



## Práctica 2 AngularJS

### Diagramas de arquitectura para contextualizar los ejercicios:

[https://drive.google.com/file/d/0B\\_7-fxyOFEa7dEtUU2k2d1BvME0/view?usp=sharing](https://drive.google.com/file/d/0B_7-fxyOFEa7dEtUU2k2d1BvME0/view?usp=sharing)

Diagrama de arquitectura AngularJS detallado en el lado izquierdo. La aplicación HTML 5 y AngularJS se ejecuta en el navegador. En el lado derecho aparece el servidor Spring desplegado en una plataforma en la nube. Ambos componentes, la capa cliente HTML5 y AngularJS, y la capa servidora Spring se comunican mediante un API REST (HTTP para el transporte y JSON para el formato de los datos):

[https://drive.google.com/file/d/0B\\_7-fxyOFEa7Z0dIZ1BkaGxPTTQ/view?usp=sharing](https://drive.google.com/file/d/0B_7-fxyOFEa7Z0dIZ1BkaGxPTTQ/view?usp=sharing)

**Funcionalidad:** Se ofrece un botón para consultar los jugadores que han conseguido 50 canastas o más.

Repositorio ejemplo (<https://github.com/alfredoruera/liga-baloncesto-practica2angular>)

## Back-end

### Repositorio Jugador:

JugadorRepository.java

/src/main/java/com/xavipandis/liga/repository/JugadorRepository.java

```
@Query("select jugador from Jugador jugador where jugador.canastasTotales  
>=:canastas")  
Page<Jugador> topPlayers(@Param("canastas") Integer canastas, Pageable  
pageable);
```

### Controller Jugador:

JugadorResource.java

/src/main/java/com/xavipandis/liga/web/rest/JugadorResource.java

```
@RequestMapping(value = "/topPlayers/{canastas}",  
method = RequestMethod.GET,
```

```

        produces = MediaType.APPLICATION_JSON_VALUE)
@Timed
public ResponseEntity<List<Jugador>> topPlayers(@PathVariable Integer
canastas, Pageable pageable)
    throws URISyntaxException {
    Page<Jugador> page = jugadorRepository.topPlayers(canastas, pageable);
    HttpHeaders headers = PaginationUtil.generatePaginationHttpHeaders(page,
"/api/jugadors/topPlayers");
    return new ResponseEntity<>(page.getContent(), headers, HttpStatus.OK);
}

```

## Front-end

Index.html:

/src/main/webapp/index.html

```

<script src="scripts/app/entities/jugador/topPlayers-
Controller.js"></script>

```

(Este div ya existe en el proyecto original generado por Jhipster)

```

<div class="container">

```

(En ui-view='content' se inyecta el template de topPlayers  
(scripts/app/entities/jugador/jugadorsTabla.html))

```

    <div class="well" ui-view="content"></div>

    <div class="footer">

        <p translate="footer">This is your footer</p>

    </div>

</div>

```

Template ORIGINAL CRUD jugadores:

jugadors.html

/src/main/webapp/scripts/app/entities/jugador/jugadors.html

```

<button class="btn btn-primary" ui-sref="topPlayers">
    <span class="glyphicon glyphicon-flash"></span> <span>Top Players</span>
</button>

```

(El ui-sref='topPlayers' enlaza con la transición de estado topPlayers definida en el  
archivo de rutas jugador.js)

Fichero de rutas:

jugador.js

/src/main/webapp/scripts/app/entities/jugador/jugador.js

```
.state('topPlayers', {
  parent: 'entity',
  url: '/jugadors/topPlayers',
  data: {
    authorities: ['ROLE_USER'],
    pageTitle: 'ligaBaloncestoApp.jugador.detail.title'
  },
  views: {
    'content@': {
      templateUrl: 'scripts/app/entities/jugador/jugadorsTabla.html',
      controller: 'TopPlayersController'
    }
  },
  resolve: {
    translatePartialLoader: ['$translate', '$translatePartialLoader',
function ($translate, $translatePartialLoader) {
  $translatePartialLoader.addPart('jugador');
  return $translate.refresh();
}],
    entity: ['$stateParams', 'Jugador', function($stateParams, Jugador) {
      return Jugador.topPlayers({canastas : 50});
    }]
  ]
})
})
```

### Service Jugador comunicación con back-end:

jugador.service.js

/src/main/webapp/scripts/components/entities/jugador/jugador.service.js

```
'topPlayers': {
  method: 'GET', isArray: true, url: 'api/topPlayers/:canastas',
  interceptor: {
    response: function(response) {
      response.resource.$httpHeaders = response.headers;
      return response.resource;
    }
  }
}
}
```

### topPlayers controller:

topPlayers-Controller.js

/src/main/webapp/scripts/app/entities/jugador/topPlayers-Controller.js

```
'use strict';

angular.module('ligaBaloncestoApp')
  .controller('TopPlayersController', function ($scope, $rootScope,
    $stateParams, entity, Jugador, Equipo, ParseLinks) {

    entity.$promise.then(function(data) {
      console.log(data);
      var link = entity.$httpHeaders('link');

      $scope.links = ParseLinks.parse(link);
    });
  });
```

```

        $scope.savedPlayers = data;
    });

```

```

});

```

## Template topPlayers:

jugadorsTabla.html

/src/main/webapp/scripts/app/entities/jugador/jugadorsTabla.html

```

<div class="table-responsive">
    <table class="table table-striped">
        <thead>
            <tr>
                <th translate="global.field.id">ID</th>
                <th translate="ligaBaloncestoApp.jugador.nombre">Nombre</th>
                <th
translate="ligaBaloncestoApp.jugador.fechaNacimiento">FechaNacimiento</th>
                <th translate="ligaBaloncestoApp.jugador.posicion">Posicion</th>
                <th
translate="ligaBaloncestoApp.jugador.canastasTotales">CanastasTotales</th>
                <th
translate="ligaBaloncestoApp.jugador.asistenciasTotales">AsistenciasTotales</t
h>
                <th
translate="ligaBaloncestoApp.jugador.rebotesTotales">RebotesTotales</th>
                <th translate="ligaBaloncestoApp.jugador.equipo">equipo</th>
            </tr>
        </thead>
        <tbody>
            <tr ng-repeat="jugador in savedPlayers track by jugador.id">
                <td><a ui-
sref="jugador.detail({id:jugador.id})">{{jugador.id}}</a></td>
                <td>{{jugador.nombre}}</td>
                <td>{{jugador.fechaNacimiento | date:'mediumDate'}}</td>
                <td>{{jugador.posicion}}</td>
                <td>{{jugador.canastasTotales}}</td>
                <td>{{jugador.asistenciasTotales}}</td>
                <td>{{jugador.rebotesTotales}}</td>
                <td>
                    <a ui-
sref="equipo.detail({id:jugador.equipo.id})">{{jugador.equipo.nombre}}</a>
                </td>
                <td>
                    <button type="submit"
                        ui-sref="jugador.detail({id:jugador.id})"
                        class="btn btn-info btn-sm">
                        <span class="glyphicon glyphicon-eye-
open"></span>&nbsp;<span translate="entity.action.view"> View</span>
                    </button>
                    <button type="submit"
                        ui-sref="jugador.edit({id:jugador.id})"
                        class="btn btn-primary btn-sm">
                        <span class="glyphicon glyphicon-
pencil"></span>&nbsp;<span translate="entity.action.edit"> Edit</span>
                    </button>
                    <button type="submit"

```

```

        ui-sref="jugador.delete({id:jugador.id})"
        class="btn btn-danger btn-sm">
        <span class="glyphicon glyphicon-remove-
circle"></span>&nbsp;<span translate="entity.action.delete"> Delete</span>
        </button>
    </td>
</tr>
</tbody>
</table>
<nav>
    <ul class="pagination">
        <li ng-show="links['first'] >= 0" ng-
click="loadPage(links['first'])"><a>&lt;&lt;</a></li>
        <li ng-show="page > 0" ng-click="loadPage(links['prev'] ||
0)"><a>&lt;</a></li>
        <li ng-show="page > 1" ng-click="loadPage(page - 2)"><a>{{page -
1}}</a></li>
        <li ng-show="page > 0" ng-click="loadPage(page -
1)"><a>{{page}}</a></li>
        <li class="active"><a>{{page + 1}}</a></li>
        <li ng-show="page < links['last']" ng-click="loadPage(page +
1)"><a>{{page + 2}}</a></li>
        <li ng-show="page < links['last'] - 1" ng-click="loadPage(page +
2)"><a>{{page + 3}}</a></li>
        <li ng-show="links['next']" ng-
click="loadPage(links['next'])"><a>&gt;</a></li>
        <li ng-show="links['last'] > 0" ng-
click="loadPage(links['last'])"><a>&gt;&gt;</a></li>
    </ul>
</nav>

    <button type="submit"
        onclick="window.history.back()"
        class="btn btn-info">
        <span class="glyphicon glyphicon-arrow-left"></span>&nbsp;<span
translate="entity.action.back"> Back</span>
    </button>
</div>

```

## EJERCICIO:

Implementar la misma funcionalidad, pero en este caso, añadir un botón en el CRUD de jugadores para obtener los jugadores que han conseguido 20 asistencias o más.

El botón tendrá que estar a la derecha del botón de top players. Los jugadores tendrán que aparecer ordenados por el número de asistencias.