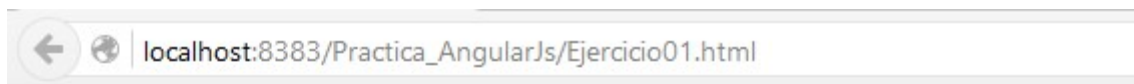


## Práctica AngularJS

### Ejercicio 1

Desarrolla una página web equivalente que permita introducir los campos de un jugador de baloncesto (te puedes basar en los atributos que hemos utilizado con Spring), y muestre en tiempo real el valor introducido. Explica, de manera muy resumida, el mecanismo que utiliza AngularJS para sincronizar los elementos de la interfaz de usuario HTML (explica la directiva específica que se utiliza en este ejemplo).



Introduce los datos del jugador:

Nombre :

Apellidos :

País :

Posición en el campo :

Nombre y apellidos del jugador : David Casaoliva Rey

País de origen del jugador : España

Posición en el campo : Pibot

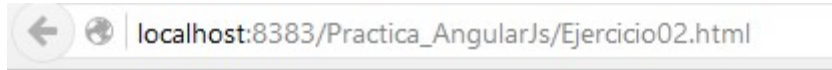
### Explicación

En este ejercicio se utiliza la propiedad ng-model, lo que permite, es que a la vez que se está introduciendo un nuevo valor instantáneamente está actualizando el controlador. Al escribir el nombre del jugador a la vez se está actualizando el controlador que lo muestra en la parte de debajo de la pantalla.

## Ejercicio 2

Extiende el ejercicio número 1, de manera que se incluya un controlador que gestione los datos de la vista. El controlador ha de proporcionar unos datos iniciales por defecto que se mostrarán en la vista. Cuando el usuario actualiza los elementos de la vista (los elementos de la interfaz de usuario en HTML), ¿se actualizan automáticamente los datos en el controlador? Justifica tu respuesta.

Campos con datos por defecto:



Cambia los datos por defecto del jugador.

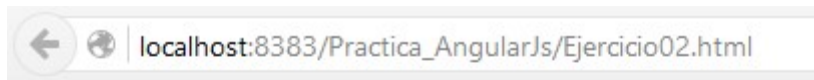
Nombre:   
Apellidos:

Nombre completo: David Casaoliva Rey

### Explicación:

Con las funciones ng-app y ng-controller, se gestiona el mostrar los valores predeterminados en los campos donde podemos modificarlos. La función ng-app nos permite crear un controlador con el scope y poder definir valores por defecto para los campos que queremos modificar.

Campos con modificaciones de los datos por defecto en los inputs:



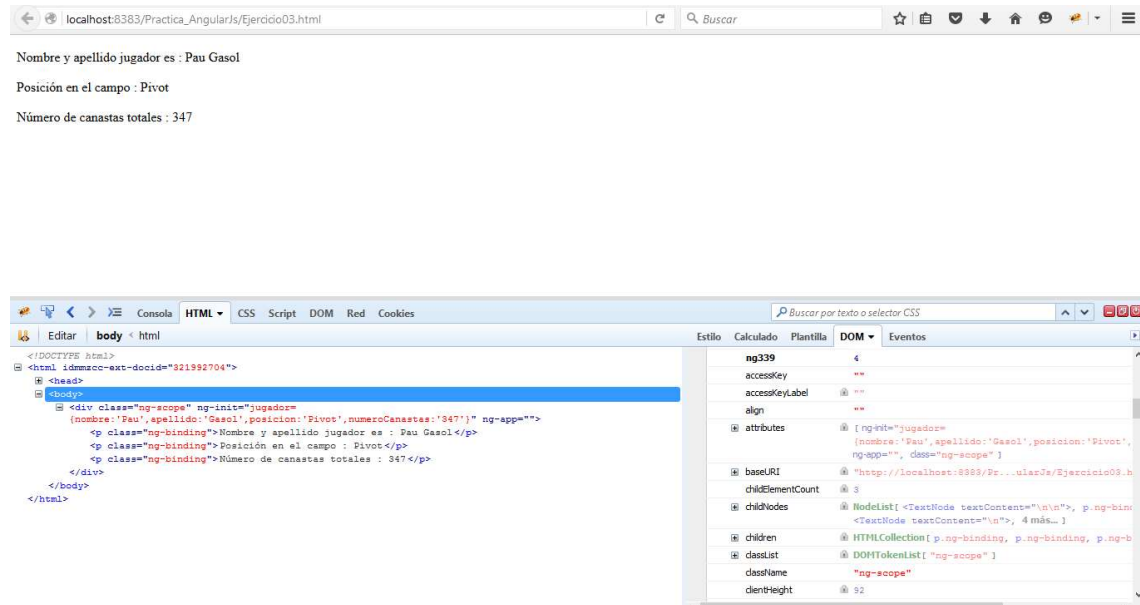
Cambia los datos por defecto del jugador.

Nombre:   
Apellidos:

Nombre completo: Pepe Perez Rey

## Ejercicio 3

Utilizando la directiva ng-init, inicializa un objeto jugador de baloncesto con sus diversos atributos. Posteriormente, muestra sus atributos en diversos elementos HTML mediante la directiva {{}}. Explica en qué componentes de AngularJS se debe manipular el DOM: directivas, controladores, vistas o servicios. Justifica tu respuesta.



## Explicación

Permite evaluar una expresión en el scope, creando el objeto jugador i definiendo sus atributos. Al examinar el DOM, comprobamos que están los datos del objeto cargados y en el scope.

## Ejercicio 4

Inicializa un array para almacenar diversos jugadores de baloncesto. Posteriormente, utiliza la directiva ng-repeat para mostrar todos los jugadores de baloncesto con sus diversos atributos. Explica cómo funciona la gestión de scopes con la directiva ng-repeat.

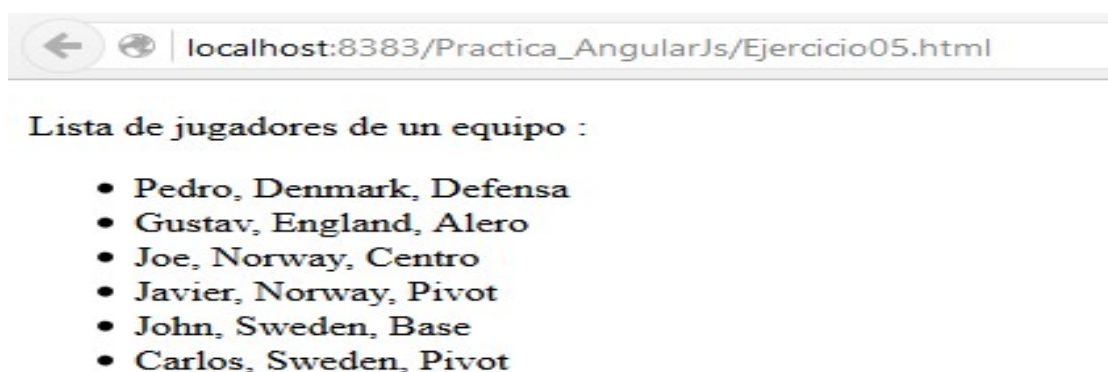


### Explicación:

La directiva ng-repeat nos permite mostrar los objetos contenidos en una variable. Al definir una serie de objetos, en este caso jugadores, podemos ir mostrando a usar esta directiva combinada en un li, mostramos una lista con los atributos que hemos guardado de cada jugador dentro de la variable jugadores.

## Ejercicio 5:

Inicializa un array que almacene diversos jugadores de baloncesto en un controlador. Posteriormente, utiliza la directiva ng-repeat para mostrar todos los jugadores. Utiliza también un filtro, para mostrar los jugadores ordenando mediante varios atributos (por ejemplo, nombre, total canastas, total asistencias, país, etc.). La idea es que puedes experimentar con diversos atributos para comprobar que la ordenación funciona correctamente con todos los tipos de datos. Justifica en qué casos es preferible realizar la ordenación en el cliente mediante AngularJS, y en qué casos es preferible realizar la ordenación en el servidor mediante Spring. Explica ventajas e inconvenientes de cada enfoque.



### Explicación:

La ordenación es más eficiente y segura en Spring, por qué el usuario no puede acceder directamente al scope y poderlo manipular. En cambio, en AngularJS es menos seguro y el usuario puede manipular los resultados. Es mejor en Spring por el tema de la seguridad y la eficiencia. El ejemplo está ordenado por país.

## Ejercicio 6:

Repaso sobre la sincronización automática que realiza angular entre la vista, el controlador y el modelo. Experimenta con el siguiente código de ejemplo, comprobando la sincronización que se efectúa con los atributos de un jugador de baloncesto.



localhost:8383/Practica\_AngularJs/Ejercicio06.html

David Casaoliva

**El nombre del jugador es : David Casaoliva**

When you change the name in the input field, the changes will affect the model, and it will also affect the name property in the controller.

Modificación del input:



localhost:8383/Practica\_AngularJs/Ejercicio06.html

David Casaoliva 1971

**El nombre del jugador es : David Casaoliva 1971**

When you change the name in the input field, the changes will affect the model, and it will also affect the name property in the controller.

## Explicación:

Con ng-controller, conseguimos que se actualice la salida directamente con lo que introducimos en el input, en este caso, al incorporar al nombre del jugador ya guardado en el scope con un número.

## Ejercicio 7:

Con JHipster y Spring, generamos un listado de equipos de la base de datos.

- FC Barcelona, Barcelona, 1895-09-08
- Bayer Munich, Munich, 1923-05-20
- Real Madrid, Madrid, 1901-03-25
- Paris Saint Germaine, Paris, 1956-09-12
- Real Asturia FC, Oviedo, 1961-12-31

## Explicación:

Generamos un controler (equiposCtrl), que nos permite conectar con la tabla de equipos y a partir de la función ng-repeat obtenemos un listado de los equipos.

## Ejercicio 9

Extiende el ejercicio número 7 para obtener el listado de jugadores del API REST Spring. En este ejercicio, debes mostrar la información de los jugadores en formato tabla, especificando una columna para cada atributo del jugador.

FC Barcelona	Barcelona	1895-09-08
Bayer Munich	Munich	1923-05-20
Real Madrid	Madrid	1901-03-25
Paris Sant Germaine	Paris	1956-09-12
Real Asturia FC	Oviedo	1961-12-31

**Explicación:** en este ejercicio se muestra en modo tabla el contenido de equipos.

## Ejercicio10:

Mejora la tabla del ejercicio anterior utilizando estilos CSS. Te puedes basar en el código de ejemplo proporcionado, y puedes proponer tus propios estilos.

FC Barcelona	Barcelona	1895-09-08
Bayer Munich	Munich	1923-05-20
Real Madrid	Madrid	1901-03-25
Paris Sant Germaine	Paris	1956-09-12
Real Asturia FC	Oviedo	1961-12-31

## Ejercicio11:

En este ejercicio, debes utilizar la nueva directiva ng-if para aplicar diversos estilos a las columnas en función de que la fila sea par o impar.

FC Barcelona	Barcelona	1895-09-08
Bayer Munich	Munich	1923-05-20
Real Madrid	Madrid	1901-03-25
Paris Sant Germaine	Paris	1956-09-12
Real Asturia FC	Oviedo	1961-12-31

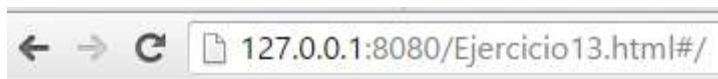
**Explicación:** con la directiva ng-if asignamos de manera par o impar el CSS en la tabla.

## Ejercicio12:

Con ng-show podemos mostrar un contenido sí cumple una condición.

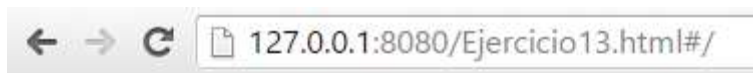
## Ejercicio 13:

Con la opción filter en ng-repeat, nos permite filtrar la lista de equipos a partir de los caracteres que vamos escribiendo en el input y en la lista van quedando los nombres que va coincidiendo.



Introduce una letra para filtrar:

- FC Barcelona, Barcelona
- Bayer Munich, Munich
- Real Madrid, Madrid
- Paris Saint Germaine, Paris
- Real Asturia FC, Oviedo
- Lugo Club de Futbol, Lugo
- FC Valencia, Valencia
- FC Alicante, Alicante



Introduce una letra para filtrar:

- FC Barcelona, Barcelona
- Bayer Munich, Munich
- Lugo Club de Futbol, Lugo



### Ejercicio 14:

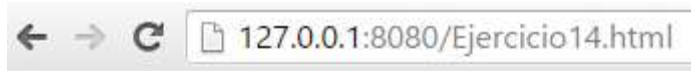
Con el controller ng-click y la función `orderBy` de Srping, podemos hacer que al hacer clic en la cabecera de cada columna, esta se ordene alfabéticamente en este caso.



Pulsa los headers para ordenar los campos

Nombre equipo	Localidad
FC Barcelona	Barcelona
Real Madrid	Madrid
Bayer Munich	Munich
Real Asturia FC	Oviedo
Paris Sant Germane	Paris

Ordenamos por nombre equipo:



Pulsa los headers para ordenar los campos

Nombre equipo	Localidad
Bayer Munich	Munich
FC Barcelona	Barcelona
Paris Sant Germane	Paris
Real Asturia FC	Oviedo
Real Madrid	Madrid



### Ejercicio 15:

Este controller nos permite crear dos campos input que no permiten grabar nuevos equipos en la base de datos.



A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:8080/Ejercicio15.html". Below the address bar, there is a form with two input fields. The first field is labeled "Nombre:" and contains the text "FC Alicante". The second field is labeled "Localidad:" and contains the text "Alicante". Below these fields is a button labeled "Save".



A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:8080/Ejercicio15.html". Below the address bar, there is a table with two columns: "Nombre equipo" and "Localidad". The table contains the following data:

Nombre equipo	Localidad
FC Barcelona	Barcelona
Bayer Munich	Munich
Real Madrid	Madrid
Paris Saint Germaine	Paris
Real Asturia FC	Oviedo
Lugo Club de Futbol	Lugo
FC Valencia	Valencia
FC Alicante	Alicante

Pulsa los headers para ordenar los campos

Nombre equipo	Localidad
FC Barcelona	Barcelona
Bayer Munich	Munich
Real Madrid	Madrid
Paris Sant Germane	Paris
Real Asturia FC	Oviedo
Lugo Club de Futbol	Lugo
FC Valencia	Valencia
FC Alicante	Alicante