

Alumno: DNI:
Nº de páginas (incluida ésta)

Una empresa de vehículos de alquiler dispone de una BBDD MySQL a la que se accederá desde una interfaz web realizada en **PHP, Laravel y Ajax**.

Dispone de una tabla llamada **vehículo** en la que se registran los vehículos que se pueden alquilar. La tabla almacena, entre otros, los siguientes campos: **IDVEHICULO** (PK), **MARCA**, **MODELO**, Incluye además una tabla **alquiler** en la que se registran los alquileres efectuados que incluye, entre otros, los siguientes campos: **IDALQUILER** (PK), **FECHA_INICIO**, **FECHA_FIN**, **IDVEHICULO** (FK), **IMPORTE_TOTAL**, **IDCLIENTE** (FK). La tabla **cliente** incluye los campos **IDCLIENTE** (PK), **NIF**, **APELLIDOS**, **NOMBRE**.

Necesitan una interfaz web que permita, entre otras cosas, **realizar consultas de alquileres realizados de un determinado vehículo**:

- Una interfaz **Ajax** que solicite el **identificador** de un vehículo, haga la solicitud al servidor y muestre al usuario en forma de tabla los siguientes datos: **IDALQUILER**, **FECHA_INICIO**, **FECHA_FIN**, **MARCA**, **MODELO**, **NIF**, **IMPORTE_TOTAL**. Para ello emplearemos **PHP** con tecnología **PDO** en el lado del servidor y **XHTML+Javascript** en el lado del cliente.
- Una ruta **Laravel** que haciendo uso de un **controlador** sencillo y una **vista**, muestre el mismo listado. No es necesario crear el formulario que solicita el **IDVehículo**. Es suficiente con la **ruta** y un **parámetro**.

Para homogeneizar las soluciones, la estructura debe ser **OBLIGATORIAMENTE** la siguiente:

- a) Para la consulta, la página de servidor se debe llamar `consulta.php`. La parte cliente incluirá únicamente **XHTML** y **Javascript** en un formulario y se llamará `consulta.html`. El formulario contendrá **únicamente los campos requeridos** y un **botón** que es el que se encargará de llamar en **segundo plano** a la página de servidor para recuperar los datos de la consulta. Los datos recibidos se mostrarán en una **tabla**. Los datos pueden recibirse **tanto en formato XML como TEXTO (se permite enviar directamente la tabla montada desde el servidor y mostrarla en un contenedor HTML)**.
- b) Para **Laravel**, la ruta será `alquileres/{Id_vehículo}`. El controlador se denominará `AlquileresController` que incluirá un único método `VerIdVehículo`. La vista se denominará `alquileres.ver`. **Puede usar, o no, Blade**. El listado puede mostrarse en modo **tabla** o como un simple **listado de varias líneas separadas por espacios**. Como **modelo**, se usará la clase estática `DB` proporcionada por **Laravel** y el método `select`.

Para simplificar todo lo relacionado con la conexión de la BBDD mediante **PDO** se supone que el servidor **MySQL** se encuentra en **localhost**, la BBDD se llama **alquileres** y el usuario **'root'**, sin contraseña.

No es necesario incluir CSS ni código orientado a mejorar la visualización de los contenidos. Centraos únicamente en la **FUNCIONALIDAD** especificada.

Puntuación:

- Página **PHP** `consulta.php` para interfaz **Ajax**: **3.5 ptos**.
- Página **XHTML** `consulta.html` con código **Ajax** y formulario para solicitar datos de consulta y mostrar datos: **3 ptos**.
- Ruta, Vista y Controlador **Laravel**: **3.5 ptos**.

SOLUCIÓN

Apartado a)

Consulta.html

- Formulario: un campo para el **IDVEHICULO** y un **botón** para enviar

```
<form name="consulta" method="get" action="" onsubmit="return  
enviarConsulta()">
```

```
IdVehiculo: <input type="text" name="IdVehiculo"><br/>
```

```
<input type="submit" value="enviar">
```

```
</form>
```

- Capa para mostrar los datos: <div id="CapaDatos">Aquí vendrán los datos</div>

- Incluir código Ajax:

- o AjaxCrearObjeto(): Podemos usar exactamente la misma de las diapositivas
- o Funcion LeerDatos()

- Evento: **onclick** del botón o **onSubmit** del formulario

- o Llamamos a Consulta.php y pasamos (por GET o POST) el IDVehiculo seleccionado en el control de formulario:

```
function enviarConsulta(){  
oXML.open('GET', 'consulta.php?IdVehiculo=' + document.forms.consulta.IdVehicul  
o.Value, true);  
oXML.send('');  
return true;  
}
```

En la función leerDatos(): procesamos archivo XML con los campos recibidos del servidor o, si hemos enviado directamente una **tabla o capa**, cambiamos el **innerHTML** de la capa en la que los mostraremos.

```
function leerDatos(){  
if (oXML.readyState == 4 && oXML.status = 200) {  
...  
// Accedemos al DIV con getElementById()  
capaDatos = document.getElementById('CapaDatos');  
capaDatos.innerHTML = oXML.responseText;  
...  
}  
}
```

Consulta.php

- Código para conectar con BBDD usando PDO: Conexion.php

```
<?php
```

```
function dbConnect(){  
$conn = null;  
$host = 'localhost';  
$db = alquileres;  
$user = 'root';  
$pwd = '';  
try {  
$conn = new PDO('mysql:host='.$host.';dbname='.$db, $user, $pwd);  
//echo 'Conectado con exito.<br>';  
}  
catch (PDOException $e) {  
echo '<p>No se puede conectar con la base de datos !!</p>';  
}
```

```
        exit;
    }
    return $conn;
}
$conn = dbConnect();
?>
```

- Código para recuperar los datos de la consulta:

```
include("conexion.php");
//Recuperamos el IdVehiculo para montar la condición del query
$IdVehiculo = $_GET['IdVehiculo'];
// Creamos la consulta
$sql = "..."; //Definido más abajo
// Creamos la consulta y asignamos el resultado a la variable $result
$result = $conn->query($sql);
// Extraemos valores de $result y los guardamos en array asociativo
$rows = $result->fetchAll();
//Generamos una cadena con la tabla montada
$res = "<table><tr><th>IdAlquiler</th><th>FechaInicio</th>...</tr>";
// Usamos foreach para extraer los valores de $rows
foreach ($rows as $row) {
    $IdAlquiler = $row['IDALQUILER'];
    $FechaInicio = $row['FECHA_INICIO'];
    ...
    $ImporteTotal = $row['IMPORTE_TOTAL'];

    //Componemos una fila de la tabla por registro
    $res = $res . "<tr><td>$IdAlquiler</td>"; //También $res .= ...
    $res = $res . "<td>$FechaInicio</td>";
    ...
    $res = $res . "<td>$ImporteTotal</td></tr>";
}
$res = $res . "</table>";
echo $res;
```

- Query BBDD:

```
$sql = "select a.IDALQUILER, a.FECHA_INICIO, a.FECHA_FIN, v.MARCA, v.MODELO,
c.NIF, a.IMPORTE_TOTAL
from alquiler a join vehiculo v on (a.IDVEHICULO = v.IDVEHICULO)
        join cliente c on (a.IDCLIENTE = c.IDCLIENTE))
where a.IDVEHICULO = $IdVehiculo"
```

Apartado b)

Ruta

routes/web.php

```
Route::get('alquileres/{Id_Vehículo}', 'AlquileresController@VerIdVehiculo');
```

Controlador

app/Http/Controllers

```
<?php
    namespace App\Http\Controllers;
    use App\Http\Controllers\Controller;
    use DB;

    class AlquileresController extends Controller{
        public function VerIdVehiculo($IdVehiculo){
            $Query = "select a.IDALQUILER, a.FECHA_INICIO, a.FECHA_FIN, v.MARCA,
v.MODELO, c.NIF, a.IMPORTE_TOTAL from alquiler a join vehiculo v on
(a.IDVEHICULO = v.IDVEHICULO join cliente c on (a.IDCLIENTE = c.IDCLIENTE))
where a.IDVEHICULO = $IdVehiculo"

            $listado = DB::select($Query);

            return view('alquileres.ver', ['listado' => $listado]);
        }
    }
}>
```

Vista

Resources/views/alquileres/ver.blade.php

```
<!DOCTYPE html>
<html>
    <body>
        <h1>Listado de alquileres:</h1>
        <table>
            <?php
                foreach ($listado as $alquiler){
                    //montar tabla dinámica
                    echo "<tr><td>$alquiler->IDALQUILER</td>";
                    echo "<td>$alquiler->FECHA_INICIO</td>";
                    ...
                }
            ?>
        </tr>
        </table>
    </body>
</html>
```