

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



E-mail

Contraseña

Entrar

Registrarme
Olvidé mi contraseña

Inicio

Quiénes somos

Formación

Comparador de salarios

Nuestros libros

Más



» Estás en: Inicio » Tutoriales » Hooks en Cordova:Desarrollo de aplicaciones móviles multiplataforma con Apa...



Rubén Aguilera Díaz-Heredero

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE



[Ver todos los tutoriales del autor](#)

Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2014-11-17

Tutorial visitado 10 veces [Descargar en PDF](#)

Desarrollo de aplicaciones móviles multiplataforma con Apache Cordova utilizando AngularJS, Ionic y ngCordova

0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Instalación y configuración del entorno de desarrollo
- 4. Creación y configuración del proyecto
- 5. Descripción de las tareas que podemos utilizar
- 6. Workflow altamente productivo
- 7. Incluimos la librería ngCordova
- 8. Caso de uso: crear una opción de menú que muestre la información del dispositivo
- 9. Conclusiones

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Mac Book Pro 15" (2,3 Ghz Intel Core i7, 16 GB DDR3)
- Sistema Operativo: Mac OS X Mavericks
- NodeJS 0.10.33
- Yeoman 1.3.3
- Grunt 0.4.5
- Bower 1.3.12
- AngularJS 1.2.26
- Ionic 1.0.0-beta.13
- Cordova 4.0.1
- ngCordova 0.1.7-alpha

2. Introducción

En este tutorial vamos a intentar describir con el mayor detalle posible la configuración para el mejor entorno de desarrollo que he podido encontrar para trabajar en este tipo de aplicaciones y vamos a ponerlo a prueba con un pequeño caso de uso.

3. Instalación y configuración del entorno de desarrollo

Síguenos a través de:



Últimas Noticias

» [Curso JBoss de Red Hat](#)

» [Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris](#)

» [Portales, gestores de contenidos documentales y desarrollos a medida](#)

» [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)

» [Screencasts de programación narrados en Español](#)

[Histórico de noticias](#)

Últimos Tutoriales

» [Paradigma publish/subscribe con Spring Data Redis](#)

» [Creación paso a paso de un webscript Alfresco](#)

» [Integración de MonkeyTalk](#)

En esta sección vamos a ver cómo instalar y configurar las herramientas necesarias para estos desarrollos en el entorno de Mac, lo bueno que tienen estas herramientas es que son multiplataforma por lo que estos pasos no difieren mucho en otras plataformas como Ubuntu.

NodeJS

A poco que hayáis metido la cabeza en el desarrollo frontend ya os habréis encontrado con esta herramienta que permite ejecutar Javascript en el servidor. Gracias a ella se han implementado multitud de herramientas como: la línea de comandos Cordova e Ionic, Yeoman, Grunt, Bower y un sin fin de aplicaciones.

Para instalarlo simplemente accedemos al web <http://nodejs.org/download/> y descargar el instalable adecuado.

En el mismo proceso de instalación también se instala la herramienta NPM (Node Package Manager) que vamos a utilizar para instalar el resto de herramientas.

Git

Para el que no lo conozca es un sistema de control de versiones distribuido. Necesitamos tener un cliente de Git instalado, ya que muchas de las herramientas que vamos a instalar descargan código que necesitan desde estos repositorios.

Para instalarlo, accedemos a la web <http://git-scm.com/downloads> y seleccionamos el adecuado a nuestra plataforma.

Ruby

Solo es necesario en caso de que queramos utilizar SASS para trabajar con CSS. Si usáis Mac seguramente ya lo tengáis instalado pero si no es el caso, recomiendo seguir cualquiera de los métodos que se mencionan en esta URL: <https://www.ruby-lang.org/en/downloads/>

Ant

El viejo Ant es utilizado por algunas de las otras herramientas para poder realizar su funcionalidad. Si no lo tenéis instalado ya, simplemente descargarlo de la [web](#).

Lo siguiente es descomprimirlo y actualizar la variable ANT_HOME y PATH.

Resto de herramientas

En este momento ya tenemos instaladas las herramientas mínimas necesarias para poder instalar el resto, que gracias a la gestión de módulos de NPM basta que instalemos una para que automáticamente se instalen el resto.

Vamos a instalar el generador de Yeoman "generator-ionic". Me gusta porque te integra perfectamente AngularJS e Ionic, y por defecto, te viene preparado para poder utilizar cualquier plugin de Cordova. Además te ofrece la posibilidad de hacer livereload en el propio dispositivo móvil aumentando nuestra productividad al no tener que estar constantemente teniendo que volver a instalar la aplicación para ver cómo queda.

Para instalar este generador simplemente tenemos que ejecutar en un terminal:

```
view plain print ?
01. npm install -g generator-ionic
```

Con la ejecución de este comando, automáticamente, npm nos ha instalado todos los módulos de los que dependen, entre los que destacan:

- **Yeoman:** es la herramienta que nos va a permitir crear la aplicación con un determinado esqueleto.
- **Bower:** es la herramienta que gestiona de manera eficaz las dependencias entre librerías Javascript de nuestro proyecto. Si queremos incluir una nueva lo haremos con el comando "bower install nombre_librería" y se encargará de almacenarla junto a las otras y añadir la referencia en el fichero index.html por nosotros.
- **Grunt:** es una herramienta de task runner, es decir al estilo de Ant, mediando la ejecución de comandos podemos realizar distintas tareas como: minificar el código, actualizar las librerías, ejecutar el proyecto en un navegador, ...
- **Cordova:** es la plataforma de ejecución de aplicaciones híbridas que nos permite mediante un único desarrollo web poder desplegar nuestra aplicación en distintas plataformas como: android, ios, windows phone, etc...
- **Ionic:** tanto la librería Javascript como la herramienta por línea de comandos que entre otras cosas nos permite hacer el livereload en el dispositivo móvil. La librería nos ofrece un catálogo de directivas de AngularJS y estilos que tienen por objetivo que apariencia de la interfaz sea lo más "nativa" posible para mejorar la experiencia de usuario.

Nota: queda a responsabilidad de cada uno tener preparado el entorno para poder desarrollar en la plataforma móvil deseada. Es decir, para Android tener el Android SDK configurado e instalado, para IOS tener el XCode con todas las herramientas necesarias, etc... Además para trabajar con Android aconsejo echar un vistazo a [Genymotion](#) es mucho más rápido que el emulador convencional de Android.

4. Creación y configuración del proyecto

Con todas las herramientas instaladas y correctamente configuradas vamos a comenzar el desarrollo de nuestra aplicación web con AngularJS, Ionic y ngCordova sobre Apache Cordova.

Una vez instalado creamos la carpeta raíz del proyecto donde queramos almacenar la aplicación y dentro ejecutamos el asistente de Yeoman.

```
view plain print ?
01. mkdir poc-autentia
02. cd poc-autentia
03. yo ionic
```

De este modo ejecutamos el asistente que nos irá haciendo una serie de preguntas:

- **Would you like to use SASS with Compass (requires Ruby)?** Nos pregunta si queremos gestionar las hojas de estilo con la herramienta SASS, que nos es más que un preprocesador de las hojas estilo CSS que nos permite tener un código más legible y reducido. Es necesario tener instalado Ruby y la gema Compass. Si no hemos querido instalar Ruby, decid no.
- **Which Cordova plugins would you like to include?** Aquí podemos seleccionar si queremos incluir ya algún plugin del dispositivo.

- en iOS
- » [Soporte de Redis con Spring: RedisTemplate](#)
- » [Embeber vídeo en MailChimp](#)

Últimos Tutoriales del Autor

- » [Hooks en Cordova: Cargar todos los plugins de forma automática](#)
- » [Mockear la capa back con Dyson](#)
- » [Automatizando los Smoke Test con TestLink y Jenkins](#)
- » [Smoke Test implementados con TestNG y Selenium](#)
- » [GitLab: Crear y gestionar nuestro servidor propio de Git](#)

- **Which starter template [T] or example app [A] would you like to use?** Podemos seleccionar el esqueleto inicial de nuestra aplicación entre los siguientes:
 - **[T] Blank:** si no queremos ningún esqueleto.
 - **[T] Tabs:** nos proporciona un esqueleto con pestañas abajo.
 - **[T] Side Menu:** nos proporciona un esqueleto con el menú corredizo. Para el ejemplo vamos a utilizar este.
 - **[T] Maps:** nos proporciona un esqueleto con un mapa de Google.
 - **[T] User Intro:** nos proporciona un esqueleto con una mini-aplicación de introducción a base de slides antes de mostrar la pantalla principal.
 - **[A] Pets:** nos proporciona el esqueleto de una aplicación completa con pestañas abajo y algo más de funcionalidad.

Al finalizar Yeoman nos habrá creado la siguiente estructura de ficheros y carpetas, divididas por funcionalidad:

- Carpetas propias de Cordova:
 - **hooks:** almacena los scripts para realizar tareas personalizadas como la carga automática de plugins.
 - **plugins:** almacena el código fuente de los plugins que vayamos a utilizar.
 - **merges:** almacena por plataforma los ficheros que queramos personalizar.
 - **platforms:** almacena el código fuente de los proyectos híbridos por cada plataforma soportada.
 - **www:** almacena el código fuente que va a ser ejecutado dentro de cada proyecto híbrido. En el caso de este generador, es aquí donde va a almacenar la salida de las tareas que ejecute.
 - **config.xml:** es el fichero que almacena información de relevancia para poder configurar los proyectos dependiendo de la plataforma.
- Carpetas propias de la aplicación web:
 - **app:** almacena el código fuente de nuestra aplicación. Será aquí donde editemos los ficheros y recursos necesarios para la implementación de nuestra aplicación. Mantiene una estructura por defecto que es buena práctica respetar.
 - **test:** almacena el código de test. El proyecto ya está configurado para ejecutar los test con Karma y PhantomJS.
- Carpetas para la gestión del proyecto:
 - **node_modules:** contiene todos los módulos de nodejs necesarios para la ejecución de tareas del proyecto. Se regenerará a través del comando “npm install”.
 - **bower.json:** utilizado por la herramienta bower para gestión del proyecto. Almacena información como el nombre y la versión del proyecto así como todas las dependencias de librerías de Javascript. A través del comando “bower install” incluye estas dependencias en la carpeta “app/lib” de nuestro proyecto y edita el fichero index.html para incluir las referencias de script correspondientes.
 - **package.json:** utilizado por la herramienta npm. Especifica que módulos de node tienen que ser instalados para el correcto funcionamiento del proyecto. Estos módulos son los que se almacenan en la carpeta “node_modules”.
 - **Gruntfile.js:** es el fichero utilizado por la herramienta Grunt para el lanzamiento de tareas del proyecto, describir que tareas podemos realizar en el proyecto como: ejecutar la aplicación en un server, minificar el código, añadir una nueva plataforma o plugin, etc...
 - **.gitignore:** por defecto, el generador nos crea el fichero .gitignore donde se incluyen todas las carpetas y ficheros que no deben subirse al control de versiones, dado que pueden volver a ser generados de forma automática.

5. Descripción de las tareas que podemos utilizar

Estas son algunos comandos que tenemos disponibles cuando desarrollamos la aplicación web:

- **grunt serve =>** Ejecuta la aplicación en el navegador. Con la opción --livereload habilitamos que cualquier cambio en el código de la aplicación se vea inmediatamente reflejado en el navegador, sin tener que reiniciar.
- **grunt serve:compress =>** Lo mismo pero con los archivos fuente minificados.
- **grunt jshint =>** comprueba si estamos cumpliendo todas las normas de estilo en nuestro código.
- **grunt karma =>** deja corriendo el testrunner favoreciendo el desarrollo de nuestra aplicación con TDD, al tener un feedback permanente del estado de los tests.
- **grunt coverage =>** ofrece un informe de la cobertura de nuestros test usando Istanbul.

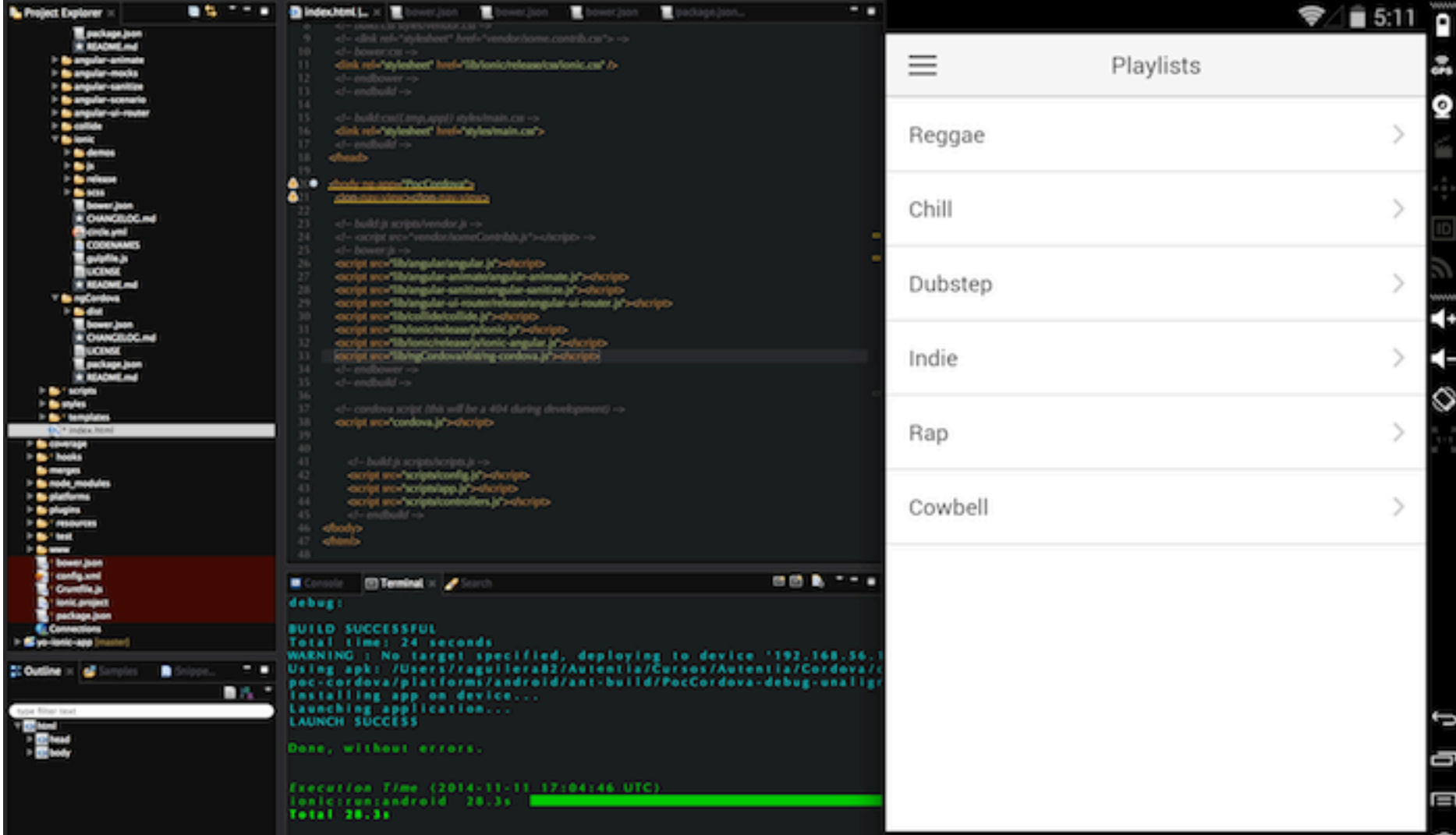
En el momento de necesitar probar el desarrollo en el dispositivo móvil, debido a que estemos utilizando funcionalidad nativa, tenemos los siguientes comandos:

- **grunt platform:add:{platform} =>** equivale a ejecutar el comando de Cordova “cordova platform add {platform}”. De esta forma añadimos todas las plataformas que queramos soportar. Ejemplo, “grunt platform:add:android”
- **grunt plugin:add:{plugin} =>** equivale al comando de Cordova “cordova plugins add {plugin}”. Ejemplo, “grunt plugin:add:org.apache.cordova.device”
- **grunt [emulate|run]:{target} =>** como el comandos de Cordova podemos decidir entre probar la aplicación en un dispositivo fijo o hacerlo en un emulador. También soporta el livereload y podemos hacer que los console.log se muestren en el terminal gracias al flag --consoleLogs. Ejemplo, “grunt run:android --livereload --consoleLogs”
- **grunt build:{target} =>** genera el ejecutable de la plataforma indicada. Ejemplo, “grunt build:android”. Acepta los flags --release o --debug para indicar el tipo de versión que se quiere obtener.

6. Workflow altamente productivo

Buscando la forma más productiva de desarrollar este tipo de aplicaciones lo que se puede hacer es tener la pantalla dividida: a la izquierda el editor que más nos guste (Sublime Text, Eclipse, ...) y a la derecha el emulador del dispositivo móvil.

Por si os ayuda yo tengo a la izquierda como editor Aptana que es un plugin de Eclipse y a la derecha tengo abierto un emulador de Genymotion. Como véis en la siguiente imagen:

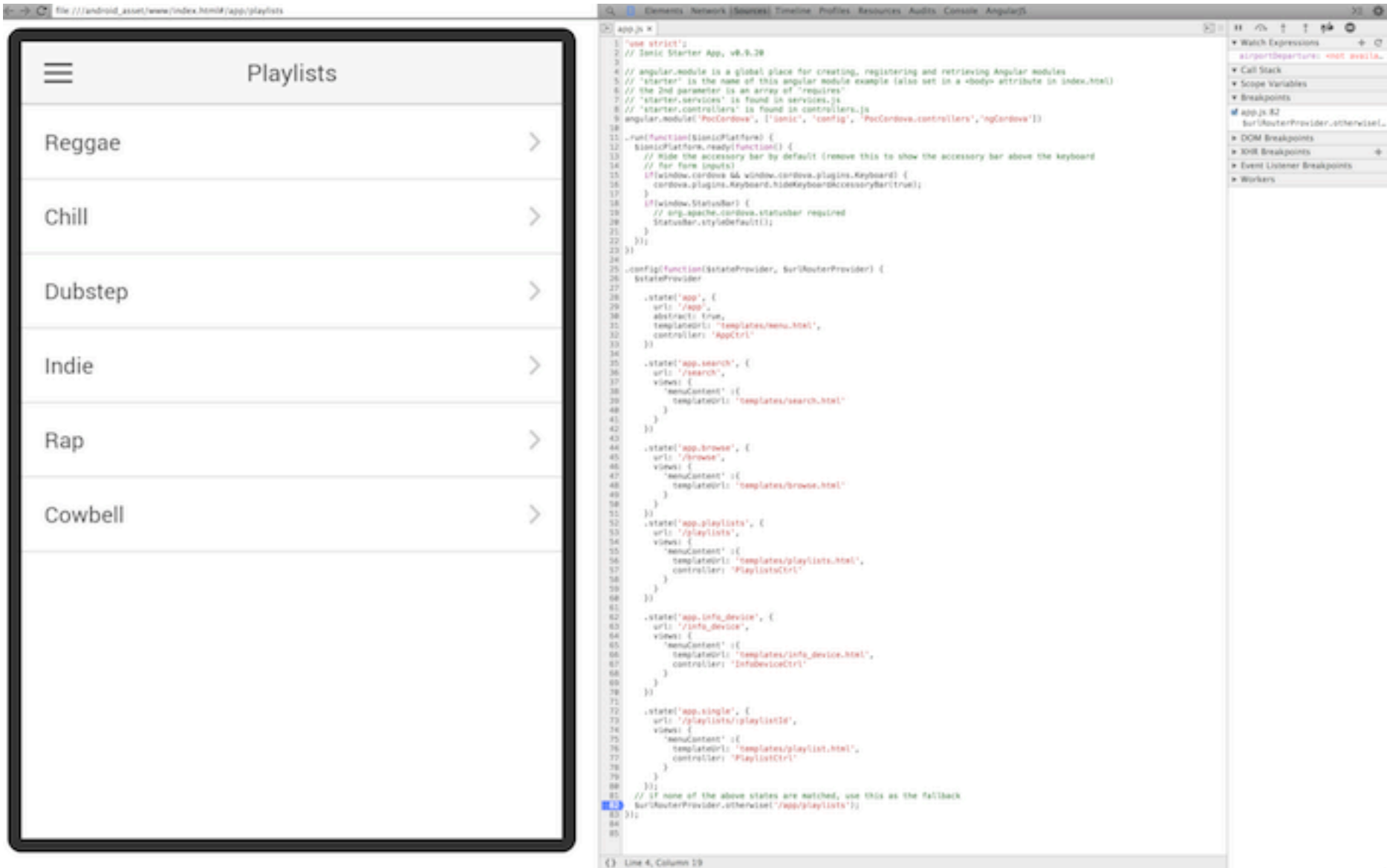


Ahora desde un terminal ejecutamos:

```
view plain print ?
01. grunt run:android --livereload
```

La aplicación se despliega en el emulador, utilizamos run porque los emuladores de Genymotion se tratan como si tuviéramos un dispositivo móvil enganchado al USB de nuestro ordenador, y con el flag "livereload" cualquier cambio en el código fuente de la aplicación inmediatamente lo vemos reflejado en el dispositivo.

Además en caso de no saber en algún momento por qué está fallando nuestra aplicación, si nuestro emulador tiene una versión superior a la 4.4, podemos utilizar el "Inspect Devices" de Chrome que nos permite depurar la aplicación: ver los fuentes, las variables, los scopes, poner puntos de ruptura, etc...



7. Incluimos la librería ngCordova

Trabajando con AngularJS contamos con la librería ngCordova que facilita el uso de una gran cantidad de plugins de Cordova registrándolos por nosotros como servicios de AngularJS, lo que facilita su utilización mediante la inyección de dependencias. Para poder hacer uso de esta librería tenemos que registrarla en nuestra aplicación con bower:

```
view plain print ?
01. bower install ngCordova --save
```

Ahora para que se incluya automáticamente la referencia a esta librería en nuestro index.html ejecutamos:

```
view plain print ?
01. grunt init
```

Por último la incluimos como dependencia en la declaración del módulo de nuestra aplicación que se encuentra en el fichero app.js.

```
view plain print ?
01. angular.module('PocCordova', ['ionic', 'config', 'PocCordova.controllers', 'ngCordova'])
```

Siguiendo estos pasos ya podemos hacer uso de los servicios que se describen en esta página: <http://ngcordova.com/docs>

8. Caso de uso: crear una opción de menú que muestre la información del dispositivo

Por ejemplo, si queremos hacer uso plugin “Device” el cual nos devuelve información relativa a las características del dispositivo, solo tenemos que registrar de la manera habitual el plugin de Cordova, en este caso:

```
view plain print ?
01. cordova plugin add org.apache.cordova.device
```

o

```
grunt plugin:add:org.apache.cordova.device
```

Para añadir una nueva opción al menú tenemos que editar el fichero menu.html añadiendo el siguiente item a la lista en la posición deseada:

```
view plain print ?
01. <ion-item nav-clear menu-close href="#/app/info_device">
02.     Info Device
03. </ion-item>
```

El siguiente paso será actualizar el router de nuestra aplicación. Para ello vamos a editar el fichero app.js y vamos a añadir un nuevo “state” al “routerProvider” que viene definido.

```
view plain print ?
01. .state('app.info_device', {
02.     url: '/info_device',
03.     views: {
04.         'menuContent' :{
05.             templateUrl: 'templates/info_device.html',
06.             controller: 'InfoDeviceCtrl'
07.         }
08.     }
09. })
```

Hemos establecido que la información se va a mostrar en la página “templates/info_device.html” por lo cual ahora vamos a crear este fichero con el siguiente contenido:

```
view plain print ?
01. <ion-view title="Info Device">
02.     <ion-nav-buttons side="left">
03.         <button menu-toggle="left" class="button button-icon icon ion-navicon"></button>
04.     </ion-nav-buttons>
05.     <ion-content class="has-header">
06.         <p>Cordova: {{cordova}}</p>
07.         <p>Model: {{model}}</p>
08.         <p>Platform: {{platform}}</p>
09.         <p>UUID: {{uuid}}</p>
10.        <p>Versión: {{version}}</p>
11.     </ion-content>
12. </ion-view>
```

Y los valores los vamos a recuperar y setear en el controlador “InfoDeviceCtrl” por lo que ahora en el fichero controller.js lo creamos, inyectándole el servicio de ngCordova que recupera la información del dispositivo (\$cordovaDevice) y el servicio de Ionic que nos permite saber si el dispositivo ya está preparado (\$ionicPlatform), el contenido quedaría de esta forma:

```
view plain print ?
01. .controller('InfoDeviceCtrl', function($scope, $cordovaDevice, $ionicPlatform) {
02. $ionicPlatform.ready(function(){
03.     $scope.device = $cordovaDevice.getDevice();
04.     $scope.cordova = $cordovaDevice.getCordova();
05.     $scope.model = $cordovaDevice.getModel();
06.     $scope.platform = $cordovaDevice.getPlatform();
07.     $scope.uuid = $cordovaDevice.getUUID();
08.     $scope.version = $cordovaDevice.getVersion();
09. });
```

Ahora ejecutamos la aplicación en un dispositivo físico ya que estamos haciendo uso de una funcionalidad nativa, por ejemplo si queremos probarlo en Android haríamos

```
view plain print ?
01. grunt platform:add:android
02. grunt run:android
```

Y en la opción de menú correspondiente tendremos que ver algo parecido a esto:

Cordova: 3.5.1

Model: Google Nexus 4 - 4.4.4 - API 19 - 768x1280

Platform: Android

UUID: ce188d6aee166d52

Versión: 4.4.4

9. Conclusiones

Como habéis podido comprobar la utilización del generator-ionic de Yeoman es un estupendo punto de partida para crear aplicaciones en la plataforma Apache Cordova con las tecnologías AngularJS e Ionic.

Cualquier duda o sugerencia en la zona de comentarios.

Saludos.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)



Por favor, vota +1 o compártelo si te pareció interesante

+

Share

|

f

t

in

m

s

g

+

0

g

+

0

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

PUSH THIS

Page Pushers

Community

Help?

no clicks

0 people

brought clicks to this page

powered by [karmacracy](#)