***“Plataforma Móvil para el Sistema Integrado de Información Sanitaria Argentino”***

Integrantes

|  |  |  |
| --- | --- | --- |
| Legajo | Nombre | E-Mail |
| 138.373-5 | Santiago Panizza | panizzasantiago@gmail.com |
| 134.775-5 | Lucas Roca | lucasr28@gmail.com |
| 134.853-0 | Nicolás Chaikh | nicolaschaikh@gmail.com |
| 104698-6 | Sebastián Palotte | spalotte@gmail.com |
| 248.593-0 | Leonardo Oneto | leojava2007@gmail.com |

***Director de Cátedra****: Lic. Carlos Tomassino*

***Profesor a cargo del curso****:* [*Ing.*](http://www.campusvirtual.frba.utn.edu.ar/especialidad/user/view.php?id=35051&course=1) *Claudio Crescentini*

***Profesor a cargo del proyecto****: Ing. Federico Casuscelli; Nicolás Rodríguez*

***Controller:*** *Ing. Gabriela Salem*

**Historial de Revisión**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 23/05/2014 | 1.0 | Versión inicial | Leonardo Oneto |
| 25/11/2014 | 1.1 | Se agrega plataforma Windows Phone 8 y Blackberry 10 | Leonardo Oneto |
| 10/03/2015 | 1.2 | Se completa el documento y se realizan correcciones. | Leonardo Oneto |
| 12/03/2015 | 1.3 | Se completa la información para el entorno de desarrollo de BlackBerry10. Agregada la sección de herramientas complementarias. | Santiago Panizza |
| 17/03/2015 |  | Se complementa la información sobre las bibliotecas de código utilizadas. | Leonardo Oneto |

Contenido

[Objetivo 4](#_Toc414316952)

[PhoneGap 4](#_Toc414316953)

[Configuración de CLI, interfaz por línea de comandos de Cordova 4](#_Toc414316954)

[Android 5](#_Toc414316955)

[Android ADT 5](#_Toc414316956)

[Instalar ANT 5](#_Toc414316957)

[Crear un proyecto de prueba 6](#_Toc414316958)

[Instalar APK 6](#_Toc414316959)

[Windows Phone 8 8](#_Toc414316960)

[Crear un proyecto de prueba 9](#_Toc414316961)

[Instalar .XAP 9](#_Toc414316962)

[Blackberry 10 12](#_Toc414316963)

[Crear un proyecto de prueba 12](#_Toc414316964)

[Preparar el dispositivo para probar la aplicación 13](#_Toc414316965)

[Instalar .BAR 14](#_Toc414316966)

[Instalación JDK 15](#_Toc414316967)

[Herramientas complementarias utilizadas en el desarrollo 16](#_Toc414316968)

[Backbone.js 16](#_Toc414316969)

[Intel App Framework 16](#_Toc414316970)

[Ripple emulator 16](#_Toc414316971)

[Weinre 16](#_Toc414316972)

[BlackBerry Web Inspector 16](#_Toc414316973)

Objetivo

El objetivo del presente documento es el de describir los pasos necesarios para configurar el entorno de desarrollo en las plataformas Android, Windows Phone 8 y Blackberry 10.

PhoneGap

La documentación generada para configurar los entornos fue tomada del sitio oficial de [Phonegap](http://docs.phonegap.com/en/3.5.0/index.html). Al momento de generar este documento utilizamos la versión 3.5 del framework.

## Configuración de CLI, interfaz por línea de comandos de Cordova

1. Instalar Node.js ([descarga](http://nodejs.org/download/)). Una vez completada la instalación verificar que se pueda ejecutar el comando *node o npm* desde el símbolo del sistema.
   1. En caso que el comando falle, buscar en la carpeta de instalación de Node.js el archivo *nodevars.bat,* que agrega la configuración necesaria a las variables de entorno para que funcione el comando.
2. Instalar un cliente Git para Windows, que es utilizado por la herramienta CLI para descargar archivos cuando se crea un nuevo proyecto. Por ejemplo, podemos utilizar el cliente de GitHub que se instala desde [aquí](https://windows.github.com/).
3. Instalar la utilidad Cordova. Se realiza mediante el siguiente comando ejecutado desde el símbolo del sistema:

npm install –g cordova

Se debería mostrar por la consola el resultado de descargar una serie de archivos, algo similar a esto:

C:\>npm install -g cordova

npm http GET https://registry.npmjs.org/cordova

npm http 200 https://registry.npmjs.org/cordova

npm http GET https://registry.npmjs.org/cordova/-/cordova-3.5.0-0.2.4.tgz

npm http 200 https://registry.npmjs.org/cordova/-/cordova-3.5.0-0.2.4.tgz

npm http GET <https://registry.npmjs.org/cordova-lib>

…………

C:\Users\PhoneGap\AppData\Roaming\npm\cordova -> C:\Users\PhoneGap \AppData\Roaming\npm\nod

e\_modules\cordova\bin\cordova

cordova@3.5.0-0.2.4 C:\Users\PhoneGap\AppData\Roaming\npm\node\_modules\cordova

├── underscore@1.4.4

├── q@0.9.7

├── optimist@0.6.0 (wordwrap@0.0.2, minimist@0.0.10)

└── cordova-lib@0.21.3 (osenv@0.0.3, bplist-parser@0.0.5, mime@1.2.11, semver@2.

0.11, shelljs@0.1.4, dep-graph@1.1.0, plist-with-patches@0.5.1, xcode@0.6.6, rc@

0.3.0, glob@3.2.9, elementtree@0.1.5, npmconf@0.1.16, tar@0.1.19, request@2.22.0

, npm@1.3.4)

C:\ >

Android

## Android ADT

1. Instalar Android SDK ([descarga](http://developer.android.com/sdk/)). Previamente haber instalado JDK. Ver [instalación JDK](#_Instalación_JDK).
2. Para que funcione la herramienta de línea de comandos de Cordova, es necesario incluir los directorios tools y platform-tools de la SDK a la variable de entorno PATH.
   1. Clic en el menú **Inicio, Equipo** y luego **Propiedades**.
   2. Seleccionar **Configuración Avanzada del Sistema**.
   3. En el cuadro de dialogo que se abre, presionar **Variables de Entorno**.
   4. Seleccionar la variable **Path** y presionar **Editar**.
   5. Agregar lo siguiente a la variable dependiendo de dónde hayas instalado la SDK de Android en el paso 1. Por ejemplo:

;C:\Development\adt-bundle-windows-x86\_64-20140321\sdk\platform-tools; C:\Development\adt-bundle-windows-x86\_64-20140321\sdk\tools

## Instalar ANT

1. Al construir aplicaciones para plataforma Android, ya sea modo de depuración o modo *release*, es necesario utilizar la herramienta Ant para compilar y construir su proyecto.
   1. Instalar Ant. Consiste en descomprimir los archivos dentro del .zip en una carpeta deseada en nuestro sistema ([descarga](http://ant.apache.org/bindownload.cgi))
      1. **Nota:** Es importante tener en cuenta que la carpeta donde situemos los archivos de Ant no tenga espacios, porque puede fallar al ejecutarse cuando se realiza la compilación. En caso de haberlo instalado en “Program Files” o “Archivos de Programa”, se debería poner “Program~1” en la variable de entorno ANT\_HOME ([Ver detalle](http://developer.android.com/tools/building/building-cmdline.html)).
   2. Por símbolo de sistema setear la variable de entorno:

set ANT\_HOME=C:\ant

set PATH=%PATH%;%ANT\_HOME%\bin

* 1. Verificamos la configuración con el siguiente comando: ant - version. Si nos muestra un resultado similar a este:

C:\Users\PhoneGap>ant -version

Apache Ant(TM) version 1.9.4 compiled on April 29 2014

Entonces quiere decir que se encuentra configurado correctamente. Si no podríamos ver un mensaje como el siguiente:

C:\Users\PhoneGap>ant -version

'ant' is not recognized as an internal or external command, operable program or batch file.

* 1. De ser así deberíamos revisar la configuración de las variables de entorno, y reiniciar la PC para asegurarnos de que los cambios hayan tenido efecto. [Ver más](http://madhukaudantha.blogspot.com.ar/2010/06/installing-ant-for-windows-7.html)

## Crear un proyecto de prueba

Creamos un proyecto de prueba para verificar la correcta instalación y configuración del entorno.

Desde el símbolo del sistema utilizamos los comandos proporcionados por la herramienta CLI de Cordova. Estos comandos se deben ejecutar dentro del directorio donde situemos nuestros archivos fuentes.

C:\>cd C:\PhoneGap\Projects\

C:\PhoneGap\Projects>cordova create hello com.example.hello HelloWorld

C:\PhoneGap\Projects>cd hello

C:\PhoneGap\Projects\hello>cordova platform add android

C:\PhoneGap\Projects\hello>cordova build android

Al finalizar se muestra un mensaje de compilación correcta.

BUILD SUCCESSFUL

Total time: 1 minute 1 second

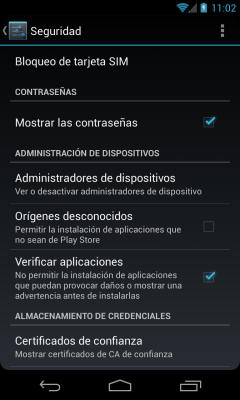
## Instalar APK

Describiremos como instalar la aplicación solamente para hacer debug y testing, utilizando directamente el archivo generado mediante la compilación.

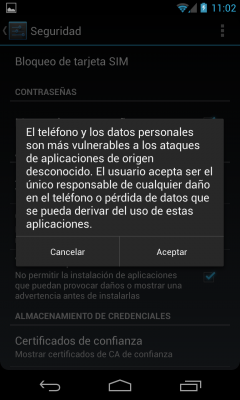
Después de hacer el build para la plataforma Android el resultado obtenido es un archivo .apk que se encuentra dentro de la carpeta de la aplicación, en /platforms/android/ant-build. Generará tres archivos, uno terminado en \*unaligned.apk, este lo obviamos porque es un producto intermedio de la compilación. El que nos importa ahora es el terminado \*debug.apk

Previamente, para poder instalar esta aplicación habrá que configurar el teléfono para que acepte aplicaciones no firmadas. No es algo que hiciéramos normalmente por cuestiones de seguridad, pero en este caso sabemos que la aplicación que instalamos no representa un riesgo.

Para permitir a nuestro Android la instalación de aplicaciones procedentes de orígenes desconocidos nos dirigiremos a: **Ajustes -> Seguridad -> Seleccionar Orígenes Desconocidos.**

****

Cuando activemos Orígenes Desconocidos nos saldrá un mensaje confirmando si estamos seguros de hacer esto con el siguiente mensaje: *El teléfono y los datos personales son más vulnerables a los ataques de aplicaciones de origen desconocido. El usuario acepta ser el único responsable de cualquier daño en el teléfono o pérdida de datos que se pueda derivar del uso de estas aplicaciones.*



Si queremos seguir adelante haremos click en “Aceptar” y ya podremos instalar aplicaciones de origen desconocido.

Después, la instalación consistirá en transferir el .apk generado mediante USB al dispositivo, buscar el archivo mediante el explorador de archivos que tengamos instalado y hacer clic sobre el mismo.

# Windows Phone 8

Como requisito es necesario contar con una máquina con Windows 8 Pro de 64 bits

Tendremos que instalar el entorno de desarrollo [Windows Phone SDK](http://dev.windows.com/en-us/develop/download-phone-sdk) (seguir el link)

Es simplemente descargar el instalador y ejecutarlo. Esto te instala el Visual Studio 2012 Express y las bibliotecas de código necesarias para el desarrollo de aplicaciones Mobile. Este proceso de instalación puede demorar bastante tiempo dependiendo del procesador y de la memoria disponible. En si no requiere de atención se realiza en forma automática.

La forma de instalación de la interfaz de comandos de Cordova es la misma que para Android. Podríamos decir que es independiente de la plataforma. Podemos remitirnos a la [sección](#_Configuración_de_CLI,_1) correspondiente para hacerlo.

## Crear un proyecto de prueba

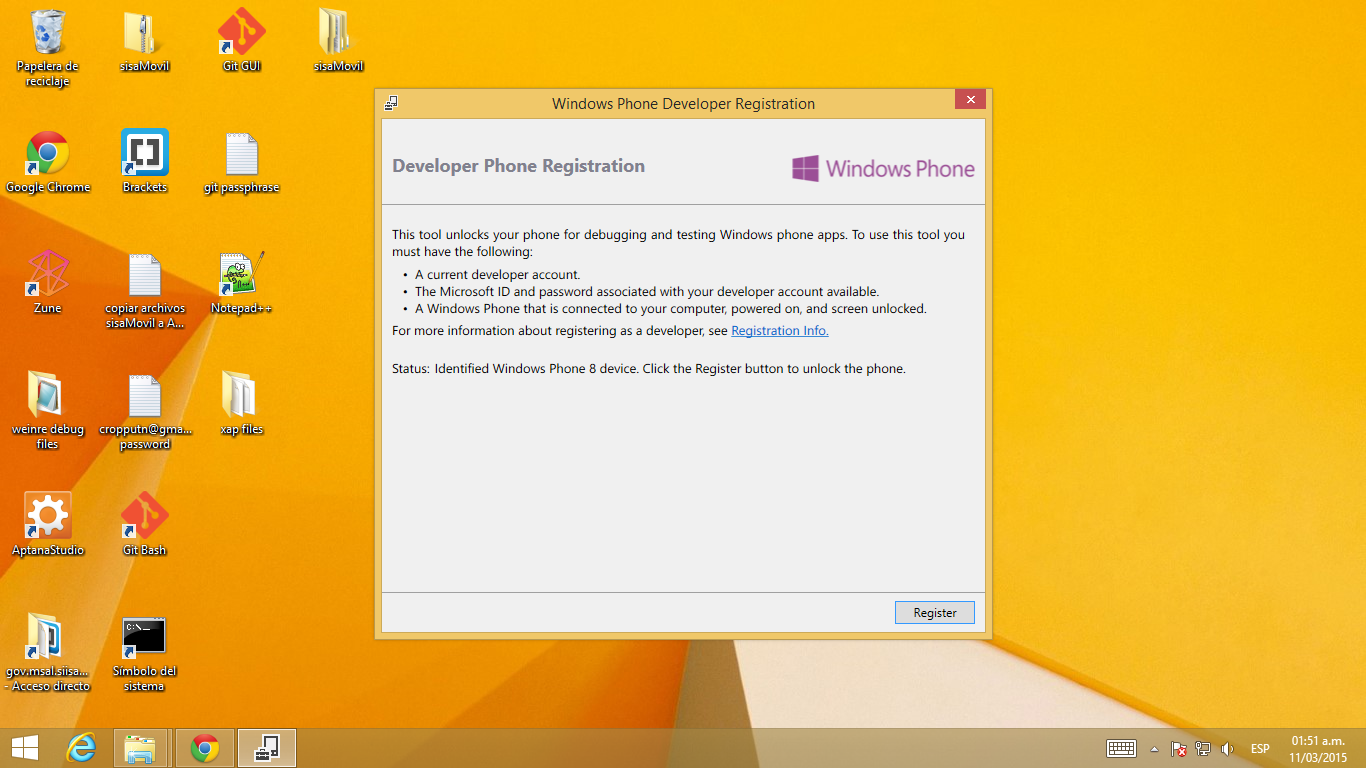
La forma de crear un proyecto de prueba es la misma que para Android, lo que difiere es la plataforma que se agrega, que corresponde a Windows Phone 8 (línea 4)

1. C:\>cd C:\PhoneGap\Projects\
2. C:\PhoneGap\Projects>cordova create hello com.example.hello HelloWorld
3. C:\PhoneGap\Projects>cd hello
4. C:\PhoneGap\Projects\hello>cordova platform add wp8
5. C:\PhoneGap\Projects\hello>cordova build wp8

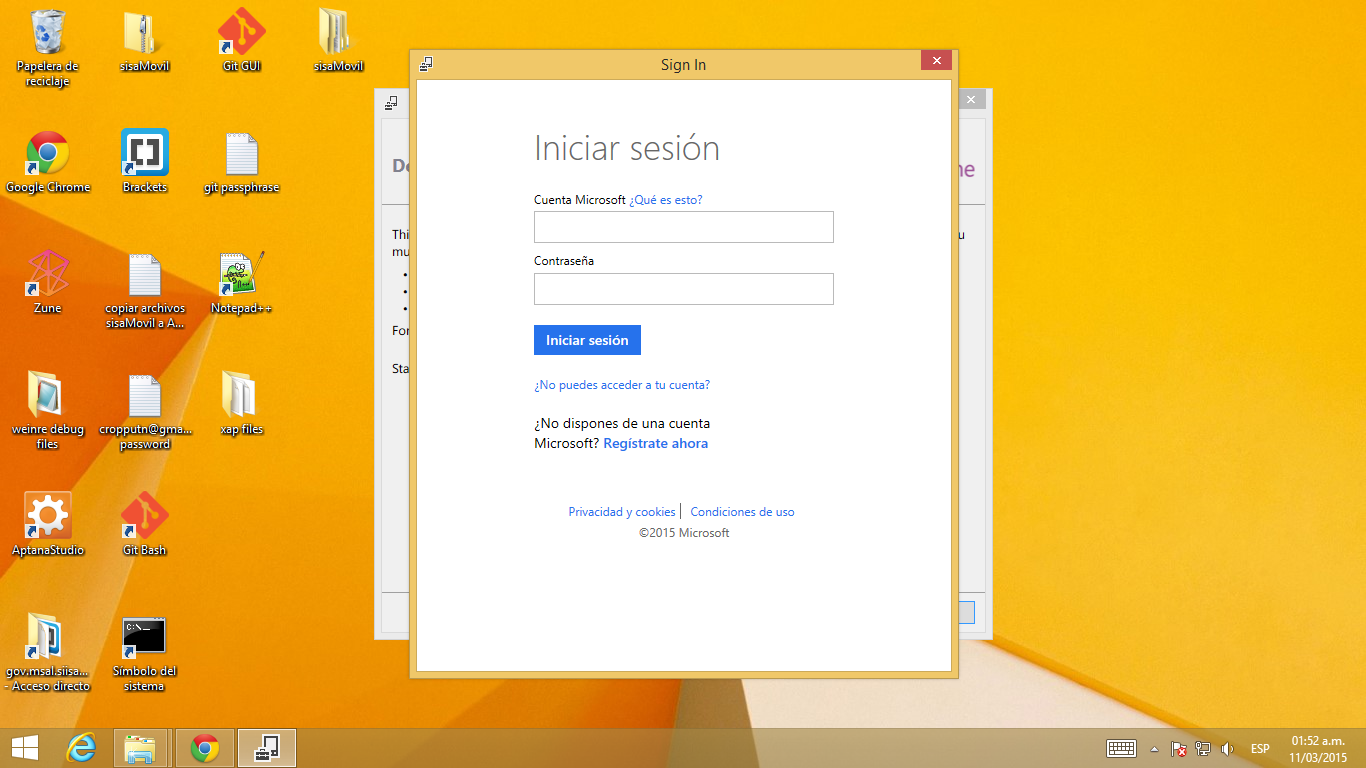
## Instalar .XAP

Al igual que para la plataforma Android, para Windows Phone 8 será necesario configurar el dispositivo para que puedan instalarse aplicaciones no firmadas con fines de desarrollo.

Previamente, deberemos tener conectado el dispositivo mediante USB. Después ingresaremos a la aplicación “Developer Phone Registration” y desde allí seleccionaremos la opción “Register” como se puede ver a continuación.



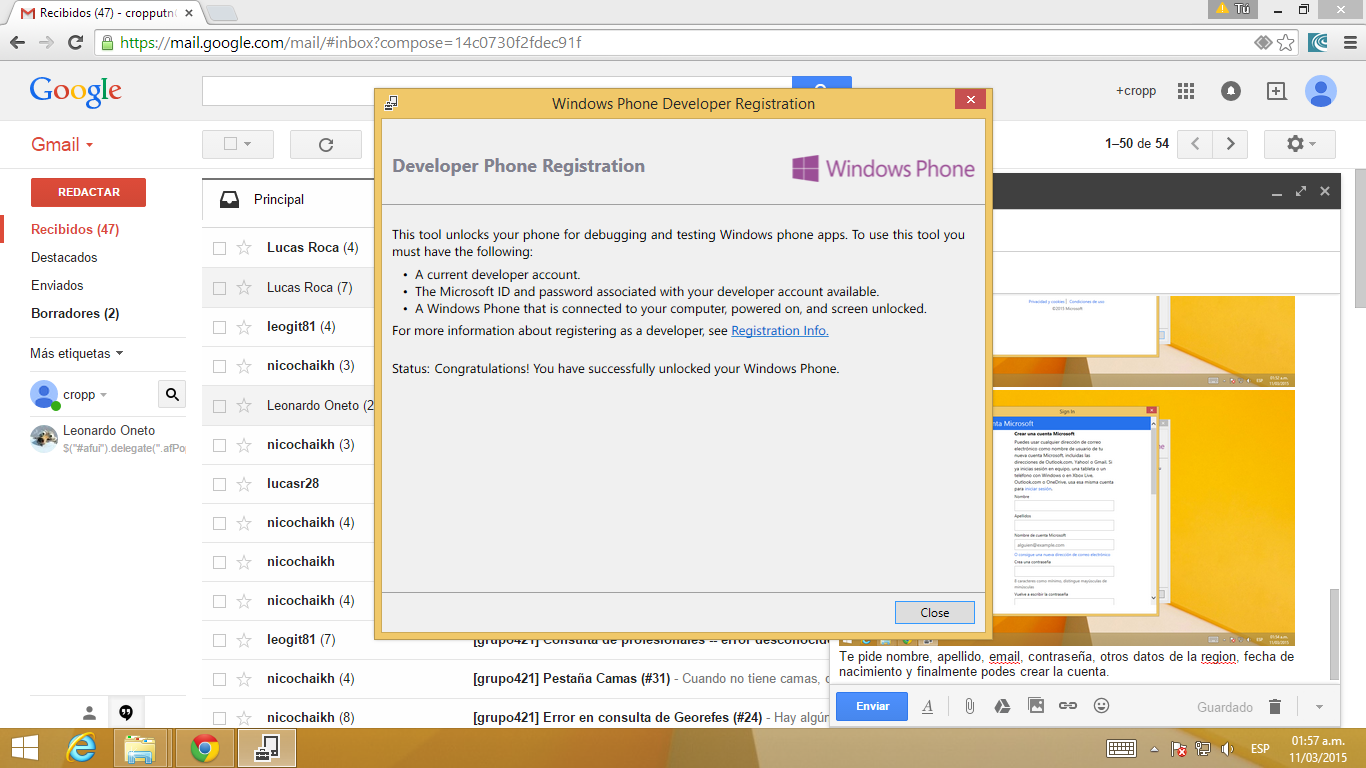
Nos pedirá que ingresemos una cuenta de Microsoft.



Si no contamos con una tenemos la posibilidad desde la misma aplicación acceder al formulario de registro, desde “Registrarse ahora”. Nos pedirá nombre, apellido, email, contraseña, otros datos de la región, fecha de nacimiento y finalmente podremos crear la cuenta.

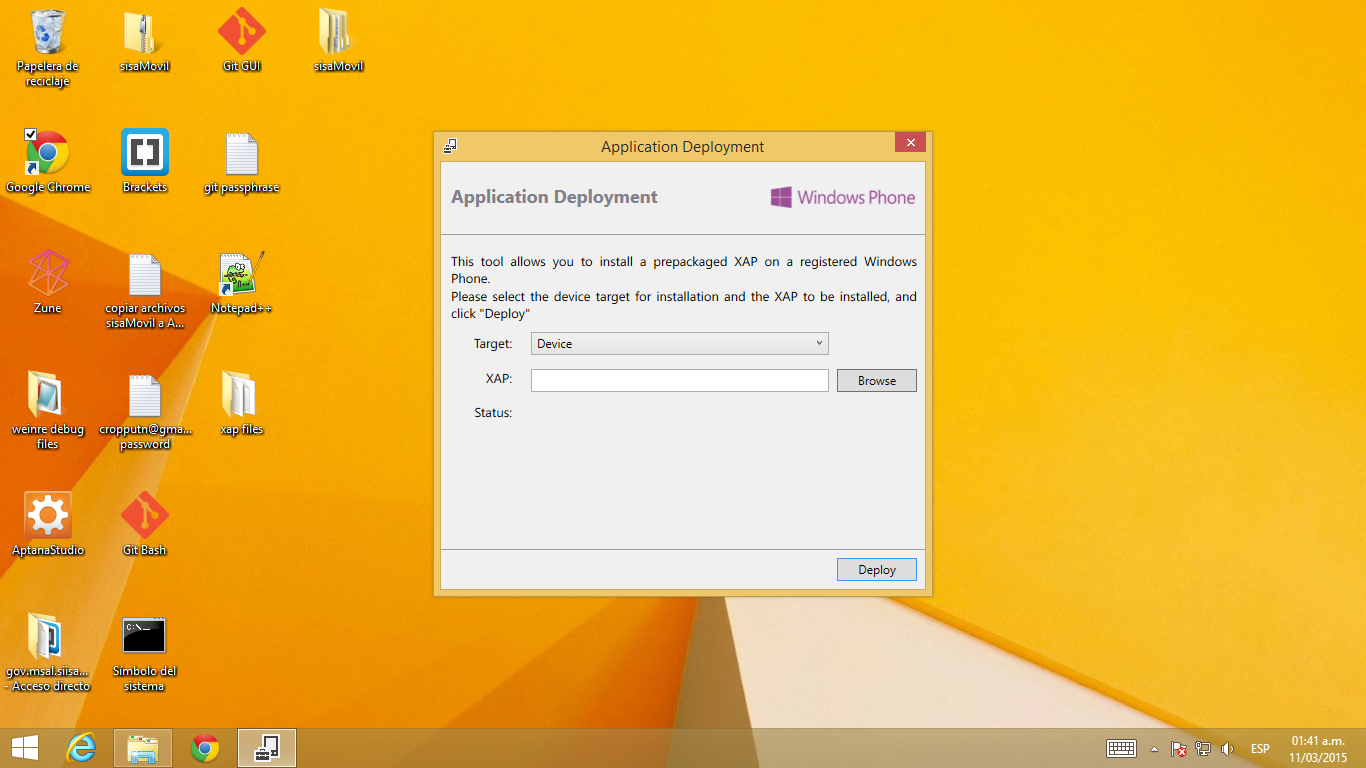


La misma aplicación nos indicará cuando el teléfono se encuentra habilitado para instalar aplicaciones con el siguiente mensaje.

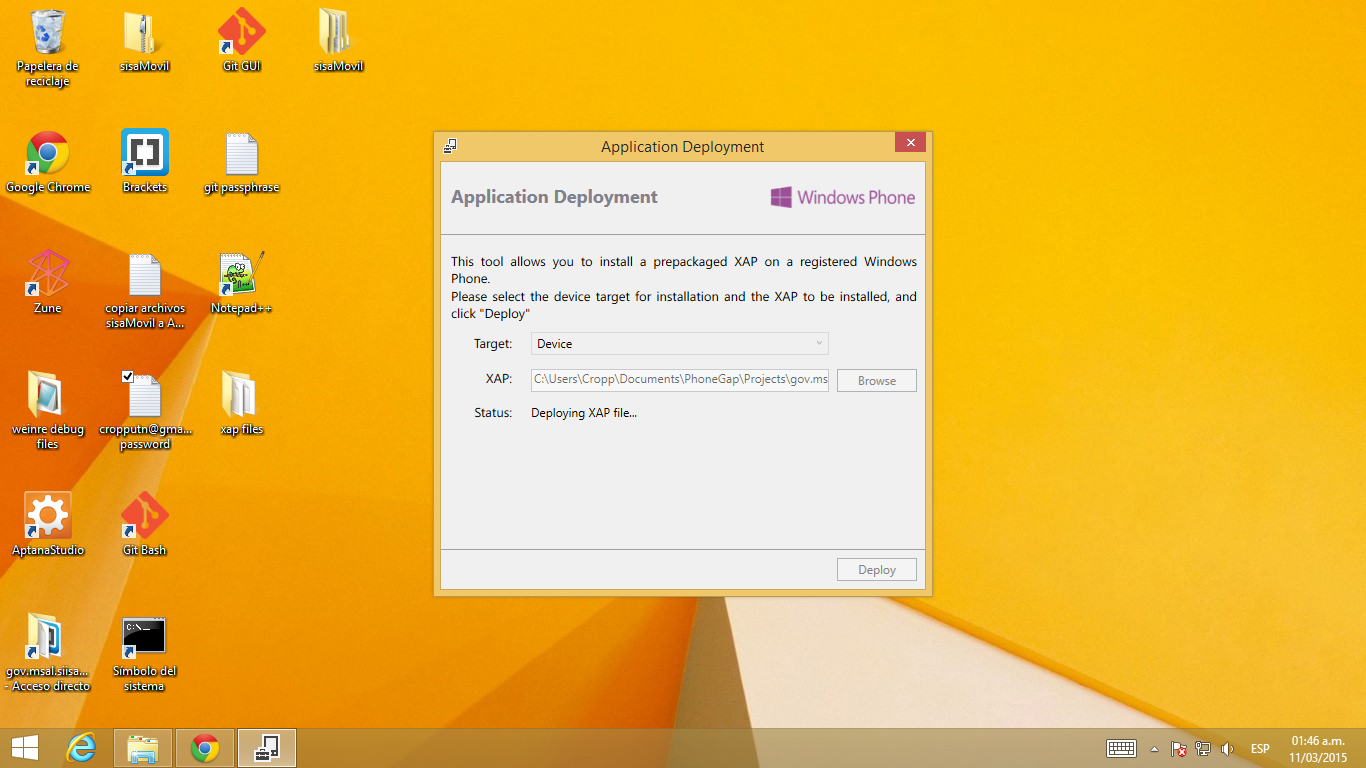


El archivo generado por la compilación se encuentra dentro de la carpeta donde tengamos el proyecto, en /platforms/wp8/Bin/Debug. Dentro de esta encontraremos un único archivo .xap que es el que utilizaremos.

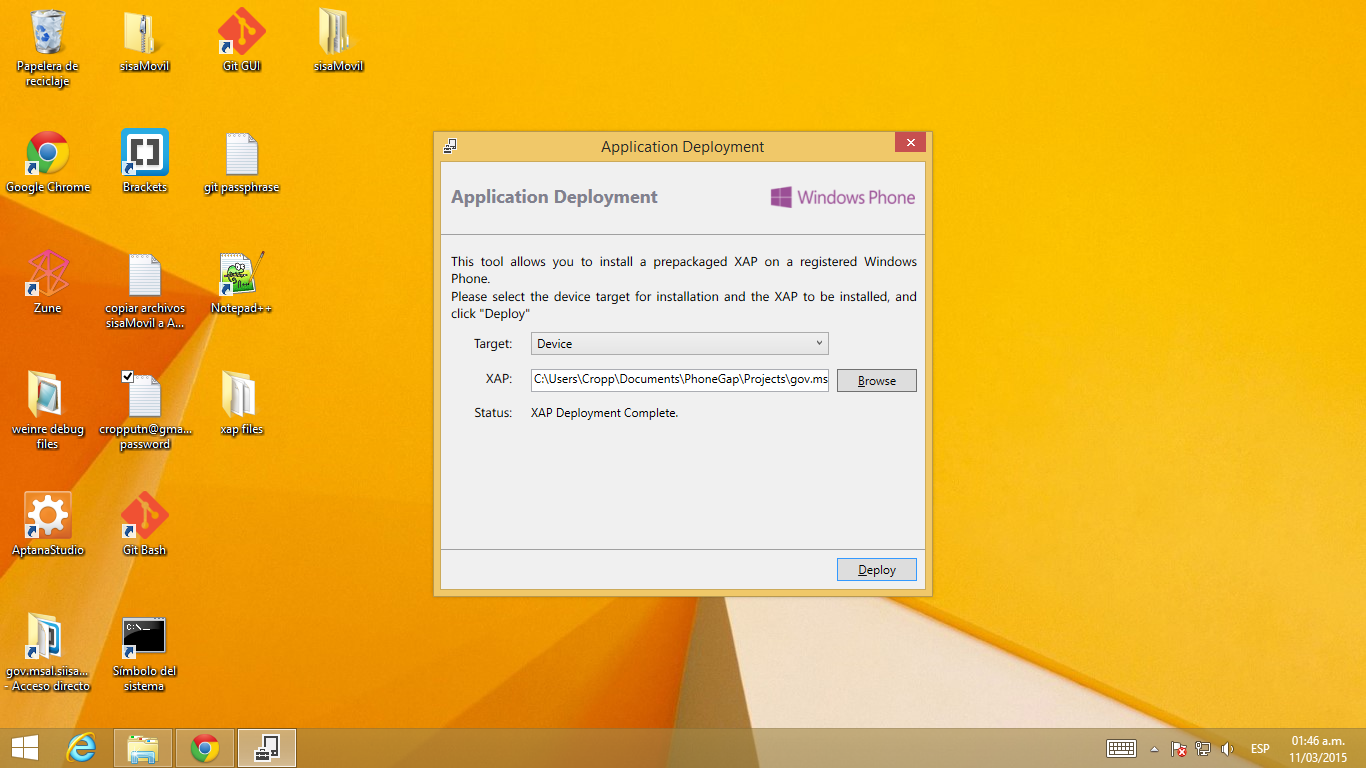
Con el instalador de aplicaciones (o Aplication deployment), deberemos seleccionar en la primera combo “Device”.



Y luego en la segunda, el archivo .xap generado en el paso anterior.



Primero desbloqueamos el dispositivo y luego hacemos clic en el botón “Deploy”. Una vez que la aplicación nos indique que el proceso ha terminado habrá finalizado la instalación.



Las dos aplicaciones antes mencionadas, **Application deployment** y **Developer Phone Registration** son instaladas cuando se instala la [Windows Phone SDK](#winPhoneSDK).

# Blackberry 10

Tendremos que instalar el entorno de desarrollo BlackBerry WebWorks SDK. Se puede descargar e instalar el mismo desde [developer.blackberry.com](https://developer.blackberry.com/html5/download/)

El instalador agregara las herramientas de líneas de comando al Path del sistema operativo.

## Crear un proyecto de prueba

Similar a las anteriores plataformas lo que difiere es que se agrega la plataforma para Blackberry 10 (línea 4)

1. C:\>cd C:\PhoneGap\Projects\
2. C:\PhoneGap\Projects>cordova create hello com.example.hello HelloWorld
3. C:\PhoneGap\Projects>cd hello
4. C:\PhoneGap\Projects\hello>cordova platform add blackberry10
5. C:\PhoneGap\Projects\hello>cordova build blackberry10

Al realizar ejecutar la línea 5, se creara dos archivos con extensión “.bar” en las siguientes rutas:

C:\PhoneGap\Projects \trunk\platforms\blackberry10\build\device

C:\PhoneGap\Projects \trunk\platforms\blackberry10\build\simulator

En la primera ruta, el archivo generado es para instalar en un dispositivo mientras que el segundo es utilizado para instalar en un simulador.

## Preparar el dispositivo para probar la aplicación

Para la instalación de la aplicación, es necesario poner al dispositivo en modo desarrollador. Para esto, en el dispositivo ingresar:

Configuración -> Seguridad y Privacidad -> Modo Desarrollador

Luego activar la opción de “Modo desarrollador”.

Una vez realizado esto, es necesario crear un “Debug Token”.

Para esto, es necesario ejecutar el ejecutable titulado “blackberry-debugtokenrequest.exe” instalado al realizar la instalación del WebWorks SDK. Para eso abrir una línea de comando, y localizarse en la carpeta contenedora.

Por defecto, la ruta es “C:\bbndk\host\_10\_3\_0\_2702\win32\x86\usr\bin”.

Luego ejecutar el siguiente comando:

blackberry-debugtokenrequest -storepass <password> -deviceid 0x999999 debugtoken.bar

Donde:

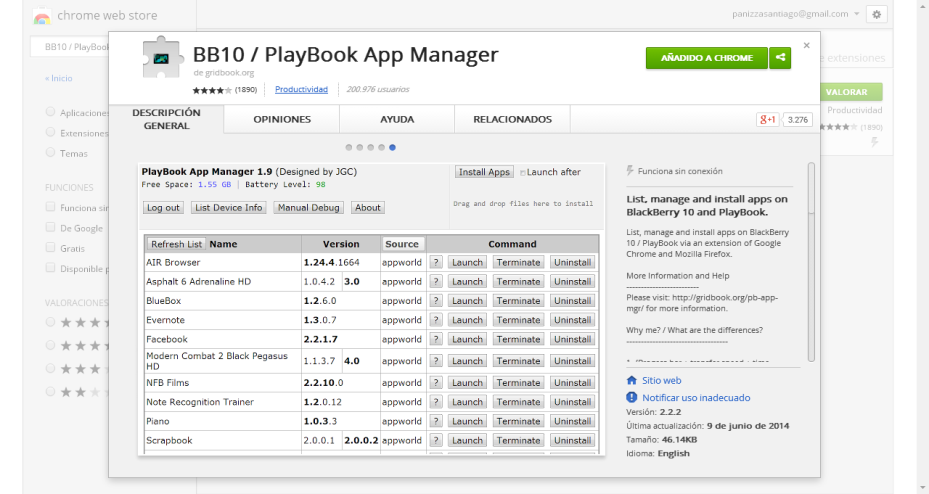
<password> Este parámetro es obligatorio aunque no tenga ningún uso en esta instancia. Se utilizara al momento subir la aplicación al BlackBerryWorld. Para esto primero es necesario crear un usuario en la siguiente página: http://developer.blackberry.com/

0x999999 Id del dispositivo donde se instalara el debugtoken. Para localizar el id del dispositivo ingresar en: Configuración –> Acerca de

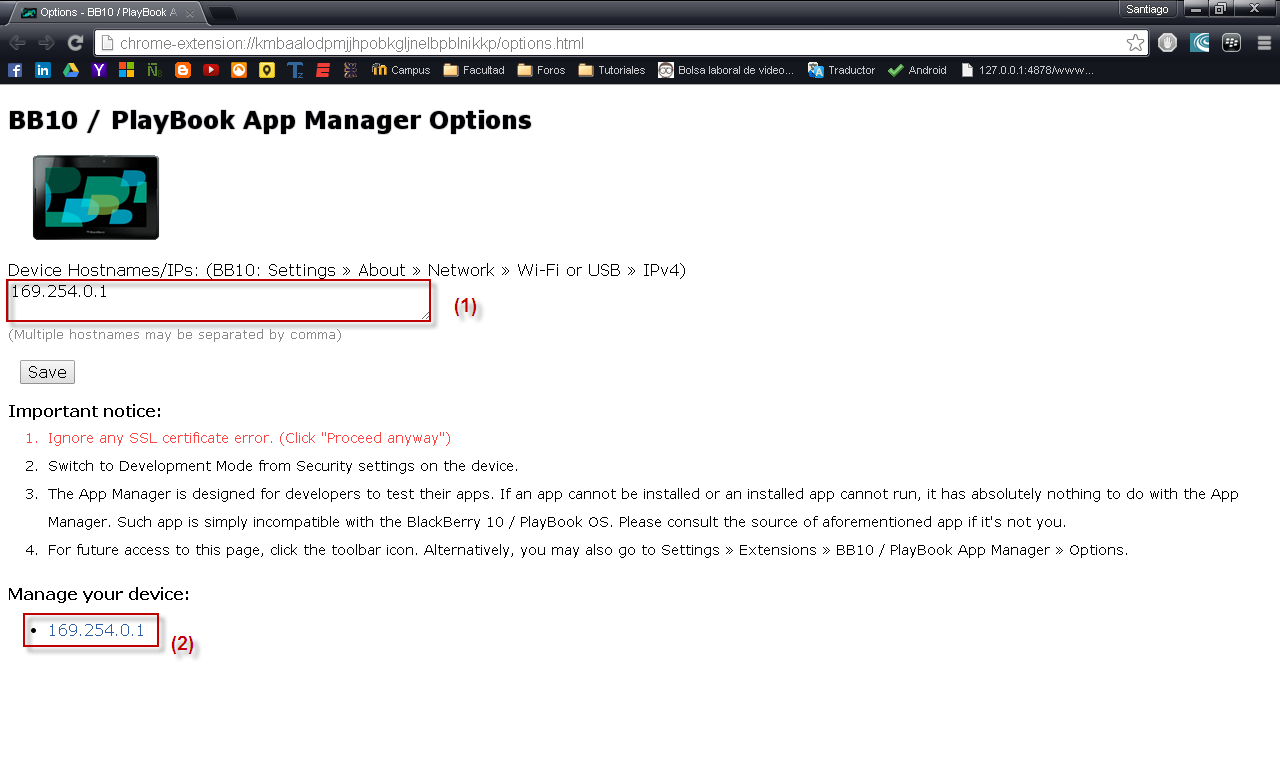
El token se creara en la misma ruta donde se encuentra el ejecutable.

## Instalar .BAR

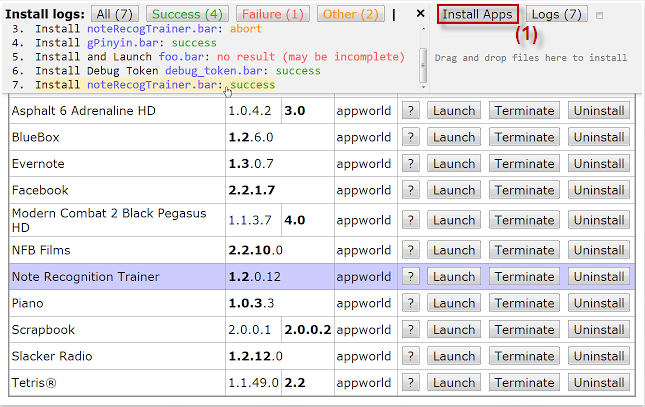
Para instalar los *“.bar”* en el dispositivo, la manera más simple de realizarlo es descargarse el complemento “*BB10 / PlayBook App Manager*” para el navegador Google Chrome. El mismo puede ubicarse Chrome Web Store de manera gratuita.



Una vez instalado el complemento, conectar el teléfono mediante USB a la computadora e ingresar al complemento.



En el cuadro de texto (1) ingresar la dirección de IP del dispositivo (por default: 169.254.0.1), presionar *“Save”* e ingresar presionando en la dirección IP (2). Se solicitara la contraseña del dispositivo.



Luego seleccionar el “*Install apps*”, seleccionar el *debugtoken* y aceptar. Luego, realizar el mismo procedimiento con el *.bar* generado al compilar la aplicación.

# Instalación JDK

* Ejecutar el instalador ([descarga JDK](http://www.oracle.com/technetwork/java/javase/downloads/index.html))
* Configurar variable de entorno PATH y JAVA\_HOME, apuntando a la carpeta del sistema donde se hayan instalado los archivos en el paso anterior. Consultar el siguiente link para [más información](http://davidg-tech.blogspot.com.ar/2012/12/java-configurar-variables-de-entorno-javahome-y-path.html).

# Herramientas complementarias utilizadas en el desarrollo

## Backbone.js

Permite definir vistas, modelos y colecciones (tomando la idea del patrón de arquitectura MVC, aunque no en su totalidad), y establecer bindeos entre las vistas y los modelos mediante eventos.

<http://backbonejs.org/>

Backbone tiene como dependencias jQuery y underscore.js Por este motivo, también se pueden encontrar a estas en el proyecto. jQuery es ampliamente conocida, y underscore es una biblioteca que agrupa un conjunto de funciones muy útiles para tareas comunes, como el manejo de arrays y strings, y también la manipulación de objetos en JavaScript.

## Intel App Framework

Esta biblioteca nos provee controles de la Interfaz Gráfica de usuario, animaciones (para las transiciones entre los paneles) y estilos que se adaptan a la plataforma en la cual se ejecuta la aplicación.

<http://app-framework-software.intel.com/>

Esta biblioteca se divide en 2 parte:

* Appframework.ui.js
* Appframework.js

La primera de ellas como dijimos antes, la utilizamos ampliamente para el diseño de la interfaz gráfica de la aplicación. La segunda cumple funciones similares a jQuery porque permite obtener elementos del DOM (Document Object Model) mediante la especificación de un selector de CSS.

## Ripple emulator

Esta herramienta es un plugin de Chrome que nos permite ejecutar aplicaciones desarrolladas con PhoneGap emulando los eventos que se dispararían desde un dispositivo. La usamos para poder hacer el debug de la aplicación sin tener que hacer el deploy a un celular.

<https://chrome.google.com/webstore/detail/ripple-emulator-beta/geelfhphabnejjhdalkjhgipohgpdnoc>

## Weinre

Esta herramienta útil para realizar debug remoto de la aplicación una vez que se encuentra instalada en el celular. Es utilizada para realizar el debug en teléfonos android.

[http://people.apache.org/~pmuellr/weinre-docs/latest/](http://people.apache.org/%7Epmuellr/weinre-docs/latest/)

## BlackBerry Web Inspector

Esta herramienta es similar a la anterior, pero se utiliza para realizar el debug en dispositivos BlackBerry. No es necesario nada, ya que viene incorporada en los celulares que posean BlackBerry10. Únicamente se debe ejecutar la aplicación conectando previamente el celular a la maquina. Al comenzar a ejecutar la aplicación, nos mostrara una IP a la cual podemos acceder desde cualquier navegador, donde nos proporcionara una plataforma para realizar el debug de la aplicación. Este herramienta solo se deshabilita cuando subimos la aplicación al BlackBerry World.