



**ULS**



**UNIVERSIDAD LUTERANA SALVADOREÑA**



**FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA**

**ASIGNATURA:**  
PROGRAMACIÓN 2

**TEMA**  
2° AVANCE SISTEMA DE RESERVA DE AUTOS

**DOCENTE:**  
LIC. JOSE LUIS ALVARADO

**PRESENTADO POR:**  
JAVIER MARQUEZ PORTILLO  
ALICIA GUADALUPE PLEITEZ  
CRISTIAN ANTONIO CERON ENRÍQUEZ  
EDENILSON ARÍSTIDES AYALA GUARDADO  
JACKELINE DEL CARMEN ASUNCION MOLINA

SAN SALVADOR, 27 DE MAYO DE 2017



ULS



## ÍNDICE

Introducción .....	2
<b>PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>3</b>
<b>BASES DE DATOS .....</b>	<b>4</b>
Modelo Entidad - Relación.....	5
Diagrama Físico.....	6
Tabla de contenidos.....	7
Diccionario de Datos.....	9
Script de la base de datos .....	11
<b>CONEXION A LA BASE DE DATOS .....</b>	<b>19</b>
<b>CARPETA EMPLEADO.....</b>	<b>22</b>
<b>CARPETA: RESERVACIONES.....</b>	<b>26</b>
<b>CARPETA: VEHICULOS.....</b>	<b>30</b>
<b>CONCLUSIÓN .....</b>	<b>34</b>

## Introducción

En el presente documento se plantea una propuesta de solución para cualquier empresa que cuente con flota vehicular, la cual sea usada por sus empleados, así como también para las que se dedican a la renta de carros; y que no cuenten con un sistema de reserva de vehículos en línea, facilitando el registro, reserva y agilizando dicho trámite.

Ya que se ha hará pensando en que sea aplicada por una empresa donde sus empleados hacen uso de los vehículos, se crea una base de datos completa donde integra las generalidades de los empleados, cargos, los detalles de vehículo (marca, modelo) y datos de las reservas, esto con el fin de llevar un adecuado registro y también para generar un historial de las reservas de los automotores.

Dicha base de datos se realizó en el gestor de bases de MySQL y el programa PHPMYADMIN con sentencias SQL, según lo aprendido en clase, aplicando el conocimiento adquirido en la materia de programación 2 e investigando más contenido que nos fue útil a la hora de estructurar el sistema. Se realizó la conexión a PHPMYADMIN y PHP para la visualización de su entorno gráfico y práctico para el uso de la persona encargada de las reservas y/o uso de los vehículos, así como también brindarle una herramienta de uso fácil y comprensible según las especificaciones de los requerimientos establecidos desde el principio de la investigación y que además le permita imprimir en PDF los datos que requiera de las reservas.

## PLANTEAMIENTO DEL PROBLEMA

Se desea diseñar un sistema con una base de datos para llevar un control y registro de las reservas de vehículos de una empresa "X".

Para acceder al sistema de reservas, se necesita ser administrador o empleado, se desea conocer todos los datos generales de estos, así como el cargo que ocupa dentro de la empresa; y como usuario se tomara el primer nombre y la contraseña será su primer apellido. También se desea conocer los datos de los vehículos disponibles.

Además se desea conocer poder visualizar que vehículos están disponibles y cuales ya están reservados; pudiendo hacer registros de reservas por rangos de tiempo.

Se desea también agregar uno o varios vehículos si es necesario; pensando en que la empresa pueda aumentar su flota vehicular.

En la visualización de la flota vehicular se debe mostrar todos los datos de cada auto; donde el sistema mostrará una imagen de cada uno, junto a sus características.

## BASES DE DATOS

Para este manual nosotros creamos una base de datos con el nombre **reservass** la cual contiene seis tablas.

La primera con el nombre **Cargos** en la cual se encuentran dos campos (Id\_Cargo “Llave primaria” y nombre\_cargo).

La segunda con el nombre **Empleados** en la cual se encuentran diez campos (Id\_empleado “Llave primaria”, nombres, apellidos, user, pass, DUI, dirección, teléfono, Id\_cargo “Llave foránea” Y estado).

La tercera con el nombre **Marcas** en la cual se encuentran dos campos (Id\_marca “Llave primaria” y nombre\_marca).

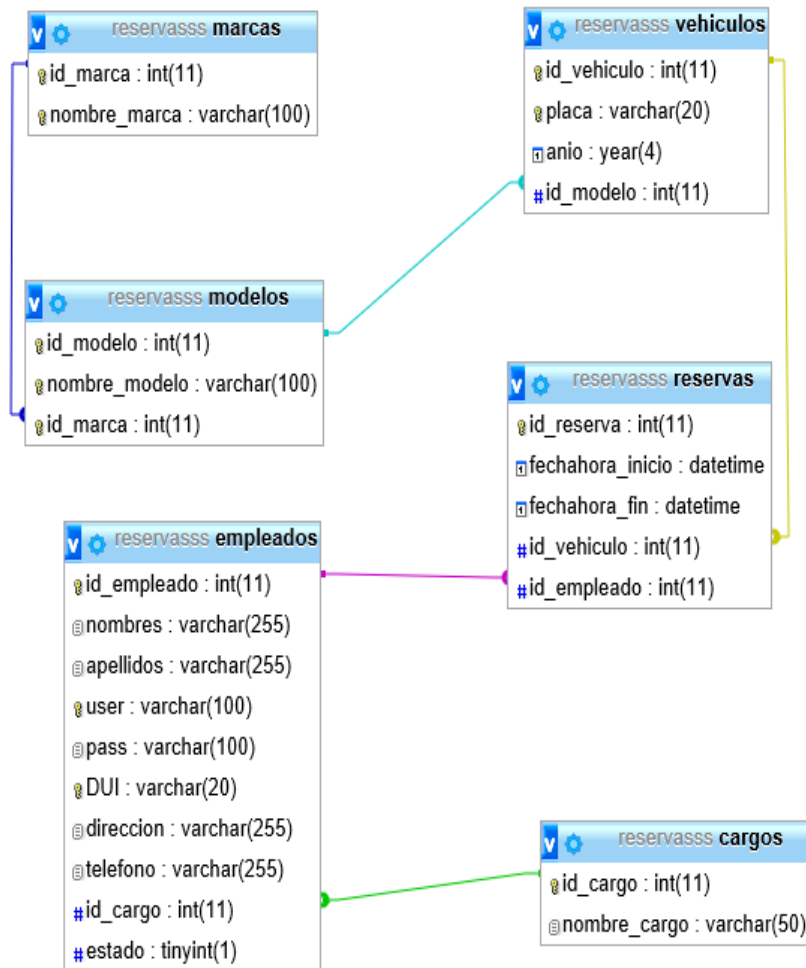
La cuarta con el nombre **Modelos** con los campos (Id\_modelo “Llave primaria”, nombre\_modelo y id\_marca “Llave foránea” ).

La quinta con el nombre **Reservas** que contiene los campos ( id\_reserva ”Llave primaria”, fechahora\_inicio, fechahora\_fin, id\_vehiculo “Llave foránea”, id\_empleado “Llave foránea”).

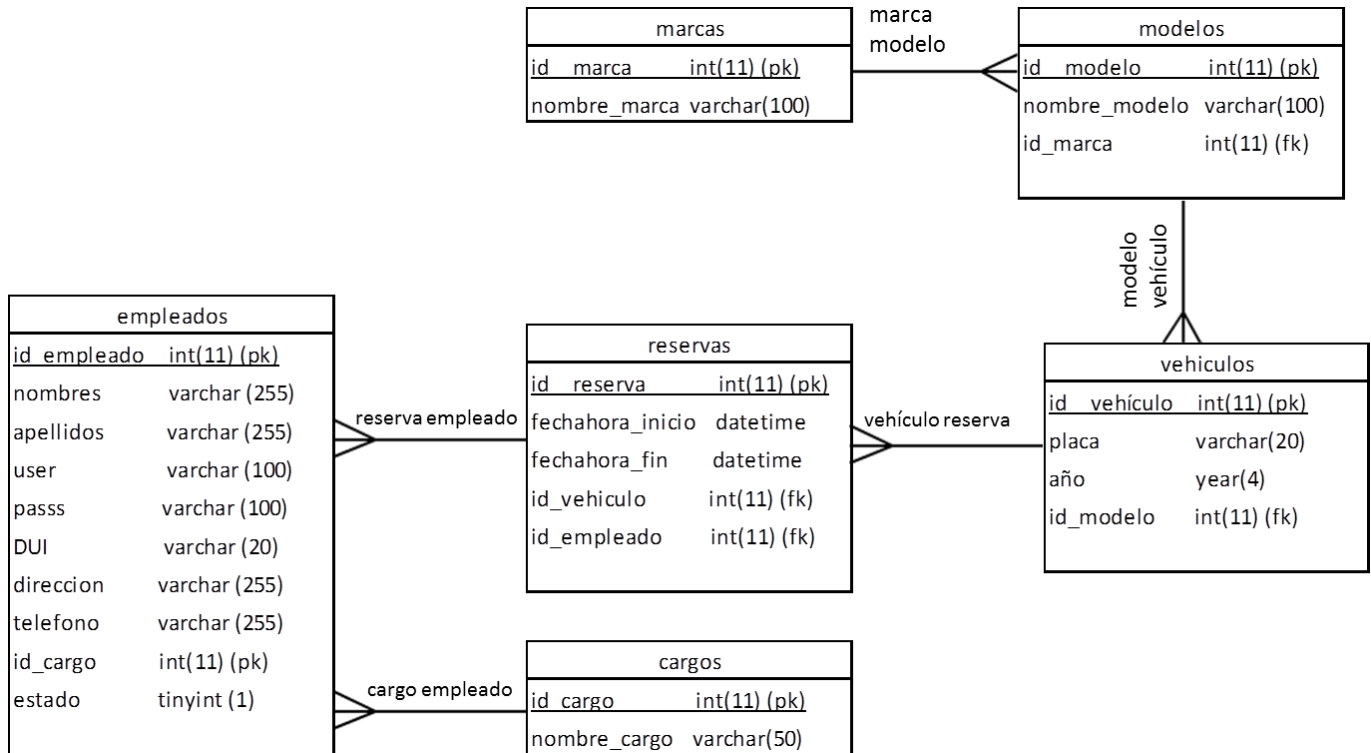
La ultima con el nombre **Vehiculos** la cual tiene los campos ( id\_vehiculo “Llave primaria”, placa, anio, id\_modelo “Llave foránea”).



## Modelo Entidad - Relación



## Diagrama Físico



## Tabla de contenidos

### 1- cargos

Columna	Tipo	Atributos	Null	redeterminado	Extra	Enlaces a	Comentarios	MIME
id_cargo	int(11)		No		auto_increment			
nombre_cargo	varchar(50)		No					

### 2- empleados

Columna	Tipo	Atributos	Null	redeterminado	Extra	Enlaces a	Comentarios	MIME
id_empleado	int(11)		No		auto_increment			
nombres	varchar(255)		No					
apellidos	varchar(255)		No					
user	varchar(100)		No	0				
pass	varchar(100)		No	0				
DUI	varchar(20)		No					
direccion	varchar(255)		No					
telefono	varchar(255)		No					
id_cargo	int(11)		No			-> cargos.id_cargo ON UPDATE RESTRICT ON DELETE RESTRICT		
estado	tinyint(1)		Sí	1				

### 3- marcas

Columna	Tipo	Atributos	Null	redeterminado	Extra	Enlaces a	Comentarios	MIME
id_marca	int(11)		No		auto_increment			
nombre_marca	varchar(100)		No					

### 4- modelos

Columna	Tipo	Atributos	Null	redeterminado	Extra	Enlaces a	Comentarios	MIME
id_modelo	int(11)		No		auto_increment			
nombre_modelo	varchar(100)		No					
id_marca	int(11)		No			-> marcas.id_marca ON UPDATE RESTRICT ON DELETE RESTRICT		



**5- reservas**

Columna	Tipo	Atributos	Nulo	redeterminado	Extra	Enlaces a	Comentarios	MIME
id_reserva	int(11)		No		auto_increment			
fechahora_inicio	datetime		No					
fechahora_fin	datetime		No					
id_vehiculo	int(11)		No			-> vehiculos.id_vehiculo ON UPDATE RESTRICT ON DELETE RESTRICT		
id_empleado	int(11)		No			-> empleados.id_empleado ON UPDATE RESTRICT ON DELETE RESTRICT		

**6- vehículos**

Columna	Tipo	Atributos	Nulo	redeterminado	Extra	Enlaces a	Comentarios	MIME
id_vehiculo	int(11)		No		auto_increment			
placa	varchar(20)		No					
anio	year(4)		No					
id_modelo	int(11)		No			-> modelos.id_modelo ON UPDATE RESTRICT ON DELETE RESTRICT		

## Diccionario de Datos

### cargos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id_cargo (Primaria)	int(11)	No				
nombre_cargo	varchar(50)	No				

### Índices

¡No se ha definido ningún índice!

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_cargo	8	A	No	

### empleados

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id_empleado (Primaria)	int(11)	No				
nombres	varchar(255)	No				
apellidos	varchar(255)	No				
user	varchar(100)	No	0			
pass	varchar(100)	No	0			
DUI	varchar(20)	No				
direccion	varchar(255)	No				
telefono	varchar(255)	No				
id_cargo	int(11)	No		cargos -> id_cargo		
estado	tinyint(1)	Sí	1			

### Índices

¡No se ha definido ningún índice!

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_empleado	13	A	No	
DUI	BTREE	Sí	No	DUI	13	A	No	
user	BTREE	Sí	No	user	13	A	No	
id_cargo	BTREE	No	No	id_cargo	13	A	No	

**marcas**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id_marca (Primaria)	int(11)	No				
nombre_marca	varchar(100)	No				

**Índices**

¡No se ha definido ningún índice!

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_marca	7	A	No	
				nombre_marca	7	A	No	

**modelos**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id_modelo (Primaria)	int(11)	No				
nombre_modelo	varchar(100)	No				
id_marca	int(11)	No		marcas -> id_marca		

**Índices**

¡No se ha definido ningún índice!

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_modelo	13	A	No	
				id_marca	13	A	No	
				nombre_modelo	13	A	No	

**reservas**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id_reserva (Primaria)	int(11)	No				
fechahora_inicio	datetime	No				
fechahora_fin	datetime	No				
id_vehiculo	int(11)	No		vehiculos -> id_vehiculo		
id_empleado	int(11)	No		empleados -> id_empleado		

**Índices**

¡No se ha definido ningún índice!

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_reserva	4	A	No	
				id_vehiculo	4	A	No	
				id_empleado	4	A	No	

**vehiculos**

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios	MIME
id_vehiculo (Primaria)	int(11)	No				
placa	varchar(20)	No				
anio	year(4)	No				
id_modelo	int(11)	No		modelos -> id_modelo		

**Índices**

¡No se ha definido ningún índice!

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_vehiculo	12	A	No	
placa	BTREE	Sí	No	placa	12	A	No	
id_modelo	BTREE	No	No	id_modelo	12	A	No	

**Script de la base de datos**

```
CREATE DATABASE Reservasss;
```

```
USE Reservasss;
```

```
DROP TABLE IF EXISTS `Cargos`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `Cargos` (
```

```
  `id_cargo` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `nombre_cargo` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
```

```
  PRIMARY KEY (`id_cargo`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
```

```
COLLATE=utf8mb4_unicode_ci;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
Dumping data for table `Cargos`
```

```
LOCK TABLES `Cargos` WRITE;
```

```
/*!40000 ALTER TABLE `Cargos` DISABLE KEYS */;
```

```
INSERT INTO `Cargos` VALUES
(1,'Director'),(2,'Presidente'),(3,'Secretaria'),(4,'Conductor'),(5,'Vigilante'),(6,'As
esor'),(7,'Ordenanza'),(8,'Coordinador_Area');
/*!40000 ALTER TABLE `Cargos` ENABLE KEYS */;
UNLOCK TABLES;
```

Table structure for table `Empleados`

```
DROP TABLE IF EXISTS `Empleados`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Empleados` (
  `id_empleado` int(11) NOT NULL AUTO_INCREMENT,
  `nombres` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `apellidos` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `user` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '0',
  `pass` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '0',
  `DUI` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL,
  `direccion` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `telefono` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `id_cargo` int(11) NOT NULL,
  `estado` tinyint(1) DEFAULT '1',
  PRIMARY KEY (`id_empleado`),
  UNIQUE KEY `DUI` (`DUI`),
  UNIQUE KEY `user` (`user`),
  KEY `id_cargo` (`id_cargo`),
```

```
CONSTRAINT `Empleados_ibfk_1` FOREIGN KEY (`id_cargo`)
REFERENCES `Cargos` (`id_cargo`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `Empleados`

```
LOCK TABLES `Empleados` WRITE;
/*!40000 ALTER TABLE `Empleados` DISABLE KEYS */;
INSERT INTO `Empleados` VALUES (1,'Carlos Antonio','Carbajal
Moz','carlosantonio','carbajal','0987765-3','San Salvador','7698-
4522',1,1),(2,'Mario Cesar','Martinez','mariocesar','martinez','09565465-3','San
Martin','7865-9854',2,1),(3,'Pedro Luis','Polanco
Lopez','pedrioluis','polanco','46453345-3','San Jacinto','7544-
5434',4,1),(4,'Steffany Beatriz','Suarez
Menjivar','steffanybeatriz','suarez','45444573','Soyapango','7433-
5344',3,1),(5,'Juan Alfredo','Cerna Lopez','juanalfredo','cerna','3235541-3','San
Marcos','7553-5223',5,1),(6,'Juan Antonio','Menjivar
Beria','juanantonio','menjivar','3264441-4','San Martin','7233-4663',6,1),(7,'Jose
Geremias','Lopez Madrid','josegeremias','lopez','6565541-4','Ilopango','7434-
44663',5,1),(8,'Elias Joel','Morales Rivas','eliasjoel','morales','6456541-
4','Santa Elena','7656-4543',8,1),(9,'Luis Angel','Aguirre
Alfaro','luisangel','aguirre','6787841-4','Santo Tomas','7645-
4356',6,1),(10,'Carlos Reymundo','Ramirez
Garcia','carlosreymundo','ramirez','78544-4','San Jacinto','7643-
4456',6,1),(12,'Javier','Portillo','javier','portillo','524352','lkajsdflkjadsflkfj 09iqf
```

```
0u','21686158613518',7,1),(13,'Cristian','Ceron','cristian','ceron','5646553-9',
'Por ahi','65484651',7,1),(15,'Aristides','Ayala','aristides','ayala','518618631-8',
'Por ahi','65496549',7,1);
/*!40000 ALTER TABLE `Empleados` ENABLE KEYS */;
UNLOCK TABLES;
```

Table structure for table `Marcas`

```
DROP TABLE IF EXISTS `Marcas`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Marcas` (
  `id_marca` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_marca` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`id_marca`),
  UNIQUE KEY `nombre_marca` (`nombre_marca`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `Marcas`

```
LOCK TABLES `Marcas` WRITE;
/*!40000 ALTER TABLE `Marcas` DISABLE KEYS */;
```

```
INSERT INTO `Marcas` VALUES
(5,'Honda'),(6,'Hyundai'),(3,'KIA'),(7,'Mazda'),(2,'Nissan'),(4,'Suzuki'),(1,'Toyota'
);
/*!40000 ALTER TABLE `Marcas` ENABLE KEYS */;
UNLOCK TABLES;
```

Table structure for table `Modelos`

```
DROP TABLE IF EXISTS `Modelos`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Modelos` (
  `id_modelo` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_modelo` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `id_marca` int(11) NOT NULL,
  PRIMARY KEY (`id_modelo`),
  UNIQUE KEY `id_marca` (`id_marca`,`nombre_modelo`),
  CONSTRAINT `Modelos_ibfk_1` FOREIGN KEY (`id_marca`) REFERENCES
`Marcas` (`id_marca`))
ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `Modelos`



```
LOCK TABLES `Modelos` WRITE;
/*!40000 ALTER TABLE `Modelos` DISABLE KEYS */;
INSERT INTO `Modelos` VALUES
(2,'Corolla',1),(1,'Hilux',1),(4,'Frontier',2),(3,'Sentra',2),(5,'Picanto',3),(6,'Rio',3),(
8,'Celerio',4),(7,'Swift',4),(10,'Accord',5),(9,'Civic',5),(11,'Accent',6),(12,'Elantra',
6),(16,'RX8-MS',7);
/*!40000 ALTER TABLE `Modelos` ENABLE KEYS */;
UNLOCK TABLES;
```

Table structure for table `Reservas`

```
DROP TABLE IF EXISTS `Reservas`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Reservas` (
  `id_reserva` int(11) NOT NULL AUTO_INCREMENT,
  `fechahora_inicio` datetime NOT NULL,
  `fechahora_fin` datetime NOT NULL,
  `id_vehiculo` int(11) NOT NULL,
  `id_empleado` int(11) NOT NULL,
  PRIMARY KEY (`id_reserva`),
  KEY `id_vehiculo` (`id_vehiculo`),
  KEY `id_empleado` (`id_empleado`),
  CONSTRAINT `Reservas_ibfk_1` FOREIGN KEY (`id_vehiculo`)
REFERENCES `Vehiculos` (`id_vehiculo`),
  CONSTRAINT `Reservas_ibfk_2` FOREIGN KEY (`id_empleado`)
REFERENCES `Empleados` (`id_empleado`))
```

```
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
Dumping data for table `Reservas`  
LOCK TABLES `Reservas` WRITE;  
/*!40000 ALTER TABLE `Reservas` DISABLE KEYS */;  
INSERT INTO `Reservas` VALUES (1,'2016-10-30 08:00:00','2016-10-30  
10:00:00',1,1),(3,'2016-11-19 10:00:00','2016-11-19 15:00:00',9,12),(4,'2016-  
11-19 08:00:00','2016-11-19 11:00:00',11,10),(5,'2016-10-30 13:00:00','2016-  
10-30 17:00:00',6,10);  
/*!40000 ALTER TABLE `Reservas` ENABLE KEYS */;  
UNLOCK TABLES;
```

Table structure for table `Vehiculos`

```
DROP TABLE IF EXISTS `Vehiculos`;  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `Vehiculos` (  
  `id_vehiculo` int(11) NOT NULL AUTO_INCREMENT,  
  `placa` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `anio` year(4) NOT NULL,  
  `id_modelo` int(11) NOT NULL,  
  PRIMARY KEY (`id_vehiculo`),  
  UNIQUE KEY `placa` (`placa`),  
  KEY `id_modelo` (`id_modelo`),  
  CONSTRAINT `Vehiculos_ibfk_1` FOREIGN KEY (`id_modelo`)  
REFERENCES `Modelos` (`id_modelo`))
```

```
ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

Dumping data for table `Vehiculos`

```
LOCK TABLES `Vehiculos` WRITE;  
/*!40000 ALTER TABLE `Vehiculos` DISABLE KEYS */;  
INSERT INTO `Vehiculos` VALUES  
(1,'P68516',2010,1),(4,'P234223',2016,3),(6,'P2245244',2001,5),(7,'P76534',20  
11,6),(8,'P76345',2013,7),(9,'P53423',2003,8),(10,'P567423',2005,9),(11,'P987  
564',0000,10),(12,'P563465',2016,11),(13,'P123454',2017,12),(14,'P288-  
123',1987,3),(16,'P123-123',2016,16);  
/*!40000 ALTER TABLE `Vehiculos` ENABLE KEYS */;  
UNLOCK TABLES;
```

## CONEXION A LA BASE DE DATOS

Creamos las conexiones hacia la base de datos especificada en la configuración.

La conexión con MySQL se hace utilizando PDO.

A continuación se dan a conocer los códigos considerados como básicos para realizar la conexión a bases de datos, los cuales son fundamentales para que el sistema funcione correctamente.

Con la función ***public function query(\$preparedStatement, \$inputParameters)***, se *realiza* una consulta a la base de datos y retorna las filas obtenidas. Este método se utiliza para operaciones de tipo SELECT.

```
*
*
* @param string $preparedStatement La consulta en SQL a ejecutar.
* @param array $inputParameters Lista con los parametros a insertar en la
* consulta SQL.
* @return array Lista con los registros obtenidos de la base de datos.
*/
public function query($preparedStatement, $inputParameters)
{
    $query = $this->connection->prepare($preparedStatement);
    $query->execute($inputParameters);

    return $query->fetchAll();
}
/*+|
```

Con la función **public function execute(\$preparedStatement, \$inputParameters)**, se ejecuta una consulta a la base de datos sin retornar datos obtenidos. Este método se utiliza para operaciones de tipo INSERT, UPDATE, DELETE.

```

*
*
*
* @param string $preparedStatement La consulta en SQL a ejecutar.
* @param array $inputParameters Lista con los parametros a insertar en
* consulta SQL.
* @return boolean True si la consuta se ejecutó con éxito, false de lo
* contrario.
*/
public function execute($preparedStatement, $inputParameters)
{
    $query = $this->connection->prepare($preparedStatement);
    return $query->execute($inputParameters);
}

```

Clase  
base  
para  
las  
lectura

s de una base de datos MySQL.

Lee la lista de los campos filtrados, y si el usuario ha ingresado un filtro, se agrega el filtro al query.

```

1 <?php
2
3 class MySQLReader
4 {
5     public $baseQuery = '';
6     protected $filters = [];
7     protected $params = [];
8     protected $connection;
9
10    public $filterFields = [
11
12    ];
13    public function __construct()
14    {
15        $this->connection = new MySQL();
16    }
17    /**
18     * @param array $input Lista con el input del usuario
19     */
20    protected function setFilters($input)
21    {
22        foreach ($this->filterFields as $filter) {
23            if (empty($input[$filter['filterName']]) === false) {
24
25                // Si el filtro es de tipo LIKE. Ej: SELECT * FROM tabla WHERE
26                // columna LIKE 'termino de búsqueda%'
27                if ($filter['searchType'] === 'LIKE') {
28                    $this->filters[] = "{$filter['tableName']}.{$filter['columnName']}
29                    } LIKE :{$filter['columnName']}";
30                    $this->params[$filter['columnName']] = $input[$filter['filterName']] . '%';
31                }
32            }
33        }
34    }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Realiza una lectura de los registros en una tabla, con filtros de búsqueda. Mediante la función **public function read(\$input)**

```

37 // Si el filtro es del tipo IGUAL A. Ej: SELECT * FROM tabla WHERE
38 columna = 'termino de búsqueda'
39 if ($filter['searchType'] === '=') {
40     $this->filters[] = "{$filter['tableName']}.{$filter['columnName']}";
41     $this->params[$filter['columnName']] = $input[$filter['filterName']];
42 }
43 }
44 }
45 }
46 /**
47  * @param array $input Contiene el input del usuario
48  * @return array Los resultados de la lectura a la BD
49  */
50 public function read($input)
51 {
52     $statement = $this->baseQuery;
53     $this->setFilters($input);
54     if (empty($this->filters) === false) {
55         $statement .= ' AND ';
56         $statement .= implode(' AND ', $this->filters);
57     }
58     return $this->connection->query($statement, $this->params);
59 }
60 }
61 }
62 }
63 }
64 }
65 }

```

}  
 Verifica en la base de datos, la existencia de los usuarios y sus contraseñas ingresadas .

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <?php
4
5 session_start();
6
7 include 'base_path.php';
8 include 'head.php';
9 include 'basededatos/MySQL.php';
10
11 if (isset($_SESSION['id'])) {
12     header('Location: ' . BASE_PATH . '/empleados');
13 }
14
15 if (
16     empty($_POST['usuario']) === false &&
17     empty($_POST['pass']) === false
18 ) {
19     $conexion = new MySQL();
20
21     $usuario = $conexion->query('SELECT id_employado, user, pass, id cargo FROM Empleados WHERE estado = 1 AND user = ?', [$_POST['usuario']]);
22
23     if (empty($usuario[0]) === false) {
24         if ($usuario[0]['pass'] === $_POST['pass']) {
25             $_SESSION['id'] = $usuario[0]['id_employado'];
26             $_SESSION['user'] = $usuario[0]['user'];
27             $_SESSION['cargo'] = $usuario[0]['id_cargo'];
28             header('Location: ' . BASE_PATH . '/empleados');
29         }
30     }

```

Se determina si una variable está vacía mediante **empty()**, si el usuario

```
30 }
31 }
32 ?>
33 <body>
34 <?php
35 include 'navbar.php';
36 ?>
37 <?php if ((isset($_POST['usuario']) || isset($_POST['pass'])) && empty($usuario[0]
38 )): ?>
39 <div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"
40 ></span> El usuario y la contraseña no existen o no coinciden.</div>
41 <?php endif; ?>
42 <div class="container-fluid">
43 <div class="row">
44 <div class="col-sm-4 col-sm-offset-4">
45 <div class="panel panel-default">
46 <div class="panel-heading">Inicia Sesión</div>
47 <div class="panel-body">
48 <form action="index.php" method="post">
49 <div class="form-group">
50 <label for="usuario">Usuario</label>
51 <input type="text" name="usuario" value="" id="usuario"
52 class="form-control">
53 </div>
54 <div class="form-group">
55 <label for="pass">Contraseña</label>
56 <input type="password" name="pass" value="" id="pass"
57 class="form-control">
58 </div>
59 <input type="submit" value="Ingresar" class="btn
60 btn-block btn-success">
61 </form>
62 </div>
63 </div>
64 </div>
65 </body>
66 </html>
```

## CARPETA EMPLEADO

En *index.php*:

Se hace la conexión a la base de datos y se consulta la cantidad de empleados que existen activos

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <?php
4 include '../base_path.php';
5
6 session_start();
7 if (isset($_SESSION['id']) === false) {
8 header("Location: ' . BASE_PATH . '/'");
9 die();
10 }
11
12 include '../head.php';
13 include '../basededatos/MySQL.php';
14 include '../basededatos/MySQLReader.php';
15
16 // Obtener la lista de empleados desde la base de datos
17 $conexion = new MySQL();
18
19 $lecturaEmpleados = new MySQLReader();
20
21 $lecturaEmpleados->baseQuery = 'SELECT id_empleado, nombres, apellidos, DUI,
22 direccion, telefono, nombre_cargo FROM Empleados INNER JOIN Cargos USING(
23 id_cargo) WHERE estado = 1';
24 $lecturaEmpleados->filterFields = [
```

Luego se guardad un array que servira para hacer la busqueda según el campo que se deseé. ***\$lecturaEmpleados->filterFields = [ ]***

```
$lecturaEmpleados->filterFields = [
    [
        'filterName' => 'nombre',
        'tableName' => 'Empleados',
        'columnName' => 'nombres',
        'searchType' => 'LIKE',
    ],
    [
        'filterName' => 'apellido',
        'tableName' => 'Empleados',
        'columnName' => 'apellidos',
        'searchType' => 'LIKE',
    ],
    [
        'filterName' => 'dui',
        'tableName' => 'Empleados',
        'columnName' => 'DUI',
        'searchType' => 'LIKE',
    ],
    [
        'filterName' => 'telefono',
        'tableName' => 'Empleados',
        'columnName' => 'telefono',
        'searchType' => 'LIKE',
    ],
    [
        'filterName' => 'cargoEmpleado',
        'tableName' => 'Empleados',
        'columnName' => 'id_cargo',
        'searchType' => '=',
    ],
];
```

**nuevo-empleado.php**

Se hace la conexión a la base de datos mediante ***include*** ***'./basededatos/MySQL.php';***

se determina que las variables esten definidas mediante ***empty***. Luego se guardan los datos que se van a crear en la variable ***\$esta\_guardado***.

```
<?php
include '../base_path.php';

session_start();
if (isset($_SESSION['id']) === false) {
    header("Location: './BASE_PATH ./'");
    die();
}

include './basededatos/MySQL.php';

$connexion = new MySQL();

$listacargos = $conexion->query('SELECT id_cargo, nombre_cargo FROM Cargos', []);

if (
    empty($_POST['nombreEmpleado']) === false &&
    empty($_POST['apellidoEmpleado']) === false &&
    empty($_POST['duiEmpleado']) === false &&
    empty($_POST['direccionEmpleado']) === false &&
    empty($_POST['telefonoEmpleado']) === false &&
    empty($_POST['cargoEmpleado']) === false
) {
    $esta_guardado = $conexion->execute('INSERT INTO Empleados (nombres, apellidos, user,
    pass, DUI, direccion, telefono, id_cargo) VALUES (:nombre, :apellidos, :user, :pass,
    :DUI, :direccion, :telefono, :id_cargo)', [
        ':nombre' => $_POST['nombreEmpleado'],
        ':apellidos' => $_POST['apellidoEmpleado'],
        ':user' => $_POST['userEmpleado'],
        ':pass' => $_POST['passEmpleado'],
        ':DUI' => $_POST['duiEmpleado'],
        ':direccion' => $_POST['direccionEmpleado'],
        ':telefono' => $_POST['telefonoEmpleado'],
        ':id_cargo' => $_POST['cargoEmpleado'];
    ]);
}
```



Luego se comprueba si la variable, esta definida y verdadera. Para guardar los datos.

```
<?php if (isset($esta_guardado) && $esta_guardado == true): ?>
```

tambien comprueba que si algunas variables no estan definidas y se comprueba que la variable es falsa, no guarda los datos.

```
<?php if (isset($_POST['nombreEmpleado']) && (isset($esta_guardado) == false || $esta_guardado == false)): ?>
```

### *eliminar-empleado.php*

Se hace la conexión a la base de datos y se muestran los empleados, luego se realiza un **update** en el campo **estado** de la tabla **empleados**, colocando **estado = 0** para inactivar ese empleado, lo cual se guarda en la variable **\$eliminado**.

```
<?php
include '../base_path.php';
session_start();
if (isset($_SESSION['id']) === false) {
    header('Location: ' . BASE_PATH . '/');
    die();
}
include '../basededatos/MySQL.php';
if (isset($_GET['id']) === false) {
    header('Location: ' . BASE_PATH . '/empleados');
}
$conexion = new MySQL;
try {
    $eliminado = $conexion->execute('UPDATE Empleados SET estado = 0 WHERE id_empleado = ?', [$_GET['id']]);
    if ($eliminado === true) {
        header('Location: ' . BASE_PATH . '/empleados?empelim=1');
    } else {
        header('Location: ' . BASE_PATH . '/empleados/editar-empleado.php?id=' . $_GET['id'] . '&empelim=0');
    }
} catch (Exception $e) {
    header('Location: ' . BASE_PATH . '/empleados/editar-empleado.php?id=' . $_GET['id'] . '&empelim=2');
}
```

### *editar-empleado.php*

Se hace la conexión a la base de datos mediante el **include**.

Se determina si las variables están definidas mediante el **\$\_POST[' ']** y **empty**.

```
<?php
include '../base_path.php';

session_start();
if (isset($_SESSION['id']) == false) {
    header("Location: ' . BASE_PATH . '/'");
    die();
}

include '../basededatos/MySQL.php';

if (empty($_REQUEST['id']) == true) {
    header("Location: ' . BASE_PATH . '/empleados');
}

$conexion = new MySQL();

$alerta = [
    'class' => 'alert-danger',
    'message' => 'Los datos no fueron guardados.',
    'icon' => 'glyphicon-exclamation-sign'
];

if (
    empty($_POST['id']) == false &&
    empty($_POST['nombreEmpleado']) == false &&
    empty($_POST['apellidoEmpleado']) == false &&
    empty($_POST['userEmpleado']) == false &&
    empty($_POST['passEmpleado']) == false &&
    empty($_POST['duiEmpleado']) == false &&
    empty($_POST['direccionEmpleado']) == false &&
    empty($_POST['telefonoEmpleado']) == false &&
    empty($_POST['cargoEmpleado']) == false
) {
```

En este caso se hace un **UPDATE** de la tabla **Empleados**, lo cual se guarda en la variable **\$esta\_guardado**. Luego se hace una nueva consulta para que aparezcan los datos actualizados. **query('SELECT id\_empleado, nombres, apellidos, user, pass, DUI, direccion, telefono, id\_cargo FROM Empleados WHERE id\_empleado=?', [\$\_REQUEST['id']])**, guardado está en la variable **\$datosEmpleado**.

## CARPETA: RESERVACIONES

### eliminar-reserva.php

Se hace la conexión a la base de datos y se muestran los empleados, se comprueba que hay una reservación y luego se modifica su estado de “1” a “0”.

Después se realiza un **query** en la tabla **Reservas** y se elimina la reserva mediante **execute('DELETE FROM Reservas WHERE id\_reserva = ?', [\$\_GET['id']]);**

```

1 <?php
2
3 include '../base_path.php';
4
5 session_start();
6 if (isset($_SESSION['id']) == false) {
7     header('Location: ' . BASE_PATH . '/');
8     die();
9 }
10
11 include '../basededatos/MySQL.php';
12
13 if (isset($_GET['id']) == false) {
14     header('Location: ' . BASE_PATH . '/reservaciones/ver-reservaciones.php?eliminado=0');
15     die();
16 }
17
18 $conexion = new MySQL();
19
20 try {
21     $eliminado = $conexion->execute('DELETE FROM Reservas WHERE id_reserva = ?', [$_GET['id']]);
22     if ($eliminado == true) {
23         header('Location: ' . BASE_PATH . '/reservaciones/ver-reservaciones.php?eliminado=1');
24         die();
25     } else {
26         header('Location: ' . BASE_PATH . '/reservaciones/ver-reservaciones.php?eliminado=2');
27         die();
28     }
29 } catch (Exception $e) {
30     header('Location: ' . BASE_PATH . '/reservaciones/ver-reservaciones.php?eliminado=3');
31     die();
32 }
33

```

### index.php

Se hace la conexión a la base de datos, para obtener la lista de empleados.

Mediante **date\_default\_timezone\_set('America/El\_Salvador')**, se establece que el horario debe coincidir con la hora del país. Luego será utilizada para guardar y buscar las reservas.

Mediante **explode** convertimos los **String** a un **array** que luego se utilizará para generar la lista de vehículos.

```

1 <?php
2 include '../base_path.php';
3
4 session_start();
5 if (isset($_SESSION['id']) == false) {
6     header('Location: ' . BASE_PATH . '/');
7     die();
8 }
9
10 include '../basededatos/MySQL.php';
11
12 date_default_timezone_set('America/El_Salvador');
13
14
15
16 $conexion = new MySQL();
17
18
19 $listaVehiculos = [];
20
21 if (empty($_POST['fechadesde']) == false && empty($_POST['horadesde']) == false && empty($_POST['fechahasta']) == false && empty($_POST['horahasta']) == false) {
22     $fechaDesde = explode('-', $_POST['fechadesde']);
23     $horaDesde = explode(':', $_POST['horadesde']);
24
25     $fechaDesde = (new DateTime)->setDate($fechaDesde[0], $fechaDesde[1], $fechaDesde[2]);
26     $fechaDesde->setTime($horaDesde[0], $horaDesde[1]);
27
28     $fechaHasta = explode('-', $_POST['fechahasta']);
29     $horaHasta = explode(':', $_POST['horahasta']);
30
31     $fechaHasta = (new DateTime)->setDate($fechaHasta[0], $fechaHasta[1], $fechaHasta[2]);
32     $fechaHasta->setTime($horaHasta[0], $horaHasta[1]);
33
34     $listaVehiculos = $conexion->query('SELECT id_vehiculo, id_modelo, placa, anio, nombre_modelo, nombre_marca FROM Vehiculos INNER JOIN Modelos USING(id_modelo) INNER JOIN Marcas USING(id_marca) WHERE id_vehiculo NOT IN (SELECT id_vehiculo FROM Reservas WHERE fechahora_inicio = :fina and fechahora_fin > :inicio)');
35     $inicioB => $fechaDesde->format('Y-m-d H:i:s');
36     $finB => $fechaHasta->format('Y-m-d H:i:s');
37
38     );
39 }
40
41 $cantidadVehiculos = count($listaVehiculos);

```

Se muestran las reservas de vehículos realizadas, utilizando un **for** para generar los vehículos disponibles.

```

</div>
<div class="row">
<?php
for ($i = 1; $i <= $cantidadVehiculos; $i++):
$vehiculo = $listaVehiculos[$i - 1];
?>
<div class="col-sm-3">
<div class="thumbnail">
" />
<div class="caption">
<h3><?php echo $vehiculo['nombre_marca'], ' ', $vehiculo['nombre_modelo'] ?></h3>
<ul>
<li>Año: <?php echo $vehiculo['anio'] ?></li>
<li>Placa: <?php echo $vehiculo['placa'] ?></li>
</ul>
<button type="button" class="btn-reservar btn btn-block btn-success" data-idvehiculo="<?php echo $vehiculo['id_vehiculo'] ?>"><span class="glyphicon glyphicon-book"></span> Reservar</button>
</div>
</div>
</div>
<?php
if ($i % 4 == 0) {
echo '</div>';
echo '<div class="row">';
}
endfor;
?>
</div>
</div>
<script type="text/javascript">
$(function(){
$('.btn-reservar').on('click', function () {
$('#fechadesde').val($('#fechadesdedisponible').val());
$('#horadesde').val($('#horadesdedisponible').val());
$('#fechahasta').val($('#fechahastadisponible').val());
$('#horahasta').val($('#horahastadisponible').val());
$('#idvehiculo').val($(this).data('idvehiculo'));
$('#formulario-reserva').submit();
});
});
</script>
</body>
</html>

```

### Registrar-reserva.php

Se comprueba que estén definidas las variables por el método **\$\_POST**, luego se muestra los vehículos disponibles.

Mediante **explode** convertimos los **String** a un **array** "-", que luego se utilizará para generar la lista de vehículos.

```

<?php
include '../base_path.php';
session_start();
if (!isset($_SESSION['id']) == false) {
header('Location: ' . BASE_PATH . '/');
die();
}
if (
empty($_POST['fechadesde']) == true ||
empty($_POST['horadesde']) == true ||
empty($_POST['fechahasta']) == true ||
empty($_POST['horahasta']) == true ||
empty($_POST['idvehiculo']) == true
) {
header('Location: ' . BASE_PATH . '/reservaciones/index.php?reservado=0');
die();
}
$fechaDesde = explode('-', $_POST['fechadesde']);
$horaDesde = explode(':', $_POST['horadesde']);
$fechaDesde = (new DateTime)->setDate($fechaDesde[0], $fechaDesde[1], $fechaDesde[2]);
$fechaDesde->setTime($horaDesde[0], $horaDesde[1]);
$fechaHasta = explode('-', $_POST['fechahasta']);
$horaHasta = explode(':', $_POST['horahasta']);
$fechaHasta = (new DateTime)->setDate($fechaHasta[0], $fechaHasta[1], $fechaHasta[2]);
$fechaHasta->setTime($horaHasta[0], $horaHasta[1]);
if ($fechaDesde->getTimestamp() >= $fechaHasta->getTimestamp()) {
header('Location: ' . BASE_PATH . '/reservaciones/index.php?reservado=2');
die();
}
include '../basededatos/MySQL.php';
$conexion = new MySQL();
try {
$reservaGuardada = $conexion->execute('INSERT INTO Reservas(fechahora_inicio, fechahora_fin, id_vehiculo, id_empleado) VALUES

```

Se hace la conexión a la base de datos y luego se utiliza un **INSERT**, para realizar la nueva reserva, la cual se guardará en la variable **\$reservaGuardada** .

```
include '../basededatos/MySQL.php';

$conexion = new MySQL();

try {
    $reservaGuardada = $conexion->execute('INSERT INTO Reservas(fechahora_inicio, fechahora_fin, id_vehiculo, id_empleado) VALUES
    (:inicio, :fin, :vehiculo, :empleado)', [
        ':inicio' => $_POST['fechadesde'],
        ':fin' => $_POST['fechahasta'],
        ':vehiculo' => $_POST['idvehiculo'],
        ':empleado' => $_SESSION['id'],
    ]);
} catch (Exception $e) {
    header('Location: ' . BASE_PATH . '/reservaciones/index.php?reservado=3');
    die();
}

if ($reservaGuardada == true) {
    header('Location: ' . BASE_PATH . '/reservaciones/index.php?reservado=1');
    die();
} else {
    header('Location: ' . BASE_PATH . '/reservaciones/index.php?reservado=4');
    die();
}
}
```

### ver-reservaciones.php

Se inicia la sesión y se hace la conexión a la base de datos. Luego se realiza una consulta y el resultado se guarda en la variable **\$listaVehiculos**, pero mediante la comprobación con el **if**, solo se mostrará las reservas realizadas.

```
<?php
include '../base_path.php';

session_start();
if (isset($_SESSION['id']) == false) {
    header('Location: ' . BASE_PATH . '/');
    die();
}

include '../basededatos/MySQL.php';
date_default_timezone_set('America/El_Salvador');

$conexion = new MySQL();

$listavehiculos = $conexion->query('SELECT id_vehiculo, id_modelo, placa, anio, nombre_modelo, nombre_marca, id_reserva,
fechahora_inicio, fechahora_fin, nombres, apellidos, nombre_cargo FROM Reservas INNER JOIN Vehiculos USING(id_vehiculo) INNER JOIN
Modelos USING(id_modelo) INNER JOIN Marcas USING(id_marca) INNER JOIN Empleados USING(id_empleado) INNER JOIN Cargos USING (
id_cargo)', []);

if (empty($_POST['fechadesde']) == false && empty($_POST['horadesde']) == false && empty($_POST['fechahasta']) == false && empty($_POST['horahasta']) == false) {
    $fechaDesde = explode('-', $_POST['fechadesde']);
    $horaDesde = explode(':', $_POST['horadesde']);

    $fechaDesde = (new DateTime)->setDate($fechaDesde[0], $fechaDesde[1], $fechaDesde[2]);
    $horaDesde->setTime($horaDesde[0], $horaDesde[1]);

    $fechaHasta = explode('-', $_POST['fechahasta']);
    $horaHasta = explode(':', $_POST['horahasta']);

    $fechaHasta = (new DateTime)->setDate($fechaHasta[0], $fechaHasta[1], $fechaHasta[2]);
    $horaHasta->setTime($horaHasta[0], $horaHasta[1]);

    $listavehiculos = $conexion->query('SELECT id_vehiculo, id_modelo, placa, anio, nombre_modelo, nombre_marca, id_reserva,
fechahora_inicio, fechahora_fin, nombres, apellidos, nombre_cargo FROM Reservas INNER JOIN Vehiculos USING(id_vehiculo) INNER JOIN
Modelos USING(id_modelo) INNER JOIN Marcas USING(id_marca) INNER JOIN Empleados USING(id_empleado) INNER JOIN Cargos
USING (id_cargo) WHERE fechahora_inicio < :finB and fechahora_fin > :inicioB', [
        ':inicioB' => $fechaDesde->format('Y-m-d H:i:s'),
        ':finB' => $fechaHasta->format('Y-m-d H:i:s'),
    ]);
}

$cantidadvehiculos = count($listavehiculos);
?>
```

Mediante `<?php if (isset($_GET['eliminado'])): ?>` y `<?php if ($_GET['eliminado'] == 1): ?>`, Se comprueba y elimina una reservación previamente realizada.

```

?>
<!DOCTYPE html>
<html lang="es">
<?php
include '../head.php';
?>
<body>
<?php
include '../navbar.php';
?>
<?php if (isset($_GET['eliminado'])): ?>
<?php if ($_GET['eliminado'] == 1): ?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> La reserva se eliminó exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo eliminar la reservación.
</div>
<?php endif; ?>
<?php endif; ?>
<div class="container-fluid">
<div class="row">
<div class="well clearfix">
<form class="form-inline" action="ver-reservaciones.php" method="post">
<div class="pull-left">
<div class="form-group">
<label for="fechadesde">Desde:</label>
<input placeholder="Fecha" type="date" name="fechadesde" value="<?php echo empty($_POST['fechadesde']) ==
false? $_POST['fechadesde']: ''; ?>" id="fechadesde" class="form-control">
<input placeholder="Hora" type="time" name="horadesde" value="<?php echo empty($_POST['horadesde']) ==
false? $_POST['horadesde']: ''; ?>" class="form-control">
</div>
<div class="form-group">
<label for="fechahasta">Hasta:</label>
<input placeholder="Fecha" type="date" name="fechahasta" value="<?php echo empty($_POST['fechahasta']) ==
false? $_POST['fechahasta']: ''; ?>" id="fechahasta" class="form-control">
<input placeholder="Hora" type="time" name="horahasta" value="<?php echo empty($_POST['horahasta']) ==
false? $_POST['horahasta']: ''; ?>" class="form-control">
</div>
</div>
<div class="pull-right">
<button type="submit" class="btn btn-success"><span class="glyphicon glyphicon-refresh"></span> Mostrar
Registros</button>
</div>
</form>
</div>
</div>
<div class="row">
?>

```

Mediante el `for` se genera una lista de los vehículos reservados, mostrando la marca, año y la placa. Además el nombre del empleado que ha realizado cada una de las reservas.

Por medio de `<a href="eliminar-reserva.php?id=<?php echo $vehiculo['id_reserva'] ?>"` se podrá eliminar las reservas realizadas, mediante un botón que modificará el `id`.

```

</div>
<div class="row">
<?php
for ($i = 1; $i <= $cantidadVehiculos; $i++):
$vehiculo = $listaVehiculos[$i - 1];
?>
<div class="col-sm-3">
<div class="thumbnail">
<?php echo $vehiculo['id_modelo'] ?>.jpg alt="<?php echo $vehiculo['nombre_modelo'] ?>" />
<div class="caption">
<h3><?php echo $vehiculo['nombre_marca'], ' ', $vehiculo['nombre_modelo'] ?></h3>
<ul>
<li>Año: <?php echo $vehiculo['anio'] ?></li>
<li>Placa: <?php echo $vehiculo['placa'] ?></li>
<li><?php echo $vehiculo['nombres'], ' ', $vehiculo['apellidos'], ' (' , $vehiculo['nombre_cargo'], ') ' ?></li>
<li>Desde: <?php echo $vehiculo['fechahora_inicio'] ?></li>
<li>Hasta: <?php echo $vehiculo['fechahora_fin'] ?></li>
</ul>
<a href="eliminar-reserva.php?id=<?php echo $vehiculo['id_reserva'] ?>" class="btn btn-block btn-danger"><span
class="glyphicon glyphicon-trash"></span> Eliminar</a>
</div>
</div>
</div>
<?php
if ($i % 4 == 0) {
echo </div>;
echo <div class="row">;
}
endfor;
?>
</div>
</body>
</html>

```

## CARPETA: VEHICULOS

### crear-marca.php

Se inicia la sesión, después define mediante el método **POST** el nombre de la nueva marca que vamos utilizar.

Se hace una conexión a la base de datos, para realizar un **INSERT** en la tabla **Marcas**, lo cual se guardará en la variable **\$marcaGuardada**. Después se comprueba mediante el if-else si la marca fue creada. Una vez creada ya podrá ser utilizada.

```
<?php
include '../base_path.php';

session_start();
if (isset($_SESSION['id']) == false) {
    header('Location: '.. BASE_PATH . '/');
    die();
}

if (empty($_POST['nombremarca']) == true) {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmarca=0');
    die();
}

include '../basedatos/MySQL.php';
$conexion = new MySQL();

try {
    $marcaGuardada = $conexion->execute("INSERT INTO Marcas(nombre_marca) VALUES(?)", [$_POST['nombremarca']]);
} catch (Exception $e) {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmarca=0');
    die();
}

if ($marcaGuardada == true) {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmarca=1');
    die();
} else {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmarca=0');
    die();
}
}
```

### crear-modelo.php

Se inicia sesión, luego se comprueba que mediante el método **POST** estén definidos **nombreModelo** y **marca**. Se verifica que se cargue la imagen que vamos a utilizar para el vehículo mediante **getimagesize(\$\_FILES['imgVehiculo']['tmp\_name']) === false || pathinfo(\$\_FILES['imgVehiculo']['name'],PATHINFO\_EXTENSION) !== 'jpg'**, para lo cual solo se aceptaran imágenes con extensiones .jpg.

Se realiza un **INSERT** a la tabla **Modelos**, guardando este en la variable **\$modeloGuardado**

si la variable **\$modeloGuardado** es verdadera, se guardará una imagen que estará almacenada en la variable **\$ultimaID**, utilizando el id con la que fue creada.

```
<?php
include '../base_path.php';

session_start();
if (isset($_SESSION['id']) == false) {
    header('Location: '.. BASE_PATH . '/');
    die();
}

if (
    empty($_POST['nombreModelo']) == true ||
    empty($_POST['marca']) == true ||
    getimagesize($_FILES['imgVehiculo']['tmp_name']) == false ||
    pathinfo($_FILES['imgVehiculo']['name'],PATHINFO_EXTENSION) !== 'jpg'
) {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmodelo=0');
    die();
}

include '../basedatos/MySQL.php';
$conexion = new MySQL();

try {
    $modeloGuardado = $conexion->execute("INSERT INTO Modelos(nombre_modelo, id_marca) VALUES(:nombre, :marca)", [
        ':nombre' => $_POST['nombreModelo'],
        ':marca' => $_POST['marca'],
    ]);
} catch (Exception $e) {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmodelo=0');
    die();
}

if ($modeloGuardado == true) {
    $ultimaID = (string) $conexion->connection->lastInsertId();
    move_uploaded_file($_FILES['imgVehiculo']['tmp_name'], '../img/' . $ultimaID . '.jpg');
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmodelo=1');
    die();
} else {
    header('Location: '.. BASE_PATH . '/vehiculos/nuevo-vehiculo.php?newmodelo=0');
    die();
}
}
```

## crear-vehiculo.php

Se inicia la sesión, y se comprueba que estén definidos mediante el método **POST: placavehiculo,anioVehiculo,modeloVehiculo**.

Se realiza la conexión a la base de datos y se guarda en la variable **\$carroGuardado**, un **INSERT** de **placa,anio y modelo** tabla **Vehículos**. Luego se verifica si el vehículo esta creado, si no lo esta se creará el nuevo vehículo.

```
<?php
include '../base_path.php';

session_start();
if (!isset($_SESSION['id']) === false) {
    header('Location: ' . BASE_PATH . '/');
    die();
}

if (
    empty($_POST['placavehiculo']) === true ||
    empty($_POST['anioVehiculo']) === true ||
    empty($_POST['modeloVehiculo']) === true
) {
    header('Location: ' . BASE_PATH . '/vehiculos/nuevo-vehiculo.php?vehcreated=0');
    die();
}

include '../basededatos/MySQL.php';
$connexion = new MySQL();

try {
    $carroGuardado = $conexion->execute('INSERT INTO Vehiculos (placa, anio, id_modelo) VALUES(:placa, :anio, :modelo)', [
        ':placa' => $_POST['placavehiculo'],
        ':anio' => $_POST['anioVehiculo'],
        ':modelo' => $_POST['modeloVehiculo'],
    ]);
} catch (Exception $e) {
    header('Location: ' . BASE_PATH . '/vehiculos/nuevo-vehiculo.php?vehcreated=0');
    die();
}

if ($carroGuardado == true) {
    header('Location: ' . BASE_PATH . '/vehiculos/nuevo-vehiculo.php?vehcreated=1');
    die();
} else {
    header('Location: ' . BASE_PATH . '/vehiculos/nuevo-vehiculo.php?vehcreated=0');
    die();
}
```

## Index.php

Se incluye la conexión a la base de datos, luego se guarda en la variable **\$listaVehiculos** la consulta realizada en la tabla **Vehículos**. En la variable **\$cantidadVehiculos** se guarda la lista de los vehículos disponibles.

Mediante el **for** se muestran todos los vehículos que fueron creados.

```
include '../basededatos/MySQL.php';

$connexion = new MySQL();

$listasVehiculos = $conexion->query('SELECT id_vehiculo, id_modelo, placa, anio, nombre_modelo, nombre_marca FROM Vehiculos INNER JOIN
Modelos USING(id_modelo) INNER JOIN Marcas USING(id_marca)', []);
$cantidadVehiculos = count($listasVehiculos);
?>
<!DOCTYPE html>
<html lang="en">
<?php
include '../head.php';
?>
<body>
<?php
include '../navbar.php';
?>
<div class="container-fluid">
<div class="row">
<?php
for ($i = 1; $i <= $cantidadVehiculos; $i++) {
    $vehiculo = $listasVehiculos[$i - 1];
?>
<div class="col-sm-3">
<div class="thumbnail">
.jpg" alt="<?php echo $vehiculo['nombre_modelo'] ?>" />
<div class="caption">
<h3><?php echo $vehiculo['nombre_marca'], ' ', $vehiculo['nombre_modelo'] ?></h3>
<ul>
<li>ID: <?php echo $vehiculo['id_vehiculo'] ?></li>
<li>Año: <?php echo $vehiculo['anio'] ?></li>
<li>Placa: <?php echo $vehiculo['placa'] ?></li>
</ul>
</div>
</div>
</div>
<?php
if ($i % 4 == 0) {
    echo "</div>";
    echo "<div class="row">";
}
endfor;
?>
</div>
</div>
```



### nuevo-vehiculo.php

Se hace una conexión a la base de datos, luego se guarda en la variable **\$listaMarcas** , la consulta que muestras las marcas creadas y en la variable **\$listaModelos**, la consulta que muestra los modelos creados

```
<?php
include '../base_path.php';

session_start();
if (isset($_SESSION['id']) === false) {
    header('Location: ' . BASE_PATH . '/');
    die();
}

include '../basedatos/MySQL.php';

$conection = new MySQL();

$listMarcas = $conection->query('SELECT id_marca, nombre_marca FROM Marcas', []);

$listModelos = $conection->query('SELECT id_modelo, nombre_modelo, nombre_marca FROM Modelos INNER JOIN Marcas USING(id_marca)', []);

?>
<!DOCTYPE html>
<html lang="en">
<?php
include '../head.php';
?>
<body>
<?php
include '../navbar.php';

if (isset($_GET['vehcreated']) === true):
if ($_GET['vehcreated'] == 1):
?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> Se creó el vehículo exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo crear un nuevo vehículo.</div>
<?php
endif;
endif;
?>

<?php
if (isset($_GET['newmarca']) === true):
if ($_GET['newmarca'] == 1):
?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> Se creó la marca exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo crear una nueva marca.</div>
<?php
endif;
endif;
?>

<?php
if (isset($_GET['newmodelo']) === true):
if ($_GET['newmodelo'] == 1):
?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> Se creó el modelo exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo crear un nuevo modelo.</div>
<?php
endif;
endif;
?>

<div class="container-fluid">
<div class="row">
<div class="col-sm-6 col-sm-offset-3">
```

Se define mediante el metodo GET **if (isset(\$\_GET['newmarca']) === true): if (\$\_GET['newmarca'] == 1)**, el nombre de la nueva marca a crear. De igual manera se utiliza **if (isset(\$\_GET['newmodelo']) === true): if (\$\_GET['newmodelo'] == 1)**, para agregar el nombre del nuevo modelo a crear.

```
?>
<!DOCTYPE html>
<html lang="en">
<?php
include '../head.php';
?>
<body>
<?php
include '../navbar.php';

if (isset($_GET['vehcreated']) === true):
if ($_GET['vehcreated'] == 1):
?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> Se creó el vehículo exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo crear un nuevo vehículo.</div>
<?php
endif;
endif;
?>

<?php
if (isset($_GET['newmarca']) === true):
if ($_GET['newmarca'] == 1):
?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> Se creó la marca exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo crear una nueva marca.</div>
<?php
endif;
endif;
?>

<?php
if (isset($_GET['newmodelo']) === true):
if ($_GET['newmodelo'] == 1):
?>
<div class="alert alert-success"><span class="glyphicon glyphicon-ok-sign"></span> Se creó el modelo exitosamente!</div>
<?php else: ?>
<div class="alert alert-danger"><span class="glyphicon glyphicon-exclamation-sign"></span> No se pudo crear un nuevo modelo.</div>
<?php
endif;
endif;
?>

<div class="container-fluid">
<div class="row">
<div class="col-sm-6 col-sm-offset-3">
```

Para agregar la nueva marca que vamos a crear se tiene que elegir la lista donde se guardará mediante **<?php foreach (\$listaMarcas as \$marca): ?>** este toma las variables de **“crear-marca.php”**.

```

<div class="panel panel-primary">
  <div class="panel-heading">Crear Nuevo Vehículo</div>
  <div class="panel-body">
    <form action="crear-vehiculo.php" method="post" enctype="multipart/form-data">
      <div class="form-group">
        <label>Placa:</label>
        <input type="text" name="placavehiculo" value="" class="form-control" placeholder="Placa del vehiculo">
      </div>
      <div class="form-group">
        <label>Año:</label>
        <input type="number" name="anioVehiculo" value="<?php echo (new DateTime())->format('Y'); ?>" class="form-control">
      </div>
      <div class="form-group">
        <label for="modeloVehiculo">Selecciona la Modelo:</label>
        <select class="form-control" id="modeloVehiculo" name="modeloVehiculo">
          <?php foreach ($listaModelos as $modelo): ?>
            <option value="<?php echo $modelo['id_modelo'] ?>"><?php echo $modelo['nombre_marca'], ' ', $modelo['nombre_modelo'] ?></option>
          </select>
      </div>
      <input type="submit" value="Crear Vehículo" class="btn btn-block btn-info">
    </form>
  </div>
</div>
</div>
<div class="row">
  <div class="col-sm-6 col-sm-offset-3">
    <div class="panel panel-primary">
      <div class="panel-heading">Crear Nueva Marca</div>
      <div class="panel-body">
        <form action="crear-marca.php" method="post" enctype="multipart/form-data">
          <div class="form-group">
            <label>Nombre:</label>
            <input type="text" name="nombremarca" value="" class="form-control" placeholder="Nombre de la marca">
            <datalist id="marca">
              <?php foreach ($listaMarcas as $marca): ?>
                <option><?php echo $marca['nombre_marca'] ?></option>
              <?php endforeach; ?>
            </datalist>
          </div>
          <input type="submit" value="Crear Marca" class="btn btn-block btn-info">
        </form>
      </div>
    </div>
  </div>
</div>

```

Creamos el nombre del nuevo modelo utilizando **<?php foreach (\$listaModelos as \$modelo): ?>**

**<option><?php echo \$modelo['nombre\_modelo'] ?></option>**, el cual agregará a la lista de modelos disponibles, eligiendo su marca previamente creada mediante **<?php foreach (\$listaMarcas as \$marca): ?><option value="<?php echo \$marca['id\_marca'] ?>">**

**<?php echo \$marca['nombre\_marca'] ?></option>**. Este formulario realizará su action en **"crear-modelo.php"**.

```

</div>
</div>
<div class="row">
  <div class="col-sm-6 col-sm-offset-3">
    <div class="panel panel-primary">
      <div class="panel-heading">Crear Nuevo Modelo</div>
      <div class="panel-body">
        <form action="crear-modelo.php" method="post" enctype="multipart/form-data">
          <div class="form-group">
            <label>Nombre:</label>
            <input type="text" name="nombreModelo" value="" class="form-control" placeholder="Nombre del modelo">
            <datalist id="modelo">
              <?php foreach ($listaModelos as $modelo): ?>
                <option><?php echo $modelo['nombre_modelo'] ?></option>
              <?php endforeach; ?>
            </datalist>
          </div>
          <div class="form-group">
            <label>Imagen:</label>
            <input type="file" name="imgVehiculo" class="form-control">
          </div>
          <div class="form-group">
            <label for="marca">Selecciona la Marca:</label>
            <select class="form-control" id="marca" name="marca">
              <?php foreach ($listaMarcas as $marca): ?>
                <option value="<?php echo $marca['id_marca'] ?>"><?php echo $marca['nombre_marca'] ?></option>
              <?php endforeach; ?>
            </select>
          </div>
          <input type="submit" value="Crear Modelo" class="btn btn-block btn-info">
        </form>
      </div>
    </div>
  </div>
</div>
</div>

```

## CONCLUSIÓN

El sistema de reserva de vehículos realizado por nuestro grupo servirá de útil herramienta para llevar un control y registro del uso de la flota vehicular de cualquier empresa, dicha herramienta está estructurada de forma comprensible y funcional para las tareas cotidianas de la persona que deban usarla; facilitando el trabajo a la persona y optimizando el uso de la tecnología al contar con registro de reservas y uso de vehículos en digital.

Con la creación del software, reforzaremos los conocimientos adquiridos en la materia de programación 2, utilizando el lenguaje PHP, el gestor de bases de datos MYSQL, las librerías Bootstrap, JQUERY y Dompdf.