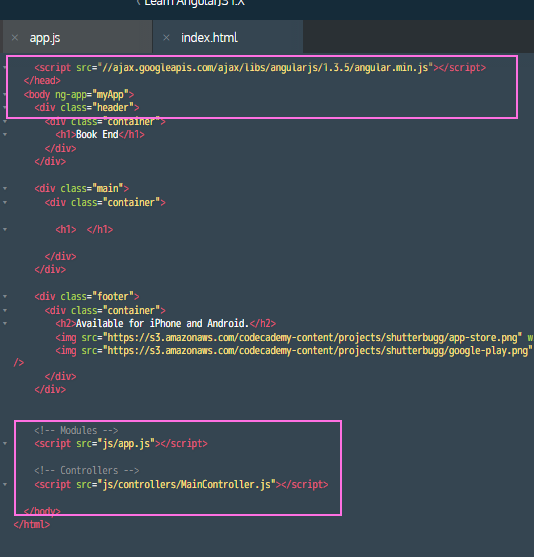
**Manual de Angular:**

**Curso 1:**

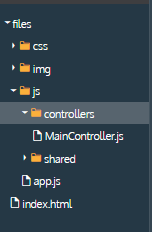
En el archivo app.js que usemos agregar esta línea

****

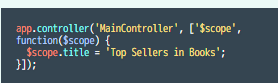
En el archivo HTML que los use hay que agregar una propiedad al body html y agregar unos scripts para su funcionamiento:



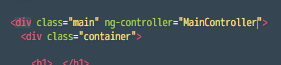
Angular Posee controladores.



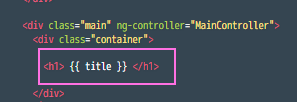
En el controlador de angular poner el siguiente código:



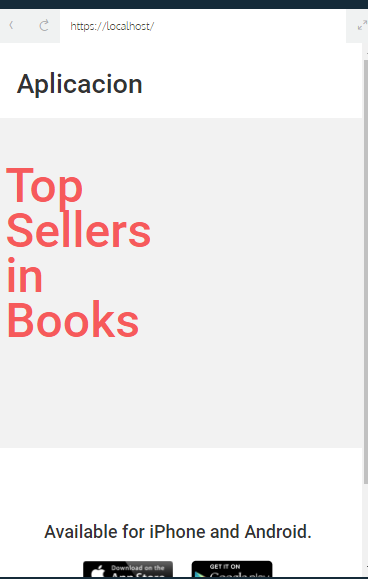
Lo asociamos al main del documento html



Dentro del HTML que contiene el Controller



Agregamos el siguiente argumento , esto transforma el contenido en la pagina web de esta forma.

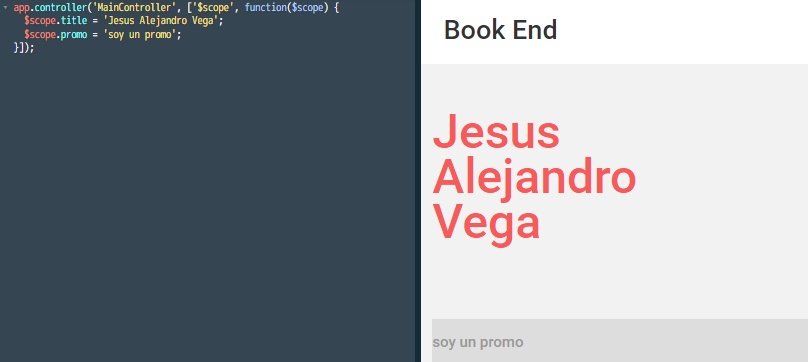


**Descripción de los elementos de una aplicación con Angular:**

1. Se necesita un modulo como por ejemplo app.js .
2. En el cuerpo (<body></body>) del documento html agregar la directiva ng-app=”myApp” , de esta forma convertimos ese html en un documento de ámbito angular.
3. En los controladores se maneja los datos del módulo.
4. Dentro de las etiquetas que vamos a manejar los procesos angular agregamos la propiedad de etiqueta ng-controller y especificamos el controlador a utilizar.
5. Podemos acceder a la variable scope.title con la sintaxis {{title}} , algo similar a laravel.

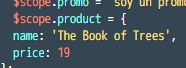
**Objeto scope**

En el controlador una vez que lo instanciamos en una sección HTML , mandar variables desde el objeto scope , podemos crear variables las cuales pueden ser enviadas a la vista con la siguiente sintaxis. (‘scope.algo’);

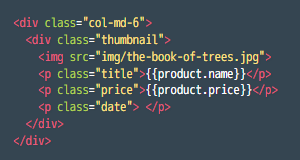


**Manejo de objetos**

Podemos instanciar objetos desde el el $scope , estos objetos se instancian así en el controlador:



Y se imprimen en pantalla de la siguiente forma :

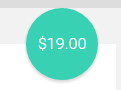


**Formato de datos**

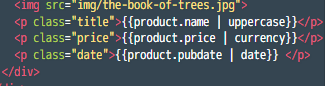
Angular posee herramientas propias para realizar formato de datos en la vista , para utilizarlos solo debemos usar la siguiente sintaxis:



Da como resultado



Angular en su núcleo posee varios filtros como por ejemplo:



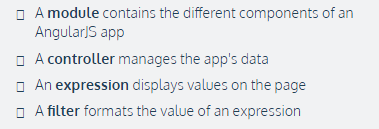
**Fechas**

Una forma de extraer la fecha es de esta forma:

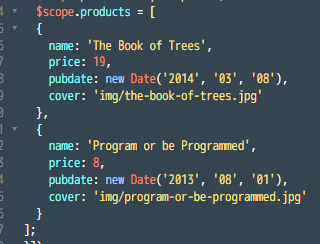


A través de la función Date();

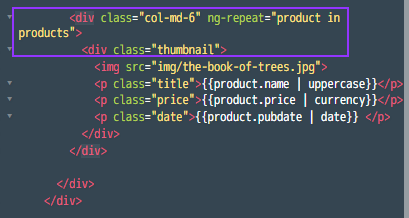
**Resumen de las propiedades de Angujar**

****

**Sintaxis de arreglo de objetos**

****

**Foreach usando Angular**

****

La sintaxis del foreach es bastante curiosa y se logrando usando una propiedad en la etiqueta contenedora llamada ng-repeat=”” donde instanciamos el objeto y su argumento para recorrerlo.

**Atributo para recoger imagen**

****

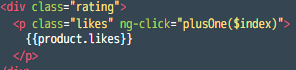
Para que detecte una ruta un recurso del servidor con un objeto impreso en pantalla por angular es primordial escribir la propiedad **ng-src** para que este funcione.

**Funciones con angular**



Con esta función podemos asumir que cada objeto que es recorrido en un arreglo con angular por “debajo” genera un index implícito el cual podemos capturar fácilmente , la ejecutar la función se hace lo siguiente :

Usamos la directiva **ng-click**



Al usar las directivas la pagina se modifica con las funciones puestas , sin necesidad de recargarse.

**Manejo de datos y utilización de directivas:**

**En el Main controller**

****

En El main controller se tienen los objetos move , shutterbugg , donde cada uno son objetos con las mismas propiedades

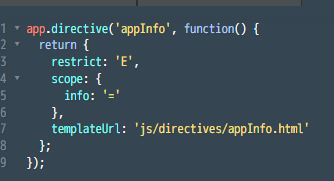
En el HTML :

Son recorridos una a uno y esto no es muy titel por esa razón usaremos una directiva.



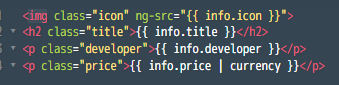
La estructura de carpetas es la siguiente:

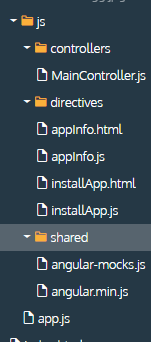
En la directiva appinfo.js



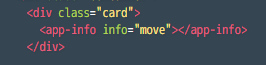
Agregamos una directiva a la cual le agregamos el template que es un archivo html.

El cual especifica un apuntador implícito que engloba varios objetos similares.



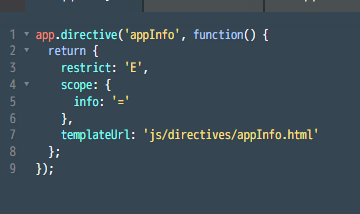


Finalmente podemos usar la directivas en el html principal , de esta forma reemplaza y resume la cantidad de líneas html en el documento principal.



Nótese que se debe usar las etiquetas <app-info></app-info> Esto se le denomina “enseñarle al navegador nuevas elementos html.”.

**Propiedades de las directivas**

****

Restrict => Especifica como la directiva va a utilizarse en la vista en este caso (E) , significa que será un nuevo elemento html.

Scope => Es el elemento que define un nombre con el cual se podrá alcanzar la directiva , ‘=’ , enseña que debe buscar una etiqueta con una estructura <app-info>

Template => Me permite obtener el código html que se usa en el proceso.

**Directivas interactivas**

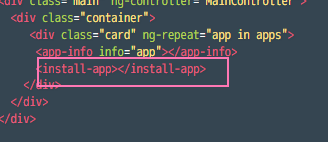
****

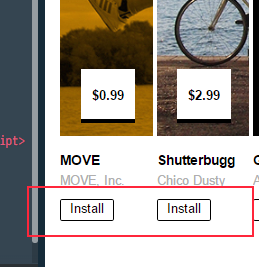
El argumento link me define una función para hacer de esta una directiva interactiva que se encarga de una función.

HTML DEL PROCESO



Inclusión en el documento principal:



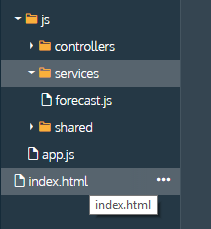


Resultado del proceso 🡪 Boton interactivo de propósito especifico.

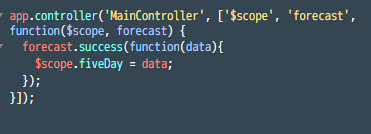
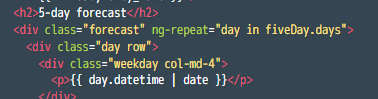
**Servicios web**

Angular necesita capturar datos de un servidor para esto angular necesita objetos en formato JSON.

Debemos tener un directorio donde guardamos los services y las conexiones a servidores externos

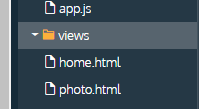


La relación entre clases se muestra de la siguiente forma :

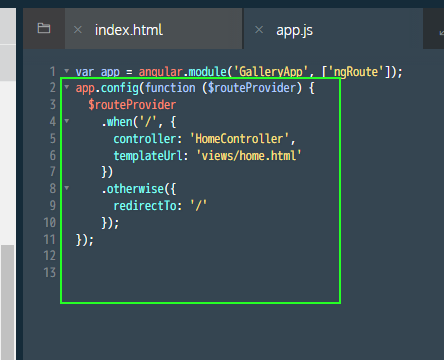
1. Tenemos un servicio que se encarga de pedir unos datos al servidor por medio de get
2. 
3. Para poder utilizar estos datos y convertirlos en funciones que nos sirvan para la operación entre datos , debemos hacerlo en el controlador de la siguiente manera :
4. 
5. Como podemos notar hay cierta relación con jquery respecto al hecho de que usamos un método .success y cuando este es exitoso podemos guardarlo en una variable.
6. La forma de acceder al objeto desde las expresiones y directivas es la siguiente
7. 
8. Como se puede notar la forma en que hacemos el foreach es flexible mientras se manden los datos adecuados y pertinentes.

**Vistas y Rutas**

Angular posee su propio sistema de rutas y adicional su sistema de vistas (‘tal cual como laravel vean pues ‘) debería crearse una carpeta de este modo:



En el módulo del programa agregamos la ruta



Nótese que para utilizarlo del mismo modo instanciamos la vista a la que irá y el controlador que utilizará

Usando el routeProvider

De esta forma definimos las rutas.

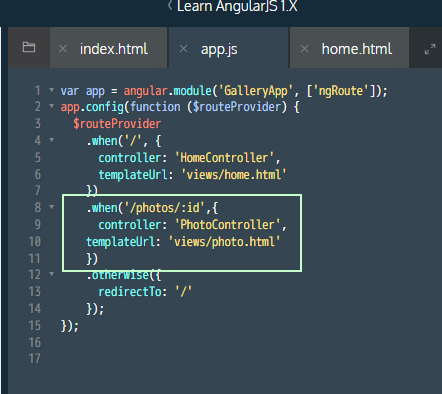
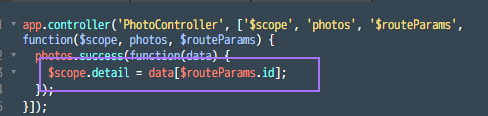
Para que esto nos funcione e obligatorio que en el index.html exista una directiva que se encargara de recargar una sección del documento html con solicitudes Ajax implícitas a través de las rutas:



Dentro del body hay un div.header en este agregamos  **ng-view** y con esta directiva ya podemos sobreescribir el html a través de las rutas.

**Como mandar datos entre rutas:**

Aca tomamos un ejemplo

1. Teniendo las rutas definidas
2. 
3. Vemos que cuando el navegador tenga un /potos/(variables) , me mandara a views/photo con el PhotoController.
4. Oh sorpresa , es muy similar como manda los datos dela url para mandar una variable por get
5. 
6. Dentro del controlador con la variable que capturamos por la ruta usamos el servicio photos con la instancia $routeParams.id. , gracias a la función $scope.detail para tomar un dato en especifico de un arreglo de datos.
7. 
8. Ya dentro del html importado .
9. 
10. Mostramos los datos de esa imagen en especifico.