



**EBook Gratis**

# APRENDIZAJE ionic-framework

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#ionic-

framework

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Comenzando con el marco iónico.....</b>	<b>2</b>
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación o configuración.....	2
1. Instale Ionic Framework y Cordova (ya que las aplicaciones de Ionic se basan en Cordova.....	2
2. Iniciar un nuevo proyecto jónico:.....	3
3. Prueba la aplicación jónica:.....	4
Introducción e instalación y configuración de Ionic Framework.....	4
Ionic Framework Hello World App.....	6
Plataforma Iónica (Ionic Cloud) para Proyectos Iónicos de Yo (Yeoman).....	7
Plataforma jónica :.....	7
Construye, empuja, implementa y escala tus aplicaciones iónicas de la manera más fácil.....	7
Generador Ionic Framework.....	8
Un generador para el Ionic Framework de Yeoman, la herramienta de andamiaje de la web para.....	8
plataforma-iónica-cliente-web.....	9
Un cliente web que proporciona interacciones con la plataforma Ionic.....	9
Despliegue jónico.....	10
Inserte actualizaciones en tiempo real de sus aplicaciones de producción y administre el h.....	10
Analítica Iónica.....	11
Ver la transmisión en vivo de eventos o el número de eventos / usuarios en bruto / único a.....	11
Empuje ionic.....	12
Envíe notificaciones push dirigidas y automatizadas a sus usuarios.....	13
Aplicación de ejemplo.....	14
<b>Capítulo 2: ¿Cómo utilizar las características de EcmaScript 6 en Ionic?.....</b>	<b>16</b>
Examples.....	16
Usando gulp-babel y gulp-plumber.....	16
<b>Capítulo 3: ¿Cuál es la diferencia entre "acumulación iónica" y "preparación iónica"?.....</b>	<b>18</b>
Examples.....	18

constitución iónica vs preparación iónica.....	18
<b>Capítulo 4: Comenzar a crear aplicaciones en Ionic.....</b>	<b>19</b>
Examples.....	19
Comenzando una aplicación jónica.....	19
<b>Comenzando una aplicación jónica.....</b>	<b>19</b>
Plantilla de inicio nombrados.....	19
Entrantes Github Repo.....	19
Codepen URL starters.....	19
Iniciadores del directorio local:.....	19
Indicadores de línea de comando / opciones:.....	19
<b>Capítulo 5: Componentes jónicos de CSS.....</b>	<b>21</b>
Observaciones.....	21
Examples.....	21
Sintaxis básica del sistema grid.....	21
<b>Rejilla basica.....</b>	<b>21</b>
<b>Rejillas de compensación.....</b>	<b>22</b>
<b>Alinear columnas.....</b>	<b>22</b>
<b>Cuadrícula sensible.....</b>	<b>23</b>
Sintaxis básica del elemento de la lista.....	23
Uso básico de colores de utilidad.....	25
<b>Capítulo 6: Conectando Ionic con cualquier base de datos.....</b>	<b>27</b>
Examples.....	27
No puedes hacerlo directamente desde el framework Ionic.....	27
<b>Capítulo 7: Crear diálogo en jónico.....</b>	<b>28</b>
Parámetros.....	28
Observaciones.....	28
Examples.....	28
Crear diálogo en jónico.....	28
<b>Capítulo 8: Desplazamiento infinito iónico para mostrar elementos de carga a pedido (datos.....</b>	<b>29</b>
Examples.....	29
Carga n número de datos disponibles bajo demanda.....	29

<b>Capítulo 9: Desplegar Ionic como un sitio web</b>	<b>30</b>
Examples	30
Simplemente copie la carpeta www a su servidor web	30
<b>Capítulo 10: Dispositivo de cámara y acceso a la biblioteca de fotos desde la aplicación I</b>	<b>31</b>
Observaciones	31
Examples	31
Camara abierta y galeria de fotos	31
Para Ionic 3 Ejemplo	34
<b>Capítulo 11: Ejecute la aplicación Ionic en el emulador o en su teléfono</b>	<b>37</b>
Examples	37
Ejecute la aplicación Ionic en el emulador o en su teléfono	37
1. Añadir un objetivo de plataforma	37
2. Construye tu aplicación	37
Aplicación Live Reload durante el desarrollo (beta)	37
Indicadores / opciones de línea de comando para run y emulate	38
3. Emulando tu aplicación	38
4. Ejecutando tu aplicación	39
4.1. Especificando su objetivo	39
<b>Capítulo 12: Extensiones Ionic AngularJS</b>	<b>40</b>
Observaciones	40
Examples	40
Entradas de formulario	40
Ventanas modales	40
<b>Creando el objeto modal en el alcance</b>	<b>40</b>
<b>Controlar el modal</b>	<b>41</b>
<b>Eventos modales</b>	<b>41</b>
<b>Capítulo 13: Ganchos CLI iónicos</b>	<b>43</b>
Observaciones	43
<b>Introducción</b>	<b>43</b>
<b>Tipos de gancho</b>	<b>43</b>
<b>Formas de definir ganchos:</b>	<b>43</b>

Examples.....	44
Comprobando errores en sus archivos de Javascript en before_prepare usando jshint.....	44
<b>Capítulo 14: Ionic - Analice su aplicación con jshint y gulp-jshint como parte de su proce.....</b>	<b>46</b>
Observaciones.....	46
¿Qué es linting y cómo instalar los paquetes requeridos?.....	46
Examples.....	46
Añadir una tarea trilla.....	46
Crear archivo .jshintrc (opcional).....	47
Añadir Makefile.....	47
<b>Capítulo 15: Ionicones.....</b>	<b>49</b>
Observaciones.....	49
Examples.....	49
Uso básico.....	49
Uso extendido.....	49
<b>Capítulo 16: Nube Jónica para Proyectos Iónicos Yeoman.....</b>	<b>51</b>
Examples.....	51
Plataforma Iónica (Ionic Cloud) para Proyectos Iónicos de Yo (Yeoman).....	51
Plataforma jónica :.....	51
Construye, empuja, implementa y escala tus aplicaciones iónicas de la manera más fácil.....	51
Generador Ionic Framework.....	52
Un generador para el Ionic Framework de Yeoman, la herramienta de andamiaje de la web para.....	52
plataforma-iónica-cliente-web.....	53
Un cliente web que proporciona interacciones con la plataforma Ionic.....	53
Despliegue jónico.....	54
Inserte actualizaciones en tiempo real de sus aplicaciones de producción y administre el h.....	54
Analítica Iónica.....	55
Ver la transmisión en vivo de eventos o el número de eventos / usuarios en bruto / único a.....	55
Empuje ionic.....	56
Envíe notificaciones push dirigidas y automatizadas a sus usuarios.....	56
Aplicación de ejemplo.....	58
<b>Capítulo 17: Probando la aplicación jónica en un navegador.....</b>	<b>59</b>

Observaciones.....	59
Ejemplo.....	59
Examples.....	60
Pruebas en un navegador.....	60
LiveReload.....	60
Laboratorio Iónico.....	61
Especificando una dirección IP para usar.....	61
Proxies de servicio.....	61
Indicadores de línea de comando / opciones.....	62
<b>Capítulo 18: Publicando tu aplicación jónica.....</b>	<b>63</b>
Examples.....	63
Versión de lanzamiento del edificio desde macOS.....	63
<b>Capítulo 19: Servicios Backonic Iónicos (ionic.io).....</b>	<b>64</b>
Examples.....	64
Introducción y configuración.....	64
<b>Creditos.....</b>	<b>66</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ionic-framework](#)

It is an unofficial and free ionic-framework ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ionic-framework.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# Capítulo 1: Comenzando con el marco iónico

## Observaciones

Ionic es un marco para desarrollar aplicaciones móviles con HTML, CSS y JavaScript. Las aplicaciones iónicas se ejecutan como aplicaciones nativas y tienen una "apariciencia" nativa.

Ionic se basa en el marco [AngularJS](#) y proporciona una solución completa para diseñar, construir y empaquetar aplicaciones móviles. El diseño se logra con una colección de herramientas de plantilla y una [biblioteca de iconos personalizados](#) . Ionic proporciona [componentes CSS / SASS personalizados](#), así como también [extensiones de IU de Javascript](#) . Las aplicaciones iónicas se pueden construir, emular y empaquetar con su [interfaz de línea de comandos \(CLI\)](#) .

Las plantillas iónicas son dinámicas y receptivas, y se adaptan a su entorno para proporcionar una "apariciencia" nativa. Esta adaptación incluye diseño, estilo e iconos. Ionic también hace [que la personalización de la plataforma](#) independiente esté disponible. Debido a que las aplicaciones iónicas usan tecnología web frontal, también pueden verse en un navegador para un desarrollo más rápido.

Las aplicaciones iónicas se construyen sobre [Apache Cordova](#) de forma predeterminada. Tienen acceso a todos los [complementos de Cordova](#) que le permiten utilizar funciones nativas, como notificaciones automáticas, cámara, acelerómetro, etc. Las aplicaciones de Cordova funcionan en [múltiples plataformas](#) y dispositivos (teléfonos, tabletas, etc.) con muy poco esfuerzo adicional. Cordova se puede cambiar con [otras tecnologías multiplataforma](#) , como [trigger.io](#) .

## Versiones

Versión	Fecha de lanzamiento
1.3.1 "el salvador"	2016-05-12
1.3.0 "delhi"	2016-04-21
1.2.0 "zirconio-zeren"	2015-12-09
1.1.0 "xenón-xerus"	2015-08-13
1.0.0 "uranio-unicornio"	2015-05-12

## Examples

### Instalación o configuración

## 1. Instale Ionic Framework y Cordova (ya que las

## aplicaciones de Ionic se basan en Cordova) utilizando npm (el Administrador de paquetes de nodos):

Asegúrese de tener una versión actualizada de Node.js instalada en su sistema. Si no tiene instalado Node.js, puede instalarlo desde [aquí](#) .

Además, para los usuarios de Mac, tener la última versión de Xcode instalada en su sistema le brinda herramientas de línea de comandos y iOS Simulator, [descargue aquí](#) .

Abra una ventana de terminal (Mac) o una ventana de comando (Windows), e instale Cordova e Ionic:

```
$ npm install -g cordova ionic
```

En una Mac, puede que tenga que usar sudo dependiendo de la configuración de su sistema:

```
$ sudo npm install -g cordova ionic
```

Si ya tiene Cordova e Ionic instalados en su computadora, asegúrese de actualizar a la última versión:

```
$ npm update -g cordova ionic
```

o

```
$ sudo npm update -g cordova ionic
```

Siga las guías de plataforma de [Android](#) e [iOS](#) para instalar las dependencias de plataforma requeridas.

Nota: el desarrollo de iOS requiere Mac OS X. El simulador de iOS a través de la CLI de Ionic requiere el paquete ios-sim npm, que se puede instalar con el comando:

```
$ sudo npm -g install ios-sim
```

---

## 2. Iniciar un nuevo proyecto jónico:

Cree un proyecto de Ionic utilizando una de las plantillas de aplicaciones listas para usar, o una en blanco para comenzar de nuevo.

```
$ ionic start myApp blank
```

o

```
$ ionic start myApp tabs
```

```
$ ionic start myApp sidemenu
```

### 3. Prueba la aplicación jónica:

Para probar su aplicación Ionic en un navegador de escritorio en plataformas iOS y Android:

```
$ ionic serve --lab
```

Si bien el `ionic serve --lab` es excelente para probar la interfaz de usuario de la aplicación en múltiples plataformas, podría dar lugar a algunos problemas para la Consola de Javascript o la Inspección de elementos, en ese caso, lo que prefiera:

```
$ ionic serve
```

Para probar su aplicación jónica en un emulador:

```
$ cd myApp
$ ionic platform add ios android
$ ionic build ios
$ ionic emulate ios
```

Sustituye ios con Android para la prueba del emulador de Android:

```
$ ionic build android
$ ionic emulate android
```

Para probar su aplicación Ionic en un dispositivo Android conectado a través de USB:

```
$ ionic run android
```

Para probar su aplicación Ionic en un dispositivo iOS conectado a través de USB:

```
$ ionic run ios --device
```

## Introducción e instalación y configuración de Ionic Framework

### Marco jónico

Un marco de desarrollo de aplicaciones móviles multiplataforma que utiliza las tecnologías web Angular JS y Front End.

**Sitio web oficial** : <http://ionicframework.com/>

**Documentación** : <http://ionicframework.com/docs/>

## Instalación y configuración

**Instalación de** Ionic requiere NPM (Administrador de paquetes de nodos) y Cordova.

Puede descargar e instalar Npde JS desde [aquí](#) que viene con NPM fuera de la caja.

Para descargar Apache Cordova puedes usar NPM desde la línea de comandos

```
npm install -g cordova
```

Si ya tiene NPM y Cordova, puede instalar el marco iónico desde la línea de comandos utilizando el siguiente comando.

```
npm install -g ionic
```

Esto instalará y configurará el marco iónico para que lo uses desde la línea de comandos.

Nota \* Según el entorno de su sistema, es posible que deba ejecutar con privilegios de administrador.

## Comenzando un nuevo proyecto

Para iniciar un nuevo proyecto de Ionic Framework, puede utilizar el siguiente comando

```
ionic start myproject
```

O

```
ionic start myproject [template_name]
```

### Plantillas:

Ionic te permite crear proyectos usando algunas plantillas integradas

`tabs` (predeterminado): que creará una aplicación simple con vista de pestañas.

`sidemenu` : que creará una aplicación iónica con menú lateral.

`blank` : lo que creará una aplicación iónica en blanco.

que creará una nueva carpeta llamada `myproject` con todos los archivos de proyectos iónicos.

para probar el proyecto en su navegador puede usar el siguiente comando

```
ionic serve --lab
```

O

```
ionic serve
```

**Ejecutar un Emulate** Para ejecutar o probar la aplicación en el emulador o el teléfono primero, tendrá que agregar una plataforma para que pueda usar el siguiente comando

```
ionic platform [Platform Name]
ionic build [Platform Name]
ionic emulate [platform name]
```

Nombres de plataforma: puede mencionar directamente android e ios para las respectivas plataformas, puede mencionar varios nombres de plataforma también separados por espacios.

Para ejecutar tu aplicación puedes usar

```
ionic run [platform name]
```

Para ayuda puedes usar

```
ionic --help
```

o

```
ionic help
```

Consulte [este enlace](#) para una explicación detallada de cli iónico.

Consulte [este enlace](#) para los componentes CSS disponibles en ionic.

Consulte [este enlace](#) para la referencia de la API de Javascript para ionic.

Para un desarrollo más rápido con ionic, también puedes probar [ionic Playground](#) .

Mucha suerte con el marco iónico ...

## Ionic Framework Hello World App

Después de toda la configuración, para hacer la aplicación Hello World.

- Para crear la aplicación Simple Blank, ejecute el siguiente comando en el terminal:

```
ionic start HelloWorld blank // create new project

cd HelloWorld // get into HelloWorld directory
```

- Abra el proyecto HelloWorld en el IDE de subline / webstrome:
  - Editar index.html, en www / ditictory

```
<body ng-app="starter">
  <ion-pane>
    <ion-header-bar class="bar-stable">
      <h1 class="title">Ionic Hello World App</h1>
    </ion-header-bar>
```

```
<ion-content>
  <div class="center">Hello World..!</div>
</ion-content>
</ion-pane>
</body>
```

- Para ejecutar en el navegador desde la terminal:

```
ionic serve // run the app in browser
```

- Para añadir plataforma:

```
ionic platform add android // add android platform
ionic platform add ios // add ios platform
```

- Para ejecutar en el dispositivo:

```
adb devices // to check devices is connected
ionic run android // to run on android devices
ionic run ios // to run on ios devices
```

- Para correr en livereload:

```
ionic run android -c -s -l // to check app in live reload with console.
```

## Plataforma Iónica (Ionic Cloud) para Proyectos Iónicos de Yo (Yeoman)



### Plataforma jónica :

## Construye, empuja, implementa y escala tus aplicaciones iónicas de la manera más fácil.

### Descripción del Título:

Ionic Platform es una plataforma en la nube para administrar y escalar aplicaciones móviles multiplataforma. Los servicios integrados le permiten a usted y a su equipo crear, implementar y hacer crecer sus aplicaciones de manera eficiente.

### Objetivo del documento:

Ionic Platform funciona bien con los proyectos estándar de Ionic. Pero los proyectos que siguen cualquier estructura de directorio no estándar pueden enfrentar algunos obstáculos. Estos documentos proporcionan los pasos para utilizar la Plataforma iónica en los proyectos iónicos creados con Yeoman.

## Alcance del documento:

Este documento cubre los pasos básicos para crear un proyecto Ionic utilizando Yeoman e integrarlo con Ionic Platform utilizando el Cliente Web de Ionic Platform. Este documento cubre los pasos básicos para utilizar Ionic Deploy, Ionic Analytics y Ionic Push.

## Público objetivo:

Los destinatarios de este documento son los desarrolladores de aplicaciones web / móviles, con experiencia tanto de nivel principiante como experto, que están familiarizados con los requisitos previos que se detallan a continuación.

## Requisitos previos:

Debe estar familiarizado con los siguientes marcos / herramientas antes de intentar este documento.

- AngularJs: <https://docs.angularjs.org/guide>
- IonicFramework: <http://ionicframework.com/docs/guide>
- Yeoman: <http://yeoman.io/codelab/index.html>
- Generador iónico: <https://github.com/diegonetto/generator-ionic>
- Plataforma iónica: <https://ionic.io/platform>

---

## Generador Ionic Framework



## Un generador para el Ionic Framework de Yeoman, la herramienta de andamiaje de la web para aplicaciones web modernas.

Node.js es un tiempo de ejecución de JavaScript creado en el motor de JavaScript V8 de Chrome. npm es el gestor de paquetes para JavaScript. Descargue e instale Node (y npm) desde <http://nodejs.org>

```
$ npm install npm -g
$ npm install -g yo
```

Yeoman lo ayuda a impulsar nuevos proyectos, prescribiendo las mejores prácticas y herramientas para ayudarlo a mantenerse productivo.

```
$ yo ionic [app-name]
```

En *package.json* incluye lo siguiente en devDependencies

```
"grunt-string-replace": "^1.2.1"
```

En *bower.json* incluyen lo siguiente en dependencias.

```
"ionic-platform-web-client": "^0.7.1"
```

En *Gruntfile.js*, cambie la carpeta de *scripts* a *'js'*. Cambie en *index.html* también si es necesario.

```
grunt.initConfig({  yeoman: {.....
  scripts: 'js',
  ..... } })
```

Entonces corre

```
$ bower install && npm install
$ grunt
$ grunt serve

$ cordova platform add android
$ grunt build:android --debug
```

---

## plataforma-iónica-cliente-web



### Un cliente web que proporciona interacciones con la plataforma Ionic.

Necesitamos algún código para permitir que su aplicación hable con la plataforma iónica. Necesitamos agregar el cliente web de la plataforma Ionic para que la aplicación Ionic interactúe con los complementos y la plataforma Ionic.io.

```
$ ionic io init
```

En su *app.js* agregue la dependencia del módulo *'ionic.service.core'*. En *Gruntfile.js* agregue la tarea grunt *'ionicSettings'* como se indica a continuación.

```
grunt.initConfig({
  ionicSettings: JSON.stringify(grunt.file.readJSON('./.io-config.json')),

  ionicIoBundlePath: 'www/bower_components/ionic-platform-web-
client/dist/ionic.io.bundle.min.js',

  'string-replace': {
    ionicSettings: {
      files: {
        '<%= ionicIoBundlePath %>': '<%= ionicIoBundlePath %>',
      },
    },
    options: {
```

```

replacements: [
  {
    pattern:
      "IONIC_SETTINGS_STRING_START";"IONIC_SETTINGS_STRING_END"',
    replacement:
      "IONIC_SETTINGS_STRING_START";var settings =<%= ionicSettings %>; return { get:
function(setting) { if (settings[setting]) { return settings[setting]; } return null; }
};"IONIC_SETTINGS_STRING_END";'
  }
]
},
  copy: {
    ionicPlatform:{
      expand: true,
      cwd: 'app/bower_components/ionic-platform-web-client/dist/',
      src: ['**'],
      dest: 'www/bower_components/ionic-platform-web-client/dist'
    }
  }
});

grunt.registerTask('ionicSettings', ['copy:ionicPlatform','string-replace:ionicSettings']);

```

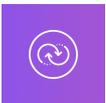
Agregue el *'ionicSettings'* en *init* y *comprima las* tareas después de la  *copia* . En *index.html*, mueva la siguiente etiqueta después de todas las declaraciones de etiquetas.

```
<script src="bower_components/ionic-platform-web-client/dist/ionic.io.bundle.min.js"></script>
```

Entonces corre

```
$ Grunt serve
```

## Despliegue jónico



### Inserte actualizaciones en tiempo real de sus aplicaciones de producción y administre el historial de versiones.

Ionic Deploy le permite actualizar su aplicación a pedido, para cualquier cambio que no requiera modificaciones binarias, ahorrando días o incluso semanas de tiempo de espera. Siga el procedimiento a continuación para configurar Ionic Deploy para su aplicación.

En *Gruntfile.js* agregue la tarea grunt *'desplegar'* como se indica a continuación.

```

grunt.registerTask('deploy', function () {
  return grunt.task.run(['init', 'ionic:upload' + this.args.join()]);
});

```

```
});
```

entonces corre

```
$ ionic plugin add ionic-plugin-deploy
```

Código de implementación iónica:

```
var deploy = new Ionic.Deploy();

// Check Ionic Deploy for new code
deploy.check().then(function(hasUpdate) {
}, function(err) {
});

// Update app code with new release from Ionic Deploy
deploy.update().then(function(result) {
}, function(error) {
}, function(progress) {
});
```

Implementando actualizaciones:

Enviar nuevo código para su aplicación.

Crea un apk e instala tu aplicación. Realice algunos cambios en su código e implemente los cambios utilizando ' *grunt deploy* '. Luego actualízalo desde tu aplicación.

También puede implementar desde el tablero de instrumentos *apps.ionic.io*. Puede implementar la aplicación sin el parámetro de despliegue. Luego, en el tablero de instrumentos, puede agregar los metadatos y los detalles de la versión y desplegar la aplicación desde allí.

```
$ grunt build:android --debug

$ grunt deploy --note "release notes"
$ grunt deploy --note "release notes" --deploy=production
```

---

## Analítica Iónica



**Ver la transmisión en vivo de eventos o el número de eventos / usuarios en bruto / único a lo largo del tiempo.**

¿Cuántos usuarios hay en tu aplicación en este momento? ¿Cuántos de ellos usarán tu aplicación mañana o la próxima semana? Sin información, no tiene forma de saber si su aplicación se está utilizando de la manera que usted espera. Siga el procedimiento a continuación para configurar

Ionic Analytics para su aplicación.

En su *app.js*, agregue la dependencia del módulo ' *ionic.service.analytics* ' después de *ionic.service.core* Ejecute el método de registro analítico en la función de ejecución de nuestro módulo.

```
$ionicAnalytics.register();
```

En Ionic Analytics, cada acción rastreada que un usuario realiza en su aplicación está representada por un objeto de evento. Un evento es una acción única realizada en un punto específico en el tiempo. Para rastrear sus propios eventos, llame a

`$ionicAnalytics.track(eventType, eventData)` siempre que ocurra una acción.

```
$ionicAnalytics.track('User Login', {  
  user: $scope.user  
});
```

La directiva *ion-track-tap* envía un evento cuando se *toca* su elemento host. La directiva de *datos de seguimiento de iones* asociada adjunta datos de eventos.

```
<button ion-track-tap="eventType" ion-track-data="expression"></button>
```

En el *panel de control* de *apps.ionic.io* puede ver los siguientes datos de análisis,

**Eventos:** vea la cantidad bruta de eventos a lo largo del tiempo, o la cantidad de usuarios únicos que completaron un evento. Un evento puede ser cualquier cosa, desde un usuario que carga la aplicación, hasta confirmar una compra.

**Embudos:** Un embudo es una secuencia de acciones que espera que los usuarios realicen en su aplicación, lo que lleva a un objetivo definido. El uso cuidadoso de los embudos le ayudará a mejorar las tasas de conversión.

**Segmentos:** vea eventos a lo largo del tiempo, agrupados por una propiedad específica. O bien, calcule el porcentaje de eventos que coinciden con una propiedad determinada. Los segmentos le ayudan a comprender su base de usuarios y ver cómo cambian las propiedades con el tiempo.

**Retención:** realice un seguimiento del tiempo que los usuarios están activos en su aplicación antes de que dejen de usarla. O bien, identifique cuánto tiempo tardan los usuarios en alcanzar un objetivo definido, como una venta completa.

**Pulse:** una transmisión en vivo de eventos que llegan de sus usuarios.

---

## Empuje ionic



# Envíe notificaciones push dirigidas y automatizadas a sus usuarios.

Ionic Push le permite crear notificaciones push dirigidas a través de un panel simple que se enviará automáticamente cuando los usuarios cumplan con criterios específicos, y ofrece una API simple para enviar notificaciones push desde sus propios servidores.

## Perfiles Push de Android:

Las notificaciones push de Android utilizan el servicio *Google Cloud Messaging* (GCM). Abra la [Consola de desarrolladores de Google](#) y cree un proyecto. Anote su *número de proyecto*. Este será el *ID de remitente de GCM* o el **Número de proyecto de GCM**.

En la sección *Administrador de API*, habilite la *API de mensajería en la nube de Google*. Luego navegue a la sección *Credenciales* y seleccione *Crear credenciales*, luego elija *Clave de API*, luego *Clave de servidor*. Asigne un nombre a su clave API y deje en blanco el campo *Aceptar solicitudes de ...* y haga clic en *Crear*. Guarde su **clave API**!

## Autenticación:

Vaya al panel de control de su aplicación en la [plataforma iónica](#) y navegue a *Configuración -> Certificados*. Si aún no lo ha hecho, cree un nuevo perfil de seguridad, luego presione *editar*. Anote la **etiqueta de perfil**.

Ahora, haga clic en la pestaña *Android* y busque la sección marcada *Google Cloud Messaging*, ingrese la *Clave API* que generó en la Consola de desarrollador de Google, luego haga clic en *Guardar*. Vaya a *Configuración -> Claves API*. Bajo *API Tokens*, cree un nuevo token y cópielo. Esto será su **token de API**.

```
$ ionic plugin add phonegap-plugin-push --variable SENDER_ID="GCM_PROJECT_NUMBER"
$ ionic config set gcm_key <your-gcm-project-number>
$ ionic config set dev_push false
$ ionic io init
```

Nota: phonegap-plugin-push requiere el repositorio de soporte de Android versión 32+

En su *app.js* agregue la dependencia del módulo '*ionic.service.push*' después de *ionic.service.core*

Código de empuje iónico:

Inicialice el servicio y registre su dispositivo en la función de ejecución de su módulo. Necesitará el token del dispositivo que está registrado por el usuario para enviar una notificación al usuario.

```
$ionicPush.init({
  debug: true,
  onNotification: function (notification) {
    console.log'token:', notification.payload);
  },
},
```

```
onRegister: function (token) {
  console.log('Device Token:', token);
  $ionicPush.saveToken(token); // persist the token in the Ionic Platform
}
});

$ionicPush.register();
```

entonces corre

```
$ grunt build:android --debug
```

Ionic Push le permite crear notificaciones push dirigidas a través del panel de control. También puede enviar notificaciones desde el servidor en el siguiente formato.

```
curl -X POST -H "Authorization: Bearer API_TOKEN" -H "Content-Type: application/json" -d '{
  "tokens": ["DEVICE_TOKEN"],
  "profile": "PROFILE_TAG",
  "notification": {
    "message": "Hello World!"
  }
  "android": {
    "title": "Hi User",
    "message": "An update is available for your App",
    "payload": {
      "update": true
    }
  }
}' "https://api.ionic.io/push/notifications"
```

Nota: los pasos para configurar Ionic Push para iOS son los mismos, excepto para crear los perfiles Push. Para crear perfiles push de iOS, consulte <http://docs.ionic.io/v2.0.0-beta/docs/ios-push-profiles>

---

## Aplicación de ejemplo



[Descarga la aplicación de muestra aquí](#) .

Una aplicación de ejemplo se adjunta aquí para referencia.

```
IonicApp:
|
|  bower.json
|  Gruntfile.js
|  package.json
|
└── app
    |  index.html
    |
    └── js
```

```
|   app.js
|   controllers.js
|
└── templates
    home.html
    menu.html
```

Nota: Este no es un proyecto independiente. El código dado es solo para comparación con un proyecto creado e implementado utilizando los procedimientos descritos anteriormente en este documento, en caso de problemas o errores.

Lea **Comenzando con el marco iónico en línea**: <https://riptutorial.com/es/ionic-framework/topic/893/comenzando-con-el-marco-ionico>

# Capítulo 2: ¿Cómo utilizar las características de EcmaScript 6 en Ionic?

## Examples

### Usando gulp-babel y gulp-plumber

Ionic usa Gulp, así que instala gulp-babel y gulp-plumber.

```
npm install --save-dev gulp-babel gulp-plumber
```

Agrega babel a `gulpfile.js` así:

```
//...
var babel = require("gulp-babel");
var plumber = require("gulp-plumber");

var paths = {
  es6: ['./src/es6/*.js'],
  sass: ['./scss/**/*.scss']
};

gulp.task('default', ['babel', 'sass']);

gulp.task("babel", function () {
  return gulp.src(paths.es6)
    .pipe(plumber())
    .pipe(babel())
    .pipe(gulp.dest("www/js"));
});

//...

gulp.task('watch', function() {
  gulp.watch(paths.es6, ['babel']);
  gulp.watch(paths.sass, ['sass']);
});

//...
```

Editar `ionic.project`:

```
"gulpStartupTasks": [
  "babel",
  "sass",
  "watch"
],
```

Ahora, cuando ejecute el `ionic serve`, el código será transpilado para usted.

Lea [¿Cómo utilizar las características de EcmaScript 6 en Ionic? en línea:](https://riptutorial.com/es/ionic-framework/topic/4847/-como-utilizar-las-caracteristicas-de-)

<https://riptutorial.com/es/ionic-framework/topic/4847/-como-utilizar-las-caracteristicas-de->



---

# Capítulo 3: ¿Cuál es la diferencia entre "acumulación iónica" y "preparación iónica"?

## Examples

### constitución iónica vs preparación iónica

De la documentación oficial:

Si desea avanzar, también puede abrir el archivo de proyecto para una plataforma específica abriendo el proyecto XCode o Android Eclipse requerido en plataformas / PLATFORM dentro de la raíz de su proyecto. Luego, puede crear y probar desde el IDE específico de la plataforma. Nota: si sigue esta ruta, le recomiendo que siga trabajando dentro de la carpeta raíz www, y cuando haya realizado cambios en esta carpeta, ejecute el comando: `$ cordova prepare ios` que actualizará el proyecto específico de iOS con el código del carpeta de `www`. Nota: esto sobrescribirá cualquier cambio que haya realizado en las `platforms/ios/www` y otras carpetas específicas de la plataforma.

Entonces, para resumir esta parte, si está utilizando XCode para probar y ejecutar su código, después de cambiar alguna parte del código, solo tiene que ejecutar `ionic prepare` para actualizar el proyecto de iOS, que luego continuará utilizando en XCode.

`ionic build` comando `ionic build` realidad prepara el archivo final (por ejemplo, en Android es el archivo `.apk`) que luego se puede copiar en su dispositivo y probar ejecutándolo manualmente en el dispositivo (o usando el comando `ionic emulate` para probarlo en el emulador).).

Lea ¿Cuál es la diferencia entre "acumulación iónica" y "preparación iónica"? en línea: <https://riptutorial.com/es/ionic-framework/topic/2911/-cual-es-la-diferencia-entre--acumulacion-ionica--y--preparacion-ionica-->

---

# Capítulo 4: Comenzar a crear aplicaciones en Ionic

## Examples

Comenzando una aplicación jónica

---

## Comenzando una aplicación jónica

```
$ ionic start myapp [template]
```

Las plantillas de inicio pueden provenir de una plantilla con nombre, un repositorio de Github, un Codepen o un directorio local. Una plantilla de inicio es lo que se convierte en el directorio `www` dentro del proyecto Cordova.

## Plantilla de inicio nombrados

- [pestañas](#) (por defecto)
- [sidemenu](#)
- [blanco](#)

## Entrantes Github Repo

- Cualquier url de repositorio de Github, por ejemplo: <https://github.com/driftyco/ionic-starter-tabs>
- Las plantillas nombradas son simplemente alias de los repositorios de inicio iónico

## Codepen URL starters

- Cualquier url de Codepen, por ejemplo: [ <http://codepen.io/ionic/pen/odqCz◆◆1> ]
- [Demostraciones Iónicas De Codepen](#)

## Iniciadores del directorio local:

- Ruta relativa o absoluta a un directorio local

## Indicadores de línea de comando / opciones:

```
--appname, -a ..... Human readable name for the app (Use quotes around the name)
--id, -i ..... Package name set in the <widget id> config ex: com.mycompany.myapp
--no-cordova, -w .... Do not create an app targeted for Cordova
```

Lea Comenzar a crear aplicaciones en Ionic en línea: <https://riptutorial.com/es/ionic-framework/topic/3945/comenzar-a-crear-aplicaciones-en-ionic>

---

# Capítulo 5: Componentes jónicos de CSS

## Observaciones

Ionic tiene una gran cantidad de componentes CSS declarados y listos para hacer tu vida más fácil mientras codificas tu próxima aplicación móvil híbrida. Estos componentes varían desde un sistema de cuadrícula básico hasta el estilo de sus formularios. Estos componentes están en uso si elige instalar Ionic con las hojas de estilo CSS preestablecidas.

Una de las funciones básicas que el Ionic CSS trae a tu mano es que viene con un conjunto de colores para comenzar, pero como regla general, los colores están destinados a ser anulados. Se agregan colores de utilidad para ayudar a establecer una convención de nomenclatura. Podrías llamarlo un tema básico de la aplicación. Para personalizar los colores, simplemente puede reemplazar los que vienen del archivo CSS de `ionic.css`. Además, como Ionic se ha creado utilizando Sass, para obtener más potencia y flexibilidad, también puede modificar y ampliar las variables de color dentro del archivo `_variables.scss`.

Puede configurar un proyecto Ionic para usar SASS muy fácilmente ejecutando el comando `ionic setup sass` en su terminal.

Puede encontrar la documentación oficial en Ionic CSS aquí:

<http://ionicframework.com/docs/components/>

## Examples

### Sintaxis básica del sistema grid

---

## Rejilla basica

En Ionic puedes declarar filas configurando la clase de `row` a un elemento. Las filas serán elementos que están alineados horizontalmente y cualquier cosa dentro de este elemento, todo pertenecerá a la fila. Dentro de la fila puedes declarar diferentes columnas de ancho. Usted tiene una opción de las siguientes declaraciones.

Clase	Anchura
<code>.col-10</code>	10%
<code>.col-20</code>	20%
<code>.col-25</code>	25%
<code>.col-33</code>	33.3333%
<code>.col-50</code>	50%

Clase	Anchura
.col-67	66.6666%
.col-75	75%
.col-80	80%
.col-90	90%

El valor máximo que pueden tener las columnas dentro de una fila es 100. Aquí hay algunos ejemplos de la cuadrícula básica.

```
<div class="row">
  <div class="col col-50">.col.col-50</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>

<div class="row">
  <div class="col col-75">.col.col-75</div>
  <div class="col">.col</div>
</div>

<div class="row">
  <div class="col">.col</div>
  <div class="col col-75">.col.col-75</div>
</div>

<div class="row">
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```

## Rejillas de compensación

También puede establecer `col-offset-<value>` en las columnas. En el siguiente ejemplo, el tercio de una columna de ancho tiene un tercio del ancho de desplazamiento, lo que significa que tendrá un ancho de un tercio de ancho y estará centrado en la página debido a su desplazamiento.

```
<div class="row">
  <div class="col col-33 col-offset-33">.col</div>
  <div class="col">.col</div>
</div>
```

## Alinear columnas

Alinear las columnas verticalmente es posible configurando el `col-<align_value>` en una columna por separado como esta.

```
<div class="row">
  <div class="col col-top">.col</div>
  <div class="col col-center">.col</div>
  <div class="col col-bottom">.col</div>
  <div class="col">1<br>2<br>3<br>4</div>
</div>
```

Lo anterior alineará cada columna por su cuenta. Si desea alinear todas las columnas dentro de la fila, puede establecer el valor de alineación en la propia fila. En el siguiente ejemplo, todas las columnas dentro de esta fila se alinearán verticalmente en el centro de la fila.

```
<div class="row row-center">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">1<br>2<br>3<br>4</div>
</div>
```

## Cuadrícula sensible

También es posible que desee que las columnas respondan, ya que se apilarán unas sobre otras en algún ancho de ventana gráfica. Tienes tres opciones.

Clase	Punto de quiebre (aproximadamente)
.responsivo-sm	Más pequeño que el teléfono paisaje
.responsivo-md	Más pequeña que la tableta retrato
.responsivo-lg	Más pequeña que la tableta de paisaje

En este ejemplo, estas columnas se apilarán bajo el ancho de aproximadamente un teléfono horizontal.

```
<div class="row responsive-sm">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```

Por supuesto, también puede realizar sus propias consultas de medios si estos puntos de interrupción no son adecuados para usted y / o si necesita puntos de interrupción más específicos.

### Sintaxis básica del elemento de la lista

Casi todas las aplicaciones tienen algún tipo de lista. Ionic tiene sus propias declaraciones CSS de elementos de listas listas para usar para que sea más fácil hacer listas dentro de su aplicación.

Puede utilizar los elementos HTML y declarar una clase para el o usar la directiva `ion-list` para hacerlos. El ejemplo de una directiva está en la parte inferior.

Sintaxis CSS básica del elemento de la lista:

```
<ul class="list">
  <li class="item"></li>
</ul>
```

Lista con los divisores:

```
<div class="list">
  <a class="item" href="#">
    List item
  </a>
  <div class="item item-divider">
    Divider that looks a bit different from items
  </div>
  <a class="item" href="#">
    Another list item
  </a>
</div>
```

Lista de artículos con iconos:

```
<div class="list">
  <a class="item item-icon-left" href="#">
    <i class="icon ion-trash-b"></i>
    List item with a trashcan icon on the left
  </a>
</div>
```

También puede establecer iconos en ambos lados de los elementos con la siguiente sintaxis:

```
<div class="list">
  <a class="item item-icon-left item-icon-right" href="#">
    <i class="icon ion-trash-b"></i>
    List item with a trashcan icon on the left and a briefcase icon on the right
    <i class="icon ion-briefcase"></i>
  </a>
</div>
```

Un elemento de lista con botón o botones puede crearse así:

```
<div class="list">
  <div class="item item-button-right">
    Item with a button on the right that has a clock icon in it
    <button class="button button-positive">
      <i class="icon ion-clock"></i>
    </button>
  </div>
</div>
```

También es posible crear elementos de lista con avatares, miniaturas e inserciones, lo que creará

relleno alrededor de los elementos de la lista. Ionic también maneja los íconos de configuración, etc. en los elementos de la lista al configurar el relleno de acuerdo con los elementos de la lista.

Ionic también tiene sus propias directivas para casillas de verificación, botones de radio, etc. Aquí hay un ejemplo de una lista de casillas de verificación con Ionic.

```
<ion-list>
  <ion-checkbox ng-model="choice1">Choice 1</ion-checkbox>
  <ion-checkbox ng-model="choice2">Choice 2</ion-checkbox>
</ion-list>
```

## Uso básico de colores de utilidad.

Preset Ionic CSS tendrá un tema y colores preestablecidos para ello. Puede modificar o anular los valores básicos en ionic.css o en su archivo CSS personalizado. También puede definirlos con SASS y para usar SASS en Ionic solo necesita ejecutar el comando `ionic setup sass` en su terminal.

Uso básico de colores en un botón. La clase de botón `button-<phrase>` hará que el fondo del botón y los bordes del color del tema establecido.

```
<button class="button button-positive">
  button-positive
</button>

<button class="button button-calm">
  button-calm
</button>

<button class="button button-balanced">
  button-balanced
</button>
```

Sus opciones de prefijo CSS son las siguientes:

- `<element>-light`
- `<element>-stable`
- `<element>-positive`
- `<element>-calm`
- `<element>-balanced`
- `<element>-energized`
- `<element>-assertive`
- `<element>-royal`
- `<element>-dark`

Estas clases se pueden agregar también, por ejemplo, en credenciales, etc. Aquí hay un ejemplo

de una credencial:

```
<span class="badge badge-positive">Positive badge</span>
```

Lea Componentes jónicos de CSS en línea: <https://riptutorial.com/es/ionic-framework/topic/6689/componentes-jonicos-de-css>

---

# Capítulo 6: Conectando Ionic con cualquier base de datos

## Examples

No puedes hacerlo directamente desde el framework Ionic.

La cosa es; no se puede conectar Ionic a ninguna base de datos (MySQL, Postgres, MSSQL, ...) **directamente** . La palabra clave aquí es *directamente* .

No, no hay solución, no hay magia involucrada, simplemente no es la forma en que se supone que funciona. Las obras iónicas en la parte superior de Angular y Angular son un marco de frontend.

Sin embargo, la forma en que debe hacerlo es que básicamente crea una API (RESTful) en su servidor.

Lo más probable es que esto se haga con algún idioma del lado del servidor (PHP, Go, Python, ...) que se comunique directamente con su base de datos y lo consulte.

Después de escribir su API (RESTful), puede consumirla a través de sus servicios en Angular utilizando el `$resource` o `$http` Angular.

Un ejemplo de consumir Giphy API con el servicio `$http` Angular:

```
var search = 'cats';
var link = 'http://api.giphy.com/v1/gifs/search?api_key=dc6zaTOxFJmzC&q=' + search;

$http.get(link).then(function(result) {
  console.log(result);
});
```

Lea Conectando Ionic con cualquier base de datos en línea: <https://riptutorial.com/es/ionic-framework/topic/4003/conectando-ionic-con-cualquier-base-de-datos>

# Capítulo 7: Crear diálogo en jónico

## Parámetros

Parámetros	Detalle
<code>title: ''</code> ,	// Cuerda. El título de la ventana emergente.
<code>cssClass: ''</code> ,	// Cadena, el nombre de la clase CSS personalizado
<code>subTitle: ''</code> ,	// Cadena (opcional). El subtítulo de la ventana emergente.
<code>template: ''</code> ,	// Cadena (opcional). La plantilla html para colocar en el cuerpo emergente.
<code>templateUrl: ''</code> ,	// Cadena (opcional). La URL de una plantilla html para colocar en el cuerpo emergente.
<code>scope: null</code> ,	// Ámbito (opcional). Un ámbito para enlazar al contenido emergente.

## Observaciones

El servicio Ionic Popup permite crear y mostrar ventanas emergentes mediante programación que requieren que el usuario responda para continuar.

## Examples

### Crear diálogo en jónico

```
// An alert dialog
$scope.showAlert = function() {
  var alertPopup = $ionicPopup.alert({
    title: 'Don\'t eat that!',
    template: 'It might taste good'
  });

  alertPopup.then(function(res) {
    console.log('Hello your first example.');
```

Lea [Crear diálogo en jónico en línea](https://riptutorial.com/es/ionic-framework/topic/6461/crear-dialogo-en-jonico): <https://riptutorial.com/es/ionic-framework/topic/6461/crear-dialogo-en-jonico>

# Capítulo 8: Desplazamiento infinito iónico para mostrar elementos de carga a pedido (datos ya disponibles que no se encuentran en la solicitud HTTP)

## Examples

### Carga n número de datos disponibles bajo demanda

#### HTML:

```
<li class="item" ng-repeat="schedule in Schedules | filter:scheduleSearch |
limitTo:numberOfItemsToDisplay">
  Display some data
</li>
<ion-infinite-scroll on-infinite="addMoreItem()" ng-if="Schedules.length >
numberOfItemsToDisplay"></ion-infinite-scroll>
```

#### Controlador :

```
$scope.numberOfItemsToDisplay = 10; // Use it with limit to in ng-repeat
$scope.addMoreItem = function(done) {
  if ($scope.Schedules.length > $scope.numberOfItemsToDisplay)
    $scope.numberOfItemsToDisplay += 10; // load number of more items
    $scope.$broadcast('scroll.infiniteScrollComplete')
}
```

Cargue 10 elementos cada vez que llame addMoreItem ().

Lea Desplazamiento infinito iónico para mostrar elementos de carga a pedido (datos ya disponibles que no se encuentran en la solicitud HTTP) en línea: <https://riptutorial.com/es/ionic-framework/topic/9490/desplazamiento-infinito-ionico-para-mostrar-elementos-de-carga-a-pedido--datos-ya-disponibles-que-no-se-encuentran-en-la-solicitud-http->

---

# Capítulo 9: Desplegar Ionic como un sitio web

## Examples

### Simplemente copie la carpeta www a su servidor web

Ionic 1.2 [admite](#) oficialmente la [implementación como un sitio web](#)

Si no está utilizando ningún complemento de Cordova, entonces no hay ningún problema (si realmente lo desea) para cargar los contenidos de la carpeta `www` en su servidor, y woila: tendrá la misma aplicación.

Sin embargo, es importante tener en cuenta que Ionic 1 nunca fue diseñado para tal uso, y los usuarios de su "sitio web" deberán tener el navegador más nuevo para poder ver el "sitio web" correctamente (no desglosado debido a alguna característica que Ionic utiliza CSS o HTML que algunos navegadores antiguos no admiten).

Lea [Desplegar Ionic como un sitio web en línea](#): <https://riptutorial.com/es/ionic-framework/topic/4001/desplegar-ionic-como-un-sitio-web>

# Capítulo 10: Dispositivo de cámara y acceso a la biblioteca de fotos desde la aplicación Ionic

## Observaciones

También refiera este [enlace](#)

## Examples

Camara abierta y galeria de fotos.

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="Content-Security-Policy" content="default-src *; script-src 'self'
'unsafe-inline' 'unsafe-eval' *; style-src 'self' 'unsafe-inline' */>
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no,
width=device-width">
    <title></title>

    <link href="lib/ionic/css/ionic.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">

    <!-- IF using Sass (run gulp sass first), then uncomment below and remove the CSS includes
above
    <link href="css/ionic.app.css" rel="stylesheet">
    -->

    <!-- ionic/angularjs js -->
    <script src="lib/ionic/js/ionic.bundle.js"></script>

    <!-- cordova script (this will be a 404 during development) -->
    <script src="js/ng-cordova.min.js"></script>
    <script src="cordova.js"></script>

    <!-- your app's js -->
    <script src="js/app.js"></script>
    <script src="js/controllers.js"></script>
    <script src="js/services.js"></script>
  </head>
  <body ng-app="starter">
    <ion-content class="has-header padding" ng-controller="ImageController">
      <button class="button button-full button-energized" ng-click="addMedia()">
        Add image
      </button>
      <button class="button button-full button-positive" ng-click="sendEmail()">
        Send mail
    </ion-content>
  </body>
</html>
```

```

</button>
<br><br>
<ion-scroll direction="y" style="height:200px; min-height: 200px; overflow: scroll; white-
space: nowrap;">
  
</ion-scroll>

</ion-content>
</body>
</html>

```

## controller.js

```

angular.module('starter')

.controller('ImageController', function($scope, $cordovaDevice, $cordovaFile, $ionicPlatform,
$cordovaEmailComposer, $ionicActionSheet, ImageService, FileService) {

  $ionicPlatform.ready(function() {
    $scope.images = FileService.images();
    $scope.$apply();
  });

  $scope.addMedia = function() {
    $scope.hideSheet = $ionicActionSheet.show({
      buttons: [
        { text: 'Take photo' },
        { text: 'Photo from library' },
        { text: '<b ng-disabled="user.length<1">Delete</b>',
          type: 'input type="file"'
        },
      ],
      titleText: 'Add images',
      cancelText: 'Cancel',
      buttonClicked: function(index) {
        $scope.addImage(index);
      }
    });
  }

  $scope.addImage = function(type) {
    $scope.hideSheet();
    ImageService.handleMediaDialog(type).then(function() {
      $scope.$apply();
    });
  }

  $scope.sendEmail = function() {
    if ($scope.images != null && $scope.images.length > 0) {
      var mailImages = [];
      var savedImages = $scope.images;
      for (var i = 0; i < savedImages.length; i++) {
        mailImages.push('base64:attachment'+i+'.jpg//' + savedImages[i]);
      }
      $scope.openMailComposer(mailImages);
    }
  }

  $scope.openMailComposer = function(attachments) {
    var bodyText = '<html><h2>My Images</h2></html>';

```

```

var email = {
  to: '',
  attachments: attachments,
  subject: 'Devdactic Images',
  body: bodyText,
  isHtml: true
};

$cordovaEmailComposer.open(email, function(){
  console.log('email view dismissed');

}, this);
}
});

```

## service.js

```

angular.module('starter')

.factory('FileService', function() {
  var images;
  var IMAGE_STORAGE_KEY = 'dav-images';

  function getImages() {
    var img = window.localStorage.getItem(IMAGE_STORAGE_KEY);
    if (img) {
      images = JSON.parse(img);
    } else {
      images = [];
    }
    return images;
  };

  function addImage(img) {
    images.push(img);
    window.localStorage.setItem(IMAGE_STORAGE_KEY, JSON.stringify(images));
  };

  return {
    storeImage: addImage,
    images: getImages
  }
});

.factory('ImageService', function($cordovaCamera, FileService, $q, $cordovaFile) {

  function optionsForType(type) {
    var source;
    switch (type) {
      case 0:
        source = Camera.PictureSourceType.CAMERA;
        break;
      case 1:
        source = Camera.PictureSourceType.PHOTOLIBRARY;
        break;
    }
    return {
      quality: 90,
      destinationType: Camera.DestinationType.DATA_URL,
      sourceType: source,

```

```

    allowEdit: false,
    encodingType: Camera.EncodingType.JPEG,
    popoverOptions: CameraPopoverOptions,
    saveToPhotoAlbum: false,
    correctOrientation:true
  };
}

function saveMedia(type) {
  return $q(function(resolve, reject) {
    var options = optionsForType(type);

    $cordovaCamera.getPicture(options).then(function(imageBase64) {
      FileService.storeImage(imageBase64);
    });
  })
}
return {
  handleMediaDialog: saveMedia
}
});

```

## Para Ionic 3 Ejemplo

Instala los complementos Cordova y Ionic Native:

```

$ ionic cordova plugin add cordova-plugin-camera
$ npm install --save @ionic-native/camera

```

Tu `app.module.ts` necesitará inyectar la cámara:

```

import { Camera } from '@ionic-native/camera';
.....
@NgModule({
  declarations: [
    MyApp
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
    .....
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp
  ],
  providers: [
    StatusBar,
    SplashScreen,
    Camera,
    {provide: ErrorHandler, useClass: IonicErrorHandler},
    .....
  ]
})
export class AppModule {}

```

La cámara se puede usar con la hoja de acción fácilmente con Ionic 3, su página será como a

continuación:

```
import { Camera, CameraOptions } from '@ionic-native/camera';
.....

export class YourPage {

  private base64:any;

  constructor(private camera: Camera,private actionsheetCtrl: ActionSheetController) { }

  cameragalleryfun(){

    let actionSheet = this.actionsheetCtrl.create({
      title: 'Camera-Gallery',
      cssClass: 'action-sheets-basic-page',
      buttons: [
        {
          text: 'Camera',
          icon: 'camera',
          handler: () => {
            //console.log('Camera');
            const options: CameraOptions = {
              quality: 60,
              destinationType: this.camera.DestinationType.DATA_URL,
              encodingType: this.camera.EncodingType.JPEG,
              mediaType: this.camera.MediaType.PICTURE,
              sourceType : this.camera.PictureSourceType.CAMERA,
              targetWidth: 500,
              saveToPhotoAlbum: false,
              correctOrientation:true
            }

            this.camera.getPicture(options).then((imageData) => {
              this.base64 = 'data:image/jpeg;base64,' + imageData;
            }, (err) => {
              // Handle error
            });
          }
        },
        {
          text: 'Gallery',
          icon: 'images',
          handler: () => {
            //console.log('Gallery');
            const options: CameraOptions = {
              quality: 60,
              destinationType: this.camera.DestinationType.DATA_URL,
              encodingType: this.camera.EncodingType.JPEG,
              mediaType: this.camera.MediaType.PICTURE,
              sourceType : this.camera.PictureSourceType.PHOTOLIBRARY,
              targetWidth: 500,
              saveToPhotoAlbum: false,
              correctOrientation:true
            }

            this.camera.getPicture(options).then((imageData) => {
              this.base64 = 'data:image/jpeg;base64,' + imageData;
            }, (err) => {
              // Handle error
            });
          }
        }
      ]
    });
  }
}
```

```
        });  
    }  
  },  
  {  
    text: 'Cancel',  
    role: 'cancel', // will always sort to be on the bottom  
    icon: 'close',  
    handler: () => {  
      //console.log('Cancel clicked');  
    }  
  }  
]  
});  
actionSheet.present();  
  
}  
}
```

Llame a la función `cameragalleryfun` desde cualquier evento como hacer clic en el botón, esto devolverá la cadena base64 para la imagen. Se puede aplicar más opción. para la opción adicional, consulte: <https://github.com/apache/cordova-plugin-camera>

Lea **Dispositivo de cámara y acceso a la biblioteca de fotos desde la aplicación Ionic en línea:** <https://riptutorial.com/es/ionic-framework/topic/7086/dispositivo-de-camara-y-acceso-a-la-biblioteca-de-fotos-desde-la-aplicacion-ionic>

---

# Capítulo 11: Ejecute la aplicación Ionic en el emulador o en su teléfono

## Examples

Ejecute la aplicación Ionic en el emulador o en su teléfono

### 1. Añadir un objetivo de plataforma

iOS:

```
$ ionic platform add ios
```

Androide:

```
$ ionic platform add android
```

Windows:

```
$ ionic platform add windows
```

---

### 2. Construye tu aplicación

iOS:

```
$ ionic build ios
```

Androide:

```
$ ionic build android
```

Windows:

```
$ ionic build windows
```

---

### Aplicación Live Reload durante el desarrollo (beta)

El comando `run` o `emulate` implementará la aplicación en los dispositivos / emuladores de plataforma especificados. También puede ejecutar la **recarga en vivo** en el dispositivo de plataforma especificado agregando la opción `--livereload`. La funcionalidad de recarga en vivo es

similar al `ionic serve` , pero en lugar de desarrollar y depurar una aplicación usando un navegador estándar, la aplicación híbrida compilada está observando cualquier cambio en sus archivos y recargando la aplicación cuando sea necesario. Esto reduce el requisito de reconstruir constantemente la aplicación para pequeños cambios. Sin embargo, cualquier cambio en los complementos requerirá una reconstrucción completa. Para que la recarga en vivo funcione, la máquina y el dispositivo de desarrollo deben estar en la misma red local, y el dispositivo debe ser compatible con [sockets web](#) .

Con la recarga activa habilitada, los registros de la consola de una aplicación también se pueden imprimir en el terminal / comando al incluir la opción `--consolelogs o -c` . Además, los registros de solicitudes del servidor de desarrollo se pueden imprimir utilizando las opciones `--serverlogs o -s` .

## Indicadores / opciones de línea de comando para `run y emulate`

```
[--livereload|-l] ..... Live Reload app dev files from the device (beta)
[--consolelogs|-c] ..... Print app console logs to Ionic CLI (live reload req.)
[--serverlogs|-s] ..... Print dev server logs to Ionic CLI (live reload req.)
[--port|-p] ..... Dev server HTTP port (8100 default, live reload req.)
[--livereload-port|-i] .. Live Reload port (35729 default, live reload req.)
[--debug|--release]
```

Mientras el servidor se está ejecutando para la recarga en vivo, puede usar los siguientes comandos dentro de la CLI:

```
restart or r to restart the client app from the root
goto or g and a url to have the app navigate to the given url
consolelogs or c to enable/disable console log output
serverlogs or s to enable/disable server log output
quit or q to shutdown the server and exit
```

## 3. Emulando tu aplicación

Despliega la aplicación Ionic en el emulador de plataforma especificado. Esto es simplemente un alias para `run --emulator` .

### iOS:

```
$ ionic emulate ios [options]
```

### Androide:

```
$ ionic emulate android [options]
```

### Windows:

```
$ ionic emulate windows [options]
```

Durante la emulación de la aplicación en dispositivos AVD o móviles, puede inspeccionar esa

aplicación en el navegador Chrome. Escriba el siguiente comando en la barra de direcciones del navegador Chrome.

```
chrome://inspect/#devices
```

---

## 4. Ejecutando tu aplicación

Despliega la aplicación Ionic en dispositivos de plataforma especificados. Si no se encuentra un dispositivo, se implementará en un emulador / simulador.

### iOS:

```
$ ionic run ios [options]
```

### Androide:

```
$ ionic run android [options]
```

### Windows:

```
$ ionic run windows [options]
```

### 4.1. Especificando su objetivo

```
$ ionic run [ios/android/windows] --target="[target-name]"
```

Puede verificar el nombre de destino de su dispositivo / emulador que ejecuta `$ adb devices`.

Lea [Ejecute la aplicación Ionic en el emulador o en su teléfono en línea](https://riptutorial.com/es/ionic-framework/topic/4175/ejecute-la-aplicacion-ionic-en-el-emulador-o-en-su-telefono):

<https://riptutorial.com/es/ionic-framework/topic/4175/ejecute-la-aplicacion-ionic-en-el-emulador-o-en-su-telefono>

---

# Capítulo 12: Extensiones Ionic AngularJS

## Observaciones

Ionic ofrece una variedad de extensiones de Javascript AngularJS para su uso. Estas extensiones pueden ser desde entradas de formularios normales hasta ventanas modales y hacen que la codificación de su aplicación básica sea mucho más rápida con estas extensiones listas para usar.

## Examples

### Entradas de formulario

Las entradas iónicas no son diferentes de las entradas HTML normales, pero tienen un estilo diferente y se usan como una directiva. También puede utilizar entradas HTML normales en aplicaciones Ionic. Aquí hay algunos ejemplos básicos que Ionic ofrece listos para usar.

#### Caja:

```
<ion-checkbox ng-model="checkbox">Label</ion-checkbox>
```

#### Boton de radio:

```
<ion-radio ng-model="radio" ng-value="'radiovalue'">Value</ion-radio>
```

#### Palanca:

```
<ion-toggle ng-model="toggle" toggle-class="toggle-calm">Toggle</ion-toggle>
```

### Ventanas modales

Ionic tiene su propia extensión para mostrar una ventana modal. Los modales se pueden crear insertando la plantilla directamente en la vista con una etiqueta `<script>` o usando un archivo de plantilla separado. En este ejemplo, asumimos que tiene un archivo html llamado `modal-template.html` en una carpeta llamada `templates`. Establece la ruta de la plantilla en la función de inicialización modal con `fromTemplateUrl`.

---

## Creando el objeto modal en el alcance.

### HTML

```
<ion-modal-view>  
  <ion-header-bar>
```

```
<h1>Modal title</h1>
</ion-header-bar>
<ion-content>
  <p>Modal content</p>
</ion-content>
</ion-modal-view>
```

## Controlador

```
$ionicModal.fromTemplateUrl('/templates/modal-template.html', {
  scope: $scope,
  animation: 'slide-in-up'
}).then(function(modal) {
  $scope.modal = modal;
});
```

---

# Controlar el modal

A continuación, puede controlar el modal con los siguientes comandos.

## Abierto modal

```
$scope.modal.show();
```

## Cerrar modal

```
$scope.modal.hide();
```

## Quitar modal

```
$scope.modal.remove();
```

---

# Eventos modales

Puedes escuchar eventos modales con las siguientes funciones.

## Modal esta oculto

```
$scope.$on('modal.hidden', function() {
  // Do something when modal is hidden
});
```

## Se elimina el modal

```
$scope.$on('modal.removed', function() {
  // Do something when modal is removed
});
```

Lea Extensiones Ionic AngularJS en línea: <https://riptutorial.com/es/ionic-framework/topic/6446/extensiones-ionic-angularjs>

---

# Capítulo 13: Ganchos CLI iónicos

## Observaciones

---

## Introducción

Los ganchos son piezas de código que Cordova CLI ejecuta en ciertos puntos de su compilación de aplicaciones Cordova / Ionic. Los ganchos se pueden usar, por ejemplo, para manipular archivos en nuestro proyecto, agregar automáticamente complementos a su aplicación o, como en el ejemplo anterior, verificar si hay errores de código en sus archivos.

**Nota** : se recomienda escribir sus ganchos con Node.js para que sean multiplataforma, pero también puede escribirlos, por ejemplo, en [Javascript](#) .

---

## Tipos de gancho

Se admiten los siguientes tipos de gancho y el orden de ejecución se explica por sí mismo de acuerdo con el nombre.

```
after_build
after_compile
after_docs
after_emulate
after_platform_add
after_platform_rm
after_platform_ls
after_plugin_add
after_plugin_ls
after_plugin_rm
after_plugin_search
after_prepare
after_run
after_serve
before_build
before_compile
before_docs
before_emulate
before_platform_add
before_platform_rm
before_platform_ls
before_plugin_add
before_plugin_ls
before_plugin_rm
before_plugin_search
before_prepare
before_run
before_serve
pre_package/ <-- Applicable to Windows 8 and Windows Phone only. This hook is deprecated.
```

# Formas de definir ganchos:

Los ganchos podrían definirse en el `config.xml` usando elementos `<hook>` , por ejemplo:

```
<hook type="after_build" src="scripts/appAfterBuild.js" />
```

Como desarrollador de complementos, puede definir secuencias de comandos de enganche usando elementos `<hook>` en un `plugin.xml` como este:

```
<hook type="after_build" src="scripts/afterBuild.js" />
```

`before_plugin_install` , `after_plugin_install` , `before_plugin_uninstall` ganchos de complemento se activarán exclusivamente para la instalación / desinstalación del complemento.

**Nota** : colocar ganchos en el directorio `root/hooks` se considera obsoleto a favor de los elementos de gancho en `config.xml` y `plugin.xml` . Sin embargo, si utiliza este enfoque, recuerde establecer los derechos de ejecución de los archivos en la carpeta `root/hooks` .

La documentación para Cordova Hooks se puede encontrar [aquí](#) .

## Examples

### Comprobando errores en sus archivos de Javascript en `before_prepare` usando `jshint`

```
#!/usr/bin/env node

var fs = require('fs');
var path = require('path');
var jshint = require('jshint').JSHINT;
var async = require('async');

var foldersToProcess = [
  'js'
];

foldersToProcess.forEach(function(folder) {
  processFiles("www/" + folder);
});

function processFiles(dir, callback) {
  var errorCount = 0;
  fs.readdir(dir, function(err, list) {
    if (err) {
      console.log('processFiles err: ' + err);
      return;
    }
    async.eachSeries(list, function(file, innercallback) {
      file = dir + '/' + file;
      fs.stat(file, function(err, stat) {
        if (!stat.isDirectory()) {
```

```

        if (path.extname(file) === ".js") {
            lintFile(file, function(hasError) {
                if (hasError) {
                    errorCount++;
                }
                innercallback();
            });
        } else {
            innercallback();
        }
    } else {
        processFiles(file);
    }
});
}, function(error) {
    if (errorCount > 0) {
        process.exit(1);
    }
});
});
}

function lintFile(file, callback) {
    console.log("Linting " + file);
    fs.readFile(file, function(err, data) {
        if (err) {
            console.log('Error: ' + err);
            return;
        }
        if (jshint(data.toString())) {
            console.log('File ' + file + ' has no errors.');
```

```

            console.log('-----');
```

```

            callback(false);
        } else {
            console.error('Errors in file ' + file);
            var out = jshint.data(),
                errors = out.errors;
            for (var j = 0; j < errors.length; j++) {
                console.error(errors[j].line + ':' + errors[j].character + ' -> ' + errors[j].reason +
                    ' -> ' + errors[j].evidence);
            }
            console.error('-----');
```

```

            setTimeout(function() {
                callback(true);
            }, 10);
        }
    });
}

```

Lea Ganchos CLI iónicos en línea: <https://riptutorial.com/es/ionic-framework/topic/6520/ganchos-cli-ionicos>

---

# Capítulo 14: Ionic - Analice su aplicación con jshint y gulp-jshint como parte de su proceso de construcción

## Observaciones

El tintado de tu aplicación iónica antes de correr tiene enormes ventajas. Analizará el código en busca de posibles errores y le ahorrará una tremenda cantidad de tiempo.

## ¿Qué es linting y cómo instalar los paquetes requeridos?

"Linting es el proceso de ejecutar un programa que analizará el código en busca de posibles errores". - ver ¿Qué es "Linting"?

Su aplicación iónica viene con un archivo package.json. Vaya a la raíz de su aplicación en una línea de comandos / terminal e instale los siguientes paquetes:

```
npm install jshint --save-dev
npm install jshint-stylish --save-dev
npm install gulp-jshint --save-dev
```

## Examples

### Añadir una tarea trilla

En la raíz de su aplicación iónica, hay un archivo gulpfile.js. Ábralo en un editor y pegue la siguiente tarea:

```
gulp.task('lint', function() {
  return gulp.src(['./www/js/**/*.js'])
    .pipe(jshint('.jshintrc'))
    .pipe(jshint.reporter('jshint-stylish'))
    .pipe(jshint.reporter('fail'))
});
```

Esto busca una carpeta llamada 'js' dentro de la carpeta 'www'. Si tiene otras carpetas que contienen archivos JavaScript, agréguelas también. Por ejemplo, también permite agregar una carpeta llamada 'vistas':

```
gulp.task('lint', function() {
  return gulp.src(['./www/js/**/*.js', './www/views/**/*.js'])
    .pipe(jshint('.jshintrc'))
    .pipe(jshint.reporter('jshint-stylish'))
    .pipe(jshint.reporter('fail'))
});
```

## Explicaciones:

- 1) `/**/*.js` - This syntax means to look at all the js files in the subfolders too
- 2) `.jshintrc` - This is a configuration file that we will create in the next example.

## Crear archivo `.jshintrc` (opcional)

Cree un archivo llamado `.jshintrc` en la raíz de su aplicación, donde `package.json` es.

\* Nota en Windows: crea un archivo llamado `"jshintrc.txt"`. Luego renómbrelo a `".jshintrc"`. (note el punto al final).

Este es un archivo de configuración. Por ejemplo, puede decirle a jshint que ignore ciertas variables y muchas otras cosas. Aquí está el mio:

```
{
  "predef": [
    "window",
    "console",
    "cordova",
    "device",
    "alert",
    "document",
    "debug",
    "setServiceVars",
    "StatusBar",
    "config"
  ],
  "globals": {
    "angular"      : false,
    "myApp"        : false,
    "myControllers" : false,
    "myDirectives" : false,
    "localStorage" : false,
    "navigator"    : false,
    "emit"         : false,
    "atob"         : false,
    "moment"       : false,
    "btoa"         : false
  },
  "node"          : true
}
```

## Añadir Makefile

1. Cree un archivo llamado: `"Makefile"` (sin extensión) en la raíz de su aplicación
2. Ábralo en un editor de texto y agregue esto:

```
android:
  gulp lint
  gulp sass
  ionic run android --device

ios:
```

```
gulp lint
gulp sass
ionic build ios
```

Esto borrará tu aplicación y, si eso pasa, compilará Sass y construirá tu aplicación.

**Uso:** para ejecutar su aplicación, en lugar de usar el "dispositivo iónico android --dispositivo", ejecute estos comandos:

```
Android: make android
iOS      : make ios
```

Lea Ionic - Analice su aplicación con jshint y gulp-jshint como parte de su proceso de construcción en línea: <https://riptutorial.com/es/ionic-framework/topic/4889/ionic---analice-su-aplicacion-con-jshint-y-gulp-jshint-como-parte-de-su-proceso-de-construccion>

---

# Capítulo 15: Ionicones

## Observaciones

En el desarrollo web moderno, es una práctica común usar fuentes para mostrar iconos. Dado que las fuentes son vectores, son independientes de la resolución y se pueden colorear fácilmente a través de CSS, solo para mencionar algunas ventajas en comparación con las imágenes de mapa de bits, etc. Ionicones fue creado por el mismo equipo con el que se creó Ionic Framework y puede usarse en cualquier proyecto desde Son 100% gratuitos y de código abierto. MIT con licencia.

Los ionicones se pueden usar solos o con componentes CSS de Ionic cuando tienen ciertos estilos de acuerdo con los elementos principales.

La página de inicio y la lista de los iconos se pueden encontrar aquí: <http://ionicones.com/>

## Examples

### Uso básico

Los íconos de fuentes generalmente se colocan dentro de una etiqueta `<i>`. Ionic tiene estilos css predeterminados para los iconos para un uso fácil. El ejemplo más básico de uso:

```
<i class="icon ion-home"></i>
```

### Uso extendido

Ionic tiene algunos componentes CSS donde puedes usar Ionicones como predeterminado que tiene un estilo preestablecido. La clase de `range` en el elemento `<div>` aplicará el estilo correcto tanto a la entrada como a los íconos dentro de ella. Aquí hay un ejemplo de un control deslizante de rango.

```
<div class="item range">
  <i class="icon ion-volume-low"></i>
  <input type="range" name="volume">
  <i class="icon ion-volume-high"></i>
</div>
```

Otro ejemplo de uso de Ionicone en pestañas iónicas que creará una pestaña como menú. Las clases `tabs-striped` `tabs-color-assertive` definen el estilo de las pestañas en sí. Los iconos se usan con etiquetas `<i>` simples y obtienen su estilo posicional de las clases aplicadas a los divs principales.

```
<div class="tabs-striped tabs-color-assertive">
  <div class="tabs">
    <a class="tab-item" href="#">
```

```
<i class="icon ion-home"></i>
Home
</a>
<a class="tab-item" href="#">
  <i class="icon ion-gear-b"></i>
  Settings
</a>
</div>
```

Lea Ionicos en línea: <https://riptutorial.com/es/ionic-framework/topic/5886/ionicos>

---

# Capítulo 16: Nube Jónica para Proyectos Iónicos Yeoman

## Examples

Plataforma Iónica (Ionic Cloud) para Proyectos Iónicos de Yo (Yeoman)



**Plataforma jónica :**

**Construye, empuja, implementa y escala tus aplicaciones iónicas de la manera más fácil.**

### **Descripción del Título:**

Ionic Platform es una plataforma en la nube para administrar y escalar aplicaciones móviles multiplataforma. Los servicios integrados le permiten a usted y a su equipo crear, implementar y hacer crecer sus aplicaciones de manera eficiente.

### **Objetivo del documento:**

Ionic Platform funciona bien con los proyectos estándar de Ionic. Pero los proyectos que siguen cualquier estructura de directorio no estándar pueden enfrentar algunos obstáculos. Estos documentos proporcionan los pasos para utilizar la Plataforma iónica en los proyectos iónicos creados con Yeoman.

### **Alcance del documento:**

Este documento cubre los pasos básicos para crear un proyecto Ionic utilizando Yeoman e integrarlo con Ionic Platform utilizando el Cliente Web de Ionic Platform. Este documento cubre los pasos básicos para utilizar Ionic Deploy, Ionic Analytics y Ionic Push.

### **Público objetivo:**

Los destinatarios de este documento son los desarrolladores de aplicaciones web / móviles, con experiencia tanto de nivel principiante como experto, que están familiarizados con los requisitos previos que se detallan a continuación.

### **Requisitos previos:**

Debe estar familiarizado con los siguientes marcos / herramientas antes de intentar este documento.

- AngularJs: <https://docs.angularjs.org/guide>

- IonicFramework: <http://ionicframework.com/docs/guide>
- Yeoman: <http://yeoman.io/codelab/index.html>
- Generador iónico: <https://github.com/diegonetto/generator-ionic>
- Plataforma iónica: <https://ionic.io/platform>

---

## Generador Ionic Framework



### Un generador para el Ionic Framework de Yeoman, la herramienta de andamiaje de la web para aplicaciones web modernas.

Node.js es un tiempo de ejecución de JavaScript creado en el motor de JavaScript V8 de Chrome. npm es el gestor de paquetes para JavaScript. Descargue e instale Node (y npm) desde <http://nodejs.org>

```
$ npm install npm -g
$ npm install -g yo
```

Yeoman lo ayuda a impulsar nuevos proyectos, prescribiendo las mejores prácticas y herramientas para ayudarlo a mantenerse productivo.

```
$ yo ionic [app-name]
```

En *package.json* incluye lo siguiente en devDependencies

```
"grunt-string-replace": "^1.2.1"
```

En *bower.json* incluyen lo siguiente en dependencias.

```
"ionic-platform-web-client": "^0.7.1"
```

En *Gruntfile.js*, cambie la carpeta de *scripts* a *'js'*. Cambie en *index.html* también si es necesario.

```
grunt.initConfig({  yeoman: {.....
  scripts: 'js',
  ..... } })
```

Entonces corre

```
$ bower install && npm install
$ grunt
```

```
$ grunt serve

$ cordova platform add android
$ grunt build:android --debug
```

## plataforma-iónica-cliente-web



### Un cliente web que proporciona interacciones con la plataforma Ionic.

Necesitamos algún código para permitir que su aplicación hable con la plataforma iónica. Necesitamos agregar el cliente web de la plataforma Ionic para que la aplicación Ionic interactúe con los complementos y la plataforma Ionic.io.

```
$ ionic io init
```

En su *app.js* agregue la dependencia del módulo *'ionic.service.core'*. En *Gruntfile.js* agregue la tarea grunt *'ionicSettings'* como se indica a continuación.

```
grunt.initConfig({
  ionicSettings: JSON.stringify(grunt.file.readJSON('./.io-config.json')),

  ionicIoBundlePath: 'www/bower_components/ionic-platform-web-
  client/dist/ionic.io.bundle.min.js',

  'string-replace': {
    ionicSettings: {
      files: {
        '<%= ionicIoBundlePath %>': '<%= ionicIoBundlePath %>',
      },
      options: {
        replacements: [
          {
            pattern:
              "IONIC_SETTINGS_STRING_START";"IONIC_SETTINGS_STRING_END"',
            replacement:
              "IONIC_SETTINGS_STRING_START";var settings =<%= ionicSettings %>; return { get:
function(setting) { if (settings[setting]) { return settings[setting]; } return null; }
};"IONIC_SETTINGS_STRING_END";'
          }
        ]
      }
    }
  },

  copy: {
    ionicPlatform:{
      expand: true,
      cwd: 'app/bower_components/ionic-platform-web-client/dist/',
      src: ['**'],
```

```
        dest: 'www/bower_components/ionic-platform-web-client/dist'
      }
    }
  });

  grunt.registerTask('ionicSettings', ['copy:ionicPlatform','string-replace:ionicSettings']);
```

Agregue el *'ionicSettings'* en *init* y *comprima las tareas* después de la *copia* . En *index.html*, mueva la siguiente etiqueta después de todas las declaraciones de etiquetas.

```
<script src="bower_components/ionic-platform-web-client/dist/ionic.io.bundle.min.js"></script>
```

Entonces corre

```
$ Grunt serve
```

## Despliegue jónico



### Inserte actualizaciones en tiempo real de sus aplicaciones de producción y administre el historial de versiones.

Ionic Deploy le permite actualizar su aplicación a pedido, para cualquier cambio que no requiera modificaciones binarias, ahorrando días o incluso semanas de tiempo de espera. Siga el procedimiento a continuación para configurar Ionic Deploy para su aplicación.

En *Gruntfile.js* agregue la tarea grunt *'desplegar'* como se indica a continuación.

```
grunt.registerTask('deploy', function () {
  return grunt.task.run(['init', 'ionic:upload' + this.args.join()]);
});
```

entonces corre

```
$ ionic plugin add ionic-plugin-deploy
```

Código de implementación iónica:

```
var deploy = new Ionic.Deploy();

// Check Ionic Deploy for new code
deploy.check().then(function(hasUpdate) {
}, function(err) {
});
```

```
// Update app code with new release from Ionic Deploy
deploy.update().then(function(result) {
}, function(error) {
}, function(progress) {
});
```

Implementando actualizaciones:

Enviar nuevo código para su aplicación.

Crea un apk e instala tu aplicación. Realice algunos cambios en su código e implemente los cambios utilizando ' *grunt deploy* '. Luego actualízalo desde tu aplicación.

También puede implementar desde el tablero de instrumentos *apps.ionic.io*. Puede implementar la aplicación sin el parámetro de despliegue. Luego, en el tablero de instrumentos, puede agregar los metadatos y los detalles de la versión y desplegar la aplicación desde allí.

```
$ grunt build:android --debug

$ grunt deploy --note "release notes"
$ grunt deploy --note "release notes" --deploy=production
```

---

## Analítica Iónica



### Ver la transmisión en vivo de eventos o el número de eventos / usuarios en bruto / único a lo largo del tiempo.

¿Cuántos usuarios hay en tu aplicación en este momento? ¿Cuántos de ellos usarán tu aplicación mañana o la próxima semana? Sin información, no tiene forma de saber si su aplicación se está utilizando de la manera que usted espera. Siga el procedimiento a continuación para configurar Ionic Analytics para su aplicación.

En su *app.js*, agregue la dependencia del módulo ' *ionic.service.analytics* ' después de *ionic.service.core* Ejecute el método de registro analítico en la función de ejecución de nuestro módulo.

```
$ionicAnalytics.register();
```

En Ionic Analytics, cada acción rastreada que un usuario realiza en su aplicación está representada por un objeto de evento. Un evento es una acción única realizada en un punto específico en el tiempo. Para rastrear sus propios eventos, llame a

`$ionicAnalytics.track(eventType, eventData)` siempre que ocurra una acción.

```
$ionicAnalytics.track('User Login', {
  user: $scope.user
});
```

La directiva *ion-track-tap* envía un evento cuando se *toca* su elemento host. La directiva de *datos de seguimiento de iones* asociada adjunta datos de eventos.

```
<button ion-track-tap="eventType" ion-track-data="expression"></button>
```

En el *panel de control* de *apps.ionic.io* puede ver los siguientes datos de análisis,

**Eventos:** vea la cantidad bruta de eventos a lo largo del tiempo, o la cantidad de usuarios únicos que completaron un evento. Un evento puede ser cualquier cosa, desde un usuario que carga la aplicación, hasta confirmar una compra.

**Embudos:** Un embudo es una secuencia de acciones que espera que los usuarios realicen en su aplicación, lo que lleva a un objetivo definido. El uso cuidadoso de los embudos le ayudará a mejorar las tasas de conversión.

**Segmentos:** vea eventos a lo largo del tiempo, agrupados por una propiedad específica. O bien, calcule el porcentaje de eventos que coinciden con una propiedad determinada. Los segmentos le ayudan a comprender su base de usuarios y ver cómo cambian las propiedades con el tiempo.

**Retención:** realice un seguimiento del tiempo que los usuarios están activos en su aplicación antes de que dejen de usarla. O bien, identifique cuánto tiempo tardan los usuarios en alcanzar un objetivo definido, como una venta completa.

**Pulse:** una transmisión en vivo de eventos que llegan de sus usuarios.

---

## Empuje ionic



## Envíe notificaciones push dirigidas y automatizadas a sus usuarios.

Ionic Push le permite crear notificaciones push dirigidas a través de un panel simple que se enviará automáticamente cuando los usuarios cumplan con criterios específicos, y ofrece una API simple para enviar notificaciones push desde sus propios servidores.

### Perfiles Push de Android:

Las notificaciones push de Android utilizan el servicio *Google Cloud Messaging (GCM)*. Abra la [Consola de desarrolladores de Google](#) y cree un proyecto. Anote su *número de proyecto*. Este será el *ID de remitente de GCM* o el **Número de proyecto de GCM**.

En la sección *Administrador de API* , habilite la *API de mensajería en la nube de Google* . Luego navegue a la sección *Credenciales* y seleccione *Crear credenciales*, luego elija *Clave de API*, luego *Clave de servidor*. Asigne un nombre a su clave API y deje en blanco el campo *Aceptar solicitudes de ...* y haga clic en *Crear* . Guarde su **clave API**!

### Autenticación:

Vaya al panel de control de su aplicación en la [plataforma iónica](#) y navegue a *Configuración -> Certificados* . Si aún no lo ha hecho, cree un nuevo perfil de seguridad, luego presione *editar* . Anote la **etiqueta de perfil** .

Ahora, haga clic en la pestaña *Android* y busque la sección marcada *Google Cloud Messaging* , ingrese la *Clave API* que generó en la Consola de desarrollador de Google, luego haga clic en *Guardar* . Vaya a *Configuración -> Claves API* . Bajo *API Tokens* , cree un nuevo token y cópielo. Esto será su **token de API** .

```
$ ionic plugin add phonegap-plugin-push --variable SENDER_ID="GCM_PROJECT_NUMBER"
$ ionic config set gcm_key <your-gcm-project-number>
$ ionic config set dev_push false
$ ionic io init
```

Nota: phonegap-plugin-push requiere el repositorio de soporte de Android versión 32+

En su *app.js* agregue la dependencia del módulo ' *ionic.service.push* ' después de *ionic.service.core*

Código de empuje iónico:

Inicialice el servicio y registre su dispositivo en la función de ejecución de su módulo. Necesitará el token del dispositivo que está registrado por el usuario para enviar una notificación al usuario.

```
$ionicPush.init({
  debug: true,
  onNotification: function (notification) {
    console.log('token:', notification.payload);
  },
  onRegister: function (token) {
    console.log('Device Token:', token);
    $ionicPush.saveToken(token); // persist the token in the Ionic Platform
  }
});

$ionicPush.register();
```

entonces corre

```
$ grunt build:android --debug
```

Ionic Push le permite crear notificaciones push dirigidas a través del panel de control. También puede enviar notificaciones desde el servidor en el siguiente formato.

```
curl -X POST -H "Authorization: Bearer API_TOKEN" -H "Content-Type: application/json" -d '{
  "tokens": ["DEVICE_TOKEN"],
  "profile": "PROFILE_TAG",
  "notification": {
    "message": "Hello World!"
  }
  "android": {
    "title": "Hi User",
    "message": "An update is available for your App",
    "payload": {
      "update": true
    }
  }
}' "https://api.ionic.io/push/notifications"
```

Nota: los pasos para configurar Ionic Push para iOS son los mismos, excepto para crear los perfiles Push. Para crear perfiles push de iOS, consulte <http://docs.ionic.io/v2.0.0-beta/docs/ios-push-profiles>

---

## Aplicación de ejemplo



[Descarga la aplicación de muestra aquí](#) .

Una aplicación de ejemplo se adjunta aquí para referencia.

```
IonicApp:
|
|  bower.json
|  Gruntfile.js
|  package.json
|
└── app
    |  index.html
    |
    ├── js
    |   app.js
    |   controllers.js
    |
    └── templates
        home.html
        menu.html
```

Nota: Este no es un proyecto independiente. El código dado es solo para comparación con un proyecto creado e implementado utilizando los procedimientos descritos anteriormente en este documento, en caso de problemas o errores.

Lea Nube Jónica para Proyectos Iónicos Yeoman en línea: <https://riptutorial.com/es/ionic-framework/topic/6389/nube-jonica-para-proyectos-ionicos-yeoman>

# Capítulo 17: Probando la aplicación jónica en un navegador

## Observaciones

Las pruebas de funciones de dispositivos nativos como Cámara, Vibración y otras, muchas de las cuales se encuentran en la documentación de [Ionic Native](#), no se pueden realizar en el navegador. Esta es una limitación inherente al hecho de que Cordova, la plataforma de la que depende Ionic para poder acceder a las API nativas de Android, iOS y Windows Mobile de un dispositivo, no puede ejecutarse en el navegador.

Se puede solucionar este problema simulando la funcionalidad del complemento nativo.

## Ejemplo

Aquí hay un ejemplo de cómo simular el complemento de la [Camera](#):

Continúe y cree una carpeta opcional en la carpeta raíz de su proyecto.

```
cd src
mkdir mocks
cd mocks
touch camera-mock.ts
```

Abra `camera-mock.ts` y copie y pegue el siguiente código:

```
export class CameraMock {
  getPicture(params) {
    return new Promise((resolve, reject) => {
      resolve("BASE_64_IMAGE_DATA");
    });
  }
}
```

A continuación, abra `src/app.module.ts` e importe la clase simulada "

```
import { CameraMock } from "../mocks/camera-mock";
```

Luego agréguelo a la matriz de proveedores de módulos:

```
@NgModule({
  declarations: [
    MyApp,
    HomePage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp)
  ]
})
```

```

],
bootstrap: [IonicApp],
entryComponents: [
  MyApp,
  HomePage
],
providers: [
  StatusBar,
  SplashScreen,
  CameraMock,
  {provide: ErrorHandler, useClass: IonicErrorHandler}
]
})
export class AppModule {}

```

Ahora puedes usarlo en cualquier componente después de importarlo.

## Examples

### Pruebas en un navegador

Utilice el `ionic serve` para iniciar un servidor de desarrollo local para el desarrollo y la prueba de aplicaciones. Esto es útil tanto para la prueba del navegador de escritorio como para la prueba dentro de un navegador de dispositivo que está conectado a la misma red. Además, este comando inicia LiveReload, que se utiliza para monitorear los cambios en el sistema de archivos. Tan pronto como guarde un archivo, el navegador se actualizará automáticamente. Eche un vistazo a los [documentos de Sass](#) si también desea que el `ionic serve` vea los archivos de Sass del proyecto.

```
$ ionic serve [options]
```

Para el navegador Chrome puede inspeccionar dispositivos (AVD o móviles), escriba el siguiente comando en la barra de direcciones del navegador Chrome.

```
chrome://inspect/#devices
```

### LiveReload

Por defecto, LiveReload observará los cambios en su directorio `www/`, excluyendo `www/lib/`. Para cambiar esto, puede especificar una propiedad `watchPatterns` en el archivo `ionic.project` que se encuentra en la raíz de su proyecto para ver (o no ver) cambios específicos.

```

{
  "name": "myApp",
  "app_id": "",
  "watchPatterns": [
    "www/js/*",
    "!www/css/**/*"
  ]
}

```

Para obtener una referencia sobre la sintaxis de patrón [global](#) , revise los [patrones globales](#) en el sitio web Grunt.

Nota:

```
$ ionic setup sass
```

agregará una `watchPatterns` con los valores predeterminados a su archivo `ionic.project` que luego puede editar, además de la propiedad `gulpStartupTasks` como se describe en la [documentación de Sass](#) .

## Laboratorio Iónico

[Ionic Lab](http://ionicframework.com/img/blog/lab.png) <http://ionicframework.com/img/blog/lab.png>

Ionic Lab es una característica del `ionic serve` que facilita la prueba de su aplicación en el marco de un teléfono y con las plataformas iOS y Android una al lado de la otra. Para usarlo, solo ejecuta

```
$ ionic serve --lab
```

Lea el [anuncio de la versión completa](#) para todos los detalles!

## Especificando una dirección IP para usar

Diga que desea especificar a qué dirección se conectará su navegador, por ejemplo, para pruebas o usuarios externos. Especifique la dirección con el argumento `--address` .

```
$ ionic serve --address 68.54.96.105
```

## Proxies de servicio

El comando `serve` puede agregar algunos proxies al servidor http. Estos proxies son útiles si está desarrollando en el navegador y necesita hacer llamadas a una API externa. Con esta función, puede enviar una solicitud de proxy a la api externa a través del servidor http iónico, evitando que el No 'Access-Control-Allow-Origin' header is present on the requested resource error de No 'Access-Control-Allow-Origin' header is present on the requested resource .

En el archivo `ionic.project` puede agregar una propiedad con una matriz de proxies que desee agregar. Los proxies son objeto con dos propiedades:

- `path` : cadena que se comparará con el comienzo de la URL de solicitud entrante.
- `proxyUrl` : una cadena con la URL de donde debe ir la solicitud de proxy.

```
{  
  "name": "appname",  
  "email": "",  
  "app_id": "",
```

```
"proxies": [
  {
    "path": "/v1",
    "proxyUrl": "https://api.instagram.com/v1"
  }
]
```

Usando la configuración anterior, ahora puede hacer solicitudes a su servidor local en

`http://localhost:8100/v1` para que las solicitudes de proxy sean `https://api.instagram.com/v1`

Por ejemplo:

```
angular.module('starter.controllers', [])
.constant('InstagramApiUrl', '')
// .constant('InstagramApiUrl', 'https://api.instagram.com')
//In production, make this the real URL

.controller('FeedCtrl', function($scope, $http, InstagramApiUrl) {

  $scope.feed = null;

  $http.get(InstagramApiUrl +
'/v1/media/search?client_id=1&lat=48&lng=2.294351').then(function(data) {
  console.log('data ' , data)
  $scope.feed = data;
})

})
```

Vea también [esta esencia](#) para más ayuda.

## Indicadores de línea de comando / opciones

```
[--consolelogs|-c] ..... Print app console logs to Ionic CLI
[--serverlogs|-s] ..... Print dev server logs to Ionic CLI
[--port|-p] ..... Dev server HTTP port (8100 default)
[--livereload-port|-i] .. Live Reload port (35729 default)
[--nobrowser|-b] ..... Disable launching a browser
[--nolivereload|-r] ..... Do not start live reload
[--noproxy|-x] ..... Do not add proxies
```

Lea [Probando la aplicación jónica en un navegador en línea: https://riptutorial.com/es/ionic-framework/topic/3256/probando-la-aplicacion-jonica-en-un-navegador](https://riptutorial.com/es/ionic-framework/topic/3256/probando-la-aplicacion-jonica-en-un-navegador)

---

# Capítulo 18: Publicando tu aplicación jónica

## Examples

### Versión de lanzamiento del edificio desde macOS

#### Construye el APK

En primer lugar tenemos que construir el APK.

```
ionic build --release android
```

#### Generar clave privada

Luego crearemos un almacén de claves para firmar el APK. `keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000`

- Cambia **mi clave de liberación** con tu nombre de clave.
- Cambie **alias\_name** con su alias clave.

#### Firma el APK

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore HelloWorld-release-unsigned.apk alias_name
```

- Cambia **mi clave de liberación** con tu nombre de clave.
- Cambia **HelloWorld-release-unsigned** con tu apk sin firmar. `ionic-project/platforms/android/build/outputs/apk .`
- Cambie **alias\_name** con su alias clave.

#### Zip el APK

```
zipalign -v 4 HelloWorld-release-unsigned.apk HelloWorld.apk
```

- Puede encontrar `zipalign` en `/Users/username/Library/Android/sdk/build-tools/XXX/`
- Cambia **HelloWorld-release-unsigned** con tu apk sin firmar. `ionic-project/platforms/android/build/outputs/apk .`
- Cambie **HelloWorld.apk** con su nombre de archivo apk preferido. Esto será subido a Google Play.

Lea **Publicando tu aplicación jónica en línea**: <https://riptutorial.com/es/ionic-framework/topic/7755/publicando-tu-aplicacion-jonica>

---

# Capítulo 19: Servicios Backonic Iónicos (ionic.io)

## Examples

### Introducción y configuración

La [plataforma iónica](#) ofrece una gama de servicios y herramientas de back-end para dispositivos móviles centrados en el híbrido para que sea fácil de escalar aplicaciones híbridas hermosas y de alto rendimiento, a un ritmo rápido.

Para utilizar la Plataforma Ionic necesita tener [instalado](#) el [Ionic Framework](#) .

#### 1.) Registro ( [inscripción](#) )

Debe ingresar: nombre, nombre de usuario, compañía, teléfono (opcional), correo electrónico y contraseña. Después de recibir el correo electrónico de confirmación, puede [iniciar sesión](#) .

Bienvenido a Ionic. ¡Vamos a crear tu primera aplicación! Cree una nueva aplicación arriba, luego siga la guía de [inicio rápido](#) para cargar su primera aplicación.

#### 2.) Configuración

En primer lugar, [crea tu primera aplicación de Ionic](#) . Ahora puedes agregar el cliente web de la plataforma:

El cliente web le permite interactuar con los servicios de Ionic Platform desde el código de su aplicación.

```
$ ionic add ionic-platform-web-client
```

Ahora, necesita la Plataforma para asignar a su aplicación un ID de aplicación único y una clave api. Para hacer eso usa el comando `ionic io init` .

```
$ ionic io init
```

Esto le pedirá automáticamente que inicie sesión en su cuenta de plataforma. La identificación de la aplicación y la clave api se almacenarán en la [configuración de la](#) plataforma iónica de su proyecto.

Su aplicación ahora está conectada a la plataforma iónica y se incluirá en su [Tablero](#) .

#### 3.) Ahora, puede continuar con la instalación de uno de los servicios de Ionic Platform :

- [Usuarios / Auth](#)
- [Desplegar](#)

- [empujar](#)
- [Paquete](#)
- [Analítica](#)

Lea [Servicios Backonic Iónicos \(ionic.io\)](#) en línea: <https://riptutorial.com/es/ionic-framework/topic/3622/servicios-backonic-ionic-ionic-io->

# Creditos

S. No	Capítulos	Contributors
1	Comenzando con el marco iónico	<a href="#">Akshay Khale</a> , <a href="#">Andrea Macchieraldo</a> , <a href="#">Community</a> , <a href="#">Devid Farinelli</a> , <a href="#">Ian Pinto</a> , <a href="#">leetheguy</a> , <a href="#">Lightbeard</a> , <a href="#">Mazz</a> , <a href="#">Newton Joshua</a> , <a href="#">the_mahasagar</a>
2	¿Cómo utilizar las características de EcmaScript 6 en Ionic?	<a href="#">Nikola</a>
3	¿Cuál es la diferencia entre "acumulación iónica" y "preparación iónica"?	<a href="#">Nikola</a>
4	Comenzar a crear aplicaciones en Ionic	<a href="#">A-Droid Tech</a> , <a href="#">Ian Pinto</a>
5	Componentes jónicos de CSS	<a href="#">thepio</a>
6	Conectando Ionic con cualquier base de datos	<a href="#">Nikola</a>
7	Crear diálogo en jónico	<a href="#">A-Droid Tech</a>
8	Desplazamiento infinito iónico para mostrar elementos de carga a pedido (datos ya disponibles que no se encuentran en la solicitud HTTP)	<a href="#">SANAT</a>
9	Desplegar Ionic como un sitio web	<a href="#">Nikola</a>
10	Dispositivo de cámara y acceso a la	<a href="#">Pritish</a> , <a href="#">sonu</a>

	biblioteca de fotos desde la aplicación Ionic	
11	Ejecute la aplicación Ionic en el emulador o en su teléfono	<a href="#">A-Droid Tech</a> , <a href="#">Gerard Cuadras</a> , <a href="#">Ian Pinto</a> , <a href="#">Ketan Akbari</a> , <a href="#">olivier</a> , <a href="#">Raymond Ativie</a>
12	Extensiones Ionic AngularJS	<a href="#">thepio</a>
13	Ganchos CLI iónicos	<a href="#">thepio</a>
14	Ionic - Analice su aplicación con jshint y gulp-jshint como parte de su proceso de construcción	<a href="#">Akilan Arasu</a> , <a href="#">Ian Pinto</a> , <a href="#">Sumama Waheed</a> , <a href="#">thepio</a>
15	Iconos	<a href="#">thepio</a>
16	Nube Jónica para Proyectos Iónicos Yeoman	<a href="#">Newton Joshua</a>
17	Probando la aplicación jónica en un navegador	<a href="#">A-Droid Tech</a> , <a href="#">Ian Pinto</a> , <a href="#">Ketan Akbari</a> , <a href="#">maninak</a>
18	Publicando tu aplicación jónica	<a href="#">Gerard Cuadras</a>
19	Servicios Backonic Iónicos (ionic.io)	<a href="#">Tomislav Stankovic</a>