

# Gestión de reservas para una vivienda vacacional

Alumno: Jesús Izquierdo Puebla  
Tutor: Francisco J. Santana Pérez

GRADO EN INGENIERÍA INFORMÁTICA  
CURSO 2019/2020

Marzo 2020

# ÍNDICE

1. INTRODUCCIÓN	4
2. ESTADO ACTUAL Y OBJETIVOS	5
2.1 AIRBNB	5
2.2 TRIPADVISOR	6
2.3 VILLA GRAN CANARIA	7
3. COMPETENCIAS ESPECÍFICAS CUBIERTAS	7
3.1 COMPETENCIAS COMUNES A LA INGENIERÍA INFORMÁTICA	8
3.2 COMPETENCIAS DE LA INTENSIFICACIÓN	9
3.3 COMPETENCIAS DEL TFG	10
4. APORTACIONES AL ENTORNO SOCIO-ECONÓMICO	11
5. DESARROLLO	12
5.1 PLAN DE TRABAJO	13
5.2 PREPARATIVOS	14
5.2.1 INSTALACIÓN SERVIDOR XAMPP	15
5.2.2 INSTALACIÓN WORKBENCH	16
5.2.3 INSTALACIÓN NETBEANS	18
5.3 LENGUAJES/LIBRERÍAS/ FRAMEWORKS EMPLEADOS	17
5.3.1 HTML	19
5.3.2 CSS	19
5.3.3 BOOTSTRAP/ TempusDominus	20
5.3.4 PHP	20
5.3.5 JAVASCRIPT	21
5.3.6 LIBRERÍA TCPDF	22
5.3.7 FRAMEWORK FONT AWESOME	23
5.4 BASE DE DATOS	24

5.4.1 INTRODUCCIÓN	24
5.4.2 CREACIÓN DE LA BASE DE DATOS	25
5.4.3 TABLAS	26
5.4.4 DIAGRAMA ENTIDAD-RELACIÓN	32
5.4.5 CONEXIÓN	33
5.5 ELABORACIÓN DEL PROYECTO "Villa Sonneland"	35
5.5.1 ESTRUCTURA DE LA APLICACIÓN	35
5.5.2 RESERVAS	36
5.5.3 CONTROL DE ERRORES	43
5.5.4 MULTIIDIOMA	47
5.5.5 SOPORTE, PUNTUACIÓN Y VERIFICACIÓN DE DOBLE PASO	48
5.5.6 LOPDGDD Y EXTRAS	52
5.5.7 RESPONSIVE	57
6. CONCLUSIONES Y TRABAJOS FUTUROS	59
6.1 RESULTADOS OBTENIDOS	59
6.2 CONSECUCCIÓN DE OBJETIVOS	59
6.3 POSIBLES EXTENSIONES	60
6.4 CONCLUSIONES PERSONALES	61
7. BIBLIOGRAFÍA	62

# 1. INTRODUCCIÓN

En un entorno tan competitivo como es la hostelería, cualquier diferenciación es clave para que un negocio prospere.

Nos encontramos con una Vivienda Vacacional que está agregada a varias empresas que muestran su información para poder realizar la reserva, pero no cuenta con una web propia.

Dentro del sector de Viviendas Vacacionales, no existen muchas que cuenten con su propia aplicación web, por lo que su diseño y creación va a permitir tener una herramienta de diferenciación fundamental con respecto a la competencia.

El propósito de este Trabajo de Fin de Título es dotar a una Vivienda Vacacional, llamada "Villa Sonneland", de una aplicación web donde los futuros clientes puedan obtener información sobre la Villa (fotos, situación, contacto), así como realizar reservas y gestionirlas (cancelar, modificar).

Con el desarrollo de la aplicación, se va a dotar a "Villa Sonneland" de un recurso fundamental para desempeñar su labor, además de proporcionar un elemento diferenciador con el resto de viviendas vacacionales, debido a la escasa o nula existencia de residencias con aplicación web propia.

La aplicación ha sido diseñada para ser muy sencilla de usar:

Desde la página inicial el cliente dispondrá de un formulario con distintos campos para realizar su reserva (Calendarios de *check-in* y *check-out*, número de adultos y número de niños).

Desde una barra de navegación, el usuario podrá acceder a las distintas opciones que le da la aplicación (fotos, contacto, soporte).

Los principales usuarios van a ser extranjeros de diferentes nacionalidades, por lo que se ha dotado a la web de la opción multiidioma (Español, Inglés, Alemán).

Los distintos apartados de la aplicación web, serán tratados posteriormente en las correspondientes secciones dedicadas a ello.



## 2. ESTADO ACTUAL Y OBJETIVOS

En la actualidad, "Villa Sonneland" forma parte del grupo "Villa Gran Canaria", empresa encargada de poner en contacto a clientes con propietarios de diferentes villas repartidas por el sur de Gran Canaria. Además, se está trabajando también con las empresas Airbnb y Tripadvisor para realizar las reservas.

Desde esas páginas, el cliente realiza la reserva y contratación de diferentes servicios.

En la actualidad, existen multitud de aplicaciones que ofrecen ese servicio, pero la gran mayoría de ellas trabajan con infinidad de viviendas y no ofrecen un servicio exclusivo.

El objetivo de este trabajo es proporcionar una herramienta exclusiva para "Villa Sonneland" con la cual dotar al negocio, si un día fuera necesario, de una independencia con la que no cuenta en la actualidad.

Además, el disponer de una plataforma propia, le permitirá establecer los precios y condiciones que se deseen sin tener que contar con la participación de terceros.

Con esto, no se pretende en ningún momento separarse del resto de plataformas, sino de disponer de otra herramienta extra para los clientes.

Las principales empresas que ofrecen una plataforma para el alquiler turístico son Airbnb, Tripadvisor y Villa Gran Canaria:

### 2.1 AIRBNB

Airbnb es la empresa que domina el mercado del alquiler de Viviendas Vacacionales. Fundada en el año 2008, tiene presencia en cerca de 200 países y cuenta con alrededor de 2 millones de viviendas.

A diferencia de otros negocios como Tripadvisor, su negocio principal es el alquiler de viviendas, permitiendo posteriormente su puntuación.



## 2.2 TRIPADVISOR

Tripadvisor es una empresa fundada en el año 2000, la cual ofrece reseñas de clientes sobre diferentes negocios (Restaurantes, Hoteles, Viviendas vacacionales). Además de ofrecer reseñas, permite realizar reservas en los citados negocios.

Al igual que Airbnb, ofrece sus servicios a nivel mundial.



## 2.3 VILLA GRAN CANARIA

Villa Gran Canaria es una empresa que lleva trabajando desde el año 2002.

Su negocio se centra en el alquiler de viviendas, principalmente de lujo, en el sur de Gran Canaria poniendo en contacto a clientes con propietarios. En los últimos tiempos, su negocio se ha ampliado a las islas de Fuerteventura y Lanzarote.



## 3. COMPETENCIAS ESPECÍFICAS CUBIERTAS

Según el documento “**Objetivos y competencias del Grado en Ingeniería Informática de la ULPGC**” el futuro Ingeniero Informático debe reunir diversas competencias las cuales vienen clasificadas en 3 apartados distintos:

- Competencias comunes a la Ingeniería Informática.
- Competencias de la intensificación.
- Competencias del Trabajo de Fin de Título.

Pasaremos a hablar de las competencias cubiertas en cada uno de los apartados, así como la explicación del motivo por el que esa competencia es cubierta en la realización de este trabajo:

## 3.1 COMPETENCIAS COMUNES A LA INGENIERÍA INFORMÁTICA

**CII01:** *“Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.”*

El proyecto ha sido diseñado y desarrollado haciendo uso de los conocimientos adquiridos durante el período de formación académica. Para su desarrollo se ha tenido en cuenta la legislación y la normativa vigente como la "Ley de Protección de Datos", ya que se está almacenando información personal.

**CII02:** *“Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.”*

El proyecto ha sido planificado, concebido y dirigido por mí en todo momento. Se ha iniciado en un servidor local con la intención de ser desplegado en un servidor hosting, además se ha tenido en cuenta la mejora continua en base a las opiniones del cliente.

**CII03:** *“Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.”*

Durante el desarrollo del trabajo se realizaron reuniones con el cliente para detallar los datos del proyecto e informarle, de forma clara y concisa, de las posibilidades y limitaciones del sistema. El cliente ha validado las proposiciones del desarrollo planteado.

**CII07:** *“Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema”*

Durante el desarrollo del trabajo se han usado diversos *arrays* como estructuras de datos en un sinnúmero de casos como por ejemplo:

Se hace uso de *arrays* asociativos para almacenar las diferentes traducciones de la web (Español, Inglés, Alemán).

Las fechas de reservas contratadas son almacenadas en *arrays* para ser pasadas a los diferentes calendarios que muestran los días disponibles. La representación de la información dentro del *array* tiene una forma específica dada por el calendario *datepicker*.

**CIH012:** *“Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.”*

Las bases de datos son parte fundamental del proyecto ya que es donde se va a proceder a almacenar la información necesaria. La Base de Datos empleada en este proyecto es “MySQL”.

**CIH017:** *“Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.”*

Toda la interfaz que se muestra al cliente ha sido diseñada para este proyecto con el objetivo de que fuese accesible y usable. El cliente ha validado el modelo propuesto.

**CIH018:** *“Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.”*

Para el buen desarrollo de este proyecto, se ha hecho uso en todo momento de la "Ley de Protección de Datos", informando al cliente de ello y garantizando su aplicación.

La información de un cliente no podrá ser almacenada si no da su consentimiento previo.

## 3.2 COMPETENCIAS DE LA ESPECIFICACIÓN “SISTEMAS DE LA INFORMACIÓN”

**SI03:** *“Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.”*

Al tratarse de un trabajo individual, las actividades de especificación y diseño de implementación han sido realizadas por el desarrollador del proyecto.

En un futuro, se pretende pasar a producción, por lo que el mantenimiento será parte fundamental.

**SI04:** *“Capacidad para comprender y aplicar los principios y prácticas de las organizaciones, de forma que puedan ejercer como enlace entre las comunidades técnica y de gestión de una organización y participar activamente en la formación de los usuarios.”*

Durante el desarrollo de este trabajo, se han establecido diferentes reuniones con personas sin conocimientos previos de Ingeniería, por lo que ha sido fundamental ejercer como enlace entre la parte técnica y la parte a nivel de usuario.

### 3.3 COMPETENCIAS DEL TRABAJO DE FIN DE GRADO

**TFG01:** *“Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.”*

El trabajo “Gestión de reservas para una Vivienda Vacacional” es un trabajo original propuesto por el propio alumno (No constaba en la lista de trabajos propuestos).

Ha sido desarrollado de forma individual y va a ser defendido frente a un tribunal universitario.

Para su desarrollo, se ha hecho uso de los conocimientos adquiridos durante mi estancia en la universidad, sobre todo en el campo de la programación, aunque se han aplicado conocimientos de diversas materias como son las Bases de datos para almacenar la información, Ingeniería del *software*, Diseño de interfaces y Redes de comunicaciones para la configuración de los

diversos ficheros que permiten poner en funcionamiento un servidor de correo electrónico.

## 4. APORTACIONES AL ENTORNO SOCIO-ECONÓMICO

La posibilidad de disponer de una aplicación web propia en un sector como el turístico es una herramienta indispensable, ya que te da una mayor visibilidad y proporciona la posibilidad de resaltar los aspectos más relevantes de tu negocio.

Dentro del sector de Viviendas Vacacionales, la gran mayoría cuentan con su apartado dentro de webs mucho más grandes donde no se tiene tanto cuidado a la hora de mostrar la vivienda a los clientes de la forma en la que le gustaría al propietario de la misma, por lo que encontrar una Vivienda Vacacional que cuente con su propia aplicación es una rareza en este sector tan concurrido.

Esto permite a los clientes hacerse una mejor idea de:

- El entorno donde se encuentra la vivienda.
- Aspecto real de la vivienda.
- Posibilidad de realizar la reserva.
- Contacto directo con el propietario.
- Gestión de forma sencilla de sus reservas.
- Conocimiento en todo momento de los Términos y Condiciones.

En cuanto al aspecto económico, si en algún momento "Villa Sonneland" se quisiese independizar, le permitiría recibir el 100% del valor de la reserva,

ya que actualmente el 20% del valor de la reserva forma parte de Villa Gran Canaria por ser ella la que provee de clientes a "Villa Sonneland".

## 5. DESARROLLO

Durante el siguiente apartado vamos a ir desgranando cómo se ha desarrollado el proyecto en sus distintas fases.

Una vez explicado las distintas fases del proyecto, nos centraremos en el funcionamiento del mismo, así como la explicación de los distintos lenguajes de programación empleados, librerías, *frameworks*, bases de datos y todas las herramientas que han sido necesarias para realizar la aplicación web.

Nos apoyaremos en distintas imágenes para que su lectura sea más amena y de fácil comprensión.

Este trabajo se ha desarrollado en un equipo con las siguientes características:

Sistema operativo: Windows 10 Pro 64 bits (10.0, compilación 18362)

Idioma: español (configuración regional: español)

Fabricante del sistema: Gigabyte Technology Co., Ltd.

Modelo del sistema: Z270X-Gaming K5

BIOS: F4

Procesador: Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz (8 CPUs), ~4.2GHz

Memoria: 16384MB RAM

Archivo de paginación: 9347MB usados, 11052MB disponibles

Versión de DirectX: DirectX 12



Nombre:	NVIDIA GeForce GTX 1070
Fabricante:	NVIDIA
Tipo de chip:	GeForce GTX 1070
Tipo de DAC:	Integrated RAMDAC
Tipo de dispositivo:	Dispositivo de pantalla completa
Memoria total aprox.:	16239 MB
Memoria de pantalla (VRAM):	8088 MB
Memoria compartida:	8151 MB

## 5.1 PLAN DE TRABAJO

La elaboración de este proyecto ha constado de unas 300 horas, resumidas en la siguiente tabla.

<b>Fases</b>	<b>Duración Estimada (horas)</b>	<b>Tareas (nombre y descripción, obligatorio al menos una por fase)</b>
Estudio previo / Análisis	15	Tarea 1.1: Análisis de viabilidad
		Tarea 1.2: Formación en tecnología Bootstrap
		Tarea 1.2: Formación en librería TCPDF
Diseño / Desarrollo / Implementación	245	Tarea 2.1: Instalación y configuración del sistema para la puesta en marcha del proyecto
		Tarea 2.2: Creación de la base de datos y las distintas tablas.
		Tarea 2.3: Diseño aspecto gráfico de la aplicación Web
		Tarea 2.4: Desarrollo e implementación apartado crear reserva
		Tarea 2.5: Desarrollo e implementación apartado gestionar reserva
		Tarea 2.6: Desarrollo e implementación apartado recuperación código de reserva
		Tarea 2.7: Desarrollo de la factura mediante la librería TCPDF
		Tarea 2.8: Adaptar la página a multilenguaje. (Español, Inglés, Alemán)
		Tarea 2.9: Implementación de las reglas de negocio
Evaluación / Validación / Prueba	20	Tarea 3.1: Migración de la aplicación de un sistema local a un servidor web
		Tarea 3.2: Pruebas de correcto funcionamiento
Documentación / Presentación	20	
		Tarea 4.2: Memoria final
		Tarea 4.3: Elaboración presentación

## 5.2 PREPARATIVOS

Previo a la puesta en marcha del proyecto, se ha tenido que realizar la instalación de las diferentes herramientas necesarias.

## 5.2.1 INSTALACIÓN SERVIDOR XAMPP

XAMPP es un servidor libre y gratuito que permite instalar Apache (Servidor web) en tu equipo. Se trata de un servidor muy versátil y multiplataforma (Windows Mac Os, Linux).

XAMPP incluye intérprete de PHP (Lenguaje que se ha usado para la comunicación con el servidor), servidor de Bases de datos MySQL con el gestor phpMyadmin.

XAMPP incluye multitud de herramientas extra que no se van a explicar ya que no han sido usadas para el desarrollo de este trabajo.

La instalación se realiza desde la página web “[www.apachefriends.org](http://www.apachefriends.org)”.

En este caso nos hemos decantado por la última versión existente a día de hoy: 7.3.11



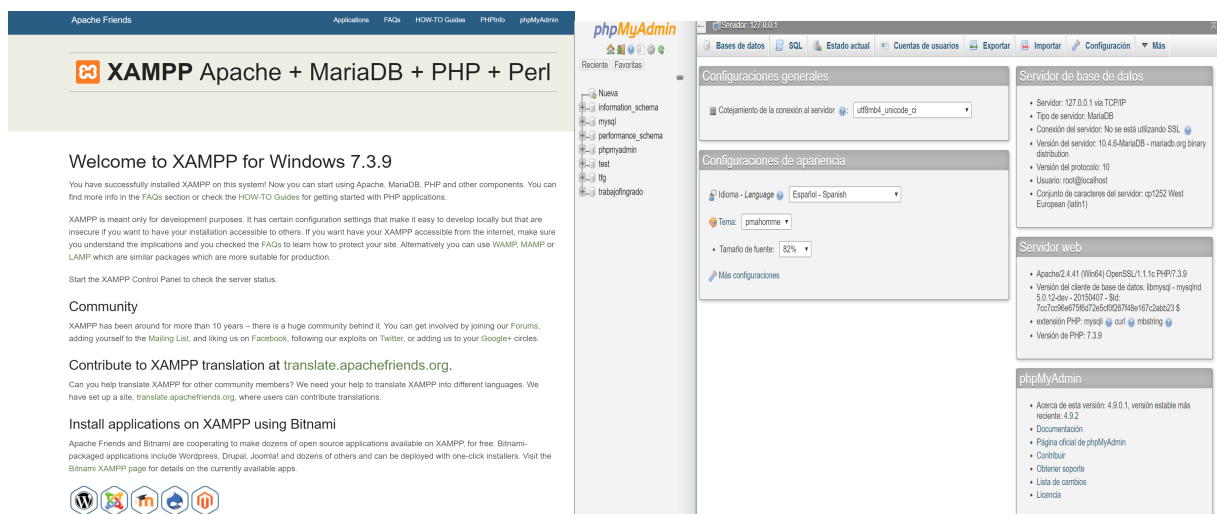
The screenshot shows the XAMPP website homepage. At the top, there is a header with the XAMPP logo (an orange square with a white 'X') and the text "XAMPP Apache + MariaDB + PHP + Perl". Below the header, on the left, is a section titled "¿Qué es XAMPP?" with a description: "XAMPP es el entorno más popular de desarrollo con PHP" and "XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar." To the right of this text is a video player with the title "Introduction to XAMPP" and a play button. At the bottom, there are four buttons: a green "Descargar" button with the text "Pulsa aquí para otras versiones", and three grey buttons for "XAMPP para Windows 7.3.11 (PHP 7.3.11)", "XAMPP para Linux 7.3.11 (PHP 7.3.11)", and "XAMPP para OS X 7.3.11 (PHP 7.3.11)".

Una vez instalado el paquete XAMPP se procede a levantar los servidores que se van a usar (Apache y MySQL) y comprobar si la instalación ha sido correcta.



Para comprobar si Apache ha sido instalado correctamente en el navegador de nuestro equipo escribimos **localhost** y nos da como resultado esta imagen, que verifica que Apache esta en funcionamiento.

Para comprobar si el servidor MySQL está activo, hacemos lo mismo en la dirección localhost/phpmyadmin.



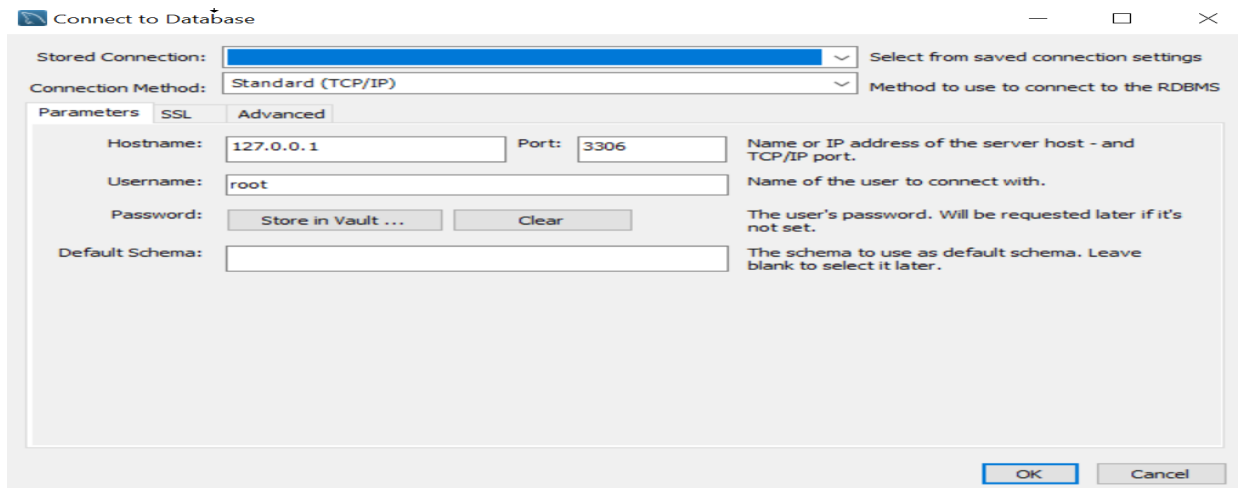
La instalación del servidor nos ha generado en nuestro disco C: un directorio llamado **C:\xampp** con todos los documentos y directorios necesarios para empezar a trabajar.

El proyecto se va a guardar, concretamente, en el directorio **C:\xampp\htdocs\TrabajoFinGrado**.

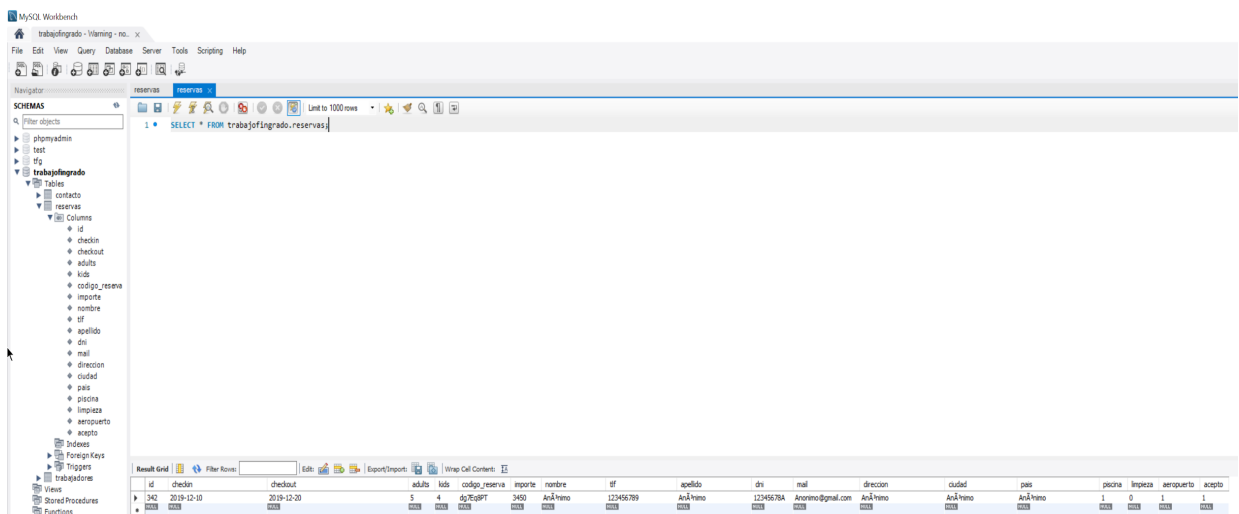
## 5.2.2 INSTALACIÓN WORKBENCH

Aunque contamos con el gestor de bases de datos “*phpmyadmin*”, nos hemos decantado por otro denominado “*workbench*” debido a su facilidad de uso, además de tratarse de una herramienta mucho más visual y amigable.

El software se descargó en “<https://www.mysql.com/products/workbench/>” y se procedió a la sincronización con mi base de datos desde el menú de conexión. La base de datos usada se llama **trabajofingrado**.



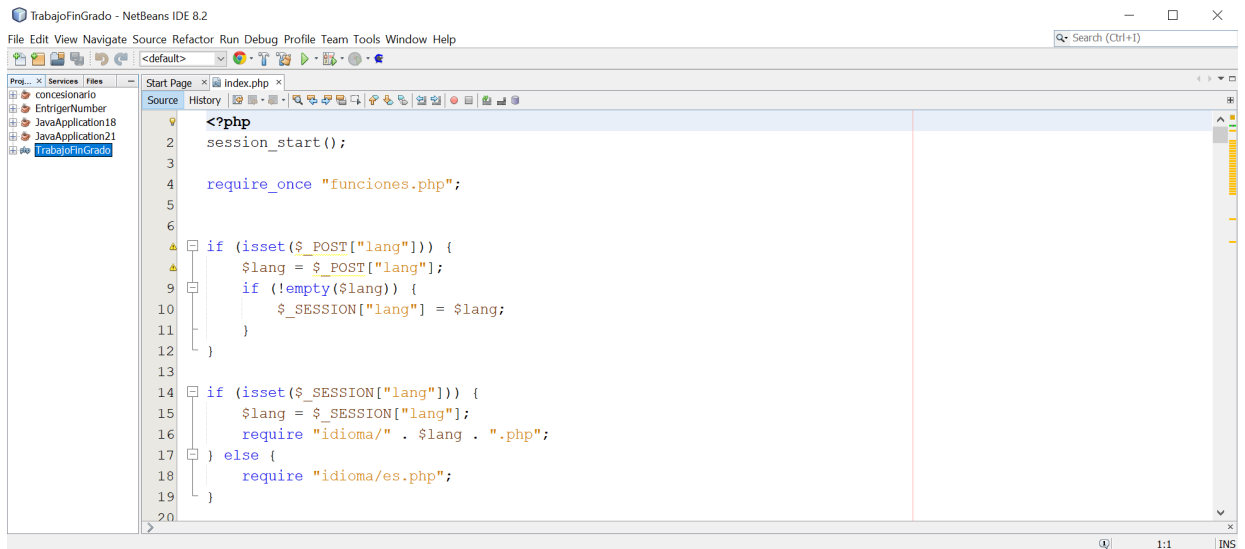
Una vez accedido con los credenciales correctos, disponemos de un menú desde donde poder hacer las consultas necesarias.



## 5.2.3 INSTALACIÓN NETBEANS

El entorno de desarrollo elegido ha sido **Netbeans** debido a su compatibilidad con multitud de lenguajes, así como su fácil manejo y conocimiento previo de la multitud de atajos de teclado existentes.

La versión empleada ha sido la 8.2 y su descarga fue desde la web oficial "<https://netbeans.org>"



Una vez instalados los programas con los que voy a trabajar, la estructura de mi proyecto se encuentra dentro de **C:\xampp\htdocs\TrabajoFinGrado** siendo "**TrabajoFinGrado**" la carpeta que contiene el proyecto, así como sus distintos elementos dentro del servidor **XAMPP**.

## 5.3 LENGUAJES/LIBRERÍAS/Frameworks EMPLEADOS

Para el desarrollo de este proyecto se han empleado multitud de lenguajes, así como librerías.

Pasaremos a hablar brevemente de cada uno de ellos incluyendo imágenes de código.

## 5.3.1 HTML

HTML (HyperText Markup Language) Se trata de un lenguaje de marcado para el desarrollo de páginas web. La versión actual es la 5, que es la que se ha empleado para este proyecto.

Ha sido combinado junto con CSS (Cascading Style Sheets) y el framework bootstrap de los cuales se hablará más tarde.

```
<div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner" role="listbox">
    <div class="carousel-item active" style="background-image:url('img/pa22.jpg')"></div>
    <div class="carousel-item" style="background-image:url('img/terrazza.jpg')"></div>
    <div class="carousel-item" style="background-image:url('img/jardin.jpg')"></div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

## 5.3.2 CSS

CSS (Cascading Style Sheets), es el lenguaje empleado para aplicar estilos al proyecto. La versión actual es CSS3 que es la empleada en el proyecto.

Ha sido combinado junto a HTML y Bootstrap.

```
{
  margin: 0;
  padding: 0;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}

body {
  background: #f2f2f2;
  font-family: Arial;
}

.texto{
  background: #f2f2f2;
}

.container {
  background: #fff;
  width: 95%;
  max-width: 1000px;
  margin: 20px auto;
}
```

### 5.3.3 Bootstrap/ TempusDominus

Bootstrap se trata de una librería multiplataforma de código abierto el cual ha sido empleado para facilitar las tareas de diseño ya que cuenta con multitud de herramientas de fácil acceso y con un acabado ideal para diseños web.

No fue necesaria su instalación, ya que se hizo uso de la llamada a las hojas de estilo dentro de la cabecera head en html.

```
<link rel="stylesheet" href="css/bootstrap.min.css" crossorigin="anonymous">
<link rel="stylesheet" href="css/bootstrap-theme.min.css" crossorigin="anonymous">
<script src="js/bootstrap.min.js" crossorigin="anonymous"></script>
```

Se ha hecho uso del *plugin* de Bootstrap “**TempusDominus**” el cual proporciona el diseño del calendario.

```
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></scrip
t>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.22.2/moment.min.js"></s
cript>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/tempusdominus-bootstrap-
4/5.0.1/js/tempusdominus-bootstrap-4.min.js"></script>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/tempusdominus-bootstrap-
4/5.0.1/css/tempusdominus-bootstrap-4.min.css" />
```

### 5.3.4 PHP

PHP(Hypertext Preprocessor) es un lenguaje de programación aparecido en 1995.

La particularidad de PHP es que se procesa en servidores y permite ser incrustado en HTML.

Durante el desarrollo de este proceso ha sido el principal lenguaje usado para diversas tareas: el traspaso de información de una página a otra mediante formularios, conexiones con el servidor MySQL, funciones etc...

La versión con la que se ha trabajado ha sido la 7.3.9.



```

<?php
session_start();

require_once "funciones.php";

if (isset($_POST["lang"])) {
    $lang = $_POST["lang"];
    if (!empty($lang)) {
        $_SESSION["lang"] = $lang;
    }
}

if (isset($_SESSION["lang"])) {
    $lang = $_SESSION["lang"];
    require "idioma/" . $lang . ".php";
} else {
    require "idioma/es.php";
}

$global_errors = isset($_GET['global_errors']);
$error_adults = isset($_GET['error_adults']);
$error_exceed_people = isset($_GET['exceedPeople']);
$error_month = isset($_GET['errorMonth']);
$error_day = isset($_GET['errorDay']);
$error_year = isset($_GET['errorYear']);
$fechaAnterior = isset($_GET['fechaAnterior']);
$pocosDias = isset($_GET['dia']);
$diasOcupados = isset($_GET['diasOcupados']);
$noAceptado = isset($_GET['noAceptado']);

```

### 5.3.5 JavaScript

JavaScript es un lenguaje de programación interpretado, definido como orientado a objetos.

Se ejecuta en el lado del cliente, permitiendo el desarrollo de páginas dinámicas, así como el uso en multitud de tareas como la validación de formularios entre otras.

Ha sido usado junto a la biblioteca conocida como jQuery versión 3.4.1, la cual permite, de forma más sencilla, la manipulación de documentos HTML.

El principal uso que se le dio a jQuery, fue la manipulación de los calendarios “*datetimepicker*”, permitiendo marcar los días reservados entre otras muchas opciones.

```

<script type="text/javascript">
    $(function () {
        $('#datetimepicker1').datetimepicker({
            locale: 'es',
            format: 'DD-MM-YYYY',
            disabledDates: [<?php echo $resultados ?>],
            minDate: new Date()
        });

        $('#datetimepicker2').datetimepicker({
            locale: 'es',
            format: 'DD-MM-YYYY',
            disabledDates: [<?php echo $resultados ?>],
            minDate: new Date()
        });

        $('#datetimepicker13').datetimepicker({
            locale: 'es',
            format: 'L',
            inline: true,
            sideBySide: true,
            disabledDates: [<?php echo $resultados ?>]
        });
    });
</script>

```

### 5.3.6 LIBRERÍA TCPDF

TCPDF es una librería de código abierto de PHP que permite la elaboración de documentos .pdf mediante datos extraídos de la Base de Datos.

Es una librería de uso complejo, ya que no ofrece una interfaz amigable, ni una organización muy clara de sus ficheros. Se puede obtener desde el enlace ["www.tcpdf.org"](http://www.tcpdf.org)

Su utilización ha sido fundamental para la generación de las facturas con los datos extraídos de la base de datos (TrabajoFinGrado).

```
// set document information
$pdf->SetCreator(PDF_CREATOR);
$pdf->SetTitle('Factura');
$pdf->SetSubject('TCPDF Tutorial');
$pdf->SetKeywords('TCPDF, PDF, example, test, guide');

// set default header data
$pdf->SetHeaderData(PDF_HEADER_LOGO, PDF_HEADER_LOGO_WIDTH, PDF_HEADER_TITLE . ' ',
//$pdf->setFooterData(array(0,64,0), array(0,64,128));
// set header and footer fonts
$pdf->setHeaderFont(Array(PDF_FONT_NAME_MAIN, '', PDF_FONT_SIZE_MAIN));
//$pdf->setFooterFont(Array(PDF_FONT_NAME_DATA, '', PDF_FONT_SIZE_DATA));
// set default monospaced font
$pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);

// set margins
$pdf->SetMargins(PDF_MARGIN_LEFT, PDF_MARGIN_TOP, PDF_MARGIN_RIGHT);
$pdf->SetHeaderMargin(PDF_MARGIN_HEADER);
//$pdf->SetFooterMargin(PDF_MARGIN_FOOTER);
// set auto page breaks
$pdf->SetAutoPageBreak(TRUE, PDF_MARGIN_BOTTOM);

// set image scale factor
$pdf->setImageScale(PDF_IMAGE_SCALE_RATIO);
```

### 5.3.7 FRAMEWORK FONT AWESOME

*Font awesome* es un *framework* que ha sido empleado en este proyecto para proporcionar un aspecto más profesional y vistoso.

Ha sido utilizado en multitud de apartados del proyecto como, por ejemplo, el diseño de los botones de acceso a las redes sociales *Facebook* e *Instagram*.



Una vez descargado el fichero “font-awesome-all.css” y añadido al directorio “C:\xampp\htdocs\TrabajoFinGrado\css” simplemente consiste en insertarlo en el fichero que sea necesario mediante un link en HTML.

```
<!-- Font Awesome Icons -->
<link rel="stylesheet" media="all" href="css/font-awesome-all.css">
```

Una vez cargado el link, buscamos el icono que se desea incrustar y se añade mediante una clase en el código HTML.

```
class="fab fa-facebook-f fa-3x"
```

## 5.4 BASE DE DATOS

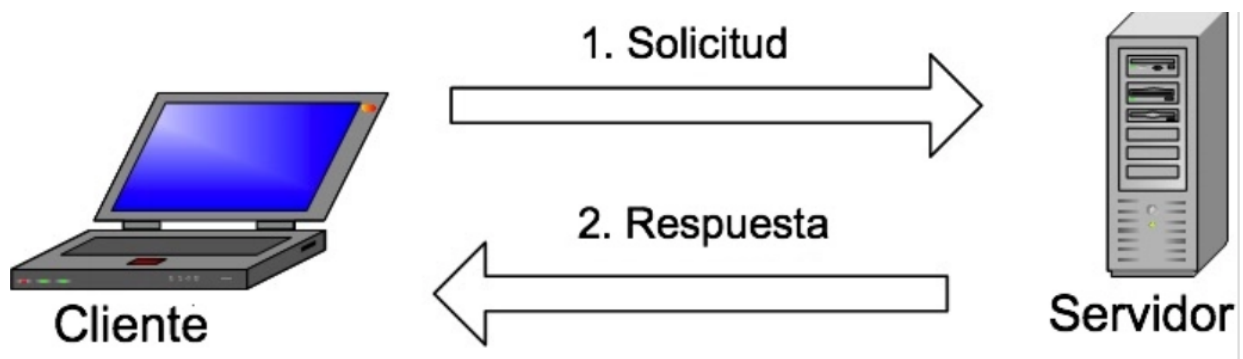
### 5.4.1 INTRODUCCIÓN

Un elemento fundamental para el desarrollo del proyecto es disponer de una base de datos.

En ella se va a almacenar la información referente a las reservas, puntuaciones etc.

Se ha hecho uso de MySQL para la creación de la base de datos.

MySQL es un sistema de gestión de bases de datos relacionales. Hace uso del sistema cliente-servidor. En este sistema, encontramos 2 grandes actores: cliente y servidor. El **cliente** es el equipo donde se encuentra instalado y el **servidor**, donde vamos a almacenar los datos.



Para la comunicación entre ambas entidades, se hace uso del lenguaje SQL (Structured query language).

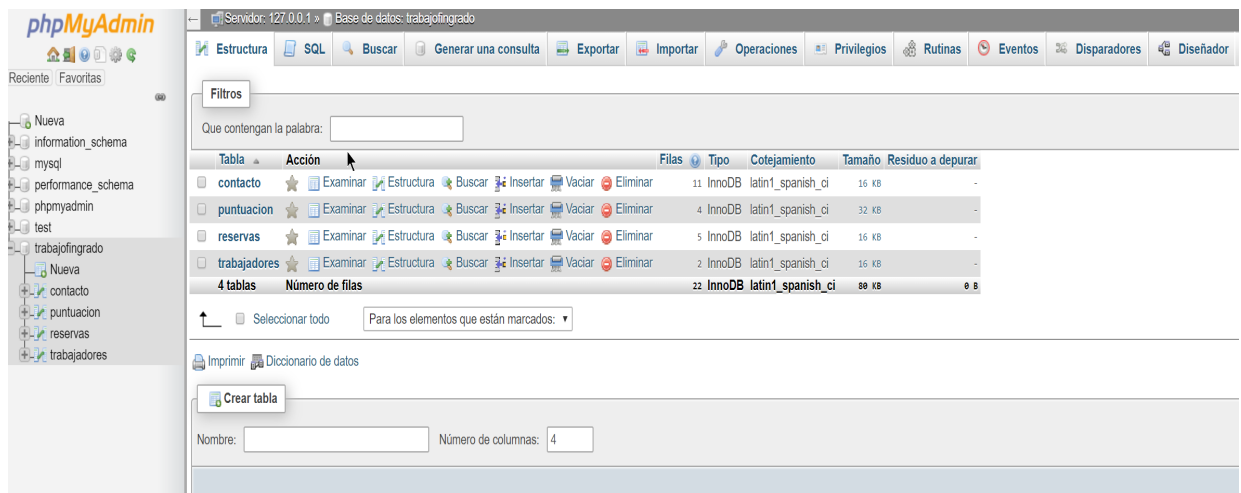
SQL es el lenguaje que le da las órdenes al servidor de lo que se quiere hacer con la información de la que dispone.

En este proyecto se ha hecho uso de las sentencias (*SELECT*, *INSERT*, *DELETE*, *JOIN* y *UPDATE*).

## 5.4.2 CREACIÓN DE LA BASE DE DATOS

Para el desarrollo de la base de datos he empleado la herramienta que provee XAMPP conocida como *phpmyadmin*, la cual me permite la gestión total de todos los elementos.

Para acceder al menú de gestión de *phpmyadmin* se hace mediante la ruta “localhost/phpmyadmin” donde accederemos a la página inicial.



El primer paso fue la creación del usuario “*root*” que va a realizar las tareas de administración, creación, gestión de la base de datos.

### Vista global de las cuentas de usuario

	Nombre de usuario	Nombre del servidor	Contraseña	Privilegios globales	Conceder	Acción
<input type="checkbox"/>	cualquiera	%	No	USAGE	No	Exportar
<input type="checkbox"/>	pma	localhost	No	USAGE	No	Exportar
<input type="checkbox"/>	root	127.0.0.1	No	ALL PRIVILEGES	Sí	Exportar
<input type="checkbox"/>	root	:::1	No	ALL PRIVILEGES	Sí	Exportar
<input type="checkbox"/>	root	ec2amaz-1qpqh3j	No	ALL PRIVILEGES	Sí	Exportar
<input type="checkbox"/>	root	localhost	No	ALL PRIVILEGES	Sí	Exportar

Una vez creado el **usuario** procedemos a la elaboración de la base de datos, la cual ha sido llamada “**trabajofingrado**”. En ella se encuentran las diversas tablas empleadas para almacenar la información necesaria.

Aunque la creación de la base de datos se ha realizado desde *phpmyadmin*, el manejo y consulta de datos se ha realizado desde un software externo denominado **Workbench**, ya que proporciona un uso más sencillo.

### 5.4.3 TABLAS

La elaboración del proyecto no ha precisado de una gran cantidad de tablas, ya que se hace uso única y exclusivamente de 2 tablas (“Reservas” y “Puntuación”), aunque existe otra tabla creada con idea de futuro para almacenar otro tipo de información (“Trabajadores”). Esta última tabla mencionada, no cuenta con ningún tipo de relación con las dos anteriores.

**Tabla Reservas:** Es la que almacena la información referente a las reservas generadas por los clientes al cumplimentar los formularios de los que dispone para dicho propósito. Cuenta con 20 campos, en los que almacenamos información muy útil para multitud de labores, entre ellas la generación de facturas mediante la librería TCPDF.

Además, es la tabla desde la que vamos a extraer la información para eliminar reservas, modificar reservas y poder realizar la trazabilidad de la aceptación por parte del cliente de los términos y condiciones de uso. Así como la comprobación de la puntuación por parte del cliente, es decir, si un cliente ha puntuado la vivienda en el apartado correspondiente de la web, quedará registrado en la base de datos y esa información se consultará para que no vuelva a realizar una valoración (Solamente se permite una valoración por reserva).

Su estructura es la siguiente:

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
id	int(11), primary Key, autoincremental	Identificador de las reservas.
check-in	date	fecha de inicio reserva
check-out	date	fecha final reserva
adults	int(11)	número adultos
kids	int(11)	número niños
codigo_reserva	varchar(60)	código generado para la reserva
importe	double	importe de la reserva
nombre	text	nombre del cliente
apellido	text	apellido del cliente
tlf	text	telefono del cliente
dni	varchar(9)	dni/pasaporte cliente
mail	text	mail del cliente
direccion	text	dirección del cliente
ciudad	text	ciudad del cliente
pais	text	país del cliente
piscina	tinyint(1)	SI el cliente desea piscina climatizada se marca a 1

<b>limpieza</b>	tinyint(1)	SI el cliente desea servicio limpieza se marca a 1
<b>aeropuerto</b>	tinyint(1)	SI el cliente desea recogida en aeropuerto se marca a 1
<b>acepto</b>	tinyint(1)	Queda a 1 si el cliente ha marcado que acepta las condiciones de uso. Desde este campo podemos realizar trazabilidad para demostrar que fueron aceptadas. nunca puede tomar otro valor.
<b>puntuado</b>	tinyint(1)	control reservas realizadas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
1	<b>id</b> 🗝️	int(11)			No	Ninguna		AUTO_INCREMENT
2	<b>checkin</b>	date			No	Ninguna		
3	<b>checkout</b>	date			No	Ninguna		
4	<b>adults</b>	int(11)			Sí	NULL		
5	<b>kids</b>	int(11)			Sí	NULL		
6	<b>codigo_reserva</b>	varchar(60)	latin1_spanish_ci		Sí	NULL		
7	<b>importe</b>	double			Sí	NULL		
8	<b>nombre</b>	text	latin1_spanish_ci		Sí	NULL		
9	<b>tlf</b>	text	latin1_spanish_ci		Sí	NULL		
10	<b>apellido</b>	text	latin1_spanish_ci		Sí	NULL		
11	<b>dni</b>	varchar(9)	latin1_spanish_ci		Sí	NULL		
12	<b>mail</b>	text	latin1_spanish_ci		Sí	NULL		
13	<b>direccion</b>	text	latin1_spanish_ci		No	Ninguna		
14	<b>ciudad</b>	text	latin1_spanish_ci		No	Ninguna		
15	<b>pais</b>	text	latin1_spanish_ci		No	Ninguna		
16	<b>piscina</b>	tinyint(1)			No	Ninguna		
17	<b>limpieza</b>	tinyint(1)			No	Ninguna		
18	<b>aeropuerto</b>	tinyint(1)			No	Ninguna		
19	<b>acepto</b>	tinyint(1)			No	Ninguna		
20	<b>puntuado</b>	tinyint(1)			No	Ninguna		



Su *script* de creación es el siguiente:

```
CREATE TABLE `reservas` ( `id` int(11) NOT NULL AUTO_INCREMENT, `check-in` date NOT NULL, `check-out` date NOT NULL, `adults` int(11) DEFAULT NULL, `kids` int(11) DEFAULT NULL, `codigo_reserva` varchar(60) COLLATE latin1_spanish_ci DEFAULT NULL, `importe` double DEFAULT NULL, `nombre` text COLLATE latin1_spanish_ci DEFAULT NULL, `tlf` text COLLATE latin1_spanish_ci DEFAULT NULL, `apellido` text COLLATE latin1_spanish_ci DEFAULT NULL, `dni` varchar(9) COLLATE latin1_spanish_ci DEFAULT NULL, `mail` text COLLATE latin1_spanish_ci DEFAULT NULL, `direccion` text COLLATE latin1_spanish_ci NOT NULL, `ciudad` text COLLATE latin1_spanish_ci NOT NULL, `pais` text COLLATE latin1_spanish_ci NOT NULL, `piscina` tinyint(1) NOT NULL, `limpieza` tinyint(1) NOT NULL, `aeropuerto` tinyint(1) NOT NULL, `acepto` tinyint(1) NOT NULL, `puntuado` tinyint(1) NOT NULL, PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=356 DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
```

**Tabla Puntuación:** Los clientes, cuando se encuentren dentro de las fechas de su reserva y posterior a su reserva, podrán realizar valoración de distintos apartados mediante un formulario al que podrán acceder mediante su código de reserva.

Solamente podrán realizar una valoración, ya que mediante PHP se ha dotado al código de mecanismos para asegurarlo. Se trata de una tabla que consta de 9 campos y está relacionada con la tabla reservas mediante su “*foreign key*” **reserva\_id**, la cual depende de la clave primaria “**id**” de la tabla reservas.

Su estructura es la siguiente:

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
<b>id</b>	int(11), primary Key, autoincremental	identificador de las opiniones
<b>anfitrión</b>	int(11)	opinión sobre el anfitrión.
<b>limpieza</b>	int(11)	opinión sobre la limpieza

<b>relacion</b>	int(11)	opinión sobre la relación calidad/precio
<b>confort</b>	int(11)	opinión sobre el confort de la vivienda
<b>instalaciones</b>	int(11)	opinión sobre las instalaciones de la vivienda
<b>descubrir</b>	text	Como descubrieron "Villa Sonneland"
<b>opinion</b>	longtext	opinión propia del liente
<b>reserva_id</b>	int(11), foreign key	relación con tabla reservas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id 	int(11)			No	Ninguna		AUTO_INCREMENT	 Cambiar  Eliminar ▼ Más
2	anfitrión	int(11)			No	Ninguna			 Cambiar  Eliminar ▼ Más
3	limpieza	int(11)			No	Ninguna			 Cambiar  Eliminar ▼ Más
4	relacion	int(11)			No	Ninguna			 Cambiar  Eliminar ▼ Más
5	confort	int(11)			No	Ninguna			 Cambiar  Eliminar ▼ Más
6	instalaciones	int(11)			No	Ninguna			 Cambiar  Eliminar ▼ Más
7	descubrir	text	latin1_spanish_ci		No	Ninguna			 Cambiar  Eliminar ▼ Más
8	opinion	longtext	latin1_spanish_ci		No	Ninguna			 Cambiar  Eliminar ▼ Más
9	reserva_id 	int(11)			No	Ninguna			 Cambiar  Eliminar ▼ Más

Su *script* de creación es el siguiente:

```
CREATE TABLE `puntuacion` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `anfitrión` int(11) NOT NULL,
  `limpieza` int(11) NOT NULL,
  `relacion` int(11) NOT NULL,
  `confort` int(11) NOT NULL,
  `instalaciones` int(11) NOT NULL,
```

```

`descubrir` text COLLATE latin1_spanish_ci NOT NULL,
`opinion` longtext COLLATE latin1_spanish_ci NOT NULL,
`reserva_id` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `RESERVA_FK` (`reserva_id`),
CONSTRAINT `RESERVA_FK` FOREIGN KEY (`reserva_id`) REFERENCES `reservas` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci

```

**Tabla Trabajadores:** Al momento de elaborar dicha memoria, no se está haciendo uso de esta tabla, pero fue petición expresa del cliente su creación para poder ser usada en un futuro.

En ella se almacenan datos de los distintos trabajadores con los que cuenta la Villa, los cuales son los relacionados con la limpieza y jardinería.

COLUMNA	TIPO DE DATO	DESCRIPCIÓN
<b>id</b>	int(11), primary Key, autoincremental	identificador de los trabajadores
<b>nombre</b>	text	nombre del trabajador
<b>telefono</b>	int(9)	telefono del trabajador
<b>mail</b>	text	mail del trabajador
<b>departamento</b>	int(1)	departamento del trabajador. 1 limpieza, 2 anfitrión, 3 jardinería, 4 informática

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	<b>id</b> 🔑	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar  Eliminar ▼ Más
2	<b>nombre</b>	text	latin1_spanish_ci		No	Ninguna			Cambiar  Eliminar ▼ Más
3	<b>telefono</b>	int(9)			No	Ninguna			Cambiar  Eliminar ▼ Más
4	<b>mail</b>	text	latin1_spanish_ci		No	Ninguna			Cambiar  Eliminar ▼ Más
5	<b>departamento</b>	int(1)			No	Ninguna			Cambiar  Eliminar ▼ Más

Su *script* de creación es el siguiente:

```
CREATE TABLE `trabajadores` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` text COLLATE latin1_spanish_ci NOT NULL,  
  `telefono` int(9) NOT NULL,  
  `mail` text COLLATE latin1_spanish_ci NOT NULL,  
  `departamento` int(1) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1  
COLLATE=latin1_spanish_ci
```

### 5.4.4 DIAGRAMA ENTIDAD-RELACIÓN

Mediante el diagrama Entidad-Relación podemos visualizar de forma clara la estructura de la base de datos (campos, tipo, nombre) así como la relación existente entre diferentes tablas.

En este caso contamos con 3 tablas: reservas, puntuación y trabajadores. Entre ellas solamente existe relación entre “reservas” y “puntuación” ya que la tabla “trabajadores” esta de momento desligada del proyecto ya que no se hace uso de ella.

Observamos la relación entre el campo **id** de la tabla “reservas” y el campo **reserva\_id** de la tabla “puntuación”.

trabajofingrado reservas
id : int(11)
checkin : date
checkout : date
# adults : int(11)
# kids : int(11)
# codigo_reserva : varchar(60)
# importe : double
# nombre : text
# tlf : text
# apellido : text
# dni : varchar(9)
# mail : text
# direccion : text
# ciudad : text
# pais : text
# piscina : tinyint(1)
# limpieza : tinyint(1)
# aeropuerto : tinyint(1)
# acepto : tinyint(1)

trabajofingrado puntuacion
id : int(11)
# anfitrión : int(11)
# limpieza : int(11)
# relacion : int(11)
# confort : int(11)
# instalaciones : int(11)
# descubrir : text
# opinion : longtext
# reserva_id : int(11)

trabajofingrado trabajadores
id : int(11)
# nombre : text
# telefono : int(9)
# mail : text
# departamento : int(1)

## 5.4.5 CONEXIÓN

Con las diferentes tablas definidas, procedemos a establecer la conexión con el proyecto.

Se han realizado multitud de conexiones en el proyecto. Todas ellas mediante el sistema **PDO\_MySQL**. Mediante este sistema, podemos definir una interfaz con la que establecer una conexión entre PHP y la base de datos y sus diferentes tablas.

Para establecer conexión, se crea un objeto de tipo “**PDO**” al que se le pasan diferentes parámetros, como por ejemplo el *host*, que en este caso es la IP local 127.0.0.1, ya que el proyecto se está ejecutando en local. A continuación, se le pasa el nombre de la base de datos y por último, el usuario “**root**” y la contraseña en el caso de que exista.

Posteriormente ejecutamos la integridad referencial. Cuando tenemos esta propiedad en estado “**on**” estamos garantizando que las relaciones existentes entre 2 tablas siempre estén en sincronía durante operaciones de eliminación y actualización de datos.

Preparamos la consulta para su ejecución (SELECT, INSERT, UPDATE, JOIN, DELETE...) y la ejecutamos pasando los datos ocultos mediante el *array*.

Por último, le decimos que queremos los datos como un *array* asociativo mediante **setFetchMode(PDO::FETCH\_NAMED)** y finalmente especificamos que queremos todos los datos a la vez con **fetchAll()**.

Vamos a mostrar 2 ejemplos de una conexión para seleccionar datos y eliminar datos.

```
<?php
$user="root";
$db = new PDO('mysql:host=127.0.0.1;dbname=trabajofingrado', $user);
$db->exec('PRAGMA foreign_keys = ON;');
$codigo="736MK-zO@V";
$res=$db->prepare("SELECT id,checkin,checkout,adults,kids,codigo_reserva,importe FROM reservas WHERE codigo_reserva=?");
$res->execute(array($codigo));
$res->setFetchMode(PDO::FETCH_NAMED);
$datos = $res->fetchAll();
```

```
$user = "root";
$db = new PDO('mysql:host=127.0.0.1;dbname=trabajofingrado', $user);
$db->exec('PRAGMA foreign_keys = ON;');
$res = $db->prepare("DELETE FROM reservas WHERE id=?");
$res->execute(array($id));
$res->setFetchMode(PDO::FETCH_NAMED);
$datos = $res->fetchAll();
```

Se ha realizado conexiones para multitud de tareas como, por ejemplo, sacar información de la base de datos para generar facturas combinando este método con la librería **TCPDF**, de la que hablamos anteriormente en la página 20. También hemos modificado datos mediante el formulario de “modificar reserva”.

El uso ha sido tan variado que exponer todos los casos no sería productivo.

## 5.5 ELABORACIÓN DEL PROYECTO "Villa Sonneland"

El proyecto consiste en dotar a una Vivienda Vacacional de una aplicación web para la gestión de las reservas.

El cliente que accede a la web podrá, mediante un formulario, seleccionar los días que desea reservar, el número de personas, así como diversas opciones extra.

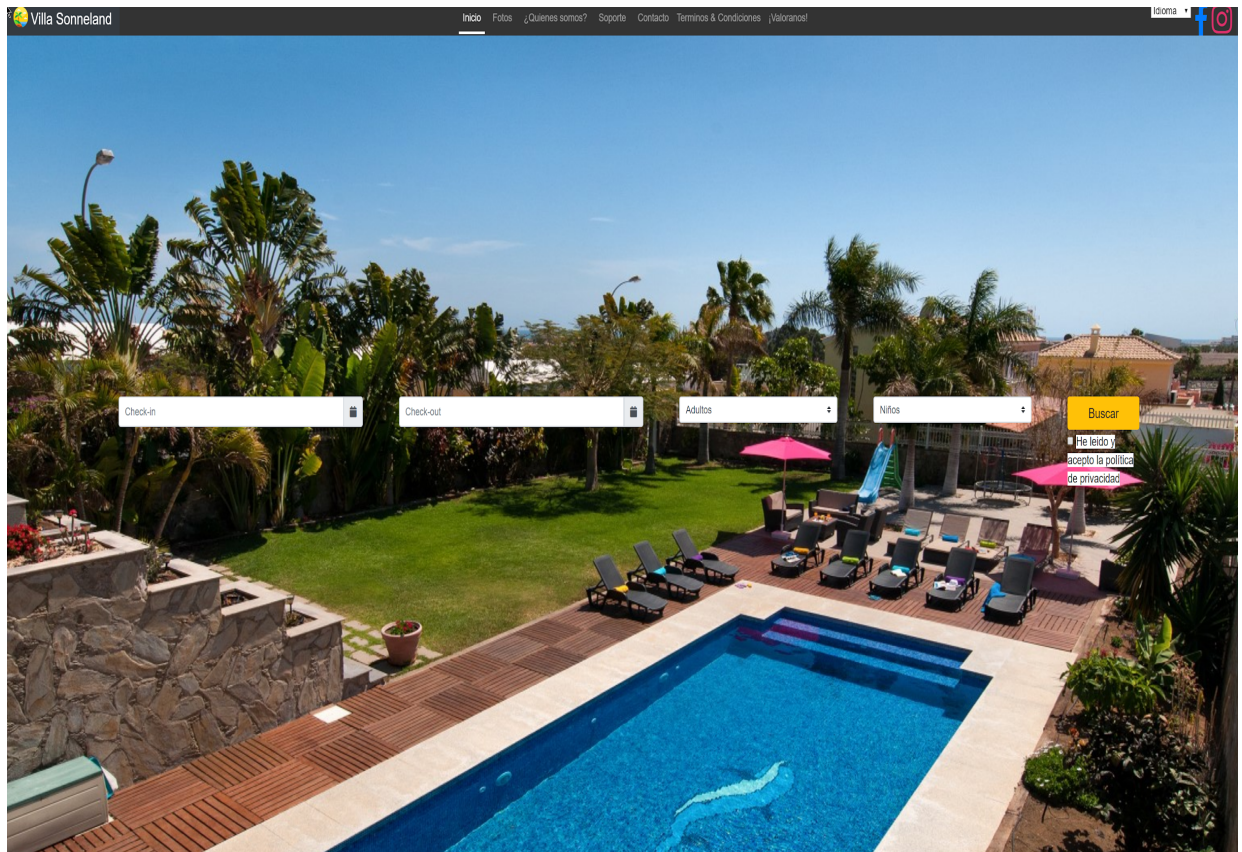
Dicha aplicación no va a precisar de ningún *login*, pero existen determinados puntos a los que no se podrá acceder sin cumplir una verificación de doble paso, como son los casos de “modificar reserva”, “eliminar reserva” y “puntuar”.

Como información extra, es importante mencionar que la integridad del proyecto se ha realizado siguiendo el modelo **responsive** para adaptar el diseño a diferentes resoluciones y tamaños de pantalla, siguiendo el modelo de *grid* propuesto por **bootstrap**.

### 5.5.1 ESTRUCTURA DE LA APLICACIÓN

La página inicial(*Landing page*) con la que nos encontramos es la relativa al fichero “index.php”.





Contamos con una barra de navegación realizada mediante **bootstrap** (*navbar*) en la que se nos da acceso a diferentes partes de la aplicación (Fotos, Soporte, Contacto, Valorar, ¿Quiénes somos?, Términos y condiciones), además de estas opciones, en la parte derecha, nos encontramos con los accesos a las redes sociales de la vivienda así como un selector de idiomas, ya que la página es multilinguaje (Español, Inglés, Alemán).

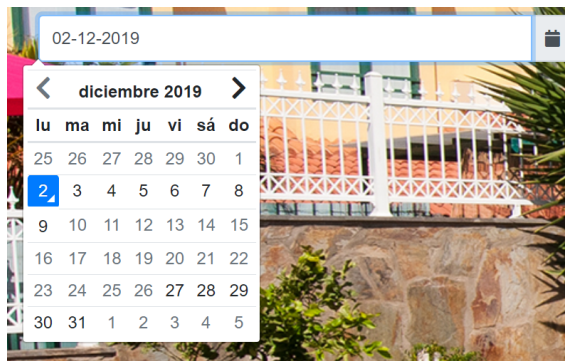
El fondo, no es una simple imagen. Se optó inicialmente por implementar mediante **bootstrap** un carrusel de fotos que fuesen cambiando cada cierto tiempo, pero finalmente esta opción fue descartada debido a la incompatibilidad que existía a la hora de incrustar el formulario de registro delante de un carrusel, por lo que se decidió explorar una vía alternativa, que consistió en almacenar varias imágenes en un *array* y mediante el uso del lenguaje **Javascript**, ir modificando dichas imágenes cada 5 segundos.

## 5.5.2 RESERVAS

Finalmente nos encontramos con el formulario de registro. Dicho formulario hace uso del *plugin* de **bootstrap** **TempusDominus** para el uso de



calendarios con los que poder realizar la selección de fechas. Se emplean 2 calendarios, uno para el *check-in* y otro para el *check-out*.



En dichos calendarios está contemplado el no poder seleccionar días, meses o años pasados. Además se ha desarrollado mediante PHP/Javascript los recursos necesarios para el que los días que ya estén reservados no puedan ser seleccionados.

```
<script type="text/javascript">
```

```
$(function () {  
    $('#datetimepicker1').datetimepicker({  
        locale: 'es',  
        format: 'DD-MM-YYYY',  
        disabledDates: [php echo $resultados ?&gt;],<br/        minDate: new Date()  
    });  
});
```

Luego tenemos la selección del número de clientes dividida entre adultos y niños. Por decisión de mi cliente, se considera niño al menor de 12 años. Entre el número de adultos y niños no puede existir más de 12 personas.

Todos estos casos de uso están contemplados mediante código PHP provocando un aviso de error a la hora de darle al "OK".

Tenemos un botón para aceptar Términos y Condiciones de Uso. Si dicho *checkbox* no es seleccionado, no se podrá continuar con la reserva y provocará otro aviso de error.

Una vez cumplidos los requisitos, pasamos a la página en la que tenemos que introducir nuestros datos, entre otras opciones.

Información	
Checkin:	24-01-2020
Checkout:	29-01-2020
Adultos:	4
Niños:	1
Nº días:	5
Precio:	1725€

Si desea confirmar la reserva, rellene los datos

Nombre:

Apellidos:

Dirección:

Ciudad:

País:

Email:

Repetir Email:

Teléfono:

DNI/Pasaporte:

Servicios adicionales

☐ ¿Piscina climatizada? 50€ día

☐ ¿Servicio de limpieza? 20€ día

☐ ¿Servicio de recogida? 25€

A petición del cliente, contamos con una tabla que nos proporciona una breve descripción de los datos de la reserva seleccionada previamente.

El precio establecido es de 345€ al día, siempre y cuando el número de adultos sea menor a 5. En caso contrario, el precio asciende a 450€. El precio se calcula mediante una simple función PHP.

```
function coste($inicio, $fin, $adultos) {
    if ($adultos <= 5) {
        $precio = 345;
    } else {
        $precio = 450;
    }
    $fecha1 = new DateTime($inicio);
    $fecha2 = new DateTime($fin);
    $resultado = $fecha1->diff($fecha2);
    $num_days = $resultado->format('%a');
    $coste = $num_days * $precio;
    return $coste;
}
```

El precio que se muestra es el de la reserva, sin contar el I.G.I.C., ni los servicios adicionales.

A continuación, encontramos el formulario con la información que se solicita al hacer la reserva. La parte inferior del formulario cuenta con distintas opciones de los servicios adicionales que se pueden contratar, mostrándose su precio.

El formulario cuenta con las verificaciones necesarias previamente mediante **Javascript** tales como que el teléfono sea numérico y tanto el mail como su copia sean iguales, entre otras opciones.

Una vez cumplimentados los datos necesarios, el sistema enviará a una página donde se muestra toda la información de nuestra reserva, además de el envío de un correo de confirmación de reserva al mail introducido en el formulario anterior.

Para el envío del correo electrónico se ha tenido que configurar un servidor de correo electrónico editando los ficheros del servidor Apache “C:\xampp\php\php.ini” y “C:\xampp\sendmail\sendmail.ini”.

A continuación, se muestra un extracto de dicha configuración:

```

SMTP=smtp.gmail.com
; http://php.net/smtp-port
smtp_port=587

; For Win32 only.
; http://php.net/sendmail-from
sendmail_from = villasonneland@gmail.com

; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
; http://php.net/sendmail-path
sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"

smtp_server=smtp.gmail.com

; smtp port (normally 25)

smtp_port=587

; SMTPS (SSL) support
; auto = use SSL for port 465, otherwise try to use TLS
; ssl  = always use SSL
; tls  = always use TLS
; none = never try to use SSL

smtp_ssl=auto

; the default domain for this server will be read from the registry
; this will be appended to email addresses when one isn't provided
; if you want to override the value in the registry, uncomment and modify
;default_domain=mydomain.com

; log smtp errors to error.log (defaults to same directory as sendmail.exe)
; uncomment to enable logging

error_logfile=error.log

; create debug log as debug.log (defaults to same directory as sendmail.exe)
; uncomment to enable debugging

;debug_logfile=debug.log

; if your smtp server requires authentication, modify the following two lines

auth_username=villasonneland@gmail.com
auth_password=

```

Además del envío del correo electrónico, se generará una factura en formato .pdf.

Su código de reserva es: SrzxBcy7	
Checkin	24-01-2020
Checkout	29-01-2020
Adultos	4
Niños	1
Precio	1725€
Nombre	Rut Barrero
Dirección	Tejina 8, Madrid, España
Piscina climatizada	250€
Servicio de limpieza	100€
Servicio de recogida	0€
Precio con servicios	2075€
I.G.I.G 6,5%	134.88€
Total	2209.88€
<a href="#">Generar PDF</a>	

Con cada reserva se genera un código de reserva único mediante una función **PHP**. Dicho código consiste en un conjunto alfanumérico de longitud de 8 caracteres seleccionados de forma aleatoria y garantizando la no repetición.

```
<?php
function generarKey() {
    $length = 8;
    $cadena = "1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
    $codigo = substr(str_shuffle($cadena), 0, $length);
    return $codigo;
}
```

La generación de la factura mediante la librería **TCPDF** ha sido una de las partes más complicadas del trabajo ya que se trata de una librería que nunca había usado y es muy poco intuitiva.

La factura generada obtiene la información de la base de datos, por lo que es única para cada cliente. Se ha tenido que hacer un diseño propio de factura ya que, hasta el momento, "Villa Sonneland" no contaba con un estilo de

factura propio, por lo que tras el visto bueno del cliente, el estilo seleccionado es el siguiente:



**Villa Sonneland**

Jesús Izquierdo García  
C/Vicente Blasco Ibañez 10  
Maspalomas  
50791638W  
Fecha: 03-12-2019  
Hora: 18:23

## Factura

**Cliente:** Rut Barrero  
**Domicilio:** Tejina 8, Madrid, España  
**C.I.F:** 50754564G

Checkin	Checkout	adultos	niños	codigo	coste
24-01-2020	29-01-2020	4	1	SrzxBcy7	1725€
Servicios adicionales					
Piscina climatizada. 50€ día					250€
Servicio Limpieza. 20€ día					100€
Recogida Aeropuerto. 25€					0€
i.g.i.c 6,5%					134.88€
Precio total					2209.88€

En la cabecera nos encontramos con el logo de la empresa, así como los datos del propietario y la hora y el día de emisión de la factura.

A continuación, tenemos los datos del cliente y la información de su reserva, así como el desglose de los precios de los distintos servicios contratados.

El formato del mail enviado al correo de confirmación de reserva es el siguiente:

Mensaje de contacto devillasonneland@gmail.com

villasonneland@gmail.com

para mí ▼

## Gracias por reservar en Villa Sonneland

Estos son los datos de su reserva

Reserva	Información
Checkin	11-02-2020
Checkout	17-02-2020
Código reserva	y5gpTVY8
Adultos	3
Niños	0
Precio	2070€
Nombre	jesus Izquierdo Puebla
Dirección	tejina
Piscina climatizada	300€
Servicio de limpieza	120€
Servicio de recogida	0€
Precio con servicios	2490€
I.G.I.C 6,5%	161.85€
Total	2651.85€

### 5.5.3 CONTROL DE ERRORES

Si se produce algún error a la hora de hacer reservas, se lanzará un mensaje de error con una descripción del mismo.

Los errores controlados son los relativos a fechas incorrectas, número máximo de personas y condiciones no aceptadas.

Disponemos de una variable de tipo *booleana* inicializada a *true*, así como una variable de errores que nos dirá qué tipo de error se ha producido.

```
$submit_status = true;
$url_errors = "refresh:0; url=index.php?global_errors=true";
```

En caso de que se produzca un error, el valor de la variable cambiará a “false” y saltará el mensaje de error.

```
/**
 * Validacion del nº de personas. Entre adultos (mínimo uno) y niños solo pueden ser doce.
 */
if ($adults + $kids > 12) {
    $submit_status = false;
    $url_errors = $url_errors .
        "&exceedPeople=true&kids=" . $kids . "&adults=" . $adults;
}
/**
 * Validacion mes fin menor inicio
 */
if ($mes_checkout < $mes_checkin) {
    $submit_status = false;
    $url_errors = $url_errors . "&errorMonth=true&mes_checkin="
        . $mes_checkin . "&mes_checkout=" . $mes_checkout;
}
```

Recuperamos nuevamente en “index.php” el error provocado de la siguiente forma:

```
$global_errors = isset($_GET['global_errors']);
$error_adults = isset($_GET['error_adults']);
$error_exceed_people = isset($_GET['exceedPeople']);
$error_month = isset($_GET['errorMonth']);
$error_day = isset($_GET['errorDay']);
$error_year = isset($_GET['errorYear']);
$fechaAnterior = isset($_GET['fechaAnterior']);
$pocosDias = isset($_GET['dia']);
$diasOcupados = isset($_GET['diasOcupados']);
$noAceptado = isset($_GET['noAceptado']);

if (isset($_GET['kids'])) {
    $no_kids = $_GET['kids'];
}
```

Y lanzamos nuestro error:

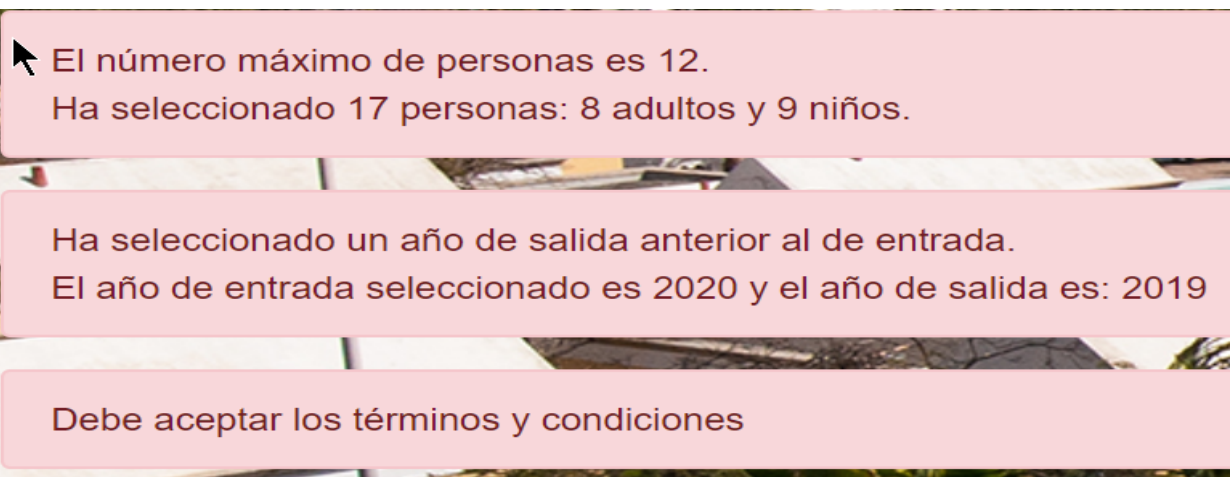


```

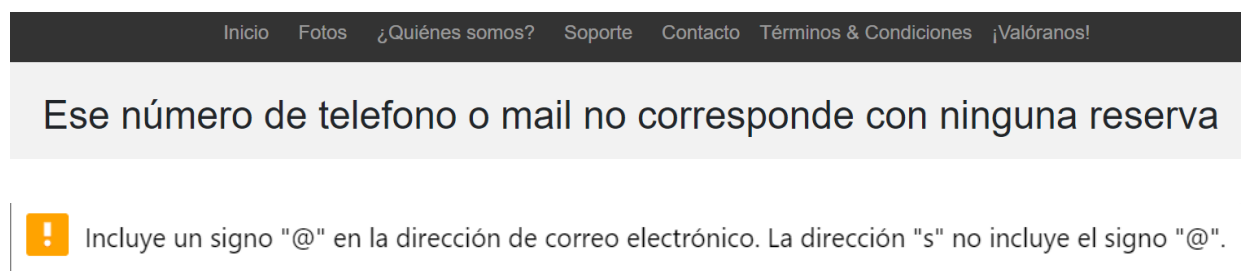
<div class="alert alert-danger" style="
if ($error_adults == false) {
    echo 'display:none';
}
?>">
    Debe seleccionar al menos un adulto
</div>

```

Como resultado, los errores lanzados se muestran de la siguiente forma:



El control de errores se extiende por diferentes apartados de la web, tales como el acceso no autorizado a páginas por no poseer un código de reserva o no introducir bien los datos de un formulario.



localhost dice

El código introducido no corresponde con el enviado a su correo.

Aceptar

localhost dice

Todos los campos son obligatorios

Aceptar

[Inicio](#) [Fotos](#) [¿Quiénes somos?](#) [Soporte](#) [Contacto](#) [Términos & Condiciones](#) [¡Valóranos!](#)

Este recurso no está disponible

En caso de que alguno de los mails enviados a la hora de generar una factura no pudiera ser enviado, se cuenta con un fichero de registro (*log*) llamado “error.log”, donde se detallan los posibles *mails* perdidos y errores producidos en el proceso.

```
19/11/02 12:10:44 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials x21sm12220180wmj.42 - gsmtpt<EOL>
19/11/02 12:12:20 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials u10sm12816863wmj.0 - gsmtpt<EOL>
19/11/02 12:15:17 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials l5sm8905438wmj.44 - gsmtpt<EOL>
19/11/02 12:16:37 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials j66sm6543227vma.19 - gsmtpt<EOL>
19/11/02 12:19:13 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials y128sm10615372wmg.10 - gsmtpt<EOL>
19/11/02 12:19:51 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials z189sm16540061wmc.25 - gsmtpt<EOL>
19/11/02 12:23:08 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials 205sm10176585wmb.3 - gsmtpt<EOL>
19/11/02 12:27:14 : Username and Password not accepted. Learn more at<EOL> https://support.google.com/mail/?p=BadCredentials x7sm21863351wrg.63 - gsmtpt<EOL>
19/11/04 12:33:49 : Message is missing sender's address
19/11/05 16:08:13 : The recipient address <dsds@> is not a valid RFC-5321 address. s17sm5685670wmh.41 - gsmtpt<EOL>
19/11/07 11:55:58 : The recipient address <12@> is not a valid RFC-5321 address. g11sm1759842wmh.27 - gsmtpt<EOL>
19/11/07 11:58:03 : The recipient address <1@> is not a valid RFC-5321 address. h7sm1621870wmb.37 - gsmtpt<EOL>
19/11/08 11:34:22 : The recipient address <de@> is not a valid RFC-5321 address. g69sm5331942wme.31 - gsmtpt<EOL>
19/11/28 15:02:35 : Message is missing recipient's address
19/11/28 15:03:37 : Message is missing recipient's address
19/11/30 13:05:09 : Message is missing recipient's address
19/12/11 10:58:03 : Message is missing recipient's address
```

## 5.5.4 MULTIIDIOMA

Al tratarse de una aplicación de uso turístico, desarrollarla en múltiples idiomas era una cuestión fundamental.

Para ello se han incorporado 3 idiomas:

- Español.
- Inglés.
- Alemán.

La forma de codificar la página ha consistido en incluir dentro de un **array dinámico** todo el texto del que se hace uso en la web, de forma que al llamar a la variable llamada *\$lang*, con la clave necesaria en el momento de su uso y estando seleccionado el idioma mediante el menú del **navbar**, se activará uno u otro.

Para que el idioma se mantenga en toda la página, se hace uso de una sesión iniciando siempre al inicio de cada apartado de la web.

```
session_start();
if (isset($_POST["lang"])) {
    $lang = $_POST["lang"];
    if (!empty($lang)) {
        $_SESSION["lang"] = $lang;
    }
}

if (isset($_SESSION["lang"])) {
    $lang = $_SESSION["lang"];
    require "idioma/" . $lang . ".php";
} else {
    require "idioma/es.php";
}
?>
```

```
<?php
```

```
$lang = array(  
    "inicio" => "Inicio",  
    "trabajadores" => "Trabajadores",  
    "fotos" => "Fotos",  
    "idioma" => "Idioma",  
    "quienes" => "¿Quiénes somos?",  
    "soporte" => "Soporte",
```

```
<?php
```

```
$lang = array(  
    "inicio" => "Start",  
    "trabajadores" => "Workers",  
    "fotos" => "Photos",  
    "idioma" => "Language",  
    "quienes" => "¿About us?",  
    "soporte" => "Support",
```

### 5.5.5 SOPORTE, PUNTUACIÓN Y VERIFICACIÓN DE DOBLE PASO

Tras la creación de una reserva por parte de un usuario, se presentan al mismo varias opciones.

Dichas opciones son:

- Editar la reserva.
- Eliminar la reserva.
- Recuperar el código de reserva.
- Puntuar la vivienda.

Tanto a la opción de “editar” como a la de “eliminar” se accede desde el mismo formulario y se precisa del código de reserva para ello. **Esta no es la única seguridad** implementada, ya que se ha hecho uso de una **verificación de doble paso** para ello.

¿Desea modificar o cancelar su reserva?

Su código de reserva es:

Modificar

Eliminar

¿Ha olvidado su código?

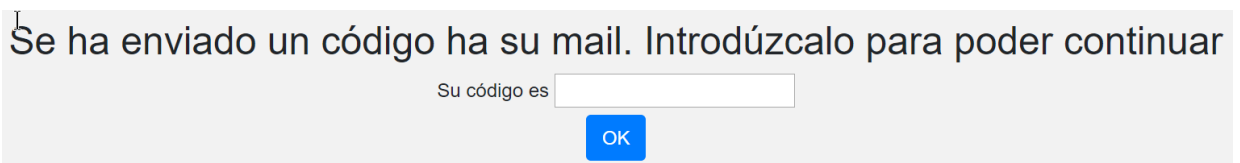
A la hora de realizar la verificación de doble paso, se estudió multitud de servicios, entre ellos, la API de google (Google Authenticator).

Para el propósito, esos servicios han sido descartados debido a que todos ellos llevaban un coste de contratación, excepto el de Google, pero éste requería que la persona tuviese instalado en su teléfono/dispositivo la aplicación “**Google Authenticator**” y eso no era viable, ya que no se puede exigir a un cliente que tenga el servicio.

Por lo que he procedido a hacer una “**simulación**” de una verificación de “doble paso” en la que una vez que el cliente desea editar, eliminar, o puntuar una reserva, se le envía a su correo un nuevo código que debe introducir para poder acceder al menú correspondiente.

**En todo momento se es consciente de que este sistema no es seguro del todo,** pero no es más que una simulación y en el momento en que el proyecto pase a ser usado en un entorno real se procederá a la contratación de uno de estos servicios y se aplicará la API correspondiente de forma sencilla.

Una vez introducido el código de reserva accedemos al siguiente paso, que consiste en introducir un nuevo código enviado al correo.

A screenshot of a web form with a light gray background. At the top, the text "Se ha enviado un código ha su mail. Introdúzcalo para poder continuar" is displayed in a dark gray font. Below this text, the label "Su código es" is positioned to the left of a white rectangular input field. Directly beneath the input field is a blue rectangular button with the white text "OK".

Se ha enviado un código ha su mail. Introdúzcalo para poder continuar

Su código es

OK

A continuación, ya podemos acceder a las diferentes funciones, tales como editar la reserva, eliminarla o puntuarla.

## ¿Desea modificar su reserva?

Checkin

24-01-2020

Checkout

29-01-2020

Nombre:

Rut

Adultos:

4

Niños:

1

Telefono:

659309039

Apellidos:

Barrero

DNI/Pasaporte:

50754564G

mail:

jesusizquierdopuebla@gmail.com

Direccion:

Tejina

Ciudad:

Madrid

País:

España

☒ ¿Piscina climatizada? 50€ día

☒ ¿Servicio de limpieza? 20€ día

☐ ¿Servicio de recogida? 25€

[Modificar](#)

Una vez terminada la edición de una reserva, se generará una nueva factura y se le enviará un correo con la nueva información.

La eliminación de la reserva, siempre que se haga al menos tres días antes de la fecha de inicio, no conllevará costes adicionales. En caso contrario, se cobrará un 60% del precio de la reserva según indicaciones del cliente. Todo esto está especificado en los **términos y condiciones**.

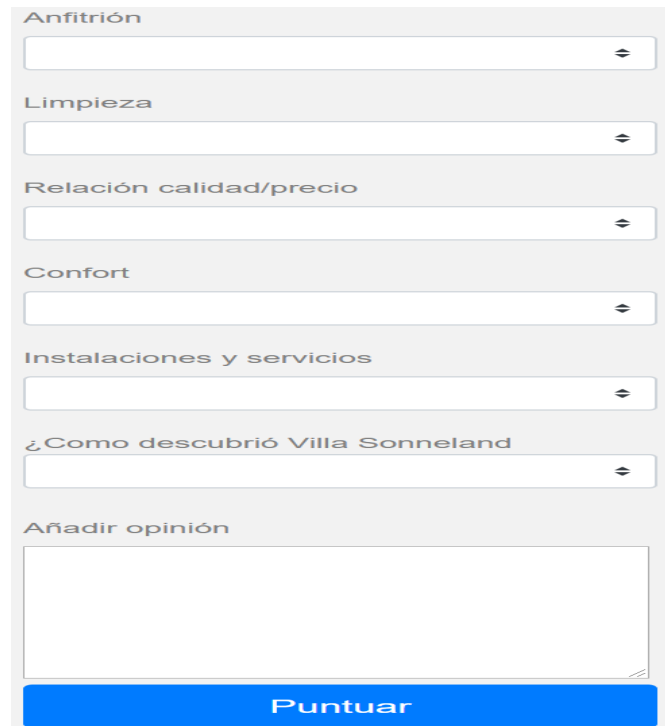
En cuanto a la puntuación de la vivienda, los pasos son similares. Se accede a ella mediante una verificación de doble paso y se ha controlado por código PHP que el cliente solamente pueda puntuar una vez la vivienda.

## Para poder valorarnos, debe disponer del código de reserva

Su código de reserva es:

[¡Valóranos!](#)

¿Ha olvidado su código?

El formulario de puntuación de Villa Sonneland está diseñado con un fondo gris claro y secciones blancas. Incluye los siguientes campos: un menú desplegable para 'Anfitrión', otro para 'Limpieza', uno para 'Relación calidad/precio', uno para 'Confort', uno para 'Instalaciones y servicios', y uno para '¿Como descubrió Villa Sonneland'. Debajo de estos campos hay un área de texto grande para 'Añadir opinión'. Al final del formulario hay un botón azul con el texto 'Puntuar' en blanco.

Los datos que se obtienen del formulario de puntuación quedan registrados en la tabla “**puntuación**” de nuestra base de datos, la cual está relacionada con la tabla “**reservas**” para saber en todo momento qué cliente hizo dicha puntuación.

```
SELECT reservas.nombre, reservas.apellido  
FROM reservas  
INNER JOIN puntuacion  
ON reservas.id=puntuacion.reserva_id;
```

Todos los formularios están protegidos de tal forma que si los códigos introducidos no son correctos, se mostrará un mensaje y se redirigirá a una página anterior.

Este recurso no está disponible

## 5.5.6 LOPDGDD Y EXTRAS

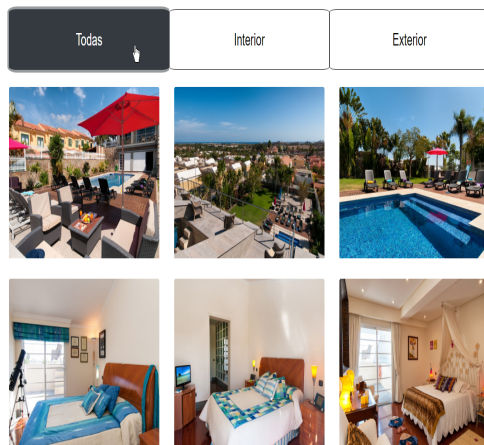
Los apartados descritos en los puntos anteriores, describen en gran medida el funcionamiento de la web, ya que en ellos se detalla como se crean reservas, cómo se editan, cómo se eliminan, la forma de realizar la puntuación, así como la explicación de distintas partes del código, fundamentales para su uso: TempusDominus, Generación de códigos, Tratamiento de errores, Generación de facturas con TCPDF, Código Multiidioma o Levantamiento del servidor de correo electrónico.

Los siguientes puntos a tratar, no intervienen en nada con los apartados anteriores, pero no por ello son menos importantes.

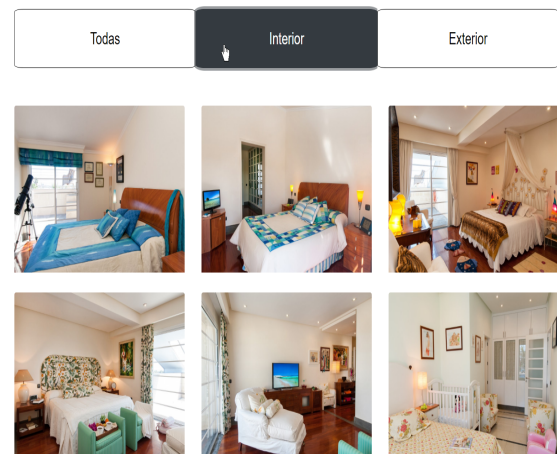
En la barra de navegación se han añadido distintos apartados como son los menús (Fotos, ¿Quiénes somos?, Contacto, Términos y condiciones). Fuera de la barra de navegación existe una opción fundamental la cual consiste en recuperar el código de la reserva en caso de pérdida.

En fotos se ha desarrollado una galería de fotos empleando el **Framework bootstrap** y se le ha aplicado un filtro con **Javascript** para filtrar las fotos por categorías: “Interior”, “Exterior”, “Todas”. Además se ha añadido la opción de un carrousel de fotos que al seleccionar una de ellas, se verá en grande y se desplegarán unos botones de paso a izquierda o derecha.

Galeria



Galeria



```
<script>
```

```
    baguetteBox.run('.cards-gallery', {animation: 'slideIn'});
```

```
</script>
```



El apartado “¿Quiénes somos?” no es más que un texto descriptivo de la vivienda. Para su diseño también se ha hecho uso del **framework Bootstrap** para poder adaptarlo de forma sencilla mediante el sistema *grid* a un formato *responsive*.

Este apartado era fundamental para el propietario, ya que en él se ven los distintos premios obtenidos en estos tres años de funcionamiento. Éstos son la obtención, por parte de Booking.com, del premio a la mejor vivienda vacacional en Gran Canaria en los años 2017 y 2018, así como el certificado 2019 de excelencia de TripAdvisor.

### ¿Quiénes somos?

Somos una empresa familiar enamorados de nuestra Villa.

Nos gustaría darle la bienvenida a nuestro proyecto que comienza en 2016 donde nuestros sueños se hicieron realidad.

Pudimos ofrecer la posibilidad de alquilar nuestra Villa sin perder el toque hogareño que nos caracteriza ya que cada rincón tiene un pedacito de nuestro corazón.

Decorado con objetos de nuestros viajes por el mundo, obras de arte y un sin fin de detalles, que hace a nuestros huéspedes sentirse como en casa en esta isla privilegiada por su clima, playas y actividades para disfrutar de las mejores vacaciones.

Descubre Villa Sonneland y podrás disfrutar de su espectacular piscina, jardín y habitaciones en familia o con amigos sin perder la intimidad por sus grandes espacios. Se sentirán como en un Hotel boutique para ustedes solos.

Para que su estancia sea lo más cómoda posible, disponemos de servicio de limpieza a diario y mantenimiento para que solamente se preocupen de disfrutar.

Al tratar directamente con nosotros, nos da la oportunidad de ofrecerle el mejor trato. Nuestro objetivo es atender a cada huésped de una manera muy personalizada.

El trato personalizado y el cariño con el que atendemos nuestra Villa nos ha hecho ganar los 2 últimos años el premio ofrecido por Booking.com a mejor Villa Vacacional en Gran Canaria. Además, hemos obtenido el certificado 2019 a la excelencia concedido por TripAdvisor.

¡Haremos todo lo posible para que sus vacaciones sean una experiencia especial que querrá vivir más de una vez!

Muchas gracias por su confianza y por pasar tiempo visitando nuestra web.



El siguiente punto es “Contacto”. Se trata de un formulario para que el cliente pueda hacer consultas al propietario de la vivienda.

En este punto se hizo uso nuevamente del servidor de correo electrónico, levantado previamente, para que una vez rellenado el formulario de contacto, el mensaje llegase a la cuenta de correo de la Villa.


Por otro lado, se añadió información como la incrustación de un mapa **Google Maps** con la posición de la Villa, además de un teléfono y dirección de contacto.

### Contaco

Rellene el formulario y le contestaremos lo antes posible.

Mensaje

Enviar




Calle Vicente Blasco Ibáñez, 10  
Calle Vicente Blasco Ibáñez, 10,  
35100 San Bartolomé de Tirajana,  
Las Palmas


Ampliar el mapa

Vista Bella  
Dalian Cueva  
Bungalows El Cardonal

Datos de mapas | Términos de uso | Notificar un problema de Maps



+34 928140006



villasonneland@gmail.com

Por último, se ha incluido un apartado donde se encuentran los **Términos y condiciones** de la Villa, así como la información referente a la "**Ley de Protección de Datos**".

Este apartado es el que se garantiza que ha sido leído una vez hecha la reserva mediante la inclusión en la base de datos de un apartado que solamente se queda registrado si se ha pulsado el *checkbox* de "**He leído y acepto los términos y condiciones**".

Se ha incluido un documento pdf en el apartado "**Términos y condiciones**", donde queda expuesto la "**Ley de Protección de Datos**" según la **LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de Derechos Personales)**.

## Términos y Condiciones

villasonneland.com requiere que todas las personas o entidades que visiten su web acepten los siguientes Términos y Condiciones.

Al acceder, usted confirma la aceptación de los siguientes Términos:

- El mínimo número de días que se podrá reservar es de 5.
- Una reserva cancelada con más de 2 días de antelación no tendrá coste.
- Una reserva cancelada con menos de 2 días de antelación tendrá un coste del 60% del total.
- No está permitida la entrada de mascotas a la villa.
- El número máximo de personas será 12 entre adultos y niños.
- Se considera niña/o al menor de 12 años.
- El coste será de 345€/día siempre que el número de adultos sea menor o igual a 5. En caso contrario, el coste será de 450€ día

## Ley de Protección de Datos

villasonneland.com garantiza el uso de la LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de Derechos Personales).

Para más información puede consultar el siguiente documento.

[Ley de Protección de Datos](#)

---

Un apartado fundamental es la recuperación del código de reserva en caso de pérdida.

En los distintos formularios donde se pide la introducción del código de reserva, se ha añadido un enlace al apartado para recuperarlo.

Funciona mediante el sistema de “doble paso”, pero esta vez hay que introducir el número de móvil y el *mail*.

## ¿Ha olvidado su código?

Para recuperarlos, debe introducir los siguientes datos

Teléfono:

Mail:

Enviar

Una vez introducido y comprobado que ese **Teléfono y Mail** corresponden a una reserva, se procede a enviar el código al correo electrónico.

Está contemplado el hecho de que un cliente tenga más de una reserva con el mismo Nombre, Apellido, Teléfono y Mail.

En dicho caso, lo que se hace es enviar al correo un mensaje con estilo HTML donde se incluye una tabla con las distintas reservas hechas y sus códigos asociados.



### Recuperación del código de reserva de: villasonneland@gmail.com

villasonneland@gmail.com <villasonneland@gmail.com>  
Para: jesuszquierdopuebla@gmail.com

**jesus izquierdo Su código de reserva es:**

**Usted tiene 9 reservas realizadas con nosotros**

Checkin	Checkout	Código
21/12/2019	26/12/2019	L6Q24yGc
06/01/2020	14/01/2020	tXk3gbqv
16/01/2020	23/01/2020	5Zpnql9C
12/03/2020	19/03/2020	KqmMkG8N
24/01/2020	29/01/2020	UvPtLaG3
05/08/2020	12/08/2020	e79zdy34
05/08/2020	12/08/2020	e79zdy34
20/03/2020	25/03/2020	oBdyKIMD
20/03/2020	25/03/2020	oBdyKIMD

## 5.5.7 RESPONSIVE

Como mandan los cánones actuales, el proyecto ha sido adaptado a un diseño *responsive* el cual implica que el diseño web ha sido adaptado para la visualización en distintos formatos de pantalla (Monitores, Tablets y Moviles).

Gracias al uso del **Framework Bootstrap** y su modelo *grid*, la estructura de los elementos del código se ha hecho mediante el empleo de etiquetas *div* con la clase *container* y clase *row*.

```
<div id="rootContainer" style="min-height: 100%; min-width: 100%; background-size: cover">  
  <div class="row">  
    <div class="col-12 col-md-10 mx-auto" style="margin-top: 20%!important">
```

De forma simple, podemos decir que el modelo *grid* consiste en la división de la página en 12 columnas por fila (*row*).

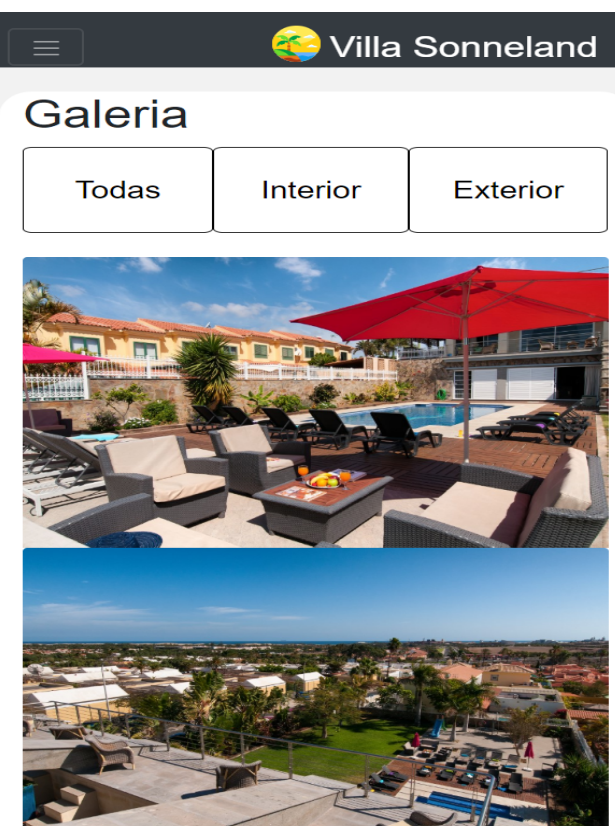
Para poder elegir el número de elementos a mostrar por fila, se hace haciendo uso de las siguientes clases.

Pantalla	Prefijo de la clase	Ancho del contenedor
Tamaño extra pequeño Teléfonos (<768px)	.col-xs-	Ninguno (automático)
Tamaño pequeño Tablets (≥768px)	.col-sm-	750px
Tamaño medio Escritorios (≥992px)	.col-md-	970px
Tamaño grande Escritorios (≥1200px)	.col-lg-	1170px

Si nos decantamos por adaptar una página a una pantalla pequeña (menor a 768px) usamos la clase **col-xs** y a continuación especificamos en cuantas columnas queremos partir esa fila de la siguiente forma **col-xs-3**, **col-xs-4**, **col-xs-12**...

A continuación, se adjuntan 4 capturas de pantalla de diferentes apartados de la web adaptados a pantalla móvil.

The screenshot shows the top navigation bar with a hamburger menu icon and the Villa Sonneland logo. Below the header, there are four input fields for search criteria: 'Check-in', 'Check-out', 'Adultos', and 'Niños'. Each field has a calendar icon on the right. A yellow 'Buscar' button is positioned below the input fields. At the bottom, there is a checkbox labeled 'He leído y acepto la política de privacidad'.



The screenshot shows a screen titled '¿Desea modificar o cancelar su reserva?'. It includes a text input field for 'Su código de reserva es:'. Below this field are two buttons: 'Modificar' (blue) and 'Eliminar' (red). At the bottom, there is a link that says '¿Ha olvidado su código?'.

The screenshot displays the 'Contaco' (Contact) screen. It features a header with the title 'Contaco' and a sub-header 'Rellene el formulario y le contestaremos lo antes posible.' Below this, there are three input fields: 'Nombre', 'Email', and 'Mensaje'. A large blue 'Enviar' button is located at the bottom of the form.

## 6. CONCLUSIONES Y TRABAJOS FUTUROS

En este apartado vamos a desglosar las conclusiones en diferentes apartados, para tener una idea clara de lo que ha supuesto el desarrollo de este proyecto.

### 6.1 RESULTADOS OBTENIDOS

Las conclusiones obtenidas tras la finalización del proyecto son positivas.

El trabajo funciona según las demandas exigidas en el inicio, además durante su desarrollo fueron sugeridos algunos añadidos extra (Valoraciones, fotos con *Javascript*, verificación doble paso), que fueron aceptados de buen grado por parte del cliente (dueño de la vivienda).

El diseño de la aplicación es más que correcto y funcional, ajustándose al aspecto de páginas similares de empresas del mismo sector.

La página es operativa al 100%. Durante su desarrollo se ha realizado una batería de pruebas de funcionamiento y estabilidad para subsanar posibles errores.

### 6.2 CONSECUCIÓN DE OBJETIVOS

Los objetivos planteados desde el inicio se han conseguido llevar a cabo en este proyecto de manera satisfactoria.

Desde que el usuario accede a la web, encuentra los recursos necesarios de forma fácil para realizar una reserva, así como otras opciones: editar, eliminar, contactar o puntuar.

Los objetivos planteados sobre seguridad han sido cumplidos con éxito, ya que finalmente se implementó un sistema de verificación de doble paso que aporta una capa extra de seguridad.

Gracias al empleo de distintas librerías tales como *Bootstrap*, *Font-awesome*, se ha conseguido dotar a la aplicación de un diseño profesional.



## 6.3 POSIBLES EXTENSIONES

El proyecto es funcional y lo hace de forma satisfactoria, pero eso no impide que quede margen de mejora, así como la posibilidad de añadir, en un futuro, funciones extra que han quedado fuera del diseño del proyecto por falta de recursos.

Las posibles extensiones que se pueden hacer son las siguientes:

- **Pasarela de pago:** Al momento de realizar este proyecto, "**Villa Sonneland**" trabaja junto a la empresa "**VillaGranCanaria**" la cual proporciona la cartera de turistas a la vivienda de clientes.

En un futuro, si se quiere ganar en autonomía, sería un aporte fundamental dotar a la aplicación de una pasarela de pago, ya que de momento lo que se hace es realizar la reservas, pero no se hacen los cobros.

- **Verificación doble paso real:** Como ya se comentó, la verificación de doble paso que se está empleando no es más que una simulación, que funciona de forma correcta, pero en el momento que se lance la aplicación para ser funcional en un entorno real, será necesario la contratación de un servicio que proporcione la posibilidad de enviar SMS al teléfono o dispositivo del usuario.

No se ha implementado de esa forma, ya que todas las empresas que proporcionan este servicio requerían de la creación de una cuenta de pago. La otra alternativa fue hacer uso de la **API** "*Google authenticator*", pero ésta requería que el usuario que quiera hacer uso de este tipo de verificación tenga instalada en su equipo la aplicación propia de Google.

- **Multipropiedad:** Con el diseño terminado y todos los elementos que conforman la aplicación, la adaptación de la aplicación a otra "**Villa**" simplemente consistiría en adaptar el diseño a las necesidades del nuevo cliente.
- **Idiomas:** Actualmente se han establecido 3 idiomas con los que el usuario de la aplicación puede interactuar.



En un futuro, cabe la posibilidad de añadir idiomas nuevos, lo que implicaría la traducción de todos los elementos que forman el *array* que contiene las palabras o frases.

- **Aplicación Móvil:** La posibilidad de contar con una Aplicación Móvil propia sería un añadido fantástico para “Villa Sonneland”.

La realización de dicha aplicación podría ser una idea para un nuevo “TFG” debido a los conocimientos necesarios que hay que tener para ello. Además, se deberían realizar 2 aplicaciones independientes, una para IOS y otra para Android.

## 6.4 CONCLUSIONES PERSONALES

La realización de este proyecto ha supuesto la prueba de que una vez finalizadas todas las asignaturas del título de Grado en Ingeniería Informática, soy capaz de llevar a cabo un proyecto desde la fase inicial de diseño, hasta su completo funcionamiento.

Al ser un proyecto propuesto por mí y no elegir uno de la bolsa de trabajos disponibles, me ha hecho estar muy motivado con su desarrollo.

Ha sido un reto que se ha afrontado con ilusión y que me deja muy satisfecho por los resultados obtenidos.

Con su desarrollo he podido incrementar mis conocimientos dentro del ámbito de la programación y me ha permitido hacer uso de herramientas que no había usado hasta ahora, así como aprender a gestionar el tiempo y los recursos con los que cuento.

He sido consciente de la diferencia que existe en el contenido estudiado en las diferentes asignaturas de programación y lo que es realizar un proyecto para un entorno real.

## 7. BIBLIOGRAFÍA

Para la realización de este trabajo se han realizado consultas en multitud de lugares con los que poner solución a los problemas que fueron surgiendo durante la fase de desarrollo de la aplicación.

**Bootstrap:** Se ha consultado la web oficial de bootstrap para la implementación de diversos elementos de este *framework*, tales como *navbar*, galería de imágenes, o la implementación del sistema *grid* para estructurar el diseño de la web y poder adaptarlo al modelo *responsive*.

<https://getbootstrap.com/>

**W3School:** Se ha consultado la web W3School para ayuda en diferentes apartados del código como son generación de código en lenguajes PHP, HTML, CSS, SQL.

El apartado más consultado en esta web ha sido lo referente a CSS para hacer un buen uso de las diferentes hojas de estilo empleadas en este proyecto.

También ha sido muy útil la posibilidad que daba con un entorno de generación de colores.

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

**Manual oficial PHP:** Durante el desarrollo de este trabajo, contar con la web oficial del lenguaje más empleado en la elaboración del proyecto ha sido fundamental, ya que en ella he podido realizar diversas consultas necesarias para conseguir los resultados propuestos.

Al tratarse del sitio oficial, la posibilidad de errores en las consultas se reducen al mínimo.

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

**Wikipedia:** El uso de la enciclopedia en línea Wikipedia ha sido muy útil para la consulta de diferentes términos desconocidos que aparecieron durante el desarrollo del proyecto.

<http://wikipedia.org/>

**Stack Overflow:** El foro de desarrolladores *Stack Overflow* ha sido una herramienta indispensable para la solución de problemas que van surgiendo durante el desarrollo de la actividad.

En dicho foro existen miles de mensajes con gente que tiene un problema similar al tuyo y puedes consultar las soluciones propuestas.

<https://es.stackoverflow.com/>

**Manual MySQL:** Para la consulta de configuración de la base de datos, así como el empleo de sentencias SQL y solución de los diferentes errores que aparecen en la elaboración de la base de datos, ha sido fundamental la consulta de la guía oficial MySQL.

<https://dev.mysql.com/doc/refman/8.0/en/>

**LOPDGDD:** Para la aplicación de la "Ley de Protección de Datos", se ha consultado ésta web:

<https://www.cookiebot.com/es/lopdgdd/>

Era un apartado que queda muy lejos de mi campo de estudio, por lo que necesite la información disponible en esta web para saber como aplicarla de forma adecuada.

El documento PDF oficial también se obtuvo de esta página.

**Youtube:** *Youtube* ha sido una herramienta muy útil para la consulta de diferentes tutoriales de cualquier lenguaje empleado en este proyecto.

La visualización del desarrollo de técnicas de programación, consejos y demás ha sido una ayuda fundamental en este proyecto.

Fue fundamental la visualización de diversos tutoriales para poder entender la forma correcta de hacer uso de la librería TCPDF.

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

**Apuntes de Asignatura:** Se ha realizado consultas en los apuntes de diferentes asignaturas para poder llevar a buen puerto la aplicación.

Las asignaturas que han sido fundamentales para este trabajo han sido.

**Base de Datos 1 y 2:** Fundamental para la creación de este proyecto, ya que sin los conocimientos recabados en esta asignatura, no se habría podido implementar parte fundamental de la actividad.

No sólo se han empleado únicamente conocimientos técnicos y prácticos, si no que se han empleado herramientas que conocí en dichas asignaturas.

**Redes de Computadores y Servicios de seguridad en Red:** Los conocimientos adquiridos en estas asignaturas han sido fundamentales para la puesta a punto del servidor donde se ha ejecutado este proyecto, así como la creación del servidor de correo electrónico.

**Programación IV:** He hecho uso de conocimientos obtenidos en esta asignatura relacionados con los lenguajes PHP, HTML, *Javascript*.

En este punto debo decir que aunque es cierto que se han realizado consultas en sus apuntes, la gran mayoría de dudas relacionadas con programación web han sido subsanadas en los apartados mencionados anteriormente, ya que el contenido de esta asignatura está muy desfasado con respecto a lo que se está empleando en un entorno real: el desconocimiento de un *framework* PHP para poder trabajar con él; el no enseñar un patrón de arquitectura de software como, por ejemplo, Modelo-Vista-Controlador; no trabajar con un sistema de control de versiones tipo Github, me ha perjudicado en gran medida. Enseñar en una misma asignatura de 4 meses HTML, CSS, PHP, JAVASCRIPT, AJAX, Programación funcional, Programación Lógica y Programación concurrente es un error, ya que acabas aprendiendo muy poco de todo lo mencionado.

**Fundamentos de sistemas Operativos y Administración de sistemas operativos:** Los conocimientos recabados en estas asignaturas han sido de gran ayuda, sobre todo para conocer el sistema donde estaba trabajando, cómo se almacenan los datos, así como el uso de *scripts* y el empleo del BASH que me ha hecho mi actividad mucho más sencilla.