



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	9
DATABASE_CONNECTION.PHP	9
BYE.PHP	10
COMPANY.PHP	10
FETCH_DATA.PHP	11
HEADER.PHP	11
INICIO_SESION.PHP	12
REGISTRAR_USUARIO.PHP	12
SUBIR.PHP	13
ACTUALIZAR_VUELO.PHP	15
COMPRAR_DATOS.PHP	15
ELIMINAR_RESERVA.PHP	16
CÓDIGO JAVASCRIPT	18
ALL.JS	18
ANIME.JS	18
BOOTSTRAP.BUNDLE.JS	18
CONTROLLER2.JS	19
CÓDIGO CSS	24
ALL.CSS	24
BOOTSTRAP.CSS	24
CUSTOM.CSS	24
BOTON.CSS	25



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.

Introducir nuevos vuelos | Consultar vuelos | Actualizar vuelos

Airlines

Origen: Destino:

Salida: Llegada:

Plazas Business: Plazas Economy: Plazas Optima:

€ Precio Business: € Precio Economy: € Precio Optima:

Introducir nuevos vuelos | Consultar vuelos | Actualizar vuelos

Airlines

ID vuelo: Origen: Destino:

Vuelo de ida	Origen	Destino	Salida	Llegada	Bussiness	Optima	Economy	Clase	Ofertadas	Vendidas	Disponibles
IB-0081	TENERIFE NORTE	GRAN CANARIA	02/07 13:30	02/07 14:00	70	59	49	Business	20	0	20
IB-0081	TENERIFE NORTE	GRAN CANARIA	03/07 13:30	03/07 14:00	70	59	49	Optima	80	0	80
IB-0081	TENERIFE NORTE	GRAN CANARIA	04/07 13:30	04/07 14:00	70	59	49	Business	19	3	16
IB-0081	TENERIFE NORTE	GRAN CANARIA	05/07 13:30	05/07 14:00	70	59	49	Business	20	0	20
IB-0081	TENERIFE NORTE	GRAN CANARIA	06/07 13:30	06/07 14:00	70	59	49	Business	20	0	20
IB-0081	TENERIFE NORTE	GRAN CANARIA	07/07 13:30	07/07 14:00	70	59	49	Optima	80	0	80
IB-0081	TENERIFE NORTE	GRAN CANARIA	08/07 13:30	08/07 14:00	70	59	49	Optima	80	0	80
IB-0081	TENERIFE NORTE	GRAN CANARIA	09/07 13:30	09/07 14:00	70	59	49	Economy	60	0	60



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" href="css/bootstrap.css">
8   <link href="css/all.css" rel="stylesheet">
9   <link href="css/custom.css" rel="stylesheet">
10
11   <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">
12   <link rel="icon" href="images/favicon.ico" type="image/x-icon">
13
14   <title>Vuelos</title>
15 </head>
16
17 <body ng-controller="ctrol" style="background-image:
18   url('images/plane.jpg'); background-size: cover;">
19   <div ng-view></div>
20
21   <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
22     integrity="sha384-
23     KJ3o2DKtIkvYIK3UEZmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
24     crossorigin="anonymous"></script>
25   <script src="js/bootstrap.bundle.js"></script>
26   <script src="js/all.js"></script>
27   <script
28     src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js">
29   </script>
30   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular-
31     route.js"></script>
32   <script
33     src="https://cdnjs.cloudflare.com/ajax/libs/animejs/2.0.2/anime.min.js">
34   </script>
35   <script src="controller2.js"></script>
36   <script src="js/anime.js"></script>
37
38 </body>
39 </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  
                     </div>  
26                 </div>  
27                 <div class="row">  
28                     <div class="form-group col-md-  
                    6">  
29                         <label for =  
                        "passwd">Contraseña:  
                        </label>  
30                     </div>  
31                     <div class="form-group col-md-  
                    6">  
32                         <input type="password"  
                        name="passwd" id="passwd"  
                        autocomplete="off"  
                        required  
                        </div>  
33                     </div>  
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>  
                    <label class="form-check-  
                    label">Cliente</label>  
71                 </div>  
72                 <div class="form-group col-md-  
                6"></div>  
73                 <div class="form-group col-md-  
                6">  
74                     <input class="form-check-  
                    input" type="radio"  
                    name="rol" id="rol"  
                    value="aerolinea" ng-  
                    model="showCliente" ng-  
                    value="false">  
                    <label class="form-check-  
                    label">Aerolínea</label>  
75                 </div>  
76                 </div>  
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).

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

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.

```
145 </div>
146 <div class="row form-inline d-flex justify-content-center mt-3">
147 <label class="mr-1">¿Desea cambiar el número de pasajeros?
   </label>
148 <input class="form-control" style="width: 20%;" type="number"
   name="n_billetes" min="1" value="{{nBilletes}}" ng-
   model="nBilletes" id="nBilletes">
149 </div>
150 </div>
151 <div class="card-footer">
152 <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>
153
154 <!-- Solo IDA SIN escalas -->
155 <button ng-if="(precioIDA > 0) && (vuelta == false) &&
   CiudadIntermedia == ''" ng-hide="fcheckOut" class="btn btn-
   success float-right mr-3" type="button" ng-click="" data-
   toggle="modal" data-target="#modalRegistroPasajeros">
   <i class="far fa-calendar-check"></i> Reservar
156 </button>
157 <!-- Solo IDA CON escalas -->
158 <button ng-if="(precioIDA > 0) && (precioDestinoIDA > 0) &&
   (vuelta == false) && CiudadIntermedia != ''" ng-hide="fcheckOut"
   class="btn btn-success float-right mr-3" type="button" ng-
   click="" data-toggle="modal" data-
   target="#modalRegistroPasajeros">
   <i class="far fa-calendar-check"></i> Reservar
159 </button>
160 <!-- IDA Y VUELTA SIN escalas -->
161 <button ng-if="(precioIDA > 0) && (precioVUELTA > 0) && (vuelta
   == true) && CiudadIntermedia2 == '' && CiudadIntermedia == ''"
   class="btn btn-success float-right mr-3" type="button" ng-
   click="" data-toggle="modal" data-
   target="#modalRegistroPasajeros">
   <i class="far fa-calendar-check"></i> Reservar
162 </button>
163
164 </div>
165
166
167
```

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 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.



```
179 <div ng-switch-default>
180   <div data-ng-repeat="item in existenBilletes = (data | filter:
      {'origen' : origen} | filter: {'destino' : destino} | filter:
      {'salida' : fcheckIn}) ">
181   </div>
182 <div class="alert alert-warning mx-auto text-center mt-3"
      style="width: 70%" role="alert" ng-if="existenBilletes.length ==
      0">
183
184     Ups, parece que no hay vuelos con origen {{origen}} y destino
      {{destino }} para las fechas seleccionadas pero
      afortunadamente para ti tenemos una ruta alternativa
185
186
187   <table>
188     <tr>
189       <td>
190         <input class="form-control" style="" type="text"
          placeholder="{{item.origen}}" ng-model="origen"
          value="{{item.origen}}" disabled>
191
192       </td>
193     </tr>
194     <tr>
195       <td class="text-left">
196         <div ng-repeat="item in data | filter: {'origen' :
          (origen | utf8_encode)} | filter: {'salida' :
          fcheckIn} | orderBy:'destino'| unique: 'destino'"
          value="{{item.destino | utf8_decode}}">
197           <div ng-repeat="items in data | filter:
            {'origen' : (item.destino | utf8_encode)} |
            filter: {'destino' : (destino | utf8_encode)}
            | filter: {'salida' : fcheckIn} |
            orderBy:'origen'| unique: 'origen' value="
            {{items.origen | utf8_decode}}">
198             <div ng-show="{{(item.destino |
              removeSpaces)}} == {{items.origen |
              removeSpaces}}">
199               <input class="form-check-input"
                type="checkbox" ng-
                click="seleccionIntermedioIDA(items.o
```

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

Tras esto, la página nos recomienda una ruta alternativa.

```
236 <div class="card mt-5 mb-5 mx-auto" style="width: max-
      content; background-color: rgba(255, 255, 255, 0.75);" ng-
      show="CiudadIntermedia != '' ">
237   <div class="card-body">
238     <table class="table table-striped bg-white" ng-
      model="tablaIDA">
239       <tr>
240         <th ng-click="setOrderIDA('vuelo')"> Vuelo de
          ida</th>
241         <th ng-click="setOrderIDA('origen')">
          Origen</th>
242         <th ng-click="setOrderIDA('destino')">
          Destino </th>
243         <th ng-click="setOrderIDA('salida')">
          Salida</th>
244         <th ng-click="setOrderIDA('llegada')">
          Llegada</th>
245         <th ng-
          click="setOrderIDA('precio_business')">
          Bussiness</th>
246         <th ng-click="setOrderIDA('precio_optima')">
          Optima </th>
247         <th ng-click="setOrderIDA('precio_economy')">
          Economy </th>
248       </tr>
249       <tr ng-repeat="item in data | filter: {'origen' :
        (origen | utf8_encode)} | filter: {'destino' :
        (CiudadIntermedia | utf8_encode)} | filter:
        {'salida' : fcheckIn} |
        orderBy:selectedOrderIDA:reverse">
250         <td> {{ item.vuelo}}</td>
251         <td> {{ item.origen | utf8_decode }}</td>
252         <td> {{ item.destino | utf8_decode }} </td>
253         <td class="text-center"> {{ item.salida |
          date: "HH:mm" }} </td>
254         <td class="text-center"> {{ item.llegada |
          date: "HH:mm" }} </td>
255         <td class="text-center"> {{
          item.precio_business }}</td>
256         <td class="text-center"> {{
          item.precio_optima }} </td>
257         <td class="text-center"> {{
```

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.



```
457 <!-- MODAL REGISTRAR PASAJEROS -->
458 <div class="modal fade" id="modalRegistroPasajeros" tabindex="-1"
459   role="dialog" aria-labelledby="exampleModallabel" aria-hidden="true">
460   <div class="modal-dialog" role="document">
461     <div class="modal-content">
462       <div class="modal-header">
463         <h5 class="modal-title" style="color: dodgerblue"><i
464           class="fas fa-user-edit"></i> Pasajeros</h5>
465         <button type="button" class="close" data-dismiss="modal"
466           aria-label="Close">
467           <span aria-hidden="true" ng-click="">&times;</span>
468         </button>
469       </div>
470       <form name="userForm" ng-submit="submitForm()">
471         <div class="modal-body">
472           <table class="table table-striped">
473             <thead>
474               <tr>
475                 <th scope="col"></th>
476                 <th scope="col">Nombre</th>
477                 <th scope="col">Apellidos</th>
478                 <th scope="col">DNI (sin letra)</th>
479               </tr>
480             </thead>
481             <tbody>
482               <tr ng-repeat="x in [].constructor(nBilletes)
483                 track by $index">
484                 <td class="align-middle" scope="col">
485                   {{ $index+1 }}</td>
486                 <td class="align-middle" scope="col">
487                   <input type="text"
488                     class="form-control" required
489                     id="nombre_{{ $index }}"></td>
490                 <td class="align-middle" scope="col">
491                   <input type="text"
492                     class="form-control" required
493                     id="apellidos_{{ $index }}"></td>
494                 <td class="align-middle" scope="col">
495                   <input type="number"
496                     class="form-control" required
497                     id="dni_{{ $index }}" style="-moz-
```

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

```
619 <!-- MODAL CANCELAR RESERVA -->
620 <div class="modal fade" id="cancelarReserva" tabindex="-1" role="dialog"
621   aria-labelledby="exampleModallabel" aria-hidden="true">
622   <div class="modal-dialog" role="document">
623     <div class="modal-content">
624       <div class="modal-header">
625         <h5 class="modal-title" style="color: dodgerblue"><i
626           class="fas fa-receipt mr-1"></i> Datos de su reserva</h5>
627         <button type="button" class="close" data-dismiss="modal"
628           aria-label="Close">
629           <span aria-hidden="true" ng-click="">&times;</span>
630         </button>
631       </div>
632       <form name="userForm" ng-submit="submitFormReserva()">
633         <div class="modal-body">
634           <div class="row">
635             <div class="col-md-2"></div>
636             <div class="col-md-4 form-inline">
637               <input class="form-check-input" type="radio"
638                 ng-model="showInputDataUser" ng-value="true"
639                 checked>
640               <label class="form-check-label">Mi
641                 billete</label>
642             </div>
643             <div class="col-md-5 form-inline">
644               <input class="form-check-input" type="radio"
645                 ng-change="resetDatosCancReserva()" ng-
646                 model="showInputDataUser" ng-value="false"
647                 checked>
648               <label class="form-check-label">Todos los
649                 billetes</label>
650             </div>
651             <div class="col-md-1"></div>
652             <div class="col-md-12 mt-2 form-inline">
653               <label class="form-check-label"><i class="far
654                 fa-file-alt mr-1"></i> Código de reserva:
655               </label>
656               <input class="form-control ml-1"
657                 type="number" ng-model="setReserva"
```

En esta parte es la encargada de las cancelaciones de las reservas.



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. En caso de que no se introduzcan las credenciales adecuados, te redirige a la página index.php.

```
1 <?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 $usuario = $_COOKIE ['aero_ck'];
12 include 'database_connection.php';
13 $sql = "SELECT id FROM aerolineas WHERE correo =:login";
14 $q = $connect->prepare($sql);
15 $q->execute(array(":login"=>$usuario));
16 $r = $q->fetch(PDO::FETCH_ASSOC);
17 $id = $r['id'];
18 $rand_1 = rand(0,9);
19 $rand_2 = rand(0,9);
20 $rand_3 = rand(0,9);
21 $rand_4 = rand(0,9);
22 echo $id;
23 echo "-";
24 echo $rand_1;
25 echo $rand_2;
26 echo $rand_3;
27 echo $rand_4;
28 $random_id = $id."-".$rand_1.$rand_2.$rand_3.$rand_4;
29
30
31 $data = json_decode(file_get_contents("php://input"));
32
33 $origen = $data->origen;
34 $destino = $data->destino;
35 $salida = $data->salida;
36 $llegada = $data->llegada;
37 $plazas_business = $data->plazas_business;
38 $plazas_optima = $data->plazas_optima;
39 $plazas_economy = $data->plazas_economy;
40 $precio_business = $data->precio_business;
41 $precio_optima = $data->precio_optima;
42 $precio_economy = $data->precio_economy;
43
44
45
46 $connect->query("INSERT INTO `vuelos` (`vuelo`,`origen`,`destino`,`
47     `salida`,`llegada`,`plazas_business`,`plazas_optima`,`plazas_economy`,`preci
48     o_business`,`precio_optima`,`precio_economy`) VALUES
49     ('".$random_id."','".$origen."','".$destino."','".$salida."','".$llegada."",
50     '".$plazas_business."','".$plazas_optima."','".$plazas_economy."','".$preci
51     o_business."','".$precio_optima."','".$precio_economy."') or
52     die(mysql_error());
53 $connect = null;
54 ?>
```

DATABASE_CONNECTION.PHP

Este código sirve para realizar la conexión con el servidor.

```
1 <?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
10 ?>
```



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
  CONTRASEÑA
4
5     setcookie("usr_ck", 1, time()-1, ";path=/");
6
7     setcookie("aero_ck", 1, time()-1, ";path=/");
8     header("location: ../index.php");
9
10 //VUELTA A LA PÁGINA DE LOGIN
11 ?>
```

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     </span>
56 </div>
57 <div class="card mx-auto" style="width: max-content;
58     background-color: rgba(255, 255, 255, 0.75);">
59     <div class="card-body">
60         <form novalidate name="f" ng-
61             submit="submitFormCompany()">
62             <div class="row">
63                 <div class="form-group col-md-6">
64                     <label><i class="fas fa-plane-
65                         departure"></i> Origen: </label>
66                     <select class="form-control" ng-
67                         model="searchOrigin">
68                         <option ng-repeat="item in data
69                             | orderBy:'origen' | unique:
70                                 'origen' value="
71                                     {{item.origen}} |
72                                     utf8_decode}}</option>
73                         <option value="{{}}"> selected
74                             disabled hidden>Desde dónde
75                             vamos?</option>
76                     </select>
77                 </div>
78                 <div class="form-group col-md-6">
79                     <label><i class="fas fa-plane-
80                         arrival"></i> Destino: </label>
81                     <select class="form-control" ng-
82                         model="searchDestiny">
83                         <option ng-repeat="item in data
84                             | orderBy:'destino' | unique:
85                                 'destino' value="
86                                     {{item.destino}} |
87                                     ng-
88                                     model="searchDestiny">
89                                     {{item.destino | utf8_decode}}
90                                     </option>
91                         <option value="{{}}"> selected
92                             disabled hidden>¿A dónde vamos?
93                         </option>
94                     </select>
95                 </div>
96             </div>
97         </div>
98     </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.

```
140 <div class="card-body">
141     <div class="row">
142         <div class="form-group col-md-4">
143             <label><i class="fas fa-plane-departure"></i> ID
144             vuelo: </label>
145             <select class="form-control" ng-model="IDVuelo">
146                 <option ng-repeat="item in data | unique: 'vuelo'
147                     | orderBy:'vuelo' value="{{item.vuelo}}">
148                     {{item.vuelo}}</option>
149                 <option value="{{}}"> selected disabled
150                     hidden>Indique la ID del vuelo</option>
151             </select>
152         </div>
153         <div class="form-group col-md-4">
154             <label><i class="fas fa-plane-departure"></i> Origen:
155             </label>
156             <select class="form-control" ng-
157                 model="RutaVueloOrigen" ng-disabled="IDVuelo ==
158                 undefined">
159                 <option ng-repeat="item in data|filter: {'vuelo'
160                     : IDVuelo } | filter: {'origen' : RutaVueloOrigen
161                     } | unique: 'origen' | orderBy:'origen' "
162                     value="{{item.origen}} | {{item.origen |
163                     utf8_decode}}</option>
164                 <option value="{{}}"> selected disabled
165                     hidden>Indique el origen</option>
166             </select>
167         </div>
168         <div class="form-group col-md-4">
169             <label><i class="fas fa-plane-departure"></i> Destino: </label>
170             <select class="form-control" ng-
171                 model="RutaVueloDestino" ng-disabled="RutaVueloOrigen
172                 == undefined">
173                 <option ng-repeat="item in data|filter: {'vuelo'
174                     : IDVuelo } | filter: {'origen' : RutaVueloOrigen
175                     } | unique: 'destino' | orderBy:'destino' "
176                     value="{{item.destino}} | {{item.destino |
177                     utf8_decode}}</option>
178                 <option value="{{}}"> selected disabled
179                     hidden>Indique el destino</option>
180             </select>
181         </div>
182     </div>
183 </div>
```

En esta parte empezamos con la creación de las tablas de 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 <?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         <!-->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         <!-->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 elseif(!isset($_COOKIE['usr_ck']) && !isset($_COOKIE['aero_ck'])){
25     echo ' <div>
26         <li class="nav-item mr-3">
27             <button type="button" class="btn btn-outline-
28 primary" data-toggle="modal" data-target="#myModal2"
29 >
30             <!--><a href="company.html"></a>-->
31             <b>Iniciar sesión</b>
32         </li>
33     </div>
34     <div>
35         <li class="nav-item mr-3">
36             <button type="button" class="btn btn-outline-
37 primary" data-toggle="modal" data-target="#myModal"
38 >
39             <b>Regístrate</b>
40         </li>
41     </div>
42     <div>
43         <li class="nav-item">
44             <button type="button" class="btn btn-outline-danger"
45 data-toggle="modal" data-target="#cancelarReserva" >
46             <b>Cancelar reserva</b>
47         </li>
48     </div>
49     </div>';
50 echo ' </ul>
51 </nav>';
52 }
```

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 }else{
12     $tipo_usuario="aerolinea";
13 }
14 $pass_cifrado=password_hash($pass, PASSWORD_DEFAULT);
15
16 try{
17     include 'database_connection.php';
18     //Primero comprobamos que no haya ningún otro usuario con ese
19     //correo, en caso de que lo haya no deja registrarse
20     $sql = "SELECT correo FROM usuarios WHERE correo =:user";
21     $resultado = $connect->prepare($sql);
22     $resultado->execute(array(":user"=>$correo_usuario));
23     if ($resultado->rowCount() > 0){
24
25     } else {
26         $sql_2="INSERT INTO usuarios (correo, passwd, nombre, apellidos,
27         rol)
28         VALUES (:usu, :pass, :nombre, :apellidos, :rol_usuario)";
29         $resultado=$connect->prepare($sql_2);
30         $resultado->execute(array(":usu"=>$correo_usuario,
31         ":pass"=>$pass_cifrado, ":nombre"=>$nombre,
32         ":apellidos"=>$apellidos, ":rol_usuario"=>$tipo_usuario));
33         echo $tipo_usuario;
34     }
35 } catch(Exception $e){
36     die ("Error" . $e -> getMessage() . $e -> getLine());
37 } finally{
38     $connect = null;
39     header("location: ../index.html");
40 }
```

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.

```
1  <?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  $data = json_decode(file_get_contents("php://input"));
12
13  $personas = count($data)-1;
14  $numero = $data[$personas]->numero;
15
16  for ($i = 0; $i < $personas; $i++){
17      $nombre = $data[$i]->nombre;
18      $apellidos = $data[$i]->apellidos;
19      $dni = $data[$i]->dni;
20      $connect->query("INSERT INTO `pasajeros` (`cod_reserva`, `numero`,
21                      `nombre`, `apellidos`) VALUES
22                      ('".$numero."', '".$dni."', '".$nombre."', '".$apellidos."') or
23                      die(mysql_error());
24  }
25
26  $vuelo_ida = $data[$personas]->vuelo_ida;
27  $vuelo_vuelta = $data[$personas]->vuelo_vuelta;
28  $salida_ida = $data[$personas]->salida_ida;
29  $salida_vuelta = $data[$personas]->salida_vuelta;
30  $tipoBilleteIda = $data[$personas]->tipoBilleteIda;
31  $tipoBilleteVuelta = $data[$personas]->tipoBilleteVuelta;
32
33  ///Intermedio
34  $vuelo_Intermedio_ida = $data[$personas]->vuelo_Intermedio_ida;
35  $vuelo_Intermedio_vuelta = $data[$personas]->vuelo_Intermedio_vuelta;
36  $salida_Intermedio_ida = $data[$personas]->salida_Intermedio_ida;
37  $salida_Intermedio_vuelta = $data[$personas]->salida_Intermedio_vuelta;
38  $tipoBilleteIntermedioIda = $data[$personas]->tipoBilleteIntermedioIda;
39  $tipoBilleteIntermedioVuelta = $data[$personas]->tipoBilleteIntermedioVuelta;
```

En esta parte conectamos con la base de datos creamos los vectores con los datos que vamos a subir a la base de datos.

```
40  $businessIda = 0;
41  $optimaIda = 0;
42  $economyIda = 0;
43  switch ($tipoBilleteIda) {
44      case 'Business':
45          $businessIda = $personas;
46          break;
47      case 'Optima':
48          $optimaIda = $personas;
49          break;
50      case 'Economy':
51          $economyIda = $personas;
52          break;
53  }
54  $businessIntermedioIda = 0;
55  $optimaIntermedioIda = 0;
56  $economyIntermedioIda = 0;
57  switch ($tipoBilleteIntermedioIda) {
58      case 'Business':
59          $businessIntermedioIda = $personas;
60          break;
61      case 'Optima':
62          $optimaIntermedioIda = $personas;
63          break;
64      case 'Economy':
65          $economyIntermedioIda = $personas;
66          break;
67  }
68
69  if($vuelo_vuelta == "" && $vuelo_Intermedio_ida == ""){
70      $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
71          `fecha_vuelo`, `vuelo`, `salida`, `npas_business`, `npas_optima`,
72          `npas_economy`) VALUES ('".$numero."', NOW(), '".$salida_ida."',
73          '".$vuelo_ida."', '".$salida_ida."', '".$businessIda."',
74          '".$optimaIda."', '".$economyIda."');") or die(mysql_error());
75      $connect->exec("UPDATE `vuelos` SET `plazas_business` =
76          `plazas_business` - ".$businessIda.", `plazas_optima` = `plazas_optima` -
77          ".$optimaIda.", `plazas_economy` = `plazas_economy` - ".$economyIda."
78          WHERE `vuelo` = '".$vuelo_ida.'" AND `salida` = '".$salida_ida.'"");
79      die(mysql_error());
80  }
```

En esta parte seguimos haciendo lo mismo que antes y empezamos con la subida de datos.



```
71 * else if($vuelo_vuelta == "" && $vuelo_Intermedio_ida != ""){
72     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
    `fecha_vuelo`, `vuelo`, `salida`, `npas_businnes`, `npas_optima`,
    `npas_economy`) VALUES ('".$numero."', NOW(),
    '".$salida_Intermedio_ida."', '".$vuelo_Intermedio_ida."',
    '".$salida_Intermedio_ida."', '".$businessIntermedioIda."',
    '".$optimaIntermedioIda."', '".$economyIntermedioIda."');") or
    die(mysql_error());
73
74
75     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
    `fecha_vuelo`, `vuelo`, `salida`, `npas_businnes`, `npas_optima`,
    `npas_economy`) VALUES ('".$numero."', NOW(), '".$salida_ida."',
    '".$vuelo_ida."', '".$salida_ida."', '".$businessIda."',
    '".$optimaIda."', '".$economyIda."');") or die(mysql_error());
76
77     $connect->exec("UPDATE `vuelos` SET `plazas_business` =
    `plazas_business` - ".$businessIda.", `plazas_optima` = `plazas_optima`
    - ".$optimaIda.", `plazas_economy` = `plazas_economy` - ".$economyIda."
    WHERE `vuelo` = '".$vuelo_ida."' AND `salida` = '".$salida_ida."' or
    die(mysql_error());
78
79     $connect->exec("UPDATE `vuelos` SET `plazas_business` =
    `plazas_business` - ".$businessIntermedioIda.", `plazas_optima` =
    `plazas_optima` - ".$optimaIntermedioIda.", `plazas_economy` =
    `plazas_economy` - ".$economyIntermedioIda." WHERE `vuelo` =
    '".$vuelo_Intermedio_ida."' AND `salida` =
    '".$salida_Intermedio_ida."' or die(mysql_error());
```

```
80 * } else if ($vuelo_vuelta != "" && $vuelo_Intermedio_vuelta == ""){
81     $businessVuelta = 0;
82     $optimaVuelta = 0;
83     $economyVuelta = 0;
84     switch ($tipoBilleteVuelta) {
85         case 'Business':
86             $businessVuelta = $personas;
87             break;
88         case 'Optima':
89             $optimaVuelta = $personas;
90             break;
91         case 'Economy':
92             $economyVuelta = $personas;
93             break;
94     }
95     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
    `fecha_vuelo`, `vuelo`, `salida`, `npas_businnes`, `npas_optima`,
    `npas_economy`) VALUES ('".$numero."', NOW(), '".$salida_ida."',
    '".$vuelo_ida."', '".$salida_ida."', '".$businessIda."',
    '".$optimaIda."', '".$economyIda."');") or die(mysql_error());
96     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
    `fecha_vuelo`, `vuelo`, `salida`, `npas_businnes`, `npas_optima`,
    `npas_economy`) VALUES ('".$numero."', NOW(), '".$salida_vuelta."',
    '".$vuelo_vuelta."', '".$salida_vuelta."', '".$businessVuelta."',
    '".$optimaVuelta."', '".$economyVuelta."');") or die(mysql_error());
97     $connect->exec("UPDATE `vuelos` SET `plazas_business` =
    `plazas_business` - ".$businessIda.", `plazas_optima` = `plazas_optima`
    - ".$optimaIda.", `plazas_economy` = `plazas_economy` - ".$economyIda."
    WHERE `vuelo` = '".$vuelo_ida."' AND `salida` = '".$salida_ida."' or
    die(mysql_error());
98     $connect->exec("UPDATE `vuelos` SET `plazas_business` =
    `plazas_business` - ".$businessVuelta.", `plazas_optima` =
    `plazas_optima` - ".$optimaVuelta.", `plazas_economy` = `plazas_economy` -
    ".$economyVuelta." WHERE `vuelo` = '".$vuelo_vuelta."' AND `salida` =
    '".$salida_vuelta."' or die(mysql_error());
99 }
```

```
101 * else {
102     $businessVuelta = 0;
103     $optimaVuelta = 0;
104     $economyVuelta = 0;
105     switch ($tipoBilleteVuelta) {
106         case 'Business':
107             $businessVuelta = $personas;
108             break;
109         case 'Optima':
110             $optimaVuelta = $personas;
111             break;
112         case 'Economy':
113             $economyVuelta = $personas;
114             break;
115     }
116     $businessIntermedioVuelta = 0;
117     $optimaIntermedioVuelta = 0;
118     $economyIntermedioVuelta = 0;
119     switch ($tipoBilleteIntermedioVuelta) {
120         case 'Business':
121             $businessIntermedioVuelta = $personas;
122             break;
123         case 'Optima':
124             $optimaIntermedioVuelta = $personas;
125             break;
126         case 'Economy':
127             $economyIntermedioVuelta = $personas;
128             break;
129     }
130
131     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
    `fecha_vuelo`, `vuelo`, `salida`, `npas_businnes`, `npas_optima`,
    `npas_economy`) VALUES ('".$numero."', NOW(),
    '".$salida_Intermedio_ida."', '".$vuelo_Intermedio_ida."',
    '".$salida_Intermedio_ida."', '".$businessIntermedioIda."',
    '".$optimaIntermedioIda."', '".$economyIntermedioIda."');") or
    die(mysql_error());
132
133
134     $connect->query("INSERT INTO `compras` (`cod_reserva`, `fecha_compra`,
    `fecha_vuelo`, `vuelo`, `salida`, `npas_businnes`, `npas_optima`,
```

En esta parte nos dedicamos a subir los datos para los diferentes casos (con escala o sin escala) para los vuelos se ida y los de ida y vuelta.



ACTUALIZAR_VUELO.PHP

Con este código lo que queremos conseguir es que las aerolíneas puedan modificar un vuelo ya existente.

```
1 <?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 $vuelo = $data->vuelo;
16 $origen = $data->origen;
17
18 $destino = $data->destino;
19 $fecha = $data->fecha;
20
21 $plazas_business = $data->plazas_business;
22 $plazas_optima = $data->plazas_optima;
23 $plazas_economy = $data->plazas_economy;
24
25 $precio_business = $data->precio_business;
26 $precio_optima = $data->precio_optima;
27 $precio_economy = $data->precio_economy;
28
29
30
31 $connect->query("UPDATE vuelos SET plazas_business='".$plazas_business'
32 ,plazas_optima='".$plazas_optima',plazas_economy='".$plazas_economy' ,
33 precio_business='".$precio_business' , precio_optima='".$precio_optima' ,
34 precio_economy='".$precio_economy' where vuelo='".$vuelo' AND origen='".$origen' AND
35 destino='".$destino' and salida='".$fecha' ")or die(mysql_error());
36
37 $connect = null;
38 ?>
```

COMPRAR_DATOS.PHP

Este código sirve para sacar los datos de la tabla “compras”, de donde podemos conseguir los billetes que se han vendido, sus fechas, etc.

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




ELIMINAR_RESERVA.PHP

Este código lo usamos para procesar el formulario de cancelación de reserva, permitiendo al cliente eliminar la reserva entera o eliminar solo a una persona de la lista de pasajeros.

```
1 <?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 $data = json_decode(file_get_contents("php://input"));
12
13 $reserva = $data->reserva;
14 $nombre = $data->nombre;
15 $apellidos = $data->apellidos;
16 $dni = $data->dni;
17
18 if (empty($nombre)) {
19     $connect->query("DELETE FROM `pasajeros` WHERE `cod_reserva` =
20         '". $reserva. "' or die(mysql_error());");
21
22     $query = "SELECT vuelo,salida,npas_businnes,npas_optima,npas_economy FROM
23         `compras` WHERE cod_reserva = '". $reserva. "'";
24
25     $res_data = $connect->query($query);
26     $vuelo = [];
27     $salida = [];
28     $npas_businnes = [];
29     $npas_optima = [];
30     $npas_economy = [];
31     $count = 0;
32     while ($line = $res_data->fetch(PDO::FETCH_ASSOC)) {
33         $vuelo[$count] = $line['vuelo'];
34         $salida[$count] = $line['salida'];
35         $npas_businnes[$count] = $line['npas_businnes'];
36         $npas_optima[$count] = $line['npas_optima'];
37         $npas_economy[$count] = $line['npas_economy'];
38         $count++;
39     }
40
41     if ($npas_businnes[0] != 0) {
42         $connect->exec("UPDATE `vuelos` SET `plazas_business` =
43             'plazas_business' + ".$npas_businnes[0]. " WHERE `vuelo` =
44             '". $vuelo[0]. "' AND `salida` = '". $salida[0]. "' AND `salida` =
45             '". $salida[0]. "' or die(mysql_error());");
46     }
47     elseif ($npas_optima[0] != 0) {
48         $connect->exec("UPDATE `vuelos` SET `plazas_optima` = `plazas_optima`
49             + ".$npas_optima[0]. " WHERE `vuelo` = '". $vuelo[0]. "' AND `salida` =
50             '". $salida[0]. "' or die(mysql_error());");
51     }
52     elseif ($npas_economy[0] != 0) {
53         $connect->exec("UPDATE `vuelos` SET `plazas_economy` =
54             'plazas_economy' + ".$npas_economy[0]. " WHERE `vuelo` =
55             '". $vuelo[0]. "' AND `salida` = '". $salida[0]. "' or
56             die(mysql_error());");
57     }
58
59     if ($count > 1) {
60         if ($npas_businnes[1] != 0) {
61             $connect->exec("UPDATE `vuelos` SET `plazas_business` =
62                 'plazas_business' + ".$npas_businnes[1]. " WHERE `vuelo` =
63                 '". $vuelo[1]. "' AND `salida` = '". $salida[1]. "' or
64                 die(mysql_error());");
65         }
66         elseif ($npas_optima[1] != 0) {
67             $connect->exec("UPDATE `vuelos` SET `plazas_optima` =
68                 'plazas_optima' + ".$npas_optima[1]. " WHERE `vuelo` =
69                 '". $vuelo[1]. "' AND `salida` = '". $salida[1]. "' or
70                 die(mysql_error());");
71         }
72         elseif ($npas_economy[1] != 0) {
73             $connect->exec("UPDATE `vuelos` SET `plazas_economy` =
74                 'plazas_economy' + ".$npas_economy[1]. " WHERE `vuelo` =
75                 '". $vuelo[1]. "' AND `salida` = '". $salida[1]. "' or
76                 die(mysql_error());");
77         }
78     }
79
80     $connect->query("DELETE FROM `compras` WHERE `cod_reserva` =
81         '". $reserva. "' or die(mysql_error());");
82 }
```

En esta parte se borraría la reserva entera de la lista de pasajeros, liberando las plazas para que puedan ser ocupadas de nuevo.



```
56 ▾ } else {
57     $connect->query("DELETE FROM 'pasajeros' WHERE 'cod_reserva' =
        '" . $reserva . "' AND 'nombre' = '" . $nombre . "' AND 'apellidos' =
        '" . $apellidos . "' AND 'numero' = '" . $dni . "'" ) or die(mysql_error());
58     $query = "SELECT vuelo,salida,npas_businnes,npas_optima,npas_economy FROM
        'compras' WHERE cod_reserva = '" . $reserva . "'";
59     $res_data = $connect->query($query);
60     $vuelo = [];
61     $salida = [];
62     $npas_businnes = [];
63     $npas_optima = [];
64     $npas_economy = [];
65     $count = 0;
66 ▾ while ($line = $res_data->fetch(PDO::FETCH_ASSOC)) {
67         $vuelo[$count] = $line['vuelo'];
68         $salida[$count] = $line['salida'];
69         $npas_businnes[$count] = $line['npas_businnes'];
70         $npas_optima[$count] = $line['npas_optima'];
71         $npas_economy[$count] = $line['npas_economy'];
72         $count++;
73     }
74 ▾ if ($npas_businnes[0] > 1){
75     $connect->exec("UPDATE 'compras' SET 'npas_businnes' =
        'npas_businnes' - 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' =
        '" . $salida[0] . "' AND 'cod_reserva' = '" . $reserva . "'" ) or
        die(mysql_error());
76     $connect->exec("UPDATE 'vuelos' SET 'plazas_business' =
        'plazas_business' + 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' =
        '" . $salida[0] . "'" ) or die(mysql_error());
77 ▾ } elseif ($npas_optima[0] > 1){
78     $connect->exec("UPDATE 'compras' SET 'npas_optima' = 'npas_optima' -
        1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' = '" . $salida[0] . "' AND
        'cod_reserva' = '" . $reserva . "'" ) or die(mysql_error());
79     $connect->exec("UPDATE 'vuelos' SET 'plazas_optima' = 'plazas_optima'
        + 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' = '" . $salida[0] . "'"
        or die(mysql_error());
80 ▾ } elseif ($npas_economy[0] > 1){
81     $connect->exec("UPDATE 'compras' SET 'npas_economy' = 'npas_economy'
        - 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' = '" . $salida[0] . "'
        AND 'cod_reserva' = '" . $reserva . "'" ) or die(mysql_error());
82     $connect->exec("UPDATE 'vuelos' SET 'plazas_economy' =
        'plazas_economy' + 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' = '" . $salida[0] . "'");
83 ▾ }
```

```
83 ▾ } elseif ($npas_businnes[0] == 1){
84     $connect->query("DELETE FROM 'compras' WHERE 'cod_reserva' =
        '" . $reserva . "'" ) or die(mysql_error());
85     $connect->exec("UPDATE 'vuelos' SET 'plazas_business' =
        'plazas_business' + 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' =
        '" . $salida[0] . "'" ) or die(mysql_error());
86 ▾ } elseif ($npas_optima[0] == 1){
87     $connect->query("DELETE FROM 'compras' WHERE 'cod_reserva' =
        '" . $reserva . "'" ) or die(mysql_error());
88     $connect->exec("UPDATE 'vuelos' SET 'plazas_optima' = 'plazas_optima'
        + 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' = '" . $salida[0] . "'"
        or die(mysql_error());
89 ▾ } elseif ($npas_economy[0] == 1){
90     $connect->query("DELETE FROM 'compras' WHERE 'cod_reserva' =
        '" . $reserva . "'" ) or die(mysql_error());
91     $connect->exec("UPDATE 'vuelos' SET 'plazas_economy' =
        'plazas_economy' + 1 WHERE 'vuelo' = '" . $vuelo[0] . "' AND 'salida' =
        '" . $salida[0] . "'" ) or die(mysql_error());
92     }
93 ▾ if ($count > 1){
94 ▾ if ($npas_businnes[1] > 1){
95     $connect->exec("UPDATE 'compras' SET 'npas_businnes' =
        'npas_businnes' - 1 WHERE 'vuelo' = '" . $vuelo[1] . "' AND 'salida' =
        '" . $salida[1] . "'" ) or die(mysql_error());
96     $connect->exec("UPDATE 'vuelos' SET 'plazas_business' =
        'plazas_business' + 1 WHERE 'vuelo' = '" . $vuelo[1] . "' AND
        'salida' = '" . $salida[1] . "'" ) or die(mysql_error());
97 ▾ } elseif ($npas_optima[1] > 1){
98     $connect->exec("UPDATE 'compras' SET 'npas_optima' =
        'npas_optima' - 1 WHERE 'vuelo' = '" . $vuelo[1] . "' AND 'salida' =
        '" . $salida[1] . "'" ) or die(mysql_error());
99     $connect->exec("UPDATE 'vuelos' SET 'plazas_optima' =
        'plazas_optima' + 1 WHERE 'vuelo' = '" . $vuelo[1] . "' AND 'salida' =
        '" . $salida[1] . "'" ) or die(mysql_error());
100 ▾ } elseif ($npas_economy[1] > 1){
101     $connect->exec("UPDATE 'compras' SET 'npas_economy' =
        'npas_economy' - 1 WHERE 'vuelo' = '" . $vuelo[1] . "' AND 'salida' =
        '" . $salida[1] . "'" ) or die(mysql_error());
102     $connect->exec("UPDATE 'vuelos' SET 'plazas_economy' =
        'plazas_economy' + 1 WHERE 'vuelo' = '" . $vuelo[1] . "' AND 'salida' = '" . $salida[1] . "'");
103     }
```

En esta parte se borraría solo las plazas que indique el cliente de la lista de pasajeros.



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.

```
3 angular.module("trabajo", ['ngRoute'])
4 .controller("ctrl", function ($scope, $http, $filter) {
5     var contadorIDA = 0;
6     $scope.elementoIDA = {};
7     $scope.elementoSalida = {};
8     $scope.elementoIntermedioSalida = {};
9     $scope.elementoIntermedioSalidaV = {};
10    $scope.elementoSalidaV = {};
11    $scope.deshabilitarIDA = false;
12
13    var contadorIntermedioIDA = 0;
14    $scope.elementoIntermedioIDA = {};
15    $scope.deshabilitarIntermedioIDA = false;
16    var contadorIntermedioVUELTA = 0;
17    $scope.elementoIntermedioVUELTA = {};
18    $scope.deshabilitarIntermedioVUELTA = false;
19
20    var contadorDestinoIDA = 0;
21    $scope.elementoDestinoIDA = {};
22    $scope.deshabilitarDestinoIDA = false;
23    var contadorDestinoVUELTA = 0;
24
25    $scope.elementoDestinoVUELTA = {};
26    $scope.deshabilitarDestinoVUELTA = false;
27
28    var contadorVUELTA = 0;
29    $scope.elementoVUELTA = {};
30    $scope.nombre = null;
31    $scope.apellidos = null;
32    $scope.dni = null;
33    $scope.nameForm = null;
34
35    var pasajeros = [];
36
37    $scope.tipoBilleteIda = '';
38    $scope.tipoBilleteVuelta = '';
39
40    $scope.deshabilitarVUELTA = false;
41    $scope.init = function () {
```

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

```
78    //paginacion
79    $scope.prevPage = function () {
80        if ($scope.currentPage > 0) {
81            $scope.currentPage--;
82        }
83    };
84
85    $scope.nextPage = function () {
86        if (parseInt($scope.currentPage) < parseInt($scope.pageSize) - 1) {
87            $scope.currentPage++;
88        }
89    };
90
91
92    $scope.submitForm = function () {
93        var billetes = document.getElementById('nBilletes').value;
94        for (var i = 0; i < billetes; i++) {
95            pasajeros.push({
96                'nombre': document.getElementById('nombre_' + i).value,
97                'apellidos': document.getElementById('apellidos_' + i).value,
98                'dni': document.getElementById('dni_' + i).value
99            });
100        }
101        var elementoVUELTA = $scope.elementoVUELTA;
102        var elementoSalidaV = $scope.elementoSalidaV;
103        var tipoBilleteVuelta = $scope.tipoBilleteVuelta;
104        if (angular.equals($scope.elementoVUELTA, {})) {
105            elementoVUELTA = "";
106            elementoSalidaV = "";
107            tipoBilleteVuelta = "";
108        }
109        var elementoDestinoIDA = $scope.elementoDestinoIDA;
110        var elementoIntermedioSalida = $scope.elementoIntermedioSalida;
111        var tipoBillete_IntermedioIda = $scope.tipoBillete_IntermedioIda;
112        if (angular.equals($scope.elementoDestinoIDA, {})) {
113            elementoDestinoIDA = "";
114            elementoIntermedioSalida = "";
115            tipoBillete_IntermedioIda = "";
116        }
117    }
```

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



```
158 ▾ $scope.submitFormCompany = function () {  
159     $http.post(  
160         "php/insertar_vuelo.php", {  
161             'origen': $scope.searchOrigin,  
162             'destino': $scope.searchDestiny,  
163             'salida': $scope.fechaCheckIn,  
164             'llegada': $scope.fechaCheckOut,  
165             'plazas_business': $scope.plazas_business,  
166             'plazas_optima': $scope.plazas_optima,  
167             'plazas_economy': $scope.plazas_economy,  
168             'precio_business': $scope.precio_business,  
169             'precio_optima': $scope.precio_optima,  
170             'precio_economy': $scope.precio_economy,  
171         }  
172     )  
173 }  
174 ▾ .then(function (respuesta) {  
175     console.log(respuesta);  
176     $scope.origen = null;  
177     $scope.destino = null;  
178     $scope.salida = null;  
179     $scope.llegada = null;  
180     $scope.plazas_business = null;  
181     $scope.plazas_optima = null;  
182     $scope.plazas_economy = null;  
183     $scope.precio_business = null;  
184     $scope.precio_optima = null;  
185     $scope.precio_economy = null;  
186     window.location = ('company.php');  
187 })  
188 };  
189  
190
```

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

```
248     /* Seleccionar orden de la tabla */  
249 ▾ $scope.setOrderIDA = function (x) {  
250     $scope.selectedOrderIDA = x;  
251 }  
252 };  
253 ▾ $scope.setOrderVUELTA = function (x) {  
254     $scope.selectedOrderVUELTA = x;  
255 }  
256 };  
257  
258     /* Selecciona elemento de la checkbox (el identificador del vuelo) y  
259     lo almacena o lo borra */  
260     /* También activa o desactiva el poder mostrar el precio */  
261 ▾ $scope.seleccionIDA = function (posicion, salida) {  
262     contadorIDA++;  
263     $scope.ida = false;  
264 ▾ if (contadorIDA % 2 != 0) {  
265         $scope.elementoIDA = posicion;  
266         $scope.elementoSalida = salida;  
267         //console.log($scope.elementoSalida);  
268         $scope.estadoDeshabilitarIDA = true;  
269         $scope.ida = true;  
270     } else {  
271 ▾ contadorIDA = 0;  
272         $scope.elementoIDA = {};  
273         $scope.elementoSalida = {};  
274         $scope.estadoDeshabilitarIDA = false;  
275         $scope.ida = false;  
276         $scope.precioIDA = 0;  
277     }  
278 }  
279 };  
280     /// para elegir la escala en la ida  
281 ▾ $scope.seleccionIntermedioIDA = function (posicion) {  
282     contadorIntermedioIDA++;  
283     $scope.intermedioIDA = false;  
284 ▾ if (contadorIntermedioIDA % 2 != 0) {  
285         $scope.elementoIntermedioIDA = posicion;  
286         $scope.estadoDeshabilitarIntermedioIDA = true;  
287         $scope.intermedioIDA = true;  
288     }  
289 }  
290 };
```

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.



```
399      /* Retorna el valor almacenado en el elemento (el identificador del
400      vuelo), si no coincide con el identificador de vuelo nuevo el
401      checkbox quedara deshabilitado */
402      $scope.deshabilitarIDA = function () {
403          return $scope.elementoIDA;
404      };
405      $scope.deshabilitarIntermedioIDA = function () {
406          return $scope.elementoIntermedioIDA;
407      };
408      $scope.deshabilitarIntermedioVUELTA = function () {
409          return $scope.elementoIntermedioVUELTA;
410      };
411      $scope.deshabilitarDestinoIDA = function () {
412          return $scope.elementoDestinoIDA;
413      };
414      $scope.deshabilitarDestinoVUELTA = function () {
415          return $scope.elementoDestinoVUELTA;
416      };
417      $scope.deshabilitarVUELTA = function () {
418          return $scope.elementoVUELTA;
419      };
420      /* Actualiza el precio del billete */
421      $scope.EstablecerPrecioIDA = function (x) {
422          $scope.precioIDA = x;
423      };
424      $scope.EstablecerPrecioIntermedioIDA = function (x) {
425          $scope.precioIntermedioIDA = x;
426      };
427      $scope.EstablecerPrecioDestinoIDA = function (x) {
428          $scope.precioDestinoIDA = x;
429      };
430      $scope.EstablecerPrecioDestinoVUELTA = function (x) {
431          $scope.precioDestinoVUELTA = x;
432      };
433      $scope.EstablecerPrecioVUELTA = function (x) {
434          $scope.precioVUELTA = x;
435      };
436      $scope.EstablecerPrecioVUELTA = function (x) {
437          $scope.precioVUELTA = x;
438      };
```

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 se actualice el precio del billete.

```
440      /* Al reservar un vuelo elimina el valor del elemento (el
441      identificador de vuelo) para que si se decide posteriormente realizar
442      otra reserva no aparezcan los checkbox deshabilitados */
443      $scope.limpiar = function () {
444          if ($scope.elementoIDA.length != undefined) {
445              $scope.seleccionIDA($scope.elementoIDA);
446          }
447          if ($scope.elementoIntermedioIDA.length != undefined) {
448              $scope.seleccionIntermedioIDA($scope.elementoIntermedioIDA);
449          }
450          if ($scope.elementoIntermedioVUELTA.length != undefined) {
451              $scope.seleccionIntermedioVUELTA($scope.elementoIntermedioVUELTA);
452          }
453          if ($scope.elementoVUELTA.length != undefined) {
454              $scope.seleccionVUELTA($scope.elementoVUELTA);
455          }
456      };
457      $scope.setTipoIda = function (tipo) {
458          $scope.tipoBilleteIda = tipo;
459      };
460      $scope.setTipoIntermedioIda = function (tipo) {
461          $scope.tipoBillete_IntermedioIda = tipo;
462      };
463      $scope.setTipoIntermedioVuelta = function (tipo) {
464          $scope.tipoBillete_IntermedioVuelta = tipo;
465      };
466      $scope.setTipoVuelta = function (tipo) {
467          $scope.tipoBilleteVuelta = tipo;
468      };
469      $scope.resetDatosCancReserva = function () {
470          $scope.setDNI = '';
471          $scope.setNombre = '';
472          $scope.setApellidos = '';
473      };
```

Esta parte se utiliza para que al reservar un vuelo se elimine el valor identificativo de ese vuelo para que si se realiza otra reserva no aparezcan los checkbox que ya se han deshabilitado.



```
480 //indica desde que valor tiene que empezar a contar, se usa para la
    paginacion de la pagina company.html
481 ▼ .filter('startFrom', function () {
482 ▼     return function (input, start) {
483 ▼         if (input) {
484             start = +start; //parse to int
485             var appended = input.slice(0, start);
486             var initialArray = input.slice(start);
487             var finalArray = initialArray.concat(appended);
488             return finalArray;
489         }
490         return [];
491     };
492 })
493 ▼ .filter('removeSpaces', [function () {
494 ▼     return function (string) {
495 ▼         if (!angular.isString(string)) {
496             return string;
497         }
498         return string.replace(/\s/g, '');
499     }
500 }])
501 /* Filtra los nombres de Origen y Destino de tal manera que no se envíen
    duplicados */
502
503 ▼ .filter('unique', function () {
504 ▼     return function (collection, keyname) {
505         var output = [];
506         keys = [];
507
508         angular.forEach(collection, function (item) {
509             var key = item[keyname];
510             if (keys.indexOf(key) === -1) {
511                 keys.push(key);
512                 output.push(item);
513             }
514         });
515         return output;
516     };
517 })
```

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.

```
518 /* Analiza letra a letra la cadena de texto y muestra caracteres
    especiales */
519
520 ▼ .filter('utf8_decode', function ($sce) {
521 ▼     return function (strData) {
522         var tmpArr = [];
523         var i = 0;
524         var c1 = 0;
525         var seqLen = 0;
526         var anterior = 0;
527         strData += ' ';
528         while (i < strData.length) {
529             c1 = strData.charCodeAt(i) & 0xFF;
530             seqLen = 0;
531             if ((c1 == 253 && anterior == 73)) {
532                 c1 = 0xC1;
533                 seqLen = 1;
534             } else if ((c1 == 253 && anterior == 79)) {
535                 c1 = 0xD1;
536                 seqLen = 1;
537             } else if ((c1 == 253 && anterior == 85)) {
538                 c1 = 0xD1;
539                 seqLen = 1;
540             } else if (c1 <= 0xBF) {
541                 c1 = (c1 & 0x7F);
542                 seqLen = 1;
543                 anterior = c1;
544             } else if (c1 <= 0xDF) {
545                 c1 = (c1 & 0x1F);
546                 seqLen = 2;
547             } else if (c1 <= 0xFF) {
548                 c1 = 0xBF;
549                 seqLen = 3;
550             } else {
551                 c1 = (c1 & 0xFD);
552             }
553             tmpArr.push(c1);
554             i++;
555         }
556         return tmpArr;
557     };
558 })
```

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



```
571  /* Mismo código pero codifica los caracteres, utilizado para las
572     consultas a la bbdd */
573  .filter('utf8_encode', function ($sce) {
574      return function (strData) {
575          var tmpArr = [];
576          var i = 0;
577          var cl = 0;
578          var seqLen = 0;
579          var anterior = 0;
580          strData += ' ';
581          while (i < strData.length) {
582              cl = strData.charCodeAt(i) & 0xFF;
583              seqLen = 0;
584              if ((cl == 0xC1 && anterior == 73)) {
585                  cl = 253;
586                  seqLen = 1;
587              } else if ((cl == 0xD1 && anterior == 79)) {
588                  cl = 253;
589                  seqLen = 1;
590              } else if ((cl == 209 && anterior == 85)) {
591                  cl = 253;
592                  seqLen = 1;
593              } else if (cl <= 0xBF) {
594                  cl = (cl & 0x7F);
595                  seqLen = 1;
596                  anterior = cl;
597              } else if (cl <= 0xDF) {
598                  cl = (cl & 0x1F);
599                  seqLen = 2;
600              } else if (cl <= 0xFF) {
601                  cl = 0xBF;
602                  seqLen = 3;
603              } else {
604                  cl = (cl & 0xFD);
605                  seqLen = 4;
606              }
607          }
608          return tmpArr;
609      }
610  }
```

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.

```
624  .config(function ($routeProvider) {
625      $routeProvider.
626      when('/', {
627          controller: 'ctrl',
628          templateUrl: 'main.php'
629      }).
630      when('/prueba/:origen/:destino/:nBilletes/:fcheckIn', {
631          controller: 'CountDownCtrl',
632          templateUrl: 'prueba.php'
633      }).
634      when('/prueba/:origen/:destino/:nBilletes/:fcheckIn/:fcheckOut', {
635          controller: 'CountDownCtrlb',
636          templateUrl: 'prueba.php'
637      }).
638      otherwise({
639          redirectTo: '/'
640      });
641  })
642  .controller('MainCtrl', function ($scope) {
643  })
644  .controller('CountDownCtrl', function ($scope, $routeParams, $timeout) {
645      $scope.origen = $routeParams.origen;
646      $scope.destino = $routeParams.destino;
647      $scope.fcheckIn = $routeParams.fcheckIn;
648      $scope.nBilletes = $routeParams.nBilletes;
649  })
650  .controller('CountDownCtrlb', function ($scope, $routeParams, $timeout) {
651      $scope.origen = $routeParams.origen;
652      $scope.destino = $routeParams.destino;
653      $scope.fcheckIn = $routeParams.fcheckIn;
654      $scope.fcheckOut = $routeParams.fcheckOut;
655      $scope.nBilletes = $routeParams.nBilletes;
656  })
657  });
658  });
659  });
660  });
661  });
```

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.



BOTON.CSS

Con este código CSS lo que hacemos es configurar la apariencia y el estilo de los botones de nuestra página.

```
1  .botonSiguiente{
2    position: absolute;
3    right: 0;
4    padding-top: 2 rem;
5  }
6
7  .botonSiguiente {
8    color: #fff !important;
9    font-size: 20px;
10   font-weight: 500;
11
12   background: #167F92;
13   border: 2px solid;
14   border-color: #167F92;
15 }
16
17 .botonSiguiente:before {
18   content: "";
19   position: absolute;
20   top: 0px;
21   left: 0px;
22   width: 0px;
23   height: 100%;
24   background: rgba(255, 255, 255, 0.1);
25   transition: all 1s ease;
26 }
27 .botonSiguiente:hover:before {
28   width: 100%;
29 }
30 .botonAtras{
31   position: absolute;
32   left: 0;
33   padding-top: 2 rem;
34 }
```

```
35
36 .botonAtras {
37   color: #fff !important;
38   font-size: 20px;
39   font-weight: 500;
40   background: rgba(255, 255, 255, 0.1);
41   background: #167F92;
42   border: 2px solid;
43   border-color: #167F92;
44 }
45
46 .botonAtras:before {
47   content: "";
48   position: absolute;
49   top: 0px;
50   left: 0px;
51   width: 100%;
52   height: 100%;
53   background: rgba(255, 255, 255, 0.1);
54   transition: all 1s ease;
55 }
56 .botonAtras:hover:before {
57   width: 0px;
58 }
```