



WEB FULL STACK DEVELOPER

*Germán Caballero Rodríguez*  
*gcaballero@pronoide.es*



# Complejidad del desarrollo móvil

## ¿Por qué Cordova?

Beneficios de Cordova en el desarrollo de apps móviles multiplataforma



# INDICE

## 1.Sistema general de información con necesidad de movilidad

### 1. Subsistemas implicados:

1. Subsistema móvil
2. Subsistema WEB
3. Subsistema WS

## 2.APIs de los dispositivos

## 3.Complejidad del desarrollo móvil

## 4.¿Por qué Cordova?

1. Características
2. Arquitectura
3. Plataformas y sistemas operativos

## 5.Resumen

# 1.Sistema general de información con necesidad de movilidad

Un sistema que necesita toma de datos desde un dispositivo móvil para un posterior almacenamiento en un sistema externo, además deberá funcionar **independientemente** de la **conexión con el exterior**.

# Subsistemas implicados

## Subsistema móvil

Aplicación instalada localmente con almacenamiento temporal en el propio dispositivo.

- Puede ser nativa o híbrida.
- Puede funcionar sin necesidad de estar conectado.

# Subsistemas implicados

## Subsistema WEB

Aplicación WEB al uso que nos sirve para realizar la explotación de los datos obtenidos desde el subsistema móvil

La interfaz debe adaptarse al tamaño de dichos dispositivos: **Diseño web responsive**

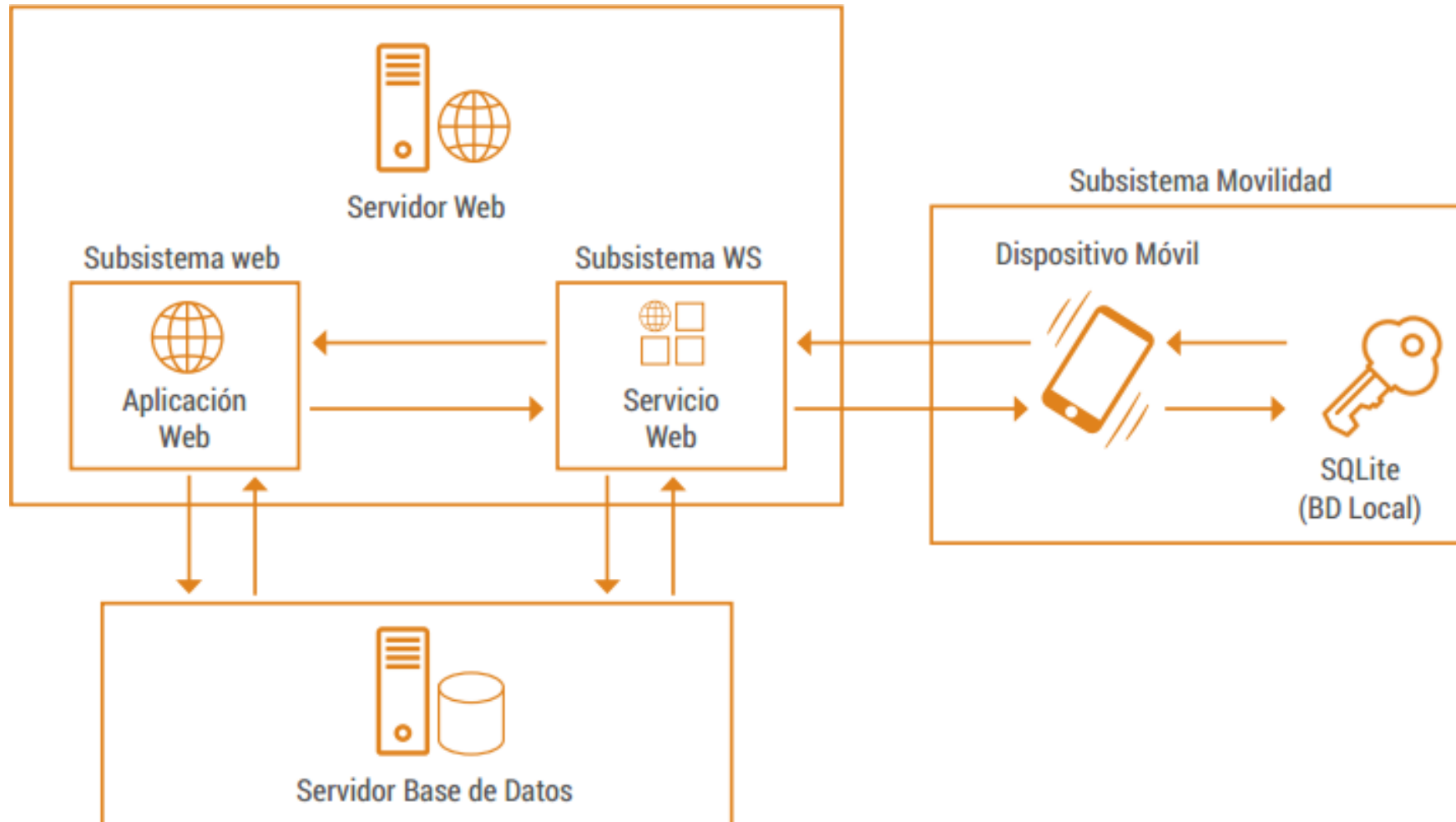
# Subsistemas implicados

## Subsistema WS

Servicio Web que nos permite el almacenamiento de datos en el sistema externo.

- Podría estar incluido en el subsistema WEB.
- Podría ser REST (mejor rendimiento), SOAP (mayor seguridad)

# Subsistemas implicados





## 2.APIs de los dispositivos

A continuación se muestra el conjunto de herramientas de desarrollo y APIs de dispositivos disponibles para cada plataforma móvil.

	android	blackberry10	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.1, 10, Phone 8.1)	OS X
<b>cordova CLI</b>	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓	✓ Mac
<b>Embedded WebView</b>	✓ (see details)	X	✓ (see details)	✓	X	X	✓
<b>Plugin Interface</b>	✓ (see details)	✓ (see details)	✓ (see details)	✓	✓ (see details)	✓	✓

## 2.APIs de los dispositivos

	android	blackberry10	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.1, 10, Phone 8.1)	OS X
Accelerometer	✓	✓	✓	✓	✓	✓	X
BatteryStatus	✓	✓	✓	✓	✓	✓ * Windows Phone 8.1 only	X
Camera	✓	✓	✓	✓	✓	✓	X
Capture	✓	✓	✓	✓	✓	✓	X
Compass	✓	✓	✓ (3GS+)	✓	✓	✓	X
Connection	✓	✓	✓	✓	✓	✓	X
Contacts	✓	✓	✓	desktop only	✓	partially	X
Device	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✓	✓	✓	X
File	✓	✓	✓	✓	✓	✓	✓
File Transfer	✓	✓ * Do not support onprogress nor abort	✓	X	✓ * Do not support onprogress nor abort	✓ * Do not support onprogress nor abort	X
Geolocation	✓	✓	✓	✓	✓	✓	X

## 2.APIs de los dispositivos

	android	blackberry10	ios	Ubuntu	wp8 (Windows Phone 8)	windows (8.1, 10, Phone 8.1)	OS X
Globalization	✓	✓	✓	✓	✓	✓	X
InAppBrowser	✓	✓	✓	✓	✓	uses iframe	X
Media	✓	✓	✓	✓	✓	✓	
Notification	✓	✓	✓	✓	✓	✓	X
Splashscreen	✓	✓	✓	✓	✓	✓	X
Status Bar	✓	X	✓	X	✓	✓ Windows Phone 8.1 only	X
Storage	✓	✓	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	X
Vibration	✓	✓	✓	✓	✓	✓ * Windows Phone 8.1 only	X

### 3.Complejidad del desarrollo móvil

Grandes expectativas para los desarrolladores a día de hoy en el innovador mundo de los dispositivos móviles.

**¿Para qué plataforma comienzo a desarrollar?**

Cada una resulta tener algo apetecible, como la cuota de mercado, el número de usuarios, la facilidad para desarrollar, en ocasiones el lenguaje en que se desarrollar las aplicaciones.

### 3.Complejidad del desarrollo móvil

Lo ideal es desarrollar para **todas en nativo**.

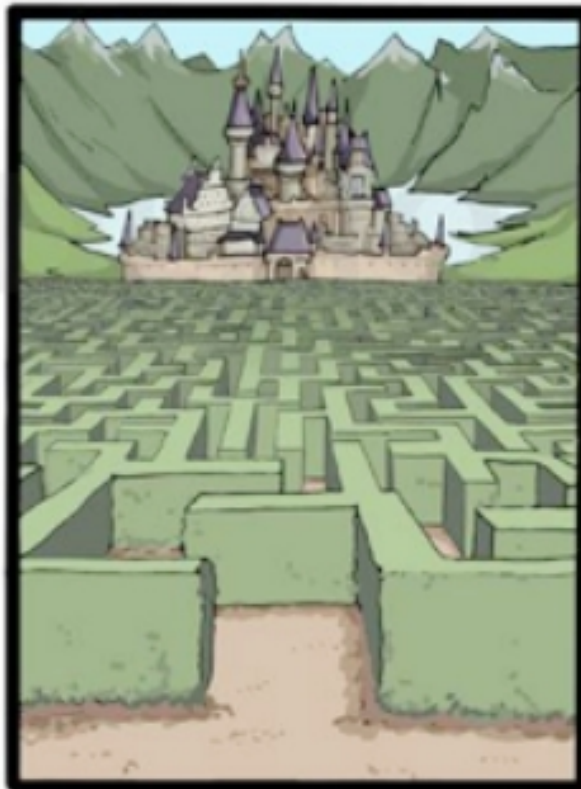
Sin embargo, esto **requiere de gran trabajo**, ¿por qué? porque las aplicaciones Android y BlackBerry se desarrollan usando Java, iOS usa Objective C como lenguaje de desarrollo y Symbian usa C++.

**Hablamos de manejar muchos lenguajes de programación** para desarrollar una sola aplicación para dispositivos móviles, eso sin mencionar **todos los SDKs** que debemos instalar para lograr esto.

### 3.Complejidad del desarrollo móvil

#### The dilemma of mobile apps development

Develop a native app for each device and maintain several projects



Use a unique framework (Phonegap, Adobe Air, Appcelerator) and maintain only one project



### 3.Complejidad del desarrollo móvil

Otras opciones multiplataforma:

- **Unity 3D / Unreal Engine:** Entornos de desarrollo de videojuegos con motores de videojuegos multiplataforma
- **Xamarin:** Desarrollo multiplataforma con C#.

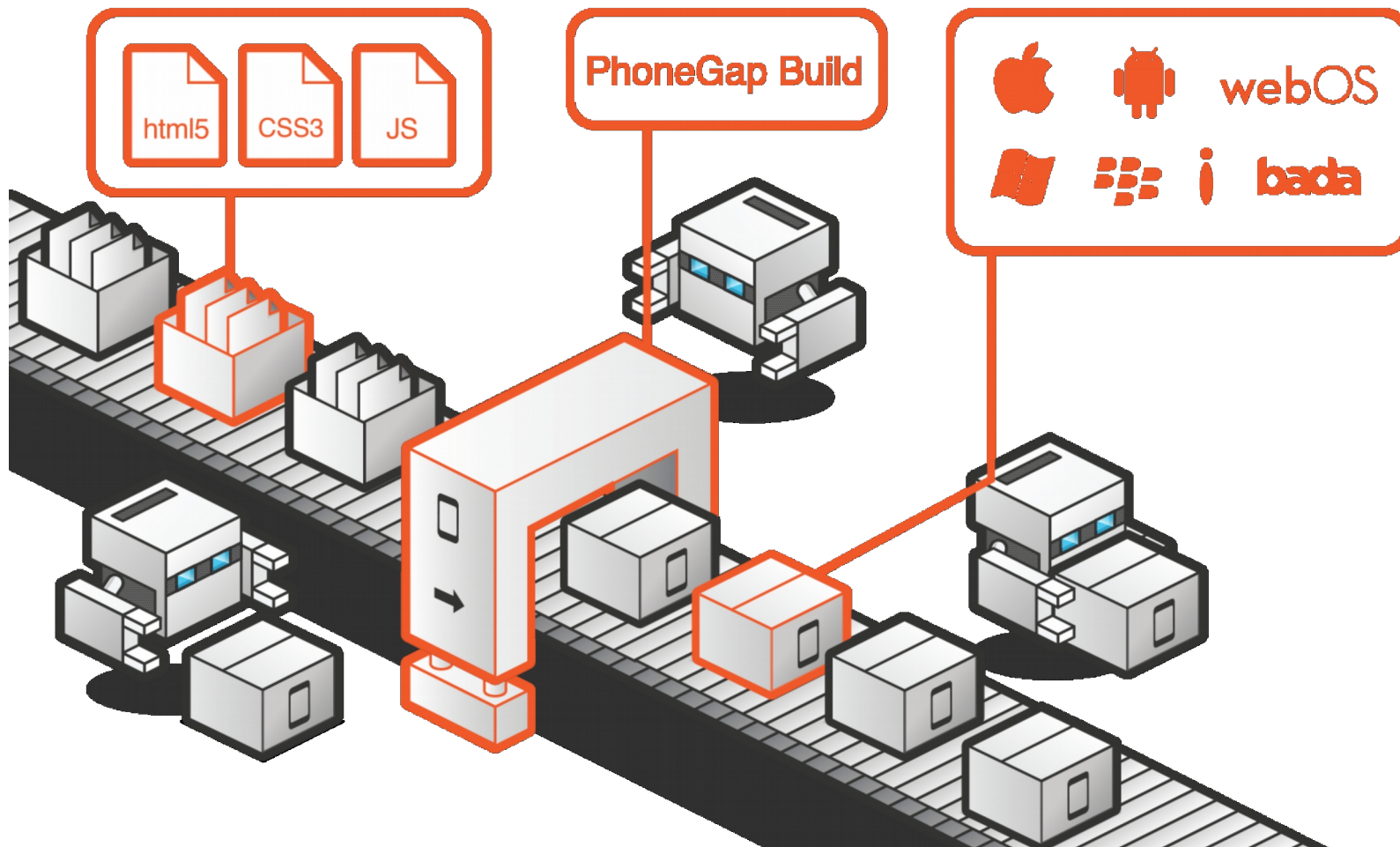
## 4.¿Por qué Cordova?

Permite trabajar para **varias plataformas** en paralelo **sin** la necesidad de una **modificación del código**. Sólo es necesario la generación de los instalables para cada uno de los Sistemas Operativos.

Nos evita el desarrollo particularizado, permitiendo que el grueso del negocio de **nuestras aplicaciones sea común para todas las plataformas**.



## 4. ¿Por qué Cordova?



## 4.¿Por qué Cordova? - Características

Cordova es un framework para el desarrollo de **aplicaciones nativas de sistemas operativos móviles**, haciendo uso de tecnologías web como **HTML5, CSS3 y JavaScript**. Con Cordova es posible desarrollar aplicaciones para los siguientes sistemas operativos:

- Android.
- iOS.
- Windows Phone.
- BlackBerry OS.
- Web OS.
- Symbian.
- Bada.

## 4.¿Por qué Cordova? - Características

Inicialmente **Cordova era PhoneGap**, desarrollado por Nitobi bajo licencias de software libre, en Octubre de 2011 **Adobe** anunció oficialmente la **adquisición de Nitobi**, pasando, así PhoneGap al control del gigante del software, gran interesado en la evolución de HTML5.

Posteriormente, **Adobe dona PhoneGap a la fundación Apache**, conservando de esta forma la integridad libre de PhoneGap.

## 4.¿Por qué Cordova? - Características

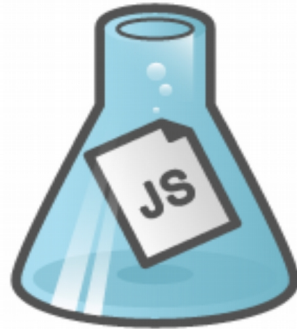
En la actualidad, el proyecto en el sitio web de la fundación Apache esta nombrado como: **"Apache Cordova"**, pero PhoneGap sigue siendo una especie de marca comercial, por lo que aún se sigue usando ese nombre para identificar al popular framework.

Cordova y PhoneGap también es compatible con frameworks de desarrollo web móvil como jQuery Mobile, Sencha Touch, Dojo, jQTouch, SproutCore, GloveBox, XUI, iScroll, entre otros.

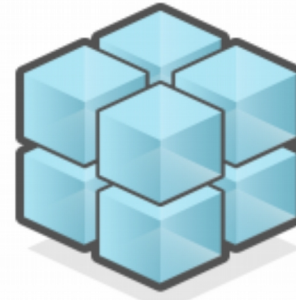
## 4.¿Por qué Cordova? - Características



Tome ventaja de **HTML5** y **CSS3**



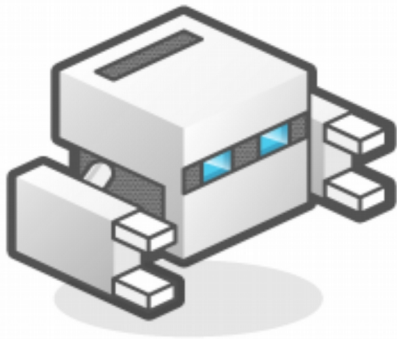
Usar **JavaScript** para escribir el código



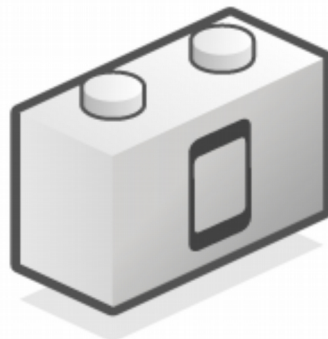
Acceda a **las funciones nativas**



Implementar la aplicación para **múltiples plataformas**



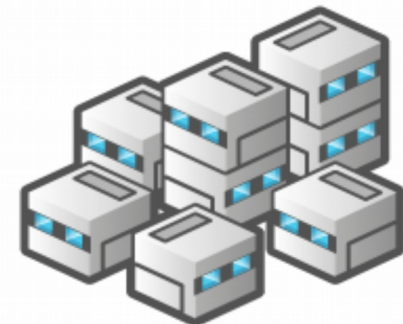
Aproveche **PhoneGap Build**



Agregar **Plugins PhoneGap** a su proyecto



Utilice **las herramientas** de la comunidad



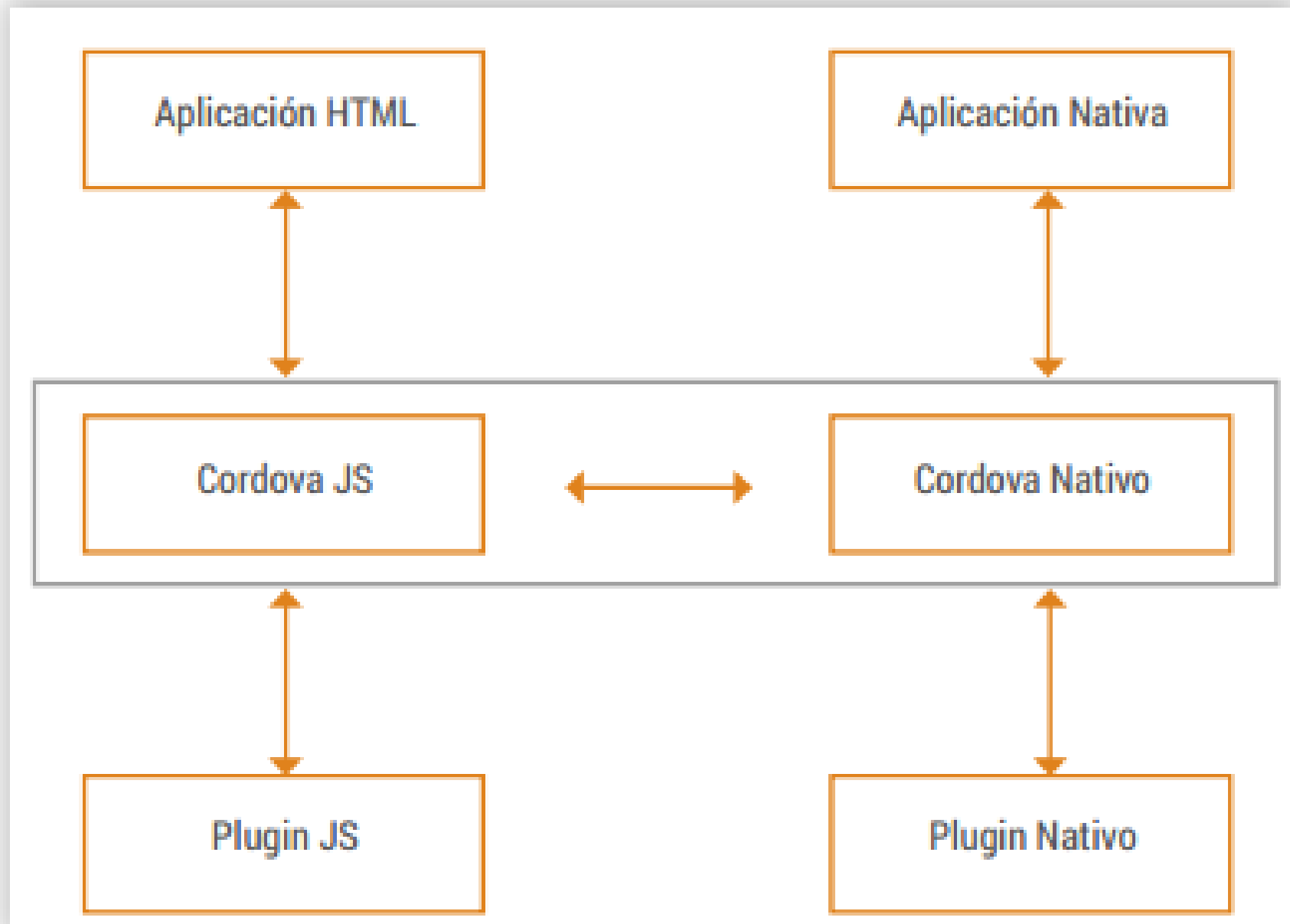
Obtenga ayuda de la **Comunidad** cada vez mayor



## 4.¿Por qué Cordova? - Arquitectura

- Tenemos por una parte el mundo HTML y por otra el mundo nativo, cada uno con sus componentes. La única vía de comunicación entre ambos mundos es a través de Cordova.
- Cordova tiene dos partes, una Javascript que reside en el mundo HTML y otra en Java (en el caso de Android) que reside en el mundo nativo.
- Toda la funcionalidad existente en PhoneGap (GPS, acelerómetro, lista de contactos,) está implementada como un plugin, con su parte Javascript y su parte nativa.

## 4.¿Por qué Cordova? - Arquitectura



## 4.¿Por qué Cordova? - Arquitectura

Podemos usar el mismo sistema para añadir nuevas funcionalidades.

Necesitaremos:

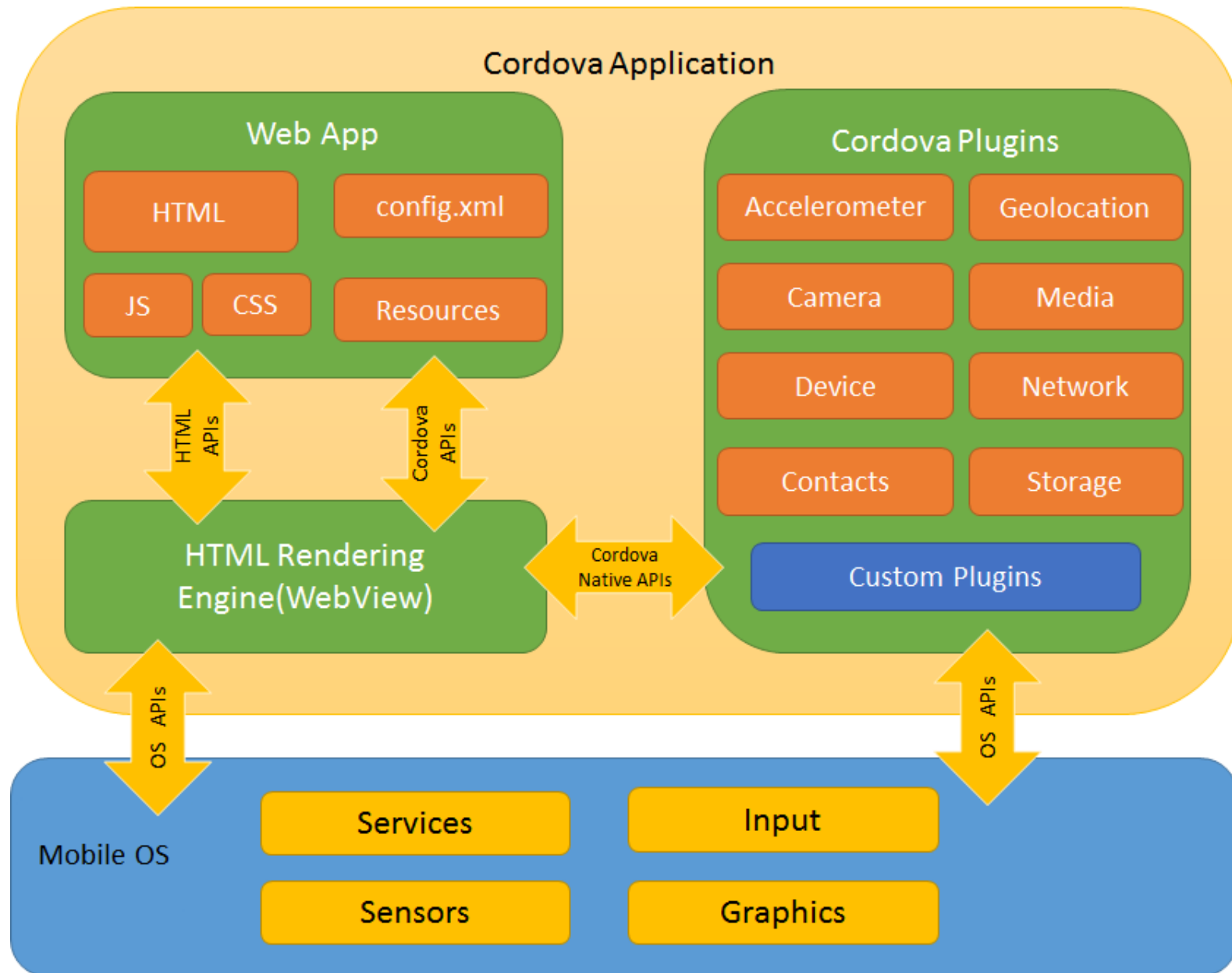
- Crear la parte Javascript del plugin, que será el interface disponible para nuestra aplicación web.
- Crear la parte nativa del plugin, que será invocada desde el código Javascript y contendrá la implementación de la funcionalidad.



## 4.¿Por qué Cordova? - Arquitectura

```
        successCallback,  
        failureCallback,  
        'GoTo',      // Nombre con el que está  
registrado el plugin Java  
        'location',  // Acción invocada  
        [address]); };  
  
cordova.addConstructor(function() {  
    cordova.addPlugin("goTo", new GoTo());  
});
```

## 4.¿Por qué Cordova? - Arquitectura



## 4.¿Por qué Cordova? - Arquitectura

Cordova permite conectar un dispositivo móvil con nuestra aplicación web mediante dos componentes principales:

### **Webview y Plugins.**

Además Cordova dispone de una serie de eventos para facilitar la comunicación entre ambas partes.

## 4.¿Por qué Cordova? - Arquitectura

### WebView

Un webview es básicamente una **instancia del navegador del dispositivo móvil**.

Con la particularidad de no tener interfaz de usuario. O sea, sin acceso a los favoritos, barra de navegación y demás características de cualquier navegador. Sólo nos quedamos con los motores de renderizado y Javascript. Al ser un componente **100% nativo, tendremos acceso directo al dispositivo**, y por lo tanto a la cámara, GPS, batería, etc.

## 4.¿Por qué Cordova? - Arquitectura

### Web App

Esta es la parte en la que reside el código de aplicación. La propia **aplicación** se implementa **como una página web**, por defecto un archivo local denominado **index.html**, que hace referencia a CSS, JavaScript, imágenes, archivos multimedia, u otros recursos son necesarios para que se ejecute. La aplicación se ejecuta en un **WebView** dentro de la envoltura de una aplicación nativa, que se distribuye a las tiendas de aplicaciones.

Este recipiente tiene un archivo muy importante, **config.xml** que proporciona información sobre la aplicación y especifica los parámetros que afectan a la forma en que funciona, por ejemplo, si responde a los cambios de orientación.

## 4.¿Por qué Cordova? - Arquitectura

### Plugins

Un plugin de Cordova es el **punto** entre nuestra aplicación web (cuyo contexto de ejecución es un WebView) y el dispositivo móvil.

Se compone de dos partes: la **implementación nativa** (ya sea en Android, iOS, etc.) y la **implementación Javascript**.

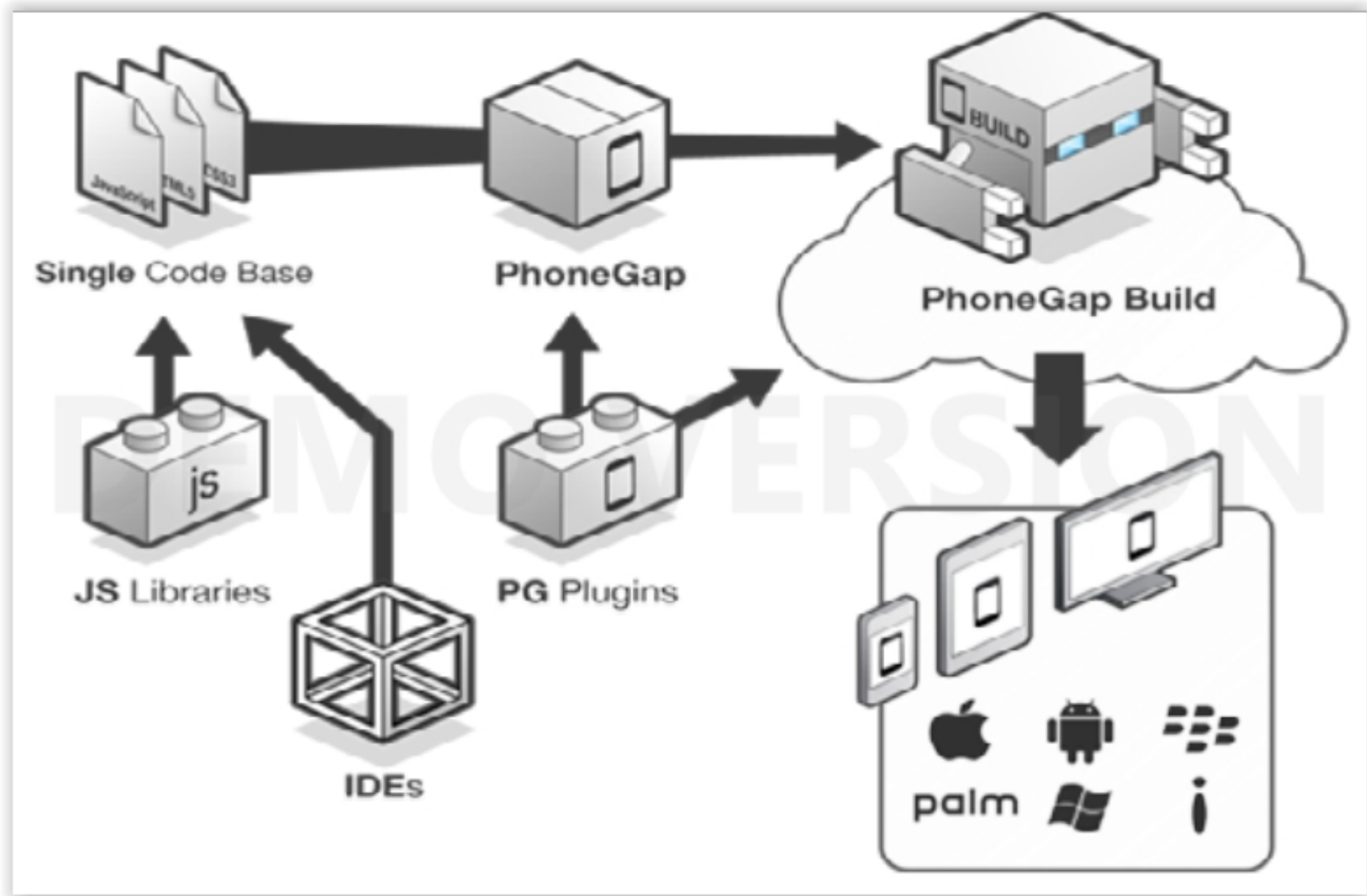
## 4.¿Por qué Cordova? - Arquitectura

### Plugins

Cordova tiene alrededor de 20 plugins oficiales. Entre los que se encuentran los más básicos: batería, cámara, lista de contactos...

Además la comunidad web ha ido desarrollando hasta el momento alrededor de **1400 plugins** para prácticamente cualquier necesidad. PlugReg tiene una buena lista de plugins organizado por plataformas. Aquí podemos encontrar plugins como barcode scanner, push notifications o social sharing.

## 4.¿Por qué Cordova? - Arquitectura





## 4.¿Por qué Cordova? - Plataformas y sistemas operativos

Es importante aclarar que Cordova sigue la normativa impuesta por los fabricantes de los dispositivos.

Si un vendedor lanza su SDK para un solo sistema operativo, tendremos que utilizar dicho sistema para construir y desplegar aplicaciones.

Cada **plataforma Cordova necesitará un SDK diferente** dependiendo del Sistema Operativo.

## 4.¿Por qué Cordova? - Plataformas y sistemas operativos

- Para aplicaciones Android podremos hacerlo en cualquiera de los S.O. de escritorio (Windows, Mac OS X o Linux).
- El SDK de BlackBerry se puede instalar en Windows o Mac OS X, pero el emulador sólo se ejecuta en Windows.
- El SDK de Windows Phone 7, lógicamente funciona sólo con Windows.
- Finalmente, para el SDK de iOS se requiere Mac OS X (y, de acuerdo con el EULA OS X, una computadora Mac también). Además para desarrollar necesitarás iOS developer certificate.

## 5. En Resumen



## 5. En Resumen

# Apps híbridas: La letra pequeña

- No utilizan UI nativo, utilizan el DOM (browser).
- Consecuencias
  - Menor desempeño.
  - Comportamiento distinto entre dispositivos.