

Ambiente de desarrollo - Release

[Instalar](#)

[Paso a paso de lo necesario](#)

[Documentación una vez armado el ambiente](#)

[A la hora de armar un release](#)

Instalar

- **SDK Android**, la última versión que haya en el link, con simplemente el SDK alcanza, pero si quieren instalar el ADT Bundle que contiene el SDK y el Eclipse con el plugin para Eclipse, go ahead:
<http://developer.android.com/sdk/>
- **Apache Cordova** v3.4 - <http://cordova.apache.org/>
pre-requisitos: **NodeJS** v0.10.28 <http://nodejs.org/>
- **Ionic framework JS** (basado en **Angular JS**)
<http://ionicframework.com/>

Paso a paso de lo necesario

1. **Android SDK Tools** - Bajar de <https://developer.android.com/sdk/index.html#ExistingIDE>
Luego ejecutar el manager e instalar Android SDK Tools, Platform Tools, Build Tools y todo el paquete API 19.
2. Si se tiene instalado la Java Development Kit, saltar este paso, si no bajar la JDK 7 e instalarla:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
3. **Apache ANT** - Bajar e instalar Winant <https://code.google.com/p/winant/>
4. **NodeJS** v0.10.28 <http://nodejs.org/> - Bajar el instalador y ejecutarlo, esto nos instala una herramienta que funciona como manejador de paquetes (del estilo Aptitude de Ubuntu, por ejemplo) que nos permite instalar diferentes herramientas con el comando "npm", al usar npm, la opción "-g" significa que el o los paquetes que se quieren instalar, se instalen globalmente en el sistema, incluso eso hace que se añada en el PATH del sistema los ejecutables de cada paquete luego de instalarlo.

5. **Apache Cordova** v3.4 - Instalar con Node JS (ingresar a consola de Node JS)
npm install -g cordova@3.4.1-0.1.0
6. **GULP** (Para compilar y minificar estilos y archivos .js y comprimir imágenes)
npm install -g gulp
7. **Ionic** (framework HTML5)
npm install -g ionic
8. **Git**: Bajar e instalar: <http://git-scm.com/download>
9. Al tener ya creada una aplicación debemos instalar algunos **plugins** de **gulp** con el siguiente comando:
npm install gulp-util bower gulp-concat gulp-sass gulp-minify-css gulp-rename gulp-jshint gulp-uglify shelljs

Documentación una vez armado el ambiente

- Docs de Ionic:
 - <http://ionicframework.com/docs/guide/preface.html>
 - <http://ionicframework.com/docs/components>
- Cordova Platform Tools Guide:
 - [Android](#)
 - [iOS](#)

(Nota: Si en alguna web ven una herramienta llamada PhoneGap, es lo mismo que Cordova)

A la hora de armar un release

Para generar los archivos javascript, css y de imágenes compilados, ubicado en la raíz del proyecto ejecutar el comando:

```
gulp default
```

Remover plugins innecesarios, por ejemplo el debug console "org.apache.cordova.console" (que sí es necesario en etapa de desarrollo)

```
cordova plugin rm [nombre del plugin]
```

Particularmente para Android:

Modificar el archivo **AndroidManifest.xml** y setear en false esta configuración:

```
<application android:debuggable="false" android:hardwareAccelerated =  
"true"...>
```

El comando para generar el release es el siguiente (plataforma: android o ios, etc...) esto genera un archivo de tipo Nombre_del_proyecto.apk

```
cordova build --release [plataforma]
```

Luego se debe firmar el archivo generado (.apk), el SDK de Android provee herramientas para eso, primero generamos una key con “keytool” (cambiar “my-release-key.keystore” por el nombre que quieran)

```
keytool -genkey -v -keystore my-release-key.keystore -alias alias_name  
-keyalg RSA -keysize 2048 -validity 10000
```

y luego aplicamos la key en el archivo apk generado con jarsigner

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

por último optimizamos el archivo apk con la herramienta zipalign:

```
zipalign -v 4 Nombre_del_proyecto-unsigned.apk  
Nombre_del_proyecto.apk
```