

# MML

---

MANUAL DE TÉCNICO

Bastidas Fabian  
UNIVERSIDAD TÉCNICA DEL NORTE



# Contenido

|   |    |
|---|----|
| 1. Introducción.....  | 1  |
| 2. Alcance.....   | 1  |
| El sistema está montado sobre una plataforma dentro del internet lo que permite que cualquier persona con acceso a internet puede hacer uso del sistema. E único requisito es la previa creación de un usuario para poder acceder. .... |    |
| 3. Usos.....  | 1  |
| 4. Requisitos.....  | 1  |
| 5. Herramientas.....  | 1  |
| 5.1. GIT.....   | 1  |
| 5.2. Openshift .....  | 5  |
| Creación de cuenta .....  | 6  |
| Registro de dominio .....   | 8  |
| Creación de una aplicación .....  | 9  |
| Instalación del cliente de OpenShift .....  | 12 |
| Ruby .....  | 12 |
| Gema RHC .....  | 14 |
| 5.3. MongoDB.....   | 16 |
| Instalar MongoDB como un servicio de Windows .....  | 18 |
| 5.4. NodeJS.....  | 19 |
| 6. Diagrama de entidades .....  | 22 |
| 7. Modelo de datos .....  | 24 |
| 8. Lista de servicios REST .....  | 30 |
| 9. Lista de rutas angular .....   | 31 |



## 1. Introducción

MML es un sistema para gestionar los primeros pasos de la generación de proyectos mediante la metodología de **MATRIZ DE MARCO LÓGICO**, está diseñado para que un usuario que tenga almacenados de manera virtual varios proyectos, además de poder compartirlos con otras personas para poder ver el avance y facilitar el trabajo en equipo.

## 2. Alcance

**El sistema está montado sobre una plataforma dentro del internet lo que permite que cualquier persona con acceso a internet puede hacer uso del sistema. E único requisito es la previa creación de un usuario para poder acceder.**

Para acceder al mismo simplemente accederemos a la siguiente dirección web en cualquier navegador, de preferencia se recomiendo tener navegadores actualizados.

**URL:** <http://mml-mfbg.rhcloud.com/#/>

## 3. Usos

El sistema netamente sirve para crear, almacenar y compartir proyectos con formato de **MATRIZ DE MARCO LÓGICO**.

## 4. Requisitos

Los requisitos para poder hacer uso del sistema son los siguientes:

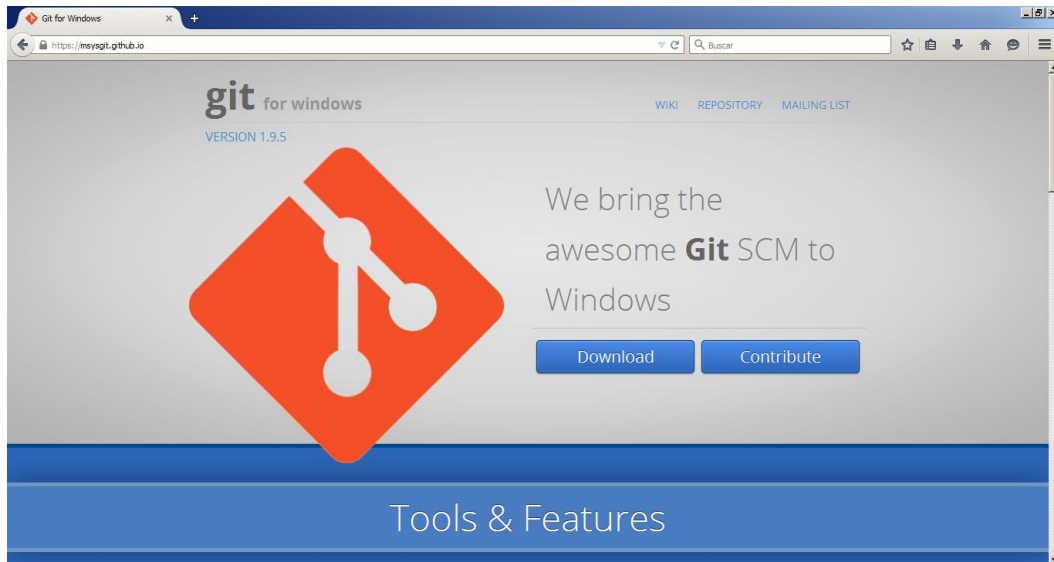
- Equipo con acceso a internet.
- Navegador web (Preferiblemente Google Chrome).

## 5. Herramientas

### 5.1. 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.

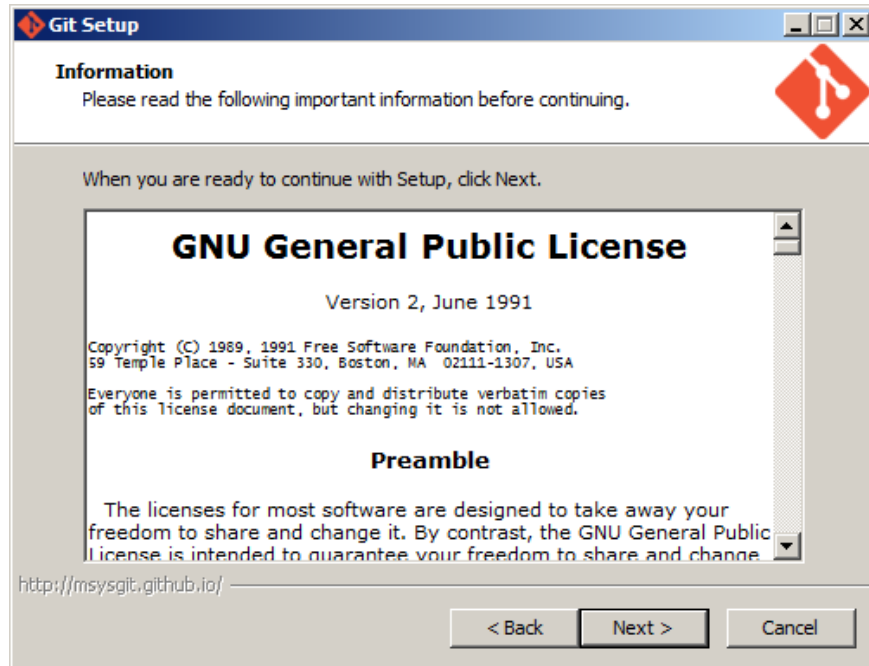
La instalación del software es muy sencilla, primero nos dirigimos a la siguiente dirección web <https://msysgit.github.io/> y pulsaremos el botón descargar.



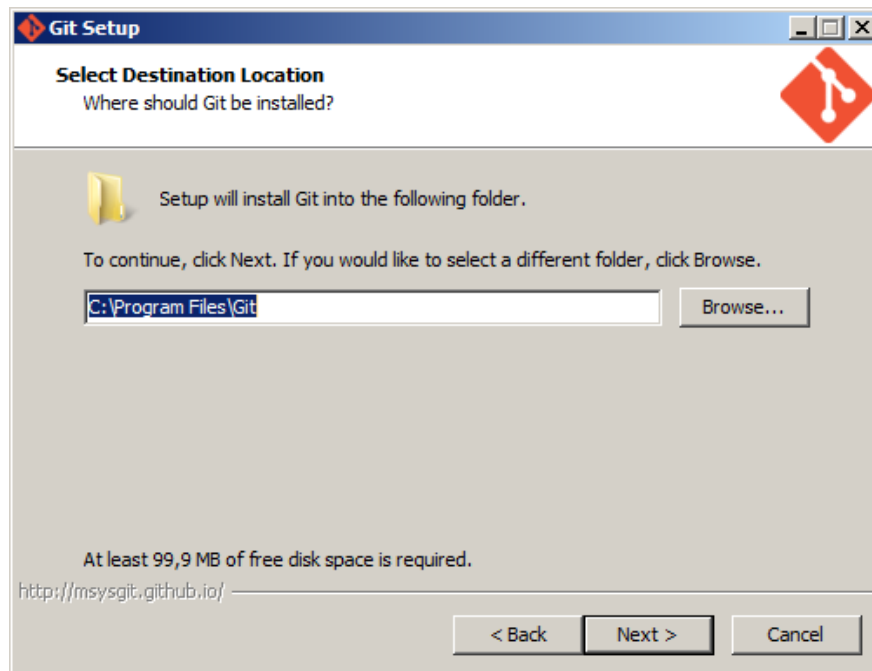
Una vez descargado el instalador, lo ejecutamos iniciando así el proceso de instalación dentro del computador.



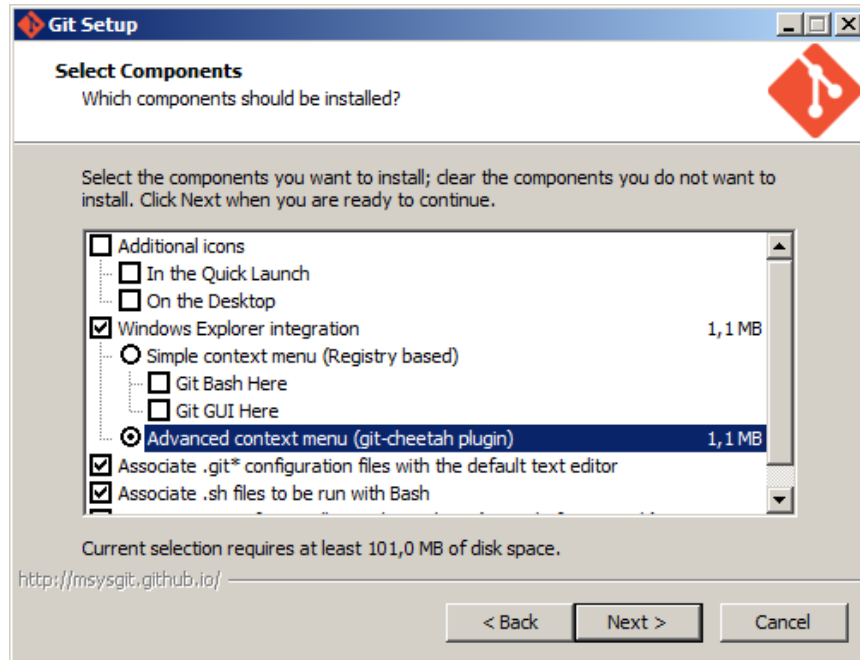
Aceptaremos los términos y condiciones de la licencia.



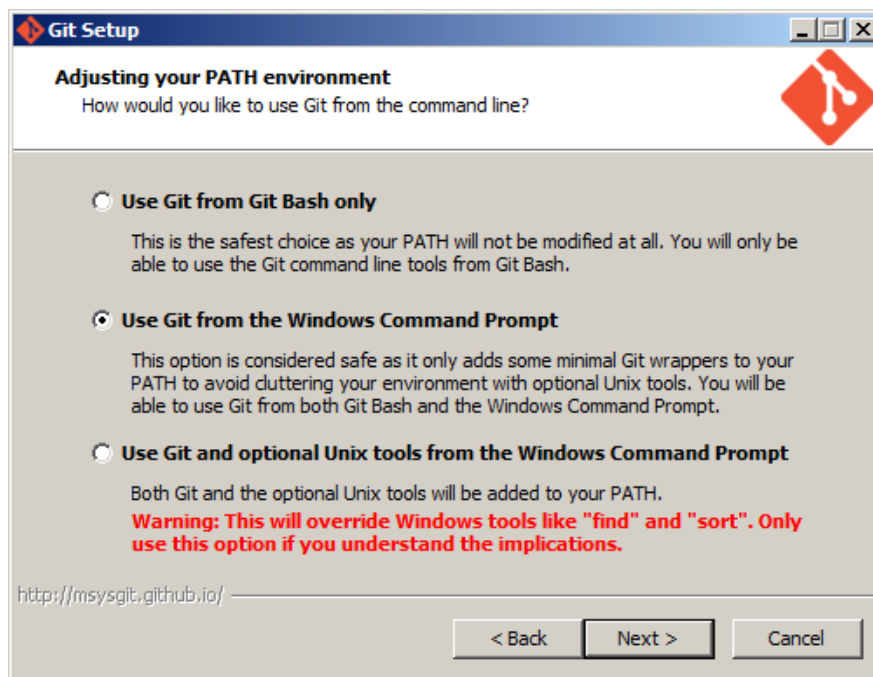
Elegimos el directorio de instalación.



Seleccionamos los componentes a instalar.

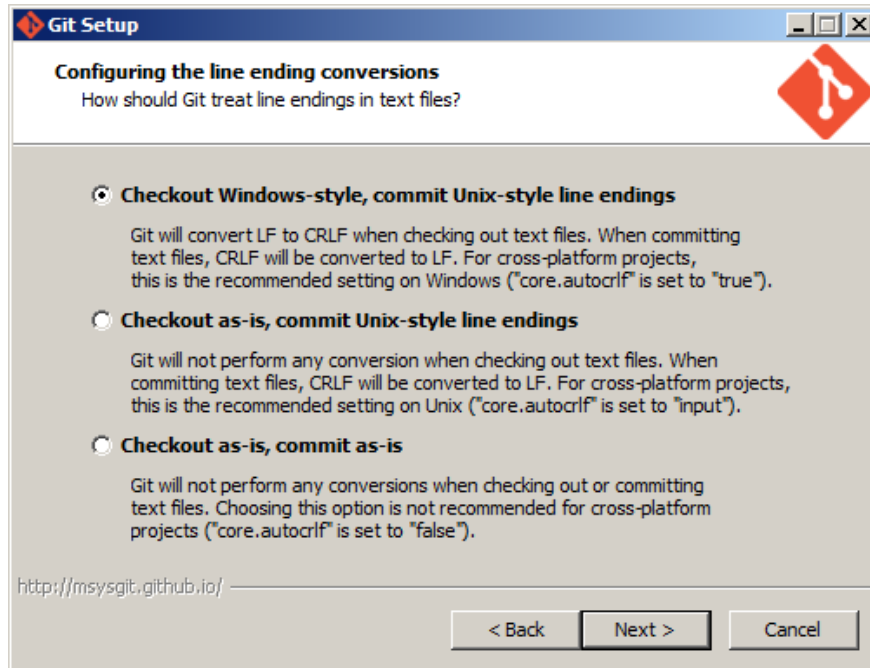


Asignaremos las variables a nuestro PATH.

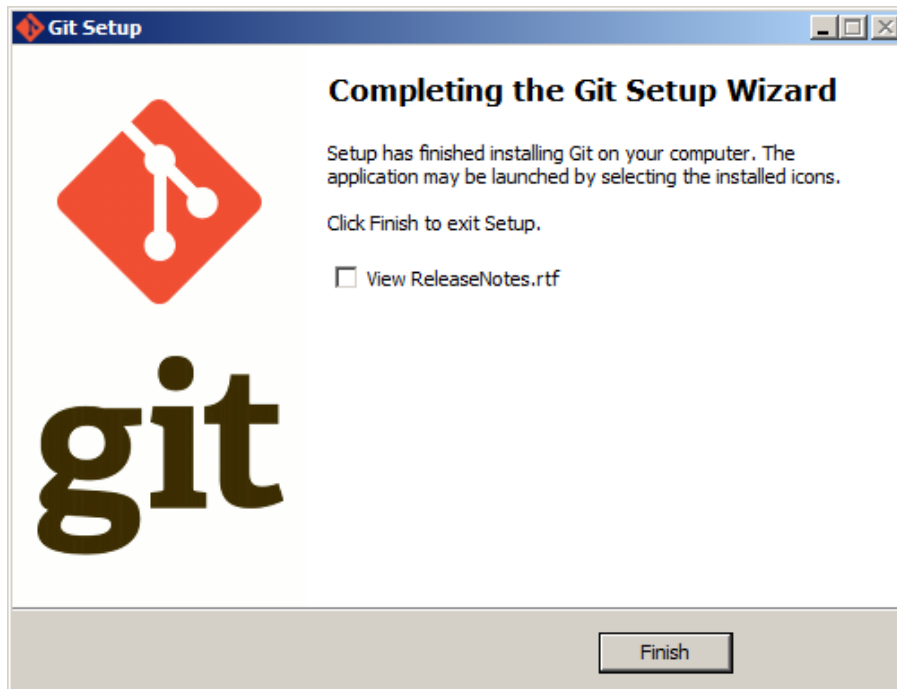


Dejaremos las últimas configuraciones por defecto e inicia el proceso de instalación.





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

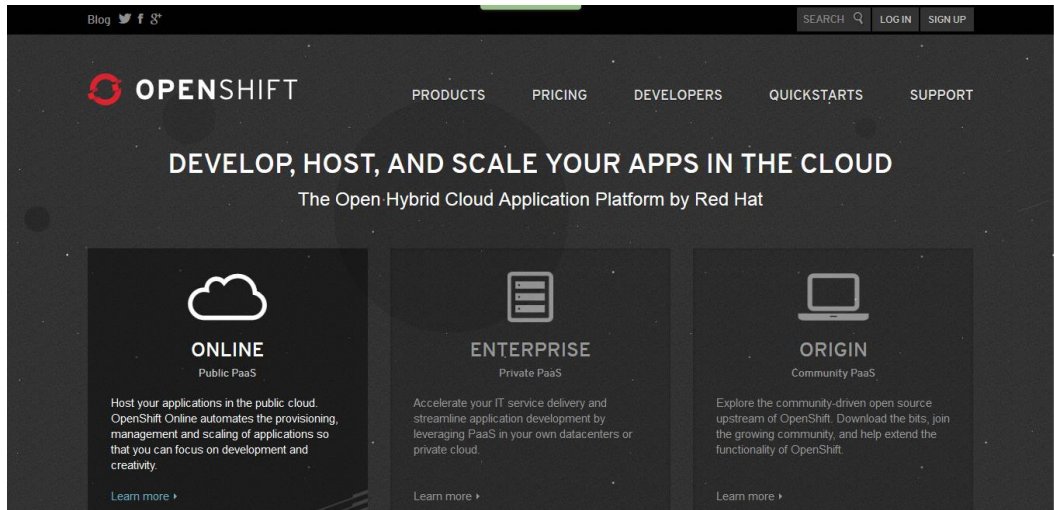


## 5.2. Openshift

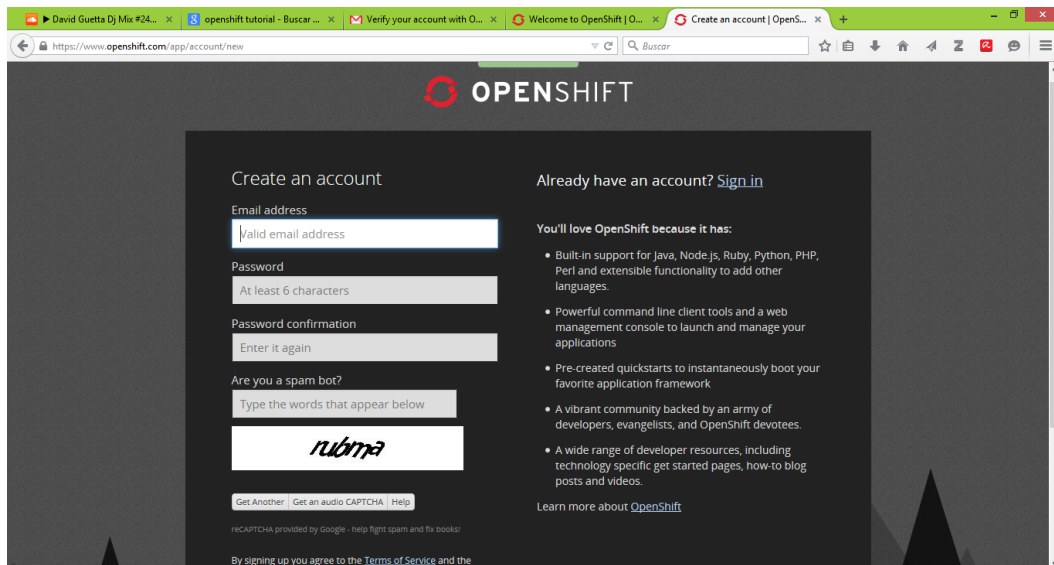
Es una plataforma en la nube que permite despliegue de aplicaciones de manera fácil y segura.

## Creación de cuenta

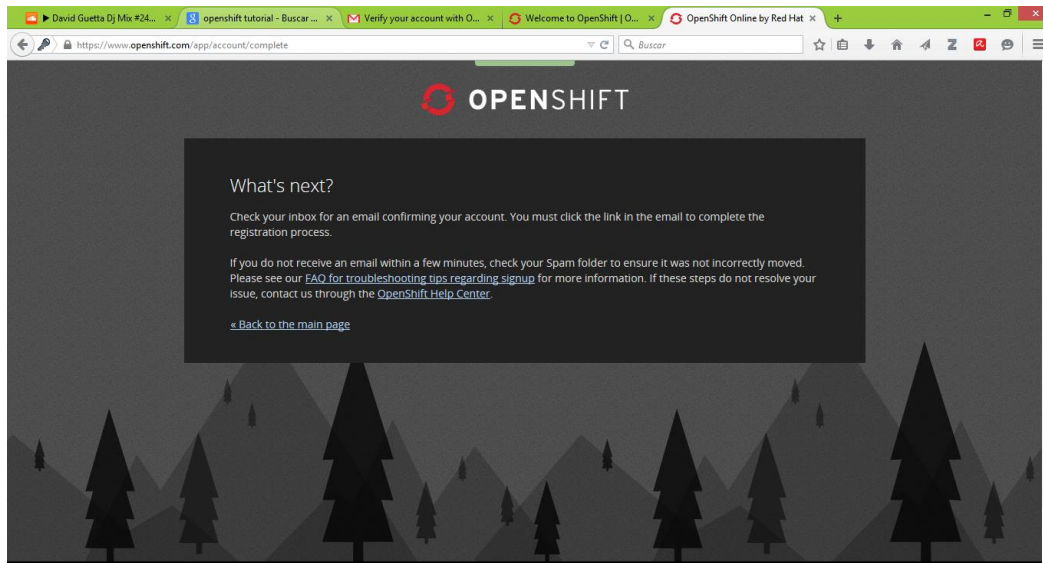
Para crear una cuenta nos dirigimos a <https://www.openshift.com/>, y pulsamos el botón **Sign Up**, que se encuentra en la parte superior derecha.



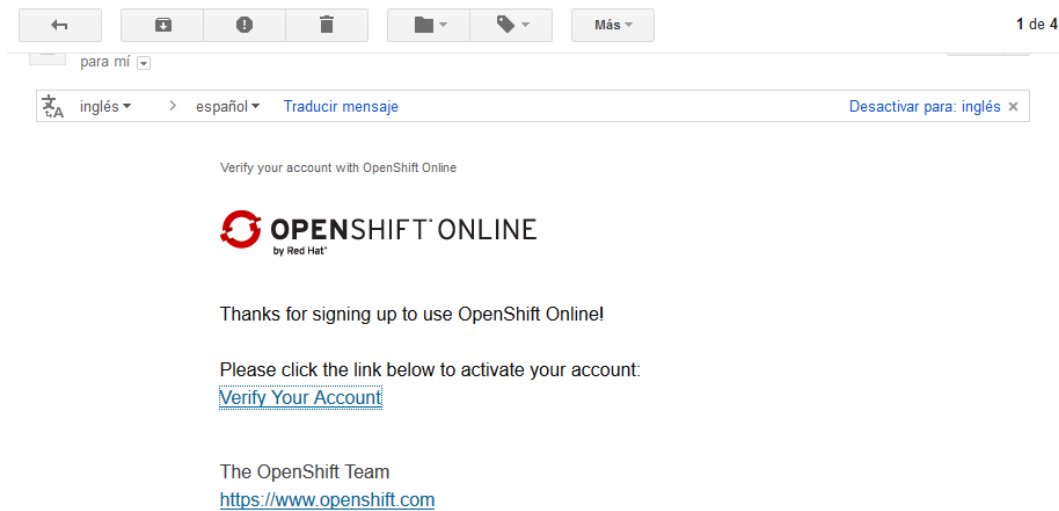
Luego se mostrara un formulario de registro donde ingresaremos los datos de nuestra cuenta, y pulsaremos el botón **Sign Up**.



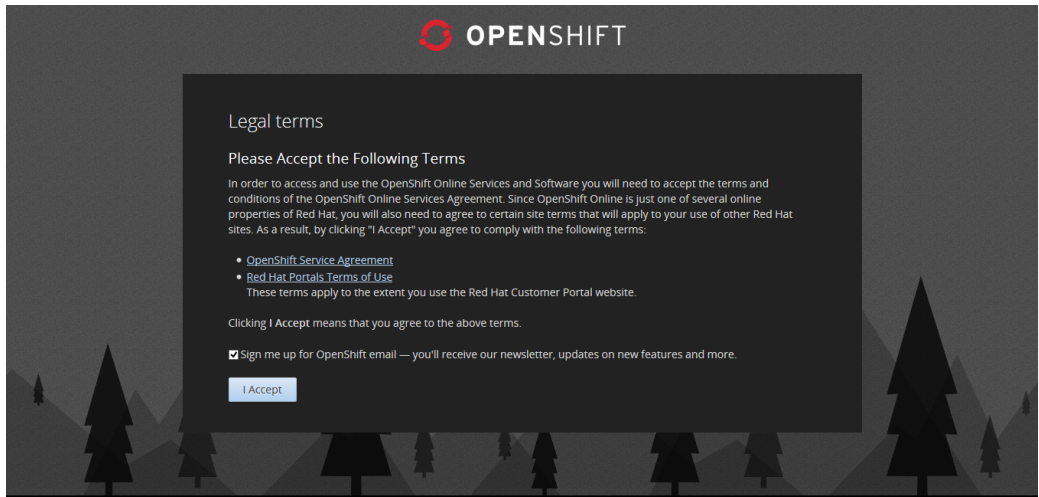
A continuación nos aparecerán las indicaciones para activar nuestra cuenta ya creada, en resumen no indican que debemos realizar este proceso mediante nuestro correo electrónico.



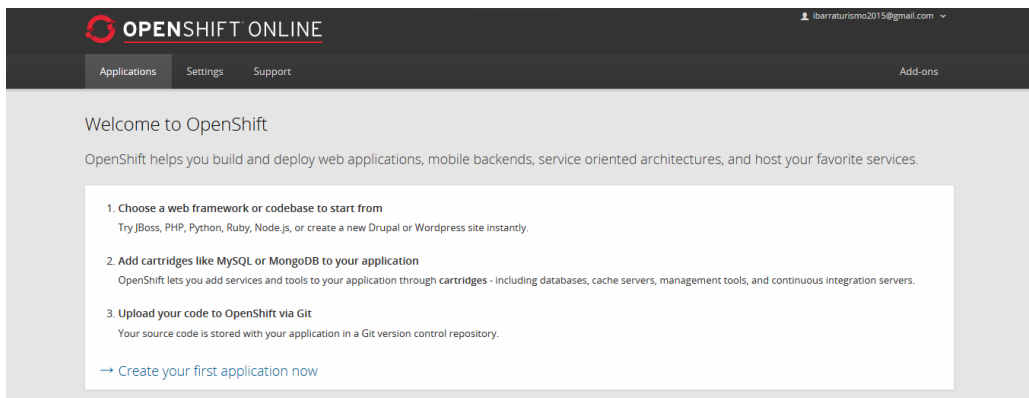
Al abrir nuestro mensaje dentro del correo electrónico, nos aparece un enlace de activación de cuenta.



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

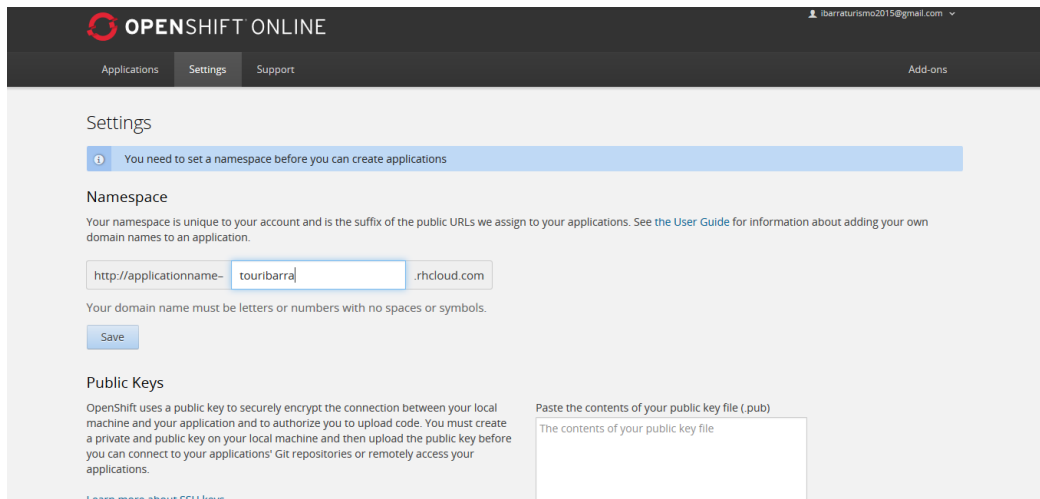


Finalmente se muestra nuestro escritorio de trabajo virtual de OpenShift.

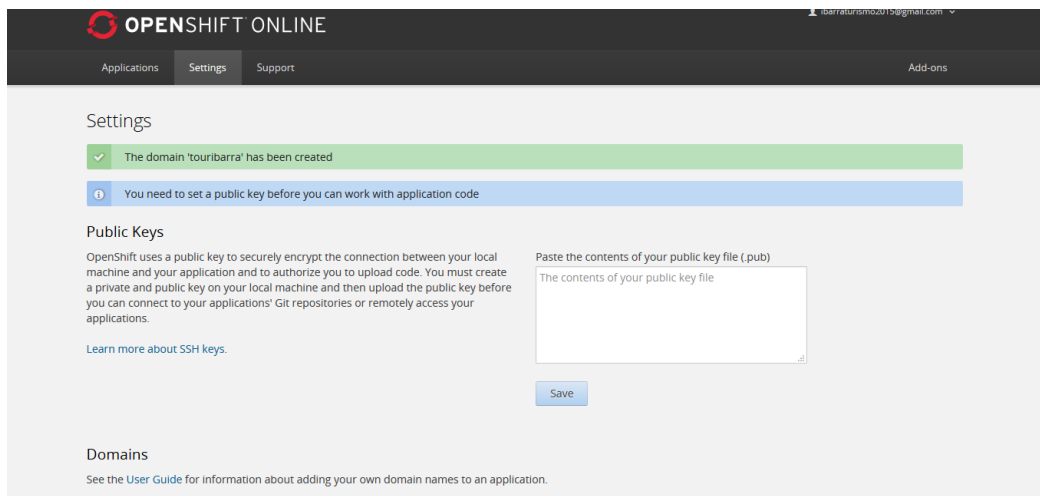


## Registro de dominio

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.

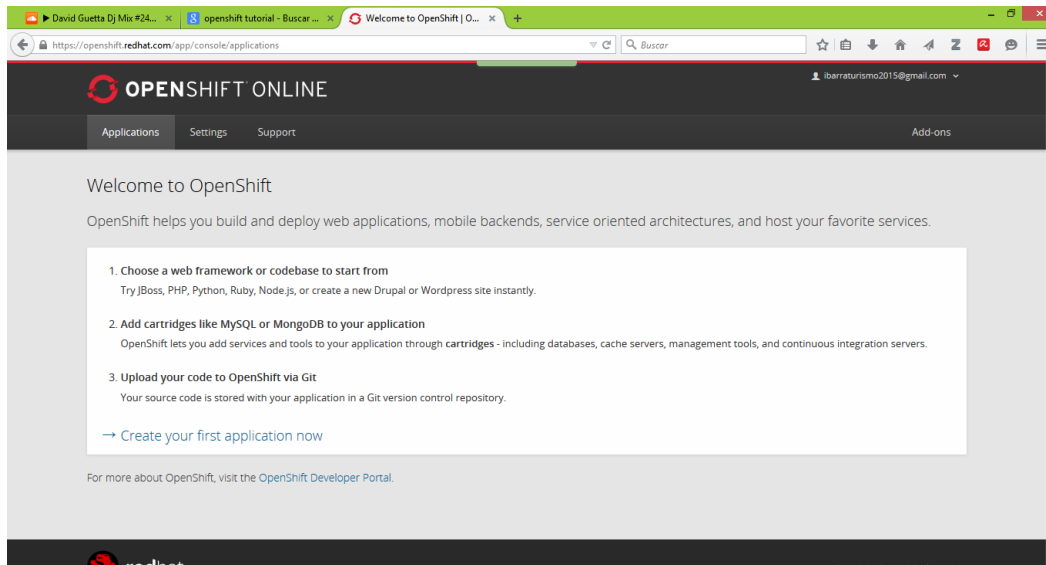


Al presionar el botón **Save** finalizaremos el proceso de creación de nuestro dominio y nos aparecerá un mensaje de éxito.

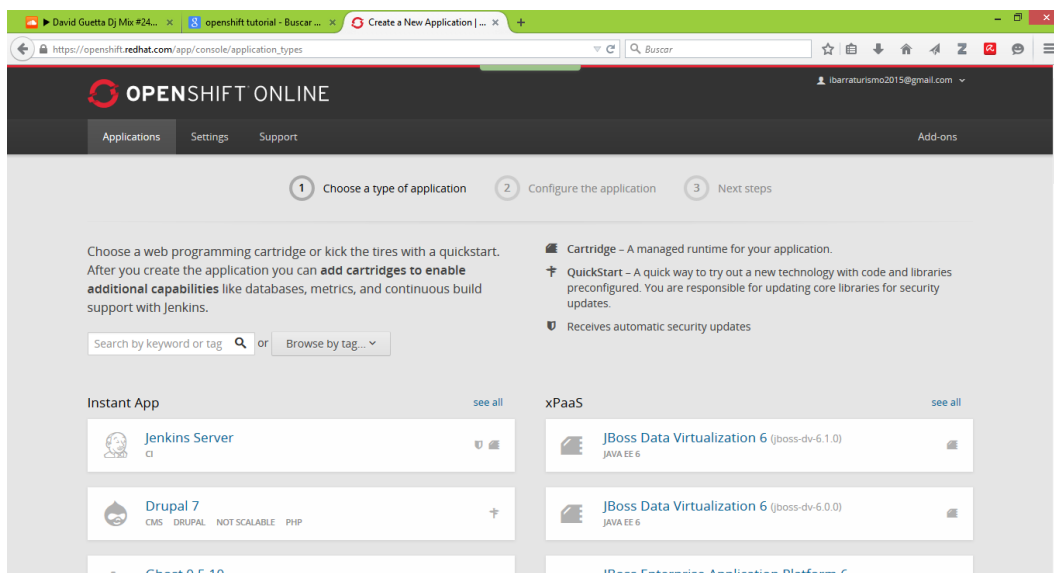


## Creación de una aplicación

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 [deploy 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**

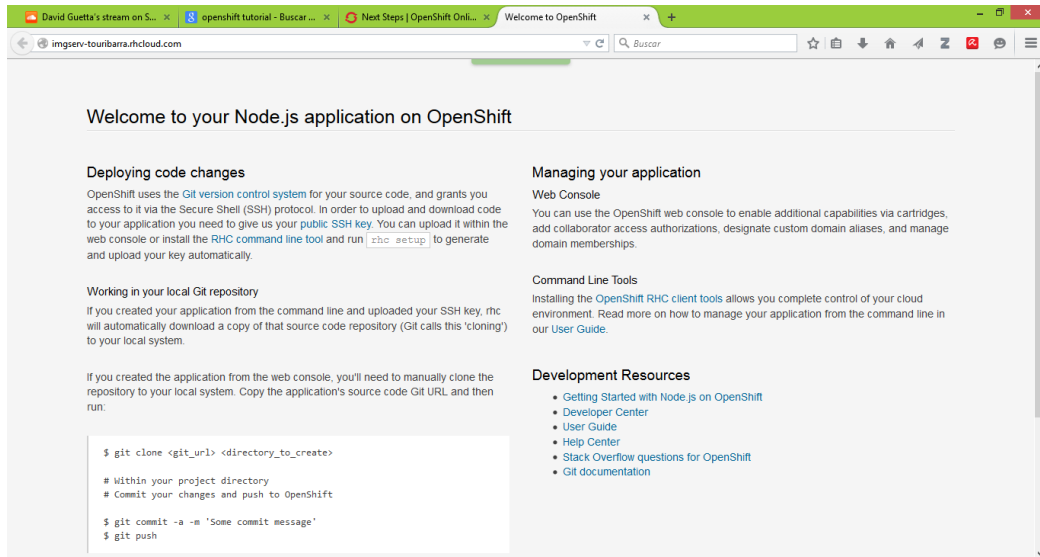
All of the capabilities of OpenShift are exposed through our command line tool, `rhc`. [Follow these steps to install the client](#) on Linux, Mac OS X, or Windows.

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.

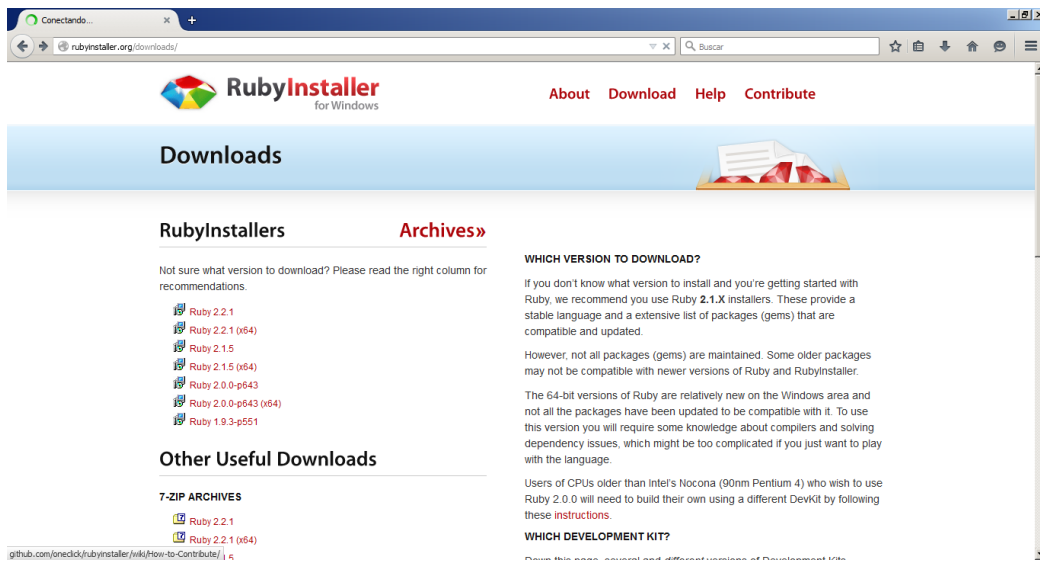


## Instalación del cliente de OpenShift

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

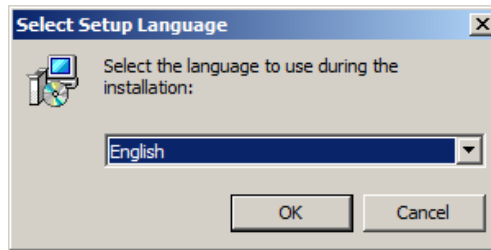
### Ruby

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

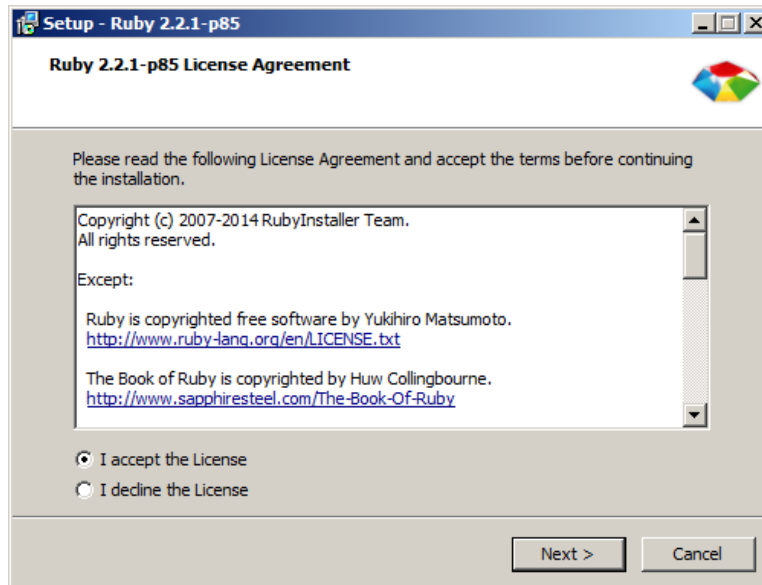


Al ejecutar el instalador nos mostrará un mensaje de selección de idioma para la instalación, lo dejaremos en English por defecto.

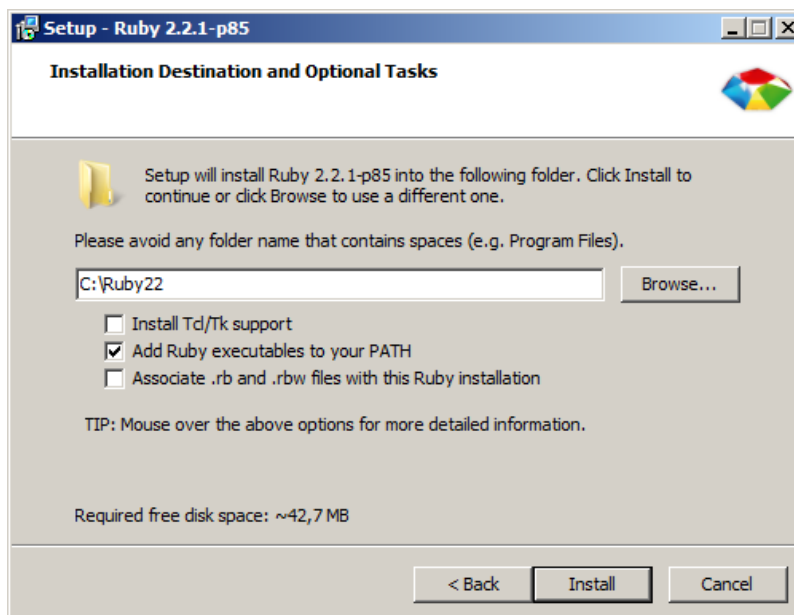




Luego aceptaremos las condiciones de la licencia del lenguaje de programación.



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>
```

### 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>
```

### 5.3. MongoDB

MondoDB es una Base de Datos NOSQL. Por desgracia, MondoDB no está pensado para equipos de 32 bits ni para Windows XP:

La versión 32 bits de MongoDB no soporta Bases de Datos superiores de 2 GB.

Desde la versión MongoDB 2.08, MongoDB ya no funciona para Windows XP, sólo funciona para Windows Vista, Windows 7, Windows 8, etc.



# mongoDB

Primero, descargar la última versión disponible de MongoDB desde:

<http://www.mongodb.org/downloads>

Extraer en **C:\mongodb** de modo que la carpeta **bin** esté en **C:\mongodb\bin**.

MongoDB necesita un directorio donde guardar los datos. Por defecto, este directorio que hay que crear es, **C:\data\db** (Hay que crear el directorio data y el directorio db).

Si se quiere utilizar otro directorio, por ejemplo **C:\mongodb\data\db**, primero hay que crear estas carpetas. A continuación, hay que ejecutar la siguiente instrucción en la terminal de comandos cmd:

```
C:\mongodb\bin\mongod.exe --dbpath C:\mongodb\data
```

Para poder recibir conexiones desde clientes, hay que iniciar el Servidor de la Base de Datos. Para ello, desde la terminal de comandos (cmd) ejecutar:

```
C:\mongodb\bin\mongod.exe
```

Ahora desde otra terminal de comandos (cmd) realizar una conexión al servidor de la Base de datos, por defecto automáticamente en la IP 127.0.0.1 puerto 27017,

```
C:\mongodb\bin\mongo.exe
```

Intentar insertar un registro en la colección por defecto test y luego realizar una consulta,

```
db.test.save( { a: 1 } )
```

```
db.test.find()
```

## Instalar MongoDB como un servicio de Windows

Para no tener que iniciar manualmente el servidor de la Base de Datos cada vez que se reinicia Windows, se va a instalar como un servicio de Windows para que el servidor se inicie automáticamente cada vez que se reinicia el equipo.

Para ello, primero hay que crear un directorio para el registro de eventos:

```
C:\mongodb\log
```

Luego, hay que crear un fichero de configuración para la opción **logpath** de MongoDB. Desde la terminal de comandos ejecutar:

```
echo logpath=C:\mongodb\log\mongo.log > C:\mongodb\mongod.cfg
```

A continuación, abrir una terminal de comandos con permisos de Administrador y ejecutar

```
C:\mongodb\bin\mongod.exe --config C:\mongodb\mongod.cfg --install
```

Iniciar el servicio:

```
net start MongoDB
```

Para detener el servicio hay que ejecutar

```
net stop MongoDB
```

Para eliminar el servicio, hay que ejecutar

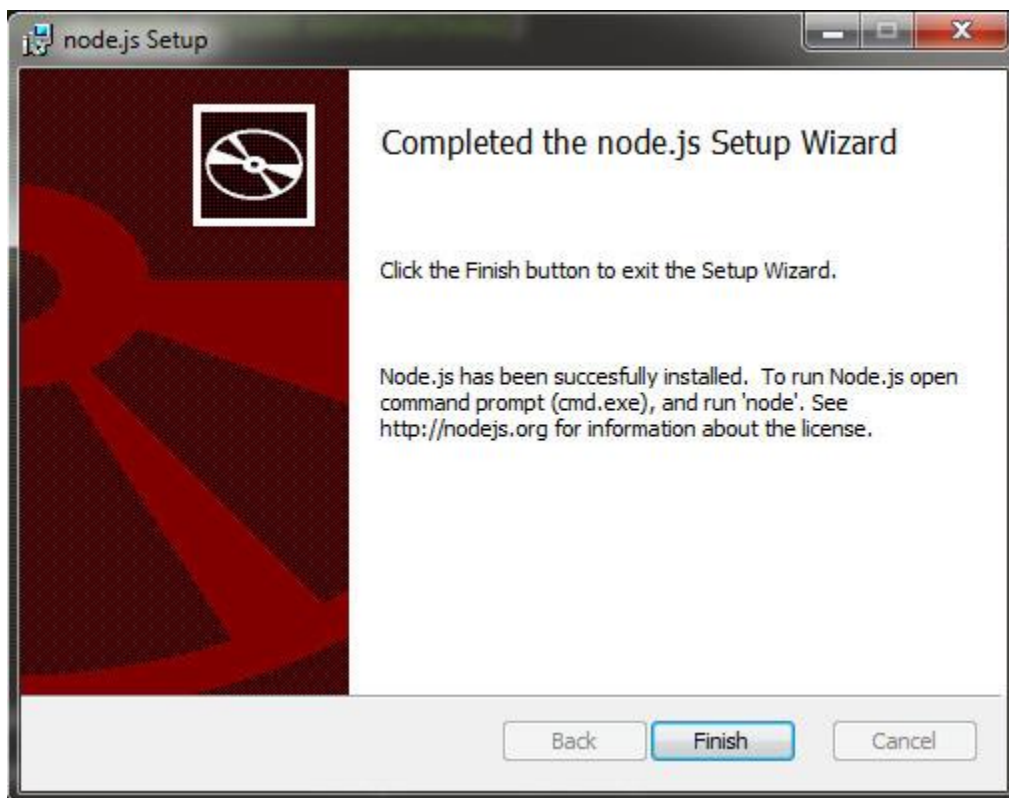
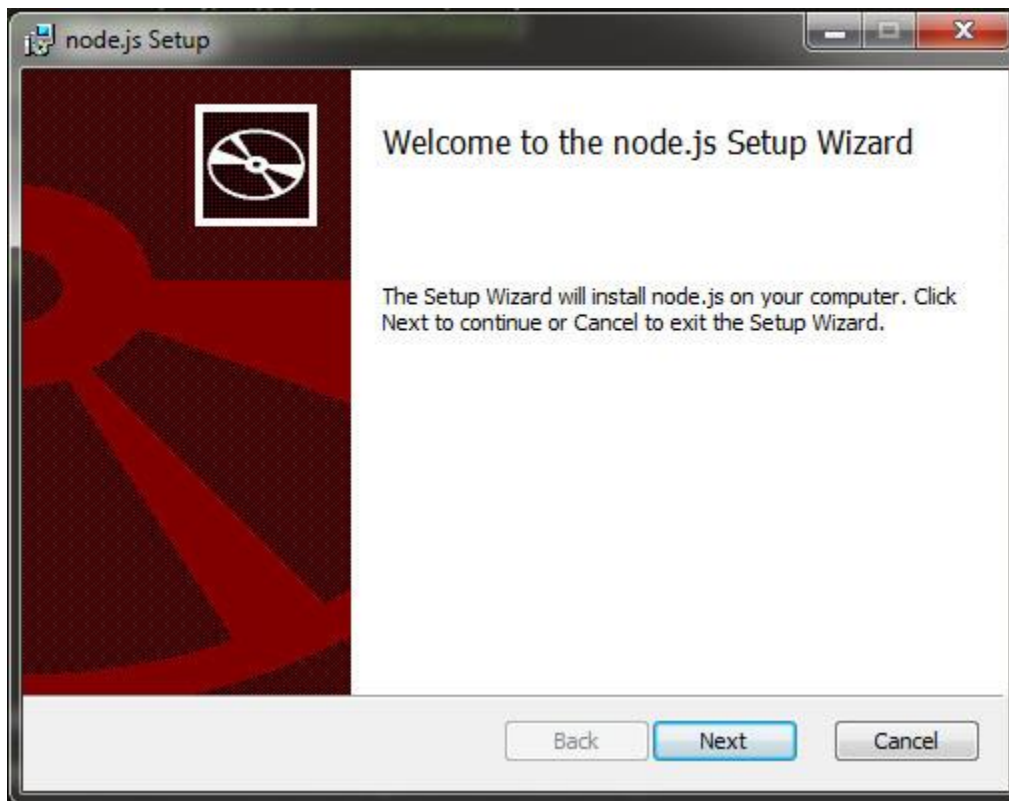
```
C:\mongodb\bin\mongod.exe --remove
```

```
CA Simbolo del sistema
C:\mongodb\bin>
C:\mongodb\bin>
C:\mongodb\bin>echo logpath=C:\mongodb\log\mongo.log > C:\mongodb\mongod.cfg
C:\mongodb\bin>C:\mongodb\bin\mongod.exe --config C:\mongodb\mongod.cfg --install
all output going to: C:\mongodb\log\mongo.log
C:\mongodb\bin>
C:\mongodb\bin>net start MongoDB
El servicio de Mongo DB se ha iniciado con éxito.
C:\mongodb\bin>
```

## 5.4. NodeJS

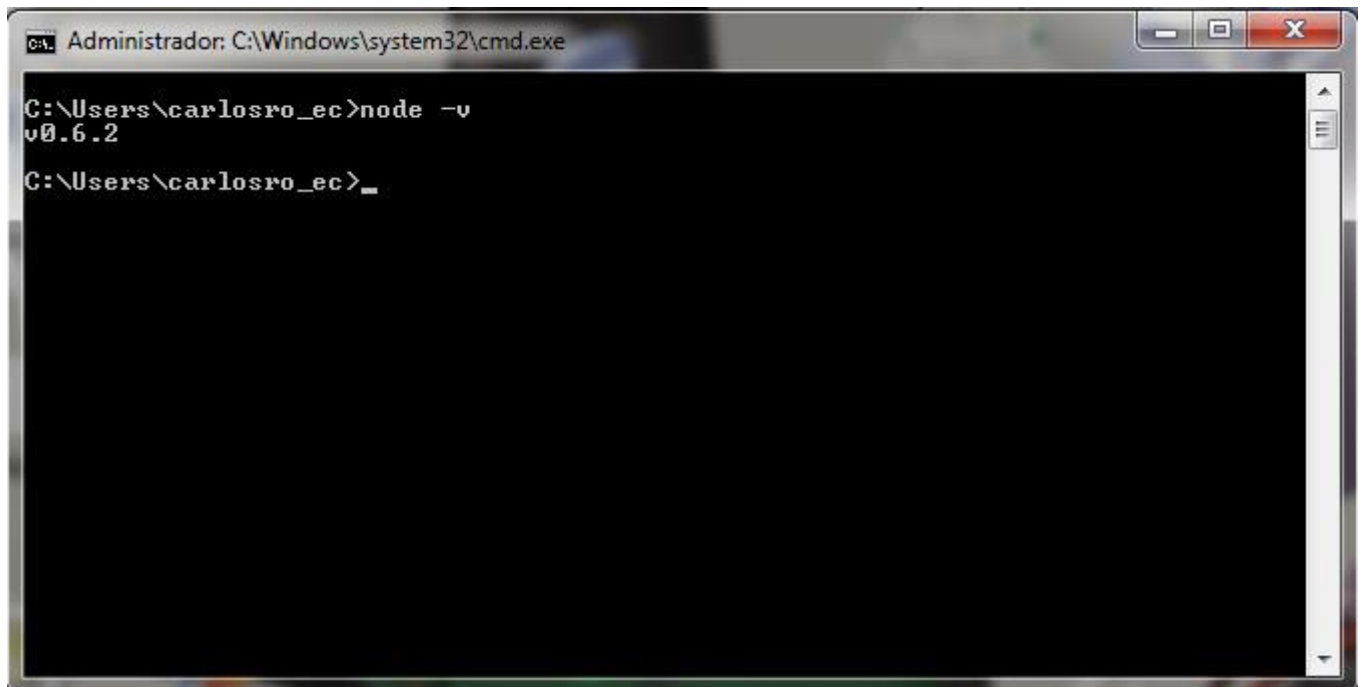
Para instalar Node.js simplemente debemos descargar el paquete msi desde node.msi Windows installer, con esto simplemente seguimos las indicaciones.







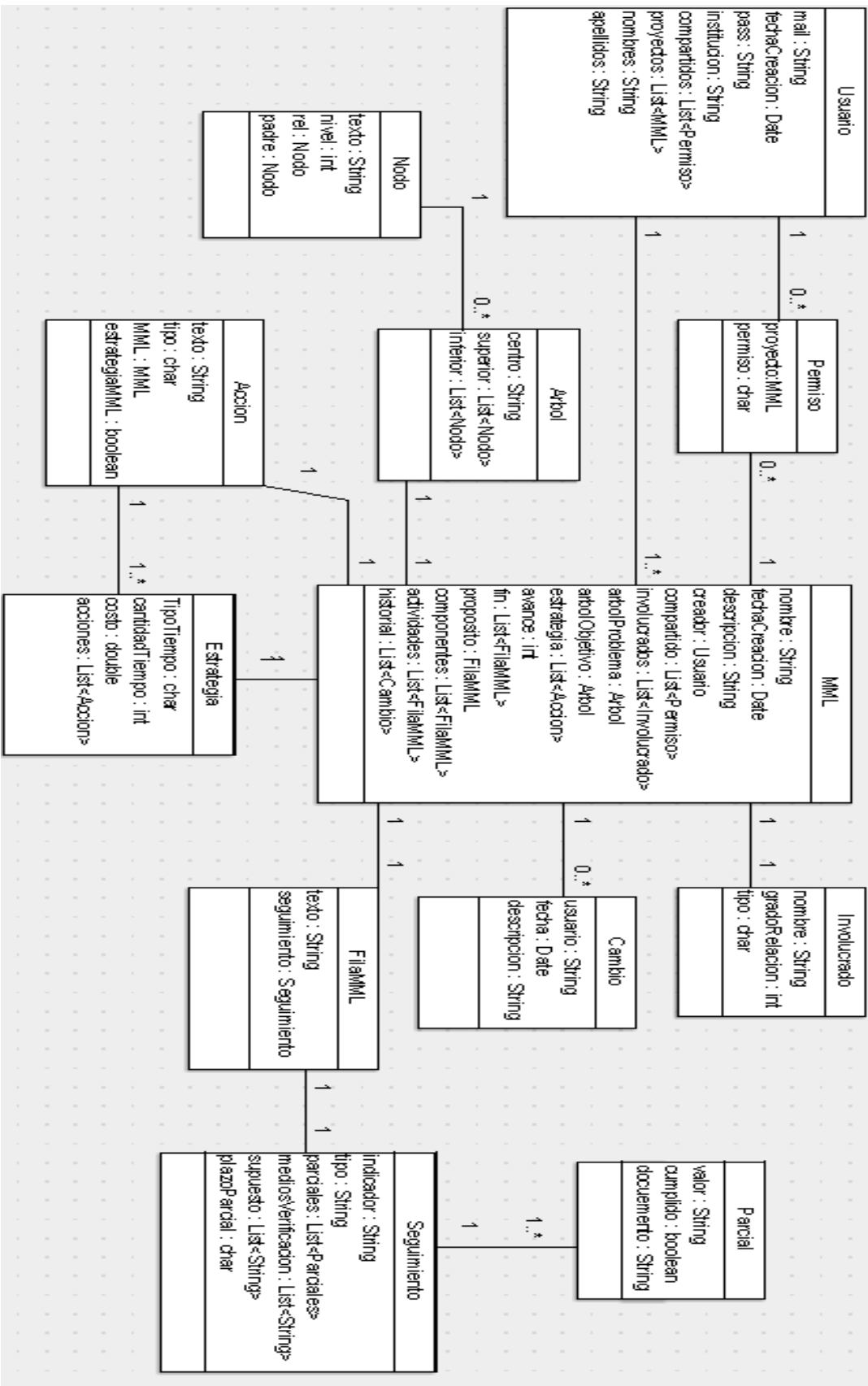
Una vez que finaliza la instalación lo cual no toma más de un minuto reiniciamos nuestro equipo y ejecutamos una consola en la cual verificamos la versión que acabamos de instalar.



```
CA. Administrador: C:\Windows\system32\cmd.exe
C:\Users\carlosro_ec>node -v
v0.6.2
C:\Users\carlosro_ec>
```

Como respuesta obtenemos v0.6.2 lo cual es correcto, con esto estamos listos para probar nuestra instalación.

## 6. Diagrama de entidades



## 7. Modelo de datos

### Usuario

```
{  
  
  mail: String,  
  
  pass: String,  
  
  fechaCreacion: Date,  
  
  nombres: String,  
  
  apellidos: String,  
  
  institucion: String,  
  
  conf:  
  {  
  
    estilo: String,  
  
    ayuda: Boolean,  
  
  }  
  
}
```

### Matriz de marco lógico (MML)

```
{  
  
  tema: String,  
  
  descripcion: String,  
  
  fechaCreacion: Date,  
  
  avance: Number,  
  
  creador: mongoose.Schema.Types.ObjectId,  
  
  compartido:  
  
  [{
```

```
        user:mongoose.Schema.Types.ObjectId,

        permiso: String, // R W

    }},

    datos:

    {

        problema:

        {

            antecedentes:String,

            situacioActual:String,

            planteamiento:String,

        },

        justificacion:String,

        alcance:String,

        costo:String,

    },

    historial:

    [{

        usuario:mongoose.Schema.Types.ObjectId,

        cambio:String,

        fecha:Date,

    }],

    involucrados:

    [{

        nombre: String,
```

```
        grado: Number,

        tipo: String,

    }],

    arbolProblema:

    {

        type:mongoose.Schema.Types.ObjectId,

        rel:'Arbol'

    },

    arbolObjetivo:

    {

        type:mongoose.Schema.Types.ObjectId,

        rel:'Arbol'

    },

    estrategia:

    {

        tiempo:

        {

            tipo:String,

            cantidad:Number,

        },

        costo: Number,

        acciones:

        [{

            type:mongoose.Schema.Types.ObjectId,
```

```
        rel:'Accion'
    }},
},
fin:
[[
    texto:String,
    seguimiento:
    {
        type:mongoose.Schema.Types.ObjectId,
        rel:'Estrategia'
    },
}],
proposito:
{
    texto:String,
    seguimiento:
    {
        type:mongoose.Schema.Types.ObjectId,
        rel:'Seguimiento'
    },
},
componentes:
[[
    texto:String,
```

```

    seguimiento:
      {
        type:mongoose.Schema.Types.ObjectId,
        rel:'Seguimiento'
      },
  },
},
actividades:
[[
  texto:String,
  seguimiento:
  {
    type:mongoose.Schema.Types.ObjectId,
    rel:'Seguimiento'
  },
}],
}

```

## Árbol

```

{
  centro: String,
  superior:
  [[
    texto:String,
    nivel: int,
    padre: mongoose.Schema.Types.ObjectId
  ]
]
}

```



```
    }},  
    inferior:  
    {{  
        texto:String,  
        nivel: int,  
        padre: mongoose.Schema.Types.ObjectId  
    }},  
}
```

## Accion

```
{  
    texto:String,  
    tipo:char,  
    estrategiaMML: boolean,  
    MML:  
    {  
        type:mongoose.Schema.Types.ObjectId,  
        rel:'MML'  
    }  
}
```

## Seguimiento

```
{  
    MML:mongoose.Schema.Types.ObjectId,  
    filaMML:String,  
    indicador:String,  
    tipo:String,
```

```

    cumplido:String,
    parciales:
    {
        fechaRevision:Date,
        resultado:String,
        documentos:[
            {
                nombre:String,
                url:String,
            },
            idxMV: Number,
        ],
        mediosVerificacion:[String],
        supuestos:[String],
    }
}

```

## 8. Lista de servicios REST

### Usuario

| Ruta                     | Métodos | Datos        | Sesión   |
|--------------------------|---------|--------------|----------|
| APIMML/usuario/register  | POST    | {usuario}    | {}       |
| APIMML/usuario/update    | PUT     | {usuario}    | {userID} |
| APIMML/usuario/log       | POST    | {mail, pass} | {}       |
| APIMML/usuario/logout    | GET     | {}           | {}       |
| APIMML/usuario/verificar | POST    | {mail}       | {userID} |
| APIMML/usuario/          | GET     | {}           | {userID} |

### MML

| Ruta                    | Métodos | Datos    | Sesión   |
|-------------------------|---------|----------|----------|
| APIMML/mml/new          | POST    | {mml}    | {userID} |
| APIMML/mml/:id/update   | PUT     | {mml}    | {userID} |
| APIMML/mml/:id          | GET     | {}       | {userID} |
| APIMML/mml/             | GET     | {}       | {userID} |
| APIMML/mml/:id/share    | POST    | {userID} | {userID} |
| APIMML/mml/:id/delete   | DELETE  | {mml}    | {userID} |
| APIMML/mml/:id/activate | POST    | {mml}    | {userID} |

## Árbol

| Ruta             | Métodos | Datos             | Sesión   |
|------------------|---------|-------------------|----------|
| APIMML/arbol     | POST    | {idMML}           | {userID} |
| APIMML/arbol/:id | PUT     | {idMML,<br>arbol} | {userID} |
| APIMML/arbol/:id | GET     | {idMML}           | {userID} |

## Acción

| Ruta              | Métodos | Datos      | Sesión   |
|-------------------|---------|------------|----------|
| APIMML/accion     | POST    | {accion}   | {userID} |
| APIMML/accion     | GET     | {idMML}    | {userID} |
| APIMML/acción/:id | GET     | {idAccion} | {userID} |
| APIMML/accion/:id | PUT     | {idAccion} | {userID} |
| APIMML/acción/:id | DELETE  | {idAccion} | {userID} |

## Seguimiento

| Ruta                   | Métodos | Datos                   | Sesión   |
|------------------------|---------|-------------------------|----------|
| APIMML/seguimiento     | POST    | {idMML,<br>seguimiento} | {userID} |
| APIMML/seguimiento     | GET     | {idMML}                 | {userID} |
| APIMML/seguimiento/:id | PUT     | {idSeg}                 | {userID} |
| APIMML/seguimiento/:id | DELETE  | {idSeg}                 | {userID} |

## 9. Lista de rutas angular

| Ruta                    | Descripción   |
|-------------------------|---|
| /#/login                | Página de acceso al sistema   |
| /#/register             | Página de registro del sistema  |
| /#/profile              | Página que muestra la información del usuario                           |
| /#/about                | Página que muestra la información acerca del aplicativo                 |
| /#/mml                  | Página que muestra los proyectos asociados, tanto como los compartidos. |
| /#/mml/:id              | Página que muestra un proyecto según el id y su estado.                 |
| /#/mml/:id/involucrados | Página que gestiona los involucrados de un proyecto según su id         |
| /#/mml/:id/AProblemas   | Página que gestiona el árbol de problemas de un proyecto según su id    |
| /#/mml/:id/AObjetivos   | Página que gestiona el árbol de problemas de un proyecto según su id    |

|                        |   |
|------------------------|---|
| /#/mml/:id/Estrategia  | Página que muestra la estrategia del proyecto por su id   |
| /#/mml/:id/proyecto    | Página que muestra para edición el resultado del proceso de generación para ser guardado o modificado.    |
| /#/mml/:id/indicadores | Muestra todos los indicadores de seguimiento de un proyecto   |
| /#/mml/:id/seguimiento | Permite agregar evidencias de seguimiento y avance del proyecto verificando el estado de los indicadores. |