

# PRÁCTICA COLABORATIVA

**Enunciado: Ejercicios UT5\_2 – Aplicación funcional en PHP. Proyecto colaborativo.**

## Indice

- 1) Introducción
- 2) Requisitos
  - a) Obligatorios
  - b) Opcionales
- 3) How to do
- 4) Pruebas
- 5) Validación de código
- 6) Notas

### 1) Introducción

Nuestra aplicación consiste en un gestor de un pequeño concesionario. Los usuarios siempre van a ser los vendedores del concesionario, los cuales pueden ser admins o no admins, esta cualidad vendría dada por el rol.

La base de datos de la aplicación sería tal que así:

Para el desarrollo de la aplicación hemos usado NetBeans como IDE.

El lenguaje de programación es PHP en su versión 8.2.

### 2) Requisitos

En este apartado vamos a explicar las funcionalidades que cumple la aplicación en cuanto al enunciado de la práctica.

#### a) Obligatorios

**LOGIN:**

Hemos implementado un Login básico en el cual, si estás registrado puedes acceder.

Nuestro login es un tanto distinto, si no tienes nuestra base de datos aparecerá un botón que activará la función crearBD(), la cual creará la base de datos, en la base de datos se insertarán varios registros creados por nosotros.

En cada archivo php que requiere una sesión activa verificamos si el usuario está registrado, en caso contrario será redirigido al login y la sesión, en caso de que exista, será cerrada.

### **GESTOR DE VERSIONES:**

Hemos usado GIT, un gestor de versiones que, aunque nos ha dado algún problema de conflicto, nos ha servido para trabajar colaborativamente y organizar bien el proyecto mediante el apartado issues y tasklists.

### **FUNCIONES CRUD:**

Hemos creado funciones CRUD

En insertar, lo primero que hago es validar los parámetros introducidos, esta es una de las capas de seguridad que tiene la aplicación:

#### **Atributos que se validan:**

DNI y contraseña para Vendedores

DNI para clientes

VIN y Matrícula para coches

Dato importante: el validador de DNI es un validador real, el DNI ha de ser un DNI de verdad, el validador de VIN solo va a necesitar un string de 17 caracteres, y el validador de contraseña va a necesitar un string con letras y números de mínimo 4 caracteres.

En ejemploVIN\_coches.txt hay ejemplos de DNI, de VIN's y de matrículas.

La matrícula se tiene que componer de cuatro números, un guión y tres letras.

A continuación adjunto una imagen de la “primera parte” de la función insertar, esta parte se encarga de validar los datos a insertar:

```
163 function insertar($tabla, $valores) {  
164  
165     $result = extraerTablas("SHOW TABLES LIKE '$tabla'");  
166  
167     if (count($result) == 1) {  
168  
169  
170         if (isset($valores['DNI'])) {  
171             if (!validarDNI($valores['DNI'])) {  
172                 throw new Exception(mensajeError("(insertar): DNI no válido."));  
173             }  
174             if (isset($valores['contrasena'])) {  
175                 if (validarContraseña($valores['contrasena'])) {  
176                     $valores['contrasena']=hash('sha256', $valores['contrasena']);  
177                 }else{  
178                     throw new Exception(mensajeError("(insertar): Error la contraseña no es válida."));  
179                 }  
180             }  
181         }else{  
182             if (isset($valores['VIN'])) {  
183                 if (!validarVIN($valores['VIN'])) {  
184                     throw new Exception(mensajeError("(insertar): VIN no válido."));  
185                 }  
186                 if (isset($valores['Matricula'])) {  
187                     if (!validarMatricula($valores['Matricula'])) {  
188                         throw new Exception(mensajeError("(insertar): Matrícula no válida."));  
189                     }  
190                 }  
191             }  
192         }  
193     }  
194 }
```

En la “segunda parte” de la función, que solo ocurre cuando los datos son válidos, se inserta el registro, en caso de que no se inserten los registros se lanzan excepciones.

```
194     try {
195         $BD = conexionPDO();
196
197         //para mostrar errores
198         $BD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
199
200         $columnasSql = implode(", ", array_keys($valores));
201         $valoresSql = ":" . implode(":", array_keys($valores));
202
203         $sql = "INSERT INTO " . $tabla . " (" . $columnasSql . ") VALUES (" . $valoresSql . ")";
204
205         //stmt se convierte en un array
206
207         $stmt = $BD->prepare($sql);
208
209         foreach ($valores as $clave => $valor) {
210             // Esto sustituye las claves por sus respectivas columnas
211             $stmt->bindValue(":" . $clave, $valor, is_int($valor) ? PDO::PARAM_INT : PDO::PARAM_STR);
212         }
213
214         if (!$stmt->execute()) {
215             throw new Exception(mensajeError("(insertar): Error en la inserción del registro.));
216         }
217     } catch (Exception $exc) {
218         echo mensajeError('No se puede insertar el registro, porque la clave principal está repetida.));
219     }
220     } else {
221         throw new Exception(mensajeError("(insertar): La tabla $tabla no existe, no es posible insertar.));
222     }
```

modificar instancia:

Compruebo si los datos a modificar son válidas, en este caso únicamente la matrícula, ya que es lo único que se puede validar, y en caso de que no se pueda modificar, devuelvo excepción.

```

224 function modificarTabla($tabla, $dato, $post) {
225     $BD = conexionPDO();
226     $arrayKeys = array_keys($post);
227
228     if (isset($post["Matricula"])) {
229         if (validarMatricula($post["Matricula"])) {
230             try {
231                 for ($i = 0; $i < count($dato); $i++) {
232                     $result = explode(" ", $post[$arrayKeys[$i + 1]]);
233                     $sql = "UPDATE $tabla SET " . $dato[$i][0] . " = '" . end($result) . "'";
234                     $stmt = $BD->prepare($sql);
235                     $stmt->execute();
236                 }
237             } catch (Exception $e) {
238                 throw new Exception(mensajeError("Algo salió mal"));
239             }
240         } else {
241             throw new Exception(mensajeError("Error matrícula errónea"));
242         }
243     } else {
244         try {
245             for ($i = 0; $i < count($dato); $i++) {
246                 $result = explode(" ", $post[$arrayKeys[$i + 1]]);
247                 $sql = "UPDATE $tabla SET " . $dato[$i][0] . " = '" . end($result) . "' WHERE";
248                 $stmt = $BD->prepare($sql);
249                 $stmt->execute();
250             }
251         } catch (Exception $e) {
252             throw new Exception('Algo salió mal!');
253         }
254     }
255 }

```

En eliminar tabla:

```

146 function eliminarTabla($tabla,$fk = null) {
147
148     $BD = conexionPDO();
149     $result = extraerTablas("SHOW TABLES LIKE '".$tabla."'");
150     if (count($result) == 1) {
151         // $BD = conexionPDO();
152         // $BD = conexionPDO();
153         if($fk!=null){
154             $sql="ALTER TABLE ".$tabla." DROP FOREIGN KEY fk_".$fk;
155             $stmt = $BD->exec($sql);
156         }
157         $sql = "DROP TABLE " . $tabla . ";";
158         $stmt = $BD->exec($sql);
159     }
160 }

```

**PREPARE:** usamos prepare un total de cinco veces

En

insertar:

```
163 function insertar($tabla, $valores) {
164     try {
165         $BD = conexionPDO();
166
167         //para mostrar errores
168         $BD->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
169
170         $columnasSql = implode(" ", array_keys($valores));
171         $valoresSql = ":" . implode(" :", array_keys($valores));
172
173         $sql = "INSERT INTO " . $tabla . " (" . $columnasSql . ") VALUES (" . $valoresSql . ")";
174
175         //stmt se convierte en un array
176
177         $stmt = $BD->prepare($sql);
178     }
```

En extraerTablas:

```
9  function extraerTablas($sql) {
10     try {
11         $BD = conexionPDO();
12         $cursorSql = $BD->prepare($sql);
13         if ($cursorSql->execute()) {
14             $tabla = $cursorSql->fetchAll();
15             return $tabla;
16         } else {
17             echo "Error en la consulta: " . $BD->error;
18         }
19     } catch (Exception $exc) {
20         mensajeError("Error general ");
21     }
22 }
```

En anadirForanea:

```
00 function anadirForanea($tabla,$foranea,$tablaForanea){
01     $BD = conexionPDO();
02     $sql = "ALTER TABLE $tabla
03         ADD COLUMN ".$foranea."_".$tablaForanea." varchar(20) NOT NULL,
04         ADD CONSTRAINT fk_".$foranea."_".$tablaForanea." FOREIGN KEY ( ".$foranea."_".$tablaForanea." )
05         REFERENCES $tablaForanea ($foranea)";
06
07     $stmt = $BD->prepare($sql);
08     $stmt->execute();
09
10 }
```

En modificarTabla:

### En eliminarDatos:

```
260 function eliminarDatos($tabla,$dato,$valor){
261     $BD = conexionPDO();
262     try {
263
264         $sql = "DELETE FROM $tabla WHERE $dato = '".$valor."'";
265         $stmt = $BD->prepare($sql);
266
267         $stmt->execute();
268
269         echo "<nav class='navbar bg-body-tertiary bg-success rounded m-2'>
270             <div class='container-fluid'>
271                 <p>
272                     'Se ha eliminado la columna correctamente '
273                 </p>
274             </div>
275         </nav>";
276     } catch (Exception ) {
277         echo "<nav class='navbar bg-body-tertiary bg-danger rounded m-2'>
278             <div class='container-fluid'>
279                 <p>
280                     'No se puede eliminar este valor, porque es foranea de otra.'
281                 </p>
282             </div>
283         </nav>";
284     }
285 }
```

### Cookies:

#### Usamos dos cookies,

```
setcookie("nombreSesion", $session["name"] . " " . $session["apellidos"], time() + 300, 'localhost')
setcookie("ultCone", $fechaActualString, time() + 300, 'localhost');
```

Una lleva como valor el nombre completo del vendedor propietario de la sesión y la otra lleva el valor de la última conexión.

La vida de las cookies es de 5 minutos.

### Estructura del código:

La estructura del código contiene funciones, estructuras de control y excepciones.

**OPCIONALES:** Hemos hecho todas las opcionales menos el log.

### **ROLES:**

Los vendedores pueden ser admin o cualquier otro rol, este atributo se llama 'rol'.

Los admin pueden acceder a gestor empleados y gestor clientes:

```
if ($_SESSION['rol'] !== 'admin') {  
    header("Location: homepage.php");  
    exit(); // Ensure that no further code is executed after the redirect  
}
```

En el menú de inicio pueden ver ambas páginas

```
45         if ($_SESSION['rol'] == 'admin'){  
46             echo'  
47                 <li class="nav-item m-1">  
48                     <a class="nav-link" href="gestionClientes.php">  
49                         <i class="fa-solid fa-children mx-2"></i>Gestion de Clientes  
50                     </a>  
51                 </li>  
52                 <li class="nav-item m-1">  
53                     <a class="nav-link" href="gestionEmpleados.php">  
54                         <i class="fa-solid fa-user-nurse mx-2"></i>Gestion de Empleados  
55                     </a>  
56                 </li>  
57             ';  
58         }
```



## CREACIÓN DE BASE DE DATOS MEDIANTE CÓDIGO PHP:

```
61 function crearBD() {
62
63
64
65
66
67     $BD = new PDO("mysql:host=localhost", 'root', '');
68     try {
69         //En insertar la letra ñ da error (puede ser la función bindValues)
70         $BD->exec('CREATE DATABASE concesionario');
71
72
73         crearTabla("vendedores", array("DNI" => "varchar(20)", "Nombre" => "varchar(20)", "Apellidos" =>
74         "RoI" => "varchar(20)", "contrasena" => "varchar(100)", 'Email' => 'varchar(100)'), array("DNI")
75
76         insertar("vendedores", array("DNI" => "06293364H", "Nombre" => "Javier", "Apellidos" => "Diaz",
77         insertar("vendedores", array("DNI" => "03245754K", "Nombre" => "Victor", "Apellidos" => "Valdes",
78         insertar("vendedores", array("DNI" => "03245755K", "Nombre" => "VictorNoAdmin", "Apellidos" => "
79
80         crearTabla("coches", array("VIN" => "varchar(20)", "Matricula" => "varchar(20)", "Marca" => "var
81         insertar("coches", array("VIN" => "JH4DC4400SS012345", "Matricula" => "3467-LKF", "Marca" => "Fc
82         insertar("coches", array("VIN" => "KLATF08Y1VB363633", "Matricula" => "0493-HGS", "Marca" => "Fe
83
84         crearTabla("clientes", array("DNI" => "varchar(20)", "Nombre" => "varchar(20)", "Apellidos" => "
85         insertar("clientes", array("DNI" => "05245677L", "Nombre" => "Rodrigo", "Apellidos" => "Pérez",
86         insertar("clientes", array("DNI" => "12304964Y", "Nombre" => "Alejandro", "Apellidos" => "Sánchez
87
88         crearTabla('ventas', array('COD_VENTAS' => 'varchar(20)'), array('COD_VENTAS'));
89         anadirForanea('ventas', 'DNI', 'vendedores');
90         anadirForanea('ventas', 'VIN', 'coches');
91         anadirForanea('ventas', 'DNI', 'clientes');
92         insertar('ventas', array('COD_VENTAS' => '1', "DNI_vendedores" => "06293364H", "VIN_coches" => "
93         insertar('ventas', array('COD_VENTAS' => '2', "DNI_vendedores" => "03245754K", "VIN_coches" => "
94
95     } catch (Exception $exc) {
96         echo $exc->getMessage();
97     }
98 }
```

La función crearBD se encarga de ello,

Voy llamando a las funciones necesarias para crear la base de datos inicial.

### CREAR TABLA:

crearTabla(): Se encarga de crear una tabla con los campos introducidos y primary keys.

### LA SESION EXPIRA / COOKIES:

La sesión expira cuando las cookies se destruyen, es decir a los 5 minutos de no refrescar la página.

```
<?php
    include_once $_SERVER['DOCUMENT_ROOT']
    session_start();
    comprobarCookie($_SESSION,$_COOKIE);
    if ($_SESSION['rol'] !== 'admin') {
        header("Location: homepage.php");
        exit(); // Ensure that no further
    }
?>
```

Comprobar cookie se encarga de ello.

### PHPMAILER:

La aplicación, usando la librería phpMailer, envía mail a la persona que se inicia sesión.

```
$mail->Username ='concesionarioconce@gmail.com';

$mail->addAddress($_SESSION['email']);
```

Si no funciona correctamente se necesita instalar PHPMailer con composer con el comando composer que indico en el comentario php

```
/*
    Importamos las dependencias de PHPMailer
    *
    * Para haccer funcionar esto usamos composer
    * - composer require phpmailer/phpmailer
    * introducimos la carpeta 'vendor' generada en libraries
    *
    * Los archivos de composer a veces se suelen borrar, para arreglar esto simplemente sobrecribir
    * vendor soluciona el problema
    */
```

### Pruebas no Funcionales

#### Prueba de velocidad de carga:

Esta prueba es cuando inicias sesión con tu usuario y contraseña, y te metes al homepage.php

Como podemos ver tarda 2.24 segundos en cargar el homepage.php debido a que cuando pasamos del index al homepage con un usuario, a ese usuario que tiene un correo temporal, se le envía un correo avisando que se ha iniciado sesión en su cuenta.

index.php	302	docum...	Other	625 B	2.24 s	
homepage.php	200	docum...	index.php	5.6 kB	10 ms	
bootstrap.min.js	200	script	homepage.php:...	(memor...	0 ms	
jefe.jpg	200	jpeg	homepage.php:...	(disk ca...	4 ms	
bootstrap.min.css	200	styleshe...	homepage.php:6	(disk ca...	3 ms	
all.min.css	200	styleshe...	homepage.php:7	(disk ca...	2 ms	
jquery-3.5.1.slim.min.js	200	script	homepage.php:...	(disk ca...	3 ms	
fa-solid-900.woff2	200	font	all.min.css	(disk ca...	2 ms	
popper.min.js	200	script	homepage.php:...	(disk ca...	1 ms	
favicon.ico	200	x-icon	Other	(disk ca...	1 ms	
jsjs	200	script	contentjs:32	1.3 kB	5 ms	
dom.js	200	script	contentjs:32	2.0 kB	3 ms	
jsjs	200	script	contentjs:32	1.3 kB	3 ms	
dom.js	200	script	contentjs:32	2.0 kB	3 ms	

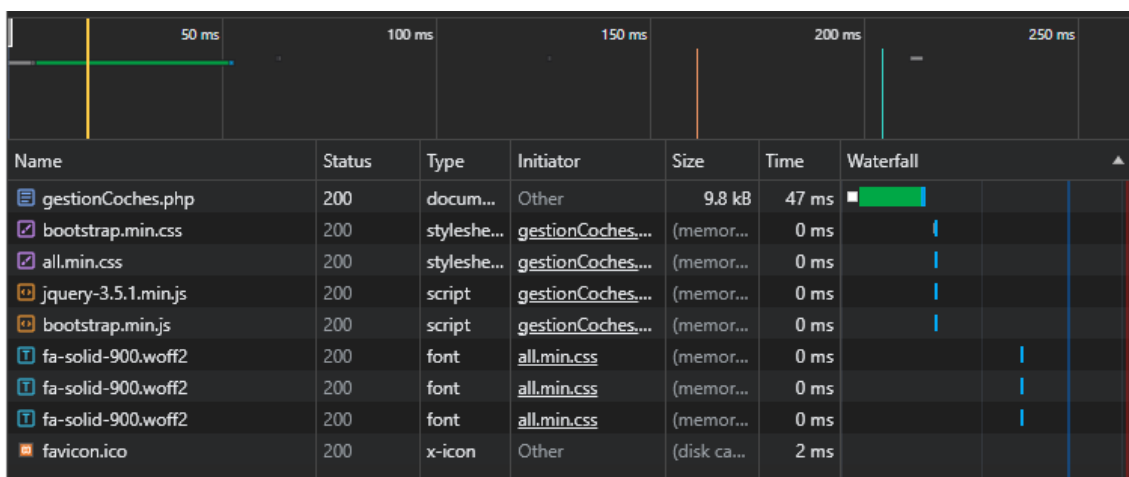
Gestión empleado:

Name	Status	Type	Initiator	Size	Time	Waterfall
gestionEmpleados.php	200	docum...	Other	11.6 kB	32 ms	
bootstrap.min.css	200	styleshe...	gestionEmplea...	(memor...	0 ms	
all.min.css	200	styleshe...	gestionEmplea...	(memor...	0 ms	
jquery-3.5.1.min.js	200	script	gestionEmplea...	(memor...	0 ms	
bootstrap.min.js	200	script	gestionEmplea...	(memor...	0 ms	
fa-solid-900.woff2	200	font	all.min.css	(memor...	0 ms	
fa-solid-900.woff2	200	font	all.min.css	(memor...	0 ms	
fa-solid-900.woff2	200	font	all.min.css	(memor...	0 ms	
favicon.ico	200	x-icon	Other	(disk ca...	1 ms	

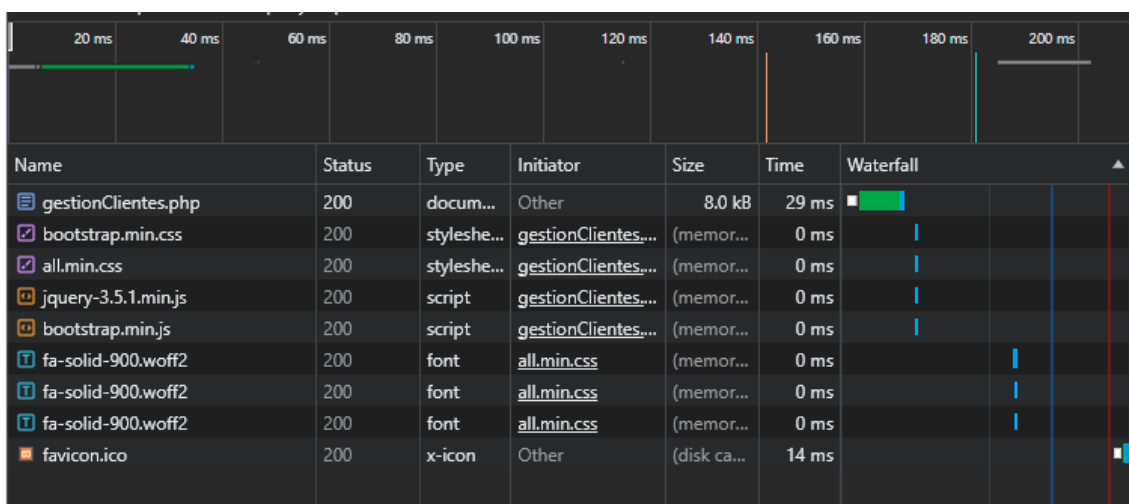
Ventas:

Name	Status	Type	Initiator	Size	Time	Waterfall
gestionVentas.php	200	docum...	Other	9.4 kB	45 ms	
bootstrap.min.css	200	styleshe...	gestionVentas.p...	(memor...	0 ms	
all.min.css	200	styleshe...	gestionVentas.p...	(memor...	0 ms	
jquery-3.5.1.min.js	200	script	gestionVentas.p...	(memor...	0 ms	
bootstrap.min.js	200	script	gestionVentas.p...	(memor...	0 ms	
fa-solid-900.woff2	200	font	all.min.css	(memor...	0 ms	
fa-solid-900.woff2	200	font	all.min.css	(memor...	0 ms	
fa-solid-900.woff2	200	font	all.min.css	(memor...	0 ms	
favicon.ico	200	x-icon	Other	(disk ca...	1 ms	

### Coches:



### Cientes:



# Fase de Pruebas

## Pruebas en Login:

Si no introducimos nada en el formulario:

Formulario no rellenado

### Inicio de sesión

Usuario

Usuario

Contraseña

Contraseña

Iniciar Sesión

Si introducimos el DNI solo, nos dice que el formulario aun no esta rellenado:

Formulario no rellenado

### Inicio de sesión

Usuario

06293364H

Contraseña

Contraseña

Iniciar Sesión

Si introducimos un DNI y una contraseña que no son correctas:

La contraseña o el usuario no es correcto

### Inicio de sesión

Usuario

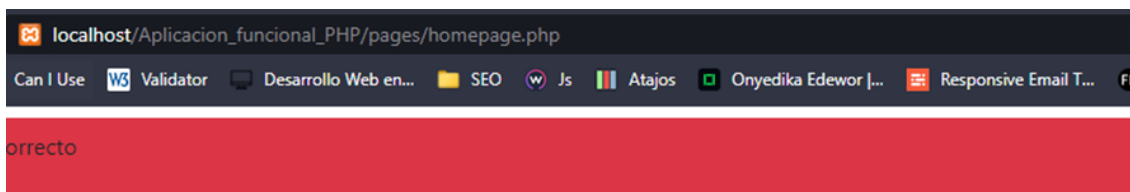
06293364H

Contraseña

Contraseña

Iniciar Sesión

Si intentamos acceder a otra página sin estar loggeado:



## Inicio de sesión



## Inicio de sesión

Usuario

Usuario

Contraseña

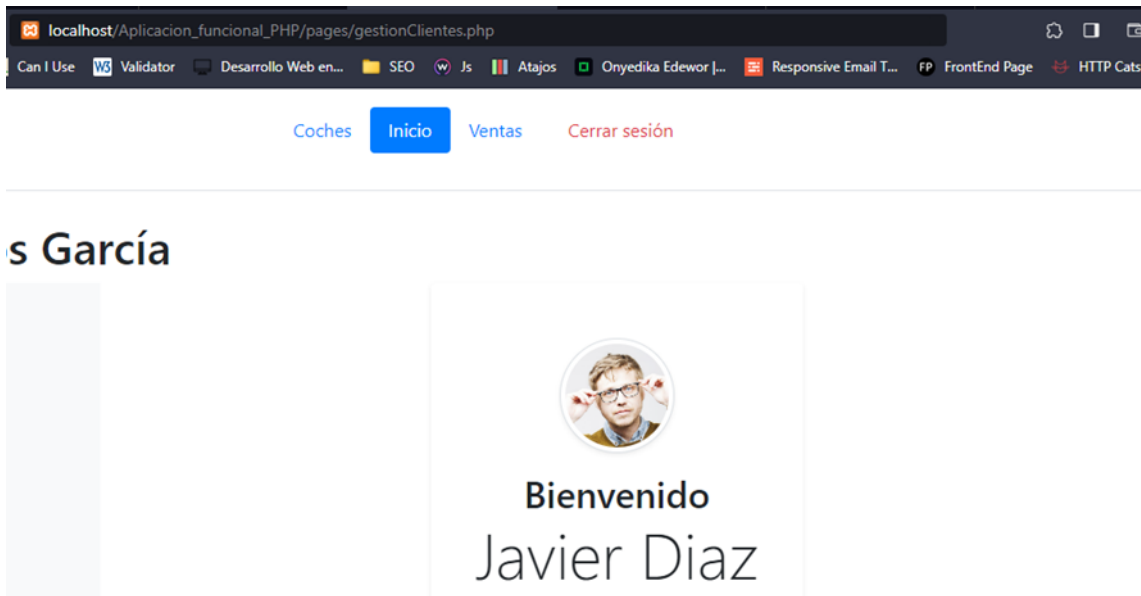
Contraseña

Iniciar Sesión

Si pasan 10 minutos desde la última conexión también nos lleva al login.

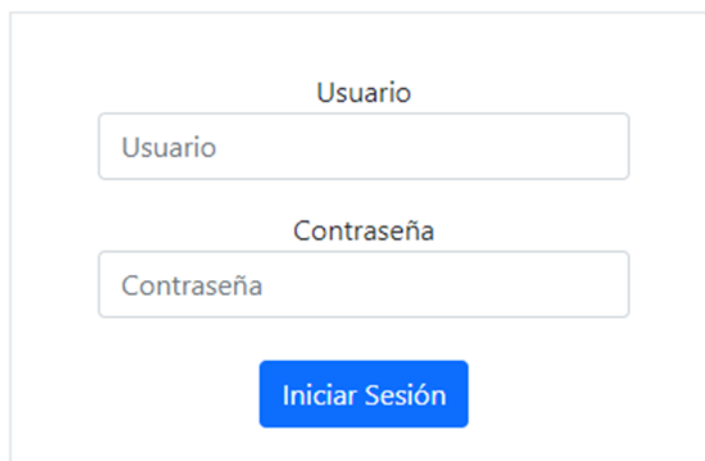
## Pruebas en homepage:

Si estamos en el rol de junior o empleado e intentamos acceder a una ruta donde no tenemos acceso nos devuelve a la misma página.



También podemos cerrar sesión y se nos irá al login.

## Inicio de sesión



The login form is a simple, clean interface with a light gray background. It contains two input fields: 'Usuario' and 'Contraseña'. The 'Usuario' field is a rounded rectangle with a light gray border and a light gray placeholder text 'Usuario'. The 'Contraseña' field is a rounded rectangle with a light gray border and a light gray placeholder text 'Contraseña'. Below the 'Contraseña' field is a blue button with white text 'Iniciar Sesión'.

## Gestión de Coches:

Si intentamos agregar un coche con el VIN y la matricula sin los parámetros correctos:

Agregar Coche

VIN

JH4DC440

Matricula

0628CHG

Marca

Toyota

Modelo

Corolla

Año

2025

Precio

25000

KM

302513

Cerrar

Guardar

Cientes

Coches

Inicio

Ventas

Empleados

Cerrar sesión

El VIN o la Matricula no tiene los valores correctos

Ha ocurrido un error inesperado al insertar los datos

### Gestión de Coches

VIN	Matricula	Marca	Modelo	Año	Precio	Km	Editar	Eliminar
JH4DC4400SS012345	5410-CLL	Ford	Fiesta	2008	2500	100001		
JH4DC4401SS012348	5248-PLP	Seat	Toledo	288	5656	65656		
KLATF08Y1VB363633	0493-HGS	Ferrarii	Roma	2018	200500	79999		



## Gestión de Clientes:

Si intentamos eliminar un cliente que tiene asignada una venta nos aparecerá el siguiente mensaje:

'No se puede eliminar este valor, primero debe borrar las instancias asociadas.'

**Cientes**

Coche




Inicio

Ventas

Empleados

Cerrar sesión

## Gestión de Clientes

DNI	Nombre	Apellidos	Domicilio	FechaNac	Editar	Eliminar
03069726P	Alejandro	Sánchezz	Calle San Isidro 5	2003-08-19		
07328669H	Rodrigo	Pérez	Calle Fernandez De l	2000-04-11		

Agregar Cliente

## Validación W3C:

### homepage.php

Message Filtering

- Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
[From line 9, column 5; to line 9, column 271](#)  

```
css">↵ <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.c...sANI9QpRohGBreCFkKxLhei6S9CQXFEBbKuqLg0DA=="
crossorigin="anonymous" referreerpolicy="no-referrer" />↵ ↵
```

Document checking completed. No errors or warnings to show.

### gestionEmpleados.php

Message Filtering

- Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).  
[From line 9, column 5; to line 9, column 271](#)  

```
ous">↵ <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.c...
```

Document checking completed. No errors or warnings to show.

### gestionVentas.php

#### Message Filtering

1. **Info** Trailing slash on void elements has no effect and interacts badly with unquoted attribute values.

From line 9, column 5, to line 9, column 271

```
ous">↵ <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.2/css/all.min.c...
```

Document checking completed. No errors or warnings to show.

### gestionClientes.php

Check by  ☐ css

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Gestión de Coches - Concesionario</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QOgp3LIm9Mao09vzitzQTwFspdyD6SVohhpucOmLASjC" crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" integrity="sha512-
z3glPd7yknf1V0lBcZqRKc4qy0r8gAkU1qmn+CSHxbuBusANI9QpRohGBreCFkKxLhe16S9CQXFEbbkuLg0DA==" crossorigin="anonymous" referrerpolicy="no-referrer" >
</head>
<body>
  <header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">

```

Check

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.

Source

1. ↵
2. <!DOCTYPE html>↵
3. <html lang="es">↵
4. <head>↵
5. <meta charset="UTF-8">↵
6. <meta name="viewport" content="width=device-width, initial-scale=1.0">↵

### gestionCoches.php

#### Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by  ☐ CSS

```
<input type="hidden" id="token" name="token" value="688c0f85dd04983858d8786ea06f2026550c1125b4a91a3735697e6b9ef74a79">
</div>
<div class="form-group m-2" style="width: 300px">
  <label for="pass">Contraseña</label>
  <input type="password" class="form-control" id="pass" name="pass" placeholder="Contraseña">
</div>
<!-- <input type="hidden" class="form-control" id="token" name="token" value="">$_SESSION['token'] ?> -->
<button type="submit" class="mt-3 btn btn-primary">Iniciar Sesión</button>
</form>
</div>
<!-- JavaScript y jQuery para habilitar los componentes de Bootstrap -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

Document checking completed. No errors or warnings to show.

Source

## index.php

#### Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by  ☐ CSS

```
<footer class="py-3 my-4">
  <ul class="nav justify-content-center border-bottom pb-3 mb-3">
    <li class="nav-item"><a href="#" class="nav-link px-2 text-body-secondary">Home</a></li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-body-secondary">Features</a></li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-body-secondary">Pricing</a></li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-body-secondary">FAQs</a></li>
    <li class="nav-item"><a href="#" class="nav-link px-2 text-body-secondary">About</a></li>
  </ul>
  <p class="text-center text-body-secondary">&copy; 2023 Company, Inc</p>
</footer>
<!-- JavaScript y jQuery para habilitar los componentes de Bootstrap -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

Document checking completed. No errors or warnings to show.

Source

## NOTAS:

La funcionalidad enviar un mail podría ser ejecutada en segundo plano o en un nuevo hilo ya que la función tiene un tiempo de espera para su ejecución de varios segundos.

Hemos usado task lists de git y diferentes ramas que nos han ayudado bastante a la hora de desarrollar la App:

Tasklist general

#1 opened 3 weeks ago by vvaldesc

Open

Filters

is:issue is:open

Labels 9

Milestones 0

New issue

7 Open

1 Closed

Author

Label

Projects

Milestones

Assignee

Sort

arreglar funciones

#14 opened 3 days ago by vvaldesc

1

Refactorizar

#13 opened 3 days ago by vvaldesc

1

DOC

#8 opened 4 days ago by vvaldesc

7 tasks

Restricciones CRUD

#7 opened 5 days ago by vvaldesc

2 tasks done

PHPmailer (OPCIONAL)

#5 opened 3 weeks ago by vvaldesc

2 of 4 tasks

Diferentes funcionalidades entre admin y usr (Incluye añadir y borrar elementos en las tablas)

#4 opened 3 weeks ago by vvaldesc

2 of 4 tasks

Tasklist general

#1 opened 3 weeks ago by vvaldesc

20 of 24 tasks

