



**EBook Gratis**

# APRENDIZAJE Cordova

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

**#cordova**

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Empezando con Cordova.....</b>	<b>2</b>
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Creando una aplicación.....	3
<b>Preliminar.....</b>	<b>3</b>
<b>Creando la aplicación.....</b>	<b>3</b>
<b>Añadiendo plataformas y plugins.....</b>	<b>3</b>
Plataformas.....	3
Complementos.....	4
<b>Ejecutando tu aplicación.....</b>	<b>4</b>
Instalar Cordova en Windows.....	4
Versiones.....	9
<b>Capítulo 2: Cómo detectar el estado de la conexión de red.....</b>	<b>10</b>
Examples.....	10
Usando el complemento de información de red de cordova.....	10
<b>Capítulo 3: Cómo instalar / desinstalar el complemento de cordova personalizado.....</b>	<b>12</b>
Examples.....	12
Utilice el comando plugman para instalar / desinstalar el complemento de cordova.....	12
<b>Capítulo 4: Cómo personalizar la plataforma de la carpeta www en cordova.....</b>	<b>13</b>
Examples.....	13
crear css / js específico para una plataforma (android / ios).....	13
<b>Capítulo 5: Complemento de imagen de la cosecha de Córdoba.....</b>	<b>15</b>
Examples.....	15
Recorte la imagen después de hacer clic con la cámara o seleccionar la imagen.....	15
<b>Capítulo 6: Complementos de Cordova: cómo instalar, cómo funcionan, ejemplos.....</b>	<b>17</b>
Examples.....	17
¿Qué son los complementos de Córdoba?.....	17

¿Cómo pueden ser útiles los complementos de Cordova? .....	17
Instalar el complemento de Cordova .....	17
Complementos más populares .....	17
<b>Capítulo 7: Cordova ios construir .....</b>	<b>19</b>
Introducción .....	19
Observaciones .....	19
Examples .....	19
cordova HelloWorld Project .....	19
<b>Capítulo 8: Creando tu primera aplicación con Cordova .....</b>	<b>22</b>
Observaciones .....	22
Examples .....	22
Usando la herramienta de línea de comandos .....	22
<b>Capítulo 9: Depurando la aplicación .....</b>	<b>24</b>
Observaciones .....	24
Examples .....	24
Depurar en dispositivo Android utilizando USB .....	24
Depurar aplicaciones de Cordova usando GapDebug .....	24
Depuración en dispositivo iOS utilizando USB .....	25
<b>Capítulo 10: Empezando con Cordova .....</b>	<b>27</b>
Introducción .....	27
Examples .....	27
Creación de la compilación de Android (.apk) .....	27
<b>Capítulo 11: Firebase Push Notification Cordova .....</b>	<b>30</b>
Examples .....	30
Firebase Push Notification en Cordova Android .....	30
Cordova Firebase Push Notification Plugin .....	30
<b>Capítulo 12: Firma la compilación de Android con Cordova 5 .....</b>	<b>33</b>
Examples .....	33
Agregue la configuración de compilación para firmar el archivo .apk .....	33
<b>Capítulo 13: Google Analytics en Córdoba .....</b>	<b>35</b>
Examples .....	35

Google Analytics en Córdoba sin ningún plugin.....	35
<b>Capítulo 14: Hacer la aplicación liberada de Cordova CLI.....</b>	<b>37</b>
Examples.....	37
Androide.....	37
iOS.....	38
<b>Capítulo 15: Notificación Push en Android y iOS.....</b>	<b>39</b>
Examples.....	39
Usando el nuevo phonegap-plugin-push.....	39
<b>Capítulo 16: Visual Studio Tools para Apache Cordova.....</b>	<b>42</b>
Examples.....	42
Obtén Apache Cordova Tools en Visual Studio.....	42
Actualizar las herramientas de Apache Cordova en Visual Studio.....	42
<b>Creditos.....</b>	<b>43</b>

---

## Acerca de

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

It is an unofficial and free Cordova 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 Cordova.

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: Empezando con Cordova

## Observaciones

Apache Cordova se utiliza para crear aplicaciones móviles con HTML, CSS y JS.

Apache Cordova se dirige a múltiples plataformas con una base de código.

Apache Cordova es libre y de código abierto.

***Cordova envuelve su aplicación HTML / JavaScript en un contenedor nativo que puede acceder a las funciones del dispositivo de varias plataformas. Estas funciones están expuestas a través de una API de JavaScript unificada, lo que le permite escribir fácilmente un conjunto de códigos para dirigirse a casi todos los teléfonos o tabletas en el mercado hoy en día y publicar en sus tiendas de aplicaciones.***

¿Quién podría usar Apache Cordova?

1. un desarrollador móvil y desea extender una aplicación a más de una plataforma, sin tener que volver a implementarla con el conjunto de herramientas y lenguajes de cada plataforma.
2. un desarrollador web y desea implementar una aplicación web empaquetada para su distribución en varios portales de tiendas de aplicaciones.
3. un desarrollador móvil interesado en mezclar componentes de aplicaciones nativas con una WebView (ventana de navegador especial) que puede acceder a las API de nivel de dispositivo, o si desea desarrollar una interfaz de complemento entre los componentes nativos y los de WebView.

**Introducción a Cordova:** <https://cordova.apache.org/docs/en/latest/>

## Examples

### Instalación o configuración

Para instalar la herramienta de línea de comandos de **cordova** , siga estos pasos:

1. Descargue e instale [Node.js](#). En la instalación, debería poder invocar **node** y **npm** en su línea de comando.

- Para ver si Node está instalado, abra su CLI (interfaz de línea de comandos). Para Windows es el símbolo del sistema de Windows, para MAC es el terminal. Tipo:

```
$ node -v
```

Esto debería imprimir un número de versión, por lo que verá algo como esto v0.10.35. Si Node no está instalado, busque su sistema operativo y siga las instrucciones aquí:

<https://nodejs.org/en/download/package-manager/>

2. (Opcional) Descargue e instale un [cliente git](#) , si aún no tiene uno. Después de la instalación, deberías poder invocar **git** en tu línea de comando. El CLI lo utiliza para descargar activos cuando se hace referencia a ellos mediante una url a un repositorio git.
3. Instale el módulo **cordova** usando la utilidad **npm** de Node.js. El módulo **cordova** será descargado automáticamente por la utilidad **npm** .

en OS X y Linux:

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

en Windows:

```
C:\>npm install -g cordova
```

La bandera **-g** de arriba le dice a **npm** que instale **cordova** globalmente. De lo contrario, se instalará en el subdirectorio **node\_modules** del directorio de trabajo actual.

Después de la instalación, debería poder ejecutar **cordova** en la línea de comandos sin argumentos y debería imprimir el texto de ayuda.

## Creando una aplicación

---

# Preliminar

Instala las herramientas cli de Cordova, si aún no lo has hecho.

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

Navegue a la carpeta de trabajo deseada.

```
$ cd /path/to/coding/folder
```

---

# Creando la aplicación

Crear una nueva aplicación

```
$ cordova create <appProjectName> <appNameSpace> <appName>
```

Para este ejemplo, crearemos una aplicación 'HelloWorld':

```
$ cordova create helloWorld com.example.helloworld HelloWorld
```

---

# Añadiendo plataformas y plugins.

## Plataformas

En primer lugar, vaya a la carpeta de la aplicación.

```
$ cd <appName>
```

Añade las plataformas para las que deseas construir. La lista de plataformas soportadas se puede encontrar [aquí](#) .

```
$ cordova platform add <platformList>
```

Estaremos agregando la plataforma Android, iOS y navegador. Usa la separación de espacio para agregar múltiples plataformas a la vez. La plataforma del `browser` será útil para las pruebas en el navegador.

El uso del argumento `--save` guardará la lista de plataformas en el archivo `config.xml` de Cordova.

```
$ cordova platform add android ios browser --save
```

Se puede encontrar una lista extensa de opciones con respecto al comando de la `platform` en la [documentación de cordova](#) .

## Complementos

Los complementos de Cordova pueden brindarle acceso al hardware del dispositivo, funciones específicas del sistema operativo y muchas más funciones.

La estructura del comando del `plugin` es la misma que la de las plataformas.

```
$ cordova plugin add <plugins.value>
```

Agregaremos el complemento de archivo de cordova (para facilitar el acceso al almacenamiento del dispositivo) y el complemento de la cámara, que le da acceso a la cámara del dispositivo para hacer fotos y videos.

```
$ cordova plugin add cordova-plugin-file cordova-plugin-camera --save
```

**Recuerde** : al usar el argumento `--save` escribe sus configuraciones en el archivo `config.xml` . Muy útil para recrear fácilmente el proyecto en otra máquina.

Cordova tiene una excelente página de búsqueda de complementos configurada para su conveniencia. Puedes encontrarlo [aquí](#) .

---

## Ejecutando tu aplicación

Ejecutar la aplicación es bastante sencillo. Simplemente use el siguiente comando.

```
$ cordova run <platform name>
```

Para nuestro ejemplo, ejecutaremos nuestra aplicación de prueba en el navegador.

```
$ cordova run browser
```

Esto abre su navegador predeterminado con su aplicación lista para probar.

## Instalar Cordova en Windows

**Primero, instale el kit de desarrollo Java SE**

Esto puede ser tan simple como descargar, hacer doble clic en el archivo descargado y seguir las instrucciones de instalación. Para instalar Java SE Development Kit, descárguelo del sitio web oficial. [Kit de desarrollo de Java SE. Descargas](#)

Una vez completada la instalación de JDK, debe agregar una nueva variable de sistema `JAVA_HOME` con ruta a su JDK

Junto a la `PATH` sistema `PATH` , agregue la ruta al direccionador bin de JDK

Ahora puedes probar la instalación. Abra el símbolo del sistema y use el comando

```
javac -version
```

¡Si ves un número de versión lo hiciste todo bien!

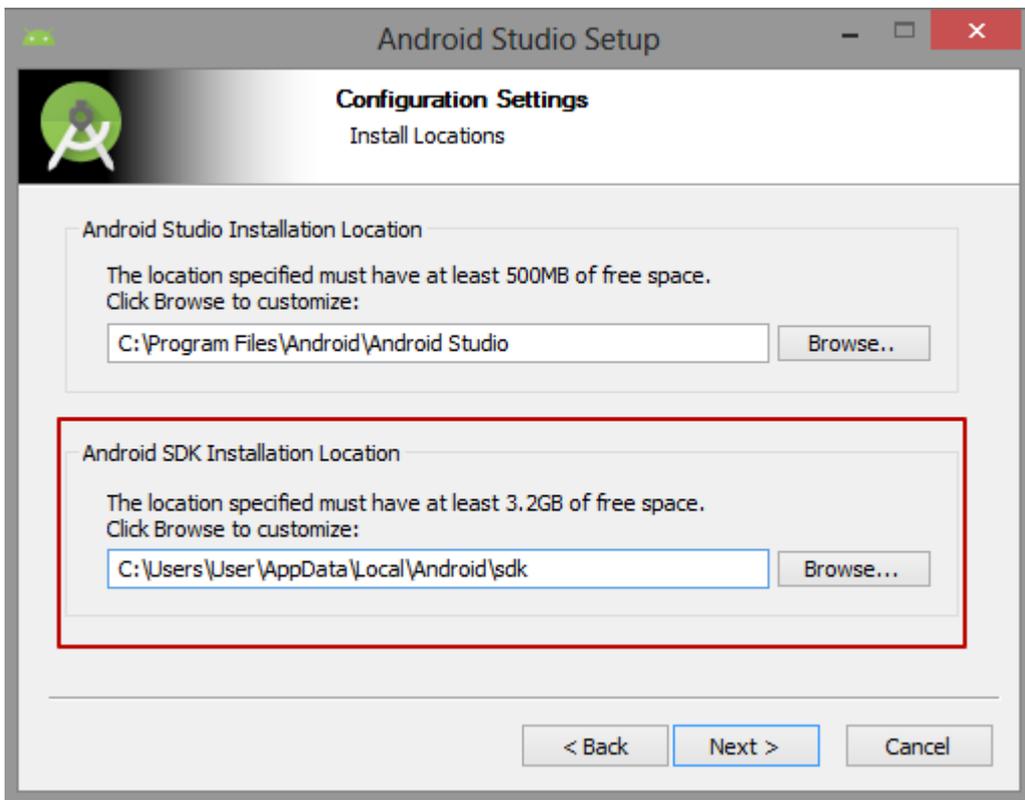
## Ahora instale Android SDK Tools con Android Studio

Recomendé instalar el Android Studio porque en este momento es la mejor manera de instalar rápida y fácilmente todas las cosas más necesarias para el desarrollo de Android. La lista de cosas incluye:

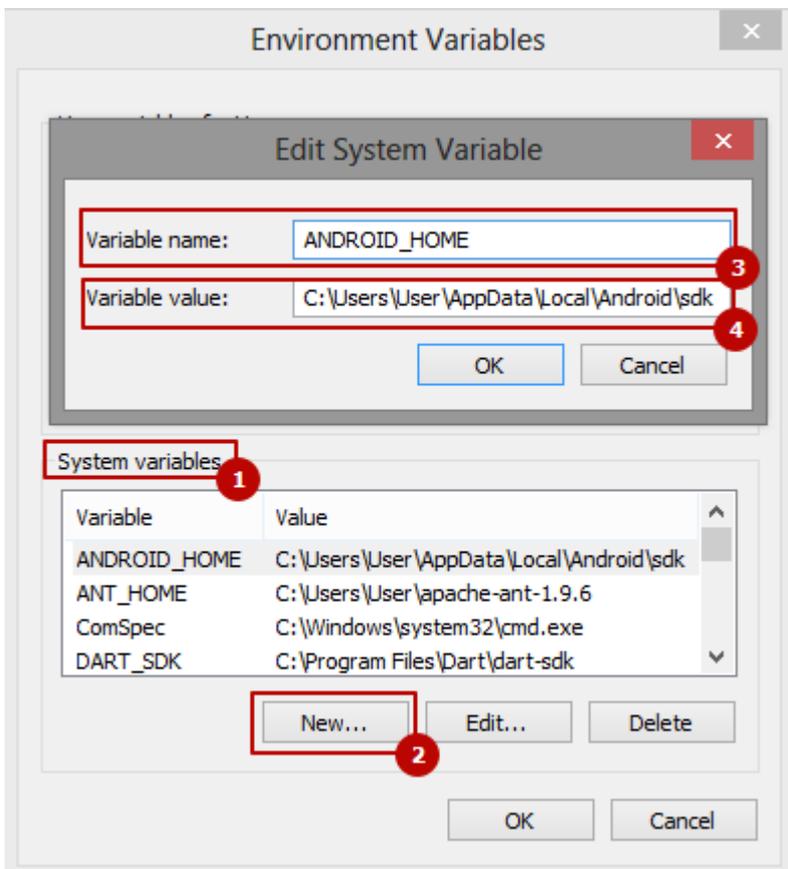
- Kit de desarrollo de Android (Android SDK, Android SDK Manager, Android SDK Platform-tools, Android SDK Build-tools)
- Emulador de Android con un gran número de configuraciones de Android.
- IDE (para el desarrollo de Android en Java)
- Gradle
- Sería muy útil si está aprendiendo Java, y en el futuro desea comenzar a desarrollar para Android en Java.

Por lo tanto, descargue Android Studio desde el sitio web oficial [developer.android.com](https://developer.android.com)

La instalación de Android Studio es muy simple y solo necesitas seguir las instrucciones. Pero debes tomar nota de la ubicación de instalación del SDK de Android

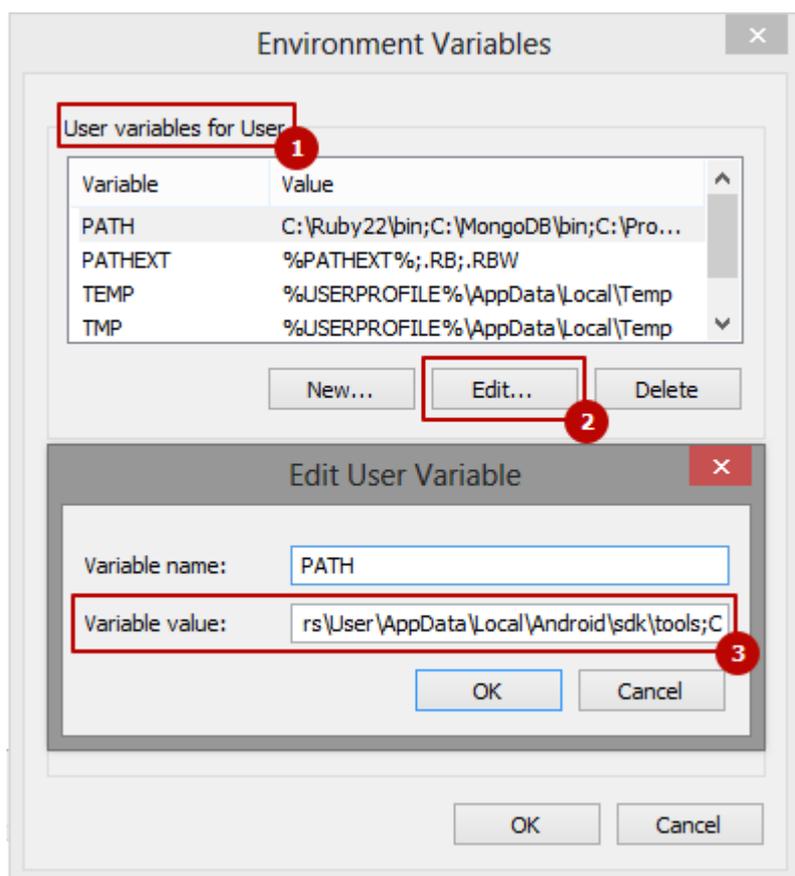


Después de que se complete la instalación de Android Studio, debe agregar una nueva variable de sistema `ANDROID_HOME` con la ruta a su `Android SDK`



Ahora debe agregar Android SDK y Android SDK Tools a PATH System Variable. En la lista Variables de usuario, seleccione RUTA y haga clic en el botón Editar. Al final del campo Valor de variable, agregue un punto y coma y siga las rutas:

```
C:\Users\User\AppData\Local\Android\sdk;C:\Users\User\AppData\Local\Android\sdk\tools;C:\Users\User\AppData\Local\Android\sdk\tools;
```



Ahora puedes probar la instalación. Abra el símbolo del sistema y use el comando

```
adb version
```

Esto debería mostrar la versión del Android Debug Bridge. ¡Si ves un número de versión lo hiciste todo bien!

Ahora abre de nuevo el símbolo del sistema y usa el comando

```
android
```

para abrir Android SDK Manager

Android SDK Manager

Packages Tools

SDK Path: C:\Users\User\AppData\Local\Android\sdk

Packages

 Name	API	Rev.	Status
<input type="checkbox"/>  Tools			
<input type="checkbox"/>  Android SDK Tools		24.4.1	 Installed
<input type="checkbox"/>  Android SDK Platform-tools		23.1	 Installed
<input type="checkbox"/>  Android SDK Build-tools		23.0.3	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		23.0.2	 Installed
<input type="checkbox"/>  Android SDK Build-tools		23.0.1	 Installed
<input type="checkbox"/>  Android SDK Build-tools		22.0.1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		21.1.2	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		20	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		19.1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Tools (Preview Channel)			
<input type="checkbox"/>  Android SDK Tools		25.0.1...	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Platform-tools		24 rc1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android SDK Build-tools		24 rc2	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android N (API 23, N preview)			
<input checked="" type="checkbox"/>  SDK Platform Android N Preview	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android TV Intel x86 Atom System Image	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Intel x86 Atom_64 System Image	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Intel x86 Atom System Image	N	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>  Android 6.0 (API 23)			
<input type="checkbox"/>  Documentation for Android SDK	23	1	 Installed
<input type="checkbox"/>  SDK Platform	23	2	 Installed
<input type="checkbox"/>  Samples for SDK	23	2	 Installed
<input checked="" type="checkbox"/>  Android TV ARM EABI v7a System Image	23	2	 Update available: rev. 3
<input checked="" type="checkbox"/>  Android TV Intel x86 Atom System Image	23	2	 Update available: rev. 3
<input type="checkbox"/>  Android Wear ARM EABI v7a System Image	23	2	 Installed
<input type="checkbox"/>  Android Wear Intel x86 Atom System Image	23	2	 Installed

Show:  Updates/New  Installed [Select New or Updates](#)

Obsolete [Deselect All](#)

Done loading packages.

En el Android SDK Manager seleccione para instalar

- Herramientas de Android SDK
- Plataforma Android SDK-herramientas
- Android SDK Build-Tools
- Android SDK Build-Tools
- Android 6.0 (API 23)
- Android 5.1.1 (API 22)
- Android 5.0.1 (API 21)
- Android 4.2.2 (API 17)
- Herramientas de depuración de GPU

- Repositorio de soporte de Android
- Biblioteca de soporte de Android
- Servicios de Google Play
- Repositorio de google
- Controlador USB de Google
- Acelerador de emulador Intel x86 (instalador HAXM)

y haga clic en el botón Instalar.

Nota:

[Cordova niveles de API compatibles con Android](#)

[Entender los niveles API de Android](#)

[Plataforma Android / Distribución de versiones API](#)

## Instalar cordova

Abra el símbolo del sistema e instale Cordova usando el comando

```
npm install -g cordova
```

## Versiones

Última versión de Córdoba:

Cordova 6.1.0 - <https://cordova.apache.org/news/2016/03/23/tools-release.html> Cordova 6.0.0 - <https://cordova.apache.org/news/2016/01/28/tools-release.html>

Última plataforma Android y plataforma de iOS

Cordova Android 5.2.2 - <https://cordova.apache.org/announcements/2016/07/02/android-5.2.0.html> Cordova iOS 4.2.1 - <https://cordova.apache.org/announcements/2016/07/11/cordova-android-5.2.1.html>

Lea [Empezando con Cordova en línea](#): <https://riptutorial.com/es/cordova/topic/884/empezando-con-cordova>

# Capítulo 2: Cómo detectar el estado de la conexión de red.

## Examples

### Usando el complemento de información de red de cordova

Detectar el estado actual de la conexión de red y responder a cualquier cambio que pueda ocurrir, puede hacerse usando uno de varios complementos. Este ejemplo es sobre el [complemento de información de red de cordova](#).

Agregue el plugin al proyecto:

```
cordova plugin add cordova-plugin-network-information
```

Después del [evento Devolución de Cordova](#), un objeto de conexión está disponible a través de `navigator.connection`. La propiedad `type` contiene el estado actual de la red:

```
document.addEventListener("deviceready", function() {
    var networkState = navigator.connection.type;
}, false);
```

`networkState` ahora contiene una de las siguientes constantes:

```
Connection.UNKNOWN // Unknown connection
Connection.ETHERNET // Ethernet connection
Connection.WIFI // WiFi connection
Connection.CELL_2G // Cell 2G connection
Connection.CELL_3G // Cell 3G connection
Connection.CELL_4G // Cell 4G connection
Connection.CELL // Cell generic connection
Connection.NONE // No network connection
```

Se puede detectar un cambio en la conexión de red conectando una función al evento en `online` o `offline`:

```
document.addEventListener("online", function() {
    // device went online
    var networkState = navigator.connection.type; // Get new network state
    ...
}, false);

document.addEventListener("offline", function() {
    // device went offline
    var networkState = navigator.connection.type; // Get new network state
    ...
}, false);
```

Lea [Cómo detectar el estado de la conexión de red.](https://riptutorial.com/es/cordova/topic/3615/como-detectar-el-estado-de-la-conexion-de-red-) en línea:

<https://riptutorial.com/es/cordova/topic/3615/como-detectar-el-estado-de-la-conexion-de-red->

---

# Capítulo 3: Cómo instalar / desinstalar el complemento de cordova personalizado

## Examples

Utilice el comando plugman para instalar / desinstalar el complemento de cordova

Puede usar el comando plugman para instalar / desinstalar complementos de cordova personalizados.

Instalar plugman

```
npm install -g plugman
```

Instalar la sintaxis del comando plugin:

```
plugman <install|uninstall> --platform <ios|android|blackberry10|wp8> --project <directory> --  
plugin <name|url|path>
```

Ejemplo:

```
plugman install --platform ios --project platforms/ios/ --plugin plugins/cordova-plugin-test/
```

Para más opciones de plugman revisa [este enlace](#).

Lea [Cómo instalar / desinstalar el complemento de cordova personalizado en línea](#):  
<https://riptutorial.com/es/cordova/topic/7506/como-instalar---desinstalar-el-complemento-de-cordova-personalizado>

---

# Capítulo 4: Cómo personalizar la plataforma de la carpeta www en cordova

## Examples

### crear css / js específico para una plataforma (android / ios)

Digamos que quieres crear un archivo css / js específico para una plataforma. Para que usted tiene que crear una carpeta de **fusiones** en la carpeta raíz de que Porject Córdoba. En la carpeta de fusión, cree el directorio para cada plataforma (android / ios ..). Luego, en la carpeta de la plataforma específica, cree una carpeta css / js y ponga su archivo css / js específico para la carpeta de la plataforma. Eso es todo, una vez que ejecute el comando **cordova build** , todos los archivos js / css correspondientes a cada plataforma se colocarán en la carpeta de la plataforma respetada

**Nota:** asegúrese de que su raíz www / index.html tenga el mismo css / js definido. Para eso, asegúrese de tener el mismo nombre de archivo correspondiente a cada plataforma en la carpeta de combinaciones.

```
//let say you are in CordovaMergesExample folder
cd CordovaMergesExample

//create test folder with com.test id and TestApp as name
cordova create test com.test TestApp

//add platform android and ios
cordova platform add android ios

-----

//create merges/android/css/override.css and merges/ios/css/override.css

-----

//In root www/index.html add this stylesheet
<link rel="stylesheet" type="text/css" href="css/override.css" />

-----

cordova build

---> cordova build engine automatically identify the platform in merges folder and add files
in respective folders. Check platforms/android/assets/www/css and platforms/ios/www/css
```

Lea [Cómo personalizar la plataforma de la carpeta www en cordova en línea:](https://riptutorial.com/es/cordova/topic/7498/como-personalizar-la-plataforma-de-la-carpeta-www-)

<https://riptutorial.com/es/cordova/topic/7498/como-personalizar-la-plataforma-de-la-carpeta-www->



# Capítulo 5: Complemento de imagen de la cosecha de Córdoba

## Examples

Recorte la imagen después de hacer clic con la cámara o seleccionar la imagen.

Recorta las imágenes en forma cuadrada.

Este proyecto de Córdoba utiliza dos complementos:

1. Cordova Camera Plugin - <https://github.com/apache/cordova-plugin-camera>
2. Cordova Crop Image Plugin - <https://github.com/jeduan/cordova-plugin-crop>

El complemento de la cámara se combina con el complemento de imagen de recorte al colocar el código del complemento de imagen de Cop dentro de la devolución de llamada exitosa del código del complemento de la cámara.

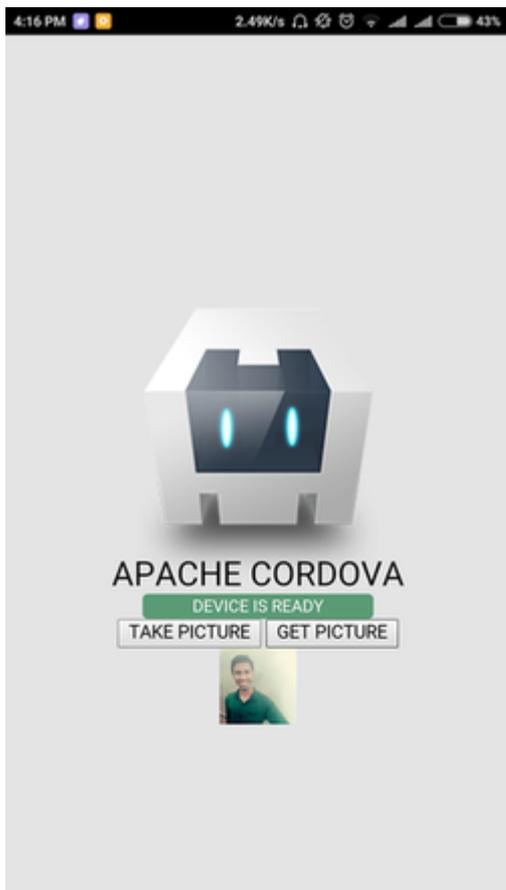
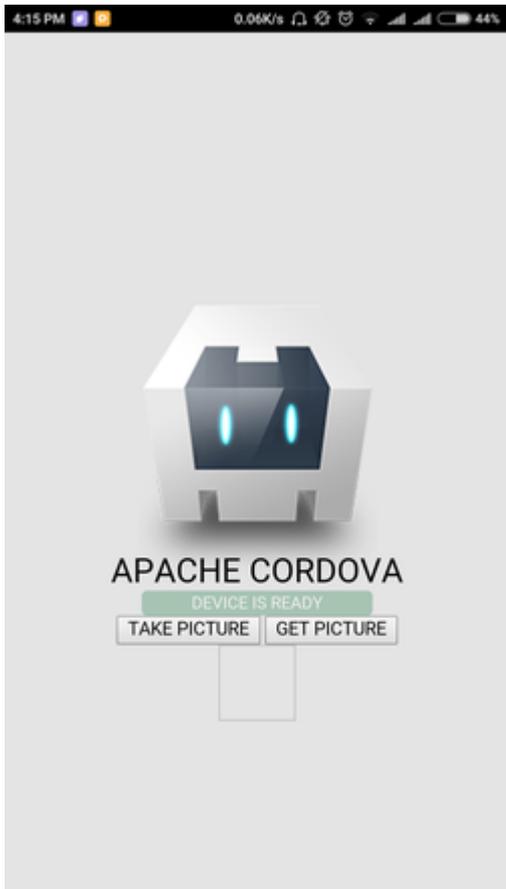
```
/*Camera Plugin Code*/
navigator.camera.getPicture(onSuccess, onFail, {
    quality: 50,
    destinationType: Camera.DestinationType.FILE_URI
});

function onSuccess(imageData) {
    console.log(imageData);

    /*Crop Image Plugin Code*/
    plugins.crop(function success (data) {
        console.log(data);
        var image = document.getElementById('myImage');
        image.src = data;
    },
    function fail () {

    }, imageData, {quality:100});
}

function onFail(message) {
    alert('Failed because: ' + message);
}
```



Lea Complemento de imagen de la cosecha de Córdoba en línea:

<https://riptutorial.com/es/cordova/topic/7078/complemento-de-imagen-de-la-cosecha-de-cordoba>

---

# Capítulo 6: Complementos de Cordova: cómo instalar, cómo funcionan, ejemplos.

## Examples

### ¿Qué son los complementos de Córdoba?

Los complementos de Cordova son en palabras simples una capa sobre la plataforma nativa respectiva.

Los complementos proporcionan una interfaz entre para acceder a la plataforma nativa.

### ¿Cómo pueden ser útiles los complementos de Cordova?

Los complementos de Cordova proporcionan una interfaz común para interactuar con el código nativo.

Cada complemento tiene un archivo JavaScript intermediario que proporciona acceso a características específicas de la plataforma.

## Instalar el complemento de Cordova

```
cordova plugin add <plugin-name>
```

Ejemplo: complemento de cordova agregar cordova-plugin-camera

El complemento debe instalarse en el directorio raíz del proyecto.

Nota:

Antes de agregar el complemento, reemplace el contenido de la carpeta www específica de la plataforma con la carpeta www externa en el directorio raíz. Esto se debe a que al agregar un complemento, el contenido de la carpeta www externa se reemplaza por la carpeta www específica de la plataforma.

Nota:

Cuando agrega una nueva plataforma y si tiene algún complemento instalado en su proyecto, no es necesario instalar los complementos existentes. Cordova agregará automáticamente los complementos instalados para la nueva plataforma.

## Complementos más populares

1. **cordova-plugin-battery-status** : se utiliza para controlar el estado de la batería del dispositivo.

2. [cordova-plugin-camera](#) : proporciona una API para tomar fotografías y para elegir imágenes de la biblioteca de imágenes del sistema.
3. [cordova-plugin-contacts](#) : proporciona acceso a la base de datos de contactos del dispositivo.
4. [cordova-plugin-device](#) - describe el hardware y el software del dispositivo.
5. [cordova-plugin-device-motion](#) : acceso al acelerómetro del dispositivo.
6. [cordova-plugin-file](#) : implementa un archivo API que permite el acceso de lectura / escritura a los archivos que residen en el dispositivo.
7. [cordova-plugin-geolocation](#) : proporciona información sobre la ubicación del dispositivo, como la latitud y la longitud.
8. [cordova-plugin-globalization](#) : obtiene información y realiza operaciones específicas para la configuración regional, el idioma y la zona horaria del usuario.
9. [cordova-plugin-inappbrowser](#) : muestra artículos útiles, videos y recursos web dentro de tu aplicación. Los usuarios pueden ver páginas web sin salir de su aplicación.
10. [cordova-plugin-network-information](#) : proporciona información sobre la conexión celular y wifi del dispositivo, y si el dispositivo tiene una conexión a Internet.
11. [cordova-plugin-vibration](#) : proporciona una forma de vibrar el dispositivo.
12. [cordova-plugin-statusbar](#) : proporciona algunas funciones para personalizar la [barra de estado de](#) iOS y Android.
13. [cordova-plugin-whitelist](#) : implementa una política de lista blanca para navegar por la aplicación webview en Cordova 4.0. **Complemento recomendado!**

Para los complementos específicos de cordova, siga el enlace <https://cordova.apache.org/plugins/>

Lea Complementos de Cordova: cómo instalar, cómo funcionan, ejemplos. en línea:  
<https://riptutorial.com/es/cordova/topic/7077/complementos-de-cordova--como-instalar--como-funcionan--ejemplos->

---

# Capítulo 7: Cordova ios construir

## Introducción

Espero que estés familiarizado con los conceptos básicos de Córdoba. Intentemos construir una compilación de ios de cordova, la compilación de ios es un poco diferente a la de la compilación de Android, necesitamos la máquina mac para realizar esta tarea. No, también puede preparar la compilación de ios en línea, pero para probar y depurar su aplicación en un simulador de mac, debe tener la máquina mac con usted.

vamos a empezar con un ejemplo.

## Observaciones

Enfrenta cualquier problema durante la preparación de la compilación de ios. Por favor, contácteme y trataré de ayudarlo.

## Examples

### cordova HelloWorld Project

Para preparar el ios build primero necesitamos crear el proyecto de cordova. vamos a crear el proyecto por la herramienta de línea de comandos.

```
cordova create hello com.example.hello "HelloWorld"
```

Ir al directorio de proyectos por `cd hello` .

Ahora estamos en el directorio del proyecto, veamos qué plataformas están disponibles para nosotros.

```
cordova platform ls
```

Como necesitamos preparar la compilación de ios, agregaremos la plataforma de ios al proyecto.

```
cordova platform add ios@version    => put the desired version you want to add.  
cordova platform add ios@latest.    => add the latest version available.
```

Ahora necesitamos **Xcode** en la máquina mac para preparar la compilación de iOS.

Descargue el último Xcode disponible para nosotros a través de la tienda de aplicaciones. Por favor ignore las versiones beta de xcode ya que tienen problemas de compatibilidad en algún momento.

Después de la instalación de Xcode, instale Xcode cli usando el terminal mediante el comando a

continuación.

```
xcode-select --install
```

Se abrirá la ventana, siga las instrucciones e instale el cli correctamente. Ahora solo necesitamos una herramienta más para implementar la compilación de ios, instalarla también.

```
npm install -g ios-deploy
```

durante la instalación de la herramienta de implementación, si obtiene el error de permiso denegado, intente con el comando sudo.

```
sudo npm install -g ios-deploy
```

vamos a preparar la compilación ios usando los comandos

```
cordova platform add ios
```

Si no has agregado la plataforma antes, agrégala.

```
cordova prepare => it will move the all required files to platform folder and create a .xcworkspace file in the platform.ios folder.
```

```
cordova build ios => to build the ios application.
```

Pero espera, no hemos ejecutado la compilación en el simulador, vamos a hacerlo también.

tienes dos formas de hacerlo

1. usando cli
2. usando Xcode

**Primero vamos a hacerlo con CLI.**

```
cordova run ios => it will run the application on the default simulator available.
```

Si desea jugar con los simuladores, explore el comando del `cordova emulator`

**usando Xcode**

salte la carpeta del proyecto y muévase a la carpeta `/platform/ios folder` de su proyecto. abra el archivo `.xcworkspace` con el comando `open <ProjectName>.xcworkspace` por ejemplo, `open MyApp.xcworkspace`.

lo redireccionará a la ventana de Xcode, allí, en la parte superior izquierda de la ventana, podrá ver su proyecto, hacer clic en el proyecto y, en el encabezado superior, podrá ver el botón Ejecutar, hacer clic en él y jugar con su aplicación.

Gracias.

Lea Cordova ios construir en línea: <https://riptutorial.com/es/cordova/topic/10666/cordova-ios-construir>

---

# Capítulo 8: Creando tu primera aplicación con Cordova

## Observaciones

En caso de ejecutar `cordova run android` falla. Asegúrese de que su dispositivo Android esté conectado a su computadora y ejecute `adb devices` para asegurarse de que las Herramientas de desarrollo de Android (ADT) puedan detectar su dispositivo.

## Examples

### Usando la herramienta de línea de comandos

Primero creas un nuevo proyecto de Córdoba:

```
cordova create HelloWorld my.application.identifier AppName
```

Esto creará un proyecto de Córdoba en blanco.

- en la carpeta *HelloWorld*
- con el identificador *my.application.identifier* (que debe ser único para cada aplicación)
- con el nombre *AppName* .

A continuación agrega las plataformas deseadas:

```
cordova platform add android
// and/or
cordova platform add browser
// and/or
cordova platform add ios // On macOS only
// etc...
```

Construye tu aplicación para generar un archivo ejecutable:

```
cordova build // Build project for all platforms
cordova build ios // Build project only for iOS platform
cordova build android // Build project only for Android platform
```

Una vez desarrollada, puede ejecutar la aplicación en una de las plataformas que agregó:

```
cordova run android --emulator // Run Android app in emulator
cordova run android --device // Run Android app on physical connected device
cordova run browser // Will run the app in the browser
```

Si desea compilar la aplicación para Eclipse, Xcode, Visual Studio, etc:

```
cordova prepare [platform_name] // Prepare copies of www folder and any plugins into the appropriate platform folder
```

Lea **Creando tu primera aplicación con Cordova en línea:**

<https://riptutorial.com/es/cordova/topic/2396/creando-tu-primer-a-aplicacion-con-cordova>

---

# Capítulo 9: Depurando la aplicación

## Observaciones

Una cosa importante que debe recordar al depurar aplicaciones cordova, si tiene un evento `OnDeviceReady` y un código que se ejecuta allí, para cuando se inicie la aplicación, su depurador aún no estará adjunto (a diferencia de la depuración de Visual Studio C # donde la aplicación espera a que el proceso de depuración adjuntar antes de continuar con el lanzamiento del programa).

Esto significa que no se capturarán los mensajes de la consola de configuración inicial o los puntos de interrupción.

La solución para esto puede ser una configuración diferida o un registro de consola retrasado con `setTimeout` cuando se dispara el evento `DeviceReady`.

## Examples

### Depurar en dispositivo Android utilizando USB

Una aplicación Cordova se ejecuta como un sitio web en un componente `WebView` dentro de la plataforma móvil nativa. Por lo tanto, la depuración de una aplicación cordova se puede hacer utilizando las herramientas de desarrollo de sus navegadores favoritos. Se necesitan los siguientes pasos para conectar la aplicación, que se ejecuta en el dispositivo, al navegador Chrome en una máquina de desarrollo:

1. Habilite la depuración USB en su dispositivo ( [puede seguir esta guía](#) )
2. Instale el `adb` Debug Bridge de Android (no se requiere en las versiones recientes de Chrome) ( [guía para OSX](#) )
3. Conecte su teléfono y ejecute `adb devices` en su terminal (no es necesario en las versiones recientes de Chrome), y seleccione `ok` en la ventana emergente de su teléfono ¿ `Allow USB debugging?` .
4. Abrir cromo
5. Busque `chrome://inspect` , o elija [Más herramientas => Inspeccionar dispositivos ...](#)
6. Seleccione su dispositivo y depure utilizando las herramientas de desarrollo de Chrome

Dependiendo de su dispositivo, es posible que necesite descargar primero los controladores USB.

También debe habilitar "fuentes desconocidas" en Seguridad en Configuración si desea cargar la aplicación en su teléfono.

### Depurar aplicaciones de Cordova usando GapDebug

<https://www.genuitec.com/products/gapdebug/>

GapDebug es una herramienta de depuración móvil integral que llena el vacío que dejan otras

opciones de depuración. Al operar en las plataformas Windows y Mac, GapDebug permite la depuración de aplicaciones móviles híbridas, como PhoneGap y Cordova, en dispositivos iOS y Android modernos. Y, GapDebug siempre es gratis para la depuración local.

El paso para la configuración por primera vez se da en este siguiente enlace:

<https://www.genuitec.com/products/gapdebug/learning-center/configuration/>

## Depuración en dispositivo iOS utilizando USB

### 1. Deshabilitar la navegación privada

Abra la configuración de Safari de su dispositivo y asegúrese de que **la navegación privada esté desactivada** . La depuración remota no funcionará si *la navegación privada* está habilitada.

### 2. Habilitar inspector web

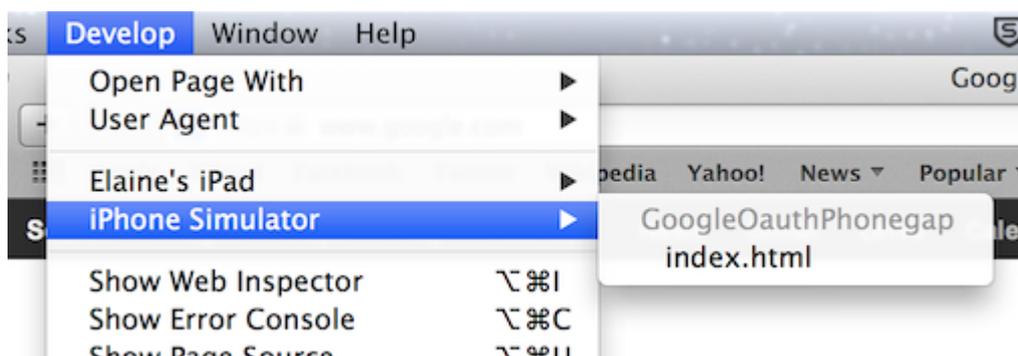
Toque la pestaña *Avanzado* en la configuración de Safari de su dispositivo y asegúrese de que **Web Inspector esté activado** .

### 3. Habilitar el menú de desarrollo de Safari

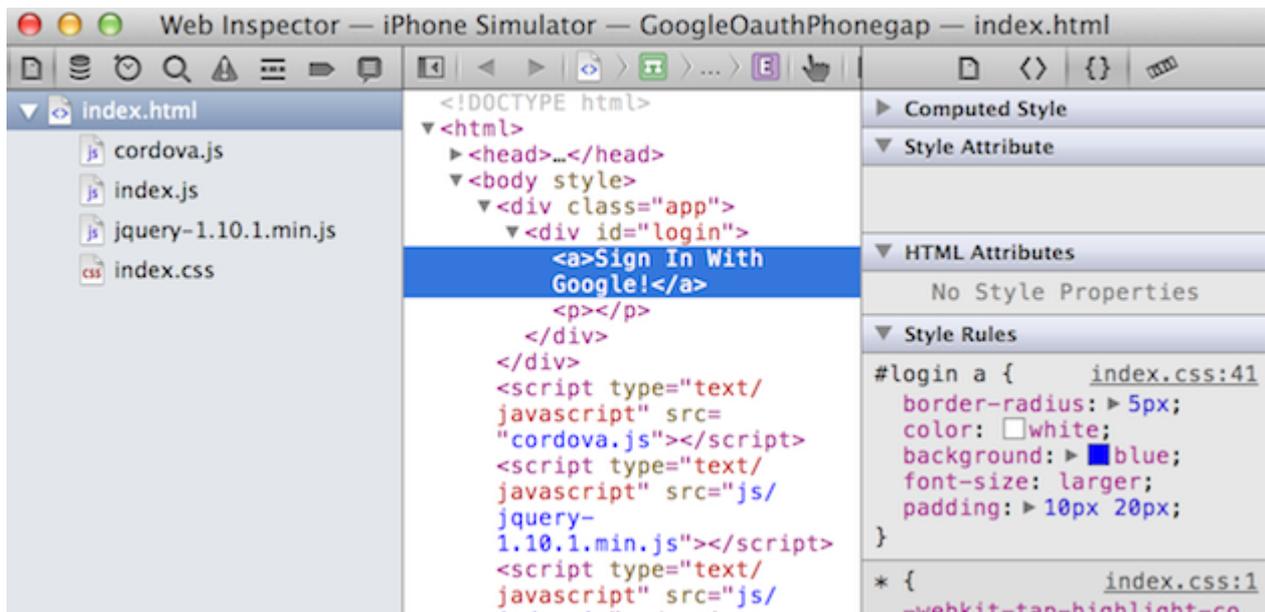
En su computadora de escritorio o portátil, abra las Preferencias de Safari y haga clic en la pestaña Avanzado. Marque la casilla para **mostrar el menú Desarrollar en la barra de menús** .

### 4. Iniciar inspector web

Inicie su aplicación en el simulador de iOS o en un dispositivo físico. Si está utilizando un dispositivo físico, deberá conectarlo a su computadora de escritorio o portátil con el cable USB estándar. Una vez que la aplicación se haya iniciado, cambie a Safari, seleccione el elemento del menú **Desarrollar** , luego busque la entrada correspondiente a la página web que desea depurar.



Ahora puede utilizar el inspector web como lo haría para depurar una página web.



Lea Depurando la aplicacion en línea: <https://riptutorial.com/es/cordova/topic/4004/depurando-la-aplicacion>

---

# Capítulo 10: Empezando con Cordova

## Introducción

Desarrollo de aplicaciones móviles usando Cordova

Apache Cordova es un marco de desarrollo móvil de código abierto. Le permite utilizar tecnologías web estándar: HTML5, CSS3 y JavaScript para el desarrollo multiplataforma.

## Examples

### Creación de la compilación de Android (.apk)

Instale cordova usando el siguiente comando **npm install -g cordova** .

Utilice cordova -version para comprobar la **versión de cordova** .

Establecer las variables de ruta ANDROID\_HOME y JAVA\_HOME.

Ejemplo:

```
exportar ANDROID_HOME = / home / geethu / android-sdk-linux
```

```
export PATH = $ PATH: $ ANDROID_HOME / tools: $ ANDROID_HOME / platform-  
tools
```

```
export JAVA_HOME = / usr / lib / jvm / java-7-openjdk-amd64
```

```
export PATH = $ PATH: $ JAVA_HOME / bin
```

Cree un proyecto Cordova en blanco usando la herramienta de línea de comandos. Navegue hasta el directorio donde desea crear su proyecto y tipo.

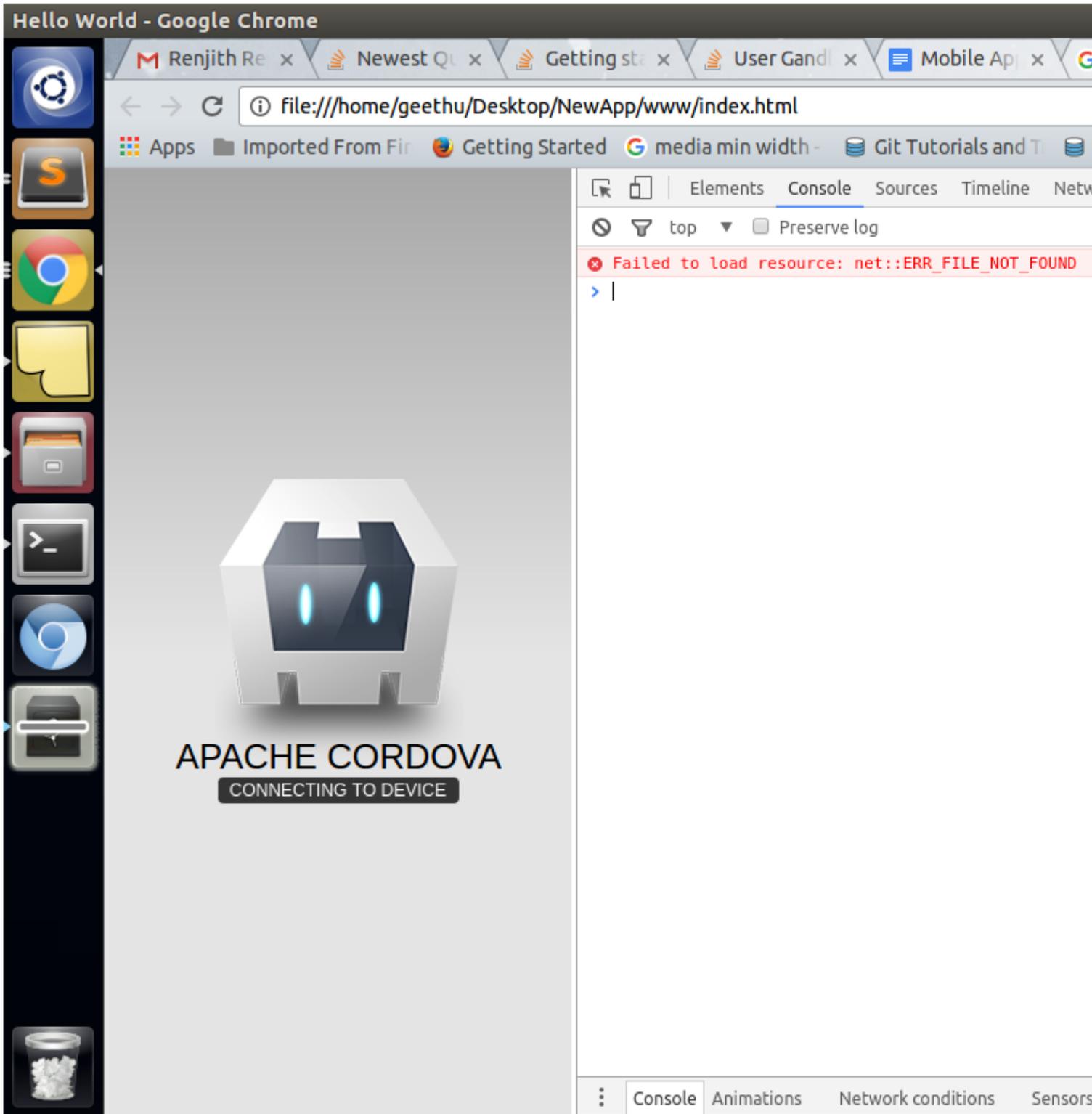
**cordova crea taller com. tu nombre. taller taller**

Cordova CLI, cree un proyecto de Cordova llamado Taller en un directorio llamado Taller.

Navegue hasta el directorio del proyecto cd workshop.

Añadir la plataforma de Android como la plataforma de **Córdoba agregar android** .

EI



] 1

Agregue estos archivos de script a su index.html en el siguiente orden.

```
<script type = "text/javascript" src = "cordova.js"></script>
```

Para cambiar el nombre de la aplicación y el icono, edite el archivo config.xml en la carpeta www.

Para construir el proyecto en el taller / plataformas / carpeta android.

**cordova construir android**

Ejecútelo en un dispositivo Android conectado a su computadora con un cable USB, escriba:

## cordova ejecuta android

---

### 1. > Cordova .apk instalación en dispositivo

---

---

Además, me gustaría agregar algunos detalles más sobre cómo puede instalar el archivo .apk en su dispositivo móvil Android, mientras implementa su aplicación en el servidor.

```
cordova plugin add cordova-plugin-mfp
```

ejecute el comando anterior, luego intente agregar el servidor en el que desea implementar la aplicación.

```
mfpdev server add
```

luego siga las instrucciones de CLI. ejecute el siguiente comando para cambiar sus ajustes de configuración.

```
mfpdev app config server <profile Name of server>  
  
cordova build
```

se creará archivo .apk en el

/ plataformas / carpeta de Android

descarga el apk en tu móvil, instálalo y juega con tu aplicación.

Lea Empezando con Cordova en línea: <https://riptutorial.com/es/cordova/topic/9287/empezando-con-cordova>

# Capítulo 11: Firebase Push Notification Cordova

## Examples

### Firebase Push Notification en Cordova Android

#### Agrega Firebase a tu proyecto de Android

##### *Agrega Firebase a tu aplicación*

Para agregar Firebase a su aplicación, necesitará un proyecto Firebase y un archivo de configuración de Firebase para su aplicación.

1. Cree un proyecto Firebase en la consola Firebase, si aún no tiene uno. Si ya tiene un proyecto de Google existente asociado con su aplicación móvil, haga clic en Importar Google Project. De lo contrario, haga clic en Crear nuevo proyecto.
2. Haga clic en Agregar Firebase a su aplicación de Android. Si está importando un proyecto de Google existente, esto puede suceder automáticamente y simplemente puede descargar el archivo de configuración.
3. Cuando se le solicite, ingrese el nombre del paquete de su aplicación. Es importante ingresar el nombre del paquete que usa tu aplicación; esto solo se puede configurar cuando agrega una aplicación a su proyecto Firebase.
4. Al final, descargará un archivo google-services.json. Puedes descargar este archivo de nuevo en cualquier momento. Si aún no lo ha hecho, copie esto en la carpeta del módulo de su proyecto, normalmente app /.

## Cordova Firebase Push Notification Plugin

<https://www.npmjs.com/package/cordova-plugin-fcm>

Para obtener el token de acceso:

```
FCMPlugin.getToken(  
  function(token) {  
    alert(token);  
  },  
  function(err) {  
    console.log('error retrieving token: ' + err);  
  }  
);
```

Devolución de llamada para recibir notificación de inserción:

```
FCMPlugin.onNotification(  
  function(data) {
```

```

    if(data.wasTapped){
        //Notification was received on device tray and tapped by the user.
        alert( JSON.stringify(data) );
    }else{
        //Notification was received in foreground. Maybe the user needs to be notified.
        alert( JSON.stringify(data) );
    }
},
function(msg){
    console.log('onNotification callback successfully registered: ' + msg);
},
function(err){
    console.log('Error registering onNotification callback: ' + err);
}
);

```

Coloque el token de acceso y la devolución de llamada para recibir una notificación de inserción dentro del archivo index.js dentro de la función receiveEvent

## Envío de notificaciones push a través de la API REST

```

//POST: https://fcm.googleapis.com/fcm/send
//HEADER: Content-Type: application/json
//HEADER: Authorization: key=AlzaSyAMMh0mdVIRXPcBejyatAtdZgmklepwoNs //key is server-key
{
  "notification":{
    "title":"Notification title", //Any value
    "body":"Notification body", //Any value
    "sound":"default", //If you want notification sound
    "click_action":"FCM_PLUGIN_ACTIVITY", //Must be present for Android
    "icon":"fcm_push_icon" //White icon Android resource
  },
  "data":{
    "param1":"value1", /Any data to be retrieved in the notification callback
    "param2":"value2"
  },
  "to":"eRIImo7algBM:APA91bHSxS0dmgsOi9su_XytEtCbei0Zi00Dgm76VHvbqeb-
WPoZcLyNVpnaLWPLw7U1u93h00ZhtBxn_hVGxPAwXXfc-yNy6_kkfzUdTpcI2QPB0vzJBmOFzX3RRZ15wmFkCUFtyhc",
//Topic or single device
  "priority":"high", //If not set, notification won't be delivered on completely closed
iOS app
  "restricted_package_name":"com.zensar.fcm" //Optional. Set for application filtering
}

```

Configure la API REST anterior utilizando el cliente de resto de Postman.

The screenshot shows the Postman interface for a POST request to `https://fcm.googleapis.com/fcm/send`. The 'Headers' tab is selected, showing two headers:

Header Name	Value
Content-Type	application/json
Authorization	key=AlzaSyAMMh0mdVIRXPcBejyatAtdZgmklepwoNs

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary Text

```
1 {
2   "notification":{
3     "title":"Notification title",
4     "body":"Notification body",
5     "sound":"default",
6     "click_action":"FCM_PLUGIN_ACTIVITY",
7     "icon":"fcm_push_icon"
8   },
9   "data":{
10    "param1":"value1",
11    "param2":"value2"
12  },
13   "to":"eRImo7alg8M:APA91bHSxS0dmgs0i9su_XytEtCbei0Zi00Dgm76VHvbqeb-WPoZcLyNVpnaLWPLw7U1u93h00ZhtBxn_h
14     -yNy6_kkfzUdTpcI2QP80vzJBmOFzX3RRZ15wmFkCUFtyhc",
15   "priority":"high",
16   "restricted_package_name":"com.zensar.fcm"
17 }
```

### Cómo funciona *Envíe una notificación de inserción a un solo dispositivo o tema.*

1.a La aplicación está en primer plano: el usuario recibe el mensaje de notificación en la barra de notificaciones del dispositivo. El usuario toca la notificación y se abre la aplicación. El usuario recibe los datos de notificación en la devolución de llamada de JavaScript '.

1.b La aplicación está en segundo plano: el usuario recibe el mensaje de notificación en su barra de notificaciones del dispositivo. El usuario toca la notificación y se abre la aplicación. El usuario recibe los datos de notificación en la devolución de llamada de JavaScript '.

Lea [Firebase Push Notification Cordova](https://riptutorial.com/es/cordova/topic/6892/firebase-push-notification-cordova) en línea:

<https://riptutorial.com/es/cordova/topic/6892/firebase-push-notification-cordova>

# Capítulo 12: Firma la compilación de Android con Cordova 5

## Examples

### Agregue la configuración de compilación para firmar el archivo .apk

1. Agregue un almacén de claves usando:

```
keytool -genkey -v -keystore example.keystore -alias example -keyalg RSA -keysize 2048 -  
validity 10000
```

Nota: Esto debe estar en la raíz del proyecto. Aunque no es un requisito difícil, facilita la referencia del archivo

2. Agregue un build.json con la configuración de release / dev para el almacén de claves, en la raíz del proyecto:

```
{  
  "android": {  
    "debug": {  
      "keystore": "..\android.keystore",  
      "storePassword": "android",  
      "alias": "mykey1",  
      "password" : "password",  
      "keystoreType": ""  
    },  
    "release": {  
      "keystore": "..\android.keystore",  
      "storePassword": "",  
      "alias": "mykey2",  
      "password" : "password",  
      "keystoreType": ""  
    }  
  }  
}
```

3. Agregue el interruptor --buildConfig al comando de compilación Cordova / Ionic:

```
cordova build android --release --buildConfig=build.json
```

o con jónico como

```
ionic build android --release --buildConfig=build.json
```

El archivo firmado se generará bajo la nueva estructura de carpetas en

```
/platforms/android/build/outputs/apk/android-release.apk
```

Lea Firma la compilación de Android con Cordova 5 en línea:

<https://riptutorial.com/es/cordova/topic/3061/firma-la-compilacion-de-android-con-cordova-5>

# Capítulo 13: Google Analytics en Córdoba

## Examples

### Google Analytics en Córdoba sin ningún plugin.

Inserte la función de análisis dentro de index.js

```
function analytics(){

    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','lib/analytics.js','ga');

    if(window.localStorage) {

        ga('create', 'UA-XXXXXXX-1', {
            'storage': 'none'
            , 'clientId': window.localStorage.getItem('ga_clientId') /*The tracker id
obtained from local storage*/
        });
        ga(function(tracker) {
            window.localStorage.setItem('ga_clientId', tracker.get('clientId'));
            /*The tracker id for each device is different and stored in local storage*/
        });
    }
    else {

        ga('create', 'UA-XXXXXXX-1', 'auto');
    }

}
```

Inserte cada una de las siguientes etiquetas de script en cada página html y modifique el nombre de la página

```
<script>
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','lib/analytics.js','ga');

    ga('set','checkProtocolTask',null);
    /*checkProtocol Task is set to null so that GA allows tracking other than http/https
*/

    ga('set', 'page', "Page Name");
    /*Page Name is name of each html page*/

    ga('send', 'pageview');
</script>
```

Lea Google Analytics en Córdoba en línea: <https://riptutorial.com/es/cordova/topic/6909/google>



---

# Capítulo 14: Hacer la aplicación liberada de Cordova CLI

## Examples

### Androide

**Paso 1:** vaya al directorio raíz del proyecto y abra la línea de comandos

```
cordova build --release android
```

Esto genera un apk sin firmar en `\platform\android\build\Output\apk` con el nombre **android-release-unsigned.apk**

**Paso 2:** Generación de claves para obtener apk firmado

Sintaxis:

```
keytool -genkey -v -keystore <keystoreName>.keystore -alias <Keystore AliasName> -keyalg <Key algorithm> -keysize <Key size> -validity <Key Validity in Days>
```

Ejemplo:

```
keytool -genkey -v -keystore ExampleApp.keystore -alias TestExampleApp -keyalg RSA -keysize 2048 -validity 10000
```

```
keystore password? : xxxxxxxx
What is your first and last name? : xxxxxxx
What is the name of your organizational unit? : xxxxxxxx
What is the name of your organization? : xxxxxxxxx
What is the name of your City or Locality? : xxxxxxxx
What is the name of your State or Province? : xxxxxx
What is the two-letter country code for this unit? : xxx
```

El almacén de claves se genera en la misma carpeta con el nombre `ExampleApp.keystore`

**Paso 3:** Mueva el almacén de claves generado a `\platform\android\build\Output\apk`

Ejecute la herramienta jarsigner en el símbolo del sistema en `\platform\android\build\Output\apk`

Sintaxis:

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore <keystorename <Unsigned APK file> <Keystore Alias name>
```

## Ejemplo:

```
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore ExampleApp.keystore android-release-unsigned.apk TestExampleApp
```

Esto genera la apk firmada con el mismo nombre.

### **Paso 4:** herramienta de alineación zip para optimizar el APK

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

El código postal se encuentra en \ Android \ sdk \ build-tools \ 23.0.3 \ zipalign

Esto genera una apk firmada con el nombre android.apk que ahora se puede cargar en la tienda de aplicaciones

## iOS

**Paso 1:** Cree un archivo build.json en el directorio raíz del proyecto.

Muestra de build.json

```
{
  "ios": {
    "debug": {
      "codeSignIdentity": "iPhone Developer",
      "provisioningProfile": "your-developer-provisioning-profile-UUID-here"
    },
    "release": {
      "codeSignIdentity": "iPhone Distribution",
      "provisioningProfile": "your-distribution-provisioning-profile-UUID-here"
    }
  }
}
```

Nota: El UUID se puede obtener abriendo el archivo .mobileprovision en un editor de texto y busque 'UUID'.

**Paso 2:** ejecute el siguiente comando desde la carpeta raíz del proyecto en el terminal

```
cordova build ios --device --release
```

Lea [Hacer la aplicación liberada de Cordova CLI en línea:](https://riptutorial.com/es/cordova/topic/7092/hacer-la-aplicacion-liberada-de-cordova-cli)

<https://riptutorial.com/es/cordova/topic/7092/hacer-la-aplicacion-liberada-de-cordova-cli>

# Capítulo 15: Notificación Push en Android y iOS

## Examples

### Usando el nuevo phonegap-plugin-push

Con el fin de enviar notificaciones push a aplicaciones cordova. El primer paso es obtener un token de dispositivo. Un "token de dispositivo" es específico para cada dispositivo y cada proyecto.

#### Requisito previo :

1. Número de proyecto de Google Cloud Messaging

Para ello, vaya a la [Consola de desarrollador de Google](#) y cree un nuevo proyecto. Bajo la información del proyecto es el número del proyecto

2. Clave de la API de Google Cloud Messaging para el proyecto anterior (necesario para el servidor)

Ir a la biblioteca -> Google Cloud Messaging -> Habilitar. Vaya a Credenciales para crear una clave API de Type server.

### Credentials

## Add credentials to your project

### 1 Find out what kind of credentials you need

We'll help you set up the correct credentials.

If you wish to you can skip this step and create an [API key](#), [client ID](#) or [service account](#).

#### Which API are you using?

Determines what kind of credentials you need.

Google Cloud Messaging

#### Where will you be calling the API from?

Determines which settings you'll need to configure.

Web server (e.g. node.js, Tomcat)

What credentials do I need?

Añadiendo el plug-in push al proyecto:

```
cordova plugin add https://github.com/phonegap/phonegap-plugin-push --variable
```

```
SENDER_ID="XXXXXXX"
```

**SENDER\_ID** representa el ID del proyecto

Coloque el siguiente código dentro de la función `receiveEvent` dentro de `index.js`

```
var push = PushNotification.init({
  android: {
    senderID: "XXXXXX"
  },
  ios: {
    alert: "true",
    badge: "true",
    sound: "true"
  },
  windows: {}
});

push.on('registration', function(data) {
  console.log("device token: " + data.registrationId);
});

push.on('notification', function(data) {
  console.log(data.message);
  console.log(data.title);
  console.log(data.count);
  console.log(data.sound);
  console.log(data.image);
  console.log(data.additionalData);
});

push.on('error', function(e) {
  console.log(e.message)
});
```

Al ejecutar el código anterior desde un dispositivo Android o iOS, se obtiene un token de dispositivo.

NOTA: El token del dispositivo se generará solo en un dispositivo real, no en un dispositivo virtual.

Para probar la notificación de inserción, haga clic aquí. Prueba de notificación de inserción en [línea](#)

**Para Android:** ingrese el token del dispositivo, el mensaje y la clave API

# GCM

**Device Token**

**Message**

**Api Key**

Submit

Lea Notificación Push en Android y iOS en línea:

<https://riptutorial.com/es/cordova/topic/6181/notificacion-push-en-android-y-ios>

---

# Capítulo 16: Visual Studio Tools para Apache Cordova

## Examples

### Obtén Apache Cordova Tools en Visual Studio

1. Abra el Panel de control -> Programas y características, elija el elemento de Visual Studio 2015 y luego elija el botón Cambiar.
2. En el asistente de configuración para Visual Studio, elija el botón Modificar.
3. En la lista de características opcionales para instalar, seleccione la casilla de verificación HTML / JavaScript (Apache Cordova), elija el botón Siguiente y luego el botón Actualizar.

### Actualizar las herramientas de Apache Cordova en Visual Studio

1. En Visual Studio, elija Herramientas-> Extensiones y actualizaciones.
2. En la pestaña Actualizaciones del cuadro de diálogo Extensiones y actualizaciones, seleccione Actualizaciones del producto. Si aparece una actualización para Visual Studio Tools para Apache, selecciónela y luego elija el botón Actualizar.

Lea Visual Studio Tools para Apache Cordova en línea:

<https://riptutorial.com/es/cordova/topic/2393/visual-studio-tools-para-apache-cordova>

# Creditos

S. No	Capítulos	Contributors
1	Empezando con Cordova	<a href="#">Akash Pal</a> , <a href="#">Community</a> , <a href="#">Hitesh Riziya</a> , <a href="#">Michiel</a> , <a href="#">Mikhail</a> , <a href="#">ModusPwnens</a>
2	Cómo detectar el estado de la conexión de red.	<a href="#">Philip Bijker</a>
3	Cómo instalar / desinstalar el complemento de cordova personalizado	<a href="#">Rahul Raghuvanshi</a>
4	Cómo personalizar la plataforma de la carpeta www en cordova	<a href="#">Rahul Raghuvanshi</a>
5	Complemento de imagen de la cosecha de Córdoba	<a href="#">Akash Pal</a>
6	Complementos de Cordova: cómo instalar, cómo funcionan, ejemplos.	<a href="#">Akash Pal</a> , <a href="#">Hristo Eftimov</a>
7	Cordova ios construir	<a href="#">sparrowTrajon</a>
8	Creando tu primera aplicación con Cordova	<a href="#">Devid Farinelli</a> , <a href="#">grgarside</a> , <a href="#">Hristo Eftimov</a> , <a href="#">James Wong</a> , <a href="#">Philip Bijker</a> , <a href="#">ProllyGeek</a>
9	Depurando la aplicacion	<a href="#">Akash Pal</a> , <a href="#">Alex Filatov</a> , <a href="#">Alexus</a> , <a href="#">Devid Farinelli</a> , <a href="#">mike nelson</a> , <a href="#">Philip Bijker</a>
10	Firestore Push Notification Cordova	<a href="#">Akash Pal</a>
11	Firma la compilación de Android con Cordova 5	<a href="#">Aditya Singh</a>

12	Google Analytics en Córdoba	<a href="#">Akash Pal</a>
13	Hacer la aplicación liberada de Cordova CLI	<a href="#">Akash Pal</a>
14	Notificación Push en Android y iOS	<a href="#">Akash Pal</a>
15	Visual Studio Tools para Apache Cordova	<a href="#">Charitha Goonewardena</a>