



# **UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**MANUAL TÉCNICO**

**TEMA:**

**“ESTUDIO DE LOS PRINCIPALES MODELOS DE BANCA MÓVIL  
Y DESARROLLO DE UN APLICATIVO”**

**AUTOR: CRISTIAN GIOVANNY NARVÁEZ CÓRDOVA**

**DIRECTOR: MSC. DIEGO TREJO ESPAÑA**

**IBARRA – ECUADOR**

**2016**

## 1 PROPÓSITO

Este documento pretende brindar información sobre la estructura del simulador de banca y las partes que la conforman, con el fin que se dé a entender cómo fue construido.

## 2 OPENSIFT

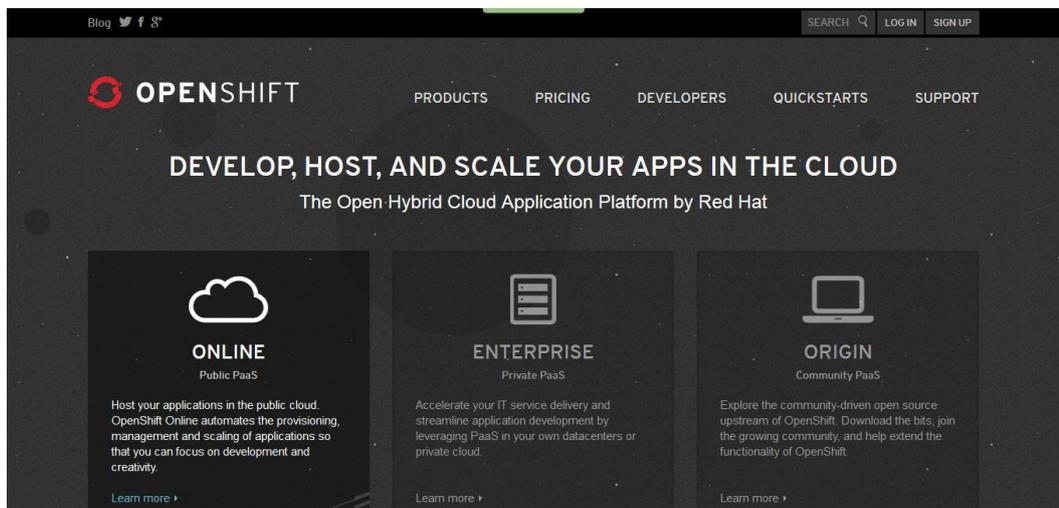
### 2.1 Introducción

OpenShift es la plataforma como servicio (PaaS) de Red Hat, que permite a los desarrolladores crear, gestionar y desplegar rápidamente aplicaciones escalables en un entorno de nube.

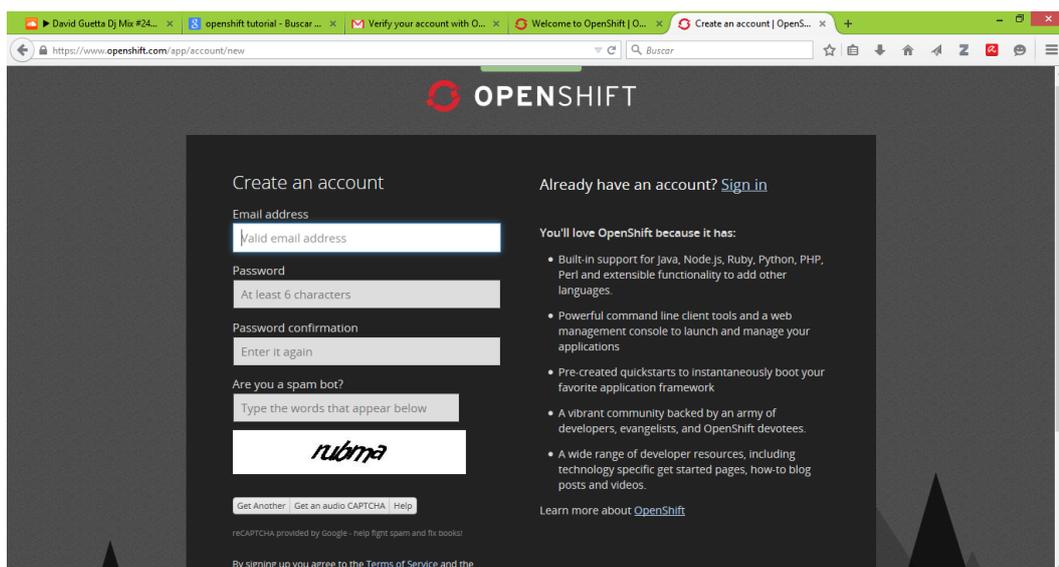
Aquí es donde se ha puesto en marcha el sistema administrador del aplicativo banca móvil, es por esta razón la vital importancia de su conocimiento.

### 2.2 Creación de cuenta

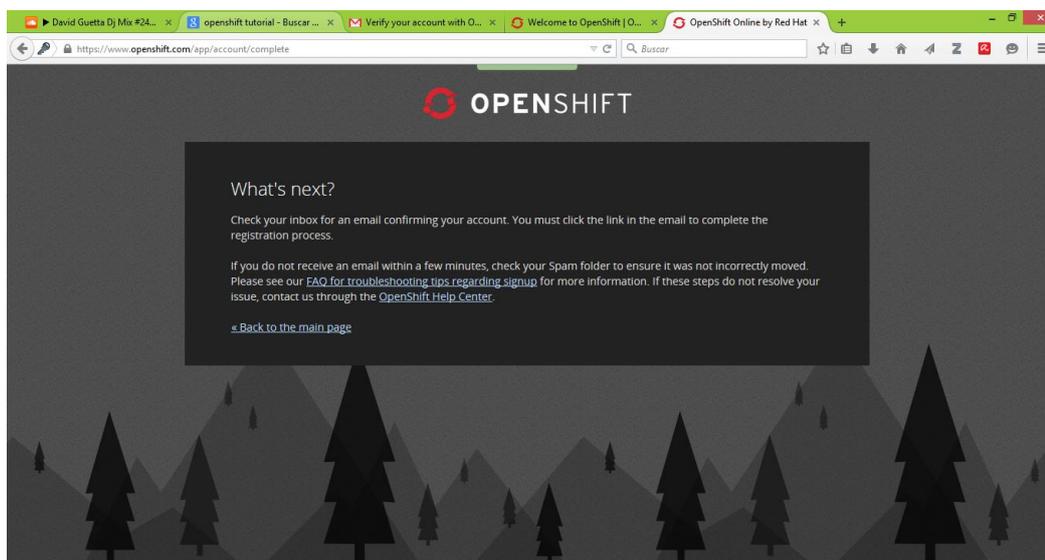
Para crear una cuenta nos dirigimos a <https://www.openshift.com/>, y pulsamos el botón **Sign Up**, que se encuentra en la parte superior derecha.



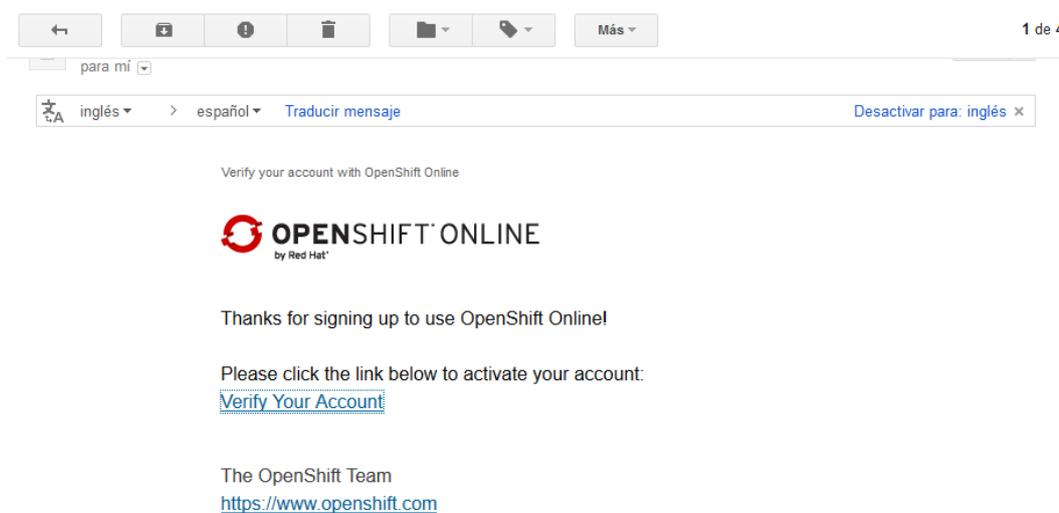
Luego se mostrara un formulario de registro donde ingresaremos los datos de nuestra cuenta, y pulsaremos el botón **Sign Up**.



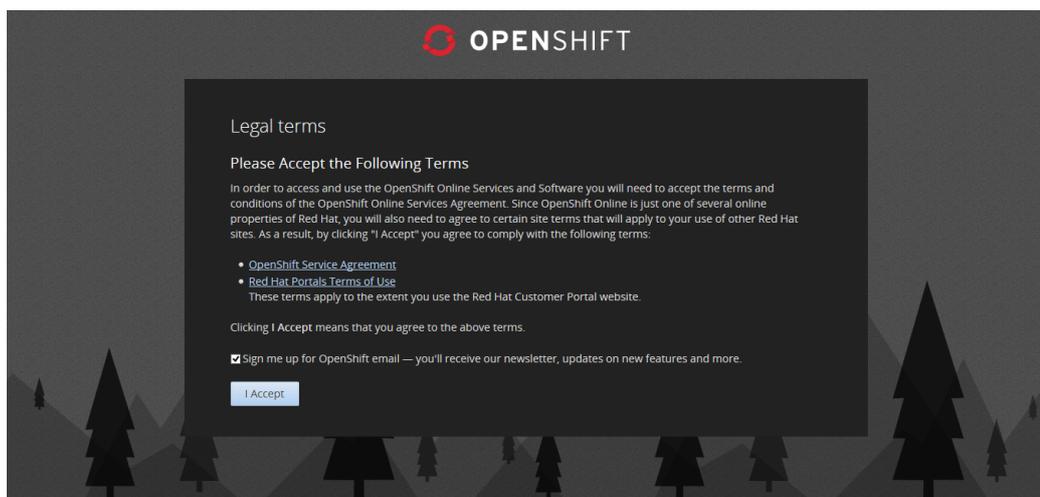
A continuación nos aparecerán las indicaciones para activar nuestra cuenta ya creada, en resumen no indican que debemos realizar este proceso mediante nuestro correo electrónico.



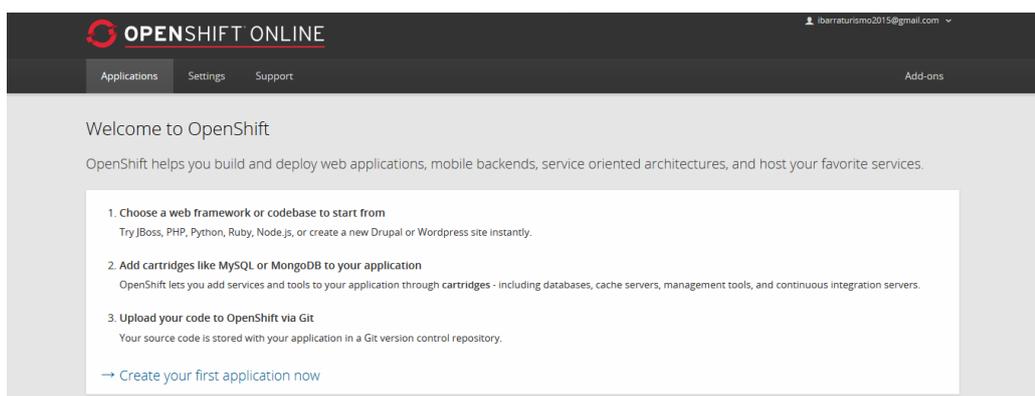
Al abrir nuestro mensaje dentro del correo electrónico, nos aparece un enlace de activación de cuenta.



Al dar clic en este nos aparece la aceptación de términos legales de uso de OpenShift.



Finalmente se muestra nuestro escritorio de trabajo virtual de OpenShift.



## 2.3 Registro de dominio

El primer paso dentro de nuestra cuenta es la creación de un dominio, para esto nos dirigimos a la pestaña de **Settings** y en la sección **Namespace** nos aparece el cuadro de texto donde escribiremos el nombre del dominio, el cual será verificado para su disponibilidad.

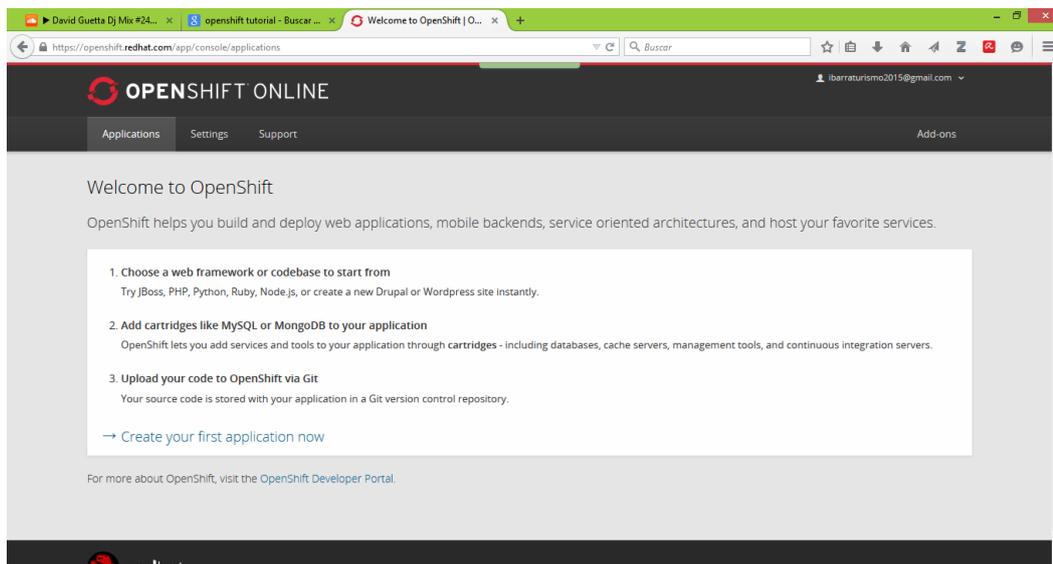
The screenshot shows the 'Settings' page in OpenShift Online. At the top, there is a navigation bar with 'Applications', 'Settings', and 'Support' tabs. A user profile 'ibarraturismo2015@gmail.com' is visible in the top right. Below the navigation bar, a blue message bar states: 'You need to set a namespace before you can create applications'. The 'Namespace' section explains that the namespace is unique to the account and is the suffix of the public URLs. A text input field contains 'http://applicationname- touribarra| .rhcloud.com'. Below the input field, a note states: 'Your domain name must be letters or numbers with no spaces or symbols.' A 'Save' button is located below the input field. The 'Public Keys' section explains that OpenShift uses a public key to securely encrypt the connection between the local machine and the application. It includes a text area for pasting the contents of a public key file (.pub) and a 'Save' button. A link 'Learn more about SSH keys' is also present.

Al presionar el botón **Save** finalizaremos el proceso de creación de nuestro dominio y nos aparecerá un mensaje de éxito.

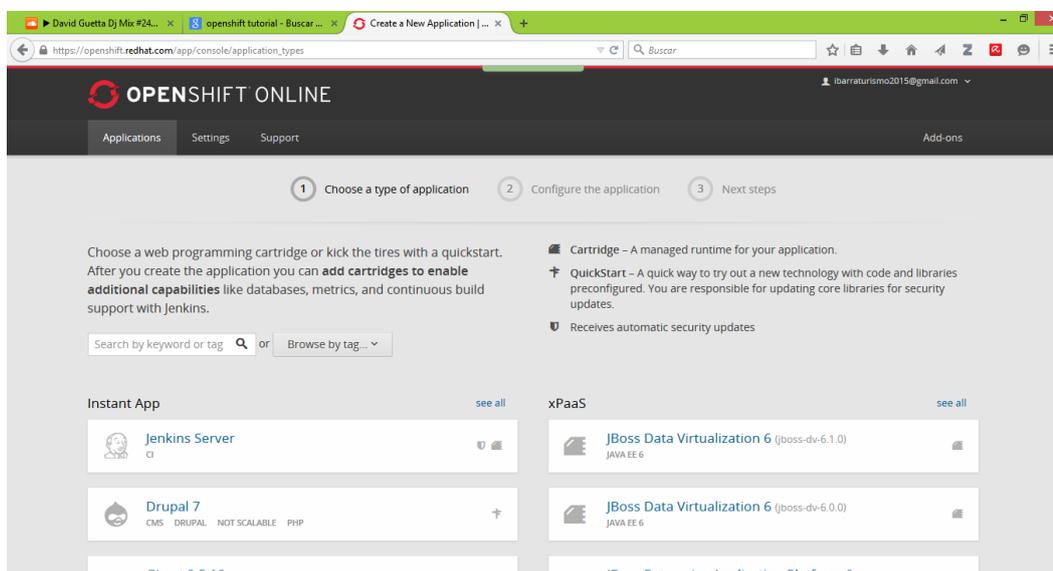
The screenshot shows the 'Settings' page in OpenShift Online after the domain creation. A green message bar at the top states: 'The domain 'touribarra' has been created'. Below this, a blue message bar states: 'You need to set a public key before you can work with application code'. The 'Public Keys' section is now the primary focus, with the same text area for pasting the contents of a public key file (.pub) and a 'Save' button. A link 'Learn more about SSH keys' is also present. The 'Domains' section is now visible at the bottom, with a note: 'See the User Guide for information about adding your own domain names to an application.'

## 2.4 Creación de una aplicación

Dentro de nuestra cuenta tenemos la posibilidad de crear nuestras aplicaciones para modificarlas posteriormente. Para crear una aplicación nos dirigimos a la pestaña **Applications** y pulsamos **Create application**.



OpenShift ofrece varias posibilidades a los desarrolladores de aplicaciones como son JAVA, PHP, PHYTON, RUBY, entre otras. Basta con seleccionar la que necesitamos y seguir los pasos de forma intuitiva.



En este caso navegamos por las opciones y seleccionamos Node.js.

Ahora configuraremos nuestra aplicación y procederemos a crearla.

1 Choose a type of application    2 **Configure the application**    3 Next steps

**Based On**    **Node.js 0.10 Cartridge** 

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js is perfect for data-intensive real-time applications that run across distributed devices.

<http://www.nodejs.org/>

- ☆ OpenShift maintained
- 🔒 Receives automatic security updates

**Public URL**   

OpenShift will automatically register this domain name for your application. You can add your own domain name later.

**Source Code**       

We'll create a Git code repository in the cloud, and populate it with a set of reasonable defaults. If you provide a Git URL, your application will start with an exact copy of the code and configuration provided in this Git repository.

**Gears**    **small**

Gears are the application containers running your code. For most applications, the small gear size provides plenty of resources. You can also [upgrade your plan](#) to get access to more gear sizes.

Una vez creada nos aparecerán opciones de ayuda y manejo de la misma.

1 Choose a type of application    2 Configure the application    3 **Next steps**

Your application has been created. [Continue to the application overview page.](#)

**Making code changes**

Install the Git client for your operating system, and from your command line run

```
git clone ssh://550f79634382ec4fab00041@imgserv-touribarra.rhcloud.com/~/.git
imgserv.git/
cd imgserv/
```

This will create a folder with the source code of your application. After making a change, **add**, **commit**, and **push** your changes.

```
git add .
git commit -m 'My changes'
git push
```

When you push changes the OpenShift server will report back its status on deploying your code. The server will run any of your configured [deployment hooks](#) and then restart the application.

**Manage your app**

The console is convenient, but if you need deeper control try our other client tools

**Command-Line**

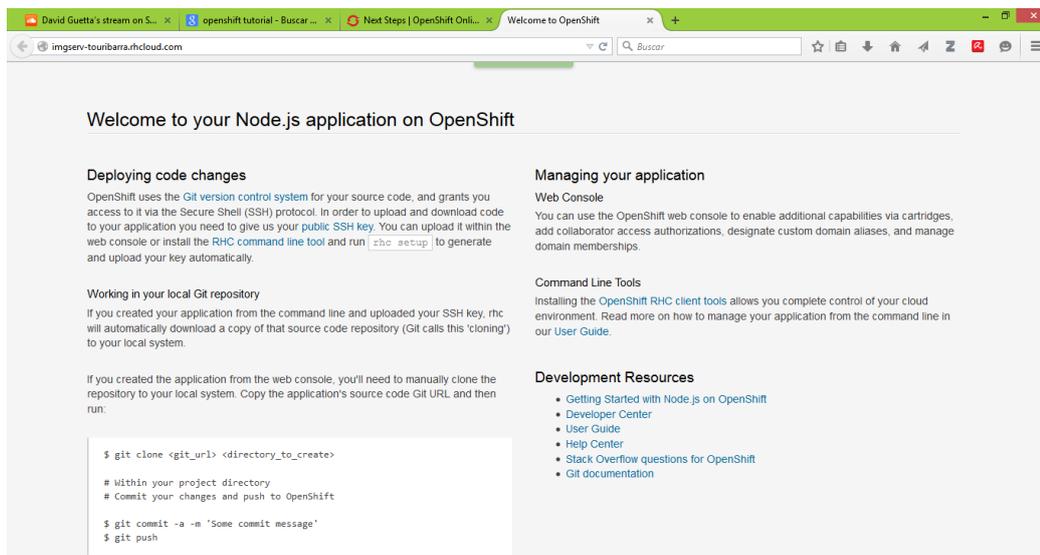
All of the capabilities of OpenShift are exposed through our command line tool, *rhc*. [Follow these steps to install the client](#) on Linux, Mac OS X, or Windows.

After installing the RHC read more on [how to manage your application from the command line](#) in our User Guide.

**JBoss Developer Studio**

The JBoss Developer Studio is a full featured IDE with OpenShift integration built in. It gives you the ability to create, edit and deploy applications without having to leave the IDE. Links to download, install and use the JBoss Developer Studio for Linux, Mac OS X, or Windows can be found on the [JBoss Developer Studio tools page](#).

Podemos verificar la creación de la aplicación escribiendo la dirección en nuestro navegador.

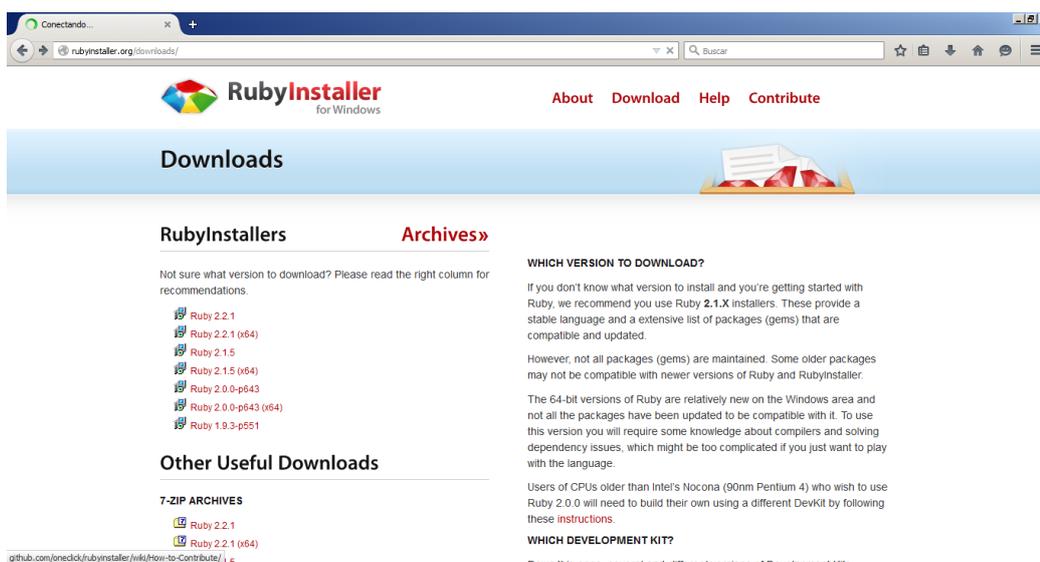


## 2.5 Instalación del cliente de OpenShift

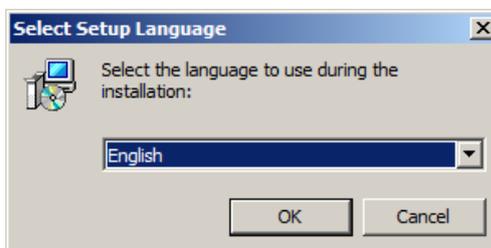
Para poder manejar OpenShift debemos instalar dependencias importantes, como son: Ruby, Git y RHC Gem.

### 2.5.1 Ruby

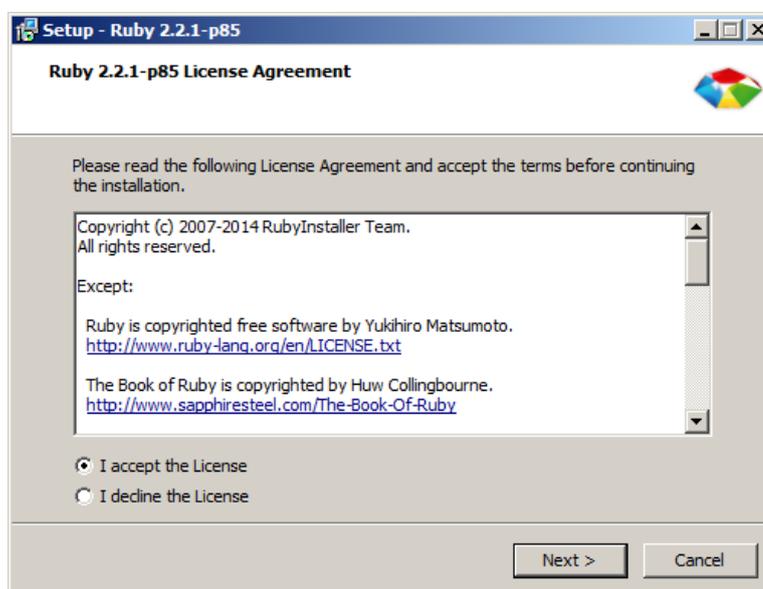
Para instalar Ruby en el sistema operativo Windows vamos a el sitio <http://rubyinstaller.org/downloads>, y elegimos la versión del instalador.



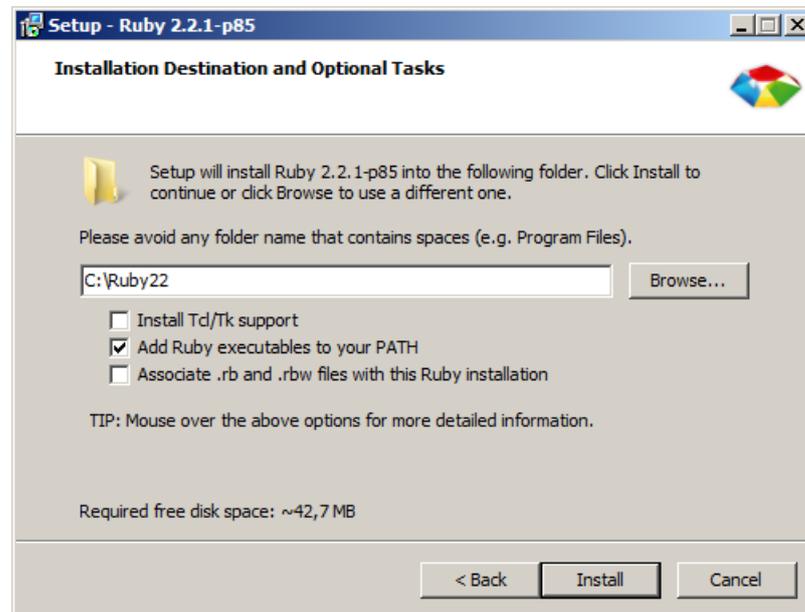
Al ejecutar el instalador nos mostrará un mensaje de selección de idioma para la instalación, lo dejaremos en English por defecto.



Luego aceptaremos las condiciones de la licencia del lenguaje de programación.



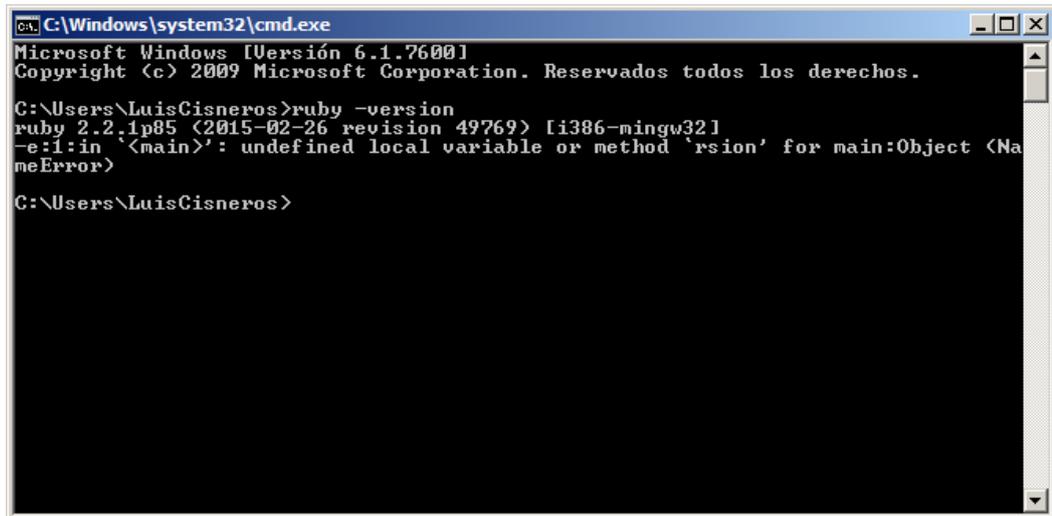
Escogeremos la carpeta de instalación y seleccionaremos la segunda casilla, permitiendo añadir las variables al PATH.



Finalmente al pulsar el botón Install se inicia el proceso de instalación y cuando este termine, se mostrará la ventana de finalización exitosa.



Si deseamos comprobar la instalación, nos dirigimos al terminal de Windows y ejecutamos el comando **ruby -version**.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

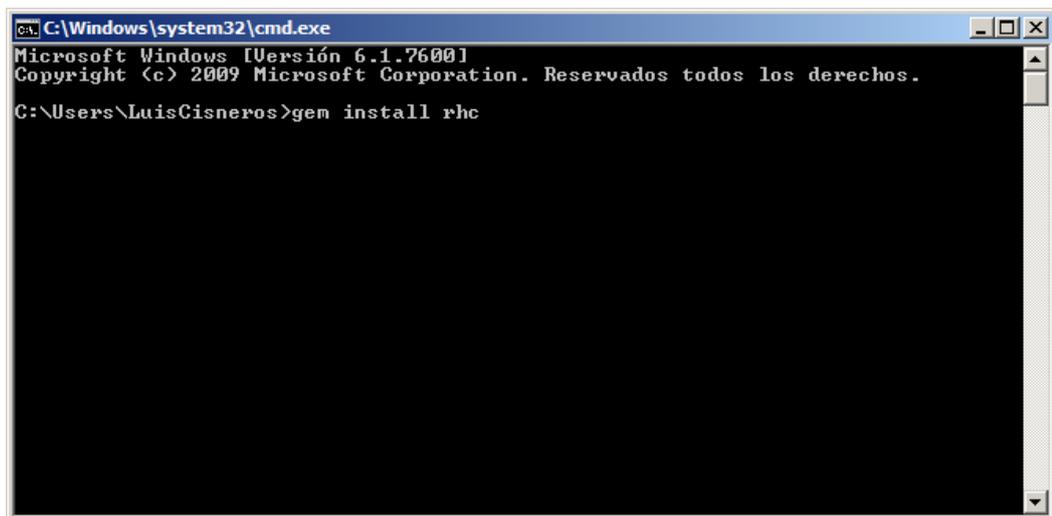
C:\Users\LuisCisneros>ruby -version
ruby 2.2.1p85 (2015-02-26 revision 49769) [i386-mingw32]
-e:1:in <main>: undefined local variable or method `rsion' for main:Object (NameError)

C:\Users\LuisCisneros>
```

### 2.5.2 Gema RHC

La gema RHC, es la herramienta por así decirlo para la conexión hacia los servicios de OpenShift y el manejo de cada uno de nuestras aplicaciones mediante SSH.

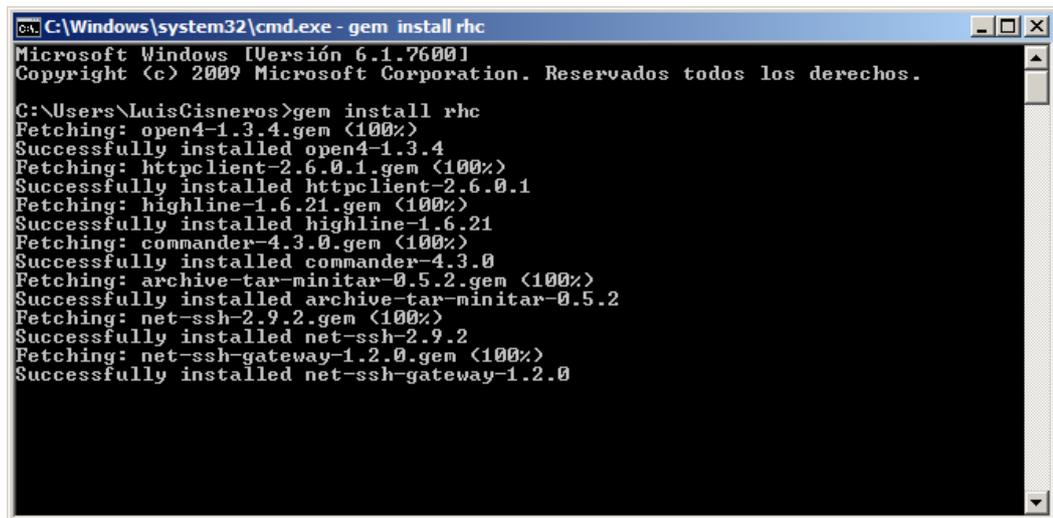
Para instalar la gema, ingresamos al terminal y ejecutamos el comando **gem install rhc**.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\LuisCisneros>gem install rhc
```

Rápidamente el proceso de instalación inicia y se instalan las otras gemas dependientes.



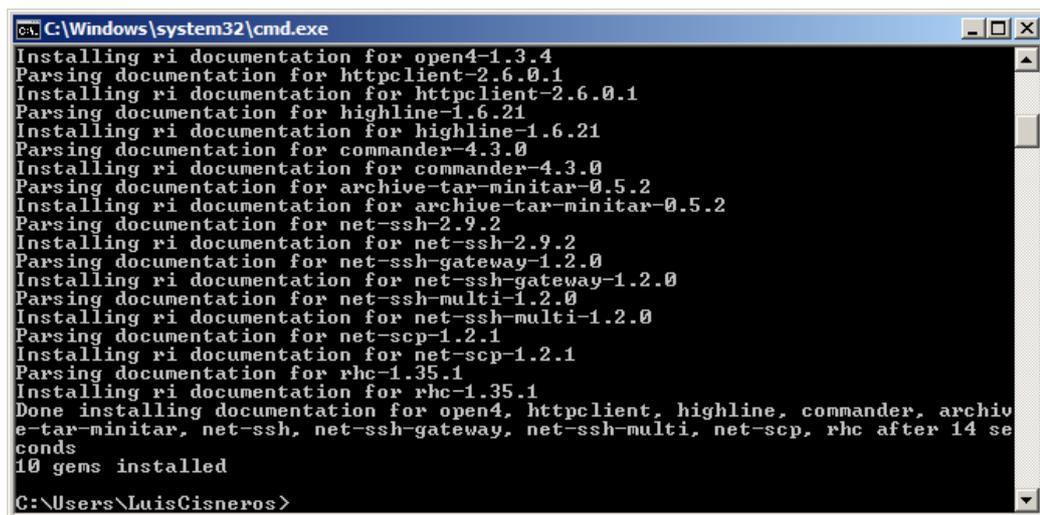
```

C:\Windows\system32\cmd.exe - gem install rhc
Microsoft Windows [Versión 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\LuisCisneros>gem install rhc
Fetching: open4-1.3.4.gem (100%)
Successfully installed open4-1.3.4
Fetching: httpclient-2.6.0.1.gem (100%)
Successfully installed httpclient-2.6.0.1
Fetching: highline-1.6.21.gem (100%)
Successfully installed highline-1.6.21
Fetching: commander-4.3.0.gem (100%)
Successfully installed commander-4.3.0
Fetching: archive-tar-minitar-0.5.2.gem (100%)
Successfully installed archive-tar-minitar-0.5.2
Fetching: net-ssh-2.9.2.gem (100%)
Successfully installed net-ssh-2.9.2
Fetching: net-ssh-gateway-1.2.0.gem (100%)
Successfully installed net-ssh-gateway-1.2.0

```

Una vez finalizado el proceso nos aparecerá un mensaje de éxito.



```

C:\Windows\system32\cmd.exe
Installing ri documentation for open4-1.3.4
Parsing documentation for httpclient-2.6.0.1
Installing ri documentation for httpclient-2.6.0.1
Parsing documentation for highline-1.6.21
Installing ri documentation for highline-1.6.21
Parsing documentation for commander-4.3.0
Installing ri documentation for commander-4.3.0
Parsing documentation for archive-tar-minitar-0.5.2
Installing ri documentation for archive-tar-minitar-0.5.2
Parsing documentation for net-ssh-2.9.2
Installing ri documentation for net-ssh-2.9.2
Parsing documentation for net-ssh-gateway-1.2.0
Installing ri documentation for net-ssh-gateway-1.2.0
Parsing documentation for net-ssh-multi-1.2.0
Installing ri documentation for net-ssh-multi-1.2.0
Parsing documentation for net-scp-1.2.1
Installing ri documentation for net-scp-1.2.1
Parsing documentation for rhc-1.35.1
Installing ri documentation for rhc-1.35.1
Done installing documentation for open4, httpclient, highline, commander, archiv
e-tar-minitar, net-ssh, net-ssh-gateway, net-ssh-multi, net-scp, rhc after 14 se
conds
10 gems installed

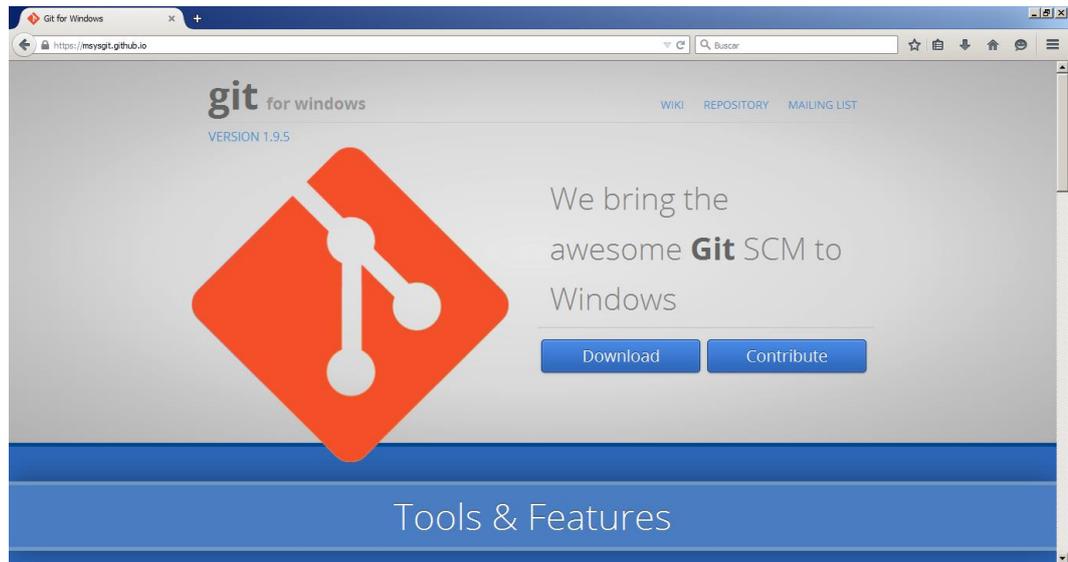
C:\Users\LuisCisneros>

```

### 2.5.3 Git

Git es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Además nos servirá para clonar nuestros archivos de las aplicaciones de la plataforma OpenShift.

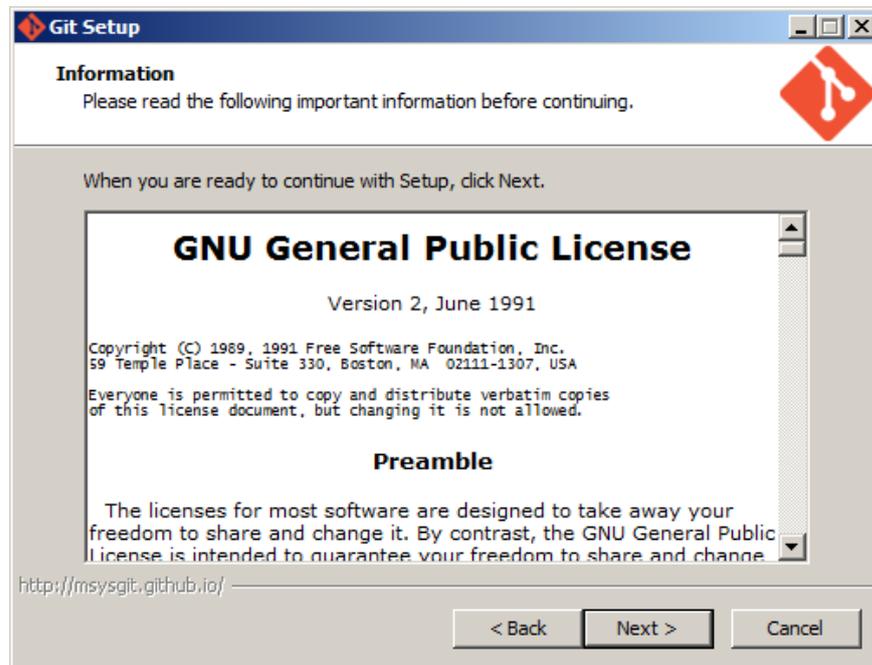
La instalación del software es muy sencilla, primero nos dirigimos a la siguiente dirección web <https://msysgit.github.io/> y pulsaremos el botón descargar.



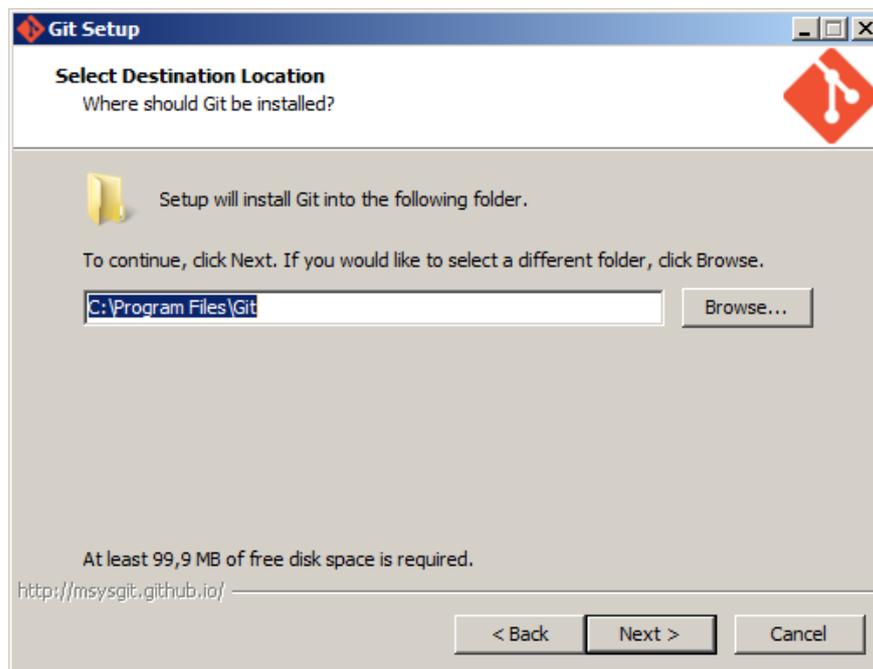
Una vez descargado el instalador, lo ejecutamos iniciando así el proceso de instalación dentro del computador.



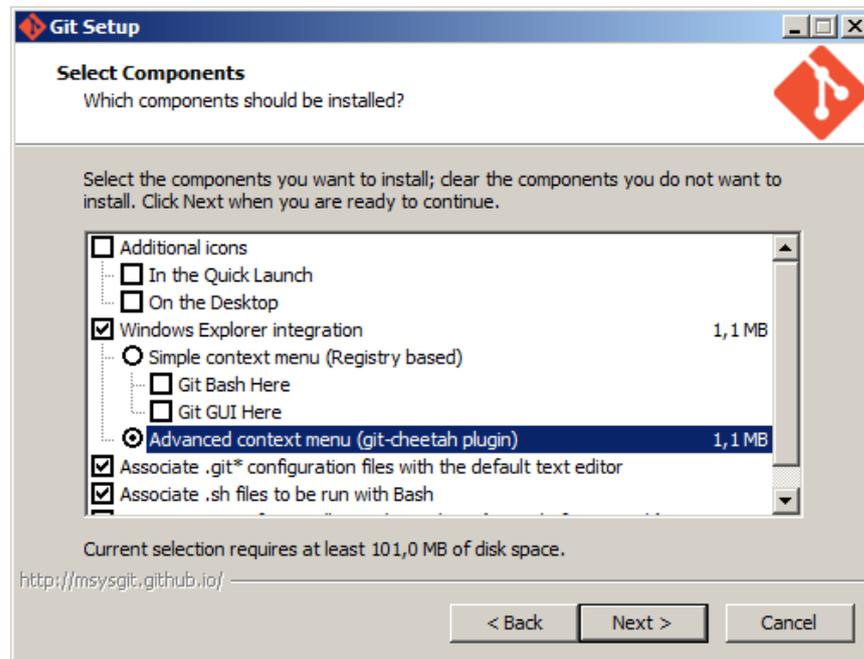
Aceptaremos los términos y condiciones de la licencia.



Elegimos el directorio de instalación.



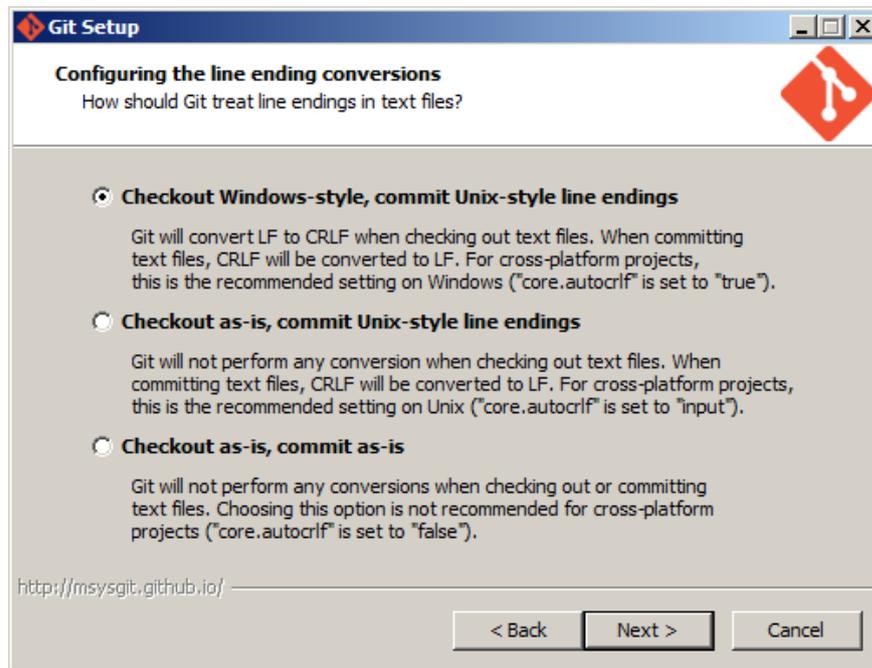
Seleccionamos los componentes a instalar.



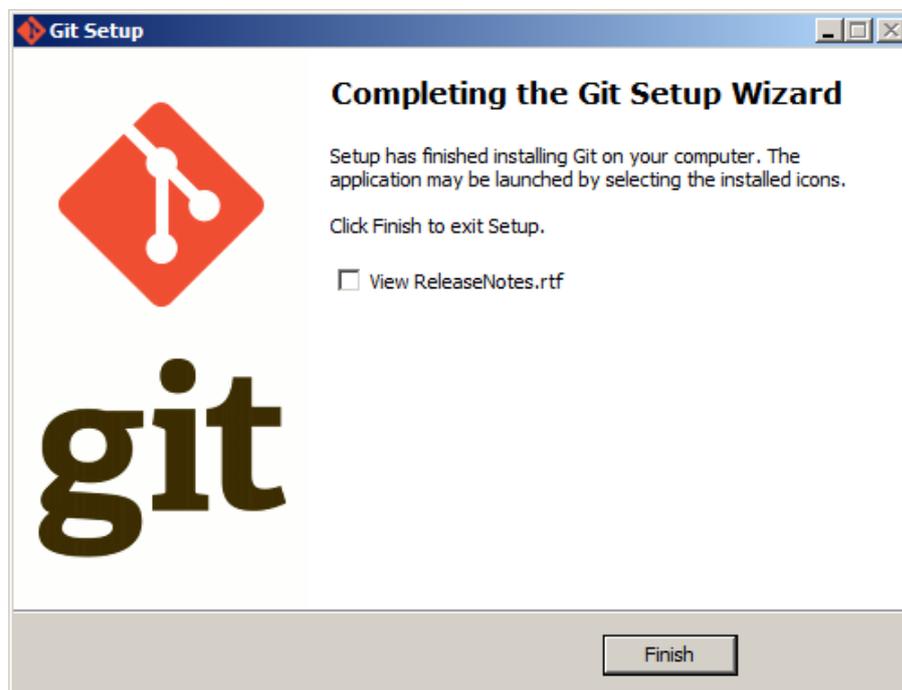
Asignaremos las variables a nuestro PATH.



Dejaremos las últimas configuraciones por defecto e inicia el proceso de instalación.



Una vez terminado saldrá una ventana con un mensaje de éxito.



#### 2.5.4 Conexión hacia OpenShift

Para realizar nuestra primera conexión hacia la plataforma y la configuración de nuestra cuenta, nos dirigimos al terminal y ejecutamos el comando **rhc setup**. En

seguida se muestra un mensaje del cliente de OpenShift y daremos **Enter** en la selección del nombre del servidor. Si poseemos un servidor propio con la plataforma OpenShift Origin, escribiremos el nombre de nuestro host.

```
C:\>rhc setup
OpenShift Client Tools (RHC) Setup Wizard

This wizard will help you upload your SSH keys, set your application namespace,
and check that other programs like Git are properly installed.

If you have your own OpenShift server, you can specify it now. Just hit enter to
use the server for OpenShift Online: openshift.redhat.com.
Enter the server hostname: !openshift.redhat.com!
```

Luego nos pedirá darnos de alta, digitando nuestro e-mail y contraseña de la cuenta creada anteriormente.

```
C:\Windows\system32\cmd.exe - rhc setup
from C:/Ruby22/bin/rhc:23:in `load'
from C:/Ruby22/bin/rhc:23:in `'

C:\>gem install net-ssh -v 2.9.3.beta1
Fetching: net-ssh-2.9.3.beta1.gem (100%)
Successfully installed net-ssh-2.9.3.beta1
Parsing documentation for net-ssh-2.9.3.beta1
Installing ri documentation for net-ssh-2.9.3.beta1
Done installing documentation for net-ssh after 3 seconds
1 gem installed

C:\>rhc setup
OpenShift Client Tools (RHC) Setup Wizard

This wizard will help you upload your SSH keys, set your application namespace,
and check that other programs like Git are properly installed.

If you have your own OpenShift server, you can specify it now. Just hit enter to
use the server for OpenShift Online: openshift.redhat.com.
Enter the server hostname: !openshift.redhat.com!

You can add more servers later using 'rhc server'.

Login to openshift.redhat.com: ibarraturismo2015@gmail.com
Password: *****
```

Crearemos las claves de conexión para evitar iniciar sesión a cada instante escribiendo **yes** y pulsando **Enter**.

```

C:\Windows\system32\cmd.exe - rhc setup
Done installing documentation for net-ssh after 3 seconds
1 gem installed

C:\>rhc setup
OpenShift Client Tools (RHC) Setup Wizard

This wizard will help you upload your SSH keys, set your application namespace,
and check that other programs like Git are properly installed.

If you have your own OpenShift server, you can specify it now. Just hit enter to
use the server for OpenShift Online: openshift.redhat.com.
Enter the server hostname: !openshift.redhat.com!

You can add more servers later using 'rhc server'.

Login to openshift.redhat.com: ibarraturismo2015@gmail.com
Password: *****

OpenShift can create and store a token on disk which allows to you to access the
server without using your password. The key is stored in your home directory and
should be kept secret. You can delete the key at any time by running 'rhc
logout'.
Generate a token now? <yes!no> yes

```

También crearemos nuestra clave SSH pública que nos permitirá manejar el entorno de OpenShift, para esto escribiremos **yes** y pulsaremos **Enter**.

```

C:\Windows\system32\cmd.exe - rhc setup
use the server for OpenShift Online: openshift.redhat.com.
Enter the server hostname: !openshift.redhat.com!

You can add more servers later using 'rhc server'.

Login to openshift.redhat.com: ibarraturismo2015@gmail.com
Password: *****

OpenShift can create and store a token on disk which allows to you to access the
server without using your password. The key is stored in your home directory and
should be kept secret. You can delete the key at any time by running 'rhc
logout'.
Generate a token now? <yes!no> yes
Generating an authorization token for this client ... lasts about 1 month

Saving configuration to C:\Users\LuisCisneros\.openshift\express.conf ... done

No SSH keys were found. We will generate a pair of keys for you.

Created: C:/Users/LuisCisneros/.ssh/id_rsa.pub

Your public SSH key must be uploaded to the OpenShift server to access code.
Upload now? <yes!no>
yes

```

Finalmente estará configurada nuestra herramienta cliente para el manejo de la plataforma OpenShift.

```

C:\Windows\system32\cmd.exe
JBoss Unified Push Server      rhc create-app <app name>
1.0.0.Beta1                   jboss-unified-push-1
Jenkins Server                rhc create-app <app name> jenkins-1
Node.js 0.10                  rhc create-app <app name> nodejs-0.10
PHP 5.3                       rhc create-app <app name> php-5.3
PHP 5.4                       rhc create-app <app name> php-5.4
PHP 5.4 with Zend Server 6.1 rhc create-app <app name> zend-6.1
Perl 5.10                     rhc create-app <app name> perl-5.10
Python 2.6                    rhc create-app <app name> python-2.6
Python 2.7                    rhc create-app <app name> python-2.7
Python 3.3                    rhc create-app <app name> python-3.3
Ruby 1.8                      rhc create-app <app name> ruby-1.8
Ruby 1.9                      rhc create-app <app name> ruby-1.9
Ruby 2.0                      rhc create-app <app name> ruby-2.0
Tomcat 6 (JBoss EWS 1.0)      rhc create-app <app name> jbossews-1.0
Tomcat 7 (JBoss EWS 2.0)      rhc create-app <app name> jbossews-2.0
Uvert.x 2.1                   rhc create-app <app name> jboss-vertx-2.1
WildFly Application Server    rhc create-app <app name> jboss-wildfly-8
8.2.0.Final

You are using 0 of 3 total gears
The following gear sizes are available to you: small

Your client tools are now configured.
C:\>

```

### 3 BANCA ADMINISTRADORA

#### 3.1 Lenguaje

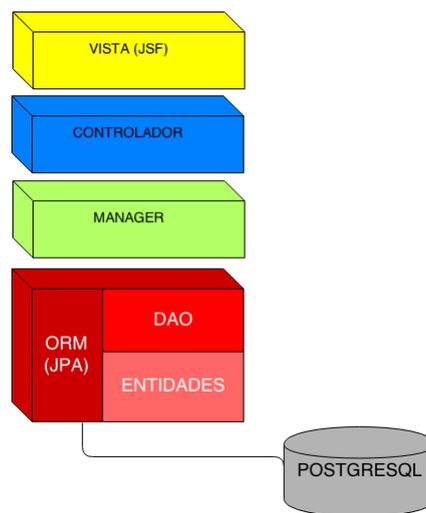
El sistema administrador de banca, está construido en el lenguaje de programación JAVA mediante el framework para aplicaciones web JSF (Java Server Faces), que permite crear aplicaciones J2EE basadas en el patrón MVC.

#### 3.2 Arquitectura

Para el desarrollo del sistema de administración de la aplicación móvil se ha establecido seguir el patrón de arquitectura MVC (Modelo Vista Controlador), ya que este trata de manera separada los datos, la lógica del negocio, y las interfaces de usuario; logrando así un software de calidad.

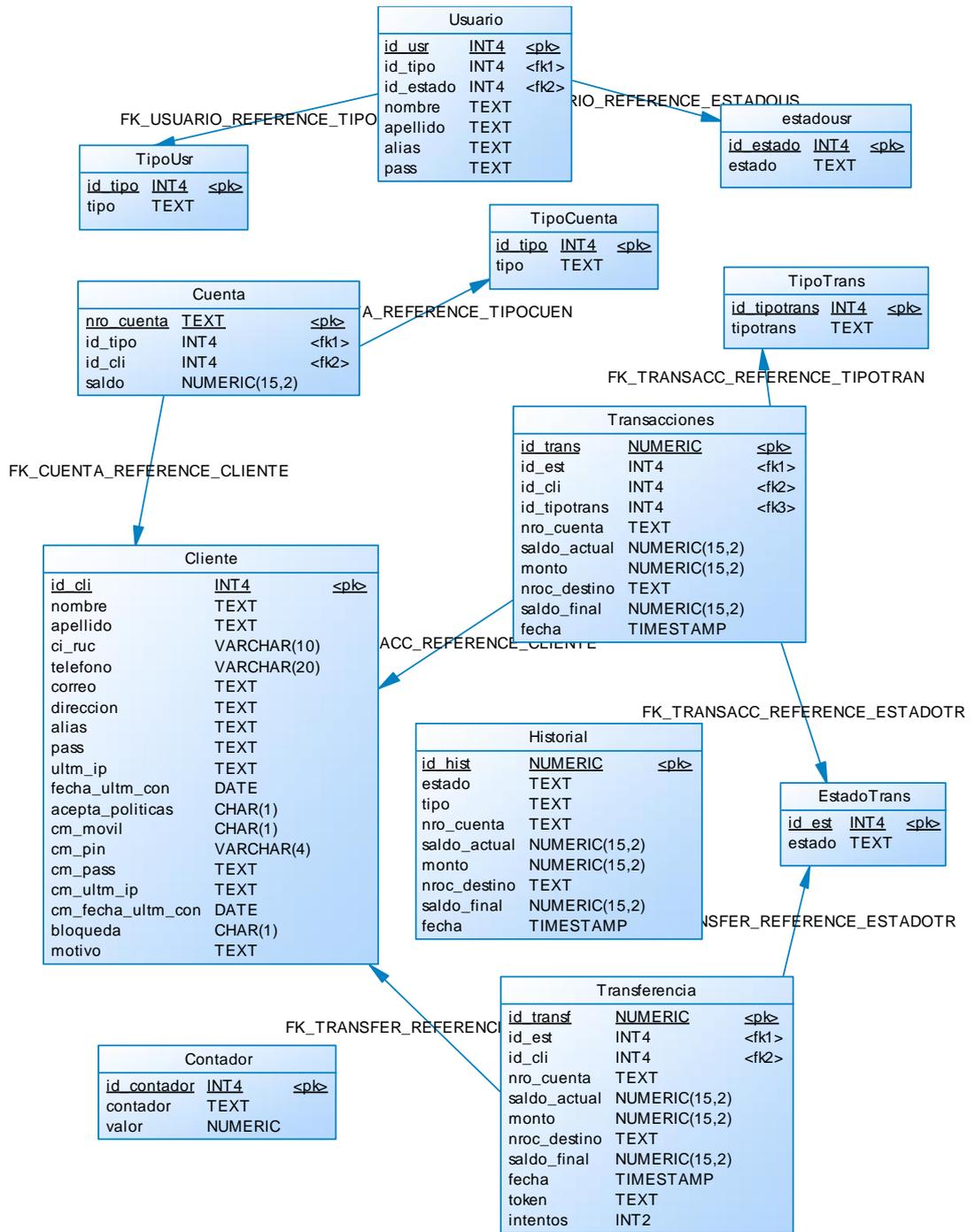
Para detallar un poco más sobre el patrón de arquitectura MVC explicaremos cada uno de sus componentes (capas):

- **Modelo.-** Es aquella capa donde trabajan todos los datos correspondientes a la aplicación. Además gestiona todo el acceso y modificación de la información.
- **Controlador.-** Posee todo el código necesario para responder a las acciones solicitadas de la aplicación.
- **Vista.-** Presenta las distintas interfaces de usuario que serán utilizadas en la aplicación. Estas usualmente son presentadas en código HTML, XHTML, PHP, entre otros.



### 3.3 Diagrama Entidad-Relación

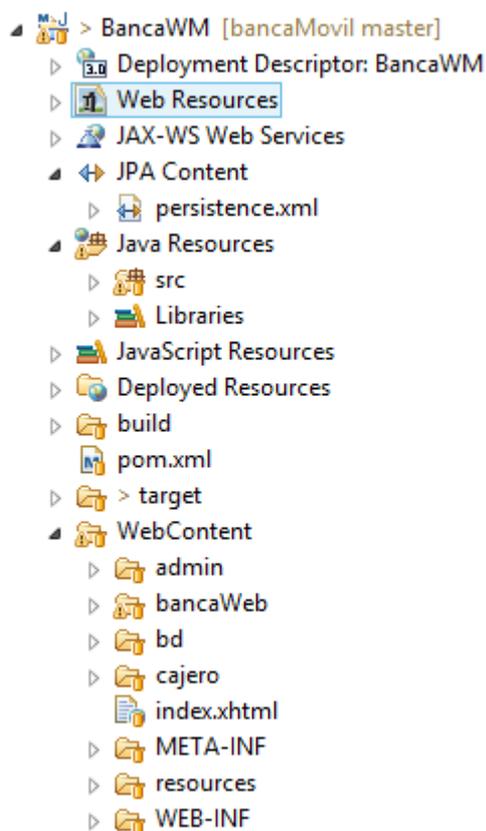
Mediante el diagrama entidad-relación se pretende mostrar el diseño del esquema de base de datos que se usará en el aplicativo.



## 3.4 Proyecto WEB

### 3.4.1 Estructura

Dentro de nuestro IDE de desarrollo, desplegamos la carpeta del proyecto y observamos su estructura en forma de árbol de directorios. Los directorios más importantes serán explicados a continuación:



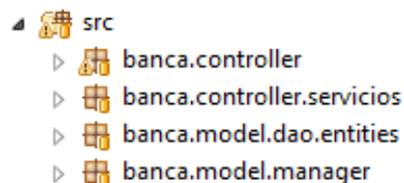
- WebContent: Contiene tanto las páginas web (vistas), como dos directorios que son necesarios para definir la configuración de la aplicación. Estos directorios son:
  - ✓ META-INF: Contiene el fichero context.xml.
  - ✓ WEB-INF: Contiene los ficheros de configuración:
    - faces-config.xml: Contiene instrucciones de configuración que atañen a las clases java de respaldo de las páginas Web. Aquí se inicializan

variables, se enlaza código Java con páginas Web, se indica cómo se salta de una página Web a otra, etc.

- web.xml: Recoge parámetros de configuración generales de la aplicación, tales como cuáles son las páginas de inicio y error, qué recursos adicionales usa la aplicación, temas de seguridad, etc.
- Java Resources: Contiene directorios con los recursos de programación; es decir, paquetes, clases y librerías. Estos directorios son:
  - ✓ Source Packages: Contiene nuestro código Java.
  - ✓ Libraries: Las librerías que necesita nuestro programa.
- JPA Content: Posee la unidad de persistencia de JPA que define un conjunto de todas las entidades (clases) que son gestionadas por la instancia del EntityManager en una aplicación. Este conjunto de clases de entidad representa los datos contenidos en una única base de datos y su conexión hacia la misma. Las unidades de persistencia se definen en el fichero de configuración persistence.xml.

### 3.4.2 Paquetes y clases

El proyecto consta de 6 paquetes que contienen distintas clases JAVA que siguiendo el patrón MVC otorgan funcionalidad al aplicativo.

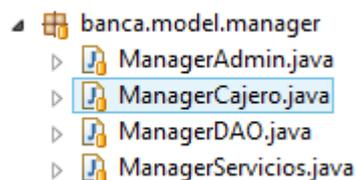


Cada uno de los paquetes será expuesto a continuación:

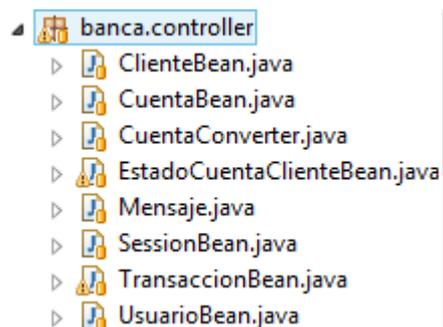
- banca.model.dao.entities.- Posee las clases entidades que representan a una tabla en el modelo de datos relacional y contienen los atributos de las mismas.



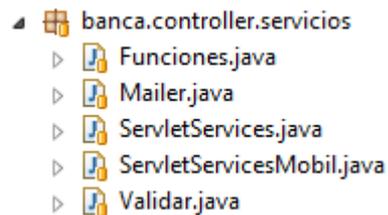
- banca.model.manager.- Posee las clases que permiten el acceso a los objetos representados por las entidades y contiene métodos que gestionan los datos dentro de cada tabla.



- banca.controller.- Posee las clases que interactúan con las vistas.

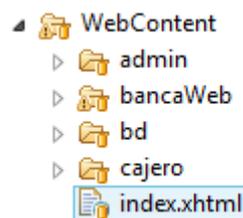


- banca.controller.servicios.- Contiene los Servlets que permite la creación de los servicios REST.

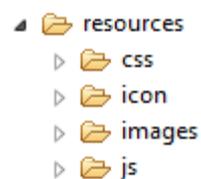


### 3.4.3 Vistas

Las representaciones de las distintas interfaces de usuario se encuentran en el WebContent, la vista principal es **index.html** y las otras se encuentran respectivamente en las carpetas **admin** y **cajero**.



Cabe recalcar que los recursos CSS y JavaScript, como las imágenes que usan cada una de las vistas se encuentran dentro de la carpeta **resources**.



## 4 BANCA WEB

Cada cliente del banco puede acceder a sus cuentas mediante el uso de la banca web, este sistema ha sido construido con el framework AngularJS y archivos HTML

para las vistas. Además se usa Bootstrap para mejorar la calidad de la usabilidad del cliente.

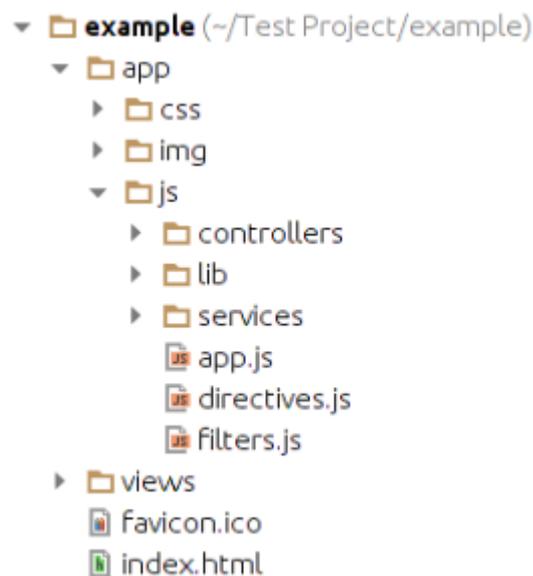
## 4.1 Angular JS

AngularJS es un framework javascript que hace uso del MVC (modelo, vista, controlador), con lo cual obtendremos un trabajo ordenado y escalable.

Para conseguirlo debemos ingresar a <https://angularjs.org/> y pulsar el botón de descargas.

## 4.2 Estructura

La siguiente imagen representa un modelo de cómo quedaría organizado nuestro proyecto de una manera sencilla y fácil de entender.



En esta estructura colocaremos cada uno de los elementos de nuestro proyecto que hemos identificado anteriormente, quedando de la siguiente manera:

- Las **Vistas** (archivos HTML) estarán dentro de la carpeta **/views**

- Los **archivos de configuración** de AngularJS (app.js, directives.js, etc.) estarán dentro de la carpeta **/app/js**
- Los **Controladores** estarán dentro de la carpeta **/app/js/controllers**
- Los **Services** estarán dentro de la carpeta **/app/js/services**
- Las **librerías de javascript** estarán dentro de la carpeta **/app/js/lib**
- Los **archivos CSS** dentro de **/app/css**
- Las **imágenes** dentro de **/app/img**
- El archivo principal **index.html** obviamente estará en el raíz

## 5 APLICACIÓN MÓVIL

### 5.1 Tipo de Aplicación

El cliente móvil desarrollado forma parte de las aplicaciones móviles híbridas, estas son creadas con tecnologías Web: HTML5, JavaScript y se ejecutan sobre un contenedor nativo utilizando el motor de un navegador móvil para procesar su contenido de manera local (WebView).

### 5.2 Plataforma de desarrollo

Apache Cordova (<http://cordova.apache.org>) es un framework gratuito de código abierto, usado para la construcción de aplicaciones multiplataforma nativas con tecnologías web (HTML, CSS y JavaScript). Córdoba implementa un conjunto de APIs que extiende el uso de las capacidades nativas de los dispositivos (tales como la cámara, acelerómetro, la aplicación contactos, entre otras.) para la creación de una aplicación.

## 5.3 Proyecto CORDOVA

### 5.3.1 Estructura de Archivos

Al crear un proyecto CORDOVA aparecerá un directorio con el nombre de nuestra aplicación, dentro de esta encontramos diferentes carpetas. Las más importantes son ***platforms*** y ***www***.

Name	Date modified	Type
.cordova	1/31/2014 12:41 AM	File folder
merges	1/31/2014 1:51 AM	File folder
platforms	1/31/2014 1:50 AM	File folder
plugins	2/2/2014 6:52 PM	File folder
www	2/16/2014 8:03 PM	File folder

PLATFORMS contiene las plataformas de los distintos sistemas operativos soportados por CORDOVA, si en nuestro caso trabajamos el proyecto para el sistema operativo Android; dentro de esa carpeta encontraremos una carpeta con el nombre android.

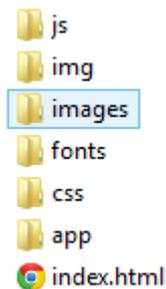
La carpeta WWW contiene todos los archivos y directorios relacionados con la estructura de nuestra aplicación. Esta estructura de directorios y archivos es la misma que una aplicación web cliente.

Name	Date modified	Type
css	4/1/2014 11:28 PM	File folder
img	4/1/2014 11:28 PM	File folder
js	4/1/2014 11:28 PM	File folder
config.xml	4/1/2014 11:28 PM	XML File
index.html	4/1/2014 11:28 PM	Opera Web Docu...

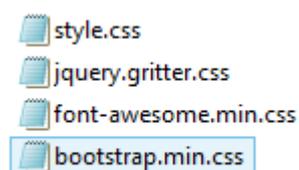
La única diferencia que encontraremos es un archivo “config.xml”, el cual no deberemos borrar, todo lo demás podemos borrarlo y pegar nuestra aplicación web; ya que este archivo contiene la configuración esencial de nuestro proyecto CORDOVA. Si revisamos ese archivo podemos ver que hay información del proyecto; como el nombre, con que archivo va a iniciar la aplicación (index.html), y datos de plugins si en un caso usamos.

### 5.3.2 Archivos utilizados

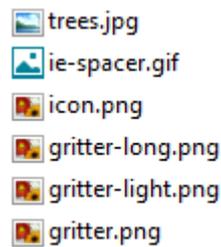
El proyecto web consta de la estructura de archivos donde existen carpetas para almacenar los archivos de extensiones .css y .js correspondientemente. También consta de una carpeta para almacenar las imágenes usadas y de nuestro archivo principal index.html.



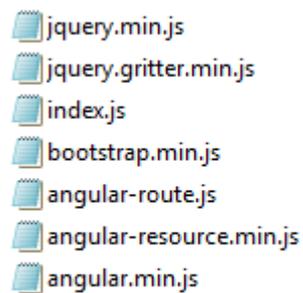
Dentro de la carpeta **css** se muestran los archivos correspondientes a los estilos del framework JQuery Mobile, nuestro theme personalizado y otros estilos personalizados. A demás consta de una carpeta para almacenar fuentes e imágenes de nuestro theme.



Como explicamos anteriormente existe una carpeta donde almacenamos las imágenes usadas dentro de nuestro diseño, las cuales no pertenecen al theme.



En la carpeta **js** se encuentran los archivos de JQuery y AngularJS, que son esenciales para el uso de este framework. También está nuestro archivo de la funcionalidad para la aplicación que es **index.js**.



Además en la carpeta **app** tenemos los archivos Angular con la estructura de controladores, rutas y servicios.

