

The Voyeur App

Miguel Angel Senén Aroca
2º DAW

Índice

Presentación y propósito del proyecto.....	3
Requisitos.....	3
Preparando el entorno de desarrollo.....	3
Inicio de un proyecto nuevo.....	6
Añadir una plataforma nueva a nuestro proyecto.....	6
Ver los resultados de nuestros avances en el navegador.....	6
Compilando nuestro proyecto.....	6
Añadir un SplashScreen y un icono a nuestra aplicación.....	6
Casos de Uso.....	7
Enfoque técnico y descripción.....	8
Estructura del proyecto.....	8
Comentarios de código enfocados a las acciones del usuario.....	10
Home.....	10
Menú.....	10
Camera.....	10
Categories.....	10
Items.....	11
Images.....	11

Presentación y propósito del proyecto

Crearemos un proyecto de aplicación para dispositivos móviles basados en la plataforma Android y usando Ionic Framework como tecnología. El objetivo de la aplicación es el de crear una red social en la que compartir imágenes relacionadas con una temática en concreto.

Requisitos

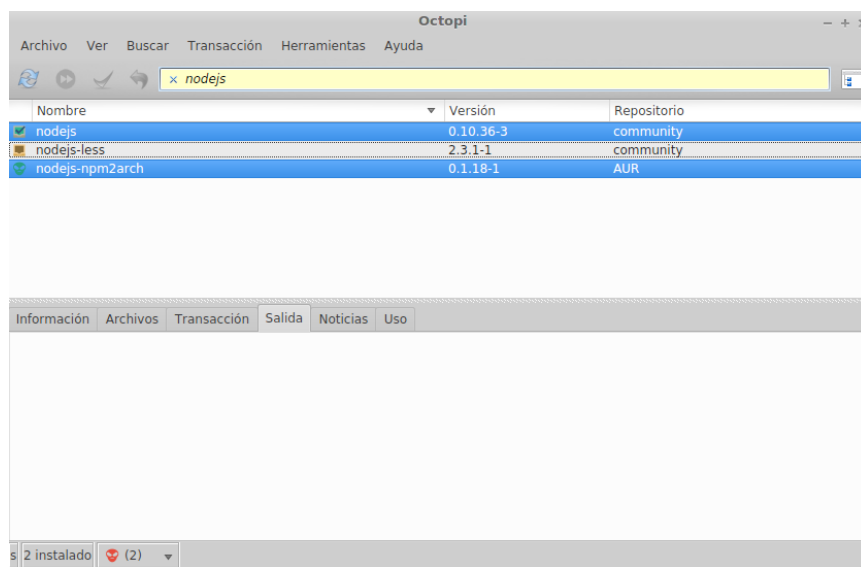
La aplicación ha de tener una pagina principal a modo de “home” en la que aparecerá una pantalla de carga (Splash Screen) con el logotipo de la aplicación y un acceso a un panel de navegación

Contará también con una pagina en la que aparecerán todas las categorías existentes, dentro de las cuales accederemos a una pagina en la que se mostraran los elementos que conforman dicha categoría. Si escogemos alguno de dichos elementos se nos mostrara en detalle el contenido de este.

Preparando el entorno de desarrollo

Lo primero que deberemos realizar es configurar nuestro sistema operativo, instalando en este los programas y las dependencias necesarias para el correcto desarrollo de la aplicación. Como el desarrollo de la aplicación se realizará bajo un sistema GNU/Linux, se usaran NPM y una herramienta gráfica para la instalación de software necesario con tal de permanecer neutral ante la variedad de sistemas de paqueteria e instalación de las diferentes distribuciones Linux existentes.

Primeramente instalaremos Node.JS y NPM. En sistemas basados en ArchLinux como el usado para el proyecto los buscaremos introduciendo en el cajón de búsqueda la entrada nodejs.



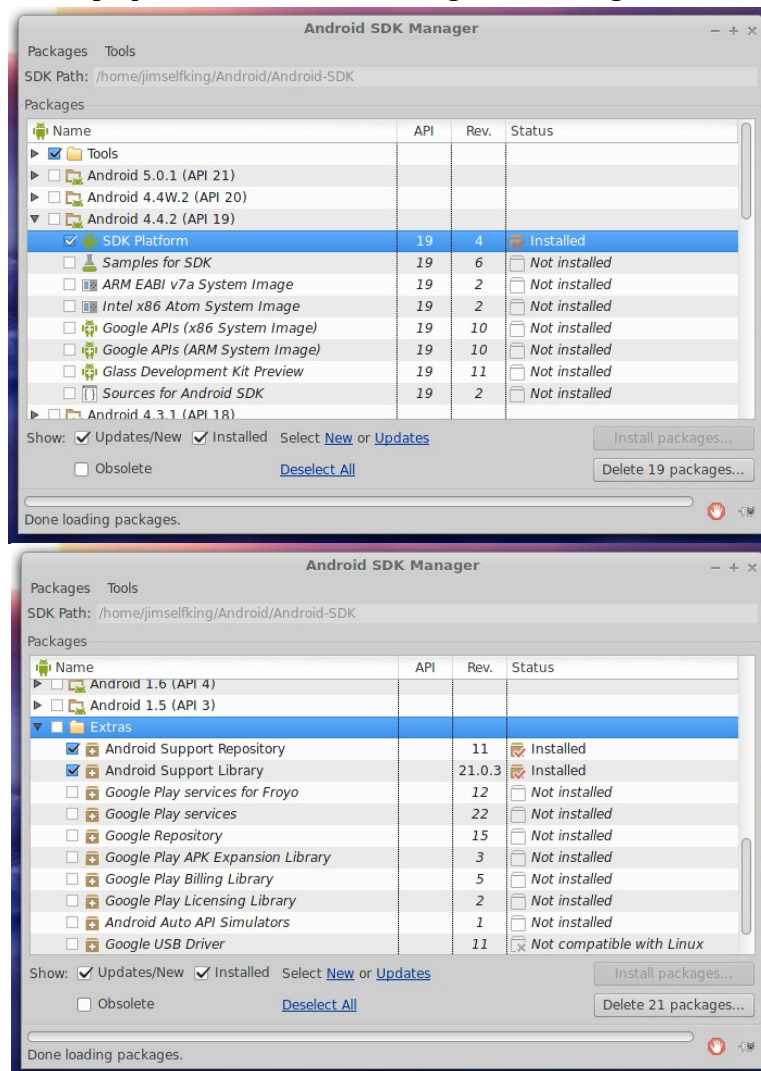
En sistemas Debian y derivados, seguiremos las instrucciones que nos proporciona el siguiente enlace [GitHub](#). También deberemos instalar Yeoman, Bower y Grunt usando npm:

```
npm install -g yo bower grunt-cli gulp
```

Debemos descargar también la última versión del SDK (Software Development Kit) de Android que podemos conseguir a través de [aquí](#). Lo descomprimiremos y emplazaremos en la carpeta Documentos de nuestro Home (el emplazamiento es opcional, pero mi recomendación es seguir estos pasos). Dentro de la carpeta que acabamos de descomprimir deberemos acceder a Tools y ejecutar en un terminal el archivo “Android”.

```
$ cd /home/$USER/Documentos/android-sdk_versionDescargada/tools/ && ./Android
```

La ventana que nos abrirá el comando anterior es el administrador del SDK de Android. En él deberemos descargar los paquetes marcados en las siguientes imágenes



A continuación instalaremos la ultima versión del JDK que tengamos en los repositorios. El siguiente paso es configurar nuestro bashrc para que cargue al inicio el path al SDK de linux. Para ello introducimos en una terminal el siguiente comando:

```
gedit /home/$USER/.bashrc
```

Sustituiremos gedit por el procesador de texto que prefiramos (pluma, kate, sublime-text. Etc.) e introduciremos al final de todo el documento las siguientes lineas:

```
export HOME="/home/$USER"
```

```
export ANDROID_HOME="$HOME/Documentos/android-sdk-linux"
```

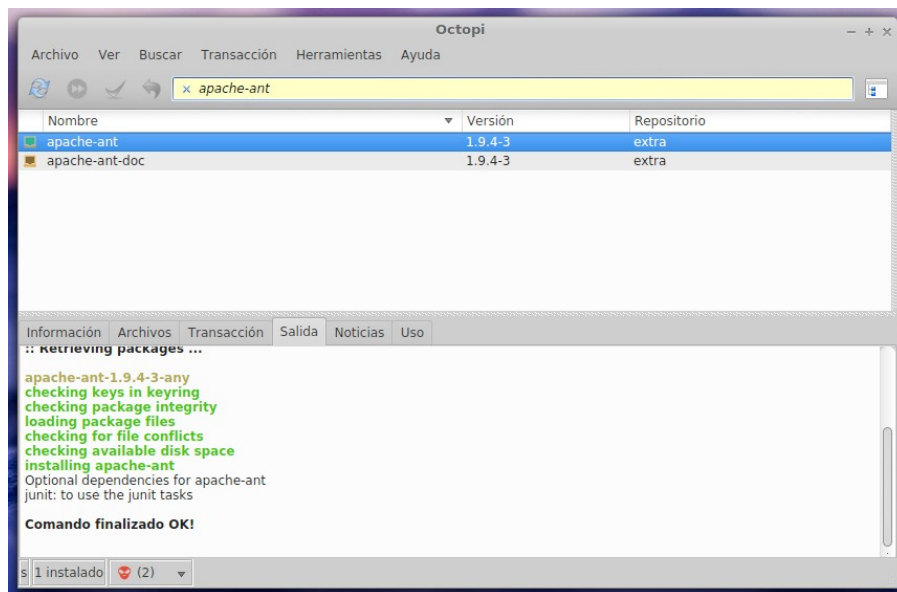
```
export PATH="$HOME/Documentos/android-sdk-linux/tools:$ANDROID_HOME/platform_tools:$PATH"
```

Estas lineas permitirán a ionic comunicarse con el sdk de android. Reiniciamos el sistema y cuando accedamos de nuevo, en una terminal escribiremos android, si nos abre el administrador del SDK abremos realizado correctamente este paso. El siguiente paso sera instalar cordova, ionic y gulp.

Para ello introducimos en la Terminal:

```
Sudo npm install -g cordova && sudo npm install -g gulp ionic
```

Para poder compilar el proyecto en un apk (paquete de instalación de android) deberemos instalar ademas apache ant:



Inicio de un proyecto nuevo

Desde terminal accedemos a la carpeta donde queramos que se cree el proyecto y tecleamos el siguiente comando

```
ionic start NombreDelProyecto blank
```

Existen dos plantillas mas de inicio, para usarlas deberemos sustituir en el comando anterior “blank” por “tabs” o “sidemenu”.

Añadir una plataforma nueva a nuestro proyecto

Desde terminal, y situados en la carpeta donde tengamos el proyecto, escribimos el siguiente comando:

```
ionic platform add android
```

Para poder trabajar sobre plataformas IOS, necesitaremos contar con un ordenador de marca Apple.

Ver los resultados de nuestros avances en el navegador

En la terminal, accedemos a la carpeta de nuestro proyecto e introducimos el siguiente comando:

```
ionic serve --lab
```

Compilando nuestro proyecto

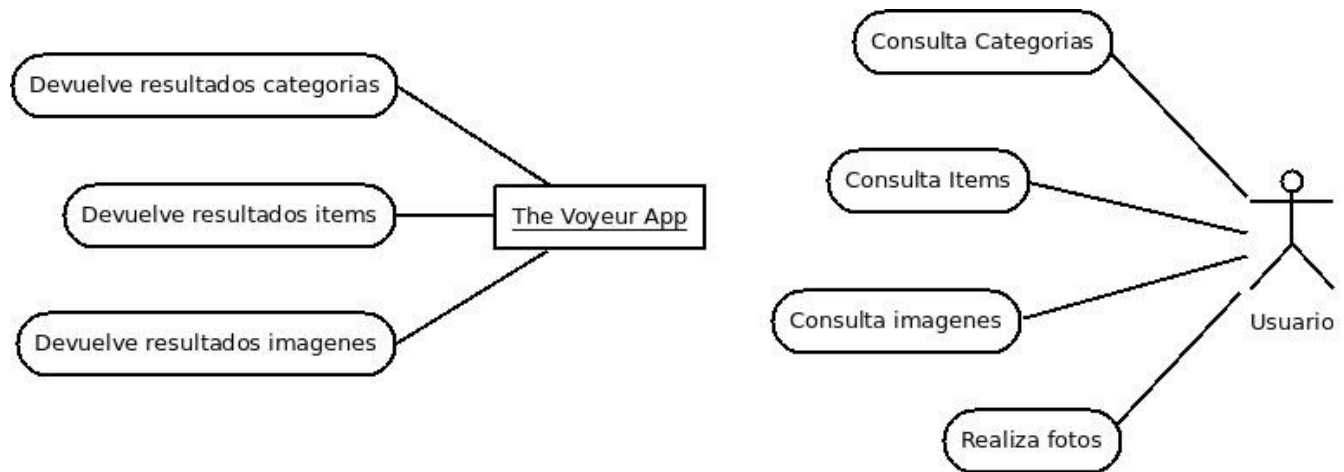
Para compilar el proyecto en un apk instalable en un terminal android, abriremos un terminal, nos situaremos en la carpeta de nuestro proyecto y teclearemos lo siguiente:

```
ionic build android
```

Añadir un SplashScreen y un icono a nuestra aplicación

Para añadir un splashScreen y un icono deberemos instalar en primer lugar el plugin de cordova para Splash: `cordova plugin add org.apache.cordova.splashscreen`. A continuación escribir en el terminal `ionic resources`. En un primer momento mostrara errores, puesto que no encuentra imágenes que tratar, pero nos sirve para que cree las carpetas necesarias. Emplazaríamos una imagen en formato png o psd en la carpeta resources con el nombre splash.png, y la otra con nombre icon.png. Solo queda repetir el comando anterior “ionic resources” para que cree los diferentes tamaños adaptados a diversos tamaños de pantalla. Nota: en la previsualización en el navegador no se podrá observar ningún cambio aparente, pero en el momento en el que la aplicación se ejecute en un dispositivo móvil, todos los cambios realizados estarán presentes.

Casos de Uso



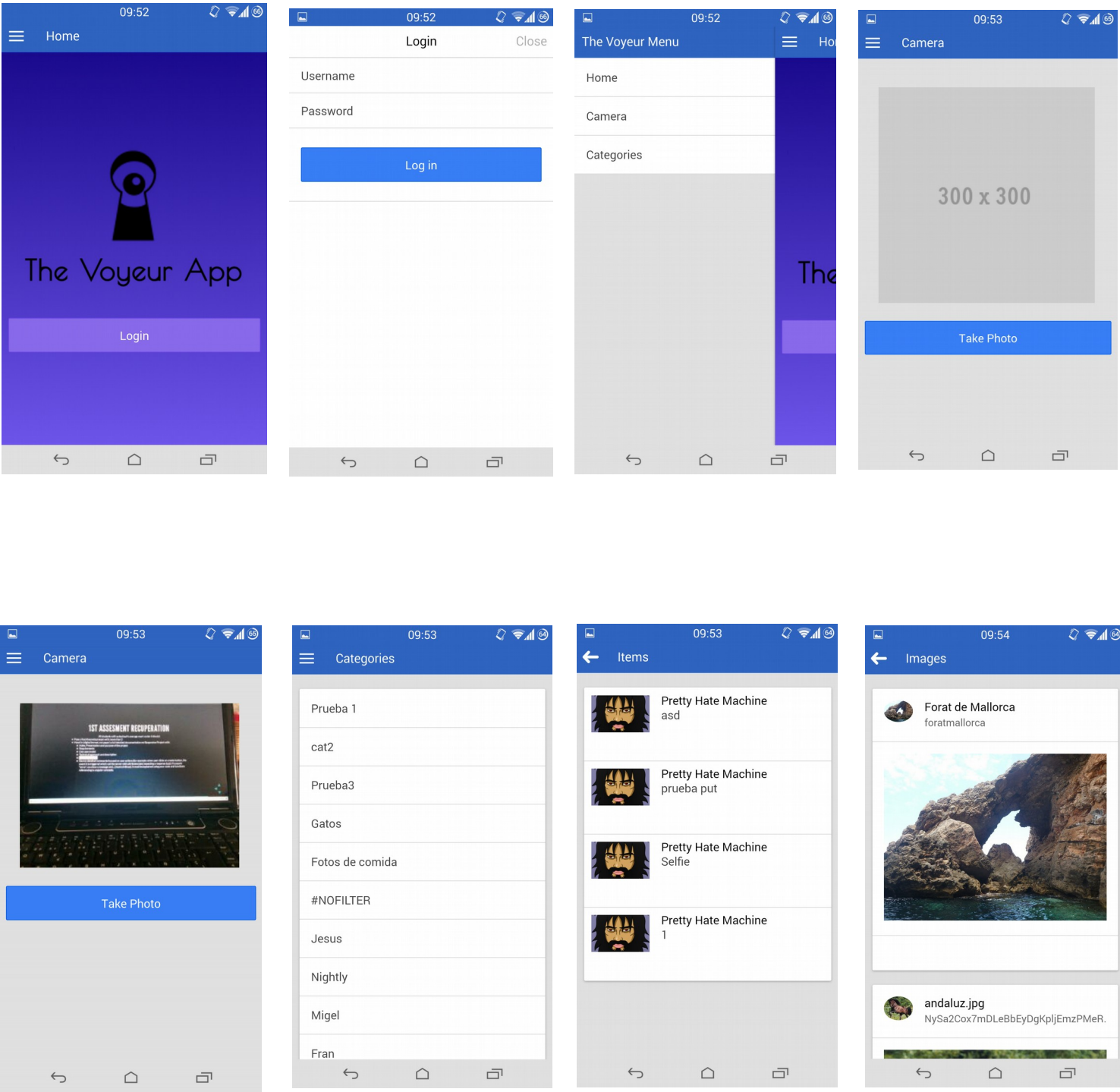
El usuario puede consultar las categorías existentes en la base de datos a través de la aplicación, que le devolverá los resultados mostrándoselos por pantalla. La misma situación se repite con las tablas de Items e imágenes. El usuario además podrá realizar fotografías a través de la aplicación.

Enfoque técnico y descripción

Todo el proyecto consta de 7 paginas escritas en lenguaje HTML que actuaran a modo de GUI para el usuario. En la pagina index.html inicializamos todos los ficheros javascript que serán los encargados de dar funcionalidad a la aplicación. Estos ficheros se encargaran de conectarse a la base de datos REST para extraer la información de ella y devolverla dentro de un array JSON para poder mostrarla mas adelante.

Estructura del proyecto

El proyecto inicia con el documento index.html que sera el encargado de inicializar todas las librerías necesarias para que funcione correctamente. En la parte visual se mostrara la pagina search.html que actúa a modo de HOME y en la que se puede ver de fondo el logotipo de la aplicación junto con un botón de login que muestra un dialogo modal de acceso, aparentemente útil pero sin función, puesto que no usamos ningún usuario ni contraseña para acceder a la base de datos. Para navegar por toda la aplicación utilizamos el documento menu.html, que contiene el panel deslizante lateral y la barra de navegación. En el panel lateral se nos muestran las tres secciones a las que podremos acceder, Home, que es la pagina en la que nos encontramos, Camera, una sección mediante la cual podremos acceder a la cámara de nuestro dispositivo y realizar fotografías, y por ultimo Categories, el área donde consultaremos el contenido de la base de datos mediante tres subsecciones, categories, items e images.



Comentarios de código enfocados a las acciones del usuario

Home

Cuando el usuario pulsa sobre el botón Login, la aplicación llama a la función login() a través de Scope. Esta función muestra el dialogo de inicio con una animación modal. Dentro de dicho dialogo hay un botón para cerrar que realiza la función inversa a la comentada anteriormente.

Si pulsamos en el icono de menú lateral, este se desplegara mostrando las secciones que componen la aplicación (ya se ha comentado con anterioridad que dicho menú se encuentra en la pagina menu.html).

Menú

Todas las opciones del menú se hayan instanciadas en app.js. Cuando seleccionamos una de estas, llamamos al state correspondiente a la pagina a la que queremos acceder, y en caso de que la pagina realice funciones adicionales, llamamos al controlador asignado.

Camera

En la sección de cámara se encuentran dos elementos, un cuadro vacío de 300x300 px en el que se situaran las capturas que realicemos con la cámara, y un botón que llama a la función takePicture() alojada en el controlador CallCam. Todos los controladores se encuentran situados en el mismo documento Controllers.js. La función takePicture accederá a la aplicación nativa del dispositivo, en caso que no tengamos asignada ninguna aplicación en concreto nos mostrara un dialogo para que seleccionemos la aplicación que deseamos utilizar. Una vez realicemos la fotografía, se emplazara en el cuadro nombrado anteriormente a través de ng-show.

Categories

Cuando el usuario selecciona la sección de categorías, automáticamente se activa el controlador asignado, que realizara una consulta al REST para obtener todas las categorías guardadas en la base de datos y las pasa a través de Scope en un array JSON a la pagina html, donde mediante ng-repeat situado en un ion-item y asignando el contenido del Scope a una variable, mostraremos uno a uno y en formato lista todo el contenido de la tabla Categories. Para acceder a dicha tabla previamente debemos configurar el acceso a la base de datos en config.js, asignándole a una variable CATEGORIES los datos para acceder al REST. Estos datos serán usados posteriormente en el Factory de categories, donde se realizara la consulta por id a la base de datos que posteriormente el controlador utilizara.

Cuando realizamos el ng-repeat en el contenido del Scope, también asignamos un enlace a la

pagina Items a la cual le pasaremos el numero de identificación de la categoría para poder observar los items hallados en ella.

Items

Items recibe en el State el id de la categoría que hemos seleccionado anteriormente y lo utiliza para llamar al controlador y a la pagina donde se mostraran los resultados. El controlador recibirá el ID de categoría y realizara una consulta de todos los elementos que se encuentran en la tabla Items. Como ocurría en el caso anterior, el acceso al REST se ha de realizar en config.js para posteriormente crear el factory de items y que el controlador realice la consulta adecuadamente. Una vez obtiene todos los elementos que conforman la tabla, los filtra mediante un bucle for para guardar en Scope solo aquellos que tengan un category_id igual al ID de la categoría seleccionada anteriormente.

En la pagina html (playlist.html), al igual que sucedía con categorías, realizaremos un ng-repeat por cada ion-item en el cual guardaremos el contenido del Scope en una variable items para mostrar el contenido seleccionado. En este caso la lista se muestra con un pequeño icono referenciando al tema tratado. Esto es posible gracias al uso de dos contenedores de clase card y list, cuyo código se puede obtener fácilmente en la web de ionic, en el apartado CSS/Cards. Como anteriormente se realizó con categorías, en esta ocasión cada uno de los elementos que se muestran enlazan con el State de imágenes enviándoles el id del Item seleccionado usando este código: `"href="#/app/images/{{items.id}}"`.

Images

Al igual que ocurrió con Items, Images recibirá en el State el valor del numero identificativo del Item que hayamos seleccionado. El State nos enlazará con la página asignada y activara el controlador, que resulta ser una copia del controlador de Items, unicamente varia la tabla a la que realiza la consulta, siendo esta la tabla de images. Al igual que los otros dos casos, es necesario declarar la conexión al REST en config.js y crear un factory que realice la consulta pertinente.

En la parte visual, y como en los casos anteriores, se mostrará el contenido filtrado y enviado por Scope desde el controlador, asignándoselo a una variable que mostrara por pantalla en forma de tarjetas divididas en tres secciones, un header donde se muestra el nombre de la imagen, la etiqueta asignada y un pequeño snapshot. El cuerpo de la tarjeta, donde se vislumbrara la imagen alojada en la base de datos, y un footer en el que se podrá leer la descripción de la imagen.