

# *InstaLui*

Luisa Vidal  
DAW2A DIW-E

## ÍNDICE

Instalación ionic .....	2
Usando ionic .....	2
Accediendo a las listas .....	3
Visualizar contenido.....	3
Una vez tenemos establecida la conexión con nuestro servicio REST hemos de crear los controladores necesarios para ir mostrando la información obtenida.....	3
Los HTML .....	3
DINAMIZING YOUR CONTENT .....	4
Aplicación móvil .....	4
Splash & icon .....	4
Emular Android .....	5

## Instalación ionic

---

Seguimos los pasos de instalación que vienen en la página de ionic <http://ionicframework.com/getting-started/>.

Es necesario instalar node.js, bower, yeoman y grunt.

Cuando tenemos todo instalado correctamente creamos una carpeta donde almacenar nuestro proyecto y una vez dentro por línea de comandos creamos el proyecto con alguna de las plantillas que nos proporciona ionic. En este caso **ionic start myApp sidemenu**

Ahora solo basta iniciar el proyecto y empezar a trabajar con él. Siempre es necesario iniciar desde línea de comandos, escribiendo `ionic serve`, desde la carpeta donde lo creamos.

## Usando ionic

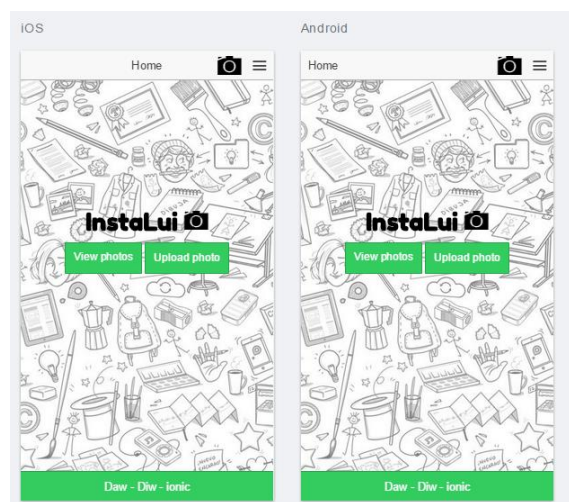
---

Para empezar a modificar el proyecto editaremos los templates que vienen por defecto que se encuentran dentro del directorio `www/templates` dentro del proyecto y los `.js` donde añadiré los controladores necesarios para ir mostrando la información y las llamadas necesarias a dichos controladores. En mi caso solo he modificado 5 páginas y he añadido un home.

He usado la página <http://ionicframework.com/docs/components/> para agregar alguno de los componentes que proporciona ionic y <http://ionicons.com/> para agregar el icono de la cámara como logo del proyecto.

En el `index.html` solo he añadido las llamadas a los scripts que utilizará la página que están detallados más adelante.

Lo primero que he hecho ha sido crearme un home atractivo y simple que consta del título y dos botones, de los cuales solo uso el `view categories` que es el que enlazará con la página donde se visualizará la primera lista que debemos mostrar. He añadido también un icono en el header que redirige al home. Una vez hecho el html lo añado a la carpeta `templates` y dentro del archivo `app.js` realiza la llamada con `en el scope`.



## Accediendo a las listas

---

Podemos acceder a la lista de categorías desde el botón del home o desde el menú desplegable que nos aparece arriba a la derecha.

En la página nos aparecerá la lista con los nombres almacenados en el REST al cual accedemos mediante los **Factory** que tenemos declarados en **categories.js**. En total hay declarados 3, uno para categorías, otro para la lista de ítems y el último para imágenes. Cada uno de ellos realiza una consulta a la base de datos REST pasándole, en el caso de los ítems e imágenes, el id necesario.

Para poder acceder al REST es necesario enlazar la base de datos con el cliente, por ello uso **config.js** que es el que se encarga de establecer la conexión.

## Visualizar contenido

---

Una vez tenemos establecida la conexión con nuestro servicio REST hemos de crear los controladores necesarios para ir mostrando la información obtenida.

En el **app.js** cargamos el resto de archivos .js dentro del module.

Dentro de **controllers.js** añadiremos los controladores necesarios.

- Para categorías, el más sencillo de los tres, basta con acceder a la base de datos con el Factory correspondiente y guardamos todo en una variable de angular. Para visualizar el contenido deberemos añadir en **app.js** (que es donde se cargan todos los “controladores”) el scope indicándole el template en que vamos a mostrar el contenido obtenido y llamando a su controlador correspondiente que deberá tener el mismo nombre que hemos indicando en el **controlles.js**.
- Para la lista de ítems realizamos el mismo paso anterior pero en este caso solamente almaceno en un array los id que coinciden con la id de categorías. Añado el scope correspondiente pero esta vez pasándole el id para que pueda compararlo en el controlador.
- Con el controlador imágenes es lo mismo que en el controlador de ítems pero esta vez comparando el id del ítem con el id de la imagen. Agrego también un scope en **app.js**

Me ha sido necesario descargarme el archivo **angular-resources.js** para realizar la conexión con el servicio REST.

### Los HTML

Las páginas que uso para mostrar cada contenido son, **playlists.html**, **playlist.html** e **ítem.html**. Las dos primeras vienen de base con el proyecto de ionic y la tercera es donde se muestra la imagen detallado aplicando un complemento **list-card** de la web oficial de ionic.

Modifício los **ng-repeat** por lo que quiero que me muestre la página indicándole en el Factory que se encuentra y luego accediendo al atributo que deseo mostrar. Este proceso es lo mismo para los otros dos simplemente modificando el Factory al que tiene que acceder y el id correspondiente para que filtre el contenido.

## DINAMIZING YOUR CONTENT

---

- Instalamos la extensión POSTMAN en Chrome.
- Una vez instalado nos conectamos a <http://www.planificaciondeportiva.es/bmoll-app/api/web/v1/>
- Importamos el contenido del ficho facilitado por el profesor donde se encuentran todos los objetos JSON

## Aplicación móvil

---

Para que nuestro proyecto funcione en nuestro dispositivo móvil es necesario instalarnos CORS y modificar las cabeceras para introducir contenido en el servicio REST.

### Splash & icon

He creado el splash y el icono que se visualizará cuando el proyecto cargue en un dispositivo móvil (no funciona en escritorio).

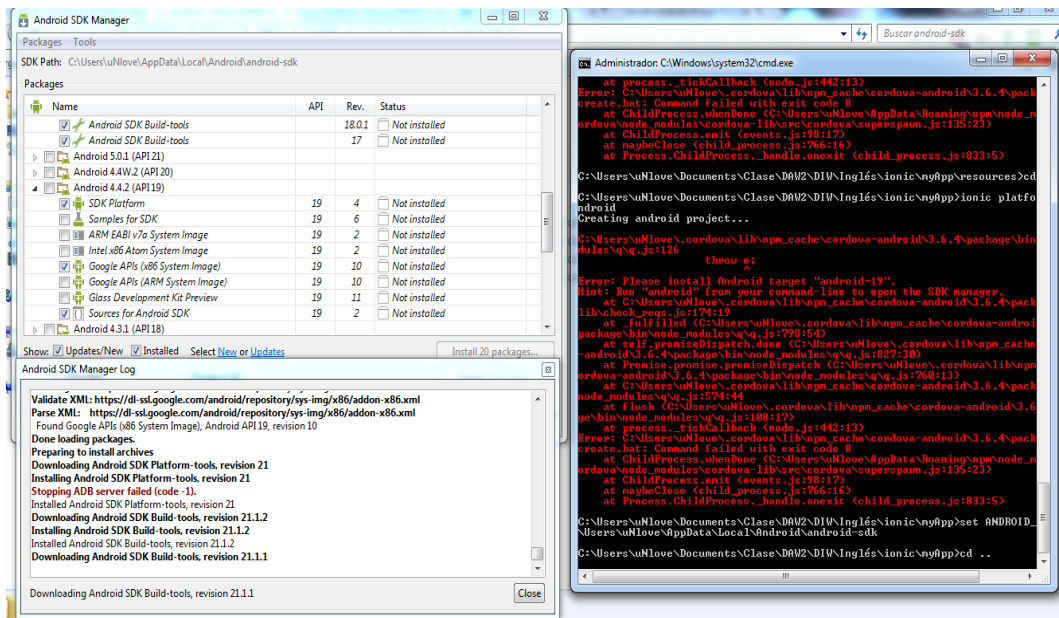
Para “activarlos” es necesario instalarse el plugin y crearse una plataforma de android (en mi caso) para que reconozca donde tenemos instalado el sdk y el home de las herramientas. Buscando un poco por google encontramos estos comandos:

- `set ANDROID_HOME=C:\\Users\\uNlove\\AppData\\Local\\Android\\android-sdk-windows`
- `set PATH=%PATH%;%ANDROID_HOME%\\tools;%ANDROID_HOME%\\platform-tools`

Si todo ha ido bien ahora ya podemos iniciar la plataforma con el comando **ionic platform add android**

Y con esto ya podemos crear el splash y nuestro icono.

- **ionic resource --splash**
- **ionic resource --icon**



Con eso nos aparecerá la carpeta resource dentro del proyecto y pegamos allí los dos archivos (tienen que estar en formato .png y .psd para que los reconozca)

## Emular Android

Para mostrar la aplicación desde el dispositivo móvil sin necesidad de instalar nada en el nuestro, podemos descargarnos Blue Stacks que sirve para emular Android y es muy sencillo de instalar. Lo único será necesario instalarnos de nuevo el sdk. Una vez instalado el sdk introducimos el comando **cordova build android** y ya tendremos nuestra aplicación en el emulador.

