

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
Ese apoyo que siempre quiso tener...

## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)  
Gestor de contenidos (Alfresco)  
Aplicaciones híbridas

Tareas programadas (Quartz)  
Gestor documental (Alfresco)  
Inversión de control (Spring)

Control de autenticación y  
acceso (Spring Security)  
UDDI  
Web Services  
Rest Services  
Social SSO  
SSO (Cas)

JPA-Hibernate, MyBatis  
Motor de búsqueda empresarial (Solr)  
ETL (Talend)

Dirección de Proyectos Informáticos.  
Metodologías ágiles  
Patrones de diseño  
TDD

BPM (jBPM o Bonita)  
Generación de informes (JasperReport)  
ESB (Open ESB)

AdictosAlTrabajo

Terrakas 1x03  
¡¡Ya está en la web!!  
terrakas.com

Entra en Adictos a través de



E-mail

Contraseña

Entrar

Deseo registrarme  
Olvidé mi contraseña
[Inicio](#)
[Quiénes somos](#)
[Formación](#)
[Comparador de salarios](#)
[Nuestro libro](#)
» Estás en: [Inicio](#) [Tutoriales](#) Introducción a PhoneGap

Francisco Javier Martínez Páez

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Telecomunicaciones

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)Catálogo de servicios  
Autentia

Fecha de publicación del tutorial: 2009-02-26

Tutorial visitado 5 veces [Descargar en PDF](#)

## Introducción a PhoneGap

## Introducción

El **desarrollo para aplicaciones móviles** está sufriendo estos últimos años el mismo problema que existía en el desarrollo de aplicaciones de escritorio antes de la aparición de la plataforma Java (concepto de máquina virtual); esto es, que para hacer que una misma aplicación pudiera "correr" en distintos sistema operativos (imagínalos un juego) era necesario **un gran esfuerzo económico** debido a las **diferencias** entre lenguajes, dispositivos etc... La situación en las aplicaciones móviles hoy día es similar. Para hacer una aplicación que básicamente funcione en Android y Iphone, ya necesito conocer dos lenguajes de programación diferentes, con entornos de desarrollo diferentes, con sistemas operativos diferentes, con posibilidades de experiencia de usuario diferentes, etc... Esto supone que **el coste de desarrollo, mantenimiento y evolución** de una misma aplicación se vea casi **multiplicado** por el **número** de diferentes **terminales** a los que se quiere llegar: Android, iOS, Windows Mobile, Blackberry, Palm, Symbian ..., vamos, la torre de babel.

Sin embargo, progresivamente en estos últimos años se han alineado varios planetas para dar lugar a una arquitectura que puede ordenar el caos que existe en este momento. Esos "planetas" son:

- **HTML5, CSS3 y JavaScript** como plataforma base de desarrollo de las aplicaciones móviles
- **Servicios Web** (ya sean SOAP o Restful) como mecanismo de comunicación entre la aplicación cliente y el servidor
- La posibilidad de usar el **navegador ("browser") embebido** en la aplicación móvil (WebView) como plataforma de ejecución de nuestra aplicación HTML5/CSS3/JavaScript. Desde el navegador embebido ya es posible acceder a través de JavaScript a las librerías nativas que permiten el acceso a los distintos dispositivos del terminal (acelerómetro, cámara, GPS, etc...), ya que todas las plataformas lo permiten (evidentemente, no todos los terminales disponen de todos esos dispositivos)

Alguno me dirá: "Paco, pero ya sería la caña que encima todas estas APIs o interfaces de acceso a los dispositivos fuera la misma". Pues sí, sería "la caña", pero no es así, sin embargo ahí quería yo llegar. Este es justo el "hueco" que PhoneGap viene a rellenar.

## ¿ Qué es PhoneGAP ?

Es un **framework** para desarrollar **aplicaciones nativas** usando **HTML, CSS y Javascript** (sí, sí, sin usar Java, ni ObjectiveC, ni C++, ni C#, ni...). Las aplicaciones construidas usando este framework serán mitad nativas, mitad HTML5/JavaScript/CSS (Lo que se ha venido a llamar aplicaciones híbridas). La parte **nativa** será la que **accede a las** capacidades del terminal, mientras que el **interfaz de usuario, la lógica de aplicación y la comunicación con el servidor** será la otra mitad. Y **PhoneGap** proporciona el **acceso "común"** a la parte **nativa** mediante un **API Javascript**, haciendo **transparente** al programador la parte nativa. Con esto cerraríamos el círculo, bueno casi, ya que aún sigue habiendo un pero. Algunos Smartphones/Tablets **no** disponen aún de la suficiente **capacidad** para soportar las necesidades de estas **aplicaciones híbridas** (hablo de memoria, disco, procesador...). Pero esto es sólo cuestión de tiempo (he tenido un "deja vú"... no os suena esto a cuando salió Java... que si es lento, que si C++, que si tal, que si JIT Compiler, que si dos procesadores, que si cuatro, que si más memoria, que si... es decir, el tiempo, uno, dos, tres años, quién sabe...).

## Empezando con PhoneGAP en Android

Para poder hacer los ejemplos, por ahora necesitamos:

1. Eclipse >= 3.4
2. JDK 1.6
3. Android SDK
4. Plugin Android ADT para Eclipse (si no lo tenéis instalado, podéis hacerlo desde el eclipse en Help > Install New Software desde el repositorio: "https://dl-ssl.google.com/android/eclipse"

Os muestro en la imagen el Android SDK Manager para que veáis las versiones que tengo instaladas:

## Siguenos a través de:



## Últimas Noticias

» ¿Por qué una empresa puede dar unos grandes beneficios y tener que despedir gente?

» Técnico en preventa: entre técnico y comercial

» Rotación en la empresa y el teorema de la niña mona

» Dime que manejas y te diré lo que ganas

» Socios para mi negocio ¿impar y menor de 2?

[Historico de noticias](#)

## Últimos Tutoriales

» Spring mvc. Servicios Rest respondiendo en Json o XML

» Trabajando con IntelliJ IDEA 11

» JSTestDriver: testea tu código Javascript

» Implantación de metodologías Ágiles en grandes organizaciones

» Tapestry frente a JSF

## Últimos Tutoriales del

Impulsores Comunidad ¿Ayuda?

0 personas han traído clicks a esta página

sin clicks + + + + + + + +

powered by karmacracy

Android SDK Manager

SDK Path: /Users/fjmpaez/android-sdks/

Packages

Name	API	Rev.	Status
Tools			
Android SDK Tools		17	Installed
Android SDK Platform-tools		11	Installed
Android 4.0.3 (API 15)			
Documentation for Android SDK	15	2	Installed
SDK Platform	15	3	Installed
Samples for SDK	15	2	Installed
ARM EABI v7a System Image	15	1	Installed
Google APIs	15	1	Not installed
Sources for Android SDK	15	1	Installed
Android 4.0 (API 14)			
SDK Platform	14	3	Installed
Samples for SDK	14	2	Installed
ARM EABI v7a System Image	14	2	Installed
Google APIs	14	2	Not installed
Sources for Android SDK	14	1	Installed
Android 3.2 (API 13)			
SDK Platform	13	1	Installed
Samples for SDK	13	1	Installed
Google APIs	13	1	Not installed
XOOM2ME	13	1	Not installed
XOOM2	13	1	Not installed
Android 3.1 (API 12)			
SDK Platform	12	3	Installed
Samples for SDK	12	1	Installed
Google APIs	12	1	Not installed
Google TV Addon	12	2	Not compatible with Mac OS X
OpenSense SDK for Tablets	12	2	Not installed
Android 3.0 (API 11)			
SDK Platform	11	2	Installed
Samples for SDK	11	1	Installed
Google APIs	11	1	Not installed
XOOM	11	2	Not installed
Android 2.3.3 (API 10)			
SDK Platform	10	2	Installed
Samples for SDK	10	1	Installed
Google APIs	10	2	Installed
OpenSense SDK for Phones	10	2	Not installed
Intel Atom x86 System Image	10	1	Not installed
Dual Screen APIs	10	1	Not installed
Real3D	10	2	Not installed
ADMIRAL	10	5	Not installed
ATRIX2	10	1	Not installed
Bionic	10	2	Not installed
defy+	10	1	Not installed
Droid4	10	1	Not installed
DroidRAZR	10	4	Not installed
MotorolaPro+	10	2	Not installed
MT870	10	2	Not installed
MT917	10	1	Not installed
PHOTON	10	1	Not installed
XT882	10	2	Not installed
XT928	10	1	Not installed
EDK 2.0	10	1	Not installed
Android 2.2 (API 8)			
SDK Platform	8	3	Installed
Samples for SDK	8	1	Installed

## Autor

- » JBossESB y JBDS
- » Apache Hadoop - HDFS
- » Apache Hadoop-MapReduce
- » Apache Camel, primeros pasos
- » Patrón Intérprete

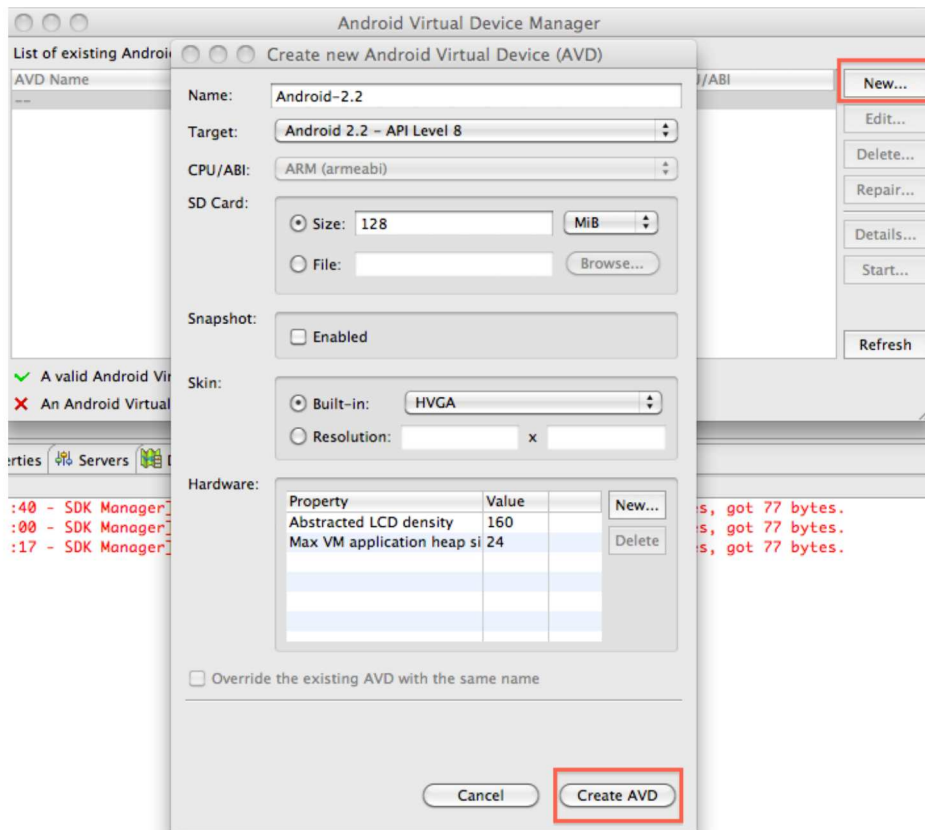
## Últimas ofertas de empleo

- 2011-09-08  
Comercial - Ventas - MADRID.
- 2011-09-03  
Comercial - Ventas - VALENCIA.
- 2011-08-19  
Comercial - Compras - ALICANTE.
- 2011-07-12  
Otras Sin catalogar - MADRID.
- 2011-07-06  
Otras Sin catalogar - LUGO.

Como se supone que tenemos instalado el plug-in ADT, tendréis dos iconos relacionados con el plugin Android ADT:

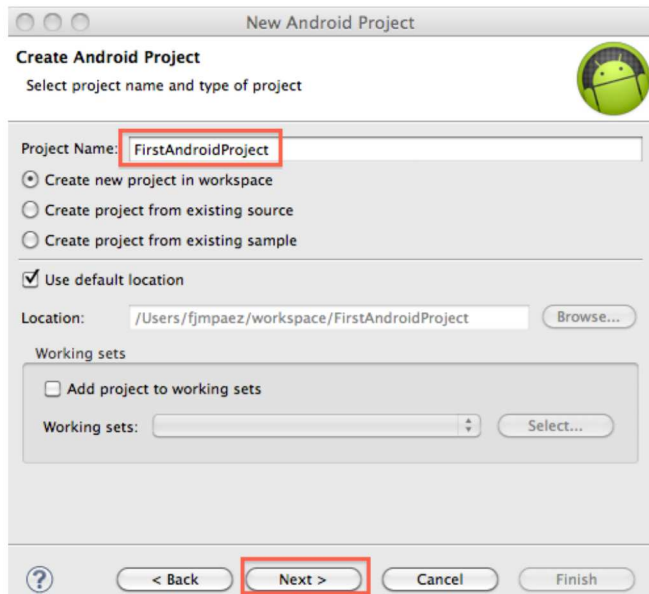


Pulsad sobre el recuadrado en rojo para añadir un nuevo dispositivo virtual (un simulador) y pulsad "New..." para crear un AVD como el mostrado:

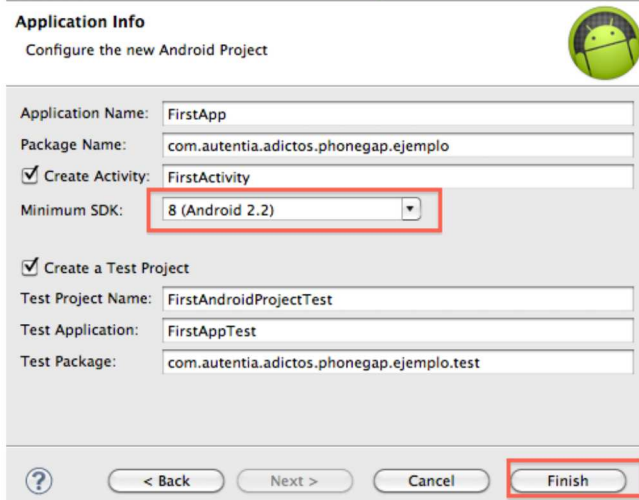
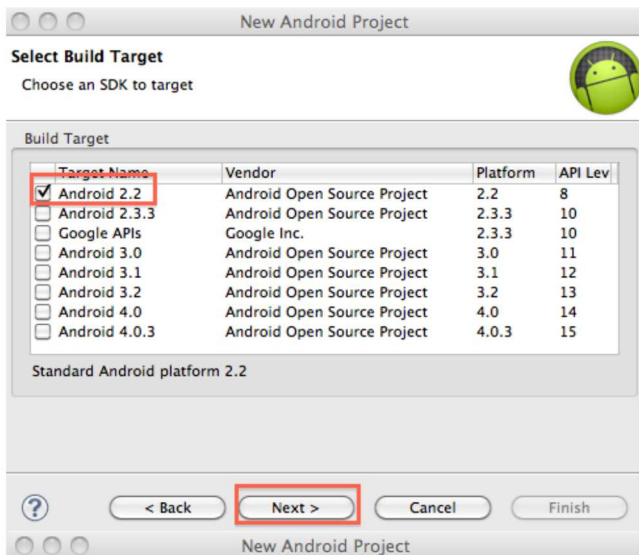


Ahora únicamente nos falta descargarnos el SDK de PhoneGap. Lo haremos desde [el sitio de PhoneGap](#). Pulsad en el icono Download PhoneGap 1.5.0 (versión del momento). Descomprimid el zip en algún lugar de vuestra máquina.

Ahora crearemos un nuevo proyecto en Eclipse: File > New Project > Android Project:

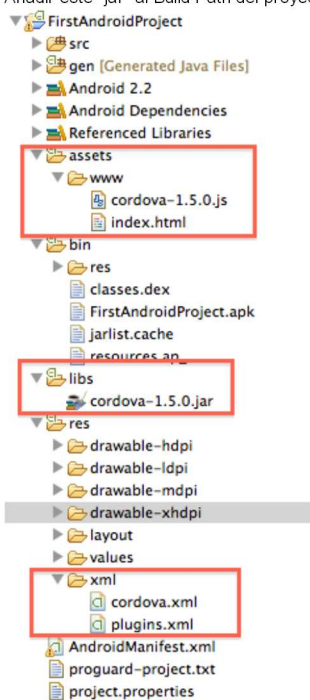






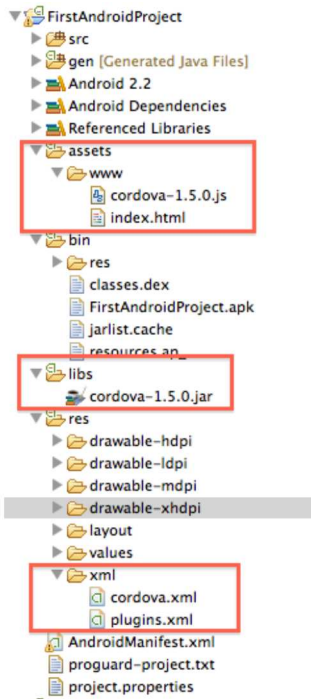
Ahora debemos añadir las librerías de PhoneGap al proyecto que acabamos de crear:

- Cread un directorio "libs" en el proyecto y copiad en él el fichero: "cordova-1.5.0.jar" que se encuentra en "lib > android" de la distribución de PhoneGap que habéis descomprimido.
- Añadir este "jar" al Build Path del proyecto:



- Cread un directorio llamado "assets" y dentro otro llamado "www" en el proyecto. En este subdirectorio "www" copiad el fichero "cordova-1.5.0.js" que se encuentra en "lib > android" de la distribución de PhoneGap que habéis descomprimido.
- Copiad el directorio "xml" que se encuentra en "lib > android" de la distribución de PhoneGap que habéis descomprimido, en el directorio "res" del proyecto.

Os debe quedar así:



Con estos pasos ya tenemos instalado lo necesario para poder usar PhoneGap. Ahora tan sólo nos faltan algunos detalles de configuración. Editad el fichero "AndroidManifest.xml" para:

- Añadir los permisos necesarios (todos los que necesita PhoneGap. Si nuestra aplicación no necesita ciertos elementos, podemos quitarlos.). Yo voy a darle todos los permisos
- Definir los tamaños de pantalla que soporta mi aplicación (todo que si)
- Añadir a la actividad la configuración necesaria para indicar que no es necesario recrear la actividad cuando rotamos la pantalla.
- Añadir una segunda actividad necesaria para configurar PhoneGap. En Android, una actividad representa una pantalla

Finalmente, el fichero os ha de quedar:

```
view plain print ?
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.autentia.adictos.phonegap.ejemplo"
    android:versionCode="1"
    android:versionName="1.0" >

    <!-- Tamaños de pantalla soportados -->
    <supports-screens
        android:anyDensity="true"
        android:largeScreens="true"
        android:normalScreens="true"
        android:resizeable="true"
        android:smallScreens="true" />

    <!-- Permisos de la aplicación -->
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.BROADCAST_STICKY" />

    <uses-sdk android:minSdkVersion="8" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".FirstActivity"
            android:configChanges="orientation|keyboardHidden"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!-- Segunda actividad -->
        <activity
            android:name="org.apache.cordova.DroidGap"
            android:configChanges="orientation|keyboardHidden"
            android:label="@string/app_name" >
            <intent-filter>
                </intent-filter>
        </activity>
```

```
</application>

</manifest>
```

Ahora cambiaremos la actividad principal para que extienda de DroidGap (DroidGap es una actividad que permite mostrar HTML en nuestra aplicación) y cargue en el arranque nuestra página de inicio (después la crearemos). Para hacer eso, la clase FirstActivity debe quedar:

```
view plain print ?
package com.autentia.adictos.phonegap.ejemplo;

import org.apache.cordova.DroidGap;

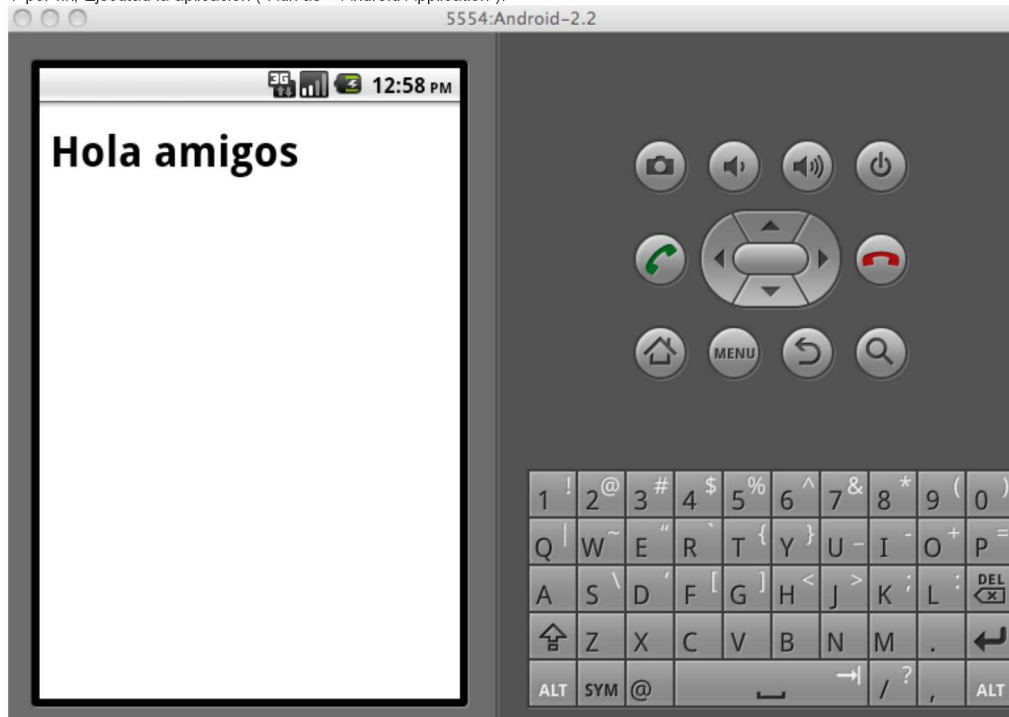
import android.os.Bundle;

public class FirstActivity extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

Cread en el directorio "assets/www" el fichero "index.html":

```
view plain print ?
<!DOCTYPE HTML>
<html>
<head>
<title>PhoneGap</title>
<script type="text/javascript" charset="utf-8" src="cordova-1.5.0.js"></script>
</head>
<body>
<h1>Hola amigos</h1>
</body>
</html>
```

Y por fin, Ejecutad la aplicación ("Run as > Android Application"):



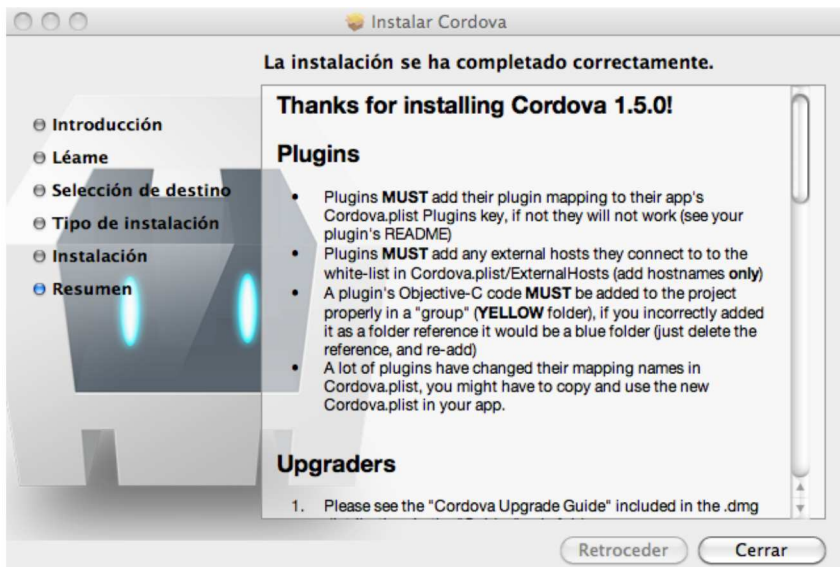
## Empezando con PhoneGAP en iOS

Para esto necesitamos:

- Un Mac... con esto no puedo ayudaros. Yo un Mac Book Pro con Mac OS X 10.6.8
- Xcode instalado (yo tengo la versión 3.2.3). Se puede descargar de <http://developer.apple.com/xcode/index.html>, pero necesitáis una cuenta de iOS Developer.

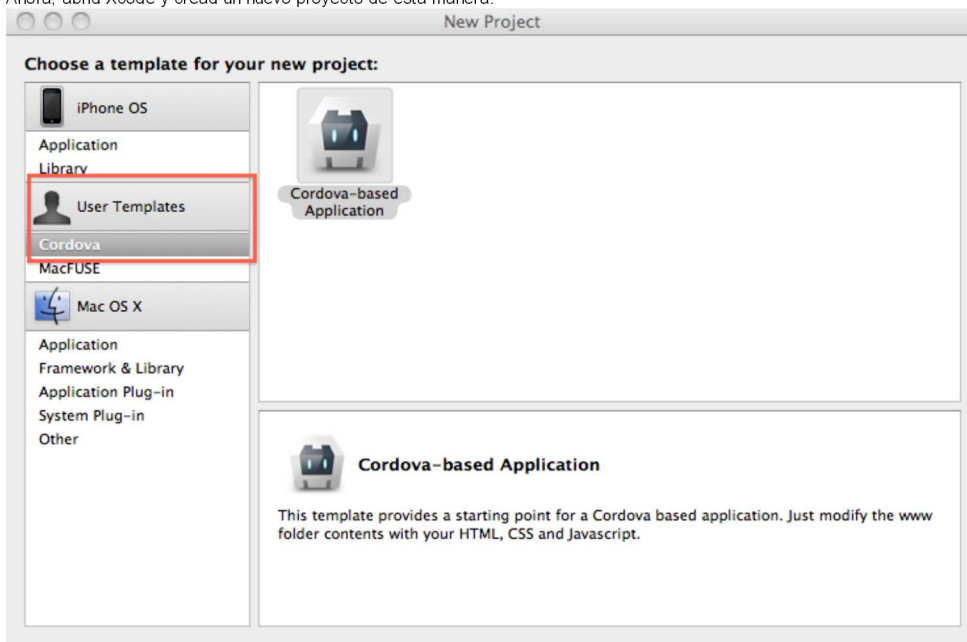
Si cumplís con los requisitos, entonces podemos empezar:

Primero, id al directorio "lib/ios" de la distribución de PhoneGap que habéis descomprimido e instalad PhoneGap:



Durante la propia instalación te dice como crear un proyecto de PhoneGap para Xcode 3 y Xcode 4. Nosotros lo vamos a hacer para Xcode 3 (tened en cuenta esto por si lo hacéis para Xcode 4).

Ahora, abrid Xcode y cread un nuevo proyecto de esta manera:

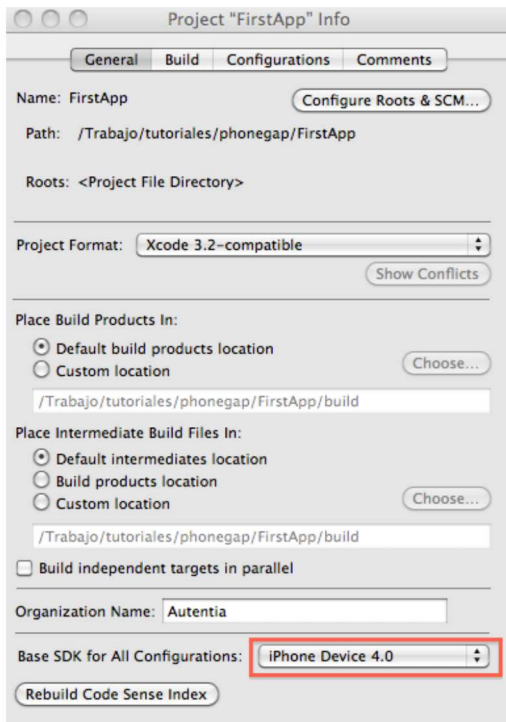


Pulsad "Choose...", dadle un nombre y guardadla en algún sitio:

Os creará una aplicación por defecto.

Si en el combo situado en la parte superior izquierda de la ventana os saliera: "Base SDK missing", entonces seleccionad: "Project > Edit Project Settings" y comprobad que tenéis correctamente seleccionado "Base SDK for ...":





Ahora pulsad sobre "Build and Run". A mi al compilar me ha dado dos errores en el recurso "CDVLocation.m". El primero se corrige:

```
view plain print ?
...
- (BOOL) isAuthorized
{
    BOOL authorizationStatusClassPropertyAvailable = [CLLocationManager respondsToSelector:@selector(authorizationStatusClassPropertyAvailable)];
    if (authorizationStatusClassPropertyAvailable)
    {
        NSUInteger authStatus = [CLLocationManager authorizationStatus];
        // COMENTAD LA LÍNEA
        // return (authStatus == kCLErrorAuthorizationStatusAuthorized) || (authStatus == kCLErrorAuthorizationStatusNotDetermined);
    }

    // by default, assume YES (for iOS < 4.2)
    return YES;
}
...
```

El segundo se corrige cambiando el nombre "NSEC\_PER\_MSEC" a "NSEC\_PER\_SEC".

Después, cambiad el index.html por el que creamos en el Eclipse. Pulsamos de nuevo sobre "Build and Run" y:

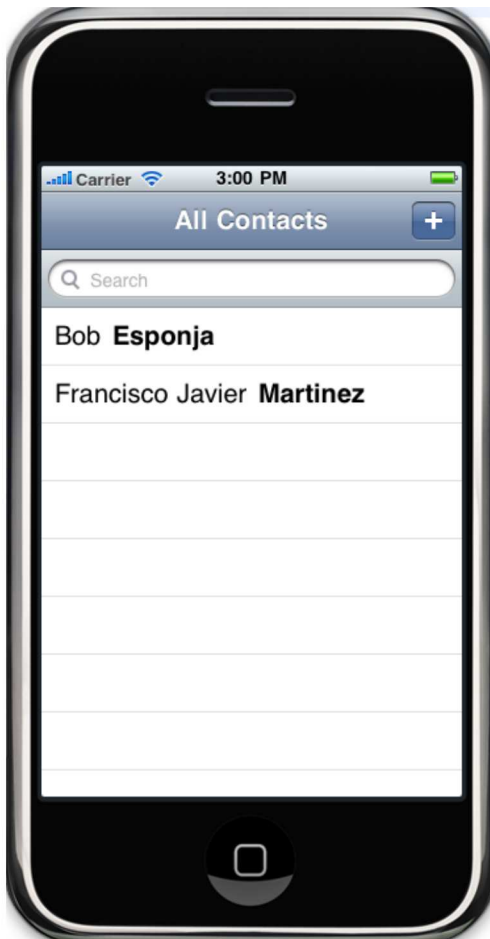


### Accediendo con PhoneGap a "cosicas" del móvil.

Para terminar el tutorial, vamos a añadir a nuestra aplicación la posibilidad de mostrar los datos de la agenda (algo que podamos probar en los simuladores).

Primeramente, en los emuladores, añadid algunos contactos:

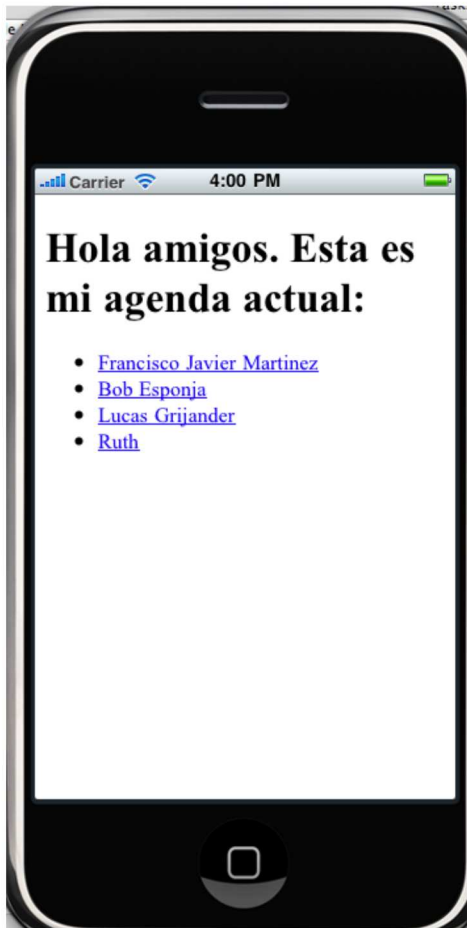
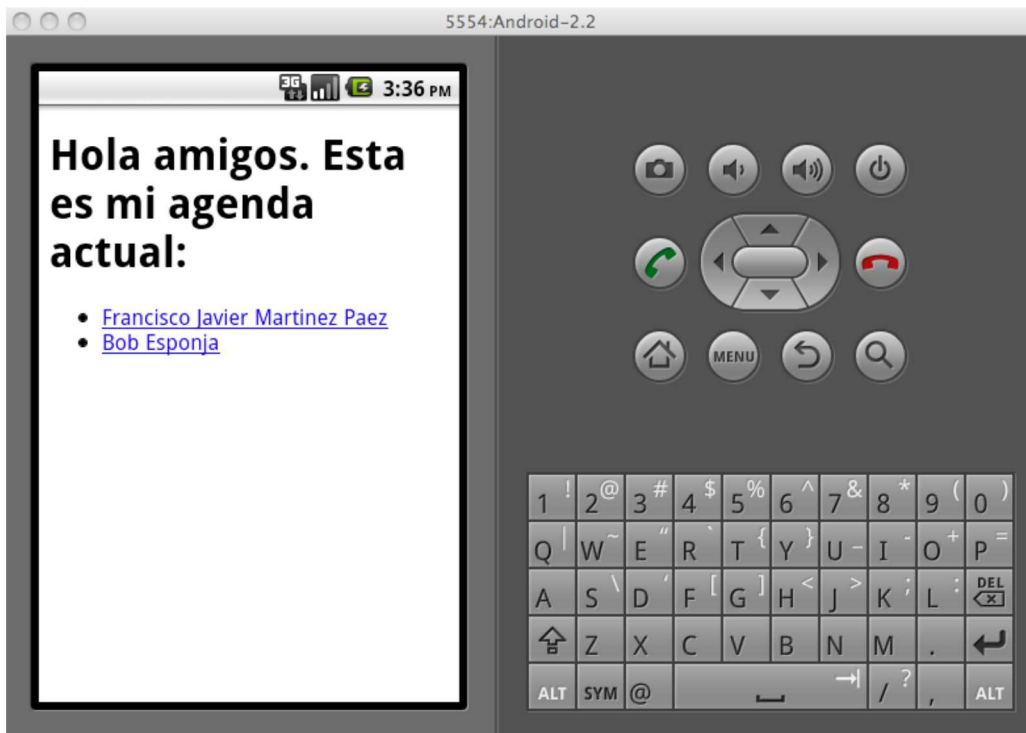




Ahora modificad el index.html de esta manera:

```
view plain print ?
<!DOCTYPE HTML>
<html>
<head>
<title>PhoneGap</title>
<script type="text/javascript" charset="utf-8" src="cordova-1.5.0.js"></script>
<script type="text/javascript">
function onDeviceReady(){
    var options = new ContactFindOptions();
    options.filter="";
    options.multiple=true;
    var fields = ["phoneNumbers", "name"];
    navigator.contacts.find(fields, onSuccess, onError, options);
}
function onSuccess(contacts) {
    var ul = document.getElementById("contactList");
    for(var index=0;index<contacts.length;index++) {
        var name = contacts[index].name.formatted;
        var phoneNumber = contacts[index].phoneNumbers[0].value;
        var li = document.createElement('li');
        li.innerHTML = "<a href='tel:/'"+phoneNumber+"\">"+name+"</a>";
        ul.appendChild(li);
    }
};
function onError() {
    alert('onError!');
};
function init(){
    document.addEventListener("deviceready", onDeviceReady, false);
}
</script>
</head>
<body onload="init()">
    <h1>Hola amigos. Esta es mi agenda actual:</h1>
    <ul id="contactList"></ul>
</body>
</html>
```

Con una única línea de código JavaScript obtenemos los contactos: "navigator.contacts.find(fields, onSuccess, onError, options);"  
Volved a compilar... y ejecutad:



## Conclusiones

En otros tutoriales seguiremos explorando las capacidades de PhoneGap. Por ahora es suficiente para comprobar que si parece rellenar "el hueco" del que hicimos mención en la introducción. No obstante, parece vislumbrarse que el **futuro** de las **aplicaciones híbridas** va por este camino marcado por el **HTML5** (no me refiero sólo a las capacidades de la nueva versión del HTML sino a la conjunción de HTML, CSS y JavaScript) junto con este tipo de soluciones que permiten trabajar de manera transparente a la hora de acceder a las capacidades del dispositivo.

**A continuación puedes evaluarlo:**

[Regístrate para evaluarlo](#)

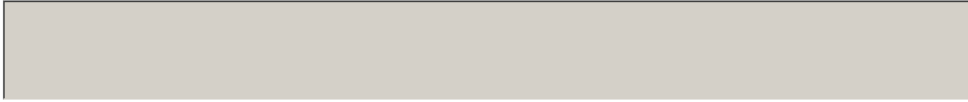


Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

0

Animate y coméntanos lo que pienses sobre este **TUTORIAL**:



» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Copyright 2003-2012 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

W3C XHTML 1.0

W3C CSS

XML RSS

XML RICH