

PROYECTO INTEGRADO

DESARROLLO DE APLICACIONES WEB



Las Fuentezuelas
INSTITUTO DE EDUCACIÓN SECUNDARIA

- DEPARTAMENTO DE INFORMÁTICA -

ÍNDICE DE CONTENIDO.

Capítulo 1: Introducción, descripción y propósito del proyecto.

Capítulo 2: Objetivos del proyecto.

Capítulo 3: Base de datos. Diseño e implementación.

Capítulo 4: Etapas por las que ha pasado el desarrollo del proyecto.

Capítulo 5: Requisitos de la aplicación.

Capítulo 6: Manual de usuario.

Capítulo 7: Manual de instalación.

Capítulo 8: Código fuente y en su caso paquete de instalación.

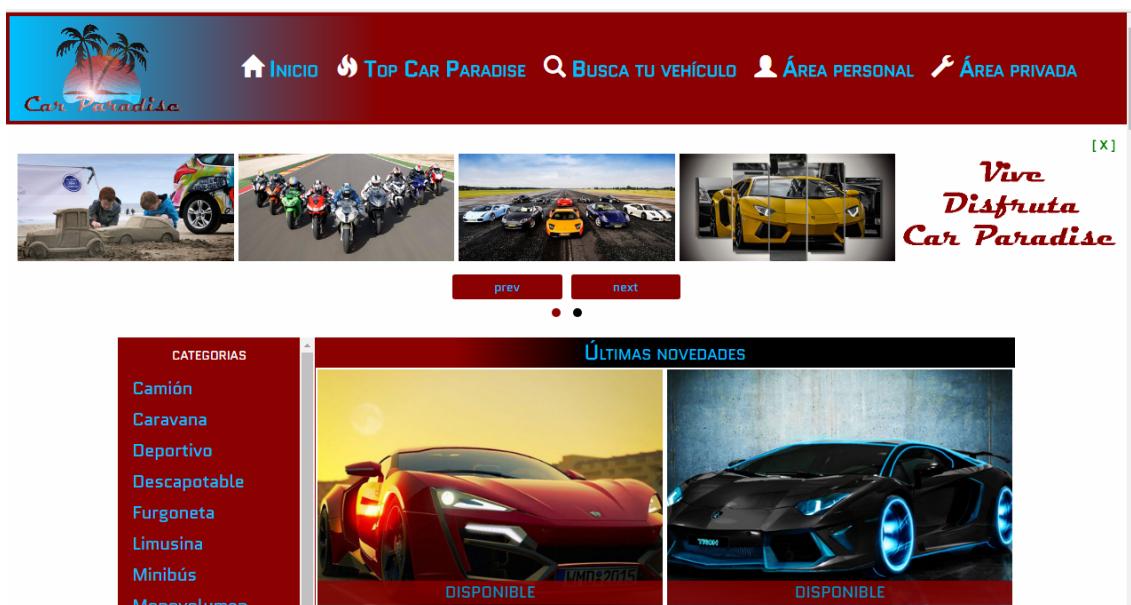
Capítulo 9: Presentación para la defensa de la aplicación.

Capítulo 1:

Introducción, descripción y propósito del proyecto.

Introducción.

- Car Paradise es una amplia empresa dedicada al alquiler de vehículos de todo tipo por un tiempo a convenir. En los últimos años dicha empresa ha crecido considerablemente hasta el punto de convertirse incluso en una gran multinacional especializada en vehículos de alta gama, deportivos y dándose fama con los vehículos de lujo en toda España desbancando a otras grandes sucursales de prestigio.



- Carparadise.dpm es una plataforma WEB diseñada y desarrollada en esencia para sus empleados y no menos importante para los consumidores, sus clientes. De igual manera se le ofrecen idénticos servicios a los usuarios que sin ser clientes puede reservar su vehículo, también es cierto que dichos usuarios disponen de un plazo estimado ya que si no hicieran el pago en oficina, su reserva quedaría expirada y el vehículo estaría de nuevo disponible automáticamente tanto en oficina como en la WEB.

FICHA DE RESERVA [1909JTG]

Datos del vehículo a reservar.

- ★ Matrícula: **1909JTG**.
- ★ Marca: **Lamborghini**.
- ★ Modelo: **Huracán**.
- ★ Kilómetros: **220**.
- ★ Combustible: **Gasolina**.
- ★ Potencia: **770 CV**.
- ★ Código base: **20016**.
- ★ Ubicación: **Madrid**.
- ★ Grupo de categoría: **Y**.
- ★ Estado actual: **Disponible**.
- ★ Color: **Rojo**.
- ★ Observaciones: **Reservable hasta el 14/11/19.**

[Ver video](#)



Datos necesarios para la reserva.

Fecha inicio:	Fecha fin:	Referencia online:
<input type="text" value="08/06/2017"/>	<input type="text" value="08/06/2017"/>	<input type="text" value="DNI del cliente"/>
Precio base (€):	Descuento online (%):	Precio final (€):
<input type="text" value=""/>	<input type="text" value="15"/>	<input type="text" value=""/>

[Reservar](#)

- El objetivo de dicha WEB es la de mostrar de forma golosa la cantidad y variedad de vehículos disponibles clasificado en un apartado con las 2 últimas novedades y otro de 12 fichas que se recargará de forma aleatoria para disfrutar de su disposición como si en el paraíso se tratase, de ahí el título como eslogan. Sino se encontrase un vehículo en particular, le pondremos a su disposición una amplia búsqueda de vehículos junto un acceso para ir al área de reserva en la que aparecería su mejor foto y / o video ilustrativo.

CATEGORIAS

- [Camión](#)
- [Caravana](#)
- [Deportivo](#)
- [Descapotable](#)
- [Furgoneta](#)
- [Limusina](#)
- [Minibús](#)
- [Monovolumen](#)
- [Motocicleta](#)
- [Turismo L](#)

TURISMO M: ÚLTIMAS NOVEDADES



DISPONIBLE

Subaru WRX
180 CV. Ubicación: Chamartín (Madrid).



DISPONIBLE

Volkswagen Golf
130 CV. Ubicación: Sagunto (Valencia).

TURISMO M: RECOMENDACIONES DISPONIBLES



Peugeot 308
90 CV con 220 Kms (Diésel).
Ubicación: ADIF A Coruña (A Coruña).

¡RESÉRVALO AQUÍ!



Renault Megane
110 CV con 332 Kms (Gasolina).
Ubicación: La Loma (Jaén).

¡RESÉRVALO AQUÍ!

Descripción.

- Se lleva a cabo la gestión de mantenimiento completo de Car Paradise en la que podremos actualizar datos de los campos de las entidades relacionadas que más tarde explicaremos en la zona de base de datos:
 - Tipo de empleados.
 - Empleados.
 - Cliente [registrado o anónimo temporal online].
 - Categoría de vehículo.
 - Base de ubicación del vehículo.
 - Vehículo.
 - Bono de ofertas para el alquiler.
 - Alquiler de vehículos [creable y borrible pero innecesariamente modificable].


Daniel Pancorbo Miranda
Thursday 08 June

Mantenimiento: Índice.

Menú principal *Car Paradise*

Vehículos		
Alquileres		
Clientes		
Bases	Bonos	Categorías
Tipos / Empleados		

- Se habilita un área para el empleado. Para entrar dentro de la administración [Área privada] se necesita tener acceso al índice de mantenimiento con un apodo de empleado y su respectiva contraseña o clave de acceso, con esto podremos entrar o restringir el acceso a nuestros empleados según el tipo que sea, es decir, no es lo mismo que entre un administrador o empleado indefinido que un empleado en prácticas o temporal. PHP.



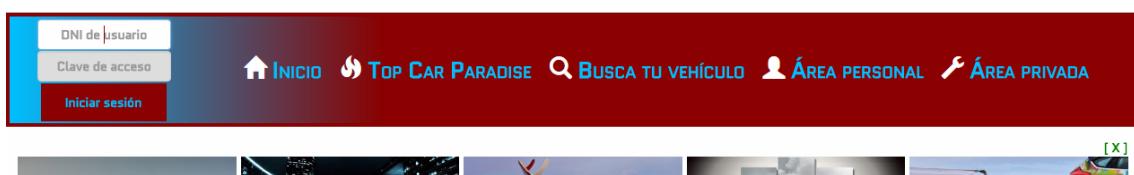
- Área privada ***Car Paradise*** para empleados -

Car Paradise

Conecta

Salir de área privada

- No menos importante, se habilita un área para el cliente. Para poder loguearse e iniciar sesión como cliente ya registrado [Área personal], se necesita, un DNI válido y contraseña o clave de acceso correcta de dicho cliente. Una vez iniciado la sesión, su menú de navegación cambiará y podremos ver un nuevo apartado denominado "Mi historial" con el detalle del historial de vehículos alquilados en Car Paradise del cliente que ha iniciado la sesión en forma de acordeón. JavaScript.



Propósito.

- El propósito de dicha WEB es la de incentivar en nuestra empresa el alquiler vacacional tanto en la costa como en el centro del país, el alquiler diario por gusto, por placer, y el alquiler laboral que necesitan empresas de renting ofreciéndoles lo mejores de nuestros servicios y precios con nuestra amplia gama de bonos de ofertas.



The screenshot shows a maintenance page titled "Mantenimiento: Bonos." at the top right. Below it is a table titled "Nuevo bono de oferta" (New offer bonus) with columns for Descuento (Discount), Descripción (Description), Alquileres (Rentals), Modificar (Modify), and Borrar (Delete). The table lists various discount offers:

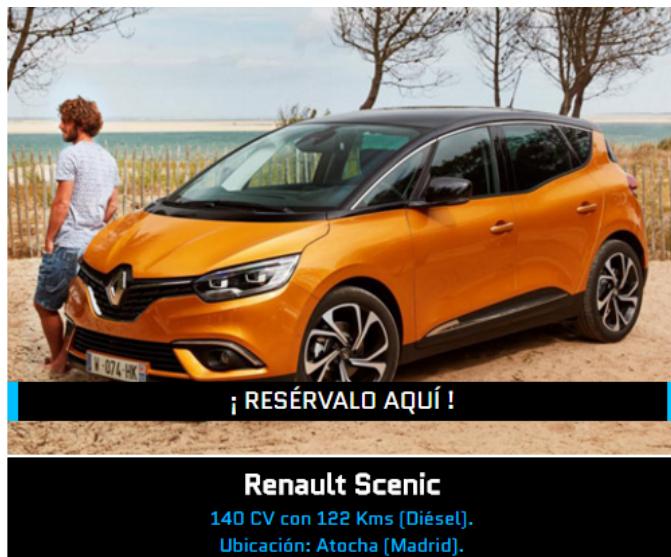
Descuento	Descripción	Alquileres	Modificar	Borrar
0 %	Sin descuento			
2 %	Lujó Ferrari			
5 %	Fin de semana			
5 %	Summer paradise			
8 %	Seat pasión			
10 %	Vacaciones Paradise			
12 %	Escapada Renault			
20 %	Opel al límite			
33 %	Audi 3 días 1 aventura			
33 %	Honda Naranja			
35 %	Navidad Suzuki			

- Conseguiremos hacer de forma cómoda y rápida el acceso de compra de lo que se le ofrece [alquiler de vehículos]. Para ello, tendremos multitud de enlaces directos para dirigirnos al apartado / ficha para alquilar / reservar en:
- La parte inferior de la caja de cada vehículo de forma de transición [la veremos tanto en el principal como en la página de la categoría elegida].
 - Un iconito mostrado en el apartado de búsqueda cuando realizamos una búsqueda con éxito.



The screenshot shows a search results page with the header "BÚSQUEDA CON FILTRO: 2 VEHÍCULO(S) ENCONTRADO(S)". Below it is a search bar with dropdown menus for Categoría, Marca, Scenic (highlighted in yellow), Combustible, Kilometraje, and Base [Localidad]. A "Filtrar" button is also present. The main area displays a table with vehicle details:

Color	Categoría	Matrícula	Marca	Modelo	Km	Combustible	Base	Ubicación	Estado	Reservar
	Monovolumen	2443JSS	Renault	Grand Scenic	3332	Diesel	Sants	Barcelona	Disponible	
	Monovolumen	3656JJK	Renault	Scenic	122	Diesel	Atocha	Madrid	Disponible	



- En la parte superior izquierda se habilita un mini menú cliqueable de nuestras categorías de vehículos ordenadas alfabéticamente donde podemos buscar y mostrar en nuestra WEB los vehículos clasificados en dicha categoría. El usuario de esta forma iría más directamente al vehículo que busca en particular y de este modo le ahorraremos tiempo.



- Ayuda y facilidad para recalar información de alguno de nuestros vehículos, de igual manera pondremos a su disposición listados TOP 10 por marcas, modelos, bonos, clientes, bases del mes anterior para descargar y las mostradas online en dicho apartado. Este apartado es esencial ya que podremos ofrecer al usuario que navega y catalogaremos como usuario parcialmente interesado en un vehículo u oferta concreta, de una buena y optimizada lista de lo mejor que se alquila / reserva en nuestra empresa.

Top marcas de vehículos ↓		
Posición	Marca	Cantidad
19.	Renault	10
29.	Ford	7
39.	Audi	5
49.	Ferrari	5
59.	Lamborghini	5
69.	Peugeot	4
79.	Volkswagen	4
89.	Kawasaki	3
99.	Seat	3
109.	Toyota	3

Top modelos de vehículos ↓		
Posición	Modelo	Cantidad
19.	Megane	6
29.	Focus	4
39.	California	3
49.	Ninja	3
59.	Prius	3
69.	Scenic	3
79.	308	2
89.	A3	2
99.	Aventador	2
109.	C4	2

Top bases en localidades ↓			
Posición	Base	Localidad	Cantidad
19.	Manresa	Barcelona	13
29.	Atocha	Madrid	12
39.	Chamartín	Madrid	9
49.	Mar Menor	Murcia	7

- Dar publicidad a nuestra empresa con un banner ampliamente llamativo que opcionalmente podríamos ocultar, en él, nuestros más prestigiosos grupos de vehículos o eslóganes lo mostraremos de forma de carrusel paginado reproducido de manera automática y de la misma manera podremos detener si lo encontramos oportuno. Lo situamos en la parte superior de todas nuestras páginas y lo podremos cerrar cuando deseemos o realicemos una acción de búsqueda o de reserva.



- Ofrecemos una amplia gama de marcas pertenecientes a los vehículos de enorme peso comercial que demandan los clientes más selectos tanto a nivel mundial como a nivel propio en nuestra empresa.



- No menos importante es mostrar nuestra información de forma fija en el pie en la que se podría contactar. Teléfono de contacto y correo electrónico puede cambiar con un solo click facilitando su información para comunicarse.

Ver email / teléfono:  [+34] 650 801 568 [INICIO](#) | [TOP CAR PARADISE](#) | [ÁREA PRIVADA](#) © Copyright 2017

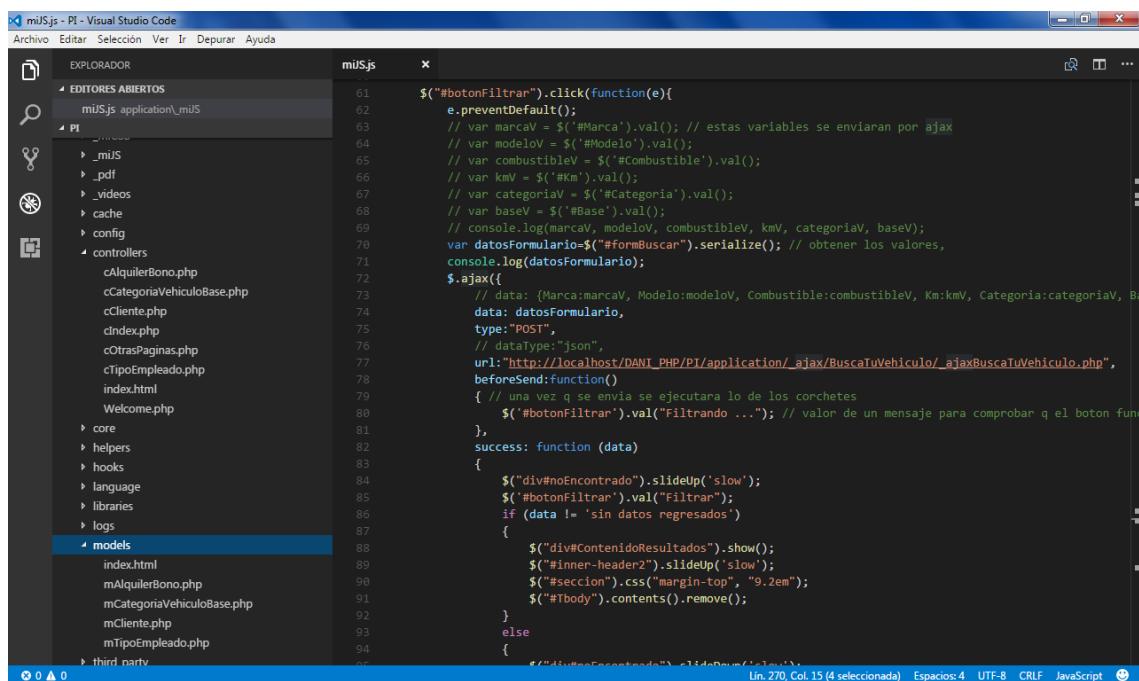
Ver email / teléfono:  info@carparadise.dpm [INICIO](#) | [TOP CAR PARADISE](#) | [ÁREA PRIVADA](#) © Copyright 2017

Capítulo 2: Objetivos del proyecto.

- El objetivo principal del proyecto es el aprendizaje y adecuación de los lenguajes de programación estudiados en estos dos años a lo largo del curso.
- Maquetación y estilos correctamente usados para una organización limpia y estructurada. Cargaremos dos librerías para ello a parte de nuestra propia hoja de estilos [miCSS.css], Bootstrap y OwlCarousel. **HTML y CSS**.
 - Uso correcto del patrón modelo – vista – controlador **MVC** en el framework “**CodeIgniter**” en el que iniciaremos con sus correspondientes controladores con su función concreta en que utilizaremos parámetros si se necesitara, utilizaremos llamadas a la base de datos en sus modelos y generaremos múltiples vistas con cabecera, cuerpo y pie para proporcionar la información necesaria a mostrar hacia donde navegemos. De igual manera validaremos los datos dentro de nuestros formularios. Uso de librerías propias para abrir y cerrar nuestro **HTML** o uso de librerías del propio framework. Especificaremos algunas rutas para darle mayor dinamismo y configuraremos nuestro controlador predeterminado y nuestro acceso a la base de datos de phpMyAdmin. **PHP**.
 - Utilizaremos **JavaScript** mediante **JQuery** para el envío de dos datos esenciales (DNI y clave de acceso) para realizar un posible inicio de sesión, cliente que se loguea, enviaremos un **AJAX** por **POST**, que validaremos con antelación, en el que realizaremos una consulta mediante **MYSQLI** para fabricar una variable de sesión con el DNI y Nombre y apellidos del cliente encontrado en dicha consulta a nuestra base de datos. Al realizar esto con éxito, se activa el enlace del historial del

cliente y se desactiva cuando evidentemente pinchemos en “desconexión” para destruir dicha variable de sesión y volver redirigido a nuestra página principal. Sino se encontrara dicho cliente o faltará algún campo, mandaremos mensajes de información para alertar al usuario de que está sucediendo.

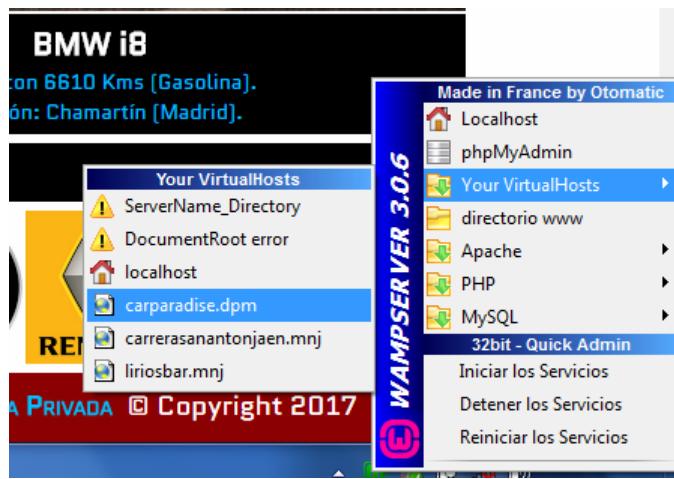
- Lograremos con **JQuery**, fabricar un acordeón para visualizar de forma opcional el detalle de un vehículo concreto en nuestro apartado de sesión iniciada de mi historial.
- Tambiénaremos uso de **JQuery** para enviar un **AJAX** por **POST** a un página PHP donde realizaremos una gran búsqueda con multitud de diferentes criterios que de igual manera enviaremos serializados, con este paquete mandado, se realizará la consulta y recorreremos con un bucle en **PDO** los datos para devolver una respuesta codificada en **JSON** que después de igual manera recorreremos con un bucle con nuestro JavaScript [miJS.js] con un bucle y lo estos datos lo meteremos en una amplia tabla. Si no existen datos, mostraremos un mensaje pero no limpiaremos la búsqueda anterior, es decir, no limpiaremos la tabla con Jquery.



```

miJS.js - PI - Visual Studio Code
Archivo Editar Selección Ver Ir Depurar Ayuda
EXPLORADOR          miJS.js      ...
EDITOR ABIERTOS
  miJS.js application\miJS
  PJ
    _miJS
    _pdf
    _videos
    cache
    config
    controllers
      cAlquilerBono.php
      cCategoriaVehiculoBase.php
      cCliente.php
      cIndex.php
      cOtrasPaginas.php
      cTipoEmpleado.php
      index.html
      Welcome.php
    core
    helpers
    hooks
    language
    libraries
    logs
    models
      index.html
      mAlquilerBono.php
      mCategoriaVehiculoBase.php
      mCliente.php
      mTipoEmpleado.php
    third_party
miJS.js
61  $("#botonFiltrar").click(function(e){
62    e.preventDefault();
63    // var marcaV = $('#Marca').val(); // estas variables se enviaran por ajax
64    // var modeloV = $('#Modelo').val();
65    // var combustibleV = $('#Combustible').val();
66    // var kmV = $('#Km').val();
67    // var categoriaV = $('#Categoria').val();
68    // var baseV = $('#Base').val();
69    // console.log(marcaV, modeloV, combustibleV, kmV, categoriaV, baseV);
70    var datosFormulario=$('#formulario').serialize(); // obtener los valores,
71    console.log(datosFormulario);
72    $.ajax({
73      // data: {Marca:marcaV, Modelo:modeloV, Combustible:combustibleV, Km:kmV, Categoria:categoriaV, Base:baseV},
74      data: datosFormulario,
75      type:"POST",
76      // dataType:"json",
77      url:"http://localhost/DANI_PHP/PI/application/_ajax/BuscaTuVehiculo/_ajaxBuscaTuVehiculo.php",
78      beforeSend:function()
79        { // una vez q se envia se ejecutara lo de los corchetes
80          $('#botonFiltrar').val("Filtrando ..."); // valor de un mensaje para comprobar q el boton funciona
81        },
82        success: function (data)
83        {
84          $("#div#noEncontrado").slideUp('slow');
85          $('#botonFiltrar').val("Filtrar");
86          if (data != 'sin datos regresados')
87          {
88            $("#div#ContenidoResultados").show();
89            $("#inner-header2").slideUp('slow');
90            $("#seccion").css("margin-top", "9.2em");
91            $("#tbody").contents().remove();
92          }
93          else
94          {
95            $("#div#ContenidoResultados").show();
96            $("#inner-header2").slideUp('slow');
97            $("#seccion").css("margin-top", "9.2em");
98            $("#tbody").contents().remove();
99          }
100        }
101      });
102    }
103  );
104  
```

- Generaremos un **Virtual Host** para trabajar en el dominio **carparadise.dpm** que apunte a nuestro localhost (temporal), con esto también conseguiremos ocultar nuestro alias de WampServer.



- Ayuda, rapidez, buena apariencia, agilidad y orden para programar la lograremos con la aplicación denominada **Visual Studio Code**, en la cual, instalaremos nuevas extensiones muy valiosas para ello.

```

<div id="principal">
    <div id="inner-principal">
        <div id="menuCategorias"><p>Categorías</p>
            <ul>
                <?php
                    foreach ($categoriasMenu as $categoria)
                {
                    >
                    <li><a href=?>?php echo site_url ('cOtrasPaginas/fPinchaCategoria/' . $categoria['GrupoCategoria'] . '/' . $categoria['NombreCategoria'])?</a></li>
                }
            </ul>
        </div>
        <div id="novedades">
            <h3>Últimas novedades</h3>
            <?php
                foreach ($vehiculos2Novedades as $v2N)
            {
                >
                <div class="vehiculos2Novedades">
                    <div class="relativo">
                        <?php echo "<div class='estadio'><h4>" . $v2N["Estado"] . "</h4></div>" ?>
                        <?php echo "<img width='100%' height='auto' src='data:image/jpeg;base64," . base64_encode($v2N["Imagen"]) . "'/>" ?>
                    </div>
                    <p><?php echo "<span class='colorMarcaModelo>" . $v2N["Marca"] . " - " . $v2N["Modelo"] . "</span>" ?></p>
                </div>
            }
        </div>
    </div>
</div>
<div id="resto">
</div>

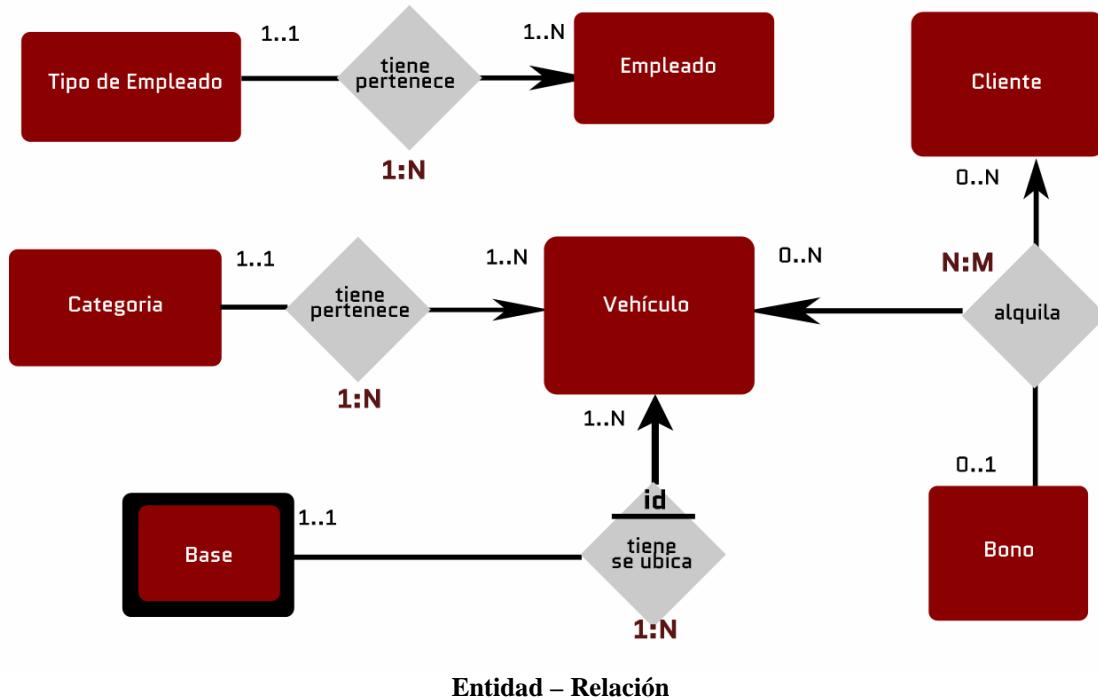
```

Capítulo 3:

Base de datos relacional, MySQL.

Ha sido creada con comandos y relacionada tras activar el diseñador de phpMyAdmin, tabla a tabla, secuencia a secuencia, sin el asistente de ayuda.

- Esquema entidad / relación.



- Transformación del esquema E-R a relacional.

- **Tipo** [IdTipo, NombreTipo].
- **Empleado** [ApellidoEmpleado, NombreEmpleado, Contraseña, IdTipo [FK, not null]].
- **Categoría** [GrupoCategoria, NombreCategoria].
- **Base** [CódigoBase, NombreBase, Localidad].
- **Cliente** [Dni, NombreCliente, Apellido1Cliente, Apellido2Cliente, Teléfono, Contraseña].
- **Vehículo** [Matrícula, Marca, Modelo, Km, Combustible, Potencia, FechaExpiración, Estado, Imagen, Color, CódigoBase [FK, not null], GrupoCategoria [FK, not null]].
- **Bono** [IdBono, Descripción, Descuento].
- **Cliente_alquila_Vehículo ALQUILER** [Dni [FK], Matrícula [FK], Fechainicio, FechaFin, PrecioAlquiler, IdBono [FK, not null]].

➤ Tablas.

- **Tipos de empleados:** Importante para diferenciar unos empleados de otros, de quienes pueden ver más o menos información o de quienes pueden hacer actualizaciones como crear, modificar o eliminar de los que no. Por ejemplo: el empleado tipo *administrador* podrá realizar cualquier proceso mientras que los empleados de tipo *práctica*, *becario* o *temporal* solo podrá ver una lista paginada correspondiente a alquiler, lo único que se le permite es entrar, ver y salir del mantenimiento.

```

1 -- CAR PARADISE, BASE DE DATOS
2 CREATE TABLE Tipo
3 (
4     IdTipo          NUMBER (3),           -- AUTO_INCREMENT
5     NombreTipo      VARCHAR2 (30)        CONSTRAINT NONULO_NombreTipo NOT NULL,    -- 30 caracteres, no puede ser nulo
6     CONSTRAINT PK_IdTipo PRIMARY KEY (IdTipo)
7 );

```

- **Empleados:** Máximos responsables del buen mantenimiento de la aplicación y acceso a nuestra base de datos para la actualización de datos para la misma, se le restringe el acceso mediante una clave personal que debe concordar con su apodo o nick. Tan sólo puede dar de alta un empleado a otro si dispone de cierta autorización, ya que se le restringe ciertos permisos a los empleados temporales, de prácticas o becarios [autorización mínima].

```

9 CREATE TABLE Empleado
10 (
11     ApodoEmpleado   VARCHAR2 (30),       -- 30 caracteres, clave primaria
12     NombreEmpleado  VARCHAR2 (30)        CONSTRAINT NONULO_NombreEmpleado NOT NULL,    -- 30 caracteres, no puede ser nulo
13     Contrasena      VARCHAR2 (30)        CONSTRAINT NONULO_ContrasenaE NOT NULL,
14     IdTipo          NUMBER(3)           CONSTRAINT NONULO_IdTipo NOT NULL,
15     CONSTRAINT PK_ApodoEmpleado PRIMARY KEY (ApodoEmpleado),
16     CONSTRAINT FK_IdTipo FOREIGN KEY (IdTipo) REFERENCES Tipo (IdTipo) ON DELETE CASCADE ON UPDATE CASCADE
17 );
18

```

- **Clientes:** Principales demandantes de nuestros servicios, pueden loguearse solamente si introducen su DNI y su clave de acceso o contraseña correctamente. Pueden ver su historiales en la empresa mostrando sus alquileres / reservas y un detalle específico del vehículo que contrataron ordenado de más reciente a menos.

```

42 CREATE TABLE Cliente
43 (
44     Dni           VARCHAR (9),
45     NombreCliente VARCHAR2 (30)  CONSTRAINT NONULO_NombreCliente NOT NULL,
46     Apellido1Cliente VARCHAR2 (30)  CONSTRAINT NONULO_Apellido1Cliente NOT NULL,
47     Apellido2Cliente VARCHAR2 (30)  CONSTRAINT NONULO_Apellido2Cliente NOT NULL,
48     Contrasena    VARCHAR2 (30)      CONSTRAINT NONULO_ContrasenaC NOT NULL,
49     Telefono      VARCHAR2 (12)     CONSTRAINT NONULO_TelefonoCliente NOT NULL,
50     CONSTRAINT PK_Dni PRIMARY KEY (Dni)
51 );
52

```

- **Categoría:** Sirve para agrupar los vehículos y saberlo identificar en el menú rápidamente. Cada categoría clasifica de forma ordenada nuestros vehículos.

```

19 CREATE TABLE Categoria
20 (
21     GrupoCategoria   VARCHAR2 (2),          -- 2 caracteres, clave primaria
22     NombreCategoria VARCHAR2 (30)  CONSTRAINT NONULO_NombreCategoria NOT NULL,      -- 30 caracteres,
23     CONSTRAINT PK_GrupoCategoria PRIMARY KEY (GrupoCategoria)
24 );
25

```

- **Base:** Es zona dónde pertenece / se ubica el vehículo concretamente, con su base podremos averiguar rápidamente su código postal como la localidad a la que pertenece.

```

26 CREATE TABLE Base
27 (
28    CodigoBase        NUMBER(5),          -- clave primaria
29    NombreBase        VARCHAR2 (30)  CONSTRAINT NONULO_NombreBase NOT NULL,      -- 30 caracteres,
30    Localidad         VARCHAR2 (30)  CONSTRAINT NONULO_Localidad NOT NULL,      -- 30 caracteres,
31     CONSTRAINT PK_CodigoBase PRIMARY KEY (CodigoBase)
32 );

```

- **Vehículo:** Es el mayor protagonista de nuestra base de datos junto el alquiler. Con sus numerosos atributos podremos saber gran cantidad de información a su respecto. Tendremos muy controlado y detallado cada vehículo en la base de datos. Su fotografía es opcional pero extremadamente recomendable para ofrecer al usuario una vista más concreta. En nuestra base de datos no debería de faltar una imagen nítida.

```

53 CREATE TABLE Vehiculo
54 (
55     Matricula       VARCHAR (7),          -- 7 caracteres, clave primaria
56     Marca          VARCHAR2 (30)  CONSTRAINT NONULO_Marca NOT NULL,      -- 30 caracteres
57     Modelo         VARCHAR2 (30)  CONSTRAINT NONULO_Modelo NOT NULL,      -- 30 caracteres
58     Km             NUMBER(5),
59     Combustible    VARCHAR2 (30),
60     Potencia       NUMBER (3),
61     Imagen         LONGBLOB,
62     FechaExpiracion DATE,
63     Estado         VARCHAR2 (30)  CONSTRAINT NONULO_Estado NOT NULL,
64     Color          VARCHAR2 (7),          -- 7 -> hexadecimal
65     CodigoBase     NUMBER(5)    CONSTRAINT NONULO_FK_CodigoBase NOT NULL,
66     GrupoCategoria VARCHAR2 (2)    CONSTRAINT NONULO_FK_GrupoCategoria NOT NULL,
67     CONSTRAINT PK_Matricula PRIMARY KEY (Matricula),
68     CONSTRAINT FK_CodigoBase FOREIGN KEY (CodigoBase) REFERENCES Base (CodigoBase) ON DELETE CASCADE ON UPDATE CASCADE,
69     CONSTRAINT FK_GrupoCategoria FOREIGN KEY (GrupoCategoria) REFERENCES Categoria (GrupoCategoria) ON DELETE CASCADE ON UPDATE CASCADE
70 );
71

```

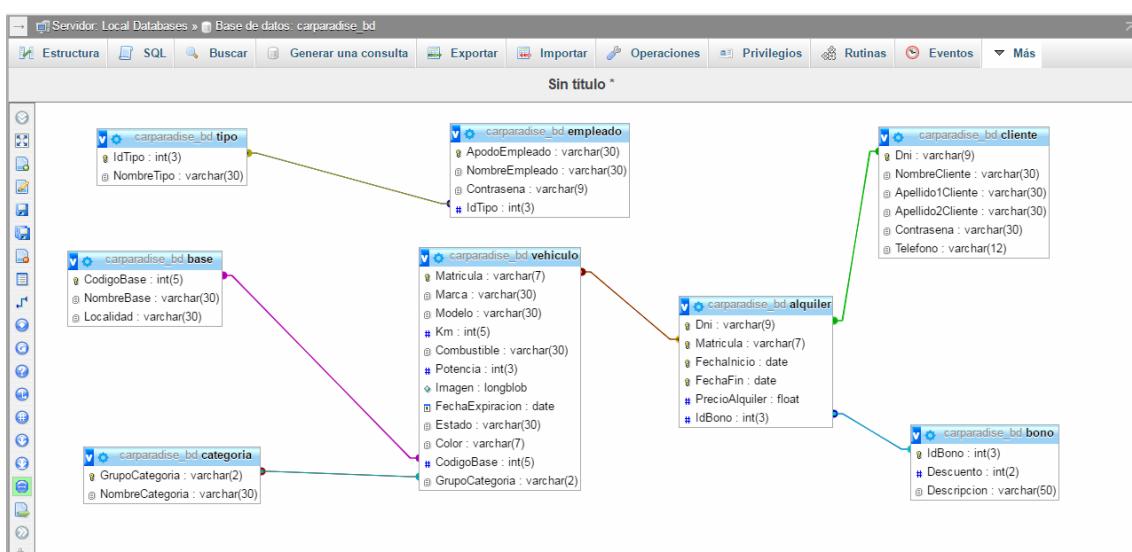
- **Alquiler:** Es indudablemente la tabla con más entradas en nuestra base de datos. Multitud de reservas se realizarán online por la WEB y gran cantidad de alquileres se efectuarán desde nuestra área privada en la que nuestros empleados con autorización llevarían a cabo. En ella podremos identificar el vehículo y el cliente junto con sus fechas de entrada y salida del mismo. Con esta tabla lograremos contar la cantidad exacta para realizar una clasificación en Top Car Paradise.

```

72  CREATE TABLE Alquiler -- Cliente_Alquila_Vehiculo
73  (
74      Dni          VARCHAR (9),
75      Matricula    VARCHAR (7),
76      FechaInicio  DATE,
77      FechaFin    DATE,
78      PrecioAlquiler NUMBER(5,2)   CONSTRAINT NONULO_PrecioAlquiler NOT NULL,
79      IdBono       NUMBER(3)     CONSTRAINT NONULO_FK_IdBono NOT NULL,
80      CONSTRAINT FK_Dni FOREIGN KEY (Dni) REFERENCES Cliente (Dni),
81      CONSTRAINT FK_Matricula FOREIGN KEY (Matricula) REFERENCES Vehiculo (Matricula),
82      CONSTRAINT FK_IdBono FOREIGN KEY (IdBono) REFERENCES Bono (IdBono) ON DELETE CASCADE ON UPDATE CASCADE,
83      CONSTRAINT PK_C_A_V PRIMARY KEY (Dni,Matricula,FechaInicio,FechaFin)
84  );

```

➤ Tablas vistas desde el diseñador de phpMyAdmin.



Capítulo 4:

Etapas por las que ha pasado el desarrollo del proyecto así como los problemas encontrados en cada una de ellas.

Etapa 0.

➤ Diseño y maquetación de CSS.

El uso constante de nuestra hoja de estilos ha sido constante y ha sido creada según gusto y necesidad, no en una etapa concreta. La fuente para mi WEB es cargada desde mi hoja de estilos que apunta la fuente, de esta manera no hace falta bajarla e instalarla en 'fonts' de Windows.

```

vCabecera.php vPie.php vindexPrincipal.php vCrearVehiculo.php mCategoriaVehiculoBase.php miCSS.css ×
1 @font-face {
2     font-family: miFuente;
3     src: url('../_fuentes/Quantico-Regular.ttf');
4 }
5 *
6     font-family: miFuente;
7 }
8 *****
9 INDEX CABECERA Y FOOTER
10 *****
11 div.relativo {
12     position: relative;
13 }
14 li::selection, td::selection, span::selection, p::selection {
15     background-color: yellow;
16     color: darkred;
17 }
18 header {
19     width: 100%;
20 }
21 .cajaLogoGif{
22     display: flex;
23     align-items: center;
24     /*justify-content: center;*/
25     height: 100%;
26 }

```

➤ Problema.

Estilos pisados por otros o imposibilidad ya que existía un estilo más específico que otro. Se soluciona constantemente sobre la marcha.

Etapa 1.

➤ Adecuación a la idea preconcebida del proyecto.

El primer punto ha sido el tipo de desarrollo WEB iba a realizar, así como el estudio del mercado, el gusto específico de los vehículos o la idea principal del corazón al que se dedicaría el proyecto.

➤ **Problema.**

Duda para dedicar nuestro proyecto al vehículo o al alquiler. Fue como fuese, uno de los mayores problemas que tardamente me he dado cuenta, ha sido el no tener en cuenta de inicio una WEB dedicada a la disponibilidad por fechas, como hacen otras páginas similares.

Mi primera idea radicaba en la importancia de los vehículos, secundados por los alquileres, pasados los dos tercios de mi proyecto, descubrí que sería más idóneo saber la disponibilidad por fechas, y acto seguido enseñar nuestros vehículos. Una información más exhaustiva de aplicaciones similares hubiera subsanado mi error o aclarado mis dudas respecto al tema.

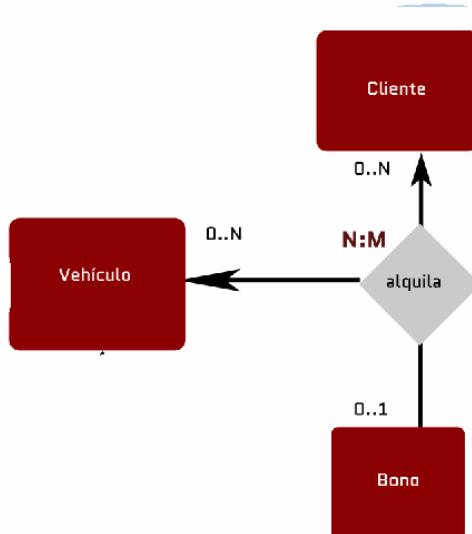
Etapa 2.

➤ **Diseño de base de datos.**

Una vez pensado el tipo de página WEB y su estilo. He tenido que trabajar una posible base de datos en modo borrador. Pensando las entidades esenciales, sus atributos, sus posibles claves primarias, estudio de fechas necesarias, no menos importante la posibilidad de entidades débiles en identidad o en existencia.

➤ **Problemas.**

Algunos de los problemas apreciados respecto a la base de datos han sido bien distintos. De inicio, los bonos de ofertas los relacionaba con vehículo, cosa errónea, ya que un bono o descuento debería repercutir en la nueva tabla creada "Alquiler", una N-Aria **1:N:Z** [ternaria].



Una contraseña para el cliente, al principio realicé la base de datos sin este esencial atributo para que se logre un cliente y así acceder a su área privada.

Propiedad color para cada vehículo, ya que podría ser que no se introduzca una imagen si no se tuviera pero así al menos sabremos identificarlo por color, esta dato lo conseguiremos en una paleta de colores que obtendremos en nuestro formulario de crear vehículo.

Hacer como clave primaria, el grupo de la categoría en la tabla "Categoría" y quitar un id autoincrementable, los vehículos disponen de una única categoría para clasificarlos y como mucho una variante con un número después de su letra.

Crear un apodo o nick para los empleados ya que su nombre podría repetirse en otros empleados, el apodo sería irrepetible aún teniendo el mismo nombre de pila del empleado concreto. Eso sí, los empleados pueden tener una misma clave de acceso ya que no habría ningún problema.

También es cierto que según me han salido problemas, he logrado solucionarlos rápidamente puesto que tenía bien claro como sería mi base de datos o los atributos que quería tener para desplegar la información.

Etapa 3.

➤ **Creación por comando en mySQL de la base de datos.**

He creado una a una cada tabla con sus respectivas restricciones, asignando claves primarias, foráneas y otros tipos de datos necesarios.

➤ **Problema.**

Distinta sintaxis entre SQL y mySQL, he tenido que modificar especificaciones en los comandos para crear las tablas a crear, así como la introducción de autoincrementables a los campos que creía oportuno.

Etapa 4.

➤ **Aclarar y especificar esquema MVC de CodeIgniter.**

Diseño y creación de modelos, vistas y controladores. Debidamente cargaríamos y configuraríamos (autoconfig) CodeIgniter para que cargue

y acceda a nuestra base de datos, asignaríamos un controlador por defecto y definiremos un `base_url` para movernos de manera correcta.

He decidido crear mis propias librerías para cargar en las cabeceras otras librerías necesarias para desarrollar mi proyecto. Tales como JQUERY, mi propia hoja de estilo, estilos de Bootstrap, JavaScript de Bootstrap, mi Slider para el banner de publicidad, mi favicon ... etc

```

EXPLORADOR          hp   vPie.php    vIndexPrincipal.php    vCrearVehiculo.php    mCategoriaVehiculoBase.php    Abrir_Cerrar_HTML.php ×    🔍    ⌂    ...
EDItores Abiertos
  miJS.js application\miJS
  ✘ vCabecera.php application\views\index
  vPie.php application\views\index
  vIndexPrincipal.php application\views\index
  vCrearVehiculo.php application\views\mantenimiento...
  mCategoriaVehiculoBase.php application\models
  Abrir_Cerrar_HTML.php application\libraries
  ✘ PI
    > config
    > controllers
      cAlquilerBono.php
      cCategoriaVehiculoBase.php
      cCliente.php
      cIndex.php
      cOtrasPaginas.php
      cTipoEmpleado.php
      index.html
      Welcome.php
    > core
    > helpers
    > hooks
    > language
      > english
        index.html
    > libraries
      Abrir_Cerrar_HTML.php
        index.html
    > logs
    > models

1  <?php
2  /*
3  * MI METODO QUE LO CARGO, ANTES CON EL REQUIRE_ONCE, AL PONERLO EN LIBRERIAS, ME CARGA SOLO,
4  * ME VIENE DE LUJO, ES PARA Abrir o Cerrar el HTML (PASANDOLE 1 TITULO COMO PARAMETRO)
5  */
6  class Abrir_Cerrar_HTML
7  {
8      public function Abrir ($titulo)
9      {
10         >>
11             <!DOCTYPE html> <!-- CARGO TODAS MIS LIBRERIAS -->
12             <html>
13                 <head>
14                     <title><?php echo $titulo ?></title>
15                     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
16                     <!--Lo de BS-->
17                     <link href=<?php echo base_url ('application/_libBS/css/bootstrap.min.css')?>" rel="stylesheet" type="text/css">
18                     <link href=<?php echo base_url ('application/_libBS/css/bootstrap-theme.min.css')?>" rel="stylesheet" type="text/css">
19                     <link href=<?php echo base_url ('application/_libOC/OwlCarousel/dist/assets/owl.carousel.css')?>" rel="stylesheet" type="text/css">
20                     <link href=<?php echo base_url ('application/_libOC/OwlCarousel/dist/assets/owl.theme.css')?>" rel="stylesheet" type="text/css">
21                     <script type="text/javascript" src=<?php echo base_url ('application/_libJS/jquery-3.2.1.min.js')?>">
22                     <script language="javascript" src=<?php echo base_url ('application/_libBS/js/bootstrap.min.js')?>">
23                     <script type="text/javascript" src=<?php echo base_url ('application/_libOC/OwlCarousel/owl.carousel.js')?>">
24                     <link href=<?php echo base_url ('application/_miCSS/miCSS.css')?>" rel='stylesheet' type='text/css'>
25                     <script type="text/javascript" src=<?php echo base_url ('application/_miJS/miJS.js')?>">
26                     <link rel="shortcut icon" href=<?php echo base_url ('application/_imagenesCSS/favicon.png')?>">
27                 </head>
28                 <body>
29                     <?php
30             }
31             public function Cerrar ()
32             {
33                 echo "</body></html>";
34             }
35         }

```

He diseñado y desarrollado un loguin en PHP para entrar a un índice en el cual nos aparecerá una especie de menú donde nos podremos digerir al apartado que deseemos.

Creamos una cabecera y un cuerpo que es lo que cambia en cada vista, a mi modo de ver, el pie lo veía innecesario aunque podría insertarlo cuando se guste. La variable de sesión siempre estará en la cabecera (común en todo el mantenimiento).

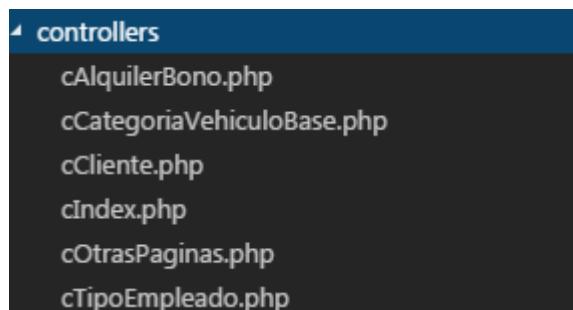
➤ Problema.

De primeras organizaba mal mis controladores y mis modelos ya que los agrupaba por cargar, modificar, borrar...etc. Este error no me fue corregido con el proyecto de clase de los muebles ni yo visualizarlo, foros y compañeros de prácticas me dieron su punto de vista para mejorarlo y optimizarlo.

Comprendí que lo idóneo era ordenarlo según la base de datos y clases, es decir, controlador cliente, controlador tipo y empleado, controlador categoría y vehículo, modelo cliente, modelo tipo y empleado, modelo categoría y vehículo... y así sucesivamente.

Con esto, si el día de mañana cambiásemos vehículos por barcos accederíamos a sus clases y no tendríamos que ir buscando funciones sueltas por cada controlador o cada modelo.

Este ha sido uno de mis mayores problemas pero satisfactoriamente he corregido. Me ha llevaba tiempo adaptarlo de nuevo cuando ya tenía dos tercios de mi proyecto avanzado.



Etapa 5.

➤ Creación del mantenimiento, accesible con un logueo.

He diseñado y desarrollado un login en PHP para entrar a un índice en el cual nos aparecerá una especie de menú donde nos podremos digerir al apartado que deseemos.

Creamos una cabecera y un cuerpo que es lo que cambia en cada vista, a mi modo de ver, el pie lo veía innecesario aunque podría insertarlo cuando se guste. La variable de sesión siempre estará en la cabecera (común en todo el mantenimiento).

➤ Problema.

Ninguno.



Etapa 6.

- Creación de páginas / vistas de mantenimiento y validación de formularios así como tablas para visualizar datos.

He elaborado vistas, algunas contenían formularios necesarios para el acceso al modelo y así lograr tocar nuestra base de datos. Es el punto que más tiempo me ha llevado elaborar. Contra más tablas, más mantenimiento y formularios necesitan.

Nuevo vehículo

Matrícula:

Marca:

Modelo:

Kilómetros:

Potencia (CV):

Combustible:

Diésel Gasolina Híbrido Eléctrico

Base:

Categoría:

Color:

Imagen:

Ejemplo de matrícula: 2399JLK
De 1 a 30 caracteres en marca y modelo.

Añadir

Volver

- Problema.

El primer error fue la no encriptación ya que se necesitaba la secuencia "enctype='multipart/form-data'".

Otro fallo, al no introducir imagen ya que siempre intentaba sacar la otra incluso cuando no existía POST de la imagen y tratar esto, el programa fallaba. Logré solucionarlo mediante if anidados para que no pasara por tal secuencia si no se introducía foto ya que diseñe la base de datos para que no fuese nula. No fue fácil solucionarlo.

```
public function fCrearVehiculo ($imagenPasar)
{
    $consultaSiExiste = $this->db->get_where('Vehiculo', array ('Matricula' => $this->input->post("Matricula")));
    if ($consultaSiExiste->num_rows() > 0)
    {
        echo "<div id='cajaCabecera' class='entradaAnadida alert alert-danger'>Upps!! Ya existe dicho vehículo con matrícula < ".$this->input->post("Matricula")."</div>";
    }
    else
    {
        if ($imagenPasar['image_type'] == "") // esto se lo he puesto para q no salte el error cnd no quiera meter foto todavía
        {
            $insertarImagen="";
        }
        else
        {
            $insertarImagen=file_get_contents($imagenPasar['full_path']); // AHORA SÍ, LA IMAGEN EN SI PARA DARLA AL INSERT
        }
        $data=array("Matricula"=>$this->input->post("Matricula"),
                   "Marca"=>$this->input->post("Marca"),
                   "Modelo"=>$this->input->post("Modelo"),
                   "Km"=>$this->input->post("Km"),
                   "Combustible"=>$this->input->post("Combustible"),
                   "Potencia"=>$this->input->post("Potencia"),
                   "Imagen"=>$insertarImagen, // OOO!! NO INPUT, YA QUE YA ESTÁ EN LA VARIABLE
                   "FechaExpiracion"=>date("Y-m-d", strtotime("+2 years")), // para que sume siempre dos años a la fecha actual, el alquiler expira
                                                               // date_add(sysdate(), INTERVAL 2 year) // la buena es la de arriba
                                                               // sysdate() + interval '2' year // la buena es la de arriba
                   "Estado"=>"Disponible",
                   "Color"=>$this->input->post("Color"),
                   "CodigoBase"=>$this->input->post("CodigoBase"),
                   "GrupoCategoria"=>$this->input->post("GrupoCategoria")
        );
    }
}
```

Etapa 7.

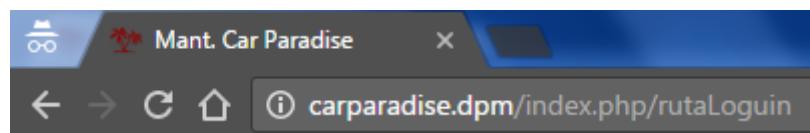
- **Elaboración de rutas con CodeIgniter.**

Algunos enlaces apuntaban a un mismo punto, por este motivo me ha visto obligado a elaborar rutas donde especifica su controlador, su función y, si fuese necesario, un parámetro.

```

52     $route['default_controller'] = 'cIndex/fIndex';
53     $route['404_override'] = '';
54     $route['translate_uri_dashes'] = FALSE;
55
56     $route['rutaIndice'] = 'cIndex/fIndice/Indice';
57     $route['rutaLoguin'] = 'cIndex/fLoguin';
58     $route['rutaIndex'] = 'cIndex/fIndex';

```



- **Problema.**

Ninguno.

Etapa 8.

- **Paginación de listados de vehículos y alquileres en CodeIgniter.**

Crear una paginación de los trs de ambas tablas en el mantenimiento para la mejora en la visualización que realice el empleado (de cualquier autorización), para ello he necesitado información del tutorial de CodeIgniter y un video explicativo. Para ello he elegido un número de cantidad de registros que deseaba que apareciesen y del mismo modo un bloque de paginación que más tarde pasaré a la vista correspondiente y que luego mostraré con un 'echo' con PHP.

Nuevo vehículo											
Matrícula	Marca	Modelo	Kilómetros	Combustible	Potencia	Fch. Exp.	Base	Estado	Alquilar	Modificar	Borrar
8115JVB	Lamborghini	Veneno	6664	Gasolina	750 CV	03/11/19	46500	Disponible			
5902JMC	Mercedes	Clase A	9982	Diésel	150 CV	15/05/19	41007	Disponible			
2409JGL	Mercedes	GLB	4424	Diésel	160 CV	15/05/19	8014	Disponible			
4409JML	Mitsubishi	Lancer	9883	Diésel	170 CV	15/05/19	15007	Disponible			
1193JNN	Nissan	Leaf	4400	Eléctrico	80 CV	01/05/19	28071	Reservado			
8892JCC	Opel	Vectra	2211	Diésel	90 CV	13/05/19	30591	Taller			
3231JPP	Peugeot	308	220	Diésel	90 CV	13/08/18	15007	Disponible			
8921JFL	Peugeot	Partner	3332	Eléctrico	90 CV	13/05/19	8014	Disponible			

➤ **Problema.**

Ninguno.

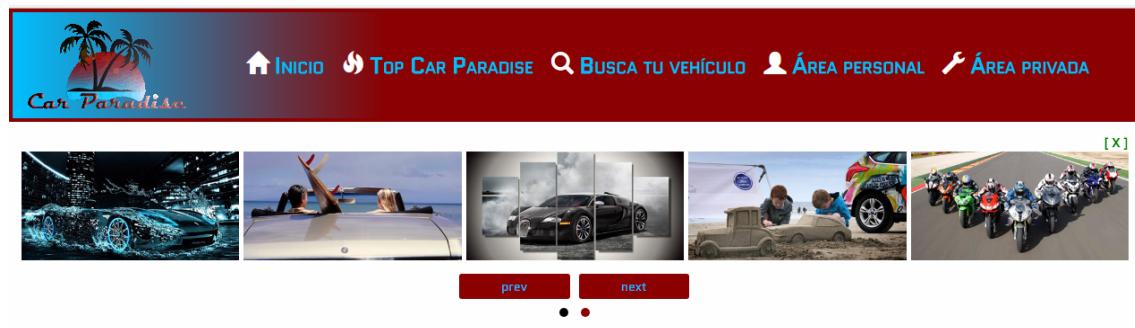
Etapa 9.

➤ **Creación del index [carparadise.dpm] de mi WEB.**

○ **Creación de la cabecera de página.**

La cabecera dispone del logo de la empresa y un menú [nav] con diferentes enlaces a otras páginas. En el podemos mostrar / ocultar la sesión iniciada por un cliente, su historial si iniciará sesión y los datos de dicho cliente logueado que más tarde se le ofrece la posibilidad de cerrar tal sesión.

La cabecera contiene de un banner de publicidad [JavaScript] que podremos cerrar con un click. Dicho banner lo hemos creado con la librería especial llamada OwlCarousel que cargamos y le daremos propiedades, funcionalidad y visibilidad con JQUERY.



○ **Creación del cuerpo de la página.**

Varía según pinchemos alguna categoría que mostraremos en otro menú en la parte izquierda y que contiene las clasificaciones de nuestros vehículos. Además, en el cuerpo encontramos dos grandes bloques:

- **Novedades**, que limitaremos en las 2 entradas más recientes a nuestra base de datos sin limitar el estado del vehículo ya que no sería reservable.
- **Disponibles recomendados**, que limitaremos en 12 ordenados según potencia de los que evidentemente solo podremos ver los disponibles, ya que si son reservables, estos aparecerán de forma aleatoria según recarguemos la página no navegemos.

The screenshot shows a sidebar with vehicle categories: Camión, Caravana, Deportivo, Descapotable, Furgoneta, Limusina, Minibús, Monovolumen, Motocicleta, and Turismo L. On the right, there's a section titled 'ÚLTIMAS NOVEDADES' with two cars: a blue Ferrari Spider (RESERVADO) and a red Lamborghini Huracán (DISPONIBLE). Below this is a section titled 'VEHÍCULOS DISPONIBLES RECOMENDADOS'.

- **Creación del pie de página.**

El pie lo componen los logotipos de nuestras marcas de vehículos con más auge y un subpie donde mostraremos información de la empresa y enlaces concretos.



➤ **Problema.**

Organización de los logotipos del pie, cuestión de si introducirlos como img o div con background-image.

Etapa 10.

➤ **Creación de otras páginas [si cuerpo sin cabecera ni pie de página].**

- **Búsqueda de vehículo:** Dispondremos de gran cantidad de criterios para filtrar, ya que tiraremos de AJAX para pasar información de los campos recogidos que enviaremos y así traernos la información de los vehículos por medio de consultas muy concretas al realizar POST. Al realizar la búsqueda, ofreceremos múltiples mensajes de información del filtro como numero de registros, mensajes de no encontrado. Nota, al realizar una búsqueda satisfactoria, el banner de publicidad se cerrará de forma automática.
- **Reservar:** Es nuestra ficha de reserva con datos, imágenes y videos [si existiera] y una pasarela de pago donde podremos elegir fechas. Puede funcionar tanto sin sesión [reserva online]

introduciendo un número perfectamente válido de DNI) o con sesión [reserva online de cliente existe, el DNI es fijo].

- **Top Car Paradise:** Son listados de información variada, limitados en 10 registros y ordenados por cantidad por medio de JOINS especializados. Cada grupo se muestra en tablas debidamente moldeadas y parece un ícono donde podremos pinchar para descargar los listados de los grupos de meses anteriores.
- **Mi historial:** No aparecerá de inicio ya para eso un cliente debería de realizar su logueo, una vez esto, aparecerá dicho enlace en nuestro menú. En este apartado podremos ver un acordeón debidamente ordenado fecha más actual a menos, se abre al cliquear y donde podremos ver la información y trayectoria de alquileres / reservas en nuestra empresa

➤ Problema.

El primer problema fue al querer traer la imagen del vehículo en un JSON en búsqueda de vehículos, por su codificación. Error la introducir * en el SELECT en el grupo de VEHÍCULO del JOIN.

El segundo, los detalles de la reserva realizada con éxito solo me aparecían al hacer clic dos veces y no una. Problema de JavaScript que me ha perturbado gravemente pero que conseguí solucionar al encontrar el problema al ejecutar la función erróneamente ya no se cargaba de inicio pero si al pinchar el botón reservar.

Etapa 11.

➤ Accesibilidad y adaptabilidad a otras pantallas en los medios como pantallas xl, tablet o smartphone.

Oculto menú o modiflico propiedades según el tamaño de la pantalla, reduzco su tamaño o cierta información la oculto de forma permanente.

Con algunos estilos específicos de clase consigo que ciertos elementos sean responsivos.

```

602 ****PARA HACERLO RESPONSIVO*****
603 ****PARA HACERLO RESPONSIVO*****
604 @media all and (max-width: 1150px){
605   div#menuCategorias{
606     height: 300px;
607   }
608 }
609 @media all and (max-width: 750px){
610   div.cajaCarParadise{
611     display:none; /* para q desaparezca */
612   }
613   div.cajaLogoGif {
614     display:none; /* para q desaparezca */
615   }
616   nav ul li{
617     font-size: 80%;
618     margin: 0;
619   }
620   div#menuCategorias {
621     width: 100%;
622     text-align: center;
623   }
624   div#novedades {
625     width: 100%;
626   }
627 }
628 div#inner-header2 {
629   display: none;
630 }
631 div#principal , section#seccion {
632   margin-top: 9.2em;
633 }

```

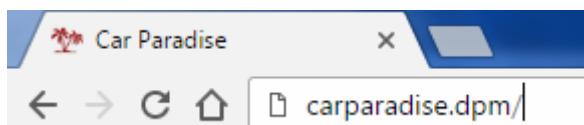
➤ Problema.

Necesario estudio y conocimiento de todas las clases de Bootstrap y un futuro menú deslizante que se oculte de manera automática. Idea de futuros submenús.

Etapa 12.

➤ Creación de un Virtual Host para un dominio concreto.

Genero un Virtual Host para trabajar en el dominio **carparadise.dpm** que apunte a nuestro localhost (temporal), con esto también consigo ocultar nuestro alias de WampServer.



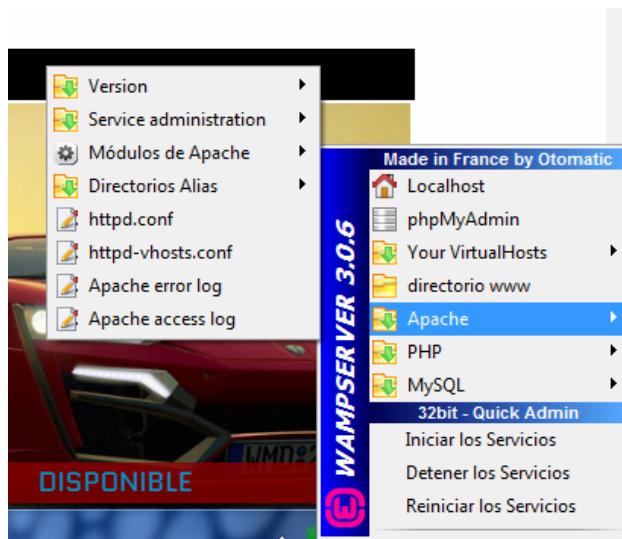
➤ Problema.

Con las rutas relativas y globales, tanto en JavaScript y en PHP [en algún submit].

Capítulo 5:

Requisitos de la aplicación.

- Servidor WEB con soporte de PHP [PHP 5.x]. Por ejemplo: Apache.
- Servidor MySQL / MariaBD funcionando con MyISAM.
- Navegador moderno con soporte de CSS 2 / 3, HTML 5 y JavaScript activado para AJAX. Por ejemplo: Google Chrome, Internet Explorer, Mozilla Firefox u Opera.



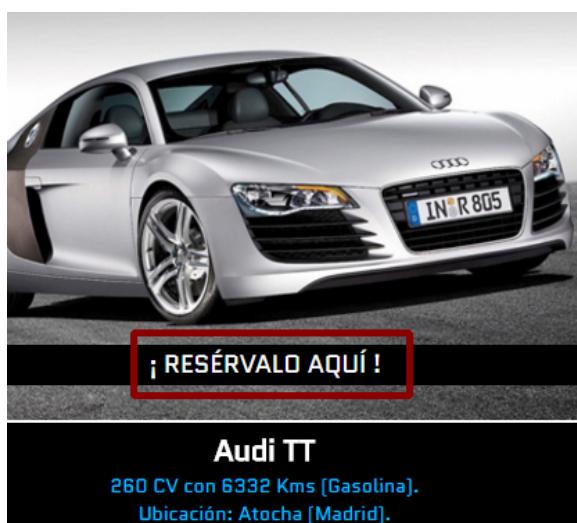
Capítulo 6:

Manual de usuario.

- La página WEB es intuitiva y responsive. Cada error de usuario será notificado y guiado.
- El usuario podrá visualizar todas las páginas si ha iniciado correctamente una sesión de cliente y alguna página menos si es un usuario no cliente. Por ejemplo: no verá su historial de alquileres / reservas. **ÁREA PERSONAL**.
- El empleado podrá entrar a la administración de mantenimiento iniciando sesión de empleado. Se limita el acceso a los empleados que no disponga de máxima autorización, es decir, un empleado de

prácticas solo podrá ver el listado alquileres mientras que un empleado indefinido se le permite ver y tocar todo lo que desee dentro del mantenimiento. De esta manera los empleados puede crear, modificar o eliminar información que pertenecen a la base de datos llamada carparadise_bd. **ÁREA PRIVADA.**

- Puede consultar una amplia clasificación de grupos interesantes para el usuario. De igual manera pede descargarse el listado top del mes anterior. **TOP CAR PARADISE.**
- El usuario / cliente registrado podrá reservar su vehículo pinchando varios enlaces que ofreceremos en la página WEB. Introduciendo correctamente fechas o DNI (si no es cliente) en la ficha podrá reservar automáticamente, recibiendo un número de referencia de reserva. **INICIO** y **BUSCA TU VEHÍCULO.**



Audi TT

260 CV con 6332 Kms [Gasolina].
Ubicación: Atocha [Madrid].

Color	Categoría	Matrícula	Marca	Modelo	Km	Combustible	Base	Ubicación	Estado	Reservar
Deportivo	3329JJP	Audi	TT	6332	Gasolina	Atocha	Madrid	Disponible		

FICHA DE RESERVA [3329JJP]

Datos del vehículo a reservar.

- ★ Matrícula: 3329JJP.
- ★ Marca: Audi.
- ★ Modelo: TT.
- ★ Kilómetros: 6332.
- ★ Combustible: Gasolina.
- ★ Potencia: 260 CV.
- ★ Código base: 26012.
- ★ Ubicación: Madrid.
- ★ Grupo de categorías: Y.
- ★ Estado actual: Disponible.
- ★ Color:
- ★ Observaciones: Reservable hasta el 13/02/18.

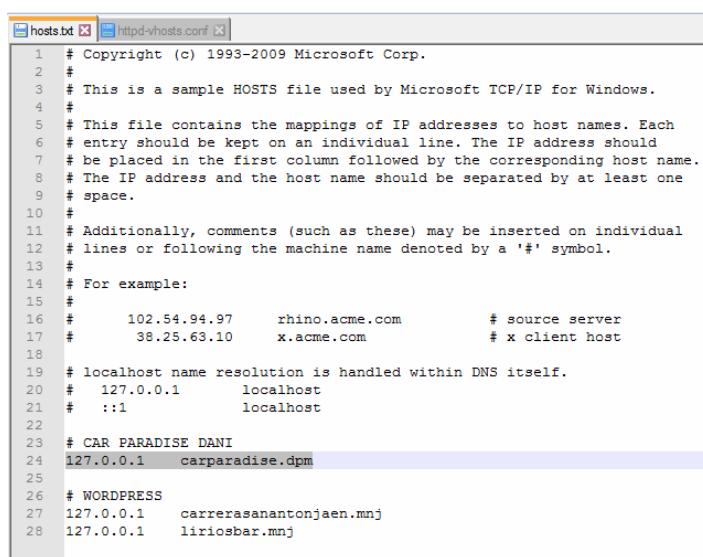
Datos necesarios para la reserva.

Fecha inicio:	Fecha fin:	Referencia online:
<input type="text" value="16/06/2017"/>	<input type="text" value="16/06/2017"/>	<input type="text" value="DNI del cliente"/>
Precio base (€):	Descuento online (%):	Precio final (€):
<input type="text"/>	<input type="text" value="15"/>	<input type="text"/>
<input type="button" value="Reservar"/>		

Capítulo 7:

Manual de instalación.

- Instalación de WampServer y sus packs necesarios de Microsoft Visual C ++ siguiendo todos sus pasos de forma intuitiva, es decir, eligiendo las opciones a convenir.
- Configurar host apuntando a 127.0.0.1 con el dominio carparadise.dpm.

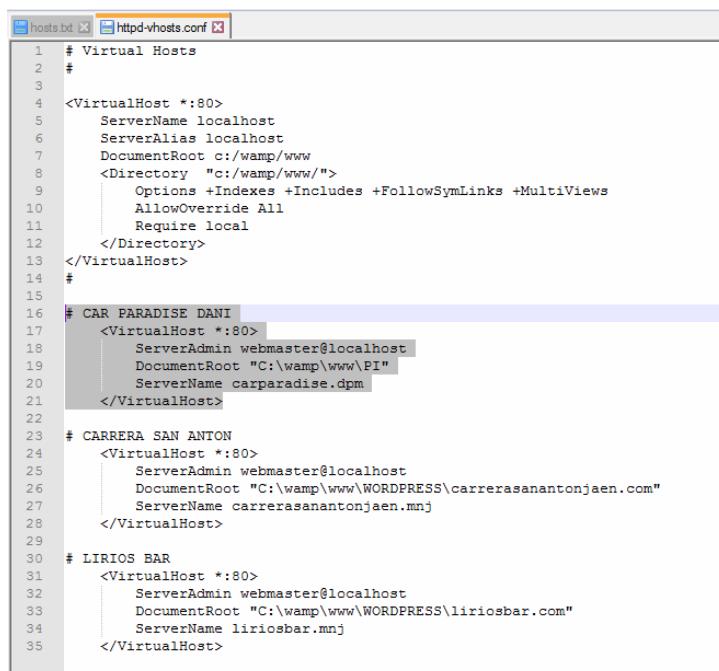


```

hosts.txt
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #      102.54.94.97    rhino.acme.com        # source server
17 #              38.25.63.10    x.acme.com          # x client host
18 #
19 # localhost name resolution is handled within DNS itself.
20 #       127.0.0.1    localhost
21 #           ::1         localhost
22 #
23 # CAR PARADISE DANI
24 127.0.0.1    carparadise.dpm
25 #
26 # WORDPRESS
27 127.0.0.1    carrerasanantonjaen.mnj
28 127.0.0.1    liriosbar.mnj

```

- Configurar host.

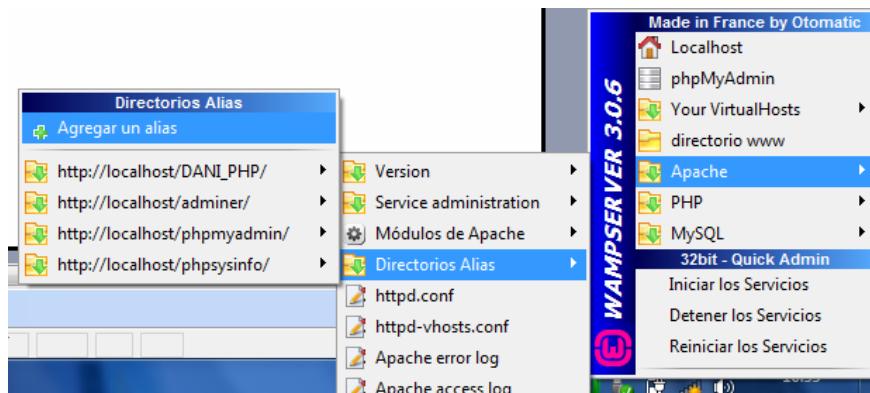


```

httpd-vhosts.conf
1 # Virtual Hosts
2 #
3 #
4 <VirtualHost *:80>
5     ServerName localhost
6     ServerAlias localhost
7     DocumentRoot c:/wamp/www
8     <Directory "c:/wamp/www/">
9         Options +Indexes +Includes +FollowSymLinks +MultiViews
10        AllowOverride All
11        Require local
12     </Directory>
13 </VirtualHost>
14 #
15 #
16 # CAR PARADISE DANI
17 <VirtualHost *:80>
18     ServerAdmin webmaster@localhost
19     DocumentRoot "C:\wamp\www\PI"
20     ServerName carparadise.dpm
21 </VirtualHost>
22 #
23 # CARRERA SAN ANTON
24 <VirtualHost *:80>
25     ServerAdmin webmaster@localhost
26     DocumentRoot "C:\wamp\www\WORDPRESS\carrerasanantonjaen.com"
27     ServerName carrerasanantonjaen.mnj
28 </VirtualHost>
29 #
30 # LIRIOS BAR
31 <VirtualHost *:80>
32     ServerAdmin webmaster@localhost
33     DocumentRoot "C:\wamp\www\WORDPRESS\liriosbar.com"
34     ServerName liriosbar.mnj
35 </VirtualHost>

```

- Configurar alias de Apache si fuese necesario y depuración si queremos activar xDebug xRemote.



```
zend_extension = "c:/wamp/bin/php/php7.0.10/zend_ext/php_xdebug-2.4.1-7.0-vc14.dll"

xdebug.remote_enable = On
xdebug.remote_autostart = 1
xdebug.remote_host = 127.0.0.1
xdebug.profiler_enable = Off
xdebug.profiler_enable_trigger = Off
xdebug.profiler_output_name = cachegrind.out.%t.%p
xdebug.profiler_output_dir ="c:/wamp/tmp"
xdebug.show_local_vars=0
xdebug.remote_log = "P:\DWE Servidor\Espacio trabajo PHP"
```

Capítulo 8: Código fuente.

- El código fuente lo entrego en un CD-ROM como se pide, dicho proyecto (PI) se vuelca donde apunte el servidor, por defecto en www, aunque es posible configurar uno o varios alias y activarlo con WampServer.
- Dicho CD-ROM dispone de todo el material necesario: videos, fotos, base de datos exportada o en carpeta intacta para ubicarla en data dentro de la instalación de mySQL, código fuente, fuente o tipografía de letra, librerías externas y propias...etc.

Capítulo 9: Presentación para la defensa de la aplicación.

- Realizado de forma específica y resumida en un PowerPoint en la ubicación de la aplicación: `/_documentacion/carParadise.ppt`.

Proyecto integrado de desarrollo de aplicaciones WEB “Car Paradise” en IES Las Fuentezuelas (Jaén) ha sido diseñado, programado, desarrollado, documentado y encuadrernado por el alumno **Daniel Pancorbo Miranda** con DNI 77345252-R en la promoción 2015 – 2017.
