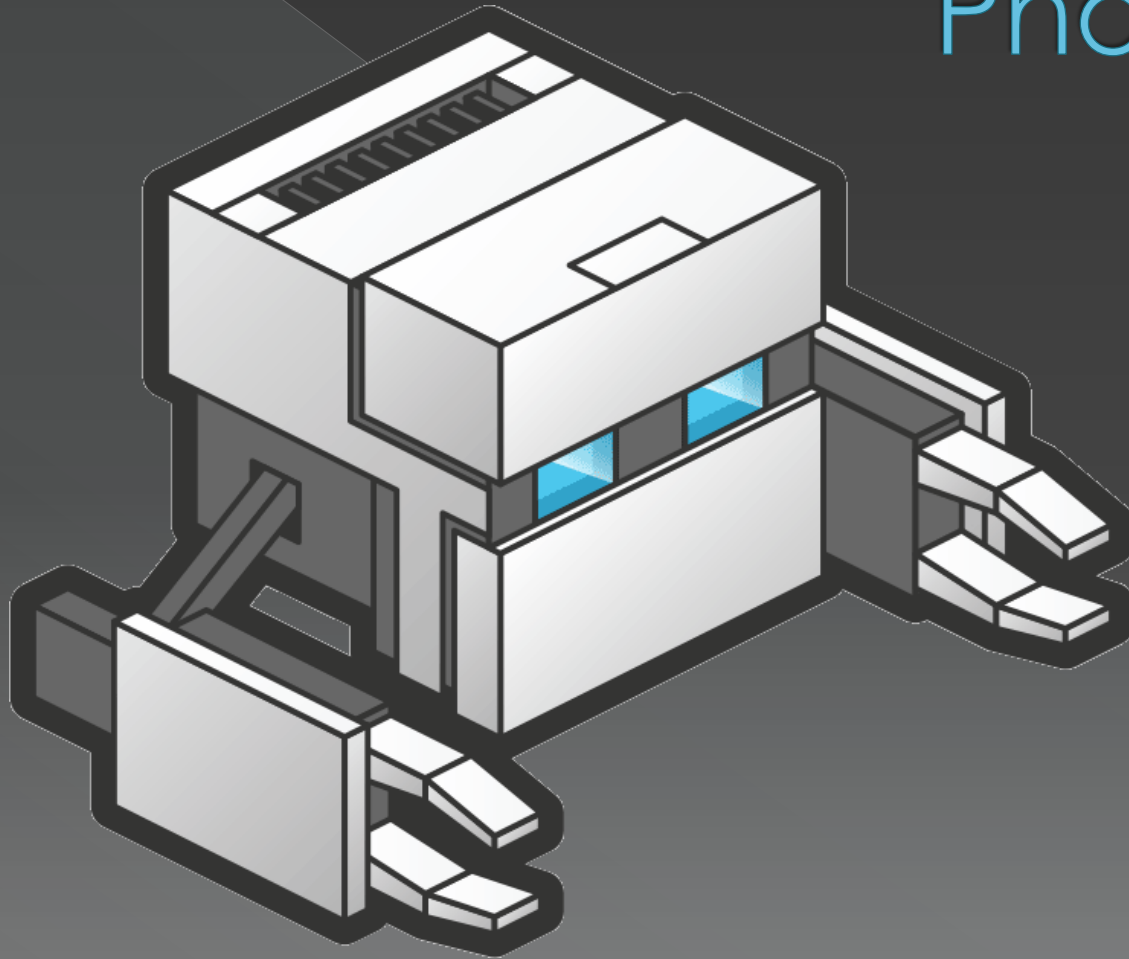


PhoneGap



¿Qué es PhoneGap?

- PhoneGap es un entorno de desarrollo para crear aplicaciones híbridas para dispositivos móviles
- Se llaman **aplicaciones híbridas** porque utilizan tecnologías web sobre una base escrita en código nativo



PhoneGap

¿Qué es Phonegap?

- PhoneGap fue creada en 2009 por la empresa Nitobi, que más tarde fue adquirida por Adobe.
- Cuando se produjo esta absorción, la base de PhoneGap fue liberada como **código abierto** a la fundación Apache.
- A esta base se la conoce también como **Cordova**,
- Puede decirse que Cordova es la base gratuita y central, y que **PhoneGap es Cordova más otros añadidos** exclusivos
- Primo pobre y primo rico.

Phonegap

- En este curso trabajaremos con el primo rico, por simplicidad.
 - > Ahorramos instalar las plataformas para las que queremos programar(android, windows,ios)
 - > Ahorramos instalación y configuración en nuestro pc de NodeJ y Cordova CLI.
 - > Ahorramos trabajar por comandos.

PhoneGap- Productos



PHONEGAP DESKTOP APP

Our desktop app is the snappiest way to create new PhoneGap apps without using the command line.

[LEARN MORE](#)



PHONEGAP DEVELOPER

Use our mobile app to connect your devices to your development machine, then see the changes you make instantly

[LEARN MORE](#)



PHONEGAP BUILD

Take the pain out of compiling locally — get app-store ready apps without the headache of maintaining native SDKs.

[LEARN MORE](#)



PLUGIN LIBRARY

Get access to the robust library of plugins to fully extend the capabilities of your hybrid mobile applications.

[GO TO THE LIBRARY](#)



THIRD PARTY TOOLS

Find additional tools created by our amazing community to help you test, debug and manage your apps.

[VIEW THE TOOLS](#)



DEVELOPER COMMUNITY

Connect and collaborate with our community of thousands, from beginners to experts, all working with PhoneGap.

[JOIN THE CONVERSATION](#)

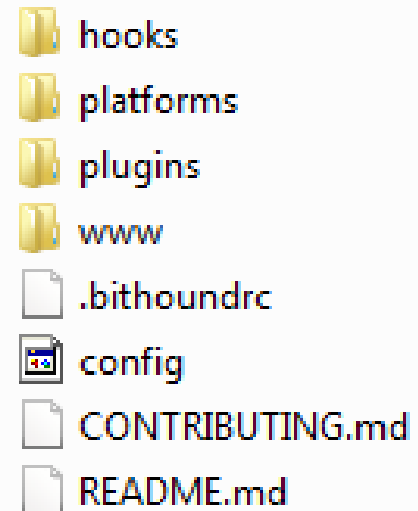
PhoneGap Desktop App

- PhoneGap Desktop es la forma más sencilla de empezar a usar PhoneGap.
- Se usa para crear aplicaciones y servir esas aplicaciones a dispositivos móviles conectados.
- Es la alternativa al uso de la CLI, utilizando las mismas bibliotecas bajo el capó.
- no hay necesidad de memorizar comandos o instalar dependencias - todos están incluidos al descargar PhoneGap Desktop.

<http://phonegap.com/products/#desktop-app-section>

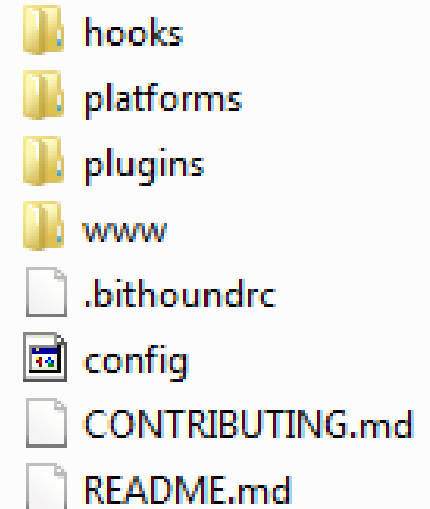
PhoneGap Desktop App

- Una vez hayas instalado la aplicación procederemos a crear nuestra primera aplicación hybrida con phonegap.
 - > Helloworld .
- Al crear el proyecto ya podemos observar la estructura del proyexto



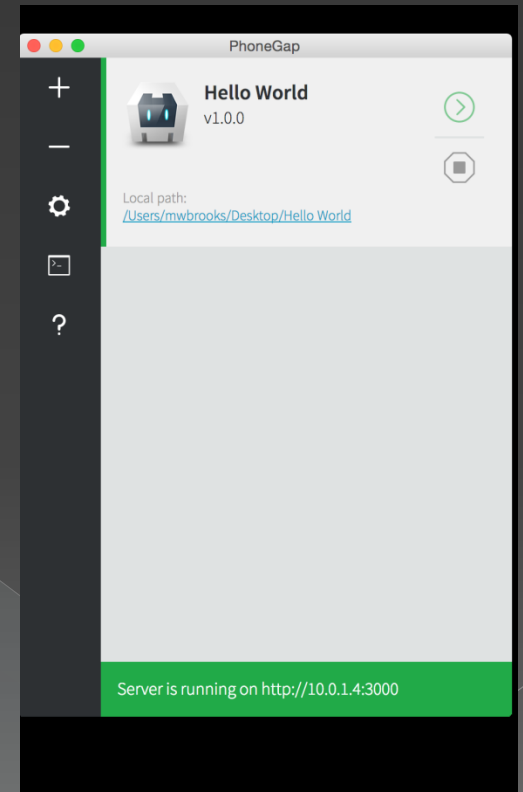
PhoneGap Desktop App

- En la carpeta **www** están todos nuestros web(html,js,css, imágenes,audios etc)
- En **plugins** estarán todas las librerías js que nos permitirán hacer uso de funciones del dispositivo(sensores,notificaciones etc..)
- En **platforms** estará el software necesario de cada plataforma en la que sustentan los plugins.x defecto solo el navegador)
- **Hooks** representa scripts especiales que pueden ser añadidos por los desarrolladores de aplicaciones o complementos o incluso por tu propio sistema de compilación para personalizar comandos de cordova.
- **Config.xml** será el archivo de configuración de phonegap para tu app.



PhoneGap Desktop App

- Una vez creado el proyecto podemos probarlo.
- PhoneGap Desktop App actuará como servidor de nuestra aplicación.
- Podemos probarlo en el inspector de código de Google Chrome en modo dispositivo móvil.
- También podemos utilizar Phonegap developer App para probarlo en un dispositivo móvil.



Inspeccionando el Hola Mundo

ViewPort

- Abra el archivo index.html
- Observe el elemento meta viewport:
 - > `<meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height, target-densitydpi=device-dpi" />`
- Esto se utiliza para indicar cuánto de la pantalla debe ser utilizado por el contenido de la aplicación y especificar cómo debe escalar.
- En la aplicación predeterminada, los ajustes se configuran para cargar el contenido al 100%, (escala inicial = 1) no permiten escalar el usuario (escalable al usuario = no) y utilizar el ancho y la altura máximos del dispositivo.

Inspeccionando el Hola Mundo

● Cordova.js

- En el archivo index.html verá una etiqueta de script apuntando a un archivo cordova.js :
 - > `<Script type = "text / javascript" src = "cordova.js"> </ script>`
- El archivo **cordova.js** es la **biblioteca PhoneGap** (con el código fuente de Apache Cordova, de ahí el nombre) y se utiliza para acceder específicamente al hardware del dispositivo nativo (cámara, contactos, GPS, etc.) desde JavaScript en nuestras aplicaciones PhoneGap.
- **No hay un archivo cordova.js** ubicado en cualquier parte de la carpeta. Esto se debe a que la versión correcta depende de la plataforma. Se inyecta en tiempo de ejecución mediante Developer app o a la hora de empaquetar la aplicación con phonegap build.

Inspeccionando el Hola Mundo

● **Index.js**

- El archivo index.js es otro archivo JavaScript al que se hace referencia en index.html. Este archivo no es obligatorio, es específico para esta aplicación predeterminada y se utiliza para agregar lógica simple.
- Determinar cuándo la biblioteca de Cordova se ha cargado y está lista para ser utilizada.
- Observe que el index.html contiene una línea para llamar a una función de inicialización a través de una variable de aplicación justo antes de la etiqueta de cuerpo HTML de cierre:

```
<Script type = "text / javascript">  
  App.initialize ();  
</ Script>
```
- Esto llama a la función initialize en la variable app definida en el archivo index.js en la carpeta www /js.

Inspeccionando el Hola Mundo

- **Deviceready**
- Otra importante característica específica de Córdoba es el evento **deviceready**. Este evento indica que las API de dispositivos de Cordova **se han cargado y están listas para acceder**.
- Si empieza a hacer llamadas a las API de Cordova sin depender de este evento, podría terminar en una situación en la que el código nativo todavía no está completamente cargado y no está disponible.
- Las aplicaciones suelen adjuntar un **detector de eventos** con `document.addEventListener` una vez que el DOM del documento HTML se haya cargado como se muestra a continuación y en la aplicación Hello predeterminada:
- `Document.addEventListener ('deviceready', this.onDeviceReady, false);`

Inspeccionando el Hola Mundo

- En el archivo index.js verá que la función `onDeviceReady` llama a una función `receiveEvent` para mostrar visualmente que el dispositivo ya está listo.

```
onDeviceReady: function() {
    app.receiveEvent('deviceready');
},
// Update DOM on a Received Event
receiveEvent: function(id) {
    var parentElement = document.getElementById(id);
    var listeningElement = parentElement.querySelector('.listening');
    var receivedElement = parentElement.querySelector('.received');

    listeningElement.setAttribute('style', 'display:none;');
    receivedElement.setAttribute('style', 'display:block;');

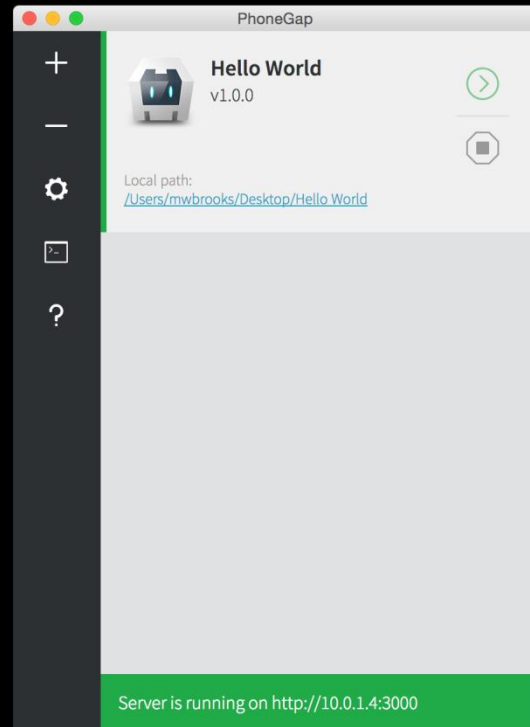
    console.log('Received Event: ' + id);
}
```

Phonegap developer mobile app

- Es una aplicación móvil para ver rápidamente cómo se ve y funciona tu aplicación en un móvil.
- El complemento perfecto para PhoneGap Desktop App.
- Puedes ver los cambios al instante en tu dispositivo móvil conectado a la misma red. sin necesidad de volver a firmar, volver a compilar o reinstalar su aplicación para probar su código.
- Al probar tu aplicación PhoneGap en un dispositivo móvil, obtienes acceso a las API del dispositivo que no están disponibles en los navegadores web.

Phonegap developer mobile app

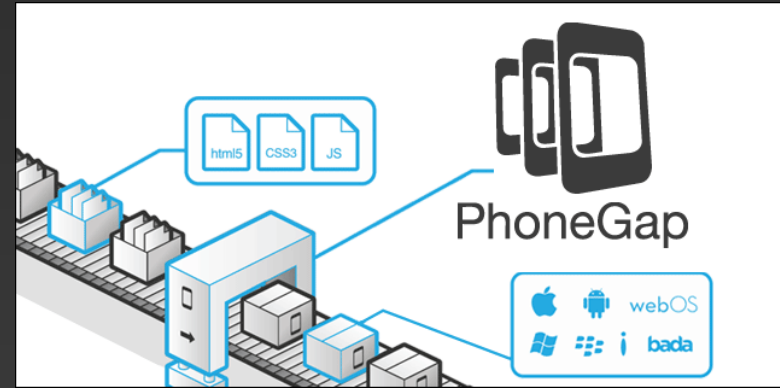
- Actualmente es compatible con teléfonos móviles Android, ios y Windows Phone.
- Aunque no es compatible con todos los moviles y versiones de Android.



Más allá

- Visto como instalar phonegap, crear aplicación y probarlas en un dispositivo móvil, solo falta ver como empaquetar estas aplicaciones para ser distribuidas.
- Existen 2 opciones:
 - > Construir y empaquetar localmente usando la CLI de PhoneGap
 - > Utilizar PhoneGap Build: el servicio en la nube para simplificar el proceso de creación y empaquetado de aplicaciones

Phonegap Build



- Compilación y firma administradas (sin SDK)
 - > El desarrollo local implica la instalación y el mantenimiento de varios SDK nativos, así como el Cordova / PhoneGap SDK. PhoneGap Build lleva este dolor de cabeza lejos!
- Varias plataformas soportadas
 - > iOS, Android, Windows (Teléfono) todos con una sola base de código. Como servicio web, PhoneGap Build puede utilizarse desde cualquier sistema operativo.
- Trabaja con tu equipo
 - > Trabaje de forma colaborativa agregando miembros del equipo y creando roles dentro de los proyectos de PhoneGap Build.
- Ciclo de desarrollo más rápido
 - > La hidratación se usa para obtener ciclos de depuración y generación más rápidos. Con la hidratación, las actualizaciones de su aplicación se aplican directamente a las aplicaciones instaladas anteriormente en sus probadores, asegurando que todos estén trabajando en la versión más actualizada.

¿Cómo estructurar mi app para utilizar phonegap build?

- El único requisito mínimo de PhoneGap Build para la estructura de tu aplicación es que haya un **index.html**
- Pero con toda seguridad necesitarás una estructura mucho más complicada que contendrá los archivos javascript, la pantalla de bienvenida y las imágenes de iconos, archivos multimedia y cualquier otra carga útil requerida por su aplicación.

¿Cómo estructurar mi app para utilizar phonegap build?

- **Config.xml**
- Necesitará un config.xml, en su paquete de aplicacione para configurar cómo se crea su aplicación.
- Esto incluye la versión PhoneGap, iconos y pantallas de bienvenida, plataformas y mucho más.

Config.xml

◉ <Widget>

El elemento widget debe ser la raíz de su documento XML - nos permite saber que usted está siguiendo la especificación W3C. Cuando utilice PhoneGap Build, asegúrese de que tiene los siguientes atributos establecidos en su elemento widget. Soporta los siguientes atributos:

- > **Id:** el identificador único de su aplicación. Para admitir todas las plataformas soportadas, debe ser estilo de nombre de dominio inverso (por ejemplo, com.suempresa.yourapp)
- > **Versión:** para obtener mejores resultados, utilice una versión de estilo mayor / menor / parche, con tres números, como 0.0.1
- > **VersiónCode:** (opcional) al construir para Android, puede establecer el código de versión especificándolo en su config.xml.

<name> El nombre de la aplicación.

<Description> Una descripción para su aplicación.

<Author> El autor de la aplicación, ya sea una empresa o un individuo (necesario para Windows 10).

<Platform> Puede tener cero o más de estos elementos presentes en su config.xml. Establezca el atributo name en uno de ios, android o windows. Si no especifica ninguno, se crearán todas las plataformas. Ejemplo de uso:

```
<Platform name = "ios" />
```

```
<Platform name = "android" />
```

```
<Platform name = "winphone" />
```

Config.xml

- Los iconos y splashscreen suelen ser específicos de la plataforma. Hay dos maneras de especificar un icono o splash para una plataforma en particular.

- > La primera forma es especificar un atributo de plataforma:
 - `<Icon src = "icon-60@3x.png" plataforma = "ios" width = "180" height = "180" />`
- > La segunda forma (recomendada) es poner el icono o splash dentro de una etiqueta de plataforma:

```
<Platform name = "ios">
```

```
  <Icon src = "icon-60@3x.png" width = "180" height = "180" />
```

```
</ Platform>
```

- Ambos fragmentos se traducirá en el icono que se utiliza para iOS.

Plugins

- Para extender la funcionalidad nativa expuesta por el contenedor de aplicación nativa de PhoneGap, PhoneGap Build es compatible con la mayoría de los complementos PhoneGap o Cordova.
- Los plugins pueden ser del [repositorio de phonegap build](#), npm o de un repositorio git.
- Los plugins deben implementarse de forma diferente para cada plataforma y es posible que no se admitan en todas las plataformas de PhoneGap. Si se está desplegando en varias plataformas, asegúrese de que la experiencia no empeora demasiado para los usuarios que no disponen del plugin
- Ejemplo de plugins: estado de la batería, gps, vibración, información del estado de la red...

Plugins

- Hay dos pasos para incluir un complemento en su proyecto:
 - > Importación del código nativo mediante el archivo config.xml
 - > Referencia del código JavaScript para el complemento

Plugins -> Config.xml

- Para añadir el plugin al config.xml debes usar la etiqueta `<plugin>`, pudiendo usar los siguientes atributos:
 - > **Name**: Los complementos deben ser referenciados por el ID de plugin que normalmente está en un formato de dominio inverso (por ejemplo: `com.phonegap.plugins.barcodescanner`). Opcional si el plugin es git-backed.
 - > **Spec**: Opcional, es la version y se recomienda indicarla. Si no se indica, se cogerá siempre la última versión de plugin..
 - > **Source**: Opcional, puede ser "pgb", "npm" o "git". El valor predeterminado es "npm" (o "git" si se detecta una URL git).
 - > **Params**: Los complementos pueden requerir parámetros para las propiedades de configuración.

Plugins -> JS

- Si un plugin utiliza el elemento js-module para cordova no se necesitarán referencias `<script>` para cargarlo. Este es el caso de los plugins core cordova, pero los plugins de terceros dependerán de la implementación. Consulte la documentación del complemento para determinar si debe incluir manualmente el javascript.
- Si necesitas incluir manualmente el plugin javascript, se incluye así:
 - > `<Script src = "cordova.js"> </ script>`
 - > `<Script src = "barcodescanner.js"> </ script>`