

PROCEDIMIENTOS

PASO 1: LA BD (NO OLVIDE ACTUALIZAR SU CLASE DE CONEXIÓN CON EL NOMBRE BD)

```
Database complete.sql +
Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift+Space]->List Function and Routine Parameters.
1  -- Creación de la BD
2  CREATE DATABASE carstore;
3  USE carstore
4
5  -- Creación de la tabla
6  CREATE TABLE automoviles
7  (
8      idautomovil      INT AUTO INCREMENT PRIMARY KEY,
9      marca            VARCHAR(30)      NOT NULL,
10     modelo           VARCHAR(30)      NOT NULL,
11     precio            DECIMAL(9,2)    NOT NULL,
12     tipocombustible   VARCHAR(20)     NOT NULL,
13     color             VARCHAR(30)     NOT NULL,
14     create_at         DATETIME        NOT NULL DEFAULT NOW(),
15     update_at         DATETIME        NULL,
16     CONSTRAINT uk_marca_auto UNIQUE (marca, modelo)
17 )ENGINE = INNODB;
18
19 -- Registros de prueba
20 -- Más ejemplos en: https://neoauto.com/
21 INSERT INTO automoviles (marca, modelo, precio, tipocombustible, color) VALUES
22     ('Nissan', 'Sentra', 45000, 'Gasolina Premium', 'blanco'),
23     ('Kia', 'Sorento', 75000, 'GLP', 'gris');
24
```

```
25 -- Procedimientos almacenados
26 -- Ejecutar todo junto desde la siguiente línea hasta "DELIMITER ;"
27 DELIMITER $$
28
29 CREATE PROCEDURE spu_automoviles_listar()
30 BEGIN
31     SELECT * FROM automoviles ORDER BY idautomovil DESC;
32 END $$
33
34 CREATE PROCEDURE spu_automoviles_registrar
35 (
36     IN _marca            VARCHAR(30),
37     IN _modelo           VARCHAR(30),
38     IN _precio           DECIMAL(7,2),
39     IN _tipocombustible   VARCHAR(20),
40     IN _color            VARCHAR(30)
41 )
42 BEGIN
43     INSERT INTO automoviles (marca, modelo, precio, tipocombustible, color) VALUES
44         (_marca, _modelo, _precio, _tipocombustible, _color);
45 END $$
46
```

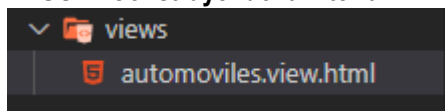
```
47
48 CREATE PROCEDURE spu_automoviles_eliminar
49 (
50     IN _idautomovil INT
51 )
52 BEGIN
53     DELETE FROM automoviles WHERE idautomovil = _idautomovil;
54 END $$
55
56 CREATE PROCEDURE spu_automoviles_obtener
57 (
58     IN _idautomovil INT
59 )
60 BEGIN
61     SELECT * FROM automoviles WHERE idautomovil = _idautomovil;
62 END $$
63
64
```

```

65
66 CREATE PROCEDURE spu_automoviles_actualizar
67 (
68     IN _idautomovil INT,
69     IN _marca VARCHAR(30),
70     IN _modelo VARCHAR(30),
71     IN _precio DECIMAL(7,2),
72     IN _tipocombustible VARCHAR(20),
73     IN _color VARCHAR(30)
74 )
75 BEGIN
76     UPDATE automoviles SET
77         marca = _marca,
78         modelo = _modelo,
79         precio = _precio,
80         tipocombustible = _tipocombustible,
81         color = _color
82     WHERE idautomovil = _idautomovil;
83 END $$
84
85 DELIMITER ; -- ejecutar hasta aquí
86

```

PASO 2: Construyendo la interfaz



Necesitaremos Bootstrap 5 y sus CDN:

[Download · Bootstrap v5.2 \(getbootstrap.com\)](https://getbootstrap.com/)

CDN via jsDelivr

Skip the download with [jsDelivr](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="style"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="

```

If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a CDN preferably.

```

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="

```

Así se ve en el navegador

Registro de vehículos

Marca

Seleccione

Modelo

Precio

Tipo de combustible

Seleccione

Color

Guardar

Reiniciar

#	Marca	Modelo	Precio	Tipo de combustible	Color	Command
1	Honda	Pilot 2023	78500.00	GNV	Azul	Quitar Editar
2	Toyota	Hilux	99999.99	Gasolina Premium	blanco	Quitar Editar
3	Kia	Sorento	75000.00	GLP	gris	Quitar Editar

Código:

```
views > automoviles.view.html > html > body > div.container > div.row.mt-3
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Automoviles</title>
8
9   <!-- BS5 -->
10  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" int
11
12 </head>
13 <body>
14
15   <div class="container">
16     <div class="row mt-3">
17
18       <!-- Aquí se construye el formulario -->
19       <div class="col-md-4">
```

A partir de la línea 15: Creamos un container, dentro una fila, la dividimos en 2 partes, 4 columnas izquierda para el formulario y 8 columnas hacia el lado derecho para la tabla

```
17
18   <!-- Aquí se construye el formulario -->
19   <div class="col-md-4">
20     <form action="" autocomplete="off" id="form-automoviles">
21       <div class="card">
22         <div class="card-header">
23           Registro de vehículos
24         </div>
25         <div class="card-body">
26           <div class="mb-3">
27             <label for="marca" class="form-label">Marca</label>
28             <select id="marca" class="form-select form-select-sm" autofocus>
29               <option value="">Seleccione</option>
30               <option value="Kia">Kia</option>
31               <option value="Nissan">Nissan</option>
32               <option value="Toyota">Toyota</option>
33               <option value="Mazda">Mazda</option>
34               <option value="Honda">Honda</option>
35             </select>
36           </div>
37           <div class="mb-3">
38             <label for="modelo" class="form-label">Modelo</label>
39             <input type="text" id="modelo" class="form-control form-control-sm">
40           </div>
41           <div class="mb-3">
42             <label for="precio" class="form-label">Precio</label>
43             <input type="text" id="precio" class="form-control form-control-sm">
44           </div>
45           <div class="mb-3">
46             <label for="tipocombustible" class="form-label">Tipo de combustible</label>
47             <select id="tipocombustible" class="form-select form-select-sm">
48               <option value="">Seleccione</option>
49               <option value="Gasolina Premium">Gasolina Premium</option>
50               <option value="Gasolina regular">Gasolina regular</option>
51               <option value="GNV">GNV</option>
52               <option value="GLP">GLP</option>
53             </select>
54           </div>
55           <div class="mb-3">
56             <label for="color" class="form-label">Color</label>
57             <input type="text" id="color" class="form-control form-control-sm">
58           </div>
59         </div>
60       </div>
61     </form>
62   </div>
```

Continúa siguiente página...

```
60 <div class="card-footer text-muted">
61   <div class="d-grid gap-2">
62     <button class="btn btn-sm btn-success" id="guardar" type="button">Guardar</button>
63     <button class="btn btn-sm btn-secondary" type="reset">Reiniciar</button>
64   </div>
65 </div> <!-- fin card-footer -->
66 </div> <!-- fin card -->
67 </form> <!-- fin formulario -->
68 </div> <!-- fin col-md-4 -->
```

En la siguiente columna se construye la tabla

```
69 <!-- Aquí se construye la tabla -->
70 <div class="col-md-8">
71   <table class="table table-sm table-striped" id="tabla-automoviles">
72     <thead>
73       <tr>
74         <th>#</th>
75         <th>Marca</th>
76         <th>Modelo</th>
77         <th>Precio</th>
78         <th>Tipo de combustible</th>
79         <th>Color</th>
80         <th>Command</th>
81       </tr>
82     </thead>
83     <tbody>
84     </tbody>
85   </tbody>
86 </table>
87 </div>
88 </div>
89 </div>
```

Luego en la parte inferior y antes creamos el MODAL

```
93 <!-- Modales -->
94 <!-- Vamos a repetir las capas y controles del formulario de registro, solo que en el ID, agregamos md-name -->
95 <div class="modal fade" id="modal-automoviles" tabindex="-1" data-bs-backdrop="static" data-bs-keyboard="false" role="dialog">
96   <div class="modal-dialog modal-dialog-scrollable modal-dialog-centered" role="document">
97     <div class="modal-content">
98       <div class="modal-header bg-info text-light">
99         <h5 class="modal-title" id="modalTitleId">Actualizar datos</h5>
100         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
101       </div>
102       <div class="modal-body">
103         <div class="mb-3">
104           <label for="md-marca" class="form-label">Marca</label>
105           <select id="md-marca" class="form-select form-select-sm" autofocus>
106             <option value="">Seleccione</option>
107             <option value="Kia">Kia</option>
108             <option value="Nissan">Nissan</option>
109             <option value="Toyota">Toyota</option>
110             <option value="Mazda">Mazda</option>
111             <option value="Honda">Honda</option>
112           </select>
113         </div>
114         <div class="mb-3">
115           <label for="md-modelo" class="form-label">Modelo</label>
116           <input type="text" id="md-modelo" class="form-control form-control-sm">
117         </div>
118       </div>
119     </div>
120   </div>
121 </div>
```

Continúa siguiente página...

```

119     <div class="mb-3">
120         <label for="md-precio" class="form-label">Precio</label>
121         <input type="text" id="md-precio" class="form-control form-control-sm">
122     </div>
123     <div class="mb-3">
124         <label for="md-tipocombustible" class="form-label">Tipo de combustible</label>
125         <select id="md-tipocombustible" class="form-select form-select-sm">
126             <option value="">Seleccione</option>
127             <option value="Gasolina Premium">Gasolina Premium</option>
128             <option value="Gasolina regular">Gasolina regular</option>
129             <option value="GNV">GNV</option>
130             <option value="GLP">GLP</option>
131         </select>
132     </div>
133     <div class="mb-3">
134         <label for="md-color" class="form-label">Color</label>
135         <input type="text" id="md-color" class="form-control form-control-sm">
136     </div>
137 </div>
138 <div class="modal-footer">
139     <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>
140     <button type="button" id="actualizar" class="btn btn-primary">Actualizar</button>
141 </div>
142 </div>
143 </div>
144 </div>
145 <!-- Fin modales -->
146

```

*El ID se actualizó en cada control, agregándole un **md-name***

Antes de pasar al código JS, no olvide las librerías de Bootstrap 5

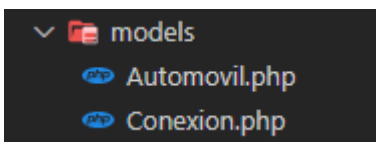
```

147
148 <!-- librerías Bootstrap 5 -->
149 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFx
150 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-oBqDVmMz9ATKxIep9tiCxS/Z9
151 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js" integrity="sha384-cuYeSxntonz0PPNlHhBs68uyIAVpI
152
153
154 <script>

```

PASO 3: Modelos

Los modelos contienen la lógica de la aplicación, aprovecharán a los SPU que construimos en el paso 1



Construya **Automovil.php**

```

Automovil.php X
models > Automovil.php
1  <?php
2
3  require_once 'Conexion.php';
4
5  class Automovil extends Conexion{
6
7      private $conexion;
8
9      //Constructor
10     public function __CONSTRUCT(){
11         $this->conexion = parent::getConexion();
12     }
13

```

Continúa en la siguiente página

```

13
14 ✓ public function listar(){
15 ✓     try{
16         $consulta = $this->conexion->prepare("CALL spu_automoviles_listar()");
17         $consulta->execute();
18         return $consulta->fetchAll(PDO::FETCH_ASSOC);
19     }
20 ✓     catch(Exception $e){
21         //Manejar el error según criterio...
22         die($e->getMessage());
23     }
24 }
25
26 ✓ public function registrar($datos = []){
27 ✓     $respuesta = [
28         "status" => false,
29         "message" => ""
30     ];
31 ✓     try{
32         $consulta = $this->conexion->prepare("CALL spu_automoviles_registrar(?,?,?,?,?)");
33 ✓         $respuesta["status"] = $consulta->execute(
34 ✓             array(
35                 $datos["marca"],
36                 $datos["modelo"],
37                 $datos["precio"],
38                 $datos["tipocombustible"],
39                 $datos["color"]
40             )
41         );
42     }
43 ✓     catch(Exception $e){
44         $respuesta["message"] = "No se ha podido completar el proceso. Código error: " . $e->getCode();
45     }
46     return $respuesta;
47 }
48

```

```

48
49 ✓ public function eliminar($idautomovil = 0){
50 ✓     $respuesta = [
51         "status" => false,
52         "message" => ""
53     ];
54 ✓     try{
55         $consulta = $this->conexion->prepare("CALL spu_automoviles_eliminar(?)");
56         $respuesta["status"] = $consulta->execute(array($idautomovil));
57     }
58 ✓     catch(Exception $e){
59         $respuesta["message"] = "No se ha podido completar el proceso. Código error: " . $e->getCode();
60     }
61
62     return $respuesta;
63 }
64
65
66 ✓ public function obtener($idautomovil = 0){
67 ✓     try{
68         $consulta = $this->conexion->prepare("CALL spu_automoviles_obtener(?)");
69         $consulta->execute(array($idautomovil));
70         return $consulta->fetch(PDO::FETCH_ASSOC);
71     }
72 ✓     catch(Exception $e){
73         die($e->getMessage());
74     }
75 }
76

```

Continúa en la siguiente página...

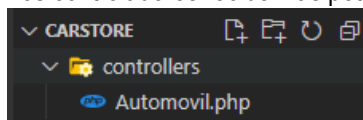
```

76
77
78 public function actualizar($datos = []){
79     $respuesta = [
80         "status" => false,
81         "message" => ""
82     ];
83     try{
84         $consulta = $this->conexion->prepare("CALL spu_automoviles_actualizar(?,?,?,?,?)");
85         $respuesta["status"] = $consulta->execute(
86             array(
87                 $datos["idautomovil"],
88                 $datos["marca"],
89                 $datos["modelo"],
90                 $datos["precio"],
91                 $datos["tipocombustible"],
92                 $datos["color"]
93             )
94         );
95     }
96     catch(Exception $e){
97         $respuesta["message"] = "No se ha podido completar el proceso. Código error: " . $e->getCode();
98     }
99     return $respuesta;
100 }
101
102 }

```

PASO 4: Controladores

Los controladores reciben las peticiones (request) y entregan respuestas (response)



```

Automovil.php X
controllers > Automovil.php
1  <?php
2  require_once '../models/Automovil.php';
3
4  if (isset($_POST['operacion'])){
5
6      $automovil = new Automovil();
7
8      if ($_POST['operacion'] == 'listar'){
9
10         $datosObtenidos = $automovil->listar();
11         if ($datosObtenidos){
12             echo json_encode($datosObtenidos);
13         }
14     }
15
16     if ($_POST['operacion'] == 'registrar'){
17         $datosGuardar = [
18             "marca" => $_POST['marca'],
19             "modelo" => $_POST['modelo'],
20             "precio" => $_POST['precio'],
21             "tipocombustible" => $_POST['tipocombustible'],
22             "color" => $_POST['color']
23         ];
24
25         $respuesta = $automovil->registrar($datosGuardar);
26         echo json_encode($respuesta);
27     }
28

```

Continúa en la siguiente página...

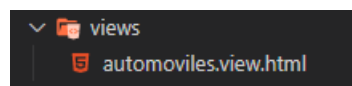
```

29
30     if ($_POST['operacion'] == 'eliminar'){
31         $respuesta = $automovil->eliminar($_POST['idautomovil']);
32         echo json_encode($respuesta);
33     }
34
35
36     if ($_POST['operacion'] == 'obtener'){
37         $respuesta = $automovil->obtener($_POST['idautomovil']);
38         echo json_encode($respuesta);
39     }
40
41
42     if ($_POST['operacion'] == 'actualizar'){
43         $datosActualizar = [
44             "idautomovil"      => $_POST['idautomovil'],
45             "marca"            => $_POST['marca'],
46             "modelo"           => $_POST['modelo'],
47             "precio"           => $_POST['precio'],
48             "tipocombustible"   => $_POST['tipocombustible'],
49             "color"            => $_POST['color']
50         ];
51
52         $respuesta = $automovil->actualizar($datosActualizar);
53         echo json_encode($respuesta);
54     }
55 }
56

```

PASO 5: Vista

Ya tenemos toda la lógica resuelta en el BACKEND, ahora y con la ayuda de FETCH, enviaremos o consultaremos los datos.



Desarrollar en `automoviles.view.html` (este archivo ya debe existir)

Iniciamos con el evento `DOMContentLoaded`

```

154 <script>
155     document.addEventListener('DOMContentLoaded', () => {
156
157         //Objetos y variables
158         let idautomovil = ``;
159         const tabla = document.querySelector("#tabla-automoviles");
160         const cuerpoTabla = tabla.querySelector("tbody");
161         const btGuardar = document.querySelector("#guardar");
162         const btActualizar = document.querySelector("#actualizar");
163         const modal = new bootstrap.Modal(document.querySelector("#modal-automoviles"));
164

```

Construyendo objetos

Continúa en la siguiente página...


```

165 //Funciones
166 function renderData(){
167     const parametros = new URLSearchParams();
168     parametros.append("operacion", "listar");
169
170     fetch("../controllers/Automovil.php", {
171         method: 'POST',
172         body: parametros
173     })
174     .then(response => response.json())
175     .then(datos => {
176         cuerpoTabla.innerHTML = '';
177         let numeroFila = 1;
178         datos.forEach(registro => {
179             const fila = `
180                 <tr>
181                     <td>${numeroFila}</td>
182                     <td>${registro.marca}</td>
183                     <td>${registro.modelo}</td>
184                     <td>${registro.precio}</td>
185                     <td>${registro.tipocombustible}</td>
186                     <td>${registro.color}</td>
187                     <td>
188                         <a href='#' class='eliminar' data-idautomovil='${registro.idautomovil}'>Quitar</a>
189                         <a href='#' class='editar' data-idautomovil='${registro.idautomovil}'>Editar</a>
190                     </td>
191                 </tr>
192             `;
193             cuerpoTabla.innerHTML += fila;
194             numeroFila++;
195         });
196     });
197 }
198

```

Puede llamar a esta función y verá como los datos se renderizan en la tabla HTML

```

198
199 function automovilRegister(){
200     if (confirm("¿Está seguro de registrar?")){
201         const parametros = new URLSearchParams();
202         parametros.append("operacion", "registrar");
203
204         //enviaremos los datos del formulario
205         parametros.append("marca", document.querySelector("#marca").value);
206         parametros.append("modelo", document.querySelector("#modelo").value);
207         parametros.append("precio", document.querySelector("#precio").value);
208         parametros.append("tipocombustible", document.querySelector("#tipocombustible").value);
209         parametros.append("color", document.querySelector("#color").value);
210
211         fetch("../controllers/Automovil.php", {
212             method: 'POST',
213             body: parametros
214         })
215         .then(response => response.json())
216         .then(datos => {
217             if (datos.status){
218                 renderData();
219                 document.querySelector("#form-automoviles").reset();
220                 document.querySelector("#marca").focus();
221             }else{
222                 alert(datos.message);
223             }
224         });
225     }
226 }
227

```

Función para registrar (guardar como datos nuevos)

Continúa en la siguiente página...

```

228 ✓ function automovilUpdate(){
229 ✓   if (confirm("¿Está seguro de actualizar?")){
230     const parametros = new URLSearchParams();
231     parametros.append("operacion", "actualizar");
232
233     //enviaremos los datos del formulario (el que se encuentra en el MODAL)
234     parametros.append("idautomovil", idautomovil);
235     parametros.append("marca", document.querySelector("#md-marca").value);
236     parametros.append("modelo", document.querySelector("#md-modelo").value);
237     parametros.append("precio", document.querySelector("#md-precio").value);
238     parametros.append("tipocombustible", document.querySelector("#md-tipocombustible").value);
239     parametros.append("color", document.querySelector("#md-color").value);
240
241 ✓   fetch("../controllers/Automovil.php", {
242     method: 'POST',
243     body: parametros
244   })
245     .then(response => response.json())
246 ✓   .then(datos => {
247 ✓     if (datos.status){
248       renderData();
249       modal.toggle();
250     }else{
251       alert(datos.message);
252     }
253   });
254 }
255 }
256

```

```

257 //Proceso eliminación
258 cuerpoTabla.addEventListener("click", (event) => {
259   if (event.target.classList[0] === 'eliminar'){
260     if (confirm("¿Está seguro de eliminar?")){
261       idautomovil = parseInt(event.target.dataset.idautomovil);
262
263       const parametros = new URLSearchParams();
264       parametros.append("operacion", "eliminar");
265       parametros.append("idautomovil", idautomovil);
266
267       fetch("../controllers/Automovil.php", {
268         method: 'POST',
269         body: parametros
270       })
271         .then(response => response.json())
272         .then(datos => {
273           console.log(datos);
274           if (datos.status){
275             renderData();
276           }else{
277             alert(datos.message);
278           }
279         });
280     }
281   }
282 }
283 });
284

```

Continúa en la siguiente página...

```

285 //Proceso edición
286 cuerpoTabla.addEventListener("click", (event) => {
287     if (event.target.classList[0] === 'editar'){
288         idautomovil = parseInt(event.target.dataset.idautomovil);
289
290         const parametros = new URLSearchParams();
291         parametros.append("operacion", "obtener");
292         parametros.append("idautomovil", idautomovil);
293
294         fetch("../controllers/Automovil.php", {
295             method: 'POST',
296             body: parametros
297         })
298         .then(response => response.json())
299         .then(datos => {
300             document.querySelector("#md-marca").value = datos.marca;
301             document.querySelector("#md-modelo").value = datos.modelo;
302             document.querySelector("#md-precio").value = datos.precio;
303             document.querySelector("#md-tipocombustible").value = datos.tipocombustible;
304             document.querySelector("#md-color").value = datos.color;
305
306             modal.toggle();
307         });
308     }
309 })
310

```

Para finalizar...

```

310
311 //Eventos
312 btGuardar.addEventListener("click", automovilRegister);
313 btActualizar.addEventListener("click", automovilUpdate);
314
315 //Funciones automáticas
316 renderData();
317
318 });
319 </script>
320
321 </body>
322 </html>

```