



Universidad
Politécnica
de Cartagena

Campus
de Excelencia
Internacional

TRABAJO FINAL: RESERVA DE VUELOS CON CLIENTE Y SERVIDOR



Javier López-Mora Giménez

Francisco Alfredo moreno Urrea

Ángel Silvente Teruel

Belén Rosa Ruz



ÍNDICE

INTRODUCCIÓN	2
CÓDIGO HTML Y PHP	3
INDEX.PHP	3
MAIN.PHP	4
PRUEBA.PHP	6
INSERTAR_VUELO.PHP	8
DATABASE_CONNECTION.PHP	8
BYE.PHP	8
COMPANY.PHP	9
FETCH_DATA.PHP	10
HEADER.PHP	10
INICIO_SESION.PHP	11
REGISTRAR_USUARIO.PHP	11
SUBIR.PHP	12
CÓDIGO JAVASCRIPT	13
ALL.JS	13
ANIME.JS	13
BOOTSTRAP.BUNDLE.JS	13
CONTROLLER2.JS	14
CÓDIGO CSS	19
ALL.CSS	19
BOOTSTRAP.CSS	19
CUSTOM.CSS	19



INTRODUCCIÓN

En este proyecto hemos desarrollado la funcionalidad de nuestro trabajo anterior, una página de reservas de vuelos, para que pase de ser una aplicación del tipo cliente/servidor.

La parte del cliente sería la encargada de recopilar los datos de nuestros clientes, el vuelo que quieren, y la clase de vuelo. Esa información se recopila para después enviarla al servidor.

La parte del servidor obtiene la información del cliente, la compara y le devuelve al cliente una lista de los vuelos que coinciden con las preferencias que nuestros clientes han especificado.

Airlines

✈ Origen:

✈ Destino:

☒ Ida y vuelta ☐ Sólo ida

📅 Ida:

👤 Número de pasajeros:

📅 Vuelta:



CÓDIGO HTML Y PHP

INDEX.PHP

En esta parte del código definimos como va a ser nuestra página. Aquí hemos definido los CSS, los documentos javascript que se van a implementar en nuestra página, y los controladores que usamos.

También hemos implementado aquí el fondo de nuestra página de presentación, que es la que se encuentran nuestros clientes nada más entrar a la página. Por otra parte, el código usado en esta parte es:

```
1 <!DOCTYPE html>
2 <html ng-app="trabajo" lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6
7   <link rel="stylesheet" type="text/css"
8     href="css/bootstrap.css">
9   <link href="css/all.css" rel="stylesheet">
10  <link href="css/custom.css" rel="stylesheet">
11
12   <title>Vuelos</title>
13 </head>
14 <body ng-controller="ctrl" style="background-image:
15   url('images/plane.jpg'); background-size: cover;">
16
17   <div ng-view></div>
18
19   <script src="https://code.jquery.com/jquery-
20     3.2.1.slim.min.js" integrity="sha384-
21     KJ3o2DKtIkvYIK3UEENzmM7KcRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG
22     5KkN" crossorigin="anonymous"></script>
23   <script src="js/bootstrap.bundle.js"></script>
24   <script src="js/all.js"></script>
25   <script
26     src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/a
27     ngular.min.js"></script>
28   <script
29     src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/a
30     ngular-route.js"></script>
31   <script
32     src="https://cdnjs.cloudflare.com/ajax/libs/animejs/2.0.2/an
33     ime.min.js"></script>
34   <script src="controller2.js"></script>
35   <script src="js/anime.js"></script>
36
37 </body>
38 </html>
```

En esta parte introducimos los CSS

En esta parte hemos definido la imagen que hemos usado para el fondo.

En esta parte hemos añadido los documentos javascript que vamos a usar, también añadimos el controlador que usaremos y otros documentos útiles para nuestra página.



MAIN.PHP

En esta parte hemos implementado la página principal de nuestra página de vuelos. Aquí, los clientes, pueden registrarse o identificarse en la página, además de realizar una búsqueda sin registrarse.

```
14         <h4 class="modal-title">¿Ya tienes  
        cuenta?</h4>  
15     |  
16     </div>  
17     <!-- Modal body -->  
18     <div class="modal-body" >  
19         <form class="container" name="registro"  
        action="php/inicio_sesion.php"  
        onsubmit="return validateForm()"  
        method="post" required  
        enctype="multipart/form-data">  
20             <div class="row">  
21                 <div class="col-md-6">  
22                     <label for = "rol">Email:  
                     </label>  
23                 </div>  
24                 <div class="form-group col-md-  
                6">  
25                     <input type="text"  
                     name="log_correo"  
                     id="log_correo"  
                     autocomplete="off"  
                     required>  
26                 </div>  
27             </div>  
28             <div class="row">  
29                 <div class="form-group col-md-  
                6">  
30                     <label for =  
                     "passwd">Contraseña:  
                     </label>  
31                 </div>  
32                 <div class="form-group col-md-  
                6">  
33                     <input type="password"  
                     name="passwd" id="passwd"  
                     autocomplete="off"  
                     required>  
34                 </div>  
35             </div>
```

En esta parte hemos puesto un pequeño cuestionario donde los usuarios registrados pueden iniciar sesión.

```
62     <!-- Modal body -->  
63     <div class="modal-body" >  
64         <form class="container" name="registro"  
        action="php/registrar_usuario.php"  
        onsubmit="return validateForm()"  
        method="post" required  
        enctype="multipart/form-data">  
65             <div class="row">  
66                 <div class="col-md-6">  
67                     <label>Tipo de usuario:  
                     </label>  
68                 </div>  
69                 <div class="form-group col-md-  
                6">  
70                     <input class="form-check-  
                     input" type="radio"  
                     name="rol" id="rol"  
                     value="cliente" ng-  
                     model="showCliente" ng-  
                     value="true" checked>  
71                     <label class="form-check-  
                     label">Cliente</label>  
72                 </div>  
73                 <div class="form-group col-md-  
                6"></div>  
74                 <div class="form-group col-md-  
                6">  
75                     <input class="form-check-  
                     input" type="radio"  
                     name="rol" id="rol"  
                     value="aerolinea" ng-  
                     model="showCliente" ng-  
                     value="false">  
76                     <label class="form-check-  
                     label">Aerolínea</label>  
77                 </div>
```

Si un usuario se quiere registrar en la página, hemos incluido un formulario donde se pueden registrar.



```
87         <div class="row">
88             <div class="form-group col-md-
89                 6">
90                 <label for =
91                     "contra">Contraseña:
92                 </label>
93             </div>
94             <div class="form-group col-md-
95                 6">
96                 <input type="password"
97                     name="contra" id="contra"
98                     autocomplete="off"
99                     required>
100             </div>
101         </div>
102         <div class="row">
103             <div class="col-md-6">
104                 <label for =
105                     "nombre">Nombre: </label>
106             </div>
107             <div class="form-group col-md-
108                 6">
109                 <input type="text"
110                     name="nombre" id="nombre"
111                     autocomplete="off"
112                     required>
113             </div>
114         </div>
115         <div class="row" ng-
116             show="showCliente">
117             <div class="col-md-6">
118                 <label for =
119                     "apellidos">Apellidos:
120                 </label>
121             </div>
122             <div class="form-group col-md-
123                 6">
124                 <input type="text"
125                     name="apellidos"
126                     id="apellidos"
127                     autocomplete="off" >
128             </div>
129         </div>
```

En esta parte los usuarios pueden introducir los datos a la hora de registrarse.

```
135         <div class="card mx-auto" style="width: max-content;
136             background-color: rgba(255, 255, 255, 0.75);">
137             <div class="card-body">
138                 <form novalidate name="f">
139                     <div class="row">
140                         <div class="form-group col-md-6">
141                             <label><i class="fas fa-plane-
142                                 departure"></i> Origen: </label>
143                             <select class="form-control" ng-
144                                 model="searchOrigin" >
145                                 <option ng-repeat="item in data
146                                     | orderBy:'origen' | unique:
147                                     'origen'" value="{{item.origen
148                                     | utf8_decode}}">{{item.origen
149                                     | utf8_decode}}</option>
150                             <option value="{{}}">selected
151                                 disabled hidden;¿Desde dónde
152                                 vamos?</option>
153                             </select>
154                         </div>
155                         <div class="form-group col-md-6">
156                             <label><i class="fas fa-plane-
157                                 arrival"></i> Destino: </label>
158                             <select class="form-control" ng-
159                                 model="searchDestiny">
160                                 <option ng-repeat="item in data
161                                     | orderBy:'destino' | unique:
162                                     'destino'" value="{{
163                                     {{item.destino | utf8_decode}}"
164                                     ng-model="searchDestiny">
165                                     {{item.destino | utf8_decode}}
166                                 </option>
167                             <option value="{{}}">selected
168                                 disabled hidden;¿A dónde vamos?
169                             </option>
170                             </select>
171                         </div>
172                     </div>
173                 </form>
174             </div>
175         </div>
```

Al igual que en el trabajo anterior, hemos incluido un cuadro para realizar una búsqueda de vuelos.

El cuadro incluye datos como el origen, el destino, la fecha de ida y vuelta o el número de pasajeros.



PRUEBA.PHP

Este código lo utilizamos tras una búsqueda de vuelos. En esta parte se nos muestran tanto los datos de los vuelos que coinciden con lo que hemos buscado, como los datos de nuestra reserva de vuelos (origen, destino, precio por billete, etc).

```
14         <h4 class="my-auto mx-auto" style="color: white"><i class="fas  
fa-edit"></i> Datos de su reserva</h4>  
15     </div>  
16 </div>  
17 <div class="card-body">  
18     <div class="row">  
19         <table class="table mb-0 bg-white">  
20             <thead class="thead-light">  
21                 <tr class="text-center">  
22                     <th></th>  
23                     <th>Origen</th>  
24                     <th>Destino</th>  
25                     <th>Fecha</th>  
26                     <th>Pasajeros</th>  
27                     <th ng-if="(precioIDA > 0) && (ida == true)" ng-  
model="Presupuesto">Precio/Persona</th>  
28                     <th ng-if="(precioIDA > 0) && (ida == true)" ng-  
model="Presupuesto">Precio</th>  
29                 </tr>  
30             </thead>  
31             <tbody>  
32                 <tr>  
33                     <th>Viaje de ida</th>  
34                     <td>{{ origen | uppercase }}</td>  
35                     <td>{{ destino | uppercase }}</td>  
36                     <td>{{ fcheckIn }}</td>  
37                     <td class="text-center">{{ nBilletes }}</td>  
38                     <td class="text-center" ng-if="(precioIDA > 0) &&  
(ida == true)" ng-model="Presupuesto">{{ precioIDA }}  
€</td>  
39                     <td class="text-center" ng-if="(precioIDA > 0) &&  
(ida == true)" ng-model="Presupuesto">{{ precioIDA *  
nBilletes }}€</td>  
40                 </tr>  
41                 <tr ng-show="fcheckOut">  
42                     <th>Viaje de vuelta</th>  
43                     <td>{{ destino | uppercase }}</td>  
44                     <td>{{ origen | uppercase }}</td>  
45                     <td>{{ fcheckOut }}</td>  
46                     <td class="text-center">{{ nBilletes }}</td>  
47                     <td class="text-center" ng-if="(precioIDA > 0) &&  
(ida == true)" ng-model="Presupuesto"><span ng-if="
```

En esta parte se le muestra al usuario los datos de los vuelos que ha reservado, junto con el presupuesto para la reserva y otros datos de sus vuelos.

```
71 <div class="row form-inline d-flex justify-content-center mt-3">  
72     <label class="mr-1">¿Desea cambiar el número de pasajeros?  
</label>  
73     <input class="form-control" style="width: 20%;" type="number"  
name="n_billetes" min="1" value="{{ nBilletes }}" ng-  
model="nBilletes" id="nBilletes" >  
74 </div>  
75 </div>  
76 <div class="card-footer">  
77     <button class="btn btn-danger float-right" type="button" ng-  
click=""><a style="color: white; text-decoration: none;"  
href="index.php"><i class="fas fa-undo"></i> Volver</a></button>  
78     <button ng-if="(precioIDA > 0) && (precioVUELTA > 0) && (vuelta ==  
true)" class="btn btn-success float-right mr-3" type="button" ng-  
click="" data-toggle="modal" data-target="#modalRegistroPasajeros">  
79         <i class="far fa-calendar-check"></i> Reservar</a>  
80     </button>  
81     <button ng-if="(precioIDA > 0) && (vuelta == false)" ng-  
hide="fcheckOut" class="btn btn-success float-right mr-3" type="button" ng-click="" data-toggle="modal" data-  
target="#modalRegistroPasajeros">  
82         <i class="far fa-calendar-check"></i> Reservar</a>  
83     </button>  
84 </div>  
85 </div>  
86 </div>  
87 <div ng-switch on="((data | filter:origen).length) && ((data |  
filter:destino).length) && ((data | filter:fcheckIn).length) && ((data |  
filter:fcheckOut).length)">  
88     <div class="alert alert-danger ng-empty mx-auto text-center mt-3"  
style="width: 70%; role="alert" ng-switch-when="0">  
89         Ups, parece que no hay vuelos programados para las fechas  
seleccionadas.  
90     </div>  
91 </div>
```

Aquí hemos incluido la opción de cambiar el número de pasajeros. También varios botones, uno para volver al formulario de búsqueda de vuelos, y los otros dos para reservar los vuelos de ida y de vuelta por separado.

En caso de que ningún vuelo coincida con nuestros parámetros de búsqueda, hemos puesto un mensaje de aviso.



```
98 <div class="card mt-5 mb-5 mx-auto" style="width: max-content;
background-color: rgba(255, 255, 255, 0.75);" ng-
if="existenBilletes.length > 0">
99 <div class="card-body">
100 <table class="table table-striped bg-white" ng-
model="tablaIDA">
101 <tr>
102 <th ng-click="setOrderIDA('vuelo')"> Vuelo de
ida</th>
103 <th ng-click="setOrderIDA('origen')"> Origen</th>
104 <th ng-click="setOrderIDA('destino')"> Destino </th>
105 <th ng-click="setOrderIDA('salida')"> Salida</th>
106 <th ng-click="setOrderIDA('llegada')"> Llegada</th>
107 <th ng-click="setOrderIDA('precio_business')">
Business</th>
108 <th ng-click="setOrderIDA('precio_optima')"> Optima
</th>
109 <th ng-click="setOrderIDA('precio_economy')">
Economy </th>
110 </tr>
111 <tr ng-repeat="item in data | filter: {'origen':
origen} | filter: {'destino': destino} | filter:
{'salida': fcheckIn} |
orderBy:selectedOrderIDA:reverse">
112 <td> {{ item.vuelo }}</td>
113 <td> {{ item.origen | utf8_decode }}</td>
114 <td> {{ item.destino | utf8_decode }} </td>
115 <td class="text-center"> {{ item.salida | date:
"HH:mm" }} </td>
116 <td class="text-center"> {{ item.llegada | date:
"HH:mm" }} </td>
117 <td class="text-center"> {{ item.precio_business }}
€</td>
118 <td class="text-center"> {{ item.precio_optima }}€
</td>
119 <td class="text-center"> {{ item.precio_economy }}€
</td>
120 <td><input class="form-check-input" type="checkbox"
ng-click="seleccionIDA(item.vuelo, item.salida)" ng-
disabled="(deshabilitarIDA() != item.vuelo &&
estadoDeshabilitarIDA)" ng-model="BuscarIDA">
```

Si se ha encontrado un vuelo que coincida con los parámetros de búsqueda, se procede a rellenar la tabla de vuelos de ida y de vuelta.

Cada vuelo de la tabla nos permite seleccionar la clase de vuelo que queremos, para poder definir el precio de nuestra reserva.

```
186 <form name="userForm" ng-submit="submitForm()">
187 <div class="modal-body">
188 <table class="table table-striped">
189 <thead>
190 <tr>
191 <th scope="col"></th>
192 <th scope="col">Nombre</th>
193 <th scope="col">Apellidos</th>
194 <th scope="col">DNI (sin letra)</th>
195 </tr>
196 </thead>
197 <tbody>
198 <tr ng-repeat="x in [].constructor(nBilletes) track
by $index">
199 <td class="align-middle" scope="col">
{{index+1}}</td>
200 <!--<tr>
201 <td class="align-middle" scope="col"></td>-->
202 <td scope="col"><input type="text" class="form-
control" required id="nombre_{{index}}"></td>
203 <td scope="col"><input type="text" class="form-
control" required id="apellidos_{{index}}">
</td>
204 <td scope="col"><input type="number"
class="form-control" required
id="dni_{{index}}> style="-moz-appearance:
textfield;"></td>
205 </tr>
206 </tbody>
207 </table>
208 </div>
209 <div class="modal-footer">
210 <button type="button" class="btn btn-secondary" data-
dismiss="modal" ng-click="">Close</button>
211 <button type="submit" class="btn btn-success"><i class="fas
fa-user-plus"></i> Registrar pasajeros</button>
```

En esta parte se nos permite registrar a los pasajeros de nuestra reserva.



INSERTAR_VUELO.PHP

Este código nos sirve para que las compañías puedan añadir vuelos a la base de datos, permite añadir los destinos, las horas, el numero máximo de pasajeros, etc.

```
1  k?php
2  try {
3      $connect = new PDO("mysql:host=localhost;dbname=lab_software", "root",
4                          "");
5      $connect->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
6      echo 'Connected to Database';
7  } catch (PDOException $e) {
8      echo $e->getMessage();
9  }
10
11
12
13  $data = json_decode(file_get_contents("php://input"));
14
15  $origen = $data->origen;
16  $destino = $data->destino;
17  $salida = $data->salida;
18  $llegada = $data->llegada;
19  $plazas_business = $data->plazas_business;
20  $plazas_optima = $data->plazas_optima;
21  $plazas_economy = $data->plazas_economy;
22  $precio_business = $data->precio_business;
23  $precio_optima = $data->precio_optima;
24  $precio_economy = $data->precio_economy;
25
26
27
28  $connect->query("INSERT INTO `vuelos` (`vuelo`,`origen`,`destino`,`
29                  `salida`,`llegada`,`plazas_business`,`plazas_optima`,`plazas_economy`,`precio_
30                  business`,`precio_optima`,`precio_economy`) VALUES ('IB-
31                  0000','" . $origen . "','" . $destino . "','" . $salida . "','" . $llegada . "'
32                  ','" . $plazas_business . "','" . $plazas_optima . "','" . $plazas_economy . "','" . $precio_
33                  business . "','" . $precio_optima . "','" . $precio_economy . "') or
34                  die(mysql_error());
35  $connect = null;
36  ?>
```

DATABASE_CONNECTION.PHP

Este código sirve para realizar la conexión con el servidor.

```
1  k?php
2
3  // database_connection.php
4
5  $connect = new PDO("mysql:host=localhost;dbname=lab_software", "root",
6                      "");
7  $connect->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
8  $connect->exec("SET CHARACTER SET utf8");
9  ?>
```

BYE.PHP

Este código elimina la cookie que hemos creado al inicio.

```
1  k?php
2
3  //DESTRUCCIÓN DE LA COOKIE ASOCIADA AL USUARIO PARA QUE RECUERDE LA
4  CONTRASEÑA
5
6  setcookie("usr_ck", 1, time()-1, "path=/");
7
8  setcookie("aero_ck", 1, time()-1, "path=/");
9  header("location: ../index.php");
10
11  //VUELTA A LA PÁGINA DE LOGIN
12  ?>
```



COMPANY.PHP

Con este código creamos la interfaz para que las compañías aéreas puedan añadir sus vuelos a la base de datos. También tenemos una tabla donde se muestran todos los vuelos de esa compañía.

```
54         white">Airlines</span>
55     </div>
56     <div class="card mx-auto" style="width: max-content;
57         background-color: rgba(255, 255, 255, 0.75);">
58         <div class="card-body">
59             <form novalidate name="f" ng-
60                 submit="submitFormCompany()">
61                 <div class="row">
62                     <div class="form-group col-md-6">
63                         <label><i class="fas fa-plane-
64                             departure"></i> Origen: </label>
65                         <select class="form-control" ng-
66                             model="searchOrigin">
67                             <option ng-repeat="item in data
68                                 | orderBy:'origen' | unique:
69                                 'origen' value="
70                                 {{item.origen}}">{{item.origen |
71                                     utf8_decode}}</option>
72                             <option value="{{{} }}" selected
73                                 disabled hidden>¿Desde dónde
74                                 vamos?</option>
75                         </select>
76                     </div>
77                     <div class="form-group col-md-6">
78                         <label><i class="fas fa-plane-
79                             arrival"></i> Destino: </label>
80                         <select class="form-control" ng-
81                             model="searchDestiny">
82                             <option ng-repeat="item in data
83                                 | orderBy:'destino' | unique:
84                                 'destino' value="
85                                 {{item.destino}}">{{item.destino |
86                                     utf8_decode}}</option>
87                             <option value="{{{} }}" selected
88                                 disabled hidden>¿A dónde vamos?
89                             </option>
90                         </select>
91                     </div>
92                 </div>
93             </div>
94         </form>
95     </div>
```

En esta parte se ha creado un formulario en el que las empresas pueden añadir los datos de los vuelos.

Aquí se añaden datos como los destinos, el número de plazas o el precio por cada clase de vuelo.

```
132 <div ng-show="showNew==2">
133     <h1 class="ml6 text-center">
134         <span class="text-wrapper">
135             <span class="letters" style="color: white">Airlines</span>
136         </span>
137     </h1>
138     <div class="card mt-5 mb-5 mx-auto" style="width: max-content;
139         background-color: rgba(255, 255, 255, 0.75);">
140         <div class="card-body">
141             <table class="table table-striped table-condensed table-
142                 hover">
143                 <thead>
144                     <tr>
145                         <th ng-click="setOrderIDA('vuelo')"> Vuelo de
146                             ida</th>
147                         <th ng-click="setOrderIDA('origen')">
148                             Origen</th>
149                         <th ng-click="setOrderIDA('destino')"> Destino
150                             </th>
151                         <th ng-click="setOrderIDA('salida')">
152                             Salida</th>
153                         <th ng-click="setOrderIDA('llegada')">
154                             Llegada</th>
155                         <th ng-click="setOrderIDA('precio_business')">
156                             Bussiness</th>
157                         <th ng-click="setOrderIDA('precio_optima')">
158                             Optima </th>
159                         <th ng-click="setOrderIDA('precio_economy')">
160                             Economy </th>
161                         <th ng-
162                             click="setOrderIDA('precio_economy')">OFER/VEND/
163                             DISP Bussiness </th>
164                         <th ng-
165                             click="setOrderIDA('precio_economy')">OFER/VEND/
166                             DISP Optima </th>
167                         <th ng-
168                             click="setOrderIDA('precio_economy')">OFER/VEND/
169                             DISP Economy</th>
170                     </tr>
171                 </thead>
```

En esta parte se crean las tablas de los vuelos de la compañía.



FETCH_DATA.PHP

Este Código se encarga de pedirle datos a la tabla de datos.

```
1 |
2 <?php
3
4 //fetch_data.php
5
6 include('database_connection.php');
7
8 $query = "SELECT * FROM vuelos";
9
10 $statement = $connect->prepare($query);
11
12 if($statement->execute())
13 {
14     while($row = $statement->fetch(PDO::FETCH_ASSOC))
15     {
16         $data[] = $row;
17     }
18     echo json_encode($data);
19 }
20
21 ?>
```

HEADER.PHP

Este Código lo usamos para comprobar las cookies que hay, y ejecuta un código u otro según el estado de las cookies.

```
1 k?php
2 echo '<nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-
3 top">
4     <a class="navbar-brand" href="#">Logo</a>
5     <ul class="navbar-nav ml-auto">';
6 if(isset($_COOKIE['usr_ck'])){
7     echo '<div>
8         //<P>Iniciado sesión como '. $_COOKIE['usr_ck'].'</p>
9         <li class="nav-item">
10             <button type="button" class="btn btn-primary" >
11                 <a id="cerrar_sesion" style ="decoration-
12 text:none;" href="php/bye.php"><b>Cerrar
13 sesión</b></a>
14             </button>
15         </li>
16     </div>';
17 }elseif(isset($_COOKIE['aero_ck'])){
18     echo '<div>
19         //<P>Iniciado sesión como '. $_COOKIE['aero_ck'].'</p>
20         <li class="nav-item">
21             <button type="button" class="btn btn-primary" >
22                 <a id="cerrar_sesion" style ="decoration-
23 text:none;" href="php/bye.php"><b>Cerrar
24 sesión</b></a>
25             </button>
26         </li>
27     </div>';
28 }
```

Se comprueba si existe una cookie existente, y según el tipo se ejecuta un código u otro.

```
24 else{
25     echo '<div>
26         <li class="nav-item" style="margin-right: 10px;">
27             <button type="button" class="btn btn-primary" data-
28 toggle="modal" data-target="#myModal" >
29                 <b>Registrar</b>
30             </button>
31         </li>
32     </div>
33     <div>
34         <li class="nav-item">
35             <button type="button" class="btn btn-primary" data-
36 toggle="modal" data-target="#myModal" >
37                 <b>Registrar</b>
38             </button>
39         </li>
40     </div>';
41 echo '</ul>';
42 </nav>';
43 }
```

Si no existe una cookie anterior se ejecuta un código que crea los botones de inicio de sesión y de registro.



INICIO_SESION.PHP

Este Código se utiliza cuando se inicia sesión. Se comprueba si el usuario está en la base de datos y que rol tiene (cliente o aerolínea), y se crea la cookie correspondiente.

```
2 <?php
3 try{
4     $correo = htmlentities($_POST['log_correo']);
5     $password = htmlentities($_POST['passwd']);
6     $cont = 0;
7     $cliente = "cliente";
8
9     include 'database_connection.php';
10
11     $sql = "SELECT * FROM usuarios WHERE correo =:login";
12     $q = $connect->prepare($sql);
13     $q->execute(array(":login"=>$correo));
14     while($r = $q->fetch(PDO::FETCH_ASSOC)){
15         if(password_verify($password, $r['passwd'])){
16             $cont++;
17         }
18     }
19     if($cont > 0){
20         $sql_2 = "SELECT rol FROM usuarios WHERE correo =:login";
21         $q_2 = $connect->prepare($sql_2);
22         $q_2->execute(array(":login"=>$correo));
23         $r_2 = $q_2->fetch(PDO::FETCH_ASSOC);
24         $rol_usuario = $r_2['rol'];
25         echo $rol_usuario;
26         if ($rol_usuario == "cliente"){
27             echo "Hola usuario";
28             setcookie("usr_ck", $correo, time()+86400, '/path=');
29             echo $_COOKIE['usr_ck'];
30             header("location: ../index.php");
31         }elseif ($rol_usuario == "aerolinea"){
32             echo "Hola aerolinea";
33             setcookie("aero_ck", $correo, time()+86400, '/path=');
34             echo $_COOKIE['aero_ck'];
35             //header("location: ../index.php");
36             header("location: ../company.php");
37         }else{
38             echo "Ninguna de las dos";
39         }
40     }
41 } catch(Exception $e){
42     die("Error " . $e->getMessage() );
43 }
```

En esta parte conectamos con la base de datos y buscamos a nuestro usuario.

Si encontramos un usuario que coincida, veremos que rol tiene y establecemos la cookie correspondiente. También definimos la redirección en caso de que sea cliente o compañía.

Si no hay una coincidencia, se manda un error.

REGISTRAR_USUARIO.PHP

Este código se usa para agregar a un usuario a la base de datos.

```
1 <?php
2
3 $correo_usuario = htmlentities(addslashes($_POST["correo"]));
4 $pass = htmlentities(addslashes($_POST["contra"]));
5 $nombre = $_POST["nombre"];
6 $apellidos = $_POST["apellidos"];
7 $tipo_usuario = $_POST["rol"];
8 //echo $tipo_usuario;
9 if ($tipo_usuario=="true"){
10     $tipo_usuario="cliente";
11 }elseif ($tipo_usuario=="aerolinea");
12 }
13 $pass_cifrado=password_hash($pass, PASSWORD_DEFAULT);
14
15 try{
16     include 'database_connection.php';
17     //Primero comprobamos que no haya ningún otro usuario con ese correo, en caso de que lo haya no deja registrarse
18     $sql = "SELECT correo FROM usuarios WHERE correo =:user";
19     $resultado = $connect->prepare($sql);
20     $resultado->execute(array(":user"=>$correo_usuario));
21     if ($resultado->rowCount() > 0){
22         echo "Baia baia hubo caquita";
23     } else {
24         $sql_2="INSERT INTO usuarios (correo, passwd, nombre, apellidos, rol)
25         VALUES (:usu, :pass, :nombre, :apellidos, :rol_usuario)";
26         $resultado=$connect->prepare($sql_2);
27         $resultado->execute(array(":usu"=>$correo_usuario,
28         ":pass"=>$pass_cifrado, ":nombre"=>$nombre,
29         ":apellidos"=>$apellidos, ":rol_usuario"=>$tipo_usuario));
30         echo $tipo_usuario;
31         echo "Baia baia lo conseguiste";
32     }
33 } catch(Exception $e){
34     die("Error " . $e -> getMessage() . $e -> getLine());
35 } finally{
36     $connect = null;
37     header("location: ../index.html");
38 }
```

Aquí estamos intentando registrar a un usuario en la base de datos.

Si existe ya un usuario igual en nuestra base de datos, mandamos un error.

Si no existe el usuario, lo incluimos en la base de datos. Tras esto, la página nos redirige a index.php.



SUBIR.PHP

Este código se utiliza para subir los datos de las reservas a la base de datos. También se encarga de actualizar el numero de plazas disponibles para un vuelo.

```
2 try {
3     $connect = new PDO("mysql:host=localhost;dbname=lab_software", "root",
4         "");
5     $connect->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
6     echo 'Connected to Database';
7 } catch (PDOException $e) {
8     echo $e->getMessage();
9 }
10 $permitted_chars = '0123456789';
11
12 function generate_string($input, $strength = 16) {
13     $input_length = strlen($input);
14     $random_string = '';
15     for($i = 0; $i < $strength; $i++) {
16         $random_character = $input[mt_rand(0, $input_length - 1)];
17         $random_string .= $random_character;
18     }
19     return $random_string;
20 }
21
22 $numero = generate_string($permitted_chars, 6);
23
24 $data = json_decode(file_get_contents("php://input"));
25
26 $personas = count($data)-1;
27 for ($i = 0; $i < $personas; $i++){
28     $nombre = $data[$i]->nombre;
29     $apellidos = $data[$i]->apellidos;
30     $dni = $data[$i]->dni;
31     $connect->query("INSERT INTO `pasajeros` (`cod_reserva`, `numero`,
32         `nombre`, `apellidos`) VALUES
33         ('".$numero.", ".$dni.", ".$nombre.", ".$apellidos.")" or
34         die(mysql_error());
35 }
36 $vuelo_ida = $data[$personas]->vuelo_ida;
37 $vuelo_vuelta = $data[$personas]->vuelo_vuelta;
38 $salida_ida = $data[$personas]->salida_ida;
39 $salida_vuelta = $data[$personas]->salida_vuelta;
40 $tipoBilleteIda = $data[$personas]->tipoBilleteIda;
```

En esta parte conectamos con la base de datos y añadimos los pasajeros.

```
57 if($vuelo_vuelta == ""){
58     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
59         `fecha_vuelo`, `vuelo`, `salida`, `npas_buissnes`, `npas_optima`,
60         `npas_economy`) VALUES ('".$numero.", NOW(), ".$salida_ida.",
61         ".$vuelo_ida.", ".$salida_ida.", ".$bussinessIda.",
62         ".$optimaIda.", ".$economyIda.")" or die(mysql_error());
63     $connect->exec("UPDATE `vuelos` SET `plazas_buissnes` =
64         `plazas_buissnes` - ".$bussinessIda.", `plazas_optima` =
65         `plazas_optima` - ".$optimaIda.", `plazas_economy` =
66         `plazas_economy` - ".$economyIda." WHERE `vuelo` =
67         '".$vuelo_ida.'" AND `salida` = '".$salida_ida.'" or
68         die(mysql_error());
69 } else {
70     $bussinessVuelta = 0;
71     $optimaVuelta = 0;
72     $economyVuelta = 0;
73     switch ($tipoBilleteVuelta) {
74         case 'Business':
75             $bussinessVuelta = $personas;
76             break;
77         case 'Optima':
78             $optimaVuelta = $personas;
79             break;
80         case 'Economy':
81             $economyVuelta = $personas;
82             break;
83     }
84     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
85         `fecha_vuelo`, `vuelo`, `salida`, `npas_buissnes`, `npas_optima`,
86         `npas_economy`) VALUES ('".$numero.", NOW(), ".$salida_vuelta.",
87         ".$vuelo_vuelta.", ".$salida_vuelta.", ".$bussinessVuelta.",
88         ".$optimaVuelta.", ".$economyVuelta.")" or die(mysql_error());
89     $connect->exec("UPDATE `vuelos` SET `plazas_buissnes` =
90         `plazas_buissnes` - ".$bussinessIda.", `plazas_optima` =
91         `plazas_optima` - ".$optimaIda.", `plazas_economy` =
92         `plazas_economy` - ".$economyIda." WHERE `vuelo` =
93         '".$vuelo_ida.'" AND `salida` = '".$salida_ida.'" or
94         die(mysql_error());
95     $connect->exec("UPDATE `vuelos` SET `plazas_buissnes` =
```

En esta parte nos encargamos de subir a la base de datos las cosas referentes a los vuelos.



CÓDIGO JAVASCRIPT

ALL.JS

Este documento Javascript lo hemos sacado de internet, junto con all.css, y son para el uso de unos iconos añadidos en nuestra página.

ANIME.JS

En este documento se ha definido el timeout para la animación del título de la página, consiguiendo el resultado esperado. El código es:

```
1 1 | setTimeout(function () {  
2   var textWrapper = document.querySelector('.ml6  
   .letters');  
3   textWrapper.innerHTML =  
   textWrapper.textContent.replace(/\S/g, "<span  
   class='letter'>${&&</span>");  
4  
   anime.timeline({loop: true})  
6   .add({  
7     targets: '.ml6 .letter',  
8     translateY: ["1.1em", 0],  
9     translateZ: 0,  
10    duration: 750,  
11    delay: (el, i) => 50 * i  
12  })  
13  .add({  
14    targets: '.ml6',  
15    opacity: 0,  
16    duration: 1000,  
17    easing: "easeOutExpo",  
18    delay: 1000  
19  });  
20 }, 1000);  
21 // Wrap every letter in a span  
22
```

Con este código definimos la duración de la animación del título.

También definimos otras propiedades, como la opacidad del texto o el retraso para que vuelva a iniciar la animación.

BOOTSTRAP.BUNDLE.JS

Este documento Javascript lo hemos sacado de internet y, junto con bootstrap.css, ayuda al diseño de nuestra página.



CONTROLLER2.JS

Este archivo es el controlador principal de nuestra página, se encarga de hacer todas las operaciones necesarias para el correcto funcionamiento de nuestra página de vuelos.

```
1  'use strict';
2
3  angular.module("trabajo", ['ngRoute'])
4  .controller("ctrol", function ($scope, $http, $filter) {
5      var contadorIDA = 0;
6      $scope.elementoIDA = {};
7      $scope.elementoSalida = {};
8      $scope.elementoSalidaV = {};
9      $scope.deshabilitarIDA = false;
10
11      var contadorVUELTA = 0;
12      $scope.elementoVUELTA = {};
13      $scope.nombre = null;
14      $scope.apellidos = null;
15
16      $scope.dni = null;
17      $scope.nameForm = null;
18
19      var pasajeros = [];
20
21      $scope.tipoBilleteIDA = '';
22      $scope.tipoBilleteVuelta = '';
23
24      $scope.deshabilitarVUELTA = false;
25      $scope.init = function () {
26          $scope.data = [];
27          $scope.error = "";
28          $scope.selectedOrderIDA = 'vuelo';
29          $scope.selectedOrderVUELTA = 'vuelo';
30          $scope.values = "origen";
31          $scope.precioIDA = 0;
32          $scope.precioVUELTA = 0;
33          $scope.vuelta = false;
34          $scope.ida = false;
35          $scope.showVuelta = true;
36          $scope.showCliente = true;
37      };
```

Al principio de este código nos vamos a dedicar a inicializar todas nuestras variables a unos valores iniciales.

```
62      $scope.submitForm = function () {
63          var billetes = document.getElementById('nBilletes').value;
64          for (var i = 0; i < billetes; i++){
65              pasajeros.push({ 'nombre':
66                  document.getElementById('nombre_' + i).value,
67                  'apellidos':
68                      document.getElementById('apellidos_' + i).value,
69                  'dni': document.getElementById('dni_' + i).value});
70          }
71          var elementoVUELTA = $scope.elementoVUELTA;
72          var elementoSalidaV = $scope.elementoSalidaV;
73          var tipoBilleteVuelta = $scope.tipoBilleteVuelta;
74          if(angular.equals($scope.elementoVUELTA, {})){
75              elementoVUELTA = "";
76              elementoSalidaV = "";
77              tipoBilleteVuelta = "";
78          }
79          pasajeros.push({ 'vuelo_ida': $scope.elementoIDA,
80              'vuelo_vuelta': elementoVUELTA,
81              'salida_ida': $scope.elementoSalida,
82              'salida_vuelta': elementoSalidaV,
83              'tipoBilleteIda': $scope.tipoBilleteIda,
84              'tipoBilleteVuelta': tipoBilleteVuelta});
85          console.log(pasajeros);
86          $http.post(
87              "php/subir.php", pasajeros
88          )
89          .then(function (respuesta) {
90              console.log(respuesta);
91              $scope.nombre = null;
92              $scope.apellidos = null;
93              $scope.dni = null;
94              window.location = ('index.php');
95          });
96      };
```

En esta parte se encarga de procesar el formulario de registro de pasajeros.



```
101 ▾ $scope.submitFormCompany = function () {  
102     $http.post(  
103         "insertar_vuelo.php", {  
104             'origen': $scope.searchOrigin,  
105             'destino': $scope.searchDestiny,  
106             'salida': $scope.fechaCheckIn,  
107             'llegada': $scope.fechaCheckOut,  
108             'plazas_business': $scope.plazas_business,  
109             'plazas_optima': $scope.plazas_optima,  
110             'plazas_economy': $scope.plazas_economy,  
111             'precio_business': $scope.precio_business,  
112             'precio_optima': $scope.precio_optima,  
113             'precio_economy': $scope.precio_economy,  
114         }  
115     )  
116     .then(function (respuesta) {  
117         console.log(respuesta);  
118         $scope.origen = null;  
119         $scope.destino = null;  
120         $scope.salida = null;  
121         $scope.llegada = null;  
122         $scope.plazas_business = null;  
123         $scope.plazas_optima = null;  
124         $scope.plazas_economy = null;  
125         $scope.precio_business = null;  
126         $scope.precio_optima = null;  
127         $scope.precio_economy = null;  
128         window.location = ('company.html');  
129     });  
130 };  
131  
132  
133  
134 /* Seleccionar orden de la tabla */  
135 ▾ $scope.setOrderIDA = function (x) {  
136     $scope.selectedOrderIDA = x;  
137 };  
138  
139 ▾ $scope.setOrderVUELTA = function (x) {  
140     $scope.selectedOrderVUELTA = x;  
141 };
```

En esta parte se encarga de procesar el formulario de registro de vuelos para las compañías.

```
144 /* Selecciona elemento de la checkbox (el identificador del vuelo)  
145 y lo almacena o lo borra*/  
146 /* Tambien activa o desactiva el poder mostrar el precio */  
147 ▾ $scope.seleccionIDA = function (posicion, salida) {  
148     contadorIDA++;  
149     $scope.ida = false;  
150     if (contadorIDA % 2 != 0) {  
151         $scope.elementoIDA = posicion;  
152         $scope.elementoSalida = salida;  
153         //console.log($scope.elementoSalida);  
154         $scope.estadoDeshabilitarIDA = true;  
155         $scope.ida = true;  
156     }  
157     else {  
158         contadorIDA = 0;  
159         $scope.elementoIDA = {};  
160         $scope.elementoSalida = {};  
161         $scope.estadoDeshabilitarIDA = false;  
162         $scope.ida = false;  
163         $scope.precioIDA = 0;  
164     }  
165 };  
166  
167 ▾ $scope.billetera = function (value) {  
168     return parseInt(value);  
169 }  
170  
171 ▾ $scope.seleccionVUELTA = function (posicion, salida) {  
172     contadorVUELTA++;  
173     if (contadorVUELTA % 2 != 0) {  
174         $scope.elementoVUELTA = posicion;  
175         $scope.elementoSalidaV = salida;  
176         $scope.estadoDeshabilitarVUELTA = true;  
177         $scope.vuelta = true;  
178     }  
179     else {  
180         contadorVUELTA = 0;  
181         $scope.elementoVUELTA = {};  
182         $scope.elementoSalidaV = {};  
183         $scope.estadoDeshabilitarVUELTA = false;  
184         $scope.vuelta = false;  
185         $scope.precioVUELTA = 0;  
186     }  
187 };
```

En esta parte, se selecciona el elemento de la checkbox (el identificador del vuelo) y lo almacena o lo borra.

También activa o desactiva el poder mostrar el precio.



```
188      /* Retorna el valor almacenado en el elemento (el identificador del
189      vuelo), si no coincide con el identificador de vuelo nuevo el
190      checkbox quedara deshabilitado */
191      $scope.deshabilitarIDA = function () {
192          return $scope.elementoIDA;
193      };
194      $scope.deshabilitarVUELTA = function () {
195          return $scope.elementoVUELTA;
196      };
197      /* Actualiza el precio del billete */
198
199      $scope.EstablecerPrecioIDA = function (x) {
200          $scope.precioIDA = x;
201      };
202      $scope.EstablecerPrecioVUELTA = function (x) {
203          $scope.precioVUELTA = x;
204      };
205
206      /* Al reservar un vuelo elimina el valor del elemento (el
207      identificador de vuelo) para que si se decide posteriormente
208      realizar otra reserva no aparezcan los checkbox deshabilitados */
209
210      $scope.limpiar = function () {
211          if ($scope.elementoIDA.length != undefined) {
212              $scope.seleccionIDA($scope.elementoIDA);
213          }
214          if ($scope.elementoVUELTA.length != undefined) {
215              $scope.seleccionVUELTA($scope.elementoVUELTA);
216          }
217      };
218
219      $scope.setTipoIda = function (tipo) {
220          $scope.tipoBilleteIda = tipo;
221      };
222
223      $scope.setTipoVuelta = function (tipo) {
224          $scope.tipoBilleteVuelta = tipo;
225      };
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
```

Esta parte retorna el valor almacenado en el elemento (el identificador del vuelo) y, si no coincide con el identificador de vuelo nuevo, el checkbox quedara deshabilitado.

Esta parte hace que al reservar un vuelo se elimine el valor del elemento (el identificador de vuelo) para que si se decide posteriormente realizar otra reserva no aparezcan los checkbox deshabilitados.

```
229      //indica desde que valor tiene que empezar a contar, se usa para la
230      paginacion de la pagina company.html
231      .filter('startFrom', function () {
232          return function (input, start) {
233              if (input) {
234                  start = +start; //parse to int
235                  var appended = input.slice(0, start);
236                  var initialArray = input.slice(start);
237                  var finalArray = initialArray.concat(appended);
238                  return finalArray;
239              }
240              return [];
241          };
242      });
243      /* Filtra los nombres de Origen y Destino de tal manera que no se
244      envíen duplicados */
245
246      .filter('unique', function () {
247          return function (collection, keyname) {
248              var output = [];
249              keys = [];
250
251              angular.forEach(collection, function (item) {
252                  var key = item[keyname];
253                  if (keys.indexOf(key) === -1) {
254                      keys.push(key);
255                      output.push(item);
256                  }
257              });
258              return output;
259          };
260      });
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Aquí se indica desde que valor tiene que empezar a contar, se usa para la paginacion de la página company.html

Esta parte filtra los nombres de Origen y Destino de tal manera que no se envíen duplicados.



```
259  /* Analiza letra a letra la cadena de texto y muestra caracteres
260      especiales */
261  .filter('utf8_decode', function ($sce) {
262      return function (strData) {
263          var tmpArr = [];
264          var i = 0;
265          var c1 = 0;
266          var seqLen = 0;
267          var anterior = 0;
268          strData += '';
269
270          while (i < strData.length) {
271              c1 = strData.charCodeAt(i) & 0xFF;
272              seqLen = 0;
273              if ((c1 == 63 && anterior == 73)) {
274                  c1 = 0xC1;
275                  seqLen = 1;
276
277              } else if ((c1 == 63 && anterior == 79)) {
278                  c1 = 0xD1;
279                  seqLen = 1;
280
281              } else if ((c1 == 63 && anterior == 85)) {
282                  c1 = 0xD1;
283                  seqLen = 1;
284
285              } else if (c1 <= 0xBF) {
286                  c1 = (c1 & 0x7F);
287                  seqLen = 1;
288                  anterior = c1;
289
290              } else if (c1 <= 0xDF) {
291                  c1 = (c1 & 0x1F);
292                  seqLen = 2;
293              } else if (c1 <= 0xFF) {
294                  c1 = 0xBF;
295                  seqLen = 3;
296              } else {
297                  c1 = (c1 & 0xFD);
298                  seqLen = 4;
299              }
300              tmpArr.push(strData[i]);
301              i++;
302          }
303          return tmpArr.join('');
304      };
305  });
```

Esta parte analiza letra a letra la cadena de texto y permite que se muestren caracteres especiales.

```
313  /* Mismo código pero codifica los caracteres, utilizado para las
314      consultas a la bbdd */
315  .filter('utf8_encode', function ($sce) {
316      return function (strData) {
317          var tmpArr = [];
318          var i = 0;
319          var c1 = 0;
320          var seqLen = 0;
321          var anterior = 0;
322          strData += '';
323          while (i < strData.length) {
324              c1 = strData.charCodeAt(i) & 0xFF;
325              seqLen = 0;
326              if ((c1 == 0xC1 && anterior == 73)) {
327                  c1 = 63;
328                  seqLen = 1;
329
330              } else if ((c1 == 0xD1 && anterior == 79)) {
331                  c1 = 63;
332                  seqLen = 1;
333
334              } else if ((c1 == 209 && anterior == 85)) {
335                  c1 = 63;
336                  seqLen = 1;
337
338              } else if (c1 <= 0xBF) {
339                  c1 = (c1 & 0x7F);
340                  seqLen = 1;
341                  anterior = c1;
342
343              } else if (c1 <= 0xDF) {
344                  c1 = (c1 & 0x1F);
345                  seqLen = 2;
346              } else if (c1 <= 0xFF) {
347                  c1 = 0xBF;
348                  seqLen = 3;
349              } else {
350                  c1 = (c1 & 0xFD);
351                  seqLen = 4;
352              }
353              tmpArr.push(strData[i]);
354              i++;
355          }
356          return tmpArr.join('');
357      };
358  });
```

Esta parte es igual a la anterior, pero aquí también estamos codificando los caracteres. Esta parte se utiliza para las consultas en la base de datos.



```
366 .config(function ($routeProvider) {
367     $routeProvider.
368     when('/', {
369         controller: 'ctrl',
370         templateUrl: 'main.php'
371     }).
372     when('/prueba/:origen/:destino/:nBilletes/:fcheckIn', {
373         controller: 'CountDownCtrl',
374         templateUrl: 'prueba.php'
375     }).
376     when('/prueba/:origen/:destino/:nBilletes/:fcheckIn/:fcheckOut', {
377         controller: 'CountDownCtrlb',
378         templateUrl: 'prueba.php'
379     }).
380     otherwise({
381         redirectTo: '/'
382     });
383 })
384 .controller('MainCtrl', function ($scope) {
385
386
387 })
388 .controller('CountDownCtrl', function ($scope, $routeParams, $timeout) {
389     $scope.origen = $routeParams.origen;
390     $scope.destino = $routeParams.destino;
391     $scope.fcheckIn = $routeParams.fcheckIn;
392     $scope.nBilletes = $routeParams.nBilletes;
393
394 })
395 .controller('CountDownCtrlb', function ($scope, $routeParams, $timeout)
396 {
397     $scope.origen = $routeParams.origen;
398     $scope.destino = $routeParams.destino;
399     $scope.fcheckIn = $routeParams.fcheckIn;
400     $scope.fcheckOut = $routeParams.fcheckOut;
401     $scope.nBilletes = $routeParams.nBilletes;
402
403 });
```

Esta parte relaciona la ruta recibida, con un controlador y un código php concreto.

En esta parte se definen los controladores para el código antes descrito, diferenciando para los casos de solo ida, o de ida y vuelta.



CÓDIGO CSS

ALL.CSS

Este documento CSS lo hemos sacado de internet, junto con all.js, y son para el uso de unos iconos añadidos en nuestra página.

BOOTSTRAP.CSS

Este documento CSS lo hemos sacado de internet y, junto con Bootstrap.bundle.js, ayuda al diseño de nuestra página.

CUSTOM.CSS

Este documento CSS se encarga de darle el formato al nombre de la página. Con este código podemos cambiar el tipo de letra, el grosor, el tamaño, la posición, etc. El código es:

```
1 |.ml6 {  
2   position: relative;  
3   font-weight: 900;  
4   font-size: 3.3em;  
5 }  
6  
7 |.ml6 .text-wrapper {  
8   position: relative;  
9   display: inline-block;  
10  padding-top: 0.2em;  
11  padding-right: 0.05em;  
12  padding-bottom: 0.1em;  
13  overflow: hidden;  
14 }  
15  
16 |.ml6 .letter {  
17   display: inline-block;  
18   line-height: 1em;  
19 }  
20  
21 |a, a:hover {  
22   text-decoration: none;  
23   color: white;  
24 }
```

En esta parte estamos definiendo el tamaño de la fuente.

Aquí, estamos alineando nuestro título para separarlo de los recuadros y que quede bien ubicado.