a extremo (E2E) es probar la experiencia del usuario final simulando el escenario del usuario real y validando el sistema bajo prueba y sus componentes para la integración y la integridad de los datos.

Los sistemas de software de hoy en día son complejos e interconectados con numerosos subsistemas. Si alguno de los subsistemas falla, todo el sistema de software podría fallar. Este es un riesgo importante y puede evitarse mediante pruebas de extremo a extremo.

7.2.2. Unitarias

En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado.

Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el tipo de lo que se devuelve, que si el estado inicial es válido, entonces el estado final es válido también.

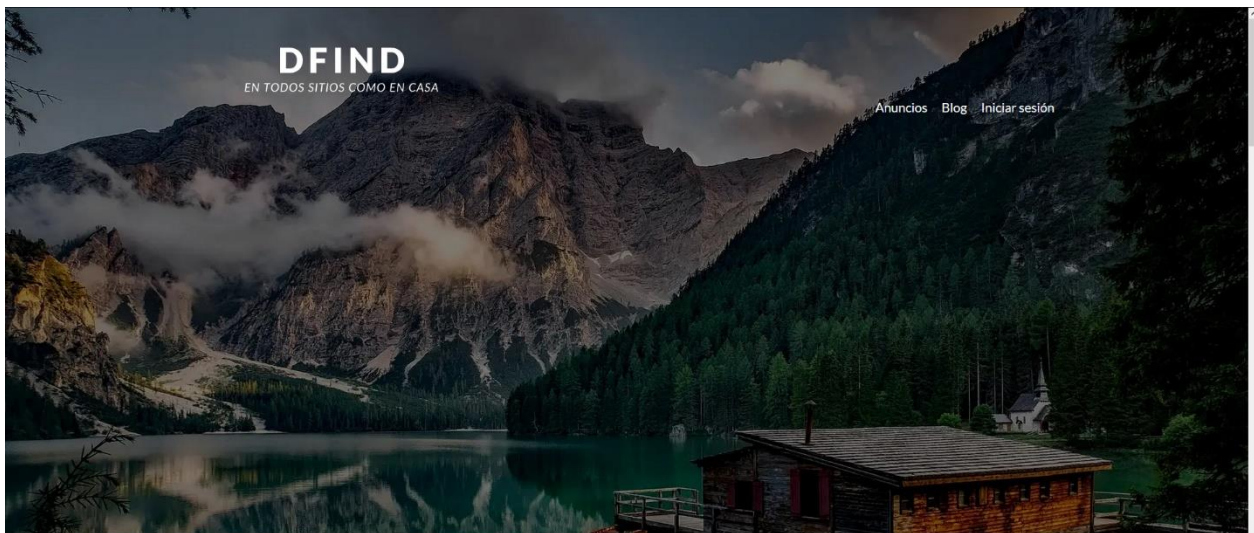
7.2.3. Integración

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban

es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo. Se centra principalmente en probar la comunicación entre los componentes y sus comunicaciones ya sea hardware o software.

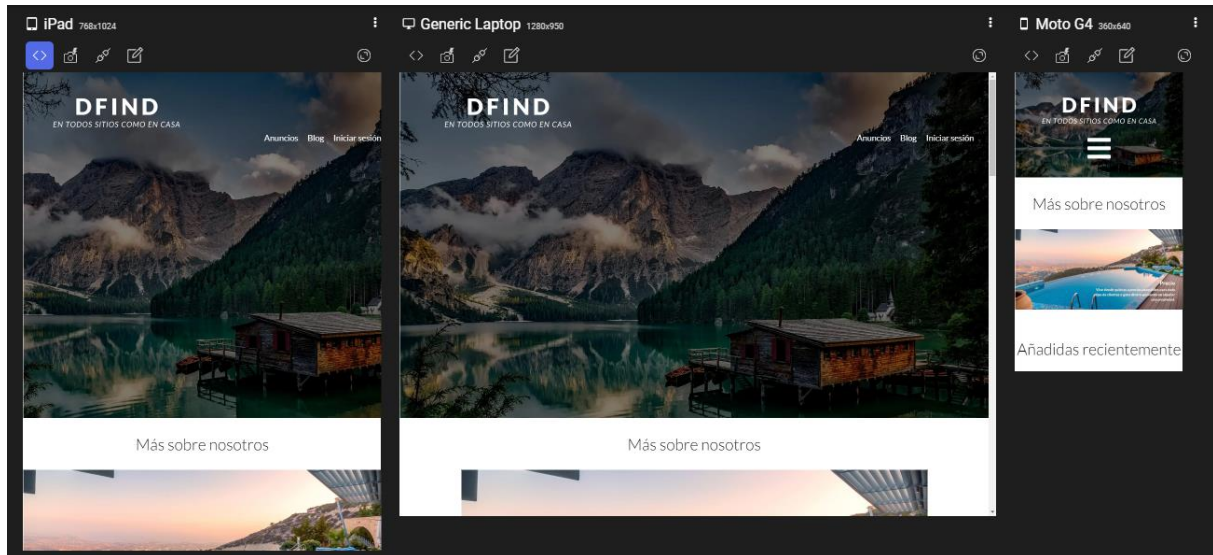
7.4. Pruebas en la plataforma

En este apartado vamos a comprobar manualmente el funcionamiento de nuestra aplicación, realizando todas y cada una de las acciones que puede hacer un usuario y verificando así el correcto funcionamiento de cada página.




Esto es lo primero que los usuarios de la plataforma verán; una cabecera con el título de nuestro proyecto y una barra de navegación a la derecha con las secciones Anuncios, Blog e Iniciar Sesión. Como podemos observar, aún no hemos iniciado sesión, cosa que nos impedirá acceder a páginas de usuarios y realizar acciones como reservar o crear un alojamiento, sin embargo, no nos impide navegar por la página para ver los distintos alojamientos.

Si estuviéramos en móvil, la página de inicio se vería con la barra de navegación escondida en un menú bocado:






Si hacemos un poco de scroll hacia abajo podemos observar un listado de los 3 últimos alojamientos registrados, pudiendo acceder a cualquiera de ellos en *Ver Propiedad* o ver un listado de todos los anuncios en *Ver Todas*.

Añadidas recientemente




Casa en plaza de la merced
Habitación privada en piso ubicado en Centro Histórico, en la Plaza de la...

25.00 € / por día




1  0  1 

[Ver Propiedad](#)




Mi casa ejemplo
Esta es mi casa de ejemplo Esta es mi casa de ejemplo Esta es mi casa de ej...

30.00 € / por día




2  1  3 

[Ver Propiedad](#)



Casa veraniega
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Provident esse fu...

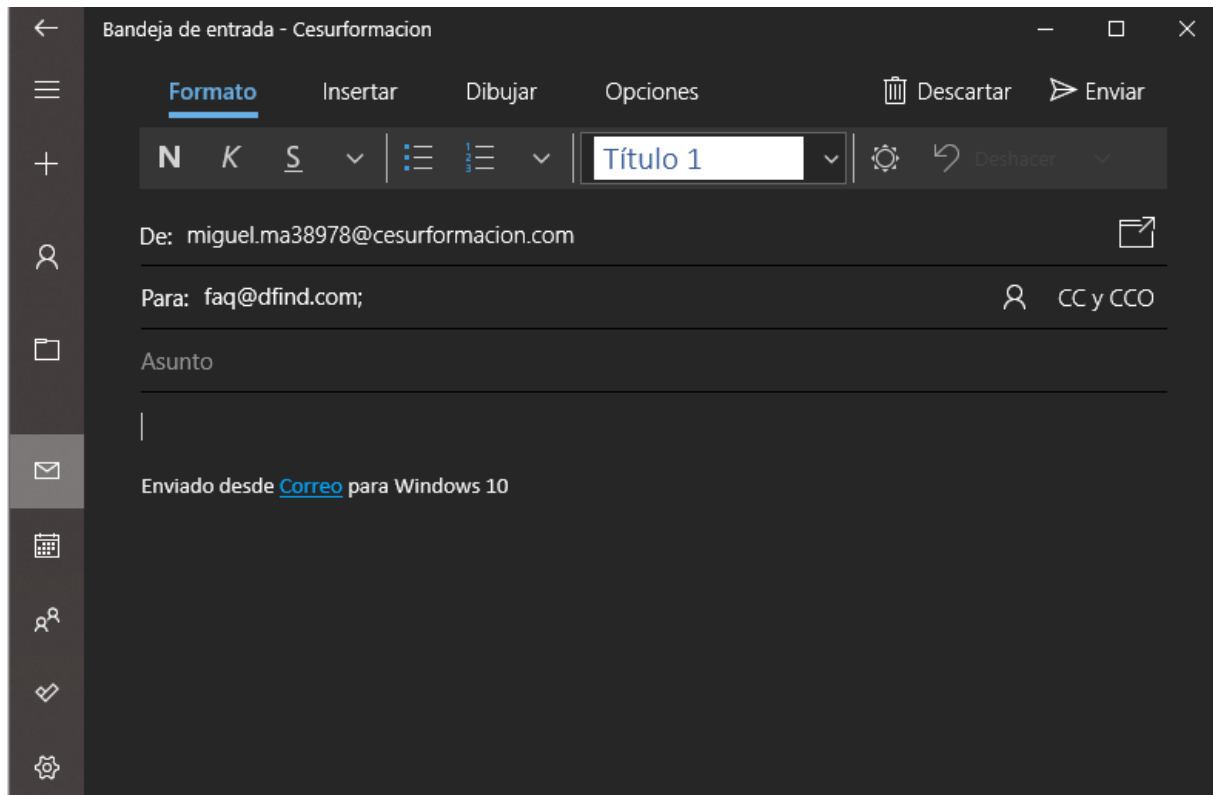
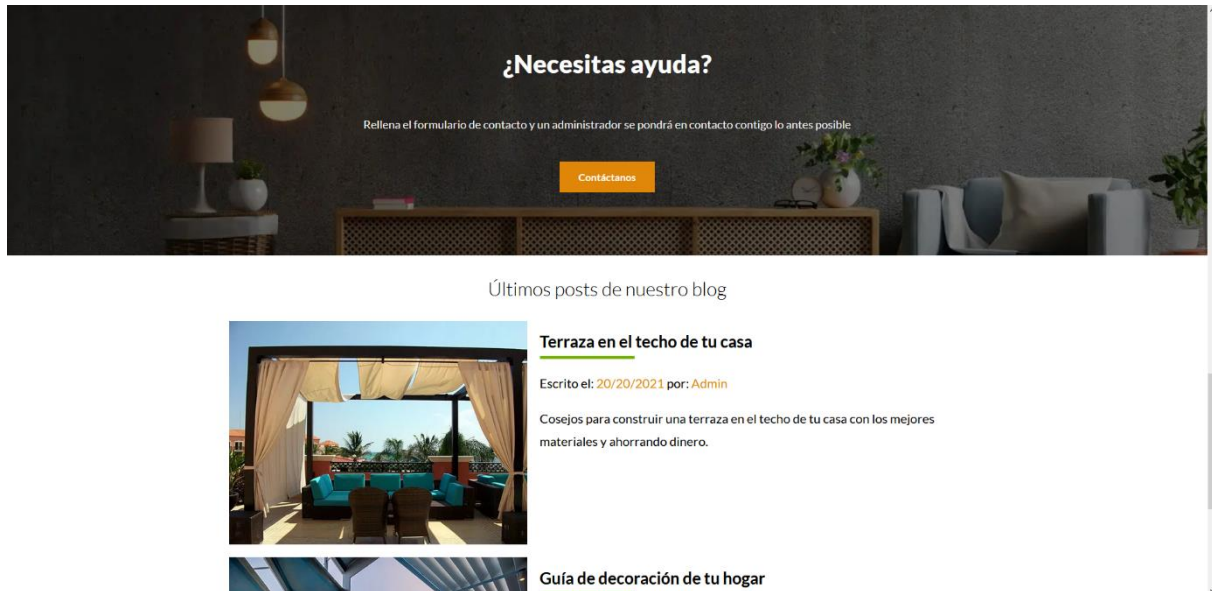
300.00 € / por día

2  1  3 

[Ver Propiedad](#)

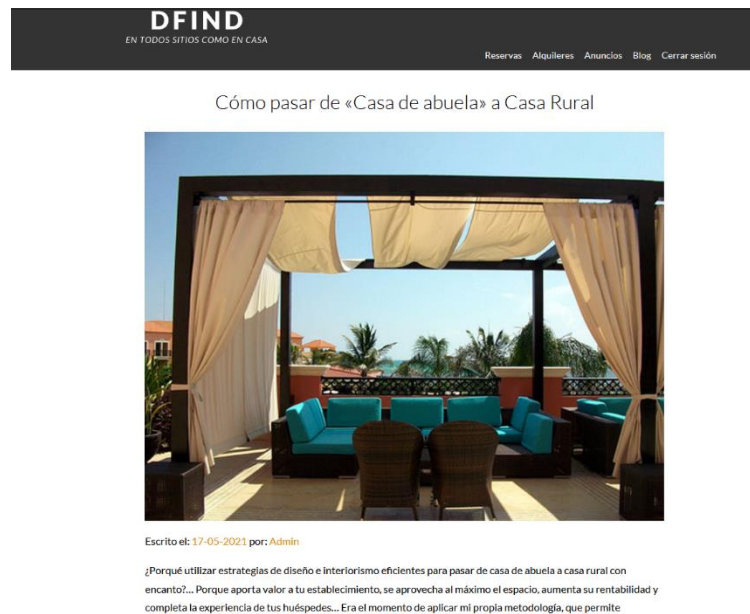
[Ver Todas](#)

Si seguimos yendo hacia abajo encontraremos un apartado de contacto, el cual nos llevará a nuestro gestor de correo predeterminado para enviar un correo a faq@dind.com.

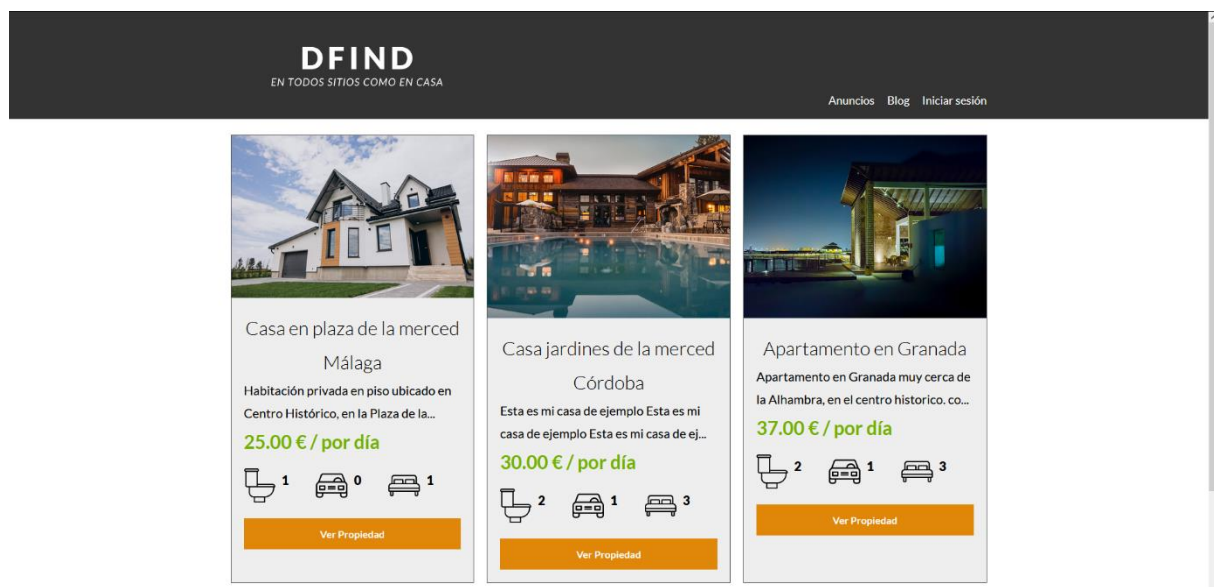


Si le damos a alguna de las entradas del blog nos mostrará la que hayamos clicado en *entrada.php*.

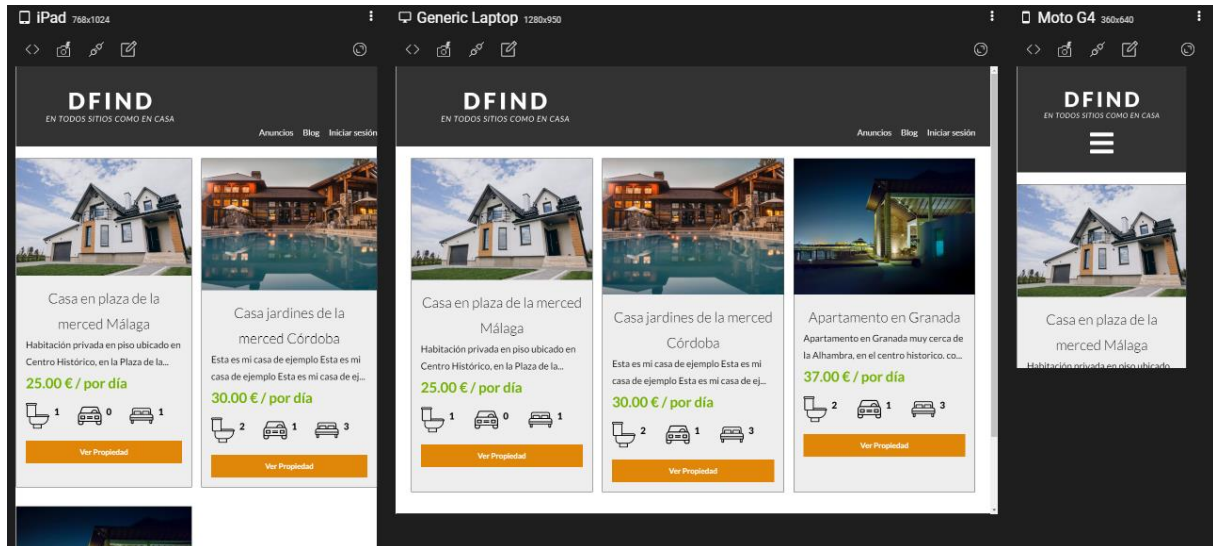
Aquí nos mostrará toda la información sobre el post seleccionado. Podemos volver a la página principal desde cualquier otra página haciendo clic al logo arriba a la izquierda.



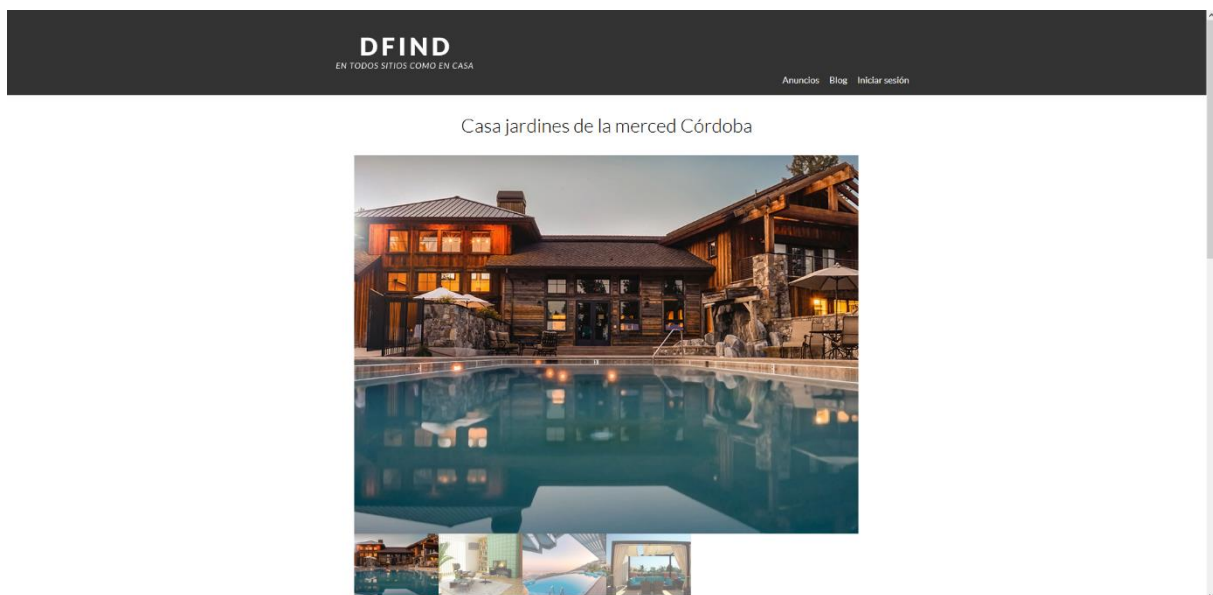
Si ahora vamos a la sección de *anuncios.php*, ya sea desde el enlace de la cabecera o desde el botón de Ver Todas de la página de inicio, se nos mostrará un listado de todos los anuncios.



Los listados de anuncios se reparten en filas de tres cuando estamos en pantalla de ordenador, de dos cuando estamos en Tablet y de uno en uno cuando estamos en móvil.



Si le damos al botón Ver Propiedad de cualquiera de los anuncios nos llevará a anuncio.php. Esta sección está compuesta de una galería de imágenes (máximo 6) de la propiedad:



Si hacemos un poco de scroll se nos mostrará información de la propiedad, como es el precio por día, la descripción detallada y las especificaciones de las habitaciones. También habrá un mapa de Google Maps con un marcador en la ubicación de la propiedad. En este mapa se puede hacer uso del StreetView para ver con más claridad de detalles el lugar:

25.00 € / por día



Descripción

Habitación privada en piso ubicado en Centro Histórico, en la Plaza de la Merced, muy cerca de bares, playas y zonas turísticas como la Alcazaba, Teatro Romano, Fundación Picasso, etc. Para hacer la estancia más agradable el piso cuenta con amplias zonas comunes muy luminosas, conexión a WIFI, una cocina completamente equipada y baños compartidos.

Ubicación



Si seguimos bajando encontraremos un formulario de reserva donde habrá de rellenar el día de ida y el de vuelta y una sección de comentarios donde los usuarios opinan sobre sus estancias en esa propiedad:

Reserva

FECHA INICIO:	FECHA FIN:
<input type="text" value="dd/mm/aaaa"/>	<input type="text" value="dd/mm/aaaa"/>
<input type="button" value="Reservar"/>	

Comentarios

Usuario: Ejemplo

Estrellas: ★★★★★

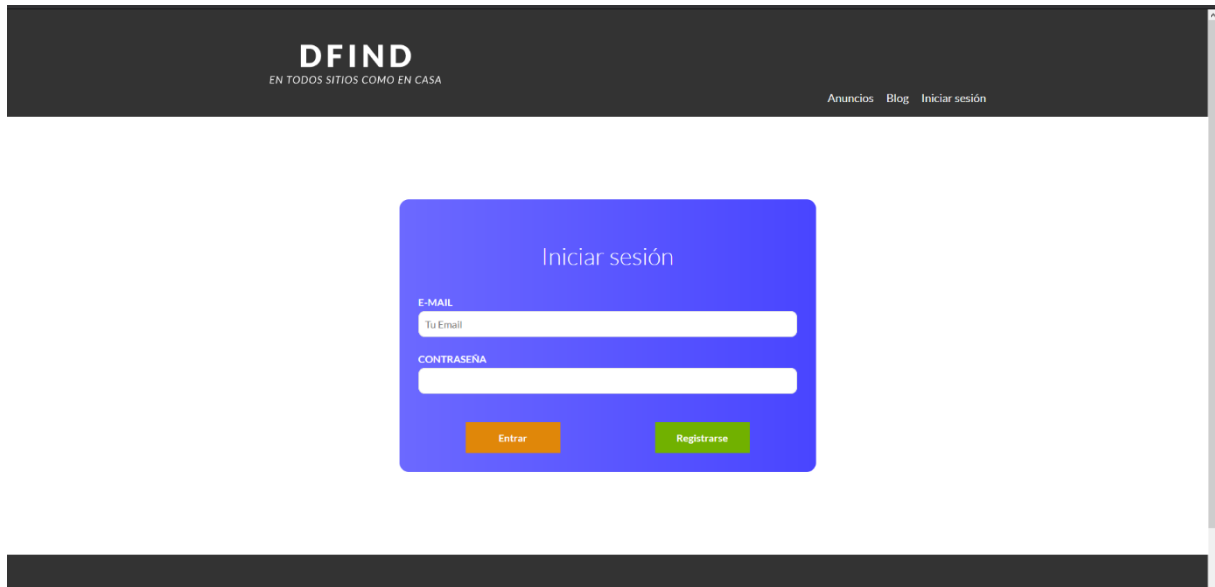
Opinión:

ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien ha estado muy bien

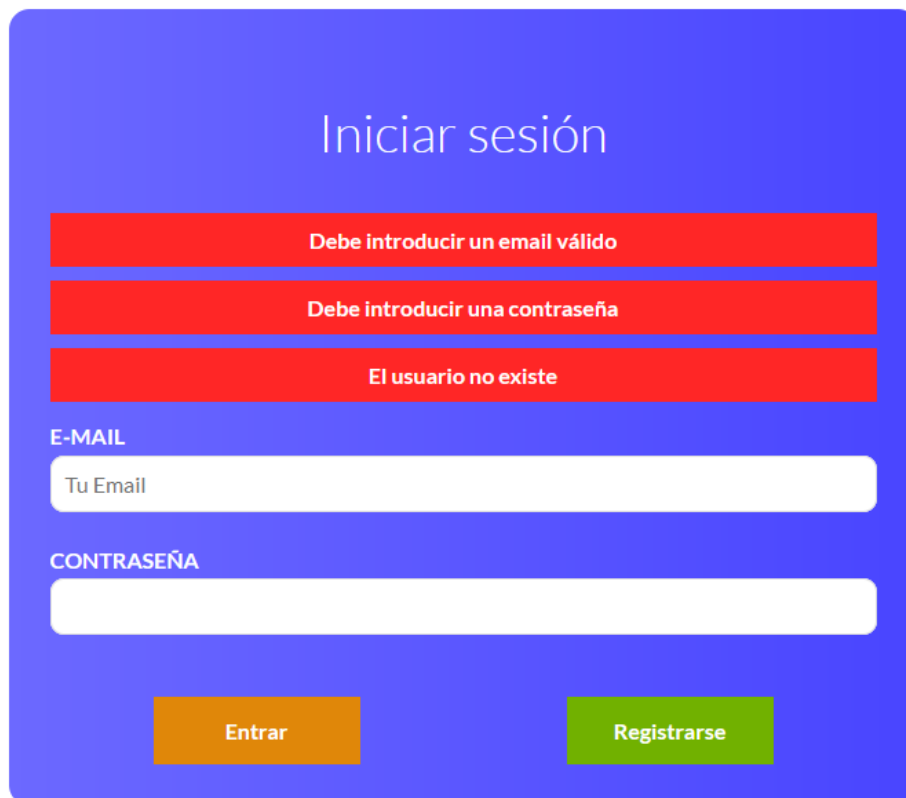
Al no haber iniciado sesión aún, si le damos al botón de *Reservar* nos mostrará un aviso de que no estamos conectados y nos redirigirá a *login.php*:

Para reservar una propiedad debe iniciar sesión.

Aquí encontraremos un formulario de registro donde nos pedirá tanto el correo como la contraseña y un botón de *Entrar* y otro de *Registrarse*:



Si le damos a *Entrar* sin escribir nada nos muestra todos los errores que puede sacar este formulario:



Como aún no tenemos usuario, vamos a crear uno dándole a *Registrarse* que nos llevará a *registro.php*.

Aquí nos pedirá un nombre, un correo y una contraseña:



Formulario de registro

NOMBRE
Tu Nombre

E-MAIL
Tu Email

CONTRASEÑA

Crear

Comprobará que el email sea válido, que la contraseña tenga más de 9 caracteres y que no exista el correo en la base de datos:



Formulario de registro

Debe introducir un email válido

Debe introducir una contraseña

La contraseña debe ser de al menos 9 caracteres

NOMBRE
Tu Nombre

E-MAIL
Tu Email

CONTRASEÑA

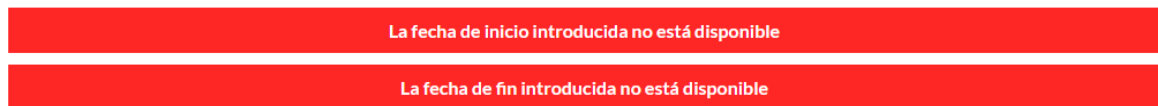
Crear

Si introducimos unos datos válidos, el usuario se creará y nos dirigirá de nuevo a la sección de inicio de sesión, donde escribiremos el usuario y la contraseña de nuevo y nos llevará a la pantalla de inicio, que ahora ha cambiado en los enlaces de la cabecera.

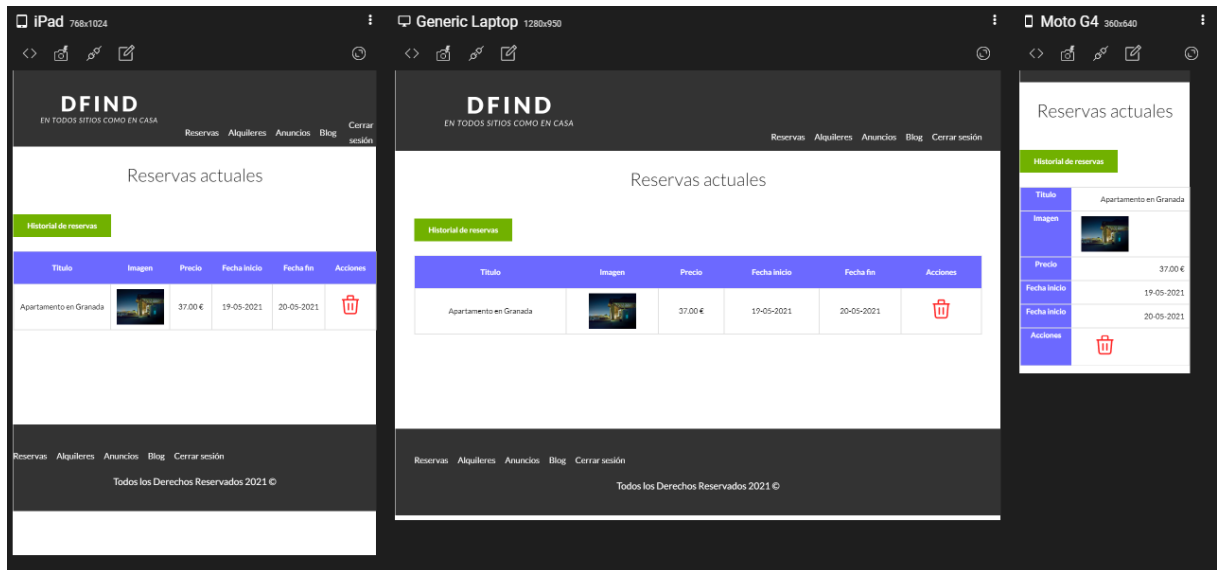
Se muestra la sección de reservas y alquileres del usuario, y ahora está la opción de Cerrar sesión:



Por lo tanto, si ahora volvemos a probar a hacer una reserva, podremos si no coincide con la fecha de reserva de otros usuarios, en caso contrario nos mostrará un error:



Si introducimos una fecha válida, se hará la reserva y se nos redireccionará a *reservas.php*, donde se nos muestra un listado de las reservas actuales y un botón para ver las pasadas. Esta tabla en una pantalla de móvil se vería en vertical en lugar de horizontal:



Podemos eliminar las reservas actuales que hemos hecho dándole a la papelera en el apartado de acciones o bien comentar una reserva pasada dándole al bacadillo de acciones, lo que nos llevará a *valoración.php*. Aquí podremos escribir una reseña de nuestra estancia y dejar una valoración en estrellas de referencia:

Dejanos tu opinión

¿Qué te ha parecido la estancia?

★ ★ ★ ★ ★

OPINIÓN

Crear

Esta sección cuenta con las siguientes alertas de control de formularios:

- Debes puntuar con estrellas la estancia
- La opinión debe contar con más de 100 caracteres

Si rellenamos todos los campos correctamente, y vamos a la propiedad que hemos comentado, nuestro comentario se verá:

Usuario: Manu (18-05-2021)

Estrellas: ★★★★★

Opinión:

La estancia ha sido genial, la casa muy acogedora y bien amueblada, pero el propietario un poco gruñón

Si en lugar de reservar alojamientos queremos poner en alquiler uno nuestro, iremos a la sección *Alquileres* en la barra de navegación, que nos llevará a *alquileres.php*, donde se muestra una tabla muy parecida a la de reservas con los campos de propiedad:

Mis alquileres

Nueva propiedad

Titulo	Imagen	Precio	Acciones
Actualmente no tiene ninguna propiedad			

Si pulsamos el botón de *Nueva Propiedad* nos llevará a *crear.php* donde se nos mostrará un formulario con todos los datos que precisa una propiedad:

Crear alojamiento

Información alojamiento

TÍTULO

Título propiedad

PRECIO

Precio por día

IMÁGENES

Examinar...

No se ha seleccionado ningún archivo.

Examinar...

No se ha seleccionado ningún archivo.

Examinar...

No se ha seleccionado ningún archivo.

Examinar...

No se ha seleccionado ningún archivo.

Examinar...

No se ha seleccionado ningún archivo.

Examinar...

No se ha seleccionado ningún archivo.

DESCRIPCIÓN

Características

HABITACIONES

Número de habitaciones

APARCAMIENTO

Plazas de aparcamiento

BAÑOS

Número de baños

Ubicación (Arrastre el marcador a su alojamiento)



Crear




Las alertas que arroja este formulario son las siguientes:

Crear alojamiento

Debes añadir un título
El campo de precio es obligatorio
Debe introducir al menos una imagen
El numero de habitaciones es obligatorio
El numero de baños es obligatorio
La ubicación del alojamiento es obligatoria

Si rellenamos los datos correctamente se añadirá la propiedad y nos llevará de nuevo al apartado de propiedades, donde podrá editarla o eliminarla al gusto:

Mis alquileres

Alojamiento registrado con éxito			
Nueva propiedad			
Título	Imagen	Precio	Acciones
Casa parque de la plata Jerez		15.00 €	 













En cuanto a la sección de administrador, se comparte la misma interfaz que la de usuario estándar, ya que no hemos requerido de muchos aspectos administrables.

Solo habrá un administrador que podrá administrar las propiedades de todos los usuarios y crear nuevos posts. Para Acceder solo habrá que introducir el correo y la contraseña del administrador en nuestra página de login.

En la sección de alquileres de nuestra página ya no solo se mostrarán las propiedades del propio usuario, si no las de todos, y podrá actualizarlas o eliminarlas cuando quiera:

Mis alquileres


Nueva propiedad

Titulo	Imagen	Precio	Acciones
Apartamento en Granada		37.00 €	 
Casa jardines de la merced Córdoba		30.00 €	 
Casa en plaza de la merced Málaga		25.00 €	 
Casa parque de la plata Jerez		15.00 €	 

Si vamos a la sección de blog, encontraremos todos los posts de la página y un botón que nos llevará a un formulario para añadir nuevas entradas de blog:

Nuestro Blog


Nuevo Post



Cómo pasar de «Casa de abuela» a Casa Rural

Escrito el: 17-05-2021 por: Admin

¿Porqué utilizar estrategias de diseño e interiorismo eficientes para pasar de casa de abuela a c...



Tendencias de color en diseño e interiorismo

Escrito el: 17-05-2021 por: Admin

En este caso, no sólo se trata de tonos. Sino que los colores, como habrás podido observar con las...

Para añadir un nuevo post haremos clic en el boton *Nuevo Post*, y nos llevará a un formulario de creación de posts, donde tendremos que introducir un título, una imagen y el contenido del post:

Crear Post

Datos del post

TÍTULO

IMAGEN

Examinar...

No se ha seleccionado ningún archivo.

CONTENIDO

Crear

Las alertas posibles del formulario son las siguientes:

Debes añadir un título

Debe introducir una imagen

El contenido del post debe ser mayor de 200 caracteres

Si hemos rellenado el formulario correctamente, le daremos al botón *Crear* y se creará nuestra entrada de blog:

Últimos posts de nuestro blog



Decoración de terraza para el verano

Escrito el: 19-05-2021 por: Admin

Las sillas de terraza, junto con los cojines para sillas de jardín y terraza, son otra forma de dec...

8. Conclusiones y trabajos futuros o posibles mejoras

En este apartado se presentan las principales conclusiones obtenidas tras la realización del proyecto y se detalla cómo se han alcanzado los objetivos específicos del proyecto.

Además, se aportan algunas posibles líneas de investigación futuras que pueden enriquecer el proyecto desarrollado.

Tal como se ha indicado durante el desarrollo del proyecto, la búsqueda de alojamientos vacacionales de corta estancia, es una costumbre que se ha ido incrementando con los años. En el apartado 3, “Objetivos y características del proyecto”. de nuestra memoria, detallamos una serie de objetivos específicos que se han ido cumpliendo paso a paso en nuestro punto 6: “Planificación del proyecto”.

Una vez desarrollado el proyecto, se han descubierto líneas de investigación con las que se puede mejorar y ampliar el espectro del problema. A continuación, se enumeran algunas de ellas:

- Utilizar la POO (Programación Orientada a Objetos) y cambiar la arquitectura del proyecto a MVC (Modelo-Vista-Controlador) para tener el código más organizado y que sea más seguro.
- Utilizar códigos QR en las direcciones físicas de cada alojamiento para que la gente que esté interesada en alojarse en esa casa solo tenga que escanear el código y ver esa propiedad en la plataforma web.
- Quitar la restricción de subida de imágenes, de manera que el usuario pueda subir todas las imágenes que quiera en lugar de solo 6 como máximo.
- Crear una sección de perfil de usuario donde el usuario pueda editar su nombre, su correo o su contraseña y que puedan añadir una imagen de perfil.

- Crear un chat en vivo en la propia aplicación para tener una mejor comunicación entre cliente y hospedador.
- Crear una plataforma de administración donde el administrador tenga un control de las estadísticas de la página, pueda modificar partes de la interfaz, controle a los usuarios y todo lo que suben a la plataforma y donde pueda crear nuevas entradas de blog.
- Añadir un botón de modo oscuro que cambie el tema de la página.
- Añadir un filtro de búsqueda según distintos parámetros (ubicación, precio, habitaciones, nombre... etc.)
- Añadir un calendario de todas las reservas de la propiedad para que el usuario sepa los días que está reservada.
- Confirmación de nuevos usuarios por envío de enlace al correo

9. Apéndices

Archivo de inicialización de la base de datos:

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-----

--
-- Estructura de tabla para la tabla `blog`
--

CREATE TABLE `blog` (
  `idblog` int(11) NOT NULL,
  `titulo` varchar(45) DEFAULT NULL,
  `contenido` longtext DEFAULT NULL,
  `imagen` varchar(100) DEFAULT NULL,
  `fecha_creado` date NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--
-- Estructura de tabla para la tabla `imagen`
--

CREATE TABLE `imagen` (
  `idimagen` int(11) NOT NULL,
  `propiedad_idpropiedad` int(11) NOT NULL,
  `imagen` varchar(200) DEFAULT NULL,
  `destacada` int(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--
-- Estructura de tabla para la tabla `propiedad`
--

CREATE TABLE `propiedad` (
  `idpropiedad` int(11) NOT NULL,
  `usuario_idusuario` int(11) DEFAULT NULL,
  `titulo` varchar(60) DEFAULT NULL,
  `precio` decimal(10,2) DEFAULT NULL,
  `descripcion` longtext DEFAULT NULL,
  `habitaciones` int(1) DEFAULT NULL,
  `wc` int(1) DEFAULT NULL,
```



```
`estacionamiento` int(1) DEFAULT NULL,
`latitud` varchar(200) DEFAULT NULL,
`longitud` varchar(200) DEFAULT NULL,
`creado` date NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Estructura de tabla para la tabla `reserva`
--

CREATE TABLE `reserva` (
  `idreserva` int(11) NOT NULL,
  `propiedad_idpropiedad` int(11) NOT NULL,
  `usuario_idusuario` int(11) NOT NULL,
  `fecha_inicio` date NOT NULL,
  `fecha_fin` date NOT NULL,
  `precio_total` decimal(10,2) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----

--
-- Estructura de tabla para la tabla `usuario`
--

CREATE TABLE `usuario` (
  `idusuario` int(11) NOT NULL,
  `nombre` varchar(50) DEFAULT NULL,
  `correo` varchar(50) NOT NULL,
  `password` char(60) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Volcado de datos para la tabla `usuario`
--

INSERT INTO `usuario` (`idusuario`, `nombre`, `correo`, `password`) VALUES
(1, 'Miguel', 'miguelschz1997@gmail.com',
'$2y$10$0VFqCgDkc5rlD6R7oLWrCOR4PA3mH1VgJ05TydTq0RuzxLnHjdInG');

-- -----

--
-- Estructura de tabla para la tabla `valoracion`
--

CREATE TABLE `valoracion` (
  `idvaloracion` int(11) NOT NULL,
  `usuario_idusuario` int(11) NOT NULL,
  `reserva_idreserva` int(11) NOT NULL,
  `valoracion` int(1) DEFAULT NULL,
  `opinion` longtext DEFAULT NULL,
  `creado` date DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
```

```
-- Indices de la tabla `blog`
--
ALTER TABLE `blog`
  ADD PRIMARY KEY (`idblog`);

--
-- Indices de la tabla `imagen`
--
ALTER TABLE `imagen`
  ADD PRIMARY KEY (`idimagen`),
  ADD KEY `fk_imagen_propiedad1_idx` (`propiedad_idpropiedad`);

--
-- Indices de la tabla `propiedad`
--
ALTER TABLE `propiedad`
  ADD PRIMARY KEY (`idpropiedad`),
  ADD KEY `fk_propiedad_usuario1_idx` (`usuario_idusuario`);

--
-- Indices de la tabla `reserva`
--
ALTER TABLE `reserva`
  ADD PRIMARY KEY (`idreserva`),
  ADD KEY `fk_reserva_propiedad1_idx` (`propiedad_idpropiedad`),
  ADD KEY `fk_reserva_usuario1_idx` (`usuario_idusuario`);

--
-- Indices de la tabla `usuario`
--
ALTER TABLE `usuario`
  ADD PRIMARY KEY (`idusuario`);

--
-- Indices de la tabla `valoracion`
--
ALTER TABLE `valoracion`
  ADD PRIMARY KEY (`idvaloracion`),
  ADD KEY `fk_valoracion_usuario1_idx` (`usuario_idusuario`),
  ADD KEY `fk_valoracion_reserva1_idx` (`reserva_idreserva`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `usuario`
--
ALTER TABLE `usuario`
  MODIFY `idusuario` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `imagen`
--
ALTER TABLE `imagen`
```

```
ADD CONSTRAINT `fk_imagen_propiedad1` FOREIGN KEY
(`propiedad_idpropiedad`) REFERENCES `propiedad` (`idpropiedad`) ON DELETE
NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `propiedad`
--
ALTER TABLE `propiedad`
ADD CONSTRAINT `fk_propiedad_usuario1` FOREIGN KEY (`usuario_idusuario`)
REFERENCES `usuario` (`idusuario`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `reserva`
--
ALTER TABLE `reserva`
ADD CONSTRAINT `fk_reserva_propiedad1` FOREIGN KEY
(`propiedad_idpropiedad`) REFERENCES `propiedad` (`idpropiedad`) ON DELETE
NO ACTION ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_reserva_usuario1` FOREIGN KEY (`usuario_idusuario`)
REFERENCES `usuario` (`idusuario`) ON DELETE NO ACTION ON UPDATE NO ACTION;

--
-- Filtros para la tabla `valoracion`
--
ALTER TABLE `valoracion`
ADD CONSTRAINT `fk_valoracion_reserva1` FOREIGN KEY (`reserva_idreserva`)
REFERENCES `reserva` (`idreserva`) ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_valoracion_usuario1` FOREIGN KEY (`usuario_idusuario`)
REFERENCES `usuario` (`idusuario`) ON DELETE NO ACTION ON UPDATE NO ACTION;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Archivo de configuración de la base de datos:

```
<?php

function conectarDB(): mysqli
{
    $db = mysqli_connect('localhost', 'id16814923_root',
'$t(*HBq(8{r?nJVk', 'id16814923_dfind');

    if (!$db) {
        echo "Error no se pudo conectar";
        exit;
    }

    return $db;
}
```

Archivo de funciones generales:

```
<?php

require 'app.php';

function incluirTemplate(string $nombre, bool $inicio = false)
{
    include TEMPLATES_URL . "/${nombre}.php";
}

function isAuth(): bool
{
    session_start();
    if (!isset($_SESSION["login"])) {
        $_SESSION["login"] = '';
    }
    $resultado = false;
    $auth = $_SESSION["login"];
    if ($auth) {
        $resultado = true;
    }
    return $resultado;
}
```

Archivo gulpfile.js:

```
const { src, dest, watch, parallel } = require("gulp");
const sass = require("gulp-sass");
const autoprefixer = require("autoprefixer");
const postcss = require("gulp-postcss");
const sourcemaps = require("gulp-sourcemaps");
const cssnano = require("cssnano");
const concat = require("gulp-concat");
const terser = require("gulp-terser-js");
const rename = require("gulp-rename");
const imagemin = require("gulp-imagemin");
const notify = require("gulp-notify");
const cache = require("gulp-cache");
const webp = require("gulp-webp");

const paths = {
    scss: "src/scss/**/*.scss",
    js: "src/js/**/*.js",
    imagenes: "src/img/**/*.",
};

// css es una función que se puede llamar automaticamente
function css() {
    return (
        src(paths.scss)
            .pipe(sourcemaps.init())
            .pipe(sass())
            .pipe(postcss([autoprefixer(), cssnano()]))
            // .pipe(postcss([autoprefixer()]))
    );
}
```

```
        .pipe(sourcemaps.write("."))
        .pipe(dest("./build/css"))
    );
}

function javascript() {
    return src(paths.js)
        .pipe(sourcemaps.init())
        .pipe(concat("bundle.js")) // final output file name
        .pipe(terser())
        .pipe(sourcemaps.write("."))
        .pipe(rename({ suffix: ".min" }))
        .pipe(dest("./build/js"));
}

function imagenes() {
    return src(paths.imagenes)
        .pipe(cache(imagemin({ optimizationLevel: 3 })))
        .pipe(dest("build/img"))
        .pipe(notify({ message: "Imagen Completada" }));
}

function versionWebp() {
    return src(paths.imagenes)
        .pipe(webp())
        .pipe(dest("build/img"))
        .pipe(notify({ message: "Imagen Completada" }));
}

function watchArchivos() {
    watch(paths.scss, css);
    watch(paths.js, javascript);
    watch(paths.imagenes, imagenes);
    watch(paths.imagenes, versionWebp);
}

exports.default = parallel(
    css,
    javascript,
    imagenes,
    versionWebp,
    watchArchivos
);
```

Archivo package.json:

```
{
  "name": "dfind",
  "version": "1.0.0",
  "description": "Plataforma de alojamiento vacacional",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "Miguel Sánchez",
  "license": "ISC",
  "devDependencies": {
```

```
"cssnano": "^4.1.10",
"gulp": "^4.0.2",
"gulp-autoprefixer": "^7.0.1",
"gulp-cache": "^1.1.3",
"gulp-clean": "^0.4.0",
"gulp-concat": "^2.6.1",
"gulp-imagemin": "^7.1.0",
"gulp-notify": "^3.2.0",
"gulp-postcss": "^9.0.0",
"gulp-rename": "^2.0.0",
"gulp-sass": "^4.1.0",
"gulp-sourcemaps": "^2.6.5",
"gulp-terser-js": "^5.2.2",
"gulp-webp": "^4.0.1",
"terser": "^5.3.8"
}
}
```

Archivo app.js:

```
document.addEventListener('DOMContentLoaded', function () {
  eventListeners();
});

function eventListeners() {
  const mobileMenu = document.querySelector('.mobile-menu');
  mobileMenu.addEventListener('click', navegacionResponsive);
}

function navegacionResponsive() {
  const navegacion = document.querySelector('.navigacion');
  navegacion.classList.toggle('mostrar');
}
```


10. Referencias bibliográficas

<https://www.php.net/manual/es/>

<https://dev.mysql.com/doc/>

<https://es.javascript.info/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://gulpjs.com/plugins/>

<https://www.youtube.com/>

<https://es.wikipedia.org/wiki/>